Assignment 7

- Title: GPS & GoogleMaps

- Problem Statement:
  Design a mobile app using google map & GPS to trace the location.

- Objective:
  - Develop an application for tracing the location through GPS.
  - use google maps to show the current location.

- Software & Hardware Packages:
  - Java JDK 8
  - Android Studio V4.x
  - 64 bit OS : Ubuntu 18.04
  - Processor : intel i5
  - 8GB RAM , HDD Storage
  - i/o devices.

- Theory:
  GPS:
  Most current smart phones are equipped with Global Positioning System, that allows technology to neet different needs of the user.

- Google Maps:
  It is a web based service that provides detailed info. about geographi regions & sites around the world.

  Google Maps API was launched for developers to incorporate maps functionality into their websites without any charge.

Current location:-
Accessing location of device needs extra permissions, this needs to be put in manifests file.

Syntax:
```
<uses-permission androd:name=
    "android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android.name=
    "android.permission.INTERNET"/>
```

Location manger & listner
```
location Manager = (LocationManager)getSystemServi
                         context.LOCATION_SERVICE);
location Manager.requestLocationupdates(
  locationManager.GPS_PROVIDER, <time_in_ms>,
  <distance_in_m>, this);
```

```
location listner= newLocation Listner() {
    @override
     public void onLocationChanged (Location
                                    Location) {
          double lat = location.getLatitude();
          double lng = location.getlongitude();
          //do Something
     }
}
```

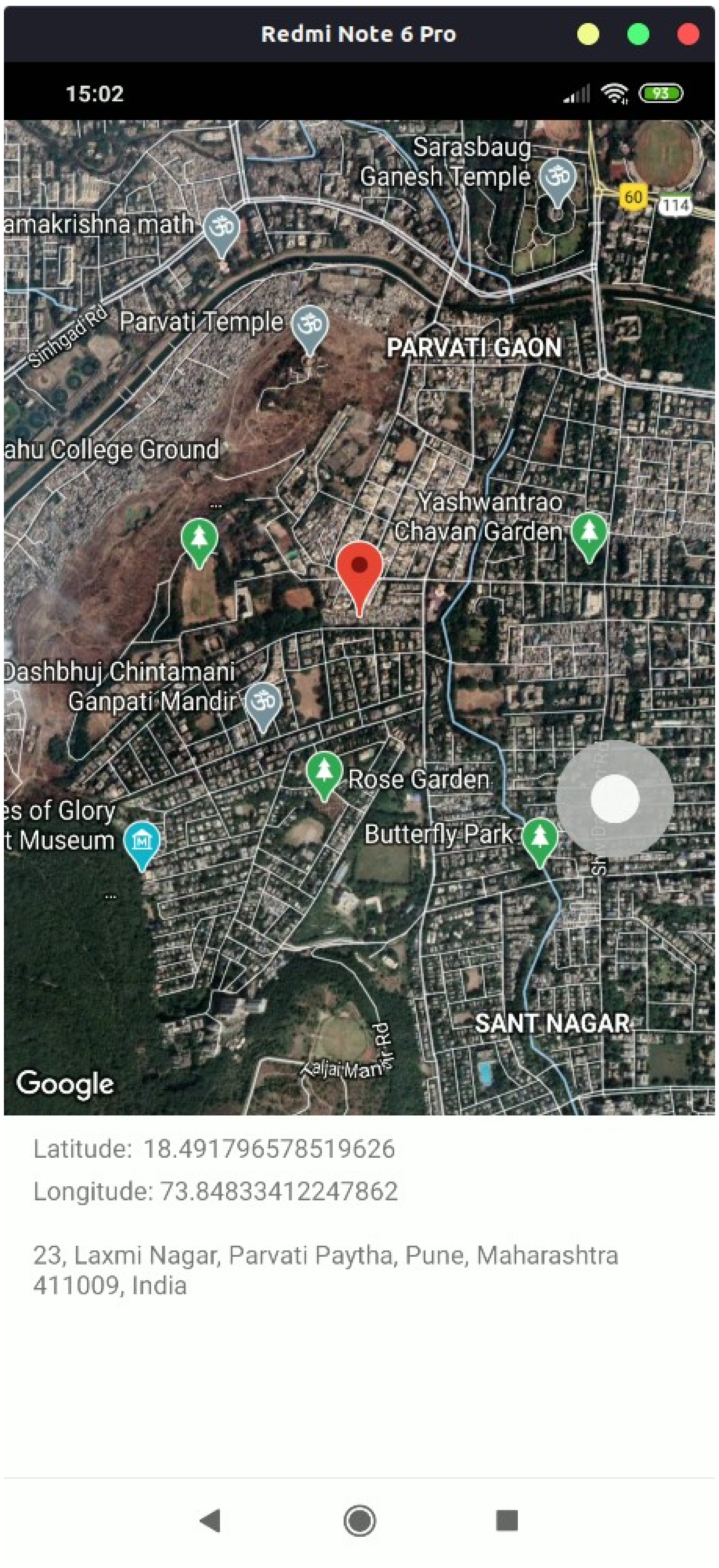
Geo Coder:
A class for handling geo coding &
reverse geo coding
Constructor:
GeoCoder(Context context, Locale.locale)
methods

```
List <Address> getFromLocation(
                    double latitude,
                    double longitude,
                    int MaxResults
               )

List ( Address) getFromLocationName(
               String location Name,
               int Max-Results,
               double lowerleftlatitude,
               double lowerleftlongitude,
               double upperrightlatitude,
               double upper right longitude
          )
```

Conclusion:
In this assignment we used GPS service for tracking the current location ~~&~~ of user & Google Maps API for displaying the location on the map with Geocoding the current coordinates.

OutPut



Latitude: 18.491796578519626

Longitude: 73.84833412247862

23, Laxmi Nagar, Parvati Paytha, Pune, Maharashtra 411009, India

```java
package com.mrunal.googlemap;

import androidx.annotation.NonNull;
import androidx.core.app.ActivityCompat;
import androidx.fragment.app.FragmentActivity;

import android.Manifest;
import android.annotation.SuppressLint;
import android.app.Activity;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;

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

public class MapsActivity extends FragmentActivity implements OnMapReadyCallback{

    private GoogleMap mMap;
    LocationManager locationManager;
    private static final int REQUEST_LOCATION_PERMISSION = 1;
    Marker marker;
    LocationListener locationListener;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        // 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);
        locationManager = (LocationManager) getSystemService(LOCATION_SERVICE);
        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{

            locationListener = new LocationListener() {
                @Override
                public void onLocationChanged(Location location) {
                    double latitude = location.getLatitude();
                    double longitude = location.getLongitude();
                    //get the location name from latitude and longitude
                    Geocoder geocoder = new Geocoder(getApplicationContext());
                    try {
                        List<Address> addresses =
                                geocoder.getFromLocation(latitude, longitude, 1);
                        String result = addresses.get(0).getLocality()+":";
                        result += addresses.get(0).getCountryName();
                        LatLng latLng = new LatLng(latitude, longitude);
                        if (marker != null){
                            marker.remove();
                            marker = mMap.addMarker(new
MarkerOptions().position(latLng).title(result));
                            mMap.setMaxZoomPreference(20);
                            mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(latLng,
12.0f));
                        }
                        else{
                            marker = mMap.addMarker(new
MarkerOptions().position(latLng).title(result));
                            mMap.setMaxZoomPreference(20);
                            mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(latLng,
21.0f));
                        }



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

                @Override
                public void onStatusChanged(String provider, int status, Bundle extras) {

                }

                @Override
                public void onProviderEnabled(String provider) {

                }

                @Override
                public void onProviderDisabled(String provider) {

                }
            };
            locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0,
locationListener);
            locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0,
locationListener);
        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
@NonNull int[] grantResults) {
        if (ActivityCompat.checkSelfPermission(this,
                Manifest.permission.ACCESS_FINE_LOCATION)
                == PackageManager.PERMISSION_GRANTED){

            locationListener = new LocationListener() {
                @Override
                public void onLocationChanged(Location location) {
                    double latitude = location.getLatitude();
                    double longitude = location.getLongitude();
                    //get the location name from latitude and longitude
                    Geocoder geocoder = new Geocoder(getApplicationContext());
                    try {
                        List<Address> addresses =
                                geocoder.getFromLocation(latitude, longitude, 1);
                        String result = addresses.get(0).getLocality()+":";
                        result += addresses.get(0).getCountryName();
                        LatLng latLng = new LatLng(latitude, longitude);
                        if (marker != null){
                            marker.remove();
                            marker = mMap.addMarker(new
MarkerOptions().position(latLng).title(result));
                            mMap.setMaxZoomPreference(20);
                            mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(latLng,
12.0f));
                        }
                        else{
                            marker = mMap.addMarker(new
MarkerOptions().position(latLng).title(result));
                            mMap.setMaxZoomPreference(20);
                            mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(latLng,
21.0f));
                        }



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

                @Override
                public void onStatusChanged(String provider, int status, Bundle extras) {

                }

                @Override
                public void onProviderEnabled(String provider) {

                }

                @Override
                public void onProviderDisabled(String provider) {

                }
            };
            locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0,
locationListener);
            locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0,
locationListener);
        }
    }

    /**
     * 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;

        // 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));
    }

    @Override
    protected void onStop() {
        super.onStop();
        locationManager.removeUpdates(locationListener);
    }
}
```

```
<?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" />
```