

# FINAL REPORT

**Project Title**

Accident Alert

## Submitted by

Zain Shoukat

# Department of Computer Science



**Govt. Post Graduate College Samanabad Faisalabad 2018**

**CERTIFICATE**

This is to certify that **Zain Shoukat** bearing Registration No. has completed the final project titled as **“Accident Alert”** at the **Department of Computer Science**, **Govt. Postgraduate College Samanabad Faisalabad**, to fulfill the partial requirement for the degree of **BS - CS**.

## Supervisor

**Abdul Manan Signature:**

**Internal Panel**

**Mudasar Maqsood Signature:**

**Qoseen Zehra Signature:**

**Adeel Shahzad Signature:**

**Project Coordinator Head of Department**

**DECLARATION**

The work reported in this project was carried by me under the supervision of **Project Supervisor, Abdul Manan,** at Government Postgraduate College Samanabad Faisalabad.

I hereby declare that this project and the contents of project are the product of my own research and no part has been copied from any other written or published source (accept the references, standard mathematical or genetics models / equation / formulas / protocol etc.).

I further declare that this work has not been submitted for award of any other degree / diploma.

The university may take action if the provided information is found inaccurate at any stage.

**Name : Zain Shoukat Registration No. :**

**ACKNOWLEDGEMENT**

Today i am quite happy on accomplishment of my project. First of all i want to thanks to ALLAH whose help me and guidance enabled me to accomplish my task. Without which i would not have been able to complete not only this project but anything of my life. I wish to express my deepest appreciation to my project advisor. Sir Abdul Manan for the supervision, dedication and commitment with this project. I want to pay tremendous tribute to him for his guidance and help.

And my special thanks to my entire teachers, who have taught me for the past four year. I am grateful to my colleagues and all friends for their moral support and encouragement.

Zain Shoukat

**ABSTRACT**

The high demand of automobiles has also increased the traffic hazards and the road accidents. Life of the people is under high risk. This is because of the lack of best emergency facilities available in our country. An automatic alarm application for vehicle accidents is introduced in this proposal. This design is a system which can detect accidents in significantly less time and sends the basic information to first aid Centre or your relatives within a few seconds covering geographical coordinates, the time and angle in which a vehicle accident had occurred. This alert message is sent to the rescue team or your emergency contacts in a short time, which will help in saving the valuable lives A termination button is also provided in order to terminate the sending of a message in rare case where there is no casualty, this can save the precious time of the medical rescue team. When the accident occurs the alert message is sent automatically to the rescue team and to the emergency contacts. The message is sent through the GSM module and the location of the accident is detected with the help of the GPS module. The accident can be detected precisely with the help of both Micro electro mechanical system (MEMS) sensor and vibration sensor. The Angle of the rolls over of the car can also be known by the message through the MEMS sensor or accelerometer.. This application provides the optimum solution to poor emergency facilities.

**Table of Contents**

**1. Introduction**

* 1. Introduction of project 1
  2. Scope 2
  3. Business Goals 7

1.5 References 7

1. Overall Description 8
   1. [Product Features 9](file:///G:\project\zaain.docx#_TOC_250010)
   2. [User Classes and Characteristics 9](file:///G:\project\zaain.docx#_TOC_250009)
   3. [Operating Environment 9](file:///G:\project\zaain.docx#_TOC_250008)
   4. [Design and Implementation Constraints 9](file:///G:\project\zaain.docx#_TOC_250007)
   5. [Assumptions and Dependencies 10](file:///G:\project\zaain.docx#_TOC_250006)
2. Functional Requirements 11
   1. Use-Case 1 11
   2. Use-Case 2 (and so on) 13
3. Nonfunctional Requirements 14
   1. [Performance Requirements 14](file:///G:\project\zaain.docx#_TOC_250005)
   2. [Safety Requirements 14](file:///G:\project\zaain.docx#_TOC_250004)
   3. [Security Requirements 14](file:///G:\project\zaain.docx#_TOC_250003)
   4. [Software Quality Attributes 14](file:///G:\project\zaain.docx#_TOC_250002)
4. Other Requirements 22

[Appendix A: Glossary 23](file:///G:\project\zaain.docx#_TOC_250001)

[Appendix B: Analysis Models 16](file:///G:\project\zaain.docx#_TOC_250000)

Appendix C: Design Models 22

Appendix D: Screenshots 22

Appendix E: Test Cases 32

**6. Coding 38**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **List of Tables** | |  |
| Table 1.1 | | Cost Estimation ……………………………………………………………… | 5 |
| Table 1.2 | | Schedule Feasibility (a) ……………………………………………………… | 5 |
| Table 1.3 | | Schedule Feasibility (b) ……………………………………………………… | 6 |
| Table 3.1 | | UC\_01 ………………………………………………………………………….. | 13 |
| Table 3.2 | | UC\_02 ………………………………………………………………………….. | 14 |
| Table 3.3 | | UC\_03 ………………………………………………………………………….. | 15 |
| Table 5.1 | | Saving Contacts ………………………………………………………………... | 36 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **List of Figures** | |  |
| Figure 5.1 | | Sequence Diagram……………………………………………………………… | 20 |
| Figure 5.2 | | Use Case………………………………………………………………………… | 21 |
| Figure 5.3 | | Contacts ………………………………………………………………………… | 22 |
| Figure 5.4 | | Context ………………………………………………………………………….. | 23 |
| Figure 5.5 | | DFD (a)………………………………………………………………………….. | 24 |
| Figure 5.6 | | DFD (b) …………………………………………………………………………. | 25 |
| Figure 5.7 | | Activity Splash …………………………………………………………… | 26 |
| Figure 5.8 | | Activity Main …………………………………………………………………… | 27 |
| Figure 5.9 | | Activity Save Contacts …………………………………………………………. | 28 |
| Figure 5.10 | | Activity Tutorials (part 1) ……………………………………………………….. | 29 |
| Figure 5.11 | | Activity Tutorials (part 2) ……………………………………………………….. | 30 |
| Figure 5.12 | | Activity Tutorials (part 3) ……………………………………………………….. | 31 |
| Figure 5.13 | | Activity Modes ………………………………………………………………….. | 32 |
| Figure 5.14 | | Activity Detection Levels ……………………………………………………….. | 33 |
| Figure 5.15 | | Activity Sensors Values ………………………………………………………… | 34 |
| Figure 5.16 | | Accident Detected ………………………………………………………………. | 35 |
| Figure 5.17 | | Activity TimeUp ………………………………………………………………… | 36 |

CHAPTER - 1

***Introduction***

**BRIEF INTRODUCTION OF THE “Accident Alert”**

In Our Life, many a people die from traffic accidents. A perfect and effective way for reducing traffic accidents hazards is to minimize the time between when an accident occurs and when rescue reacts. We propose to develop an application which will detect accident using the sensors present in android cellular phone. Our proposed system consists of two phases; the detection phase which is used to detect car accident with location and the notification phase which will notify the respondents or rescue team. We will also add an “HELP ME” soft button, will be used in the emergency situation else than accident (like stuck in bad circumstances or having an attack of medical disease etc.) in case of using this button a predefined massage will be sent to appropriate responder.

**1.2 Problem:**

Problem is that when an accident occurred, the appropriate responder does not know that what happened to our loved one. So, no one come to help the victim. Problem is Acknowledgment and current location of an accident and secondly problem is that we do not have a way to instantly notify our responders of an emergency happening other than accidents. So, this app will resolve these issues.

**1.3 SCOPE OF PROJECT:**

The rapid advancements in the area of mobile applications development and the popularity of android have encouraged the provision of utility applications. So, the major objectives of my application to detect the accident and notify about accident or other emergencies, the algorithm will detect an accident and send a message to responders encapsulated location in it.

Actually, i have two major phases of Feature one of automatic mode:

* Accident detection
* Alerting the responders

And we have also a HELP ME button:

* Button pressed by user
* Alert message sent to responders instantly So this is the total scope of our project.

**Introduction of Accelerometer:**

The motion sensors in accelerometers can even be used to detect earthquakes and may be used in medical devices such as bionic limbs and other artificial body parts. Several devices, part of the quantified self-movement, use accelerometers.

An accelerometer is an electromechanical device used to measure acceleration forces. Such forces may be static, like the continuous force of gravity or, as is the case with many mobile devices, dynamic to sense movement or vibrations.

**Introduction of GPS:**

The Global Positioning System (GPS) is a satellite-based navigation system made up of a network of 24 satellites placed into orbit by the U.S. Department of Defense. GPS was originally intended for military applications, but in the 1980s, the government made the system available for civilian use. GPS works in any weather conditions, anywhere in the world, 24 hours a day. There are no subscription fees or setup charges to use GPS.

**1.3.1 User Friendly Environment:**

The success of our application is dependent on the end user acceptance. Though making the interface that is exposed to the end user and crucial part of the application is to keep in mind that all the user interface will be design in such a way that the end users will easy to interact with them. In our project all the process should be able to interact with each other. It is the success of project that the entire module in the proposed system still interacts with each other ensuring that the manual routines should not be followed.

**1.3.2 Graphical User Interface:**

The interface of our application is attractive and helps the end user to use software easily and reliably.

So, the end user does not find it difficulty or bother to interact with the software. Client side will he developed on an Android handset. So, it is just the tapping process to input and Minn. Any lay man can

easily get it. The detailed presentation of User interface is defined below in User Interface design.

**1.4 Objectives**

* Detect Accident Automatically
* Fetch the location of incident
* Encapsulate the informal message and current location coordinates of GPS in SMS
* And that SMS will be sent to proper responders/family
* Emergency alert Manually by pressing a button instantly
* User will press “HELP-ME” button and it will Encapsulate the informal message and current location coordinates of GPS in SMS
* And will be sent to family

**1.5 Feasibility Report**

**1.5.1 Technical Feasibility: -**

My project is completely feasible within the limits of current technology. Now a day’s technology is used in almost every android handset. Both software and hardware are required for this application. And it integrates with the DATABASE. So, there won't be any technical issue regarding to feasibility. Following are the technical elements of our project.

* Android
* Sdk
* NetBeans
* Java Sdk
* Mysql Lite

Database According to the user perspective. If the user has an android device, so whenever he suffers from and accident or an emergency, that will be notified to his family and friends instantly. I designed very user-friendly environment so even the computer illiterate user can also interact with our application.

**1.5.2 Operational Feasibility:**

Operational feasibility of our project depends on following elements.

1: Android Smartphone

2: GPS

3: Accelerometer

4: GSM Connection for Message

**Economic Feasibility**

My project is very feasible according to economic issues.

### 

Cost and benefit analysis:

One off cost:

**Our project contains the following elements as the onetime cost:**

|  |  |
| --- | --- |
| **Name** | Specification Price in Market Quantity Reason of use |
| **Android Device**  **Android Device** | Low definition 10,000 RS 1 Worst testing  High definition 30,000 RS 1 Happy testing |

**Table 1.1 Cost Estimation**

**Schedule Feasibility:**

|  |  |
| --- | --- |
| **Coding** | **1 Month** |
| Data Management Classes  Data Base Classes Functionalities  Classes Internet connectivity  Integration of UI designs |  |

**Table 1.2 Schedule Feasibility (a)**

|  |  |
| --- | --- |
| **Testing** | **2 Month** |
| Unit Testing  Integration Testing Black Box testing  White Box Testing  Compatibility Testing |  |

**Table 1.3 Schedule Feasibility (b)**

Over All time required for our project = 3 Months Approximately

**Informational Feasibility:**

The project while based on the modern and advance terminologies it will provide feasibility,

Reliability and will be consistent.

**Motivational Feasibility:**

Our product will be of self-explanatory nature and the end user would not require intensive

training session or learning material for using our software. The end user would have no

confusion about whatever he/she wants to do. The graphical interface (GUI) of our product will

be so attractive and user-friendly that it will motivate the end user about the use of our

application, so that user will have a feeling of easiness and contentedness while interacting with

our application.

**Legal and Ethical Feasibility:**

This project supports legally and ethically to all the application user. we provide our disclaimer

and our privacy policy to user. I his warranty gives specific legal rights and user may also

have other legal rights that vary from state to state.

In our project there is no ethical issue as it has no copied material and any copyright violation

content. And the business has to be registered first by giving there all details. And in the case of

any issue that detail will be given to the concerned peoples and I will place an agreement before creation of an account to get our state legally strong will place an agreement before creation of an account to get our state legally strong.

And in the case of any issue that detail will be given to the concerned peoples and we

will place an agreement before creation of an account to get our state legally strong.

**Specification Feasibility:**

While looking at this consideration we are optimistic that all the requirements and specification

regarding this software are definite and crystal clear to our development team. Besides that, we

also have well define scope and boundaries of our product.

**Business Goals:**

This app is valuable according to business goals too. We will upload it on Google play store for

customers and by Google ads developer can earn money.

# References:

#### <http://developer.android.com/guide/topics/sensors/index.html>

1. [**http://developer.android.com/guide/topics/sensors/sensors\_overvie w.html**](http://developer.android.com/guide/topics/sensors/sensors_overview.html)
2. [**http://developer.android.com/reference/android/hardware/Sensor. html**](http://developer.android.com/reference/android/hardware/Sensor.html)
3. Professional Android Sensor Programming

Book by Adam Stroud and Greg Milette

1. [**https://stackoverflow.com/**](https://stackoverflow.com/)

CHAPTER - 2

*Overall Description*

**Chapter 2: - Overall Description**

**2.1. Existing system:**

Actually, there is no system like this in our country right now, so there would be no proper

Comparison of existing system and present one. Although there are many applications

Developed by foreign developers.

**Reasons of failure of existing systems:**

* Not better GUI
* Not supported to all android smartphones
* There is no app that having dual detections of G-Value and Acceleration
* Not having support of multi levels of detection
* User is not guided properly
* More false alarm possibilities
* No false alarm handler included
* Applications do not run when user locks phone

**Reasons of choosing our application:**

* A good and simple GUI
* Less false alarm possibility
* Timer included if this is false alarm user can disable it by pressing “I’m safe” button within countdown time
* Proper guidance for user
* Dual detections by G-Value and Acceleration
* Can support all android phones (exceptional cases can occur)
* Multi-type sensor support
* Detection can run even phone is locked
* Timer activity can show upon lock screen
* Phone automatically lock is including in application

**Product Features:**

Our app is designed to overcome the danger of deaths in accident only because of no rescue or

help and second purpose is to give a fast way to alert our family of any emergency situation we

got into. So, app automatically detects accidents and send a message to the responders for that

Person.

**Main features are:**

* Provides two modes manual and auto-detection
* Manually user can send an alert message with his current location only by pressing a button
* Automatically detects a collision/crash of a car
* Send message to three phone numbers with user’s current location

**User Classes and Characteristics:**

The designed application is made for only android users. And it can only run if a android

smartphone has a gps and an accelerometer sensor (which are almost present on 99%

devices today). User must have little bit experience to run android smartphone. GPS must be

enabled before launching the app. Almost all exceptions are handled e.g. if GPS is off then

app will redirect user to location settings so user can turn on GPS. Mobile must have a standard

accelerometer which can calculate G-value and current acceleration.

**Operating Environment:**

The designed application can run on all the android devices. It supports from android Kit Kat to

onward. Application can run on all custom GUIs developed by mobile branding companies like

Samsung, Huawei etc. It also run smoothly on stock android devices like Google nexus and

others.

**Design and Implementation Constraints:**

Interface of the application is designed in Android studio using XML (Extensible Markup

Language).

Android studio is an IDE (Integrated development environment) suggested by Google itself for

android development. And back-end developing language is used for application is JAVA. All

coding is done in android studio. Designing elements e.g. buttons are created and adobe

Photoshop and used in app using drawable in android studio.

**Assumptions and Dependencies**

**Dependencies:**

1. Proper GPS
2. This application is totally dependent on presence of accelerometer.
3. Kit Kat and newer than Kit Kat android OS needed.
4. Only run in android devices licensed by Google.
5. Mobile should be properly functioning (Not damaged or stopped working etc.)

**Assumptions:**

All type of android custom operating systems can run it e.g. Samsung’s Touch Wiz.

GPS is present on almost all devices.

Users are more adaptive now days.

**Permissions:**

Every android app uses permissions from user to perform its tasks and operations. If user does not accept the permissions then application could not be installed or work properly. My app uses some of permissions:

* Current location (By GPS)
* Send SMS
* Mobile vibration
* Audio settings
* Access system location settings
* Wake Lock (To keep mobile awake)

# 

CHAPTER - 3

*Functional Requirements*

**Chapter 3: -Functional Requirements**

**Adding emergency Contacts**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | Saving Contacts in database | |
| **Purpose** | | To send an alert message | |
| **Priority** | | High | |
| **Pre-conditions** | | Run application | |
| **Post-conditions** | | To choose auto detection or manually alert | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Add or update contacts. | | System will show contact screen |
| **2** | User will add at least one contact | | System show the added contacts |
| **3** | User press button to save contacts | | System will save the contacts and return main activity |

Table 3.1 UC-1

**Manually send alert message:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | Manual alert | |
| **Purpose** | | Manually send alert message to saved contacts | |
| **Priority** | | Medium | |
| **Pre-conditions** | | Contact should be saved and GPS should be turn on. | |
| **Post-conditions** | | Send alert message by pressing Help Me button. | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| 1 | User go to alert activity | | System will show manual and auto detection mode. |
| 2 | User presses the Help Me Button | | System will send alert message to saved contacts with location. |

**Table 3.2 UC-2**

**3.3 Auto detection Mode:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | Auto detection mode | |
| **Purpose** | | Automatically detect accident and send an alert sms | |
| **Priority** | | High | |
| **Pre-conditions** | | Turn on auto detection mode | |
| **Post-conditions** | | On detection of accident , alert sms will send. | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| 1 | User go to alert activity | | System will show manual and auto detection mode. |
| 2 | User turn on auto detection mode | | System will show detection level choice |
| 3 | User will choose any level of detection from 1 to 5 | | System will show the detection screen and start observing sensor data, Location set by GPS |
| 4 | User put mobile on holder placed in car | | System detects accident and send message to contacts with current location |

**Table 3.3 UC-3**

CHAPTER - 4

*Nonfunctional Requirements*

**Chapter 4: - Nonfunctional Requirements**

4.**1 Performance Requirements**

* The designed application will respond every user that will access the application, the access time depends on the speed of GPS of the user and it depends on availability of accelerometer sensor.
* Simple interface leads to error free access to the application.
* Excellent performance of app on all supported android platform.

**4.2: Safety Requirements**

* The designed application will not use contacts for any other use.
* App will not do anything without user’s permission.

**4.3: Security Requirements**

The designed application will not use credentials of the user for any other purpose. Application will not provide access unauthorized system resources.

**4.4: Software Quality Attributes**

* The designed application must be efficient enough to use the system resource efficiently.
* This application will not disturb any other process on platform when it is running. This application will access the data with the permission of User it will not access any other information of the user.
* The application is easy to use and reliable.
* The application has the capacity of reusability

CHAPTER - 5

*Other Requirements*

**Appendix A: Glossary**

UC Use case

SRS Software Requirements specification

TC Test Case

**Appendix B: Analysis Models**

**Sequence Diagram:**

User

**User enter the application**

**System asks user to turn on GPS if it is not User turned on GPS**

**System shows main activity User click on add contacts**

**System displays add contacts activity**

**User add contacts upto 3 and click on save button**

**System save contacts in database and shows main activity User click on Go to App button**

**System shows the alert activity If User click on Help Me Button**

**System send sms to contacts with location If User select the auto-detection**

**System will ask for detection level User select any level from 1 to 5 System observe sensor data**

**Syestem detects accident, send sms with location**

System

**Figure 5.1 Sequence Diagram**

# Use Cases:

Add contacts

Select auto mode Or

Send alert message

**User**

System observe sensor data

Send message with location to contacts

Fig

**Contacts:**

Go to Add contacts

**User**

Write and Save

contacts

**Figure 5.3 Contacts**

**Context:**

User

Accelerometer

GPS

Contacts saved in

DB

iSmash Pocket Rescue

d

**Figure 5.4 Context**

**DFD:**

Open Main App

GPS on,

Location Fetched

Auto detection

mode

Manual Mode

Sensor Data

Observed

User Press

Button

Accident

Detected

Message sent

After 20 seconds, Message will Sent to contacts

Add Contacts

User

**Figure 5.5 DFD (a)**

**Saving or Removing Contacts:**

Delete Contacts

Add Contacts

Contact Saved

User

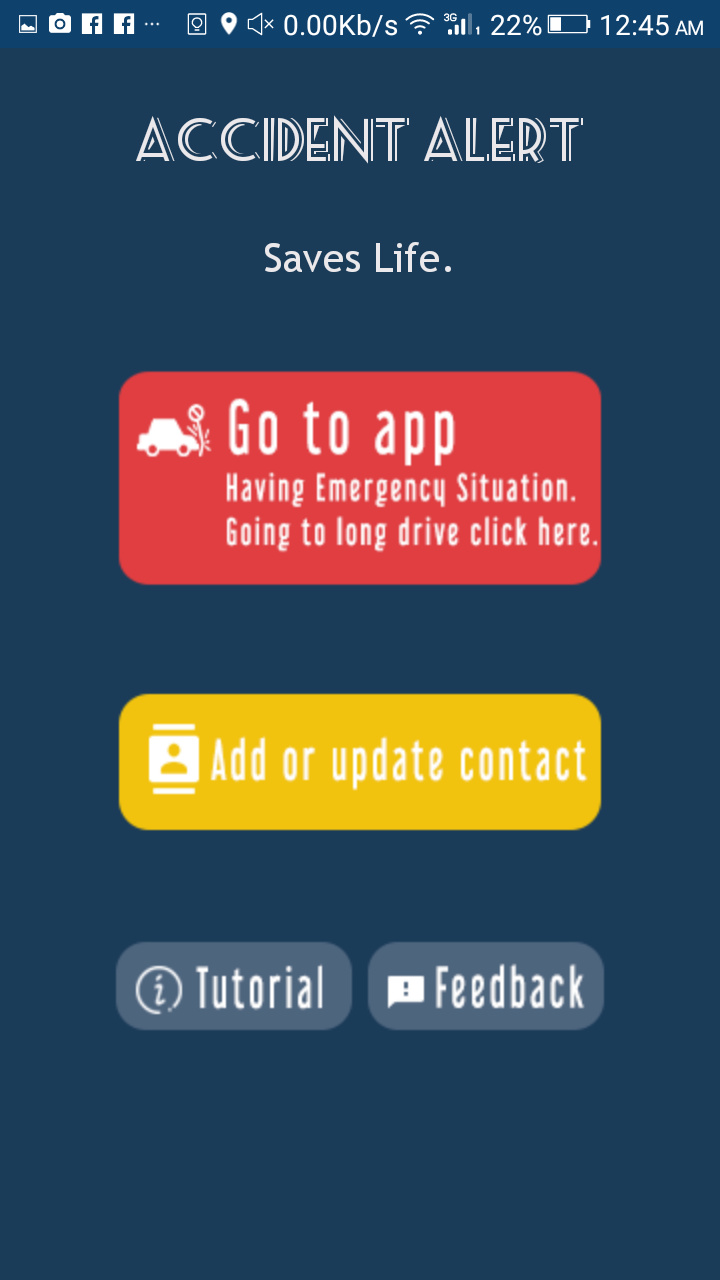
**Appendix C: Design Models**

Screen shots:

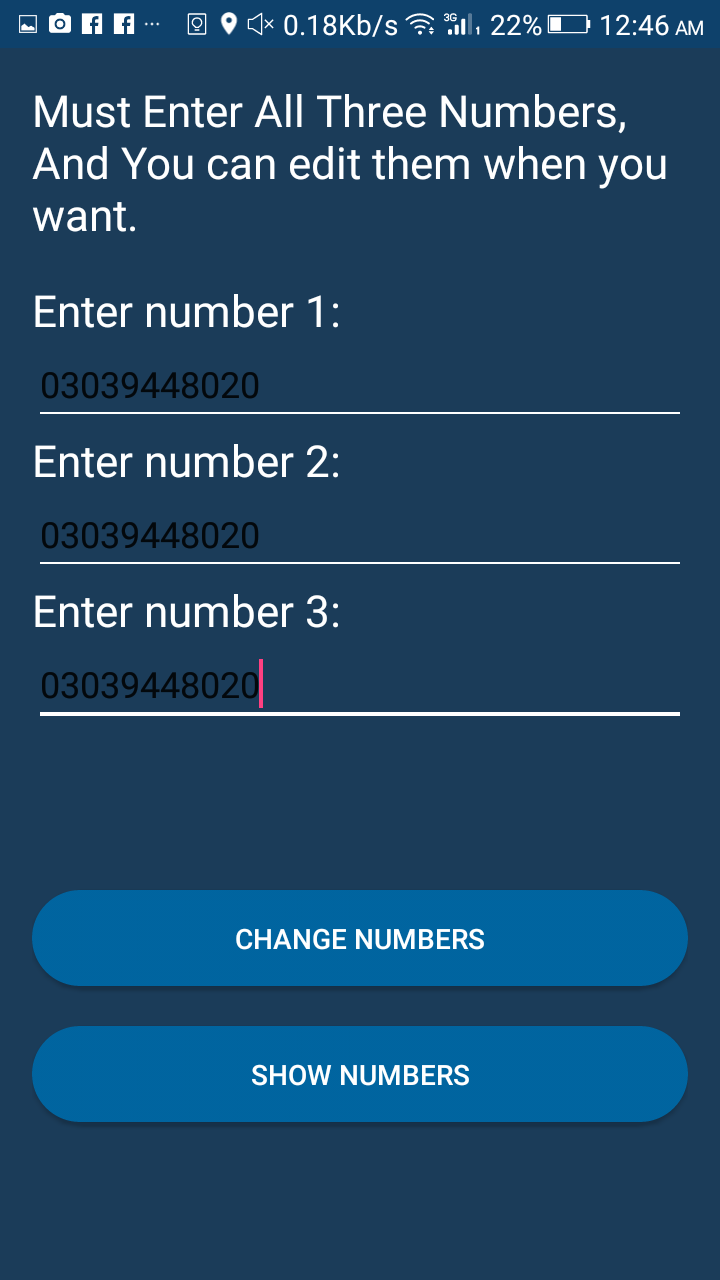
**Splash Screen.**

**Figure 5.7 Splash Screen** 

**Main Screen.**

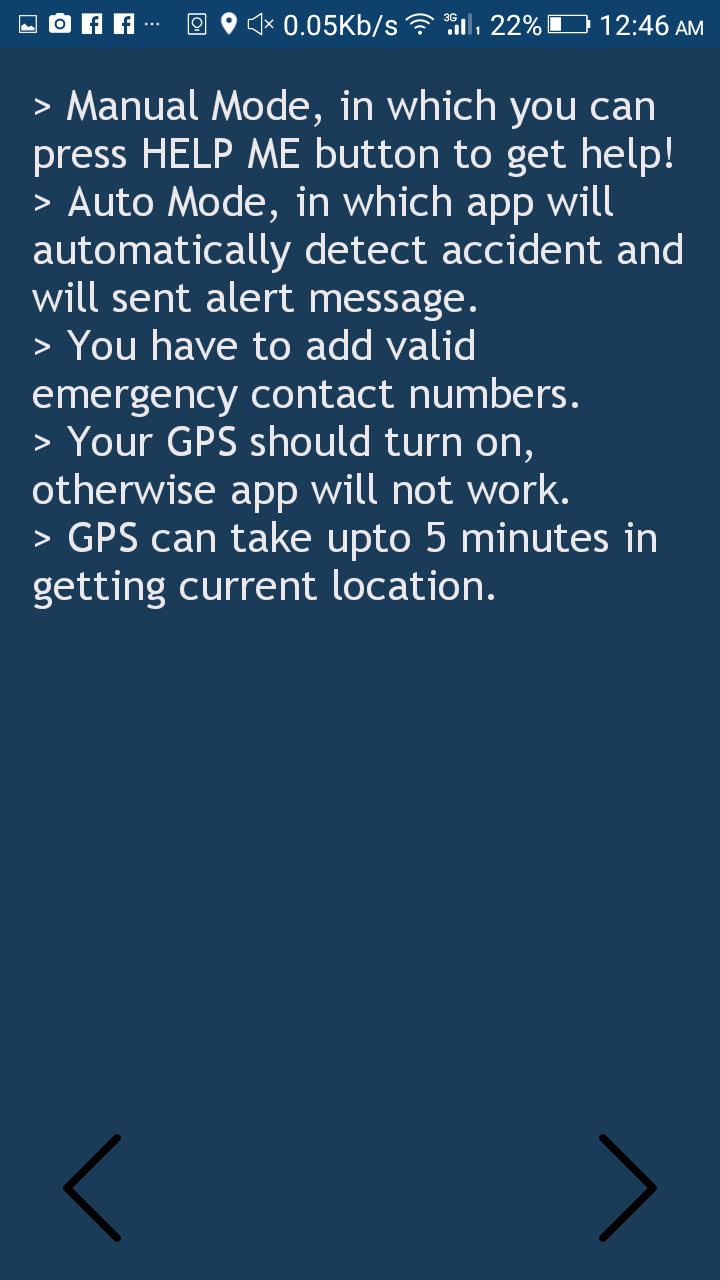
**Figure 5.8 Main Screen** 

**Contact Screen.**

**Figure 5.9 Saving Contacts** 

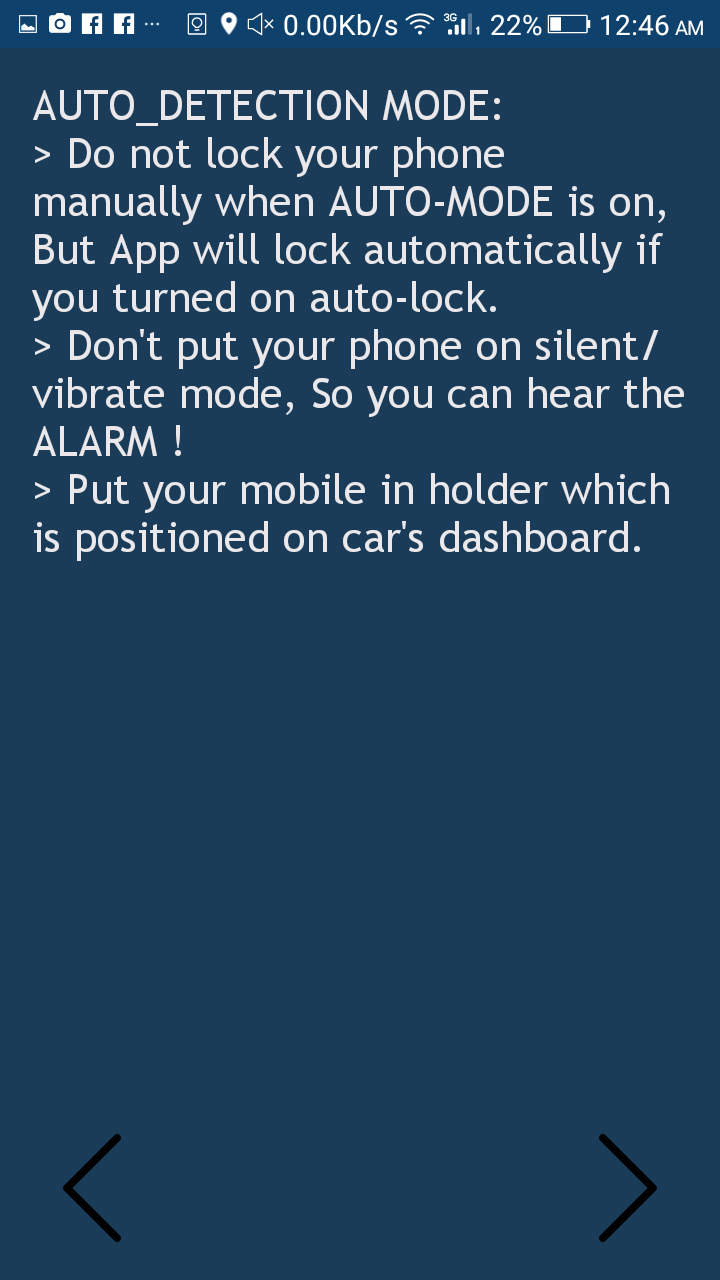
**Tutorials (For Guidelines)**

**Part 1**

**Figure 5.10 Tutorials**  

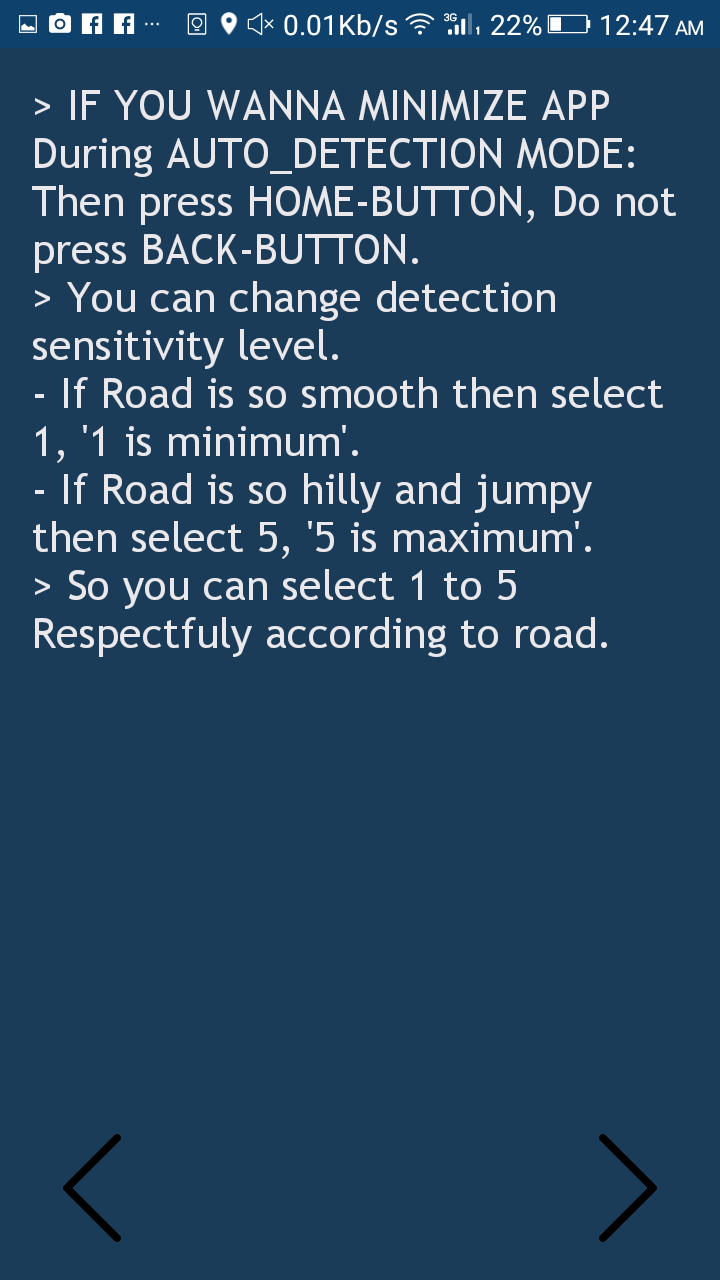
**Tutorials (For Guidelines)**

**Part 2**

**Figure 5.11 Tutorials** 

**Tutorials (For Guidelines)**

**Part 3**

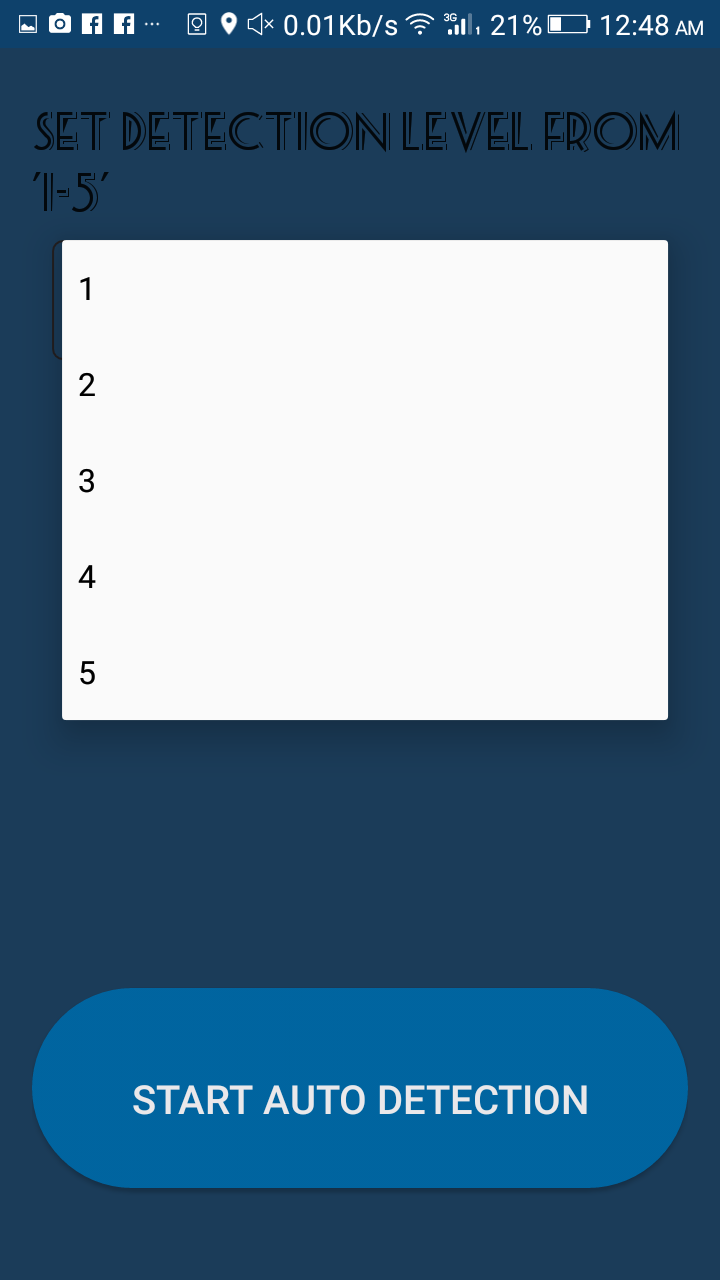
**Figure 5.12 Tutorials** 

**Selection Screen between send emergency alert or Auto accident detection**

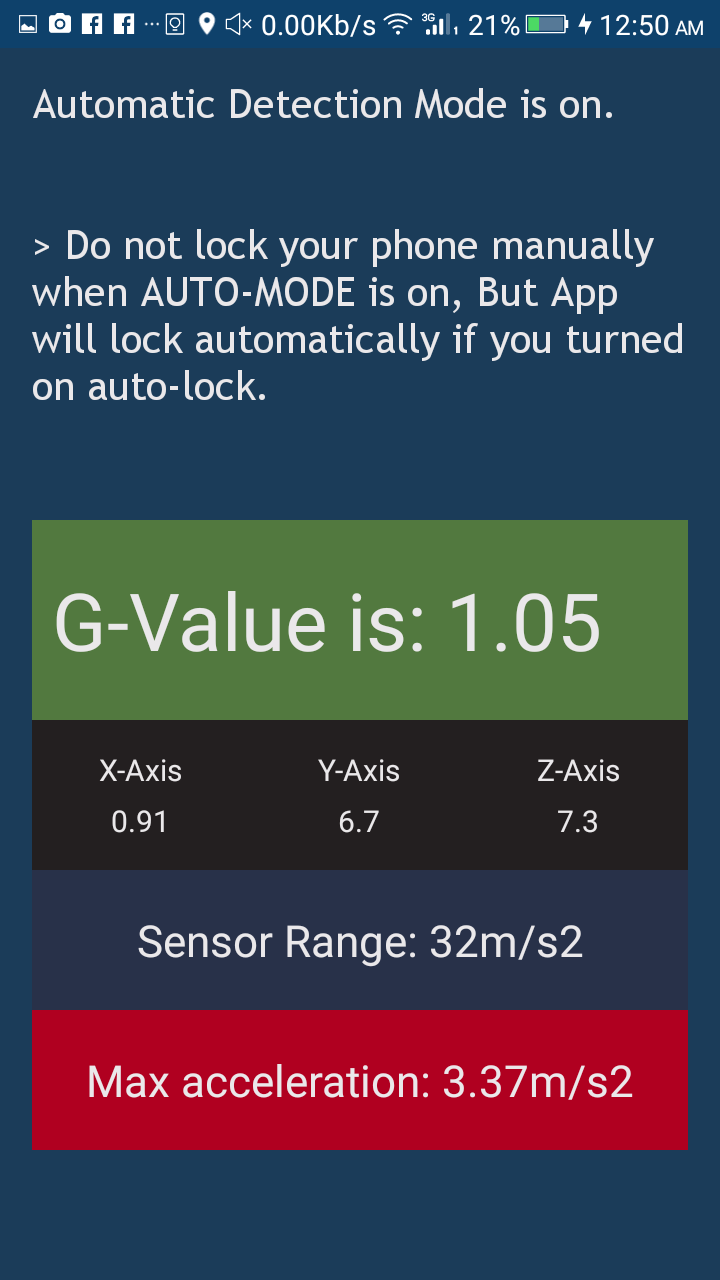
**mode.**

**Figure 5.13 Modes**  

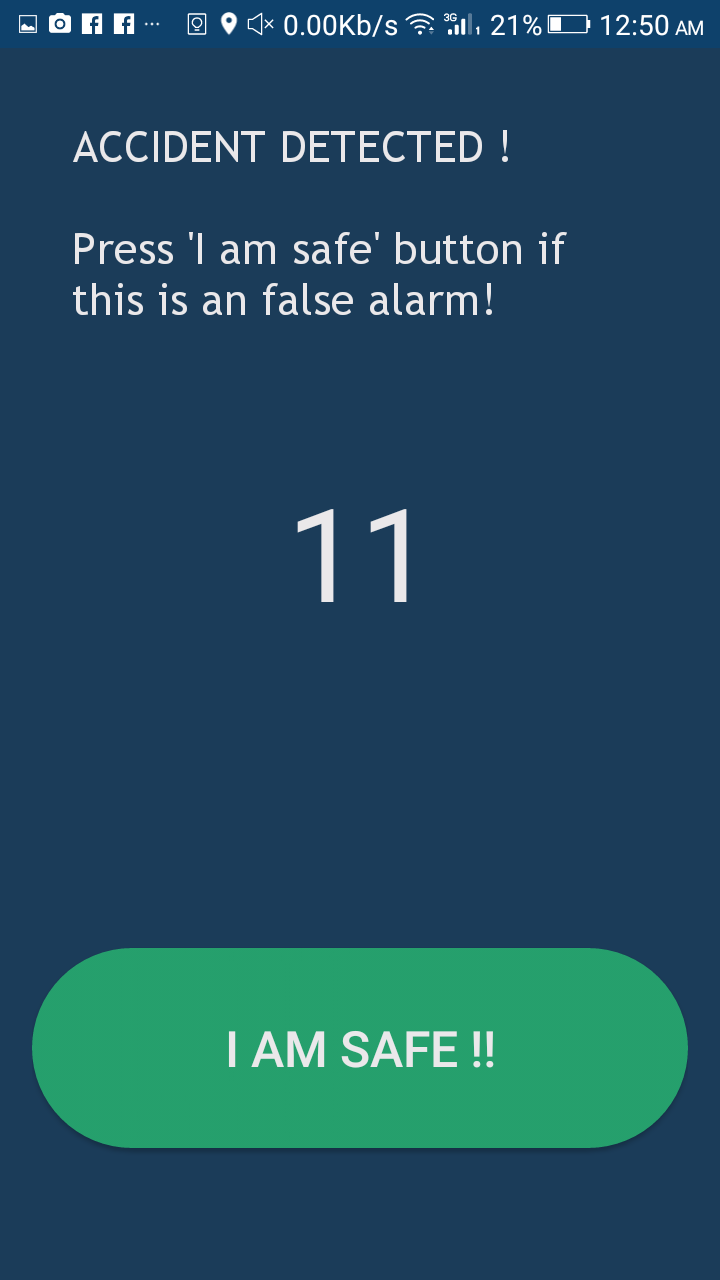
**Detection level screen.**

**Figure 5.14 Detection Levels** 

**Auto detection mode enabled.**

**Figure 5.15 Sensor Values **

**Accident detected.**

**Figure 5.16 Accident Detected **

**Appendix D: Test Cases**

**Saving Contacts:**

|  |  |
| --- | --- |
| **Identifier** | Contacts |
| **Priority** | High |
| **Related requirements(s)** | System basic |
| **Short description** | User will enter contact numbers |
| **Pre-condition(s)** | System will not be used without contacts |
| **Input data** | Contacts |
| **Detailed steps** | Firstly, enter the contact number then click on save button. |
| **Expected result(s)** | System will save entered 1-3 contacts in database. |
| **Post-condition(s)** | It is authenticated that users are willingly adding contacts. |

**Table 5.1 Saving Contacts**

**Chapter 6: - CODING**

**MainActivity.java**

package com.example.zain.accidentAlert;

import android.Manifest; import android.app.Activity; import android.content.Context;

import android.content.DialogInterface;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.graphics.Typeface; import android.location.Location; import android.location.LocationListener;

import android.location.LocationManager;

import android.os.Bundle;

import android.provider.Settings;

import android.support.v4.app.ActivityCompat; import android.support.v7.app.AlertDialog; import android.view.View;

import android.view.animation.Animation; import android.view.animation.AnimationUtils; import android.widget.Button;

import android.widget.TextView;

import android.widget.Toast;

public class MainActivity extends Activity implements LocationListener { DatabaseHelper myDb;

Button setting, manual, auto, feed, tut; Animation bounce;

private LocationManager locationManager; private double lat;

private double lot;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

bounce = AnimationUtils.loadAnimation(getApplicationContext(), R.anim.bounce);

myDb = new DatabaseHelper(this); AddData();

locationManager = (LocationManager) getSystemService(Context.LOCATION\_SERVICE);

if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED && ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {

// TODO: Consider calling

// ActivityCompat#requestPermissions

// here to request the missing permissions, and then overriding

// public void onRequestPermissionsResult(int requestCode, String[] permissions,

// int[] grantResults)

// to handle the case where the user grants the permission. See the documentation

// for ActivityCompat#requestPermissions for more details.

return;

}

locationManager.requestLocationUpdates(LocationManager.GPS\_PROVIDER, 60000, 100, this);

// Font path

String fontPath = "fonts/m.ttf";

String fontPath2 = "fonts/Trebuchet MS.ttf";

// text view label

TextView txt1 = (TextView) findViewById(R.id.textView9); TextView txt2 = (TextView) findViewById(R.id.textView10);

// Loading Font Face

Typeface tf = Typeface.createFromAsset(getAssets(), fontPath); Typeface tf2 = Typeface.createFromAsset(getAssets(), fontPath2);

// Applying font

txt1.setTypeface(tf); txt1.startAnimation(bounce); txt2.setTypeface(tf2); txt2.startAnimation(bounce);

setting = (Button) findViewById(R.id.button3); manual = (Button) findViewById(R.id.button5); tut = (Button) findViewById(R.id.button0);

tut.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

Intent i = new Intent(MainActivity.this, tut1.class); startActivity(i);

}

});

setting.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v)

{

Intent i = new Intent(MainActivity.this, Data.class); startActivity(i);

}

});

manual.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

}

});

Intent i = new Intent(MainActivity.this, Crash.class); startActivity(i);

}

private void AddData() {

boolean isInserted = myDb.insertData( "1",

"",

"",

"");

if (isInserted = true) {

Toast.makeText(Menu.this,"Data Inserted",Toast.LENGTH\_LONG).show();

vall1();

} else

Toast.makeText(getApplicationContext(), "Database error !", Toast.LENGTH\_LONG).show();

}

@Override

public void onLocationChanged(Location l) {

this.lat = l.getLatitude();

this.lot = l.getLongitude();

}

manual.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

}

});

Intent i = new Intent(MainActivity.this, Crash.class); startActivity(i);

}

private void AddData() {

boolean isInserted = myDb.insertData( "1",

"",

"",

"");

if (isInserted = true) {

Toast.makeText(Menu.this,"Data Inserted",Toast.LENGTH\_LONG).show();

vall1();

} else

Toast.makeText(getApplicationContext(), "Database error !", Toast.LENGTH\_LONG).show();

}

@Override

public void onLocationChanged(Location l) {

this.lat = l.getLatitude();

this.lot = l.getLongitude();

}

@Override

public void onStatusChanged(String provider, int status, Bundle extras) {

}

@Override

public void onProviderEnabled(String provider) {

}

@Override

public void onProviderDisabled(String provider) {

Intent intent = new Intent(Settings.ACTION\_LOCATION\_SOURCE\_SETTINGS); startActivity(intent);

Toast.makeText(getBaseContext(), "Gps is turned off! Turn ir on nd wait. ", Toast.LENGTH\_SHORT).show();

}

@Override

public void onBackPressed() {

new AlertDialog.Builder(this)

.setMessage("Are you sure you want to exit?")

.setCancelable(false)

.setPositiveButton("Yes", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int id) { onDestroy(); MainActivity.this.finish();

}

})

.setNegativeButton("No", null)

.show();

}

}

**Crash.java**

package com.example.mbk82.pocketresq;

import android.Manifest;

import android.content.Context; import android.content.DialogInterface; import android.content.Intent;

import android.content.pm.PackageManager;

import android.database.Cursor; import android.graphics.Color; import android.graphics.Typeface; import android.location.Location;

import android.location.LocationListener; import android.location.LocationManager; import android.os.Bundle;

import android.os.Vibrator;

import android.provider.Settings;

import android.support.v4.app.ActivityCompat;

import android.support.v7.app.AlertDialog;

import android.support.v7.app.AppCompatActivity;

import android.telephony.SmsManager;

import android.view.View;

import android.view.animation.Animation; import android.view.animation.AnimationUtils; import android.widget.Button;

import android.widget.TextView;

import android.widget.Toast;

public class Crash extends AppCompatActivity implements LocationListener, Animation.AnimationListener {

public String sms; public Vibrator v; Button save;

DatabaseHelper myDb; gpss gs;

Data db1;

String id, cell1, cell2, cell3; TextView textSMS; AlertDialog.Builder builder; Typeface tf2;

TextView locc; Animation animFadein; Animation bounce;

private LocationManager n; private double lat;

private double lot; private String gps;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState); setContentView(R.layout.activity\_crash);

v = (Vibrator) this.getSystemService(Context.VIBRATOR\_SERVICE);

animFadein = AnimationUtils.loadAnimation(getApplicationContext(), R.anim.fade\_in);

bounce = AnimationUtils.loadAnimation(getApplicationContext(), R.anim.bounce);

myDb = new DatabaseHelper(this);

db1 = new Data();

gs = new gpss();

String fontPath2 = "fonts/Trebuchet MS.ttf"; tf2 = Typeface.createFromAsset(getAssets(), fontPath2);

n = (LocationManager) getSystemService(LOCATION\_SERVICE);

if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_FINE\_LOCATION)

!= PackageManager.PERMISSION\_GRANTED)

&&ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_COARSE\_LOCATION) !=

PackageManager.PERMISSION\_GRANTED) {

// TODO: Consider calling

ActivityCompat#requestPermissions

here to request the missing permissions, and then overriding

public void onRequestPermissionsResult(int requestCode, String[] permissions,

int[] grantResults)

return;

}

n.requestLocationUpdates(LocationManager.GPS\_PROVIDER, 60000, 100, this);

vall1();

save = (Button) findViewById(R.id.save);

save1 = (Button) findViewById(R.id.save2);

save1.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

Intent b = new Intent(Crash.this, Data.class); startActivity(b);

finish();

}

});

save = (Button) findViewById(R.id.save);

save1 = (Button) findViewById(R.id.save2);

save1.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

Intent b = new Intent(Crash.this, Data.class); startActivity(b);

finish();

}

});

textSMS = (TextView) findViewById(R.id.textView6);

textSMS.setText("> Press HELP ME button if any emergency situation occurred to you !" +

"\n" +

"> Press AUTO DETECTION button for turning on accident detection service !" + "\n> After seeing notification 'New location fetched', you can use application.\n"); textSMS.setTypeface(tf2);

locc = (TextView) findViewById(R.id.loc); locc.setTypeface(tf2); locc.startAnimation(animFadein);

/\*

textSMS.setText("Help me, im at: [location], Im in critical situation !");

\*/

sms = "Help me please im at: ";

save.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

if (cell1.toString().length() <8 &&cell2.toString().length() <8 &&cell3.toString().length()

<8) {

showMessage2("Error", "Please enter valid numbers to go on !");

} else if (gps != null) {

mesg1();

mesg2();

mesg3();

} else {

showMessage("Error", "Fetching Location, Please wait");

}

}

});

final Button auto = (Button) findViewById(R.id.button2);

assert auto != null;

auto.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

if (cell1.toString().length() <8 &&cell2.toString().length() <8 &&cell3.toString().length()

<8) {

showMessage2("Error", "Please enter valid numbers to go on !");

} else if (gps == null) {

showMessage3("Warning!", "Fetching Location, Please wait");

} else if (cell1.toString().length() >9 || cell2.toString().length() >9 ||

cell3.toString().length() >9) {

Intent i = new Intent(Crash.this, Level.class);

// i.putExtra("gpsf", sms + gps);

startActivity(i);

Crash.this.finish();

} else {

}

}

});

}

public void vall1()

{

Cursor c = myDb.getAllData(); c.moveToFirst();

cell1 = (c.getString(1)); cell2 = (c.getString(2)); cell3 = (c.getString(3));

}

public void showMessage(String title, String Message) { builder = new AlertDialog.Builder(this); builder.setCancelable(true);

builder.setTitle(title); builder.setMessage(Message);

builder.setNeutralButton("Ok", new DialogInterface.OnClickListener() { @Override

public void onClick(DialogInterface dialog, int which) {

}

});

builder.show();

}

public void showMessage2(String title, String Message) { builder = new AlertDialog.Builder(this); builder.setCancelable(true);

builder.setTitle(title); builder.setMessage(Message);

builder.setPositiveButton("Add Number?", new DialogInterface.OnClickListener() { @Override

public void onClick(DialogInterface dialog, int which) { Crash.this.finish();

Intent i = new Intent(Crash.this, Data2.class); startActivity(i);

finish();

{

Cursor c = myDb.getAllData(); c.moveToFirst();

cell1 = (c.getString(1)); cell2 = (c.getString(2)); cell3 = (c.getString(3));

}

public void showMessage(String title, String Message) { builder = new AlertDialog.Builder(this); builder.setCancelable(true);

builder.setTitle(title); builder.setMessage(Message);

builder.setNeutralButton("Ok", new DialogInterface.OnClickListener() { @Override

public void onClick(DialogInterface dialog, int which) {

}

});

builder.show();

}

public void showMessage2(String title, String Message) { builder = new AlertDialog.Builder(this); builder.setCancelable(true);

builder.setTitle(title); builder.setMessage(Message);

builder.setPositiveButton("Add Number?", new DialogInterface.OnClickListener() { @Override

public void onClick(DialogInterface dialog, int which) { Crash.this.finish();

Intent i = new Intent(Crash.this, Data2.class); startActivity(i);

finish();

}

});

builder.setNeutralButton("Not Now", new DialogInterface.OnClickListener() { @Override

public void onClick(DialogInterface dialog, int which) {

}

});

builder.show();

}

public void showMessage3(String title, String Message) { builder = new AlertDialog.Builder(this); builder.setCancelable(true);

builder.setTitle(title); builder.setMessage(Message);

builder.setPositiveButton("Ignore!", new DialogInterface.OnClickListener() { @Override

public void onClick(DialogInterface dialog, int which) { Crash.this.finish();

Intent i = new Intent(Crash.this, Level.class); startActivity(i);

finish();

}

});

builder.setNeutralButton("I can wait :)", new DialogInterface.OnClickListener() { @Override

public void onClick(DialogInterface dialog, int which) {

}

});

builder.show();

}

public void mesg1() {

if (cell1.toString().length() >10) {

try {

SmsManager smsManager = SmsManager.getDefault(); smsManager.sendTextMessage(cell1.toString(), null, sms + gps, null, null); Toast.makeText(getApplicationContext(), "Your message sent to 1st Contact

successfuly !",

Toast.LENGTH\_LONG).show();

} catch (Exception e) { Toast.makeText(getApplicationContext(),

"SMS failddddd!",

Toast.LENGTH\_LONG).show(); e.printStackTrace();

}

} else {

Toast.makeText(getApplicationContext(), "Message not sent to 1st Contact\nYou have not entered number !", Toast.LENGTH\_LONG).show();

}

}

public void mesg2() {

if (cell2.toString().length() >10) {

try {

SmsManager smsManager = SmsManager.getDefault(); smsManager.sendTextMessage(cell2.toString(), null, sms + gps, null, null); Toast.makeText(getApplicationContext(), "Your message sent to 2nd Contact

successfuly !",

Toast.LENGTH\_LONG).show();

} catch (Exception e) { Toast.makeText(getApplicationContext(),

"SMS failddddd!",

Toast.LENGTH\_LONG).show(); e.printStackTrace();

}

} else {

Toast.makeText(getApplicationContext(), "Message not sent to 2nd Contact\n""You have not entered number !", Toast.LENGTH\_LONG).show();

}

}

if (cell3.toString().length() >10) {

try {

SmsManager smsManager = SmsManager.getDefault(); smsManager.sendTextMessage(cell3.toString(), null, sms + gps, null, null); Toast.makeText(getApplicationContext(), "Your message sent to 3rd Contact

successfuly !",

Toast.LENGTH\_LONG).show();

} catch (Exception e) { Toast.makeText(getApplicationContext(),

"SMS failddddd!",

Toast.LENGTH\_LONG).show(); e.printStackTrace();

}

} else {

Toast.makeText(getApplicationContext(), "Message not sent to 3rd Contact\n"

+

"You have not entered number !", Toast.LENGTH\_LONG).show();

}

}

@Override

public void onLocationChanged(Location l) {

this.lat = l.getLatitude();

this.lot = l.getLongitude();

this.gps = "\n" +

"[www.google.com/maps/place/"](http://www.google.com/maps/place/) + lat + "," + lot + "\nIm in critical situation.\n" +

"Sent by iSMASH Pocket resQ !!";

locc.setTextColor(Color.WHITE); locc.setText("New Location Fetched !"); v.vibrate(1000); locc.startAnimation(bounce);

Toast.makeText(getApplicationContext(), "New Location fetched !", Toast.LENGTH\_SHORT).show();

}

@Override

public void onStatusChanged(String provider, int status, Bundle extras) {

}

@Override

public void onProviderEnabled(String provider) {

}

@Override

public void onProviderDisabled(String provider) {

Intent intent = new Intent(Settings.ACTION\_LOCATION\_SOURCE\_SETTINGS); startActivity(intent);

Toast.makeText(getBaseContext(), "Gps is turned off! Turn ir on nd wait. ", Toast.LENGTH\_SHORT).show();

}

@Override

public void onAnimationStart(Animation animation) {

locc.startAnimation(bounce);

}

@Override

public void onAnimationEnd(Animation animation) {

locc.startAnimation(bounce);

}

@Override

public void onAnimationRepeat(Animation animation) {

locc.startAnimation(bounce);

}

};

package com.example.zain.AccidentAlert;

import android.database.Cursor;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.view.View; import android.widget.Button; import android.widget.EditText; import android.widget.TextView; import android.widget.Toast;

public class Data extends AppCompatActivity { TextView nmbr1, nmbr2, nmbr3;

EditText no1, no2, no3; Button nmbr, show;

DatabaseHelper myDb; String id, cell1, cell2, cell3;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState); setContentView(R.layout.activity\_data);

myDb = new DatabaseHelper(this);

nmbr1 = (TextView) findViewById(R.id.textView); nmbr2 = (TextView) findViewById(R.id.textView2); nmbr3 = (TextView) findViewById(R.id.textView3);

no1 = (EditText) findViewById(R.id.editText); no2 = (EditText) findViewById(R.id.editText2); no3 = (EditText) findViewById(R.id.editText3);

nmbr = (Button) findViewById(R.id.button);

show = (Button) findViewById(R.id.button4);

AddData(); UpdateData(); vall1();

no1.setText(cell1); no2.setText(cell2); no3.setText(cell3);

}

public void AddData() {

boolean isInserted = myDb.insertData( "1",

"",

"",

"");

if (isInserted = true) {

// Toast.makeText(Menu.this,"Data Inserted",Toast.LENGTH\_LONG).show();

// vall1();

} else

Toast.makeText(Data.this, "Data not Inserted", Toast.LENGTH\_LONG).show();

}

public void UpdateData() { nmbr.setOnClickListener( new View.OnClickListener() { @Override

public void onClick(View v) {

if ((no1.getText().length() >8 &&no2.getText().length() <1 &&no2.getText().length() <1)

||

(no1.getText().length() >8 &&no2.getText().length() >8

&&no3.getText().length() <1) ||

(no1.getText().length() >8 &&no2.getText().length() >8

&&no3.getText().length() >8))

{

boolean isUpdate = myDb.updateData( "1",

no1.getText().toString(), no2.getText().toString(), no3.getText().toString());

if (isUpdate == true &&no1.getText().length() >8) {

Toast.makeText(Data.this, "Numbers Updated", Toast.LENGTH\_LONG).show();

Data.this.finish();

} else

Toast.makeText(Data.this, "Data not Updated", Toast.LENGTH\_LONG).show();

} else {

Toast.makeText(Data.this, "Enter valid number of more than 8 digits.", Toast.LENGTH\_LONG).show();

}

}

}

);

}

public void vall1()

{

Cursor c = myDb.getAllData(); c.moveToFirst();

cell1 = (c.getString(1)); cell2 = (c.getString(2)); cell3 = (c.getString(3));

show.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

Cursor c = myDb.getAllData(); c.moveToFirst();

id = ("Database ID: " + c.getString(0) + "\n"); cell1 = ("Number 1: " + c.getString(1) + "\n"); cell2 = ("Number 2: " + c.getString(2) + "\n"); cell3 = ("Number 3: " + c.getString(3) + "\n");

Toast.makeText(Data.this, id + cell1 + cell2 + cell3, Toast.LENGTH\_LONG).show();

}

});

}

}

package com.example..pocketresq;

import android.content.ContentValues; import android.content.Context; import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

public class DatabaseHelperextends SQLiteOpenHelper {

public static final String DATABASE\_NAME = "mbk1010"; public static final String TABLE\_NAME = "table1";

public static final String COL\_1 = "ID"; public static final String COL\_2 = "CELL"; public static final String COL\_3 = "CELL2"; public static final String COL\_4 = "CELL3";

public DatabaseHelper(Context context) {

super(context, DATABASE\_NAME, null, 1);

}

@Override

public void onCreate(SQLiteDatabase db) {

db.execSQL("create table " + TABLE\_NAME + " (ID INTEGER PRIMARY KEY AUTOINCREMENT,CELL TEXT,CELL2 TEXT,CELL3 TEXT)");

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

//db.execSQL("DROP TABLE IF EXISTS "+TABLE\_NAME);

//onCreate(db);

}

public boolean insertData(String id, String cell, String cell2, String cell3) {

SQLiteDatabase db = this.getWritableDatabase(); ContentValues contentValues = new ContentValues(); contentValues.put(COL\_1, id);

contentValues.put(COL\_2, cell); contentValues.put(COL\_3, cell2); contentValues.put(COL\_4, cell3);

long result = db.insert(TABLE\_NAME, null, contentValues);

return result != -1;

}

public Cursor getAllData() {

SQLiteDatabase db = this.getWritableDatabase();

Cursor c = db.rawQuery("select \* from " + TABLE\_NAME, null);

return c;

}

public boolean updateData(String id, String cell1, String cell2, String cell3) { SQLiteDatabase db = this.getWritableDatabase();

ContentValues contentValues = new ContentValues(); contentValues.put(COL\_1, id); contentValues.put(COL\_2, cell1); contentValues.put(COL\_3, cell2); contentValues.put(COL\_4, cell3);

db.update(TABLE\_NAME, contentValues, "ID = ?", new String[]{id});

return true;

}

}

Trigger.java

package com.example.mbk82.pocketresq;

import android.Manifest;

import android.content.Context; import android.content.DialogInterface; import android.content.Intent;

import android.content.pm.PackageManager;

import android.database.Cursor; import android.graphics.Color; import android.graphics.Typeface; import android.hardware.Sensor; import android.hardware.SensorEvent;

import android.hardware.SensorEventListener; import android.hardware.SensorManager; import android.location.Location;

import android.location.LocationListener; import android.location.LocationManager; import android.os.Bundle;

import android.os.PowerManager; import android.os.Vibrator; import android.provider.Settings;

import android.support.v4.app.ActivityCompat;

import android.support.v7.app.AlertDialog;

import android.support.v7.app.AppCompatActivity;

import android.telephony.SmsManager; import android.view.WindowManager; import android.widget.TextView; import android.widget.Toast;

import java.text.DecimalFormat;

public class Trigger extends AppCompatActivity implements SensorEventListener, LocationListener {

Start the service

public void startNewService(View view) { startService(new Intent(this, Serv.class));

}

\*/

public Vibrator v; String smsf; TextView tv, acc;

float accmx;

PowerManager.WakeLock wakeLock; DatabaseHelper myDb;

Crash msg;

String id, cell1, cell2, cell3; double gForce;

String gps; double gFmax = 0;

TextView max, may, maz; float x, y, z;

DecimalFormat df; float mxr;

float maxr; double maxrv; private double lat; private double lot;

private LocationManager n; private float lastX, lastY, lastZ;

private SensorManager sensorManager; private Sensor accelerometer;

private float deltaX = 0; private float deltaY = 0; private float deltaZ = 0;

private double vibrateThreshold; private float deltaXMax = 0; private float deltaYMax = 0; private float deltaZMax = 0; private double gvalue;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState); setContentView(R.layout.activity\_trigger);

myDb = new DatabaseHelper(this);

msg = new Crash();

df = new DecimalFormat("###.##");

PowerManager powerManager = (PowerManager) getSystemService(POWER\_SERVICE);

wakeLock = powerManager.newWakeLock(PowerManager.SCREEN\_DIM\_WAKE\_LOCK,

"MyWakelockTag"); wakeLock.acquire();

n = (LocationManager) getSystemService(LOCATION\_SERVICE);

if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_FINE\_LOCATION)

!= PackageManager.PERMISSION\_GRANTED && ActivityCompat.checkSelfPermission(this,

Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {

TODO: Consider calling

ActivityCompat#requestPermissions

// here to request the missing permissions, and then overriding

public void onRequestPermissionsResult(int requestCode, String[] permissions,

int[] grantResults)

to handle the case where the user grants the permission. See the documentation

for ActivityCompat#requestPermissions for more details.

return;

}

equestLocationUpdates(LocationManager.GPS\_PROVIDER, 60000, 100, this);

getWindow().addFlags(WindowManager.LayoutParams.FLAG\_ALLOW\_LOCK\_WHILE\_SC REEN\_ON);

vibrateThreshold = getIntent().getDoubleExtra("v", 0);

gvalue = getIntent().getIntExtra("g", 0);

mxr = getIntent().getFloatExtra("mg", 0);

String fontPath2 = "fonts/Trebuchet MS.ttf";

Typeface tf2 = Typeface.createFromAsset(getAssets(), fontPath2);

tv = (TextView) findViewById(R.id.textView);

assert tv != null;

tv.setText("Automatic Detection Mode is on.\n\n\n" +

"> Do not lock your phone manually when AUTO-MODE is on, " + "But App will lock automatically if you turned on auto-lock.\n"); tv.setTypeface(tf2);

AlertDialog.Builder builder1 = new AlertDialog.Builder(this); builder1.setTitle("READ CAREFULLY !!"); builder1.setMessage(

"> Do not lock your phone manually when AUTO-MODE is on," +

"But App will lock automatically if you turned on auto-lock.\n\n" +

"> Don't put your phone on silent/vibrate mode, to hear the ALARM sound !\n\n"

+

"GO to another app by pressing HOME-BUTTON, Do not press BACK-BUTTON."); builder1.setCancelable(true);

builder1.setNeutralButton(

"OK",

new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int id) {

}

});

AlertDialog alert11 = builder1.create(); alert11.show();

TextView textView = (TextView) alert11.findViewById(android.R.id.message); textView.setTextSize(16);

vall1();

sensorManager = (SensorManager) getSystemService(Context.SENSOR\_SERVICE);

if (sensorManager.getDefaultSensor(Sensor.TYPE\_ACCELEROMETER) != null) {

// success! we have an accelerometer

accelerometer = sensorManager.getDefaultSensor(Sensor.TYPE\_ACCELEROMETER); sensorManager.registerListener(this, accelerometer, SensorManager.SENSOR\_DELAY\_NORMAL);

} else {

// we dont have an accelerometer!

}

maxr = accelerometer.getMaximumRange(); maxrv = maxr / mxr;

TextView sens = (TextView) findViewById(R.id.sens); sens.setText("Sensor Range: " + df.format(maxr) + "m/s2");

Toast.makeText(Trigger.this, vibrateThreshold + "\n" + gvalue + "\n" + maxrv +

"", Toast.LENGTH\_LONG).show();

//initialize vibration

v = (Vibrator) this.getSystemService(Context.VIBRATOR\_SERVICE);

}

public void vall1()

{

Cursor c = myDb.getAllData(); c.moveToFirst();

cell2 = (c.getString(2)); cell3 = (c.getString(3));

}

//onResume() register the accelerometer for listening the events protected void onResume() {

super.onResume(); sensorManager.registerListener(this, accelerometer,

SensorManager.SENSOR\_DELAY\_NORMAL);

}

//onPause() unregister the accelerometer for stop listening the events protected void onPause() {

super.onPause(); sensorManager.unregisterListener(this);

}

cell1 = (c.getString(1));

@Override

public void onAccuracyChanged(Sensor sensor, int accuracy) {

}

@Override

public void onSensorChanged(SensorEvent event) {

max = (TextView) findViewById(R.id.max); may = (TextView) findViewById(R.id.may); maz = (TextView) findViewById(R.id.maz);

displayMaxValues();

// get the change of the x,y,z values of the accelerometer

deltaX = Math.abs(lastX - event.values[0]); deltaY = Math.abs(lastY - event.values[1]); deltaZ = Math.abs(lastZ - event.values[2]);

// if the change is below 10, it is just plain noise

if (deltaX <10)

deltaX = 0;

if (deltaY <10)

deltaY = 0;

if (deltaZ <10)

deltaZ = 0;

// set the last know values of x,y,z

lastX = event.values[0]; lastY = event.values[1]; lastZ = event.values[2];

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

float gX = x / 9.8f; float gY = y / 9.8f; float gZ = z / 9.8f;

gForce = Math.sqrt(gX \* gX + gY \* gY + gZ \* gZ);

if (gForce >gFmax) {

gFmax = gForce;

}

TextView gfM = (TextView) findViewById(R.id.textViewg);

gfM.setText("G-Value is: " + df.format(gForce));

double res2 = (gvalue / 100.0f) \* 40;

double res3 = (gvalue / 100.0f) \* 80;

if (gForce < res2) {

gfM.setBackgroundColor(Color.parseColor("#52793F"));

}

if (gForce > res2 &&gForce < res3)

{

gfM.setBackgroundColor(Color.parseColor("#7f5200"));

}

if (gForce > res3 &&gForce > res2) { gfM.setBackgroundColor(Color.parseColor("#990000"));

}

se(); vibrate(); wkl();

detect();

if (maxr<52.00) { se(); vibrate(); wkl();

detect();

}

if (maxr > 52.00 && maxr <76.00) { highsens1();

se1(); vibrate1(); wkl1();

}

if (maxr > 76.00) { highsens2(); se2(); vibrate2(); wkl2();

}

}

//////////detection started :P /////////

//normal private void se() { if

((deltaX >vibrateThreshold &&deltaY >vibrateThreshold)

|| (deltaX >vibrateThreshold &&deltaZ >vibrateThreshold)

|| (deltaY >vibrateThreshold &&deltaZ >vibrateThreshold)

|| (gForce >gvalue)

|| (gForce >maxrv)

) {

sensorManager.unregisterListener(this);

}

}

private void wkl() {

if

((deltaX >vibrateThreshold &&deltaY >vibrateThreshold)

|| (deltaX >vibrateThreshold &&deltaZ >vibrateThreshold)

|| (deltaY >vibrateThreshold &&deltaZ >vibrateThreshold)

|| (gForce >gvalue)

|| (gForce >maxrv)

) {

wakeLock.release();

}

}

private void detect() {

if ((deltaX >vibrateThreshold &&deltaY >vibrateThreshold)

|| (deltaX >vibrateThreshold &&deltaZ >vibrateThreshold)

|| (deltaY >vibrateThreshold &&deltaZ >vibrateThreshold)

|| (gForce >gvalue)

|| (gForce >maxrv)

) {

Intent i = new Intent(Trigger.this, Detect.class); i.putExtra("gp", smsf);

putExtra("gps", gps);

startActivity(i); finish();

}

}

public void vibrate() {

if ((deltaX >vibrateThreshold &&deltaY >vibrateThreshold)

|| (deltaX >vibrateThreshold &&deltaZ >vibrateThreshold)

|| (deltaY >vibrateThreshold &&deltaZ >vibrateThreshold)

|| (gForce >gvalue)

|| (gForce >maxrv)

) {

brate(3000);

}

}

//high

private void highsens1() {

if ((deltaX > vibrateThreshold && deltaY > vibrateThreshold)

|| (deltaX > vibrateThreshold && deltaZ > vibrateThreshold)

|| (deltaY > vibrateThreshold && deltaZ > vibrateThreshold)

|| (gForce > 16)

|| (gForce > maxrv)

) {

Intent i = new Intent(Trigger.this, Detect.class); i.putExtra("gp", smsf);

putExtra("gps", gps);

startActivity(i); finish();

}

}

private void se1() {

*if*

*((deltaX > vibrateThreshold && deltaY > vibrateThreshold)*

*|| (deltaX > vibrateThreshold && deltaZ > vibrateThreshold)*

*|| (deltaY > vibrateThreshold && deltaZ > vibrateThreshold)*

*|| (gForce > 16)*

*|| (gForce > maxrv)*

*) {*

*sensorManager.unregisterListener(this);*

*}*

*}*

*private void vibrate1() {*

*if ((deltaX > vibrateThreshold && deltaY > vibrateThreshold)*

*|| (deltaX > vibrateThreshold && deltaZ > vibrateThreshold)*

*|| (deltaY > vibrateThreshold && deltaZ > vibrateThreshold)*

*|| (gForce > 16)*

*|| (gForce > maxrv)*

*) {*

*v.vibrate(3000);*

*}*

*}*

*private void wkl1() {*

*if*

*((deltaX > vibrateThreshold && deltaY > vibrateThreshold)*

*|| (deltaX > vibrateThreshold && deltaZ > vibrateThreshold)*

*|| (deltaY > vibrateThreshold && deltaZ > vibrateThreshold)*

*|| (gForce > 16)*

*|| (gForce > maxrv)*

*) {*

*wakeLock.release();*

*}*

*}*

*//very high*

*private void highsens2() {*

*if ((deltaX > vibrateThreshold && deltaY > vibrateThreshold)*

*|| (deltaX > vibrateThreshold && deltaZ > vibrateThreshold)*

*|| (deltaY > vibrateThreshold && deltaZ > vibrateThreshold)*

*|| (gForce > 24)*

*|| (gForce > maxrv)*

*) {*

*Intent i = new Intent(Trigger.this, Detect.class); i.putExtra("gp", smsf);*

*i.putExtra("gps", gps);*

*startActivity(i); finish();*

*}*

*}*

*private void se2() {*

*if ((deltaX > vibrateThreshold && deltaY > vibrateThreshold)*

*|| (deltaX > vibrateThreshold && deltaZ > vibrateThreshold)*

*|| (deltaY > vibrateThreshold && deltaZ > vibrateThreshold)*

*|| (gForce > 24)*

*|| (gForce > maxrv)*

*) {*

*sensorManager.unregisterListener(this);*

*}*

*}*

*private void vibrate2() {*

*if ((deltaX > vibrateThreshold && deltaY > vibrateThreshold)*

*|| (deltaX > vibrateThreshold && deltaZ > vibrateThreshold)*

*|| (deltaY > vibrateThreshold && deltaZ > vibrateThreshold)*

*|| (gForce > 24)*

*|| (gForce > maxrv)*

*) {*

*v.vibrate(3000);*

*}*

*}*

*private void wkl2() {*

*if ((deltaX > vibrateThreshold && deltaY > vibrateThreshold)*

*|| (deltaX > vibrateThreshold && deltaZ > vibrateThreshold)*

*|| (deltaY > vibrateThreshold && deltaZ > vibrateThreshold)*

*|| (gForce > 24)*

*|| (gForce > maxrv)*

*) {*

*wakeLock.release();*

*}*

*}*

*///////// detection ended //////////*

private void displayMaxValues() {

if (deltaX >deltaXMax) {

deltaXMax = deltaX;

}

if (deltaY >deltaYMax) {

deltaYMax = deltaY;

}

if (deltaZ >deltaZMax) {

deltaZMax = deltaZ;

}

accmx = (deltaXMax + deltaYMax + deltaZMax) / 3;

acc = (TextView) findViewById(R.id.accmx);

acc.setText("Max acceleration: " + df.format(accmx) + "m/s2");

max.setText("" + df.format(x));

may.setText("" + df.format(y));

maz.setText("" + df.format(z));

}

public void mesg1() {

if (cell1.toString().length() >10 &&this.gps != null) {

try {

SmsManager smsManager = SmsManager.getDefault(); smsManager.sendTextMessage(cell1.toString(), null, smsf, null, null); Toast.makeText(getApplicationContext(), "Your message sent to 1st Contact

successfuly !",

Toast.LENGTH\_LONG).show();

} catch (Exception e) { Toast.makeText(getApplicationContext(),

"SMS failddddd!",

Toast.LENGTH\_LONG).show(); e.printStackTrace();

}

} else {

Toast.makeText(getApplicationContext(), "Message not sent to 1st Contact\n"

+

"Because You have not entered number\n" +

"OR GPS is turned off and did not fetched location yet !!", Toast.LENGTH\_LONG).show();

}

}

public void mesg2() {

if (cell2.toString().length() >10 &&this.gps != null) {

try {

SmsManager smsManager = SmsManager.getDefault(); smsManager.sendTextMessage(cell2.toString(), null, smsf, null, null); Toast.makeText(getApplicationContext(), "Your message sent to 2nd Contact

successfuly !",

Toast.LENGTH\_LONG).show();

} catch (Exception e) { Toast.makeText(getApplicationContext(),

"SMS failddddd!",

Toast.LENGTH\_LONG).show(); e.printStackTrace();

}

} else {

Toast.makeText(getApplicationContext(), "Message not sent to 2nd Contact\n"

+

"Because You have not entered number\n" +

"OR GPS is turned off and did not fetched location yet !!", Toast.LENGTH\_LONG).show();

}

}

public void mesg3() {

if (cell3.toString().length() >10 &&this.gps != null) {

try {

SmsManager smsManager = SmsManager.getDefault(); smsManager.sendTextMessage(cell3.toString(), null, smsf, null, null); Toast.makeText(getApplicationContext(), "Your message sent to 3rd Contact

successfuly !",

Toast.LENGTH\_LONG).show();

}

public void mesg3() {

if (cell3.toString().length() >10 &&this.gps != null) {

try {

SmsManager smsManager = SmsManager.getDefault(); smsManager.sendTextMessage(cell3.toString(), null, smsf, null, null); Toast.makeText(getApplicationContext(), "Your message sent to 3rd Contact

successfuly !",

Toast.LENGTH\_LONG).show();

} catch (Exception e) { Toast.makeText(getApplicationContext(),

"SMS failddddd!",

Toast.LENGTH\_LONG).show(); e.printStackTrace();

}

} else {

Toast.makeText(getApplicationContext(), "Message not sent to 3rd Contact\n"

+

"Because You have not entered number\nOR GPS is turned off and did not fetched location yet !!", Toast.LENGTH\_LONG).show();

}

}

@Override

public void onBackPressed() {

new AlertDialog.Builder(this)

.setMessage(

"Are you sure to exit AUTO-Detection-Mode?\n\n" +

"If you wanna minimize app then press HOME BUTTON after pressing 'NO' !\n")

.setCancelable(false)

.setPositiveButton("Yes", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int id) { Trigger.this.finish();

}

})

.setNegativeButton("No", null)

.show();

}

@Override

public void onLocationChanged(Location l) {

this.lat = l.getLatitude();

this.lot = l.getLongitude();

this.gps = String.valueOf(this.lat + this.lot); this.smsf = "Help me please, I am at: \n" +

"[www.google.com/maps/place/"](http://www.google.com/maps/place/) + lat + "," + lot + "\nIm in critical situation.\n" +

"Sent by iSMASH Pocket resQ !! AUTO\_DETECTION";

// Toast.makeText(getApplicationContext(), "New Location fetched!", Toast.LENGTH\_SHORT).show();

}

@Override

public void onStatusChanged(String provider, int status, Bundle extras) {

}

@Override

public void onProviderEnabled(String provider) {

Toast.makeText(getApplicationContext(), "Gps is turned on. ", Toast.LENGTH\_SHORT).show();

}

@Override

public void onProviderDisabled(String provider) {

Intent intent = new Intent(Settings.ACTION\_LOCATION\_SOURCE\_SETTINGS); startActivity(intent);

Toast.makeText(getApplicationContext(), "Gps is turned off! Turn ir on nd wait. ", Toast.LENGTH\_SHORT).show();

}

}

import android.hardware.Sensor;

import android.hardware.SensorEvent; import android.hardware.SensorEventListener; import android.hardware.SensorManager; import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.view.View;

import android.widget.ArrayAdapter; import android.widget.Button; import android.widget.Spinner; import android.widget.TextView; import android.widget.Toast;

import java.util.ArrayList;

import java.util.List;

public class Levelextends AppCompatActivity implements SensorEventListener { String smsf;

String pn; TextView info;

float mxr; float maxr; double maxrv;

private Spinner spinner1; private Button btnSubmit; private Button start; private TextView tv5;

private double vibrateThreshold; private int gv;

private SensorManager sensorManager; private Sensor accelerometer;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState); setContentView(R.layout.activity\_level);

// smsf = getIntent().getExtras().getString("gpsf");

// pn = getIntent().getExtras().getString("pn");

//Toast.makeText(Level.this, smsf+pn,Toast.LENGTH\_LONG).show();

String fontPath = "fonts/m.ttf";

Typeface tf = Typeface.createFromAsset(getAssets(), fontPath); TextView txt2 = (TextView) findViewById(R.id.textView11); txt2.setText("Set detection level from '1-5'"); txt2.setTypeface(tf);

String fontPath2 = "fonts/Trebuchet MS.ttf";

Typeface tf2 = Typeface.createFromAsset(getAssets(), fontPath2);

info = (TextView) findViewById(R.id.tv5);

info.setText(

"(1) is best for smooth and fine roads\n" +

"(5) is best for hilly or jumpy roads\n\n" +

"So Choose 1-5 carefully according to your road conditions.\n"); info.setTypeface(tf2);

sensorManager = (SensorManager) getSystemService(Context.SENSOR\_SERVICE);

if (sensorManager.getDefaultSensor(Sensor.TYPE\_ACCELEROMETER) != null) {

// success! we have an accelerometer

accelerometer = sensorManager.getDefaultSensor(Sensor.TYPE\_ACCELEROMETER); sensorManager.registerListener(this, accelerometer, SensorManager.SENSOR\_DELAY\_NORMAL);

} else {

// we dont have an accelerometer!

}

maxr = accelerometer.getMaximumRange();

spinner1 = (Spinner) findViewById(R.id.spn);

List<Integer> list = new ArrayList<Integer>(); list.add(1);

list.add(2);

list.add(3);

list.add(4);

list.add(5);

ArrayAdapter<Integer> dataAdapter = new ArrayAdapter<Integer>(this, R.layout.spin, list);

dataAdapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_it em);

spinner1.setAdapter(dataAdapter);

addListenerOnButton(); addListenerOnSpinnerItemSelection();

spinner1.setSelection(2);

}

public void addListenerOnSpinnerItemSelection() {

spinner1.setOnItemSelectedListener(new CustomOnItemSelectedListener());

}

public void addListenerOnButton() { spinner1 = (Spinner) findViewById(R.id.spn); btnSubmit = (Button) findViewById(R.id.sub);

btnSubmit.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

int id = Integer.parseInt(String.valueOf(spinner1.getSelectedItem()));

// Toast.makeText(Level.this,v,Toast.LENGTH\_LONG).show();

//2g,4g,8g,16g//20-40,40-80,80-160,160-320//2-4,4-8,8-16,16-32///

if (id == 1) {

Toast.makeText(Level.this, "Senstivity level = 1.", Toast.LENGTH\_LONG).show();

vibrateThreshold = maxr \* 1.25;

if (maxr >00.00 &&maxr <20.00) {

gv = 2;

} //4

if (maxr >20.00 &&maxr <40.00) {

gv = 3;

} //8

if (maxr >40.00 &&maxr <80.00) {

gv = 5;

} //16

if (maxr >80.00 &&maxr <256.00) {

gv = 8;

} //32

mxr = 8.16f;

} else if (id == 2) {

Toast.makeText(Level.this, "Senstivity level = 2.", Toast.LENGTH\_LONG).show();

vibrateThreshold = maxr \* 1.50;

if (maxr >00.00 &&maxr <20.00) {

gv = 3;

} //4

if (maxr >20.00 &&maxr <40.00) {

gv = 5;

} //8

if (maxr >40.00 &&maxr <80.00) {

gv = 10;

} //16

if (maxr >80.00 &&maxr <256.00) {

gv = 15;

} //32

mxr = 6.53f;

} else if (id == 3) {

Toast.makeText(Level.this, "Senstivity level = 3.", Toast.LENGTH\_LONG).show();

vibrateThreshold = maxr \* 2.25;

if (maxr >00.00 &&maxr <20.00) {

gv = 3;

} //4

if (maxr >20.00 &&maxr <40.00) {

gv = 7;

} //8

if (maxr >40.00 &&maxr <80.00) {

gv = 12;

} //16

if (maxr >80.00 &&maxr <256.00) {

gv = 18;

} //32

mxr = 4.9f;

} else if (id == 4) {

Toast.makeText(Level.this, "Senstivity level = 4.", Toast.LENGTH\_LONG).show();

vibrateThreshold = maxr \* 2.50;

if (maxr >00.00 &&maxr <20.00) {

gv = 5;

} //4

if (maxr >20.00 &&maxr <40.00) {

gv = 10;

} //8

if (maxr >40.00 &&maxr <80.00) {

gv = 16;

} //16

if (maxr >80.00 &&maxr <256.00) {

gv = 23;

} //32

mxr = 4.00f;

} else if (id == 5) {

Toast.makeText(Level.this, "Senstivity level = 5.", Toast.LENGTH\_LONG).show();

vibrateThreshold = maxr \* 3.00;

if (maxr >00.00 &&maxr <20.00) {

gv = 6;

} //4

if (maxr >20.00 &&maxr <40.00) {

gv = 12;

} //8

if (maxr >40.00 &&maxr <80.00) {

gv = 18;

} //16

if (maxr >80.00 &&maxr <256.00) {

gv = 25;

} //32

mxr = 3.00f;

} else {

vibrateThreshold = maxr \* 2.25;

if (maxr >00.00 &&maxr <20.00) {

gv = 4;

} //4

if (maxr >20.00 &&maxr <40.00) {

gv = 8;

} //8

if (maxr >40.00 &&maxr <80.00) {

gv = 14;

} //16

if (maxr >80.00 &&maxr <256.00) {

gv = 21;

} //32

mxr = 4.9f;

}

Intent i = new Intent(Level.this, Trigger.class); i.putExtra("v", vibrateThreshold); i.putExtra("g", gv);

i.putExtra("mg", mxr);

// i.putExtra("gpsf", smsf);

// i.putExtra("pn", pn);

startActivity(i);

finish();

}

});

}

@Override

public void onSensorChanged(SensorEvent event) {

}

@Override

public void onAccuracyChanged(Sensor sensor, int accuracy) {

}

}

**Detect.java**

package com.example.mbk82.pocketresq;

import android.database.Cursor; import android.graphics.Typeface; import android.media.AudioManager; import android.media.MediaPlayer;

import android.os.Bundle;

import android.os.CountDownTimer;

import android.support.v7.app.AppCompatActivity;

import android.telephony.SmsManager;

import android.view.View;

import android.view.WindowManager;

import android.widget.Button; import android.widget.TextView; import android.widget.Toast;

public class Detectextends AppCompatActivity {

private final long startTime = 15 \* 1000; private final long interval = 1 \* 1000; public TextView text;

MediaPlayer mMediaPlayer; AudioManager au;

int uv;

DatabaseHelper myDb; TextView tv;

String id, cell1, cell2, cell3; String gps;

String smsf;

private CountDownTimer countDownTimer; private Button safe;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState); setContentView(R.layout.activity\_detect);

myDb = new DatabaseHelper(this);

smsf = getIntent().getExtras().getString("gp");

gps = getIntent().getExtras().getString("gps");

Toast.makeText(getApplicationContext(), ""+smsf+gps, Toast.LENGTH\_SHORT).show();

vall1();

getWindow().addFlags(WindowManager.LayoutParams.FLAG\_TURN\_SCREEN\_ON); getWindow().addFlags(WindowManager.LayoutParams.FLAG\_KEEP\_SCREEN\_ON);

getWindow().addFlags(WindowManager.LayoutParams.FLAG\_SHOW\_WHEN\_LOCKED);

text = (TextView) findViewById(R.id.timer);

countDownTimer = new MyCountDownTimer(startTime, interval); text.setText(text.getText() + String.valueOf(startTime / 1000));

countDownTimer.start();

safe = (Button) findViewById(R.id.safe);

safe.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

finish();

}

});

au = (AudioManager) getSystemService(AUDIO\_SERVICE);

uv = this.au.getStreamVolume(AudioManager.STREAM\_MUSIC);

au.setStreamVolume(AudioManager.STREAM\_MUSIC, this.au.getStreamMaxVolume(AudioManager.STREAM\_MUSIC), AudioManager.FLAG\_PLAY\_SOUND);

mMediaPlayer = new MediaPlayer();

mMediaPlayer = MediaPlayer.create(this, R.raw.s);

mMediaPlayer.setVolume(100, 100); mMediaPlayer.setAudioStreamType(AudioManager.STREAM\_MUSIC); mMediaPlayer.setLooping(true);

mMediaPlayer.start();

String fontPath2 = "fonts/Trebuchet MS.ttf";

Typeface tf2 = Typeface.createFromAsset(getAssets(), fontPath2);

tv = (TextView) findViewById(R.id.textView2);

assert tv != null; tv.setText("ACCIDENT DETECTED !" +

"\n\n" +

"Press 'I am safe' button if this is an false alarm!\n"); tv.setTypeface(tf2);

}

@Override

protected void onDestroy() {

// TODO Auto-generated method stub

if (mMediaPlayer.isPlaying())

mMediaPlayer.stop();

mMediaPlayer.release(); countDownTimer.cancel();

super.onDestroy();

}

public void vall1()

{

Cursor c = myDb.getAllData();

public void mesg1() {

if (cell1.toString().length() >10 &&this.gps != null) {

try {

SmsManager smsManager = SmsManager.getDefault(); smsManager.sendTextMessage(cell1.toString(), null, smsf, null, null); Toast.makeText(getApplicationContext(), "Your message sent to 1st Contact

successfuly !",

Toast.LENGTH\_LONG).show();

} catch (Exception e) { Toast.makeText(getApplicationContext(),

"SMS failddddd!",

Toast.LENGTH\_LONG).show(); e.printStackTrace();

}

} else {

Toast.makeText(getApplicationContext(), "Message not sent to 1st Contact\n"

+

"Because You have not entered number\n" +

"OR GPS is turned off and did not fetched location yet !!", Toast.LENGTH\_LONG).show();

}

}

public void mesg2() {

if (cell2.toString().length() >10 &&this.gps != null) {

try {

SmsManager smsManager = SmsManager.getDefault(); smsManager.sendTextMessage(cell2.toString(), null, smsf, null, null); Toast.makeText(getApplicationContext(), "Your message sent to 2nd Contact

successfuly !",

Toast.LENGTH\_LONG).show();

} catch (Exception e) { Toast.makeText(getApplicationContext(),

"SMS failddddd!",

Toast.LENGTH\_LONG).show(); e.printStackTrace();

}

} else {

Toast.makeText(getApplicationContext(), "Message not sent to 2nd Contact\n"

+

"Because You have not entered number\n" +

"OR GPS is turned off and did not fetched location yet !!", Toast.LENGTH\_LONG).show();

}

}

public void mesg3() {

if (cell3.toString().length() >10 &&this.gps != null) {

try {

SmsManager smsManager = SmsManager.getDefault(); smsManager.sendTextMessage(cell3.toString(), null, smsf, null, null); Toast.makeText(getApplicationContext(), "Your message sent to 3rd Contact

successfuly !",

Toast.LENGTH\_LONG).show();

} catch (Exception e) { Toast.makeText(getApplicationContext(),

"SMS failddddd!",

Toast.LENGTH\_LONG).show();

e.printStackTrace();

}

} else {

Toast.makeText(getApplicationContext(), "Message not sent to 3rd Contact\n"

+

"Because You have not entered number\nOR GPS is turned off and did not fetched location yet !!", Toast.LENGTH\_LONG).show();

}

}

private class MyCountDownTimer extends CountDownTimer { public MyCountDownTimer(long startTime, long interval) { super(startTime, interval);

}

@Override

public void onTick(long millisUntilFinished) {

text.setText("" + millisUntilFinished / 1000);

}

@Override

public void onFinish() {

text.setText("Time's up!"); safe.setText("Message Sent !");

if (mMediaPlayer.isPlaying())

mMediaPlayer.stop();

mesg1();

mesg2();

mesg3();

}

}}