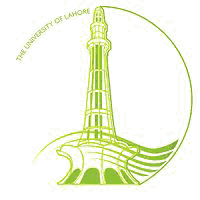
**FINAL YEAR PROJECT REPORT**

**On**

**Location Based Task Management Application**



**Submitted By:**

**Reg#: BCS07123001 Kamran Ali**

**Reg#: BCS07123032 Mudassar Nawaz**

**In Partial Fulfillment of**

**The Requirements For the Degree of**

**Bachelor of Science in Computer Sciences**

**Supervised By: Muhammad Imran Hafeez**

**DEPARTMENT OF COMPUTER SCIENCES**

**UNIVERSITY OF LAHORE, SARGODHA CAMPUS**

**Session: Fall 2012 – Winter 2016**

**Final Approval**

This Project titled

“Location Based Task Management Application”

By

Kamran Ali

BCS07123001

Mudassar Nawaz

BCS07123032

Has been approved for the

Department of Computer Sciences,

The University of Lahore – Sargodha Campus

**External Examiner**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Muzamil Asghar

**Supervisor**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: Muhammad Imran Hafeez

Lecturer,

Department of Computer Sciences,

University of the Lahore,

Sargodha Campus

**FYP Committee Head: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Name: Miss Tanzila Kehkashan

Lecturer,

Department of Computer Sciences,

University of the Lahore,

Sargodha Campus

**Head of Department: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Name:Professor. Dr. Pervaiz Rabbani Khan

Department of Computer Sciences,

University Of Lahore,

Sargodha campus

**Declaration**

We hereby declare that this project, neither as a whole nor as apart thereof has been copied out from any source. It is further declare that we developed this project and this report entirely on the basis of our personal efforts made under the sincere guidance of our project supervisor.No portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other University or institute of learning. We further declare that this software and all associated documents, report and records are submitted as partial requirements for the degree of master in information technology.

Kamran Ali

Reg. #: BCS07123001

Mudassar Nawaz

Reg. # BCS07123032

**Acknowledgement**

Our thanks and appreciation to our teacher Muhammad Imran Hafeez for persevering with us as our advisor throughout the time it took us to complete this project. Whose steadfast support of this project was greatly needed and deeply appreciated. The program was one of the most important and formative experiences in our lives. We are grateful as well to all teachers of CS department for their throughout coordination. We must acknowledge as well our families and many friends for always being there for us.

Table of Contents

[Chapter 1 Introduction and Background](#_Toc466862509) 1

[1.1 Statement of Problem Area 1](#_Toc466862510)

[1.2 Background History 1](#_Toc466862511)

[1.3 Previous and Current Work 2](#_Toc466862512)

[1.4 Project Description 2](#_Toc466862513)

[1.5 Purpose 2](#_Toc466862514)

[1.6 Objectives 3](#_Toc466862515)

[1.7 Scope 3](#_Toc466862516)

[1.8 Introduction 3](#_Toc466862517)

[1.9 Tools and Technologies used 4](#_Toc466862518)

[1.11 Supervisor Name 4](#_Toc466862519)

[1.12 GANTT Chart](#_Toc466862520) 5

[Chapter 2 Software Requirements Specification 6](#_Toc466862522)

[2.1 Functional Requirements 6](#_Toc466862523)

[2.1.1 Location 6](#_Toc466862524)

[2.1.2 Reminder 6](#_Toc466862525)

[2.1.3 Message 8](#_Toc466862526)

[2.1.4 Mobile Profile 9](#_Toc466862527)

[2.2 Non-Functional Requirements 9](#_Toc466862528)

[2.3 Project/Product Feasibility Report 9](#_Toc466862529)

[2.3.1 Technical Feasibility 10](#_Toc466862530)

[2.3.2 Operational Feasibility 10](#_Toc466862531)

[2.3.3 Legal and Ethical Feasibility 11](#_Toc466862532)

[2.3.4 Economic Feasibility 11](#_Toc466862533)

[2.3.5 Schedule Feasibility 11](#_Toc466862534)

[2.3.6 Motivation Feasibility 11](#_Toc466862535)

[2.3.7 Information Feasibility 12](#_Toc466862536)

[Chapter 3 System Performance & Requirement Specification 13](#_Toc466862537)

[3.1 Efficiency 12](#_Toc466862538)

[3.2 Reliability 12](#_Toc466862539)

[3.3 Security 12](#_Toc466862540)

[3.4 Maintainability 12](#_Toc466862541)

[3.5 Modifiability 13](#_Toc466862542)

[3.6 Portability 13](#_Toc466862543)

[Chapter 4 System Analysis & Design Overview 14](#_Toc466862544)

[4.1 Use Case Diagrams 14](#_Toc466862545)

[4.2 Software Process Model 19](#_Toc466862546)

[4.3 Data Models 22](#_Toc466862547)

[4.4 Behavioral Models 23](#_Toc466862548)

[Data Flow Models 23](#_Toc466862549)

[4.5 System Sequence diagrams 28](#_Toc466862551)

[4.6 State Models 31](#_Toc466862552)

[Set Reminder 32](#_Toc466862553)

[4.7 Object Models 34](#_Toc466862554)

[4.8 Data dictionary 35](#_Toc466862555)

[4.9 Required Support Software (pre­existing) 36](#_Toc466862556)

[Chapter 5 User Interface Design 37](#_Toc466862557)

[5.1 User Interface Specification 32](#_Toc466862558)

[5.1.1 User Interface Design (Screens) 32](#_Toc466862559)

[Dashboard 32](#_Toc466862560)

[Reminder Screen 33](#_Toc466862561)

[View Reminder 34](#_Toc466862562)

[Delete Reminder 35](#_Toc466862563)

[Message Screen 36](#_Toc466862564)

[Add Contacts 37](#_Toc466862565)

[Delete Message Task 38](#_Toc466862566)

[Map Screen 40](#_Toc466862567)

[Chapter 6 System Verification & Validation 42](#_Toc466862568)

[6.1 System Verification & Validation 42](#_Toc466862569)

[6.2 Testing Levels 43](#_Toc466862570)

[Unit Testing 43](#_Toc466862571)

[6.2 Schedule 45](#_Toc466862572)

[6.3 Control Activities 45](#_Toc466862573)

[6.4 Acceptance Testing 45](#_Toc466862574)

[6.5 Validation Testing 46](#_Toc466862575)

[6.5.1 Location: 46](#_Toc466862576)

[6.5.2 Contacts: 46](#_Toc466862577)

[6.5.3 Message: 47](#_Toc466862578)

[6.5.4 Reminder: 47](#_Toc466862579)

[6.5.5 Mobile Profile: 48](#_Toc466862580)

[6.4 Justification of test cases 48](#_Toc466862581)

[6.5 Test Run Procedures & Result 48](#_Toc466862582)

[Chapter 7 Conclusions 50](#_Toc466862583)

[7.1 Summary 49](#_Toc466862584)

[7.2 Problems Encountered and Solved 49](#_Toc466862585)

**[Bibliography](#_Toc466862586)** [51](#_Toc466862586)

**List of Figures**

[Figure 1.1 GANTT chart 5](file:///C:\Users\Awan\Desktop\mudassar%20fyp\New%20folder\finaldocumentation1%20-%20Copy.docx#_Toc466862587)

[Figure 4.1 Use Case of Set Reminder 15](#_Toc466862588)

[Figure 4.2 Use Case of Define Message 17](#_Toc466862589)

[Figure 4.3 Use Case of Set Mobile Profile 19](#_Toc466862590)

[Figure 4.4 Waterfall Model 21](#_Toc466862591)

[Figure 4.5 ER Diagram 22](#_Toc466862592)

[Figure 4.6 Context Level Data Flow Diagram 23](#_Toc466862593)

[Figure 4.7 Level 0 Data Flow Diagram 24](#_Toc466862594)

[Figure 4.8 Level 1 25](#_Toc466862595)

[Figure 4.9 Sequence Diagram of Define Message 28](#_Toc466862596)

[Figure 4.10 Sequence Diagram of Set Mobile Profile 29](#_Toc466862597)

[Figure 4.11 Sequence Diagram of Set Reminder 30](#_Toc466862598)

[Figure 4.12 Activity Diagram of Define Message 31](#_Toc466862599)

[Figure 4.13 Activity Diagram of Set Reminder 32](#_Toc466862600)

[Figure 4.14 Activity Diagram of Set Mobile Profile 33](#_Toc466862601)

[Figure 4.15 Class Inheritance Model 34](#_Toc466862602)

[Figure 4.16 Data Dictionary Diagram](#_Toc466862603) 40

[Figure 5.1 Dashboard 32](#_Toc466862604)

[Figure 5.2 Reminder Screen 33](#_Toc466862605)

[Figure 5.3 View Reminder 34](#_Toc466862606)

[Figure 5.4 Delete Reminder 35](#_Toc466862607)

[Figure 5.5 Message Screen 36](#_Toc466862608)

[Figure 5.6 Add Contacts 37](#_Toc466862609)

[Figure 5.7 Delete Message Task 38](#_Toc466862610)

[Figure 5.8 Profile Screen 39](#_Toc466862611)

[Figure 5.9 Map Screen 40](#_Toc466862612)

[Figure 5.10 Map Screen 41](#_Toc466862613)

**List of Tables**

[Table 1.1 Deployment Platforms 4](#_Toc466862623)

[Table 1.2 Supervisor Name 4](#_Toc466862624)

[Table 2.1 Location 6](#_Toc466862625)

[Table 2.2 Reminder 7](#_Toc466862626)

[Table 2.3 Message 8](#_Toc466862627)

[Table 2.4 Mobile Profile 9](#_Toc466862628)

[Table 4.1 Set Reminder 14](#_Toc466862629)

[Table 4.2 Define Message 16](#_Toc466862630)

[Table 4.3 Mobile Profile 18](#_Toc466862631)

# 

# **Introduction and Background**

## Statement of Problem Area

We are living in digital telecommunication era where human undergoes with busy schedules. User of today more focused to applications when it comes to his cell phones. People find facilitating applications that help users in some tasks. User sometimes not is able to perform activities due to schedule or nature of environment. Application will target users with busy schedules and will perform some of mobile related activities.

Android cell phone user’s deals with activities like messages, mobile profile, route finder, near location and reminders on daily basis. Messages today are linked up when mobile user want to send them. There are many situations where user may not be able to send messages due to environment. Reminders in android cell phone associated with time. User sometimes forgot to set mobile profile according to situations. Tasks listed upward firstly not on single place. Tasks have many constraints and limitations like time. There is need of single application that will integrate all the activities and perform these activities with respect to saved desired location.

## Background History

With location positioning system such as GPS becoming popular, there is a growing demand for location-based applications. It is easier, these days to utilize map information by connecting GPS receiver to PC and PDA. Corresponding to this momentum, GPS receivers are now embedded into mobile phones and applications using the location of the user in real-time are widely available. GPS chips are now included in many devices to analyses satellite signals and determine the user’s location with high accuracy. In a large social event, such as a big conference, since people come to communicate with each other, they are more likely to release their location information and the location privacy is not an essential concern. In addition, people usually need to register to join a conference so a location server can easily get the participants’ profiles. Hence, the server-centric mode is an economic way to handle location detection for big conferences. One of the key technological advances for the development of location-based applications is the use and availability of positioning systems.

## Previous and Current Work

Location Based Service (LBS) LBS is mobile service that has the capability to provide real time information based on the user’s location. Geographical Information System (GIS) has been the heart of LBS in order to provide all the functionalities in LBS. First, we may send location information to remote parties. This set of services are commonly used today, e.g., in location tracking applications. Second, use location information to make communication decisions, e.g., a user agent may automatically disable instant messaging when driving. Third, location changes can trigger communication actions, e.g., when a person’s user agent gets a location notification indicating the person enters a room, the user agent may automatically turn on the light of the room. Sending location information to remote parties for location tracking Locations are usually represented in geospatial coordinates or civil addresses for tracking. By enabling to upload real time location and to create the content “on the spot”, we can expect more variety of location-based services.

## Project Description

Location Based Task Management Application is an android based application where cell phone activities like reminders, mobile profile, messages and route finder, near location have been linked up with GPS. Location is key trigger in this application that initiates one of the custom defined tasks and performs these custom defined tasks.

## Purpose

Technology has improved a lot over the last few decades. One of the best and biggest technological advancements is the invention of smart phone. A smart phone is a device which offers more advanced computing and connectivity than regular mobile phones. In the last one or two year the smart phone users have rapidly increased and the count is still on. People using smartphones demand for better applications and updates for existing ones, which in turn creates a huge scope of Android mobile application development .Android is a fully open source platform which was created completely for smart phones and similar devices like tablets.

This project is developed for Android based smart phones. The app should keep an eye on the GPS information collected by user’s smart phone to trigger the alert when the user approaches a destination. User can find route easily and near places like schools, hospitals and restaurants

## 1.6 Objectives

Location Based Task Management Application is struggle from us in order to automate the daily routine mobile tasks. Main objectives of application listed.

* Automate cell phones daily tasks.
* Facilitate user community
* Integrate basic activities on single platform
* Automation of activities with respect to location.
* Route Finder Just pin start point and end point and get route
* To find nearby places

## 1.7 Scope

[Product is android application that is allowing user to perform basic cell phone task according to trigger (Location). It has solved the above listed problems faced by user community. If user has defined its own desired location, then messages will be sent to target person automatically when user will enter that location. Application will resolve the reminder issue. Cell phone reminder no longer associated with time but it will also be manageable easily with location. Mobile profile issue has been resolved as profile automatically set to silent when user will enter saved desired location. User can find route easily by drop the pin on map and user can see near places like schools, hospitals and restaurants. If user has forgotten to set the profile to silent then there is no issue because application sole purpose is to address user issue.

## 1.8 Introduction

Mankind of today era is living in world of information technology. Most of persons on this planet have android cell phones and they are downloading and using millions of applications. User of today era demands application that must been able to perform mobile related activities. There are many applications facilitating user by completing their required works. User of today generally deals with messages, reminders, route finder, nearby places and mobile profile, when it comes to usage of android cell phones. User activities listed not been integrated on a single platform. Peoples sometime not been able to perform these tasks due to workplace, meetings, locations. Reminder usually is time based. Sometimes user may want to set the reminder for specific place or location instead of time. Messages and mobile profiles not been linked up with respect to location on a single platform.

## 1.9 Tools and Technologies used

Project completed with the help of following tools.

* Android Studio

Project completed with help of following technologies

* Java
* XML

1.10 Deployment Platform

Table 1.1 Deployment Platform

|  |  |
| --- | --- |
| **Window name** | **Architecture** |
| Windows XP | 32bit,64 bit |
| Windows 7 | 32bit,64 bit |
| Windows 8 | 32bit,64 bit |
| Windows 8.1 | 32bit,64 bit |
| Windows 10 | 32bit,64 bit |

## 1.11 Supervisor Name

|  |  |
| --- | --- |
| **Name** | **Email** |
| Muhammad Imran Hafeez | Imranhafiz79@gmail.com |

Table 1.2 **Supervisor Name**

## 1.12 GANTT Chart

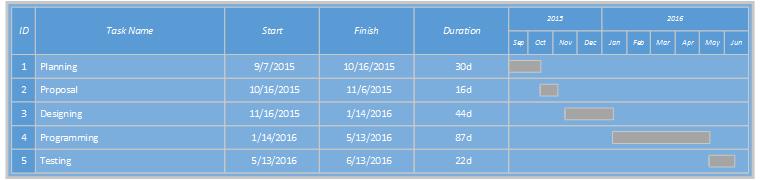


Figure 1.1 Gantt chart

# 

# 

# Software Requirements Specification

## 2.1 Functional Requirements

Software product is developed using software development life cycle. Analysis and design of system will start from completion of multi-step investigation of requirements. This step is base for running of project life cycle. There are many different models and methodologies but each generally consist of series of defined steps and stages. This section of reports will cover essential functional requirements of projects. System response time, startup time and processing time along with quality attributes listed in detail in this section. This section will also tell the running platform (Software Requirements) and minimum hardware needed for project.

### 2.1.1 Location

Table 2.1 Location

|  |  |
| --- | --- |
| **Requirement ID:** Req-001 | **Requirement Name:** Define Location |
| **Primary Actor:** User | **Criticality:** High Top |
| **Description:** User of application will open location task management application. User will select any of three main requirements. User then will ask for defining location. Searching of location through google map and by zooming in the map for your desired location. User can do following task related to location.   * **Important:** Define Location * Save Location (longitude, Latitude, Id) * View Locations * Delete Locations | |

### 

### 2.1.2 Reminder

Table 2.2 Reminder

|  |  |
| --- | --- |
| **Requirement ID:** Req-002 | **Requirement Name:** Ring Reminder |
| **Primary Actor:** User | **Criticality:** High Top |
| * **Description:** User of application will open location task management application. User will select reminder section. User then will ask for defining location. User will define location with explained in REQ-001. User will define notifications and save location. When target application person will enter set location that has been compared with current location application will ring reminder with mentioned notification. Following functions linked up with this requirement **Important:** Ring Reminder * Set Reminder * Write Notifications * Save Notification * View Notification | |

### 

### 2.1.3 Message

Table 2.3 Message

|  |  |
| --- | --- |
| **Requirement ID:** Req-003 | **Requirement Name:** Send Message |
| **Primary Actor:** User | **Criticality:** High Top |
| **Description:** User of application will open location task management application. User will select message section. User then will ask for defining location. User will define location with explained in REQ-001. Application will search contacts from phonebook and user will set contact along with defined location, message. when target application person will enter set location that has been compared with current location application will send message with to listed contact with listed message. Following functions linked up with this requirement   * **Important:** Send message * Write Message * View Message * Save Message * Search contacts * Save contacts | |

### 

### 2.1.4 Mobile Profile

Table 2.4 Mobile Profile

|  |  |
| --- | --- |
| **Requirement ID:** Req-004 | **Requirement Name:** Set Mobile Profile |
| **Primary Actor:** User | **Criticality:** High Top |
| **Description:** User of application will open location task management application. User will select profile section. User then will ask for defining location. User will define location which explained in REQ-001. User will set profile to “Silent” along with defined location. When target application person will enter set location that has been compared with current location application will set profile to silent. Following functions linked up with this requirement.   * **Important:** Set Mobile Profile. | |

## 

## 2.2 Non-Functional Requirements

**Startup Time:** The system will be operational within 10 seconds.

**Response Time:** Average response time will be less than 5 seconds for any of listed functional requirement

**Capacity:** Application can accommodate unlimited number of locations and has no limit for setting up the alarm and message.

## 2.3 Project/Product Feasibility Report

Project will be carried out with proper feasibility analysis. Analysis focus on these major questions

* What are user demonstrable needs
* How does candidate system meets needs
* What resources available for given system
* What are Impacts of project
* Whether is it worth to solve the problem

### 2.3.1 Technical Feasibility

**Front End Selection**

Feasibility study conducted before starting of project and following points concluded for front end technical feasibility

* 10 Graphical user interfaces will be designed to assist user in fulfilling of functional requirement
* Application must be easy to debug and maintain
* Front end must be flexible and have space for extension

**Back End Selection**

Back end technical feasibility of application has following points

* Efficient data Handling
* Multiple user support
* Easy to install
* Easy to integrate with front end
* Efficient Data retrieval and maintenance
* SQLite as back end databases

### 2.3.2 Operational Feasibility

Project is operationally feasible as it is easy to use the application due its reliability. Interfaces and user help section facilitate end user for understanding the application. Application is operational 24 hours after its installation. There are no special skills required to use the application.

### 2.3.3 Legal and Ethical Feasibility

In legal feasibility we tell that our project is not in conflict with society rules. This project is not .Illegal or do not harm any person in society. Our system is user friendly and authentication facility also provided so that personal data of organization or person will be save. This system will not leak out any information.

### 2.3.4 Economic Feasibility

It includes cost benefit analysis. We weight the cost and the benefit associated with this application. Economic feasibility analysis for our application conducted and economic feasibility of application as follows

* There is no amount required for development of this application
* The application can easily return ROI in limited span of time
* Per click download and installs will return cost if any due to large demand of android based task management application
* So application is economically feasible for development.

### 2.3.5 Schedule Feasibility

Project schedule feasibility was done in start of project. Project is for partial fulfillment of degree requirement so series of systematic steps carried out in completion of this project. Project developed in short time span of 7 months.

### 2.3.6 Motivation Feasibility

This project is developed through team work and with the help of supervisor. In team we discuss every aspect of our project and everything about the project. We work together so that there is no overhead of work on a single person.

### 2.3.7 Information Feasibility

The system provides end user with timely accurate and formatted information. Provide them correct information about place and about their work.it will provide meaning full and accurate data all the time. The information handling in current mode is done manually .So there is result in scrawling of data in this system is computerized. The human errors will be minimal. The data can be easily modified updated when required.

.

# 

# System Performance

# & Requirement Specification

## 3.1 Efficiency

Project is android application which will serve as platform to integrate basic android cell phone activities in a single location with respect to location. It is being built as new self-contained product. Market has task management application that runs with GSM and locations but all the type of applications are separate. This application is one of such type of application that will integrate activities in a single application that will run on location. So that’s why it is not part of larger system or it is not up gradation of existing application.

## 3.2 Reliability

There is no down time for application. It will be reliable at any time. Moreover, mean time for recovery in case of downtime is 2 hours.

## 3.3 Security

Android is a Linux kernel mobile platform. Android runs on a wide range of devices, from mobile smartphones and tablets, to set-top boxes. The Android mobile operating system is dependent upon the mobile device’s processer capabilities for its performance.

Android's Five Key Security Features:

* Security at the operating system level through the Linux kernel
* Mandatory application sandbox
* Secure inter process communication
* Application signing
* Application-defined and user-granted permissions

## 3.4 Maintainability

After the deployment of the project if any error occur it can easily maintain by the developer and user can also handle some problems. The system provides ease of maintenance because data is stored at single place.so there is no need of storing different books at different places. To perform any operation and to understand the functioning of the software is very easy. If user want to make some changing in software it is very easy.

## 3.5 Modifiability

Modification term is used for making changing in the functionality of the system. The system is flexible. If organization want to modify the functionality of the system it is possible. Software is adaptive to changing. Changing in the software is very easy

## 3.6 Portability

This application can be run in android devices and Can easily install and run into android support devices.

# 

# System Analysis

# & Design Overview

## 4.1 Use Case Diagrams

* **Use Case Set Reminder**

Table 4.1 Set Reminder

|  |  |
| --- | --- |
| **Use Case UC1: Reminders** | |
| **Scope:** | Android application |
| **Primary Actor:** | Mobile user |
| **Stockholder & interest:** | User wants to set reminder for any specified location. |
| **Description:** | User wants to save reminder. |
| **Precondition:** | User must have app installed on his android device. And must be selected the reminder option from the dashboard. |
| **Success Guarantee (post condition):** | Show a reminder message to user, when user reaches on the saved location. |
| **Main Success scenario:** | 1. User selects the reminder option from the dashboard screen of the app. 2. Search the location and save the desired location, and then enter the reminder name. After that click on save button to save the reminder.   3. If the user saves another reminder on the saved location, then the previous reminder will be over write. |
| **Extension:** | 1. System gives an error message. 2. At any time system fails to connect database. 3. System may be crash. |
| **Special Requirement:** | This process work well if the Google play service is updated. |



Figure 4.1 Use Case of Set Remind

* **Use Case Define Message**

Table 4.2 Define Message

|  |  |
| --- | --- |
| **Use Case UC2: Messages** | |
| **Scope:** | Android application |
| **Primary Actor:** | Mobile user |
| **Stockholder & interest:** | User wants to send messages on any specified location. |
| **Description:** | User wants to send messages automatically to other users, on previously saved location. |
| **Precondition:** | User must have app installed on his android device. And must be selected the message option from the dashboard. |
| **Success Guarantee (post condition):** | Send message to other users, when he reaches on the saved location. |
| **Main Success scenario:** | 1. User selects the message option from the dashboard screen of the app. 2. Search the location and save the desired location, and then enter the message and selects the contacts. After that click on save button to save the message.   3. When the user reaches on that location, message will automatically send to other users. |
| **Extension:** | 1. System gives an error message. 2. At any time system fails to connect database. 3. System may be crash. |
| **Special Requirement:** | This process work well if the Google play service is updated. |



Figure 4.2 Use Case of Define Message

* **Set Mobile Profile**

Table 4.3 Mobile Profile

|  |  |
| --- | --- |
| **Use Case UC3: Set Mobile Profile** | |
| **Scope:** | Android application |
| **Primary Actor:** | Mobile user |
| **Stockholder & interest:** | User wants to silent his phone on any specified location. |
| **Description:** | User wants to silent his phone automatically on previously saved location. |
| **Precondition:** | User must have app installed on his android device. And must be selected the profile option from the dashboard. |
| **Success Guarantee (post condition):** | Mobile phone will automatically go to silent mode, when he reaches on the saved location. |
| **Main Success scenario:** | 1. User selects the profile option from the dashboard screen of the app. 2. Search the location and save the desired location. After that click on save button to save the profile.   3. When the user reaches on that location, his phone will automatically go on silent mode. |
| **Extension:** | 1. System gives an error message. 2. At any time system fails to connect database. 3. System may be crash. |
| **Special Requirement:** | This process work well if the Google play service is updated. |



Figure 4.3 Use Case of Set Mobile Profile

## 4.2 Software Process Model

The waterfall model is a popular version of the systems development life cycle model for software engineering. Often considered the classic approach to the systems development life cycle, the waterfall model describes a development method that is linear and sequential. Waterfall development has distinct goals for each phase of development. Imagine a waterfall on the cliff of a steep mountain. Once the water has flowed over the edge of the cliff and has begun its journey down the side of the mountain, it cannot turn back. It is the same with waterfall development. Once a phase of development is completed, the development proceeds to the next phase and there is no turning back.

**Requirement Analysis & Definition:**

This phase is focused on possible requirements of the system for the development are captured. Requirements are gathered subsequent to the end user consultation.

**System & Software Design:**

Prior to beginning the actual coding, it is inevitable to understand what actions are to be taken and what they should like. The requirement specifications are studied in detail in this phase and the design of the system is prepared. The design specifications are the base for the implementation and unit testing model phase.

**Implementation & Unit Testing:**

Subsequent to receiving the system design documents, the work is shared into various modules and the real coding is commenced. The system is developed into small coding units. These units are later integrated in the subsequent phase. Every unit is tested for its functionality.

**Integration & System Testing:**

The modules that are divided into units are integrated into a complete system and tested for proper coordination among modules and system behaves as per the specifications. Once the testing is completed, the software product is delivered to the Buyer.

**Operations & Maintenance:**

It is a never ending phase. Once the system is running in production environment, problems come up. The issues that are related to the system are solved only after deployment of the system. The problems arise from time to time and need to be solved; hence this phase is referred as maintenance

**Diagram:**

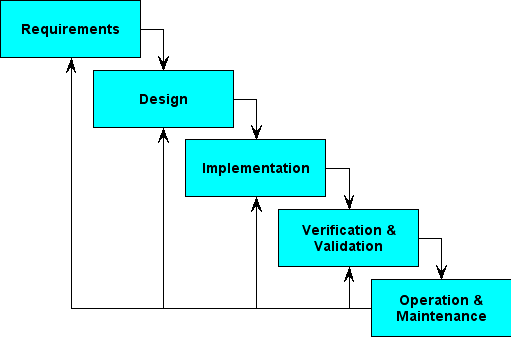


Figure 4.4 Waterfall Model

## 

## 4.3 Data Models

**ERA Model:**



Figure 4.5 ER Diagram

**Description of ERD:**

Location Based App is an offline location based android application. App users have completely held on application and its features. User can set reminder and save the location. On that location he will receive a notification of the reminder. He can also select and save the location for mobile profile. There is also the option for the user to write and save the messages and location.

## 

## 4.4 Behavioral Models

### Data Flow Models

**Context Level:**



Figure 4.6 Context Level Data Flow Diagram

**Description of Context Level:**

Location based application manages the mobile profile, reminders and messages for different locations. User can search and save location. Then there will be different options for him like set reminder, set profile and save messages. In case of messages, the system will automatically send the message to selected users.

**Level 0:**



Figure 4.7 Level 0 Data Flow Diagram

**Description of Level 0:**

The user get himself involved with the basic functionality of the application, in which he can search and save location and can also save reminders, profiles. He can also write and save the messages according to the location.

**Level 1:**



Figure 4.8 Level 1



Figure 4.9 Level 1



Figure 4.10 Level 1

**Description of Level 1:**

The user selects the features, and though them he can save the data about that feature in database.

## 

## 4.5 System Sequence diagrams

Sequence diagram describes an interaction by focusing on the sequence of messages that are exchanged, along with their corresponding occurrence.

**Define Message**

Figure 4.11 Sequence Diagram of Define Message

**Description**

The user can define messages. To define message means to write message and select contacts. Then he has to save location. When the user will reach that particular location, message will automatically send to the saved contacts.

* **Set Mobile Profile**

Figure 4.12 Sequence Diagram of Set Mobile Profile

**Description**

The user can set his mobile profile. Then he has to save location. When the user will reach that particular location, his phone will automatically go to silent mode.

* **Set Reminder**

Figure 4.13 Sequence Diagram of Set Reminder

**Description**

The user can set the reminder. Then he has to save location. When the user will reach that particular location, he will receive a reminder notification.

## 4.6 State Models

* **Define Message**



Figure 4.14 Activity Diagram of Define Message

**Description**

The figure above shows the behavior of the message feature of the app. First the user search location, then save that location, after that he have to write message and select the recipients for the message. Then the system will compare his saved location with his current location. If the user reaches at that location, then the message will automatically send to the saved recipients. Otherwise, the user can again go from the whole process.

## Set Reminder



Figure 4.15 Activity Diagram of Set Reminde

**Description**

The figure above shows the behavior of the reminder feature of the app. First the user search location, then save that location, after that he have to write notification and save it. Then the system will compare his saved location with his current location. If the user reaches at that location, then he will receive the notification. Otherwise, he has the ability to again save the Reminder and location.

* **Set Mobile Profile**



Figure 4.16 Activity Diagram of Set Mobile Profile

**Description**

The figure above shows the behavior of the mobile profile feature of the app. First the user Search location then save that location. Then the system will compare his saved location with his current location. If the user reaches at that location, his mobile phone will automatically go to the silent mode. Otherwise, he has the ability to again save the reminder and location.

## 4.7 Object Models

* **Class Inheritance Model**



Figure 4.17 Class Inheritance Model

**Class diagram Description**

In our project multiple entities are involved and all those entities and attribute of those entities is explained in class diagram. All the relationships are defined for better understanding of our application.

**4.8 Data dictionary**

**Silent Table**

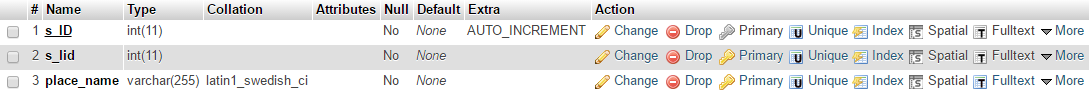
****

Figure 4.17 Silent Table

**Location Table**

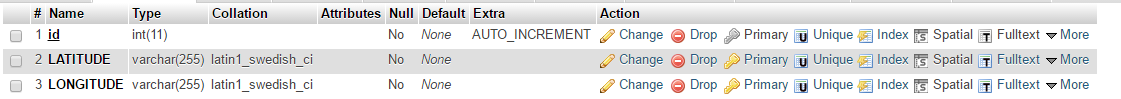
****

Figure 4.18 location Table

**Message Table**

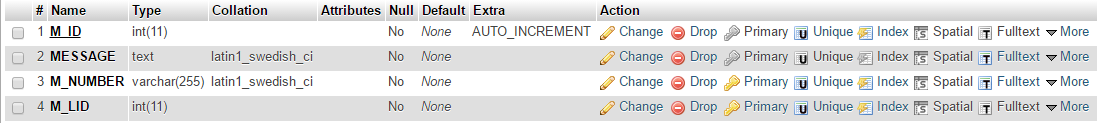
****

Figure 4.19 Message Table

**Notification Table**

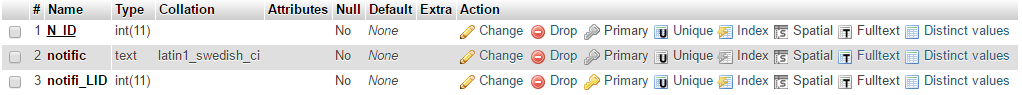
****

Figure 4.20 Notification Table

**Data Dictionary Diagram Description**

The figure above illustrates the Data Dictionary Diagram of our project. As it is shown from the figure, there are four tables in our Database. The tables “Message”, “Profile” and “Notification” has strongly depend on the table ”Location”. The table Location can only use only one instance of tables Profile and Notification. It means that for any particular location there can be only one profile and only one notification. While there can be many instances of the table Message. It means for any location message can be send to one or many users. The relation between Profile and Location and Notification and Location is of one to one relation. While, in the case of Message and Location it is one to many relation.

**4.9 Implementation Language**

* Write once, run anywhere JAVA technology promises you that you only have to write your application once for the java platform & then you will able to run it anywhere (any operating system)
* Security: The capacity of java platform allows users to download entrusted code over a network & run it in secure environment in which cannot do any harm. Because the security guarantee, nearly as strong as java makes.

**4.10 Required Support Software (pre­existing)**

This product is android application. Software requirement for applications as follows.

* OS: ANDROID 2.2 or above
* LANGUAGE: Java, xml
* CASE Tool: Android Studio

# 

# User Interface Design

## [5.1 User Interface Specification](#_Toc411446767)

A User interface specification (UI specification) is a document that captures the details of the software user interface into a written document. The specification covers all possible actions that an end user may perform and all visual, auditory and other interaction elements.

### 5.1.1 User Interface Design (Screens)

### Dashboard

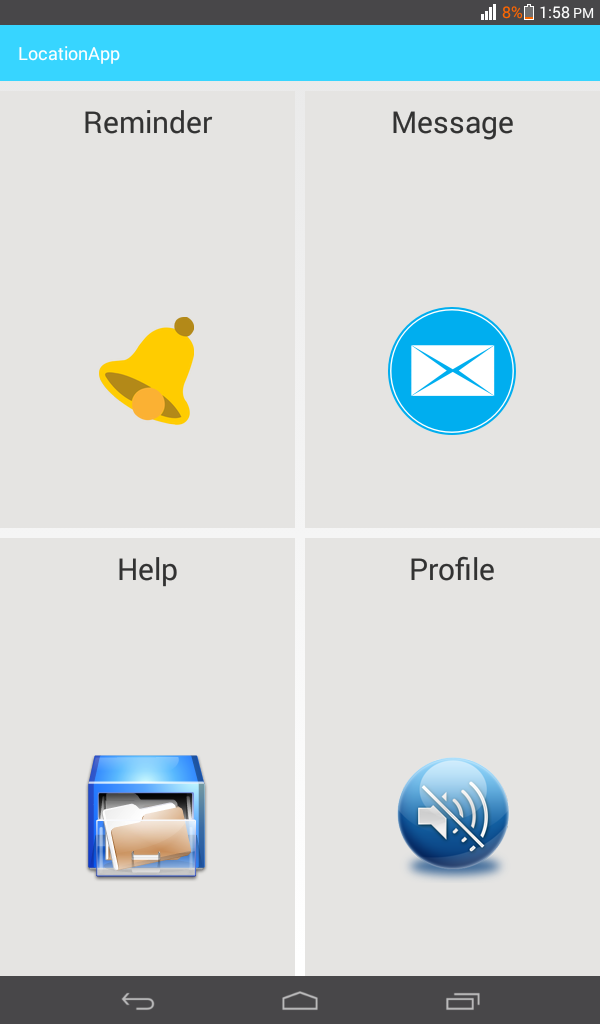
****

Figure 5.1 Dashboard

**Description:**

The screen below is the main screen of the application. This picture shows all the features and the functionality of the application.

### Reminder Screen

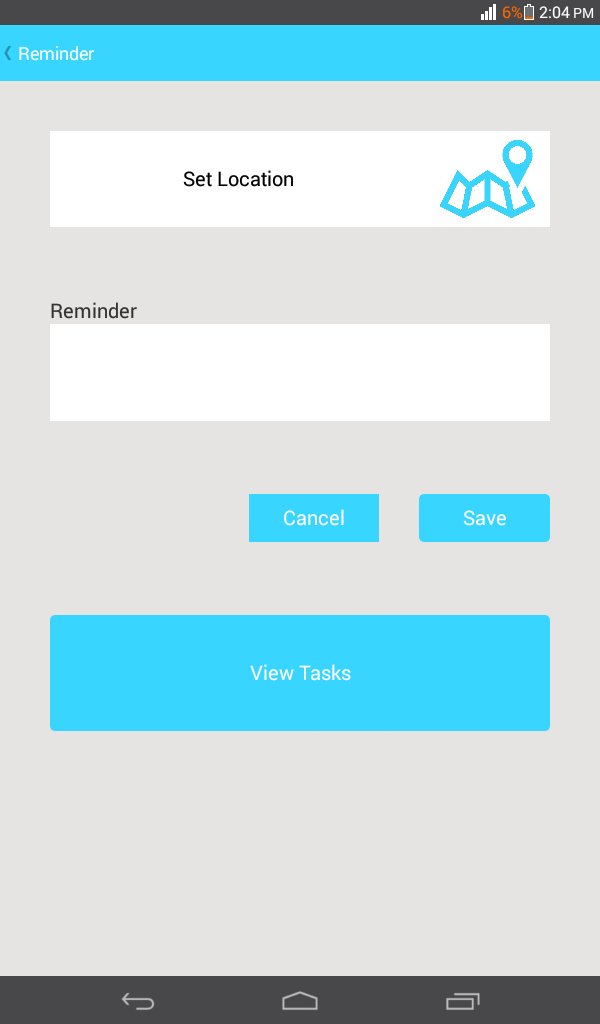
****

Figure 5.2 Reminder Screen

**Description:**

The reminder screen of the application is shown below. This picture illustrates that there is an option of selecting location. After selecting the location, the user can set the reminder, according to his desire with respect to that specific location. Then user can save his reminder. Also he can cancel the option. And can view the previous saved tasks.

### View Reminder

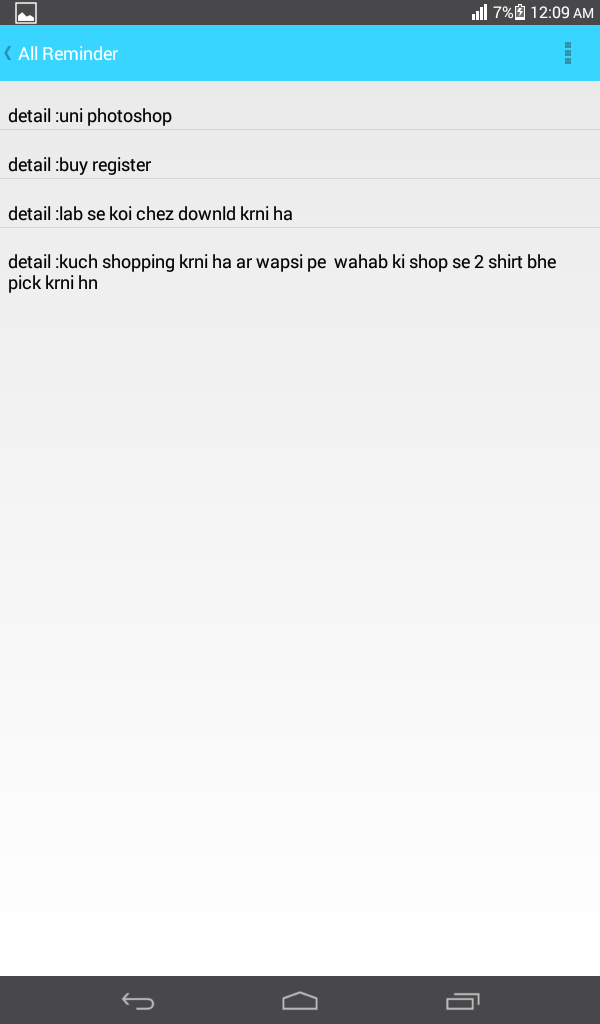
****

Figure 5.3 View Reminder

**Description:**

The user can view the previous saved reminders. In that case all the reminders details will be displayed in a list view as shown in the figure.

### Delete Reminder

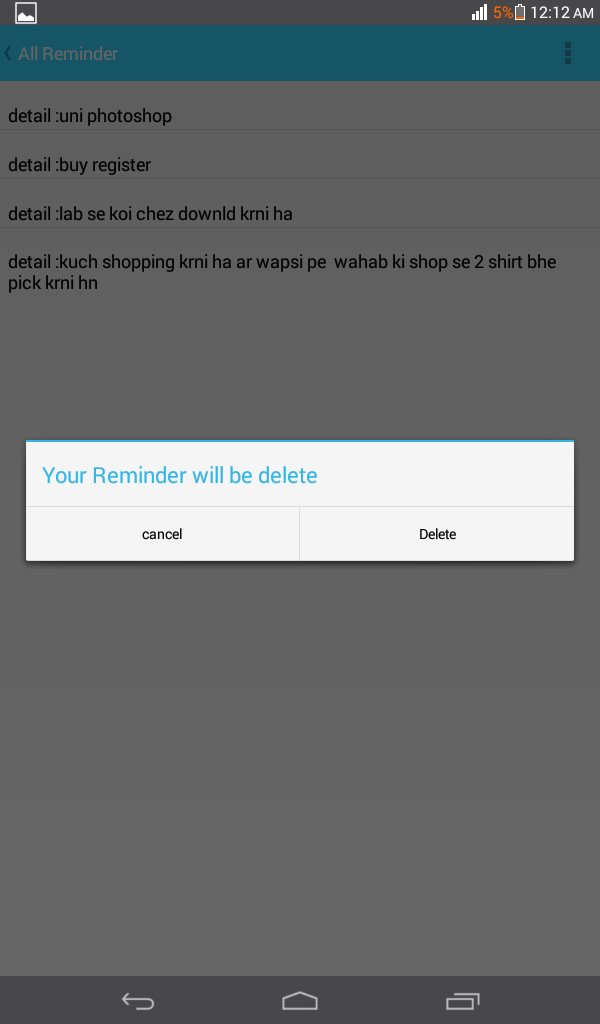
****

Figure 5.4 Delete Reminder

**Description:**

After viewing all the saved reminders, the user can delete any or all of them if need. In that a confirmation dialog will appear, that confirms that user is sure to delete the reminder.

### Message Screen

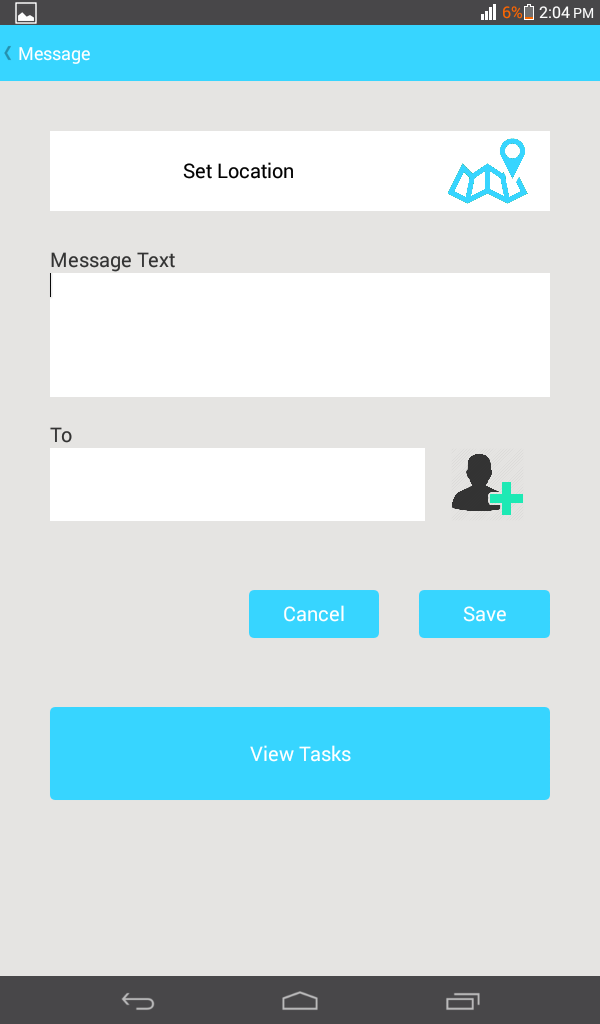
****

Figure 5.5 Message Screen

**Description:**

The message screen of the application is shown below. This picture illustrates that there is an option of selecting location. After selecting the location, the user can type the message. Then choose the contact, from the contact list. The user can select multiple recipients at a time. After saving, the message will automatically send to the recipients when the user will reach to that location.

### Add Contacts

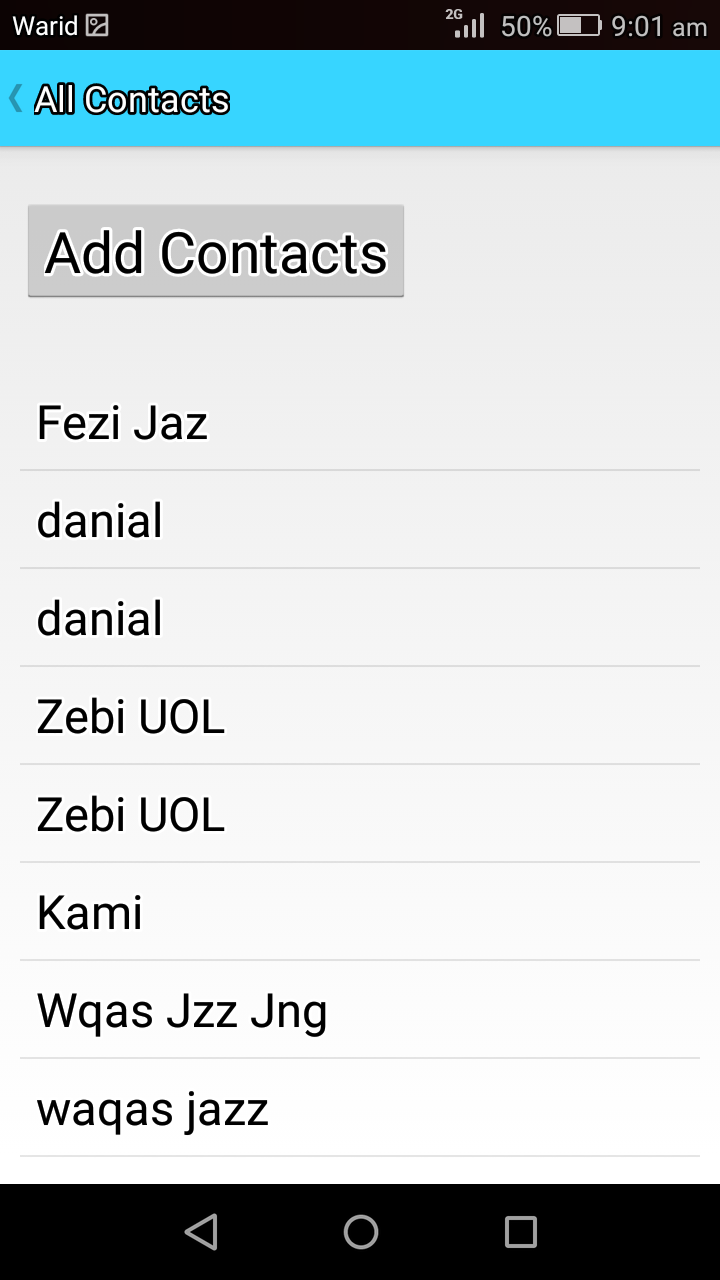
****

Figure 5.6 Add Contacts

**Description:**

The user can select the contacts from his contact book. All the contacts from his contact book will be displayed in an activity in a list view.

### Delete Message Task

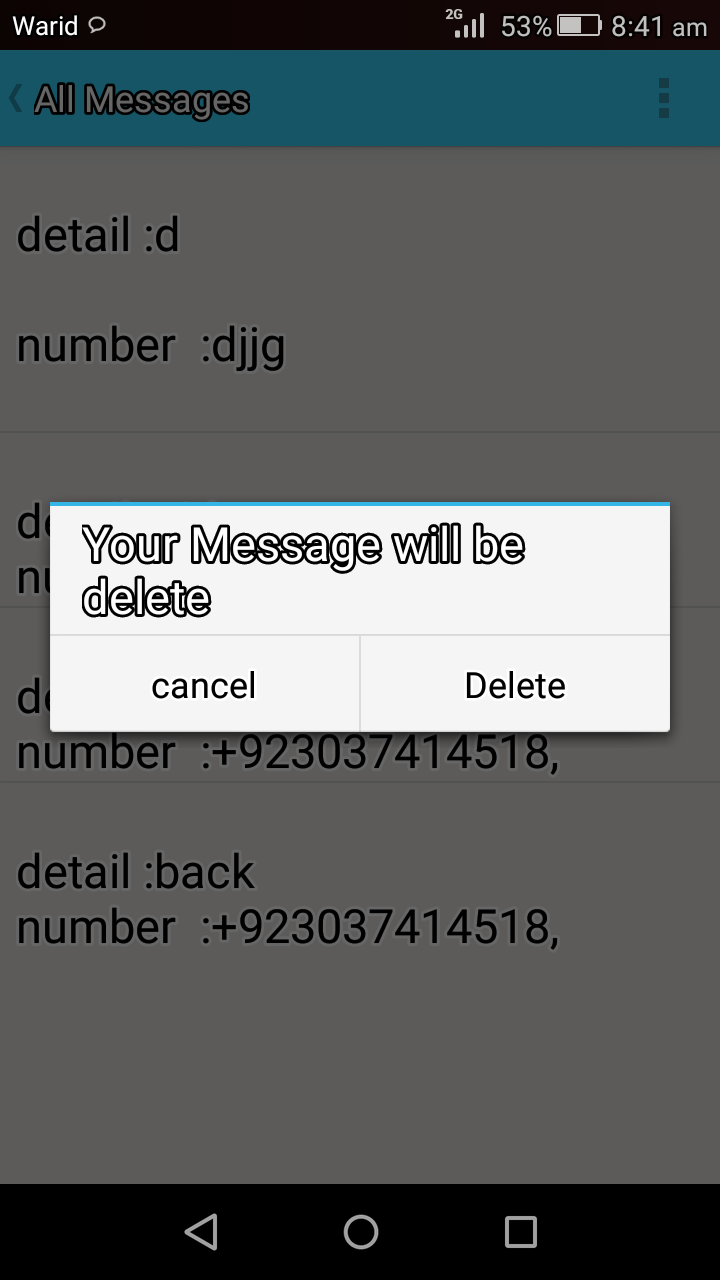


Figure 5.7 Delete Message Task

**Description:**

After viewing the entire saved task, the user can delete any or all of them if need. In that a

Confirmation dialog will appear, that confirms that user is sure to delete the message or not.

* **Profile Screen**

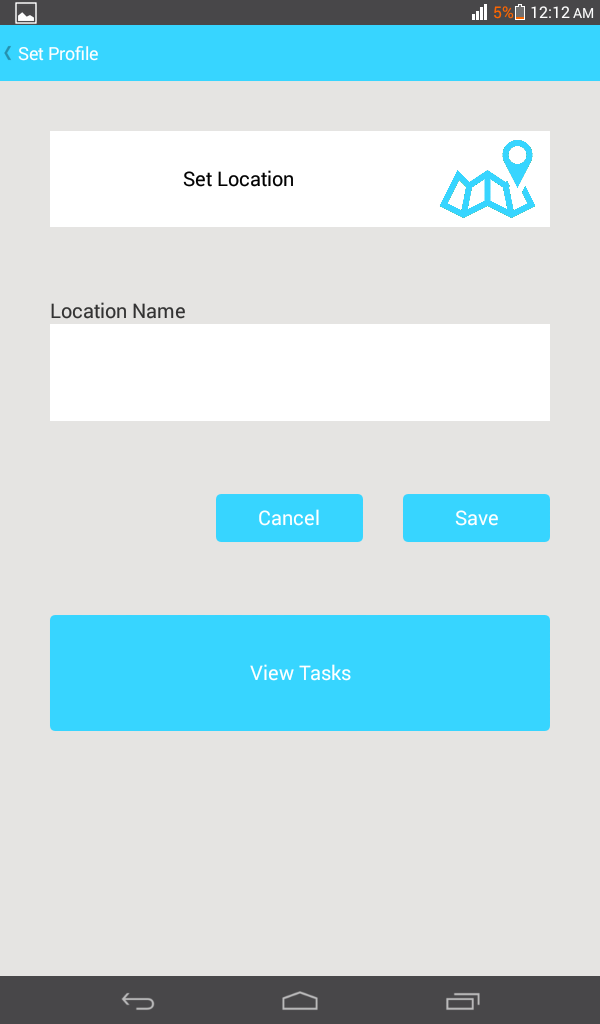
****

Figure 5.8 Profile Screen

**Description:**

The profile screen of the application is shown below. This picture illustrates that there is an

Option of selecting location. After saving the location, when user will reach that location,

His phone will automatically switch off.

### Map Screen



Figure 5.9 Map Screen

**Description:**

The picture below, demonstrates how the map activity will be shown to the user.

****

Figure 5.10 Map Screen

# 

# System Verification & Validation

## 6.1 Introduction

Testing is process of evaluating a system or its components with intent to finds bugs in software and comparing with specified requirements. Testing is execution of system in order to find gaps, errors or missing requirements.

We are IT students and we follow IEEE standards and principles. IEEE definition of testing state that “A process of analyzing a software item to detect the difference between existing and required condition and to evaluate the features of software item.

**Roles in Testing**

Roles in software testing totally depend upon the need of stakeholder of project. Large companies have proper team with responsibilities to evaluate the given requirements. Developers also conduct testing called unit testing. The following are roles involved in testing.

* Software Tester
* Software Developer
* Configuration Manager
* Project Lead/ Manager
* End user

Roles may be change from company to company and their designations. The following are some designation in software testing.

* Software Quality Assurance Engineer

**Starting of Testing**

Testing is done in different form at every phase of SDLC.

* Requirement gathering phase leads towards analysis and verification of requirements
* Reviewing the design with intent to improve the design is also testing
* Testing performed upon completion of code is also categorized as testing

**Stop Testing**

It is difficult to stop testing as testing is never ending process and no one claim that software is 100% tested. But you may stop testing

* Completion of deadline
* Completion of test case execution
* Completion of code coverage
* Management Decisions

Verification Validations

Table 6.1 Verification Validations

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Verification** | **Validation** |
| 1 | Verification addresses the concern: "Are you building it right?" | Validation addresses the concern: "Are you building the right thing?" |
| 2 | Ensures that the software system meets all the functionality. | Ensures that the functionalities meet the intended behavior. |
| 3 | Verification takes place first and includes the checking for documentation, code, etc. | Validation occurs after verification and mainly involves the checking of the overall product. |
| 4 | Done by developers. | Done by testers. |
| 5 | It has static activities, as it includes collecting reviews, walkthroughs, and inspections to verify software. | It has dynamic activities, as it includes executing the software against the requirements. |

## 6.2 Testing Levels

### Unit Testing

This type of testing perfumed by developers before project is handed over to the testing team which formally executes test cases. Goal is to isolate each part of program and show that individuals part are corrects in terms of requirements.  **Limitations of Unit Testing**

* Difficult to execute each and every path in software application
* There is limit to number of testing scenarios and test data

**Integration Testing**

Integration testing is defined as the testing of combined parts of an application to determine if they function correctly. Integration testing can be done in two ways: Bottom-up integration testing and Top-down integration testing.

Table 6.2 Integration Testing

|  |  |
| --- | --- |
| **S.N.** | **Integration Testing Method** |
| 1 | Bottom-up integration  This testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules or builds. |
| 2 | Top-down integration  In this testing, the highest-level modules are tested first and progressively, lower-level modules are tested thereafter. |

## 

## 6.2 Schedule

Table 6.3 schedule

|  |  |  |
| --- | --- | --- |
| **Testing Activities** | **Begin** | **End** |
| Designing Test Cases | **25/04/16** | **03/05/16** |
| Executing Test Cases | **04/05/16** | **20/06/16** |
| Unit Testing | **21/06/16** | **15/07/16** |
| Integration Testing | **16/07/16** | **19/08/16** |
| System Testing | **20/08/16** | **23/09/16** |

## 6.3 Control Activities

The review meeting will be conducted at June 20, 2016 and in this meeting the participants will all the members

## 6.4 Acceptance Testing

Final testing based on specifications of the end-user or Buyer, or based on use by end-users/Buyers over some limited period of time

* As we have completed our software and give it to some users to check whether it is user friendly or not?
* Are all the parts a user can use easily?
* Is it fulfilling all the requirements of the end user?

The result was satisfactory, and our system is working according to all the needs and requirements.

## 

## 6.5 Validation Testing

Following are some test cases

### 6.5.1 Location:

Table 6.4 Location

|  |  |
| --- | --- |
| **Requirement Name:**  Search Location , Show Location , Define Location, Show Location Record, Delete Location Record | |
| **Requirement ID:** Test case 001 | **Tester:** Kamran Ali |
| **Test Date:** Monday 16.09.2016 | **Results:** Cleared Yes |
| **Brief Description:** Search Location for defining of location used to perform daily tasks | |
| **Remarks:** Requirement is tested completely and hereby presented for evaluation | |

### 

### 6.5.2 Contacts:

Table 6.5 Contacts

|  |  |
| --- | --- |
| **Requirement Name:**  Search Contacts, Save Contacts | |
| **Requirement ID:** Test case 002 | **Tester:** Kamran Ali |
| **Test Date:** Monday 16.09.2016 | **Results:** Cleared Yes |
| **Brief Description:** Search contacts for saving in database. Saved contacts along with its set location will be used by application for sending of SMS | |
| **Remarks:** Requirement is tested completely and hereby presented for evaluation | |

### 6.5.3 Message:

Table 6.6 Message

|  |  |
| --- | --- |
| **Requirement Name:**  Save Message, Show Messages Record, Delete Message Record , Send Message | |
| **Requirement ID:** Test case 003 | **Tester:** Mudassar Nawaz |
| **Test Date:** Monday 16.09.2016 | **Results:** Cleared Yes |
| **Brief Description:** Application will send message to specified contacts when user will enter the defined location area. | |
| **Remarks:** Requirement is tested completely and hereby presented for evaluation | |

### 6.5.4 Reminder:

Table 6.7 Reminder

|  |  |
| --- | --- |
| **Requirement Name:**  Set Reminder, View Reminder, Delete Reminder , Write Notification, Save Notification | |
| **Requirement ID:** Test case 004 | **Tester:** Kamran Ali |
| **Test Date:** Monday 16.09.2016 | **Results:** Cleared Yes |
| **Brief Description:** Application will ring with the saved notification according to location defined by user | |
| **Remarks:** Requirement is tested completely and hereby presented for evaluation | |

### 6.5.5 Mobile Profile:

Table 6.8 Mobile Profile

|  |  |
| --- | --- |
| **Requirement Name:**  Set Mobile Profile | |
| **Requirement ID:** Test case 005 | **Tester:** Mudassar Nawaz |
| **Test Date:** Monday 16.09.2016 | **Results:** Cleared Yes |
| **Brief Description:** Application will set profile” Silent” according to defined location | |
| **Remarks:** Requirement is tested completely and hereby presented for evaluation | |

## 6.4 Justification of test cases

Justification term is use for checking of cases in other words this is a action of showing for something is right or reasonable. In justification of test case we check our software or our android based application fully meat to our requirements or not. Justification of the test cases makes sure the working of our application.

## 6.5 Test Run Procedures & Result

Running a Test enables you to simulate real end user android activity and to collect performance data from the components of the system under test. Use the results you produce during a Test-run to help evaluate the performance of target android Application Environments. You can monitor the progress of a Test-run and all of the Task Groups it contains from the Monitoring tab view of the Test Pane. Running a Test is a straightforward procedure, because the Task Group settings of the Collectors and Scripts you include in the Test have already been specified during Test creation

# 

# Conclusions

## 7.1 Summary

Our project is an android application which will serve as platform to integrate basic android cell phone activities in a single location with respect to location. It is being built as new self-contained product. Market has task management application that runs with GSM and locations but all the type of applications are separate. This application is one of such type of application that will integrate activities in a single application that will run on location. So that’s why it is not part of larger system or it is not up gradation of existing application.

## 7.2 Problems Encountered and Solved

**Problems Due to Lack of Knowledge and Experience on Subject**

Our team has encountered in an area in which we have no background; therefore, lack of knowledge and experience will probably get each of team members into problems. If the team member is in luck at that moment, this problem will just be equal to loss of time. However, this problem can get the team into a break point as well. Not to encounter this kind of problems we are trying to be well prepared for weekly progress meetings, as they secure to get us into sufficient knowledge.

**Problems Due to Misunderstood Requirements**

Although our team recorders try hard for understanding the requirements; our team can ever misunderstand the requirements for the project in any design phase; which cause loss of time. To overcome this problem, we agreed on that the most important part of our weekly meetings should be the exchange of ideas.

**Problems Due to Large Project Size and Unrealistic Scheduling**

Our team is supposed to arrange the schedule in an early phase of the project. However, three of us have never participated in such a large project and we do not have enough experience either. Therefore, the probability of arranging an unreasonable schedule is round the corner for us. Later in the project, if we ever feel that our schedule is unreasonable to follow, then we will enhance our schedule according to the situation we will be in.

**Problems Due to Choice of Tools, Libraries**

As a team, we are taking pain over choosing the tools, libraries that will be used in our project.

However, wrong decisions about them could already be made. In such a case, changing these elements after design phase will be probably difficult; at least causing loss of time owing to the extra reading needed.

**Problems Due to the Communication Gap between Team Members**

In such a large project without any experience, it is unavoidable for each of team members that in developing process there would be divergence about emphasizing their own crucial ideas. Therefore, this can give birth to hidden competition. Nevertheless, by playing cards open about what everyone wants to perform within in the project, this insidious enemy could be eradicated from the start.

**7.3 Suggestions for Future Extensions to Project**

We are living in digital world of information technology where change in software system is necessary and you cannot claim that developed product is complete solution for rest of life. Requirements of users change day by day. Every product carries a space for improvement and extension in requirements. This developed product also has space for future work. The following are some of considerations for future work and enhancement in the application.

* Application will be merged with other location based application designed for nearest hospitals, hostels, restaurant’s, shopping malls.
* Application will be used for tracking locations for family safety
* In next phase we automate more tasks of our daily routine.
* In case of Mobile switch off, we predict the reason behind and inform desired person.

**Bibliography**

**Books**

Java: The Complete Reference - by Herbert Scheldt

Introduction to Java Programming - by Y. Daniel Liang

## References

1. Shah, P., Gadgil, R., & Tamhankar, N. (2012). Location based reminder using gps for mobile (android). ARPN Journal of science and Technology, 2(4), 377-380.
2. Bejerano, Y., & Cidon, I. (2001). Efficient location management based on moving location areas. In INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE (Vol. 1, pp. 3-12). IEEE.
3. Zhu, T., Wang, C., Jia, G., & Huang, J. (2010, August). Toward context-aware location based services. In Electronics and Information Engineering (ICEIE), 2010 International Conference On (Vol. 1, pp. V1-409). IEEE.
4. <https://github.com/codepath/android_guides/wiki/Must-Have-Libraries>
5. <https://developer.android.com/guide/components/activities.html>