



ANDROID PLATFORM FOR THE MOBILE APPLICATION SECURITY SYSTEM

Mir Humaira Jan

Research Scholar, CMJ University Meghalaya

Dr. Shaik Abdul Nabi

Professor Head of Department Computer Science & Engineering

AVN Institute of Engineering & Technology Hyderabad

Abstract

The aspect of mobile application security, the Android platform, which is one of the mobile operating systems that is utilised the most all over the world, presents both potential and obstacles. Because mobile devices are becoming more and more integrated into personal, professional, and financial domains, it is of the utmost importance that applications that run on Android contain adequate security measures. The purpose of this study is to investigate the most important security features and vulnerabilities of the Android platform, with a particular emphasis on the platform's capacity to protect user data and guarantee secure application behaviour. It takes a look at the different layers of security that Android offers, such as app sandboxing, encryption, and secure communication protocols, as well as the problems that are presented by issues such as malware, app permissions, and hazards associated with third-party app stores. In addition to this, it investigates the role that security frameworks, such as Google Play Protect and the app permission model of Android, play in reducing risks and improving overall security. For the purpose of developing secure mobile applications, the research identifies best practices that developers should adhere to. These best practices include secure coding methods, user data protection, and regular security upgrades respectively. The findings of this study highlight the significance of adopting a comprehensive security strategy in order to safeguard users against newly emerging dangers and guarantee the integrity of mobile applications that run on the Android platform.

Keyword: Mobile Technology, Android platform, Google Play, App Store, Mobile Application

INTRODUCTION

Mobile improvement is illustrated by a multiplicity of applications with dissimilar excellence requirements. Online application stores, like the Apple App Store and Google Play, offer thousands of market-oriented apps—mobile games, utilities, navigators, social networks, and clients for web capital. At the same time, the interest in critical mobile applications is growing. For instance, online banking has evolved into mobile banking, mobile social alerts are extensively used to report accidents or warning about hurricanes, and special apps exist to monitor traffic and help cardiac patients. Increased actuality apps are used for complex direction-finding by using an assortment of sensors. An original tendency is to use elegant phones as components for mobile cyber-physical systems since the powerful hardware has a diversity of sensors. Mobile applications are even being measured to support processes at such dangerous facilities as nuclear power plants. These trends need high levels of dependability and excellence for mobile software systems. They influence testing, the whole mobile improvement process in general. Thus the mobile improvement process ends with the submission of a social application to an online store.

The plan is to gain a wider consultation of users in a shorter time, but this does not assure the quality of the produce and non-critical bugs are typically established. Some surveys have established that mobile developers typically deal with small apps and do not adhere to a formal progress process. In dissimilarity, a completely dissimilar approach is necessary for dangerous or business-critical mobile applications, including mobile clients for dependable venture systems and solutions; for example, Facebook's iOS app is crucial for maintaining the company's shape and standing and thus is rebuilt to overcome the deprived superiority.

To guarantee these mobile applications' dependability and safety, enough testing is required on a diversity of varied devices as well as on different OS. Android enlargement is the most envoy example of how dissimilar applications should purpose amid a plethora of hardware-software combinations. Sufficiently testing all of these platforms is too expensive—perhaps not possible—particularly for small resource-constrained mobile improvement companies. Mobile improvement has a set of characteristic challenges and features. Mobile application testing has some similarity to website difficult as both engage corroboration in many environments. The universal requirements for both types of testing are similar: applications should function properly, professionally, and be reliable and secure in all environments. Though, mobile testing presents new activities and requires more effort because it includes web applications that work within mobile browsers or hybrid variants wrapped in native code.

OBJECTIVE

1. Study on increase subtleties inside the middle vicinity with the guide of the application.
2. Analysis via experts of mobile app improvement and project managers had been interviewed for taking the response to opinions to use ISM and specific variants of DEMATEL techniques

PRINCIPLES OF MOBILE APPLICATION

Transportability: The mobile computing environment is able to accommodate a group of devices thanks to this facility. There is a possibility that these tactics have insufficient device capabilities and a restricted power supply; yet, they should have sufficient meeting out capabilities and physical portability to work in environments that exhibit mobility.

Accordance: It is tied to the smallest possible amount of downtime, without being exacerbated by moves.

Interactivity: In a mobile computing system, the nodes that are located in the appropriate locations are linked with one another so that they may communicate with one another and engage in active information exchanges.

TECHNOLOGY IN MOBILE APPLICATION

In recent times, the open-source Linux operating system has been modified to be available for use on portable computers of varying hardware configurations. Additionally, several manufacturers are expecting that Linux will be used for their portable PCs. Considering that this operating system does not need a license fee, it has the potential to further reduce the cost of portable portable computers. In developing nations like as India, voice-activated PCs based on Linux have been created for mass customisation in rural regions where the computing connections are insufficient. A user is able to receive information about nearby restaurants, theatres, coffee shops, maps, driving directions, traffic, weather, news, tourist sites, and other such things via the use of location aware mobile computing, which is also an essential aspect of the technology.

A wide variety of wireless system connection options are also available for PCs that are being held in custody. A significant number of portable computers are equipped with infrared ports that may be used to replace information with a system or another

computer that is located within a short range. A significant number of them are able to connect to wireless local area networks (LANs). A number of cellular telephone service providers are also in the process of developing mobile phone modules that can be attached to the expansion slots of portable PCs. There are also some portable computers that are capable of supporting Bluetooth, which is a new wireless standard for personal area networking.

Wireless technologies, such as 802.11b, also known as Wi-Fi, Infrared Data Association (IrDA), Ultra-Wideband Radio (UWB), and Home Radio Frequency (RF), are being used to technologies that are comparable to Bluetooth, with varying degrees of success. 802.11 is the most well-known networking technology, with the exception of Bluetooth. Both of these technologies utilise the same radio frequency, which indicates that they are not well-matched since they produce interference with each other. There are airports all across the United States that have adopted the 802.11 technology, and three of the most well-known airlines in the United States have shown their support for its implementation.

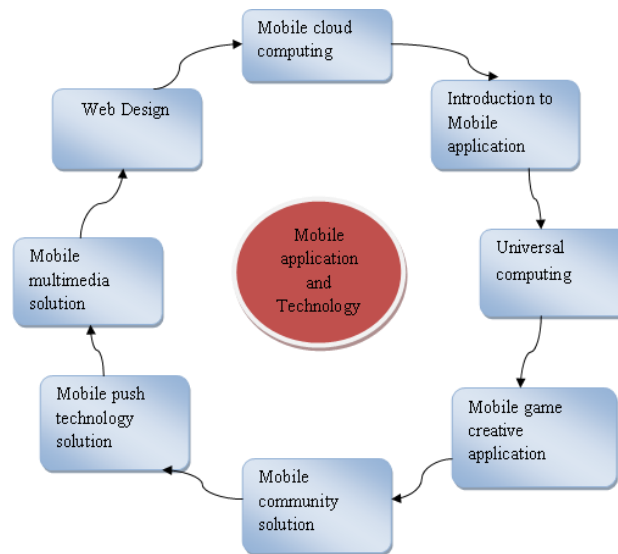


Fig. 1 Utilising Technology in Mobile Applications

When compared to Bluetooth, the data connection provided by infrared is much lower. There are several restrictions associated with it, such as the fact that it can only communicate with one another point-to-point, that it requires a line of sight, and that its speed is fifty-six kilobytes per second, while Bluetooth's speed is one megabyte per second. When compared to Bluetooth, the Ultra-Wideband Radio is superior since it is able to communicate over longer distances while using just half of the power that Bluetooth does.

ADVANTAGES FOR MOBILE APPLICATION

Users may now work from any location as long as there is a documented connection thanks to position elasticity. It is possible for a user to work without being at a certain place. Their mobility ensures that they can fulfil their specified functions and do many tasks at once. The amount of time spent commuting to and from work or from other places has been drastically reduced. Now, one may work as if they were on their computer and get access to all the important papers and data via a limited channel or gateway. In many businesses, it has made telecommuting better. Additionally, it has reduced unnecessary purchased expenditures. Because of the increased productivity, users may work effectively and professionally from whichever posture they are happy with. They become more efficient as a result.

Mobile computing has made it possible to broadcast entertainment that started with audio and video recordings while on the move. With the development and availability of high-speed data connections at significant expense, it is simple to access a vast array of films and instructional and informative materials,

allowing one to enjoy all the entertainment they want while browsing the internet for streaming data.

FEATURES OF THE ANDROID PLATFORM

The SDK tools are now available for download, and the platform has been officially introduced. At the moment, there is just one cell phone that offers the Android operating system. As seen in Figure 5, the executive Android website states that the architecture of the platform is comprised on the following four fundamental characteristics:

Application essentials

There is a teaching language called Java that is used to build Android apps. Nevertheless, it is essential to commit to memory that they are not performed by the Java Virtual Machine (JVM) that is often used. Instead, Google has developed a specialised virtual machine (VM) that it calls Dalvik. This VM is in charge of transforming Java byte code and doing the execution of it. Before any custom Java classes can be performed inside an Android operating system, they must first be turned into a Dalvik well-matched coaching set. It is the responsibility of the Dalvik Virtual Machine to merge the Java class files that have been created into one or more Dalvik Executable (.dex) files. It does this by reusing and substituting information from a large number of class files, which effectively reduces the minimum amount of space required for a standard.jar file by fifty percent. Dalvik was designed to maintain the characteristics of lightweight mobile operating systems, which is required due to the limited hardware capabilities of mobile devices in comparison to desktop computers or laptops whose capabilities are more predictable.

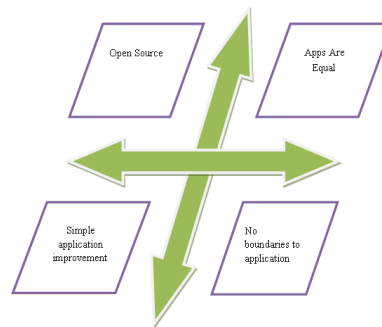


Fig. 2The Android platform's four main characteristics

Android Platform overview

An operating system, middleware, and essential apps are all components of the Android software stack, which is designed for use on mobile devices. The Android provides the tools necessary to begin developing applications on the Android platform using the Java training language. The scheme services such as safety, memory management, process organization are prohibited by Linux. Figure 4 depicts the structural architecture of the android.



Fig. 3 Mobile computing essentials

Rising Android Applications

The Android operating system offers a comprehensive collection of application programming interfaces that are up to date and in good health. The services that are provided by the Android smartphone core scheme are completely open and accessible to any apps. Android apps are able to exchange data with one another and make consistent use of shared resources on the system when they are given the appropriate permissions. There is a teaching language called Java that is used to build Android apps.

Application structure: Through the provision of an open development platform, Android gives developers the opportunity to create apps that are extraordinarily rich in content and imaginative in thought. The tool hardware, information on the location of admission, the ability to run background services, the ability to set alarms, the ability to add alerts to the status bar, and a great deal more are all available to developers without restriction. To the furthest extent possible, developers are given access to the same structural application programming interfaces as are used by the main programs. Because the application's structural design is intended to simplify the usage of components, any application may publish its capabilities, and any other application can then utilise those capabilities. This allows for the application to be used by any other application. Through the use of this same device, the user is able to replace individual components.

Android Runtime: There is a collection of nucleus libraries that are included with Android. These libraries provide the majority of the functionality that can be found in the core libraries of the Java training language. With each Android submission, the Dalvik practical machine is executed in its own unique manner, and each Android submission follows its own method. A technique that is capable of knocking down a large number of virtual machines (VMs) has been developed using Dalvik. The Dalvik Virtual Machine carries out file operations using the Dalvik Executable format, which is displayed with a memory footprint that is small. It is a register-based virtual machine (VM) that executes programs that have been built by a Java compiler and then converted into the .dex format using the "dx" tool that is included in the package. Underlying functionality, such as strand and low-down-stage memory organisation, is provided by the Linux kernel, which is used by the Dalvik virtual machine environment.

PROPOSED SYSTEM FOR COATED APPROACH ANDROID APPLICATION

The coated technique has the potential to be helpful for the development of Android applications. The enhancement of web-based applications is one possible use for this. Figure 8 illustrates the overall strategy that was used for the optimisation of the Android application. Only the HTTP layer, which is responsible for sending HTTP get and post requests to the server and receiving the response, is considered to be the lowest level of the hierarchy. The API layer immediately follows. In order to formulate the question and send it to the HTTP layer, this is for the purpose of processing the answer that was received from the server. Following the retrieval of the response string from the HTTP layer, the API layer proceeds to parse the cord.

In addition to this, it assists in the removal of the necessary fields and then transmits them to the data layer. Components that involve the design of business levels and the implementation of features like as caching, outstanding organisation, logging, and corroboration are included inside the Generic Data layer. Taking the data from the API layer and putting it to use is the next layer, which is the platform-needy data layer. The manner in which it keeps the data is depending on the platform. On the platform, some classes, such as Adapter and List view, are responsible for storing the necessary data. The user interface layer is the last one.

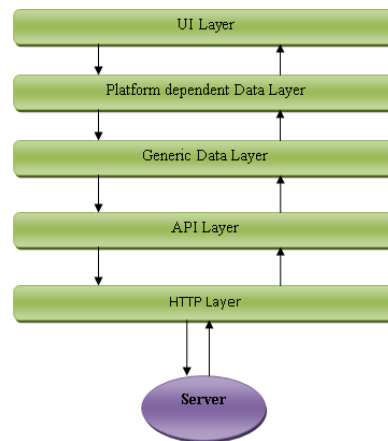


Fig 5 Coated Approaches

Application model: In Android's application paradigm, each application is comprised of a collection of mechanisms, each of which may be created and executed according to the requirements of the situation. Components may be broken down into the following categories:

The basis of the client boundary is comprised of activity components; in most cases, this means that each window of the program is restricted by some movement. Service components operate in the background and remain active even if windows are switched between or between windows. The interfaces that allow for communication with other programs may be exposed via services.

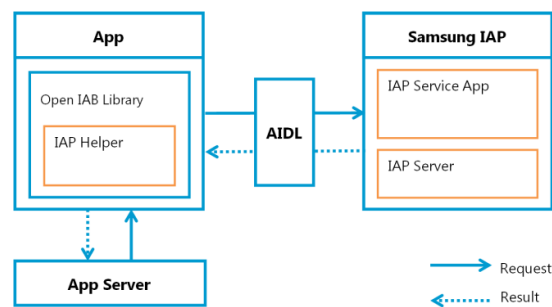


Fig. 6 Android application model

Non-synchronous responses are provided by receiver mechanisms in response to communication from other applications. Data that is relevant to the application is often stored in a file via the techniques that a provider uses. These kinds of data may be shared across different programs. Consider, for example, an application for viewing photos online that is compatible with a phone that runs Android. There may be a great deal of components in this application. There is a performance function that allows users to examine the images stored on their phone in the form of a list or a network. There is a possibility that there will be the test for downloading a picture in the background. There may be receivers that are able to receive an application when a call is received, and there may also be receivers that may resume the application after the call is over. The program should not put undue burden on the high-priority functions of the device, such as the incoming call, the incoming text message, the notification that the battery is low, and so on. Last but not least, there is a possibility that there is a provider that stores the images and the information of the photos on the phone.

Android SDK in Activity class

There is a base class in the Android Software Development Kit (SDK) for every kind of constituent. This base class has callback methods that are applied to the linked component at different stages during its life cycle. There is a life cycle unique to each element. In order to create each component of an application, one must first extend one of the basic classes and then replace the methods currently included inside that class. In a careful manner:

- There are methods in the movement class that are executed whenever an activity is formed, whenever an activity calls another activity, or whenever an activity returns to the activity.
- There are methods in the Service class that are executed whenever the service is launched, or whenever any component binds to this service, or even when both of these things are grouped together.
- It is the responsibility of the Receiver class to execute a method whenever a message is sent to this receiver.
- The data that is saved by this provider may be deleted, queried, and updated using the services provided by the Provider class.

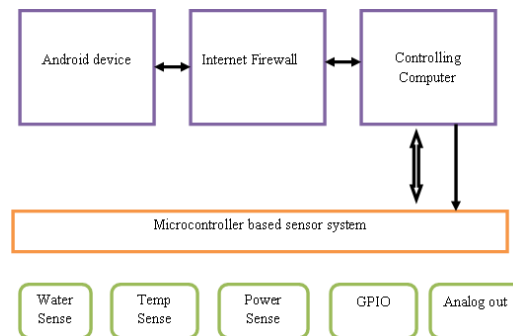


Fig. 7 Android SDK

Component classes and methods

It is safe to say that the Google Android mobile phone platform is among the most reliable operating systems for electronic devices. The usage of smart phones may serve as an alternative to computers and laptops. There are many more opportunities for the development of unique apps as mobile devices continue to improve their capabilities. In the current cellular market, the amount of new growth in mobile application development has reached its maximum point. When it comes to mounting mobile apps, Android introduces a new structure that is based on components. This framework allows for each program to be composed of a different number and kind of components. The activity components serve as the foundation of the user interface; each screen that the user may access is a different activity.

The background processing that service components perform continues long after the application that they are associated with has lost attention. Information is shared between the various components of the Content Provider in relational database format. The relational database capability provided by Android includes SQLite as an integrated component. As an example, the system consists of an application that is equipped with a Content Provider that is dedicated to sharing the user's contact book with other apps so that they may query it. Broadcast Receiver mechanisms, as a last point of discussion, provide the function of an asynchronous mailbox for messages coming from the system and other applications. As a total, this application architecture allows for a variable degree of collaboration across apps, which is necessary in situations where addition may be as simple or as complicated as the situation calls for.

PERFORMANCE ANALYSIS

According to the World Mobile programs Market - Advanced Technologies, Global Forecast, there are around sixty percent of programs (free, paid, and ad supported) that are downloaded on their own, which results in revenues that are at the lower end of the spectrum. As time went on, other market participants like as Android, Google, and Play store began to establish a marketplace for themselves inside the mobile applications industry, which coincided with the development of the smart phone market. The findings of the market study indicate that the market for mobile applications will continue to be competitive, as it is anticipated that the number of applications that are downloaded will continue to rise. The profits generated by mobile applications (on a global scale) will also see a parallel expansion.

The number of mobile app videos, display and other, search and social, as well as other types of videos, has significantly increased as a result of the growing popularity of mobile apps. It has also been noted that there has been an increase in the number of mobile app developers that have an audience that is comprised of more than twenty percent of active monthly users. Furthermore, this increase in the number of app developers shows that there are numerous original applications being merged into the ones that are currently in existence.

Table 1: Mobile android app market number of range statistics

Mobile Platform		Number of range calculated		
Mobile Android Apps	Year	Video	Display and others	Search/Social
Google Play Store	2014	2.5	2.0	2.35
Google search	2015	3.0	3.2	2.56
Google Maps	2016	6.5	1.8	3.21
Amazon App Store	2017	4.5	2.8	5.21

Mobile Application Using Software Development process

Through the process of software testing, software testers are able to identify flaws in the program and eliminate them, ultimately leading to an improvement in the quality of the product. In recent times, software testing has grown more widespread and riskier among software development businesses. Testing software may be carried either manually or automatically, depending on customer preference. The process of manually inscribing test cases and carrying them out without the assistance of any tool is known as manual testing. Manual testing is a method of testing in which a tester executes the testing process by carefully travelling through the many interfaces of the system that is being tested, testing with different input standards, noting the findings of the experiments, and comparing them to the outcomes that were expected from the tests. The use of an automated testing tool is necessary in order to finish the automated testing process. When compared to manual testing, the automated testing tool offers a computer-controlled testing experience that is much superior. The test cases are carried out by the testing tool in order to evaluate the functionality and compatibility of the program that is being tested.

The purpose of automated testing is to reduce the amount of human effort that is required, similar to what is done in manual testing; nevertheless, it does not completely remove the need for manual testing. Mobile platforms are gaining popularity all over the globe as a result of the availability of a large variety of applications to consumers on handheld and portable devices. Additionally, testing is being used as a method of quality assurance for mobile applications now. This is the reason why a great number of tools are being developed and deployed. There has previously been an analysis and comparison of these tools with regard to their distinctive capabilities, supported platforms, code reporting, and overall competency. On the other

hand, the automated testing tools that are now available for mobile applications have not been assessed and compared for the many quality features that they may enhance in the applications that are being tested.

On mobile devices, such as smartphones and tablets, software programs known as mobile apps are designed specifically for usage on these devices. An application is often referred to as an app store, but after it has been produced, it is sold via an application allocation platform. It is the market that drives app development. Comparable to market-driven software that is more conservative, the requirements for an application are usually derived from the business objectives that have been determined or from the opportunities that are available in the market. During the course of an app's development, application developers have limited opportunities to interact with potential consumers. The number of downloads and the amount of money made from the app are the metrics that are used to evaluate success. The concept of app stores has helped to democratise the software business, making it possible for almost anybody to create and sell applications to consumers that are located all over the globe via app shops. There will be significant difficulties associated with the settlement of app shops. The market for mobile applications is very competitive and congested, and as a consequence, an application may fail (get few or no downloads) owing to aspects that are unrelated to its functionality and usefulness. These elements may include the app's name, its icon, or the amount of disclosure it provides customers. In order for an application to be successful, it should preferably be beneficial to a large number of users all over the globe. This is because the profit limitations from app sales are rather low (Section 1.2). As a consequence of this, two different goals for exploration are developed, and they are as follows:

1. To evaluate various testing tools for mobile applications with the primary goal of determining the quality characteristics that these tools help to accomplish in the applications that are being tested.
2. To use automated testing methods in order to determine the general trend of important quality factor attainment when it comes to mobile applications that are being tested.

Online Quotes Analysis

During the first phase, one may establish the online quotations that are made available by firms on the internet. The objective of this phase is to extract an original set of topic with solution. According to the findings of the survey, every business should have a website and provide a web-based form for potential clients to fill out in order to get a price about the expansion of mobile applications. In order to conduct the study, the researcher made use of a standard search tool known as Google-Scraper1, which is freely accessible and serves as an open resource.

Review with Expert

In this stage of the research project, the purpose is to conduct an analysis of the behaviour and consultation, as well as a semi-structured survey, by an expert who has a thorough understanding of the development of mobile applications. The objective is to cultivate the concerned professionals so that they can identify and organise issues and the potential solutions to those problems.

Conclusion

The process of developing mobile applications is the primary focus of that study. We are using mobile application devices for the sake of conducting research and experiments. On a number of different dimensions, three different kinds of solutions are presented. In the first solution, the Android Platform for the Mobile Application Security System is discussed. This is a coated technique for improving Android applications. It has been claimed that this may be used for the enhancement of web-based applications. In the second recommendation, the process of developing mobile applications using software programming is discussed. This method proposes that the user has to afford precise username and password which is provided at the time of registration, if login achievement, it means it will take up to major side else or it will remain in the login page itself. Third suggestion discusses the Mobile Application Framework for Software Engineering. The previously mentioned distress area from the Mobile Application Security viewpoint is spoken about in the part that came before this one. With a complete advance, it is probable to moderate an discussion these risks to a greater amount.

References

1. Dejan Kovachev, Yiwei Cao, Ralf Klamma, “Mobile Cloud Computing: A Comparison of Application Models, Information Systems and Database Technologies”, RWTH Aachen University.
2. Milos Stojmenovic, “Mobile Cloud Computing for Biometric Applications”, Department of Informatics and Computation, 15th International Conference on Network-Based Information Systems, Melbourne, 26-28 Sept. 2012, pp 654-659, Print ISBN: 978-1-4673-2331-4, DOI: 10.1109/NBiS.2012.147.
3. Debessay Fesehaye, Yunlong Gao, Klara Nahrstedt, Guijun Wang, “Impact of Cloudlets on Interactive Mobile Cloud Applications”, IEEE 16th International Enterprise Distributed Object Computing Conference, Beijing, 10-14 Sept. 2012, pp 123-132, Print ISBN: 978-1-4673-2444-1, DOI:10.1109/EDOC.2012.23.
4. M.T Nkosi, F. Mekuria SM, “Cloud Computing for Enhanced Mobile Health Applications”, 2nd IEEE International Conference on Cloud Computing Technology and Science, Indianapolis, Nov. 30 2010-Dec. 3 2010, pp 629-633, Print ISBN: 978-1-4244-9405-7, DOI: 10.1109/CloudCom.2010.31.
5. Peng Jianjing, Shen Shenglin Fang, “Application of Mobile Cloud Computing in Operational Command Training Simulation System”, IEEE 12th International Conference on Computer and Information Technology (CIT), Chengdu, 27-29 Oct. 2012, pp 532-535, Print ISBN: 978-1-4673-4873-7, DOI: 10.1109/CIT.2012. 116.
6. Ricky K. K. Ma, King Tin Lam, Cho-Li Wang, “eXCloud: Transparent Runtime Support for Scaling Mobile Applications in Cloud”, International Conference on Cloud and Service Computing (CSC), Hong Kong, pp 103-110, Print ISBN: 978-1-4577-1635-5, DOI: 10.1109/CSC.2011.6138505.
7. Saeid Abolfazli, Zohreh Sanaei, Abdullah Gani, ”Mobile Cloud Computing: A Review on Smartphone Augmentation Approaches”, First International Conference on Computing,

Information Systems, and Communication, May 2012, Singapore.

8. Yan Gu, Verdi March, Bu Sung Lee, "GMOCA: GreenMobile Cloud Applications", First International Workshop on Green and Sustainable Software (GREENS), Zurich, 3-3 June 2012, pp 15-20, Print ISBN: 978-1-4673-1833-4, DOI: 10.1109/GREENS.2012.6224265.
9. Han Qi, Abdullah Gani, "Research on Mobile Cloud Computing: Review, Trend and Perspectives", Second International Conference on Digital Information and Communication Technology and it's Applications (DICTAP), Bangkok, pp 195- 202, Print ISBN: 978-1-4673-0733-8, DOI: 10.1109/DICTAP. 2012.6215350.
10. Byung-Gon Chun, Sunghwan Ihm, Petros Maniatis, Mayur Naik, "CloneCloud: Boosting Mobile Device Applications Through Cloud Clone Execution", eprint: arXiv:1009.3088, December 2012, DOI: 2010arXiv1009.3088C.
11. Keerthi S. Shetty, Sanjay Singh, "Cloud Based Application Development for Accessing Restaurant Information on Mobile Device using LBS", International Journal of UbiComp, Vol. 2, No. 2, 2011, pp 37-49, DOI: 10.5121/iju.2011.2404.
12. M. Chen, Y. Wu and A. V. Vasilakos, "Advances in Mobile Cloud Computing", Mobile Network Applications, no. 19, (2014), pp. 131-132.