TABLE OF CONTENTS

9.4 9.5 9.6	References Bibliography Appendix	

1. Introduction to project

1.1 Introduction

Now-a-day's women security is big question and this project is an answer to this question. This saves time and at the same time creates sense of security where ever she is.

There was need felt that should recognise voice for first time when the App is started.

The SMS shall be automatically sent as soon as the word Help is heard.

This application is generally meant for the attention of the authorities or public in the emergency response capabilities such as terrorist attacks and the natural disaster by facilitating the communication with their respective along the mobile phones.

The difficulties in the existing application are the lack of situational awareness and communication terminology among their respective. Due to this response and recovery is Difficult to the authorities. In respect of the public safety with the support of the network provider the application runs in the android phones in efficient way to identify and recover the problem by the natural disaster or terrorist attacks etc…

Furthermore users are likely to operate the mobile devices for the security purpose to intimate the problem detection to their respective in the emergency cases.

This software is very usable for those who wish to seek help when they in condition of needing help from anyone. The project is related to Women security and makes them independent to go anywhere. They only have to say the word Help to seek help from others.

This software will be used by:

One anyone who is interested in seeking Help from other?

1.2 Problem definition

"Women are at risk". Everyone is acquainted with the fact and try out in every possible way to help them in any situation. But the fact is that we want something that can alert us regarding the place of incident. But with the changing world the developments are also very fast. We are now more or less carefree regarding our women relative.

1.3 Motivation

The main aim of developing the project is to help women in every possible way in every situation they are and whenever they ask for. And obviously the more important of this is the reporting of the activity to the buddies and police and the location of the activity. These fundamental requirements has laid the foundation of such an app which is a necessity for the time.

1.4 Objective

"Women Security Android App" is a Software Developed to aid any woman in getting help from the predefined number she has saved on her mobile. This is special software developed only for girls and women who are in a danger zone and wish to get help from their near and dear ones or it can be police. The App thus developed needs internet connection for the successful execution. The App works on the principle of voice recognition and shall send an SMS to the person whom she considers can help at that time.

The App is doing the work of guard who is there all the time with her and if she says the word "help" from her mouth that special voice shall be recognized and thus SMS sent to the Contact Number she has stated when starting this App.

We will develop the software for the sake of learning new things in Android and Windows and using some of the hidden features of the language.

1.5 Proposed solution

The proposed system i.e. ***"Women Security Android App"*** *is* an App based system with advanced security feature. The system is developed in Android as a front-end tool under the windows platform. The proposed system will take over the existing system very efficiently. The scope of the system includes the following feature:

- A number is required for SMS.
- Internet connection needs to be present.
- The App shall recognise the voice of the user.
- The SMS is feed beforehand.

1.6 Platform Specification

Modern handheld devices such as smart phones and PDA's have become increasingly powerful in recent years. Dramatic breakthroughs in processing power along with the number of extra features included in these devices have opened the doors to a wide range of commercial

possibilities. In particular, most cell phones regularly include cameras, processors comparable to PCs from only a few years ago, and internet access. However even with all these abilities, there are a few applications that allow much passing of the environmental information and location based services.

As mobile devices become more like PCs they will come to replace obHCETts we tend to carry around such as cheque books, credit cards, cameras, planners, mp3 players, etc. In short, we will be using them to accomplish our daily tasks. On application that falls into this category is the HCET College Android App developed for Android phones.

The prime obHCETtive of "HCET College Android App" is to create a full fledged Android application which could locate a list of doctors, hospitals based on location and type of specialty entered by the user. The user not finds a list of Doctors or Hospitals but also he can make a choice of the best doctor based on the rating and specialty he chooses to have. The user has the advantage of placing an emergency call to any of the doctors. The applications fruits doesn't end over here, further going ahead user can track all the donors wishing to donate a specified blood group. The donors list shall be present as soon as the blood group is specified by the user.

The project shall be developed using Java Programming Language by using Netbeans IDE. We shall be making use of Android Software Development Kit which includes a variety of custom tools that help us develop mobile applications on the Android platform. The most important of these are the Android Emulator and Android Development Tools plug-in for Netbeans.

Android

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. Android is a software and operating system for mobile devices based on the Linux Operating System and developed by Google and the Open Handset Alliance. It allow developers to write managed code in JAVA-like language that utilizes Google-developed Java libraries, but does not support programs developed in native code.

The unveiling of the Android platform on 5 November 2007, was announced with the founding of Open Handset Alliance, consortium of 34 hardware, software and telecom companies devoted to advancing open standards for mobile devices. When released in 2008,

most of the Android platform will be made available under the Apache free-software and open-source license.

There are a number of technological and organizational skills involved in building a software project, and the process presented with an opportunity to foster such skills – firstly, to learn a new platform that will likely find its way to the top of mobile segment within the next few years, along with an array of client-server technologies associated with the system; secondly, to get a chance to work with a real project that concerns a charitable cause; thirdly, to create a foundation for future development of actual Android-based hardware with the inputting and tracing functions built in it.

Android is a new player in the mobile industry, but it is rapidly gaining support in the software development circles.

Its software is written in JAVA, perhaps the most commonly used language today with support of thousands of modules, extensions and pages of documentation. It's open and linux based and gives direct access to hardware, meaning that "the only limitation is imagination" in designing Android software.

Google has made steps to cement the platform's reputation among developers and ensure its future development – steps such as producing regular updates to the SDK and founding the Open Handset Alliance.

Because of the system's openness and flexibility it's easy to produce a budget Android-based device to be sold in the targeted areas - the deciding factor of choosing Android over the closed iPhone framework that provides a similar set of features.

Unfortunately, the fact that Android first came to light only a couple of years ago presented a challenge; some of its functions were still poorly documented and required outside research. In addition, the platform has undergone drastic changes and much of code had to be redone.

Mobile Telephony

An Android-based device can receive text messages; in this context it serves as a hub for several mobile phones in vicinity limited only by the network provider's range of service. The

idea for this function originated from a designer who said, “it’s easy to design an Android device. It’s harder to mimic its functionality on a cheap mobile phone.”

The mobile phone users can send coded text messages to the android device in order to input product-related data, or request market information, in which case a response is sent back to the phone. The advantages of this scheme are evident:

SMS functionality can be found on virtually any mobile phone that’s less than 2 decades old;

Text messages are a universally accepted way of communication and can be sent by someone with minimal technological skills;

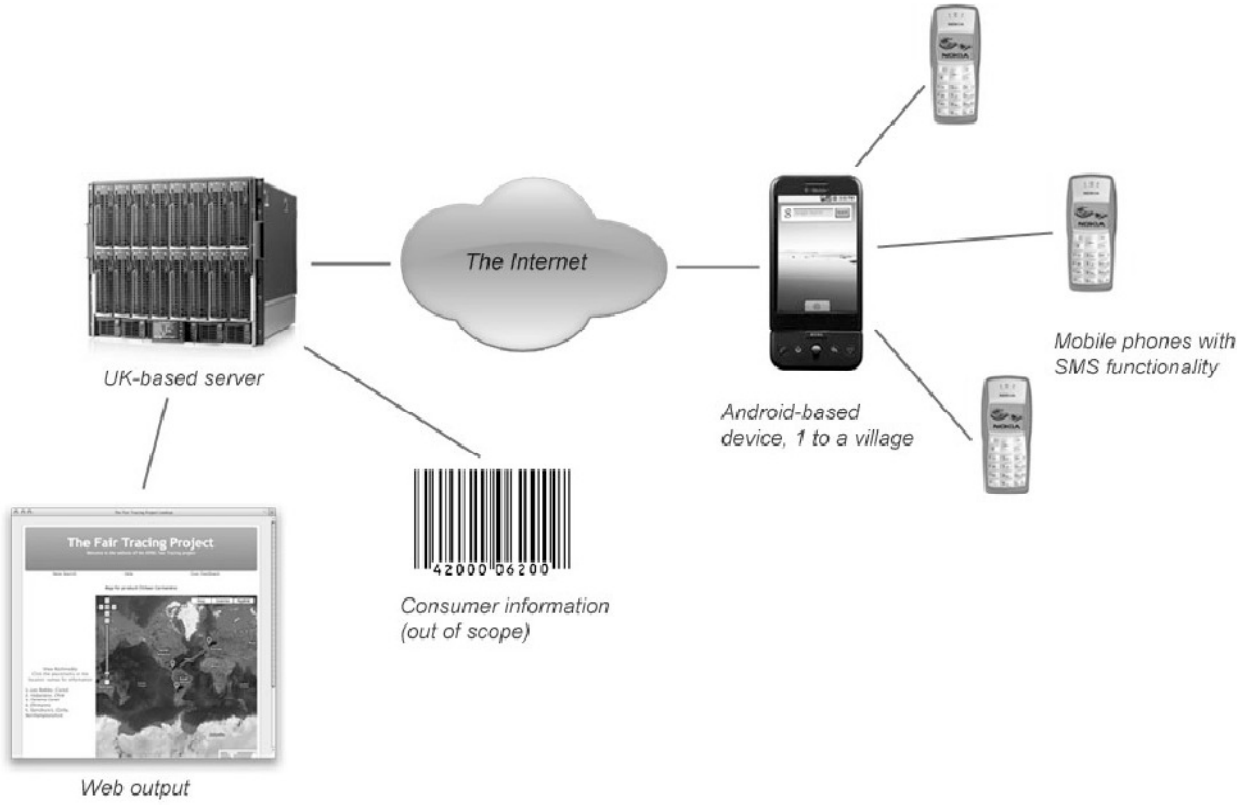
There’s no need for a 3rd-party receiving service that charges additional fees, the only cost is the cost of sending a message.

There are certain limitations too:

The receiving device has to be switched on and the application has to be running;

The sender must be within the range of service of the network provider.

Therefore the following arrangement takes place from the producer’s point of view:



1.6.1 Hardware specification

Any Mobile with Android OS

1.6.2 Software Interfaces

- Front End Tool :Netbeans/Eclipse
- Language : JAVA
- Android SDK

1.6.3 Tools and technology

1.1.1 ANDROID

The Android SDK tools available from the SDK Manager provide additional command-line tools to help you during your Android development. The tools are classified into two groups: SDK tools and platform tools. SDK tools are platform independent and are required no matter which Android platform you are developing on.

Platform tools are customized to support the features of the latest Android platform.

1.1.1.1 SDK Tools:

The SDK tools are installed with the SDK starter package and are periodically updated. The SDK tools are required if you are developing Android applications.

The most important SDK tools include the Android SDK Manager (android sdk), the AVD Manager (android avd) the emulator (emulator), and the Dalvik Debug Monitor Server (ddms). A short summary of some frequently-used SDK tools is provided below.

Virtual Device Tools:

Android Virtual Device Manager

The AVD Manager provides a graphical user interface in which you can create and manage Android Virtual Devices (AVDs) that run in the Android Emulator.

Android Emulator (emulator)

A QEMU-based device-emulation tool that you can use to debug and test your applications in an actual Android run-time environment.

SDK Manager

Lets you manage SDK packages, such as installed platforms and system images.

Sqlite3

Lets you access the SQLite data files created and used by Android applications. ANDROID DEBUGGER BRIDGE

Android Debug Bridge (adb) is a versatile command line tool that lets you communicate with an emulator instance or connected Android-powered device. It also provides access to the device shell.

1.7 Scope and marketing

This software is very usable for those who wish to seek help when they in condition of needing help from anyone. The project is related to Women security and makes them independent to go anywhere. “Women are at risk”. Everyone is acquainted with the fact and try out in every possible way to help them in any situation. But the fact is that we want something

that can alert us regarding the place of incident. But with the changing world the developments are also very fast. We are now more or less carefree regarding our women relative.

They only have to say the word Help to seek help from others.

This software will be used by:

One anyone who is interested in seeking Help from other?

The App is free to any user.

2 Back ground and related work

2.1 Existing system

There are many such project in the market. The existing method is of calling if any such situation arises. There are many such products available in market. But to put together the kind we are developing is not exactly the one. This is a kind multiple techniques to gain something new. The very effort of our team is to get fruitful results from the software, while there can be chances of up gradation.

2.2 Proposed system

The proposed system i.e. ***"Women Security Android App"*** *is* an App based system with advanced security feature. The system is developed in Android as a front-end tool under the windows platform. The proposed system will take over the existing system very efficiently. The scope of the system includes the following feature:

- A number is required for SMS.
- Internet connection needs to be present.
- The App shall recognise the voice of the user.
- The SMS is feed beforehand.

3 System analysis and deign

3.1 Feasibility study

The feasibility study is not a full-blown systems study. Rather, the feasibility study is used to gather broad data to make a decision on whether to proceed with system study. System project feasibility is assessed in three principal ways:

- Economically

- Technically
- Operationally

The organization has evaluated cost of software and hardware required for the system including the storage of data. The benefits expected from the system are studied to assess the reduced cost due to the new system.

3.1.1 Technical Feasibility:

Organization has shown willingness to purchase all hardware and software tools which we recommend to successfully implement the system. Hence technically there are no limitations for the development of the system. As far as programming efforts are concerned, we are familiar with java programming. Thus the project is technically feasible.

3.1.2 Economical Feasibility:

The organization has evaluated cost of software and hardware required for the system including the storage of data. The benefits expected from the system are studied to assess the reduced cost due to the new system.

3.1.3 Operational Feasibility:

Operational feasibility is dependent on the humans who will be using the software once it's ready and installed for use. The software will have a user friendly interface which will be much convenient . Thus the project is operationally feasible.

3.2 Non functional requirement

The various non functional requirements are-

- **Effective User Interface**- Since the application is being designed to be used by the normal user's, it was essential that the user interface of the application would be effective and should appear good in its interface.

- **Portability** – Since the application could be installed on any platform supporting Android, the user's can install it on their devices and run, irrespective of their location or migration from one location to other.

- **User friendly** – Our app should be more users friendly. The user interface should be kept simple and uncluttered. Since different type of people will interact in this process so our project should be very easy to them to understand.

- **Flexibility:**

  Our project should be so flexible that whenever we want to make changes in it very easily it can be done on.

- **Security:**

  All the records of users are more secure and arrange in a good manner

- **Extensibility:**

  Our project should be able to accommodate the variations like:

  Different features should be handled easily.

  Client interaction after sending his/her request.

- **Reusable:**

  All the client that are using our project should be easily get processed so that many clients can interact with us very easily and very fast without any information destroy.

- **Cost:**

  Our project should be very feasible and of lower cost so that maximum users will be able to get its benefit.

3.3 Functional requirement

The complete project is divided in to two modules:

- Page designing module
- Programming and logic module
- Registration Module
- Voice Detection Module
- SMS with Location Send Module

1.1.2 Page Designing Module:

In this module we create pages, that are desired for giving help to the person who wishes to seek help. Also by these pages we will send SMS on recognition of voice of that user. Also by

this module we will register the user. Interface developed shall be Android based pages for running on mobile.

1.1.3 Programming & Logic Module:

This is the very important module of this project. This module is responsible for all the required under laying programming and logic for the project. In this module we will make possible to put up the information and processing this information of the user. This module will detect all the clicking events and controls the outputs of the system according to the user request. This module will define all the logic and programming aspects of the system. This module will define all the security constraint and validation for the entire system.

In this module we will use android.

SMS send Module

Text messaging, or texting, refers to the exchange of written messages between fixed-line phone or mobile phone and fixed or portable devices over a network. SMS Manager contains a powerful rule editor which can be used to automate message processing.

3.4 Design

Registration Module – This is the first module, without which user cannot move ahead.

Login Component – All users registered need to login for accessing the service provided by the app.

Buddy Module – This maintains a list of all people whom the user can seek help. Their contact are stored which are informed about the help.

Speak Component – This component is for fetching for the word HELP and then automatically sending SMS with information.

Help Button Component – This is to send SMS only when the HELP button is pressed.

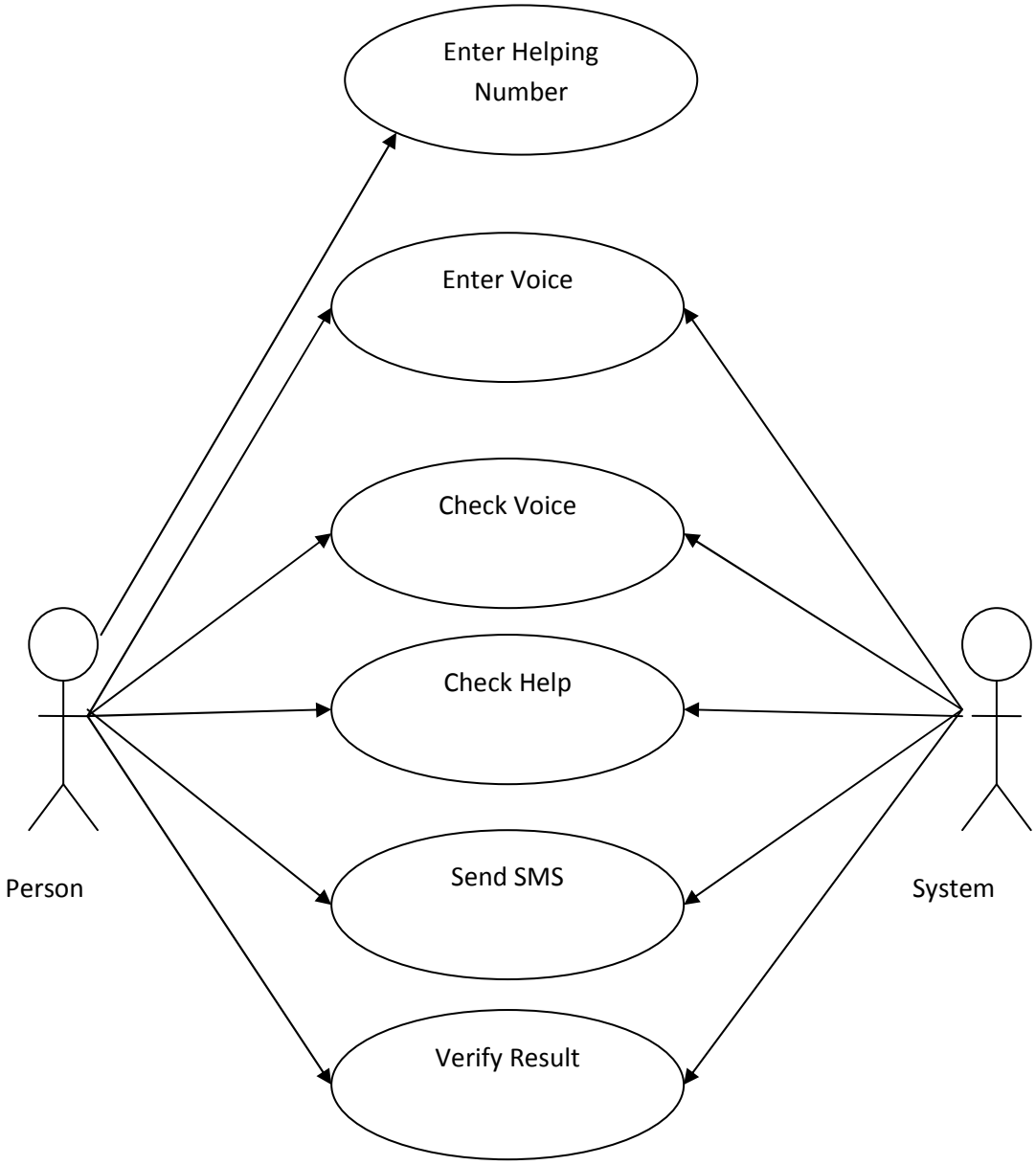
SMS Module – This will send SMS to the Buddies.

Location Fetch Component – This component shall fetch users exact location. It will be sent in the SMS.

Google map Component – This will help in fetching location map of the user to be sent via SMS.

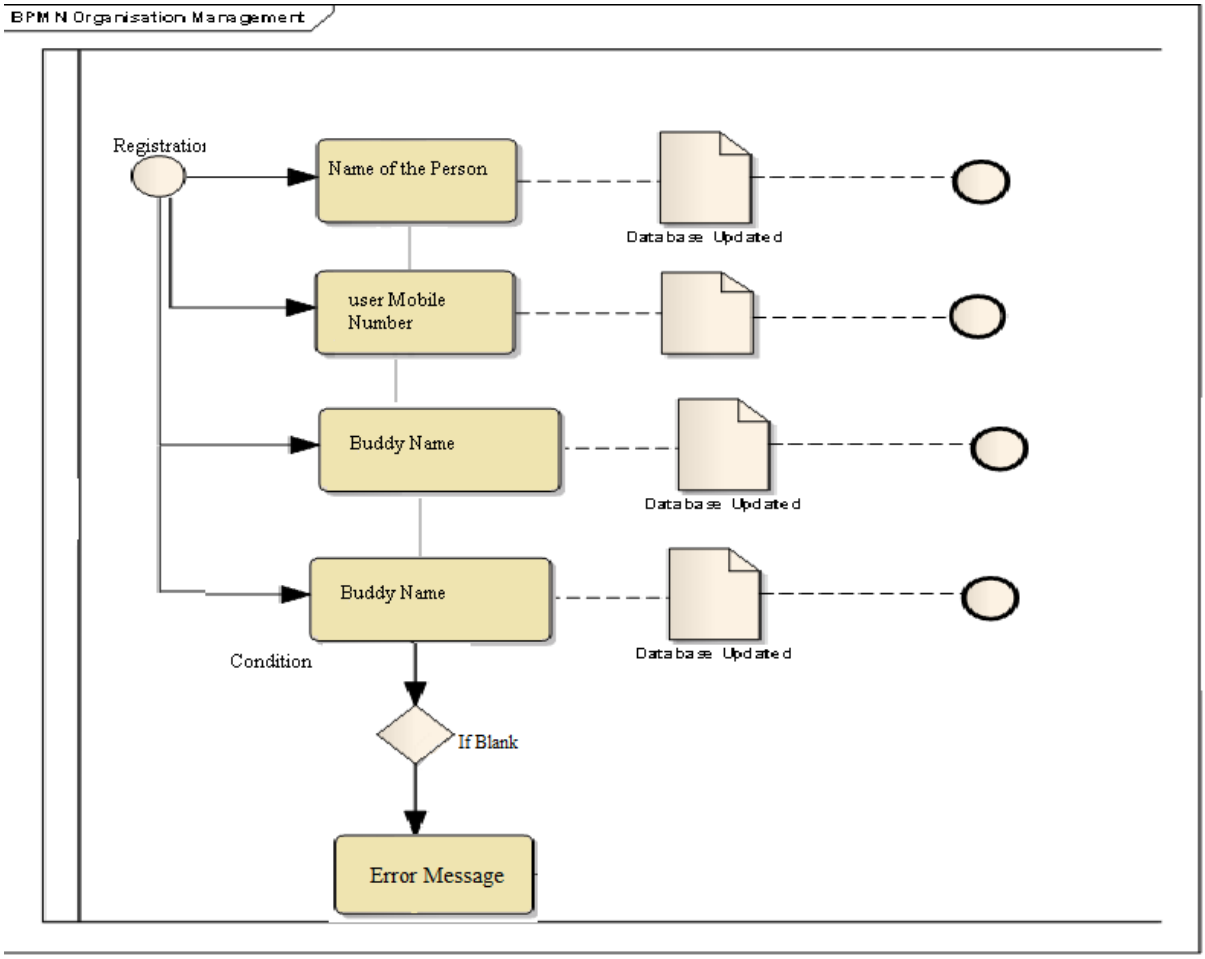
SOS Module – This module is meant for calling the Police for help.

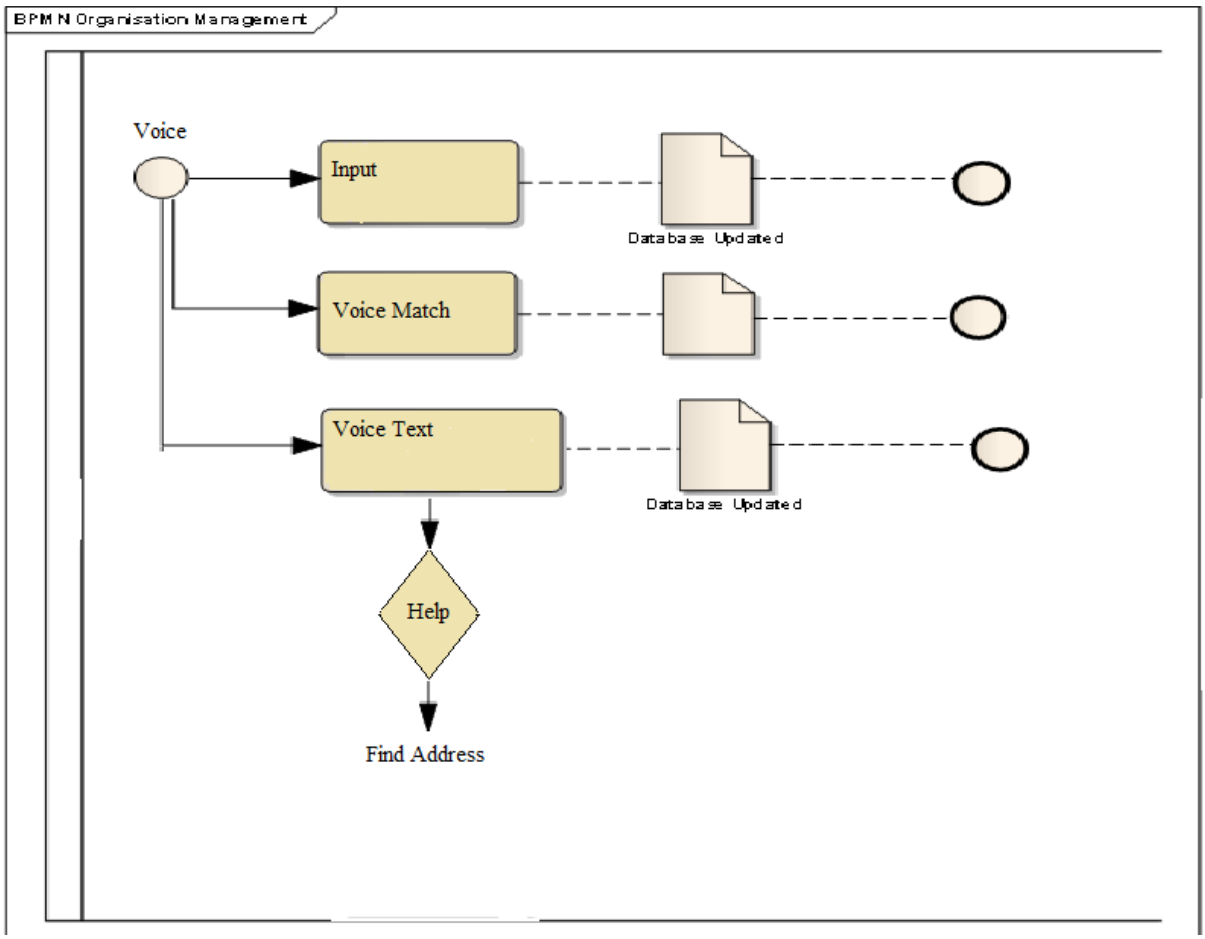
Behavior Model

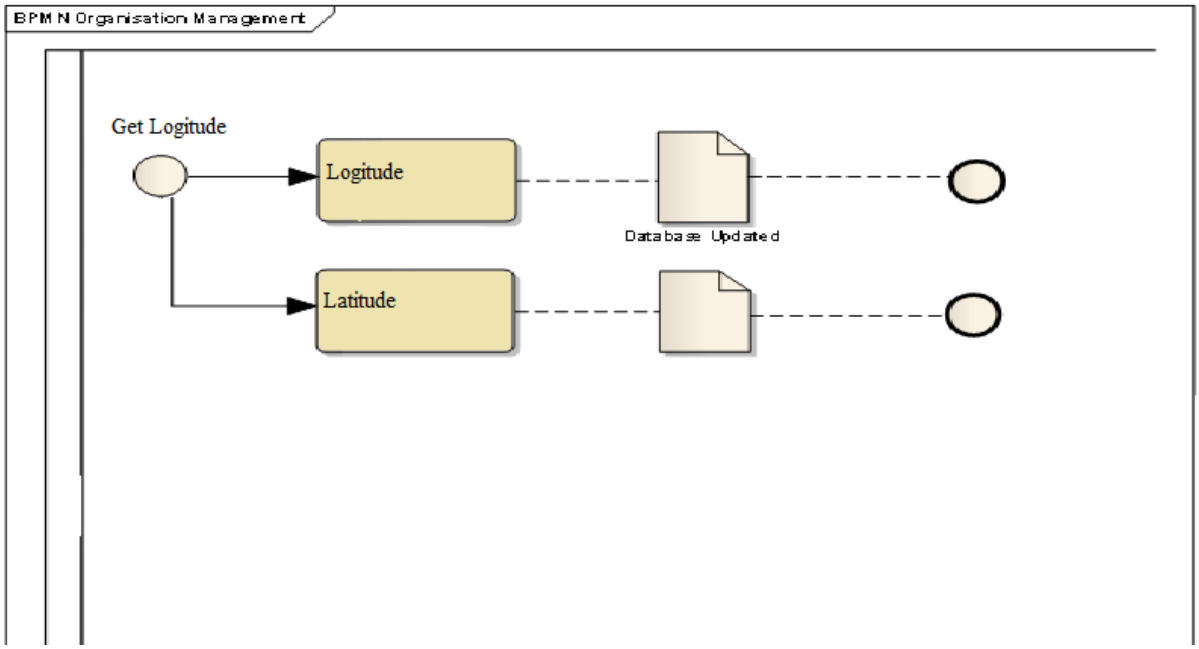


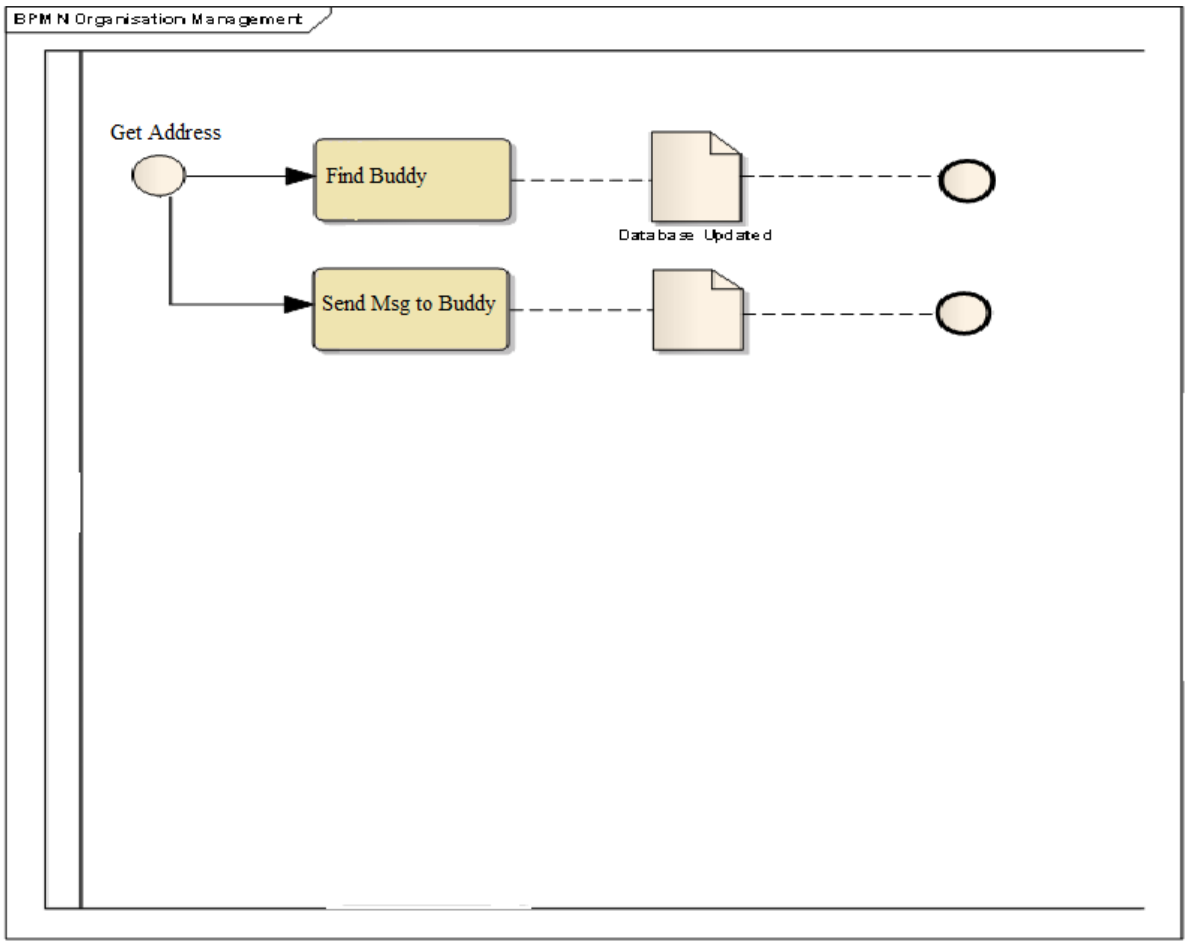
Organisation Management:

Business Process Diagram

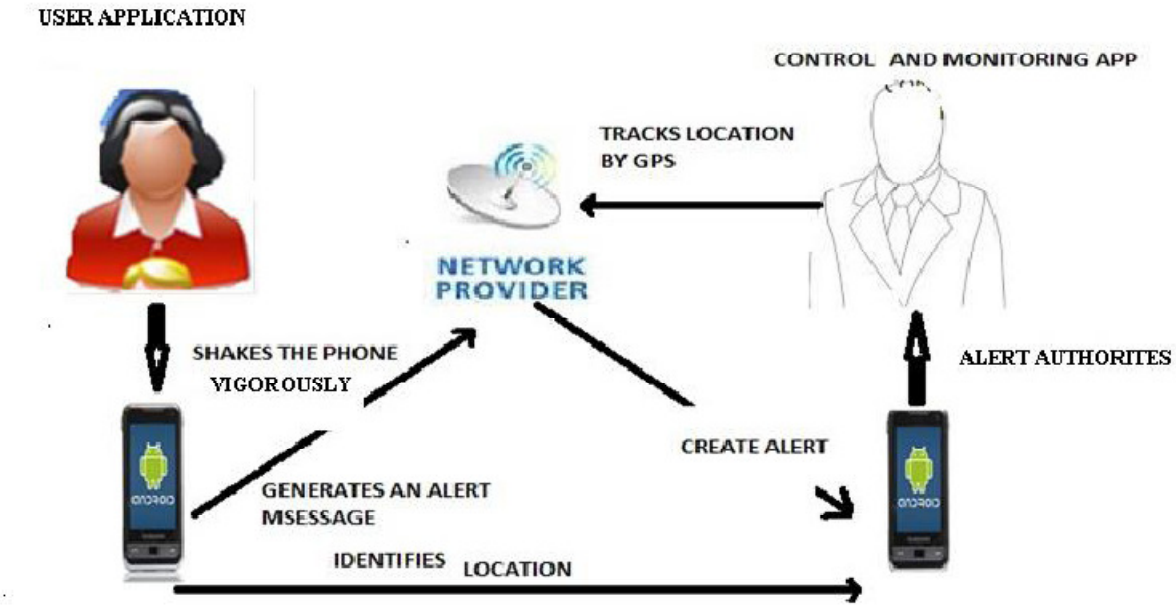



BPMN Organisation Management
Voice
Input
Database Updated
Voice Match
Voice Text
Database Updated
Help
Find Address

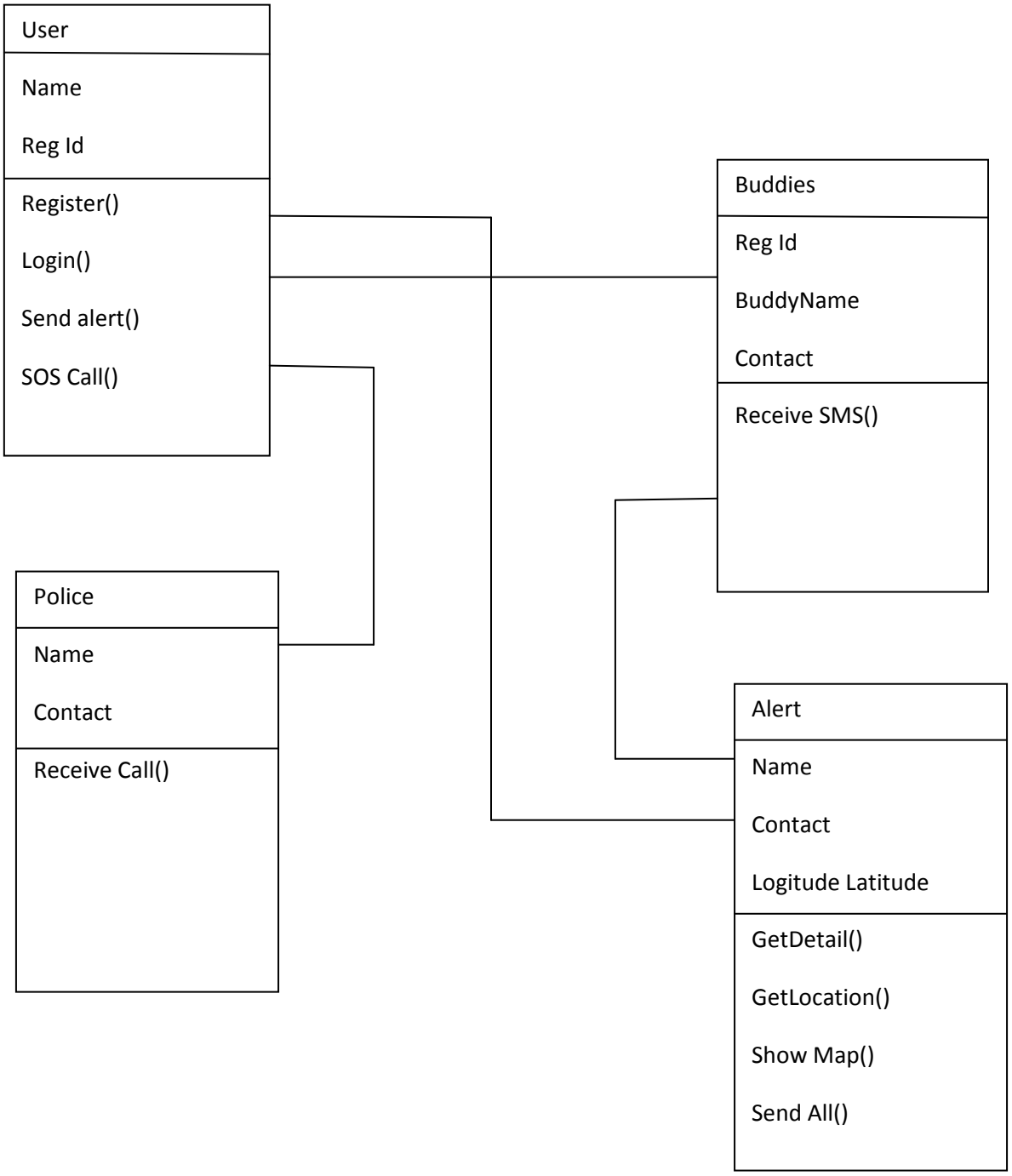

BPMN Organisation Management
Get Logitude
Logitude
Database Updated
Latitude


BPMN Organisation Management
Get Address
Find Buddy
Database Updated
Send Msg to Buddy

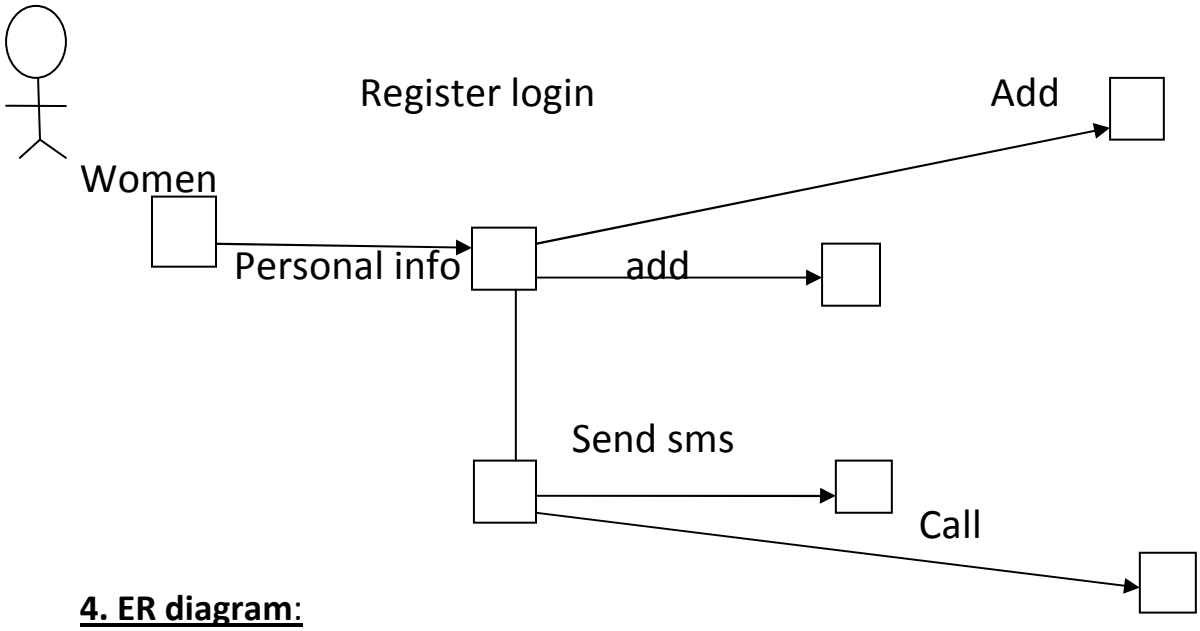
Functional Modeling



Class diagram

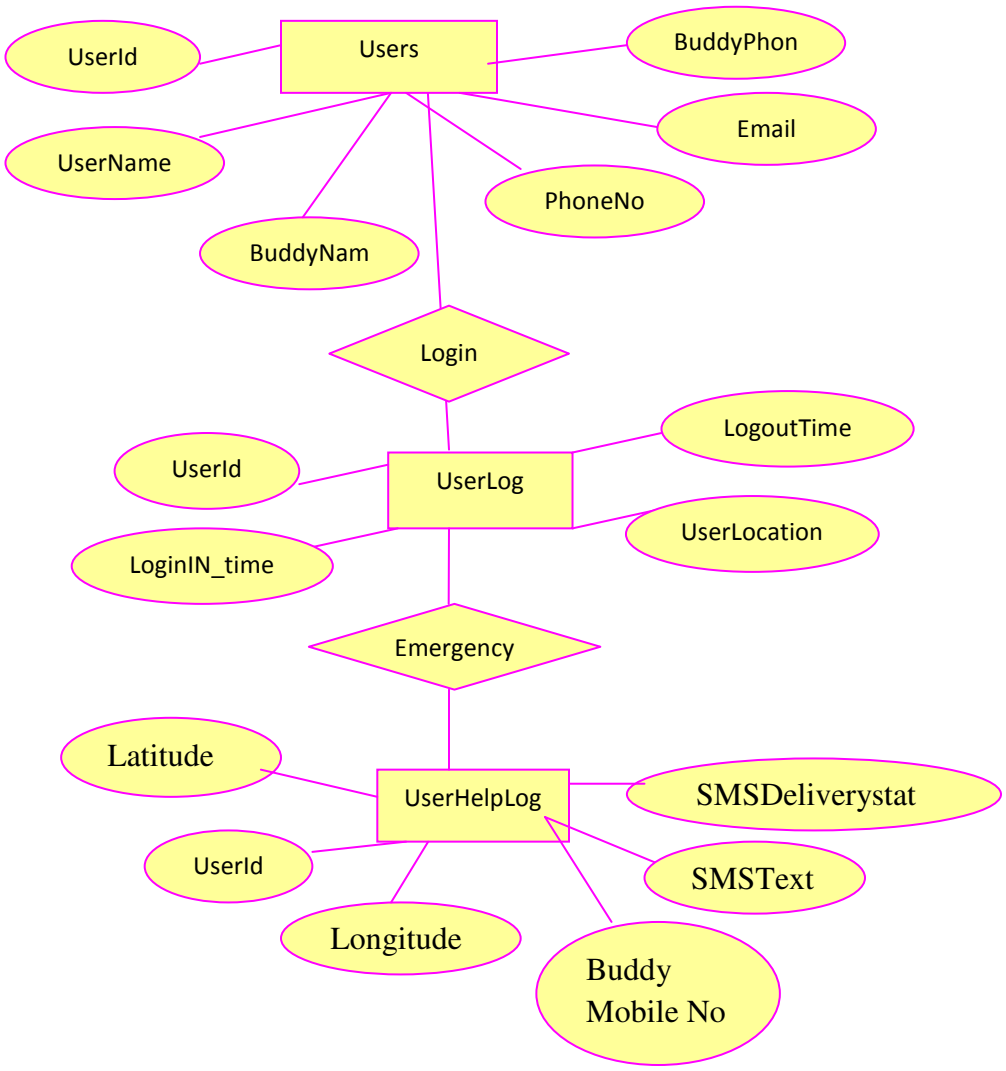


Sequence Diagram



4. ER diagram:

An entity relationship(ER) diagram is a graphical representation of an information system that shows the relationship between people, object, places, concepts or event within that system. An ER diagram is a data modelling technique that can help define business process and can be used as the foundation for a relational database.



5. Data Flow Diagram(DFD)

DataFlow Diagram isthe graphical description of the system's data and how the processes transform the data. The information flow and the transform that are applied as data move from the input to output. It is starting point of the design phase that functionally decomposes the requirement specifications down to the lowest level of details. Thus a DFD describes what data flow(logical) rather than how they are processed.

Unlike details flowchart, data flow diagram do no supply detailed description of the module but graphically describes a system's data interact with the system. to construct a dataflow diagram, we use-

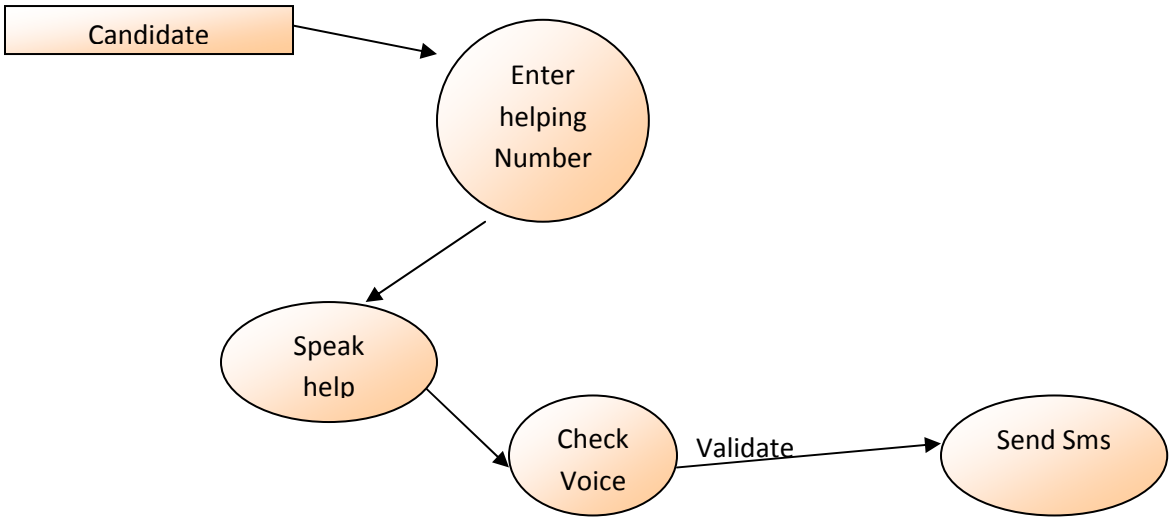
- Arrows
- Circles
- Open end box
- Square

An arrow identifies the dataflow in motion. it is a pipeline though which information is flows like the rectangle in the flowchart. A circle stands for process that converts data into information. An open-ended box represents a data store, Data at rest or a temporary repository of data. Square defines a source or destination of system data.

Rule for constructing a data flow diagram

- Arrows should not cross each other.
- Square, circle and file must be name.
- Decomposed data flow square and circle can have same name.
- Choose meaningful names for dataflow.

Data Flow Diagram

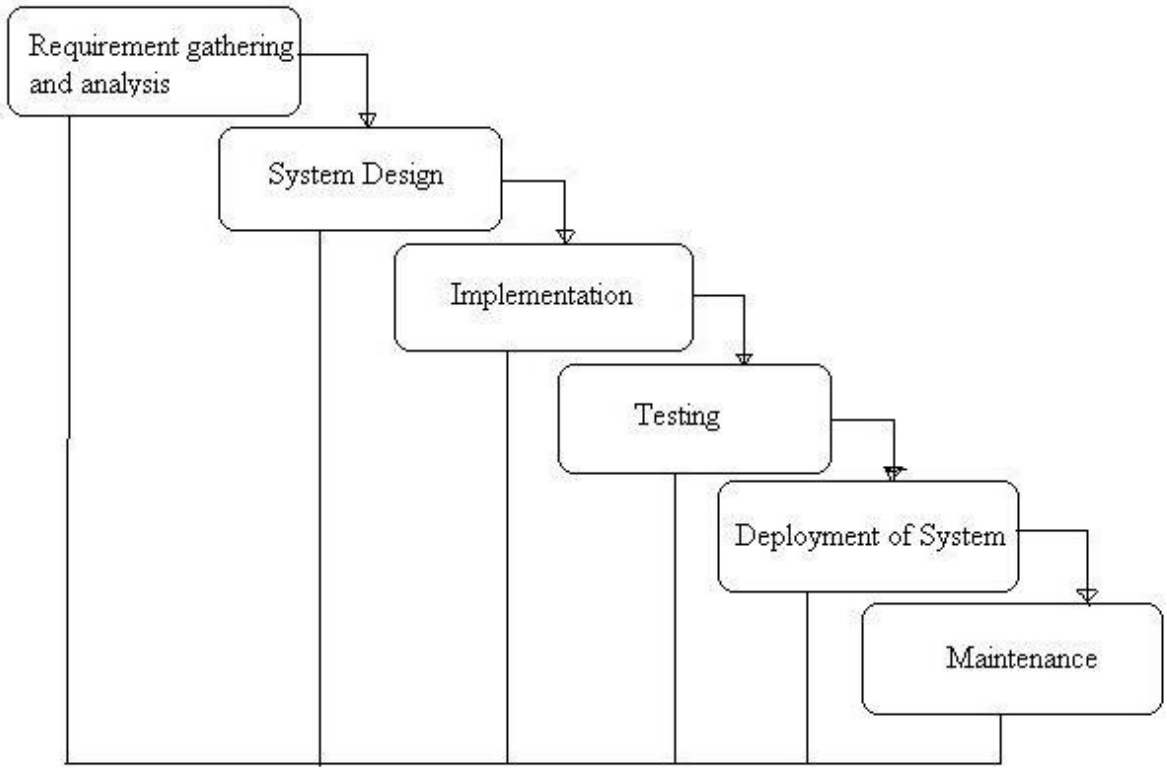


1. Software process model used

The Waterfall Model was first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed fully before the next phase can begin. This type of software development model is basically used for the for the project which is small and there

are no uncertain requirements. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project. In this model software testing starts only after the development is complete. In waterfall model phases do not overlap.

General Overview of "Waterfall Model"



This model is simple and easy to understand and use.

- It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
- In this model phases are processed and completed one at a time. Phases do not overlap.
- Waterfall model works well for smaller projects where requirements are very well understood.
- This model is used only when the requirements are very well known, clear and fixed.
- Product definition is stable.
- Technology is understood.
- There are no ambiguous requirements

- Ample resources with required expertise are available freely

Very less customer interaction is involved during the development of the product. Once the product is ready then only it can be demoed to the end users. Once the product is developed and if any failure occurs then the cost of fixing such issues are very high, because we need to update everywhere from document till the logic.

Looking at the project it must be clear that every step needs to be completed before moving on to the next like a hand or foot needs to be recognized, then the values to be set and other than that if first step is not complete then the next won't work.

6 Implementation

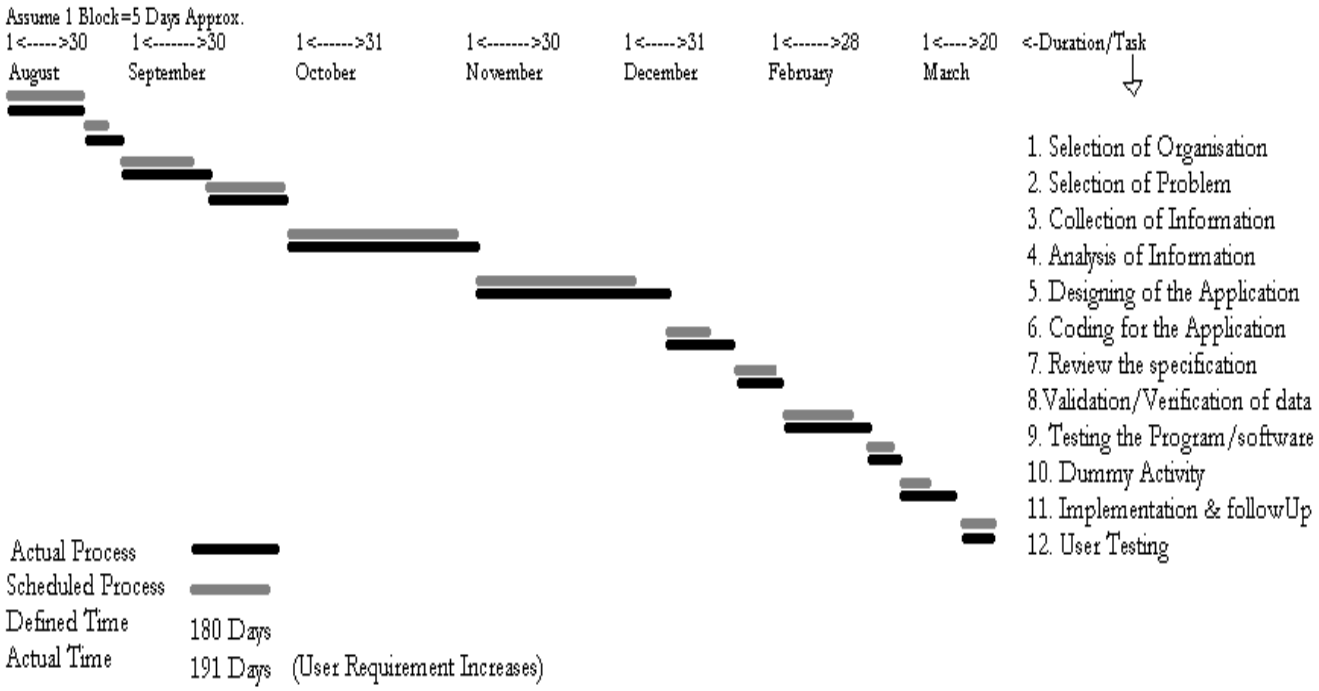
6.1 Implementation

Project execution plan

Which uses a calendar–oriented chart to represent the project schedule? Each activity is represented as a bar in the calendar, starting from the starting date of activity and ending at the ending date for that activity. The start and end of each activity become milestones (or check points) for the project. Coloring each milestone when completed can represent progress. The milestone or checkpoints are usually at the completion of each task.

Let us draw the Gantt chart for each task identified in these project .the horizontal bars indicates the duration of each task and all project tasks are listed in the left handed column. We estimate the no. of days for each task as follows: -Selection of Organization-10 days, Selection of problem-7days,Collection of information –25 days ,Analysis of information –10 days ,Designing the application – 40 days ,Coding for the application -45 days , Review the specification –7days, Verification/Validation – 7 days, Testing -18 days , Dummy activity –5 days , Implementation and follow-up –10 days, user training –7 days

Drawback of GANNT Chart: The main drawback of Gantt chart is that it does not depict the dependency relationships among the different activity.



6.2 History and feature

Now-a-day's women security is big question and this project is an answer to this question. This saves time and at the same time creates sense of security where ever she is.

There was need felt that should recognize voice for first time when the App is started.

The SMS shall be automatically sent as soon as the word Help is heard.

This application is generally meant for the attention of the authorities or public in the emergency response capabilities such as terrorist attacks and the natural disaster by facilitating the communication with their respective along the mobile phones.

The difficulties in the existing application are the lack of situational awareness and communication terminology among their respective. Due to this response and recovery is Difficult to the authorities. In respect of the public safety with the support of the network provider the application runs in the android phones in efficient way to identify and recover the problem by the natural disaster or terrorist attacks etc…

Furthermore users are likely to operate the mobile devices for the security purpose to intimate the problem detection to their respective in the emergency cases.

The main aim of developing the project is to help women in every possible way in every situation they are and whenever they ask for. And obviously the more important of this is the

reporting of the activity to the buddies and police and the location of the activity. These fundamental requirements has laid the foundation of such an app which is a necessity for the time.

6.3 Application

ANDROID APK (Package Installer)

The Android SDK tools available from the SDK Manager provide additional command-line tools to help you during your Android development. The tools are classified into two groups: SDK tools and platform tools. SDK tools are platform independent and are required no matter which Android platform you are developing on.

Platform tools are customized to support the features of the latest Android platform.

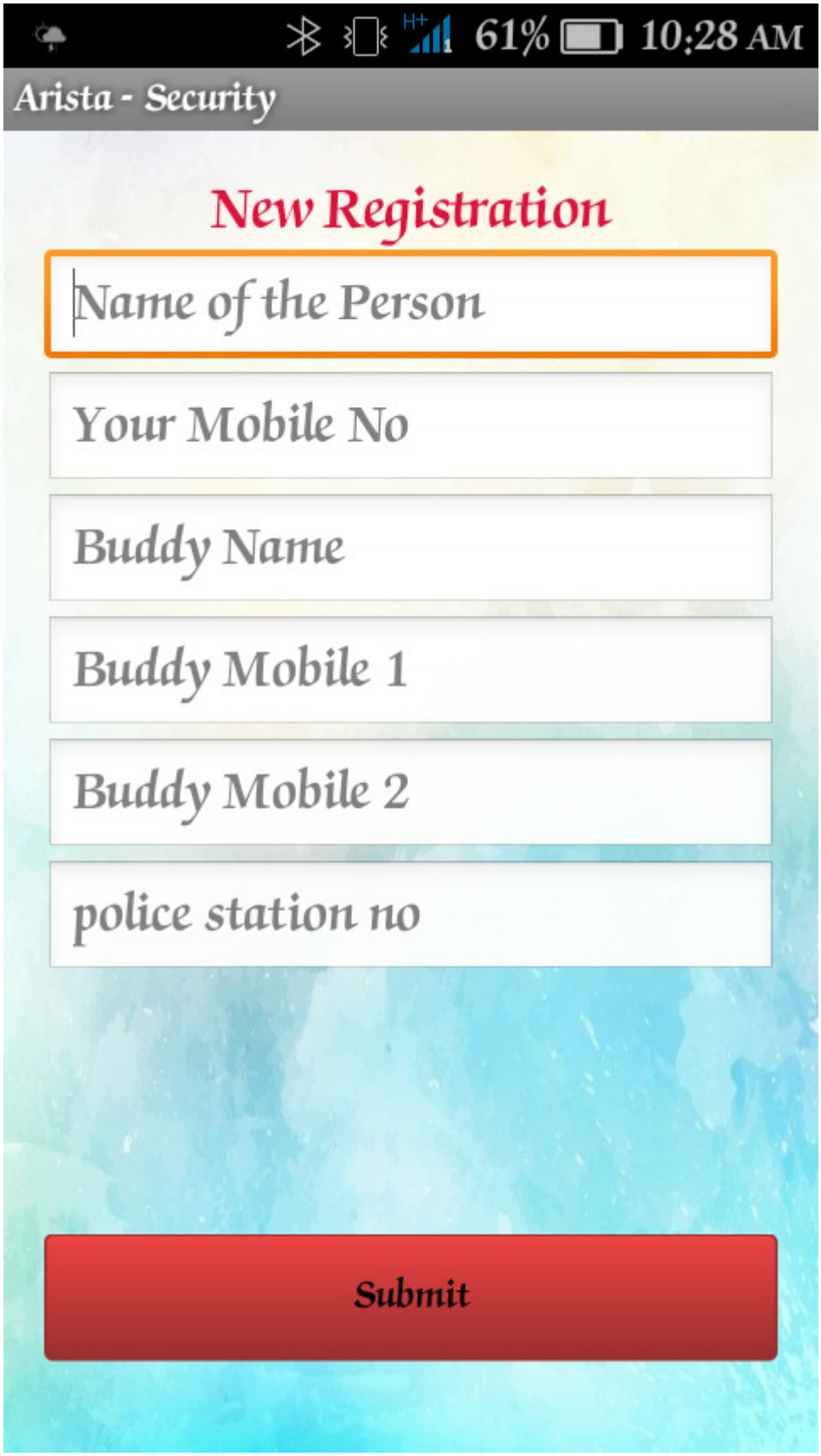
SDK Tools:

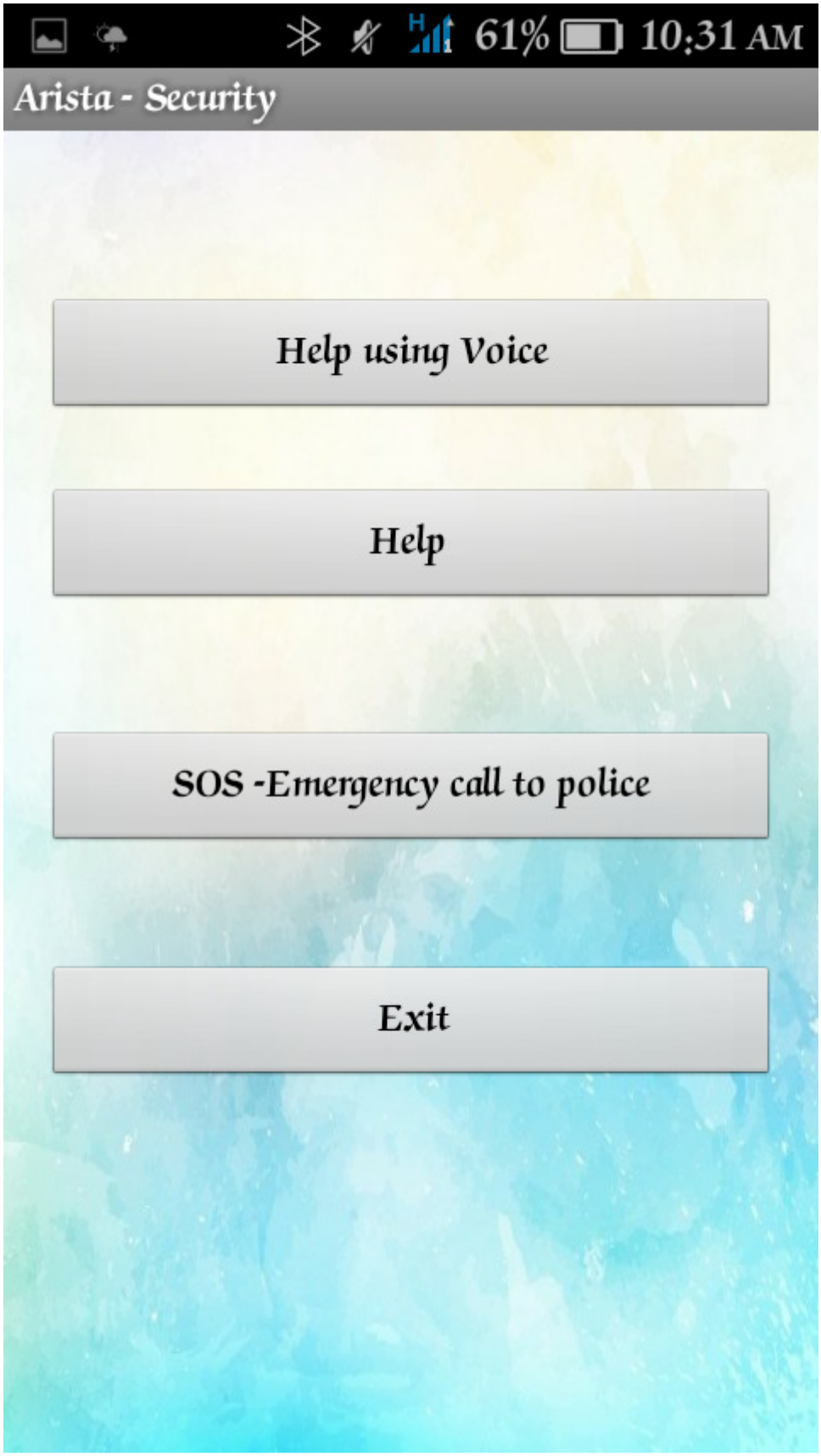
The SDK tools are installed with the SDK starter package and are periodically updated. The SDK tools are required if you are developing Android applications.

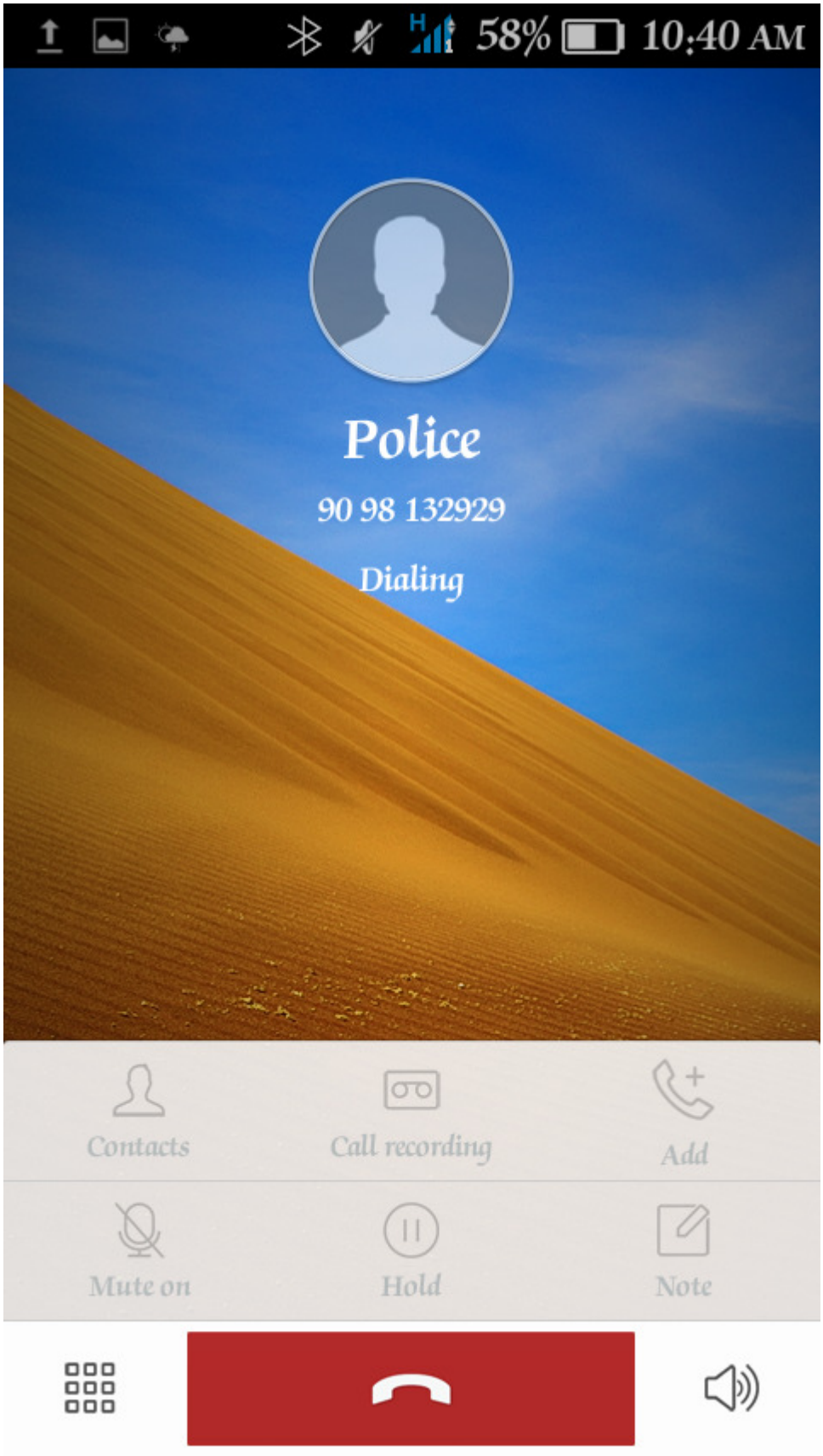
The most important SDK tools include the Android SDK Manager (android sdk), the AVD Manager (android avd) the emulator (emulator), and the Dalvik Debug Monitor Server (ddms). A short summary of some frequently-used SDK tools is provided below.

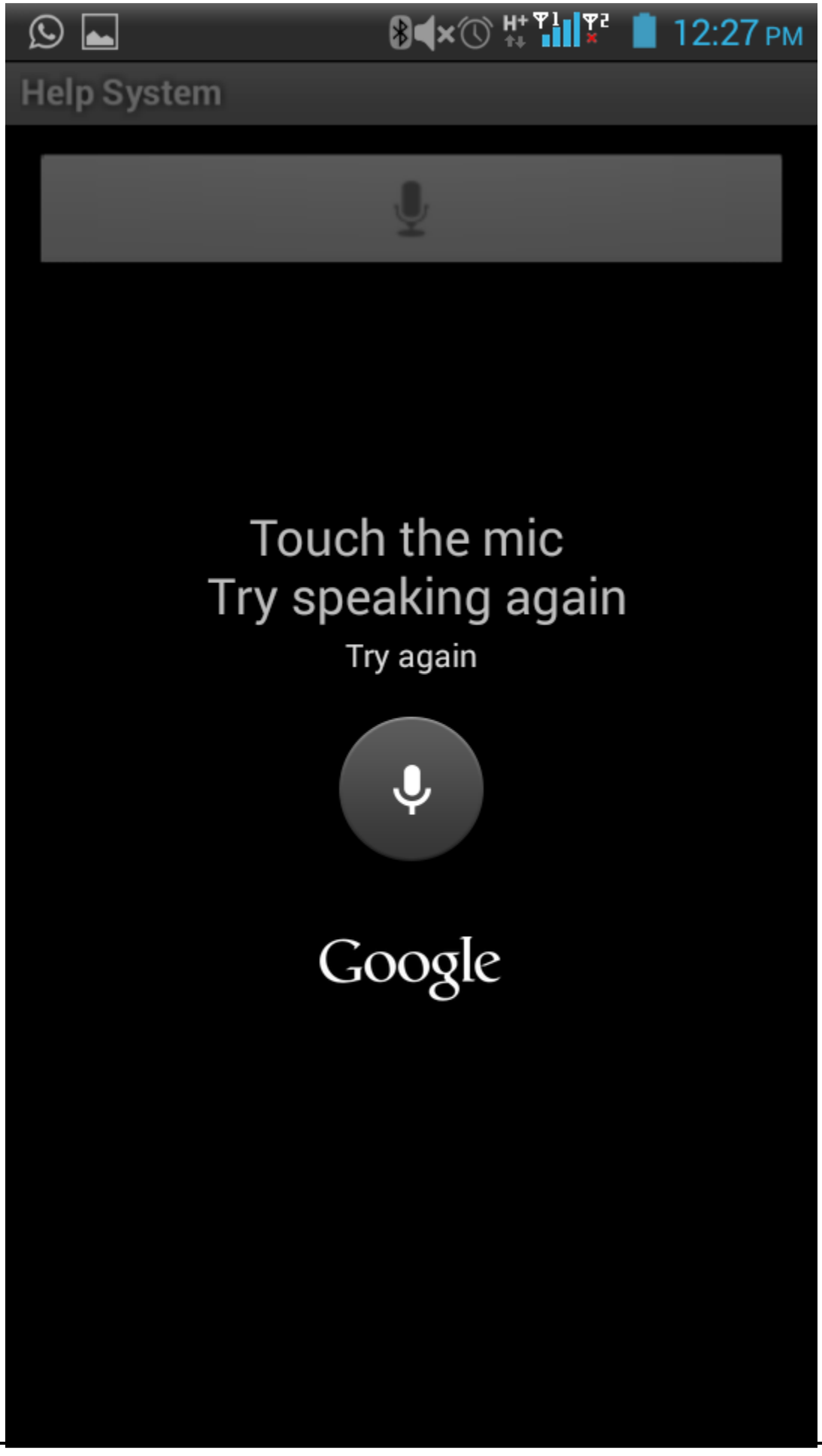
6.4 Screenshots with detail

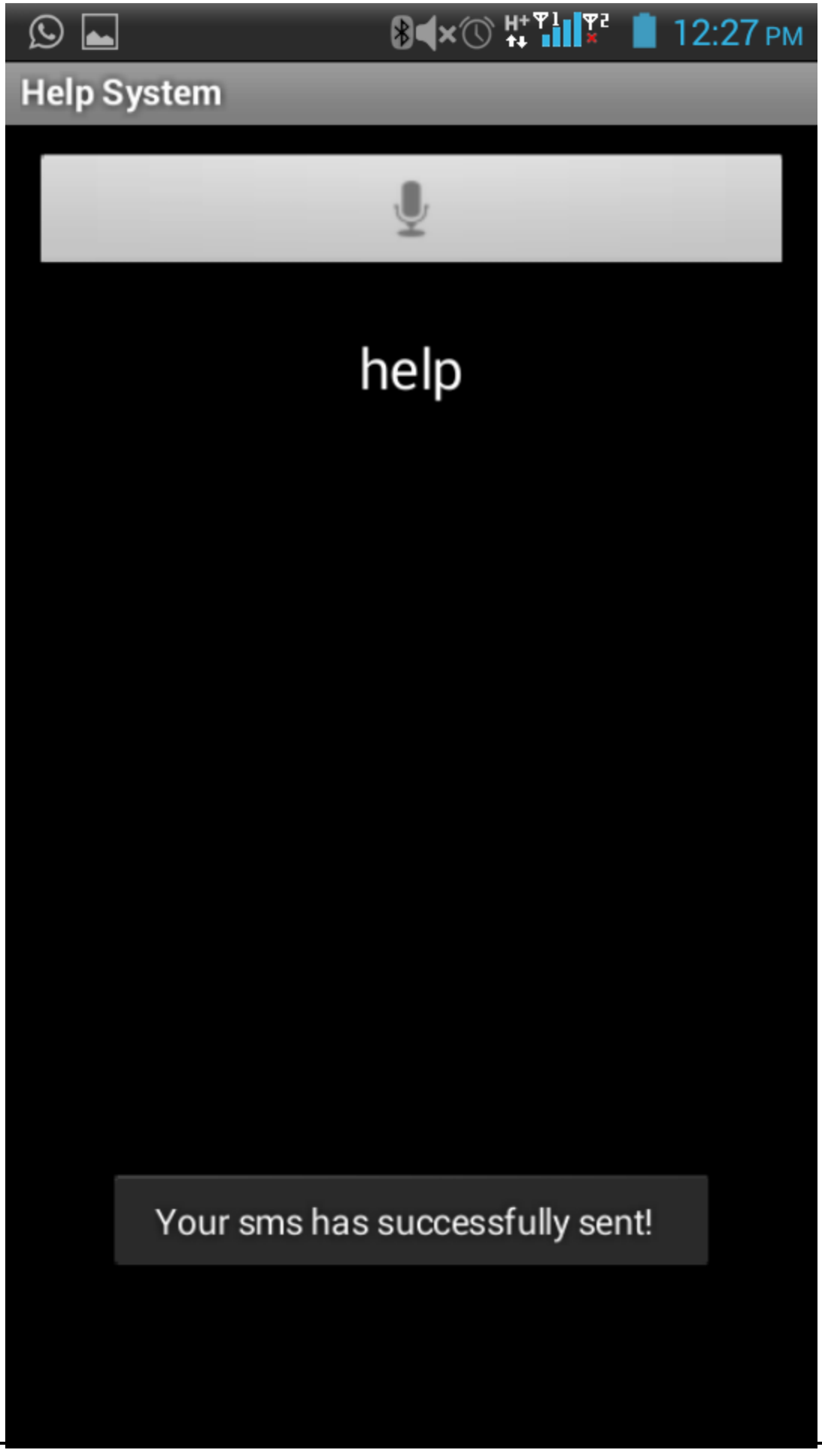
5554:AVD_for_Nexus_S_by_Google


61% 10:28 AM
Arista - Security
New Registration
Name of the Person
Your Mobile No
Buddy Name
Buddy Mobile 1
Buddy Mobile 2
police station no
Submit


61% 10:31 AM
Arista - Security
Help using Voice
Help
SOS -Emergency call to police
Exit


58%
10:40 AM
Police
90 98 132929
Dialing
Contacts
Call recording
Add
Mute on
Hold
Note


12:27 PM
Help System
Touch the mic
Try speaking again
Try again
Google


12:27 PM
Help System
help
Your sms has successfully sent!

6.5 Code

main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/back"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".NoticeToClassMainActivity" >

    <LinearLayout
        android:id="@+id/linearLayout1"
        android:layout_width="match_parent"
        android:layout_height="350dp"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:gravity="top|center"
        android:orientation="vertical" >

        <TextView
            android:id="@+id/textView1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="New Registration"
            android:textAppearance="?android:attr/textAppearanceMedium"
            android:textColor="#dc143c"
            android:textSize="20sp"
            android:textStyle="bold" />

         <TextView
        android:id="@+id/textView5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/textView2"
        android:layout_below="@+id/textView1"
        android:textAppearance="?android:attr/textAppearanceMedium" />

      <TextView
          android:id="@+id/textView2"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content" />

        <EditText
            android:id="@+id/name"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="@string/applname"
            android:digits="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
             />

          <EditText
```

```
            android:id="@+id/mobile"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="Your Mobile No"
            android:inputType="numberSigned" />

        <EditText
            android:id="@+id/bname"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:digits="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
            android:hint="Buddy Name" />

        <EditText
            android:id="@+id/bmobile"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="Buddy Mobile"
            android:inputType="numberSigned" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="match_parent" >

</LinearLayout>

    <Button
        android:id="@+id/button1"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:layout_alignLeft="@+id/linearLayout1"
        android:layout_below="@+id/linearLayout1"
        android:background="@drawable/redbutton"
        android:text="Submit"
        android:onClick="submitclick"  />

</RelativeLayout>
```

Start.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/back"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <ImageView
```

```
        android:id="@+id/imageView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@+id/button1"
        android:layout_alignLeft="@+id/imageView1"
        android:layout_marginBottom="15dp"
        android:src="@drawable/bus" />

    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:src="@drawable/logo" />

    <Button
        android:id="@+id/button1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/imageView2"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="16dp"
        android:background="@drawable/redbutton"
        android:text="Start >>>"
        android:textColor="#ffffff"
        android:textSize="16sp"
        android:textStyle="bold"
        android:onClick="startclick" />

</RelativeLayout>
```

Speak.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_above="@+id/textView1"
    android:layout_toLeftOf="@+id/textView1"
    android:gravity="center"
    android:orientation="vertical" >

    <Button
        android:id="@+id/btnSpeak"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_marginRight="10dp"
        android:layout_marginTop="10dp"

        android:text="Speak now"
    android:onClick="speakClick"
    />

    <TextView
```

```
        android:id="@+id/txtText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:layout_marginTop="10dp"
        android:textAppearance="?android:attr/textAppearanceLarge" />

</LinearLayout>
```

MainActivity.java

```
package a.b;


import android.app.Activity;

import static android.app.Activity.RESULT_OK;

import android.content.ActivityNotFoundException;

import android.content.Intent;

import android.os.Bundle;

import android.os.Handler;

import android.os.Message;

import android.speech.RecognizerIntent;

import android.telephony.SmsManager;

import android.view.View;


import java.io.*;

import android.widget.*;

import java.util.ArrayList;


public class MainActivity extends Activity {


    protected static final int RESULT_SPEECH = 1;

    /**
```

```
     * Called when the activity is first created.
     */

    TextView txtText;
    String bmobileno = null;
    GPSTracker gps;
String finalMsg="";
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.start);
    }

    public void startclick(View v)
    {
            try {

            File myFile = new File("/sdcard/userdata.txt");
            FileInputStream fIn = new FileInputStream(myFile);
            BufferedReader myReader = new BufferedReader(
                                new InputStreamReader(fIn));
            String aDataRow = "";
            String aBuffer = "";
            while ((aDataRow = myReader.readLine()) != null)
            {
               aBuffer += aDataRow ;
```

```
            }



            setContentView(R.layout.speak);
        txtText = (TextView) findViewById(R.id.txtText);
             Toast.makeText(this,aBuffer,Toast.LENGTH_LONG).show();
             String tempdata[] = aBuffer.split(";");
             bmobileno = tempdata[3];



            myReader.close();
           // Toast.makeText(this,"Done reading SD
'mysdfile.txt'",Toast.LENGTH_SHORT).show();
        }
        catch (Exception e)
        {
             setContentView(R.layout.main);
           // Toast.makeText(this, e.getMessage(),Toast.LENGTH_SHORT).show();
        }



    }

    public void submitclick(View v) {
        EditText name = (EditText) findViewById(R.id.name);
        EditText mobile = (EditText) findViewById(R.id.mobile);
        EditText bname = (EditText) findViewById(R.id.bname);
```

```
EditText bmobile = (EditText) findViewById(R.id.bmobile);
bmobileno = bmobile.getText().toString();

if(name.getText().toString().length()==0
        || mobile.getText().toString().length()==0
        || bname.getText().toString().length()==0
        || bmobile.getText().toString().length()==0)
{
    Toast.makeText(this, "Value cannot be empty",Toast.LENGTH_SHORT).show();
    return;
}



try {
    String firstmsg = "Hi " + bname.getText().toString()
            + "\n, i have added as a helping person";

    SmsManager smsManager = SmsManager.getDefault();
    smsManager.sendTextMessage(bmobile.getText().toString(),
            null, firstmsg, null, null);
    Toast.makeText(getApplicationContext(), "SMS Sent!",
            Toast.LENGTH_LONG).show();
} catch (Exception e) {
    Toast.makeText(getApplicationContext(),
            "SMS faild, please try again later!",
            Toast.LENGTH_LONG).show();
```

```
            e.printStackTrace();
        }

        String msg = name.getText().toString() + ";"
              + mobile.getText().toString() + ";"
              + bname.getText().toString() + ";"
              + bmobile.getText().toString() + ";";

        try {

            File myFile = new File("/sdcard/userdata.txt");
            myFile.createNewFile();
            FileOutputStream fOut = new FileOutputStream(myFile);
            OutputStreamWriter myOutWriter =new OutputStreamWriter(fOut);
            myOutWriter.append(msg);
            myOutWriter.close();
            fOut.close();
            Toast.makeText(v.getContext(),"Done writing SD 'mysdfile.txt'",
Toast.LENGTH_SHORT).show();

        } catch (Exception e) {
            Toast.makeText(this, e.getMessage(), Toast.LENGTH_LONG).show();
            e.printStackTrace();
        }

        setContentView(R.layout.speak);
        txtText = (TextView) findViewById(R.id.txtText);
```

```
}

public void speakClick(View v) {
    Intent intent = new Intent(
            RecognizerIntent.ACTION_RECOGNIZE_SPEECH);

    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
            "en-US");

    try {
        startActivityForResult(intent, RESULT_SPEECH);
        txtText.setText("");
    } catch (ActivityNotFoundException a) {
        Toast t = Toast.makeText(getApplicationContext(),
                "Ops! Your device doesn't support Speech to Text",
                Toast.LENGTH_SHORT);
        t.show();
    }
}

protected void onActivityResult(int requestCode, int resultCode,
        Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    switch (requestCode) {
        case RESULT_SPEECH: {
```

```
            if (resultCode == RESULT_OK && null != data) {

                ArrayList<String> text = data
                        .getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);

                txtText.setText(text.get(0));
                if (txtText.getText().equals("help")) {
                    helpSMS();
                }
                //  SendSMS(msg, mob);
            }
            break;
        }

    }
}

public void helpSMS() {

    gps = new GPSTracker(MainActivity.this);

    // check if GPS enabled
    String msg = null;
    if (gps.canGetLocation()) {

        double latitude = gps.getLatitude();
        double longitude = gps.getLongitude();
```

```
        // \n is for new line
        Toast.makeText(getApplicationContext(), "Your Location is - \nLat: " + latitude +
"\nLong: " + longitude, Toast.LENGTH_LONG).show();
        msg = "Your Location is - \nLat: " + latitude + "\nLong: " + longitude;

        String lat = String.valueOf(latitude);
        String lng = String.valueOf(longitude);

        String url = "http://maps.google.com/maps?&daddr="+
              lat+"," +lng;

        ////Intent intent = new Intent(android.content.Intent.ACTION_VIEW,
        //         Uri.parse(url));
        // startActivity(intent);

        try {
           String firstmsg = "i need help \n my location is " +url ;

           SmsManager smsManager = SmsManager.getDefault();
           smsManager.sendTextMessage(bmobileno,
                 null, firstmsg, null, null);
           Toast.makeText(getApplicationContext(), "SMS Sent!",
```

```
                Toast.LENGTH_LONG).show();
            } catch (Exception e) {
                Toast.makeText(getApplicationContext(),
                        "SMS faild, please try again later!",
                        Toast.LENGTH_LONG).show();
                e.printStackTrace();
            }
        } else {
                            // can't get location
            // GPS or Network is not enabled
            // Ask user to enable GPS/network in settings
            gps.showSettingsAlert();
        }
}
}
```

LocationAddress.java

```
package a.b;

import android.content.Context;
import android.location.Address;
import android.location.Geocoder;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.util.Log;
```

```
import java.io.IOException;
import java.util.List;
import java.util.Locale;

public class LocationAddress {
   private static final String TAG = "LocationAddress";

   public static void getAddressFromLocation(final double latitude, final double longitude,
                                             final Context context, final Handler handler) {
      Thread thread = new Thread() {
         @Override
         public void run() {
            Geocoder geocoder = new Geocoder(context, Locale.getDefault());
            String result = null;
            try {
               List<Address> addressList = geocoder.getFromLocation(
                     latitude, longitude, 1);
               if (addressList != null && addressList.size() > 0) {
                  Address address = addressList.get(0);
                  StringBuilder sb = new StringBuilder();
                  for (int i = 0; i < address.getMaxAddressLineIndex(); i++) {
                     sb.append(address.getAddressLine(i)).append("\n");
                  }
                  sb.append(address.getLocality()).append("\n");
                  sb.append(address.getPostalCode()).append("\n");
                  sb.append(address.getCountryName());
```

```
                result = sb.toString();
            }
        } catch (IOException e) {
            Log.e(TAG, "Unable connect to Geocoder", e);
        } finally {
            Message message = Message.obtain();
            message.setTarget(handler);
            if (result != null) {
                message.what = 1;
                Bundle bundle = new Bundle();
                result = "Latitude: " + latitude + " Longitude: " + longitude +
                        "\n\nAddress:\n" + result;
                bundle.putString("address", result);
                message.setData(bundle);
            } else {
                message.what = 1;
                Bundle bundle = new Bundle();
                result = "Latitude: " + latitude + " Longitude: " + longitude +
                        "\n Unable to get address for this lat-long.";
                bundle.putString("address", result);
                message.setData(bundle);
            }
            message.sendToTarget();
        }
    }
};
thread.start();
```

```
    }
}
```

End of Code

7 Testing

Software testing in an essential phase in the development life cycle of an application. Testing ensures that the developed system meets its functional and non-functional requirements. Two important terms in software testing are Verification and Validation. Verification is the process of evaluating work-products like requirement specs, design specs and test cases etc. of different development phases to make sure that they meet the requirements for that phase. It ensures that the system is built in the right way. Whereas Validation is the process of evaluating the software at the end of the development phase to make sure that it meets the business requirements. It is used to make sure that the product fulfills its intended use and that the end product is built right. In this chapter we mainly validate this Android app to make sure it meets the requirements set initially.

7.1 Test steps

Purpose

To determine the list of Test Cases that needs to be executed when assuring the system's quality. Each test case is described below in accordance with Test Case Specification.

Background

The purpose of project is fight against Brute force hacking that should have a feature of virtually locking a customer account by providing a new concept of authorization (for eg. Email), which can be easily implemented and used, have future scope and flexibility and will provide new way of Store At Your Service account’s security.

Scope

Test Cases document contains the list of test cases which has to be executed on the project. All the account users will get benefited.

Test Level (Functional Testing)

Functional testing is a quality assurance (QA) process and a type of black box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (not like in white-box testing). Functional Testing usually describes what the system does.

Functional testing differs from system testing in that functional testing "verifies a program by checking it against ... design document(s) or specification(s)", while system testing "validate[s] a program by checking it against the published user or system requirements" (Kaner, Falk, Nguyen 1999, p. 52).

Functional testing means testing the application against business requirements. Functional testing is executed using the functional specifications given by the client or by the design specifications according to use cases given by the design team. Role of functional testing is to validating the behavior of an application.

Functional testing is more important because it always verifies that your system is fixed for release. The functional tests define your working system in a useful manner. In functional testing tester has to validate the application to see that all specified requirements of the client whatever we have said in SRS or BRS have been incorporated or not.

Functional testing is always concentrating on customer requirements and whereas the Non Functional testing is always concentrating on customer expectations.

Functional test cases target business goals and Non Functional test cases target performance, resource utilization, usability, compatibility etc. Functional testing is a part of system testing.

Example of functional testing is explained below –

Considering example if you are functionally testing a word processing application, a partial list of checks you would perform minimally includes creating, saving, editing, spell checking and printing documents.

Types of Functional Testing

Functional testing falls in to two categories –

Positive functional testing

This testing carries exercising the application's functions with valid input and also verifying that the outputs are correct.

Example - Again continuing with the word processing example, a positive test for the printing function might be to print a document containing both text as well as graphics to a printer that is online, filled with paper and for which the correct drivers are installed.

Negative functional testing

This testing involves exercising application functionality using a combination of invalid inputs, some unexpected operating conditions and by some other "out-of-bounds" scenarios.

Example - Again continuing with the word processing example, a negative test for the printing function might be to disconnect the printer from the computers while a document is printing. What probably should happen in these scenarios are a plain-English error message displayed, informing the user what happened and instructing him/her on how to fix the problem.

Testing Conclusion

At last we conclude that in functional testing functionality of the module is tested and structure is not considered. It is performed, based on user's perspective. These tests ensure that the system does what users are expecting it to do.

This type of testing means testing the functionality which include input the proper data and checking the output as per the requirement documents.

7.2 Test case

MODULE 1 (Registration)

Module Name:	**Registration**
Dependencies:	Connection with Database

Screen Name:	Main Page/Form

Test Case#	Test Objective	Precondition	Test Steps	Test Data	Expected Result	Actual Result	Status
FT001	Successful display of the pages with registration information	User has started the App	User clicks on the choice "Start"	Button click	Show required page as per choice made	Done	Pass

MODULE 2 (Voice)

Module Name:	Voice/Speak Module
Dependencies:	Connection with Database
Screen Name:	Speak

Test Case#	Test Objective	Pre - condition	Test Steps	Test Data	Expected Result	Actual Result	Status
FT002	To recognize the voice of the user	_	Voice has to be matched, and display text	Display text	Found correct data of voice.	Successfully view the Text	Pass

MODULE 3 (Current Location /GPS)

Module Name:	GPS Module
Dependencies:	Connection with GPS/Netword
Screen Name:	GetCurrentLocation

Test Case#	Test Objective	Pre - condition	Test Steps	Test Data	Expected Result	Actual Result	Stat us
FT003	To find the GPS location of user	_	Gps location has to be found	Latitude Longitude	Found correct GPS location data.	Successfully view the GPS Location	Pass

MODULE 4 (SMS)

Module Name:	SMS Module
Dependencies:	Connection with Database
Screen Name:	SMS

Test Case#	Test Objective	Pre - condition	Test Steps	Test Data	Expected Result	Actual Result	Status
FT004	To send sms to buddy of the user	_	SMS send to successfully and Display map to buddy	Mobileno of buddy	Found correct Location of user for buddy	Successfully view MAP of user on buddy mobile	Pass

8. Database description

8.1 List of tables

Table name	Description
Registration	This table will contain the personal information of user.
Primary key	User_id

Table name	Description
UserLog	This table will contain the Login/Logout information of user.
Foreign keys	User_id

Table name	Description
UserHelpLog	This table will contain the gps location with sms information of user.
Foreign keys	User_id

8.2 Structure of table

Registration

FIELD NAME	DATA TYPE	CONSTRAINTS	COMMENTS
User_id	Int	PK	
First name	Varchar(50)	NN	
Lastname	Varchar(50)	NN	
Phone number	Varchar(50)	NN	
Buddy Name	Varchar(50)	NN	

Buddy Mobile No	Varchar(50)	NN	

UserLog

FIELD NAME	DATA TYPE	CONSTRAINTS	COMMENTS
User_id	Int	FK	
userlogindatetime	DateTime	NN	
UserLogoutDatetime	DateTime	NN	
UserLocationn	Varchar(50)	NN	

UserHelpLog

FIELD NAME	DATA TYPE	CONSTRAINTS	COMMENTS
User_id	Int	FK	
Latitude	Varchar(50)	NN	
Longitude	Varchar(50)	NN	
SMSText	Varchar(50)	NN	
SMSDeliverystatus	bit	NN	
Buddy Mobile No	Varchar(50)	NN	

9 Conclusion and Discussion

9.1 Conclusion and Discussion

The project "Women Security Women Security Android App" which has been developed using Net beans and Android as a front-end tool meets the requirement of the organization. The

problem of the women safety is increased rapidly in this environment, so I proposed as an effective Android application to prevent such type of the suspicious or natural disaster, by alerting the concern authorities using the android mobile phone which helps to stop such type of illegal activates and to trace the concern.

A low cost women tracking system using GPS and GPRS of GSM network, suitable for wide range of applications all over the world. The combination of the GPS and GPRS provides continuous and real time tracking. The cost is much lower compared to SMS based tracking systems. Free Google map and the use of HTTP protocol as data sending method reduces the monthly bundle cost for the individual user and also for the small business owner. It is expected that the full implementation of the proposed system would ultimately replace the traditional and costly SMS based tracking system

Developing this software has been a good experience for me. During the development of the project I had got enough to learn and got the chance to increase my knowledge in the field of software. I am satisfied that the system meets all the requirements.

Future Scope of the Project

It can be used to track each and every woman on earth there by having bymetric details alongwith there diseases they are or were suffering from.

This details can be futher used in any medical or emergency condition.

Snapshots

5554:AVD_for_Nexus_S_by_Google

Code

Start.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/back"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@+id/button1"
        android:layout_alignLeft="@+id/imageView1"
        android:layout_marginBottom="15dp"
        android:src="@drawable/bus" />

    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:src="@drawable/logo" />

    <Button
        android:id="@+id/button1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/imageView2"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="16dp"
        android:background="@drawable/redbutton"
        android:text="Start >>>"
        android:textColor="#ffffff"
        android:textSize="16sp"
        android:textStyle="bold"
        android:onClick="startclick" />

</RelativeLayout>
```

Main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/back"
```

```
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context=".NoticeToClassMainActivity" >

<LinearLayout
    android:id="@+id/linearLayout1"
    android:layout_width="match_parent"
    android:layout_height="350dp"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:gravity="top|center"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Registration"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textColor="#dc143c"
        android:textSize="20sp"
        android:textStyle="bold" />

     <TextView
    android:id="@+id/textView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textView2"
    android:layout_below="@+id/textView1"
    android:textAppearance="?android:attr/textAppearanceMedium" />

  <TextView
      android:id="@+id/textView2"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content" />

    <EditText
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="@string/applname"
        android:digits="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
         />

      <EditText
          android:id="@+id/mobile"
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:ems="10"
          android:hint="Your Mobile No"
          android:inputType="numberSigned" />

    <EditText
        android:id="@+id/bname"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```
            android:ems="10"
            android:digits="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
            android:hint="Buddy Name" />

        <EditText
            android:id="@+id/bmobile"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="Buddy Mobile"
            android:inputType="numberSigned" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="match_parent" >

</LinearLayout>

    <Button
        android:id="@+id/button1"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:layout_alignLeft="@+id/linearLayout1"
        android:layout_below="@+id/linearLayout1"
        android:background="@drawable/redbutton"
        android:text="Submit"
        android:onClick="submitclick"  />

</RelativeLayout>
```

MainActivity.java

```
package a.b;



import android.app.Activity;

import static android.app.Activity.RESULT_OK;

import android.content.ActivityNotFoundException;

import android.content.Intent;

import android.os.Bundle;

import android.os.Handler;

import android.os.Message;

import android.speech.RecognizerIntent;

import android.telephony.SmsManager;
```

```
import android.view.View;


import java.io.*;
import android.widget.*;
import java.util.ArrayList;


public class MainActivity extends Activity {


   protected static final int RESULT_SPEECH = 1;
   /**
    * Called when the activity is first created.
    */


   TextView txtText;
   String bmobileno = null;
   GPSTracker gps;
String finalMsg="";
   @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);


      setContentView(R.layout.start);
   }


   public void startclick(View v)
   {
         try {
```

```
   File myFile = new File("/sdcard/userdata.txt");
   FileInputStream fIn = new FileInputStream(myFile);
   BufferedReader myReader = new BufferedReader(
                new InputStreamReader(fIn));
   String aDataRow = "";
   String aBuffer = "";
   while ((aDataRow = myReader.readLine()) != null)
   {
      aBuffer += aDataRow ;
   }



   setContentView(R.layout.speak);
txtText = (TextView) findViewById(R.id.txtText);
    Toast.makeText(this,aBuffer,Toast.LENGTH_LONG).show();
    String tempdata[] = aBuffer.split(";");
    bmobileno = tempdata[3];



   myReader.close();
  // Toast.makeText(this,"Done reading SD 'mysdfile.txt'",Toast.LENGTH_SHORT).show();
}
catch (Exception e)
{
    setContentView(R.layout.main);
  // Toast.makeText(this, e.getMessage(),Toast.LENGTH_SHORT).show();
```

```
    }



}

public void submitclick(View v) {
   EditText name = (EditText) findViewById(R.id.name);
   EditText mobile = (EditText) findViewById(R.id.mobile);
   EditText bname = (EditText) findViewById(R.id.bname);
   EditText bmobile = (EditText) findViewById(R.id.bmobile);
   bmobileno = bmobile.getText().toString();

   if(name.getText().toString().length()==0
         || mobile.getText().toString().length()==0
         || bname.getText().toString().length()==0
         || bmobile.getText().toString().length()==0)
   {
      Toast.makeText(this, "Value cannot be empty",Toast.LENGTH_SHORT).show();
      return;
   }



   try {
      String firstmsg = "Hi " + bname.getText().toString()
            + "\n, i have added as a helping person";
```

```
    SmsManager smsManager = SmsManager.getDefault();
    smsManager.sendTextMessage(bmobile.getText().toString(),
            null, firstmsg, null, null);
    Toast.makeText(getApplicationContext(), "SMS Sent!",
            Toast.LENGTH_LONG).show();
} catch (Exception e) {
    Toast.makeText(getApplicationContext(),
            "SMS faild, please try again later!",
            Toast.LENGTH_LONG).show();
    e.printStackTrace();
}

String msg = name.getText().toString() + ";"
        + mobile.getText().toString() + ";"
        + bname.getText().toString() + ";"
        + bmobile.getText().toString() + ";";

try {

    File myFile = new File("/sdcard/userdata.txt");
    myFile.createNewFile();
    FileOutputStream fOut = new FileOutputStream(myFile);
    OutputStreamWriter myOutWriter =new OutputStreamWriter(fOut);
    myOutWriter.append(msg);
    myOutWriter.close();
    fOut.close();
    Toast.makeText(v.getContext(),"Done writing SD 'mysdfile.txt'", Toast.LENGTH_SHORT).show();
```

```
    } catch (Exception e) {
        Toast.makeText(this, e.getMessage(), Toast.LENGTH_LONG).show();
        e.printStackTrace();
    }

    setContentView(R.layout.speak);
    txtText = (TextView) findViewById(R.id.txtText);

}

public void speakClick(View v) {
    Intent intent = new Intent(
            RecognizerIntent.ACTION_RECOGNIZE_SPEECH);

    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
            "en-US");

    try {
        startActivityForResult(intent, RESULT_SPEECH);
        txtText.setText("");
    } catch (ActivityNotFoundException a) {
        Toast t = Toast.makeText(getApplicationContext(),
                "Ops! Your device doesn't support Speech to Text",
                Toast.LENGTH_SHORT);
        t.show();
    }
```

```
}

protected void onActivityResult(int requestCode, int resultCode,
      Intent data) {
   super.onActivityResult(requestCode, resultCode, data);

   switch (requestCode) {
      case RESULT_SPEECH: {
         if (resultCode == RESULT_OK && null != data) {

            ArrayList<String> text = data
                  .getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);

            txtText.setText(text.get(0));
            if (txtText.getText().equals("help")) {
               helpSMS();
            }
            //  SendSMS(msg, mob);
         }
         break;
      }

   }
}

public void helpSMS() {
```

```
gps = new GPSTracker(MainActivity.this);


// check if GPS enabled
String msg = null;
if (gps.canGetLocation()) {


   double latitude = gps.getLatitude();
   double longitude = gps.getLongitude();









   // \n is for new line
   Toast.makeText(getApplicationContext(), "Your Location is - \nLat: " + latitude + "\nLong: " +
longitude, Toast.LENGTH_LONG).show();
   msg = "Your Location is - \nLat: " + latitude + "\nLong: " + longitude;


   String lat = String.valueOf(latitude);
   String lng = String.valueOf(longitude);


   String url = "http://maps.google.com/maps?&daddr="+
         lat+"," +lng;


   ////Intent intent = new Intent(android.content.Intent.ACTION_VIEW,
   //       Uri.parse(url));
   // startActivity(intent);
```

```
            try {
                String firstmsg = "i need help \n my location is " +url ;


                SmsManager smsManager = SmsManager.getDefault();
                smsManager.sendTextMessage(bmobileno,
                        null, firstmsg, null, null);
                Toast.makeText(getApplicationContext(), "SMS Sent!",
                        Toast.LENGTH_LONG).show();
            } catch (Exception e) {
                Toast.makeText(getApplicationContext(),
                        "SMS faild, please try again later!",
                        Toast.LENGTH_LONG).show();
                e.printStackTrace();
            }




        } else {
                                // can't get location
            // GPS or Network is not enabled
            // Ask user to enable GPS/network in settings
            gps.showSettingsAlert();
        }


}
```

```
}
```

Speak.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_above="@+id/textView1"
    android:layout_toLeftOf="@+id/textView1"
    android:gravity="center"
    android:orientation="vertical" >

    <Button
        android:id="@+id/btnSpeak"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_marginRight="10dp"
        android:layout_marginTop="10dp"

        android:text="Speak now"
    android:onClick="speakClick"
    />

    <TextView
        android:id="@+id/txtText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:layout_marginTop="10dp"
        android:textAppearance="?android:attr/textAppearanceLarge" />

</LinearLayout>
```

LocationAddress.java

```
package a.b;


import android.content.Context;

import android.location.Address;

import android.location.Geocoder;

import android.os.Bundle;

import android.os.Handler;
```

```
import android.os.Message;
import android.util.Log;

import java.io.IOException;
import java.util.List;
import java.util.Locale;

public class LocationAddress {
   private static final String TAG = "LocationAddress";

   public static void getAddressFromLocation(final double latitude, final double longitude,
                                 final Context context, final Handler handler) {
      Thread thread = new Thread() {
         @Override
         public void run() {
            Geocoder geocoder = new Geocoder(context, Locale.getDefault());
            String result = null;
            try {
               List<Address> addressList = geocoder.getFromLocation(
                     latitude, longitude, 1);
               if (addressList != null && addressList.size() > 0) {
                  Address address = addressList.get(0);
                  StringBuilder sb = new StringBuilder();
                  for (int i = 0; i < address.getMaxAddressLineIndex(); i++) {
                     sb.append(address.getAddressLine(i)).append("\n");
                  }
                  sb.append(address.getLocality()).append("\n");
```

```
            sb.append(address.getPostalCode()).append("\n");
            sb.append(address.getCountryName());
            result = sb.toString();
        }
    } catch (IOException e) {
        Log.e(TAG, "Unable connect to Geocoder", e);
    } finally {
        Message message = Message.obtain();
        message.setTarget(handler);
        if (result != null) {
            message.what = 1;
            Bundle bundle = new Bundle();
            result = "Latitude: " + latitude + " Longitude: " + longitude +
                    "\n\nAddress:\n" + result;
            bundle.putString("address", result);
            message.setData(bundle);
        } else {
            message.what = 1;
            Bundle bundle = new Bundle();
            result = "Latitude: " + latitude + " Longitude: " + longitude +
                    "\n Unable to get address for this lat-long.";
            bundle.putString("address", result);
            message.setData(bundle);
        }
        message.sendToTarget();
    }
}
```

```
        };
        thread.start();
    }
}
```

References

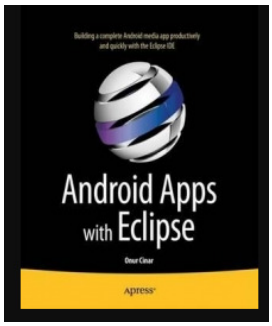
The various resources from where we have taken information to complete this project.

1. Wikipedia for the general aspects presented in the application, issues like SOS and its use, safety measures while moving.
2. https://www.Youtube.com for the videos on safety measurements/Women Security.
3. Google Maps to locate and identify the current location of a user.
4. A developed system to choose location and stations and provide the measurements.
5. Android tutorials on https://developer.android.com .

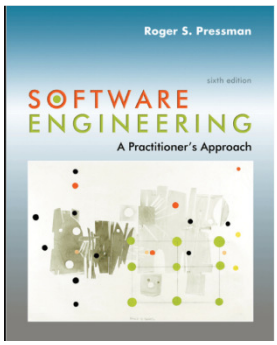
Bibliography

1. Android & Ios Development

2. Android App with Eclipse



3. Mc. Graw hill Software engineering and design Pressman



4.. Programming Android - ORelly