International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-10 Issue-2, July 2021

Paas EUD Tool for Developing Expert Context-Aware Mobile Applications

Sahar Elshafei, Ehab Hassanein, Hanan Elazhary

Abstract*: Context-awareness enables systems to be tailored to the needs of users and their real circumstances at certain times. A noteworthy trend in software development is that an increasing number of software systems are being developed by individuals with expert knowledge in other sectors. Because most of the current context-aware development toolkits are intended for software developers, these types of systems cannot be easily developed by non-technical consumers. The development of tools for designing context-aware frameworks by consumers who are not programming experts but are specialists in the area of implementation would result in faster adoption of such services by businesses. This paper provides a cloud-based framework for people without programming experience to create context-aware mobile applications. The platform can provide a lightweight distribution of packaged applications that allows experts to send specified information to mobile users based on their context data without overlapping between the rules of the application. An energy-efficient mobile application was developed to acquire contextual information from the user device and to create quality data accordingly. The framework adopts Platform as a Service (PaaS) and containerization to facilitate development of context-aware mobile applications by experts in various domains rather than developing a tool for each domain in isolation, while considering multitenancy.

***Keywords:* End-user development; domain expert; context-awareness; containerization**

I. INTRODUCTION

As a result of modern and new mobile phone functions, mobile phones are increasingly considered the new personal computers [1]. The growing number of mobile device users has provided a plethora of opportunities for mobile applications. Application developers are designing and developing many applications for mobile devices in different categories such as image processing, entertainment, social media, fitness, industry, and real-time monitoring. While mobile devices can run a variety of advanced applications, they have limitations such as processing power, battery life, and storage space. These limitations build an obstacle to moving forward with the resource-constrained applications [2]. Intelligent applications are giving rise to a rapidly developing topic known as mobile cloud computing. Many smart applications are now considering the cloud computing model as an enabler to exploit its computing capacity, memory, and storage resources in order to solve the resource constraints of mobile devices [3]. As it forces new implementation and integration criteria, this approach necessitates an upfront advancement in the approach of how to design mobile cloud applications.

The real cloud refers to the conventional cloud infrastructure that supplies virtually unlimited resources, such as Amazon Elastic Compute Cloud (EC2), Microsoft Azure, and Google App Engine as well as include software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS), also includes multiple cloud deployment models, such as private, community, public, and hybrid. on the other hand, a virtual cloud refers to nearby infrastructures, such as servers and personal computers, providing services to the mobile device [4]. Moreover, many of the applications try to become more effective, smarter, and available for all people categories. They aim to understand and anticipate the situation of their users by alerting them in time, or adjust their behavior according to the new context, and this without any interference of their users [5].

The mobile cloud-computing model from a context-conscious scene can be thought of as a research path that focuses on finding the most successful options to improve cloud services used in mobile applications aware of their users' and applications' context. It is essential to decouple the application logic and its adaptation logic from context acquisition and inference at the same time. it is equally important to provide efficient and effective ways to utilize the obtained context to ease the development of context-aware applications [6]. That is why, when developing broad and scalable smart mobile applications in the cloud world, it is important to consider not only the structure of both the cloud services and the management of context information, but also the communication and collaboration between all parties concerned that support the application's smart adaptive nature. Because of the diversity and evolving nature of the user's requirements, and the difficulty in specifically identifying his requirements, the user input in the initial design of systems is part of the solution. Going through traditional development periods with software professionals to keep up with evolving contexts would be too slow, time-consuming, and costly [7]. As a result, the flexible approach is to rely on the users themselves; they must be able to continuously adapt the systems to their comfort and needs.

Manuscript received on July 16, 2021.
Revised Manuscript received on July 23, 2021.
Manuscript published on 30 July, 2021.
* Correspondence Author
Sahar Elshafei*, Department of Information Systems, Cairo University, Cairo, Egypt. Email: saharsayed_is@hotmail.com
Ehab Hassanein, Department of Information Systems, Cairo University, Cairo, Egypt. Email: e.ezat@fci-cu.edu.eg
Hanan Elazhary, Department of Computers and Systems, Electronics Research Institute, Cairo, Egypt. Email: hananelazhary@eri.sci.eg

End users are usually not as experienced or involved in adapting their programs as tech professionals are. However, it is quite likely that users will be authorized to adapt systems at a level of complexity that is appropriate for their personal, experience, and situation. The main aim of EUD is to teach end-users how to build and adopt systems on their own. Some existing research partially addresses this issue, advocating casting users as the initiators of a fast, inexpensive, and tight coevolution with the systems they are using [8]. Currently, some popular types of EUD are shown to be used with certain effectiveness in commercial software, such as documenting macros in word processors, setting up spreadsheets for calculations, and defining e-mail filtering. Although these implementations only understand a small portion of EUD's potential and still have several issues, they demonstrate why encouraging end-users to improve the technologies they use is an essential step toward allowing them to become active members of the Information Society [9].

The rest of the sections are set out as follows: Section 2 the related work is presented. In section3 presents the implemented context-aware platform and a completely description for it. Section 4 analyze the evaluation of the Platform efficiency, Section 5 discuss the application user evaluation. Finally, Section 6 offers conclusions and advice for future directions.

II. RELATED WORK

During the last years, context-aware computing has been studied by different authors for context concept simplification. Context is defined as the area of interactivity between the user and the application whereas the user tasks as the main context input for the framework through the supplied information which describes the entity case [10, 11]. Other authors describe context as the needed information to personalize the situation of an entity [12].

Context-aware computing entered in different application in different fields with deferent methodology. We will present some of them below. Different authors that depended in their research on rule-based techniques. David Martín [13] presented a cloud-based framework that has been designed to customize context-aware mobile services and applications through a web application for users with no programming experience. The platform architecture follows the design instructions of the contextual toolkits, which have been revised to include the appropriate user interface, so that all the requirements and parameters are configured. Anis Amilah Shari [14] presented a Rule-based technique to recommend and suggesting suitable food for allergic babies using IF-THEN while IF is a condition to be tested and THEN producing the action or results. To design and monitor the system development process, the Mobile Application Development Life Cycle (MADLC) model was chosen. The suggestion framework would assist parents and caregivers to reduce their time to look for the right food for their allergic baby. Hanan Elazhary [15, 16] presented two research one presented a comprehensive cloud-based framework that defined the necessary components and how they can be integrated to direct the creation of adaptive context-aware intelligent mobile user interfaces. The framework integrates several different functionalities to guide the development of interfaces with variable levels of complexity as needed. And the other one provided a cloud-based global server platform was provided to communicate general information based on the rule-based technique with a pilgrim mobile application. A mobile application for a campaign manager enables each manager to identify rules for the context-aware communication of campaign-specific information, even though these managers usually lack programming capabilities. Giuseppe Ghiani [17] presented an environment that allows end-users to use an intuitive trigger-action paradigm to customize the context-dependent behavior of their AAL Web applications. The tools to help older people achieve their goals with IT products by working with helpers are that caregivers and patients can be effectively empowered to facilitate the management of typical reminders configuration tasks, alarms, messages, medication adherence, and monitoring functionalities [18]. Proposed a framework called AUIDP (Adaptive User Interface Design Patterns) to support the generation of adaptive mobile applications by integrating ontology-based models with UI design pattern methods. This structure makes, according to user requirements, a dynamic range of appropriate UI design patterns. However, all these efforts for developing such of these approaches there is still a necessity for developing a plethora of effective aggregate solutions, ready for wide use by end-users to build context-aware mobile apps.

Briefly our framework in comparison to similar research that we represented in the related work is that our framework is superior to other with its simplicity of using for the nun technical user to create the rules which responsible of customizing the user's mobile interface by using the domain panel which developed by Laravel toolkit. Our framework is provide the ability of Multi-tenancy in which a single instance of a software application serves multiple customers in different domain at the same time and portable anywhere by applying containerization mechanism while other frameworks which are specialist in a specific field [13-16] or like Brahem [18] presented framework to examine how design patterns can help to support the generation of adaptive mobile application, but all of these efforts are still insufficient to provide a plethora of effective aggregate solutions that ready for wide use by end-users to create context-aware mobile apps. Containerization also provide a lightweight distribution of packaged applications for deployment and management, additionally using of containerization provide cost savings through the use of less system resources by packaging software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings. As it is become necessary to have a container to create applications during the development process, the applications must be uploaded to PaaS cloud. We also using a centralized DB (Amazon RDS), S3 Bucket to store data so can approach the database as an on-demand utility.

III. DESCRIPTION OF THE CONTEXT-AWARE PLATFORM

The developed prototype is a context-aware platform that allows individuals without programming skills to configure mobile application behavior established on context data.

Quite specifically, the platform was prepared for experts in the field of tourism. This way, according to their situation, they can provide relevant information to the users on the move. Based on some context parameters that can be configured through the platform's web-based interface, this data is automatically transmitted to the user's mobile device. Context factors such as the mobile user's position and language range of dates, and time range may be used to configure knowledge transmission. The position context factors can be calculated by latitudes and longitudes using the platform's web interface to construct circular areas over a layer of Google Maps, which are specified by numerals inside a comma-separated text string of six decimal places of precision. As a result, the platform user may customize the state of being within or outside the created field. These areas can be both outdoors and indoors. According to a WYSIWYG text editor, the app also assists experts in determining the information that would be sent to the visitor's mobile device when contextual conditions are met. Most of these configurations can be completed without the use of a programming line by using the web interface. The platform was created with four major technological and scientific developments in mind: The Internet of Things, End-User Creation, Containerization, and Cloud Computing.

The Internet of things is ‘‘Things having identities and virtual personalities operating in smart spaces using intelligent interfaces to connect and communicate within social, environmental, and user contexts” [19] The powerful IoT platforms can define precisely what data is useful and what can be ignored safely. Such data may be used to identify trends, make recommendations, and identify future concerns before they arise. Sensor-built-in devices and objects are linked to an Internet of Things network that incorporates data from user mobile and uses analytics to share the most useful information with applications built to meet user’s specific needs [20]. This means the framework exposes a programming interface (API) for the REST application that can be used by context sources e.g., moveable devices, smart sensors, network services) to send data to the platform.

With EUD principles, we allowed our target audience to shift from programmers to a broader audience: creators with no programming experience (from now on just producers). Supplying producers with tools for determining how their desired frameworks should be supplemented is a good solution for assisting them in the value and users of their application, leading to better software experiences [21, 22].

Cloud computing used to deliver hosted services over the internet as demand and platform as a service (PaaS) one of its types that is responsible for providing a platform with tools to test, develop and host applications in the same environment so it is a suitable option in our research to enable us to provide the expert domain with the platform tool that helps him to develop his application without the need to storage and other computing resources [23]. However, container services (CaaS) are gaining popularity in the programming world. Experts are beginning to see a shift toward container services as the primary means of handling an application's specifications, even though the two have coexisted for some time together. According to Gartner, 70% of PaaS users would choose a container service to build apps during the development process.

A. Virtualization

Container-based virtualization has been suggested as an alternative to hypervisor-based virtualization to satisfy the need for device resource savings [24] and has received a lot of publicity in recent years. Unlike traditional Hypervisor-based virtualization, which provides isolation at the hardware abstraction layer, container-based virtualization provides abstraction and operating system. container-based virtualization has been supported by Google Cloud Platform and Microsoft Azure in their cloud services due to its lightweight size, speed deployment, and migration [25, 26], and improving the efficiency and flexibility of resource allocation in cloud data centers. By comparing container-based virtualization with traditional Hypervisor-based VMs we concluded that container outperforms VM exceeded in performance, lightweight and scalability than traditional Hypervisor-based VMs [26].

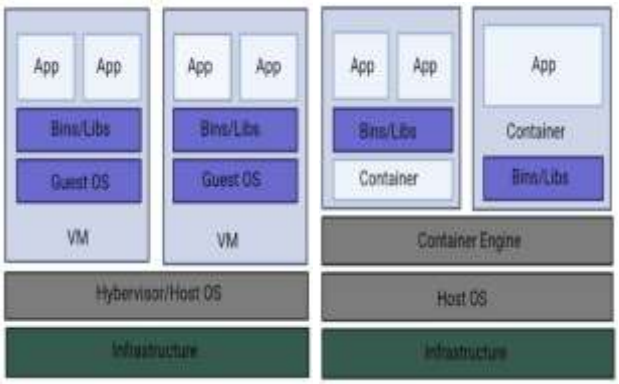

Fig 1. (a) Hypervisor-based VMs, (b) Docker container

There are several implementations for the container, ex, Solaris10, Linux Vserver, and Docker which is an open-source project to enable the reuse of the common supported libraries. Figure.1 represents the differences between Hypervisor-based VMs and Docker containers. “See Figure 1(a)” Hypervisor based VM requires a separate operating system to run and install all needed libraries before deploying applications, while Docker benefits from the services of sharing the supporting libraries using image layering “see Figure 1(b)”. Virtualization presented the needed continence such as isolation, accountability, resource allocation, resource fair sharing, and so on by evolving virtualization technologies with computational environmental development. Containerization has emerged as a major phenomenon in software growth, serving as an alternative to or supplement to virtualization. It requires encapsulating or packaging software code and all its dependencies can be able to operate on any infrastructure uniformly and consistently wherein traditional methods code is created in a particular programming method that often results in bugs and errors when moved to a new location, containerization enables us to deploy applications faster and more securely [24].

B. Context data

Context is quite an ambiguous word. Therefore, before the design and implementation of any context-aware program, it must be disambiguated. Our model should distinguish between the definition of the context from the system view and the user view. Context is described according to a collection of measurements [25]. External measurements are obtained from external sensors and networks of wireless sensors, while internal measurements are obtained from handheld internal sensors. In other words, the model does not limit developers to a single sensor category but allows applications with varying levels of complexity to be developed as required. On the other hand, the user views context as a collection of parameter values including but not limited to, location, time, environmental conditions such as temperature and sound level, and the role performed by the user. The device seamlessly translates any system view (external and internal) measurements obtained into data user view values.

It should be noted that each context user-view data is an object that allows each data to have different levels of information and to have methods to update and access it. To better describe and manage context data, Context data can be modeled using a variety of data structures, including key-value pairs, relational models, ontologies, and object-oriented models. Object-oriented models and ontology-based models are the most used context data frameworks, which are more descriptive than the previous methods models and can be used in conjunction with strong reasoning engines to extrapolate high-level meaning or circumstances from low-level context, i.e., raw context data from the sources of context. Object-oriented models are very common among programmers since this coding style is used to develop many of the current programming languages. These models allow rich relationships between context entities to be formed in addition, rule-based reasoning engines could be used to deduce higher-level context.

C. Architecture

One of the most used approaches is to conceive of context-aware systems as practical cases that change their behavior based on the context data collected. This is how the framework architecture is designed to generate outcomes (e.g., text, info, JSON, URL) prompted by logical rules and delivered to mobile applications depending on a mobile user's background. All the architecture and context data model layers have been implemented. Java is the preferred programming language [26]. Using an object-oriented model due to its simplicity and quick management. Figure. 2 shows a blocked diagram of the proposed framework and its different layers and components. The layers are the layer of background, processing layer, a layer of administration, and application layer. This allows the platform to be a black box that receives input from defined context sources and generates outputs activated by the stated context rules.

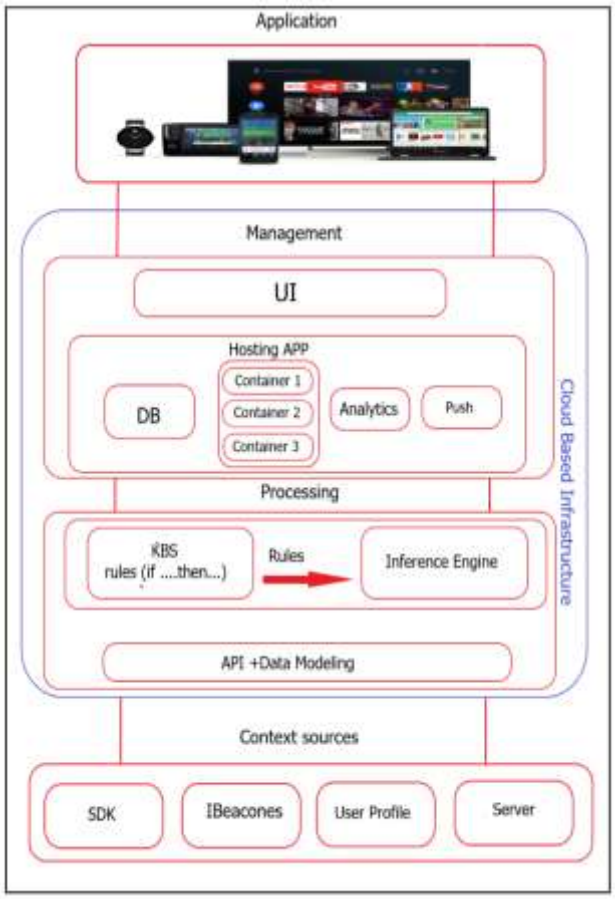

Fig 2. The Architecture of the platform.

1) Context sources

The first layer is where all the sources for the context are. Typically, these sources are dispersed and provide heterogeneous data like networks, web services, mobile devices, or databases. Four separate context resources have been used to implement the platform. The user's mobile device is the most important context source. The device user must install an application to get data from the smartphone. This application has an embedded SDK that, using a background service, obtains the locale of the user profile data (identity, name, gender, age, education, demographic, characteristics), preferences (audiovisual, presentation), activities, using history and the location data every 30 s. We get these data from the user's mobile while he registers to our application. The time and date of the server where the platform is being deployed must also be detected. Finally, we need to obtain the user location.

2) Processing

This layer receives and transforms context data from the layer of context source, these data from row-level into a context data model that are more readable for the human and easy to be understood. The context data are represented in the form of the object-oriented model as we mentioned before because it is simple in programming.

API is exposed to link a specific development environment with developers, to allow these developers to benefit from the services of this environment without the need to build everything from scratch therefore we can use it to send all the context resources to the platform. Most experts find it simple to communicate their expertise in the form of problem-solving laws [27].

Our IMUI integration strategy is based on an extremely simple, changeable, and expandable information representation technique [28]. A rule-based operating system is generally referred to as a knowledge-based system. The rules are provided as instructions and directives in this document, and all the specifications are dependent on our context model and the user model derived from experiments conducted and developed for the applications [29]. Context-aware computing helps us define the parameters that an application should use as a basis to change its actions. The expert system is the critical part of the framework, it must infer high-level context based on low-level or raw context data. The expert system is divided into two parts. The knowledge contains the knowledge of the domain which is useful to designers. Awareness is expressed in a rule-based approach by a set of rules, each rule defining a relationship, suggestion, or directive, and having the form of IF (condition) THEN (action). The rule is said to be "fire" when the condition of a rule is met, and the operation is enforced. We used laravel content management system for its simplicity use by the domain expert, it is a versatile PHP platform that allows non-technically savvy users to build usable pages or upload and change content without needing to outsource the work to a web designer or understand programming languages like HTML or PHP “see Figure 3”.

Fig 3. Add rules interface.

The rules are said to fire when a condition of a rule is matched, and the resulting action occurs. In the proposed framework, adaptation rules are stored in the database in the Rules table which holds all the rules which are created by the admins. The inference engine matches the action against the condition sections of the rules as context changes. The rule fires and the corresponding action is executed if the conditions of a rule match the context, resulting in an intelligent user interface adaptation. If none of the rules burn, on the other hand, adaptive mobile features are reset to their default values.

3) Management

This layer is responsible for the rules that specify the adaptations that need to be made to a mobile user interface in a specific context. For doing adapting process we need to get the user data these facts are obtained from the user mobile. When a domain expert recognizes an environment, background conditions, or new knowledge, new rules are generated and stored in the platform (Rules). All the data and conditions coming from the user platform are stored in the (DB). If any of the generated rules are executed, the information identified by the specific domain is delivered to the user's smartphone via a push message service. We used containerization technology to encapsulate the framework as a single software executable package that includes all associated configuration files, libraries, and dependencies needed to run it. Containerized applications are "isolated" since they do not bundle an operating system with a copy. Instead, on the host's operating system, so we used an open-source runtime engine called Docker to become the gateway for containers to share an operating system on the same computer system with other containers. The host operating system abstraction renders containerized apps versatile and able to run across any device or cloud uniformly and continuously.

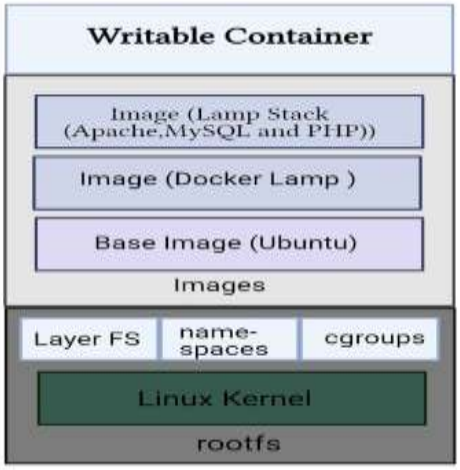

Fig 4. Container Image Architecture.

Containers are operating system virtualization strategies that are especially well suited to application management in the PaaS cloud. Containers are defined by lightweight images; VMs, on the other hand, it based on absolute, monolithic images. Containerized systems are almost fully separated. Container images serve as the foundation upon which containers are built. Docker will be used to show how containerization works since it is the most common container solution. A Docker image is made up of layered file systems, like the Linux virtualization stack, which uses the LXC mechanisms “see Figure 4”.

- During a standard Linux boot, the kernel mounts the root file system as read-only and then checks its integrity before converting the rootfs volume to read-write mode.

Docker installs a writable file system is an essential component of a read-only file system using a union load. Instead of shifting the file system to read-write mode as it does in a conventional boot.

- Multiple read-only file systems could be stacked on top of one another using union mount, allowing you to create new images by building on top of existing ones. The container engine loads each of these file system layers as a separate image for execution.
- The top layer is the container itself and it is the only writable layer which has state and can be executed. It can be compared to a directory that contains all the files needed for execution.

The typical layering includes (top to bottom,"see Figure 4"): writable container image for applications, LAMP stack (Apache, MySQL, and PHP) and Lamp image as sample platform components, a Linux image (a distribution such as Ubuntu), and the rootfs kernel image. Layers made up of individual images formed on top of a base image that can be expanded makeup containers. Docker images in their entirety are used to build lightweight application containers. They are also components of program stacks. The method is simple since single images can be easily modified and distributed.

4) Application

To obtain customized information from the platform based on the background conditions specified by the system expert, the mobile user must install an application. To visualize this data, an Android application was created, which is also in charge of handling and transmitting user context data to the platform. The mobile user receives warning information with relevant information for that case when the background conditions are aligned with those configured on the platform, and the mobile device vibrates. Alerts may be handled by the mobile user, removed, or saved on the mobile device.

IV. APPLICATION OF THE FRAMEWORK

An application of the AMA (Adaptive Mobile Application) framework is discussed in this part in order to bring our research closer to practice. We have developed a prototype for a mobile application called MBA (Mobile Based Assistant), which is an app for tourists who search for tourism places. It is able to track users' current locations, time and date conditions around them and notify them with interested location for these users. A case study begins with a description of many user scenarios that demonstrate how to utilize distinct applications and emphasize how user interfaces adapt to changing user requirements.

First scenario

A case study is offered to demonstrate how the AMA prototype can deal with distinct user case scenarios in order to assess the applicability of the proposed framework .For example: If the domain expert fills the domain panel with data (lat, long, radios, data, time, list which the user belongs) which based on those data when the user context satisfies these data a specific information well be displayed on the user UI. If we supposed that a tourist (who belongs to a tourism company which support our application) where in Giza at the morning during the time period he decided to stay in Egypt, he will receive a notification message about Giza pyramids "see Figure 5".

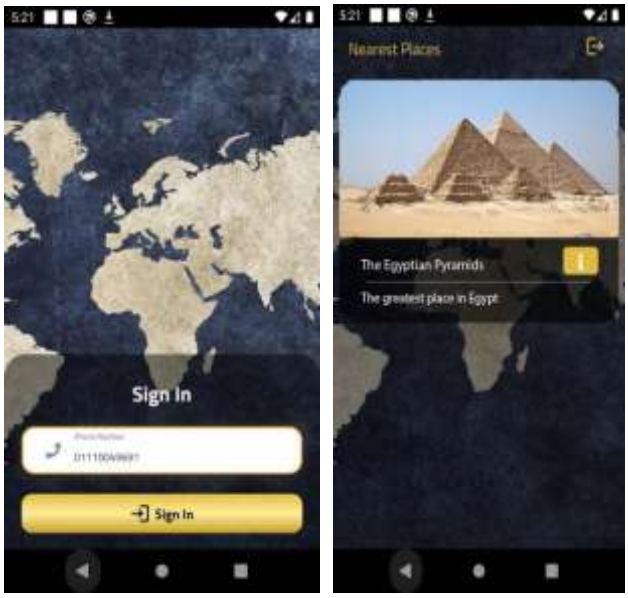

Fig 5. Scenario case adapted to tourists users.

Second scenario

The second scenario is an elderly people who are forced to live independently at home. Providing personalized services is critical, because elderly people often need various individual needs, requirements and disabilities. We can empower the role of the patients and caregivers as they can be effectively empowered to help in the management of traditional tasks relating to the configuration of alerts, alarms, notifications, adherence to prescriptions, and functionality monitoring. In this application we give the caregivers the ability to help patient and elderly people to customize the context-dependent behavior based on rule-based approach "see Figure 6". Adaptation application had to change continuously according to the context changing and the user needs and this nearly impossible that we can satisfy anticipate all the requirements at design time. There are several rules that can be taken into account and classified based on the main goal. For example: Rules related to reminder or alarm: these rules aim to sending a notification message to the user when a critical situation about to occur. Example:

IF <the time to take the medicine came >

AND <the elderly doesn't take the medicine >

Action <send the elderly reminder message>.

As presented different applications and platform components can be combined in a container by constructing them from individual images, based on base images from the repositories which depicts a containerized application. Containers can encapsulate several application components such as resources, dependences, libraries, and databases through the image layering and the user only required to run that image to deploy the application.

Containers make storage and network management easier by packaging application for interoperable and dispersed contexts. A data volume container enables sharing persistent data between application containers through a dedicated, separate data storage container by passes the union file system to provide features for persistent or shared data without interlaping between applications.

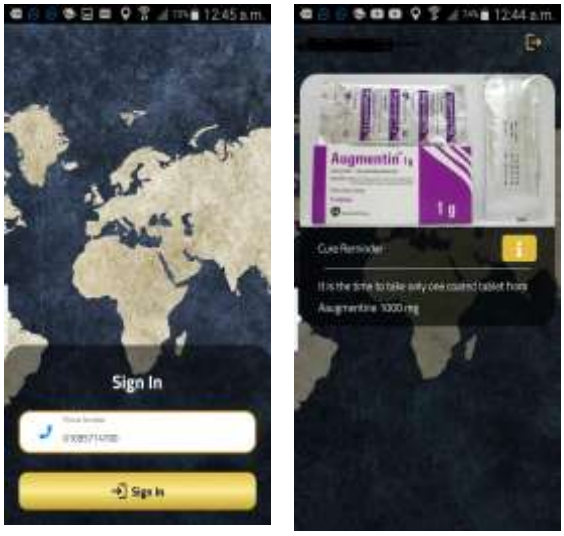

Fig 6. Scenario case adapted elderly people.

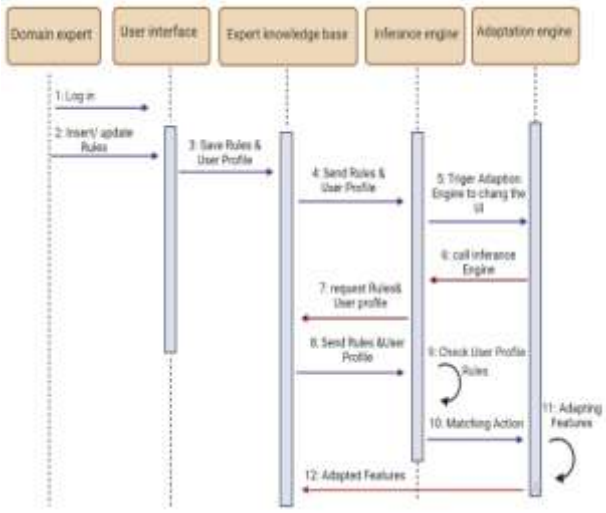

Fig 7. Sequence diagram representing the UI adaptation process.

The UI adaptation engine and the engine interface component are responsible for generating adaptive interfaces that fit the needs of users. It takes as its input a set of user preferences and rules that are retrieved from the inference engine and then adapts them to the user needs. “See Figure 7” shows a sequence diagram showing the process of UI adaptation within our framework. Firstly, a connection occurs between the domain expert and the user interface by logging in the application and begins to insert, delete, or update the rules he wants to apply on the users. These rules save in the expert knowledge base and send to the inference engine which trigger the adaption engine to change adaption UI based on the change occur in the context. The adaptation engine calls the inference engine to contact with the expert knowledge which contained the user profile, rules, and context data. After that the expert knowledge sent these data to the inference engine to help in the reasoning process and matching between the context data and the rules. When a condition of a rule is matched, the rule is called to fire and sent to the adaptation engine generating the adaptation message and sent it to the KB where they store until send to the mobile user interface.

V. EVALUATION OF THE PLATFORM EFFICIENCY

Two major shortfalls might influence the efficiency of the entire process. The First one is about the location data collection process of the mobile user's Google maps sensor. It is a daily method that can destroy the smartphone's battery in a matter of a few hours. The other would be associated with the platform's reasoning process. Depending on millions of data, a lengthy process may influence the consistency of the platform introduced. Several methods have been created to address these major issues.

A. Context data gathering process

To process and control the platform, the software development kit (SDK) integrated into the framework regularly obtains context data from the user's smartphone. The battery declines rapidly when it collects data from the mobile device's GPS, this is one of the primary challenges in such systems [30]. To achieve an energy-efficient algorithm, the location data collection process is managed. The algorithm uses Google Play Services as a behavior recognition service. To detect the stillness, standing, or walking of the smartphone user [13]. The GPS is turned off when the user remains in its position. The algorithm turns on the GPS if the user is walking. Every 10 seconds, the activity control process is repeated. In addition, every 10 seconds the accessibility of the Internet connection is tested. If the Internet connection is not accessible, the GPS is turned off because the site coordinates cannot be sent to the platform without an internet connection. The GPS is turned on if the Internet connection is open. Every 30 seconds, the GPS location collection process is conducted.

The SDK will send the coordinates to the platform until the location is reached. The distance to the nearest area from the position of the user is returned in response to the platform. This data is used to predict the time required for reaching this location, taking into consideration the average speed of human walking as1 m per second. The GPS is turned off for 4min if the time reaches 5 minutes. The position coordinates are obtained after a delay of 30 seconds if the duration is less than 5 minutes.

This algorithm is illustrated in “Figure 7”. A one-hour walking tour with four separate outdoor areas on the platform has been planned to relay some information to the smartphone. Two smartphones of Samsung note 4 have been used. The first device was installed with the optimization algorithm. While the other had a software without an optimization algorithm. This second app was designed to get and send GPS location information to the platform every two minutes.

Five times, the two smartphones walked on the route checking the battery consumption of the GPS. The application without an optimization algorithm acquired a total of consumption 933.68 J, while the energy consumption for the application with an optimization algorithm was 21.1J, which is significantly better.

```
Open application
Do while connection is not available
        Get internet connection
        If not connection available then
                Swich off GBS
                Wait 10 seconds
        End if
End do
Do while Get activity is not walking
        Get activity
        If activity is not walking then
                Swich off GPS
                Wait 10 seconds
        End if
End do

SwichGPS:
Swich on GPS
Getlocation:
Get location
Send location to platform
Receive the distance to the closest area from the platform
Estimate the needed time to reach the closest area
Set estimate Reach Time to needed time to reach the closest area
        If estimate Reach Time > 5 minutes Then
                Swich off GPS
                Wait 4 minutes
                Goto SwichGPS
        Else
                Wait 30 second
                Go to Getlocation
        End if
```

Fig 7. Performance optimization algorithm

Table- I: Results of the satisfaction questionnaire towards salient features of the proposed framework.

Tested features	Mean	SD
I think that I would like to use Context-Aware Mobile app frequently.	3.1	0.29
I found Context-Aware Mobile app unnecessarily complex.	1.2	0.03
I thought Context-Aware Mobile app was easy to use.	4.8	0,03
I think that I would need the support of a technical person to be able to use Context-Aware Mobile app.	1.1	0.01
I found the various functions in Context-Aware Mobile app were well integrated.	4.7	0.05
I thought there was too much inconsistency in Context-Aware Mobile app.	1.2	0.03
I would imagine that most people would learn to use Context-Aware Mobile app very quickly.	4.7	0.20
I found Context-Aware Mobile app very cumbersome (awkward) to use.	1.2	0.03
I felt very confident using Context-Aware Mobile app.	4.5	0.25
I needed to learn a lot of things before I could get going with Context-Aware Mobile app.	1	0.00

Table- II: Results of the user evaluation.

Participants	System Usability Score (SUS)
User1	85
User2	90
User3	90
User4	90
User5	85
User6	90
User7	87.5
User8	90
User9	95
User10	92.5

B. Justification method

The management of context data collection needs to be organized by applying a reasoning process that is responsible for obtaining high-level context data from low-level context by using a reasoning engine. In the reasoning process, we consider all the low-level data as a fact and stored in the KB. The reasoning process is designed to take a second to perform a task if it takes more the process is launched again automatically until the process is carried out. To facilitate the reasoning process and perform it faster we need to decrease the number of rules created by the platform to configure the context circumstances [31].

The management of context data collection needs to be organized by applying a reasoning process that is responsible for obtaining high-level context data from low-level context by using a reasoning engine. In the reasoning process, we consider all the low-level data as a fact and stored in the KB. The reasoning process is designed to take a second to perform a task if it takes more the process is launched again automatically until the process is carried out. To facilitate the reasoning process and perform it faster we need to decrease the number of rules created by the platform to configure the context circumstances [31].

VI. USER EVALUATION

We evaluated our platform with ten non-technical people that are expert in tourism domain and do not have any programing skills. Those domain experts had to answer about ten questions that measured the performance and the user satisfaction of the platform using a Likert scale between 1(strongly disagree) and 5(strongly agree) and all the other responses in between getting the number in between. Although only some features were proposed, the participants had a positive attitude towards the proposed system.

It is obvious that, even though only some features were proposed, the participants had a positive attitude towards the proposed system. The participants should complete a questionnaire depending on the system Usability Scale (SUS), which is one of the most used questionnaires for testing usability impressions. The assessment is based on the points assigned by users to each question as well as the SUS scores.

According to this scale, the platform is open to all participants, with an overall score of 90 points. The lowest possible score was 87.5, and the highest possible score was 95. This evidence demonstrates that the platform introduced was regarded as a valuable tool by the evaluation participants and that they found it simple to use.

VII. CONCLUSION AND FUTURE DIRECTION

This article addresses the problem of customizing mobile services and applications via a web platform built for people who do not have the knowledge of programming. To configure the required context-aware conditions and parameters without a programming language, the architecture of the framework follows the design guidelines of the updated context-aware toolkits with an adapted user interface. It also adopts Platform as a Service (PaaS) and containerization to facilitate development of context-aware mobile applications by experts in various domains rather than developing a tool for each domain in isolation, while considering multitenancy.

As the results of the user assessment demonstrate, the built framework helps end-users to create mobile application contexts. The user evaluation participants noticed that the platform could be easily used and configured to create a context-aware mobile application. This provides the context-aware computing research and mobile industry a new opportunity. The introduction of systems such as the one in this research will enable users with no technical skills early adoption of context-aware systems. The increasing use of these innovations would give customers a better user experience and a more customizable service. As future work, we will consider more sophisticated user interfaces and rules.

REFERENCES

1. Allen, S., V. Graupera, and L. Lundrigan, *Pro smartphone cross-platform development: iPhone, blackberry, windows mobile and android development and distribution*. 2010: Apress.
2. Satyanarayanan, M., *Mobile computing.* Computer, 1993. **26**(9): p. 81-82.
3. Gu, F., J. Niu, and Z. He, *A Research on Mobile Cloud Computing and Future Trends.* EAI Endorsed Trans. Indust. Netw. & Intellig. Syst., 2016. **3**(7): p. e4.
4. Rahimi, M.R., et al., *Mobile cloud computing: A survey, state of art and future directions.* Mobile Networks and Applications, 2014. **19**(2): p. 133-143.
5. Huang, D., *Mobile cloud computing.* IEEE COMSOC Multimedia Communications Technical Committee (MMTC) E-Letter, 2011. **6**(10): p. 27-31.
6. Bricon-Souf, N. and C.R. Newman, *Context awareness in health care: A review*. international journal of medical informatics, 2007. **76**(1): p. 2-12.
7. Liu, W., X. Li, and D. Huang. *A survey on context awareness*. in *2011 International Conference on Computer Science and Service System (CSSS)*. 2011. IEEE.
8. Constantine, L.L. and L. Lockwood, *Process agility and software usability: Toward lightweight usage-centered design.* Information Age, 2002. **8**(8): p. 1-10.
9. Dey, A.K. and J. Häkkilä, *Context-awareness and mobile devices*, in *Handbook of research on user interface design and evaluation for mobile technology*. 2008, IGI Global. p. 205-217.
10. Chen, G. and D. Kotz, *A survey of context-aware mobile computing research.* 2000.
11. Henricksen, K., *A framework for context-aware pervasive computing applications.* 2003.
12. Dey, A.K., G.D. Abowd, and D. Salber, *A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications.* Human–Computer Interaction, 2001. **16**(2-4): p. 97-166.
13. Martín, D., C. Lamsfus, and A. Alzua-Sorzabal, *A cloud-based platform to develop context-aware mobile applications by domain experts.* Computer Standards & Interfaces, 2016. **44**: p. 177-184.
14. Shari, A.A., et al. *Mobile Application of Food Recommendation For Allergy Baby Using Rule-Based Technique*. in *2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*. 2019. IEEE.
15. Elazhary, H., *A cloud-based framework for context-aware intelligent mobile user interfaces in healthcare applications.* Journal of Medical Imaging and Health Informatics, 2015. **5**(8): p. 1680-1687.
16. Elazhary, H., *Cloud-based Context-aware Mobile Applications and Framework for Hajj and Umrah Management.* Int. J. Comput. Trends Technol., 2017. **47**(2): p. 106-115.
17. Ghiani, G., et al. *End-user personalization of context-dependent applications in AAL scenarios*. in *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*. 2016.
18. Braham, A., et al. *Generation of adaptive mobile applications based on design patterns for user interfaces*. in *Multidisciplinary Digital Publishing Institute Proceedings*. 2019.
19. Enterprise, N., *RFID & Micro & Nanosystems, 2008.* Proceedings of Co-operation with the Working Group RFID of the ETP EPOSS, Internet of Things in, 2020.
20. Pittarello, F. and L. Bertani. *CASTOR: learning to create context-sensitive and emotionally engaging narrations in-situ.* in *Proceedings of the 11th International Conference on Interaction Design and Children*. 2012.
21. Scaffidi, C., M. Shaw, and B. Myers. *Estimating the numbers of end users and end user programmers*. in *2005 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05)*. 2005. IEEE.
22. Stolee, K.T., S. Elbaum, and A. Sarma. *End-user programmers and their communities: An artifact-based analysis*. in *2011 International Symposium on Empirical Software Engineering and Measurement*. 2011. IEEE.
23. Seifert, J., et al. *Mobidev: a tool for creating apps on mobile phones*. in *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*. 2011.
24. Dua, R., A.R. Raja, and D. Kakadia. *Virtualization vs containerization to support paas*. in *2014 IEEE International Conference on Cloud Engineering*. 2014. IEEE.
25. Zouhaier, L., Y.B. Hlaoui, and L.J.B. Ayed. *Building adaptive accessible context-aware for user interface tailored to disable users*. in *2013 IEEE 37th Annual Computer Software and Applications Conference Workshops*. 2013. IEEE.
26. Bauer, C. and A. Novotny, *A consolidated view of context for intelligent systems.* Journal of Ambient Intelligence and Smart Environments, 2017. **9**(4): p. 377-393.
27. Alnanih, R., O. Ormandjieva, and T. Radhakrishnan. *Empirical evaluation of intelligent mobile user interfaces in healthcare*. in *Canadian Conference on Artificial Intelligence*. 2014. Springer.
28. Sultan, S. and P. Mohan, *Transforming usage data into a sustainable mobile health solution.* Electronic Markets, 2013. **23**(1): p. 63-72.
29. Alnanih, R., T. Radhakrishnan, and O. Ormandjieva, *Characterising context for mobile user interfaces in health care applications.* Procedia Computer Science, 2012. **10**: p. 1086-1093.
30. Ryoo, J., H. Kim, and S.R. Das. *Geo-fencing: Geographical-fencing based energy-aware proactive framework for mobile devices*. in *2012 IEEE 20th International Workshop on Quality of Service*. 2012. IEEE.
31. Wolfgang, L., *Design Patterns in Production Systems*. 2013.

AUTHORS PROFILE

Sahar Elshafei, earned her B.Sc. from the Department of Information Systems, Misr University for science and technology, 6th of October, Egypt. Currently she is a teaching assistant in the same department and is pursuing her M.Sc. degree in computers and information systems at Cairo University.

Ehab Hassanein, earned a Master degree in Computer Science from Northwestern University in 1989 and a Ph. D. degree from the same university in December 1992. He is the formal Chairman of the Information Systems Department, Faculty of Computers and Information, Cairo University, Egypt. He is currently a professor in the same department.

Hanan Elazhary, earned her B.Sc. and M.Sc. degrees from the Department of Electronics and Communications Engineering, Cairo University. She earned her Ph.D. degree in Computer Science and Engineering from the University of Connecticut, USA. Currently, she is a professor in the Computers and Systems Department, Electronics Research Institute, Cairo, Egypt. She has numerous publications in mobile cloud computing.