

Sabarivasan V  
205001085  
CSE - B

### **Ex. No. 6 Android Application for Location Tracking**

**Aim:** 1. Develop an Android Application that uses Geographical Positioning System (GPS) to display the user's current location in terms of Latitude and Longitude. 2. Develop a mobile app to display the Geo location of a given place.

**Layouts Use:** None. TextViews.

**Code:**

**MainActivity.java:**

```
package com.example.ex6;

import android.Manifest;
import android.annotation.SuppressLint;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.Looper;
import android.provider.Settings;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
```

```
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

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

import android.location.Address;
import android.location.Geocoder;
import android.widget.Toast;
import java.io.IOException;
import java.util.List;
```

```
public class MainActivity extends AppCompatActivity {

    // initializing
    // FusedLocationProviderClient
    // object
    FusedLocationProviderClient mFusedLocationClient;

    // Initializing other items
    // from layout file
    TextView latitudeTextView, longitudeTextView;
    int PERMISSION_ID = 44;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```

setContentView(R.layout.activity_main);

latitudeTextView = findViewById(R.id.latTextView);
longitudeTextView = findViewById(R.id.lonTextView);

mFusedLocationClient =
LocationServices.getFusedLocationProviderClient(this);

// method to get the location
getLastLocation();

Button display = findViewById(R.id.dis);
display.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String loc = ((EditText)
findViewById(R.id.loc)).getText().toString();
        Log.d("debug",loc);
        getLatLngForPlace(loc);
    }
});
}

@SuppressWarnings("MissingPermission")
private void getLastLocation() {
    // check if permissions are given
    if (checkPermissions()) {

        // check if location is enabled
        if (isLocationEnabled()) {

            // getting last
            // location from
            // FusedLocationClient

```

```

        // object

mFusedLocationClient.getLastLocation().addOnCompleteListener(new
OnCompleteListener<Location>() {
    @Override
    public void onComplete(@NonNull Task<Location> task) {
        Location location = task.getResult();
        if (location == null) {
            requestNewLocationData();
        } else {
            latitudeTextView.setText(location.getLatitude() + "");
            longitTextView.setText(location.getLongitude() + "");
        }
    }
});
    } else {
        Toast.makeText(this, "Please turn on" + " your location...",
Toast.LENGTH_LONG).show();
        Intent intent = new
Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
        startActivity(intent);
    }
} else {
    // if permissions aren't available,
    // request for permissions
    requestPermissions();
}
}

@SuppressLint("MissingPermission")
private void requestNewLocationData() {

    // Initializing LocationRequest
    // object with appropriate methods
    LocationRequest mLocationRequest = new LocationRequest();

```

```
mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
```

```
    mLocationRequest.setInterval(5);
```

```
    mLocationRequest.setFastestInterval(0);
```

```
    mLocationRequest.setNumUpdates(1);
```

```
    // setting LocationRequest
```

```
    // on FusedLocationClient
```

```
    mFusedLocationClient =
```

```
    LocationServices.getFusedLocationProviderClient(this);
```

```
    mFusedLocationClient.requestLocationUpdates(mLocationRequest,
    mLocationCallback, Looper.myLooper());
```

```
    }
```

```
    private LocationCallback mLocationCallback = new LocationCallback() {
```

```
        @Override
```

```
        public void onLocationResult(LocationResult locationResult) {
```

```
            Location mLastLocation = locationResult.getLastLocation();
```

```
            latitudeTextView.setText("Latitude: " + mLastLocation.getLatitude()
+ "");
```

```
            longitTextView.setText("Longitude: " +
mLastLocation.getLongitude() + "");
```

```
        }
```

```
    };
```

```
    // method to check for permissions
```

```
    private boolean checkPermissions() {
```

```
        return ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) ==
```

```
PackageManager.PERMISSION_GRANTED &&
```

```
ActivityCompat.checkSelfPermission(this,
```

```
Manifest.permission.ACCESS_FINE_LOCATION) ==
```

```
PackageManager.PERMISSION_GRANTED;
```

```

        // If we want background location
        // on Android 10.0 and higher,
        // use:
        // ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_BACKGROUND_LOCATION) ==
PackageManager.PERMISSION_GRANTED
    }

    // method to request for permissions
    private void requestPermissions() {
        ActivityCompat.requestPermissions(this, new String[]{
            Manifest.permission.ACCESS_COARSE_LOCATION,
            Manifest.permission.ACCESS_FINE_LOCATION},
PERMISSION_ID);
    }

    // method to check
    // if location is enabled
    private boolean isLocationEnabled() {
        LocationManager locationManager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);
        return
locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER) ||
locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDE
R);
    }

    // If everything is alright then
    @Override
    public void
onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
    }

```

```

        if (requestCode == PERMISSION_ID) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                getLastLocation();
            }
        }
    }
}

```

@Override

```

public void onResume() {
    super.onResume();
    if (checkPermissions()) {
        getLastLocation();
    }
}

```

```

private void getLatLngForPlace(String placeName) {
    Geocoder geocoder = new Geocoder(this);

```

```

    try {
        List<Address> addresses =
geocoder.getFromLocationName(placeName, 1);
        if (addresses != null && !addresses.isEmpty()) {
            Address address = addresses.get(0);
            double latitude = address.getLatitude();
            double longitude = address.getLongitude();
            latitudeTextView.setText("" + latitude);
            longitTextView.setText("" + longitude);
        } else {
            // Handle the case where the place name couldn't be geocoded
            Toast.makeText(this, "Place not found",
Toast.LENGTH_SHORT).show();
        }
    } catch (IOException e) {

```

```

        e.printStackTrace();
    }
}
}

```

### **Activity main.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/textView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="48dp"
        android:fontFamily="sans-serif-black"
        android:text="Enter location:"
        android:textSize="24sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/lonTextView"
        tools:ignore="MissingConstraints" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"

```



```
android:layout_height="wrap_content"
android:layout_marginTop="124dp"
android:fontFamily="sans-serif-black"
android:text="Latitude:"
android:textSize="24sp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.4"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
tools:ignore="MissingConstraints" />
```

<TextView

```
android:id="@+id/latTextView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="36dp"
android:text=""
android:textSize="24sp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.406"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView"
tools:ignore="MissingConstraints" />
```

<TextView

```
android:id="@+id/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="24dp"
android:fontFamily="sans-serif-black"
android:text="Longitude:"
android:textSize="24sp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.427"
app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toBottomOf="@+id/latTextView"  
tools:ignore="MissingConstraints" />
```

```
<TextView
```

```
    android:id="@+id/lonTextView"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="48dp"  
    android:text=""  
    android:textSize="24sp"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.44"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/textView2"  
    tools:ignore="MissingConstraints" />
```

```
<Button
```

```
    android:id="@+id/dis"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Display"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.498"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/textView4"  
    app:layout_constraintVertical_bias="0.639" />
```

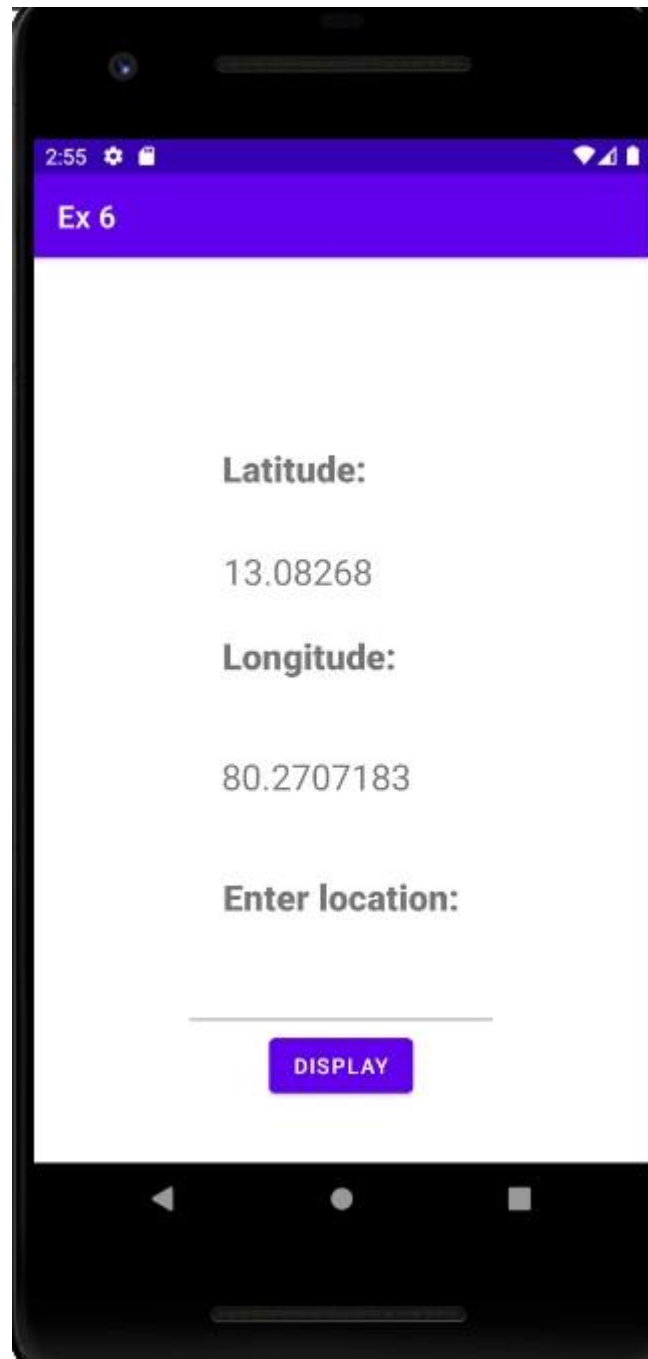
```
<EditText
```

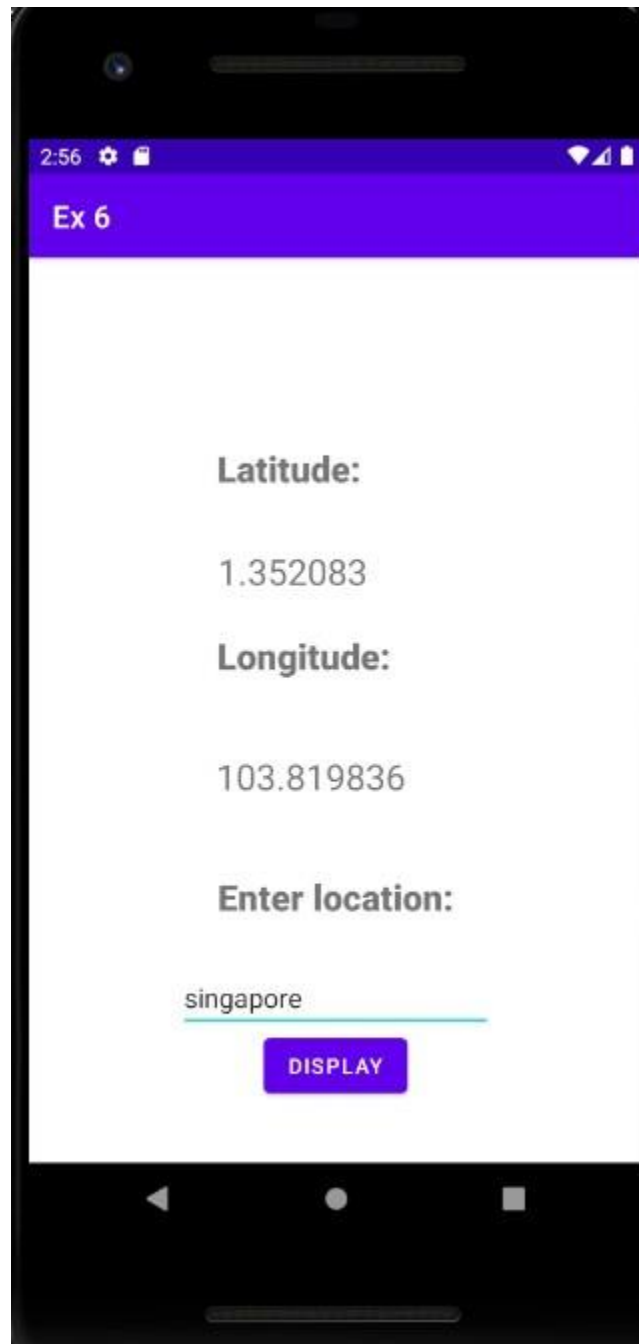
```
    android:id="@+id/loc"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:ems="10"  
    android:inputType="textPersonName"  
    app:layout_constraintBottom_toBottomOf="parent"
```

```
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.497"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toBottomOf="@+id/textView4"  
app:layout_constraintVertical_bias="0.25" />
```

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

**Output:**





**Best Practices:**

- Used meaningful ids
- Aligned the textviews

**Learning Outcomes:**

- Learnt to retrieve coordinates of current locations
- Learnt to retrieve coordinates of a different location using geocoder