

FORTIFY ME



By

Maj Faisal Jatoi

MajUsamaNisar Khan

Capt Imran Aziz

Submitted to the Faculty of Computer Science, Military College of Signals
National University of Sciences and Technology, Rawalpindi in partial fulfillment for

The requirements of a B.E Degree in Computer Software Engineering

June 2015

CERTIFICATE

Certified that the contents and form of project report entitled **“FORTIFY ME”** submitted by Major Faisal Jatoi, Major Usama Nisar Khan and Captain Imran Aziz have been found satisfactory for the requirement of the degree.

Supervisor: _____

Brigadier (Doctor) Faheem Arif

Department of CS

Military College of Signals

ABSTRACT

FORTIFY ME

Mobile phone is basic need of modern society. To cater any unforeseen tragedy in a routine day to day life, it is extremely important to have a quick information facility to let the loved ones know the current situation and status. Manual exercise in this causal task may become a rampage in itself and therefore be impractical. This application provides an easy answer to this problem, application itself updates the location and status. User, in any kind of emergency would just have to press one click on his smart phone and a consolidated information about his problem is delivered to his dear ones within seconds. The system consists of two parts. An android application running on smart phone and a java based server. The aim of system is to bring maximum possible ease to the user during any emergency. Predefined messages stored in the application are triggered on the single click of the user. Messages are directed to predefined numbers using GSM module, moreover location and profile of the user is sent to the nearby people by using internet module. Should there be any delay in propagation or navigation due to signal strength, the application re-directs continuously until the message is sent. The server side consists of a running application which caters the requests from the user, it also contains a database which is constantly updated with the location of users. The server application decides number of users in the closest locality and re-directs the user request to nearby people. The GUI of the application is built with minimum user interactions so as to reduce time. Multithreading has been implemented so the application may perform various tasks simultaneously in order to reduce processing time.

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

To all the Reformed Institutions of Pakistan, dedicated to Allah and His kingdom and,

In particular to the faculty, staff and students of Military College of Signals

ACKNOWLEDEMENTS

We would like to express our profound gratitude to our parents for their prayers and continuous moral support throughout degree program in general and our project in particular. We are very much grateful to our Project Supervisor Brig. Dr. Fahim Ariff for his support and critique which helped us reach new levels of perfection. We are highly indebted to all the faculty members who throughout the degree trained, mentored and guided us to achieve new levels of professionalism which is the hallmark of this prestigious institute.

1. Table of Content

1. Introduction.....	8
1.1 Purpose	8
1.1.1 Server Module:	8
1.1.2 Cell phone Module:	8
1.2 Background.....	9
1.3 Problem Statement.....	10
1.4 Objectives	10
1.5 Deliverables	10
1.6 Technological requirements.....	11
1.6.1 Operating Systems	11
1.6.2 Software Packages	11
1.6.3 Hardware Components	11
2 Literature Review	11
2.1 Introduction.....	11
2.1.1 Background.....	11
2.2 Overall Description.....	12
2.3 Similar Software and work	12
2.3.1 Red Alert.....	12
2.3.2 Panic App.....	13
2.3.3 SOS Me !	14
2.4 Related Papers/Articles.....	15
2.4.2 Managing the Android Activity Lifecycle.....	Error! Bookmark not defined.
1.1. Classes in Properties Animation API.....	Error! Bookmark not defined.
2. 3 System Requirements	Error! Bookmark not defined.
2.1. 3.1 Introduction.....	Error! Bookmark not defined.
2.2. 3.2 Product Perspective.....	Error! Bookmark not defined.
2.3. Red Alert.....	20
2.4. 3.3 Basic Assumptions.....	21
2.5. 3.4 Operating System.....	21
2.6. 3.5 Operating Conditions	22
3.6 System Features	22
3.6.1 SOS SMS Generation	22
Description and Priority	22
Stimulus/Response Sequences.....	22
2.6.1 Normal Flow:	22
2.6.2 Alternate Flow:.....	22
Functional Requirements.....	22
3.6.2 SOS Server Generation.....	23
Description and Priority	23
Stimulus/Response Sequences.....	23
2.6.3 Normal Flow:	23
2.6.4 Alternate Flow:.....	23
Functional Requirements.....	23
3.6.3 Facebook Integration	24
Description and Priority	24
Stimulus/Response Sequences.....	24
Functional Requirement	24
3.6.4 Profile Creation.....	24
Description and Priority	24
Stimulus/Response Sequences.....	25
Functional Requirement	25
3.6.5 Profile edit	25
Description and Priority	25
Stimulus/Response Sequences.....	25
Functional Requirement	26

3.7	Nonfunctional requirements	26
	Performance Requirements	26
3.7.1	System Response.....	26
3.7.2	Time of Data Backup	26
3.6	Safety Requirements.....	26
3.7	Security Requirements.....	27
	Data encryption will be used when data is communicated over the communication channel in order to cater for security concerns.....	27
3.8	Software Quality Attributes	27
3.8.1	Adaptability.....	27
3.8.2	Availability.....	27
3.8.3	Backup.....	27
3.8.4	Capacity.....	27
3.8.5	Compatibility.....	27
3.8.6	Competitive edge.....	27
3.8.7	Compliance.....	27
3.8.8	Documentation	27
3.8.9	Efficiency	27
3.8.10	Flexibility	28
3.8.11	Interoperability	28
3.8.12	Maintainability	28
3.8.13	Portability	28
3.8.14	Reliability	28
3.8.15	Robustness.....	28
3.8.16	Scalability.....	28
3.8.17	Security.....	28
3.8.18	Stress	29
3.8.19	Start-up time	29
3.8.20	Timeliness	29
3.8.21	Usability	29
4	Design and Development	29
4.1	Introduction.....	29
4.2	Architectural Model.....	29
2.6.5	Client.....	30
2.6.6	Server.....	30
4.2	Design Pattern.....	30
4.3	Logical View	31
4.3.1	Flow Chart Diagrams	31
4.3.1.1	First is the flow chart of client module.....	31
2.6.7	Here is the flow chart of server module	32
4.3.2	Use Case Diagrams	32
4.3.2.1	System Use Case Diagram	32
4.3.2.1.1	Create Profile Use Case Diagram	33
4.3.2.1.2	Edit Profile Use Case Diagram.....	34
4.3.2.1.3	View Profile Use Case Diagram.....	35
4.3.2.1.4	SOS Use Case Diagram.....	36
4.3.2.1.5	SOS Use Case Diagram.....	37
4.3.3	Sequence Diagrams	38
4.3.3.1	Create Profile.....	38
4.3.3.2	View Profile Sequence Diagram	38
4.3.3.3	SOS SMS Sequence Diagram	39
4.3.3.4	SOS Server Message Sequence Diagram	40
4.3.3.5	Facebook update status Sequence Diagram.....	40
4.3.3.6	Receive SOS Sequence Diagram.....	41
4.4	Implementation View	42
4.4.1	System Class Diagram.....	42
4.5	Dynamic View	44

4.5.1	Activity Diagrams.....	44
4.6	User Interface Design	49
4.6.1	Main Window	49
4.6.2	SOS Window	50
4.6.3	Profile Window.....	51
4.6.4	Create / Edit Profile Window.....	52
5	System Implementation	52
3.1	Tools and Technologies	52
3.1.1	Eclipse SDK.....	52
3.2	Software Implementation.....	53
3.2.1	Profile Creation.....	53
3.2.2	SMS implementation	54
3.2.3	SOS Server Message.....	54
3.2.4	Facebook Status Update.....	54
3.2.5	Location Retrieval	55
3.2.6	Street Location.....	55
6	Project Analysis and Evaluation.....	55
6.1	Testing.....	55
6.1.1	Testing Introduction.....	55
6.1.2	Testing Levels.....	55
6.1.3	Unit Testing.....	55
6.1.4	Integration Testing	60
a.	First phase of testing.....	62
b.	Second phase of testing	63
a.	First phase.....	64
b.	Second phase	64
a.	First phase.....	64
b.	Second phase	64
c.	Third phase	64
a.	First phase:	65
i.	Entry criteria:.....	65
ii.	Exit criteria:	65
iii.	Integration Technique to be used:	65
iv.	Test configuration set-up:.....	65
b.	Second phase:.....	66
i.	Entry criteria:.....	66
ii.	Exit criteria:.....	66
iii.	Integration Technique to be used:	66
iv.	Test configuration set-up:.....	66
c.	Third phase:.....	66
i.	Entry criteria:.....	66
ii.	Exit criteria:.....	66
iii.	Integration Technique to be used:	67
iv.	Test configuration set-up:.....	67
a.	First phase:	67
i.	Interface Integrity	67
b.	Second phase:.....	68
6.1.5	System Testing	69
6.1.5.1	Basic Tests.....	70
6.1.5.2	Functionality Tests	70
6.1.5.3	Scalability Tests	71
6.1.5.4	Performance Tests	71
6.1.5.5	Robustness Tests	71
6.1.5.6	Documentation Tests.....	72
6.1.6	Acceptance Testing	72
6.1.6.1	Functional Correctness and Completeness.....	73
6.1.6.2	Accuracy.....	73

6.1.6.3	Data Integrity.....	74
2.6.8	75	
6.1.6.4	Backup and Recoverability.....	75
6.1.6.5	Usability	75
6.1.6.6	Performance.....	76
6.1.6.7	Robustness.....	77
6.1.6.8	Confidentiality.....	77
2.6.9	78	
6.1.6.9	Compatibility and interoperability	78
6.1.6.10	Load Testing.....	79
6.1.6.11	Schedule:	79
6.1.6.12	Human Resource:	79
6.	80	
6.2	Summary.....	80
6.3	Results and Analysis.....	80
6.3.1	Results and Analysis Introduction	80
6.3.2	Results.....	80
6.3.3	Analysis	80
7	Conclusion and Future Work	81
3.	Appendix A: Glossary.....	81
4.	Appendix B: Bibliography	84
Appendix C:	User Manual	121

1. Introduction

1.1 Purpose

“Fortify Me” is an android based solution for catering emergency situations to any common man in routine life. The concept revolves around an SOS notification generator from the victims smart phone and primarily be delivered to all nearby recipients. The message will use GSM along with 3G and Wi-Fi module to propagate its navigation to the recipients. This will help in reliable and swift flow of desired information. Multiple channel redeemer is used for delivering it across maximum people. The Notifications will be generated through pre-defined responses coupled with the senders profile and location based on GPS input. Apparently single interface will be used for notifications to be delivered to all nearby persons in locality, pre-defined close contacts and a status update on Facebook. The Application will have two modules

1.1.1 Server Module:

The Server module will contain database of the users containing their profiles and current location. One receipt of any SOS message from any user, query will be triggered to search for the individuals in close locality, the message shall then be delivered to all the users who fall in the criteria.

1.1.2 Cell phone Module:

The cell phone module will be the core. It will process all the requests that user wishes to generate. Predefined responses will be shown on the main screen, location and profile name will be sent

through GSM, the same will be sent through server, which will contain profile picture and location plotter as well.

The application when installed on mobile phone would be able to convey predefined or custom responses to 5 selected numbers through GSM and as many contacts as possible in closed locality. The main features are

- You can Create, Edit and Manage profile
- You can generate alert notification by Selection of Emergency types
- SOS message generation will be sent through two SMS and Internet
- Automatic Selection of Intended Audience based on close locality
- Security of Data through Encrypted Transmission
- Automatic status updating through Facebook integration.
- Location Monitoring in case of kidnapping.

1.2 Background

The root cause of creating this application is that there is no such product of this type in the market. The aftermath of 9/11 has brought a new environment which is rife with security concerns for every person. Not only security, but the pace of life has shrewdly increased and has waded off personal relationship. Fortify me is one such action to bring back our cultural life. It is an attempt to create a family atmosphere aiming to provide all possible help in case any eventuality occurring nearby you. This app will utilize all possible means to swiftly convey emergency message to close friends and persons.

1.3 Problem Statement

Suppose you are travelling on road and all of sudden u encounter terrorist who aim to attack you for kidnapping purpose, the only way out of help in such a situation would be to contact your friends, family, co-workers or any individual of your known circles. Now in the heat of emergency, making call would simply be out of question, on the other hand writing text would take long workout to describe the situation. One requires an assistant who should understand the situation be less demanding attention.

Keeping in mind these facts we have developed an application that minimize the effort and time required to convey the message. Provides efficient user interface and data sending techniques, since the possibility may arise whereby one channel of transmission may be temporary unavailable, therefore all the necessary means of transportation have been catered for.

1.4 Objectives

Objective is to develop android based application which will send an emergency message to all the nearby contacts along with location, and profile via internet protocol and SMS message to 5 friends and family numbers via GSM module. Alongside a server application which aims to provide a host for all the requests received through internet, and relay the message to all the nearby contacts.

1.5 Deliverables

First Progress Report: including SRS Document

Second Progress Report: including System Design (Hardware and Software)

Third Progress Report: Demo of integrated system

Final Report: including complete documentation of the system and user manual.

1.6 Technological requirements

1.6.1 Operating Systems

- Windows
- Android

1.6.2 Software Packages

- Java
- Eclipse
- Android SDK

1.6.3 Hardware Components

- Personal Computer
- Android Phone

2 Literature Review

2.1 Introduction

2.1.1 Background

Android is the leading operating system in smart phones. With a market share of more than 80 percent it is the leading operating system in smart phones and tablets. With the major brands like Sony, HTC and Samsung developing android based devices it is the most widely used operating system. As of September 3, 2013, 1 billion Android devices have been activated.

Android is an operating system based on the Linux kernel, and designed primarily for touchscreen mobile devices such as smartphones and tablets. Android is open source and Google releases the source code under the Apache License. This open-source code and permissive licensing allows the software to be freely modified and distributed by device manufacturers, wireless carriers and enthusiast developers. As of May 2013, 48 billion apps have been installed from the Google Play store.

2.2 Overall Description

The concept of this software originates when effectively using your android smart phone while working in emergency. Usability, time management, improved efficiency is the aim.

The detailed description of the software is presented in the software requirement specification document above.

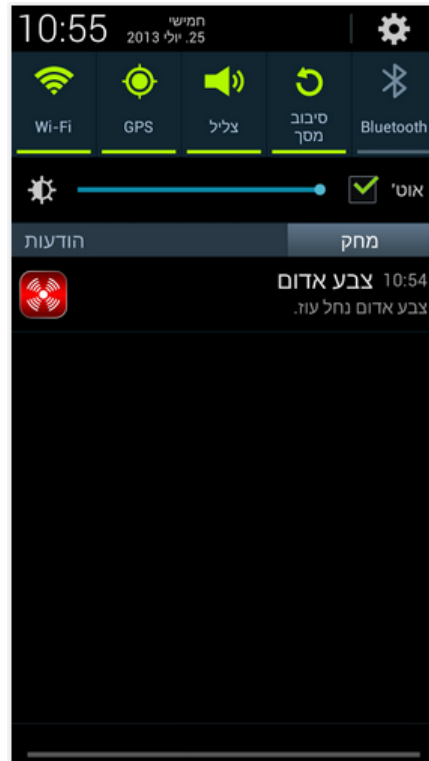
2.3 Similar Software and work

2.3.1 Red Alert

Israeli based safety app use for alerting citizen's in case of any attack from Hamas, provides location and emergency type (restricted use only). The official app for Android now red color enhanced version and the iPhone version is identical. Red color is a tool that will allow you to experience explained the grim lives caused to the residents of the south.

The app will allow you to receive alerts in real-time red paint directly from your iPhone.

You will also receive regular updates on launches, landings and security events in the South.



2.3.2 Panic App

This security application is designed to be an aid during distress/panic. It acts as an emergency Panic Button that alerts your family and friends in the possible moments of danger. After a brief set up, the panic app is ready to be used at any time you require assistance. Just HIT the virtual panic button in the center, and the application will immediately dial one pre-defined number and send a Panic SMS to up to 5 contacts informing them that you require assistance and your location on Google Maps.

A second call option is also provided in case the primary contact is not available. The application can also be used to call Emergency Services such as the Police, Fire department or Medical Services. The purpose of this application is to assist you when in need. However, it is dependent on various factors that may need your attention:

1. The Call and Text messages are charged as per usual network rates. It is therefore important to check your airtime balance status.
2. The GPS co-ordinates received are network dependent.

3. Depending on network traffic a slight delay may occur during transmission of coordinates.

4. The recipient of the text message will require a web browser enabled phone to receive the locations on Google maps. With phones without a web browser, only coordinates will be shown.



2.3.3 SOS Me !

If you are in danger and need help you can send with SOS ME an alert message to your email list and share your location and help message in real-time to the World in the website SOSME.AT. You

only have to configure your settings and click over the S.O.S. button when you need help. While the alert is activated, you will see a red light flashes in the upper right of the screen.



2.4 Related Papers/Articles

2.4.1 The Android Operating System

Butler, G., Finley, D., & Kasavana

The paper gives in an overall view of Android operating system. It covers the following topics:

- Android history
- Differences of Android from other mobile operating system
- Linux kernel

- Libraries and Dalvik virtual machine
- Android based devices and their specifications
- Android applications framework

It also provides detail on android's market share along with advantages.

2.4.2 Managing the Android Activity Lifecycle

Article on Android Developer Website explaining different states and android application can be and how to handle events pertaining to application state.

2.4.3 What's the big IDE? Comparing Eclipse and NetBeans

Webpage on Comparison of IDE options for our Android Application Development Project.

2.4.4 Development Techniques for Android Platform Mobile Device Application

Ivan Njunjic

This thesis focuses on Android application development techniques needed to implement a mobile application. This thesis provides a unique approach to Android development due to its single focus on the data. The topics of interest in this paper were:

- Android application environment
- Android application structure
- Android application file structure
- Development techniques
- Android API's access and understanding.

2.4.5 Android Application Fundamentals

Developers.android

This article is a complete guide to understand android application structure and fundamentals. It explains all the application components by detail along with examples.

The topics covered are:

- Android application lifecycle
- Activities
- Services
- Content providers
- Broadcast receivers
- Android manifest file
 - o User permissions
 - o API levels
 - o API libraries

It gives a complete guide about these topics explaining with different examples and code segments.

2.4.6 Android Application Files

Developers.Android

This article deeply discusses the files and directories that android application is composed of. The author explains all the files in a very good way which is easy to learn and implement. The android application consists of the following files

2.4.6.1 Activity.java

This file contains all the java code that runs the application. It contains all the business logic and the related java code. It can be of any name.

2.4.6.2 Activity.xml

It contains the xml code of the application. The GUI of the application is built using xml. The views, lists, content providers all the made using xml. It can be of any name.

2.4.6.3 Androidmanifest.xml

This file is one of the most important files of the app. It contains the related permissions the app is going to use. The permissions to access specific hardware and software e.g. camera, GPS etc.

This article also gives an overview of the folders and directories of content e.g. pictures that is to be used in the android application.

2.4.7 Sockets programming in Java

Qusay H. Mahmoud

This is a complete guide to sockets and socket programming in java. As require network communication so it is necessary that we understand how to communicate between two devices programmatically. This article provides deep understanding to sockets, ports and how to program them and communicate effectively. The topics covered are:

- TCP and UDP introduction
- Programming sockets
- Using ports
- Java libraries that are required for socket programming
- Client server introduction

2.4.8 Android working with GPS

Ravi Tamada

A tutorial that covers how to access and work with GPS coordinates in a number of ways. It also describes how to handle and store images taken. The tutorial clearly explains:

- Accessing GPS module
- Permissions required
- Taking and storing coordinates

2.4.9 Classes in Properties Animation API

The superclass of the animation API is the Animator class. Typically the ObjectAnimator class is used to modify the attributes of an object.

You can also add an AnimatorListener class to your Animator class. You can use this listener to perform actions before or after a certain animation by calling the Listener object. Using animate () method on a View object is used to perform simultaneous animations. You can set different attributes of animation like the duration.

2.4.10 Views in Android

Views are responsible for measuring, layout and drawing themselves and their child elements. Everything that draws on the UI of the Android application utilizes the view class and is an instance of the view class. Views are also responsible for saving their UI state and handling touch events. You can also create a new view from existing View Objects. The following topics related views are covered here:

- Responsibility of views

- Reasons for creating views
- Defining views

3 System Requirements

3.1 Introduction

This chapter describes the requirements specifications for Fortify Me.

3.2 Product Perspective

The existing applications which allows to cater emergency situations are integrated in Fortify Me and some new and necessary features are added which were not in any of existing applications. The ancestors of Fortify Me are:

Red Alert	Available on Google Play Store with a rating of 4.5/5 with approximately 5 – 10 million installations per month.
Panic App	Call and text facility manager for emergency contacts, very popular with developer community.
SOS ME!	Application basically focused on HTTP protocol, for transferring of location via email to a few email contacts.

In order to cope up with the increasing efficiency with regards to processing time, fortify me is intended to prove itself as an optimum tool for effective data and task management and also improving not only the productivity but also the improving the effectiveness and efficiency. Comparison between our project and some of the similar products available in market is given below.

	Fortify Me	Panic App	SOS ME	Red Alert
Call	No	Yes	Yes	No
Sms	Yes	Yes	Yes	Yes
Location	Yes	No	Yes	No
Internet	Yes	No	Yes	no
Profiles	Yes	No	No	No
Social media integration	Yes	No	No	No

Table 3-1: Comparison of Fortify Me with Similar Applications

It is a complete system whose one module is installed on server and other on mobile. The system model consists of a cell phone mainly and a hosted application on server, it provides GUI for mobile applications only. A java based desktop application is run on server whereas an android application runs on the android device. The communication between the two devices is through Wi-Fi.

3.3 Basic Assumptions

The project is based on following assumptions:

- Android based mobile will be available.
- Windows pc will be available.
- Wi-Fi connection will be present for the connection between mobile and computer.
- User will be familiar with the windows and android.

3.4 Operating System

To run the software user be will required to have a smart phone with android operating system.

3.5 Operating Conditions

Wi-Fi and GSM signals should be available to run the application successfully.

3.6 System Features

3.6.1 SOS SMS Generation

Description and Priority

This feature will allow user to send SMS from the app to the required recipients with certain details. The priority of this feature is 9 (note 9-highest).

Stimulus/Response Sequences

Normal Flow:

Stimulus	User clicks SOS Button.
Response	System gives the option to choose from Emergency type.
Stimulus	User clicks on a type of Emergency.
Response	Form appears with certain fields that need to be filled.
Stimulus	User clicks send Button.
Response	System retrieves the required details (name, location, type of emergency) and sends an SMS to the intended recipients.

Alternate Flow:

- 1- If the user wants to go back and presses the back button in selecting emergency type window the system will go the main window of module.

Functional Requirements

Requirement 1: The system should send SMS to recipients.

Requirement 2: The system should retrieve the users location based on GPS.

3.6.2 SOS Server Generation

Description and Priority

This feature allows system to send SOS message to server which then sends messages to intended recipients. The priority of this feature is 9.

Stimulus/Response Sequences

Normal Flow:

Stimulus	User clicks SOS button.
Response	System gives the option to choose from Emergency type.
Stimulus	System choses emergency typ.
Response	System shows a form where fields are needed to be filled.
Stimulus	User fills and clicks send button.
Response	System will retrieve the information and send message to sever. Server calculates the nearest people based on their location and sends them the message which pops up on the recipient screen.

Alternate Flow:

- 1- If the user wants to go back and presses the back button in selecting emergency type window the system will go the main window of module.

Functional Requirements

Requirement 1: The system should be able send message to server over Wi-Fi.

Requirement 2: The server end should be able to calculate nearest people.

Requirement 3: The server should send message to the required recipients.

Requirement 4: A pop up should appear on the recipients phone about SOS.

3.6.3 Facebook Integration

Description and Priority

The feature will allow Facebook integration and update status in case of an SOS. The priority of this feature is 8.

Stimulus/Response Sequences

Normal Flow:

Stimulus	User clicks SOS button
Response	System gives option of emergency type.
Stimulus	User selects emergency type
Response	System shows a form where fields need to be filled.
Stimulus	User clicks the send button
Response	System updates a status about SOS on Facebook.

Alternate Flow:

- 1- If the user wants to go back and presses the back button in selecting emergency type window the system will go the main window of module.

Functional Requirement

Requirement 1: System should connect to user's Facebook account.

Requirement 2: System should update a status.

3.6.4 Profile Creation

Description and Priority

The feature will allow user to create profile. The priority of this feature is 8.

Stimulus/Response Sequences

Normal Flow:

Stimulus	User clicks Profile button
Response	System gives fields which you need to fill.
Stimulus	User fills the form fields.
Response	System saves the required fields.
Stimulus	User clicks the save button.
Response	System saves the profile permanently.

Alternate Flow:

- 1- If the user wants to go back and presses the back button window the system will go the main window of module.

Functional Requirement

Requirement 1: System should allow user to create a profile.

Requirement 2: System should save the profile.

3.6.5 Profile edit

Description and Priority

The feature will allow user to edit profile. The priority of this feature is 8.

Stimulus/Response Sequences

Normal Flow:

Stimulus	User clicks Profile edit button
Response	System gives fields which you need to fill.

Stimulus **User fills the form fields.**
Response System saves the required fields.

Stimulus **User clicks the save button.**
Response System saves the profile permanently.

Alternate Flow:

- 1- If the user wants to go back and presses the back button window the system will go the main window of module.

Functional Requirement

Requirement 1: System should allow user to edit a profile.

Requirement 2: System should save the profile.

3.7 Nonfunctional requirements

Performance Requirements

3.7.1 System Response

The system should response to the user's action within approximately 2-5 seconds or more depending on the hardware and connection speed between the two modules. Response here implies that the task started should be initiated in this duration otherwise an error should be indicated.

3.7.2 Time of Data Backup

Data backup of server will greatly depend on the amount of data present. The size of data determines the backup time along with underlying hardware.

3.6 Safety Requirements

During message transfer between phone and server, the changing of data will be prevented in order to prevent data inconsistency.

3.7 Security Requirements

Data encryption will be used when data is communicated over the communication channel in order to cater for security concerns.

3.8 Software Quality Attributes

Some of the quality attributes identified includes:

3.8.1 Adaptability

As already mentioned the system should be compatible to devices running Android version 2.3.3 (gingerbread) to Android lollipop. Updating the Android OS version will not have any impact on the working of the system.

3.8.2 Availability

The application will be available all the time, mobile is in working state, server is running and connection is available and up.

3.8.3 Backup

The server's data will be backup if requested. The backup time depends on the size of data.

3.8.4 Capacity

The system shall provide storage up to storage capacity of phone. In case of limited space it should prompt the user.

3.8.5 Compatibility

The application is compatible with windows and android. It is also compatible with their older versions. Minimum version of android should be gingerbread and minimum windows version should be windows Vista.

3.8.6 Competitive edge

Fortify me is providing a multi communication based solution. It is not only incorporating features of other similar apps but is including new mechanisms to make an effective solution.

3.8.7 Compliance

Not applicable

3.8.8 Documentation

A video tutorial and a user manual would be available along with software.

3.8.9 Efficiency

The system takes approximately 2-5 seconds to respond to the user's request depending on the underlying hardware and connection speed. The SMS speed depends on multiple factors like GSM connection and signal strength. The server request and respond depends on the connection speed.

3.8.10 Flexibility

The layout/architecture of the application will be flexible enough so if addition or change can be done if needed.

3.8.11 Interoperability

Semantic interoperability is provided by the system that allows exchange of information between mobile and sever. Also it is operable on different versions of OS.

3.8.12 Maintainability

Whenever there is a change in requirement or bug found, the application will be easily maintainable.

3.8.13 Portability

In API, portability can be defined as "compatibility of application with platform (Android's version) upgraded or downgraded versions. In Android's platform when an up gradation is done, application requires some changes for compatibility with new version. Extensive testing will be required for verification & validation to address compatibility issues. Supported base android version will be Android 2.3.3 with support up to Android L preview.

3.8.14 Reliability

System will have extensive error handling mechanism. Any task if not executed properly or changing state within approx.5-8 seconds will generate an error.

3.8.15 Robustness

If the connection between the user and the system is broken prior to a request being either confirmed or canceled the system shall prompt the user that the connection has been interrupted.

3.8.16 Scalability

Multiple users can generate and receive SOS messages at a time. However one interface of application will run on mobile at a time.

3.8.17 Security

System will be secure as will be encrypted transmission of data during communication.

3.8.18 Stress

The system shall be capable of handling extremely high or stressful load.

3.8.19 Start-up time

Not applicable because there is no specific requirement.

3.8.20 Timeliness

Not applicable

3.8.21 Usability

The user interface of the application will be designed user friendly and self-explanatory. Any android user can easily use it. Although if required minimum training of 5 minutes will be enough for the understanding of the system.

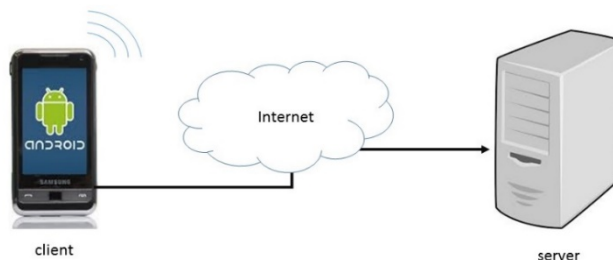
4 Design and Development

4.1 Introduction

This chapter describes Software Design Document for Fortify Me.

4.2 Architectural Model

This section provides a detailed and comprehensive architectural overview of the system. Client



server pattern is followed in which Mobile acts as a client and initiates a message for server.

Figure 4-1: Client Server Distribution

4.2.1 Client

Client is a mobile, it runs applications of Mobile phone by sending GSM SMS and TCP/IP requests to server.

4.2.2 Server

Server records the current location of all the clients continuously and receives the request packets from client and processes them. It stores the locations, calculates the nearest located people and send message to them.

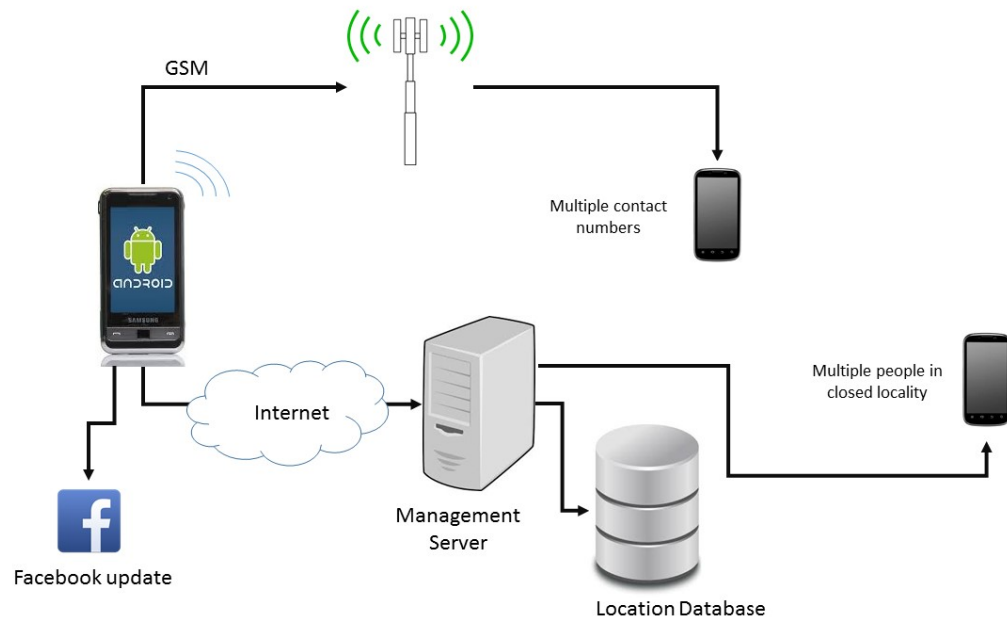


Figure 3.2: Detailed working

4.3 Design Pattern

The design pattern used here is **Facade Pattern**. It provides a unified interface to a set of interfaces in a subsystem and defines a higher-level interface that makes the subsystem easier to use. A facade exposes simplified functions that are mostly called and the implementation conceals the complexity that clients would otherwise have to deal with. In

general the implementation uses multiple packages, classes and function there in. Well written facades make direct access of other classes rare.

In our system the facade class is the main window. It provides a fairly simple interface and delegates the client request to the particular subsystem or class. All the other classes constitute the systems that provide different functionalities. The main window class provides an interface to access and utilize all the functionality of the system.

4.4 Logical View

4.4.1 Flow Chart Diagrams

4.4.1.1 First is the flow chart of client module

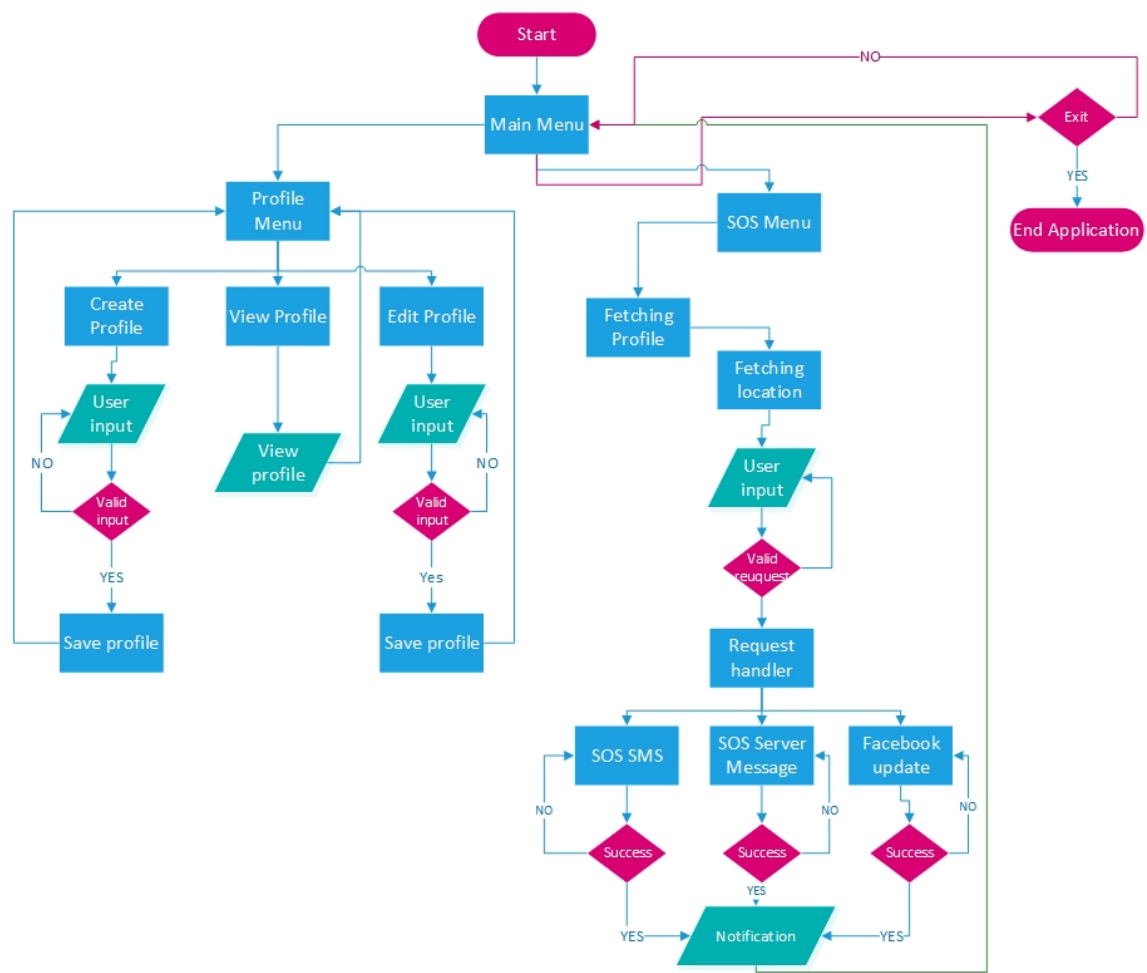


Figure 4.3: Client flow chart

4.2.3

Here is the flow chart of server module

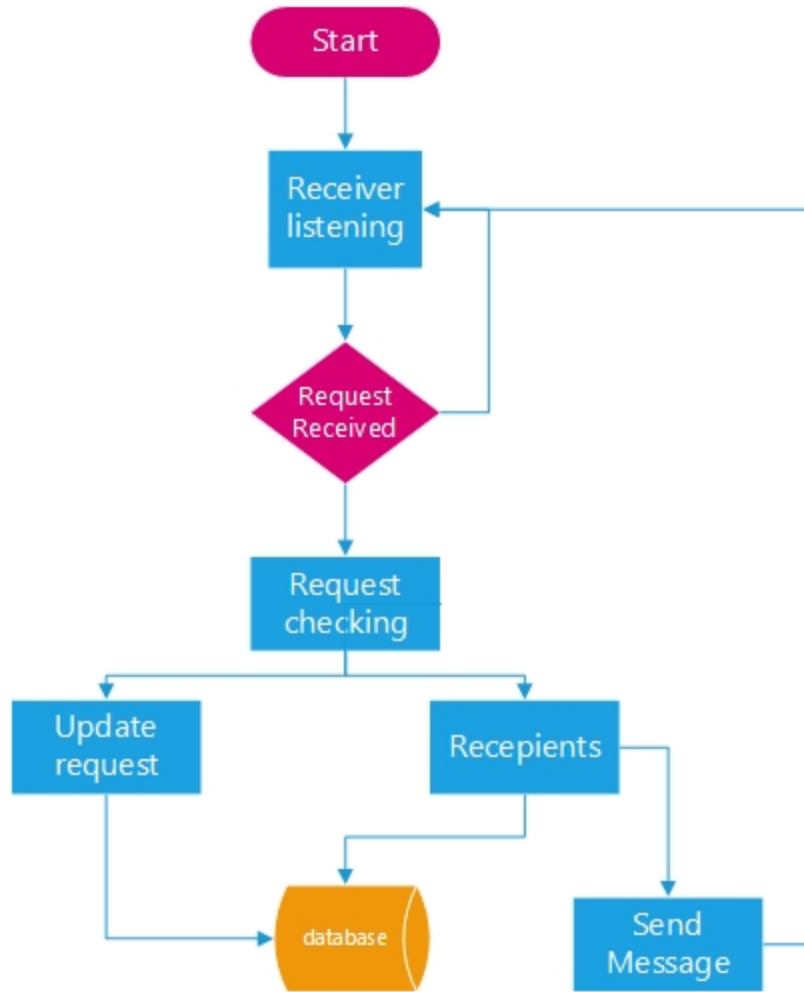


Figure 4.4: Server flow chart

4.4.2 Use Case Diagrams

4.4.2.1 System Use Case Diagram

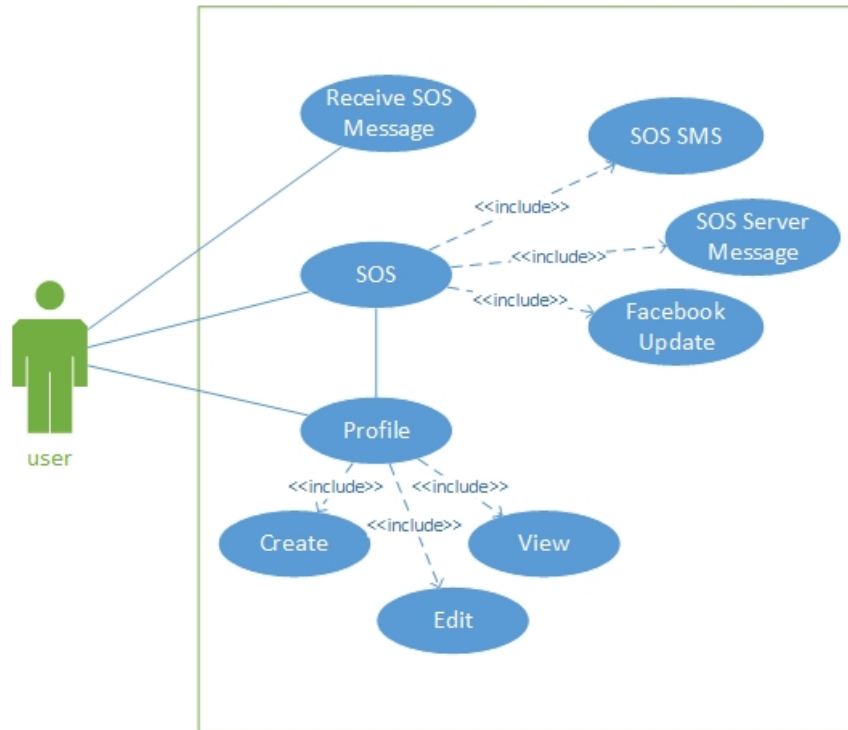


Figure 4.2: System Use Case Diagram

4.4.2.1.1 Create Profile Use Case Diagram

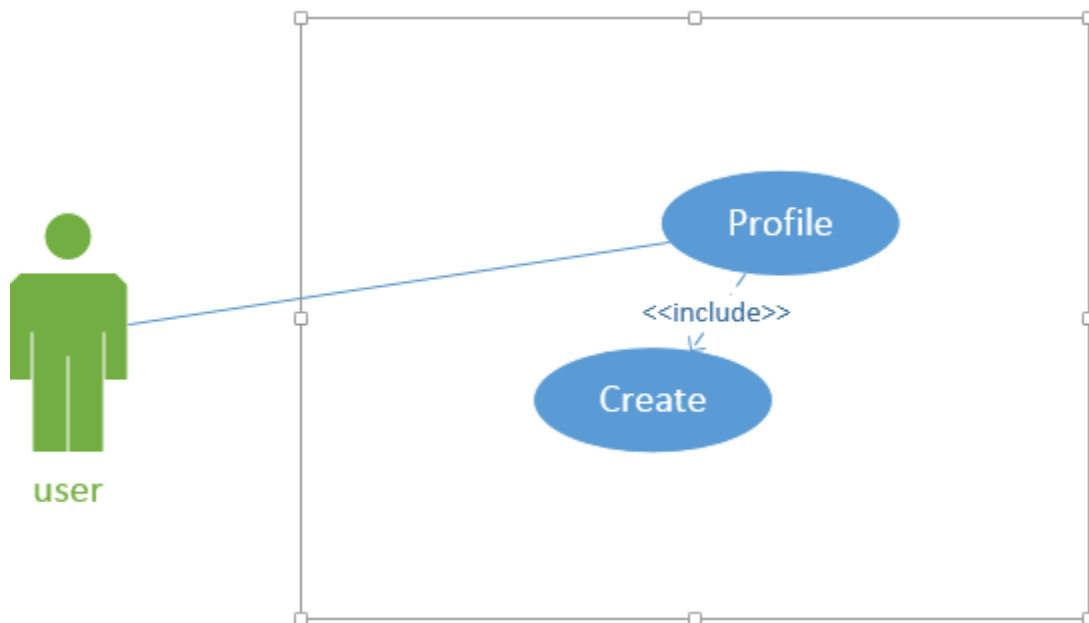


Figure 4.3: Create Profile Use Case

Use Case Description

This use case describes how the user creates a profile on the mobile phone.

Actors

- User.

Pre-Conditions

1. User must have Fortify Me installed on his Mobile.
2. Application should be running
3. User must have some space available to store profile

Basic Flow of Events

1. User starts the application.
2. User inputs personal data, friend and family numbers
3. User enters Facebook details
4. User selects picture

Alternate Flow

- 2- In case of invalid data user must enter data again

Post-Conditions

1. User profile is created.

4.4.2.1.2 Edit Profile Use Case Diagram

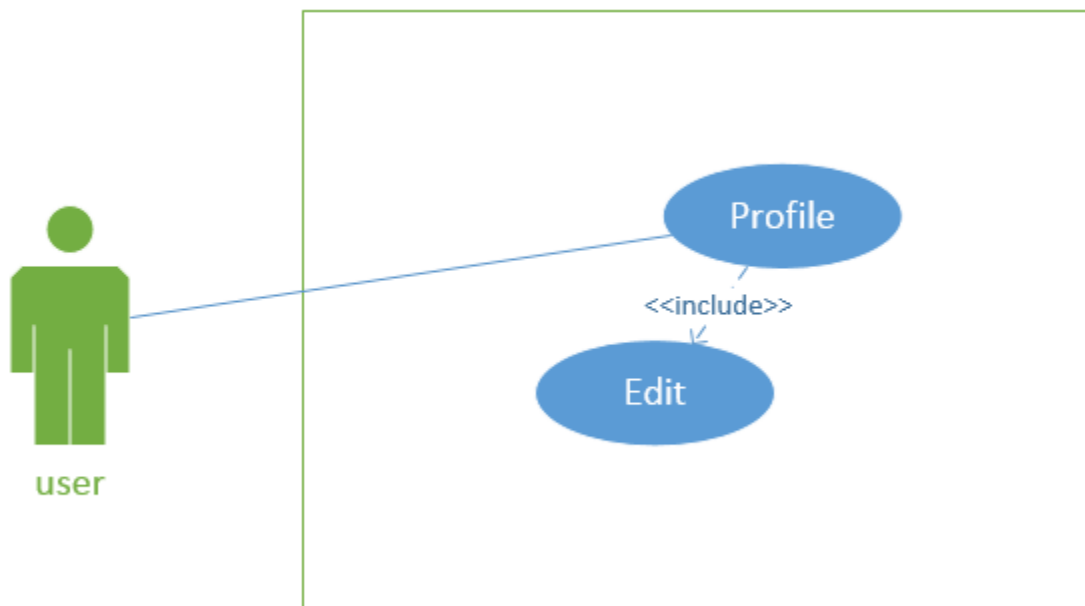


Figure 4.4: Edit profile Use Case

Use Case Description

This use case describes how the user is edit his profile.

Actors

1. User.

Pre-Conditions

1. User must have application installed.
2. Profile has been already created

Basic Flow of Events

1. User starts the application
2. Clicks on edit profile
3. Edits details
4. Presses save button

Alternate Flow

1. If the close button is pressed the system should close the logs module.

Post-Conditions

1. Profile is updated with new details

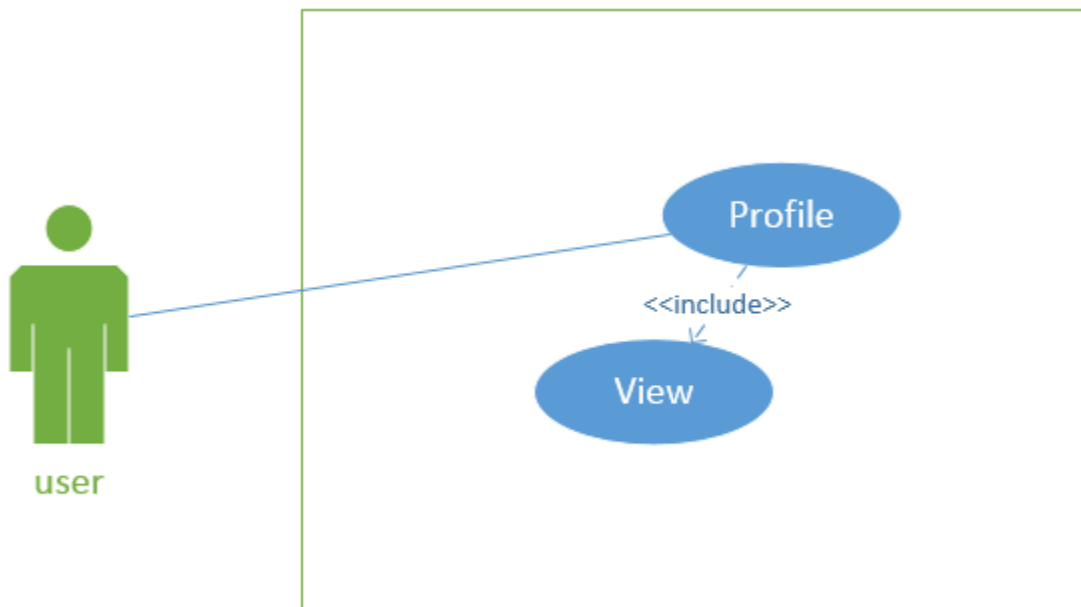
4.4.2.1.3 View Profile Use Case Diagram

Figure 4.5: View Profile Use Case

Use Case Description

This use case describes user views profile.

Actors

1. User

Pre-Conditions

1. User must have application installed.

Basic Flow of Events

1. User starts the application.
2. User clicks on view profile
3. Profile is displayed.

Alternate Flow

1. If the back button is pressed the system should take the user back to main menu.

Post-Conditions

1. Successful Completion

User is able to get the view the profile.

4.4.2.1.4 SOS Use Case Diagram

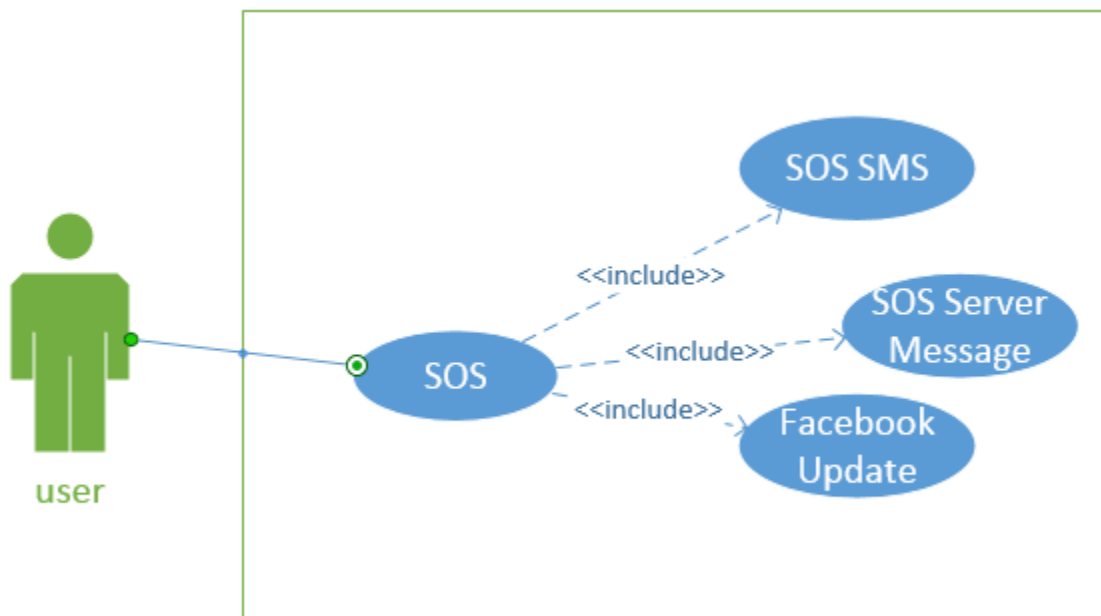


Figure 4.6: SOS Use Case

Use Case Description

This use case describes how the application allows the user to send SOS Message. User will be able to send SMS, server message and also update a Facebook status.

Actors

1. User

Pre-Conditions

1. User must have application installed.

2. User must have Internet.
3. User should have credit in mobile.
4. GSM signals should be available.

Basic Flow of Events

1. User starts the application.
2. User selects the emergency and enters comments.
3. System gets the location.
4. User clicks send button.

Alternate Flow

- 1- If the close button is pressed the system should close application.
- 2- If back button is pressed the app should take user to main menu.

Post-Conditions

- **Successful Completion**
User is able to send the SMS, send server message and update status
- **Unsuccessful Completion**
User is unable to perform the tasks.

4.4.2.1.5 SOS Use Case Diagram

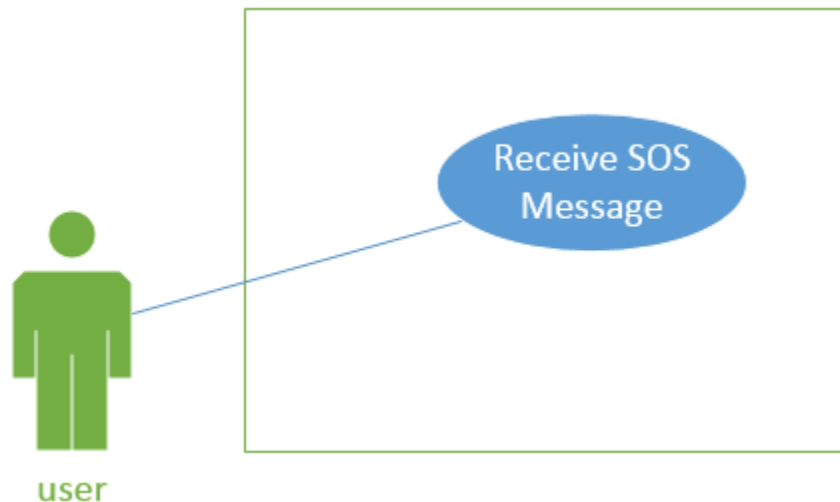


Figure 4.7: Receive SOS Use Case

Use Case Description

This use case describes how the app allows the receive SOS messages sent by other users.

Actors

1. User

Pre-Conditions

1. User must have application installed.
2. User must have Internet available.

Basic Flow of Events

1. User starts the application.
2. User views the pop message shown by application when SOS message is received.

Alternate Flow

1. If the close button is pressed the system should close the pop up.

Post-Conditions

- **Successful Completion**
User is able to view the SOS message

4.4.3 Sequence Diagrams

4.4.3.1 Create Profile

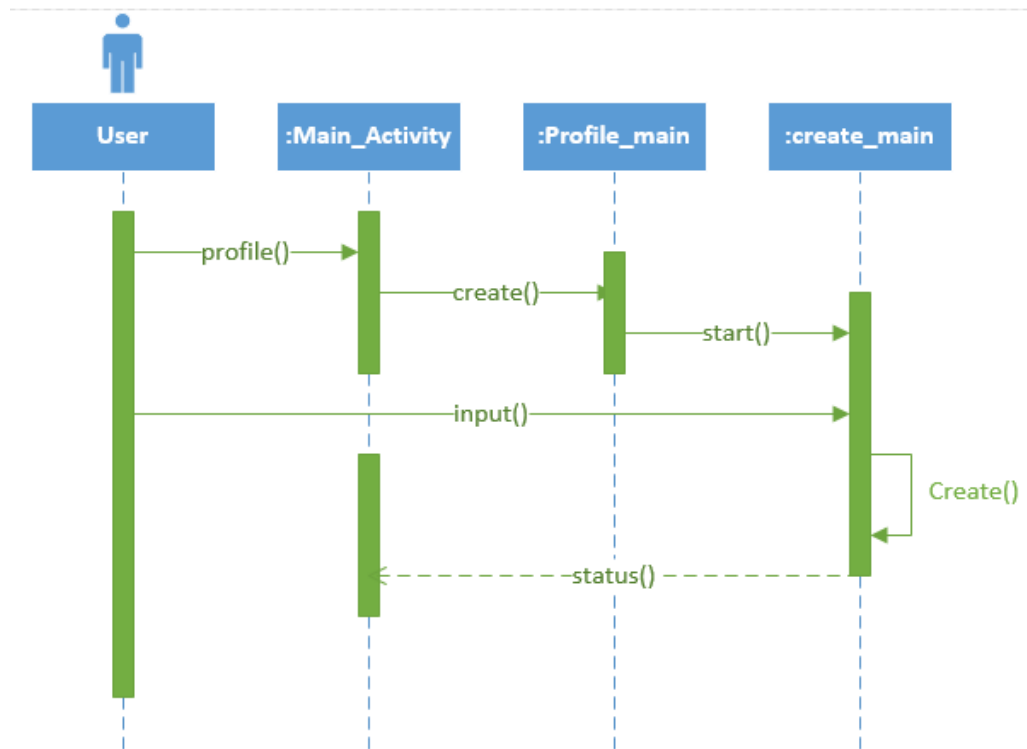


Figure 4.8: Create Profile Sequence Diagram

4.4.3.2 View Profile Sequence Diagram

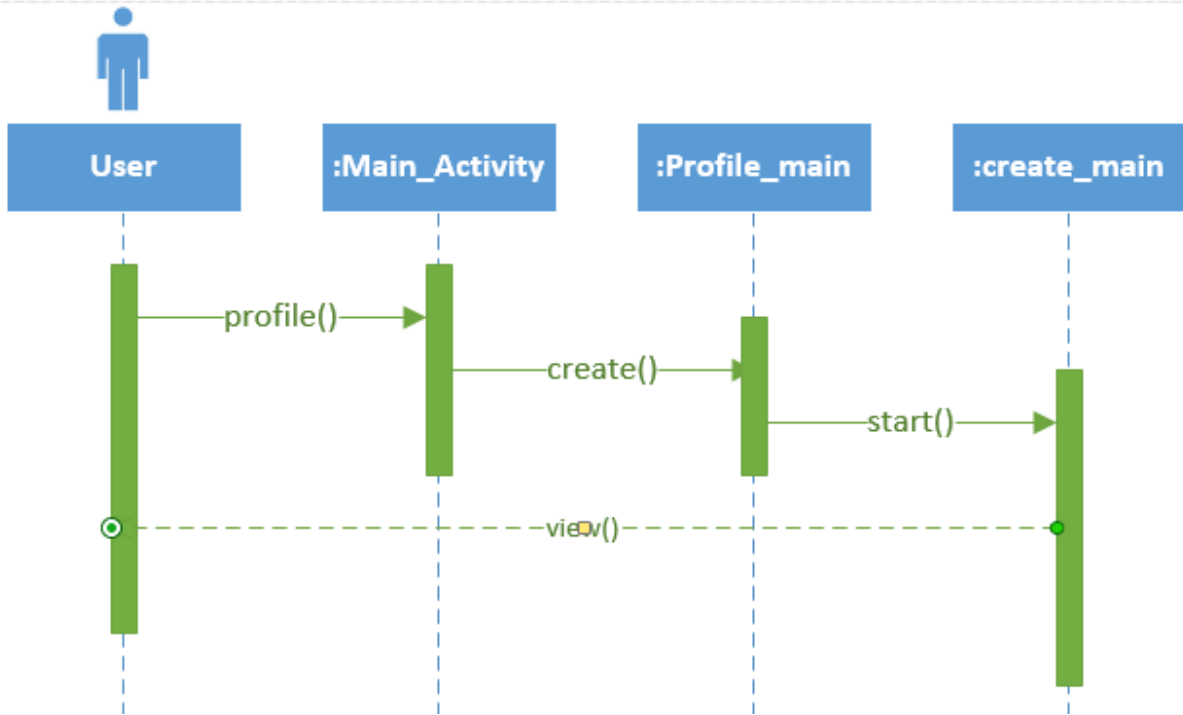


Figure 4.9: View profile Sequence Diagram

4.4.3.3 SOS SMS Sequence Diagram

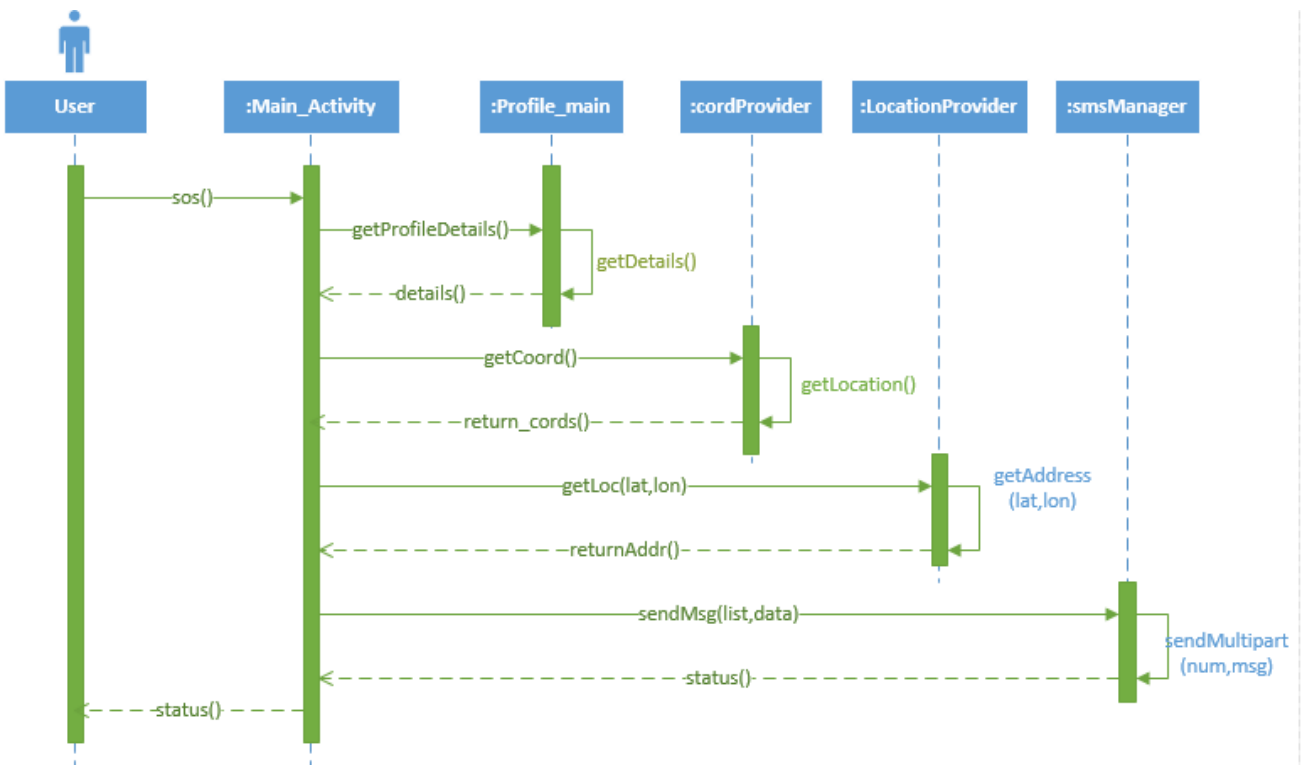


Figure 4.10: SOS SMS Sequence Diagram

4.4.3.4 SOS Server Message Sequence Diagram

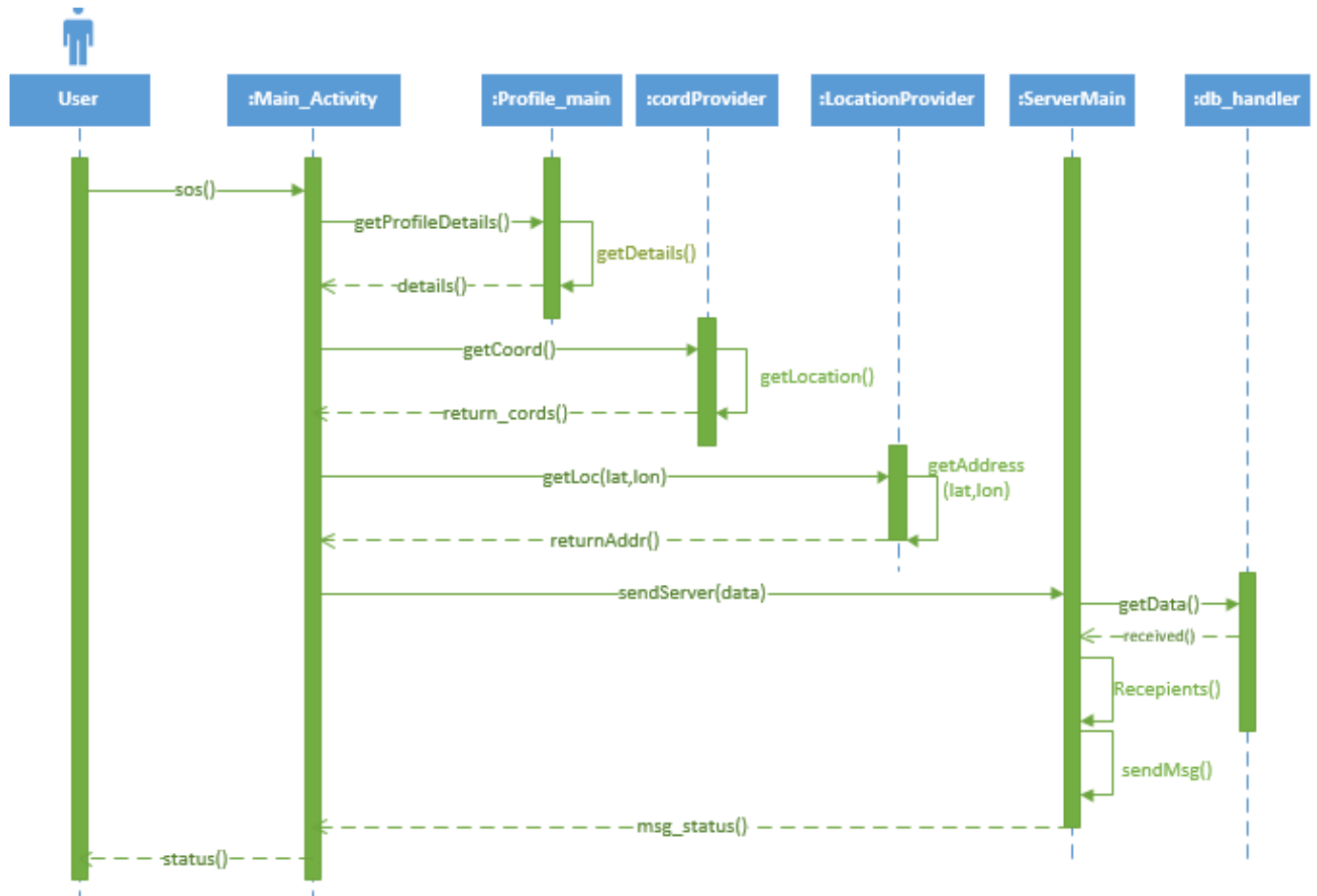


Figure 4.11: SOS Server Message Sequence Diagram

4.4.3.5 Facebook update status Sequence Diagram

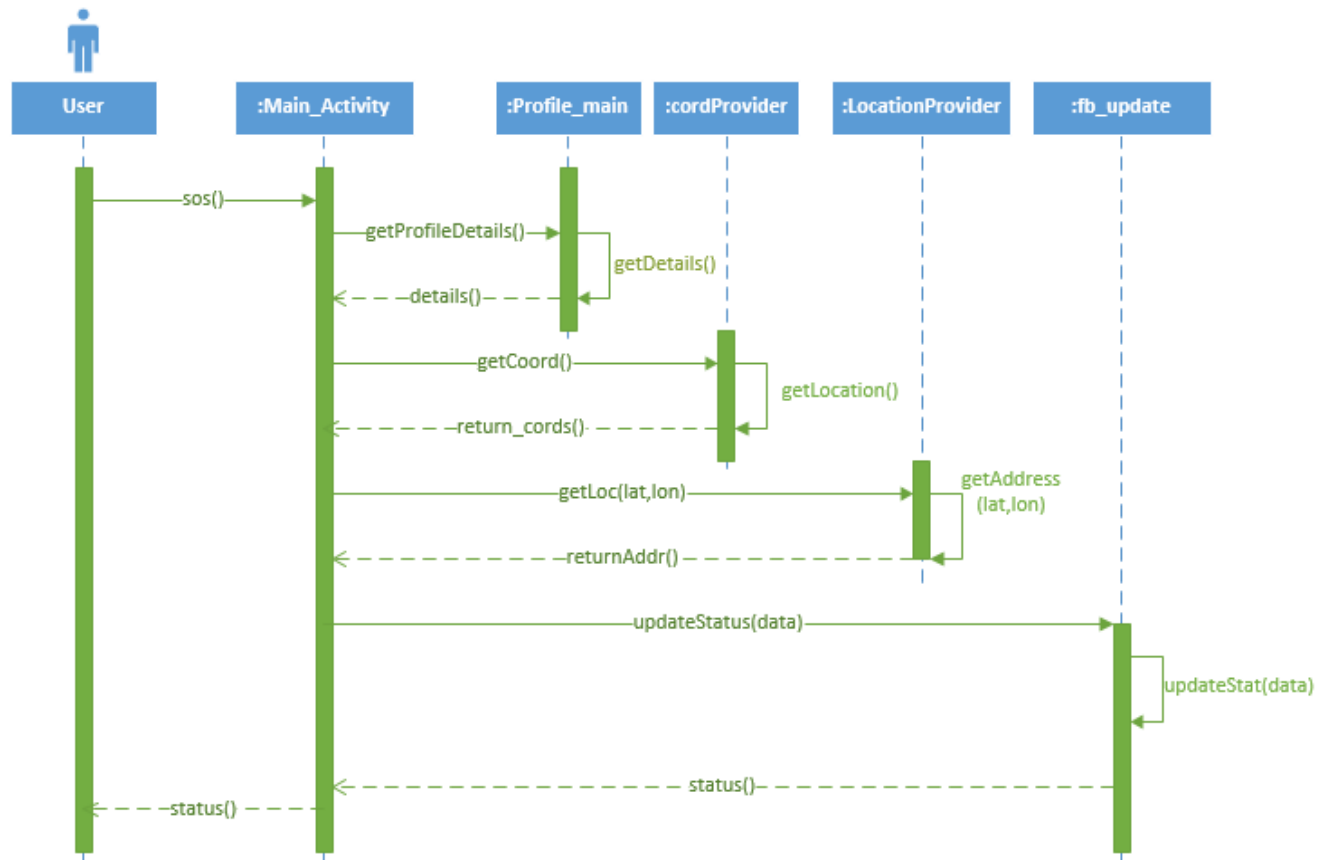


Figure 4.12: Facebook update Sequence Diagram

4.4.3.6 Receive SOS Sequence Diagram

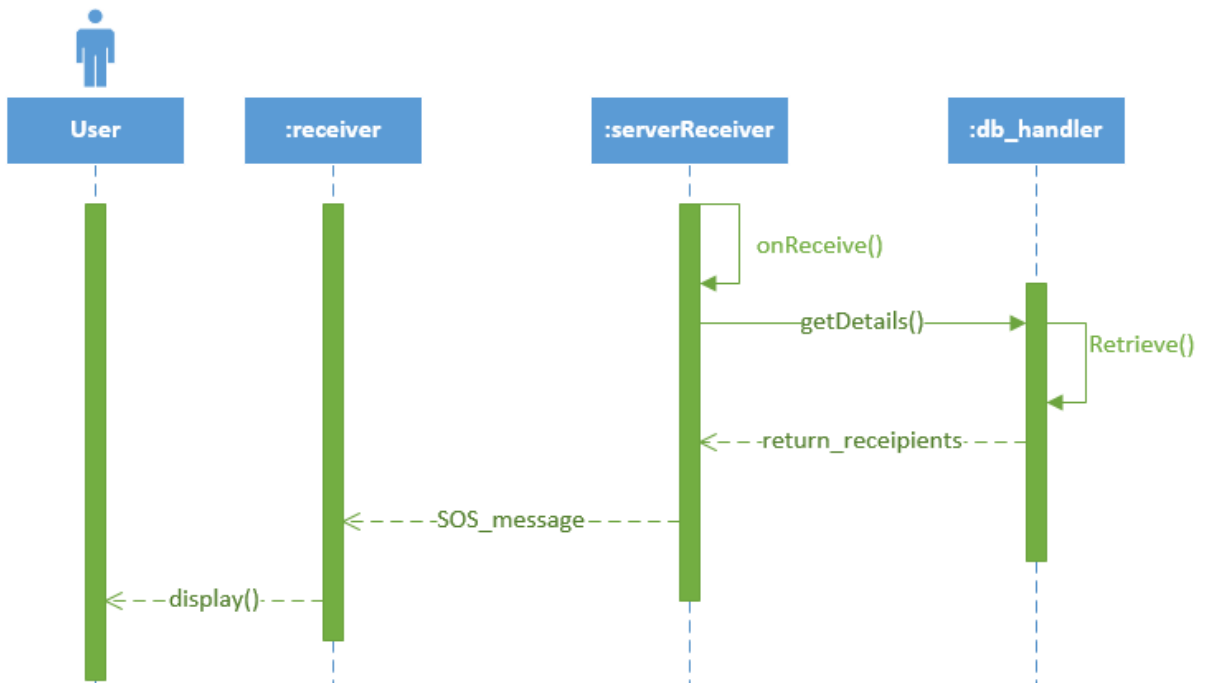
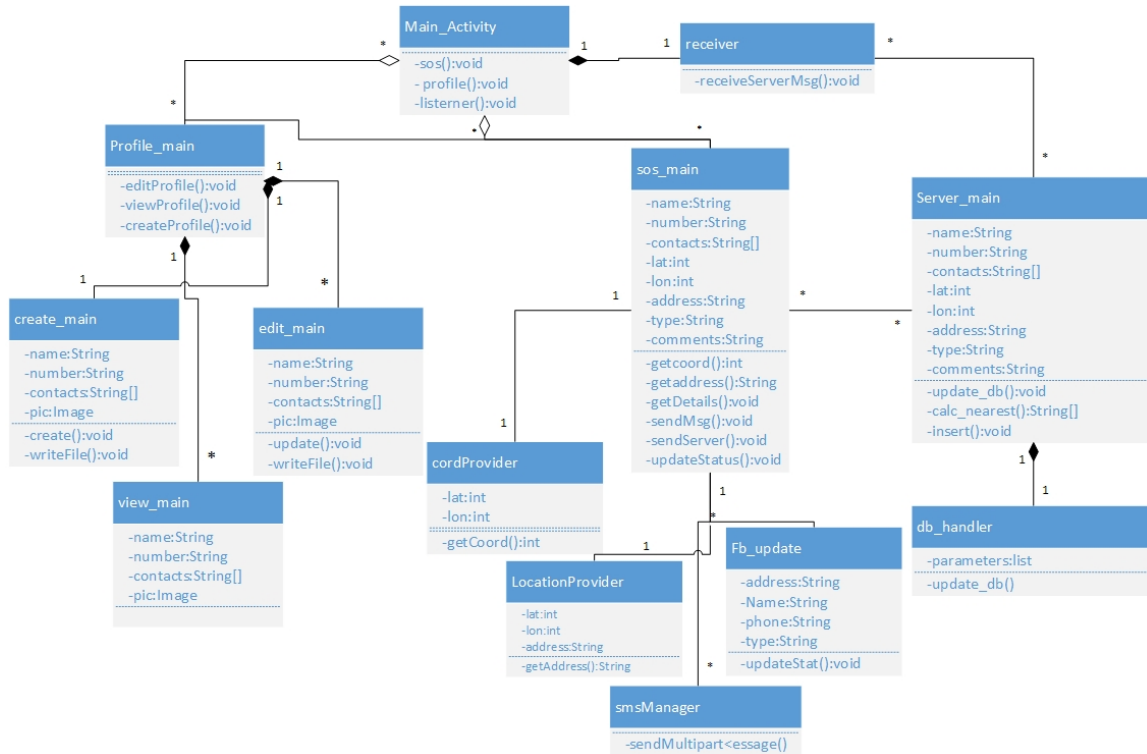


Figure 4.13: Receive SOS Message Sequence Diagram

4.5 Implementation View

4.5.1 System Class Diagram



4.5.2 System Classes Description

4.5.2.1 Main_Activity Class

Name	Main_Activity
Description	This is the main class and contains the user interface of the android app. It invokes all the other features of app and handles the data.
Functions	sos(): Starts the SOS procedure to initiate the process. profile(): Starts the profile handler which gives options regarding the profile listener(): It manages the procedure when SOS message is received.

4.5.2.2 Profile_main Class

Name	Profile_main
Description	This class is responsible for managing the profile options. User can perform

	different functions with his profile. This class allows that.
Functions	editProfile(): It starts the process to allow user to edit his profile viewProfile(): It starts the process to allow user to view his profile createProfile(): It starts the process to allow user to create his profile

4.5.2.3 create_main Class

Name	create_main
Description	This class contains the methods to allow user to create profile.
Functions	create(): It processes the input and creates a profile writeFile(): Writes the data to a text file and stores it to storage.

4.5.2.4 edit_main Class

Name	edit_main
Description	This class contains the methods to allow user to create profile.
Functions	update(): It updates the data. Overwrites previous data with new. writeFile(): Writes the data to a text file and stores it to storage

4.5.2.5 view_main Class

Name	View_main
Description	This class contains the methods to allow user to view profile
Functions	display(): Displays the data into the required format.

4.5.2.6 sos_main Class

Name	sos_main
Description	This class contains the methods and functionality to allow users to send SOS message.
Functions	getcoord(): Retrieves latitude and longitude of current user. getAddress(): Retrieves address from latitude and longitude. getDetails(): Gets profile data. sendMsg(): Sends SMS to required recipients sendServer(): Responsible for sending message to server. updateStatus(): Responsible for updating status.

4.5.2.7 reciever Class

Name	receiver
Description	This class handles the incoming SOS messages.
Functions	receiveServerMsg(): Receives the messages and displays them in proper format.

4.5.2.8 Server Class

Name	Server_main
Description	Handles messages received on server and processes them accordingly.
Functions	Update_db(): Updates the database Calculate_nearest(): Gets values from database and calculates the nearest located people. Also sends them message Insert(): Inserts data in db.

4.5.2.9 DB handler

Name	db_handler
Description	Manages the database

4.6 Dynamic View

4.6.1 Activity Diagrams

4.5.1.1 System Activity Diagram

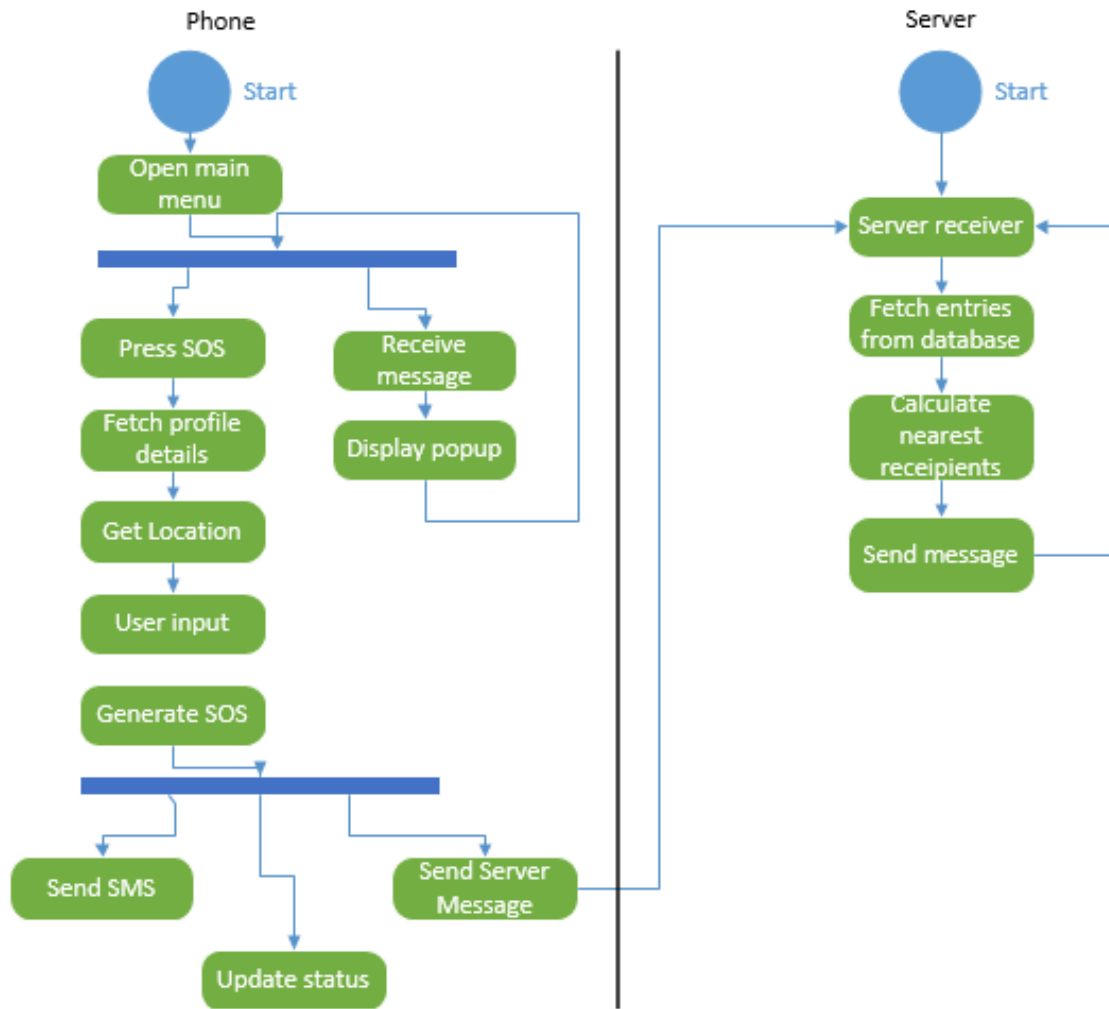


Figure 4.14 System Activity Diagram

4.5.1.2 SOS Message Activity

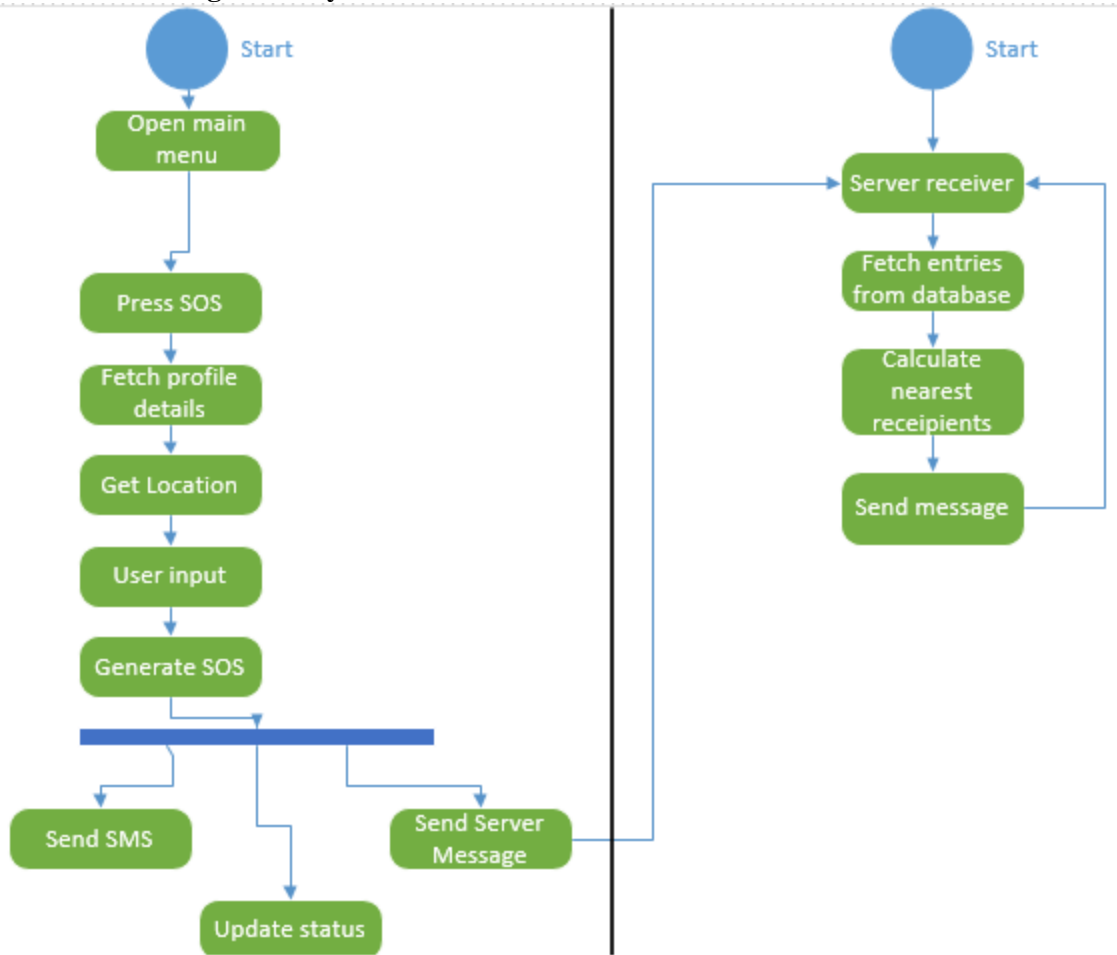


Figure 4.15: SOS Message Activity Diagram

4.5.1.3 Create Profile Activity

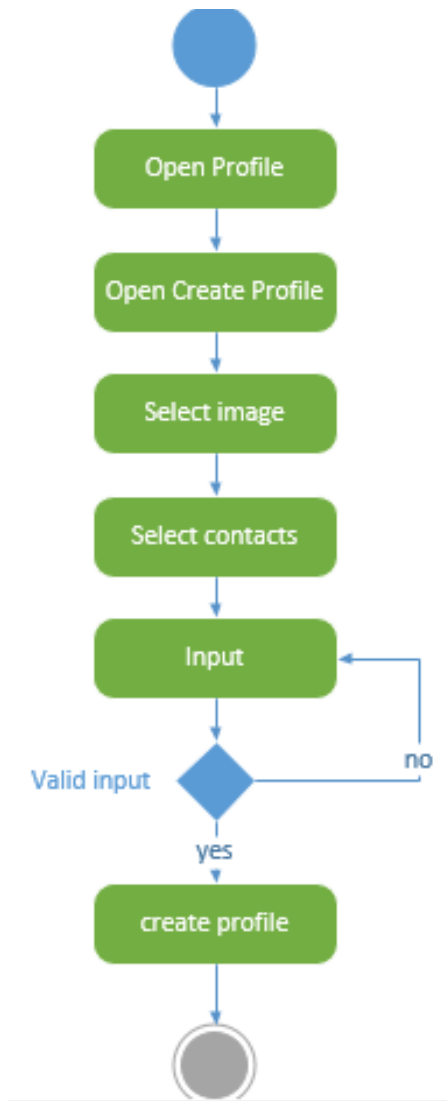


Figure 4.16 Create Profile Activity Diagram

4.5.1.4 Edit Profile Activity

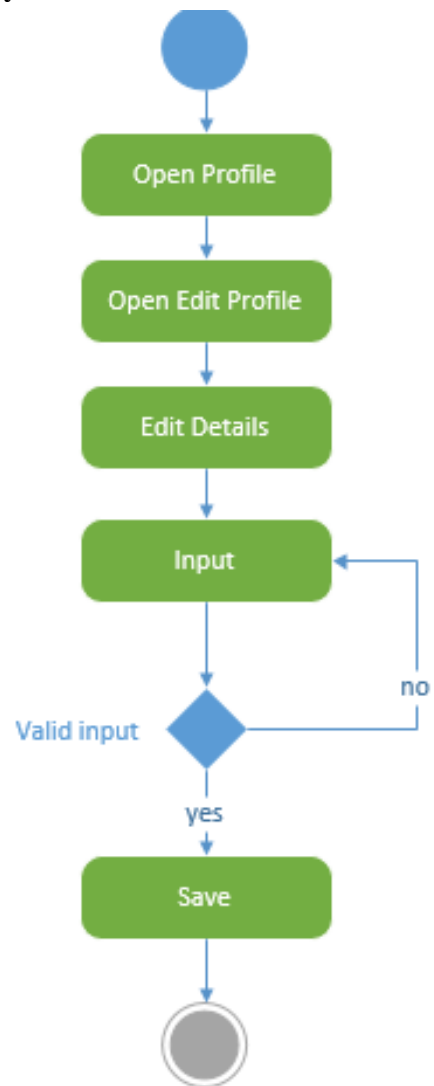


Figure 4.17 Edit Profile Activity Diagram

4.5.1.5 SOS Receive Activity

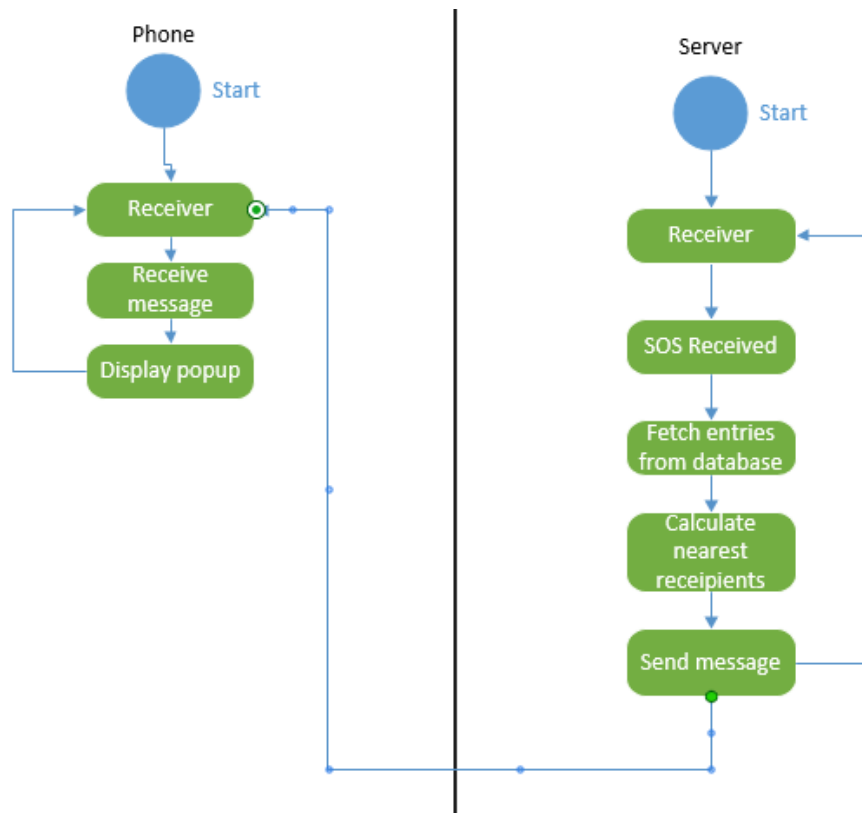


Figure 4.18 SOS Receive Activity Diagram

4.7 User Interface Design

The user interface consists of multiple parts.

The user interface consists of multiple parts.

4.7.1 Main Window

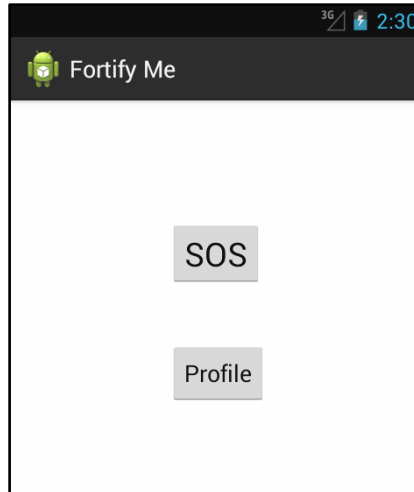


Figure 4.19: Main Menu Interface

The main window appears when the application starts. It includes the options to either go to the profile or start the SOS Process.

4.7.2 SOS Window

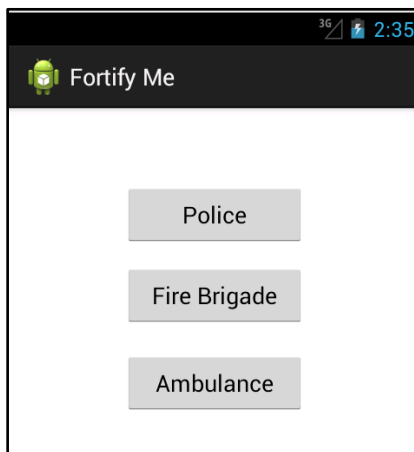


Figure 4.20: Emergency Window

The screen will be used to select emergency type.

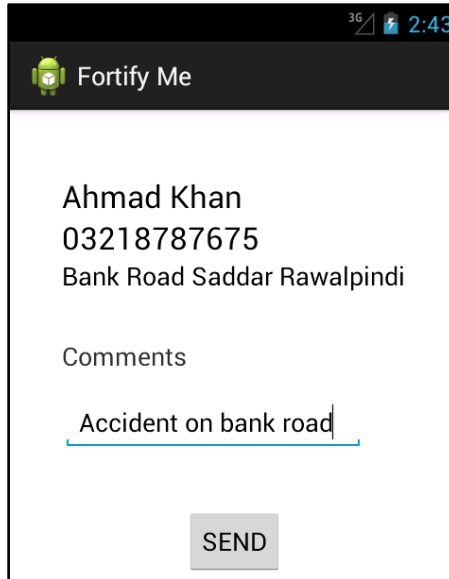


Figure 4.21: SOS Message View

After getting all the details from user profile and location details the user adds comments if he wants to and sends the message.

4.7.3 Profile Window

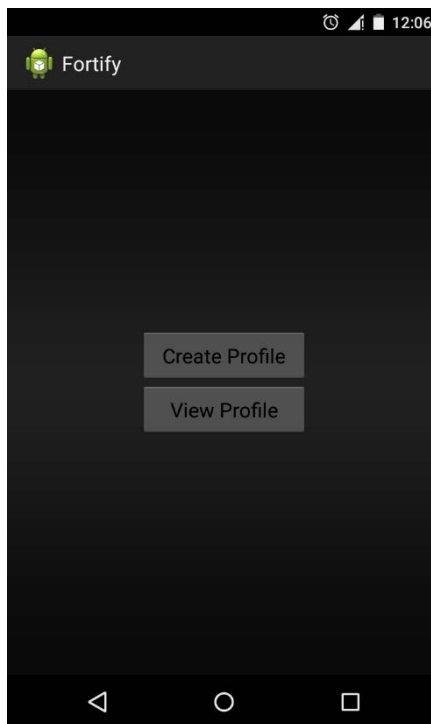


Figure 4.22: Profile Window Interface

The Profile window will be used to create/edit and view profile.

4.7.4 Create / Edit Profile Window

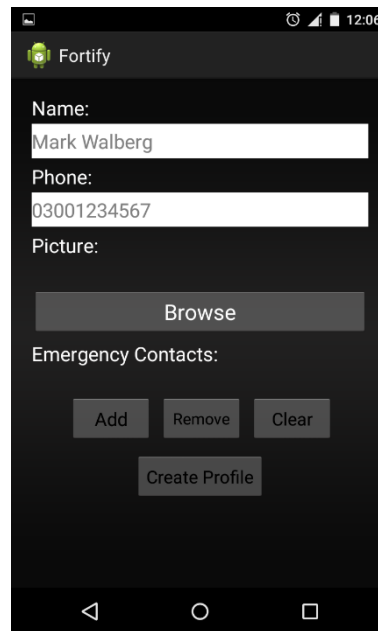


Figure 4.23: create/edit Window Interface

This window will allow user to create and edit his profile. We have all the details present here that the user needs to insert in order to create his profile.

5 System Implementation

3.1 Tools and Technologies

3.1.1 Eclipse SDK

In computer programming, **Eclipse** is an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse can be used to develop both android and desktop applications. By means of various plug-ins, Eclipse may also be used to develop applications in other programming languages like Ada, ABAP, C, C++, COBOL, Fortran, Haskell, JavaScript, Lasso, Natural, Perl, PHP, Python, R, Ruby(including Ruby on Rails framework), Scala, Clojure, Groovy, Scheme,

and Erlang. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

The initial codebase originated from IBM VisualAge. The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules.

then be combined with a Java project by binding the UI to the application's logic.

3.2 Software Implementation

The system follows a client server based architecture so there are two apps running one on the phone and one on PC. Some global variables are needed for effective communication between the two devices. These include the IP addresses of both devices plus the ports that the sockets use for communication.

3.2.1 Profile Creation

This module creates user profile. The information provided in profile is then fetched when SOS is generated. The Profile is stored in a file in the android mobile. It includes various information that is required. The FileStream is used to store data inside a file. The FileStreamReader is used to then read data from file and use it accordingly

3.2.2 SMS implementation

The user enters the message and recipient. The PC sends the request containing the recipient, body and the key of the request. The number and body are then separated. The SmsManager API is being used.

```
smsManager.sendMultipartMessage(Body,recipient);
```

The implementation converts long messages to multiple parts and sends them in multiple messages. It also caters for multiple recipients. SMS is sent to recipients that were provided in the profile. These are selected from contacts.

3.2.3 SOS Server Message

After receiving request for SOS the app sends a SOS message to server as well. The data is fetched from profile and coordinates are gathered along with address. These are then sent to server. Using sockets and HTTP requests these are sent to server where they are forwarded accordingly.

3.2.4 Facebook Status Update

Facebook status is updated as a result of SOS generation. Facebook SDK is required for the integration of Facebook feature. Facebook application is first registered using a profile and then it is embedded with an application code which you get from developers.facebook.com. The application then must be activated to perform all features correctly. Various functions are used in the SDK to get login and update status.

3.2.5 Location Retrieval

Latitude and longitude are gathered from either GPS or network. GPS API is used to gather these coordinates with maximum accuracy. In case GPS is not available network location is used. Network location API is used.

3.2.6 Street Location

Street location is gathered from latitude and longitude. Geocoder API is used to get location address from coordinates

6. Project Analysis and Evaluation

6.1 Testing

6.1.1 Testing Introduction

To ensure quality of the product, testing is conducted. Accuracy and efficiency of tasks performed by our system had to be tested to analyze the system and verify and validate it. Software testing techniques and results obtained are discussed in the coming sections.

6.1.2 Testing Levels

Separate modules were developed to provide different functionalities of the system. All of these modules were tested at different levels during development and after integration. Different levels at which the system has been tested and results obtained are described in this section.

6.1.3 Unit Testing

Each module was developed and tested individually, each and every component was tested at unit level to ensure that they were functioning properly.

6.1.3.1 Profile Creation Module

Pair wise

Use orthogonal array to design test cases:

	Enter profile attributes	Select Picture
Row 1	valid	empty
Row 2	valid	Wrong
Row 3	invalid	Empty
Row 4	Empty	Empty
Row 5	Invalid	Invalid
Row 6	Valid	Valid

Test cases:

<Valid profile attributes and empty picture, profile creation unsuccessful >

<Valid profile attributes and incorrect picture, profile creation unsuccessful >

<Invalid profile attributes and empty picture, profile creation unsuccessful >

<Emptyprofile attributes and empty picture, profile creation unsuccessful >

<Incorrectprofile attributes and validpicture, profile creation unsuccessful>

<Valid profile attributes and correctpicture, profile creation unsuccessful >

Test case ID	Input	Expected output	Actual output	Status (pass/fail)
TC 1	Invalid profile attributes and picture	Profile creation unsuccessful	Profile creation unsuccessful	Pass
TC 2	Valid profile attributes and empty picture	Profile creation unsuccessful	Profile creation unsuccessful	Pass
TC 3	Invalid profile attributes and empty picture	Profile creation unsuccessful	Profile creation unsuccessful	Pass
TC 4	Emptyprofile	Profile	Profile	Pass

	attributes and empty picture	creationunsuccessful	creationunsuccessful	
TC 5	Valid profile attributes and valid picture	Profile creation successful	Profile creation successful	Pass

6.1.3.2 SMS module

Cases:

Case 1: SMS sent successfully

Case 2: SMS not sent successfully

Test cases:

<Valid phone number selected, Case 1 executed >

<Invalid number form contact list selected, Case 2 executed>

Test case ID	Input	Expected output	Actual output	Status (pass/fail)
TC 1	Valid phone number selected	Case 1 executed	Case 1 executed	Pass
TC 2	Invalid number form contact list selected	Case 2 executed	Case 2 executed	Pass

6.1.3.3 Server Message module

Test cases:

Pair wise

Use orthogonal array to design test cases:

	SOS Message Attributes	Enter comments
Row 1	valid	empty
Row 2	invalid	Not empty
Row 3	invalid	empty
Row 4	valid	Not empty

Test cases:

<Valid attributes gathered, comments empty, SOS Sent successfully>

<Invalid attributes, message sending failed>

<Invalid attributes, message sending failed>

<Valid attributes,message sent successfully>

Test case ID	Input	Expected output	Actual output	Status (pass/fail)
TC 1	Valid attributes and empty comments	SOS sent	SOS sent	Pass
TC 2	Invalid attributes and comments	message sending failed	message sending failed	Pass
TC 3	Invalid attributes and empty comments	message sending failed	message sending failed	Pass
TC 5	Valid attributes and comments	SOS sent	SOS Sent	Pass

6.1.3.4 Facebook Update Module

Cases:

Case 1: Facebook Status updated successfully

Case 2: Facebook status not updated

Test cases:

<Valid attributes and data connection available, Case 1 executed >

<Invalid attributes and data connection available, Case 2 executed>

<Valid attributes and no data connection, Case 2 executed>

Test case ID	Input	Expected output	Actual output	Status (pass/fail)
TC 1	Valid attributes and data connection available	Case 1 executed	Case 1 executed	Pass
TC 2	Invalid attributes and data connection available	Case 2 executed	Case 2 executed	Pass
TC 3	Valid attributes and no data	Case 2 executed	Case 2 executed	Pass

	connection available			
--	----------------------	--	--	--

6.1.3.5 GPS Location Module

Cases:

Case 1: GPS coordinates fetched successfully

Case 2: Could not fetch coordinates

Test cases:

<GPS signal strength is high and weather is clear Case 1 executed >

<GPS signal strength is low, Case 2 executed>

<Weather is not clear, Case 2 executed>

Test case ID	Input	Expected output	Actual output	Status (pass/fail)
TC 1	GPS signal strength is high	Case 1 executed	Case 1 executed	Pass
TC 2	GPS signal strength is low	Case 2 executed	Case 2 executed	Pass
TC 3	Weather is not clear	Case 2 executed	Case 2 executed	Pass
TC 4	Weather is clear	Case 2 executed	Case 2 executed	Pass

6.1.3.6 Street Address module

Cases:

Case 1: Gets address successfully from GPS coordinates

Case 2: Unable to get address

Test cases:

<Data Connection available and valid coordinates, Case 1 executed >

<No data connection and Valid coordinates, Case 2 executed>

<Data connection and invalid coordinates, Case 2 executed>

Test case ID	Input	Expected output	Actual output	Status (pass/fail)
TC 1	Data connection available and valid coordinates	Case 1 executed	Case 1 executed	Pass
TC 2	No data connection and valid coordinates	Case 2 executed	Case 2 executed	Pass
TC 3	Invalid coordinates and data connection available	Case 2 executed	Case 2 executed	Pass

6.1.4 Integration Testing

This Test Plan describes the integration tests that will be conducted on the implementation of Fortify Me.

The interfaces between the following modules will be tested:

1. Connection
2. Profile creation
3. SOS SMS
4. SOS Server Message
5. Facebook update
6. GPS Coordinates
7. Street Address
8. Server side

It is assumed that unit testing already provided. Wi-Fi and GPS signals are available.

SYSTEM ARCHITECTURE:

The aim is to check if the modules after integrating with each other perform as they were required. The user of the system should be able to make successful connection after entering valid information and should be able to perform functionalities like profile creation, SOS SMS, SOS server message, Facebook update and recipients calculation. The system performance would be tested under valid and invalid inputs.

Functional Testing will focus on each and every use case that is included in the version currently being worked on. Testing will mainly consist of execution of test cases written to address the gap identified. It will focus on inputs, outputs and system changes due to the actions.

1. User can create profile
2. SOS SMS
3. SOS Server Message
4. Facebook Update
5. GPS Location.

For this purpose we are using the incremental approach to integrate the modules together.

Establish connection between the client and server:

First of all we will integrate two basic modules. Client application (Mobile module of the application) with server (Server module of the application) to ensure that client is successfully making connection with server so that requests could be handled. We will first perform the test cases to validate their interaction. We will make sure that PC module is successfully connecting to mobile module.

Data transfer:

After this we can start integrating the modules related to transfer of data between mobile and Server.

First we will integrate connection with the main menu and then with the initiate call module and make sure that it provides the correct results. We will then integrate it with the Profile creation, integration with the already integrated modules works fine and doesn't disturb their performance as well and then SOS SMS and Server Message are integrated. Performance check is done then. Then Facebook status update is added as a final module. All these are run parallel to server module.

The advantage is that the integration testing process can start earlier, as soon as related modules have been successfully unit tested. An incremental approach makes it easier to find errors since the application environment introduces only one (or a few) modules at a time. Finally, this approach results in more overall testing, since the earlier modules get tested repeatedly as you add new modules.

6.1.4.1 Structure of Integration levels:

Integration test phases:

a. First phase of testing

In this phase tests will be designed to uncover functional errors in each module after it is integrated with the system. Errors associated with local or global data structures will be uncovered with such tests.

We will integrate the connection module with SMS module which sends SOS SMS.

We then integrate Server SOS Message which sent messages to our Server which

does the recipients calculation and forwards the message. Facebook status message is added last when all the other functionality has been done.

Each new module is added after successful debugging from errors and extensive testing. The performance variables are tested any invalid output is corrected and then it is moved forward to the next phase. The testing is incremental so that each module is tested thoroughly and is in accordance.

b. Second phase of testing

Pair wise validity will be performed in which we will combine two modules together and check the functionality which is dependent on both modules. First profile is created and SMS module tested pair wise. Then SMS and server Message tested pair wise.

Then Facebook update is tested along with server message. At the end of the phase all modules are tested as a whole to check the overall functionality of system as a whole so that they confirm with the requirements.

c. Third phase of testing

In third phase endurance testing will be performed that how much stress our system can with stand i.e. How many messages can be sent. Are all messages correctly sent etc.

6.1.4.2 Modules to be integrated in each phase:

a. First phase

In first phase we need to check functional validity of all the modules. So, we will use all of our modules in this phase.

b. Second phase

In end to end tests. We will utilize all of our modules to perform the task starting from one of the module to the module that accepting information from other modules.

c. Third phase

For checking the stress the system can withstand we will again require our whole integrated system in place. This phase validates that critical functions will meet production performance requirements during peak transaction volumes.

6.1.4.3 Building process and Schedule for each phase:

a. First phase

We started first phase on 30th April and checked for functionality of each module after integration and it took almost 4 hours so ended at the same day

b. Second phase

The second phase checks end to end validity. It started on 1st may 2015 after the 1st phase. It will be completed in 7 hours.

c. Third phase

The endurance testing is the last phase in integration testing. It started on 2nd may 2015 .System endurance will be checked by testing it for 4 continuous hours under different conditions and for valid and invalid inputs.

6.1.4.4 Environment to be set up and resources required in each phase:

Make sure that computer is running Windows with Wi-Fi signals available

- 64/32 bit windows OS
- Testers and developers set up manual and configuration guides
- Wi-Fi signals
- Android phone

6.1.4.5 Criteria for each Integration test phase n:

a. First phase:

i. Entry criteria:

Entry criteria for functional testing is that each module has been unit tested individually and all errors have been corrected. Interface integrity ensures that the two modules communicate with each other in the desired way.

ii. Exit criteria:

All modules are integrated incrementally. Exit criteria for the first phase is fully functionally tested modules integrated together. In the end we get a successfully *functionally* tested integrated system. The integrated system has passed all the integration tests and no defect is outstanding.

iii. Integration Technique to be used:

Incremental technique

iv. Test configuration set-up:

Normal test conditions are required (computers and environment with normal temperature).

b. Second phase:

i. Entry criteria:

Functionality of each module is tested after integrations and validated, is Entry criteria for End to end and pair wise.

ii. Exit criteria:

Fully integrated system with no more modules to integrate is Exit criteria .we get a, All the system integration tests for the phases have passed, all the defects found have been fixed and also documented.

iii. Integration Technique to be used:

Incremental technique

iv. Test configuration set-up:

Normal test conditions are required (computers and environment with normal temperature).

c. Third phase:

i. Entry criteria:

The entry criteria of this phase is that we have a validated system that works together from end to end. There are no major defects remaining while testing the end to end working of the integrated system.

ii. Exit criteria:

The exit criteria of this phase is that we have an integrated system to which when the load is applied can sustain the load and does not crash.

iii. Integration Technique to be used:

Incremental technique.

iv. Test configuration set-up:

Normal test conditions are required (computers and environment with normal temperature).

6.1.4.6 Test specification for each integration phase:

a. First phase:

This phase will include performing tests on integrated modules to check their interface integrity and functional validity. Same test cases as used in unit testing for modules functionality are tested again by replacing stubs with actual modules. These are given below:

i. Interface Integrity

Test Case Id	Input	Initial conditions	Test procedure	Expected Result	Actual Result	Status
TC1	Invalid Profile Inputs	Application has prompted to enter correct profile details	Inputs are provided using mobile keyboard	Display error message	Error message	Pass
TC2	Valid profile inputs	Application has prompted to create profile	User presses create profile button	Main menu screen shown and user prompted	Main menu screen shown	Pass
TC3	Generate SOS	Secondary Menu shown	User sees profile and enters comments and emergency type	Window showing keyboard,	Window showing keyboard,	Pass
TC4	Facebook Status window	Facebook status prompt	Enter status	Status updated along with other details	Status updated along with other details	Pass

b. Second phase:

This will include designing test cases to check end to end validity. In our system this can be checked by designing test cases that will start from getting user's credentials from the login screen till the result is retrieved.

For Pair Wise:

Test Case Id	Input	Initial conditions	Test procedure	Expected Result	Actual Result	Status
TC1	Invalid Profile Inputs	Application has prompted to enter correct profile details	Inputs are provided using mobile keyboard	Display error message	Error message	Pass
TC2	Valid profile inputs	Application has prompted to create profile	User presses create profile button	Main menu screen shown and user prompted	Main menu screen shown	Pass
TC3	Generate SOS	Secondary Menu shown	User sees profile and enters comments and emergency type	Window showing keyboard,	Window showing keyboard,	Pass
TC4	Facebook Status window	Facebook status prompt	Enter status	Status updated along with other details	Status updated along with other details	Pass

For the End to End:

For end to end testing we will perform a test which will include almost all the modules so that it can be checked that they all are working as required together.

We will start from user input code, then main menu will be displayed and after that all the features are checked for all functionalities from main menu.

c. Third phase:

This will include designing test cases to check system endurance:

Due to high availability, load test carries great importance. The metrics are as follows:

Metrics	Minimum	Good	Excellent
Maximum Number of Concurrent Operations Supported	1	5	8

Endurance Test cases:

Test Case Id	Input	Initial condition	Test Procedure	Expected Result	Actual Result	Status
TC1	4 days	System is executed	System functions are checked for 4 days without termination	System works properly, no unexpected errors with condition that host machine is running normally.	System works properly, no unexpected errors with condition that host machine is running normally.	Pass
TC2	1 week	System is executed	System functions are checked for 1 week without termination	System works properly, no unexpected errors with condition that host machine is running normally.	System works properly, no unexpected errors with condition that host machine is running normally.	Pass

4.2.3.1 Actual test results for each integration test phase:

The Actual results are already shown in the test cases specifications for each integration phase above.

6.1.5 System Testing

System testing was performed at the end of development and integration of all the components. The system testing enabled us to detect bugs and defects in our code as well as assembly of the system and improve the system by removing a maximum number of them. The current library of test cases helps us to keep development teams on track and also verify that system conforms to requirements which are described in the requirement document.

6.1.5.1 Basic Tests

a. Boot tests:

Test Case Id	Input	Initial conditions	Test procedure	Expected Result	Actual Result	Status
TC1	System launched by clicking button on system file that was saved in memory	Integrated and Working Version is Available	The exe file for the built system is clicked as it is the available boot option provided	The system boots up its image from memory	System gets its software image	Pass

6.1.5.2 Functionality Tests

a. Communication System Tests:

The System requires that both devices be connected with a same Wireless Access Point.

b. Module tests:

Same as done in integration testing.

c. GUI tests:

Test case id	Input	Initial condition	Test procedure	Expected output	Actual output	Status
TC1	Users checks the system's GUI	System has executed and screen is opened	When screen appeared. testing team checks whether the font is readable and text is interpretable	Team should find the GUI user friendly	Team found the GUI user friendly	Pass
TC2	Users calculate the system navigate time	System has executed and screen is opened	User navigates through whole GUI	The navigation time should not be more than 3 to 4 seconds	The navigation time is 3 seconds	Pass

4.2.3.2

d. Security Tests

i. Availability and Integrity tests:

Test Case Id	Input	Initial conditions	Test procedure	Expected Result	Actual Result	Status
TC1	Valid code input of user	Integrated and Working Version is Applicable	Valid code for the user will be provided to the system	Information validated user will be able to view main menu	Information validated and user will be able to view main menu	Pass

ii. Confidentiality tests:

Test Case Id	Input	Initial condition	Test Procedure	Expected Result	Actual Result	Status
TC1	Invalid code	System is executed	Invalid code entered	Error message of invalid code displayed and Access not allowed	Error message of invalid code displayed and Access not allowed	Pass

6.1.5.3 Scalability Tests

Not Applicable

6.1.5.4 Performance Tests

a. INTEROPERABILITY AND COMPATIBILITY TESTS:

Test case id	Input	Test procedure	Expected output	Actual output	Status
TC1	Windows test	Execute the software on windows7/8 and check every functionality that the system is compatible with windows platform	System should work on that environment	System works on that environment	Pass
TC2	Android test	Execute the software on android ICS, jellybean, lollipop	System should work on that environment	System works on that environment	Pass

6.1.5.5 Robustness Tests

a. Boundary value tests:

Not applicable.

b. Power cycling tests:

Test Case Id	Input	Initial condition	Test Procedure	Expected Result	Actual Result	Status
TC1	Power glitch	System is executed	System checked that after power glitch any error occurred on not	Connection will break and need to restart the system	Connection will break and need to restart the system	pass

c. High-Availability Tests:

Not applicable

6.1.5.6 Documentation Tests

Test Case Id	Input	Initial condition	Test Procedure	Expected Result	Actual Result	Status
TC1	User manual is viewed by testing team	System is not executed	Team reads the manual and execute the system side wise	System should be successfully executed and used by testing team	Team successfully used the system	pass

6.1.6 Acceptance Testing

This is acceptance test plan for Fortify Me. By testing we want to achieve that the system matches the customer requirements and is acceptable by them. This is the first version of the software, to provide quick SOS in case of emergency

Date of approval: 5nd May 2014.

6.1.6.1 Functional Correctness and Completeness

Operational Environment:

The operational environment required to test the software is the operating system it requires. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windows xp/7/8.

Test Case Specifications:

Test Case ID	Title	Objective	Procedure
TC 1	Connection In Feature	Is to test that the authentication feature's functionality is according to what user requirement is and do the work it is programmed to do	Run the software and provide the correct code ,if the menu appears,it will show that the test is successful
TC 2	SOS SMS	Is to test that SOS SMS is sent to the required recipients	Establish connection with the module and send SOS SMS to recipients with required details that are needed to be sent
TC 3	SOS Server Message	Is to test that Server Message is sent to the Server and then to the calculated recipients	Establish Connection that SOS Server message is sent to server over internet and then to the nearest calculated recipients.
TC 4	Facebook Update	Is to test that facebook status is updated with required information	Establish Connection facebook status is updated along with particular details
TC 5	Location Retrieval	Is to test that GPS coordinates are gathered	Establish Connection coordinates are gathered using GPS or network
TC6	Address calculation	Is to test that street address can be taken from coordinates	Establish Connection street address is gathered from coordinates

6.1.6.2 Accuracy

Operational Environment:

The operational environment required to test the software is the operating system it requires. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windowsxp/7/8.

Test Case Specifications:

Test Case ID	Title	Objective	Procedure
TC 1	Connection In Feature	Is to test that the authentication feature's functionality is according to what user requirement is and do the work it is programmed to do	Run the software and provide the correct code ,if the menu appears, it will show that the test is successful
TC 2	SOS SMS	Is to test that SOS SMS is sent to the required recipients	Establish connection with the module and send SOS SMS to recipients with required details that are needed to be sent
TC 3	SOS Server Message	Is to test that Server Message is sent to the Server and then to the calculated recipients	Establish Connection that SOS Server message is sent to server over internet and then to the nearest calculated recipients.
TC 4	Facebook Update	Is to test that facebook status is updated with required information	Establish Connection facebook status is updated along with particular details
TC 5	Location Retrieval	Is to test that GPS coordinates are gathered	Establish Connection coordinates are gathered using GPS or network

6.1.6.3 Data Integrity

Operational Environment:

The operational environment required to test the software is the operating system it requires. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windows xp/7/8.

Test Case Specifications:

Test Case ID	Title	Objective	Procedure
TC 1	Profile	To test the data preservation i.e. Profile is saved and remains unchanged	Establish Connection then Profile is validated and created and stored. The profile is then accessed when needed. The profile data is required using SOS message because data is gathered from it

4.2.4

6.1.6.4 Backup and Recoverability

Operational Environment:

The operational environment required to test the software is the operating system and database server, it requires. As it is not hardware related software so it only requires a computer and operating system to be executed i.e. Windows 7/8. Backup call records and application is generated and is stored in a separate 1tb hard disk

Test Case Specifications:

Test Case ID	Title	Objective	Procedure
TC 1	Data backup	To test that if system fails or shuts down the data backup is save.	Establish Connection profile remains saved.

6.1.6.5 Usability

Operational Environment:

The operational environment required to test the software is the operating system. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windows 7/8.

Test Case Specifications:

Test Case ID	Title	Objective	Procedure
TC 1	Connection In Feature	Is to test that the authentication feature's functionality is according to what user requirement is and do the work it is programmed to do	Run the software and provide the correct code ,if the menu appears, it will show that the test is successful
TC 2	SOS SMS	Is to test that SOS SMS is sent to the required recipients	Establish connection with the module and send SOS SMS to recipients with required details that are needed to be sent
TC 3	SOS Server Message	Is to test that Server Message is sent to the Server and then to the calculated recipients	Establish Connection that SOS Server message is sent to server over internet and then to the nearest calculated recipients.
TC 4	Facebook Update	Is to test that facebook status is updated with required information	Establish Connection facebook status is updated along with particular details
TC 5	Location Retrieval	Is to test that GPS coordinates are gathered	Establish Connection coordinates are gathered using GPS or network
TC6	Address calculation	Is to test that street address can be taken from coordinates	Establish Connection street address is gathered from coordinates
Test Case ID	Title	Objective	Procedure
TC 1	Connection In Feature	Is to test that the authentication feature's functionality is according to what user requirement is and do the work it is programmed to do	Run the software and provide the correct code ,if the menu appears, it will show that the test is successful
TC 2	SOS SMS	Is to test that SOS SMS is sent to the required recipients	Establish connection with the module and send SOS SMS to recipients with required details that are needed to be sent
TC 3	SOS Server Message	Is to test that Server Message is sent to the Server and then to the calculated recipients	Establish Connection that SOS Server message is sent to server over internet and then to the nearest calculated recipients.

6.1.6.6 Performance**Operational Environment:**

The operational environment required to test the software is the operating system it requires. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windows XP/7/8.

Test Case Specifications:

Test Case ID	Title	Objective	Procedure
TC 1	Display response time	To test the systems displaying response time according to the customer criteria	Establish connection then main menu appears click on any of features if response time is according to defined then test successful
TC 2	Opening response time	To test the systems opening time according to the customer criteria	Open the software and check the response time until code screen shows up.

6.1.6.7 Robustness

Operational Environment:

The operational environment required to test the software is the operating system it requires. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windows 7/8

Test Case Specifications:

Test Case ID	Title	Objective	Procedure
TC 1	input screen	Determine how the system reacts when erroneous inputs are given.	Invalid input generates message
TC 2	Send SMS	Determine how the system reacts when erroneous inputs are given.	Invalid criteria results in SMS not sent
TC 3	Send Server Message	Determine how the system reacts when erroneous inputs are given.	Invalid criteria results in Message not sent

6.1.6.8 Confidentiality

Operational Environment:

The operational environment required to test the software is the operating system it requires. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windows 7/8.

Test Case Specifications:

Test Case ID	Title	Objective	Procedure
TC 1	authentication	Determine the protection of data from Server to connecting with phone	Start the system. The communication is encrypted and secure between server and phone. Security of profile is also considered as it will not be accessible by other apps. Only our app can access the profile.

4.2.5

6.1.6.9 Compatibility and interoperability

Operational Environment:

The operational environment required to test the software is the operating system it requires. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windows 7/8.

Test Case Specifications:

Test Case ID	Title	Objective	Procedure
TC 1	Windows test	To test whether or not system work on the windows	Execute the software on windows 7/8 and check every functionality that the

			system is compatible with windows platform system works on all of these so test successful
TC 2	Android test	To test whether or not system work on the platform keep windows 8 and change android	Android OS version 2.3 ,3.0, 4.0, 4.1, 4.4, 5 system works on all of these so test successful

6.1.6.10 Load Testing

Operational Environment:

The operational environment required to test the software is the operating system it requires. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windows 7/8.

Test Case Specifications:

The software is not a server base software so maximum no. of users that can operate it at a time is one and all the test cases described will also check for the load testing as it will be running on maximum load.

6.1.6.11 Schedule:

The system is medium and no. of test cases are acceptable so it will take one to two weeks for the acceptance testing to complete.

6.1.6.12 Human Resource:

The following testers will carry out acceptance testing for the Fortify Me

1. Maj Faisal Jatoi
2. Maj Usama Nisar Khan
3. Capt Imran Aziz

6.2 Summary

Testing not only maintains the software and system quality but also improves over all usability and stability of the project. At different stages of development suitable testing techniques were used to ensure product worked accurately and efficiently. Almost all the errors detected during testing were removed.

6.3 Results and Analysis

6.3.1 Results and Analysis Introduction

Fortify me has been developed to effectively and quickly send SOS messages. With minimum user interaction maximum stuff is performed by the application itself in the background. User just needs to enter minimum information. The limitation is that it has not been tested on different devices. There are thousands of type of devices available in the market and android development requires that you test your products to ensure compatibility with maximum devices.

6.3.2 Results

The major purpose was to make emergency response fast by creating an environment where help can arrive to a person as quickly as possible. The app requires minimum user interaction and provides a framework for quick emergency response.

6.3.3 Analysis

Architecturally speaking the system lacks cohesion of components. This way improvements and add-on of new features can be done without any hassle.

7 Conclusion and Future Work

The purpose of this project was to learn various aspects of android and java. Implementing https programming in order to fetch the required information, bind them together and dispatch all this in the shortest possible time frame. The interface is designed to maintain an easy approach for a common man mainly not equipped with any of the IT paradigm. The primary aim to develop this app is to create a homogeneous atmosphere among comrades and colleagues and neighbors to form a progressive society with bondage of love, in line with the teachings of Islam. Future work will be focused on improving the quality of this application with add-on features including follow me and power button operations. Efforts would also be made for incorporating with NADRA for the verification of SIM along with interface design with twitter, LinkedIn and my5.

Due to limitation of time we were able to provide only a few features while some features were left. Future work would aim at targeting maximum audience.

Appendix A: Glossary

Activity diagram: An analysis model that shows a dynamic view of a system by depicting the flow from one activity to another. Similar to a flowchart.

Assumption: A statement that is believed to be true in the absence of proof or definitive knowledge.

Class: A description of a set of objects having common properties and behaviours, which typically correspond to real-world items (persons, places, or things) in the business or problem domain.

Class diagram: An analysis model that shows a set of system or problem domain classes and their relationships.

Compatibility: Capable of orderly, efficient integration and operation with other elements in a system with no modification or conversion required.

Constraint: A restriction that is imposed on the choices available to the developer for the design and construction of a product.

API (application programming interface): specifies how some software components should interact with each other. In addition to accessing databases or computer hardware, such as hard disk drives or video cards, an API can be used to ease the work of programming graphical user interface components.

Wi-Fi: Wi-Fi is the name of a popular wireless networking technology that uses radio waves to provide wireless high-speed Internet and network connections

Wi-Fi Router: used to provide access to the Internet.

OS: Operating System

Data flow diagram: An analysis model that depicts the processes, data collections, and flows among them that characterize the behaviour of a business process or of a software system.

Dependency: A reliance that a project has on an external factor, event, or group outside its control.

External interface requirement: A description of an interface between a software system and a user, another software system, or a hardware device.

Feature: A set of logically related functional requirements that provides a capability to the user and enables the satisfaction of a business objective.

Flowchart: An analysis model that shows the processing steps and decision points in the logic of a process or of a program.

Hardware: A computer and the associated physical equipment directly involved in the performance of data-processing or communications functions.

Hardware Interface: The logical and physical characteristics of each interface between the software product and the hardware components of the system.

Implementation: Execution of a plan, idea, model, design, specification, standard, algorithm, or policy.

Interface: A point where two systems, subjects, organizations, etc., meet and interact.

Non-functional requirement: A description of a property or characteristic that a software system must exhibit or a constraint that it must respect, other than an observable system behaviour.

Operating Environment: The circumstances surrounding and potentially affecting something that is operating.

Operating System: A collection of software that manages computer hardware resources and provides common services for computer programs.

Perspective: The way in which objects appear to the eye.

Post condition: A condition that describes the state of a system after a use case is successfully completed.

Precondition: A Condition that must be satisfied before a use case may begin.

Procedure: A written description of a course of action to be taken to perform a given activity, describing how the activity is to be accomplished.

Process: A sequence of activities performed for a given purpose. A process description is a documented definition of those activities.

References:List of any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.

Response:A reaction, as that of an organism or a mechanism, to a specific stimulus.

Scope: The portion of the ultimate product vision that the current project will address. The scope draws the boundary between what's in and what's out for the project.

Software requirements specification: A collection of the functional and non-functional requirements for a software product.

Specification, requirements: The process of documenting a system's requirements in a structured, shareable, and manageable form. Also, the product from this process.

Stimulus:Something causing or regarded as causing a response.

Supplementary Information:Something added to complete the information.

System requirement: A top-level requirement for a product that contains multiple subsystems, which could be all-software or software and hardware.

Usability:Fit for use; convenient to use.

Use case: A description of an interaction between an actor and a system that results in an outcome that provides value to the actor.

Use case diagram: An analysis model that identifies the actors who can interact with a system to accomplish valuable goals and the various use cases that each actor will perform.

User: A customer who will interact with a system either directly or indirectly (for example, using outputs from the system but not generating those outputs personally). Also called end user.

User class: A group of users for a system who have similar characteristics and requirements for the system.

User Interface: the logical characteristics of each interface between the software product and the users.

User requirement: User goals or tasks that users must be able to perform with a system, or statements of the user's expectations of system quality.

Validation: The process of evaluating a work product to determine whether it satisfies customer requirements.

Verification: The process of evaluating a work product to determine whether it satisfies the specifications and conditions imposed on it at the beginning of the development phase during which it was created.

Vision: A long-term strategic concept of the ultimate purpose and form of a new system.

Vision and scope document: A document that presents the business requirements for a new system, including a product vision statement and a project scope description.

SMS: short message service

PC: personal computer

Call log: display of missed, received and dialled calls

Appendix B: Bibliography

- [1]. IT103 Research Paper; The Android Operating System; "Benefits of the Android Operating System" October 2011 <http://jyangit103.blogspot.com/2011/10/benefits-of-android-operating-system.html>
- [2]. Android Developers Website "Managing The Activity Lifecycle" <http://developer.android.com/training/basics/activity-lifecycle/index.html>
- [3]. The Server Side Enterprise Java Community; "What's the big IDE? Comparing Eclipse and NetBeans" <http://www.theserverside.com/feature/Whats-the-Big-IDE-Comparing-Eclipse-vs-NetBeans>
- [4]. Eastern Michigan University; "Development Techniques for Android Platform and Mobile Device Application" January 2012 <http://commons.emich.edu/cgi/viewcontent.cgi?article=1762&context=theses>
- [5]. Android Developers Website ; "Application Fundamentals" <http://developer.android.com/guide/components/fundamentals.html>

- [6]. Javaworld Developers Website ; “Socket Programming in Java: A Tutorial ” ; Dec 1996
<http://www.javaworld.com/jw-12-1996/jw-12-sockets.html?page=1>
- [7]. AndroidHive ; “Android Working with Android API” ; Sept. 2013
<http://www.androidhive.info/2013/09/android-working-with-camera-api/>
- [8]. Vogella Tutorials ; “Android Animations”
<http://www.vogella.com/articles/AndroidAnimation/article.html>
- [9]. Vogella Tutorials; “Android Logging ”
<http://www.vogella.com/articles/AndroidLogging/article.html>
- [10]. Vogella Tutorials; “Android Custom Views”
<http://www.vogella.com/articles/AndroidCustomViews/article.html>
- [11]. Vogella Tutorials ; “Android Drag and Drop”
<http://www.vogella.com/articles/AndroidDragAndDrop/article.html>
- [12]. Vogella Tutorials ; “Android ListViews ”
<http://www.vogella.com/articles/AndroidListView/article.html>
- [13]. Tutorials Point Blog ; “Java Multithreading”
http://www.tutorialspoint.com/java/java_multithreading.htm

Appendix C: User Manual

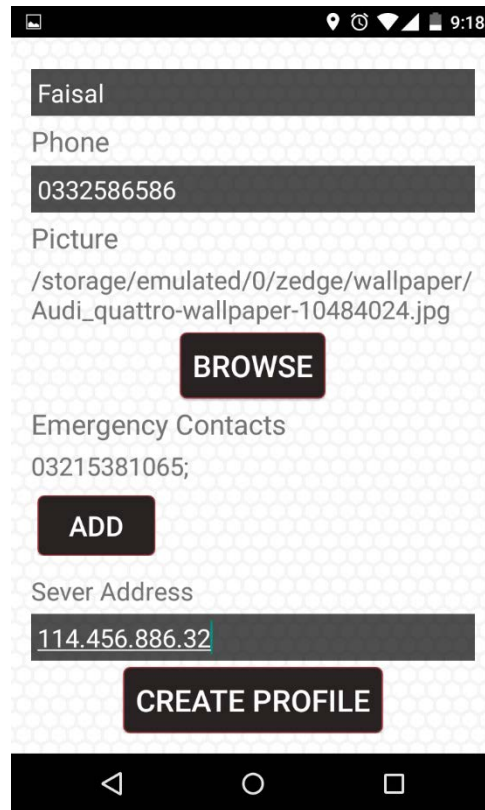
Fortify Me consists of two basic components. These include profile creation and sending SOS Message.

Profile Creation:

To create a profile Go to the profile menu



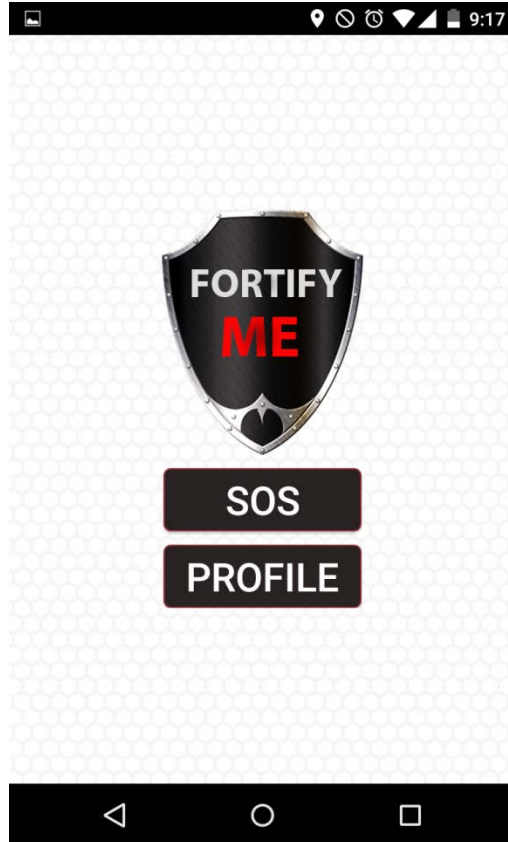
Select **Create Profile** and You will go the profile page.



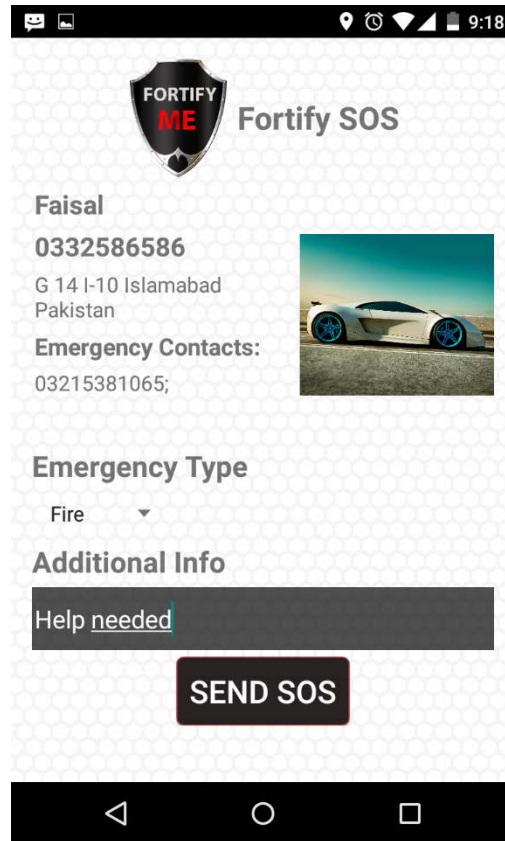
Enter all the input details, select picture and press **Create Profile**. Profile will be created and then it will ask to login to Facebook. Provide our Facebook credentials and you are then logged in to Facebook as well.

SOS Generation:

To generate SOS Message first press **SOS** button



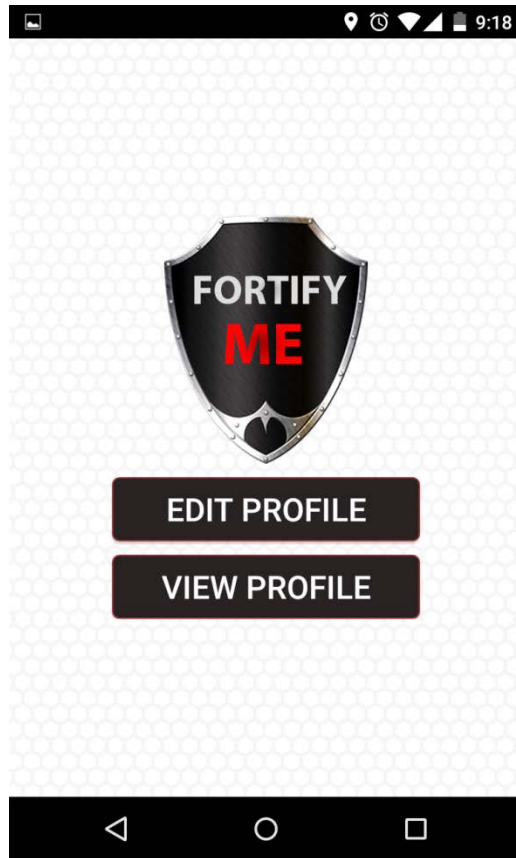
You will be presented to a page where data will be fetched from your profile and address will be fetched using coordinates. The user then needs to add the Emergency type and Comments.



Press **Send SOS** and SOS Message will be generated and Facebook prompt will be presented.

View Profile:

To view profile simply press **View Profile** Button.



You will get the profile



Fortify Profile

Faisal

0332586586

Emergency Contacts:
03215381065;

