

DEVELOPMENT OF A LOCATION-BASED APPROACHING NOTIFICATION SYSTEM USING ANDROID PLATFORM

Hisham AlMajed and Abdelaziz Khamis

Department of Computer,
Arab East Colleges, Riyadh, Saudi Arabia

ABSTRACT

Mobile application uses and development is a rapidly growing sector. Nowadays mobile devices are more powerful and portable with plenty of useful tools for assisting people handle daily life. The main objective of this paper is to develop a mobile application that solves the problems facing bus drivers and parents when parents do not show up, and when kids wait for a long time. The application also produces the current drivers map to be used for bus fleet management purposes. The application makes use of the location service on Android to specify the current location of the driver, and the Google's cloud to device messaging to push approaching notifications to parents. The application is developed using an Extreme Programming (XP) based methodology that performs the analysis, design, implementation, and testing iteratively.

KEYWORDS

Notification; Mobile application development; Location-based services; Android studio; Android SDK; Software engineering; Extreme Programming.

1. INTRODUCTION

Since the beginning of the so-called smart-phones and their associated mobile software applications or apps, users could experience the functionality of personal computers on pocket-sized devices. These apps are becoming increasingly everywhere in our daily life [1]. Currently, categories of mobile applications include: games, banking, travel, social networking, location based apps, fitness, and medical apps.

The target audiences of many applications have shifted from the use of traditional personal computers to using mobile devices such as smart phones for performing the tasks they want or obtaining the information they seek [2]. This shift has motivated the software engineering community to provide guidance for many issues that are related to the development of mobile applications.

Different mobile operating systems are provided by different mobile companies. Mobile application developers have the choice of developing apps for each platform or a platform

independent app [3]. The process of choosing between developing platform-dependent and platform-independent applications involves many parameters, such as budget, project timeframe, target audience and app functionality.

Numerous development tools have been created to aid developers in building mobile applications. These tools may be classified into two categories. The first category of tools is for platform-dependent app development, while the second category is for cross-platform app development. In [4], the authors introduced a framework for evaluating cross-platform mobile application development tools.

A mobile application is nothing but a software product with a different level of complexity. A SDLC is required to develop high quality software products that meet or exceed customer expectations, and reach completion within times and cost estimates. The phases of the SDLC include: analysis, design, implementation, and testing. Each phase is itself composed of a series of steps, which rely on techniques that produce deliverables [5].

There are many development methodologies that can be used to implement the SDLC. These methodologies vary in terms of the progression that is followed through the phases of the SDLC. Methodology options include waterfall and agile development [6].

In the Saudi community, bus drivers and kids' parents are facing problems when kids do not show up, and when parents wait for a long time. In this paper, a mobile application has been developed to solve these problems. The state of the art technology in mobile application development has been used. In addition, the location-based services and the cloud to device messaging have been investigated to make use of their latest technology in the developed application.

This paper is structured as follows. Section 2 provides the objectives of this project, and presents the project methodology used. The literature review is introduced in section 3. The adopted SDLC methodology is given in section 4. The analysis phase is presented in section 5. The design phase is presented in section 6. The implementation and testing phases are presented in sections 7 and 8. Finally, the conclusion is given in section 9.

2. PROJECT METHOD

2.1. Project Questions

The main project questions of this paper are:

- What are the tools that best support Android platform development?
- What is the best fit methodology that can be used for mobile application development?
- What is the latest technology that can be used for cloud to device messaging?
- What are the latest location-based services that can be used in mobile application development?

2.2. Development Approach

In the next section, we start with a literature study to gather information from different resources to answer the above mentioned project questions. To guarantee the integrity of the information gathered, the literature study will depend on information that will be collected from published academic literature and industry whitepapers.

Using the build methodology, it will be possible to explore the application of the state of the art tools and methodology in mobile application development. The following good practices will be considered:

- Reuse components. Component-based development has been successful in many application domains.
- Test, Test, and Retest. Waiting until after an application has been implemented to uncover any deficiencies can be costly, and time-consuming. To minimize these kinds of problems, user interfaces must be continually tested and refined as development proceeds.

3. BACKGROUND

3.1. Mobile Application Development

There are many software engineering issues that are related to the development of mobile applications [7]. In this section, we will focus only on four issues namely, portability, development tools, and development process.

3.1.1. Portability

An important issue to consider by the mobile application developer is portability, since there are a lot of different platforms available nowadays including: Apple iOS, Google Android and Microsoft Windows.

There are various types of mobile applications: native applications, mobile web applications, and hybrid applications [8, 9, and 10]. A native mobile application is built specifically for a particular platform, using tools provided by the operating system vendor.

The main benefit of a native application is performance; native applications deal directly with the mobile operating system, and make full use of all the functionality that modern mobile devices have to offer [11]. The most critical limitation of native applications is portability; code written for one mobile platform cannot run on another. That is, native mobile applications are not portable.

A mobile web application is developed using web technologies such as HTML, CSS, and JavaScript, and accessed through the mobile device's web browser. Such a browser is in itself a native app that has direct access to the OS APIs, but these APIs are only partially available to web apps or not available at all [11].

A hybrid mobile application is a blended application that combines native development with web technology. A significant portion of such an application is developed in web technology. The native portion enables the hybrid application to make full use of all the features of mobile devices.

Developer can choose between coding the native part of the hybrid app or use ready-made solutions such as PhoneGap that provide a uniform JavaScript interface to selected device features that is consistent across different platforms [12].

3.1.2. Development Tools

Development tools allow developers to write, test and deploy applications into the target platform environment. Each platform has its own development tools. For example, Android Studio IDE is the official tool for the android platform [13]. Android Studio is a powerful open source integrated development environment that provides great features for Android developers.

Before any work can begin on the development of an Android application, the first step is to configure a computer system to act as the development platform. This involves a number of steps consisting of installing the Java Development Kit (JDK) and the Android Studio Integrated Development Environment (IDE) which also includes the Android Software Development Kit (SDK) [13].

Once an Android application has been developed, an Android emulator environment is required to perform a test run of the application. Android Virtual Devices (AVDs) are essentially emulators that allow Android applications to be tested without the necessity to install the application on a physical Android based device [13].

3.1.3. Development Process

The first and foremost reason for low quality mobile application development is that app developers are not conforming to the development life cycle phases [5]. Other reasons include: lack of experience on the app development SDKs, and not enough testing is done. Therefore, app developers need to use a software development life cycle to develop high quality applications.

Choosing a methodology to use by application developers is not a simple task, because no one methodology is always best. The methodology selection criteria include: clarity of user requirements, familiarity with technology, application complexity, application reliability, short time schedules, and schedule visibility [6].

Various existing SDLC methodologies have been adapted to mobile application development [14]. Suitability and contribution of some effective and commonly used agile methods has been discussed [15]. An appropriate agile method could be selected for a given project and can be tailored to a specific requirement based upon project complexity, and time schedule [16].

3.2. Location-Based Services

Location-Based Services (LBSs) are services that make use of the geographic location of an object. LBSs have two major types: Position Aware Services and Location Tracking Services

[17]. The first type of LBSs aims to provide some useful information within the environment around the location of an object (i.e. Restaurants, Banks, Parks, etc.). It answers questions like: Where am I? And where is the nearest point of interest? [18].

The second type of LBSs aims to provide the user's location to another party to track his/her locations (For example, Parcel shipment). These services rely on others' locations and keep tracking it to provide the location information to the recipient. These services answer questions like: Where is my object of interest? And how do I get there? A survey on location based services, which includes their history and generations, exists in [19].

3.3. Cloud to Device Messaging

Many of the mobile applications rely on remote services on the cloud. Google Cloud Messaging (GCM) is one of such services which allow developers to send push messages to Android devices. Here, GCM acts as proxy server in between the android client and server [20].

Understanding the performance of GCM is essential for time sensitive applications, such as fire alert and instant messaging. An evaluation of GCM shows that the GCM message arrival latency is unpredictable. That is, a reliable connection to the Google's GCM servers does not guarantee a timely message arrival [21].

Recently, Google announced that it has acquired Firebase, a backend service that helps developers build real time mobile applications for iOS, Android and the web [22]. Using firebase cloud messaging (FCM), you can send notification messages to drive user reengagement and retention [23].

4. THE ADOPTED SDLC METHODOLOGY

Agile Development methods are considered to have the potential to help deliver enhanced speed and quality for mobile application development [24]. These methods focus on the iterative development of applications. They break the SDLC into smaller iterations to reduce the risk and allow the development to adapt rapid modification. Examples of Agile Development methodologies include Scrum, and extreme programming (XP).

In this paper, an XP-based methodology will be adopted. The main features of this methodology include the close interactions with end users and continuous testing. After a shallow planning process, the developer performs the analysis, design, implementation, and testing phases iteratively and incrementally, as shown in figure 1. Iterations used in the proposed system are: client-server connectivity, maps drawing, and notification messaging.

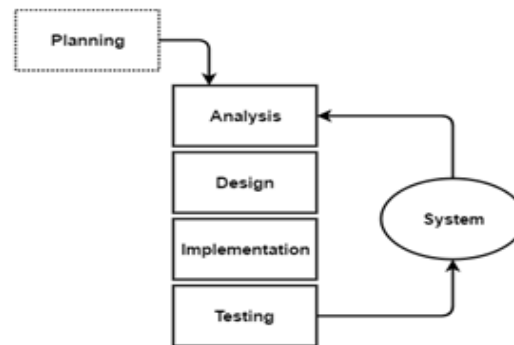


Figure 1. Extreme Programming Methodology

5. ANALYSIS PHASE

The purpose of the analysis phase is to express what the system should do by drawing process models and data models. For interactive applications, the first activity in the analysis phase is developing use cases as a means of expressing user requirements.

In this paper we present one of the use cases required for the “client-server connectivity” iteration that aims to create the application connectivity with the database. The selected use case, as shown in figure 2, creates a journey for the bus’s driver to all kids’ locations. The primary actor is the driver who launches the application and press the journey creation button.

| | | | |
|--|-----------------|--------------------|------------------------|
| Use case name: Create joureny | | ID: 1 | Importance level: High |
| Primary actor: Driver | | | |
| Short description: The driver creates a journey from the stored client's locations in the client DB. | | | |
| Trigger: The driver clicks the fetch locations button Type: External | | | |
| Major Inputs: | | Major outputs: | |
| Description | source | Description | Destination |
| Driver ID | Driver | Clients' locations | Client DB |
| Journey request | Driver | Created journey | Driver |
| Current location | Location module | | |

Figure 2. Create Journey Use Case

The second activity in the analysis phase is process modelling as a means of describing the business processes. In this paper we present the context diagram that shows the entire system in context with its environment. In addition, we present a DFD fragment corresponding to the use case “Create journey”.

The context diagram of the proposed system is shown in figure 3. It shows the entire system as just one process and shows the data flows to and from external entities.

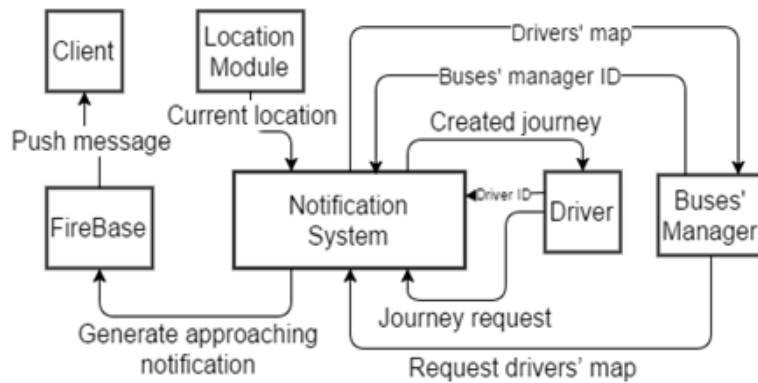


Figure 3. Notification System Context Diagram

The context diagram is decomposed to a more detailed process model called level 0 DFD that contains a fragment for each use case. The DFD fragment corresponding to the “Create journey” use case is given in figure 4.

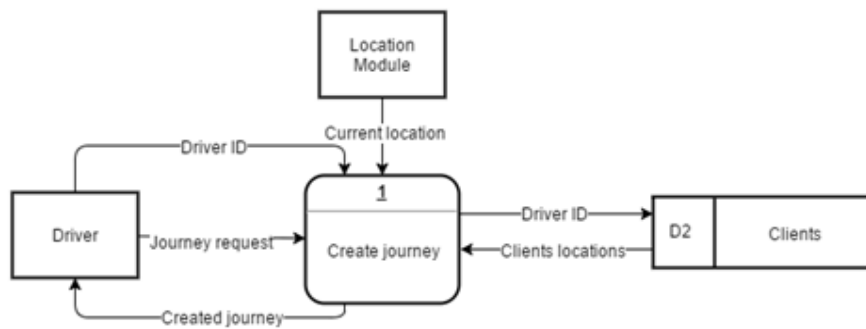


Figure 4. DFD Fragment for the “Create journey” Use Case

The third activity in the analysis phase is data modelling as a means of describing the data that are used and created by a business system. It demonstrates people, places, or things about which information is captured and how they are related to each other. The logical data model of the proposed system is shown in figure 5.

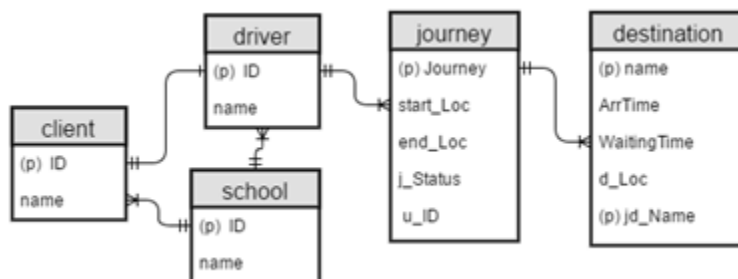


Figure 5. Notification System Logical Data Model

6. DESIGN PHASE

The design phase describes how the system will work. The deliverables of this phase include: architecture design, user interface design, and data storage design. Architecture design describes the system's hardware, software, and network environment.

In the proposed system, the client-server architecture is used, as shown in figure 6. The client is a native Android application that is responsible for the presentation logic. The server is responsible for the data access logic and data storage. PHP is used to connect the client application to a MySQL database via a Web service.



Figure 6. Notification System Architecture Design

In the proposed system, the user interface is designed to be pleasing to the eye and simple to use. It is also designed to minimize the amount of effort needed to accomplish tasks. Figure 7 shows a high fidelity prototype of the main screen of the proposed system. The Android Studio Material Design is used in designing the interface.

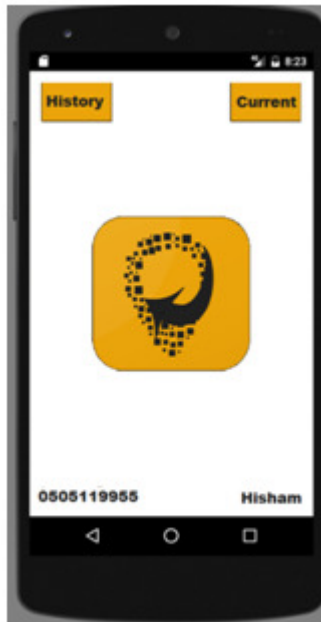


Figure 7. Notification System Main Screen

A driver clicks the button in the middle to request Clients' locations and starts the journey. A small icon will be in the notification bar to indicate that a journey is ongoing and list number of destinations left.

In the second iteration of the system development, Google maps are used to determine the best route to the clients' locations. The driver launches the Google navigator to start the journey. The locations of the clients on Google maps are shown in figure 8.

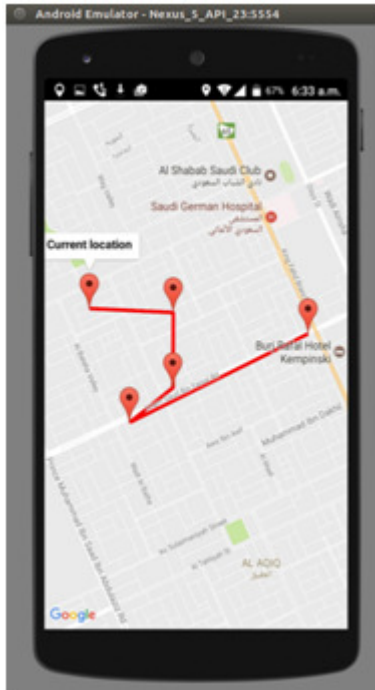


Figure 8 Clients' locations Map

Another activity of the design phase is designing the data storage component of the system. The logical data model will be converted into a physical data model. The physical data model of the proposed system is shown in figure 9.

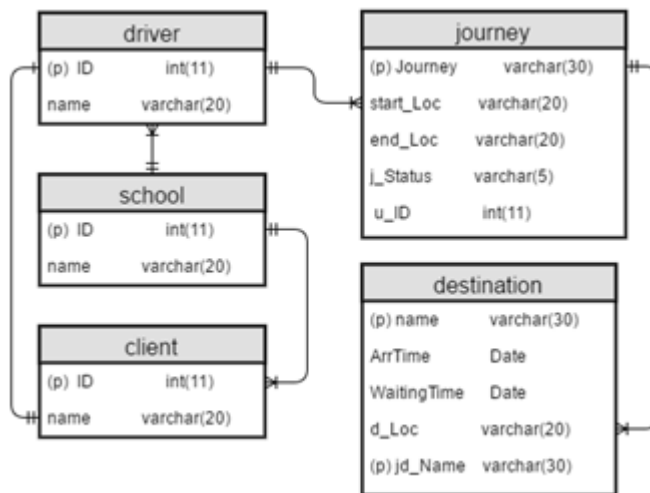


Figure 9. Notification System Physical Data Model

7. IMPLEMENTATION PHASE

The main activity of the implementation phase is writing programs to build the system. The proposed system is built using Android Studio version 2.2. For the Server side, POST and GET provided by PHP version 5.6 are used to send and receive data from the database. PHP code is hosted and accessed using the domain “approaching.halmajed.sa” defined directly in the client side to connect the application to the database. Finally, the database is constructed using MySQL version 5.7, and accessed using phpMyAdmin version 4.6.4.

For the “client-server connectivity” iteration, the Volley package is used to connect the application to the PHP server. Volley package is ready made library make Android networking easy to implement. For the “Notification messaging” iteration, Firebase API is used to notify client of arriving driver. Finally, for the “maps drawing” iteration, the Google Maps API is used to draw drivers’ locations for busses’ management department and to draw the clients’ locations for busses’ drivers.

8. TESTING PHASE

A program is not considered finished until it has passed its testing. For this reason, programming and testing are tightly coupled. Unit testing is done after an iteration programming. Integration testing is done after integrating a new iteration with the system.

Another important type of testing is the so called usability testing. Usability is a qualitative attribute that assesses how easy user interfaces are to use. This type of testing is done early in the development process. A prototype of the user interface is developed and used to get the users’ feedback. Figure 10 shows screens of the initial user interface prototype.



Figure 10. UI Prototype used for Usability Testing

In the usability testing three questions are considered. 1. Does the application logo indicate its purpose? 2. Is the application easy to use? 3. Is the application pleasing to the eye? After collecting answers, the overall user satisfaction is calculated. The percentage of the users who are satisfied with the initial prototype was only 40%. Therefore, the UI prototype is redesign and the usability test is repeated. This process is repeated until we got the improved results shown in figure 11. Screens of the revised user interface are shown in figure 12.

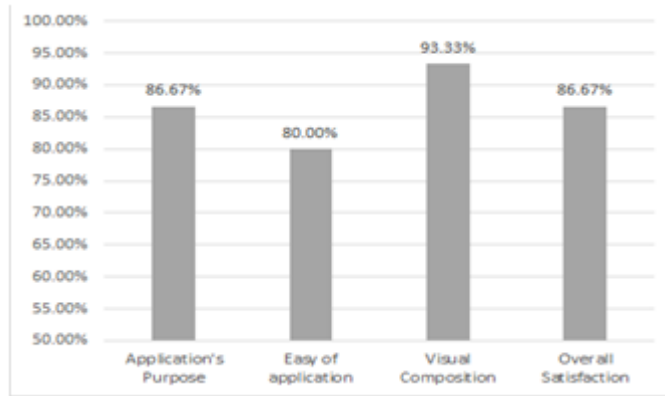


Figure 11. Results of Usability Testing

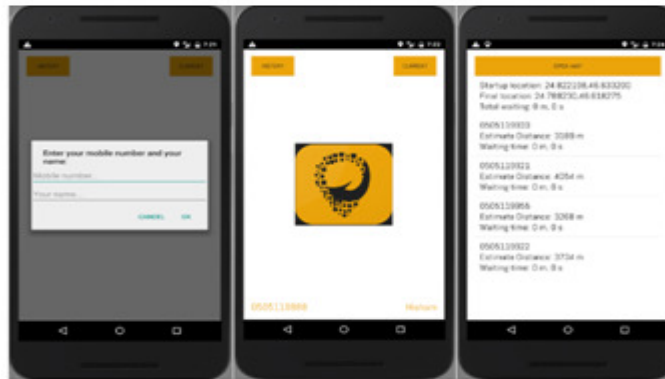


Figure 12. Revised UI Prototype

The final type of tests is done during an actual use of the application. The application is uploaded to Google play store as a beta version. Then, some selected users are invited to try the application in a real situation.

The application is integrated with Firebase to collect application logs. Figure 13 shows a statistical report produced from the Firebase console for the last 30 days. It indicates that there are 5 errors during the specified time frame, and only one user is impacted. It also indicates that these errors fall into 4 clusters (error categories).

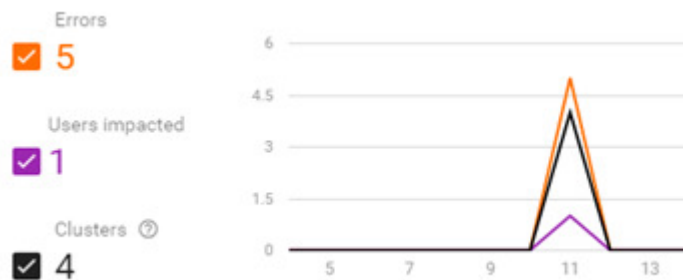


Figure 13. Application Log Report for the Last 30 Days

In addition, Firebase gives details about all error in the log file. An example of the details about an error is shown in figure 14. “NullPointerException” indicates that the root cause of the error is a null value is passed to the function. While “destEst (Destination.java:90)” indicates the class, function name and line number that cause the application to crash.

| Cluster | Stack trace |
|---|---|
| java.lang.NullPointerException Location.java - Line 434 Fatal | android.location.Location.distanceTo (Location.java:434) sa.halmajed.approaching.Destination.destEst (Destination.java:90) |

Figure 14. Error Details Example

9. CONCLUSIONS

The good practices that have been considered in our development process include reuse components and continuous testing. The reused components include third party packages and created codes. The third party packages that have been used include: Firebase package and Volley library. On the other hand, the created code for reuse include ListView adapter that is used to list the destinations of the current journey. Reuse of components reduce the error and increase the quality of mobile applications.

The second good practice is continuous testing. Impact of ignoring testing until the finishing of the application development can be frustrating, costly, and time-consuming. Therefore, during the design phase, the user interface was continually tested and refined before it is implemented. Unit and integration testing is done for each iteration of the application. Final testing is done during an actual use of the application, and Firebase console logging.

REFERENCES

- [1] T. Rakestraw, R. Eunni, and R. Kasuganti, “The mobile apps industry: A Case Study”, Journal of Business Cases and Applications, Volume 9 - September 2013.
- [2] L. Williamson, “A Mobile Application Development Primer: A Guide for Enterprise Teams Working on Mobile Application Projects”, IBM Whitepaper, 2012.
- [3] M. Curran, N. McKelvey, K. Curran, and S. Nadarajah, “Mobile App Stores”, IGI Global, 2015.
- [4] S. Dhillon and Q. Mahmoud, “An Evaluation Framework for Cross-Platform Mobile Application Development Tools”, Softw. Pract. Exper, 2015.
- [5] V. Inukollu, D. Keshamoni, T. Kang, and M. Inukollu, “Factors Influencing Quality of Mobile Apps: Role of Mobile App Development Life Cycle”, International Journal of Software Engineering & Applications, Vol. 5, No. 5, September 2014.
- [6] A. D. Wixom, B. Wixom, and R. Roth, “Systems Analysis and Design”, John Wiley & sons, 2012.
- [7] A. Wasserman. “Software Engineering Issues for Mobile Application Development”. Proceedings of the FSE/SDP workshop on Future of software engineering research. November 7-8, 2010.

- [8] A. Charland and B. Leroux. "Mobile Application Development: Web vs. Native", Communications of the ACM 54.5, 2011.
- [9] S. Avinash and P. Anandkumar, "Implementation of Cross-Platform Mobile Application using PhoneGap Framework", IJCSE, Voli. 3, Issue 3, May 2014.
- [10] IBM Whitepaper, "Establishing an Effective Application Strategy for your Mobile Enterprise", 2012.
- [11] IBM Whitepaper, "Native, Web or Hybrid Mobile-App Developments", 2012.
- [12] J. Wargo "PhoneGap Essentials Building Cross-Platform Mobile Apps", Pearson Education, 2012.
- [13] N. Smyth, "Android Studio Development Essentials", eBookFrenzy, 2015..
- [14] A. Kaur, and K. Kaur, "Suitability of Existing Software Development Life Cycle (SDLC) in Context of Mobile Application Development Life Cycle (MADLC)", International Journal of Computer Applications, Vol. 116, No. 19, April 2015.
- [15] A. Khalid, S. Zahra, and M. Khan, "Suitability and Contribution of Agile Methods in Mobile Software Development", I. J. Modern Education and Computer Science, Vol. 2, 2014.
- [16] H. Flora and S. Chande, "A Review and Analysis on Mobile Application Development Processes using Agile Mythologies", International Journal of Research in Computer Science, Vol. 3, No. 3, 2013.
- [17] I. Junglas and R. Watson, "Location-Based Services", Communications of the ACM, Vol. 51, No.3, 2008.
- [18] A. Kushwaha and V. Kushwaha. "Location Based Services using Android Mobile Operating System", International Journal of Advances in Engineering & Technology, 2011.
- [19] M. Mohammadi, E. Molaei, and A. Naserasadi. "A Survey on Location Based Services and Positioning Techniques". In: International Journal of Computer Applications, Vol. 24, No.5, 2011.
- [20] C. Tamilselvi and B. Kumar," Cloud to Device Messaging with Voice Notification Using GCM", Proceedings of the World Congress on Engineering and Computer Science 2015 Vol I, WCECS 2015, October 21-23, 2015.
- [21] Y. Selim, B. Aydin, and M. Demirbas, "Google Cloud Messaging (GCM): An Evaluation", Symposium on Selected Areas in Communications: GC14 SAC Internet of Things, Globecom 2014.
- [22] F. Lardinois, "Google Acquires Firebase To Help Developers Build Better Real-Time Apps", <https://techcrunch.com/2014/10/21/google-acquires-firebase-to-help-developers-build-better-realtime-apps/>, Posted Oct 21, 2014
- [23] "Firebase Cloud Messaging", <https://firebase.google.com/docs/cloud-messaging/>
- [24] F. Harleen, S. Chande, and X. Wang. "Adopting an agile approach for the development of mobile applications." International Journal of Computer Applications Vol. 94, No.17, 2014.