Självständigt arbete på avancerad nivå

Independent degree project – second cycle

Huvudområde

Master of Science with a major in computer Engineering

Title

Influencing Factors on Quality of Experience (QoE) in Mobile Computing

Adnan Akbar

Mittuniversitetet
MID SWEDEN UNIVERSITY

Campus Härnösand Universitetsbacken 1, SE-871 88. **Campus Sundsvall** Holmgatan 10, SE-851 70 Sundsvall.
Campus Östersund Kunskapens väg 8, SE-831 25 Östersund.
Phone: +46 (0)771 97 50 00, Fax: +46 (0)771 97 50 01.

Abstract

Applications that can change their behaviour based on the user's contextual appearance are called context-aware applications. Applications developed for smart phones, which carry a multitude of different sensors, and actuators have gained a huge penetration within the market. Frequent usage of mobile applications such as home automation, friend finder, car accident notification and tracking etc; have seen an increased growth from the user's perspective. However, for such applications, it is necessary to have knowledge of the performance and cost parameters, which directly affect the QoE of the application users. This thesis investigates possible context aware behaviours in real time situations and measures the Quality of Experience (QoE) as well as identifying the Packet Delay Variation (PDV). The background and some state-of-the-art technologies are studied, and based on these studies, three scenarios are designed and implemented and based on their QoE, results and conclusions are presented, with the results obtained shown by means of graphical representations of the QoE and PDV values.

Keywords: Sensors, Context-aware-applications, QoE, PDV.

Acknowledgements

First of all, I would like to thank Stefan Forstrom who supervised me throughout this project. I would also like to thank and extend my sincere gratitude to Prof. Tingting Zhang and last, but not least, obviously my beloved family who have supported me both morally and financially.

Table of Contents

Terminology

Acronyms / Abbreviations

GPS	Global Positioning System
PDV	Packet Delay Variation
RFID	Radio Frequency Identification
QoE	Quality of Experience
QoS	Quality of Service
Wi-Fi	A wireless communication medium

1 Introduction

Context aware applications change their behaviour on the basis of the user's context. The introduction of smart phones has meant that there has been a significant penetrative spread with regards to these applications. At the present time advanced sensing technology and research are being adopted in relation to the development of user's contextual behaviour in different environments and for different purposes. It is possible for each sensor-oriented application to be measured for its own usability and behaviour in uncertain environments and situations and thus, this thesis starts by considering the range of possible and practically useful context-aware applications such as friend finder, car accidental location and etc., which would fulfil the Quality of Experience (QoE) as perceived by the users. The aim of this thesis, on the basis of the proposed scenarios, is, to some extent, to support the work which is currently being conducted in ongoing research with regards to real time context-aware applications and response times in terms of the user's context.

1.1 Background and problem motivation

In a context-awareness situation, computers or mobile devices may use the fact that they have the ability to sense and react based on their sensing ability to the current situation in which they are working. The main goal of the context-awareness is to minimize the situation in which user's are required to react's manually by automatically responding to the environmental changes. Thus, according to this statement, any information or data that is collected by sensing is context; therefore an increase in sensor technology has also increased the context-aware scenarios. The most common scenario in context-aware systems is the user's geo location (i.e. latitude and longitude). Extensive investigations have already been conducted with regards to these systems. The most popular technology, used as source of sensing for location-based applications, is GPS. The output of the GPS receiver is time and location. If seen in a real time context, the accuracy of this approach totally depends on the number of available visible satellites and their signal strengths. At the beginning, context-awareness was widely perceived on the basis of a user's location. However, there could be many other sources of applications for the measurement of context awareness such as light, temperature, sound, motion orientation and etc.

As described above, the response time for context-aware real time applications has a major impact on their performance and accuracy. If the performance time is good then the QoE for a perceived user will be satisfied. However, for such applications, it is necessary to have knowledge of their performance and cost parameters, which directly affect the QoE of the application for the users.

1.2 Overall aim

The overall aim of this thesis is to investigate, design and develop such context-aware applications, which have a good response time in real time situations, thus fulfilling the QoE from the user's perspective. The problem statement defined in this thesis is that most context-aware systems have accuracy problems which do not show the exact proximity of objects (friend's location) and have more latency time in real time applications, such as home automation and car accident notification and tracking. If time and accuracy are not acceptable it may lead to a dangerous situation such as "fire due to Owen in home automation scenario". Thus, the problem is to investigate the end user's feelings and behaviour based on the response of context-aware applications in real time situations. Therefore, the aim in this thesis is to evaluate the perceived QoE in real time context-aware applications.

1.3 Concrete and verifiable goals

The goals derived from the aforementioned problem statement are as follows:

1. The theory background and work, which has already been performed i.e., home automation, friend finder and car accident notification in experimental research work
2. Designed and developed case scenarios as a proof of concept
3. Evaluate the applications with respect to response time in real time situations
4. Find appropriate methods for strong QoE i.e. subjective and objective methods

To achieve the end goal, all the aforementioned points should be answerable in order to fulfil the thesis requirements.

1.4 Scope

The scope of this thesis is wide; therefore the main focus will be on specific areas, which will be investigated in more depth. The areas that will be investigated in more depth are:

- Understanding and developing context-aware applications. Applications such as friend finder and car accident notification and tracking are the prominent examples of context-aware applications.
- Measuring the response time in real time situations where time critical applications are developed. Time measurements are important aspects of an event notifications system.
- Fulfil the Quality of Experience (QoE) requirements. These requirements could be gathered by surveying people's behaviours and their satisfaction level (see Appendix).

The areas, which will only be partially investigated, are:

- Sensing devices and sensors i.e. mobile sensors and GPS
- Tracking systems i.e. sensors embedded in vehicles
- Communication Protocols i.e. HTTP, Bluetooth and Wi-Fi

And the area that falls outside the scope of this thesis is:

- Communication Security i.e. cryptography techniques

1.5 Outline

Chapter 1 describes the background and problem motivation behind this thesis. Chapter 2 describes the overall theory related to the work conducted in this project and answers to the questions that are asked in chapter 1. Chapter 3 describes methodology concepts and approaches to problems. Chapter 4 describes the design, while Chapter 5 explains the implemented part of this thesis. Chapter 6 describes the obtained results. Chapter 7 describes the conclusions and a discussion about the project and this all is followed by references and attached appendices.

1.6 Contributions

Author solely conducted the research and scenarios implementation in this thesis. However, credit goes to Stefan Forstrom who supervised me throughout the completion of this project. The credit also goes to my friend Qaiser Mehmood for his support in implementation and reviewing the report thoroughly.

2 Theory

This chapter describes the literature overview about the technologies, which more or less relate to this thesis. This part also includes the related works which have been conducted up to this point. Based on theory, the proposed solutions are designed and implemented.

2.1 Mobile Computing

Mobile computing can be referred to as ubiquitous access to information, anywhere, anytime by using small portable and handheld computing devices such as mobile phones, laptop and PDA, etc. in a wireless enabled network. If this is examined in a broader sense, it is a usage of handheld computing device without cables [1].

2.2 Context awareness

The term context awareness may be applied more flexibly in relation to computing technology, namely mobile computing or to the so-called pervasive computing, and in more a general way, as ubiquitous computing [2]. In computer science importance of context has increased in recent decades because computers have become more pervasive in everyday life. The "topic context awareness" has gained popularity in recent years as computers can sense and react to a user's current situation. A variety of areas are contained within the term context, making the lives of users' to be easier, for example, when searching for nearby friends, transportation and restaurants. The most ubiquitous tool is the mobile device and their potential, for context awareness has increased enormously and has permeated into everyday life. The idea of mobile context awareness depends on the mobility of these devices where sensing and reacting are enabled [3]. The core concept behind context awareness is to obtain the right information, for the right people at the right time, with regards to information that is shared. Figure 1 depicts a pictorial view of context awareness, where information in real time is collected and shared by different objects, such as devices being accompanied by people, moving cars, buildings etc. There are streams of information shared in real time, which seek out and inform us. All this information is collected in an intelligent manner; and thus users are not required to manually enter a web page to obtain the information or to inform themselves about a particular event. The implemented scenarios relate directly to the context awareness concept and to the QoE of the end users.

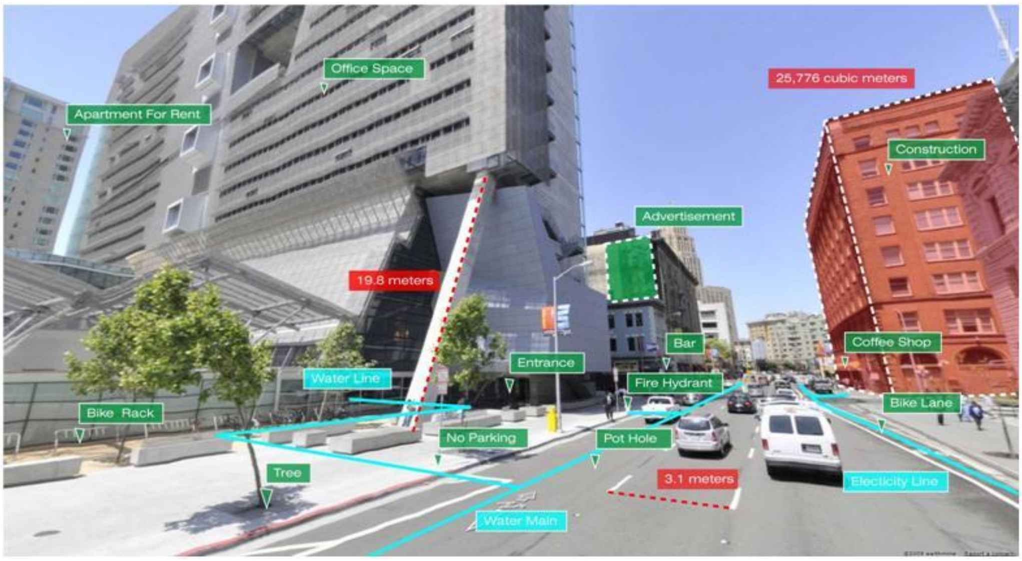


Figure 2.1: Context-awareness in different objects [4]

2.2.1 Active context

This is when application has the functionality to automatically adapt the discovered context, based on application behavioral variations. These application systems automatically process information and adapt the context. Researchers [5] have begun to examine computers system that change their functionality automatically based on observations of whom and what is around them. Based on context observation, the sensor systems that produce data are distributed over the environment and computing devices could personalize themselves according to the user currently interacting with them. These systems are able to adapt their behavior according to the user's location, or can react to their surroundings.

2.2.2 Passive context

Applications [6] that merely present the updated context to the user and enable the users to determine how and what notion of the application should change. An interested user is presented with new updates and contents by the application; these contents persist for the particular user to retrieve at a later stage. Although passive context is important and could be relevant to an application, it is not critical.

2.3 Context aware applications

In our day to day living experiences, the activity performed by any individual, processes of very large corporations such as IBM, Ericsson etc, can be simplified by the computing systems that are aware of a worker's context. Context means information about any particular entity and its surrounding environment that can be used to infer in such a way that computer systems can serve at their best for the individuals.

The word context may include location, blood pressure, calendar appointment, or current functional activities [7]. An individual's context may also involve airline schedules, friend finding, car tracking, or a set of people in the same room. This could involve people being located within a few feet of coordinates in a building coordinate system.

Further applications, which are related to context awareness, may include advising a driver to take a particular route towards a location, current traffic condition and the required destination. These applications also may include advising a nurse to attend to a patient based on medical telemetry being received from a particular patient; delivering a notification to someone depending on the recipient current's contextual location [7].

2.3.1 Real-time context aware applications

The study of incorporated hardware and software systems based on real-time constrained functionalities are called real time systems in computer science [8]. Real time must guarantee a response within strict time constrains i.e., in milliseconds and sometimes even in microseconds [9]. The applications scenarios i.e., Geo-location, Friend finding and Car accident notification and tracking system, designed in this thesis are real time applications although not time critical applications.

2.3.2 Evaluating context aware applications

Evaluating context aware applications has proved to be rather difficult. Evaluating other systems' applications or subjects might be strictly black or white, while, on the other side, context awareness appears to be a somewhat grey area to evaluate. For context aware systems, the targeted group plays an important role for product evaluation. Based on this targeted group's answers, an attempt can be made to broaden the use of this product in order for it to be available in an easy and efficient manner. Since both people and context maybe changed unexpectedly, this makes it difficult to test. The solutions which are suggested mostly rely on 1) bug-free and 2) should not be slow. The evaluations are mostly achieved by means of surveys [10, 11] and from the feedback of users who use such types of applications.

2.4 Notification and Tracking System

The tracking system has become an important factor and has already been introduced in the transport industry. Every advanced car making company fits tracking facilities in cars so as to track the car if it is stolen or lost by someone. In this thesis a step beyond the tracking system will be taken i.e., the notification system. A notification system is a combination of software and hardware that provides a means of delivering a message to a set of recipients [12]. The idea is to generate an accident notification event: if a car passenger gets into an accident, all the family members are automatically notified and the location of that particular person is also sent to the family members, so that they can track the car's location. Figure 2.2 shows a pictorial representation of a tracking system for, a vehicle.

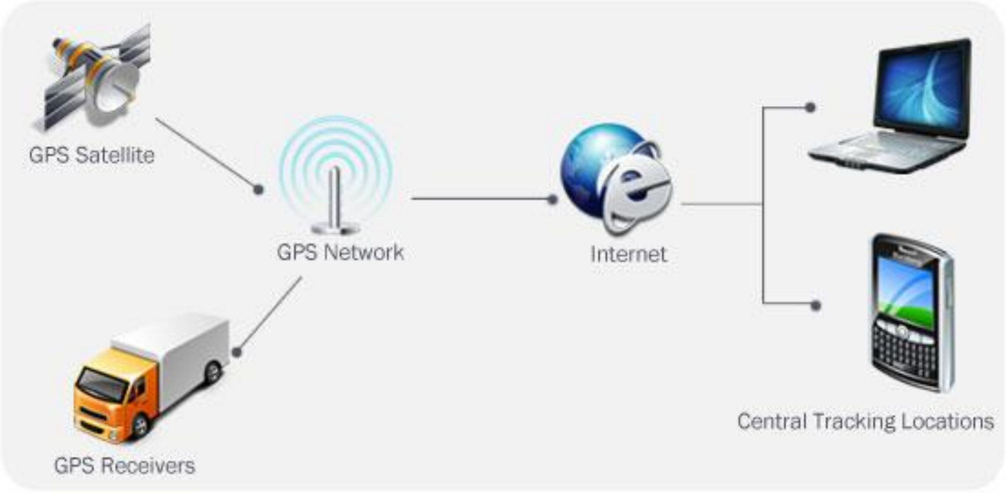


Figure 2.2: Vehicle tracking system [13]

2.5 Quality of experience

In context awareness Quality of Experience (QoE) [14, 15] is difficult because it is based on human experiences. Feedback provided by humans will differ depending on different societies, locations and regions and these will thus results in; different ideas regarding both quality and the perception of experiences. QoE is also very important in an application development life cycle as the end users care about the product selection, and, based on their usage experience, a product could be valuable and placed in the market otherwise; people might hesitate to use this product. Price may not be an issue for People but, most prefer user-friendly products with many built in functionalities. The end user should feel that the product is just for them as per their requirements. Applications running on mobile handheld devices with wireless capabilities are gaining penetration in the market and more than 5 billion global mobile connections have already entered the market. Due to the large amount of applications developed for similar purposes, the possibility of succeeding over other applications will depend not only on the technical quality but, to a larger extent, on the appreciation of the application by the user based on his/her subjective experience and satisfaction level. Figure 2.3 depicts the conceptual representation of a user to measure the different parameters of QoE.

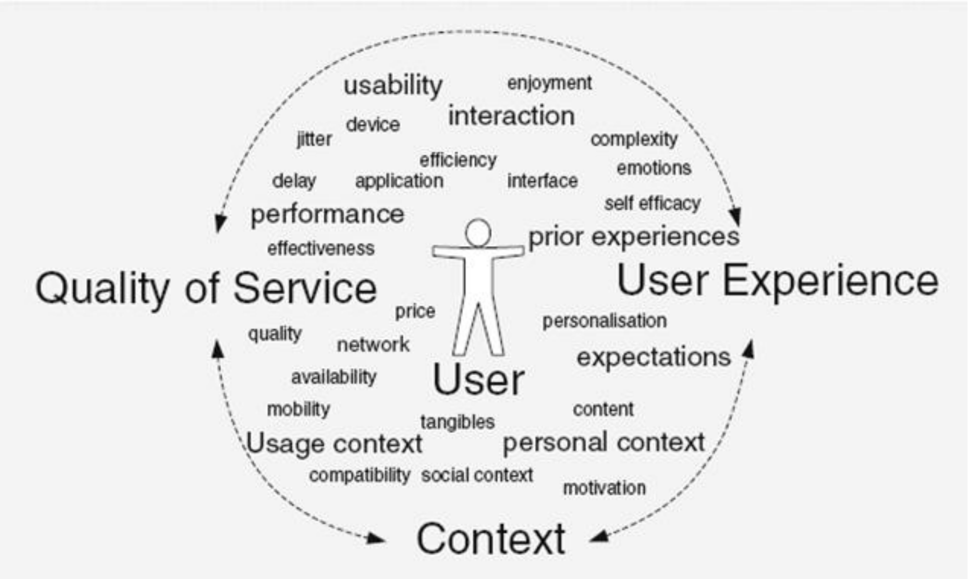


Figure 2.3: Quality of Experience (QoE) [6]

2.5.1 Quality of experience in mobile environments

QoE plays an important role in application development [14]. Since QoE varies from user to user as discussed in *Section 2.3*, it is, therefore, very important to realize that QoE can change with location updates, formal and informal meetings. QoE is also used outside mobile environments, meaning that it defines the experience of the user. While many tasks in the life of a user just "go the way they go", QoE attempts to define how these tasks qualify within the life of the user [15]. If a task/experience can be made more pleasant or different in order to benefit the user, then it becomes possible to adapt to this. Sometimes systems/products will differ according to the requirements of a specific user, while, in other cases this might involve an entire user group.

2.5.2 Evaluation of QoE

Defining QoE is one thing, but evaluating this causes the difficulty to rise to an even higher level. The QoE is very difficult to measure, and although there are users who are able to define what it is that they specifically like or do not like about an application, for the majority of the time, this is very difficult and depends on the environment in which the user is using the application. When using this in a formal environment, it should appear more formal, official and different to when it is being used in an informal environment with friends or family. It us generally the cause that tester, who uses the application for a defined time, practises QoE. After this testing time, all the testers provide a number based on their experience of the application, and what points they would like to see changed. After changing this, a new testing will occur [12, 13, 14, and 15]. While this will take a long time, it will secure the usage of the application and will filter out almost all the bugs. In the relation to the increase in the application development for mobile devices, the users,' acceptance is a critical point based on their perceived QoE. Considering an application for a payment procedure, if it is slow, which could be opposite to the user expectations, the user will hesitate to provide credit card number because of the threat of any attack, which indicates the low QoE level of the user. As depicted in Figure 2.4, the quality of service can be quantified by taking the measurements delay, jitter and carrying network capacity. The measurement of quality of service directly affects the QoE.

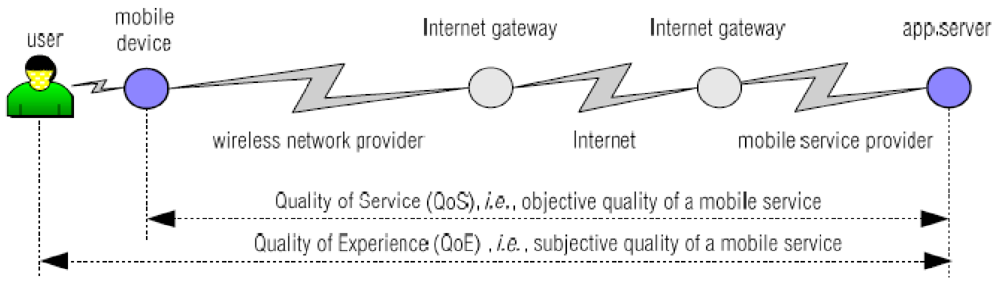

user
mobile device
Internet gateway
Internet gateway
app.server
wireless network provider
Internet
mobile service provider
Quality of Service (QoS), i.e., objective quality of a mobile service
Quality of Experience (QoE) , i.e., subjective quality of a mobile service

Figure 2.4: A concept of QoE and QoS in mobile service delivery [17]

2.6 Related work

It has been observed that in user's applications, geographical and location-based services are gaining attention, especially social networking. Applications such as Google, Facebook, IM clients etc., facilitate their users by mean of a register which can trace the geo-location of their friends and family. In this section an explanation regarding the work which has been conducted by different researchers, relating to the implemented scenarios in the current document is provided.

2.6.1 Home Automation System

The concepts behind home automation systems have existed for many years. Two terms i.e., "Smart Home", "Intelligent Home" have been used to introduce the idea of remotely controlling the appliances in houses. This field with its huge potential has penetrated the emerging market. In [18] a home automation system is implemented using a Bluetooth connection. In android devices, although Bluetooth has some advantages, such as low consumption of battery as compared to Wi-Fi, some drawbacks do still exit when using Bluetooth in automation systems. These drawbacks include, first and foremost, stability and range issues, discovery problems by the carriers' [19] data transmission rate [20] while on the other hand Wi-Fi is faster and is a long-range technology.

2.6.2 Friend Finder

The motivation behind all location-based information systems is "to assist with exact information, at the right place in real time with personalized setup and location sensitiveness". In this era mobile applications have gained popularity and have replaced old desktop systems. In [21] location-based information systems are explained in a smooth manner in order to understand the location services. Although trust and security of mobile communication is not part of this thesis, [22] does, however, elaborate some important concepts in location-based services and points out the basic and important issues and solutions.

2.6.3 Car tracking

There has a great deal of work conducted in which tracking systems are implemented for moving objects such as car tracking, wheelchairs etc. In [23] the author has explicated the motor's performance, monitoring and tracking to a mobile using Bluetooth as the conduit. Real time car detection in mobile device [24] has been implemented, where authors present a car tracking system using a mobile device with a camera, which uses a variant of [25], and a car tracker employing Lucas-Kanade optical algorithm.

3 Methodology

The experimental work, which will contribute to this thesis, is based on scientific methods. The proposed solutions will be evaluated by comparing different solutions, which have already been developed for this experimental research area. The main objective of this thesis is to investigate and develop context-aware applications, which can be both fast and accurate in real time, where time critical situations are supposed to be required. In order to achieve the required goals, the intention is to conduct a study in order to understand the state- of- art context-aware applications and their evaluation methods. The development process will start after the related work has been analysed and an evaluation, which should fulfil the expected results, will be performed.

3.1 Considered Scenarios

The following scenarios are supposed to be designed and implemented.

a) **Home automation:** in which a user can control the home appliances using a mobile phone.

b) **Friend finder:** in this scenario user can find a friend's location in a real time situation.

c) **Car accident notification and tracking:** this scenario demonstrates an augmented reality concept of how someone can track a car which belongs to him/her, when a passenger gets into an accident.

3.2 Development

In order to fulfil the required goals, the following tools and devices will be used.

1. Eclipse and Netbeans will be used as development tools

2. Java language will be used for android applications and server coding

3. Wi-Fi as a communication medium between the client and the server

4. Google maps in both friend finding and car tracking scenarios

3.3 Test Model

A test model will be formulated on platforms involved in the design scenario to fulfil and satisfy the requirements. The following measurements will be performed:

- **Context-aware based sensors' response time:** different types of available proximity sensors will be tested and compared with respect to data collection time in real time scenarios.

- **Mobile processing time:** along with sensor's data collection time from the environment; the mobile processing time will also be considered.

- **Server request reply time:** as client / server communication is involved in the home automation scenario, the request reply time will be calculated.

- QoE will also be a part of this test model on the application level.

- **Total calculated time:** this involves considered three different scenarios, therefore, the accumulated time measurements will be conducted for entities / component involved in each scenario, separately.

3.4 Evaluation

In order to evaluate the expected results, which are mentioned in *section 3.3*, the implemented applications will be installed on different devices, varying in processing powers. Evaluation will be conducted for the latency which occurs during communication flow between clients and servers. Moreover, there will also be an evaluation based on the designed scenarios by comparing these with solutions, already existing. The evaluation will also be based upon the user's friendliness perspective i.e., QoE by adopting the subjective and objective methods.

4 Design

This chapter focuses on design consideration, which will be implemented in a later chapter. Conceptual designs of the aforementioned scenarios are elaborated in this chapter.

As described, both hardware and software are part of this thesis. Nexus S is used for the developed android apps. In home automation, a laptop is used pointing a fixed server at home. Furthermore, Wi-Fi is used for the communication medium between the server and the mobile app, which acts as a client.

The following Figure 4.1 depicts the elements involved in overall architecture for considered solutions.

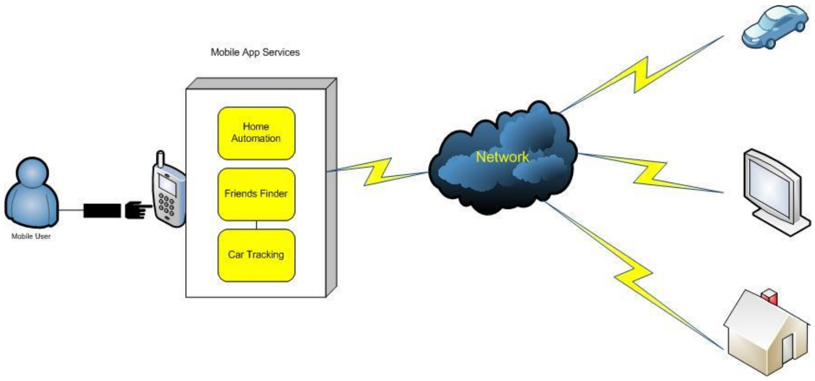


Figure 4.1: proposed architecture

Technologies introduced in smart phones extend the usability to different prospective i.e. home automation, friend finding and augmented reality such as car tracking. Two types of communication are involved with regards to the considered scenarios.

4.1 Home automation scenario design

Home automation, for those not familiar with that technology involves it having the ability to be pointed to the systems that allow the family members to remotely control home appliances such as doors, lights and multimedia systems. This technology consists of two parts i) user interface ii) communication protocol. *Wi-Fi* is used to operate between a mobile and a laptop. The scenario designed in this thesis consists of three home appliances that are described in the following paragraphs:

If a member of a family arrives home earlier than expected and requires to enter the house, but, unfortunately has no keys, if the home automation app is installed on their smart phones then there would be no requirement to wait for the key holder to return.

The developed app has two buttons on the screen to lock and unlock the door. If the door has to be opened, then this button is pressed and, an authenticated "open" message will be sent under a custom developed communication protocol towards the server, which unlocks the door. The same event is applied to lock the door but, in reverse order, with a "close" message.

The developed app also has the ability to turn "Off" or "On" the house's lights. Messages such as "turn-on" and "turn-off" are exchanged in this event between the app and the server.

If the TV' remote control falls down and loses its functionality due to some unknown reasons or if it cannot be found ,then an app in a mobile is able to perform the functionality same way as the remote control.

The app, which was developed in this thesis, is an alternative to a broken or lost TV remote. It is becomes possible to control a TV remotely with the developed app. The functionalities of this app include "On" or "Off" TV and setting a desired channel.

4.2 Friend finder scenario design

If it is necessary to discover friends no matter where you or your friends are, then this application integrates with Goole Maps and will assist in finding the location of each entity subscribed to the system. The app enables the selection of a friend from a list and an event is fired off by just clicking on the button; then both will automatically be notified showing geo-location to both. This app saves the phone call e.g., you say to your friend "I am in Stockholm" and he/she asks "at which location in Stockholm you are now?" In this app development *online communication* is involved, which operates GPS for location finding. A GPS is the technology used for navigation systems, which communicate via a satellite channel.

4.3 Car accident notification and tracking scenario design

The concept behind vehicle tracking is not significantly different to that for the friend tracking system except that it dispatches an accident notification to the family members along with the location of the accident place. The only difference lies in the sources or devices that publish data (latitude & longitude) over a cloud and it from this that the app can obtained the location of the vehicle. GPS is also used so that the vehicle and app can communicate.

4.4 QoE Evaluation Design

To ensure the best solutions from the designed scenarios, consideration was given to the fact that the mobile interfaces should be user-friendly and enabled for the best human performance. It was ensured that it was possible to evaluate the QoE and implementation life cycle by the adopted methodology (*see Chapter 3*). After the design phase it is shown, in chapter 6, the results, which address the outcomes, findings and recommendations.

Evaluation based on customer's satisfaction concerning the service provider is becoming a more and more important topic. Therefore, the QoE methods for the measurements are adopted in this thesis where evaluation is based on Quality of Services (QoS) parameters. The transmission capabilities of the network and resources availability are the measurements in QoS but not for the satisfaction with regards to the customers. Thus, in this regards, the measurements for the quality of services are very closely related to the QoE for the end users. Thus, in this thesis the evaluation is achieved by means of a correlation model of QoE and QoS, where QoS is used to measure the metrics such as packet delay, loss and jitter of the ongoing communication between the involved objects. A questionnaire research has been conducted in order to evaluate the design scenario, and the research is based on the following:

I. What are the QoE ratings which effect and influence the user's behaviour?

II. What QoE metrics for the developed applications are most influential?

III. At what pattern does an Operating system (OS) of the mobile impact the users?

5 Implementation

This chapter describes the implementation of the proposed scenarios. This includes technologies and hardware used in the implementation. The main objective of this implementation chapter is to demonstrate the overall functionalities performed by the systems developed for this thesis.

Java language is used for apps' development. Two different platforms i) Eclipse for android and ii) Netbeans are used for the development processes. The ultimate goals of this thesis are not to create commercialized applications but, to prove the concepts designed and to produce results. Finally, to draw conclusions from the implementations.

5.1 Home Automation Scenario implementation

The overall idea behind this development is discussed in more detail in *section 4.2*. The following Figure 5.1 depicts the hardware devices used in this scenario implementation.

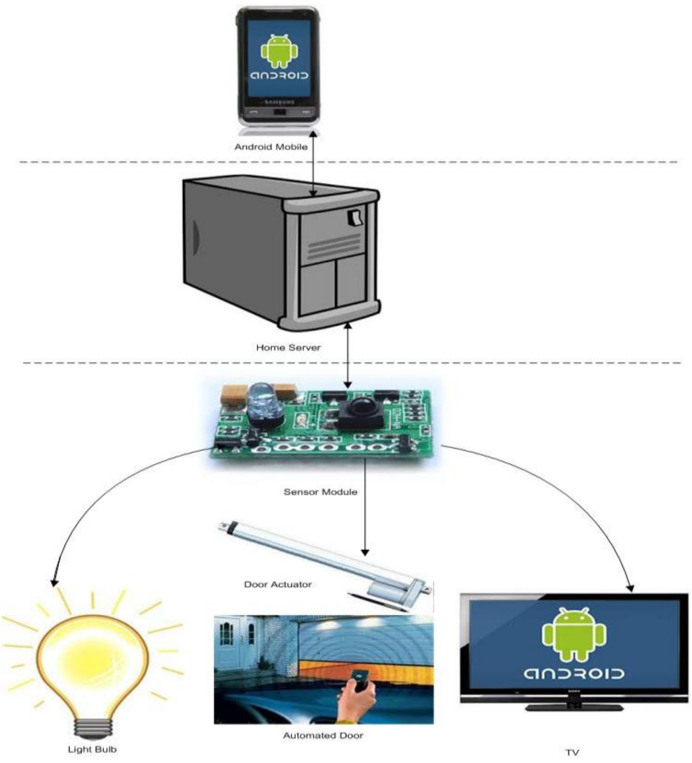


Figure 5.1: Used Hardwar

The above Fig 5.1 depicts the communication flow between the nodes with their respective layers in the designed architecture. An Android mobile communicates with the server via Wi-Fi, while the server communicates with a sensor module, which further interacts with home internal appliances.

5.1.1 Android Mobile Phone

In the proposed scenario the android mobile is an important asset that meets all the functional requirements with the developed app installed on it. The application consists of a user-friendly interface and the Wi-Fi communication capability to interact with the home server.

5.1.2 Home Server

The most important and secure part of the designed scenario is the home server, which communicates with the mobile via Wi-Fi, and which ,authenticates the home users already saved in server's database. The Home server, at the same time, communicates with the door's actuator system if the mobile user wants to open or lock the door, the home server also communicates with a sensor module, which is used to coordinate with internal home appliances i.e. TV and lights.

5.1.3 Actuator

An Actuator is the key component, which mechanically acts to open or close the door. It receives an event containing the "Open", "Close" objects and acts on the server command.

5.1.4 Sensors module

Sensor module is used internally within the home to communicate with the home appliances such as light bulb and TV.

5.1.5 Application flow

The following Figure 5.2 depicts the component diagram of the developed home scenario in which each component exposes its interface for input and output.

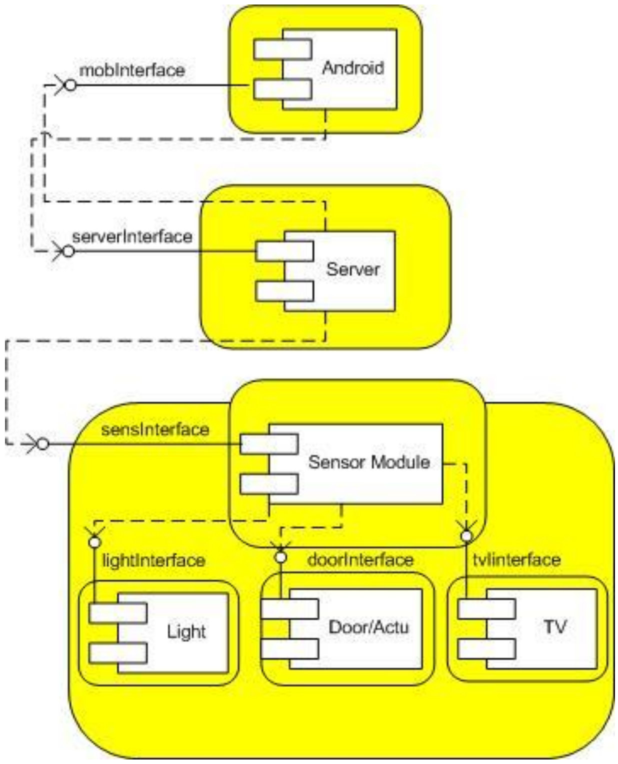


Figure 5.2: system component diagram

Security concepts that are not required to be investigated in more depth in the sense of communication do not form part this thesis. However, authentication is required to be involved to some extent, especially in a mobile application. This is because, there is always a threat of mobile access by unwanted persons when people lose their mobile phones, especially as the developed application involves the locking and unlocking of the home door, electricity dependent compliances such as light bulb and TV.

When a user attempts to launch the application, an activity with a password field comes to the foreground in front of the application launcher. This activity requires a password in order for it to go to the main activity with the door lock and unlock buttons. Until a password is accepted, the main activity will never come to the front and the Wi-Fi connection to communicate with the home server will not be enabled. The flow chart Figure 5.3 below depicts the application access mechanism.

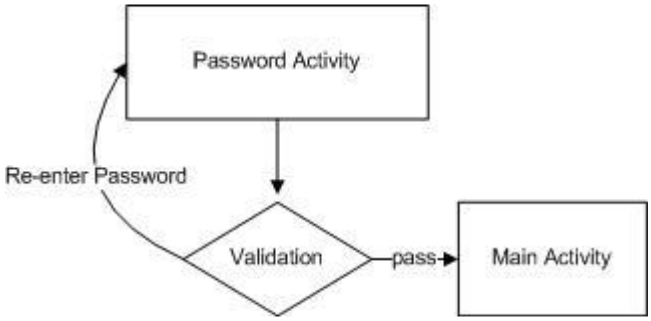


Figure 5.3: Application Access flow chart

If application access is passed with password acceptance, a Wi-Fi connection is enabled for application usage. When a door is required to be opened or closed, the mobile user presses the corresponding button on the mobile activity interface. For example, when the user presses a button an event is fired; the event envelops the required message data in a packet format having such as "Open" or "Close", with some additional user related parameters. The packet is forwarded to the home server via a Wi-Fi connection.

The home server first and foremost authenticates the sender; if authentication passes, the server sends an event containing a target message to the actuator, lights or TV. The actuator mechanically acts on the door either by opening or closing it.

When a user is authenticated and the home server has received the packet, the server checks what type of packet it is and forwards it to an appropriate device, either to the actuator, lights or TV. A sensor module is used to internally communicate with either the light bulbs or the TV interface.

The following Figure 5.4 depicts the overall home automation architecture with the message flow in a sequence diagram. The figure represents the time when an authentic user has an android mobile running the deployed application, and how the communication flow in the overall system takes place. The diagram involves both, local internal communication between the server and home appliances as well as the communication between the mobile and the server.

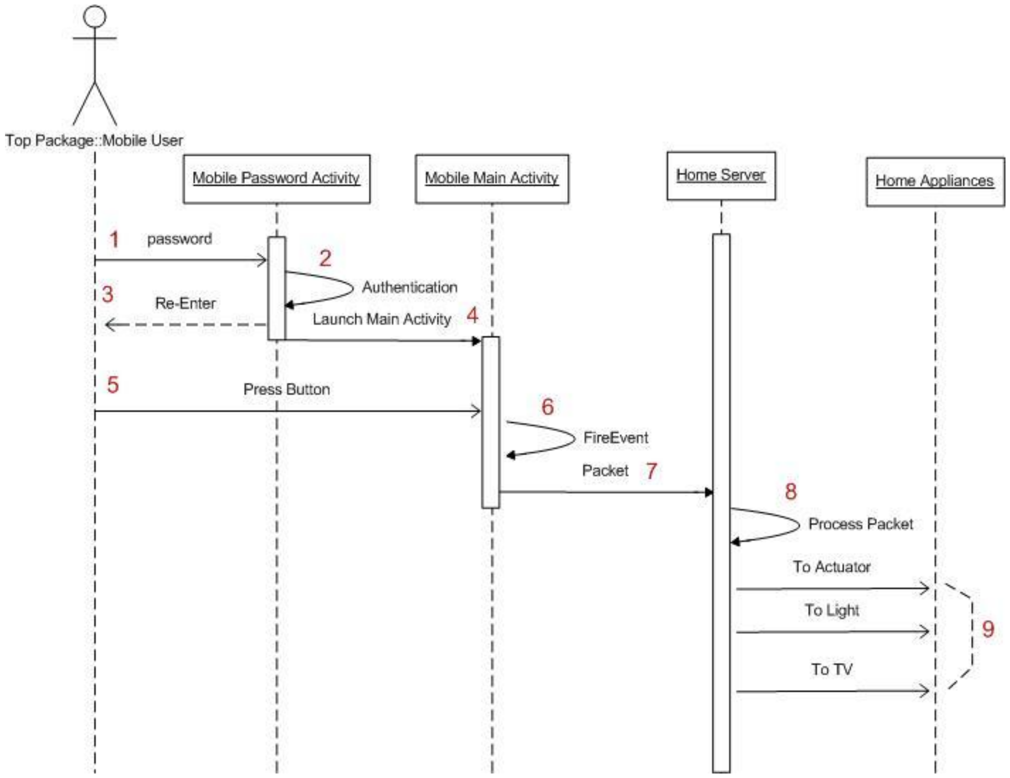


Figure 5.4: Home Automation Sequence diagram

The numbering of messages shows the process flows, which are as follows:

1. Whenever a user is required to run the application, the user enters a password to access the application
2. The authentication process starts for the application access. The user's credentials (password) are stored internally in the android local storage
3. If the password given by the user is wrong, the user is asked to re-enter the password. The application will not be launched until the password is correct

4. If the authentication process succeeds then the main activity is launched having the required components i.e. open door button, etc.

5. The user presses the button required for any action to be performed such as “Open”

6. On pressing the button an event is fired

7. The event contains a special packet which has a required action command for a particular task

8. On the server, this packet is processed and forwarded via sensor module towards a target resource

9. Target sources are door, TV, and lights

5.2 Friend Finder scenario implementation

Friend finder is a location-based system (LBS), which shows the geographical location of the application users. People who have installed this app on android phones can find the global position of their friends and family members. *Section 4.3* describes the background of this scenario. The hardware used in this scenario is an Android mobile phone. The system allows for the search of location of individuals.

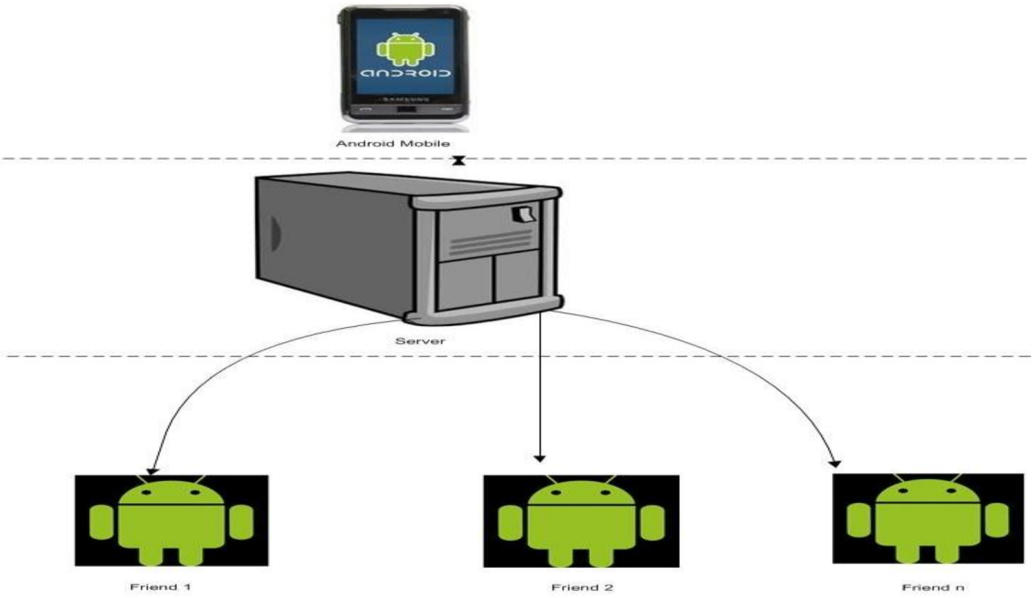


Figure 5.5: Used Hardwar

The above Fig 5.5 depicts the communication flow between nodes with their respective layers in the designed architecture.

5.2.1 Android Mobile

In the proposed scenario the android mobile is an important asset as the Location Finder application is developed and installed on the android platform. It sends the user's presence and location updates to the location engine.

5.2.2 Global Positioning System GPS

The Android platform provides a Location Manager api to obtained the geographical location by using a GPS that is a satellite-based navigation system. The friend finder application employs this communication technology to the location base.

5.2.3 Google Map

The application employs android's Google map api to render the location information on Google maps and this is visualized in the android application.

5.2.4 Location Engine LE

The most important asset of the implemented LBS is the location engine (LE). LE keeps the registered user's data (users name, email address etc.) as well as the updated information (location) from their current geographical areas that are sent from a mobile with a fixed time interval. LE is a restful web service that performs the aforementioned tasks.

5.2.5 Application flow

The interactions between different entities of the developed system are depicted in Figure 5.6, a component diagram. The diagram presents which components are exposing what interface for the other components.

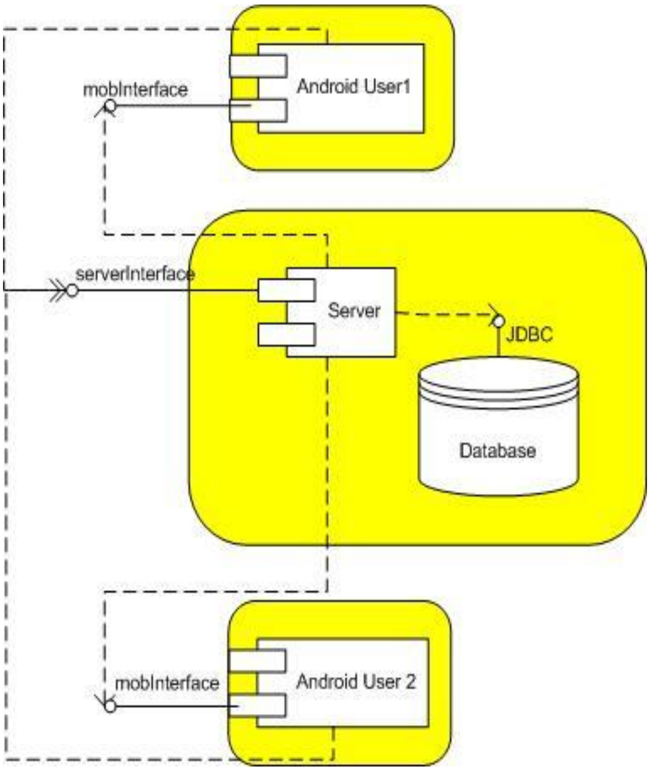


Figure 5.6: Friend finder component diagram

Considering a user, who is accompanied by an android phone with an installed friend finder application on this phone. When the user requires viewing the current location of his friend, he launches the app. As the user launches the application, an updated location packet in JSON format is sent to the web server. The web server parses the JSON packet and calls the inserting operation. The updated value as <Key, Value> pair is stored in the SQL database DB. The Key is a particular user whereby the value shows the updated location against the key. Considering an example, say a mobile user, named Stefan, would like information regarding the location of his friend Victor. Stefan enters the name of Victor in a text field and presses the search button. An event is triggered, which is actually a JSON request to the web server to return the geographical location of Victor. The web server parses this incoming request and, internally matches Victor against the location and replies with the required information to Stefan's request. When Stefan's mobile application receives this result from the server, it enables Victor's location to be seen in a Google map. Stefan can easily see the visualized location of Victor from Google Maps in the developed application.

Following Figure 5.7 depicts a sequence diagram, which elaborates the message flow between different entities involved in the LBS system. The elements involved in the designed system are mobile users, the web server and the database DB, used to store different parameter values.

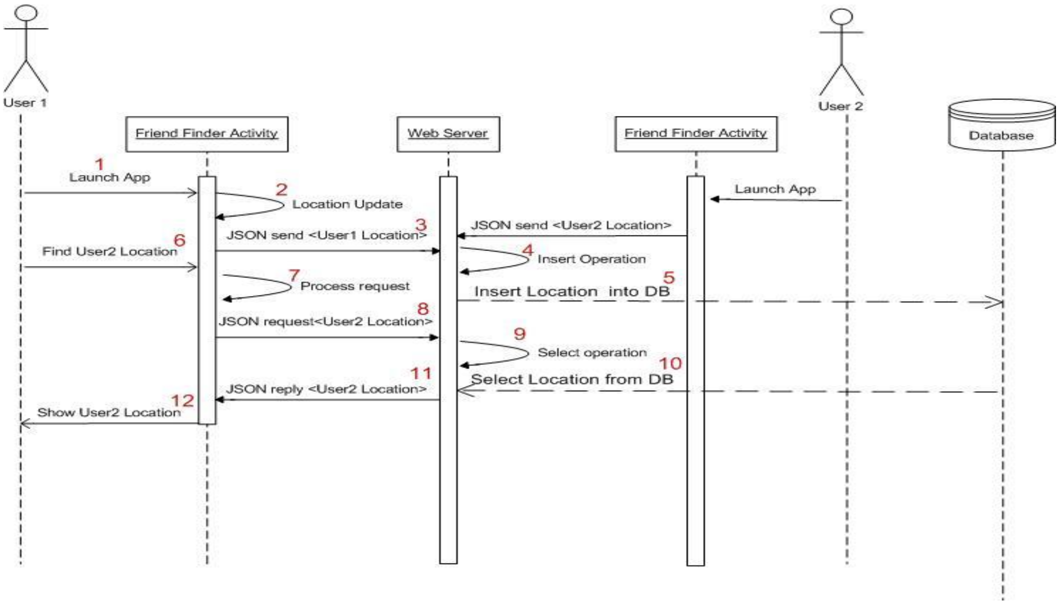


Figure 5.7: LBS Sequence diagram

The numbering of messages shows the process flows, which are as follows:

1. Whenever a mobile user wants to see the current location of his/her friends or family members wherever they are, the user launches the LBS app on a mobile.

2. As soon as the LBS app is launched an internal function is called and starts a backend service. By employing the android built-in LocationManager api, this service continuously receives an updated geographical location from GPS at a fixed interval of time.

3. The updated location of the mobile user, as a JSON packet format, is then sent to a central web server.

4. The web server processes and parses the JSON packet, obtains the location from it and passes this data to a method to insert into DB.

5. After obtaining the values against a particular user, the server inserts these updated values into DB as <Key, Value> pair.

6. When a user wishes to see the current location of his/her friends or family members, the user enters the name of a particular friend whose current location he/she wants.

7. After the search button is pressed by a user, two operations are performed; the first one converts the raw data into a JSON format so as to communicate with the server in a standard protocol.

8. As the search button is pressed, the second operation sends a request in JSON format to the server to obtain the current location of the required person.

9. The web server, after receiving a request from user1, maps the required user2 as Key, as the data in DB is stored as a pair <Key, Value>.

10. The updated location against user2's Key is obtained from the database, if the data against the requested Key is not null and updated.

11. The web server replies in the defined JSON protocol to user1.

12. The LBS app renders the updated location of the friend in Google map view.

5.3 Car accident notification scenario implementation

Car accident notification implementation is not very different to that of the aforementioned friend's finder scenario (*Section 5.2*), but does have some constraints. This scenario also employs a location-based system (LBS), where the geographical location of a car is tracked, whenever an accident happens to a car and this is shown in the user's application interface. People who have installed this app on android phones can find the global position of a registered car in the LBS system. *Section 4.4* describes the background of this implemented scenario. The hardware used in this scenario is an Android mobile phone, an event triggering mechanism and the Location Signals Broadcaster (LSB) fitted in a car. The system allows for the search of the location of the targeted car.

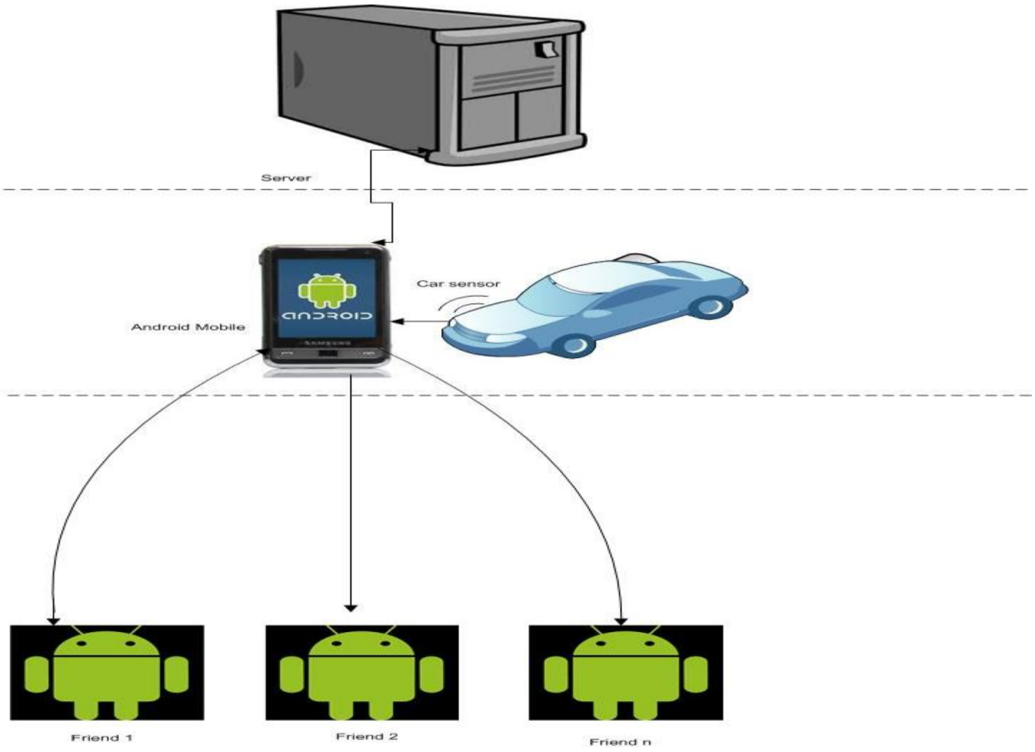


Figure 5.8: Used Hardwar

The above Fig 5.8 depicts the communication flow between nodes with their respective layers in the designed architecture.

5.3.1 Android Mobile

In the proposed scenario an android mobile is an important asset as the developed tracking application is installed on the android platform. It tracks the car in real time to a geographical area whenever an event is triggered from a car.

5.3.2 Global Positioning System GPS

The Android platform provides a Location Manager api to obtain the geographical location by using GPS, which is a satellite-based navigation system. The car tracking application employs this communication technology to the location base.

5.3.3 Google Map

The application employs an android Google map api to present the accident location information of the car on Google maps and to visualize it in the android application.

5.3.4 Location Engine LE

The most important asset of the implemented LBS is the location engine (LE). LE keeps the registered car's data (car number) as well as the updated information (location) from their current geographical areas that are sent from the LSB whenever an accident notification is generated from a car. LE is a restful web service that performs the aforementioned tasks.

5.3.5 Application flow

The interactions between different entities of the developed system are depicted in Figure 5.9, a component diagram. The diagram represents which component is exposed to what interface for the other components.

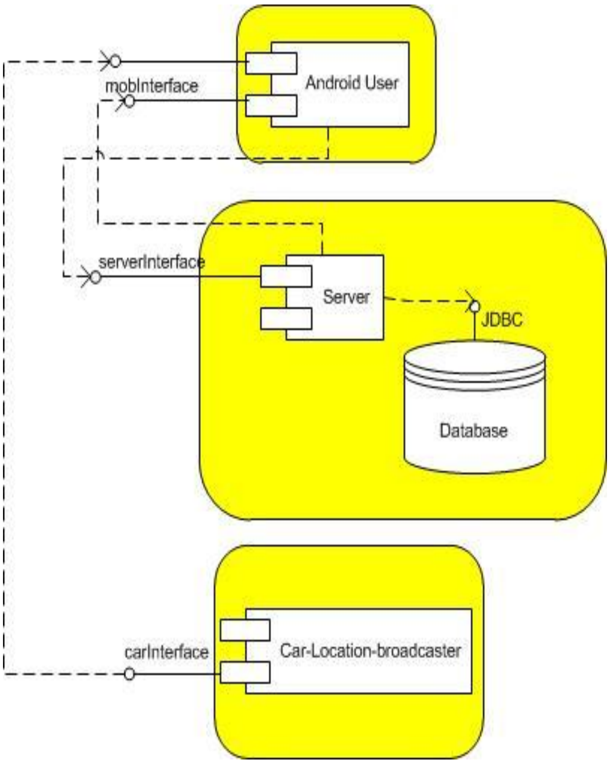


Figure: 5.9 Car accident notification component diagram

Considering a user travelling in a car, the user being accompanied by an android phone with an installed car accident notification application it. When an accident occurs to a particular car, an event of its geographic location is dispatched via the car location engine to the android user. A user, with the launched application, receives a notification that the car has got into an accident. In the case of an accident, a notification dispatched by the location engine is a packet in JSON format and this is sent to the web server. The web server listens and stores the car's location in its storage for future investigation and analysis by the authorities. The updated value as a pair <Key, Value> of the car accident is stored in the SQL's database DB. Key is a particular car number, for which the value shows the updated location against the Key received from the LSB. Considering an example, say a mobile user named User1 has installed this developed application on a mobile. User1's friend, named User2, is travelling by car equipped with an accident event generator and location engine. Unfortunately, User2 gets into a car accident; an event is then triggered, which is actually a JSON packet that has been sent to the web server to return the geographical location of the car. The web server parses this incoming packet, internally matches the car against its location and dispatches an accident notification containing the accident event and, the location of the accident to User1. When User1's mobile application receives this event from the server, it renders the car's location visually in a Google map. User1, from the Google map in the developed, application is able to easily see the visualized location of the car.

Figure 5.10 depicts a sequence diagram, which elaborates the message flow between different entities involved in the LBS system. The objects involved in the designed system are, a mobile user, a car having LSB, the web server and a database DB to store different parameter values.

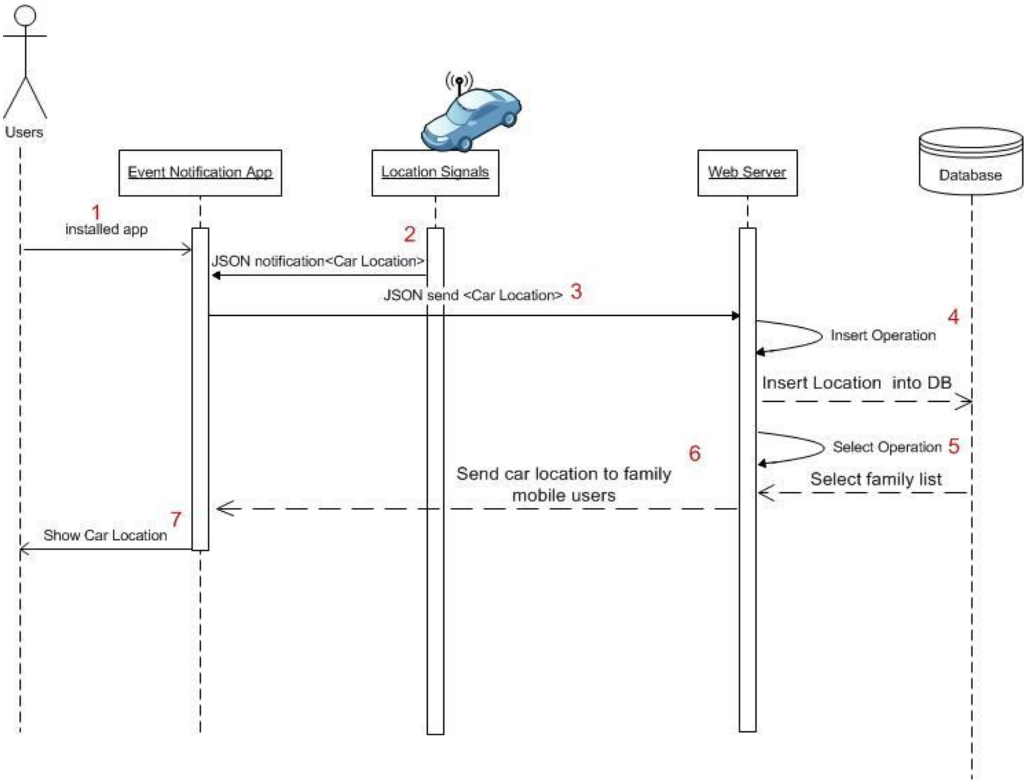


Figure: 5.10 Car accident notifications and tracking sequence diagram

The numbering of messages shows the process flows, which are as follows:

1. The car accident event notification app is installed and running as a backend service on the user's mobile.

2. The user, who is sitting and driving the car, his/her mobile is in communication locally with the LSB. Whenever an accident occurs the LSB sends a notification to the driver's mobile.

3. The app installed on driver's mobile broadcasts the accident notification message to the web server.

4. The web server inserts the current location of the car for future analysis.

5. The server from the DB also selects the list of family members to whom the notification messages have to be sent.

6. After selecting the list of family members the server dispatches the notification message to the family members about the accident message.

7. The family members have installed the app with the popup message and this show, in the Google map, the car’s geographic location where the incident occurred.

6 Results

The results obtained from the experimental work are discussed in this chapter. In design (*chapter 4)*, the proposed scenarios were explained and discussed, i) Home automation ii) Friend finder and iii) Car tracking. These implementations would prove to be useful in real world application scenarios.

The developed applications enhanced the mobile usability of the users. Installing these apps on mobiles can enhance the usability benefits for automation and tracking systems.

6.1 QoS and QoE metrics based acquisition

The result obtained for all scenarios is based on QoS and QoE metrics and involves the Computing Packet Delay Variation (PDV). The algorithm can be explained as, for each packet which is received there is a continuous communicating current time calculation, which involves the dispatching time $T_{S,n}$ for a packet *n* sent from the sender and, at the destination side, the arrival time $T_{R,n}$ for the packet *n.* The PDV can be calculated by subtracting the value of the receiving and dispatching packets.

PDV← deviation of $T_{R,n}$-$T_{S,n}$ for a specific time

The abovementioned algorithm calculates the end-to-end communication measurements between the network-involved objects. The latency time for a single packet D_n is calculated by subtracting the sender $T_{S,n}$ from the receiver $T_{R,n.}$

$$D_n = T_{R,n} - T_{S,n}$$

To calculate the end-to-end latency rate, PDV is defined as the standard deviation of the packets measured in a certain interval of time. This can be stated as:

$$\text{PDV}=std\{\text{certain time period}\}$$

6.2 Home automation scenario results

The scenario resulted in a developed prototype; a user accompanying an android mobile phone with the installed home automation application on the mobile can robustly and intelligently control the home appliances. Figure 6.1 depicts a snapshot of this particular android app with the features: i) turn-On & turn-Off the lights and ii) lock and unlock the door, more functionalities can be added if required.

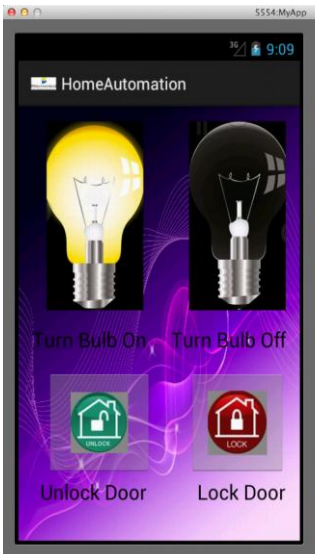
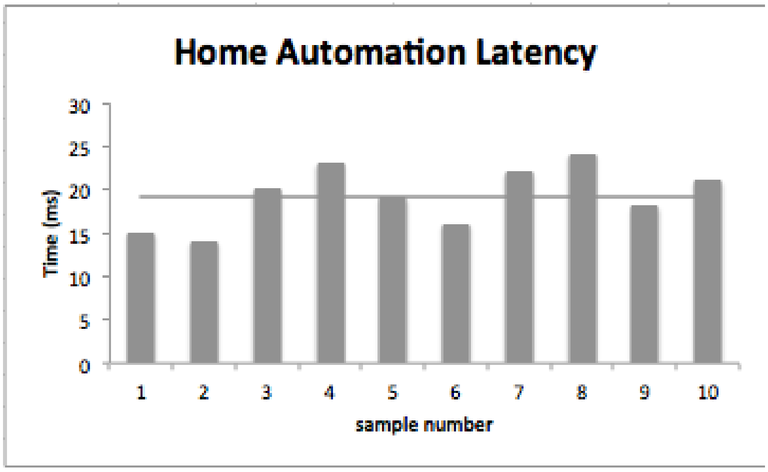


Figure 6.1: Android interface of Home automation

Measurements were taken from different perspectives. The results obtained from this scenario were reasonable.

When the project was completed, the mobile application was able to communicate with the home server from approximately twenty-five feet away. It was noticed that the total time taken to complete one action was, on average, 17ms. The latency rate between both points was calculated by employing the PDV algorithm.



Graph 1: Time measurements by different users

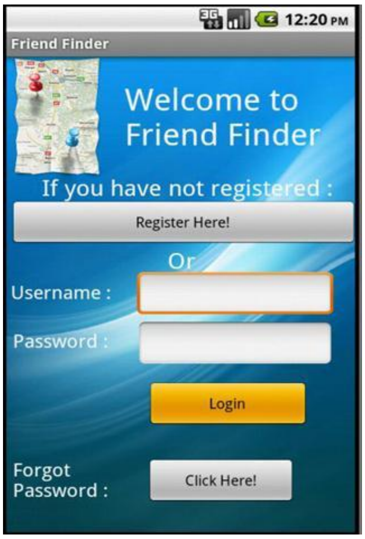
Graph 1 represents the measurements taken by 10 different users in different time intervals of the day, each data series represents the corresponding estimated time in milliseconds with respect to a particular user's QoE satisfaction level. The sample number 1 depicts the latency time, which is less than 15 milliseconds. Similarly, the other samples were taken and represent their corresponding time measurements. The variation in time depends on the behaviour of the system, such as processing power, network delay, the particular time of the day when measurement taken and etc. Graph 1 represents the average of the measurements taken by 10 randomly selected users. It was noticed that the average was around 17 milliseconds, which basically shows the users' QoE level. Since PDV is the standard deviation of the time variation, therefore the PDV was obtained by calculating the variance and the standard deviation. To calculate the variance each value was taken as its difference from the mean, which is 19 and the difference value was squared and then the average result was obtained by the following formula:

$$Std : s = 3.42ms$$

Since PDV shows the variations within a certain time period, the PDV standard deviation is about 3.42ms.

6.3 Friend finder

The friend finder scenario resulted in a developed prototype; a user accompanying an android mobile phone with the installed application on the mobile can search for the location of friends and family in a visualized form. Measurements were taken from different perspectives. Figure 6.2 depicts the friend finder application snapshot taken from an android phone.



Friend Finder

Welcome to Friend Finder

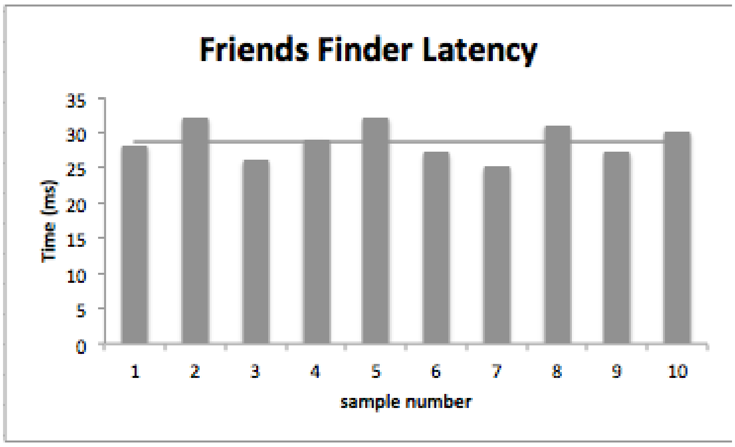
If you have not registered :

Fig 6.2: Friend finder snapshot

When the project was completed, the mobile application was able to communicate and show the result in a reasonable time interval. It was noticed that the total time taken was reasonable. The latency rate between both points was calculated by employing the PDV algorithm. Graph 2 represents the reading for 10 users and their QoE level. PDV as the standard deviation for Graph 2 is calculated and represented in the following equation:

$$Std : s = 2.36ms$$

The value 2.36ms represents the standard deviation for the time variation, which is basically the PDV.

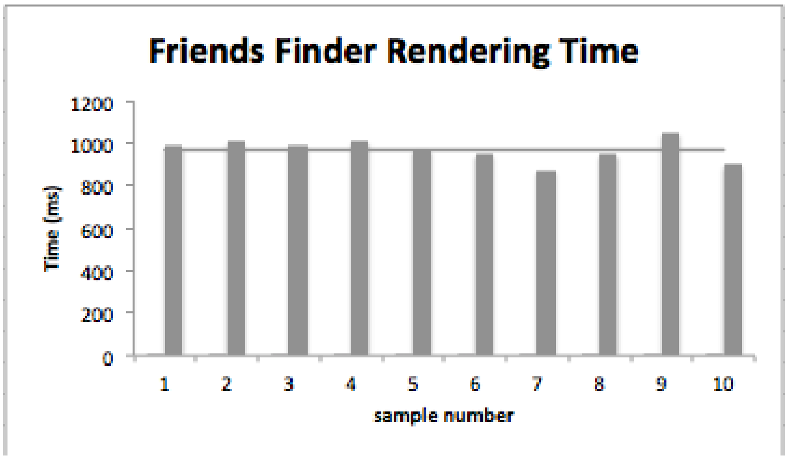


Graph 2: Mobile to mobile packet transmission time

The time measurements were also taken for message processing, rendering the geographical location within a mobile. Graph 3 represents the visualization time in a Google map, which was an increase as compared to the packet transmission time between two mobiles. The standard deviation for PDV is calculated by the aforementioned formula as follows:

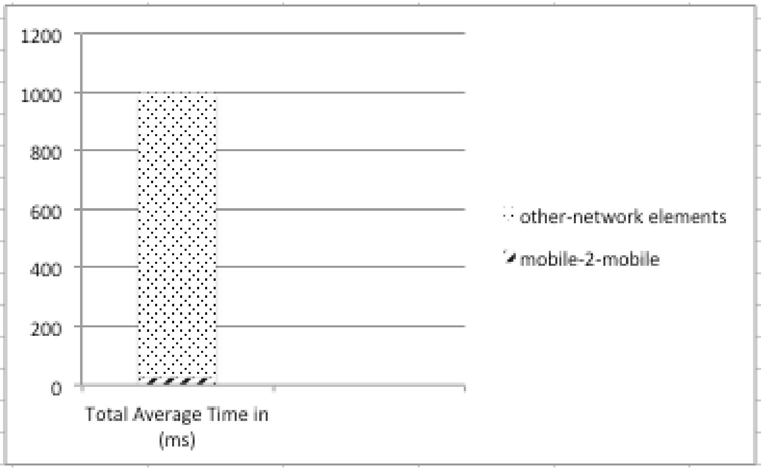
$$Std : s = 49.60ms$$

The calculated value for the standard deviation is 49.60ms, which represents the friend finder rendering time in Google map.



Graph 3: Goole map rendering and mobile processing time

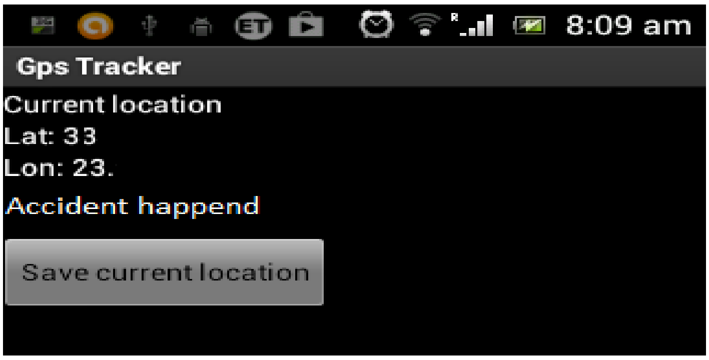
The corresponding average standard deviation time to accomplish an action is represented as a horizontal line in both Graphs 2 and 3, which is basically the users' QoE. The variation with respect to the time depends on the different behaviour of the system, such as processing, network delay, loading Google map in the app etc. The following Graph 4 represents the correlation time for the overall system, where 0.28 (almost 2% of the time) is for mobile to mobile while the rest of the time which is about 98% of the total correlation time, is the time taken by other network elements.



Graph 4: System Correlation Time

6.4 Car Accidental Notification

This prototype was developed to inform / notify the friends and family about the sad event which has occurred at a particular location to a person sitting in the car that got into an accident. Figure 6.3 depicts the snap shot of car tracking and notification system, which represent the latitude and longitude of car and a message expressing the accidental message.



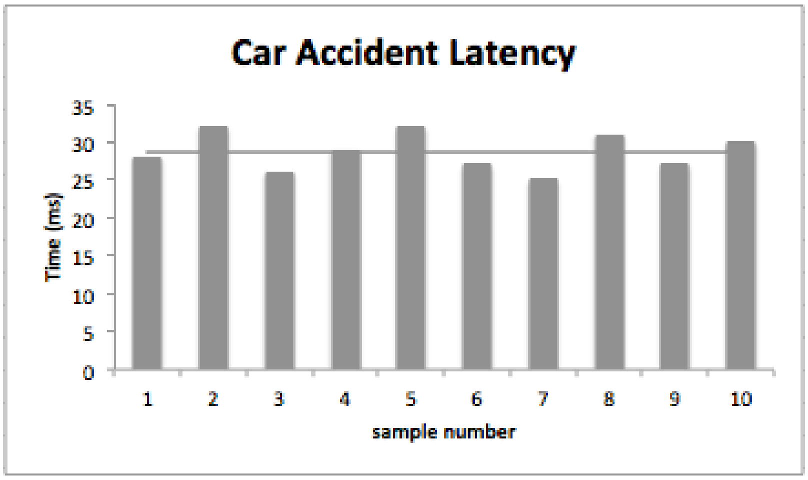
Current location
Lat: 33
Lon: 23.
Accident happend

Figure 6.3: Car accident location and notification application

All the close people of a particular person are notified about the car's location. This process is managed by the central management system, which dispatches the events to people if any event occurs. It was noticed the event processing time was acceptable. Since at the experimental time the real objects such as car, the notification sensor were not available, therefore, the prototype was just implemented as a simulation. The results obtained during this simulation are depicted in the following Graph 5. The average transmission time between the source and a mobile is about 27 ms, while the standard deviation is about:

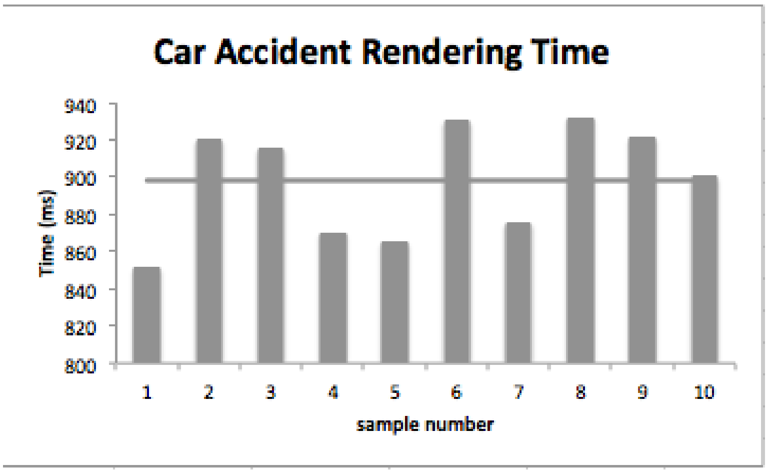
$$Std : s = 2.30ms$$

The standard deviation for the car accident is 2.30 which is the variation between the time i.e., PDV.



Graph 5: Transmission time between the car source and the mobile

Graph 6 represents the event processing time and the Google map rendering time. The variation represented in the graph may be due to the network delay and the behaviour of the system. The average rendering time for a particular event processing is about 897 ms, the horizontal line shows the average time.

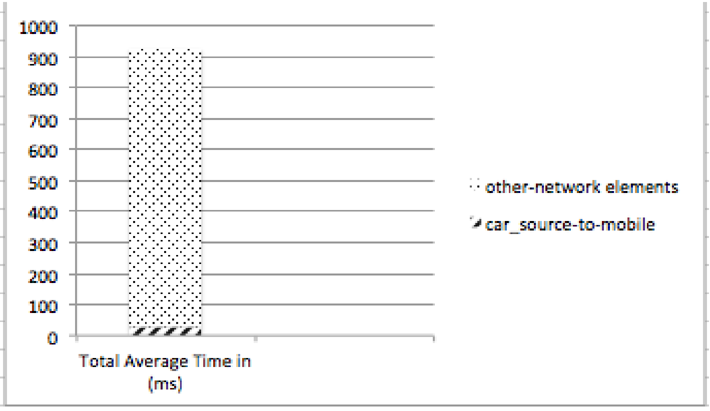


Graph 6: Google map rendering and event processing time

In Graph 6 the corresponding average time of the car accident event notification is measured, which represents the average QoE of the application users. The standard deviation for PDF is almost 27.68 and defined in the following equation:

$$Std : s = 27.68ms$$

The following Graph 7 represents the correlation time for the overall system, where 0.43 (almost 4% of the time) is for mobile to mobile while the rest of the time, which is about 96% of total correlation time, is the time taken by other network elements.



Graph 7: System correlation time

6.5 Mean Opinion Score (MOS) based on the Quality of Experience (QoE) Assessment

Based on the aforementioned results, an assessment was calculated by conducting a survey among different people of different ages and their mobile experiences. A formula was deduced to calculate the Mean Opinion Score (MOS) as people's interest level and variations in bitrates.

Applications' performance measurements are performed via objective and subjective methods. Objective methods are the actual data/information transmitted with no observation by the users, while subjective methods are the parameters, which are examined and judged by humans. The aforementioned results depicted in different graphs are actually the objective methods while the QoE based on users' personal experiences and feedback are the subjective methods.

Subjective methods are the metrics, which are directly concerned about the data collected as user's/observer's experiences, which quantify the user's perceptions about the QoE.

6.5.1 Methodology used for the subjective assessment

The methodology to setup the subjective test covered the variations in the latency when the source and the destination communicate. By changing the latency it enable the user's interest or QoE to be inferred. In the graphs above, the result for the corresponding scenario is obtained as the standard deviation, by changing the delay for different samples.

6.5.2 Selection of observers

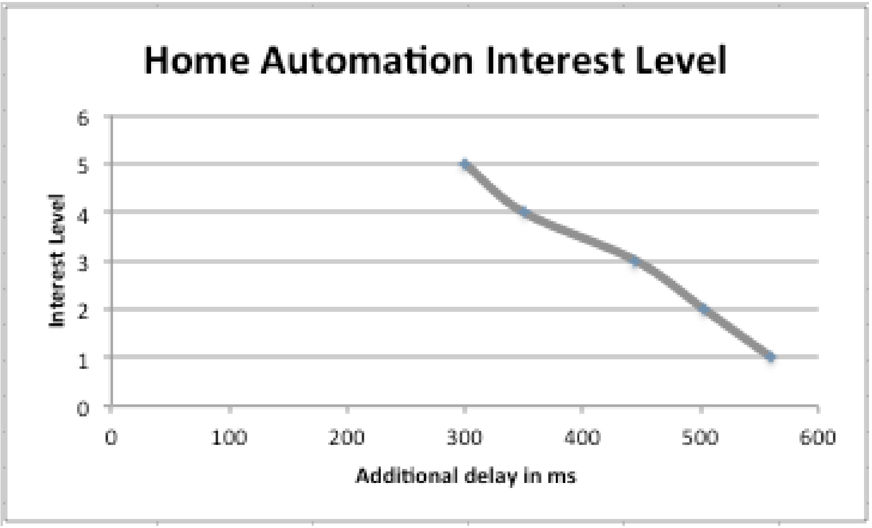
This was required in order to meet certain criteria to choose the observer, who participated in the MOS test. Therefore, to have an adequate assessment, the selected observers did not possess, any technical expertise, i.e., not directly concerned with any network bandwidth knowledge etc.

6.5.3 Subjective assessment

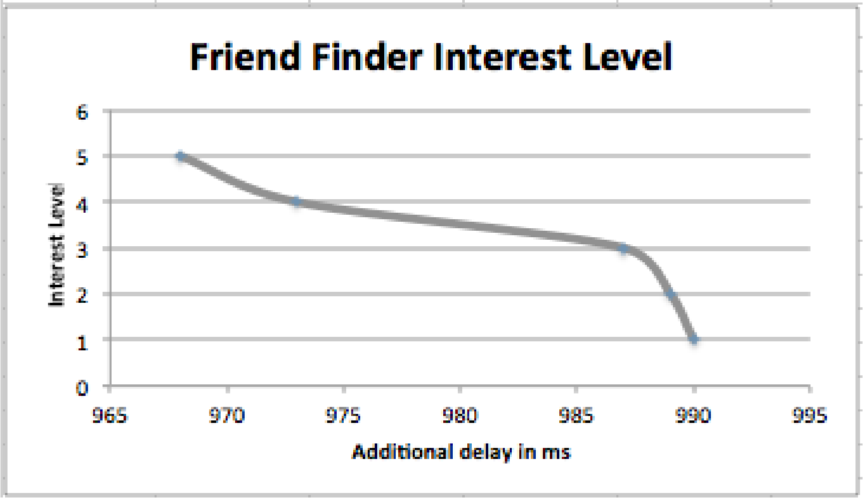
During the assessment test, by changing the latency, the observer ranked the quality of the network speed. For the Home Automation scenario the variation of the network delay time was retained as a scale between 0 and 0.5 ms, for the Friend finder scenario the scale was between 1 and 5ms and for the Car accident scenario the scale was kept between 1 and 100ms network delay time. The corresponding MOS results for all three scenarios were taken, based on the observer's best interest.

6.5.4 Data analysis

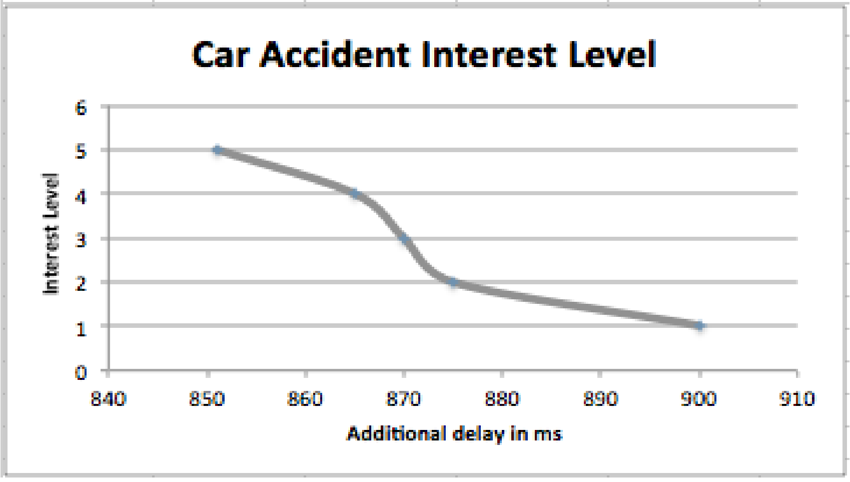
To collect the final MOS result, almost 31 rating responses were collected from the surveyed people, who were selected for subjective assessment. The collected 31 responses were the interest levels (IL) which ranged from 1 to 5. Graphs 8, 9 and 10 respectively depict the corresponding computed average results for the IL threshold. Graph 8, which depicts the Home Automation scenario, represents the MOS 300ms for IL. Graph 9 shows the Friend finder scenario where the MOS for IL is almost 968ms while Graph 10 depicts the Car Accident Notification scenario where the MOS for IL is nearly 850ms. The IL levels in all graphs show that, with the increase of latency time, the IL level decreases respectively and vice versa.



Graph 8: IL level for Home Automation



Graph 9: IL for Friend Finder



Graph 10: IL for Car Accident Notification

6.6 QoE Results obtained from survey

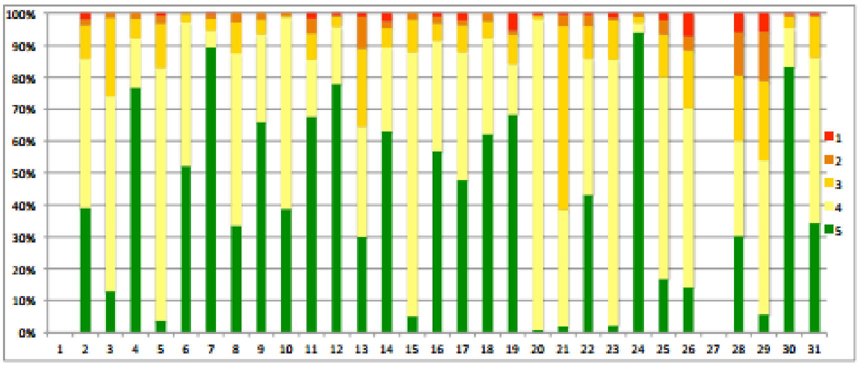
A survey was designed to discover the usage of mobile phones used by users. The survey was conducted based on the following parameters i.e., how long the users are using the phone, what type of mobile phone are they using, which carrier service are they connected to, in general the user's experience with the current mobile phone being used by him. The participants' selection procedure was mainly based on the user's having android phones, since this particular phone was used for the development purposes. The survey was conducted randomly on different days and contained the questions that are given bellow:

- User's were asked to rate the QoE in terms of (poor to excellent) based on the satisfaction

- User's location (home, office, indoor or outdoor) were also noticed

- Social interaction with the known people that how much the user is socialized

In order to obtained the correct rating as much as was possible, the users were asked to provide accurate ratings regarding satisfaction level/ subjective QoE in their minds. As described in (*section 6.5.4*) about 31 rating were collected from the people who did participate in the conducted survey. The graph 11 depicts that the high rating in ratio (4, 5) has more occurrences as compared to the lower rating ratio (1, 2). The mobile user participant found the QoE acceptable, depending on the coverage and the network service they were connected to.



Graph 11: distributed rating for the mobile users

7 Conclusions

This chapter describes the conclusions based on the experiments performed and their outputs. This part provides information regarding the problem statements and how they were tackled, moreover the issues faced during this thesis and goals that were set in chapter 1 and whether these goals were achieved by performing various experiments.

As discussed in chapter 1, the following goals were required to fulfil the credit requirements of this thesis:

1. Theory, background and related work explained in *chapter 2* were performed thoroughly by examining the state-of-the-art technologies and scientific publications.

2. Designed and developed case scenarios *(chapter 4 & 5)* as a proof of concepts. This goal was achieved by developing the application testing for the implemented real time applications on test bed. The apps were developed on the android platform and the servers were developed on a laptop which interacted with the corresponding apps.

3. Evaluating the applications with respect to their response time in a real time situation was discussed in the Results *(chapter 6)*. Various tests were performed on multiple users to determine their interest level, which shows the QoE of each user. Graphs represented in the Results *(chapter 6)* were drawn using the standard deviation.

4. Finding appropriate methods for strong QoE Mean Opinion Score evaluation test was conducted in order to test the QoE of various mobile users. A survey, as discussed in *chapter 6,* was conducted on different groups of people and it was noticed that by decreasing the latency rate the QoE increases. It means that the real time applications that have smaller latency rate are of more interest and are accepted by the end users.

Based on the challenges that were faced while studying and during the implementation it could be a great initiative to introduce the study for the participants to become familiar with the QoE monitoring software, which would assist them, as well as for the researchers who conduct the QoE experiment or surveys. Moreover, one suggestion for accurate measurements, one-way delay etc., could be to take measurements within the Operating System Kernel.

7.1 Discussion

In this report three scenarios are designed and implemented which improve the users' experiences. The first scenario is a low cost home automation system, in which a user having an android phone can remotely control the home appliances (e.g., lights, door etc.). The system is designed with a user-friendly interface, where the user can easily operate the application. The results and observations were taken as QoE and QoS bases. A number of observations were taken focusing on the PDV algorithm and it appeared that the system was adequate, reliable and fast considering QoE. The developed prototype maximizes the end user's satisfaction and the use of the Wi-Fi technology makes it possible to obtained benefits of the short-range wireless technology. The server inside the home directly communicates with the appliance via Zig-bee and new components are easy to install in the system. The system was tested on a multitude of different android devices running different android versions, thus providing its portability and compatibility. Moreover, this app is not only convenient to an ordinary person but it is more suitable for a disabled person to control the appliances remotely. The second scenario is to find the people for someone in a closed relation, such as friends and family members. With the developed app a person can easily trace the people to their location at the current time. Even though developed location-based mobile applications already exist, the developed application differs from the existing applications with respect to most of its properties such as performance, robustness and data storage, where data is stored in a key-value pair. The key-value pairs provide a better and robust storage and retrieval process. Moreover, our developed friend finder application, due to the JSON protocol, has a property such as openness, which may interact with other sensor devices such as surveillance devices. Our third implemented scenario is an innovative application in which friends and family members are automatically notified of a person who got into an accident in a car. In its current state the third scenario is only a simulation, since no real devices were available. However, our third developed application may also be extended to other entities used for assistance such as, for example, the ambulance authorities, etc., which can be added into the system. Questions may be asked as to, why the three application scenario were designed and tested? The answer for this is that; since the thesis mainly focuses on the QoE therefore, three different applications to target the same user were developed to calculate the average QoE of that particular end user. The QoS was taken, as the parameter to calculate the QoE, since this is an important factor to measure the QoE. The questionnaires, asked of the users, are mentioned in the Appendix chapter.

7.1.1 Comparison to the related work

In the paper "Bluetooth Remote Home Automation System Using Android Application" [18], the researchers have implemented a home automation system by employing the Bluetooth technology and, they present their prototype as a well designed application. However, in this thesis the prototype is developed by employing Wi-Fi because it is more secure, fast and far-ranging compared to Bluetooth. Moreover, the home server uses a proprietary protocol to communicate internally with the appliances; it uses the 802.15.4 low latency rate personal area network, which makes it possible that it is more secure as compared to Bluetooth.

In two research papers, "Location Based Services using Android Mobile" and "Towards a Position Sharing Approach for Location-based Services" [21, 22] the concepts of location-based services from an excellent architectural point of view have been discussed. In [21] the authors explain how the users' mobiles, by inter acting with a middle-ware system, trace the location of a particular person; moreover their system has the ability to retain the history of people, however they did not mention how the middle-ware maintains the buddies record. The prototype developed in this thesis not only keeps all the properties that [21] contains but additionally maintains the buddy list i.e., who is friends with whom and this is achieved by employing the key-value pair mechanism and the publisher/subscriber mechanism, which maintains the records of users as subscribers to the system. In [21] they did not conduct any analysis about the system performance and QoE. In contrast to [21] the prototype implemented in this thesis measures the system performance based on QoE. Moreover, the implemented prototype for communication uses the JSON protocol, which is easy and suitable to interact with other devices such as surveillance devices and other sensory devices. Hence it provides the openness features. In [22] the authors more or less used the same technology as in [21] but implemented additional security features, which falls outside the scope of this thesis.

7.1.2 Ethical impact on society

The first scenario i.e., Home Automation Scenario, the impact of this application could be expressed in such a way that it can improve the usage of mobile phone, not only for calling purposes, but also for remotely controlling the home appliances. But such applications, which have concerns with the security aspects, should be more secure in order to communicate. Another aspect to be noted is that these devices should not be lost, or snatched by someone otherwise anybody can attempt to gain access to the house control. If such applications become commercialized then these could be the ultimate replacement for physical door keys. However, for such applications the QoE should be acceptable for the end users. The second and third scenarios i.e., Friends finder and Car accident notification systems respectively, the impact of both applications on society could be “awareness” in the visualized form of their loved one’s location in real time. The visualized information could be the locality of friends or an unpleasant event happening to a family member or someone’s friends. The most important impact of such applications could be to “improve” the socialized interaction among each other.

7.2 Future work

For the home automation scenario, the enhancements to the system are possible to be made include the , integration of an intelligent controller that would have the ability to control the appliances based on various factors such as temperature, humidity etc. Moreover, as Google provides the cloud messaging services, that can be used to remotely control the home from any part of the world. Further extensions can be made such as vice detection features to control the home appliances.

In the friend finder application a query-processing feature can be added, which would facilitate the system administrator to monitor and analyse the involved entities. A real time car accident notification can be tested on real devices and by introducing the XMPP protocol, which is fast and reliable the system performance could be improved.

References

[1] Mobile Computing, http://www.theitlibrary.com/mobile_computing.html, Retrieved 2011-07-28.

[2] http://en.wikipedia.org/wiki/Context_awareness

[3] Lovett, Tom; O'Neill, Eamonn, “Mobile Context Awareness”

[4] http://www.wikinomics.com/blog/index.php/2010/01/21/what-you-need-when-you-need-it-how-context-aware-machines-will-change-how-we-access-information/

[5] Ward, Andy, Alan Jones, and Andy Hopper. "A new location technique for the active office." Personal Communications, IEEE 4.5 (1997): 42-47.

[6] Chen, Guanling, and David Kotz. A survey of context-aware mobile computing research. Vol. 1. No. 2.1. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, 2000.

[7] Norman H. Cohen, James etl, “Building Context-Aware Applications with Context Weaver”, RC23388 (W0410-156) October 22, 2004

[8] http://en.wikipedia.org/wiki/Real-time_computing

[9] Ben-Ari, M., "Principles of Concurrent and Distributed Programming", Prentice Hall, 1990. ISBN 0-13-711821-X. Ch16, Page 164.

[10] Bloem, *Me-the Media*, Sogeti, 2009

[11] L. Jedrzejczyk, “On The Impact of Real-Time Feedback on Users' Behaviour in Mobile Location-Sharing Applications (Buddy Tracker)”, The Open University/University of Limerick, 12 pages.

[12] http://en.wikipedia.org/wiki/Notification_system

[13] http://www.transportrankings.com/GPS-tracking-system-in-auto-transport-industry.php

[14] J. Palhais, “Quality of Experience Assessment in Internet TV”, Instituto Superior Tecnico, 14 pages.

[15] R. Jain, “Quality of experience”, Georgia Institute of Technology, 2 pages.

[16] http://www.wica.intec.ugent.be/research/quality-of-experience

[17] Katarzyna Wac, Anind K. Dey “Mobile User Experience Beyond the Labortory: Towards a Methodology for QoE-QoS Evaluation in Natural User Environments”

[18] R.A Ramlee, M.H Leong, etl “Bluetooth Remote Home Automation System Using Android Application” The International Journal of Engineering And Science (IJES), ISSN: 2319 – 1813 ISBN: 2319 – 1805

[19] http://mobilesociety.typepad.com/mobile_life/2013/01/wi-fi-direct-vs-bluetooth.html

[20] http://forums.androidcentral.com/google-nexus-7-tablet/192414-tethering-what-advantages-wifi-vs-bluetooth.html

[21] Amit Khuswaha, Vineet Khuswaha, etl. “Location Based Services using Android Mobile”. International Journal of Advances in Engineering & Technology, Mar 2011.

[22] Frank Durr, Marius Wernke, etl. “Towards a Position Sharing Approach for Location-based Services”

[23] Tahat, Ashraf, et al. "Android-based universal vehicle diagnostic and tracking system." Consumer Electronics (ISCE), 2012 IEEE 16th International Symposium on. IEEE, 2012.

[24] Acharya, Ayan, Jangwon Lee, and An Chen. "Real Time Car Detection and Tracking in Mobile Devices."

[25] Y. Freund and R.E. Schapire. A decision-theoretic general- ization of on-line learning and an application to boosting. *In EuroCOLT 95*, pages 2337. Springer-Verlag, 1995.

Appendix: Result of questionnaire survey

S	**Gender**	**Profession**	**Age**	**Phone Type**	**QoE Survey**	**MOS**	**Total No of Rat.**
1	M	IT Student	20-22	Android	5	4,7	3
2	M	IT Student	18-23	Android	5	3,0	2
3	F	Business & Marketing Student	18-20	Samsung	3	4,1	4
4	M	Electronic Student	21-23	I Phone	5	3,4	3
5	M	IT Student	22-23	Android	4	5,2	4
6	M	Social Sciences Student	20-20	Android	2	3,7	3
7	F	Arts Student	19-23	Nokia	5	5,0	4
8	M	IT Student	20-22	Android	5	5,7	3
9	M	MBA Student	21-24	Android	3	4,3	2
10	M	Electronic Student	23-27	Android	4	5,9	3