Department of
Computer Science and Engineering

Title: Development of an application that uses GPS location information

Mobile Application Development

CSE 402

Green University of Bangladesh

1 Objective(s)

- In android application, we can get user's location information with proper permissions granted. This experiment is designed to implement location manager in android development environment.

2 Problem analysis

A location may consist of a latitude, longitude, timestamp, and other information such as bearing, altitude and velocity.All locations generated through LocationManager are guaranteed to have a valid latitude, longitude, and timestamp (both UTC time and elapsed real-time since boot). In this experiment, we will create an application which will use location manager class and provide the user all the necessary location information when asked using internet.

3 Implementation of Creation of database and table along with insertion operation on table

3.1 XML implementation

```
<LinearLayout 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=".MainActivity">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Location"
        android:textSize="20sp"
        android:id="@+id/show_location"
        />
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Get Location"
        android:id="@+id/getLocation"
        />

</LinearLayout>
```

3.2 Add Permissions in the Manifest file

- Allows the API to use WiFi or mobile cell data (or both) to determine the device's location. Provides an estimate of the device's location, to within about 1 mile (1.6 km)

```
    <uses-permission android:name="android.permission.
       ACCESS_COARSE_LOCATION">
    </uses-permission>
```

- Provides an estimate of the device's location that is as accurate as possible, usually within about 50 meters and sometimes as accurate as within a few meters or better.

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

- Provides Internet access to the application

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

3.3 Java File for Executing the Overall Operations

```
package com.example.locationinfo;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

import android.Manifest;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationManager;
import android.os.Bundle;
import android.provider.Settings;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    private static  final int REQUEST_LOCATION=1;

    Button getlocationBtn;
    TextView showLocationTxt;

    LocationManager locationManager;
    String latitude,longitude;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        showLocationTxt=findViewById(R.id.show_location);
        getlocationBtn=findViewById(R.id.getLocation);

        //Add permission
        ActivityCompat.requestPermissions(this,
                new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
                   REQUEST_LOCATION);
 getlocationBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                locationManager=(LocationManager) getSystemService(Context.
                   LOCATION_SERVICE);
                //Check gps is enable or not
                if (!locationManager.isProviderEnabled(LocationManager.
                   GPS_PROVIDER))
                {
                    //Write Function To enable gps
```

```
49
50                      OnGPS();
51                  }
52                  else
53                  {
54                      //GPS is already On then
55                      getLocation();
56                  }
57              }
58
59
60          });
61
62      }
```

3.4 Creation of getLocation function outside onCreate

```
1   private void getLocation() {
2           //Check Permissions again
3
4           if (ActivityCompat.checkSelfPermission(MainActivity.this,Manifest.
                permission.ACCESS_FINE_LOCATION)
5                   != PackageManager.PERMISSION_GRANTED &&
6                   ActivityCompat.checkSelfPermission(MainActivity.this, Manifest.
                        permission.ACCESS_COARSE_LOCATION)
7                           != PackageManager.PERMISSION_GRANTED){
8
9               ActivityCompat.requestPermissions(this,new String[]
10                          {Manifest.permission.ACCESS_FINE_LOCATION}, REQUEST_LOCATION
                                );
11          }
12          else {
13
14              Location LocationGps= locationManager.getLastKnownLocation(
                    LocationManager.GPS_PROVIDER);
15              Location LocationNetwork=locationManager.getLastKnownLocation(
                    LocationManager.NETWORK_PROVIDER);
16              Location LocationPassive=locationManager.getLastKnownLocation(
                    LocationManager.PASSIVE_PROVIDER);
17
18              if (LocationGps !=null)
19              {
20                  double lat=LocationGps.getLatitude();
21                  double longi=LocationGps.getLongitude();
22                  latitude=String.valueOf(lat);
23                  longitude=String.valueOf(longi);
24                  showLocationTxt.setText("Your Location:"+"\n"+"Latitude= "+
                        latitude+"\n"+"Longitude= "+longitude);
25              }
26                 else if (LocationNetwork !=null)
27              {
28                  double lat=LocationNetwork.getLatitude();
29                  double longi=LocationNetwork.getLongitude();
30                  latitude=String.valueOf(lat);
31                  longitude=String.valueOf(longi);
32                  showLocationTxt.setText("Your Location:"+"\n"+"Latitude= "+
                        latitude+"\n"+"Longitude= "+longitude);
```

```
            }
            else if (LocationPassive !=null)
            {
                double lat=LocationPassive.getLatitude();
                double longi=LocationPassive.getLongitude();
                latitude=String.valueOf(lat);
                longitude=String.valueOf(longi);
                showLocationTxt.setText("Your Location:"+"\n"+"Latitude= "+
                    latitude+"\n"+"Longitude= "+longitude);
            }
            else
            {
                Toast.makeText(this, "Can't Get Your Location", Toast.
                    LENGTH_SHORT).show();
            }
        }
    }
```

3.5 Creation of OnGps funtion outside onCreate

```
  private void OnGPS() {

        final AlertDialog.Builder builder= new AlertDialog.Builder(this);

        builder.setMessage("Enable GPS").setCancelable(false).setPositiveButton(
            "YES", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {

                startActivity(new Intent(Settings.
                    ACTION_LOCATION_SOURCE_SETTINGS));

            }
        }).setNegativeButton("NO", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {

                dialog.cancel();
            }
        });
        final AlertDialog alertDialog=builder.create();
        alertDialog.show();
    }
```

4 Input/Output

Run the code and observe the output in the virtual device.

5 Discussion & Conclusion

From this experiments we learn about how we can get location information using location service and granting other permissions in an android environment. This experiment is designed in a way to teach the students about using android location manager for developing better applications .

6 Lab Task (Please implement yourself and show the output to the instructor)

1. Implement timestamp along with latitude and longitude information.

6.1 Problem analysis

Implement the lab task with the help of Location Manager class. Use the same permissions in the manifest file.

7 Lab Exercise (Submit as a report)

- Show nearby restaurants and hospital from your current location.

8 Policy

Copying from internet, classmate, seniors, or from any other source is strongly prohibited. 100% marks will be *deducted* if any such copying is detected.