**Ex. No. : 07 Date: 28/04/2025**

**Register No.: 221701050 Name: D Sathiya Sri**



**Telephony Services**

**Aim**

Implement an application to get Telephony services.

***Procedure:***

1. Create a New Android Project

* Open Android Studio and create a new project with an Empty Activity.
* Choose Kotlin as the programming language.

1. Add Permissions in AndroidManifest.xml

* Add necessary permissions to the AndroidManifest.xml to access telephony services.

1. Request Runtime Permissions

* Request runtime permissions to access phone state and other telephony-related features.

1. Get Telephony Manager Service

* Access the telephony manager system service to interact with telephony services.

1. Access Telephony Information

* Retrieve different telephony data such as phone number, SIM operator, network type, and device ID.

1. Display Telephony Information

* Set up UI components (like TextView) to display the retrieved telephony data.

1. Handle Button Clicks (Optional)

* If needed, set an OnClickListener to fetch and display telephony information when the user interacts with the app.

1. Run and Test the Application

* Build and run the app on a physical device to ensure that telephony data is correctly retrieved and displayed.

***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" package="com.example.exp\_7" >

<uses-permission android:name="android.permission.READ\_PHONE\_STATE" />

<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:supportsRtl="true"

android:theme="@style/Theme.Exp7"

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"?>

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:padding="16dp"

android:gravity="center"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<Button

android:id="@+id/btnGetInfo"

android:text="Get Telephony Info"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" />

<TextView

android:id="@+id/txtInfo"

android:layout\_marginTop="20dp"

android:textSize="16sp"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

</LinearLayout>

***MainActivity.kt***

package com.example.exp\_7

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.appcompat.app.AppCompatActivity

import androidx.core.app.ActivityCompat

import androidx.core.content.ContextCompat

class MainActivity : AppCompatActivity() {

private lateinit var btnGetInfo: Button

private lateinit var txtInfo: TextView

private val REQUEST\_PHONE\_STATE = 100

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

btnGetInfo = findViewById(R.id.btnGetInfo)

txtInfo = findViewById(R.id.txtInfo)

btnGetInfo.setOnClickListener {

if (ContextCompat.checkSelfPermission(this, Manifest.permission.READ\_PHONE\_STATE) != PackageManager.PERMISSION\_GRANTED) {

ActivityCompat.requestPermissions(this, arrayOf(Manifest.permission.READ\_PHONE\_STATE), REQUEST\_PHONE\_STATE)

}

else {

displayTelephonyInfo()

}

}

}

private fun displayTelephonyInfo() {

val telephonyManager = getSystemService(TELEPHONY\_SERVICE) as TelephonyManager

val builder = StringBuilder()

builder.append("Network Operator Name: ${telephonyManager.networkOperatorName}\n")

builder.append("SIM Operator Name: ${telephonyManager.simOperatorName}\n")

builder.append("SIM Country ISO: ${telephonyManager.simCountryIso}\n")

if (ActivityCompat.checkSelfPermission(this, Manifest.permission.READ\_PHONE\_STATE) == PackageManager.PERMISSION\_GRANTED) {

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.O) {

builder.append("IMEI: ${telephonyManager.imei}\n")

} else {

builder.append("Device ID: ${telephonyManager.deviceId}\n")

}

builder.append("Phone Number: ${telephonyManager.line1Number}\n")

}

txtInfo.text = builder.toString()

}

override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<out String>, grantResults: IntArray) {

super.onRequestPermissionsResult(requestCode, permissions, grantResults)

if (requestCode == REQUEST\_PHONE\_STATE) {

if (grantResults.isNotEmpty() && grantResults[0] == PackageManager.PERMISSION\_GRANTED) {

displayTelephonyInfo()

} else {

Toast.makeText(this, "Permission Denied", Toast.LENGTH\_SHORT).show()

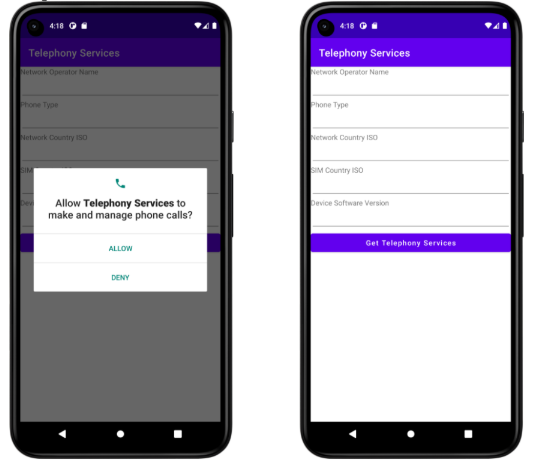
}

}

}

}

***Output***

******

**Result:**

| The app successfully retrieves and displays telephony services information, such as phone number, network type, and SIM operator, by using the Telephony Manager. |
| --- |