



Project Report on
“Safety Over SMS”

Submitted By

Dhaval Shah [1698028]

Khushil Shah [1698030]

Heet Vora [1698042]

Guided By

Mr. Pankaj Rathod

Term – December’18 to April’19

Department of Information Technology

SVKM’S

SHRI BHAGUBHAI MAFATLAL POLYTECHNIC

Irla, N.R.G.Marg, Vile Parle (West), Mumbai-400056

ACKNOWLEDGEMENT

Through this acknowledgement, our group exercises sincere gratitude to all people who have been associated with our project and helped us making it a wonderful journey.

*Our group owes immense gratitude to our institution **Shri Bhagubhai Mafatlal Polytechnic***

*and its entire involved staff. Foremost, a special thanks to our Head of Department **Mr. Abhijit Dongaonkar** and our vice principle **Mrs. Neeta Kadukar** for encouraging us to do this project.*

*We also give special thanks to **Mr. Pankaj Rathod** for guiding us throughout the journey not only like a mentor but also like a friend. He always empowered us and also praised us to do our project unique and perfect. His support helped us in qualitative development of our own skills and furthermore in project completion.*

We also thank Mr. Awadheskuhmar Pal for giving a helping hand in all types of technical problems and installation difficulties faced by us throughout the journey.

-By Group Members



SAFETY OVER SMS

ABSTRACT

Currently smartphone and other mobile devices have become incredibly significant in every face of our life. They are practically offering same abilities as desktop work-stations and are also powerful in terms of CPU, storage space and installing several applications. Therefore, security is deliberated as a vital factor in wireless communication technologies. Furthermore, based on growing range of mobile applications within variety of platforms, security is regarded as one of the most valuable and substantial debate in terms of issues, consistencies and accurateness. The worst security threat is that when your device has been lost or stolen. This project aims to introduce an offline android application wherein the mobile phones which get lost by mistake or stolen by thief, then the application will help to get the device back by performing various operations on remote mobile devices. The approach will be to send offline messages from the genuine user with help of android devices to the one which is stolen. This developed application will be installed on any android device. The user of the app will set the verification code on his device. This code is the key which the user need to type along with the contact name to get recognized. The app always runs in background waiting for the incoming message.

Table of Contents

Chapter 1	Introduction.....	1
1.1	Project Overview	3
Chapter 2	Problem Statement and Proposed System.....	4
2.1	Problem Statement	5
Chapter 3	Literature Review.....	8
3.1	Existing System	9
3.2	Literature Survey	10
Chapter 4	Design and Implementation.....	12
4.1	System Architecture	13
4.2	System Requirements	27
4.3	Implementation	28
Chapter 5	Results and Conclusion	70
5.1	Test Cases	71
5.2	Limitations	79
5.3	Future Scope	80
5.4	Conclusion	81
Chapter 7	Bibliography.....	82

Table of Figures

Figure 4.1	System Architecture	13
Figure 4.2	Flowchart for requesting permission	23
Figure 4.3	Use Case diagram	24
Figure 4.4	Sequence diagram	25
Figure 4.5	Activity diagram	13
Table 5.1	Test case	78

INTRODUCTION



CHAPTER 1: INTRODUCTION

Considering today's modern life in which mobile Phones have become an inseparable part of everyone's life. As android mobile phones are becoming popular both because of their cost and ability to offer numerous applications, hence this project commences application in Android. The user of the proposed system will set the verification code by him. This feature is the key which the user needs to type along with the command in order to get recognized that the message sent is intended for this proposed system by a legitimate user. The proposed system will always run in the background without draining battery waiting for the incoming messages. Once message is received by the user phone, the app will be triggered for analysis of the message. The app will then [2] perform various steps of text reorganization and will finally come to the conclusion whether the message is normal one or its intended for the app. If the message is intended for the app then the further processing will take place. The app will then verify the code and compare the function/module triggered. The app will then run to complete the process requested. If the user forgets the pin, he/she can reset the pin with the help of security question which is set by the user itself during the installation

1.1 PROJECT OVERVIEW

The first Android-powered phone was sold in October 2008, and by the end of 2011 Android had become the [18] world's leading smart phone platform with about more than 70,000 apps available to users for free. Millions of people already use Android because it makes mobile device so much more powerful and useful. User purpose is fulfilled to get contact, Get IMEI, Trace Location [7], Lock Mobile, Ring, Send SMS to intended receiver with auto-SMS reply. Once user lose their mobile it is very less possible to have your mobile connected to the internet. This app now has redefined the Old process and has brought all the features are offline. Features of this app is start recording, send SMS automatically to an intended receiver, get all phone contacts, remote contact finder, trace the location [8], get the IMEI information, lock your phone, automatic phone ringing even when the phone is on Silent. The paper reviews various Android applications related to SMS like 'SMS auto reply' which is an auto reply application that sends reply to each incoming SMS. This feature of auto-reply has been implemented in our application.



PROBLEM DOMAIN

CHAPTER 2: PROBLEM STATEMENT AND PROPOSED SYSTEM

2.1 PROBLEM STATEMENT

The proposed system has redefined the existing system and has brought all the features offline controlled through SMS. The user will be able to retrieve contacts, location, IMEI, record audio, send a message to an intended receiver, ring as well as lock the mobile phone; all of these just by sending a message consisting an appropriate command with and the verification PIN on your lost or stolen phone. This application consists of total 7 modules which are as follows: -

➤ Retrieval of Contact:

This module is used for getting the important contact number. This method consists of the command “**SOSCon**”, PIN and the name of the contact. This method finds out the given contact, after finding them this method sends contact numbers along with name.

➤ Retrieve Location:

This module gives back accurate current location of lost or stolen device. The command used for this module is “**SOSLocate**” and the PIN. A link will be returned to the requested mobile device which will open in Google Maps and gives the precise location.

➤ Retrieve IMEI:

IMEI is most important factor to get back device if it is lost or stolen. This module gives back the IMEI information of the device. This number can be further used for the tracking the lost or stolen device. The command used here is “**SOSImei**” and the PIN. While complaining about lost or stolen device in authorized police station or cyber-crime station, the most important factor is that they ask for the IMEI number in order to track the device. Whenever any such complaint is registered, most of the times the user doesn’t know about the IMEI number of his/her device, so at that time this command helps user a lot!

➤ Record Audio:

This module helps you to record an audio secretly anytime you want. The command used is “**SOSRecord**”,PIN and the duration in seconds. This command starts recording the audio for the requested duration. The recorded audio is saved on your device.

➤ Send Message to an Intended Receiver:

This module helps you send a message to an intended receiver. The command used is “**SOSSms**”,PIN, name of the contact and the message. With the command, you need to send the message as a string parameter and the contact to whom you want to send this message. With the help of this you can send message to anyone in your contact list without the need of retrieving a contact.

➤ Ring Device:

This module causes an alarm ring on the device. Even if the device is on silent mode, this alarm rings in full volume. The command used is “**SOSRing**” and the PIN. This command changes the previously set sound values to maximum. This produces a siren voice to identify location and get the device back if it is near the user.

➤ Lock Device:

This module performs locking operation on that stolen or lost device. While performing lock operation user sends password along with the command “**SOSLock**”. This password consists of 0-9 integer numbers. After executing this command, the device gets completely locked and unauthorized user cannot access that device. After entering password that matches with the one which is sent through message, the device gets unlocked. The basic application of this command is to protect device data from unauthorized user and also to stop the user from making modifications in system settings like uninstalling some apps, changing the settings etc.

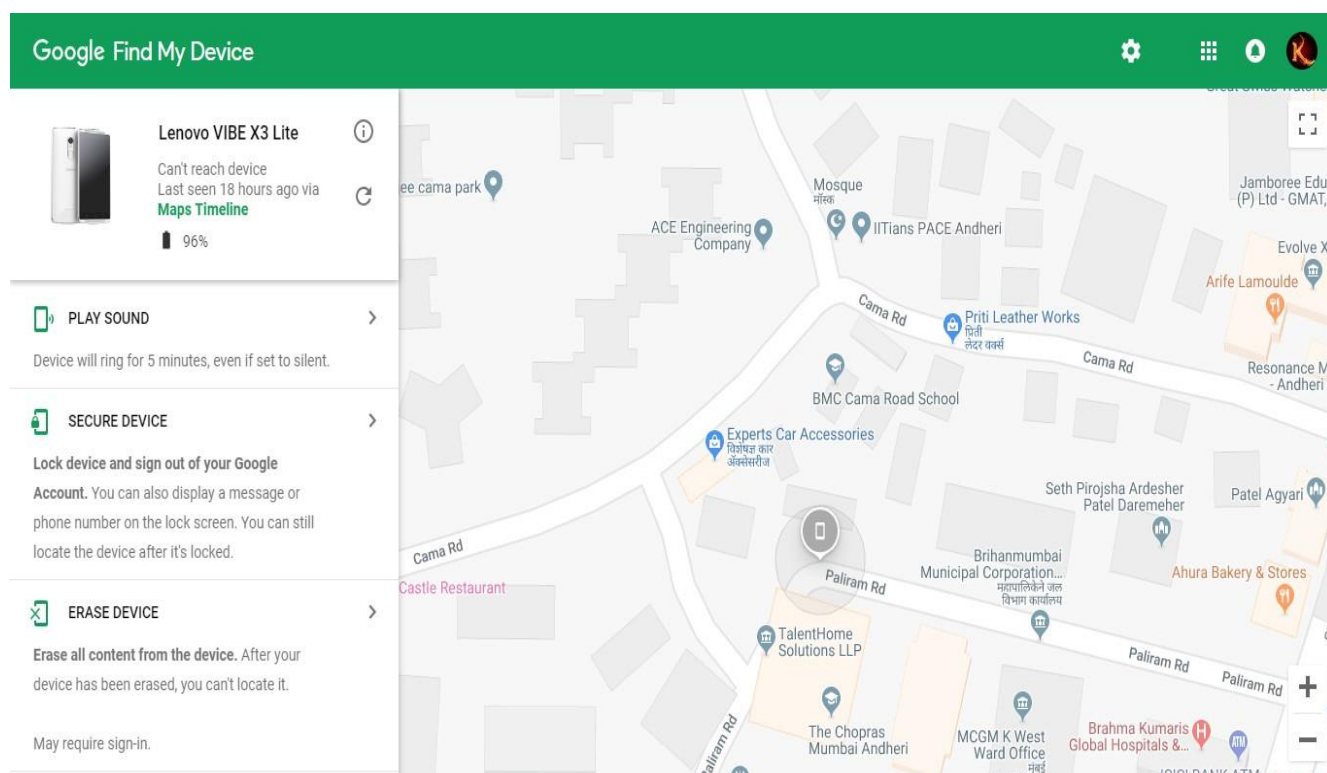
LIERATURE SURVEY



CHAPTER 3: LITERATURE SURVEY

3.1 EXISTING SYSTEM

Android Device Manager is Google's tracking tool for Android device owners. If you're logged into your device with your Google account, you should be able to locate it or take more extreme measures and wipe all data off it. If your Android device is synced with your Google account, the odds of recovering missing contacts are in your favour. As you can back up your data such as contacts from Android to your Google Account, you can of course restore the data from it. But you need to make sure that "Back up and sync" is on so that it would backup data automatically to Google Account.



One of the major limitations of this existing system is that it requires internet to perform any function. **Safety over SMS** provides offline security to the users without the need of Internet just with the help of SMS.

3.2 LITERATURE SURVEY

- Gargi Nigam , Prity Singh, Priyanshi Agarwal, (March 2014),
 - Smart Phone Remote Lock System

As enabling all their users to receive benefits and convenience, the smartphone have been applied in a variety range of fields, and it broadens a range of security threat. Specially, security threat of the smartphone by loss or stolen may cause the users' private data disclosure such as contacts, SMS message, photos, smart car keys, credit cards, login IDs, and so on. To prevent these problems, network operators should provide a certain method not to use mobile devices by bad guys and thus need to support the remote lock and wipe services which delete users' data as in the state of factory reset. By this paper, we propose the remote lock and wipe system using SMS push notification with integrity checking of the commands without sacrificing security level. To this end, we employ password-based key derived function (PBKDF) in PKCS5 which re-quires users to put only a password in and outputs 20 bytes long authentication code. The basic idea in our proposal is that Hash-based Message Authentication Code (HMAC) for authenticating SMS command message and the secret key to HMAC is derived from the passwords keyed in from the user. The HMAC is based on SHA-1 algorithm

- Shreya Patil, Bhawana Sarode, P.D.Chowhan,
 - Detection of Lost Mobile on Android Platform

One of the most interesting things about cell phone is that it is really a radio an extremely sophisticated radio, which uses some band of frequency that has the basic working similar to the ordinary cordless phone. The mobile cellular communication has been appreciated since its birth in the early 70's and the advancement in the field of VLSI has helped in designing less power, smaller size but efficient transceiver for the purpose of communication. But however the technology has not yet answered the loss or misplacement of the lost mobile phone which is significantly increasing. In this paper we discuss the problem and the probable solution that could be done. The IMEI number is a unique number that is embedded in the mobile phone. The main purpose of which is the blocking of calls that is made by unauthorized person

once the mobile is reported as stolen but here we use it effectively for the purpose of detection of lost mobile.

- Shubhangini Parmar, Ashwini Suryatale, Pooja Patil (1-3-2015)
 - Dynamic Web Based Mobile (Android) Application (Traffic System)

In recent years, smart phones are becoming more and more popular. So the data can be collected through it and all the record can be kept in database. This reduces the manual work and error and previous data can be fetched in future for reference. In this paper, we propose a web application where IMEI number is used for registration of a new user. The user will be created from the web application and user name and password will be assigned to his/her IMEI number. We are also transferring the images from android phone to web application. We will be generating reports which will show the daily collections of police as well as faulty drivers and will also plot the network on Google Maps using the data sent from android phone. By this paper, with location positioning system such as GPS becoming popular, there is a growing demand for location-based applications. It is easier, these days to utilize map information by connecting GPS receiver to PC and PDA Corresponding to this momentum, GPS receivers are now embedded into mobile phones and applications using the location of the user in real-time are widely available. GPS chips are now included in many devices to analyse satellite signals and determine the user's location with high accuracy.



IMPLEMENTATION

CHAPTER 4: DESIGN AND IMPLEMENTATION

4.1 SYSTEM ARCHITECTURE

The System Architecture of the proposed system is shown in Figure 3.1. The architecture of the proposed system is divided into 3 main levels: -

1. Application UI (User Interface)
2. Background processing
3. Application data storage

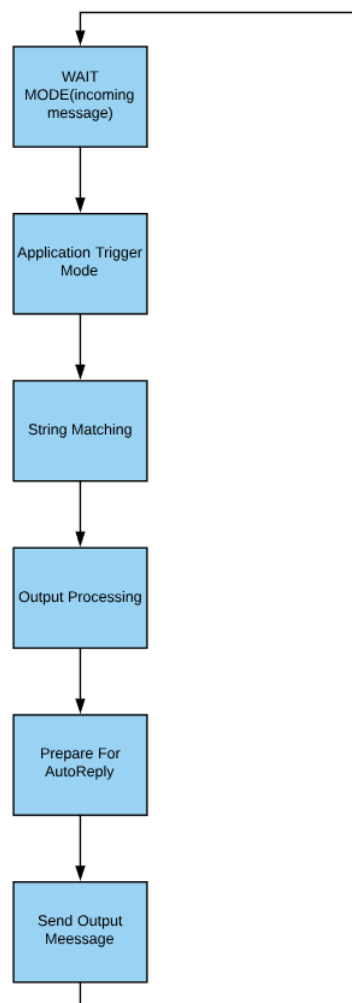


Figure 4.1: SYSTEM ARCHITECTURE

4.1.1 APPLICATION UI

The user interface (UI) is everything designed into an information device with which person may interact. When application is installed it contains default password "0000". User needs to enter default password in beginning for log in purpose. After logging in user have to fill some pre-requisites as these pre-requisites play important role for getting back lost/stolen device. UI also contains help button where all commands of modules are written. Additional information like how to use it is also given inside help button. Following are the pre-requisites:

- Changing the default password/pin

4.1.1.1 Changing the default password

Initial step after installing app is to set password given by the user. Change PIN button is present in UI, when user clicks on this button two text boxes are present in which new password is supposed to be entered twice. This password should be 4 digits numeric only, as it is going to use before input of every command to check whether that message is from an authenticate user. After confirming password app reads command followed by pin and then performs given task accordingly. If the user forgets the password, he/she can change the password with the help the security question which is set during the installation of the app.

2:01 PM 4G LTE 3G 32%

SOS

Security Question: My pet name

Answer: Tom

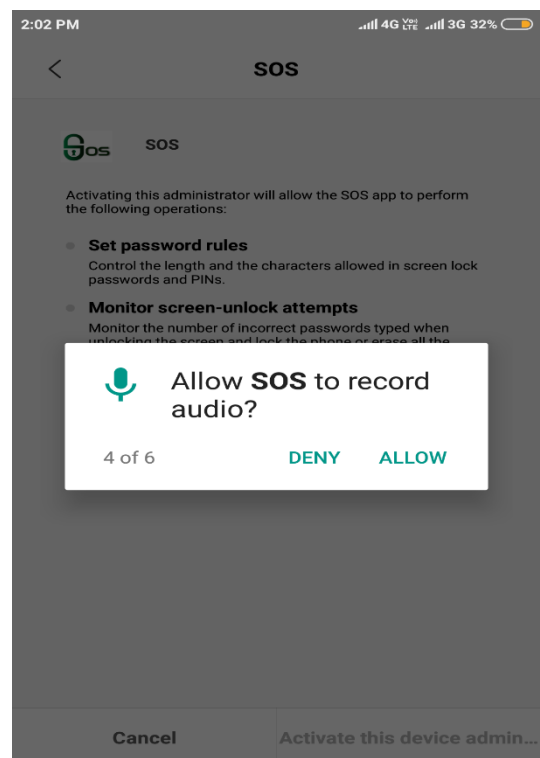
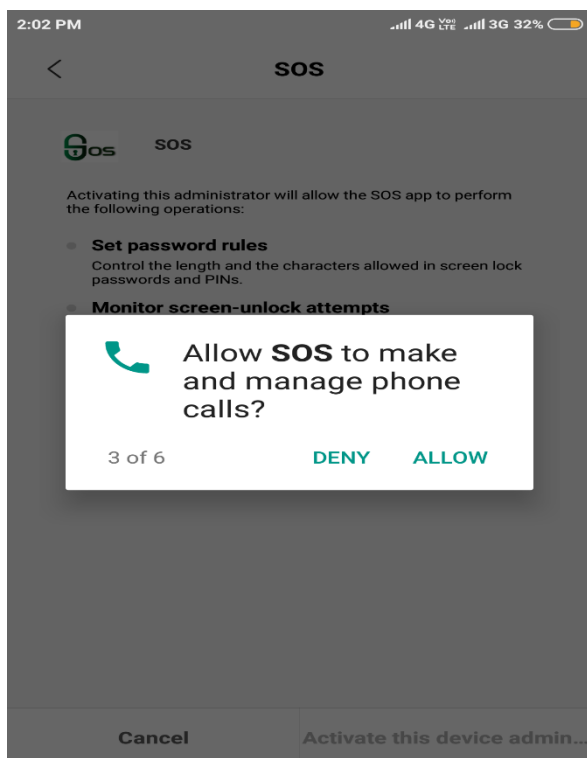
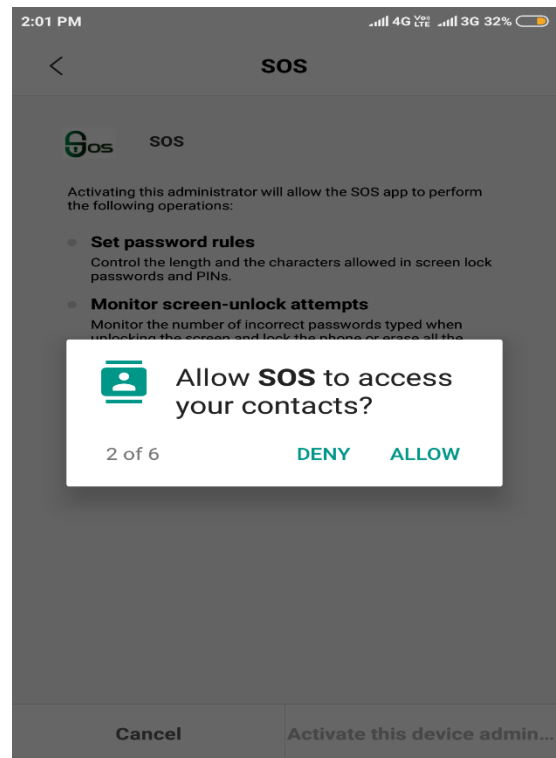
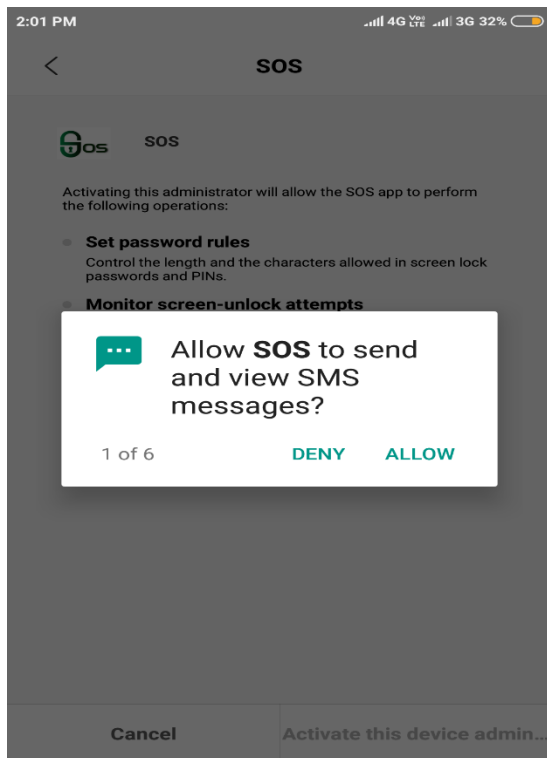
NEXT

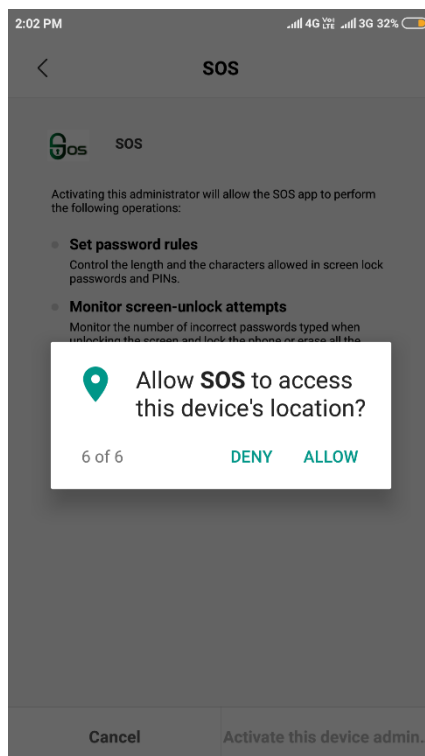
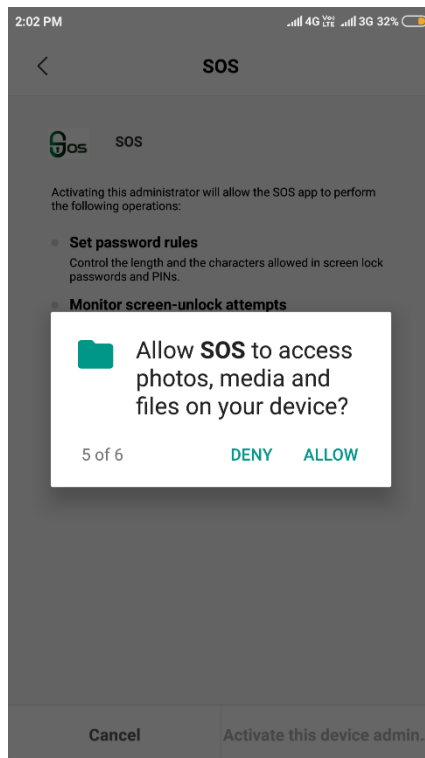
2:01 PM 4G LTE 3G 32%

SOS

PIN (4-Digit) 9998

NEXT





2:02 PM

4G VoLTE 3G 32%



SOS



SOS

Activating this administrator will allow the SOS app to perform the following operations:

- **Set password rules**
Control the length and the characters allowed in screen lock passwords and PINs.
- **Monitor screen-unlock attempts**
Monitor the number of incorrect passwords typed when unlocking the screen and lock the phone or erase all the phone's data if too many incorrect passwords are typed.
- **Lock the screen**
Control how and when the screen locks.

Cancel

Activate this device admin...

2:02 PM

4G VoLTE 3G 32%

SOS



PIN : 9998

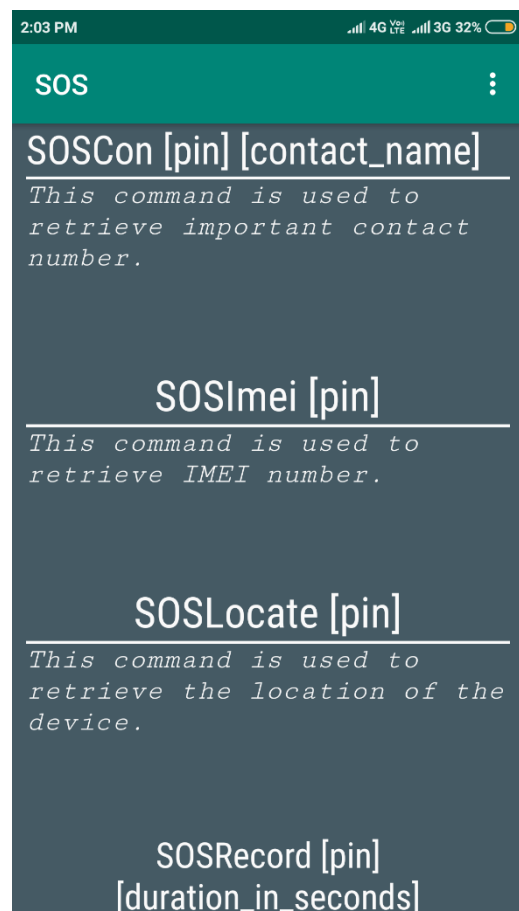
CHANGE PIN

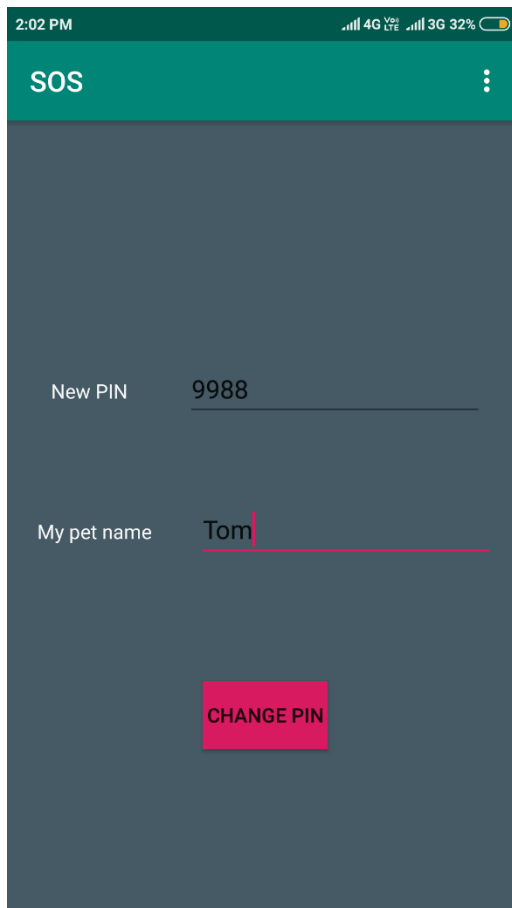


ABOUT US



HELP





4.1.2 BACKGROUND PROCESSING

Background processing contains background working of application. location module requires GPS latitude and longitude values. Get contact module performs searching operation on contact list to find out require results. lock module requires special kernel permissions to lock device. Ring the phone even if it is on silent. Record the audio for the requested seconds. Fig.4.1 shows the system architecture and background processing of the application.

In waiting mode application will wait for incoming message. As any message arrives the application will get triggered and it will check for message is intended for or application not by verification of password. If the password is matched by string matching, then application will read next string or command. If string matching fails, then application will again go to waiting state. String matched message will perform command given by the user. If user requires some data then application will fetch out data from device as device has granted permissions to do so, else user can perform some of the operations on device. User requested data is being prepared in output analysis. Auto reply function provides required data in particular pattern. This output message is again sent on that number from which user have requested data. If password and command do not match then application will not perform any task.

4.1.3 APPLICATION DATA STORAGE

This application requires database storage for storing password . Android provides many ways of storing data of an application. One of this way is called Shared Preferences. Shared Preferences allow to save and retrieve data in the form of key-value pair. A key-value pair (KVP) is a set of two linked data items: a key, which is a unique identifier for some item of data, and the value, which is either the data that is identified or a pointer to the location of that data. Key-value pairs are frequently used in lookup tables, hash tables and configuration files. Here we have stored password as well as three authorized numbers using key value pair.

FLOW CHART

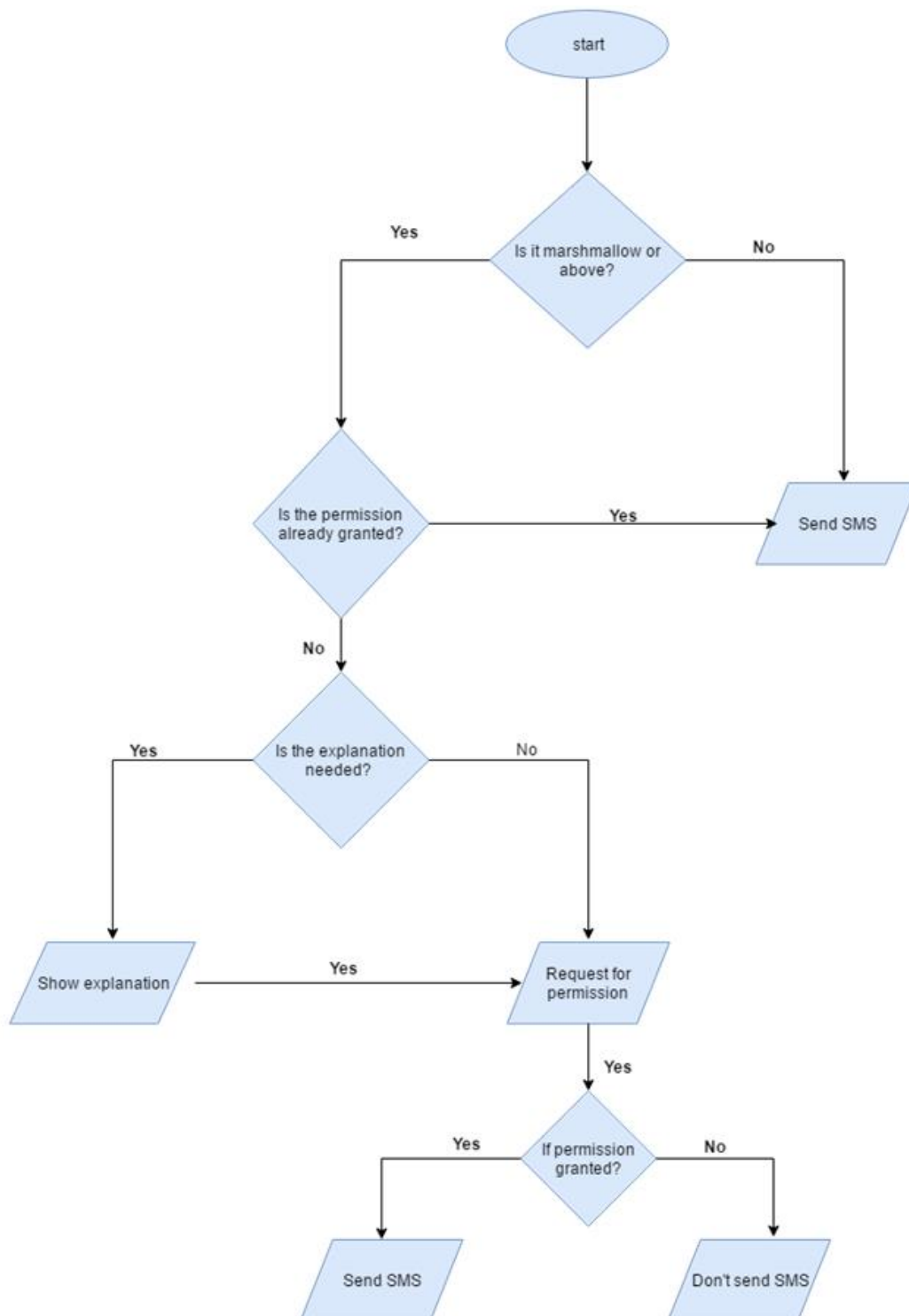


Figure 4.2: FLOWCHART

USE CASE DIAGRAM

Figure 3.7 denotes the Use Case Diagram of the proposed system. It shows the user's interaction with the systems. The user and the developer are the actors involved. All the interactions with the system are represented by the Use Cases.

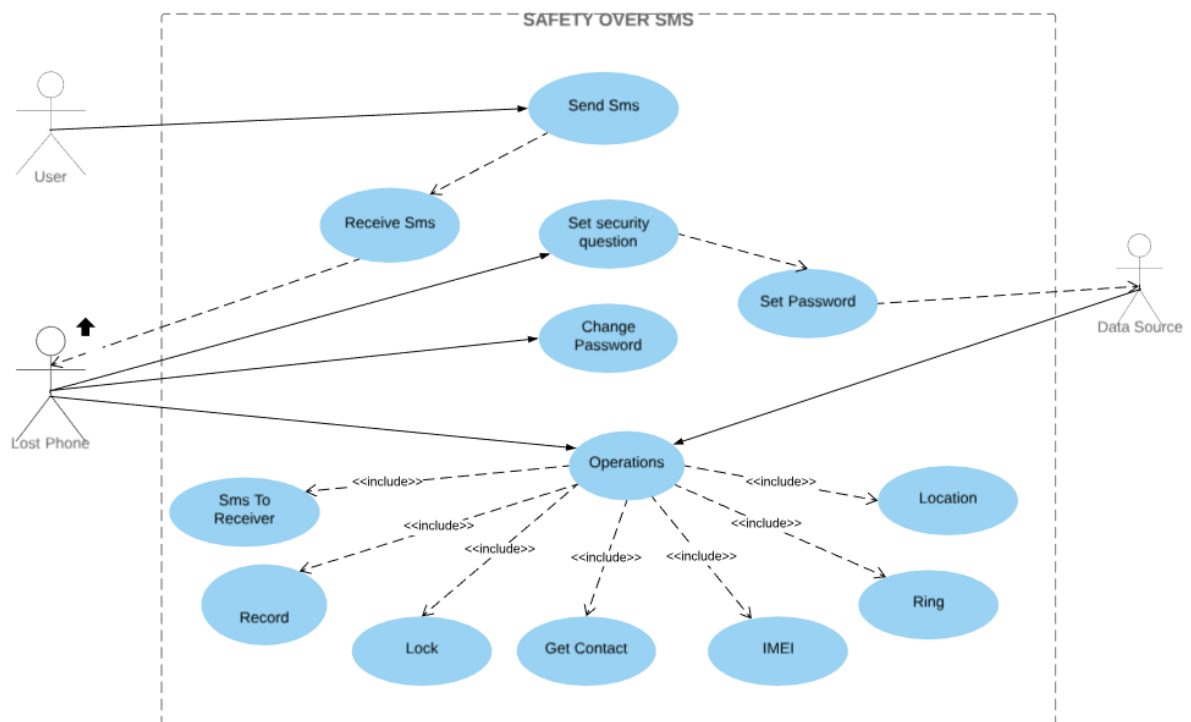


Figure 4.3: USE CASE DIAGRAM

SEQUENCE DIAGRAM

A Sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence.

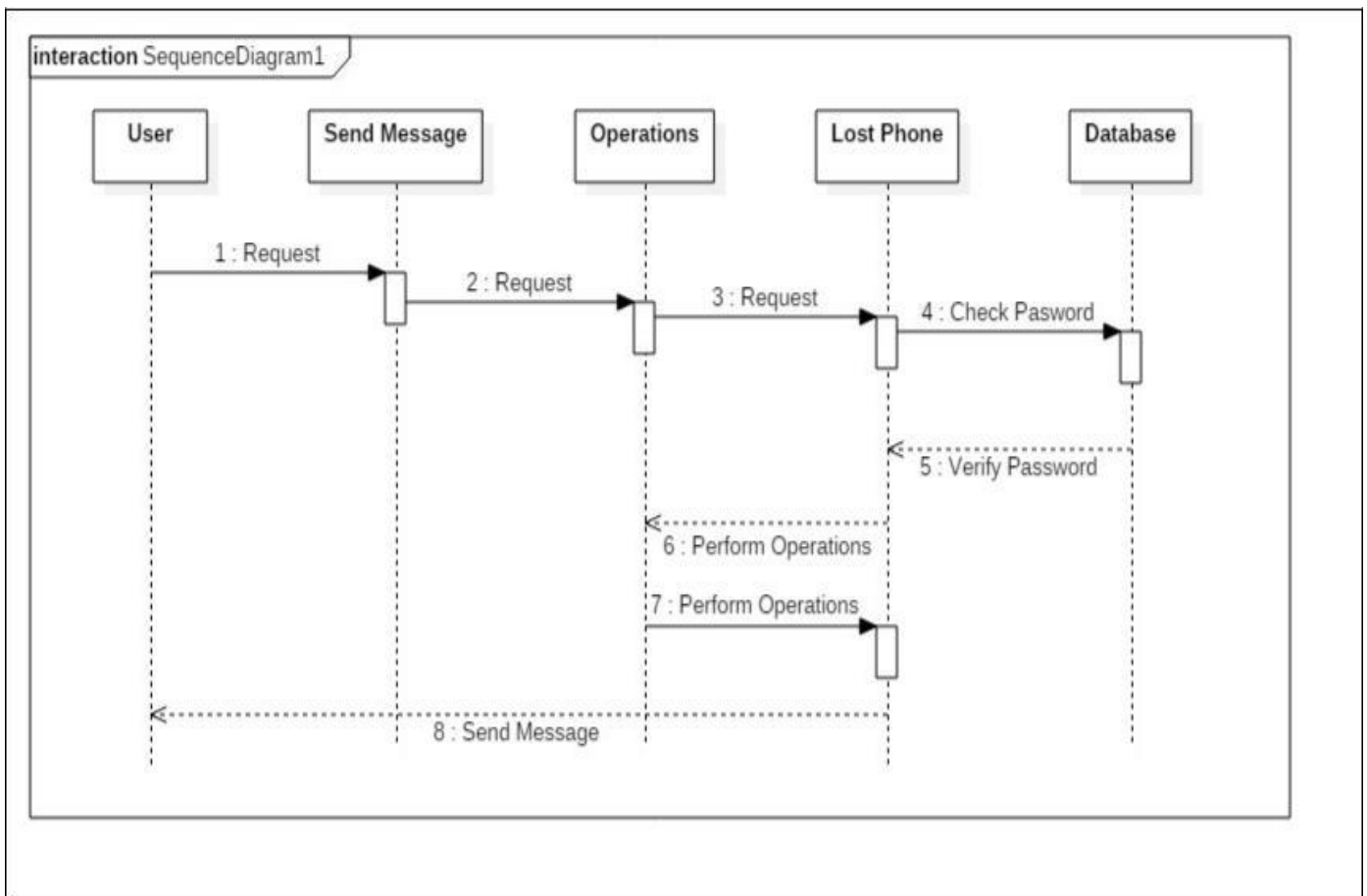


Figure 4.4: SEQUENCE DIAGRAM

It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. 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. Figure 3.8 represents the Sequence Diagram of the proposed system. It shows how the objects like the User, Developer, Terminal, HDFS, YARN operate with one another and in what order. The flow of the messages is also shown by the timelines.

ACTIVITY DIAGRAM

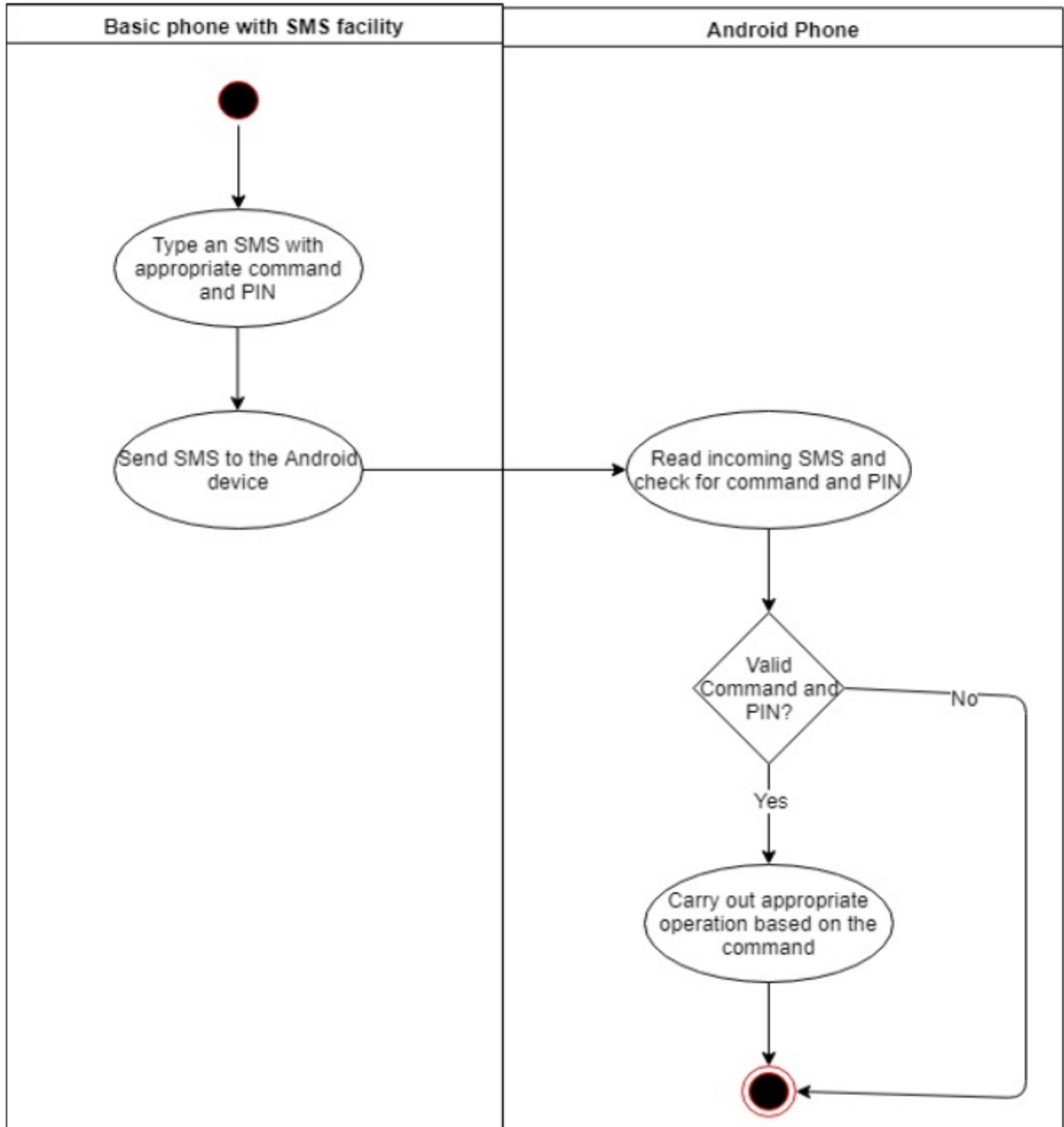


Figure 4.5: FLOW OF THE APPLICATION

4.2 SYSTEM REQUIREMENTS

This section will provide the user the required specification of the hardware and software components on which the proposed system is to be implemented.

4.2.1 Hardware Requirements

This subsection will provide the minimum requirements that must be fulfilled by the hardware components. The hardware requirements are as follows:-

- Android smartphone
- SMS gateway
- another mobile device (any OS)

4.2.2 Software Requirements

This subsection will provide the versions of software applications that must be installed. The software requirements are as follows: -

- Android studio
- Minimum SDK version on smartphone: 4.1
- Application permissions after installation

4.3 IMPLEMENTATION

activity_question.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="#455a64"
tools:context=".Question">

    <TextView
        android:id="@+id/question"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="#fafafa"
        android:layout_marginStart="5dp"
        android:layout_marginTop="181dp"
        android:layout_marginEnd="15dp"
        android:text="Security Question:"
        app:layout_constraintEnd_toStartOf="@+id/editQuestion"
        app:layout_constraintHorizontal_chainStyle="packed"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/editQuestion"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName"
        android:text=""
        app:layout_constraintBaseline_toBaselineOf="@+id/question"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/question" />

    <TextView
        android:id="@+id/answer"
        android:layout_width="wrap_content"
        android:layout_height="19dp"
        android:layout_marginTop="280dp"
        android:layout_marginEnd="32dp"
        android:text="Answer:"
        android:textColor="#fafafa"
```

```

        app:layout_constraintEnd_toStartOf="@+id/editAnswer"
        app:layout_constraintTop_toTopOf="parent" />

<EditText
    android:id="@+id/editAnswer"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="16dp"
    android:ems="10"
    android:inputType="textPersonName"
    app:layout_constraintBaseline_toBaselineOf="@+id/answer"
    app:layout_constraintEnd_toEndOf="parent" />

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

    android:layout_marginStart="36dp"
    android:layout_marginTop="84dp"
    android:background="@color/colorAccent"
    android:onClick="btn_click"
    android:text="Next"
    app:layout_constraintStart_toEndOf="@+id/answer"
    app:layout_constraintTop_toBottomOf="@+id/editAnswer" />

</android.support.constraint.ConstraintLayout>

```

Question.java

```
package com.example.sos;
```

```
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.content.pm.PackageManager;  
import android.net.Uri;  
import android.os.Bundle;  
import android.support.v7.app.AppCompatActivity;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.Toast;
```

```
public class Question extends AppCompatActivity {
```

```
    SharedPreferences prefs = null;  
    EditText question = null;  
    EditText answer = null;
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_question);  
  
        question = (EditText) findViewById(R.id.editQuestion);  
        answer = (EditText) findViewById(R.id.editAnswer);  
        prefs = getSharedPreferences("DB", MODE_PRIVATE);  
  
        if (!(prefs.getString("Question", "null").equals("null"))) {  
            {  
                Intent i = new Intent(Question.this, PIN.class);  
                startActivity(i);  
            }  
        }  
    }
```

```
    @Override
```

```
    public boolean onCreateOptionsMenu(Menu menu) {  
        getMenuInflater().inflate(R.menu.menu, menu);  
        return true;  
    }
```

```
    @Override
```

```
    public boolean onOptionsItemSelected(MenuItem item) {  
        Intent emailIntent = new Intent(Intent.ACTION_SEND);  
        emailIntent.setData(Uri.parse("mailto:"));
```

```

        emailIntent.setType("message/rfc822");
        emailIntent.putExtra(Intent.EXTRA_EMAIL, new
String[]{ "safetyoversms@gmail.com" });
        emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Safety Over SMS - Feedback");
        try {
            startActivity(Intent.createChooser(emailIntent, "Send mail..."));
            finish();
        } catch (Exception ex) {
        }
        return true;
    }

    public void btn_click(View view)
    {

        if(!(question.getText().toString().equals("") &&
answer.getText().toString().equals("")))
        {
            prefs = getSharedPreferences("DB", MODE_PRIVATE);
            prefs.edit().putString("Question", question.getText().toString()).apply();
            prefs.edit().putString("Answer", answer.getText().toString()).apply();

            Intent i = new Intent(Question.this, PIN.class);
            startActivity(i);
        }
        else
        {
            Toast.makeText(getApplicationContext(), "Enter Security Question and
Answer", Toast.LENGTH_LONG).show();
        }
    }
}

```

Activity_pin.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="#455a64"
tools:context=".PIN">

<TextView
    android:id="@+id/pin"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="10dp"
    android:text="PIN (4-Digit)"
    app:layout_constraintBaseline_toBaselineOf="@+id/editPIN"
    app:layout_constraintEnd_toStartOf="@+id/editPIN"
    app:layout_constraintStart_toStartOf="parent" />

<EditText
    android:id="@+id/editPIN"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="177dp"
    android:layout_marginEnd="7dp"
    android:ems="10"
    android:inputType="number"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/pin"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="129dp"
    android:layout_marginTop="106dp"
    android:background="@color/colorAccent"
    android:onClick="onClick"
    android:text="Next"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editPIN" />

</android.support.constraint.ConstraintLayout>
```

PIN.java

```
package com.example.sos;

import android.content.Intent;
import android.content.SharedPreferences;
import android.net.Uri;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class PIN extends AppCompatActivity {

    SharedPreferences prefs = null;
    EditText pin ;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_pin);

        prefs = getSharedPreferences("DB",MODE_PRIVATE);
        if(!(prefs.getString("PIN","null").equals("null"))){
            startActivity(new Intent(PIN.this,Home.class));
        }

    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu,menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        Intent emailIntent = new Intent(Intent.ACTION_SEND);
        emailIntent.setData(Uri.parse("mailto:"));
        emailIntent.setType("message/rfc822");
        emailIntent.putExtra(Intent.EXTRA_EMAIL, new
String[]{"safetyoversms@gmail.com"});
        emailIntent.putExtra(Intent.EXTRA_SUBJECT,"Safety Over SMS -
Feedback");
        try {
            startActivity(Intent.createChooser(emailIntent, "Send
mail..."));
            finish();
        } catch (Exception ex) {
        }
        return true;
    }

    public void onClick(View v)
    {
        pin = findViewById(R.id.editPIN);
    }
}
```

```
String regex = "\\d{4}$";
String PIN = pin.getText().toString();

if(PIN.matches(regex))
{
    prefs = getSharedPreferences("DB", MODE_PRIVATE);
    prefs.edit().putString("PIN", PIN).apply();
    Intent i = new Intent(PIN.this, Home.class);
    startActivity(i);
}
else
{
    Toast.makeText(getApplicationContext(), "ENTER A 4-DIGIT
PIN", Toast.LENGTH_LONG).show();
}
}
```


activity_home.xml

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:layout_marginBottom="10dp"
    android:layout_margin="1dp"
    android:background="#455a64">

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:background="#455a64"
        android:padding="15dp"
        android:src="@drawable/sos" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:orientation="horizontal">

        <TextView
            android:id="@+id/textview1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="end"
            android:layout_weight="0.25"
            android:background="#455a64"
            android:fontFamily="monospace"
            android:paddingTop="5dp"
            android:shadowColor="#eeeeee"
            android:text="PIN : "
            android:textAlignment="textEnd"
            android:textColor="#fafafa"
            android:textFontWeight="1"
            android:textSize="30dp"
            android:textStyle="bold" />

        <TextView
            android:id="@+id/displayPIN"
            android:layout_width="wrap_content"
            android:paddingTop="10dp"
            android:layout_height="wrap_content"
            android:layout_weight="0.25"
            android:background="#455a64"
```

```

        android:layout_gravity="start"
        android:paddingLeft="5dp"
        android:shadowColor="#eeeeee"
        android:textAlignment="textStart"
        android:textColor="#fafafa"
        android:textFontWeight="1"
        android:textSize="25dp"
        android:textStyle="italic" />
</LinearLayout>

```

```

<ImageButton
    android:id="@+id/changepin"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_margin="1dp"
    android:background="#455a64"
    android:cropToPadding="false"
    android:padding="2dp"
    android:paddingBottom="10dp"
    android:src="@drawable/chp" />

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:orientation="horizontal">

```

```

    <ImageButton
        android:id="@+id/aboutus"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="1dp"
        android:layout_weight="0.5"
        android:background="#455a64"
        android:cropToPadding="false"
        android:src="@drawable/abu" />

```

```

    <ImageButton
        android:id="@+id/help"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="1dp"
        android:layout_weight="0.5"
        android:background="#455a64"
        android:cropToPadding="false"
        android:src="@drawable/h" />

```

```

</LinearLayout>

```

```

</LinearLayout>

```

Home.java

```
package com.example.sos;

import android.Manifest;
import android.app.admin.DeviceAdminReceiver;
import android.app.admin.DevicePolicyManager;
import android.content.ComponentName;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.support.annotation.NonNull;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

public class Home extends AppCompatActivity {

    ImageButton help,about,change;
    TextView displayPIN;
    SharedPreferences prefs = null;
    String[] permission = new String[]{
        Manifest.permission.SEND_SMS,
        Manifest.permission.READ_SMS,
        Manifest.permission.RECEIVE_SMS,
        Manifest.permission.READ_CONTACTS,
        Manifest.permission.READ_PHONE_STATE,
        Manifest.permission.RECORD_AUDIO,
        Manifest.permission.WRITE_EXTERNAL_STORAGE,
        Manifest.permission.ACCESS_FINE_LOCATION
    };

    public static DevicePolicyManager dpm;
    public ComponentName cm;
    protected static final int REQUEST_ENABLE = 1;
    protected static final int SET_PASSWORD = 2;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);
        prefs = getSharedPreferences("DB",MODE_PRIVATE);
    }
}
```

```

displayPIN = findViewById(R.id.displayPIN);
displayPIN.setText(prefs.getString("PIN","Not Set"));

about = (ImageButton)findViewById(R.id.aboutus);

about.setOnClickListener(new View.OnClickListener() {
    public void onClick(View arg0) {
        Intent myIntent = new Intent(Home.this, AboutUs.class);
        startActivity(myIntent);
    }
});

help = (ImageButton)findViewById(R.id.help);

help.setOnClickListener(new View.OnClickListener() {
    public void onClick(View arg0) {
        Intent myIntent = new Intent(Home.this, Help.class);
        startActivity(myIntent);
    }
});

change = (ImageButton)findViewById(R.id.changepin);

change.setOnClickListener(new View.OnClickListener() {
    public void onClick(View arg0) {
        Intent myIntent = new Intent(Home.this, ChangePIN.class);
        startActivity(myIntent);
    }
});

}

@Override
public void onStart()
{
    super.onStart();

    dpm = (DevicePolicyManager)
getSystemService(Context.DEVICE_POLICY_SERVICE);
    cm = new ComponentName(this, MyDevicePolicyReceiver.class);
    // receiveSMSReciever=new ReceiveSMS();
    // IntentFilter intentFilter=new
IntentFilter("android.provider.Telephony.SMS_RECEIVED");
    // registerReceiver(receiveSMSReciever,intentFilter);
    if(dpm != null &&dpm.isAdminActive(cm)) {
        // admin active
    }
}

```

```

        else {
            Intent intent = new
Intent(DevicePolicyManager.ACTION_ADD_DEVICE_ADMIN);
            intent.putExtra(DevicePolicyManager.EXTRA_DEVICE_ADMIN, cm);
            startActivityForResult(intent, REQUEST_ENABLE);
        }

        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.READ_PHONE_STATE) !=
PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(Home.this, permission, 100);
            return;
        }
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (resultCode == RESULT_OK) {
            switch (requestCode) {
                case REQUEST_ENABLE:

                    dpm.setPasswordQuality(cm,
DevicePolicyManager.PASSWORD_QUALITY_SOMETHING);
                    boolean isSufficient = dpm.isActivePasswordSufficient();
                    if (!isSufficient) {
                        Intent setPasswordIntent = new
Intent(DevicePolicyManager.ACTION_SET_NEW_PASSWORD);
                        startActivityForResult(setPasswordIntent, SET_PASSWORD);
                    }
                    break;
            }
        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        if (requestCode == 100) {
            //check if all permissions are granted
            boolean allgranted = false;

            for (int i = 0; i < grantResults.length; i++) {
                if (grantResults[i] == PackageManager.PERMISSION_GRANTED) {
                    allgranted = true;
                } else {
                    allgranted = false;
                    break;
                }
            }
        }
    }

```

```

    }
}

if (!allgranted) {
    ActivityCompat.requestPermissions(Home.this, permission, 100);
}
}
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    Intent emailIntent = new Intent(Intent.ACTION_SEND);
    emailIntent.setData(Uri.parse("mailto:"));
    emailIntent.setType("message/rfc822");
    emailIntent.putExtra(Intent.EXTRA_EMAIL, new
String[]{"safetyoversms@gmail.com"});
    emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Safety Over SMS - Feedback");
    try {
        startActivity(Intent.createChooser(emailIntent, "Send mail..."));
        finish();
    } catch (Exception ex) {
    }
    return true;
}

public static class MyDevicePolicyReceiver extends DeviceAdminReceiver {

    @Override
    public void onDisabled(Context context, Intent intent) {
        Toast.makeText(context, "Device Admin Disabled",
Toast.LENGTH_SHORT).show();
    }

    @Override
    public void onEnabled(Context context, Intent intent) {
        Toast.makeText(context, "Device Admin is now enabled",
Toast.LENGTH_SHORT).show();
    }

    @Override
    public CharSequence onDisableRequested(Context context, Intent intent) {
        CharSequence disableRequestedSeq = "Requesting to disable Device Admin";
        return disableRequestedSeq;
    }
}

```

```

    }

    @Override
    public void onPasswordChanged(Context context, Intent intent) {
    }

    @Override
    public void onPasswordExpiring(Context context, Intent intent) {
    }

    @Override
    public void onPasswordFailed(Context context, Intent intent) {
    }

    @Override
    public void onPasswordSucceeded(Context context, Intent intent) {
    }

    @Override
    public void onReceive(Context context, Intent intent) {
        super.onReceive(context, intent);
    }
}
}

```

activity_change_pin.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="#455a64"
tools:context=".ChangePIN">

    <TextView
        android:id="@+id/newPIN"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="#fafafa"
        android:layout_marginStart="5dp"
        android:layout_marginTop="181dp"
        android:layout_marginEnd="15dp"
        android:text="New PIN"
        app:layout_constraintEnd_toStartOf="@+id/enterPIN"
        app:layout_constraintHorizontal_chainStyle="packed"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/enterPIN"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="number"
        android:text=""
        app:layout_constraintBaseline_toBaselineOf="@+id/newPIN"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/ques" />

    <TextView
        android:id="@+id/ques"
        android:layout_width="wrap_content"
        android:layout_height="19dp"
        android:layout_marginTop="280dp"
        android:layout_marginEnd="32dp"
        android:textColor="#fafafa"
        app:layout_constraintEnd_toStartOf="@+id/enterAnswer"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/enterAnswer"
```



```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginEnd="16dp"
android:ems="10"
android:inputType="textPersonName"
app:layout_constraintBaseline_toBaselineOf="@+id/ques"
app:layout_constraintEnd_toEndOf="parent" />
```

<Button

```
android:id="@+id/button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="36dp"
android:layout_marginTop="84dp"
android:background="@color/colorAccent"
android:onClick="change_pin"
android:text="Change PIN"
app:layout_constraintStart_toEndOf="@+id/ques"
app:layout_constraintTop_toBottomOf="@+id/enterAnswer" />
```

</android.support.constraint.ConstraintLayout>

ChangePIN.java

```
package com.example.sos;

import android.content.Intent;
import android.content.SharedPreferences;
import android.net.Uri;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class ChangePIN extends AppCompatActivity {

    SharedPreferences prefs = null;
    TextView question;
    EditText answer,newPIN;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_change_pin);
        prefs = getSharedPreferences("DB",MODE_PRIVATE);

        question = findViewById(R.id.ques);
        question.setText(prefs.getString("Question","Null"));
    }

    public void change_pin(View view)
    {
        answer = findViewById(R.id.enterAnswer);
        newPIN = findViewById(R.id.enterPIN);
        prefs = getSharedPreferences("DB",MODE_PRIVATE);

        String oldPIN = prefs.getString("PIN","0000");
        String new_PIN = newPIN.getText().toString();
        String ans = answer.getText().toString();
        String stored_ans = prefs.getString("Answer","null");

        if(ans.equals(stored_ans))
        {
            if(oldPIN.equals(new_PIN))
            {
                Toast.makeText(getApplicationContext(),"New PIN cannot be the same as old PIN",Toast.LENGTH_LONG).show();
            }
        }
    }
}
```

```

    }
    else
    {
        if(new_PIN.matches("^\\d{4}$"))
        {
            prefs = getSharedPreferences("DB", MODE_PRIVATE);
            prefs.edit().putString("PIN", new_PIN).apply();
            Toast.makeText(getApplicationContext(), "PIN
Changed", Toast.LENGTH_SHORT).show();
            Intent i = new Intent(ChangePIN.this, Home.class);
            startActivity(i);
        }
        else
        {
            Toast.makeText(getApplicationContext(), "ENTER A 4-DIGIT
PIN", Toast.LENGTH_LONG).show();
        }
    }
}
else
{
    Toast.makeText(getApplicationContext(), "Wrong
Answer", Toast.LENGTH_LONG).show();
}
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    Intent emailIntent = new Intent(Intent.ACTION_SEND);
    emailIntent.setData(Uri.parse("mailto:"));
    emailIntent.setType("message/rfc822");
    emailIntent.putExtra(Intent.EXTRA_EMAIL, new
String[]{"safetyoversms@gmail.com"});
    emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Safety Over SMS - Feedback");
    try {
        startActivity(Intent.createChooser(emailIntent, "Send mail..."));
        finish();
    } catch (Exception ex) {
    }
    return true;
}
}

```

activity_about_us.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView 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="#455a64">

    <LinearLayout 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="#455a64"
        android:orientation="vertical"
        tools:context=".AboutUs">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:orientation="vertical"
            android:paddingTop="15dp"
            android:paddingBottom="30dp">

            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginBottom="10dp"
                android:fontFamily="sans-serif-condensed"
                android:gravity="center"
                android:text="Developers:"
                android:textColor="#fafafa"
                android:textSize="40sp"
                android:textStyle="bold" />

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="2dp"
                android:layout_marginStart="40dp"
                android:layout_marginEnd="40dp"
                android:layout_marginBottom="10dp"
                android:background="#455a64"
                android:foreground="#fafafa"></LinearLayout>

            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
```

```
    android:fontFamily="cursive"
    android:gravity="center"
    android:text="Dhaval Shah"
    android:textColor="#fafafa"
    android:textSize="38sp" />
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:fontFamily="cursive"
    android:gravity="center"
    android:text="Khushil Shah"
    android:textColor="#fafafa"
    android:textSize="38sp" />
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:fontFamily="cursive"
    android:gravity="center"
    android:text="Heet Vora"
    android:textColor="#fafafa"
    android:textSize="38sp" />
```

```
</LinearLayout>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:orientation="vertical"
    android:paddingTop="20dp"
    android:paddingBottom="30dp">
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:fontFamily="sans-serif-condensed"
    android:gravity="center"
    android:text="Contact Us."
    android:textColor="#fafafa"
    android:textSize="33sp"
    android:textStyle="bold" />
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="2dp"
    android:layout_marginStart="40dp"
```

```
    android:layout_marginEnd="40dp"
    android:layout_marginBottom="10dp"
    android:background="#bdbdbd"
    android:orientation="horizontal" />
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:fontFamily="cursive"
    android:gravity="center"
    android:text="+91 96198 18332"
    android:textColor="#fafafa"
    android:textSize="25sp" />
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:fontFamily="cursive"
    android:gravity="center"
    android:text="+91 99302 89930"
    android:textColor="#fafafa"
    android:textSize="25sp" />
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:fontFamily="cursive"
    android:gravity="center"
    android:text="+91 81691 31878"
    android:textColor="#fafafa"
    android:textSize="25sp" />
```

```
</LinearLayout>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:orientation="vertical"
    android:paddingTop="20dp"
    android:paddingBottom="30dp">
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:fontFamily="sans-serif-condensed"
```

```
    android:gravity="center"
    android:text="Email Us."
    android:textColor="#fafafa"
    android:textSize="33sp"
    android:textStyle="bold" />
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="2dp"
    android:layout_marginStart="40dp"
    android:layout_marginEnd="40dp"
    android:layout_marginBottom="10dp"
    android:background="#bdbdbd"
    android:orientation="horizontal" />
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:fontFamily="sans-serif-light"
    android:gravity="center"
    android:text="safetyoversms@gmail.com"
    android:textColor="#fafafa"
    android:textSize="25sp" />
```

```
</LinearLayout>
```

```
<Space
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1" />
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="30dp"
    android:layout_margin="10dp"
    android:fontFamily="serif-monospace"
    android:gravity="center"
    android:text="We will always get back to you ASAP."
    android:textColor="#fafafa"
    android:textSize="20sp" />
```

```
</LinearLayout>
</ScrollView>
```

AboutUs.java

```
package com.example.sos;

import android.content.Intent;
import android.net.Uri;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;

public class AboutUs extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_about_us);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        Intent emailIntent = new Intent(Intent.ACTION_SEND);
        emailIntent.setData(Uri.parse("mailto:"));
        emailIntent.setType("message/rfc822");
        emailIntent.putExtra(Intent.EXTRA_EMAIL, new
String[] { "safetyoversms@gmail.com" });
        emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Safety Over SMS - Feedback");
        try {
            startActivity(Intent.createChooser(emailIntent, "Send mail..."));
            finish();
        } catch (Exception ex) {
        }
        return true;
    }
}
```


activity_help.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView 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="#455a64">

    <LinearLayout 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="#455a64"
        android:orientation="vertical"
        tools:context=".Help">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:orientation="vertical"
            android:paddingTop="15dp"
            android:paddingBottom="30dp">

            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginBottom="1dp"
                android:fontFamily="sans-serif-condensed"
                android:gravity="left"
                android:text="SOSCon [pin] [contact_name]"
                android:textColor="#fafafa"
                android:textSize="28sp" />

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="2dp"
                android:background="#455a64"
                android:gravity="center"
                android:foreground="#fafafa"></LinearLayout>

            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:fontFamily="serif-monospace"
                android:gravity="left"
                android:textStyle="italic"
                android:text="This command is used to retrieve important contact number."
                android:textColor="#fafafa"
```

```

        android:textSize="20sp" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:orientation="vertical"
    android:paddingTop="15dp"
    android:paddingBottom="30dp">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="1dp"
        android:fontFamily="sans-serif-condensed"
        android:gravity="center"
        android:text="SOSImei [pin]"
        android:textColor="#fafafa"
        android:textSize="30sp" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="2dp"
        android:background="#455a64"
        android:gravity="center"
        android:foreground="#fafafa"></LinearLayout>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:fontFamily="serif-monospace"
        android:gravity="left"
        android:textStyle="italic"
        android:text="This command is used to retrieve IMEI number."
        android:textColor="#fafafa"
        android:textSize="20sp" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:orientation="vertical"
    android:paddingTop="15dp"
    android:paddingBottom="30dp">

    <TextView

```

```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginBottom="1dp"
android:fontFamily="sans-serif-condensed"
android:gravity="center"
android:text="SOSLocate [pin]"
android:textColor="#fafafa"
android:textSize="30sp" />
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="2dp"
    android:background="#455a64"
    android:gravity="center"
    android:foreground="#fafafa"></LinearLayout>
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:fontFamily="serif-monospace"
    android:gravity="left"
    android:textStyle="italic"
    android:text="This command is used to retrieve the location of the device."
    android:textColor="#fafafa"
    android:textSize="20sp" />
```

```
</LinearLayout>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:orientation="vertical"
    android:paddingTop="15dp"
    android:paddingBottom="30dp">
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="1dp"
    android:fontFamily="sans-serif-condensed"
    android:gravity="center"
    android:text="SOSRecord [pin] [duration_in_seconds]"
    android:textColor="#fafafa"
    android:textSize="25sp" />
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="2dp"
    android:background="#455a64"
```

```

        android:gravity="center"
        android:foreground="#fafafa"></LinearLayout>

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:fontFamily="serif-monospace"
    android:gravity="left"
    android:textStyle="italic"
    android:text="This command is used to record the audio for specific
duration."
    android:textColor="#fafafa"
    android:textSize="20sp" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:orientation="vertical"
    android:paddingTop="15dp"
    android:paddingBottom="30dp">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="1dp"
        android:fontFamily="sans-serif-condensed"
        android:gravity="center"
        android:text="SOSSMS [pin] [contact_name]\n:[message]"
        android:textColor="#fafafa"
        android:textSize="25sp" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="2dp"
        android:background="#455a64"
        android:gravity="center"
        android:foreground="#fafafa"></LinearLayout>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:fontFamily="serif-monospace"
        android:gravity="left"
        android:textStyle="italic"
        android:text="This command is used to send SMS to an intended receiver."
        android:textColor="#fafafa"
        android:textSize="20sp" />

```

```
</LinearLayout>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_margin="10dp"  
    android:orientation="vertical"  
    android:paddingTop="15dp"  
    android:paddingBottom="30dp">
```

```
<TextView
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginBottom="1dp"  
    android:fontFamily="sans-serif-condensed"  
    android:gravity="center"  
    android:text="SOSRing [pin] "  
    android:textColor="#fafafa"  
    android:textSize="30sp" />
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"  
    android:layout_height="2dp"  
    android:background="#455a64"  
    android:gravity="center"  
    android:foreground="#fafafa"></LinearLayout>
```

```
<TextView
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:fontFamily="serif-monospace"  
    android:gravity="left"  
    android:textStyle="italic"  
    android:text="This command is used to ring the device."  
    android:textColor="#fafafa"  
    android:textSize="20sp" />
```

```
</LinearLayout>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_margin="10dp"  
    android:orientation="vertical"  
    android:paddingTop="15dp"  
    android:paddingBottom="30dp">
```

```
<TextView
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
    android:layout_marginBottom="1dp"
    android:fontFamily="sans-serif-condensed"
    android:gravity="center"
    android:text="SOSLock [pin]"
    android:textColor="#fafafa"
    android:textSize="30sp" />
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="2dp"
    android:background="#455a64"
    android:gravity="center"
    android:foreground="#fafafa"></LinearLayout>
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:fontFamily="serif-monospace"
    android:gravity="left"
    android:textStyle="italic"
    android:text="This command is used to lock the device."
    android:textColor="#fafafa"
    android:textSize="20sp" />
```

```
</LinearLayout>
```

```
</LinearLayout>
</ScrollView>
```

Help.java

```
package com.example.sos;

import android.content.Intent;
import android.net.Uri;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;

public class Help extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_help);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        Intent emailIntent = new Intent(Intent.ACTION_SEND);
        emailIntent.setData(Uri.parse("mailto:"));
        emailIntent.setType("message/rfc822");
        emailIntent.putExtra(Intent.EXTRA_EMAIL, new
String[] { "safetyoversms@gmail.com" });
        emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Safety Over SMS - Feedback");
        try {
            startActivity(Intent.createChooser(emailIntent, "Send mail..."));
            finish();
        } catch (Exception ex) {
        }
        return true;
    }
}
```

ReceiveSMS.java

```
package com.example.sos;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.database.Cursor;
import android.location.LocationManager;
import android.media.AudioManager;
import android.media.MediaRecorder;
import android.media.Ringtone;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.os.Looper;
import android.provider.ContactsContract;
import android.telephony.SmsManager;
import android.telephony.SmsMessage;
import android.telephony.TelephonyManager;
import android.widget.Toast;
import java.util.Random;

import static android.content.Context.MODE_PRIVATE;

public class ReceiveSMS extends BroadcastReceiver{

    public static final String SMS_BUNDLE = "pdus";
    String[] command;
    TelephonyManager telephonyManager;
    static String smsBody, address;
    String AudioSavePathInDevice = null;
    MediaRecorder mediaRecorder;
    SharedPreferences prefs = null;

    @Override
    public void onReceive(Context context, Intent intent) {
        Bundle intentExtras = intent.getExtras();
        if (intentExtras != null) {
            Object[] sms = (Object[]) intentExtras.get(SMS_BUNDLE);
            String smsMessageStr = "";
            for (int i = 0; i < sms.length; ++i) {
                SmsMessage smsMessage = SmsMessage.createFromPdu((byte[]) sms[i]);
                telephonyManager = (TelephonyManager)
                context.getSystemService(Context.TELEPHONY_SERVICE);
```



```

        smsBody = smsMessage.getMessageBody();
        address = smsMessage.getOriginatingAddress();

        smsMessageStr += "SMS From: " + address + "\n";
        smsMessageStr += smsBody + "\n";
        command = smsBody.split(" ");
    }

    prefs = context.getSharedPreferences("DB",MODE_PRIVATE);
    String pin = prefs.getString("PIN","Null");

    if(command[1].equals(pin))
    {
        switch (command[0]) {
            case "SOSImei":
                sendIMEI(address, telephonyManager);
                break;

            case "SOSCon":
                String search = "";

                for (int j = 2; j < command.length; j++) {
                    search += command[j];
                    if (!(j == command.length - 1)) {
                        search += " ";
                    }
                }
                showContact(search, address, context);
                break;

            case "SOSRing":

                try {

                    final AudioManager audioManager = (AudioManager)
context.getSystemService(Context.AUDIO_SERVICE);
                    if (audioManager.getStreamVolume(AudioManager.STREAM_RING) ==
0) {

audioManager.setRingerMode(AudioManager.RINGER_MODE_NORMAL);
                        audioManager.setStreamVolume(AudioManager.STREAM_RING,
audioManager.getStreamMaxVolume(AudioManager.STREAM_RING), 0);
                    }

                    Uri uri =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_RINGTONE);
                    final Ringtone ringtone = RingtoneManager.getRingtone(context, uri);
                    ringtone.setLooping(true);
                    ringtone.play();
                }
            }
        }
    }

```

```

Thread th = new Thread(new Runnable() {
    @Override
    public void run() {
        try {
            Thread.sleep(30000); //30000 is for 30 seconds, 1 sec =1000
            if (ringtone.isPlaying())
                ringtone.stop(); //for stopping the ringtone
        } catch (InterruptedException e) {
        }
    }
});
th.start();
} catch (Exception e) {
}
break;

case "SOSSMS":
    String searchRec = "";
    String message = "";
    int j;

    for (j = 2; j < command.length; j++) {
        if (command[j].charAt(0) != ':') {
            searchRec += command[j];
            if (!(command[j + 1].charAt(0) == ':')) {
                searchRec += " ";
            }
        } else
            break;
    }

    for (int i = j; i < command.length; i++) {
        message += command[i];
        if (i == j) {
            message = message.substring(1, message.length());
        }

        if (!(i == j - 1)) {
            message += " ";
        }
    }

    sendSMS(searchRec, message, context);
    break;

case "SOSRecord":
    int duration = Integer.parseInt(command[1]);
    recordAudio(context, duration);
    break;

```

```

        case "SOSLocate":
            context.startService(new Intent(context, Locate.class));
            break;

        case "SOSLock":
            context.startService(new Intent(context, Lock.class));
            break;

        default:
            break;
    }
}
}

public void sendIMEI(String address, TelephonyManager telephonyManager) {
    SmsManager smsManager = SmsManager.getDefault();
    smsManager.sendTextMessage(address, null, "IMEI: " +
        telephonyManager.getDeviceId(), null, null);
}

public void showContact(String search, String address, Context context) {
    Cursor c =
        context.getContentResolver().query(ContactsContract.CommonDataKinds.Phone.CONTENT
        _URI, null, null, null, null);

    String contact = "";
    String num = "";
    String name = "";

    try {
        while (!name.equals(search)) {
            assert c != null;
            c.moveToNext();
            name =
                c.getString(c.getColumnIndex(ContactsContract.CommonDataKinds.Phone.DISPLAY_NAME));
        }

        Cursor phones = context.getContentResolver().query(
            ContactsContract.CommonDataKinds.Phone.CONTENT_URI, null,
            ContactsContract.CommonDataKinds.Phone.DISPLAY_NAME + " = ?",
            new String[]{name}, null);
        while (phones.moveToNext()) {
            num +=
                phones.getString(phones.getColumnIndex(ContactsContract.CommonDataKinds.Phone.NUMBER)) + "\n";
        }
    } catch (Exception e) {
    }
}

```

```

        if (name.equals(search)) {
            contact += name + ":\n";
            contact += num;

            SmsManager smsManager = SmsManager.getDefault();
            smsManager.sendTextMessage(address, null, contact, null, null);
        }
        c.close();
    }

    public void sendSMS(String searchRec, String message, Context context) {
        Cursor c =
        context.getContentResolver().query(ContactsContract.CommonDataKinds.Phone.CONTENT
        _URI, null, null, null, null);

        String num = "";
        String name = "";

        try {
            while (!name.equals(searchRec)) {
                assert c != null;
                c.moveToNext();
                name =
                c.getString(c.getColumnIndex(ContactsContract.CommonDataKinds.Phone.DISPLAY_NA
                ME));
            }

            Cursor phones = context.getContentResolver().query(
                ContactsContract.CommonDataKinds.Phone.CONTENT_URI, null,
                ContactsContract.CommonDataKinds.Phone.DISPLAY_NAME + " = ?",
                new String[]{name}, null);
            while (phones.moveToNext()) {
                int type =
                phones.getInt(phones.getColumnIndex(ContactsContract.CommonDataKinds.Phone.TYPE));

                if (type == 2)
                    num =
                    phones.getString(c.getColumnIndex(ContactsContract.CommonDataKinds.Phone.NUMBER
                    ));
            }
        } catch (Exception e) {
        }

        if (name.equals(searchRec)) {
            SmsManager smsManager = SmsManager.getDefault();
            smsManager.sendTextMessage(num, null, message, null, null);
        }
        c.close();
    }

```

```

public void recordAudio(final Context context, final int duration) {
    Random ran = new Random();
    int r = ran.nextInt();
    AudioSavePathInDevice =
        Environment.getExternalStorageDirectory().getAbsolutePath() + "/" +
            "AudioRecording" + r + ".mp3";

    MediaRecorderReady();

    try {
        mediaRecorder.prepare();
        mediaRecorder.start();
        Toast.makeText(context, "Recording Started", Toast.LENGTH_LONG).show();

        Thread th = new Thread(new Runnable() {
            @Override
            public void run() {

                try {
                    Looper.prepare();
                    Thread.sleep(duration * 1000); //30000 is for 30 seconds, 1 sec =1000
                    mediaRecorder.stop();
                    Toast.makeText(context, "Recording Stopped",
Toast.LENGTH_LONG).show();
                } catch (InterruptedException e) {
                }
            }
        });
        th.start();

        } catch (Exception e) {
        }
    }

    public void MediaRecorderReady() {
        mediaRecorder = new MediaRecorder();
        mediaRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);
        mediaRecorder.setOutputFormat(MediaRecorder.OutputFormat.MPEG_4);
        mediaRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.AAC);
        mediaRecorder.setOutputFile(AudioSavePathInDevice);
    }

    public static void sendLocation(String url)
    {
        SmsManager smsManager = SmsManager.getDefault();
        smsManager.sendTextMessage(address, null, url, null, null);
    }
}

```

Locate.java

```
package com.example.sos;

import android.app.Service;
import android.content.Context;
import android.content.Intent;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.IBinder;

public class Locate extends Service {

    private LocationManager locManager;
    private LocationListener locListener = new MyLocationListener();

    private boolean gps_enabled = false;
    private boolean network_enabled = false;

    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startid) {
        locManager = (LocationManager)
this.getSystemService(Context.LOCATION_SERVICE);
        try {
            network_enabled =
locManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER);
        }
        catch (Exception ex) {
        }

        if (network_enabled)
            locManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0,
0, locListener);

        stopSelf();
        return START_STICKY;
    }

    class MyLocationListener implements LocationListener {
        @Override
```

```

public void onLocationChanged(Location location) {
    if (location != null) {
        // This needs to stop getting the location data and save the battery power.
        locationManager.removeUpdates(locListener);

        String latitude = String.valueOf(location.getLatitude());
        String longitude = String.valueOf(location.getLongitude());
        String url = "https://maps.google.com/?q="+latitude+"", "+longitude;

        ReceiveSMS.sendLocation(url);
    }
}

@Override
public void onProviderDisabled(String provider) {
    // TODO Auto-generated method stub

}

@Override
public void onProviderEnabled(String provider) {
    // TODO Auto-generated method stub

}

@Override
public void onStatusChanged(String provider, int status, Bundle extras) {
    // TODO Auto-generated method stub
}
}
}

```

Lock.java

```
package com.example.sos;

import android.app.Service;
import android.app.admin.DevicePolicyManager;
import android.content.ComponentName;
import android.content.Context;
import android.content.Intent;
import android.os.IBinder;

public class Lock extends Service {
    DevicePolicyManager devicePolicyManager;
    ComponentName componentName;

    @Override
    public IBinder onBind(Intent intent) {

        return null;
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startid) {
        devicePolicyManager = (DevicePolicyManager)
getSystemService(Context.DEVICE_POLICY_SERVICE);
        componentName = new ComponentName(this, Home.MyDevicePolicyReceiver.class);
        devicePolicyManager.lockNow();
        stopSelf();
        return START_STICKY;
    }
}
```


AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.sos">

    <uses-permission android:name="android.permission.WRITE_SMS" />
    <uses-permission android:name="android.permission.READ_SMS" />
    <uses-permission android:name="android.permission.RECEIVE_SMS" />
    <uses-permission android:name="android.permission.READ_CONTACTS" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.SEND_SMS" />
    <uses-permission android:name="android.permission.RECORD_AUDIO" />
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/logo"
        android:label="@string/app_name"
        android:roundIcon="@drawable/sos"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".ChangePIN"/>

        <activity android:name=".Help" />

        <activity android:name=".AboutUs" />

        <activity android:name=".Home" />
        <receiver
            android:name=".Home$MyDevicePolicyReceiver"
            android:permission="android.permission.BIND_DEVICE_ADMIN" >
            <meta-data
                android:name="android.app.device_admin"
                android:resource="@xml/device_policies" />

            <intent-filter>
                <action android:name="android.app.action.DEVICE_ADMIN_ENABLED"
/>
                <action
android:name="android.app.action.DEVICE_ADMIN_DISABLE_REQUESTED" />
                <action android:name="android.app.action.DEVICE_ADMIN_DISABLED"
/>
                <action
android:name="android.app.action.ACTION_PASSWORD_CHANGED" />
            </intent-filter>
        </receiver>
    </application>
</manifest>
```

```

        <action
android:name="android.app.action.ACTION_PASSWORD_EXPIRING" />
        <action
android:name="android.app.action.ACTION_PASSWORD_FAILED" />
        <action
android:name="android.app.action.ACTION_PASSWORD_SUCCEEDED" />
    </intent-filter>
</receiver>

<activity android:name=".PIN" />

<activity android:name=".Question">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>

<service android:name=".Lock"
    android:process=":remote"/>

<service android:name=".Locate"/>
<receiver android:name=".ReceiveSMS"
    android:enabled="true">
    <intent-filter>
        <action android:name="android.provider.Telephony.SMS_RECEIVED" />
    </intent-filter>
</receiver>
</application>

</manifest>

```

Device_policies.xml

```
<?xml version="1.0"?>
<device-admin xmlns:android="http://schemas.android.com/apk/res/android">
  <uses-policies>
    <limit-password />
    <watch-login />
    <force-lock />
  </uses-policies>
</device-admin>
```

Menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:title="Mail Us"
        android:id="@+id/feedback"/>
</menu>
```

Conclusion



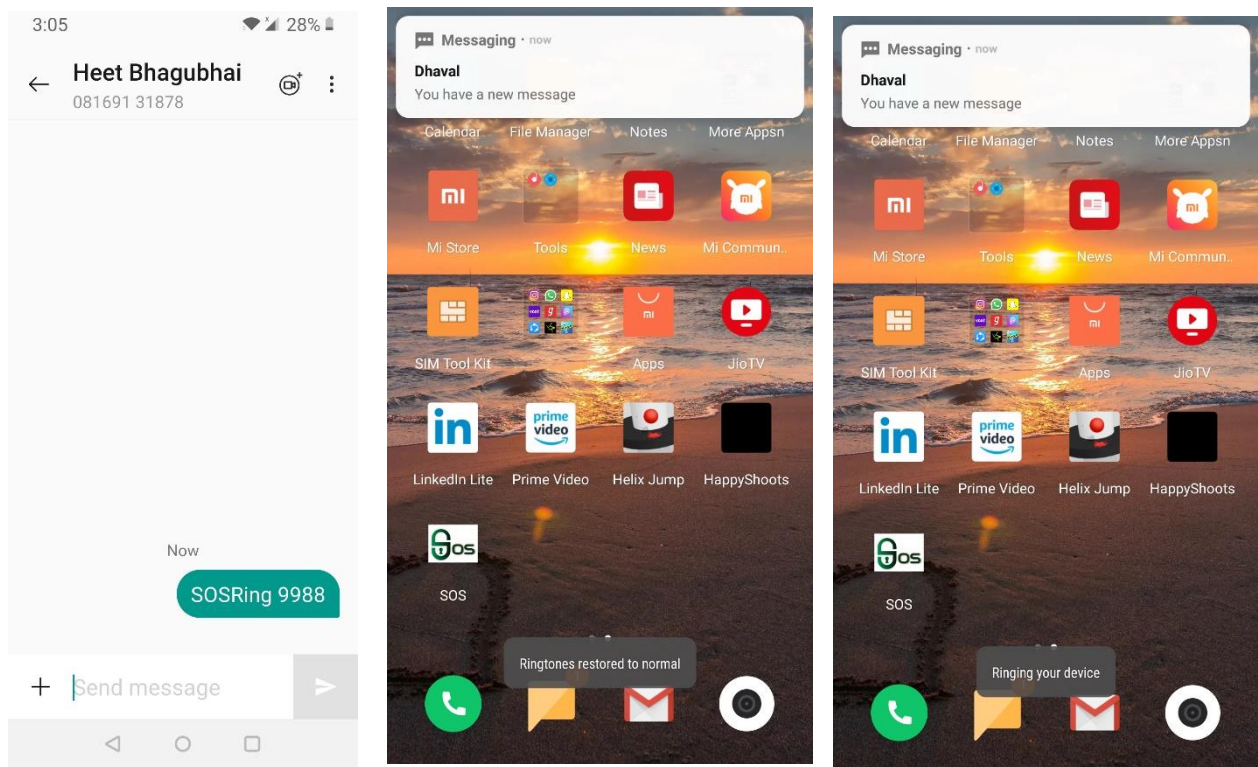
CHAPTER 5: RESULTS & CONCLUSIONS

5.1 TEST CASES

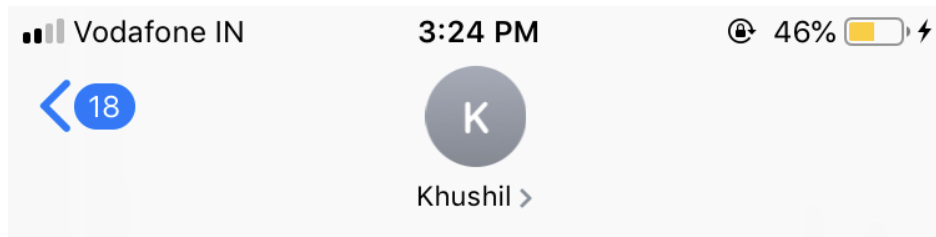
A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of an application. Table 3.1 gives detailed information on the test cases for the proposed system.

Following are the Test cases:

- Test case 1 - The purpose of this test case is to find when lost phone is in the silent or normal mode the phone will get ring.



- Test case 2 - The purpose of this test case is to find the accurate location of the phone.

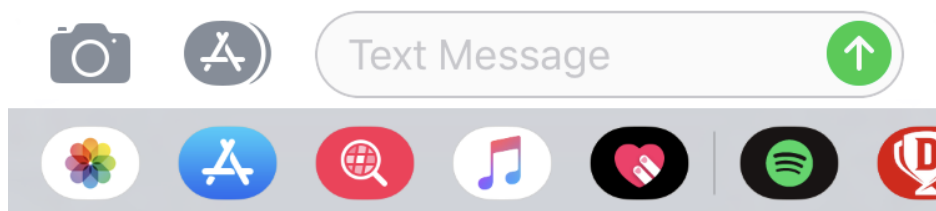


Text Message
Today, 3:21 PM

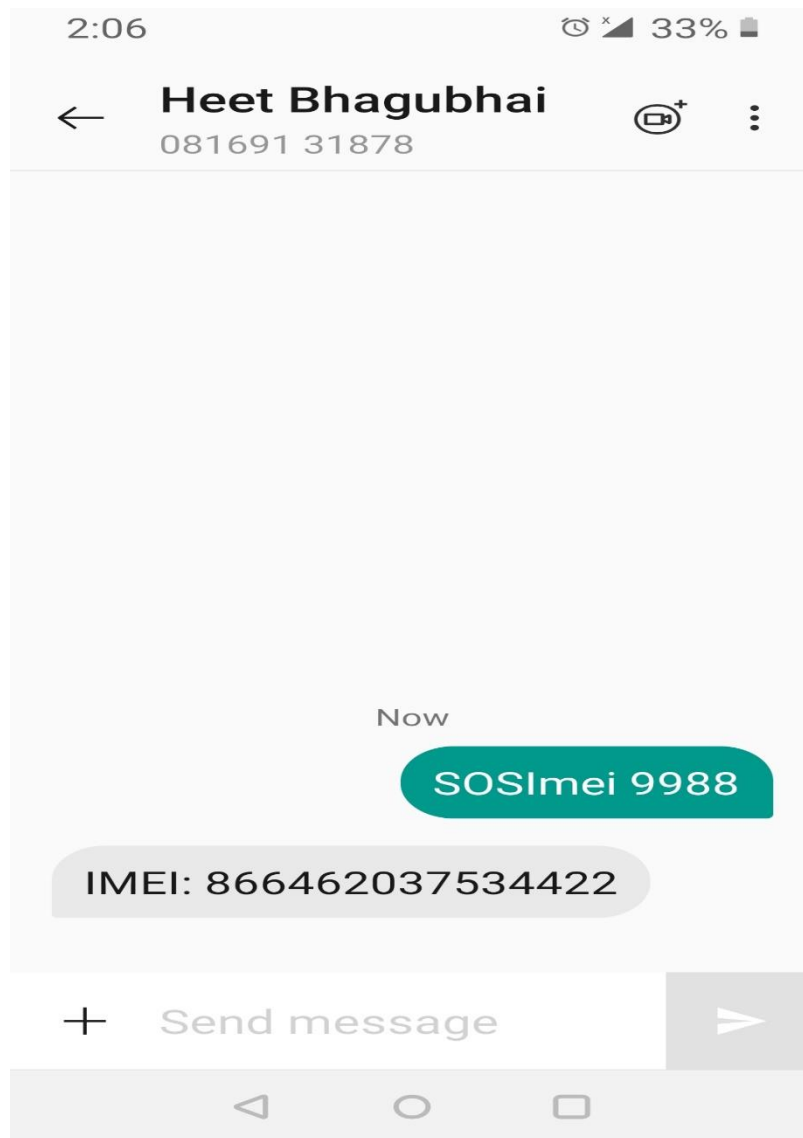
SOSLocate 1234

Tap to Load Preview

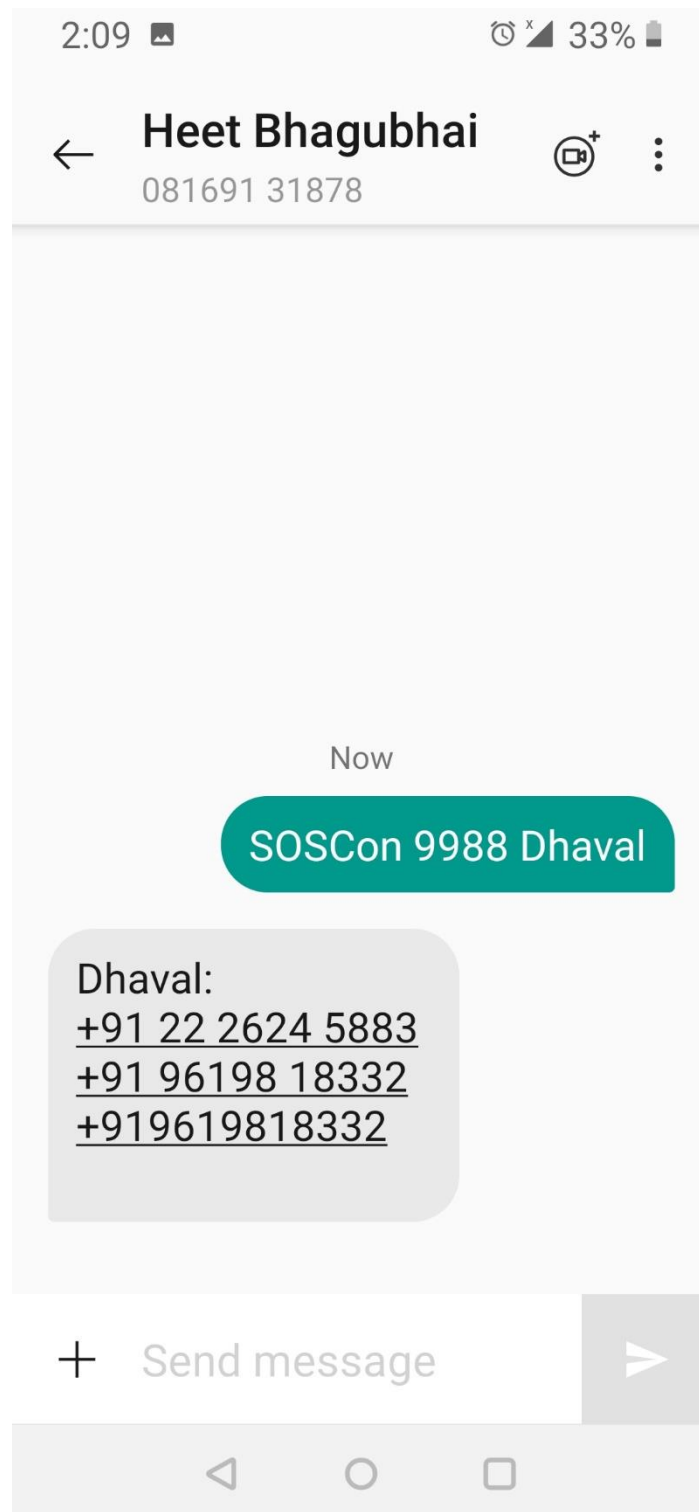
maps.google.com



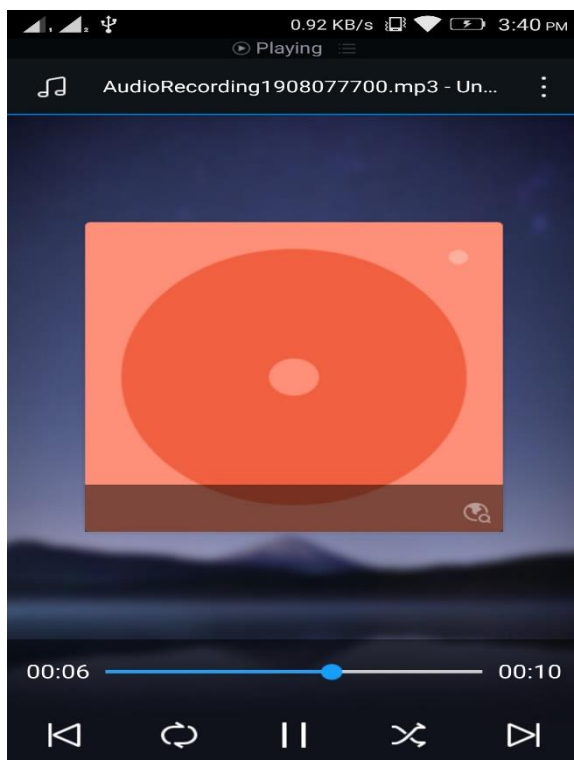
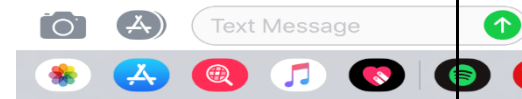
- Test case 3 - The purpose of this test case is to find the IMEI of the lost phone.



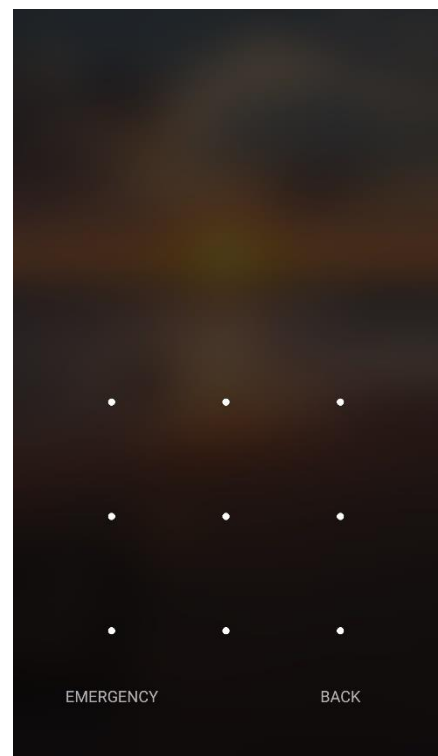
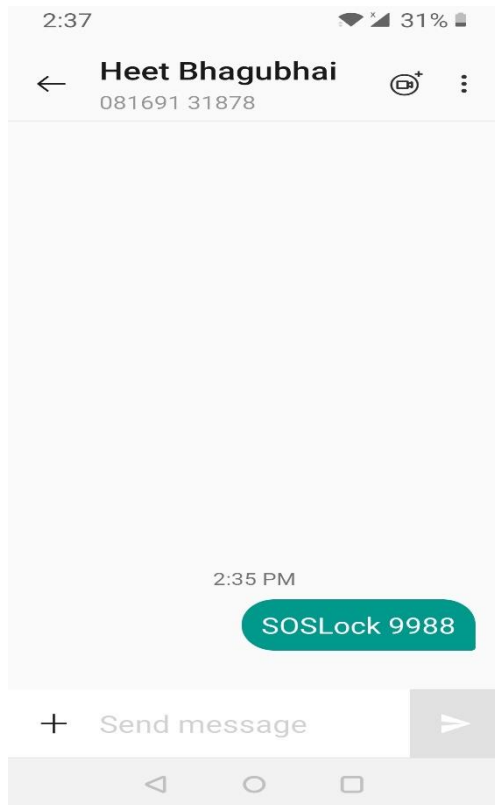
- Test case 4 - The purpose of this test case is to find the contact details of the respective person.



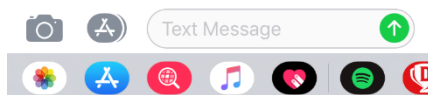
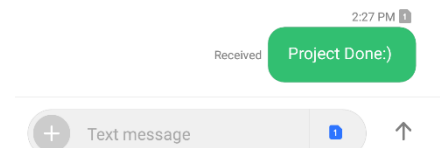
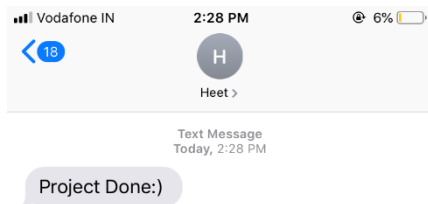
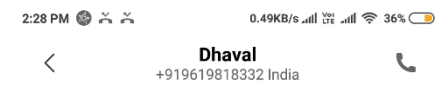
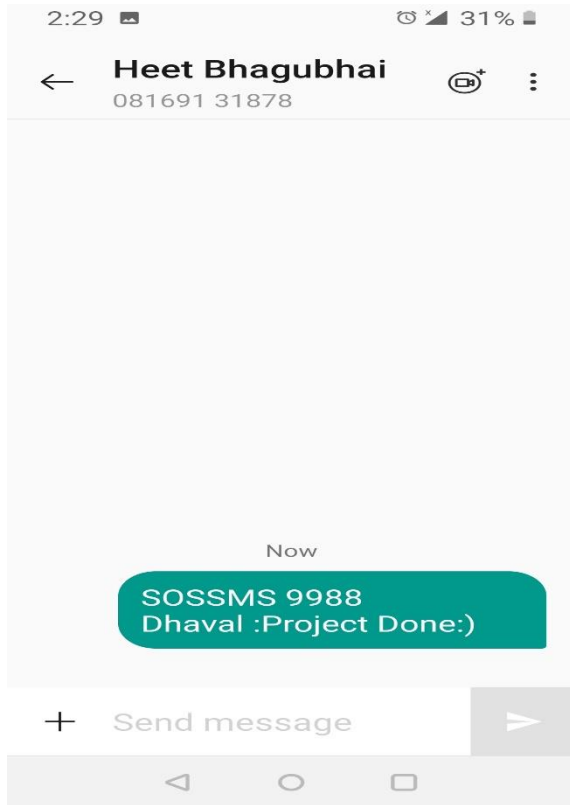
- Test case 5 - The purpose of this test case is to record the audio from the background.



- Test case 6 – The purpose of this test case is to Lock the phone to avoid misuse of the Phone data.



- Test case 7- The purpose of this test case is to send the SMS to an intended receiver.



Sr No	Module Name	Input	Output
1	Retrieve Contact	The phone sends a message which sends the contact name whose contact number is required as a parameter.	Retrieves the requested contact.
2	Retrieve Location	The Phone sends the command used by our app to retrieve the Location.	The location of the phone is retrieved, and the link is sent to the requested device.
3	Retrieve IMEI	The phone sends the command to retrieve IMEI number to trace the phone if stolen	Retrieves the IMEI of the device
4	Record audio	Send SMS command with the time limit as the parameter	Records the audio for the requested seconds sent as the parameter with the input message
5	Ring Device	Send SMS command to ring the phone for 30 seconds.	The phone rings loudly even if it is on silent.
6	Lock Device	The phone sends the SMS command to the lost/stolen phone to lock the device.	Locks the device and the phone is unlocked with the pin set by the user during the installation.
7	Send SMS to an intended Receiver	SMS command is sent with the name of the contact to whom you want to send the message with the message string as the parameter.	The SMS is sent to the intended receiver without the need of the Contact number.

Table 5.1:TEST CASES

5.2 LIMITATIONS

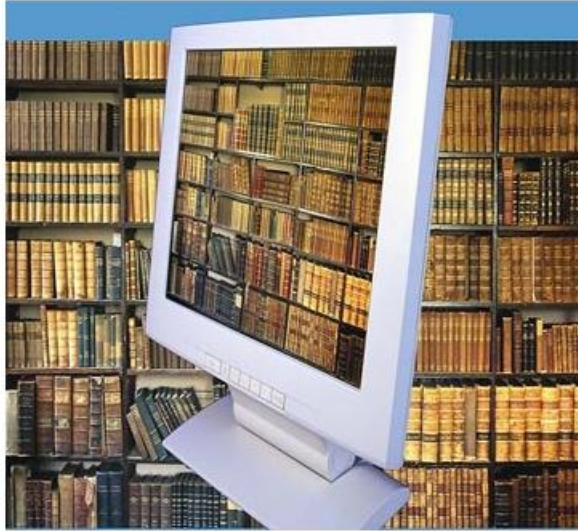
- The device cannot record the audio if sufficient storage is not available in your device.
- The basic functionalities of your mobile phone should be intact i.e. your phone should have sufficient balance, should have network to send/receive SMS, etc.
- In Dual Sim, one Sim should be selected by default to send SMS.
- The contact name for the commands is case sensitive.

5.3 FUTURE SCOPE

- More facilities can be provided in this application by providing the authentication of the E Mail of the users.
- Security can be made more powerful .If your device cannot be unlocked with 5 attempts, a picture will be captured and sent to the E-Mail.
- Additional features such as SIM change alert, etc can be sent.

5.4 CONCLUSION

This app provides security to android smartphone. The basic security provided by this app is to remotely lock an android device. One of the functionalities of the app is to retrieve the contact you want. At the same time, the app can ring the device and receive the current longitude and latitude to know where the device is located. To report a complaint of a missing phone, we need an IMEI number to track the phone. The app can also retrieve IMEI number of the device by just sending a SMS. You can also secretly record audio on your phone anytime you want. This app also helps you to send a message to the intended user without the need of retrieving the contact. Lock phone function performs lock operation on lost/stolen device so that thief will be barred from using that device.



BIBLIOGRAPHY

References:

- [1] Y.F. Chang, C.S. Chen, and H. Zhou, " Smart Phone For Mobile Commerce", Computer Standards Interfaces, Vol. 31, Issue 4, June, 2009"
- [2] Shafik G. Punfa and, "Smartphone Device Analysis", Scale Digital Device Forensics Journal, Vol. 2, No. 1, June, 2008"
- [3] Schiller , "Mobile Communication", Pearson Education 1 Edition, 7th reprint-2003.
- [6] J.ManiBharathi,S.Hemalatha,V.Aishwarya,C.Meenapriya, L.Hepzibha Shekinah Grace,"Advancement in Mobile Communication using Android", International Journal of Computer Applications (0975 – 8887),Volume 1 – No. 7,2010
- [7] B. Rao, L. Minakakis, Evolution of mobile location-based services, Communication of the ACM Vol. 46.
- [9] Can Komar and Cenersory, "Location tracking and location based services using IEEE 802.11 WLAN infrastructure",Europen. wireless, 2004,Barcelona Spain,24-27feb2004.
- [10] Kumar, Sandeep, Mohammed Abdul Qadeer, and Archana Gupta. "Location based services using android." In IMSAA'09: Proceedings of the 3rd IEEE International Conference on Internet Multimedia Services Architecture and Applications,pp.335-339.2009.

Website References

[1] <https://developer.android.com/>

[2] <https://www.tutorialspoint.com/android/>