Telephony Services

Aim

Develop an android application to display the information of the telephony services.

Procedure:

Step 1 : File → New Project

Provide the application name (e.g.,"TelephonyServices") 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

Request necessary permissions Request in the manifest file.

Use Kotlin code in MainActivity.kt to implement telephony features '

such as dialing a number or retrieving SIM info

Step 6 : Run the application

Two ways to run the application:

1. Running through emulator (with SD card configured)

2. Running through mobile device

AndroidManifest.xml

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

    <!-- Telephony permissions -->
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.READ_PHONE_NUMBERS" />

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

    <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.TelephonyServices">
        <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
<?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:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Telephony Services"
        android:textSize="22sp"
        android:textStyle="bold"
        android:textColor="@android:color/white"
        android:padding="16dp"
        android:background="#6200EE" />

    <!-- Network Operator Name -->
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Network Operator Name"
        android:textSize="14sp"
        android:paddingStart="16dp"
        android:paddingTop="8dp"
        android:textColor="#757575"/>

    <TextView
        android:id="@+id/networkOperatorNameTextView"
        android:layout_width="match_parent"
```

```xml
        android:layout_height="wrap_content"

        android:background="@drawable/text_view_border"

        android:padding="16dp"

        android:textSize="16sp"

        android:textColor="#212121"

        tools:text="Android" />


    <!-- Phone Type -->

    <TextView

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:text="Phone Type"

        android:textSize="14sp"

        android:paddingStart="16dp"

        android:paddingTop="8dp"

        android:textColor="#757575"/>


    <TextView

        android:id="@+id/phoneTypeTextView"

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:background="@drawable/text_view_border"

        android:padding="16dp"

        android:textSize="16sp"

        android:textColor="#212121"

        tools:text="GSM" />


    <!-- Network Country ISO -->

    <TextView

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:text="Network Country ISO"

        android:textSize="14sp"

        android:paddingStart="16dp"

        android:paddingTop="8dp"

        android:textColor="#757575"/>
```

```xml
<TextView
    android:id="@+id/networkCountryIsoTextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@drawable/text_view_border"
    android:padding="16dp"
    android:textSize="16sp"
    android:textColor="#212121"
    tools:text="US" />

<!-- SIM Country ISO -->
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="SIM Country ISO"
    android:textSize="14sp"
    android:paddingStart="16dp"
    android:paddingTop="8dp"
    android:textColor="#757575"/>

<TextView
    android:id="@+id/simCountryIsoTextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@drawable/text_view_border"
    android:padding="16dp"
    android:textSize="16sp"
    android:textColor="#212121"
    tools:text="US" />

<!-- Device Software Version -->
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Device Software Version"
    android:textSize="14sp"
    android:paddingStart="16dp"
```

```xml
        android:paddingTop="8dp"
        android:textColor="#757575"/>


    <TextView
        android:id="@+id/deviceSoftwareVersionTextView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@drawable/text_view_border"
        android:padding="16dp"
        android:textSize="16sp"
        android:textColor="#212121"
        tools:text="---" />


    <View
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1" />


    <Button
        android:id="@+id/getServicesButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Get Telephony Services"
        android:textColor="@android:color/white"
        android:backgroundTint="#6200EE"
        android:padding="16dp"
        android:layout_margin="16dp"/>


</LinearLayout>
```

MainActivity.kt

```kotlin
package com.example.telephony services

import android.Manifest

import android.content.Context

import android.content.pm.PackageManager

import android.os.Bundle

import android.telephony.TelephonyManager

import android.widget.Button

import android.widget.TextView

import android.widget.Toast

import androidx.appcompat.app.AlertDialog

import android.appcompat.app.AppCompatActivity

import android.core.app.ActivityCompat

import androidx.core.content.ContextCompat


class MainActivity : AppCompatActivity() {

    private lateinit var networkOperatorNameTextView: TextView

    private lateinit var phoneTypeTextView: TextView

    private lateinit var networkCountryIsoTextView: TextView

    private lateinit var simCountryIsoTextView: TextView

    private lateinit var deviceSoftwareVersionTextView: TextView

    private lateinit var getServicesButton: Button


    private val TELEPHONY_PERMISSION_CODE = 101


    override fun onCreate(savedInstanceState: Bundle?) {

        super.onCreate(savedInstanceState)

        setContentView(R.layout.activity_main)


        // Initialize UI components
```

```kotlin
        networkOperatorNameTextView =
findViewById(R.id.networkOperatorNameTextView)

        phoneTypeTextView = findViewById(R.id.phoneTypeTextView)

        networkCountryIsoTextView =
findViewById(R.id.networkCountryIsoTextView)

        simCountryIsoTextView = findViewById(R.id.simCountryIsoTextView)

        deviceSoftwareVersionTextView =
findViewById(R.id.deviceSoftwareVersionTextView)

        getServicesButton = findViewById(R.id.getServicesButton)


        // Set click listener for button

        getServicesButton.setOnClickListener {

            checkTelephonyPermission()

        }

    }


    private fun checkTelephonyPermission() {

        if (ContextCompat.checkSelfPermission(

                this,

                Manifest.permission.READ_PHONE_STATE

            ) != PackageManager.PERMISSION_GRANTED

        ) {

            // Show permission explanation dialog

            showPermissionDialog()

        } else {

            // Permission already granted, get telephony info

            getTelephonyInfo()

        }

    }


    private fun showPermissionDialog() {

        AlertDialog.Builder(this)

            .setTitle("Allow Telephony Services")
```

```kotlin
            .setMessage("to make and manage phone calls?")

            .setIcon(R.drawable.ic_phone)

            .setPositiveButton("ALLOW") { _, _ ->

                // Request permission

                ActivityCompat.requestPermissions(

                    this,

                    arrayOf(Manifest.permission.READ_PHONE_STATE),

                    TELEPHONY_PERMISSION_CODE

                )

            }

            .setNegativeButton("DENY") { dialog, _ ->

                dialog.dismiss()

                Toast.makeText(

                    this,

                    "Permission denied. Cannot access telephony services.",

                    Toast.LENGTH_SHORT

                ).show()

            }

            .create()

            .show()

    }


    private fun getTelephonyInfo() {

        val telephonyManager = getSystemService(Context.TELEPHONY_SERVICE) as
TelephonyManager


        // Get network operator name

        val networkOperatorName = if (ActivityCompat.checkSelfPermission(

                this,

                Manifest.permission.READ_PHONE_STATE

            ) == PackageManager.PERMISSION_GRANTED

        ) {
```

```kotlin
            telephonyManager.networkOperatorName.ifEmpty { "Android" }

        } else {

            "Permission required"

        }

        networkOperatorNameTextView.text = networkOperatorName


        // Get phone type

        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"

        }

        phoneTypeTextView.text = phoneType


        // Get network country ISO

        val networkCountryIso =
telephonyManager.networkCountryIso.uppercase().ifEmpty { "US" }

        networkCountryIsoTextView.text = networkCountryIso


        // Get SIM country ISO

        val simCountryIso = telephonyManager.simCountryIso.uppercase().ifEmpty
{ "US" }

        simCountryIsoTextView.text = simCountryIso


        // Get device software version

        val deviceSoftwareVersion = if (ActivityCompat.checkSelfPermission(

                this,

                Manifest.permission.READ_PHONE_STATE

            ) == PackageManager.PERMISSION_GRANTED

        ) {
```

```kotlin
            telephonyManager.deviceSoftwareVersion ?: "---"

        } else {

            "---"

        }

        deviceSoftwareVersionTextView.text = deviceSoftwareVersion


        Toast.makeText(this, "Telephony information updated",
Toast.LENGTH_SHORT).show()

    }


    override fun onRequestPermissionsResult(

        requestCode: Int,

        permissions: Array<String>,

        grantResults: IntArray

    ) {

        super.onRequestPermissionsResult(requestCode, permissions,
grantResults)

        if (requestCode == TELEPHONY_PERMISSION_CODE) {

            if (grantResults.isNotEmpty() && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {

                // Permission granted, get telephony info

                getTelephonyInfo()

            } else {

                Toast.makeText(

                    this,

                    "Permission denied. Cannot access telephony services.",

                    Toast.LENGTH_SHORT

                ).show()

            }

        }

    }

}
```
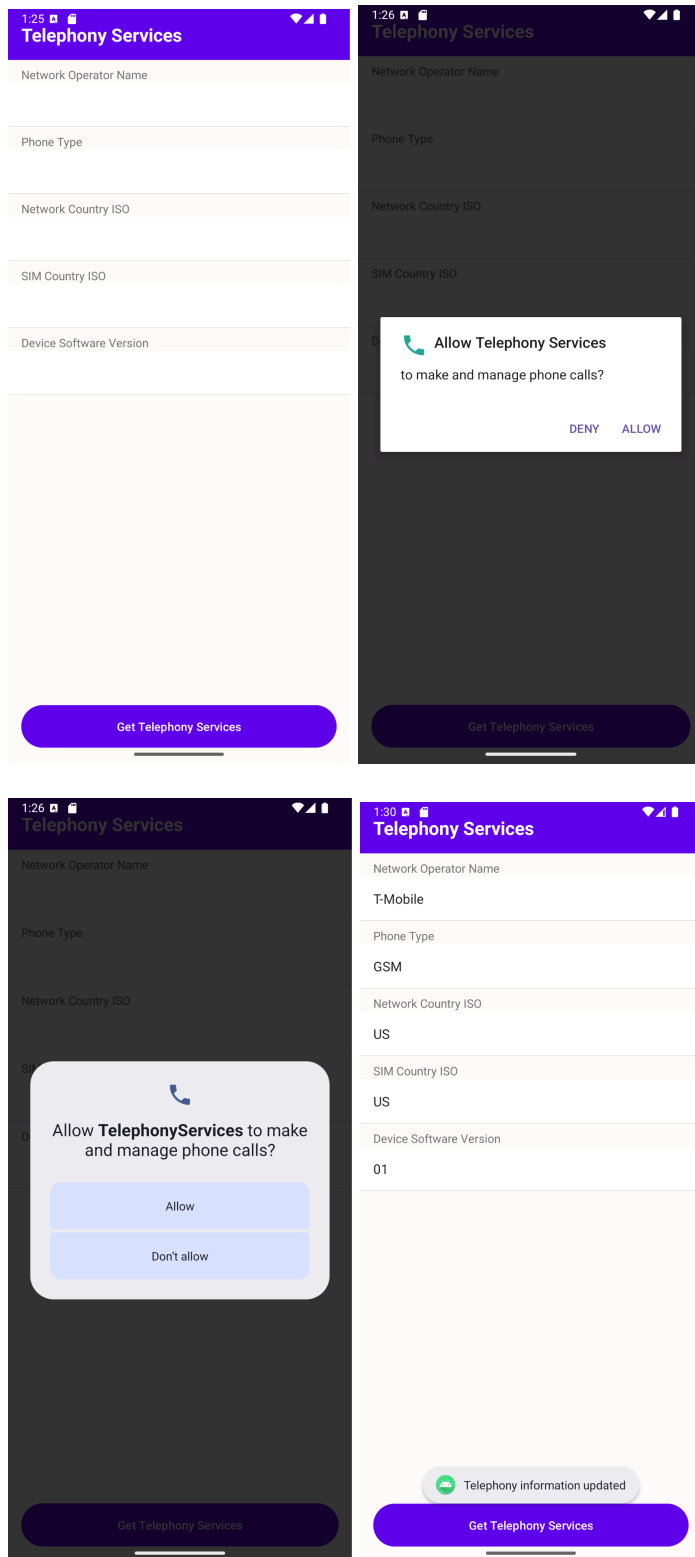
Output

Result:

The Telephony Services application successfully accesses telephony features like initiating phone calls or displaying SIM and network information on a mobile device with granted permissions.