Department of Computer Science and Engineering

Shivanirudh S G, 185001146, Semester VII

27 September 2021

UCS1711 - Mobile Application Development Lab

Exercise 6: Application that uses GPS location information

Aim:

Develop a native application that uses GPS location information

Code:

Main Activity:

Design:

```
<?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/textView"
        android:layout_width="120dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="84dp"
        android:layout_marginTop="108dp"
        android:onClick=""
        android:text="Latitude"
        android:textColor="#FFFFFF"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="120dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="84dp"
        android:layout_marginTop="144dp"
        android:text="Longitude"
        android:textColor="#FFFFFF"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textViewLatitude"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="248dp"
        android:layout_marginTop="108dp"
        android:text=""
        android:textSize="20sp"
```

```
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textViewLongitude"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="248dp"
        android:layout_marginTop="144dp"
        android:text=""
        android:textSize="20sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="158dp"
        android:layout_marginTop="393dp"
        android:text="GPS"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

Constants:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/
    android">
    <item
        android:id="@+id/action_settings"
        android:orderInCategory="100"
        android:showAsAction="never"
        android:title="@string/action_settings"/>
</menu>
```

```
<resources>
    <string name="app_name">GPS</string>
    <string name="addressLine">Address</string>
    <string name="action_settings">Action</string>
    <string name="GPSAlertDialogMessage">Alert!</string>
    <string name="GPSAlertDialogTitle">Alert!</string>
```

```
    <string name="cancel">Cancel</string>
    <string name="latitude">Latitude</string>
    <string name="longitude">Longitude</string>
    <string name="country">Country</string>
    <string name="city">City</string>
    <string name="postalCode">Postal Code</string>
</resources>
```

Behaviour:

```
package com.example.gps;

import java.io.IOException;
import java.util.List;
import java.util.Locale;

import android.annotation.SuppressLint;
import android.app.AlertDialog;
import android.app.Service;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
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 android.os.IBinder;
import android.provider.Settings;
import android.util.Log;


public class GPSTracker extends Service implements
    LocationListener {
    // Get Class Name
    private static String TAG = GPSTracker.class.getName();

    private final Context mContext;

    // flag for GPS Status
    boolean isGPSEnabled = false;

    // flag for network status
    boolean isNetworkEnabled = false;
```

```
35
36      // flag for GPS Tracking is enabled
37      boolean isGPSTrackingEnabled = false;
38
39      Location location;
40      double latitude;
41      double longitude;
42
43      // Number of Geocoders to be returned by GPSTracker
44      int geocoderMaxResults = 1;
45
46      // The minimum distance to change updates in meters
47      private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES
        = 10; // 10 meters
48
49      // The minimum time between updates in milliseconds
50      private static final long MIN_TIME_BW_UPDATES = 1000 * 60
        * 1; // 1 minute
51
52      // Declaring a Location Manager
53      protected LocationManager locationManager;
54
55      // Store LocationManager.GPS_PROVIDER or LocationManager.
      NETWORK_PROVIDER information
56      private String provider_info;
57
58      public GPSTracker(Context context) {
59          this.mContext = context;
60          getLocation();
61      }
62
63      /**
64       * Try to get my current location by GPS or Network
      Provider
65       */
66      @SuppressLint("MissingPermission")
67      public void getLocation() {
68
69          try {
70              locationManager = (LocationManager) mContext.
      getSystemService(LOCATION_SERVICE);
71
72              //getting GPS status
73              isGPSEnabled = locationManager.isProviderEnabled(
      LocationManager.GPS_PROVIDER);
```

```
74
75              //getting network status
76              isNetworkEnabled = locationManager.
    isProviderEnabled(LocationManager.NETWORK_PROVIDER);
77
78              // Try to get location if you GPS Service is
    enabled
79              if (isGPSEnabled) {
80                  this.isGPSTrackingEnabled = true;
81
82                  Log.d(TAG, "Application use GPS Service");
83
84                  /*
85                   * This provider determines location using
86                   * satellites. Depending on conditions, this
    provider may take a while to return
87                   * a location fix.
88                   */
89
90                  provider_info = LocationManager.GPS_PROVIDER;
91
92              } else if (isNetworkEnabled) { // Try to get
    location if you Network Service is enabled
93                  this.isGPSTrackingEnabled = true;
94
95                  Log.d(TAG, "Application use Network State to
    get GPS coordinates");
96
97                  /*
98                   * This provider determines location based on
99                   * availability of cell tower and WiFi access
     points. Results are retrieved
100                  * by means of a network lookup.
101                  */
102                 provider_info = LocationManager.
    NETWORK_PROVIDER;
103
104             }
105
106             // Application can use GPS or Network Provider
107             if (!provider_info.isEmpty()) {
108                 locationManager.requestLocationUpdates(
109                         provider_info,
110                         MIN_TIME_BW_UPDATES,
111                         MIN_DISTANCE_CHANGE_FOR_UPDATES,
```

```
112                         this
113                 );
114 
115                 if (locationManager != null) {
116                     location = locationManager.
    getLastKnownLocation(provider_info);
117                     updateGPSCoordinates();
118                 }
119             }
120         }
121         catch (Exception e)
122         {
123             //e.printStackTrace();
124             Log.e(TAG, "Impossible to connect to
    LocationManager", e);
125         }
126     }
127 
128     /**
129      * Update GPSTracker latitude and longitude
130      */
131     public void updateGPSCoordinates() {
132         if (location != null) {
133             latitude = location.getLatitude();
134             longitude = location.getLongitude();
135         }
136     }
137 
138     /**
139      * GPSTracker latitude getter and setter
140      * @return latitude
141      */
142     public double getLatitude() {
143         if (location != null) {
144             latitude = location.getLatitude();
145         }
146 
147         return latitude;
148     }
149 
150     /**
151      * GPSTracker longitude getter and setter
152      * @return
153      */
154     public double getLongitude() {
```

```
155         if (location != null) {
156             longitude = location.getLongitude();
157         }
158 
159         return longitude;
160     }
161 
162     /**
163      * GPSTracker isGPSTrackingEnabled getter.
164      * Check GPS/wifi is enabled
165      */
166     public boolean getIsGPSTrackingEnabled() {
167 
168         return this.isGPSTrackingEnabled;
169     }
170 
171     /**
172      * Stop using GPS listener
173      * Calling this method will stop using GPS in your app
174      */
175     public void stopUsingGPS() {
176         if (locationManager != null) {
177             locationManager.removeUpdates(GPSTracker.this);
178         }
179     }
180 
181     /**
182      * Function to show settings alert dialog
183      */
184     public void showSettingsAlert() {
185         AlertDialog.Builder alertDialog = new AlertDialog.
    Builder(mContext);
186 
187         //Setting Dialog Title
188         alertDialog.setTitle(R.string.GPSAlertDialogTitle);
189 
190         //Setting Dialog Message
191         alertDialog.setMessage(R.string.GPSAlertDialogMessage
    );
192 
193         //On Pressing Setting button
194         alertDialog.setPositiveButton(R.string.
    action_settings, new DialogInterface.OnClickListener() {
195 
196             @Override
```

```
197             public void onClick(DialogInterface dialog, int
    which)
198             {
199                 Intent intent = new Intent(Settings.
    ACTION_LOCATION_SOURCE_SETTINGS);
200                 mContext.startActivity(intent);
201             }
202         });
203 
204         //On pressing cancel button
205         alertDialog.setNegativeButton(R.string.cancel, new
    DialogInterface.OnClickListener() {
206 
207             @Override
208             public void onClick(DialogInterface dialog, int
    which)
209             {
210                 dialog.cancel();
211             }
212         });
213 
214         alertDialog.show();
215     }
216 
217     /**
218      * Get list of address by latitude and longitude
219      * @return null or List<Address>
220      */
221     public List<Address> getGeocoderAddress(Context context)
    {
222         if (location != null) {
223 
224             Geocoder geocoder = new Geocoder(context, Locale.
    ENGLISH);
225 
226             try {
227                 /**
228                  * Geocoder.getFromLocation - Returns an
    array of Addresses
229                  * that are known to describe the area
    immediately surrounding the given latitude and longitude.
230                  */
231                 List<Address> addresses = geocoder.
    getFromLocation(latitude, longitude, this.
    geocoderMaxResults);
```

```
232
233                 return addresses;
234             } catch (IOException e) {
235                 //e.printStackTrace();
236                 Log.e(TAG, "Impossible to connect to Geocoder
    ", e);
237             }
238         }
239
240         return null;
241     }
242
243     /**
244      * Try to get AddressLine
245      * @return null or addressLine
246      */
247     public String getAddressLine(Context context) {
248         List<Address> addresses = getGeocoderAddress(context)
    ;
249
250         if (addresses != null && addresses.size() > 0) {
251             Address address = addresses.get(0);
252             String addressLine = address.getAddressLine(0);
253
254             return addressLine;
255         } else {
256             return null;
257         }
258     }
259
260     /**
261      * Try to get Locality
262      * @return null or locality
263      */
264     public String getLocality(Context context) {
265         List<Address> addresses = getGeocoderAddress(context)
    ;
266
267         if (addresses != null && addresses.size() > 0) {
268             Address address = addresses.get(0);
269             String locality = address.getLocality();
270
271             return locality;
272         }
273         else {
```

```
274             return null;
275         }
276     }
277 
278     /**
279      * Try to get Postal Code
280      * @return null or postalCode
281      */
282     public String getPostalCode(Context context) {
283         List<Address> addresses = getGeocoderAddress(context)
    ;
284 
285         if (addresses != null && addresses.size() > 0) {
286             Address address = addresses.get(0);
287             String postalCode = address.getPostalCode();
288 
289             return postalCode;
290         } else {
291             return null;
292         }
293     }
294 
295     /**
296      * Try to get CountryName
297      * @return null or postalCode
298      */
299     public String getCountryName(Context context) {
300         List<Address> addresses = getGeocoderAddress(context)
    ;
301         if (addresses != null && addresses.size() > 0) {
302             Address address = addresses.get(0);
303             String countryName = address.getCountryName();
304 
305             return countryName;
306         } else {
307             return null;
308         }
309     }
310 
311     @Override
312     public void onLocationChanged(Location location) {
313     }
314 
315     @Override
316     public void onStatusChanged(String provider, int status,
```

```
    Bundle extras) {
    }

    @Override
    public void onProviderEnabled(String provider) {
    }

    @Override
    public void onProviderDisabled(String provider) {
    }

    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }
}
```

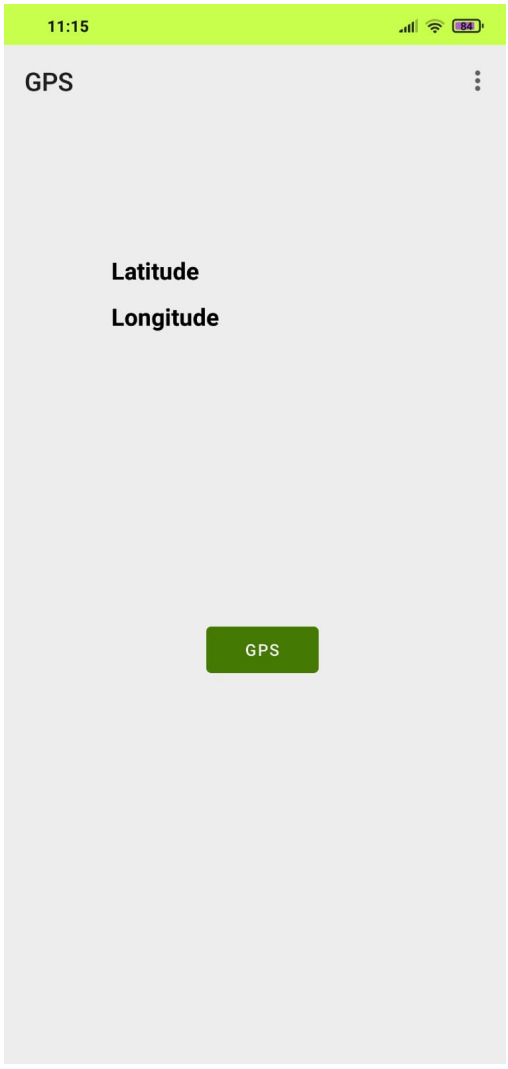
```
package com.example.gps;

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

import android.Manifest;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import java.util.List;

public class MainActivity extends AppCompatActivity {

    TextView textview;

    Button button;
    public boolean checkLocationPermission()
    {
        String permission = "android.permission.
    ACCESS_FINE_LOCATION";
        int res = this.checkCallingOrSelfPermission(
    permission);
        return (res == PackageManager.PERMISSION_GRANTED);
```

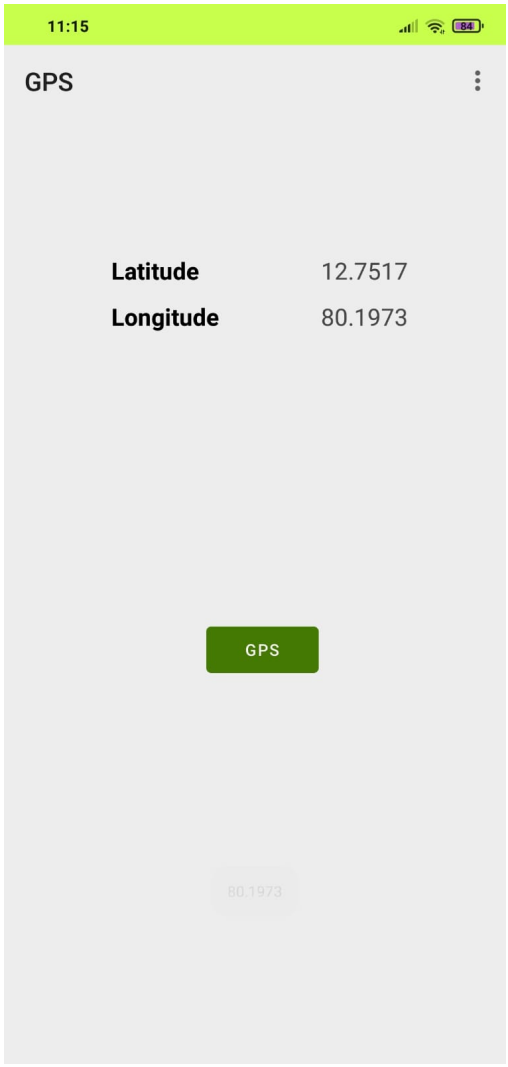
```
27      }
28
29
30      public void onRequestPermissionsResult(int requestCode,
    String permissions[], int[] grantResults) {
31          super.onRequestPermissionsResult(requestCode,
    permissions, grantResults);
32          switch (requestCode) {
33              case 1: {
34                  // If request is cancelled, the result arrays
     are empty.
35                  if (grantResults.length > 0
36                          && grantResults[0] == PackageManager.
    PERMISSION_GRANTED) {
37
38                  } else {
39                      // Permission denied. Disable the
40                      // functionality that depends on this
    permission.
41                  }
42                  return;
43              }
44              // other 'case' lines to check for other
45              // permissions this app might request
46          }
47      }
48
49      @Override
50      protected void onCreate(Bundle savedInstanceState) {
51          super.onCreate(savedInstanceState);
52          setContentView(R.layout.activity_main);
53
54          ActivityCompat.requestPermissions(this, new String[]{
    Manifest.permission.ACCESS_FINE_LOCATION}, 1);
55
56          //Check if GPS is enabled
57          GPSTracker gpsTracker = new GPSTracker(this);
58
59          button = (Button) findViewById(R.id.button);
60          button.setOnClickListener(new View.OnClickListener()
    {
61              @Override
62              public void onClick(View view) {
63
64                  if (gpsTracker.getIsGPSTrackingEnabled()) {
```

```
65                  String stringLatitude = String.valueOf(
    gpsTracker.latitude);
66                  if (stringLatitude.length() > 9) {
67                      stringLatitude = stringLatitude.
    substring(0, 8);
68                  }
69                  Toast.makeText(MainActivity.this,
    stringLatitude, Toast.LENGTH_SHORT).show();
70                  if(stringLatitude.equals("0.0")){
71                      stringLatitude = "12.7517";
72                  }
73                  Toast.makeText(MainActivity.this,
    stringLatitude, Toast.LENGTH_SHORT).show();
74                  textview = (TextView) findViewById(R.id.
    textViewLatitude);
75                  textview.setText(stringLatitude);
76
77                  String stringLongitude = String.valueOf(
    gpsTracker.longitude);
78                  if (stringLongitude.length() > 9) {
79                      stringLongitude = stringLongitude.
    substring(0, 8);
80                  }
81                  Toast.makeText(MainActivity.this,
    stringLongitude, Toast.LENGTH_SHORT).show();
82                  if(stringLongitude.equals("0.0")){
83                      stringLongitude = "80.1973";
84                  }
85                  Toast.makeText(MainActivity.this,
    stringLongitude, Toast.LENGTH_SHORT).show();
86                  textview = (TextView) findViewById(R.id.
    textViewLongitude);
87                  textview.setText(stringLongitude);
88
89              } else {
90                  System.err.println("Cant get details!");
91                  // can't get location
92                  // GPS or Network is not enabled
93                  // Ask user to enable GPS/network in
    settings
94                  gpsTracker.showSettingsAlert();
95              }
96          }
97      });
98  }
```

```
99      @Override
100     public boolean onCreateOptionsMenu(Menu menu) {
101         // Inflate the menu; this adds items to the action
        bar if it is present.
102         getMenuInflater().inflate(R.menu.menu_main, menu);
103         return true;
104     }
105 }
```

Output:




11:15
GPS
Latitude
12.7517
Longitude
80.1973
GPS
80.1973