



UNIVERSITY MALAYSIA TERENGGANU
FACULTY OF COMPUTER SCIENCE & MATHEMATICS

NATIVE MOBILE PROGRAMMING
CSM 3123

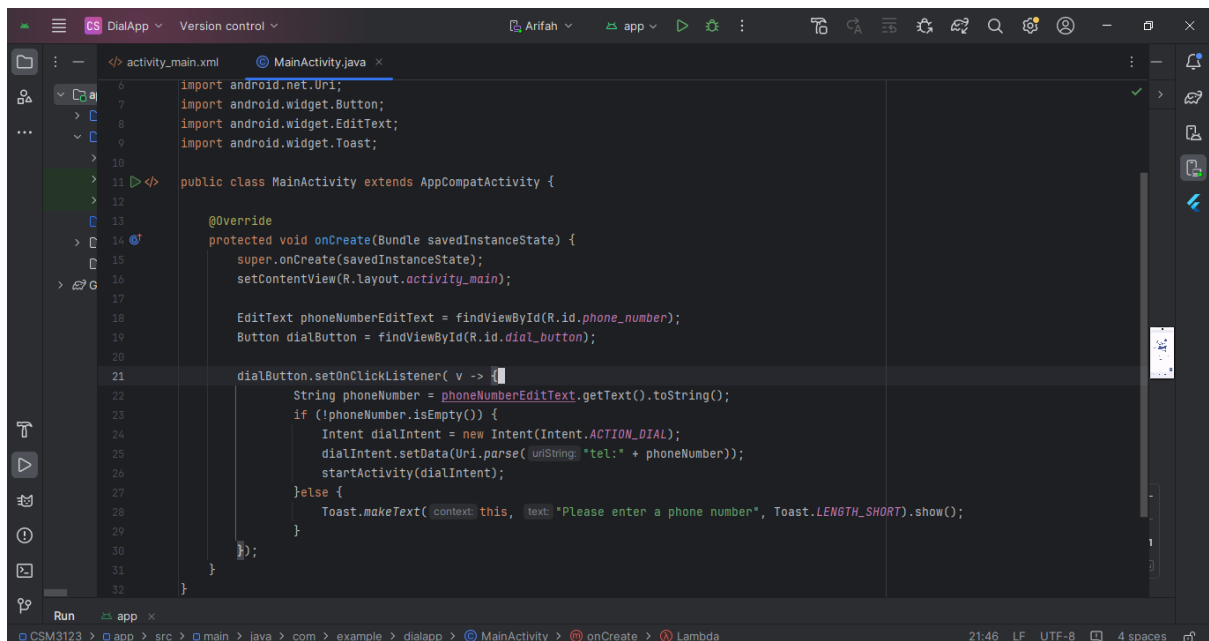
LAB REPORT 5
<https://github.com/arfhnfh/CSM3123-Arifah-S66428-Lab5>

Prepared by:
NUR ARIFAH BINTI MOHD HANAFIAH
S66428

Prepared for:
DR RABIEI BIN MAMAT

BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING) WITH HONORS
SEMESTER I 2024/2025

Activity 1

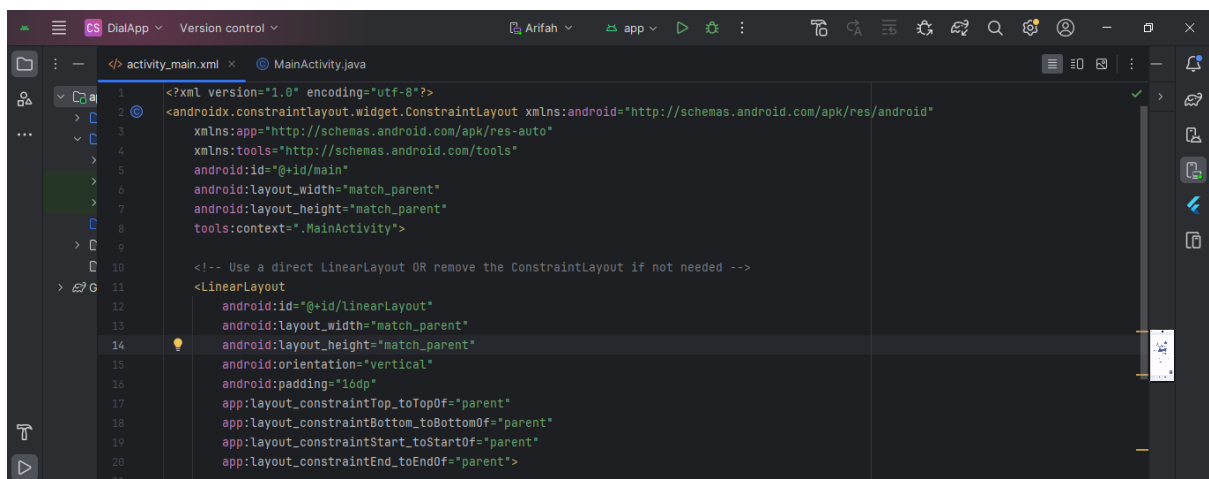


```
1  import android.net.Uri;
2  import android.widget.Button;
3  import android.widget.EditText;
4  import android.widget.Toast;
5
6  public class MainActivity extends AppCompatActivity {
7
8      @Override
9      protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12
13         EditText phoneNumberEditText = findViewById(R.id.phone_number);
14         Button dialButton = findViewById(R.id.dial_button);
15
16         dialButton.setOnClickListener( v -> {
17             String phoneNumber = phoneNumberEditText.getText().toString();
18             if (!phoneNumber.isEmpty()) {
19                 Intent dialIntent = new Intent(Intent.ACTION_DIAL);
20                 dialIntent.setData(Uri.parse("tel:" + phoneNumber));
21                 startActivity(dialIntent);
22             } else {
23                 Toast.makeText(context, this, "Please enter a phone number", Toast.LENGTH_SHORT).show();
24             }
25         });
26     }
27 }
```

Run app

CSM3123 > app > src > main > java > com > example > dialapp > MainActivity > onCreate > Lambda

21:46 LF UTF-8 4 spaces

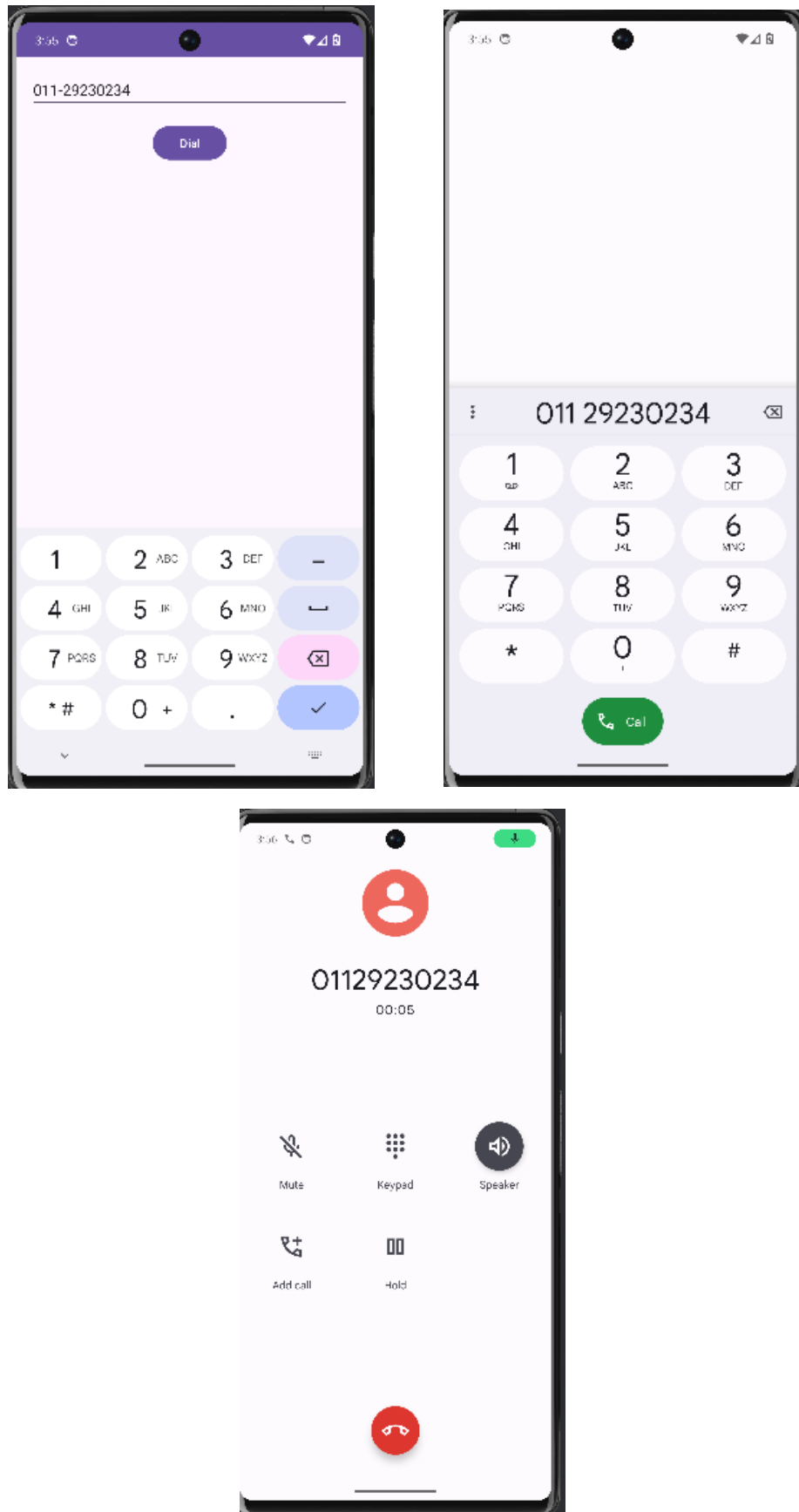


```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:id="@+id/main"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      tools:context=".MainActivity">
9
10     <!-- Use a direct LinearLayout OR remove the ConstraintLayout if not needed -->
11     <LinearLayout
12         android:id="@+id/linearLayout"
13         android:layout_width="match_parent"
14         android:layout_height="match_parent"
15         android:orientation="vertical"
16         android:padding="16dp"
17         app:layout_constraintTop_toTopOf="parent"
18         app:layout_constraintBottom_toBottomOf="parent"
19         app:layout_constraintStart_toStartOf="parent"
20         app:layout_constraintEnd_toEndOf="parent">
```

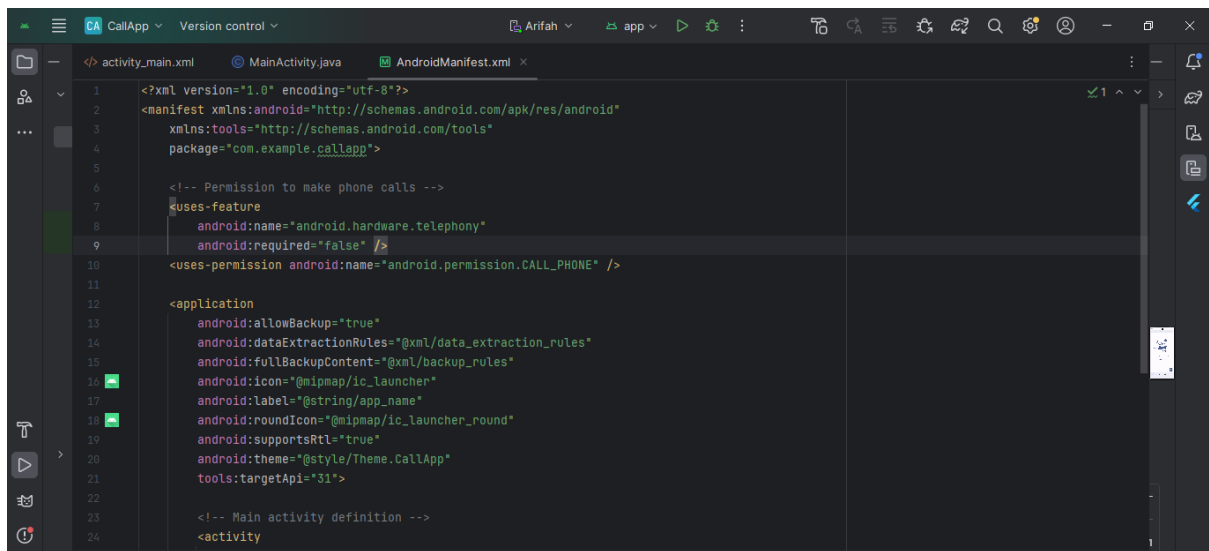


```
21     <EditText
22         android:id="@+id/phone_number"
23         android:layout_width="match_parent"
24         android:layout_height="wrap_content"
25         android:hint="Enter phone number"
26         android:inputType="phone" />
27
28     <Button
29         android:id="@+id/dial_button"
30         android:layout_width="wrap_content"
31         android:layout_height="wrap_content"
32         android:text="Dial"
33         android:layout_gravity="center_horizontal"
34         android:layout_marginTop="16dp" />
35
36     </LinearLayout>
37
38 </androidx.constraintlayout.widget.ConstraintLayout>
```

Output



Activity 2



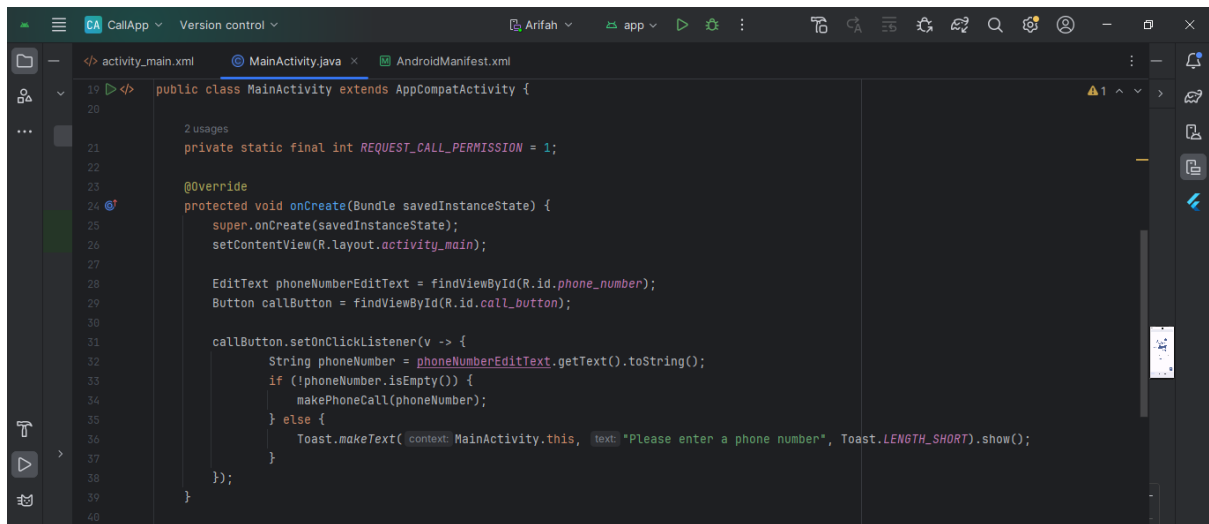
This screenshot shows the `AndroidManifest.xml` file in an IDE. The file is for an application named `com.example.callapp`. It includes a permission to make phone calls and defines the main activity.

```
<?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.callapp">

    <!-- Permission to make phone calls -->
    <uses-feature
        android:name="android.hardware.telephony"
        android:required="false" />
    <uses-permission android:name="android.permission.CALL_PHONE" />

    <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:supportRtl="true"
        android:theme="@style/Theme.CallApp"
        tools:targetApi="31">

        <!-- Main activity definition -->
        <activity
```



This screenshot shows the `MainActivity.java` file in an IDE. The class extends `AppCompatActivity` and implements the `onCreate` method. It also includes a click listener for the `callButton` that checks for the `CALL_PHONE` permission and makes a phone call if it is granted.

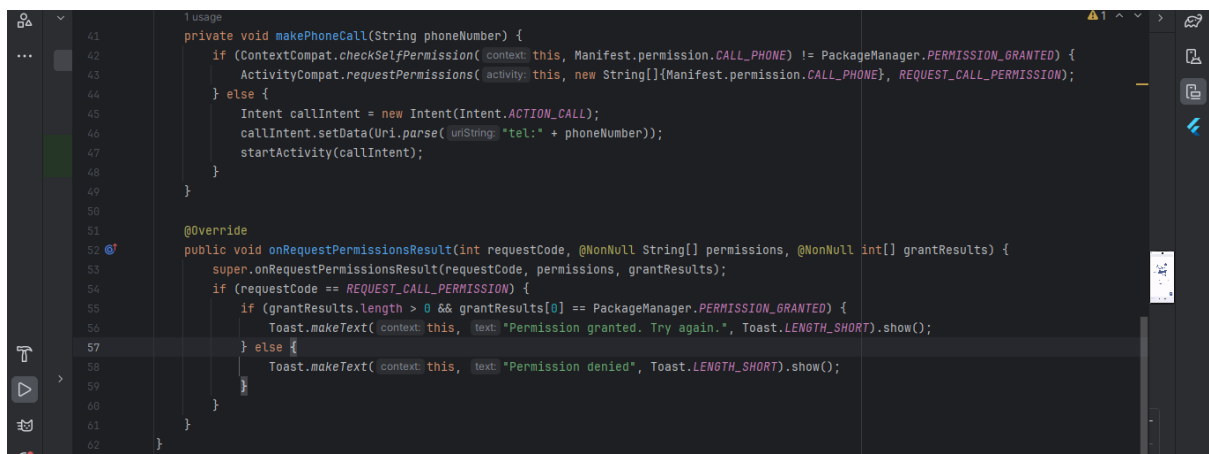
```
public class MainActivity extends AppCompatActivity {

    2 usages
    private static final int REQUEST_CALL_PERMISSION = 1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        EditText phoneNumberEditText = findViewById(R.id.phone_number);
        Button callButton = findViewById(R.id.call_button);

        callButton.setOnClickListener(v -> {
            String phoneNumber = phoneNumberEditText.getText().toString();
            if (!phoneNumber.isEmpty()) {
                makePhoneCall(phoneNumber);
            } else {
                Toast.makeText(context, MainActivity.this, "Please enter a phone number", Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

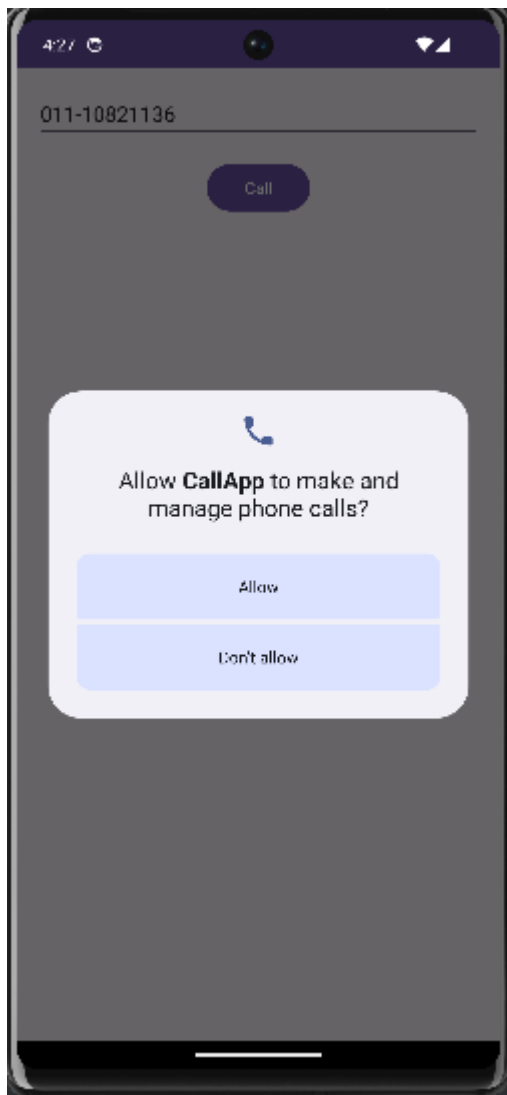


This screenshot shows the `makePhoneCall` and `onRequestPermissionsResult` methods in the `MainActivity.java` file. The `makePhoneCall` method checks for the `CALL_PHONE` permission and makes a phone call if it is granted. The `onRequestPermissionsResult` method handles the result of the permission request.

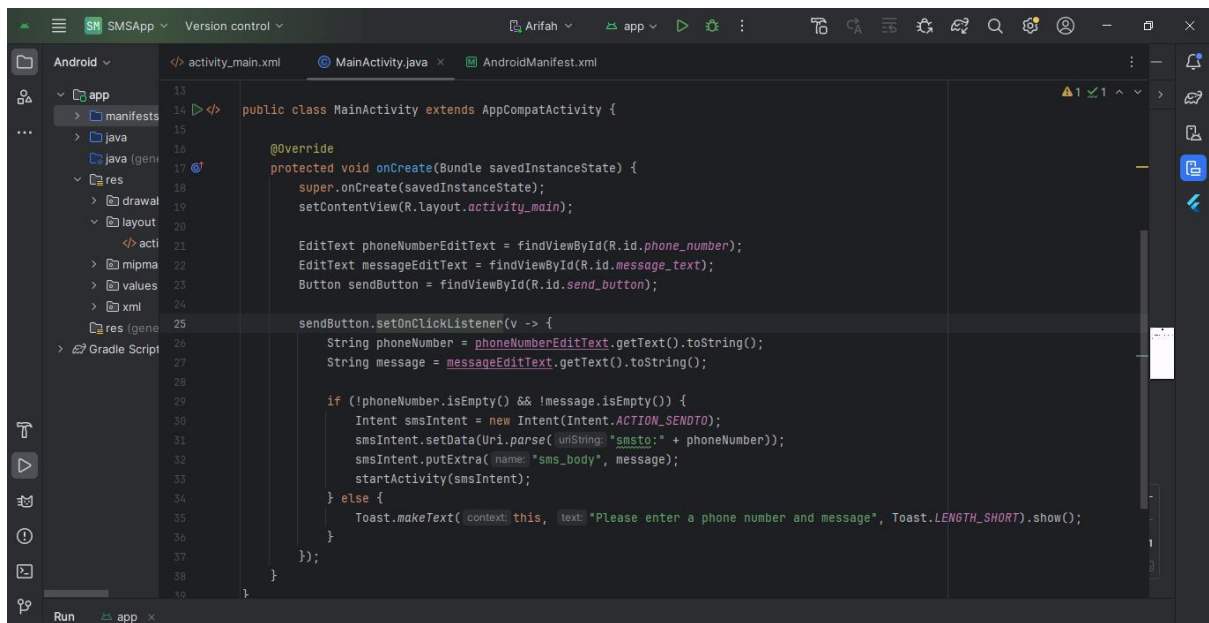
```
1 usage
private void makePhoneCall(String phoneNumber) {
    if (ContextCompat.checkSelfPermission(context, Manifest.permission.CALL_PHONE) != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(activity, this, new String[]{Manifest.permission.CALL_PHONE}, REQUEST_CALL_PERMISSION);
    } else {
        Intent callIntent = new Intent(Intent.ACTION_CALL);
        callIntent.setData(Uri.parse("tel:" + phoneNumber));
        startActivity(callIntent);
    }
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if (requestCode == REQUEST_CALL_PERMISSION) {
        if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            Toast.makeText(context, this, "Permission granted. Try again.", Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(context, this, "Permission denied", Toast.LENGTH_SHORT).show();
        }
    }
}
```

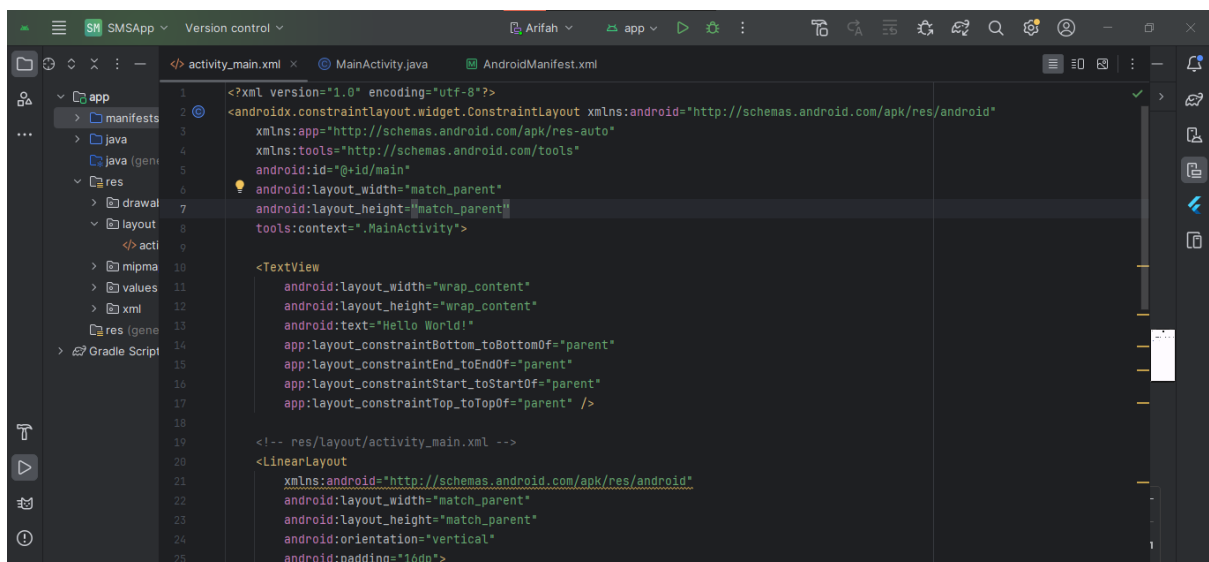
Output



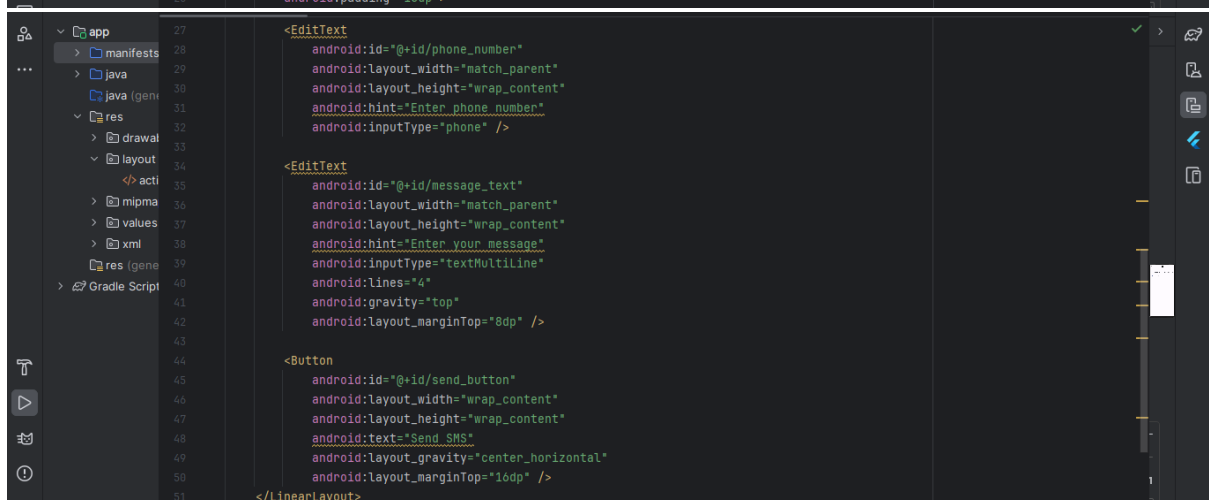
Activity 3



```
13 public class MainActivity extends AppCompatActivity {
14
15     @Override
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.activity_main);
19
20         EditText phoneNumberEditText = findViewById(R.id.phone_number);
21         EditText messageEditText = findViewById(R.id.message_text);
22         Button sendButton = findViewById(R.id.send_button);
23
24         sendButton.setOnClickListener(v -> {
25             String phoneNumber = phoneNumberEditText.getText().toString();
26             String message = messageEditText.getText().toString();
27
28             if (!phoneNumber.isEmpty() && !message.isEmpty()) {
29                 Intent smsIntent = new Intent(Intent.ACTION_SENDTO);
30                 smsIntent.setData(Uri.parse("sms:" + phoneNumber));
31                 smsIntent.putExtra("sms_body", message);
32                 startActivity(smsIntent);
33             } else {
34                 Toast.makeText(this, "Please enter a phone number and message", Toast.LENGTH_SHORT).show();
35             }
36         });
37     }
38 }
```

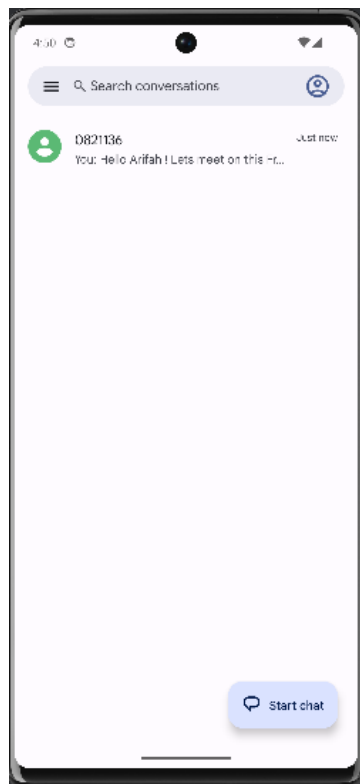
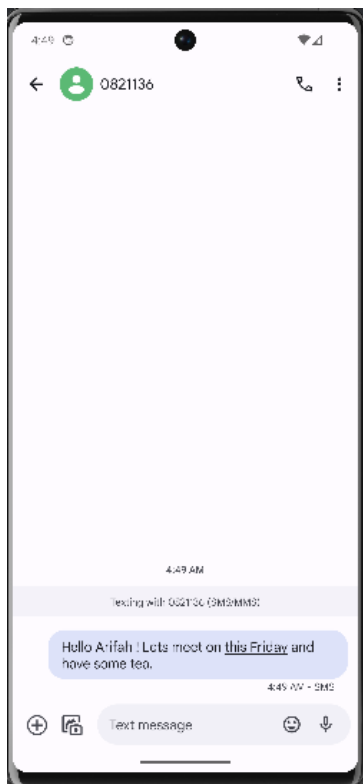
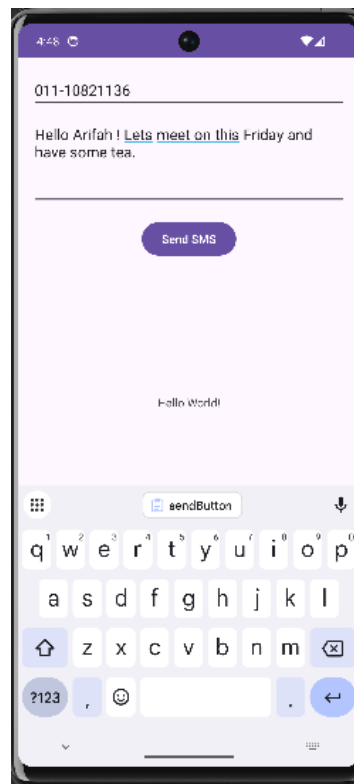
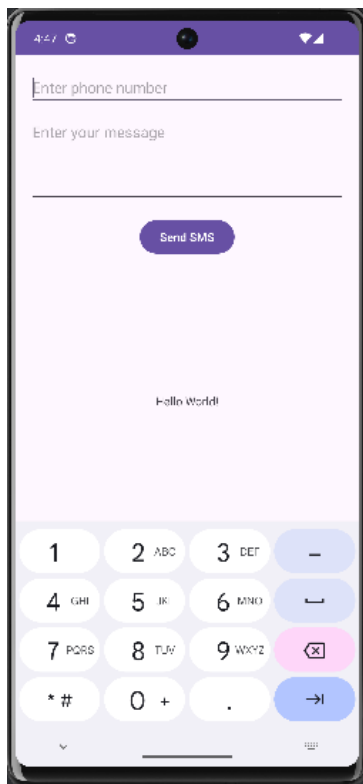


```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/main"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context=".MainActivity">
9
10     <TextView
11         android:layout_width="wrap_content"
12         android:layout_height="wrap_content"
13         android:text="Hello World!"
14         app:layout_constraintBottom_toBottomOf="parent"
15         app:layout_constraintEnd_toEndOf="parent"
16         app:layout_constraintStart_toStartOf="parent"
17         app:layout_constraintTop_toTopOf="parent" />
18
19     <!-- res/layout/activity_main.xml -->
20     <LinearLayout
21         xmlns:android="http://schemas.android.com/apk/res/android"
22         android:layout_width="match_parent"
23         android:layout_height="match_parent"
24         android:orientation="vertical"
25         android:padding="16dp">
```

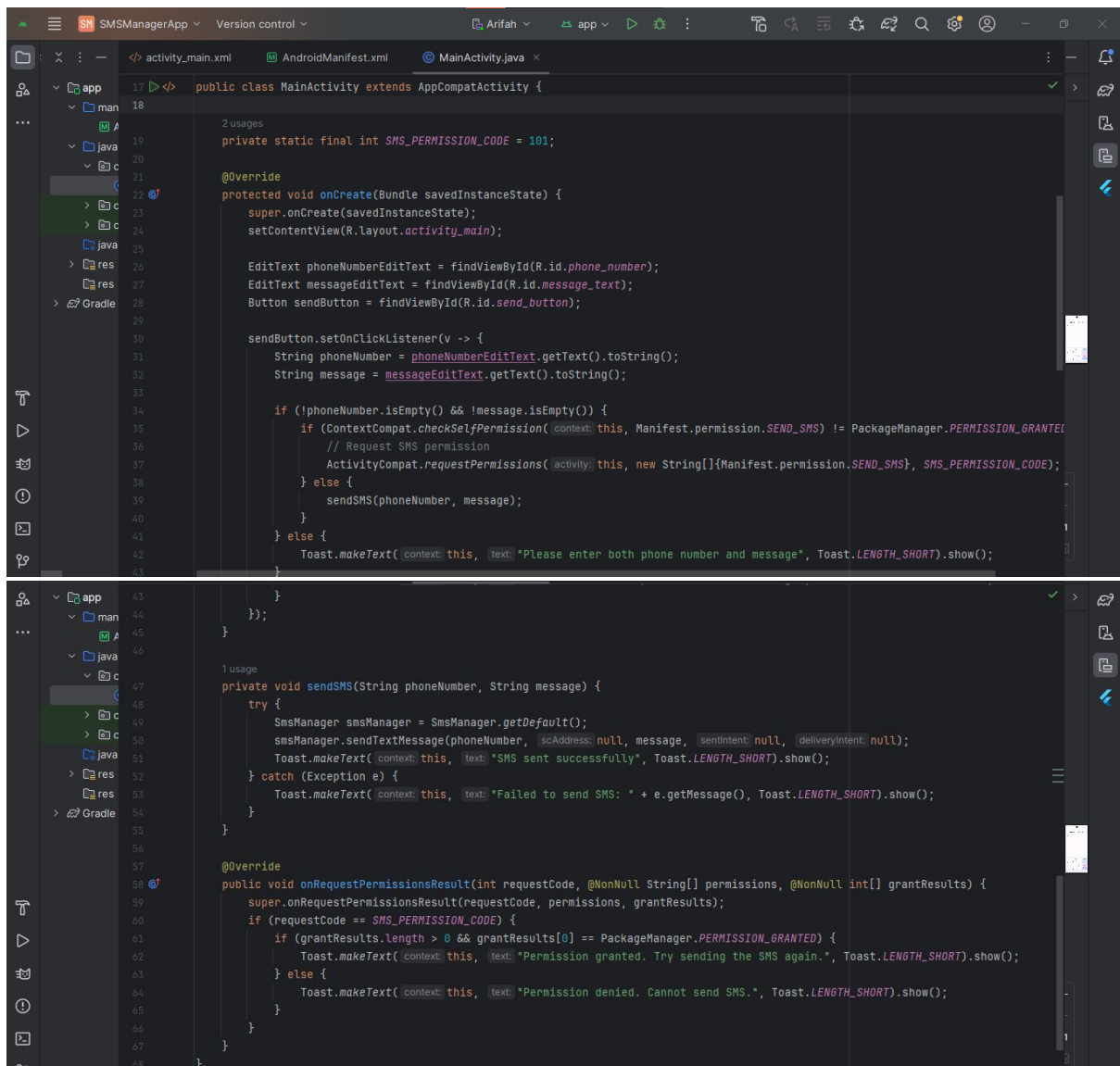


```
26     <EditText
27         android:id="@+id/phone_number"
28         android:layout_width="match_parent"
29         android:layout_height="wrap_content"
30         android:hint="Enter phone number"
31         android:inputType="phone" />
32
33     <EditText
34         android:id="@+id/message_text"
35         android:layout_width="match_parent"
36         android:layout_height="wrap_content"
37         android:hint="Enter your message"
38         android:inputType="textMultiline"
39         android:lines="4"
40         android:gravity="top"
41         android:layout_marginTop="8dp" />
42
43     <Button
44         android:id="@+id/send_button"
45         android:layout_width="wrap_content"
46         android:layout_height="wrap_content"
47         android:text="Send SMS"
48         android:layout_gravity="center_horizontal"
49         android:layout_marginTop="16dp" />
50 </LinearLayout>
```

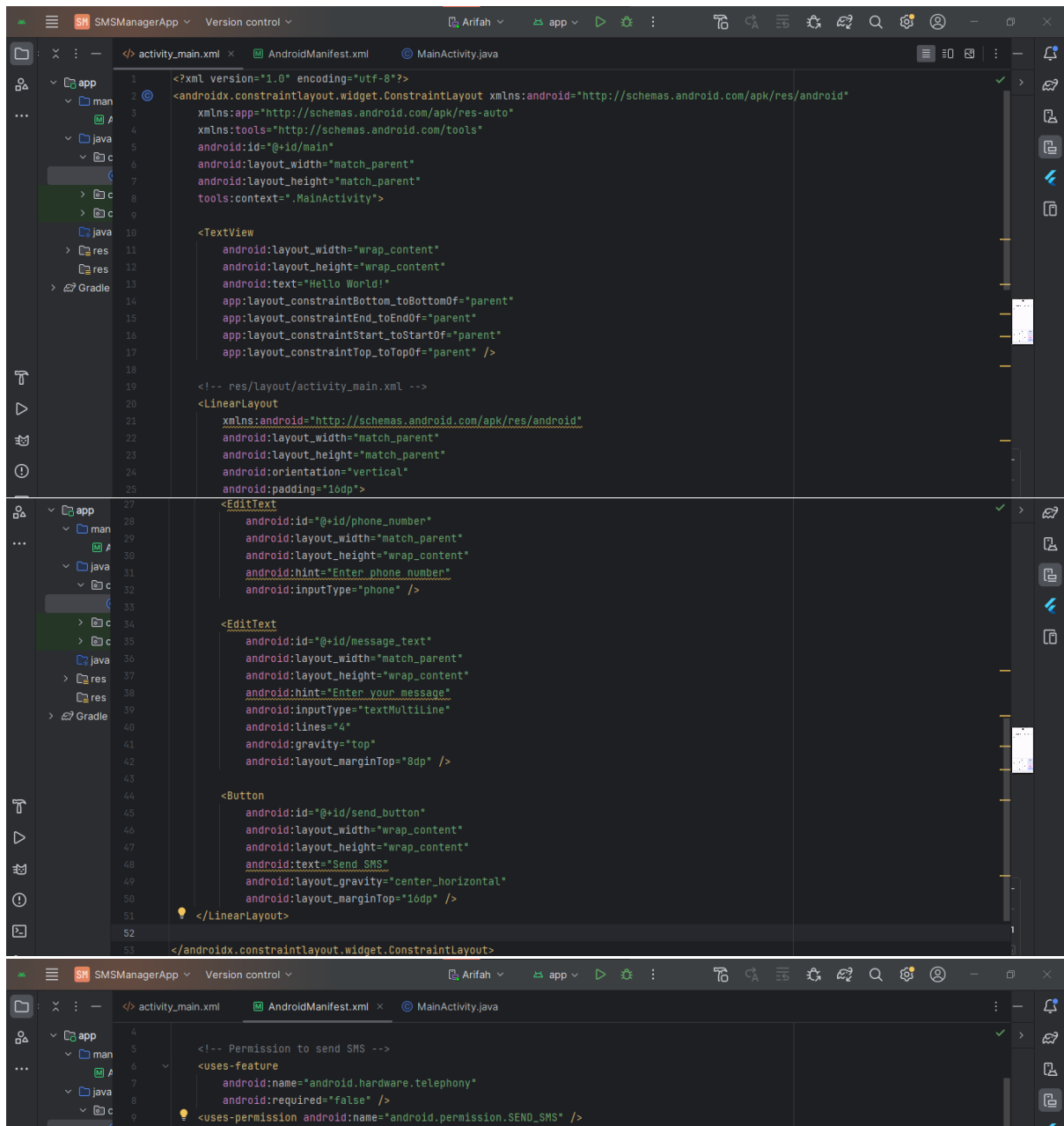
Output



Activity 4



```
17 <?xml version="1.0" encoding="utf-8" >
18 <activity android:name=".MainActivity"
19     android:label="@string/app_name"
20     android:theme="@style/Theme.AppCompat" >
21     <include layout="@layout/activity_main" />
22 </activity>
23 </xml>
24
25 package com.example.smsmanagerapp;
26
27 import androidx.appcompat.app.AppCompatActivity;
28 import android.Manifest;
29 import android.content.Context;
30 import android.content.pm.PackageManager;
31 import android.os.Bundle;
32 import android.provider.Settings;
33 import android.telephony.SmsManager;
34 import android.telephony.SmsMessage;
35 import android.view.View;
36 import android.view.View.OnClickListener;
37 import android.widget.Button;
38 import android.widget.EditText;
39 import android.widget.Toast;
40
41 public class MainActivity extends AppCompatActivity {
42
43     private static final int SMS_PERMISSION_CODE = 101;
44
45     @Override
46     protected void onCreate(Bundle savedInstanceState) {
47         super.onCreate(savedInstanceState);
48         setContentView(R.layout.activity_main);
49
50         EditText phoneNumberEditText = findViewById(R.id.phone_number);
51         EditText messageEditText = findViewById(R.id.message_text);
52         Button sendButton = findViewById(R.id.send_button);
53
54         sendButton.setOnClickListener(v -> {
55             String phoneNumber = phoneNumberEditText.getText().toString();
56             String message = messageEditText.getText().toString();
57
58             if (!phoneNumber.isEmpty() && !message.isEmpty()) {
59                 if (ContextCompat.checkSelfPermission(this, Manifest.permission.SEND_SMS) != PackageManager.PERMISSION_GRANTED) {
60                     // Request SMS permission
61                     ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.SEND_SMS, SMS_PERMISSION_CODE});
62                 } else {
63                     sendSMS(phoneNumber, message);
64                 }
65             } else {
66                 Toast.makeText(this, "Please enter both phone number and message", Toast.LENGTH_SHORT).show();
67             }
68         });
69     }
70
71     private void sendSMS(String phoneNumber, String message) {
72         try {
73             SmsManager smsManager = SmsManager.getDefault();
74             smsManager.sendTextMessage(phoneNumber, null, message, null, null);
75             Toast.makeText(this, "SMS sent successfully", Toast.LENGTH_SHORT).show();
76         } catch (Exception e) {
77             Toast.makeText(this, "Failed to send SMS: " + e.getMessage(), Toast.LENGTH_SHORT).show();
78         }
79     }
80
81     @Override
82     public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
83         super.onRequestPermissionsResult(requestCode, permissions, grantResults);
84         if (requestCode == SMS_PERMISSION_CODE) {
85             if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
86                 Toast.makeText(this, "Permission granted. Try sending the SMS again.", Toast.LENGTH_SHORT).show();
87             } else {
88                 Toast.makeText(this, "Permission denied. Cannot send SMS.", Toast.LENGTH_SHORT).show();
89             }
90         }
91     }
92 }
```

Output

