**VEHICULAR TRACKING APPLICATION**

**AIM:**   
To implement a place tracking App using Android Studio.

**OBJECTIVE:**

To understand how to implement a tracking App.

**SOFTWARE REQUIREMENT:**

Android Studio,Android Sdk 27.

**DESCRIPTION:**

The system allows the app user to track and check vehicle fuel entries , servicing data and even repair/ maintenance status. This system allows users to keep track of their vehicles provided on rent or lease to someone else. Also for tourist vehicle owners to keep track of multiple vehicles. It provides a fuel entry form for each vehicle in that keeps track of its fuel entries for every month.A servicing Entry form is used to maintain the servicing data for each vehicle in per month.Repair & maintenance Entry form for each Vehicle in allows us to track its monthly repair/maintenance status.The vehicle tracking system Uses the drivers GPS Enabled Mobile to track the vehicle on a google maps. This system helps admin to keep track of the driver so that driver cannot do any type of cheating.This system helps admin to keep record of attendance of the driver which helps in calculating salary of the driver

**ALGORITHM:**

1. Design all the layouts required for the vehicle tracking app
2. Design the xml file for the app
3. A pop up window appears which asks us to start tracking.
4. It redirects to the map and tracking takes place.
5. Tracking can be stopped by clicking at stop tracking.

**PROCEDURE:**

1. Open Android Studio
2. Create a new Project
3. Use the design window to create the layouts
4. Enter the respective java code in the respective java files
5. Copy all the pictures to the res folder
6. Connect the google map into the project
7. Start and stop tracking with this app.

**PROGRAM:**

**ACTIVITY3.JAVA:**

package com.example.user.tracklocation;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

public class Activity3 extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_3);

}

}

**ACTIVITY3.XML:**

<?xml version="1.0" encoding="utf-8"?>

<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:layout\_height="match\_parent"

tools:context="com.example.user.tracklocation.Activity3">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:layout\_weight="25">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="Speed" />

<TextView

android:id="@+id/textView1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:layout\_weight="25">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="Acceleration" />

<TextView

android:id="@+id/textView2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:layout\_weight="25">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="Distance" />

<TextView

android:id="@+id/textView3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:layout\_weight="25">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="Driving Behaviour" />

<TextView

android:id="@+id/textView4"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

/>

</LinearLayout>

</LinearLayout>

</RelativeLayout>

**MANIFEST.XML:**

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.example.user.tracklocation">

<!--

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\_COARSE\_LOCATION" />

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />

<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android:name="com.google.android.providers.gsf.permission.READ\_GSERVICES" />

<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/AppTheme">

<!--

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/.

-->

<activity android:name=".MainActivity">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<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=".Activity3"></activity>

</application>

</manifest>

**GOOGLEAPI.XML:**

<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=2B:6A:70:64:EA:C9:69:C4:BE:05:65:9D:CB:08:52:14:15:69:32:FE%3Bcom.example.user.tracklocation

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

Package name:

2B:6A:70:64:EA:C9:69:C4:BE:05:65:9D:CB:08:52:14:15:69:32:FE

SHA-1 certificate fingerprint:

2B:6A:70:64:EA:C9:69:C4:BE:05:65:9D:CB:08:52:14:15:69:32:FE

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">AIzaSyBlGn1q2URZmfJGrIelJlQPEOYTViGFNC4</string>

</resources>

**MAINACTIVITY.JAVA:**

package com.example.user.tracklocation;

import android.content.Intent;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

public void track(View v)

{

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

startActivity(i);

}

}

**MAPS.XML:**

<RelativeLayout 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:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context="com.example.user.tracklocation.MapsActivity">

<Button

android:id="@+id/button3"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentStart="true"

android:layout\_alignParentTop="true"

android:onClick="analyse"

android:text="Button" />

<fragment

android:id="@+id/map"

android:name="com.google.android.gms.maps.SupportMapFragment"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_below="@+id/button3"

/>

<Button

android:id="@+id/btn2"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

android:layout\_centerHorizontal="true"

android:text="StopTracking"

android:onClick="stop"/>

</RelativeLayout>

**MAPSACTIVTY.JAVA:**

package com.example.user.tracklocation;

import android.Manifest;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.graphics.Color;

import android.location.Location;

import android.os.Build;

import android.support.annotation.NonNull;

import android.support.annotation.Nullable;

import android.support.v4.app.ActivityCompat;

import android.support.v4.app.FragmentActivity;

import android.os.Bundle;

import android.support.v4.content.ContextCompat;

import android.view.View;

import android.widget.TextView;

import android.widget.Toast;

import com.google.android.gms.common.ConnectionResult;

import com.google.android.gms.common.api.GoogleApiClient;

import com.google.android.gms.location.LocationListener;

import com.google.android.gms.location.LocationRequest;

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.BitmapDescriptorFactory;

import com.google.android.gms.maps.model.LatLng;

import com.google.android.gms.maps.model.Marker;

import com.google.android.gms.maps.model.MarkerOptions;

import com.google.android.gms.maps.model.Polyline;

import com.google.android.gms.maps.model.PolylineOptions;

import java.text.DateFormat;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Date;

public class MapsActivity extends FragmentActivity implements OnMapReadyCallback,

GoogleApiClient.ConnectionCallbacks,

GoogleApiClient.OnConnectionFailedListener,LocationListener {

private GoogleMap mMap;

private GoogleApiClient client;

private LocationRequest locationRequest;

private Location lastLocation;

private Marker currentLocationMarker;

public static final int REQUEST\_LOCATION\_CODE=99;

private ArrayList<LatLng> points; //added

Polyline line; //added

LatLng latLng;

public int i,k;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_maps);

if(Build.VERSION.SDK\_INT>=Build.VERSION\_CODES.M)

{

checkLocationPermission();

}

points = new ArrayList<LatLng>();

SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()

.findFragmentById(R.id.map);

mapFragment.getMapAsync(this);

i=0;

k=0;

}

@Override

public void onMapReady(GoogleMap googleMap) {

mMap = googleMap;

if(ContextCompat.checkSelfPermission(this,Manifest.permission.ACCESS\_FINE\_LOCATION)==PackageManager.PERMISSION\_GRANTED) {

buildGoogleApiClient();

mMap.setMyLocationEnabled(true);

}

}

protected synchronized void buildGoogleApiClient()

{

client=new GoogleApiClient.Builder(this).addConnectionCallbacks(this).

addOnConnectionFailedListener(this).addApi(LocationServices.API).build();

client.connect();

}

@Override

public void onLocationChanged(Location location) {

k++;

lastLocation=location;

latLng=new LatLng(location.getLatitude(),location.getLongitude());

points.add(latLng); //added

redrawLine();

}

@Override

public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {

switch (requestCode)

{

case REQUEST\_LOCATION\_CODE:

if(grantResults.length>0&&grantResults[0]==PackageManager.PERMISSION\_GRANTED)

{

if(ContextCompat.checkSelfPermission(this,Manifest.permission.ACCESS\_FINE\_LOCATION)==PackageManager.PERMISSION\_GRANTED)

{

if(client==null)

{

buildGoogleApiClient();

}

mMap.setMyLocationEnabled(true);

}

}

else{

Toast.makeText(this, "Permission Denied", Toast.LENGTH\_SHORT).show();

}

}

return;

}

public boolean checkLocationPermission()

{

if(ContextCompat.checkSelfPermission(this,Manifest.permission.ACCESS\_FINE\_LOCATION)!=PackageManager.PERMISSION\_GRANTED)

{

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

{

ActivityCompat.requestPermissions(this,new String[]{Manifest.permission.ACCESS\_FINE\_LOCATION},REQUEST\_LOCATION\_CODE);

}

else

{

ActivityCompat.requestPermissions(this,new String[]{Manifest.permission.ACCESS\_FINE\_LOCATION},REQUEST\_LOCATION\_CODE);

}

return false;

}

else

{

return true;

}

}

@Override

public void onConnected(@Nullable Bundle bundle) {

locationRequest=new LocationRequest();

locationRequest.setInterval(10000);

locationRequest.setFastestInterval(10000);

locationRequest.setPriority(LocationRequest.PRIORITY\_BALANCED\_POWER\_ACCURACY);

if(ContextCompat.checkSelfPermission(this,Manifest.permission.ACCESS\_FINE\_LOCATION)==PackageManager.PERMISSION\_GRANTED) {

LocationServices.FusedLocationApi.requestLocationUpdates(client, locationRequest, this);

}

}

@Override

public void onConnectionSuspended(int i) { }

@Override

public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {

}

private void redrawLine(){

mMap.clear();

PolylineOptions options = new PolylineOptions().width(5).color(Color.BLUE).geodesic(true);

for (int z = 0; z < points.size(); z++) {

LatLng point = points.get(z);

options.add(point);

}

addMarker(); //add Marker in current position

line = mMap.addPolyline(options); //add Polyline

addMarker();

}

public void addMarker()

{

MarkerOptions markerOptions=new MarkerOptions();

markerOptions.position(latLng);

markerOptions.title(""+lastLocation.getLatitude()+","+lastLocation.getLongitude());

markerOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE\_RED));

currentLocationMarker=mMap.addMarker(markerOptions);

mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng,18));

}

public void stop(View v)

{

LocationServices.FusedLocationApi.removeLocationUpdates(client,this);

}

public void analyse(View v)

{

Intent i=new Intent(MapsActivity.this,Activity3.class);

startActivity(i);

}

}

**MAIN.XML:**

<?xml version="1.0" encoding="utf-8"?>

<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:layout\_height="match\_parent"

tools:context="com.example.user.tracklocation.MainActivity">

<Button

android:id="@+id/bt1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentTop="true"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="195dp"

android:text="Start Tracking"

android:onClick="track"/>

</RelativeLayout>

**OUTPUT:**



