

MainActivity.java

```
package com.example.pract_googlemap;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.content.Intent;
import android.location.Address;
import android.location.Geocoder;
import android.os.Bundle;
import android.os.Message;
import android.os.Parcelable;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.Locale;
import java.util.logging.Handler;

public class Main extends AppCompatActivity {
    private static final String TAG = "Main";
    EditText ed1;
    Button b;
    static List add;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ed1=findViewById(R.id.ed1);
        b=findViewById(R.id.show);
        b.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String s=ed1.getText().toString();
                add= getAddress(s,getApplicationContext());
            }
        });
    }
    public static List getAddress(final String s,final Context context)
    {
        Thread t=new Thread(){
            @Override
            public void run() {
                Geocoder geocoder=new Geocoder(context, Locale.getDefault());
                String result=null;
                try {
                    List add=geocoder.getFromLocationName(s,1);
                    if (add!=null||add.size()>0)
                    {
                        Log.d(TAG, "run: "+"add is not null");
                        Address address= (Address) add.get(0);

                        Log.d(TAG, "onClick: "+address.getLatitude()
+" ,"+address.getLongitude());
                        Bundle bundle=new Bundle();
                        bundle.putParcelableArrayList("list", (ArrayList<? extends
Parcelable>) add);
                        Intent intent=new Intent(context,MapsActivity.class);
```

```

        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);

        intent.putExtra("bundle", bundle);
        context.startActivity(intent);
    }
    else {
        Log.d(TAG, "run: "+"add iss Null");
    }

    } catch (IOException e) {
        e.printStackTrace();
    }
}

};
t.start();
Log.d(TAG, "getAddress: "+"in ");
return add;
}

}

```

activity_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"
    android:orientation="vertical"
    tools:context=".Main">
    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="#5D77D6"
        android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"
        app:popupTheme="@style/ThemeOverlay.AppCompat.Dark"
        android:elevation="4dp"
        app:title="Map"
    />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/toolbar"
        android:layout_marginTop="230dp"
    >
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:layout_centerInParent="true"

            android:orientation="vertical">
            <EditText
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:padding="5dp"
                android:id="@+id/ed1"
                android:hint="search"
                android:textSize="20sp"
                android:layout_marginTop="50dp"
            />
        </LinearLayout>
    </LinearLayout>

```

```

        />
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="15dp"
            android:layout_gravity="center_horizontal"
            android:background="#0C55D8"
            android:text="show"
            android:id="@+id/show"
            android:textColor="#fff"
            android:textSize="15sp" />
    </LinearLayout>

</LinearLayout>

</RelativeLayout>

```

MapActivity.java

```

package com.example.pract_googlemap;

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

import androidx.core.view.MenuItemCompat;
import androidx.fragment.app.FragmentActivity;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.net.Uri;
import android.os.Bundle;
import android.os.Parcelable;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import com.google.android.gms.location.FusedLocationProviderClient;
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.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.maps.model.PointOfInterest;
import com.google.android.gms.maps.model.Polyline;
import com.google.android.gms.maps.model.PolylineOptions;
import com.google.android.gms.tasks.OnCompleteListener;

```

```

import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.Locale;

import static com.example.pract_googlemap.Main.add;

public class MapsActivity extends AppCompatActivity implements OnMapReadyCallback {

    private static final int REQUEST_LOCATION_PERMISSION = 101;
    private GoogleMap mMap;
    private Polyline currentPolyline;
    Button btn;
    ArrayList<LatLng> listPoints;
    private Location mLastLocation, myLocation;
    List list;
    private final LatLng defaultLocation = new LatLng(-33.8523341, 151.2106085);
    private static final int DEFAULT_ZOOM = 12;
    private Location lastKnownLocation;
    FusedLocationProviderClient mFusedLocationClient;
    private static final String TAG = "MapsActivity";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        Bundle bundle=getIntent().getBundleExtra("bundle");
        listPoints = new ArrayList<>();
        btn=findViewById(R.id.btn);
        if (bundle!=null)
        {
            Log.d(TAG, "onCreate: "+"bundle is not null");
            list=bundle.getParcelableArrayList("list");
            if (list!=null||list.size()>0) {
                Log.d(TAG, "run: " + "add is not null");
                Address address= (Address) list.get(0);

                Log.d(TAG, "onClick: "+address.getLatitude()
+" ,"+address.getLongitude());
            }
        }
        else {
            Log.d(TAG, "onCreate: "+"bundle is null");
        }
        // Obtain the SupportMapFragment and get notified when the map is ready to be
        used.
        SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager()
        .findFragmentById(R.id.map);
        mFusedLocationClient= LocationServices.getFusedLocationProviderClient(this);
        assert mapFragment != null;
        mapFragment.getMapAsync(this);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String url;
                if (listPoints.size()>1)
                {
                    url="http://maps.google.com/maps?
saddr="+listPoints.get(0).latitude+", "+listPoints.get(0).longitude+"&daddr="+listPoints.
get(1).latitude+", "+listPoints.get(1).longitude;
                }
                else
                {

```

```

url="google.navigation:q="+listPoints.get(0).latitude+","+listPoints.get(0).longitude+"&
mode=d";
    }

    Intent mapintent=new
Intent("android.intent.action.VIEW", Uri.parse(url));
    mapintent.setPackage("com.google.android.apps.maps");
    if (mapintent.resolveActivity(getPackageManager())!=null)

    {
        startActivity(mapintent);
    }
    }
});

}
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    // getLocation();
    Address address= (Address) list.get(0);
    LatLng latLng=new LatLng(address.getLatitude(),address.getLongitude());
    listPoints.add(latLng);
    Log.d(TAG, "onClick: "+address.getLatitude()+" ,"+address.getLongitude());
    // Add a marker in Sydney and move the camera
    LatLng sydney = new LatLng(address.getLatitude(), address.getLongitude());
    mMap.addMarker(new MarkerOptions().position(sydney));
    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(sydney,12f));

    setMapLongClick(mMap);

    setPoiClick(mMap);
    enableMyLocation();
}
private void enableMyLocation() {
    if (ContextCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_FINE_LOCATION)
        == PackageManager.PERMISSION_GRANTED) {
        mMap.setMyLocationEnabled(true);
    } else {
        ActivityCompat.requestPermissions(this, new String[]
            {Manifest.permission.ACCESS_FINE_LOCATION},
            REQUEST_LOCATION_PERMISSION);
    }
}
@Override
public void onRequestPermissionsResult(int requestCode,
                                       @NonNull String[] permissions,
                                       @NonNull int[] grantResults) {
    // Check if location permissions are granted and if so enable the
    // location data layer.
    switch (requestCode) {
        case REQUEST_LOCATION_PERMISSION:
            if (grantResults.length > 0
                && grantResults[0]
                == PackageManager.PERMISSION_GRANTED) {
                enableMyLocation();
                getLocation();
                break;
            }
    }
}
private void getLocation() {
    if (ActivityCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_FINE_LOCATION)
        != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(this, new String[]

```

```

        {Manifest.permission.ACCESS_FINE_LOCATION},
        REQUEST_LOCATION_PERMISSION);
    } else {
        mFusedLocationClient.getLastLocation().addOnSuccessListener(
            new OnSuccessListener<Location>() {
                @Override
                public void onSuccess(Location location) {
                    if (location != null) {
                        mLastLocation = location;
                        Log.d(TAG, "onSuccess: ");
                    } else {
                        Log.d(TAG, "onSuccess: "+R.string.no_location);
                    }
                }
            });
    }
}

private void setMapLongClick(final GoogleMap map) {
    map.setOnMapLongClickListener(new GoogleMap.OnMapLongClickListener() {
        @Override

        public void onMapLongClick(LatLng latLng) {
            String snippet = String.format(Locale.getDefault(),
                "Lat: %1$.5f, Long: %2$.5f",
                latLng.latitude,
                latLng.longitude);

            map.addMarker(new MarkerOptions()
                .position(latLng)
                .title(getString(R.string.dropped_pin))
                .snippet(snippet));

            if (listPoints.size() == 2) {
                listPoints.clear();
                mMap.clear();
                btn.setVisibility(View.GONE);
            }
            //Save first point select
            listPoints.add(latLng);
            //Create marker
            MarkerOptions markerOptions = new MarkerOptions();
            markerOptions.position(latLng);

            mMap.addMarker(markerOptions);

        }

    });
}

private void setPoiClick(final GoogleMap map) {
    map.setOnPoiClickListener(new GoogleMap.OnPoiClickListener() {
        @Override
        public void onPoiClick(PointOfInterest poi) {
            Marker poiMarker = mMap.addMarker(new MarkerOptions()
                .position(poi.latLng)
                .title(poi.name));
            poiMarker.showInfoWindow();
        }
    });
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu, menu);
}

```

```

MenuItem menuItem=menu.findItem(R.id.search);
SearchView searchView=(SearchView)menuItem.getActionView();
searchView.setQueryHint("search here");
searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
    @Override
    public boolean onQueryTextSubmit(String query) {
        callsearch(query);
        return false;
    }

    @Override
    public boolean onQueryTextChange(String newText) {
        return false;
    }
});

return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Change the map type based on the user's selection.
    switch (item.getItemId()) {
        case R.id.normal_map:
            mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
            return true;
        case R.id.hybrid_map:
            mMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);
            return true;
        case R.id.satellite_map:
            mMap.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
            return true;
        case R.id.terrain_map:
            mMap.setMapType(GoogleMap.MAP_TYPE_TERRAIN);
            return true;
        case R.id.search:
            //callsearch();
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}

```

```

private void callsearch(String str) {

```

```

    Geocoder geocoder=new Geocoder(getApplicationContext(), Locale.getDefault());
    String result=null;
    try {
        List add = geocoder.getFromLocationName(str, 1);
        if (add != null || add.size() > 0) {
            Log.d(TAG, "run: " + "add is not null");
            Address address = (Address) add.get(0);

            LatLng latLng = new LatLng(address.getLatitude(),
address.getLongitude());
            if (listPoints.size() == 2) {

                listPoints.clear();
                mMap.clear();
                // btn.setVisibility(View.GONE);
            }
            //Save first point select
            listPoints.add(latLng);
            //Create marker
            MarkerOptions markerOptions = new MarkerOptions();
            markerOptions.position(latLng);

```

```

        mMap.addMarker(markerOptions);
        Log.d(TAG, "onClick: " + address.getLatitude() + " , " +
address.getLongitude());
        // Add a marker in Sydney and move the camera
        LatLng sydney = new LatLng(address.getLatitude(),
address.getLongitude());
        mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in
Sydney"));
        mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(sydney, 12f));
    } else {
        Log.d(TAG, "run: " + "add iss Null");
    }

} catch (IOException e) {
    e.printStackTrace();
}

}

private void getDeviceLocation() {
    /*
     * Get the best and most recent location of the device, which may be null in
rare
     * cases when a location is not available.
     */
    try {
        //if (locationPermissionGranted)
        {
            Task<Location> locationResult = mFusedLocationClient.getLastLocation();
            locationResult.addOnCompleteListener(this, new
OnCompleteListener<Location>() {
                @Override
                public void onComplete(@NonNull Task<Location> task) {
                    if (task.isSuccessful()) {
                        // Set the map's camera position to the current location of
the device.

                        lastKnownLocation = task.getResult();
                        if (lastKnownLocation != null) {
                            mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(
                                new LatLng(lastKnownLocation.getLatitude(),
                                lastKnownLocation.getLongitude()),
DEFAULT_ZOOM));
                        }
                    } else {
                        Log.d(TAG, "Current location is null. Using defaults.");
                        Log.e(TAG, "Exception: %s", task.getException());
                        mMap.moveCamera(CameraUpdateFactory
                            .newLatLngZoom(defaultLocation, DEFAULT_ZOOM));
                        mMap.getUiSettings().setMyLocationButtonEnabled(false);
                    }
                }
            });
        }
    } catch (SecurityException e) {
        Log.e("Exception: %s", e.getMessage(), e);
    }
}

}

```


activity_maps.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:name="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="match_parent"
    android:orientation="vertical"
    android:layout_height="match_parent"
    tools:context=".MapsActivity">
    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar2"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="#5D77D6"
        android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"
    />
    <fragment xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_weight="1"
        android:id="@+id/map"
        android:name="com.google.android.gms.maps.SupportMapFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
    />

    <Button
        android:layout_marginTop="0dp"
        android:id="@+id/btn"
        android:layout_marginLeft="5dp"
        android:layout_marginRight="5dp"
        android:layout_marginBottom="2dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

        android:text="start"

    />
</LinearLayout>
```

menu.xml

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

    <item android:id="@+id/normal_map"
        android:title="@string/normal_map"
        app:showAsAction="never"/>
    <item android:id="@+id/hybrid_map"
        android:title="@string/hybrid_map"
        app:showAsAction="never"/>
    <item android:id="@+id/satellite_map"
        android:title="@string/satellite_map"
        app:showAsAction="never"/>
    <item android:id="@+id/terrain_map"
```

```
        android:title="@string/terrain_map"
        app:showAsAction="never"/>
<item android:id="@+id/search"
    app:showAsAction="always"
    android:icon="@drawable/search"
    android:title="search"
    app:actionViewClass="androidx.appcompat.widget.SearchView" />
</menu>
```