BATCH AND ROLL NO: P8 42405
EXPERIMENT NO.10
TITLE: Design a mobile app using Google Map and GPS to trace the location.
DATE OF PERFORMANCE
DATE OF CHECKING

Title: Design a mobile app using Google Map and GPS to trace the location.

Requirements:
1 Android studio
2.Google Play service Packages

Theory:

Android allows us to integrate Google Maps in our application. For this Google provides us a library via Google Play Services for using maps. In order to use the Google Maps API, you must register your application on the **Google Developer Console** and enable the API.

Follow these steps to create the project in Android Studio.

Step 1. Install the Google Play Services SDK

Add Google Play services to Android Studio. To make the Google Play services APIs available to your app:

1. Open the build.gradle file inside your application module directory.

2. Add a new build rule under dependencies for the latest version of play-services, using one of the APIs listed below.

Implementation ‘com.google.android.gms:play-services-maps:+ googlePlayVersion’ (other Google Play API
https://developers.google.com/android/guides/setup)

3. Save the changes and click Sync Project with Gradle Files in the toolbar.

4. If you receive an error, check that your top-level build.gradle contains a reference to the google() repo.

Step 2. Create a new Google Maps project

These are the steps to create a new project with Google Maps activity:

1. Start Android Studio.

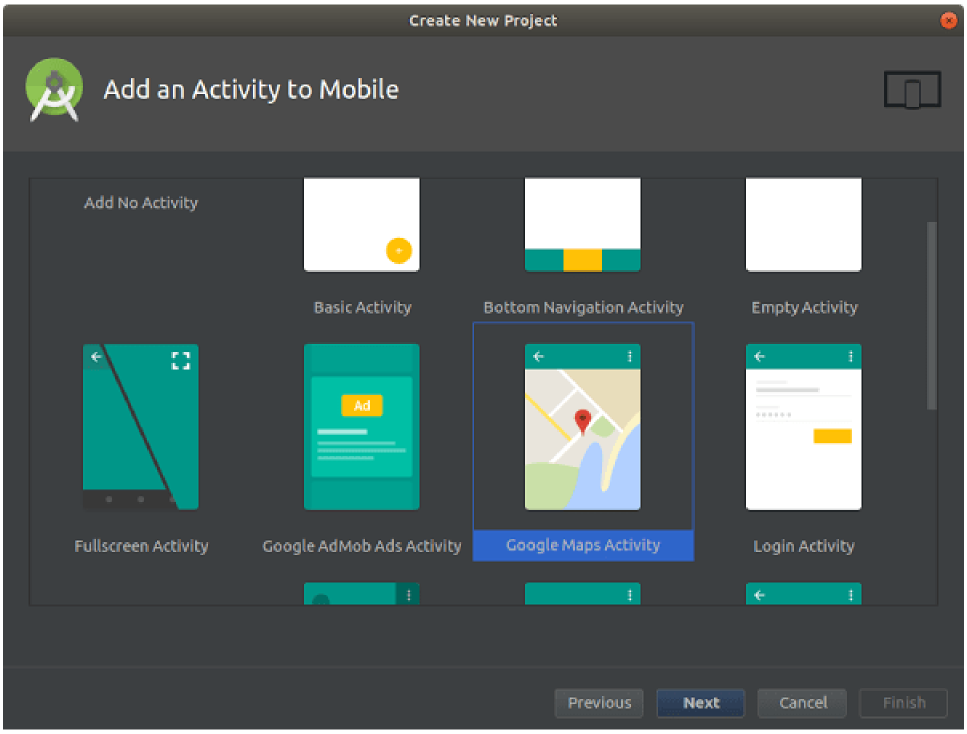
2. Start a new project::

- Click **Start a new Android Studio project** in the Welcome to Android Studio dialog
- If you don't see this dialog, then click **File –> New –> New Project**.

3. Fill in the app name, company domain, and location.

4. Choose form factors, for example Phone and Tablet.

5. Select the Google Maps Activity in the **Add an Activity to Mobile** dialog.



6. After filling in the activity name and title, click **Finish**. Don't touch the default values – they're okay.

After a few seconds, your project will be built by Gradle. Then you'll see google_maps_api.xml and the MapsActivity.java files in the editor.**Step 3. Get a Google Maps API key**

The Android Maps API is used to access Google's servers, and there are several ways you can get a key. An API key is free and supports any number of users.

- The easiest way to get a key is to use the link provided in the google_maps_api.xml file that Android Studio created for you:

1. Copy the link and paste it into your browser. It will direct you to the Google API Console and you won't have to fill in anything by yourself.

2. Select a project you've created before or create a new one.

3. Create an Android-restricted API key for your project.

4. Copy the API key, go back to Android Studio, and paste it into the element in the google_maps_api.xml file.

- Another option for getting an API key is this:

Use the credentials provided in the google_maps_api.xml file that Android Studio created for you:

1. Copy the credentials provided in the google_maps_api.xml file.

2. Go to the Google API Console in your browser.

3. Use the copied credentials to add your app to an existing API key or to create a new API key.

Step 4. Check your code

Now, when everything is set up, look at the code and check these files in your project.

Code:

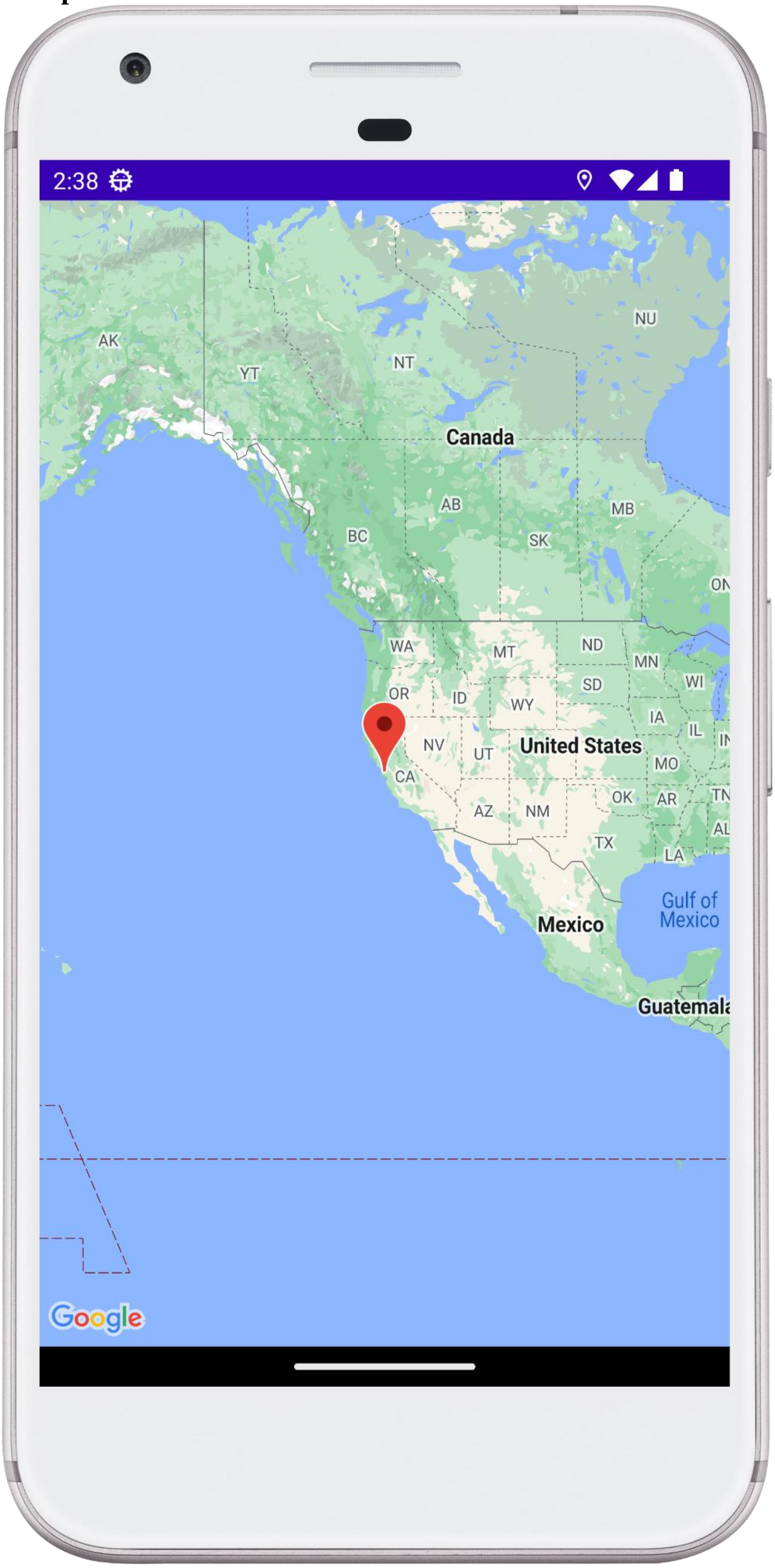
MapsActivity.java

```
package com.example.assign10;

import androidx.annotation.NonNull;
import androidx.core.app.ActivityCompat;
import androidx.fragment.app.FragmentActivity;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.Manifest;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
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.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import com.example.assign10.databinding.ActivityMapsBinding;

public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {

    private GoogleMap mMap;
    private ActivityMapsBinding binding;
    private LocationListener locationListener;
    private LocationManager locationManager;
    private final long MIN_DIST=5;
    private final long MIN_TIME=1000;
    private LatLng latLng;

    @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);

        ActivityCompat.requestPermissions(this, new String[]{
                Manifest.permission.ACCESS_FINE_LOCATION
        }, PackageManager.PERMISSION_GRANTED);
        ActivityCompat.requestPermissions(this, new String[]{
                Manifest.permission.ACCESS_COARSE_LOCATION
        }, PackageManager.PERMISSION_GRANTED);
    }

    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;
        // Add a marker in Sydney and move the camera
        LatLng sydney = new LatLng(-34, 151);
        mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in Sydney"));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
        locationListener = new LocationListener() {
            @Override
            public void onLocationChanged(@NonNull Location location) {
                latLng = new LatLng(location.getLatitude(),location.getLongitude());
                mMap.addMarker(new MarkerOptions().position(latLng ).title("My position"));
                mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));
            }
        };
        locationManager = (LocationManager) getSystemService(LOCATION_SERVICE);
        try{

locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,MIN_TIME,MIN_DIST
,locationListener);
        } catch (SecurityException e){
            e.printStackTrace();
        }
    }
}
```

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" />
```

Output:


2:38
AK
YT
NT
NU
Canada
AB
MB
BC
SK
ON
WA
MT
ND
MN
OR
ID
SD
WI
WY
IA
IL
IN
NV
UT
United States
MO
CA
OK
AR
TN
AZ
NM
AL
TX
LA
Gulf of Mexico
Mexico
Guatemala
Google

CONCLUSION:

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………