Module 5

Location based services

What is Location based Service?

- It enables us to create an application that is capable of detecting the current location of our devices.
- So LBS is the feature that Android provides us using the Location Framework.
- This framework provides us basically with certain classes and interfaces, which are key components.

Location based Services

Android gives your applications access to the location services supported by the device through classes in the android.location package.

The central component of the location framework is the LocationManager system service, which provides APIs to determine location and bearing of the underlying device (if available).

As with other system services, you do not instantiate a LocationManager directly.

Rather, you request an instance from the system by calling getSystemService(Context.LOCATION_SERVICE).

This method returns a handle to a new LocationManager instance.

Components of Location based Services

- LocationManager Class

  This class provides access to the system location services. These services allow applications to obtain periodic updates of the device's geographical location, or to be notified when the device enters the proximity of a given geographical location.

- LocationListener Interface

  The LocationListener interface, which is part of the Android Locations API is used for receiving notifications from the **LocationManager** when the location has changed.

- LocationProvider

  The LocationProvider class is the superclass of the different location providers which deliver the information about the current location. This information is stored in the Location class.

- Location Class

  A data class representing a geographic location.
  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

Methods of LocationListener

- **onLocationChanged(Location location)** : Called when the location has changed.
- **onProviderDisabled(String provider)** : Called when the provider is disabled by the user.
- **onProviderEnabled(String provider)** : Called when the provider is enabled by the user.
- **onStatusChanged(String provider, int status, Bundle extras)** : Called when the provider status changes.

Location based Services

Once your application has a LocationManager, your application is able to do three things:

- Query for the list of all LocationProviders for the last known user location.
- Register/unregister for periodic updates of the user's current location from a location provider (specified either by criteria or name).
- Register/unregister for a given Intent to be fired if the device comes within a given proximity (specified by radius in meters) of a given lat/long.

The LocationProvider class is the superclass of the different location providers which deliver the information about the current location. This information is stored in the Location class.

Location Providers

The Android's location APIs use three different providers to get location -

- LocationManager.GPS_PROVIDER — This provider determines location using satellites. Depending on conditions, this provider may take a while to return a location fix.
- LocationManager.NETWORK_PROVIDER — This provider determines location based on availability of cell tower and WiFi access points. Results are retrieved by means of a network lookup.
- LocationManager.PASSIVE_PROVIDER — This provider will return locations generated by other providers. You passively receive location updates when other applications or services request them without actually requesting the locations yourself.

What is Latitude and Longitude?

- Just like every actual house has its address (which includes the number, the name of the street, city, etc), every single point on the surface of earth can be specified by the *latitude and longitude coordinates*.
- Therefore, by using latitude and longitude we can specify virtually any point on earth.
- The **latitude** has the symbol of *phi*, and it shows the angle between the straight line in the certain point and the equatorial plane.
- The latitude is specified by degrees, starting from 0° and ending up with 90° to both sides of the equator, making latitude Northern and Southern.
- The equator is the line with 0° latitude.
- The **longitude** has the symbol of lambda and is another angular coordinate defining the position of a point on a surface of earth.
- The longitude is defined as an angle pointing west or east from the Greenwich Meridian, which is taken as the Prime Meridian. The longitude can be defined maximum as 180° east from the Prime Meridian and 180° west from the Prime Meridian.
- Both **latitude and longitude** are measured in **degrees**, which are in turn divided into minutes and seconds.

Challenges in Determining User Location

Some sources of error in the user location include:

- Multitude of location sources -GPS, Cell-ID, and Wi-Fi can each provide a clue to users location. Determining which to use and trust is a matter of trade-offs in accuracy, speed, and battery-efficiency.
- User movement - Because the user location changes, you must account for movement by re-estimating user location every so often.
- Varying accuracy - Location estimates coming from each location source are not consistent in their accuracy. A location obtained 10 seconds ago from one source might be more accurate than the newest location from another or same source.

These problems can make it difficult to obtain a reliable user location reading.

Requesting Location Updates

Getting user location in Android works by means of callback. You indicate that you'd like to receive location updates from the **LocationManager** by calling **requestLocationUpdates(),** passing it a **LocationListener.**

LocationListener must implement several callback methods that the Location Manager calls when the user location changes or when the status of the service changes.

For example, the following code shows how to define a LocationListener and request location updates.

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

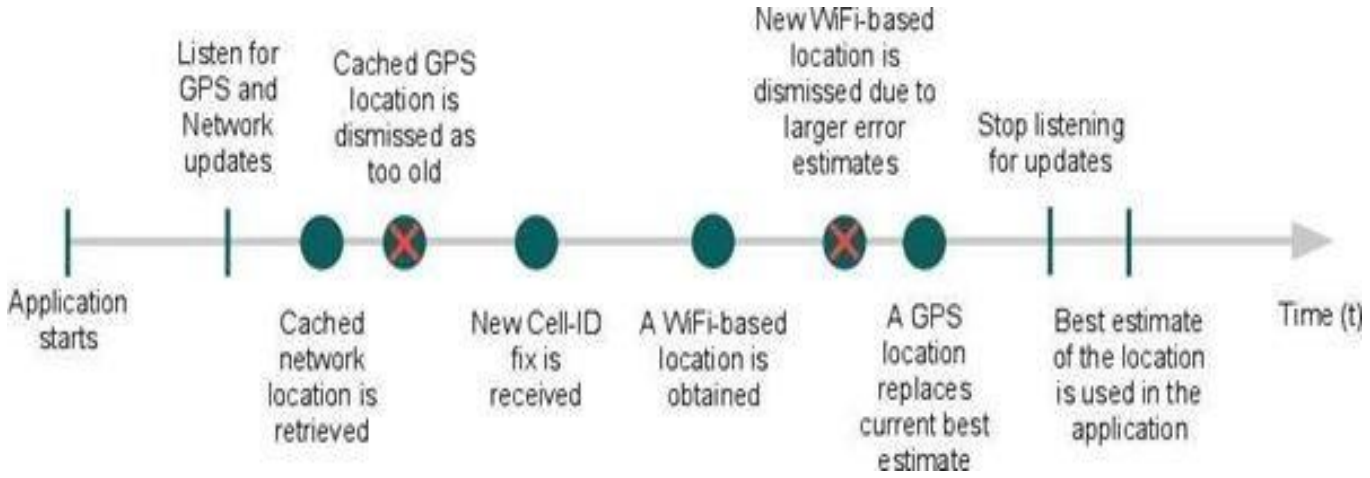
Requesting Location Updates

```
// Acquire a reference to the system Location Manager
LocationManager locationManager = (LocationManager) this.getSystemService(Context.LOCATION_SERVICE);

// Define a listener that responds to location updates
 LocationListener locationListener = new LocationListener() {

      // Called when the location has changed
      public void onLocationChanged(Location location) {}

      // Called when the provider status changes
      public void onStatusChanged(String provider, int status, Bundle extras) {}

      // Called when the provider is enabled by the user
      public void onProviderEnabled(String provider) {}

      // Called when the provider is disabled by the user
      public void onProviderDisabled(String provider) {}
   };

// Register the listener with the Location Manager to receive location updates
locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0, locationListener);
```

Flow for obtaining user location

Here's the typical flow of procedures for obtaining the user location:

1. Start application.
2. Sometime later, start listening for updates from desired location providers.
3. Maintain a "current best estimate" of location by filtering out new, but less accurate fixes.
4. Stop listening for location updates.
5. Take advantage of the last best location estimate.



- **Create an android application to display the current location of your device (longitude & latitude)**

- Getting current location through GPS provider

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<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:layout_margin="10dp"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="55dp"
        android:text="TextView" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="Check Location" />
</LinearLayout>
```

```
package com.example.mylocationapp;

import static androidx.constraintlayout.motion.widget.Debug.getLocation;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import android.Manifest;
import android.annotation.SuppressLint;
import android.content.Context;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity implements LocationListener {
Button button;
TextView textView;
LocationManager locationManager;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        textView=findViewById(R.id.textView);
        button=findViewById(R.id.button);
        if(ContextCompat.checkSelfPermission(MainActivity.this, Manifest.permission.ACCESS_FINE_LOCATION)
          != PackageManager.PERMISSION_GRANTED)
        {
            ActivityCompat.requestPermissions(MainActivity.this,new String[]{
                    Manifest.permission.ACCESS_FINE_LOCATION,
                    Manifest.permission.INTERNET},100);
        }
```

```
 button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                getLocation();
            }
        });
    }
    @SuppressLint("MissingPermission")
    private void getLocation() {
        try{
            locationManager=(LocationManager) getApplicationContext().getSystemService(LOCATION_SERVICE);
            locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,5000,5,(LocationListener)
this);
        }catch (Exception e){
            e.printStackTrace();
        }
    }
    @Override
    public void onLocationChanged(@NonNull Location location) {
        textView.setText("Latitude:"+location.getLatitude()+"\nLongitude:"+location.getLongitude());
    }
}
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.location">

   <uses-permission android:name="android.permission.INTERNET"></uses-permission>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"></uses-permission>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"></uses-
permission>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Location">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

- Getting current location through Network provider

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<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:layout_margin="10dp"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="55dp"
        android:text="TextView" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="Check Location" />
</LinearLayout>
```

```
package com.example.mylocationapp;

import static androidx.constraintlayout.motion.widget.Debug.getLocation;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import android.Manifest;
import android.annotation.SuppressLint;
import android.content.Context;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity implements LocationListener {
Button button;
TextView textView;
LocationManager locationManager;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        textView=findViewById(R.id.textView);
        button=findViewById(R.id.button);
        if(ContextCompat.checkSelfPermission(MainActivity.this, Manifest.permission.ACCESS_FINE_LOCATION)
          != PackageManager.PERMISSION_GRANTED)
        {
            ActivityCompat.requestPermissions(MainActivity.this,new String[]{
                    Manifest.permission.ACCESS_FINE_LOCATION,
                    Manifest.permission.INTERNET},100);
        }
```

```
 button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                getLocation();
            }
        });
    }
    @SuppressLint("MissingPermission")
    private void getLocation() {
        try{
            locationManager=(LocationManager) getApplicationContext().getSystemService(LOCATION_SERVICE);

locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER,5000,5,(LocationListener)
this);
        }catch (Exception e){
            e.printStackTrace();
        }
    }
    @Override
    public void onLocationChanged(@NonNull Location location) {
        textView.setText("Latitude:"+location.getLatitude()+"\nLongitude:"+location.getLongitude());
    }
}
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.location">

   <uses-permission android:name="android.permission.INTERNET"></uses-permission>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"></uses-permission>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"></uses-
permission>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Location">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```