ACKNOWLEDGEMENTS

Before all, for giving me the understanding, the strength and commitment to begin and finalize the study, a great thank is to God.

Then, I would like to extend my thanks to my research advisor for his all-round guides and advices. Without your comments, encouragement and follow up in the progress of the whole processes of the study, it was impossible to complete the work within the constrained time. For all of your delegations, thank you Mr. Asrat Mulatu!

The last but not the least, I would like to thank SMU for giving me the chance for the MSc study as a scholar being employed in the institution. Finally, I do not forget the initiations and cooperation of the staff members of computer science department in SMU.

For all, thank you the Almighty God again! And again!

Table of Contents

List of Acronyms

ADT – Android Development Tools
API – Application Program Interface
AVD – Android Virtual Device
C2DM - Cloud to Device Messaging
FCM – Firebase Cloud Messaging
GCM – Google Cloud Messaging
HCI - human computer interaction
IDE – Integrated Development Environment
JSON - Java Script object notation
MOODLE – Modular Object-Oriented Dynamic Learning Environment
MVC – Model View Controller
OHA – Open Handset Alliance
OS – Operating System
PC – Personal Computer
PDA – Personal Digital Assistant
SDK – Software Development Kit
UI – User Interface
USB – Universal Serial Bus

List of Figures

List of Table

Page

Abstract

Mobile phones are primarily designed for enabling communication over wireless network infrastructure like cellular network through audio calling. In addition to this, now days, they are getting smarter and becoming the center of entertainments by running different applications for gaming, playing audio to video data formats with facilities comparable to PCs. As the devices are very much portable to the day to day activities of the public, their usage can further be exploited in the education industry. This can be realized by designing a specific architecture for developing apps like Mobile learning applications. Technically, the mobile learning concept is one aspect of mobile computing applied in the education domain. This approach was attempted to make the traditional classroom confined learning experiences very flexible and independent of time and location.

According to literatures in the area, through distance learning and electronic learning, there has been a significant effort of extending learning delivery out of a classroom. Since e-learning was primarily intended for higher capacity computing nodes like desktops and laptops via wired network, it is mostly location dependent. To overcome this issue, the usability aspect of mobile devices in their wireless connectivity, popularity and mobile nature have been tried to be utilized through native and client-server architecture based mobile learning apps. In this study, hybrid architecture is proposed for higher education mobile learning scenario. This architecture is used to combine the strength of both the native and web apps in order to make the mobile learning app on time through push notification. Basically it includes mobile devices, mobile app, mobile devices optimized web app, cloud services and data based server as major component.

Mobile apps developed in hybrid architecture are installable and can pull content from Internet independent of browsers. Generally, by designing the whole systems physical architecture, a prototype implementation was made in agile process model. Finally the prototype was evaluated and validated to perform well for the core functionalities stipulated as usage scenarios.

***Keywords**: Mobile Learning Architecture, Mobile Learning, Higher Education*

CHAPTER ONE

INTRODUCTION

1.1. Background

The great advancement, in the area of information communication and information management has made a significant transformation in various aspects of the day to day activities of modern society. The transformation can obviously be seen in several domains like transactional business, health, educational, agricultural and many other sectors. The major factor for all these changes is due to the wide utilization of computing digital devices such as computers, smartphones, tablets and laptops. These computing machines have got higher capacity in processing of data and information as they can execute substantially large number of applications on different platform.

As a consequence of this, a considerable advancement achieved in the information communication and management was the conversion of different manual way of processing of information to a computer based system which is termed as automation. This can possibly be applied on a standalone or networked computer(s). Hence, based on the context of the domain under consideration, the suitable application for automation can either be desktop application for standalone computer or web page application for a large network of computers.

Recently, the architecture and interest of developing application software has been highly inclined to the web based scenario because of the usability and flexibility it offers. Web page applications can be developed using a wide variety of programming languages, platforms and technologies as they are intended to be accessed by different networked computing devices ranging from mobile phones to desktop computers. The computational capacity, portability, physical platform and versatility of those devices are also highly diversified and have an impact on architectural design of the web page and the way each device should be connected to the information network system.

Regarding the media used in information network system, there are two types of network systems as wired network that uses guided media and wireless network which uses air for information communication. Both systems are widely used by those earlier mentioned domains based on their respective context.

Mobile devices are mostly designed to use the wireless media to access networked systems. The infrastructure of wireless networks such as Wi Fi and cellular are becoming part of the standard in most of public service providing sectors. This has situated the environment for computing while on the move which is the most required flavor in the forth-coming computing technology. Furthermore, to enhance this approach of computing, now a day cloud computing (with concept of providing computing as a service) is playing a significant role. Therefore, in this paper, the core focus is on how to utilize these emerging features of the technologies in teaching and learning experiences.

Currently, there is a wider range of mobile and portable computing devices including laptops, tablet PCs and smart phones designed for different kinds of computation which can also be used for mobile learning applications. In this study, among those device tablet PCs and smart phones are targeted as they are suitable with the android operating system. This is mainly aimed to utilize the open sourced resources like android and to incorporate Google cloud services.

Generally, the idea or concept description of this study entirely revolves around designing a specific architecture for mobile learning application which has Google cloud service integrated functionalities. Hence, next, basics of the architectures of mobile learning application and the available options for improvements are discussed. Refining the existing problems regarding mobile learning applications with respect to the state of the art found in the literature, the objectives and the contribution of this research work has been set.

In many literatures related to the mobile learning application, from the realm of learning scenarios, mobile learning application is one part of the overall e-learning paradigm. Yet, the e-learning itself is included within the category of distance learning which was intended to make the traditional way of learning flexible[1]. The obvious problem related to the traditional learning method is being space and time dependent as its offerings are confined to a specific classroom and instructor led. So, the idea of mobile learning is rooted from the distance learning and e-learning. Besides, the continuous advancement in mobile computing and wireless technologies has given a sustainable ground for the mobile application currently and for the upcoming futures.

Furthermore, the future direction of computing, ubiquitous computing in line with the current intensive utilization of mobile devices is attracting app developers towards developing many kinds of mobile application of which mobile learning is one aspect. Either Mobile learning or mobile e-learning or M-learning is all about designing and developing application to provide educational content in anywhere-anytime-anyhow(any media format, formally/informally through mobile devices) mode on mobile devices through wireless network as presented in[1] [2]. Particularly as stated in [1] the mobile learning definition was conceptualized as
"Any sort of learning that happens when the learner is not at a fixed, predetermined location, or learning that happens when the learner takes advantage of the learning opportunities offered by mobile technologies."

Similarly, several authors [1]-[3] have coined the terms "*wireless, ubiquitous, seamless, nomadic or pervasive learning/education*" to mean mobile learning.
From teaching and learning points of view mobile learning is being facilitated in the 21st century learning environment structures which includes the principles of student-centered pedagogies, ICT implementation & integration strategies, innovative teaching practices, learning objectives and teacher's competencies as in [1].

As far as the architecture of the mobile learning application is concerned, based on the literatures regarding mobile learning application, the most suggested architecture was the client-server one as described in [4] [11]. The other category of architecture proposed for mobile learning was native. To some extent, there was also a trial of mixing the features of the two architectures as presented in[4] [5]. Native architecture mobile app s are heavy weighted, installable and hold static content. But these apps are good at utilizing the local device resource like camera, GPS as discussed in [4]. Mobile apps with client server architecture are simple mobile devices optimized web apps. The major drawback with this class of apps is less utilizing of local devices resources as compared to the native ones [4].

Therefore, in this study, hybrid architecture is proposed. The hybrid mobile app is used to combine the strength of both client-server and native features. Since the app of this approach is browser independent internet enabled, it has got the advantage to be seamlessly integrated with the cloud service in order to make it on time through push notification.

The proposed architecture of mobile learning application is also intended to utilize the opportunities provided by Google cloud and the android operating system. In Google cloud, there was a functionality to let application developers in sending message to a device running Android OS. This is through GCM (Google Cloud Messaging) which is a service developed by Google for device to device communication[6] [7]. GCM works in client-server architecture to enable communication between client and server apps. This can be done through upstream message from client to server and downstream message the reverse direction. Fortunately, Android OS has got the feature to support GCM and the latest version FCM (Firebase Cloud Messaging). The FCM has the cross-platform features unlike to the GCM.

Android is an open source platform initiated by OHA (Open Handset Alliance) in which Google is playing a leading membership role. On this platform, application can be developed using Java to run on a client machine and can transfer message to application running on server. Android also have several features as it is based on a monolithic Linux kernel which includes drivers for mobile device's hardware like screen, keyboard, camera, USB (Universal Serial Bus), Bluetooth. Besides, the kernel provides interface hardware and memory management, processes and other resources. There are also native libraries dependent on the hardware architecture of the mobile device. All these have made android a highly preferable OS for wider range of mobile devices. Taking these advantages of Android and Google cloud as an option, it is feasible to design and develop a notification and data message enhanced mobile learning application.

Generally, in this research process, the proposed framework of architectural design of mobile learning application for higher education" incorporates major components such as

- Educational contents management system that hosts the contents to be accessed by mobile devices through mobile app and any other internet connected device using browsers.
- Mobile device used as a terminal to deploy the mobile app and to launch the learning content from the content management system.
- Mobile app that is developed based on the proposed architecture.
- Part of cloud service offered by Google.
- Persistent data storage component.
- Both mobile and non-mobile devices that interact with the learning content management.
- And the line of interactions between these component.

In addition to these, the way the architectural components communicate through both wireless and wired line communication technology is seamlessly considered. Finally, to test the proper functionality of the overall components, the implementation aspect was done in agile process model through prototyping.

1.2. Statement of the Problem

It is believed that the communities in higher education institutions, students, instructors and other staffs are capable of utilizing mobile application for different purposes. In line with this, wireless infrastructures like Wi Fi and cellular networks are becoming common standard services in those institutions. As the devices are very much portable to the day to day activities, their usage can further be exploited in education industry. Therefore, designing a specific architecture for mobile learning application to best fit for delivering educational materials and supportive information in such institutes is crucial.

In this approach, it is possible to make the traditional classroom confined offerings of learning very flexible and independent of time and location. The possible mobile learning application architectures, native and web based client-server as it has been mentioned in the literature review section, have got their own respective drawbacks. Therefore, in this research work the mobile learning application has used the hybrid architecture of the two architectures by integrating with cloud services. The architectural design of the application has also considered all these issues against the constraints on mobile devices and opportunities available in computation technologies.

Using the hybrid architecture for mobile learning application, it is possible to design and develop a notification and data message enhanced mobile learning application to realize the idea of how to timely deliver educational content and material independent of time and space. Similarly, it is possible to get support and provide supportive information on lesson at anywhere anytime basis. Furthermore, the approach provides an option to virtually extend learning process to life-long through informal learning. Generally, this work aims to enhance the learning experience by settling architecture for mobile learning application to facilitate educational delivering activities as a complementary option.

1.3. Objectives

In order to solve the aforementioned problems, the objectives of the research are set to come up with proper solution. The general objective and specific objectives are described in the following subsections.

1.3.1. General Objective

The main objective of this study is designing architecture and implementing a prototype of mobile learning application for higher educational institutes.

1.3.2. Specific Objectives

In order to achieve the intended objective in this study, there will be several specific activities to be performed sequentially. These include designing a mechanism of

1. Developing mobile learning application prototype interface that will work in hybrid architecture.
2. Integrating the application with cloud service specially FCM which is the latest version of GCM.
3. Using hybrid mobile learning app for accessing educational contents from learning content management system
4. Sending and receiving push notification messages through the hybrid mobile learning app
5. Utilizing features of android platform for the cases from 2 to 5 above
6. Handling the heterogeneity of mobile devices screen size and computational constrains.

1.4.Contributions and Limitations

1.4.1. Contributions

The proposed research topic is aimed to solve the existing problem of timely delivering of learning content and supportive information using the available technological options. In doing so, it will show one way of utilizing cloud services on portable mobile devices for educational purposes. Similarly, it will bring forth a new insight on how to broadcast push notification messages from web page interfaces for educational app.

As the architecture intended for developing internet enabled, generic and cloud service integrated mobile learning application, it is a new technique of designing instant message enhanced mobile learning application. Thus it can be applied in every higher education institutes with a minimum customization. In addition the approach of this study is also scalable to the other industries and application domain.

1.4.2. Limitations

In its broader sense, mobile learning application for higher education institution can be considered as an option to the traditional face to face learning. However, in this study, the major focus is on how to design architecture and implement a prototype mobile learning application for higher education that can serve as a complementary option. Therefore, it can be used to facilitate the learning environment in higher education institution by enabling on time delivery of lecture notes, assignments and other supportive materials. In general, the formats of data that are best supported include text and light weighted graphics considering the overheads in wireless communication bandwidth and computing capacity of mobile device.

1.5. Organization of the Rest of the Thesis

The rest of this thesis report is organized based on the following sequence. In the next chapter, the literature related to the topic of the study is discussed. Then, in the third chapter the approach followed for the whole process is explained. Design and implementation goes in chapter four and five respectively. Finally the conclusion and recommended future works are include in the sixth chapter.

CHAPTER TWO

LITERATURE REVIEW

2.1 General Overview

As the use of information and communications technology (ICT) has become very significant in the public, it has made vital effect in the mode of delivering educational contents. The first use of ICT in education was practiced as e-learning which is electronic learning through the Internet and desktop system. In line with this, the rapid growth of portable computing devices and the wireless technologies and infrastructures has brought an opportunity to anywhere and anytime access and sharing of information. For instance, mobile phones are primarily designed for enabling communication over wireless network infrastructures like cellular networks through audio calling.

In addition to this, now days, they are getting smarter and becoming the center of entertainments by running different applications for gaming, playing audio and multimedia-rich contents with facilities comparable to PCs. These devices can also support web browsing for accessing social media and other mobile web sites. As the devices are very much portable to the day to day activities of the public, their usage in the education industry was tried to be exploited using apps like mobile learning applications.

In this literature survey, an attempt has been made to discuss the issues regarding how mobile learning concept is technically defined and specified by different literatures so far. Then, investigating the trends in mobile learning application is followed. Furthermore, in context to the research work, it is possible to categorize them into two based on their perspectives. The first one can be seen from mobile learning point of view while the other can be considered from cloud service implementation point of view. For both of them several research works are critically reviewed and a summary of all the reviews is included to indicate existing research gaps.

2.2 Mobile Learning Application Trends

The effort of extending the traditional way of offering learning has been there since the era of industrial revolution of the 18th century as described in [8]. In line with the emerging of electronic and wireless technology, this start is taken as a base for the evolution of e-learning and m-learning respectively as shown in figure 2.1. Furthermore, the figure also illustrates that the m-learning can be considered as the next generation of e-learning and the starting point for ubiquitous learning (cloud base mobile computing).

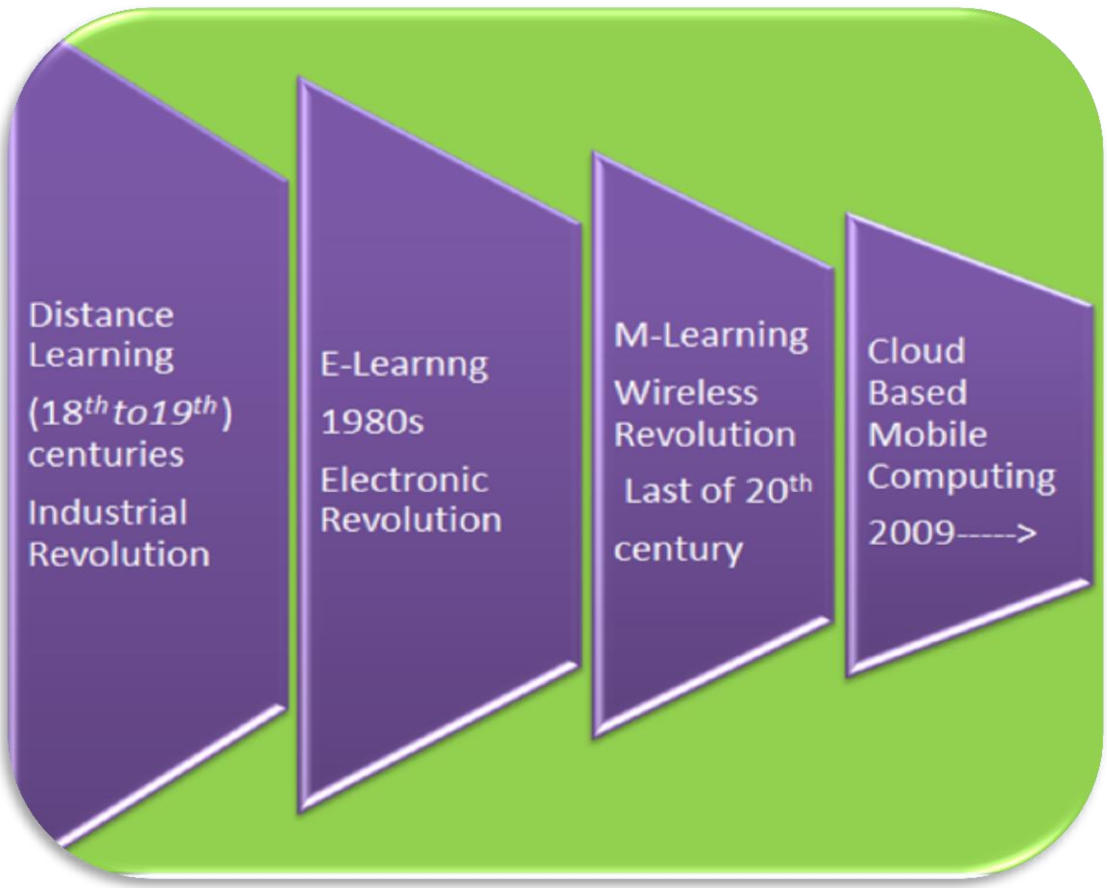

Figure 2.1: Mobile learning trends adopted from [8]

Technically, a mobile learning application is one aspect of mobile computing applied in education domain. The main intention of using mobile learning applications is to make the traditional classroom confined learning approach very flexible and independent of time and location[2] [9]. According to many literatures in the area through paper based distance learning and electronic learning (e-learning), there has been a significant effort of extending learning delivery out of a classroom [1] [9]. Since e-learning was intended for higher capacity computing nodes like desktops and laptops via wired network, it is mostly location dependent. In addition, because of its larger data size, e-learning contents are inconvenient to be accessed on mobile devices as they are storage and communication medium constrained.

In order to address this problem, several research works were done for designing mobile learning application architectural frameworks mostly in client-server based. As surveyed in[2] [10] different authors have coined certain terms like 'Digital natives', 'new millennium learners', 'the net-generation', 'the gamer generation' ,'generation M' and finally m-learning for mobile learning application. Similarly, several authors [1]-[3] have also coined the terms "wireless, ubiquitous, seamless, nomadic or pervasive learning/education" to mean mobile learning. Therefore, to refine this idea in more details, the literatures studied related to all aspects of the mobile learning are discussed as below.

2.3 Attitudes towards Using Mobile Learning Applications

As indicated in several previous research works, the involvement of mobile learning on mobile devices is a new approach in learning experiences. This approach imposes new features and different psychological attitudes on the targeted users. Literatures specify the issues in to various perspectives like human computer interaction (HCI), culture, gender and age of users, collaboration and privacy of the users and contextualization to a given situation. In order to get a good user acceptance, a mobile learning application should have a simple and user friendly interfaces for touching, navigating and linking functionalities as presented in [11]. This illustrates how the interaction with the mobile learning app can affect the user's attitudes.

Considering the factors such as gender, age, country, level of study and smartphone ownership, as described in [12] the results of the research conducted on 383 students and 54 instructors from five universities, indicated that M-learning can be one of the promising pedagogical technologies to be employed in the higher educational environments. Similarly, the finding in [13] shows that the impact of culture inherited from the physical environment on the use of technologies like mobile learning is very minimum and rather, the effect of technologies in shaping culture is high. Generally, as far as the attitude of the users' is concerned, mobile learning can fit to the situation of higher educations as studied in [2] [3].

2.4 Mobile Learning Application

To extend the approaches and experiences of e-learning on portable computing devices like smartphones, several research works have been conducted on m-learning. Hence, currently, mobile learning is becoming one aspect of e-learning [2]. What makes this mobile learning so convenient is that, it makes use of pocket and hand held devices such as smartphone and mobile phone to have access to learning scenarios. But, e-learning was basically intended for higher capacity computing devices like desktop and laptop computers. For instance, MOODLE is a well-developed online learning management system containing suite of larger sized learning tools. This feature has put flexibility limitation on the accessibility of e-learning through resource constrained mobile devices. Thus, mobile learning applications can be considered as an extension of e-learning to personal mobile devices[1] [2].

Mobile learning applications has got the advantage in delivering educational contents over portable mobile devices using wireless network like cellular and Wi-Fi independent of place and time [2] [11]. Unlike the other sort of learning experiences, mobile learning application can facilitate both formal and informal learning scenarios. As presented in [11] formal learning means a planned, an instructor led certificate oriented way of delivering education. But, informal learning includes all aspects of acquiring knowledge and skill randomly through exposure to internet via mobile devices. Therefore, taking all this promising features, the mobile learning application approach has overcome the access limitation of e-learning. Generally, the impact of this mode of learning on the traditional one, together with the architecture proposed and associated challenges are discussed in the coming sub-sections.

2.4.1 Components and Roles

According to the literature works in this subject matter, the interacting components in the mobile learning application system included [14]

1. **Mobile Devices** which are wireless network enabled

The devices are laptops, tablet and smart phone. But, for mobile learning application the most targeted devices are tablet and smart phone which are hand held.

2. **Software application** which runs on the mobile devices to provide interfacing and management facilities for the users.

The software could be categorized as cross-platform and platform dependent. In addition, it is possible to specify mobile learning applications as native and web based.

3. **Educational Contents** which are all the educational materials, supports and supportive information. The educational materials are, lecture notes, reference books, laboratory platforms/software tools, assessment information, assignment and exercise

The data format for these materials includes text, image, audio and video. The contents could be linked, loaded attached to the learning application.

The supports and supportive information are notices, alert or reminder communications or discussion through chatting and forums

4. **Software platforms, technologies**

The popular platforms are Android, iOS, Windows Phone. The technologies are also Phonegap, JQuery mobile, Web services.

5. **Wireless infrastructures**

The common wireless infrastructures are Wi Fi, Cellular and Ad hoc.

The researches have also identified that, up on the traditional inside class rooms learning activities; mobile learning can play the roles as ,optional supplement which is partially extended version, complement which is enhance it by adding features and substitute that is fully replacing [9] [14].

2.4.2 Architectures and Challenges

Many research studies have been performed on designing the interface and the architecture of mobile learning applications. According to these studies, the most general architecture proposed for mobile learning applications was the client-server [4] [14] [15] as summarized in table 2.1 where the contents could be accessed through mobile web interface apps. In this approach, the mobile learning application is hosted on server and accessed through web browsers from client side. The other type of architecture suggested for mobile learning apps by the literatures is the native application. This way the application is developed and deployed on standalone devices with static contents.

The studies have also indicated the different challenges associated with the issue as[11]

- Lack of standardized interface that can match with the variation in mobile devices' screen size.
- Variation of platforms' architecture used by different vendors of mobile devices for instance Android, iOS, Windows Phone.
- Limited computational capacity of mobile devices and communication capacity over wireless media.

2.4.3 Architectural Solutions: A Summary

For these prevailing challenges, several research works have proposed different approaches to solve. As far as the underlying platform is concerned, android was the most preferred as it offers several features as shown in table 2.1. To develop the mobile web interface for the mobile learning contents, the researchers have used the web services technology, JQuery mobile AngularJS, Bootstrap, Phnoegap, Adobe Air etc. Through these technologies the problem of customizing the mobile learning application interface to different mobile devices capacity was tried to be addressed. Besides, using the technologies like Web View (Crosswalk Rendering runtime engine) as presented in [6], there was an attempt to add cross-platform functionality to a new mobile optimized remote laboratory.

Generally, the possible mobile learning architectures proposed in different literatures together with the corresponding technologies are summarized as in table 2.1. The table is containing the survey of several research works conducted regarding mobile learning applications. The main focuses of the surveying were architecture followed, technologies used and the purpose for which a particular technology is applied.

Table 2.1: Summary of the proposed architectures of mobile learning applications with the available technologies for implementation

Literature Paper	Architecture chosen	Technology Used	Purpose	Comment/ Critique
[16]	Client-Server	CentOS 5.x	Operating system	• Intended only for lecture notes • Did not consider popular mobile device OS platforms
		My-SQL 5.0.x	Database	
		PHP 5.1.x	For code scripting	
		JQuery Mobile	For video playing method with embedded 'iframe' tag	
[14]	Client-Server	Android	Operating system	• The authors did not discuss how to make the app a cross-platform. • There is no mechanism of content management discussed. • No clear implementation of Phonegap
		Notepad ++	Editing environment	
		JQuery Mobile	Designing web Interface supported by Android, Blackberry, iOS, and Windows Phone platforms	
		Phonegap	Framework used to build native like web apps.	
[4]	Mixed Client-Server	Android, iOS, Windows Mobile, Blackberry, Amazon Fire OS, Firefox OS, Ubuntu Mobile OS, and Tizen.	Operating system platforms	• Mixed native and web based is a good approach. • Being Cross-platform is also strength. • But limited to Laboratory sessions only. • Could not include functionalities like group discussion and chatting.
		Web View	Crosswalk Rendering runtime engine	
		AngularJS, HTML5,	Web technologies	

		Apache Cordava, Phonegap		• Did not include a mechanism for managing contents of educational materials.
		LabVIEW	Interface to Lab Software	
		REST (Representational State Transfer)	Web service Message passing	
		HTML, JavaScript, CSS	Programming Language for both client and server side	
[17]	Client-Server	HTML5, JavaScript, CSS	Scripting language	• Methods of making mobile learning application development very flexible are discussed in detail. • But in this case the implementation of the application was seen from teachers' side only. • There is no way of letting students to use the application.
		JQuery Mobile	Cross-platform mobile application	
		JSON	Interoperability	
		NodeJS modules	Allows designing scalable applications	
		MongoDB	Database Management	
		Mongoose	For the connection to the MongoDB database	
		Jade	Template engine to render HTML	
		Express	Web-application framework to build multi-page web-applications	
[15]	Client-Server	LMS (Learning Management System)	Software systems for administrating, tracking, and reporting the on e-learning programs	• Trying to integrate with the existing learning management system is good idea. • But the app is intended only for limited formula.
		PLE(Personal Learning Environment)		
		HTML5 CSS3	To build the application	

		CofeeScript	To implement the domain logics	• Relevant educational content should be included.
		Twitter Bootstrap	Have grid system to correctly position the interface's elements and the framework itself independently of the device resolution	
[18]	Client-Server	PHP, HTML, Java JavaScript-AJAX	Languages used for developing interface	• Implementing low level interfaces to hardware using C, C^{++} and python is strength. • But mobile devices with lower screen size than tablet pcs are not considered.
		C and C^{++}	Programming languages to control input/output signals for a specific lab device	
		Python	For reading data from laboratory hardware devices	
		Apache server	WebLab server for managing the system	

2.5 Google Cloud Messaging (GCM) Implementations

In Google, there is an interesting messaging service provided as GCM with its new version as FCM. This service was initiated for the first time freely by Google in the framework called C2DM (Cloud to Device Messaging) [6]. The main purpose of C2DM was to facilitate the communication and linking between android application on mobile devices and server in case of updated data. In such a way, the server sends push notification message of 1KB to the client app. As android applications support intents, on the client side, up on receiving the notification, the app would get triggered by intent and connected to sever to fetch updated data.

C2DM was fully replaced later as described in [6] with a better functionality service named as GCM. The later service was intended to enable a third party developers to downstream and upstream data packets between servers and their Android applications respectively using mobile devices running Android. It is also offered for free by Google. The transfer data packet could be a lightweight data message to trigger the Android application for a new message to be fetched from the server (i.e., a linker or alarm data to link a huge-sized uploaded system data), or it could be an exact data message containing a maximum of 4KB of payload data to be used instantly[19]. Both queuing and delivering of messages are supported by GCM. Recently, Google is introducing FCM as a latest version of GCM. Unlike the GCM, cross-platform functionality is included on FCM.

Since Google is the member of OHA, the GCM service is targeted to only Android platform. But, for other platforms, there are also similar services like Apple Push Notification Service (APNs) for iOS and XTIFY that supports Android, Blackberry and iOS. XTIFY was owned by IBM as of 2013[6].

GCM service was tried to be implemented in energy efficient way on smart phones for different wireless technologies [7]. According to the experimental result of the article GCM notification is more energy efficient on Wi-Fi than on cellular 3G or 4G network. In addition, for the real work, GCM was tested [19]for home automation system. In this scenario, it was used to facilitate the notification message communication between home appliance, web server and application on mobile device. In a similar way the proper functionality of GCM was tested [20] with third party web application and android application developed on eclipse.

Therefore, as per the literatures discussed so far, the messaging service was tested in different scenarios. In addition, they have suggested the applicability of the services in various domains. However, the studies did not settle a comprehensive architectural framework for integrating the service in a targeted scenario. Furthermore, how to make the service a cross-platform was not investigated to the extent required.

2.6 Summary

According to the discussion in previous sections, on one hand, there were several researches conducted to identify the significances of mobile learning applications in education industry. The studies have also specified the role and impact of these applications in computing technology as one aspect of electronic learning on mobile devices. In line with that, mobile learning applications can be considered as starting point for ubiquitous learning for the future [3][21].

Furthermore, the studies have also proposed options on how to design and implement the architecture and interface of mobile learning application using different technologies. The studies mainly identified two possible architectures with available technologies as shown in the table 2.1 above. These include the native and the web-based mobile learning application. The native mobile learning applications are standalone app deployed on mobile device. The applications are simply installed and play on mobile platforms. This approach has an advantage as it can be used independent of internet connection. But, as mentioned in[4], this mobile learning application architecture has the following problems:

- Native or Standalone application has limitations of fixed or static content, not convenient to deploy on large number of devices and not suitable to be used on different platforms

The web-based applications are implemented in client-server architecture and accessed through mobile web browser. These applications are totally web applications formalized for mobile devices. The approach was suggested as a solution to the problems of the native above. Even though the web based approach has solved some limitations of the native one, there are still other drawbacks related to it as:

- Web based client-server applications
  - Are browser based and require platform independent technology to be accessed
  - Have limited access to local device hardware resources.

As it can be observed from the summary in table 2.1, the most preferred mobile learning application architecture is the web based due to the reasons described earlier. In [4], it was tried to add native like functionalities to the optimized web based mobile learning application for laboratory sessions. The common problem to the mobile learning applications was being limited to narrow domains as stated in comment/critique column of table 2.1.

On the other hand, there were separate research works on how to utilize the cloud messaging service provided by Google as GCM on mobile devices running android. The research studies in the area also have tried to test the functionality of the services in different scenarios discussed earlier. As presented for future work in [6],GCM can be implemented in domains such as medical, education, civil/ government applications, corporate applications. Despite suggesting the possibilities, there is no a general framework specified to apply the service in those domains. In addition, a mechanism of making this service a cross-platform was left as remaining research issues in the area.

Therefore, to exploit the flexible features of the mobile devices for complementing the formal learning in higher education, a more feature rich mobile learning application can be developed by integrating the two research perspectives. The method of combining will be accomplished by making the mobile learning application architecture a fully hybrid of native and client server. This way the application will be browser independent but Internet connection enabled. Beside, hybrid mobile applications can launch contents from the Internet. In doing so, it is also possible to integrate mobile learning application with cloud infrastructure to utilize the services and resource and hence increasing features and functionalities to the app.

CHAPTER THREE

METHODS AND TOOLS

3.1. Methods

In the proposed architecture of mobile learning application, several system components have been incorporated to be physically modeled and integrated. The components are mainly educational content management, mobile app and interfaces to the cloud and other components. To accomplish all these stuffs, a specific software process model has been followed. In the coming subsections, all the relevant hardware and software tools and technologies used together with proper process model are explained.

3.2. Process model

As there are different components to be integrated in the designed system, critically analyzing all the functional and non-functional requirements at a time was impractical. To deal with the dynamicity in analysis and integration of the components of the system, agile process model was used. The rationale behind utilizing this process model is for it permits an iterative prototype development approach to entertain changes on the system aspects. Whereas, the other approach like waterfall enforce the complete specifications of all the requirements. Therefore in the process of developing a prototype for the suggested mobile learning application, the agile approach was used starting from designing and implementing the Internet enabled mobile app. Then, for integrating the app with ***Google cloud messaging i.e. Firebase***. Finally, the approach was used for enabling the app to launch contents from online learning content management system.

Generally, based on the gap identified in the summary of the literature review in the last section, to come up with a specific design model or architectural framework of the mobile learning application of this study, the iterative prototype testing was done separately for each the aforementioned cases. Finally, the overall aspects of the system integration were tested through agile approach.

3.3. Software Tools

There are various kinds of software tools used for the proposed research study in different instances. The type of software includes platforms, programming languages, editors, database server, Google cloud server FCM and related AVD (android virtual device) tools.

3.3.1. The Possible Platform Used

Presently, there are several operating systems being used on different mobile devices manufactured by different brands. These operating systems include iOS for Apple, Android for Samsung and Symbian for Nokia devices. Among them, Android is the most popularly used one. Furthermore, it is freely available as open source with many important features and functionalities for the development of the prototype in this study.

Generally, android is an OS based on Linux kernel with a Java programming interface consisting of several necessary components as shown in figure 3.1. The major components include: a complete stack of OS, middleware and applications, Android SDK **(**Software Development Kit**)** which provides all the necessary tools to develop Android applications (Complier, Debugger, Device emulator, Virtual machine, and rich user interface library). Furthermore, it supports 2-D and 3-D graphics using the OpenGL libraries, Webkit, access to the file system and provides an embedded SQLite database.

From the available ranges of android platform versions, Android 4.3 (Jelly Bean) was used since it can support a wide range of users as indicted in figure 3.3. Regarding this research study, the relaying of push notification, messaging, chatting and forum discussions using the mobile learning is the central focus. These could be accomplished using several features more specifically elements indicated on the platform architectural layers presented in figure 3.1.

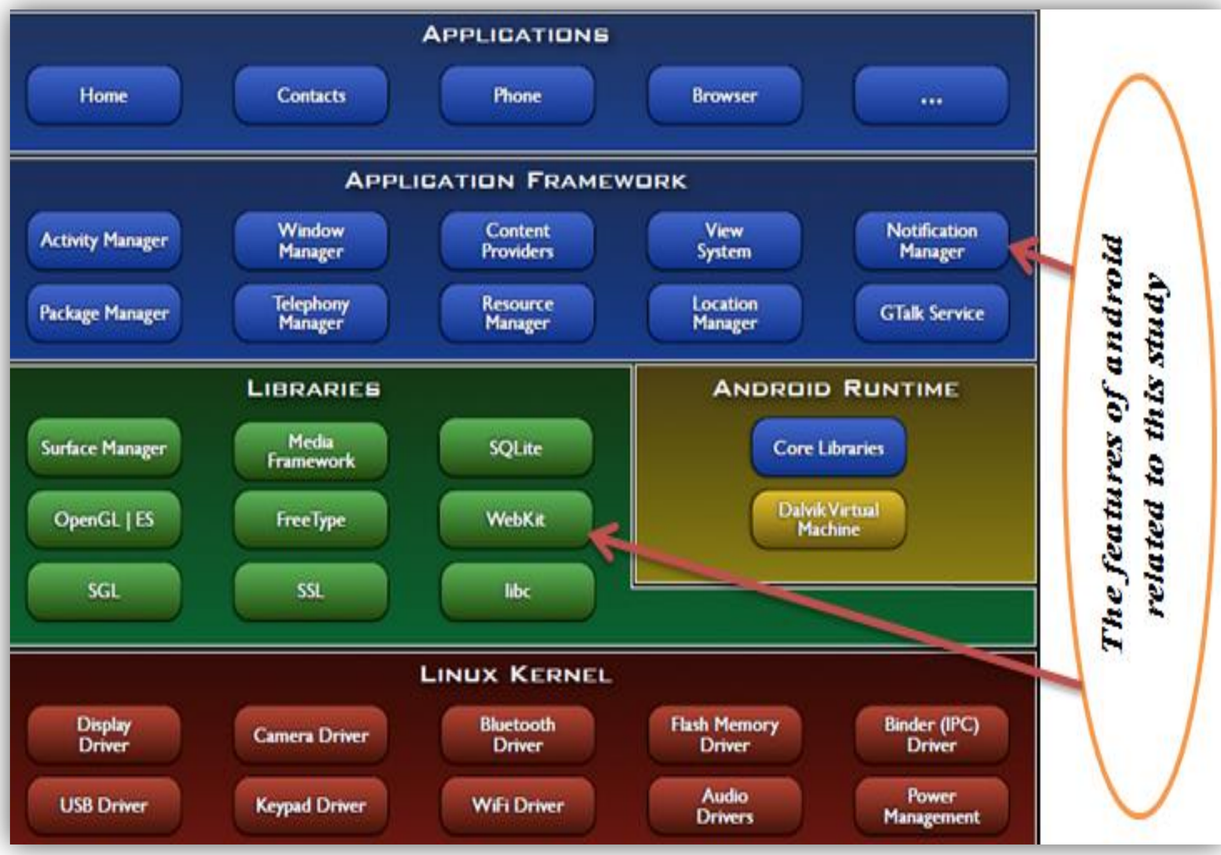

Figure 3.1: Android OS general structure

3.3.2. Android Development Environment

One of the main sub-parts of the mobile learning application architecture proposed is the android app that runs on mobile devices. On Android there are specific features intended to facilitate application development by a developer. These include:-

- **Eclipse** which is a separate multi-language software development environment containing a base workspace and an extensible plug-in system for customizing the environment for development on Android platform.
- **ADT(**Android Development Tools**)** is a plugin for the Eclipse IDE(Integrated Development Environment) that is designed to
  - Provides a powerful, integrated environment in which Android applications built.

- Extends the capabilities of Eclipse for quick setting up of a new Android project, create an application UI(user interface), add packages based on the Android Framework API and
  - Provides debug applications utilities.
- **SDK** provides the API (Application Program Interface) libraries and developer tools space necessary to build, test, and debug apps for Android.

By separately integrating those platform components, it is possible to develop the required mobile learning application.

On the other hand, there is a full functional Android Studio designed for developing android apps. In this research process, the Android Studio version 2.2.3 is used as it contains features to integrate apps with the internet and cloud. Furthermore, the development studio consists of all the required components like ***Eclipse ADT*** and ***Gradle*** scripts as shown in figure 3.2.

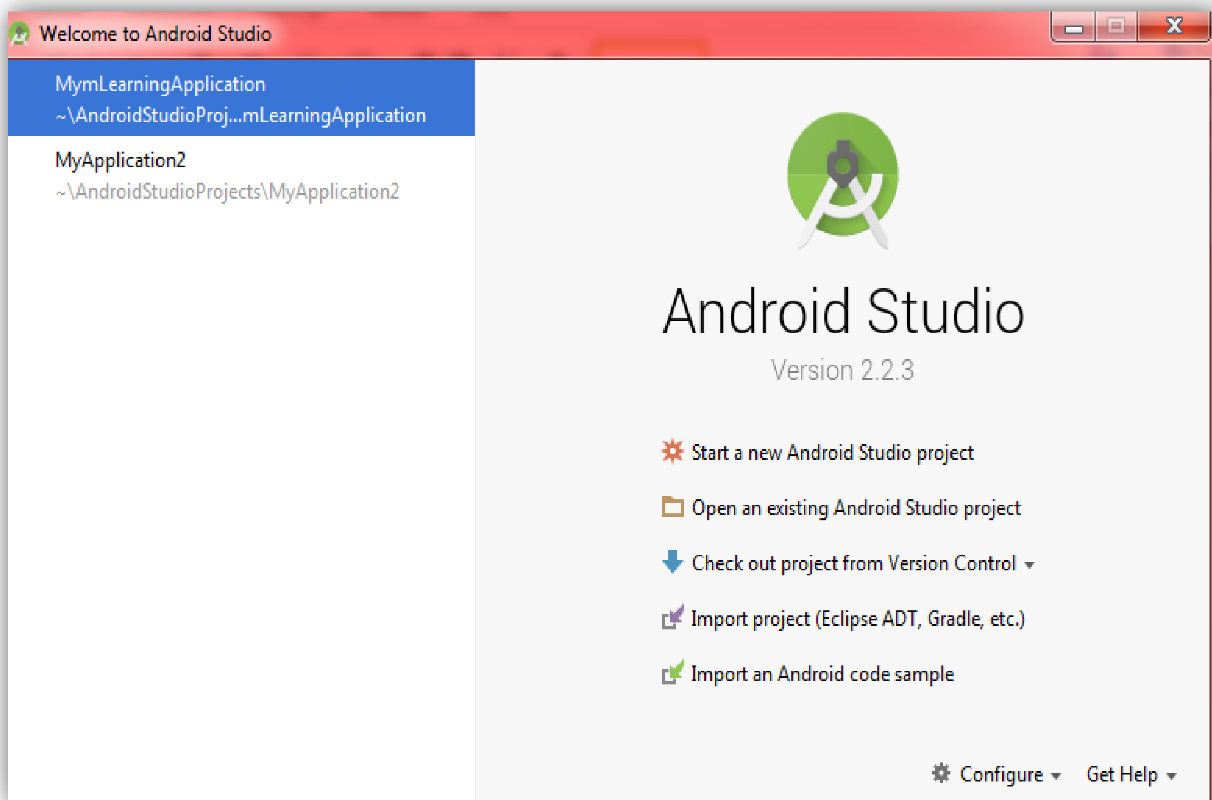

Figure 3.2: The version of android studio used

3.3.3. AVD and API level

For the purpose of testing the proper functionality of the prototype of the mobile learning application in this study, several AVDs with range of API level was used. The AVDs are software that emulates the real device sharing the hardware resources of the hosting device. Generally, the sample of AVDs used with their corresponding APIs is specified in figure 3.3. On the figure, the necessary features like platform version, API level and cumulative distribution (which is the usage statistics) are clearly illustrated.

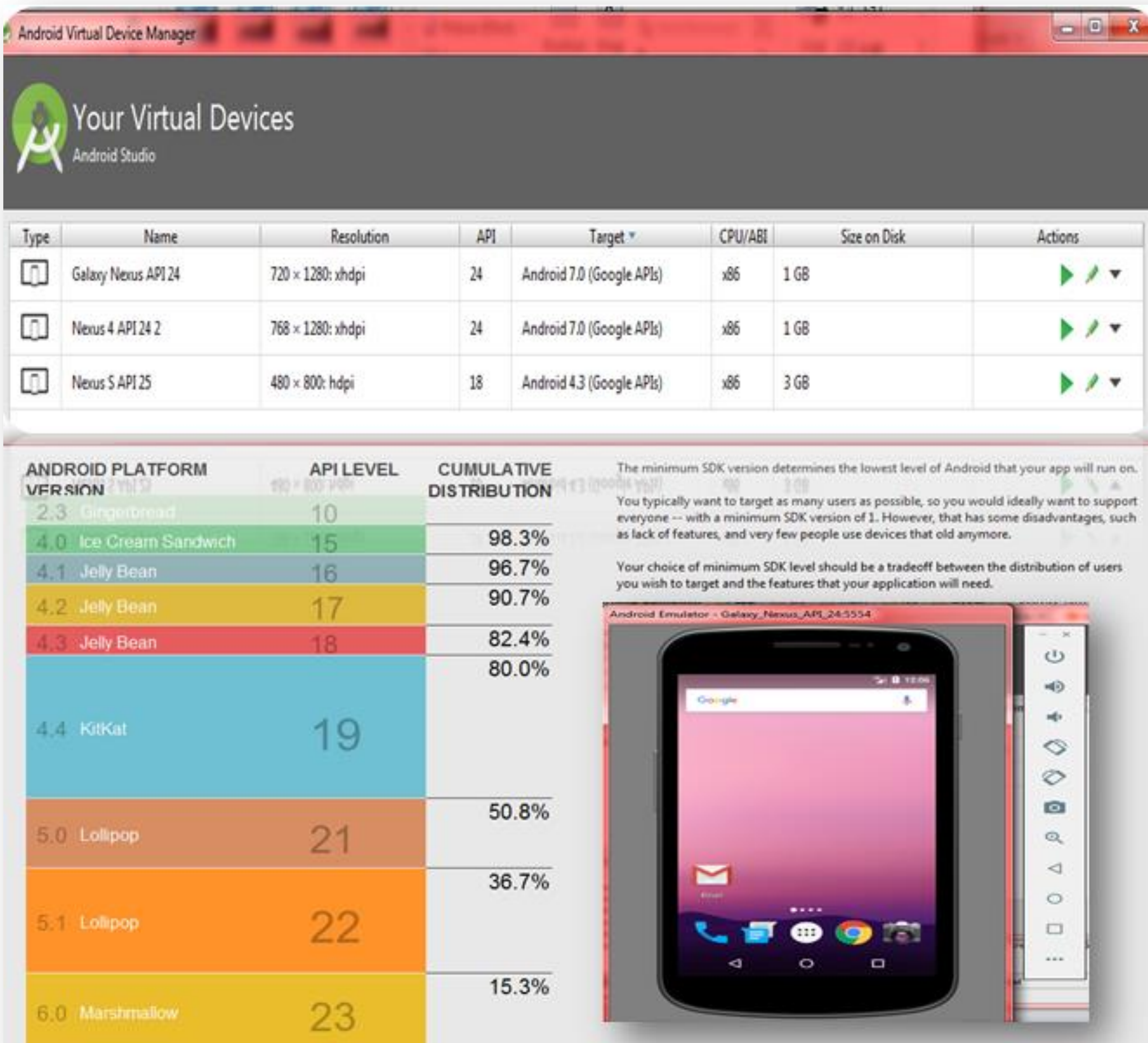

Figure 3.3: Sample of the virtual devices used for the prototype testing.

3.3.4. Google Cloud API

Currently, Google is introducing a new version of messaging service known as FCM. According to the information on the firebase console, the new version has cross-platform characteristics as in figure 3.4. The interface in the figure was used to add, configure, and synchronize all the necessary features and functionalities of the firebase to the mobile learning application. From all the available options on the dashboard, the ***Auth*** and ***Notifications*** are used for this study. Regarding the platforms the case of the iOS is not the part of the study. Therefore, on the console/dashboard,

- The ***'Add firebase to Your Android app'*** option is used to create a firebase project that handles all the interaction of android studio project. This option serves to integrate all the firebase dependency and library files to the android app project file.
- The ***' Add firebase to Your web app'*** option is used to link the firebase service to the web app
- The ***'Auth'*** option is used to configure setting for authentication sign up, sign in method.
- The ***'Notification option'*** option is used to send simple notification message.

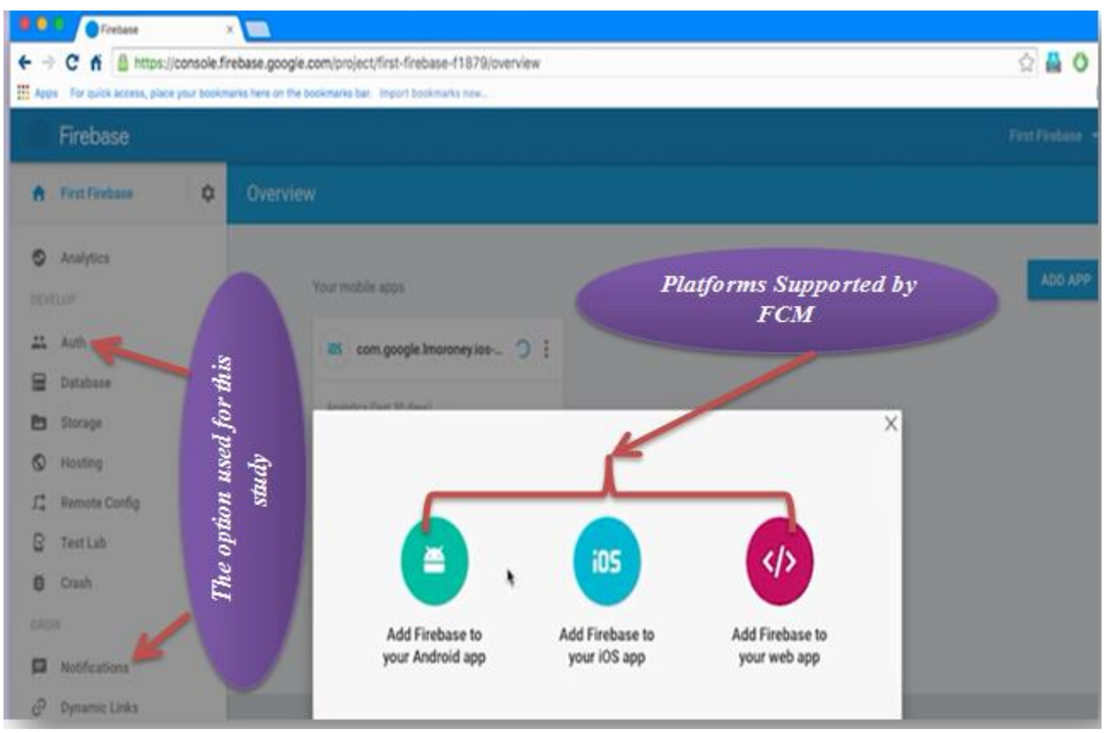

Figure 3.4: The firebase console options used

3.3.5. Web Technologies Used

The architecture of mobile learning application system in the study consists of a web application used in online learning scenarios. In order to enhance the access this systems through mobile devices, several web development technologies like JQuery Mobile, Bootstrap, PhoneGap and AngularJS had been used by different authors as shown in Table 2.1 of the previous chapter. According to that table web technologies like JQuery Mobile and PhoneGap intend to enhance the functionalities of web app to use local device resources[4]. However, in this study, since one of the major emphases was making the mobile learning content management web application a responsive one, ***Bootstrap was preferred*** as it was primarily intended for this functionality.

3.3.6. Bootstrap

Bootstrap is a simple HTML, CSS and JavaScript framework for developing a responsive and mobile friendly website. A responsive website means that, the property of the site to automatically adjust itself for good user experiences on all devices, from smart phones to desktops. In Bootstrap, there is ***div*** class attribute called ***container***. The attribute is used to set page content's margins dealing with the responsive issues of the layout. It has the row elements and the row elements are the container of columns also known as ***grid system***. Therefore, in this study, to adjust the mobile learning content management web application to different screen sized device, '***container-fluid***' was applied with PHP scripting.

Finally, for the prototype implementation, Notepad++ editor and the PHP script language for server side scripting were used. The reason behind using PHP was it allows to encode and decode JSON (Java Script object notation) by the help of ***json_encode()*** and ***json_decode()*** functions. In this manner, the JSON which is an open standard for exchanging data on the web was utilized for relaying message form the web server to the FCM server.

3.3.7. Persistent Data Management

Obviously, the mobile learning application in this work requires a back end permanent data storage part. For this purpose one of the commercially available database servers like SQL Server, Oracle database, MYSQL etc. can be used. In the prototype test of the study, the MYSQL was implemented.

Generally, based on the required functionality of the intended mobile learning application, the latest version of all the tools were preferred to be used. Furthermore, each tool used for a specific implementation aspect was selected considering the behavior and compatibility it has with the other related tools. Thus, the tools specified in this study are focusing on these issues. Regarding the process model, only the context of the subject matter is considered.

3.4. Method of Evaluation

Finally, to evaluate the performance of the system prototype in the real environment, a networked system was configured. The networked system consists of Ethernet switch, five desktop computers and one laptop computer. The desktops were used for accessing learning content and sending push notification. The laptop computer was used to host the components of the mobile learning application. This setup was done in single room of computer laboratory. In doing so, the proper functionality of the system prototype was tested by letting the experts and students to use and observe against the objectives set.

CHAPTER FOUR

PROPOSED MOBILE LEARNING ARCHITECTURE

4.1. Learning System Model

In order to come up with the intended architecture of mobile learning application in the study, the overall realm learning system was modeled as shown in figure 4.1 below. As it can be observed from the model, the mobile learning application can be considered as a sub part of online learning. Specifically, the app can further be refined to serve for formal and informal learning based on the intended flexibility. In this manner, formal learning can be referred as a kind of learning which is certificate oriented and offered by instructor led approach whereas informal learning is a random way of learning. In this chapter, the whole system's architectural aspects are discussed.

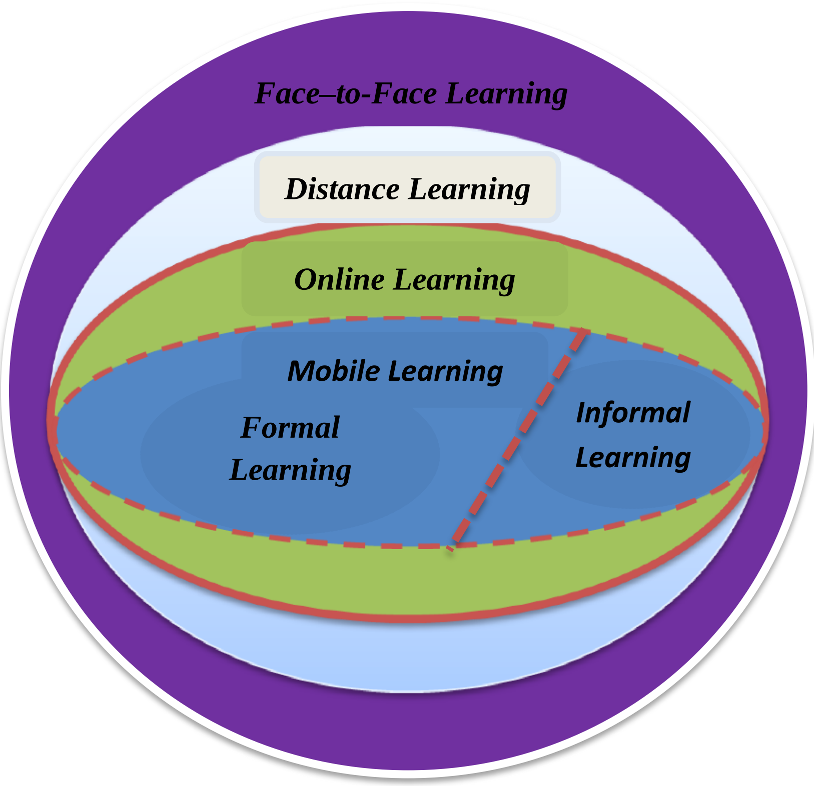

Figure 4.1: Mobile learning from the general learning system perspective

4.2. Design Procedure

The proposed architecture of mobile learning system of the higher education has a couple of interacting components as depicted in figure 4.2. In order to achieve the proper functionality of the whole system, the role and the method of interaction for every component was specified independently through agile approach. As the core of the design process was on developing hybrid architecture for mobile learning application, modeling this component was the kick off procedure. Then, the sequential integration the remaining components followed. Furthermore, the internal structure design took the MVC (Model View Controller) pattern. This was applied to manage separation concern so that when modifying one aspect the other should remain unchanged. Generally, the proposed system physical architecture is as in figure below.

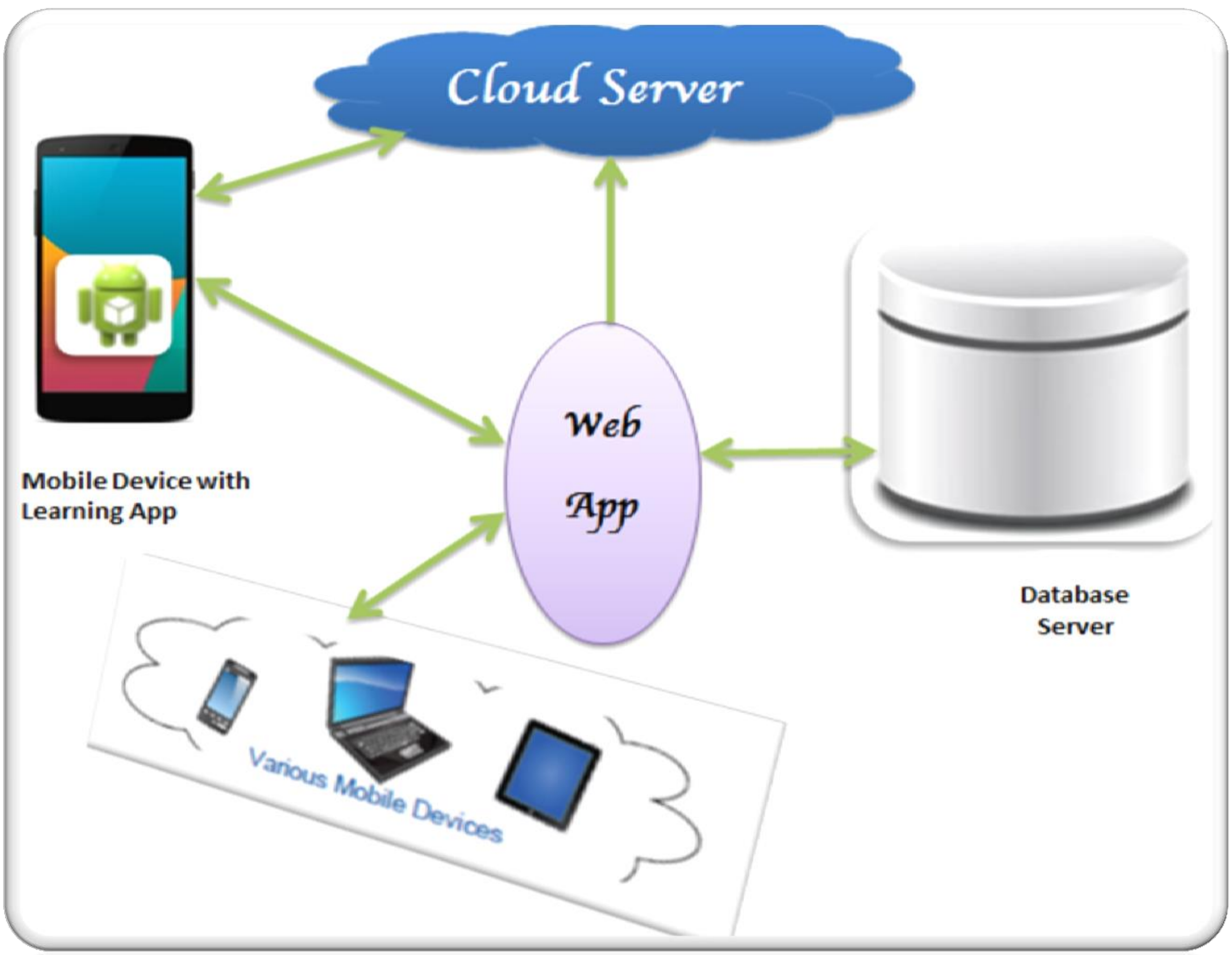

Figure 4.2: The proposed hybrid mobile learning architecture

4.3. Description of System physical Architecture

The general architecture of mobile learning application of this research study has major subcomponents described in the above subsection. As it is represented in figure 4.2, the whole physical system architectural framework consists of

- **Mobile Device**

  According to the context of the study, the mobile devices are those android powered and used to deploy the mobile app. These include mainly cell phones and tablet PCs. In the system architecture, these devices have the highest interaction and number. As the topic itself is concerned with mobile and mobile features, the devices play a central role in interacting with all the other parts. The wireless communication between the device and the other components is seamlessly integrated so that it is not depicted on the physical structure.

- **Mobile App**

  Here comes the core concept of the study as this app is based on hybrid architecture. It was made so in order to integrate with cloud services since mobile apps in hybrid architecture are browser independent internet enabled.

- **Cloud Server**

  In the suggested mobile learning application system, both the mobile app and the web application were integrated with the cloud server. This was intended to enhance the functionality of mobile learning application by technically including the available services appropriate to the context.

- **Web App**

  This is the component that facilitates and manages the communication between the frontend interface applications with the back end persistent storage. The whole components can only through it only. Thus, this part plays as a bridge of interfaces and communication.

- **Database**

  It was intended to store crucial data consistently and permanently. In the physical architecture the database is used to store all the data related to the mobile learning application.

Apart from the aforementioned components, the mobile learning application can be accessed in different devices using the web application sub component. In this perspective, it is possible to say that the application has a good portability and accessibility features. For more illustrations of the necessary logical steps undertaken in the whole system operations, the sequences of interactions are stated in the implementation subsection.

CHAPTER FIVE

IMPLEMENTATION AND EVALUATION

5.1. Mobile Learning System Application

In this study, the intended mobile learning system architecture consists of external and internal components. The users/actors of the system are considered and modeled as outside environment/component of the system. Whereas, the actual mobile learning application's architectural framework is analyzed and designed separately as internal aspects of the system. Generally, the functional requirements are specified as per the specification of the proposed system architecture related literature works. Therefore, the detailed specification and description of each component is as below.

5.1.1. Actors

The potential users that interact with this system application, specifically the learning content management are categorized as below based on their respective roles.

Central Admin: has full control over the system in accessing, managing and providing privileges for the other users.

Instructors: have the privilege to access, add and remove on specific and authorized contents.

Students: have the privilege to access and add on specific and authorized contents.

Staffs: have the privilege to access, add and remove on specific and authorized contents.

5.1.2. Mobile Learning Application

The mobile learning application suggested in this study by itself has two sub-components. The major subcomponents are the installable mobile app and a mobile device optimized learning content and messaging management web application. The two components are integrated and abled to communicate through cloud messaging service offered by Google. In the following subsections, the detailed specification of these components is discussed.

5.1.3. Mobile App

This is a hybrid application which was developed for android platform using android studio version 2.2.3 development environment. In the category of mobile applications, hybrid apps are installable app that can utilize local device resources efficiently. Furthermore, these apps are internet enabled and hence capable of launching content from there. Regarding the context of this study, one of the logic behind of modeling the mobile app hybrid is to integrate it with cloud service FCM. Since it is internet enabled, it can access internet network independent of browsers. Thus, using the app it is possible to access any online learning contents like lecture notes, references and assignment works through mobile device. The other rationale behind of making the app internet enabled is to acquire the capability of sending and receiving push notification offered by cloud messaging service in Google. The general working structure and the sub-systems' interaction framework of the app is as illustrated in figure 5.2. In order to exploit cloud functionality, the app was seamlessly integrated with Google cloud.

5.1.4. Google Cloud Integration with Mobile App

Currently, Google is providing a cloud service called FCM (Firebase Cloud Messaging). FCM is the latest version of messaging services that works on different platforms such as iOS, Android and also web app. This service is mainly intended for delivering push notification and it is offered for free. Push notification is the way of broadcasting and delivering notification message from the server to a registered and up online mobile devices. For these functionalities, in Google FCM dashboard/console which is available at this url '***https://console.firebase.google.com'*** was used as an interface. This dashboard has several library files that can be downloaded and included to android application project in android studio. Some of these files are

- *compile 'com.android.support:appcompat-v7:25.3.1'*
- *compile 'com.google.firebase:firebase-messaging:9.2.0'*

- *compile 'com.google.firebase:firebase-auth:9.2.0'*
- *compile 'com.firebase:firebase-client-android:2.2.3'*

On FCM dashboard, there is also an option for creating a firebase project that could be integrated with the package name of the android studio project. For the prototyping of the mobile learning app of this study, a firebase project was created by the name '***MymLearningApplication***' and assimilated with the android studio application project package name '***com.example.addisu.mymlearningapplication'.*** Furthermore, the dashboard provides several options like

- Setting authentication methods for instance through Gmail, Facebook, Google++,
- Sending simple message to the client application on the mobile device
- Supporting real time database and other as can be seen in figure 5.8(b).

Finally, after the dependency file called '***google-services.json'*** was downloaded and included to android project. Then, the above library file are included to the '***Gradle'*** files of the project and synchronized with all the project files.

5.1.5. Design of Basic Functional Requirements

For the prototype implementation in this study, some major and basic functional requirements were identified by agile approach. Since the suggested mobile app is for android platform, in the android studio project app file has the following contents. These are:-

- **Activity classes**
  This represents the user interface components of the app on screen of a mobile terminal. It can hold several subcomponents as themes, views etc.
- **Views**
  It is a single element on a screen of activities and can be considered as a building block of the activities. Example Button, ImageView, TextView, WebView
- **Services**
  Used to handle functionalities that run at the background

- **Intents**

  This is the part that handles the mechanisms of navigating between activities through intent messages relaying.

- **Firebase instance classes and services**

  This part is intended for capturing the functionalities of integrating the mobile app with the Google cloud service FCM.

By applying the above stuffs on android studio, all the necessary functionalities of the mobile app have been accomplished. The required functionalities are used for interfacing with the whole system components. Thus, the basic requirements are user interfaces for interacting with every unit of the system. These include registering a user to the system database, accessing learning contents, receiving and sending push notification. Besides, the functionality of navigating within the system component is also an important issue.

Generally, for the implementation of the system, the detail specification of the architectural framework and the unit components are discussed in the following section.

5.1.6. Content and Messaging Management

In a mobile learning or any electronic learning, persistently managing the actual learning content is the major issue. In line with that, controlling the interaction of the user with the system needs a great focus. To deal with such a crucial aspect, in this study the learning content and messaging management subcomponent of the mobile learning application was considered separately from the mobile app. This subcomponent basically is intended to incorporate the back end persistent large sized data storage and hosting web page interfaces for facilitating interaction of the users with the system. The learning content and messaging management is purely a web application. Thus, in this approach the system architecture would take client-server architecture.

As this component is simply a normal web application, it could be accessed by different ranges of terminals including desktop computers and all Internets enabled mobile devices. Therefore, for the learning content management web application in this study, to be accessed with a good user experiences on all devices of different screen size, the bootstrap web development technology was used. The bootstrap technology provides options for rendering web page contents being optimized to varieties of devices' screen size. In this manner, the bootstrap is used to develop a responsive web application that adjusts itself to a given mobile devices' screen.

Since this component needs backend persistent data storage, a database server should be in place. For the prototype implementation of the research work, the phpMyAdmin MySQL XAMPP server was used. For serve side scripting PHP was implemented while Notepad++ was used for the front HTML scripting. The formats of data managed in this part are as illustrated in figure 5.1. The reason behind of limiting the data format was to optimize the size of data for resource constrained mobile devices.

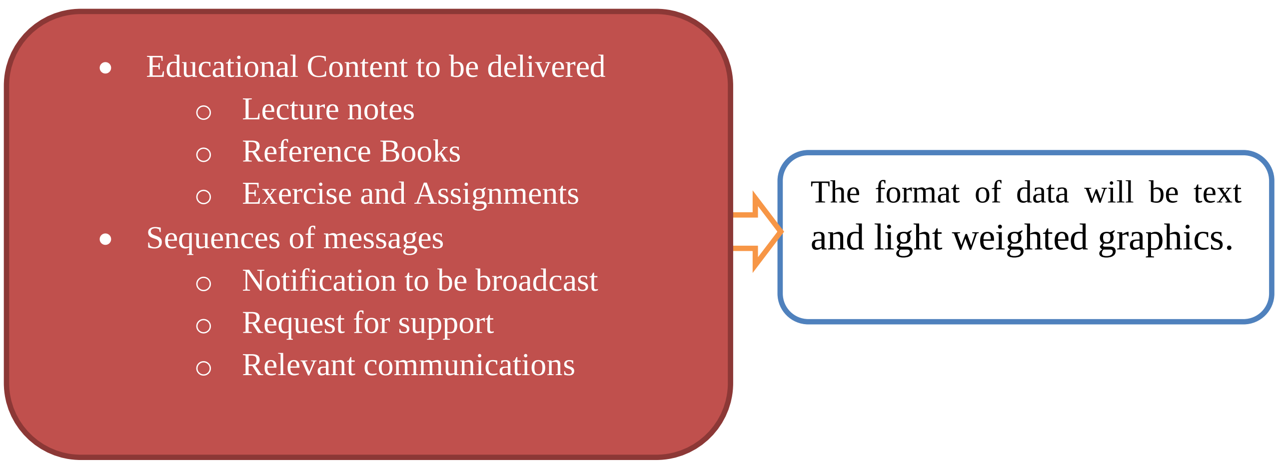

Figure 5.1: The data format supported by the system.

5.1.7. Google cloud integration with Learning content and Messaging Management

In this study, the core part is settling an architectural framework and the mechanism for sending push notification and message from learning content management web page. The process of managing tasks related to notification is facilitated by FCM. However, to enable a third party web application for utilizing this cloud service, on the console of the firebase there is an option of adding firebase to web app as in figure 5.8(b). Therefore, the learning content management web application suggested has incorporated a web page integrated with the firebase. Hence, for the mobile learning application in the study, this is the major component to broadcast a notification message to the android mobile app.

According to the literature survey so far in mobile learning perspective, all of them had used browsers to have access to the learning content on the internet. Suggesting hybrid mobile app for learning context in this research study is new of its kind. Besides, seamlessly integrating the FCM service to the learning scenario is also another important and new approach. The architectural framework of the whole system was framed by integrating those sub-unit models. In the next section, the detail specification of how the general systems' interacting components integrated is also included.

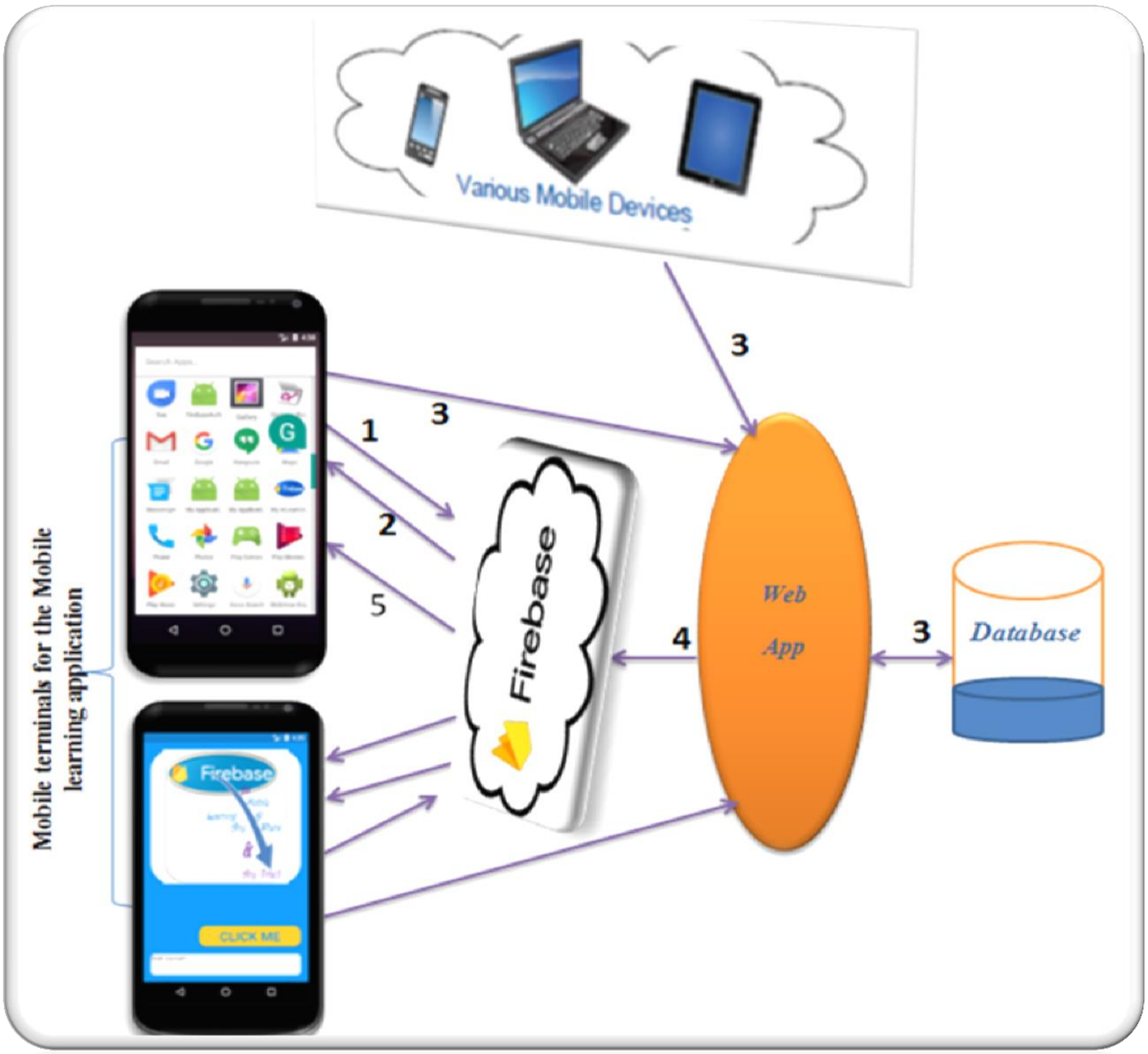

Figure 5.2: Designed system components physical architecture and interaction

5.1.8. Summary of System Components Interactions

The general interaction between the major structural components of the mobile learning application as shown in figure 5.2 takes the following steps

1. The mobile learning app on a mobile terminal will request for Reg_ID from FCM server. This is performed only once when the app runs on a particular device for the first time. The step can be termed as registration process on FCM.
2. The FCM server will send back the unique Reg_ID for the requesting terminal/mobile device.
   It is the granting of the FCM services up on successful registration.
3. The terminal then stores its Reg_ID on a separate database through web app.
   This activity is performed to make the Reg_ID available to the third party application for certain purposes. The web application part of the mobile learning system is accessible by various devices. When someone wants to send a push notification message to a registered mobile terminal from the web application, the Reg_ID of the specified terminal is retrieved from the database.
4. Send the push notification to the FCM server.
   Retrieving the Reg_ID from the database, the web application can attach a notification message to the ID and send to the FCM server.
5. Finally the FCM server will send the push notification to the specified terminal/s.
   The process of delivering the push notification to the targeted device/s is done by the FCM server only. Therefore after receiving the message from the web application in step 4, the FCM pushes the message to the particular device.

5.1.9. Implementation of the Mobile App Interfaces

In order to realize the basic requirement functionalities in the design specification of the earlier section, within the mobile app project package, several android files were included. Basically, in android application development on android studio, app development project structure always takes the MVC (Model View Controller) form. Generally, the prototype of the app was implemented on the android studio by the project package name "***com.example.addisu.mymlearningapplication***" to have all the required interface functionalities. For the implementation of this app, one sample indication is as in figure 5.3. As it can be from the figure, in the project view of "***MymlearningApplication***", there are major source files. Some of these file directories are

- **App**

  This is the core directory which holds all the source code files for the MVC within the **src**→subdirectory. The internal library and ***build.gradle*** is also in this directory. Moreover, the directory contains the included dependency file "***google-services.json***" for synchronizing the app to the cloud service FCM. This file is in ***.json*** format and it was downloaded directly from the firebase console. To facilitate the communication between the android app and the FCM, the ***.json*** file should sync with the android studio library files.

- **External Library**

  This part of the project file is responsible for issues regarding the platform and the SDK. It also contains the detail of associated file and the emulator properties

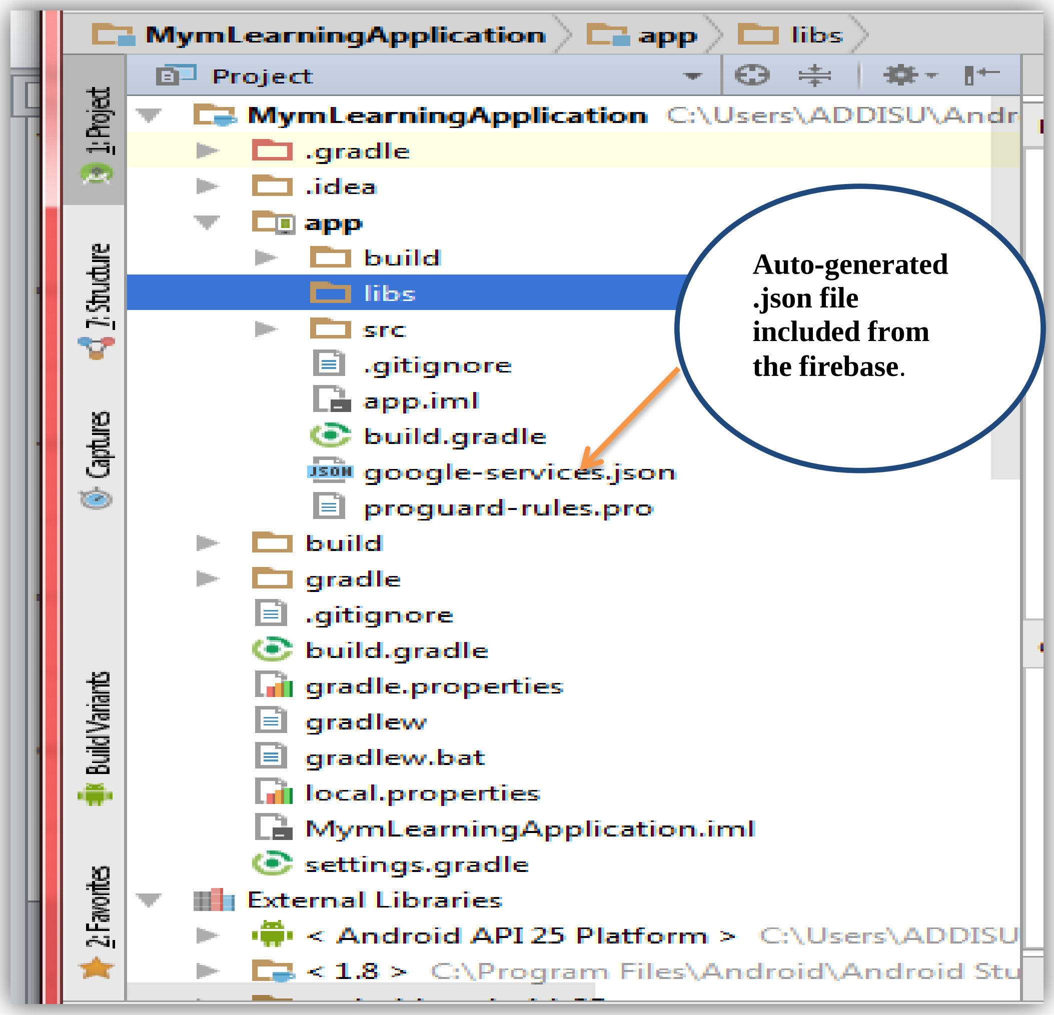

Figure 5:3 sample of the android mobile app implementation.

5.1.10. MVC Structure

In android studio, MVC is the inherent structure for every app project file. This way of organizing the files has several advantages like man aging separation of concerns. In doing so, it is possible to modify one aspect of the project without affecting the other. For example, the view part of the file can be updated or changed separately of the model and control part.

The prototype design of the mobile app in this study followed the MVC project file structure as in figure 5.12. The sample screenshot of the model, view and control files of the app are included below.

- **Model**

The model sub-file is used to specify the attribute and structure of the widget in the screen elements of an activity. The attributes of the widget is scripted as ***.xml*** format in name value pair. The preview layout of the whole element is viewed in design view. For the mobile app in this study, there was couple of model layout. The sample model of the activity that launches the learning content is depicted in figure 5.4(a) in design view and the text view is in figure 5.4(b)

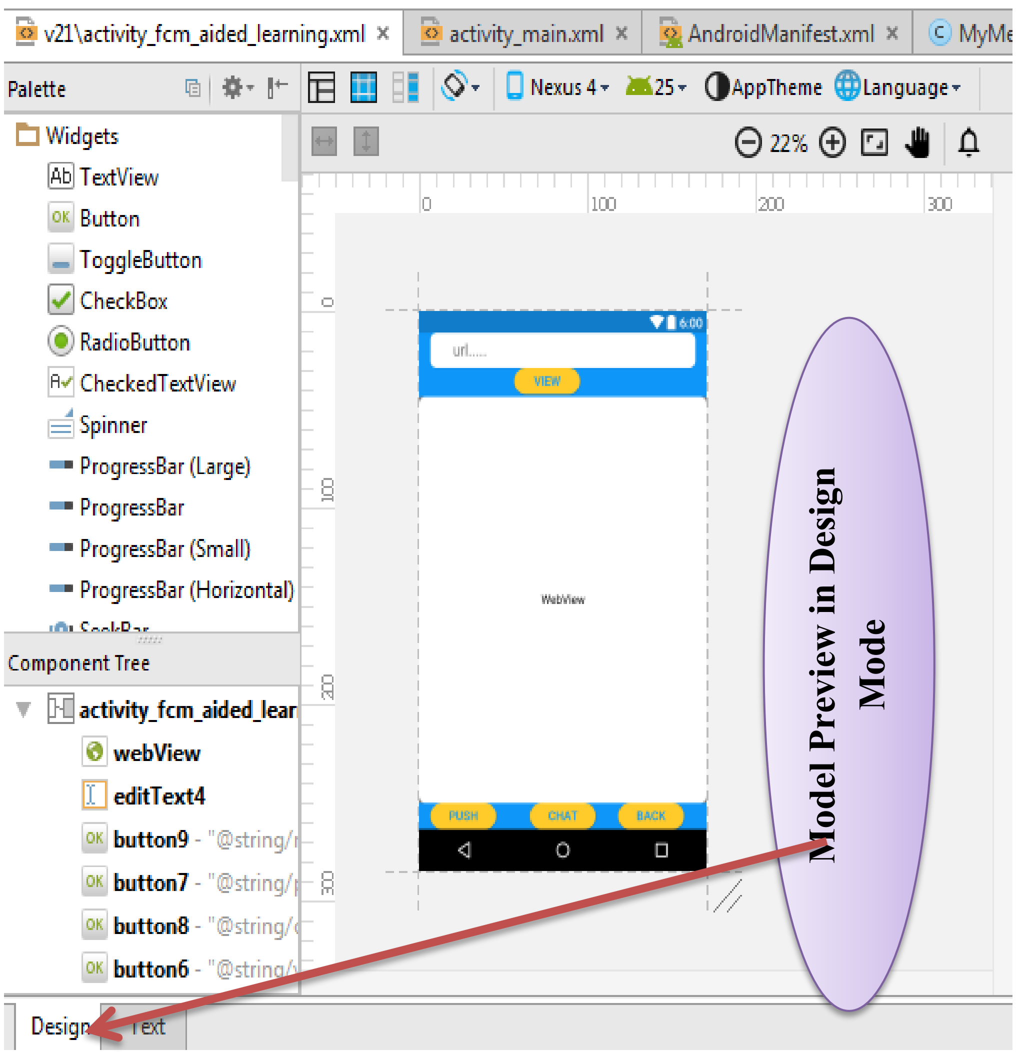

Figure 5.4 (a): The sample of model in design view

```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:id="@+id/activity_fcm_aided_learning"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:background="@color/colorPrimary"
  tools:context="com.example.addisu.mymlearningapplication.fcmAidedLearning">

  <WebView
    android:layout_width="match_parent"
    android:layout_height="500dp"
    android:layout_alignParentTop="true"
    android:layout_alignParentStart="true"
    android:background="@drawable/input_1"
    android:layout_marginTop="73dp"
    android:layout_marginBottom="30dp"
    android:layout_alignParentEnd="true"
    android:id="@+id/webView" />

  <EditText
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:inputType="textPersonName"
    android:ems="10"
    android:id="@+id/editText4"
    android:background="@drawable/input_1"
    android:hint="@string/url"
    android:elevation="1dp"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:paddingRight="30dp"
    android:paddingLeft="30dp"
    android:layout_marginLeft="15dp"
    android:layout_marginRight="15dp" />
  <Button
    android:text="@string/view"
    android:layout_width="wrap_content"
    android:layout_height="30dp"
    android:id="@+id/button6"
    android:background="@drawable/input_bakground"
    android:textColor="@color/colorPrimary"
    android:textSize="15sp"
    android:paddingLeft="5dp"
    android:paddingRight="5dp"
    android:layout_marginLeft="15dp"
    android:layout_marginEnd="21dp"
    android:layout_below="@+id/editText4"
    android:layout_alignEnd="@+id/button8"
    tools:ignore="RtlHardcoded" /> </RelativeLayout>
```

Model Sample for the learning content viewer in text Mode

Figure 5.4 (b): The sample of model in text view

View

It is java class file that is used to define the actions and events we can apply on the widget elements of an activity. In the prototype of this design, a number of view files were included. As it is illustrated in the figure 5.5 below, a sample screenshot of the view file of the ***'fcmAidedLearning'*** activity was taken. The activity is mainly responsible for accessing contents from learning management web app.

```
package com.example.addisu.mymlearningapplication;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.Button;
import android.widget.EditText;
public class fcmAidedLearning extends AppCompatActivity {
  private  Button back;
    private  Button VIEW;
    private EditText Uri;
    private WebView webView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_fcm_aided_learning);
        webView=(WebView)findViewById(R.id.webView);
        WebSettings web= webView.getSettings();
        web.setJavaScriptEnabled(true);
webView.getSettings().setRenderPriority(WebSettings.RenderPriority.HIGH);
        webView.getSettings().setCacheMode(web.LOAD_CACHE_ELSE_NETWORI
        webView.getSettings().setAppCacheEnabled(true);
       // webView.loadUrl("https://www.youtube.com");
        webView.setWebViewClient(new WebViewClient());
        ///////////////////////// BACK BUTTUN//////////////////
        back=(Button)findViewById(R.id.button9);
        VIEW=(Button)findViewById(R.id.button6);
        Uri=(EditText)findViewById(R.id.editText4);
        back.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent myInt= new Intent(fcmAidedLearning.this,MainActivity.class);
                startActivity(myInt);  }});
        //////////////////////////VIEW bUTTON////////////////////////
        VIEW.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String uri ="https://"+ Uri.getText().toString().trim();
                if(TextUtils.isEmpty(uri)){
                    webView.loadUrl("https://www.youtube.com");}
                else {    webView.loadUrl(uri);}}});}}
```

The view Sample

Figure 5.5: The sample of the view part of MVC

- **Controller**

This file is also ***.xml*** file and it is used to control the sequence of execution of the activities. The main xml file of the prototype that launches the application on emulators screen is shown on the figure 5.6.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.example.addisu.mymlearningapplication">
  <uses-permission android:name="android.permission.INTERNET" />

  <application
    android:allowBackup="true"
    android:icon="@mipmap/firebase_123"
    android:label="@string/app_name"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".MainActivity">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
    <activity android:name=".MainMlearnigActivity" />
    <service android:name=".MyFirebaseInstanceIDService">
      <intent-filter>
        <action android:name="com.google.firebase.INSTANCE_ID_EVENT" />
      </intent-filter>
    </service>
    <service android:name=".MyMessagingService">
      <intent-filter>
        <action android:name="com.google.firebase.MESSAGING_EVENT" />
      </intent-filter>
    </service>
    <activity android:name=".fcmAidedLearning"></activity>
  </application>

</manifest>
```

The controller sample

Figure 5.6: the sample of controller of the Learning content management

- **Virtual Devices**

For testing the output display and behavior of the program, the prototype was tasted on a couple of android virtual devices. The app has got its own icon in app menu of the devices as shown below in figure 5.7. Thus, the sample use case interface of the prototype is illustrated in emulator system image. The android emulator was Galaxy_Nexus with API-24 and android version 7.0. The app can also run on any emulator provided that the API is not obsoleted.

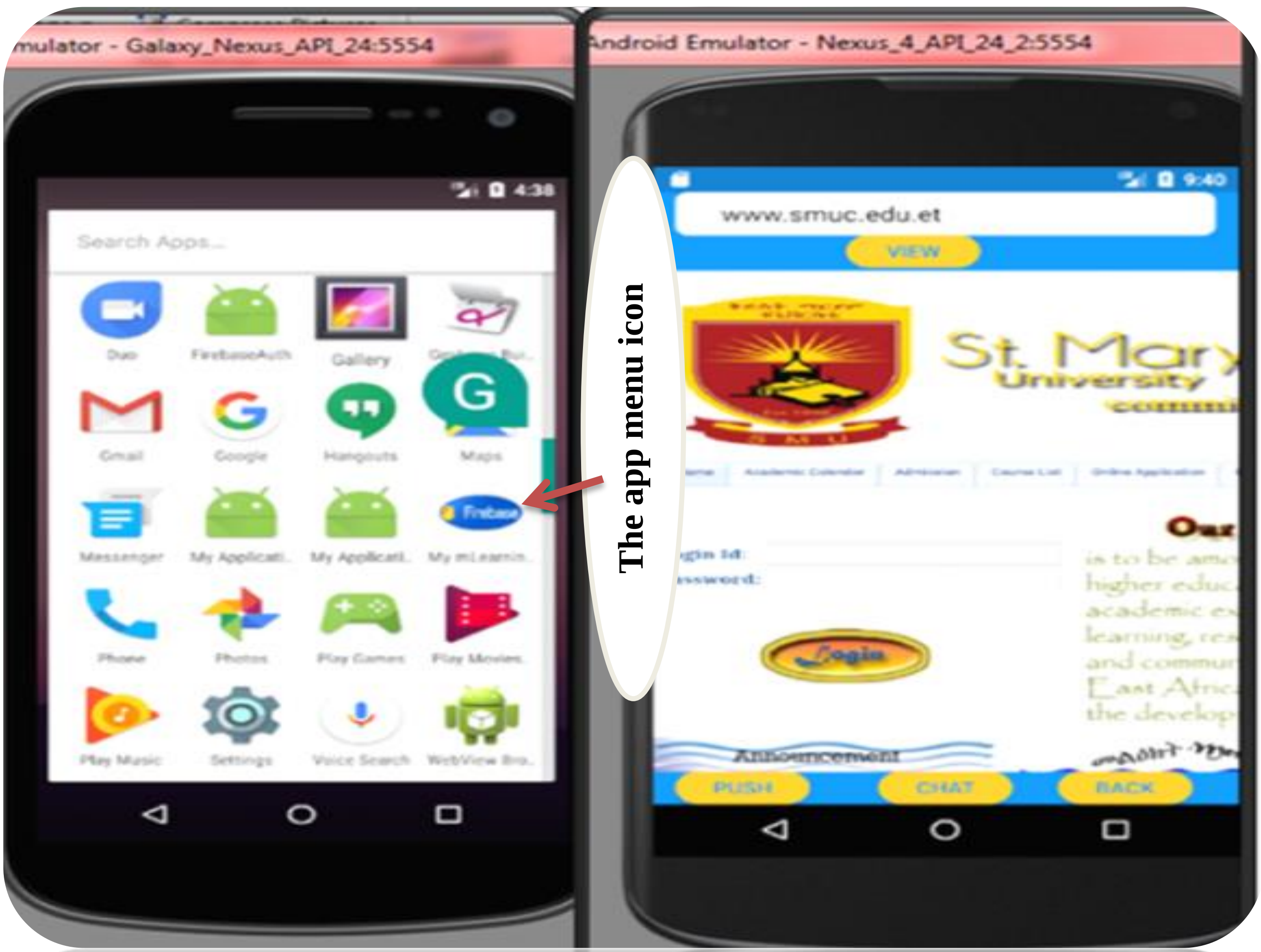

Figure 5.7: The sample of emulators used in accessing mobile learning content management scenario

5.1.11. Implementation of the FCM Integration

To facilitate the integration of the prototyped android studio app, a firebase project was created on the firebase console by the name 'MymlearningApplication' as in figure 5.8(b). Then, the auto-generated ***google_service.json*** file was included to the app file of the project. Finally the necessary dependency file was copied to the ***Gradle*** script file of the project as indicate in figure 5.8(a). To customize the functionality of fire base to the context of the study, a couple of java class files have been added to the app. These are:-

- ***MyMessagingService***

  This is the crucial part the mobile app that is responsible for handling relaying of message and notification between the FCM and specific device using the internet network. For a given mobile device, to receive any update information from FCM server through push notification, it should be active online in Wi Fi or Cellular network. The class extended its super class called ***FirebaseMessagingServic***e and implemented the method as ***super.onMessageReceived(remoteMessage).*** The detail implementation of the way messages and notifications was included in the class. The sample implementation of this service is illustrated in the figure 5.9.

- ***MyFirebaseInstanceIDService***

  In FCM service, every device is identified by string called device token which assigned for all device uniquely up on registration. This token is used as Reg_ID to broadcast notification message the devices. In the case of this study, the Reg_ID is stored in a separate database table at Mobile learning content management server. Therefore, on the android studio mobile app project side, this stuff was modeled to be implemented within this java class file. This class has also extended its super ***FirebaseInstanceIdService*** to override and implement the ***onTokenRefresh( )*** method.

Generally, for the above services to function their respective class packages were imported. In addition, to incorporate the entire associated library files, a number of dependencies file are copied to the Gradle scripts as in figure 5.8(a).

Figure 5.8(a): The sample of firebase dependencies included to the android studio app project

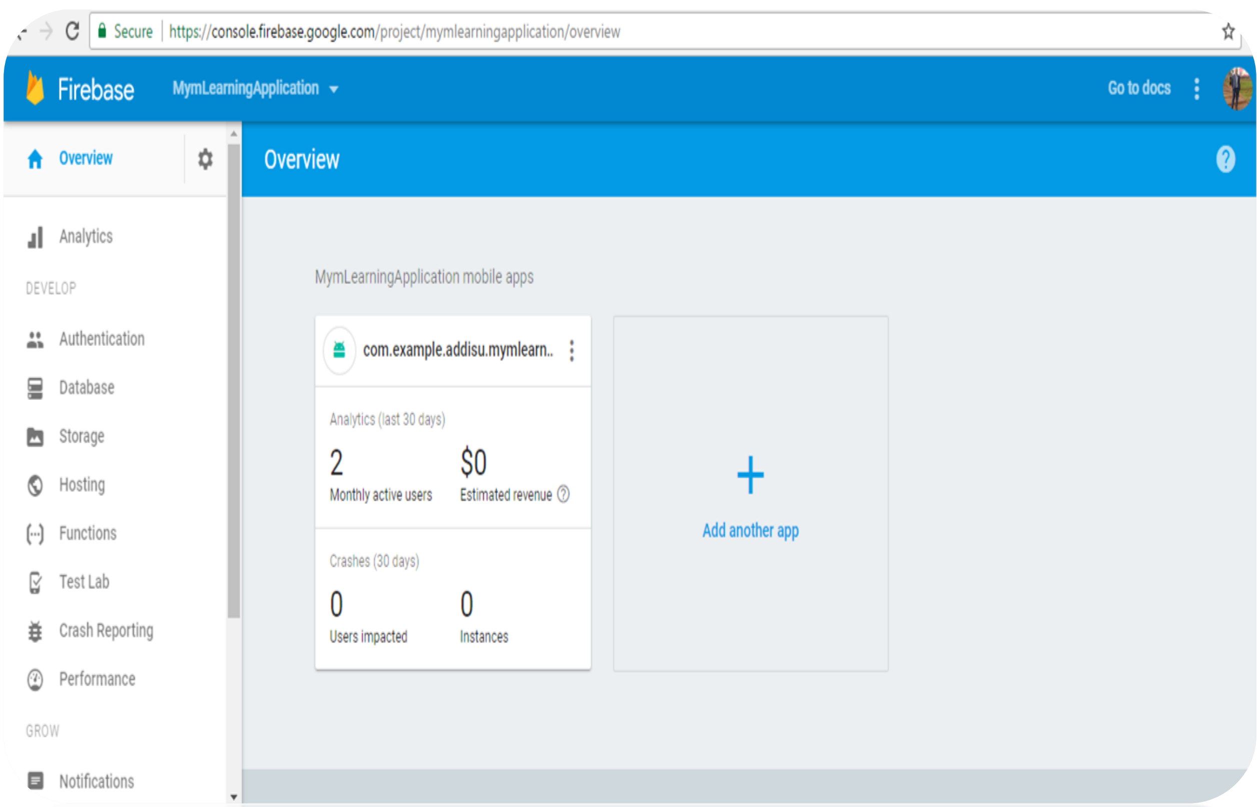

Figure 5.8(b): The scenario of associating firebase project to the android studio app project

```
package com.example.addisu.mymlearningapplication;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.app.Service;
import android.content.Context;
import android.content.Intent;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.IBinder;
import android.support.v7.app.NotificationCompat;
import android.util.Log;
import com.google.firebase.messaging.FirebaseMessagingService;
import com.google.firebase.messaging.RemoteMessage;

import static com.google.android.gms.internal.zzt.TAG;
public class MyMessagingService extends FirebaseMessagingService {
   private NotificationCompat.Builder Notify;
   private static final int x=123;
   public MyMessagingService() { }

   @Override
   public void onMessageReceived(RemoteMessage remoteMessage) {
      super.onMessageReceived(remoteMessage);
      // TODO(developer): Handle FCM messages here.
      receiveMessage(remoteMessage.getNotification().getBody());
         }
   private void receiveMessage(String body) {
      Notify=new NotificationCompat.Builder(this);
      Notify.setAutoCancel(true);
            //@Override
      Intent myInt= new Intent(MyMessagingService.this,MainActivity.class);
      myInt.addFlags(myInt.FLAG_ACTIVITY_CLEAR_TOP);
      startActivity(myInt);
      PendingIntent p=
PendingIntent.getActivity(MyMessagingService.this,0,myInt,PendingIntent.FLAG_ONE_SHOT);
      Uri ur= RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
      Notify.setSmallIcon(R.mipmap.ic_launcher);
      Notify.setTicker("Here");
      Notify.setWhen(System.currentTimeMillis());
      Notify.setContentTitle("The First Notice Checked");
      Notify.setContentText(body);
      Notify.setSound(ur);
      Notify.setContentIntent(p);
      NotificationManager nm=(NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
      nm.notify(x,Notify.build()); }
}
```

Figure 5.9: The firebase implementation on android studio for the case push notification

5.1.12. Web Application

The overall functionality of the mobile learning application needs one more complementary component in addition to the mobile app in the previous subsection. This component is a normal web application that can be accessed by all sorts of networked devices. However, the application has got a special functional property added from the firebase console options as shown in figure 5.8(b). The web application in this study can be considered as single page that is added to the learning management for extending the FCM service.

To assimilate the FCM services to this application, web associated file was included to its source code file from firebase doc. The file was added to the source code of the web application on editor. The editor used was Notepad++ and the scripting language was PHP. Basically, this section is the fundamental part of the study as it is the component responsible for broadcasting notification message from the learning management system to mobile devices.

As per the discussion in the earlier section, every registered device has got a Reg_ID or device token from the firebase server. The Reg_ID is then used for relaying of notification message to a particular mobile device from the FCM server. The process of sending a push notification or message is done only by firebase. Therefore, to dispatch message being on the web application, the device Reg_ID should be stored in database table of a server.

For the backend persistent data storage of the prototype implementation in this study, MYSQL provided by XAMPP Server was used. On the server, a database called '***FCM'*** was created containing '***userfcm'*** table using phpMyAdmin. The table was basically created to store the Reg_ID of the mobile devices. The ID is auto-generated by the FCM when the mobile app runs for the first time on the mobile device. To capture the ID to the database at that instant, the necessary program code was implemented within the ***FirebaseInstanceIdService*** class.

The web application in the subsection had the functionality to read the device Reg_ID from the specified database. In addition to that, the application had the capability to communicate the ID and an attached message to the FCM server. The format of this communication is in JSON object. Once the FCM server has received the message from the web app, it then broadcast it to a particular device based on the ID value. In this manner, it was tried to send push notification message to mobile device (emulator).

5.2. System Evaluation and Validation

The proposed hybrid architecture mobile learning application system was tried to be implemented through a prototype approach. As per the discussion made in the last subsections, some major aspects of the overall system were attempted to be applied for practical works. From a number of specific objectives set at the beginning of this research study, only the core elements were selected for the evaluation of the general system working in the real cases. The selected functional elements are then evaluated and validated based on their own respective cases in the following subsections.

5.2.1. The Case of User Authentication with the Firebase System

One of the main objectives set was seamlessly integrating the Google cloud services with the mobile learning application. In this case the particular service used was FCM. So, based on the implementation in the last section, this functionality was evaluated through authentication. Actually the FCM console can provide different authentications like in Gmail Google++, Facebook etc. But, for this case, only the Gmail case was tested and worked well as in the figure 5.10 below.

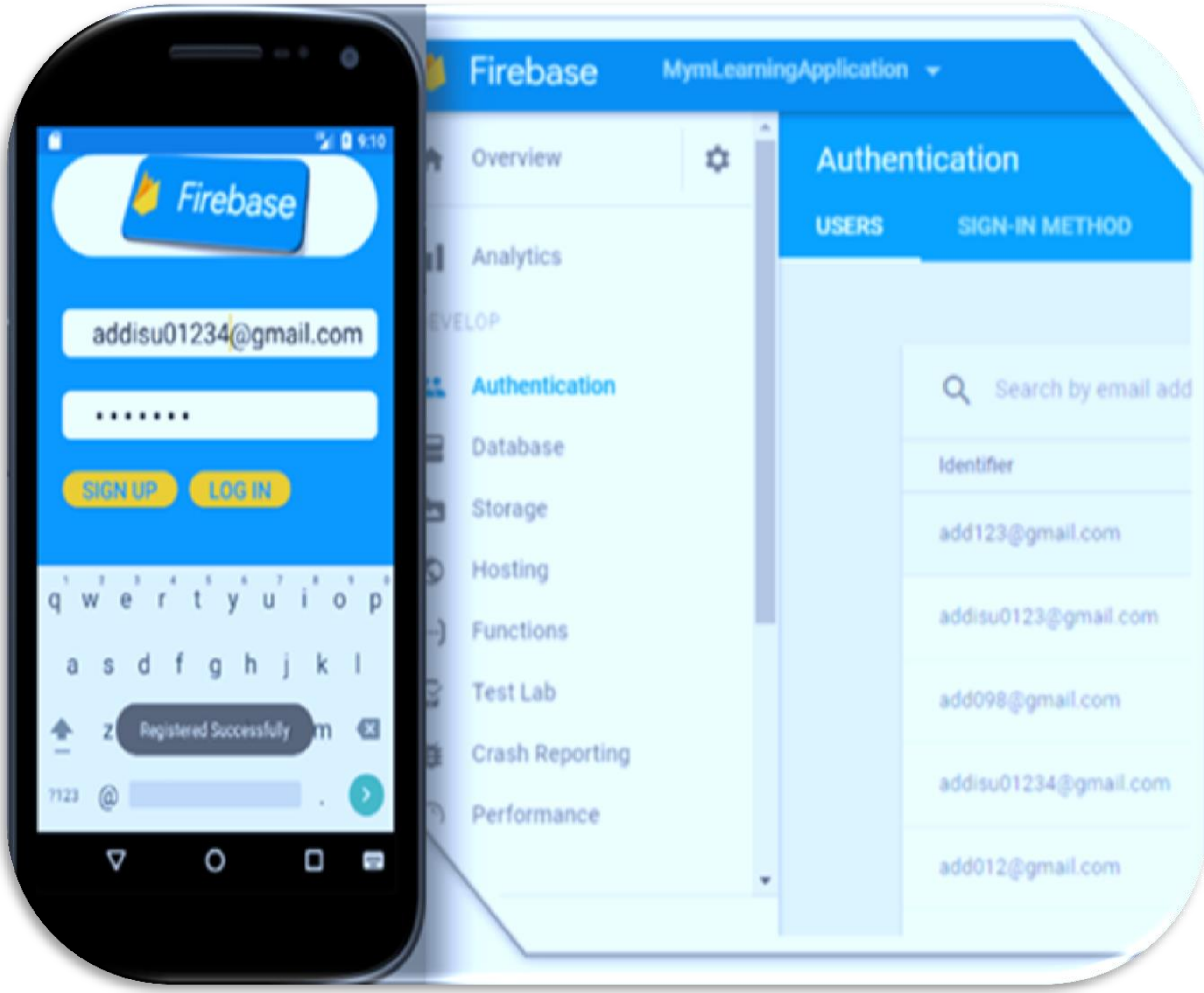

Figure 5.10: A scenario for a user authentication case

5.2.2. The Case of Sending and Receiving Push Notifications

The other core functional element selected for the evaluation of the test case was the scenario of relaying notification message particularly from the server to online mobile devices running the app of mobile learning implemented in the previous section. The proper functionality of this case was tested by sending simple push notification from the firebase server using the console interfaces as illustrated in the figure 5.11. As it can be observed from the figure the activity of sending the notification message from the server has worked fine.

Figure 5.11: A scenario of sending push notification

5.2.3. The Case of Launching Learning Content from Learning Content Management Systems

One more functionality tested as per the primary objective set was the ability of launching learning contents from online learning management systems using the mobile app independent of any browser. In this case, the sample scenario was tested for the case of accessing the contents of some well-known web sites like www.youtube.com, www.facebook.com, and others. Regarding the context of the objective, the sample learning management considered was www.smuc.edu.et . As far as the user experience is concerned, the necessary functionalities like navigation across pages, linking and scrolling have behaved well. The scenario of this case is partially depicted in figure 5.12.

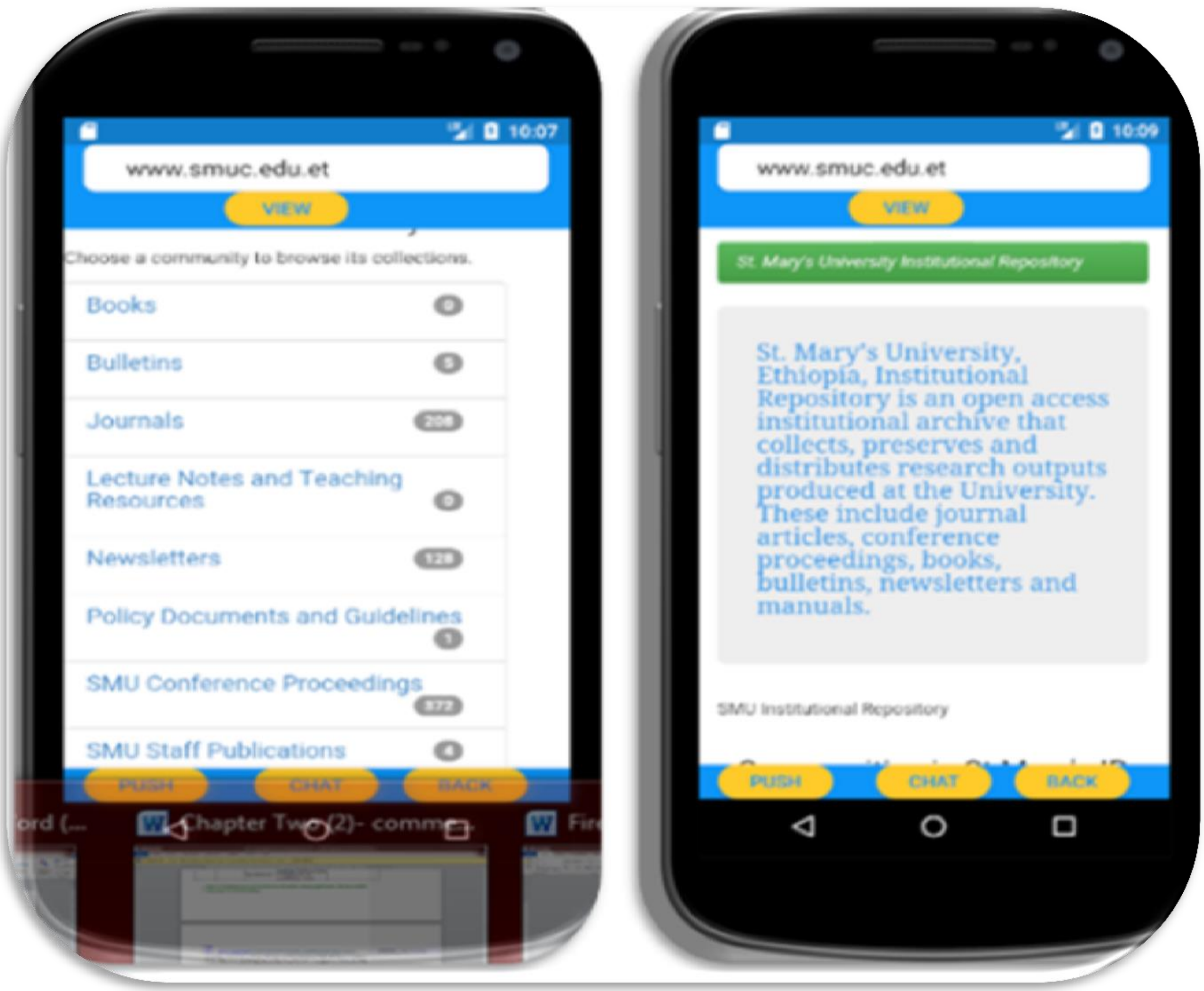

Figure 5.12: A scenario of launching contents from online learning content management.

5.3. Summary

The prototype implementation of the hybrid architecture of mobile learning application has worked well in the selected functionalities as illustrated. The process of seamlessly integrating the android studio app with Google cloud messaging service (FCM) was implemented through user registration and authentication. With respect to the stated objectives, all the functionalities of Sign Up, Log In and Log Out activities have been evaluated and validated to work very fine.

Regarding the process of sending and receiving push notification from the FCM server using the firebase console to the online connected virtual device has also been evaluated and validated to work well. In case of accessing and pulling contents form online sources, the implemented prototype has all the required functionality worked very well. Generally, from scenarios of the above cases the suggested system features are tested to be valid.

From the user perspectives, the system prototype was evaluated in networked environment of one computer laboratory room. The participated users are five experts and 20 postgraduate students in SMU. The networked system consisted of desktop computers and laptop computer.

The processes of evaluation were performed for all the above mentioned scenarios. The cases of accessing learning content from online learning management system was tested only through the laptop as all the tools like emulators and android studio was configured on this device. But, sending of the push notification message was conducted from all the desktop computers through the firebase console. Based on the test cases and the observation of the users, the performance of the system prototype was approved under strong internet connection. However, in a compromised internet connection, delayed performance was observed.

CHAPTER SIX

CONCLUSIONS AND FUTURE WORKS

6.1. Conclusions

In this research process, to achieve the targeted objective by solving the identified problem, several procedural activities were performed to this end. The primary activity was specifying the proper research problem by assessing the available literature works in the perspective of mobile learning. As per the survey made, designing hybrid architecture of the web app and native mobile app was taken as a suitable option for mobile learning application. For enhanced functionality like receiving and sending push notifications, the mechanism of integrating the app FCM cloud services was also incorporated.

Agile process model was followed to deal with all the dynamicity of functionalities in the design and implementation of the proposed architecture of mobile learning application. Tools like Android studio 2.2.3, Notepad++, platform and programming language independent web technologies such bootstrap and firebase console were used for the practical implementation.

As the proposed architecture of mobile learning application is the integration of subcomponents, the architectural organization of each physical component was specified sequentially in agile approach. Then, similarly, the lines of interaction of those components were also identified on the overall system architecture. The communication between the physical components was logically implemented as a prototype for the core functionalities.

Basically, in implementation, the application has two major aspects as learning content management and android application. The android application was developed using android studio and tested on virtual mobile devices or emulators. Then it was verified for the scenario to enable accessing of the learning content from online learning management system. In addition to that the application has had the capability to send and receive push notification message from the cloud messaging service provided by Google. The learning content was a mobile web app and also has the new capability to send push notification message to the cloud messaging service provided by Google.

Generally, from the evaluation of the system functionality in the prototyped implementation, it is possible to say that suggested physical architecture can work. Hence, with respect to the specific objectives set at the beginning of the process, the proposed architecture can be a solution to the stated problem.

6.2. Future works

The primary focus in designing and implementing the hybrid architecture mobile learning application in this study is for the case of android platform only aided by FCM. However, recently Google has introduced FCM as a cross-platform approach for android, iOS, and for web app. For cloud integration of mobile apps this option is very good. Yet, the remaining work will be settling architecture or a mechanism for designing a single system for all platforms.

To come up with the FCM integrated and cross-platform architecture of mobile learning application, the possible option will be

- Designing a plugin architecture that can handle the process of integrating mobile applications with cloud services. The architecture should also deal with the task managing cross-platform related issues.
- Designing a middleware architecture that can handle all the stuffs of making the mobile learning application cloud integrated and cross-platform.

References

[1] J. Rikala, *Mobile Learning – a Review of Current Research*, vol. 35. 2013.

[2] S. Bidin and A. Abu, “Adoption and application of mobile learning in the education industry,” *Procedia - Soc. Behav. Sci.*, vol. 90, no. InCULT 2012, pp. 720–729, 2013.

[3] F. Martin and J. Ertzberger, “Computers & Education Here and now mobile learning : An experimental study on the use of mobile technology,” *Comput. Educ.*, vol. 68, pp. 76–85, 2013.

[4] N. Wang, S. Member, X. Chen, S. Member, G. Song, Q. Lan, and H. Parsaei, “Design a New Mobile Optimized Remote Laboratory Application Architecture for M-Learning,” vol. 46, no. c, 2016.

[5] M. Abarghooei, “Designing a Cross-Platform Mobile Learning System,” vol. 3, no. 3, pp. 3–6, 2015.

[6] S. Malgaonkar, V. Maurya, M. Kulkarni, and G. S. Majithia, “Multipurpose Android Based Mobile Notifier,” pp. 1–4, 2014.

[7] A. Sharma, P. Eastham, and F. Nerieri, “Designing an Energy- Efficient Cloud Messaging Service for Smartphones,” pp. 84–88, 2014.

[8] Y. S. Bhosale and J. L. L. M, “Mobile Cloud Computing for M-learning Applications,” vol. 3, no. 2, pp. 2800–2804, 2014.

[9] A. Des and S. Journal, “A Cloud Based M-learning Architecture for Higher Education,” no. January 2013, 2017.

[10] I. Journal, I. M. Technologies, M. Shuib, U. Sains, S. N. Azizan, T. Gunasegaran, U. Sains, A. Evaluation, R. Towards, M. English, C. P. Subject, P. Speaking, S. Needs, E. Majors, U. Sains, and T. Gunasegaran, “Designing an Intelligent Mobile Learning Tool for Grammar Learning (i-MoL),” no. January, 2015.

[11] J. Haag and P. Berking, “Design Considerations for Mobile Learning,” pp. 1–15, 2014.

[12] M. Al-emran, H. M. Elsherif, and K. Shaalan, “Computers in Human Behavior Investigating attitudes towards the use of mobile learning in higher education,” *Comput. Human Behav.*, vol. 56, pp. 93–102, 2016.

[13] O. Viberg and Å. Grönlund, “Computers & Education Cross-cultural analysis of users ’ attitudes toward the use of mobile devices in second and foreign language learning in higher education : A case from Sweden and China,” *Comput. Educ.*, vol. 69, pp. 169–180, 2013.

[14] K. T. Martono and O. D. Nurhayati, “IMPLEMENTATION OF ANDROID BASED MOBILE LEARNING APPLICATION AS A FLEXIBLE LEARNING MEDIA,” vol. 11, no. 3, pp. 168–174, 2014.

[15] A. Jesús, I. López, I. L. Zorrozua, E. S. Ruiz, and M. Rodríguez-artacho, “Design and development of a responsive web application based on scaffolding learning,” no. April, pp. 308–313, 2016.

[16] S. Lee, T. Han, and Y. Jeon, “Design of Small Sized Mobile Lecture Support System,” vol. 73, no. Fgcn, pp. 7–10, 2014.

[17] J. Zbick, M. Jansen, and M. Milrad, “Towards a web-based framework to support end-user programming of mobile learning activities,” 2014.

[18] Bahaa I. Kazem and Mazin Rahman Ali, “Design and Implementation of Experiments with Real-Time Shared Architecture using Different Mobile Systems,” *iJOE*, vol. 11, no. 2, pp. 9–19, 2015.

[19] I. Korkmaz, S. K. Metin, A. Gurek, C. Gur, and C. Gurakin, “A cloud based and Android supported scalable home automation system,” *Comput. Electr. Eng.*, 2014.

[20] P. Li, Y. Chen, T. Li, R. Wang, and J. Sun, “Implementation of Cloud Messaging System Based on GCM Service,” 2013.

[21] S. H. Chung and E. T. Khor, “Development of Interactive Mobile-Learning Application in Distance Education via Learning Objects Approach,” pp. 373–380, 2015.

APPENDICES

Appendix A: JSON File

One of the crucial aspects of this study was integrating android mobile app with Google cloud messaging service that is FCM. This activity was done by creating one specific project on the firebase console for the mobile learning application of this case. The main objective of this step was to download a specific .json file that has to be included to the app file of the android studio project. Basically this file contains the necessary information about the android studio project on the firebase side as in below


```
{
 "project_info": {
  "project_number": "153864025966",
  "firebase_url": "https://mymlearningapplication.firebaseio.com",
  "project_id": "mymlearningapplication",
  "storage_bucket": "mymlearningapplication.appspot.com"
 },
 "client": [
  {
   "client_info": {
    "mobilesdk_app_id": "1:153864025966:android:2c1eddf6fa44ab85",
    "android_client_info": {
     "package_name": "com.example.addisu.mymlearningapplication"
    }
   },
   "oauth_client": [
    {
     "client_id": "153864025966-
3vr2tq5e83s7u7jaci99j22a79slpegq.apps.googleusercontent.com",
     "client_type": 1,
     "android_info": {
      "package_name": "com.example.addisu.mymlearningapplication",
      "certificate_hash": "7a34cba2cd70ae38448606f55a1caf73353227c7"
     }
    },
    {
```

```
        "client_id": "153864025966-ltohjn1jf81vpfdq78urteki83js0s8b.apps.googleusercontent.com",
        "client_type": 3
      }
    ],
    "api_key": [
      {
        "current_key": "AIzaSyCZlf4PFG9TidebpSx5mBsrAgmR6udH6Lg"
      }
    ],
    "services": {
      "analytics_service": {
        "status": 1
      },
      "appinvite_service": {
        "status": 2,
        "other_platform_oauth_client": [
          {
            "client_id": "153864025966-ltohjn1jf81vpfdq78urteki83js0s8b.apps.googleusercontent.com",
            "client_type": 3
          }
        ]
      },
      "ads_service": {
        "status": 2
      }
    }
  }
 ],
 "configuration_version": "1"
}
```

Appendix B: Dependencies

The other most important file included to the android studio app for the purpose of firebase integration was the dependencies file. The file is added to the app level Gradle script file of the application project file. The main purpose of this file is to integrate the necessary packages to be imported the app. Finally, for the proper functionalities, the dependency file is synchronized through command with the whole gradle script file of the project file. The major content of the file is as below.

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 25
    buildToolsVersion "25.0.3"
    defaultConfig {
        applicationId "com.example.addisu.mymlearningapplication"
        minSdkVersion 18
        targetSdkVersion 25
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
```

```
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:25.3.1'
    compile 'com.google.firebase:firebase-auth:10.2.6'
    compile 'com.google.firebase:firebase-messaging:10.2.6'
    compile 'com.squareup.okhttp3:okhttp:3.2.0'

    testCompile 'junit:junit:4.12'
}
apply plugin: 'com.google.gms.google-services'
```

Appendix C: Firebase Cloud Messaging

Implementing the firebase cloud messaging service provided by Google was the core concern of this study. This activity is used to deal with the relaying of notification messages between the android app and the cloud part. To capture such functionality to the mobile learning application in this case, a specific service should be implemented. In the following code the content of this service file is illustrated.

```java
package com.example.addisu.mymlearningapplication;
import android.util.Log;
import com.google.firebase.iid.FirebaseInstanceId;
import com.google.firebase.iid.FirebaseInstanceIdService;
import java.io.IOException;
import okhttp3.FormBody;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.RequestBody;
import static com.google.android.gms.internal.zzt.TAG;

/**
 * Created by ADDISU on 5/29/2017.
 */
public class MyFirebaseInstanceIDService extends FirebaseInstanceIdService {
   @Override
   public void onTokenRefresh() {
      // Get updated InstanceID token.
      String refreshedToken = FirebaseInstanceId.getInstance().getToken();

      // TODO: Implement this method to send any registration to your app's servers.
      sendRegistrationToServer(refreshedToken);
   }
```

```
    private void sendRegistrationToServer(String refreshedToken) {
        OkHttpClient client = new OkHttpClient();
        RequestBody body = new FormBody.Builder()
                .add("Token",refreshedToken)
                .build();
        Request reuest=new Request.Builder()
                .url("http://192.168.56.1/htdocs/fcm/fcmtest.php")
                .post(body)
                .build();
        try{
            client.newCall(reuest).execute();


        }
        catch(IOException e){
          e.printStackTrace();
        }
    }

}
```