Pimpri Chinchwad Education Trust's
Pimpri Chinchwad College of Engineering

Assignment-05

Roll No: 123M1H048
Name of Student: PRATIK RATHOD
Submission Date: 17 / 10 / 24

---

1. Develop an Android application that allows the user to send and receive SMS messages. The app should have an input field for the phone number and message content. Upon clicking the Send button, the message should be sent to the specified phone number using the SMS Manager API. Additionally, implement a broadcast receiver to listen for incoming SMS messages and display the message content in a TextView.

AndroidManifest.xml

```
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
```

Activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <EditText
        android:id="@+id/phone"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Phone Number" />

    <EditText
        android:id="@+id/message"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Message" />

    <Button
        android:id="@+id/send"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Send" />

    <TextView
        android:id="@+id/received"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```
        android:text="" />
</LinearLayout>
```

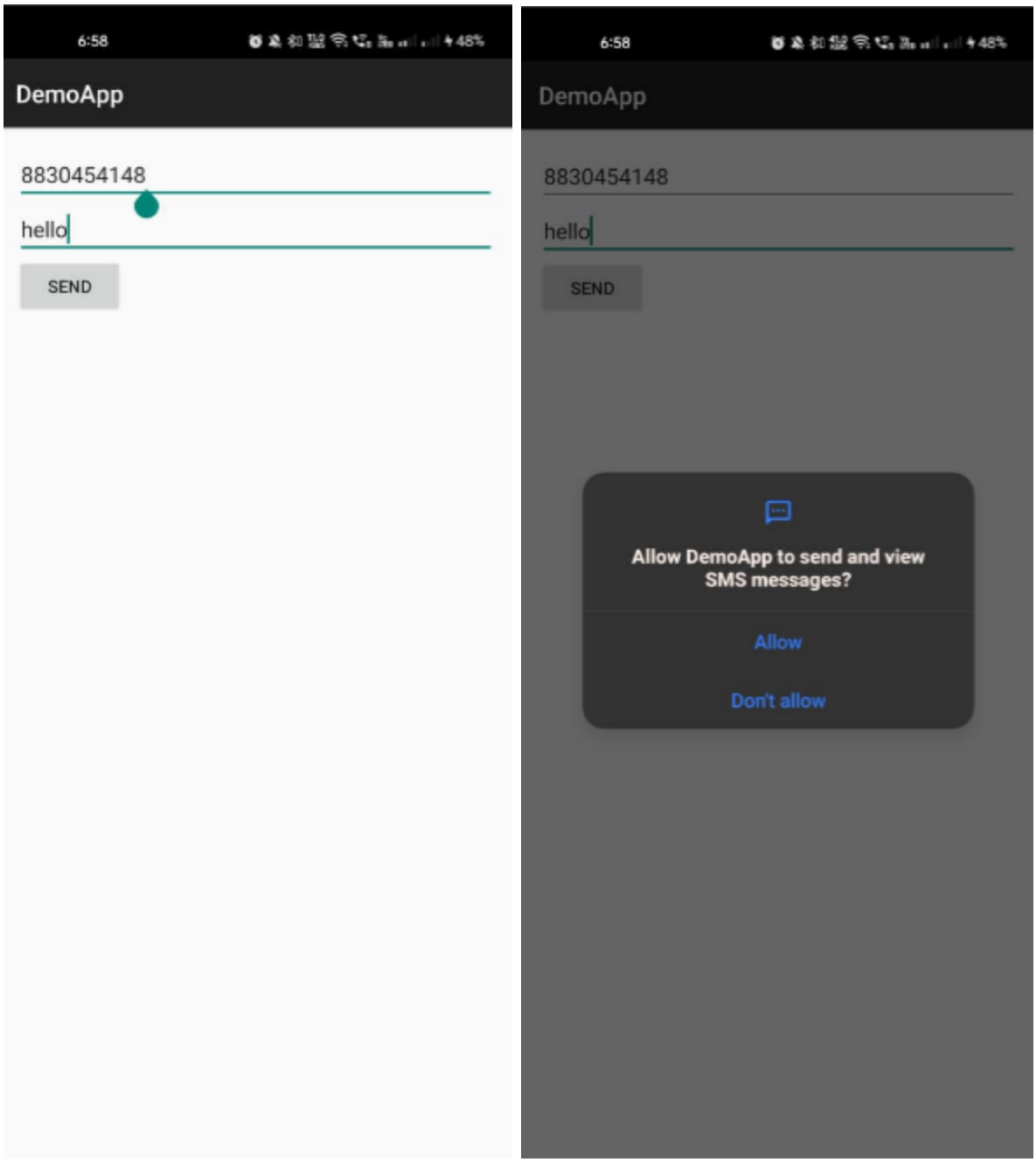
MainActivity.java

```
package com.example.forpractice;


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

public class MainActivity extends AppCompatActivity {

    private static final int MY_PERMISSIONS_REQUEST_SEND_SMS = 0;

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

        EditText phoneT = findViewById(R.id.phone);
        EditText messageT = findViewById(R.id.message);
        Button send = findViewById(R.id.send);
        TextView received = findViewById(R.id.received);

        send.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String phone = phoneT.getText().toString();
                String message = messageT.getText().toString();

                if (ContextCompat.checkSelfPermission(MainActivity.this,
                        Manifest.permission.SEND_SMS) !=
PackageManager.PERMISSION_GRANTED) {
                    ActivityCompat.requestPermissions(MainActivity.this,
                            new String[]{Manifest.permission.SEND_SMS},
MY_PERMISSIONS_REQUEST_SEND_SMS);
                } else {
                    SmsManager smsManager = SmsManager.getDefault();
                    smsManager.sendTextMessage(phone, null, message, null,
null);
                }
            }
        });
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, String[]
permissions, int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
        switch (requestCode) {
            case MY_PERMISSIONS_REQUEST_SEND_SMS: {
                if (grantResults.length > 0 && grantResults[0] ==
```

```
PackageManager.PERMISSION_GRANTED) {

                } else {

                }
                return;
            }
        }
    }
}
```

SmsReceiver.java

```
package com.example.forpractice;


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

public class SmsReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        Bundle bundle = intent.getExtras();
        SmsMessage[] msgs = null;
        String str = "";

        if (bundle != null) {
            Object[] pdus = (Object[]) bundle.get("pdus");
            msgs = new SmsMessage[pdus.length];
            for (int i = 0; i < msgs.length; i++) {
                msgs[i] = SmsMessage.createFromPdu((byte[]) pdus[i]);
                str += "SMS from " + msgs[i].getOriginatingAddress();
                str += ": " + msgs[i].getMessageBody().toString();
                str += "\n";
            }
            TextView receivedMessage = ((MainActivity)
context).findViewById(R.id.received);
            receivedMessage.setText(str);
        }
    }
}
```



2. Create an application that opens the camera interface to capture photos. Once the photo is taken, it should be displayed in an ImageView on the apps main screen. Use the Camera API or Intent with ACTION_IMAGE_CAPTURE to invoke the device's camera. Ensure proper handling of the permissions required for accessing the camera.

AndroidManifest.xml

```
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Activity_main.xml

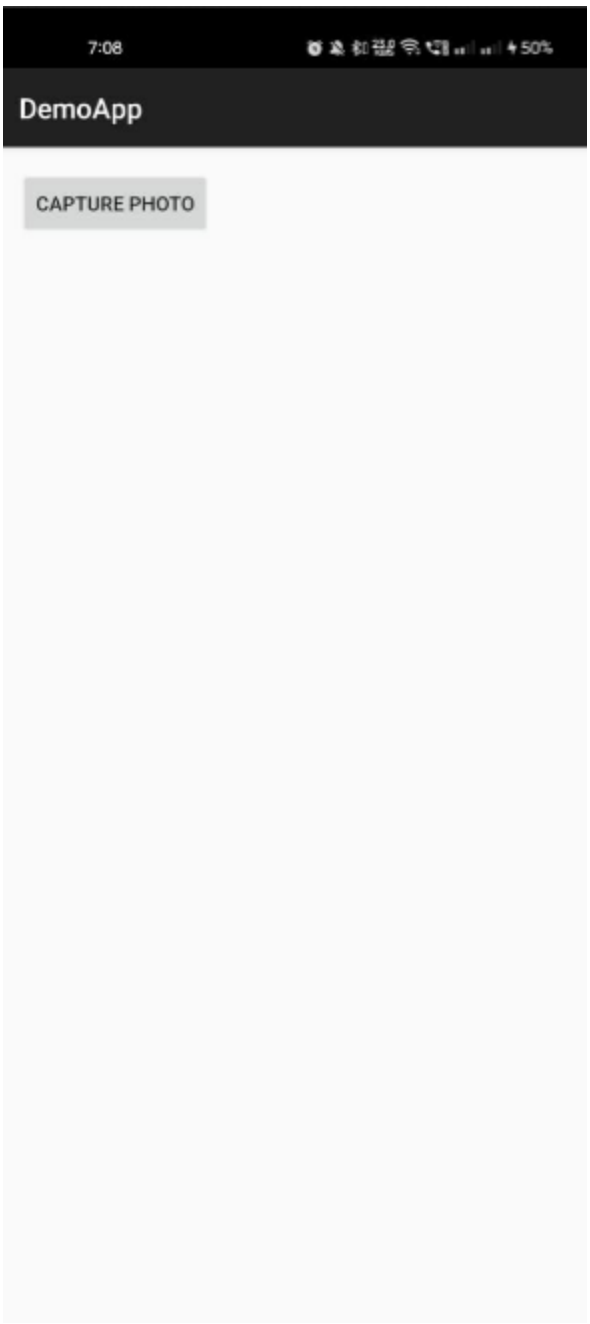
```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

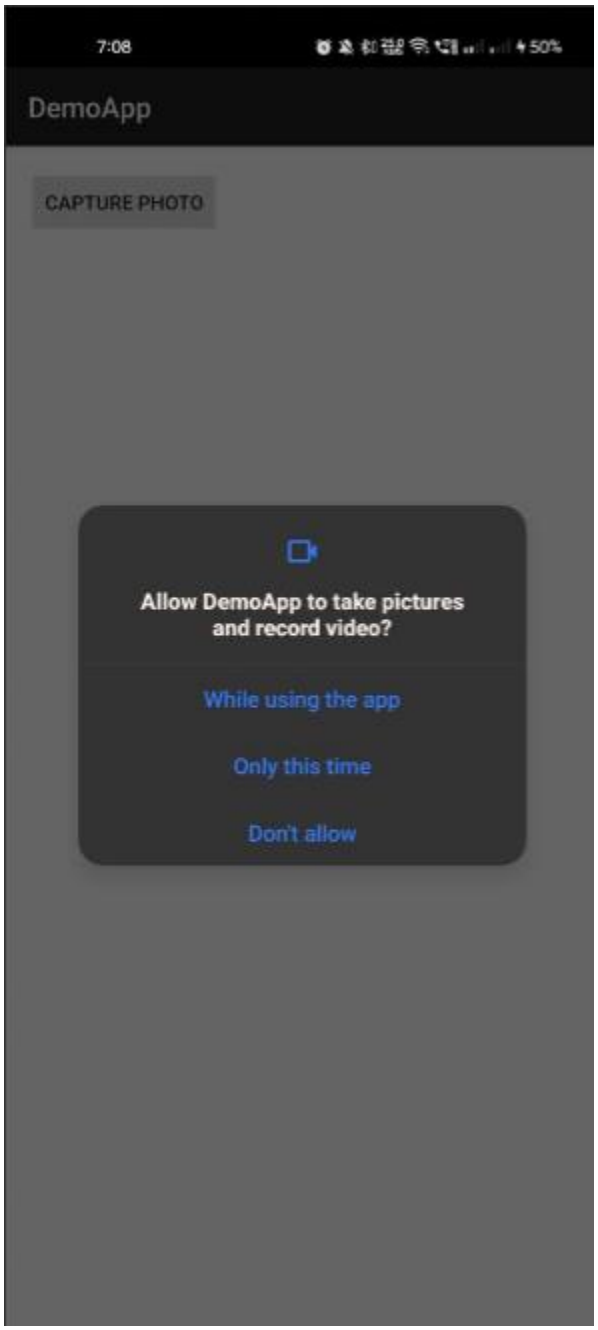
```
    android:orientation="vertical"
    android:padding="16dp">

    <Button
        android:id="@+id/btnCapture"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Capture Photo" />

    <ImageView
        android:id="@+id/imgPhoto"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:scaleType="centerCrop" />
</LinearLayout>
```

MainActivity.xml

```
package com.example.forpractice;



import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

public class MainActivity extends AppCompatActivity {

    private static final int REQUEST_IMAGE_CAPTURE = 1;
    private ImageView imgPhoto;

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

        Button btnCapture = findViewById(R.id.btnCapture);
        imgPhoto = findViewById(R.id.imgPhoto);

        btnCapture.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED) {
                    ActivityCompat.requestPermissions(MainActivity.this, new
String[]{Manifest.permission.CAMERA}, REQUEST_IMAGE_CAPTURE);
                } else {
                    dispatchTakePictureIntent();
                }
            }
        });
    }
```

```
    private void dispatchTakePictureIntent() {
        Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        if (takePictureIntent.resolveActivity(getPackageManager()) != null) {
            startActivityForResult(takePictureIntent, REQUEST_IMAGE_CAPTURE);
        }
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable
Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {
            Bundle extras = data.getExtras();
            Bitmap imageBitmap = (Bitmap) extras.get("data");
            imgPhoto.setImageBitmap(imageBitmap);
        }
    }
}
```





3. Design an Android app that allows users to initiate phone calls by entering a phone number and clicking a “Call” button. Additionally, implement functionality to listen for changes in call states (e.g., ringing, answered, idle) using the Telephony Manager API. Display the current call state in a TextView when it changes.

AndroidManifest.xml

```
<uses-permission android:name="android.permission.CALL_PHONE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
```

Activity_main.xml

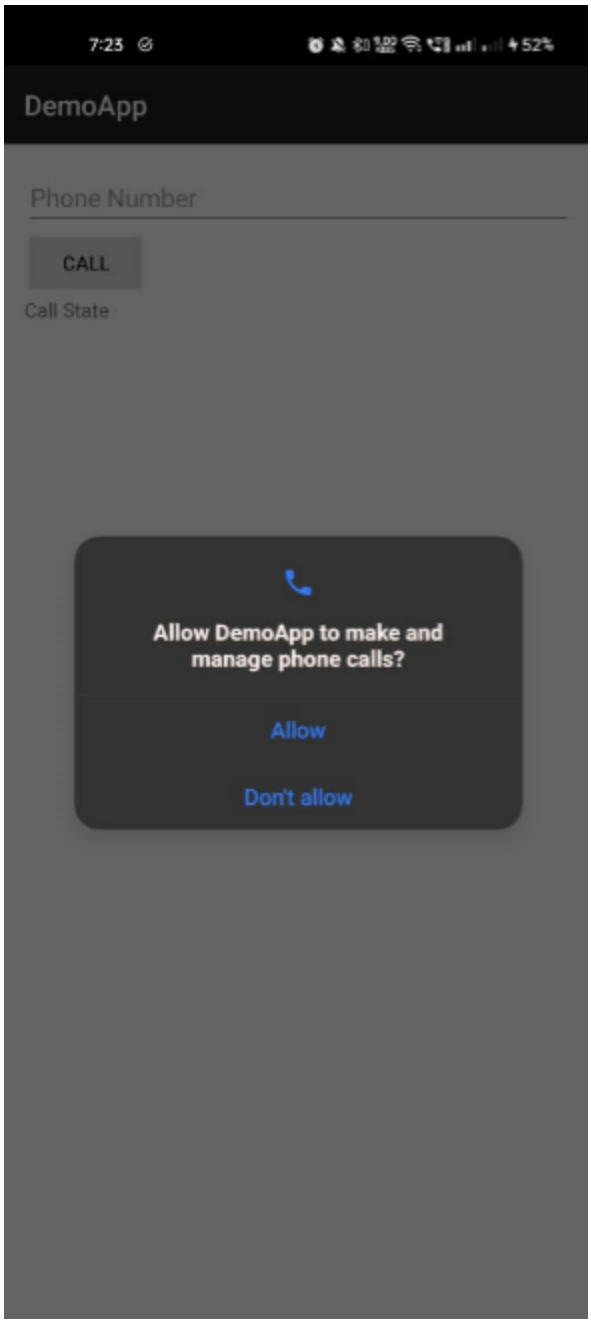
```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <EditText
        android:id="@+id/etNum"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Phone Number" />

    <Button
        android:id="@+id/btnCall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Call" />

    <TextView
        android:id="@+id/tvState"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Call State" />
</LinearLayout>
```

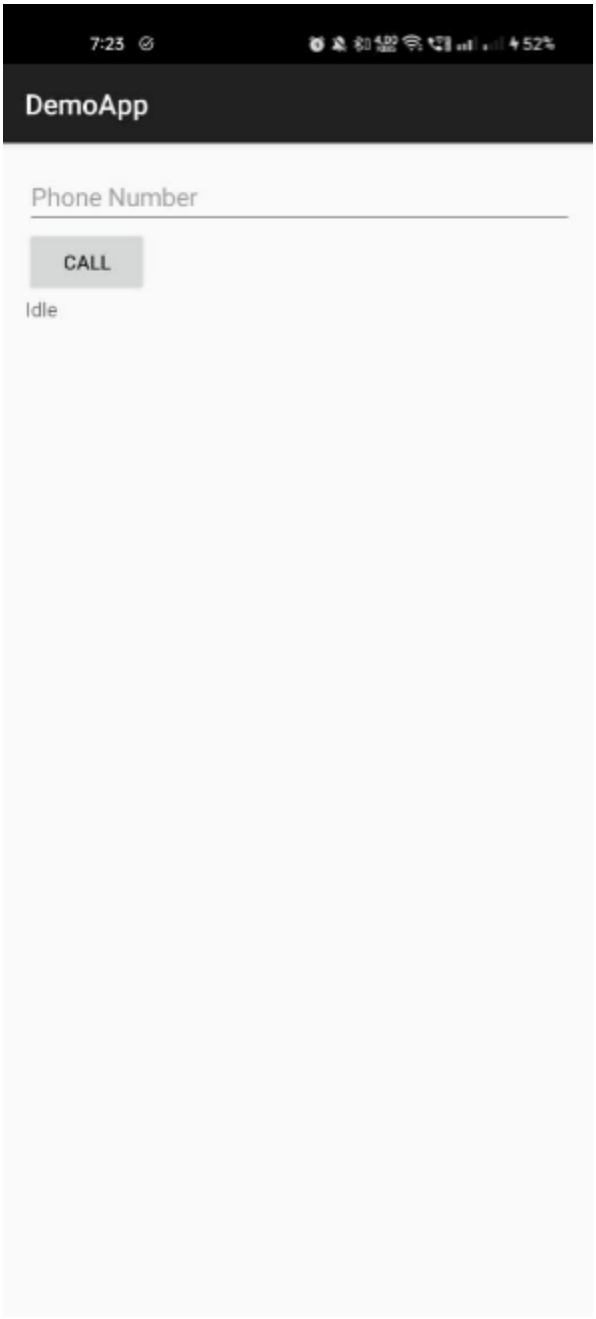
MainActivity.java

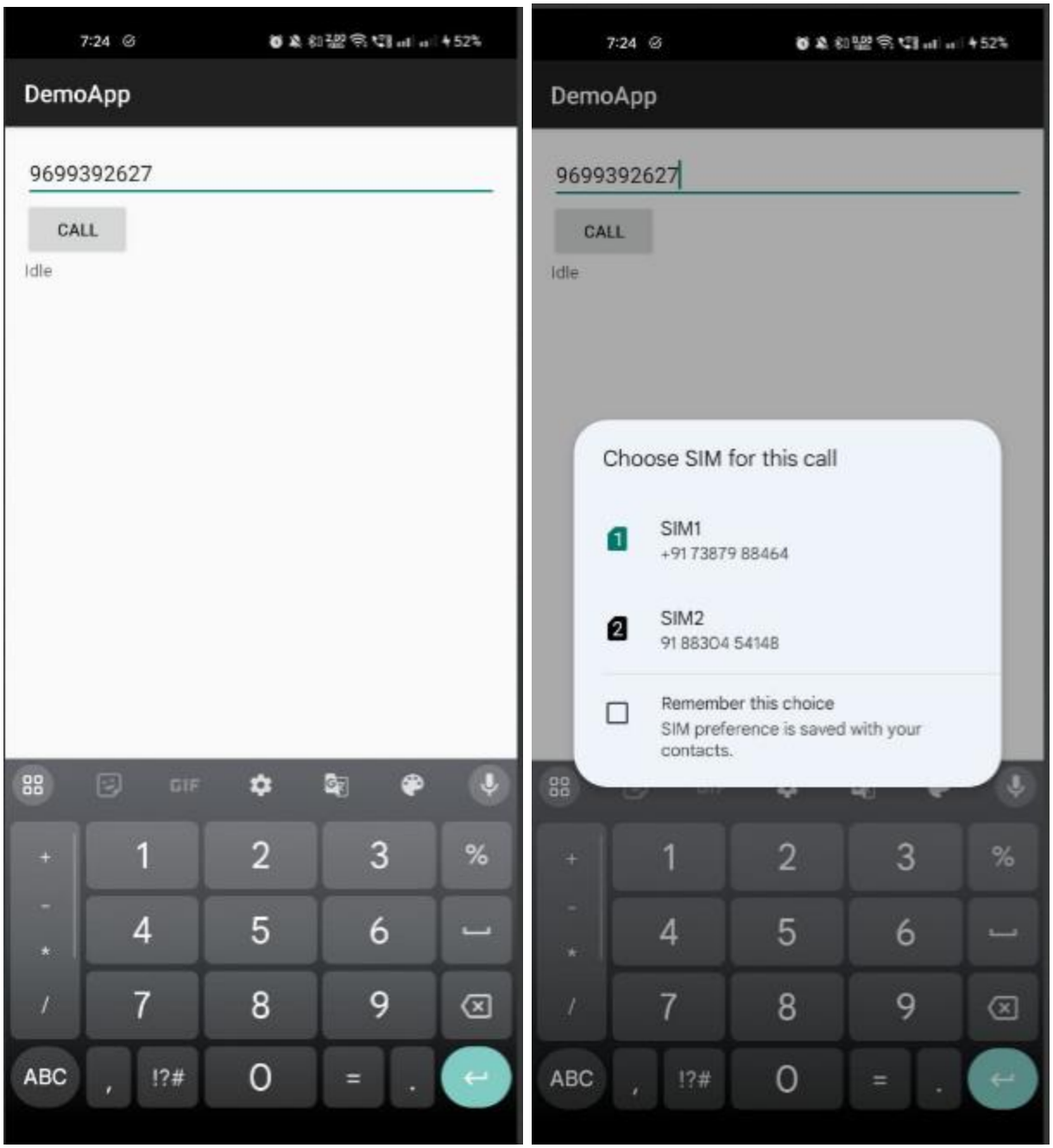
```
package com.example.forpractice;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.telephony.TelephonyCallback;
import android.telephony.TelephonyManager;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

public class MainActivity extends AppCompatActivity {
```

```
    private static final int REQUEST_CALL_PERMISSION = 1;
    private static final int REQUEST_PHONE_STATE_PERMISSION = 2;
    private EditText etNum;
    private TextView tvState;

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

        etNum = findViewById(R.id.etNum);
        Button btnCall = findViewById(R.id.btnCall);
        tvState = findViewById(R.id.tvState);

        btnCall.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.CALL_PHONE) != PackageManager.PERMISSION_GRANTED) {
                    ActivityCompat.requestPermissions(MainActivity.this, new
String[]{Manifest.permission.CALL_PHONE}, REQUEST_CALL_PERMISSION);
                } else {
                    makeCall();
                }
            }
        });

        if (ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.READ_PHONE_STATE) != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(MainActivity.this, new
String[]{Manifest.permission.READ_PHONE_STATE},
REQUEST_PHONE_STATE_PERMISSION);
        } else if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.S) {
            listenForCallStateChanges();
        }
    }

    private void makeCall() {
        String phoneNumber = etNum.getText().toString();
        Intent callIntent = new Intent(Intent.ACTION_CALL);
        callIntent.setData(Uri.parse("tel:" + phoneNumber));
        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.CALL_PHONE) == PackageManager.PERMISSION_GRANTED) {
            startActivity(callIntent);
        }
    }

    @RequiresApi(api = Build.VERSION_CODES.S)
    private void listenForCallStateChanges() {
        TelephonyManager telephonyManager = (TelephonyManager)
getSystemService(TELEPHONY_SERVICE);
        if (telephonyManager != null) {
            telephonyManager.registerTelephonyCallback(getMainExecutor(), new
CustomTelephonyCallback());
        } else {
            Log.e("MainActivity", "TelephonyManager is null");
        }
    }

    @RequiresApi(api = Build.VERSION_CODES.S)
    private class CustomTelephonyCallback extends TelephonyCallback implements
TelephonyCallback.CallStateListener {
```

```
        @Override
        public void onCallStateChanged(int state) {
            switch (state) {
                case TelephonyManager.CALL_STATE_RINGING:
                    tvState.setText("Ringing");
                    break;
                case TelephonyManager.CALL_STATE_OFFHOOK:
                    tvState.setText("Answered");
                    break;
                case TelephonyManager.CALL_STATE_IDLE:
                    tvState.setText("Idle");
                    break;
                default:
                    Log.e("CustomTelephonyCallback", "Unknown call state: " +
state);
                    break;
            }
        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
        if (requestCode == REQUEST_PHONE_STATE_PERMISSION) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.S) {
                    listenForCallStateChanges();
                }
            } else {
                Log.e("MainActivity", "Phone state permission denied");
            }
        } else if (requestCode == REQUEST_CALL_PERMISSION) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                makeCall();
            } else {
                Log.e("MainActivity", "Call permission denied");
            }
        }
    }
}
```


7:23
52%
DemoApp
Phone Number
CALL
Call State
Allow DemoApp to make and manage phone calls?
Allow
Don't allow


7:23
52%
DemoApp
Phone Number
CALL
Idle



4. Create a voice command application that uses the Speech API to recognize spoken words and convert them into text. The app should have a “Start Listening” button that initiates speech recognition, and the recognized text should be displayed in a TextView. Provide functionality for handling errors or when speech input is not detected.

AndroidManifest.xml

```
<uses-permission android:name="android.permission.RECORD_AUDIO" />
```

Activity_main.xml

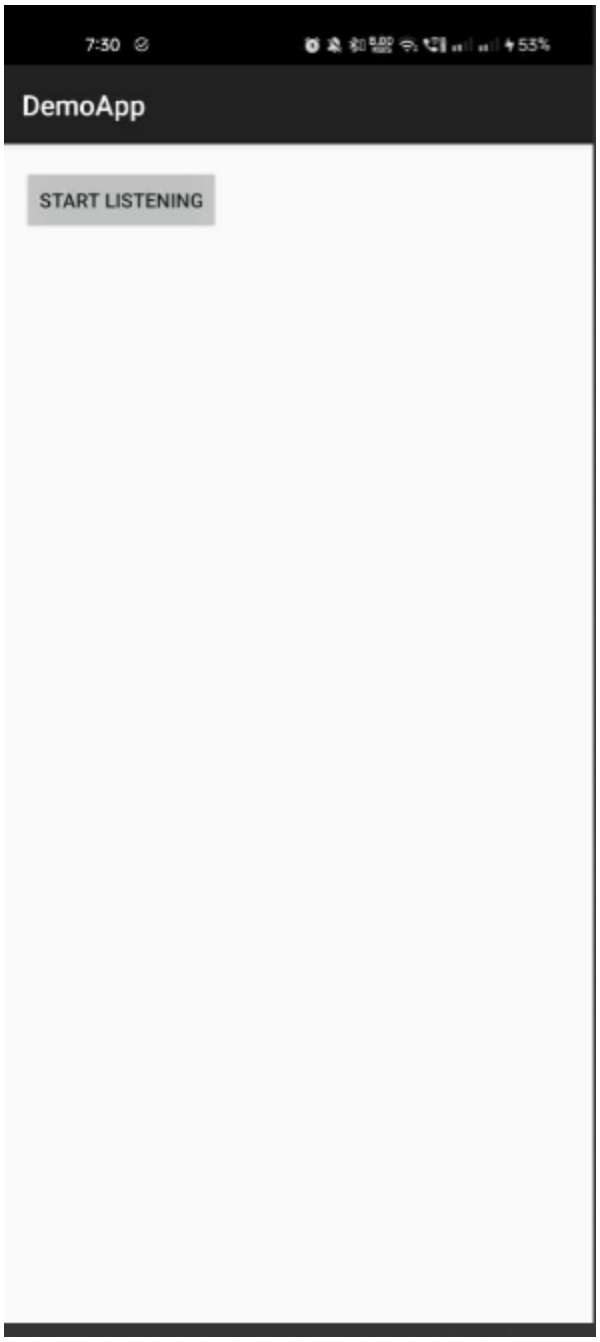
```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">
```

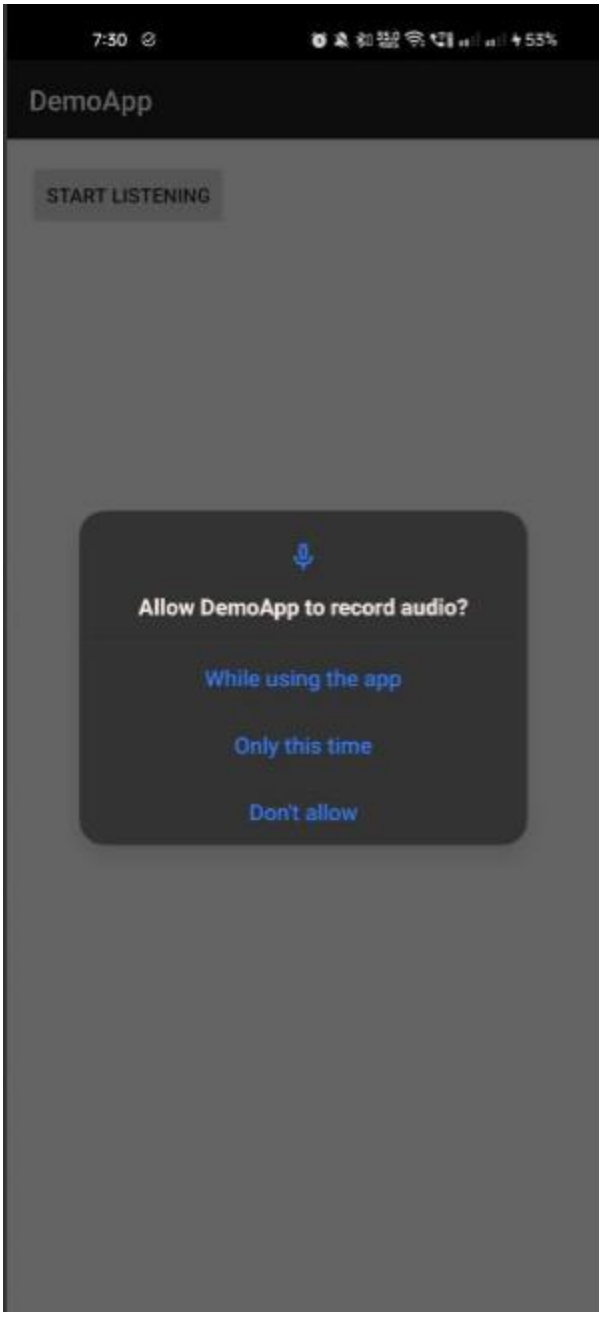
```
    <Button
        android:id="@+id/listen"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Start Listening" />

    <TextView
        android:id="@+id/result"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="" />
</LinearLayout>
```

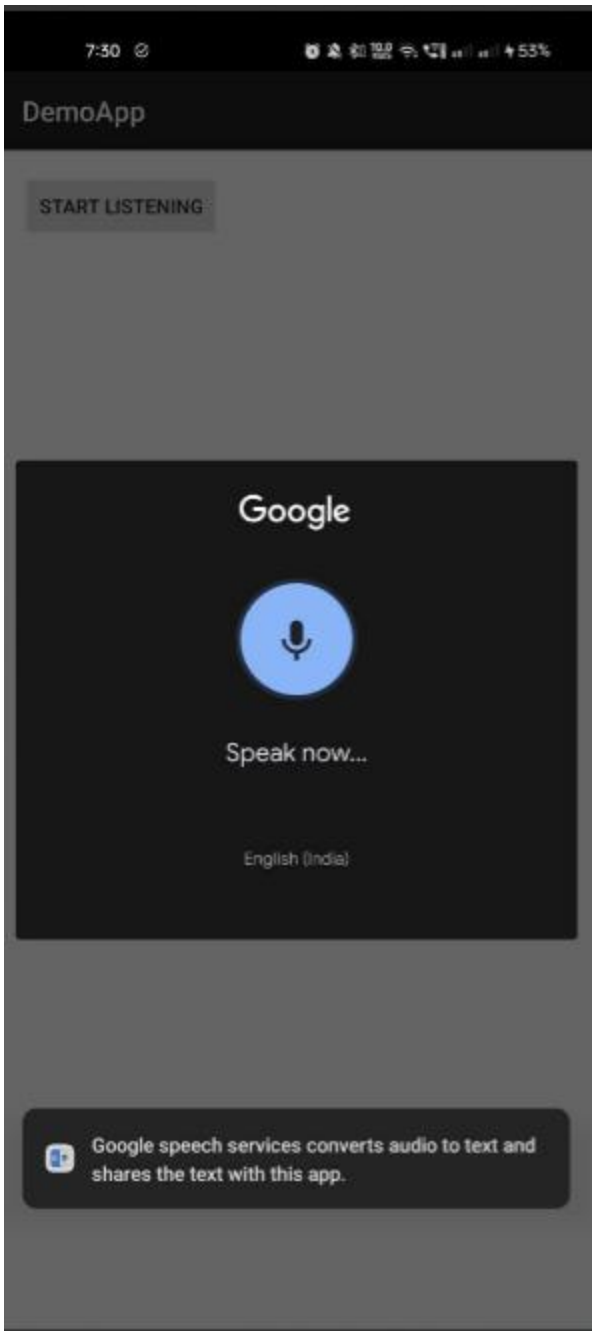
MainActivity.java

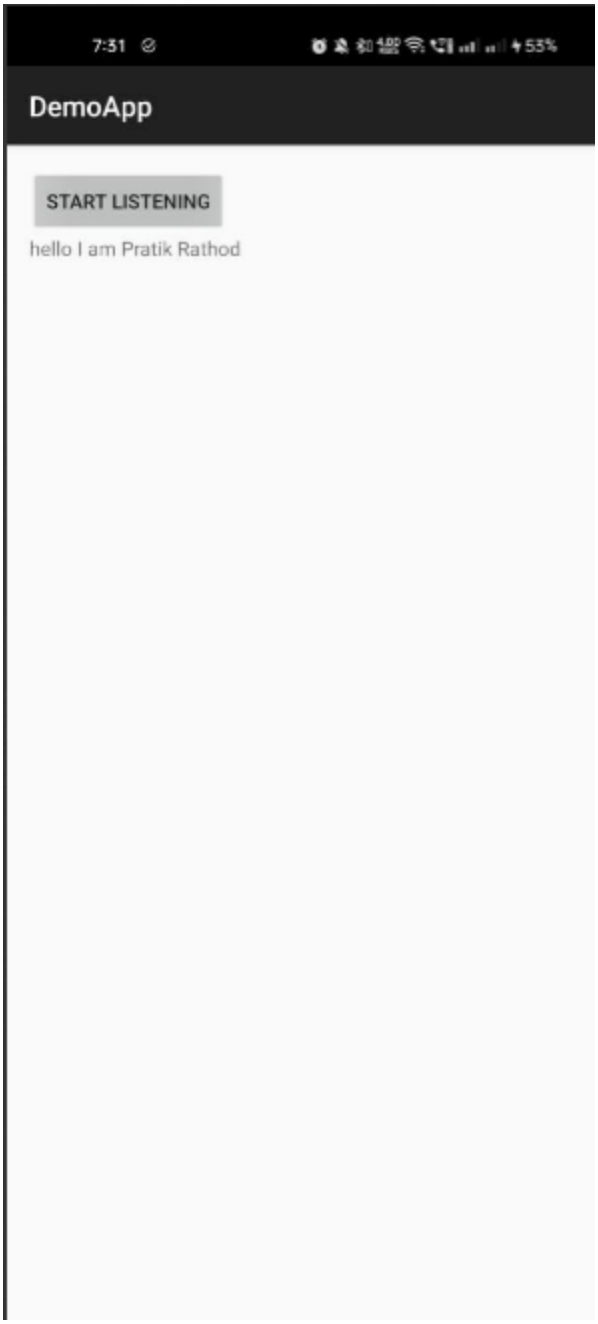
```
package com.example.forpractice;

import android.Manifest;
import android.content.ActivityNotFoundException;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.speech.RecognizerIntent;
import android.speech.SpeechRecognizer;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import java.util.ArrayList;
import java.util.Locale;

public class MainActivity extends AppCompatActivity {

    private static final int REQUEST_RECORD_AUDIO_PERMISSION = 1;
    private static final int REQ_CODE_SPEECH_INPUT = 2;
    private TextView result;

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

        result = findViewById(R.id.result);
        Button btnListen = findViewById(R.id.listen);

        btnListen.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.RECORD_AUDIO) != PackageManager.PERMISSION_GRANTED) {
                    ActivityCompat.requestPermissions(MainActivity.this, new
String[]{Manifest.permission.RECORD_AUDIO}, REQUEST_RECORD_AUDIO_PERMISSION);
                } else {
                    startListening();
                }
```

```
            }
        });
    }

    private void startListening() {
        Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.getDefault());
        intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Speak now...");

        try {
            startActivityForResult(intent, REQ_CODE_SPEECH_INPUT);
        } catch (ActivityNotFoundException a) {
            Toast.makeText(getApplicationContext(), "Speech recognition not
supported", Toast.LENGTH_SHORT).show();
        }
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
        super.onActivityResult(requestCode, resultCode, data);

        switch (requestCode) {
            case REQ_CODE_SPEECH_INPUT: {
                if (resultCode == RESULT_OK && data != null) {
                    ArrayList<String> res =
data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
                    if (res != null && !res.isEmpty()) {
                        result.setText(res.get(0));
                    } else {
                        result.setText("No speech detected");
                    }
                } else {
                    result.setText("Recognition error");
                }
                break;
            }
        }
    }

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


7:30
55%
DemoApp
START LISTENING


7:30
55%
DemoApp
START LISTENING
Allow DemoApp to record audio?
While using the app
Only this time
Don't allow





5. Develop an application that retrieves and displays the useRs current location (latitude and longitude) using the Location API. Use either FusedLocationProviderClient or LocationManager to obtain the location data. Display the location in a TextView and provide a button that refreshes the location. Additionally, show the location on a map using Google Maps API.

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <Button
        android:id="@+id/refresh"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Refresh Location" />

    <TextView
        android:id="@+id/res"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Location: " />

    <fragment
        android:id="@+id/map"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:name="com.google.android.gms.maps.SupportMapFragment" />
</LinearLayout>
```

MainActivity.xml

```
package com.example.forpractice;

import android.Manifest;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;

public class MainActivity extends AppCompatActivity implements
OnMapReadyCallback {

    private static final int REQUEST_LOCATION_PERMISSION = 1;
    private FusedLocationProviderClient fusedLocationClient;
    private TextView res;
    private GoogleMap map;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d("MainActivity", "onCreate started");

        res = findViewById(R.id.res);
        Button refresh = findViewById(R.id.refresh);

        fusedLocationClient =
LocationServices.getFusedLocationProviderClient(this);
        Log.d("MainActivity", "FusedLocationProviderClient initialized");
```

```
        refresh.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Log.d("MainActivity", "Refresh button clicked");
                if (ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED)
{
                    ActivityCompat.requestPermissions(MainActivity.this, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION},
REQUEST_LOCATION_PERMISSION);
                } else {
                    getLocation();
                }
            }
        });

        SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.map);
        if (mapFragment != null) {
            mapFragment.getMapAsync(this);
        } else {
            Log.e("MainActivity", "MapFragment is null");
        }

        Log.d("MainActivity", "onCreate finished");
    }

    private void getLocation() {
        try {
            Log.d("MainActivity", "Attempting to get location");
            fusedLocationClient.getLastLocation()
                    .addOnSuccessListener(this, location -> {
                        if (location != null) {
                            double lat = location.getLatitude();
                            double lng = location.getLongitude();
                            res.setText("Location: " + lat + ", " + lng);
                            LatLng latLng = new LatLng(lat, lng);
                            map.addMarker(new
MarkerOptions().position(latLng).title("You are here"));

map.moveCamera(CameraUpdateFactory.newLatLngZoom(latLng, 15f));
                        } else {
                            res.setText("Unable to retrieve location");
                            Log.d("MainActivity", "Location is null");
                        }
                    });
        } catch (SecurityException e) {
            Log.e("MainActivity", "SecurityException: " + e.getMessage());
        }
    }

    @Override
    public void onMapReady(GoogleMap googleMap) {
        map = googleMap;
        Log.d("MainActivity", "Map is ready");
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
        if (requestCode == REQUEST_LOCATION_PERMISSION) {
```

```
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                getLocation();
            } else {
                Toast.makeText(this, "Permission denied",
Toast.LENGTH_SHORT).show();
                Log.d("MainActivity", "Permission denied");
            }
        }
    }
}
```

6. Build an application that sends SMS messages to a specified phone number. Ensure the app properly requests and handles SMS permissions at runtime. Implement functionality to show a confirmation message or status update in a TextView after sending the SMS. Also, handle scenarios where the user denies the permission and provide an appropriate message to the user.

AndroidManifest.xml

```
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
```

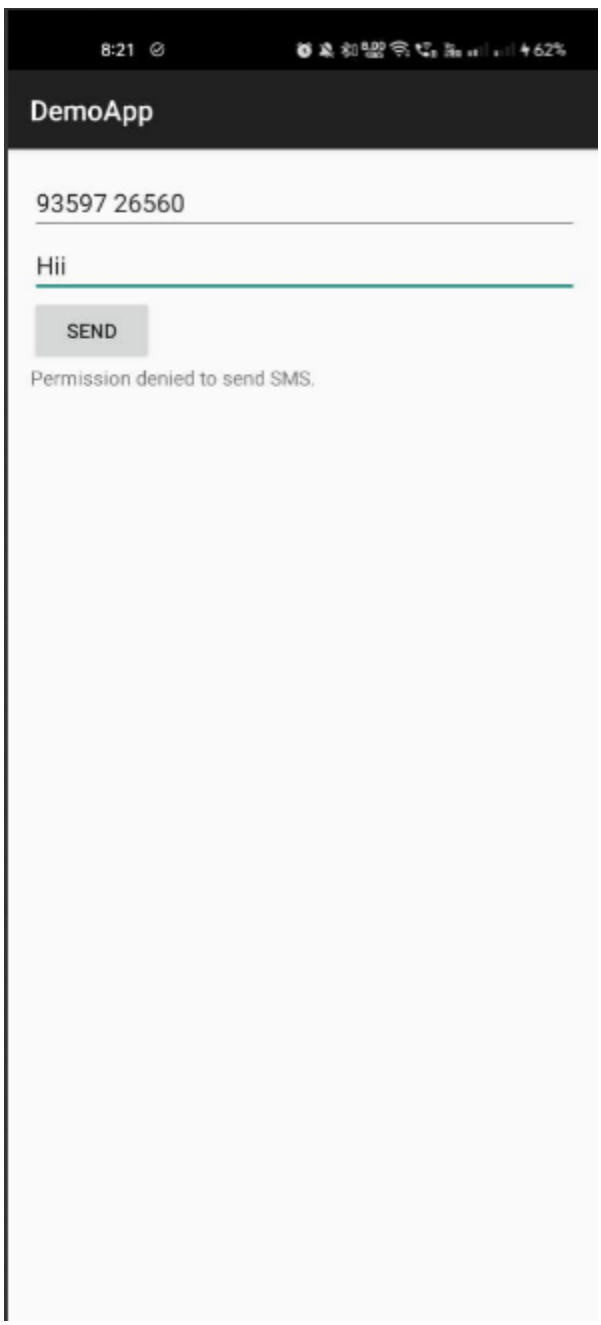
Activity_main.xml

```
<?xml version="1.0" encoding="utf-8" ?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <EditText
        android:id="@+id/num"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Phone Number" />

    <EditText
        android:id="@+id/msg"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Message" />

    <Button
        android:id="@+id/send"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Send" />

    <TextView
        android:id="@+id/status"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Status" />
</LinearLayout>
```

MainActivity.xml

```java
package com.example.forpractice;

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.TextView;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

public class MainActivity extends AppCompatActivity {

    private static final int REQUEST_SEND_SMS = 1;
    private EditText num;
    private EditText msg;
    private TextView status;

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

        num = findViewById(R.id.num);
        msg = findViewById(R.id.msg);
        status = findViewById(R.id.status);
        Button send = findViewById(R.id.send);

        send.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.SEND_SMS) != PackageManager.PERMISSION_GRANTED) {
                    ActivityCompat.requestPermissions(MainActivity.this, new
String[]{Manifest.permission.SEND_SMS}, REQUEST_SEND_SMS);
                } else {
                    sendSMS();
                }
            }
        });
    }

    private void sendSMS() {
        String phoneNumber = num.getText().toString();
        String message = msg.getText().toString();

        try {
            SmsManager smsManager = SmsManager.getDefault();
            smsManager.sendTextMessage(phoneNumber, null, message, null, null);
            status.setText("SMS sent successfully!");
        } catch (Exception e) {
            status.setText("Failed to send SMS.");
            e.printStackTrace();
        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[]
```

```
permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
        if (requestCode == REQUEST_SEND_SMS) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                sendSMS();
            } else {
                status.setText("Permission denied to send SMS.");
            }
        }
    }
}
```



7. Design an app that captures a photo using the device's camera and saves it to the external storage. After taking the photo, the app should display it in an ImageView and save the photo

to a specified directory. Implement proper handling of storage permissions and ensure the photo is stored with a unique filename to avoid overwriting existing files.

AndroidManifest.xml

```
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

Activity_main.xml

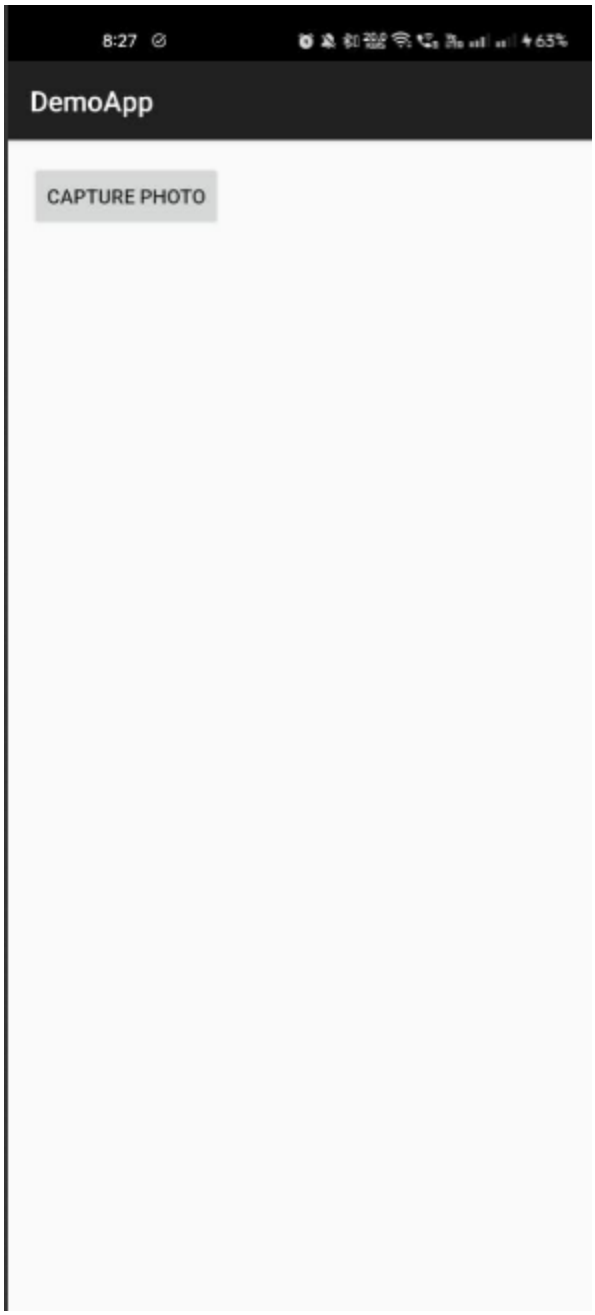
```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <Button
        android:id="@+id/capture"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Capture Photo" />

    <ImageView
        android:id="@+id/photo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:scaleType="centerCrop" />
</LinearLayout>
```

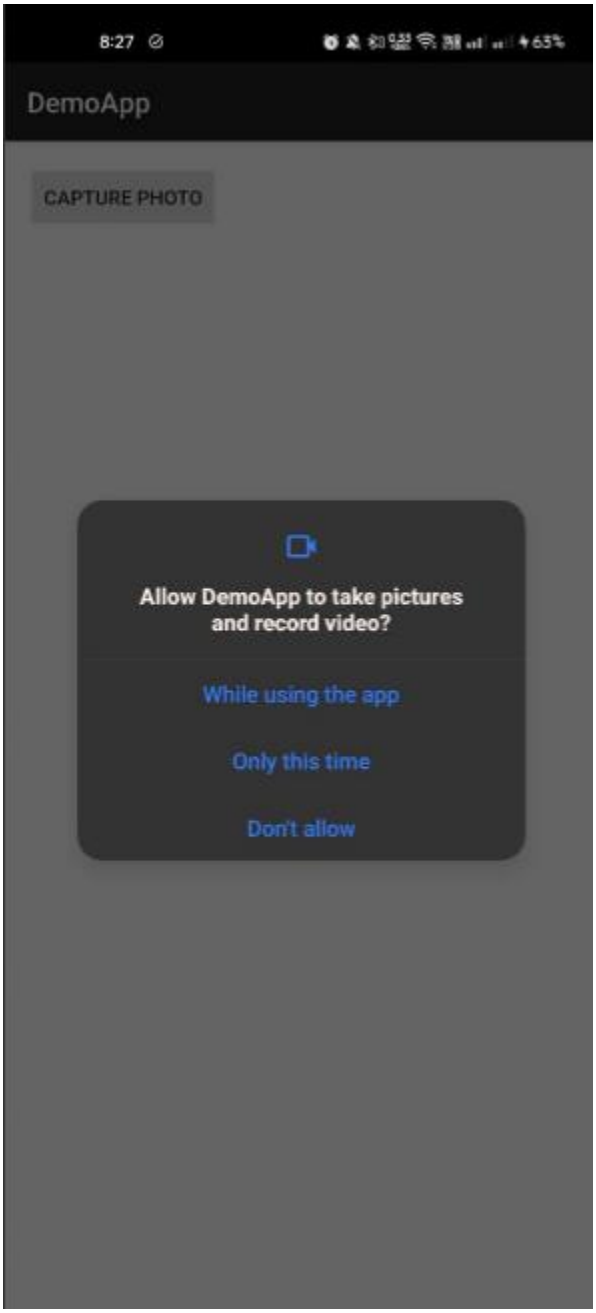
MainActivity.java

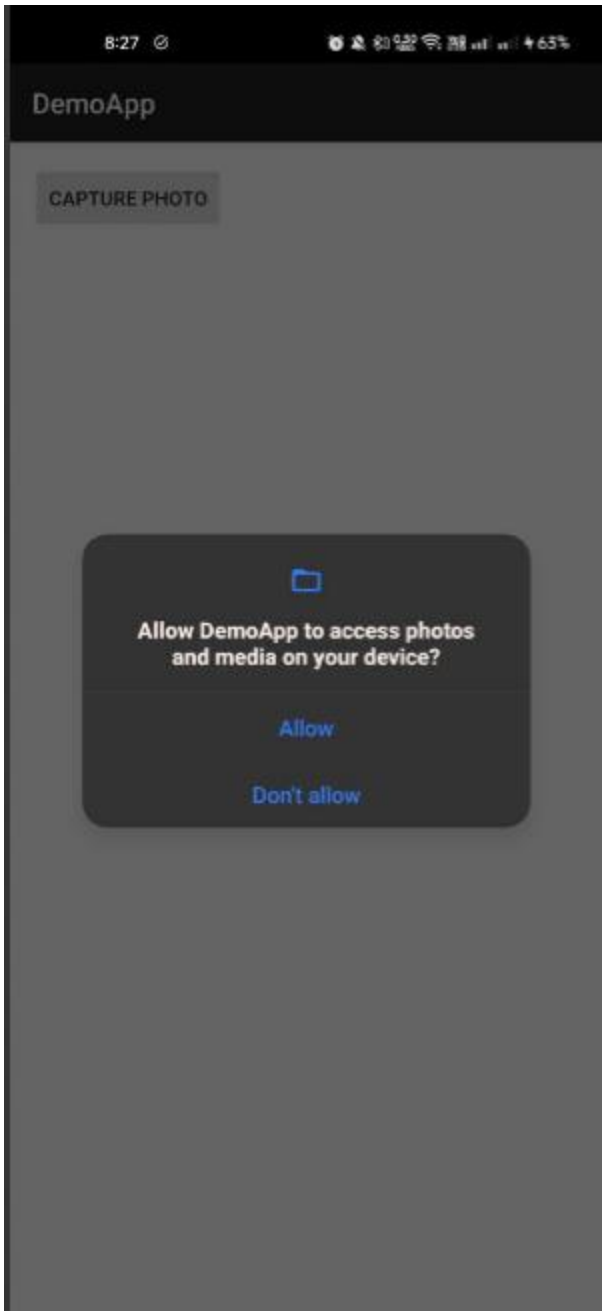
```
package com.example.forpractice;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.core.content.FileProvider;
import java.io.File;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Locale;

public class MainActivity extends AppCompatActivity {
```

```
    private static final int REQUEST_IMAGE_CAPTURE = 1;
    private static final int REQUEST_PERMISSIONS = 2;
    private ImageView photo;
    private String currentPhotoPath;

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

        photo = findViewById(R.id.photo);
        Button capture = findViewById(R.id.capture);

        capture.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED ||
                        ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.WRITE_EXTERNAL_STORAGE) !=
PackageManager.PERMISSION_GRANTED) {
                    ActivityCompat.requestPermissions(MainActivity.this, new
String[]{Manifest.permission.CAMERA,
Manifest.permission.WRITE_EXTERNAL_STORAGE}, REQUEST_PERMISSIONS);
                } else {
                    dispatchTakePictureIntent();
                }
            }
        });
    }

    private void dispatchTakePictureIntent() {
        Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        if (takePictureIntent.resolveActivity(getPackageManager()) != null) {
            File photoFile = null;
            try {
                photoFile = createImageFile();
            } catch (IOException ex) {
                Log.e("MainActivity", "Error occurred while creating the file:
" + ex.getMessage());
            }
            if (photoFile != null) {
                Uri photoURI = FileProvider.getUriForFile(this,
"com.example.android.fileprovider", photoFile);
                takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT, photoURI);
                startActivityForResult(takePictureIntent,
REQUEST_IMAGE_CAPTURE);
            }
        }
    }

    private File createImageFile() throws IOException {
        String timeStamp = new SimpleDateFormat("yyyyMMdd_HHmmss",
Locale.getDefault()).format(new Date());
        String imageFileName = "JPEG_" + timeStamp + "_";
        File storageDir = getExternalFilesDir(Environment.DIRECTORY_PICTURES);
        File image = File.createTempFile(imageFileName, ".jpg", storageDir);
        currentPhotoPath = image.getAbsolutePath();
        return image;
    }

    @Override
```

```
    protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {
            setPic();
        }
    }

    private void setPic() {
        int targetW = photo.getWidth();
        int targetH = photo.getHeight();

        BitmapFactory.Options bmOptions = new BitmapFactory.Options();
        bmOptions.inJustDecodeBounds = true;
        BitmapFactory.decodeFile(currentPhotoPath, bmOptions);
        int photoW = bmOptions.outWidth;
        int photoH = bmOptions.outHeight;

        int scaleFactor = Math.min(photoW / targetW, photoH / targetH);

        bmOptions.inJustDecodeBounds = false;
        bmOptions.inSampleSize = scaleFactor;
        Bitmap bitmap = BitmapFactory.decodeFile(currentPhotoPath, bmOptions);
        photo.setImageBitmap(bitmap);
    }

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


8:27
65%
DemoApp
CAPTURE PHOTO


8:27
65%
DemoApp
CAPTURE PHOTO
Allow DemoApp to take pictures and record video?
While using the app
Only this time
Don't allow



8. Create an application that monitors both incoming and outgoing phone calls. Use the Telephony API to listen for call state changes and record details such as the callers phone number and call duration. Display this information in a ListView or RecyclerView, and ensure the app handles call logs and permissions appropriately.

AndroidManifest.xml

```
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.READ_CALL_LOG"/>
<uses-permission android:name="android.permission.READ_CONTACTS"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <Button
        android:id="@+id/refresh"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Refresh" />

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rv"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```

Item_call_log.xml

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

    <TextView
        android:id="@+id/phone"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Phone Number" />

    <TextView
        android:id="@+id/duration"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Duration" />

    <TextView
        android:id="@+id/type"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Type" />
</LinearLayout>
```

MainActivity.java

```
package com.example.forpractice;

import android.Manifest;
import android.content.ContentResolver;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.os.Bundle;
import android.provider.CallLog;
import android.util.Log;
import android.view.View;
import android.widget.Button;
```

```
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {

    private static final int REQUEST_PERMISSIONS = 1;
    private RecyclerView rv;
    private CallLogAdapter adapter;
    private List<CallLogItem> callLogList = new ArrayList<>();

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

        rv = findViewById(R.id.rv);
        Button refresh = findViewById(R.id.refresh);

        rv.setLayoutManager(new LinearLayoutManager(this));
        adapter = new CallLogAdapter(callLogList);
        rv.setAdapter(adapter);

        refresh.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.READ_CALL_LOG) != PackageManager.PERMISSION_GRANTED ||
                        ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.READ_PHONE_STATE) != PackageManager.PERMISSION_GRANTED) {
                    ActivityCompat.requestPermissions(MainActivity.this, new
String[]{
                            Manifest.permission.READ_CALL_LOG,
                            Manifest.permission.READ_PHONE_STATE},
REQUEST_PERMISSIONS);
                } else {
                    getCallDetails();
                }
            }
        });

        if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_CALL_LOG) == PackageManager.PERMISSION_GRANTED &&
                ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_PHONE_STATE) == PackageManager.PERMISSION_GRANTED) {
            getCallDetails();
        }
    }

    private void getCallDetails() {
        callLogList.clear();
        ContentResolver cr = getContentResolver();
        Cursor cursor = cr.query(CallLog.Calls.CONTENT_URI, null, null, null,
CallLog.Calls.DATE + " DESC");

        if (cursor != null) {
            while (cursor.moveToNext()) {
```

```
                String phoneNumber =
cursor.getString(cursor.getColumnIndex(CallLog.Calls.NUMBER));
                String callType =
cursor.getString(cursor.getColumnIndex(CallLog.Calls.TYPE));
                String callDate =
cursor.getString(cursor.getColumnIndex(CallLog.Calls.DATE));
                String callDuration =
cursor.getString(cursor.getColumnIndex(CallLog.Calls.DURATION));

                String dir = null;
                int dirCode = Integer.parseInt(callType);
                switch (dirCode) {
                    case CallLog.Calls.OUTGOING_TYPE:
                        dir = "OUTGOING";
                        break;
                    case CallLog.Calls.INCOMING_TYPE:
                        dir = "INCOMING";
                        break;
                    case CallLog.Calls.MISSED_TYPE:
                        dir = "MISSED";
                        break;
                }

                callLogList.add(new CallLogItem(phoneNumber,
Integer.parseInt(callDuration), dir));
            }
            cursor.close();
        }

        adapter.notifyDataSetChanged();
    }

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

CallLogAdapter.java

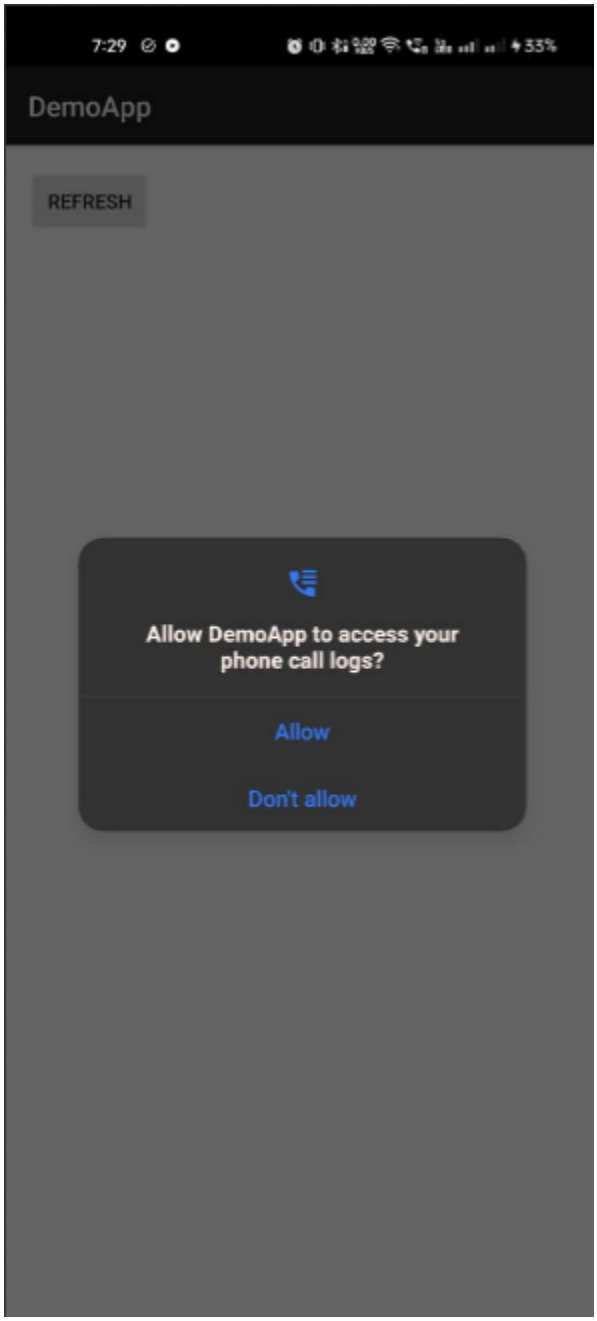
```
package com.example.forpractice;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
import java.util.List;

public class CallLogAdapter extends
RecyclerView.Adapter<CallLogAdapter.ViewHolder> {
```

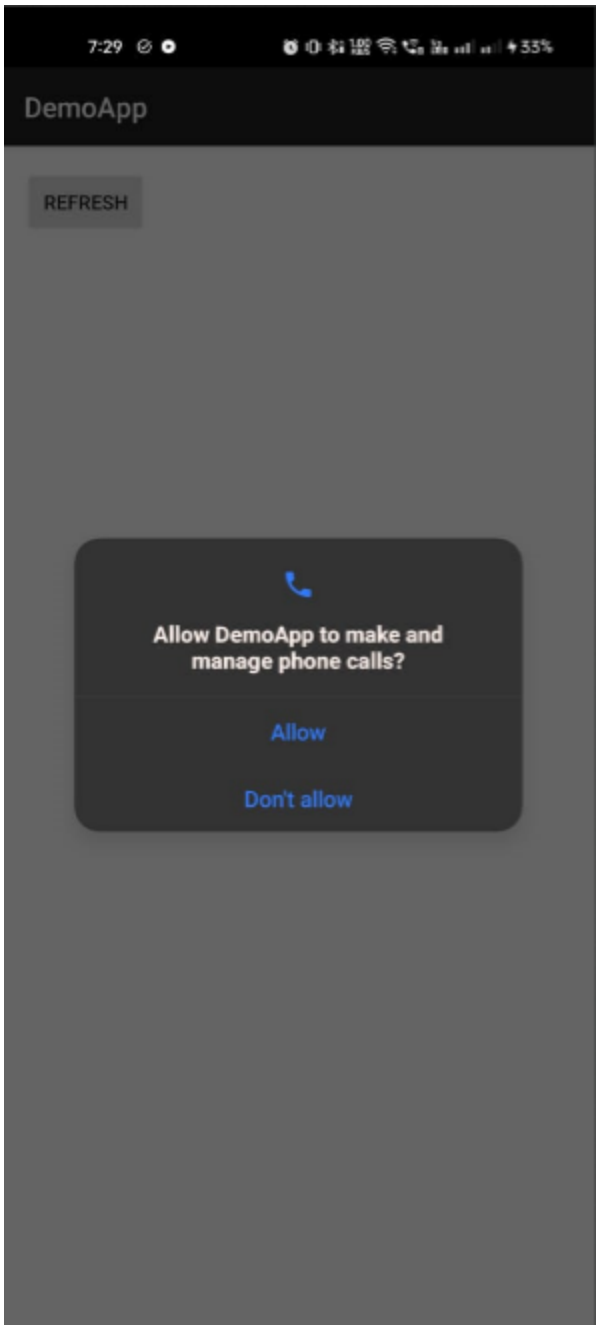
```
    private final List<CallLogItem> callLogList;

    public CallLogAdapter(List<CallLogItem> callLogList) {
        this.callLogList = callLogList;
    }

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {
        View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_call_log,
parent, false);
        return new ViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
        CallLogItem callLog = callLogList.get(position);
        holder.phone.setText(callLog.getPhoneNumber());
        holder.duration.setText(String.valueOf(callLog.getCallDuration()));
        holder.type.setText(callLog.getCallType());
    }

    @Override
    public int getItemCount() {
        return callLogList.size();
    }

    public static class ViewHolder extends RecyclerView.ViewHolder {
        public TextView phone, duration, type;

        public ViewHolder(View itemView) {
            super(itemView);
            phone = itemView.findViewById(R.id.phone);
            duration = itemView.findViewById(R.id.duration);
            type = itemView.findViewById(R.id.type);
        }
    }
}
```

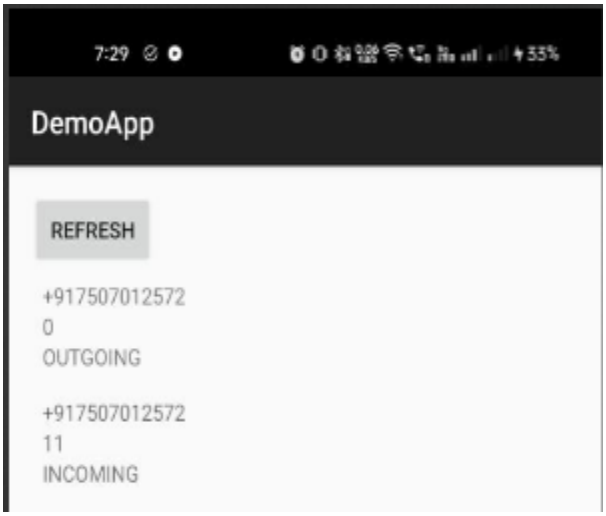
CallLogItem.java

```
package com.example.forpractice;

public class CallLogItem {
    private String phoneNumber;
    private int callDuration;
    private String callType;

    public CallLogItem(String phoneNumber, int callDuration, String callType) {
        this.phoneNumber = phoneNumber;
        this.callDuration = callDuration;
        this.callType = callType;
    }

    public String getPhoneNumber() {
        return phoneNumber;
    }

    public int getCallDuration() {
```

```
        return callDuration;
    }

    public String getCallType() {
        return callType;
    }
}
```







9. Develop an app that uses speech recognition to convert spoken words into text and provides spoken feedback using Text-to-Speech. Implement a button to start speech recognition and another button to convert text into speech. Display the recognized text in a TextView and use Text-to Speech to read the text aloud when the user clicks the corresponding button.

AndroidManifest.xml

```
<uses-permission android:name="android.permission.RECORD_AUDIO"/>
```

Acvitivity_main.xml

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

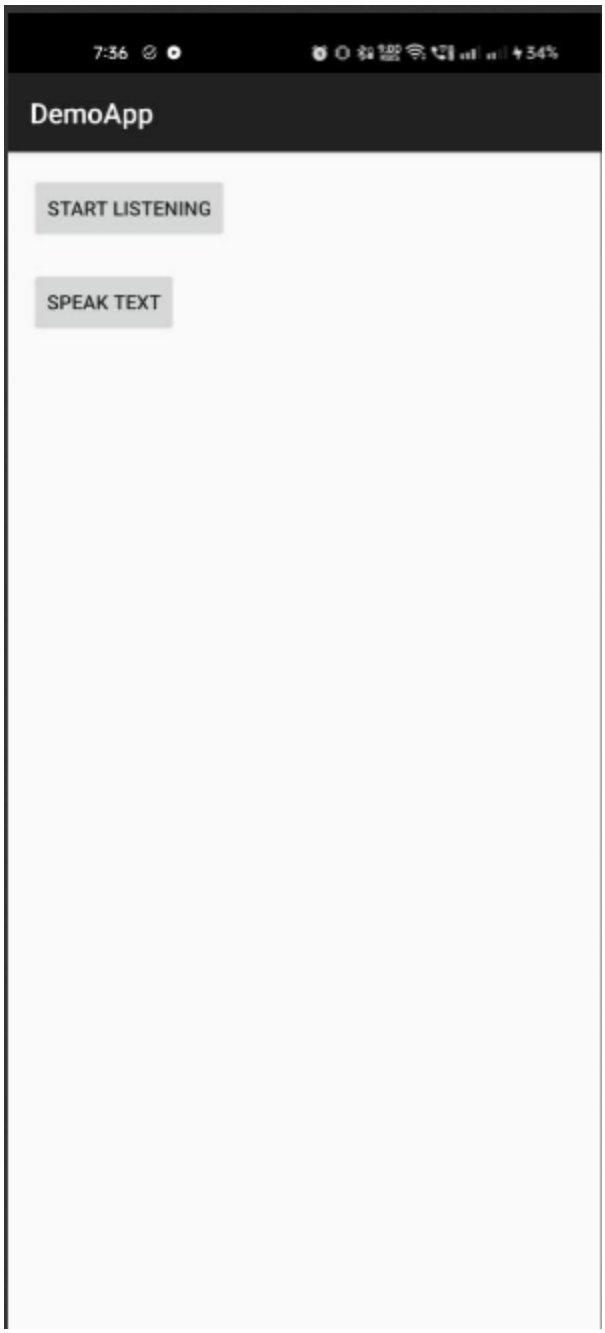
```
    android:padding="16dp">

    <Button
        android:id="@+id/btnRec"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Start Listening" />

    <TextView
        android:id="@+id/txtResult"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="" />

    <Button
        android:id="@+id/btnSpeak"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Speak Text" />
</LinearLayout>
```

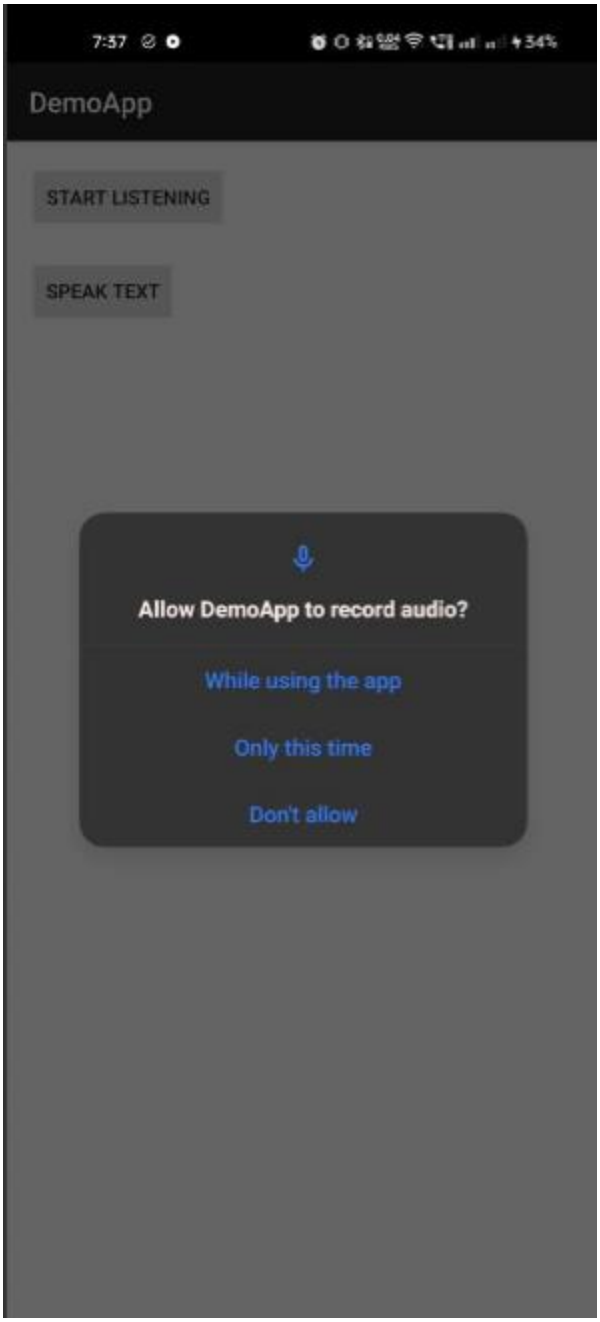
MainActivity.java

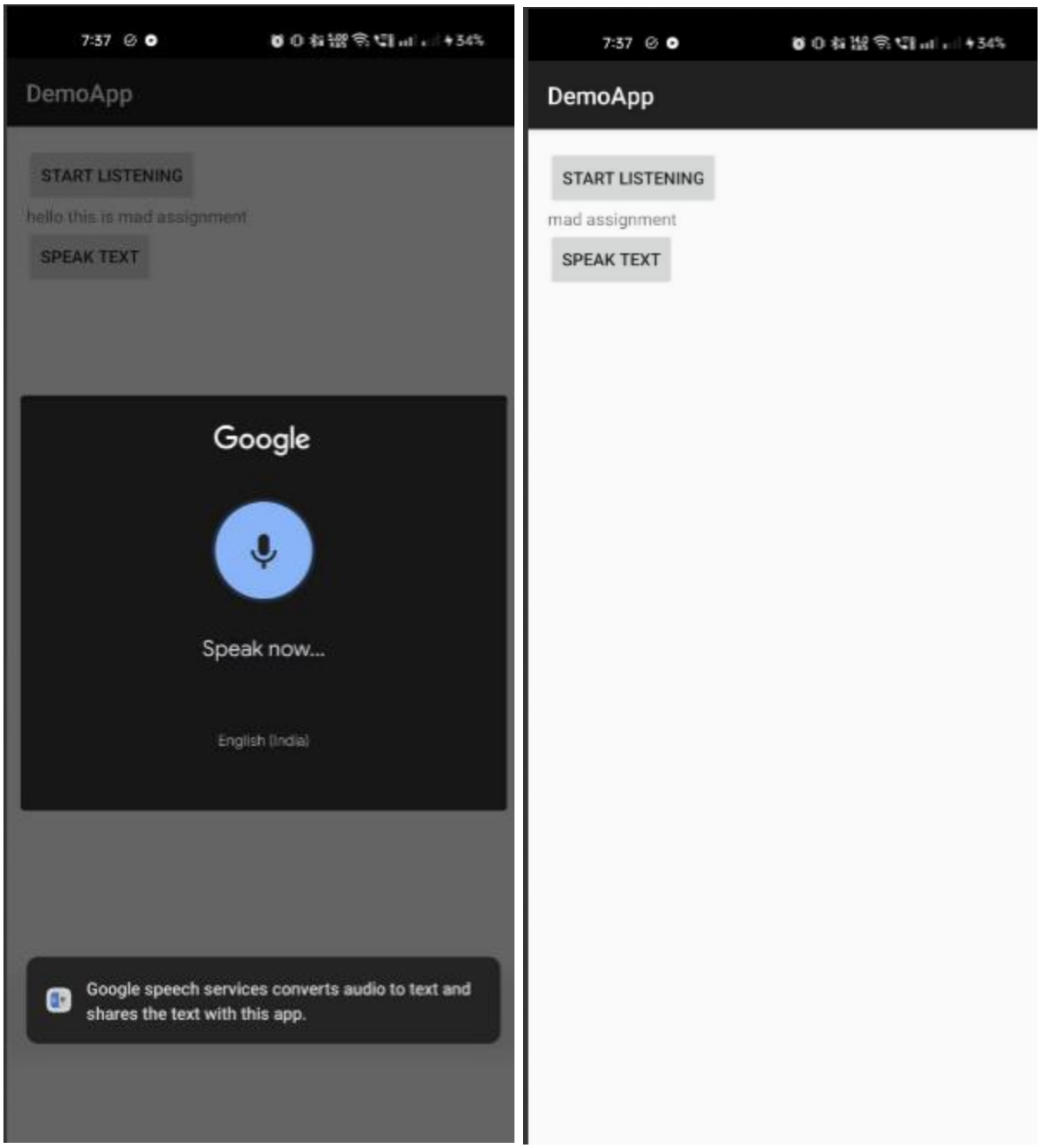
```
package com.example.forpractice;

import android.Manifest;
import android.content.ActivityNotFoundException;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.speech.RecognizerIntent;
import android.speech.tts.TextToSpeech;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import java.util.ArrayList;
import java.util.Locale;

public class MainActivity extends AppCompatActivity {

    private static final int REQUEST_RECORD_AUDIO_PERMISSION = 1;
    private static final int REQ_CODE_SPEECH_INPUT = 2;
    private TextView txtResult;
    private TextToSpeech tts;

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

        txtResult = findViewById(R.id.txtResult);
        Button btnRec = findViewById(R.id.btnRec);
        Button btnSpeak = findViewById(R.id.btnSpeak);

        btnRec.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
```

```
                if (ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.RECORD_AUDIO) != PackageManager.PERMISSION_GRANTED) {
                    ActivityCompat.requestPermissions(MainActivity.this, new
String[]{Manifest.permission.RECORD_AUDIO}, REQUEST_RECORD_AUDIO_PERMISSION);
                } else {
                    startListening();
                }
            }
        });

        btnSpeak.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                speakText();
            }
        });

        tts = new TextToSpeech(this, new TextToSpeech.OnInitListener() {
            @Override
            public void onInit(int status) {
                if (status == TextToSpeech.SUCCESS) {
                    int result = tts.setLanguage(Locale.getDefault());
                    if (result == TextToSpeech.LANG_MISSING_DATA || result ==
TextToSpeech.LANG_NOT_SUPPORTED) {
                        Log.e("TTS", "Language not supported");
                    }
                } else {
                    Log.e("TTS", "Initialization failed");
                }
            }
        });
    }

    private void startListening() {
        Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.getDefault());
        intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Speak now...");

        try {
            startActivityForResult(intent, REQ_CODE_SPEECH_INPUT);
        } catch (ActivityNotFoundException a) {
            Toast.makeText(getApplicationContext(), "Speech recognition not
supported", Toast.LENGTH_SHORT).show();
        }
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
        super.onActivityResult(requestCode, resultCode, data);

        switch (requestCode) {
            case REQ_CODE_SPEECH_INPUT: {
                if (resultCode == RESULT_OK && data != null) {
                    ArrayList<String> result =
data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
                    if (result != null && !result.isEmpty()) {
                        txtResult.setText(result.get(0));
                    } else {
                        txtResult.setText("No speech detected");
                    }
```

```
                } else {
                    txtResult.setText("Recognition error");
                }
                break;
            }
        }
    }

    private void speakText() {
        String text = txtResult.getText().toString();
        if (!text.isEmpty()) {
            tts.speak(text, TextToSpeech.QUEUE_FLUSH, null, null);
        } else {
            Toast.makeText(this, "No text to speak",
Toast.LENGTH_SHORT).show();
        }
    }

    @Override
    protected void onDestroy() {
        if (tts != null) {
            tts.stop();
            tts.shutdown();
        }
        super.onDestroy();
    }

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


7:36
34%
DemoApp
START LISTENING
SPEAK TEXT


7:37
34%
DemoApp
START LISTENING
SPEAK TEXT
Allow DemoApp to record audio?
While using the app
Only this time
Don't allow



10. Create an application that tracks the users location and calculates the distance traveled between two points. Use the Location API to obtain the user's current location at different intervals. Implement functionality to calculate the distance between the starting location and the current location and display this distance in a TextView.

AndroidManifest.xml

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.INTERNET" />
```

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
```

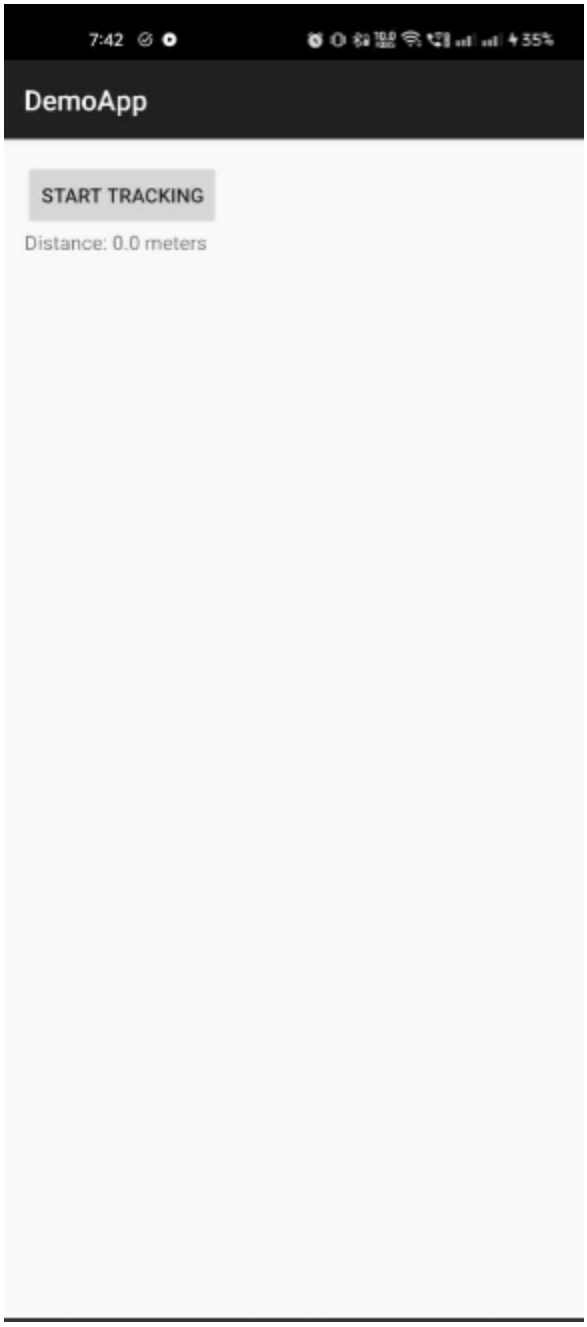
```
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <Button
        android:id="@+id/start"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Start Tracking" />

    <TextView
        android:id="@+id/distance"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Distance: 0.0 meters" />
</LinearLayout>
```

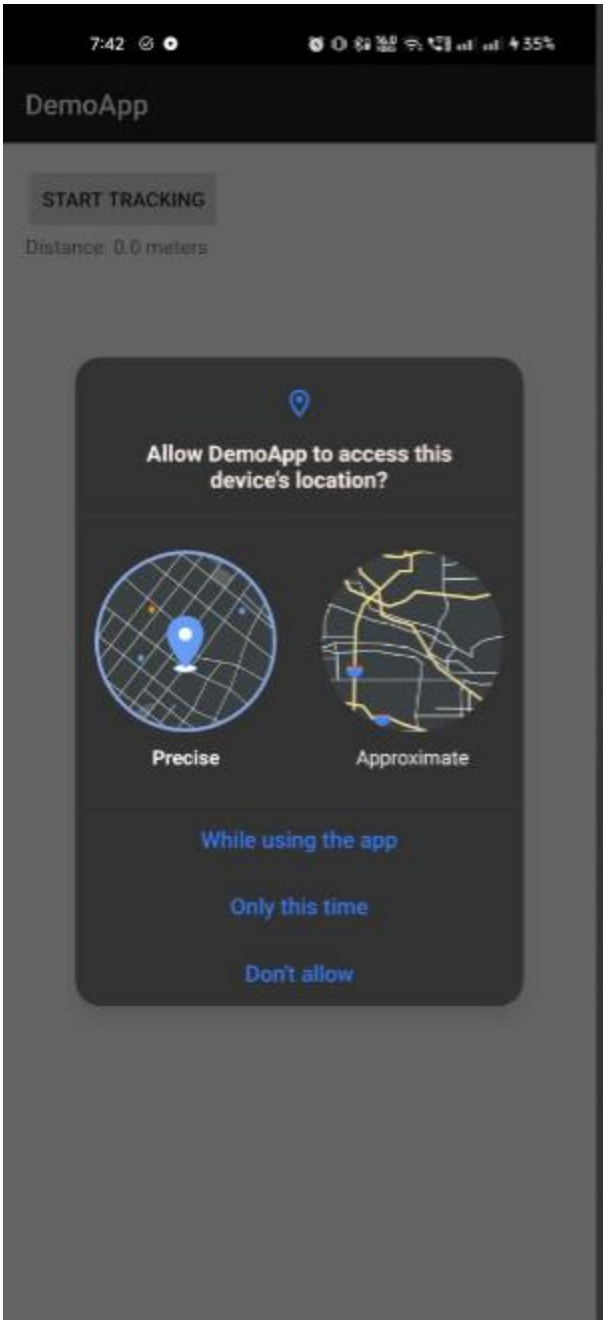
MainActivity.java

```
package com.example.forpractice;

import android.Manifest;
import android.content.pm.PackageManager;
import android.location.Location;
import android.os.Bundle;
import android.os.Handler;
import android.os.Looper;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationCallback;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationResult;
import com.google.android.gms.location.LocationServices;

public class MainActivity extends AppCompatActivity {

    private static final int REQUEST_LOCATION_PERMISSION = 1;
    private FusedLocationProviderClient fusedLocationClient;
    private LocationCallback locationCallback;
    private TextView distance;
    private Location startLocation;
    private float totalDistance = 0;

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

        distance = findViewById(R.id.distance);
        Button start = findViewById(R.id.start);

        fusedLocationClient =
LocationServices.getFusedLocationProviderClient(this);
```

```
        start.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED)
{
                    ActivityCompat.requestPermissions(MainActivity.this, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION},
REQUEST_LOCATION_PERMISSION);
                } else {
                    startLocationUpdates();
                }
            }
        });

        locationCallback = new LocationCallback() {
            @Override
            public void onLocationResult(@NonNull LocationResult
locationResult) {
                if (locationResult == null) {
                    return;
                }
                for (Location location : locationResult.getLocations()) {
                    if (startLocation == null) {
                        startLocation = location;
                    }
                    totalDistance += startLocation.distanceTo(location);
                    startLocation = location;
                    distance.setText("Distance: " + totalDistance + " meters");
                }
            }
        };
    }

    private void startLocationUpdates() {
        LocationRequest locationRequest = LocationRequest.create();
        locationRequest.setInterval(10000);
        locationRequest.setFastestInterval(5000);
        locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);

        try {
            fusedLocationClient.requestLocationUpdates(locationRequest,
locationCallback, Looper.getMainLooper());
        } catch (SecurityException e) {
            Log.e("MainActivity", "SecurityException: " + e.getMessage());
        }
    }

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

```
        }
}
```


7:42
35%
DemoApp
START TRACKING
Distance: 0.0 meters


7:42
35%
DemoApp
START TRACKING
Distance: 0.0 meters
Allow DemoApp to access this device's location?
Precise
Approximate
While using the app
Only this time
Don't allow