CSE 162 Mobile Computing

Lab 4 Location and Map Programming

Hua Huang

Tasks

- Creating a google map
- Using Location services

Maps SDK for Android

- Set up the development environment
- Set up an Android device
- Create a Google Maps project in Android Studio
- Set up in Cloud Console
- Add the API key to your app
- Deploy and run the app

Set up the development environment

- Android Studio is required. If you haven't already done so, download and install it.
- Add the Google Play services SDK to Android Studio. The Maps SDK for Android is distributed as part of the Google Play services SDK, which you can add through the SDK Manager.

Set up an Android device

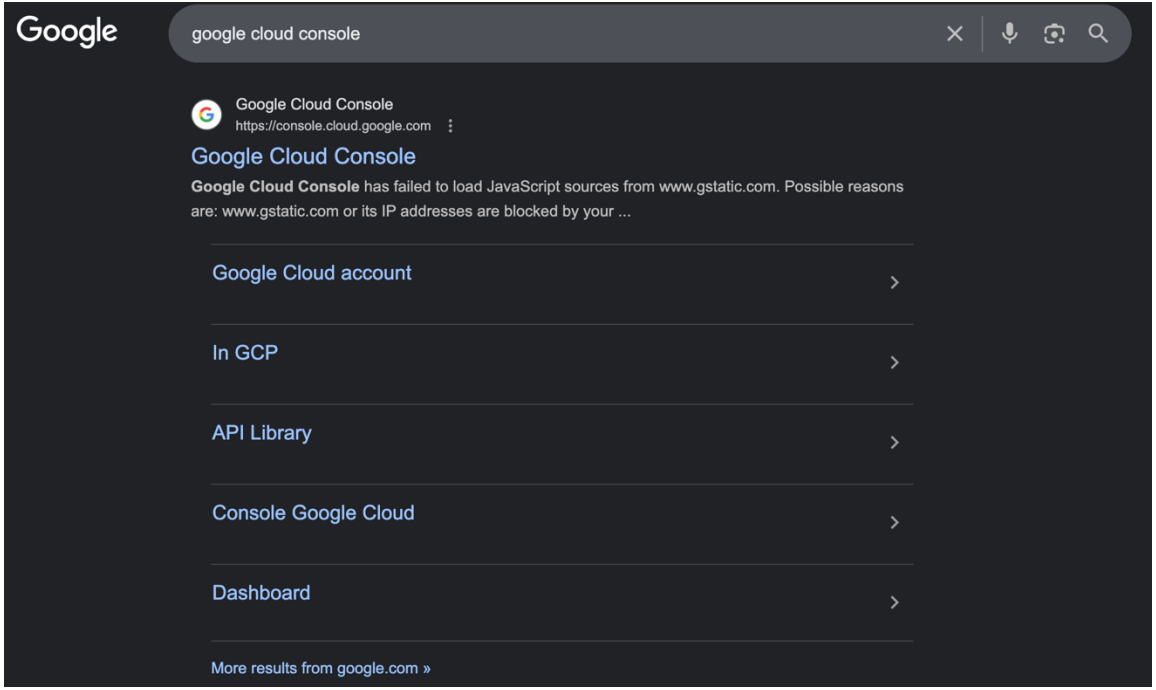
- To run an app that uses the Maps SDK for Android, you must deploy it to an Android device or Android emulator that is based on Android 4.0 or higher and includes the Google APIs.

Create a Google Maps project in Android Studio

- Open Android Studio, and click Create New Project.
- In the New Project window, under the Phone category, select the Empty Activity
- Complete the Google Maps Activity form:
  - Set Language to Java
  - Set Minimum SDK to an SDK version compatible with your test device. You must select a version greater than the minimum version required by the Maps SDK for Android version 18.0.x, which is currently Android API Level 19 (Android 4.4, KitKat) or higher.

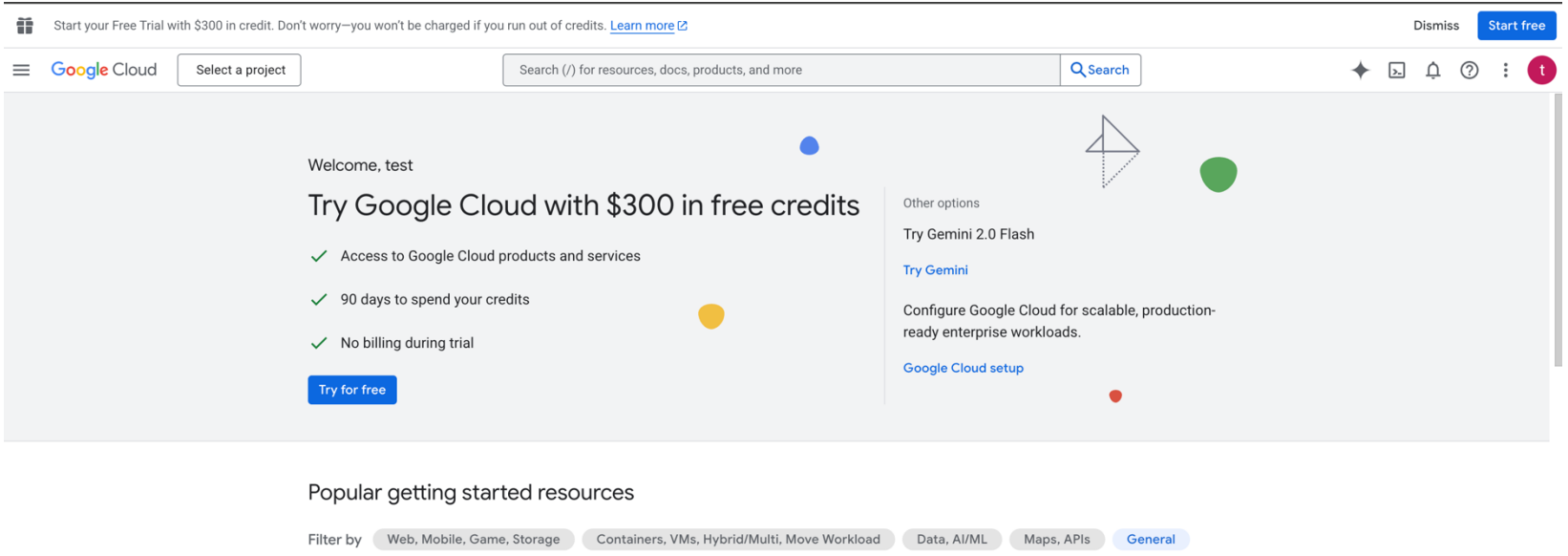
Set up in Cloud Console

- On your browser look up Google Cloud Console



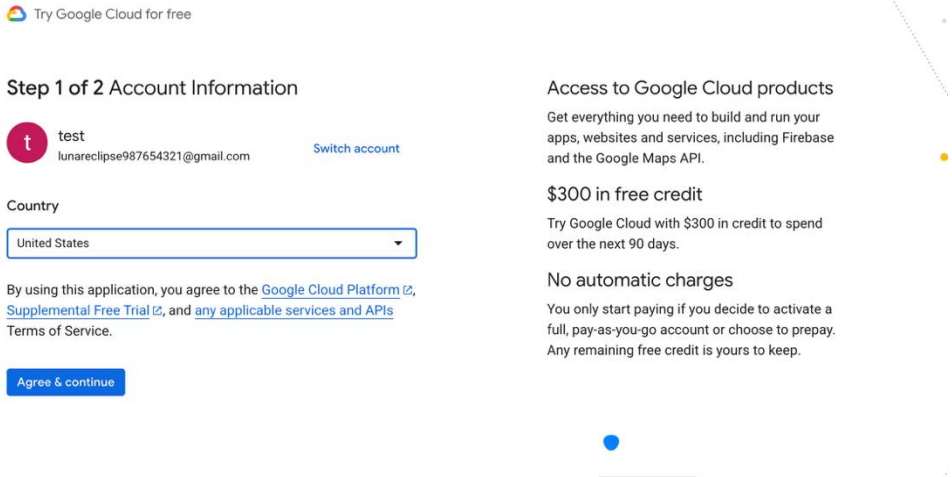
Set up in Cloud Console

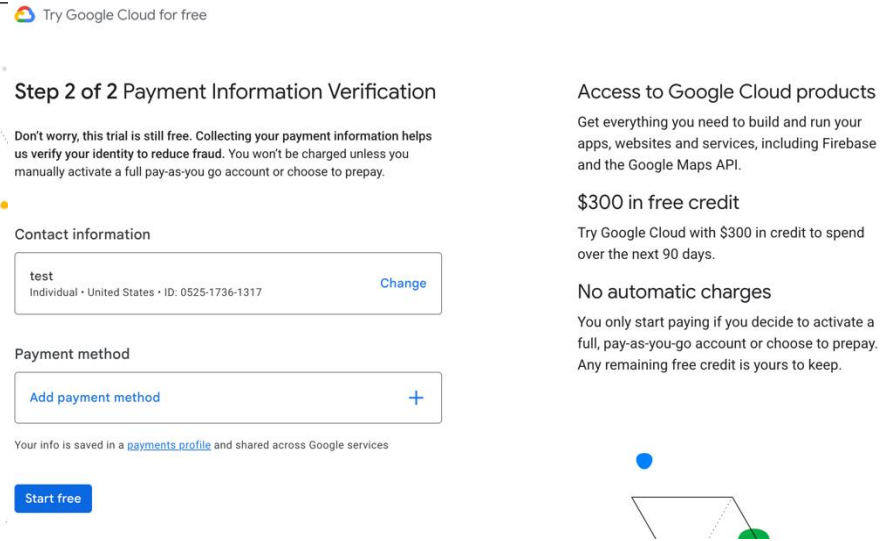
- You need a Gmail account
- For first time users get $300 free credits, click Try for free



Set up in Cloud Console

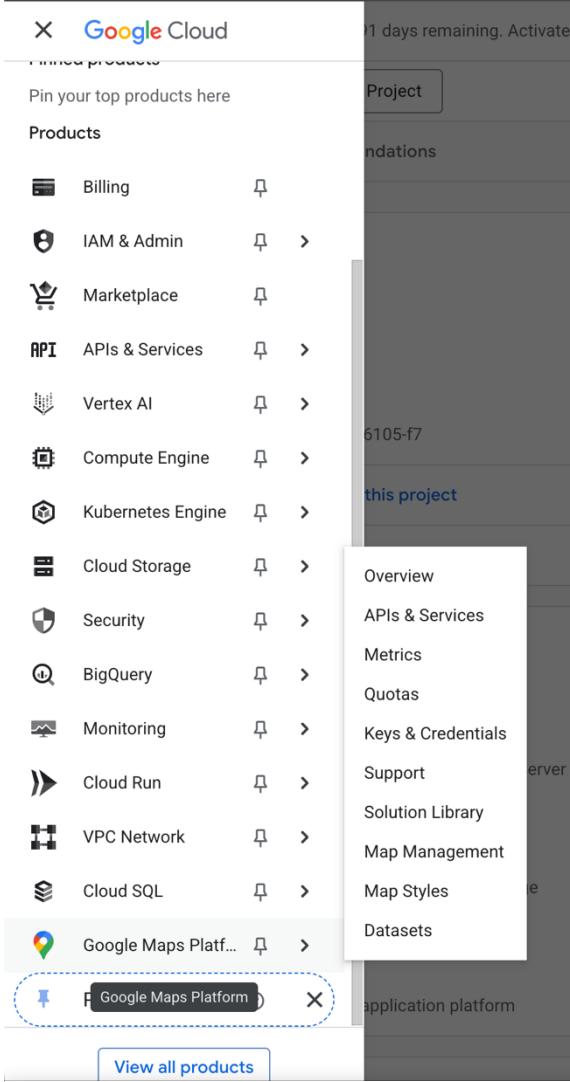
- You will need to setup a billing account and enable the Google Maps API.





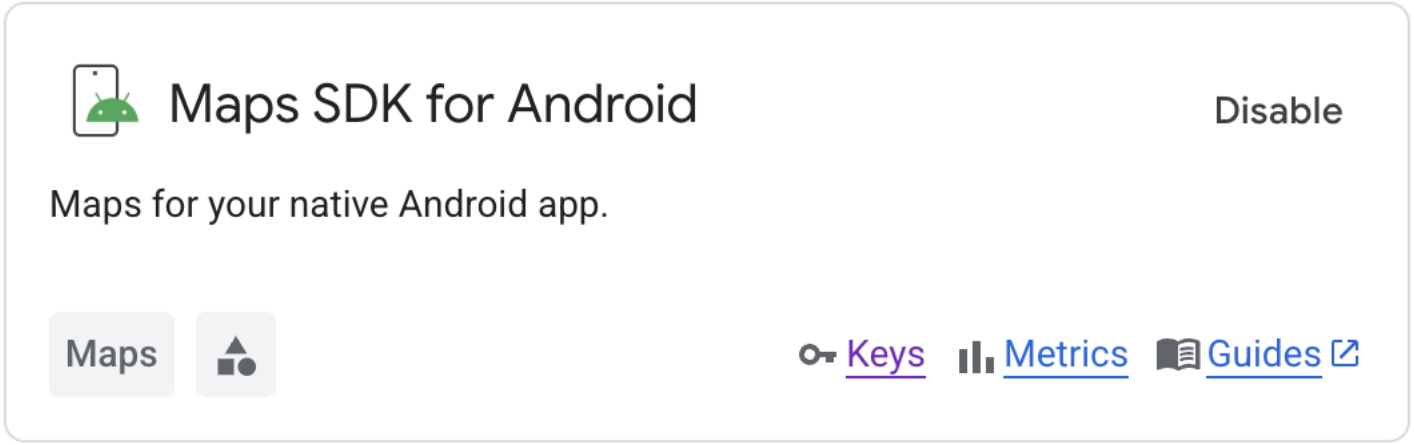
Set up in Cloud Console

- Once you've created a billing account, and new
- From the navigation menu choose Google Maps Platform option -> API&Services



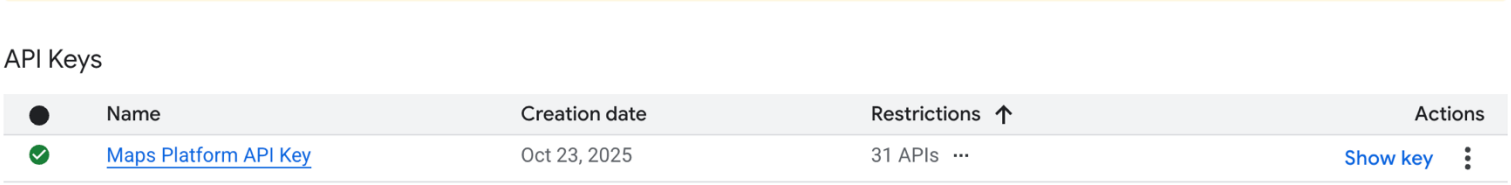
Set up in Cloud Console

- After you click API & Services locate Maps SDK for Android and Enable it



Set up in Cloud Console

- Once enabled you will see your API key which will be used later.



Creating a map api key string

Inside layout -> values -> strings.xml

You need to create a new key and paste it over the “YOUR_KEY_HERE” placeholder

Add the API key to your app

- In Android Studio, open your project-level build.gradle file and add the following code to the dependencies element under buildscript.

```
dependencies {

  implementation libs.appcompat
implementation libs.material
testImplementation libs.junit
androidTestImplementation libs.ext.junit
androidTestImplementation libs.espresso.core

implementation 'com.google.android.gms:play-services-maps:18.1.0'
implementation 'com.google.android.gms:play-services-location:21.0.1'
}
```

- Next, open your module-level build.gradle file and add the following code to the plugins element.

```
id 'com.google.android.libraries.mapsplatform.secrets-grad
le-plugin'
```

- Save the file and sync your project with Gradle.

Reference the API Key in Manifest

- In your AndroidManifest.xml file, go to com.google.android.geo.API_KEY and update the android:value attribute as follows:

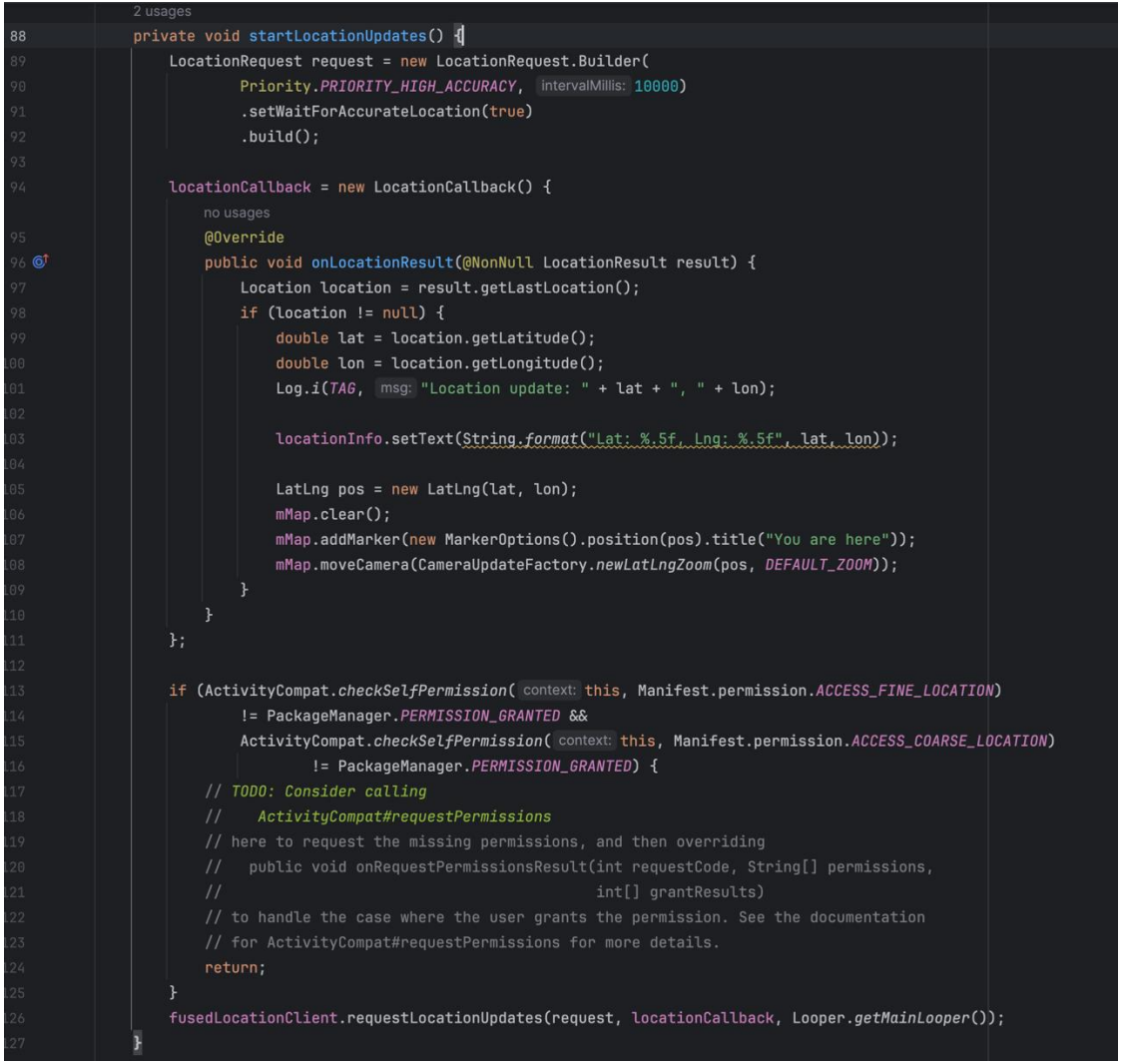
```
<meta-data
android:name="com.google.android.geo.API_KEY"
  android:value="@string/google_maps_key" />
```

Location Acquisition

Registering for Location Updates

- The LocationManager handles registrations for GPS and network location updates
- In order for an object to receive updates from GPS, it must implement the LocationListener interface
- Once the LocationManager is obtained, an object registers for updates by calling requestLocationUpdates (there are multiple versions you can use)
- The arguments passed into the requestLocationUpdates method determine the granularity of location changes that will generate an event
  - send updates that are at least X meters apart
  - send updates at least this far apart in time
  - send updates that have this minimum accuracy

Getting Location Info and updates



```
2 usages
private void startLocationUpdates() {
    LocationRequest request = new LocationRequest.Builder(
            Priority.PRIORITY_HIGH_ACCURACY, intervalMillis: 10000)
            .setWaitForAccurateLocation(true)
            .build();

    locationCallback = new LocationCallback() {
        no usages
        @Override
        public void onLocationResult(@NonNull LocationResult result) {
            Location location = result.getLastLocation();
            if (location != null) {
                double lat = location.getLatitude();
                double lon = location.getLongitude();
                Log.i(TAG, msg: "Location update: " + lat + ", " + lon);

                locationInfo.setText(String.format("Lat: %.5f, Lng: %.5f", lat, lon));

                LatLng pos = new LatLng(lat, lon);
                mMap.clear();
                mMap.addMarker(new MarkerOptions().position(pos).title("You are here"));
                mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(pos, DEFAULT_ZOOM));
            }
        }
    };

    if (ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_FINE_LOCATION)
            != PackageManager.PERMISSION_GRANTED &&
            ActivityCompat.checkSelfPermission( context: 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;
    }
    fusedLocationClient.requestLocationUpdates(request, locationCallback, Looper.getMainLooper());
}
```

Location Providers

- The phone's location can be determined from multiple providers
  - GPS
  - Network
- GPS location updates consume significantly more power than network location updates but are more accurate
  - GPS: 25 seconds * 140mA = 1mAh
  - Network: 2 seconds * 180mA = 0.1mAh
- The provider argument determines which method will be used to get a location for you
- You can also register for the PASSIVE_PROVIDER which only updates you if another app is actively using GPS / Network location

String	GPS_PROVIDER Name of the GPS location provider.
String	NETWORK_PROVIDER Name of the network location provider.
String	PASSIVE_PROVIDER A special location provider for receiving locations without actually initiating a location fix.

Being a Good Citizen…

- It is very important that you unregister your App when you no longer need updates
- For example, you should always unregister your listener when your Activity is paused!
- If you unregister when you pause, you must also reregister when you resume
  - This is true for all sensors!

```
protected void onPause() {
   super.onPause();
   try{
      locationManager.removeUpdates(this);}
   catch(SecurityException e){ Log.e("Err","No Location update permission remover");}
}
protected void onResume() {
   super.onResume();
   if(ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION)== PackageManager.PERMISSION_GRANTED)
      locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
10,Criteria.ACCURACY_COARSE, this);
}
```

MapsActivity.java Overview

MapsActivity.java

```java
package rahul.cse_162_2l.maps;


import android.Manifest;
import android.content.pm.PackageManager;
import android.location.Location;
import android.os.Bundle;
import android.os.Looper;
import android.util.Log;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationCallback;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationResult;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.location.Priority;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.MapsInitializer;
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 section

MapsActivity.java Overview

```java
public class MapsActivity extends AppCompatActivity implements OnMapReadyCallback {

    1 usage
    private static final String TAG = "MAPS_AUTO";
    2 usages
    private static final int LOCATION_PERMISSION_REQUEST_CODE = 101;
    1 usage
    private static final float DEFAULT_ZOOM = 15f;

    8 usages
    private GoogleMap mMap;
    4 usages
    private FusedLocationProviderClient fusedLocationClient;
    4 usages
    private LocationCallback locationCallback;
    2 usages
    private TextView locationInfo;
    1 usage
    private ImageButton myLocationBtn;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);

        locationInfo = findViewById(R.id.location_info);
        myLocationBtn = findViewById(R.id.btn_my_location);
        fusedLocationClient = LocationServices.getFusedLocationProviderClient( activity: this);

        SupportMapFragment mapFragment =
                (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.map);
        if (mapFragment != null) mapFragment.getMapAsync( callback: this);

    }
```

OnCreate() function

MapsActivity.java

Variables, onCreate, onMapReady

```
no usages
@Override
public void onMapReady(@NonNull GoogleMap googleMap) {
    mMap = googleMap;
    MapsInitializer.initialize( context: this, MapsInitializer.Renderer.LATEST,  callback: null);
    mMap.getUiSettings().setZoomControlsEnabled(true);
    mMap.getUiSettings().setMyLocationButtonEnabled(false);
    checkLocationPermission();
}

1 usage
private void checkLocationPermission() {
    if (ContextCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_FINE_LOCATION)
            == PackageManager.PERMISSION_GRANTED) {
        enableLocationFeatures();
    } else {
        ActivityCompat.requestPermissions( activity: this,
                new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
                LOCATION_PERMISSION_REQUEST_CODE);
    }
}

2 usages
private void enableLocationFeatures() {
    if (ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_FINE_LOCATION)
            != PackageManager.PERMISSION_GRANTED) return;
    mMap.setMyLocationEnabled(true);
    startLocationUpdates();
}
```

```
2 usages
88  private void startLocationUpdates() {
89      LocationRequest request = new LocationRequest.Builder(
90              Priority.PRIORITY_HIGH_ACCURACY, intervalMillis: 10000)
91              .setWaitForAccurateLocation(true)
92              .build();
93
94      locationCallback = new LocationCallback() {
            no usages
95          @Override
96          public void onLocationResult(@NonNull LocationResult result) {
97              Location location = result.getLastLocation();
98              if (location != null) {
99                  double lat = location.getLatitude();
100                 double lon = location.getLongitude();
101                 Log.i(TAG, msg: "Location update: " + lat + ", " + lon);
102
103                 locationInfo.setText(String.format("Lat: %.5f, Lng: %.5f", lat, lon));
104
105                 LatLng pos = new LatLng(lat, lon);
106                 mMap.clear();
107                 mMap.addMarker(new MarkerOptions().position(pos).title("You are here"));
108                 mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(pos, DEFAULT_ZOOM));
109             }
110         }
111     };
112
113     if (ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_FINE_LOCATION)
114             != PackageManager.PERMISSION_GRANTED &&
115             ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_COARSE_LOCATION)
116                     != PackageManager.PERMISSION_GRANTED) {
117         // TODO: Consider calling
118         //    ActivityCompat#requestPermissions
119         // here to request the missing permissions, and then overriding
120         //   public void onRequestPermissionsResult(int requestCode, String[] permissions,
121         //                                          int[] grantResults)
122         // to handle the case where the user grants the permission. See the documentation
123         // for ActivityCompat#requestPermissions for more details.
124         return;
125     }
126     fusedLocationClient.requestLocationUpdates(request, locationCallback, Looper.getMainLooper());
127 }
```

MapsActivity.java

- startLocationUpdates() listens for location change

```java
    @Override
    protected void onPause() {
        super.onPause();
        if (fusedLocationClient != null && locationCallback != null) {
            fusedLocationClient.removeLocationUpdates(locationCallback);
        }
    }

    @Override
    protected void onResume() {
        super.onResume();
        if (mMap != null &&
                ContextCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_FINE_LOCATION)
                        == PackageManager.PERMISSION_GRANTED) {
            startLocationUpdates();
        }
    }

    14 usages
    @Override
    public void onRequestPermissionsResult(int requestCode,
                                           @NonNull String[] permissions,
                                           @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        if (requestCode == LOCATION_PERMISSION_REQUEST_CODE) {
            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                enableLocationFeatures();
            } else {
                Toast.makeText( context: this,  text: "Location permission denied", Toast.LENGTH_SHORT).show();
            }
        }
    }
}
```

MapsActivity.java

- OnResume() and OnPause()

activity_maps.xml

```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=".MapsActivity">

    <fragment
        android:id="@+id/map"
        android:name="com.google.android.gms.maps.SupportMapFragment"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ImageButton
        android:id="@+id/btn_my_location"
        android:layout_width="56dp"
        android:layout_height="56dp"
        android:layout_marginEnd="26dp"
        android:layout_marginBottom="85dp"
        android:backgroundTint="@color/purple_500"
        android:src="@android:drawable/ic_menu_mylocation"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:tint="@android:color/white" />
```

activity_maps.xml

```
    <TextView
        android:id="@+id/location_info"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:background="#80000000"
        android:padding="8dp"
        android:text="Locating..."
        android:textColor="@android:color/white"
        android:textSize="14sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java Overview

MainActivity.java

```java
package rahul.cse_162_2l.maps;


import android.content.Intent;
import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Launch the map automatically
        Intent intent = new Intent( packageContext: MainActivity.this, MapsActivity.class)
        startActivity(intent);

        // Optional: close MainActivity so back button doesn't return here
        finish();
    }
}
```

activity_main.xml Overview

activity_main.xml

```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:id="@+id/titleText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Launching Map..."
        android:textSize="22sp"
        android:textStyle="bold"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"/>

</androidx.constraintlayout.widget.ConstraintLayout>
```

AndroidManifest.xml

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

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="Maps"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Maps">
    <!-- Google Maps API Key -->
    <meta-data
        android:name="com.google.android.geo.API_KEY"
        android:value="@string/google_maps_key" />

    <activity android:name=".MapsActivity" />
    <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>
```

AndroidManifest.xml

Build grade (project)

build.gradle (Maps)

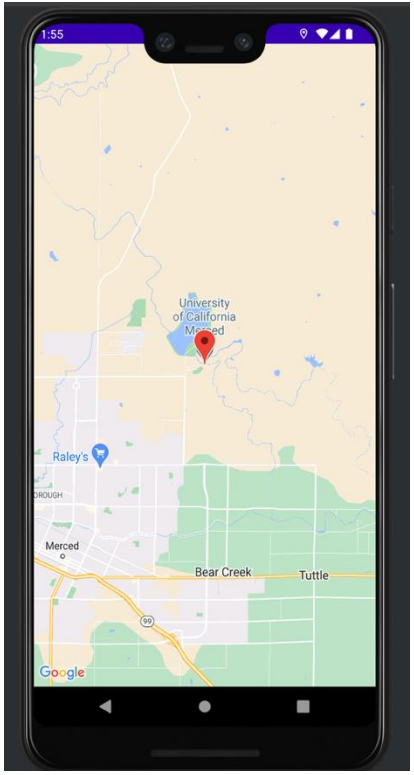
```
1  // Top-level build file where you can add configuration options common to all sub-projects/modules.
2  plugins {
3  alias(libs.plugins.android.application) apply false
4  }
```

build.gradle (:app)

```
plugins {
    alias(libs.plugins.android.application)
}

android {
    namespace 'rahul.cse_162_2l.maps'
    compileSdk 36

    defaultConfig { DefaultConfig it ->
        applicationId "rahul.cse_162_2l.maps"
        minSdk 24
        targetSdk 36
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes { NamedDomainObjectContainer<BuildType> it ->
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro
        }
    }
    compileOptions { CompileOptions it ->
        sourceCompatibility JavaVersion.VERSION_11
        targetCompatibility JavaVersion.VERSION_11
    }
}

dependencies {

    implementation libs.appcompat
    implementation libs.material
    testImplementation libs.junit
    androidTestImplementation libs.ext.junit
    androidTestImplementation libs.espresso.core
    implementation 'com.google.android.gms:play-services-maps:18.1.0'
    implementation 'com.google.android.gms:play-services-location:21.0.1'
}
```

Build grade (app/module)

Final App


1:55
University
of California
Raley's
Merced
Bear Creek
Tuttle
99
Google

Assignments

- Display the map
- Set the map view based on the GPS tracking results