# Development of a Model for Simple Educational Mobile Applications: A Case Study of Evaluation Matrix

# 간편 교육용 모바일 앱 구현을 위한 모델 개발: 평가행렬법에 대한 사례 연구

By

**YUNUS EMRE OZTURK**

Master's Thesis

Submitted to the Department of Education and the

Faculty of the Graduate School of Seoul National University

In partial fulfillment of the requirements for the

Master's Degree in Education

Major: Educational Technology

August 2017

# Development of a Model for Simple Educational Mobile Applications: A Case Study of Evaluation Matrix

# 간편 교육용 모바일 앱 구현을 위한 모델 개발: 평가행렬법에 대한 사례 연구

By

**YUNUS EMRE OZTURK**

Academic Advisor: Cheolil Lim, Ph. D.

Submitted to the Department of Education and the
Faculty of the Graduate School of Seoul National University

In partial fulfillment of the requirements for the
Master's Degree in Education
June 2017

Approved by the committee:

| | | |
|---|---|---|
| Chair Person | Dr. Ilju Rha | _____ |
| Vice Chair | Dr. Cheolil Lim | _____ |
| Examiner | Dr. Young Hoan Cho | _____ |

# ABSTRACT

## Development of a Model for Simple Educational Mobile Applications:

## A Case Study of Evaluation Matrix

Yunus Emre OZTURK
Department of Education
The Graduate School
Seoul National University

Mobile devices, especially smartphones, have become one of the most indispensable parts of our lives. The popularity of smartphones and mobile applications has been increasing and mobile technology has been enhanced day by day. Accordingly, educators too have been trying to utilize these technologies for educational purposes. Through the various educational studies on mobile devices, these technologies already have proved how much they are powerful in higher education settings. The unique capabilities of mobile technologies, including connectivity, cameras, sensors, and GPS provide a variety of learning experiences and offer new opportunities for learners both inside and outside the classroom.

This study has been progressed as followed. First, the initial Model for Developing Simple Educational Mobile Applications (MODSEMA) has been constructed based on the corresponding literature review. Second, According to the feedbacks derived from the interviews conducted with three experts on MODSEMA, it has been revised. After that, the modified MODSEMA has been validated through a case study in which the app named Evaluation Matrix, used for promoting creativity, has been developed by following it. During and after the implementation of the case study, MODSEMA has been revised whenever is needed, and then, the ultimate MODSEMA has been proposed for educators, researchers and developers.

This study has a significance due to the followings. First, the proposed model, MODSEMA, encompasses all development process, provides a detailed guidance on how to apply it and demonstrates its implementation method via a case study. Besides, it serves information on major mobile platforms, development environments, tools and user interface (UI) frameworks, which are continuously changing and developing technologies providing new opportunities and alternatives for developers. In addition, MODSEMA guides on how to evaluate the mobile application depending on its development stage. Second, this study clearly shows that thanks to evolving technology enabling to develop hybrid apps working on multiple mobile platforms, it is possible to develop an educational mobile application which has a high usability including effectiveness, efficiency and satisfaction and which copes with the heterogeneity problem of classrooms. The advantages of hybrid apps are not only that they are available in multiple mobile services including IOS, Android and Windows Phone, but also that they provide high performance and high-quality UI as much as native apps.

Lastly, this study also verified that learners want to utilize mobile applications more for educational purposes, a conclusion derived from the analyses of usability questionnaires conducted as a part of the case study.

**Keywords:** educational mobile application development, educational mobile application, mobile learning, m-learning, PhoneGap, Evaluation Matrix

**Student ID:** 2015-23310

# TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

# CHAPTER I. INTRODUCTION

## 1. Background of the Study

Mobile devices, especially smartphones, have become one of the most indispensable parts of our lives. We use smartphones not only for talking or sending messages, but also for shopping, playing games, surfing on the Internet, entering Social Networks such as Facebook and Twitter, online banking, navigation, taking pictures and videos or recording audio, and even for learning informally and formally. We do these activities and much, much more through *applications* installed in these devices. Day by day, the number of smartphone users has been increasing. According to PEW 2015 Report, 88% of Korean adults and 72% of U.S adults own a smartphone (Anderson & Caumont, 2015). Even this number is much higher among young generations. EDUCAUSE Reports indicated that while the smartphone ownership among undergraduates was 86% in 2014, this number has been reached to 95% in 2015 (Chen et al., 2015). Similarly, the number of mobile applications has been increasing each passing day. For example, the mobile applications (apps) available in Google Play Store was around 1,000,000 in July 2013, but as the date of September 2016, this number has been increased to 2,400,000 (Statista, 2016a).

In this situation, in which the popularity of smartphones and mobile apps has been increasing and mobile technology has been improved day by day, naturally, educators too have been trying to utilize these technologies for

educational purposes. Through the various educational studies on mobile devices, these technologies already have proved how much they are powerful in higher education settings (Johnson et al., 2013). Therefore, educators make efforts to utilize educational mobile apps in order to promote learning and teaching.

**Statement of the Problem**

The unique capabilities of mobile technologies, containing connectivity, cameras, sensors, and GPS, (Berking et al., 2013) have great potential to leverage learning experience and offer new opportunities for learners both inside and outside the classroom (Chen et al., 2015). However, there is a gap between accessibility of mobile devices and their productive use in the classrooms (Sun et al., 2016). Relevant and appropriate apps to the educational contexts are not always available, especially an app that is used for facilitating specific teaching purposes (Hsu & Ching, 2013; Sun et al., 2016), so that there is a need for developing one's own mobile apps for teaching and learning purposes (Hsu & Ching, 2013; Sun et al., 2016). Yet, designing educational mobile applications remains a challenge for educators if they do not have programming experience (Hsu & Ching, 2013). For this reason, they need to be encouraged for creating educational mobile apps. Therefore, there is a need for providing models or guidelines for developing educational mobile applications in order to help them about how they can design and develop an app, which kinds of the process they need to perform for this and what they should be careful in that process like design principles and user interface.

One of those mentioned specific teaching purposes might be to

develop learners' creativity. As is known, creativity is one of the most indispensable competencies required for being successful in the 21st century (Larson & Miller, 2011), and it is the responsibility of schools to prepare young generations for the future. Creativity can be cultivated via either teaching creative thinking techniques or engaging learners in creativity-promoted environments (Lim et al., 2014). There are creative process models enhancing creative thinking, such as Creative Problem Solving (Osborn, 1953; Treffinger et al., 2000, 2006), Synectics (Gordon, 1961) and TRIZ (Terninko, Zusman & Zlotin, 1998). Creative Problem Solving (CPS), one of them, has been applied successfully in the education field, from college to the primary grades (Treffinger & Isaksen, 2005), and its effectiveness has been empirically proved in a variety of studies (Treffinger & Isaksen, 2005). With CPS, individuals go through divergent and convergent thinking processes in order to solve the problem creatively (Treffinger et al., 2006), and creativity shows up during this process. For this, CPS offers various tools categorized as divergent thinking tools including Brainstorming, Forced Connection Method and Attribute Listing and convergent thinking tools containing HIT, Highlighting, ALU (Advantage, Limitation, and Unique Qualities), PMI (Plus, Minus and Interesting), and Evaluation Matrix (Higgins, 2006). Besides, Lim and his colleagues have developed a web-based system of CPS (2008), named it as $CPS^3$ (Creative Problem Solving Support System), and then implemented in the actual college context (Lim et al., 2009, 2011, 2012). Also, they upgraded it according to those studies, which is called as $S^3CPS$, and implemented several times on university context (Lim et al., 2013, 2014, 2016). Although their findings (2013) were mostly positive, there were improvable points for their system. Firstly, the

students demanded the mobile application version of all $S^3$CPS (Lim et al., 2014) system in order to reach it anytime and anywhere and to perform those convergent and divergent thinking activities much comfortably. Also, they requested more user-friendly interface of the system (Lim et al., 2014). In addition, the case study conducted in 2015 (Lim et al., 2016) has revealed that the learners look for additional features and improvements for the existing Evaluation Matrix tool. Those requested additional features and improvements of Evaluation Matrix tool include the data extraction feature, which enables users to extract the data on Evaluation Matrix into some formats such as Excel file format, copy and paste feature from Evaluation Matrix to Excel, enabling to add table automatically, enabling to write f(x) functions on the cells, an alternative system to the monotony of using statistics terms (such as mean, numbers, weight) on Evaluation Matrix, and so on. Therefore, there is a need for not only upgrading Evaluation Matrix tool's feature but also developing it as a mobile application.

In conclusion, this study tries to deal with not only constructing a development model for simple educational mobile applications but also developing an educational mobile application having specific teaching and learning purpose, which is called Evaluation Matrix for CPS, by following that development model, for its validation.

## 2. Research Questions

In conclusion, this study aims to construct a model for developing a simple educational mobile application, and then revise and validate it via a

case study that develops “Evaluation Matrix” app having a high usability by following that model. Therefore, the research questions of this study are as followed:

I. What is the model for developing a simple educational mobile application?

II. What is the usability of that model?

## 3. Definition of Terms

**Mobile Application:** Wikipedia defines mobile application (or mobile app) as a software application designed to run on mobile devices such as smartphones and tablet computers (https://en.wikipedia.org/wiki/Mobile_app).

**Evaluation Matrix:** Evaluation matrix is a convergent thinking tool used in Creative Problem Solving (Treffinger et al., 2006). Learners list the possible ideas/solutions on one axis of a matrix (or grid) and put important criteria to be evaluated on to the other axis. This allows individuals to compare and contrast several ideas (e.g. solutions) according to similar criteria (Lim, 2013). Through evaluation matrix, the possible solutions or ideas are systematically evaluated according to the evaluation criteria.

# CHAPTER II. LITERATURE REVIEW

## 1. Mobile Devices for Learning

Mobile devices such as smartphones have vastly gained popularity and their capabilities have been continuously increasing with each passing year (Johnson et al., 2010). Especially smartphone market is huge and continuously growing, which means that an enormous and increasing number of people all over the world now have and use a computer that fits in their hands and is able to connect to the internet anywhere (Johnson et al., 2010). According to PEW 2015 Report, while 88% of Korean adults own a smartphone, this number is 72% in U.S. (Anderson & Caumont, 2015). In addition, millions of mobile applications (apps) developed to promote a wide range of tasks on smartphones are readily available and have continuously growing market. These mobile computing tools are used for business, capturing audio-videos and editing, measurement, geolocation, social networking, personal productivity, references, just-in-time learning and so on (Johnson et al., 2010). As the date of June 2016, while there are 2.200.000 apps in Google Play, 2.000.000 apps in Apple App Store and 669.000 apps in Windows Store are available (Statista, 2016b).

On the other hand, mobile devices have very high potential for the use of educational purposes because of rapidly increasing smartphone ownership among young generations and benefits on teaching and learning. For instance, according to EDUCAUSE Report, while the smartphone ownership among undergraduates was 86% in 2014, this number has reached

95% in 2015 (Chen et al., 2015). People started to expect to be able to work, learn, and study whenever and wherever they want to (Johnson et al., 2013). For this purpose, they utilize different kinds of mobile devices with various apps. Besides, since technologies such as tablets and smartphones now have proven applications in higher education institutions (Johnson et al., 2013), and learners already use those devices in their daily lives, educators have been trying to find ways to effectively utilize those devices for educational purposes. The potential of mobile technologies is already being shown in hundreds of projects at higher education institutions (Johnson et al., 2013).

Hsu and Ching (2012) have stated the main advantages of mobile technologies for learning as "(a) mobility, the small sizes of the devices, making them highly portable, which enhances user mobility and easy access to mobile devices; (b) computing power, relatively strong computing power, which enables users to complete tasks on small devices as effectively as on larger and less portable devices; and (c) connectivity, always-on and stable Internet connectivity with high bandwidth, which allows for instant access to large amounts of information and real-time communication regardless of location." Therefore, with these abilities, there are various new ways in using those devices for educational purposes. Thanks to smartphones, students are able to learn not only at schools or homes, but also while moving, they are able to download the course materials such as course videos, audios, texts and study anywhere anytime, communicate with their peers or instructors to discuss any course content, make collaboration or cooperation with their colleagues via Internet, and utilize various features of features such as GPS, camera and audio recording through mobile apps for learning purposes.

# 2. Mobile Application Development

## 2.1.Major Mobile Operating Systems (Platforms)

Mobile operating systems (platforms) are the operating systems to run the mobile devices such as smart phones, tablets, PDAs and other hand held devices. Mobile devices consist of various features such as touch screen, cellular, Bluetooth, Wi-Fi, GPS navigation system, camera, speech recognition, voice recorder, music player and so on (Ram, 2016).

Various operating systems (OS) are available in the market with different mobile devices. Five major mobile operating systems are Google's Android, Apple' iOS, Microsoft's Windows Phone and RIM's Blackberry OS. With respect to their market shares, as for February 2015 data, while Android possess 56% of the market and iOS has 39% share, Windows Phone's ratio is only 5% and the share of the rest platforms including Blackberry OS is no more than 0.1% (Divya & Kumar, 2016). Therefore, most of the smartphone market (95%) belongs to Android and iOS. For this reason, this paper examines these 2 platforms in detail, as followed.

**Android:** Android developed by Google has the biggest market pie (Divya & Kumar, 2016) and is the most widely used mobile platform. Android is a complete set of software or software stack for mobile devices which includes an operating system, middleware and key mobile applications (Ram, 2016). It is based on the Linux Kernel, developed by Google and designed firstly for touchscreen mobile devices such as smartphones and tablets (Divya & Kumar, 2016). It was unveiled in 2007 along with the foundation of Open

Handset Alliance, which is a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices. Android consists of an open source code released by Google under the Apache License (Ram, 2016). This license allows device manufacturers, wireless carriers and developers to freely modify and distribute the software. Its native language is Java. The members of the Open Handset Alliance are technology companies including Google, device manufacturers such as HTC, Sony and Samsung, wireless carriers such as Sprint Nextel and T-Mobile, and chipset makers such as Qualcomm.

**IOS:** iOS is one of the best operating system created and developed Apple Inc. and its native language is Objective-C. It is right behind of Google's Android by having 39% market share (Divya & Kumar, 2016). iOS is Apple's mobile version of the OS X operating system which shares the Darwin foundation and various application frameworks. This operating system is used in iPhone, iPad, and iPod touch. The summary information of major mobile platforms is presented in Table 1.

Table 1. Major Mobile Platforms (Ribeiro & Silva, 2012)

| Vendor | Operating System | Programming Language | Development Environment | Application Store |
|---|---|---|---|---|
| Google and Open Handset Alliance | Android | Java | Eclipse | Google Play |
| Apple | iOS | Objective-C | Xcode | iPhone App Store |
| Microsoft | Windows Phone | Visual C#/C++ | Visual Studio | Windows Phone Marketplace |
| RIM | Blackberry OS | Java | Eclipse | Blackberry App World |

## 2.2. Existing Approaches to Developing Mobile Applications for Educational Purposes

There are various studies concentrating on the design and development aspects of m-learning applications (or mobile applications for educational purposes). Many of the approaches proposed in these studies have focused one or several sides of m-learning design and missed some respects of m-learning application design such as learners' requirements (Al-Harrasi et al., 2015) and choosing a suitable platform for developing m-learning applications, which is indispensable parts of developing a mobile application. Besides, most of them failed at providing the detailed steps of developing an m-learning application. The m-learning design framework provided by Parsons and his colleagues (2007) consists of three categories of components: design issues, dimensions of the learning context, structural factors, and their instantiation and objectives. The components of the "design issue" category are user role and profile, work on the move, interface design, media types, collaboration support. Through these elements, the dimensions of other categories are formed. On the other hand, this framework shows some drawbacks. For example, it does not give a detailed process for developing the application, and also it does not provide a step considering the functional requirements for learners, learning content and the application (Al-Harrasi et al., 2015). Another approach provides a framework for mobile learning design requirements (Nordin et al., 2010). The significance of this approach is that they emphasized the learning aspects in developing m-learning applications. It consists of four main components which are theories of learning, generic mobile environment, mobile learning context, learning experience and learning objectives. Theories of learning (e.g., cognitivism,

behaviorism and constructivism) are critical not only for designing learning materials but also for interaction approach (Al-Harrasi et al., 2015). The generic mobile environment encompasses 1) user dimension, 2) mobility of content, device and stakeholders, 3) mobile interface design of the application, 4) media types of content and 5) support services for users. The next element, mobile learning context consists of identity, learner, activity, spatial-temporal, facility and collaboration factors. Learning experience corresponds to organized contents, outcome and goals and objectives, story, challenge and social interaction. Lastly, learning objectives means newly acquired or improved skills including both of social and team skills as well (Nordin et al., 2010). They stressed that m-learning applications should focus on the effect of application design on the learners since learners seek for satisfactory and attractive experience together with acquiring information and that goals and objectives considerably important due to the fact that they provide a direction to learners for learning.

Another approach for developing an m-learning application is ADL M-learning Framework based on ADDIE model, as shown in Figure 1 (Berking et al., 2012). This approach benefits from Rapid Prototyping or Agile Model and ADDIE model (Analysis, Design, Development, Implementation and Evaluation) (Berking et al., 2012), which is an Instructional System Design (ISD) approach, and which enables to integrate pedagogy, learning theories, and other instructional design principles (Koneru, 2010). Besides, it emphasizes the importance of iteration among the steps and m-learning design considerations and learning theories and strategies in both macro and micro level. However, this approach does not give clear explanations about m-learning design considerations, and it does

not include any technological or technical considerations about the application, such as the type of application and the platform in which the application is coded, in the process of developing an m-learning application.

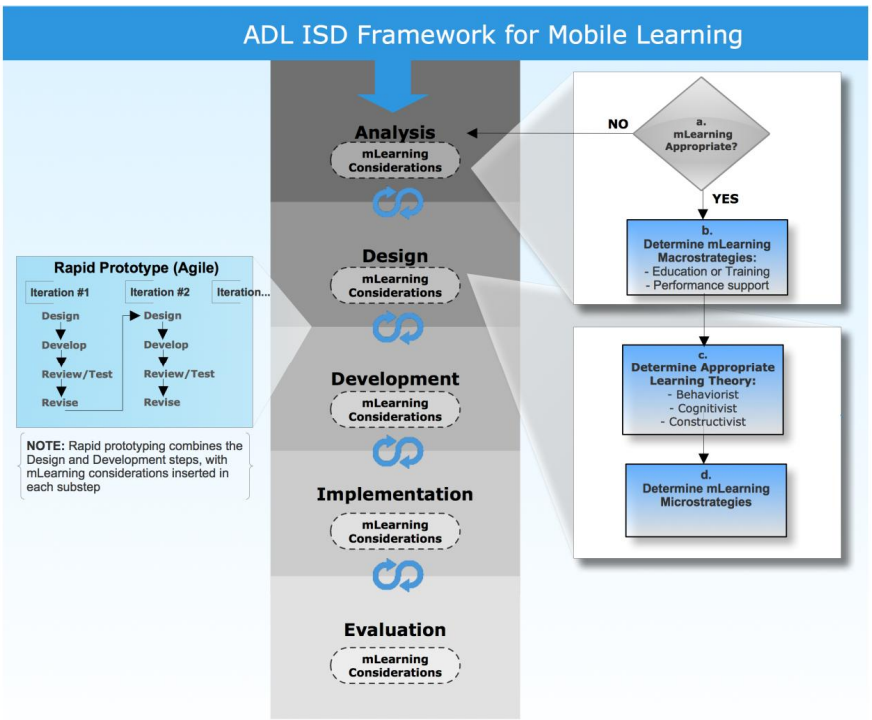



Figure 1. ADL Mobile Learning Framework (Berking et al., 2012)

Al-Harrasi, Al-Khanjari, and Sarrab (2015) also proposed a design approach for m-learning applications (see *Figure 2*). They have investigated 8 different approaches from the literature and provided a new approach because of the insufficiency of the existing approaches for developing an m-learning application (Al-Harrasi et al., 2015). However, they also fail at sufficiently considering technological aspect even if they inserted

"technology infrastructure" component into their m-learning application development model. According to their model, the tasks should be performed in the technology infrastructure phase are to identify the targeted mobile devices, the target platform (e.g.: Android and IOS) and type of app (web or native) and the type of database for storing the content and other information such as learner information.

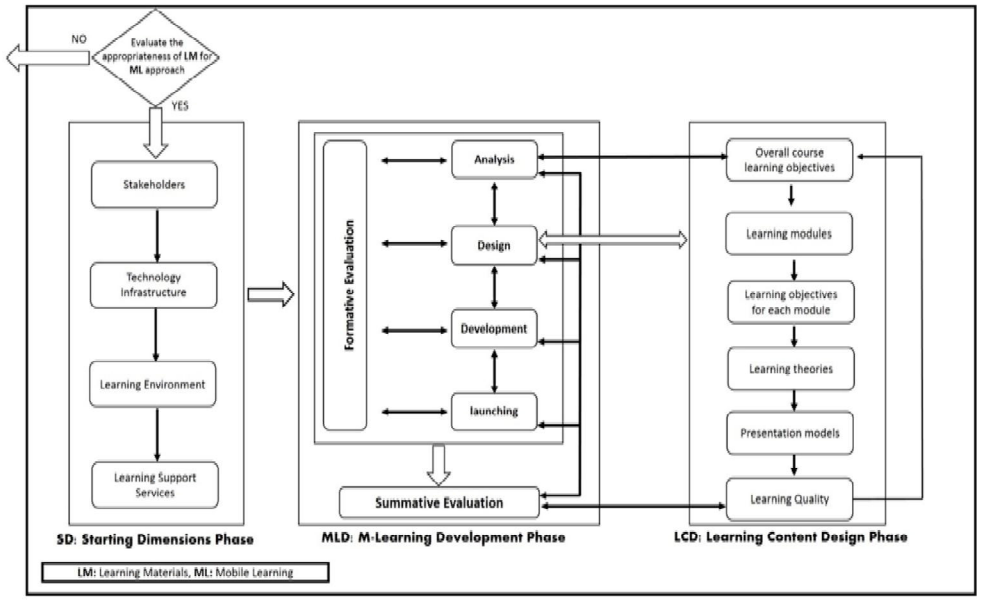



Figure 2. Design approach for m-learning application (Al-Harrasi et al., 2015)

On the other hand, they do not present any information or task about which kind of application (e.g.: native, web, hybrid app, which will be discussed later) should be developed in which situation and about which platform or development tool should or can be used to develop an m-learning

application. This is highly crucial because each platform or development tool has different and unique characteristics that can give negatively or positively impact to the process of an m-learning application development. For example, choosing to develop a native application, which provides the best performance and user experience, may have some drawbacks such as high cost due to the required expertise on the target mobile platform (Table 1). Also, in the case of the classroom environments, in which the possibility of students' having mobile devices of different platforms such as Android and iOS, the developer has to develop the application separately for each platform, which means additional time and the requirement of expertise on all those target platforms. This is almost impossible for educators. As a solution to this challenge, in order to provide an application for multi-platforms, which means making the application available for all students, the educators can develop either a hybrid app or web app, but they also have pros and cons. For instance, web apps cannot reach the mobile device's some hardware features such as camera and GPS, and this causes the limitation of the applications. So, for example, if taking a picture via the m-learning application is an indispensable part of the learning process, then choosing to develop a web app will be an improper choice.

Nevertheless, this study will utilize this approach (Al-Harrasi et al., 2015) and ADL ISD Framework (Berking et al., 2012) with some modifications according to the other sources in literature and the researcher's need.

## 2.3.Challenges in Mobile Application Development

Before constructing a model for developing a simple educational mobile application, it is highly crucial to realize why developing a normal mobile application is a difficult work. Therefore, this section will discuss these challenges.

**The incongruity of hardware utilities:** The existence multiple standards of mobile platforms are one of the challenges in developing mobile apps. Even with the same platform, various smartphones have different hardware specifications such as RAM, CPU, Screen Size and storage capacity (Kumar et al., 2016). Furthermore, there are different standards at the operating system level, such as fragmentation on Android devices with different screen resolutions (Phyo, 2014).

**Heterogeneity of platforms:** Each platform uses its own programming language and provides different development tools to develop an app. For example, while Android uses Java, iOS uses Objective-C, as shown in Table 1. This causes that a mobile app developed for Android does not work on iOS, which means incompatibility among platforms because of having different languages (Kumar et al., 2016; Phyo, 2014; Ribeiro & Silva, 2012). This lack of compatibility pushes the developers to rewrite the application for each one of the target platforms increasing the effort and the time to market of that application since if a someone wants to develop an iOS app, then he/she needs high technical skills in Objective-C, and similarly, if an Android app is wanted, then he/she has to have Java expertise. Therefore, developing a mobile app for each platform needs another language expertise, which is a one of the biggest challenges in developing apps. In order to cope

with this challenge and to develop a mobile application properly working in various platforms, some tools (cross-platform development tools) have been created. Nevertheless, if the full performance is expected from a mobile app, then the developer has to be an expert on each specific language of that target platforms.

**Frequent Version Releases:** Android uses an open source OS, but, there is fragmentation across each version of Android, and some manufacturers modify the OS source code according to their hardware specifications. This causes multiple standards, which means limited portability from one version to another and more complexity for application development. The issue of frequent version releases of a mobile operating system is another challenge. Developers must learn different programming languages and APIs for those fragmented platforms and keep up to date with software development kit (SDK) updates. Each version may provide a platform is different from the previous one, in terms of tools and user interface design necessitating the challenge of learning new development techniques. (Phyo, 2014).

Due to these challenges, the developers are provided alternative ways for developing mobile apps. They are provided a chance to develop a number of kinds of applications, which are commonly classified as *Native App, Hybrid App* and *Web App*, and a variety of type of development environments to develop those applications, which will be discussed later.

## 3. Evaluation Matrix for CPS

This study intends to construct a model for developing a simple educational mobile application. However, it will test and validate this model by conducting a case study, meaning that it will develop a mobile application for educational purposes via this model. The subject of this app will be Evaluation Matrix, which already exists in real-life. It is one of the tools used in *Creative Problem Solving* systems, which are utilized in order to increase the creativity competence of individuals. In order to understand the intended app, recognizing its context is considerable critical. For this reason, this section will provide information about firstly CPS, then the tools used in CPS and lastly, the evaluation matrix itself.

### 3.1.Creative Problem Solving (CPS)

Before explaining Creative Problem Solving in detail, it is important to understand its constructive elements, which are *creative*, *problem* and *solving*. CPS creators view *creative* as having an element of newness, *problem* as being relevant to the one who creates solutions and *solving* as creating ways to answer or to meet or satisfy the problem, adapting yourself to the situation or adapting the situation to yourself (Noller 1979).

CPS can be viewed as "a process, a method, a system for approaching a problem in an imaginative way resulting in an effective action" (Noller, 1979). In the book of *Creative Approaches to Problem Solving: A Framework for Innovation and Change*, Isaksen and his colleagues (2011) defines CPS as follow:

> CPS is an applicable framework for organizing specific tools to help design and develop new and useful outcomes. The structure of CPS

> provides an organized system. Using the system involves applying productive thinking tools to understanding problems and opportunities; generating ideas; and evaluating, developing and implementing potential solutions. The system includes the framework of components, stages, phases, and tools, as well as considering the people involved, the situation or context, and the nature of the content or the desired outcome. CPS enables individuals and groups to recognize and act on opportunities, respond to challenges, and overcome concerns (p.26).

### 3.2.Tools used in CPS

In CPS, individuals are exposed to the divergent and convergent thinking processes in order to solve a problem creatively (Treffinger et al., 2006), and creativity emerges during this process. Therefore, CPS offers various kinds of convergent and divergent tools to increase the chance of happening creativity. Divergent Thinking tools are used for creating various ideas and options while convergent thinking tools are used for evaluating ideas and options, and making. While practitioners of CPS use divergent tools for gathering data effectively, they use convergent tools to evaluate whether a purpose, challenge, or opportunity is appropriate for their situation (Creative Education Foundation, 2014). Divergent and convergent thinking must be in balanced for successful implementation of CPS (Treffinger & Isaksen, 2005). Therefore, the tools promoting divergent and convergent thinking and the process including deciding which tools are most appropriate to the context should be well-developed. Higgins (2006) categorized a lot of

thinking tools as divergent thinking tools containing Brainstorming, Forced Connection Method and Attribute Listing and convergent thinking tools including HIT, Highlighting, ALU (Advantage, Limitation, and Unique Qualities), PMI (Plus, Minus and Interesting), and Evaluation Matrix.

Figure 3 is an example of the system which utilizes the divergent and convergent tools in order to promote CPS in an online environment. This system called as CPS$^3$ (Creative Problem Solving Support System), developed by Lim and his colleagues (2009), and implemented many times on college (Lim et al., 2009, 2011, 2012). This figure is important in such a way that it briefly shows that how a CPS-supported system provides various convergent and divergent thinking tools according to each phase of the CPS. It benefits from Attribute Listing, Brainstorming, HIT, PMI, ALU and Evaluation Matrix.

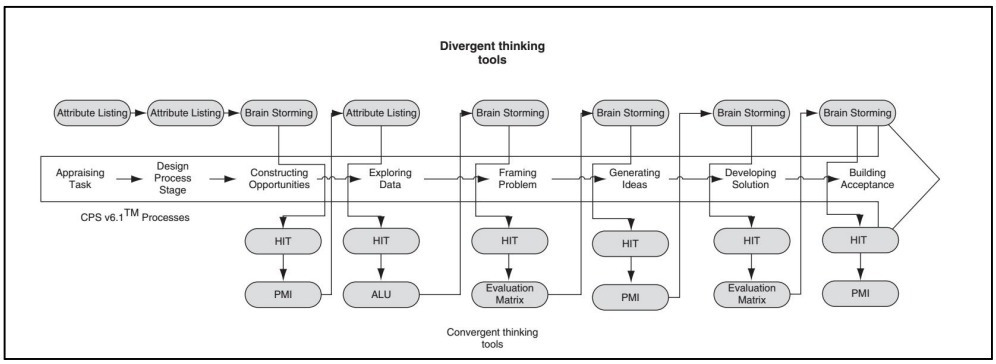



Figure 3. An example to CPS Framework implementation into a system: CPS$^3$ (Lim et al., 2012)

### 3.3.Evaluation Matrix

With evaluation matrix, the practitioners of CPS systematically evaluate each options, possibilities or solutions with respect to specific criteria. They list the possible ideas/solutions on one axis of a grid and puts important criteria to be evaluated on to the other axis (Lim, 2013). This tool helps them guide judgment and selection of options (Treffinger, 2007) by comparing and contrast those items.

Stead and Dorval (2001) has utilized evaluation matrix tool in an actual work setting. They concluded that thanks to the evaluation matrix tool, they could realize colleagues' evaluations and understanding on an idea or solution, and this caused to sensible, positive, co-operative, energetic and productive dialogue since the evaluation matrix enabled them to create shared understanding. Some examples of Evaluation Matrix are as follow:

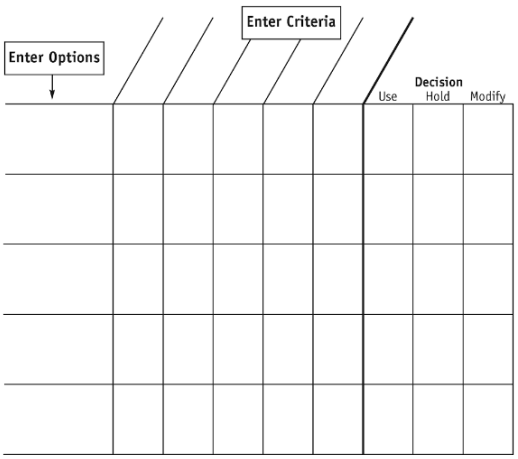



Figure 4. Evaluation Matrix, Example 1 (Treffinger, 2000)


Figure 5. Evaluation Matrix, Example 2 (Creative Education Foundation, 2014)

Treffinger (2000) also provided guidelines for effectively use of Evaluation Matrix tool. According to this guideline, 1) the practitioner should take each criterion and evaluate all the options on it before proceeding to the next criterion; 2) the practitioner should remember that the goal is not just to find "one winning choice", but to find the strongest or most effective solution, which might represent several options – modified, combined, used together, or used sequentially; 3) the practitioner should save more complex variations (such as weighting each of the criteria differently) for complex problems and experienced groups; 4) the practitioner should be certain that the criteria are relevant and important for all the options which she/he will be examining.

Item 1. 설계대안 장단점 분석표

각 제한조건을 충족하는 정도에 따라 3(충족 정도가 높음) 2(충족 정도가 보통) 1(충족정도가 낮음)으로 평가하시오

| Item/Criteria | 평면에서 전후 이동 속도가 빨라야 한다. O | 받침대를 기준으로 위치조정이 정확해야 한다. O | 골프공을 안정적으로 잡을 수 있어야 한다. O | 골프공을 안정적으로 운반 할 수 있어야한다. O | 골프공을 안정적으로 놓을 수 있어야한다. O | 기능을 모두 수행하고 복귀하는 시간이 빨라야 한다. O | 로봇의 질량이 최소화 되어야 한다. O | 제작이 용이해야한다. O | 제조원가가 최대한 적게 들어야한다. O |
|---|---|---|---|---|---|---|---|---|---|
| 설계대안 1 | 2.25 | 1.625 | 2.125 | 1.125 | 1.125 | 2.125 | 2.375 | 1 | 1 |
| 설계대안 2 | 2.125 | 1.625 | 1.5 | 1.75 | 1.5 | 1.875 | 1.375 | 1.75 | 1.5 |
| 설계대안 3 | 1.75 | 1.5 | 2.25 | 1.625 | 1.625 | 1.625 | 1.5 | 1.375 | 1.75 |
| 설계대안 4 | 1 | 1.375 | 1.75 | 2.375 | 2.25 | 1.125 | 1 | 1.25 | 1.25 |
| 설계대안 5 | 2 | 2.375 | 2 | 2.75 | 2.125 | 1.875 | 1.875 | 1.5 | 2 |
| 설계대안 6 | 2.5 | 2 | 2.75 | 2.25 | 2.625 | 2.625 | 2.5 | 2.5 | 2.25 |
| 설계대안 7 | 2.625 | 2.125 | 1.75 | 2.75 | 2.125 | 2.625 | 1.25 | 1.25 | 2 |
| 설계대안 8 | 2.5 | 1.75 | 2.75 | 2.625 | 2.875 | 2.75 | 2.25 | 2.25 | 2.375 |

Figure 6. Actual usage of Evaluation Matrix in S$^3$CPS system (Lim et al., 2016)

Lim and his colleagues have conducted study on CPS-supporting systems. Some of their studies (2014, 2015), which investigates S$^3$CPS (Smart Support Systems for Creative Problem Solving) providing CPS-promoted online environment, have revealed that the learners require additional features and improvements of the evaluation matrix tool of the system. Those requested additional features and improvements encompass the data extraction feature enabling users to extract the data on Evaluation Matrix into some formats such as Excel file format, copy and paste feature between Evaluation Matrix and Excel, enabling automatically adding table, making possible to write f(x) functions on the cells, an alternative system to the boredom of statistics terms use (such as mean, numbers, weight) on Evaluation Matrix, and a mobile app version of the system. Besides, in their next year's study (2016), which was conducted on engineering education context, even if most students had provided positive feedbacks on evaluation matrix tool, such as being seen of the evaluation results at one glance and the

reliability of the evaluation results, it is also found out that the evaluation matrix tool showed weakness on supporting students' discussion for decision-making.

# CHAPTER III. RESEARCH METHODOLOGY

## 1. Research Method

The purpose of this study is to construct a model for developing a simple educational mobile application. It employs Design and Development Research – Model Research (Type II) (Richey & Klein, 2007). Model studies are the most generalized of design and development studies. “The ultimate object of this research is the production of new knowledge, often in the form of a new (or an enhanced) design and development model” (Richey & Klein, 2007). There are three types of model research; model development, model validation and model use. This study utilizes model development combined with model validation due to the fact that they can be integrated (Richey & Klein, 2007). Model validation is a process that “demonstrate the effectiveness of a model’s use in the workplace...”, and external validity, one type of model validation, “confirms a model by documenting the impact of the model’ use” (Richey & Klein, 2007). Accordingly, after this study constructs an initial model for developing a simple educational mobile application on the basis of the literature review, it revises the model through the feedbacks derived from the interviews conducted with experts. Then, this study implements the case developing a product by following that revised model. It would be much healthier if the case study is conducted by another person rather than the researcher with respects to its validity. However, since there is no such person, the model is validated by the researcher himself, implying the case study being conducted by the researcher. During the case

study, the aim is to create the most optimized product, so that the model is being experienced several modifications again whenever needed. Thus, the final revised model for developing an educational mobile application is produced whereby the case study. Model validity is ensured by the researcher through the case study's usability questionnaire and interviews disclosing the impact of the product on the learners. To sum up, the steps followed and the corresponded methods applied in this study are presented in Table 2.

Table 2. Research steps and methods used in the study

| **Components** | **Steps** | **Method** | **Output** |
|---|---|---|---|
| Model Development | Literature Review | Literature Review | Model and Components |
| | Expert Review | Interview | Revised Model |
| Model Validation | External Validation | Case Study (Usability questionnaire and interviews conducted with learners) | Final Revised Model |

## 2. Research Procedure

The research procedure implemented in this study is as followed. First, the research problem, the necessity of the research and the purpose of the research have been defined. Second, a literature review has been conducted on the use of mobile devices for educational purposes. Literature review has been also performed for mobile applications, type of mobile

applications, operating services, development approaches used for developing mobile applications and challenges encountered in developing a mobile application. In addition, another literature review has been conducted related to Evaluation Matrix tool and the system using Evaluation Matrix tool. Third, the initial Model for Developing a Simple Educational Mobile Application (MODSEMA) has been developed according to the related literature review. After that, interviews have been conducted with three mobile application development experts, and the initial MODSEMA has been revised on the basis of the interview results. Fifth, a case study has been conducted, which develops a simple mobile application named "Evaluation Matrix" app by following the revised MODSEMA. The case study has been performed as followed. In *Analysis* phase of MODSEMA, the required analyses has been conducted. In *Design* phase, the prototype of "Evaluation Matrix" app has been developed. Then, this prototype has been evaluated by 3 experts and 3 representative users, who are graduate students, by employing different usability evaluation methods and revised by the researcher according to those evaluation results. On the basis of this prototype, in *Development* phase, "Evaluated Matrix" app has been coded and developed by the researcher in the mobile application development environment that is determined in analysis phase. After that, the developed app has been evaluated by 2 experts and 5 representative users, and revised ultimately by the researcher according to those evaluation results. In *Implementation & Evaluation* phase, the app has been implemented in a graduate class having 16 students in which $S^3$CPS was being implemented and Evaluation Matrix tool was needed, that means the app was tested in the learning environment where it is needed. Right after it was implemented in

the class, each participated learner filled a questionnaire. Lastly, semi-structured interviews have been conducted with three available learners who attended to class implementation. Through the questionnaires and interviews, the researcher has tried to find out and interpret the usability of “Evaluation Matrix”. Through the conducted case study in which “Evaluation Matrix” app has been developed and tested in a classroom setting, there have been made ultimate revisions on MODSEMA whenever it is needed. Thus, MODSEMA has been tried to be optimized, and “Evaluation Matrix” app has been developed in a best way as much as possible. Then the final MODSEMA has been proposed for educators, instructors and developers, who are interest in developing educational mobile applications. Lastly, the research results have been synthesized and the researcher has made discussions and interpretations on MODSEMA and its validity that is measured indirectly through the usability questionnaires and interviews conducted in the case study, since it was not possible to measure it directly. Overall research procedure is summarized in Figure 7.

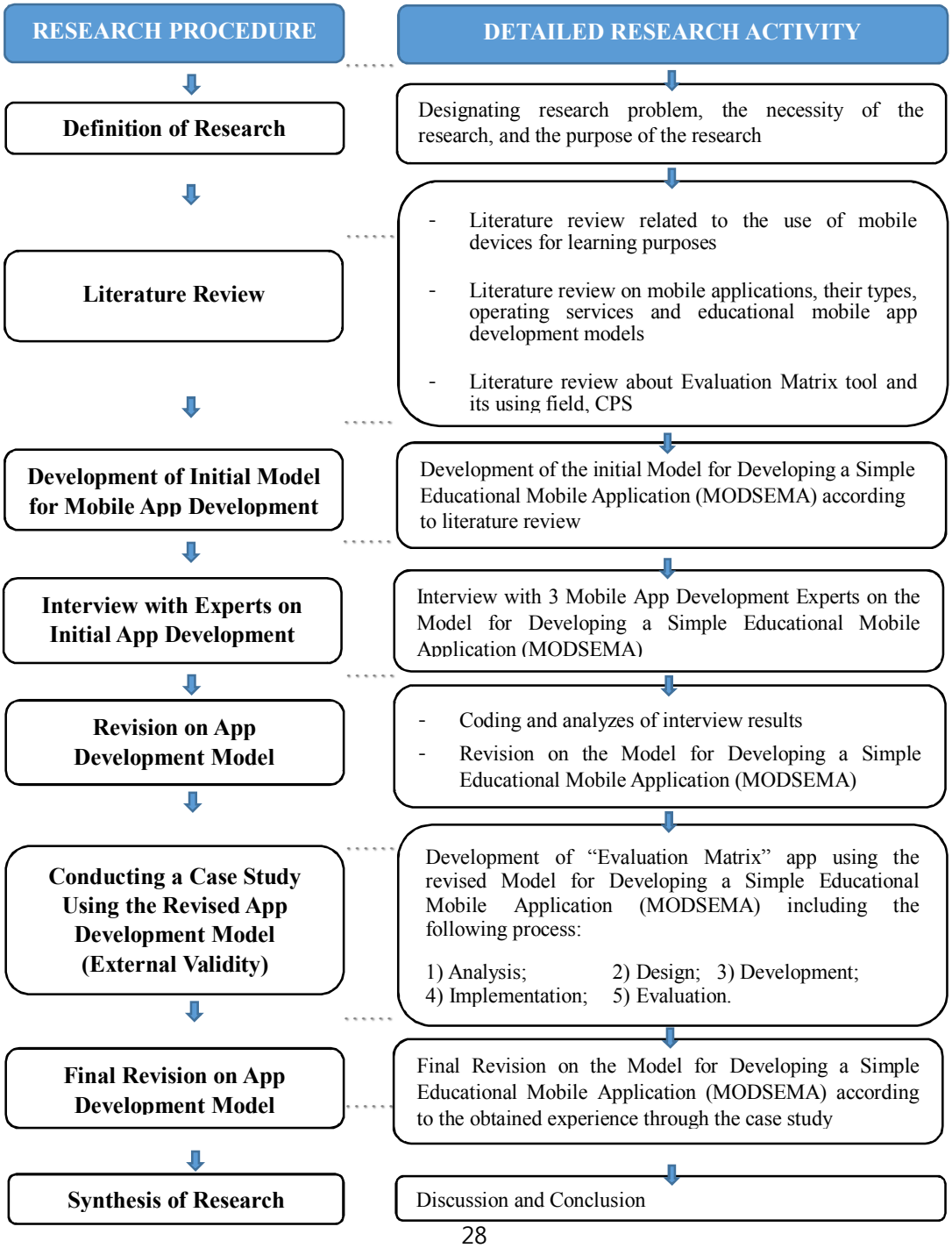

RESEARCH PROCEDURE
DETAILED RESEARCH ACTIVITY
Definition of Research
Designating research problem, the necessity of the research, and the purpose of the research
Literature Review
- Literature review related to the use of mobile devices for learning purposes
- Literature review on mobile applications, their types, operating services and educational mobile app development models
- Literature review about Evaluation Matrix tool and its using field, CPS
Development of Initial Model for Mobile App Development
Development of the initial Model for Developing a Simple Educational Mobile Application (MODSEMA) according to literature review
Interview with Experts on Initial App Development
Interview with 3 Mobile App Development Experts on the Model for Developing a Simple Educational Mobile Application (MODSEMA)
Revision on App Development Model
- Coding and analyzes of interview results
- Revision on the Model for Developing a Simple Educational Mobile Application (MODSEMA)
Conducting a Case Study Using the Revised App Development Model (External Validity)
Development of "Evaluation Matrix" app using the revised Model for Developing a Simple Educational Mobile Application (MODSEMA) including the following process:
1) Analysis; 2) Design; 3) Development;
4) Implementation; 5) Evaluation.
Final Revision on App Development Model
Final Revision on the Model for Developing a Simple Educational Mobile Application (MODSEMA) according to the obtained experience through the case study
Synthesis of Research
Discussion and Conclusion

Figure 7. Research procedure and associated research activities

# CHAPTER IV: RESULTS

## 1. The Initial Model for Developing a Simple Educational Mobile Application (MODSEMA)

The initial MODSEMA (Figure 8) is derived from literature review, especially benefits from Berking et al.'s ADL Mobile Learning Framework (2012), Al-Harrasi's m-learning design approach (2015) and user-centered design (Abras et al., 2004). As suggested by Berking et al. (2012), it employs ADDIE Model containing analysis, design, development, and implementation and evaluation phases. The main aim of analysis phase is to conduct all required analysis, and it consists of 7 steps respectively: 1) Determining the target Operation Service(s) (Android, IOS, etc.), 2) Determining the type of application: native, hybrid or web app, which is not mentioned by most development models, 3) Determining the development environment or tool according to the type of application and your programming experience, which is a missed step by other development models, 4) Identifying the aim, scope and needs of the app, 5) Identifying the appropriate learning theory, as referred in the models of Berking et al. (2012) and Nordin et al. (2010), and the learning environment as mentioned by Al-Harrasi et al. (2015), 6) Identifying the required features/functions for the app, and 7) Identifying the user interface design requirements for the app. In design phase, the purpose is to develop the prototype of an app according to the feedbacks derived from usability evaluations in order to ensure user-centered design (Abras et al., 2004). Therefore, it recommends developers to

create two prototypes having UI elements and features. Thus, the users are able to choose the one they favor the most. Each prototype development follows the steps including concept design, in which the general conceptual design of the intended app is established, prototype development in which the prototype is designed by using an analog or digital tool like paper or MS PowerPoint, expert validation where experts evaluate the prototype, user testing where the representative users make evaluations on the prototype, and revision in which the prototype is modified and enhanced through the results of usability evaluations. Meanwhile, the most favorable prototype is selected by representative users, and the developer continues his/her app development process with the chosen one. In development phase, the goal is to develop and code the app in the initially selected development environment and to revise it on the basis of feedbacks coming from usability evaluations. Lastly, the purpose of implementation and evaluation phases is to test the finalized app in a learning environment to understand to what extent the app serves its duty well via observations and usability questionnaires and interviews.

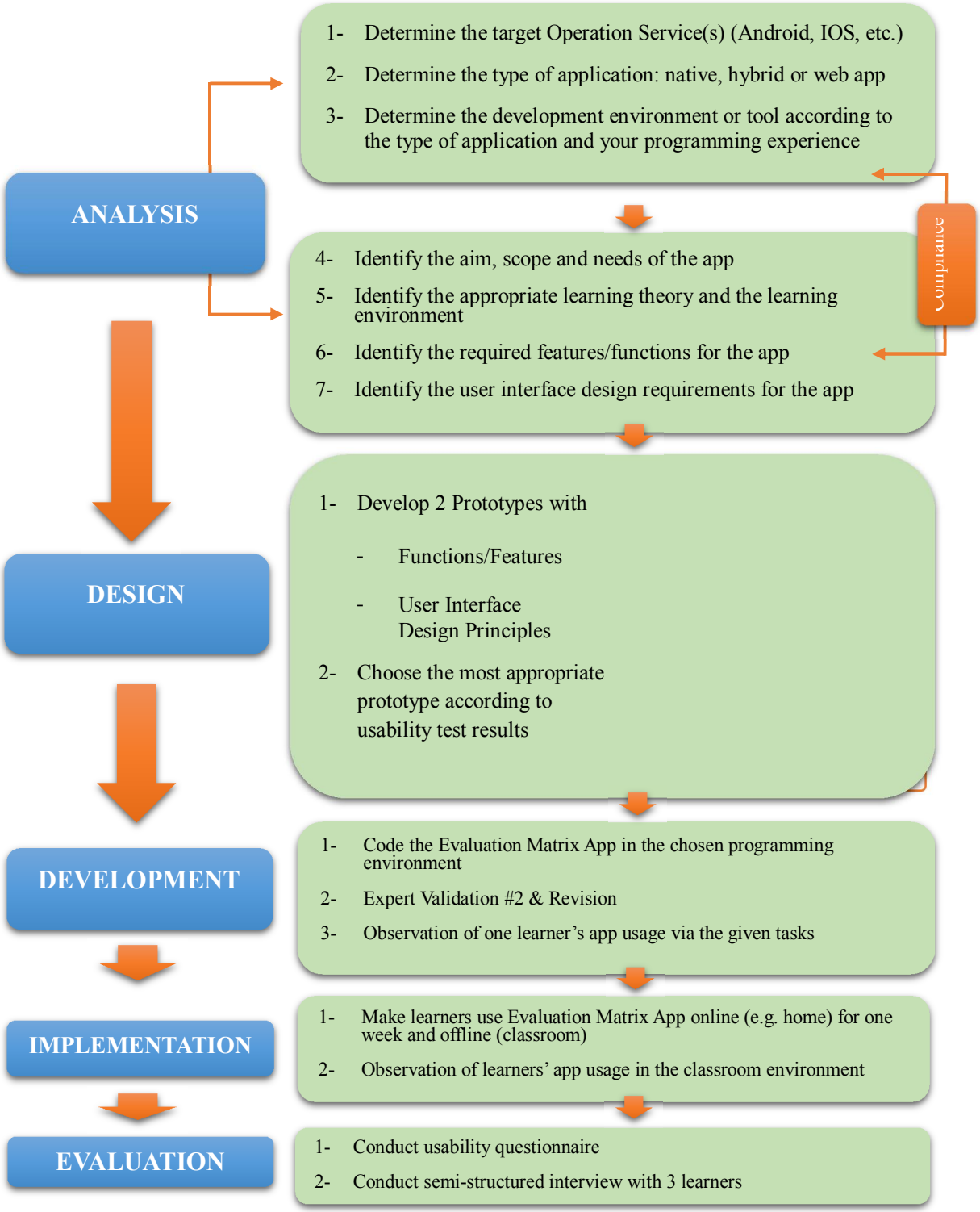

ANALYSIS
1- Determine the target Operation Service(s) (Android, IOS, etc.)
2- Determine the type of application: native, hybrid or web app
3- Determine the development environment or tool according to the type of application and your programming experience
4- Identify the aim, scope and needs of the app
5- Identify the appropriate learning theory and the learning environment
6- Identify the required features/functions for the app
7- Identify the user interface design requirements for the app
Compliance
DESIGN
1- Develop 2 Prototypes with
- Functions/Features
- User Interface Design Principles
2- Choose the most appropriate prototype according to usability test results
DEVELOPMENT
1- Code the Evaluation Matrix App in the chosen programming environment
2- Expert Validation #2 & Revision
3- Observation of one learner's app usage via the given tasks
IMPLEMENTATION
1- Make learners use Evaluation Matrix App online (e.g. home) for one week and offline (classroom)
2- Observation of learners' app usage in the classroom environment
EVALUATION
1- Conduct usability questionnaire
2- Conduct semi-structured interview with 3 learners

Figure 8. The initial Model for Developing a Simple Educational Mobile Application (MODSEMA)

## 2. Interview with Experts on the initial MODSEMA

In order to evaluate the *Model for Developing a Simple Educational Mobile Application (MODSEMA)*, a semi-constructed interview has been conducted with three experts in mobile app development. The interview has been implemented online via Skype or Facebook Video Chat. One day before the each interview, MODSEMA has been sent to each expert to inspect and think about it for a time. The language used in interviews was Turkish and each interview lasted between 1 hour and 90 minutes. After they are informed about the study and its purpose, the interview has been started. Besides, each interview has been recorded and the researcher has analyzed the recorded audio files after all interviews are finished. The demographic information of those experts is as followed:

Table 3. Demographic information of experts

| **Experts** | **Occupation** | **Expert Fields** | **Experience in app development** |
|---|---|---|---|
| Expert 1 | Freelancer | App development, Software development | 7 years |
| Expert 2 | Master's student in computer sciences | App Development (Android) Machine Learning | 3 years |
| Expert 3 | Mobile App Developer in a | App Development (IOS) | 2 years |

| | | | |
|---|---|---|---|
| | Turkish Bank | Software development | |

Expert 1 is now a freelancer, but has worked in various big Turkish corporates including communication and transportation companies, and has seven years' experience in this field. He has performed various app development methodologies including Agile, Waterfall and Spiral, and has worked with app development teams of various sizes. In addition, he has experience in all stages of app development cycles including analyzing, designing, coding and testing. Therefore, it was possible to acquire very useful feedback from him. Expert 2 is a master's student in computer sciences field in a German university. He has three years' experience in app development, and he has developed various Android apps since undergraduate degree. However, he has no experience with large-sized app development teams. Expert 3 is currently a mobile app developer in a Turkish Bank. Normally, he was working in the field of software development, but now he develops mobile apps for 2 years for the company. He works with a big app development team, having different members for different roles. He is expert in coding, but does not have sufficient information about designing an app since there is another team for app design in his company, as he said. The feedbacks and recommendations on MODSEMA, provided by the experts, are as followed.

Table 4. Expert recommendations on MODSEMA

| Stages of MODSEMA | Recommendation | Expert name |
|---|---|---|
| **Analysis** | Export function sets of the app, in detail | Expert 1 |
| | Provide information about the capabilities of the mobile app development environments/tools, such as explaining which native functions are supported | Expert 2 |
| | Provide information about the languages used by mobile app development environments/tools | Expert 2 |
| | Change the place of the first part with the second part of Analysis phase | Expert 2 |
| | Change the place of the first step with the second step | Expert 3 |
| **Design** | Use iterative process in prototyping in order to develop user-oriented prototype | Expert 1 |
| | Check Agile Development manifestos for better development | Expert 1 |
| | The representative users and experts may provide conflicting feedback. For instance, while one user likes one design, another user may dislike it. In this kind of cases, the developer should make a choice | Expert 1 |
| | Apply classical Agile method | Expert 2 |
| | Develop a prototype, and make an iteration through the feedbacks acquired from Expert Review and User Testing. For this, draw an arrow from the "revision" step to "concept development" step | Expert 2 |
| | The feedbacks coming from users and experts may have some conflicts. The developer should decide it. | Expert 2 |

| | | |
|---|---|---|
| | Provide different tools and methods for prototype development | Expert 2 |
| | Developing two prototype is loss of time. Also, even if two prototypes are developed, while one prototype will be chose, the other one will be thrown. The latter one may have favorable functions that the former does not have. In this situation, the liked functions/UI will be rubbish as well. Therefore, develop one prototype and make an iteration. | Expert 2 |
| | Performing app development through one prototype would be much logical in terms of effort and time. | Expert 3 |
| **Development** | The developer might not be completely successful. He/she might be not fully reflect the prototype to the real app, or he might not code the app as he/she thought. In this kind of cases, he/she has to find alternative way to deal with this challenge. Therefore, expert review should be conducted again in this phase. | Expert 2 |
| **Implementation** | It is much better to combine implementation and evaluation phases | Expert 1 |
| **Evaluation** | | |

The experts have approached positively on the model and have confirmed that this model consists of sufficient methods and steps for the development of a simple mobile application for educational purposes. They specifically highlighted the importance of iteration process and the communication between users and the developer(s) in app development process. The revised version of MODSEMA is as followed:

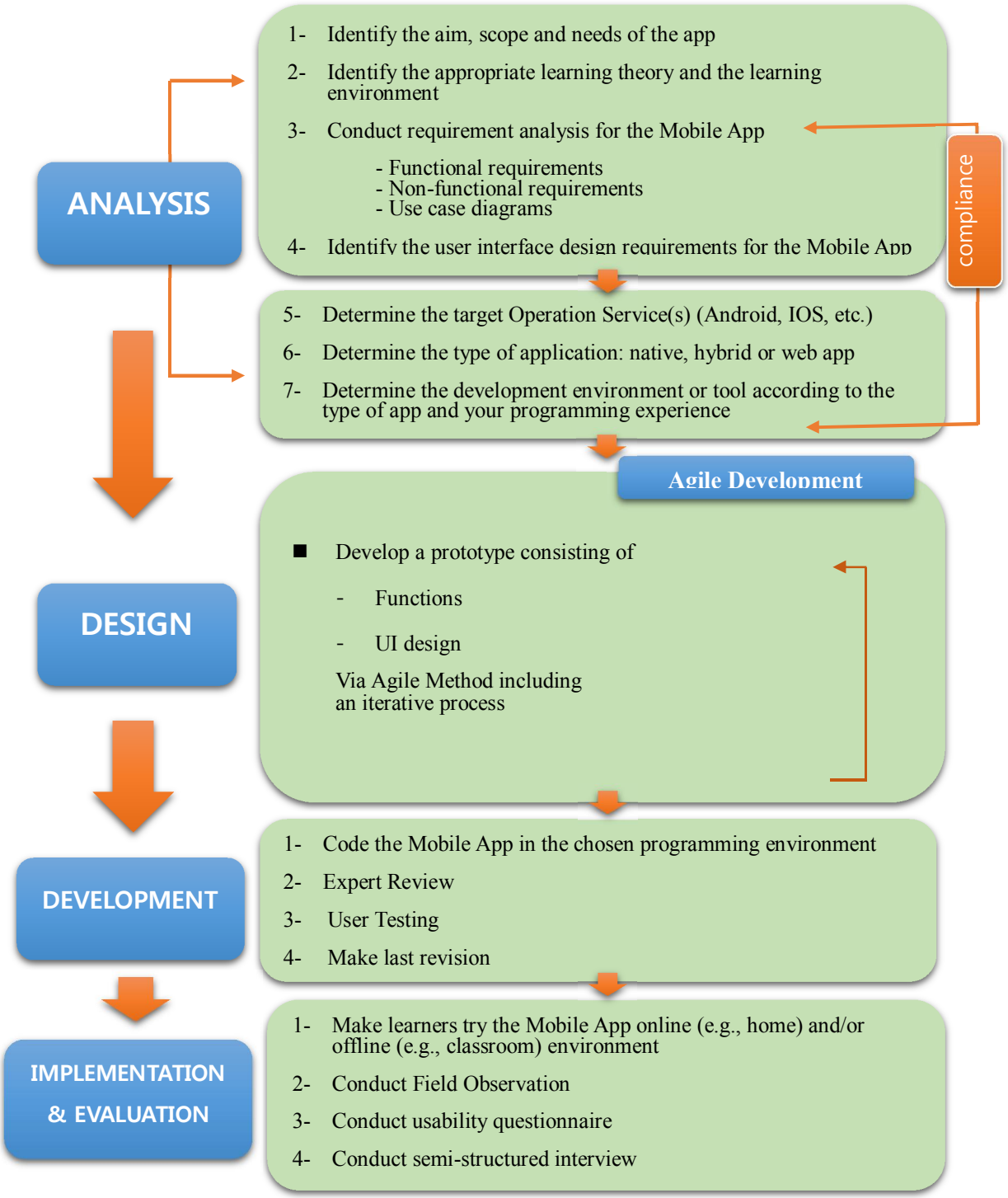

ANALYSIS
1- Identify the aim, scope and needs of the app
2- Identify the appropriate learning theory and the learning environment
3- Conduct requirement analysis for the Mobile App
- Functional requirements
- Non-functional requirements
- Use case diagrams
4- Identify the user interface design requirements for the Mobile App
compliance
5- Determine the target Operation Service(s) (Android, IOS, etc.)
6- Determine the type of application: native, hybrid or web app
7- Determine the development environment or tool according to the type of app and your programming experience
Agile Development
DESIGN
Develop a prototype consisting of
- Functions
- UI design
Via Agile Method including an iterative process
DEVELOPMENT
1- Code the Mobile App in the chosen programming environment
2- Expert Review
3- User Testing
4- Make last revision
IMPLEMENTATION & EVALUATION
1- Make learners try the Mobile App online (e.g., home) and/or offline (e.g., classroom) environment
2- Conduct Field Observation
3- Conduct usability questionnaire
4- Conduct semi-structured interview

Figure 9. The revised MODSEMA reflecting the expert reviews

# 3. The Revised MODSEMA

The researcher has employed ADDIE in the app development model (Figure 9) since ADDIE is suitable to apply in developing any educational software (Dick and Carey). There are many educational apps developed and models and frameworks proposed for developing an educational app by using ADDIE or a kind of ADDIE model (Orjuela et al., 2015; Jeon & Kim, 2016; Dennen & Hao 2014; Al-Harrasi et al., 2015; Ibrahim et al., 2016; Berking et al., 2012).

## 3.1.ANALYSIS

In order to develop a simple mobile application for educational purposes, the first stage is ANALSIS phase. In this phase, the developer deals with not only educational issues, but also handles with technical matters.

### 3.1.1. Identify the need, aim and scope of the app

In the *Analysis* phase, the first thing the developer needs to do is to identify the aim, scope and needs of the app, as suggested by Al-Harrisi et al. (2015). Determining why the app is needed justifies the app development, and designating the aim and scope provides the main functionalities of the app. This helps developer defining the detailed functional and non-functional requirements.

### 3.1.2. Identify the appropriate learning theory and

**environment**

The next step the developer performs is to identify the appropriate learning theory and environment as recommended step by Al-Harrasi et al. (2015), Berking et al. (2012) and Nordin et al. (2010). This step is required for designing effectively learning activities performed in a mobile app. Mobile learning technologies support various kinds of learning theories and the developer should carefully choose the most appropriate learning theory for utilizing in the mobile application.

Although there are various categorizations of mobile learning theories created by different researchers (Kukulska-Hulme, 2005; Naismith et al., 2004; Oberer, 2016; Stanton & Ophoff, 2013), this paper suggests Keskin and Metcalf (2011)'s extensive classification of learning theories for mobile technologies (Table 5), including the definition and focus of the theory and the related examples, provided in the following table. Apart from Keskin and Metcalf's classification, the researcher has inserted *Learning & Teaching Support* into the table, as proposed by Naismith et al., 2004, Oberer, 2016 and Stanton and Ophoff, 2013. The reason for adding this is that the mobile technologies in education are not only used to exploit learners for learning activities, but also are used in supporting learning and teaching without explicitly being part of the learning activities themselves. Education as a process depends on the coordination of learners and resources and mobile technologies as learning and supportive tools like mobile LMSs improve positively learning performance (Ayoma, 2012).

Table 5. Learning Theories Along with Mobile Technologies (Keskin & Metcalf, 2011)

| **Theories** | **Definitions** | **Focus** | **Examples with mobile technologies** |
|---|---|---|---|
| **Behaviorist Learning** | Learning has occurred when learners evidence the appropriate reinforcement of an association between a particular response and stimulus (Smith and Ragan, 2005) | **Information and content delivery in mobile learning**<br>- Language learning: Test, practices, quiz, listening-practice speaking<br>- Drill and feedback: Mobile Response System<br>- Content delivery by text messages. | - English learning applications<br>*SMS, MMS, Voice recorder software*<br>- Mobile Response System: *Qwizdom* |
| **Cognitivist learning** | Learning is the acquisition or reorganization of the cognitive structures through which humans process and store information (Good and Brophy, 1990) | **Information and content delivery in mobile learning**<br>- Using Multimedia learning (Dual code, Cognitive Load Theory): Images, audio, video, text, animations | - Multimedia (text, video, audio, animation, images)<br>*SMS, MMS, e-Mail*<br>*Podcasting*<br>*Mobile TV* |
| **Constructive learning** | Learning is an activity process in which learners construct new idea or concepts based on their current and past knowledge (Bruner, 1966) | **Context and content-dependent mobile learning**<br>- Questions for Exploration<br>- Cases and examples<br>- Problem-solved and Decision making applications<br>- Multiple representations<br>- Authentic contexts based-information database<br>**Collaboration and interaction in mobile learning**<br>- Collaboration and interaction between students<br>- Communication via mobile phones | - Handheld games<br>- Simulation<br>- Virtual reality<br>- Interactive Podcasting and SMS<br>- Interactive mobile TV and SMS |
| **Situated learning** | Learning is not merely the acquisition of knowledge by individuals, but instead a process of social participation (Brown et al., 1989). | **Social Context and Social participant-dependent mobile learning**<br>- Authentic domain activity<br>- Collaborative social interaction<br>- Cooperative activities<br>- Expert modeling<br>- Situated mentoring | - Natural science learning<br>- Medical education<br>- Multimedia museum<br>- Virtual experts by artificial intelligence tech. |

| | | | |
|---|---|---|---|
| | | - Workplace learning | - Mobile performance support system |
| **Problem-based learning** | Learning aims to develop students' critical thinking skills by giving them an ill-defined problem that is reflective of what they would encounter as a practicing professional (Koschmann et al., 1996) | **Problem based context and solved based content dependent mobile learning**<br>- Problems – Solutions<br>- Case-centered activities<br>- Collaborative social interaction | - Medical education<br>- Business administration<br>- Nursing<br>- *Simulations*<br>- *SMS*<br>- *MMS*<br>- *Voice response systems* |
| **Context awareness learning** | Context awareness means gathering information from the environment to provide a measure of what is currently going on around user an the device (Naismith et al, 2004) | **Context aware in mobile learning**<br>- Context-dependent content management<br>- Contextual event notification<br>- Context-aware communication<br>- Navigation and retrieval of learning materials<br>- User interface adapted according to time and location contexts | - Multimedia museum and gallery<br>- Pre-class podcasts<br>- *Films*<br>- *e-books*<br>- *Podcasting* |
| **Socio-cultural theory** | Learning occurs first through interpersonal (interaction with social environment) than intrapersonal (internalization) (Vygotsky, 1978). | **Social Context and Social participant dependent mobile learning**<br>- Mobile experts<br>- Community of practice<br>- Workplace learning<br>- Mobile communication | - Mobile performance support system<br>- Virtual experts<br>- Mobile forum, E mail<br>- Social network (Web 2.0 tools) |
| **Collaborative learning** | Learning is promoted, facilitated and enhanced by interaction and collaborations between students. | **Collaboration and interaction dependent mobile learning**<br>- Actively participation<br>- Social context<br>- Communication between peers via mobile phones. | - Mobile Assisted Language Learning<br>- Mobile Response System<br>- Mobile computer supported collaborative learning<br>- *Forum, Web 2.0 tools, email, mobile portal, games* |
| **Conversational learning** | Learning is in terms of conversations between different systems of knowledge (Sharples, 2002) | **Interaction and communication dependent mobile learning**<br>- Solving a problem<br>- Exploring an environment<br>- Communication between peers via mobile phones. | - Laboratory classes<br>- Field trip<br>- Mobile computer supported collaborative learning |
| **Lifelong learning** | Learning happens all the time and is influenced both by our | **Lifelong information and interaction with education content in mobile learning**<br>- Podcasting | - Social networks (Blogs, Wikipedia, Twitter, YouTube) |

| | | | |
|---|---|---|---|
| | environment and the particular situations we are faced with (Sharples, 2000). | - Information resources<br>- Mobile web site | - Podcast<br>- E-mail<br>- Mobile Forums |
| **Informal learning** | Learning is a process of learning that occurs autonomously and casually without being tied to highly directive curricula or Instruction (Vavoula, 2004) | **Information and interaction with education content in informal mobile learning setting**<br>- Mobile information resources<br>- Mobiles in a museum setting<br>- Field Trips<br>- Science Field Work | - Social networks (Blogs, Wikipedia, Twitter, YouTube)<br>- Podcast<br>- E-mail<br>- Mobile Forums |
| **Activity theory** | Learning occurs with three features- involving a subject (the learners), an object (the task or activity) and tool or mediating artefacts and human behaviour is situated within a social context that influences their actions (Vygotsky, 1987). | **User actions in social context dependent mobile learning**<br>- Actively participation<br>- Social context<br>- Activities | - Museum Art Gallery exhibit via SMS, polls, calling<br>- Mobile Games<br>- Multimedia |
| **Connectivism** | Learning is process of connecting specialized nodes or information sources (Siemens, 2004). | **Diversity of information sources in mobile learning**<br>- Connecting specialized nodes<br>- Information sources<br>- Facilitate continual learning environment<br>- Knowledge management activities<br>- Decision-making | - Social networks (Blogs, Wikipedia, Twitter, YouTube)<br>- Podcast<br>- E-mail<br>- Mobile Forums<br>- Discussion Platforms<br>- Podcasting |
| **Navigationism** | Learning is a process of connecting specialized nodes or information sources (Brown, 2005). | **Complex of information sources in mobile learning**<br>- Connecting specialized nodes<br>- Information sources<br>- Facilitate continual learning environment<br>- Knowledge management activities<br>- Decision-making<br>- Manage information (identify, analyse, organize, classify, assess, evaluate, etc.)<br>- Sense making and chaos management. | - Social networks (Blogs, Wikipedia, Twitter, Youtube)<br>- Podcast<br>- E-mail<br>- Mobile Forums<br>- Diccussion Platforms<br>- Podcasting |
| **Location-based Learning** | Location-based learning holds promise for just- in-time learning tied to a student's physical location (Johnson et | **Location context in mobile learning**<br>- Conceptual knowledge<br>- Conceptual application<br>- Constructive environment<br>- Partnership with location<br>- Immersive activities | - Field trips<br>- Archaeology studies<br>- Location based game<br>- Virtual world<br>- *Google Map, GPS, network triangulation* |

| | | | |
|---|---|---|---|
| | al., 2009) | | |
| **Learning & Teaching Support (Naismith et al., 2004; Oberer, 2016; Stanton & Ophoff, 2013)** | Administration and coordination of resources | **Learning and teaching support of mobile technologies**<br><br>- Administration of resources and learners<br>- Accessing course resources | - Taking attendance<br>- Reviewing students' marks,<br>- Accessing course materials<br>- Managing schedule including due dates of assignments<br>- *LMSs* |

The learning environment should be carefully considered as well. Learning environment is a location where learners can meet, work together and promote each other (Wilson, 1995). Learning environments include the learner, the space and the learning tools used by learners to gain information and knowledge from the entire learning environment (Wilson, 1995). Thanks to mobile technologies, the interaction among the learners, the teachers and the physical location has been reshaped (Al-Harrasi, 2015). Therefore, the developer creating a mobile app for educational purposes needs to take into consideration the learning environment that his or her app will be used. The learning environments and example cases to the way of use of mobile technologies in those are as followed:

Table 6. Integration ways of mobile devices into according to learning environments (McQuiggan et al., 2015)

| **Environments** | **Examples to the utilization ways of mobile technologies by learners** |
|---|---|
| Traditional, direct instruction | Note taking |

| | |
|---|---|
| Lecture | Utilizing PowerPoint presentations, using classroom response systems like *Poll Everywhere* to evaluate learners' understanding or ask learners' opinions |
| Blended and Flipped Classroom | Watching videos, making collaboration and communication with others |
| Problem-Based, Inquiry-Based, and Experiential Learning | Making research on internet, Collecting data inside and outside the classroom |
| Virtual and Homeschool | Creating projects and presenting them |

### 3.1.3. Conduct the System Requirement Analysis for the Mobile App

Implementing system requirement analysis for the mobile app is the next step the developer should perform. It includes identifying functional and non-functional requirements and use cases. Functional requirements are the essential matter for any product development. The terms of functional and non-functional requirements and use cases stem from requirements engineering. In this regard, requirement engineering is used in the software development cycle to define context for the software design (Nuseibeh & Easterbrook, 2000).

*Functional requirements* defines what the product has to perform or what processing move it should make (Robertson & Robertson, 2012). Functional requirements can be extracted from case studies (Inukollu et al., 2014), interviews with field experts (Dyli, 2016), content analysis (Ayobami et al., 2013) literature reviews and similar apps.

On the other hand, *non-functional requirements* are the characteristics which those functions should have (Robertson & Robertson,

2012). Non-functional requirements have a great significance as much as functional requirements and they are extremely crucial for success of a mobile app. Non-functional requirements for mobile applications include performance (efficient use of device resources, responsiveness, scalability), reliability (robustness, connectivity, stability), quality (usability, installability), and security (Wasserman, 2010).

Lastly, the developer should draw a *use case* diagram. The use case diagram is a simple way of displaying what the user should expect to be able to do in a system or application. It identifies the boundaries between the users (actors) and the product or application (Robertson & Robertson, 2012). The primary motive of designing this use case is to get a clear understanding of the system at first sight.

### 3.1.4. Identify the User Interface (UI) Design Requirements for the Mobile App

The User Interface (UI) design is a crucial task during the development process of a mobile application. The quality of the UI design can determine the success or not of a mobile application. Particularly, it influences not only the attractiveness of the mobile application but also, and perhaps more importantly, its usability. However, designing user interfaces is a challenging task since it varies according to the natural preferences and tendencies of the designer. Therefore, during years, UI design experts gathered and defined UI design patterns in order to ease the development of user interfaces. These design patterns are nothing more than cases of success that solve a certain recurring problem. Additionally, the mobile platform companies have also defined a set of UI guidelines with the goal of helping

and standardizing the development of user interfaces for their platforms. Thus, in this section, not only the principles but also patterns of UI design for mobile applications are provided.

#### 3.1.4.1.Principles for Mobile UI Design

It is extremely critical to provide the best user experience on mobile platforms in order to create an effective app. User interface design play a highly crucial role in creating optimum user experience. In this regard, Gong and Tarasewich (2004) proposed several guidelines for mobile device interface design. Weiss (2003) also provided several principles for mobile interface designs. The overall principles they stated are as followed:

**Design for multiple and dynamic context:** The mobile context is different from desktop computers. The mobility of handheld devices can lead to changes in the environment, e.g. brightness, locations, noise level. Therefore, different aspects such as type of input, font size or colors need to be considered.

**Design for small devices:** The screen size is a key difference between mobile devices and computers. Therefore, physical limitations need to be overcome by designing content, buttons and input in an appropriate size to make sure easy operation on small screens.

**Design for limited and split attention:** Users may focus on more than one task when using a mobile device. Their main focus may not lay on only the use of the mobile device. Mobile interfaces need to require as little as possible attention from its users. Visual attention, interaction, input and output need to simply design as much as possible.

**Design for speed and recovery:** Research has shown that users are less patient when it comes to mobile use than computer use. Mobile users are more demanding towards loading times. Therefore, time constraints should be taken into consideration.

**Design for top-down interaction:** Reduce information overload by providing hierarchy or multi-level mechanisms. This will prevent excessive scrolling and information load.

**Allow for personalization:** Mobile devices are more personal than desktop computers. A user should be able to personalize a mobile application regarding usage patterns, skills and preferences.

**Design for enjoyment:** Joy of use is a rather new quality attribute. Its main concerns are aesthetics and positive user emotions. Especially aesthetics have become an important acceptance factor for users. An appealing design can lead to positive user reaction.

**Consistency between platforms:** While overall consistency is a main usability attribute, further dimensions of consistency need to be considered for different mobile platforms and devices.

**Select vs. type:** Input which requires typing can become cumbersome on touch devices. The soft keyboard is less precise and often implies more work and time. Also, the keyboard can hide some content on the small screens. Therefore, where it is applicable, users should be presented with selection mechanisms, e.g. buttons, checkboxes or dropdowns, instead of keyboard input.

**Clickable graphics should look clickable:** Buttons and clickable icons

should always be recognized by the users. This can be done through appropriate styling, e.g. high contrast and typical button layout.

**Use icons to clarify concepts:** Icons are important design elements for mobile applications. They can enhance the aesthetic appeal and provide additional assistance for users.

#### 3.1.4.2.Mobile UI Design Patterns

Implementing UI guidelines or principles is very effective, but it is typically hard to correlate them to the software architecture (Folmer & Bosch, 2004). Therefore, in order to make software design easier, the usability engineering community has provided various design solutions such as usability patterns that can be applied to improve usability. A mobile app developer has to take into consideration UI design patterns in order to develop an app having an optimum UI design. Simply, a UI design pattern is a reusable solution to a common problem encountered every day. Bank (2014) defines it as "a formalized best practice, a guide or template, that designers, developers, and product managers (and anyone else who touches product) can use to solve common problems when designing a mobile application or system". The UI design characteristics and patterns of mobile devices are presented by Bank (2014), as followed:

##### 3.1.4.2.1. Gestures

Traditional gestures in computers are clicking, hovering and scrolling. However, mobile devices have new and different design patterns and their implementations. These gestures are touch, double touch, double

touch drag. Long press, swipe or drag, long press drag and pinch open and pinch close (see Figure 10).

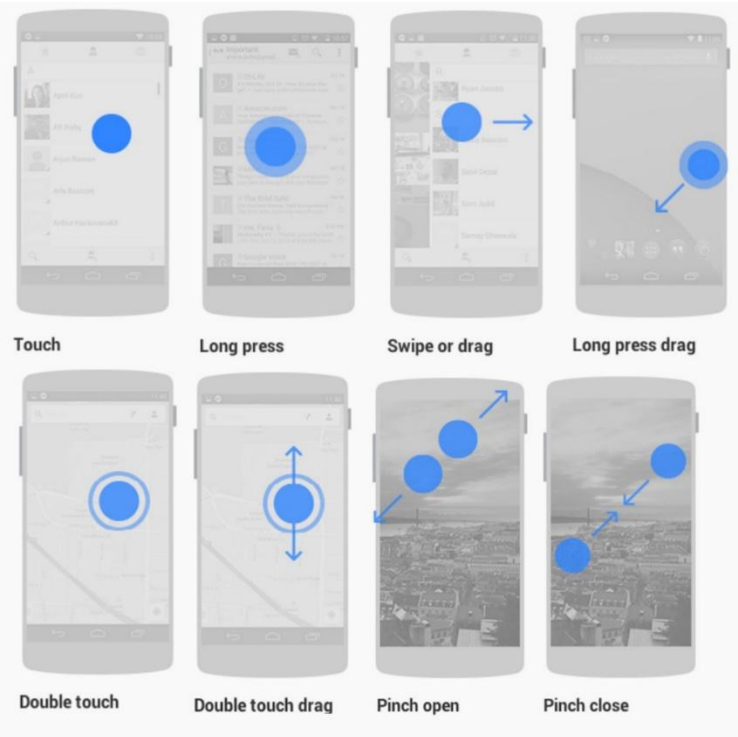



Figure 10. Gestures on mobile devices

#### 3.1.4.2.2. Getting Input

The patterns of getting input on mobile devices are as followed:

**Smart keyboards:** Provide the user the keyboard according to required data. For instance, when entering phone numbers in address books, the user does not need the full keyboard.

**Default values and autocomplete:** Make data entry comfortable for the

users by providing them default values. This can be combined with autocomplete functionality, to enhance the user experience by speeding things up.

**Immediate immersion or "lazy sign-ups"**: A number of apps enable their users to download and use the app before requesting them to register themselves. Registration mostly comes with an added benefit, like cross-device syncing. Late registrations might not always be a good idea, but the availability of trying app without registration can increase engagement with the app.

**Action bars:** Enable quick access for the important actions from the app's action bar (or "toolbar" in iOS terminology). While navigation bars have dominated web and early mobile application design, the use of other patterns like drawers, slideouts and sidebars, links to everything, button transformations, vertical and content-based navigation have allowed for more simple app views.

**Social login:** Integrate the app with social login solutions allowing users to login through their existing accounts, which implies they have one less username/password combination to worry about.

**Huge buttons:** The ideal size might be 72px, but some apps also provide huge buttons as well. Thus, the user knows exactly what to do and can do it quickly wherever he/she is.

**Swiping for actions:** Enable content to be swiped or moved out of the way. Thus, users will be able to handle the information on screen in an intuitive way.

**Notifications:** Stress recent activities by visually marking new content.

**Discoverable controls:** Users want quick access to controls that are secondary or only related to certain sections or content in the app. For this, clear up the mess and let users discover specific actions only when they need them.

**Expandable inputs:** Users want to focus on the content in the screen rather than sacrificing screen real estate to controls. For this, design controls in such a way that they expand when the users taps on them. Thus, this will keep most controls out of the way until the users need them.

**Undo:** Provide an easy way for users to undo their actions rather than just asking them to confirm deletions beforehand. Undo functionality hinder situations where an action can cause inconvenience or loss of data if done by accident.

#### 3.1.4.2.3. Navigation

Navigation is also one of the crucial elements of the UI since it shows users how to switch to different parts of the app. The navigation patterns are as followed:

**Walkthroughs and Coach Marks:** Users want to know how to use all features of the app, from core to subtle features. For this, design a walkthrough or a tutorial showing how each function of the app works. For example, some apps highlight important parts of the UI with “coach marks” to explain what they do, and the some use the first launch to demonstrate a slideshow that walks users through the entire experience, effectively explaining what can be accomplished with the app.

**Overflow menus:** Hide extra options and buttons in an overflow menu, so that the main interface will not be perceived as complicated.

**Sliders:** Make transitions among selections by swiping a finger.

**Content-Based navigation:** Make transitions between overview and detail states seamless. This will cause a fluid and intuitive user experience and flow.

**Morphing controls:** Change buttons and on-screen controls with their alternative functionalities, depending on which situation the user is, such as replacing the "+" into an "x" button.

**"Sticky" fixed navigation:** Users want to reach the menu anytime during using the app. For this, keep the top, side or bottom navigation' places fixed while a page is scrolled.

**Vertical navigation:** Users may need a way to navigate between different sections of the app, though there is limited space to show this information. For this, present important sections of the UI in a list, in which users can scroll through to get what they want.

**Popovers:** Users might want to view associated information without losing their current place in the UI. For this, utilize popovers.

**Slideouts, sidebars and drawers:** Users might need to navigate between different sections of the app without being impacted in each particular section. For this, use slideouts, sidebars or drawers, which appears in secondary section of the application as a collapsible panel and does not show up when it is not needed.

**Links to everything:** Users may want to interact with a part of content in the

app. For this, create links for everything, so that they can tap on it and go to a new view for a more detailed experience.

**Advanced scrollbars:** Users need to see their current position in the context of an entire content set. For instance, beyond scrolling with a swipe gesture, mobile lists and galleries have a persistent or temporarily scroll bar, and it can be a scroll index – dates, alphabetical letters, categories, etc.

**Swipe views:** Enable users to move from an item to another item by swiping through content without having to go back to the index, such as browsing through photo albums.

#### 3.1.4.2.4. Data and content management

It is very crucial to present the information properly and briefly. Thus, users do not encounter inconvenience to read it. The followings are best UI patterns to utilize when presenting content in a mobile app:

**Full-Screen modes:** Users might need to focus on content instead of being distracted by other UI elements. In this respect, hide or minimize the UI around content when it is needed. This will help users focus on what really matters for them.

**Inline expanding areas:** Make metadata invisible unless users explicitly want to see it.

**Transparency:** Utilize gradients and fading overlays to show that there is content layered below.

**Grids:** Present snippets of content in a grid for showing content in an organized form.

**Cards:** Enable browsing through content quickly and present pieces of information in bite-sized cards that can be customized to show more information if users want.

**Empty states:** Users need to know what section of the app is empty and what to do next with the app. For this, design the app for the "blank state" in which there is no any data entered by the user. Thus, UI will provide a good first impression to the users.

**Direct manipulation of content and data:** Enable users to edit content directly without having transition between editing or deleting modes.

**Draggable objects:** In order to sort and organize items in a way that makes sense to users in the current view, allow moving items around, including pressing-and-holding and dragging-and-dropping them.

**Pull to refresh:** Users also might want to be able to refresh the content manually. For this, provide a *refresh* button.

#### 3.1.4.3.UI Frameworks

Knowing and applying UI patterns and principles might be a challenge for the developer. One solution to this problem is using UI frameworks (or called as UI libraries) in app development. Frameworks is a set of reusable codes containing collections of functions, objects and templates. They assist developers to avoid painful start of developing applications from scratch (Ghatol, & Patel, 2012). The pre-written codes provided by UI Frameworks can be utilized in building new applications' user interface. Most frameworks have a standard way of development and consistency. For native apps, mobile platforms already offer UI frameworks

(or guidelines). While Android provides *Android Material Design* guidelines as a visual language for Android designers and developers to follow, IOS provides *Human Interface Guidelines* as for IOS designers and developers. In addition to them, *UWP Windows Apps Design* provides UI design fundamentals, responsive design techniques and detailed guidelines to create Windows apps[1]. On the other hand, for hybrid apps, there are various UI frameworks, but since they have different characteristics and the some have novelty, it is important to realize pros and cons of these UI Frameworks. Blanco (2016) has compared some popular UI Frameworks, as seen in Table 7.

Table 7. Comparison of some UI Frameworks (Blanco, 2016)

| **UI Framework** | **Advantages** | **Disadvantages** |
|---|---|---|
| jQuery | - Many components.& features<br>- Good support<br>- Good documentation | Not the best performance |
| Onsen UI | - Uses AngularJS<br>- Focused on performance<br>- Supports many OS<br>- Good documentation | Small community only in StackOverflow |
| Mobile Angular UI | - AngularJS used | Few information<br>Few support<br>Unclear platforms compatibility |

[1] https://docs.microsoft.com/en-us/windows/uwp/porting/android-ios-uwp-map#user-interface-ui

| | | |
|---|---|---|
| Chocolate Chip UI | - Native look & feel for each platform | Focused on iOS<br>Small community<br>Few components |
| Sencha Touch | - Based on MVC<br>- Focused on native look and performance<br>- Supports many platforms<br>- Many components | Not open source, payment framework |
| Framework 7 | - Native look & feel for both platforms<br>- Material design on Android<br>- Many components | Focused only on iOS and Android |
| Ionic | - AngularJS used<br>- Good looking components<br>- Clear documentation<br>- Large community and Ionic forum site | - |

#### 3.1.4.4.Usability Guidelines for Educational Apps

On the other hand, due to the rapid increase of the distribution of mobile applications within a short time, the importance of usability in developing an effective mobile application has become a crucial issue. In the same way, the studies on usability has increased due to the effect of usability on developing a successful app. Since the focus of this study is on the development of mobile applications for educational purposes, it presents usability guidelines especially related to mobile learning applications. While various researchers provide guidelines for mobile learning applications (Fetaji et al., 2011; Seong, 2006), Hujainah et al. (2016) presented the most comprehensive one including nine guidelines, as followed:

**Understand the level of users:** The users naturally refer to the learners for an educational mobile app. This principle consists of two main factors of an app. One is the level of the learner, skills, and background factor and the other one is the readability factor of the app. For example, learners' age and whether having special needs or physical disability can play an important role on the use of the app. Therefore, the developer should study on the learners who will use it before developing it.

**Avoid much content in one page:** Due to the small screen of the mobile devices, displaying and organizing the information on the mobile devices remains as challenge. So that, the developer should do followings: 1) Displaying only the relevant and important information and removing the others on the screen; 2) Dividing into subpages if the page includes long information in order to provide learner a clear view of information.

**Design convenient navigation system:** Navigation system have a crucial impact on developing an effective app. For this reason, the developer should pay special attention on navigation system. For this, the developer should provide learners selection option instead of input data, use similar navigation systems with other systems, and avoid using horizontal scrolling.

**Utilize the advantage of the feature provided by mobile devices:** The developer should use the available features of mobile device to enhance the performance of the app as well as learners' satisfaction.

**Consistency:** Consistency is one of the most indispensable factors in usability. The developer should be especially meticulous on consistency. For this, the developer should design the layout including labels, colors and

appearance across different mobile devices and operating services. Also, the functions of the app should have a start and end scenario and the users should be informed about that.

**Provide freedom to the users for controlling the app:** Giving users freedom on control the app enhances their satisfaction. This principle can be accomplished by prompting users to start the action instead of requesting them to reply to the action, by enabling enter and exit the app and by supporting *Undo* and *Redo* functions promoting their control ability on app.

**Preventing and handling error:** This principle related with the reliability of the app, which is one of the critical non-functional requirements of an app. Even if it is impossible to eliminate all errors, there are some ways of reducing the possibility of errors. These are preventing wrong or invalid inputs by using elements such as dropdown lists, spin buttons and calendar controls, preventing incomplete inputs by giving user a warning sign, preventing invalid actions via disabling unneeded buttons, preventing disastrous actions by providing a confirmation dialog, using accurate screen elements and following the usual flow of control.

**Design the app to be suitable with variety type of mobile screens:** With the development of mobile technologies, mobile devices has been varied day by day with respect to screen sizes. Therefore, the developer should develop the interface effectively regardless of different screen sized devices.

**Reduce the short-term memory load of users:** Due to the nature of usage mobile devices, it is important to reduce users' short-term memory load. It can be accomplished by designing screens with visible options or pull-down

menus and icons, reducing the response time of the app as much as possible, and making workflow obvious rather than expecting user to remember workflow.

### 3.1.5. Determine the target Operation Service(s) (Android, IOS, etc.)

The next step the developer should do is determining the target mobile Operation Services (OS or mobile platforms). The major OSs are Android, IOS and Windows Phone and Blackberry OS. Deciding which OS will be targeted is extremely crucial because it has an effect on which kind of mobile app will be developed and which development platform/tool will be used.

### 3.1.6. Determine the type of application: native, hybrid or web app

After that, the developer should decide which type of app he/she would develop. Mobile apps are mainly categorized as native, hybrid and web apps. This decision depends on the target OS. If the targeted OS is only one, the app is mostly developed as a native app, and if it is more than one, the app is developed as a web mobile or as a hybrid app.

**Native Apps:** The native apps are developed using the tools and programming languages provided for a specific mobile platform. These apps run only on the target platform. Also, they can be downloaded from the store (El-Kassas et al., 2015). The benefits of this kind of apps are providing best performance, having native look and feel of the user interface, and having

full access to device features such as GPS, file storage, camera, sensors, network access, database, SMS, and email (El-Kassas et al., 2015). However, they have drawbacks as well. One of the most critical disadvantages of native applications development is that the source code written for one mobile platform cannot be used on another (iiiiii, 2014). For example, an application developed for Android cannot used on iOS. Therefore, the native apps need to be developed separately for each platform. Also, it is more though to develop native apps than others and it needs a high level of expertise (El-Kassas et al., 2015).

**Web Mobile Apps:** The web apps are developed using the web technologies such as HTML, HTML5, JavaScript, and CSS. They do not require to be installed from the store and are accessed through a URL entered in the mobile web browser (El-Kassas et al., 2015; Phyo, 2014). The pros of this kind of apps are 1) easy to develop since they required web languages like HTLM, CSS and JavaScript; 2) easy update since the data are on the server, not on the device, 3) the same app developed once runs on different platforms without occurring any problem (El-Kassas et al., 2015). On other hand, they have various disadvantages. Since the data of these apps are stored on server, always Internet connection is needed to run them. Also, they are not available in app stores. More importantly, they have less performance and cannot reach and utilize the mobile device's software and hardware such as camera and GPS (El-Kassas et al., 2015; Phyo, 2014).

**Hybrid Apps:** The hybrid apps aggregates both native development and web technology, and it looks like a native app, has less capability than native app, but much easy to develop than native one. Therefore, this method saves

development and maintenance cost and time since it produces a single code base for multiple platforms and hardware (Phyo, 2014). They can be downloaded from app stores (El-Kassas et al., 2015). The following table compares the important sides of these three mobile app types and gives example apps created in that platform.

Table 8. Comparison of Mobile App Types (Dalmasso et al., 2013; Phyo, 2014)

| App type / Comparison Criteria | Native App | Web App | Hybrid App |
|---|---|---|---|
| **Development Language** | Native Only | Web Only | Native and Web or Web Only |
| **Device Access** | Full | Partial | Full |
| **Speed** | Very Fast | Fast | Native Speed |
| **Development Cost** | High | Low | Medium to low |
| **App Store** | Yes | No | Yes |
| **Advanced Graphic** | High | Moderate | Moderate |
| **Easy of updating** | Complex | Medium to complex | Simple |
| **Potential Users** | Limited to a particular mobile platform | Large- as it reaches to users of different platforms | Maximum including smartphones, tables and other feature phones |
| **Some Example Apps** | Angry Birds, Instagram | http://m.faceboook.com, http://m.bbc.com | Ebay, PayPal |

Overall, the native app development approach is the best choice for excellent performance, but requires high expertise on programming language and needs more time to develop. The web app development approach is not

costly, but limited in functionality, cannot offer exceptional user experience and mostly does not work without Internet connection. The hybrid app development approach is a middle way for many situations, especially if the developers want to create a single app that targets various platforms (Phyo, 2014). Nevertheless, it should be remembered that the potential app users (learners) in school environments have various OS-based mobile devices such as Android and IOS devices.

### 3.1.7. Determine the development environment or tool according to the type of app and your programming experience

As mentioned in the previous section, firstly, in order to develop a native app, platform-specific development tools should be used. These development tools are *Eclipse* requiring expertise on Java for Android or Blackberry OS, *Xcode* requiring proficiency in Objective-C language, Visual Studio needing C#/C++ for Windows Phone (Table 1). Secondly, web development tools requiring expertise on HTML, HTML5, JavaScript, and CSS can be used to develop a web mobile app. Lastly, there exists many ways for developing a hybrid app. A developer can create a hybrid app by using various kinds of programming languages and development tools, which are called as cross-platform tools.

**Cross-Platform Tools:** The challenge of developing a mobile application separately for each platform caused to exist alternative ways such as cross-platform solutions. The cross-platform solutions help the app developers in such a way that the developer writes an app's source code only one time and runs the produced application on different platforms without any problem

(El-Kassas et al., 2015). There are various kinds of cross-platform tools differing with respect to their target users. For example, for web designers, the cross-platform tools that require to proficiency in CSS, HTML, and JavaScript are provided. Some examples of such cross-platform tools are PhoneGap and Sencha Touch. Cross-platform tools have different approaches, technology, and programming language. For detailed information, it is highly recommended to read the article named as "Taxonomy of Cross-Platform Mobile Applications Development Approaches" written by El-Kassas and his colleagues (2015). They explain not all types of cross-platform tools, but also give detailed information on their approaches, architectures, technologies and advantages and disadvantages, as well as providing comparisons of those tools.

Table 9. Comparison of Some Class-Platform Tools (Litayem et al., 2015)

| Platform | Category of Mobile App | Type of Environment | Programming Language | Cross Platform Deployment |
|---|---|---|---|---|
| PhoneGap | Hybrid | Web view framework | HTML, CSS, JavaScript | iPhone, Android, Windows Phone, Blackberry, Symbion |
| JQTouch | Web, Native | Desktop, Browser environment | JQuery Plug-in, JavaScript, CSS | Desktop PC, Windows Mobile |
| Appecelator Titanium | Native | Rich APIs and low level TCP sockets | JavaScript | Android, iPhone, Blackberry |
| iWebKit | Web based | Desktop, Browser environment | HTML5, CSS3 | Desktop PC, iOS mobiles |
| Adobe Air | Hybrid | Rich Internet application platform | Action Script, HTML, CSS, JavaScript, MXML | iOS(iPhone, iPad, iPod Touch), Android, Blackberry |
| Sencha Touch | Hybrid, Web based | Access to subset of phone native API | HTML, CSS, JavaScript | iOS, Android, Blackberry, Windows Phone |
| iUI | Web based | Browser Environment for iOS | JavaScript, CSS | iOS based devices |
| QT | Hybrid | Deployable UI and applications | C++, QML | Symbian, Maemo |
| Mobile Voice agent | Hybrid | Client server app with proprietary speech engine | HTML, JavaScript, speex audio format | iPhone, Android |
| xUI | Web based | Browser Environment | JavaScript | iPhone, Blackberry |
| Rhodes | Native<br>Web | MVC support Real business logic | Ruby, HTML, JavaScript, CSS | iPhone, Android, Blackberry, Windows Phone, Symbian |
| CiUI | Web based | Browser Environment | CSS | iPhone |
| DragonRAD | Web, Native | Database driven drag and drop environment | Visual Drag & Drop Tiles | Android, Blackberry, Windows Mobile |
| aCME | Web based | Database driven Rich Internet App | Java | Desktop PC, Smartphone, tablets |
| QuickConnect Family | Native<br>Web | Database access | HTML, CSS, JavaScript | iPhone, Android, Blackberry, Web OS |

Nevertheless, giving short information about cross-platform tools would be useful for the developer who intends to develop an app running on different mobile platforms such as IOS and Android. In this regard, Table 9 provides a comparison of the cross-platform mobile app development tools with respect to the mobile app types, type of environment, programming language and platform deployment type, which they support. The developer can choose the most appropriate tool by considering the functional requirements of the app, his/her programming experience and the context of the app.

Table 10. A Comparison of the supported APIs by selected cross-platforms (Palmieri et al., 2012)

| **API Name** | **Rhodes** JavaScript | **PhoneGap** JavaScript | **MoSync** JavaScript | **MoSync** C, C++ | **DragonRad** |
|---|---|---|---|---|---|
| Accelerometer | | √ | √ | | |
| Barcode | √ | √ | | | √ |
| Bluetooth | √ | √ | | √ | |
| Calender | √ | √ | √ | √ | √ |
| Camera | √ | √ | | √ | |
| Capture | | √ | √ | √ | √ |
| Compass | | √ | √ | | |
| Connection | | √ | √ | √ | |
| Contacts | √ | √ | | | √ |
| Device | √ | √ | √ | √ | √ |
| File | √ | √ | √ | √ | |
| Geolocation | √ | √ | √ | √ | √ |
| Menu | √ | | | | √ |
| NFC | √ | √ | √ | √ | √ |
| Notification | √ | √ | √ | √ | |
| Screen Rotation | √ | √ | | √ | |
| Storage | √ | √ | √ | √ | √ |

Additionally, it should be reminded that some cross-platform tools may not support a number of device features like camera and GPS. Table 10 shows which features supported by some cross-platforms. According to this table, for example, MoSync cross-platform tool does not reach the device's camera feature, which means that the developer using this tool cannot develop an app having camera function. Therefore, it is highly recommended the developer to check to what extend the cross-platform he/she chose supports mobile device features.

**Tools not requiring coding:** On the other hand, for the people who have no knowledge and experience on programming, simple app development tools such as *App Inventor* and *App Pie* are provided. These tools include visual tools that allow users to develop their app without code. Usually, they are based on templates, and provide drag and drop features to generate the code. The biggest drawback of these tools is that they do not allow the developer to control the code and the design flexibility. Nevertheless, these tools provide a very crucial opportunity for non-developers as well as teachers who might not have programming experience at all.

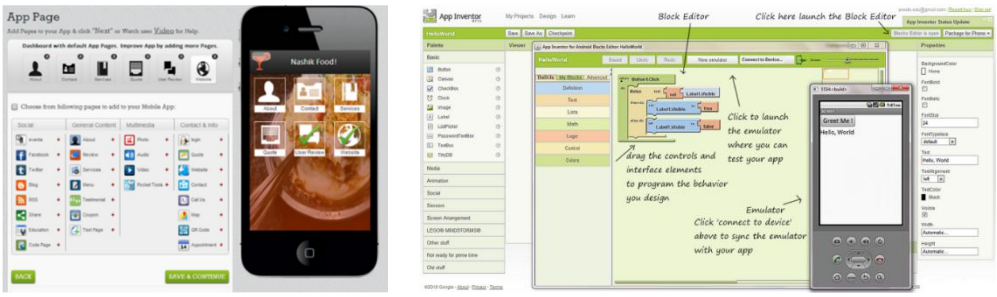



Figure 11. App Pie (on the left) and App Inventor (on the right)

App Inventor is an open-source web application originally created by Google and now owned by the Massachusetts Institute of Technology (MIT). It allows beginners with no programming knowledge to create Android apps by providing a graphical interface, very similar to Scratch and the StarLogo TNG user interface, which allows users to drag-and-drop visual objects and blocks to create an application (Gupta et al., 2016). The other tool is App Pie owned by Google. It also uses drag and drop tools and does not need coding skill or experience on app development (Gupta et al., 2016).

## 3.2.DESIGN

In the *Design* phase, the developer creates a prototype in accordance with the requirement analysis including functions, use cases, and UI design elicited in Analysis phase. This phase tries to answer the question of that what the user interface will look like (Berking at el., 2012). It contains the steps of concept design, prototype development, expert review, user testing and revision in an iterative loop. Thus, it is expected to develop a user-centered prototype.

### 3.2.1. Concept Design

The first step is to generate a concept design. Concept design is an indispensable step in any product development (Krishnan & Ulric, 2001). Likewise, it should be performed in mobile app development as well. In this step, the developer should make a decision on what the app concept will be and how the app will look like. This step is important because of mobile

devices' characteristics including small screen size, anytime and anywhere usage potential, and so on.

### 3.2.2. Prototype Development

The next step is the development of the prototype, a tangible and visual representation of a design concept that extracted from the previous step. Building a prototype is a crucial part of any product design and development process (Yang, 2005). Testing a design via prototyping can reduce the design risk without affording time and cost of the full production (Houde & Hill, 1997). By building the prototype of a design concept, potential problems about the design or its certain aspects can be handled. Besides, prototypes are a communication way of an idea to others (Kolodner & Wills, 1996). Through the prototype, the design concept is shared with others, and their opinions on the design concept are extracted. Warfel (2009) also states the advantages of prototyping as followed: 1) Prototyping is generative (of ideas), 2) Prototypes communicate through show and tell, 3) Prototyping reduces misinterpretation, 4) Prototyping saves time, effort, and money, and 5) Prototyping creates a rapid feedback loop, which ultimately reduces risk.

Arnowitz et al. (2010) distinguished prototyping into two categories with respect to the content fidelity of the prototype: low-fidelity and high-fidelity prototyping (Figure 12). Fidelity is the degree of detail that content is rendered in the interface, and more specifically, it is related to visual look, interaction behaviors, navigation flows and other sides of user experience as reflected by prototyping content (Arnowitz et al., 2010). Low-fidelity prototyping is sketches created early as the ideation of a design concept. It

enables developers to conceptualize page layouts like the rough position of menu content places, banners, toolbars and content (Arnowitz et al., 2010). Some low-fidelity prototyping methods are wireframes, storyboard, paper prototyping and paper mockup (Yamazaki, 2009). On the other hand, high fidelity prototyping has a more accurate look and feel of the final product. It is rich in detail with the products' all attributes. The aim of a high-fidelity prototype is to test the content with end users or to get their direct feedbacks Arnowitz et al. (2010). Digital prototyping, video prototyping, coded prototyping, detailed mock-up are some means of high-fidelity prototyping (Yamazaki, 2009).


Figure 12. Low-fidelity prototyping (on the left) and high-fidelity prototyping (on the right) (Arnowitz et al., 2010)

Prototyping is an inevitable part of mobile app development process as well. With the mobile prototyping, the developer gets feedback correctly about user experience on page flow, UI interaction and device interaction. Mendoza (2013) has introduced three types of making prototypes for mobile:

paper prototyping, desktop prototyping and using the mobile device itself to prototype.

#### 3.2.2.1.Paper Prototyping

Prototyping on paper can make it easier to gather quick feedback on the fly or to generate multiple ideas. The strengths of paper prototyping are 1) fast, cheap (basically free), and easy, 2) You can use it anywhere and anytime, no computer necessary and 3) It's one of the few tools that is suitable for collaborative design Warfel (2009). Mendoza (2013) has indicated two types of paper prototyping for mobile: Mobile Sheet and Notecard (Figure B). While Mobile Sheet prototyping uses sheets like A4 papers to get user feedbacks, Notecard allows for the most collaboration when laying out or editing a mobile user experience.

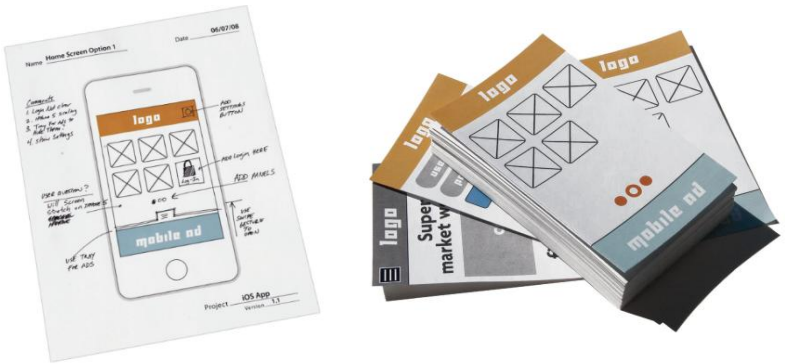



Figure 13. The Mobile Sheet (on the left) and the Notecard (on the right) paper prototyping (Mendoza, 2013)

#### 3.2.2.2.Desktop Prototyping

Desktop prototyping is the prototyping prepared with a software. Apart from well-known software including PowerPoint and Apple Keynote, specific programs such as OmniGraffle and Axure can be used for prototyping (Mendoza, 2013).

#### 3.2.2.3.Using the device to prototype

The last prototyping method for mobile is to use the device itself (Figure E). With this method, the developer (or designer) can get feedback about the correct size and proportion of elements on the screen and the correct colors and feel of the screen, that are very difficult feedbacks to extract from paper or desktop prototyping (Mendoza, 2013).

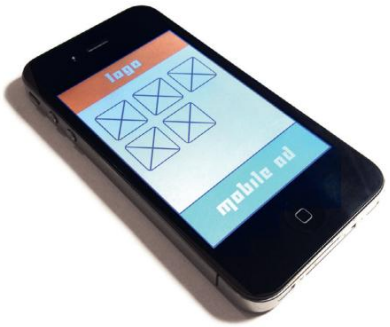



Figure 14. Prototyping with a device (Mendoza, 2013)

### 3.2.3. Expert Review

After the prototype is created, the next step of mobile app development is the validation and evaluation of the prototype, with respect to the app design. Prototypes are required to be reviewed and validated by

external stakeholders, such as domain experts and end-users. There is a variety of evaluation methods in software development in accordance with the stage of software development. Zhang (2001) has identified three types of usability evaluation method including usability inspecting, usability testing and

usability inquiry. Folmer and Bosch (2004) classified commonly used evaluation tools and techniques according to these evaluation methods (Table 11). They also differentiated these evaluation tools and techniques with respect to the software development stages in which they are used (Table 12).

Table 11. Evaluation techniques used in software development according to the type of evaluation method (Folmer & Bosch, 2004)

| **Type of Evaluation Method** | **Purpose/Usage Way** | **Participants** | **Evaluation techniques/tools** |
|---|---|---|---|
| **Usability Inspection** | To examine and evaluate whether each element of a user interface or prototype follows established usability principles. | - Usability specialists or software developers,<br>- Representative Users<br>- Other professionals | - Heuristic evaluation (Nielsen, 1994).<br>- Cognitive walkthrough (Wharton et al., 1994; Rowley et al., 1992).<br>- Feature inspection (Nielsen, 1994).<br>- Pluralistic walkthrough (Bias, 1994).<br>- Perspective-based inspection (Zhang et al., 1998).<br>- Standards inspection/guideline checklists (Wixon et al., 1994). |
| **Usability Testing** | To test the attributes of the final product even if it is not ready as a prototype, in order to see how UI supports the users to do their tasks. | - Representative users | - Coaching method (Nielsen, 1993).<br>- Co-discovery learning (Nielsen, 1993; Dumas and Redish, 1999; Rubin, 1994).<br>- Performance measurement (Nielsen, |

| | | | |
|---|---|---|---|
| | | | 1993; Soken et al., 1993).<br>- Question-asking protocol (Dumas and Redish, 1999).<br>- Remote testing (Hartson et al., 1996).<br>- Retrospective testing (Nielsen, 1993).<br>- Teaching method (Vora & Helander, 1995).<br>- Thinking aloud protocol (Nielsen, 1993). |
| **Usability Inquiry** | To acquire information about users likes, dislikes, needs and understanding of the system by talking to them, observing them using the system in real work (not for the purpose of usability testing) or letting them answer questions verbally or in written form. | - Usability evaluators (e.g., users) | - Field observation (Nielsen, 1993).<br>- Interviews/focus groups (Nielsen, 1993).<br>- Surveys (Alreck and Settle, 1994).<br>- Logging actual use (Nielsen, 1993).<br>- Proactive field study (Nielsen, 1993)<br>- Questionnaire (Zhang, 2001) |

As the design phase of mobile app development, in which prototype is continuously developed, validated, evaluated and improved, this study suggests developer to apply expert evaluation using heuristics since heuristic evaluation is an appropriate method for evaluating the design of the software (Folmer & Bosch, 2004). It also suggests combining this evaluation method with cognitive walkthrough method (Sears, 1997), which implies providing pre-defined tasks evaluators not only for exploring the features of the app but also being accustomed to its system when they perform the evaluation. The core purpose of expert evaluation utilizing heuristics and cognitive walkthrough is to examine and evaluate whether each element of the prototype follows established usability principles.

Table 12. Evaluation techniques/tools according to the software development stages (Folmer & Bosch, 2004)

| Evaluation method | Stages in software development cycle | | | | |
|---|---|---|---|---|---|
| | Requirement analysis | Design | Code | Test | Deployment |
| Proactive field study | ✓ | | | | |
| Pluralstic walktroughs | | ✓ | | | |
| Teaching method | | ✓ | ✓ | ✓ | |
| Shadowing method | | ✓ | ✓ | ✓ | |
| Co-discovery learning | | ✓ | ✓ | ✓ | |
| Question-asking protocol | | ✓ | ✓ | ✓ | |
| Scenario based checklists | | ✓ | ✓ | ✓ | ✓ |
| Heuristic evaluation | | ✓ | ✓ | ✓ | ✓ |
| Thinking-aloud protocol | | ✓ | ✓ | ✓ | ✓ |
| Cognitive walkthroughs | | ✓ | ✓ | ✓ | ✓ |
| Coaching method | | ✓ | ✓ | ✓ | ✓ |
| Performance measurement | | ✓ | ✓ | ✓ | ✓ |
| Interviews | | ✓ | ✓ | ✓ | ✓ |
| Retrospective testing | | ✓ | ✓ | ✓ | ✓ |
| Remote testing | | ✓ | ✓ | ✓ | ✓ |
| Feature inspection | | | ✓ | ✓ | ✓ |
| Focus groups | | | | ✓ | ✓ |
| Questionaires | | | | ✓ | ✓ |
| Field oberservation | | | | ✓ | ✓ |
| Logging actual size | | | | ✓ | ✓ |

Heuristic evaluation developed by Nielsen (1993) is a widely used evaluation method where experts systematically judge aspects of user interface design (Doubleday et al, 1997). It is easy to utilize, cheap and able to find many usability problems including both major and minor problems (Inostroza et al., 2013). At least three experts are suggested for the evaluation. If there are three experts carrying out the evaluation, 60% of the usability violations can be detected (Nielsen 1993), as seen in Figure 15.

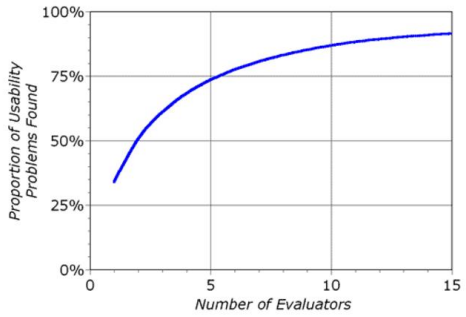



Figure 15. The proportion of usability violations found by using heuristics according to the number of evaluators used (Nielsen and Landauer, 1993)

It consists a set of heuristics to evaluate an interface according to recognized usability principles, such as users' language, consistency, minimizing memory load. He has also proved that errors are detected faster if evaluators are experts in the domain or in user interaction, maximally if in both (Nielson, 1995). In this evaluation method, evaluators decide on their own how to proceed in evaluating the interface with evaluation criteria proposed by Nielson (1993), but it is also recommended that they go through the interface at least twice. The first inspect is to be familiar with and get a feel for the flow of the interaction and the general scope of the system, and the second is to focus on specific interface aspects (Doubleday et al, 1997).

However, usability evaluations for the software have been affected by the emerging new technologies and have encountered new challenges such as evaluating touchscreen technology. Inostroza et al. (2013) have modified Nielson's heuristics by adding two new heuristics to adjust it for the touchscreen-based mobile devices (Table 13). In their study, they compared their new heuristics with Nielson's heuristics and found out that

the new one is able to detect errors much more than Nielson's heuristics. Accordingly, this study suggests developers use this *Touchscreen-based mobile devices heuristics*, as expert evaluation using heuristics.

Table 13. Items of Touchscreen-based mobile devices heuristics (Inostroza et al., 2013)

| **Item** | **Description** |
|---|---|
| Visibility of system status | The device should keep the user informed about all the processes and state changes through the use of a specific kind of feedback, in a reasonable time. |
| Match between system and the real world | The device should speak the users' language with words, phrases and concepts familiar to the user, instead of system-oriented concepts and/or technicalities. The device should follow the real world conventions and physical laws, displaying the information in a logical and natural order. |
| User control and freedom | The device should allow the user to undo and redo his actions, and it should provide "emergency exits" to leave the unwanted state. These options should be clearly pointed, preferably through a physical button or similar; the user should not be forced to pass through an extended dialogue. |
| Consistency and standards | The device should follow the established conventions, on condition that the user should be able to do things in a familiar, standard and consistent way. |
| Error prevention | The device should have a careful graphic user interface and physical user interface design, in order to prevent errors. The non-available functionalities should be hidden or disabled and the user should be able to get additional information about all available functionality. Users should be warned when errors are likely to occur. |
| Minimize the user's memory load | The device should minimize the user's memory load by making objects, actions, and options visible. The user should not have to |

| | |
|---|---|
| | remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate. |
| Customization and shortcuts | The device should provide basic configuration options and should give expert users access to advanced configuration options. The device should provide shortcuts to the most frequent tasks and should allow their customization and/or definition. |
| Efficiency of use and performance | The device should be able to load and display the required information in a reasonable time and minimize the required steps to perform a task. Animations and transitions should be displayed smoothly. |
| Aesthetic and minimalist design | The device should avoid displaying irrelevant or rarely needed information. Each extra information unit reduces the system performance. |
| Help users recognize, diagnose, and recover from errors | Error messages in the device should be expressed in plain language (no codes), precisely indicating the problem, and constructively suggesting a solution. |
| Help and documentation | The device should provide easy-to-find documentation and help, centered on the 664 664user's current task. A list of concrete (and not too large) steps to carry out should be provided. |
| Physical interaction and ergonomics | The device should provide physical buttons or similar user interface elements for main functionalities. Elements should be placed in a recognizable position. The device dimensions, shape, and user interface elements in general should fit the natural posture of the hand. |

On the other hand, Doubleday et al. (1997) have indicated that if various evaluation methods are used, evaluation leads to more successful design since it enables the design to be considered from a variety of perspectives and increases the probability of uncovering issues that may go undetected in one method.

Therefore, this study suggests developers use user testing too in the evaluation process of a prototype, as the second step in the design phase of development model for simple educational mobile applications, which is discussed in the next section.

### 3.2.4. User Testing

Usability testing or user testing, as described by Doubleday et al. (1997), is used to evaluate a product, with the aim of identifying any problems and measuring to what extent the users are satisfied with it. It is conducted to drive the development of the application forward and to validate the final version of the application. Its participants are actual or representative users. However, as seen in Table 11, there is a variety kind of user testing methods. Among those techniques and tools, this study suggests developers use *question-asking protocol* (Dumas and Redish, 1999). With this technique, during a user testing, the tester not only lets representative users verbalize their thoughts, feelings, and opinions when interacting with the system. Besides, the tester encourages representative users by asking direct questions about the product, in order to understand their mental model of the system and the tasks, and where they have trouble in understanding and using the system. The essential goal of user testing is to test the attributes of the prototype, in order to see and check to what extend UI supports the users to do their tasks.

The procedure of user testing utilizing question-asking protocol is as followed. First, the representative users are provided the prototype to be tested along with a set of tasks to perform. The tester asks them to perform

the tasks using the prototype, and explain what they are thinking about while working on the prototype. Additionally, the tester asks direct questions about the prototype with respect to UI preferences of the representative users for further enhancement of the application. Their feedbacks drawn from their prototype usage or their past experiences provide insights into their mental model of the prototype.

### 3.2.5. Revision

In revision step, the developer revises the prototype according to the feedbacks coming out of the expert review and user testing conducted in prior steps. Besides, the developer re-perform all these steps including concept design, prototype development, expert review, user testing and revision respectively until the concerned people are satisfied, as an agile process.

It has been proved that agile methodologies are one of the best means of mobile software development (Abrahamsson et al., 2003; Flora, & Chande, 2013). Flora et al. (2014) also have investigated the best practices of mobile application development process and conducting an extensive questionnaire with 130 participants including mobile companies, mobile development team members, mobile experts, researchers and related stakeholders. The survey has found out that 50 % of participants suggest using Agile Method for successful mobile app development, and they have concluded that agile methods are naturally fit for mobile app development (Flora et al., 2014). Even though there are various agile methodologies including Mobile D, RaPiD 7, Hybrid Methodology Design, MASAM and SLeSS, their common points are putting emphasis more on the human aspects of software

engineering than the perspective processes, thereby employing human interaction over tools and processes (Flora, & Chande, 2013). The core values of agile methodologies include individuals and their interactions, customer collaboration and responding to change (Flora, & Chande, 2013). Accordingly, in order to develop a successful mobile app for educational purposes too, the developer should apply an agile method, which means that he/she continuously revise the prototype reflecting UI design and functions of the designated app by conducting expert reviews and user testing in an iterative process until the stakeholders are satisfied.

Overall, evaluating prototypes via expert reviews and user testing enables developers to determine the degree to which requirements are being satisfied as well as the need for iterative improvement. Thus, a successful user-centered prototype will be generated.

## 3.3.DEVELOPMENT

### 3.3.1. Code the Mobile App in the chosen programming environment

In this step, the developer codes the app in the app development environment selected in *Analysis* phase, according to the final version of the prototype generated in the *Design* phase.

### 3.3.2. Expert Review

After that, the developer conducts an expert review again to check whether the prototype is well reflected in the app, and revises it according to

the feedbacks gathered from it. Since heuristic evaluation is a proper evaluation technique for the development phase of a product, as seen in Table 12, heuristic evaluation is used as expert review, same with the one conducted in the DESIGN phase.

This evaluation is considerable crucial because the expected app and the produced app may not be same due to the technical or human-sourced reasons like the expertise level of the developer, the nature of coding, the potential impracticability of reflecting some UI design or functions of the app in the chosen app development environment/tool. If this kind of situation occurs, the developer has to find another solution way to handle.

### 3.3.3. User Testing

User testing utilizing thinking-aloud protocol, which is an appropriate method for user testing (Folmer & Bosch, 2004), is performed as the next step. Whereas question-asking protocol is used for user testing in the *design phase,* thinking-aloud protocol is used as user testing in the *development* phase. The reason why question-asking protocol is not utilized that the design of app with respect to its concept is completed in the *design* phase. Therefore, thinking-aloud method is applied in order to disclose what users like and do not like and what their preference on the user interface is.

The procedure of this user testing is the same with the prior one, but its tasks are encompassed all detailed functions of the developed app. The representative user tests the app with those provided tasks. During this process, the participant is requested to think-aloud. Thus, it will be possible to deeply understand and interpret the actual app UI experience of users. This step is required because the expected UI experience and the real UI

experience might be different.

### 3.3.4. Make last revision

According to the results from the conducted those evaluations, the developer makes the last revisions on the app. With this step, the app production is finalized and thus prepared for the final evaluation, which is the issue of the next two phases.

## 3.4.IMPLEMENTATION & EVALUATION

### 3.4.1. Make learners try the Mobile App online (e.g., home) and/or offline (e.g., classroom)

The finalized app is tested via a pilot test in this phase. Several representative users use the app in a natural setting. Since the purpose is to develop an app for educational purposes, the representative users are learners and the setting is learning environment. Therefore, the app is tested by learners in the learning environment containing online (e.g., home) and/or offline (e.g., classroom) environment. However, if there is no such situation requiring using the app for learning purposes in the learning environment, then the learners are provided an artificial scenario by the tester.

### 3.4.2. Conduct field observation

During the app testing by the learners in the learning environment, field observation is conducted since field observation is an appropriate evaluation method of a software in the testing phase, as shown in Table 12. The learners are observed to understand how they are using the app to

accomplish the tasks and what kind of mental model the learners have about the system (Nielsen, 1993). The learners should be allowed to use the app not only in a classroom environment but also outside of school.

### 3.4.3. Conduct usability questionnaire

After learners tested the app, usability questionnaire is conducted with the learners who attended to field test, in order to measure the usability score of the app. Questionnaires have long been used to evaluate user interfaces (Root & Draper, 1983) and it is also an appropriate method as the evaluation method of the testing app (Folmer & Bosch, 2004). There is a variety of proposed questionnaire instruments in order to measure the usability level of a product, such as widely used SUS (System Usability Scale) (Brooke, 1996), Questionnaire for User Interface Satisfaction (Chin et al, 1988), Perceived Usefulness and Ease of Use (Davis, 1989), Purdue Usability Testing Questionnaire and USE Questionnaire (Lund, 2001). However, some studies state that traditional guidelines and methods used in usability testing of desktop applications might not be directly applicable to a mobile environment since mobile devices having unique challenges such as mobile context, connectivity, small screen size, different display resolutions, limited processing capability and power, and data entry methods (Zhang & Adipat, 2005). Therefore, recent studies have focused on developing new usability questionnaire (Hoehle & Venkatesh, 2015) as well as creating a generic framework for developing and conducting usability testings for mobile applications (Harrison et al., 2013; Zhang & Adipat, 2005). Harrison et al. (2013) proposed a usability model called PACMAD (People At the Centre of Mobile Application Development), including the following seven

attributes:

**Effectiveness:** the ability of a user to complete a task in a specified context.

**Efficiency:** the ability of the user to complete their task with speed and accuracy.

**Errors:** how well the user can complete the desired tasks without errors.

**Learnability**: the ease with which users can gain proficiency with an application.

**Memorability**: the ability of a user to retain how to use an application effectively.

**Cognitive Load:** analyzes the impact that using the mobile device will have on the user's performance.

**Satisfaction:** the perceived level of comfort and pleasantness afforded to the user through the use of the software.

Because mobile technologies have been evolving day by day, their characteristics have been changing as well, such as interaction with a multi touch screen, displays of different resolutions and dimensions, device orientation changes, and gestures like tap, flick, and pinch (Nayebi et al., 2012). Therefore, this study suggests developers utilize this PACMAD usability model when creating their usability questionnaire items that suitable for their apps and their technology. However, if there is a lack of time for creating own usability questionnaire, it is also possible to apply straight usability questionnaires such as SUS scale because they are also utilized for mobile apps. For instance, Kortum and Sorber (2015) has

successfully applied SUS scale in order to measure the usability of several mobile applications. If the SUS score of a system like mobile apps is above 68, then it is considered that the system is above the average of other systems. Figure 16 shows the acceptable range of SUS scores (Bangor et al., 2009).

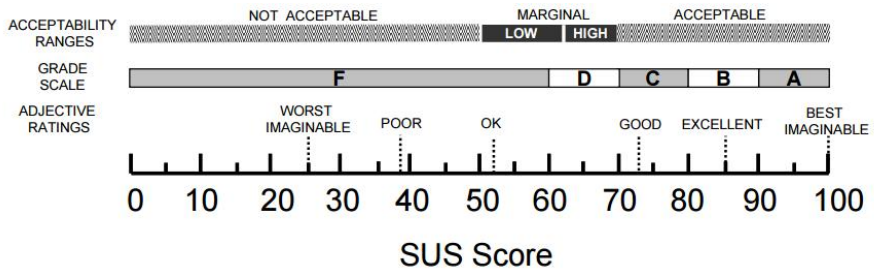



Figure 16. A comparison of mean System Usability Scale (SUS) scores by quartile, adjective ratings, and the acceptability of the overall SUS score (Bangor et al., 2009)

### 3.4.4. Conduct semi-structured interview with learners

After conducting field test and usability questionnaire, the next step is to make interviews with the learners who participated in the field test, in order to gather detailed information such as their likes and dislikes about the app. Interviews are an effective and appropriate evaluation method for evaluating the app in the testing phase (Folmer & Bosch, 2004). There are two methods of interviewing consisting of the unstructured and structured interview. While unstructured interviews do not have a well-defined agenda and are not concerned with any detailed aspects of the systems, structured interviews have a specific, predetermined agenda with specific questions (Nielson, 1993).

The interviewer can utilize the following guidelines when holding a

usability interview (Nielson, 1993). First, record the interview. Taking notes is a kind of distraction to the interviewee, who will have to restrain him/herself from looking the what is being written. Second, express the questions in an open or neutral way. In addition, encourage the user to reply with full sentences, rather than a simple "yes" or "no". Third, begin with less complicated issues and move to more complex subjects. Fourth, ask questions to disclose more information, not to approve the investigator's beliefs. Sixth, include instructions about the answer. For instance, answers can range from long descriptions to simple explanations, to identification or brief selection, to a simple "yes" or "no". Seventh, do not try to express to a subject why the system behaved in a particular way. Do not justify the design decision. Eighth, avoid using jargon and use terms that the interviewee can understand. Ninth, do not ask leading questions. A leading question implies that a situation exists and influences the direction of response. Tenth, do not agree or disagree with the user; remain neutral. Eleventh, use probes to acquire more information after the original question is answered. Probes are used to promote the subjects to continue speaking, or to guide their answer in a particular direction so a maximum amount of useful information is gathered.

However, this study suggests conducting semi-structured interviews with at least two learners participated in the field test, as suggested by Nielson (1993). Since the semi-structured interviews have some predetermined questions, the acquired information will be neither from a very broad view nor from very narrow perspective. Such questions might be “What parts of the system did you think were well designed?”, “Which parts of the system did you think were inadequately designed?”, and “Do you have

any other comments about the system functions and regarding its usability?", which are used by Georgsson and Staggers (2016) for the evaluation of a mobile system.

# 4. A Case Study: Evaluation Matrix App

## 4.1.ANALYSIS

### 4.1.1. Identify the need, aim and scope of the app

"Evaluation Matrix" app fundamentally needs the following functionalities: 1) The app should enable to create an evaluation matrix; 2) The app should enable to perform evaluation as alone and as a group; 3) The users should be enabled to access group evaluation via a unique code; and 4) The app should enable communication among the members when conducting a group evaluation. The aim of "Evaluation Matrix" app is to judge and find the most effective or strongest solution, choice, or idea among the many by comparing them according to specific criteria. The scope of "Evaluation Matrix" app is restricted with respect to possible platforms. It is limited to Android and IOS platforms. The users of this app are university students. It does not require any storing big-sized data, and the biggest data the app handle is images. However, it has to provide a real-time database for allowing real-time chat and allow saving dynamic data. Additionally, it will have native look & feel design to promote high usability. Its general features will include a login page, a page for creating a new evaluation matrix by

inputting its name, number of ideas and criteria, a page for performing evaluation, a page for group evaluation, a page for chat among group members, a page to check formerly completed evaluations, a page to edit account information and a help page.

### 4.1.2. Identify the appropriate learning theory and environment

Among learning theories, conversational learning and learning & teaching support will be implemented in this app. This app will be a supportive tool for learners in performing problem-based learning activities. When learners have various solutions or choices for the problem they cope with, this app will help them judge and find the best solution by comparing and rating all the possible solutions according to the specific criteria. “Evaluation Matrix” app will be developed especially for performing CPS (Creative Problem Solving) activities. Apart from that, conversational learning will also be promoted in this app by enabling chat function. With respect to the learning environment, this app will be enabled to use in problem-based learning environments which are blended or traditional learning environments.

### 4.1.3. Conduct the System Requirement Analysis for the Mobile App

The system requirement analysis consists of both of functional and non-functional analysis. Functional Analysis has been accomplished via literature review and prior two steps. Accordingly, the following functional requirements have been targeted for “Evaluation Matrix” app:

Table 14. Functional requirements of “Evaluation Matrix” app

| ID | Description |
|---|---|
| FR01 | The users must be able to sing up to the system by entering a username, email address, and a password. |
| FR02 | The users must be able to change their passwords. |
| FR03 | The users must be provided an alternative way for the case of forgetting their passwords. |
| FR04 | The users must be able to login via Facebook and Google accounts. |
| FR05 | The users must be welcomed with an "Intro" page that introduces the app when they sing up for the first time. |
| FR06 | The users must be able to create a new evaluation matrix by entering its name and number of criteria and idea. |
| FR07 | The users must be able to share the unique code belonging to the evaluation matrix for doing group evaluation. |
| FR08 | The users must be able to evaluate ideas, solutions or choices by selecting emoticons representing rating scores from 1 to 5. |
| FR09 | The users must be able to edit their ratings. |
| FR10 | The users must be able to access to a group evaluation via a unique code provided for each evaluation matrix. |
| FR11 | The users must be able to comment about group evaluation matrix. |
| FR12 | The users must be able to access their formerly completed evaluation matrixes. |
| FR13 | The users must be able to change their account information such as username. |
| FR14 | The users must be provided a help page for explaining the app |
| FR15 | The users must be able to log out from the system. |

On the other hand, "Evaluation Matrix" app has aimed the following non-functional requirements:

| ID | Name | Meaning | Aim |
|---|---|---|---|
| NR01 | Usability | The degree of ease to use software technology and having user-friendly interface (UI) | "Evaluation Matrix" app must have high usability. |
| NR02 | Portability | The degree to which software running on one platform can easily be transformed to run on | "Evaluation Matrix" app must work on both IOS and Android devices. |

| | | | |
|---|---|---|---|
| | | another platform | |
| NR03 | Security | Blocking unauthorized access to the system and its data | - "Evaluation Matrix" app must have login/password system<br>- The users must be only able to access to the group evaluation via a unique code. |

Table 15. Non-functional requirements for "Evaluation Matrix" app

Lastly, use case diagram for "Evaluation Matrix" app is as followed:

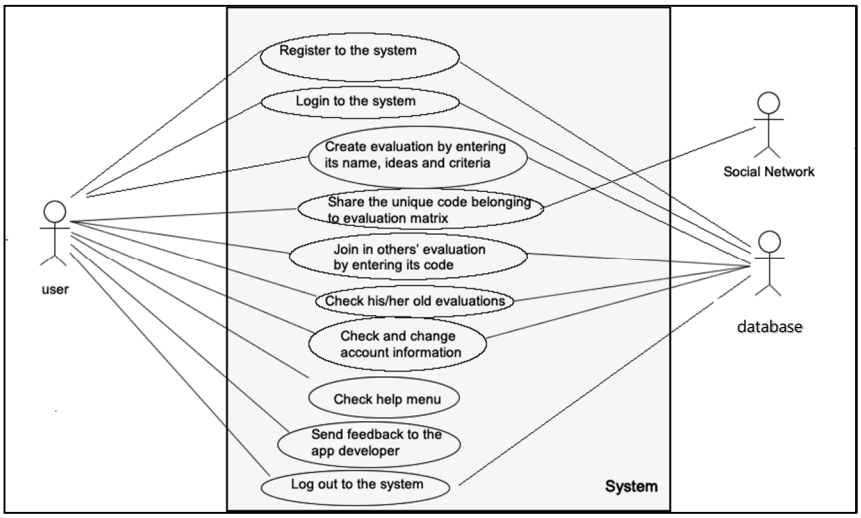



Figure 17. Use Case Diagram of "Evaluation Matrix" app

The initial requirements have been identified until now. However, these requirements might be modified later since the next steps consist of expert evaluation and user testing, which means there might be a suggestion of changing or adding some functions.

### 4.1.4. Determine the target Operation Service(s) (Android, IOS,

**etc.)**

It is highly possible that mobile device users in schools own different mobile devices from different platforms. For this reason, "Evaluation Matrix" app has targeted Android and IOS platforms, which are the most used two platforms.

### 4.1.5. Determine the type of application: native, hybrid or web app

"Evaluation Matrix" app will be developed as a hybrid app because of the availability of code sharing for different mobile platforms. Because "Evaluation Matrix" app targets not only Android but also IOS platforms, developing a hybrid app would be much logical since the concept of a hybrid app is that code one time, publish in various platforms, meaning it is more affordable with respect to time and effort.

### 4.1.6. Determine the development environment or tool according to the type of app and your programming experience

The researcher has six years experience on web technologies including HTML, CSS, and JavaScript. Therefore, he has looked for platforms that use HTML, CSS, and JavaScript to develop hybrid app working on IOS and Android. Among the available platforms seen in Table 9, he chose PhoneGap platform[1] in order to develop "Evaluation Matrix" app

[1] http://phonegap.com/

in a hybrid app form.

The reasons for choosing PhoneGap are as followed. First, PhoneGap allows hybrid mobile app development by using HTML5, CSS3, and JavaScript, and the researcher has expertise on them. Second, development of an app in PhoneGap is similar to the development of any site and so it offers greater ease of learning (de Andrade et al., 2015). Also, it supports Android, iOS, BlackBerry, Windows Phone, Symbian and Bada (Allen et al., 2010) platforms, meaning it is possible to develop an app for those platforms by using PhoneGap. Lastly, PhoneGap is the most preferred tool among cross-platforms tools with respect to the criteria of capability, performance, development speed, native UI, learning curve and device access (Appiah et al., 2015). PhoneGap is able to access most functions of a mobile device including accelerometer, camera, geolocation, media, compass, network, contacts, notifications, file and storage (Pierre et al., 2015). The following figure shows how PhoneGap platform works.

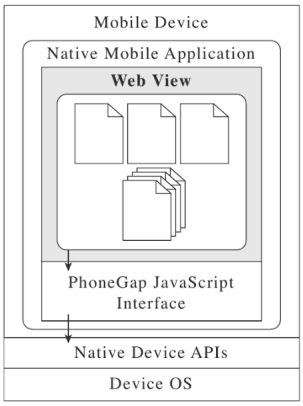



Figure 18. PhoneGap Structure (Wargo, 2012)

### 4.1.7. Identify the User Interface (UI) Design Requirements for the Mobile App

Since knowing and applying UI patterns and principles is challenge and time consuming, the researcher has decided to use a UI framework, consisting reusable codes containing collections of functions, objects, and templates. Among those available frameworks including JQuery, Onsen UI, Chocolate Chip UI, Sencha Touch and Framework7, the researcher has decided to utilize Framework7[1] for "Evaluation Matrix" app. The reasons for that is 1) Framework7 provides native look and feel UI for both Android and IOS; 2) It is compatible with PhoneGap; 3) It is free and open source mobile HTML framework to develop hybrid apps; 4) It has well-written documentation consisting of examples and live previews of the components; and 5) It has good tutorials and demo applications showing how to use it.

On the other hand, the researcher has decided to apply *Usability Guidelines for Educational Apps* (Hujainah et al., 2016) when developing "Evaluation Matrix" app in order to increase its usability. It consists of the following principles: 1) Understand the level of users; 2) Avoid much content in one page; 3) Design convenient navigation system; 4) Utilize the advantage of the feature provided by mobile devices; 5) Consistency; 6) Provide freedom to the users for controlling the app; 7) Preventing and handling error; 8) Design the app to be suitable with variety type of mobile screens; and 9) Reduce the short-term memory load of users.

[1] https://framework7.io/

## 4.2.DESIGN

The purpose of the *Design* phase is to develop a prototype consisting of functions and UI design by using Agile Method including an iterative process for user-centered design. For this, concept design, prototype development, expert review and revision steps have been performed respectively.

### 4.2.1. Concept Design

Concept design has been extracted from the literature review and other internet-based sources. Accordingly, there should be a page for creating an evaluation matrix by entering its name, ideas and criteria, there should be a page for giving scores to ideas according to criteria, the system should automatically calculate the evaluation result, and there should be a way for joining to a group evaluation. For the last one, the researcher has decided to use a unique code system. Accordingly, the app will assign a unique code for each evaluation matrix, and a user will be able to access to that evaluation by entering that unique code. Therefore, there should be also a share button for sharing that code via Social Networks such as WhatsApp and KakaoTalk.

In addition, it has been decided to use emoticons as rating scores for making evaluation more interesting. Lastly, while the horizontal axis is normally used for putting criteria and vertical axis used for putting ideas (as seen Figure 19), it has been decided that the places of these elements should be replaced for mobile devices, implying horizontal one is for putting ideas and vertical one is for putting criteria. The reason for that is the screen size of mobile devices is a portrait, meaning horizontal size smaller than the

vertical one, and the number of criteria is mostly higher than the number of ideas when performing an evaluation. Therefore, in order to adjust the screen to the data, it has been determined to replace the places of ideas and criteria.

Item 1. 설계대안 장단점 분석표

각 제한조건을 충족하는 정도에 따라 3(충족 정도가 높음) 2(충족 정도가 보통) 1(충족정도가 낮음)으로 평가하시오

| Item/Criteria | 평면에서 전후 이동 속도가 빨라야 한다. | 받침대를 기준으로 위치조정이 정확해야 한다. | 골프공을 안정적으로 잡을 수 있어야 한다. | 골프공을 안정적으로 운반 할 수 있어야 한다. | 골프공을 안정적으로 놓을 수 있어야한다. | 기능을 모두 수행하고 복귀하는 시간이 빨라야 한다. | 로봇의 질량이 최소화 되어야 한다. | 제작이 용이해야한다. | 제조원가가 최대한 적게 들어야한다. |
|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 설계대안 1 | 2.25 | 1.625 | 2.125 | 1.125 | 1.125 | 2.125 | 2.375 | 1 | 1 |
| 설계대안 2 | 2.125 | 1.625 | 1.5 | 1.75 | 1.5 | 1.875 | 1.375 | 1.75 | 1.5 |
| 설계대안 3 | 1.75 | 1.5 | 2.25 | 1.625 | 1.625 | 1.625 | 1.5 | 1.375 | 1.75 |
| 설계대안 4 | 1 | 1.375 | 1.75 | 2.375 | 2.25 | 1.125 | 1 | 1.25 | 1.25 |
| 설계대안 5 | 2 | 2.375 | 2 | 2.75 | 2.125 | 1.875 | 1.875 | 1.5 | 2 |
| 설계대안 6 | 2.5 | 2 | 2.75 | 2.25 | 2.625 | 2.625 | 2.5 | 2.5 | 2.25 |
| 설계대안 7 | 2.625 | 2.125 | 1.75 | 2.75 | 2.125 | 2.625 | 1.25 | 1.25 | 2 |
| 설계대안 8 | 2.5 | 1.75 | 2.75 | 2.625 | 2.875 | 2.75 | 2.25 | 2.25 | 2.375 |

Figure 19. Evaluation Matrix used in $S^3$CPS system (Lim et al., 2016)

### 4.2.2. Prototype Development

According to the functions and UI principles extracted in Analysis phase, the researcher has developed a prototype by using Microsoft PowerPoint 2016, as desktop prototyping. Hyperlink function of PowerPoint has been utilized for creating a clickable prototype. Since the testers were able to click on the prototype for page transitions, usability evaluations were more effective on exploring users' app usage. Besides, Adobe Fireworks CS6 has been utilized for creating and editing pictures. All pictures of the prototype can be found in Appendix 1. Main screens of the prototype of "Evaluation Matrix" app are as followed:

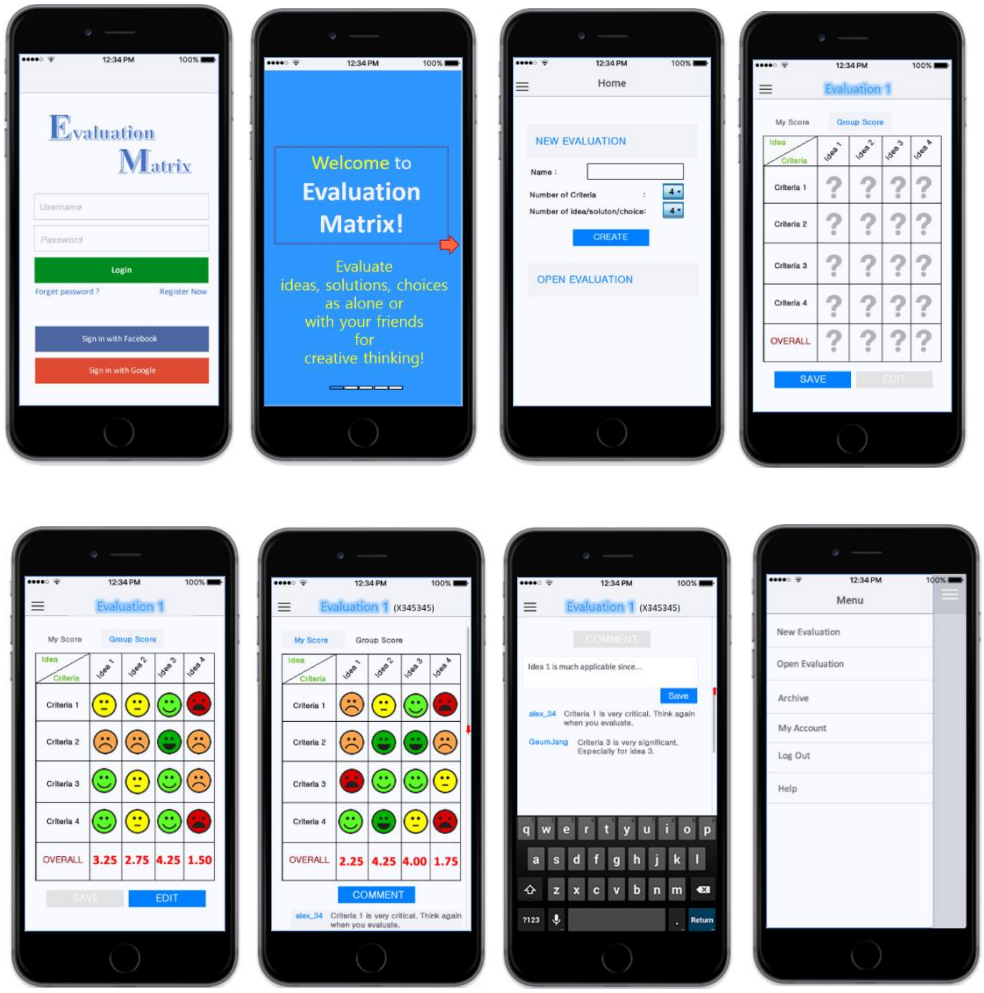



Figure 20. Main screens of the prototype of "Evaluation Matrix" app: Login Screen, Intro/Help Screen, Home Screen and Individual Evaluation Screen, Individual Evaluation Screen, Group Evaluation Screen, Comment Screen as a part of Group Evaluation Screen and Navigation Menu Screen (from left to right)

### 4.2.3. Expert Review

After a prototype for "Evaluation Matrix" app has been developed, expert reviews have been conducted with three experts by utilizing usability heuristics. The purpose of the expert reviews was to find out usability problems by examining and evaluating whether each element of the prototype follows established usability principles. The demographic information of the experts who participated in expert reviews has been followed:

Table 16. Demographic information of experts attending to prototype evaluation

| **Experts** | **Occupation** | **Expert Field** | **Experience** |
|---|---|---|---|
| Expert 4 | Consultant | E-learning Design<br>Creative Problem Solving | 2 years |
| Expert 5 | Web Developer | Web Development | 3 years |
| Expert 6 | Master's student in Educational Technologies | Design | 5 years |

Although making interviews with all experts would be a much better choice, the researcher has conducted an interview with only one expert, Expert 6. Because, the other two were workers and it was not possible to meet and make an interview with them. Therefore, the researcher has sent them the required files via for reviewing the prototype, containing a PPT file including the prototype of "Evaluation Matrix" app and a Word file including an evaluation instruction explaining how to conduct the evaluation of the

prototype via an email (Appendix 2). That Instruction has been consisted of the introduction of "Evaluation Matrix" app, a scenario and a set of tasks to perform while using the prototype, usability heuristics (Table 13), which are the evaluation criteria, a ranking scheme for experts to express to what extent the problem is serious, and a template for inputting usability problems, the criteria number violated, severity ranking and recommendations in solving those problems. Overall results of the expert review are as followed:

Table 17. Usability problems detected through Expert Review

| Issue ID | Issue/Problem Description | Heuristic violated | Severity | Recommendation | Expert Name |
|---|---|---|---|---|---|
| #1 | It seems that users can't edit or delete the records they enter into the app. | #3 | 4 | Edit and Delete button should be placed. Edit and Delete pages should be present either. | Expert 4 |
| #2 | It seems that the app never gives error feedback to users when they do something wrong, for example when they don't enter all the needed data. What's going to happen if a user omits one data input field and click the save button? | #10 | 3 | Provides the required error feedbacks in a popup form | Expert 4 |
| #3 | App introduction/help page that is set up as moving to other pages by clicking red arrow. However, it may cause a confusion for the users since it is not clearly indicated that this page is help screen.<br>➔ The user has to attempt to click many times on the screen and check the other pages to realize that this page is the help screen (which means confusion and loss of time for the user.) | #1 | 3 | The fact that the first screen is Intro/Help screen should be indicated more clearly. | Expert 5 |
| #4 | Evaluating multiple comparison objects via emoticons is not a general method in real-life.<br>➔ When evaluating/comparing various objects, rating method is | #2 | 3 | Other than emoticons, rating method should be added as well for evaluation. | Expert 5 |

| | | | | | |
|---|---|---|---|---|---|
| | (more) general. | | | | |
| #5 | It seems that calculating an "overall score" as the result of evaluation using emoticons is inappropriate. | #4 | 4 | Produce the total score when only using rating method for evaluation,. Do not calculate the total score when using emoticons for evaluation. | Expert 5 |
| #6 | After login, the help/intro page appears, and then it needs 5 times clicking on the arrow button to start to use the app (There is no skipping method.) | #8 | 3 | Enable the ability of immediately starting to use the app by providing a "Skip" button | Expert 5 |
| #7 | When an error happens during app use, there is no menu for reporting or coping with it. | #10 | 2 | Alongside with Help, provide a page/menu for sharing and handling Error or for Q&A | Expert 5 |
| #8 | The users may want to see the evaluation matrixes they performed formerly on the main screen of the app. | #8 | 3 | Provide a connection to reach formerly performed evaluation matrixes in the main screen. | Expert 6 |
| #9 | It seems that idea and evaluation names are automatically written as "Idea1, Idea2…" and "Criteria1, Criteri2..." by the app. Even if the users decide which idea and which criteria there will be before starting to use the app, it is difficult to remember them. | #6 | 4 | It should be enabled that the user can change and customize idea and criteria names. | Expert 6 |
| #10 | In the evaluation page, evaluating all ideas according to various criteria in one screen would be difficult since the screen has too many elements and may seem too complicated. | #9 | 3 | The users should evaluate the ideas one by one according to criteria on a simpler screen. | Expert 6 |
| #11 | The users can share the code of the evaluation matrix right after they created it. However, after they enter an evaluation matrix, there is no button for sharing the code although the code of the evaluation matrix appears in the top of the screen. | #7 | 2 | Provide a share button on the screen showing up after the users have created an evaluation matrix. | Expert 6 |
| #12 | Emoticon-based scoring screen shows up in the center of the screen. When the user gives a score for the elements placed in the center of the screen, the emoticon screen may block the user's sight and this may cause improper rating.<br>➔ All evaluation elements of the | #8 | 2 | The emoticon-based rating screen should be placed on the bottom of the screen. | Expert 6 |

| | | | | | |
|---|---|---|---|---|---|
| | screen should be seen properly. Even the emoticon screen appears all elements should be seen easily. | | | | |
| #13 | Emoticons are old-fashioned. The colors of the emoticon symbols are a little bit crude. | #2 | 3 | Emoticons that are more modernistic and professional should be applied. | Expert 6 |
| #14 | "Evaluation Matrix" text in login page is sans serif. It is difficult to read. | #2 | 2 | It should be converted to serif font for easily recognizing. | Expert 6 |
| #15 | The code of the evaluation matrix is too long. Even if the app can remember the code, and the user can share it via Social Networks, it is not a simple effort to do that because the user has to click on the share button, choose the Social Network service, and then find the ID of the person who she/he wants to share on the Social Network. | #8 | 3 | It should be enabled for users to remember the code. Thus, she/he can directly say the code to the person who she/he wants to share. For this, the code provided for each evaluation matrix should be shortened. | Expert 6 |

Since their expertise is on different fields, their evaluation results were also not the same. They have provided various feedbacks about "Evaluation Matrix" app and each of them were considerably precious. Expert 4 has experience in the field of Creative Problem Solving, and she has examined the prototype of "Evaluation Matrix" app to check especially to what extent this "Evaluation Matrix" app is properly designed as a tool of Creative Problem Solving. The feedbacks came from Expert 5 was particularly related to technical and logical issues. Lastly, Expert 6, an expert on design field, has provided feedback with regard to User Interface

### 4.2.4. User Testing

After expert reviews have been accomplished, user testings via interviews by using Question-asking Protocol have been conducted with the prototype of "Evaluation Matrix" app. Three master students have joined to

user testings. The researcher has conducted a face-to-face interview in the Korean language with each of them in a comfortable and silent place. Each interview also has been audio recorded, and the researcher has analyzed usability problems through listening to the records. The procedure applied in interviews was as followed: 1) A document including a consent form, the introduction of "Evaluation Matrix" app, the goal of the usability testing and a set of tasks which will be performed using a prototype (Appendix 3) has been provided to the interviewee; 2) The interviewee has read the document and has signed it; 3) The interviewee has completed the provided tasks by using the prototype of "Evaluation Matrix" app in a notebook; 4) The researcher has asked additional usability questions such as "if there is this kind of button on this screen, what would you think about it?" and "What about putting share function here instead of that page?" in order to find out users' UI preferences. The results of user testings are as followed:

Table 18. Usability problems detected through User Testings

| Issue ID | Issue/Problem Description | Heuristicviolated or other issues | Recommendation | User Name |
|---|---|---|---|---|
| #16 | In the login page, the font of "Evaluation Matrix" title is not easily readable. | #9 | Use sans font, not sans serif. | User 1 |
| #17 | It is not possible to start immediately to use the app. The user has to expose "Intro" menu first. However, the user may want to use the app directly. | #8 | Provide a "Skip" button for skipping the Intro menu | User 1 |
| #18 | In the page where the user creates an evaluation matrix, the terms of "New Evaluation" and "Open Evaluation" words are confusing. "New" and "Open" words sound similar and make misleading. | Wording | Instead of "new", use "create" term. ("~~New evaluation~~" → "Create evaluation") | User 1 |
| #19 | Realizing that there exists "comments" section for each | #8 | Instead of placing comments in the bottom of the screen, create a new | User 1, User 3 |

| | | | | |
|---|---|---|---|---|
| | evaluation matrix may be difficult for the user because the user has to do scroll down to see the comments. Also, it can be burdensome to reach "comments" part for the user since it requires additional efforts. | | "Comment" button near to "My Evaluation" and "Group Evaluation" buttons. | |
| #20 | To remember the code provided for each evaluation matrix to use by multiple users may be difficult for the users. | #6 | Make it shortened. | User 1 |
| #21 | Even if a code for each evaluation is provided for enabling group evaluation, it is still burdensome for the user. The reason is that the person knowing the code is still expected to run the app and write that code in the required area. This means additional efforts for the user. | #8 | Instead of sharing a code, send a link to the group members. Thus, they can directly open that shared matrix via the link and can perform the evaluation task in fewer steps. | User 1 |
| #22 | The readability of the "intro/help menu" seems low since the font size is quite big. | Font style | Use much smaller font for "Intro/Help" menu. | User 2 |
| #23 | Distinguishing that which emoticon icon corresponds to which score is difficult. | #2 | Show the value of each emoticon icon. | User 2 |
| #24 | In the evaluation page, "My Score" and "Group Score" buttons are difficult to differentiate. It is difficult to understand whether "My Score" page is active or "Group Score" page is active. | #9 | Clearly, indicate that which button is active or inactive. | User 2, User 3 |
| #25 | The code of evaluation matrix does not exist in the "My Score" page. | #4 | Provide the code in "My Score" page. | User 2 |
| #26 | The terms of "Open Evaluation" and "New Evaluation" seems confusing. | Wording | Instead of "open" word, use "join" word. {~~Open Evaluation~~ → Join Evaluation} | User 2 |
| #27 | Currently, the app automatically defines the name of ideas as "Idea 1, Idea 2, etc." and the name of criteria as "Criteria 1, Criteria 2, etc." However, it might be difficult to remember what Idea 1 represents, or the user might forget what it was. | #6 | Enable to write the name of ideas and criteria when creating an evaluation matrix. | User 2 |
| #28 | There is no any method for reporting an unexpected error encountered while using the app. | #10 | Create "Report Error" page for user to contact with the app developer in order to report any error. | User 2 |
| #29 | In the first page of "Intro" menu, the font style, color and size of are not easily readable and distinguishable. | Font style | Use smaller font size, and write the "Intro" menu content as a simple text. Instead of using yellow color | User 3 |

| | | | | |
|---|---|---|---|---|
| | | | as font color, use simple colors. | |
| #30 | Using orange color in "Intro" menu increases the number of the different colors used, and this creates a disordered and complicated screen. | Font style | Do not use orange color. Instead, use similar colors existing in the screen. | User 3 |
| #31 | In the "Intro" menu, it is difficult to differentiate the text of app explanation with the associated screen. | #9 | Decrease the opacity/transparency of the screen, and stress the text of app explanation. | User 3 |
| #32 | It is difficult to differentiate the colors of emoticon icons. They have similar colors, and therefore it is not easy to recognize the values of emoticon icons according to their colors. | #9 | Change the colors of emoticons to the extent that the users can easily differentiate. | User 3 |

The feedbacks extracted from users were mostly related to the User Interface such as font size and color. According to these results, although some user feedbacks have shown similarities with each other, most of them have been different. Besides, a few of them were the same with the feedbacks derived from expert reviews. Apart from identifying usability problems, they have also recommended some new additional features for "Evaluation Matrix" app (Table 19).

Table 19. Additional features recommended by representative users.

| Feature ID | Features Suggested | User who recommended it |
|---|---|---|
| 1 | In the case of having too many ideas and criteria, it may be difficult to realize easily which idea has the greatest evaluation score. Provide a ranking page in an apart page. | User 1 |
| 2 | It is not available to share the result of a performed evaluation matrix. Provide a share button for sharing the final score of an evaluation. | User 1, User 2 |
| 3 | In "Group Score" page, there is no any sign for showing how many people attended to the evaluation and how many of them have sent their | User 2 |

| | | |
|---|---|---|
| | evaluation scores.<br>Provide a sign for that. (Such as "4/7" meaning among seven people, the four sent their evaluation scores.) | |
| 4 | Which time and when the comment has been sent should be shown in chat. | User 2 |
| 5 | The latest comments should appear in the first order in chat | User 2 |
| 6 | Like button for comments should be presented. | User 2 |

Through the question-asking protocol, the research has tried to understand deeply what users like, what they prefer with respect to color, font style, etc., and which features they want to see on which screen. This method quite helped the researcher in investigating and revising the concept of the app. For instance, although the researcher put the chat feature below to evaluation matrix (Figure I6-comment screen) in the initial phase, he had some doubts about it. However, when he asked representative users whether putting chatting feature below to evaluation matrix or creating a new tab button near to "My Score" and "Group Score" tabs and making enable chatting in that tab. They have preferred the latter one. User 3 has explained why she has preferred chatting feature in a new tab button like this: "… *Scroll down is not good for mobile. Seeing all things in one screen without doing scroll down is much appropriate for mobile devices... Rather, create a new tab button near these tab buttons and put chatting function there*". Another thing the researcher has some doubts was the style of designated "Evaluation Matrix" app. There were two styles the researcher could implement in the app. One is utilizing the style of the existing Evaluation Matrix tool in CPS, in which practitioners are able to see all evaluation elements including all criteria and ideas at one glance, and are able to give ratings to all ideas on one screen. The other one is seeing only one idea on

the screen and giving rating it according to criteria, which means evaluating ideas one by one. Thus, the number of elements appearing on the screen would decrease, and this would help minimalize the complexity of the screen. This concept was also applicable since mobile devices have small screens. Even though the researcher has developed the prototype by implementing the former concept, he asked users about it again, as a part of the question-asking protocol. All three users have opposed the latter one. For example, user 1 stated "…I want to see all ideas and criteria at one glance, so that I can give rating them by comparing them". Thus, thanks to question-asking protocol, the researcher has clarified the ambiguous design issues related to User Interface.

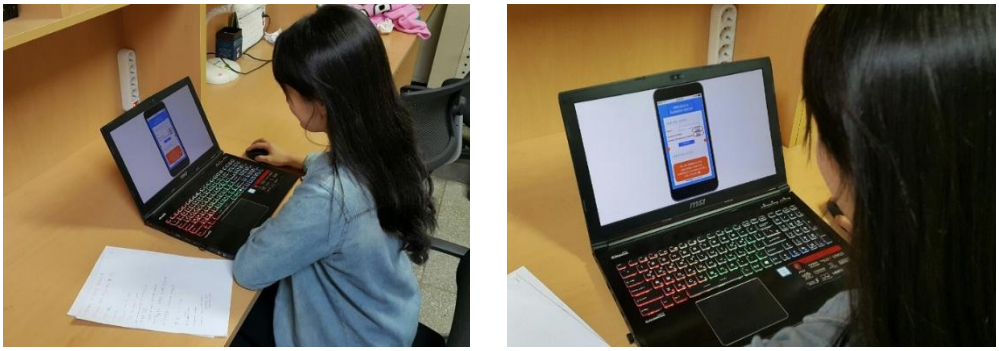


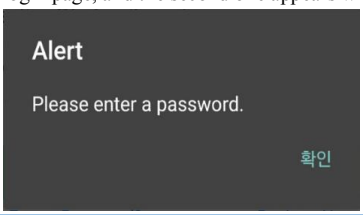
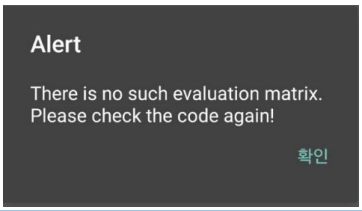
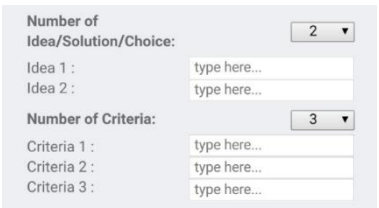
Figure 21. User performing tasks in user testing

### 4.2.5. Revision

After gathered data through expert reviews and user testings, the researcher has revised the prototype as much as possible. While some problems/issues have been fixed in the prototype of "Evaluation Matrix", some of them could not. The following table gives the detailed information about the revision situation of the usability issues/problems (Table 20). In

addition to this, because some usability feedbacks obtained from expert reviews were the same with the one extracted from user testings, these similar usability issues have been stated in the same cell in the table.

Table 20. The status of the usability issues derived from expert reviews and user testings

| Issue ID | Fixed or not? | If fixed, how? If not, why? |
|---|---|---|
| #1 | Fixed | - It has been enabled that the users are able to edit and delete the evaluation matrixes he/she developed.<br>- It has been enabled that the users are able to edit their account information.<br>- It has been enabled that the users able to edit their ratings before saving or sending it to the group. |
| #2 | Fixed | Alert pop-ups have been designed in such a way that a pop-up alert appears in the center of the screen when the user does not input a required data or when the user enters wrong information. Some example alerts are as followed. The first one appears when the user does not input a password in the login page, and the second one appears when the user does not enter the name of evaluation matrix:<br>  <br>  |
| #3 | Fixed | A button named “Skip Intro/Help” has been placed on the top of the screen in Intro/help page. The users will realize where they are. |
| #4, #5, #23 | Fixed | Instead of using solely emoticons, numbers (from 1 to 5) have been integrated with emoticons. Also, it has been designed in such a way that only the numerical scores will appear in the matrix. The users will see emoticons only when they give a rating. |
| #6, #17 | Fixed | A button named “Skip Intro/Help” has been placed in the top of the screen in Intro/help page. Thus, the users can skip the intro by clicking this button. |
| #7, #28 | Fixed | A new page called “Feedback” has been created to report errors occurred while using the app. The users can reach this page via the navigation menu. |
| #8 | Fixed | A new section showing the names of formerly completed evaluations has been created on the main screen. It has been designed in such a way that when a user clicks one name, the result of that named evaluation will appear. |
| #9, #27 | Fixed | It has been designed in such a way that when a user enters the number of ideas or criteria, textboxes for entering the content of ideas or criteria will appear in right under of the associated place, as seen in the following picture:<br>  |

| | | |
|---|---|---|
| #10 | Not Fixed | This suggestion was rejected by the representative users during user testings. Therefore, it has not been implemented. |
| #11 | Fixed | A share button has been placed in the top of the screen |
| #12 | Fixed | Emoticon-based rating has been placed in the bottom of the page. |
| #13, #32 | Fixed | The emoticons used has been changed with the more stylish ones. |
| #14, #16 | Fixed | Instead of putting "Evaluation matrix" text in text format, the app logo has been placed there. |
| #15, #20 | Fixed | The unique code provided for each evaluation has been shortened as a four-digit number. |
| #18 | Fixed | "New evaluation" term has been modified as "Create evaluation" term |
| #19 | Fixed | A new tab called "Comments" has been created next to "My Score" and "Group Score" buttons. |
| #21 | Not Fixed | Making this possible as mentioned in the recommendation suggested way exceeds the coding ability of the researcher |
| #22 | Fixed | Smaller fonts have been used in "Intro/Help" page. |
| #24 | Fixed | Tab buttons have been used. Thus, the users will be able to easily differentiate which button is active or inactive. |
| #25 | Fixed | The unique code belonging to a matrix app will appear in the top of the screen when a user creates a new evaluation matrix. |
| #26 | Fixed | "Open Evaluation" term has been changed as "Join in Evaluation". |
| #28 | Fixed | Font size has been decreased. The content of "Intro/Help" menu has been simplified with respect to font style and text amount |
| #30 | Fixed | Similar colors used in the app have been applied in the "Intro/Help" page. |
| #31 | Fixed | App screenshots have been placed to differentiate the text with the associated screen in "Intro/Help" menu. |

Some usability issues were could not solved or the researcher has not revised as recommended. The issue #10, mentioned by the design expert, is about changing the main concept of the app. However, her suggestion has not been accepted by the users during conducting user testings. Therefore, the researcher has not implemented it in the app. On the other hand, the issue #21 also could not be applied in the app. The reason for that is that the researcher, who is the developer, does not know at all how to do that. Realizing it requires very high experience in app development. On the other hand, the status of newly suggested features are shown in the following table:

Table 21. The status of the newly suggested features during user testings

| Feature ID | Added or not? | If added, how? If not, why? |
|---|---|---|
| #1 | Not added | Lack of time / Not a compulsory feature / It seems that the app can be used sufficiently without it. |
| #2 | Added | A button named "Share the results" has been inserted to "Group Score" page, in order to share the results in the form of screenshot via Social Media. |
| #3 | Added | A new section has been created in "Group Score" page for showing the names of the people who have sent their evaluation results. |
| #4 | Not added | Lack of time / Not a compulsory function / It seems that the app can be used sufficiently without it. |
| #5 | Not added | Lack of time |
| #6 | Not Added | Lack of time / Not a compulsory feature |

Among the new features suggested by the representative users during user testings, the researcher has only added the features he thought important because of the limited time. Therefore, he has added; 1) A share button in "Group Score" page in order to share the overall evaluation results in screenshot form via Social Media; 2) A section has been created in "Group Score" page in order to show the names of people who sent their evaluation scores to group scores. The other features suggested by the users have not been added by the researcher.

After the prototype revision has been completed, the researcher did not re-perform the steps of the design phase, as a part of the iterative process due to the fact that he has felt it was unnecessary for "Evaluation Matrix"

app since it is a simple app. Therefore, he has continued with *Development* phase.

## 4.3.DEVELOPMENT

### 4.3.1. Code the Mobile App in the chosen programming environment

As mentioned before, PhoneGap as app development environment and Framework7 as UI Framework have been selected. Because this was the first time for the researcher to develop a mobile application and he has had no knowledge of how to use PhoneGap and Framework7, he has had to learn about them. While coding "Evaluation Matrix" app, it was quite easy to learn and apply Framework7 whereby its official website. However, it was not the same with PhoneGap. Learning how to utilize PhoneGap through only its official website was to some degree difficult due to the fact that PhoneGap has a variety of products including PhoneGap Desktop App, PhoneGap CLI, PhoneGap Developer Mobile App and PhoneGap Build and each of them have different usage area. Therefore, the researcher has studied on it through its official website, blogs, YouTube, and eBooks including *PhoneGap 3.x Mobile Application Development Hotshot* (Shotts, 2014) and *PhoneGap Mobile Application Development Cookbook* (Gifford, 2012), which were available on SNU Library website[1]. Besides, he has examined and tried out sample PhoneGap apps, available on the internet. These tutorials show and explain all development process of a PhoneGap app, from creating it to

[1] http://library.snu.ac.kr/

packaging it as IOS and Android app. Thus, the researcher has learned PhoneGap by examining and trying out those samples. Some examined samples and their links are as shown in Table 22. He had practiced by using those sources before starting to code "Evaluation Matrix' app.

Table 22. Some tutorials studied by the researcher to learn PhoneGap

| Tutorial Name | What it teaches | Link |
|---|---|---|
| Framework7 & PhoneGap – Getting Started | How to use Framework7 and PhoneGap together in a mobile app | http://thejackalofjavascript.com/framework7-phonegap-getting-started/ |
| PhoneGap Quick Start | How to use PhoneGap Build | http://thejackalofjavascript.com/phonegap-quick-start/ |
| A Complete PhoneGap Tutorial (Beginners to Advanced) | A Complete PhoneGap Tutorial from beginner level to advanced level | https://codesundar.com/phonegap-tutorial/ |
| Apache Cordova Tutorial | Basics of PhoneGap app development including how to use PhoneGap CLI | http://ccoenraets.github.io/cordova-tutorial/ |
| Introducing the PhoneGap Developer App | How to use PhoneGap Developer Mobile App | http://devgirl.org/2014/04/22/introducing-the-phonegap-developer-app/ |

After that, it has been initiated to code "Evaluation Matrix" app. The followings have been performed for this. First, the prerequisite programs for PhoneGap CLI have been installed, which are Node.js[1], which is an open-source, cross-platform JavaScript runtime environment for running JavaScript code server-side, and Git[2], which is a version control system

[1] https://nodejs.org

[2] https://git-scm.com/

(VCS) for tracking modifications in computer files. Then, PhoneGap CLI, a command line interface for creating PhoneGap apps, has been installed and launched. Overall, the following commands have been used in PhoneGap CLI:

Table 23. Main commands used in PhoneGap CLI

| Code | Aim |
|---|---|
| `phonegap create EvaluationMatrix --template Framework7` | Create a Framework7 based PhoneGap app, named as "EvaluationMatrix" |
| `phonegap platform add ios`<br>`phonegap platform add android` | Enable app to work in IOS and Android platforms. |
| `phonegap plugin add cordova-plugin-splashscreen` | Enable to use splash screen feature |
| `phonegap plugin add cordova-plugin-x-socialsharing` | Enable to use social sharing feature |
| `phonegap plugin add https://github.com/gitawego/cordova-screenshot.git` | Enable to use screenshot feature |
| `phonegap serve` | Establish a server for the app |

The next step after the Framework7-based PhoneGap app has been created was to code "Evaluation Matrix" app, which was just like a coding a web page. For this, Brackets[1] as code editor has been utilized. Only HTML, JavaScript and CSS have been used. When the researcher encountered any errors or when he had no idea to implement something during coding, he looked for the solution on the internet, especially https://stackoverflow.com/ and https://www.w3schools.com/.

Meanwhile, the researcher has realized that the database structure of

[1] http://brackets.io/

mobile apps is not the same with web pages' because mobile devices had local storage as well. The real problem was that the users of intended "Evaluation Matrix" app had to be connected with each other and internet, data exchange among all devices having "Evaluation Matrix" app should be enabled and the users should be enabled to chat with each other, meaning real-time chat function. Therefore, the researcher had looked for a solution on the internet; two blog pages had enlightened this issue. One explains the local storage options for PhoneGap apps[1], and the other one provides very clear and brief information related to backend solutions for the database[2]. Thus, the researcher has decided to use *Firebase*[3] as the database solution of "Evaluation Matrix" app. Firebase is a backend service owned by Google and it permits to develop mobile and web applications with no server-side programming. It provides various services such as *Analytics, Cloud Messaging, Authentication, Realtime Database, Storage, Hosting and Performance Monitoring*. Among them, the researcher has benefited from authentication, real-time database, and storage services. However, learning and utilizing Firebase took quite long time since it was the first time for the researcher to use it and since Firebase employs NoSQL data structure, which is a very new technology the researcher does not know. Though its official website explains briefly how to use it, it was not enough, and so the researcher has looked for other sources such as are tutorials and templates. In

[1] https://www.joshmorony.com/a-summary-of-local-storage-options-for-phonegap-applications/

[2] https://www.joshmorony.com/a-summary-of-backend-options-for-html5-mobile-applications/

[3] https://firebase.google.com/

order to utilize effectively Firebase's authentication feature, a template[1] has been employed and a tutorial[2] has helped for using Firebase's real-time database feature.

[1] https://github.com/firebase/quickstart-js

[2] https://codelabs.developers.google.com/codelabs/firebase-web/#0

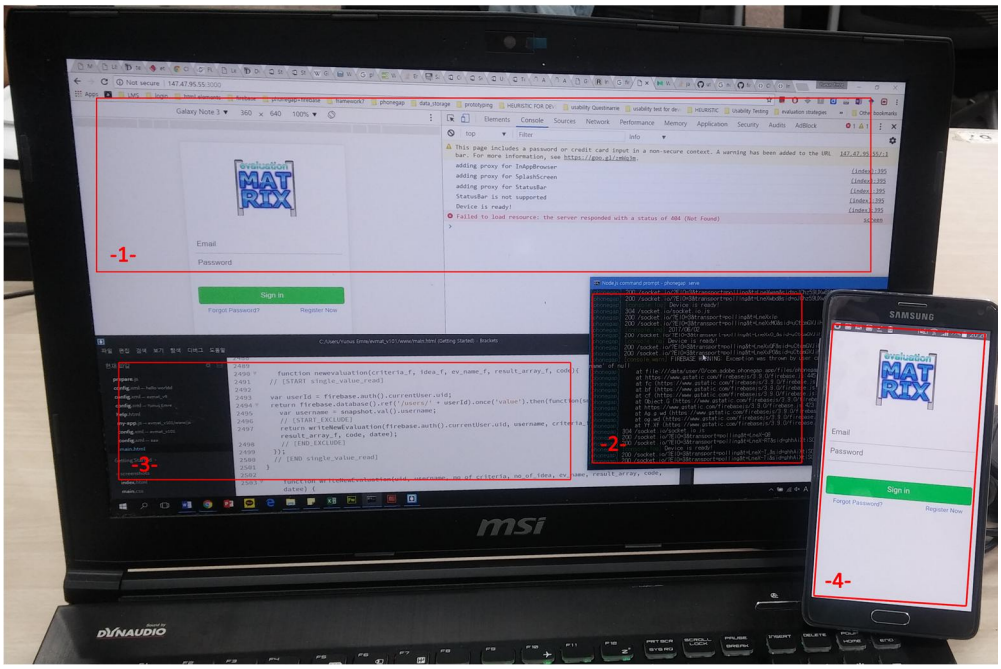



Figure 22. The development environment the researcher used (1- Google Chrome Developer Tools, 2- PhoneGap CLI, 3- Brackets and 4- PhoneGap Developer App)

Google Chrome Developer Tools[1] has been employed in order to debug the code while developing "Evaluation Matrix" app. It consists of a set of web authoring and debugging tools built into Google Chrome. It has been used to catch errors and to analyze the content of a web application. Besides, PhoneGap Developer App, which runs on a mobile device has been utilized. It has been used to preview the building "Evaluation Matrix" app quickly on a mobile device. For this, writing the IP address of "Evaluation Matrix" app being served from the PhoneGap CLI onto PhoneGap

[1] https://developer.chrome.com/devtools

Developer app was enough. Lastly, after the all coding has been completed, PhoneGap Build, which is PhoneGap Build is a cloud service for compiling PhoneGap applications, has been employed in order to produce “Evaluation Matrix” app as Android and IOS app.

However, when the researcher has uploaded all HTML5, CSS, and JavaScript assets of “Evaluation Matrix” app, PhoneGap Build has built it only as an Android app. This is because the researcher has no Apple Developer key, which costs 99$ yearly. Therefore, the researcher decided to go with the only Android app. Overall, the used programs to develop “Evaluation Matrix” app are summarized in Table 24.

Table 24: Development Environment Used for developing “Evaluation Matrix” App

| Tools Used | Usage Purpose |
|---|---|
| **PhoneGap CLI** | To create PhoneGap app |
| **Brackets** | As code editor for HTML, CSS, and JavaScript |
| **Firebase** | As the database platform |
| **PhoneGap Developer App** | To check how ”Evaluation Matrix” app works in an actual mobile device |
| **Google Chrome Developer Tools** | To debug (run and test) “Evaluation Matrix” app |
| **PhoneGap Build** | To build “Evaluation Matrix” app as IOS and Android app |

Lastly, the researcher could not accomplish perfectly although most

of the features extracted from requirement analysis have been successfully implemented during this step. There are two points the researcher has failed. First, FR04 (from Table 13) could not be implemented during coding the app. Even though Firebase and PhoneGap platforms support social login function, it is not possible this function to work without error when these two platforms are used together. This problem occurred not the because of the developer, but because of the compatibility of these platforms. The second was NR02 (from table 15). The initial intention was to develop "Evaluation Matrix" app for two main platforms, IOS and Android due to the heterogeneity of the school environment. However, it has been failed because of the issue mentioned previously.

### 4.3.2. Expert Review

After "Evaluation Matrix" app was developed, expert reviews and user testings have been conducted in order to increase the usability of "Evaluation Matrix" by eliminating usability errors and revising it. However, before starting expert reviews, the researcher has conducted a pilot test with one university student who is an Android user. The purpose of this pilot test was to discover obvious usability problems and overcome them in order to conduct healthier usability evaluations including expert reviews and user testings. The detected usability errors and their status, whether they were handled or not, are as followed:

Table 25. Usability problems detected through Pilot Test

| Issue ID | Description | Heuristics violated | Fixed or not? | If fixed, how? If not, why? |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| | | or other issues | | |
| #1 | When clicking “Back” button, it does not work as expected. The app returns “Intro/Help” page no matter where I am in the app if clicking “Back” button. | Coding Error | Not | Could not find the cause of the error in the code. |
| #2 | In comments page, “send image” button is not working. | Coding Error | Partially | Whereas “send image” button does not work in one click, it works completely fine in consecutive four clicks. The cause of this error could not find in the code. |
| #3 | It seems that it is not possible to edit my account information. | #3 | Fixed | The required code has been added to app |
| #4 | The navigation menu opening from the left is not automatically closing itself when clicking buttons in the navigation menu. | #4 | Fixed | The required code has been added to app |
| #5 | When sending a report related to errors, I click the send button and the app says it has been sent. But, text area where I input error content does not reset itself. This makes me feel like that the error message could not send. | #1 | Fixed | The required code has been added to app |
| #6 | When clicking navigation menu button, which is on the top right of the screen, the navigation menu has been opened on the left side of the screen. This was out of my expectation. The navigation menu and its button should be on the same side of the screen. | #4 | Fixed | The navigation menu button has been placed on the top left side of the screen. |

Although the researcher is an expert on the field web technologies including HTML, CSS, JavaScript, he had no experience of mobile app development and this was the first time he developed a mobile app. Therefore, he has studied and learned about it while conducting this research. Therefore, the researcher could solve some coding errors including issue #1 and #2 whereas most usability errors or issues have been overcome.

After “Evaluation Matrix” app has been revised according to the outcomes of the pilot test, the researcher has conducted expert reviews with two experts. One is Expert 5 who has also participated in the evaluation of the prototype of “Evaluation Matrix” app in the design phase of the study. The other one is a software engineer. The demographic information and the

smartphone models they use are as followed:

Table 26. Demographic information of experts participating in the evaluation of coded "Evaluation Matrix" app and smartphone models they use

| **Experts** | **Occupation** | **Expert Field** | **Experience** | **Owned brand and model of smartphone** |
|---|---|---|---|---|
| Expert 5 | Web Developer | Web Development | 3 years | Samsung Galaxy Note 2 |
| Expert 7 | Software Engineer | Software (mainly using C# - t-SQL - XAML) | 3 years | LG G4 |

The process implemented in the expert review was similar with the one conducted in design phase except the fact that while experts have evaluated "Evaluation Matrix" app via a prototype in the design phase, they evaluated the actually coded "Evaluation Matrix" app. While it was possible to interview with Expert 5, to meet with Expert 7 was not possible. Therefore, all required files (Appendix 4) and the installation file of "Evaluation Matrix" app have been sent to him, and he has conducted the evaluation of "Evaluation Matrix" app himself.

Whatever it was progressed via an interview or not, they used "Evaluation Matrix" app by completing a set of predetermined tasks, and evaluated it according to usability heuristics, which are the evaluation criteria. Besides, they completed this evaluation by using their own mobile devices, meaning they downloaded and installed "Evaluation Matrix" app to

their devices. Thus, it was possible to check to what extent "Evaluation Matrix" app works fine on different mobile devices having different screen size and resolution. The result is that Expert 7 has not reported any installing or working error of app whereas Expert 5 has indicated that the app did not work properly on his Samsung Galaxy Note 2 phone. Some functions did not work and even some buttons did not appear at all on his phone. For this reason, he used another Android device having no problem on running "Evaluation Matrix" app in order to continue to the evaluation. The results derived from experts review are as followed:

Table 27. Usability problems detected through Expert Reviews

| Issue ID | Issue/Problem Description | Heuristic violated | Severity | Recommendation | Expert Name |
|---|---|---|---|---|---|
| #1 | Only in the "Intro/Help" page, there is no navigation menu button. This may confuse users. | #3, #4, #12 | 2 | Put navigation menu button into "Intro/Help" page. | Expert 5 |
| #2 | "Skip Intro/Help" button might be a cognitive burden because it is a new thing for the users to learn. | **#4** | 1 | Remove "Skip/Help" button from there | Expert 5 |
| #3 | In "Feedback" page, the user might want to see the error reports he/she sent. Thus, he/she will be able to track his/her reports and see whether they are checked by the app developer | #1 | 1 | Add a new section for this, below to "Send" button | Expert 5 |
| #4 | The users high probably want to see the date of evaluation matrixes they performed in "Archive" page. | #8 | 1 | Insert date data for each evaluation matrix | Expert 5 |
| #5 | In My Account section, I can't change my account picture using "Change Picture" button. It just didn't get activated by clicking on it. | #7 | 2 | By clicking "Change Picture" button, the user should change his/her profile picture. | Expert 7 |
| #6 | When I click back action (all smartphones have one on the left bottom of their screen), it always leads me to Intro section. | #7 | 4 | This action should take the user to the preceding menu/section. | Expert 7 |

| | | | | | |
|---|---|---|---|---|---|
| #7 | While creating evaluation, Number of Idea/Solution/Choice and Number of Criteria lists have no 1 option. These lists start by 2 options. | #1 | 2 | If this is not an error, some short explanation should be given. | Expert 7 |
| #8 | After creating a successful evaluation, an alert message pop-up as "The matrix has been created successfully!". | #4 | 1 | It should be an "info" message, not alert. This could confuse the user. | Expert 7 |
| #9 | After creating evaluation, I didn't calculate it. Then I returned to main page to see if I could find my new evaluation. But there was just my "calculated" evaluation. So, I couldn't found my new evaluation in the "Recently Accomplished Works" list. | #8 | 3 | The user could see his/her evaluations if s/he is the one who created it. | Expert 7 |
| #10 | When I typed 2475 at "Join in Evaluation"->"Enter Code" and clicked Join button, I redirected to "BuyHouse" evaluation. But here, I got an alert message which said "Welcome to BuyHouse evaluation matrix.". | #4 | 1 | It should be an "info" message, not alert. This could confuse the user. | Expert 7 |
| #11 | I entered some values to MyHouse evaluation Matrix then closed it. Then I entered MyHouse evaluation Matrix to see what I just entered but couldn't see them at all. | #6 | 3 | The user should see what value/evaluation they entered when visiting the same Evaluation, they once evaluated. | Expert 7 |
| #12 | When I entered an evaluation such as MyHouse using "Recently accomplished works" it only shows me what I just valued/evaluated. But I wanted to re-evaluate some of them but not allowed to. | #6 | 3 | The user could change some values they gave by using "Recently accomplished works". And also s/he could see what is group average and comments about it. | Expert 7 |
| #13 | When I entered "Archive" section, I encountered a text as "Here is the list of evaluation **matrixed** you've completed.". | #4 | 1 | Are you sure it shouldn't be as …matrix you've completed? | Expert 7 |
| #14 | When I was evaluating BuyHouse evaluation matrix, I saw "Calculate" and "EDIT" buttons were side by side. And Edit button was upper case. | #8 | 1 | The user could only see "Calculate" button when there were no evaluated values, and "Edit" button when there was evaluated value. Also, Edit button shouldn't be completely uppercase. | Expert 7 |

### 4.3.3. User Testing

User testings via interviews by using Think-aloud Protocol have been conducted with four master's students (Appendix 5). User testings have been conducted in similar times with expert reviews. One was conducted in English, one was in Turkish and the other two were in the Korean language. The implementation method of user testings was similar with the one

conducted in the design phase of the study. However, one difference was that users directly evaluated “Evaluated Matrix” app this time, instead of the prototype. In addition, if the users are the owner of an Android device, they downloaded and installed “Evaluation Matrix” app into their devices, and they tested it by accomplishing sets of tasks via “Evaluation Matrix” app existing in their devices. The purpose of doing like that was to find out to what extent “Evaluation Matrix” app works fine in mobile devices having different sizes and resolutions. The result is that there was no any error or problem in working of “Evaluation Matrix” app. Since iPhone and Android may have some kind of different design principles, the researcher has tried to find both platforms’ users for user testings. To sum up, the mobile devices used by the users participated in user testing are as followed:

Table 28. Smartphone models of the users participated to user testings

| **Users** | **The brand of owned smartphone** | **The model of owned smartphone** |
|---|---|---|
| User 4 | Samsung | Note 5 |
| User 5 | iPhone | 5S |
| User 6 | iPhone | 7 plus |
| User 7 | Samsung | Galaxy S6 |

As mentioned before, the implementation method of user testings was similar with the before one performed in the design phase. First, the researcher has provided “Evaluation Matrix” app to the user to install it

his/her phone if the user is an Android user. Then, the researcher requested users to accomplish a set of task with an order, and making thinking-aloud was expected from the users. All interviews were audio recorded, and the researcher coded and analyzed them by listening to the records. The usability problems derived from user testings are shown in Table 29. The same usability issues discovered by different users have been inputted as one issue.

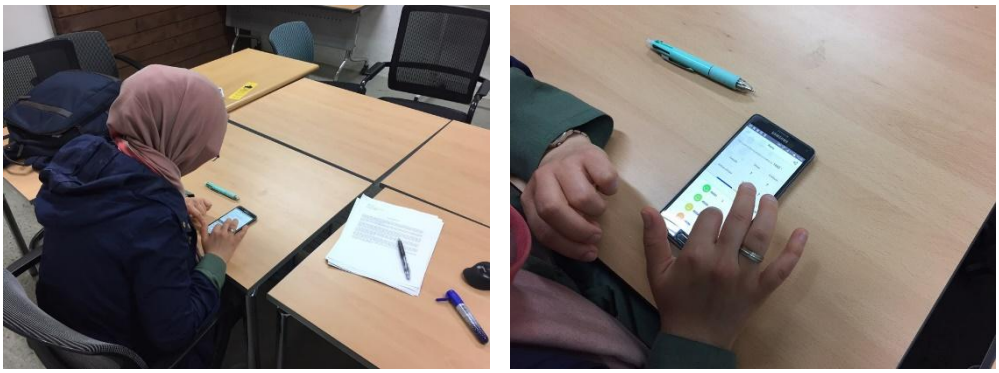


Figure 23. A user testing "Evaluation Matrix" app in a user testing

Table 29. Usability issues derived from user testings

| Issue ID | Issue/Problem Description | Heuristic violated or other issues | Recommendation | User Name |
|---|---|---|---|---|
| #15 | The input textbox used for entering criteria is short. | #3 | Increase its capacity. | User 4 |

| | | | | |
|---|---|---|---|---|
| #16 | Send picture button in the comments section is not working in one click. | Coding Error | Fix it. | User 4 |
| #17 | “Back” button is not working. | #7 | Fix it. | User 4, User 5 |
| #18 | When opening an archived evaluation matrix, the app does not show the mean score of that evaluation. | #4 | Show mean score as well in there | User 4 |
| #19 | The same page is opened when clicking two different buttons, “Create Evaluation” and “Join in Evaluation” buttons. Why? If they are the same, it is pointless. | #8 | There should be either different pages or just one button. | User 4 |
| #20 | The alert box for “evaluation name” textbox and the alert boxes for “Idea number/name” and “criteria number/name” are the same. | #4, #1 | Create a particular alert for each textbox. | User 4 |
| #21 | Sharing a code for making group evaluation might be burdensome since the user has to open the app and enter that code in the required place to access to the group evaluation owning that code. | #8 | Instead of enabling a code sharing for making evaluation together, enable to share a direct link in such a way that when a user clicks the shared link, the group evaluation matrix is directly opened. | User 4 |
| #22 | In “Archive” page, there is no any data about when an evaluation matrix has been accomplished. | #8 | Show the dates of completed evaluation matrixes in “Archive” page | User 4, User 5 |
| #23 | The images of emoticons are too big. Therefore, it occupies much more place on the screen and seeing all rating scores becomes not possible. | #8 | Use much smaller emoticon pictures so that the user can easily see all scores at one glance. | User 6 |
| #24 | When changing account information such as username, there is no sign showing the change has been saved. | #1 | Create a popup saying “It has been saved” for this. | User 6 |
| #25 | In “”Intro/help” page, the orders of the third picture and fourth picture create confusion on the user in understanding app explanation. | #2 | Change the orders of third and fourth pictures. | User 6 |
| #26 | In “Archive” page, there is one sentence including an expression like “… matrixed… ”. It is wrong grammatically. | #4 | Change it as “...matrixes…” | User 6 |
| #27 | The content of the “password reset” email is confusing. It says “… Firebase2 app …”. | #4 | Change its content as “… Evaluation Matrix app”. | User 6 |
| #28 | It is not possible to redo old evaluations. | #3 | Create a “Re-do” button in the page where the archived evaluation matrix is opened. | User 7 |
| #29 | It is possible to re-enter group evaluation page even if individual score has been sent to group. So, a user can send own score more than one time. | #5 | It should be blocked. | User 7 |
| #30 | In “Intro/Help” page, there are some grammatical errors. “Let’s start to evaluation” and “Chat with the group” are wrong grammatically. | #4 | Change “Let's start to evaluation” as “Let’s start to do evaluation” and “Chat with the group” as “Chat with group” | User 7 |
| #31 | When clicking “Log out” button, the opening popup, “Are you sure to log out”, is too long. | #4 | Change it as “Are you sure?” | User 7 |

On the other hand, the users participated to user testings have suggested new features as well for “Evaluation Matrix” app, which were presented in Table 30.

Table 30. Additional features suggested by users.

| Feature ID | Features Suggested | The user who suggested it |
|---|---|---|
| #1 | Adding a sign, symbol or icon for distinguishing whether the evaluation matrix is performed as alone or as a group in “Archive” page. | User 4 |
| #2 | Enabling to insert a picture into criteria or idea elements | User 4 |
| #3 | Enabling to insert detailed information into criteria and idea elements was possible or creating another tab for providing detailed information about criteria and idea elements in Group Evaluation page. | User 5 |
| #4 | Enabling to assign a due date for group evaluation when they are created. | User 5 |
| #5 | Enabling to add a picture for evaluation matrixes when they are created | User 6 |
| #6 | Enabling to check the evaluation scores of each individual who has joined into the group evaluation in “Group Score” page. | User 6 |

### 4.3.4. Make last revision

The last revision has been implemented according to expert reviews and user testings. Whereas some of them have been reflected in the app, some of them could not due to various reasons including the lack of time and the lack of coding expertise of the researcher. Table 31 gives detailed information about revision of “Evaluation Matrix” app. Some usability issues detected by users were the same with the one experts found out. However, most of them different. Those same usability issues have been placed in the same cell in the table.

Table 31. The status of the usability issues derived from expert reviews and user testings

| Issue ID | Fixed or not? | If fixed, how? If not, why? |
|---|---|---|
| #1 | Not fixed | Lack of time |
| #2 | Not fixed | "Skip Intro/Help" button has been requested by users and Expert 1 in design phase. Therefore, this button was not removed. |
| #3 | Not fixed | Lack of time |
| #4, #22 | Fixed | Date information of archived evaluation matrixes has been added |
| #5 | Not fixed | Lack of time (change account picture) |
| #6, #17 | Not fixed | Could not find the error code / Lack of coding expertise of the researcher |
| #7 | Not fixed | There is no need to do this. Because, if there is one idea or criteria, then there is no need to use this app. |
| #8, #10 | Not fixed | Lack of time / Not a relatively important problem |
| #9 | Not fixed | Lack of time / Not a compulsory function |
| #11 | Not fixed | Lack of time / Not a compulsory function |
| #12, 28 | Not fixed | Lack of time |
| #13 | Fixed | The text has been revised. |
| #14 | Not fixed | No need for it since the mentioned button already becomes disabled when it is not needed. |
| #15 | Fixed | The capacity of mentioned textbox has been increased. |
| #16 | Not fixed | Send image button works fine in consecutive four clicks. Its reason could not be found in the code (lack of coding expertise of the researcher) |
| #18 | Not fixed | Lack of time |
| #19 | Not fixed | Lack of time |
| #20 | Fixed | A particular alert for each textbox has been created. |
| #21 | Not fixed | Realizing it exceeds the ability of the researcher |
| #23 | Fixed | The sizes of emoticon images have been decreased. |
| #24 | Fixed | A new popup has been created for this. |
| #25 | Fixed | The third and the fourth pictures have been replaced in "Intro/Help" page. |
| #26 | Fixed | The mentioned text has been revised. |
| #27 | Fixed | The content of "password reset" email has been revised. |
| #29 | Not fixed | Lack of time |

| | | |
|---|---|---|
| #30 | Fixed | The text content of “Intro/Help” page has been revised. |
| #31 | Fixed | The mentioned text has been revised. |

On the other hand, the new features suggested by the users who participated to user testings could not be added to “Evaluation Matrix” app due to the lack of time. Another reason is that the researcher has worried of making “Evaluation Matrix” app more complicated by adding new features. In addition, even if he would have added those new features, he had had to ask the usability of “Evaluation Matrix” app to the users and experts due to the fact that there will might be some users who find the app complicated because of new features and that experts will might find new additional technical problem related to new features. Therefore, the researcher has not added new features and tried to keep “Evaluation Matrix” app simple in order to increase its usability.

In addition to all these, the researcher has tried to conduct user testings with users who own different mobile operating services (OS) such as Android, IOS and Windows Phone users since there might exist differences on user interface according to the OS of the owned mobile device. Therefore, the users participated to user testings were selected in such a way that the half of them is Android users and the other half is IOS users. However, the researcher could not realize any difference, conflict or issue in feedbacks, depending on using different mobile OS devices.

In conclusion, the usability issues derived from expert reviews and user testings have been tried to solve as far as possible though some of them could not be fixed due to lack of time or of the researcher’s ability.

Nevertheless, “Evaluation Matrix” app has been developed and improved as much as being ready to use “Evaluation Matrix” app in the school environment. The researcher has moved to the last step when he has believed that “Evaluation Matrix” app fulfills all indispensable requirements including functional and non-functional ones and when all crucial usability issues have been solved. A showcase to the finalized “Evaluation Matrix” app can be accessed through this link; https://youtu.be/fcds6hK-G_g, and it can be downloaded through this link; https://goo.gl/gXVHMU.

## 4.4.IMPLEMENTATION & EVALUATION

### 4.4.1. Make learners try the Mobile App online (e.g., home) and/or offline (e.g., classroom) environment

The next step after “Evaluation Matrix” app has been developed and revised was its implementation and evaluation. In this step, “Evaluation Matrix” app has been tried out in a real classroom environment, which was a graduate course having 16 students. Before introducing it to the class, the situation of students was that they have already performed various activates through $S^3$CPS system, which utilizes CPS. As an activity, they have already had a problem: “What can be an alternative to “relative evaluation” method?”, and also some solution ideas and associated criteria. “Evaluation Matrix” app has been introduced to class in this situation, where “Evaluation Matrix” app is needed. Overall, the following process has been implemented. First, one day before the class, the researcher has sent the required documents, (Appendix 6) including an introduction of “Evaluation matrix”

app, an instruction about how to use it in the school environment and a download link for it, to one student, who become class leader later for the activity. Then, this student sent the download link and an explanation of "Evaluation Matrix" app to other students and the students having Android device downloaded and installed it to their own devices before coming to class. Besides, he has created a group evaluation activity by entering its name, ideas, and criteria in "Evaluation Matrix" app.

### 4.4.2. Conduct Field Observation

Figure 24. Teamed students using "Evaluation Matrix" app in the class

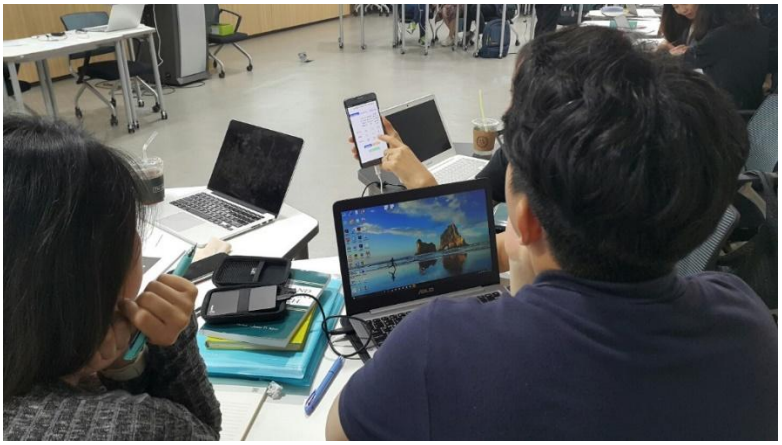


Figure 25. Students using “Evaluation Matrix” app as a team

During the class, the researcher had only observer role and that class leader directed all activity. There were two students who could not install the app; one was could not open the download link, which was a google drive link, and the other one had a problem with her phone. The researcher provided another download link for the former student, and thus her problem was solved. “Install” button was not working in the other student’ device, which is not a problem of “Evaluation Matrix”. Therefore, she just gave up on installing it. Among students, there were only six Android users and one’s phone did not work on installing the app. Hence, there were five active Android devices working “Evaluation Matrix” app without any problem. For this reason, the students formed as teams in such a way that each team has one Android device containing “Evaluation Matrix” app, as seen in Figure 24 and Figure 25. After that, the class leader introduced “Evaluation Matrix” app and explained how to use it and what features it has to the class. Then, the proposed three ideas explained by the students who suggested them. Next, the students started to use “Evaluation Matrix” app. They discussed with other team members, and they gave the common ratings agreed by the all team members to those ideas according to the given criteria by using “Evaluation Matrix” app. Thus, they came to a conclusion about which idea is the most favorable. During all activity, the researcher has observed the students, and it seemed that there was no any problem encountered by the students while using “Evaluation Matrix” app. The following screenshot is from the evaluation matrix performed by the students.

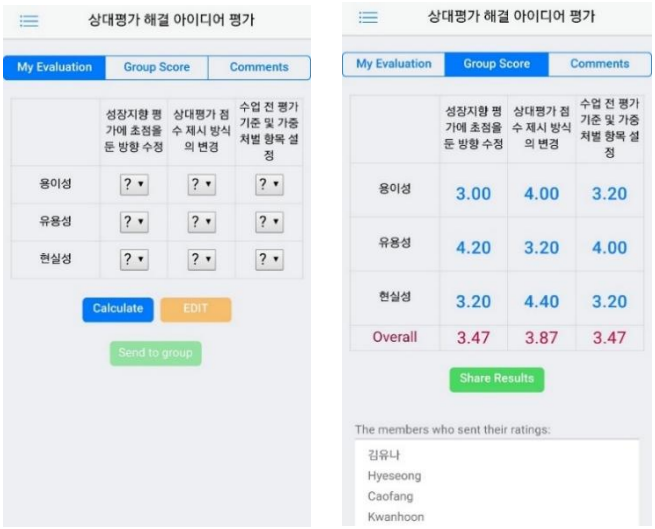



Figure 26. Screenshots from the group evaluation performed by students via "Evaluation Matrix" app: Scoring page (on the left) and Group score page for checking the average of all class scores (on the left)

### 4.4.3. Conduct usability questionnaire

After students have used "Evaluation Matrix" app, the researcher has provided a questionnaire to all students in order to measure the usability of "Evaluation Matrix" app. The researcher has provided it even for the students who did not directly use "Evaluation Matrix" app due to not having Android device. This is because they saw what "Evaluation Matrix" looks like and what it can do by using "Evaluation Matrix" app as teams. The applied questionnaire, a 5-point Likert scale (1- Strongly Disagree, 2-Agree, 3-Neutral, 4-Agree, 5-Strongly Agree), has consisted of 15 items; the first 10 items derived from System Usability Scale (SUS) scale (Brooke, 1996) and the rest 5 items have been created by the researcher. However, the first item

of SUS was modified as “I think that I would like to use “Evaluation Matrix” app frequently for this kind of learning activities.” from “I think that I would like to use this system frequently”, in order to match it to the context, as suggested by Alnuaim (2015). SUS scale is an effective, reliable tool for measuring the usability of a wide variety of products and services (Bangor et al., 2009) as well as mobile apps (Kortum and Sorber, 2015). SUS scale assesses both learnability and usability consisting of effectiveness, efficiency, and satisfaction (Brooke, 2013).

The results of overall usability questionnaire are shown in Table 32. The answers to SUS questions were converted to numbers and calculated according to the SUS scoring formula (Brooke, 1988). The rest five items were interpreted differently. The SUS score of “Evaluation Matrix” was 78.35, which is higher than the average score of 68. According to Bangor et al. (2009), this score is good within the acceptable range of SUS scores and is about in the middle between the markers for good and excellent (Figure 16). These results imply “Evaluation Matrix” app being successful with respect to usability and learnability.

Table 32. Usability Questionnaire items and their mean scores (N=16)

| # | Statements | Mean |
|---|---|---|
| 1 | I think that I would like to use “Evaluation Matrix” app frequently for this kind of learning activities. | **4.00** |
| 2 | I found “Evaluation Matrix” app unnecessarily complex. | **2.13** |
| 3 | I thought “Evaluation Matrix” app was easy to use. | **4.38** |

| | | |
|---|---|---|
| 4 | I think that I would need the support of a technical person to be able to use "Evaluation Matrix" app. | **2.13** |
| 5 | I found the various functions in "Evaluation Matrix" app were well integrated. | **3.88** |
| 6 | I thought there was too much inconsistency in "Evaluation Matrix" app. | **1.81** |
| 7 | I would imagine that most people would learn to use "Evaluation Matrix" app very quickly. | **4.38** |
| 8 | I found "Evaluation Matrix" app very cumbersome to use. | **1.88** |
| 9 | I felt very confident using "Evaluation Matrix" app. | **4.38** |
| 10 | I needed to learn a lot of things before I could get going with "Evaluation Matrix" app. | **1.69** |
| 11 | "Evaluation Matrix" app has helped in conducting group evaluation. | **4.50** |
| 12 | "Evaluation Matrix" app has supported creative learning environment. | **4.31** |
| 13 | "Evaluation Matrix" app has supported collaborative learning environment. | **4.50** |
| 14 | "Evaluation Matrix" app has similar User Interface with the apps I use daily. | **3.63** |
| 15 | I would use similar apps for supporting learning environment if possible. | **3.75** |

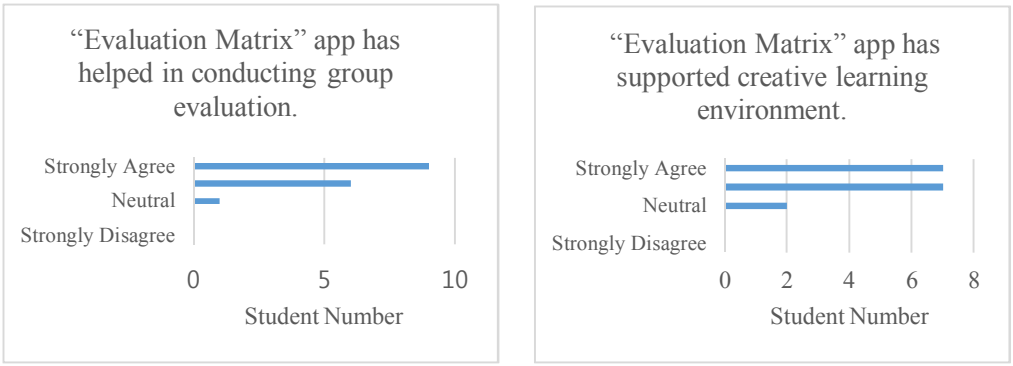



Figure 27. Students' responds to Question 11 (on the left) and 12 (on the right) (N=16)

On the other hand, the results of the rest 5 items are as followed. 94% of students have agreed or strongly agreed with the statement “Evaluation Matrix” app has helped in conducting group evaluation”. 88% of them have agreed or strongly agreed with the statement “Evaluation Matrix app has supported creative learning environment”. All students have agreed or strongly agreed with the statement “Evaluation Matrix app has supported creative learning environment”. The responses to these three questions indicate that “Evaluation Matrix” app is successful with respect to the pedagogical aspect. Also, 63% of students have agreed or strongly agreed with the expression “Evaluation Matrix” app has a similar user interface with the apps I use daily”. This means that “Evaluation Matrix” app has a similar user interface with native apps, which are frequently used daily. That is because “Evaluation Matrix” app utilizes Framework7 UI promising native look & feel interface for hybrid apps. Therefore, it can be easily said that Framework7 kept its word. Lastly, 69% of them have agreed or strongly agreed with the statement “I would use similar apps for supporting learning environment if possible”. This implies that the students are open to using mobile apps for learning purposes.

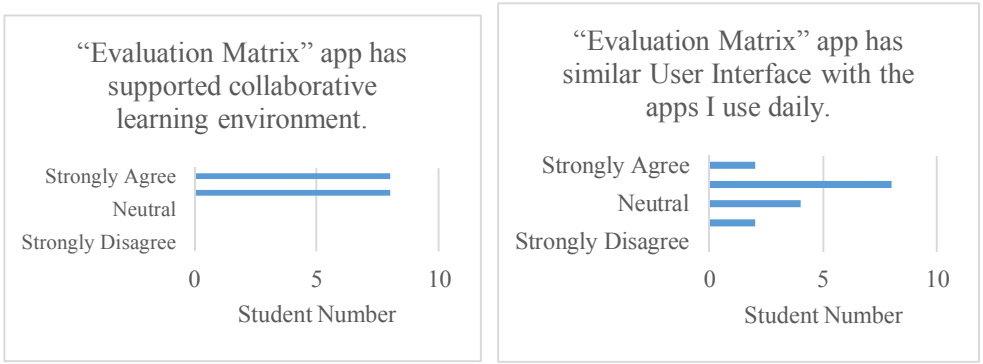



Figure 28. Students’ responds to Question 13 (on the left) and 14 (on the right) (N=16)

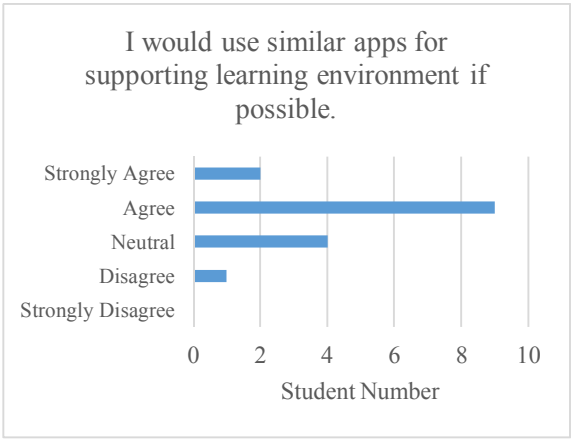



Figure 29. Students' responds to Question 15 (N=16)

### 4.4.4. Conduct semi-structured interview

Semi-constructed interviews were conducted with three students after "Evaluation Matrix" app was tried in a classroom environment. Each interview took between 20 and 30 minutes. The researcher asked them about what they liked the most and the least about the app, what difficulty they experienced when using "Evaluation Matrix" app, and what can be added to "Evaluation Matrix" app. All interviews were recorded and analyzed later by the researcher. The interview with Student 1 was conducted right after the classroom activity. However, it was conducted with Student 2 the day after the classroom activity and it with Student 3 two-days after the classroom activity. The two own an Android device and the last one owns an IOS device. Therefore, he used a friend's Android device for this activity. During the activity, he was the group leader and he was the one who creates evaluation matrix by entering its name, ideas and criteria and shares the unique code with others. Besides, he was the one who controls and facilities

all class activity in which “Evaluation Matrix” app is utilized. Therefore, his opinions were considerably important. The mobile devices owned by these students and their roles during class activity are shown in the following table.

Table 33. The mobile devices owned by the interviewees and their roles in class activity

| **Student Name** | **Mobile device owned** | **Role in class activity** |
|---|---|---|
| Student 1 | Samsung Galaxy S8 | Group member |
| Student 2 | Samsung Note 5 | Group member |
| Student 3 | iPhone 8 | Group leader |

While Table 34 presents what interviewed students liked the most about “Evaluation Matrix” app, Table 35 shows what they liked the least about it. The researcher has analyzed the interviews and categorized their contents as *Functional feature, Non-functional, User Interface* and *Pedagogy*. Student 1 mostly liked functional features of “Evaluation Matrix” app, such as being able to see all ideas and criteria at one glance when performing an evaluation and comments feature. Although the researcher has asked for only three things they liked the most, Student 2 stated more. She mostly commented on “Evaluation Matrix” app with respect to its user interface. She stated that *it is easy to use,* which was also indicated by Student 3, and *it is a lightweight app* and *has a convenient interface*. The last interviewed student, Student 3, who controls and facilities all class activity, expressed that it is *helpful to make collaboration with other people*. To sum up, these interview results imply that “Evaluation Matrix” app is a successful app with respect its functional features, user interface and pedagogical aspects.

Table 34. The strengths of "Evaluation Matrix" app perceived by interviewed students

| Category | Statement | Student Name |
|---|---|---|
| Functional feature | Being able to see all ideas and criteria at one glance while making an evaluation. | Student 1 |
| Function feature | Being able to compare and contrast ideas and criteria through their scores | Student 1 |
| Functional feature | Comments feature | Student 1 |
| User Interface | Easy to use | Student 2, Student 3 |
| User Interface | Lightweight app | Student 2 |
| User Interface | Having a convenient interface | Student 2 |
| Functional feature | Easy registration system | Student 2 |
| User Interface | The letters being easy to recognize | Student 2 |
| Functional feature | Easy to remember the code and enter a group evaluation by using it since the code is 4-digits. | Student 2 |
| Pedagogy | Helpful to make collaboration with other people | Student 3 |
| Functional feature | Being able to share via social networking services | Student 3 |

On the other hand, the interviewed students' responses were related mostly to the functional features of "Evaluation Matrix" app. Besides, since only Android version was available, other problems occurred and mentioned during the interview. Student 1 specified that when they perform a group evaluation activity, giving ratings with only one device as one team was difficult. That is because team members have have to persuade each other and come to an agreement on one rating score, which was quite tough. However, these results imply that students were satisfied with the user interface of the app since there were a few critics on the user interface, which

were relatively less important.

Table 35. The weaknesses of "Evaluation Matrix" app perceived by the interviewed students

| Category | Statement | Student Name |
|---|---|---|
| Non-functional feature | Having only Android version | Student 1 |
| Pedagogy | Giving rating with only one device as one team when performing a group evaluation activity | Student 1 |
| Functional feature | Not being able to write an additional explanation for ideas and criteria of an evaluation matrix. | Student 2 |
| Functional feature | Not refreshing automatically the list of the users who sent their scores, in "Group Score" page. | Student 2 |
| User Interface | Having inconsistent looks in different devices | Student 2 |
| Functional feature | Having some errors in some points | Student 3 |
| User Interface | Term (word)-centered interface | Student 3 |
| Functional feature | Being not working as expected in some Android devices | Student 3 |

When interviewed students asked about additional features for "Evaluation Matrix", they suggested seven features, which were functional. Additional features recommended by Student 2 were for promoting scoring part of the app. Student 2 recommended some additional features for the commenting feature of it. Lastly, student 3 has advised putting an additional instruction into the main page of the app. Overall suggested features are summarized in Table 36.

Table 36. Additional features suggested by the interviewed students

| Category | Statement | Student Name |
|---|---|---|
| Functional feature | Enabling users to write an explanation for the score they gave | Student 1 |
| Functional feature | Enabling to check others' evaluation scores when performing a group evaluation | Student 1 |
| Functional feature | Enabling to write an explanation for ideas and criteria of an evaluation matrix | Student 2 |
| Functional feature | Making enable to select a subject when writing a comment and list the comments by the concerned subject | Student 2 |
| Functional feature | Enabling to be refreshed the list of the group members who sent their ratings as real-time | Student 2, Student 3 |
| Functional feature | Adding a "Like" button for comments | Student 2 |
| Functional feature | Adding a new section into the main page, explaining briefly the app usage | Student 3 |

The responses about experienced difficulties when using "Evaluation Matrix" app were different. Even if Student 1 did not experience any problem, error or difficulty when using the app, Student 2 and 3 exposed some difficulties. They specified that some buttons or textboxes did not appear at all on their mobile devices. The root of this problem was that they use different font styles on their mobile devices, rather than the default font of the phone and the size of that font was bigger than the default one. During the interview with Student 2, the researcher asked what font she uses in her phone, and she said different from the default one. When the researcher requested her to change it as default, there was no problem in the app and it was working fine. However, when she changed the font of mobile device again, the problem occurred again. The reason for why this problem happened is that "Evaluation Matrix" app utilizes the font used by the mobile

device it is installed, and because the device's customized font style is bigger than normal, the buttons or other elements of interface and the font used by the app change and even some of them disappear.

Table 37. Difficulties experienced by the learners

| Student Name | Statement | Mobile device used |
|---|---|---|
| Student 2 | The textboxes used for entering the names of ideas and criteria did not appear in my device | Samsung Note 5 |
| Student 3 | Share button did not appear on my device | Samsung Galaxy S3 |

In the end of the interview, the researcher asked them their last comments about "Evaluation Matrix" app with respect to its user interface and features. Student 1 stated that the app is good and intuitive, and it can be used well with $S^3$CPS, which utilizes CPS. However, she also stated that she confused about what she should do with the app in beginning of the class activity. Student 2 indicated that the user interface elements look much bigger, differently than others' devices. Therefore, she suggested controlling the appearance of "Evaluation Matrix" in such a way that it will be the same regardless of the model of Android device having different screen sizes. Lastly, she strongly recommended regulating the font style used in "Evaluation Matrix", so that it will not change on different devices utilizing different font style and sizes.

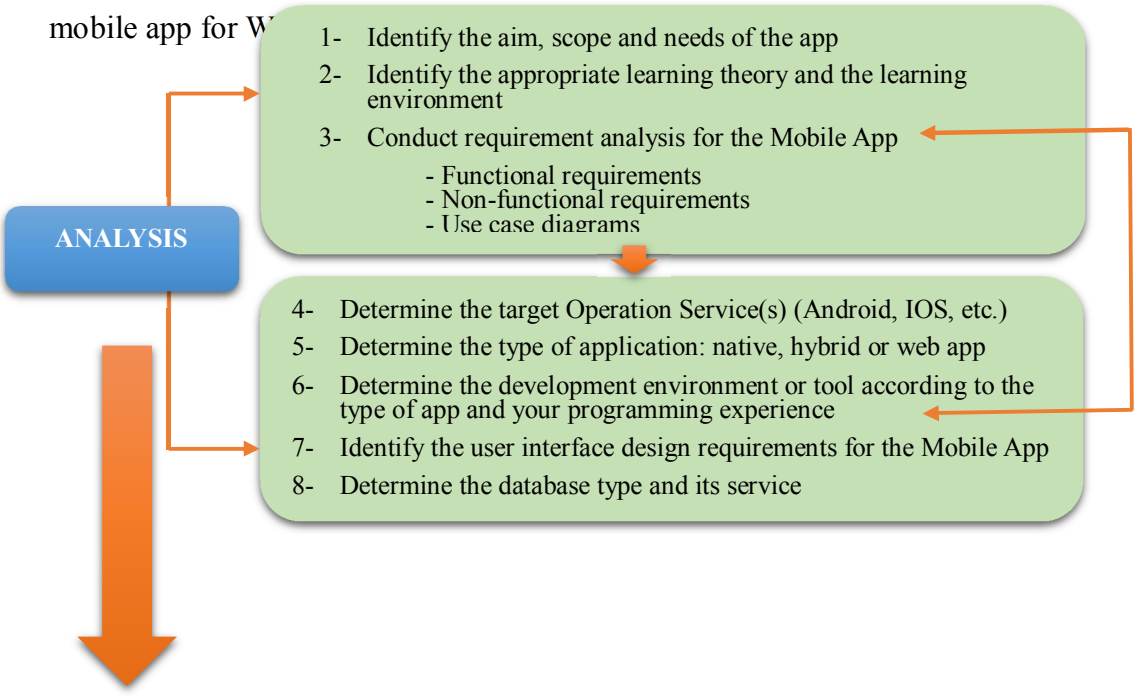
## 5. The Final Model for Developing a Simple Educational Mobile Application (MODSEMA)

compliance

According to the experience obtained through the case study conducted, the final model proposed in this research is presented in Figure 30. The revised or added steps are discussed in this section.

### 5.1. Changing the Place of "Identify the user interface design requirements for the Mobile App" Step

The first change in the model is on the step "Identify the user interface design requirements for the Mobile App" placing in *Analysis* phase. Previously, it was right after conducting requirement analysis. However, during conducting the case, the researcher has realized that some UI frameworks only works for the certain type of mobile apps. For instance, it is not possible to utilize Mobile Angular UI for developing a native app. Besides, each of UI Frameworks supports different mobile platforms. For example, Framework7 utilized in the case study of this research does support only IOS and Android devices, meaning it is not possible to develop a mobile app for W



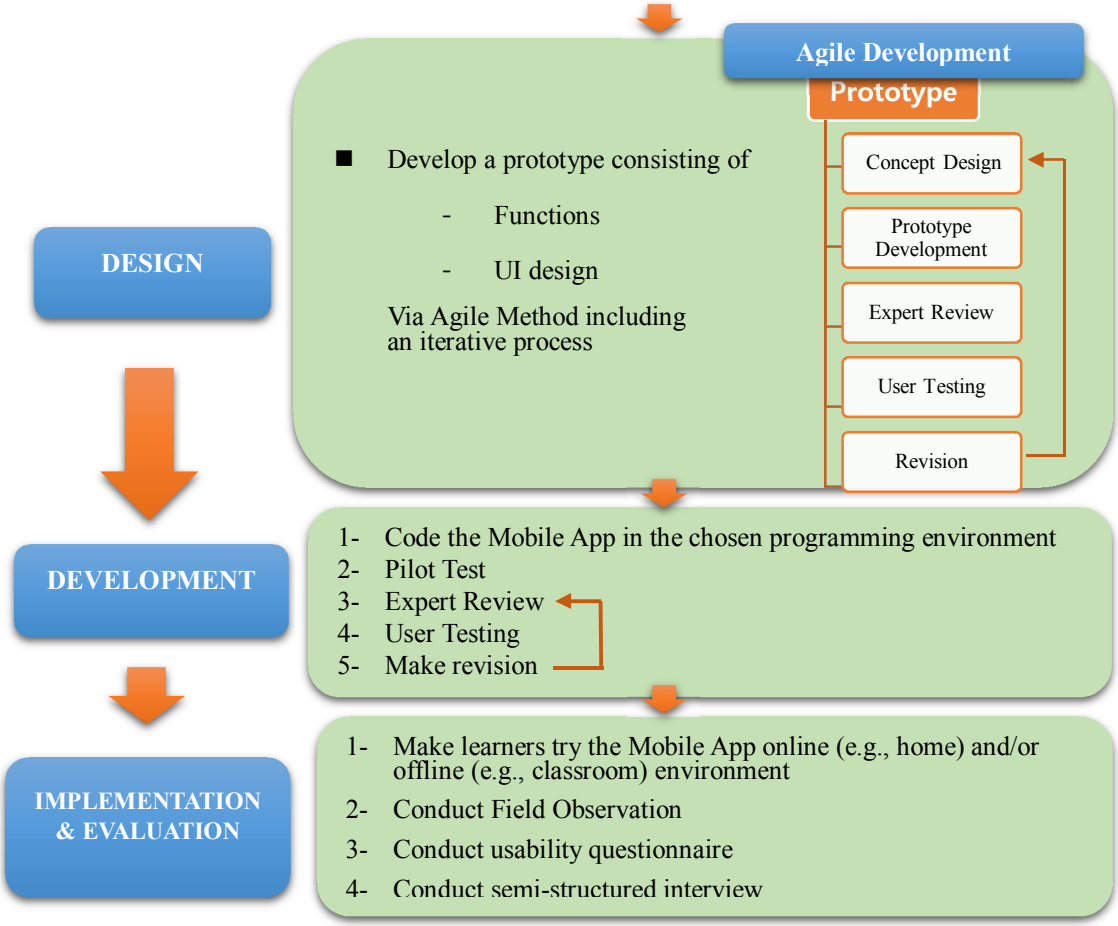



Figure 30. The final Model for Developing a Simple Educational Mobile Application (MODSEMA)

Therefore, after the developer decides which kind of app including hybrid, native or web he/she will develop and which kind of development environment such as PhoneGap he/she will use for it, he/she should perform the step "*Identify the user interface design requirements for the mobile app*".

### 5.2.Inserting "Determine the database type and its service" Step

**into Analysis Phase**

The opportunities provided for developers have become enriched through the evolving technology day by day. This is same for database systems as well. One option, SQL (Structured Query Language) databases have been a primary data storage mechanism for more than four decades. Usage exploded in the late 1990s with the rise of web applications and open-source options such as MySQL, PostgreSQL and SQLite. While SQL owns a relational database structure, NOSQL, which is the second option, provided a non-relational database structure. NoSQL databases have existed since the 1960s, but have been recently gaining traction with popular options such as MongoDB, CouchDB, Redis and Apache Cassandra. However, these two options have own pros and cons. Therefore, when a developer selects a database structure, he should be realized their pros and cons and choose the most appropriate one to his/her designated database.

On the other hand, there exists a third option, BaaS (Backend as a Service), implying utilizing online services for all database work. The only thing the developer needs to do is to integrate their APIs with his/her mobile app. These services set up all required database related works. They usually provide simple interfaces for database works such as data storage and retrieval, authentication and social media integrations. In the case study performed in this research, the researcher has utilized Firebase service, which is a BaaS, for dealing with all database works of “Evaluation Matrix” app.

### 5.3.Adding “Pilot Test” Step into Development Phase

After the researcher has coded “Evaluation Matrix” app for the first time, the researcher has felt the need of conducting a pilot study with a representative user. That was because the researcher intended to eliminate the obvious usability issues before starting to conduct an expert review and user testing. The results of the pilot test show that this was the case. Thanks to pilot test, the researcher fixed the evident usability problems that the researcher could not realize. Thus, pilot test helped conduct healthier usability evaluations including expert review and user testing.

### 5.4.Providing an Iterative Process inside of Development Phase

The last revision applied to the model is related to the *Development* phase. Normally, there was no iteration in development phase because the researcher thought that it is needless for a simple app and the mobile app development experts also confirmed it during interviews. However, during conducting the case study, 27 usability issues have been detected through user testings and expert reviews. Although 20 of 21 usability issues discovered in design phase have been resolved, still there were another 27 usability issues detected in the development phase. Therefore, after revising the app according to discovered usability issues, the developer should conduct evaluations again for the app in the development phase. Therefore, it is much appropriate to provide an iteration process inside of development phase.

# CHAPTER V: DISCUSSION & CONCLUSION

## 1. DISCUSSION

The purpose of this study was to construct a development model for simple educational mobile applications and to test this development model whereby a case study producing a simple educational mobile app by using this model for validation. For this, it built a development model according to literature and revised it through the expert reviews. Then, the researcher himself developed an educational mobile app named Evaluation Matrix by following this model in order to validate it via the usability questionnaires and interviews that are the resultant of the case study. This chapter discusses the implications of the essential results of the study.

### 1.1.Implications on MODSEMA

The initial MODSEMA was developed through the literature review, especially benefited from Berking et al.'s ADL Mobile Learning Framework (2012), Al-Harrasi's m-learning design approach (2015) and user-centered design (Abras et al., 2004). Specifically, its *analysis* and *design* phase has experienced a variety of modifications and enhancements whereby expert interviews on MODSEMA and the case study. Concerning to *analysis* phase, the logical order of its steps were set thanks to those expert interviews and it has been ensured to extract the planned educational mobile application's functional and non-functional features and use case diagram. Although it might take a long time to extract all these features and it might get modifications due to usability evaluations conducted as a part of the app development process, it seems that analyzing the features in advance as much as possible is helpful and efficient with respect to time for the app development process. Another issue needed to be pointed out in analysis phase is database solutions. The evolving technology improve not only

hardware systems, but also software systems. In earlier times, the developer was mostly applying the one he/she knows. However, recently, there have been developed new technologies and solutions for database issues. There is not only SQL but also NoSQL database solutions. Besides, thanks to newly developed cloud technologies, there are online database services such as Firebase provided by Google. Whereas all of them have own pros and cons, the some are extremely easy and simple to apply and the some are especially useful for some conditions. Therefore, having a core knowledge on database solutions will be considerably helpful for developers. Accordingly, the analysis phase of ultimate MODSEMA contains this step as well and provides a basic guideline on database solutions. The other issue required to be stated is the availability of various development environments, tools and UI. Thanks to evolving technology, developing a mobile application becomes much easy each passing day by providing much easy and simple new tools and environments for app development. However, each of them has its own pros such as creating apps for multiple platforms including IOS and Android with a single coding and cons such as not supporting some APIs like camera and GPS functions. In other words, in earlier times, it was only available to develop a *good* mobile application by creating it as a native app that is produced through a platform-specific development environment and language such as using Eclipse with Java for developing an app for Android devices and using Xcode with Objective-C for creating IOS app. However, recently, it is possible to develop a *powerful* app as much as a native app with respect to its performance and UI by utilizing other programming languages and environments, and even to create an app for multiple platforms with a single code is possible, which meaning huge saving on time,

effort and money. Yet, they also have some drawbacks due to the novelty of the technology. Therefore, it might be beneficial to have knowledge on these various environments instead of knowing only one system/environment/tool/programming language. Thus, MODSEMA contains the corresponding step and guideline, which is absent in most models proposed in literature for app development.

With regard to *design* phase, it was assumed in the initial MODSEMA that developing more than one prototype would be much better because there might developed various prototypes having different concepts and the user would choose one of them for ensuring user-centered design. However, the implications of interviews conducted with experts on app development show that this method is problematic. First, it is loss of time, and second is that the developer has to discard the prototype not chosen even if users favor of its several parts that are disliked in the chosen one. In addition, it is loss of effort due to creating various prototypes. Therefore, the ultimate MODSEMA suggests creating one prototype and making modifications on it through reflecting the results of the usability evaluations in an iterative loop, which was suggested by those experts and confirmed through the case study.

Another issue needed to be discussed related to *design* and *development* phase is how to interpret and reflect the feedbacks derived from the usability evaluations. Usability evaluations suggested in this study consist of expert evaluation conducted with experts and user testing performed with representative app users. To understand what users like and dislike and what experts see a problem on the app is still an issue. In order to

deal with this challenge, this study suggests applying *usability heuristics* (Nielson, 1994) to usability evaluation process. Thus, to comprehend what is the problem with the app, whether it is a prototype or coded, might be much effortless and smooth for the developer. However, users sometimes might provide opposed comments with each other. In this situation, listening the dominant idea might be much logical so that the more conducting user testing, the better user-centered app. However, what the developer should do when there is a conflict between the comments of users and experts? The developer might follow according the experts' views since they are experts. However, since the people who will use the app is the users, not experts, it might be much convenient to listen to users' feedbacks instead of experts'. In this study, for example, the suggestions provided by experts was not accepted a few times by the representative users. Lastly, especially in design phase, obtaining and reflecting users' feedbacks and suggestions seems much helpful for effective concept construction. If the prototype were constructed perfectly in design phase according to the results of usability evaluations, then the workload of the developer would become less in development phase since he/she just needs to develop the app identical with the prototype and does not have to consider its UI.

### 1.2.Implications on "Evaluation Matrix" app

The produced app through the case study conducted as a part of this research is "Evaluation Matrix" app and used as a convergent thinking tool of CPS which promotes creativity. It is suitable for only offline and blended courses.

From a technical aspect, "Evaluation Matrix" app was developed as a

hybrid app in PhoneGap platform by utilizing Framework7 as UI design and Firebase as database solution. The usability questionnaire and interview results explicitly showed that the app is successful as much as a native app with respect to its high quality UI and high performance. This implies that even if people do not know platform-based programming languages, Java for Android and Objective-C for IOS, which are relatively difficult languages, they can create a powerful Android or IOS apps by utilizing different programming languages. In this study, for example, the researcher benefited from his web development experience and used PhoneGap utilizing HTML, JavaScript and CSS. Therefore, anyone who has expertise on any programming language is able to develop a mobile application successfully.

"Evaluation Matrix" app accomplished its duty successfully in the classroom implementation as interpreted from the classroom observation and usability questionnaires and interviews. It was tested in a class in which $S^3$CPS, a CPS-based system, was being implemented and it was right time for utilizing "Evaluation Matrix" app. However, this app might be useful the most only when CPS and CPS-based systems are implemented in a class. That is because "Evaluation Matrix" tool is not sufficient alone in promoting creativity since it is used only for finding out the most optimal solution/idea/choice among variety of solutions/ideas/choices. They need to be extracted first through activities like brainstorming, then "Evaluation Matrix" app can be used. Furthermore, whereas this app can be utilized effectively for offline and blended course, it might be difficult to use in 100% online courses due to its lack of functionality supporting online courses. If it is utilized in an online course, then its users might not comprehend the context of the ideas and criteria and the evaluated topic, and this might cause

inaccurate user evaluations. Therefore, it should be ensured that the learners utilizing this app need to fully understand what the inputted idea or criteria actually means, which might be assured through a blended class or offline class in which learners interact with each other.

### 1.3.Suggestions for App Developers

The first is that developing a mobile application is not an easy work as alone especially for novices due to the required expertise and different roles like acting as an analyst, designer, coder and tester. The corporates developing mobile apps have different-sized app development teams including product managers, UI (User Interface) and UX (User Experience) designers, developers, database experts, testers and so on. Even some big companies have separate teams for each of these roles, as mentioned by Expert 3. However, it is still possible to develop successful simple mobile applications if having full knowledge of overall mobile app development process, as referred by Expert 2. Therefore, instructors, educators and developers who intend to develop an educational mobile app should know the full process of app development.

Secondly, although this development model provides steps for developing a simple educational mobile app, it might also be used for developing big-scaled apps. The only thing the developer(s) should do differently is to divide the intended big-scaled app into *modules* and apply this development model for each module. That is because Expert 1 and Expert 2 has stated during the interview that they split the big-scaled app that they develop into parts, named modules, in order to develop it more efficiently and effectively thought this development model does not need it

since it is for developing simple educational mobile apps.

Thirdly, with respect to *Analysis* phase, the person who wants to develop a mobile app should know the fact that the results of requirement analysis conducted in *Analysis* phase can (or should) change or be modified during app development process due to the usability evaluations resulting in feature suggestions for the app. In the case study conducted in this study, although the required features have been extracted from literature possessing a vast information on it, these features were improved thanks to user testings providing additional feature suggestions by the representative users.

Related to the *Design* phase, in order to evaluate the user interface or the prototype of the mobile app, the developers should apply not only usability inspection methods such as expert review using heuristics, but also user testing methods like user testing utilizing question-asking protocol. While usability inspection methods are utilized in order to examine and evaluate whether each element of a user interface or prototype follows established usability principles, user testing methods are for testing the attributes of the product in order to see and check how user interface promotes the users to do their task (Folmer & Bosch, 2004). Therefore, both of them have own benefits. Besides, the development model proposed in this study recommends developers to conduct *expert review using heuristics* with experts from various fields such as mobile app developers, software engineers, design experts and the expert of the domain the app is developed whereas there are various available usability inspection strategies, as mentioned in Chapter IV. Thus, it will be possible to perform systematically and neatly mobile app evaluations by evaluators, and it will also help the

developer understand more clearly the usability issues on UI or the prototype of the app and resolve them. On the other hand, utilizing question-asking protocol can be quite useful especially for the situations when the developers are unsure about where he/she should put UI elements such as buttons and features into the app. The users are sure on what they like or dislike and they can provide answers developers on those unsure issues, as in the case study conduct in this research. Lastly, the developers need to know about the effects of the evaluators' role that experts mostly detect logical & technical issues such as whether a feature works fine and whether it provides user error controller, the representative users as evaluators mainly identify interface problems such as font style and wording used in the app.

Concerning to *Development* phase, the most important issue the developers should take into consideration during app coding is the compatibility of the platforms, services they use. Since a new platform or service has been released every day, the developers have plentiful options. However, there might occur compatibility problems among them due to their novelty, implying that they might have not a well-established system. Even some code sets, working in one platform, might not work when using several platforms, which was happened during the case study. Therefore, the developers should be careful on compatibility issue. The second significant issue is dealing with unexpected errors/problems occurred during coding. No matter how much the developer is a professional developer, there is always the possibility of encountering unexpected errors/problems he/she cannot solve. For this kind of situations, one of the best ways is to apply the community of that code environment, platform or service. The bigger the community is, the higher to find a solution for those unexpected

errors/problems. Therefore, the developers should take into account to what extend the platform, service or code environment they use has a big community in an online environment when they code their apps. Lastly, the developers should test their apps on different mobile devices having different sizes and resolutions, as much as possible because each different mobile device model means another variable for the app. For this, one solution might be conducting an expert review and the user testing by using the evaluators' mobile devices, as happened in the case study part of this research. In addition, testing the app in different devices of different users is crucial since there might be encountered another sort of problems or issues. For instance, in the case study conducted in this research, the research tested "Evaluation Matrix" app in five different models of Samsung and LG. However, during implementing the app in a classroom environment, two students have encountered quite unexpected problems with the app, which was mentioned in Chapter 3- Section 4.1. Therefore, the developers should test their apps in different devices having different variations, as much as possible in the *Development* phase as preventive measures against possible problems that are confronted during implementing the app in a classroom environment.

With respect to *Implementation & Evaluation* phase, the class facilitators should be well informed about the app and provide user manual related to how to use it in classroom activities. Besides, these facilitators whether they is an instructor, teacher, teaching student or student are required to sufficiently explain students the app, how to use it and what will they perform with the app. In this study, one interviewed student has specified that she is confused about what she should do with "Evaluation

Matrix" app when the facilitator explained it in the beginning of the classroom activity. Additionally, students should be exposed to the app for at least a short time before starting to the classroom activity using it. Thus, it will be much easier for them to perform activities by using it. It should also be ensured that every student installed and launched the app without no problem since there were some students experienced difficulty on it during the case study conducted.

# 2. CONCLUSION

## 2.1.Summary and Conclusion

The research questions of this study were what the steps in developing a mobile application for educational purposes are and what the usability of "Evaluation Matrix" app developed following these steps is.

In order to find out answers to these questions, first, this study has constructed an initial model for developing a simple educational mobile application (MODSEMA) according to the analysis results of the literature. Second, in-depth interviews with three mobile app development experts have been conducted on MODSEMA. According to its results, MODSEMA has been revised. After that, the researcher has validated MODSEMA through a case study. He has developed an educational mobile app named "Evaluation Matrix" by implementing MODSEMA. This process mainly consisted of analysis phase, in which required analysis for the app development is

conducted, design phase, where the intended app's prototype including its UI design and features is developed and revised through the expert review and user testing, development phase, where the app is coded on the basis of the revised prototype in the predetermined coding environment and revised whereby expert review and user testing, and implementation & evaluation phase, where the developed app is implemented in school environment and evaluated via usability questionnaire and interviews with students. After that, MODSEMA has been revised finally according to the obtained experience through the case study, and it has proposed as a result of this study.

Overall, MODSEMA includes "Analysis", "Design", "Development" and "Implementation & Evaluation" phases. Analysis phase consists of "1- Identify the aim, scope and needs of the app", "2- Identify the appropriate learning theory and the learning environment", "3- Conduct requirement analysis for the Mobile App" encompassing "Functional requirements", "Non-functional requirements" and "Use case diagrams", "4- Determine the target Operation Service(s) (Android, IOS, etc.)", "5- Determine the type of application: native, hybrid or web app", "6- Determine the development environment or tool according to the type of app and your programming experience", "7- Identify the user interface design requirements for the Mobile App", "8- Determine the database structure and service". Design phase includes developing a prototype consisting of the designated app's functions and UI design by performing iteratively the following steps;"1- Concept Design", "2-Prototype Development", "3-Expert Review", and "4- User Testing". Development phase contains the followings in an iterative process: "1-Code the Mobile App in the chosen programming environment", "2- Pilot Test", "3- Expert Review", "4- User Testing" and "5- Make last

revision". Implementation & Evaluation phase involves "1- Make learners try the Mobile App online (e.g., home) and/or offline (e.g., classroom) environment", "2- Conduct Field Observation", "3- Conduct usability questionnaire" and "4- Conduct semi-structured interview".

During conducting the case study, the followings have been experienced. In the design phase, expert review using heuristics, conducted with three experts, has detected 15 usability issues and user testing applying question-asking protocol has identified 17 usability problems. Even though there were some similar issues detected by both expert review and user testing, there were 24 unique usability problems. Among them, only one of them could not be fixed. Also, 6 new features have been suggested by representative users during user testing and only two of them have been added due to reasons mentioned in Chapter IV-3.2. In the development phase, thanks to pilot test, 6 usability errors have been detected and 5 of them have been solved. Besides, 14 and 17 usability issues have been extracted through the expert review using heuristic and the user testing applying think-aloud protocol, respectively. 27 of them were unique problems and 16 of them could be fixed because of various reasons such as lack of time. In implementation & evaluation phase, the usability questionnaire, which encompasses SUS scale items and the researcher-created items, has been conducted with 16 students, and usability score derived from it was 78.34, implying "Evaluation Matrix" app is a good app with respect to not only usability but also learnability criteria (Bangor et al., 2008), which also validates MODSEMA. The student responds to the researcher-generated questionnaire items has shown that "Evaluation Matrix" app has become successful with respect to pedagogical perspective because they found

"Evaluation Matrix" app as helpful in conducting group evaluation and supportive for creative and collaborative learning environments. Besides, 63% of them have found "Evaluation Matrix" app having similar UI with their daily used apps. This means that thanks to Framework7 used as UI framework for "Evaluation Matrix" app, it has successfully possessed a native look & feel UI whereas it is originally a hybrid app. Lastly, through the interviews with 3 students conducted after the classroom implementation, these results have been proved again. When students were asked about what are the three things they liked the most about "Evaluation Matrix" app, they provided feedbacks on not only its user interface but also its functional features and pedagogical aspects. Two of them have stated *it is easy to use* (Student 2 and Student 3), as their first comments. Student 2 also emphasized that it is a *lightweight app* and *having a convenient interface.* Apart from these, Student 1 approved its functional features including commenting feature and its concept of the evaluation method. Furthermore, Student 3 stated it is *helpful to make collaboration with other people,* with respect to its pedagogical aspect. However, the results of interviews have shown some weaknesses of "Evaluation Matrix" app as well. The most crucial weakness was being available on Android devices. Since there is no IOS version, the classroom activity using it has been affected negatively. Other weaknesses derived from interviews were small-scaled issues, and they could have been fixed if there is enough time. The summary of evaluation techniques applied in the case study is presented according to the applied phase in Table 38.

Table 38. Summary of usability evaluations performed in case study

| **Phase** | **Type of Evaluation Method** | **Usability technique/ tool applied** | **Purpose** | **Participants** | **Number of detected usability issue** |
|---|---|---|---|---|---|
| Design | Usability Inspection | Expert Review using heuristics | To examine and evaluate whether each element of a prototype follows established usability principles | 1 design expert, 1 domain expert, 1 web development expert | 15 issues |
| Design | Usability Testing | User Testing using question-asking protocol | To test the attributes of the prototype, in order to see and check how UI supports the users to do their tasks | 3 representative users (graduate students) | 17 issues |
| Development | Usability testing | Pilot Test | To conduct more healthier expert review and user testing by eliminating obvious usability issues | 1 representative user (undergraduate student) | 6 issues |
| Development | Usability Inspection | Expert Review using Heuristics | To examine and evaluate whether each element of the app follows established usability principles | 1 software expert, 1 web development expert | 14 issues |
| Development | Usability Testing | User Testing using Think-aloud protocol | To test the attributes of the app in order to see how UI supports the users to do their tasks | 4 representative users (graduate students) | 17 issues |
| Implementation & Evaluation | Usability Inquiry | Observation | To observe classroom implementation of the app | 16 students | - |
| Implementation & Evaluation | Usability Inquiry | Questionnaire (SUS scale + Researcher-generated 5 items) | To acquire information about users likes, dislikes, needs and understanding of the app by letting them answer questions in written form | 16 students | - |

| | | | | | |
|---|---|---|---|---|---|
| Implementation & Evaluation | Usability Inquiry | Interview (semi-constructed interview) | To acquire information about users likes, dislikes, needs and understanding of the app by letting them answer questions verbally | 3 students | - |

This study has a significance due to the followings. First, there was no study encompassing the all app development process at this scope, from A to Z. This study provides not only a model for developing educational mobile apps but also delivers a detailed guidance with respect to how to apply this model. Even it shows how to apply the model whereby a case study conducted following this model. It also serves information on major platforms, development environments, tools, UI frameworks and so on. Besides, it guides on how to evaluate the developing app in all stages, whether it is only a prototype including UI and main functions or a testing app. Second, this study clearly shows that thanks to evolving technology that enable to develop hybrid apps working on multiple mobile platforms, it is possible to develop an educational mobile application which has a high usability including effectiveness, efficiency and satisfaction and which copes with the heterogeneity problem of classrooms, implying students having different mobile devices from different OSs such as Android and IOS, in a short time. The researcher who was the developer as well in this study had no knowledge on mobile app development even if he had experience of web technologies including HTML, CSS and JavaScript. However, he has learned how to transform his web programming knowledge into mobile app development and developed “Evaluation Matrix” app having high usability and available in Android as well as possible to make it available in IOS during this research. The advantages of hybrid apps are not only that they are available in multiple mobile services including IOS, Android and Windows

Phone, but also that they provide high performance and high-quality UI as much as native apps.

In conclusion, the fact that the usability questionnaire results showing students want to use mobile applications more for educational purposes implies that the stakeholders should benefit from this tendency and develop mobile applications for educational purposes. Hopefully, this research will be helpful for researchers, instructors, educators and developers who intend to develop a mobile application having high usability for educational purposes. Also, the case study documented in this study can be useful especially for the people who have expertise on HTML, CSS and JavaScript to transfer their skills into mobile app development area.

### 2.2.Limitations

The most crucial limitation of this study is that the researcher is not only the researcher but also the developer who develops "Evaluation Matrix" app by applying the proposed model for educational app development. Therefore, there might be a sort of bias when testing that model. In order to avoid this bias as much as possible, the researcher applied objective testing methods. For expert reviews, experts mostly themselves conducted the app evaluation according to predetermined criteria that are heuristics and severity table, which are proposed by other researchers and which are one of the most used evaluation methods in evaluating software. Besides, he also coded and analyzed user testing results according to those heuristics as much as possible. In addition, the results of usability questionnaire conducted in the classroom implementation prove to what extend the proposed model for educational app development is powerful.

Another weakness of this study is related to the requirement analysis step. The requirement analysis for “Evaluation Matrix” app was conducted very easily because there exists an actual Evaluation Matrix tool in the form of paper and web page and the literature provides a vast information for this. However, requirement analysis is not always easy as much as the conducted one in this study due to the fact that educational mobile apps are can be developed on the basis of new ideas for the particular learning activities requiring methods or strategies for conducting requirement analysis. In this regard, this study does not answer which kind of methods or strategies should be used for requirement analysis.

The fact that the model proposed in this study does not provide a guide for learning content development for the intended app is another weakness of this study. Learning content development for mobile devices is different from the normal one due to their small screen size. Despite learning content development is a crucial task for any educational mobile app having learning content, the model proposed in this study does not provide any step or guide for this. This is because this study focused on educational mobile app development in a general perspective, rather than paying attention to the development of a specific type of an educational mobile app.

The final limitation of this study is associated with the classroom implementation of “Evaluation Matrix” app. Among 16 students, the number of Android device owner was only 5. For this reason, the students formed a group in such a way that each group has one member having an Android device. Therefore, it might be said that not all students exposed perfectly to “Evaluation Matrix” app even though they performed class activities using

“Evaluation Matrix” app together with their groups. This might have caused to inaccurate evaluation of “Evaluation Matrix” app by those students.

### 2.3.Future Works

The first suggestion for future works is to conduct studies testing the model proposed in this study by developing educational mobile apps via that model. Thus, this model will be validated and/or revised. Second is to adopt this model for big-scoped apps having various variations such as complex database systems. This model might not be suitable for big-scoped educational apps.

There is also a need for studies analyzing the platforms used for developing hybrid apps due to the fact that the daily evolving technology not only produces hardware products but also leads to software improvements, which can provide various new alternative solutions in developing educational mobile apps. The researcher should catch up with this daily evolving technology by conducting continuous studies on this field. Thus, learning and learning environment will be promoted through these studies.

With respect to “Evaluation Matrix” app, there is a need to enable it for IOS devices. Besides, it should be revised on the basis of the feedbacks derived from user testings and expert reviews so that it will be much effective, efficient and satisfactory app. Since this study had a time limitation, an iterative process could not be performed adequately, but it is strongly recommended for future studies.

# REFERENCES

Abrahamsson, P., Warsta, J., Siponen, M. T., & Ronkainen, J. (2003, May). New directions on agile methods: a comparative analysis. In *Proceedings of 25th International Conference on Software Engineering* (pp. 244-254).

Abras, C., Maloney-Krichmar, D., & Preece, J. (2004). User-centered design. *Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications, 37*(4), 445-456.

Al-Harrasi, H., Al-Khanjari, Z., & Sarrab, M. (2015). Proposing a New Design Approach for M-learning Applications. *International Journal of Software Engineering and Its Applications*, *9*(11), 11-24.

Allen, S., Graupera, V., & Lundrigan, L. (2010). *Pro smartphone cross-platform development: iPhone, blackberry, windows mobile and android development and distribution*. Apress.

Alnuaim, A. (2015). *Designing and evaluating a contextual mobile learning application to support situated learning*. PhD Thesis, the University of the West of England, UK.

Alreck, P.L., Settle, R.B., 1994. *The Survey Research Handbook*. Irwin Professional Publishing, Chicago, IL.

Anderson, M., & Caumont, A. (2015). Pew Research

Center. *Technology Device Ownership: 2015*, *29*.

Andrade, P., & Albuquerque, A. (2015). Cross platform app. *International Journal of Computer Science & Information Technology, 7*(1), 33–40.

Appiah, F., Hayfron-Acquah, J. B., Panford, J. K., & Twum, F. (2015). A Tool Selection Framework for Cross Platform Mobile App Development. *International Journal of Computer Applications*, *123*(2).

Arnowitz, J., Arent, M., & Berger, N. (2010). *Effective prototyping for software makers*. Elsevier.

Ayobami, A., S., & Sheikh, O., Wan, R. (2013). Functional Requirements of Mobile Applications for Fishermen. *UACEE Second International Conference on Advances in Automation and Robotics – AAR'13*. Available at SSRN: https://ssrn.com/abstract=2261638

Ayoma, M. (2012). *M-learning support services for corporate learning case for a Kenyan Corporate Institution*. PhD Thesis, University of Nairobi, Kenya.

Bangor, A., Kortum, P., & Miller, J. (2009). Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of usability studies*, *4*(3), 114-123.

Bank, C. (2014). *Mobile UI Design Patterns*. eBook. Retrieved from http://studio.uxpin.com/ebooks/mobiledesign-patterns/.

Berking, P., Birtwhistle, M., Gallagher, S., & Haag, J. (2013). The Motif Project: Mobile Learning Survey Report. *Advanced Distributed*

*Learning*.

Berking, P., Haag, J., Archibald, T., & Birtwhistle, M. (2012). Mobile learning: Not just another delivery method. In *Proceedings of the 2012 Interservice/Industry Training, Simulation, and Education Conference*.

Blanco, A., S. (2016). *Development of Hybrid Mobile Apps Using Ionic framework*. Bachelor's Thesis, Mikkeli University, Finland.

Bias, R. G. (1994, June). The pluralistic usability walkthrough: coordinated empathies. In *Usability inspection methods* (pp. 63-76). John Wiley & Sons, Inc.

Brooke, J. (1996). SUS-A quick and dirty usability scale. *Usability evaluation in industry*, *189*(194), 4-7.

Brooke, J. (2013). SUS: a retrospective. *Journal of usability studies, 8*(2), 29-40.

Brown, J. S., Collins, A., & Duguid, P. (1989). Situated cognition and the culture of learning. *Educational Researcher, 18*, 32-42.

Brown, T.H. (2005). Beyond constructivism: Exploring future learning paradigms. *Education Today*, 2005(2).

Bruner, J. (1966). *Toward a Theory of Instruction*. Cambridge, MA: Harvard University Press.

Chen, B., Seilhamer, R., Bennett, L., & Bauer, S. (2015). Students' mobile learning practices in higher education: A multi-year study. *EDUCAUSE Review July/August*.

Chin, J. P., Diehl, V. A., & Norman, K. L. (1988, May). Development of an instrument measuring user satisfaction of the human-computer interface. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 213-218). ACM.

Creative Education Foundation. (2014). *Creative Problem Solving Resource Guide*. Creative Problem Solving Institute, Buffalo, NY.

Dalmasso, I., Datta, S. K., Bonnet, C., & Nikaein, N. Survey, comparison and evaluation of cross platform mobile application development tools. In *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 323-328.

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly, 13*(3), 319-340.

de Andrade, P. R., Albuquerque, A. B., Frota, O. F., Silveira, R. V., & da Silva, F. A. (2015). Virtualizing Attendance and Reducing Impacts on the Environment with a Mobile Application. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, *9*(8), 2035-2040.

Dennen, V. P., & Hao, S. (2014). Intentionally mobile pedagogy: The M-COPE framework for mobile learning in higher education. *Technology, Pedagogy and Education*, *23*(3), 397-419.

Divya, K., Kumar, S. V. K. (2016). Comparative Analysis of Smart Phone Operating Systems Android, Apple iOS And Windows. *International Journal of Scientific Engineering and Applied Science 2*(2).

Doubleday, A., Ryan, M., Springett, M., & Sutcliffe, A. (1997). A comparison of usability techniques for evaluating design. In *Designing interactive Systems (DIS '97) Conference* (pp. 101–110). New York: ACM.

Dumas, J. S., & Redish, J. (1999). *A practical guide to usability testing*. Intellect books.

Dyli, J. (2016). *An app for land administration: criteria, functional requirements and a prototype in Ethiopia*. Master's Thesis, University of Twente, Netherlands.

El-Kassas, W. S., Abdullah, B. A., Yousef, A. H., & Wahba, A. M. (2015). Taxonomy of Cross-Platform Mobile Applications Development Approaches. *Ain Shams Engineering Journal,* Available from http://dx.doi.org/10.1016/j.asej.2015.08.004.

Folmer, E., & Bosch, J. (2004). Architecting for usability: a survey. *Journal of systems and software*, *70*(1), 61-78.

Gifford, M. (2012). *PhoneGap mobile application development cookbook*. Packt Publishing Ltd.

Gong, J., & Tarasewich, P. (2004). Guidelines for handheld mobile device interface design. In *Proceedings of DSI 2004 Annual Meeting* (pp. 3751-3756).

Fetaji, B., Ebibi, M., & Fetaji, M. (2011). Assessing effectiveness in mobile learning by devising MLUAT (Mobile Learning Usability Attribute Testing) methodology. *International Journal of Computers and Communications*, *5*(3), 178-187.

Fischer, G. (2007). Designing socio-technical environments in support of meta-design and social creativity. In *Proceedings of the 8th international conference on Computer supported collaborative learning* (pp. 2-11). International Society of the Learning Sciences.

Flora, H. K., & Chande, S. V. (2013). A Review And Anaysis On Mobile Application Development Processes Using Agile Methodologies. *International Journal of Research in Computer Science*, *3*(4), 9.

Flora, H. K., Chande, S. V., & Wang, X. (2014). Adopting an agile approach for the development of mobile applications. *International Journal of Computer Applications*, *94*(17).

Georgsson, M., & Staggers, N. (2016). An evaluation of patients' experienced usability of a diabetes mHealth system using a multi-method approach. *Journal of biomedical informatics*, *59*, 115-129.

Ghatol, R., & Patel, Y. (2012). *Beginning PhoneGap: Mobile Web Framework for JavaScript and HTML5.* New York: Apress Media.

Good, T. L., Brophy, J. E. (1990). *Educational psychology: A realistic approach.* (4th ed.).White Plains, NY: Longman.

Gordon, W. J. (1961). *Synectics*. Harper and Row Publishers.

Gupta, S. G., Chanekar, P. A., Lodhe, J. L., Parekar, P. P. (2016) Comparative Study of Online App Designing Tools. *International Journal of Research In Science & Engineering,* Special Issue: Techno-Xtreme 16: 640-646.

Harrison, R., Flood, D., & Duce, D. (2013). Usability of mobile applications: literature review and rationale for a new usability model. *Journal of Interaction Science*, *1*(1), 1.

Hartson, H. R., Castillo, J. C., Kelso, J., & Neale, W. C. (1996, April). Remote evaluation: the network as an extension of the usability laboratory. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 228-235). ACM.

Higgins, J. M. (2006). *101 creative problem solving techniques: The handbook of new ideas for business*. New Management Publishing Company.

Hoehle, H., & Venkatesh, V. (2015). Mobile application usability: conceptualization and instrument development. *Mis Quarterly, 39*(2), 435-472.

Houde, S., & Hill, C. (1997). What do prototypes prototype. *Handbook of human-computer interaction*, *2*, 367-381.

Hsu, Y. C., & Ching, Y. H. (2013). Mobile app design for teaching and learning: Educators' experiences in an online graduate course. *The International Review of Research in Open and Distributed Learning, 14*(4), 117-139.

Hsu, Y. C., & Ching, Y. H. (2012). Mobile microblogging: Using Twitter and mobile devices in an online course to promote learning in authentic contexts. *The International Review of Research in Open and Distributed Learning*, *13*(4), 211-227.

Hujainah, F., Dahlan, H., Al-Haimi, B., Hujainah, A., Al-Bashiri, H.,

& Abdulgabber, M. A. (2016). New Usability Guidelines with Implementation Ways of Mobile Learning Application based on Mobile Learning Usability Attributes. *Indian Journal of Science and Technology*, *9*(37).

Ibrahim, N. H., Ali, M., Surif, J., Samsudin, S. A., Abdullah, A. H., & Ismail, N. (2016, May). My reflection mobile app promotes critical reflective practice. In *Information and Communication Technology (ICoICT), 2016 4th International Conference on* (pp. 1-6). IEEE.

Inostroza, R., Rusu, C., Roncagliolo, S., & Rusu, V. (2013, November). Usability heuristics for touchscreen-based mobile devices: update. In *Proceedings of the 2013 Chilean Conference on Human-Computer Interaction* (pp. 24-29). ACM.

Inukollu, V. N., Keshamoni, D. D., Kang, T., & Inukollu, M. (2014). Factors influencing quality of mobile apps: Role of mobile app development life cycle. *International Journal of Software Engineering & Applications (IJSEA), 5*(5), 15-34.

Isaksen, S. G., Dorval, K. B., & Treffinger, D. J. (2011). *Creative approaches to problem solving: A framework for innovation and change*. Sage Publications.

Jeon, J. H., & Kim, K. (2016). Development of Mobile App for Self-Management Performance of Patients with CHB. *Advanced Science and Technology Letters, 129*, 229-233.

Johnson, L., Adams, S., & Cummins, M. (2013). *NMC Horizon Report: 2012 Higher Education Edition*. Austin, Texas: The New Media

Consortium.

Johnson, L., Levine, A., Smith, R., & Stone, S. (2010). *The 2010 Horizon Report*. Austin, Texas: The New Media Consortium.

Johnson, L., Levine, A., & Smith, R. (2009). *The 2009 Horizon Report*. Austin, Texas: The New Media Consortium.

Keskin, N. O., & Metcalf, D. (2011). The current perspectives, theories and practices of mobile learning. *TOJET: The Turkish Online Journal of Educational Technology*, *10*(2).

Kolodner, J. L., & Wills, L. M. (1996). Powers of observation in creative design. *Design studies*, *17*(4), 385-416.

Koneru, I. (2010). Addie: Designing web-enabled information literacy instructional modules. *Desidoc journal of library & information technology*, *30*(3), 23.

Kortum, P., & Sorber, M. (2015). Measuring the usability of mobile applications for phones and tablets. *International Journal of Human-Computer Interaction*, *31*(8), 518-529.

Koschmann, T., Kelson, A. C., Feltovich, P. J., & Barrows, H. S. (1996). Computer-supported problem-based learning: A principled approach to the use of computers in collaborative learning. In T. D. Koschmann (Ed.), *CSCL: Theory and Practice of an Emerging Paradigm, pp. 83–124. Erlbaum.*

Krishnan, V., & Ulrich, K. T. (2001). Product development decisions: A review of the literature. *Management science*, *47*(1), 1-21.

Kukulska-Hulme, A., & Traxler, J. (2005). Mobile teaching and learning. In A. Kukulska-Hulme & J. Traxler (Eds.), *Mobile learning: A handbook for educators and trainers* (pp. 25–44). London/New York: Routledge.

Kumar, N. A., Krishna, K. H., & Manjula, R. (2016). Challenges and Best Practices in Mobile Application Development. *Imperial Journal of Interdisciplinary Research*, *2*(12).

Larson, L. C., & Miller, T. N. (2011). 21st century skills: Prepare students for the future. *Kappa Delta Pi Record*, *47*(3), 121-123.

Lim C., Hong M., Seo S., Lee C. (2016). A Study on the Application and Effects of Smart Support System for Creativity in Engineering Education. *Korean Society for Engineering Education, 19* (2), 34-44.

Lim, C., Kim, S., Han, H., Seo, S. (2014). Application of Smart Support System for Creative Problem Solving: Case Study of Art and Design Courses. *Asian journal of education 15*(3), 171-201.

Lim, C., Lim, W., & Hong, M. (2013). A Developmental Study for the Design of the Creative Problem Solving Support System . In R. Luckin, S. Puntambekar, P. Goodyear, B. L. Grabowski, J. Underwood, & N. Winters (Eds.), *Handbook of the Design in Educational Technology* (pp. 217-229). New York: Routledge.

Lim, C., Lee, J., & Lee, S. (2014). A theoretical framework for integrating creativity development into curriculum: the case of a Korean engineering school. *Asia Pacific Education Review*, *15*(3), 427-442.

Lim, C., Yoon, S., Park, K., & Hong, M. (2009). A study of an integrated instructional model in a collegiate course for creative problem solving with online support system. *Korean Journal of Educational Technology*, *25*(1), 171-203.

Litayem, N., Dhupia, B., & Rubab, S. (2015). Review of Cross-Platforms for Mobile Learning Application Development. *International Journal of Advanced Computer Science and Applications*, *6*(1).

Lund, A. M. (2001). Measuring Usability with the USE Questionnaire. *Usability interface*, *8*(2), 3-6.

McQuiggan, S., McQuiggan, J., Sabourin, J., & Kosturko, L. (2015). *Mobile learning: A handbook for developers, educators, and learners*. John Wiley & Sons.

Mendoza, A. (2013). *Mobile user experience: patterns to make sense of it all*. Newnes.

Naismith, L., Lonsdale, P., Vavoula, G., and Sharples, M. (2004). *Literature Review in Mobile Technologies and Learning*. NESTA (National Endowment for Science Technology and the Arts), Bristol, UK.

Nayebi, F., Desharnais, J. M., & Abran, A. (2012). The state of the art of mobile application usability evaluation. In *Electrical & Computer Engineering (CCECE), 2012 25th IEEE Canadian Conference on* (pp. 1-4). IEEE.

Noller, R. B., & Mauthe, E. (1977). *Scratching the surface of creative problem-solving: a bird's eye-view of CPS*. New York: Dok Pub.

Nordin, N., Embi, M. A., & Yunus, M. M. (2010). Mobile learning framework for lifelong learning. *Procedia-Social and Behavioral Sciences*, *7*, 130-138.

Nielsen, J. (1993). *Usability engineering*. Elsevier.

Nielsen, J. (1994). Usability inspection methods. In *Conference companion on Human factors in computing systems* (pp. 413-414). ACM.

Nielsen, J., & Landauer, T. K. (1993). A mathematical model of the finding of usability problems. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems* (pp. 206-213). ACM.

Nuseibeh, B., & Easterbrook, S. (2000). Requirements engineering: a roadmap. In *Proceedings of the Conference on the Future of Software Engineering* (pp. 35-46). ACM.

Oberer, B. (2016). 'Flipped MIS'. The Mobile Flipped Classroom Approach Shown In The Example Of MIS Courses. *International Journal of u-and e-Service, Science and Technology*, *9*(3), 379-390.

Orjuela, M. A. V., Uribe-Quevedo, A., Jaimes, N., & Perez-Gutierrez, B. (2015, October). External automatic defibrillator game-based learning app. In *Games Entertainment Media Conference (GEM), 2015 IEEE* (pp. 1-4). IEEE.

Osborn, A. F. (1953). Applied imagination: *Principles and procedures of creative thinking*. New York: Charles Scribner's Sons.

Palmieri, M., Singh, I., & Cicchetti, A. (2012, October). Comparison

of cross-platform mobile development tools. In *Intelligence in Next Generation Networks (ICIN), 2012 16th International Conference on* (pp. 179-186). IEEE.

Parsons, D., Ryu, H., & Cranshaw, M. (2007). A design requirements framework for mobile learning environments. *JCP*, *2*(4), 1-8.

Phyo, M. T. (2014). Choosing a Mobile Application Development Approach. *ASEAN Journal of Management & Innovation 1*(1), 69-74.

Pierre, N. J., Xiumei, F., Daniel, G., & Claude, G. J. (2015). Cross-platform mobile geolocation applications based on PhoneGap. *Lecture Notes on Software Engineering*, *3*(2), 78-82.

Ram, S. G. (2016). A Comparative Study of Mobile Operating Systems. *International Journal of Recent Trends in Engineering & Research 2*(2), 57-61.

Ribeiro, A., & da Silva, A. R. (2012, September). Survey on cross-platforms and languages for mobile apps. In *Quality of Information and Communications Technology (QUATIC), 2012 Eighth International Conference on the* (pp. 255-260). IEEE.

Richey, R.C. & Klein, J.D. (2007). *Design and Development Research*. New Jersey, USA: Lawrence Erlbaum Associates, Inc.

Robertson, S., & Robertson, J. (2012). *Mastering the requirements process: Getting requirements right*. Addison-Wesley.

Rowley, D. E., & Rhoades, D. G. (1992, June). The cognitive jogthrough: a fast-paced user interface evaluation procedure. In *Proceedings*

*of the SIGCHI conference on Human factors in computing systems* (pp. 389-395). ACM.

Rubin, J., 1994. Testing Two Participants at a Time. Handbook of Usability Testing. John Wiley and Sons, New York.

Sauro, J. (2011). *A practical guide to the System Usability Scale: Background, benchmarks, & best practices*. Denver, CO: Measuring Usability LLC.

Savransky, S. D. (2000). *Engineering of creativity: Introduction to TRIZ methodology of inventive problem solving*. CRC Press.

Sears, A. (1997). Heuristic walkthroughs: Finding the problems without the noise. *International Journal of Human-Computer Interaction*, *9*(3), 213-234.

Seong, D.S.-K. (2006). Usability guidelines for designing mobile learning portals. In *Proceedings of the 3rd international conference on Mobile technology, applications & systems*, October 25-27, Bangkok, Thailand. Article No. 25.

Sharples, M. (2000) The Design of Personal Mobile Technologies for Lifelong Learning. *Computers and Education, 34*, 177-193.

Sharples, M. (2002). Disruptive devices: mobile technology for conversational learning. *International Journal of Continuing Engineering Education and Life Long Learning, 12*(5/6), 504-520.

Shotts, K. (2014). *PhoneGap 3. x Mobile Application Development Hotshot*. Packt Publishing Ltd.

Shuib, M., Abdullah, A., Azizan, S. N., & Gunasegaran, T. (2015). Designing an Intelligent Mobile Learning Tool for Grammar Learning (i-MoL). *International Journal of Interactive Mobile Technologies*, *9*(1), 41-46.

Siemens, G. (2004). Connectivism: A learning theory for the digital age. *International Journal of Instructional Technology and Distance Learning*. Retrieved on March 09, 2017, from http://www.itdl.org/Journal/Jan_05/article01.htm.

Smith, P.L., & Ragan, T.J. (2005). *Instructional Design* (3rd ed.). New York: Merill

Soken, N., Reinhart, B., Vora, P., & Metz, S. (1993). *Methods for Evaluating Usability (Section 5B)*. Honeywell, Dec.

Stanton, G., & Ophoff, J. (2013). Towards a method for mobile learning design. *Issues in Informing Science & Information Technology*, *10*, 501-523.

Statista (2016a). *Number of available applications in the Google Play Store from December 2009 to September 2016.* Retrieved from https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/.

Statista (2016b). *Number of apps available in leading app stores as of June 2016*. Retrieved from https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/.

Stead, S. & Dorval, K. B. (2001). 'MASTER BLASTER' The

Power of the Evaluation Matrix. *CPSB's Communiqué, 11*, 24-27.

Sun, B., Igbiriki, R., Pere, P., & Edwards, A. (2016, November). Kizzle-an app for learning. In *Interactive Mobile Communication, Technologies and Learning (IMCL), 2016 International Conference on* (pp. 28-32). IEEE.

Terninko, J., Zusman, A., & Zlotin, B. (1998). *Systematic innovation: an introduction to TRIZ (theory of inventive problem solving)*. CRC press.

Treffinger, D. J. (2000). *Creative Problem Solver's Guidebook*. Prufrock Press Inc.

Treffinger, D. J. (2007). Creative problem solving (CPS): Powerful tools for managing change and developing talent. *Gifted and Talented International*, *22*(2), 8-18.

Treffinger, D. J., & Isaksen, S. G. (2005). Creative problem solving: The history, development, and implications for gifted education and talent development. *Gifted Child Quarterly*, *49*(4), 342-353.

Treffinger, D. J., Isaksen, S. G., & Stead-Dorval, K. B. (2006). *Creative problem solving: An introduction*. Prufrock Press Inc.

Vavoula, G. (2004). *KLeOS: A Knowledge and Learning Organisation System in Support of Lifelong Learning*. Unpublished PhD Thesis, University of Birmingham, UK.

Vora, P. R., & Helander, M. G. (1995). A teaching method as an alternative to the concurrent think-aloud method for usability testing. *Advances in Human Factors/Ergonomics*, *20*, 375-380.

Vygotsky, L.S. (1978). *Mind and society: The development of higher mental processes*. Cambridge, MA: Harvard University Press.

Vygotsky, L. S. (1987). *Collected works* (Volume 1). New York: Plenum.

Warfel, T. Z. (2009). *Prototyping: a practitioner's guide*. Rosenfeld media.

Wargo, J. M. (2012). *PhoneGap Essentials: Building Cross-platform Mobile Apps*. Addison-Wesley.

Wasserman, A. I. (2010, November). Software engineering issues for mobile application development. In *Proceedings of the FSE/SDP workshop on Future of software engineering research* (pp. 397-400). ACM.

Weiss, S. (2003). *Handheld usability*. John Wiley & Sons.

Wharton, C., Rieman, J., Lewis, C., & Polson, P. (1994). The Cognitive Walkthrough: A practitioner's guide. In J. Nielsen & R. L. Mack (Eds.), *Usability inspections methods* (pp. 105-140). New York: Wiley.

Wilson, B. G. (1995). Metaphors for Instruction: Why We Talk about Learning Environments. *Educational Technology*, *35*(5), 25-30.

Wixon, D., Jones, S., Tse, L., & Casaday, G. (1994). Inspections and design reviews: framework, history and reflection. In *Usability inspection methods* (pp. 77-103). John Wiley & Sons, Inc.

Yamazaki, K. (2009). Approach to human centered design innovation by utilized paper prototyping. In M. Kurosu (Ed.), *Human*

*Centered Design, HCII 2009, LNCS 5619*, pp. 367–373.

Yang, M. C. (2005). A study of prototypes, design activity, and design outcome. *Design Studies*, *26*(6), 649-669.

Zhang, Z., 2001. *Overview of usability evaluation methods*. Available from http://www.cs.umd.edu/~zzj/UsabilityHome.html/.

Zhang, D., & Adipat, B. (2005). Challenges, methodologies, and issues in the usability testing of mobile applications. *International journal of human-computer interaction*, *18*(3), 293-308.

Zhang, Z., Basili, V., & Shneiderman, B. (1998, October). An empirical study of perspective-based usability inspection. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 42*(19), 1346-1350.

# APPENDIXES

**Appendix 1.** The Prototype of "Evaluation Matrix" app

**Appendix 2.** Expert Evaluation Using Heuristics for the Prototype of "Evaluation Matrix" App

**Appendix 3.** Informed Consent Form for User Testing of the Prototype of "Evaluation Matrix" app

**Appendix 4.** Expert Evaluation Using Heuristics for "Evaluation Matrix" App

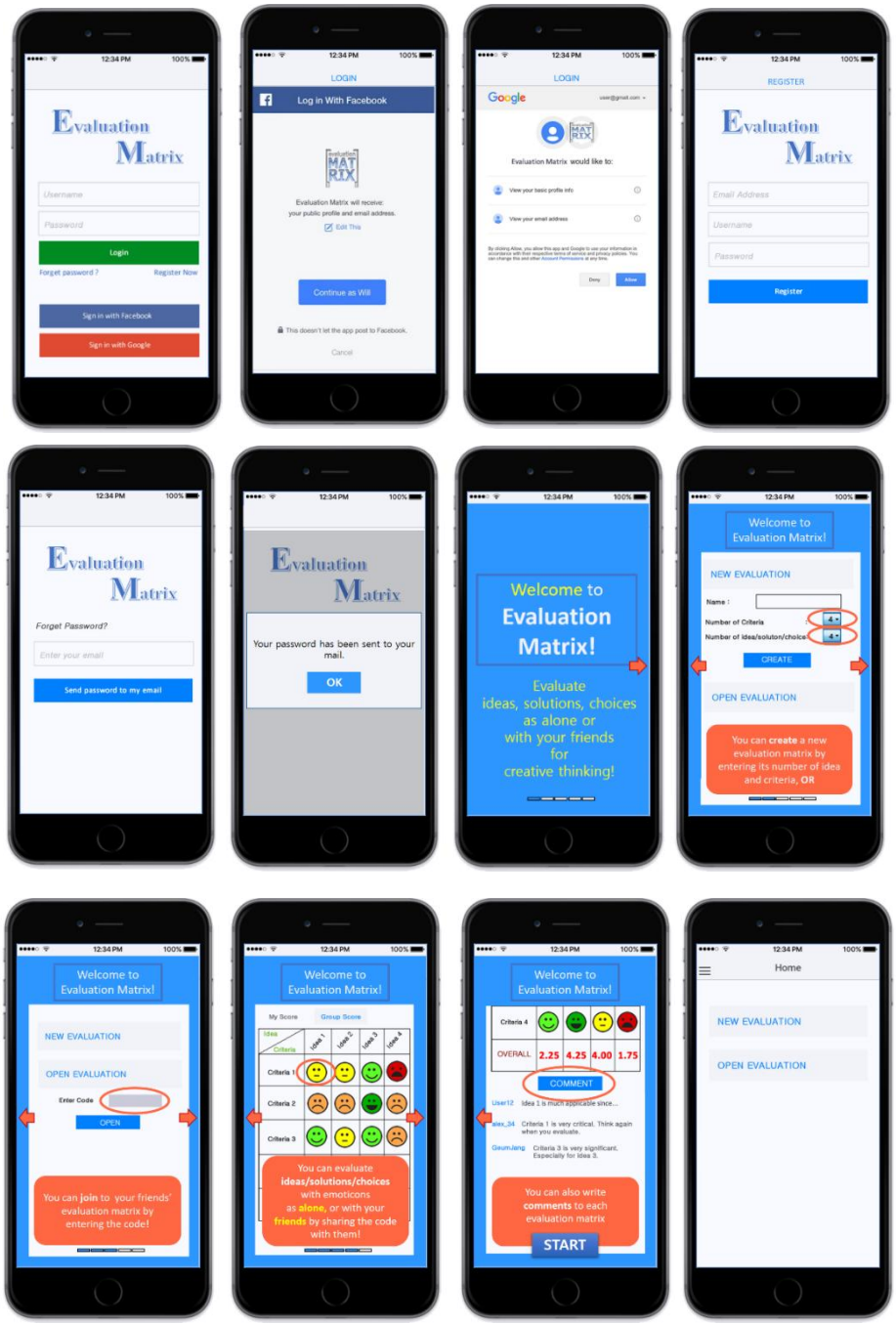
**Appendix 5.** Informed Consent Form for User Testing of "Evaluation Matrix" app

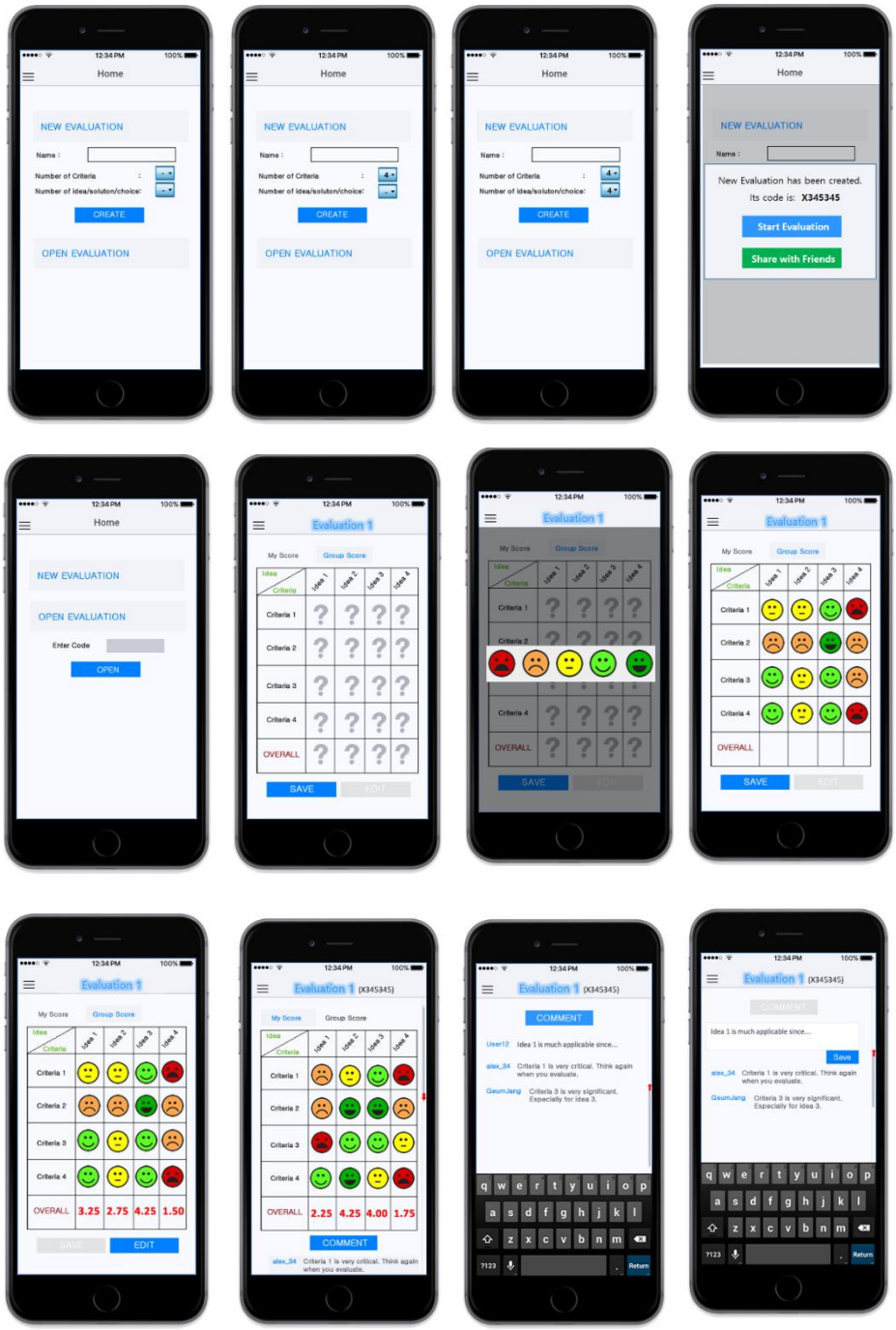
**Appendix 6.** Usage Method of "Evaluation Matrix" app in classroom environment

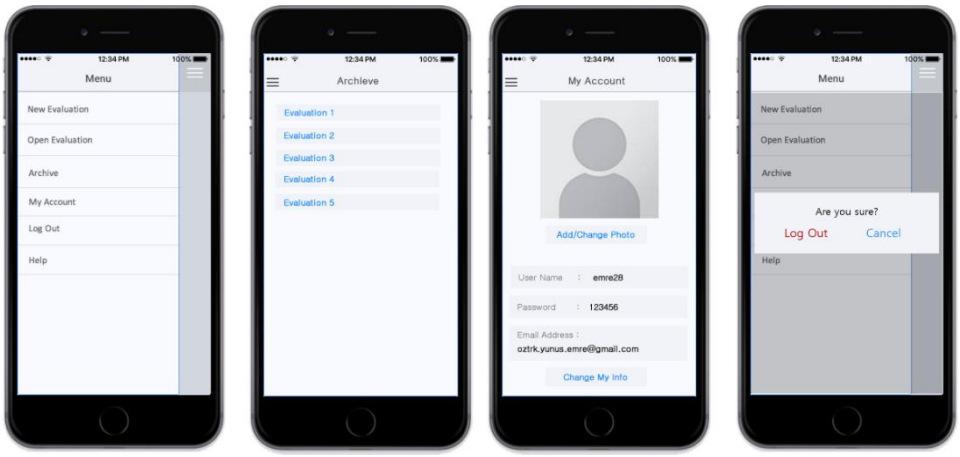
**Appendix 7.** Usability Questionnaire for "Evaluation Matrix" App

**Appendix 8.** Usability Interview

# Appendix 1. The Prototype of “Evaluation Matrix” App




Home
NEW EVALUATION
Name :
Number of Criteria :
Number of idea/soluton/choice:
CREATE
OPEN EVALUATION
New Evaluation has been created.
Its code is: X345345
Start Evaluation
Share with Friends
Enter Code
OPEN
Evaluation 1
My Score
Group Score
Idea
Criteria
Idea 1
Idea 2
Idea 3
Idea 4
Criteria 1
Criteria 2
Criteria 3
Criteria 4
OVERALL
SAVE
EDIT
3.25
2.75
4.25
1.50
2.25
4.25
4.00
1.75
Evaluation 1 (X345345)
COMMENT
User12 Idea 1 is much applicable since...
alex_34 Criteria 1 is very critical. Think again when you evaluate.
GeumJang Criteria 3 is very significant. Especially for idea 3.
Save


12:34 PM
100%
Menu
New Evaluation
Open Evaluation
Archive
My Account
Log Out
Help
12:34 PM
100%
Archieve
Evaluation 1
Evaluation 2
Evaluation 3
Evaluation 4
Evaluation 5
12:34 PM
100%
My Account
Add/Change Photo
User Name : emre28
Password : 123456
Email Address :
oztrk.yunus.emre@gmail.com
Change My Info
12:34 PM
100%
Menu
New Evaluation
Open Evaluation
Archive
Are you sure?
Log Out
Cancel
Help

# Appendix 2. Expert Evaluation Using Heuristics for the Prototype of "Evaluation Matrix" App

## Expert Evaluation Using Heuristics for "Evaluation Matrix" App

Hello, I am Yunus Emre OZTURK and a master's student in Educational Technology, Seoul National University. I am conducting a research on providing guidelines about developing a simple mobile application for educational purposes. For this purpose, I have already developed a guideline, and now I am developing a mobile app called as "Evaluation Matrix" by following this guideline. Now I am in the phase of developing a prototype for this app, and need you to evaluate the prototype of "Evaluation Matrix" app since you are an expert in the domain, app development or design. Your personal information will be kept confidential.

I sincerely appreciate you for giving your precious time and effort for this study.

Yunus Emre OZTURK

Seoul National University, Department of Education,
Educational Technology Major

**First, please enter your demographic information:**

| **Name** | **Occupation** | **Field having expertise** (domain, mobile app development or design) | **Experience (in years)** |
|---|---|---|---|
| | | | |

**Then, I would like to request you to perform the following steps, respectively:**

1- Read the "Introduction of "Evaluation Matrix" app".

2- Read carefully "Usability Heuristics" (**Table 1**), consisting of general UI principles for mobile devices.

3- Go through the interface of the app at least one time in order to get the feel for the flow of the interaction and the general scope of the system.

4- Examine and judge the prototype according to the scenario containing the tasks, which is provided in the next section.

5- During the use of the prototype, take notes about the design if needed.

6- After you completed the try-out of prototype, I fill the "Heuristic Evaluation Template" (**Table 3**) according to Heuristics (Table 1), your notes, opinions and feelings.

7- Also, express the severity of the issue/problem related to UI in that template, according to **Table 2**.

Note: You can perform this evaluation as you wish. There is no any time limitation. Whether getting printed document of the template or filling it on the computer depends on you.

The followings are the needed materials including the introduction of the designated app, heuristics for mobile devices, a scenario used for evaluation, and a template for easily recording the evaluation data.

**The Introduction of "Evaluation Matrix" app**

Evaluation Matrix is an actually existing tool used in Creative Problem Solving (CPS) systems promoting creativity, as supporting convergent thinking. With this tool, individuals list the possible ideas/solutions/options about a problem or a case on one axis of a matrix (or grid), and puts important criteria to be evaluated on to the other axis. This allows individuals to compare and contrast several ideas (e.g. solutions) according to similar criteria. Through Evaluation Matrix, the possible solutions or ideas are systematically evaluated according to the evaluation criteria. Another feature of the app is that the users are able to perform evaluation work as alone or as groups and write comments about the evaluation in the group evaluation.

**Scenario:**

Assume that there is one group of students trying to solve a problem in a project and there are several ideas proposed by group members to cope with that problem. However, the group members cannot decide to which idea they should follow because each member has different views.

In this situation, they use "Evaluation matrix" app. Each member separately evaluates those ideas according to the criteria that are the requirements an idea should have to solve the problem. The app gathers all members' evaluation results and provides the average evaluation scores in the group evaluation page. In addition, the members are able write a comment about the idea or the criteria in the

group evaluation page to affect other members' evaluation.

In short, when trying the prototype, perform the following tasks, respectively:

1. Sign in or Login to "Evaluation Matrix" app with any method,
2. Create a new evaluation by entering the number of criteria and idea.
3. Evaluate the ideas according to the criteria, by using that evaluation matrix.
4. Check the group evaluation results.
5. Write a comment in the group evaluation page, and save it.
6. Log out in the "Evaluation Matrix" app.

Lastly, you can try any other function of the prototype as you wish.

**Table 1: Usability Heuristics**

| # | Heuristic | Description | Example of good practices |
|---|---|---|---|
| #1 | Visibility of system status (visibility offeedback) | The device should keep the user informed about all the processes and state changes through feedback and in a reasonable time. | Display a progress bar when something takes a long time to load a screen |
| #2 | Match between system and the real world | The device should speak the users' language with words, phrases and concepts familiar to the user, instead of system-oriented concepts and/or technicalities. The device should follow the real world conventions and physical laws, displaying the information in a logical and natural order. | When designing for children, using terms with which they are familiar and displaying information in formats they are used to seeing. |
| #3 | User control and freedom | The device should allow the user to undo and redo his actions, and provide clearly pointed "emergency exits" to leave unwanted states. These options should be preferably through a physical button or similar. | Providing the functionality to Undo and Redo actions and to easily exit the system |
| #4 | Consistency and standards | The device should follow the established conventions, on condition that the user should be able to do things in a familiar, standard and consistent way. | Using icons with which people are familiar, rather than creating new designs that mean the same thing. |
| #5 | Error prevention | The device should hide or deactivate unavailable functionalities, warn users about critical actions and provide access to additional information. | Offering user a way to re-establish the account within a certain time period if he/she cancels his/her account, |

| | | | |
|---|---|---|---|
| #6 | Minimize the user's memory load | The device should offer visible objects, actions and options in order to prevent users to memorize information from one part of the dialogue to another. | Allowing easy access to previously entered information, such as serial numbers. |
| #7 | Customization and shortcuts | The device should provide basic and advanced configuration options, allow definition and customization of (or to provide) shortcuts to frequent actions. | Macintosh' Command+Q to quit an application |
| #8 | Efficiency of use and performance | The device should be able to load and display the required information in a reasonable time and minimize the required steps to perform a task. Animations and transitions should be displayed smoothly. | - |
| #9 | Aesthetic and minimalist design | The device should avoid displaying unwanted information in a defined context of use. | Not replicating a large number of persistent navigation options across all pages of a mobile site |
| #10 | Help users recognize, diagnose, and recover from errors | The device should display error messages in a language familiar to the user, indicating the issue in a precise way and suggesting a constructive solution. | Providing an error message, If the user enters an invalid email address. |
| #11 | Help and documentation | The device should provide easy-to-find documentation and help, centered on the user's current task and indicating concrete steps to follow. | Providing users an example of how to input the information using the required formatting, such as entering a phone number as xxx-xxx-xxxx when filling a label or textbox. |
| #12 | Physical interaction and ergonomics | The device should provide physical buttons or similar for main functionalities, located in recognizable positions by the user, which should fit the natural posture of the user's hands. | Providing notifications by the app when using the device for other purposes. |

**Table 2: To what extent the issue/problem is serious**

| Ranking | Meaning |
|---|---|
| **0** | I don't agree that this is a usability problem at all. |
| **1** | Cosmetic problem only: need not be fixed unless extra time is available on project. |
| **2** | Minor usability problem: fixing this should be given low priority. |
| **3** | Major usability problem: important to fix, so should be given high priority. |
| **4** | Usability catastrophe: imperative to fix this before product can be released. |

**Table 3: Heuristic Evaluation Template for "Evaluation Matrix" app**

| # | Issue/Problem Description | Heuristic violated (#1~#12, from Table 1) | Severity (0~4, from Table 2) | Recommendation |
|---|---|---|---|---|
| **1** | | | | |
| **2** | | | | |
| **3** | | | | |
| **4** | | | | |
| **5** | | | | |
| **6** | | | | |
| **7** | | | | |
| **8** | | | | |
| **9** | | | | |
| **10** | | | | |

\Note: *Heuristic violated* refers to which heuristic is violated by that issue; *Severity* refers to what extent the issue is serious; *Recommendation* refers to the suggestion provided by you, to handle with that issue, if possible

# Appendix 3. Informed Consent Form for User Testing of the Prototype of "Evaluation Matrix" app

## Informed Consent Form

I agree to participate to the study conducted by Yunus Emre OZTURK at Seoul National University, Department of Education.

The purpose of this study is to evaluate the design of "Evaluation Matrix" app via the prototype developed with Microsoft PowerPoint.

The procedures involve the monitored use of the prototype. I will be asked to perform specific tasks using the prototype. In addition, I will be asked open-ended questions about the prototype while I experience it.

I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

All information collected in the study is confidential, and my name will not be exposed at any time.

Please sign below to indicate that you have read and understand the information on this form and that any questions you might have about the session have been answered.

Date: _____

Name: _____

Signature: _____

Thank you!

I sincerely appreciate you for giving your invaluable time and effort for this study.

Yunus Emre OZTURK

Department of Education, Seoul National University

**Introduction of "Evaluation Matrix" App**

With Evaluation Matrix app, users list the possible ideas/solutions/options about a problem or a case on one axis of a matrix (or grid), and puts important criteria to be evaluated on to the other axis. This allows them to compare and contrast several ideas (e.g. solutions) according to similar criteria. Through Evaluation Matrix, the possible solutions or ideas are systematically evaluated according to the evaluation criteria. Another feature of the app is that the users are able to perform evaluation work as alone or as groups and write comments about the evaluation in the group evaluation.

**The goal of Usability Testing**

The goal of testing is to find missing features and problem areas with the user interface of "Evaluation Matrix" App via the prototype. This is not a contest and it is important to emphasize that it is not you that is being evaluated, but the interface and how its functions are.

**Tasks for Usability Testing**

The followings are the tasks for you to complete. As you complete the tasks, please tell us what you are thinking, doing and looking for. Besides, I will ask open-ended questions about the design, such as the place of buttons, the color of textboxes, etc. and about the difficulties you experience while using the prototype. They will produce very significant data that will give me the basis for understanding and evaluating the user interface.

**Tasks:**

1. Use the prototype for a few minutes to become acquainted with the app.
2. Sign in or Login to "Evaluation Matrix" app with any method,
3. Create a new evaluation by entering the number of criteria and idea.
4. Evaluate the ideas according to the criteria, by using that evaluation matrix.
5. Check the group evaluation results.
6. Write a comment in the group evaluation page, and save it.
7. Log out in the "Evaluation Matrix" app.
8. Lastly, you can try any other function of the prototype as you wish.

# Appendix 4. Expert Evaluation Using Heuristics for "Evaluation Matrix" App

## Expert Evaluation Using Heuristics for "Evaluation Matrix" App

Hello, I am Yunus Emre OZTURK and a master's student in Educational Technology, Seoul National University. I am conducting a research on providing guidelines about developing a simple mobile application for educational purposes. For this purpose, I have already developed a guideline, and now I am developing a mobile app called as "Evaluation Matrix" by following this guideline. Now I have developed it, and need you to examine this app since you are an expert in the domain, app development or design. Your personal information will be kept confidential.

I sincerely appreciate you for giving your precious time and effort for this study.

Yunus Emre OZTURK

Seoul National University, Department of Education,
Educational Technology Major

**First, please enter your demographic information:**

| **Name** | **Occupation** | **Field having expertise** (domain, app/software development or design) | **Experience (in years)** | **Brand and Model of your smartphone** (Ex: Samsung, Note2) |
|---|---|---|---|---|
| | | | | |

**Then, I would like to request you to perform the following steps, respectively:**

1- Read the "Introduction of "Evaluation Matrix" app".

2- Read carefully "Usability Heuristics" (**Table 1**), consisting of general UI principles for mobile devices.

3- Go through the interface of the app at least one time in order to get the feel for the flow of the interaction and the general scope of the system.

4- Examine and judge the app according to the scenario containing the tasks, which is provided in the next section.

5- During the use of the app, take notes about the design if needed.

6- After you completed the try-out of the app, please fill the "Heuristic Evaluation Template" (**Table 3**) according to Heuristics (Table 1), your notes, opinions and feelings.

7- Also, express the severity of the issue/problem related to UI in that template, according to **Table 2**.

Note: You can perform this usability test as you wish. There is no any time limitation. Whether getting printed document of the template or filling it on the computer depends on you.

The followings are the needed materials including the introduction of the app, heuristics for mobile devices, a scenario used for evaluation, and a template for easily recording the evaluation data.

**The Introduction of "Evaluation Matrix" app**

Evaluation Matrix is an actually existing tool used in Creative Problem Solving (CPS) systems promoting creativity, as supporting convergent thinking. With this tool, individuals list the possible ideas/solutions/options about a problem or a case on one axis of a matrix (or grid), and puts important criteria to be evaluated on to the other axis. This allows individuals to compare and contrast several ideas (e.g. solutions) according to similar criteria. Through Evaluation Matrix, the possible solutions or ideas are systematically evaluated according to the evaluation criteria. Another feature of the app is that the users are able to perform evaluation work as alone or as groups and write comments about the evaluation in the group evaluation. Lastly, it is possible to share the final score of the group evaluation in the form of screenshot.

**Scenario:**

Assume that there is one group of students trying to solve a problem of a project and there are several ideas proposed by group members to cope with that problem. However, the group members cannot decide to which idea they should follow because each member has different views.

In this situation, they use “Evaluation matrix” app. Each member separately evaluates those ideas according to the criteria that are the requirements an idea should have to solve the problem. The app gathers all members’ evaluation results and provides the average evaluation scores in the group evaluation page. In addition, the members are able write a comment about the idea or the criteria in the group evaluation page to persuade other members’ ratings.

In short, when trying the prototype, perform the following tasks, respectively:

**Part 1:**

1. Sign in to “Evaluation Matrix” app and use it for a few minutes to become acquainted with the app.
2. Create a new evaluation by entering anything to the required places.
3. Calculate your rating in this newly created evaluation matrix and save it.
4. Send the code of the evaluation to anyone.

**Part 2:**

5. Click the “Join in Evaluation” button.
6. Enter the code “2475” to join in an evaluation.
7. Calculate your rating in the evaluation matrix named “BuyHouse”.
8. Check the overall group evaluation result of “BuyHouse”.
9. Share the overall group result of “BuyHouse” with anyone.
10. Check the comments about “BuyHouse” evaluation matrix and make a comment or send a picture.
11. Try any other function of the app as you wish.
12. Log out in the “Evaluation Matrix” app.

**Table 1: Usability Heuristics**

| # | Heuristic | Description | Example of good practices |
|---|---|---|---|
| #1 | Visibility of system status (visibility of feedback) | The device should keep the user informed about all the processes and state changes through feedback and in a reasonable time. | Display a progress bar when something takes a long time to load a screen |
| #2 | Match between system and the real world | The device should speak the users' language with words, phrases and concepts familiar to the user, instead of system-oriented concepts and/or technicalities. The device should follow the real world conventions and physical laws, displaying the | When designing for children, using terms with which they are familiar and displaying information in formats they are used to seeing. |

| | | | |
|---|---|---|---|
| | | information in a logical and natural order. | |
| #3 | User control and freedom | The device should allow the user to undo and redo his actions, and provide clearly pointed "emergency exits" to leave unwanted states. These options should be preferably through a physical button or similar. | Providing the functionality to Undo and Redo actions and to easily exit the system |
| #4 | Consistency and standards | The device should follow the established conventions, on condition that the user should be able to do things in a familiar, standard and consistent way. | Using icons with which people are familiar, rather than creating new designs that mean the same thing. |
| #5 | Error prevention | The device should hide or deactivate unavailable functionalities, warn users about critical actions and provide access to additional information. | Offering user a way to re-establish the account within a certain time period if he/she cancels his/her account, |
| #6 | Minimize the user's memory load | The device should offer visible objects, actions and options in order to prevent users to memorize information from one part of the dialogue to another. | Allowing easy access to previously entered information, such as serial numbers. |
| #7 | Customization and shortcuts | The device should provide basic and advanced configuration options, allow definition and customization of (or to provide) shortcuts to frequent actions. | Macintosh' Command+Q to quit an application |
| #8 | Efficiency of use and performance | The device should be able to load and display the required information in a reasonable time and minimize the required steps to perform a task. Animations and transitions should be displayed smoothly. | - |
| #9 | Aesthetic and minimalist design | The device should avoid displaying unwanted information in a defined context of use. | Not replicating a large number of persistent navigation options across all pages of a mobile site |
| #10 | Help users recognize, diagnose, and recover from errors | The device should display error messages in a language familiar to the user, indicating the issue in a precise way and suggesting a constructive solution. | Providing an error message, If the user enters an invalid email address. |
| #11 | Help and documentation | The device should provide easy-to-find documentation and help, centered on the user's current task and indicating concrete steps to follow. | Providing users an example of how to input the information using the required formatting, such as entering a phone number as xxx-xxx-xxxx when filling a label or textbox. |
| #12 | Physical interaction and ergonomics | The device should provide physical buttons or similar for main functionalities, located in recognizable positions by the user, which should fit the natural posture of the user's hands. | Providing notifications by the app when using the device for other purposes. |

**Table 2: To what extent the issue/problem is serious**

| Ranking | Meaning |
|---|---|
| 0 | I don't agree that this is a usability problem at all. |
| 1 | Cosmetic problem only: need not be fixed unless extra time is available on project. |
| 2 | Minor usability problem: fixing this should be given low priority. |
| 3 | Major usability problem: important to fix, so should be given high priority. |
| 4 | Usability catastrophe: imperative to fix this before product can be released. |

**Table 3: Heuristic Evaluation Template for "Evaluation Matrix" app**

| # | Issue/Problem Description | Heuristic violated (#1~#12, from Table 1) | Severity (0~4, from Table 2) | Recommendation |
|---|---|---|---|---|
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |
| 5 | | | | |
| 6 | | | | |
| 7 | | | | |
| 8 | | | | |
| 9 | | | | |
| 10 | | | | |

Note: *Heuristic violated* refers to which heuristic is violated by that issue; *Severity* refers to what extent the issue is serious; *Recommendation* refers to the suggestion provided by you, to handle with that issue, if possible.

# Appendix 5. Informed Consent Form for User Testing of "Evaluation Matrix" app

**Informed Consent Form**

I agree to participate to the study conducted by Yunus Emre OZTURK at Seoul National University, Department of Education.

The purpose of this study is to evaluate the usability of "Evaluation Matrix" app.

The procedure involves the monitored use of the "Evaluation Matrix" app. I will be asked to download "Evaluation Matrix" app in my phone and perform specific tasks using the "Evaluation Matrix" app while doing think-aloud.

I understand that participation in this usability study is voluntary and I agree to raise immediately any concerns or areas of discomfort during the session with the study administrator.

All information collected in the study is confidential, and my name will not be exposed at any time.

Please sign below to indicate that you have read and understand the information on this form and that any questions you might have about the session have been answered.

Date: _____

Name: _____

Brand of your phone: _____ Model of your phone: _____

Signature: _____

Thank you!

I sincerely appreciate you for giving your invaluable time and effort for this study.

Yunus Emre OZTURK

Department of Education, Seoul National University

**Introduction of "Evaluation Matrix" App**

With Evaluation Matrix app, users list the possible ideas/solutions/options about a problem or a case on one axis of a matrix (or grid), and puts important criteria to be evaluated on to the other axis. This allows them to compare and contrast several ideas (e.g. solutions) according to similar criteria. Through Evaluation Matrix, the possible solutions or ideas are systematically evaluated according to the evaluation criteria. Another feature of the app is that the users are able to perform evaluation work as alone or as groups and write comments about the evaluation in the group evaluation.

Lastly, it is possible to share the final score of the group evaluation in the form of screenshot.

**The goal of Usability Testing**

The goal of testing is to find missing features and problem areas with the user interface of "Evaluation Matrix" App via the prototype. This is not a contest and it is important to emphasize that it is not you that is being evaluated, but the interface and how its functions are.

**Tasks for Usability Testing**

The followings are the tasks for you to complete. As you complete the tasks, please tell us what you are thinking, doing and looking for. Besides, I will ask open-ended questions about the design, such as the place of buttons, the color of textboxes, etc. and about the difficulties you experience while using the prototype. They will produce very significant data that will give me the basis for understanding and evaluating the user interface.

**Tasks:**

**Part 1:**

1. Sign in to "Evaluation Matrix" app and use it for a few minutes to become acquainted with the app.
2. Create a new evaluation by entering anything to the required places.
3. Calculate your rating in this newly created evaluation matrix and save it.
4. Send the code of this evaluation matrix to anyone.

**Part 2:**

5. Click the "Join in Evaluation" button.
6. Enter the code "2475" to join in an evaluation.

7. Calculate your rating in the evaluation matrix named “BuyHouse”.
8. Check the overall group evaluation result of “BuyHouse”.
9. Share the overall group result of “BuyHouse” with anyone.
10. Check the comments about “BuyHouse” evaluation matrix and make a comment or send a picture.
11. Try any other function of the app as you wish.
12. Log out in the “Evaluation Matrix” app.

# Appendix 6. Usage Method of “Evaluation Matrix” app in classroom environment

## 학교환경에서 “Evaluation Matrix” 앱의 활용방법

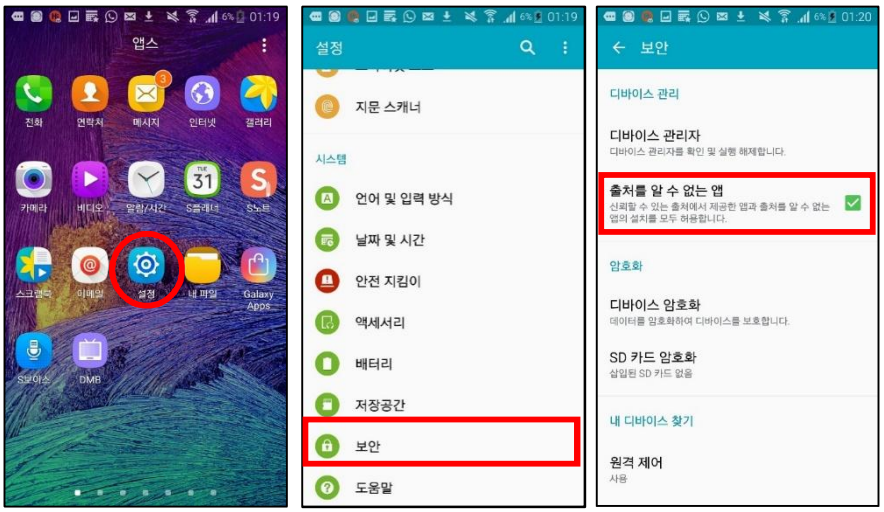
### “Evaluation Matrix” 앱이란?

창의성을 증가시키는 Creative Problem Solving (CPS) 시스템이 활용된 창의성교육에서 사용된 도구중에 하나가 “Evaluation Matrix”라는 도구이다. “Evaluation Matrix” 앱은 이 도구의 앱 버전이다. 이 앱은 수렴적 사고를 지원한다. 이 앱을 통해 사용자가 어떤 문제나 케이스에 대한 가능한 아이디어, 해결안 혹은 옵션을 어떤 행렬(메트릭스)의 한 축에 리스트한다. 다음에는 이들을 평가하기 위해서 중요다고 생각하는 기준들을 다른 축에 놓는다. 그러므로 사용자가 여러가지 아이디어 (혹은 옵션이나 해결안)를 그 기준들에 따라 평가한다. “Evaluation Matrix” 앱을 통해 가능한 해결안 혹은 아이디어가 평가기준에 따라 평가됨으로써 그 해결안이나 아이디어가 체계적으로 평가된다. 이 앱의 또 다른 기능은 각각의 평가활동이 고유 코드를 가지며 다른 사람들과 함께 평가하기 위해서 이 고유 코드를 SNS를 통해 공유할 수 있다. 마지막으로 그룹으로 진행된 평가활동의 결과도 SNS를 통해 공유할 수 있다. 요약하면 “Evaluation Matrix” 라는 앱은 여러 가지 아이디어를 평가 기준에 따라 각각의 강점과 약점을 파악하여 선택하는데 사용되는 앱이다.

### “Evaluation Matrix” 앱의 설치 방법:

- “Evaluation Matrix” 앱은 Android 디바이서에서만 가능하다.
- 설치하기 위해서 먼저 자기 핸드폰에서 다음과 같은 configuration를 해주세요:

설정 → 시스템 → 디바이스 관리 → “출처를 알 수 없는 앱” 라는 옵션을 표시해주세요. (다음과 같은 사진을 참고해주세요.)




- 그 다음에는 다음과 같은 링크를 통해 “Evaluation Matrix” 앱을 다운받아서 설치해주세요:

  https://goo.gl/gXVHMU

User Scenario:

1- 교실 안에 있는 모든 학습자들은 한 그룹이 된다. “Android” 사용자가 아닌 학습자는 “Android” 디바이스를 가진 학습자와 함께 평가활동을 진행한다.

2- 학습자들은 앱을 열어서 새로운 계정을 만든다.

3- 학습자들은 앱에 익숙해지기 위해서 2~3 분 동안 앱을 해본다.

4- 한 명 학생은 그룹 팀장이 된다.

5- 팀장은 “Create Evaluation” 버튼을 클릭하여 이 전에 있었던 학습활동에서 도출된 데이터를 통해 새로운 평가표를 만든다. 먼저 “Evaluation Name” 라는 부분에 평가활동의 이름을 입력한다. 다음에 “Number of Idea” 부분에서 아이디어, 옵션 혹은 해결안의 수를 선택한 후에 그들의 이름을 새로 생긴 텍스트상자에 입력한다. 똑 같은 것을 “Number of Criteria” 부분을 위해서도 한다. 그 다음에는 “Create” 버튼을 클릭함

으로서 새로운 평가표가 구성된다. 기준에 따라서 여러 아이디어를 평가한 다음에 “Calculate”를 글릭하여 자기 평가결과를 확인한다. 그 다음에는 “Save” 버튼을 클릭하여 평가표가 저장된다. 마지막으로 공유하기 버튼을 클릭해서 다른 학습자들과 평가표의 코드를 SNS를 통해 공유한다.

6- 학습자들은 메인 화면에서 “Join in Evaluation” 버튼을 클릭하여 “Evaluation Code”에 팀장이 제공하는 코드를 입력해서 “Join” 버튼을 클릭한다.

7- 열린 페이지에서 학습자들은 각각의 아이더, 옵션 혹은 해결안을 기준에 따라서 평가한다. “Calculate” 버튼을 통해 학습자가 자기의 평가결과를 확인한 다음에 “Send to group”을 클릭하여 자기의 평가결과를 그룹평가에 보낸다. “Group Score”에서 그룹평가 활동의 결과를 확인할 수 있으며 어느 학생이 자기 평가결과를 보냈는지 확인할 수 있다. 또한 “Comments” 페이지에 들어가서 서로를 설득시키기 위해서 댓글이나 사진을 올릴 수 있다.

8- 이 와중에 팀장도 코드를 입력하고 나서 그룹평가에 들어가서 그룹의 평가 진행을 관찰한다. “Group Scores” 페이지에서 어느 학습자가 자기 평가결과를 보냈는지 확인한다. 모든 학습자가 자기 평가결과를 보낸 다음에 “Share results”를 클릭하여 SNS를 통해 교수자와 그룹평가활동결과를 공유한다.

# Appendix 7. Usability Questionnaire for "Evaluation Matrix" App

## Usability Questionnaire for "Evaluation Matrix" App

Hello, I am Yunus Emre OZTURK and a master's student in Educational Technology, Seoul National University. I am conducting a research on providing guidelines related to developing a simple mobile application for educational purposes. For this purpose, I have created a guideline, and then have developed "Evaluation Matrix" app by following that guideline. Now, through this usability questionnaire, I aim to explore its usability.

This usability questionnaire will ask about your demographic information and experience with "Evaluation Matrix" app. All data obtained from this study will be kept anonymous.

Participation to this study is not compulsory and participants can withdraw at any time without consequences.

I sincerely appreciate you for giving your precious time and effort for this study.

Yunus Emre OZTURK

Seoul National University, Department of Education,
Educational Technology Major

Firstly, please fill the following:

Gender: ☐ Female ☐ Male

Age: ☐ 18~25 ☐ 26~35 ☐ 36+

Date: _____

Brand of your phone: _____ Model of your phone: _____

Then, please answer the following questions based on your experience with "Evaluation Matrix" app. Mark one box for each statement that best describes your reactions to "Evaluation Matrix" app. If you feel that you cannot respond to a particular statement, then mark the center point of the scale for that statement.

| # | Statements | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree |
|---|---|---|---|---|---|---|
| 1 | I think that I would like to use "Evaluation Matrix" app frequently for this kind of learning activities. | | | | | |
| 2 | I found "Evaluation Matrix" app unnecessarily complex. | | | | | |
| 3 | I thought "Evaluation Matrix" app was easy to use. | | | | | |
| 4 | I think that I would need the support of a technical person to be able to use "Evaluation Matrix" app. | | | | | |
| 5 | I found the various functions in "Evaluation Matrix" app were well integrated. | | | | | |
| 6 | I thought there was too much inconsistency in "Evaluation Matrix" app. | | | | | |
| 7 | I would imagine that most people would learn to use "Evaluation Matrix" app very quickly. | | | | | |
| 8 | I found "Evaluation Matrix" app very cumbersome to use. | | | | | |
| 9 | I felt very confident using "Evaluation Matrix" app. | | | | | |
| 10 | I needed to learn a lot of things before I could get going with "Evaluation Matrix" app. | | | | | |
| 11 | "Evaluation Matrix" app has helped in conducting group evaluation. | | | | | |
| 12 | "Evaluation Matrix" app has supported creative learning environment. | | | | | |
| 13 | "Evaluation Matrix" app has supported collaborative learning environment. | | | | | |
| 14 | "Evaluation Matrix" app has similar User Interface with the apps I use daily. | | | | | |
| 15 | I would use similar apps for supporting learning environment if possible. | | | | | |

Lastly, answer the following question:

Have you ever experienced any error or problem while using "Evaluation Matrix" app? If yes, please describe it: _____

_____

_____

# Appendix 8. Usability Interview

## Informed Consent Form

I agree to participate to the study conducted by Yunus Emre OZTURK at Seoul National University, Department of Education.

The purpose of this study is to evaluate the usability of "Evaluation Matrix" app. I agree to be asked questions about my experience on "Evaluation Matrix" app, and I agree to be audiotaped for this study.

I understand that participation in this usability study is voluntary and I agree to raise immediately any concerns or areas of discomfort during the session with the study administrator.

All information collected in the study is confidential, and my name will not be exposed at any time.

Please sign below to indicate that you have read and understand the information on this form and that any questions you might have about the session have been answered.

Date: _____

Name: _____

Brand of your phone: _____ Model of your phone: _____

Signature: _____

Thank you! I sincerely appreciate you for giving your invaluable time and effort for this study.

Yunus Emre OZTURK
Department of Education, Seoul National University

The questions will be asked in this interview are as followed:

1- What three things did you like most about “Evaluation Matrix” app?

2- What three things did you like least about “Evaluation Matrix” app?

3- Was there any error/problem occurred when using “Evaluation Matrix” app?

4- If you want to add something in “Evaluation Matrix” app, what it could be?

5- Do you have any other comments about the system functions and regarding its usability?

# 한글초록

## 간편 교육용 모바일 앱 구현을 위한 모델 개발: 평가행렬법에 대한 사례 연구

현대사회에서 스마트폰과 같은 모바일 기기들은 우리 생활 속에 중요한 부분이 되었다. 부분 중에 하나가 되어 버렸다. 모바일 기기가 나날이 발전하고 있고 이를 교육에서 활용하는 비율도 증가하고 있다. 교육자들은 모바일 기기를 교육적 목적으로 활용하기 위해서 많은 노력을 기울이고 있다. 특히, 고등교육에서 모바일 기기 활용의 효과성은 다양한 연구를 통해 증명되고 있다. 모바일 기기의 초연결성, 카메라, 센서, GPS 등의 독특한 기능들은 다양한 학습경험을 가능하게 할 뿐만 아니라 학생들에게 학교 내외에서 일어나는 학습에 대하여 새로운 기회를 제공한다.

본 연구는 다음과 같은 과정으로 진행되었다. 첫째, 선행문헌 고찰을 통해 간편 교육용 모바일 앱 개발 위한 초기 모델(MODSEMA)이 개발되었다. 둘째, 모바일 앱 개발 전문가 3명과 ‘MODSEMA’에 대한 전문가 타당화가 진행되었으며, 전문가 타당화의 결과에 따라 ‘MODSEMA’가 수정되었다. 셋째, 실제 모바일 앱을 개발하는 사례 연구를 통해 MODSEMA의 타당화 및 추가 수정이 진행되었다. 본 사례 연구에서는 MODSEMA의 과정에 따라 교육용 모바일 앱을 개발하였으며, 그 결과 창의성을 촉진하기 위한 평가행렬법을 구현하는 모바일 앱이 개발되었다. 이 과정에서 필요에 따라 ‘MODSEMA’ 모델을 수정하였다. 최종적으로 교육자, 연구자, 개발자 등을 위해서 간편 교육용 모바일 앱 개발을 위한

모델인 ‘MODSEMA’가 제안되었다.

본 연구는 다음과 같은 시사점을 가진다. 첫째, 제안된 모델인 ‘MODSEMA’은 모든 교육용 모바일 앱 개발과정을 제시하며 어떻게 모델이 적용되는지에 관한 매우 구체적인 안내와 실제적인 사례 연구를 포함한다. 또한, 계속적으로 변화•개발되고 있는 주요 모바일 플랫폼과 앱 개발 환경 및 도구, 그리고 사용자 인터페이스(UI) Framework에 대한 정보와 안내를 제공하여 개발자들에게 새로운 기회 및 대안을 제시한다. 뿐만 아니라 각 단계마다 모바일 앱에 대한 다양한 사용성 평가 방법들을 제공한다. 둘째, 하이브리드 앱(hybrid app)개발을 지원하는 테크놀로지를 활용하여 용의성 (효과성, 효율성, 만족도)가 높은 모바일 앱을 개발하였다. 이를 통해 하이브리드 앱은 다양한 모바일 플랫폼(IOS, Android 등)에서 쉽게 작동이 되어 교수실행의 기술적 어려움을 해결할 수 있으며, 네이티브 앱(native app)과 비슷한 고성능과 고품질의 UI를 가질 수 있다는 강점이 확인되었다. 마지막으로 사례 연구에서 실시된 사용성 평가 설문 결과에서는 교육목적으로 다양한 모바일 앱을 활용하고자 하는 학습자의 요구와 필요성이 확인되었다.

**주요어**: 교육용 모바일 앱 개발, 교육용 모바일 앱, 모바일 러닝, 모바일 앱, PhoneGap, 평가행렬법

**학 번**: 2015-23310