

MIRRA (Memory - Identity - Rights in Records – Access) Research Project output:

A participatory recordkeeping application Software Requirements Specification (SRS)

September 2021

Department of Information Studies, University College London

Elizabeth Shepherd: Principal Investigator

Anna Sexton: Co-investigator

Pete Williams: Research Associate

Elizabeth Lomas: Policy lead

OLM Systems

Mark Denton: Group and Systems Product Director

Tanya Marchant: Head of User Experience

Laura Donohue: UX (User Experience) Content Writer

Information

<https://blogs.ucl.ac.uk/mirra/about>

<https://www.olmsystems.com/>

Project funded by the UK Arts and Humanities Research Council. Grant Ref: AH/T005343/1. Grant Title: Memory - Identity - Rights in Records - Access: Embedding Participatory Recordkeeping in Child Social Care.

Contents

INTRODUCTION: BACKGROUND AND CONTEXT	1
THE NATURE OF A SOFTWARE REQUIREMENTS SPECIFICATION	2
THE MIRRA APP: TECHNICAL DETAILS	3
SERVER COMPONENT	3
CLIENT APPLICATION.....	4
INTEGRATION COMPONENT	4
THE MIRRA APP: INTERFACE AND FEATURES	5
GREETING PAGE.....	5
PERSONALISATION.....	6
<i>Posting / creating journals</i>	6
JOURNAL REFLECTION.....	8
ENGAGEMENT AND REWARDS	9
ARCHIVE/DELETE	10

Introduction: background and context

MIRRA (Memory - Identity - Rights in Records – Access) is a major research project being undertaken by researchers at the Department of Information Studies at University College London, phase two of which has been with our commercial partner OLM Systems. Phase one (2017-2019) examined information rights in the context of child social care in England, from the perspective of care-

experienced people who sought access to records of their childhoods at different stages of their lives. Findings showed strong support towards participatory recordkeeping approaches. At present these are oriented towards the needs of service providers, with a focus on risk management, reporting to central government and the needs of social workers. As a result the voices and feelings of looked after children, carers and families are often missing or mis-represented. The research demonstrated that records are a vital resource, especially for care experienced people themselves, in understanding the circumstances of their lives.

It seemed clear that using participatory recordkeeping systems would enable the child in care to be more actively involved in the creation and management of the records that document their lives. This would help to alleviate the disconnection, isolation and disempowerment they face. Such systems would balance the access needs of different record stakeholders whilst also meeting key legislative requirements (such as the Data Protection Act 2018). Issues of privacy, confidentiality, accuracy and conflict could be addressed in transparent and efficient ways. However, the project found a lack of appropriate enabling software on the market to adopt these practices. Thus, with further funding from the Arts and Humanities Research Council, we undertook a second phase of the project (2020-2021), during which we worked with software developers and user experience experts from OLM to focus on potential technology solutions to the problems elicited. The project developed a specification for a participatory digital recordkeeping system for use in child social care settings in the UK. This document represents that specification.

The nature of a Software Requirements Specification

This stage of the software development translates the stakeholder requirements as expressed in the problem statement into a prototype system that seeks to facilitate the expressed needs of those who will either use the system or rely on others to use it in their service. A formal definition is offered by the IEEE (Institute of Electrical and Electronics Engineers) (2011: p4): the SRS is a 'specification for a particular software product, program, or set of programs that performs certain functions in a specific environment.' If this seems little more than a tautology ('a software requirements specification' is a 'specification for a particular software ...'), a somewhat more helpful description (if not 'definition') is that offered by Le Vie (2010) in the technical magazine 'Techwhirl'. In an article for 'technical writers' Le Vie explains that an SRS is 'an organization's understanding (in writing) of a ... client's system requirements and dependencies *at a particular point in time* [original emphasis] (usually) prior to any actual design ... [It] states in precise and explicit language those functions and capabilities a software system ... must provide, as well as [any] constraints by which the system must abide'.

An SRS deals with the following features of the software:

- *Functionality*: What the software is designed to do;
- *External interfaces*: Its interaction with people, hardware and other software;
- *Performance*: Speed, response time, recovery times etc.;
- *Attributes*: Portability, 'maintainability', security, etc.;
- *Design constraints*: Policies for database integrity, resource and other limitations, etc. (IEEE, 2011)

For the MIRRA app, the technical details are detailed first, followed by an account of the interface and the features. The former is of most relevance to app developers, and the latter to potential users. To put this into context first, however, the app was developed after the formulation by the

UCL and OLM team, of the following 'problem statement', in the form of responses to questions such as 'who is the app for?', 'What is the issue or problem needing to be addressed?' etc.:

Who? Looked after children aged 13-17;

What? May not have the tools, experience, or opportunity to express themselves;

When/where? When trying to explain their lives and what they're feeling, what's happening to them, and what's important to them;

Why? Autonomy over their own life story. This is important to children in care, because it gives control over their life story and enables them to speak for themselves.

We put this statement into a 'fuller picture', reasoning that giving the young person some autonomy in adding to their record is also important for the professionals involved in the child's life, enabling them to make better, informed, and child-centred decisions.

In a nutshell, the purpose of the app is to create a safe, simple way for a young person in care to record a digital personalised diary, parts of which could be shared with their social worker/ carers. Another key objective is to give the young person control, and this is done through the app via the personalisation features where customisation can be applied. Lastly, the app uses gamification for engagement as a method to nurture and help the young person in getting the most from it.

The MIRRA App: technical details

The application as designed has two components (the client app and server) and an optional third component (integration to case management). This document provides suggestions on how each component could be built in such a way that it would allow them to be developed and deployed.

Server Component

The server component will handle the storage of the data recorded by young people alongside security information such as usernames and passwords, etc. This would be run in an appropriate cloud environment, and it is suggested that use of a public cloud service such as Microsoft Azure, Google Cloud or Amazon Web Services is sufficient.

The server software must be designed with the 14 Cloud Security Principles in mind, as recommended by the National Cyber Security Centre (NCSC, n.d.) and this will determine some of the design decisions. For example, following these 14 Principles, it would be recommended that each fostering agency or local authority have their own instance of the app to provide data segregation and help with loss prevention. The principles would also say that data exchange between the client app and the server is protected using TLS 2.1 encryption to prevent eavesdropping.

The server will need to present a number of application end points and an application programming interface (API) to the client application. It is recommended that this is provided in GraphQL which is an API technology that allows simple update whilst limiting changes to the server breaking the client application. GraphQL interfaces provide several tools to help client application developers build and test their application. This would allow for the possibility of building different versions of the client app against a single server.

The server software can be developed in any of the common technologies for this purpose such as Java, C#, C++, Ruby on Rails, PHP, JavaScript / Node, etc. The exact choice is up to the developer based on their skills and abilities but all would be suitable. It is recommended that as many of the

cloud based services provided are used such as database-as-a-service as this will reduce the amount of development needed.

Client Application

The client application is the part the young person has on their device and uses. Within client applications there are two choices: native apps and installable progressive web apps. There is no clear way of choosing between them although there are a lot of opinions.

Native apps are developed specifically for each device operating system either Apple iOS or Android. Android has many variants, and a native app will need to take account of these because they impact the look and feel of the application on different devices from different manufacturers. For Apple iOS applications Apple provide Swift which provides an easy-to-use development environment. For Android there is Kotlin provided by Google. However, for both operating systems other options are available and can be considered. Cross platform tools, capable of developing apps for both operating systems are also available. Once developed the app will need to be certified and delivered by an appropriate app store. The exact choice will depend upon the developer based on their skills and abilities.

Progressive web apps are developed once for all devices and work within the browser on the device. They can be installed if the user chooses to do so and then work just like any other app. There are many tools for building progressive web apps including low code tools in addition to common web development technologies such as JavaScript, React and Angular.

For this application the use of progressive web apps (PWA) may be preferable. It will allow the app to work on a range of devices, including laptops, without having to develop multiple versions for each device and potentially a web-based version. With this approach also, companies no longer need to build, maintain, and update separate versions for iOS and Android operating systems. The overall cost of a PWA solution, in comparison to a native app, is lower. This also avoids having to get the application certified for the app store. Whenever a native app is ready, however, it should be added to app stores such as Apple's and Google's.

Integration Component

Integration opportunities between the app and current Case Management Systems (CMS) would also need further research and investigation, but the most viable option would be through an Application Programming Interface (API) (a connection between computers or between computer programs). What would be updated from the app into the CMS and the level of moderation and governance of those entries would need to be decided. The preservation of the app entries within the CMS would be another area that would require significant understanding and action.

The integration component will be needed to integrate to other systems such as case management to share information. Integration could be provided by the same, or similar, API layer used between the server and the application. GraphQL could be used to expose several APIs that would allow a case management system to query the data and re-use it in its own datastore. Access by the case management system must be secure and authenticated and we would again recommend following the Cloud Security Principles. Any provider should be Cyber Essentials Plus certified to show that their operation of the app meets a high standard of resistance to cyber-attacks. Facilities must be provided to meet the needs of GDPR such as the rights of access and erasure.

The MIRRA App: Interface and features

Greeting page

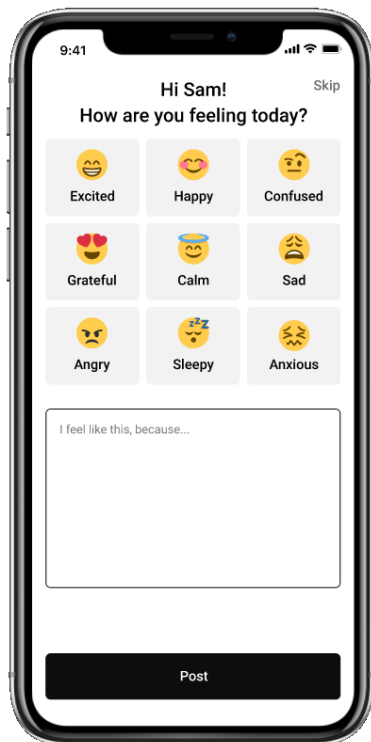


Figure 1: Greetings page

When the young person opens the app, they are greeted with the same question once a day: How are you feeling today? (see Figure 1). The message is aimed at encouraging them to post about how they are feeling. They have the option to either complete the question and a post will be recorded in the journal or skip it.

The mood icons displayed can be changed and personalised by the young person to ensure they feel in control and adapt the message to what they like and feel helps them.

Personalisation

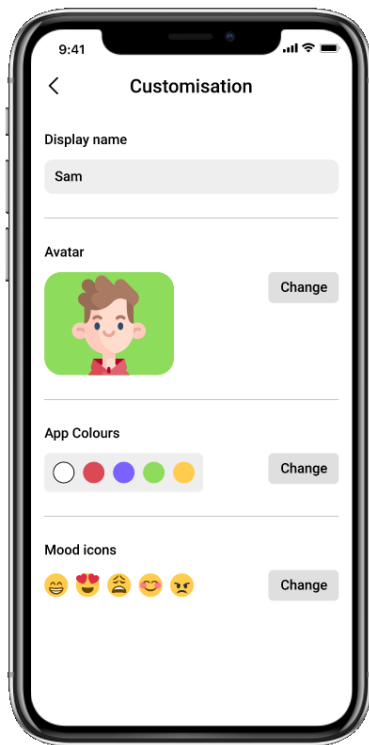


Figure 2: customisation settings

Within the profile section of the app, the young person can create their app profile and add in their own personalisation to it (Figure 2), as follows:

- Display name – free text field so that the young person can again create their own name.
- Avatar - there would be static options of cartoon images of young people, alongside famous popular options such as Marvel, Star Wars and Disney characters. The idea is to ensure the space is playful and personalised by the young person.
- App colours – applying the option here to allow further control and customisation of the app colours.
- Mood icons – here the young person has the option to add their own emoji icons that they can use throughout the app.
- Journal entry colours - Within the journal entry there is also the option to change the colour of the background. This is a small yet significant step to allow the young person to again feel that control that we know is important to them.
- Tags – Whilst the app will populate some suggestions here, such as Friends, Family, Sports etc, this will be an area unique to each young person. The tags create a way to use search to discover and group entries. Tags can be created by the young person.

Posting / creating journals

The main element of the app is the creation of a journal. This is done by the young person creating entries (see Figure 3, Figure 4 and Figure 5).

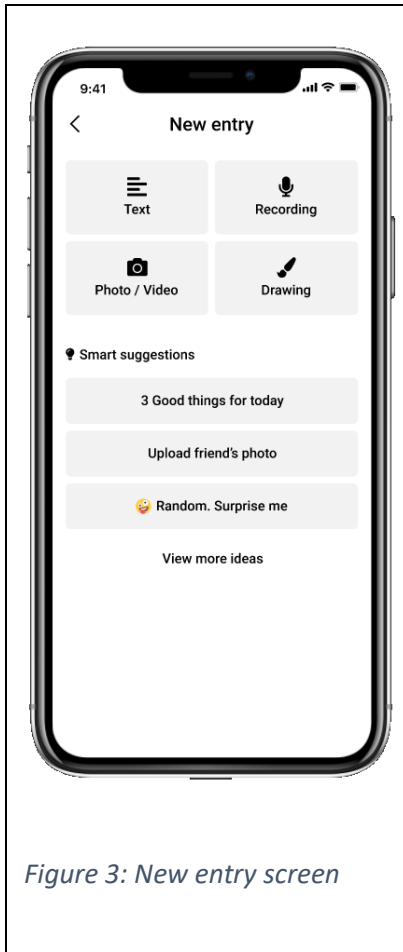


Figure 3: New entry screen



Figure 4: Post screen

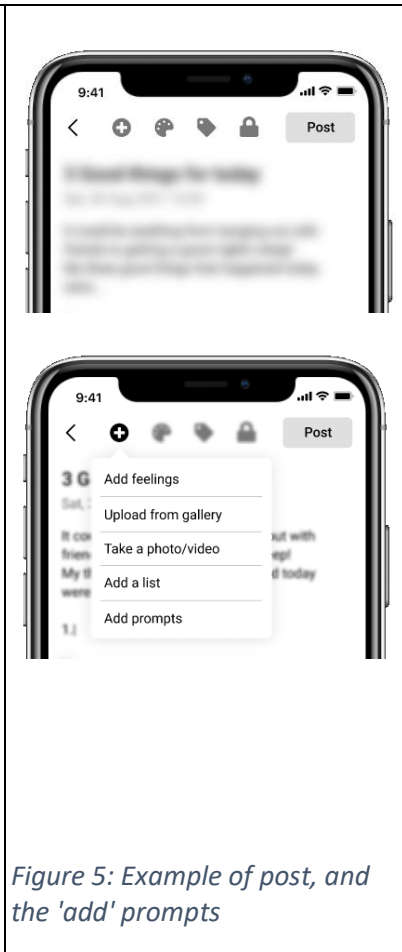
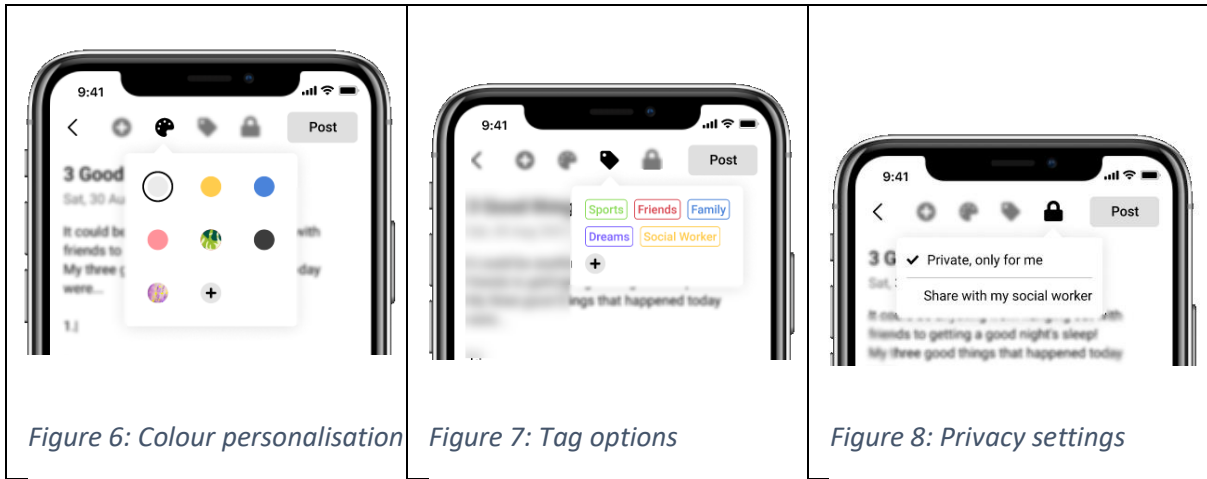


Figure 5: Example of post, and the 'add' prompts

Each entry must have a date, but can take the form of any of the following types: text, audio, video, drawing and image. There are further options to select via the smart suggestions. These are 'templates' that help guide the young person by suggesting what they might want to write about. They are to be randomised to again encourage the young person to create a broad range of entries into their journal.

Once within the entry creation screen, the young person can adapt the entry using the tool bar at the top of the screen. They have the option to add more to the post. So, if they had selected text only, they still have the option later to include a photo or feeling without having to start the create process again.



In Figure 6 we can see the personalisation option for the ability to change the entry background colour. Again, they can also add colours or patterns themselves. Tags for entries (Figure 7) create a fun label that can also be used to search through the journal. The ability to create unique tags is again important for the young person to feel in control and customise the app for them. Each entry is defaulted to private, to enforce the feeling of a safe space for the young person. However, the young person can share an entry with their social worker (see Figure 8).

Journal reflection

Once the app is being used and entries have been created it is important that young person should be able to find previous entries, quickly and easily.

From the first home screen in the app, a scrollable list is present in date order. The young person can simply scroll and find past entries (Figure 9). There is also the option to search using the search bar from the home screen – this would search by a tags or word search keywords (Figure 10). Finally, there is the option to use the calendar view again to scroll and find past entries (Figure 11).



Figure 9: 'scrollable' entries

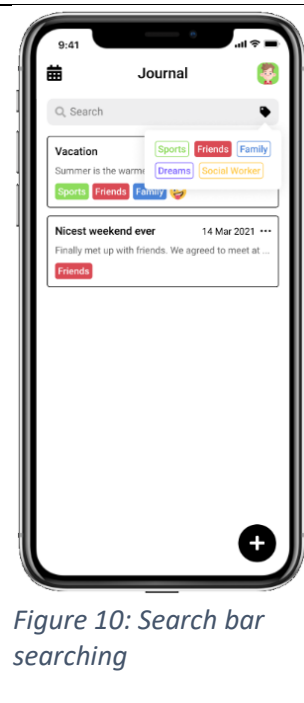


Figure 10: Search bar searching



Figure 11: Calendar searching

Engagement and rewards

The app itself is designed to encourage and reward those using it, and this is achieved by building successful onboarding. Each time a young person creates a post they are given a reward of a status achievement and several stars. Messages appear as action is taken and a total count and list can be found in the profile section of the app. The stars can be used within the app to unlock new avatars or colours/patterns etc.

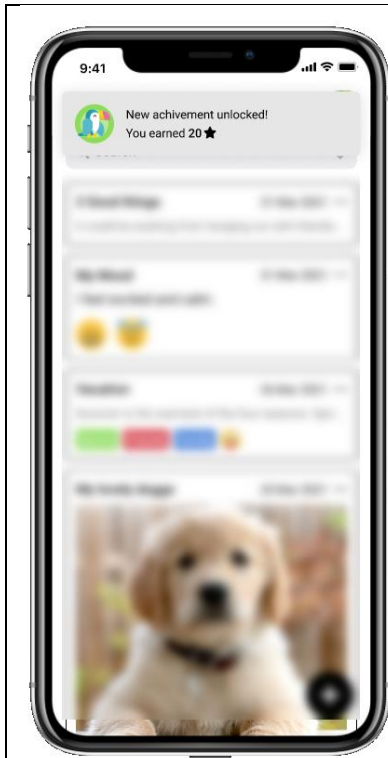


Figure 12: App reward notification

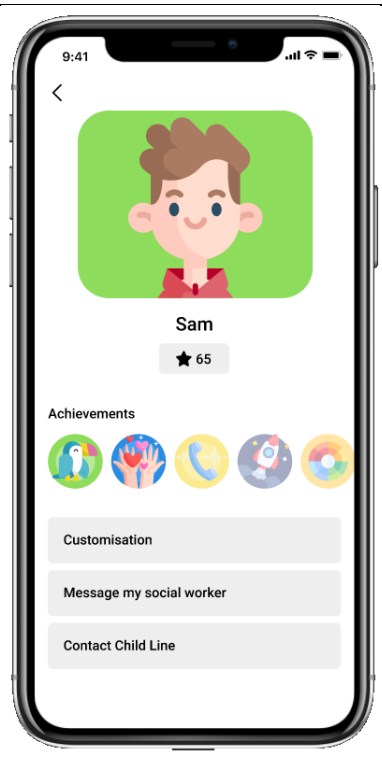


Figure 13: App achievements

The app will also allow for parents/carers to prompt entry creation.

Archive/delete

The app is designed to allow for the young person to remove entries. However, as this action might be in done error or in a moment of hesitation, the app will preserve and hide entries for a set period. The reason for this is that most children in care destroy things at points of high emotion and this will ensure that entries are archived and that the remove action can be undone.