Exercise 6 – Finding Geo-coordinates of a Location and Reverse Geocoding

Objective

a) Develop an android application to find the latitude and longitude of current location and the selected location in a google map using anyone of the below options:

1) Location Manager
2) Network Provider
3) GPS Provider

b) Also perform Reverse Geocoding i.e. given a latitude and longitude of a location, app should display the location name or given a location name it should display the latitude and longitude of that place.

Algorithm

1. Initialize the Project:
   - Open Android Studio and create a new Android Project.

2. Design the UI:
   - Create a layout with a Google Map view for displaying the map.
   - Add UI elements for selecting location and displaying latitude/longitude.

3. Request Location Permission:
   - In the manifest file, request location permissions.

4. Handle Location Updates:
   - Use either Location Manager, Network Provider, or GPS Provider to get the current location.
   - Set up listeners to receive location updates.

5. Display Current Location:
   - Show the current location on the Google Map.

6. Select a Location:
   - Implement functionality to allow the user to select a location on the map.

7. Reverse Geocoding:
   - Implement reverse geocoding to convert latitude and longitude to a human-readable location name.

8. Geocoding:
   - Implement geocoding to convert a location name to latitude and longitude.

9. Display Results:
   - Display the results (latitude, longitude, or location name) to the user.

10. Stop Location Updates:
    - Stop location updates when the app is paused or closed.

Features used

Main Features:
- Google Map integration for location display.
- Choice of Location Manager, Network Provider, or GPS Provider for location updates.
- Reverse geocoding to convert coordinates to location names.
- Geocoding to convert location names to coordinates.
- User-friendly UI for selecting locations and displaying results.

Source Code

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:padding="16dp"
tools:context=".MainActivity">
<!-- Previous UI elements -->
<Button
android:id="@+id/getLocationButton"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_centerHorizontal="true"
android:text="Get Location" />
<TextView
android:id="@+id/latitudeTextView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@id/getLocationButton"
android:layout_centerHorizontal="true"
android:layout_marginTop="16dp"
android:text="Latitude: "
android:textSize="18sp" />
<TextView
android:id="@+id/longitudeTextView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@id/latitudeTextView"
android:layout_centerHorizontal="true"
android:layout_marginTop="8dp"
android:text="Longitude: "
android:textSize="18sp" />
<!-- New UI elements for search -->
<EditText
android:id="@+id/locationNameEditText"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_below="@id/longitudeTextView" android:layout_marginTop="16dp"
android:hint="Enter Location Name" />
<Button
android:id="@+id/searchLocationButton"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
```

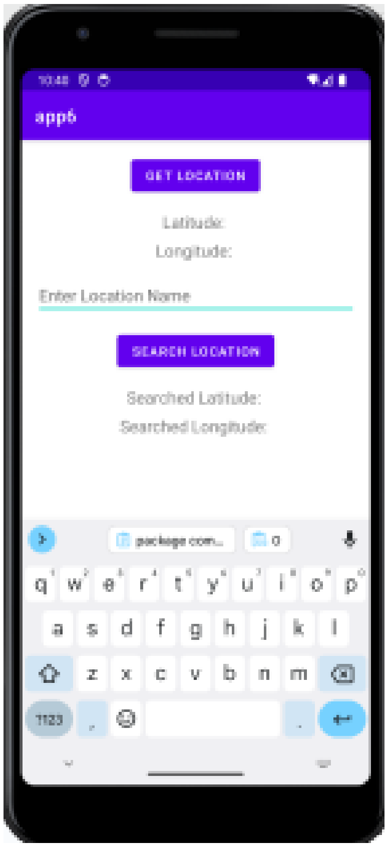
```
android:layout_below="@id/locationNameEditText"
android:layout_centerHorizontal="true"
android:layout_marginTop="16dp"
android:text="Search Location" />
<TextView
android:id="@+id/searchLatitudeTextView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@id/searchLocationButton"
android:layout_centerHorizontal="true"
android:layout_marginTop="16dp"
android:text="Searched Latitude: "
android:textSize="18sp" />
<TextView
android:id="@+id/searchLongitudeTextView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"

android:layout_below="@id/searchLatitudeTextView"
android:layout_centerHorizontal="true"
android:layout_marginTop="8dp"
android:text="Searched Longitude: "
android:textSize="18sp" />
</RelativeLayout>
```

MainActivity.java

```
package com.example.app6;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
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 androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import com.example.app6.R;
import com.google.android.gms.location.FusedLocationProviderClient; import
com.google.android.gms.location.LocationServices;
import com.google.android.gms.tasks.OnSuccessListener;
import java.io.IOException;
import java.util.List;
import java.util.Locale;
public class MainActivity extends AppCompatActivity {
private static final int LOCATION_PERMISSION_REQUEST = 1;
private FusedLocationProviderClient fusedLocationProviderClient; private
EditText locationNameEditText;
private Button searchLocationButton;
private TextView latitudeTextView, longitudeTextView, searchLatitudeTextView,
searchLongitudeTextView;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
fusedLocationProviderClient =
LocationServices.getFusedLocationProviderClient(this);
locationNameEditText = findViewById(R.id.locationNameEditText);
searchLocationButton = findViewById(R.id.searchLocationButton);
latitudeTextView = findViewById(R.id.latitudeTextView);
longitudeTextView = findViewById(R.id.longitudeTextView);
searchLatitudeTextView = findViewById(R.id.searchLatitudeTextView);
searchLongitudeTextView = findViewById(R.id.searchLongitudeTextView);
findViewById(R.id.getLocationButton).setOnClickListener(new
View.OnClickListener() {
@Override
public void onClick(View view) {
getLocation();
}
});
searchLocationButton.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
searchLocation();
}
});
}
private void getLocation() {
```

```
if (ContextCompat.checkSelfPermission(this,
android.Manifest.permission.ACCESS_FINE_LOCATION) ==
PackageManager.PERMISSION_GRANTED) {
fusedLocationProviderClient.getLastLocation().addOnSuccessListener(new
OnSuccessListener<Location>() {
@Override
public void onSuccess(Location location) {
if (location != null) {
double latitude = location.getLatitude();
double longitude = location.getLongitude();
latitudeTextView.setText("Latitude: " + latitude);

longitudeTextView.setText("Longitude: " + longitude);
} else {
showToast("Location not available");
}
}
});
} else {
ActivityCompat.requestPermissions(this, new
String[]{android.Manifest.permission.ACCESS_FINE_LOCATION},
LOCATION_PERMISSION_REQUEST);
}
}
private void searchLocation() {
String locationName = locationNameEditText.getText().toString().trim();
if (!locationName.isEmpty()) {
Geocoder geocoder = new Geocoder(this, Locale.getDefault());
try {
List<Address> addresses =
geocoder.getFromLocationName(locationName, 1);
if (addresses != null && !addresses.isEmpty()) {
Address address = addresses.get(0);
double latitude = address.getLatitude();
double longitude = address.getLongitude();
searchLatitudeTextView.setText("Searched Latitude: " +
latitude);
searchLongitudeTextView.setText("Searched Longitude: " +
longitude);
} else {
showToast("Location not found");
}
} catch (IOException e) {
e.printStackTrace();
showToast("Geocoding error");
}
} else {
showToast("Please enter a location name");
}
}
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
super.onRequestPermissionsResult(requestCode, permissions, grantResults); if
(requestCode == LOCATION_PERMISSION_REQUEST) {
if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {

getLocation();
} else {
showToast("Location permission denied");
}
}
}
private void showToast(String message) {
Toast.makeText(this, message, Toast.LENGTH_SHORT).show();
}
```

Output Screenshots



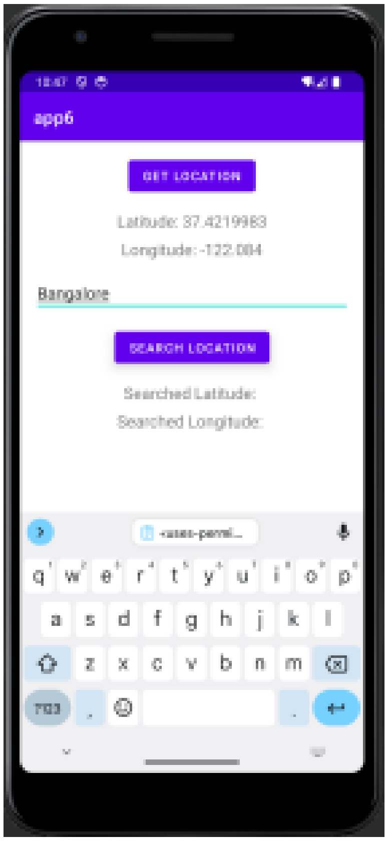
Latitude:

Longitude:

Enter Location Name

Searched Latitude:

Searched Longitude:



Latitude: 37.4219983

Longitude: -122.084

Bangalore

Searched Latitude:

Searched Longitude:

Result

Thus geolocation was implemented to find latitude and longitude of a give location

Best Practices

1. User friendly design
2. Readable layouts
3. Modularity
4. Used apt names for xml and java files.
5. Set padding and margins for dynamically added elements

Learning outcomes

- An android application to find the latitude and longitude of a selected location was implemented.
- Geocoding and Reverse geocoding was implemented.