

**Ex. No.** : 07

**Date:**04-05-25

**Register No.:** 221701055

**Name:** S Sneha

---

## **Telephony Services**

### **Aim**

Develop an android application to display the information of the 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***

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

    <uses-feature
        android:name="android.hardware.telephony"
        android:required="false" />

    <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.READ_PRIVILEGED_PHONE_STATE"
        tools:ignore="ProtectedPermissions" />
    <uses-permission android:name="android.permission.READ_PHONE_NUMBERS"/>
    <uses-permission android:name="android.permission.READ_SMS" />

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.TelePhony"
        tools:targetApi="31">
        <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"?>
<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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textViewTitle"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="Telephony Services"
        android:textSize="20sp"
        android:textStyle="bold"
        android:textColor="@android:color/white"
        android:gravity="center"
        android:background="@android:color/holo_purple" />

    <ScrollView
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintTop_toBottomOf="@id/textViewTitle"
        app:layout_constraintBottom_toTopOf="@id/btnFetch"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:padding="8dp">

            <TextView
                android:id="@+id/tvDeviceName"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Device Name:"
                android:textSize="16sp"
                android:padding="8dp"
                android:textColor="#000000" />
```

```
<TextView
    android:id="@+id/tvPhoneType"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Phone Type:"
    android:textSize="16sp"
    android:padding="8dp"
    android:textColor="#000000" />
```

```
<TextView
    android:id="@+id/tvNetworkISO"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Network Country ISO:"
    android:textSize="16sp"
    android:padding="8dp"
    android:textColor="#000000" />
```

```
<TextView
    android:id="@+id/tvSimISO"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="SIM Country ISO:"
    android:textSize="16sp"
    android:padding="8dp"
    android:textColor="#000000" />
```

```
<TextView
    android:id="@+id/tvIMEI"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="IMEI:"
    android:textSize="16sp"
    android:padding="8dp"
    android:textColor="#000000" />
```

```
<TextView
    android:id="@+id/tvSoftwareVersion"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Device Software Version:"
    android:textSize="16sp"
    android:padding="8dp"
    android:textColor="#000000" />
```

```
</LinearLayout>
</ScrollView>
```

```
<Button
    android:id="@+id/btnFetch"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Get Telephony Services"
    android:layout_marginBottom="16dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />

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



## **MainActivity.kt**

```
package com.example.mad7

import android.Manifest
import android.content.pm.PackageManager
import android.os.Build
import android.os.Bundle
import android.telephony.TelephonyManager
import android.widget.Button
import android.widget.TextView
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat
import androidx.core.content.ContextCompat
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat

class MainActivity : AppCompatActivity() {

    private lateinit var telephonyManager: TelephonyManager

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

        // Apply window insets to handle notches and system bars
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
            insets
        }

        telephonyManager = getSystemService(TELEPHONY_SERVICE) as TelephonyManager

        val btnFetch = findViewById<Button>(R.id.btnFetch)
        val tvDeviceName = findViewById<TextView>(R.id.tvDeviceName)
        val tvPhoneType = findViewById<TextView>(R.id.tvPhoneType)
        val tvNetworkISO = findViewById<TextView>(R.id.tvNetworkISO)
        val tvSimISO = findViewById<TextView>(R.id.tvSimISO)
        val tvIMEI = findViewById<TextView>(R.id.tvIMEI)
        val tvSoftwareVersion = findViewById<TextView>(R.id.tvSoftwareVersion)
```

```

btnFetch.setOnClickListener {
    if (ContextCompat.checkSelfPermission(this, Manifest.permission.READ_PHONE_STATE)
        != PackageManager.PERMISSION_GRANTED
    ) {
        ActivityCompat.requestPermissions(
            this,
            arrayOf(Manifest.permission.READ_PHONE_STATE),
            1
        )
    } else {
        // ● Fetch and display phone info
        val phoneType = when (telephonyManager.phoneType) {
            TelephonyManager.PHONE_TYPE_GSM -> "GSM"
            TelephonyManager.PHONE_TYPE_CDMA -> "CDMA"
            TelephonyManager.PHONE_TYPE_SIP -> "SIP"
            TelephonyManager.PHONE_TYPE_NONE -> "None"
            else -> "Unknown"
        }

        val imei = if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            if (ActivityCompat.checkSelfPermission(
                this,
                Manifest.permission.READ_PHONE_STATE
            ) == PackageManager.PERMISSION_GRANTED
            ) {
                telephonyManager.imei ?: "Not available"
            } else {
                "Permission denied"
            }
        } else {
            "Not supported on this Android version"
        }

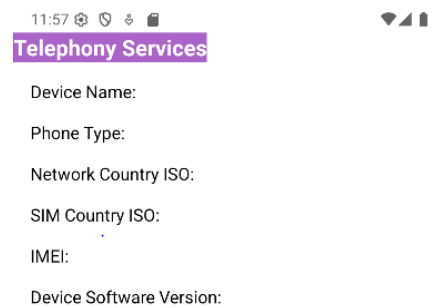
        tvDeviceName.text = "Device Name: ${Build.MODEL}"
        tvPhoneType.text = "Phone Type: $phoneType"
        tvNetworkISO.text =
            "Network Country ISO: ${telephonyManager.networkCountryIso.uppercase()}"
        tvSimISO.text =
            "SIM Country ISO: ${telephonyManager.simCountryIso.uppercase()}"
        tvIMEI.text = "IMEI: $imei"
        tvSoftwareVersion.text =
            "Device Software Version: ${telephonyManager.deviceSoftwareVersion ?:
            "Unavailable"}"
    }
}

```



```
// 🔄 Handle permission result
override fun onRequestPermissionsResult(
    requestCode: Int,
    permissions: Array<out String>,
    grantResults: IntArray
) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults)
    if (requestCode == 1) {
        if (grantResults.isNotEmpty() && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            // Retry button click now that permission is granted
            findViewById<Button>(R.id.btnFetch).performClick()
        } else {
            Toast.makeText(this, "Permission denied", Toast.LENGTH_SHORT).show()
        }
    }
}
}
```

## Output



## Result:

The experiment was conducted successfully

