

Ex. No. : 07

Date: 27.02.2025

Register No.: 221701020

Name: Hemalatha R

Telephony services

Aim

Implement an application to get Telephony services.

Procedure:

Step 1 : File -> NewProject

Provide the application name and Click "Next"

Step 2 : Select the target android devices

Select the minimum SDK to run the application. Click "Next".

Step 3 : Choose the activity for the application (By default choose "Blank Activity) Click "Next".

Step 4 : Enter activity name and click "Finish".

Step 5 : Edit the program.

Step 6 : Run the application, 2-ways to run the application.

1. Running through emulator

2. Running through mobile device



AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.ex8">

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

    <application
        android:allowBackup="true"
        android:label="Telephony Info"
        android:theme="@style/Theme.EX8">
        <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>
```

Activity_main.xml

```
<?xml version="1.0"
    encoding="utf-8"?>
<ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
```

```
android:layout_width="match_parent"
android:layout_height="match_parent">
```

```
<LinearLayout
android:orientation="vertical"
```

```
android:padding="16dp"
```

```
android:layout_width="match_parent"
```

```
android:layout_height="wrap_content">
```

```
<TextView
```

```
android:id="@+id/tvTelephonyInfo"
```

```
android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
```

```
android:text="Telephony Info"
```

```
android:textStyle="bold"
```

```
android:textSize="16sp" />
```

```
<TextView
```

```
android:id="@+id/tvLocation"
```

```
android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
```

```
android:text="Location"
```

```
android:textStyle="bold"
```

```
android:layout_marginTop="20dp"
```

```
android:textSize="16sp" />
```

```
<TextView
```

android:id="@+id/tvAddress"

android:layout_width="match_parent"

android:layout_height="wrap_content"

android:text="Address"

android:textStyle="bold"

android:layout_marginTop="20dp"

android:textSize="16sp" />

<Button

android:id="@+id/btnFetch"

android:layout_width="match_parent"

android:layout_height="wrap_content"

android:text="Fetch Info"

android:layout_marginTop

"30dp"/>

</LinearLayout>

</ScrollView>

MainActivity.kt

```
package com.example.ex7

import android.Manifest
import android.content.pm.PackageManager
import android.location.Geocoder
import android.location.Location
import android.location.LocationManager
import android.os.Bundle
import android.telephony.TelephonyManager
import android.widget.Button
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat
import java.util.*

class MainActivity : AppCompatActivity() {

    private lateinit var tvTelephonyInfo: TextView
    private lateinit var tvLocation: TextView
    private lateinit var tvAddress: TextView
    private lateinit var btnFetch: Button
    private val LOCATION_PERMISSION = 101
    private lateinit var locationManager: LocationManager

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        tvTelephonyInfo = findViewById(R.id.tvTelephonyInfo)
        tvLocation = findViewById(R.id.tvLocation)
        tvAddress = findViewById(R.id.tvAddress)
        btnFetch = findViewById(R.id.btnFetch)

        btnFetch.setOnClickListener {
            if (checkPermissions()) {
                displayTelephonyInfo()
                fetchLocation()
            } else {
                requestPermissions()
            }
        }
    }
}
```

```

    }
    }

    private fun checkPermissions(): Boolean {
        return ActivityCompat.checkSelfPermission(this,
            Manifest.permission.ACCESS_FINE_LOCATION) ==
            PackageManager.PERMISSION_GRANTED &&
            ActivityCompat.checkSelfPermission(this,
            Manifest.permission.READ_PHONE_STATE) ==
            PackageManager.PERMISSION_GRANTED
    }

    private fun requestPermissions() {
        ActivityCompat.requestPermissions(this,
            arrayOf(Manifest.permission.ACCESS_FINE_LOCATION,
            Manifest.permission.READ_PHONE_STATE),
            LOCATION_PERMISSION)
    }

    override fun onRequestPermissionsResult(requestCode: Int, permissions:
        Array<out String>, grantResults: IntArray) {
        if (requestCode == LOCATION_PERMISSION && grantResults.isNotEmpty() &&
            grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            displayTelephonyInfo()
            fetchLocation()
        } else {
            tvTelephonyInfo.text = "Permission Denied"
            tvLocation.text = "Permission Denied"
        }
    }

    private fun displayTelephonyInfo() {
        val telephonyManager = getSystemService(TELEPHONY_SERVICE) as
            TelephonyManager
        val info = ""
        Network Operator: ${telephonyManager.networkOperatorName} SIM
        Country: ${telephonyManager.simCountryIso}
        SIM Operator: ${telephonyManager.simOperatorName}
        Phone Type: ${when (telephonyManager.phoneType) {

            TelephonyManager.PHONE_TYPE_GSM -> "GSM"
            TelephonyManager.PHONE_TYPE_CDMA -> "CDMA"
        }}
    }

```

```

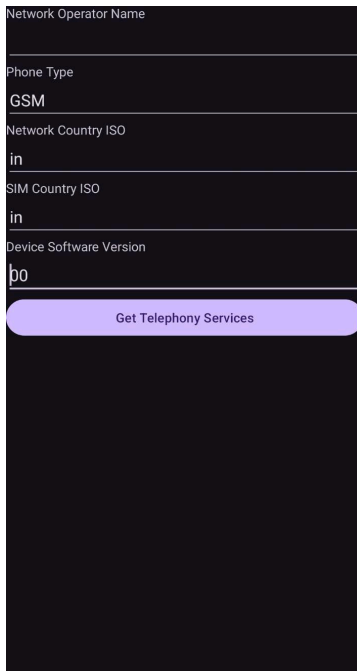
else -> "Unknown"
}}
"".trimIndent()
tvTelephonyInfo.text = info
}
private fun fetchLocation() {
    locationManager = getSystemService(LOCATION_SERVICE) as
    LocationManager
    if (ActivityCompat.checkSelfPermission(this,
    Manifest.permission.ACCESS_FINE_LOCATION) !=
    PackageManager.PERMISSION_GRANTED) return
    val location: Location? =
    locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER) ?:
    locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVIDER )

    if (location != null) {
        val lat = location.latitude
        val lon = location.longitude
        tvLocation.text = "Latitude: $lat\nLongitude: $lon"
        getAddress(lat, lon)
    } else {
        tvLocation.text = "Unable to get location."
    }
}

private fun getAddress(lat: Double, lon: Double) {
    val geocoder = Geocoder(this, Locale.getDefault())
    try {
        val addressList = geocoder.getFromLocation(lat, lon, 1)
        if (!addressList.isNullOrEmpty()) {
            val address = addressList[0]
            val fullAddress = address.getAddressLine(0)
            tvAddress.text = "Address:\n$fullAddress"
        } else {
            tvAddress.text = "Unable to get address."
        }
    } catch (e: Exception) {
        e.printStackTrace()
        tvAddress.text = "Geocoder error: ${e.localizedMessage}"
    }
}
}

```

Output :



Result:

The Application was developed using Kotlin in Android Studio.