Ex. No. : 8A **Date: 04/05/2025**

Register No.: 221701056 **Name: SNEKA SORNA.P.S**

Send SMS

Aim

Develop an application to send SMS.

Procedure:

Step 1: Setting Up Your Environment

1. Install Android Studio: If it's not already installed, download and install Android Studio from the official Android Developer website.

2. Create a New Project: Open Android Studio, start a new project by selecting "New Project" > "Empty Activity". Name your project (e.g., "SendSMSApp"), select Kotlin as the language, and choose an API level.

Step 2: Updating the Manifest for Permissions

1. Modify AndroidManifest.xml: To send SMS messages, you need to request permission in your AndroidManifest.xml. Open this file and add the following permission

Step 3: Designing the User Interface

1. Open activity_main.xml: Navigate to app > res > layout > activity_main.xml.

2. Add UI Elements: Design your layout to include EditText fields for entering the recipient's phone number and the message, and a Button to send the SMS.

Step 4: Implementing SMS Sending Functionality

Update MainActivity.kt: You'll need to use SmsManager to send the SMS. Update your MainActivity.kt with logic to send an SMS when the button is clicked.

Step 5: Handling Permissions

Request SMS Permission at Runtime: Since sending SMS is a sensitive permission, you should request it at runtime, especially for devices running Android 6.0 (Marshmallow) and above.

Step 6: Running Your Application

1. Select a Device: Choose a real Android device (SMS functionality may not work on emulators).

2. Run the App: Click the "Run" button in Android Studio to build and run your application.

3. Test the App: Enter a phone number and a message in the provided fields, and click the "Send SMS" button. Ensure that the app has permission to send SMS.

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_8a">
    <uses-permission android:name="android.permission.SEND_SMS"/>
    <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.Exp8a"
        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>
```

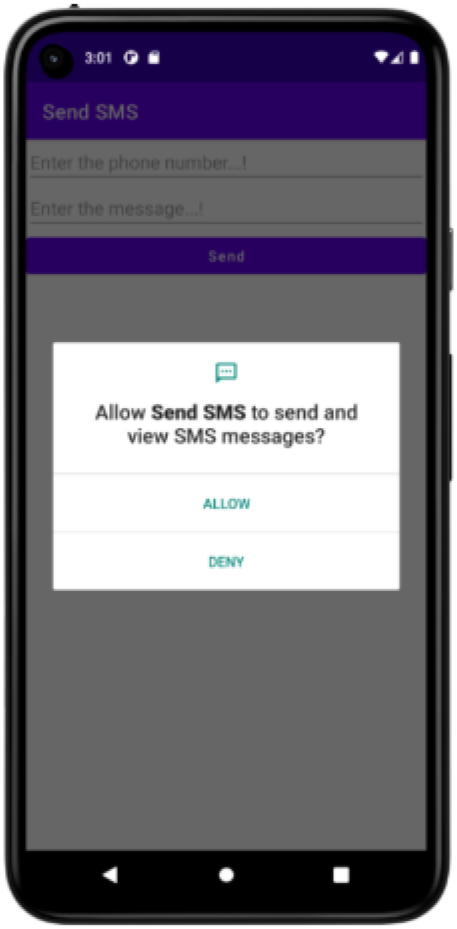
Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="20dp"
    android:orientation="vertical">

    <EditText
        android:id="@+id/editTextPhone"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Phone Number"
        android:inputType="phone" />

    <EditText
        android:id="@+id/editTextMessage"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Message"
        android:inputType="textMultiLine"
        android:minLines="3"
        android:gravity="top" />

    <Button
        android:id="@+id/buttonSend"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Send SMS" />
</LinearLayout>
```

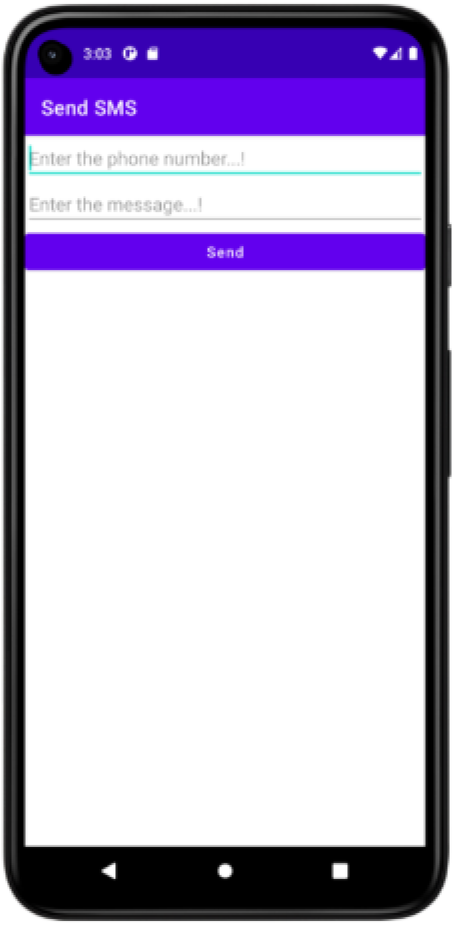
MainActivity.kt

```
package com.example.exp_8a

import android.Manifest
import android.content.pm.PackageManager
import android.os.Bundle
import android.telephony.SmsManager
import android.widget.Button
import android.widget.EditText
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 editTextPhone: EditText
    private lateinit var editTextMessage: EditText
    private lateinit var buttonSend: Button

    private val SMS_PERMISSION_CODE = 100

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

        editTextPhone = findViewById(R.id.editTextPhone)
        editTextMessage = findViewById(R.id.editTextMessage)
        buttonSend = findViewById(R.id.buttonSend)
        buttonSend.setOnClickListener {
            if (checkPermission()) {
                sendSMS()
            } else {
                requestPermission()
            }
        }
    }

    private fun checkPermission(): Boolean {
        val result = ContextCompat.checkSelfPermission(this,
Manifest.permission.SEND_SMS)
        return result == PackageManager.PERMISSION_GRANTED
    }

    private fun requestPermission() {
        ActivityCompat.requestPermissions(
            this,
            arrayOf(Manifest.permission.SEND_SMS),
            SMS_PERMISSION_CODE
        )
    }

    private fun sendSMS() {
        val phoneNumber = editTextPhone.text.toString()
        val message = editTextMessage.text.toString()

        if (phoneNumber.isNotEmpty() && message.isNotEmpty()) {
            try {
                val smsManager = SmsManager.getDefault()
                smsManager.sendTextMessage(phoneNumber, null, message, null, null)
                Toast.makeText(this, "SMS Sent Successfully!", Toast.LENGTH_SHORT).show()
            } catch (e: Exception) {
                Toast.makeText(this, "Failed to send SMS: ${e.message}", Toast.LENGTH_LONG).show()
            }
        } else {
            Toast.makeText(this, "Please enter phone number and message", Toast.LENGTH_SHORT).show()
        }
    }

    // Handle permission result
    override fun onRequestPermissionsResult(
        requestCode: Int, permissions: Array<out String>, grantResults: IntArray
    ) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults)

        if (requestCode == SMS_PERMISSION_CODE) {
            if (grantResults.isNotEmpty() && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                sendSMS()
            } else {
                Toast.makeText(this, "Permission denied", Toast.LENGTH_SHORT).show()
            }
        }
    }
}
```

Output



Allow **Send SMS** to send and view SMS messages?



Result:

The app sends SMS messages to a specified phone number by using the SMSManager class, with proper permissions and runtime handling and it is verified successfully.