GEOFENCING

CS8611 - MINI PROJECT

Submitted by

PAVITHRA.K (Reg.No.421320104024)

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

KRISHNASAMY

COLLEGE OF ENGINEERING AND TECHNOLOGY

CUDDALORE 607 109

ANNA UNIVERSITY :: CHENNAI 600 025

JUNE 2023

ANNA UNIVERSITY :: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report "**GEOFENCING**" is the bonafide work of "**PAVITHRA.K (Reg.No.421320104024)**" who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Er. C. REIKHA, M.E., M.B.A.,	Er. R. SIVARANJINI, M.E.,
HEAD OF DEPARTMENT	**SUPERVISOR**
Department of Computer Science & Engineering,	Department of Computer Science & Engineering,
Krishnasamy College of Engineering & Technology,	Krishnasamy College of Engineering & Technology,
Cuddalore - 607 109.	Cuddalore - 607 109.

Submitted for the project Viva - Voce held on …………………

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

I take this privilege to express a few words of gratitude and respect to all those who helped me in this completion of my project.

I feel honoured to place my warm salutation to **Krishnasamy College of Engineering & Technology, Cuddalore and Department of Computer Science & Engineering,** which has given me the opportunity to obtain chaired goal of beckoning professional.

I would like to extend my heartfelt and sincere thanks to our chairman **Dr. K. Rajendran, M.S., F.I.C.S., F.A.I.S.,** who has given us an excellent opportunity to exhibit our mode of creativity by our project.

I would like to avail this opportunity to express my gratitude, sincere and heartfelt thanks to our Principal **Dr. G. Elango, M.E., Ph.D.,**Vice Principal **Dr. K. Raghu, M.Sc., M.Phil., Ph.D.,** and our Administrative Officer **Mr. G. Balakrishnan, M.C.A., M.Phil.,** who has provided all help in executing the project successfully,

I acknowledge my heartfelt and sincere thanks to the Head of the Department, **Er. C. Reikha, M.E., M.B.A.,** Associate Professor, Department of Computer Science & Engineering, who has been so helpful in completing the project at the stipulated time and encouraged us to complete our project successfully. I take this opportunity to express my sincere thanks to my guide **Er. R. Sivaranjini, M.E.,** Assistant Professor, Department of Computer Science & Engineering, who encouraged me in each and every step of this project which helped me to make my task a full-fledged one.

Finally, I would like to use this opportunity to express my sincere thanks to all teaching and non-teaching staff for providing all help for successful completion of my project.

ABSTRACT

The widespread availability of smartphones, presents unprecedented opportunity to device creative software solutions that leverage on the powerful hardware embedded in this devices that helps to aid and improve interactions between the user and the environment.

The project follows through those presets and proposes the design of an android based application that makes use of Geo-fencing a Location Based Monitoring (LBM). Geo-fencing is software program feature that enables the use of GPS and RFID capability to define a virtual perimeter around a “hotspot” and trigger event notifications (as ‘people-devices’ move into or out of the established perimeter). Immediate examples of the application of geofencing range from its use in supporting mobile based market campaigns, under a scenario where customers would receive offers granted their proximity of a place of interest providing regional informational services which can support health and safety campaigns for users on a certain regional context to more particular uses such as home monitoring activities etc.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

GPS	:	Global Positioning System
LBS	:	Location Based Services
LBM	:	Location Based Monitoring
RFID	:	Radio Frequency Identification
Cell-Id	:	Cellular Identification
XML	:	Extensible Markup Language
HTML	:	Hyper Text Markup Language
JS	:	JavaScript
SQL	:	Structured Query Language
SGML	:	Standard Generalized Markup Language

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Geofencing is a Location-Based Service (LBS) by which an app or other software program uses Radio Frequency Identification (RFID), Wi-Fi, Global Positioning System (GPS), cellular data or Cellular Identification (Cell-Id) method to trigger a targeted marketing action (such as a text, email, social media advertisement, or app notification) when a mobile device or RFID tag enters or exits a virtual geographic boundary, known as geofencing.

It creates a virtual geographical boundary that triggers a marketing action to a mobile device when a user enters or exits that boundary
While businesses and marketers commonly use geofencing, individual users can also set up geofencing in the mobile apps.

Conclusively, once the application is developed and implemented it will provide a monitoring system and personalized Location-Based Service

1.2 OBJECTIVE

To investigate information on applications and particulars of Geo-fencing. Explore the different available geo-fencing setups and implementations. Propose a Geo-fencing application concept. Provide users with context aware services based on geo-fences.

1.3 EXISTING SYSTEM

Currently many applications are available for geofencing and location based services and monitoring system. Some of the most popular geofencing app is AirDroid geofencing, Life360, Find My Kids, Geotracker. Every existing apps are more functionable but most of the apps are expensive to use the available resources in geofencing app. One of the major drawback is battery and data draining in smart phone by this the users will disable location tracking and mobile data to prolong battery life.

Another drawback is that the maintenance of geofencing apps are time consuming, geofencing needs to be maintained and updated to function optimally. Additionally, its reliable depends on the technology you use. In some cases and with certain hardware, you may experience some bugs along the way. Apart from checking the status of equipment and maintaining the geofences regularly.

1.4 PROPOSED SYSTEM

The proposed system is cost effective and minimizes the data and optimized battery life while using the application. Existing system mostly uses GPS for live location tracking which is one of the main reasons for data and battery draining so instead of using GPS the proposed system is created using RFID Which minimizes the data and battery draining Additionally creating of geofences and allocating perimeters are mostly time consuming so the proposed system has additional option of default perimeter which is already allocated so the user should only enter the name of specific area to be monitored and also provide live tracking of devices. The basic concept of this application is to allow the user a cost efficient location based monitoring with customizable features.

CHAPTER 2

SYSTEM REQUIREMENT SPECIFICATION

2.1 SOFTWARE REQUIREMENTS

1.Laptop or PC

- Windows 7 or higher
- Java, XML
- SQL database
- Android Studio

2.Android Phone or Table

- Android v6.0 or higher

2.2 HARDWARE REQUIREMENTS

1.Laptop or PC

• Processor	:	Intel i3 Processor Based Computer or Higher
• RAM	:	6 GB (Minimum)
• Hard Disk	:	10 GB (Minimum)

2.Android Phone or Tablet

- 1.2 Quad Core Processor or higher
- 2 GB RAM(Minimum)

CHAPTER 3

SOFTWARE DESCRIPTION

3.1 EXTENSIBLE MARKUP LANGUAGE

Extensible Markup Language (XML) lets you define and store data in a shareable manner. XML supports information exchange between computer systems such as websites, databases, and third-party applications. Predefined rules make it easy to transmit data as XML files over any network because the recipient can use those rules to read the data accurately and efficiently. Some important points need to be noticed about XML are as followed:

- XML stands for Extensible Markup Language.
- XML is used to describe the content and structure of data in a document.
- XML is a simplified version of Standard Generalized Markup Language (SGML).
- XML is similar to HTML but tags are not predefined in XML.
- XML was designed to carry data.
- XML simplifies data sharing.
- XML simplifies data transport.
- XML simplifies data availability.

3.1.1 WHY USE XML?

Extensible Markup Language (XML) is a markup language that provides rules to define any data. Unlike other programming languages, XML cannot perform computing operations by itself. Instead, any programming languages or software can be implemented for structured data management.

XML lets you transfer data along with the data's description, preventing the loss of data integrity. You can use this descriptive information to do the following:

- Verify data accuracy.
- Automatically customize data presentation for different users.
- Store data consistently across multiple platforms

Some of the functions in XML are,

- XML lets you transfer data among corporate databases and Web sites without losing crucial descriptive information.
- It lets you automatically customize the presentation of data rather than display the same page to all comers. And it makes searches more efficient because search engines can sort through precise tags rather than long pages of text.
- XML brings sophisticated data coding to Web sites, it helps companies integrate their information flows.
- The Revolutionary power of XML lies in supporting transactions between businesses.
- Using XML all the necessary information can be shared electronically, allowing complex deals to be closed without any human intervention.
- XML can be used for offloading and reloading of databases.
- XML can easily be merged with style sheets to create almost any desired output.
- Virtually, any type of data can be expressed as an XML document.

3.2 JAVASCRIPT

JavaScript is used to create client-side dynamic pages. JavaScript is an object-based scripting language which is lightweight and cross-platform. JavaScript is not a compiled language, but it is a translated language. The JavaScript Translator (embedded in the browser) is responsible for translating the JavaScript code for the web browser.

3.2.1 WHAT IS JAVASCRIPT?

JavaScript (JS) is a light-weight object-oriented programming language which is used by several websites for scripting the webpages. It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an Hyper Text Markup Language (HTML) document. It was introduced in the year 1995 for adding programs to the webpages in the Netscape Navigator browser. Since then, it has been adopted by all other graphical web browsers. With JavaScript, users can build modern web applications to interact directly without reloading the page every time. The traditional website uses JS to provide several forms of interactivity and simplicity.

Although, JavaScript has no connectivity with Java programming language. The name was suggested and provided in the times when Java was gaining popularity in the market. In addition to web browsers, databases such as CouchDB and MongoDB use JavaScript as their scripting and query language.

3.2.2 FEATURES OF JAVASCRIPT

There are following features of JavaScript:

- All popular web browsers support JavaScript as they provide built-in execution environments.
- JavaScript follows the syntax and structure of the C programming language. Thus, it is a structured programming language.
- JavaScript is a weakly typed language, where certain types are implicitly cast (depending on the operation).
- JavaScript is an object-oriented programming language that uses prototypes rather than using classes for inheritance.
- It is a light-weighted and interpreted language.
- It is a case-sensitive language.
- JavaScript is supportable in several operating systems including, Windows, macOS, etc.
- It provides good control to the users over the web browsers.
- The translation of data into a secret code.

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

Geofencing requires communication technologies such as GPS, RFID, Wi-Fi or cellular data. Once a monitorer sets up a geofence, a preprogrammed action is automatically triggered when the monitoring persons mobile device or RFID tag enters or exits the geofence.

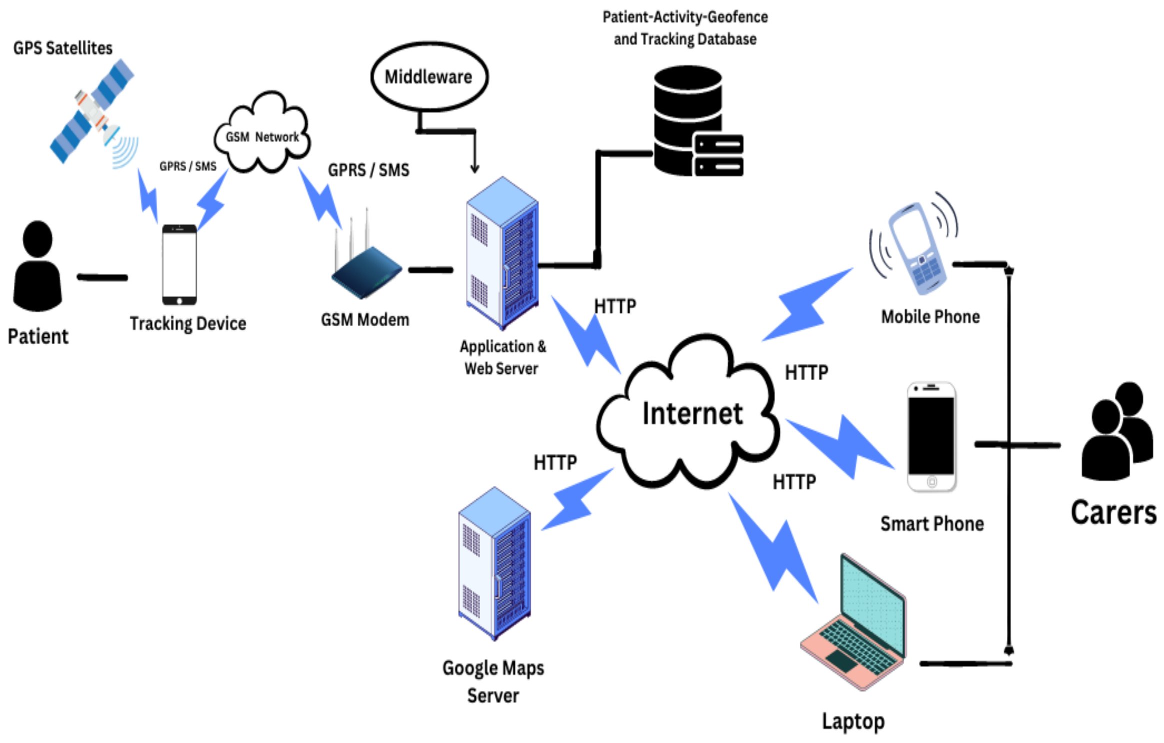


Figure 4.1 System Architecture

GPS satellites circle the earth twice a day in a very precise orbit and transmit signal information to earth. Essentially, the GPS receiver compares the time a signal was transmitted by a satellite with the time it was received. The time difference tells the GPS receiver how far away the satellite is. Now, with distance measurements from a few more satellites, the receiver can determine the user's position and display it on the unit's electronic map.

Using the help of Global System for Mobile Communication (GSM) network the details of the monitoring person is transmitted to GSM Modem using General Packet Radio Service (GPRS) then the Web Server Transmits the location to Internet using middleware.

Hyper Text Transfer Protocol (HTTP) is used to transmit the location details to Google Maps Server (GMS).

The Final location details which are provided by GSM are stored in the tracking database and send the information to tracking persons smart phones.

- **GPRS / SMS –** General Packet Radio Service / Short Message Service allows information to be transmitted through mobile networks.
- **GSM –** Global System for Mobile Communication Receives serial data and transmits the data as text SMS to a host server.
- **Middleware –** Acts as a bridge between the database and web server.
- **Google Maps Server –** It provides detailed information about geographical regions around worldwide.
- **Patient Activity and Tracking Database –** Every tracking details are stored in the Database.

CHAPTER 8

MODULE DESCRIPTION

5.1 LIST OF MODULES

A module description provides detailed information about the module and its supported components, which is accessible in different manners. It as a few distinct types of module:

- Admin Module
- Parent / User Module
- Child Module
- Register / Sign Up Module

5.1.1 ADMIN MODULE

- **Register:** User need to register themselves by filling up basic registration details and by creating valid login credentials.
- **Login:** After successful registration, user needs to login using their valid login credentials.
- **Add Task:** User can add task using Google Maps.
- **Add Waypoint:** The use can add waypoints using Maps for the task added.
- **Start Task**: The user can track his journey to the destination in Google maps where he can see all the way points, current location and destination.
- **View Logs**: The user can see for his journey logs for e.g.: what time he entered the waypoint or when did he reach the destination, but it’s possible only once the task is over.
- **Tracking:** The user can track his location.

5.1.2 PARENT / USER MODULE

- **Registration:** User needs to register first by filling up his/her basic registration details and need to select the type of user.
- **Login:** User can login his respective account and access their respective module.
- **Add/Delete Child:** Parent can add or delete details of child from the application. While registering the name of child, parent need to create login credentials for children and also need to set a geo fence around their children.
- **View Child List:** Parent can view list of registered children’s and track a child.
- **Update Geo Fence:** Parent can update geo fence of their child whenever required.
- **Track Child:** Parent can track child location anytime anywhere.
- **View Notification:** Parent application will receive notification whenever a child exists or enters the geo fence.
- **Logout:** Respective users can logout from their account.

5.1.3 CHILD MODULE

- **Login:** Child can login his account by entering valid login credentials provided by a parent.
- **Emergency:** Will send a notification to parent in case of any emergency occurs.
- **Logout:** Child can logout from his account.

5.1.4 REGISTER / SIGN UP MODULE

- **Register:** User need to register themselves by filling up basic registration details and by creating valid login credentials.
- **Sign Up:** After successful registration, user needs to login using their valid login credentials which are listed below,

1. User Mail Id
2. User Password

CHAPTER 6

CONCLUSION

Geofencing is a powerful and versatile technology that can offer a range of benefits to businesses and individuals. However, careful planning and consideration are required to ensure it is used effectively and ethically. As technology continues to evolve, it will be interesting to see how geofencing is used in new and innovative ways to solve complex problems and improve our experience with mobile apps.

APPENDIX 1

SAMPLE CODING

MainActivity.java

```
package com.eebax.geofencing;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

ActivityMain.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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:layout_height="match_parent"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Activity_maps.xml

```
<?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" />
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.eebax.geofencing">
    <!--
         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_COARSE_LOCATION" />
    <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
```

```
    <uses-permission
android:name="android.permission.ACCESS_BACKGROUND_LOCATION"
/>
    <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.Geofencing">
        <receiver
            android:name=".GeofenceBroadcastReceiver"
            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="AIzaSyDJn0B-nvXOV8sog0U_3BTe4r9MzZIvWAQ"
/>

        <activity
            android:name=".MapsActivity"
            android:exported="true"
            android:label="@string/title_activity_maps">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

GeofenceBroadcastReceiver.java

```
package com.eebax.geofencing;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
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 GeofenceBroadcastReceiver extends BroadcastReceiver {
    private static final String TAG = "GeofenceBroadcastReceiv";
    @Override
    public void onReceive(Context context, Intent intent) {
        // TODO: This method is called when the BroadcastReceiver is receiving
        // an Intent broadcast.
        // Toast.makeText(context, "Geofence triggered",
Toast.LENGTH_SHORT).show();
        NotificationHelper notificationHelper = new NotificationHelper(context);

        GeofencingEvent geofencingEvent = GeofencingEvent.fromIntent(intent);

        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();
        switch (transitionType) {
            case Geofence.GEOFENCE_TRANSITION_ENTER:
```

```
            Toast.makeText(context, "GEOFENCE_TRANSITION_ENTER",
Toast.LENGTH_SHORT).show();
notificationHelper.sendHighPriorityNotification("GEOFENCE_TRANSITION
_ENTER", "", MapsActivity.class);
            break;
          case Geofence.GEOFENCE_TRANSITION_DWELL:
            Toast.makeText(context, "GEOFENCE_TRANSITION_DWELL",
Toast.LENGTH_SHORT).show();
notificationHelper.sendHighPriorityNotification("GEOFENCE_TRANSITION
_DWELL", "", MapsActivity.class);
            break;
          case Geofence.GEOFENCE_TRANSITION_EXIT:
            Toast.makeText(context, "GEOFENCE_TRANSITION_EXIT",
Toast.LENGTH_SHORT).show();
notificationHelper.sendHighPriorityNotification("GEOFENCE_TRANSITION
_EXIT", "", MapsActivity.class);
            break;
        }
    }
}
```

<u>GeofenceHelper.java</u>

```
package com.eebax.geofencing;
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(Geofenc
ingRequest.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, GeofenceBroadcastReceiver.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();
    }
}
```

MapsActivity.java

```
package com.eebax.geofencing;
import androidx.annotation.ColorInt;
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.Circle;
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.eebax.geofencing.databinding.ActivityMapsBinding;
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 ActivityMapsBinding binding;
    GeofencingClient geofencingClient;
    private GeofenceHelper geofenceHelper;
    private float GEOFENCE_RADIUS = 200;
    private String GEOFENCE_ID = "SOME_GEOFENCE_ID";
    private int FINE_LOCATION_ACCESS_REQUEST_CODE = 10001;
    private int BACKGROUND_LOCATION_ACCESS_REQUEST_CODE =
10002;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityMapsBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());
        // 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 cuddalore = new LatLng(11.7468, 79.7558);
        mMap.addMarker(new MarkerOptions().position(cuddalore).title("Marker
in cuddalore"));
        float zoomLevel = 15.0f; //This goes up to 21
        mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(cuddalore,
zoomLevel
        ));
        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_DWELL |
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 unused) {
                Log.d(TAG, "onSuccess: Added...");
            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                String errorMessage = geofenceHelper.getErrorString(e);
                Log.d(TAG, "onFailure: " + e);
            }
        });
    }
    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);
    }
}
```

NotificationHelper.java

```
package com.eebax.geofencing;
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_PUBLI
C);
        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").setBigConten
tTitle(title).bigText(body))
                .setContentIntent(pendingIntent)
                .setAutoCancel(true)
                .build();
        NotificationManagerCompat.from(this).notify(new Random().nextInt(),
notification);
```

```
    }
}
```

DBHandler

```
import android.content.ContentValues;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
public class DBHandler extends SQLiteOpenHelper {
    // creating a constant variables for our database.
    // below variable is for our database name.
    private static final String DB_NAME = "coursedb";
    // below int is our database version
    private static final int DB_VERSION = 1;
    // below variable is for our table name.
    private static final String TABLE_NAME = "mycourses";
    // below variable is for our id column.
    private static final String ID_COL = "id";
    // below variable is for our course name column
    private static final String NAME_COL = "name";
    // below variable id for our course duration column.
    private static final String DURATION_COL = "duration";
    // below variable for our course description column.
    private static final String DESCRIPTION_COL = "description";
    // below variable is for our course tracks column.
    private static final String TRACKS_COL = "tracks";
    // creating a constructor for our database handler.
    public DBHandler(Context context) {
        super(context, DB_NAME, null, DB_VERSION);
    }
    // below method is for creating a database by running a sqlite query
    @Override
    public void onCreate(SQLiteDatabase db) {
        // on below line we are creating
        // an sqlite query and we are
        // setting our column names
        // along with their data types.
        String query = "CREATE TABLE " + TABLE_NAME + " ("
                + ID_COL + " INTEGER PRIMARY KEY AUTOINCREMENT, "
                + NAME_COL + " TEXT,"
                + DURATION_COL + " TEXT,"
                + DESCRIPTION_COL + " TEXT,"
                + TRACKS_COL + " TEXT)";
        // at last we are calling a exec sql
        // method to execute above sql query
        db.execSQL(query);
    }
    // this method is use to add new course to our sqlite database.
    public void addNewCourse(String courseName, String courseDuration,
String courseDescription, String courseTracks) {
        // on below line we are creating a variable for
        // our sqlite database and calling writable method
        // as we are writing data in our database.
        SQLiteDatabase db = this.getWritableDatabase();
        // on below line we are creating a
        // variable for content values.
        ContentValues values = new ContentValues();
        // on below line we are passing all values
        // along with its key and value pair.
        values.put(NAME_COL, courseName);
        values.put(DURATION_COL, courseDuration);
        values.put(DESCRIPTION_COL, courseDescription);
        values.put(TRACKS_COL, courseTracks);
        // after adding all values we are passing
        // content values to our table.
        db.insert(TABLE_NAME, null, values);
        // at last we are closing our
        // database after adding database.
        db.close();
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
{
        // this method is called to check if the table exists already.
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db);
    }
}
```

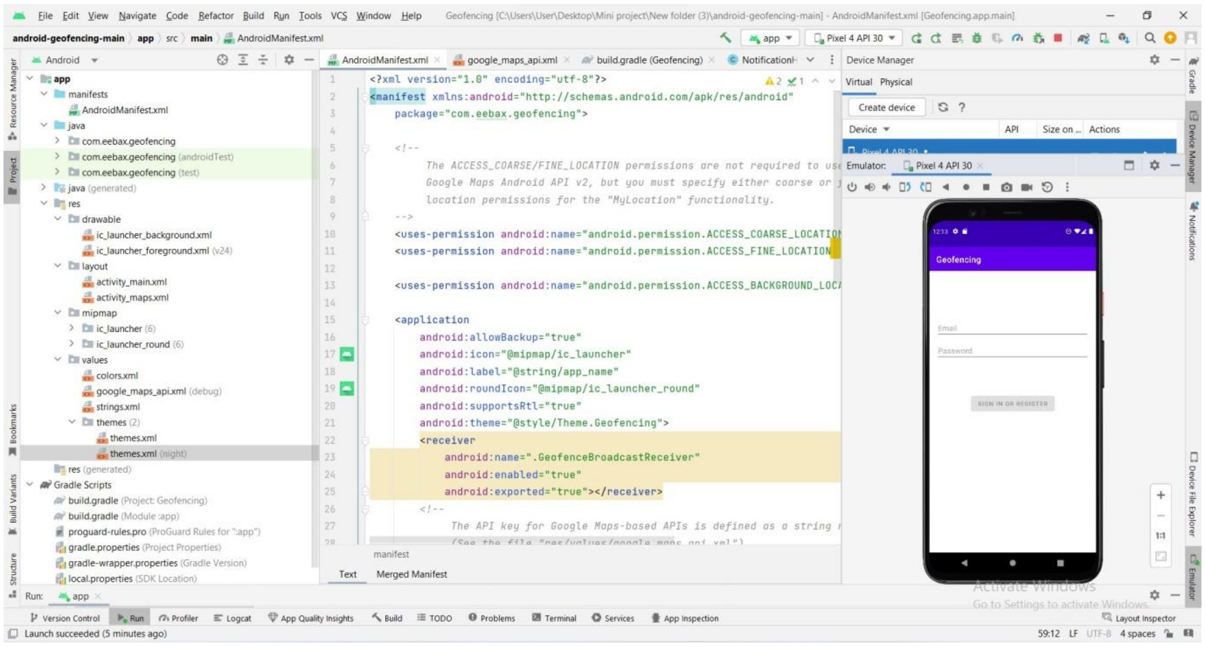
google_maps_api.xml

```
<resources>
    <string name="google_maps_key" templateMergeStrategy="preserve"
translatable="false">AIzaSyDJn0B-
nvXOV8sog0U_3BTe4r9MzZIvWAQ</string>
</resources>
```

APPENDIX 2

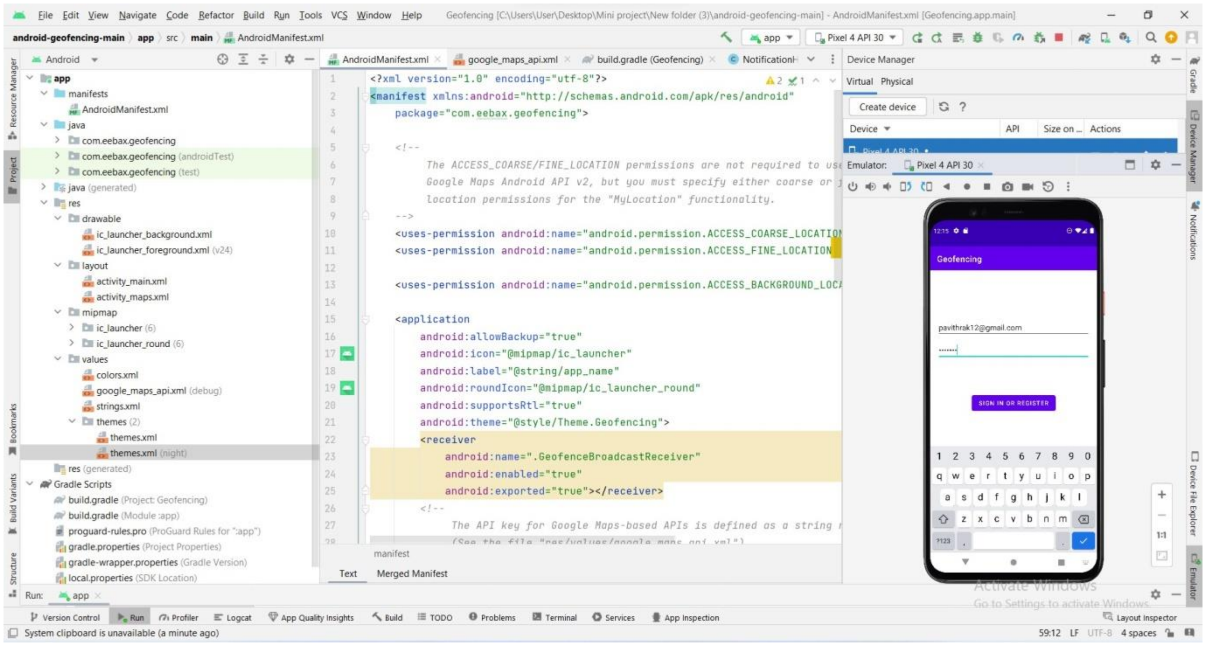
SCREENSHOTS

LOGIN MODULE



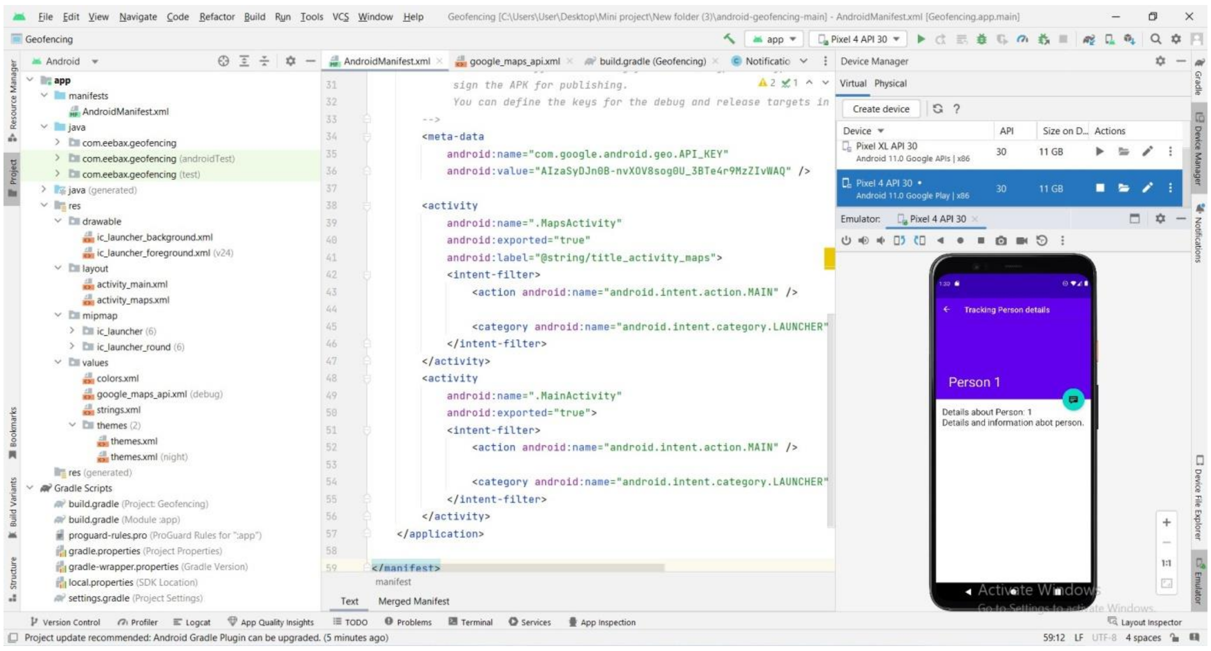
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.eebax.geofencing">

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

    <uses-permission android:name="android.permission.ACCESS_BACKGROUND_LOC

    <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.Geofencing">
        <receiver
            android:name=".GeofenceBroadcastReceiver"
            android:enabled="true"
            android:exported="true"></receiver>
        <!--
             The API key for Google Maps-based APIs is defined as a string r
```

REGISTER USER



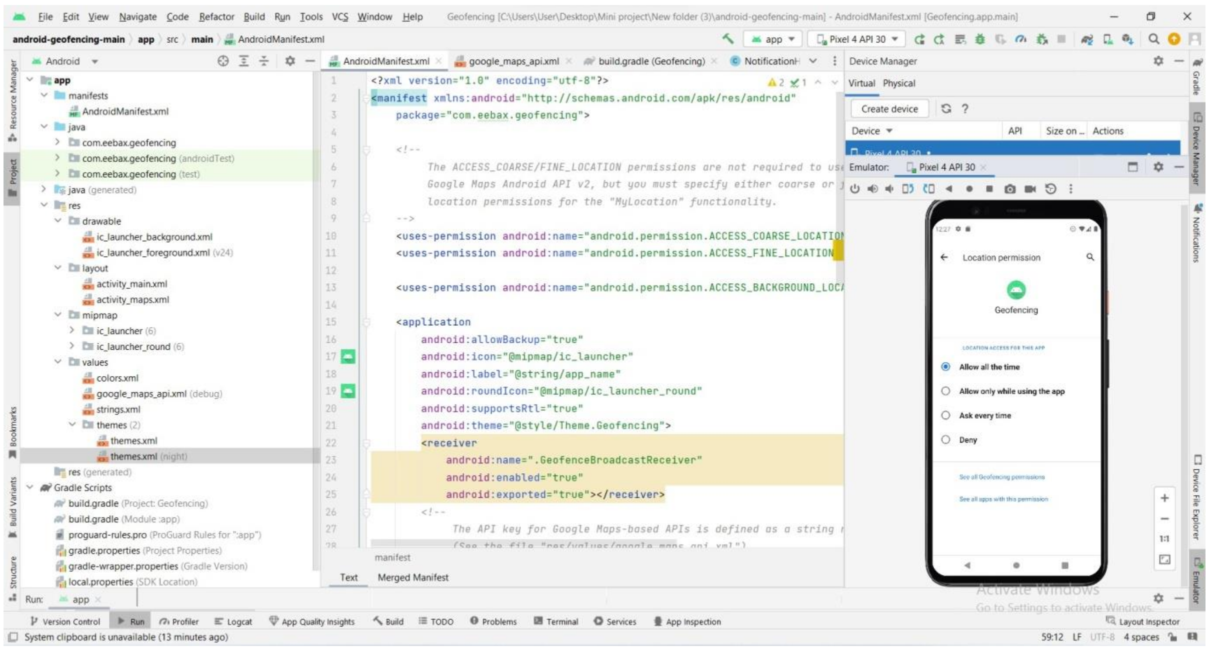
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.eebax.geofencing">

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

    <uses-permission android:name="android.permission.ACCESS_BACKGROUND_LOC

    <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.Geofencing">
        <receiver
            android:name=".GeofenceBroadcastReceiver"
            android:enabled="true"
            android:exported="true"></receiver>
        <!--
             The API key for Google Maps-based APIs is defined as a string r
```

ADD TRACKING PERSON DETAILS



```
            sign the APK for publishing.
            You can define the keys for the debug and release targets in
        -->
        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="AIzaSyDJn0B-nvXOV8sog0U_3BTe4r9MzZIvWAQ" />

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

                <category android:name="android.intent.category.LAUNCHER"
            </intent-filter>
        </activity>
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

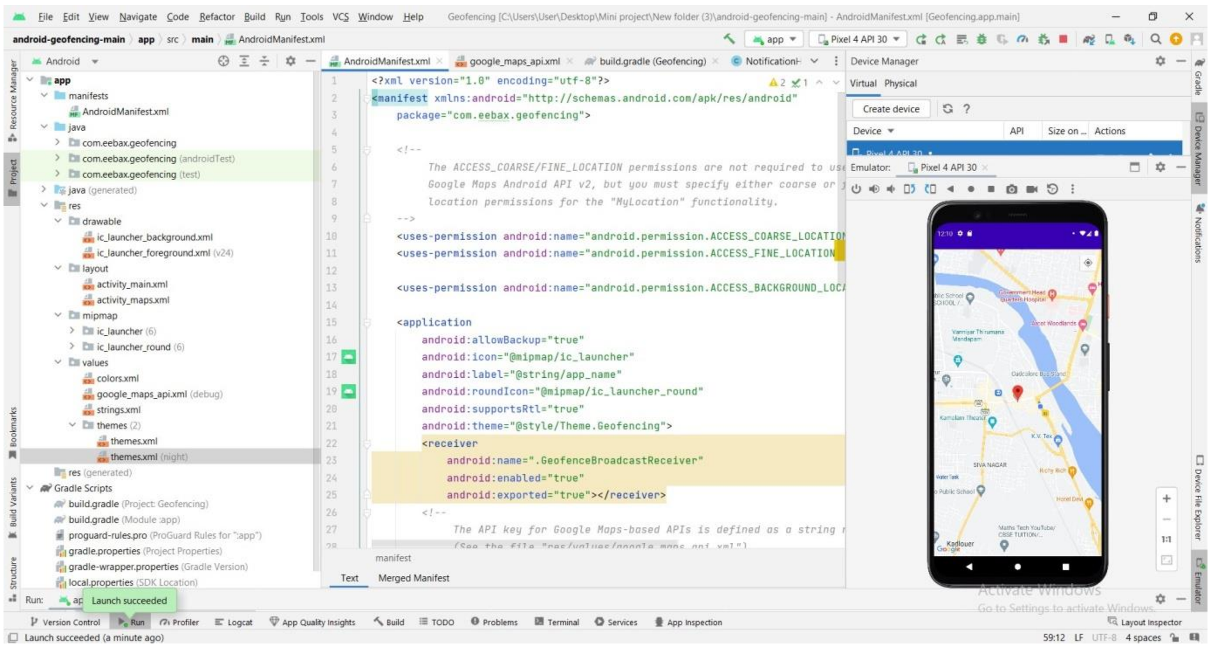
PERMISSION FOR LOCATION ACCESS



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

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

    <uses-permission android:name="android.permission.ACCESS_BACKGROUND_LOCA

    <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.Geofencing">
        <receiver
            android:name=".GeofenceBroadcastReceiver"
            android:enabled="true"
            android:exported="true"></receiver>
        <!--
             The API key for Google Maps-based APIs is defined as a string r
```

Location permission
Geofencing
LOCATION ACCESS FOR THIS APP
Allow all the time
Allow only while using the app
Ask every time
Deny
See all Geofencing permissions
See all apps with this permission

LOCATION BASED SERVICES AND MAP NAVIGATION

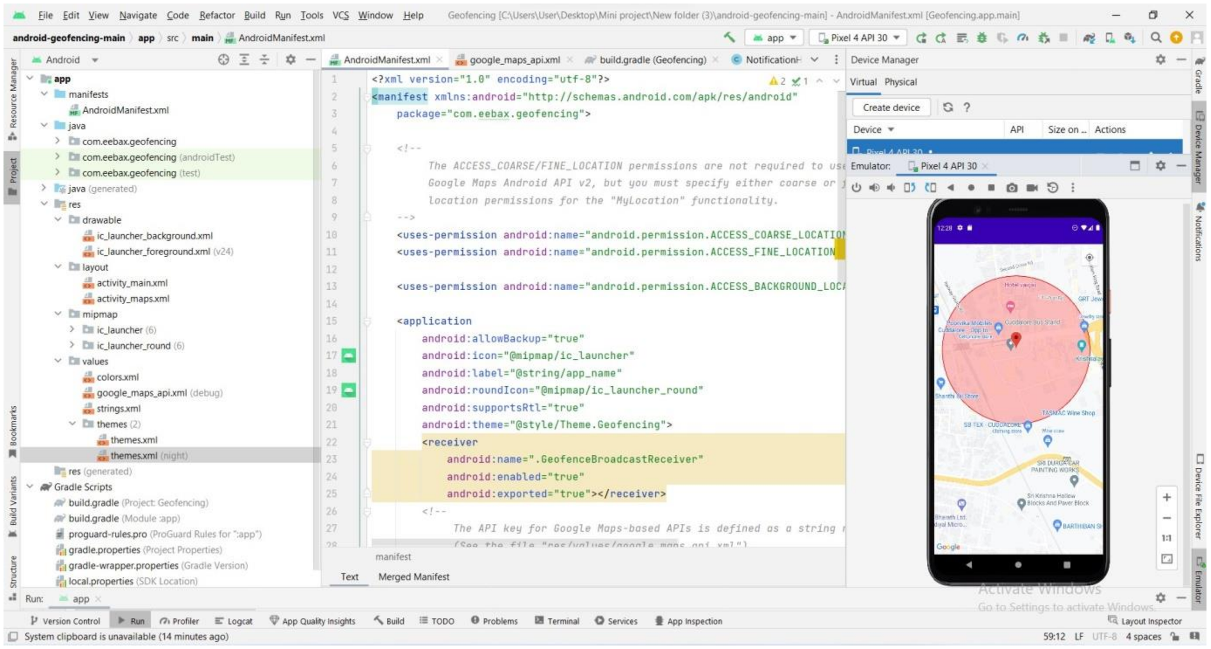


```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.eebax.geofencing">

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

    <uses-permission android:name="android.permission.ACCESS_BACKGROUND_LOCA

    <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.Geofencing">
        <receiver
            android:name=".GeofenceBroadcastReceiver"
            android:enabled="true"
            android:exported="true"></receiver>
        <!--
             The API key for Google Maps-based APIs is defined as a string r
```

Launch succeeded

Launch succeeded (a minute ago)

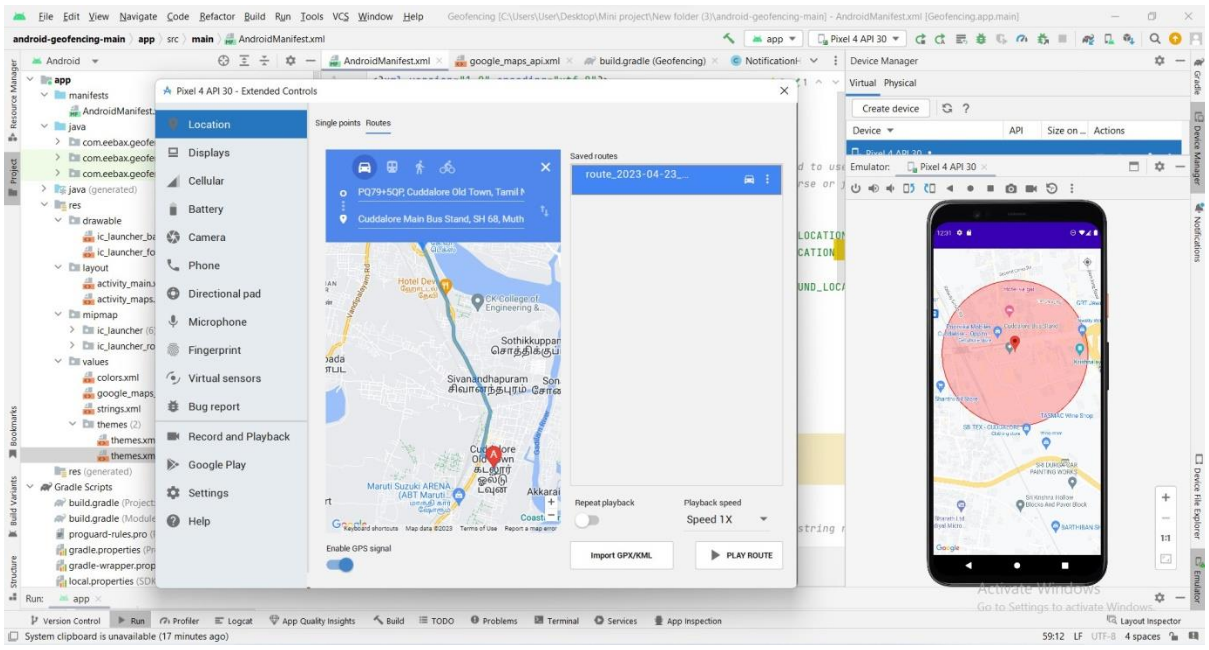
CREATING A GEOFENCE



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

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

    <uses-permission android:name="android.permission.ACCESS_BACKGROUND_LOCA

    <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.Geofencing">
        <receiver
            android:name=".GeofenceBroadcastReceiver"
            android:enabled="true"
            android:exported="true"></receiver>
        <!--
             The API key for Google Maps-based APIs is defined as a string r
```

System clipboard is unavailable (14 minutes ago)

MARK THE DESTINATION OR TRACKING PERSON CURRENT LOCATION IS TAKEN FOR MONITORING



PQ79+5QP, Cuddalore Old Town, Tamil N

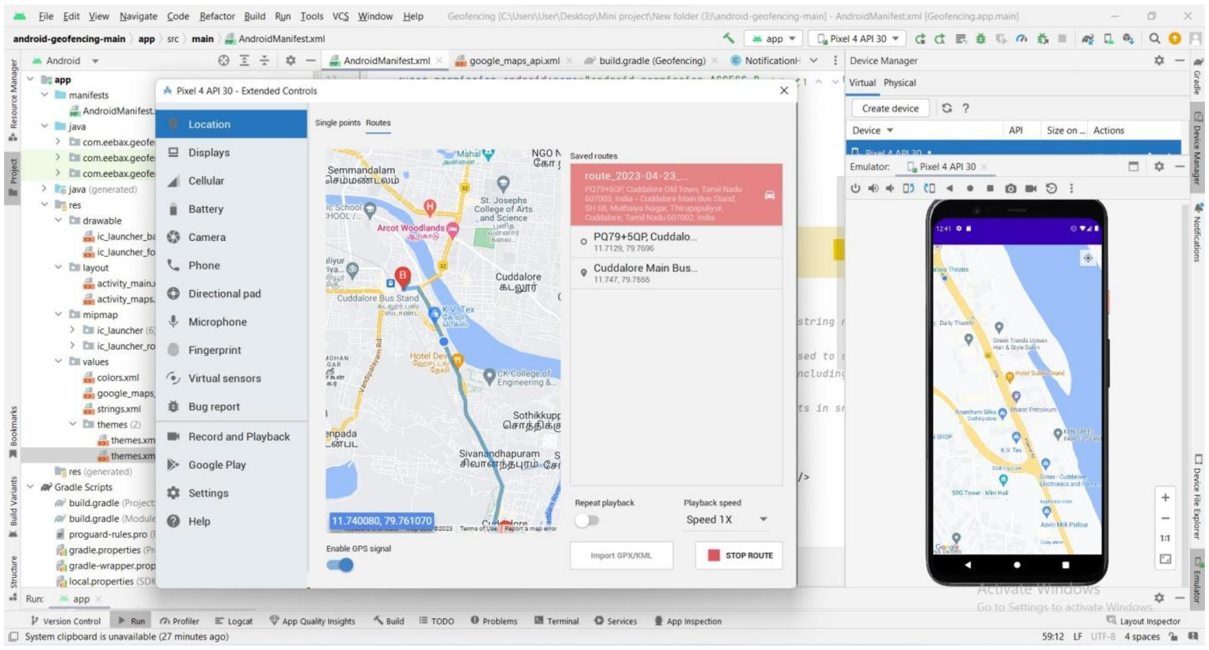
Cuddalore Main Bus Stand, SH 68, Muth

route_2023-04-23_

Speed 1X

System clipboard is unavailable (17 minutes ago)

MONITORING THE TRACKING PERSON LOCATION



route_2023-04-23_

PQ79+5QP, Cuddalore Old Town, Tamil Nadu 607003, India – Cuddalore Main Bus Stand, SH 68, Muthaiya Nagar, Thirupapuliyur, Cuddalore, Tamil Nadu 607002, India

PQ79+5QP, Cuddalo... 11.7129, 79.7696

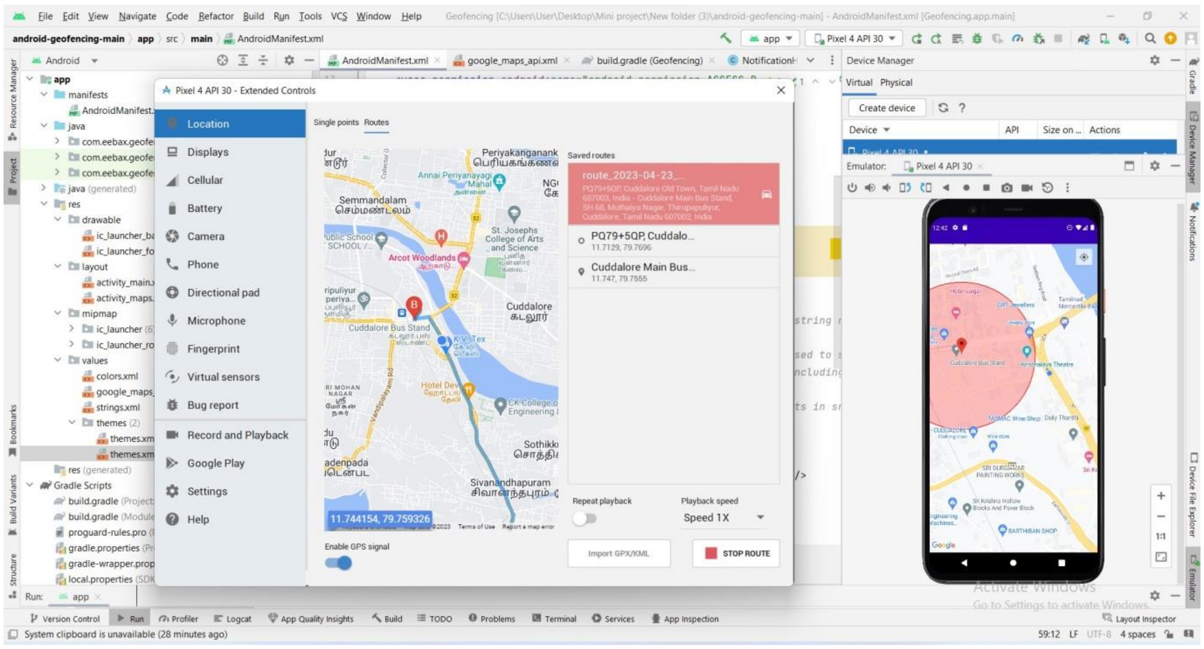
Cuddalore Main Bus... 11.747, 79.7555

11.740080, 79.761070

Speed 1X

System clipboard is unavailable (27 minutes ago)

TRACKING PERSON ENTERING THE GEOFENCE



11.744154, 79.759326

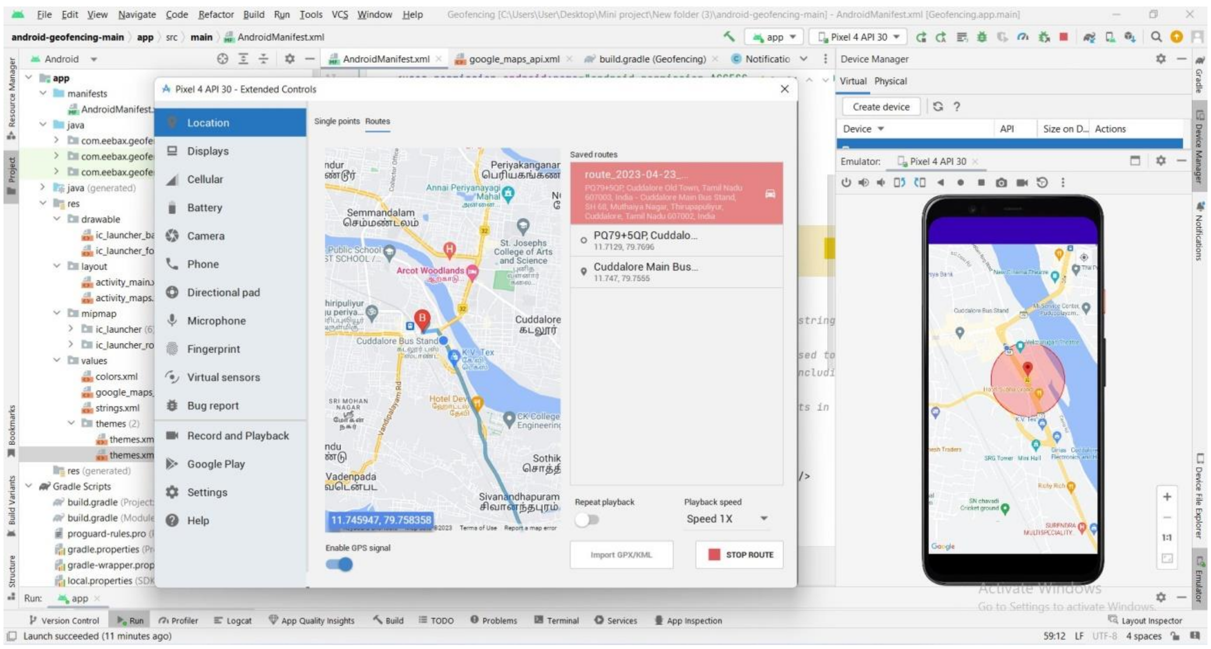
route_2023-04-23_
PQ79+5QP, Cuddalore Old Town, Tamil Nadu 607003, India - Cuddalore Main Bus Stand, SH 68, Muthaiya Nagar, Thirupapuliyur, Cuddalore, Tamil Nadu 607002, India

PQ79+5QP, Cuddalo...
11.7129, 79.7696

Cuddalore Main Bus...
11.747, 79.7555

System clipboard is unavailable (28 minutes ago)

TRACKING PERSON EXITING THE GEOFENCE



11.745947, 79.758358

route_2023-04-23_
PQ79+5QP, Cuddalore Old Town, Tamil Nadu 607003, India - Cuddalore Main Bus Stand, SH 68, Muthaiya Nagar, Thirupapuliyur, Cuddalore, Tamil Nadu 607002, India

PQ79+5QP, Cuddalo...
11.7129, 79.7696

Cuddalore Main Bus...
11.747, 79.7555

Launch succeeded (11 minutes ago)

REFERENCES

[1] A. Boukerche, H. Oliveira, E. Nakamura and A. Loureiro, "Vehicular Ad Hoc Networks: A New Challenge for Localization-Based Systems," ELSEVIER (Science Direct).

[2] B. Jeon and R. C. K. Young, "A System for detecting the Stray of Objects within User defined Region using Location-Based Services," International Journal of Software Engineering and Its Applications, vol. 7, no. 5, pp. 355-362.

[3] A. Küpper, U. Bareth and B. Freese, "Geofencing and Background Tracking – The Next Features," Informatik schafft Communities.

[4] D. Namiot and M. Sneps-Sneppe, "Geofence and Network Proximity".

[5] R. Rajachandrasekar, Z. Ali, S. Hedge and V. Meshram, "Location-Based Query processing: Sensing our Surroundings".

[6] P. Shah, R. Gadgil and N. Tamhankar, "Location Based Reminder Using GPS For Mobile (Android)," ARPN Journal Of Science and Technology, vol. 2.

[7] W.J. Yi and S. J., "System Architecture and Design Flow of Smart Mobile Sensing Systems," Journal of Sensor Technology, pp. 47-56.

[8] https://developer.android.com/training/location/geofencing

[9] https://developers.google.com/android/reference/com/google/android/location/Geofence

[10]https://www.rlogical.com/blog/how-to-integrate-geofencing-in-an-android-app