SSN COLLEGE OF ENGINEERING, KALAVAKKAM

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

UCS1711 - MOBILE APPLICATION DEVELOPMENT LAB

Assignment 8

Name: Jayannthan P T *Dept: CSE 'A'* *Roll No.: 205001049*

SMS Sending and Notification

Ex. No:8

Title of the Program: Develop an android app that sends SMS and creates an alert upon receiving the SMS with text in the notification.

Objective:

The objective of the SMSNotifier Android App project is to create an application that sends SMS messages and generates a notification upon receiving an SMS with text. The app provides a user interface for entering a contact number and a message. It sends the SMS and displays incoming messages in a scrollable view. Additionally, it triggers a notification with the received message content.

Algorithm:

1. Create an Android app with MainActivity and MessageReceiver (BroadcastReceiver) classes.
2. MainActivity includes EditTexts for contact number and message, along with a button to send the SMS.
3. Implement a method (sendMsg) to send an SMS to the specified contact number.
4. Register a BroadcastReceiver (MessageReceiver) to capture incoming SMS messages and generate a notification.
5. Create a method (makeNotification) to build and display a notification with the received SMS content.
6. Request notification permissions if the Android version is Tiramisu or later.
7. Utilize PendingIntent to handle notifications and trigger an action when the notification is clicked.
8. Display incoming SMS messages in a LinearLayout within a ScrollView for scrollable view.
9. Toast message for displaying incoming SMS details.

Features used:

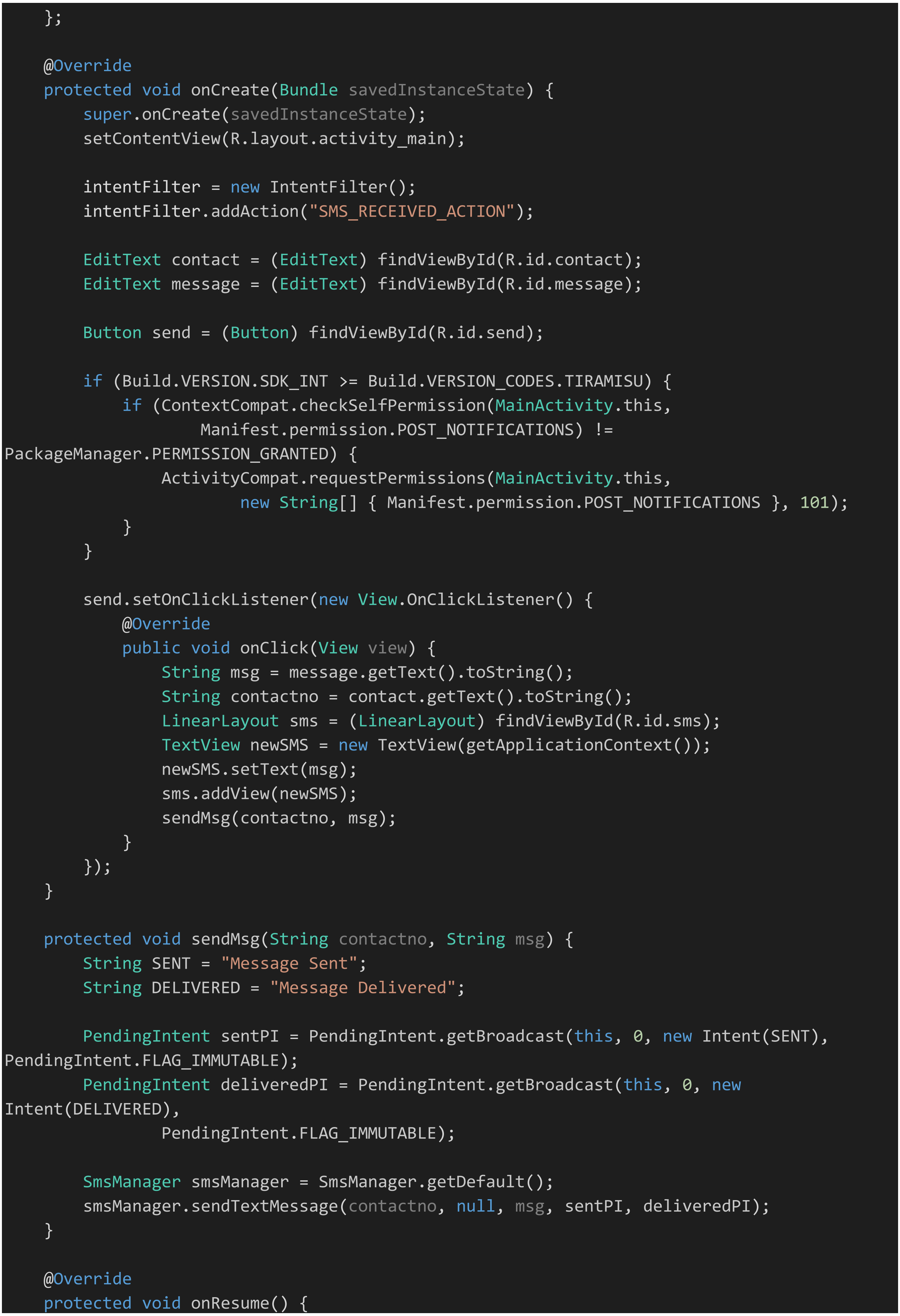
1. SMSManager to send SMS messages.

2. BroadcastReceiver for capturing incoming SMS messages.
3. NotificationCompat.Builder for building notifications.
4. PendingIntent for handling notifications and actions.
5. ScrollView and LinearLayout for displaying incoming SMS messages.
6. EditText, Button, and other UI elements for user interaction.
7. Toast messages for user feedback.

Source code:

- `MainActivity.java`

```java
package com.example.exercise8;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.app.NotificationCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.content.pm.PackageManager;
import android.graphics.Color;
import android.os.Build;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.Gravity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    IntentFilter intentFilter;

    private BroadcastReceiver intentReceiver = new BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {
            LinearLayout sms = (LinearLayout) findViewById(R.id.sms);
            TextView newSMS = new TextView(getApplicationContext());
            String msg = intent.getExtras().getString("message");
            newSMS.setText(msg);
            newSMS.setGravity(Gravity.RIGHT);
            sms.addView(newSMS);
            makeNotification(msg);
        }
```



```
    };

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

        intentFilter = new IntentFilter();
        intentFilter.addAction("SMS_RECEIVED_ACTION");

        EditText contact = (EditText) findViewById(R.id.contact);
        EditText message = (EditText) findViewById(R.id.message);

        Button send = (Button) findViewById(R.id.send);

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.TIRAMISU) {
            if (ContextCompat.checkSelfPermission(MainActivity.this,
                    Manifest.permission.POST_NOTIFICATIONS) !=
PackageManager.PERMISSION_GRANTED) {
                ActivityCompat.requestPermissions(MainActivity.this,
                        new String[] { Manifest.permission.POST_NOTIFICATIONS }, 101);
            }
        }

        send.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String msg = message.getText().toString();
                String contactno = contact.getText().toString();
                LinearLayout sms = (LinearLayout) findViewById(R.id.sms);
                TextView newSMS = new TextView(getApplicationContext());
                newSMS.setText(msg);
                sms.addView(newSMS);
                sendMsg(contactno, msg);
            }
        });
    }

    protected void sendMsg(String contactno, String msg) {
        String SENT = "Message Sent";
        String DELIVERED = "Message Delivered";

        PendingIntent sentPI = PendingIntent.getBroadcast(this, 0, new Intent(SENT),
PendingIntent.FLAG_IMMUTABLE);
        PendingIntent deliveredPI = PendingIntent.getBroadcast(this, 0, new
Intent(DELIVERED),
                PendingIntent.FLAG_IMMUTABLE);

        SmsManager smsManager = SmsManager.getDefault();
        smsManager.sendTextMessage(contactno, null, msg, sentPI, deliveredPI);
    }

    @Override
    protected void onResume() {
        registerReceiver(intentReceiver, intentFilter);
        super.onResume();
    }

    @Override
    protected void onPause() {
        unregisterReceiver(intentReceiver);
        super.onPause();
    }

    public void makeNotification(String msg) {
        String channelID = "CHANNEL_ID_NOTIFICATION";

        Intent activityIntent = new Intent(this, MainActivity.class);
        PendingIntent contentIntent = PendingIntent.getActivity(this, 0, activityIntent,
PendingIntent.FLAG_IMMUTABLE);

        NotificationCompat.Builder builder = new
NotificationCompat.Builder(getApplicationContext(), channelID);
        builder.setSmallIcon(R.drawable.ic_notifications)
                .setContentTitle("Notification")
                .setContentText(msg)
                .setAutoCancel(true)
                .setPriority(NotificationCompat.PRIORITY_DEFAULT)
                .setContentIntent(contentIntent);

        NotificationManager notificationManager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            NotificationChannel notificationChannel =
notificationManager.getNotificationChannel(channelID);
            if (notificationChannel == null) {
                int importance = NotificationManager.IMPORTANCE_HIGH;
                notificationChannel = new NotificationChannel(channelID, "Notification",
importance);
                notificationChannel.setLightColor(Color.GREEN);
                notificationChannel.enableVibration(true);
                notificationManager.createNotificationChannel(notificationChannel);
            }
        }
        notificationManager.notify(0, builder.build());
    }
}
```

- activity_main.xml

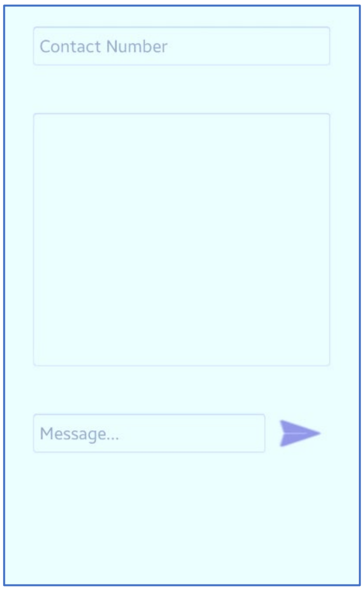
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
android:layout_height="match_parent" android:orientation="vertical"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

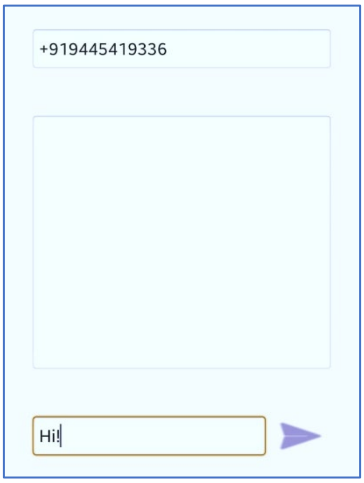
```xml
    <EditText android:id="@+id/contact" android:layout_width="350dp"
android:layout_height="50dp" android:layout_marginTop="100dp"
android:layout_gravity="center_horizontal" android:inputType="phone"
android:textSize="20dp" android:hint="Contact Number" android:singleLine="true"
android:background="@android:drawable/editbox_background" />

    <ScrollView android:layout_width="350dp" android:layout_height="300dp"
android:layout_gravity="center_horizontal" android:gravity="center_vertical"
android:layout_marginTop="50dp" android:background="@android:drawable/editbox_background">

        <LinearLayout android:id="@+id/sms" android:layout_width="match_parent"
android:layout_height="wrap_content" android:orientation="vertical">

        </LinearLayout>

    </ScrollView>

    <LinearLayout android:layout_width="350dp" android:layout_height="50dp"
android:layout_marginTop="50dp" android:orientation="horizontal"
android:layout_gravity="center_horizontal">

        <EditText android:id="@+id/message" android:layout_width="275dp"
android:layout_height="50dp" android:layout_gravity="left"
android:layout_marginBottom="50dp"
android:background="@android:drawable/editbox_background" android:text=""
android:hint="Message..." android:inputType="text" android:textSize="20dp" />

        <Button android:id="@+id/send" android:layout_width="75dp"
android:layout_height="50dp" android:background="@android:drawable/ic_menu_send"/>

    </LinearLayout>

</LinearLayout>
```

- ReceiveActivity.java

```java
    package com.example.exercise8;

    import android.content.BroadcastReceiver;
    import android.content.Context;
    import android.content.Intent;
    import android.os.Bundle;
    import android.telephony.SmsMessage;
    import android.widget.Toast;

    public class MessageReceiver extends BroadcastReceiver {
        @Override
        public void onReceive(Context context, Intent intent) {
            Bundle bundle = intent.getExtras();
            SmsMessage[] messages;
            String str = "";

            if (bundle != null) {
                Object[] pdus = (Object[]) bundle.get("pdus");
                messages = new SmsMessage[pdus != null ? pdus.length : 0];
                for (int i = 0; i < messages.length; i++) {
                    messages[i] = SmsMessage.createFromPdu((byte[]) (pdus != null ?
pdus[i] : null));
                    str += messages[i].getOriginatingAddress();
                    str += "\n";
                    str += messages[i].getMessageBody();
                    str += "\n";
                }

                Toast.makeText(context, str, Toast.LENGTH_SHORT).show();

                Intent broadcastIntent = new Intent();
                broadcastIntent.setAction("SMS_RECEIVED_ACTION");
                broadcastIntent.putExtra("message", str);
                context.sendBroadcast(broadcastIntent);
            }
        }
    }
```

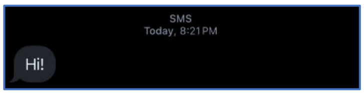
Output:





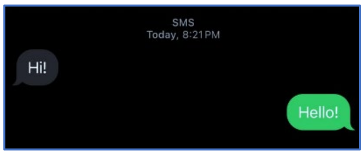
+919445419336

Hi!



SMS
Today, 8:21 PM

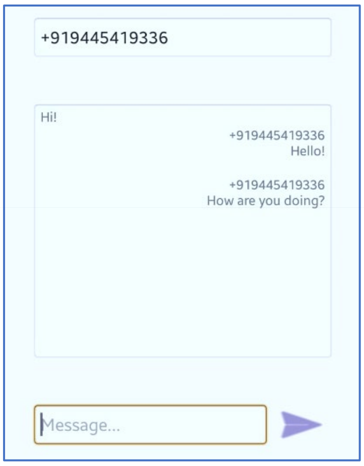
Hi!



SMS
Today, 8:21 PM

Hi!

Hello!



+919445419336

Hi!

+919445419336
Hello!

+919445419336
How are you doing?

Result:

The mobile application was completed successfully

Best Practices:

1. Use **PendingIntent.FLAG_IMMUTABLE** for creating immutable pending intents.
2. Check for permissions and request them if necessary.
3. Register and unregister BroadcastReceiver in **onResume** and **onPause** to manage resources efficiently.
4. Create a notification channel for compatibility with Android Oreo (API level 26) and above.
5. Utilize separate classes for different functionalities to maintain a clean code structure.
6. Display meaningful toast messages for better user understanding.
7. Handle exceptions related to SMS sending and notification creation for robustness.

Learning Outcomes:

1. Understanding the usage of SMSManager to send SMS messages programmatically.
2. Implementing a BroadcastReceiver to capture incoming SMS messages.
3. Building and displaying notifications using NotificationCompat.Builder.
4. Handling user input through EditText and triggering actions with buttons.
5. Utilizing PendingIntent for managing notification actions.
6. Creating a scrollable view for displaying a list of incoming SMS messages.
7. Requesting and handling runtime permissions in Android.
8. Implementing best practices for efficient resource management and error handling.