Experiment2.2

Student Name: Sanchit Singal **UID: 21BCS1569**
Branch: CSE **Section/Group: 606/B**
Semester: 6th **Date of Performance: 01/03/24**
Subject Name: Mobile App Development **Subject Code: 21CSH-355**

1. **Aim:** To design an android application Send SMS using Intent.

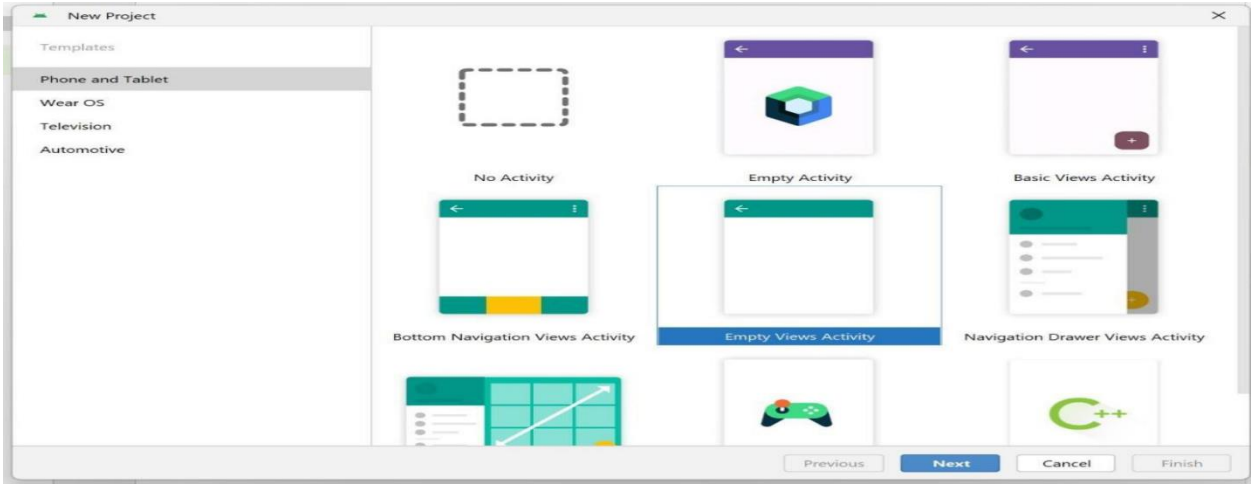
2. **Objective:** The objective of an Android-based application that uses Intent to send SMS can be to create a convenient and user-friendly tool for sending text messages. This type of app aims to leverage the Android platform's capabilities to provide a seamless and efficient way for users to compose and send SMS messages.

3. **Input/Apparatus Used:**

- **Android Studio:** The official IDE for Android development. Download and install Android Studio from the official website: Android Studio.
- **Android SDK:** The Android Software Development Kit (SDK) is essential for developing Android applications. Android Studio usually comes bundled with the SDK, but you may need to update it through the SDK Manager within Android Studio.
- **Java Development Kit (JDK):** Android apps are primarily written in Java or Kotlin. Make sure you have the Java Development Kit installed. Android Studio supports JDK. You can download it from the Oracle website: Java SE Downloads.
- **Android Virtual Device (AVD) or Physical Android Device:** You need a device to test your Android application. You can use an emulator (AVD) that comes with Android Studio or a physical Android device connected to your computer.

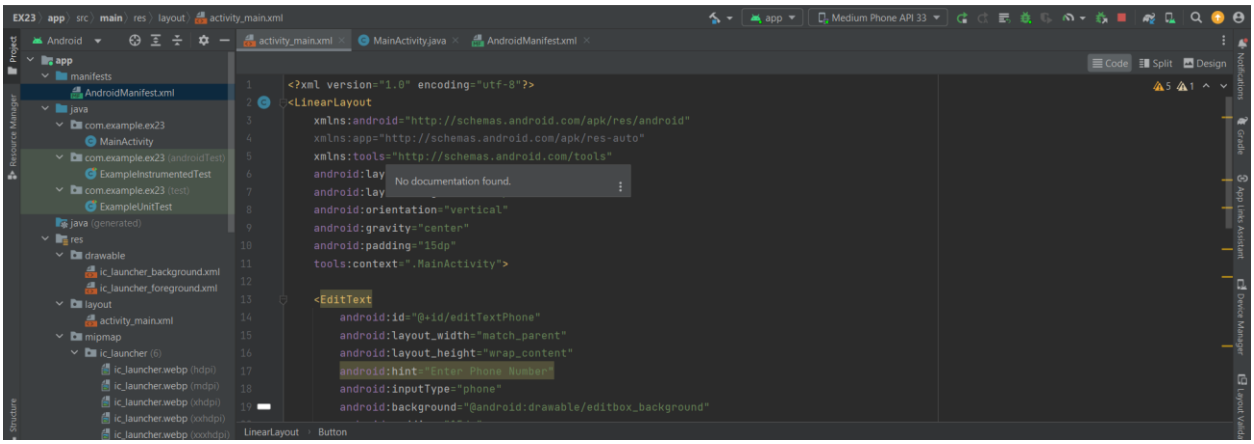
4. **Procedure:**

Step 1: Open Android Studio: Open Android Studio and create a new project.

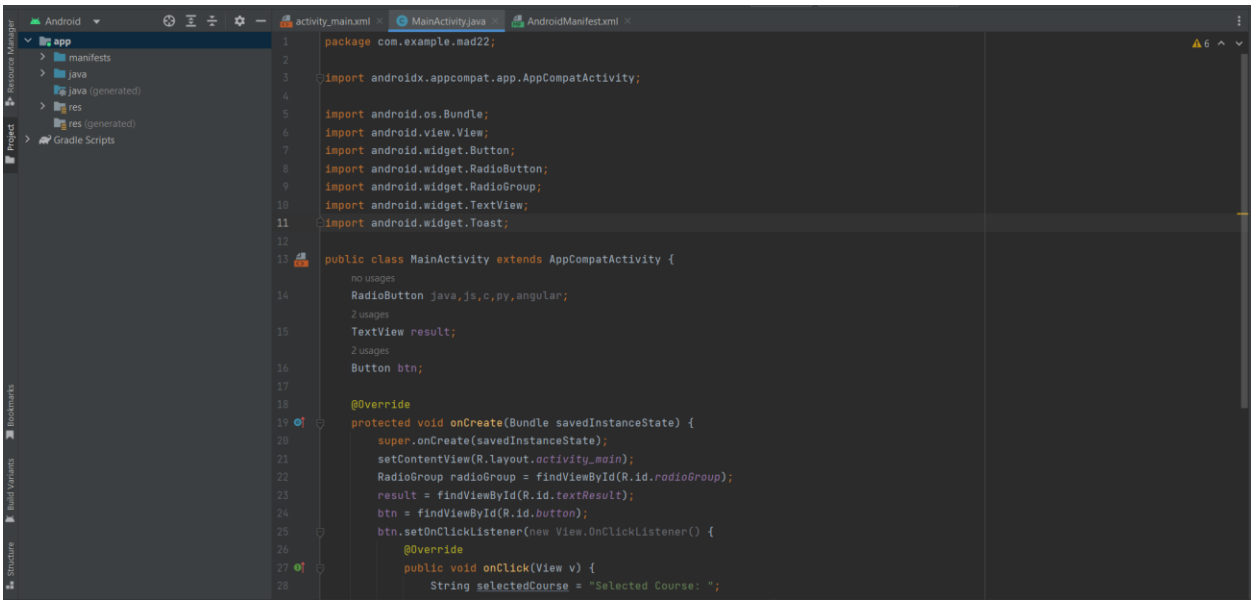


Step 2: Open the "activity_main.xml" file and add the following widgets in aRelative Layout:

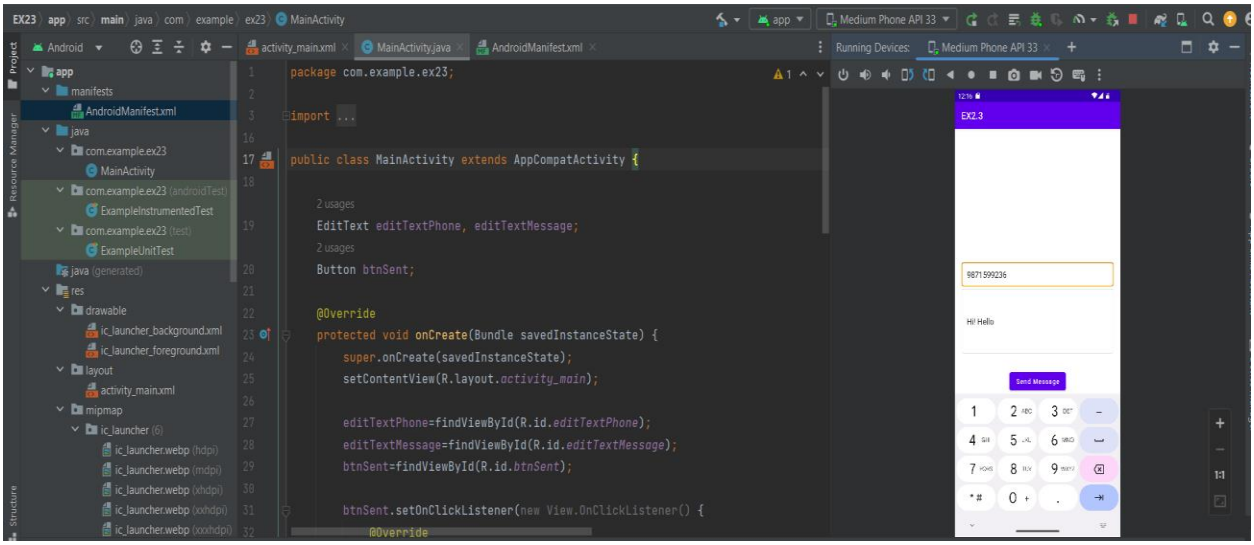
- Two Edit Text to get the number and message
- A Send message button.



Step 3: Now, after the UI, this step will create the Backend of Application. For this, open the "MainActivity.java" file and instantiate the components made in the XML file (TextView and Send Button) using findViewById() method. This method binds the created object to the UI Components with the help of the assigned ID.



Step 4: Create an emulated virtual device in Device Manager and Run the App:



SOURCECODE:

MainActivity:

```
package com.example.ex23;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import android.Manifest;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
  EditText editTextPhone, editTextMessage;
  Button btnSent;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    editTextPhone=findViewById(R.id.editTextPhone);
    editTextMessage=findViewById(R.id.editTextMessage);
    btnSent=findViewById(R.id.btnSent);
    btnSent.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        if(ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.SEND_SMS)
            == PackageManager.PERMISSION_GRANTED){
          sendSMS();
        }
        else{
          ActivityCompat.requestPermissions(MainActivity.this,new
String[]{Manifest.permission.SEND_SMS},100);
        }
      }
    });
  }
```

```
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
@NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        if(requestCode==100 && grantResults.length>0 &&
grantResults[0]==PackageManager.PERMISSION_GRANTED){
            sendSMS();
        }
        else{
            Toast.makeText(this,"Permission Denied", Toast.LENGTH_SHORT).show();
        }
    }

    private void sendSMS(){
        String phone=editTextPhone.getText().toString();
        String message=editTextMessage.getText().toString();
        if(!phone.isEmpty() && !message.isEmpty()){
            SmsManager smsManager=SmsManager.getDefault();
            smsManager.sendTextMessage(phone,null,message,null,null);
            Toast.makeText(this,"SMS sent successfully",Toast.LENGTH_SHORT).show();
        }
        else{
            Toast.makeText(this, "Please enter number and
message",Toast.LENGTH_SHORT).show();
        }
    }
}
```

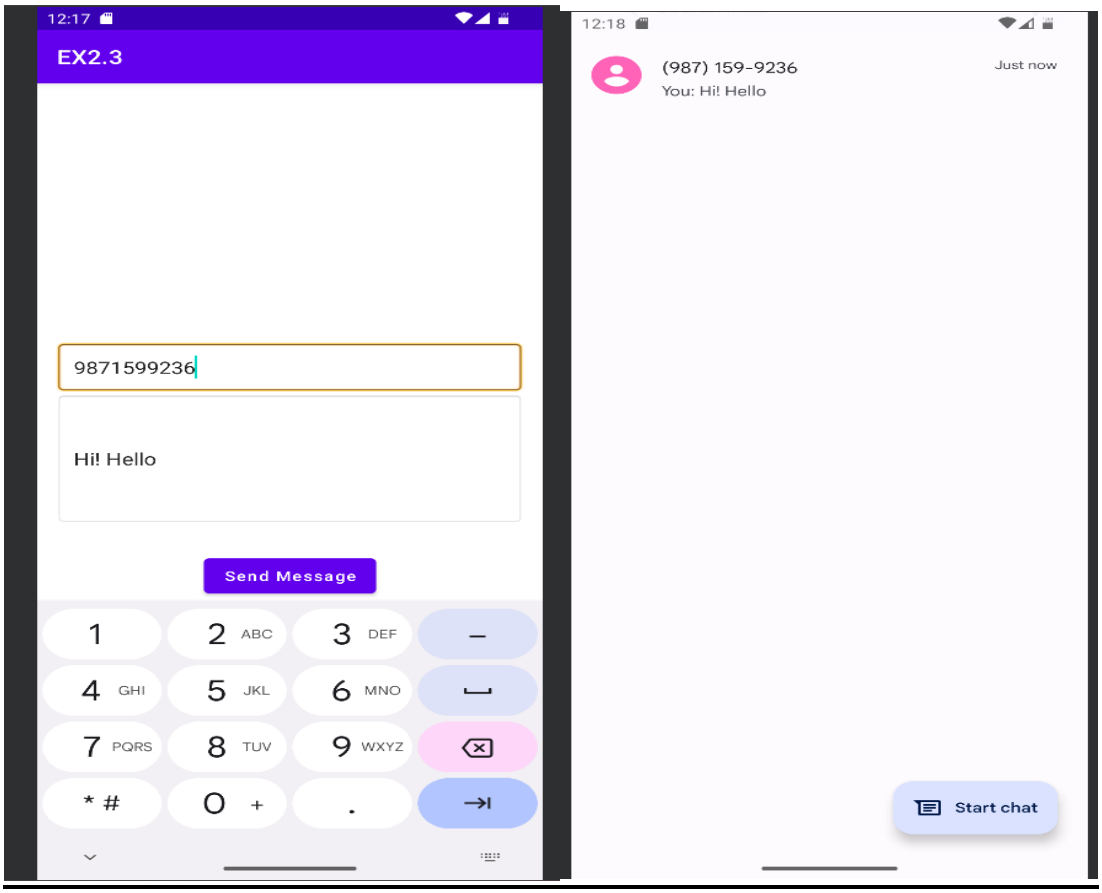
Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="15dp"
```

```
  tools:context=".MainActivity">
<EditText
    android:id="@+id/editTextPhone"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter Phone Number"
    android:inputType="phone"
    android:background="@android:drawable/editbox_background"
    android:padding="15dp"
    android:maxLength="10" />
<EditText
    android:id="@+id/editTextMessage"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter Message"
    android:inputType="textMultiLine"
    android:background="@android:drawable/editbox_background"
    android:padding="15dp"
    android:lines="5" />
<Button
    android:id="@+id/btnSent"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Send Message"
    android:textAllCaps="false"
    android:layout_marginTop="30dp" />


</LinearLayout>
```

5. Output:



6. Learning Outcomes:

1. I have learned the process of installing Android Studio, a tool for Android app development.
2. I understand the importance of configuring SDKs and virtual devices for a smooth development environment.
3. I now understand the significance of testing applications on a virtual device, ensuring a well-prepared development setup.