Ex. No. 8	**Sending SMS and Email using Intent**
Date of Exercise	**18.09.2024**

Aim

To write a Kotlin program to send SMS and Email using Intent in android.

Description:

Steps to Implement Email Sending:

1. **Creating an Intent:** To send an email, we will create an Intent with the action Intent.ACTION_SENDTO. This action is used to send data to an app that can handle the specified data type. In our case, we will specify the email address and subject.
2. **Setting Email Data:** We will use the Uri class to define the email recipient, subject, and body of the email. The mailto: URI scheme will help format these details correctly.
3. **Starting the Email Client:** After setting up the email data, we will start the email client using startActivity(), which allows the user to choose their preferred email application. The email app will display the pre-filled email for the user to review and send.
4. **User Permissions:** While sending emails through an intent does not require specific permissions, it's good practice to inform users why you are initiating an email action, especially if your app collects any personal data.
5. **Error Handling:** It's essential to implement error handling to manage scenarios where no email client is available on the device. This can improve the robustness of your application and enhance user experience.
6. **Testing on Different Devices:** Make sure to test the email functionality on various devices and email clients to ensure compatibility and that the pre-filled fields are displayed correctly.
7. **UI Integration:** Consider integrating this email functionality into your app's user interface, such as a contact or feedback section, to make it easily accessible for users.

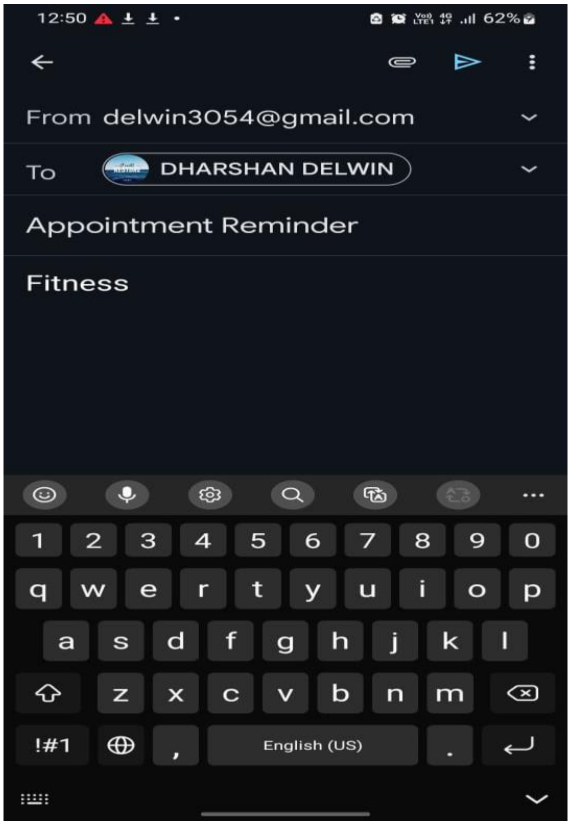
8. **Security Best Practices:** Avoid hardcoding sensitive information like email addresses in your code. Instead, consider using a configuration file or environment variables to manage such data securely.

Program

MainAcrivity code:

```
package com.example.ex_8

import android.Manifest
import android.content.Intent
import android.content.pm.PackageManager
import android.os.Bundle
import android.provider.Telephony
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 com.example.ex_8.R

class MainActivity : AppCompatActivity() {

    private lateinit var appointmentDetails: EditText
    private lateinit var recipientEmail: EditText
    private lateinit var recipientPhone: EditText
    private lateinit var sendEmailButton: Button
    private lateinit var sendSmsButton: Button

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        appointmentDetails = findViewById(R.id.appointmentDetails)
        recipientEmail = findViewById(R.id.recipientEmail)
        recipientPhone = findViewById(R.id.recipientPhone)
        sendEmailButton = findViewById(R.id.sendEmailButton)
        sendSmsButton = findViewById(R.id.sendSmsButton)

        sendEmailButton.setOnClickListener {
            sendEmail()
        }

        sendSmsButton.setOnClickListener {
            sendSms()
        }

        // Request SMS permission
        ActivityCompat.requestPermissions(this,
            arrayOf(Manifest.permission.SEND_SMS),
            1)
    }

    private fun sendEmail() {
        val email = recipientEmail.text.toString()
        val subject = "Appointment Reminder"
        val message = appointmentDetails.text.toString()

        val intent = Intent(Intent.ACTION_SEND).apply {
            type = "message/rfc822"
            putExtra(Intent.EXTRA_EMAIL, arrayOf(email))
            putExtra(Intent.EXTRA_SUBJECT, subject)
            putExtra(Intent.EXTRA_TEXT, message)
        }

        try {
            startActivity(Intent.createChooser(intent, "Send Email..."))
        } catch (ex: android.content.ActivityNotFoundException) {
            Toast.makeText(this, "No email client installed.", Toast.LENGTH_SHORT).show()
        }
    }

    private fun sendSms() {
        val phoneNumber = recipientPhone.text.toString()
        val message = appointmentDetails.text.toString()

        if (ActivityCompat.checkSelfPermission(this, Manifest.permission.SEND_SMS) !=
PackageManager.PERMISSION_GRANTED) {
            Toast.makeText(this, "SMS permission not granted", Toast.LENGTH_SHORT).show()
            return
        }

        try {
            SmsManager.getDefault().sendTextMessage(phoneNumber, null, message, null, null)
            Toast.makeText(this, "SMS sent.", Toast.LENGTH_SHORT).show()
        } catch (ex: Exception) {
            Toast.makeText(this, "Failed to send SMS: ${ex.message}", Toast.LENGTH_SHORT).show()
        }
    }
}
```

Output:

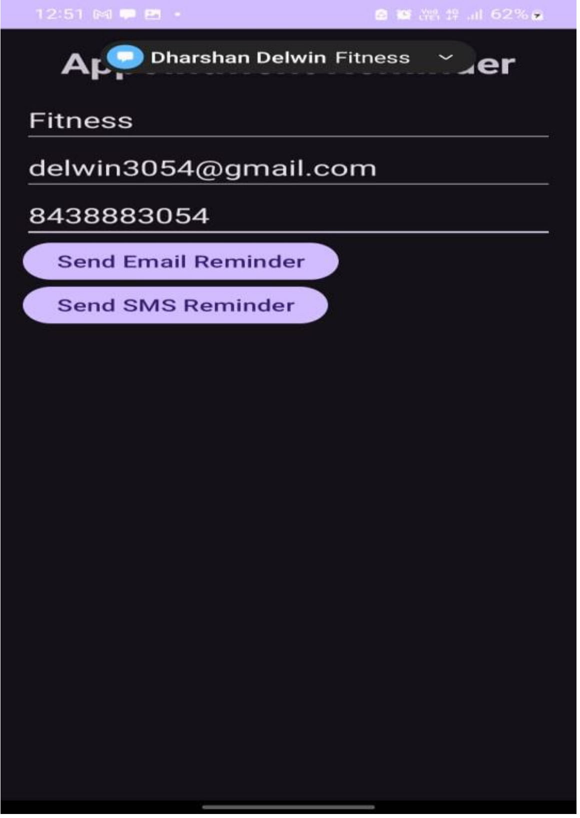


From delwin3054@gmail.com

To DHARSHAN DELWIN

Appointment Reminder

Fitness



Fitness

delwin3054@gmail.com

8438883054

Result:

Thus the code is executed successfully and the output is displayed in the console window.