



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment-2.3

Student Name: Atanu Bera

UID: 20BCS2564

Branch: CSE

Section/Group: 707/A

Semester: 6th

Date of Performance: 10/04/2023

Subject Name: Mobile Application Development Lab

Subject Code: 20CSP-356

1. Aim:

To design an android application, Send SMS using Intent.

2. Apparatus / Simulator Used:

- Linux OS/ Windows 7 or above
- Android Studio
- Ram 4 GB and above
- Java (Including JDK & JRE)

3. Objective:

- To understand the concept of SMS.
- To implement the TextView in Android Studio.
- To implement the Button in Android Studio.
- To implement the Intent in Android Studio.

4. Code:

MainActivity.java

```
package com.example.sendsms;

import android.os.Bundle;
import android.app.Activity;
import android.app.PendingIntent;
import android.content.Intent;
import android.telephony.SmsManager;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends Activity {

    EditText mobilenos,message;
    Button sendsms;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mobilenos=(EditText)findViewById(R.id.editText1);
        message=(EditText)findViewById(R.id.editText2);
        sendsms=(Button)findViewById(R.id.button1);

        //Performing action on button click
```

```
sendsms.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View arg0) {
        String no=mobilenumber.getText().toString();
        String msg=message.getText().toString();

        //Getting intent and PendingIntent instance
        Intent intent=new
Intent(getApplicationContext(),MainActivity.class);
        PendingIntent pi=PendingIntent.getActivity(getApplicationContext(),
0, intent,0);

        //Get the SmsManager instance and call the sendTextMessage method
to send message
        SmsManager sms=SmsManager.getDefault();
        sms.sendTextMessage(no, null, msg, pi,null);

        Toast.makeText(getApplicationContext(), "Message Sent
successfully!",
        Toast.LENGTH_LONG).show();
    }
});
}
```



```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.activity_main, menu);
    return true;
}
```

```
}
```

activity_main.xml

```
<RelativeLayout
```

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

```
xmlns:tools="http://schemas.android.com/tools"
```

```
android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
```

```
tools:context=".MainActivity" >
```

```
<EditText
```

```
android:id="@+id/editText1"
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:layout_alignParentRight="true"
```

```
android:layout_alignParentTop="true"
```

```
android:layout_marginRight="20dp"
```

```
android:ems="10" />
```

```
<EditText
```

```
android:id="@+id/editText2"
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:layout_alignLeft="@+id/editText1"
```

```
android:layout_below="@+id/editText1"
```

```
android:layout_marginTop="26dp"
```

```
android:ems="10"
```

```
android:inputType="textMultiLine" />
```

```
<TextView
```

```
android:id="@+id/textView1"
```

```
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignBaseline="@+id/editText1"  
android:layout_alignBottom="@+id/editText1"  
android:layout_toLeftOf="@+id/editText1"  
android:text="Mobile No:" />
```

<TextView

```
android:id="@+id/textView2"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignBaseline="@+id/editText2"  
android:layout_alignBottom="@+id/editText2"  
android:layout_alignLeft="@+id/textView1"  
android:text="Message:" />
```

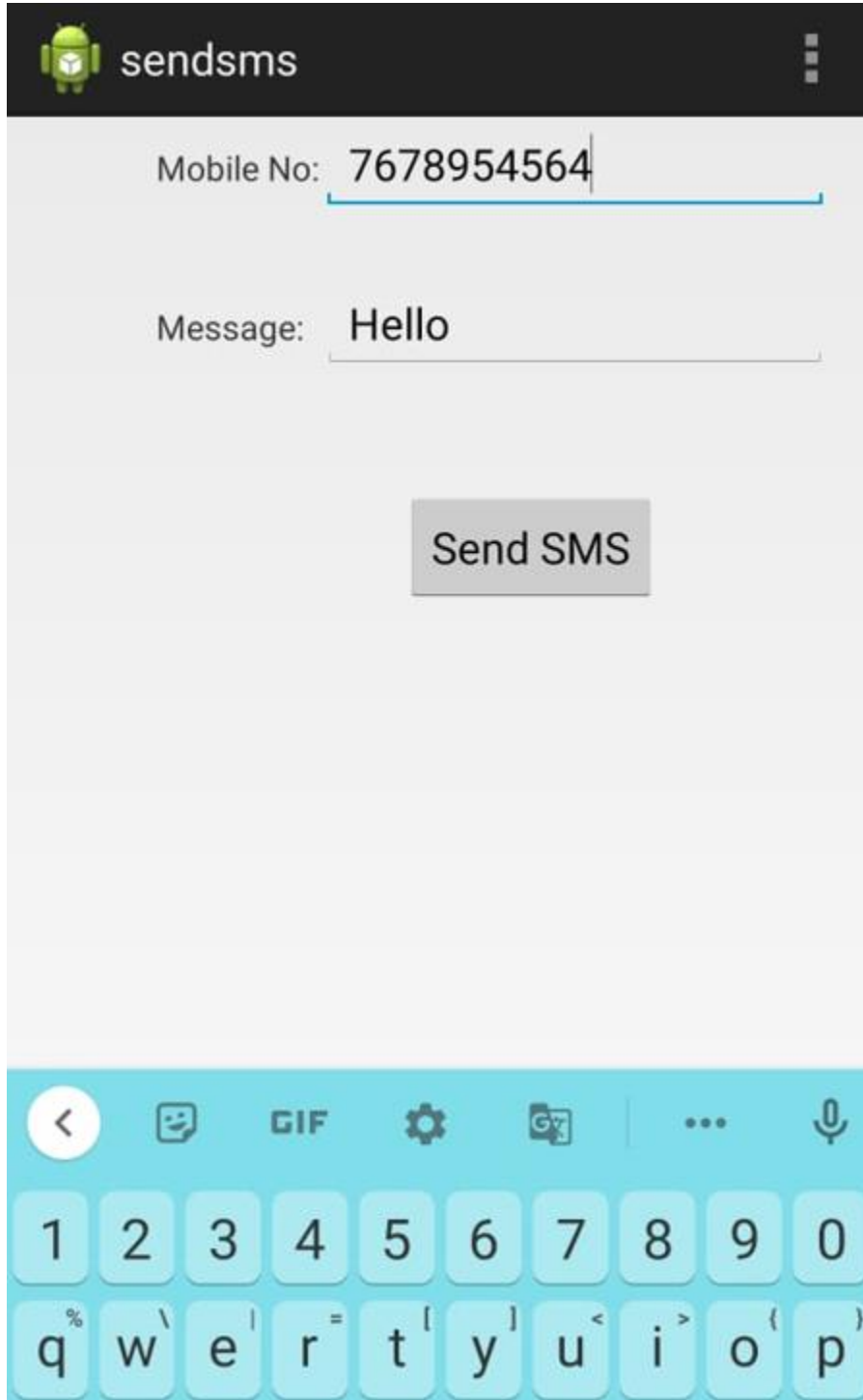
<Button

```
android:id="@+id/button1"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignLeft="@+id/editText2"  
android:layout_below="@+id/editText2"  
android:layout_marginLeft="34dp"  
android:layout_marginTop="48dp"  
android:text="Send SMS" />
```

</RelativeLayout>

5. Output:

Entered details



The screenshot shows an Android application interface for sending SMS. The app's title bar is dark grey with a green Android robot icon and the text "sendsms". Below the title bar, there are two input fields. The first field is labeled "Mobile No:" and contains the number "7678954564". The second field is labeled "Message:" and contains the text "Hello". Below these fields is a grey button labeled "Send SMS". At the bottom of the screen is a light blue keyboard with various icons and keys.

Mobile No: 7678954564

Message: Hello

Send SMS



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Sent sms

A screenshot of an Android application titled 'sendsms'. The app has a dark header bar with a green Android robot icon on the left and three vertical dots on the right. Below the header, there is a light gray background. The form consists of two input fields: 'Mobile No:' with a blue underline and 'Message:' with a gray underline. Below these fields is a gray button labeled 'Send SMS'. At the bottom of the screen, there is a dark gray banner with the text 'Message Sent successfully!' in white.

Mobile No:

Message:

Send SMS

Message Sent successfully!