SEVENTH FRAMEWORK PROGRAMME

FP7-ICT-2009-6

BlogForever

Grant agreement no.: 269963

BlogForever: D4.1 User Requirements and Platform Specifications Report

Revision:	First Version
Dissemination Level:	Public
Author(s):	Hendrik Kalb, Nikolaos Kasioumis, Jaime García Llopis, Senan Postaci, Silvia Arango-Docio, Ilias Trochidis, Vangelis Banos
Due date of deliverable:	**31/12/2011**
Actual submission date:	**Thursday, 22 March 2012**
Start date of project:	01 March 2011
Duration:	30 month
Lead beneficiary name:	Technische Universität Berlin

Abstract:

This report specifies requirements for the BlogForever archive and preservation system. Therefore, requirements descriptions are assembled from several sources like the description of work and former deliverables. Additionally, 26 semi-structured interviews were conducted to identify additional requirements as well as to validate the already described requirements. The report illustrates the method of interview conduction and qualitative analysis. It includes a description of relevant stakeholders and requirement categories.

The identified requirements are specified in a standardised template and modelled with the unified modelling language (UML). Thus, they can be easily explored and utilised by developers. Additional views provide further support for navigation. Overall, the requirements are the foundation for the design phase because they represent the perspective of demand.

The **BlogForever** Consortium consists of:

Aristotle University of Thessaloniki (AUTH)	Greece
European Organization for Nuclear Research (CERN)	Switzerland
University of Glasgow (UG)	UK
The University of Warwick (UW)	UK
University of London (UL)	UK
Technische Universität Berlin (TUB)	Germany
Cyberwatcher	Norway
SRDC Yazilim Arastrirmave Gelistrirmeve Danismanlik Ticaret Limited Sirketi (SRDC)	Turkey
Tero Ltd (Tero)	Greece
Mokono GMBH	Germany
Phaistos SA (Phaistos)	Greece
Altec Software Development S.A. (Altec)	Greece

Revision History

Version	Description of Change	Author	Date
0.1	Draft of Structure, Proposal for Stakeholders	TUB (M. Trier)	18/05/11
0.1.1	Extension of the Preface chapter, Notes/References to D2.1 included	TUB (H. Kalb)	15/09/11
0.2	Restructuring and extending chapter 1	TUB (Kalb)	06/12/11
0.2.1	Chapter 2 written	TUB (Kalb)	08/12/11
0.2.2	Preface and Chapter 1 reviewed	CERN	08/12/11
0.3	Requirements included	CERN	12/12/11
0.9	Many updates and formatting	CERN & TUB	13/12/11
1.0	Finalizing the first draft	CERN & TUB	16/12/11
1.1	Corrections	CERN & TUB	20/12/11
1.2	Minor revisions	TUB	01/02/12

Table of Contents

Preface

In this document the software requirements for the BlogForever software platform are assembled, developed and specified. The structure is adapted from the guidelines of the IEEE 830 SRS - Software Requirements Specification Standard (1998) (IEEE Computer Society, 2009).

The requirement specifications in this document are the outcome of Task 4.1 "User requirements and platform specifications" in Work Package 4 (WP4). According to the description of work, the following approach was chosen to obtain the requirements:

1. Identification of potential individual users and user groups (stakeholders).
2. Gathering of views, needs and expectations through questionnaires and semi-structured interviews. Thereby, identification of functional and non-functional requirements, ranging from usability and accessibility to system security and expected performance.
3. Formal composition of the user requirements and platform specifications. Modelling of the requirements using the Unified Modelling Language (UML).

Requirements limit the range of the valid design, but do not specify any particular design. A requirement specifies any externally visible function or attribute of a system. A design describes a particular subcomponent of a system and/or its interfaces with other subcomponents. The requirement specification has to be correct, unambiguous, complete, consistent, ranked for importance and/or stability, verifiable, modifiable, and traceable (IEEE Computer Society, 2009).

If it is necessary to use the phrase "to be determined" (TBD) in the specification then it should be accompanied by a description of the conditions causing the TBD so that the situation can be resolved, and description of what must be done to eliminate the TBD.

Ranking of requirements can be done by the distinction between essential, recommended, and optional requirements. The specification of a requirement is verifiable if there exists some finite cost-effective process to validate that the system meets the requirement (the requirement can be measured).

To identify the requirements, the following instruments were used:

- Review of the project's description of work (DoW),
- Survey (questionnaires) of blog authors and readers (see BlogForever Deliverable 2.1 Weblogs Survey Report),
- Evaluation of blog technologies (see D 2.1),
- Conceptual work on social network analysis (see D 2.1),
- Literature review of blog life cycles (see D 2.1),
- Interviews with relevant stakeholders,
- Evaluation of written feedback from dissemination activities (e.g. Mokono blog),
- Review of relevant literature, and
- Requirement evaluation by the project partners.

Special emphasis is given on the user requirements section.

1 Introduction

The following chapter describes the aim and the process of requirement specification. Relevant concepts are defined to facilitate a common understanding.

1.1 Purpose and Scope

The purpose of this document is to specify the requirements that have to be fulfilled by the blog aggregation, preservation, management and dissemination system which will be developed in the BlogForever project. Thereby, the requirements describe functional and qualitative necessities of the system but do not postulate any demands regarding the design of the system.

On one hand, the description of requirements should help the intended users to express their needs. Therefore, the requirements have to be understandable by system users. On the other hand, the specification of requirements facilitates the development of the system by providing measurable demands. It narrows down the number of possible designs. Therefore, the specification has to be understandable by developers and the requirements have to be formulated in a way that allows the assessment of the fulfilment of requirements. The fulfilment of requirements that specify the quality at runtime (e.g. performance) can be evaluated during the use cases described in chapter 1.4.

The requirements specify the software characteristics of the weblog aggregation, preservation, management and dissemination platform. The software will feature the following added-value services[1]:

- Weblog content browsing including advanced searching, sorting and topic clustering,
- Context-based provision of selected subsets of the preserved information, and
- Provision of facilities for knowledge extraction, classification, and data interaction.

The intended weblog aggregation, preservation, management and dissemination system comprises[2]

- An aggregation component (weblog spider) that captures the weblog data from the blogosphere,
- An ingestion and input component to ensure the captured data is correctly transferred from the aggregating component to the storage component,
- A storage component to archive the data efficiently (digital repository),
- A maintenance component to manage the stored content for preservation, and
- A dissemination and output component to provide the access to the data to end and third-party users and services.

Therefore, requirements can address one or more of these components.

[1]BlogForever "Description of Work" Part B, p. 12.

[2]BlogForever "Description of Work" Part B, p. 19.

1.2 Overall approach

The requirement specifications assembled in this document are the results of the collective effort of various project members. The requirements were developed in Work Package 4 (WP4) using both primary and secondary data as foundation.

26 semi-structured interviews were conducted with interviewees from six stakeholder groups to survey primary data for the identification of user requirements. Afterwards, the qualitative data were analysed and 114 different requirements were specified. Chapter 3 reveals our approach for the interview conduction and the analysis of the qualitative data.

Additionally, secondary data from internal and external sources were reviewed. The project already indicates requirements in the documents of the description of work (DoW) and the survey implementation report (Deliverable 2.1: Survey Implementation Report). The requirements were collected and rewritten to harmonise them with other requirement specifications in this document and to enhance developers' understanding. Furthermore, existing literature was reviewed for further support of the identified requirements.

All the identified requirements were described in a homogenous structure. Additionally, requirements were modelled with unified modelling language (UML) when suitable. Thus, the formalisation of requirements was enhanced in order to facilitate design and implementation of the platform.

Last but not least, developers reviewed the requirement specifications to ensure understandability and appropriateness.

1.3 Requirement Categories

The requirements are classified into different categories. On the one hand, this facilitates the process of identification because each category can be examined separately and, hence, the cognitive demand is reduced. On the other hand, the multitude of identified requirements can be better explored by the classification.

In the identification process, analysts normally stick to categories and sometimes do not identify requirements besides them. Therefore, it is important that the categories encompass all potential requirements. This problem was addressed by reviewing literature (IEEE Computer Society, 2009; Anon., 2005; The Open Group, 2009; McEwen, 2004) as well as other requirement specifications (Fitzner, 2010) for possible categories. In the following paragraphs, the consolidated categories are described. Thereby, subcategories are also included. Figure 1 gives an overview of the main categories.

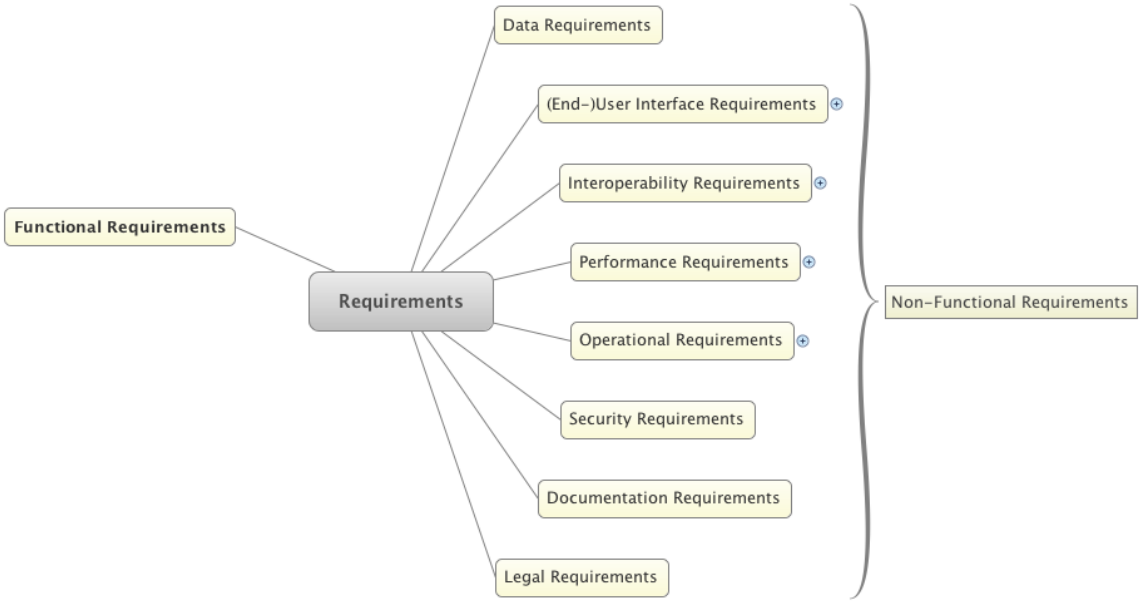


Figure 1: Requirement categories

Functional Requirements "should define the fundamental actions that must take place in the software in accepting and processing the inputs and in processing and generating the outputs" (IEEE Computer Society, 2009, p.16). They can be partitioned into sub-functions or sub-processes if appropriate. The partitioning does not imply decisions for the design of the software. A required software function could be strongly connected to the data that the function should process as well as to the interface that a user or admin of the software needs to use the function. This is especially true if user requirements were development based on user perceptions. Therefore, the restrictions of a classification are attenuated and it is also allowed to sort a requirement into more than one category.

Beside functional requirements, various **non-functional requirements** can exist to define an information system. In the following, they are differentiated into more specific categories to facilitate the requirement identification process by avoiding the ambiguousness of a single non-functional requirement category.

Data requirements describe visible data and data the user needs to export or to process. They specify content and semantics but not the format in which the data are processed in the system. Data requirements do not assume nor mandate a specific type of database (Anon., 2005).

Interoperability requirements specify the ability of the system to share information and services (The Open Group, 2009). Subcategories can be external interface requirements and standard requirements. The former specifies the external interfaces of the system that are required to enable interoperability with other systems. The latter focuses on standards that the system must support to provide interoperability with other systems.

The **(end-)user interface requirements** specify the interface between the software and the user. This includes software configuration characteristics (e.g. required screen formats, page or window layouts, content of any reports or menus, etc.) as well as aspects that optimize the ease of use of the interface (e.g. specific layout which is familiar to the user). A subcategory for further specialisation of the requirements could be style of the product respectively of the appearance. Another subcategory can be the ease of use or the ease of learning. Style of the product describes

requirements for the appearance of the user interface that are not driven by specific user needs regarding the usability of the system (e.g. requirements based on company guidelines). The latter usability requirements "describe the ease with which the system can be learned or used" (McEwen, 2004).

Performance requirements "specify both the static and the dynamic numerical requirements placed on the software or on human interaction with the software as a whole" (IEEE Computer Society, 2009, p.16). Dynamic numerical requirements specify an amount of actions that are done in a certain time period, e.g. amount of web pages that are captured in one hour. Hence, dynamic requirements can be encapsulated in the sub-category of **speed/performance requirements**. Static numerical requirements will be specified mainly under the sub-category of **capacity and scalability requirements**. They define how much or how many the system must be able to process or to store. Examples are how many users can access the system at the same time, how many blogs can be captured, managed and preserved by the system or how much storage capacity must be managed by the system (IEEE Computer Society, 2009). Another subcategory is **reliability and availability requirements**. Thereby, reliability "specifies the factors required to establish the required reliability of the software system at time of delivery" (IEEE Computer Society, 2009, p.18). In other words, it "describes the degree to which the system must work for the users" (McEwen, 2004). Availability requirements "specify the factors required to guarantee a defined availability level for the entire system such as checkpoint, recovery, and restart" (IEEE Computer Society, 2009, p.18). Numerical requirements that belong to a specific function should be specified as part of the function requirement and not as part of the performance requirements category or of the sub-categories.

The **operational requirements** specify required capabilities of the system to ensure a stable operation of the system. These requirements should not be covered in the other requirement categories. The operational requirements can be further differentiated in the sub-categories of standard requirements, support and maintainability requirements, and storage and persistence requirements. **Standard requirements** describe the standards the system has to fulfill to facilitate a stable operation of the system. **Maintainability** "specifies attributes of software that relate to the ease of maintenance of the software itself" (IEEE Computer Society, 2009, p.18). The **storage and persistence** category collects specific requirements for the storage of the data, e.g. to prevent data loss.

Security requirements "specify the factors that protect the software from accidental or malicious access, use, modification, destruction, or disclosure" (IEEE Computer Society, 2009, p.20). Examples are the usage of cryptographic techniques, assignation of certain functions to different modules, restriction of the communication between some areas of the program, and checking the data integrity for critical variables.

Next to the requirements that describe or restrict directly the planned archive software, the two additional categories of legal requirements and documentation requirements are more general and contained requirements may probably influence just indirectly the software. Nevertheless, they are not less important for adoption and success of the software. **Legal requirements** cover any constraints through legal regulation, e.g. licenses, laws, patents, etc. and **documentation requirements** relate to the software documentation, e.g. languages that are required (Fitzner, 2010).

1.4 Case Studies for the Evaluation of the System

Each requirement description in this document includes a measurement statement to evaluate the requirement fulfilment by the software. However, the assessment of a measurement statement is often only possible for an already deployed system. Thus, six case studies will be implemented in the project that can be used as a test -bed for assessing the requirements fulfilment.

The case studies are implemented sequentially and increase in their complexity. They start with two small domain specific weblog groups in university environments (less than 100 blogs), followed by three general weblog groups from the business project partners (less than 2000 blogs), and end with a wide blogging group with at least 500.000 weblogs. A full description of the case studies can be found in the DoW of the project[3].

[3] BlogForever "Description of Work" Part B, p. 43.

2 Stakeholder of the Archive Software

Before requirements are collected, analysed and specified, the main stakeholders have to be defined. Eight stakeholder groups were identified based on the DoW and internal discussions in the project. Interviews were conducted with representatives of each of the stakeholder groups. Special emphasis is given on the group that contributes blog content, especially individual blog authors. Figure 2 gives an overview of the different stakeholders. The stakeholders represent a specific role with regard to the archive and, therefore, the roles are distinctive. Nevertheless, a concrete individual or institution can have several roles, e.g. a person can be researcher, blog author, and blog reader at the same time.

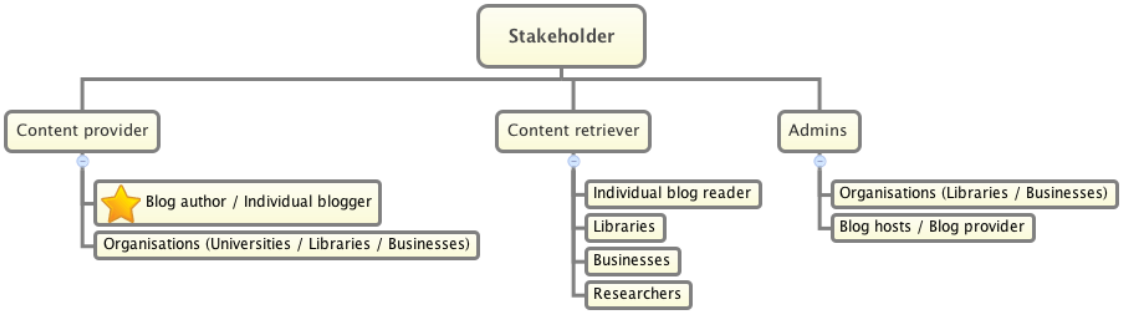


Figure 2: Stakeholders of the archive software

As a first step, the stakeholders of the software were distinguished between content provider, content retriever, and administrator.

Content providers are people or organisations which maintain one or more blogs and, hence, produce blog content that can or should be preserved in the archive. Content providers are owners of their contents and decide whether they wish to contribute their content to a preservation system or not. Therefore, it is crucial to address their needs. For content providers, it is differentiated between individual blog authors and organisations which can have one or more members who blog for them, e.g. business and corporate blogs.

Survey results in D2.1 indicate a majority of **individual blog authors**. Individual blog authors maintain their own blog. Thereby, maintaining means creation of blog posts, answering comments, designing the layout of the blog, etc. Blog authors may also interpret themselves as individual authors even if they maintain their blog in connection with an organisation. It is essential to know what individual blog authors need and expect. Therefore, special emphasis was put on the examination of how blog authors currently behave and what they think or expect by a blog archive.

Organisations can serve as content providers if they maintain their own corporate blogs. Organisations with their own blogs vary from public organisations like libraries and universities to businesses. In order to be considered as content providers, they should all have in common that they are allowed to publish and distribute their blog content, and that they have an interest in its long-term preservation. Their needs have to be considered to support their organisational purposes of preservation and, thus, to increase the probability of contribution.

Content retrievers are people or organisations which have an interest in the content stored in a blog archive and, therefore, they like to search, read, export, etc. that content. The purpose of their

interest can vary broadly. They can be divided into individual blog readers, libraries, businesses, and researchers.

Individual blog readers are people who already read blogs for various reasons, e.g. family, hobbies, professional. A blog reader may also be interested in a blog archive because he/she could find blogs that he/she has read in the past but which are not available anymore. In the future, a blog reader could also be interested in the blog posts at a specific point in time, e.g. his birthday, a scientific breakthrough date, etc. Additionally, an archive could provide special functionalities that go beyond the single blog, e.g. visualize the network of blogs and recommend similar blogs. Thus, individual blog readers are an important stakeholder group from the perspective of the consumption of archived content.

In contrast, **libraries** operate more as a gatekeeper for individual retrievers. They provide access to various kinds of information sources, e.g. books, journals, movies, etc. Thereby, the access includes value added services like selecting and sorting the sources as well as adding metadata. However, libraries in their role as a gatekeeper often do not keep the content themselves, especially in the case of digital resources. Instead, they manage the references to various sources (e.g. literature databases) and if the user would like to retrieve the concrete resource, the library forwards it to the user or retrieves and delivers the resource. Libraries, in their role as gatekeepers, are very important for the adoption of the blog archive. They may have special needs for integration and access.

Businesses also offer value added services based on the available information. But contrary to libraries, they are normally more interested in processing the information to provide a unique selling proposition (USP) to their customers. Such USP could be the detection of trends or sentiments in the business field of the customers. Therefore, they collect or access available information from various sources. A real-time archive of blogs may be an interesting alternative to capturing information themselves. Thus, businesses are a promising stakeholder when business models of blog archives are considered because businesses would possibly pay for special access to archived information. Therefore, the needs and demands of businesses are also considered.

A special group of individual content retrievers is the group of **researchers**. Research on blogs can be conducted for various purposes, e.g. observation of social behaviour, inquiry of historical developments and examination of communication behaviour... However, researchers need “good” data for their research. Depending on whether they do qualitative or quantitative analysis, criteria for “good” data could be the amount of data, how representative they are, or if the author could be identified. The impact of a blog archive will increase enormously for scientific purposes if researchers’ requirements are considered carefully.

Next to the people who contribute to the archive or who utilise the archive, **administrators** are considered as another important stakeholder group for requirement identification. Administrators (admins) maintain installed software and will probably be responsible for a stable and robust operation of the preservation system. Thus, administrators have a different perspective on the requirements of the software and may emphasise more on technical issues, e.g. scalability. Additionally, admins can be more informative with regard to benchmarking data of current usage. Especially **admins of blog hosts** could provide valuable data about the current blogging landscape. Therefore, it is further distinguished between admins of blog hosts and admins of organisations. Thereby, the focus for the latter is on these organisations that would probably run a blog archive. Libraries that preserve digital information and businesses that process social media were identified as relevant organisations.

3 Method to inquire the user requirements

This chapter describes how the user requirements based on the interviews were developed.

The DoW describes several requirements from a project planning perspective. Additionally, D2.1 deduces some social network analysis requirements from a conceptual perspective and indicates several additional requirements in the results of the quantitative study. However, it was lacking a systematic examination of user requirements of a blog archiving, preservation, management and dissemination system. Therefore, an explorative qualitative study was conducted with the following steps

1. Stakeholder identification and requirement categorisation,
2. Interview conduction,
3. Qualitative data analysis, and
4. Preparation of the analysis results.

Before the interviews were conducted, it was necessary to describe the field of examination for the study. On one hand, it is necessary to know who are the stakeholders of our archive and, therefore, should be interviewed regarding their needs, demands and anxieties. On the other hand, an overview of requirement categories facilitates the creation of the interview structure as well as the analysis of the qualitative data.

The concept of stakeholder can be used in a narrower or broader sense. Stakeholders in a narrow sense are the individuals or groups that an organisation or system depends on. Stakeholders in a broader sense include every individual or group that could influence or could be influenced by the organisation or system (Freeman & Reed, 1983). For our study, the focus was on a narrow view and, therefore, especially these groups were considered which have power to influence the success of the system, legitimacy with respect to the operation of the system, and possible urgent needs (Mitchel et al., 1997). The stakeholder groups were identified and refined through internal discussions among the project members. The different groups are presented in chapter 2.

The description of requirement categories serves as a foundation for the development of interview questions. Different interviewers conducted the interviews and, therefore, the interviews had to be structured with specific questions to support a harmonisation of the interview results. The requirement categories are described in chapter 1.3.

Each project partner had already some contacts that fit properly to a stakeholder group (e.g. blog provider have access to bloggers, university member know social media researchers, etc.). Therefore, the project partners suggested potential interviewees. The suggestions were reviewed and a set of people was selected to represent the different stakeholder groups. Additionally, the fact that the potential interviewees came mainly from existing contacts increased their willingness to participate and to answer the interview questions. Therefore, each partner was requested to invite and to interview the people that the partner had suggested. Thus, it was possible to conduct 26 prolific interviews. However, this approach implied that the interviewers were often laymen with respect to the conduction of an interview. Hence, an extensive interviewer guide was created to ensure a common interview quality. Additionally, the interviewers were further assisted on an individual level (e.g. problems in the understanding of questions were solved, technical questions for the recording were answered, etc.).

The interviewer guide facilitated the conduction of interviews by several interviewers. The guide was developed in collaboration with an expert of qualitative analysis and contains

1. Pre-test instructions for the interviewers,
2. Interview instructions and assistance,
3. A consent script,
4. The interview schedule with the questions, and
5. An interview debriefing sheet.

The pre-test instructions request the interviewer to check the recording technique for the interview and to test the interview under realistic circumstances. Thus, it should be ensured that the each interviewer is familiar with the schedule of the interview and could clarify any problems before the interview. The pre-test instructions can be found in appendix A.1.

Additionally, the interview instructions and assistance provide a step-by-step guidance to conduct the interview as well as additional hints, e.g. how to motivate the interviewee during the interview or how to build follow-up questions to acquire further information based on an interviewee statement. The interview instructions and assistance can be found in appendix A.2.

The consent script contains the text that should be read to the interviewee before the recording. It informs the interviewee about the purpose and effort of the interview, and asks for agreement on recording the interview or at least the agreement on taking notes of the interviewee's answers. The interview could only be conducted if the interviewee gave her/his consent. The consent script can be found in appendix A.3.

The interview schedule consists of questions that have been developed based on the requirement categories and on the identified stakeholders. Therefore, eight different schedules have been developed for the eight stakeholder groups. Each interview schedule starts with questions about the context of the interviewee. The interview schedules can be found in the appendices A.4 - A.11.

The interview debriefing sheet should ensure that on one hand the recordings and notes from the interviews are stored in a consistent way and on the other hand problems and peculiarities are documented. The interview debriefing sheet can be found in appendix A.12.

Overall, 26 interviews were conducted. The interviewees were proposed and invited by several project members. Preferably, the project member, who invited the interviewee, also conducted the interview. The sample of interviews was selected in a way that each of the defined stakeholder groups is represented by at least one interviewee. Table 1 shows the distribution of interviews on the stakeholder groups. The high relevance of blog authors is represented by a higher number of conducted interviews.

Table 1: Conducted interviews

Stakeholder group		**Number of interviews**
Content provider	Blog author	7
	Organisation	3

Content retriever	Blog reader	3
	Business	4
	Library	3
	Researcher	2
Admin	Organisation	2
	Blog Host	2

After the interviews were completed, each interviewer could decide if he/she would analyse the interviews by herself/himself or if the interviews should be analysed centrally at TUB. Interviews that should be analysed at TUB had to be transcribed and if necessary translated. Interview analyses that were performed by the interviewer could be done on the recordings and notes without transcription. The optional decision about transcription and translation by the project members was necessary due to time and budget restrictions. Overall, the interviews have been analysed by six different persons. The results of the distributed analyses were integrated through a collaborative process of the analysts.

The method for analysing the interviews is based on the grounded theory (Glaser & Strauss, 1967). Hence, requirements were not deduced a priori and tested by the interviews. Instead, the requirements were developed explorative from the interview transcriptions. Thereby, an incremental approach was followed. Requirements developed from the first interview transcription were tested in the second transcription for additional support. Additionally, the first transcription was checked again with newly generated requirements from the second interview. This loop was conducted again for the third and each additional transcription. Thereby, the analysis was facilitated by Atlas.ti[4], which is a software for qualitative data analysis that.

A template facilitated a consistent way of describing identified requirements. The template can be found in appendix A.13. It consists of

- The categories a requirement belongs to (see chapter 1.3).
- The degree of necessity.
- A full verbal description of the requirement.
- The stakeholder that mentioned the requirement (see chapter 2).
- The justification/foundation for the development of the requirement.
- A measurement statement.
- Requirement authors.

The degree of necessity indicates the priority for the users. It does not automatically lead to a decision about the implementation of the requirement but it should be taken into consideration for the decisions in the design phase. The levels for the degree of necessity are essential,

[4]http://www.atlasti.com/de/

recommended, and optional. **Essential** implies that the software will not be acceptable for the stakeholder unless these requirements are provided in an agreed manner. **Recommended** implies that these are requirements that would enhance the software product, but would not make it unacceptable for the stakeholder if they are absent. **Optional** implies a class of functions that may or may not be worthwhile. This gives the supplier the opportunity to propose something that exceeds the software requirement specification.

For the requirements based on interviews, the justification/foundation consists of representative word-by-word of translated statements from the interviews. This should support the verification of the requirement development, e.g. by third parties.

The measurement statement facilitates the evaluation of the requirement fulfilment. Therefore, the statement has to be measurable value or has to state a method on how the fulfilment will be evaluated. A requirement is presumed as fulfilled if the measurement statement is fulfilled.

The described requirements were additionally modelled as use cases with the unified modelling language (UML). The specification in a formal modelling language facilitates the utilisation of the requirement descriptions in the following project phases (e.g. design) because it reduces the degrees of freedom for the interpretation and it is more familiar for developers. The deduced UML use cases include five main actors. Table 2 gives an overview about the actors, their short names in the UML models, and a full description of the actors.

Table 2: UML-descriptions - Actors involved in the system

Actor	Short Name	Description
Blog Content Aggregator	Spider	The component that can aggregate text contents of the blog pages along with the comments, style information or embedded material etc.
Blog Archiving Component	Platform	The archive system where the blog posts, their metadata are kept.
Platform User - Content provider and related services	Content Provider	People who provide content (write in blogs), that should be preserved
Platform User - Content retriever	Content Retriever	The user that retrieves content from the Platform
System Administrator	Administrator	The person that is responsible for management of the whole system

In the following chapter 4, the requirements identified from the interviews are listed. This is followed by the requirements from the DoW in chapter 5, before the requirements identified in D2.1.

4 Requirement Specifications from the interviews

In the following chapter, the requirements that were identified from the interview analysis are specified. Thereby, the requirements are organised in subchapters that correspond to the requirement categories presented in chapter 1.3. UML models supplement the requirement descriptions. Requirements were modelled with activity diagrams if appropriate. The use case diagrams on the beginning of each subchapter give an overview on the modelled requirements including the responsible actors.

4.1 Functional Requirements

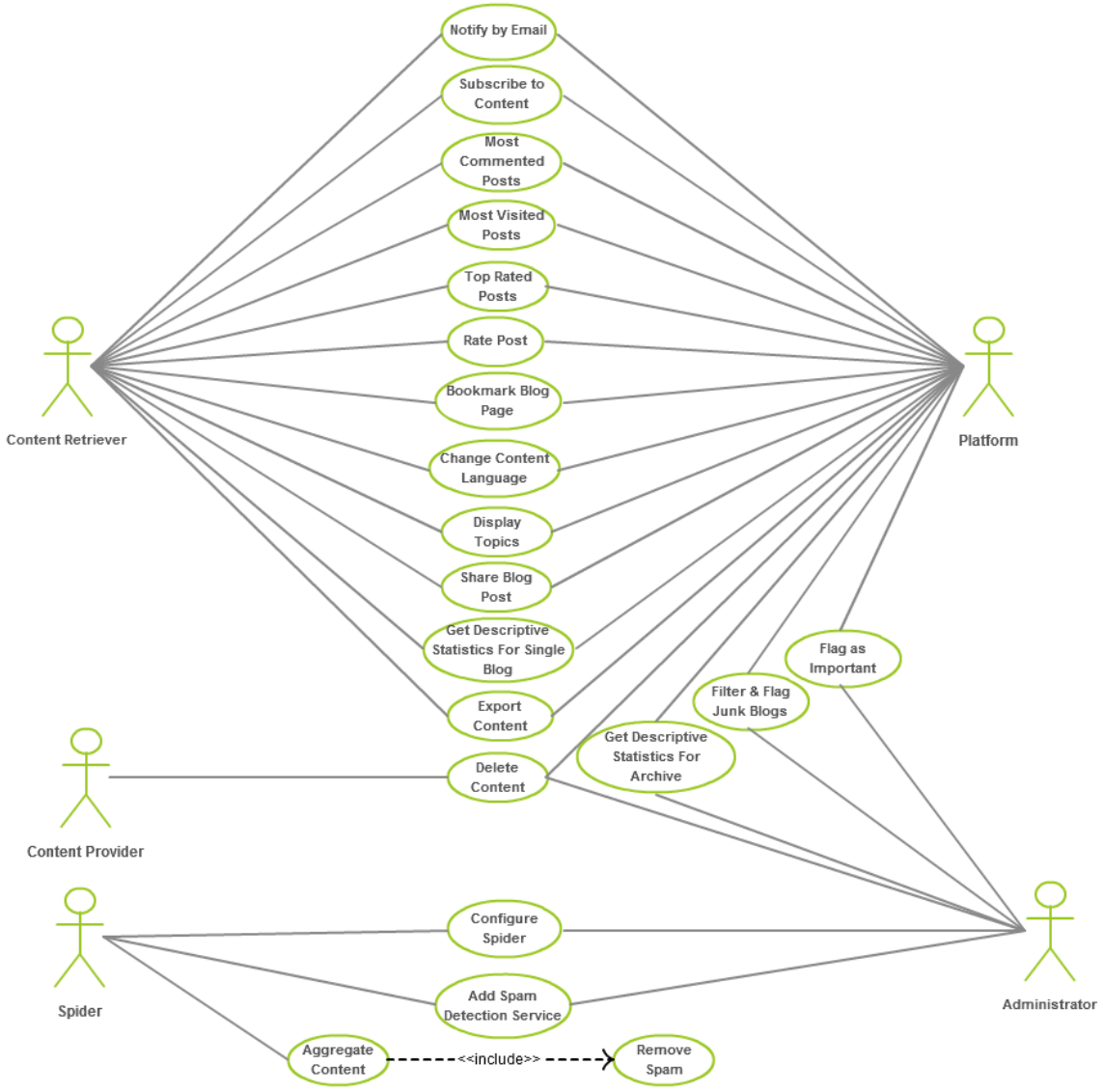


Figure 3: Use Case Diagram 1 for the Functional Requirements

Figure 4: Use Case Diagram 2 for the Functional Requirements

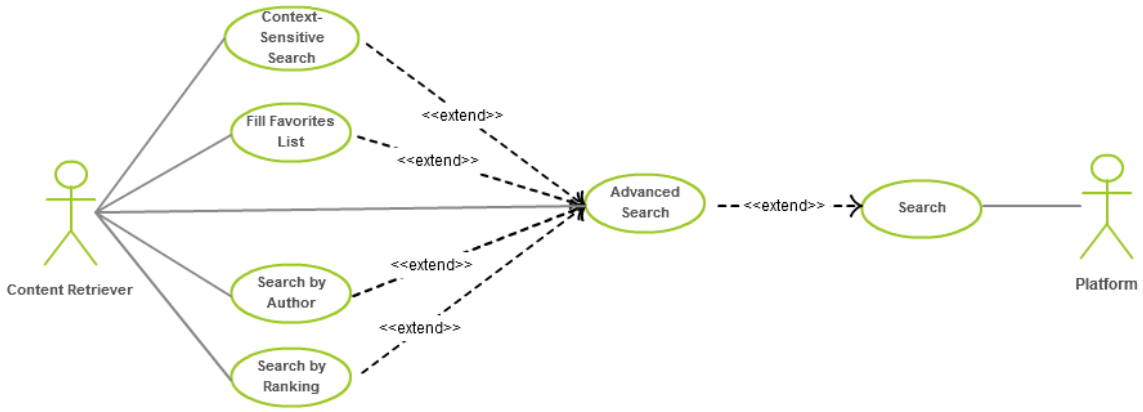


Figure 5: Use Case Diagram 3 for the Functional Requirements

FR1 – Deletion by the blog author

Requirement category	Functional Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	The blog author must have the ability to arrange the deletion of a blog post or a whole blog that is archived in the system. The author can only arrange the deletion for own blogs or posts. The post or blog has to be deleted permanently.
Stakeholder	Admin Organisation, Blog Author
Justification / Foundation	Admin: "we do not have a policy of automatically deleting blogs, unless the owner of a blog asks us to do so or, indeed, does it for themselves" Blog Author: "I would be happy to just have a point of contact to say: "Actually, I would like to take that down."" Blog Author: Either to contact an admin via email or to exist a menu in a control panel with the option delete of the backup, with the usual prompts, "Do you really want to delete this backup?" Blog Author: "And then, secondly, the function that actually does the removing, whereas that function has a built-in safety check like: "Delete? Delete forever? Really? Are you sure? This will delete it forever!" And then, when I do that, don't actually delete it, but just change its status to "show to nobody". Because, I will make a mistake even though it said "are you sure, are you sure"… I will at some point make a mistake. So you shouldn't actually delete it, you should just make it not viewable, unless the decision to delete has been overturned."
Assessment / Measures	An author can arrange the permanent deletion of her/his blog or blog posts.
Author(s) of the requirement description	• Hendrik Kalb • Stella Kopidaki

Use Case UCFR 1.1: Delete Content

Use Case UCFR 1.1		Delete Content
Description		Content Providers are allowed to delete their own content
Actors		Content Provider, Administrator, Platform
Assumptions	**Preconditions**	1. The Content Provider owns a blog
	Postconditions	1. The Platform has deleted the content in the archive
Steps		1. The Content Provider sends a deletion request to the Administrator 2. The Administrator views the Content Provider's blog 3. The Administrator clicks on a "Delete" button 4. The Platform deletes all the content related to the blog
Parent		
Trigger		1. The Content Provider sends a deletion request to the Administrator
Variations		
Issues		

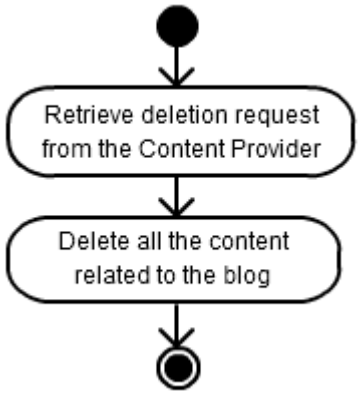


Figure 6: Activity Diagram for deleting a blog

FR2 – Capturing filter

Requirement category	Functional Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	There must be a possibility that the captured content can be filtered. Filtering can be necessary to avoid irrelevant content, and for reasons of disk space and efficiency. The filter depends on the purpose for the respective operator of the archive. Therefore, it cannot be a general filter.
Stakeholder	Admin Organisation
Justification / Foundation	Admin: "There is filtering on the way in, we don't do any kind of filtering on the archive itself as such. [...] There are two reasons. One, because we have got archives which are designed for a particular purpose and then you only put things in if they meet that purpose - institutional repository, say. And the other is, for reasons of time, disk space, or efficiency, where we do not want to archive stuff that we do not need to."
Assessment / Measures	A filter can be added that filters the content when it is captured.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCFR 2.1: Add Filter

Use Case UCFR 2.1		Add Filter
Description		Administrators can add filter to be applied on the captured content for disk space and efficiency purposes
Actors		Administrator, Platform
Assumptions	**Preconditions**	1. The Administrator has displayed the Admin Interface
	Postconditions	1. The Administrator has added filter
Steps		1. The Administrator defines a filter that specifies what kind of information will be filtered on the captured content 2. The Administrator saves the changes
Parent		
Trigger		1. The Administrator creates a filter
Variations		
Issues		

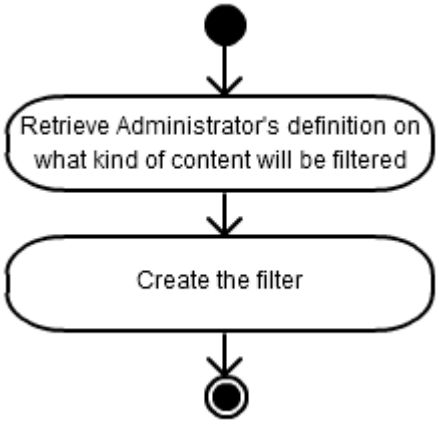


Figure 7: Activity Diagram for adding filter

Use Case UCFR 2.2: Filter Content

Use Case UCFR 2.2		Filter Content
Description		If filter(s) exists to be applied to captured content, the BlogForever platform filters captured content before archiving
Actors		Platform
Assumptions	**Preconditions**	1. The Spider has captured content
	Postconditions	1. The Platform has filtered the captured content
Steps		1. The Platform retrieves the captured content from the Spider 2. The Platform applies the filter(s) to the captured content
Parent		
Trigger		The Platform retrieves the captured content from the Spider
Variations		
Issues		

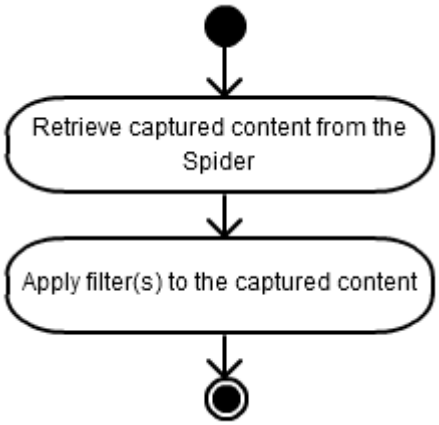


Figure 8: Activity Diagram for filtering captured content

FR3 – Descriptive statistics for the archive

Requirement category	Functional Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	Descriptive statistics (e.g. amount of blog posts in the archive) should be calculated on demand.
Stakeholder	Admin Organisation
Justification / Foundation	Admin Organisation: "for something like our institutional repository, you might say, well, every now and then, people come and say: how many papers do you hold from physics or how many papers did you put into the archive in the calendar year 2010. And then you want to be able to go and inspect the archive and produce the answers to those kinds of questions." Admin Organisation: "We would analyse what is in the archive where we either need to be able to provide data to somebody about the content of the archive, because they have a regulatory or other need to know."
Assessment / Measures	Functionality to calculate descriptive statistics is available.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCFR 3.1: Get Descriptive Statistics for Archive

Use Case UCFR 3.1		Get Descriptive Statistics For Archive
Description		Administrators can get descriptive statistics about the archive
Actors		Administrator, Platform
Assumptions	**Preconditions**	1. The Administrator views the Admin Interface
	Postconditions	1. The Administrator has viewed the descriptive statistics about the archive
Steps		1. The Administrator clicks on "Statistics" button/link/tab 2. The Platform retrieves statistics by executing queries 3. The Platform displays all the statistical information about the archive to the Administrator
Parent		
Trigger		The Administrator clicks on a button/link/tab
Variations		

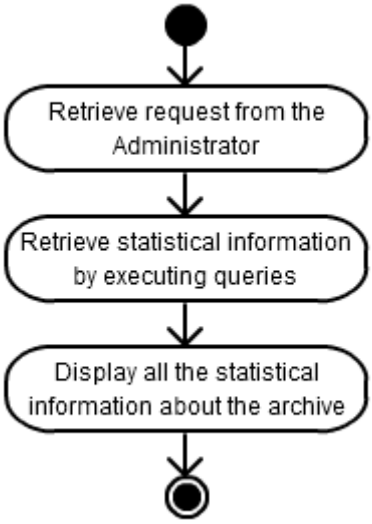


Figure 9: Activity Diagram for getting descriptive statistics about the archive

FR4 – Blog export

Requirement category	Functional Requirement, Interoperability Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	It should be possible to the export a blog with all its parts. The format of the export should be processable by other applications.
Stakeholder	Admin Organisation
Justification / Foundation	Admin Organisation: "we want to be able to give them the data that represents their blog: the words and images, in such a way that they can take it and instantiate it somewhere else." Library: It would be good to be able to republish content from the archive in a blog quickly, with the press of a button.
Assessment / Measures	A blog with all its parts can be exported in a processable format.
Author(s) of the requirement description	• Hendrik Kalb • Vangelis Banos

Use Case UCFR 4.1: Export Content

Use Case UCFR 4.1		Export Content
Description		The Content Retriever can export blog content in various output formats
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever views a blog page
	Postconditions	1. The Content Retriever has exported content
Steps		1. The Content Retriever clicks on the name of output format (MARCXML, PDF, JPEG, etc.) 2. The Platform converts the content from the default encoding format to the selected output format 3. The Platform displays the converted content or provides a file to download
Parent		
Trigger		The Content Retriever clicks on the name of an output format
Variations		
Issues		

Note: Same use case used in the requirements gathered from the DOW (UC6.1)

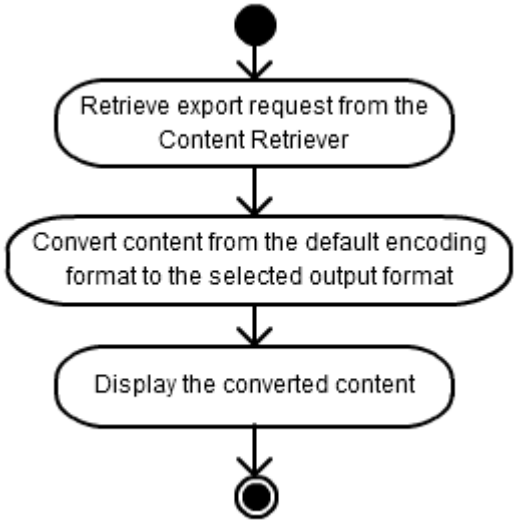


Figure 10: Activity Diagram for exporting content

FR5 – Descriptive statistics for a single blog or blog post

Requirement category	Functional Requirement, Data Requirement, User Interface Requirement
Degree of necessity	() **Essential** (X) **Recommended** () **Optional**
Description of the requirement	Descriptive statistics should be calculated per archived blog. The data should be stored with the blog. Descriptive statistics contain for example: • How many visitors. • Where the people who examine the blog are from. • Which search keywords lead to the blog. • How many comments the blog has. Google Analytics can be taken as example. The information about a blog may include visitors' fidelity (come back rate): percentage of visitors of a blog post that have also visited other posts of the same blog.
Stakeholder	Blog Author, Researcher, Content Retriever, Content Provider, Content Retriever Library
Justification / Foundation	Blog Author: "I use Google Analytics for that already if I am interested, on which I can do all of those things on. You can check whether someone look at it in the UK or from South-East China. And can check what keywords they used to search it." Interview #2 ~18:20: "Another interesting statistic would be popularity, not only in the sense how many people come, but how many people come back, the fidelity..." Content Provider: The main 2 reasons the Research Institute staff understands the point of blogging are: 1. Users' views statistics: the team is excited about their users figures going up (someone is actually reading their posts and they have figures to corroborate this via the views statistics) 2. Currently the Research Institute links the blog new posts with their Facebook page (they just only started 6 weeks ago). Usage/users figures have gone up as well via this method of disseminating their new blogs posts so staff is excited and keeps adding content to the Research Institute blogs so that has made people at the Research Institute to blog internally. Would you like to aggregate, manage & analyse all your users' feedback?

	Please describe an ideal scenario. The answer for this question was: They use whatever the platform use, it is at the individual blog level but they would like to have that across the board: what sort of posts get the most views, what sort of posts get the most comments…better sense of the kinds of things are working (content users like). Usage is very important for the Research Institute. They have all sorts of performance indicators. Would you be interested in an analysis of your blog (or part of your blog) to extract for example: statistics (popularity, visits, etc.), keywords, and sentiments and why? The answer was: Very much, understanding what the Research Institute's users want and then the Research Institute could do more the same, to improve and enhance their blog service and their blog communication strategy. Very interested in statistics and blog popularity. What would you describe as a dream application feature for a blog archiving service? The answer was: Analysis of statistics: easily regenerated stats with visual representation. Blogger is good but extra easy and very visual tools will be ideal. Users' stats from Twitter and FB and the blog. Their entire social media crossed in one place so they could analyse their entirely social media strategy in one place. Showcasing the best of blogs (interesting and subjective). Content Provider: Would you be interested in an analysis of your blog (or part of your blog) to extract for example: statistics (popularity, visits, etc.), keywords, and sentiments and why? Users Stats are very important and they currently get those stats for the Museum blog within the University museums blog. They can see the footprint but they don't have a huge number of users. It will be interesting for them to know how people find them outside the stats (words they use to find them). People who search will be relevant for museum blogs changing overtime and for the preservation of the blog and the potential museums researchers in the future. Tracking figures are relevant for future research. What would you describe as a "killer" application feature for a blog archiving service? Easy to use and that it should provide stats and popularity… Library: Then, we track the access to the database and receive feedbacks from users. We evaluate all these issues, if we have a positive opinion about the database, then we allow access completely.
Assessment / Measures	Descriptive statistics are displayed for a single blog and blog post. Fidelity information is displayed with the blog information.

Author(s) of the requirement description	• Hendrik Kalb • Jaime García • Nikolaos Kasioumis • Silvia Arango-Docio • Senan Postaci

Use Case UCFR 5.1: Get descriptive statistics for single blog

Use Case UCFR 5.1		Get Descriptive Statistics For Single Blog
Description		Content Providers are able to get descriptive statistics for their own blogs
Actors		Content Provider, Platform
Assumptions	**Preconditions**	1. The Content Provider views his/her own blog
	Postconditions	1. The Content Provider has viewed the descriptive statistics about his/her own blog
Steps		1. The Content Provider clicks on “Statistics” button/link/tab 2. The Platform retrieves statistics by executing queries 3. The Platform displays all the statistical information about the blog to the Content Provider
Parent		
Trigger		1. The Content Provider clicks on a button/link/tab
Variations		

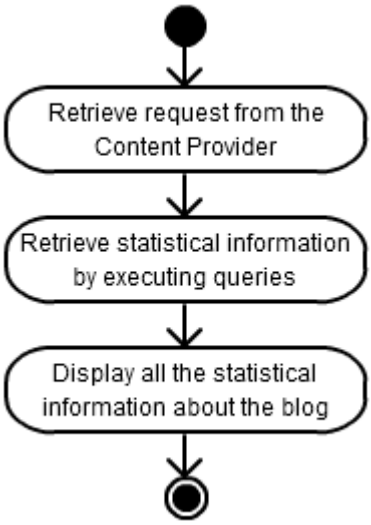


Figure 11: Activity Diagram for getting descriptive statistics about a single blog

FR6 – Processing of Licenses

Requirement category	Functional Requirement, Data Requirement, User Interface Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	Licence has to be respected or followed if a blogger indicate a licence for his blog or parts of the blog. It could be e.g. • That the licence indicate that content has to be excluded from archiving or • That the licence has to be stored and presented with the archived content. If a blog has procedures to prevent search engines to index them (such as robot.txt) the platform should conserve them and apply them to the archived version as well.
Stakeholder	Blog Author, Businesses, Content Retrievers
Justification / Foundation	Blog Author: "I just said that if you had something that needed licensing, I would use a licensing of some sort and you'd hope that you could maintain the citation level should someone want it to cite it in their own work." Businesses: "basically we conform to the rules of the platforms, when we gather data from platforms, there's usually some rules, for example how often you are allowed to query data, so that is not exactly perhaps legal restriction but it's an overall behaviour restriction when you're operating online and there are restrictions on what you can do with the data so that the copyright stays where it originally was, so that has to be restricted" Businesses: (What legal challenges do you face when you access or integrate external content and provide it to your users?) "We follow robot text." Interview #1 ~18:00: "IP restriction is one thing we use, although it's not so great when people change their location and hence their IP, Single, sign-on, robots.txt ..."
Assessment / Measures	The archive can understand and process (important) licences. Licences have to be stored and displayed with the archived content. A blog that originally restricts search engines from indexing it conserves those restrictions within the platform as well.
Author(s) of the requirement description	• Hendrik Kalb • Jaime García • Nikolaos Kasioumis

Use Case UCFR 6.1: Archive Content

Use Case UCFR6.1		Archive Content
Description		The Platform archives the aggregated content from the blog pages
Actors		Platform
Assumptions	**Preconditions**	1. The Platform has retrieved content from the Spider
	Postconditions	1. The Platform has archived the blog content.
Steps		1. If the retrieved content has been archived before the process ends 2. Otherwise, the Platform archives the blog content if the blog content and checksum are OK 3. The Platform checks if the blog author has indicated a license/data protection for the blog content 4. If the blog content has a license, The Platform processes the license and applies to the archived content
Parent		
Trigger		Spider sends the aggregated content to the Platform
Variations		
Issues		

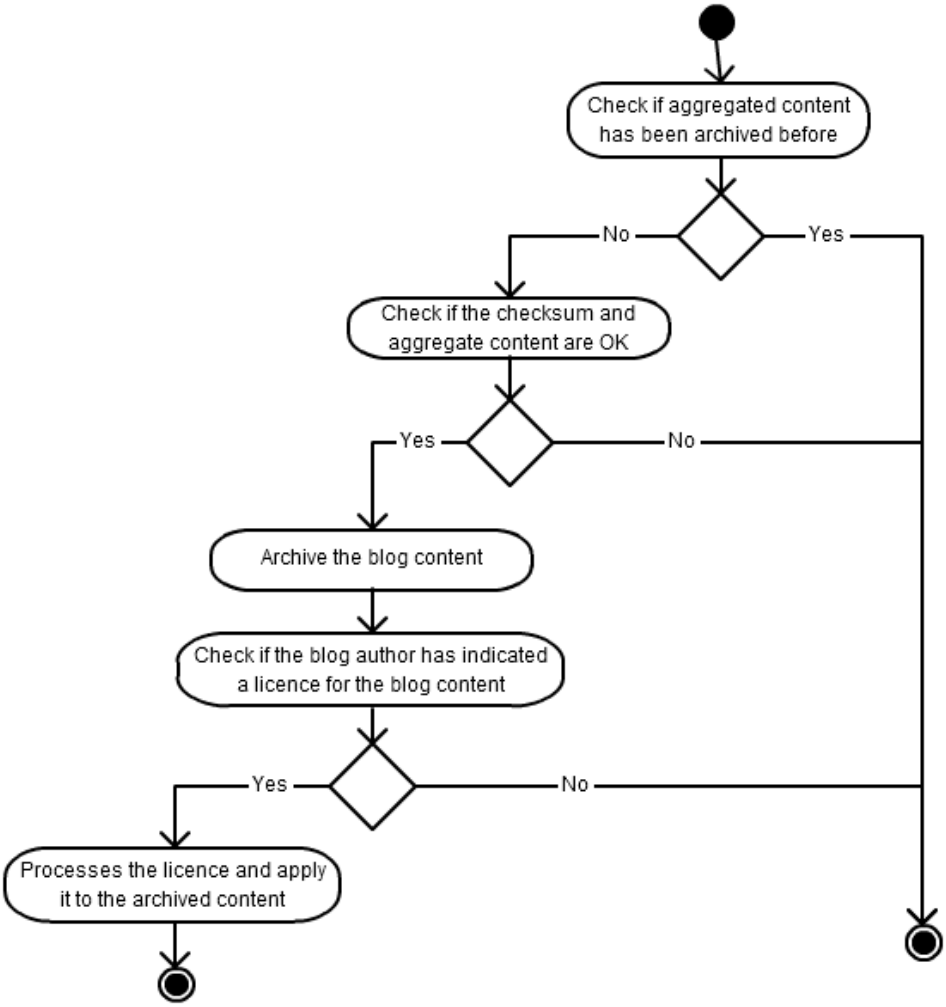


Figure 12: Activity Diagram for archiving blog content

FR7 – User dissemination channels for blog post

Requirement category	Functional Requirement, User Interface Requirements
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	A user (reader) of the archive should have the possibility to disseminate a blog post easily by • Email resp. Newsletter, • Link. • Twitter • Facebook • Google+
Stakeholder	Blog author, Businesses, Blog Reader
Justification / Foundation	Blog author: "But mostly I use a little lines of communication to highlight that something has appeared or something needs to be revisited. [...] I actually do use posters, and meetings and email and newsletters and forum posts. I use all different channels depending who I am trying to contact." Blog author: "When I first started, when I posted something in my newsletter like posting links, and when I'd post a link of so and so I would then send an email to so and so and say "I posted a link to your article in my newsletter." Blog author: "Other than that, I post short Twitter posts to indicate when I have written a longer article." Blog author: "I've been manually posting posts recently in Google+."
Assessment / Measures	Posts can be disseminated (at least) by email, Twitter, and Facebook directly from the presented post (e.g. a button for each dissemination channel).
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCFR 7.1: Share Blog Post

Use Case UCFR 7.1		Share Blog Post
Description		Content Retrievers are able to share blog posts on social networks or send them with emails
Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Content Retriever views a blog post
	Postconditions	1. The Content Retriever has shared a blog post
Steps		1. The Content Retriever clicks on an image of a dissemination channel (email, Facebook, Twitter, etc.) 2. The Platform retrieves necessary information from the Content Retriever (who to send email or any sharing comments for Facebook, Twitter, etc.) 3. The Platform sends email or shares the blog post in the social network using their API's
Parent		
Trigger		The Content Retriever clicks on an image of the dissemination channel
Variations		

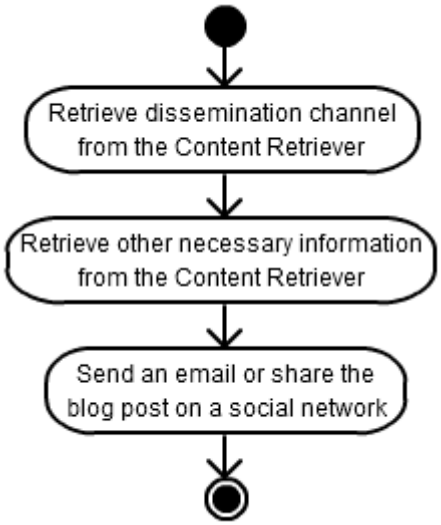


Figure 13: Activity Diagram for dissemination of a blog post

FR8 – Topics (Categories) for blogs and blog posts

Requirement category	Functional Requirement, Data Requirement, User Interface Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	Blogs and blog posts should be sorted by topics (resp. categories). Therefore, the topic of a blog and of a blog post has to be identified, stored with belonging entity (blog or blog post), and displayed to the user (e.g. navigation of the archived content by topic). The user should be able to select a topic and view records (blog posts, comments, and blogs) associated with it. The user interface should enable easy topic selection and browsing. E.g. search by topic "scientific blogs".
Stakeholder	Blog author, Researcher, Business, Blog Reader
Justification / Foundation	Blog author: "being able to pick up themes would be very high on list" Researcher: "Maybe concerning the subject of the blog or a sort of categorization if it's a hobby-blog or if it's a professional blog and also meta-data concerning the owner, if it's an individual blog or a company-blog or institution-blog." Business: "Thematic streams" Business: "our customers want to segment blogs [...] and topics such as industries." Researcher: I would like to see how research/discussions on a topic have progressed during time. Blog Author: The criteria of ranking blogs are according to the topic of each blog.
Assessment / Measures	The archive can identify topics for a blog and for a blog post. These topics are stored with the content and displayed to the archive user. The indexing of the system will support grouping & filtering. The user interface will expose this functionality to users.
Author(s) of the requirement description	• Hendrik Kalb • Vangelis Banos • Stella Kopidaki

Use Case UCFR 8.1: Display Topics

Use Case UCFR 8.1	Display Topics
Description	The Platform is able to categorize blogs and blog posts into topics and displays them to the users

Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Content Retriever has sent a request to display the Platform's main page
	Postconditions	1. The Content Retriever has viewed all the topics in the archive
Steps		1. The Platform retrieves all the indexed topics 2. The Platform displays the main page with the list of topics
Parent		
Trigger		The Content Retriever displays the Platform's main page
Variations		

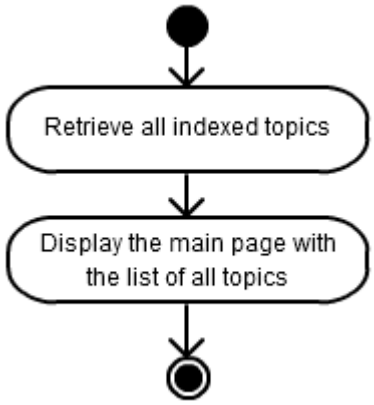


Figure 14: Activity Diagram for displaying topics to the user

FR9 – Content translation

Requirement category	Functional Requirement, User Interface Requirement
Degree of necessity	( ) **Essential** ( ) **Recommended** (X) **Optional**
Description of the requirement	It would be beneficial if content can be automatically translated.
Stakeholder	Blog Author
Justification / Foundation	Blog Author: "Translations between languages would be rather useful."
Assessment / Measures	Content can be translated automatically into several languages.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCFR 9.1: Change Content Language

Use Case UCFR 9.1		Change Content Language
Description		Content language can be translated other languages.
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever views a blog page
	Postconditions	1. The language of the content has been translated into other language
Steps		1. The Content Retriever chooses a language from a "Translate to" box/list 2. The Platform changes the language of the content by using API's (Google Translate, etc.) 3. The Platform displays the content in translated language
Parent		
Trigger		Content Retriever's selection of the functionality
Variations		
Issues		

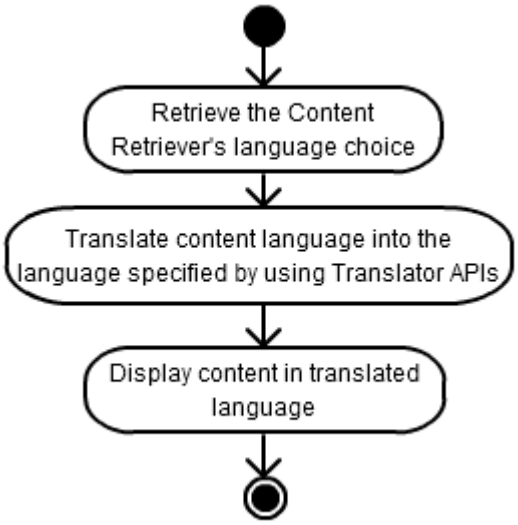


Figure 15: Activity Diagram for changing content language

FR10 – Bookmarking of blog posts

Requirement category	Functional Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	The archive user has the possibility to bookmark a blog post inside the archive or with an external bookmarking service.
Stakeholder	Blog Reader
Justification / Foundation	Blog Reader: "And, if I don't have time to read them before, I bookmark them and go back to them later." Blog Reader: I imagine it like delicious, a list of tags, popular tags, bookmarked items and a search button in order to find what I want.
Assessment / Measures	A user of the archive can bookmark a blog post.
Author(s) of the requirement description	• Hendrik Kalb • Stella Kopidaki

Use Case UCFR 10.1: Bookmark Blog Page

Use Case UCFR 10.1		Bookmark Blog Page
Description		Platform users can bookmark blog pages so that they can read them later.
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever views a blog page
	Postconditions	1. The blog page has been bookmarked
Steps		1. The Content Retriever clicks on a “Bookmark” link/button 2. The Platform bookmarks the blog page for the Content Provider
Parent		
Trigger		Content Retriever clicks on a button/link
Variations		
Issues		

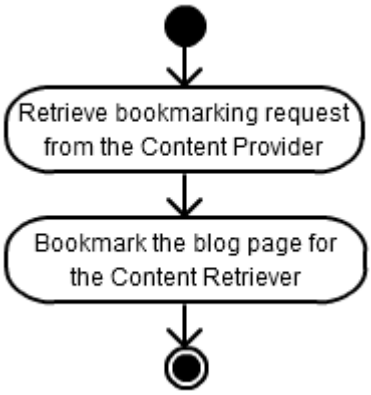


Figure 16: Activity Diagram for bookmarking

FR11 – Recommendation system

Requirement category	Functional Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	User can assess or recommend blogs, blog posts, and comments. Other user can sort or filter blogs, blog posts, and comments by the recommendations of other users (e.g. of users who they trust).
Stakeholder	Blog Reader, Blog Author
Justification / Foundation	Blog Reader: "But I am almost getting the same sort of effect of Twitter these days, there whoever is putting up a sort of "this is interesting: read" or put a brief reason to why, but even briefer than you would in a headline list because you have less than 200 characters." Blog Reader: "I know some platforms produce star ratings, where you can actually rate the post on, usually, a five-star rating. Not really taken that as a, to me, a guide. It is subjective, I think, in some ways. You do not know who it is who is doing the rankings." Blog Author: I would be interested everyone who reads a blog to be able to give a rank for it. Blog Author: It would be interesting ranking to be based not only to the hits but also a user to be able to rank the quality of posts, i.e. the texts, the blog layout, the photos, the videos.
Assessment / Measures	A user can sort or filter blogs, blog posts, and comments by the assessment or recommendations of other users.
Author(s) of the requirement description	• Hendrik Kalb • Stella Kopidaki

Use Case UCFR 11.1: Rate Post

Use Case UCFR11.1		Rate Post
Description		Users are allowed to rate posts
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever has displayed a blog page
	Postconditions	1. The Content Retriever has rated the post.
Steps		1. The Content Retriever rates the post. 2. The Platform receives the rating and calculate ratings accordingly.
Parent		
Trigger		Content Retriever clicks on the symbols (stars, etc.) that represent points from 1 to 5 or 1 to 10
Variations		
Issues		

Note: Same use case used in the requirements gathered from the survey (UC11.1)

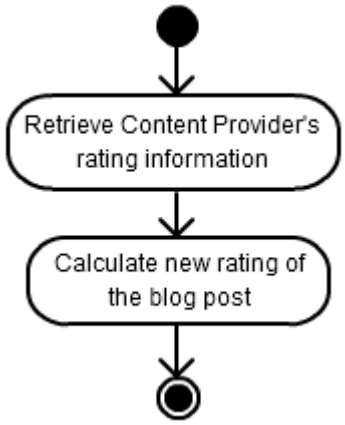


Figure 17: Activity diagram for post rating

Use Case UCFR 11.2: Top Rated Posts

Use Case UCFR11.2		Top Rated Posts
Description		Users can view most rated blog posts.
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever is logged in.
	Postconditions	1. The Content Retriever has viewed top rated posts
Steps		1. The Content Retriever selects “Top Rated” option from the ranking methods list through the Search Interface 2. The Content Retriever clicks on “Search” button 3. The Platform retrieves top rated blog posts 4. The Platform displays the top rated blog posts
Parent		
Trigger		The Content Retriever clicks on a button
Variations		
Issues		

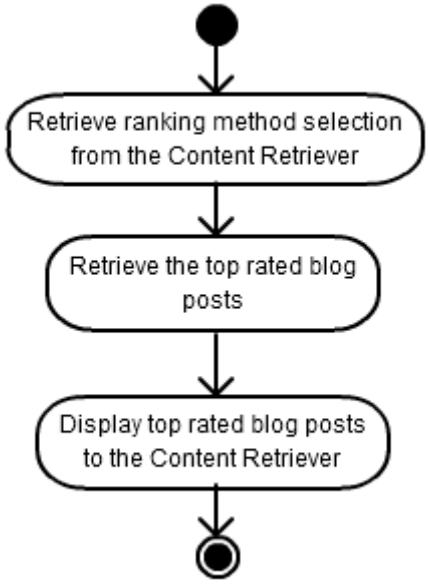


Figure 18: Activity Diagram for getting top rated posts

Use Case UCFR 11.3: Most Visited Posts

Use Case UCFR11.3		Most Visited Posts
Description		Users can view most visited blog posts.
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever is logged in.
	Postconditions	1. The Content Retriever has viewed most visited posts
Steps		1. The Content Retriever selects “Most Visited” option from the ranking methods list through the Search Interface 2. The Content Retriever clicks on “Search” button 3. The Platform retrieves most visited blog posts 4. The Platform displays the most visited blog posts
Parent		
Trigger		The Content Retriever clicks on a button
Variations		
Issues		

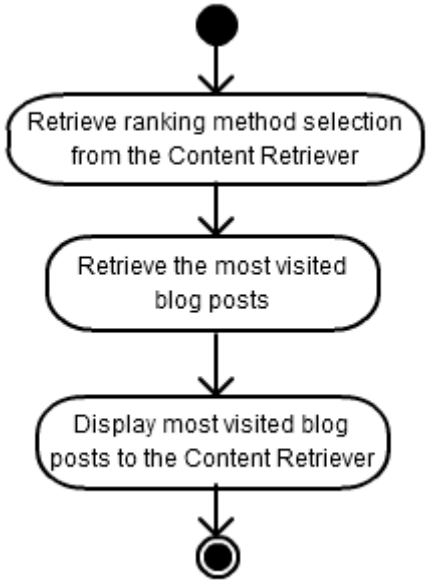


Figure 19: Activity Diagram for getting most visited posts

Use Case UCFR 11.4: Most Commented Posts

Use Case UCFR11.4		Most Commented Posts
Description		Users can view most commented blog posts.
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever is logged in.
	Postconditions	1. The Content Retriever has viewed most commented posts
Steps		1. The Content Retriever selects “Most Commented” option from the ranking methods list through the Search Interface 2. The Content Retriever clicks on “Search” button 3. The Platform retrieves most commented blog posts 4. The Platform displays the most commented blog posts
Parent		
Trigger		The Content Retriever clicks on a button
Variations		
Issues		

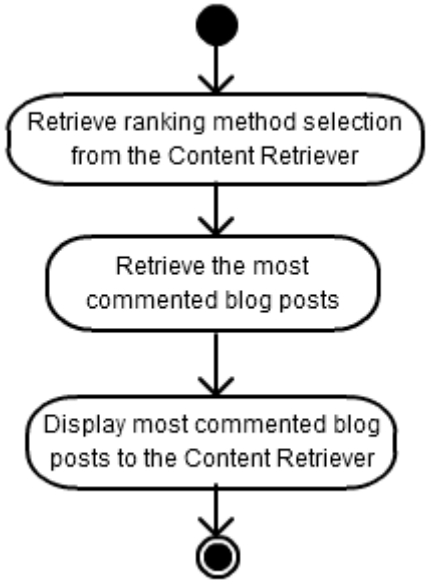


Figure 20: Activity Diagram for getting most commented posts

FR12 – Notification about changes in the archive

Requirement category	Functional Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	A user of the archive can subscribe to blogs or topics and gets notified when new content is added to the blog or topic.
Stakeholder	Blog Reader, Content Retrievers – Libraries, Blog Authors
Justification / Foundation	Blog Reader: (How do you discover new blogs or topics for that list?) "[...] You get notified." Library: We provide an alerting system that alerts users when a new resource is added to the repository. For example, a person who is interested in a certain topic, he/she is alerted when a new magazine content about that topic is added to the repository. Blog Author/Reader: Like social media, the user can subscribe to communities and to be notified for new posts, e.g. in blog X there is a new post. Blog Author: I want a table with notifications to exist, i.e. blog X has new post.
Assessment / Measures	A user of the archive can subscribe to blogs or topics and gets notified when new content is added to the blog or topic.
Author(s) of the requirement description	• Hendrik Kalb • Senan Postaci • Stella Kopidaki

Use Case UCFR 12.1: Subscribe to Content

Use Case UCFR 12.1		Subscribe to Content
Description		A user of the archive can subscribe to blogs/topics in the archive to get notified when a new content is added to the blog/topic
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever views a blog/topic
	Postconditions	1. The Content Retriever has subscribed to the blog/topic
Steps		1. The Content Retriever clicks on a "Subscribe" link/button 2. The Platform adds the Content Retriever to the blog's/topic's subscription list
Parent		
Trigger		Content Retriever clicks on a button/link
Variations		
Issues		

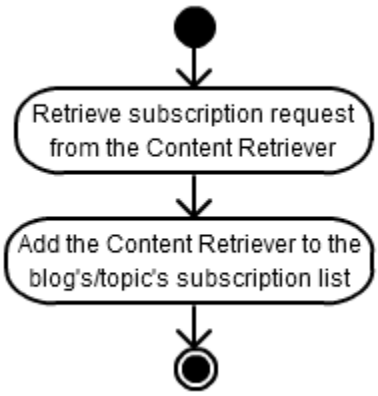


Figure 21: Activity Diagram for subscription to a blog/topic in the archive

Use Case UCFR 12.2: Notify by Email

Use Case UCFR 12.2		Notify by Email
Description		The user of the archive gets notified when a new content is added to a blog or topic
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. A new content has been added to a blog or topic
	Postconditions	1. The Content Retriever has got notified
Steps		1. The Platform retrieves the subscription list of the blog/topic 2. The Platform sends email all the user in the list about the new content
Parent		
Trigger		A new content has been added to the archive
Variations		
Issues		

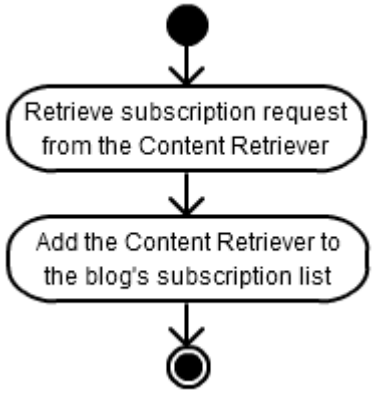


Figure 22: Activity Diagram for notification by email about a new content

FR13 – Keyword/Metadata search

Requirement category	Functional Requirements, User Interface
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	A keyword search allows searching for blogs, blog posts and comments in the archive for specific keywords. The BlogForever platform should index archived content metadata (Title, Creation Date, etc.). The user interface should enable users to search archived data by metadata. When searching in the platform, different elements of blog posts (title, body, comments, etc.) should have different weights. Those weights should be configurable.
Stakeholder	Content Retriever Library, Researcher, Content Retriever
Justification / Foundation	Content Retriever Library: A user who looks in a library system for specific topics, e.g. a political statement of an important person, should find as well related blogs. Researcher: "I used keywords to find lists of science-blogs and I used keywords also to identify blogs belonging to scientists and then I would used keywords like 'research-blog' or 'scientific-blog' or similar" Researcher: "I don't think it's particularly challenging if you realize that there two different levels and that you need both. But I could for instance imagine that you can very quickly forget the blog-level and just go on the post-level. If you search in Google for instance you are searching only on the post-level, that's one of the problems with the Google-search because you are looking for blog-posts and you can't make any difference in where you actually are looking so I might be interested in blogs by mathematicians but there might be also blog-posts about from the area of mathematics written by an IT or computer-scientist which I'm not interested in because he is a computer-scientist and not a mathematician." Researcher: I use http://www.mendeley.com/ and I like it very much! [...] Metadata are populated automatically, I can search / browse etc. Content Retriever Interview #2 ~20:50: "You could pick a word [search keyword] from anywhere, from the text, from the title...comments would also be interesting to search... with different weights..."
Assessment / Measures	A keyword search for blogs, blog posts and comments is available. A user can access an advanced search form/page. There, all available metadata types can be used to generate a complex search query and browse the results. When a user performs a search, the sorting of the results takes into account where within the post the search keywords were found.

Author(s) of the requirement description	• Hendrik Kalb • Vangelis Banos • Jaime García • Nikolaos Kasioumis

Use Case UCFR 13.1: Advanced Search

Use Case UCFR 13.1		Advanced Search
Description		The Content Retriever can construct complex queries and search blog content in the Platform.
Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Content Retriever has displayed Advance Search panel
	Postconditions	1. The Platform has displayed search results
Steps		1. The Content Retriever enters the search keywords 2. The Content Retriever selects a field from the available fields 3. The Platform executes the query retrieved from the Content Retriever 4. The Platform retrieves the search results 5. The Platform groups the search results according to their categories 6. The Platform displays the search results in different tabs, one for each category
Parent		UCFR14.1
Trigger		The Content Retriever enters the keywords and presses a button
Variations		
Issues		

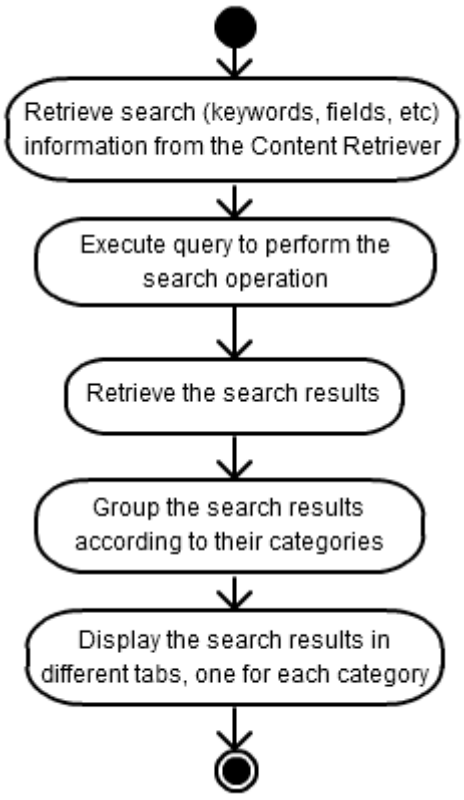


Figure 23: Activity Diagram for advanced searching

FR14 – Full-text search

Requirement category	Functional Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	A full-text search allows searching for blogs, blog posts and comments in the archive for specific keywords.
Stakeholder	Content Retriever Library, Administrators - Organizations
Justification / Foundation	Content Retriever Library: A possibility to search the full-text is needed. Admin: We also support full text search in all archived content.
Assessment / Measures	Full-text search is available.
Author(s) of the requirement description	• Hendrik Kalb • Vangelis Banos

Use Case UCFR 14.1: Search

Use Case UCFR 14.1		Search
Description		The Content Retriever can search blog content in the Platform.
Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Content Retriever is logged in
	Postconditions	1. The Platform retrieves the search result and displays to the Content Retriever
Steps		1. The Content Retriever enters the search keywords 2. The Platform executes the query retrieved from the Content Retriever 3. The Platform retrieves the matching and similar records 4. The Platform displays the search results
Parent		
Trigger		The Content Retriever clicks on a button
Variations		
Issues		

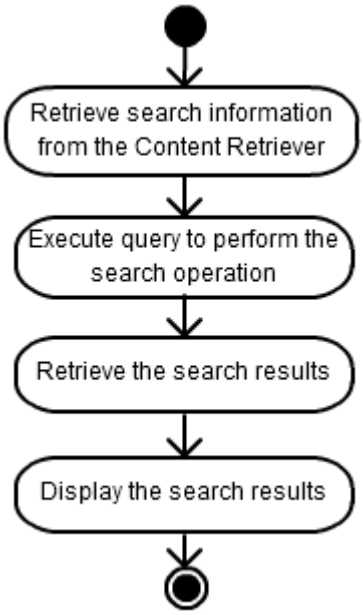


Figure 24: Activity Diagram for searching content

FR15 – Selection of blogs to archive

Requirement category	Functional Requirements
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	A possibility to select the list of blogs that should to be captured and archived is needed. This list should be adaptable because the addition of blogs could be necessary at a later point in time.
Stakeholder	Content Retriever Library, Content Retriever Researcher
Justification / Foundation	Content Retriever Library: I do not want to archive every blog. I have to be able to tell the capturing system which blogs to archive, and how often, etc. Content Retriever Researcher: "I have used manually compiled lists of science-blogs"
Assessment / Measures	The selection of blogs that are captured and archived can be adapted by the archive administration.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCFR 15.1: Configure Spider

Use Case UCFR 15.1		Configure Spider
Description		The Spider can be configured so that it can support multiple platforms and their different versions
Actors		Spider, Administrator
Assumptions	**Preconditions**	1. The Spider has a configuration panel
	Postconditions	1. The Spider enables the modular functionality to process the selected type/version of blog provider
Steps		1. The Administrator views the Spider's Configuration Panel. 2. The Administrator defines how many blogs to be aggregated in a certain time 3. The Administrator defines how much content will be aggregated (comments, style, embedded material, snapshots etc.) 4. The Administer defines topics/subjects for blogs to be aggregated 5. The Administrator selects supported blog provider options. 6. If selected blog provider has different versions that should be dealt with separately, a version list also displayed. 7. The Administrator selects different versions of blog provider platform that the spider will support. 8. In case of the Spider does not support the selected version, the Administrator select the "Try similar versions" option. 9. The Administrator saves the changes 10. The Spider configures itself
Parent		
Trigger		Administrator's selection of the functionality
Variations		
Issues		

Note: Same use case used in the requirements gathered from the survey (UC2.1)

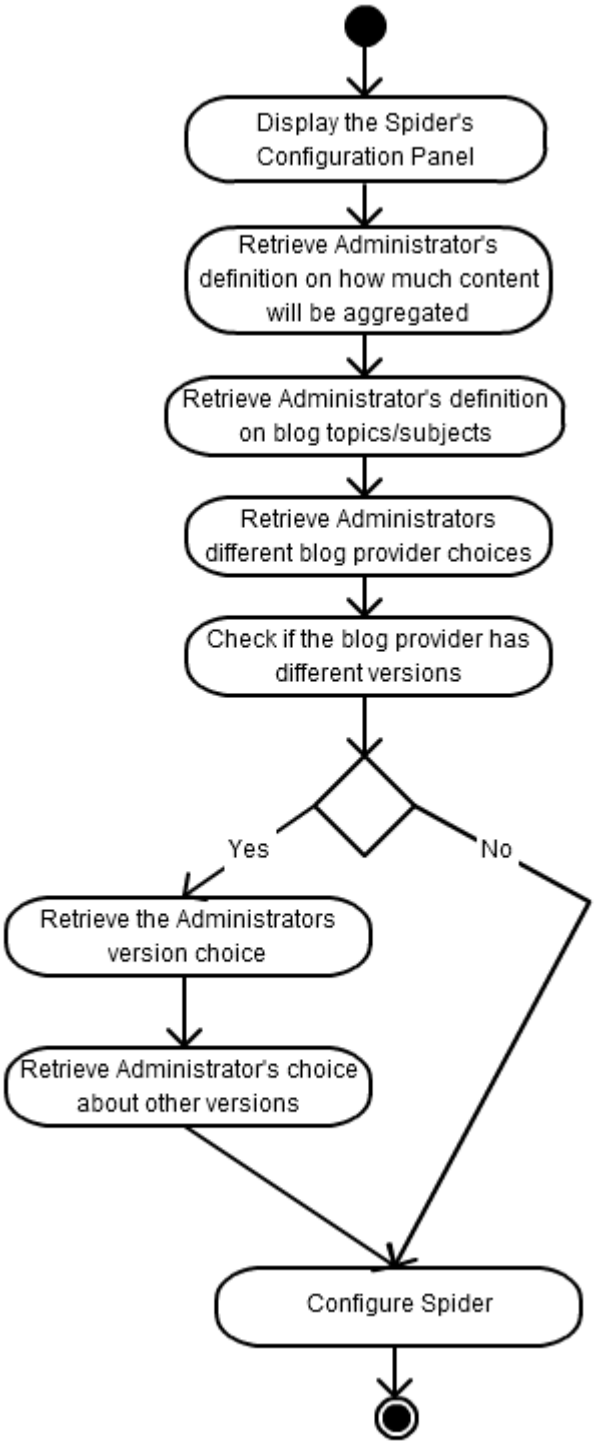


Figure 25: Activity Diagram for configuring Spider to filter aggregated content

FR16 – Search by author

Requirement category	Functional Requirement, User Interface
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	It has to be possible to search for content of a specific person/author. A query for an author should return every blog, blog post, and comment that the author has created. When displaying an author profile, a list of the latest posts by the same author should be displayed as well.
Stakeholder	Content Retriever Library, Businesses, Researcher, Content Retriever
Justification / Foundation	Content Retriever Library: You should receive as well the blog if you search for a specific author in our system. Businesses: "our customers want to segment blogs – e.g. important persons politicians to be separated from teenagers" Researcher: I use http://www.mendeley.com/ and I like it very much! [...] Metadata are populated automatically, I can search / browse etc. Interview #2 ~24:00: "It would be nice to see posts just from one author..." Interview #2 ~29:30: "If you are writing about a specific subject and you know that a different blogger in your group is writing about the same subject from a different aspect, it'd be nice to see them all [all the previous relative posts], it'd be good t o find them easily..."
Assessment / Measures	A search for authors/people returns the blogs, blog posts, and comments that the author has created. A list of latest posts by the same author is showed when displaying a blog post.
Author(s) of the requirement description	• Hendrik Kalb • Vangelis Banos • Jaime García • Nikolaos Kasioumis

Use Case UCFR 16.1: Search by Author

Use Case UCFR16.1		Search by Author
Description		The Content Retriever can search blog content by author in the Platform.
Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Content Retriever is logged in
	Postconditions	1. The Platform retrieves the search result and displays to the Content Retriever
Steps		1. The Content Retriever selects "author" from the available fields 2. The Content Retriever enters the search keywords 3. The Platform executes the query retrieved from the Content Retriever 4. The Platform retrieves matching and similar records 5. The Platform displays the search results
Parent		UCFR13
Trigger		The Content Retriever enters the keywords and presses a button
Variations		
Issues		

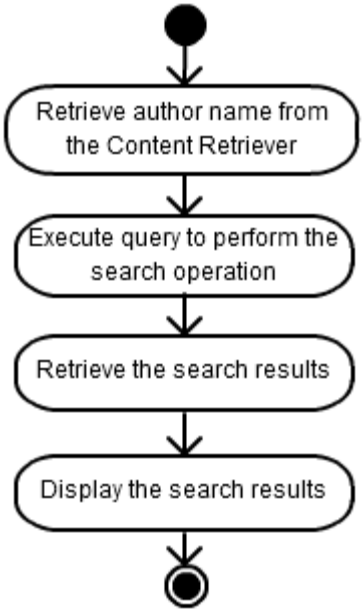


Figure 26: Activity Diagram for searching by author

FR17 – Print/Export as PDF, JPEG, etc.

Requirement category	Functional Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	It should be possible to print or to export blogs and blog posts as PDF, JPEG and other formats.
Stakeholder	Researcher, Blog Author
Justification / Foundation	Researcher: ": Actually through printing the pages as a PDF or photographing them as JPEG. Because I use qualitative analyzing-software Atlas.ti and for this I need document to uploaded into the program so then I would either printed as a PDF or save it as a JPEG" Researcher: "Yes, to begin with every blog - every blog looks differently so whenever you print it to a PDF it will look differently - and also increasing the blogs have widgets, side widgets and sidebars so this can make it very difficult to print - you know it moves the formatting around, so you have two pages of widgets and then finally the text comes - yes that was quite inconvenient. Sometimes printed as a JPEG even though then I lost the possibility to work with full-text but sometimes I was simply unprintable as a PDF." Blog Author: It would be good an easy tool that would allow you to download your blog as a PDF or txt file or anything else format, that will be printable.
Assessment / Measures	Printing or exporting of blogs and blog posts is possible.
Author(s) of the requirement description	• Hendrik Kalb • Stella Kopidaki

Note: Use Case UCFR 4.1"Export Content" can apply here

FR18 – Analyze the network structure of blogs

Requirement category	Functional Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	The archive should analyse the connections between the blogs to provide estimations about importance, influence or community structures.
Stakeholder	Researcher, Blog Author
Justification / Foundation	Researcher: "Blogs are connected into networks through the blog-rolls so if there was a tool which could also compile the connected blogs that would help identifying relevant blogs because connected blogs usually blog about the same subject or in case of science they are professionally connected, so that would have been very helpful." Blog Author: "You are thinking of tracking the relationships between blogs, I think that is a very useful thing to do." Blog Author: "X uses an image, Y uses the same image, that creates a link between X and Y even though may have never connected with each other. They may use the same images at the same URL, which is easy to detect, they may use the same image where the copies of the images are located at different URLs, even though it is the same image. One may use the cropped version of the same image of another."
Assessment / Measures	Network measurements for importance, influence and community structure can be calculated.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCFR 18.1: Search by Ranking

Use Case UCFR18.1		Search by Ranking
Description		The Content Retriever can order search results by page ranks
Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Content Retriever views the Advanced Search Interface
	Postconditions	1. The Platform retrieves the search result and displays to the Content Retriever
Steps		1. The Content Retriever enters the search information through the Search Interface 2. The Content Retriever selects the "Retrieve Pages by Rankings" option 3. The Platform calculates the page rankings of the blog pages executes the query retrieved from the Content Retriever 4. The Platform retrieves the search results which are sorted by their page ranks 5. The Platform displays the search results
Parent		UCFR13.1
Trigger		The Content Retriever enters the keywords and presses a button
Variations		
Issues		

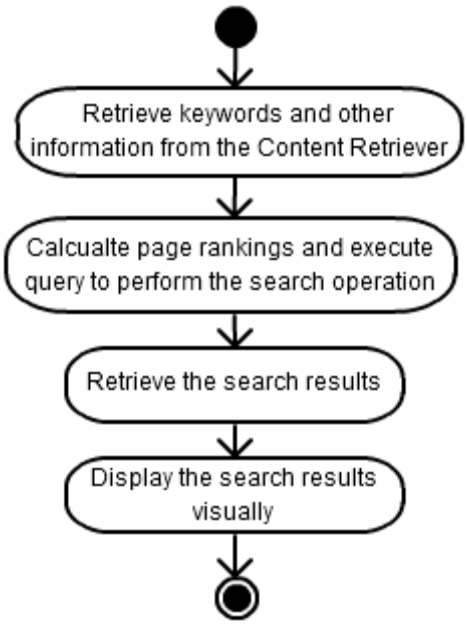


Figure 27: Activity Diagram for page ranking

FR19 – Sharing and collaboration

Requirement category	Functional Requirement
Degree of necessity	( ) **Essential** ( ) **Recommended** (X) **Optional**
Description of the requirement	Users would possibly like to share their annotations, favourite list or similarities. Additionally, they would possibly like to work together on the archived content (e.g. common annotations).
Stakeholder	Researcher, Blog Author
Justification / Foundation	Researcher: "Yes, I think I would be interesting for research because I actually do a lot of research with other scientists so I would be necessary to be able to share (share projects, share codes, searches) but it would be important to be able to decide what I want to share and what I don't want to share - and for private purposes or for informational purposes professionally I read blogs related to say research methods which of course I like to share with my colleagues. So I would also need a space which I can share with others, but which would be still separate from my private space." Researcher: I use http://www.mendeley.com/ and I like it very much! I can share, download, search PDF documents. I can sync multiple machines with the server and also share with others. Metadata are populated automatically, I can search / browse etc. I can highlight a part of a document and save this as a note. I can share notes with other people in groups. Blog Author: Bloggers to have relationships between them like friend, known or something like that.
Assessment / Measures	Users can share their data with other users and can work together with other users in the archive system.
Author(s) of the requirement description	• Hendrik Kalb • Vangelis Banos • Stella Kopidaki

Use Case UCFR 19.1: Share Favorites List

Use Case UCFR19.1		Share Favourites List
Description		Users of the platform can share their favorites list with other users
Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Content Retriever has displayed his/her favorites list
	Postconditions	1. The Content Retriever has shared his/her favorites list
Steps		1. The Content Retriever clicks on the "Share" button/link 2. The Platform makes the Content Retriever's favorites list public so that all other user can see it.
Parent		
Trigger		The Content Retriever clicks a button/link
Variations		
Issues		

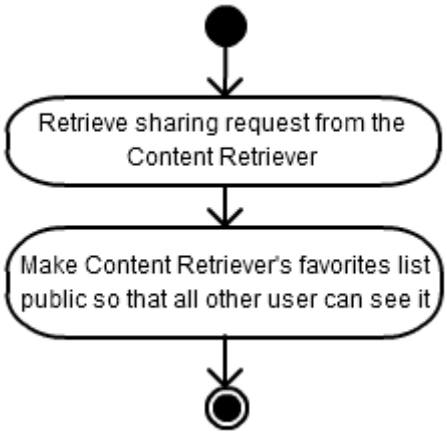


Figure 28: Activity Diagram for sharing favorites list

FR20 – List of favorite blogs and topics

Requirement category	Functional Requirements, User Interface Requirements
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	It should be possible that a user can define a list of blogs or topics he/she wants to observe.
Stakeholder	Business, Blog Reader, Researcher, Content Retrievers - Libraries
Justification / Foundation	Business: "Thematic streams of information monitored should be user defined and be capable to vary from user to user and from type of company to type of company. Of course some default scenarios may be offered but they should be subject to customisation." Researcher: If I could also use a tool to "tag" a set of blogs that interest me and be able to monitor them, I would be very happy to use such an app. Library: It would be good for each user to monitor his/her own set of blogs and also add a new blog to the system.
Assessment / Measures	Customization of user specific lists of favourite blogs and topics is possible. A user interface component enables users to tag blogs as favourite and be able to monitor their contents.
Author(s) of the requirement description	• Hendrik Kalb • Vangelis Banos

Use Case UCFR 20.1: Fill Favorites List

Use Case UCFR 20.1		Fill Favorites List
Description		The users of the platform can insert searching criteria to automatically fill their favorites list
Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Content Retriever views the Advanced Search Interface
	Postconditions	1. The Platform has added all the search results to his/her favorites list
Steps		1. The Content Retriever enters the search keywords and other information 2. The Content Retriever marks "Add to Favorites" option 3. The Platform executes the query retrieved from the Content Retriever 4. The Platform retrieves the matching and similar records 5. The Platform fills the Content Retriever's favorites list with the search results 6. The Platform displays the search results
Parent		UCFR13.1
Trigger		The Content Retriever clicks on a button
Variations		
Issues		

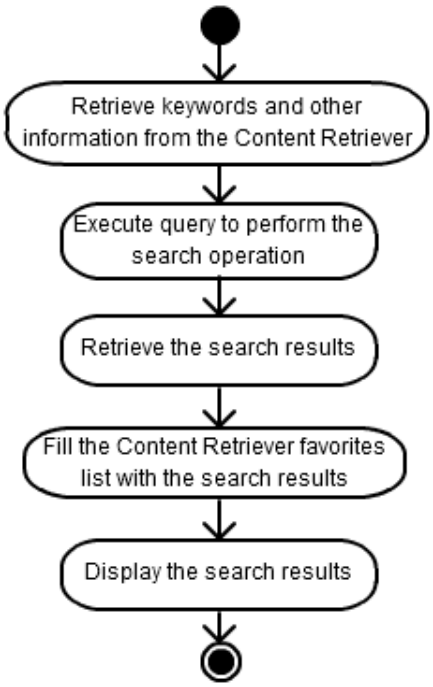


Figure 29: Activity Diagram for filling user's favorites list by inserting a searching criteria

Use Case UCFR 20.2: Add to Favorites

Use Case UCFR 20.2		Add to Favorites
Description		The users of the platform can add blogs or blog posts to their personal favorites list so that they can reach them later easily
Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Content Retriever has viewed a blog or a blog post
	Postconditions	1. The Content Retriever has viewed a blog or a blog post
Steps		1. The Content Retriever clicks on “Add to Favorites” button/link 2. The Platform adds blog or blog post to Content Retriever’s favorites list
Parent		
Trigger		The Content Retriever clicks on a button/link
Variations		
Issues		

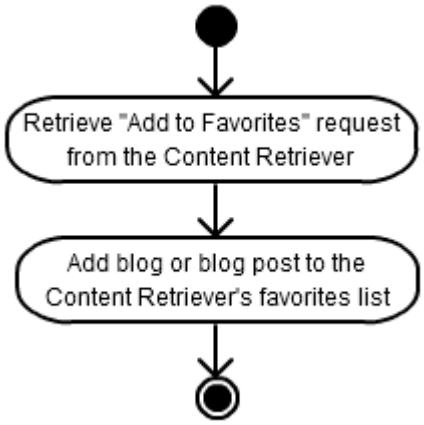


Figure 30: Activity Diagram for adding a blog/blog post to the user’s favorites list

FR21 – Sentiments analysis on blog post level

Requirement category	Functional Requirement
Degree of necessity	( ) **Essential** ( ) **Recommended** (X) **Optional**
Description of the requirement	The sentiments analysis of blog posts indicates if the blog has a positive or negative attitude regarding something the post is about.
Stakeholder	Businesses, Content Retriever
Justification / Foundation	Businesses: "also information about weaknesses and spots for criticism for our products and the products and services of the competition" Businesses: "be able to detect the mood of their installed base in terms of people's attitude over their products, changes in the demographics, etc." Businesses: "Build a repository of people's attitudes against products services or other immaterial parts like a company culture, image etc. based solely on information coming from conversations, blog postings etc." Interview #1 ~11:15: "It's kind of fun to play with, you can know so much..."
Assessment / Measures	Sentiments of posts can be analysed. The sentiment analysis score for a blog post is displayed to the end user.
Author(s) of the requirement description	• Hendrik Kalb • Jaime García • Nikolaos Kasioumis

Use Case UCFR 21.1: Analyze Sentiments

Use Case UCFR 21.1		Analyze Sentiments
Description		The Platform can analyze sentiments and calculate a score for a blog post which is displayed to the end user.
Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Content Retriever has provided a sentiment on a blog post
	Postconditions	1. The Content Retriever has calculated a score for the blog post
Steps		1. The Platform analysis the sentiment and determines score for it 2. The Platform calculates the overall score for the blog post 3. The Platform updates the score for the blog post
Parent		
Trigger		The Content Retriever has provided a comment
Variations		
Issues		

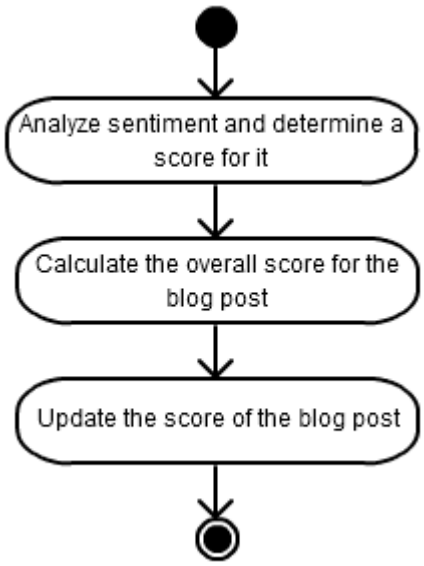


Figure 31: Activity Diagram for sentiment analysis of a blog post

FR22 – Summaries/Journals about new archive content

Requirement category	Functional Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	A user can subscribe a newsletter or electronic journal that provides a summarization of interesting new blog posts on a regular basis (e.g. once a week).
Stakeholder	Businesses, Blog Reader/Author
Justification / Foundation	Businesses: "we provide automatic e-mails that are either summaries or either reports from the dashboard" Blog Reader/Author: In order to be notified for updates I would prefer the classic way of email or a Facebook like mechanism, that gathers the new elements and when you log in the service and you are going to use it, it shows you what have happened in the past, what interests you, but there are also 10 new stories and if you want to see and read them click here, so I can choose to see the recent content if I have time or to read it later. Blog Author: "So, to make any sort of system like this useful to users, it is going to have to provide that prompt that will send users to whatever information is that you want to receive on a regular basis." Blog Author: "We have 260 separate blogs that people write. And it is essential for us to have a central newsletter that we send out to every participant every weekday, so when somebody has posted a new post at one of those 260 blogs it shows up in the newsletter. Because it is impossible for somebody to go to those 260 blogs and even they are not going to come to our course website to see what's there."
Assessment / Measures	A user can subscribe a summary document that will be sent on a regular basis (e.g. once a week) and informs about new posts that could be interesting to the user.
Author(s) of the requirement description	• Hendrik Kalb • Stella Kopidaki

Note: Use cases UCFR 12.1 and 12.2 can apply here

FR23 – Detection of duplicates

Requirement category	Functional Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	Duplicates of blog posts should be detected.
Stakeholder	Businesses, Blog Author
Justification / Foundation	Businesses: (Can you identify any problems or issues with the procedures you are currently following?) "[...] finding duplicates [...]" Blog Author: "I think you will have an issue with duplicates, but, I mean, I do not want to be the one editing duplicates."
Assessment / Measures	Duplicates can be detected automatically.
Author(s) of the requirement description	• Hendrik Kalb

Note: Use case UCFR 6.1 can apply here

FR24 – User specific collections/projects

Requirement category	Functional Requirements
Degree of necessity	( ) **Essential** ( ) **Recommended** (X) **Optional**
Description of the requirement	A person who uses the archive for several different reasons, e.g. private vs. professional or in different projects, would benefit from a possibility to manage preferences (e.g. lists of favourite blogs) and data (e.g. annotations) in different projects or areas.
Stakeholder	Researcher
Justification / Foundation	Researcher: "For scientific purposes it would be important that the data can be sorted into different projects"
Assessment / Measures	A user can manage different projects or areas under one user account.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCFR 24.1: Create Project

Use Case UCFR 24.1		Create Project
Description		Users of the platform can manage different projects where they can categorize blogs for different purposes under one user account.
Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Content Retriever is logged in
	Postconditions	1. The Content Retriever has created a project
Steps		1. The Content Retriever clicks on “Create Project” button/link 2. The Content Retriever enters project information (name, description, etc.) 3. The Platform creates a project for the Content Provider
Parent		
Trigger		The Content Retriever clicks on a button/link
Variations		
Issues		

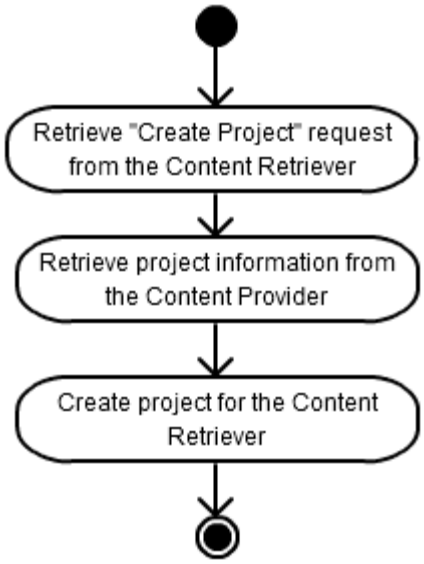


Figure 32: Activity Diagram for creating a project

Use Case UCFR 24.2: Add to Project

Use Case UCFR 24.1		Add to Project
Description		Users of the platform can add blogs to their projects
Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Content Retriever has displayed a blog
	Postconditions	1. The Content Retriever has created a project
Steps		1. The Content Retriever clicks on “Add to Project” button/link 2. The Platform retrieves and displays the Content Retriever’s project list 3. The Content Retriever selects a project from the list 4. The Platform adds the blog content to the selected project
Parent		
Trigger		The Content Retriever clicks on a button/link
Variations		
Issues		

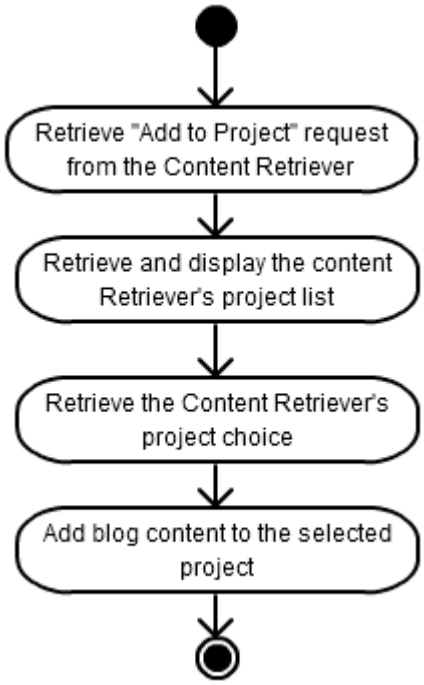


Figure 33: Activity Diagram for adding blog content to a project

Use Case UCFR 24.3: Delete Project

Use Case UCFR 24.3		Delete Project
Description		Users of the platform can delete their project if they do not want to use them anymore
Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Content Retriever has displayed a project
	Postconditions	1. The Content Retriever has deleted his/her project
Steps		1. The Content Retriever clicks on “Delete Project” button/link 2. The Platform deletes the project
Parent		
Trigger		The Content Retriever clicks on a button/link
Variations		
Issues		

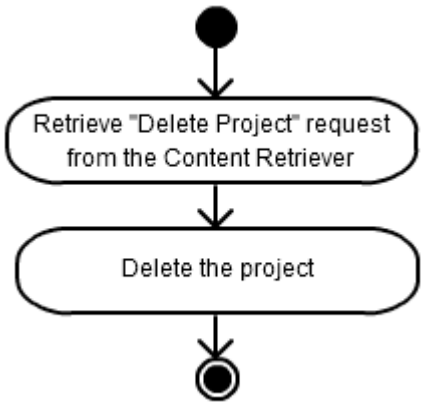


Figure 34: Activity Diagram for deleting a project

Use Case UCFR 24.4: Delete from Project

Use Case UCFR 24.4		Delete from Project
Description		Users of the platform can delete blogs from their projects
Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Content Retriever has displayed a project
	Postconditions	1. The Content Retriever has deleted his/her project
Steps		1. The Content Retriever selects blogs from the project 2. The Content Retriever clicks on a “Delete” button 3. The Platform removes the selected blogs from the project
Parent		
Trigger		The Content Retriever clicks on a button/link
Variations		
Issues		

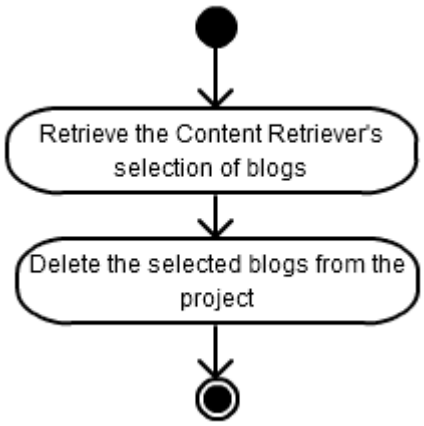


Figure 35: Activity Diagram for deleting a blog from a project

FR25 – Paid access/Billing system

Requirement category	Functional Requirements
Degree of necessity	( ) **Essential** ( ) **Recommended** (X) **Optional**
Description of the requirement	Businesses models can be based on the different access (e.g. basic and premium) to the archived content or functionalities of the archive. Therefore, a functionality to restrict access to paying users and a billing system inside the archive would be beneficial for such business models.
Stakeholder	Businesses
Justification / Foundation	Businesses: "We would be happy to work with some other members of a value chain for the joint offer of a paid service" Businesses: "Access should be given for free - or at least to the customers that are having a contract."
Assessment / Measures	Functionalities are available that can restrict the access on content and/or services to paying users.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCFR 25.1: Upgrade Account

Use Case UCFR25.1		Upgrade Account
Description		The users of the BlogForever platform can upgrade their accounts to get additional services.
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever views the "Pricing" interface
	Postconditions	1. The Content Retriever's account has been upgraded and Content Retriever can use additional services.
Steps		1. The Content Retriever enters necessary information (paying method such as credit card or PayPal and etc.…) to upgrade his/her account 2. The Platform retrieves the information and processes it. 3. If the retrieved information is correct (valid credit number / PayPal account etc.…) the Platform sets an upgraded account policy for the Content Retriever 4. Otherwise, The Platform displays an error message
Parent		
Trigger		The Content Retriever enters necessary information and clicks a button
Variations		
Issues		

Note: Same use case used in the requirements gathered from the DOW (UC12.1)

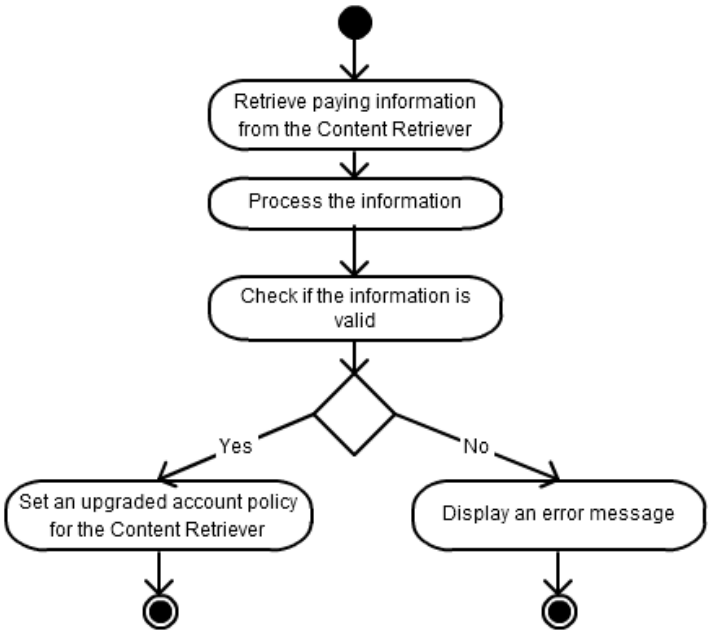


Figure 36: Activity Diagram for upgrading user accounts

FR26 – Context-sensitive search by keyword

Requirement category	Functional Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	The end-user is able to search using context-specific keywords, i.e. keywords are interpreted under a specific topic or context, or field.
Stakeholder	Content retrievers
Justification / Foundation	Interview#2 ~10:45 "We were doing a search on LHC (CERN) on social media, but LHC also stands for the name of a hair loss product, and disappointed customers were complaining against this product that doesn't work... You need to have a search engine that is quite smart..."
Assessment / Measures	Keywords contain semantic information and search results can be categorized based on that information.
Author(s) of the requirement description	• Jaime García • Nikolaos Kasioumis

Use Case UCFR 26.1: Context-Sensitive Search

Use Case UCFR 26.1		Context-Sensitive Search
Description		The Content Retriever can search blog content in the Platform.
Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Content Retriever has displayed Advance Search panel
	Postconditions	1. The Platform has displayed search results
Steps		1. The Content Retriever enters the search keywords 2. The Content Retriever selects a field from the available fields 3. The Content Retriever selects a topic from the available "topics" list 4. The Platform performs a search among the blogs having the selected "topic" 5. The Platform retrieves the matching records 6. The Platform displays the search results
Parent		UCFR13.1
Trigger		The Content Retriever enters the keywords and presses a button
Variations		
Issues		

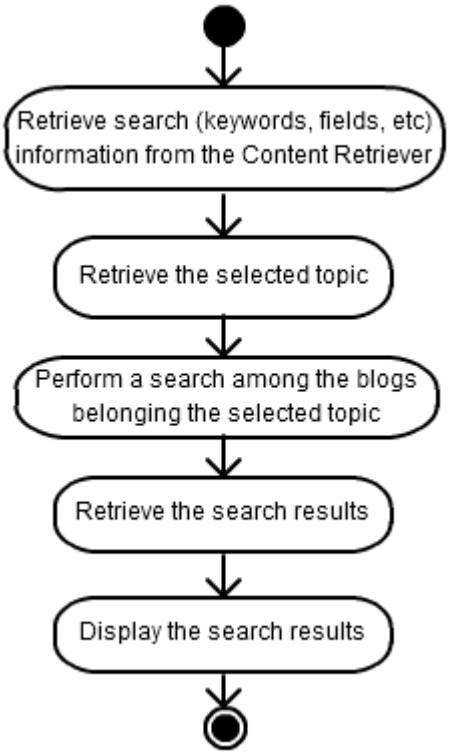


Figure 37: Activity Diagram for context-sensitive search

FR27 – Ranking of blogs and blog posts

Requirement category	Functional Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	The ranking criteria for blogs may include popularity, comments sentiment analysis, geographic al location, frequency of update and language. Ranking of blogs should be based on frequency of new posts and validity of content. Also ranking in a category should be done according to the number of links, posts, comments they have, and the user to know which these criteria are. More ranking criteria: how much other speak about it, how frequently it is updated, amount of content, how frequently the Facebook page is updated and with what content, if there are answers in the comments from the owners of the blog, if there is twitter activity.
Stakeholder	Content retrievers, Blog Author/Reader
Justification / Foundation	Interview#2 ~15:05 "The sentiment, that to me is very important...it's respected, it's interesting" Interview#1 ~10:05 "I think frequency of updates is pretty good. Probably the language in which people blog. Maybe location." Blog Author/Reader: The criteria that might be the same for all blogs, independent of their type, are that they should be updated frequently and the user to check if the published content is valid. In case that the information is invalid the blogger loses his reliability. Ranking have meaning among the blogs of a specific topic, e.g. if top-5 blogs includes 5 technology blogs and 5 others but if I am interested to flowers ranking has no sense. If I get into category floriculture and see a ranked list of blogs based on a specific ranking, these can be the number of links, posts or comments they have, this can make sense if I know which are these criteria, in order to know how relevant this is to me. I can think various criteria, e.g. how much other speak about it, how frequently it is updated, what amount of content it has, how many comments it has, how frequently the Facebook page is updated and with what content, if there are answers in the comments from the owners of the blog, if there is twitter activity, a combination of

	things that shows that someone runs and cares about this blog and not something in the internet that generates content. Blog Reader: I want to see a list with the top-5 or top-N blogs and to have information about the category they belong to.
Assessment / Measures	The platform provides different ranking algorithms based on the mentioned criteria.
Author(s) of the requirement description	• Jaime García • Nikolaos Kasioumis • Stella Kopidaki

Note: Use Case UCFR 11.2, 11.3 and 11.4 can apply here

Use Case UCFR 27.1: Most Updated Posts

Use Case UCFR27.1		Most Updated Posts
Description		Users can view most updated blog posts.
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever is logged in.
	Postconditions	1. The Content Retriever has viewed most updated blog posts
Steps		1. The Content Retriever selects “Most Updated” option from the ranking methods list through the Search Interface 2. The Content Retriever clicks on “Search” button 3. The Platform retrieves most updated blog posts 4. The Platform displays the most updated blog posts
Parent		
Trigger		The Content Retriever clicks on a button
Variations		
Issues		

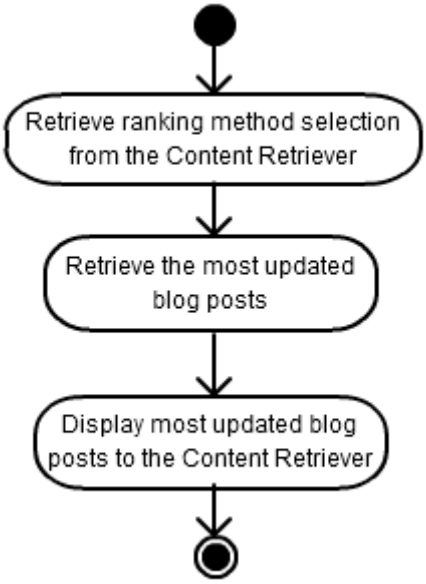


Figure 38: Activity Diagram for getting most updated blog posts

Use Case UCFR 27.2: Most Shared Posts

Use Case UCFR27.2		Most Shared Posts
Description		Users can view blog posts that are most shared on Facebook
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever is logged in.
	Postconditions	1. The Content Retriever has viewed most shared blog posts
Steps		1. The Content Retriever selects “Most Shared on Facebook” option from the ranking methods list through the Search Interface 2. The Content Retriever clicks on “Search” button 3. The Platform retrieves most shared blog posts 4. The Platform displays the most shared blog posts
Parent		
Trigger		The Content Retriever clicks on a button
Variations		
Issues		

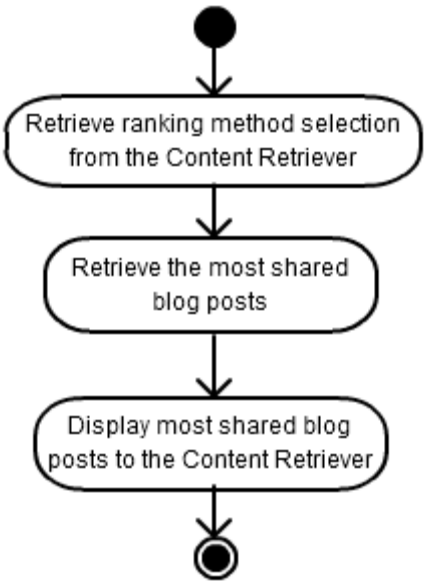


Figure 39: Activity Diagram for getting most shared blog posts on Facebook

Use Case UCFR 27.3: Most Tweeted Posts

Use Case UCFR27.3		Most Tweeted Posts
Description		Users can view blog posts that are most tweeted on Twitter
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever is logged in.
	Postconditions	1. The Content Retriever has viewed most shared blog posts
Steps		1. The Content Retriever selects “Most Tweeted” option from the ranking methods list through the Search Interface 2. The Content Retriever clicks on “Search” button 3. The Platform retrieves most tweeted blog posts 4. The Platform displays the most tweeted blog posts
Parent		
Trigger		The Content Retriever clicks on a button
Variations		
Issues		

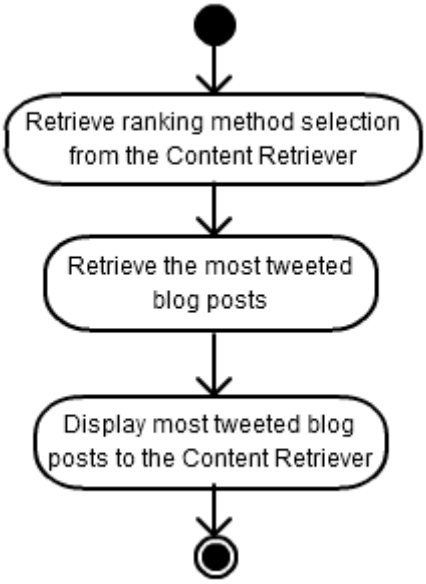


Figure 40: Activity Diagram for getting most shared blog posts on Twitter

FR28 – Recommend a cluster of blogs according to user preferences

Requirement category	Functional Requirement
Degree of necessity	( ) **Essential** (x) **Recommended** ( ) **Optional**
Description of the requirement	Having a list of blogs that the user is interested in, clusters to be constructed based on the topic in order to suggest to the user blogs that he might is interested in.
Stakeholder	Content retrievers - Blog Readers.
Justification / Foundation	Phrases from the interview • I would like to start with a list of blogs that the system has recognized that I am interested in, and I have initially denote that I am interested in, e.g. I have a list with 2 technological blogs, 2 blogs of photography, 2 music blogs, 1 blog x, over it I want to be able to build and to add or delete items instead of searching what I like. Since there are tools that someone can understand the fields I like, looking at my Facebook likes, twitter tweets, what I read in Google reader, so with this data I want the service to automatically create a cluster, to suggest it to me and ask my participation by asking if I am happy with this cluster or if I want to add or delete some items and with this way to build the cluster. The cluster can get smaller or bigger with suggestions and related items.
Assessment / Measures	A cluster of blogs is recommended based on the preferences of the user.
Author(s) of the requirement description	• Stella Kopidaki

Use Case UCFR 28.1: Suggest Blogs

Use Case UCFR28.1		Suggest Blogs
Description		A cluster of blogs is recommended based on the preferences of the user.
Actors		Platform
Assumptions	**Preconditions**	1. The Content Retriever is logged in
	Postconditions	1. The Platform has suggested a cluster of blogs to the Content Retriever
Steps		1. The Platform analyses a list of blogs that the Content Retriever had interested in 2. The Platform retrieves a list/cluster of blogs that fits to the Content Retrievers interest 3. The Platform displays the suggested list/cluster of blogs
Parent		
Trigger		The Content Retriever logs in to the BlogForever Platform
Variations		
Issues		

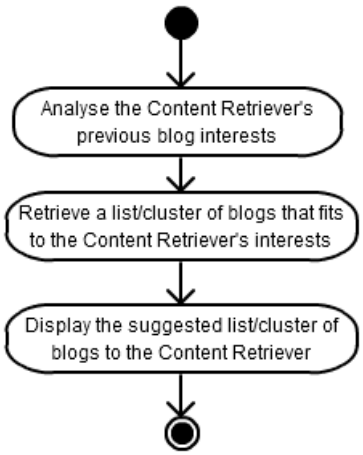


Figure 41: Activity Diagram for suggesting blog content to the Content Retriever

FR29 – Detect and remove spam

Requirement category	Functional Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	Spam should be detected and excluded from further analysis or search queries.
Stakeholder	Businesses
Justification / Foundation	Businesses: (Can you identify any problems or issues with the procedures you are currently following?) "[...] removing spam and sort of extracting the right content [...]"
Assessment / Measures	Spam is detected and excluded from analysis and search results.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCFR 29.1: Add Spam Detection Service

Use Case UCFR29.1		Add Spam Detection Service
Description		The Administrator can add online spam detection services for Spider
Actors		Spider, Administrator
Assumptions	**Preconditions**	1. The Administrator is logged in
	Postconditions	1. The Spider has validated content of the blog page
Steps		1. The Administrator views the Spider's Configuration Panel 2. The Administrator selects the online spam detection services to be used. 3. The Spider configures itself to use selected services
Parent		
Trigger		The Administrator selects online spam detection services
Variations		
Issues		

Note: Same use case used in the requirements gathered from the DOW (UC17.1)

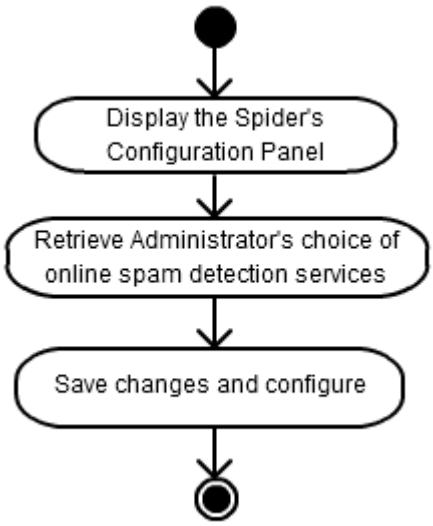


Figure 42: Activity Diagram for adding spam detection services

Use Case UCFR 29.2: Remove Spam

Use Case UCFR29.2		Remove Spam
Description		The Spider detects and removes spam using online spam detection web services
Actors		Spider
Assumptions	**Preconditions**	1. The Spider has established a connection with the blog page
	Postconditions	1. The Spider has detected and removed spam in the blog content
Steps		1. The Spider retrieves the list of the selected online spam detection services 2. The Spider detects and removes spam in the blog content by using the spam detection services
Parent		
Trigger		The Spider has retrieved blog content
Variations		
Issues		

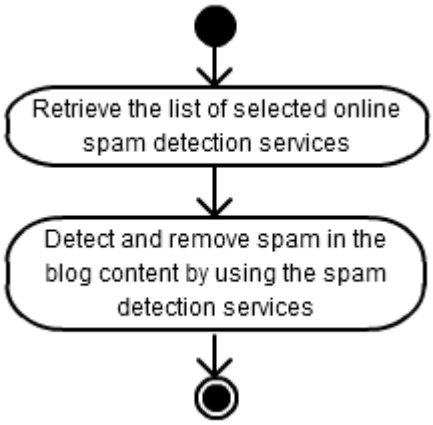


Figure 43: Activity Diagram for removing spam

FR30 – Extract bibliographic metadata from blog contents

Requirement category	Functional and User Interface Requirements
Degree of necessity	( ) **Essential** (**X**) **Recommended** ( ) **Optional**
Description of the requirement	The BlogForever Platform will be able to extract bibliographic metadata from various types of content embedded in blogs (e.g. PDF documents and latex files). Indexes will be populated with these metadata and searching/browsing will be available to users via special web interfaces.
Stakeholder	Admins - Organizations
Justification / Foundation	Interviewee says: We have extracted bibliographic metadata from PhD theses and scientific publications. […] The challenge we faced had to do with the many different types of content from images (in various formats) to PDF & latex documents.
Assessment / Measures	The user is able to view bibliographic metadata for archive content. Browsing and searching based on this type of metadata is also supported using appropriate web interface controls.
Author(s) of the requirement description	Vangelis Banos

Use Case UCFR 30.1: Extract Metadata

Use Case UCFR30.1		Extract Metadata
Description		The Platform is able to extract bibliographic metadata from various types of content embedded in blogs
Actors		Platform
Assumptions	**Preconditions**	1. The Platform has retrieved a blog content with embedded content such as PDF, latex files
	Postconditions	1. The Platform has extracted bibliographic metadata from the embedded content
Steps		1. The Platform parses the embedded content 2. The Platform extracts bibliographic metadata from the parsed content 3. The Platform associates the extracted metadata with the archived blog content
Parent		
Trigger		The Platform has retrieved a blog content with embedded content
Variations		
Issues		

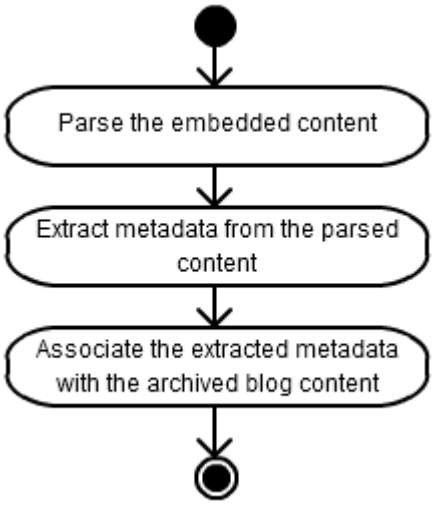


Figure 44: Activity Diagram for extracting metadata from embedded files such pdf, latex, etc.

FR31 – Define important blogs and filter junk

Requirement category	Functional Requirements
Degree of necessity	( ) **Essential** ( ) **Recommended** (X) **Optional**
Description of the requirement	The BlogForever platform should be able to suggest a list of "important" blogs and filter "junk" blogs
Stakeholder	Content Retrievers - Libraries
Justification / Foundation	Interviewee says: He haven't thought of archiving blogs and providing them to students because they are very "casual", people write whatever they want with no control. He would like to have some "official" blogs from known entities. Example: the blog of NASA. It is difficult to filter junk blog from important ones.
Assessment / Measures	The BlogForever platform admin interface should suggest "important" blogs which should be added to the archive and flag "junk" blogs which should be removed.
Author(s) of the requirement description	Vangelis Banos

Use Case UCFR 31.1: Filter & Flag Junk Blogs

Use Case UCFR31.1		Flag Junk Blogs
Description		The Administrator can define a blog as “junk“
Actors		Administrator, Platform
Assumptions	**Preconditions**	1. The Administrator has displayed a blog
	Postconditions	1. The Platform has flagged a blog as “junk”
Steps		1. The Administrator selects “junk” from the available flag list 2. The Platform flags the blog as “junk” 3. The Platform removes all the blogs flagged as junk on a prescheduled time
Parent		
Trigger		The Platform has retrieved Administrator’s flag choice
Variations		
Issues		

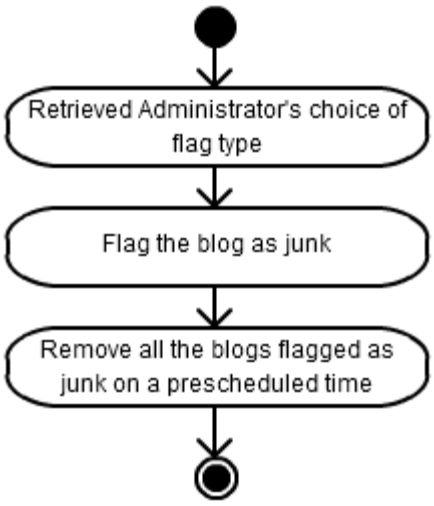


Figure 45: Activity Diagram for filtering and junk blogs as junk

Use Case UCFR 31.2: Flag as Important

Use Case UCFR31.2		Flag as Important
Description		The Administrator can define a blog as “important“
Actors		Administrator, Platform
Assumptions	**Preconditions**	1. The Administrator has displayed a blog
	Postconditions	1. The Platform has flagged a blog as “important”
Steps		1. The Administrator selects “important” from the available flag list 2. The Platform flags the blog as “important”
Parent		
Trigger		The Platform has retrieved Administrator’s flag choice
Variations		
Issues		

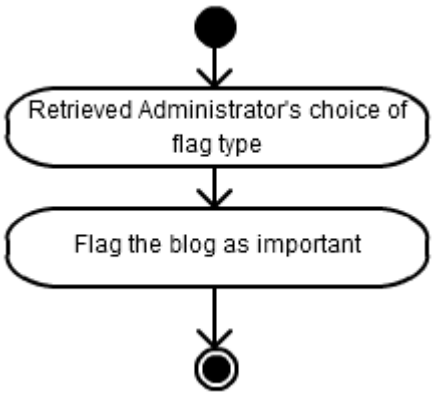


Figure 46: Activity Diagram for flagging a blog as “important“

Use Case UCFR 31.3: Suggest Important Blogs

Use Case UCFR31.3		Suggest Important Blogs
Description		The Administrator can define a blog as “important“
Actors		Platform
Assumptions	**Preconditions**	1. The Content Retriever is logged in
	Postconditions	1. The Platform has suggested a list of blogs flagged as “important”
Steps		1. The Platform retrieved blogs flagged as “important” 2. The Platform displays the list of blogs flagged as “important”
Parent		
Trigger		The Content Retriever logs in to the BlogForever Platform
Variations		
Issues		

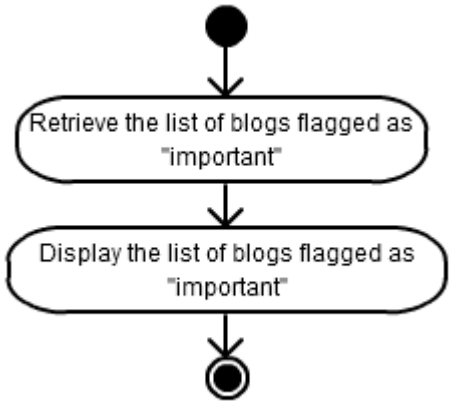


Figure 47: Activity Diagram for suggesting blogs flagged as “important“

FR32 – Add user suggested blogs to the archive

Requirement category	Functional Requirements
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	The BlogForever platform should enable users to suggest a new blog to be added in the archive for aggregation, preservation, management & dissemination. The suggestion should be done by adding an RSS URL.
Stakeholder	Content Retrievers – Libraries, Blog Author
Justification / Foundation	Interviewer: It would be good for each user to monitor his/her own set of blogs and also add a new blog to the system. It would be good to submit an RSS of a blog and the system would start monitoring it. Blog Author: "You don't want just 13 hundred random blogs. You don't want: "anyone can submit a blog", because you get ten useful blogs and 25 spam blogs, which would be a big problem."
Assessment / Measures	The user can add a blog RSS URL to the suggested blogs list. The BlogForever platform will start aggregating this blog according to the system policies. (Probably, there should be a confirmation from the administrator)
Author(s) of the requirement description	• Vangelis Banos • Hendrik Kalb

Use Case UCFR 32.1: Aggregate Content

Use Case UCFR32.1		Aggregate Content
Description		The Spider aggregates content from the blog pages
Actors		Spider
Assumptions	**Preconditions**	1. The Content Retriever has provided URL of the blog page
	Postconditions	1. The content of the blog page has been aggregated by the Spider
Steps		1. The Spider establishes a connection with the blog page 2. The Spider checks for intellectual property rights if the Content Provider restricted access to the blog content or not 3. If it is allowed, the Spider retrieves the content of the blog and distinguishes specific elements 4. UCFR29.2 5. The Spider creates checksum for the aggregate content 6. The Spider sends all the content (weblog posts, user comments, metadata, etc.) to the Platform 7. Otherwise, display an error message about the failure
Parent		
Trigger		The Content Retriever's aggregation request or the Administrator's approval
Variations		
Issues		

Note: Same use case used in the requirements gathered from the DOW (UC16.1)

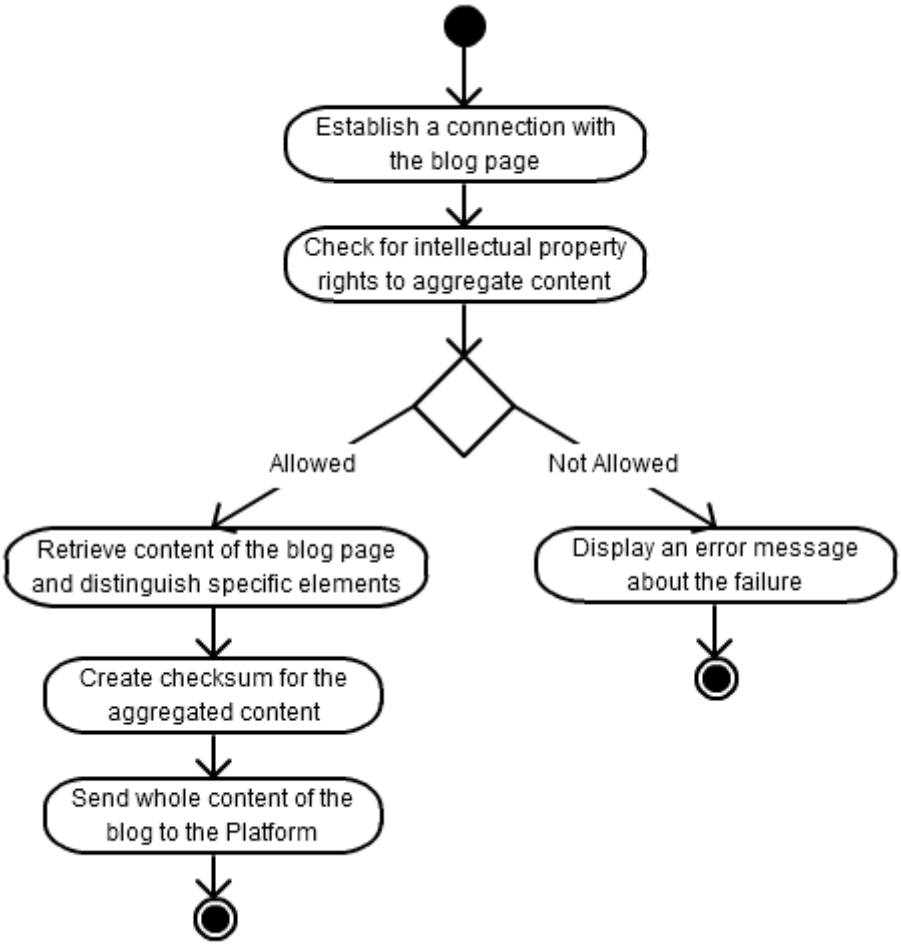


Figure 48: Activity Diagram for content aggregation

FR33 – Dissemination of newly archived items in external social platforms (ex. Twitter) in connection with author profiles

Requirement category	Functional Requirement
Degree of necessity	( ) **Essential** ( ) **Recommended** (X) **Optional**
Description of the requirement	The idea is that the BlogForever platform has an account in a social network (e.g. Twitter). If the author wants this service, he/she inserts his/her account (or the account related to their blog). When a new post is archived, the platform account will post a message (e.g. a tweet) mentioning the author's account.
Stakeholder	Blog author
Justification / Foundation	Interview#2~4:00: "We try to broaden our audience, CERN is followed by 300.000 people on twitter, so whenever we put a new blog there, we advertise it on this twitter thing, from that I get a few thousand people to look at it..."
Assessment / Measures	The platform disseminates newly archived items in social networks and mentions thereby the author's account.
Author(s) of the requirement description	• Jaime García • Nikolaos Kasioumis

Use Case UCFR 33.1: Add Social Network Account

Use Case UCFR 33.1		Add Social Network Account
Description		Blog authors can insert their social network accounts to the Platform so that when a new content is archived, the Platform's account can post a message by mentioning the authors account
Actors		Content Provider, Platform
Assumptions	**Preconditions**	1. The Content Provider is logged in
	Postconditions	1. The Content Provider has added a social network account
Steps		1. The Content Provider displays his/her profile page 2. The Content Provider enters his/her social account details 3. The Platform stores the information
Parent		
Trigger		The Content Retriever clicks on button
Variations		

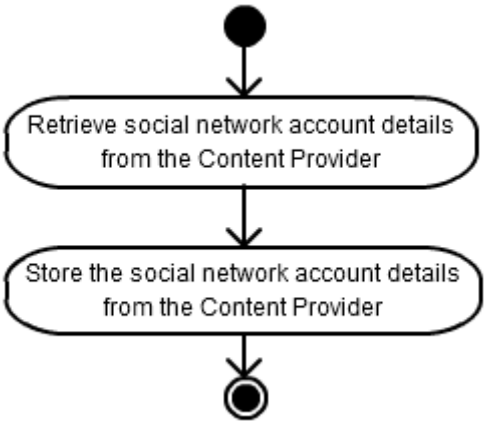


Figure 49: Activity Diagram for inserting a social network account to the Platform

Use Case UCFR 33.2: Disseminate Blog Post

Use Case UCFR 33.2		Disseminate Blog Post
Description		The platform disseminates newly archived items in social networks and mentions thereby the author's account.
Actors		Platform
Assumptions	**Preconditions**	1. The Platform has retrieved aggregated content from the Spider
	Postconditions	1. The Platform has posted a message about the new content
Steps		1. The Platform extracts the author information from the aggregated content 2. The Platform checks if the author had inserted a social network account 3. If yes, The Platform posts a message about the newly archived content by mentioning the authors account in the social network
Parent		
Trigger		The Platform retrieves aggregated content from the Spider
Variations		

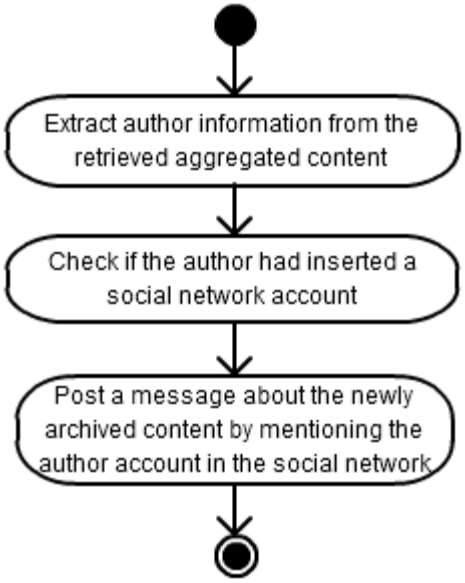


Figure 50: Activity Diagram for dissemination of a blog post

FR34 – Topic/Subject detection

Requirement category	Functional Requirement
Degree of necessity	( ) **Essential** ( ) **Recommended** (X) **Optional**
Description of the requirement	Topic or subject of a blog and blog post should be detected automatically. Thereby, blogs and blog posts can be differentiated and aggregated by topics or subjects.
Stakeholder	Blog author
Justification / Foundation	Blog author: "I would something that is a little bit more insightful. I know that there are text analysis software packages available that you run it through, EPSS or whatever. So, that would be kind of interesting. Comparing the focus, you know, Stephen talks a lot about cognitive structures than George who talks mostly about social structures that would be kind of interesting. I think that would be really hard to do though. But because it is hard to do, that is probably why it would be interesting.
Assessment / Measures	Topics/subjects of blogs and blog posts are detected automatically.
Author(s) of the requirement description	• Hendrik Kalb

Note: UCFR2.1 and UCFR32.1 can apply here

FR35 – Detection and ranking of the originality

Requirement category	Functional Requirement
Degree of necessity	( ) **Essential** ( ) **Recommended** (X) **Optional**
Description of the requirement	It would be very beneficial for archive users to explore which blogs or authors provide the most original posts. Similar criteria like uniqueness or factual trueness are possible as well.
Stakeholder	Blog author
Justification / Foundation	Blog author: "What really interests me about my blog (with respect to inter-relations to other people) is whether it is first to come out with a concept or an idea, and I have no idea if you guys can rank that or if you can rank that automatically. It interests me if I authored a unique (well, no, not even unique, it doesn't need to be unique), an informed, an insightful perspective or a point of view that matters. I would prefer to be right more often than other blogs. To me, the number one ranking would be: "I have more factually true statements in my blog than any other blog", demonstrably and knowably so. But who is going to rank then based on that? And you do not want to rank that trivially, because someone will just start posting dictionary articles, and they get lots of factually true statements. So, say, the highest number of factually true statements that are contextually relevant to the current debate."
Assessment / Measures	Originality and similar concepts are detected and ranked.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCFR 35.1: Originality Ranking

Use Case UCFR 35.1		Originality Ranking
Description		The Platform is able to detect and rank the originality of the blog posts
Actors		Platform
Assumptions	**Preconditions**	1. The Platform has been schedule to go through all the blog posts
	Postconditions	1. The Platform has ranked the blog posts according to their originality
Steps		1. The Platform starts to go through all the blog posts 2. The Platform determines the originality of each blog post 3. The Platform ranks the blog post according to its originality
Parent		
Trigger		The prescheduled time has come
Variations		

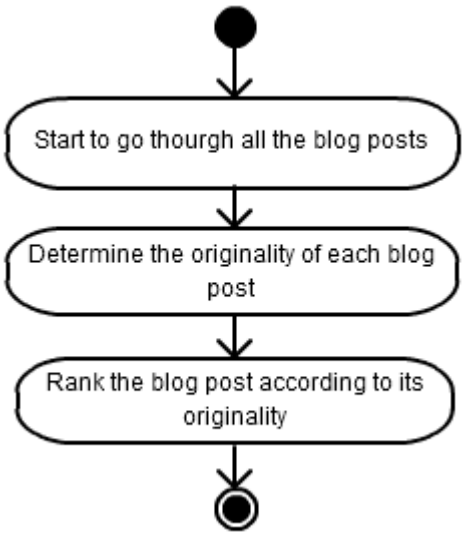


Figure 51: Activity Diagram for originality ranking

4.2 Data Requirements

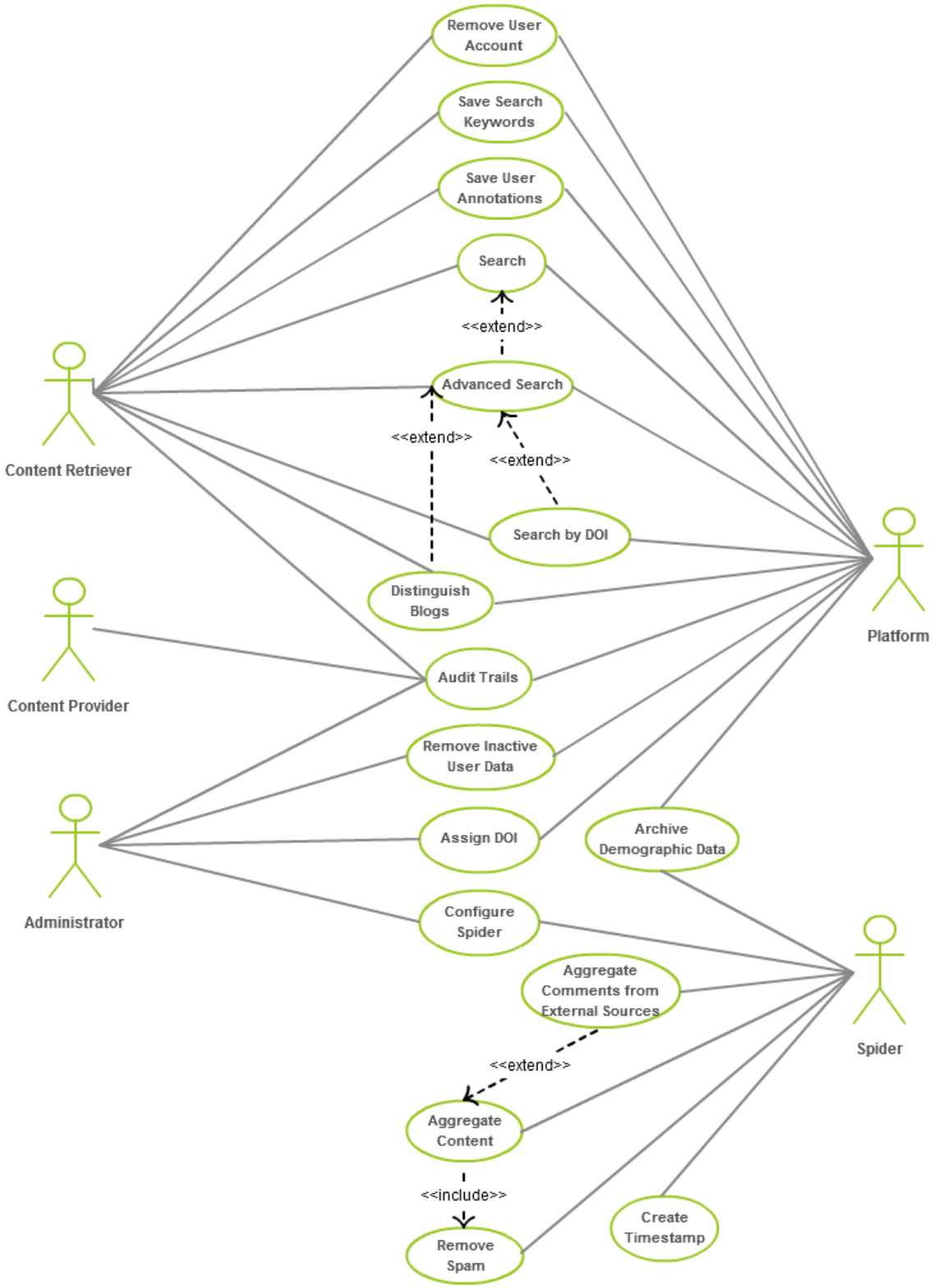


Figure 52: Use Case Diagram 1 for the Data Requirements

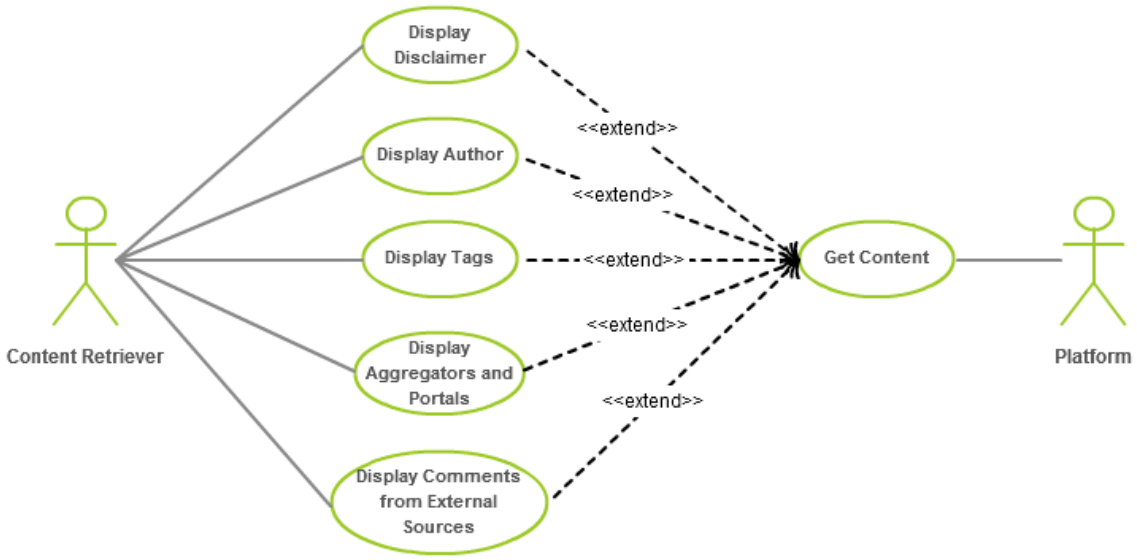


Figure 53: Use Case Diagram 2 for the Data Requirements

The following requirements are also data requirements but have already been defined above. Therefore, they are presented here but are not described again:

- FR5 – Descriptive statistics for a single blog or blog post
- FR6 – Processing of Licenses
- FR8 – Topics (Categories) for blogs and blog posts
- FR9 – Content translation

DR1 – Rights and Licenses

Requirement category	Data Requirement, User Interface
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	Blogs (and embedded objects) can be published under a specific license or with specific rights. That information has to be stored and presented to the user. Content from other sources should be identified and associated with a disclaimer around copyright infringement to protect the archive and hosting institution.
Stakeholder	Blog author, Content Retrievers – Libraries, Content Providers/Retrievers
Justification / Foundation	Blog author: "If I had anything that I thought would need rights on it, I would use Creative Commons." Library: We sign licence agreements with the external resources, so some of the articles, journals or content can only be used for academic purposes. If any users violate this rule, we kindly warn them. Interview#1~4:30: "At our organization we say: You can't use that content, unless you ask for a specific permission, and there is a single website (with the copyright information) that describes that..." Blog author: "I use a Creative Commons, by-non-commercial share-like. It is supposed to be 3.0 [...]"
Assessment / Measures	Licenses and Rights of a blog are stored in the archive and are presented to the user for every view/access on the blog or parts of the blog. The user should be able to view copyright information of the content. The copyright or license of the original content should be conserved, visible and applicable to the archived content as well.
Author(s) of the requirement description	• Hendrik Kalb • Senan Postaci • Jaime García • Nikolaos Kasioumis

Use Case UCDR 1.1: Get Content

Use Case UCDR1.1		Get Content
Description		The Content Retriever can access blog content if s/he is authorized, otherwise cannot
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever clicks on a blog page to retrieve its contents
	Postconditions	1. The Content Retriever accesses blog content or gets an "access not authorized" message
Steps		1. The Platform checks if the Content Retriever is able to access blog content (against copyright, privacy, data protection laws) 2. If yes, the Platform retrieves all the blog content and displays 3. The Platform logs the Content Retriever information so that it can be tracked later 4. Otherwise, The Platform displays an "access not authorized" message
Parent		
Trigger		The Content Retriever clicks on a link to the blog post
Variations		
Issues		

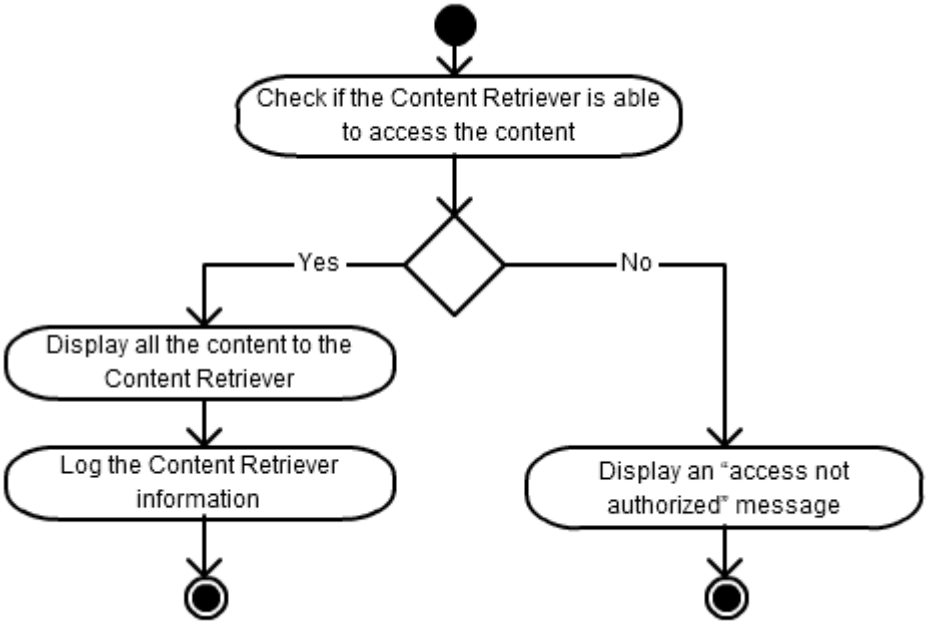


Figure 54: Activity Diagram for getting content

DR2 – URI and metadata for referencing/citing

Requirement category	Data Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	It must be technically possible to reference or cite content from the archive. Therefore, a unique address for the content is required (unique resource identifier) and metadata about the author and date of creation should be presented if available.
Stakeholder	Blog Author, Content Retriever Library
Justification / Foundation	Blog Author: "But I would want it to be cited properly. If people were going to reuse content. If it were exhibition quality that people would going to reuse it for other purposes that they could actually correctly cite it. So, I think that would be important for many of the blog users I know. But if it was going to be some useful reference elsewhere, and not just for your access for reminding you what you have posted, then correct citation would rank pretty highly." Library: Every electronic publication gets a permanent address in [in our archive]. It will be accessible under this address forever.
Assessment / Measures	Every content that is presented as a single entity to the user (e.g. blog, blog post) has a unique address. Additionally, Metadata about the author and date of creation are presented.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCDR 2.1: Assign DOI

Use Case UCDR 2.1		Assign DOI
Description		The Administrator can assign DOI to the content in the archive
Actors		Platform, Administrator
Assumptions	**Preconditions**	1. The Administrator views a blog page
	Postconditions	1. The Administrator has assigned DOI to a blog page
Steps		1. The Administrator enters a new DOI to a box provided 2. The Platform checks if the DOI provided, used before 3. If yes, the Platform displays an error message 4. Otherwise, the Platform changes the DOI of the blog page
Parent		
Trigger		The Administrator enters a DOI to a box provided
Variations		
Issues		

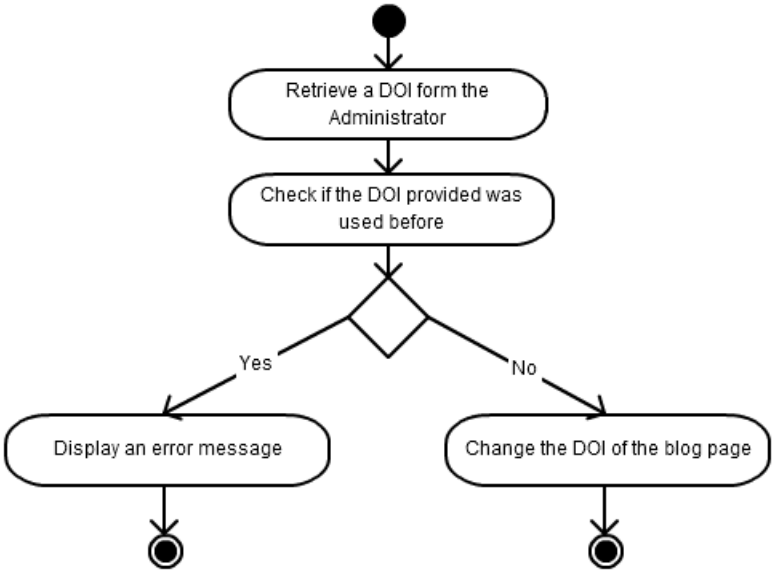


Figure 55: Activity Diagram for assigning a DOI to a blog page

Use Case UCDR 2.2: Search by DOI

Use Case UCDR 2.2		Search by DOI
Description		The Content Retriever can search a blog page by its DOI
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever has viewed the Advanced Search panel
	Postconditions	1. The Content has viewed search results
Steps		1. The Content Retriever enters the DOI of the blog page through the Search Interface 2. The Content Retriever select “DOI” from the “category” box 3. The Platform displays the search result
Parent		UCFR13.1
Trigger		The Content Retriever enters the DOI of the blog page
Variations		
Issues		

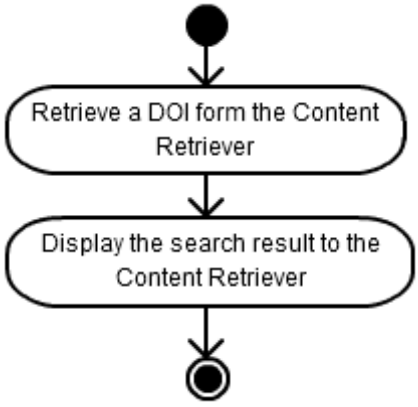


Figure 56: Activity Diagram for searching a blog page by its DOI

DR3 - Disclaimer

Requirement category	Data Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	The disclaimer of a blog should be stored and presented with every part/content of the blog.
Stakeholder	Blog Author
Justification / Foundation	Blog Author: "you can always have a disclaimer when you are producing any publication to say that you have done your best to capture what you have done with the authority that you thought you had to do it. And should anyone have any reason why they would like anything removed or corrected, then they have some point of contact. But that's the general way, in this day and age; produce a disclaimer on publicising or archiving or reproducing any piece of work."
Assessment / Measures	The disclaimer of a blog is stored and presented with every part/content of the blog.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCDR 3.1: Display Disclaimer

Use Case UCDR3.1		Display Disclaimer
Description		The disclaimer of a blog is displayed with every part/content of the blog.
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever clicks on a blog page to retrieve its contents
	Postconditions	1. The Platform displays the disclaimer of the blog
Steps		1. The Platform retrieves the disclaimer of the blog along with the blog content 2. The Platform displays the disclaimer with the blog content
Parent		UCDR1.1
Trigger		The Content Retriever clicks on a link to the blog post
Variations		
Issues		

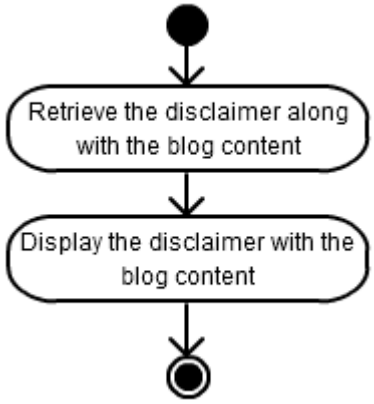


Figure 57: Activity Diagram displaying the disclaimer

DR4 – Display the author of the blog, blog post, and comment

Requirement category	Data Requirement, User Interface Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	The author of a blog, blog post or comment has to be presented.
Stakeholder	Blog Reader, Businesses
Justification / Foundation	Blog Reader: "That depends on who is the author. I think I have got to the point of selecting those that I will read, and I will have some confidence that they know what they are talking about." Blog Author: "When I make comment on the other websites, much of the time (almost all of the times these days) they ask for an email address, name and website URL, which means my comment appears on the other website and when they click on my name, they go to my website."
Assessment / Measures	The author of the blog, blog post, or comment is presented with the blog, blog post, or comment.
Author(s) of the requirement description	• Hendrik Kalb

Note: "Use Case UCFR 15.1: Configure Spider" can apply here

Use Case UCDR 4.1: Display Author

Use Case UCDR4.1		Display Author
Description		The author of a blog, blog post and comment is displayed with the blog, blog post, or comment.
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever clicks on a blog page to retrieve its contents
	Postconditions	1. The Platform displays the author(s) of the blog, blog post, or comments
Steps		1. The Platform retrieves the author(s) of the blog, blog post, comments along with the blog content 2. The Platform displays the author(s) of blog, blog posts and comments with the blog content
Parent		UCDR1.1
Trigger		The Content Retriever clicks on a link to the blog post
Variations		
Issues		

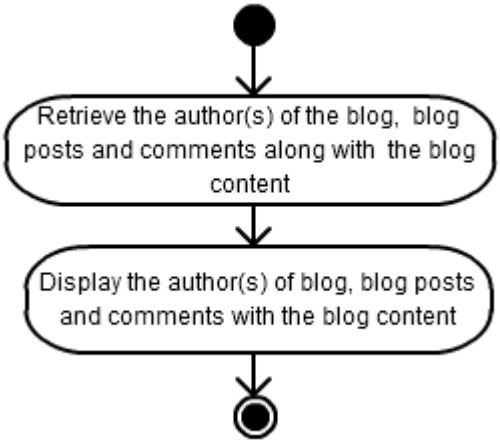


Figure 58: Activity Diagram displaying the author

DR5 – Tags of the blog or blog post

Requirement category	Data Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	Tags of the blog or blog post should be stored and presented to the archive user.
Stakeholder	Blog Reader
Justification / Foundation	Blog Reader: “Basic level tags [...]” Blog Reader: “they are good at making sure they have tagged up”
Assessment / Measures	Tags of the blog or blog post are stored and presented to the archive user.
Author(s) of the requirement description	• Hendrik Kalb

Note: “Use Case UCFR 15.1: Configure Spider” can apply here

Use Case UCDR 5.1: Display Tags

Use Case UCDR5.1		Display Tags
Description		Tags of a blog or blog post are displayed to the users of the platform
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever clicks on a blog page to retrieve its contents
	Postconditions	1. The Platform displays the tags of the blog and blog posts
Steps		1. The Platform retrieves the tags of the blog and blog posts along with the blog content 2. The Platform displays the tags of the blog and blog posts
Parent		UCDR1.1
Trigger		The Content Retriever clicks on a link to the blog post
Variations		
Issues		

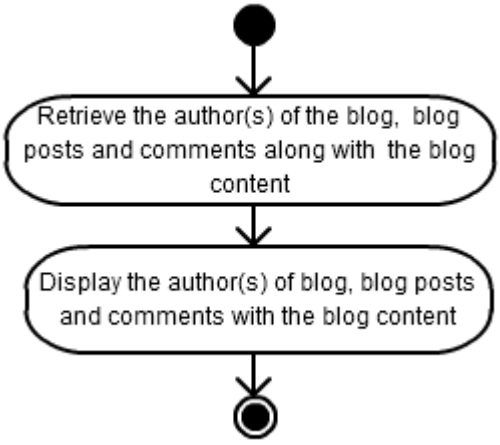


Figure 59: Activity Diagram displaying the tags of the blogs and blog posts

DR6 – Metadata for captured contents

Requirement category	Data Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	Every captured digital object needs a description with regard to the content of the object (e.g. topic, language, etc.) and to technical aspects (e.g. formats).
Stakeholder	Content Retriever Library, Researcher, Businesses
Justification / Foundation	Content Retriever Library: For every publication that goes into our archive, we describe the publication and we must have a technical description.
Assessment / Measures	Descriptions with regard to the content and to technical aspects are stored for every digital object.
Author(s) of the requirement description	• Hendrik Kalb

Note: “Use Case UCFR 15.1: Configure Spider” can apply here

DR7 – Date/timestamp for creation and capturing

Requirement category	Data Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	It has to be stored when the captured content (e.g. blog post) has been created and when it has been captured.
Stakeholder	Content Retriever Library
Justification / Foundation	Library: Time is shown, when the Blog was gathered
Assessment / Measures	Date/Time is stored for the creation and for the capturing of content in the archive.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCDR 7.1: Create Timestamp

Use Case UCDR7.1		Search by Ranking
Description		The Spider creates a timestamp information for the aggregated content
Actors		Spider
Assumptions	**Preconditions**	1. The Spider has aggregated content
	Postconditions	1. The Spider has sent the content to the Platform
Steps		1. The Spider creates a timestamp information 2. The Spider appends the time information to the content 3. The Spider sends all the content to the Platform
Parent		
Trigger		The Spider aggregates blog content
Variations		
Issues		

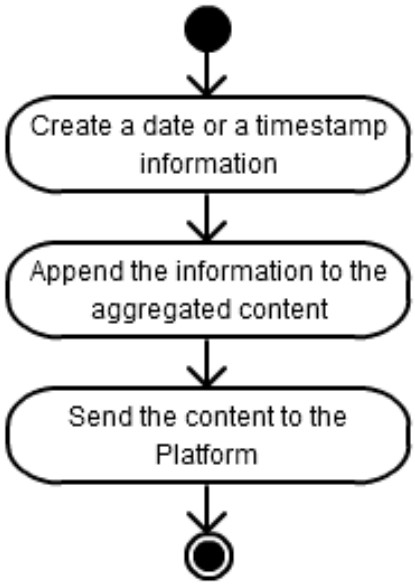


Figure 60: Activity diagram for creating time information

DR8 – Aggregators & Portals

Requirement category	Data Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	A blog or blog posts can be published in several portals or presented by several aggregators. It should be stored where the captured blog or blog post occurred.
Stakeholder	Researcher
Justification / Foundation	Researcher: "Because my focus is on science-blogs, I have used manually compiled lists of science-blogs, so there are some platforms which aggregate science-blogs like scienceblogs.com or scienceblogs.de or wissenslogs.de which is a German aggregator of scientific blogs"
Assessment / Measures	It is stored with the blog or blog post in which portal or aggregator the blog or blog post occurred.
Author(s) of the requirement description	• Hendrik Kalb

Note: "Use Case UCFR 15.1: Configure Spider" can apply here

Use Case UCDR 8.1: Display Portal and Aggregators

Use Case UCDR8.1		Display Portal and Aggregators
Description		Portal and aggregator in which blog or blog post occurred is displayed with the blog content
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever clicks on a blog page to retrieve its contents
	Postconditions	1. The Platform displays the tags of the blog and blog posts
Steps		1. The Platform retrieves portal and aggregator information of the blog and blog posts along with the blog content 2. The Platform displays the portal and aggregator information of the blog and blog posts
Parent		UCDR1.1
Trigger		The Content Retriever clicks on a link to the blog post
Variations		
Issues		

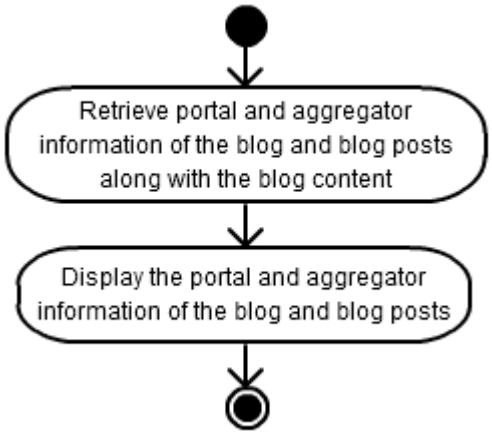


Figure 61: Activity Diagram displaying the portal and aggregator information of the blogs and blog posts

DR9 – Connections/Links

Requirement category	Data Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	A blog creates connections to other blogs in several ways, e.g. blog roll, links in blog posts, etc. These connections have to be stored in an accessible way that allows a processing of the interconnections of blogs. The archive software should allow original blog interconnections data to be available to understand content connections and original cross –promotion paths.
Stakeholder	Researcher, Businesses, Content providers and content retrievers
Justification / Foundation	Researcher: "Blogs are connected into networks through the blog-rolls so if there was a tool which could also compile the connected blogs that would help identifying relevant blogs" Businesses: "Knowing more about the blog site: [...] influencers etc. - for segmentation and analyzing" Content providers and content retrievers: One of the last Museum blog post in the University museums blog was linked to other blogs (Post Grad Blog). They used just linking (connect) tools and they provide cross-promotion. It is important aspect although some of the links don't stay overtime.
Assessment / Measures	Different kinds of links (blog roll, links inside posts, pingback, etc.) are stored separately.
Author(s) of the requirement description	• Hendrik Kalb • Silvia Arango-Docio

DR10 – User Annotations

Requirement category	Data Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	Archive users may want to annotate the content in the archive for several reasons, e.g. highlighting, commenting, etc. Therefore, the archive should be able to store the annotations for different users.
Stakeholder	Researcher
Justification / Foundation	Researcher:" because working with the websites would be to actually be able to access the websites while they are online and put the codes in an extra layer - not sure if you are familiar with the Diigo bookmarking platform - there you can add your notes and markings to websites but they are not actually added to website but they are saved in Diigo as an extra layer and I think that would have been actually the solution which have been ideal so that you don't have to save the pages, so that you - you know you sort of bypass all this problem with format-saving and also with saving personal data which you have to do if you want to otherwise added into ATLAS.ti, so I think that would have been a nicely solution." Researcher: I use http://www.mendeley.com/ and I like it very much! [...] I can highlight a part of a document and save this as a note.
Assessment / Measures	The archive stores user annotations on archived content.
Author(s) of the requirement description	• Hendrik Kalb • Vangelis Banos

Use Case UCDR 10.1: Save User Annotations

Use Case UCDR10.1		Save User Annotations
Description		The Platform can store the annotations for different users.
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever has displayed a blog
	Postconditions	1. The annotations of the Content Retriever has been stored
Steps		1. The Content Retriever annotates on the content 2. The Platform links the annotation with the Content Retriever's profile. 3. The Platform stores the annotation
Parent		
Trigger		The user creates an annotation on archived content.
Variations		
Issues		

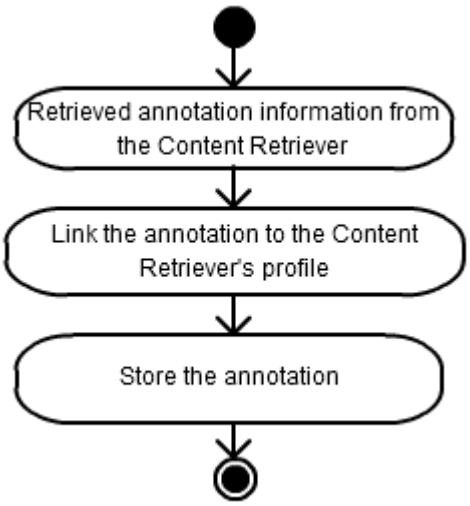


Figure 62: Activity Diagram for saving user annotations

DR11 – Differentiate between blog and blog post

Requirement category	Data Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	The whole blog as well as single blog posts can be interesting for archive users. Therefore, the archive should distinguish between the blog and blog posts. Access should be available for both perspectives.
Stakeholder	Researcher
Justification / Foundation	Researcher: “I think it would be important for a software or platform to differentiate between the blog, as an entity which contains the text pages like the authors profile or description or whatever, and the posts which are chronological added.”
Assessment / Measures	Blogs and blog posts are individually addressable/ can be accessed differently.
Author(s) of the requirement description	• Hendrik Kalb

DR12 - Demographics

Requirement category	Data Requirement
Degree of necessity	( ) **Essential** ( ) **Recommended** (X) **Optional**
Description of the requirement	Analysis on blogs and blog content would benefit if demographic data about blog authors were available.
Stakeholder	Businesses
Justification / Foundation	Businesses: “Within this getting metadata are there something metadata that’s sort of like the big issue? One really would need to have all that really challenge to get?” “Demographics”
Assessment / Measures	Demographic data are stored with the blog authors.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCDR 12.1: Archive Demographic Data

Use Case UCDR12.1		Archive Demographic Data
Description		The Platform can archive the demographic data about the authors
Actors		Platform, Spider
Assumptions	**Preconditions**	1. The Spider has aggregated blog content
	Postconditions	1. The demographic data of the author has been stored
Steps		1. The Spider extracts demographic data of the author from the aggregated content 2. The Spider sends the demographic data to the Platform 3. The Platform archives the demographic data of the author
Parent		
Trigger		The Platform retrieves demographic data from the Spider
Variations		
Issues		

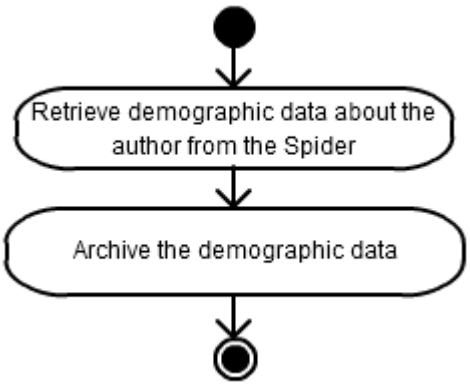


Figure 63: Activity Diagram for archiving demographic data about the author

DR13 - Comments

Requirement category	Data Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	Comments of a blog post have to be stored.
Stakeholder	Businesses
Justification / Foundation	Businesses: “Comments - we currently don’t cover this - which is regarded very interesting”
Assessment / Measures	Comments of a blog post are stored in the archive.
Author(s) of the requirement description	• Hendrik Kalb

Note: “Use Case UCFR 15.1: Configure Spider” can apply here

DR14 – Embedded objects

Requirement category	Data Requirement
Degree of necessity	( )**Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	Embedded objects (e.g. YouTube videos) should be stored in the archive.
Stakeholder	Businesses
Justification / Foundation	Businesses: "Links and meta data - today we only capture text. Especially pictures and videos would be useful."
Assessment / Measures	Embedded objects of a blog/blog post are stored in the archive.
Author(s) of the requirement description	• Hendrik Kalb

Note: "Use Case UCFR 15.1: Configure Spider" can apply here

DR15 – Visits of blogs and blog posts

Requirement category	Data Requirement
Degree of necessity	( )**Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	It should be stored for each blog or blog post how often does who visit the blog or post in the archive.
Stakeholder	Blog Author
Justification / Foundation	Blog Author: You can see the visitors and what they have visited.
Assessment / Measures	Visitors/Visits of a blog or blog post are stored.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCDR 15.1: Audit Trails

Use Case UCDR15.1		Audit Trails
Description		The Administrator can track all the operations (Who accessed? When? Content updated?, etc.) on a blog page
Actors		Platform, Administrator, Content Retriever, Content Provider
Assumptions	**Preconditions**	1. The user (Admin, CR, CP) displays a blog page
	Postconditions	1. The user views all the operations on blog pages
Steps		1. The user clicks on a link/button/tab 2. The Platform retrieves a list of all the operations performed on a blog page (all accesses, access times, changes, etc.…) 3. The Platform displays the operations to the user
Parent		
Trigger		The user clicks on a link/button/tab
Variations		
Issues		

Figure 64: Activity diagram for auditing trails on a blog page

DR16 – Search key words

Requirement category	Data Requirement
Degree of necessity	( )**Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	It should be stored by which key words a blog or blog post in the archive has been searched and found in a search via search machine (e.g. Google search)..
Stakeholder	Blog Author
Justification / Foundation	Blog Author: I can see which key words they have used in the search machine.
Assessment / Measures	It is stored which key words were used to find the blog or blog post via search machine.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCDR 16.1: Save Search Keywords

Use Case UCDR16.1		Save Search Keywords
Description		The Platform stores the keywords that are used to find a blog or blog post via search machine
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever has entered keywords and pressed "Search" button
	Postconditions	1. The Platform has saved the keywords/queries for the blogs that are in the search result
Steps		1. The Platform retrieves the search results 2. The Platform saves the keywords/queries for the blogs in the search results
Parent		UCFR14
Trigger		The user clicks on a button
Variations		
Issues		

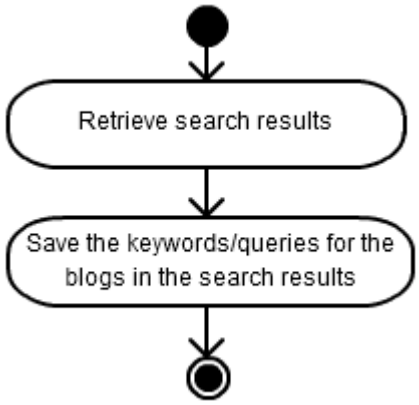


Figure 65: Activity diagram for saving search results

DR17 – Metadata for blogs

Requirement category	Data Requirement
Degree of necessity	( )**Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	Metadata should be available for a blog. Metadata describe • Language of the blog • Categorisation in hobby-blog, professional blog, etc. • Is it a individual blog, company blog, or institution blog • Subject/Topic of the blog, • Geographical information
Stakeholder	Researcher, Businesses
Justification / Foundation	Researcher: "[...] concerning the subject of the blog or a sort of categorization if it's a hobby-blog or if it's a professional blog and also meta-data concerning the owner, if it's an individual blog or a company-blog or institution-blog." Businesses: "in some cases the border between journalism and blogging, professional journalism and blogging, is very thin and we basically want to keep the data so that it serves consumer, brands as well as possible and sometimes we for example because of that we exclude news items or news comments even though they are regarded as social media we exclude that from the data but that's optional so basically just try to have as good reach as possible within the right language and geographical context."
Assessment / Measures	Metadata of a blog are stored in the archive.
Author(s) of the requirement description	• Hendrik Kalb

Note: "Use Case UCFR 32.1: Aggregate Content" can apply here

DR18 – Remove private data of archive users

Requirement category	Data Requirement
Degree of necessity	( )**Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	It may be necessary that user data are removed/deleted when the archive user does not use the archive anymore. Therefore, usage data of archive users should be captured only if really necessary and should be removed after the user has left the archive.
Stakeholder	Content Retriever Library
Justification / Foundation	Library: We do neither track what current users are reading or accessing nor do we store user data a long time after they have visit us the last time. We delete the data as soon as the library card ends.
Assessment / Measures	User data are removed automatically after a specific period (e.g. six month after the last login) or point in time (e.g. the user has to prolong archive membership every year).
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCDR 18.1: Remove Inactive User Data

Use Case UCDR18.1		Remove Inactive User Data
Description		Remove the user data (profile of user, comments, etc.) if the user does not login for a specific period.
Actors		Platform, Administrator
Assumptions	**Preconditions**	1. The Administrator has displayed the Admin Interface
	Postconditions	1. Inactive users and their private data are removed.
Steps		1. The Administrator enters a specific period. 2. The Platform checks every user's last login time. 3. If there are users who are inactive during the specified period, remove their data.
Parent		
Trigger		The Administrator specifies a period for the Platform to scan inactive users.
Variations		
Issues		

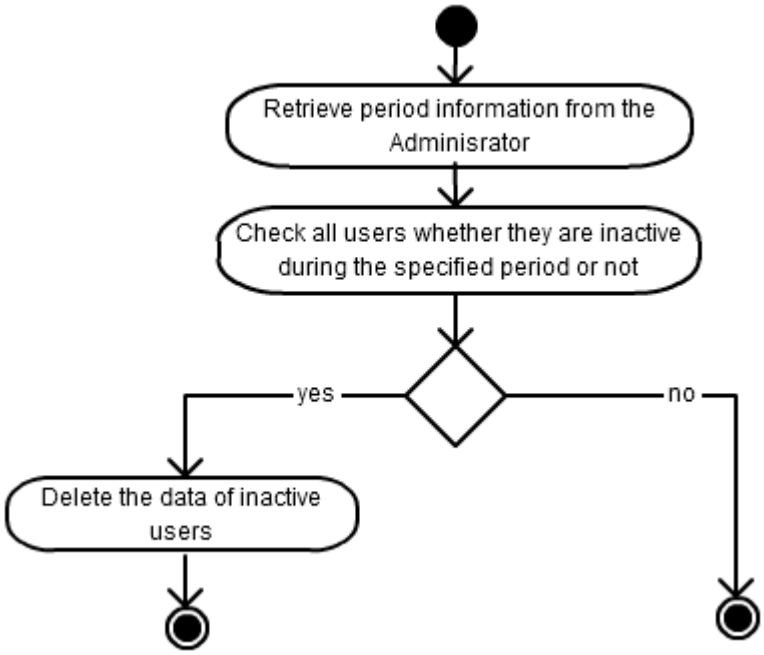


Figure 66: Activity Diagram for removing inactive users

Use Case UCDR 18.2: Remove User Account

Use Case UCDR18.2		Remove User Account
Description		The users of the platform can delete their account and user data (profile of user, comments, etc.)
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever has displayed his/her profile page
	Postconditions	1. The account and user data of the Content Retriever has been deleted
Steps		1. The Content Retriever clicks on a "Remove Account" button 2. The Platform displays a dialog page about removing/keeping the Content Retriever's user data 3. The Content Retriever selects the "Remove" option (rather than "Keep") 4. The Platform removes the Content Retriever's account and his/her user data
Parent		
Trigger		The Content Retriever clicks on a button
Variations		
Issues		

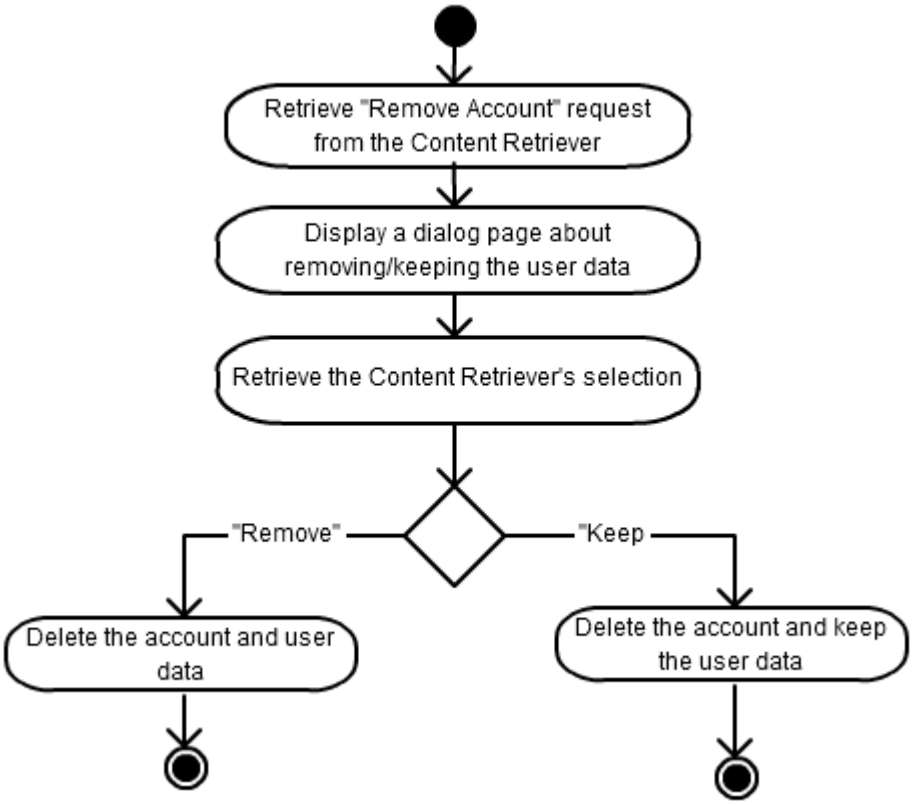


Figure 67: Activity Diagram for removing user accounts and data

DR19 – Distinguish institutional/corporate blogs from personal blogs

Requirement category	Data Requirement
Degree of necessity	( ) **Essential** ( ) **Recommended** (X) **Optional**
Description of the requirement	If the repository size becomes very big, it might be easier to navigate if institutional/corporate blogs are separated from personal blogs.
Stakeholder	Content retrievers
Justification / Foundation	Interview#1 ~16:10 "If you imagine a system where there are personal blogs, versus corporate, institutional blogs, you mind want to separate them, with different facets..."
Assessment / Measures	Institutional/corporate blogs and personal blogs are distinguished and can be searched separately.
Author(s)of the requirement description	• Jaime García • Nikolaos Kasioumis

Use Case UCDR 19.1: Distinguish Blogs

Use Case UCDR19.1		Distinguish Blogs
Description		The user is able to narrow down the search result (e.g. results from personal blogs).
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever has displayed Advance Search panel
	Postconditions	1. The search results are displayed
Steps		1. The Content Retriever enters the search keywords 2. The Content Retriever selects a field from the available fields 3. The Content Retriever selects a blog type (personal, institutional, etc.) 4. The Platform executes the query retrieved from the Content Retriever 5. The Platform retrieves the matching records 6. The Platform displays the search results
Parent		UCFR13.1
Trigger		The clicks search button.
Variations		
Issues		

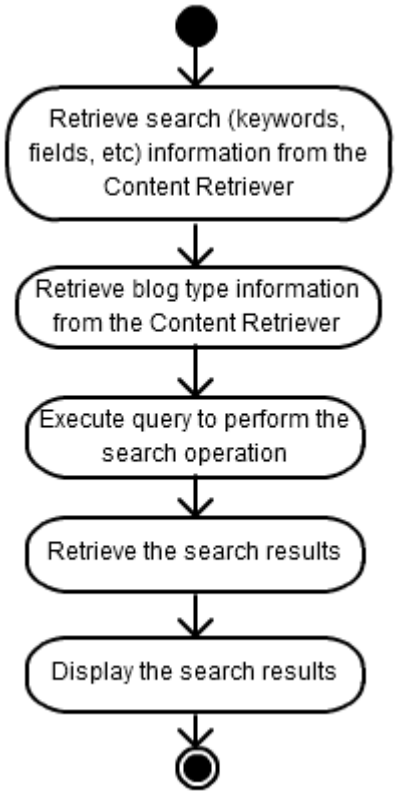


Figure 68: Activity Diagram for distinguishing blogs

DR20 - Blog comments from several sources

Requirement category	Data Requirements
Degree of necessity	( )**Essential** (X) **Recommended** ( )**Optional**
Description of the requirement	Data comments attached to their content from different sources amalgamated in one place of the blog post archived, easy to access and well connected to their corresponding post, in particular the comments coming from social media links (the Research Institute Facebook page has increased their comments levels so they need to see those in the blog archive). The comments originated in social media applications should be listed under the logo of these social media sources.
Stakeholder	Content Providers, Content providers and content retrievers
Justification / Foundation	Phrases from the interview: The low number of comments is maybe due to the Research Institute very institutional or very academic background. It seems that the Research Institute Facebook page linked to their blog posts is providing a more flexible platform for comments. People engaging with the historical content to enhance their content (a better resource for everybody). The Research Institute is keen in additions and enhancement to their content. Would you like to aggregate, manage & analyze all your users' feedback? Please describe an ideal scenario. The answer for this question was: This software should fulfill: keeping content and the social media context around that content. Less concern about their blogs look and feel but they are very concerned about their content and its comments. Phrases from the interview: What would you describe as a "killer" application feature for a blog archiving service? Easy to use and that it should provide stats and popularity, comments and tweets feeds plus 'Like' icon usages as well. Social commentary around the blog should be available within this application. (Twitter is like a word of mouth so for example the comment from a German museum in their unofficial blog). Do you have any general comments on the development of blog aggregation, preservation, management & dissemination software?

	Easy to use + social media aspect of the blog. New things coming up and changing of the social media, allowance to changes in social media in the future.
Assessment / Measures	A user can find all the comments listed for a specific blog post divided into boxes with the comments source logo to recognize the original source easily and clearly.
Author(s) of the requirement description	• Silvia Arango-Docio

Use Case UCDR 20.1: Aggregate Comments from External Sources

Use Case UCDR20.1		Aggregate Comments from External Sources
Description		The Spider aggregates comments from external sources
Actors		Spider
Assumptions	**Preconditions**	1. The Content Retriever has provided URL of the blog page
	Postconditions	1. The content of the blog page has been aggregated by the Spider
Steps		1. The Spider looks for the comments of the blog/blog post in external sources 2. The Spider aggregates the comments from external sources 3. The Spider sends the aggregated content and comments from external sources to the Platform
Parent		UCFR32
Trigger		The Spider finds comments in external sources about the blog/blog posts
Variations		
Issues		

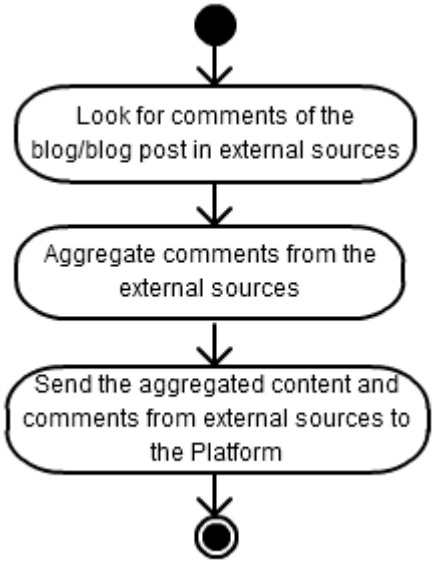


Figure 69: Activity Diagram for aggregating comment from external sources

Use Case UCDR 20.2: Display Comments from External Sources

Use Case UCDR20.2		Display Comments from External Sources
Description		Comments of a blog or blog post from external sources are displayed to the users of the platform
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever clicks on a blog page to retrieve its contents
	Postconditions	1. The Platform displays the comments of the blog and blog posts from external sources
Steps		1. The Platform retrieves the comments of the blog and blog posts from external sources the along with the blog content 2. The Platform groups the comments according to external source (i.e. comments from Facebook, comments from Twitter, etc.) 3. The Platform displays the comment of the blog and blog posts in separate boxes each having the logo of the external source
Parent		UCDR1.1
Trigger		The Content Retriever clicks on a link to the blog post
Variations		
Issues		

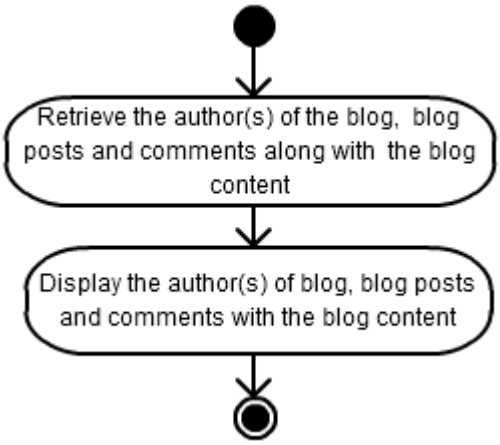


Figure 70: Activity Diagram displaying the tags of the blogs and blog posts

4.3 Interoperability Requirements

Figure 71: Use Case Diagram for the Interoperability Requirements

The following requirement is also an interoperability requirement but has already been defined above. Therefore, it is presented here but is not described again:

- FR4 – Blog export

IR1 – Single Sign On/Interoperates with Authentication System especially LDAP

Requirement category	Interoperability Requirement
Degree of necessity	( )**Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	The archive should be able to interoperate with common authentication systems. This facilitates the integration of the archive in the existing IT environment and enables the provision of single sign on for users. The BlogForever Platform authentication mechanism should be able to connect to LDAP servers.
Stakeholder	Admin Organisation, Content Retrievers – Libraries, Blog Author
Justification / Foundation	Admin: "then I would say that we would be interested in questions like authentication. We use Shibboleth as our single sign on platform, so we would be very keen on applications that would work well with that." Admin: "It is not, really, an API type question, but we use OAuth, so that people can get their content which is in some sense protected without having to have an account with us. They can have an account with someone else and OAuth would let them get the things" Library: We provide a search engine to the users which enables them to search many articles, journals from external repositories and these repositories are connected with our LDAP server. Library: Our repositories are connected with the LDAP server of the University. Students and other users are authenticated via the LDAP to access the repositories. Apart from this, there is no other integration between repositories and library services. Blog Author: "So, it would be nice if such a system would actually support access to my blogging system, whatever that may be, through a mechanism such as OAuth or some such thing."
Assessment / Measures	The archive can interoperate with authentication systems. The users can connect to a BlogForever archive by using LDAP authentication.
Author(s) of the requirement	• Hendrik Kalb • Senan Postaci • Vangelis Banos

Use Case UCIR 1.1: Universal Login

Use Case UCIR1.1	Universal Login
Description	The users of the BlogForever Platform connect to the system with existing third party services or authentication mechanisms

Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. An account for the Content Retriever from the third party service or authentication mechanism must exist.
	Postconditions	1. The Content Retriever has been authenticated
Steps		1. The Content Retriever specifies required information for login (for example: username, password…). 2. The Platform sends the credentials of the Content Retriever to the third party service or authentication mechanism 3. The Platform receives the result from the third party service or authentication mechanism 4. The Platform navigates Content Retriever according to result received from the third party service or authentication mechanism
Parent		
Trigger		Content Retriever clicks on appropriate icon among the available third party services or authentication mechanisms
Variations		
Issues		

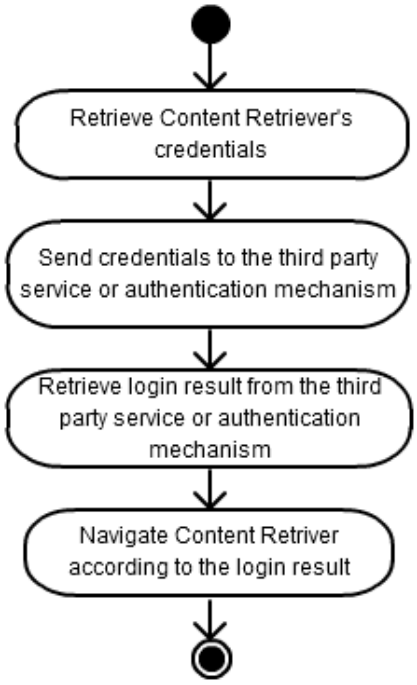


Figure 72: Activity Diagram for universal login

IR2 – Capturing is possible for various platforms

Requirement category	Interoperability Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	Blogs are available on different platforms. The archive should not be restricted to only one kind of platform or software because it would limit the amount of blogs that can be archived extremely. Therefore, the archive respectively the spider should be able to capture blogs ideally from every platform or at least from the most common platforms.
Stakeholder	Business
Justification / Foundation	Business: “diversity of the platforms where we can collect the data”
Assessment / Measures	The archive respectively the spider captures blogs from the most common platforms.
Author(s) of the requirement description	• Hendrik Kalb

Note: “Use Case UCFR 15.1: Configure Spider” can apply here

IR3 - Export data using OAI-PMH protocol and Dublin Core schema

Requirement category	Interoperability Requirements
Degree of necessity	(**X**) **Essential** ( ) **Recommended** ( )**Optional**
Description of the requirement	The BlogForever platform should enable data export using OAI-PMH protocol. & Dublin Core metadata schema in order to interoperate with other digital libraries.
Stakeholder	Admins – Organizations
Justification / Foundation	Interviewee says: It is quite common to extract data from legacy systems in MARC and import them in a repository with Dublin Core metadata. Digital repositories support OAI-PMH and give data to europeanaLocal and other projects. Interviewer: Are you supporting any APIs/Technologies? Interviewee says: The most common is OAI-PMH, all repositories support it.
Assessment / Measures	An OAI-PMH interface is enabled, supporting at least Dublin Core Metadata Schema.
Author(s) of the requirement description	Vangelis Banos

Use Case UCIR 3.1: Add OpenURL for OAI-PMH

Use Case UCIR3.1		Add OpenURL for OAI-PMH
Description		The Administrator can provide OpenURLs for the content that is exposed over the OAI-PMH protocol
Actors		Platform, Administrator
Assumptions	**Preconditions**	1. The Administrator has displayed the Admin Interface
	Postconditions	1. The Platform has created links to the content that are ready to export over OAI-PMH protocol
Steps		1. The Administrator provides an OpenURL of the content 2. The Platform parses the OpenURL and constructs links to the content in the specified format 3. The Platform stores the links to content
Parent		
Trigger		The Administrator enters an OpenURL
Variations		
Issues		

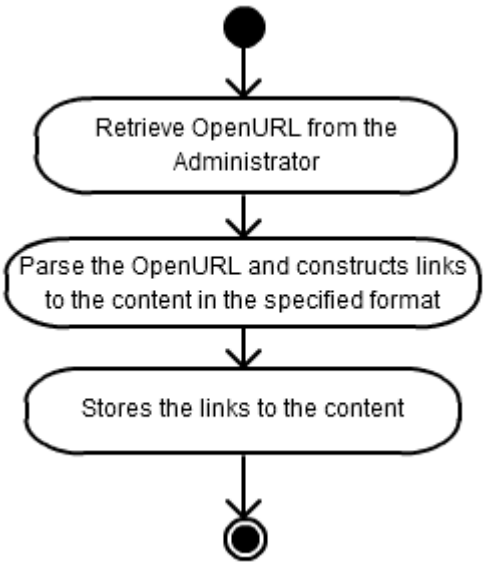


Figure 73: Activity Diagram for adding an OpenURL to export content over OAI-PMH

IR4 – Expose parts of the archive via OAI-PMH based on specified criteria

Requirement category	Interoperability requirements, Security Requirements
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	The BlogForever platform should enable exposing different parts of the archive to different clients via OAI-PMH according to specified parameters & policy. (E.g. client identity, admin settings).
Stakeholder	Admins - Organizations
Justification / Foundation	Interviewee says: We are looking into how OAI-PMH is mapping available data sets and how we can expose different data to different clients or present the same dataset in different ways.
Assessment / Measures	An administrator can select which parts of the archive are available via OAI-PMH. The administrator can setup different OAI-PMH endpoints and define which parts of the collection are available via these endpoints. Access to the endpoints is controlled via authentication (username/password, IP address) or is public.
Author(s) of the requirement description	Vangelis Banos

Use Case UCIR(IR) 4.1: Expose Archive

Use Case UCIR4.1		Expose Archive
Description		The BlogForever platform can expose different parts of the archive to different clients via OAI-PMH according to specified parameters
Actors		Platform, Administrator
Assumptions	**Preconditions**	1. The Administrator has displayed the Admin Interface
	Postconditions	1. The Platform has created a new OAI-PMH end point for a specific part of the archive
Steps		1. The Administrator defines a new OAI-PMH end point 2. The Administrator specifies the security policy for the new endpoint (IP restrictions, authentication settings, etc.) 3. The Administrator specifies the parts of the archive that will be available via new end point 4. The Platform exposes the specified content over the new OAI-PMH endpoint
Parent		
Trigger		Content Retriever sends a request over OAI-PMH protocol
Variations		
Issues		

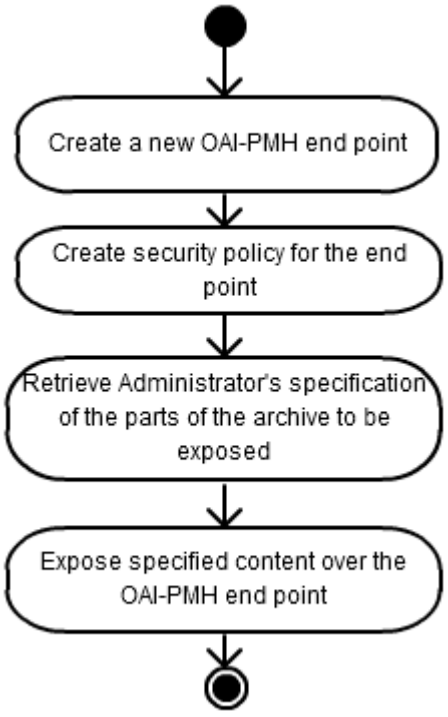


Figure 74: Activity Diagram for exposing content over an OAI-PMH endpoint

IR5 – Connection with federated search engine dbwiz

Requirement category	Interoperability Requirements
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	The BlogForever platform should be able to interoperate with federated search engine software dbwiz (http://researcher.sfu.ca/dbwiz) via OAI-PMH.
Stakeholder	Content Retrievers - Libraries
Justification / Foundation	Interviewee says: The library has a federated search engine (http://pantou.lib.uom.gr ,dbwiz based) which is connected with many sources including the open repository of the University.
Assessment / Measures	A dbwiz software installation can add BlogForever archive in its sources and perform federated search in its content.
Author(s) of the requirement description	Vangelis Banos

Use Case UCIR(IR) 5.1: Search in dbwiz

Use Case UCIR5.1		Search in dbwiz
Description		The users of the platform are able to search content in the archive by using a dbwiz installation
Actors		Content Retriever
Assumptions	**Preconditions**	1. Content Retriever has displayed the dbwiz search interface
	Postconditions	1. The Content Retriever has retrieved the search results
Steps		1. The Content Retriever enters the search keywords 2. dbwiz retrieves a list of the search results from the BlogForever Platform 3. dbwiz displays the search results
Parent		
Trigger		The Content Retriever has provided the search keywords in dbwiz
Variations		
Issues		

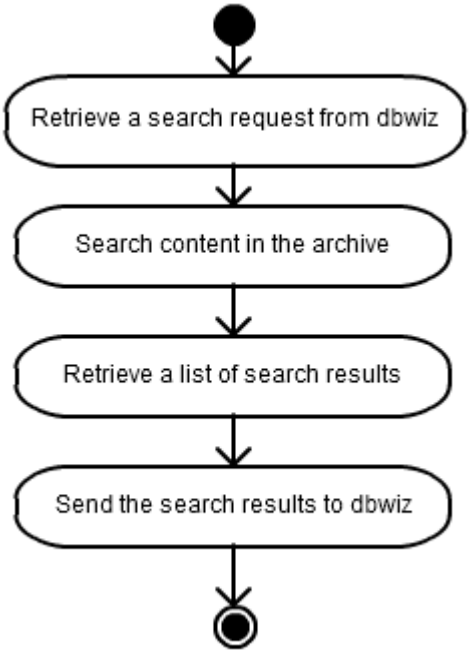


Figure 75: Activity Diagram for searching in dbwiz

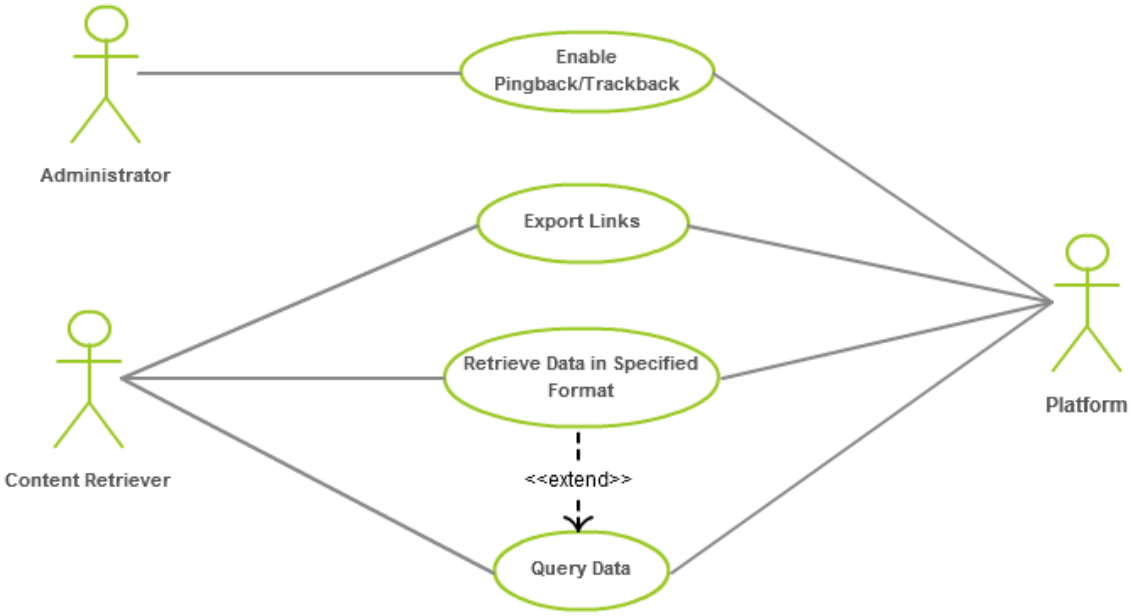


Figure 76: Use Case Diagram for the External Interface Requirements

EI1 – API for external clients to query data

Requirement category	External Interface Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	An application programming interface (API) for external clients enables the access on the archive and facilitates the development of third party applications based on the archive. Therefore, the API should provide the possibility to query and to retrieve data from the archive. The platform should be able to export data sets with special characteristics (large volume, machine readable, ready to use, with links between content – graphs) on demand.
Stakeholder	Businesses, Content Retriever Library, Researcher
Justification / Foundation	Businesses: "we provide an API so that you can query from the API both chart data and [...] data" Business: "Well, in the future I see the combining the data that we provide with web analytics data and perhaps even business intelligence data on a higher level but there is a need to combine different data sources" Researcher: I have analysed graphs coming from Facebook. I am looking into datasets from many sources, so I have also handled social graph datasets. The techniques for analysing graphs are common. Researcher: I use data sets created by others. I work on Information Retrieval that is very relevant to blogs, there is much information in blogs that I can extract and use in my research. Researcher: I would also be able to use such a tool to extract large amounts of data which would be real and ready to use. Then I could extract the links and generate the graphs for my research. There is so much interest in real graph data nowadays.
Assessment / Measures	An API is available that enables external clients to query and to retrieve data from the archive.
Author(s) of the requirement description	• Hendrik Kalb • Vangelis Banos

Use Case (EI) 1.1: Query Data

Use Case UCEI1.1	Query Data
Description	The Content Retriever can use an API to query data in the archive.
Actors	Platform, Content Retriever

Assumptions	**Preconditions**	1. The Content Retriever is registered.
	Postconditions	1. Query results are provided to Content Retriever.
Steps		1. The Content Retriever provides his/her account information to the API. 2. The Content Retriever enters keywords that s/he wants to query 3. The Content Retriever executes a method of the API 4. The Platform checks authentication. 5. The Platform executes the query retrieved from Content Retriever. 6. The Platform returns the query results to the Content Retriever
Parent		
Trigger		The Content Retriever executes a method of the API
Variations		
Issues		

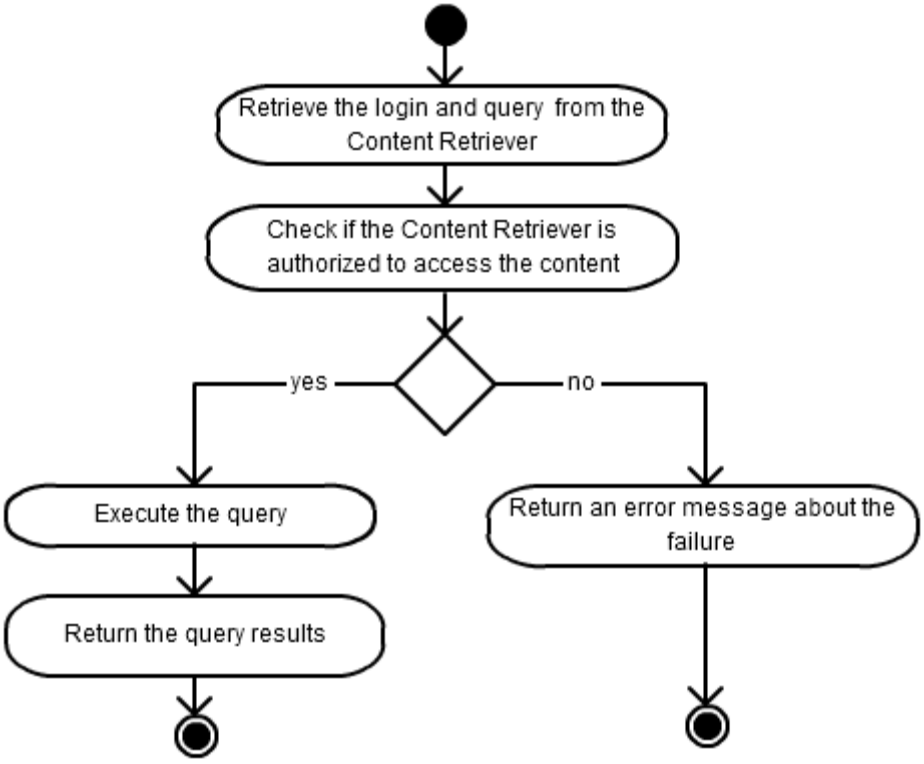


Figure 77: Activity Diagram for querying data using the API

EI2 – Data access/export as XML

Requirement category	External Interface Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	The archived data/contents should be available for external clients in XML format. This facilitates the processing of the data in external applications.
Stakeholder	Business, Admin Organisation
Justification / Foundation	Business: "Just generally, I think social media can be just through XML, I mean just getting all the documents at the same format [...], that's one way to look at it" Admin: "We do the same with our discussion forum system, where you can ask for the content of a particular forum to be given to you as an XML document of some sort that you can take and use elsewhere."
Assessment / Measures	The archived data/contents are accessible in a XML format.
Author(s) of the requirement description	• Hendrik Kalb

Use Case (EI) 2.1: Retrieve Data in Specified Format

Use Case UCEI2.1		Retrieve Data in Specified Format
Description		The Content Retriever can access/export the content in the format s/he specifies.
Actors		Platform & Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever is logged in.
	Postconditions	1. The Content Retriever accessed the content in the format s/he specified.
Steps		1. The Content Retriever enters the necessary information (login, keywords, etc.) 2. The Content Retriever specifies the content format (XML, CSV, etc.) 3. The Content Retriever executes a method of the API 4. The Platform checks authentication. 5. The Platform executes the query retrieved from Content Retriever. 6. The Platform returns the query result in the specified format
Parent		UCEI1.1
Trigger		The Content Retriever executes a method of the API
Variations		
Issues		

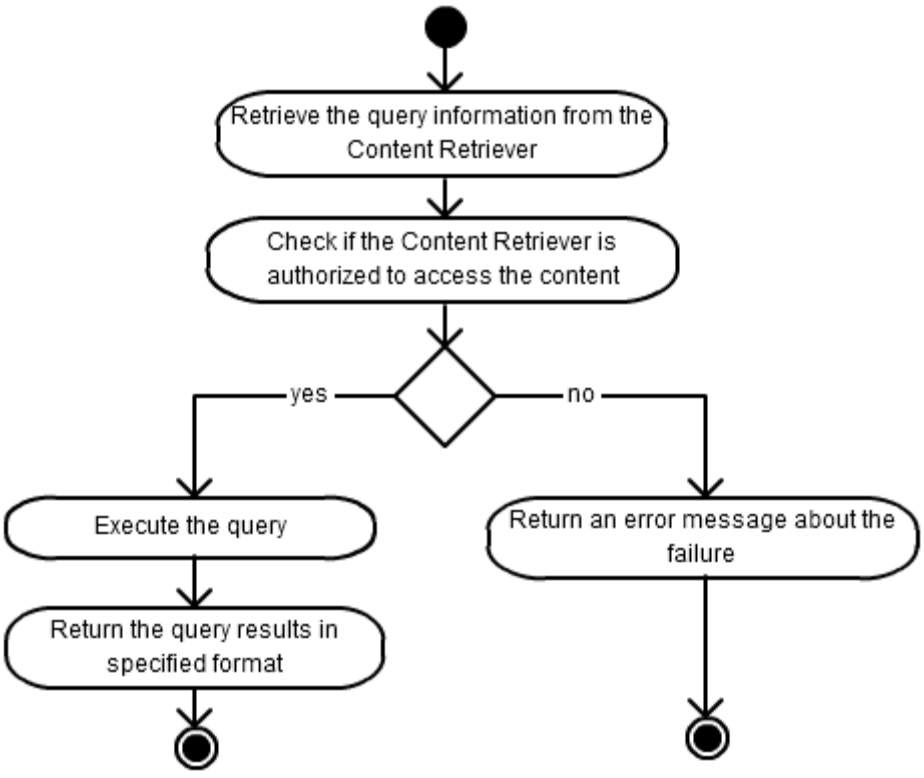


Figure 78: Activity diagram for retrieving data in specified data format

EI3 – Pingback, Trackback

Requirement category	External Interface Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	The archive should enable pingback / trackback to facilitate connections between the archived contents and external resources. Thereby, the archive can show if external blogs are referencing archived content.
Stakeholder	Admin Blog Host
Justification / Foundation	Admin: We use trackback, pingback, etc.
Assessment / Measures	Pingback/trackback is enabled for archived contents.
Author(s) of the requirement description	• Hendrik Kalb

Use Case (EI) 3.1: Enable Pingback/Trackback

Use Case UCEI3.1		Enable Pingback/Trackback
Description		Pingback/trackback is enabled for archived contents.
Actors		Platform, Administrator
Assumptions	**Preconditions**	1. Administrator has displayed the Admin Interface
	Postconditions	1. The Platform enables Pingback/Trackback
Steps		1. Administrator enables Pingback/Trackback so that the archive will show if external sources references the archived content
Parent		
Trigger		The Administrator's selection of choice
Variations		
Issues		

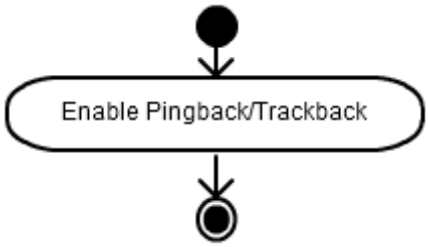


Figure 79: Activity diagram for enabling Pingback/Trackback

EI4 – Accessible via search machines

Requirement category	External Interface Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	The archive should be accessible with external search machines, e.g. Google.
Stakeholder	Content Retriever Library
Justification / Foundation	Library: The simplest way would be if we could manage that the repository could be retrieved via common search machines.
Assessment / Measures	The archive is accessible via search machines.
Author(s) of the requirement description	• Hendrik Kalb

EI5 – Export as CSV

Requirement category	External Interface Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	The archived content/data can be processed/analysed in external applications if they can be exported as comma-separated values (CSV, character-separated values).
Stakeholder	Researcher
Justification / Foundation	Researcher: "But I think also for quantitative analyses something like CSV-format so it could be imported to Excel or SPSS whatever would be useful."
Assessment / Measures	Archived content/data can be exported as CSV.
Author(s) of the requirement description	• Hendrik Kalb

Note: "Use Case (EI) 2.1: Retrieve Data in Specified Format" can apply here

EI6 - Export links between blog content

Requirement category	External Interface Requirement
Degree of necessity	( ) **Essential** (**X**) **Recommended** ( ) **Optional**
Description of the requirement	The BlogForever platform should be able to provide users with a graph of links between blogs & blog posts archived in the system. The format of this graph is not restricted.
Stakeholder	Researcher
Justification / Foundation	Interviewee says: I think it would be great to have on demand a graph of links between blogs & blog posts generated by the system. It would be very easy for me to get this and apply network algorithms, navigation, visualization, etc. I do not care about the protocol/format, anything would be fine. A simple text file with lists would be OK.
Assessment / Measures	A user can export a graph of links between blog content.
Author(s) of the requirement description	Vangelis Banos

Use Case (EI) 6.1: Export Links

Use Case UCEI6.1		Export Links
Description		The Content Retriever can export a graph of links between blog content.
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever has displayed “Network View” interface
	Postconditions	1. Graph of links has provided to the Content Retriever.
Steps		1. The Content Retriever selects blogs/blog posts to export links between them. 2. The Platform creates a graph of links between related blogs/blog posts. 3. The Platform provides this graph to the Content Retriever.
Parent		
Trigger		The Content Retriever clicks on a button/link.
Variations		
Issues		

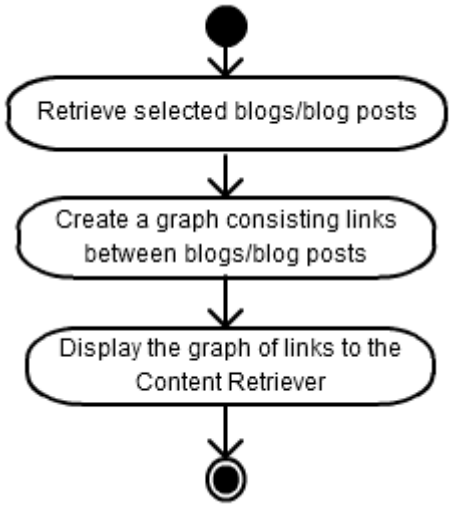


Figure 80: Activity Diagram for exporting links between blog content

4.4 (End-)User Interface Requirements& Usability

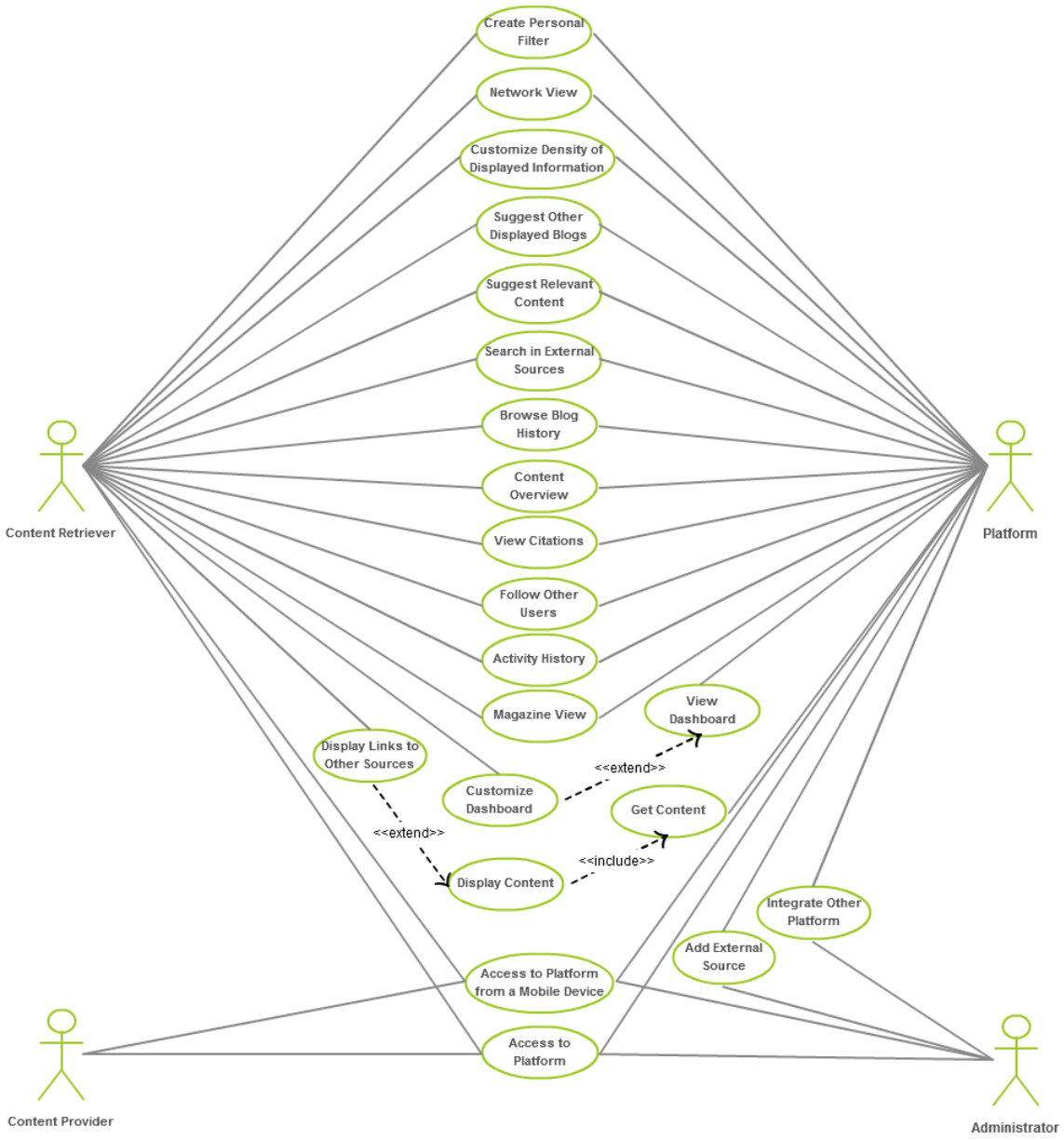


Figure 81: Use Case Diagram 1 for the (End)-User Interface Requirements

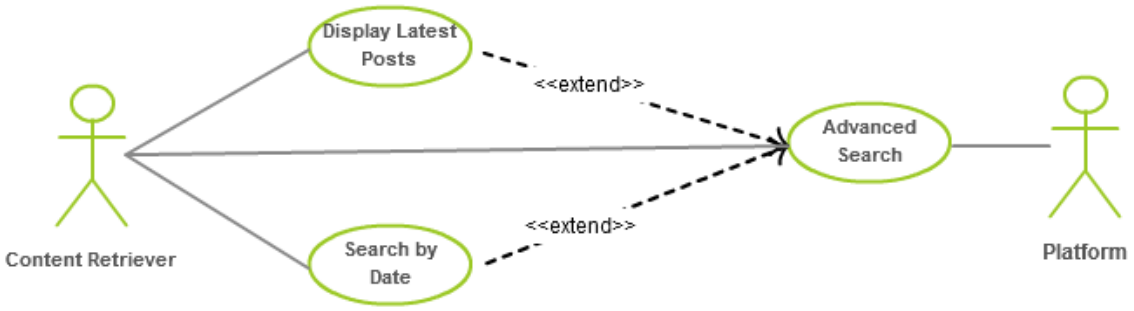


Figure 82: Use Case Diagram 2 for the (End)-User Interface Requirements

The following requirements are also user interface requirements but have already been defined above. Therefore, they are presented here but are not described again:

- FR5 – Descriptive statistics for a single blog or blog post
- FR6 – Processing of Licenses
- FR7 – User dissemination channels for blog post
- FR8 – Topics (Categories) for blogs and blog posts
- FR13 – Keyword/Metadata search
- FR16 – Search by author
- FR20 – List of favorite blogs and topics
- DR1 – Rights and Licenses
- DR4 – Display the author of the blog, blog post, comment

UI1 – Web Interface

Requirement category	User Interface Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	Users and administrators of the archive want to use the archive via web interface. The web interface should run in the most common browsers (Internet Explorer, Firefox, etc.) on the most common operating systems (Windows, Mac OS, Linux).
Stakeholder	Admin Organisation
Justification / Foundation	Admin Organisation: “It is mostly driven through a web interface; very occasionally someone will go and do something directly at the database level.”
Assessment / Measures	A web interface is provided for users and admins.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCUI(UI) 1.1: Access to Platform

Use Case UCUI1.1		Access to Platform
Description		Users and administrators can access the archive via web interface
Actors		Platform, Content Retriever, Content Provider, Administrator
Assumptions	**Preconditions**	1. User have Internet access
	Postconditions	1. User has accessed the Platform
Steps		1. The user (CR, CP or Admin) enters the URL of the Platform in a web browser 2. The user accesses to the Platform
Parent		
Trigger		Users provide the Platform's URL
Variations		
Issues		

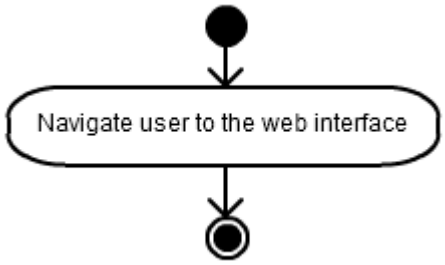


Figure 83: Activity Diagram for accessing to the Platform

UI2 – Magazine/Journal view

Requirement category	User Interface Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	The archive should provide a view that aggregates various current blog posts in way like magazines or journals do. This view can be presented on the archive or can be sent out to subscribers.
Stakeholder	Blog Author, Blog Reader
Justification / Foundation	Blog Author: "At the moment there are interesting tools around that people would have used to aggregate sort of like newspaper feed. I don't know if you have seen the sort of like 'paper' in various aggregators that produce, like an, magazine digest. I have noticed a few of my blog posts and Twitter posts (if you count Twitter as a blog in that refined area), they get picked up by these. And it is quite interesting to see how you then end up with your post being put alongside with other people's different posts. That is quite a visual way of doing, and then producing these magazine digest pages. So you end up with a set of boxes on that side saying: "this is certain people's blogs" and you get yourself a place alongside other people's postings." Blog Author: "It is just put together lots of blog post feeds and just produce a publication that then get circulated in a magazine style." Blog Author: "I think that is quite interesting potential to where that could go. Especially if you are thinking about an archive as well - an archive magazine!" Blog Reader: "Digests are quite useful in some of the work I am doing, rather than getting an individual notifications having a digest on say a weekly basis, if it is going to be an email update."
Assessment / Measures	Magazine or journal like view is provided.
Author(s) of the requirement description	• Hendrik Kalb

Note: "Use Case UCFR 12.1: Subscribe to Content" can apply here

Note: "Use Case UCFR 12.2: Notify by Email" can apply here

Use Case UCUI(UI) 2.1: Magazine View

Use Case UCUI2.1		Magazine View
Description		The Platform provides a view that aggregates various current blog posts in way like magazines do
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever is logged in
	Postconditions	1. The Platform has provided a number of blogs in a magazine like page
Steps		1. The Content Retriever clicks on “Magazine View” button/link/tab 2. The Platform retrieves a list of popular/top rated blog posts 3. The Platform constructs the magazine like page with the retrieved list of popular/top rated blog posts 4. The Platform displays the magazine like page
Parent		
Trigger		The Content Retriever clicks on a button/link/tab
Variations		
Issues		

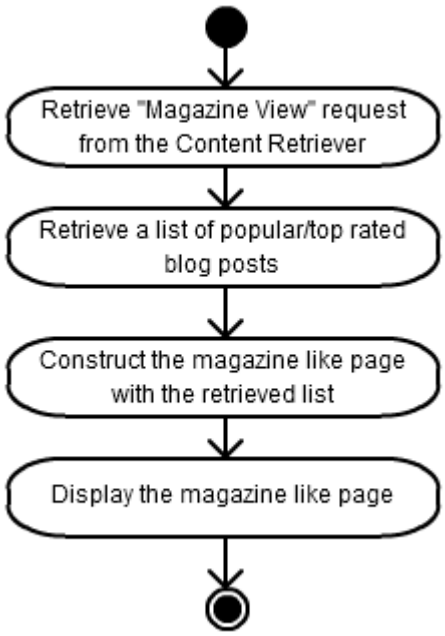


Figure 84: Activity Diagram for magazine view

UI3 – History of own activities in the archive

Requirement category	User Interface Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	A user should be able to search and browse which blogs he/she has annotated rated or other user activities.
Stakeholder	Blog Author
Justification / Foundation	Blog Author: “this is how you can go back and review what you have commented”
Assessment / Measures	A user can navigate in a history of his/her own activity.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCUI(UI) 3.1: Activity History

Use Case UCUI3.1		Activity History
Description		Platform users can search and browse which blogs he/she has annotated, rated or other user activities.
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever has displayed his/her profile page
	Postconditions	1. The Platform has displayed the user activity to the Content Retriever
Steps		1. The Content Retriever clicks on “Activity History” button/link/tab 2. The Platform retrieves the Content Retrievers activity history 3. The Platform displays the activity history to the Content Retriever
Parent		
Trigger		The Content Retriever clicks on a button/link/tab
Variations		
Issues		

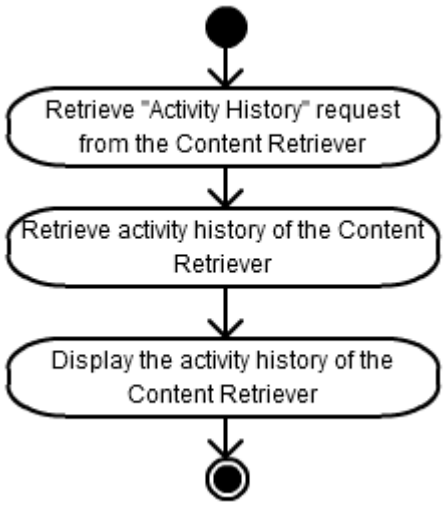


Figure 85: Activity Diagram for getting activity history

UI4 – Subscribe and Navigate activities of other users

Requirement category	User Interface Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	An archive user may want to see or follow the activities (e.g. annotations, highlighting, ratings, etc.) of other people (e.g. friends, colleagues, etc.). Therefore, it should be possible to subscribe and to navigate other archive users' activities.
Stakeholder	Blog Author
Justification / Foundation	Blog Author: "this is how you did, this is how you can go back and review what you have commented on or your colleagues have commented on."
Assessment / Measures	A user can follow and navigate other users' activities.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCUI(UI) 4.1: Follow Other Users

Use Case UCUI4.1		Follow Other Users
Description		A Platform user can follow the activities (e.g. annotations, highlighting, ratings, etc.) of other people (e.g. friends, colleagues, etc.).
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever has displayed another user's profile page
	Postconditions	1. The Content Retriever has subscribed to the other user's activities
Steps		1. The Content Retriever clicks on "Follow" button 2. The Platform adds the Content Retriever to the other user's subscription/followers list
Parent		
Trigger		The Content Retriever clicks on a button
Variations		
Issues		

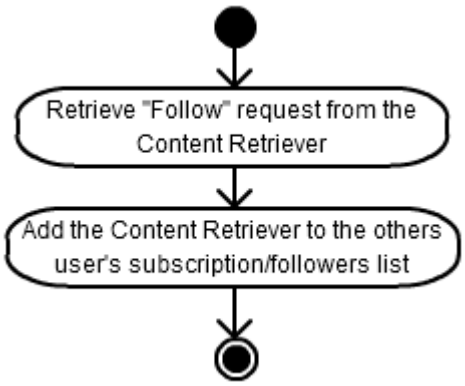


Figure 86: Activity Diagram for following other users

UI5 – Citation is presented prominently

Requirement category	User Interface Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	A user needs to link and cite the content of the archive. Therefore, the way how to link (URI) and how to cite should be presented prominently with the content. A citation includes at least author, title, date of creation, and URI.
Stakeholder	Blog Author
Justification / Foundation	Blog Author: "But I would want it to be cited properly. If people were going to reuse content. If it were exhibition quality that people would going to reuse it for other purposes that they could actually correctly cite it. So, I think that would be important for many of the blog users I know. But if it was going to be some useful reference elsewhere, and not just for your access for reminding you what you have posted, then correct citation would rank pretty highly."
Assessment / Measures	Citation is prominently presented with the content.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCUI(UI) 5.1: View Citations

Use Case UCUI5.1		View Citations
Description		The users of the platform can see the citations of a blog content
Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Content Retriever has displayed a blog
	Postconditions	1. The Platform has displayed citations of the blog
Steps		1. The Content Retriever clicks on “Citations“ tab/link 2. The Platform retrieves the citations of the blog 3. The Platform displays the citations of the blog to the Content Retriever
Parent		
Trigger		The Content Retriever clicks on a tab/link
Variations		
Issues		

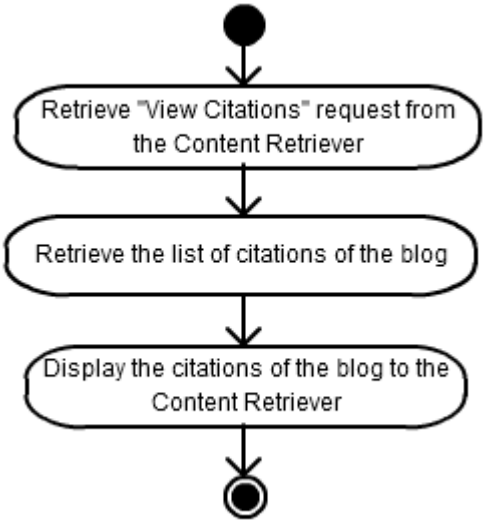


Figure 87: Activity Diagram for displaying citations of a blog

UI6 – Latest posts

Requirement category	User Interface Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	Often, users will be interested in what is new in the archive. Therefore, latest posts should be presented or highlighted.
Stakeholder	Blog Author, Researcher
Justification / Foundation	Blog Author: "you can choose which tags you want to highlight and latest posts." Researcher: "I also think maybe if I are using for information-purposes then some sort of thing like access to new or most used, or some sort of statistics I think might be helpful"
Assessment / Measures	Latest posts are presented to the archive user.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCUI(UI) 6.1: Display Latest Posts

Use Case UCUI6.1		Display Latest Posts
Description		The users of the platform can see the latest posts in the archive
Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Content Retriever has displayed the Advanced Search Interface
	Postconditions	1. The Platform has displayed the latest posts in the archive
Steps		1. The Content Retriever selects “Latest Posts“ option from the ranking methods list 2. The Content Retriever clicks on "Search" button. 3. The Platform retrieves the latest posts from the archive 4. The Platform displays the latest posts to the Content Retriever
Parent		UCFR13.1
Trigger		The Content Retriever clicks on a button
Variations		
Issues		

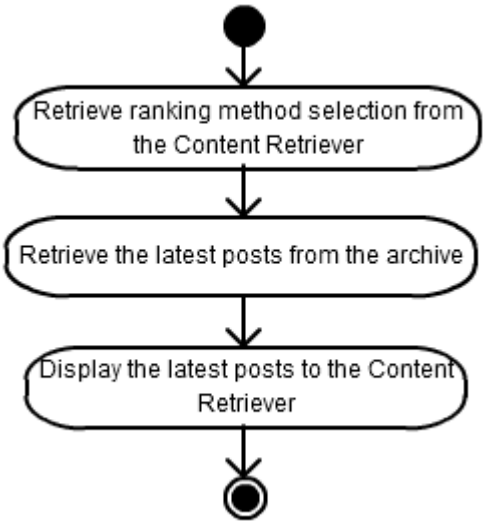


Figure 88: Activity Diagram for displaying latest posts to the Content Retriever

UI7 – Tags for blogs and blog posts

Requirement category	User Interface Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	Often, blogs and blog posts are related with tags that describe the topic or context. Therefore, tags should be presented similar for blogs and blog posts in the archive.
Stakeholder	Blog Author, Blog Reader, Researcher
Justification / Foundation	Blog Author: "you can choose which tags you want to highlight and latest posts." Blog Reader: "My archive in Delicious is all tagged. So, filter by tags." Blog Author: I imagine it like an aggregator that aggregates blogs with similar topics, the user to click a specific tag and to see the related topics.
Assessment / Measures	Tags are presented for blogs and blog posts.
Author(s) of the requirement description	• Hendrik Kalb • Stella Kopidaki

Note: "Use Case UCUI(UI) 10.1: Display Content" can apply here

UI8 – Overview with metadata and summary

Requirement category	User interface
Degree of necessity	( ) **Essential** (X) **Recommended** ( )**Optional**
Description of the requirement	Archive users need a meaningful overview over blogs and blog posts to find relevant content in the archive. This overview should show at least the title, author, amount of comments, and a brief summary of the content.
Stakeholder	Blog Reader, Researcher
Justification / Foundation	Blog Reader: "But I still filter out of it quite a lot (because of time), and just look at ones that have been commented on and ones that I recognise the author of in the first place (cause that is one of the ways that I define if I think they are going to be worth reading)" Blog Reader: "That would be the way of ranking some blogs, by the number of comments as opposed to just quality of the original posts." Researcher: "it would be interesting to have meta-date like attributes of the author, number of posts or frequency of posts, things like that. That's what I mean with meta-data and this is something I might like to analyze in an outside program - to see how many posts in average were done and in what frequency and which authors post what - that's sort of thing."
Assessment / Measures	Overview with title, author, amount of comments, and brief summary is available for blogs and blog posts.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCUI(UI) 8.1: Content Overview

Use Case UCUI8.1		Content Overview
Description		Archive users can see a overview of blogs and blog posts to find relevant content in the archive
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. Content Retriever has displayed a blog/blog post
	Postconditions	1. The Platform has displayed the overview of the blog/blog post
Steps		1. The Content Retriever clicks on a “Overview“ button/link/tab 2. The Platform retrieves the metadata (title, author, summary, etc.) of the blog/blog post 3. The Platform displays the overview of the blog
Parent		
Trigger		The Content Retriever clicks on button/link/tab
Variations		
Issues		

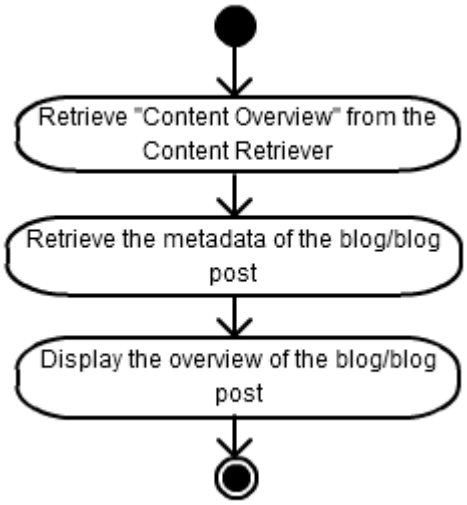


Figure 89: Activity Diagram for content overview

UI9 - Network view for topics, blogs, posts, authors, etc.

Requirement category	User Interface Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	A view on the relations among topics, blogs, posts, authors, etc. will enhance the possibility for users to explore the archived part of the blogosphere. Contrary to the view on single blogs or posts, it emphasizes the interconnections. The relations can be deduced e.g. from links between blogs. A visualization of author hierarchies and relations should be available to the end-user (including who mentions whom, etc.)
Stakeholder	Blog Reader, Researcher, Blog Author, Content Providers and (Higher Education) Blog Users
Justification / Foundation	Blog Reader: "They have done some work on linking various topics and producing a diagrammatic form of where different topics inter-relate and it produced a very interesting visual chart. So you had the possible inter-connections between different subjects, but it produced a visualisation of it. Where you get more complex and what it is you are looking for… I quite liked the way that gives you a visualisation as to how different topics could link together. So, you might not necessarily link the two in your head but the tool is helping you to sort of realise that there might be a different stage as to get from one subject area to another." Blog Reader: "As I said, I was quite excited by this visualisation tool that I saw earlier this year, because it seem to capture those people who aren't very good at looking at lists and picking things out. That's the group of people who would prefer to see things graphically presented." Researcher: "Blogs are connected into networks through the blog-rolls so if there was a tool which could also compile the connected blogs that would help identifying relevant blogs because connected blogs usually blog about the same subject or in case of science they are professionally connected, so that would have been very helpful." Researcher: "And since we are talking about networks, I guess it been able to export or create network-views of blogs would also be interesting for research. Blogs which are connected through links in a network-view or directed graph-view." Content Providers and (Higher Education) Blog Users: How interested are you in possible interconnections between your blog and others? In what way? How would you imagine a system supporting this feature? Very interested. The blog author explained the best of the blogs for history in Higher Education example as an example of

	interconnected blogs that they want to develop. They are very interested in external content alongside the Research Institute content. This will be very important for them. They will use some kind of aggregator that they would develop themselves if funding available. Showcasing the best of blogs (interesting and subjective). If there was an aggregator service (not just a list of feeds) they will definitely use it. For example best of monthly listings with nominations. Other institutions are using this but the Research Institute are not using this tool at the moment although they are very interested in this side of blogging interconnections. Graphical representation of all of the blogs together alongside each other. This overall view will be useful for the Research Institute with charts…but at the moment they have different platforms. Blog Author: "When you are talking about the inter-connections between the blogs, it is hard to describe how visually it should be represented, although, I have my own ideas on that, but it should be represented visually as a function of contribution and a function of time." Interview #1 ~7:40: "Trust is very important. For example in twitter, if you follow someone that you like and consider influential, you can see who they follow...There could be some hierarchies of trust...If an author mentions another author, that's the trust link for me..."
Assessment / Measures	Network views for topics, blogs, blog posts and authors are available. A user can view the original connections/aggregations of the blog selected. As well as that, users can interconnect blogs within a field within the archive in a simple and fast way.
Author(s) of the requirement description	• Hendrik Kalb • Silvia Arango-Docio • Jaime García • Nikolaos Kasioumis

Use Case UCUI(UI) 9.1: Network View

Use Case UCUI9.1		Network View
Description		Blogs, blog posts, topics, authors in the archive can be explored with network view which provides the users to analyze the interconnections among them
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever has clicked on "Network View" button/link/tab
	Postconditions	1. The Platform displays the interconnections among blogs, blog posts, topics or authors in a network view
Steps		1. The Content Retriever selects one of the network view types (blogs, blog post, topic or author) 2. The Platform retrieves a list of all the items of the selected network view type (blogs, blog posts, topics, authors) from the archive 3. The Platform constructs a page showing the interconnections between the items 4. The Platform displays the page to the Content Retriever
Parent		
Trigger		The Content Retriever clicks on a button/link/tab
Variations		
Issues		

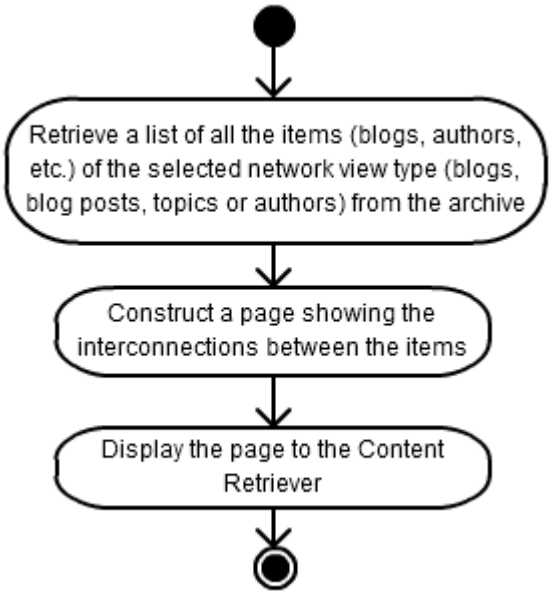


Figure 90: Activity Diagram for network view

UI10 – Available services depend on the content rights

Requirement category	User Interface Requirement, Security Requirements
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	Digital content could be available with different rights, e.g. if the user is allowed to print and to download. The user interface should support these rights by enabling/disabling the allowed services/functionalities.
Stakeholder	Content Retriever Library
Justification / Foundation	Library: Nowadays, it is state of the art that a library user can access everything in the library but the owner of rights can restrict the access to just reading or printing as well. Normally, downloading is not allowed. Library: Some authors put legal restrictions to the content they publish in our repositories. For instance a student can restrict public access to his thesis for a period of time (max 3 years). Some paperwork is required to do this. Another example, a Master thesis included military data which was confidential and should not be published, only kept in the repository.
Assessment / Measures	The user interface enables/disables access functionalities depending on the content rights.
Author(s) of the requirement description	• Hendrik Kalb • Vangelis Banos

Use Case UCUI(UI) 10.1: Display Content

Use Case UCUI10.1		Display Content
Description		The Platform enables/disables services/functionalities depending on the content rights of the users.
Actors		Platform, The Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever has sent a request to the Platform to display a blog/blog post
	Postconditions	1. Content is displayed with its enabled/disabled functionalities on it.
Steps		1. The Platform checks if the Content Retriever is authorized to access blog content 2. If authorized, the Platform retrieves all the blog content (blog itself, tags, comments and their sources, copyright, disclaimer, author, annotations, snapshots, etc.) 3. The Platform checks the Content Retriever's rights and constructs a page according to the them by enabling/disabling services/functionalities 4. The Platform displays the page to the Content Retriever 5. The Platform logs the Content Retriever information so that it can be tracked later 6. Otherwise, The Platform displays an "access not authorized" message
Parent		UCDR1.1
Trigger		The Content Retriever sends a request to the Platform to display a blog/blog post
Variations		
Issues		

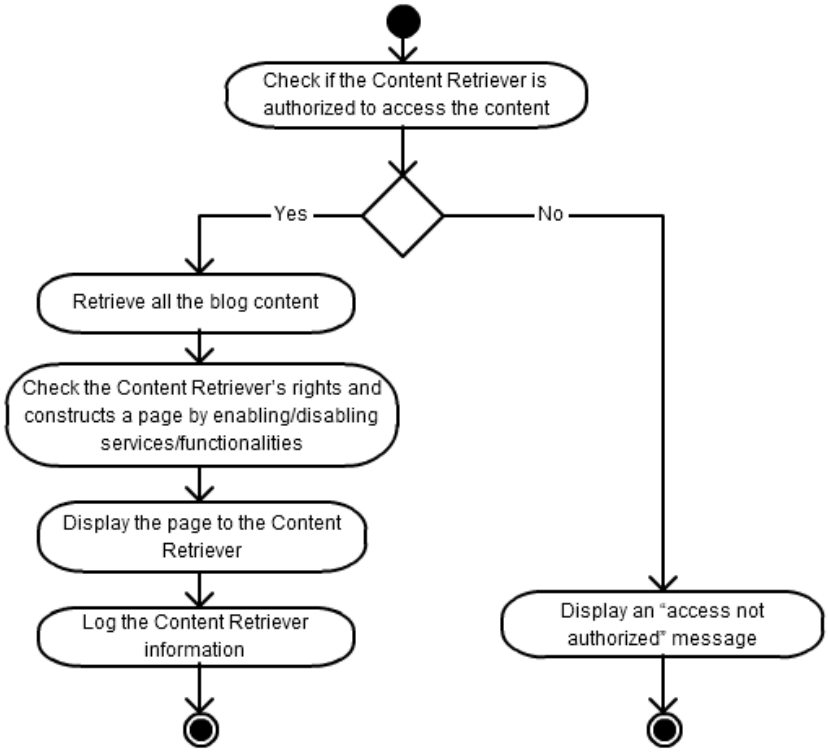


Figure 91: Activity diagram for displaying content

UI11 – Historical/Chronological view on a blog

Requirement category	User Interface Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	The user can view the blog at a particular date. One option would to navigate the blog on a time line. The user should be able to select a topic and view records (blog posts, comments, and blogs) associated with it, displayed in a chronological order. The user should be able to see what was published / discussed in the monitored blogs on a specific topic during selected time periods. The user interface should enable easy date-time period selection and browsing.
Stakeholder	Content Retriever Library, Researcher, Blog Reader, Content providers and blog archive users (content retrievers), Blog Author
Justification / Foundation	Library: A blog is archived with the date when the blog have been captured. Now, you can go to a particular date and see how the blog looked at this particular day and time. Like for a newspaper. Researcher: I would like to see how research/discussions on a topic have progressed during time. Blog Reader: I would like something like Google's search preferences that allow me to read topics that occur now or topics that created the last 24 hours or the last week. This is a typical way for me to choose and to know how fresh are the news. Content providers and blog archive users (content retrievers): If there was a real time, continuous and viewable archive of your blog, how would you use it? The answer to this question was: Look like a snapshot with a timeline approach, date will be a trigger. The snapshot will be created every time a new post goes life. If we are dealing with a very slow moving blog maybe once a week but if the level of material is high or changes often it would not be easy that way. 1.- Change of content 2.- Change of platform looks What would it look like? The answer for this second question is: They will use to provide information about the History of the Web. Changes of design, new features, how the platforms change. Content Provider Organisation: If there was a real time, continuous and viewable archive of your blog, how would you use it? What would it look like? She doesn't know, but snapshots with time available maybe. Blog Author: "So, I think there should be a time axis." Blog Author: "The Wayback Machine was really good for that, and it saved

	me a bunch of times. And what I really liked about it is - I didn't need to do anything. The problem with the Wayback Machine is that it wasn't complete. It would capture snapshots, but on a dynamic site like mine snapshots are hit and miss."
Assessment / Measures	The appearance of a blog (content, layout, etc.) can be viewed for any particular date. The indexing of the system will support date/time searching. The user interface will expose this functionality to users.
Author(s) of the requirement description	• Hendrik Kalb • Vangelis Banos • Stella Kopidaki • Silvia Arango-Docio

Use Case UCUI 11.1: Search by Date

Use Case UCUI11.1		Search by Date
Description		The results of a search can be filtered and/or sorted by date
Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Content Retriever views the Advanced Search Interface
	Postconditions	1. The Platform retrieves the search result and displays to the Content Retriever
Steps		1. The Content Retriever enters the search information through the Search Interface 2. The Content Retriever picks a start and an end date 3. The Platform executes the query retrieved from the Content Retriever 4. The Platform retrieves the search results (blogs and their usage statistics) which are in the specified date range 5. The Platform displays the search results
Parent		UCFR13.1
Trigger		The Content Retriever enters the keywords and presses a button
Variations		
Issues		

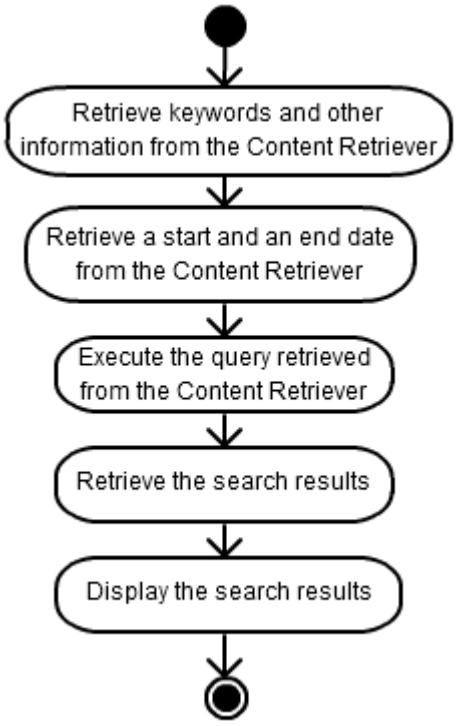


Figure 92: Activity Diagram for searching by date

Use Case UCUI 11.2: Browse Blog History

Use Case UCUI11.2		Browse Blog History
Description		The Content Retriever can view a blog at a particular date (old versions)
Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Content Retriever has displayed a blog
	Postconditions	1. The Platform has displayed the blog content and layout at the selected date
Steps		1. The Content Retrieves picks a date from a calendar like interface 2. The Platform retrieves the blog content, layout and statistics for the selected date 3. The Platform displays the blog content for the selected date
Parent		
Trigger		The Content Retriever clicks on a date
Variations		
Issues		

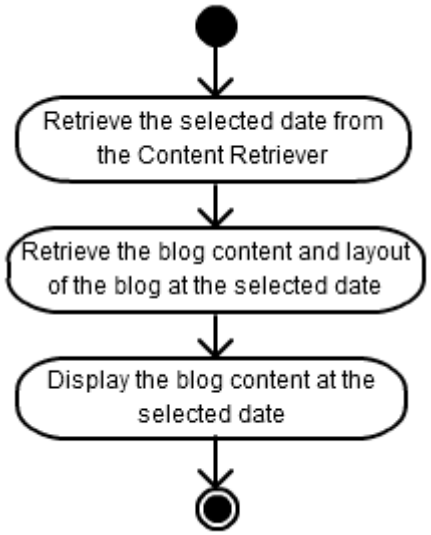


Figure 93: Activity Diagram for browsing blog history

UI12 – Annotations and Highlighting

Requirement category	User Interface Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	A user can directly annotate and highlight archived contents. The annotations and highlighting are displayed on the content, similar as the user would have done the annotations on a paper based version. Therefore, a layer for annotations (or something similar) is needed. This layer does not affect the content in the archive resp. database but displays the content with the annotations of the user.
Stakeholder	Researcher
Justification / Foundation	Researcher: "because working with the websites would be to actually be able to access the websites while they are online and put the codes in an extra layer - not sure if you are familiar with the diigo bookmarking platform - there you can add your notes and markings to websites but they are not actually added to website but they are saved in diigo as an extra layer and I think that would have been actually the solution which have been ideal so that you don't have to save the pages" Researcher: I use http://www.mendeley.com/ and I like it very much! [...] I can highlight a part of a document and save this as a note.
Assessment / Measures	Annotations and highlighting it possible directly on the content.
Author(s) of the requirement description	• Hendrik Kalb • Vangelis Banos

Note: "Use Case UCDR 10.1: Save User Annotations" can apply here

Note: "Use Case UCUI(UI) 10.1: Display Content" can apply here

UI13 – Customizable user dashboard

Requirement category	User interface requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	A user can customize his/her user dashboard with the favourite services, blogs, etc.
Stakeholder	Researcher, Businesses
Justification / Foundation	Researcher: "I guess it could be helpful to customize the entry, the dashboard or whatever to call it - because we all have different needs I think it would be useful if I could customize it" Businesses: "User should be able to customise his access to information services according to his own style and preferences" Businesses: "online user interface when you can log in and access the data, you have monitoring views, you have dashboards"
Assessment / Measures	The archive provides the user a possibility to customize a user dashboard easily.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCUI 13.1: View Dashboard

Use Case UCUI13.1		View Dashboard
Description		The Content Retriever can display his/her dashboard
Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Content Retriever is logged in
	Postconditions	1. The Content Retriever has displayed his/her dashboard
Steps		1. The Content Retrieves clicks on “Dashboard“ tab/link 2. The Platform display the Content Retriever’s dashboard
Parent		
Trigger		The Content Retriever a tab/link
Variations		
Issues		

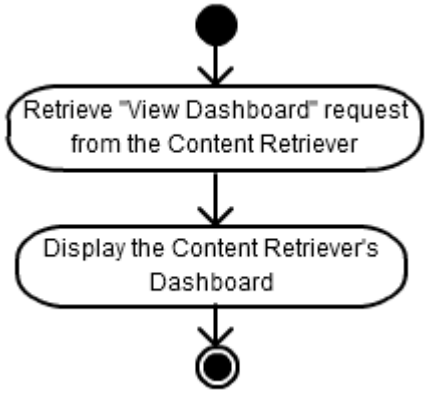


Figure 94: Activity Diagram for displaying the users’ dashboard

Use Case UCUI(UI) 13.2: Customize Dashboard

Use Case UCUI13.2		Customize Dashboard
Description		The Content Retriever can customize his/her dashboard
Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Content Retriever has displayed his/her dashboard
	Postconditions	1. The Content Retriever has displayed his/her dashboard
Steps		1. The Content Retriever clicks on "Customize" button/tab/link 2. The Platform displays the Content Retriever's current dashboard preferences 3. The Content Provider customizes his/her dashboard 4. The Platform applies the changes
Parent		UCUI13.1
Trigger		The Content Retriever a button/tab/link
Variations		
Issues		

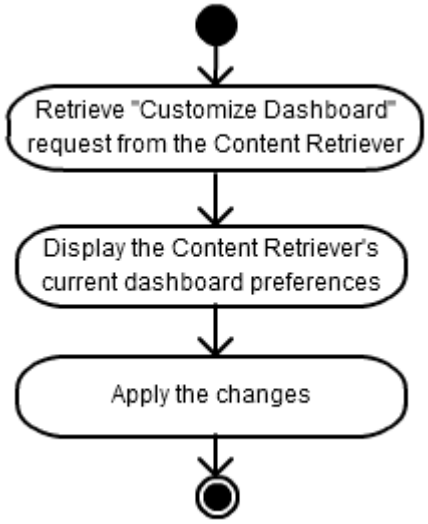


Figure 95: Activity Diagram for customizing dashboard

UI14 – User interface for mobiles

Requirement category	User Interface Requirement
Degree of necessity	( ) **Essential** ( ) **Recommended** (X) **Optional**
Description of the requirement	A user interface for mobiles, smartphones, and tablets has specific constraints, e.g. size of the display.
Stakeholder	Businesses
Justification / Foundation	Businesses: “In addition we need to present it as apps for mobiles and iPads
Assessment / Measures	A mobile version of the user interface is available.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCUI(UI) 14.1: Access to Platform from a Mobile Device

Use Case UCUI14.1		Access to Platform From a Mobile Device
Description		Users and administrators can access the archive via their mobile devices
Actors		Platform, Content Retriever, Content Provider, Administrator
Assumptions	**Preconditions**	1. User have Internet access on his/her mobile device
	Postconditions	1. User has accessed the Platform on his/her mobile device
Steps		1. The user (CR, CP or Admin) enters the URL of the Platform in a mobile browser 2. The user accesses to the Platform
Parent		
Trigger		The Content Retriever provides the Platform's URL
Variations		
Issues		

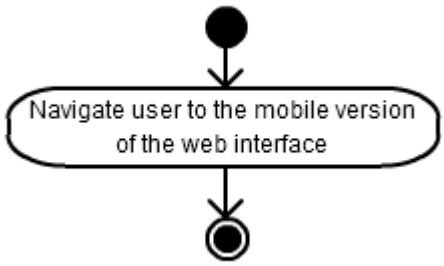


Figure 96: Activity Diagram for accessing to the archive from a mobile device

UI15 – Search interface

Requirement category	User Interface Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	The archive should provide a search interface that facilitates the different search functionalities described in the functional requirements.
Stakeholder	Researcher, Blog Author
Justification / Foundation	Researcher: "would be important to me as well as access to search-possibilities" Author: The user can search for new posts via a blogs search engine.
Assessment / Measures	Search interface is available for the implemented search functionalities.
Author(s) of the requirement description	• Hendrik Kalb • Stella Kopidaki

Note: "Use Case UCFR 13.1: Advanced Search" can apply here

Note: "Use Case UCFR 14.1: Search" can apply here

UI16 – Easy to learn/Intuitive

Requirement category	User Interface Requirement
Degree of necessity	(X) **Essential** ( )**Recommended** ( ) **Optional**
Description of the requirement	The user interface has to be easy to learn and to use. The archive software should be easy to use by all content providers within the Institute independently of their technical background. If any extra features are available (example: dissemination cases), they have to provide an extra benefit that it is clear to use, so there are no barriers to see how the dissemination channels within the software work.
Stakeholder	Businesses, Content Retrievers Libraries, Blog Author, Content providers: contributors and less technical staff
Justification / Foundation	Businesses: "So any software solution should require no training from the user at all" Businesses: "Any software should be extremely simple for use - I'd rather say ideal for use by naive users" Library: Question: What would be the important aspects that such software would have to fulfil? Interviewee: It should be easy to use. Blog Author: The information and the design of a page should help the user to find what he needs. If a page contains redundant elements, many advertisements or it is not well organised, then the user will not be able find what he needs. A page can be complicated, as a platform includes many components, but if it is structured in a way that the user can easily find them, and then the user will be able to find what he needs. Content providers: contributors and less technical staff: Do you have any general comments on the development of a blog aggregation, preservation, management and dissemination software? Easy to use (the most important), for the less technical people, extra features need to be an extra/clear benefit. Dissemination is hard: analysis, easy statistics, and easy to use for a multiple background team, no barriers to see how the channels of dissemination (routes) work. Content provider: What are important aspects that such software has to fulfil? The answer was: Easy to use, mainly content and comments plus the look and feel of the blog but she is not too bothered about this and she thinks we need to be realistic about how far back we should look into it.

	What would you describe as a "killer" application feature for a blog archiving service? Easy to use and that it should provide stats and popularity, comments and tweets feeds plus 'Like' icon usages as well. Social commentary around the blog should be available within this application. Do you have any general comments on the development of a blog aggregation, preservation, management and dissemination software? Easy to use + social media aspect of the blog.
Assessment / Measures	A normal user without previous experience with the archive can easily (e.g. without any help) process several given tasks. A user can adapt to the software changes independently of the users' technical knowledge.
Author(s) of the requirement description	• Hendrik Kalb • Senan Postaci • Stella Kopidaki • Silvia Arango-Docio

UI17 – Display references (links) to other sources inside or outside the archive

Requirement category	User Interface Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	Sometimes the content of a post is extracted from a trusted source, and a link to that source is included in the post. The platform should explicitly display these reference links.
Stakeholder	Content retriever
Justification / Foundation	Interview#2 ~8:00 "When somebody I knew had the right information, had a very good explanation, then I put a link to his blog because I knew it was accurate information, I put links to other material, be it a blog, be it a YouTube video, explaining something that I was talking about. I'm referring to other sources..."
Assessment / Measures	The user interface displays in a clear way the links used as references when showing a post.
Author(s) of the requirement description	• Jaime García • Nikolaos Kasioumis

Use Case UCUI 17.1: Display Links to Other Sources

Use Case UCUI17.1		Display Links to Other Sources
Description		The user interface displays in a clear way the links used as references when showing a post.
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. TheContentRetrieverhassentarequesttothe Platformtodisplayablog/blogpost
	Postconditions	1. The Platform has explicitly displayed the links to the other sources
Steps		1. The Platform retrieves all the blog content and links/references to other sources 2. The Platform explicitly displays the links/references to other sources in the blog post
Parent		UCUI10
Trigger		The Content Retriever has sent a request to the to display a blog/blogpost
Variations		
Issues		

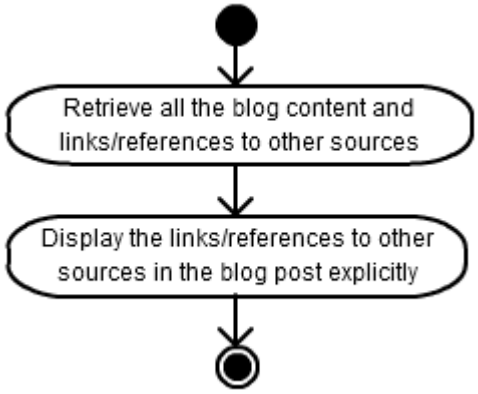


Figure 97: Activity Diagram for displaying links to other sources

UI18- Search in external sources

Requirement category	User Interface Requirement
Degree of necessity	() **Essential** ( ) **Recommended** (X) **Optional**
Description of the requirement	Search engine should optionally include external sources. The search results would include those external results, either in a separate tab or merged with the results found within the platform.
Stakeholder	Content retrievers
Justification / Foundation	Interview#2 ~10:10 "If your search engine offers you to look for blogs, for forums, for news articles, then it would be interesting, because you could see only blogs or what people are commenting..."
Assessment / Measures	There is an option in the search interface that enables to include external sources in the search results.
Author(s)of the requirement description	• Jaime García • Nikolaos Kasioumis

Use Case UCUI 18.1: Search in External Sources

Use Case UCUI18.1		Search in External Sources
Description		The Content Retriever can make a search in both archive and external sources via advanced search interface.
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever has displayed the Advanced Search interface
	Postconditions	1. Search results from external sources are displayed.
Steps		1. The Content Retriever enters keywords and other information (Boolean operators, date, sorting method, ranking, etc.…) 2. The Content Retriever selects one/more of listed external sources that s/he wants to search in. 3. The Platform retrieves the matching content from archive and external sources. 4. The Platform categorizes the results according to their sources 5. The Platform displays the search results
Parent		
Trigger		The Content Retriever includes external sources in search.
Variations		
Issues		

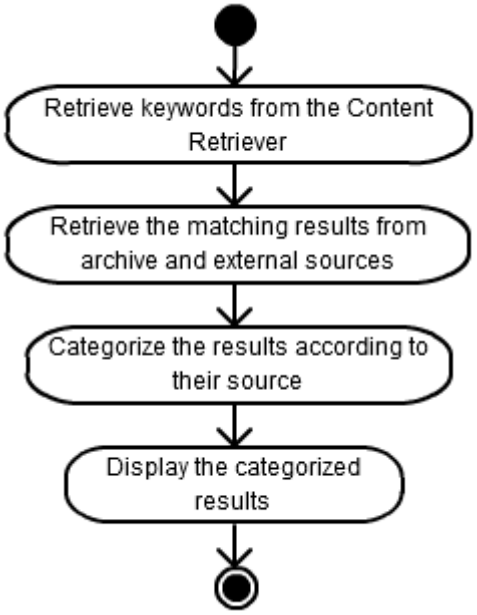


Figure 98: Activity Diagram for searching in external sources

Use Case UCUI 18.2: Add External Source

Use Case UCUI18.2		Add External Source
Description		The Administrator can add new external sources to external source list that the Platform can search in.
Actors		Platform, Administrator
Assumptions	**Preconditions**	1. The Administrator is logged in
	Postconditions	1. New external source is added.
Steps		1. The Administrator opens admin interface. 2. The Administrator adds external source information that the Platform can search in.
Parent		
Trigger		The Administrator clicks a button to add new external source.
Variations		
Issues		

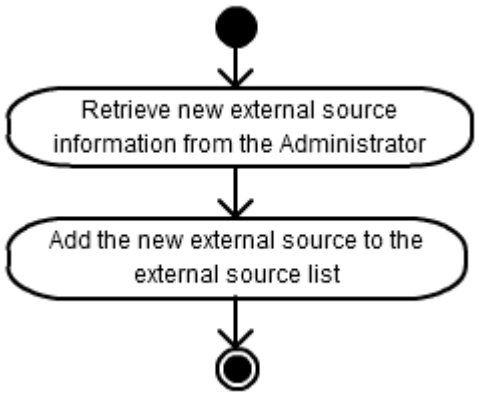


Figure 99: Activity Diagram for adding a new external source

UI19 – Display similar blogs and posts

Requirement category	User Interface Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	When displaying a post, links to other posts on the same subject should be displayed as suggestions. In a blogs page exists a component with similar blogs (blogs with analogous topics).
Stakeholder	Content retrievers, Blog Author/Reader
Justification / Foundation	Interview#1 ~6:48 "It'd be great to have: Show me all the blogs on this specific subject, in a single place. I think it'd be useful..." Blog Author/Reader: > I would be interested when I read a blog, to show similar blogs under it that will be generated with a specific way or blogs with analogous topics or people that read this blog, read and the second and some other analogous blogs. > I would like to be able to find topics and also to have suggestions for similar topics (related items). > I want to see related blogs when I choose a blog, I don't want to search in order to discover, but the system to suggest me things that maybe will be interesting for me based on my preferences and my actions, i.e. if I see a technological blog then to suggest me 2 or 3 other technological blogs, if I see a blog that is related to technology and music then to suggest me 2 other technological and 2 other music blogs. Blog Reader: I want to see a list with the top-5 or top-N blogs and to have information about the category they belong to. And after I want to see blogs related to the blog I have chosen, I don't want to search in order to discover, but the system to suggest me things that maybe will be interesting for me based on my preferences and my actions, i.e. if I see a technological blog then to suggest me 2 or 3 other technological blogs, if I see a blog that is related to technology and music then to suggest me 2 other technological and 2 other music blogs.
Assessment / Measures	Suggestions of similar blogs and posts are showed when displaying a blog or post.
Author(s) of the requirement	• Jaime García

description	• Nikolaos Kasioumis • Stella Kopidaki

Use Case UCUI 19.1: Suggest Relevant Content

Use Case UCUI19.1		Suggest Relevant Content
Description		Platform suggests posts/blogs relevant to the post/blog that the Content Retriever currently displays.
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever has displayed a blog content
	Postconditions	1. The Platform has suggested relevant content.
Steps		1. The Content Retriever clicks on “Similar Blogs/Blog Posts” button/link 2. The Platform retrieves a list of relevant blogs/blog posts by using a word similarity algorithm based upon the tagging information of the original blog/blog post 3. The Platform displays the list of relevant blogs/blog posts
Parent		
Trigger		The Content Retriever clicks on a button/link
Variations		
Issues		

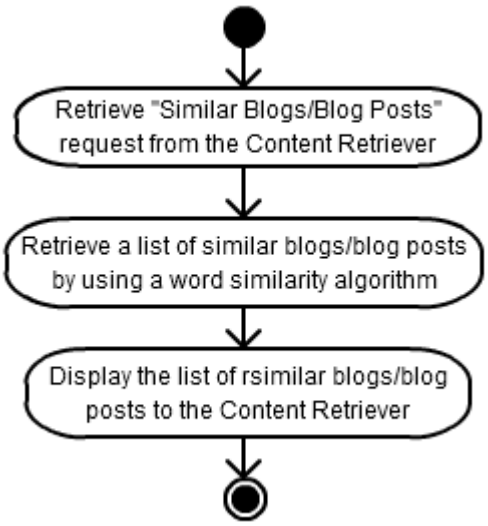


Figure 100: Activity Diagram for suggesting relevant content

UI20 – Display blogs that were read by people who have read a specific blog

Requirement category	Functional Requirements
Degree of necessity	( ) **Essential** (x) **Recommended** ( ) **Optional**
Description of the requirement	To have a component like Amazon that will inform the user that people who read this blog, also read blog X and blog Y.
Stakeholder	Content Provider – Blog Authors.
Justification / Foundation	Phrases from the interview • To show people that read this blog, and the second blog and third, read my blog and someone else's, like Amazon cross-referencing sales, so as to group blogs.
Assessment / Measures	For a displayed blog, blogs are shown that were read by people who have read the displayed blog.
Author(s) of the requirement description	• Stella Kopidaki

Use Case UCUI 20.1: Suggest Other Displayed Blogs

Use Case UCUI20.1		Suggest Other Displayed Blogs
Description		The Platform provides a list of blogs which were read by people who have read displayed blog
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever displays a blog
	Postconditions	1. The Platform suggests blogs that were read by readers of displayed blog
Steps		1. The Content Retriever clicks on “Usage Statistics” tab/link 2. The Platform retrieves the list of blogs that were read by other users who have already viewed currently displayed blog 3. The Platform displays the suggestions
Parent		
Trigger		The Content Retriever clicks on a tab/link
Variations		
Issues		

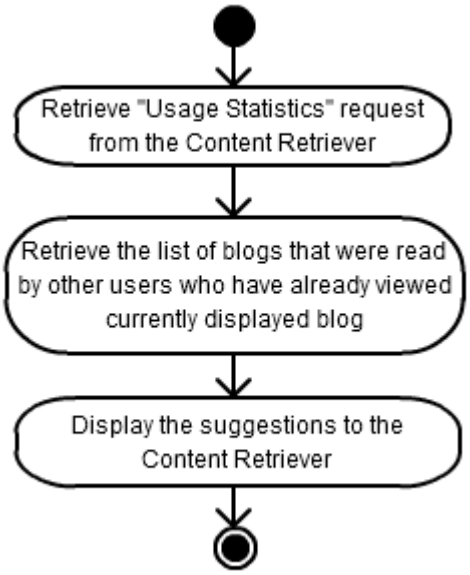


Figure 101: Activity Diagram for suggesting other displayed blogs

UI21 – Archived content is clearly stated as such

Requirement category	User Interface
Degree of necessity	( ) **Essential** (X)**Recommended** ( ) **Optional**
Description of the requirement	In order to avoid confusion and misunderstandings, it has to be clear to the end user that the content is an archived copy and not the original.
Stakeholder	Content retrievers
Justification / Foundation	Interview#1 ~21:05 "This potentially raises the question where the original content comes from, are they aware that this is an archive, an aggregator...it needs to be clear to the user that this is what it is..."
Assessment / Measures	The user interface states in a clear way that the content available is an archive of the original source.
Author(s) of the requirement description	• Jaime García • Nikolaos Kasioumis

UI23 – Categories/Topics are shown in different tabs

Requirement category	User Interface Requirement
Degree of necessity	( ) **Essential** (x) **Recommended** ( ) **Optional**
Description of the requirement	Use different tabs or links to show the different categories.
Stakeholder	Content Provider – Blog Author
Justification / Foundation	Phrases from the interview • Use of external links according to categories or different tabs that link to various categories based on the user interests can help the user to find what he wants.
Assessment / Measures	Different categories/topics are shown in different tabs or links.
Author(s) of the requirement description	• Stella Kopidaki

Note: “Use Case UCFR 13.1: Advanced Search” can apply here

UI24 – Display with only core information

Requirement category	User Interface Requirement
Degree of necessity	( ) **Essential** (x) **Recommended** ( ) **Optional**
Description of the requirement	like Instapaper: to see only the information itself Instapaper saves a clear text to the users account since the text parser transforms web pages into unstyled text versions suitable for mobile devices and e-book readers. Important information refers to the text, image or video that an article can contain. The user wants only the important information to be presented in a simple way with no redundant elements.
Stakeholder	Content retrievers - Blog Readers.
Justification / Foundation	Phrases from the interview • Do something like what Instapaper does, i.e. to extract only the information that is important, only the text, the image and the video, like reader of iOS Safari does now, and to aggregate all these information. So I will not see advertisements or anything else than the clear information.
Assessment / Measures	A user can see a blog post free of redundant elements like advertisements, so only the text and images/videos of it will be displayed.
Author(s) of the requirement description	• Stella Kopidaki

Note: “Use Case UCUI 22.1: Customize Density of Displayed Information” can apply here

UI25 – Filtered, personalized aggregation of content for end-users

Requirement category	User Interface
Degree of necessity	() **Essential** () **Recommended** (X) **Optional**
Description of the requirement	Users should be able to configure a personalized filter for the content they view in the platform, including old and new content.
Stakeholder	Content retrievers
Justification / Foundation	Interview#1~7:20 "I would like to see really specific topics and features that I can control. Otherwise it becomes too noisy, too messy..."
Assessment / Measures	A user can configure a filter to display his personal aggregation of blogs or posts.
Author(s) of the requirement description	• Jaime García • Nikolaos Kasioumis

Use Case UCUI 25.1: Create Personal Filter

Use Case UCUI25.1		Create Personal Filter
Description		Users can configure a filter to display his personal aggregation of blogs or posts
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever has displayed his/her profile page
	Postconditions	1. The Platform has created a personal filter for the Content Retriever
Steps		1. The Content Retriever defines a filter that specifies what kind of content filtered 2. The Content Retriever saves the changes
Parent		
Trigger		The Administrator clicks link/button to start integration.
Variations		
Issues		

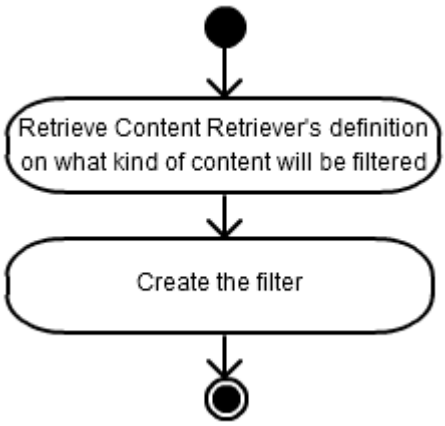


Figure 102: Activity Diagram for creating personalized filters

UI26 – Historical/Chronological view on blogs combined with corresponding statistics

Requirement category	User Interface
Degree of necessity	( ) **Essential** ( ) **Recommended** (X) **Optional**
Description of the requirement	The archive user will have snapshot data of changes of an archived blog in chronological order. These different versions of the archived blog will have specific analytical data attached to them to preserve the stats as well as the content, look and feel of the blog.
Stakeholder	Content providers and content retrievers
Justification / Foundation	Phrases from the interview: What are important aspects that such software has to fulfill? Design and changes plus comments and content with links and its corresponding analytics Stats disappear after a while in WordPress so more stable statistics will be very beneficial. What would you describe as a "killer" application feature for a blog archiving service? Image of the actual blog over time plus the analytics all on one screen: blog post per month... Viewable over time and for specific periods of time. Same screen for image of the blog (the look and feel) at the time and the analytics attached at that point, as much as possible and able to switch off the extra elements if needed.
Assessment / Measures	A user can find the statistics attached to different versions of a blog.
Author	Silvia Arango-Docio

Note: "Use Case UCUI 11.2: Browse Blog History" can apply here

UI27 – Dynamic network view on topics, blogs, posts, etc.

Requirement category	User Interface Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	A view on the relations among topics, blogs, posts, authors, etc. will enhance the possibility for users to explore the archived part of the blogosphere. Contrary to the view on single blogs or posts, it emphasizes the interconnections. The connections among blogs are often directed and change over time. Therefore, exploration will be further improved by a dynamic network view that displays the changes in the represented part of the blogosphere.
Stakeholder	Blog Author
Justification / Foundation	Blog Author: "When you are talking about the inter-connections between the blogs, it is hard to describe how visually it should be represented, although, I have my own ideas on that, but it should be represented visually as a function of contribution and a function of time. So, I think there should be a time axis. Typical representations of the linkages between blogs are never indexed to time. It is always this network and a network guy who says: "here's a blog, here's blog, and there is a line between them". But in real world, the relationships between blogs aren't like that. They are not static. It is not a one-time thing such as: "here is a network - forever and a day". It is very fluid, very dynamic. So, having a way of representing that would be important."
Assessment / Measures	Blogs and blog posts in the archive can be explored with a dynamic network view.
Author(s) of the requirement description	• Hendrik Kalb

Note: "Use Case UCUI 9.1: Network View" can apply here

UI28 – Integration/Combination with other systems

Requirement category	User Interface Requirement
Degree of necessity	( ) **Essential** ( ) **Recommended** (X) **Optional**
Description of the requirement	For people who analyse the data in relation to other data in a company, it would be beneficial to have an integrated or combined view on several systems (e.g. business intelligence, customer relationship management, etc.) instead of using each system separately.
Stakeholder	Businesses
Justification / Foundation	Businesses: "in the future I see the combining the data that we provide with web analytics data and perhaps even business intelligence data on a higher level but there is a need to combine different data sources and whether you do it locally so that you have your customer feedback function that combines customer feedback from different sources or if you have a performance data that combines for example sales and social media activity that's another story but the thing is that the data should be aggregated so that there is more data in one place so that companies' clients can make better decisions on it" Businesses: "so that instead that they have web analytics data in one place and marketing campaign data on activity clicks and displays, they have that in one place, and social media activity data in one place, they would have a common dashboard that would show all the three different data set and sales of course is the fourth"
Assessment / Measures	The user interface of the archive can easily integrated in other user interfaces or the user interface of the archive can easily integrate user interfaces of other systems.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCUI 28.1: Integrate Other Platform

Use Case UCUI28.1		Integrate Other Platform
Description		The Platform can easily be integrated in other platforms
Actors		Platform, Administrator
Assumptions	**Preconditions**	1. The Administrator has displayed the Admin Interface
	Postconditions	1. The Platform has been integrated in another platform
Steps		1. The Administrator enters the url of the other platform. 2. The Platform integrates itself with the other platform so that the data from the other platform can be retrieved within the Platform
Parent		
Trigger		The Administrator clicks link/button to start integration.
Variations		
Issues		

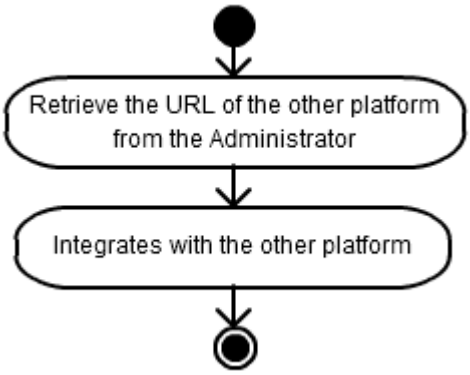


Figure 103: Activity Diagram for integration with other platforms

4.4.1 Ease of Use / Ease of Learning Requirements

UI22 –Density of displayed information

Requirement category	(End-)User Interface Requirements & Usability > Ease Of Use / Ease Of Learning Requirements
Degree of necessity	( ) **Essential** (x) **Recommended** ( ) **Optional**
Description of the requirement	The user is able to define the density of information e.g. if only text or images or video will appear
Stakeholder	Content retrievers - Blog Readers.
Justification / Foundation	Phrases from the interview • I want to be able to define how dense the content will be, e.g. if I want to see many photographs or not, if I want to see many videos related with the specific topic. But every time I will choose what to see based on my free time. • If I don't have time I want to see the titles and a part of the text, if I can spend more time then to see photos and video, if exist, for each post.
Assessment / Measures	The user can adjust the density of the displayed information.
Author(s) of the requirement description	• Stella Kopidaki

Use Case UCUI 22.1: Customize Density of Displayed Information

Use Case UCUI22.1		Customize Density of Displayed Information
Description		The Content Retriever can customize the density of displayed content (e.g. only videos or texts will be displayed).
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever is logged in.
	Postconditions	1. The changes that the Content Retriever made are saved by the Platform.
Steps		1. The Content Retriever displays his/her profile settings. 2. The Platform displays the Content Retriever's current preferences. 3. The Content Retriever selects/changes the content types (video, text, image...) that s/he wants to see. 4. The Platform saves and applies the changes
Parent		
Trigger		The Content Retriever clicks on a button to save the changes
Variations		
Issues		

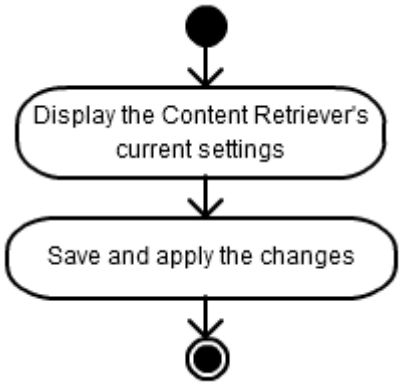


Figure 104: Activity Diagram for customizing density of displayed information

4.5 Performance Requirements

The performance requirements are further distinguished in speed/performance requirements, capacity and scalability requirements, and reliability and availability requirements.

4.5.1 Speed / Performance Requirements

This describes the required dynamic aspects, e.g. maximum amount of time for the response on a user request.

PR1 – Amount of blog posts to capture

Requirement category	Performance Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	The archive resp. the spider must be able to capture 500.000 to 1.000.000 blog posts per day. The number varies depending on the organization that will use the archive and the purpose of their archiving activities.
Stakeholder	Businesses, Admin Organisation
Justification / Foundation	Business: “the daily volumes are actually somewhere around, they might be even millions, or 1 point something millions per day blogs [blog posts], so the monthly numbers would probably be millions instead of hundreds of thousands” Business: “I think we are doing five hundred thousand a day, and that’s posts.” Admin: “ Business: “About 150,000 per day as input. However most of this is spam - so 25-30,000 is relevant.”
Assessment / Measures	The archive resp. the spider can capture up to 1.000.000 blog posts per day.
Author(s) of the requirement description	• Hendrik Kalb

PR2 – Storage data concurrently

Requirement category	Performance Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	Real time capturing can produce a lot of data in peak times. Therefore, it should be possible to store data concurrently in the archive.
Stakeholder	Business
Justification / Foundation	Business: "And then we need to be able to have the data logistics as smooth as possible because the data is being queried real time, it's not like a query run that's done during the night or anything but it's a real time query interface, so we need to concurrently store more data every second and be able every second to respond to queries; that at least creates challenges technically" Business: "Frequency from every 5th minutes to a couple of times per day"
Assessment / Measures	The archive can store data concurrently.
Author(s) of the requirement description	• Hendrik Kalb

4.5.2 Capacity and Scalability Requirements

CS1 – Amount of archived blogs

Requirement category	Capacity and Scalability Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	The platform has to be scalable in terms of the amount of archived blogs. The number of blogs can vary from some thousands of blogs up to one million or more. The archive has to be scalable to higher numbers of blogs as well.
Stakeholder	Admin Organisation, Admin Blog Host
Justification / Foundation	Admin Organisation: (Could you please estimate and describe how many blogs you are archiving / would consider archiving?) "I do not know of the top of my head. The system was running from 2004. I would estimate that there are around thousands of blogs, but not tens of thousands of blogs." Admin Blog Host: At the moment, we have almost one million of blogs. Nowadays, the number of our hosted blogs increases only slowly.
Assessment / Measures	The system is scalable in terms of the amount of archived blogs.
Author(s) of the requirement description	• Hendrik Kalb

CS2 – Amount of blog posts per day

Requirement category	Capacity and Scalability Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	The archive must be scalable regarding the content respective blog posts that are added daily.
Stakeholder	Businesses, Admin Organisation
Justification / Foundation	Businesses: "the daily volumes are actually somewhere around, they might be even millions, or '1 dot something' millions per day blogs [blog posts], so the monthly numbers would probably be millions instead of hundreds of thousands." (The answer relates to blog posts captured) Businesses: "I think we are doing five hundred thousand a day, and that's posts." (The answer relates to blog posts captured) Admin: "If we are talking specifically about blogs, than new blog entries are created on average at about 10 a day. It is more during the term time and fewer out of the term time. But we create at about 10 entries a day, so 70-80 entries a week." Business: "About 150,000 per day as input. However most of this is spam - so 25-30,000 is relevant."
Assessment / Measures	The archive is scalable regarding the content respective blog posts that are added daily.
Author(s) of the requirement description	• Hendrik Kalb

CS3 – Amount of users

Requirement category	Capacity and Scalability Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	The archive must be scalable in terms of archive users. Archive users can be blog authors as well as people who just read blogs or explore the archive.
Stakeholder	Admin Blog Host
Justification / Foundation	Blog Host: We have currently 700.000 Users.
Assessment / Measures	The archive is scalable in terms of archive users.
Author(s) of the requirement description	• Hendrik Kalb

4.5.3 Reliability and Availability Requirement

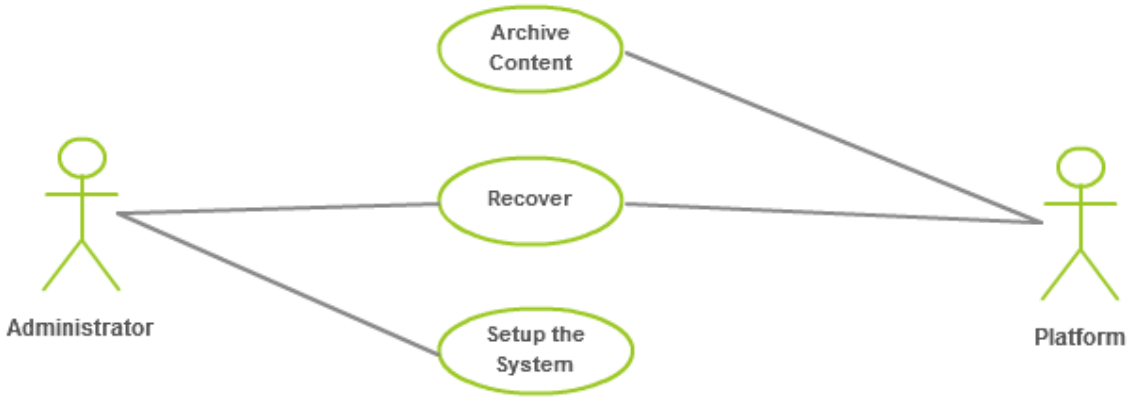


Figure 105: Use Case Diagram for the Reliability and Availability Requirements

RA1 – Recovery of the system

Requirement category	Reliability and Availability Requirement
Degree of necessity	( ) **Essential** ( ) **Recommended** (X) **Optional**
Description of the requirement	A productive use of the system would require a permanent availability. Therefore, it should be possible to recover the archive fast in the case of an unplanned close down of the archive.
Stakeholder	Admin Organisation
Justification / Foundation	Admin: "Blogs is fair to say a low priority service for us than our other services. So, if our main [...] website went down, and there was no page available when you went to [...] page, we would have got into a fairly serious emergency and people would start working on it immediately to get it back as quickly as possible. If blogs were similarly unavailable, we wouldn't have any particular commitment to say it must be back within an hour, a day or a week. We would work on it and bring it back, but there is no service agreement on that particular service."
Assessment / Measures	The system can be recovered in less than one hour.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCRA 1.1: Recover

Use Case UCRA1.1		Recover
Description		In case of an unplanned close down, the BlogForever Platform is able to recover the archive
Actors		Platform, Administrator
Assumptions	**Preconditions**	1. The administrator has displayed the Admin Interface
	Postconditions	1. Archive has been recovered.
Steps		1. The Platform displays the list of backup history 2. The Administrator selects one of the backup dates 3. The Platform starts the recovery process 4. The Platform restores the archive back to the state of the selected backup
Parent		
Trigger		The Administrator starts recovery.
Variations		
Issues		

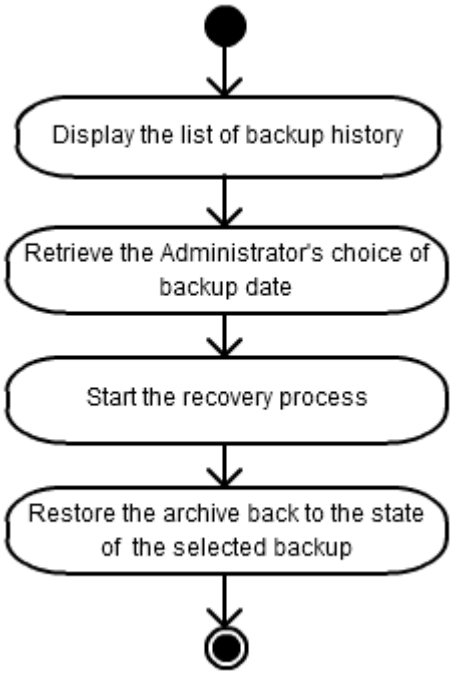


Figure 106: Activity diagram for recovery

RA2 – Correct information in the archive

Requirement category	Reliability and Availability Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	The archive has to store the information correctly as they can be found in the original blog.
Stakeholder	Blog Author
Justification / Foundation	Blog Author: "And you have to be very careful to making sure that you have got the right information in the archive"
Assessment / Measures	It has to be tested with random samples if the archive contains the information from blogs in the correct fields in the database.
Author(s) of the requirement description	• Hendrik Kalb

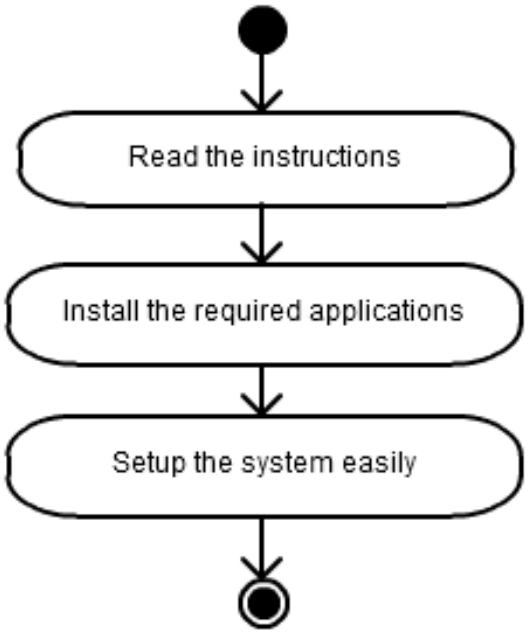
Note: "Use Case UCFR 6.1: Archive Content" can apply here

RA3 – Application High Availability

Requirement category	Reliability and Availability Requirement
Degree of necessity	( )**Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	The BlogForever platform should support high availability architectures.
Stakeholder	Admins - Organizations
Justification / Foundation	Interviewer: Our relationship with users is a traditional one and most of the time, our services are for free (we are a public institution). As a result, we do not have SLA with them; most of our users don't know what an SLA is. We would like to have 100% availability of course despite not having SLAs.
Assessment / Measures	The BlogForever Platform should be based on proven software which supports high availability architectures. (Web servers, application servers, database, and storage). There should be a clear guide to setup and use the BlogForever platform in high availability architectures.
Author(s) of the requirement description	Vangelis Banos

Use Case UCRA 3.1: Setup the System

Use Case UCRA3.1		Setup the System
Description		The Administrator can setup the system easily with clear guides
Actors		Administrator
Assumptions	**Preconditions**	1. The Administrator has downloaded the source codes
	Postconditions	1. The Administrator has setup the system and started to use
Steps		1. The Administrator reads the instructions 2. The Administrator installs the required applications 3. The Administrator setups the system easily
Parent		
Trigger		The Administrator starts the setup
Variations		
Issues		



Activity Diagram for building the system up

4.6 Operational Requirements

The operational requirements specify required capabilities of the system to ensure a stable operation of the system. They are further distinguished in supportability and maintainability requirements, and storage and persistence requirements.

4.6.1 Supportability and Maintainability Requirements

Figure 107: Use Case Diagram for the Supportability and Maintainability Requirements

SM1 – Migration/Updating without down time

Requirement category	Supportability and Maintainability Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	An update of the archive software should be possible without a shutdown of the archive.
Stakeholder	Admin Organisation
Justification / Foundation	Admin: "There are challenges, in particular, if you are looking to provide a very high level of continuity, for example. You want to go from the old system to the new system, but nobody wants it to be a period when there is no system at all. You have to try to look for a way to manage the migration in such a way that it is either continuous or it is possible to use the old system during the migration and then switch it to the new system immediately when it is finished."
Assessment / Measures	The archive software can be updated without shutdown of the archive.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCSM 1.1: Update Archive Software

Use Case UCSM1.1		Update Archive Software
Description		During an update or migration to new version, to provide continuity of platform, old version of platform will be in use.
Actors		Platform, Administrator
Assumptions	**Preconditions**	1. The Administrator has displayed the Admin Interface
	Post conditions	1. The software is updated.
Steps		1. The Administrator integrates the new updated component 2. The Administrator activates the new component
Parent		
Trigger		The Administrator activates the updated component
Variations		
Issues		

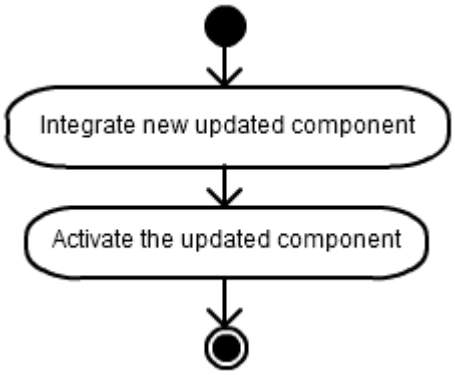


Figure 108: Activity Diagram for updating software

SM2 – Software updates

Requirement category	Supportability and Maintainability Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	It is important for IT-departments that the software runs without bugs and security gaps. Therefore, updates of the software are necessary to close possible security gaps and to repair bugs. It has to be clear to the IT departments how (e.g. how often, how get they informed) the software will be updated.
Stakeholder	Admin Blog Host
Justification / Foundation	Admin: We regularly update the software to close security gaps, to address user requests for features, and to improve services.
Assessment / Measures	An appropriate approach how to update the software is defined.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCSM 2.1: Check Software Updates

Use Case UCSM2.1		Check Software Updates
Description		The BlogForever Platform checks for software updates
Actors		Platform, Administrator
Assumptions	**Preconditions**	1. The Administrator is logged in.
	Post conditions	1. The software is updated
Steps		1. The Platform checks for the software updates 2. The Platform notifies Administrator about the software update 3. The Administrator starts the update process 4. UCSM1.1
Parent		
Trigger		The Administrator is logged in.
Variations		
Issues		

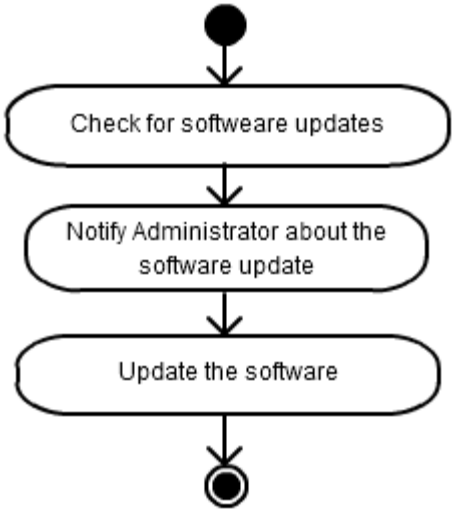


Figure 109: Activity Diagram for checking software updates

SM3 – Data export for migration

Requirement category	Supportability and Maintainability Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	It should be possible to export all the data from the archive to avoid a lock in effect to the archive software and to facilitate migration.
Stakeholder	Admin Blog Host
Justification / Foundation	Admin: If the software will not be developed further, we cannot work with the software anymore. In this case, we have to export the data and migrate it to another system.
Assessment / Measures	All data from the archive can be exported for migration.
Author(s) of the requirement description	• Hendrik Kalb

Note: “Use Case UCSP 2.2: Manual Backup” can apply here

Note: “Use Case UCRA 1.1: Recover” can apply here

SM4 – Compliance with nagios and cacti monitoring software

Requirement category	Supportability and Maintainability Requirement
Degree of necessity	(X) **Essential** () **Recommended** ( ) **Optional**
Description of the requirement	IT Infrastructure monitoring systems Nagios & Cacti must be compatible with the BlogForever Platform in order to enable instant awareness of problems and status.
Stakeholder	Admins - Organizations
Justification / Foundation	Interviewer: Which important functionalities do you normally use to monitor and maintain server software and services in your organization? Interviewee says: We are using nagios (http://www.nagios.org/**)** and cacti (http://www.cacti.net/). Cacti is used for system performance but we also use it to measure page response time for web apps.
Assessment / Measures	A Nagios system must be able to connect with the BlogForever platform and monitor performance and uptime. A Cacti system must be able to connect with the BlogForever platform and monitor performance and uptime.
Author(s) of the requirement description	Vangelis Banos

Use Case UCSM 4.1: Start Nagios and Cacti

Use Case UCSM4.1		Configure Nagios and Cacti
Description		The Administrator configures Nagios and Cacti.
Actors		Platform, Administrator
Assumptions	**Preconditions**	1. Nagios and Cacti are installed.
	Post conditions	1. The Nagios and Cacti start monitoring.
Steps		1. The Administrator installs Nagios and Cacti 2. The Administrator configures Nagios and Cacti by entering system related information (infrastructure of platform, alerting system…) 3. The Administrator starts Nagios and Cacti 4. If there is an error or important event, they send alert to the Administrators.
Parent		
Trigger		Nagios and Cacti are installed.
Variations		
Issues		

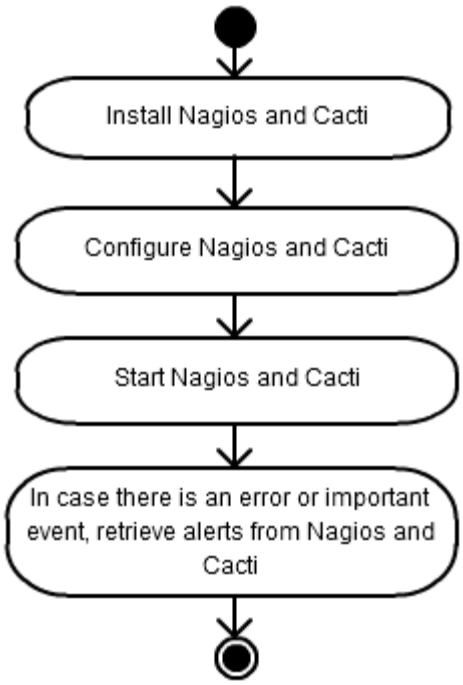


Figure 110: Activity Diagram for using Nagios and Cacti systems

4.6.2 Storage and Persistence Requirements

This describes specific requirements for the storage of the data, e.g. to prevent data loss.

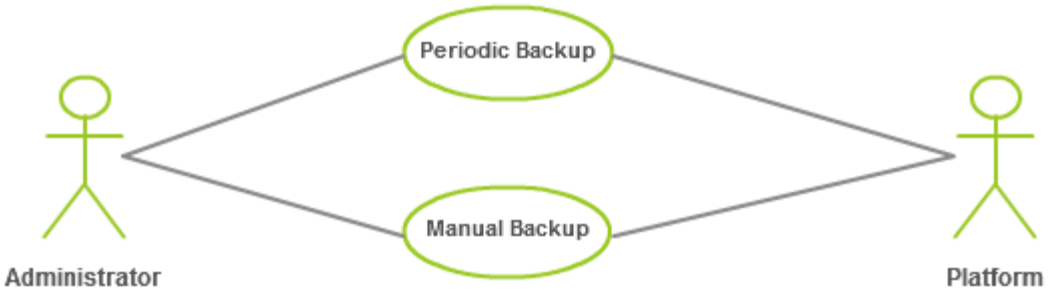


Figure 111: Use Case Diagram for the Storage and Persistence Requirements

SP1 – Support for different SQL-data bases

Requirement category	Storage and Persistence Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	The archive should support different SQL-databases to ease the administration for different institutions.
Stakeholder	Admin Organisation
Justification / Foundation	Admin: "We might care about what associated technology has come alongside. We might favour applications which use an Oracle database, rather than applications that use, say, SQL Server database."
Assessment / Measures	The archive supports/works on different SQL-data bases.
Author(s) of the requirement description	• Hendrik Kalb

SP2 – Mechanisms to avoid data loss

Requirement category	Storage and Persistence Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	Mechanisms or methods (e.g. data redundancy) have to be implemented to avoid data loss.
Stakeholder	Admin Blog Host
Justification / Foundation	Admin: The data has to be recoverable every time. Therefore, data have to be stored redundant.
Assessment / Measures	Mechanisms to avoid data loss are implemented.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCSP 2.1: Periodic Backup

Use Case UCSP2.1		Periodic Backup
Description		The BlogForever Platform can be scheduled to back up the archive
Actors		Platform, Administrator
Assumptions	**Preconditions**	1. The Administrator has displayed the Admin Interface
	Postconditions	1. The Platform has backed up the archive
Steps		1. The Administrator schedules the Platform to back up the archive periodically 2. The Platform checks for the backup time in certain time intervals 3. If it is time for the backup, the Platform backs up the archive 4. Otherwise it repeats the periodic check
Parent		
Trigger		The Administrator schedules backup.
Variations		
Issues		

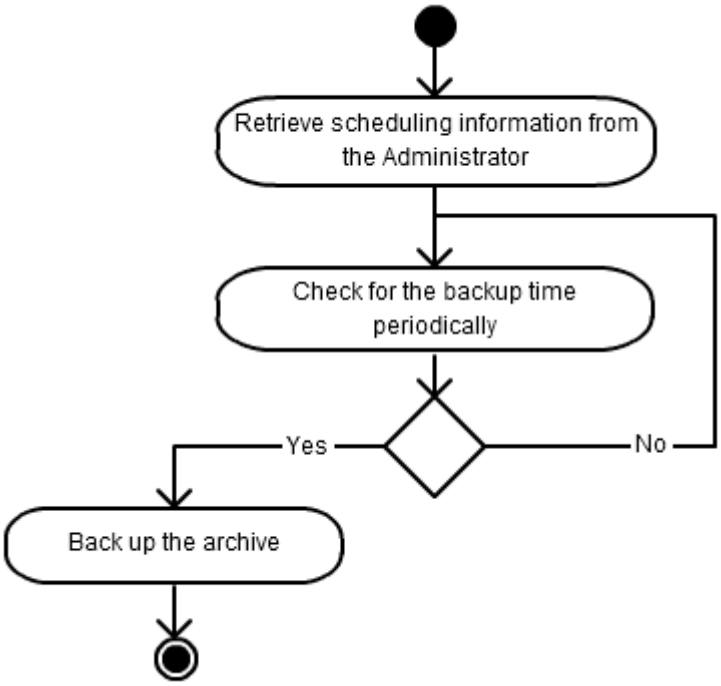


Figure 112: Activity diagram for scheduled backup

Use Case UCSP 2.2: Manual Backup

Use Case UCSP2.2		Manually Backup
Description		The administrator can start a backup manually.
Actors		Platform, Administrator
Assumptions	**Preconditions**	1. The Administrator has displayed the Admin Interface
	Postconditions	1. The Platform has backed up the archive
Steps		1. The Administrator starts the backup process 2. The Platform backs up the archive
Parent		
Trigger		The administrator starts backup process
Variations		
Issues		

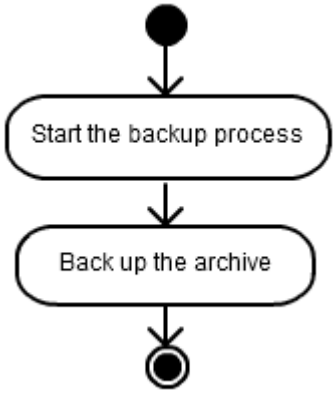


Figure 113: Activity diagram for manual backup

4.7 Security Requirements

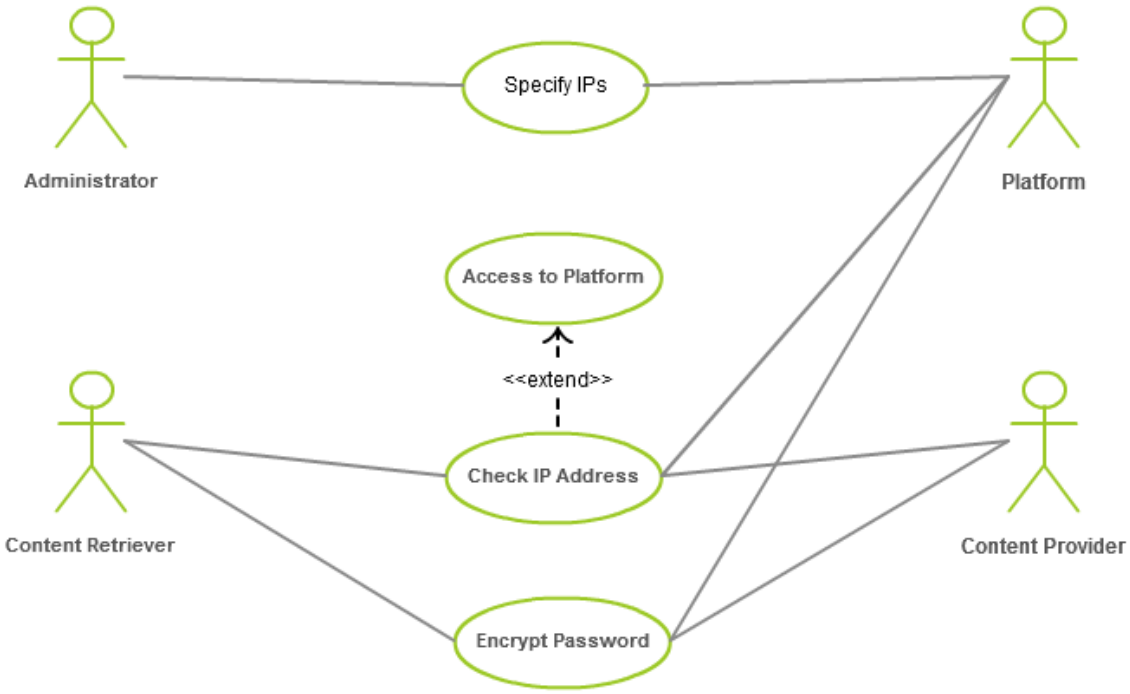


Figure 114: Use Case Diagram for the Security Requirements

The following requirements are also security requirements but have already been defined above. Therefore, they are presented here but are not described again:

- IR4 – Expose parts of the archive via OAI-PMH based on specified criteria
- UI10 – Available services depend on the content rights

SR1 – Passwords are stored encrypted

Requirement category	Security Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	Password has to be stored encrypted.
Stakeholder	Admin Blog Host
Justification / Foundation	Admin: Passwords are encrypted.
Assessment / Measures	Passwords are stored encrypted.
Author(s) of the requirement description	• Hendrik Kalb

Use Case UCSR 1.1: Encrypt Password

Use Case UCSR1.1		Encrypt Password
Description		The Platform encrypts users‘ passwords and store them encrypted
Actors		Platform, Content Retriever, Content Provider
Assumptions	**Preconditions**	1. A new user has provided registration information
	Postconditions	1 The Platform has stored the encrypted password.
Steps		1. The Platform retrieves the registration information from the new user 2. The Platform encrypts the password by using an encryption algorithm
Parent		
Trigger		The Content Retriever retrieves the registration information from the new user
Variations		
Issues		

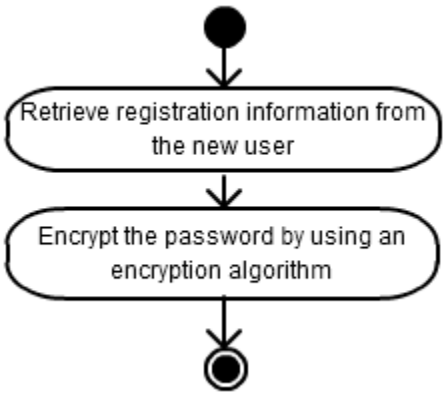


Figure 115: Activity diagram for encryption of password

SR2 – Access restricted to IP-range

Requirement category	Security Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	The BlogForever Platform can be accessible for specific IP addresses or an IP range and enables a security policy.
Stakeholder	Content Retrievers – Libraries, Content Providers/Retrievers
Justification / Foundation	Library: The repositories can be accessed anywhere in the METU campus. A person from outside the campus needs an account to access the repositories. Library: We restrict access based on IP (we provide access only to University network users)
Assessment / Measures	A security policy can be enabled for a BlogForever archive to specify certain IP addresses or an IP range, from which the archive is accessible. An administrator can specify a security policy and allow or deny access to the archive or a part of the archive (e.g. collection, blog) based on specific IP or IP range. Interview#1~3:00: “WehaveaninternalblogandweuseauthentiactiontohideitandIPrestrictions...”
Author(s) of the requirement description	• Senan Postaci • Vangelis Banos • Jaime García • Nikolaos Kasioumis

Use Case UCSR 2.1: Specify IPs

Use Case UCSR2.1		Specify IPs
Description		The Administrator can specify IP addresses or ranges from which users can access
Actors		Platform, Administrator
Assumptions	**Preconditions**	1. The Administrator has displayed the Admin Interface
	Postconditions	1. The Administrator has specified IP addresses or ranges
Steps		1. The Administrator enters IP addresses or ranges 2. The Platform stores the IP information
Parent		
Trigger		The Administrator has entered IP information
Variations		
Issues		

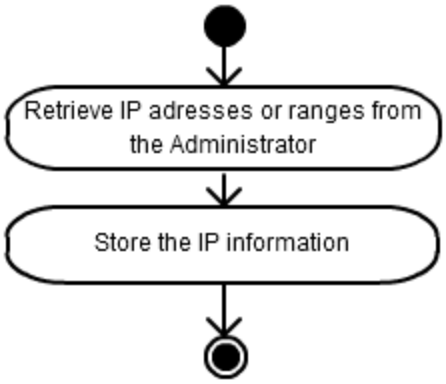


Figure 116: Activity Diagram for specifying IP addresses or ranges

Use Case UCSR 2.2: Check IP Address

Use Case UCSR2.2		Check IP Address
Description		The BlogForever Platform can check the IP addresses of the users since it can be accessible for specific IP addresses or an IP range
Actors		Platform, Content Retriever, Content Provider
Assumptions	**Preconditions**	1. User have Internet access
	Postconditions	1. User has accessed the Platform or navigated to an error page
Steps		1. The user (CR or CP) enters the URL of the Platform in a web browser 2. The Platform checks if the user's IP address is in the IP-range specified by the Administrator 3. If it is the main/welcome page is displayed to the user 4. Otherwise an error page is displayed to the user
Parent		UCUI1.1
Trigger		The Content Retriever provides the Platform's URL
Variations		
Issues		

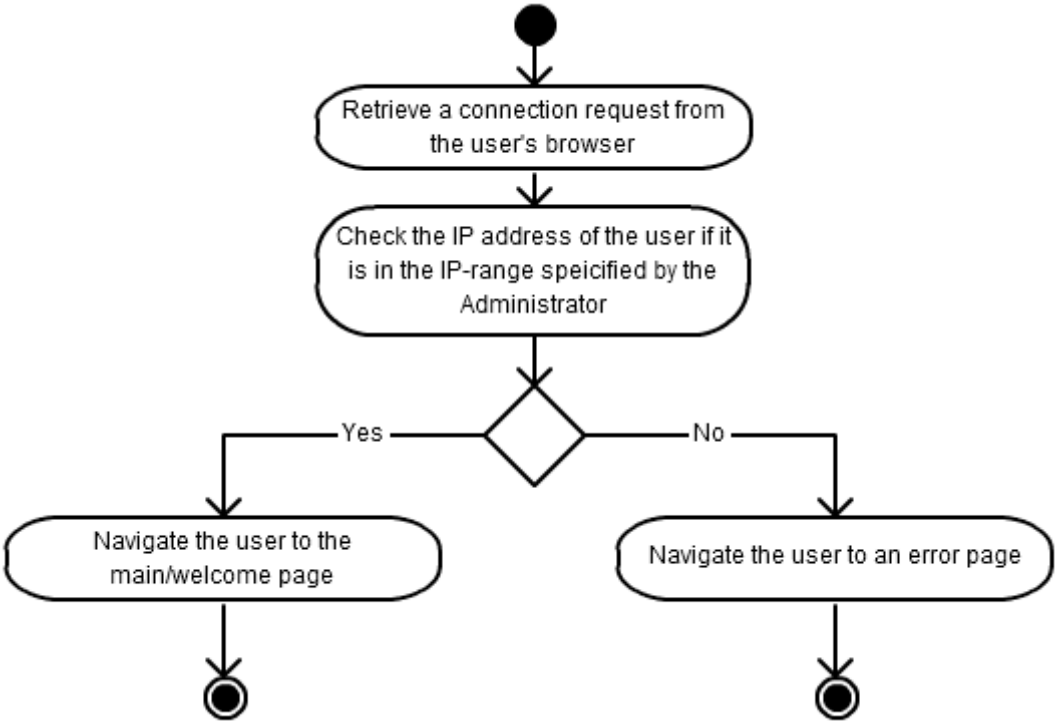


Figure 117: Activity Diagram for checking the IP addresses of the users

4.8 Legal Requirements

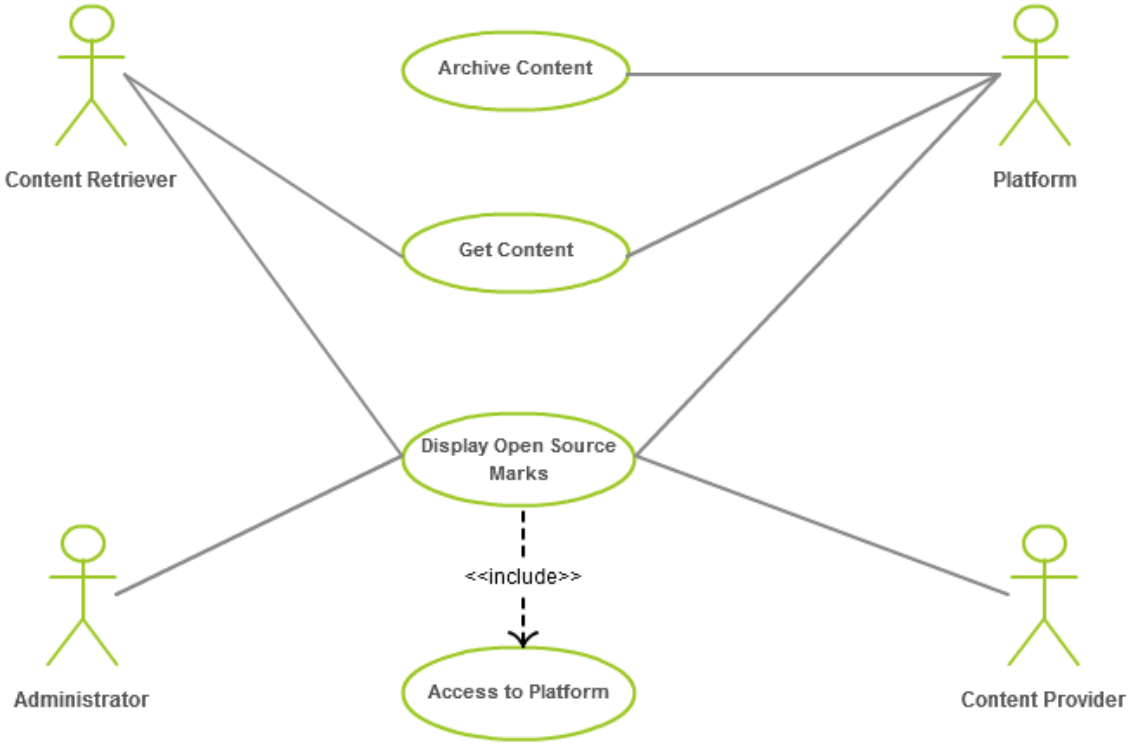


Figure 118: Use Case Diagram for the Legal Requirements

LR1 – Copyright laws

Requirement category	Legal Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	The archive has to follow copyright laws. Copyright laws exist on international as well as on national level.
Stakeholder	Libraries, Businesses
Justification / Foundation	Library: Copyrights give the author of digital content the right to decide what we are allowed to do with the content. Library: The copyright allows viewing the content when you are inside the library. Business: "As copyright, jurisdiction of copyright is usually, if I understand correctly, by country, of course there are directions globally, for example on European level, but basically as any legal structure."
Assessment / Measures	The archive follows international and national copyright laws.
Author(s) of the requirement description	• Hendrik Kalb

Note: "Use Case UCFR 6.1: Archive Content" can apply here

Note: "Use Case UCDR 1.1: Get Content" can apply here

LR2 – Privacy laws

Requirement category	Legal Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	The archive has to follow privacy and data protection laws. These laws exist on international as well as national level.
Stakeholder	Admin Blog Host, Library
Justification / Foundation	Admin: We have to follow the German Privacy Act. Library: We are very strict with respect to privacy rules. It is neither logged what users are reading nor what they are accessing. The data are not stored anymore if the user has not used the library for a given time. When validity of the library card ends then we have a given time and after that the privacy says that we are not allowed to store the data anymore.
Assessment / Measures	The archive follows privacy and data protection laws.
Author(s) of the requirement description	• Hendrik Kalb

Note: “Use Case UCFR 6.1: Archive Content” can apply here

Note: “Use Case UCDR 1.1: Get Content” can apply here

LR3 – Additional national laws

Requirement category	Legal Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	The archive must respect additional laws of organisations in the country in which the archive is running. For Germany, it would be e.g. the Tele-Media law.
Stakeholder	Library
Justification / Foundation	Library: The Tele-Media Law is relevant to us.
Assessment / Measures	An approach is defined how it can be ensured that relevant national rules are followed.
Author(s) of the requirement description	• Hendrik Kalb

Note: “Use Case UCFR 6.1: Archive Content” can apply here

Note: “Use Case UCDR 1.1: Get Content” can apply here

LR4 – License of the content

Requirement category	Legal Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	The archive has to respect the license under that the content is published. This could imply that e.g. the content is free for non-commercial purposes.
Stakeholder	Blog Author
Justification / Foundation	Blog Author: "I prefer something like Creative Commons, because it does give the ability to people to reuse without taking away the credit from the person who originally authored." Blog Author: Someone could use my texts. That does not matter to me. But I think it is mean if someone would take my texts and would sell them. Blog Author: "I use a Creative Commons, by-non-commercial share-like. It is supposed to be 3.0 [...]"
Assessment / Measures	The archive respects the license of contents that are archived.
Author(s) of the requirement description	• Hendrik Kalb

Note: "Use Case UCFR 6.1: Archive Content" can apply here

Note: "Use Case UCDR 1.1: Get Content" can apply here

LR5 – Open source software license is preferable

Requirement category	Legal Requirements
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	The BlogForever platform should be licensed under open source software licence.
Stakeholder	Admins – Organizations
Justification / Foundation	Interviewee says: We prefer open source licenses. We like to be able to anticipate costs and commercial software does not allow us to do this. It is easier to calculate how many man months we will need to customize software according to our needs. Interviewer says: Are you using Free Software (without source code)? Interviewee says: For specific utilities, we use free software. For mission-critical apps, we would not do it because it is depended on the community that supports it and this is not reliable.
Assessment / Measures	The BlogForever platform is published under an open source license. This is stated on the website and in the source code.
Author(s) of the requirement description	Vangelis Banos

Use Case UCLR 5.1: Display Open Source Marks

Use Case UCLR5.1		Display Open Source Marks
Description		The BlogForever platform is published under an open source license. This is stated on the website and in the source code.
Actors		Platform, Content Retriever, Content Provider, Administrator
Assumptions	**Preconditions**	1. User have Internet access
	Postconditions	1. User has accessed the Platform or navigated to an error page
Steps		1. UCUI1.1 2. The Platform displays the Open Source marks and signs
Parent		UCUI1.1
Trigger		The Users provide the Platform's URL
Variations		
Issues		

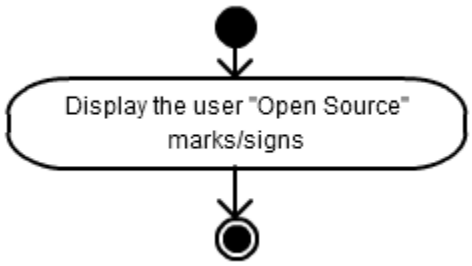


Figure 119: Activity Diagram for displaying open source marks/signs

5 Requirement Specifications from the DoW

In the following, the requirements are specified that were identified from the description of work. Thereby, the requirements are organised in subchapters that correspond to the requirement categories presented in chapter 1.3. UML models supplement the requirement descriptions. Requirements were modelled with activity diagrams if appropriate. The use case diagrams in the figures below give an overview on the modelled requirements including the responsible actors.

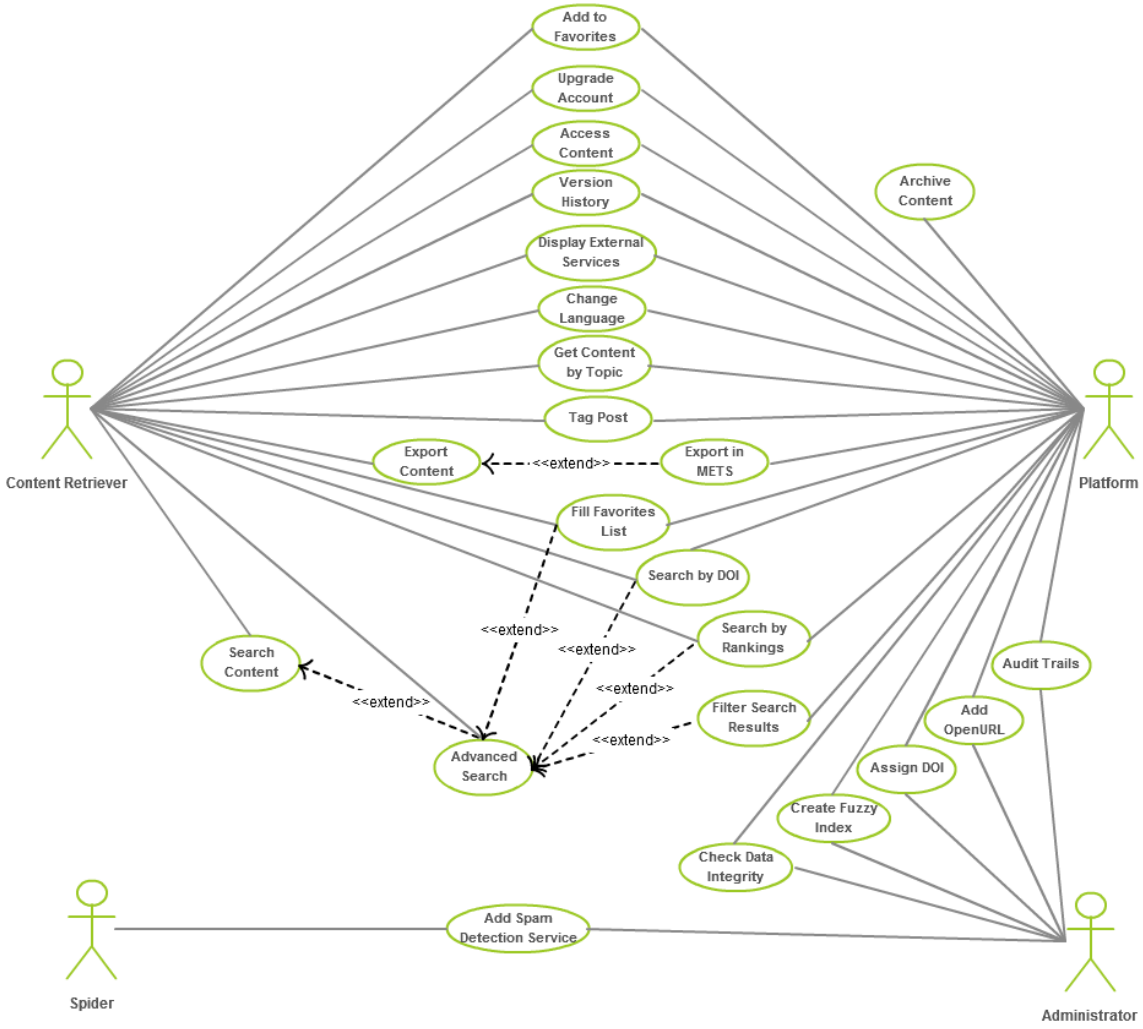


Figure 120: General Use Case Diagram 1

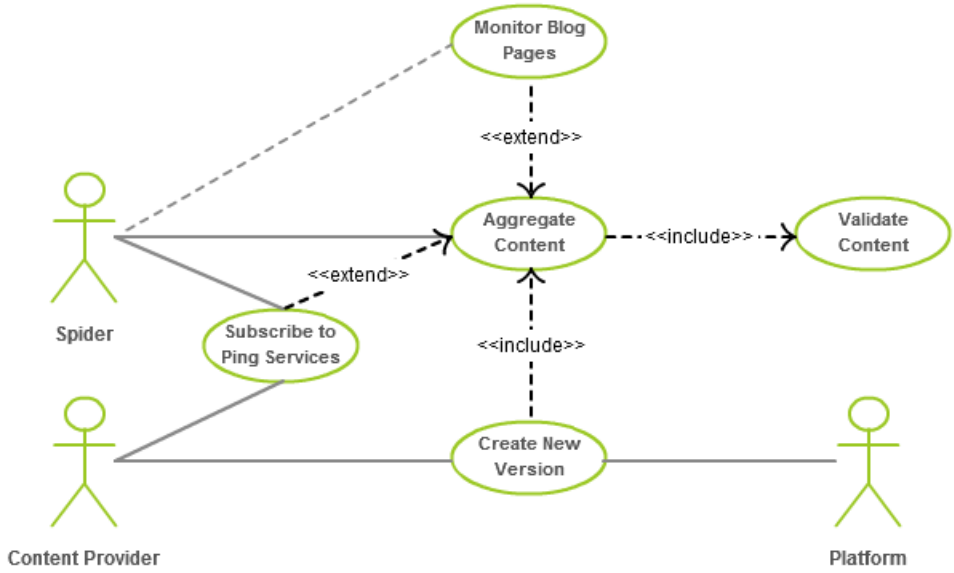


Figure 121: General Use Case Diagram 2

5.1 Functional Requirements

The data requirement "DR21 – Long term digital preservation" is also a functional requirement. It is described below in the next subchapter.

FR36 – Meme tracking and trend detection

Requirement category	Functional Requirement
Degree of necessity	(x) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	To answer the fragmented and mutually referring nature of weblog contents as persistent conversation (as a mixture of communication and content), new algorithms of prioritizing and recognizing importance of contents for information retrieval requests will also be developed using means of network analysis. Network based metrics will help to establish a ranking among weblogs. A way of computing this ranking is in a similar way to the PageRank algorithm used for web pages: the rank of a page is directly proportional to the number of incoming links to that page. This means that a page with many incoming links (corresponding to many other pages referring to and citing this page) is a highly ranked page. Similarly, the number of incoming inks can allow the computation of a ranking between weblogs. This is only one of the possible measurements of the importance of a Weblog. Other parameters, such as the topic, etc. will also be taken into consideration. Further means of Information Visualization will be developed to amplify cognition in the process of accessing the complex network structures in the archive efficiently. The preservation of weblogs has also to take into consideration the dynamic nature of contents evolving through networked discourses.
Stakeholder	Content retrievers
Justification / Foundation	Extracted from the Description of work.
Assessment / Measures	The final software supports meme tracking and trend detection functionalities.
Author(s) of the requirement	• Jaime García Llopis • Nikolaos Kasioumis

Use Case UCFR UC36.1: Search by Ranking

Use Case UC36.1		Search by Ranking
Description		The Content Retriever can order search results by page ranks
Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Content Retriever views the Advanced Search Interface
	Postconditions	1. The Platform retrieves the search result and displays to the Content Retriever
Steps		1. The Content Retriever enters the search information through the Search Interface 2. The Content Retriever checks the "Retrieve Pages by Rankings" option 3. The Platform saves the Content Retriever's search information 4. The Platform calculate the page rankings of the blog pages executes the query retrieved from the Content Retriever 5. The Platform retrieves the search results 6. The Platform displays the search results visually by using a node structure (with arrows directed to other blogs)
Parent		UCFR 44.1
Trigger		The Content Retriever enters the keywords and presses a "Search" button
Variations		
Issues		

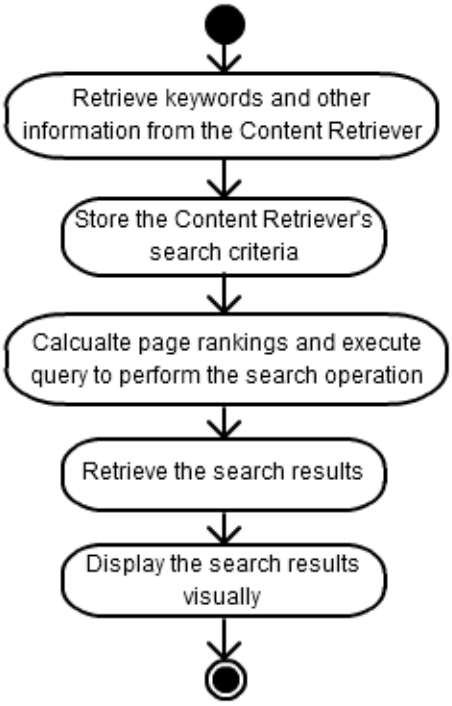


Figure 122: Activity Diagram for page ranking

FR37 – Web portal

Requirement category	Functional Requirement, Data Requirement, (End)-User Interface Requirement
Degree of necessity	(x) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	The digital repository application must facilitate the creation of a web portal which will function as a point of access to information and as an interface to the Web Services created.
Stakeholder	Content retrievers, content providers, administrators
Justification / Foundation	Extracted from the Description of work.
Assessment / Measures	
Author(s) of the requirement description	• Jaime García Llopis • Nikolaos Kasioumis

Use Case UCFR UC37.1: Search Content

Use Case UC37.1		Search Content
Description		The Content Retriever can search blog content in the Platform.
Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Content Retriever is logged in
	Postconditions	1. The Platform retrieves the search result and displays to the Content Retriever
Steps		1. The Content Retriever enters the search keywords 2. The Platform executes the query retrieved from the Content Retriever 3. The Platform retrieves the matching and similar records 4. The Platform displays the search results
Parent		
Trigger		The Content Retriever enters the keywords and presses a “Search” button
Variations		
Issues		

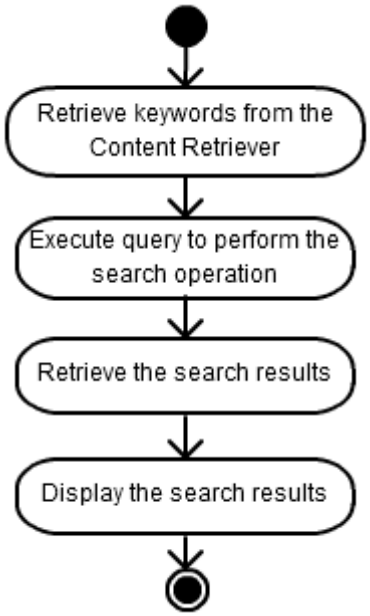


Figure 123: Activity Diagram for searching content in the Platform

FR38 – Multidimensional indexing

Requirement category	Functional Requirement
Degree of necessity	(x) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	Multidimensional indexing / fuzzy indexing: A key feature in the successful usage of a system like the one here is a simple yet powerful search methodology. On the one hand, this requires queries supported by standard SQL commands for meta-tags and simple word based queries for the content. However, in many cases users will only have a rough idea about what they are searching for. In this case methods using fuzzy search on the content (segments) will be needed. Fuzzy search on huge databases (several millions of segments) will require optimized search trees and indexing methods. One candidate for this are KD-Trees which are heavily used in text retrieval to find similar segments. To support normalized queries on terms the index should be built on lemmatizes or stemmed terms too.
Stakeholder	Content retrievers, Administrators
Justification / Foundation	Extracted from the Description of work.
Assessment / Measures	The system supports standard queries and fuzzy search. Users could find what they are looking for even if they type an approximate string.
Author(s) of the requirement description	• Jaime García Llopis • Nikolaos Kasioumis

Use Case UCFR 38.1: Create Fuzzy Index

Use Case UC38.1		Create Fuzzy Index
Description		The Administrator can create fuzzy indexes in the archive
Actors		Platform, Administrator
Assumptions	**Preconditions**	1. The Administrator has viewed the Admin Interface related to indexes
	Postconditions	1. The Administrator has created a fuzzy index
Steps		1. The Administrator specifies a name for the index 2. The Administrator selects a field to be indexed 3. The Platform creates a fuzzy index on the field
Parent		
Trigger		The views the Admin Interface
Variations		
Issues		

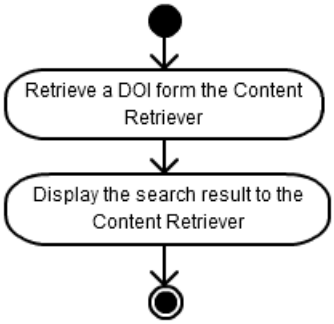


Figure 124: Activity Diagram for creating a fuzzy index

FR39 – Digital Rights Management

Requirement category	Functional Requirement
Degree of necessity	(x) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	A robust Digital Rights Management (DRM) system which involves the description, layering, analysis, valuation, trading and monitoring of the rights over the repository's assets. Also, support for Open Digital Rights Language (ODRL), which is supported by the W3C should be enabled.
Stakeholder	Content providers, content retrievers
Justification / Foundation	Extracted from the Description of work.
Assessment / Measures	The platform supports a fully functional DRM system
Author(s) of the requirement description	• Jaime García Llopis • Nikolaos Kasioumis

Use Case UCFR 39.1: Access Content

Use Case UC39.1		Access Content
Description		The Content Retriever can access blog content if s/he is authorized, otherwise cannot
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever clicks on a blog page to retrieve its contents
	Postconditions	1. The Content Retriever accesses blog content or not
Steps		1. The Platform checks if the Content Retriever is authorized to access blog content 2. If yes, the Platform retrieves the blog content and displays 3. Otherwise, The Platform displays an error message about the failure
Parent		
Trigger		The Content Retriever clicks on a link to the blog post
Variations		
Issues		

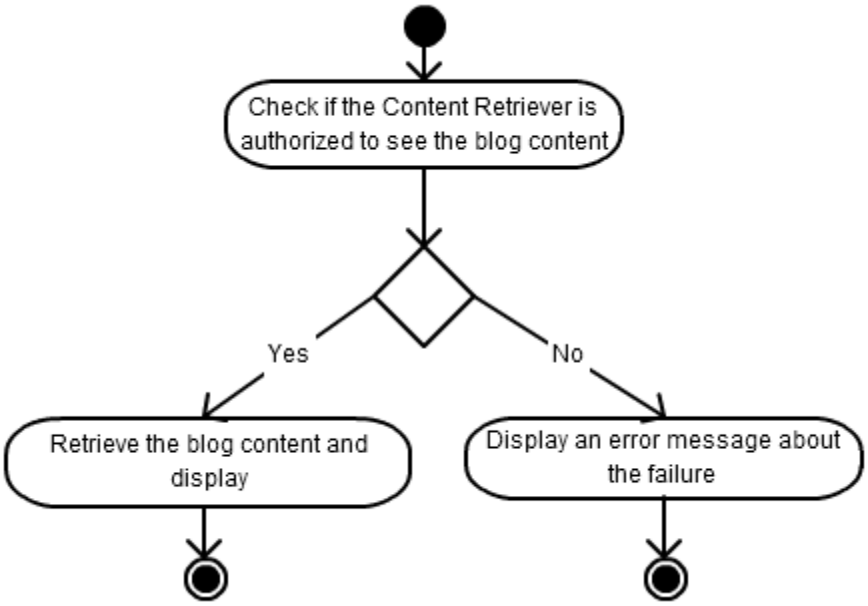


Figure 125: Activity Diagram for accessing content

FR40 – Billing system

Requirement category	Functional Requirement
Degree of necessity	(x) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	Billing system will be introduced into the code base. This will allow the system's administrators to exploit added value services. For instance, disk space in the digital repository could be free until users reach a specific quota. Additional disk space could be available for a monthly fee.
Stakeholder	Administrators, content retrievers, content providers
Justification / Foundation	Extracted from the Description of work.
Assessment / Measures	The platform provides a billing system service
Author(s) of the requirement description	• Jaime García Llopis • Nikolaos Kasioumis

Use Case UCFR 40.1: Upgrade Account

Use Case UC40.1		Upgrade Account
Description		The users of the BlogForever platform can upgrade their accounts to get additional services.
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever views the "Pricing" interface
	Postconditions	1. The Content Retriever's account has been upgraded and Content Retriever can use additional services.
Steps		1. The Content Retriever enters necessary information (paying method such as credit card or Paypal and etc.…) to upgrade his/her account 2. The Platform retrieves the information and processes it. 3. If the retrieved information is correct (valid credit number / paypal account etc.…) the Platform sets an upgraded account policy for the Content Retriever 4. Otherwise, The Platform displays an error message
Parent		
Trigger		The Content Retriever enters necessary information and clicks "Upgrade" button
Variations		
Issues		

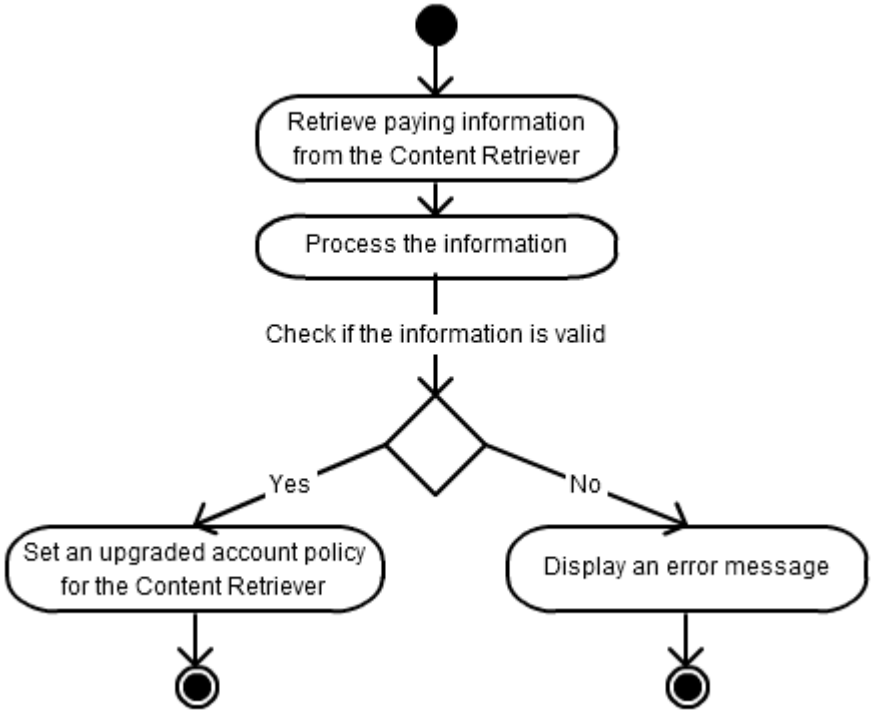


Figure 126: Activity Diagram for upgrading user accounts

FR41 – Retrieving semi-structured information

Requirement category	Functional Requirement
Degree of necessity	(x) **Essential** **( ) Recommended** **( ) Optional**
Description of the requirement	BLOGFOREVER's web spider will retrieve semi-structured information in contrary with current techniques, which retrieve unstructured data. The web spider's aim will be to retrieve weblog content and distinguish specific elements such as: 1. Weblog posts 2. User comments 3. Tags assigned to each post 4. Sociological discourse structures among authors created by hyperlinks to other posts. 5. Metadata a. Content metadata b. Digital Rights Management metadata c. Technical metadata 6. Semantic information extracted from microformats
Stakeholder	Administrators
Justification / Foundation	Extracted from the Description of work.
Assessment / Measures	Semi-structured information received from the spider is correctly formatted more than 99% of the cases.
Author(s) of the requirement description	• Jaime García Llopis • Nikolaos Kasioumis

Use Case UCFR 41.1: Aggregate Content

Use Case UC41.1		Aggregate Content
Description		The Spider aggregates content from the blog pages
Actors		Spider
Assumptions	**Preconditions**	1. The Content Retriever has provided URL of the blog page
	Postconditions	1. The content of the blog page has been aggregated by the Spider
Steps		1. The Spider establishes a connection with the blog page 2. The Spider checks for intellectual property rights if the Content Provider restricted access to the blog content or not 3. If it is allowed, the Spider retrieves the content of the blog and distinguishes specific elements 4. UCFR 42.2 5. The Spider sends all the content (weblog posts, user comments, metadata, etc.) to the Platform 6. Otherwise, display an error message about the failure
Parent		
Trigger		The Content Retriever's aggregation request or the Administrator's approval
Variations		
Issues		

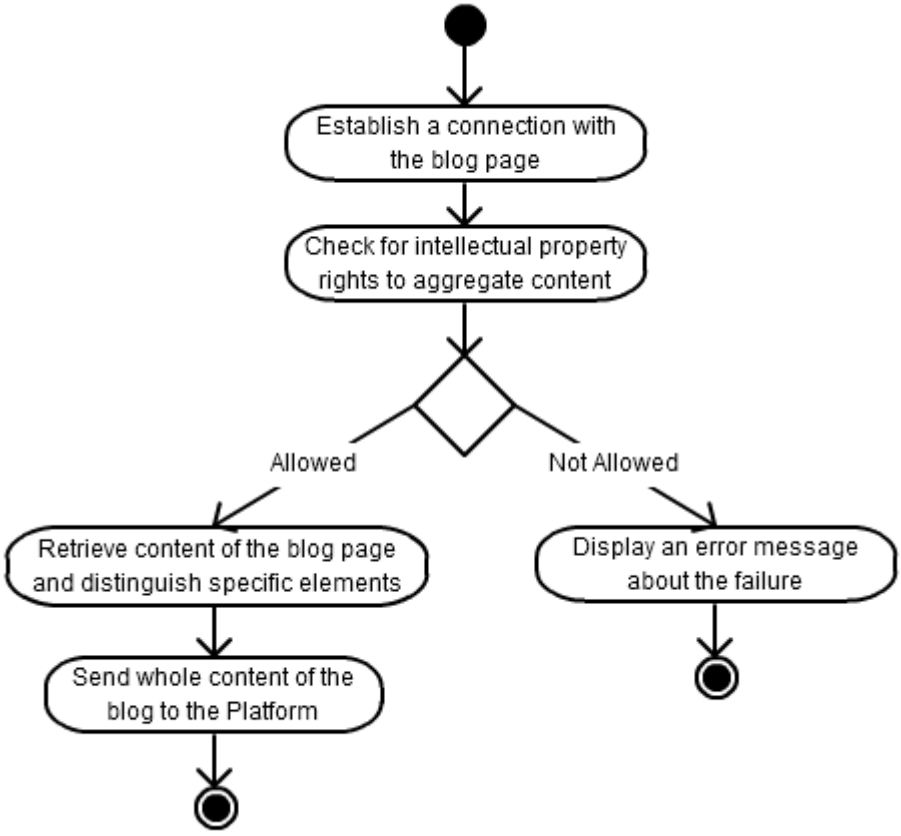


Figure 127: Activity Diagram for aggregating content from blogs

FR42 – Weblog content validation and spam filtering

Requirement category	Functional Requirement
Degree of necessity	(x) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	BLOGFOREVER's weblog spider will validate weblog content using online spam detection web services such as Akismet and Splogspot. BLOGFOREVER will use multiple weblog spam detection services in conjunction with custom filtering and auditing procedures in order to determine a robust weblog spam filtering policy balancing high quality content as well as acceptable performance.
Stakeholder	Administrators
Justification / Foundation	Extracted from the Description of work.
Assessment / Measures	The amount of spam in the platform is almost zero: a user or an administrator would very rarely find a blog that they could consider spam.
Author(s) of the requirement description	• Jaime García Llopis • Nikolaos Kasioumis

Use Case UCFR 42.1: Add Spam Detection Service

Use Case UC42.1		Validate Content
Description		The Spider validates blog content using online spam detection web services
Actors		Spider, Administrator
Assumptions	**Preconditions**	1. The Administrator is logged in
	Postconditions	1. The Spider has validated content of the blog page
Steps		1. The Administrator views the Spider's Configuration Panel 2. The Administrator selects the online spam detection services to be used. 3. The Spider configures itself to use selected services
Parent		
Trigger		The Spider has retrieved blog content
Variations		
Issues		

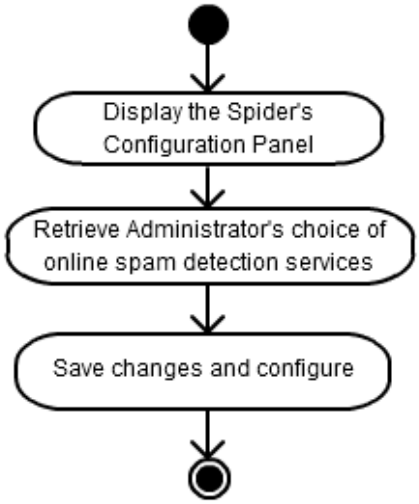


Figure 128: Activity Diagram for adding spam detection services

Use Case UCFR 42.2: Validate Content

Use Case UC42.2		Validate Content
Description		The Spider validates blog content using online spam detection web services
Actors		Spider
Assumptions	**Preconditions**	1. The Spider has established a connection with the blog page
	Postconditions	1. The Spider has validated content of the blog page
Steps		1. The Spider retrieves the list of the selected online spam detection services 2. The Spider validates blog content by using the spam detection services
Parent		
Trigger		The Spider has retrieved blog content
Variations		
Issues		

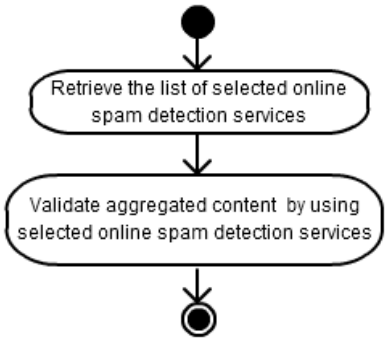


Figure 129: Activity Diagram for validating blog content

FR43 – Access to content in a harmonized way

Requirement category	Functional Requirement
Degree of necessity	(x) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	Different blogs may have different layouts and different menu structure. This could make final user browsing experience inconsistent and confusing. Blogs and blog posts should have a common way of being displayed to avoid this, and navigation menus should also be the same in all of them.
Stakeholder	Content retrievers
Justification / Foundation	Extracted from the Description of work.
Assessment / Measures	Blogs and blog posts have a common way of being displayed and they have the same navigation menu.
Author(s) of the requirement description	• Jaime García Llopis • Nikolaos Kasioumis

Note: UCFR 37.1 can be applied here

FR44 – Advanced searching

Requirement category	Functional Requirement, Data Requirement, (End)-User Interface Requirement
Degree of necessity	(x) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	Advanced searching, sorting and clustering capabilities. An innovative information retrieval component will be introduced into the code base, allowing users to have better access to information. Topic clustering as well as ranking and matching algorithms should be implemented. Visual means of presenting search results should be developed
Stakeholder	Content retrievers, content providers, administrators
Justification / Foundation	Extracted from the Description of work.
Assessment / Measures	The platform content is indexed, and the web portal offers a clear searching interface and a user manual to use the advanced search.
Author(s) of the requirement description	• Jaime García Llopis • Nikolaos Kasioumis

Use Case UCFR 44.1: Advanced Search

Use Case UC44.1		Advanced Search
Description		The Content Retriever can search blog content in the Platform with advanced features and operators
Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Content Retriever views the Advanced Search Interface
	Postconditions	1. The Platform retrieves the search result and displays to the Content Retriever
Steps		1. The Content Retriever enters the search keywords and other information (Boolean operators, date, sorting method, ranking, etc.…) 2. The Platform saves the Content Retriever's search information 3. The Platform executes the query retrieved from the Content Retriever 4. The Platform retrieves the matching and similar records 5. The Platform displays the search results
Parent		UCFR 37.1
Trigger		The Content Retriever enters search information and presses a "Search" button
Variations		
Issues		

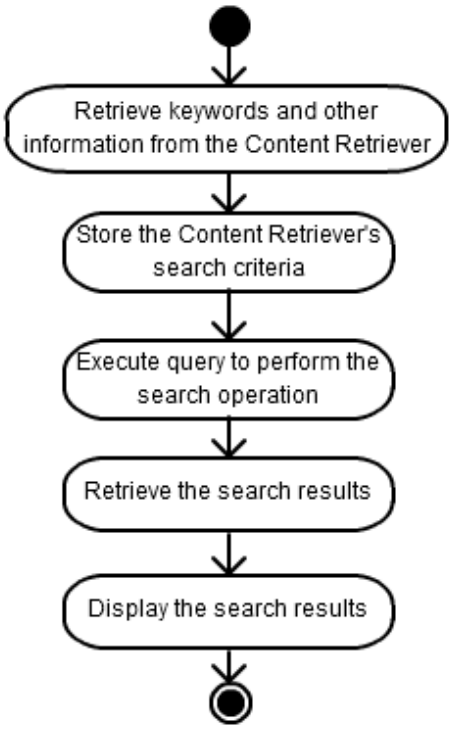


Figure 130: Activity Diagram for advanced search

Use Case UCFR 44.2: Get Content by Topic

Use Case UC44.2		Get Content by Topic
Description		The Content Retriever can retrieve blog posts according to their topics
Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Platform displays all the topics in the archive when the user is logged in
	Postconditions	1. The Content Retriever is able to see all the blog posts that have a certain topic
Steps		1. The Content Retrieves clicks on a topic 2. The Platform retrieves blog posts that have the same topic 3. The Platform displays blog posts have the same topic
Parent		
Trigger		The Content Retriever clicks on a topic name
Variations		
Issues		

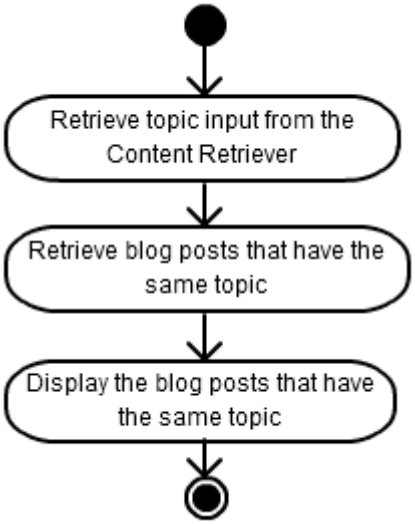


Figure 131: Activity Diagram for retrieving blog posts by their topics

FR45 – Personalized filtering services

Requirement category	Functional Requirement
Degree of necessity	(x) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	Personalized services that allow registered users to systematically filter their searches based on their individual research interests. A user should be able to use personalized lists of favourite blogs or blog posts. This list could also be filled automatically with blogs and posts matching certain searching criteria inserted by the user.
Stakeholder	Content retrievers, content providers, administrators
Justification / Foundation	Extracted from the Description of work.
Assessment / Measures	A user can use personalized lists. The user can insert searching criteria to automatically fill the list.
Author(s) of the requirement description	• Jaime García Llopis • Nikolaos Kasioumis

Use Case UCFR 45.1: Fill Favorites List

Use Case UC45.1		Fill Favorites List
Description		The users of the platform can insert searching criteria to automatically fill their favorites list
Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Content Retriever views the Advanced Search Interface
	Postconditions	1. The Platform has added all the search results to his/her favorites list
Steps		1. The Content Retriever enters the search keywords and other information 2. The Content Retriever marks "Add to Favorites" option 3. The Platform executes the query retrieved from the Content Retriever 4. The Platform retrieves the matching and similar records 5. The Platform fills the Content Retriever's favorites list with the search results 6. The Platform displays the search results
Parent		UCFR 44.1
Trigger		The Content Retriever clicks on a "Search" button
Variations		
Issues		

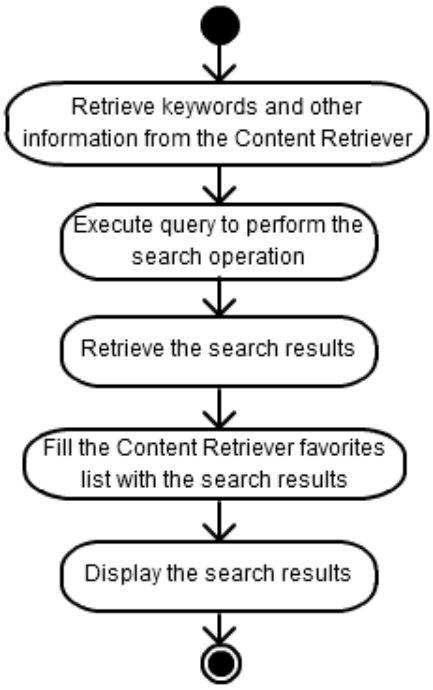


Figure 132: Activity Diagram for content a user's favorites list by using inserting a searching criteria

Use Case UCFR 45.2: Add to Favorites

Use Case UC45.2		Add to Favorites
Description		The users of the platform can add blogs or blog posts to their personal favorites list so that they can reach them later easily
Actors		Content Retriever, Platform
Assumptions	**Preconditions**	1. The Content Retriever has viewed a blog or a blog post
	Postconditions	1. The Content Retriever has viewed a blog or a blog post
Steps		1. The Content Retriever clicks on “Add to Favorites” button/link 2. The Platform adds blog or blog post to Content Retriever’s favorites list
Parent		
Trigger		The Content Retriever clicks on a button/link
Variations		
Issues		

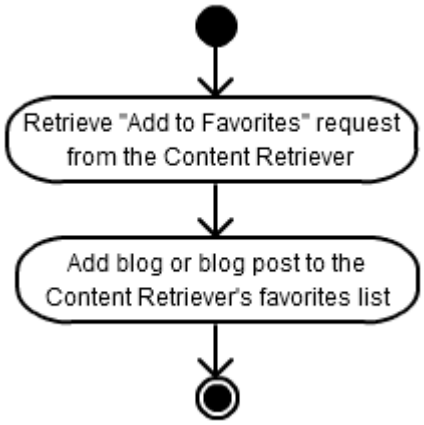


Figure 133: Activity Diagram for adding a blog/blog post to a user’s favorites list

Use Case UCFR 45.3: Filter Search Results

Use Case UC45.3		Filter Search Results
Description		The Platform can filter search results according to the Content Retriever's previous searching criteria
Actors		Platform
Assumptions	**Preconditions**	1. The Content Retriever has sent an advanced search request
	Postconditions	1. The Platform retrieves and filters the search results
Steps		1. The Platform retrieves the search results 2. The Platform filters the search results according to the Content Retriever's previous search criteria. 3. The Platform displays the search results
Parent		UCFR 44.1
Trigger		The Content Retriever enters the keywords and presses a "Search" button
Variations		
Issues		

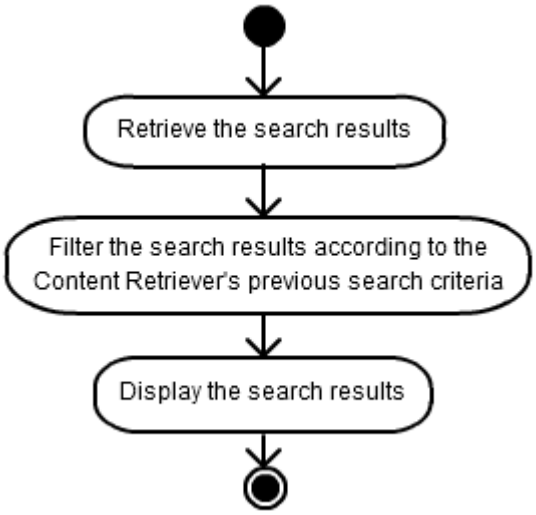


Figure 134: Activity Diagram for filtering search results

FR46 – Internationalization

Requirement category	Functional Requirement, Data Requirement, (End)-User Interface Requirement
Degree of necessity	(x) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	Internationalization will be also one of the primary concerns during the development of the digital repository and particularly the user interface. All European languages will be supported. Internationalization will be part of search and retrieval. Using databases which have native Unicode character support, it is possible to enable data storage, indexing, searching and retrieval in virtually any language without excessive developer effort.
Stakeholder	Content retrievers, content providers, administrators
Justification / Foundation	Extracted from the Description of work.
Assessment / Measures	A user can easily change the language used in the interface choosing any European language. Databases are encoded in UTF-8.
Author(s) of the requirement description	• Jaime García Llopis • Nikolaos Kasioumis

Use Case UCFR 46.1: Change Language

Use Case UC46.1		Change Language
Description		Users are allowed to change language.
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever is logged in.
	Postconditions	1. The Content Retriever has changed the language
Steps		1. The Content Retriever chooses the language from the available languages list 2. The Platform changes the language
Parent		
Trigger		Content Retriever's selection of the functionality
Variations		
Issues		

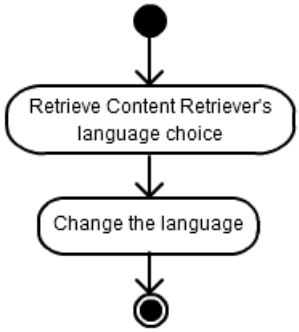


Figure 135: Activity diagram for changing the language

FR47 – Data integrity

Requirement category	Functional Requirement, Data Requirement
Degree of necessity	(x) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	The repository should include managers' tools for data quality and integrity checking.
Stakeholder	Content retrievers, administrators
Justification / Foundation	Extracted from the Description of work.
Assessment / Measures	An administrator can configure the platform to perform periodic checks on data integrity that will raise an alert if they find a corrupted file.
Author(s) of the requirement description	• Jaime García Llopis • Nikolaos Kasioumis

Use Case UCFR 47.1: Check Data Integrity

Use Case UC47.1		Check Data Integrity
Description		The BlogForever platform can be scheduled to check integrity of the data in archive
Actors		Platform, Administrator
Assumptions	**Preconditions**	1. The Administrator has viewed the Platform's Admin Interface
	Postconditions	1. The Platform has notified the Administrator about the status of the archive
Steps		1. The Administrator configures the Platform to perform periodic checks on data integrity in the archive 2. The Platform scan all the data in the archive on scheduled time 3. The Platform notifies the Administrator about the status of the data in the archive
Parent		
Trigger		The Administrator configures the Platform
Variations		
Issues		

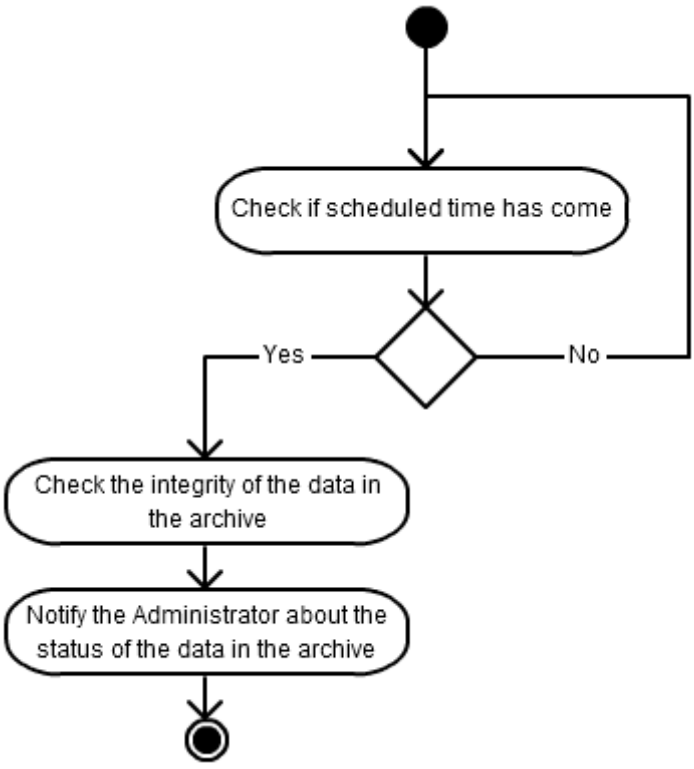


Figure 136: Activity diagram for periodic checking of the data integrity in the archive

5.2 Data Requirements

The following requirement is also a data requirement but has already been defined above as a functional requirement. Therefore, it is presented here but is not described again:

- FR37 – Web portal

DR21 – Long term digital preservation

Requirement category	Data Requirement, Operational Requirement, Functionality Requirement
Degree of necessity	(x) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	Long term digital preservation of the archived content, validation of files, checksums, functional preservation, "identifying, safeguarding and preserving archival records and ensuring that these are accessible and understandable". Sophisticated electronic records management systems will therefore include not only functions necessary for document management, but also features supporting these longer-term perspectives. These include: the association of contextual and structural data within a document; the construction and management of audit trails; document version control; support for disposition scheduling; and scheduling; and maintenance of the relationships between records in files, file series, and the corporate filing plan.
Stakeholder	Content retrievers, content providers, administrators
Justification / Foundation	Extracted from the Description of work.
Assessment / Measures	The platform will implement the Digital Preservation Policy that will be developed in WP3.
Author(s) of the requirement description	• Jaime García Llopis • Nikolaos Kasioumis

Use Case UCDR 21.1: Tag Post

Use Case UC21.1		Tag Post
Description		Users are allowed to tag posts with their own metadata
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever is logged in.
	Postconditions	1. The Content Retriever has tagged post.
Steps		1. The Content Retriever displays a blog post 2. The Content Retriever enters necessary information into a box to tag the post 3. The Content Retriever clicks on a button to add tagging information 4. The Platform stores the tagging information
Parent		
Trigger		The Content Retriever clicks a button to add tags to the posts
Variations		
Issues		

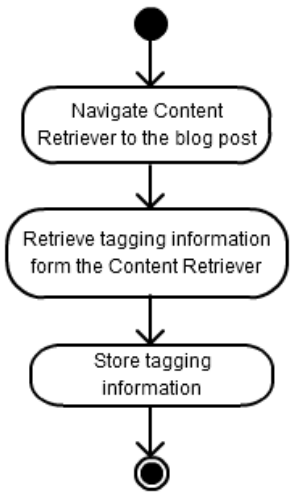


Figure 137: Activity Diagram for post tagging

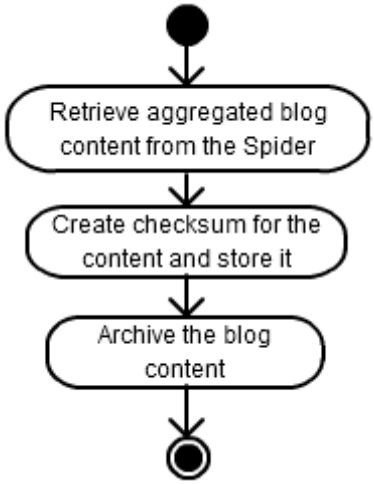
Use Case UCDR 21.2: Audit Trails

Use Case UC21.2		Audit Trails
Description		The Administrator can track all the operations (Who accessed? When? Content updated?, etc.) on a blog page
Actors		Platform, Administrator
Assumptions	**Preconditions**	1. The Administrator views a blog page
	Postconditions	1. The Administrator views all the operations on blog pages
Steps		1. The Administrator clicks on a link/button/tab 2. The Platform retrieves a list of all the operations performed on a blog page (all accesses, access times, changes, etc.…) 3. The Platform displays the operations to the Administrator
Parent		
Trigger		The Administrator clicks on a link/button/tab
Variations		
Issues		

Figure 138: Activity diagram for auditing trails on a blog page

Use Case UCDR 21.3: Archive Content

Use Case UC21.3		Archive Content
Description		The Platform archives the aggregated content from the blog pages
Actors		Platform
Assumptions	**Preconditions**	1. The Spider has aggregated content
	Postconditions	1. The Platform has archived the blog content.
Steps		1. The Platform retrieves the content from the Spider 2. The Platform creates a checksum for the content and stores it 3. The Platform archives the blog content
Parent		
Trigger		Spider sends the aggregated content to the platform
Variations		
Issues		



DR22 – METS

Requirement category	Data Requirement, Operational Requirement
Degree of necessity	(x) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	Metadata Encoding and Transmission Standard (METS) schema is a standard for encoding descriptive, administrative, and structural metadata regarding objects within a digital library expressed using the XML Schema language of the W3C.
Stakeholder	Content retrievers, administrators
Justification / Foundation	Extracted from the Description of work.
Assessment / Measures	The system uses METS to keep and export metadata.
Author(s) of the requirement description	• Jaime García Llopis • Nikolaos Kasioumis

Use Case UCDR 22.1: Export in METS

Use Case UC22.1		Export in METS
Description		The Content Retriever can export blog content in METS format
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever views a blog page
	Postconditions	1. The Content Retriever has exported content
Steps		1. The Content Retriever clicks on "METS" from available output formats list 2. The Platform converts the content from the default encoding format to METS 3. The Platform displays the content in METS
Parent		UC UCIR 6.1
Trigger		The Content Retriever clicks on "METS" link/button
Variations		
Issues		

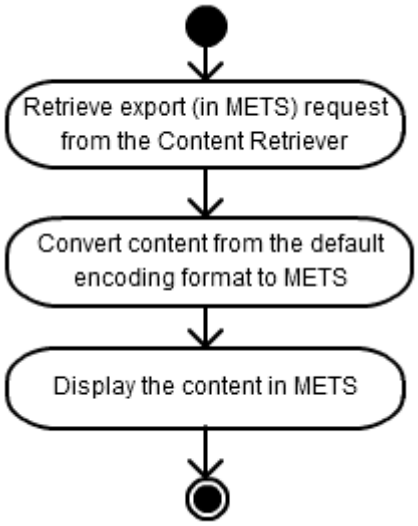


Figure 139: Activity Diagram for exporting content in METS

5.3 Interoperability Requirements

IR6 – Facilities to enable interoperability

Requirement category	Interoperability Requirement
Degree of necessity	(x) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	Facilities to enable interoperability with existing European digital libraries and infrastructures such as the European Digital Library project will be necessary. The system must become part of the Open Archives Initiative and be capable of publishing OAI metadata in a variety of schemas (including unqualified DC, the PKP (Open Journal Systems/Open Conference Systems) Dublin Core extension, MODS, and MARCXML). Additional schema must be supported via plugins. **BlogForever will integrate its content with existing digital libraries and digital preservation facilities.
Stakeholder	Content retrievers, content providers, administrators
Justification / Foundation	Extracted from the Description of work.
Assessment / Measures	Proven interoperable functionality with other standard systems such as the European Digital Library project.
Author(s) of the requirement description	• Jaime García Llopis • Nikolaos Kasioumis

Use Case UCIR 6.1: Export Content

Use Case UC6.1		Export Content
Description		The Content Retriever can export blog content in various output formats
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever views a blog page
	Postconditions	1. The Content Retriever has exported content
Steps		1. The Content Retriever clicks on the name of output format (DC, MODS, MARCXML, etc.) 2. The Platform converts the content from the default encoding format to the selected output format 3. The Platform displays the converted content
Parent		
Trigger		The Content Retriever clicks on the name of an output format
Variations		
Issues		

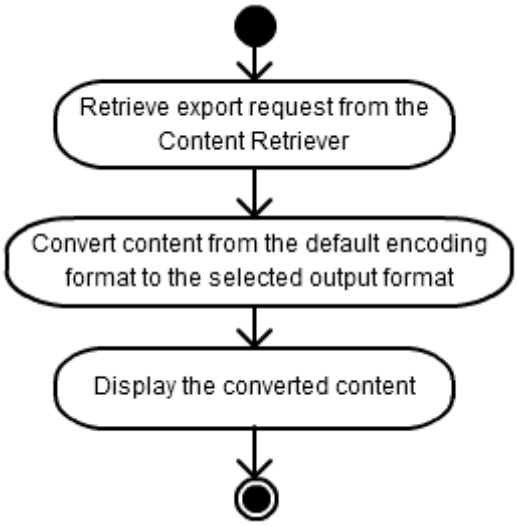


Figure 140: Activity Diagram for exporting content

IR7 – OpenURL support

Requirement category	Interoperability Requirement
Degree of necessity	(x) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	OpenURL Support is also required. The OpenURL standard is designed to support mediated linking from information resources (sources) to library services (targets). A "link resolver", or "link-server", parses the elements of an OpenURL and provides links to appropriate services as identified by a library. A source is generally a bibliographic citation or bibliographic record used to generate an OpenURL. A target is a resource or service that helps satisfy user's information needs. Examples include full-text repositories; abstracting, indexing, and citation databases; online library catalogues; and other Web resources and services.
Stakeholder	Content retrievers
Justification / Foundation	Extracted from the Description of work.
Assessment / Measures	The system supports OpenURL.
Author(s) of the requirement description	• Jaime García Llopis • Nikolaos Kasioumis

Use Case UCIR 7.1: Add OpenURL

Use Case UC7.1		Add OpenURL
Description		The Administrator can enter an OpenURL to retrieve links to the services such as licensed and/or free e-copies of full-text articles, online catalogues and etc.
Actors		Platform, Administrator
Assumptions	**Preconditions**	1. The Administrator is logged in
	Postconditions	1. The Platform has retrieved services
Steps		1. The Administrator provides an OpenURL to the Platform 2. The Platform parses the OpenURL and retrieves links to the services 3. The Platform stores the links to the services
Parent		
Trigger		The Administrator enters an OpenURL
Variations		
Issues		

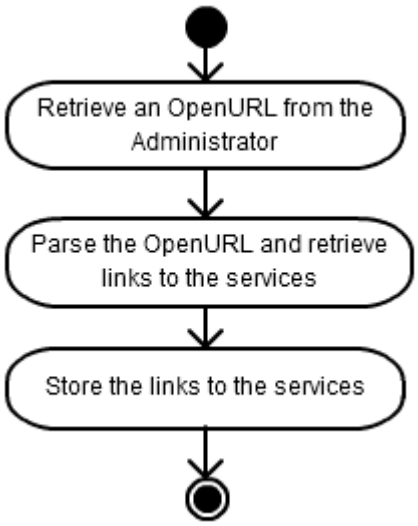


Figure 141: Activity Diagram for adding an OpenURL

Use Case UCIR 7.2: Display External Services

Use Case UC7.2		Display External Services
Description		The Content Retriever can display the links to the services stored in the Platform
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever is logged in
	Postconditions	1. The Content Retriever views the links to the services
Steps		1. The Content Retriever clicks “External Services” link/tab/button 2. The Platform retrieves the list of links to the services 3. The Platform displays the list of the links to the services
Parent		
Trigger		The Content Retriever clicks a link/tab/button
Variations		
Issues		

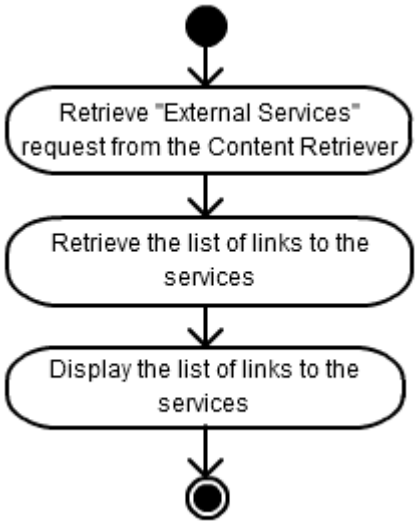


Figure 142: Activity Diagram displaying links to the external services

IR8 – Digital Object Identifier

Requirement category	Interoperability Requirement
Degree of necessity	(x) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	Digital Object Identifier (DOI) is a permanent identifier given to an electronic resource that, in contrast to a URL, does not depend on the electronic document's location. The International DOI Foundation (IDF) defines DOI name as "a digital identifier for any object of intellectual property"; it explains that the DOI is used for "persistently identifying a piece of intellectual property on a digital network and associating it with related current data in a structured extensible way".
Stakeholder	Content retrievers, administrators
Justification / Foundation	Extracted from the Description of work.
Assessment / Measures	The system uses DOI to identify the objects archived and exported.
Author(s) of the requirement description	• Jaime García Llopis • Nikolaos Kasioumis

Use Case UCIR 8.1: Assign DOI

Use Case UC8.1		Assign DOI
Description		The Administrator can assign DOI to the content in the archive
Actors		Platform, Administrator
Assumptions	**Preconditions**	1. The Administrator views a blog page
	Postconditions	1. The Administrator has assigned DOI to a blog page
Steps		1. The Administrator enters a new DOI to a box provided 2. The Platform checks if the DOI provided, used before 3. If yes, the Platform displays an error message 4. Otherwise, the Platform changes the DOI of the blog page
Parent		
Trigger		The Administrator enters a DOI to a box provided
Variations		
Issues		

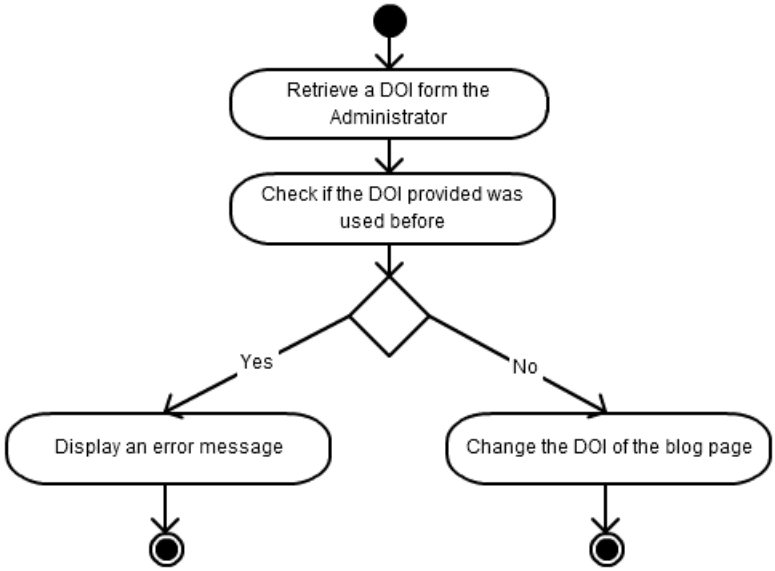


Figure 143: Activity Diagram for assigning a DOI to a blog page

Use Case UCIR 8.2: Search by DOI

Use Case UC8.2		Search by DOI
Description		The Content Retriever can search a blog page by its DOI
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever has viewed the Advanced Search panel
	Postconditions	1. The Content has viewed search results
Steps		1. The Content Retriever enters the DOI of the blog page through the Search Interface 2. The Content Retriever select “DOI” from the “category” box 3. The Platform displays the search result
Parent		UCFR 44.1
Trigger		The Content Retriever enters the DOI of the blog page
Variations		
Issues		

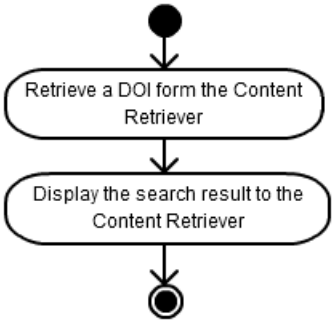


Figure 144: Activity Diagram for searching a blog page by its DOI

5.4 (End-)User Interface Requirements& Usability

The functional requirement “FR37 – Web portal” is also a user interface requirement. It has already been defined above.

5.5 Performance Requirements

PR3 – Real time archiving

Requirement category	Performance Requirement
Degree of necessity	(x) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	The web spider will constantly monitor weblogs' RSS feeds in order to detect updates and perform instant weblog archive synchronization. BLOGFOREVER will utilize Blog Pings to perform Real Time Archiving. BLOGFOREVER will actively monitor weblogs and perform online content aggregation as soon as content is published online, improving dramatically the archive's fidelity.
Stakeholder	Administrators, content providers
Justification / Foundation	Extracted from the Description of work.
Assessment / Measures	Blog content in the platform is kept up-to-date with its sources by a specified amount of time defined by the administrator.
Author(s) of the requirement description	• Jaime García Llopis • Nikolaos Kasioumis

Use Case UCPR 3.1: Subscribe to Ping Services

Use Case UC3.1		Subscribe to Ping Services
Description		The Spider can subscribe to the ping services that notify the Spider when a change occurs, provided by the blog providers.
Actors		Spider, Content Provider
Assumptions	**Preconditions**	1. The Content Provider is logged in.
	Postconditions	1. The Spider has subscribed to the ping services
Steps		1. The Spider sends a "subscribe" request to the Content Provider 2. The Content Provider accepts this request 3. The Content Provider notifies the Spider when a change in the blog content occurs 4. The Spider aggregates the content
Parent		UCFR 41.1
Trigger		The Spider sends a subscribe request
Variations		
Issues		

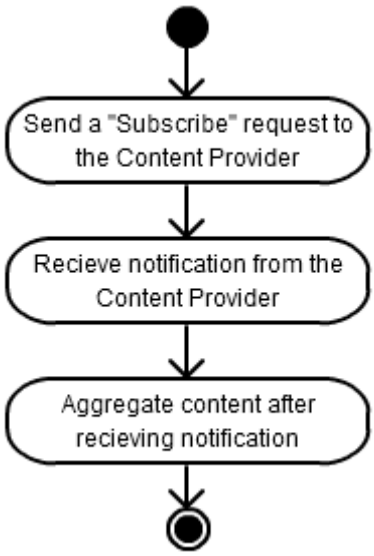


Figure 145: Activity Diagram for subscribing to ping service

Use Case UCPR 3.2: Monitor Blog Pages

Use Case UC3.2		Monitor Blog Pages
Description		The Spider can monitor blog pages to detect updates and aggregate content immediately.
Actors		Spider
Assumptions	**Preconditions**	1. The Content Retriever has provided URL of the blog page.
	Postconditions	1. The Spider has aggregated the content of the blog page
Steps		1. The Spider establishes a connection with the blog page 2. The Spider monitors the changes/updates in the blog page 3. The Spider aggregates the content when a change in the content occurs
Parent		UCFR 41.1
Trigger		The Content Retriever provides URL of the blog page
Variations		
Issues		

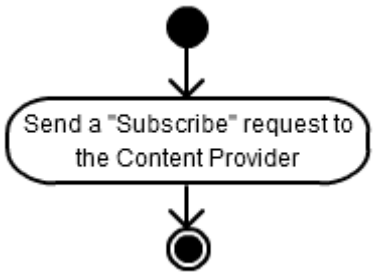


Figure 146: Activity Diagram for monitoring blog pages

5.5.1 Capacity and Scalability Requirements

CS4 – Clustering and high availability architectures

Requirement category	Capacity and Scalability Requirement
Degree of necessity	(x) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	**Clustering and high availability architectures** must be supported in order to achieve high levels of reliability and accessibility. The system's design and implementation should accommodate at least 10 million items (weblog posts), assuming adequate underlying capacity. • Dynamic deployment of BLOGFOREVER's nodes and services • Integration of external services requiring minimal effort • Provision of existing services to other service providers. • The system must scale from a single-server to the largest organizational multi-server system.
Stakeholder	• Developers • Content retrievers • Administrators
Justification / Foundation	Extracted from the Description of work.
Assessment / Measures	The system must have the expected functionality for all use cases described in the DoW.
Author(s) of the requirement description	• Jaime García Llopis • Nikolaos Kasioumis

CS5 – Modularity

Requirement category	Capacity and Scalability Requirement
Degree of necessity	(x) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	Service-oriented Architecture (SOA) style for creating and using processes packaged as services, throughout their life cycle. SOA allow different applications to exchange data and participate in business processes. These functions are loosely coupled with the operating systems and programming languages underlying the applications. SOA separates functions into distinct units (services), which can be distributed over a network and can be combined and reused to create business applications.
Stakeholder	• Developers • Administrators
Justification / Foundation	Extracted from the Description of work.
Assessment / Measures	The system architecture is divided in modules, with clear boundaries and well-documented interfaces between them.
Author(s) of the requirement description	• Jaime García Llopis • Nikolaos Kasioumis

5.6 Operational Requirements

The following requirements are also operational requirements but have already been defined above in the data requirements. Therefore, they are presented here but are not described again:

- DR21 – Long term digital preservation
- DR22 – METS

OP1 – Versioning

Requirement category	Operational Requirement
Degree of necessity	(x) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	Versioning will be enabled for every data object stored in the digital repository. Versioning, in which multiple revisions of items are stored in a repository, will become increasingly important as the BLOGFOREVER repository grows older and content is migrated to new formats and technologies. Versioning in our model covers linear revisions, not alternative representations. Hence, video, audio and transcribed-text "versions" of a particular lecture, say, could be deposited as different Manifestations within an Item, or different Items related to one another, but would not be considered as two Versions of the same Item.
Stakeholder	Content providers, content retrievers
Justification / Foundation	Extracted from the Description of work.
Assessment / Measures	The system supports versioning for every data object type.
Author(s) of the requirement description	• Jaime García Llopis • Nikolaos Kasioumis

Use Case UCOP 1.1: Create New Version

Use Case UC1.1		Create New Version
Description		The Platform creates a new version of the blog page when its content is updated
Actors		Spider, Platform, Content Provider
Assumptions	**Preconditions**	1. The Content Provider is logged in
	Postconditions	1. The Platform has created a new version of the blog page
Steps		1. The Content Provider updates the content of the blog page 2. UCFR 41.1 3. The Platform creates a new version for the blog page 4. The Platform archives the content
Parent		
Trigger		The Content Provider updates the blog page
Variations		
Issues		

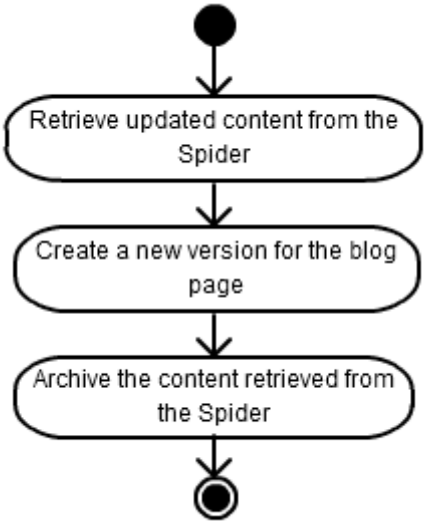


Figure 147: Activity Diagram for creating new version for blog content

Use Case UCOP 1.2: Version History

Use Case UC1.2		Version History
Description		The Content Retriever is able to view all the versions of a blog page
Actors		Platform, Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever views a blog page
	Postconditions	1. The Content Retriever has viewed an old version
Steps		1. The Content Retriever clicks on “Version History” button 2. The Platform retrieves all the versions of the blog page 3. The Platform displays all versions of the blog page 4. The Content Retriever clicks on a specific old version 5. The Platform retrieves the specific version of the blog page 6. The Platform displays the specific version of the blog page
Parent		
Trigger		The Content Retriever clicks on “Version History” button
Variations		
Issues		

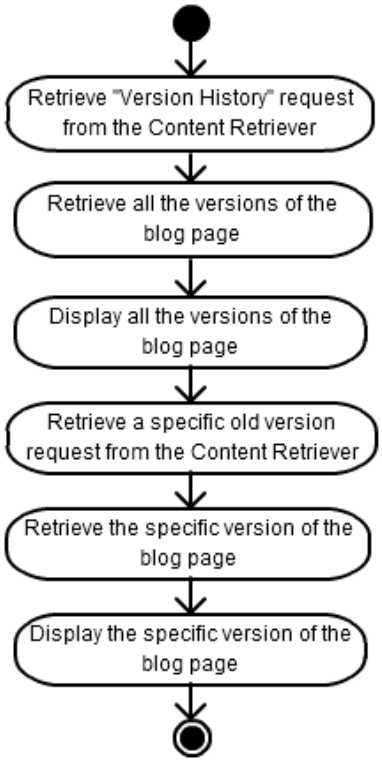


Figure 148: Activity Diagram for viewing an old version

OP2 – OAIS

Requirement category	Operational Requirement
Degree of necessity	(x) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	BlogForever will utilize OAIS. As part of OAIS, BlogForever will support digital preservation activities such as keeping track of the location, provenance, and syntactic and semantic information required to be able to validate the authenticity, integrity and completeness of a digital object and to be able to reproduce a facsimile of its initial instantiation, etc.
Stakeholder	Content providers, content retrievers, administrators
Justification / Foundation	Extracted from the Description of work.
Assessment / Measures	The system is OAIS compliant.
Author(s) of the requirement description	• Jaime García Llopis • Nikolaos Kasioumis

OP3 – APIs for developers

Requirement category	Operational
Degree of necessity	(x) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	BlogForever will provide APIs for developers to interact with weblog data: read, write, maintenance and configuration functions may be provided.
Stakeholder	Developers
Justification / Foundation	Extracted from the Description of work.
Assessment / Measures	Developers can use an API to read and write data, configure the platform and perform maintenance actions.
Author(s) of the requirement description	• Jaime García Llopis • Nikolaos Kasioumis

6 Requirement Specifications from the deliverable D2.1

In the following, the requirements are specified that were identified from the BlogForever Deliverable 2.1 Weblogs Survey Report. Thereby, the requirements are organised in subchapters that correspond to the requirement categories presented in chapter 1.3. UML models supplement the requirement descriptions. Requirements were modelled with activity diagrams if appropriate. The use case diagrams in the figures below give an overview on the modelled requirements including the responsible actors.

Figure 149: General Use Case Diagram 1

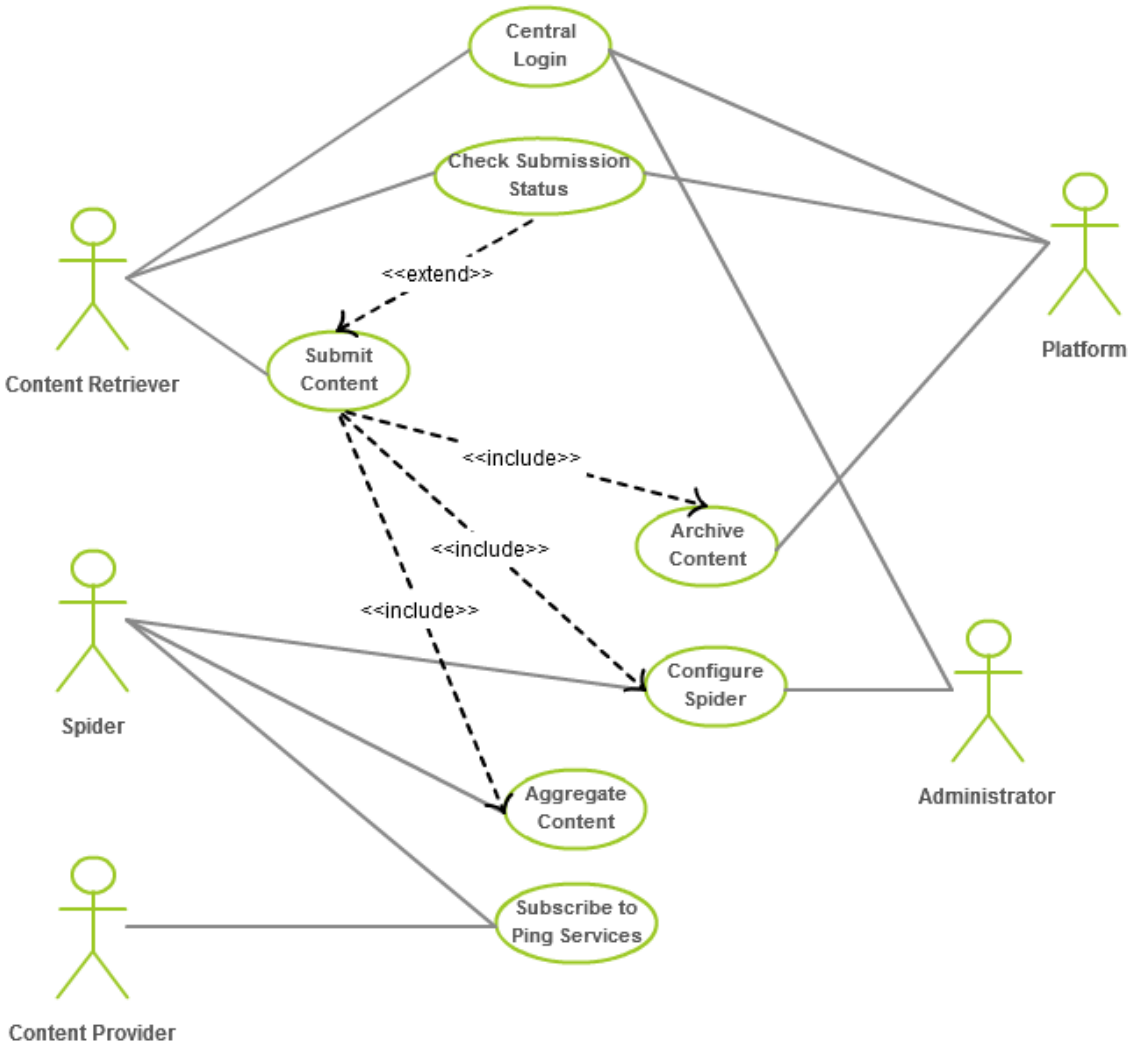


Figure 150: General Use Case Diagram 2

6.1 Functional Requirements

FR48 – Crawler/Spider Support Platform Flexibility

Requirement category	Functional Requirements
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	The archive intake spider/crawler should be able to process blogs authored on as many platforms as possible, but most importantly Mokono blog technology, Wordpress, Blogger...
Stakeholder	Content provider, content retriever and administrator
Justification / Foundation	Common blog authoring practices (Table 16) and 5.2.1 Platforms and software used (Figure 19) The survey identified that 61.3% participants were using a blog providers and the list of these blog providers and its different versions is comprehensive. Table 16 in 4.3.2.2 of the survey data indicates the most popular blogging platforms and Figure 19 from 5.2.1 details information of the considerable variation across most popular software platforms. Therefore, the platform flexibility support by the spider is essential.
Assessment / Measures	The crawler works for any platform and any version of it
Author(s) of the requirement description	• Silvia Arango-Docio

Use Case UCFR 48.1: Aggregate Content

Use Case UC48.1		Aggregate Content
Description		The Spider aggregates content from the blog pages
Actors		The Spider
Assumptions	**Preconditions**	1. The Content Retriever has provided URL of the blog page
	Postconditions	1. The content of the blog page has been aggregated by the Spider
Steps		1. The Spider establishes a connection with the blog page 2. The Spider retrieves the content of the blog 3. The Spider sends the content to the Platform
Parent		
Trigger		The Content Retriever's aggregation request or the Administrator's approval
Variations		
Issues		

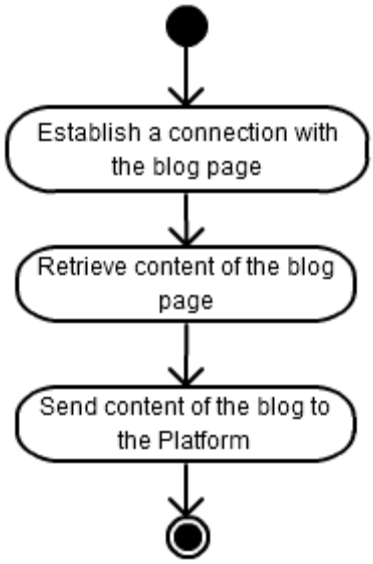


Figure 151: Activity Diagram for aggregating content from blogs

Use Case UCFR 48.2: Archive Content

Use Case UC48.2		Archive Content
Description		The Platform archives the aggregated content from the blog pages
Actors		Platform
Assumptions	**Preconditions**	1. The Spider has aggregated content
	Postconditions	1. The Platform has archived the blog content.
Steps		1. The Platform retrieves the content from the Spider 2. The Platform archives the blog content
Parent		
Trigger		Spider sends the aggregated content to the platform
Variations		
Issues		

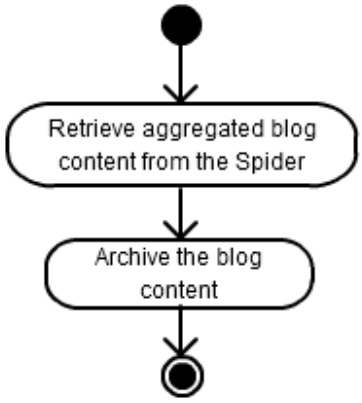


Figure 152: Activity Diagram for archiving blog contents

FR49 – Support Different Versions of Blogging Software

Requirement category	Functional Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	As well as being able to interpret multiple platforms the spider/crawler will have to be aware of the different versions of the platforms, especially with self -installed solutions such as WordPress.
Stakeholder	Content provider, content retriever and administrator
Justification / Foundation	5.2.1 Platforms and software used (Figure 20) The technical survey identified that while the number of earlier platforms across active blogs remains substantial, the majority of software platforms (average around 75%) use more recent versions. These results are limited to the providers of software packages that do specify their versions. Among the providers that do not specify information about the software version are: Blogger, Typepad and Joomla.
Assessment / Measures	The crawler works for any platform and any version of it
Author(s) of the requirement description	• Silvia Arango-Docio

Use Case UCFR 49.1: Configure Spider

Use Case UC49.1		Configure Spider
Description		The Spider can be configured so that it can support multiple platforms and their different versions
Actors		Spider & Administrator
Assumptions	**Preconditions**	1. The Spider has a configuration panel
	Postconditions	1. The Spider enables the modular functionality to process the selected type/version of blog provider
Steps		1. The Administrator views the Spider's Configuration Panel. 2. The Administrator defines how much content will be aggregated (comments, style, embedded material, snapshots etc.) 3. The Administrator selects supported blog provider options. 4. If selected blog provider has different versions that should be dealt with separately, a version list also displayed. 5. The Administrator selects different versions of blog provider platform that the spider will support. 6. In case of the Spider does not support the selected version, the Administrator selects the "Try similar versions" option. 7. The Administrator saves the changes 8. The Spider configures itself
Parent		
Trigger		Administrator's selection of the functionality
Variations		
Issues		

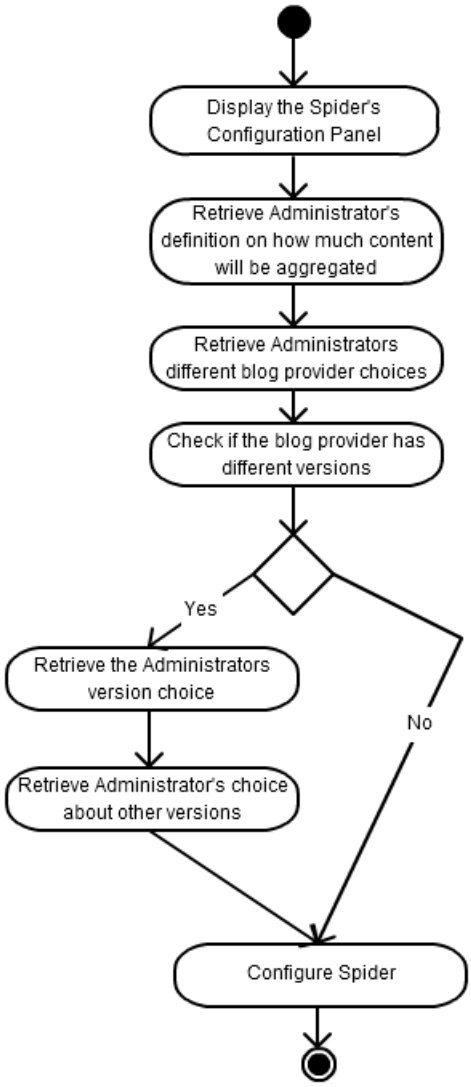


Figure 153: Activity Diagram for configuring the spider

FR50 – Most Important Content Type (Text/Html)

Requirement category	Functional Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	Given the nature of the content it would be unwise to try to separate the text from the HTML as stylistic elements within the HTML may be important to the understanding of the text, for example: indentation or use of unmarked tables to indicate quotes.
Stakeholder	Content provider, content retriever and administrator
Justification / Foundation	5.2.2 Document Character Set (Figure 21) The results for evaluating the use of content and characters attributes across the blogs studied, suggested that "text/html" is the most widely (61%) specified content type within this studied.
Assessment / Measures	Stylistic elements within the HTML are considered to the understanding of the text so we obtain correct content representation.
Author(s) of the requirement description	• Silvia Arango-Docio

FR51 – UTF-8 - The Default Character Encoding

Requirement category	Functional Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	The default character encoding on the archive should be UTF-8.
Stakeholder	Administrators
Justification / Foundation	• 5.2.2 Document Character Sets: In addition to content type Req 3, information about encoding has also captured and analyzed in 5.2.2. UTF-8 is most frequently used encoding. Other char sets did not exceed 6%. Encoding information remained unidentified in 39% of the cases (Figure 22). The overwhelming majority of studied resources are distributed in Unicode as text/html documents. • 55% of blogs are encoded in UTF-8 and it is widely adopted (http://developers.sun.com/dev/gadc/technicalpublications/articles/utf8.html)
Assessment / Measures	All the large number of characters for different written languages are rendered and represented correctly
Author(s) of the requirement description	• Silvia Arango Docio

Use Case UCFR 51.1: Check Character Encoding

Use Case UC51.1		Check Character Encoding
Description		The Platform checks for the character encoding of the blog page.
Actors		Platform
Assumptions	**Preconditions**	1. The Spider sends the content of the blog page to the Platform
	Postconditions	1. The Platform converts the content to UTF-8 if necessary then archives with original character encoding
Steps		1. The Platform retrieves content from the Spider component 2. The Platform checks the character encoding of the content 3. If the character encoding is not UTF-8, the Platform converts the content to UTF-8 4. The platform archives the content with original character encoding information
Parent		
Trigger		The Spider sends the content of the blog page
Variations		
Issues		

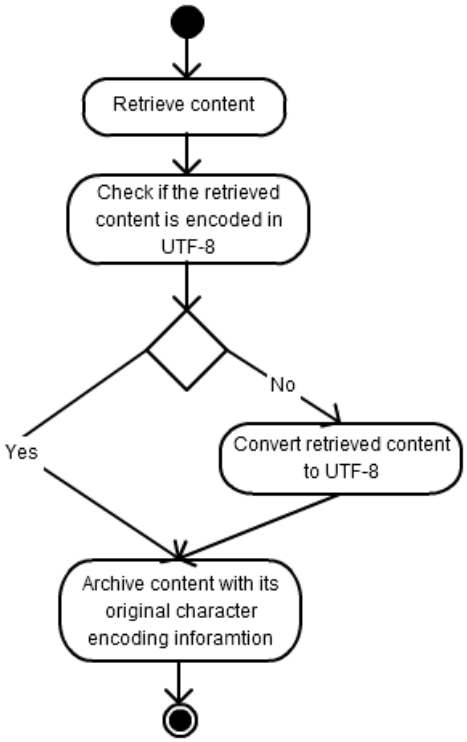


Figure 154: Activity Diagram for checking the character encoding

FR52 - Sharing and Promotion Beyond the Archive

Requirement category	Functional Requirement, (End)- User Interface Requirements
Degree of necessity	( ) **Essential** ( ) **Recommended** (X) **Optional**
Description of the requirement	Archive users who comment on blogs within the archive may want their comments to be pushed to the source blog, if still available. In addition to 'keep private' or 'public' an additional option should be available when submitting comments 'push to source'. Selecting this option will initiate an attempt by the archive to push a comment to the source blog
Stakeholder	Content providers and content retrievers
Justification / Foundation	4.3.2.6 Aspect of blogs for preservation (Table 29) 47% of author respondents indicated that the increasing readership and building a blog community was a key goal for blog authors within an archive. By having their blog archived authors get access to highly interested group of active readers. This satisfies their target of increased readership to an extent, but without the ability to easily push comment out of the archive to the source blog, the benefits will not be apparent to the authors. Also content retrievers will be keen to only have to post once within the archive. This will add to the usefulness of the archive from retrievers perspective
Assessment / Measures	Active archive users use the commenting tool extensively within the archive and their comments can be pushed to the source blog if still available.
Author(s) of the requirement description	• Silvia Arango Docio

Use Case UCFR 52.1: Push Comment

Use Case UC52.1		Push Comment
Description		Users are allowed to push comments to original blog
Actors		Platform & Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever is logged in.
	Postconditions	1. The Platform has pushed the Content Retriever‘s comment to the original post
Steps		1. The Content Retriever views a blog post 2. The Content Retriever enters his/her comment on a box provided below the blog post. 3. The Content Retriever marks the “Push to source“ option 4. The Content Retriever clicks on the “Add Comment” button 5. The Platform stores the comment information 6. The Platform checks if the original blog is still available. 7. If yes, the Platform checks whether it has right to write the comment 8. If yes, The Platform adds the comment to the original blog. 9. The Platform displays a message about the status of pushing operation
Parent		UC UCUI 24.1
Trigger		Content Retriever’s selection of the functionality
Variations		
Issues		

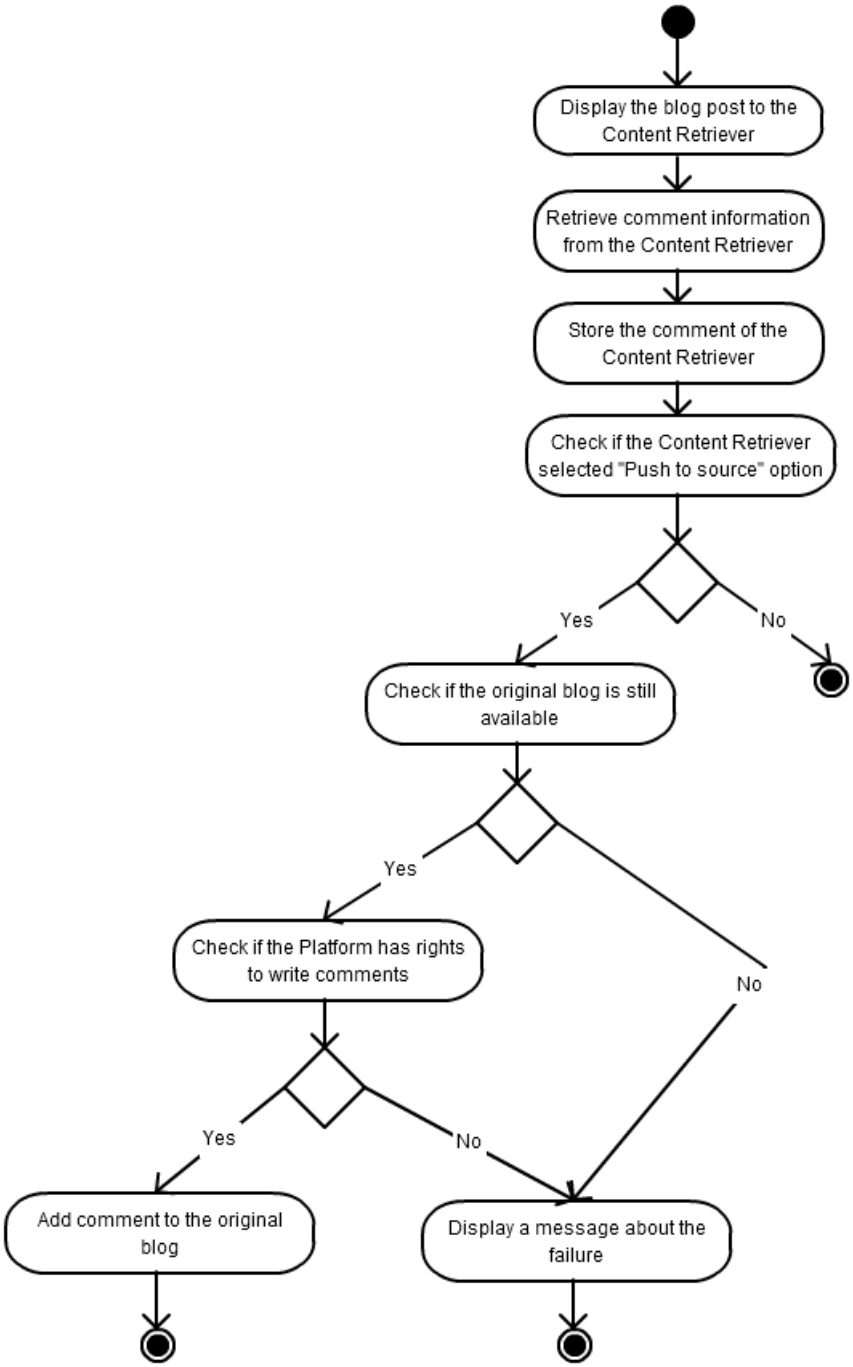


Figure 155: Activity Diagram for pushing comment to the original blog

FR53 – Snapshots Versions of Blog Designs in the Archive

Requirement category	Functional Requirement
Degree of necessity	( ) **Essential** ( ) **Recommended** (X) **Optional**
Description of the requirement	To capture the original design of a blog, the platform could snapshot the screen at the point a blog is archived. Different snapshots for the source blog could be available once the overall blog design changes.
Stakeholder	Content provider, content retriever and administrator
Justification / Foundation	4.3.2.2 Common Blog Authoring (Table 12) In terms of blog design activities, like changing the appearance or the feel of the blog, nearly 60% of the respondents declared that rarely applied those changes.
Assessment / Measures	A user can find and download the different blog design snapshots easily.
Author(s) of the requirement description	• Silvia Arango-Docio

Use Case UCFR 53.1: Subscribe to Ping Services

Use Case UC53.1		Subscribe to Ping Services
Description		The Spider can subscribe to the ping services that notify the Spider when a change occurs, provided by the blog providers.
Actors		Spider & Content Provider
Assumptions	**Preconditions**	1. The Content Provider is logged in.
	Postconditions	1. The Spider has subscribed to the ping services
Steps		1. The Spider sends a “subscribe” request to the Content Provider 2. The Content Provider accepts this request
Parent		
Trigger		The Spider sends a subscribe request
Variations		
Issues		

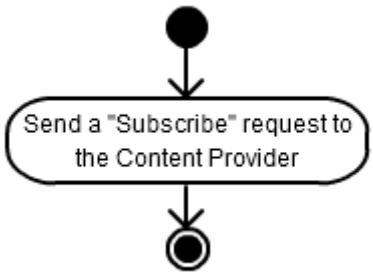


Figure 156: Activity Diagram for subscribing to ping services

FR54 – What to Archive: Text and Comments

Requirement category	Functional Requirement
Degree of necessity	(X) **Essential** ( ) **Recommended** ( ) **Optional**
Description of the requirement	The archive should contain the full text of a post plus embedded material where possible.
Stakeholder	Content providers and content retrievers
Justification / Foundation	4.3.2.6 Aspect of Blogs for Preservation (Table 30 & Figure 16 - Importance of preserving blog data) 46.3% of authors regard their entirely blog content as the highest priority to be preserved, with posts second with 45.7% of all authors indicating these elements are key to archive. Combine this with the 25% who indicate that comments are the main way of ranking a post and a compelling argument can be made to preserve comments as well as the original content.
Assessment / Measures	A user can find the archived blog content as closed to the original as possible.
Author(s) of the requirement description	• Silvia Arango-Docio

FR55 – Universal Login & Central Login

Requirement category	Functional Requirement
Degree of necessity	( ) **Essential** ( ) **Recommended** (X) **Optional**
Description of the requirement	Once logged into the platform users should be able to access all archives on the same platform - provided the archives owner has granted that user the rights to view their archive As well as the archives own login system should users be able connect form third party sites like Facebook, Google+... On the other hand, archives accessed through libraries may need/have an admin panel where library staff create and assign logins for the duration of a course or just for the day. Could there also be an option to allow users to be synced across from and LDAP server?
Stakeholder	Administrators
Justification / Foundation	Facebook and Google+ integration reduces barriers to entry and also further promotes the sharing of content from the archive. Promoting both the authors content and the archive
Assessment / Measures	A user can easily promote the archive content via a universal login.
Author(s) of the requirement description	• Silvia Arango-Docio

Use Case UCFR 55.1: Central Login

Use Case UC55.1		Central Login
Description		A login request and the login credentials of the User are retrieved from platform to be used in secure communication.
Actors		Platform & Content Retriever & Administrator
Assumptions	**Preconditions**	1. An account for the Content Retriever or Administrator must exist.
	Postconditions	1. The Content Retriever or Administrator is logged in
Steps		1. The Content Retriever or Administrator specifies required information for login (for example: username, password…). 2. The Platform checks for the Content Retriever's or Administrator's credentials. 3. The Platform navigates Content Retriever or Administrator accordingly
Parent		
Trigger		Content Retriever or Administrator clicks on a button
Variations		
Issues		

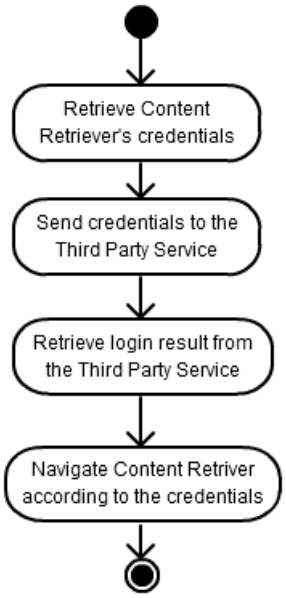


Figure 157: Activity Diagram for central login

Use Case UCFR 55.2: Universal Login

Use Case UC55.2		Universal Login
Description		A login request and the login credentials, for the Third Party Service (Facebook, Google+), of the User are retrieved from platform and sent to the service.
Actors		Platform & Content Retriever
Assumptions	**Preconditions**	1. An account for the Content Retriever from the Third Party Service must exist.
	Postconditions	1. The Content Retriever is logged in
Steps		1. The Content Retriever specifies required information for login (for example: username, password…). 2. The Platform sends the credentials of the Content Retriever to the Third Party Service 3. The Platform receives the result from the Third Party Service. 4. The Platform navigates Content Retriever according to result received from the Third Party Service.
Parent		
Trigger		Content Retriever clicks on appropriate icon among the available Third Party Services
Variations		
Issues		

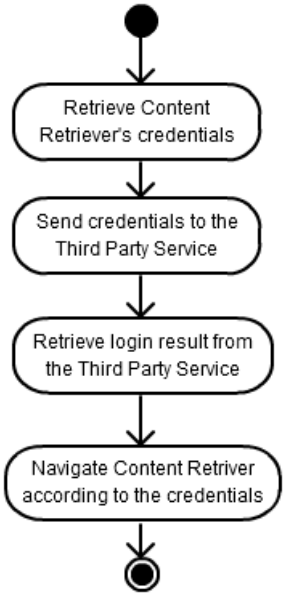


Figure 158: Activity Diagram for universal login

Use Case UCFR 55.3: Create Account for Users

Use Case UC55.3		Create Account for Users
Description		Accounts are created by administrators for the end-users
Actors		Administrator & Platform
Assumptions	**Preconditions**	1. An admin panel for the operation must exist.
	Postconditions	1. Account for the end-user is created
Steps		1. The Administrator logs in to the platform with his/her credentials. 2. The Administrator navigates to the Admin Panel 3. The Administrator sends a request to create an account 4. The Platform creates an account 5. The Administrator notifies the end-user and sends the necessary information.
Parent		
Trigger		The Administrator clicks on a button
Variations		
Issues		

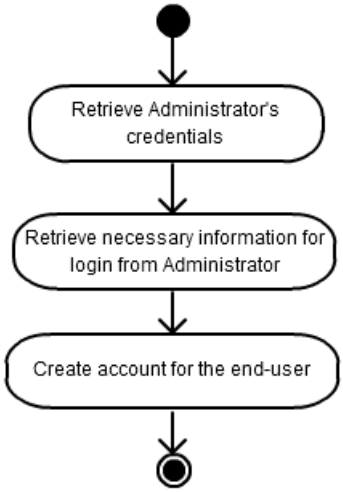


Figure 159: Activity Diagram for creating account for end-users

6.2 Data Requirements

DR23 – Mashup Activities

Requirement category	Data Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	Content from other sources should be identified and associated with a disclaimer around copyright infringement to protect the archive and hosting institution.
Stakeholder	Content provider, content retriever and administrator
Justification / Foundation	4.3.2.2 Common Blog Authoring Practices For the case of mixing, quoting and reusing content from other sources, the results were that more than 30% of the survey participants rarely mixed, quoted or reused content from others and it was never done by 25% of the respondents (Table 11)
Assessment / Measures	A user can find the required disclaimer around copyright infringement easily.
Author(s) of the requirement description	• Silvia Arango-Docio

Use Case UCDR 23.1: Get Copyright Information

Use Case UC23.1		Get Copyright Information
Description		The Platform displays a disclaimer around copyright infringement
Actors		Platform & Content Retriever
Assumptions	**Preconditions**	1. Copyright information has been received from the Spider
	Postconditions	1. Display the Disclaimer around the copyright infringement
Steps		1. The Content Retriever clicks on a link to the blog post. 2. The Platform retrieves the copyright information from the database. 3. The Platform displays the content along with the copyright information.
Parent		
Trigger		The Content Retriever clicks on a link to the blog post
Variations		
Issues		

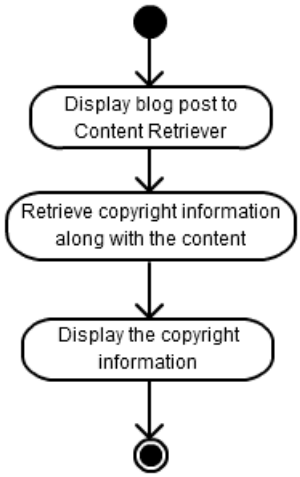


Figure 160: Activity Diagram for getting copyright information

6.3 (End-)User Interface Requirements& Usability

UI29 - Multiple Language Support

Requirement category	(End)-User Interface Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	It should be easy to switch the language of the archive to the users preference. Switching the language would localize all the menu items and explanatory texts within the archive platform.
Stakeholder	Content provider, content retriever and administrator
Justification / Foundation	4.3.2.1 author survey respondents The survey identified that at least 5 languages were present amongst significant numbers of authors (5%+ to 43%), to be clear 67% of authors did not pick English as their preferred language. Similar language distributions exist in the reader results also. Therefore, the multi language support can enhance reader and provider adoption.
Assessment / Measures	The platform supports English and two other languages (e.g. German and Greek).
Author(s) of the requirement description	• Silvia Arango Docio • Hendrik Kalb

Use Case UCUI 29.1: Change Language

Use Case UC29.1		Change Language
Description		Users are allowed to change language.
Actors		Platform & Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever is logged in.
	Postconditions	1. The Content Retriever has changed the language
Steps		1. The Content Retriever chooses the language from the available languages list 2. The Platform changes the language
Parent		
Trigger		Content Retriever's selection of the functionality
Variations		
Issues		

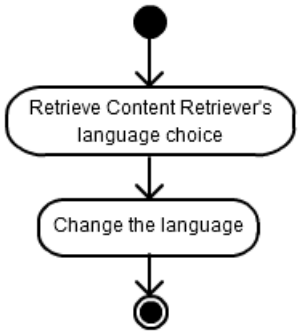


Figure 161: Activity diagram for changing the language

UI30 - Creation of a Community of Providers and Recipients within the Archive Platform

Requirement category	(End)-User Interface Requirement
Degree of necessity	() **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	Archive users should be able to comment on blog posts and share these as follows: • Do not share, keep private • Share to other archive users Sharing of comments builds an online community and thus works to deliver adoption. Also if the blog is no longer out in the world then the archive is the only place to comment *Example use case: A historian researching archived political blogs might want to leave a comment for other historical researchers*
Stakeholder	Content providers and content retrievers
Justification / Foundation	4.3.2.6 Aspect of blogs for preservation 47.3% prefer a blog community within the archive; the same percentage showed interest about 'sharing and rating' tools; 32.6% presented interest about 'blog news portals'
Assessment / Measures	A user can find the blog archive community tools easily and these tools are popular and easy to use.
Author(s) of the requirement description	• Silvia Arango Docio

Use Case UCUI 30.1: Add comment

Use Case UC30.1		Add Comment
Description		Users are allowed to leave comments on a blog post
Actors		Platform & Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever is logged in.
	Postconditions	1. The Content Retriever has commented on a blog post
Steps		1. The Content Retriever views a blog post 2. The Content Retriever enters his/her comment on a box provided below the blog post. 3. The Content Retriever clicks on the "Add Comment" button 4. The Platform stores the comment
Parent		
Trigger		Content Retriever clicks on a button
Variations		
Issues		

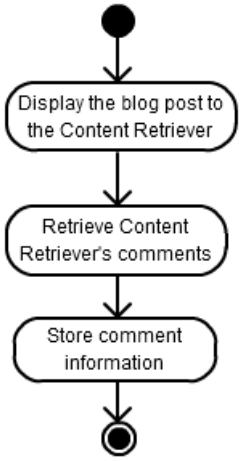


Figure 162: Activity Diagram for commenting on a blog post

Use Case UCUI 24.2: Change Comment Sharing Preferences

Use Case UC24.2		Change Comment Sharing preference
Description		Users are allowed to change sharing visibility
Actors		Platform & Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever is logged in.
	Postconditions	1. The Content Retriever has changed sharing preferences
Steps		1. The Content Retriever views his/her profile page 2. The Content Retriever changes the default sharing preference 3. The Content Retriever clicks on the “Save” button 4. The Platform saves the changes to the database
Parent		
Trigger		Content Retriever’s selection of the functionality
Variations		
Issues		

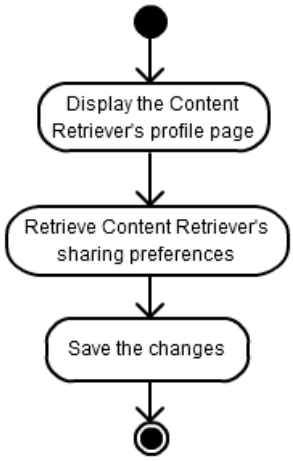


Figure 163: Activity Diagram for changing sharing preferences

UI31 – Ranking of Archived Posts

Requirement category	(End)-User Interface Requirement
Degree of necessity	( ) **Essential** ( ) **Recommended** (X) **Optional**
Description of the requirement	Within the archive users should be able to rate posts allowing a ranking or most popular/relevant blogs to be assembled within the archive. A voting system could be used in a similar way to popular Q&A sites - like Stack Overflow and Quora - http://www.quora.com/StackOverflow posts.
Stakeholder	Content providers and content retrievers
Justification / Foundation	4.3.2.4 Network-Based Metrics (Table 22) Ranking of posts has been identified in the survey as a key point of interest for users, with authors focusing upon comments, citations, WordPress stats and subscribers numbers as ways to justify engagement (Table 22 - 'Ranking Analysis Tools'). A formal ranking system in addition to comments, citations etc. will provide a way to aid archive users in finding good content.
Assessment / Measures	Users are able to rate and find good content in the archive easily
Author(s) of the requirement description	• Silvia Arango-Docio

Use Case UCUI 31.1: Rate Post

Use Case UC31.1		Rate Post
Description		Users are allowed to rate posts
Actors		Platform & Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever is logged in.
	Postconditions	1. The Content Retriever has rated the post.
Steps		1. The Content Retriever displays a blog post 2. The Content Retriever rates the post. 3. The Platform receives the rating and calculate rankings accordingly.
Parent		
Trigger		Content Retriever clicks on the symbols (stars, etc.) that represent points from 1 to 5 or 1 to 10
Variations		
Issues		

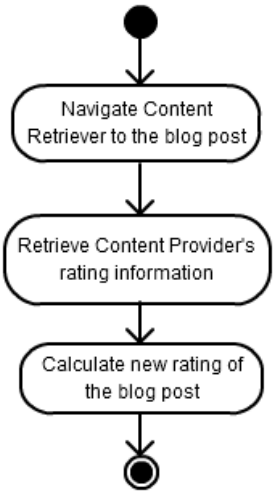


Figure 164: Activity Diagram for post rating.

Use Case UCUI 31.2: View Most Popular Posts

Use Case UC31.2		View Most Popular Posts
Description		Users are allowed to view most popular blog posts.
Actors		Platform & Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever is logged in.
	Postconditions	1. The Content Retriever has viewed most popular posts
Steps		1. The Content Retriever selects “Most Popular” option from the ranking methods list through the Search Interface 2. The Content Retriever clicks on “Search” button 3. The Platform retrieves most viewed blog posts 4. The Platform displays the Most Popular Blog Posts
Parent		
Trigger		The Content Retriever clicks on “Search” button
Variations		
Issues		

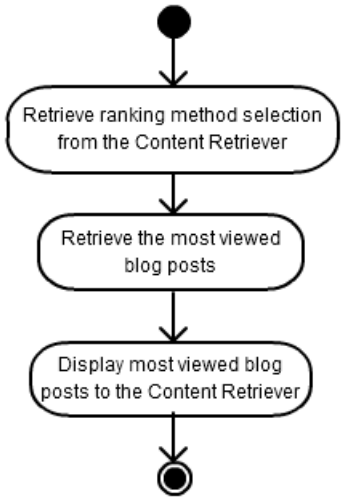


Figure 165: Activity Diagram for getting most popular posts

UI32 – Tagging System

Requirement category	(End)-User Interface Requirement
Degree of necessity	( ) **Essential** ( ) **Recommended** (X) **Optional**
Description of the requirement	Within the archive users should be able to tag posts with their own metadata (in addition to any tagging metadata already associated with the original post). These tags should be visible to all users of the archive and could be used to group content together. They can also be used to drive advanced discovery features.
Stakeholder	Content providers and content retrievers
Justification / Foundation	Appendix D – Readers Survey Data Summary Tagging of content combined with ranking can be used to aid search and also suggest good quality relevant posts based upon users profiles. Appendix D: Table 7 – Preferred Methods of Accessing a Blog Post – 39.7% access blog posts by category and 21.7% by tag to tag cloud.
Assessment / Measures	Users will be able to group and discover interesting content in an efficient way
Author(s) of the requirement description	• Silvia Arango-Docio

Use Case UCUI 32.1: Tag Post

Use Case UC32.1		Tag Post
Description		Users are allowed to tag posts with their own metadata
Actors		Platform & Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever is logged in.
	Postconditions	1. The Content Retriever has tagged post.
Steps		1. The Content Retriever displays a blog post 2. The Content Retriever enters necessary information into a box to tag the post 3. The Content Retriever clicks on a button to add tagging information 4. The Platform stores the tagging information
Parent		
Trigger		The Content Retriever clicks a button to add tags to the posts
Variations		
Issues		

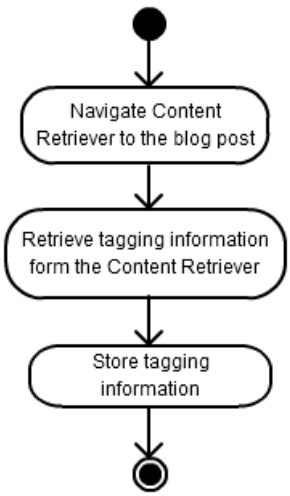


Figure 166: Activity Diagram for post tagging

Use Case UCUI 32.2: Search Posts by Tags

Use Case UC32.2		Search Posts by Tags
Description		Users are allowed to search posts by tags in the Platform
Actors		Platform & Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever is logged in.
	Postconditions	1. The Content Retriever retrieves search results.
Steps		1. The Content Retriever selects the "tag" option from the field list through the Search Interface 2. The Content Retriever fills in the search input 3. The Platform retrieves posts according to the Content Retriever's input 4. The Platform displays the search results
Parent		UC UCUI 26.1
Trigger		The Content Retriever enters the keyword and clicks Search button
Variations		
Issues		

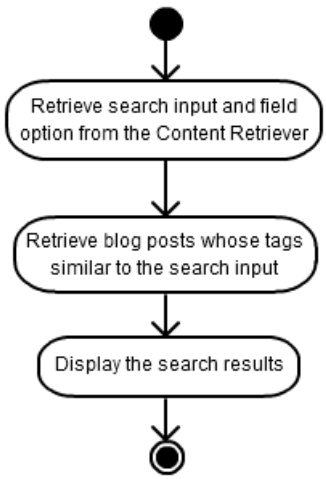


Figure 167: Activity Diagram for searching posts by their tags

UI33 – User Profiles

Requirement category	(End)-User Interface Requirement
Degree of necessity	( ) **Essential** (X) **Recommended** ( ) **Optional**
Description of the requirement	Users should be encouraged to create user profiles. Logged in users can access saved articles and, based upon the tagging of the saved articles (see Req. 13) the archive can suggest other relevant content, in a similar way to 'you might also be interested in' feature in www.amazon.co.uk for example.
Stakeholder	Content retrievers
Justification / Foundation	Appendix D – Readers Survey Data Summary Appendix D: Table 11 - "Details of Blog Content Search" indicates that readers are most interested in content relevant to their own interests (72.9%)
Assessment / Measures	A user can find relevant content via their own profiles
Author(s) of the requirement description	• Silvia Arango-Docio

Use Case UCUI 33.1: Register

Use Case UC33.1		Register
Description		Users can be registered to the Platform
Actors		Platform & Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever has not been registered before
	Postconditions	1. Account for the Content Retriever is created
Steps		1. The Content Retriever clicks on a "Register" or "Sign Up" button. 2. The Content Retriever fills in the necessary information 3. The Platform creates account for the user 4. The Platform creates and initializes a profile for the user
Parent		
Trigger		The Content Retriever clicks a button to register
Variations		
Issues		

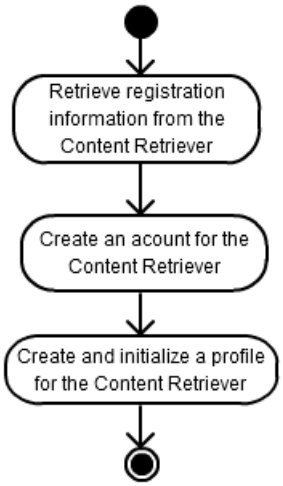


Figure 168: Activity Diagram for registering

Use Case UCUI 33.2: Suggest Relevant Content

Use Case UC33.2		Suggest Relevant Content
Description		Platform suggests posts relevant to the post that the Content Retriever currently displays.
Actors		Platform & Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever is logged in.
	Postconditions	1. The Platform has suggested relevant content
Steps		1. The Content Retriever views a blog post. 2. The Platforms displays a “Similar Posts“ link to the Content Retriever 3. The Content Retriever clicks on the link 4. The Platform retrieves a list of relevant posts by using a word similarity algorithm based upon the tagging information of the original post 5. The Platform displays the list of relevant posts
Parent		
Trigger		The Content Retriever views a blog post.
Variations		
Issues		

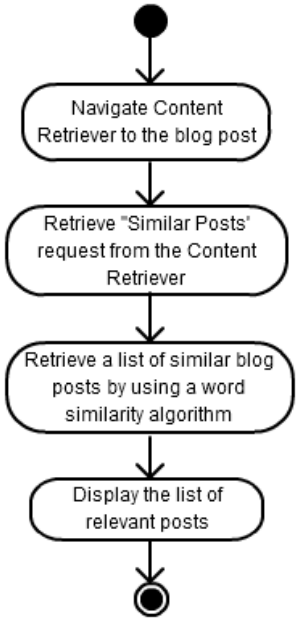


Figure 169: Activity Diagram for suggesting relevant content

UI34 – Simple Submission by Authors

Requirement category	(End-)User Interface Requirement
Degree of necessity	( ) **Essential** ( ) **Recommended** (X) **Optional**
Description of the requirement	Submission of content to an archive should be transparent and easy, i.e. complete a form and enter a URL for the spider to go to. The archive platform should notify the author of the submission and its status (new/under review/ pending approval...) and should be able to archive the blog with no further input from the author.
Stakeholder	Content providers
Justification / Foundation	4.3.2.6 Aspects of Blogs for Preservation (Table 27 - Use of external services for blog preservation) 85.7 % of blog authors have never used a blog preservation service, so the barriers to submitting their content should be low. And if vetting is to occur then the rules around this should be clearly displayed on the site.
Assessment / Measures	A user can easily submit content in the archive
Author(s) of the requirement description	• Silvia Arango-Docio

Use Case UCUI 34.1 Submit Content

Use Case UC34.1		Submit Content
Description		Users are enabled to submit contents to be archived in the Platform.
Actors		Platform & Content Retriever & Administrator
Assumptions	**Preconditions**	1. The Content Retriever is logged in.
	Postconditions	1. The Content Retriever has submitted content
Steps		1. The Content Retriever suggests blog pages through the Platform 2. The Platform notifies the Administrator about the submission details 3. UCFR 49.1 4. UCFR 48.1 5. UCFR 48.2
Parent		
Trigger		The Content Retriever clicks a button
Variations		
Issues		

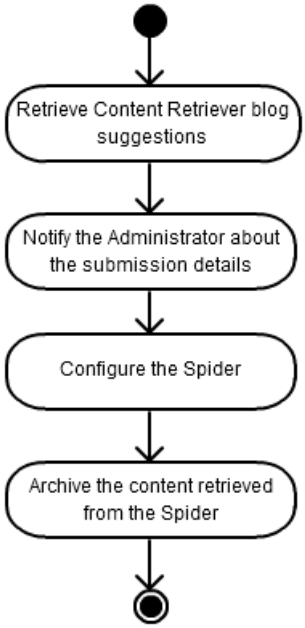


Figure 170: Activity Diagram for content submission

UI35 – Workflow to Manage Blog Submissions

Requirement category	(End)-User Interface Requirement
Degree of necessity	( ) **Essential** ( ) **Recommended** (X) **Optional**
Description of the requirement	Once content is selected for an archive a workflow tool should be implemented to track the submission and deliver automatic messages as well as personal reminders about their progress. This workflow will allow content providers to obtain information about submission progress of their blog towards its permanent preservation.
Stakeholder	Administrators
Justification / Foundation	4.3.2.6 Aspects of Blogs for Preservation (Table 27 - Use of external services for blog preservation) 85.7 % of blog authors have never used a blog preservation service, so the barriers to submitting their content should be low
Assessment / Measures	A user can easily submit content in the archive
Author(s) of the requirement description	• Silvia Arango-Docio

Use Case UC UCUI 35.1 Check Submission Status

Use Case UC30.1		Check Submission Status
Description		Users are enabled to track submission details on the system
Actors		Platform & Content Retriever
Assumptions	**Preconditions**	1. The Content Retriever has sent a content submission request
	Postconditions	1. The Content Retriever has submitted content.
Steps		1. The Content Retriever suggest blog pages to be aggregated 2. The Platform tracks for changes during the submission process 3. The Platform sends automatic messages to the Content Retriever about the submission progress.
Parent		UC UCUI 29.1
Trigger		A change in the submission status
Variations		
Issues		

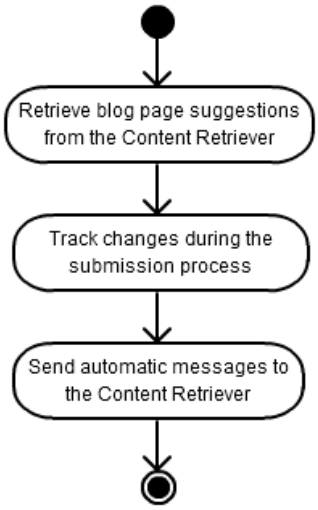


Figure 171: Activity Diagram for checking the submission status

7 Requirement Specifications by Topic

In this chapter, some requirements are ordered by different topics (only these requirements that fit to one of the topics). This provides another view that could facilitate a navigation through the requirements by topic interest, e.g. for system designers and developers. The topics are subjective and were created some of the developers who will implement the BlogForever preservation system. Hence, the topics do not claim to be generally valid. However, they could be also useful to other developers and, therefore, they are included in this public report.

Each topic starts with a description of the topic followed by the requirements that belong to this topic.

Editing: Requirements about editing and modifications to the blog content by the content providers themselves on the platform.

- FR1 – Deletion by the blog author

Personalization: Requirements concerning features offered to end users of the repository to personalize the way the use and experience the platform.

- FR2 – Capturing filter
- FR10 – Bookmarking for blog posts
- FR12 – Notification about changes in the archive
- FR20 – Favorite list of blogs and topics
- FR22 – Summaries/Journals about new archive content
- FR24 – User specific collections/projects
- FR28 – Recommend a cluster of blogs according to user preferences
- FR45 – Personalized filtering services
- UI3 – History of own activities in the archive
- UI12 – Annotations and highlighting
- UI13 – Customizable user dashboard
- UI25 – Filtered, personalized aggregation of content for end–users
- UI29 – Multiple language support
- UI33 – User profiles

Statistics and analysis: Requirements related to statistics and data analysis of the blog content.

- FR3 – Descriptive statistics for the archive
- FR5 – Descriptive statistics for a single blog or blog post
- FR18 – Analyze the network structure of blogs
- FR21 – Sentiments analysis on blog post level
- FR34 – Topic/Subject detection
- FR35 – Detection and ranking of the originality
- FR47 – Data integrity
- DR9 – Connections/Links
- DR15 – Visits of blogs and blog posts

Output & export: Requirements concerning the platform's export and output methods.

- FR4 – Blog export
- FR9 – Content Translation
- FR17 – Metadata for blogs
- EI1 – API for external clients to query data
- EI2 – Data access/export as XML
- EI4 – Accessible via search machines
- EI5 – Export as CSV
- EI6 – Export links between blog content
- IR3 – Export data using OAI–PMH protocol and Dublin Core schema
- IR4 – Expose parts of the archive via OAI–PMH based on specified criteria.
- IR5 – Connection with federated search engine dbwiz
- IR6 – Facilities to enable interoperability
- IR7 – OpenURL support
- IR8 – Digital Object Identifier
- SM3 – Data export for migration

License: Requirements about detecting, preserving and enforcing digital rights and authority information on the blog content.

- FR6 – Processing of Licenses
- FR39 – Digital Rights Management
- DR1 – Rights and Licenses
- DR3 – Disclaimer
- LR1 – Copyright laws
- LR2 – Privacy laws
- LR3 – Additional national laws
- LR4 – License of the content
- LR5 – Open source software license is preferable

Searching, indexing & ranking: Requirements related to the way blog content is searched, indexed and displayed in the platform.

- FR13 – Keyword/metadata search
- FR14 – Full–text search
- FR16 – Search by author
- FR26 – Context–sensitive search by keyword
- FR27 – Ranking of blogs and blog posts
- FR36 – Meme tracking and trend detection
- FR38 – Multidimensional indexing
- FR44 – Advanced searching
- DR16 – Search key words
- UI6 – Latest posts
- UI15 – Search interface
- UI18 – Search in external source

Input & ingestion: Requirements concerning the platform's input and ingestion methods.

- FR15 – Selection of blogs to archive
- FR32 – Add user suggested blogs to the archive
- FR41 – Retrieving semi-structured information
- FR48 – Crawler/Spider support platform flexibility
- FR49 – Support different versions of blogging software
- FR54 – What to archive: Text and comments
- IR2 – Capturing is possible for various platforms

Community and social features: Requirements about functionality that uses information about users (origin, language, feedback, what they like, what they read, etc.) and functionalities that use social networks to disseminate information about the archived blogs.

- FR7 – User dissemination channels for blog post
- FR11 – Recommendation System
- FR19 – Sharing and collaboration
- FR33 – Dissemination of newly archived items in external social platforms (ex. Twitter) in connection with author profiles
- DR10 – User Annotations
- DR20 – Blog comments from several source
- UI4 – Subscribe and navigate activities of other users
- UI20 – Display blogs that were read by people who have read a specific blog
- UI30 – Creation of a community of providers and recipients within the archive platform
- UI31 – Ranking of archived posts
- UI32 – Tagging system

Display: Requirements related to the way the platform displays content and information.

- FR8 – Topics (Categories) for blogs and blog posts
- FR37 – Web portal
- FR43 – Access to content in a harmonized way
- FR55 – Universal login & central login
- DR4 – Author of the blog, blog post, comment
- DR5 – Tags of the blog or blog post
- DR9 – Connections/Links
- DR11 – Differentiate between blog and blog post
- DR19 – Distinguish institutional/corporate blogs from personal blogs
- DR20 – Blog comments from several source
- UI1 – Web interface
- UI2 – Magazine/journal view
- UI5 – Citation is presented prominently
- UI7 – Tags for blogs and posts
- UI8 – Overview with metadata and summary
- UI9 – Network view on topics, blogs, posts, authors, etc.
- UI11 – historical/chronological view on a blog
- UI14 – User interface for mobiles
- UI16 – Easy to learn/Intuitive
- UI17 – Display references (links) to other sources inside or outside the archive
- UI19 – Display similar blogs and posts
- UI21 – Archived content is clearly stated as such
- UI22 – Density of displayed information
- UI23 – Categories/Topics are shown in different tabs
- UI24 – Display with only core information
- UI26 – Historical/Chronological view on blogs combined with corresponding statistics
- UI27 – Dynamic network view on topics, blogs, posts, etc.
- UI28 – Integration/Combination with other systems
- UI29 – Multiple language support
- UI31 – Ranking of archived posts
- UI32 – Tagging system
- UI34 – Simple submission by authors

Administrative: Requirements defining actions to be made in order to configure the system or to maintain the data quality.

- FR23 – Detection of duplicates
- FR25 – Paid access / Billing system
- FR29 – Detect and remove spam
- FR30 – Extract bibliographic metadata from blog contents
- FR31 – Define important blogs and filter junk
- FR47 – Data integrity
- DR2 – URI and metadata for referencing/citing
- DR18 – Remove private data of archive users
- UI10 – Available services depend on the content rights
- UI35 – Workflow to manage blog submissions
- EI3 – Pingback, Trackback
- IR1 – Single Sign On/Interoperates with Authentication System especially LDAP
- IR7 – OpenURL support
- IR8 – Digital Object Identifier
- RA1 – Recovery of the system
- RA2 – Correct information in the archive

Metadata: Requirements stating what kind of metadata should be extracted and stored in the platform about blogs and blog posts.

- DR6 – Metadata for captured content
- DR7 – Date/Timestamp for Creation and Capturing
- DR8 – Aggregators & Portals
- DR12 – Demographics
- DR13 – Comments
- DR14 – Embedded objects
- DR16 – Metadata for blogs
- DR22 – METS
- OP1 – Versioning
- OP2 – OAIS

Ungrouped (Technical): Requirements related to the purely technical aspects of the platform.

- Capacity and Scalability Requirements: CS*
- Speed/Performance Requirements: PR*
- RA3 – Application High Availability
- Storage and Persistence Requirements: SP*
- Supportability and Maintainability Requirements: SM*
- Security Requirements: SR*

8 Requirement Specifications by Importance

Each requirement has been rated with an importance level. The levels essential, recommended, and optional indicate the importance for the interviewee or rather the stakeholder group. The importance levels should facilitate – together with an effort calculation – design decisions in the following project phases. Therefore, an additional view on the requirements is presented in this chapter that ordered the requirements by their level of importance.

Essential Requirements:

From a stakeholder perspective, the software would not be acceptable unless these requirements are provided in an agreed manner.

FR1 – Deletion by the blog author
FR2 – Capturing filter
FR6 – Processing of Licenses
FR13 – Keyword/Metadata search
FR14 – Full–text search
FR15 – Selection of blogs to archive
FR16 – Search by author
FR36 – Meme tracking and trend detection
FR37 – Web portal
FR38 – Multidimensional indexing
FR39 – Digital Rights Management
FR40 – Billing system
FR41 – Retrieving semi-structured information
FR42 – Weblog content validation and spam filtering
FR43 – Access to content in a harmonized way
FR44 – Advanced searching
FR45 – Personalized filtering services
FR46 – Internationalization
FR47 – Data integrity
FR48 – Crawler/Spider Support Platform Flexibility
FR49 – Support Different Versions of Blogging Software
FR51 – UTF-8 – Default Character Encoding
FR54 – What to archive: Text and comments
DR1 – Rights and Licenses
DR2 – URI and metadata for referencing/citing
DR3 – Disclaimer
DR4 – Author of the blog, blog post, comment
DR7 – Date/Timestamp for Creation and Capturing
DR9 – Connections/Links
DR11 – Differentiate between blog and blog post
DR13 – Comments
DR21 – Long term digital preservation
DR22 – METS
UI1 – Web Interface
UI5 – Citation is presented prominently
UI6 – Latest posts
UI7 – Tags for blogs and blog posts
UI11 – Historical/Chronological view on a blog
UI15 – Search interface
UI16 – Easy to learn/Intuitive
CS1 – Amount of archived blogs
CS2 – Amount of blog posts per day
CS3 – Amount of users
CS4 – Clustering and high availability architectures
CS5 - Modularity
IR2 – Capturing is possible for various platforms
IR3 – Export data using OAI–PMH protocol and Dublin Core schema
IR6 – Facilities to enable interoperability
IR7 – OpenURL support
IR8 – Digital Object Identifier
PR1 – Amount of blog posts to capture
PR2 – Storage data concurrently
PR3 – Real time archiving

continued on next page...

RA2 – Correct information in the archive

SP2 – Mechanisms to avoid data loss
SM2 – Software updates
OP1 – Versioning
OP2 – OAIS
OP3 – APIs for developers
SR1 – Passwords are stored encrypted
LR1 – Copyright laws
LR2 – Privacy laws
LR3 – Additional national laws
LR4 – License of the content

Recommended Requirements:

From a stakeholder perspective, these are requirements that would enhance the software product, but would not make it unacceptable if they are absent.

FR3 – Descriptive statistics for the archive
FR4 – Blog export
FR5 – Descriptive statistics for a single blog or blog post
FR7 – User dissemination channels for blog post
FR8 – Topics (Categories) for blogs and blog posts
FR10 – Bookmarking of blog posts
FR11 – Recommendation system
FR12 – Notification about changes in the archive
FR17 – Print/Export as PDF, JPEG, etc.
FR18 – Analyze the network structure of blogs
FR20 – Favorite list of blogs and topics
FR22 – Summaries/Journals about new archive content
FR23 – Detection of duplicates
FR26 – Context–sensitive search by keyword
FR27 – Ranking of blogs and blog posts
FR28 – Recommend a cluster of blogs according to user preferences
FR29 – Detect and remove spam
FR30 – Extract bibliographic metadata from blog contents
FR32 – Add user suggested blogs to the archive
FR50 – Most important content type (Text/Html)
DR5 – Tags of the blog or blog post
DR6 – Metadata for captured contents
DR8 – Aggregators & Portals
DR10 – User Annotations
DR14 – Embedded objects
DR15 – Visits of blogs and blog posts
DR16 – Search key words
DR17 – Metadata for blogs
DR18 – Remove private data of archive users
DR20 – Blog comments from several source
DR23 – Mashup activities
UI2 – Magazine/Journal view
UI3 – History of own activities in the archive
UI4 – Subscribe and Navigate activities of other users
UI8 – Overview with metadata and summary
UI9 – Network view on topics, blogs, posts, authors, etc.
UI10 – Available services depend on the content rights
UI12 – Annotations and Highlighting
UI13 – Customizable user dashboard
UI17 – Display references (links) to other sources inside or outside the archive
UI19 – Display similar blogs and posts
UI20 – Display blogs that were read by people who have read a specific blog
UI21 – Archived content is clearly stated as such
UI22 – Density of displayed information
UI23 – Categories/Topics are shown in different tabs
UI24 – Display with only core information
UI27 – Dynamic network view on topics, blogs, posts, etc.
UI29 – Multiple language support
UI30 – Creation of a community of providers and recipients within the archive platform
UI31 – Ranking of archived posts
UI32 – Tagging system
UI33 – User profiles
UI34 – Simple submission by authors
UI35 – Workflow to manage blog submissions
EI1 – API for external clients to query data
EI2 – Data access/export as XML
EI3 – Pingback, Trackback
EI4 – Accessible via search machines
EI5 – Export as CSV
EI6 – Export links between blog content
IR1 – Single Sign On/Interoperates with Authentication System especially LDAP
IR4 – Expose parts of the archive via OAI–PMH based on specified criteria.

continued on next page…

IR5 – Connection with federated search engine dbwiz
RA3 – Application High Availability
SP1 – Support for different SQL–data bases
SM1 – Migration/Updating without down time
SM3 – Data export for migration
SR2 – Access restricted to IP–range
LR5 – Open source software license is preferable

Optional Requirements:

From the stakeholder perspective, the following requirements represent a class of functions that may or may not be worthwhile. This gives the supplier the opportunity to propose something that exceeds the SRS.

FR9 – Content translation

FR19 – Sharing and collaboration

FR21 – Sentiments analysis on blog post level

FR24 – User specific collections/projects

FR25 – Paid access/Billing system

FR31 – Define important blogs and filter junk

FR33 – Dissemination of newly archived items in external social platforms (ex. Twitter) in connection with author profiles

FR34 – Topic/Subject detection

FR35 – Detection and ranking of the originality

FR52 – Sharing and promotion beyond the archive

FR53 – Snapshots versions of blog designs in the archive

FR55 – Universal login & central login

DR12 – Demographics

DR19 – Distinguish institutional/corporate blogs from personal blogs

UI14 – User interface for mobiles

UI17 – Display references (links) to other sources inside or outside the archive

UI18 – Search in external sources

UI25 – Filtered, personalized aggregation of content for end–users

UI26 – Historical/Chronological view on blogs combined with corresponding statistics

UI28 – Integration/Combination with other systems
RA1 – Recovery of the system

9 Conclusion

This report specifies requirements for the BlogForever blog aggregation, preservation, management and dissemination platform. The specifications are based on existing data and documents as well as qualitative data captured in 26 semi-structured interviews.

Overall, 153 requirements are described above. Thereby, 114 requirements were identified through interviews. For the interview conduction, eight relevant stakeholder groups were identified. Thereby, blog authors are the most important group because it is crucial that they agree in the archiving of their content. Therefore, the group of blog authors is higher represented in the sample of the 26 interviewees.

Different project members conducted the interviews in several languages (e.g. English, Greek, German). Thus, perspectives are probably less biased by national or cultural characteristics. The interviewers were unequally experienced in performing interviews. Hence, a comprehensive interviewer guide was developed and provided for the interviewers. The included interview schedule consists of questions that were adjusted to the particular stakeholder group.

Due to the different interview languages, the interview transcripts were analysed by several project members. The analysis followed an explorative and incremental process. The requirements were described in a standardised template to ensure a common understanding by the analysts and to facilitate later exploration and usage by developers. The resulting requirements from the different analysts were integrated in a collaborative process. Additionally, the requirements were modelled in the unified modelling language (UML) based on the verbal descriptions to further enhance the usability for developers (e.g. in the design phase).

Next to the requirements from the interviews, 23 requirements were collected from the description of work and 16 from the deliverable D2.1 Weblogs Survey Report. They were also described in the standardised template and modelled with UML.

The initial descriptions and models of the requirements are ordered in requirement categories, which were already used for developing interview questions and analysing the interviews. Furthermore, two views are provided that facilitate exploration and one view structures the requirements with regard to different topics of the software. Thereby, the requirements in a topic are connected or interdependent, e.g. they could share a common purpose like searching. Thus, the creation of project or focus groups, which would further work on the requirements of a topic, is facilitated.

The second view ordered the requirements by their importance for the users/stakeholders. Thus, decision processes about the implementation are better supported. For example, the estimation of implementation effort could be focussed in a first step on essential requirements.

Overall, a comprehensive list of requirements has been created. As opposed to a contractual requirements specification, this list should not be understood as a mandatory guideline for assessing the implemented system. Due to the explorative and innovative character of the project, it is rather a broad and deep insight in the examined area of blog preservation. Thus, it should facilitate further focusing on project aims and guide to successful case studies and prototypes. The latter will be used to identify new or still overseen demands.

10 Acknowledgements

Many thanks to the 26 interviewees who voluntary spend their time and provided their knowledge and experiences in the interviews. Additionally, many thanks for the contributions of all the interviewers, transcribers, and analysts that gave us the possibility of an international survey.

Special thank to Stephen Downes who provided interesting insights in the perspective of a long time blog author. The full transcription of his interview can be found on his blog: http://halfanhour.blogspot.com/2011/11/this-is-interview-conducted-by-karen.html.

Furthermore, special thank to Carolyn Hank who supported us with her methodical knowledge.

11 References

Anon., 2005. *Data Requirement*. [Online] Available at: http://www.opfro.org/index.html?Components/WorkProducts/RequirementsSet/Requirements/DataRequirements.html~Contents [Accessed 22 September 2011].

Fitzner, D., 2010. *D8.1 Requirements specification of the TELEIOS User Community*. Fraunhofer IGD.

Freeman, R.E. & Reed, D.L., 1983. Stockholders and stakeholders: A new perspective on corporate governance. *California Management Review*, 25(3), pp.88-106.

Glaser, B.G. & Strauss, A.L., 1967. *The discovery of grounded theory: Strategies for qualitative research*. New Brunswick (USA) and London (UK): AldineTransaction.

IEEE Computer Society, 2009. *IEEE Recommended Practice for Software Requirements SpeciÞcations*. IEEE Std 830™-1998 (R2009). New York, NY: Software Engineering Standards Committee.

McEwen, S., 2004. *Requirements: An introduction*. [Online] Available at: http://www.ibm.com/developerworks/rational/library/4166.html [Accessed 22 September 2011].

Mitchel, R.K., Agle, B.R. & Wood, D.J., 1997. Toward a theory of stakeholder identification and salience: Defining the principle of who and what really counts. *Academy of Management Review*, 22(4), pp.853-86.

The Open Group, 2009. *Interoperability Requirements*. [Online] Available at: http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap29.html [Accessed 22 September 2011].

A. Appendix A – Interviewer Guides

The following chapters contain the components of the six interviewer guides.

A.1 Pre-Test Instructions for Interviewers

Conduct at least one pre-test interview to ensure that

- The interviewer has fully understood the interview questions and can explain the questions to the interviewee,
- The interviewer is familiar with the schedule of the interview, and
- There are not any technical problems with the recordings.

The pre-test interview can be done with a colleague, friend or any other person but the circumstances of the interview should be as realistic as possible. Therefore, use the same technology (e.g. Skype) that you will use in the interview.

Take notes for any problem that occurs! Report the problems to Hendrik (email[5]), Nikos (email5) or Jaime (email5) to get clarification.

Reporting of problems will help us as well to avoid the same problems for other interviewers. Thanks.

[5]Email-address was made anonymously for this report.

A.2 Interview instruction and assistance

In the following, we give some instructions and assistance to conduct a proper interview. Read it carefully BEFORE the interview!

The interview questions are provided in English but you should do the interview in the language that is most familiar to the interviewee and you. Therefore, you may have to translate the questions into your language. Ask Hendrik (email5), Nikos (email5) or Jaime (email5) for help if you have any problems to translate the sense of the questions.

How should you lead the interview?

- The interviewee should do the most of the talking. Therefore, interrupt the interviewee only if you feel that the talk goes into a topic that is not relevant to the interview purposes!
- Respect the interviewee's pace and do not be afraid of pauses or silences!
- Keep the interview focused on the topics of the questions previously defined!
- Ask for more clarification or elaboration if you feel that the interviewee could tell you more on a topic.
- Also, allow the interviewee to ask questions, as well, if more clarification is needed. Respond to the interviewee's questions, confirming that your response was satisfactory (e.g., is that what you had in mind? Or, was that helpful? Or, did that answer your question?)
- Pay attention to the language you use, avoiding jargon or terms the interviewees may not be familiar with. You should adopt the interviewee's worldview, rather than expect the interviewee to adopt yours.
- Keep an eye on the time limits for questions!
- Remember your objectives throughout the interview. While an interview may mimic a natural conversation, it is not. It is a strategic and deliberate data collection strategy.
- Be authentic, acting relaxed and natural. Even though you have practiced the schedule, do not give the interviewee the impression you are reading from a script. Ask your questions in such a way that each seems a natural extension of the one preceding it, and as if the question just came to you rather than having been vetted and prepared and practices well in advance of the interview.
- Be an engaged, authentic listener. Just don't simply leap into the next question once the interviewee has responded to the preceding one. Present yourself as being aware, interested and attentive to what is being said. You might say, "Oh, that is really interesting," or, "Thanks for that detailed response," before moving on to the next question."
- Interview in a comfortable, private place and one where you will be free from distractions and interruptions. If conducting the interview from work, consider hanging a sign on your door, "interview in progress" so people don't knock or pop their heads in your office. If conducting the interview from home, stay clear of barking dogs (if a pet owner) and keep the TV or radio off.
- Be respectful, appreciative and kind.

How should you ask to get more information for a topic?

- Avoid suggesting any answers! (e.g. "I assume that you do it in that way….")

- Ask open questions! Avoid closed questions (e.g. Yes/No questions)!
- If an interviewee provides a Yes/No response, prompt them to give a more detailed answer through using probing questions (e.g., Thanks for your reply; could you tell me a little about why you responded [yes/no]?" Or, "How come?").
- Ask clear and direct questions such as How? Where? When? Who? What? Why? How much? How many?

What are the steps of your interview?

1. Read the consent script to the interviewee to inform her/him about the interview and to get her/his consent for recording the interview!
2. Start the recording! (And do a test of the equipment. This relaxes both parties and ensures that you get a good recording.)
3. Ask the questions of the interview schedule! You should make every attempt to ask every question on the schedule. You can improvise slightly, using prompts, as necessary, to try to solicit a response. However, the interviewee may not want to answer every question. That's okay. You can skip a question and move on to the next one.
4. Try to stick to the order of the questions on the schedule. However, if from the interviewee's responses you think the interview would be better served by changing the flow of questions, you may do so. For example, the interviewee may raise a point that you will come back to in a question scheduled to be asked later in the interview. When appropriate (that is, when it works with rather than impedes the flow of the interview) you may reorder the questions during the interview. However, this should be done sparingly and only when appropriate.
5. When you have completed the schedule, ask the interviewee if s/he have any questions for you. If so, respond. After answering, ask if they have any more questions. Confirm they have no more questions before you conclude the interview. If no more questions (or if no questions to start with), let them know the interview is over (e.g., "that concludes the interview").
6. Before ending the interview, be sure to thank the interviewee for her/his participation! Don't just simply say thanks and hang up. Let them know you appreciate them taking time out of their day to talk with you and that you value their contributions to the research study.
7. Store the recording in a safe place!
8. Fill the interview debriefing sheet!
9. Inform Hendrik (email[6]), Nikos (email6) or Jaime (email6) that the interview is done.

[6]Email-address was made anonymously for this report.

A.3 Consent script

Each interviewee has to be informed about the interview conditions and has to declare her/his consent.

Hello [Name of the Interviewee], it's [Name of the Interviewer], calling for our scheduled phone interview. How are you today? **(Wait for reply).**

Good to hear. Before we begin, I first wanted to thank you for agreeing to participate in this important interview. Also, I have a prepared consent script as I will need to get your consent to go forward with the interview. May I read you the consent script now? **(Wait for reply)**.

Thank you. As you may already know, I am member of the EU-funded project BlogForever. In the project, we aim on the development of an archiving and preservation system for weblogs. We would like to address various needs and therefore, we investigating different perspectives with questionnaires, interviews and other research methods.

This interview will last approximately 30 minutes. Your participation in this interview is completely voluntary. This means that you do not have to participate in this interview unless you want to. You don't have to answer any question that you choose not to answer. We will just skip that question and go on to the next one.

I or any other member of the BlogForever team will not identify you, your blog, or your institution, or use any information you share during the interview that would make it possible for anyone to identify you in any presentation or written report about this study. I will not even record your name with your responses. If it is okay with you, I might want to use direct quotes from you, but these would only be quoted as coming from a person of a certain label or title, like "one blogger from Germany."

There are no other expected risks to you for helping us with this study. There is no financial benefit for your participation. The cost to participate is your time, and for that I am very appreciative.

We offer you that we will send you the results of our study when finalized. To make sure that your email address will not be related to this interview, you can write an email to *email*[7] or to *email*[7] to announce your interest in the results. Alternatively, you can give your email address to me and I will inform them to send you the results. I will not store your email address related to this interview.

I would like to record this interview. The digital recording will be maintained by me, with only access by BlogForever project members who are involved in the analysis of the interviews. The recording will be permanently deleted after completion of our analysis. Further, if a transcript is made, in no way will your name be associated with the transcript. Any identifying personal information received in the course of this interview will be de-identified. If a transcript of our interview is produced, only BlogForever project members who are involved in the analysis of the interviews will have access to the transcript.

Lastly, you may contact as well the management of the BlogForever project, anonymously if you wish, with any questions or concerns,

[7] Email-address was made anonymously for this report.

Thank you. At this point, do you have any questions for me? [**Respond to any inquiries**].

Do I have your permission to begin recording and asking you questions? **(If yes, continue)**

> **If No:** We can still complete the interview even if I cannot record you. Do you give me permission to take notes during our interview? These notes may include direct quotes from you. Again, in no way will these notes be associated with you in any way when reporting findings from this study. (**If yes, continue**)
>
> **If No to Note-Taking**: Thank you. I very much appreciate the time you took to talk with me today and for your past participation in the study. This ends the interview. At this point, can I answer any questions for you? [**Respond to any inquiries**]. Again, thank you for your time. [**End call**].

A.4 Interview Schedule – Content Provider – Blog author

Part A) *(approx. 5 min)*

1. Give a brief description of BlogForever and clearly state to the interviewee the purpose of the interview

Question	Hints for the Interviewer
First, we would like to understand your background. This will help us to understand the context of your answers. 1. Could you please tell us a bit about your blogging experiences and why are you blogging?	It should be covered at least • how long, • how often, • about what, • to whom.

Part B)

Mandatory Questions	Hints for the Interviewer
2. How do you facilitate or prevent that your blog will be found by other people	e.g. the use specific key words, tags, dissemination activities
3. Who has the right to do what with your blog content or any data from blog? How do you indicate and control the rights for your content?	e.g. common creative license, DRM
4. How interested are you in the possible interconnections between your blog and others? Follow up questions: • In what way? • How would you imagine a system supporting this feature?	The follow up questions should be used if they are not already covered by the answer to the main question.
5. In a platform where you browse and search for blogs and the relations between them what would make the user interface comfortable and intuitive? Give us a couple of examples.	The question aims on User interfaces and Look & Feel
6. How interested are you in how your blog is ranked among blogs for the different subjects and how do you check that? 7. By what other criteria would you like to see your blog ranked?	The blogger might be interested in specific fields on which to rank their blog by.
8. What would you like to know about your visitors or other analysis of your blog? For example: statistics (popularity, visits, etc.), keywords or sentiments. • And why?	Analysis of part of the blog to extract keywords, sentiment, etc. What would be extracted? Why?

9. How do you archive or backing up your blog(s)? **If yes:** a. Describe the process of archiving or backing up the blog(s) you are authoring. b. Describe the process of accessing or restoring information from your archive. c. Can you identify any problems/issues with the procedures you are currently following? **If no:** 10. What is keeping you from archiving your blog(s)? Please elaborate! Possible follow up question: • Are there any constraints (e.g. technical) or you just didn't care about it?	
11. If there was a real time, continuous and viewable archive of your blog, how would you use it? What would it look like?	The respondent might talk about desired functions or visual requirements.
12. If there would be a preservation or archiving system for blogs how would you like to control which of your content is captured and stored? Follow up questions: • Would you like to be notified and if yes, how? • How would you like to indicate that content should be removed from the preservation system?	The follow up questions should be used if they are not already covered by the answer to the main question.

Optional Questions	**Hints for the Interviewer**
13. How do you facilitate or prevent technically that your blog will be found and disseminated by search engines?	The blogger can use for example special code in the html of the blog to make it discoverable, or they can list it in blog directories, etc.
14. How do you facilitate the readers of your blog that they find related posts inside your blog?	Try to figure out how the blogger approaches e.g. • Tags: How do decide which tags are used? When do you use new tags and do you re-tag you older posts? • Categories: How do you developed the categories? How often and why do you change the categories?

15. If there would be a preservation or archiving system for blogs and if there would be a special access or interface for blog authors how should it work?	
16. What other general comments do you have on the development of a blog aggregation, preservation, and management & dissemination software?	

A.5 Interview Schedule – Content Provider – Organisation

Part A) *(approx. 5 min)*

1. Give a brief description of BlogForever and clearly state to the interviewee the purpose of the interview

Question	**Hints for the Interviewer**
First, we would like to understand your background. This will help us to understand the context of your answers. 1. Could you please tell us about your position in your organization?	
2. What are the main services or products that your organization offers to whom and how are they represented in your blogs?	
3. Could you please tell us about your experiences with blogs (professional or private)?	It should be covered at least: • how many years • how often • are they only using them for their company

Part B)

Mandatory Questions	**Hints for the Interviewer**
4. How do you facilitate the dissemination of the blogs in your organisation? 5. Is there a difference how you disseminate them inside the organization or to the public? Please elaborate.	There could be organisational as well as technical aspects. In case of technical aspects, listing them will suffice.
6. What recommendations or constraints do you have for the appearance or the layout of the blogs in your organisation? Follow up question: • How do you control it and how would that affect a blog archiving service in your institution?	e.g. corporate identity
7. How do you control the content in the blogs to avoid inappropriate or illegal behaviour? Follow up question: • Have you ever had to remove content from an already posted blog post (or the whole blog) and how did you do it?	

8. What are you interested in the discussions and comments happening in your blogs? Follow up question: • Would you like to aggregate, manage & analyse all your users' feedback? Please describe an ideal scenario.	
9. How do you archive or backing up the blogs of your organisation? **If yes:** a. Describe the process of archiving or backing up the blogs? b. Describe the process of accessing or restoring information from your archive? c. Did you ever need to search, filter, export or migrate archived blogs? How did you do it? d. Can you identify any problems/issues with the procedures you are currently following? **If no:** a. Why don't you use a special software capable of aggregating, archiving & managing all your blogs? Follow up question: • What are important aspects that such software has to fulfil?	
10. If there was a real time, continuous and viewable archive of your blog, how would you use it? • What would it look like?	The respondent might talk about desired functions or visual requirements.
11. How interested are you in possible interconnections between your blog and others? Follow up question: • In what way? • How would you imagine a system supporting this feature?	Blog relations might include the use of semantics to indicate these relations. If needed, record the type of semantics used / desired.
12. Would you be interested in an analysis of your blog (or part of your blog) to extract for example: statistics (popularity, visits, etc.), keywords, and sentiments and why?	Analysis of part of the blog to extract keywords, sentiment, etc. What would be extracted? Why?

Optional Questions	Hints for the Interviewer
13. What would you describe as a “killer” application feature for a blog archiving service?	
14. Do you have any general comments on the development of a blog aggregation, preservation, and management & dissemination software?	

A.6 Interview Schedule – Content Retriever – Blog Reader

Part A) *(approx. 5 min)*

1. Give a brief description of BlogForever and clearly state to the interviewee the purpose of the interview

Question	**Hints for the Interviewer**
First, we would like to understand your background. This will help us to understand the context of your answers. 1. Could you please tell us a bit about your experiences with blogs and why do you read blogs?	It should be covered at least • how long, • how often, • about what, • why.
2. Do you favour the information in blogs over the information from other websites? **If yes:** Why? **If no:** Why not?	Short explanation is sufficient.

Part B)

Questions	**Hints for the Interviewer**
3. What ways are you using to access blogs today? Follow up questions: • How do you explore relationships between blogs, blog posts or bloggers? • What do you miss for your exploring or reading activities? • Do you use any blog aggregators or blog search machines? **If yes:** • What do you use or like on this platforms? • What do you miss?	
4. What personal list(s) of blogs or topics are you monitoring? **If yes**: • How are you managing it? • How do you discover new blogs or topics for that list?	
5. Do you follow entire blogs or choose the specific	

content you are interested in and why? Follow up questions: • How would you filter such content? • How do you imagine such a feature?	
6. How are you archiving blog content you find useful, important or interesting? Follow up questions: • How are you browsing, searching & accessing archived information?	
7. If there would be a preservation or archiving system for blogs what benefits would you expect by exploring such system instead of the original blog? Follow up questions: • How would you explore it? • How would you like to be informed about updates in such a system?	
8. If there would be a preservation or archiving system for blogs where you browse and search for blogs and the relations between them what would make the user interface comfortable and intuitive? Give us some examples.	The question aims on User interface and Look & Feel.
9. How would you browse or retrieve blogs by ranking? **If yes**: • What ranking criteria would you like to use? • Which ranking methods do you know and which would you prefer?	
10. How would you use sub categorized clusters to browse or search the blogs in a thematic archive?	

A.7 Interview Schedule – Content Retriever – Libraries

Part A) *(approx. 5 min)*

1. Give a brief description of BlogForever and clearly state to the interviewee the purpose of the interview

Question	**Hints for the Interviewer**
First, we would like to understand your background. This will help us to understand the context of your answers. 1. Could you please tell us about your position in your organization?	
2. What are the main services or products that your organization offers to whom?	
3. Could you please tell us about your experiences with blogs (professional or private)?	It should be covered at least: • how many years • how often • for what purpose

Part B)

Questions	**Hints for the Interviewer**
4. How are you collecting and using blog content for the benefit of your audience? **If yes:** e. How are you doing it? i. How do you choose what to collect? ii. How do you keep that content? f. What access, search, sort or retrieval facilities do you provide to your audience? g. Can you identify any problems or issues with the procedures you are currently following? h. Can you identify any best practices or success stories regarding your blogs collection? **If no:** a. Would you use a special software capable of aggregating, archiving & managing blogs of your academic interest? Follow up question: • What would be the important aspects that such	

software would have to fulfil?	
5. How do you integrate external repositories with digital content into your services? Follow up question: • What are the challenges for such integration?	E.g. standards, protocols, necessary meta data, interface, etc. A list of technologies will suffice
6. What legal challenges do you face when you access or integrate external content and provide it to your users?	
7. When you consider integrating an external repository into your services, how do you assess or control the quality and performance of the external repository?	
8. What kind of access restriction policy do you apply to your users?	e.g. blocking off-campus IP address, LDAP authentication, etc. A list of technologies will suffice
9. What personalized services do you provide to your users and what do they prefer?	e.g. custom searches, bookmarks, alerts, etc.
10. If there would be a preservation or archiving system for blogs what ways would you use to access and retrieve content from such a platform? Follow up question: • Do you have preference in specific schemas? What schema and why?	

A.8 Interview Schedule – Content Retriever – Businesses

Part A) *(approx. 5 min)*

1. Give a brief description of BlogForever and clearly state to the interviewee the purpose of the interview

Question	**Hints for the Interviewer**
First, we would like to understand your background. This will help us to understand the context of your answers. 1. Could you please tell us about your position in your organization?	
2. What are the main services or products that your business offers to whom?	
3. Could you please tell us about your experiences with blogs (professional or private)?	It should be covered at least: • how many years • how often • for what purpose

Part B)

Questions	**Hints for the Interviewer**
4. How are you collecting and using blog content for the benefit of your clients? **If yes:** a. Why and how are you doing it? iii. How do you choose what to collect? iv. How do you keep that content? b. What access, search, sort or retrieval facilities do you provide to your audience? c. Can you identify any problems or issues with the procedures you are currently following? d. Can you identify any best practices or success stories regarding blogs of your business interest? e. Could you please estimate how many blogs and how often you collect (at present and in the future)? • What technical challenges does this imply? f. Is there one or more persons in your company, a service or both responsible of collecting, managing	Question "If yes: … e." aims on requirements for capacity and scalability.

and analysing the blog contents? • If you should classify them as simple users or techies, how would you describe them and why? **If no:** a. Would you use a special software capable of aggregating, archiving & managing blogs of your business interest? b. What blogs do you plan to collect and why? c. Could you please estimate how many blogs and how often you would collect? o What technical challenges does this imply? d. What parts of the blog content analysis would you focus on? e. How do you feel your clients could benefit from such a platform?	Question "If no: … c." aims on requirements for capacity and scalability.
5. What legal challenges do you face when you access or integrate external content and provide it to your users?	
6. What kind of access restriction policy do you apply to your users?	e.g. blocking off-company IP address, LDAP authentication, etc. A list of technologies will suffice.
For the following questions, assume that there would be a software or system capable of aggregating, archiving & managing blogs…	
7. What federated search service do you have in your organization? 8. What kind of protocols and standards would be required to ensure interoperability with such system?	Federated search is an information retrieval technology that allows the simultaneous search of multiple searchable resources. Listing of protocols and standards suffices.
9. Would you be interested in monitoring blogs covering specific topics? Please elaborate.	
10. What personalized services would you or do you offer your users and clients and why?	e.g. custom searches, bookmarks, alerts, etc.
11. Would you offer a different interface and functionality to your users within the company and your business clients? • What are the main aspects in which they would differ?	
12. Would you be interested in offering paid services to your	

end users and how would you envision such a system?	
13. How would you offer access or provide information to your end users in such a platform?	e.g. web interface, periodical e-mail digest, etc.

A.9 Interview Schedule – Content Retriever – Researcher

Part A) *(approx. 5 min)*

1. Give a brief description of BlogForever and clearly state to the interviewee the purpose of the interview

Questions	Hints for the Interviewer
First, we would like to understand your background. This will help us to understand the context of your answers. 1. Could you please tell us a bit about your research especially in relation to blogs or social media?	It should be covered at least • how long, • about what,
2. Do you blog or read blogs? **If yes:** Why? **If no:** Why not?	Short explanation is sufficient.

!!! Note for the interviewer: If the interviewee does not research about blogs or social media, proceed with the questions for blog readers or blog authors!!!

Part B)

Questions	Hints for the Interviewer
3. How are you collecting information from blogs for your research? **If yes:** • Describe the purpose and the processes you are following?	Ask the questions for social media instead of blogs if the researcher research about other social media than blogs.
4. What search functionalities, aggregation or analysis have you used to get a proper sample of blogs for your research? Follow up question: • How was it supported by software and how could it be better supported?	
5. How are you managing, searching, retrieving & analysing the collected blog information? • Can you identify any problems or issues with the procedures you are currently following? • Can you identify any benefits or best practices with the procedures you are currently following?	
6. What if there was an online service containing blogs relevant to your research topic. What would you use it for and what should it provide that you could use it efficiently?	

7. Would you install a special software capable of aggregating, archiving & managing blogs relevant to your research? • For what purpose?	
8. If you would use the software to research collaboratively with other researchers how would you use it? • What should the software or platform offer to facilitate collaborative research on blogs?	
9. If there would be a preservation or archiving system for blogs where you browse and search for blogs and the relations between them what would make the user interface comfortable and intuitive? Give us a some examples. • What examples do you know that could be called as good practices?	The questions aims on User interface and Look & Feel.
10. What export functionalities would you expect to process the blog contents in another specific research software? Why?	

A.10 Interview Schedule – Admin – Organisation

Part A) *(approx. 5 min)*

1. Give a brief description of BlogForever and clearly state to the interviewee the purpose of the interview

Questions	Hints for the Interviewer
First, we would like to understand your background. This will help us to understand the context of your answers. 1. Could you please tell us about your position in your organization?	
2. How many people maintain which amount of IT hardware and services in your organization?	Estimations are sufficient. The answer should just help to get an overview over the size and importance of the IT department in the organization.

Part B)

Questions	Hints for the Interviewer
3. Do you have an archive for the blogs of your organization? **If yes:** a. Could you describe the process of archiving the blogs of your organization? b. Could you describe the process of restoring information from your archive? c. Did you ever need to filter the archived blogs? • Why and how did you do it? • What challenges did you face? d. Did you ever need to analyse the archived blogs? • Why and how did you do it? • What challenges did you face? e. Did you ever need to export the archived blogs? • Why and how did you do it? • What challenges did you face? f. Did you ever need to migrate archived blogs? • How did you do it? • What challenges did you face? **if not for blogs:** a. Could you describe the process of archiving	

digital content of your organization? b. Could you describe the process of restoring information from your archive? c. Did you ever need to filter the archived content? • Why and how did you do it? • What challenges did you face? d. Did you ever need to analyse the archived content? • Why and how did you do it? • What challenges did you face? e. Did you ever need to export the archived content? • Why and how did you do it? • What challenges did you face? f. Did you ever need to migrate digital content? • How did you do it? • What challenges did you face?	
4. Could you please estimate and describe how many blogs you are archiving / would consider archiving? Follow up question: • What technical challenges does this imply?	e.g. capacity, scalability
5. How would you like to manage the list of blogs you archive and how often would you expect to change that list?	Meaning the list of blogs that should be archived. How should that list be managed by the platform admin: web interface, database, text file. How often would blogs be added/subtracted/edited in that list.
6. How often would you choose to check and update the archived blogs' status and new posts and why?	If there are dependencies then try to figure them out. E.g. the frequency depends from subject.
7. Are you supporting any APIs/Technologies for digital content interoperability in your platform? • Which and why?	The interviewee should describe the technologies they employ to import/export/handle different types of digital content
8. How many users access your services normally and how many in peak times?	
9. Which characteristics or qualities must software have	e.g. a special software architecture, a special quality of security, a

that you (can) integrate it in your existing IT environment?	special certificate for software quality
10. Which important functionalities do you normally use to monitor and maintain server software and services in your organisation?	
11. Have you defined any service levels for the services that you offer to your users? • Which service levels do you have and why?	e.g. quality or availability assurances

A.11 Interview Schedule – Admin – Blog Host

Part A) *(approx. 5 min)*

1. Give a brief description of BlogForever and clearly state to the interviewee the purpose of the interview

Questions	**Hints for the Interviewer**
First, we would like to understand your background. This will help us to understand the context of your answers. 1. Could you please tell us about your position in your organization?	
2. How many people maintain which amount of IT hardware and services in your organization?	Estimations are sufficient. The answer should just help to get an overview over the size and importance of the IT department in the organization.

Part B)

Questions	**Hints for the Interviewer**
3. What kind of software or platform do you use to publish blogs? If it is a 3rd party platform: • Do you regularly follow the software updates and why/why not?	
4. What kind of archive or backup do you have? **If yes:** a. Could you describe the process of archiving the blogs of your organization? b. Could you describe the process of restoring information from your archive? c. Did you ever need to filter the archived blogs? • Why and how did you do it? • What challenges did you face? d. Did you ever need to analyse the archived blogs? • Why and how did you do it? • What challenges did you face? e. Did you ever need to export the archived blogs? • Why and how did you do it? • What challenges did you face?	In all the following questions the terms archive* and backup* are interchangeable

f. Did you ever need to migrate archived blogs? • How did you do it? • What challenges did you face?	
5. What are or have been your biggest challenges in backup or archiving blogs?	
6. Which APIs/Technologies do you support for digital content interoperability in your platform? • And why?	The interviewee should describe the technologies they employ to import/export/handle different types of digital content
7. Which amount of blogs, blog content and users do you handle now and what do you expect for the future? Follow up question: • What challenges does this imply?	e.g. capacity, scalability
8. What security levels do you have to fulfill for your blog hosting services?	
9. What service levels do you have to fulfill for your blog hosting services?	e.g. quality, availability assurances
10. What do you think are the main advantages and disadvantages of installing a special software capable of aggregating, archiving & managing all your blogs?	

A.12 Interview Results / Interview Debriefing Sheet

The results of each interview should be

- A recording of the interview (preferable as a mp3 file).
- The notes of the interviewer (preferable in a digital form).

Additionally, each interviewer has to fill in the following interview debriefing sheet. The information in the sheet will help to manage the outcomes of the interview and to improve further interviews.

Identification of the Interview
Date of Interview:
Start Time:
End Time:
Interviewer:
Interview ID (Name of the Interviewer + Unique 3-number ID):
Language of the Interview:
How was the interview conducted? (e.g. Skype)

Recording	
Was there a problem with the recording?	( ) Yes ( ) No
If yes, describe:	
What, if any, measures will be taken to correct problem:	
Is the recording still viable?	( ) Yes ( ) No
Was the recording saved digitally and labeled accordingly (mmddyy_interviewID_rec)?	( ) Yes ( ) No

Note-taking	
Were brief notes taken during interview session?	( ) Yes ( ) No
Were brief notes saved digitally and labeled accordingly (mmddyy_interview _prel_notes)?	( ) Yes ( ) No
Were extensive notes taken immediately following the interview session?	( ) Yes ( ) No
Were extensive notes saved digitally and labeled accordingly (mmddyy_interviewID_notes)?	( ) Yes ( ) No

Interview and Schedule	
Were any unexpected topics introduced by subject?	( ) Yes ( ) No
If yes, which of these topics should be considered in future interviews:	
Were there questions that did not work?	( ) Yes ( ) No
If yes, describe:	
How many minutes did it take to complete the interview, from start to finish:	
Was there sufficient time to complete the interview?	( ) Yes ( ) No
If no, please estimate reason why:	

Additional Comments
Please list/describe any additional information regarding this particular interview session, if not covered in previous questions:

A.13 Requirement template

Req. number - Title of the requirement (Template)

Requirement category	The category of the requirement.
Degree of necessity	( )**Essential** ( )**Recommended** ( )**Optional**
Description of the requirement	The description specifies the requirement.
Stakeholder	The stakeholders that mentioned / are concerned by the requirement.
Justification / Foundation	This should explain, why the requirement is required. That could be based e.g. on • Survey data • Phrases from the interview • Literature • etc.
Assessment / Measures	How will we evaluate if the requirement is fullfilled. Example: A user can find the required functionality in less than 5 seconds.
Author(s) of the requirement description	List of the authors who contributed tot he requirement description. That is needed for later queries and clarification. •