

# **Department of CSE**

## **SSN College of Engineering**

**Vishakan Subramanian - 18 5001 196 - Semester VII**

**10 October 2021**

---

### **UCS 1711 - Mobile Application Development Lab**

---

#### **Exercise 8: SMS Message Application using Android**

##### **Aim:**

To develop a native application that can send an SMS to other phone numbers, and to display notifications upon reception of a new message.

## Code - SMS: Main Activity:

```
1 package com.example.sms;
2
3 import static android.Manifest.permission.READ_PHONE_STATE;
4 import static android.Manifest.permission.READ_SMS;
5 import static android.Manifest.permission.SEND_SMS;
6
7 import androidx.appcompat.app.AppCompatActivity;
8 import androidx.core.app.ActivityCompat;
9
10 import android.app.PendingIntent;
11 import android.content.BroadcastReceiver;
12 import android.content.Context;
13 import android.content.Intent;
14 import android.content.IntentFilter;
15 import android.os.Bundle;
16 import android.telephony.SmsManager;
17 import android.view.View;
18 import android.widget.EditText;
19 import android.widget.ImageButton;
20 import android.widget.TextView;
21
22 public class MainActivity extends AppCompatActivity {
23     IntentFilter intentFilter;
24     TextView historyText;
25
26     private BroadcastReceiver intentReceiver = new BroadcastReceiver() {
27         @Override
28         public void onReceive(Context context, Intent intent) {
29             //To set message text upon reception of message from
30             BroadcastReceiver
31             historyText.setText(
32                 historyText.getText() + intent.getExtras().getString("
33                 message"));
34         }
35     };
36
37     @Override
38     protected void onCreate(Bundle savedInstanceState) {
39         super.onCreate(savedInstanceState);
40         setContentView(R.layout.activity_main);
41
42         //Request user permissions for SMS
43         ActivityCompat.requestPermissions(MainActivity.this,
44             new String[]{READ_SMS, SEND_SMS, READ_PHONE_STATE}, 1);
45
46         historyText = findViewById(R.id.historyText);
47     }
48 }
```

```

46     final EditText phoneText = findViewById(R.id.phoneText);
47     final EditText messageText = findViewById(R.id.messageText);
48     final ImageButton sendButton = findViewById(R.id.sendButton);
49
50     intentFilter = new IntentFilter();
51     intentFilter.addAction("SMS_RECEIVED_ACTION");
52
53     sendButton.setOnClickListener(new View.OnClickListener() {
54         @Override
55         public void onClick(View v) {
56             String phone = phoneText.getText().toString();
57             String message = messageText.getText().toString();
58
59             sendMessage(phone, message);
60         }
61     });
62 }
63
64 void sendMessage(String phone, String message){
65     //To send a message using SmsManager
66     final String SENT = "Message Sent";
67     final String DELIVERED = "Message Delivered";
68
69     PendingIntent sentIntent = PendingIntent.getBroadcast(this, 0, new
70     Intent(SENT), 0);
71     PendingIntent deliveredIntent = PendingIntent.getBroadcast(this,
72     0, new Intent(DELIVERED), 0);
73     SmsManager smsManager = SmsManager.getDefault();
74     smsManager.sendTextMessage(phone, null, message, sentIntent,
75     deliveredIntent);
76 }
77
78 @Override
79 protected void onPause(){
80     unregisterReceiver(intentReceiver);
81     super.onPause();
82 }
83
84 @Override
85 protected void onResume(){
86     registerReceiver(intentReceiver, intentFilter);
87     super.onResume();
88 }
89 }

```

## Code - SMS: Message Receiver:

```
1 package com.example.sms;
2
3 import android.app.NotificationChannel;
4 import android.app.NotificationManager;
5 import android.app.PendingIntent;
6 import android.content.BroadcastReceiver;
7 import android.content.Context;
8 import android.content.Intent;
9 import android.os.Build;
10 import android.os.Bundle;
11 import android.telephony.SmsMessage;
12 import android.util.Log;
13 import android.widget.Toast;
14
15 import androidx.core.app.NotificationCompat;
16
17 import java.util.Date;
18
19 public class MessageReceiver extends BroadcastReceiver {
20     @Override
21     public void onReceive(Context context, Intent intent) {
22         Bundle bundle = intent.getExtras();
23         SmsMessage[] messages;
24         String msgThread = "";
25
26         if(bundle != null){
27             //Received messages as raw Protocol Data Units (PDUs)
28             Object[] pdus = (Object[]) bundle.get("pdus");
29             messages = new SmsMessage[pdus != null ? pdus.length : 0];
30
31             //Get all SMS Messages and store it in the thread
32             for(int i = 0; i < messages.length; i++){
33                 messages[i] = SmsMessage.createFromPdu((byte[]) (pdus !=
34 null ? pdus[i] : null));
35                 msgThread += messages[i].getOriginatingAddress();
36                 msgThread += ": ";
37                 msgThread += messages[i].getMessageBody();
38                 msgThread += "\n";
39             }
40
41             Toast.makeText(context, msgThread, Toast.LENGTH_SHORT).show();
42
43             //Notification for received message
44             //Prefer NotificationCompat over Notification
45             NotificationCompat.Builder msgBuilder =
46                 new NotificationCompat.Builder(context, "
47 notify_channel")
```

```

46         .setSmallIcon(R.mipmap.ic_launcher_round)
47         .setContentTitle("You have received a message.
")
48         .setContentText(msgThread)
49         .setAutoCancel(true)
50         .setPriority(NotificationCompat.
PRIORITY_DEFAULT);
51
52         //Set the notification through a MainActivity intent
53         Intent notificationIntent = new Intent(context, MainActivity.
class);
54         notificationIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
55         PendingIntent contentIntent = PendingIntent.getActivity(
context, 0, notificationIntent, 0);
56         msgBuilder.setContentIntent(contentIntent);
57
58         //Add a notification through the notify_channel channel
59         NotificationManager manager = (NotificationManager)
60             context.getSystemService(Context.NOTIFICATION_SERVICE)
;
61
62         if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.O){
63             NotificationChannel channel = new NotificationChannel("
notify_channel",
64                 "Notification Channel", NotificationManager.
IMPORTANCE_HIGH);
65             manager.createNotificationChannel(channel);
66             msgBuilder.setChannelId("notify_channel");
67         }
68
69         //Unique ID, NotificationCompat object (msgBuilder)
70         manager.notify((int) (new Date().getTime()/1000 % Integer.
MAX_VALUE), msgBuilder.build());
71         Log.i("SMS", "SENT NOTIFICATION!");
72
73         //Broadcast the message reception to the MainActivity to
trigger onReceive() method
74         Intent broadcastIntent = new Intent();
75         broadcastIntent.setAction("SMS_RECEIVED_ACTION");
76         broadcastIntent.putExtra("message", msgThread);
77         //put the msgThread in key "message" to correspond with
MainActivity's onReceive()
78         context.sendBroadcast(broadcastIntent);
79     }
80 }
81 }

```

## Code - SMS: Main Activity Layout:

```
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:layout_width="match_parent"
6   android:layout_height="match_parent"
7   tools:context=".MainActivity">
8
9   <TextView
10      android:id="@+id/textView2"
11      android:layout_width="wrap_content"
12      android:layout_height="wrap_content"
13      android:layout_marginTop="20dp"
14      android:fontFamily="@font/chelsea_market"
15      android:text="SMS"
16      android:textColor="?android:attr/textColorPrimary"
17      android:textSize="24sp"
18      android:textStyle="bold"
19      app:layout_constraintEnd_toEndOf="parent"
20      app:layout_constraintStart_toStartOf="parent"
21      app:layout_constraintTop_toTopOf="parent" />
22
23   <TextView
24      android:id="@+id/textView3"
25      android:layout_width="wrap_content"
26      android:layout_height="wrap_content"
27      android:layout_marginStart="10dp"
28      android:layout_marginTop="100dp"
29      android:fontFamily="@font/chelsea_market"
30      android:text="Enter Number:"
31      android:textColor="?android:attr/textColorPrimary"
32      android:textSize="16sp"
33      android:textStyle="bold"
34      app:layout_constraintStart_toStartOf="parent"
35      app:layout_constraintTop_toBottomOf="@+id/textView2" />
36
37   <ImageButton
38      android:id="@+id/sendButton"
39      android:layout_width="75dp"
40      android:layout_height="50dp"
41      android:layout_marginBottom="20dp"
42      android:backgroundTint="@color/teal_200"
43      app:layout_constraintBottom_toBottomOf="parent"
44      app:layout_constraintEnd_toEndOf="parent"
45      app:layout_constraintStart_toEndOf="@+id/messageText"
46      app:srcCompat="@android:drawable/ic_menu_send" />
```

```

47
48 <EditText
49     android:id="@+id/phoneText "
50     android:layout_width="250dp"
51     android:layout_height="wrap_content"
52     android:layout_marginTop="80dp"
53     android:ems="10"
54     android:fontFamily="@font/chelsea_market "
55     android:hint="+15555215554"
56     android:inputType="number"
57     android:textColor="?android:attr/textColorPrimary"
58     app:layout_constraintEnd_toEndOf="parent "
59     app:layout_constraintStart_toEndOf="@+id/textView3"
60     app:layout_constraintTop_toBottomOf="@+id/textView2" />
61
62 <EditText
63     android:id="@+id/messageText "
64     android:layout_width="300dp"
65     android:layout_height="50dp"
66     android:layout_marginStart="10dp"
67     android:layout_marginBottom="20dp"
68     android:editable="true"
69     android:ems="10"
70     android:fontFamily="@font/chelsea_market "
71     android:hint="Message"
72     android:inputType="textPersonName"
73     android:textColor="?android:attr/textColorPrimary"
74     app:layout_constraintBottom_toBottomOf="parent"
75     app:layout_constraintStart_toStartOf="parent" />
76
77 <ScrollView
78     android:id="@+id/scrollView"
79     android:layout_width="fill_parent"
80     android:layout_height="347dp"
81     android:layout_marginStart="10dp"
82     android:layout_marginEnd="10dp"
83     app:layout_constraintBottom_toTopOf="@+id/messageText"
84     app:layout_constraintEnd_toEndOf="parent"
85     app:layout_constraintStart_toStartOf="parent"
86     app:layout_constraintTop_toBottomOf="@+id/phoneText">
87
88     <LinearLayout
89         android:layout_width="match_parent"
90         android:layout_height="wrap_content"
91         android:orientation="vertical" >
92
93         <TextView
94             android:id="@+id/historyText"
95             android:layout_width="match_parent"
96             android:layout_height="wrap_content"
97             android:fontFamily="@font/chelsea_market"

```

```
98         android:textColor="?android:attr/textColorPrimary"
99         android:textSize="16sp" />
100     </LinearLayout>
101 </ScrollView>
102
103 </androidx.constraintlayout.widget.ConstraintLayout>
```

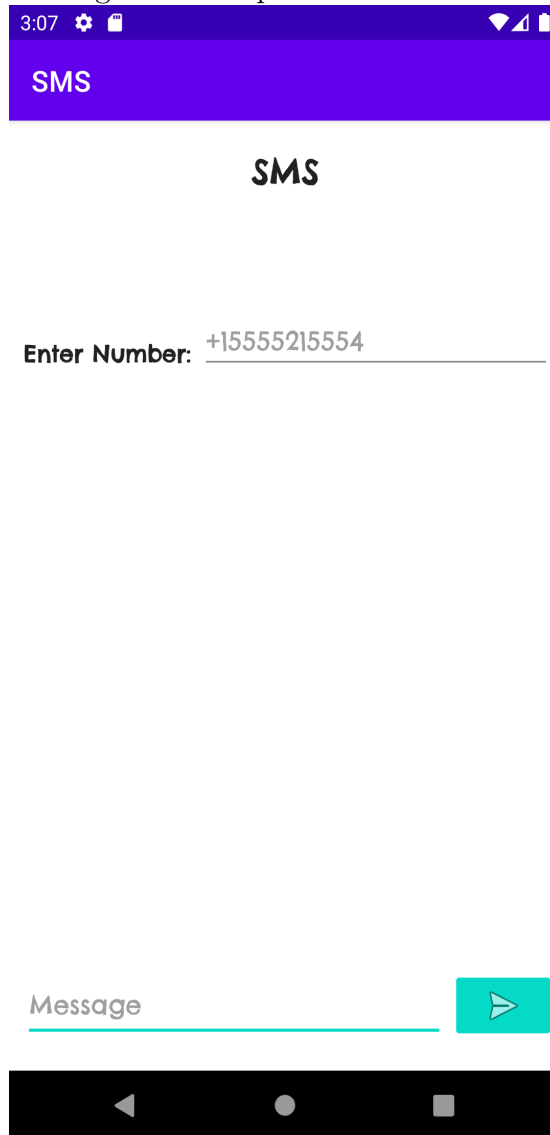


## Code - SMS: Android Manifest XML:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.sms">
4
5     <!--Permissions for allowing SMS functionality-->
6     <uses-permission android:name="android.permission.READ_SMS"/>
7     <uses-permission android:name="android.permission.RECEIVE_SMS"/>
8     <uses-permission android:name="android.permission.SEND_SMS"/>
9     <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
10
11     <application
12         android:allowBackup="true"
13         android:icon="@mipmap/ic_launcher"
14         android:label="@string/app_name"
15         android:roundIcon="@mipmap/ic_launcher_round"
16         android:supportsRtl="true"
17         android:theme="@style/Theme.SMS">
18         <activity
19             android:name=".MainActivity"
20             android:exported="true">
21             <intent-filter>
22                 <action android:name="android.intent.action.MAIN" />
23
24                 <category android:name="android.intent.category.LAUNCHER"
25             />
26             </intent-filter>
27         </activity>
28
29         <!--Message Receiver-->
30         <receiver android:name=".MessageReceiver">
31             <intent-filter>
32                 <action android:name="android.provider.Telephony.
33 SMS_RECEIVED"/>
34             </intent-filter>
35         </receiver>
36
37         <meta-data
38             android:name="preloaded_fonts"
39             android:resource="@array/preloaded_fonts" />
40     </application>
41 </manifest>
```

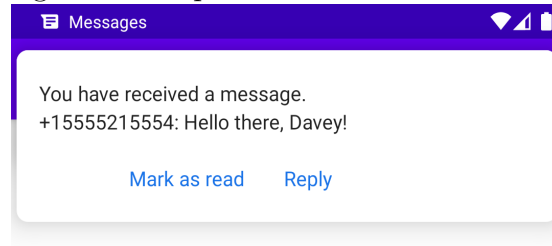
## Output: Main Screen:

Figure 1: Output: Main Screen.



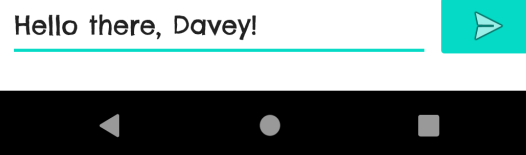
## Output: SMS Sent & Received:

Figure 2: Output: SMS Sent & Received.



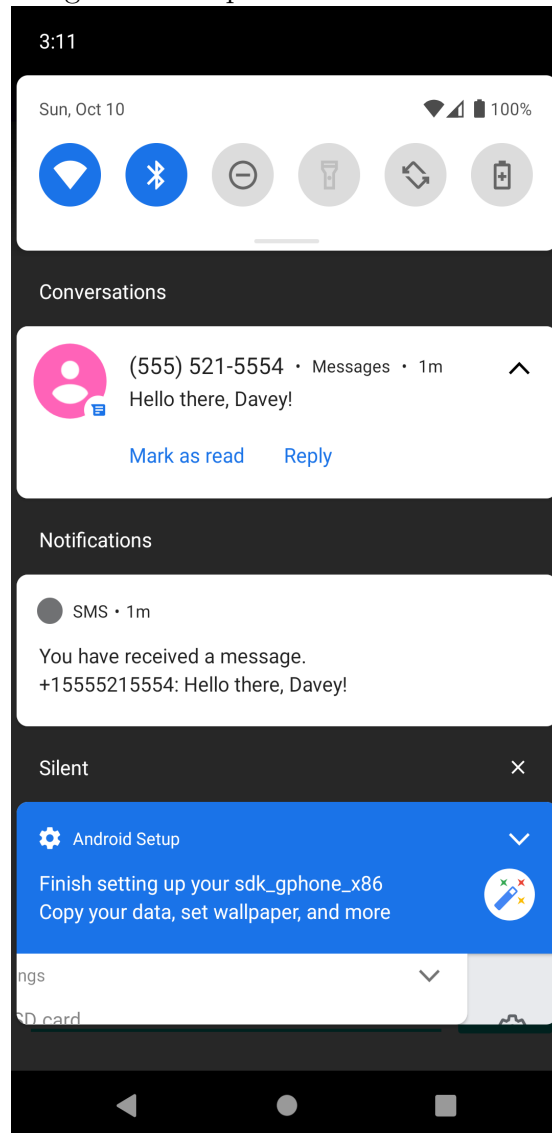
Enter Number: 5554

+15555215554: Hello there, Davey!



## Output: SMS Notification:

Figure 3: Output: SMS Notification.



## Learning Outcome:

- I learnt to set permissions like **READ\_SMS**, **SEND\_SMS** etc. to enable SMS telephony services for the application.
- I implemented methods to prompt the user to allow the SMS permissions to be given for the application.
- I implemented a **BroadcastReceiver** class to take care of message reception handling.
- I understood how to set a **ScrollView** and how to work with **LinearLayout**.
- I learnt how to send a message using the **SmsManager** class.
- I understood how to work with BroadcastReceiver and Intents.
- I learnt how to parse the received SMS messages using **Bundle** object as **PDUs** (Protocol Data Units)
- I implemented a notification alert to the user upon reception of a message using the **NotificationCompat.Builder** and **NotificationManager** classes.
- I learnt how to set the notification to a channel and how to notify using the manager using the **notify()** method.
- I learnt how to send the received message from the MessageReceiver handler to the MainActivity using BroadcastReceiver and Intent, using **sendBroadcast()** method.