Board of Technical Education

SRI JAYACHAMARAJENDRA (GOVT.) POLYTECHNIC

Department of computer science and Engineering

Seshadri Road, K R Circle, Bangalore -01

PROJECT REPORT

ON

"GEOFENSE BASED AUTO SILENT APP"

Submitted by:

Nanda kumar (102CS18027)

Nehan ashin tony(102CS18031)

Prajwal P (102CS18033)

Sarvesh V (102CS18044)

in fulfilment of requirement for the award of

DIPLOMA IN COMPUTER SCIENCE AND ENGINEERING

Under the guidance of

Mrs. Poojitha K B

Lecturer, Department of computer science and Engineering

S.J. (Govt.) Polytechnic

CERTIFICATE

Board of Technical Education

SRI JAYACHAMARAJENDRA (GOVT.) POLYTECHNIC

Department of computer science and Engineering

Seshadri Road, K R Circle, Bangalore -01

This is to certify that the Project Work Entitled

"GEOFENSE BASED AUTO SILENT APP"

Is Work of:

Nanda kumar (102CS18027)
Nehan ashin tony(102CS18031)
Prajwal P (102CS18033)
Sarvesh V (102CS18044)

The bonfire students of S.J (Govt.) Polytechnic in partial fulfilment of requirement for the award of Diploma in Computer Science and Engineering from the Board of Technical Education during the year 2020-2021

Internal Guide	HOD	Principal
Mrs. Poojitha K B Lecturer Dept. of Computer Science	Smt.Haritha S M Head of the Department Dept. of Computer Science	Mr. K L Ramesh Principal S.J.(Govt.)Polytechnic

ACKNOWLEDGMENT

First, we have to say that, we feel very happy to be part of Sri Jayachamarajendra (Govt.) Polytechnic, Bangalore and Department of Computer Science and Engineering, which has given us an opportunity to acquire technical knowledge that has helped us in all aspect form the past three years.

We thank our Principal **Mr. K L Ramesh** for this sincere help in providing us excellence atmosphere in this institute.

We heartily thank our beloved H.O.D **Smt. Haritha S M** for her encouragement, guidance and suggestion given to us in the course of our project and providing excellent faculties which enhance our knowledge and has given us a better confidence to perform and kind of technical tasks with no difficulties.

We would like to express our deep gratitude and wholehearted thanks **to Mrs. Poojitha K B**, for her valuable guidance, precious suggestions and encouragement in completing the project successfully. We wish to convey our sincere thanks to all the lectures for their support throughout the project.

We are also grateful to all the administrative staff and the technical staff of Computer Science Department who have been helpful throughout our project. Lastly, we would like to thank all our friends who tested our applications in their devices and gave their suggestions and valuable comments.

1. Nanda kumar (102CS18027)
2. Nehan ashin tony (102CS18031)
3. Prajwal P (102CS18033)
4. Sarvesh V (102CS18044)

ABSTRACT

There are many places like Hospitals, Universities, Corporate offices etc.where it is clearly mentioned, “KEEP YOUR MOBILE PHONES SILENT!!” and every one of us will be having a mobile phone and we can’t switch off our phone all the time in such places as we may get some important calls or messages. Many times we may have experienced or come across this situation like mobile ringing in hospitals, universities and in important meetings, and that is an awkward situation if it happens to us. Many times people forget to switch the mobile to the “Silent Mode” which is not feasible every time like in an important meeting, lectures etc.

There may be some of the options like mute ,DND mode, vibrate mode in mobile phones but it should me done manually by oneself and it also does not allow us to edit the profile settings like to vibrate only for important calls etc.

So an Android application will be the best solution for this problem like for automatic profile switching will provide near about completely automated profile switching(silent mode, DND mode etc.). This application will enable the device to switch to the Silent Mode based on the geofence selected by the user.

Table of Contents

INTRODUCTION

AUTO SILENT will enable the device to switch to the Silent Mode in locations like Hospitals, schools, colleges, Universities, offices etc. as per customized by the user. The user just needs to select the geofence coordinates of the locations along with the required radius dimensions that he wants to be in the silent zone. The stored data will be compared via the GPS and the profile will be changed accordingly.

The user can contact them back later without being disturbed. In User- Defined Switching Mode user set location where their device need to switch their audio profile. The application will use GPS Service provided by GPS Satellites for finding locations. In profile switching operation application actually switch the ringer mode of profile

Here user can choose among Silent/DND mode for switching purpose. There is a provision made to neglect the calls while on the silent profile to avoid the disturbance.

By developing this AUTO SILENT application software, it will improve the lifestyle of people. The user's smartphone will be automatically turned into silent mode according to the selected schedules such as day, times or places

1.1 Problem Definition

This new application will provide complete automatic profile switching according to location. This application will enable the device to switch to the Silent Mode in selected locations like Hospitals, Corporate offices, Universities, Well known Educational Complexes, Petrol pumps, Government offices etc.

In profile switching operation application actually switch the ringer mode of profile. There will be some important profile modes as default so that here user can choose silent/DND mode for switching purpose(silent mode, DND mode etc.) as per the user's convenience for the selected location.

1.2 Existing System

Existing system is a manual profile switcher available in all smart phones. Here many people forget to change their phone profile mode where the phone starts ringing in many places such as universities, conferences, hospitals etc . And switching the ringer profile in particular locations is always a burden and maintaining these profiles is a bit complicated for the users.

1.3 Proposed System

The proposed system is better and more efficient than existing system by speeding up the process of switching profiles in particular location. To overcome the drawbacks of the existing system, the proposed system has been evolved by avoiding all this conflicts. This project aims to provide auto audio profile switcher in particular locations. The application provides with the best user interface. The location to perform auto profile switcher can be generated by using this proposed system.

Advantages of proposed System

- It is trouble-free to use.
- It is a relatively fast approach for auto profile switcher.
- Highly reliable.
- Provides best user Interface.

.

1.4 Objectives

AUTO SILENT plans to achieve the following objectives:

- Easy to use
- Automated profile switching
- Accuracy
- Increase usability
- User-friendly

TOOLS AND ENVIRONMENT USED

2.1 Tools/ Technology

▯ Extensible Markup Language (XML)

Extensible Markup Language (**XML)** is a markup language that defines a set of rules for encoding documents in a format that is both human readable and machine readable. The world wide web consortium 's XML 1.0 Specification of 1998 and several other related specifications all of them free open standards define XML.

The design goals of XML emphasize simplicity, generality, and usability across the internet It is a textual data format with strong support via unicode for different human languages. Although the design of XML focuses on documents, the language is widely used for the representation of arbitrary data structures such as those used in web services.

Several schema systems exist to aid in the definition of XML-based languages, while programmers have developed many application programming interfaces (APIs) to aid the processing of XML data

▯ JAVA

JAVA is a high level class based object oriented language that is designed to have as few implementation dependencies as possible. It is a general purpose programming language intended to let programmers *write once, run anywhere* (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any java virtual machines (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to c and c++, but has fewer low level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages.

▯ Android Emulator/Physical device

An Android emulator is a tool that creates virtual Android devices (with software and hardware) on your computer. Note that:

- It is a program (a process that runs on your computer's operating system).
- It works by mimicking the guest device's architecture (more on that in a bit).

In order to better understand what Android emulators are capable of, you should know how they work.

▯ Google MAPS

Google Maps is a web mapping platform and consumer application offered by google. It offers satellite imagery, aerial photography street maps, 360° interactive panoramic views of streets (Street view), real-time traffic conditions, and route planning for traveling by foot, car, air (in beta) and public transportation

▯ Global positioning system (GPS)

The **Global Positioning System** (**GPS**), originally **Navstar GPS**, is a satellite based radio navigation system owned by the united states government and operated by the united states space force It is one of the Global navigation satellite system (GNSS) that provides geolocation and time information to a GPS receiver anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. Obstacles such as mountains and buildings can block the relatively weak GPS signals.

The GPS does not require the user to transmit any data, and it operates independently of any telephonic or Internet reception, though these technologies can enhance the usefulness of the GPS positioning information. The GPS provides critical positioning capabilities to military, civil, and commercial users around the world. The United States government created the system, maintains and controls it, and makes it freely accessible to anyone with a GPS receiver. \

2.2 SOFTWARES USED

▯ Android Studio

Android Studio is the official[7] integrated development environment (IDE)
for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download
on Windows, macOS and Linux based operating systems or as a subscription-based service in 2020.It is a replacement for the Eclipse Android Development Tools (E-ADT) as the primary IDE for native Android application development.

The following features are provided in the current stable version:

- Gradle-based build support
- Android-specific refactoring and quick fixes
- lint tools to catch performance, usability, version compatibility and other problems

- Proguard integration and app-signing capabilities
- Template-based wizards to create common Android designs and components
- A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations
- Support for building android wear apps
- Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine
- Android Virtual Device (Emulator) to run and debug apps in the Android studio.

Google API key

To use the Maps Static API you must have an API key. The API key is a unique identifier that is used to authenticate requests associated with your project for usage and billing purposes.

Depending on your usage, a digital signature may also be required (see other usage limits) The digital signature allows our servers to verify that any site generating requests using your API key is authorized to do so.

To get an API key:

1. Go to the google cloud console.
2. Click the project drop-down and select or create the project for which you want to add an API key.
3. Click the menu button and select **Google Maps Platform > Credentials**.
4. On the **Credentials** page, click **+ Create Credentials > API key**.
   The **API key created** dialog displays the newly created API key.
5. Click **Close**.
   The new API key is listed on the **Credentials** page under **API Keys**.
   (Remember to restrict the API key before using it in production.)

ANALYSIS DOCUMENT

A software requirements specification (SRS) is a description of a software system to be developed. It is modelled after business requirements specification (CONOPS), also known as a stakeholder requirements specification (SRS). The software requirements specification lays out functional and non-functional requirements, and it may include a set of use cases that describe user interactions that the software must provide.

Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers on how the software product should function (in a market-driven project, these roles may be played by the marketing and development divisions). Software requirements specification is rigorous assessment of requirements before the more specific system design stages, and its goal is to reduce later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules. Used appropriately, software requirements specifications can help prevent software project failure.

The software requirements specification document lists sufficient and necessary requirements for the project development. To derive the requirements, the developer needs to have clear and thorough understanding of the products under development. This is achieved through detailed and continuous communications with the project team and customer throughout the software development process.

3.1 User Requirements

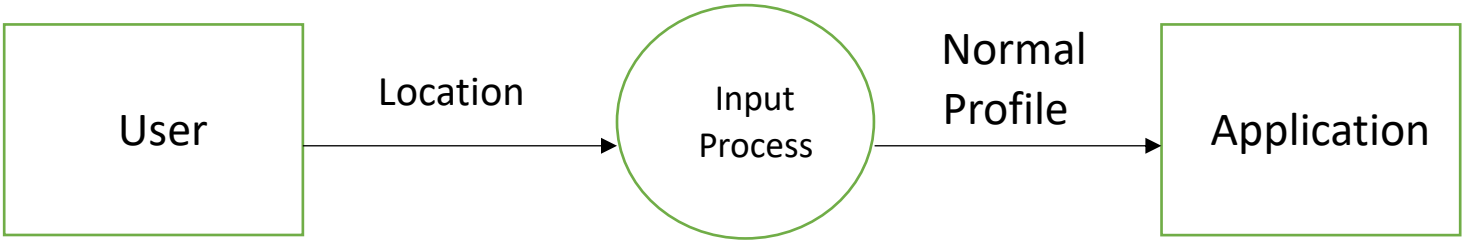
- Need an android smart phone
- The smart phone should be GPS enabled
- Active internet connection is required
- The android operating system version should be above 4.0
- Minimum SDK version is 16

3.4 Data Flow Diagrams

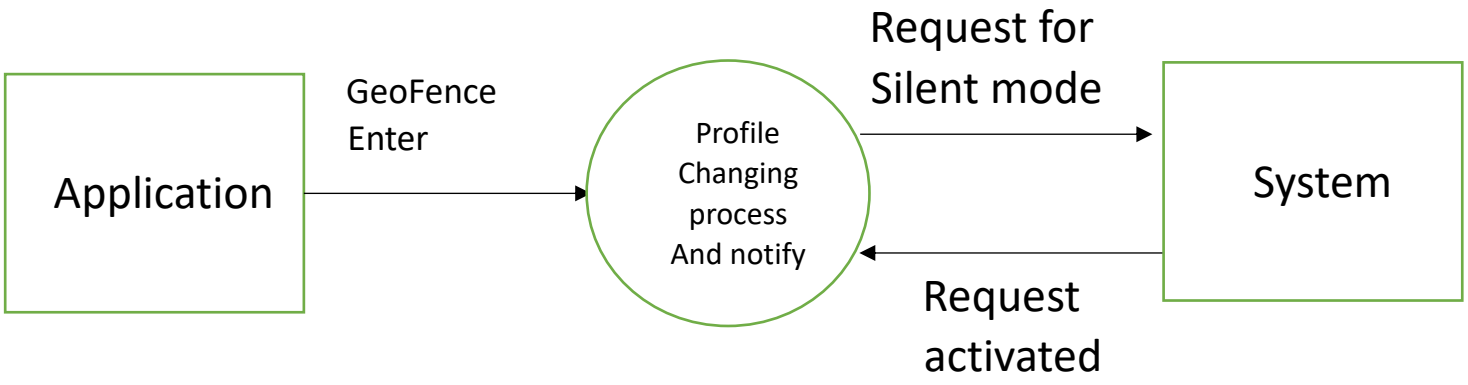
A data flow diagram(DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an over view of the system, which can later be elaborated.

DFD can also be used for the visualization of data processing (Structured design).

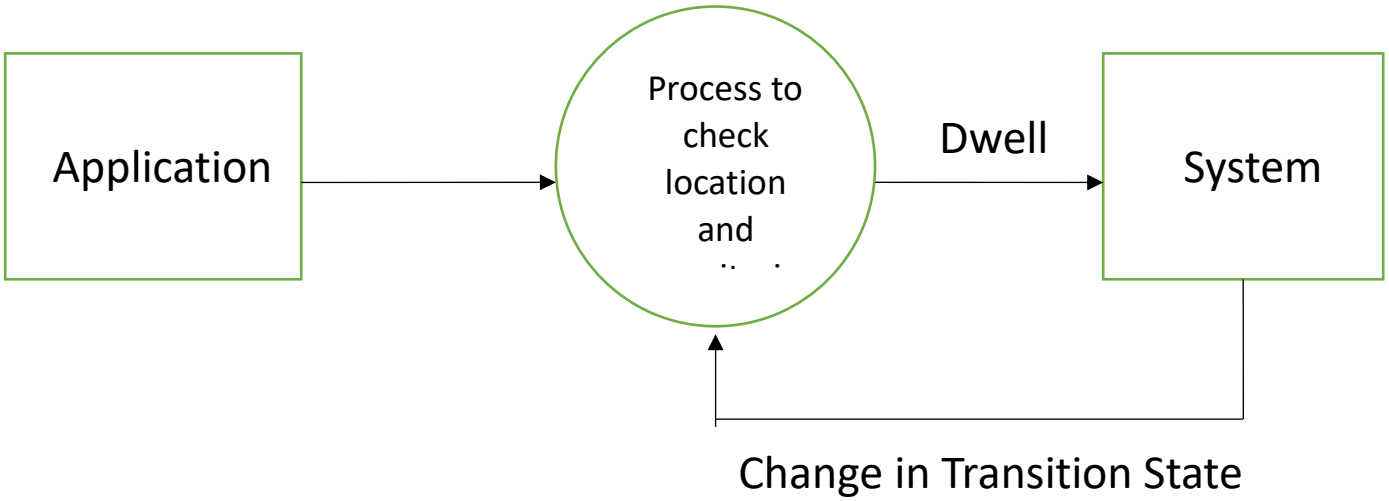
DFD level (0) :- Select location and create geofence



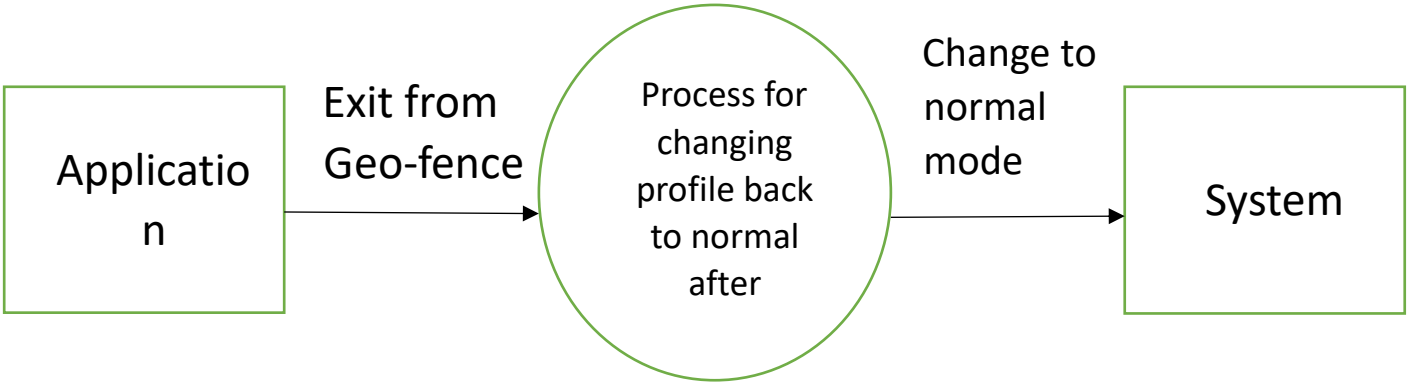
DFD level(1) :- Processing to change to silent profile when entered the geofence



DFD level(2) :- Monitor the location of the device and check for transition

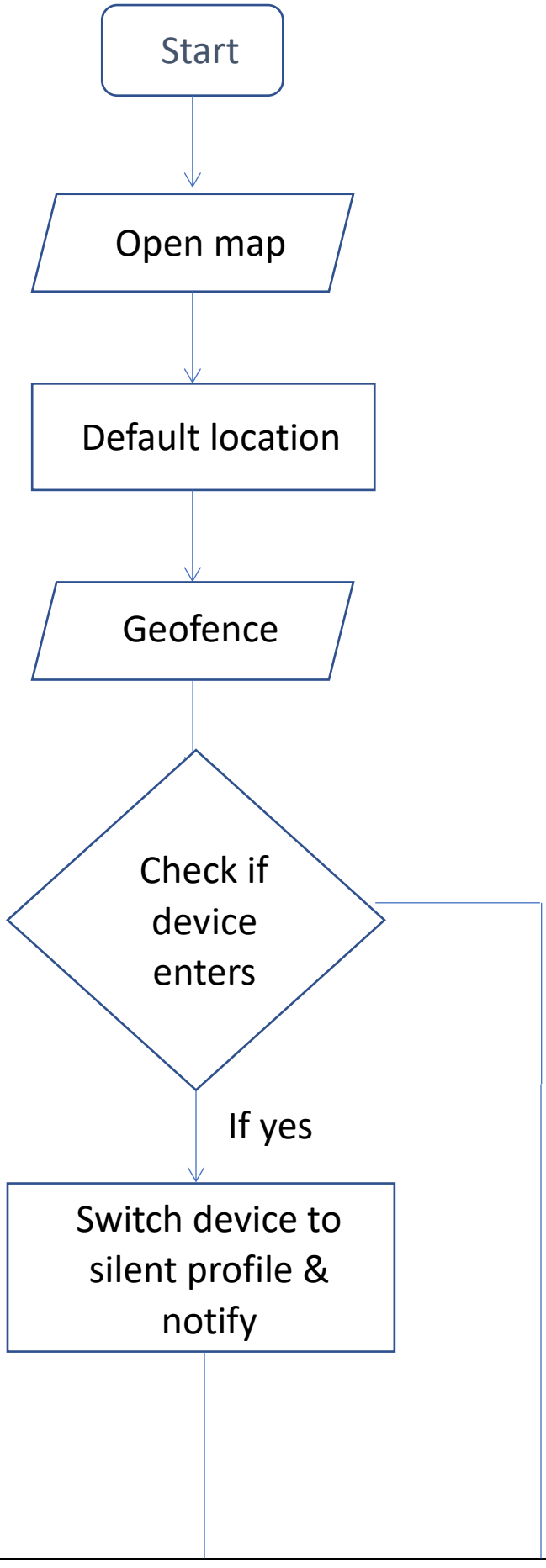


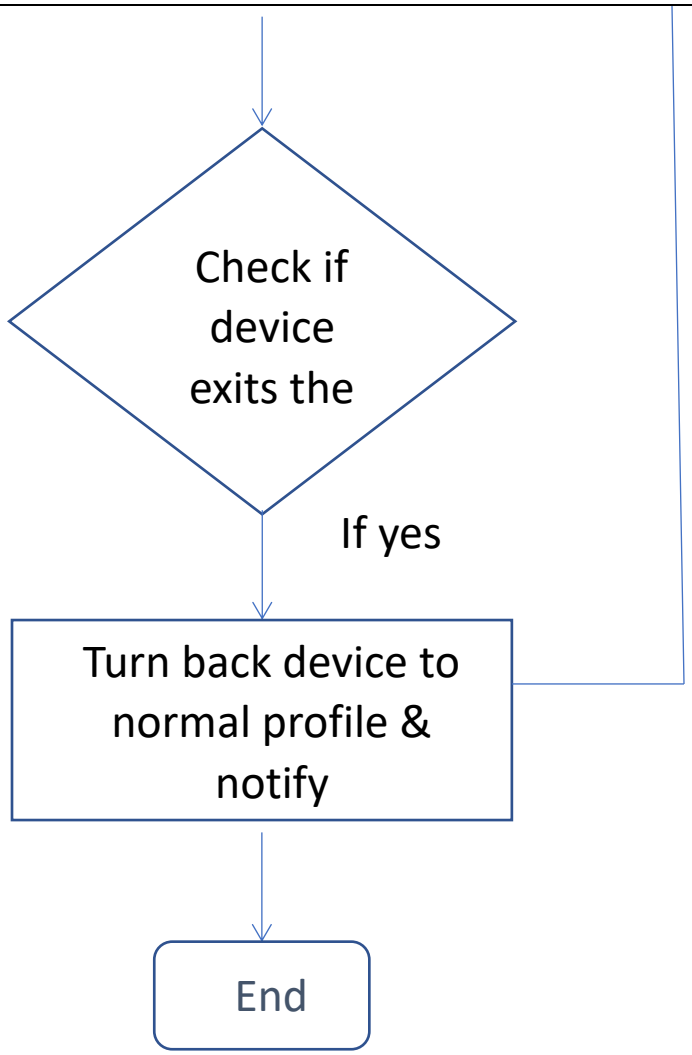
DFD level(3) :- Changing the profile back to normal status after exiting the geofence



3.5 Flow Charts

A flowchart is a type of diagram that represents an algorithm, workflow or process, Showing the steps as boxes of various kinds, and their order by connecting the with arrows. This diagrammatic representation illustrates a solution model to a given model to given problem. Flow charts are used in analysing, designing, documenting or managing a process or program in various field


Start
Open map
Default location
Geofence
Check if device enters
If yes
Switch device to silent profile & notify



DESIGN

4.1 Introduction

Software design is a process by which an agent creates a specification of a software artefact, intended to accomplish goals, using a set of primitive components and subject to constraints. Software design may refer to either "all the activity involved in conceptualizing, framing, implementing, commissioning and ultimately modifying complex systems" or "the activity following requirements specifications and before programming, as a stylized software engineering process".

Software design usually involves problems solving and planning a software solution. This includes both a low-level component and algorithm design and a high-level, architecture design.

4.2 Over View

The word system is possibly the most overused and abused terms in the technical lexicon. System can be defined as the "a set of fact, principles, rules etc. classified and arranged in an

orderly form so as to show a logical plan linking the "various parts" here the system design defines the computer based information system. The primary objective is to identify user requirements and to build a system that satisfies these requirements.

Design is much more creative process than analysis, Design is the first step in the development of any Systems or product. Design can be defined as "the process of applying various techniques and principles for the purpose of detaining, a processes or a system in sufficient detail to permit its physical address".

It involves four major steps:

1. Understand how the system is working now.
2. Finding out what system does now.
3. Understanding what the new system will do.
4. Understanding how the new system will work.

So as to avoid these difficulties, a new system was designed to keep these requirements in mind. Therefore, the manual process operation has been changed into GUI based environment, such that the user can retrieve the records in a user- friendly manner and it is very easy to navigate to the corresponding information.

4.3 User Interface

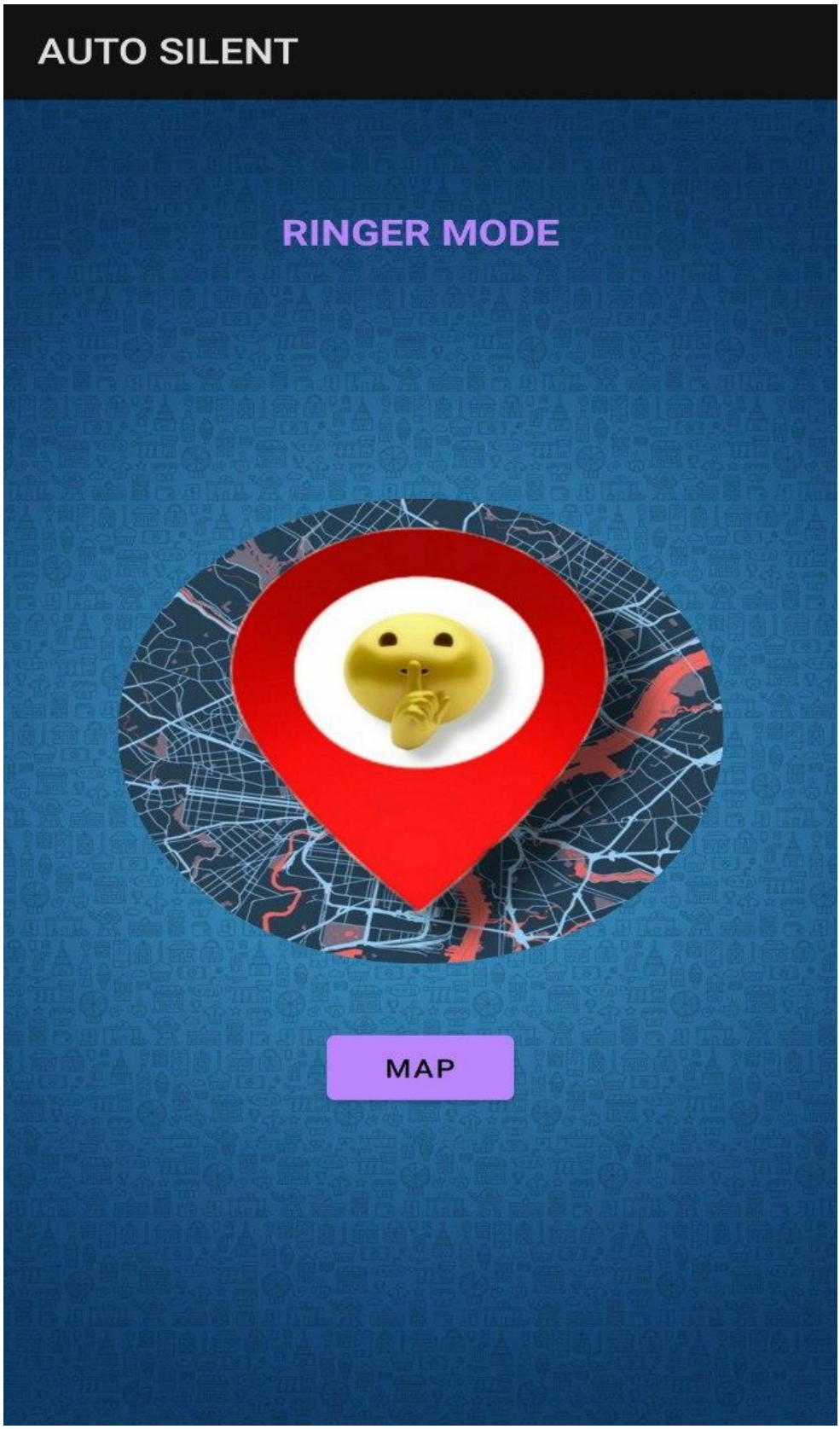
User interface is the front-end application view to which user interacts in order to use the software. User control the software by means of user interface Today, user interface is found at almost every place technology exists, right from computers, mobile phones, cars, music players, airplanes, ships

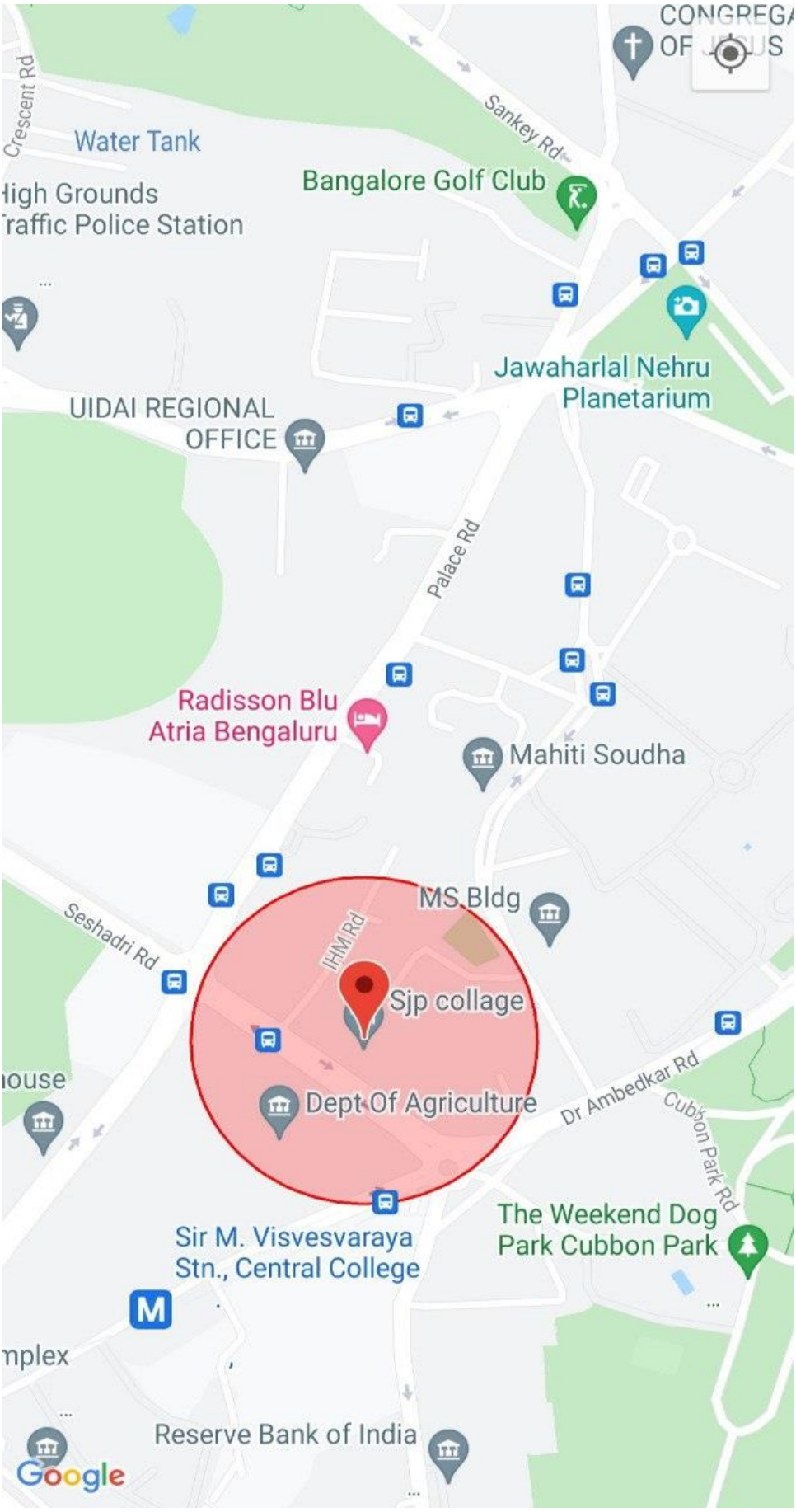
User interface is part of software and is designed such a way that it is expected to provide the user insight of the software. UI provides fundamental platform for human-computer interaction.

The application has the following characteristics:

- Attractive
- Simple to use
- Responsive
- Clear to understand
- Consistent on all interfacing screen

Mobile View:-




Crescent Rd
Water Tank
High Grounds
Traffic Police Station
Sankey Rd
Bangalore Golf Club
Jawaharlal Nehru
Planetarium
UIDAI REGIONAL
OFFICE
Palace Rd
Radisson Blu
Atria Bengaluru
Mahiti Soudha
MS Bldg
Seshadri Rd
IHM Rd
Sjp collage
Dept Of Agriculture
Dr Ambedkar Rd
Cubbon Park Rd
The Weekend Dog
Park Cubbon Park
Sir M. Visvesvaraya
Stn., Central College
Reserve Bank of India
Google

IMPLEMENTAION

5.1 Program Code

FRONT END XML CODE of app:-

```
<!--XML Code for your activity_main.xml-->
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:background="@drawable/background"
    android:layout_height="match_parent"
    tools:context=".MainActivity">



    <!-- Textview to display the heading of the app-->
    <!-- Textview to display the current mode of the Ringer mode-->

    <TextView
        android:id="@+id/idTVHeading"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_marginTop="6dp"
        android:layout_marginBottom="484dp"
        android:gravity="center_horizontal"
        android:text=""
        android:textAlignment="center"
        android:textAllCaps="true"
        android:textColor="@color/purple_200"
        android:textSize="20sp"
        android:textStyle="bold"
```

```
    tools:ignore="HardcodedText" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAlignment="center"
    android:text=""
    android:textAllCaps="true"
    android:textSize="20sp"
    android:layout_marginTop="60dp"
    android:textColor="@color/purple_200"
    android:textStyle="bold"
    android:id="@+id/TVCurrentMode"
    android:gravity="center_horizontal"
    tools:ignore="HardcodedText" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAlignment="center"
    android:text=""
    android:textAllCaps="true"
    android:textSize="20sp"
    android:layout_marginTop="60dp"
    android:textColor="@color/purple_200"
    android:textStyle="bold"
    android:id="@+id/idTVCurrentMode"
    android:gravity="center_horizontal"
    tools:ignore="HardcodedText" />

<ImageButton
    android:id="@+id/idIBSilentMode"
```

```
    android:layout_width="207dp"
    android:layout_height="187dp"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginStart="10dp"
    android:layout_marginLeft="10dp"
    android:layout_marginTop="10dp"
    android:layout_marginEnd="10dp"
    android:layout_marginRight="10dp"
    android:layout_marginBottom="308dp"
    android:layout_weight="1"
    android:background="@drawable/app_logo"
    android:contentDescription="@string/silent_mode"
    app:tint="@color/white"
    tools:ignore="HardcodedText" />

<!-- Button to display map-->

  <Button
    android:id="@+id/map"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginStart="10dp"
    android:layout_marginLeft="10dp"
    android:layout_marginTop="10dp"
    android:layout_marginEnd="10dp"
    android:layout_marginRight="10dp"
    android:layout_marginBottom="176dp"
    android:layout_weight="1"
    android:onClick="openMap"
```

```
        android:text="Map" />

</RelativeLayout>
```

Maps front end

```
<?xml version="1.0" encoding="utf-8"?>
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:map="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MapsActivity" />
```

BACK END JAVA CODE:-

MainActivity

```
package com.example.app2;

import androidx.appcompat.app.AppCompatActivity;

import android.app.NotificationManager;
import android.content.Context;
import android.content.Intent;
import android.media.AudioManager;
import android.os.Build;
import android.os.Bundle;
import android.provider.Settings;
import android.view.View;
```

```
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity{
        void change_mode(int rec){
              switch (rec){
                 case 1:
                  audioManager.setRingerMode(AudioManager.RINGER_MODE_SILENT);
                  break;
                 default:
                    audioManager.setRingerMode(AudioManager.RINGER_MODE_VIBRATE);
                    break;
               }
        }

       public void openMap(View view){
          Intent intentControls = new Intent(this, MapsActivity.class);
          startActivity(intentControls);
       }



       // Textview to display the current ringer mode
       TextView currentStateTV;
       //image buttons to switch ringer mode.
       ImageButton silentIB, vibrateIB, ringtoneIB;
       // object class variable for audio manager class.
       private AudioManager audioManager;
       // current mode to store integer value of ringer mode.
       int currentmode;

      @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        currentStateTV = findViewById(R.id.idTVCurrentMode);
        silentIB = findViewById(R.id.idIBSilentMode);
     //   vibrateIB = findViewById(R.id.idIBVibrateMode);
          audioManager = (AudioManager) getSystemService(Context.AUDIO_SERVICE);
        //current mode will store current mode of ringer of users device..
       currentmode = audioManager.getRingerMode();

        switch (currentmode) {
            case AudioManager.RINGER_MODE_NORMAL:
                currentStateTV.setText("Ringer Mode");
                break;
            case AudioManager.RINGER_MODE_SILENT:
                currentStateTV.setText("Silent Mode");
                break;
            case AudioManager.RINGER_MODE_VIBRATE:
                currentStateTV.setText("Vibrate Mode");
                break;
            default:
                currentStateTV.setText("Fail to get mode");
                break;
        }

         silentIB.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                NotificationManager notificationManager =
                        (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
                // the below code is to check the permission that the access notification policy settings from
users device..
```

```
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M
                    && !notificationManager.isNotificationPolicyAccessGranted()) {

                Intent intent = new
Intent(Settings.ACTION_NOTIFICATION_POLICY_ACCESS_SETTINGS);

                startActivity(intent);
            }
            // set ringer mode here will sets your ringer mode to silent mode

           // currentStateTV.setText("Silent Mode Activated..");
        }
    });
  }

}
```

MapsActivity

```
package com.example.app2;

import androidx.annotation.NonNull;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.fragment.app.FragmentActivity;

import android.Manifest;
import android.app.PendingIntent;
import android.content.pm.PackageManager;
import android.graphics.Color;
import android.os.Build;
import android.os.Bundle;
```

```
import android.util.Log;
import android.widget.Toast;

import com.google.android.gms.location.Geofence;
import com.google.android.gms.location.GeofencingClient;
import com.google.android.gms.location.GeofencingRequest;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.CircleOptions;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;

public class MapsActivity extends FragmentActivity implements OnMapReadyCallback,
GoogleMap.OnMapLongClickListener {

    private static final String TAG = "MapsActivity";

    private GoogleMap mMap;
    private GeofencingClient geofencingClient;
    private GeofenceHelper geofenceHelper;

    private float GEOFENCE_RADIUS = 200;
    private String GEOFENCE_ID = "Geofence";

    private final int FINE_LOCATION_ACCESS_REQUEST_CODE = 10001;
    private int BACKGROUND_LOCATION_ACCESS_REQUEST_CODE = 10002;
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_maps);
    // Obtain the SupportMapFragment and get notified when the map is ready to be used.
    SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
            .findFragmentById(R.id.map);
    mapFragment.getMapAsync(this);

    geofencingClient = LocationServices.getGeofencingClient(this);
    geofenceHelper = new GeofenceHelper(this);

}

/**
 * Manipulates the map once available.
 * This callback is triggered when the map is ready to be used.
 * This is where we can add markers or lines, add listeners or move the camera. In this case,
 * we just add a marker near Sydney, Australia.
 * If Google Play services is not installed on the device, the user will be prompted to install
 * it inside the SupportMapFragment. This method will only be triggered once the user has
 * installed Google Play services and returned to the app.
 */
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;

    // Add a marker in Sydney and move the camera
    LatLng sjp = new LatLng(12.9804279, 77.5865788);
    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(sjp, 16));

    enableUserLocation();
```

```
        mMap.setOnMapLongClickListener(this);
    }

    private void enableUserLocation() {
        if (ContextCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION)
== PackageManager.PERMISSION_GRANTED) {
            mMap.setMyLocationEnabled(true);
        } else {
            //Ask for permission
            if (ActivityCompat.shouldShowRequestPermissionRationale(this,
Manifest.permission.ACCESS_FINE_LOCATION)) {
                //We need to show user a dialog for displaying why the permission is needed and then ask for
the permission...
                ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION},
FINE_LOCATION_ACCESS_REQUEST_CODE);
            } else {
                ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION},
FINE_LOCATION_ACCESS_REQUEST_CODE);

            }
        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
@NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        if (requestCode == FINE_LOCATION_ACCESS_REQUEST_CODE) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
```

```
            //We have the permission
            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;
            }
            mMap.setMyLocationEnabled(true);
        } else {
            //We do not have the permission..
        }
    }

    if (requestCode == BACKGROUND_LOCATION_ACCESS_REQUEST_CODE) {
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            //We have the permission
            Toast.makeText(this, "You can add geofences...", Toast.LENGTH_SHORT).show();
        } else {
            //We do not have the permission..
            Toast.makeText(this, "Background location access is neccessary for geofences to trigger...",
Toast.LENGTH_SHORT).show();
        }
    }
}
```

```
    @Override
    public void onMapLongClick(LatLng latLng) {
        if (Build.VERSION.SDK_INT >= 29) {
            //We need background permission
            if (ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_BACKGROUND_LOCATION) ==
PackageManager.PERMISSION_GRANTED) {
                handleMapLongClick(latLng);
            } else {
                if (ActivityCompat.shouldShowRequestPermissionRationale(this,
Manifest.permission.ACCESS_BACKGROUND_LOCATION)) {
                    //We show a dialog and ask for permission
                    ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.ACCESS_BACKGROUND_LOCATION},
BACKGROUND_LOCATION_ACCESS_REQUEST_CODE);
                } else {
                    ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.ACCESS_BACKGROUND_LOCATION},
BACKGROUND_LOCATION_ACCESS_REQUEST_CODE);
                }
            }

        } else {
            handleMapLongClick(latLng);
        }

    }

    private void handleMapLongClick(LatLng latLng) {
        mMap.clear();
        addMarker(latLng);
        addCircle(latLng, GEOFENCE_RADIUS);
        addGeofence(latLng, GEOFENCE_RADIUS);
```

```
    }

    private void addGeofence(LatLng latLng, float radius) {
        Geofence geofence = geofenceHelper.getGeofence(GEOFENCE_ID, latLng, radius,
Geofence.GEOFENCE_TRANSITION_ENTER | Geofence.GEOFENCE_TRANSITION_EXIT);
        GeofencingRequest geofencingRequest = geofenceHelper.getGeofencingRequest(geofence);
        PendingIntent pendingIntent = geofenceHelper.getPendingIntent();

        if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_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;
        }
        geofencingClient.addGeofences(geofencingRequest, pendingIntent)
                .addOnSuccessListener(new OnSuccessListener<Void>() {
                    @Override
                    public void onSuccess(Void aVoid) {
                        Log.d(TAG, "onSuccess: Geofence Added...");
                    }
                })
                .addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {
                        String errorMessage = geofenceHelper.getErrorString(e);
                        Log.d(TAG, "onFailure: " + errorMessage);
                    }
                });
```

```
    }

    private void addMarker(LatLng latLng) {
        MarkerOptions markerOptions = new MarkerOptions().position(latLng);
        mMap.addMarker(markerOptions);
    }

    private void addCircle(LatLng latLng, float radius) {
        CircleOptions circleOptions = new CircleOptions();
        circleOptions.center(latLng);
        circleOptions.radius(radius);
        circleOptions.strokeColor(Color.argb(255, 255, 0,0));
        circleOptions.fillColor(Color.argb(64, 255, 0,0));
        circleOptions.strokeWidth(4);
        mMap.addCircle(circleOptions);
    }
}
```

GeofenceHelper

```
package com.example.app2;

import android.app.PendingIntent;
import android.content.Context;
import android.content.ContextWrapper;
import android.content.Intent;

import com.google.android.gms.common.api.ApiException;
import com.google.android.gms.location.Geofence;
import com.google.android.gms.location.GeofenceStatusCodes;
import com.google.android.gms.location.GeofencingRequest;
import com.google.android.gms.maps.model.LatLng;
```

```
public class GeofenceHelper extends ContextWrapper {

    private static final String TAG = "GeofenceHelper";
    PendingIntent pendingIntent;

    public GeofenceHelper(Context base) { super(base); }

    public GeofencingRequest getGeofencingRequest(Geofence geofence) {
        return new GeofencingRequest.Builder()
                .addGeofence(geofence)
                .setInitialTrigger(GeofencingRequest.INITIAL_TRIGGER_ENTER)
                .build();
    }

    public Geofence getGeofence(String ID, LatLng latLng, float radius, int transitionTypes) {
        return new Geofence.Builder()
                .setCircularRegion(latLng.latitude, latLng.longitude, radius)
                .setRequestId(ID)
                .setTransitionTypes(transitionTypes)
                .setLoiteringDelay(5000)
                .setExpirationDuration(Geofence.NEVER_EXPIRE)
                .build();
    }

    public PendingIntent getPendingIntent() {
        if (pendingIntent != null) {
            return pendingIntent;
        }
        Intent intent = new Intent(this, NewReciever.class);
        pendingIntent = PendingIntent.getBroadcast(this, 2607, intent,
PendingIntent.FLAG_UPDATE_CURRENT);
```

```
        return pendingIntent;
    }
    public String getErrorString(Exception e) {
        if (e instanceof ApiException) {
            ApiException apiException = (ApiException) e;
            switch (apiException.getStatusCode()) {
                case GeofenceStatusCodes
                        .GEOFENCE_NOT_AVAILABLE:
                    return "GEOFENCE_NOT_AVAILABLE";
                case GeofenceStatusCodes
                        .GEOFENCE_TOO_MANY_GEOFENCES:
                    return "GEOFENCE_TOO_MANY_GEOFENCES";
                case GeofenceStatusCodes
                        .GEOFENCE_TOO_MANY_PENDING_INTENTS:
                    return "GEOFENCE_TOO_MANY_PENDING_INTENTS";
            }
        }
        return e.getLocalizedMessage();
    }
}
```

NewReciever

```
package com.example.app2;

import android.annotation.SuppressLint;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.media.AudioManager;
import android.util.Log;
```

```
import android.widget.Toast;

import com.google.android.gms.location.Geofence;
import com.google.android.gms.location.GeofencingEvent;

import java.util.List;

public class NewReciever extends BroadcastReceiver {

    private static final String TAG = "NewReceiver";
    public static int check;

    @SuppressLint("LongLogTag")
    @Override
    public void onReceive(Context context, Intent intent) {
        // TODO: This method is called when the BroadcastReceiver is receiving
        // an Intent broadcast.

        NotificationHelper notificationHelper = new NotificationHelper(context);

        GeofencingEvent geofencingEvent = GeofencingEvent.fromIntent(intent);
        AudioManager audioManager = (AudioManager)
context.getSystemService(Context.AUDIO_SERVICE);

        if (geofencingEvent.hasError()) {
            Log.d(TAG, "onReceive: Error receiving geofence event...");
            return;
        }

        List<Geofence> geofenceList = geofencingEvent.getTriggeringGeofences();
        for (Geofence geofence : geofenceList) {
            Log.d(TAG, "onReceive: " + geofence.getRequestId());
```

```
        }
//      Location location = geofencingEvent.getTriggeringLocation();
        int transitionType = geofencingEvent.getGeofenceTransition();
       if (transitionType == Geofence.GEOFENCE_TRANSITION_ENTER)
        {
            audioManager.setRingerMode(AudioManager.RINGER_MODE_SILENT);
            notificationHelper.sendHighPriorityNotification("GEOFENCE_TRANSITION_ENTER", "",
MapsActivity.class);
            Toast.makeText(context,"now enterd the geofence and in silent
mode",Toast.LENGTH_SHORT).show();
        }
        if (transitionType == Geofence.GEOFENCE_TRANSITION_EXIT)
        {
            audioManager.setRingerMode(AudioManager.RINGER_MODE_NORMAL);
            notificationHelper.sendHighPriorityNotification("GEOFENCE_TRANSITION_EXIT", "",
MapsActivity.class);
            Toast.makeText(context,"now exited the geofence and in normal
mode",Toast.LENGTH_SHORT).show();
        }
        else
        {
            //Log Error
            Log.e(TAG, String.format("Unknoen transition : %d", transitionType));
            //No need to do anything else
            return;
        }

    }
}
```

NotificationHelper

```
package com.example.app2;

import android.app.Notification;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.ContextWrapper;
import android.content.Intent;
import android.graphics.Color;
import android.os.Build;

import androidx.annotation.RequiresApi;
import androidx.core.app.NotificationCompat;
import androidx.core.app.NotificationManagerCompat;

import java.util.Random;

public class NotificationHelper extends ContextWrapper {

    private static final String TAG = "NotificationHelper";

    public NotificationHelper(Context base) {
        super(base);
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            createChannels();
        }
    }

    private String CHANNEL_NAME = "High priority channel";
```

```
    private String CHANNEL_ID = "com.example.notifications" + CHANNEL_NAME;

    @RequiresApi(api = Build.VERSION_CODES.O)
    private void createChannels() {
        NotificationChannel notificationChannel = new NotificationChannel(CHANNEL_ID,
CHANNEL_NAME, NotificationManager.IMPORTANCE_HIGH);
        notificationChannel.enableLights(true);
        notificationChannel.enableVibration(true);
        notificationChannel.setDescription("this is the description of the channel.");
        notificationChannel.setLightColor(Color.RED);
        notificationChannel.setLockscreenVisibility(Notification.VISIBILITY_PUBLIC);
        NotificationManager manager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
        manager.createNotificationChannel(notificationChannel);
    }

    public void sendHighPriorityNotification(String title, String body, Class activityName) {

        Intent intent = new Intent(this, activityName);
        PendingIntent pendingIntent = PendingIntent.getActivity(this, 267, intent,
PendingIntent.FLAG_UPDATE_CURRENT);

        Notification notification = new NotificationCompat.Builder(this, CHANNEL_ID)
//                .setContentTitle(title)
//                .setContentText(body)
                .setSmallIcon(R.drawable.ic_launcher_background)
                .setPriority(NotificationCompat.PRIORITY_HIGH)
                .setStyle(new
NotificationCompat.BigTextStyle().setSummaryText("summary").setBigContentTitle(title).bigText(bod
y))
                .setContentIntent(pendingIntent)
                .setAutoCancel(true)
                .build();
```

```
        NotificationManagerCompat.from(this).notify(new Random().nextInt(), notification);
    }

}
```

AndroidManifest file

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.app2">

    <!--
        The ACCESS_COARSE/FINE_LOCATION permissions are not required to use
        Google Maps Android API v2, but you must specify either coarse or fine
        location permissions for the "MyLocation" functionality.
    -->
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_BACKGROUND_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NOTIFICATION_POLICY" />
    <!-- <uses-library android:name=/> -->

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.App2">
```

```
        <receiver
            android:name=".NewReciever"
            android:enabled="true"
            android:exported="true"></receiver>
        <!--
             The API key for Google Maps-based APIs is defined as a string resource.
             (See the file "res/values/google_maps_api.xml").
             Note that the API key is linked to the encryption key used to sign the APK.
             You need a different API key for each encryption key, including the release key that is used to
             sign the APK for publishing.
             You can define the keys for the debug and release targets in src/debug/ and src/release/.
        -->
        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="@string/google_maps_key" />

        <activity
            android:name=".MapsActivity"
            android:label="@string/title_activity_maps" />
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

GRADLE SCRIPTS

Buildgradle

```
// Top-level build file where you can add configuration options common to all sub-projects/modules.
buildscript {
    repositories {
        google()
        jcenter()
    }
    dependencies {
        classpath "com.android.tools.build:gradle:4.1.2"

        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
}

allprojects {
    repositories {
        google()
        jcenter()
    }
}

task clean(type: Delete) {
    delete rootProject.buildDir
}
```

buildgradle module

```
plugins {
    id 'com.android.application'
}

android {
    compileSdkVersion 30
    buildToolsVersion "30.0.3"

    defaultConfig {
        applicationId "com.example.app2"
        minSdkVersion 16
        targetSdkVersion 30
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
}
```

```
dependencies {

    implementation 'androidx.appcompat:appcompat:1.3.1'
    implementation 'com.google.android.material:material:1.4.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.0'
    implementation 'com.google.android.gms:play-services-maps:17.0.1'
    implementation 'com.google.android.gms:play-services-location:18.0.0'
    testImplementation 'junit:junit:4.+'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
}
```

Gradleproperties

```
## For more details on how to configure your build environment visit
# http://www.gradle.org/docs/current/userguide/build_environment.html
#
# Specifies the JVM arguments used for the daemon process.
# The setting is particularly useful for tweaking memory settings.
# Default value: -Xmx1024m -XX:MaxPermSize=256m
# org.gradle.jvmargs=-Xmx2048m -XX:MaxPermSize=512m -XX:+HeapDumpOnOutOfMemoryError
-Dfile.encoding=UTF-8
#
# When configured, Gradle will run in incubating parallel mode.
# This option should only be used with decoupled projects. More details, visit
# http://www.gradle.org/docs/current/userguide/multi_project_builds.html#sec:decoupled_projects
# org.gradle.parallel=true
#Thu Sep 02 18:52:50 IST 2021
systemProp.http.proxyHost=
systemProp.http.proxyPort=80
systemProp.https.proxyHost=
```

systemProp.https.proxyPort=80

Gradlewrapperproperties

#Sat Sep 04 16:25:41 IST 2021
distributionBase=GRADLE_USER_HOME
distributionPath=wrapper/dists
zipStoreBase=GRADLE_USER_HOME
zipStorePath=wrapper/dists
distributionUrl=https\://services.gradle.org/distributions/gradle-6.5-bin.zip

proguard-rules.pro

```
# Add project specific ProGuard rules here.
# You can control the set of applied configuration files using the
# proguardFiles setting in build.gradle.
# For more details, see
#   http://developer.android.com/guide/developing/tools/proguard.html

# If your project uses WebView with JS, uncomment the following
# and specify the fully qualified class name to the JavaScript interface
# class:
#-keepclassmembers class fqcn.of.javascript.interface.for.webview {
#   public *;
#}

# Uncomment this to preserve the line number information for
# debugging stack traces.
#-keepattributes SourceFile,LineNumberTable

# If you keep the line number information, uncomment this to
# hide the original source file name.
```

#-renamesourcefileattribute SourceFile

gradle.properties Project properties

# Project-wide Gradle settings.
# IDE (e.g. Android Studio) users:
# Gradle settings configured through the IDE *will override*
# any settings specified in this file.
# For more details on how to configure your build environment visit
# http://www.gradle.org/docs/current/userguide/build_environment.html
# Specifies the JVM arguments used for the daemon process.
# The setting is particularly useful for tweaking memory settings.
org.gradle.jvmargs=-Xmx2048m -Dfile.encoding=UTF-8
# When configured, Gradle will run in incubating parallel mode.
# This option should only be used with decoupled projects. More details, visit
# http://www.gradle.org/docs/current/userguide/multi_project_builds.html#sec:decoupled_projects
# org.gradle.parallel=true
# AndroidX package structure to make it clearer which packages are bundled with the
# Android operating system, and which are packaged with your app"s APK
# https://developer.android.com/topic/libraries/support-library/androidx-rn
android.useAndroidX=true
# Automatically convert third-party libraries to use AndroidX
android.enableJetifier=true

settings.gradle

include ':app'
rootProject.name = "App2"
local properties
This file is automatically generated by Android Studio.
Do not modify this file -- YOUR CHANGES WILL BE ERASED!
#

```
# This file should *NOT* be checked into Version Control Systems,
# as it contains information specific to your local configuration.
#
# Location of the SDK. This is only used by Gradle.
# For customization when using a Version Control System, please read the
# header note.
sdk.dir=C\:\\Users\\nanda\\AppData\\Local\\Android\\Sdk
```

Additional codes:-

ExampleInstrumentedTest

```
package com.example.app2;

import android.content.Context;

import androidx.test.platform.app.InstrumentationRegistry;
import androidx.test.ext.junit.runners.AndroidJUnit4;

import org.junit.Test;
import org.junit.runner.RunWith;

import static org.junit.Assert.*;

/**
 * Instrumented test, which will execute on an Android device.
 *
 * @see <a href="http://d.android.com/tools/testing">Testing documentation</a>
 */
@RunWith(AndroidJUnit4.class)
public class ExampleInstrumentedTest {
    @Test
    public void useAppContext() {
```

```
        // Context of the app under test.
        Context appContext = InstrumentationRegistry.getInstrumentation().getTargetContext();
        assertEquals("com.example.app2", appContext.getPackageName());
    }
}
```

ExampleUnitTest

```
package com.example.app2;

import org.junit.Test;
import static org.junit.Assert.*;

/**
 * Example local unit test, which will execute on the development machine (host).
 *
 * @see <a href="http://d.android.com/tools/testing">Testing documentation</a>
 */
public class ExampleUnitTest {
    @Test
    public void addition_isCorrect() {
        assertEquals(4, 2 + 2);
    }
}
```

BuildConfig

```
/**
 * Automatically generated file. DO NOT MODIFY
 */
package com.example.app2;
```

```
public final class BuildConfig {
  public static final boolean DEBUG = Boolean.parseBoolean("true");
  public static final String APPLICATION_ID = "com.example.app2";
  public static final String BUILD_TYPE = "debug";
  public static final int VERSION_CODE = 1;
  public static final String VERSION_NAME = "1.0";
}
```

ic_launcher_background - xml code

```
<?xml version="1.0" encoding="utf-8"?>
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="108dp"
    android:height="108dp"
    android:viewportWidth="108"
    android:viewportHeight="108">
    <path
        android:fillColor="#3DDC84"
        android:pathData="M0,0h108v108h-108z" />
    <path
        android:fillColor="#00000000"
        android:pathData="M9,0L9,108"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M19,0L19,108"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M29,0L29,108"
```

```
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M39,0L39,108"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M49,0L49,108"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M59,0L59,108"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M69,0L69,108"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M79,0L79,108"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M89,0L89,108"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
```

```
<path
    android:fillColor="#00000000"
    android:pathData="M99,0L99,108"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M0,9L108,9"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M0,19L108,19"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M0,29L108,29"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M0,39L108,39"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M0,49L108,49"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
```

```
    android:pathData="M0,59L108,59"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M0,69L108,69"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M0,79L108,79"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M0,89L108,89"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M0,99L108,99"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M19,29L89,29"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M19,39L89,39"
    android:strokeWidth="0.8"
```

```
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M19,49L89,49"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M19,59L89,59"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M19,69L89,69"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M19,79L89,79"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M29,19L29,89"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M39,19L39,89"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
```

```
        android:fillColor="#00000000"
        android:pathData="M49,19L49,89"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M59,19L59,89"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M69,19L69,89"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M79,19L79,89"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
</vector>
```

ic_launcher_foreground - xml code

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:aapt="http://schemas.android.com/aapt"
    android:width="108dp"
    android:height="108dp"
    android:viewportWidth="108"
    android:viewportHeight="108">
    <path android:pathData="M31,63.928c0,0 6.4,-11 12.1,-13.1c7.2,-2.6 26,-1.4 26,-
1.4l38.1,38.1L107,108.928l-32,-1L31,63.928z">
        <aapt:attr name="android:fillColor">
```

```
        <gradient
            android:endX="85.84757"
            android:endY="92.4963"
            android:startX="42.9492"
            android:startY="49.59793"
            android:type="linear">
            <item
                android:color="#44000000"
                android:offset="0.0" />
            <item
                android:color="#00000000"
                android:offset="1.0" />
        </gradient>
    </aapt:attr>
</path>
<path
    android:fillColor="#FFFFFF"
    android:fillType="nonZero"
    android:pathData="M65.3,45.828l3.8,-6.6c0.2,-0.4 0.1,-0.9 -0.3,-1.1c-0.4,-0.2 -0.9,-0.1 -1.1,0.3l-3.9,6.7c-6.3,-2.8 -13.4,-2.8 -19.7,0l-3.9,-6.7c-0.2,-0.4 -0.7,-0.5 -1.1,-0.3C38.8,38.328 38.7,38.828 38.9,39.228l3.8,6.6C36.2,49.428 31.7,56.028 31,63.928h46C76.3,56.028 71.8,49.428 65.3,45.828zM43.4,57.328c-0.8,0 -1.5,-0.5 -1.8,-1.2c-0.3,-0.7 -0.1,-1.5 0.4,-2.1c0.5,-0.5 1.4,-0.7 2.1,-0.4c0.7,0.3 1.2,1 1.2,1.8C45.3,56.528 44.5,57.328 43.4,57.328L43.4,57.328zM64.6,57.328c-0.8,0 -1.5,-0.5 -1.8,-1.2s-0.1,-1.5 0.4,-2.1c0.5,-0.5 1.4,-0.7 2.1,-0.4c0.7,0.3 1.2,1 1.2,1.8C66.5,56.528 65.6,57.328 64.6,57.328L64.6,57.328z"
    android:strokeWidth="1"
    android:strokeColor="#00000000" />
</vector>
```

ic_launcher -xml code

```
<?xml version="1.0" encoding="utf-8"?>
<adaptive-icon xmlns:android="http://schemas.android.com/apk/res/android">
```

```
    <background android:drawable="@color/ic_launcher_background"/>
    <foreground android:drawable="@mipmap/ic_launcher_foreground"/>
</adaptive-icon>
```

ic_launcher_round - xml code

```
<?xml version="1.0" encoding="utf-8"?>
<adaptive-icon xmlns:android="http://schemas.android.com/apk/res/android">
    <background android:drawable="@color/ic_launcher_background"/>
    <foreground android:drawable="@mipmap/ic_launcher_foreground"/>
</adaptive-icon>
```

VALUES:-

colors - xml code

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
</resources>
```

google_maps_api - xml code

```
<resources>
    <!--
    TODO: Before you run your application, you need a Google Maps API key.
```

To get one, follow this link, follow the directions and press "Create" at the end:

https://console.developers.google.com/flows/enableapi?apiid=maps_android_backend&keyType=CLIENT_SIDE_ANDROID&r=CF:A8:7B:2B:35:40:4D:92:B3:AF:01:74:9F:DF:DE:8E:AA:4E:0B:61%3Bcom.example.app2

You can also add your credentials to an existing key, using these values:

Package name:
com.example.app2

SHA-1 certificate fingerprint:
CF:A8:7B:2B:35:40:4D:92:B3:AF:01:74:9F:DF:DE:8E:AA:4E:0B:61

Alternatively, follow the directions here:
https://developers.google.com/maps/documentation/android/start#get-key

Once you have your key (it starts with "AIza"), replace the "google_maps_key"
string in this file.
-->
<string name="google_maps_key" templateMergeStrategy="preserve" translatable="false">AIzaSyBrbdnzGADow5gcjRyTJXsQOF7VXrwRqAM</string>
</resources>

ic_launcher_background - xml code

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="ic_launcher_background">#113B6F</color>
</resources>
```

strings - xml code

```
<resources>
    <string name="app_name">AUTO SILENT</string>
    <string name="silent_mode">silent_mode</string>
    <string name="current_mode">current_mode</string>
    <string name="title_activity_maps">Map</string>
</resources>
```

THEMES:-

theme1 - xml code

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Theme.App2" parent="Theme.MaterialComponents.DayNight.DarkActionBar">
        <!-- Primary brand color. -->
        <item name="colorPrimary">@color/purple_500</item>
        <item name="colorPrimaryVariant">@color/purple_700</item>
        <item name="colorOnPrimary">@color/white</item>
        <!-- Secondary brand color. -->
        <item name="colorSecondary">@color/teal_200</item>
        <item name="colorSecondaryVariant">@color/teal_700</item>
        <item name="colorOnSecondary">@color/black</item>
        <!-- Status bar color. -->
        <item name="android:statusBarColor" tools:targetApi="l">?attr/colorPrimaryVariant</item>
        <!-- Customize your theme here. -->
    </style>
</resources>
```

theme2 - xml code

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Theme.App2" parent="Theme.MaterialComponents.DayNight.DarkActionBar">
        <!-- Primary brand color. -->
        <item name="colorPrimary">@color/purple_200</item>
        <item name="colorPrimaryVariant">@color/purple_700</item>
        <item name="colorOnPrimary">@color/black</item>
        <!-- Secondary brand color. -->
        <item name="colorSecondary">@color/teal_200</item>
        <item name="colorSecondaryVariant">@color/teal_200</item>
        <item name="colorOnSecondary">@color/black</item>
        <!-- Status bar color. -->
        <item name="android:statusBarColor" tools:targetApi="l">?attr/colorPrimaryVariant</item>
        <!-- Customize your theme here. -->
    </style>
</resources>
```

TESTING

6.1 Introduction

Testing is the process used to help identify the correctness, completeness, security, and quality of developed computer software. A technical investigation, performed on behalf of stakeholders, that is intended to reveal quality-related information about the product with respect to the context in which it is intended to operate.

White-box and black-box testing

White box and black box testing are terms used to describe the point of view a test engineer takes when designing test cases. Black box being an external view of the test object and white box being an internal view. Software testing is partly intuitive. but largely systematic.

Good testing involves much more than just running the program a few times to see whether it works. Thorough analysis of the program under test, backed by a broad knowledge of testing techniques and tools are prerequisites to systematic testing.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test.

System testing is based on process descriptions and flows, emphasizing pre driven process links and integration points.

Performance Test

The Performance test ensures that the output be produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

Integration Testing

The phase of software testing in which individual software modules are combined and tested as a group. The purpose of integration testing is to verify functional, performance and reliability requirements placed on major design items. These "design items", i.e. assemblages (or groups of units), are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and interprocess communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test that all components within assemblages interact correctly.

6.2 Software Testing Cycle

Although testing varies between organizations, there is a cycle to testing:

1. Requirements Analysis: Testing should begin in the requirements phase of the software development life cycle.

2. During the design phase, testers work with developers in determining what aspects of a design are testable and under what parameter those tests work.
3. Test Planning: Test Strategy, Test Plan(s), Test Bed creation.
4. Test Development: Test Procedures, Test Scenarios, Test Cases, and Test Scripts to use in testing software.
5. Test Execution: Testers execute the software based on the plans and tests and report any errors found to the development team.
6. Test Reporting: Once testing is completed, testers generate metrics and make i final reports on their test effort and whether or not the software tested is ready for release.
7. Retesting the defects

Not all errors or defects reported must be fixed by a software development team. Some may be caused by errors in configuring the test software to match the development or production environment. Some defects can be handled by a workaround in the production environment. Others might be deferred to future releases of the software, or the deficiency might be accepted by the business user. There are yet other defects that may be rejected by the development team (of course, with due reason) if they deem it inappropriate to be called a defect.

6.3 Testing fundamentals

Software testing is an important element of software quality assurance and represents the ultimate review of specification, design and coding. The increasing visibility of software as a system element and the costs associated with a software failure are motivating forces for well planned, through testing.

There are at least three options for integrating Project

Builder into the test phase:

- Testers do not install Project Builder, use Project Builder functionality to compile and source-control the modules to be tested and hand them off to the testers, whose process remains unchanged.
- The testers import the same project or projects that the developers use.
- Create a project based on the development project but customized for the testers (for example, it does not include support documents, specs, or source), who import it.

A combination of the second and third options works best. Associating the application with a project can be useful during the testing phase, as well. We can create actions to

automatically run test scripts or add script types and make them dependent on the modules to test.

6.4 Test Cases

White box testing:-

Description	check alignment of elements of dashboard
Type	Positive
Steps	check xml code and align the dashboard elements properly
Expected result	the alignment of the elements should comply perfectly for all devices
Actual result	as expected
Remarks	Pass

Description	check alignment of elements of dashboard
Type	Negative
Steps	check xml code and align the dashboard elements improperly
Expected result	the alignment of the elements should vary for different devices
Actual result	as expected
Remarks	Pass

Description	on click of map button maps should be opened
Type	positive
Steps	connect map to the map button
Expected result	on click of map button map should be displayed
Actual result	as expected
Remarks	Pass

Description	on click of map button maps should be opened
Type	negative
Steps	disconnect map to the map button
Expected result	on click of map button map shouldn't be displayed
Actual result	as expected
Remarks	Pass

Black box testing:-

Enter testing

Description	on entering geofence, profile need to switch to silent
Type	positive
Steps	set geofence enter transition
Expected result	on entering geofence, profile need to switch to silent
Actual result	as expected
Remarks	Pass

Description	on entering geofence, profile need to switch to silent
Type	negative
Steps	set geofence enter transition
Expected result	on entering geofence, profile is in normal mode
Actual result	as expected
Remarks	Pass

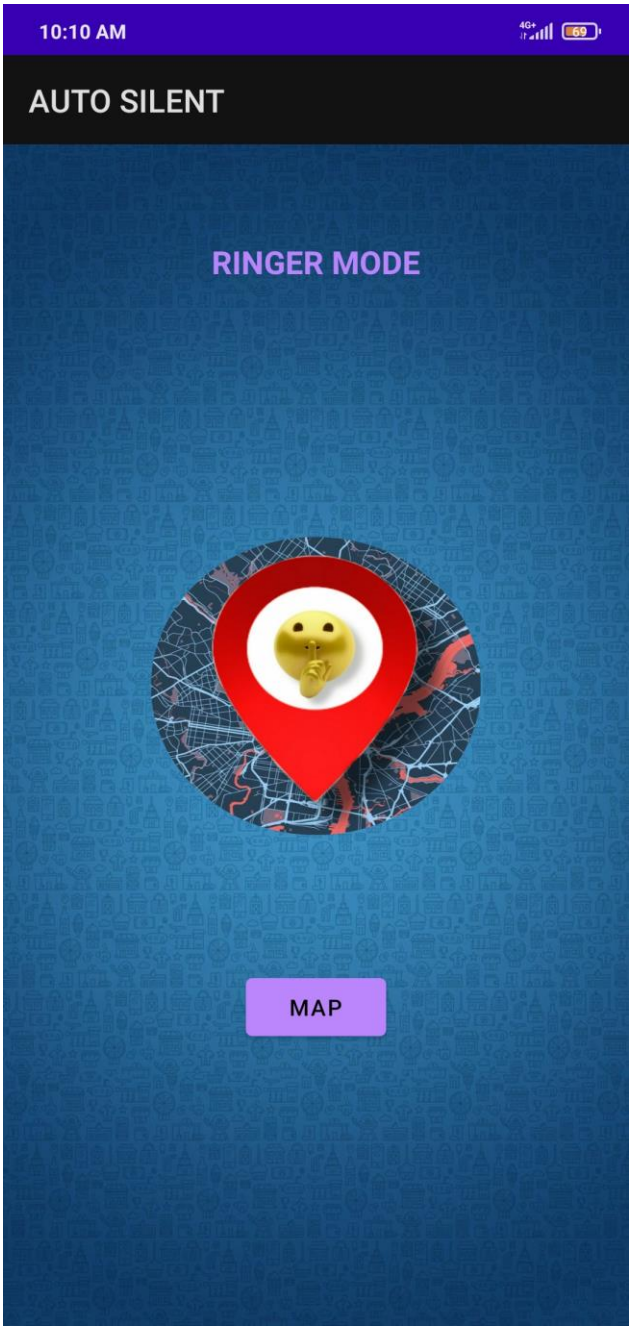
Exit testing

Description	on exiting geofence, profile need to switch to normal mode
Type	positive
Steps	set geofence exit transition
Expected result	on exiting geofence, profile need to switch to normal mode
Actual result	as expected
Remarks	Pass

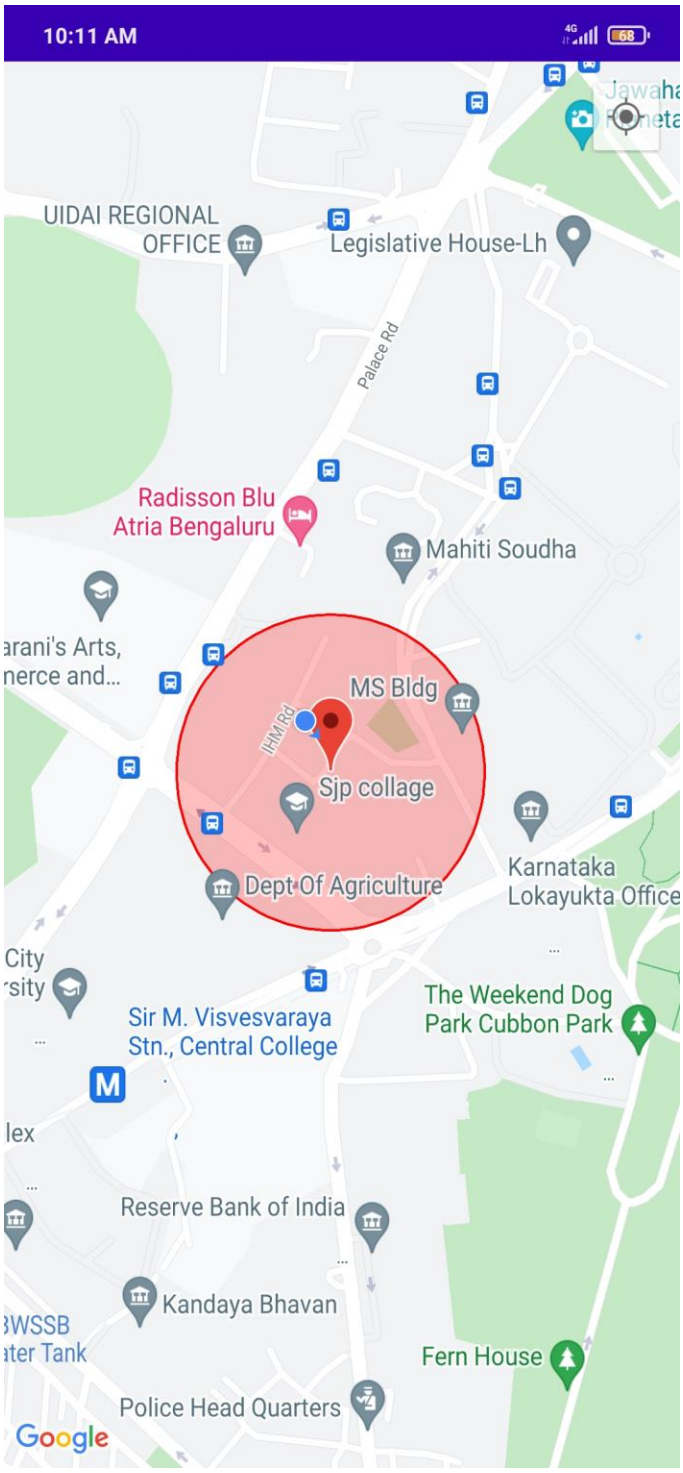
Description	on exiting geofence, profile need to switch to normal mode
Type	negative
Steps	set geofence exit transition
Expected result	on exiting geofence, profile mode should me silent
Actual result	as expected
Remarks	Pass

INPUT AND OUTPUT SCREENS

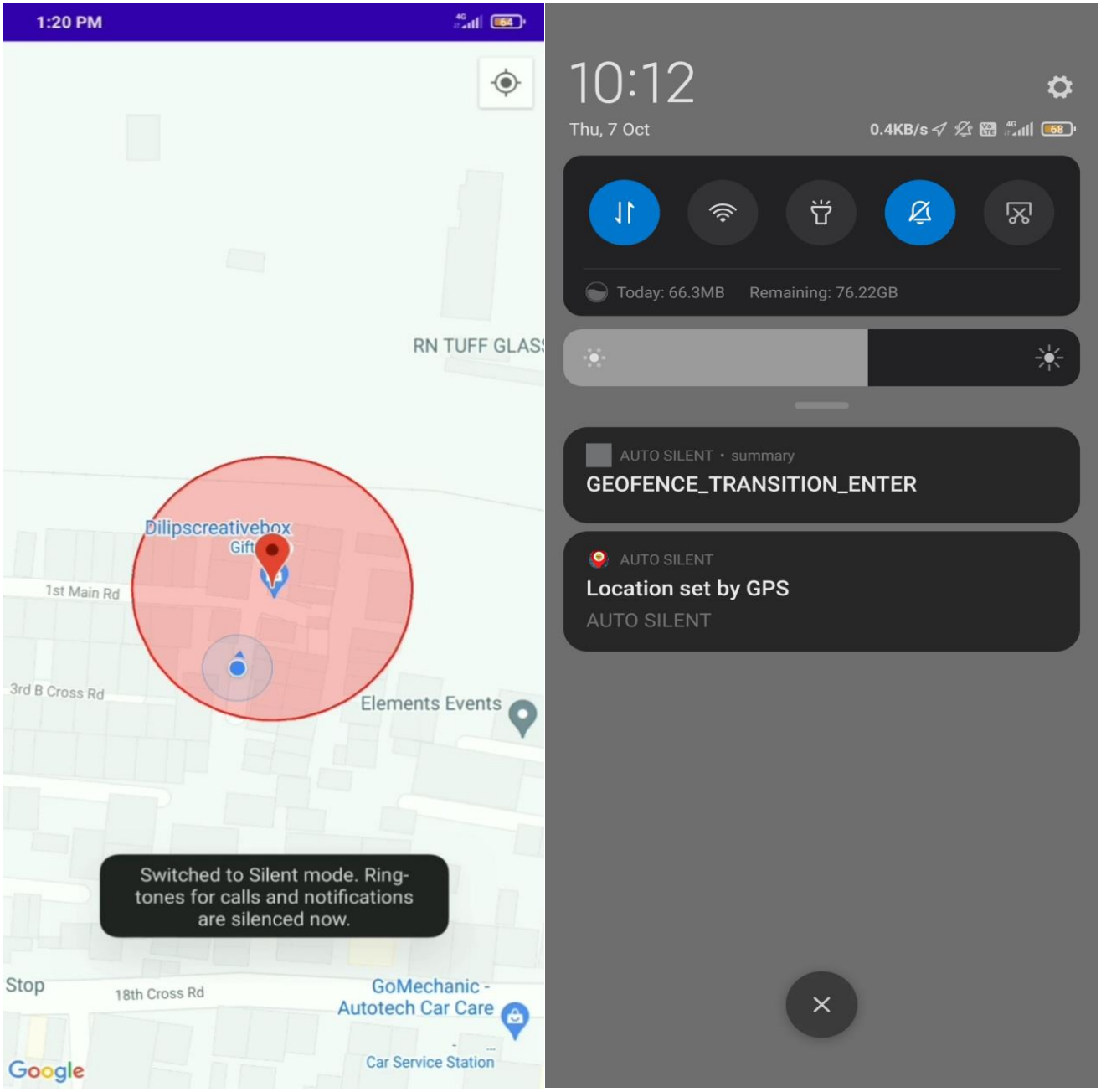
Application home page:



Select Geofence to switch profile to silent mode:



Profile changes to silent mode when users enters the selected geofence:



Profile switches to normal mode when user exists the selected geofence:

IMPLEMENTAION OF SECURITY

Security is the most concerning aspect of application involving sensitive data. Projects implemented without security is equal to void. Security is essential to protect the confidentiality, privacy, integration, authentication and privacy of the data as invasion of these have repercussions that can be fatal. It becomes all the way necessary to implement security in the most convincing manner by factoring in all essential considerations.

In the Application, Security is indispensable to safeguard the data of the end users of the applications. 'The end users may encompass the targeted users, administrators and include even developers. This includes mainly the authentication information as circumventing this will compromise the security aspects of the user's data. Security should also encompass aspects so that an application doesn't gain access to un-supposed data.

This project provides security to all the modules by making sure that only the intended user can access his module by a secure means of authentication. Some of the security measures for the project taken include:

- The auto profiler only works if the DND permission is enabled.
- The user needs to allow GPS to access user location.
- The build version of the SDK should be greater than 28 for high security.
- .Allows to disable the permission whenever required from the app setting

LIMITIATIONS OF THE PROJECT

There are some limitations to the current implementation of the project and the project can be further improved in the future to overcome these limitations.
Some of the limitations of this project are:

- Currently this application can be used only by android devices
- Requires accurate GPS
- Requires active internet connection.
- Lottering delay of 5 seconds(so that it consumes less power).
- Multiple geofence locations cannot be added.

FUTURE ENHANCEMENT OF THE PROJECT

After having developed the application for Geofence based auto audio profile switcher there are a few improvements that could be added to improve the application and the overall usage of the application.

- Can add more audio profiles (vibrating mode, airplane mode)
- Allows multiple Geofences.
- Can specify particular audio profiles for selected mobile numbers.
- Search option will be included in apps.

BIBLIOGRAPHY

Reference Websites:

- www.w3schools.com
- www.stackoverflow.com
- www.github.com
- www.androiddevelopers.com
- www.quora.com
- www.google.com
- www.youtube.com
- www.duckduckgo.com

Bibliography:

- Software Engineering by Pankaj jalote
- OOP with java by Rajesh hongal
- Software Testing by K V K K Prasad