GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

ANDROID SAFETY APPLICATION

A Report for the Evaluation 3 of Project 2

Submitted by

MANAS SRIVASTAVA

(1613101367)

In partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

Under the Supervision of

KARTHICK R.

Assistant Professor

MAY- 2020

SCHOOL OF COMPUTING AND SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report "ANDROID SAFETY APPLICATION" is the bonafide work of "MANAS SRIVASTAVA (1613101367)" who carried out the project work under my supervision.

SIGNATURE OF HEAD

Dr. MUNISH SHABARWAL,
PhD (Management), PhD (CS)
Professor & Dean,
School of Computing Science & Engineering

SIGNATURE OF SUPERVISOR

Karthick R., M.Tech.,
Assistant Professor
School of Computing Science & Engineering

Table of Content

CHAPTER 1

INTRODUCTION

1.1 Introduction

This report explains the details of development process for the **Android Safety Application Using GPS and Messaging System** software package.
The main feature of this application is to provide location tracking functionality to Android devices using SMS. This application locates a device by making a call to the device and gets its location in the form of the latitude and longitude of that Android device. The application also has the capability of authentication to allow the sender to share the location information with the receiver of SMS.
The application features a personal safety alarm that sends an emergency message to the user's chosen contacts with the push of single button. In this application all 'guardians' receive a text message with a link to a map showing the user's location via GPS. It gives the user two additional levels of safety: A risk mode with real time GPS tracking of the user's positions and a timer mode with automatic alarm activation.

1.2 Statement of the Problem

People talk much about safety; but are unable to do anything to prevent unsafe incidents. Citizens fight for human rights on the television and other public forums, but are often unable to take any concrete actions to help the general public. Here we have tried to develop an Android application to help the users enhance their security with the help of their mobile device.

1.3 Motivation

As a result of technological progress we are facing an incredible variety of possibilities to communicate regardless of the distance. Smart phones provide a great choice of features that facilitate the life for the users as well as make it more comfortable.
Every day the features and capabilities of handheld mobile phones are increasing at a surprising rate. For this reason, we wanted to create an Android application which gives us the opportunity to improve our knowledge of Mobile application development. We find this project a great opportunity to combine many technologies and languages in the same software system, and learn how to work as a team in projects. An important motivation for us was the challenge to solve all the problems that arise while the project is developed. In addition, we wanted to make both the user experience comfortable and the program portable, reliable, secure, stable and intuitive.

1.4 Basic concepts and Tools

1.1 Introduction to Java

Java is a programming language created by James Gosling from Sun Microsystems (Sun) in 1991. The first publicly available version of Java (Java 1.0) was released in 1995. Sun Microsystems was acquired by the Oracle Corporation in 2010. Over time new enhanced versions of Java have been released. The current version of Java is Java 1.7 which is also known as *Java 7*. From the Java programming language, the *Java platform* evolved. The Java platform allows software developers to write program code in other languages than the Java programming language and still runs on the Java virtual machine. The *Java platform* is usually associated with the *Java virtual machine* and the *Java core libraries*.

Java Virtual machine

The Java virtual machine (JVM) is a software implementation of a computer that executes programs like a real machine. The Java virtual machine is written specifically for a specific operating system, e.g. for Linux a special implementation is required as well as for Windows.

Java Runtime Environment vs. Java Development Kit

A Java distribution comes typically in two flavors, the *Java Runtime Environment* (JRE) and the *Java Development Kit* (JDK). The Java runtime environment (JRE) consists of the JVM and the Java class libraries and contains the necessary functionality to start Java programs. The JDK contains in addition the development tools necessary to create Java programs. The JDK consists therefore of a Java compiler, the Java virtual machine, and the Java class libraries.

Characteristics of Java

The target of Java is to write a program once and then run this program on multiple operating systems.
Java has the following properties:

Platform independent: Java programs use the Java virtual machine as abstraction and do not access the operating system directly. This makes Java programs highly portable. A Java program

(which is standard complaint and follows certain rules) can run unmodified on all supported platforms, e.g. Windows or Linux.

Object-orientated programming language: Except the primitive data types, all elements in Java are objects.

Strongly-typed programming language: Java is strongly-typed, e.g. the types of the used variables must be pre-defined and conversion to other objects is relatively strict, e.g. must be done in most cases by the programmer.

Interpreted and compiled language: Java source code is transferred into the byte code format which does not depend on the target platform. These byte code instructions will be interpreted by the Java Virtual machine (JVM). The JVM contains a so called Hotspot-Compiler which translates performance critical byte code instructions into native code instructions.

Automatic memory management: Java manages the memory allocation and de-allocation for creating new objects. The program does not have direct access to the memory. The so-called garbage collector deletes automatically objects to which no active pointer exists.[4]

1.2 Android

Android is a software platform and operating system for mobile devices. Android is available as open source. It allows developers to write managed code in the Java language, controlling the device via Google-developed Java libraries.

Android SDK was released by Open Handset Alliance in the month of November of the year 2007. Android was actually developed using the kernel of Linux 2.6 and the highlighting features of Android include the following:

• No fees for licensing, distribution and release approval
• GSM, 3G EDGE networks for telephony
• IPC message passing
• Background processes and applications
• Shared data stores
• Complete multimedia hardware control
• API’s for location based services such as GPS.

1.2.1 Architecture of Android OS

The skeleton of Android framework and its constituents are shown in the following figure:

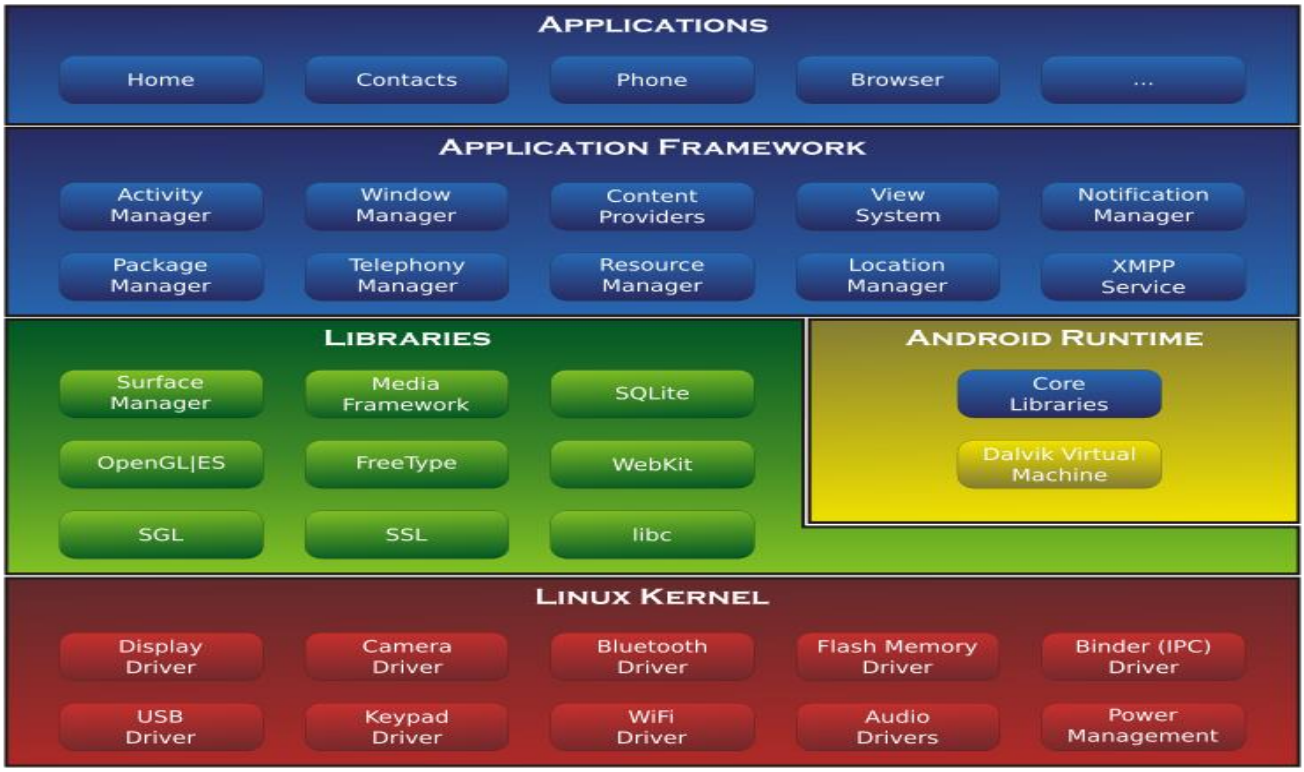

Figure 1.1: Architecture of Android OS

Applications Layer

Android ships with a set of core applications including an email client, SMS program, calendar, maps, browser, contacts and others. All applications are built using the Java. Each of the applications aims at performing a specific task that it is actually intended to do.

Application Framework Layer

The next layer is the application framework. This includes the programs that manage the phone's basic functions like resource allocation, telephone applications, switching between processes or programs and keeping track of the phone's physical location. Application developers have full access to Android's application framework. This allows them to take advantage of Android's processing capabilities and support features when building an Android application. We can think of the application framework as a set of basic tools with which a developer can build much more complex tools.

Libraries Layer

The next layer contains the native libraries of Android. These shared libraries are all written in C or C++, compiled for the particular hardware architecture used by the phone and preinstalled by the phone vendor.

Android Runtime Layer

Android Runtime layer includes Dalvik Virtual Machine (DVM) and a set of core java libraries. Every Android app gets its own instance of DVM. Dalvik has been written so that a device can run multiple virtual machines efficiently and it executes files with .dex (Dalvik Executable Format) extension optimized for minimum memory.

3.2.2 Components of Android

The basic components of an Android application include Activity, Broadcast Receiver, Service, and Content Provider. Each of the above, which when used for any application, has to be declared in the AndroidManifest.xml. The user interface of the component is determined by the Views. For the communication among these basic components we use Intents and Intent filters which play crucial role during app development.

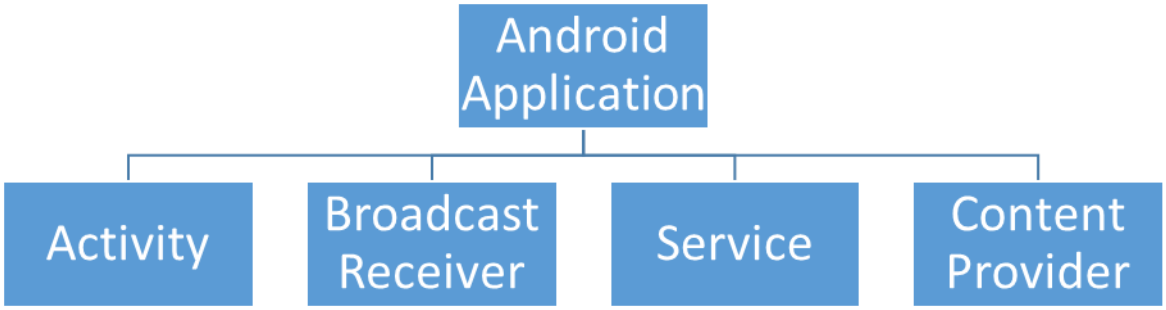

Figure 1.2 Android Components

Activity

An Activity is, fundamentally, an object that has a lifecycle. An Activity is a chunk of code that does some work, as necessary. The work can include displaying a UI to the user, though it doesn't have to as some Activities never display UIs. Typically, we designate one of our application's Activities as the entry point to our application.

Broadcast Receiver

Broadcast Receiver is yet another type of component that can receive and respond to any of the broadcast announcements.

Service

A Service is a body of code that runs in the background. It can run in its own process, or in the context of another application's process, depending on its needs. Other components "bind" to a Service and invoke methods on it via remote procedure calls. An example of a Service is a media player; even when the user quits the media-selection UI, she probably still intends for her music to keep playing. A Service keeps the music going even when the UI has completed.

Content Provider

Content Provider is a data storehouse that provides access to data on the device; the classic example is the Content Provider that is used to access the user's list of contacts. Our application can access data that other applications have exposed via a Content Provider, and we can also define our own Content Providers to expose data of our own.

1.2.3 Location based Services in Android

Android's Network Location Provider determines user location using cell tower and Wi-Fi signals, providing location information in a way that works indoor and outdoor, responds faster, and uses less battery power. The purpose of location-based services is to find the Physical location of the device. Access to the location-based services is handled by the Location Manager system Service. To access the Location Manager, request an instance of the LOCATION_SERVICE using the get System Service() method. Current Location can be fetched using two ways:

1. GPS (Global Positioning System)
2. Network Service Location

GPS (Global Positioning System)

The Global Positioning System (GPS) uses a constellation of 24 satellites orbiting the earth. GPS finds the user position by calculating differences in the times the signals, from different satellites, take to reach the receiver. GPS signals are decoded, so the smart phone must have in-built GPS receiver. To get access to GPS hardware of android we request using following statement **LocationManager.GPS_PROVIDER;**

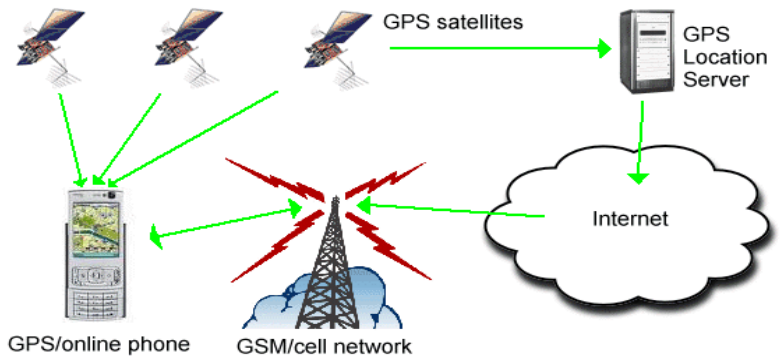

Figure 1.3 Architecture of a GPS System

Network Service Location

The current cell ID is used to locate the Base Transceiver Station (BTS) that the mobile phone is interacting with and the location of that BTS. It is the most basic and cheapest method for this purpose as it uses the location of the radio base station that the cell phone is connected to. A GSM cell may be anywhere from 2 to 20 kilometers in diameter. Other approaches used along with cell ID can achieve location granularity within 150 meters. The granularity of location information is poor due to Wide Cell Range. The advantage is that no additional cost is attached to the handset or to the network to enable this service.

To get access to Network Provider android we request using following statement **LocationManager.NETWORK_PROVIDER.**

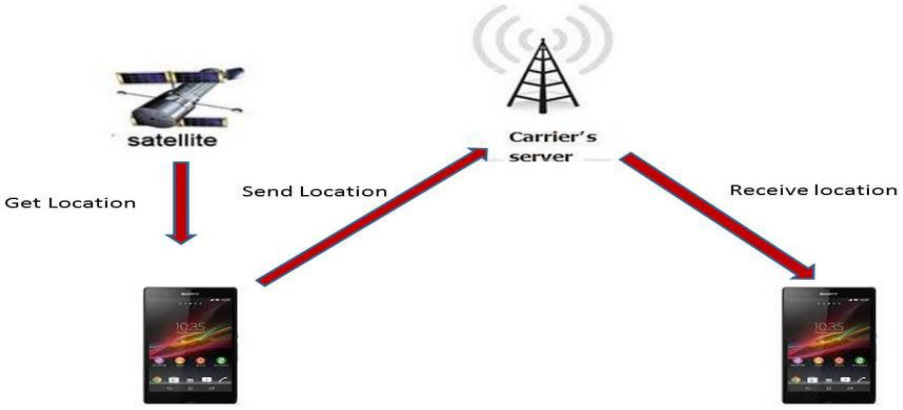

Figure 1.4 Showing network services

3.3 Eclipse

Eclipse is a multi-language software development environment comprising an integrated development environment (IDE) and an extensible plug-in system. It can be used to develop applications in Advanced Java and, by means of various plug-ins with other programming languages like C and C++. Its features are:

- It is Open Source.
- It is strongly recommended by Android developers.
- It is directly linked with the compiler, so we don’t need to compile the program.
- It has a good User Interface.

CHAPTER 2

EXISTING SYSTEM

2.1 Existing Systems

Ringer

A silent phone can be extremely tricky to find. People in the habit of losing a silent cell phone may wish to invest in a phone sensor, also known as a phone detector. These are tools that, when placed near a cell phone, will actually pick up the call signal and make sounds to indicate that the phone is somewhere within proximity. If the phone is lost, all you need to do is have someone call you as you walk around with the sensor until the device begins to indicate that a call signal is nearby. When you hear the signal, you then have a basic idea of where to start looking for your cell phone.

Phone Tracking Using IMEI Number

Every phone comes with a unique International Mobile Equipment Identify Number which can come in useful to track it in case of loss or theft. This number can be accessed by dialing *#06# and it is advisable to make a note of it as soon as you purchase your handset. In case the phone gets stolen, one is advised to file an FIR with the police and give them its identity number.

A copy of the FIR and IMEI number can be passed on to the service provider who will then be able to track the lost handset. With its IMEI number, a device can be traced even if it is being used with another SIM or even if it is switched off. Once the handset is located, the user can request the service provider to block it from being used the user is able to get your hands on it again.

CHAPTER 3

PROPOSED SYSTEM

3.2 Proposed System

Ringer

Using simple SMS commands you can ring your Android Device even though it is in silent mode and thus locate your device locally.

Location Tracking

In this proposed system the user can locate any phone that has been misplaced or stolen. Once the App is installed on the phone, it can be located by sending an SMS with predefined keyword. The system gets you current updated location.

CHAPTER 4

IMPLEMENTATION

4.1 Introduction

After designing the new system, the whole system is required to be converted into a language understood by the computer. This is accomplished by coding. Coding is an important stage where the defined procedures are transformed into control specifications with the help of a computer language. This is also called the programming phase in which the programmer converts the program specifications into computer instructions, which we refer to as programs. The programs coordinate the data movements and control the entire process in a system.

It is generally felt that programs must be modular in nature. This helps in fast development, maintenance and future change, if required.

The validity and proper functionality of all the modules of the developed application is assured during the process of implementation. Implementation is the process of assuring that the information system is operational and then allowing the user to take over its operation for use and evaluation.

Implementation is the stage in the project where the theoretical design is turned into a working system. The implementation phase constructs, installs and operates the new system. The most crucial stage in achieving a new successful system is that it works effectively and efficiently.

4.2 Modules

1. Register four numbers: Firstly the user has to register the four numbers as per their choice.

2. Send predefined text: Then after clicking on send option a message can send on the four registered numbers.

3. Get latitude and longitude of device and create a Google map link: Send device location to the sender of SMS.

4. Exit Application

The Broadcast receiver then alerts the application when each new SMS arrives.

4.2.1 Implementation of the modules

1. **Get Location and Acknowledge user.**

Example received predefined text is “I am in trouble. My location is……...”.

In this module we provide the functionality of getting location details of the device.

Step 1: Start

Step 2: Check that the internet is available.

Step 3: If the internet is available then get location details from Network Provider.

Step 4: If internet is not available, then check if GPS is turned on.

Step 5: If GPS is available then get location details.

Step 6: Send location information to the receiver.

Step 7: End

4.3 State Machine Diagram

A State diagram is a graph whose nodes are states and whose directed arcs are transitions between the states. It specifies the state sequences caused by event sequences. State names must be unique within the scope of the diagram. State diagrams are used to give an abstract description of the behavior of a system. This behavior is analyzed and represented in series of events that could occur in one or more possible states.

4.4 Sequence Diagram

A Sequence diagram shows how a set of objects communicate with each other to perform a complex task. This type of diagram allows the other developer to verify that the interaction is correct.

A Sequence diagram shows, as parallel **vertical lines (lifelines),** different processes or objects that live simultaneously, and as **horizontal arrows**, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

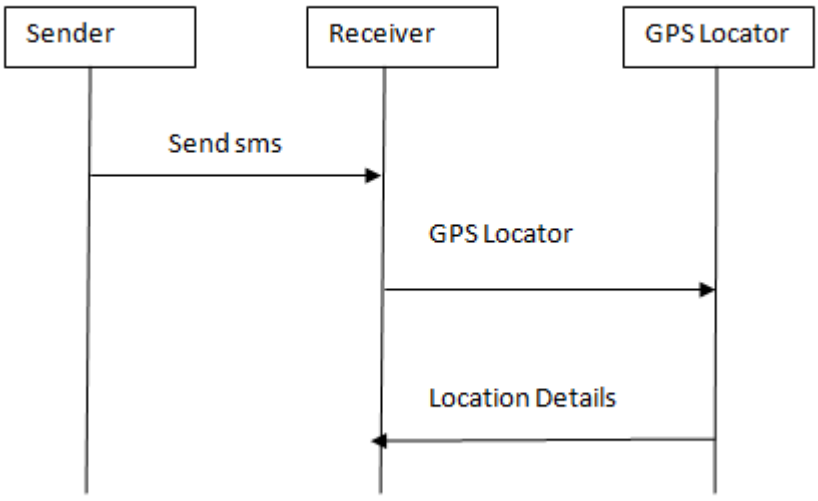

Figure 4.1 Showing state diagram

CHAPTER 5

RESULTS

After testing the application we get the following results which are available in the form of screen shots as below.

Snapshot 1: It is the home page of the application from where we send the message..

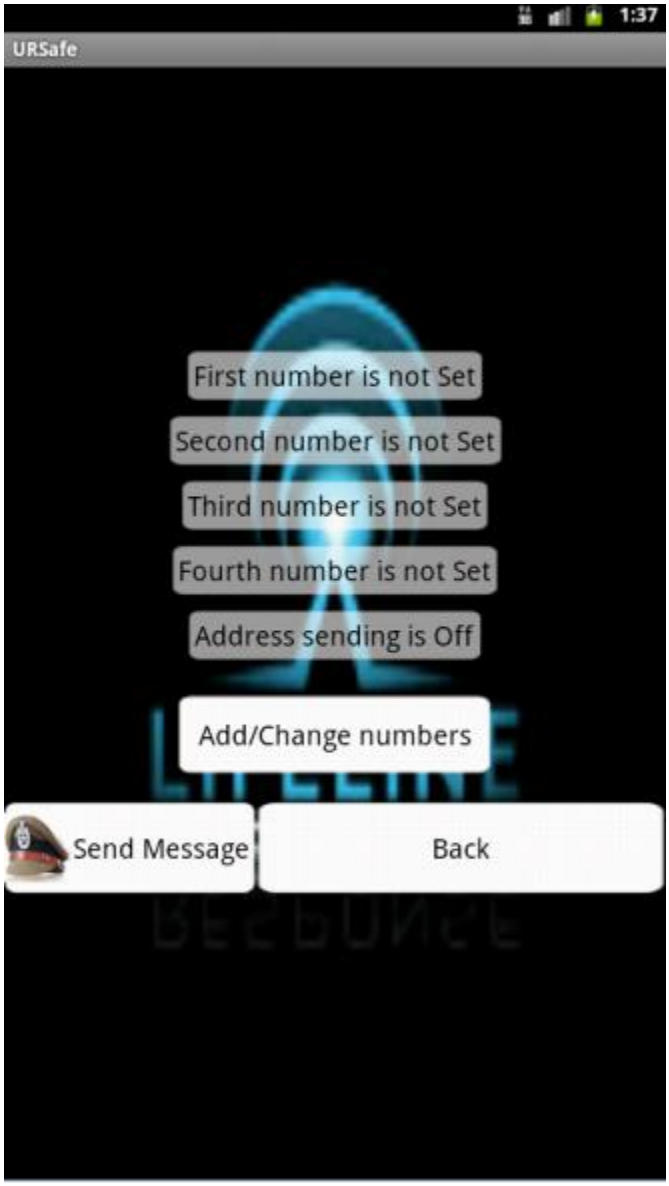

Figure 5.1

Snapshot 2: Shows the screen where the user has to register the four numbers as per their choice.

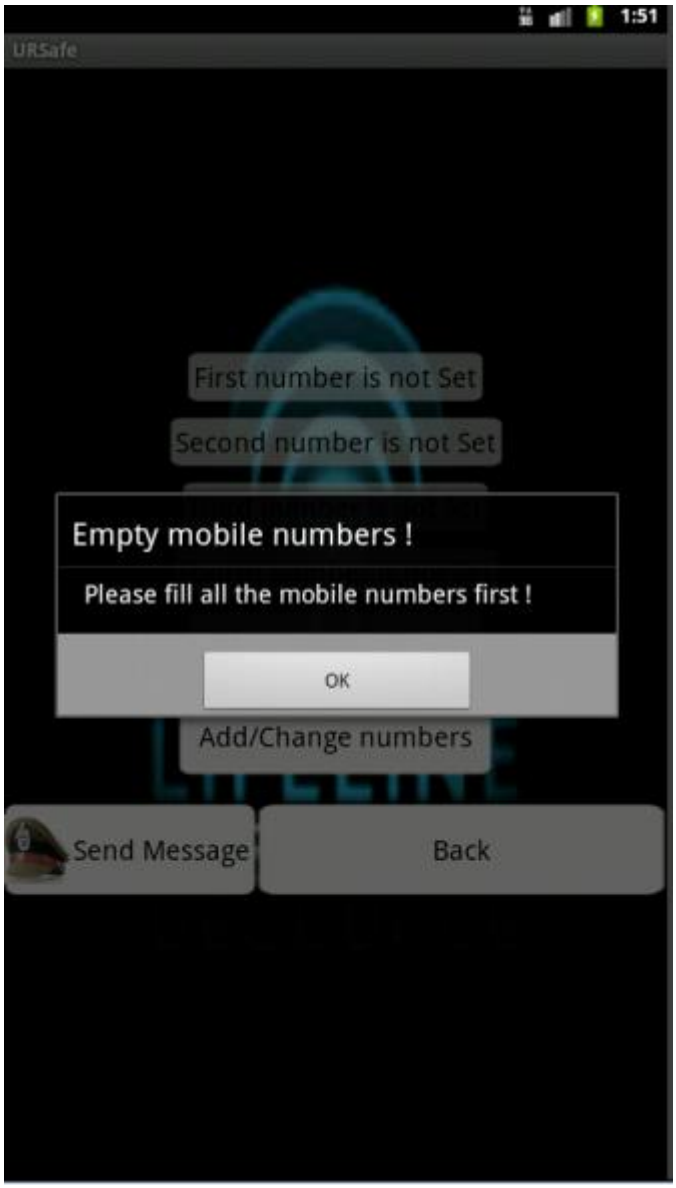

Figure5.2

Snapshot 3: In this the chosen numbers are filled out.

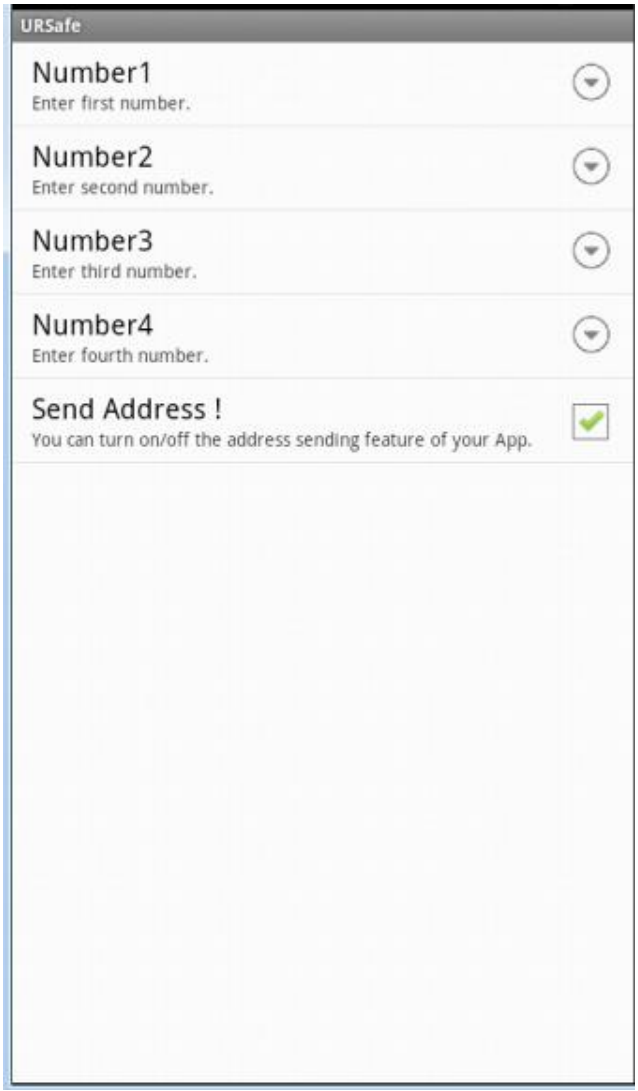

Figure 5.3

Snapshot 4: In this, the system looks for Internet and GPS services to be available before sending the message.

Figure 5.4

Snapshot 5: In this, the system recognizes the internet connection as disabled and asks for it to be enabled.

Figure 5.5

CHAPTER 6

CONCLUSION AND SCOPE FOR FUTURE WORK

Conclusion

"A safe (community) means that all people, regardless of gender, race, ethnicity, language, disability, age or sexual orientation, have an equal right to freedom from fear and violence. We as a community have a responsibility to address the issue of violence because it belongs to everyone."

All the features work on SMS basis. Therefore, incoming SMS format plays a vital role. Our android application running in the cell monitors all the incoming messages. If the SMS is meant for the application, it reads the same and performs the expected task.

We have created features, which will enhance the existing tracking system. Application stands different from the existing system as it's not only the GPS value it makes use of but it works on text messaging services which makes application a simple & unique one.

Future Work

There is scope for improvement in the existing application. In particular the following facilities can be added:

- It should be able to work without internet facility.
- It should have calling facility.
- It should be able to work with zero balance.

These facilities will improve the usability of this application in the event of a genuine emergency.