**EXPERIMENT-8a Send SMS**

**AIM:**

To implement an application to Send SMS.

**PROCEDURE:**

* Go to File → New Project. f
* Provide the Application Name and click Next.
* Select the Target Android Devices.
* Choose the Minimum SDK required.
* Choose the activity (Blank Activity by default).
* Enter the Activity Name.
* Click Finish.
* Edit the program (design layout and write Kotlin code).
* Run the Application in 2 ways:
  1. Using an Emulator.
  2. Using a Real Mobile Device (via USB debugging).

**PROGRAM:**

## **AndroidManifest.xml :**

<?xml version="1.0" encoding="utf-8"?>

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

package="com.example.smsapp">

<!-- Permissions -->

<uses-permission android:name="android.permission.SEND\_SMS" />

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

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

<application

android:allowBackup="true"

android:label="Simple SMS Sender"

android:icon="@mipmap/ic\_launcher"

android:roundIcon="@mipmap/ic\_launcher\_round"

android:supportsRtl="true"

android:theme="@style/Theme.AppCompat.Light">

<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:orientation="vertical"

android:padding="24dp">

<EditText

android:id="@+id/editTextPhone"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Enter Phone Number"

android:inputType="phone"

android:padding="12dp" />

<EditText

android:id="@+id/editTextMessage"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Type your message"

android:inputType="textMultiLine"

android:minLines="3"

android:gravity="top"

android:padding="12dp"

android:layout\_marginTop="12dp"/>

<Button

android:id="@+id/buttonSend"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Send SMS"

android:layout\_marginTop="20dp"/>

</LinearLayout>

**MainActivity.kt :**

package com.example.smsapp

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 = 123

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 {

checkPermissionAndSend()

}

}

private fun checkPermissionAndSend() {

val phone = editTextPhone.text.toString().trim()

val message = editTextMessage.text.toString().trim()

if (phone.isEmpty() || message.isEmpty()) {

Toast.makeText(this, "Phone number and message cannot be empty", Toast.LENGTH\_SHORT).show()

return

}

if (ContextCompat.checkSelfPermission(this, Manifest.permission.SEND\_SMS)

!= PackageManager.PERMISSION\_GRANTED) {

ActivityCompat.requestPermissions(

this,

arrayOf(Manifest.permission.SEND\_SMS),

SMS\_PERMISSION\_CODE

)

} else {

sendSMS(phone, message)

}

}

private fun sendSMS(phone: String, message: String) {

try {

val smsManager = SmsManager.getDefault()

smsManager.sendTextMessage(phone, null, message, null, null)

Toast.makeText(this, "SMS sent successfully", Toast.LENGTH\_SHORT).show()

editTextPhone.text.clear()

editTextMessage.text.clear()

} catch (e: Exception) {

Toast.makeText(this, "Failed to send SMS: ${e.message}", Toast.LENGTH\_LONG).show()

e.printStackTrace()

}

}

override fun onRequestPermissionsResult(

requestCode: Int,

permissions: Array<String>,

grantResults: IntArray

) {

super.onRequestPermissionsResult(requestCode, permissions, grantResults)

if (requestCode == SMS\_PERMISSION\_CODE && grantResults.isNotEmpty()

&& grantResults[0] == PackageManager.PERMISSION\_GRANTED

) {

val phone = editTextPhone.text.toString().trim()

val message = editTextMessage.text.toString().trim()

sendSMS(phone, message)

} else {

Toast.makeText(this, "SMS permission denied", Toast.LENGTH\_SHORT).show()

}

}

}