**Experiment 31**

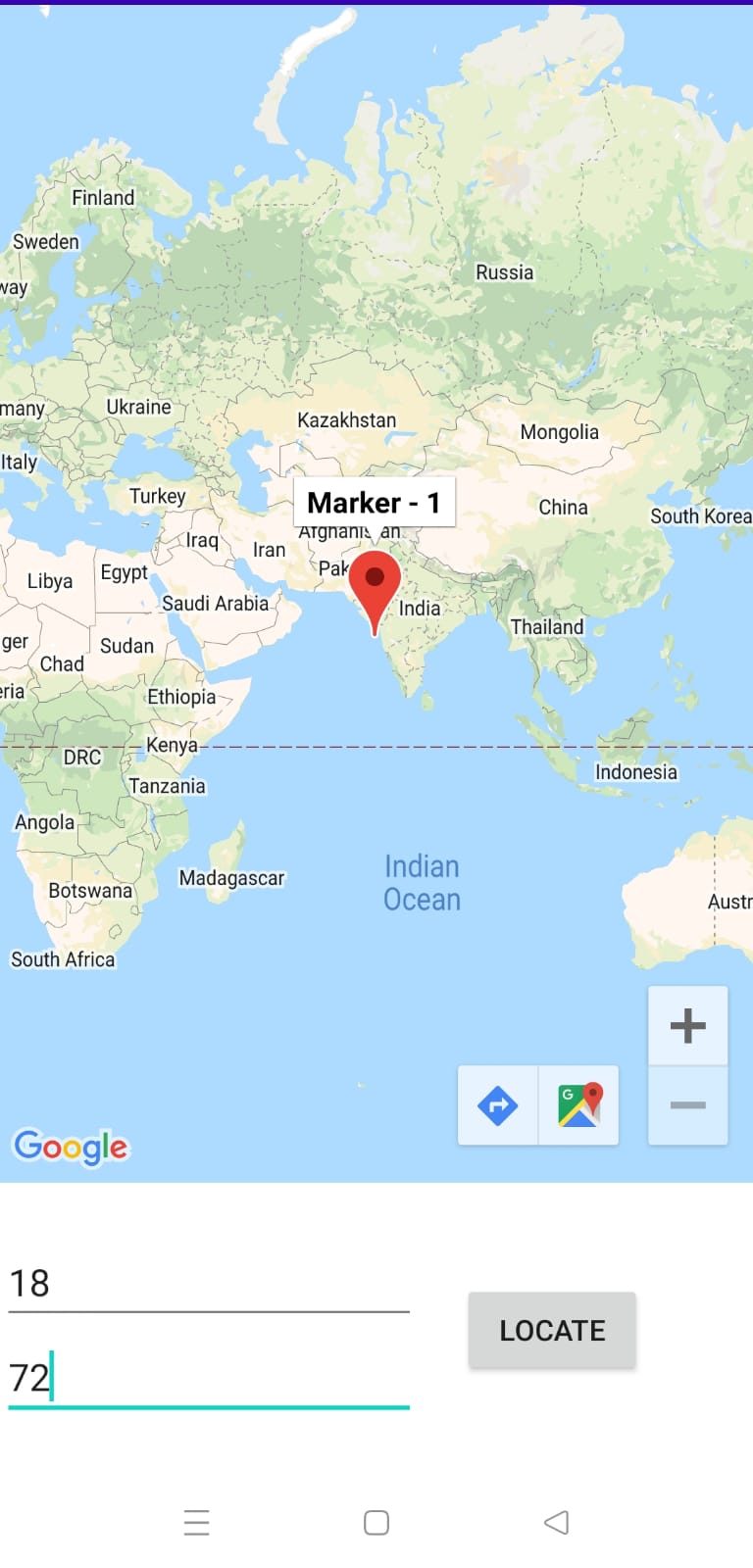
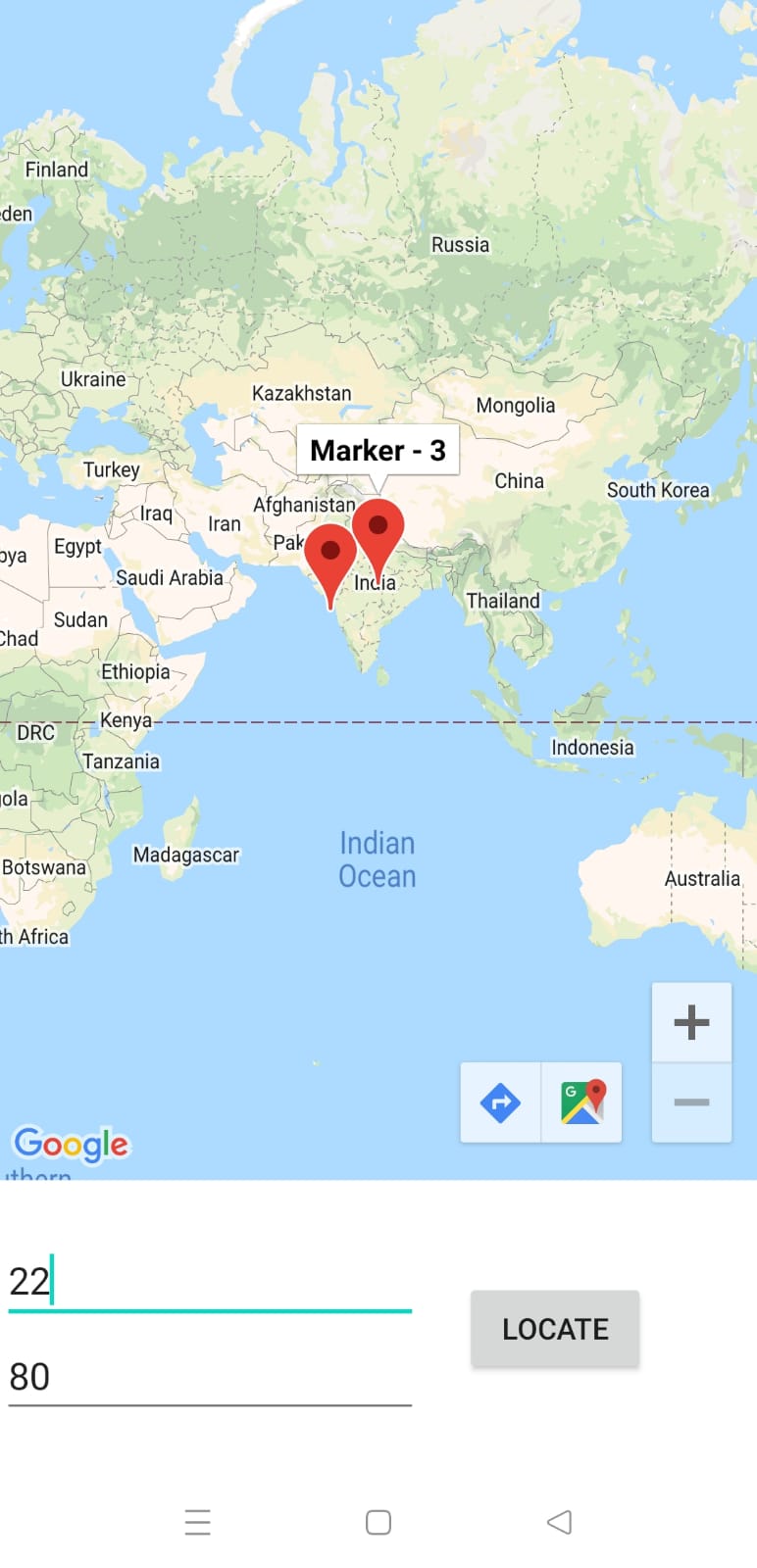
**Q. Navigating to a Specific Location, Set/Unset Zoom Controls**

**ACTIVITY\_MAPS.XML:**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:weightSum="10"  
 xmlns:android="http://schemas.android.com/apk/res/android">  
 <fragment  
 android:layout\_weight="2"  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 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" />  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="8">  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:gravity="center"  
 android:orientation="vertical">  
 <EditText  
 android:id="@+id/ed\_lat"  
 android:layout\_width="200dp"  
 android:layout\_height="wrap\_content"  
 android:hint="Latitude"/>  
 <EditText  
 android:id="@+id/ed\_long"  
 android:layout\_width="200dp"  
 android:layout\_height="wrap\_content"  
 android:hint="Longitude"/>  
 </LinearLayout>  
 <Button  
 android:id="@+id/b1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_margin="20dp"  
 android:text="Locate"/>  
 </LinearLayout>  
</LinearLayout>

**MAPSACTIVITY.JAVA:**

package com.example.test99;  
import androidx.core.app.ActivityCompat;  
import androidx.fragment.app.FragmentActivity;  
import android.Manifest;  
import android.content.pm.PackageManager;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
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.test99.databinding.ActivityMapsBinding;  
public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {  
 private GoogleMap mMap;  
 private ActivityMapsBinding binding;  
 EditText la, lo;  
 int count = 0;  
 Button b1;  
 @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);  
 }  
 /\*\*  
 \* 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;  
 lo = findViewById(R.id.ed\_long);  
 la = findViewById(R.id.ed\_lat);  
 b1 = findViewById(R.id.b1);  
 b1.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 count++;  
 LatLng temp = new LatLng(Float.parseFloat(la.getText().toString()), Float.parseFloat(lo.getText().toString()));  
 mMap.addMarker(new MarkerOptions().position(temp).title("Marker - " + count));  
 mMap.moveCamera(CameraUpdateFactory.newLatLng(temp));  
 }  
 });  
 mMap.getUiSettings().setZoomControlsEnabled(true);  
 //displays + - option used to zoom in & out on map (default - false)  
 mMap.getUiSettings().setZoomGesturesEnabled(false);  
 //sets the pinch in & out gestures (default - true)  
 mMap.getUiSettings().setCompassEnabled(false);  
 //sets the compass gesture, visible when map is rotated (default - true)  
  
 }  
}

**OUTPUT:**

**Q. Geocoding & Reverse Geocoding**

**ACTIVITY\_MAPS.XML:**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout android:layout\_width="match\_parent"  
android:layout\_height="match\_parent"  
android:orientation="vertical"  
android:weightSum="10"  
xmlns:android="http://schemas.android.com/apk/res/android">  
 <fragment xmlns:android="http://schemas.android.com/apk/res/android"  
 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\_weight="2"  
 android:layout\_height="match\_parent"  
 tools:context=".MapsActivity" />  
<LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:gravity="center">  
 <EditText  
 android:id="@+id/ed\_ad"  
 android:layout\_width="220dp"  
 android:layout\_height="wrap\_content"  
 android:hint="Address"/>  
 <Button  
 android:id="@+id/b1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_margin="20dp"  
 android:text="FIND"/>  
 </LinearLayout>  
 <TextView  
 android:id="@+id/t1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text=""/>  
</LinearLayout>  
  
</LinearLayout>

**MAPSACTIVITY.JAVA:**

package com.example.test999;  
  
import androidx.annotation.NonNull;  
import androidx.fragment.app.FragmentActivity;  
import android.location.Address;  
import android.location.Geocoder;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
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.test999.databinding.ActivityMapsBinding;  
import java.io.IOException;  
import java.util.List;  
  
public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {  
 private GoogleMap mMap;  
 private ActivityMapsBinding binding;  
 private Geocoder geo;  
 EditText ed\_ad;  
 Button b1;  
 TextView t1;  
 @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);  
 geo = new Geocoder(this);  
 ed\_ad = findViewById(R.id.ed\_ad);  
 b1 = findViewById(R.id.b1);  
 t1 = findViewById(R.id.t1);  
 }  
  
 @Override  
 public void onMapReady(GoogleMap googleMap) {  
 mMap = googleMap;  
 //Geocoding to find lat,long from address  
 b1.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 try {  
 List<Address> addresses = geo.getFromLocationName(ed\_ad.getText().toString(), 1);  
 if(addresses.size()>0){  
 Address ads = addresses.get(0);  
 LatLng latlo = new LatLng(ads.getLatitude(), ads.getLongitude());  
 t1.setText("Latitude = "+ads.getLatitude()+" Longitude = "+ads.getLongitude());  
 mMap.addMarker(new MarkerOptions().position(latlo).title(ads.getLocality()));  
 mMap.moveCamera(CameraUpdateFactory.newLatLng(latlo));  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 });  
  
 //REVERSE GEO-CODING (to find address from lat,long)  
 mMap.setOnMapLongClickListener(new GoogleMap.OnMapLongClickListener() {  
 @Override  
 public void onMapLongClick(@NonNull LatLng latLng) {  
 Toast.makeText(getApplicationContext(), latLng.toString(), Toast.LENGTH\_SHORT).show();  
 try {  
 List<Address> addresses = geo.getFromLocation(latLng.latitude, latLng.longitude, 1);  
 if (addresses.size()>0){  
 Address ads = addresses.get(0);  
 String txt = ads.getAddressLine(0);  
 mMap.addMarker(new MarkerOptions().position(latLng).title(txt));  
 t1.setText(txt);  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 });  
  
 }  
}

**OUTPUT:**

**Map

Description automatically generatedMap

Description automatically generated**

**Map

Description automatically generatedMap

Description automatically generated**