TABLE OF CONTENTS

List of Figures

ABSTRACT

Safety for women has become a major issue as the day-by-day cases increasing all over the world. This proposed project explains women's safety and their security byusing an application to send alert messages, showing safe locations & alert authorities. It suggests a replacement view to the use of technology to guard women.

In this project, an android based smartphone with an integrated feature that alerts and provides location-based information. It provides self-defence and SMS alerts when a woman is in danger. The voice recognition will happen when the woman is shouting. It will recognize the voice and send the alert message to the contact and send alert SMS with (GPS) Global positioning system location to emergency contacts.

INTRODUCTION

Introduction to Android Development

Android software development is the process by which applications are created for devices running the Android operating system. Google states that "Android apps can be written using Kotlin, Java, and C++ languages" using the Android software development kit (SDK), while using other languages is also possible. All non-Java virtual machine (JVM) languages, such as Go, JavaScript, C, C++ or assembly, need the help of JVM language code, that may be supplied by tools, likely with restricted API support. Some programming languages and tools allow cross-platform app support (i.e. for both Android and iOS). Third party tools, development environments, and language support have also continued to evolve and expand since the initial SDK was released in 2008. The official Android app distribution mechanism to end users is Google Play; it also allows staged gradual app release, as well as distribution of pre-release app versions to testers.

Even though most of Android OS is open source, phones come packaged with closed-source Google applications for functionality such as the Google Play and GPS navigation. Google has asserted that these applications can only be provided through approved distribution channels by licensed distributors. Android provides its own GUI classes, and does not provide Java AWT, Swing or JavaFX. It does not support the full Java Beans API.
Android was created by the Open Handset Alliance, which is led by Google. The early feedback on developing applications for the Android platform was mixed. Issues cited include bugs, lack of documentation, inadequate QA infrastructure, and no public issue-tracking system. As of July 2013, more than one million applications have been developed for Android, with over 25 billion downloads. Android smartphone shipments are forecast to exceed 1.2 billion units in 2018 with an 85% market share.
The first publicly available application was the Snake game.

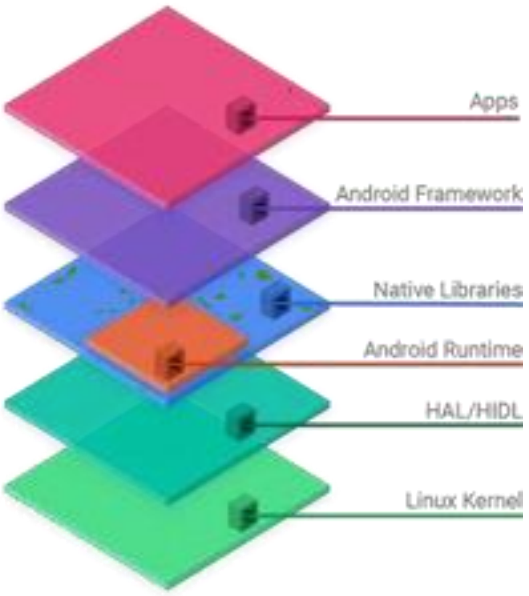


Fig. 1.1: The Android Stack

(Google announced an issue tracker on January 18, 2008.) In December 2007, Merge Lab mobile start-up founder Adam MacBeth stated, "Functionality is not there, is poorly documented or just doesn't work... It's clearly not ready for prime time." Despite this, Android-targeted applications began to appear the week after the platform was announced.

Applications of Android Development

Android initially came into existence with the sure-fire idea that developments are given the power and freedom to create enthralling Mobile applications while taking advantage of everything that the mobile handset has to offer.

Android is built on the open Linux Kernel. This particular software for Mobile applications is made to be open source, thereby giving the opportunity to the developers to introduce and incorporate any technological advancement. Build on a custom virtual machine android gives its users the additional usage and application power, to initiate an interactive and efficient application and operational software for your phone. Android software development is the process by which applications are created for devices running the Android operating system. Google states that "Android apps can be written using Kotlin, Java, and C++ languages" using the Android software development kit (SDK), while using other languages is also possible.

Features of Android

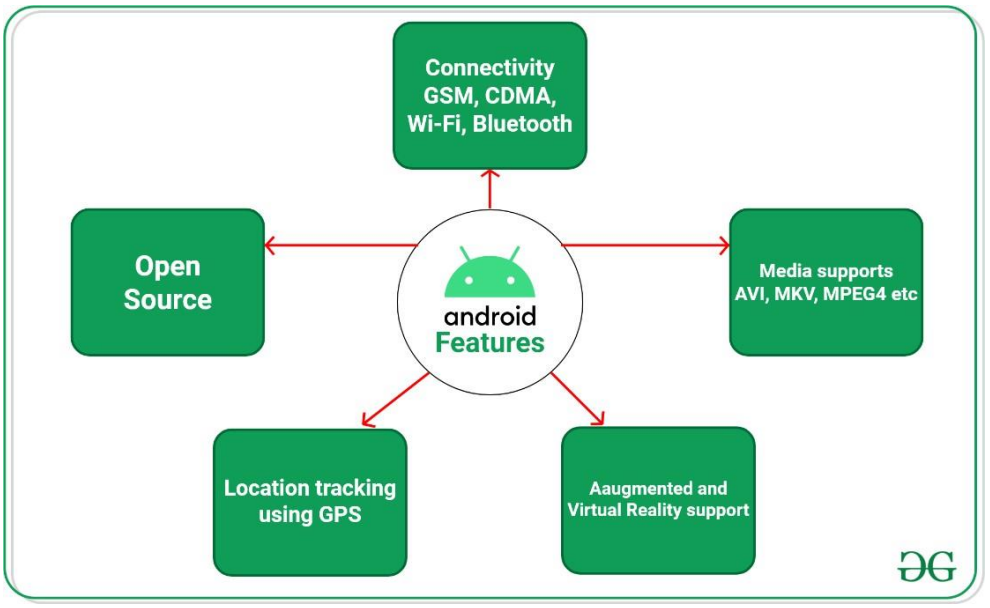


Fig. 1.2: The Android Features

Android is a powerful open-source operating system that open-source provides immense features and some of these are listed below.

- Android Open-Source Project so we can customize the OS based on our requirements.
- Android supports different types of connectivity for GSM, CDMA, Wi-Fi, Bluetooth, etc. for telephonic conversation or data transfer.
- Using wifi technology we can pair with other devices while playing games or using other applications.
- It contains multiple APIs to support location-tracking services such as GPS.
- We can manage all data storage-related activities by using the file manager.
- It contains a wide range of media supports like AVI, MKV, FLV, MPEG4, etc. to play or record a variety of audio/video.
- It also supports different image formats like JPEG, PNG, GIF, BMP, MP3, etc.
- It supports multimedia hardware control to perform playback or recording using a camera and microphone.
- Android has an integrated open-source WebKit layout-based web browser to support User Interfaces like HTML5, and CSS3.
- Android supports multi-tasking means we can run multiple applications at a time and can switch between them.
- It provides support for virtual reality or 2D/3D Graphics.

Android Versions

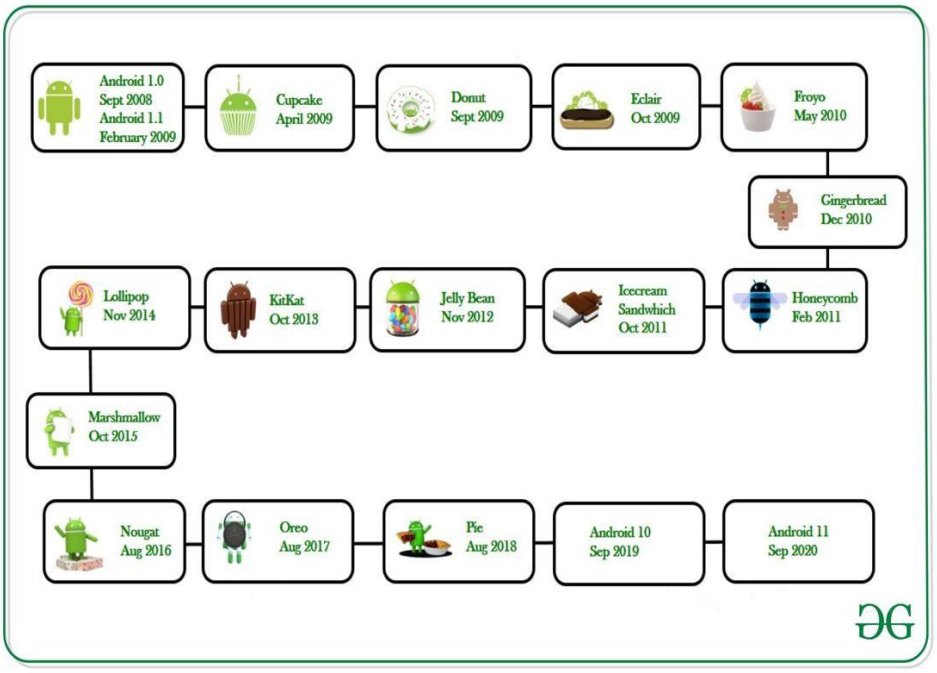


Fig. 1.3: The Android Versions

Google first publicly announced Android in November 2007 but was released on 23 September 2008 to be exact. The first device to bring Android into the market was the HTC Dream with the version Android 1.0. Since then, Google released a lot of android versions such as Apple Pie, Banana Bread, Cupcake, Donut, Éclair, Froyo, Gingerbread, Jellybeans, KitKat, Lollipop, marshmallow, Nougat, Oreo, etc. with extra functionalities and new features.

Programming Languages used in Developing Android Applications:-

- Java
- Kotlin

Developing the Android Application using Kotlin is preferred by Google, as Kotlin is made an official language for Android Development, which is developed and maintained by JetBrains. Previously before Java is considered the official language for Android Development. Kotlin is made official for Android Development in Google I/O 2017.

Advantages of Android Development

- The Android is an open-source Operating system and hence possesses a vast community for support.
- The design of the Android Application has guidelines from Google, which becomes easier for developers to produce more intuitive user applications.
- Fragmentation gives more power to Android Applications. This means the application can run two activities on a single screen.
- Releasing the Android application in the Google play store is easier when it is compared to other platforms.

Disadvantages of Android Development

- Fragmentation provides a very intuitive approach to user experience but it has some drawbacks, where the development team needs time to adjust to the various screen sizes of mobile smartphones that are now available in the market and invoke the particular features in the application.
- The Android devices might vary broadly. So the testing of the application becomes more difficult.
- As the development and testing consume more time, the cost of the application may increase, depending on the application's complexity and features.

Android Architecture

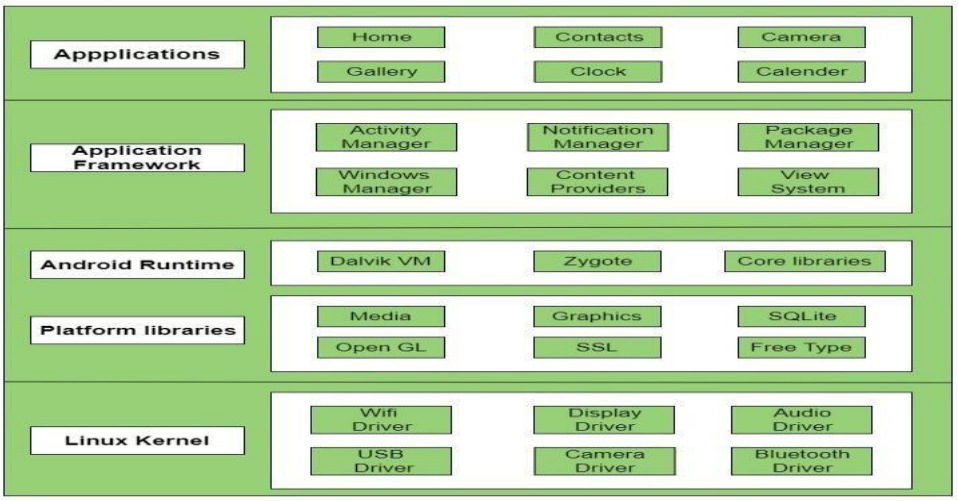


Fig. 1.4: Android Architecture

The android architecture contains a different number of components to support any android device's needs. Android software contains an open-source Linux Kernel having a collection of a number of C/C++ libraries which are exposed through application framework services.

Among all the components Linux Kernel provides the main functionality of operating system functions for smartphones and Dalvik Virtual Machine (DVM) provides a platform for running an android application.

The main components of the android architecture are the following:-

- Applications
- Application Framework
- Android Runtime
- Platform Libraries
- Linux Kernel

Chapter 2

SYSTEM REQUIREMENTS

2.1 Software Requirements

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. The following are the software requirements for the application:

- Operating System : Windows 10
- Java Development kit
- Android Studio

2.2 Hardware Requirements

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware.

- CPU : Intel or AMD processor.
- Cores : Dual-Core (Quad-Core recommended).
- RAM : minimum 4GB (>4GB recommended).
- Graphics : Intel Integrated Graphics or AMD Equivalent.
- Display Resolution : 1366x768 (1920x1080 recommended).

DESCRIPTION

The sequence of the executions within the system are in the following order:

- When the user registers their contact, they need to register their details along with emergency contact.

When they have problem, they need to press SOS button so that emergency message will send to the contacts along with live location.

Along with the process audio recording option also activated and they get the help on time. They can also further can view their safe region.

Initially the user needs to login to the system with their registered email id and they need to provide the emergency contact details. They can also sign up through their email id and password. After login, the system needs to be turned on, then it will start working in the background. After getting the command the system will start working following the previously discussed methods.

During the emergency user needs to press the start service button and the emergency message already stored has been send to the contacts and also audio recording started and live streaming of location send to the user emergency contact stored. This application has also the feature of finding the save hospital region and police station region shown as safe region. Some safety tips also provided in them along with the defence tips. In this project related many android applications in the related work part having similar functions to this application. But in this application, having some unique features which makes it different from other existing systems.

They are Safe Zone, though there is another app which shows the safe zone to the victim but the working procedure is not similar. To show the safe places other app needs to use the app user who will tag himself/herself as a safe place and victim will reach them out. But if there is no one to mark themselves at the emergency moment then the victim will not get this option. Also, it is mandatory to be a user of the app to mark their place as safe zone. But in this system, have marked the police stations as safe zone. Therefore, it will be easier and reliable for the victim to get the place which will be shown in the maps. In any case if any criminals use the app to mark themselves as a safe zone it will be problem for the victim in that case. So, in this project tagging police station as the safe zone. They can also be able to see some safety tips

Chapter 3

Source Code and Snapshots

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayoutxmlns:android="http://schemas.andr
oid.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"
tools:context=".MainActivity">
<Button
android:id="@+id/button_clear"
android:layout_width="87dp"
android:layout_height="53dp"
android:layout_marginTop="30dp"
android:text="C"
app:layout_constraintStart_toStartOf="@+id/button_add"
app:layout_constraintTop_toBottomOf="@+id/button_add" />
<Button
android:id="@+id/button_sub"
android:layout_width="87dp"
android:layout_height="53dp"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="-"
app:layout_constraintStart_toEndOf="@+id/button_three"
app:layout_constraintTop_toBottomOf="@+id/button_mul" />
<Button
android:id="@+id/button_add"
android:layout_width="87dp"
android:layout_height="53dp"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="+"
app:layout_constraintStart_toEndOf="@+id/button_equal"
app:layout_constraintTop_toBottomOf="@+id/button_sub" />
<Button
android:id="@+id/button_mul"
android:layout_width="87dp"
android:layout_height="53dp"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="*"
```

```
app:layout_constraintStart_toEndOf="@+id/button_six"

app:layout_constraintTop_toBottomOf="@+id/button_div" />

<Button
android:id="@+id/button_equal"
android:layout_width="62dp"
android:layout_height="53dp"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="="
app:layout_constraintStart_toEndOf="@+id/button_zero"
app:layout_constraintTop_toBottomOf="@+id/button_three" />
<Button
android:id="@+id/button_zero"
android:layout_width="62dp"
android:layout_height="53dp"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="0"
app:layout_constraintStart_toEndOf="@+id/button_dot"
app:layout_constraintTop_toBottomOf="@+id/button_two" />
<Button
android:id="@+id/button_dot"
android:layout_width="62dp"
android:layout_height="53dp"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="."
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/button_one" />
<Button
android:id="@+id/button_three"
android:layout_width="62dp"
android:layout_height="53dp"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="3"
app:layout_constraintStart_toEndOf="@+id/button_two"
app:layout_constraintTop_toBottomOf="@+id/button_six" />
<Button
android:id="@+id/button_two"
android:layout_width="62dp"
android:layout_height="53dp"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="2"
app:layout_constraintStart_toEndOf="@+id/button_one"
app:layout_constraintTop_toBottomOf="@+id/button_five" />
```

```
<Button
android:id="@+id/button_one"
android:layout_width="62dp"
android:layout_height="53dp"




android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="1"
app:layout_constraintStart_toStartOf="parent"


app:layout_constraintTop_toBottomOf="@+id/button_four"  />
<Button
android:id="@+id/button_six"
android:layout_width="62dp"
android:layout_height="53dp"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="6"
app:layout_constraintStart_toEndOf="@+id/button_five"
app:layout_constraintTop_toBottomOf="@+id/button_nine" />
```

Activity Main.java

Accelerometer Listener

```
package com.prabhu.womensafetyapp;


public interface AccelerometerListener {

    public void onAccelerationChanged(float x, float y, float z);

    public void onShake(float force);

}
```

Accelerometer Manager

```
package com.prabhu.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;
    }

    /**
     * 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;
    }

    /**
     * Configure the listener for shaking
     * @param threshold
     *             minimum acceleration variation for considering shaking
     * @param interval
     *             minimum interval between to shake events
     */
```

```
    public static void configure(int threshold, int interval) {
        AccelerometerManager.threshold = threshold;
        AccelerometerManager.interval = interval;
    }

    /**
     * Registers a listener and start listening
     * @param accelerometerListener
     *             callback for accelerometer events
     */
    public static void startListening( AccelerometerListener
accelerometerListener )
    {

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

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

        if (sensors.size() > 0) {

            sensor = sensors.get(0);

            // Register Accelerometer Listener
            running = sensorManager.registerListener(
                    sensorEventListener, sensor,
                    SensorManager.SENSOR_DELAY_GAME);

            listener = accelerometerListener;
        }


    }

    /**
     * Configures threshold and interval
     * And registers a listener and start listening
     * @param accelerometerListener
     *             callback for accelerometer events
     * @param threshold

              minimum acceleration variation for considering shaking
     * @param interval
     *             minimum interval between to shake events
     */
    public static void startListening(
            AccelerometerListener accelerometerListener,
            int threshold, int interval) {
        configure(threshold, interval);
        startListening(accelerometerListener);
    }
```

```
    /**
     * The listener that listen to events from the accelerometer listener
     */
    private static SensorEventListener sensorEventListener =
        new SensorEventListener() {

        private long now = 0;
        private long timeDiff = 0;
        private long lastUpdate = 0;
        private long lastShake = 0;

        private float x = 0;
        private float y = 0;
        private float z = 0;
        private float lastX = 0;
        private float lastY = 0;
        private float lastZ = 0;
        private float force = 0;

        public void onAccuracyChanged(Sensor sensor, int accuracy) {}

        public void onSensorChanged(SensorEvent event) {
            // use the event timestamp as reference
            // so the manager precision won't depends
            // on the AccelerometerListener implementation
            // processing time
            now = event.timestamp;

            x = event.values[0];
            y = event.values[1];
            z = event.values[2];

            // if not interesting in shake events
            // just remove the whole if then else block
          if (lastUpdate == 0) {
                lastUpdate = now;
                lastShake = now;
                lastX = x;
                lastY = y;
                lastZ = z;
                Toast.makeText(aContext,"No Motion detected",
                   Toast.LENGTH_SHORT).show();


      } else {
                timeDiff = now - lastUpdate;

                if (timeDiff > 0) {

                    /*force = Math.abs(x + y + z - lastX - lastY - lastZ)
                                / timeDiff;*/
                    force = Math.abs(x + y + z - lastX - lastY - lastZ);

                    if (Float.compare(force, threshold) >0 ) {
```

```
                        //Toast.makeText(Accelerometer.getContext(),
                        //(now-lastShake)+" >= "+interval, 1000).show();

                        if (now - lastShake >= interval) {

                            // trigger shake event
                            listener.onShake(force);
                        }
                        else
                        {
                            Toast.makeText(aContext,"No Motion detected.",
                                Toast.LENGTH_SHORT).show();

                        }
                        lastShake = now;
                    }
                    lastX = x;
                    lastY = y;
                    lastZ = z;
                    lastUpdate = now;
                }
                else
                {
                    Toast.makeText(aContext,"No Motion detected",
Toast.LENGTH_SHORT).show();

                }
            }

            // trigger change event
            listener.onAccelerationChanged(x, y, z);
        }

    };

}
```

BG Service

```
package com.prabhu.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.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) {

           // 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_B
ACKGROUND);
          thread.start();

          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
         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_S
ERVICE);
                  // String ph_number=tmgr.getLine1Number();
                SQLiteDatabase db;
                      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);

            //          SmsManager smsManager=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();


}
    }



     @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();
            }

      }




      // 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();


      }
```

```
}
```

Main Activity

```
package com.prabhu.womensafetyapp;


import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;

public class MainActivity extends Activity {

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


      public void register(View v) {
            Intent i_register=new Intent(MainActivity.this,Register.class);
            startActivity(i_register);

      }

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

      }

public void instruct(View v) {
      Intent i_help=new Intent(MainActivity.this,Instructions.class);
     startActivity(i_help);
}

public void verify(View v) {
      Intent i_verify=new Intent(MainActivity.this,Verify.class);
     startActivity(i_verify);
}
}
```

Verify

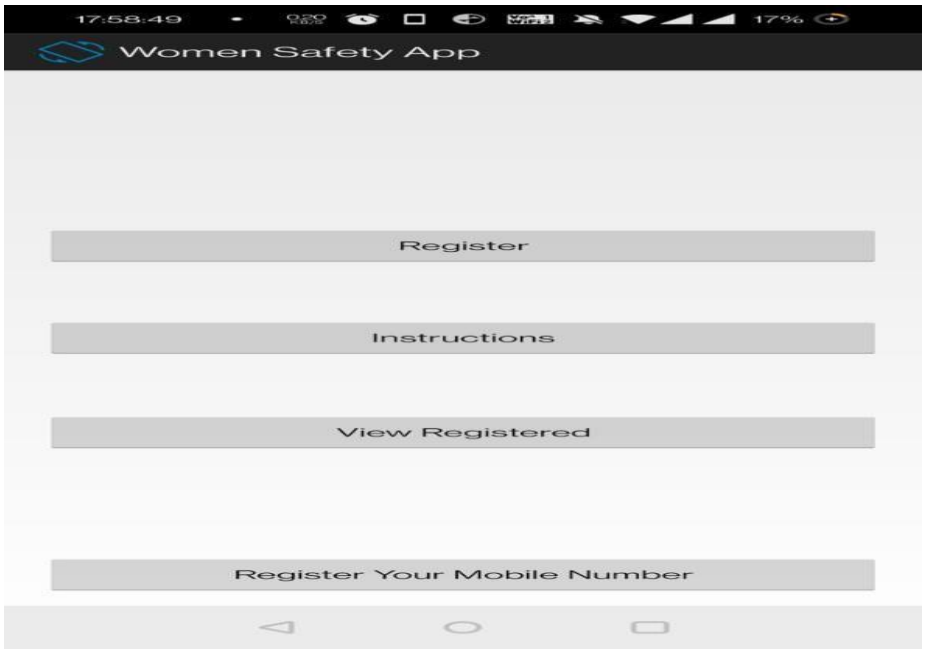
```
package com.prabhu.womensafetyapp;

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+"');");
		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);
		}
```

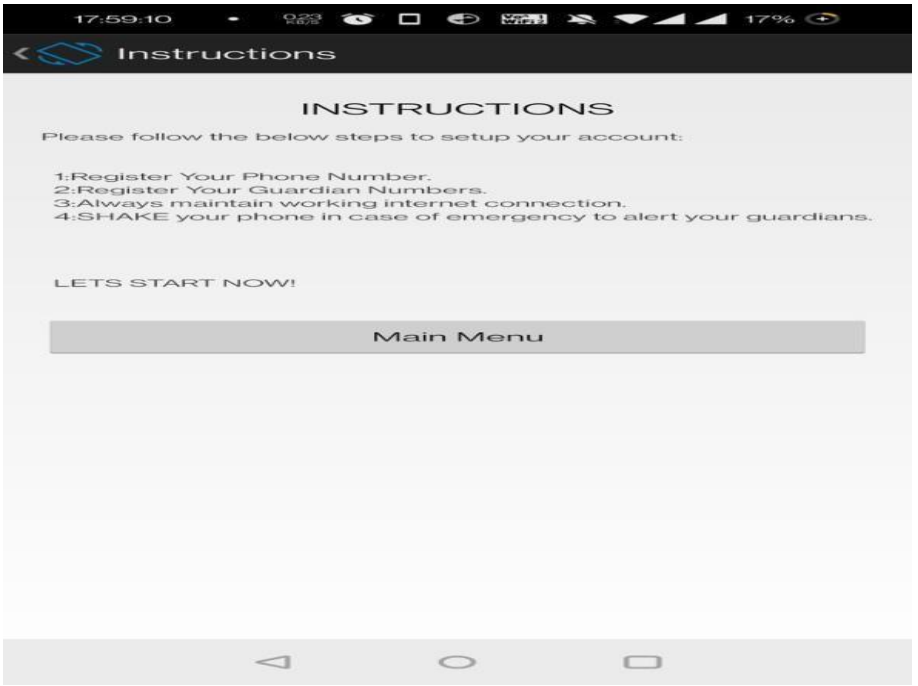
```
	}

	@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
			//
			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);

		}

}
```

SNAPSHOTS

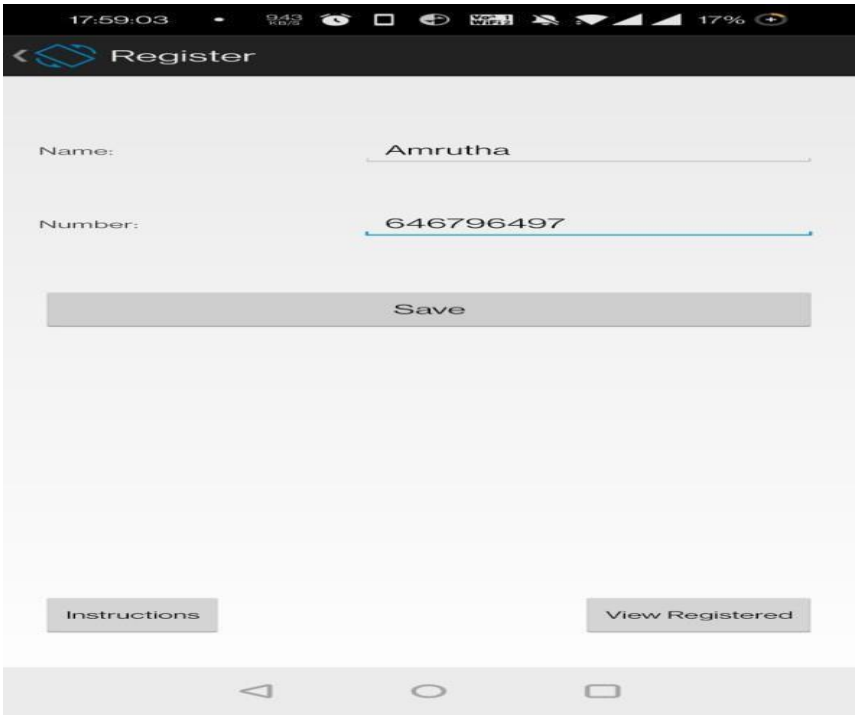
1) Depiction of the front end of the application



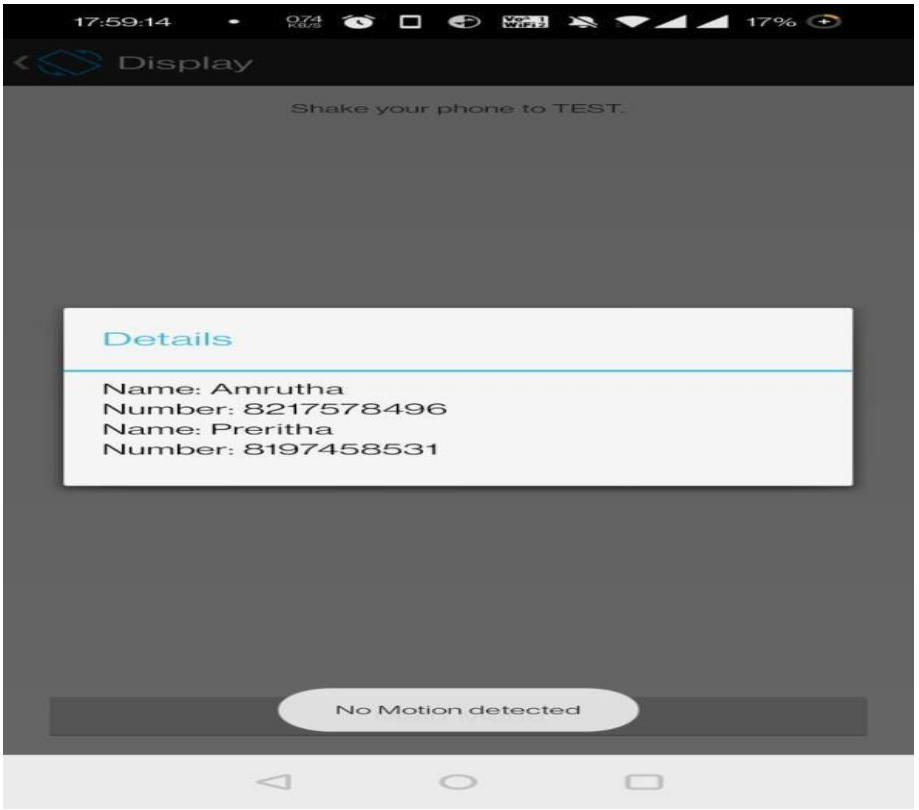
2) Depiction of the general instructions that needs to be followed by the user



3) Registration window of the application

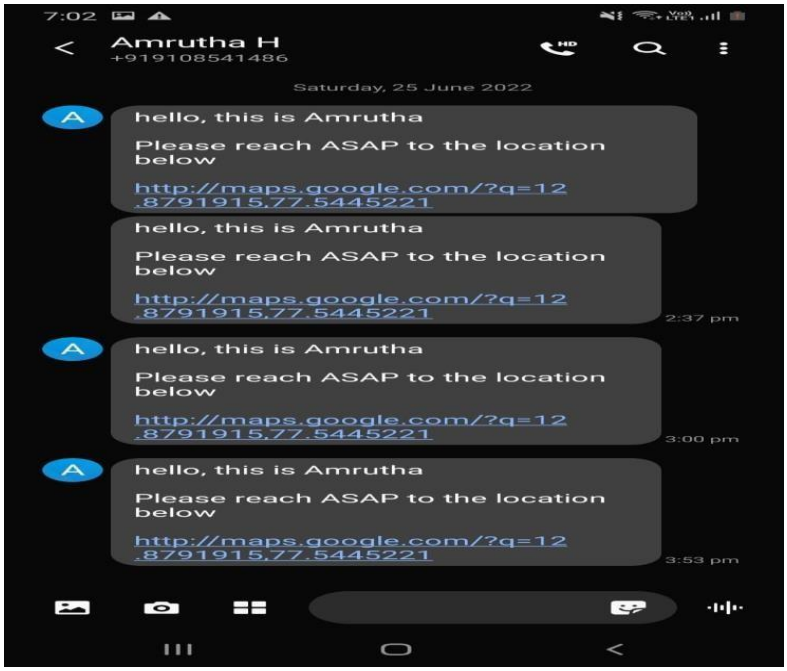


4) The verified database of the user registration

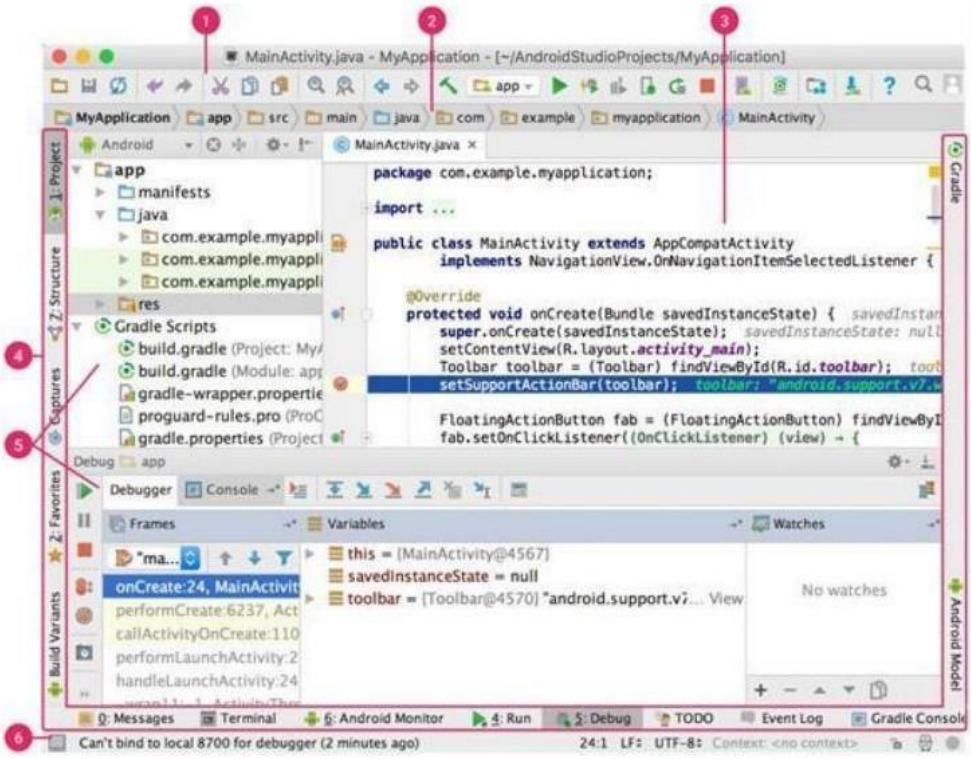


5) Screenshot of the Victim Zonal region

6) Emergency SMS Alert to the respective contacts along with GPS Coordinates



7) The Android Studio main window



Chapter 4
IMPLEMENTATION

4.1 Description

- **MainActivity.java**

It used to display the home page which contains the appropriate fragments to display the Timetable information.

- **private void gettimetable(String timetable)**

This function is used to find the Timetable information.

- **private void getsyllabus()**

This function is used to find the syllabus and subject information.

- **private void letsdoSomeNetworking(RequestParams params)**

This function is used to get the JSON data of TimeTable with the help of API.

- **private void updateUI(Timetabe)**

This function is used to display the Timetable information.

The application‟s key benefit is it is integrated with every possible feature to help someone in need. Offline mode will help by sending message and call and recording the surroundings even when the data connection is not available. Safe zone will help to find nearest police station. So, by combining all the helpful features, this application will overcome the lacking of other related existing systems.

COMPARISON OF THE PROPOSED SYSTEM WITH EXISTING SYSTEMS

Table 1. Comparison of the proposed system and other existing system

Features	Raksha	I Go safely	Shake to alert	Safety pin	Abhaya	Proposed system
Alert message	Yes	Yes	Yes	Yes	Yes	Yes
Send location	Yes	Yes	No	Yes	Yes	Yes
Call	Yes	No	Yes	No	No	Yes
Safe region	No	No	No	Yes	No	Yes
Audio recording	No	Yes	No	No	No	Yes
Offline mode	Yes	No	No	No	No	Yes
Voice command	No	No	No	No	No	Yes
Has all feature?	No	No	No	No	No	**Yes**

The above table represents that the proposed system is unique and attains the required features for the application.

CONCLUSION

This project proposed a new women's safety system that aims to give a very secure environment. Many unwanted incidents took place in the case of women. Problems can come from everywhere. This paper analyzes the main point of the intelligent security process with technology and system building findings. Analyzing and predicting such incident is not possible hence to minimize it this project designed mobile application will be very helpful.

Not only in harassing related problem, it can be used when someone faces accident or hijacking or public problems Whenever anyone is in any kind of danger, this system will help to decrease the risk and make the world a better and safer place to live. In future will work on designing it more secure for decreasing the crimes at the lowest level. Additionally, planning to introduce two unique features in this application. That is hidden camera.

As this is also a safety issue for women. Also identifying microphones. The method which includes user can move his/her phone around doubtful area, if a strong field is detected, user can be aware about hidden device that is secreted within the particular place. Another feature needs to implement in future is marking the unsafe places. If any user faces felt that place is unsafe, they can tag that place as unsafe zone. This helps another user who are in that place. These features will make it more useful and reliable.

Android as a full, open and free mobile device platform, with its powerful function and good user experience rapidly developed into the most popular mobile operating system.

After studying some papers including the apps mentioned above, this project conclude fact that, though there are numerous applications which provides an equivalent purpose as this project, they lack some features. For example, some sends the situation URL but cannot show safe places; some can do live streaming but cannot do any recording to maintain evidence etc.

REFERENCES

- https://www.geeksforgeeks.org/how-to-build-a-timetable-app-in-android/
- https://www.javatpoint.com/
- https://timetable.org/
- https://www.tutorialspoint.com/
- https://stackoverflow.com/
- Android Programming: The Big Nerd Ranch Guide
- D. S. Prashanth, G. Patel and B. Bharathi, "Research and development of a mobile based women safety application with real-time database and data-stream network," 2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT), 2017.
- "Women safety applications," [Online]. Available: enggjournal.com. [Accessed 30 august 2019].
- Albert Mayan.J, Julian Menezes. R and Breezely George M,"Designing a Customized Testing Tool for Windows Phones Utilizing Background Agents", International Conference on Soft Computing Systems, (ICSCS) 2015.
- Prof.Yadhu naik "Sthree Raksha -an Android app", International Journal of Recent Trends in Engineering & Research (IJRTER), vol. 02, issue 10, Oct.2016, pp. 302-307.
- Albert Mayan.J, Julian Menezes. R and Breezely George M,"Designing a Customized Testing Tool for Windows Phones Utilizing Background Agents", International Conference on Soft Computing Systems, (ICSCS) 2015.
- IEEE INDICON 2015 Journal Publication.
- Nirbhaya: Be Fearless: http://www.nirbhaya.mobi. Accessed 2015-06-01. International Labour Organization, 2012, "ILO Global Estimate of Forced Labour: Results and Methodology," p. 14, Geneva.
- "I go safely app," [Online]. Available: http://www.igosafely.com/. [Accessed 25 august 2019