

# **1.INTRODUCTION**

## **1.1 Project overview**

In a densely populated country like India, it is very difficult to prevent the community transmission even during lockdown without social awareness and precautionary measures taken by the people. Recently, several containment zones had been identified throughout the country and divided into red, orange and green zones, respectively. The red zones indicate the infection hotspots, orange zones denote some infection and green zones indicate an area with no infection. This project mainly focuses on development of an Android application which can inform people of the Covid-19 containment zones and prevent trespassing into these zones. This Android application updates the locations of the areas in which are identified to be the containment zones. The application also notifies the users if they have entered a containment zone and uploads the user's mobile number to the database. To achieve all these functionalities, many tools, and APIs from Google like Firebase and Geofencing API are used in this application. Therefore, this application can be used as a tool for creating further social awareness about the arising need of precautionary measures to be taken by the people of India.

## **1.2 Purpose**

This mobile based application to provide information regarding the Covid-19 containment zones. The application further tracks the user's location and provides notification alert if the user has entered a containment zone. The application also provides daily Covid19 case statistics to the users to keep them updated. The application is developed on Android SDK and uses Firebase Cloud Firestore to store the location data. Android's geofencing client is used to create geofences around the containment zones and notification manager is used to provide notifications. We have tested our application with different users in different location and it works efficiently and is able to attain our target.

## 2. LITERATURE SURVEY

### 2.1 Existing problems

The parameters used are similar, but the exact criteria applied varies, and usually depends on local conditions. These have also evolved with time, and are under constant review. In general, containment zones are getting smaller with time as the number of cases are increasing — from entire localities, to colonies or neighbourhood, to streets and lanes, to particular buildings, and now just particular floors. As of now, in [Delhi](#), a containment zone is declared if three or more infections are detected. In Gurgaon, if five positive cases emerge within a 1-km radius, that area is designated a containment zone. In Noida, an area within a radius of 250 m, or one floor of a building, is declared a containment zone, even if one person is found positive.

While some cities de-notify areas after 28 days if no fresh case emerges from the containment zone, this period is seven or 14 days in some other cities. The perimeter of the containment zone is also different in different cities.

### 2.2 References

Android developer-Locations (2020)

[https://developer.android.com/reference/android/location/Location#distanceBetween\(double,%20double,%20double,%20double,%20float\[\]\)](https://developer.android.com/reference/android/location/Location#distanceBetween(double,%20double,%20double,%20double,%20float[]))

### 2.3 Problem Statement Definition

The survey shows that there are several apps developed in the country to fight and contain COVID-19. Most of the states of our country have their own apps with specific features and functionality to help their citizens to stop COVID-19 spread, get medical assistance during a crisis, create awareness, and

understand safety precautions. The study also shows that there are a limited number of apps which show the COVID-19 containment zones in the country or state and out of these none has the functionality of notifying and alerting the user when they have entered a containment zone. Therefore, no app in the Google Playstore is comparable with our proposed application because the idea behind the development of the proposed app is different. This highlights the novelty of the proposed app.

### 3. IDEATION AND PROPOSED SOLUTION

#### 3.1 Empathy Map Canvas

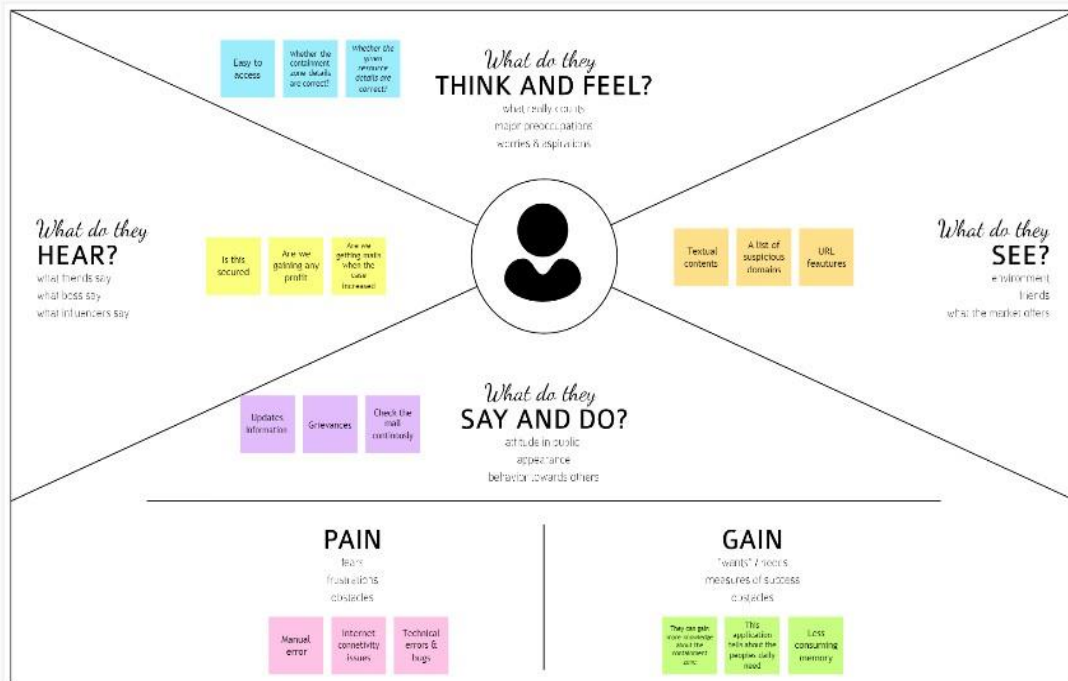
# Empathy Map Canvas

Gain insight and understanding on solving customer problems.

## Containment Zone Alerting Application

1

Build empathy and keep your focus on the user by putting yourself in their shoes.



Share your feedback

## 3.2 Ideation & Brainstroming

### Brainstorm & Idea prioritization

Use this template to plan your brainstorming sessions and to capture and organize your ideas and prioritizing concepts that you're not adding in the same way.

- Brainstorming
- Idea prioritization
- Idea selection

#### Before you collaborate

1. Set up your session (date, time, location, duration, etc.)
2. Invite participants

[View instructions](#)

#### Define your problem statement

Write a clear, concise statement of the problem you want to solve. This will help you focus your brainstorming efforts.

[View instructions](#)

#### Brainstorm

Write down all your ideas, no matter how silly or outrageous they may seem. The more ideas you have, the better.

[View instructions](#)

Category	Item 1	Item 2	Item 3	Item 4	Item 5
Category 1	Item 1.1	Item 1.2	Item 1.3	Item 1.4	Item 1.5
Category 2	Item 2.1	Item 2.2	Item 2.3	Item 2.4	Item 2.5
Category 3	Item 3.1	Item 3.2	Item 3.3	Item 3.4	Item 3.5
Category 4	Item 4.1	Item 4.2	Item 4.3	Item 4.4	Item 4.5
Category 5	Item 5.1	Item 5.2	Item 5.3	Item 5.4	Item 5.5

#### Group ideas

Write down all your ideas, no matter how silly or outrageous they may seem. The more ideas you have, the better.

[View instructions](#)

#### Prioritize

Write down all your ideas, no matter how silly or outrageous they may seem. The more ideas you have, the better.

[View instructions](#)

#### After you collaborate

Write down all your ideas, no matter how silly or outrageous they may seem. The more ideas you have, the better.

[View instructions](#)

### 3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	contain potentially harmful particles from circulating outside the construction area. They also help with disease control. It notify the user when she/he trespasses the boundary of the containment zone.
2.	Idea / Solution description	All the information you need about your containment zone can be in your packet. With mobile solutions and cloud based software.
3.	Novelty / Uniqueness	After the user logged into the app it will track the user's location and update the database with the current location. If the user is visiting the containment zone he will get an alert notification.
4.	Social Impact / Customer Satisfaction	It helps the users to know if they are in a safe zone are in a containment area. It helps to track people going out of containment zones in violation. It gives personal information regarding their health scenario.
5.	Business Model (Revenue Model)	With proper containment zone application we spent money on inventory that sells. So cash is always moving.
6.	Scalability of the Solution	To increase the scalability of this application we gives updates over a particular period of time.

### 3.4 Problem Solution Fit

<b>1. CUSTOMER SEGMENT(S)</b> The people who are in the affected area are the customers. <b>CS</b>	<b>6. CUSTOMER CONSTRAINTS</b> Spending power, budget, no cash, network connection, available devices are the constraints which prevent our customers from taking action or limit their choices. <b>CS</b>	<b>5. AVAILABLE SOLUTIONS</b> Gives us the exact location and the details about the vaccine wherever the government provides. <b>CS</b> Gives us the exact location of the available food source. <b>CS</b>	<b>Explore AS, different</b>
<b>2. JOBS-TO-BE-DONE / PROBLEMS</b> Finding infected locations and providing foods are some of the jobs to be done. <b>CS</b>	<b>9. PROBLEM ROOT CAUSE</b> The real reason that this problem exists because of unawareness of the spread of virus. <b>RC</b>	<b>7. BEHAVIOUR</b> Find the right application calculate usage and benefits; indirectly associated: customers spend free time on volunteering work. <b>BE</b>	
<b>Focus on J&amp;P, tap into BE, understand RC</b>			<b>Focus on J&amp;P, tap into BE, understand RC</b>

## 4. REQUIREMENT ANALYSIS

### 4.1 Functional Requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Account	Has all access Receives all Notification
FR-4	User Database	Store details

## 4.2 Non-Functional Requirement

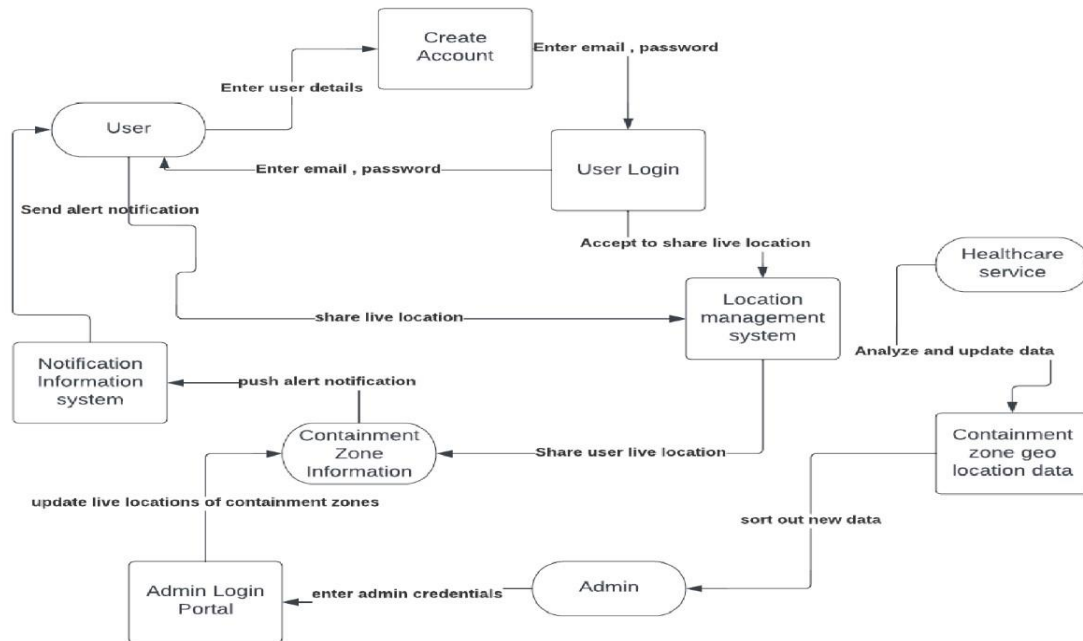
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	<b>Usability</b>	It indicates how effectively they can learn and use a system.
NFR-2	<b>Security</b>	Ensure your softwares safety against espionage.
NFR-3	<b>Reliability</b>	Probability and percentage of the software performing with out failure for a specific amount of time.
NFR-4	<b>Performance</b>	How fast a system can respond to a particular users actions.
NFR-5	<b>Availability</b>	The amount of time the system is running.
NFR-6	<b>Scalability</b>	It assesses the highest workloads under the system will meet.

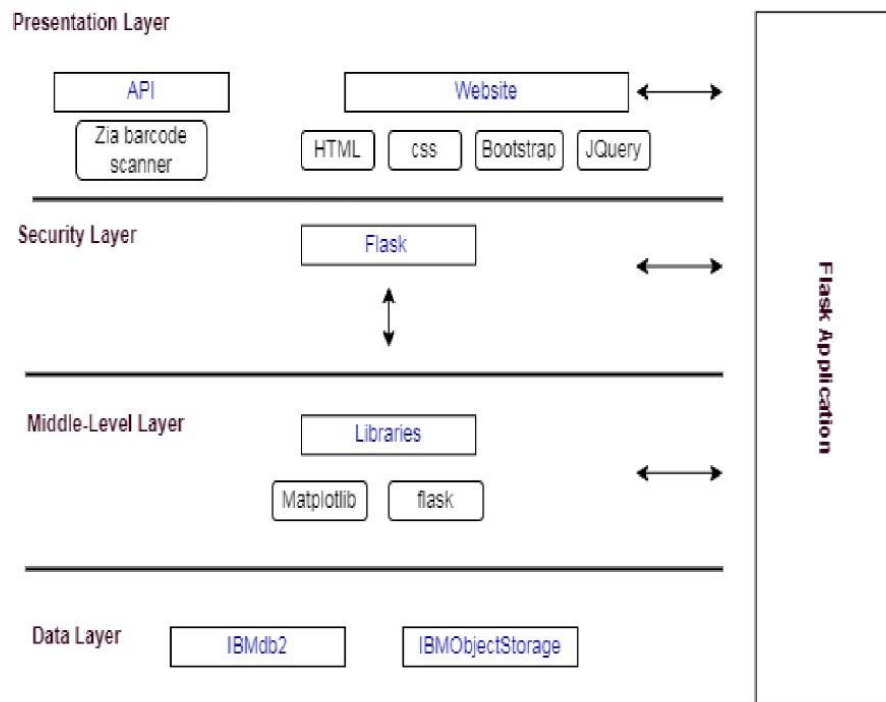
## 5. PROJECT DESIGN

### 5.1 DataFlow Diagrams





## 5.2 Solution & Technical Architechture



## 5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register for the application through gmail.	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can login by entering gmail and password.	High	Sprint-1
	Dashboard	USN-6	As a user, I can track data of containment alerting application.	I can track data of containment alerting application.	High	Sprint-1
Customer (Web user)	Registration	USN-7	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
Customer Care Executive	Support	USN-8	As a executive, I provide answers for the queries asked by users.	I provide answers for the queries asked by users.	High	Sprint-1
Administrator	Manage the application	USN-9	As a administrator, I manage the user details by adding, deleting, and storing.	I manage the user details by adding, deleting, and storing.	High	Sprint-1

## 6. PROJECT PLANNING & SCHEDULING

### 6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	User can create an account by providing mail id and password.	5	High	1,2,3,4
Sprint-2	Registration/Login	USN-2	Two step authentication using one time password to provide mail id or phone number.	10	High	1,2,3,4
Sprint-1	Login	USN-3	Using registered mail id.	5	high	1,2,3,4
Sprint-1	Main dashboard	USN-4	User need to complete account settings like giving the details about their address.	10	high	1,2,3,3
Sprint-2	Hub maintenance	USN-5	User can able create separate account for individual hub.	10	High	1,2,3,4
Sprint-4	Communication System	USN-6	Users and Hub managers can get the details of the people through via mail.	20	Medium	1,2,3,4

### 6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05-Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

## 7. CODING & SOLUTIONING

### 7.1 Feature 1

#### Videos

We can see containment zone related videos in this feature.

```
package com.teapink.damselindistress.activities;
```

```
import android.net.Uri; import  
android.os.Environment;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle; import android.view.View;  
import android.widget.ImageButton; import  
android.widget.ImageView; import  
android.widget.MediaController; import  
android.widget.Toast;  
import android.widget.VideoView;
```

```
import com.teapink.damselindistress.R;
```

```
import java.io.File;
```

```
public class VideoActivity extends AppCompatActivity {
```

```
    int i=0;  
    String path;  
    VideoView videoView;  
    MediaController mediaController;
```

```
    File[] files;  
    @Override
```

```

        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_video);
            videoView =(VideoView)findViewById(R.id.videoView);

            //Creating MediaController
            mediaController= new MediaController(this);
            mediaController.setAnchorView(videoView);

            path=Environment.getExternalStorageDirectory().getPath()+"/Videos";
            //specify the location of media file

            File directory = new File(path);
            files = directory.listFiles();

            Uri uri=Uri.parse(path+"/"+files[i].getName());
            videoView.setMediaController(mediaController);
            videoView.setVideoURI(uri);      videoView.requestFocus();
            videoView.start();

            Toast.makeText(getApplicationContext(),
                files[i].getName()+String.valueOf(files.length),
                Toast.LENGTH_LONG).show();

            ImageButton img_prev=(ImageButton)
            findViewById(R.id.imageButton);

            img_prev.setOnClickListener(new View.OnClickListener() {

```

## 7.2 Feature 2

## Emergency Contacts

This feature helps to show the alert message to the selected contacts.

```
package com.teapink.damselindistress.activities;
```

```
import android.content.DialogInterface; import  
android.content.Intent; import  
android.content.SharedPreferences; import  
android.database.Cursor; import android.net.Uri;  
import android.os.Bundle; import  
android.preference.PreferenceManager; import  
android.provider.ContactsContract;  
import android.support.design.widget.FloatingActionButton;  
import android.support.design.widget.Snackbar; import  
android.support.v7.app.AlertDialog; import  
android.support.v7.app.AppCompatActivity; import  
android.support.v7.widget.LinearLayoutManager; import  
android.support.v7.widget.RecyclerView; import  
android.support.v7.widget.Toolbar; import android.util.Log;  
import android.view.View;  
import android.widget.Toast;
```

```
import com.google.gson.Gson; import  
com.google.gson.reflect.TypeToken; import  
com.teapink.damselindistress.R; import  
com.teapink.damselindistress.adapters.Cont  
actListAdapter; import  
com.teapink.damselindistress.models.Contac  
t; import  
com.teapink.damselindistress.models.User;  
import
```

```

com.teapink.damselindistress.utilities.RecyclerViewClickListener;

import java.lang.reflect.Type;
import java.util.ArrayList;

public class EmergencyContactsActivity extends AppCompatActivity {

    private final String TAG = this.getClass().getSimpleName();
    private static final int CONTACT_PICKER_RESULT = 1001;
    private User user;    private ContactListAdapter
contactListAdapter;    private ArrayList<Contact>
contactArrayList;
    private SharedPreferences sp;
    DataBaseAdapter db;
    String currentUser;
    String stuff;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_emergency_contacts);    Toolbar
toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        db=new DataBaseAdapter(this);
        user = new User();
        contactArrayList = new ArrayList<>();

        RecyclerView emergencyContactsList = (RecyclerView)
findViewById(R.id.emergencyContactsList);
        ArrayList<Contact> contactList = new ArrayList<Contact>();

```

## **8. Testing**

### **8.1 Testcases**

A test case is a document, which has a set of test data, preconditions, expected results and postconditions, developed for a particular test scenario in order to verify compliance against a specific requirement.

Test Case acts as the starting point for the test execution, and after applying a set of input values, the application has a definitive outcome and leaves the system at some end point or also known as execution postcondition.

### **8.2 User Acceptance Testing**

User Acceptance Testing (UAT) is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing is done.

## **9. RESULTS**

### **9.1 Performance Matrices**





## 10. ADVANTAGES AND DISADVANTAGES

### Advantages

This mobile based application to provide information regarding the Covid-19 containment zones. The application further tracks the user's location and provides notification alert if the user has entered a containment zone. It provides daily case statistics to the users to keep them updated. The application is developed on Android SDK and uses Firebase Cloud Firestore to store the location data. Android's geofencing client is used to create geofences around the containment zones and notification manager is used to provide notifications.

## **Disadvantages**

The parameters used are similar, but the exact criteria applied varies, and usually depends on local conditions. These have also evolved with time, and are under constant review. In general, containment zones are getting smaller with time as the number of cases are increasing — from entire localities, to colonies or neighbourhood, to streets and lanes, to particular buildings, and now just particular floors.

## **11.CONCLUSION**

The application provides an efficient way of showing the identified COVID-19 containment zones to the users in a Google map. With the alarming increase of COVID-19 affected cases throughout the world, this developed application can be employed as a tool for creating further social awareness among the people. This application further tracks the user's location and checks whether it is present in the list of identified containment zones. It sends separate notification alerts to the user on entering and exiting the containment areas

## **12. FUTURE SCOPE**

The developed android application further extracts the IMEI Number of the trespasser in the containment zones which can be useful to the local police to track and identify people who are frequently trespassing the containment zones. Thereby this application identifies the containment zones and highlights the need for taking further precautionary measures for combating COVID-19. The application has been tested in various locations and has been found to yield accurate results.

## 12.APPENDIX

### Source Code

```
package com.teapink.damselindistress.activities;

import android.Manifest; import android.content.Context;
import android.content.DialogInterface; import
android.content.Intent; import
android.content.SharedPreferences; import
android.content.pm.PackageInfo; import
android.content.pm.PackageManager; import
android.database.Cursor; import
android.location.Location; import
android.media.AudioManager; import
android.media.MediaPlayer; import
android.net.ConnectivityManager; import
android.os.Build; import android.os.Bundle; import
android.preference.PreferenceManager; import
android.support.annotation.NonNull; import
android.support.design.widget.NavigationView; import
android.support.v4.app.ActivityCompat; import
android.support.v4.content.ContextCompat; import
android.support.v4.view.GravityCompat; import
android.support.v4.widget.DrawerLayout; import
android.support.v7.app.ActionBarDrawerToggle; import
android.support.v7.app.AlertDialog; import
android.support.v7.app.AppCompatActivity; import
android.support.v7.widget.Toolbar; import
android.telephony.SmsManager; import
```

```

import android.text.SpannableString; import
import android.text.method.LinkMovementMethod;
import android.text.util.Linkify; import
import android.util.Log;
import android.view.MenuItem; import
import android.widget.CompoundButton; import
import android.widget.TextView; import
import android.widget.Toast;
import android.widget.ToggleButton;

import com.teapink.damselindistress.R;

import com.teapink.damselindistress.services.ShakeSensorService;

import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices; import
import com.google.android.gms.tasks.OnSuccessListener;

import static android.Manifest.permission.ACCESS_FINE_LOCATION;

public class MainActivity extends AppCompatActivity
    implements NavigationView.OnNavigationItemSelectedListener {

    private final String TAG = this.getClass().getSimpleName();
    private final String TAG_SENSOR =
ShakeSensorService.class.getSimpleName(); private static final int
PERMISSION_ALL = 1; private static final double RADIUS_OF_EARTH =
6371e3; // metres private static final double NEARBY_DISTANCE =
2000; // metres private ToggleButton toggleButton; private
MediaPlayer mediaPlayer;
    DataBaseAdapter db;
    String phoneNo;
    String message;

```

```

String stuff;
String msg_loc;
    private SharedPreferences sp;    private
FusedLocationProviderClient client;
    private static final int MY_PERMISSIONS_REQUEST_SEND_SMS =0 ;
@Override
    protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    requestPermission();
    client =
LocationServices.getFusedLocationProviderClient(MainActivity.this);

    // prepareDistressAlert();
    startService(new Intent(getApplicationContext(),
ShakeSensorService.class));
    // startService(new Intent(getApplicationContext(),
LocationTrackerService.class));
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
setSupportActionBar(toolbar);

    db=new DataBaseAdapter(this);
    // Bundle bundle = getIntent().getExtras();
    // stuff=bundle.getString("stuff");

sp=PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
    stuff=sp.getString("stuff","defaultStringIfNothingFound");

    // getEmergencyContacts1(stuff);

```

```
// retrieve user information
```

```
    if (ActivityCompat.checkSelfPermission(MainActivity.this,
ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED )
{
    return;
}
    client.getLastLocation().addOnSuccessListener(MainActivity.this,
new OnSuccessListener<Location>() {
    @Override
    public void onSuccess(Location location) {
if (location != null) {

        String lati=String.valueOf(location.getLatitude());
        String longi=String.valueOf(location.getLongitude());
        //textView.setText(lati+","+longi);
        // ed_lat.setText(lati);
        // ed_long.setText(longi);

        msg_loc=lati+","+longi;
        // Toast.makeText(getApplicationContext(),msg_loc,
Toast.LENGTH_LONG).show();

    }
}
});
DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
```

```
        final ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
this, drawer, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close);    drawer.addDrawerListener(toggle);
toggle.syncState();
```

```
        NavigationView navigationView = (NavigationView)
findViewById(R.id.nav_view);
        navigationView.setNavigationItemSelectedListener(this);
```

```
        // The request code used in ActivityCompat.requestPermissions()
        // and returned in the Activity's onRequestPermissionsResult()
String[] PERMISSIONS = new String[0];    try {
        PERMISSIONS = getPermissions(getApplicationContext());
    } catch (PackageManager.NameNotFoundException e) {
e.printStackTrace();
    }
```

```
        if(!hasPermissions(this, PERMISSIONS)){
            ActivityCompat.requestPermissions(this, PERMISSIONS,
PERMISSION_ALL);
        }
```

```
        toggleButton = (ToggleButton) findViewById(R.id.panicBtn);
```

```
        toggleButton.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            public void onCheckedChanged(CompoundButton buttonView,
boolean isChecked) {
                if (isChecked) {
prepareDistressAlert();            //
The toggle is enabled
```

```

toggleButton.setBackground(ContextCompat.getDrawable(getApplication
nContext(), R.drawable.circle_bg_off));
        Toast.makeText(getApplicationContext(),
R.string.sound_playing_message, Toast.LENGTH_SHORT).show();
        //setMediaVolumeMax();
        mediaPlayer = MediaPlayer.create(getApplicationContext(),
R.raw.sample);
        mediaPlayer.setLooping(true);
        mediaPlayer.start();

    } else {
        // The toggle is disabled
toggleButton.setBackground(ContextCompat.getDrawable(getApplication
nContext(), R.drawable.circle_bg_on));
        Toast.makeText(getApplicationContext(),
R.string.sound_stopped_message, Toast.LENGTH_SHORT).show();
        mediaPlayer.stop();
        mediaPlayer.reset();
    }
}
});

```

## Github Link

IBM-EPBL/IBM-Project-39316-1660405778

## Demo video Link

[https://drive.google.com/file/d/1\\_HA3NhLfZVd3tmsbFuD5Ly9TCwKVSCs/view?usp=drivesdk](https://drive.google.com/file/d/1_HA3NhLfZVd3tmsbFuD5Ly9TCwKVSCs/view?usp=drivesdk)



