

## EX 10

### ANDROID MANIFEST

```
<?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-permission android:name="android.permission.READ_PHONE_STATE" />

    <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:supportsRtl="true"
        android:theme="@style/Theme.Ex10"
        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 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">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Network Operator Name:" />

    <EditText
        android:id="@+id/networkOperatorNameEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="@color/black"
        android:enabled="false" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Phone Type:" />

    <EditText
        android:id="@+id/phoneTypeEditText"
        android:layout_width="match_parent"
```

```

        android:layout_height="wrap_content"
        android:textColor="@color/black"
        android:enabled="false" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Network Country ISD:" />

<EditText
    android:id="@+id/networkCountryISDEditText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColor="@color/black"
    android:enabled="false" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="SIM Country ISD:" />

<EditText
    android:id="@+id/simCountryISDEditText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColor="@color/black"
    android:enabled="false" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Device Software Version:" />

<EditText
    android:id="@+id/deviceSoftwareVersionEditText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColor="@color/black"
    android:enabled="false" />

<Button
    android:id="@+id/getTelephonyServiceButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:text="Get Telephony Service" />

</LinearLayout>

```

## MAIN ACTIVITY

```

package com.example.ex10

import android.Manifest
import android.content.Context
import android.content.pm.PackageManager
import android.os.Build
import android.os.Bundle
import android.telephony.TelephonyManager
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat

```

```

class MainActivity : AppCompatActivity() {

    private lateinit var networkOperatorNameEditText: EditText
    private lateinit var phoneTypeEditText: EditText
    private lateinit var networkCountryISDEditText: EditText
    private lateinit var simCountryISDEditText: EditText
    private lateinit var deviceSoftwareVersionEditText: EditText
    private lateinit var getTelephonyServiceButton: Button

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

        networkOperatorNameEditText =
            findViewById(R.id.networkOperatorNameEditText)
        phoneTypeEditText = findViewById(R.id.phoneTypeEditText)
        networkCountryISDEditText = findViewById(R.id.networkCountryISDEditText)
        simCountryISDEditText = findViewById(R.id.simCountryISDEditText)
        deviceSoftwareVersionEditText =
            findViewById(R.id.deviceSoftwareVersionEditText)
        getTelephonyServiceButton = findViewById(R.id.getTelephonyServiceButton)

        getTelephonyServiceButton.setOnClickListener {
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                if (checkSelfPermission(Manifest.permission.READ_PHONE_STATE) ==
                    PackageManager.PERMISSION_GRANTED) {
                    getTelephonyInfo()
                } else {
                    ActivityCompat.requestPermissions(
                        this,
                        arrayOf(Manifest.permission.READ_PHONE_STATE),
                        PERMISSION_REQUEST_CODE
                    )
                }
            } else {
                getTelephonyInfo()
            }
        }
    }

    private fun getTelephonyInfo() {
        val telephonyManager = getSystemService(Context.TELEPHONY_SERVICE) as
            TelephonyManager

        networkOperatorNameEditText.setText(telephonyManager.networkOperatorName)
        phoneTypeEditText.setText(getPhoneType(telephonyManager.phoneType))
        networkCountryISDEditText.setText(telephonyManager.networkCountryIso)
        simCountryISDEditText.setText(telephonyManager.simCountryIso)
        if (ActivityCompat.checkSelfPermission(
            this,
            Manifest.permission.READ_PHONE_STATE
        ) != PackageManager.PERMISSION_GRANTED
        ) {
            // TODO: Consider calling
            //     ActivityCompat#requestPermissions
            // here to request the missing permissions, and then overriding
            //     public void onRequestPermissionsResult(int requestCode, String[]
permissions,
            //                                     int[] grantResults)
            // to handle the case where the user grants the permission. See the
documentation
            // for ActivityCompat#requestPermissions for more details.
            return
        }
    }
}

```

```

deviceSoftwareVersionEditText.setText(telephonyManager.deviceSoftwareVersion)
}

private fun getPhoneType(phoneType: Int): String {
    return when (phoneType) {
        TelephonyManager.PHONE_TYPE_CDMA -> "CDMA"
        TelephonyManager.PHONE_TYPE_GSM -> "GSM"
        TelephonyManager.PHONE_TYPE_NONE -> "None"
        else -> "Unknown"
    }
}

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

companion object {
    private const val PERMISSION_REQUEST_CODE = 100
}
}

```