

A
Micro Project Report
on

Women Safety App

Submitted by
Anuradha Partudkar.[30]
Pawar Shweta [31]
SalunkeChetna .[35]
Shinde Menka.[36]
Zare Sangita.[43]

Under the Guidance of

Prof. Galshetwar

In the Partial Fulfilment of sixth Semester of Diploma in
Computer Engineering

Department of ComputerEngineering
P.L.G.P, Latur

Affiliated to

Maharashtra State
Board of Technical Education

Academic Year 2020-21

Part – A Micro-Project Proposal
Women safety application

1.0 Aims/Benefits of theMicro-Project:

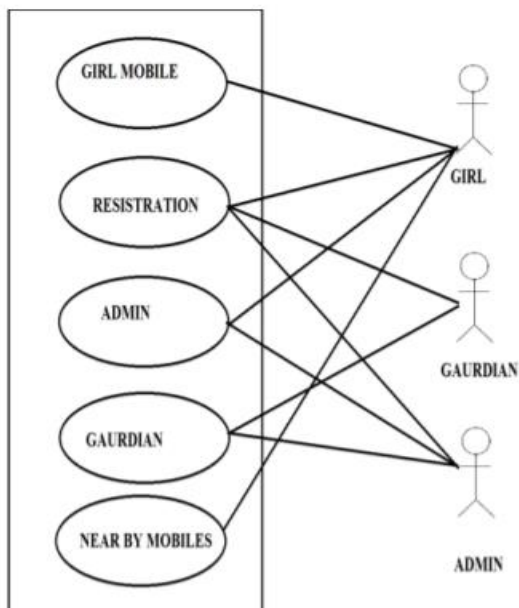
This system is for women's safety and overcomes existing systems. This GPS system is the "Women's Safety App". It consists of a GPS device, an Android phone. The unit will provide status information such as latitude, the longitude of the user.

The proposed App is based on advanced sensors. Each time a user makes a phone call, an emergency signal will be generated automatically and then an information alert will be sent to the contacts that have been added to the emergency call.

The proposed system will be implemented with the help of android application. Which will alert the nearby people who having this application by sending alert messages to them and alert sound in the guardian mobile on shaking of victim mobile. Also sends messages and alert sound to the saved contacts in the application and police station. Which also show the location of the victim with the help of GPS tracker system. Which also make sound in guardian mobile when his/her mobile in silent mode.

2.0 Course OutcomesAddressed:

- a) Interpret features of Andriod operating system.
- b) Configure Android environment and development tools.
- c) Develop rich users Interfaces by using layouts and controls.
- d) Use User Interface components for android application development.
- e) Create Android application using database.
- f) Publish Android applications.



3.0 Proposed Methodology:

A Use case is a description of set of sequence of actions graphically it is rendered as an ellipse with solid line including only its name. Use case diagram is a behavioural diagram that shows a set of use cases and actors and their relationship. It is an association between the use cases and actors. An actor represents a real-world object.

Use case diagram consists mainly three modules :

- Guardian app which is installed in the guardian mobile. and initiated by giving there mobile number
- . Girl app which is installed in the girls mobile. and initiated by giving girls name, id, mobile number.
- Server login module which consists data of guardian location informatio

4.0 ActionPlan

S. No	Details of activity	Planned Start date	Planned Finish date	f Responsible Team Members
1	Formation of group for microproject	01/04/2021	04/04/2021	All group members
2	Searching of microproject topic	05/04/2021	06/04/2021	All group members
3	Selection of microproject topic	07/04/2021	10/04/2021	All group members
4	Searching of required information for microproject	08/04/2021	15/04/2021	All group members
5	Collection of information from goggle	16/04/2021	25/04/2021	All group members
6	Discussion on all information with each other	26/04/2021	30/04/2021	All group members
7	Started assembling of microproject	01/05/2021	31/05/2021	All group members

5.0 Resources required:

Sr No	Name of Resource/Material	Specification	Qty.	Remarks
1	Hardware Resource	Desktop with Intel Core 2 Duo 2.93 GHz, RAM 2GB, HDD 160 GB	1	-
2	Software Resource	Android Studio	1	-
3	Any OtherResource	Internet	-	-

1.0 Part – B Micro-Project Report

2.0 Rationale:

In today's fast moving world, Women Security is an issue of growing concern. We have read about many unfortunate incidents happening with women and the rate is increasing. Women these days are working women and the globalization has made us aware of gender equality. Earlier the women were restricted only to the household chores. With the changing scenario, women are competing with men in all fields. We can see women going to great success levels in all fields, may it be corporate, scientific, education, business or any other field. Safety of women matters a lot whether at home, outside the home or working place. Last few crimes against women especially the case in Delhi was very dread and fearful. Because of such crimes, women safety has become a major topic. According to the statistics, it is found that every two out of three women have suffered trauma in the last year. According to the survey of women, it is found that women are losing their confidence because of such incidents. By the survey of Delhi government's Women and Child Development Department, around 80% of the women in national capital have fear regarding their safety. Women are harassed not only in the night or evening but also in the day time at their home, working places, or other places like street, club, etc. It is found through the survey that the reason of safety concern is the lack of gender-friendly environment and improper functional infrastructure such as consumption of alcohol and drugs in open area, lack of adequate lighting, safe public toilets, sidewalks, lack of effective police service, lack of properly working helpline numbers, etc. A huge percentage of women have no faith that police can curb such harassment cases. There is an urgent need to understand and solve this problem of women safety so that they can also grow equally like men in their own country.

Aims/Benefits of the Micro-Project:

In today's world, people using smart phones have increased rapidly and hence, a smart phone can be used efficiently for personal security or various other protection purposes. The heinous incident that outraged the entire nation have waken us to go for the safety issues and so a host of new apps have been developed to provide security systems to women via their phones. This paper presents women security an Android Application for the Safety of Women and this app can be activated this app by a shaking the mobile, whenever need arises. This app identifies the location of place through GPS and sends a message comprising this location URL to the registered contacts and also send messages to near by mobile which are having this app.

The system will be implemented as a 3 module application :

• Victim app module

1. Initially the victim app need to be installed in the mobile.
2. Then victim should enrol her ID, Name, Phone number.
3. And then click on register button.
4. Then it display registered successfully.
5. If that mobile number is already registered it will display this number is already registered.

• Guardian app module

1. Initially the guardian app need to be installed in the mobile.
2. Guardian should enroll their phone numbers.
3. And click on register button.
4. Then it display registered successfully.
5. If that mobile number is already registered it will display this number is already registered.

- **Server module**

1. After installation of guardian app and victim app two tables are created in server model.
2. Two tables are victim table and guardian table.
3. Victim table contain five columns VID, VNAME, PHONE -NUMBER, LATITUDE, LONGITUDE.
4. Guardian table contains three columns PHONE NUMBER, LATITUDE, LONGITUDE.

3.0 Course Outcomes Achieved:

- g) Develop program using GUI framework (ATW and Swing).
- h) Handle events of AWT and swing components.
- i) Develop programs to handle events in java programming.
- j) Develop java programs using networking concepts.
- k) Develop program using database.
- l) Develop programs using servlets.

4.0 Actual Methodology Followed:



Figure : Spiral Model of Software Development

Figure shows the spiral model of software development. This model is a combination of well-known waterfall model and iterative prototyping. It yields rapid development of more complete version of software. Using spiral model software is developed as series of Evolutionary releases. During the initial releases, it may just paperwork or prototype. But during later releases the version goes towards more completed stage.

The spiral model can be adopted to apply throughout the entire life cycle of the application from concept development to maintenance. The spiral model is divided into a set of framework activities defined by software engineer team. The initial activity is shown from center of circle and developed in clockwise

Mobile application development

direction. Each spiral of the model includes following four stages:

Planning Phase: This stage includes requirement gathering, cost estimation, resource allocation.

Risk Analysis Phase: This stage involves strengths and weaknesses of the project. Design Phase: This stage includes coding, internal testing and deployment.

Evaluation Phase: Last stage involves client evaluation (client side testing) to get the feedback.

Various activities which are performed according to the spiral model phases are shown Tab

5.0 Actual Resources Used:

Sr No	Name of Resource/Material	Specification	Qty.	Remarks
1	Hardware Resource	Processor i3/HDD-1TB/RAM-8GB	1	
2	Software Resource	Android Studio, jdk1.8.0	1	
3	Any Other Resource	Printer	1	

Team Members with Roll Nos.

- 1) Anuradha Partudkar[30]
- 2) Pawar Shweta [31]
- 3) Salunke Chetna .[35]
- 4) Shinde Menka .[36]
- 5) Zare Sangita.[43]

Code:

//XML CODE

1. Home Page

```
<RelativeLayoutxmlns: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: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" >
```

```
<Button
android:id="@+id/button1"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_above="@+id/button2"
android:layout_alignLeft="@+id/button2"
android:onClick="register"
android:layout_marginBottom="71dp"
android:text="Register" />
```

```
<Button
android:id="@+id/button3"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/button2"
android:layout_below="@+id/button2"
android:onClick="display_no"
android:layout_marginTop="74dp"
android:text="View Registered" />
```

```
<Button
android:id="@+id/button2"
```

Mobile application development

```
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_centerHorizontal="true"
android:onClick="instruct"
android:layout_centerVertical="true"
android:text="Instructions" />
```

```
<Button
android:id="@+id/button4"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_alignParentBottom="true"
android:layout_centerHorizontal="true"
android:onClick="verify"
android:text="Register Your Mobile Number" />
```

```
</RelativeLayout>
```

2. Register:

```
<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: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=".Register" >
```

```
<TextView
android:id="@+id/textView1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentTop="true"
android:layout_marginTop="64dp"
android:text="Name:" />
```

```
<TextView
```


Mobile application development

```
android:id="@+id/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/textView1"
android:layout_below="@+id/textView1"
android:layout_marginTop="71dp"
android:text="Number:" />
```

```
<Button
android:id="@+id/button1"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_below="@+id/textView2"
android:layout_centerHorizontal="true"
android:layout_marginTop="68dp"
android:onClick="storeInDB"
android:text="Save" />
```

```
<EditText
android:id="@+id/editText2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/textView2"
android:layout_alignBottom="@+id/textView2"
android:layout_alignRight="@+id/button1"
android:ems="10"
android:hint="Mobile Number"
android:inputType="phone" />
```

```
<EditText
android:id="@+id/editText1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/textView1"
android:layout_alignBottom="@+id/textView1"
android:layout_alignLeft="@+id/editText2"
android:ems="10"
android:hint="Person Name"
android:inputType="textPersonName" >
```

```
<requestFocus />  
</EditText>
```

```
<Button  
  android:id="@+id/button2"  
  style="?android:attr/buttonStyleSmall"  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:layout_alignLeft="@+id/button1"  
  android:layout_alignParentBottom="true"  
  android:layout_marginBottom="21dp"  
  android:onClick="instructions"  
  android:text="Instructions" />
```

```
<Button  
  android:id="@+id/button3"  
  style="?android:attr/buttonStyleSmall"  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:layout_alignRight="@+id/button1"  
  android:layout_alignTop="@+id/button2"  
  android:onClick="display"  
  android:text="View Registered" />
```

```
</RelativeLayout>
```

3. Verify:

Mobile application development

```
<RelativeLayoutxmlns: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: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=".Verify" >
```

```
<Button
android:id="@+id/button1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentBottom="true"
android:layout_centerHorizontal="true"
android:layout_marginBottom="176dp"
android:onClick="verify_no"
android:text="Submit" />
```

```
<EditText
android:id="@+id/editText1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_above="@+id/button1"
android:layout_centerHorizontal="true"
android:layout_marginBottom="76dp"
android:ems="10"
android:hint="Your Phone Number"
android:inputType="phone" >
```

```
<requestFocus />
</EditText>
```

```
<Button
android:id="@+id/button2"
```

Mobile application development

```
style="?android:attr/buttonStyleSmall"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentBottom="true"
android:layout_centerHorizontal="true"
android:layout_marginBottom="94dp"
android:onClick="back"
android:text="Back" />
```

```
</RelativeLayout>
```

4. Instructions:

```
<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: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=".Instructions" >
```

```
<TextView
android:id="@+id/textView1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentTop="true"
android:layout_centerHorizontal="true"
android:layout_marginTop="16dp"
android:text="INSTRUCTIONS"
android:textAppearance="?android:attr/textAppearanceLarge" />
```

```
<Button
```

Mobile application development

```
android:id="@+id/button1"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_below="@+id/help"  
android:layout_centerHorizontal="true"  
android:layout_marginTop="34dp"  
android:onClick="back"  
android:text="Main Menu" />
```

```
<TextView  
android:id="@+id/help"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignLeft="@+id/button1"  
android:layout_below="@+id/textView1"  
android:layout_marginTop="15dp"  
android:text="@string/help" />
```

```
</RelativeLayout>
```

5. Display:

```
<RelativeLayoutxmlns: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: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=".Display" >
```

```
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_centerHorizontal="true"
```

```
android:text="Shake your phone to TEST." />
```

```
<Button
android:id="@+id/button1"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_alignParentBottom="true"
android:layout_centerHorizontal="true"
android:onClick="back"
android:text="Main Menu" />
```

```
</RelativeLayout>
```

//JAVA CODE

1. AccelerometerManager.java

```
package com.sid.womensafetyapp;

import java.util.List;
import android.content.Context;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.widget.Toast;

public class AccelerometerManager {

    private static Context aContext=null;

    /** Accuracy configuration */
    private static float threshold = 15.0f;
    private static int interval = 200;

    private static Sensor sensor;
    private static SensorManager sensorManager;
    // you could use an OrientationListener array instead
    // if you plans to use more than one listener
    private static AccelerometerListener listener;

    /** indicates whether or not Accelerometer Sensor is supported */
    private static Boolean supported;
    /** indicates whether or not Accelerometer Sensor is running */
    private static boolean running = false;

    /**
     * Returns true if the manager is listening to orientation changes
     */
    public static boolean isListening() {
        return running;
    }

    /**
```

Mobile application development

```
        * Unregisters listeners
        */
    public static void stopListening() {
        running = false;
        try {
            if (sensorManager != null && sensorEventListener != null) {
                sensorManager.unregisterListener(sensorEventListener);
            }
        } catch (Exception e) {}
    }

    /**
     * Returns true if at least one Accelerometer sensor is available
     */
    public static boolean isSupported(Context context) {
        aContext = context;
        if (supported == null) {
            if (aContext != null) {

                sensorManager = (SensorManager) aContext.
                    getSystemService(Context.SENSOR_SERVICE);

                // Get all sensors in device
                List<Sensor> sensors = sensorManager.getSensorList(
                    Sensor.TYPE_ACCELEROMETER);

                supported = new Boolean(sensors.size() > 0);

            } else {
                supported = Boolean.FALSE;
            }
        }
        return supported;
    }
}
```

2. BgService.java

Mobile application development

```
package com.sid.womensafetyapp;

import android.annotation.SuppressLint;
import android.app.Service;
import android.content.Context;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.os.Handler;
import android.os.HandlerThread;
import android.os.IBinder;
import android.os.Looper;
import android.os.Message;
import android.telephony.SmsManager;
import android.util.Log;
import android.widget.Toast;

@SuppressLint("HandlerLeak")

public class BgService extends Service implements AccelerometerListener {

    String str_address;

    private Looper mServiceLooper;
    private ServiceHandler mServiceHandler;

    // Handler that receives messages from the thread.
    private final class ServiceHandler extends Handler {

        public ServiceHandler(Looper looper) {

            super(looper);
        }

        @Override
        public void handleMessage(Message msg) {
```

Mobile application development

```
// REPLACE THIS CODE WITH YOUR APP CODE

// Wait before Toasting Service Message
// to give the Service Started message time to display.

// Toast Service Message.
/*      Context context = getApplicationContext();
        CharSequence text = "Service Message";
        int duration = Toast.LENGTH_LONG;
        Toast toast = Toast.makeText(context, text, duration);
        toast.show();
*/

// Service can stop itself using the stopSelf() method.
// Not using in this app. Example statement shown below.
//stopSelf(msg.arg1);
}

}

@Override
public IBinder onBind(Intent arg0) {

    return null;
}

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

    if (AccelerometerManager.isSupported(this)) {

AccelerometerManager.startListening(this);
    }
    HandlerThread thread = new
HandlerThread("ServiceStartArguments", android.os.Process.THREAD_PRIORITY_BACKGROUND);
    thread.start();
}
```

Mobile application development

```
mServiceLooper = thread.getLooper();

mServiceHandler = new ServiceHandler(mServiceLooper);
}

@Override
public int onStartCommand(Intent intent, int flags, int startId) {

    // Get message from message pool using handler.
    Message msg = mServiceHandler.obtainMessage();

    // Set start ID (unique to the specific start) in message.
    msg.arg1 = startId;

    // Send message to start job.
    mServiceHandler.sendMessage(msg);

    // Toast Service Started message.
    // Context context = getApplicationContext();

    /* CharSequence text = "Service Started";
       int duration = Toast.LENGTH_SHORT;
       Toast toast = Toast.makeText(context, text, duration);
       toast.show();
    */

    // Start a sticky.
    return START_STICKY;
}

public class GeocoderHandler extends Handler {
    @Override
```

Mobile application development

```
public void handleMessage(Message message) {

    Toast.makeText(getApplicationContext(), "geocoderhandler started",
    Toast.LENGTH_SHORT).show();

    switch (message.what) {
    case 1:

        Bundle bundle = message.getData();
        str_address = bundle.getString("address");
        // TelephonyManager
        tmgr=(TelephonyManager)BgService.this.getSystemService(Context.TELEPHONY_SERVICE);
        // Stringph_number=tmgr.getLine1Number();
        SQLiteDatabasedb;
        db=openOrCreateDatabase("NumDB", Context.MODE_PRIVATE, null);
        Cursor c=db.rawQuery("SELECT * FROM details", null);
        Cursor c1=db.rawQuery("SELECT * FROM SOURCE", null);

        String source_ph_number=c1.getString(0);
        while(c.moveToNext())
        {
            String target_ph_number=c.getString(1);

            // SmsManagersmsManager=SmsManager.getDefault();
            // smsManager.sendTextMessage("+918121668944", "+918121668944", "Please help me. I
            need help immediately. This is where i am now:"+str_address, null, null);

            Toast.makeText(getApplicationContext(),
            "Source:"+source_ph_number+"Target:"+target_ph_number, Toast.LENGTH_SHORT).show();

        }

        db.close();

        break;
        default:
            str_address = null;
            }
            Toast.makeText(getApplicationContext(), str_address, Toast.LENGTH_SHORT).show();

        }
    }
```

Mobile application development

```
}
```

```
@Override
```

```
public void onAccelerationChanged(float x, float y, float z) {
```

```
    // TODO Auto-generated method stub
```

```
}
```

```
@Override
```

```
public void onShake(float force) {
```

```
    GPSTracker gps;
```

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

```
    if(gps.canGetLocation()){
```

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

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

```
        RGeocoder RGeocoder = new RGeocoder();
```

```
        RGeocoder.getAddressFromLocation(latitude, longitude, getApplicationContext(), new  
GeocoderHandler());
```

```
        Toast.makeText(getApplicationContext(), "onShake", Toast.LENGTH_SHORT).show();
```

```
    }
```

```
else{
```

```
    gps.showSettingsAlert();
```

```
    }
```

```
}
```

Mobile application development

```
// onDestroy method. Display toast that service has stopped.
@Override
public void onDestroy() {
    super.onDestroy();

    // Toast Service Stopped.
    Context context = getApplicationContext();

    Log.i("Sensor", "Service distroy");

    if (AccelerometerManager.isListening()) {

        AccelerometerManager.stopListening();

    }

    CharSequence text = "Women Safety App Service Stopped";
    int duration = Toast.LENGTH_SHORT;
    Toast toast = Toast.makeText(context, text, duration);
    toast.show();

}

}
```

3. Verify.java

Mobile application development

```
package com.sid.womensafetyapp;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;
import android.support.v4.app.NavUtils;
import android.annotation.TargetApi;
import android.content.Context;
import android.content.Intent;
import android.database.sqlite.SQLiteDatabase;
import android.os.Build;

public class Verify extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_verify);
        // Show the Up button in the action bar.
        setupActionBar();
    }

    public void verify_no(View v) {
        EditText source_no = (EditText) this.findViewById(R.id.editText1);
        String str_source_no = source_no.getText().toString();
        SQLiteDatabase db;
        db = openOrCreateDatabase("NumDB", Context.MODE_PRIVATE, null);
        // if(source_no.getText() != null){

        db.execSQL("CREATE TABLE IF NOT EXISTS source(number VARCHAR);");
        db.execSQL("INSERT INTO source VALUES('"+str_source_no+"');");
    }
}
```

Mobile application development

```
        Toast.makeText(getApplicationContext(), str_source_no+" Successfully
Saved",Toast.LENGTH_SHORT).show();
        db.close();
        back(v);
//    }
//    else{
//        Toast.makeText(getApplicationContext(), "Enter Your
Number.",Toast.LENGTH_SHORT).show();
//    }
}

/**
 * Set up the {@link android.app.ActionBar}, if the API is available.
 */
@TargetApi(Build.VERSION_CODES.HONEYCOMB)
private void setupActionBar() {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
        getActionBar().setDisplayHomeAsUpEnabled(true);
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.verify, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:
            // This ID represents the Home or Up button. In the case of this
            // activity, the Up button is shown. Use NavUtils to allow users
            // to navigate up one level in the application structure. For
            // more details, see the Navigation pattern on Android Design:
            //
            // http://developer.android.com/design/patterns/navigation.html#up-vs-back

```


Mobile application development

```
        //
        NavUtils.navigateUpFromSameTask(this);
        return true;
    }
    return super.onOptionsItemSelected(item);
}

public void back(View v) {
    Intent i_back=new Intent(Verify.this,MainActivity.class);
    startActivity(i_back);

}

}
```

4. Display.java

```
package com.sid.womensafetyapp;

import android.app.Activity;
import android.app.AlertDialog.Builder;
import android.content.Context;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;

public class Display extends Activity{

    Cursor c;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_display);
    }
}
```

```
        SQLiteDatabase db;
        db=openOrCreateDatabase("NumDB", Context.MODE_PRIVATE, null);

        c=db.rawQuery("SELECT * FROM details", null);
        if(c.getCount()==0)
        {
            showMessage("Error", "No records found.");
            return;
        }
        StringBuffer buffer=new StringBuffer();
        while(c.moveToNext())
        {
            buffer.append("Name: "+c.getString(0)+"\n");
            buffer.append("Number: "+c.getString(1)+"\n");
        }
        showMessage("Details", buffer.toString());
        Intent i_startservice=new Intent(Display.this,BgService.class);
        startService(i_startservice);
    }

    public void showMessage(String title,String message)
    {
        Builder builder=new Builder(this);
        builder.setCancelable(true);
        builder.setTitle(title);
        builder.setMessage(message);
        builder.show();
    }

    public void back(View v) {
        Intent i_back=new Intent(Display.this,MainActivity.class);
        startActivity(i_back);

    }

}
```

5. RGeocoder.java

Mobile application development

```
packagecom.sid.womensafetyapp;

importjava.io.IOException;
importjava.util.List;
importjava.util.Locale;

importandroid.content.Context;
importandroid.location.Address;
importandroid.location.Geocoder;
importandroid.os.Bundle;
importandroid.os.Handler;
importandroid.os.Message;
importandroid.util.Log;

public class RGeocoder {

    private static final String TAG = "LocationAddress";

    public void getAddressFromLocation(final double latitude, final double longitude,
final Context context, final Handler handler) {

        Thread thread = new Thread() {
            @Override
            public void run() {
                Geocodergeocoder = 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);
                        StringBuildersb = 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");
                    }
                }
            }
        };
        thread.start();
    }
}
```

Mobile application development

```
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\nUnable to get address for this lat-long.";
        bundle.putString("address", result);
        message.setData(bundle);
    }
    message.sendToTarget();
}
};
thread.start();
}
```

Outputs:Home Page

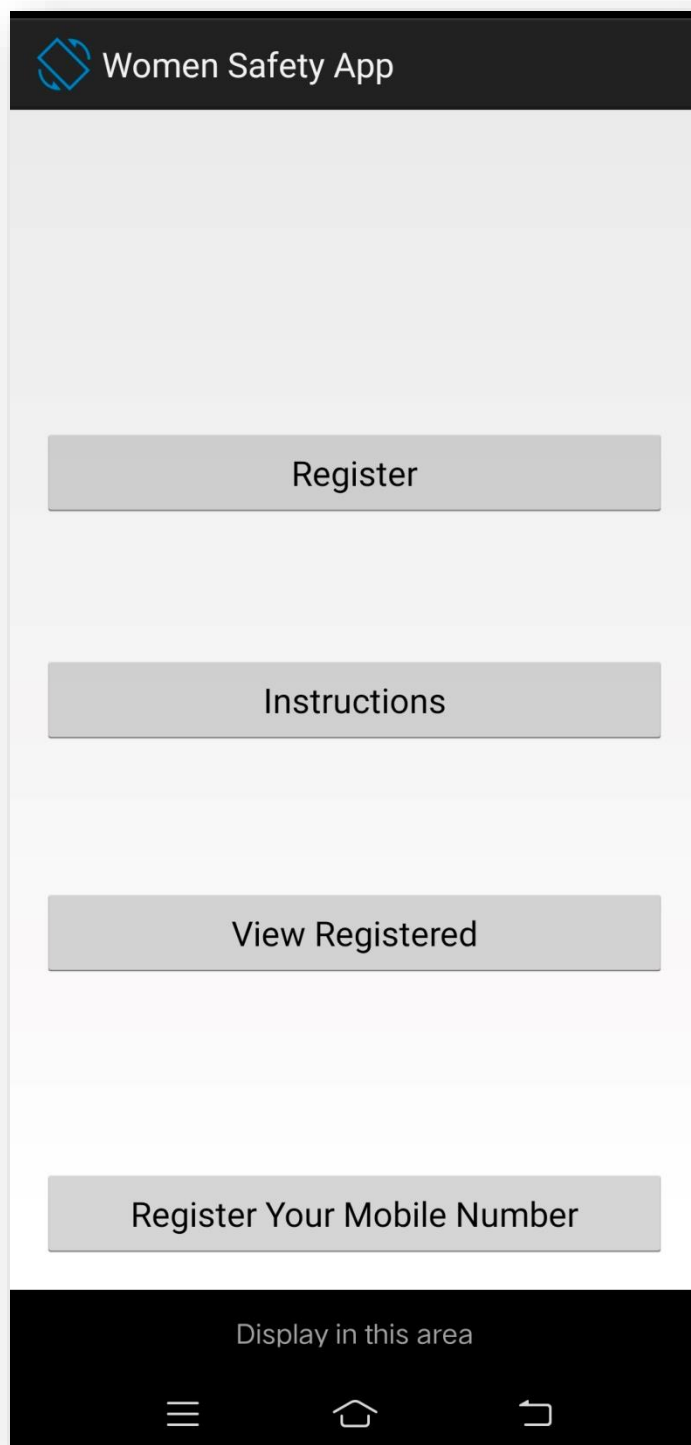
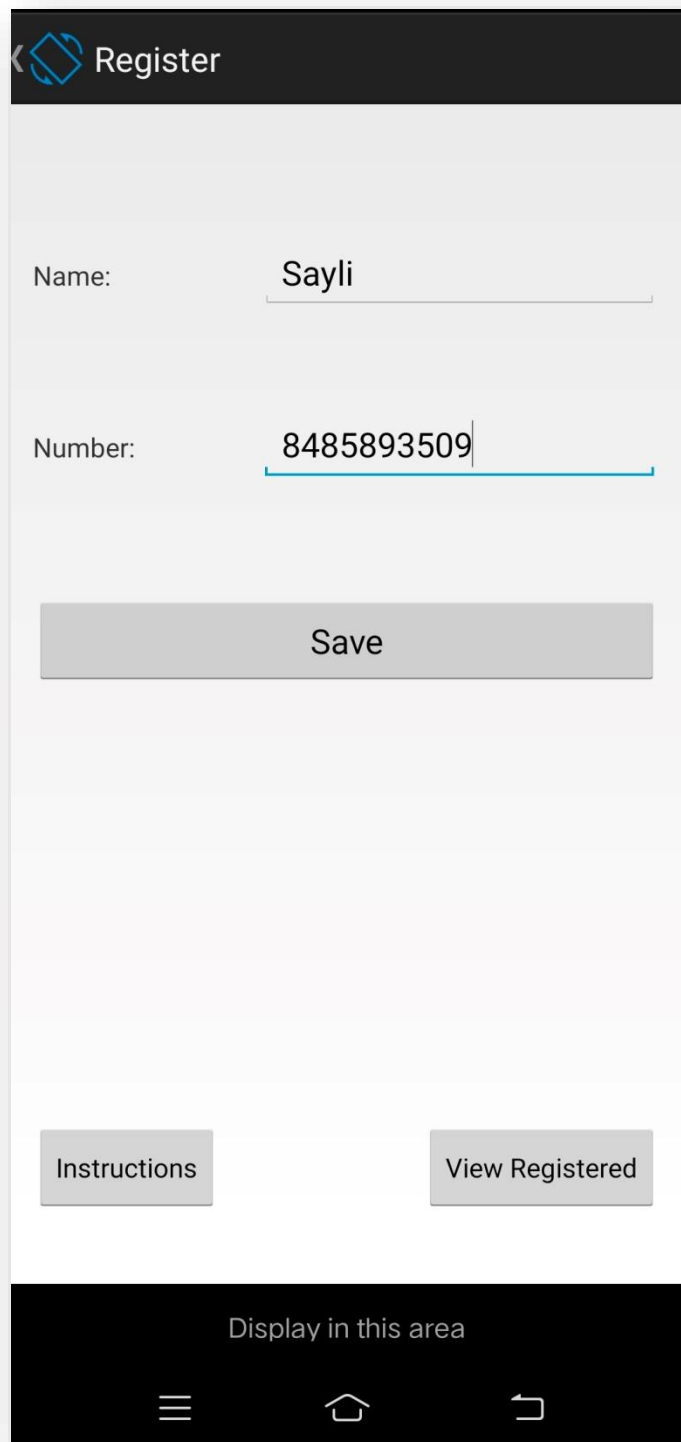


Fig. Home

Register Page



The image shows a mobile application register page. At the top, there is a dark header bar with a back arrow icon and the text "Register". Below the header, the page has a light gray background. There are two input fields: "Name:" with the value "Sayli" and "Number:" with the value "8485893509". Below these fields is a large gray button labeled "Save". At the bottom of the form area, there are two smaller gray buttons: "Instructions" and "View Registered". The bottom of the screen features a dark navigation bar with the text "Display in this area" and three icons: a hamburger menu, a home icon, and a back arrow.

Register

Name: Sayli

Number: 8485893509

Save

Instructions View Registered

Display in this area

Fig. Register

Verify Page

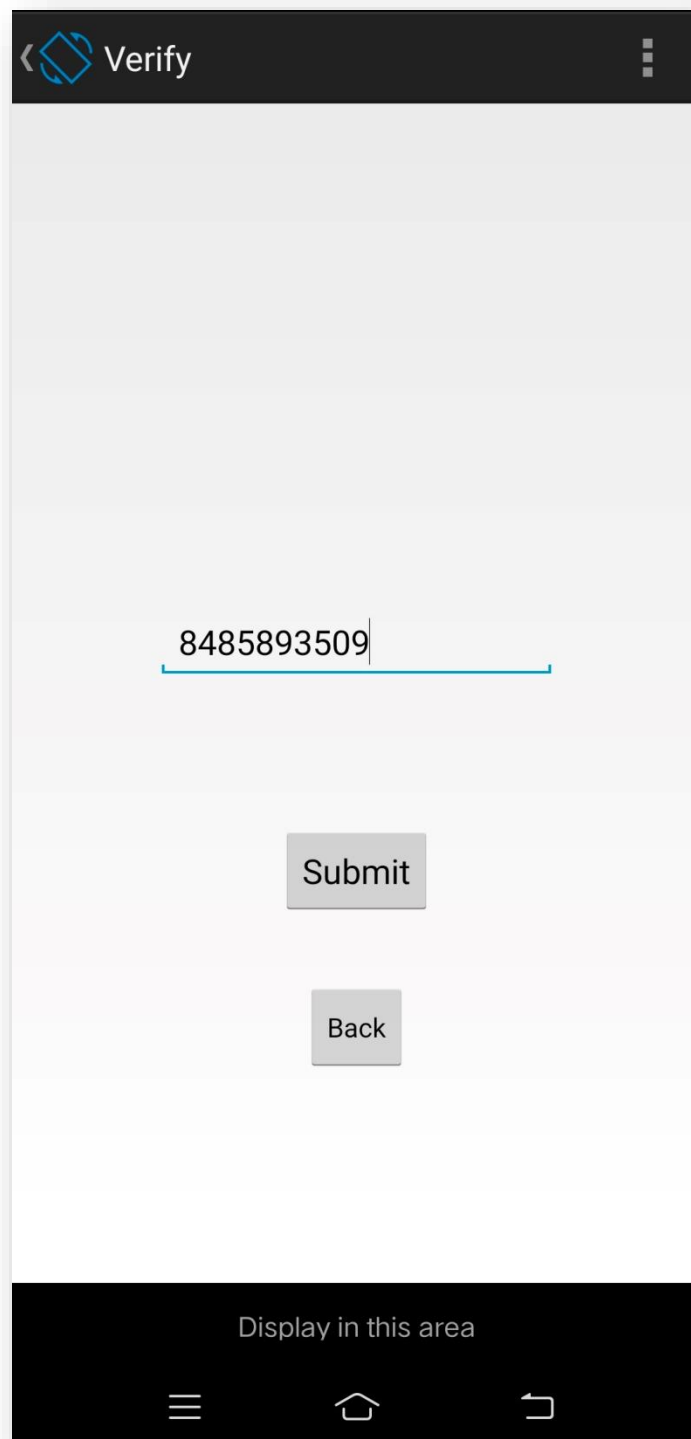


Fig. Verify

Instructions Page

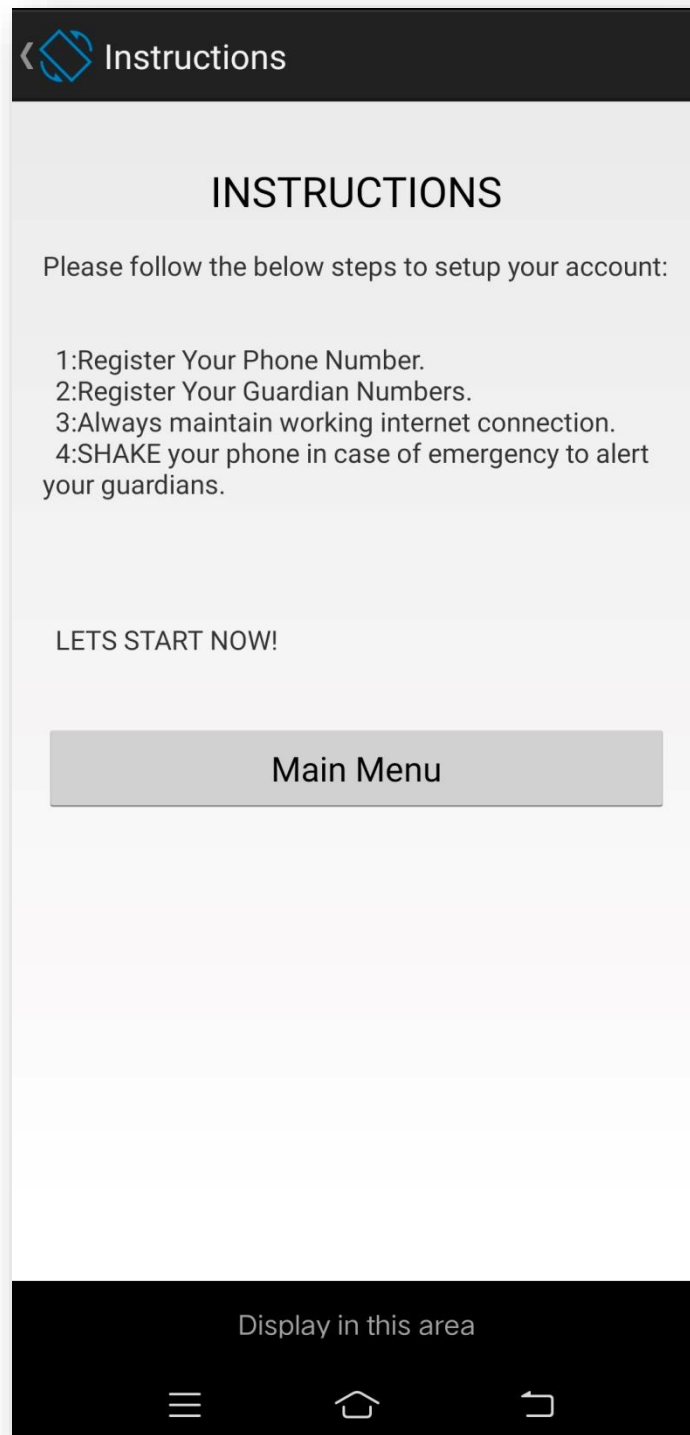


Fig. Instruction

Display Page

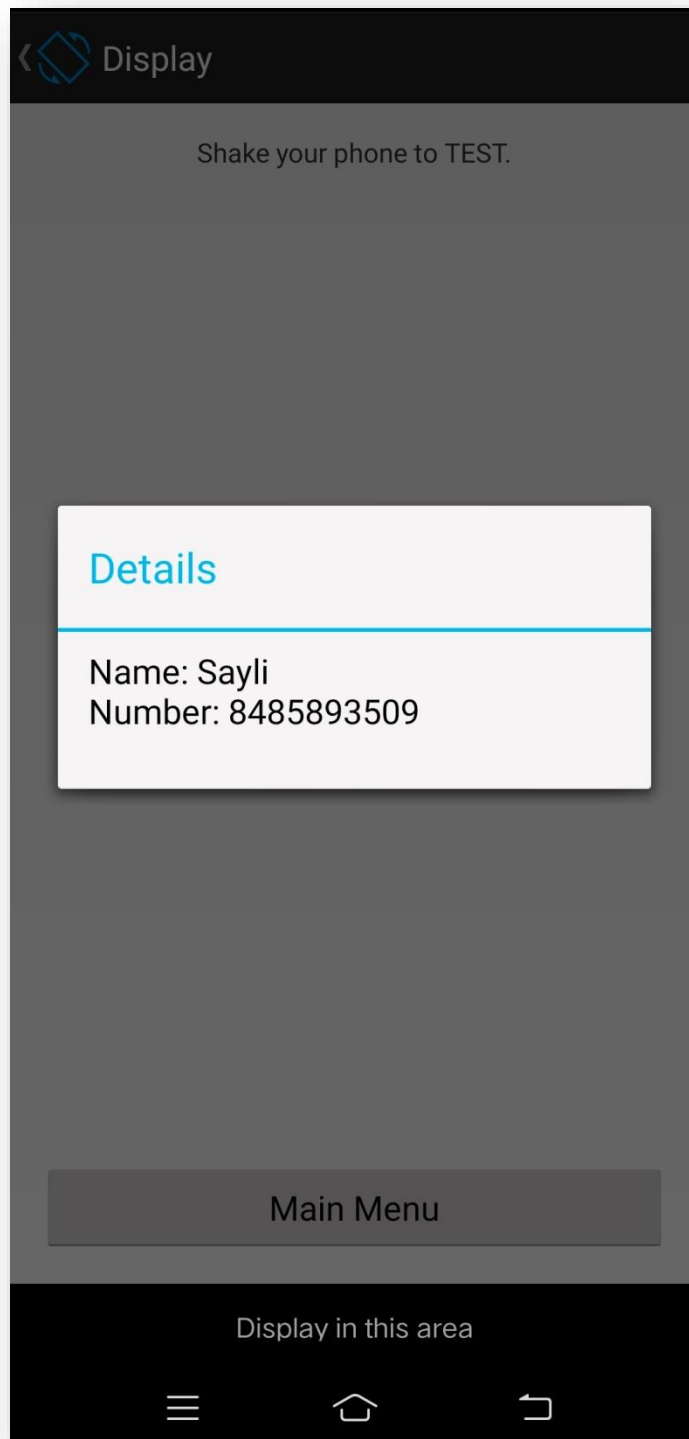


Fig. Display

7.0 Skilled Developed/Learning Outcome of thisMicro-Project:

- a. Programming skills.
- b. Application development
- c. Team work.
- d. Technical skills.
- e. Communication skills.
- f. Handling android studio

From this project we learn how to build actual application through coding.

8.0 Scope ofthisMicro-Project:

As an independent nation, we must ensure respect and security of women and we cannot deny them this basic right. It is now time to initiate action to eradicate the menace of security issues with women. Violence against women remains embedded in our societies, both as a daily reality and a difficult situations .Gender justice is impossible in a world where at least one in three women faces violence in her lifetime, regardless of her culture, religion, socioeconomic class, or education level Our country can be a true democracy only when all women have the security and freedom from violence.

Team Members with Roll Nos.

- 6) Anuradha Partudkar[30]
- 7) Pawar Shweta [31]
- 8) SalunkeChetna .[35]
- 9) Shinde Menka .[36]
- 10) Zare Sangita.[43]