

## Experiment No 5

### Develop GMaps application by using Linear Layout Views with different attributes

**Aim:** To Develop a GMaps application by using Linear Layout Views with different Attributes.

#### Procedure:

**Step1:** Create the New Project->Empty Views Activity

**Step 2:** Design the User Interface in the **activity\_main.xml** file

#### activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/fetchLocationButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Fetch Current Location"
        android:textColor="@android:color/white"
        android:layout_marginBottom="16dp" />

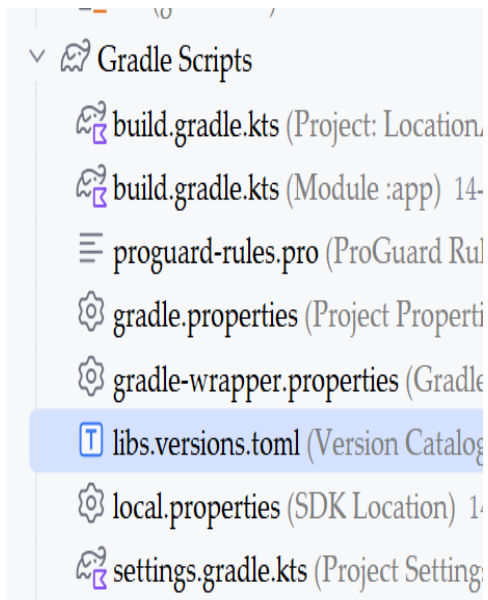
    <org.osmdroid.views.MapView
        android:id="@+id/map"
        android:layout_width="match_parent"
        android:layout_height="0dp"
```

```
android:layout_weight="1" />
```

```
</LinearLayout>
```

**Note:** For using GMaps API we have to configure the two step Verification Process which includes your card and Payment Gateway process even for using free trial of API. So, for that alternate I'm using OSM (Open Street Map) OSMDroid in this Application.

For Configuring OSMDroid in your application follow the Steps,



In libs.versions.toml file add this two things.

```
[versions]
```

```
osmdroid = "6.1.18"
```

```
[libraries]
```

```
osmdroid-android = { module =
```

```
"org.osmdroid:osmdroid-android",
```

```
version.ref = "osmdroid" }
```

**Step 3:** Add this Dependency in your Gradle Scripts build.gradle.kts

```
implementation(libs.osmdroid.android)
```

**MainActivity.kt**

```
package com.example.locationapp
```

```
import android.Manifest
```

```
import android.annotation.SuppressLint
```

```
import android.content.pm.PackageManager
import android.os.Bundle
import android.widget.Button
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat
import androidx.core.content.ContextCompat
import org.osmdroid.config.Configuration
import org.osmdroid.util.GeoPoint
import org.osmdroid.views.MapView
import org.osmdroid.views.overlay.Marker
import org.osmdroid.views.overlay.mylocation.MyLocationNewOverlay
import org.osmdroid.views.overlay.mylocation.GpsMyLocationProvider
import org.osmdroid.tileprovider.tilesource.TileSourceFactory

class MainActivity : AppCompatActivity() {

    private lateinit var mapView: MapView
    private lateinit var locationOverlay: MyLocationNewOverlay
    private lateinit var fetchLocationButton: Button

    companion object {
        private const val REQUEST_LOCATION_PERMISSION = 1
    }

    @SuppressWarnings("MissingInflatedId")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        // Load OSMDroid configuration
        Configuration.getInstance().load(this, getPreferences(MODE_PRIVATE))
        setContentView(R.layout.activity_main)

        // Initialize the MapView
        mapView = findViewById(R.id.map)
        mapView.setTileSource(TileSourceFactory.MAPNIK) // Set tile source
        mapView.setMultiTouchControls(true) // Enable multi-touch gestures

        // Set up button to fetch current location
```

```

    fetchLocationButton = findViewById(R.id.fetchLocationButton)
    fetchLocationButton.setOnClickListener {
        fetchCurrentLocation()
    }

    // Setup location overlay
    setupLocationOverlay()
}

private fun setupLocationOverlay() {
    // Check for location permission
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION)
    != PackageManager.PERMISSION_GRANTED) {
        // Request location permissions if not already granted
        ActivityCompat.requestPermissions(
            this,
            arrayOf(Manifest.permission.ACCESS_FINE_LOCATION),
            REQUEST_LOCATION_PERMISSION
        )
    } else {
        // Initialize the location overlay
        locationOverlay =
        MyLocationNewOverlay(GpsMyLocationProvider(this), mapView)
        locationOverlay.enableMyLocation() // Enable current location
        mapView.overlays.add(locationOverlay)
        mapView.invalidate() // Refresh the map view
    }
}

private fun fetchCurrentLocation() {

    val currentLocation: GeoPoint? = locationOverlay.myLocation
    if (currentLocation != null) {
        val currentGeoPoint = GeoPoint(currentLocation.latitude,
currentLocation.longitude)
        mapView.controller.setCenter(currentGeoPoint) // Center the map on the
current location
    }
}

```

```
        Toast.makeText(this, "Latitude: ${currentLocation.latitude}, Longitude:
${currentLocation.longitude}", Toast.LENGTH_SHORT).show()
```

```
        addMarker(currentGeoPoint, "Current Location")
    } else {
        Toast.makeText(this, "Current Location: Not available",
Toast.LENGTH_SHORT).show()
    }
}
```

```
private fun addMarker(geoPoint: GeoPoint, title: String) {
    val marker = Marker(mapView)
    marker.position = geoPoint
    marker.setAnchor(Marker.ANCHOR_CENTER,
Marker.ANCHOR_BOTTOM)
    marker.title = title
    mapView.overlays.add(marker)
    mapView.invalidate()
}
```

```
override fun onRequestPermissionsResult(
    requestCode: Int,
    permissions: Array<String>,
    grantResults: IntArray
) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults)
    if (requestCode == REQUEST_LOCATION_PERMISSION) {
        if (grantResults.isNotEmpty() && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            setupLocationOverlay()
        } else {
            Toast.makeText(this, "Location permission denied",
Toast.LENGTH_SHORT).show()
        }
    }
}
```

```

override fun onResume() {
    super.onResume()
    mapView.onResume()
}

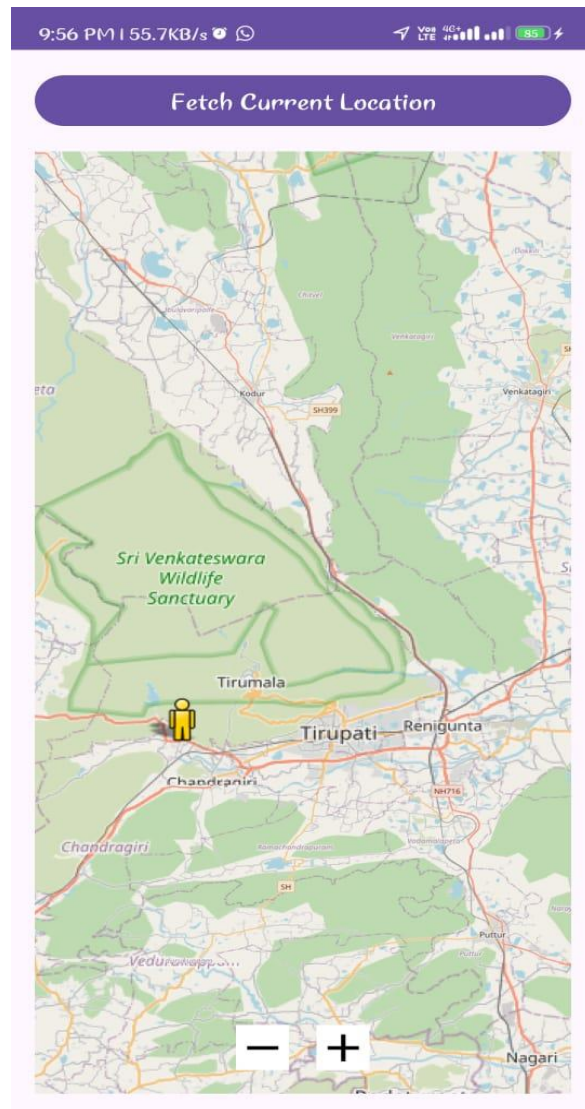
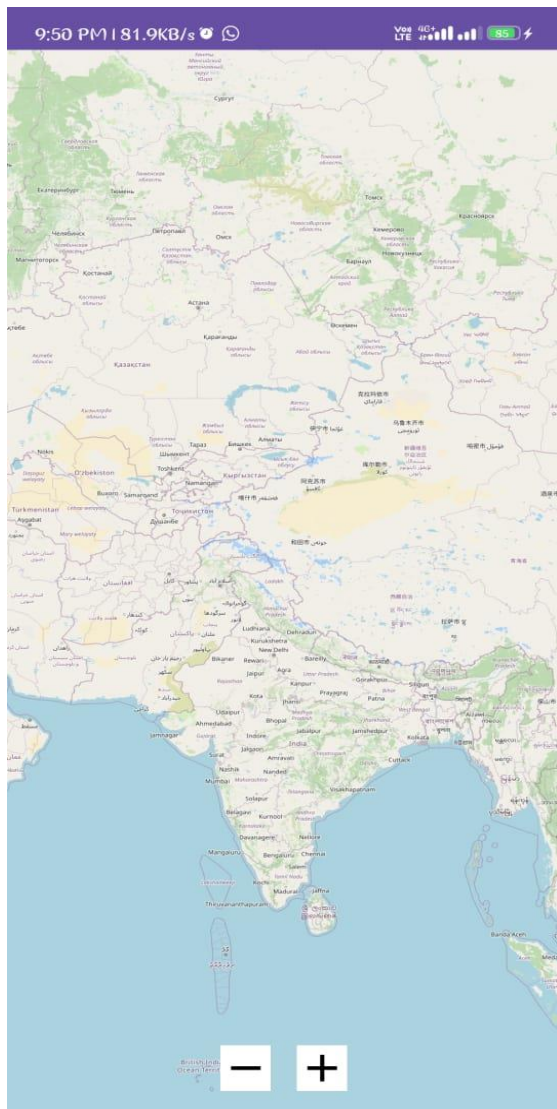
```

```

override fun onPause() {
    super.onPause()
    mapView.onPause()
}
}

```

**Output:**





9:57 PM | 17.6KB/s

VoLTE 4G+ 85%

## Fetch Current Location

