

# **SNAPDIRECTION ANDROID APPLICATION**



Session 2011-2015

**BY**  
**SAAD REHMAN**  
**AND**  
**ALI RAZA MIR**  
Bachelor of Science in Computer Science

**Department of Computer Science**  
**City University of Science and Information Technology**  
**Peshawar, Pakistan**  
**August, 2015**

# **SNAPDIRECTION ANDROID APPLICATION**



**Session 2011-2015**

**By  
SAAD REHMAN  
AND  
ALI RAZA MIR**

**Supervised By  
MR.MUMTAZ ALI**

**Department of Computer Science  
City University of Science and Information Technology  
Peshawar, Pakistan  
August, 2015**

# **SNAPDIRECTION ANDROID APPLICATION**

*By*  
SAAD REHMAN (4971)  
*and*  
ALI RAZA MIR (4855)

## **CERTIFICATE**

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR  
THE DEGREE OF BACHLOER OF COMPUTER SCIENCE/SOFTWARE ENGINEERING

We accept this thesis as conforming to the required standards

1. \_\_\_\_\_  
(Supervisor)  
Mr. xyz

2. \_\_\_\_\_  
(Internal Examiner)

3. \_\_\_\_\_  
(External Examiner)

4. \_\_\_\_\_  
(Head of the Department)

5. \_\_\_\_\_  
Coordinator FYP

Date. \_\_\_\_\_  
(Approved)

**Department of Computer Science**  
**City University of Science and Information Technology**  
**Peshawar, Pakistan**  
**August, 2015**

# Abstract

Mobile phones are nowadays far more than merely devices to communicate with. Especially, Smartphone's are products that help to make our work and everyday life easier. Based on concept of communicating peoples we are focusing on developing an Android Application SnapDirection. Most of the information features available in the applications are location & time based. SnapDirection is a mobile application based on ANDROID operating system. This application will send the information of the user to intended number according to various tasks based on Time.

The main purpose of application which we are going to develop is a revolutionary way of notifying intended numbers while traveling. As sending SMS or calling your friends and family just to tell them about your location while driving is dangerous. It sends SMS triggered by time. It auto sends an SMS to selected recipient. Our project is divided into 3 features.

The first feature is direction which means that when user enters his/her traveling destination the app calculates the estimated distance, time and path on the map. The second feature which is very important is SNAPSMS which is used for storing information regarding destination in defined time interval. The third feature is ICE which is used for sending location stored to second person who enquires the location of first person.

# **Declaration**

We hereby declare that this application as a whole has not been copied from any source. Further, we declare that we have developed this software and the accompanied reports entirely on the basis of our personal efforts, made under the sincere guidance of our supervisor. No portion of this work presented in this report has been submitted in support of an application for another degree or qualification of this or any other university or institution learning.

**Saad Rehman**

**Ali Raza Mir**

**August, 2015**

## **Dedication**

We dedicate this project to our beloved parents who are dearer to us and our teachers whom are like candle light on the way to our destiny.

## **Acknowledgements**

Perpetual admiration to almighty Allah, the creator and the perfect guide to all the mankind towards the right path. He is the most, merciful, the most beneficent, who bestowed upon the possibility and tenacity to complete the work assigned. We would like to express my deep felt gratitude to our supervisor Mr. Mumtaz Ali for his patience and guidance. We would also take this opportunity to thank our parents, our friends and our school and college teachers without their guidance and support we couldn't reach to this stage of my life. We are also very thankful to our brothers and sisters for their continuous love and support through a difficult and trying period of my life.

We would also take this opportunity to thank my friends for their support.

# Contents

Abstract .....	i
Declaration.....	ii
Dedication .....	iii
Acknowledgements .....	iv
List of Tables.....	x
List of Figures .....	xi
Chapter 1 .....	1
Introduction .....	1
1.1 Motivation.....	2
1.2 Scope & Objective .....	2
1.3 Problem statement .....	3
1.4 Problem Solution.....	3
1.4.1 First module:.....	4
1.4.2 Second module: .....	4
1.4.3 Third module.....	4
1.5 Project Structure .....	4
1.5.1 Direction .....	4
➤ Searching .....	5
➤ Directional Map .....	5
1.5.2 SNAPSMS.....	5



➤ Selecting Location.....	5
➤ SnapSMS.....	6
1.5.3 ICE.....	6
➤ ICE Number .....	6
➤ ICE Message .....	6
1.6 Background.....	7
1.7 Tools & Techniques .....	7
1.7.1 Android Studio IDE.....	7
1.7.2 Android AVD Manager .....	8
1.7.3 Photoshop CS6.....	9
1.7.4 Microsoft Project .....	10
1.7.5 Microsoft Visio .....	11
1.7.6 Microsoft Word .....	12
1.7.7 Google Maps API: .....	12
1.7.8 Places API:.....	13
1.8 Thesis Outline.....	14
Chapter 2 .....	16
Software Project Planning.....	16
2.1 Project Objectives.....	16
2.2 Project Characteristics .....	17
2.3 Products & activities.....	17
2.4 Activities Estimation.....	18

a) Milestone Plan .....	18
b) Documentation Plan .....	19
2.5 Activities risks .....	20
2.6 Resources Allocation .....	21
a) Resource Plan .....	21
b) Gantt chart.....	22
Chapter 3 .....	23
System Requirements .....	23
3.1 Functional Requirements.....	23
3.2 Non-Functional Requirements.....	24
3.3 Use-Case.....	24
3.3.1 Use Cases Actors .....	24
3.3.2 Associations .....	24
Chapter 4 .....	26
System Design .....	26
4.1 Logical design.....	26
4.1.1 DFD .....	26
4.2 DFD Levels Used .....	27
4.2.1 Level 0 DFD .....	27
4.2.2 Information Flow of Level 0 DFD.....	27
4.2.3 Level 1 DFD .....	28
4.2.4 Information Flow of Level 1 DFD.....	29

➤ Main Menu .....	29
➤ Direction .....	29
➤ SnapSMS.....	29
➤ ICE .....	29
➤ SMS Receive .....	30
4.3 Physical design.....	30
4.3.1 Interface Design.....	30
4.3.2 Application Startup .....	32
4.3.3 Application Main Page:.....	33
4.3.4 Direction View:.....	34
4.3.5 SNAPSMS:.....	35
4.3.6 Set Timer:.....	36
4.3.7 Cancel SMS Timer:.....	37
4.3.8 ICE:.....	38
4.3.9 Emergency System:.....	39
4.3.10 ICE Message:.....	40
Chapter 5 .....	41
System Testing.....	41
5.1 Unit and Integration Testing .....	41
5.2 User Acceptance Testing.....	41
5.2.1 Definition.....	41
5.3 Usability Testing .....	44

5.3.1 Effectiveness .....	44
5.3.2 Efficiency .....	44
5.3.3 Satisfaction .....	44
Chapter 6 .....	45
Conclusion .....	45
References .....	46

## List of Tables

Table 2.1: Project Phases.....	17
Table 2.2: Milestone Plan .....	18
Table 2.3: Documentation Plan .....	19
Table 2.4: Risk & Minimization technique.....	20
Table 2.5: Resource Plan.....	21
Table 5.1: User Task .....	42
Table 5.2: Measuring System Effectiveness .....	43
Table 5.3: Measuring System Efficiency.....	43

# List of Figures

Figure 1.1: Android Studio .....	8
Figure 1.2: Android AVD Manager .....	9
Figure 1.3: Photoshop CS6 .....	10
Figure 1.4: Microsoft Project for Gantt chart .....	11
Figure 1.5: Microsoft Visio for UML Diagram .....	11
Figure 1.6: Microsoft Word for Documentation .....	12
Figure 1.7: Google MAPS v2.....	13
Figure 1.8: Google Places API.....	14
Figure 2.1: Gantt chart .....	22
Figure 3.1: Use Case Diagram 1 .....	25
Figure 3.2: Use Case Diagram 2 .....	25
Figure 4.1: Level 0 DFD represents the Initial System and User Interaction.....	27
Figure 4.2: Level 1 DFD represents the Entire General Working and Dataflow .....	28
Figure 4.3: Application Startup Screen .....	32
Figure 4.4: Main Menu .....	33
Figure 4.5: Route with Distance & Duration .....	34
Figure 4.6: Selecting the Destination .....	35
Figure 4.7: Add Contact & Set Time .....	36
Figure 4.8: Stop Timer .....	37
Figure 4.9: Select Number and Turn Service ON .....	38

Figure 4.10: Add Number & Turn ON ICE system.....	39
Figure 4.11: Create ICE Message.....	40

# Chapter 1

## Introduction

**Android** is a powerful Operating System supporting a large number of applications in Smart Phones. These applications make life more comfortable and advanced for the users. Hardware's that support Android is mainly based on ARM architecture platform.

Android originates from a small software company, acquired by Google and is now owned by Open Handset Alliance (OHA), where Google is a member. OHA has over a hundred member companies such as mobile operators, semiconductor companies, handset manufacturers, software companies and commercialization companies. Driven through the Apache License, anyone can use the Android Software Development Kit (SDK) to develop applications to run on the Android OS. Especially interesting for Android is its use of common non-proprietary techniques such as the Java programming language in combination with Extensible Markup Language (XML). This makes it open, simple and easy to use for all [1].

Android is an open system, and is free to use by anyone. A handset manufacturer can use Android if they follow the agreement stated in the Software Development Kit. There are no restrictions or requirement for the handset manufacturer to share their extensions with anyone else. Android is a software environment and not a hardware platform, which includes an OS, built on Linux kernel-based OS hosting the Dalvik virtual machine (DVM). The Dalvik virtual machine runs Android applications as instances of the virtual machine. Android contains a rich user interface, application framework, Java class libraries and multimedia support [2]. Android also comes with built-in applications containing features such as short message service functionality (messaging), phone capabilities and an address book (contacts).



The introduction chapter of this document gives us the notion about the idea of the software project that is going to be developed.

## **1.1 Motivation**

We will put it in simple words, if we see out to our world population its growing daily and as much as the population grows, a lot of peoples are using smart phones, if someone went some different place without inform any one and get some serious problem like i.e. car accident etc. And no one there who helps him. To avoid these kinds of issues, we introduce a SnapDirection an android application which perform these tasks.

Emergency auto reply system. Auto reply system of application will work on some specific code given by user and also specific number defined by user.

Directional and location based Alert app. We are using Google map v2 in our application for directions and locations with GPS. For Notification of location we are using GPRS services and also SMS services in android. We are going to use Google API key for map.

Time base notification of location to specific number. The application will check the signal strength and battery levels of mobile and if level is low it will send notification to user defined number.

## **1.2 Scope & Objective**

In this bachelor thesis we are focusing on developing an application, based on connecting and information sharing concept. The purpose of this project is to first and foremost gain further experience in Android Application Development as our passion. Furthermore we wish to increase skills in different areas of Android Application Development such as working with maps, services, GPS and developing advance applications etc. We have studied, in detail, quite a few of the different steps of Android application, such as, development and implementation of application.

Proposed android application is direction system which main purpose is that user can get all the features of direction and emergency system in one application and there is no need that user will install apps of tracking separately all the requirements related to direction will be easily available for the user in this application . Moreover our application will send information of user's location, battery percentage and signal strength in specific time intervals defined by user.

In our SNAPDIRECTION which is android application consists of three modules are as under:

- Direction
- SNAPSMS
- ICE

### **1.3 Problem statement**

The problem due to which we are developing SnapDirection application are sending SMS or calling your friends and family just to tell them about your location while driving is dangerous. Keep an eye on children's location and activities is impossible in this busy world. In some previous available apps there is no mechanism of auto reply system I.C.E.

Sometimes because of low battery and signal strength there are a lot of problems for families and beloved ones due to no contact and information. While driving if battery level and signal strength is going low and the person don't have any idea about it to and the family members want to contact user and failed to contact.

### **1.4 Problem Solution**

The solution for given problems that to develop SnapDirection which will be divided into three modules:

### **1.4.1 First module:**

Directions are highlighted on the map to easily display the current location with duration and distance of selected destination.

### **1.4.2 Second module:**

User will define his/her location where he/she is going. Specific number of people he wants to share his locations and also define time interval. GPS will monitor the location of person and send it to given numbers in defined time interval.

### **1.4.3 Third module**

IN case of emergency the application will auto reply with the current location of user. The application will match the ICE number and code word defined by user.

## **1.5 Project Structure**

In This application SNAPDIRECTION that will work on the android based smart phones. There are basically three main modules which manage all the requirements.

- Direction
- SNAPSMS
- ICE

### **1.5.1 Direction**

Direction used to know the easiest and simplest way of traveling, directions are highlighted on the map to easily display the current location with duration and distance of selected destination. In this module there are two screens designed for the users. The names of the screens are as under:

- Searching
- Directional Map

➤ **Searching**

When user will select that screen so user will have option to enter location of their own choice. A keyboard will appear and user will give address and our application will search it. The auto-complete text field will help user to show relevant matching results of user input.

➤ **Directional Map**

In the directional map the searched location is shown starting from current location to searched destination, with path highlighted. Calculated time and distance will also be shown on screen.

## **1.5.2 SNAPSMS**

SNAPSMS used for creating message of current location with time, battery level and signal strength. This module will send the information of user to his/her defined contact with specific time interval. In this module there are two screens designed for the users. The names of the screens are as under.

- Selecting Location
- SNAPSMS

➤ **Selecting Location**

When user will select that screen so user will have option to enter location of their own choice. A keyboard will appear and user will give address and our application will search it. The location which user search is displayed with a distance and duration of that address with mark point on the map. On clicking this mark point the user will be directed to SnapSMS screen.

### ➤ **SnapSMS**

This screen will help user to setup SMS setting like selecting number of recipient, time duration of message. After setting up the user will press start button to start the service.

### **1.5.3 ICE**

In case of emergency ICE was used for sending SMS if the user is not responding so the selected recipient can check the location and battery percentage of user by simply sending message with the keyword defined by user, before accessing this application the user must active their GPS location mode on their Smartphone's, when selecting ICE module for the first time the popup message will appears "ICE number not set" go to setting and insert ICE Number, ICE Message.

- ICE Number
- ICE Message

### ➤ **ICE Number**

In settings the user will set the ICE number this number will be able to get the information of user's location and battery percentage.

### ➤ **ICE Message**

The ICE Message defined by the user used for verification and auto reply to the person who wants to get information regarding the user's location in simple SMS.

In reply message the current location and battery level were added.

## 1.6 Background

Current Applications like place SMS, trackers, and GPS app are mostly paid on android market. And secondly these apps are together not available on the market and also mostly applications on Google play of tracking person, SMS tracking are that user will manually write number in our application the user can select from contacts. Some application triggered SMS based on location of user when user enter or leave specific area what if user didn't reach the location or some emergency might happen. In most ICE applications on Google play the user will have to perform some action in emergency. There isn't any application on store which uses auto reply system with specific SMS criteria.

## 1.7 Tools & Techniques

Different Tools that are used for the software development are ADT, Eclipse, Dreamweaver, MS Project, MS Power point, MS Word. MS Visio is used to draw diagrams and MS Project for drawing Gantt charts and for different activities like scheduling, deliverables etc.

Following are the tools and technologies used in the proposed project.

### 1.7.1 Android Studio IDE

**Android Studio** is an integrated development environment (IDE) for developing on the Android platform [3]. We use android studio for coding and debugging of SNAPDIRECTION.

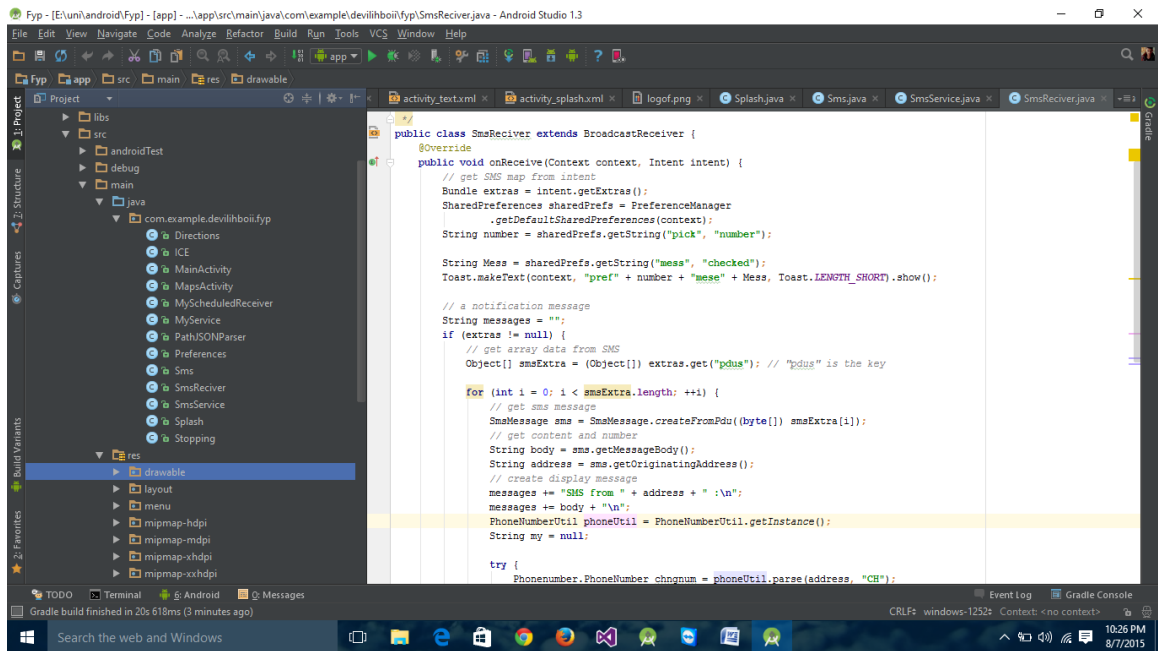


Figure 1.1: Android Studio

## 1.7.2 Android AVD Manager

The AVD Manager provides a graphical user interface in which you can create and manage Android Virtual Devices (AVDs), which are required by the Android Emulator. You can launch the AVD Manager in one of the following ways [8]:

- In Android Studio: select Tools > Android >AVD Manager, or click the AVD Manager icon in the toolbar.
- In other IDEs: Navigate to your SDK's tools/ directory and execute android avd.

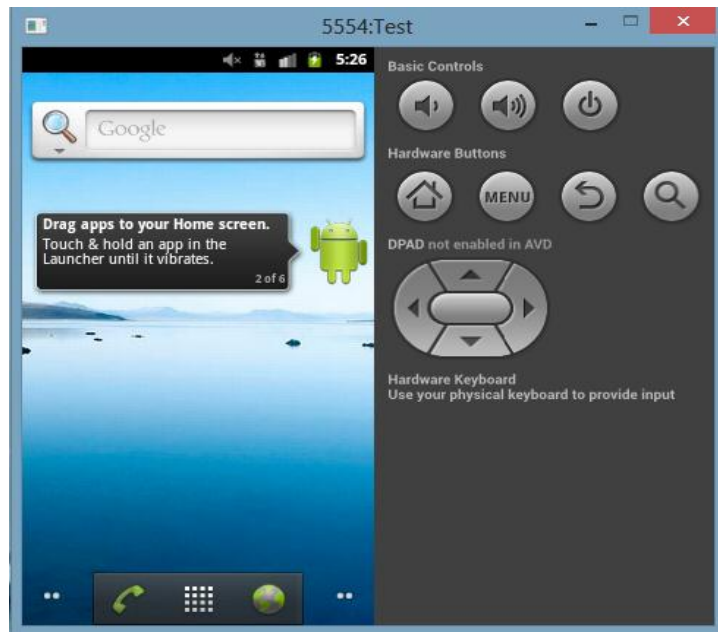


Figure 1.2: Android AVD Manager

### 1.7.3 Photoshop CS6

**Adobe Photoshop** is a raster graphics editor developed and published by Adobe Systems for Windows and OS X [9]. We use adobe Photoshop for designing of logo, icons and background images of SNAPDIRECTION.



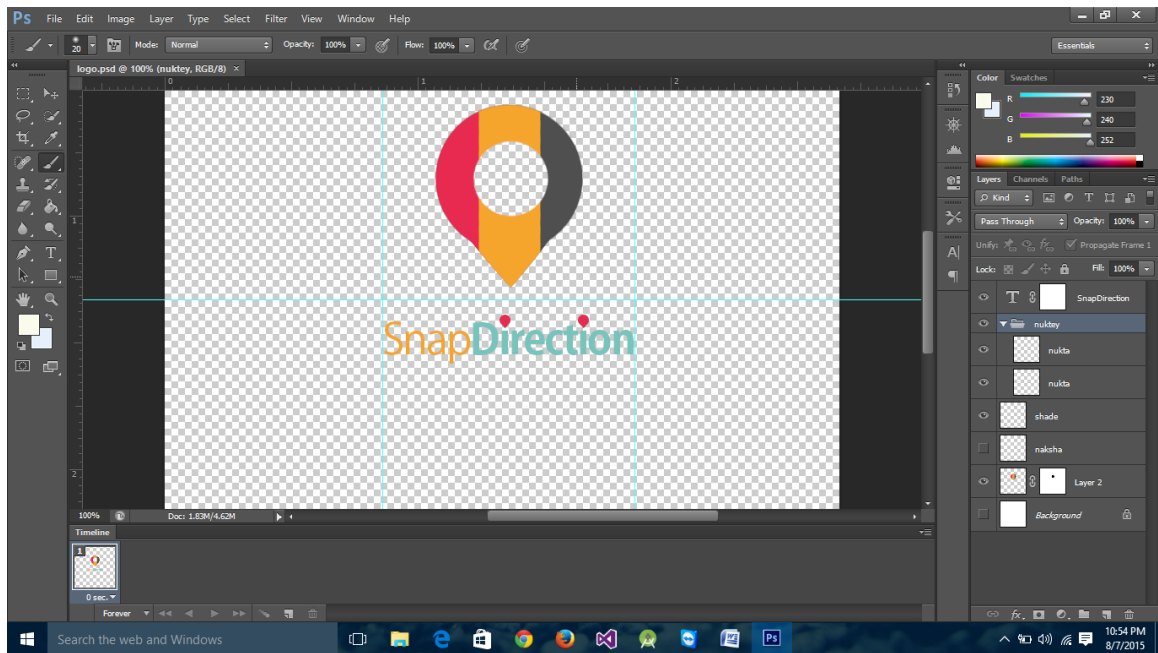


Figure 1.3: Photoshop CS6

### 1.7.4 Microsoft Project

Microsoft Project is a software application sold by Microsoft that provides project management tools to manage projects [10]. Gantt chart of this project is created in Microsoft project.

.

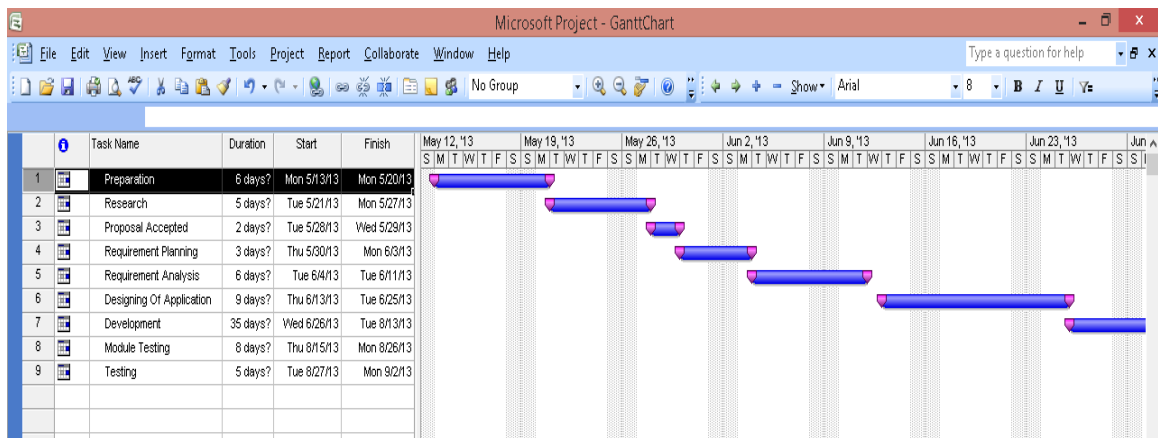


Figure 1.4: Microsoft Project for Gantt chart

## 1.7.5 Microsoft Visio

Microsoft Visio is used to diagram complicated processes, flowcharts, software, and organizational charts. I also used Microsoft Visio for creating different diagrams of my application project and making full use of those diagrams related to my project.

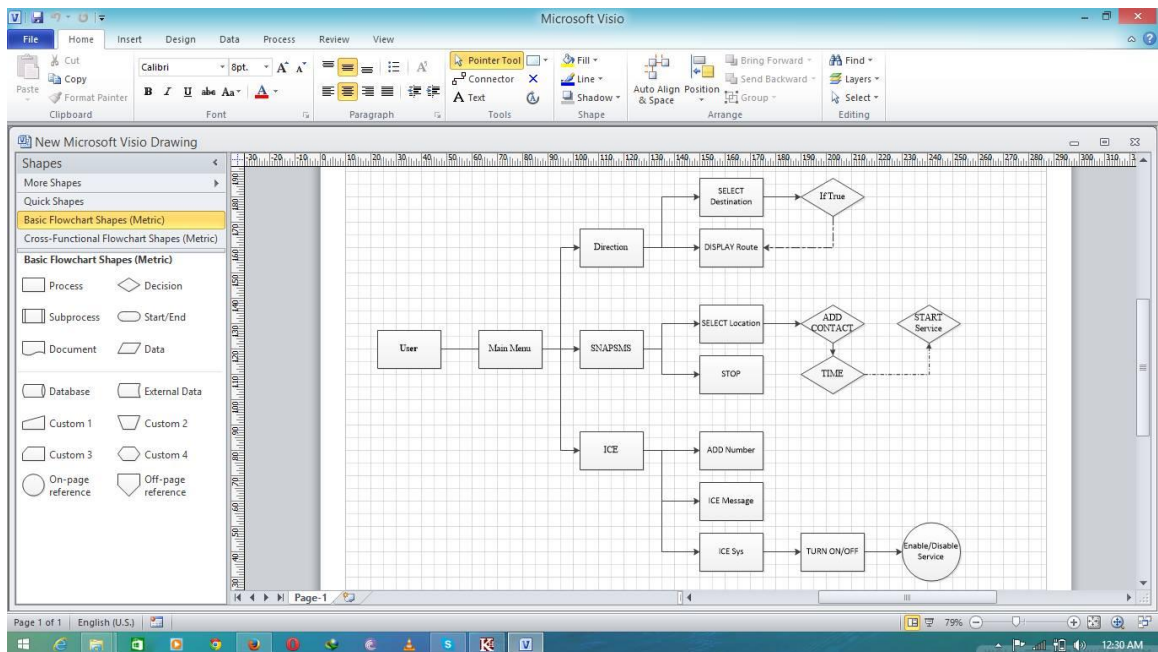


Figure 1.5: Microsoft Visio for UML Diagram

## 1.7.6 Microsoft Word

Microsoft Word is used for different purposes but we have used this tool for making thesis of our software project.

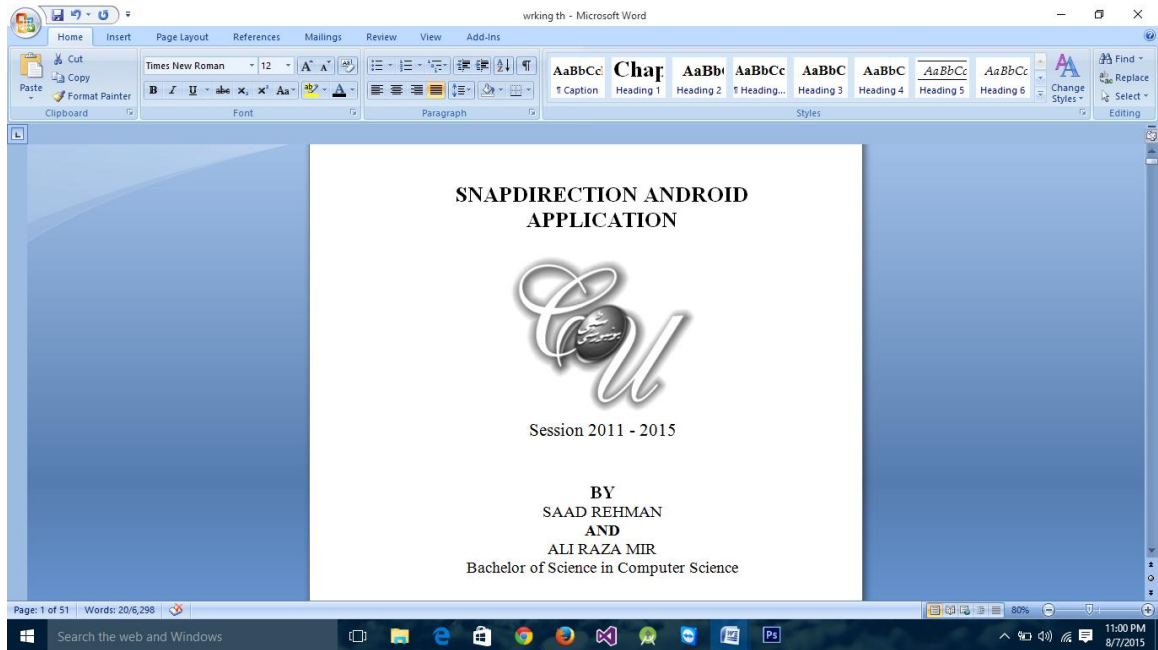
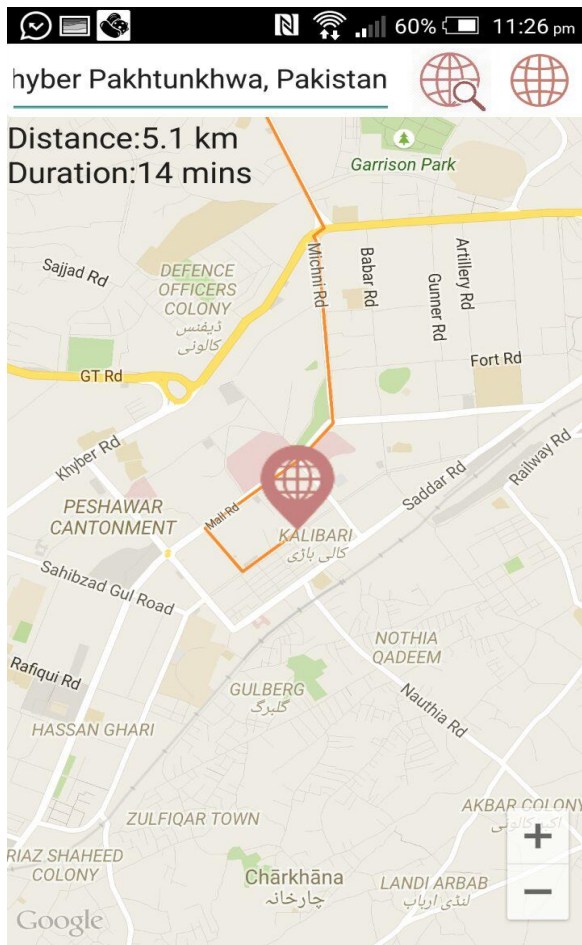


Figure 1.6: Microsoft Word for Documentation

## 1.7.7 Google Maps API:

Google Maps Android API, you can add maps based on Google Maps data to your application [11]. Google map is used in our application for directions and locations.



**Figure 1.7: Google MAPS v2**

### 1.7.8 Places API:

The main entry points for the Google Places API for Android are the `PlacePicker` UI widget, the `GeoDataApi`, and the `PlaceDetectionApi` [12]. Places API is used for searching location address as auto complete text field.

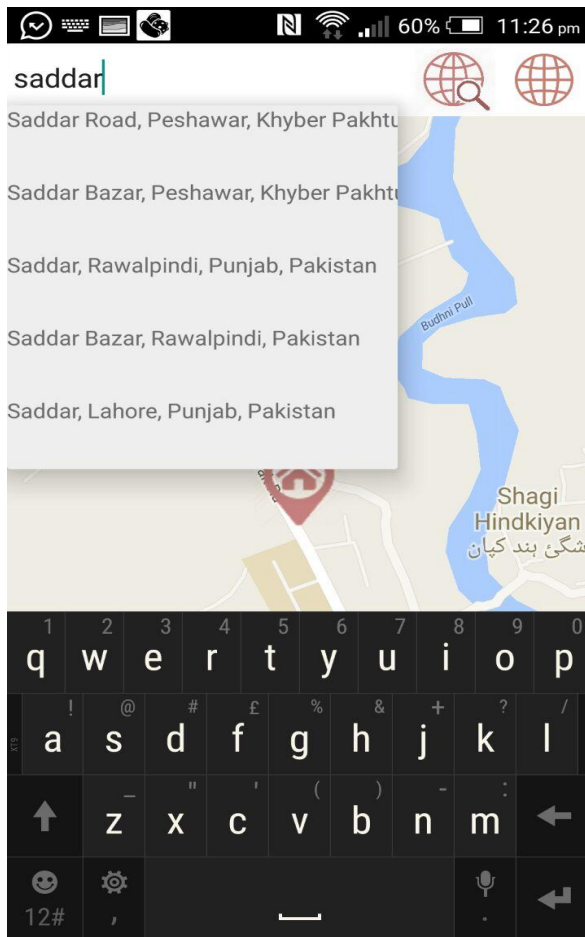


Figure 1.8: Google Places API

## 1.8 Thesis Outline

In this chapter we gave an introduction of the project, discussed project structure and described the main components that would constitute the project. We also discuss about the tools and technologies used in designing our Application.

In chapter 2 we will concentrate on software project plan and discuss all the matter associated to project management like identifying project deliverables and how to plan the project.

In Chapter 3, we discuss what requirements need to be fulfilled and what features should the product exhibit we also tried to give a complete description of the behavior of a system to be developed. It also contains functional and non-functional requirements.

In chapter 4 we discuss the system logical and physical design of our application. The interfaces and data flow diagram is discussed in this chapter.

In Chapter 5 illustrates the test plans for testing the Application. This chapter also contains a comprehensive list of tests that will provide help to complete the product productively without any bugs. In this chapter we categorized the requirements to be tested, outline the testing approach to be used, identify the test cases and identify the anticipated results for each test.

# Chapter 2

## Software Project Planning

The project plan reflects the current status of all project activities and is used to monitor and control the project [13]. In this chapter different software project planning and management activities are discussed and different plans like documentation plan milestone plans and project deliverables are identified. This chapter throws light on software project plan, in which all matter associated to project management is discussed. Project deliverables, how to plan the project. The project plan generally deals with the following:

- i. The resources available to the project
- ii. The work breakdown
- iii. A schedule for the work

### 2.1 Project Objectives

The main purpose of our project is to design an application which would be android base application for anyone to use with quite a number of nice features. The application which we are going to develop is SNAPDIRECTION, which is a revolutionary way of notifying loved ones while traveling. It will send SMS triggered by time. It will auto sends an SMS to selected recipient. Our project is divided into 3 features. The first feature is direction which means that when user enters his/her traveling destination the app calculates the estimated distance, time and path on the map. The second feature is SNAPSMS which is used for storing information regarding destination in defined time interval. The third feature is ICE which is used for sending location stored to second person who enquires the location of first person.

Our aim is to developed market base application to facilitate users which will be used for notifying and also for checking your loved ones location status and some other information.

## 2.2 Project Characteristics

There are various applications available on Google play store for notifying and emergency systems but most of them are paid. And secondly these apps are together not available on the market and also mostly applications on Google play. In current available application triggered SMS based on location of user when user enter or leave specific area what if user didn't reach the location or some emergency might happen.

Our proposed application overcomes the limitations of currently available application like our application triggered SMS based on time interval defined by user. The ICE system of our application will auto replay the information of user when the SMS is received by user defined number simply by matching the text and number. User will just have to set the ICE number and ICE text.

## 2.3 Products & activities

All the information on the activities and expected documented output from each project phase is shown in Table 2.1.

**Table 2.1: Project Phases**

Phase	Activities	Output
Documentation Phase	Project Planning	Project Plan
Application Design Phase	Application Design Documentation	Application Design Document
Pre-Alpha Phase	Test Plan, Basic	Test Plan
Alpha Phase	Google maps, Directions, time triggered SMS	Alpha Version Product
Beta Phase	SMS Broadcast Receiver, Auto replay system	Beta Version Product



Postmortem Phase	Bug fixed, Integration Test, System testing	Test Report/ (Final Product)
Post-Product Phase	Final Report, User manual	Final Report /User Manual
Presentation Phase	Present the project	Presentation.ppt

## 2.4 Activities Estimation

### a) Milestone Plan

Table 2.2 provides information on milestones and their expected completion date.

**Table 2.2: Milestone Plan**

<b>Milestone</b>	<b>Scope</b>	<b>Output</b>	<b>Completion Date</b>
Milestone 1	Pre-production	Project Plan	30 <sup>th</sup> Nov 2014
Milestone 2	Application Design	Application Design Document	28 <sup>th</sup> Feb 2015
Milestone 3	Test Planning	Test Plan	1 <sup>st</sup> April 2015
Milestone 4	First-playable	Working prototype of Application	5 <sup>th</sup> April 2015
Milestone 5	Alpha Testing	Prototype of application updated with some features	16 <sup>th</sup> April 2015
Milestone 6	Beta Testing	Prototype of application updated with user interface	10 <sup>th</sup> June 2015
Milestone 7	Application Release	Final Report/ User Manual	20 <sup>th</sup> July 2015

**b) Documentation Plan**

Table 2.3 provides information on dates when the documents related to the project will be delivered.

**Table 2.3: Documentation Plan**

<b>Document</b>	<b>Purpose</b>	<b>Date</b>
Project Plan	Organize the project	30 <sup>th</sup> Nov 2014
Application Design Document	Describe the design of application	10 <sup>th</sup> Feb 2015
Test Plan	Define all test activities	28 <sup>th</sup> March 2015
Test Report	Define all test cases	16 <sup>th</sup> April 2015
Final Report	Summarize the project	14 <sup>th</sup> July 2015
User Manual	Describe how to use the Application	07 <sup>th</sup> August 2015

## 2.5 Activities risks

Table 2.5 shows the list of risks and also states the risk avoiding and minimizing techniques to corresponding risks.

**Table 2.4: Risk & Minimization technique**

Risks	Risk Avoiding & minimizing technique
Getting ill	If any of us becomes ill with a sickness more serious than a simple cold. <b>Avoidance:</b> As a single person, not much. Sicknesses happen. <b>Minimize the Risk:</b> Have my responsibilities overlapped. That way, if I becomes too ill to work, there won't be a complete section of the project that goes unworked on.
Programming difficulties	A feature we want to add is difficult for me to implement. <b>Avoidance:</b> Read tutorials on how to code better in the Android Application Language. <b>Minimize the risk:</b> Making the design suitable for features which we will be able to competently code.
App Development Tool does not have the functionality we need.	A needed or desired function for my Application is not one that is supported by App Development Tool. <b>Avoidance:</b> Check the available functions within Tool. If the function is not directly supported, search for alternatives. <b>Minimize the risk:</b> Do not try to implement anything too detailed, or drop the function altogether.
can't find appropriate design	If we would unable to locate a suitable design that we want to use for a section in the application. <b>Avoidance:</b> Choose a simpler design. <b>Minimize the risk:</b> Create the interface design myself.

## 2.6 Resources Allocation

### a) Resource Plan

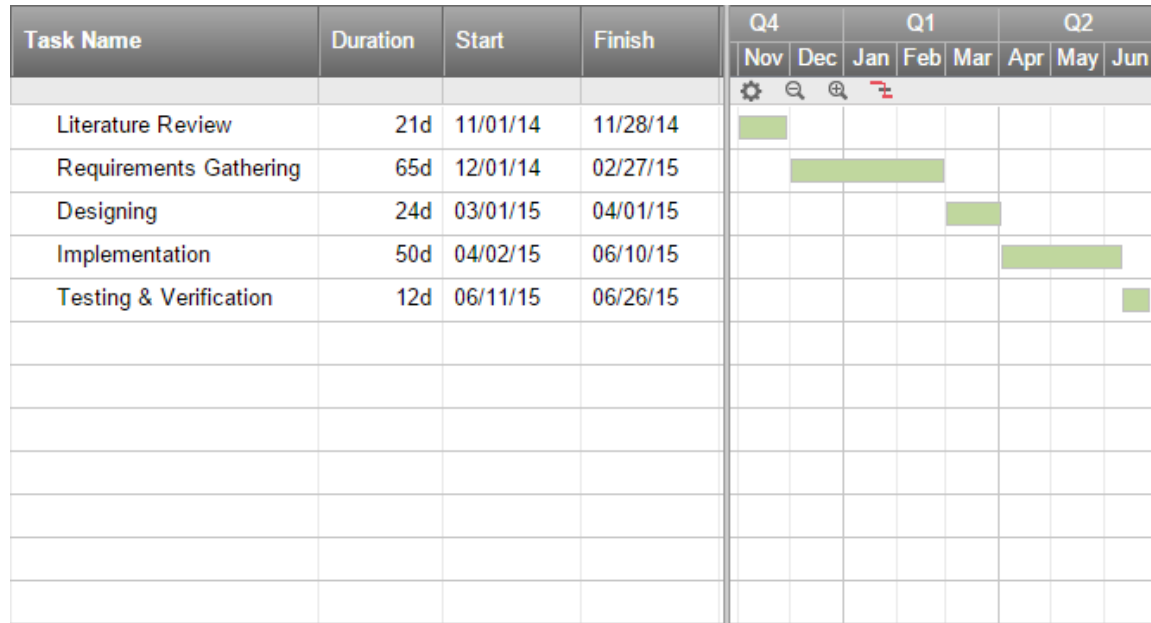
Table 2.4 below shows the time estimation on human resources specified by each role.

**Table 2.5: Resource Plan**

<b>Task Name</b>	<b>Estimate</b>	<b>Resources</b>
Layout Design	16 Days	Layout Designer
Application Mechanics	2 Days	Application Designer
Project Plan	7 Days	Project Manager
Development	84 Days	Developer
Testing	10 Days	Tester

**b) Gantt chart**

c) Figure 2.1 shows the set of activities to be performed and the time taken by each activity to complete.



**Figure 2.1: Gantt chart**

# Chapter 3

## System Requirements

This Chapter of the thesis is about analyzing requirement of the project. Requirements analysis is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed [14]. In this chapter we discussed different requirements that the application that needs to fulfill. This chapter will help understand the requirements of the application to be developed.

The system requirements chapter is related about analyzing requirement. The term requirement means a detailed, definition of the game functions. An SRS is basically defining the requirements. If the requirements are done well, the application design flows logically and smoothly. Conversely, if the requirements are done poorly, the resulting design is awkward and the coding is more difficult.

### 3.1 Functional Requirements

These are the statement of the services the system should provide. How the system should react to particular input and how the system should behave in particulate situation. The functional requirements for the system describe what the system should do. These requirements depend on the type of the application being developed. Expected users of the applications and the general approach taken by the organization when writing requirements.

Functional requirements for SNAPDIRECTION are. The user should be able to download the application from the Google play store into the system. And user should be able to start the launcher Icon into the smart phone. So that he/she shall be able to use the application.

## **3.2 Non-Functional Requirements**

Nonfunctional requirements as the name suggests, are those requirements which are not directly concerned with the specific function delivered by the system. They may relate to emergent system properties such as reliability, response time, and performance.

## **3.3 Use-Case**

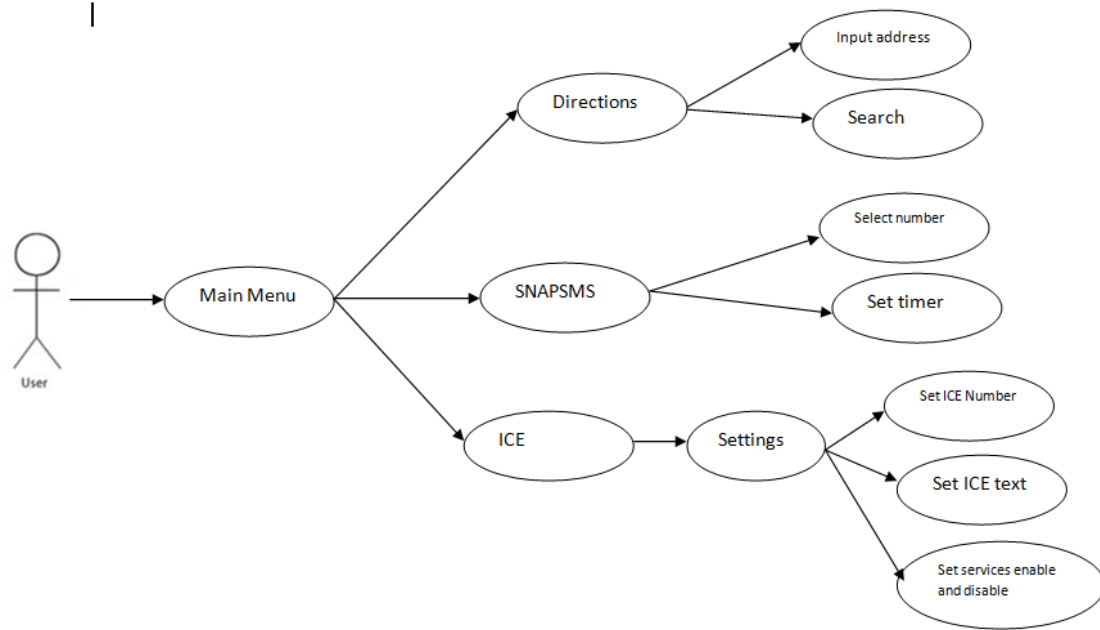
A Use-Case Diagram is used to depict the main aspects of the system and the ways a user can interact with the system [15]. Use case analysis is one of the first and primary means of gathering requirements in the behavioral methodology. Use cases are standard technique for gathering requirements in many modern software development methodologies in fact; In use case the end user requirements are defined Objects also define that they are internal to our system along with their scopes.

### **3.3.1 Use Cases Actors**

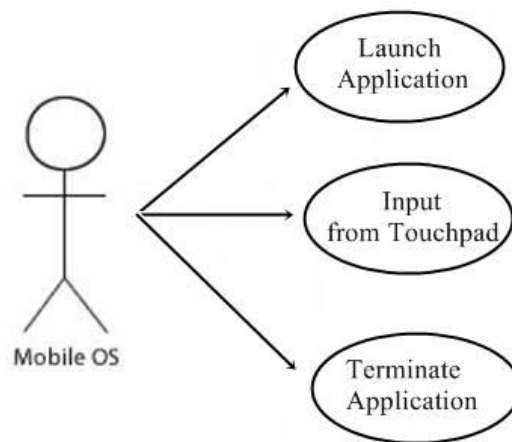
An actor is a person, organization, or external system that plays a role in one or more interactions with your system. Actors are drawn as stick figures.

### **3.3.2 Associations**

Associations between actors & use cases are indicated in use case diagram by solid lines. An association exists whenever an actor is involved with an interaction described by a use case. Associations are modeled as line connecting use cases and actors to one another, with an optional arrowhead on one end of the line.



**Figure 3.1: Use Case Diagram 1**



**Figure 3.2: Use Case Diagram 2**



# Chapter 4

## System Design

### 4.1 Logical design

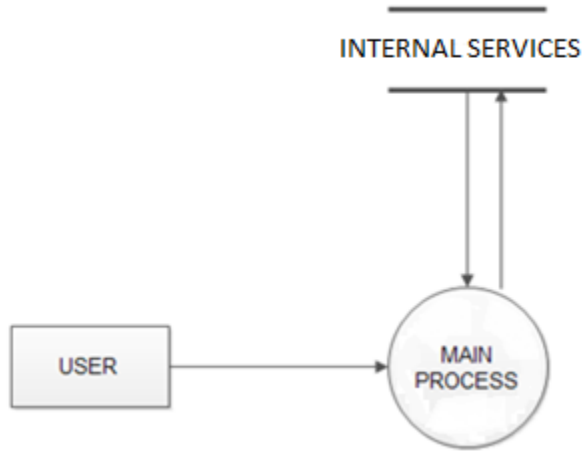
#### 4.1.1 DFD

Data Flow Diagram is used to represent how data are transformed; data modeling considers data independent of the processing that transforms the data. DFD is the way we represent our data flow through the system. We make data flow diagram to show that what are the processes, what the inputs to these processes are and what the outputs are. These diagrams show how users put data into processes, how processes utilize and manipulate the data and what information the processes give to the end user. Search start from some specific point in the system circle through tables to find those records and bring them to screens or output devices.

Unlike ERD, which only describes objects and relationships between objects, DFD describes how the system operates and how it accepts inputs to transform outputs. DFD shows the flow/movement of data through a system. DFD is also known as Transformation Graph, Process Model or Data Flow Graph etc.[16]

## 4.2 DFD Levels Used

### 4.2.1 Level 0 DFD



**Figure 4.1: Level 0 DFD represents the Initial System and User Interaction**

A view of entire Data Flow Diagram is explained and understood easily by level 0 DFD in Fig 4.1 where data starts flowing from user and ends up in main process

### 4.2.2 Information Flow of Level 0 DFD

- User Work on Main Process.
- Main process will further service three modules Directions, SNAPSMS, ICE System.
- Main process connected with an internal database.
- Data or Entries comes to internal services.
- Main Process Means Just Connect with Front End.

In this level we define,

- All external entities.
- What Work Done by User.
- All input/output data.

Level 0 data flow diagram is the overall sketch of our system. User only connected with the main process in which can access their information related to their applications present in mobile and also user can access their contacts from main process. This is simply presented because we are at level zero data flow diagram and as we go further deep into the system, we will increase the level of DFD.

### 4.2.3 Level 1 DFD

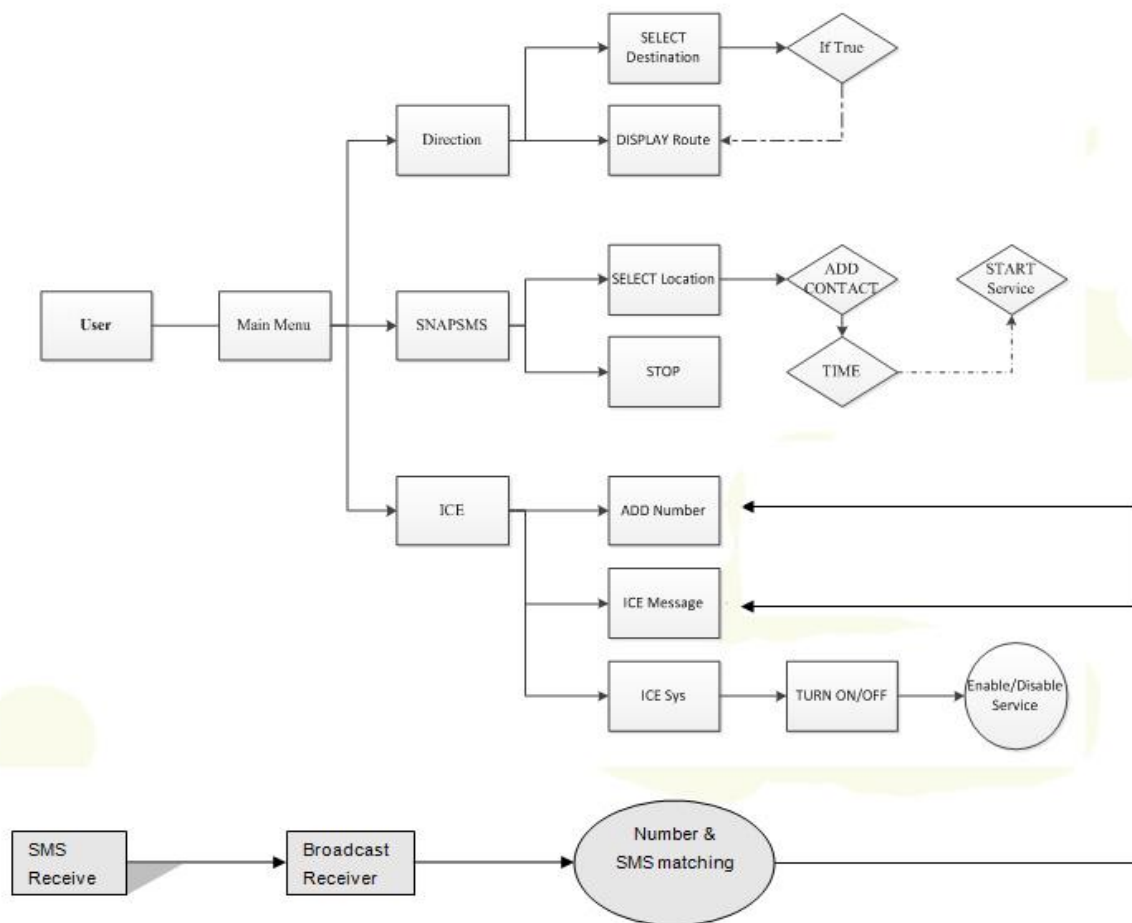


Figure 4.2: Level 1 DFD represents the Entire General Working and Dataflow

Fig 4.2 expands the level 0 DFD as shown in Fig 3 where the working of the entire scenario is clearly explained. The flow of data remains the same which starts from user and ends at different situations.

#### **4.2.4 Information Flow of Level 1 DFD**

##### **➤ Main Menu**

1. Interactive animated main menu button.
2. On pressing the main button 3 buttons are popped up.
3. Every button starts users desired module from application.

##### **➤ Direction**

1. User will select the location from front screen of application.
2. User can enter the address of their own choice.
3. The duration of selected path was displayed.

##### **➤ SnapSMS**

1. User will select the SNAPSMS from front screen of application.
2. User can add number or select number from contacts.
3. User can set the time limit for SMS.

##### **➤ ICE**

1. User will select the ICE from front screen of application.
2. User can write the number manually.
3. User can select number from contact list.
4. User can set ICE Message.
5. User can turn ON or OFF the ICE service.

### ➤ SMS Receive

1. On SMS received the broadcast receiver will invoke my ICE service.
2. ICE service will match the text and number of received SMS with user defines text and number.
3. If both match the ICE service will send an auto generated reply SMS to defined number with location, battery percentage and signal strength of user's smartphone.

## 4.3 Physical design

### 4.3.1 Interface Design

Android's default user interface is based on direct manipulation, using touch inputs, that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, and a virtual keyboard. The response to user input is designed to be immediate and provides a fluid touch interface, often using the vibration capabilities of the device to provide haptic feedback to the user. Internal hardware such as accelerometers, gyroscopes and proximity sensors are used by some applications to respond to additional user actions, for example adjusting the screen from portrait to landscape depending on how the device is oriented, or allowing the user to steer a vehicle in a racing game by rotating the device, simulating control of a steering wheel.[17]

Android devices boot to the home screen, the primary navigation and information point on the device, which is similar to the desktop found on PCs. Android home screens are typically made up of app icons and widgets; app icons launch the associated app, whereas widgets display live, auto-updating content such as the weather forecast, the user's email inbox, or a news ticker directly on the home screen. A home screen may be made up of several pages that the user can swipe back and forth between, though Android's home screen interface is heavily customizable, allowing the user to adjust the look and feel of the device to their tastes. Third-party apps available on Google Play and other app stores can extensively re-theme the home screen, and even mimic the look of other operating systems, such as Windows. Most

manufacturers, and some wireless carriers, customize the look and feel of their Android devices to differentiate themselves from their competitors. Android provides the ability to run applications which change the default launcher and hence the appearance and externally visible behavior of Android. These appearance changes include a multi-page dock or no dock, and many more changes to fundamental features of the user interface.

### 4.3.2 Application Startup



Figure 4.3: Application Startup Screen

### 4.3.3 Application Main Page:



Figure 4.4: Main Menu



#### 4.3.4 Direction View:

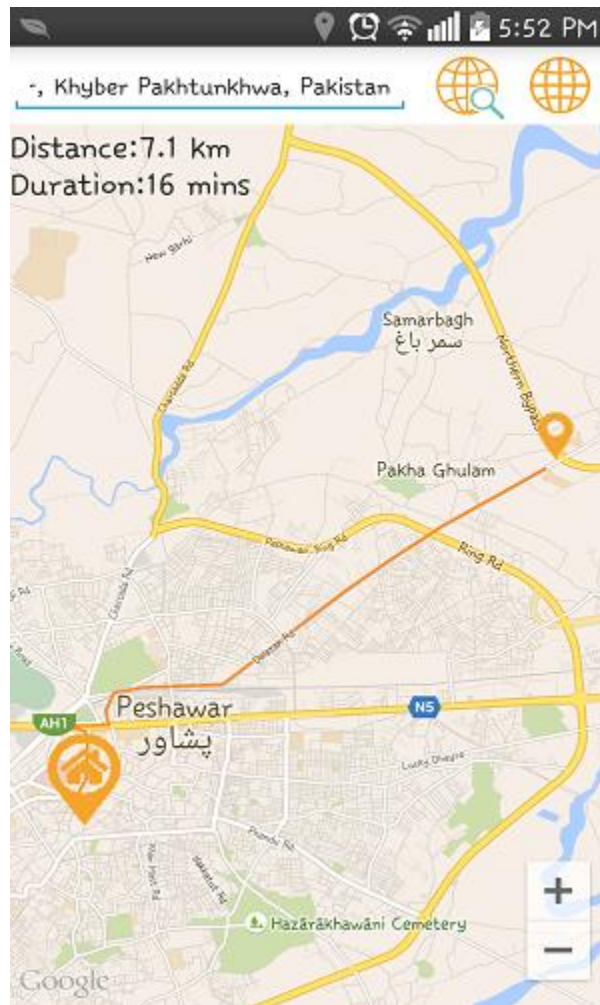


Figure 4.5: Route with Distance & Duration

#### 4.3.5 SNAPSMS:

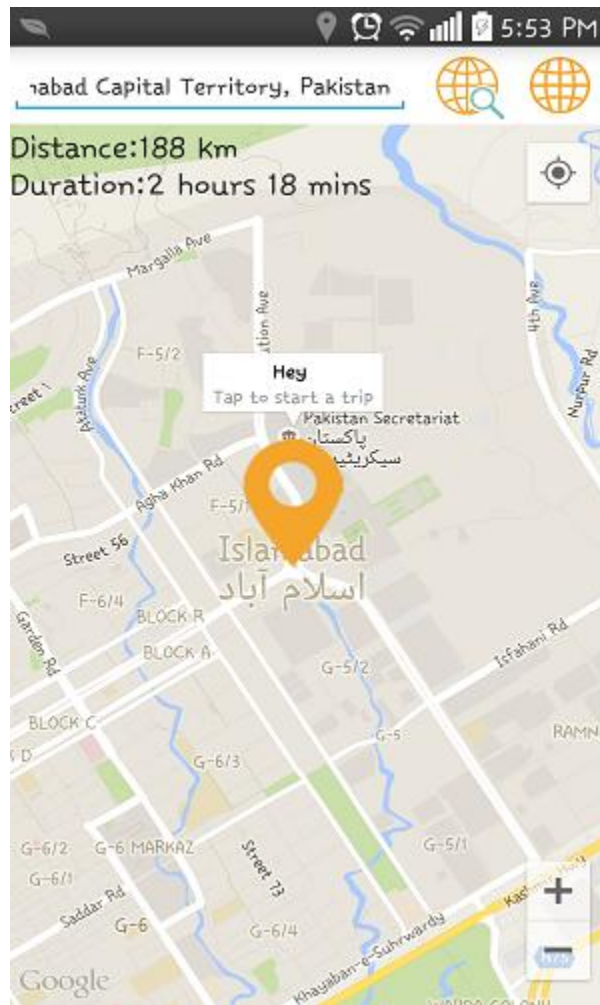


Figure 4.6: Selecting the Destination

#### 4.3.6 Set Timer:

The screenshot shows a mobile application interface with a status bar at the top displaying location, signal, and time (5:57 PM). The main content area is a light gray box with a rounded border. Inside, the text "Islamabad, Islamabad Capital Territory, Pakistan" is displayed, followed by "Latitude=33.7293882 Longitude=73.0931461". Below this is a text input field containing "03451923255". Underneath the input field is the "Set Timer" section, which has two columns: "Hours" and "Minutes". Each column has three input fields with values "24", "0", and "1" for Hours, and "60", "1", and "2" for Minutes. At the bottom of the interface are three icons: a gray square with "ch", a teal clock icon with a play button, and a red clock icon with a stop button.

Islamabad, Islamabad Capital Territory,  
Pakistan  
Latitude=33.7293882 Longitude=73.0931461

03451923255

Set Timer

Hours	Minutes
24	60
0	1
1	2

ch

Figure 4.7: Add Contact & Set Time

#### 4.3.7 Cancel SMS Timer:



Figure 4.8: Stop Timer

#### 4.3.8 ICE:

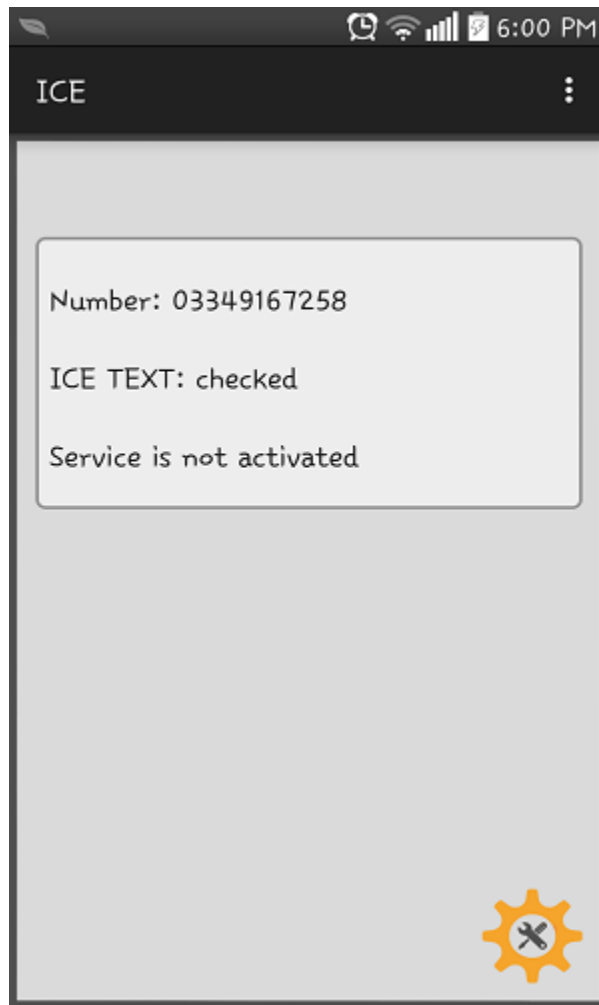
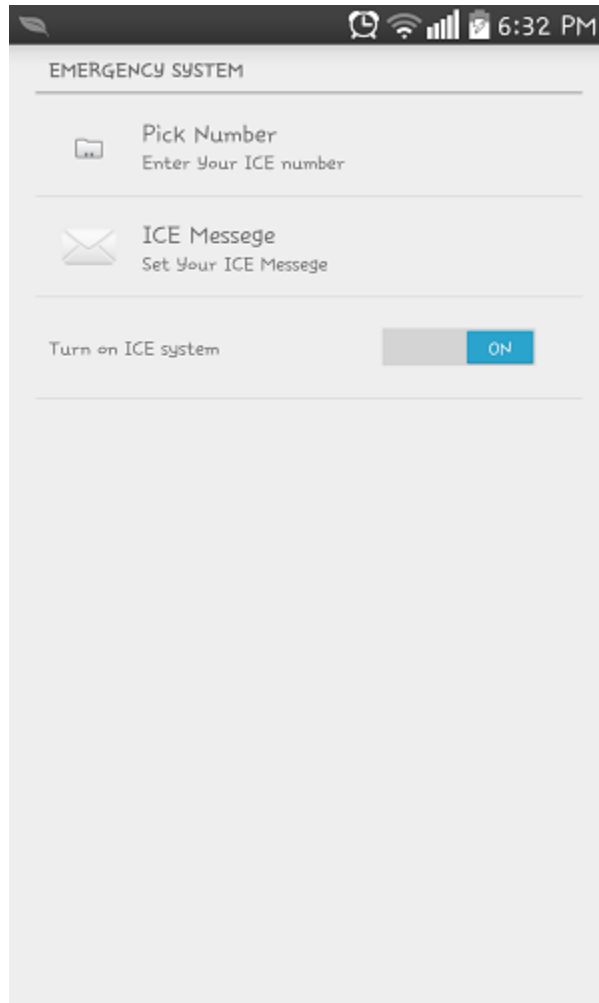


Figure 4.9: Select Number and Turn Service ON

### 4.3.9 Emergency System:



**Figure 4.10: Add Number & Turn ON ICE system**

#### 4.3.10 ICE Message:

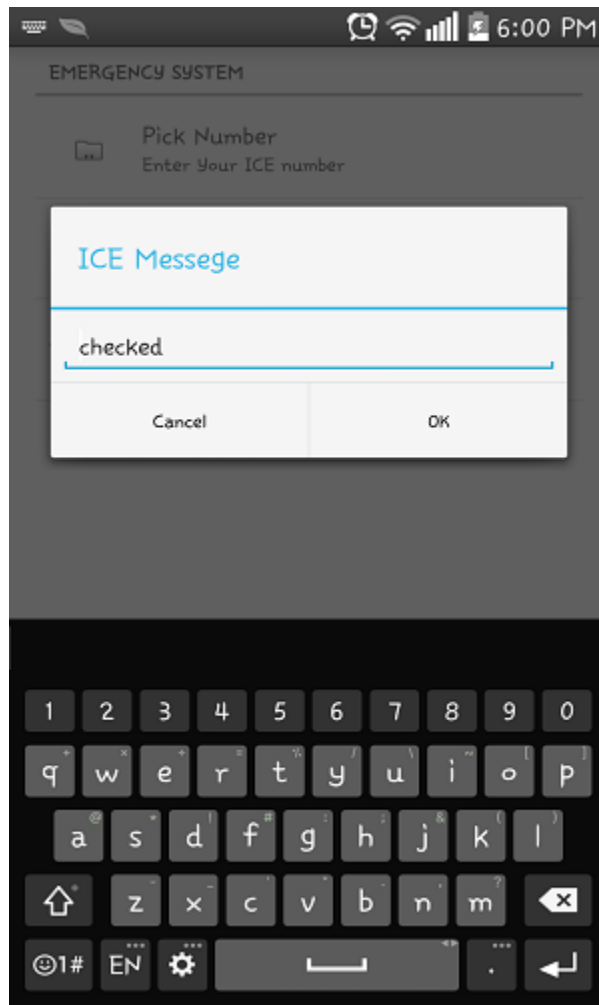


Figure 4.11: Create ICE Message

# Chapter 5

## System Testing

System testing is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements and show the presence of errors in a program. In the world of software testing, system testing is the testing of a complete and fully integrated software product. Usually software is only one element of a larger computer based system. Ultimately, software is interfaced with other software/hardware systems. System testing is actually a series of different tests whose sole purpose is to exercise the full computer based system.[18]

In testing of our system, we focus on using well-known methods to identify problems such as the GUI testing, Usability testing, scalability testing or etc.

### 5.1 Unit and Integration Testing

Unit testing is the process of testing individual units in the system in order to check that it is working as required without any debugging errors. We test the following components and that are all working correctly. Then we tested the SnapDirection application by adding each component to the previous one to be sure that combination of components works correctly until all components combine each other.

### 5.2 User Acceptance Testing

#### 5.2.1 Definition

User acceptance testing (UAT) is the last phase of the software testing process. During UAT, actual software users test the software to make sure it can handle required tasks in real-world scenarios, according to specifications. UAT is one of the final and critical software project



procedures that must occur before newly developed software is rolled out to the market. UAT is also known as beta testing, application testing or end user testing.[19]

In software development, user acceptance testing (UAT) - also called beta testing, application testing, and end user testing - is a phase of software development in which the software is tested in the "real world" by the intended audience. UAT can be done by in-house testing in which volunteers or paid test subjects use the software, by installing the app in android phone for different users. The experiences of the early users are forwarded back to the developers who make final changes before releasing the software commercially.

**Table 5.1: User Task**

<b>Task#</b>	<b>Task Description</b>
<b>T1</b>	<b>Checking The applications present in mobile</b>
<b>T2</b>	<b>Checking the contacts present in mobile</b>
<b>T3</b>	<b>Checking the GPS service available.</b>

There are 4 different task allocated as per usage of the SnapDirection android app.

1. T1 describes the user experience of checking the locations of maps and contacts available in smart phone.
2. T2 is the verification and satisfaction from users that either all the contacts they wanted are available, if available then how can they select a specified contact number.
3. T3 is the testing of response of app after connecting with internet.

**Table 5.2: Measuring System Effectiveness**

Task#	Number Of Errors					AVG
	User1	User2	User3	User4	User5	
T1	0	0	0	0	0	0
T2	0	1	1	3	0	1
T3	2	2	0	1	0	1

The numbers of errors that are made by the users while performing to the specific task show the effectiveness of the application. Results of effectiveness for each user are presented in Table II. The results show that Task1 and Task4 are performed by users without errors. The remaining tasks are completed by the users with one error averagely.

The time spent to perform the specified task by the user presents efficiency. The efficiency for each user is presented in the Table IV. Making errors by the user affects the time spent. So that T1 and T4 were completed in a short time but the remaining took time at least 5 times longer than the T1 and T4.

**Table 5.3: Measuring System Efficiency**

Task #	The time requires to perform the tasks by the user (second)					AVG
	User1	User2	User3	User4	User5	
<b>T1</b>	03:01	15:06	05:08	03:00	05:06	6.4s
<b>T2</b>	02:01	36:09	21:06	01:01	24:01	21s
<b>T3</b>	26:06	50:07	15:01	05:03	30:01	12s
<b>T4</b>	01:05	01:0	08:05	05:02	01:00	3.2s

## 5.3 Usability Testing

Usability means making products and systems easier to use, and matching them more closely to user needs and requirements. The international standard, ISO 9241-11, provides guidance on usability and defines it as:

The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.

Usability is about:

- 5.3.1 **Effectiveness** - can users complete tasks, achieve goals with the product, i.e. do what they want to do?
- 5.3.2 **Efficiency** - how much effort do users require to do this? (Often measured in time)
- 5.3.3 **Satisfaction** – what do users think about the products ease of use?

These Usability factors are affected by:

**The users** - who is using the product? e.g., are they highly trained and experienced users, or novices?

**Their goals** - what are the users trying to do with the product - does it support what they want to do with it?

Usability should not be confused with 'functionality', however, as this is purely concerned with the functions and features of the product and has no bearing on whether users are able to use them or not. Increased functionality does not mean improved usability.

# Chapter 6

## Conclusion

The final result of many months of designing and implementing is the Android application SNAPDIRECTION. SNAPDIRECTION is the first Android application which provides nice features. The application SNAPDIRECTION can be best used for the purpose of connecting peoples and communication. This application can be used by any user of any age anywhere around the world. The application will be available soon at Google play store.

Some basic features which the application will provide us:

- ☒ It will provide information about the user's location to intended number.
- ☒ It will provide information about the directions to user.
- ☒ User can track their loved one's location in case of emergency.

In this application SNAPDIRECTION anyone can easily get the information about their intended ones. Specific contact numbers in the list to which user's want to know their current location information. An auto generated SMS replied to those who wants the acknowledgement about their nearest and dearest ones, message is based on location address, battery level, and signal strength. The users just enter number and specific text and turn on the ICE, and then the message replied to those who send a defined SMS to the user.

## References

- [1] Matt Egan (Editor), Rosie Hattersley (Editor).The Complete Guide to Google Android. United Kingdom, IDG Communications – UK, 2011,pp.23-25.
- [2] What is Android? <http://developer.android.com/guide/basics/what-is-android.html>, [August 4th, 2015].
- [3] Android Studio Plugin. <http://android.googlesource.com>. Google. [august 5th, 2015].
- [4] Download Android Studio.<https://developer.android.com/sdk/index.html>. [August 6th, 2015].
- [5] VentureBeat. Google Launches Android Studio And New Features For Developer Console, Including Beta Releases And Staged Rollout. <http://venturebeat.com/2014/12/08/google-releases-android-studio-1-0-the-first-stable-version-of-its-ide/>. December 8, 2014. [August 6th, 2015].
- [6] Ducrohet, Xavier; Norbye, Tor; Chou, Katherine (May 15, 2013). "Android Studio: An IDE built for Android". *Android Developers Blog*. Google. [August 6th, 2015].
- [7] "Getting Started with Android Studio". *Android Developers*. Google. [August 6th, 2015].
- [8] Android Developers. <http://developer.android.com/tools>. [August 6th, 2015].
- [9] Developers. [https://en.wikipedia.org/wiki/Adobe\\_Photoshop](https://en.wikipedia.org/wiki/Adobe_Photoshop). [August 8th, 2015].
- [10] Microsoft. <http://office.microsoft.com/en-001/products>. [August 8th, 2015].
- [11] Google Developers. <https://developers.google.com/maps/documentation/android/intro>. [August 8th, 2015].

- [12] Google Developers. <https://developers.google.com/places/android/start>. [August 8th, 2015].
- [13] RationalPLan. <http://www.rationalplan.com/project-planning-software.php>. [August 8th, 2015].
- [14] Margaret Rouse. techtarget: <http://searchsoftwarequality.techtarget.com/definition/requirements-analysis>. [August 8th, 2015].
- [15] Wikipedia:  
[http://en.wikibooks.org/wiki/Introduction\\_to\\_Software\\_Engineering/Planning/Requirements#Software\\_requirements\\_specification](http://en.wikibooks.org/wiki/Introduction_to_Software_Engineering/Planning/Requirements#Software_requirements_specification). [August 10th, 2015].
- [16] SYBASE:  
<http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.infocenter.dc38088.1610/doc/html/rad1232026266129.html>. **January 30, 2012**. [August 10th, 2015].
- [17] Tutdsdady. What is Android ? What are its Features and Applications?. <http://tutsdaddy.com/what-is-android.html>. [August 10th, 2015].
- [18] Abhijit A. Sawant, Pranit H. Bari and P. M. Chawan.” Software Testing Techniques and Strategies”. International Journal of Engineering Research and Applications, vol.2, pp. .980-986, May-Jun 2012.
- [19] Matthew Setter. “USER ACCEPTANCE TESTING – HOW TO DO IT RIGHT!”. Usersnap: <http://usersnap.com/blog/user-acceptance-testing-right/>. [August 11th, 2015].



Copyright by  
Saad Rehman (4971) and Ali Raza Mir(4855)  
August, 2015