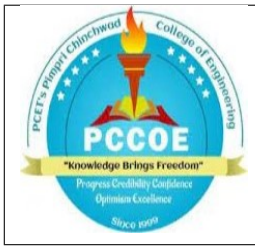


Pimpri Chinchwad Education Trust's
Pimpri Chinchwad College of Engineering

Lab Assignment - 05

PRN: 123M1H041 **Batch: 2**
Name of Student: Darshan Pathak **Submission Date: Oct. 24, 2024**

Questions

Q1. 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.

Solution

`MainActivity.java`

```
1  package com.darshan.smsapp;
2
3
4  import android.Manifest;
5  import android.content.pm.PackageManager;
6  import android.os.Bundle;
7  import android.telephony.SmsManager;
8  import android.view.View;
9  import android.widget.Button;
10 import android.widget.EditText;
11 import android.widget.TextView;
12 import androidx.appcompat.app.AppCompatActivity;
13 import androidx.core.app.ActivityCompat;
14 import androidx.core.content.ContextCompat;
15
16 public class MainActivity extends AppCompatActivity {
17
18     private static final int MY_PERMISSIONS_REQUEST_SEND_SMS = 0;
19
20     @Override
21     protected void onCreate(Bundle savedInstanceState) {
22         super.onCreate(savedInstanceState);
```

```
23          setContentView(R.layout.activity_main);
24
25          EditText phoneT = findViewById(R.id.phone);
26          EditText messageT = findViewById(R.id.message);
27          Button send = findViewById(R.id.send);
28          TextView received = findViewById(R.id.received);
29
30          send.setOnClickListener(new View.OnClickListener() {
31              @Override
32              public void onClick(View v) {
33                  String phone = phoneT.getText().toString();
34                  String message = messageT.getText().toString();
35
36                  if (ContextCompat.checkSelfPermission(MainActivity.
                        this,
37                           Manifest.permission.SEND_SMS) !=
                              PackageManager.PERMISSION_GRANTED) {
38                       ActivityCompat.requestPermissions(MainActivity.
                            this,
39                                new String[]{Manifest.permission.SEND_SMS
                                    }, MY_PERMISSIONS_REQUEST_SEND_SMS);
40                  } else {
41                      SmsManager smsManager = SmsManager.getDefault();
42                      smsManager.sendTextMessage(phone, null, message,
                            null, null);
43                  }
44              }
45          });
46      }
47
48      @Override
49      public void onRequestPermissionsResult(int requestCode, String[]
            permissions, int[] grantResults) {
50          super.onRequestPermissionsResult(requestCode, permissions,
                grantResults);
51          switch (requestCode) {
52              case MY_PERMISSIONS_REQUEST_SEND_SMS: {
53                  if (grantResults.length > 0 && grantResults[0] ==
                        PackageManager.PERMISSION_GRANTED) {
54
55                  } else {
56
57                  }
58                  return;
59              }
60          }
61      }
62  }
```

SmsReceiver.java

```java
package com.darshan.smsapp;

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);
        }
    }
}
```

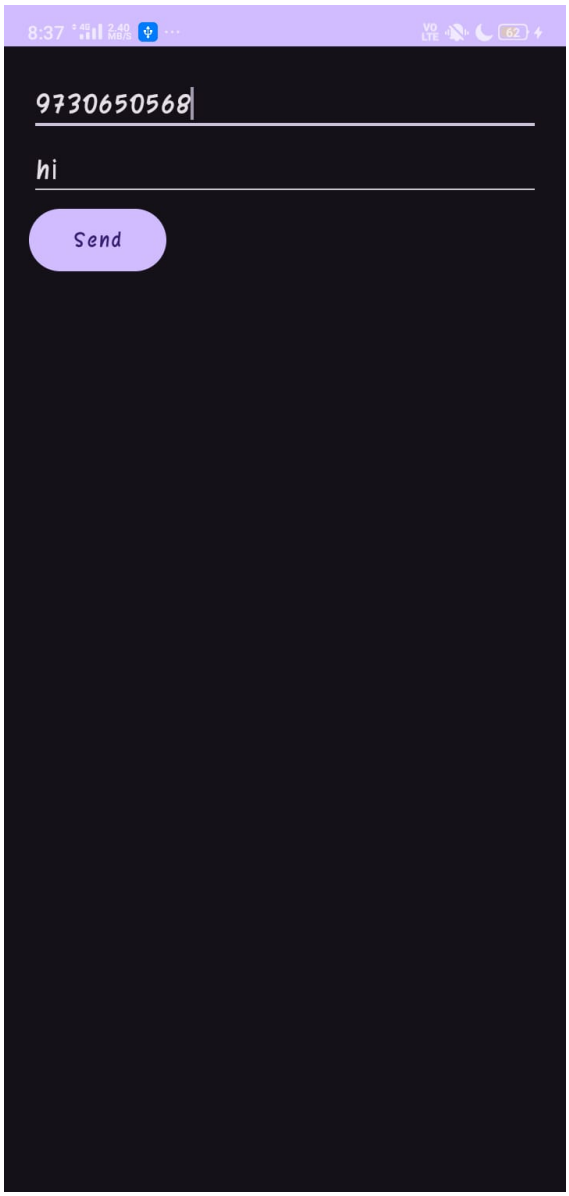
---

activity_main.xml

```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>
```

Output



(a) Main Screen

Q2. 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.

Solution

`MainActivity.java`

```java
package com.darshan.cameraapp;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
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
```

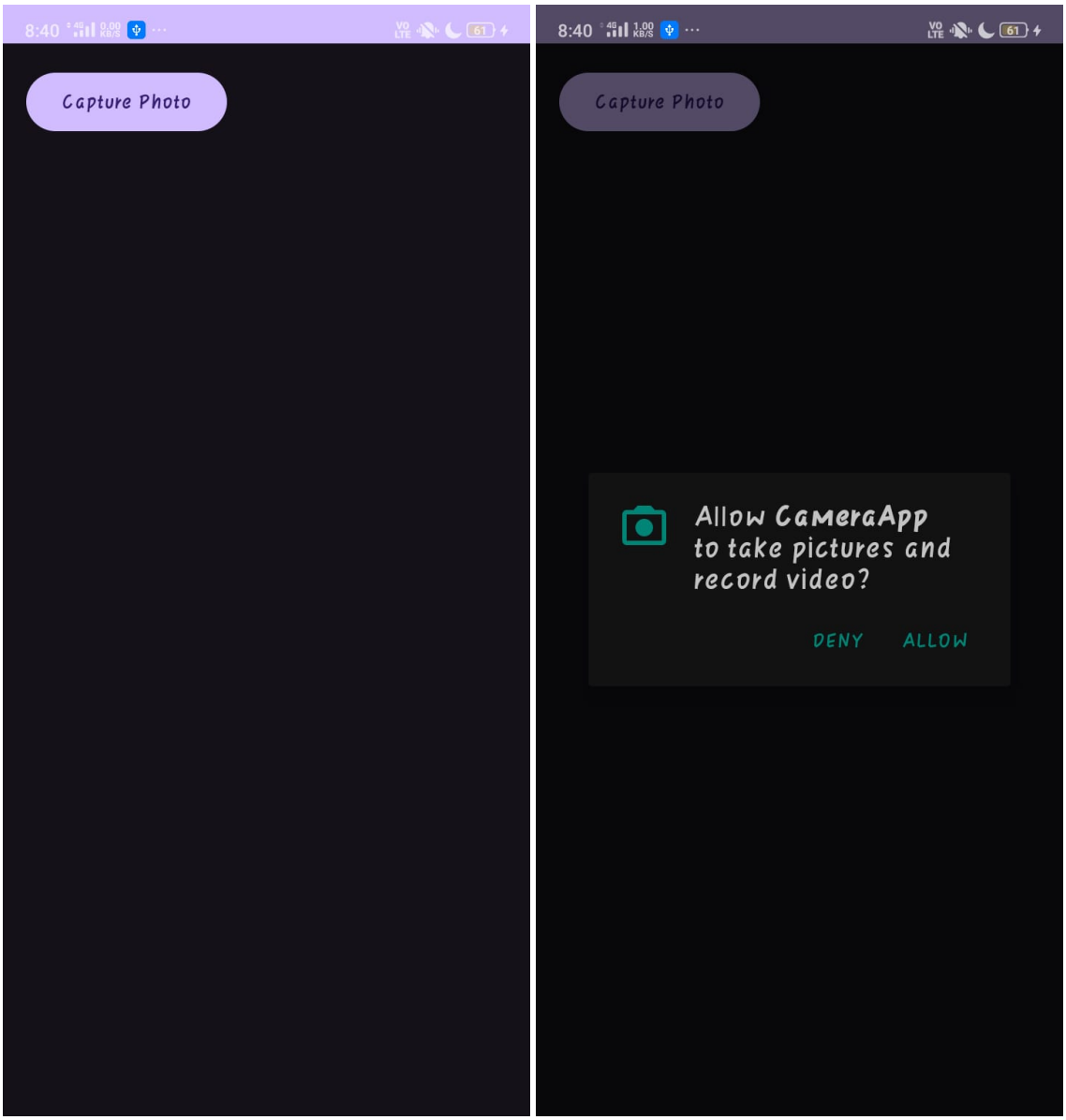
```
45      protected void onActivityResult(int requestCode, int resultCode,
            @Nullable
46      Intent data) {
47          super.onActivityResult(requestCode, resultCode, data);
48          if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode ==
                RESULT_OK) {
49              Bundle extras = data.getExtras();
50              Bitmap imageBitmap = (Bitmap) extras.get("data");
51              imgPhoto.setImageBitmap(imageBitmap);
52          }
53      }
54  }
```

---

activity_main.xml

```
1  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/
       android"
2      android:layout_width="match_parent"
3      android:layout_height="match_parent"
4      android:orientation="vertical"
5      android:padding="16dp">
6      <Button
7          android:id="@+id/btnCapture"
8          android:layout_width="wrap_content"
9          android:layout_height="wrap_content"
10         android:text="Capture Photo" />
11     <ImageView
12         android:id="@+id/imgPhoto"
13         android:layout_width="match_parent"
14         android:layout_height="wrap_content"
15         android:scaleType="centerCrop" />
16 </LinearLayout>
```

Output



(a) Main Screen (b) Notification

Q3. 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.

Solution

`MainActivity.java`

```java
package com.darshan.callsapp;

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) {
```

```
47              ActivityCompat.requestPermissions(MainActivity.this, new
48                              String[]{Manifest.permission.
                                    READ_PHONE_STATE},
49                      REQUEST_PHONE_STATE_PERMISSION);
50          } else if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.S) {
51              listenForCallStateChanges();
52          }
53      }
54      private void makeCall() {
55          String phoneNumber = etNum.getText().toString();
56          Intent callIntent = new Intent(Intent.ACTION_CALL);
57          callIntent.setData(Uri.parse("tel:" + phoneNumber));
58          if (ActivityCompat.checkSelfPermission(this,
59                  Manifest.permission.CALL_PHONE) == PackageManager.
                        PERMISSION_GRANTED) {
60              startActivity(callIntent);
61          }
62      }
63      @RequiresApi(api = Build.VERSION_CODES.S)
64      private void listenForCallStateChanges() {
65          TelephonyManager telephonyManager = (TelephonyManager)
66                  getSystemService(TELEPHONY_SERVICE);
67          if (telephonyManager != null) {
68              telephonyManager.registerTelephonyCallback(
                        getMainExecutor(), new
69                      CustomTelephonyCallback());
70          } else {
71              Log.e("MainActivity", "TelephonyManager is null");
72          }
73      }
74      @RequiresApi(api = Build.VERSION_CODES.S)
75      private class CustomTelephonyCallback extends TelephonyCallback
            implements
76              TelephonyCallback.CallStateListener {
77          @Override
78          public void onCallStateChanged(int state) {
79              switch (state) {
80                  case TelephonyManager.CALL_STATE_RINGING:
81                      tvState.setText("Ringing");
82                      break;
83                  case TelephonyManager.CALL_STATE_OFFHOOK:
84                      tvState.setText("Answered");
85                      break;
86                  case TelephonyManager.CALL_STATE_IDLE:
87                      tvState.setText("Idle");
88                      break;
89                  default:
90                      Log.e("CustomTelephonyCallback", "Unknown call
                                state: " +
91                              state);
92                      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");
            }
        }
    }
}
```

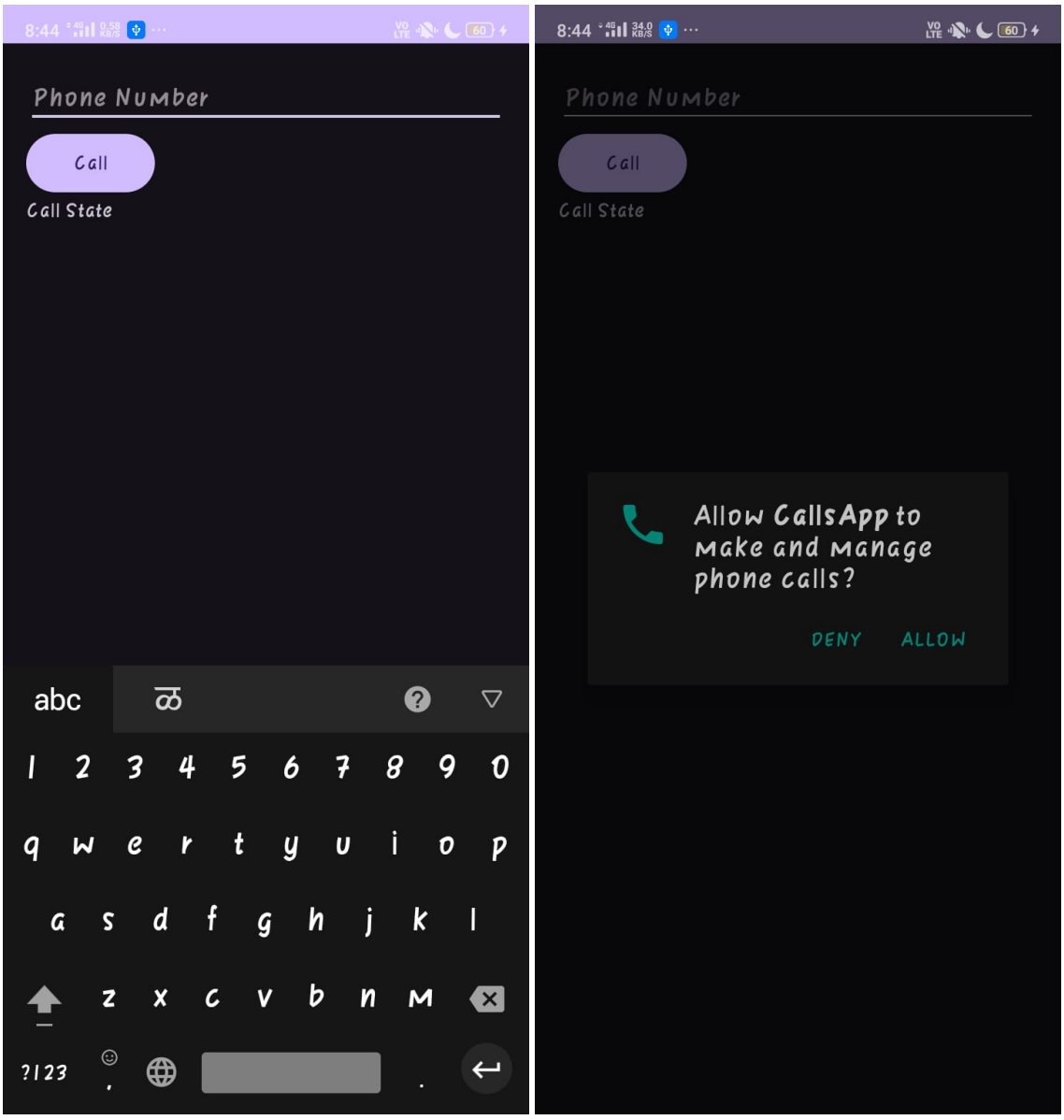
---

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
```

```
17          android:id="@+id/tvState"
18          android:layout_width="match_parent"
19          android:layout_height="wrap_content"
20          android:text="Call State" />
21  </LinearLayout>
```

Output



(a) Main Screen (b) Notification

Q4. 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.

Solution

`MainActivity.java`

```
package com.darshan.myvoiceapp;

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.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);
```

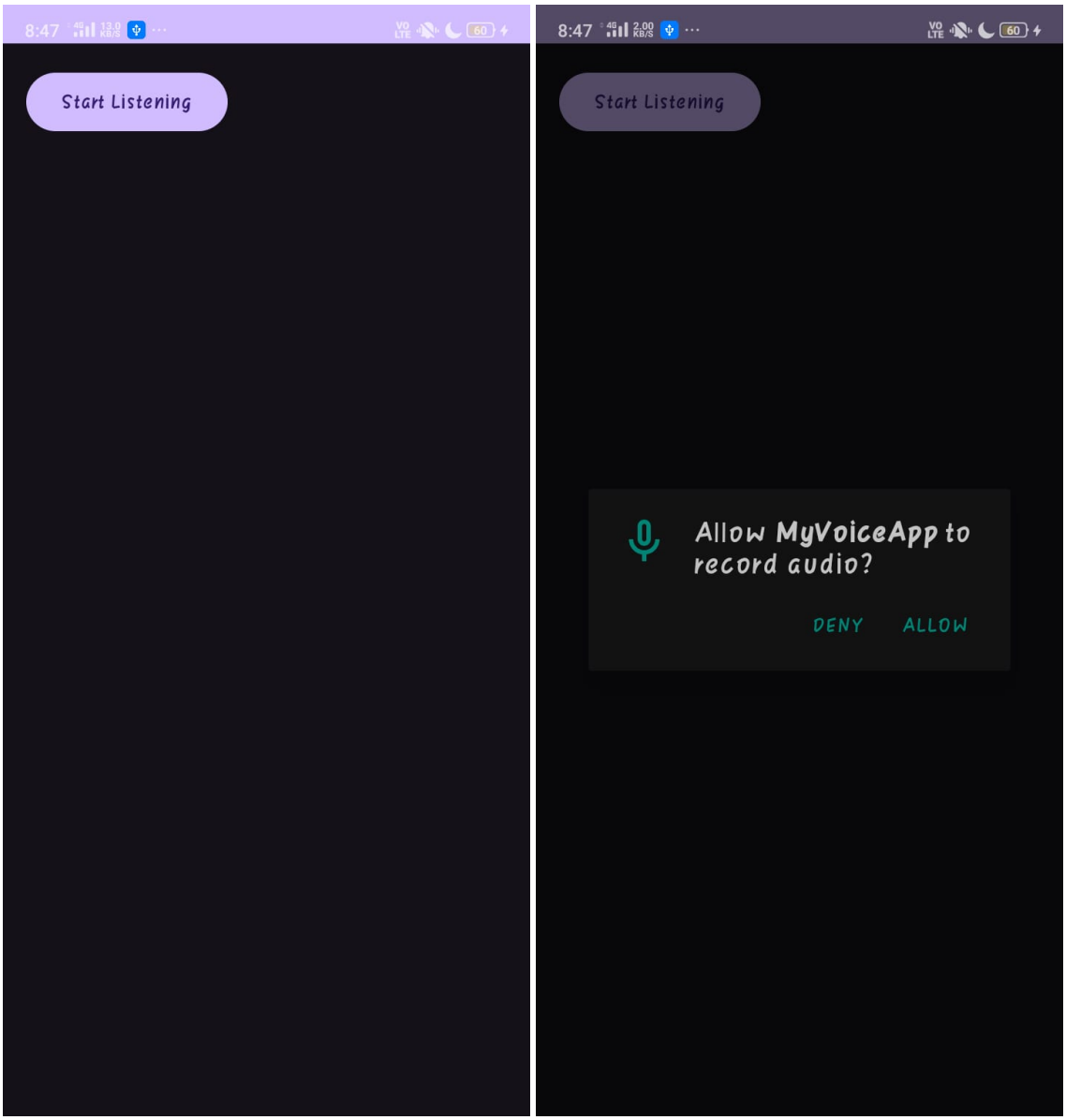
```
46          intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.
               getDefault());
47          intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Speak now..."
               );
48          try {
49              startActivityForResult(intent, REQ_CODE_SPEECH_INPUT);
50          } catch (ActivityNotFoundException a) {
51              Toast.makeText(getApplicationContext(), "Speech
                   recognition not supported", Toast.LENGTH_SHORT).show()
                   ;
52          }
53      }
54      @Override
55      protected void onActivityResult(int requestCode, int resultCode,
           Intent
56              data) {
57          super.onActivityResult(requestCode, resultCode, data);
58          switch (requestCode) {
59              case REQ_CODE_SPEECH_INPUT: {
60                  if (resultCode == RESULT_OK && data != null) {
61                      ArrayList<String> res =
62                              data.getStringArrayListExtra(
                                   RecognizerIntent.EXTRA_RESULTS);
63                      if (res != null && !res.isEmpty()) {
64                          result.setText(res.get(0));
65                      } else {
66                          result.setText("No speech detected");
67                      }
68                  } else {
69                      result.setText("Recognition error");
70                  }
71                  break;
72              }
73          }
74      }
75      @Override
76      public void onRequestPermissionsResult(int requestCode, @NonNull
           String[]
77              permissions, @NonNull int[] grantResults) {
78          super.onRequestPermissionsResult(requestCode, permissions,
79                  grantResults);
80          if (requestCode == REQUEST_RECORD_AUDIO_PERMISSION) {
81              if (grantResults.length > 0 && grantResults[0] ==
82                      PackageManager.PERMISSION_GRANTED) {
83                  startListening();
84              } else {
85                  Toast.makeText(this, "Permission denied",
86                          Toast.LENGTH_SHORT).show();
87              }
88          }
89      }
```

```
90 }
```

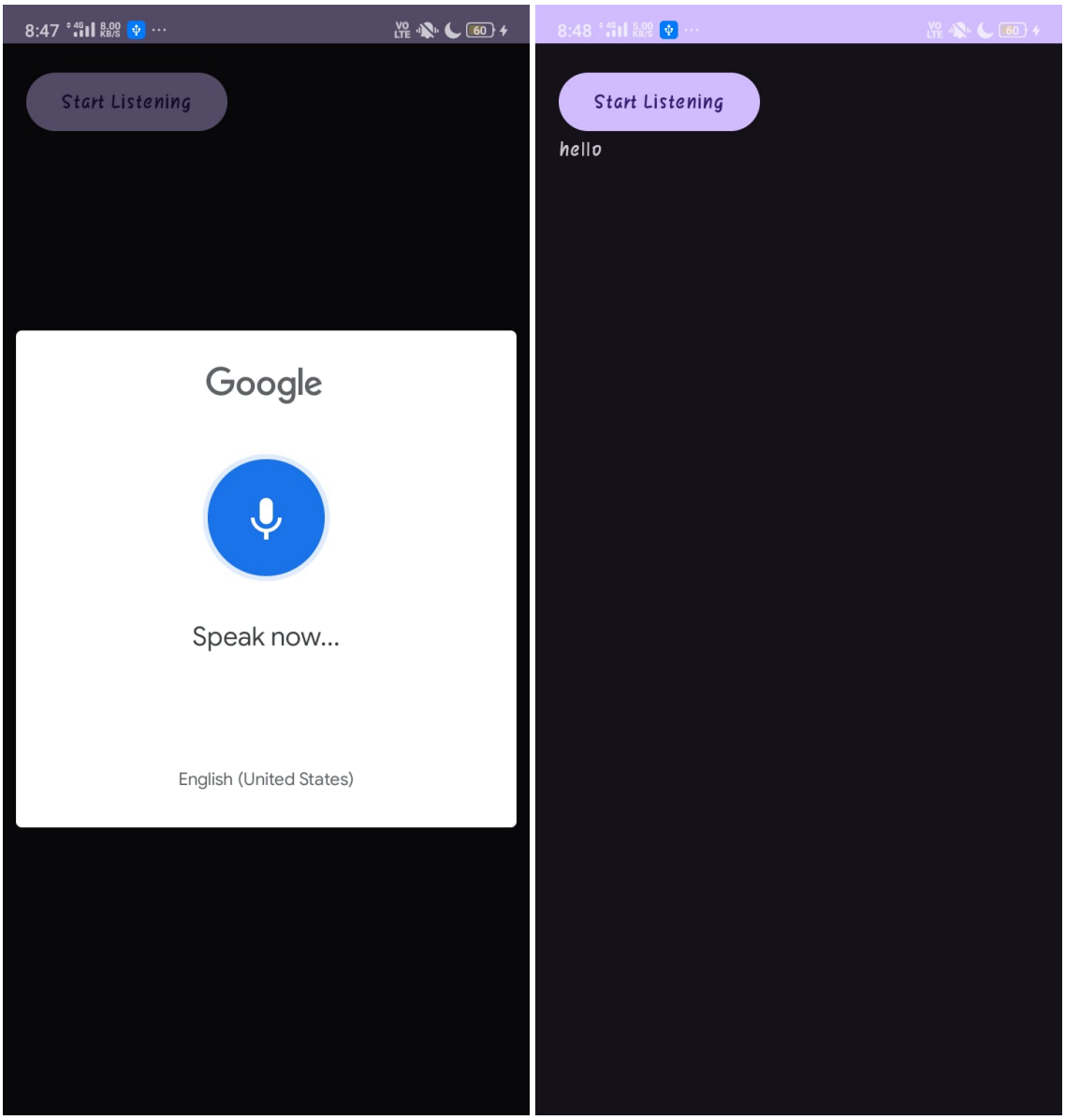
activity_main.xml

```
1  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/
       android"
2      android:layout_width="match_parent"
3      android:layout_height="match_parent"
4      android:orientation="vertical"
5      android:padding="16dp">
6      <Button
7          android:id="@+id/listen"
8          android:layout_width="wrap_content"
9          android:layout_height="wrap_content"
10         android:text="Start Listening" />
11     <TextView
12         android:id="@+id/result"
13         android:layout_width="match_parent"
14         android:layout_height="wrap_content"
15         android:text="" />
16 </LinearLayout>
```

Output



(a) Form Screen (b) Success Page



(a) Form Screen (b) Success Page

Q5. 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.

Solution

`MainActivity.java`

```
1 package com.darshan.mylocationapp;
2
```

```
import android.Manifest;
import android.annotation.SuppressLint;
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");
        }
        @SuppressLint("SetTextI18n")
        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(@NonNull GoogleMap googleMap) {
            map = googleMap;
```

```
91          Log.d("MainActivity", "Map is ready");
92      }
93      @Override
94      public void onRequestPermissionsResult(int requestCode, @NonNull
            String[]
95              permissions, @NonNull int[] grantResults) {
96          super.onRequestPermissionsResult(requestCode, permissions,
97                  grantResults);
98          if (requestCode == REQUEST_LOCATION_PERMISSION) {
99              if (grantResults.length > 0 && grantResults[0] ==
100                     PackageManager.PERMISSION_GRANTED) {
101                 getLocation();
102             } else {
103                 Toast.makeText(this, "Permission denied",
104                         Toast.LENGTH_SHORT).show();
105                 Log.d("MainActivity", "Permission denied");
106             }
107         }
108     }
109 }
```

---

activity_main.xml

```
1  <?xml version="1.0" encoding="utf-8" ?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/
       android"
3      xmlns:tools="http://schemas.android.com/tools"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      android:orientation="vertical"
7      android:padding="16dp">
8      <Button
9          android:id="@+id/refresh"
10         android:layout_width="wrap_content"
11         android:layout_height="wrap_content"
12         android:text="Refresh Location"
13         tools:ignore="HardcodedText" />
14     <TextView
15         android:id="@+id/res"
16         android:layout_width="match_parent"
17         android:layout_height="wrap_content"
18         android:text="Location: "
19         tools:ignore="HardcodedText" />
20
21     <androidx.fragment.app.FragmentContainerView
22         android:id="@+id/map"
23         android:layout_width="match_parent"
24         android:layout_height="match_parent"
25         android:name="com.google.android.gms.maps.SupportMapFragment"
```

```
                />
</LinearLayout>
```

Q6. 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.

Solution

`MainActivity.java`

```
1  package com.darshan.smspermissionsapp;
2
3  import android.Manifest;
4  import android.annotation.SuppressLint;
5  import android.content.pm.PackageManager;
6  import android.os.Bundle;
7  import android.telephony.SmsManager;
8  import android.view.View;
9  import android.widget.Button;
10 import android.widget.EditText;
11 import android.widget.TextView;
12 import androidx.annotation.NonNull;
13 import androidx.appcompat.app.AppCompatActivity;
14 import androidx.core.app.ActivityCompat;
15 import androidx.core.content.ContextCompat;
16 public class MainActivity extends AppCompatActivity {
17     private static final int REQUEST_SEND_SMS = 1;
18     private EditText num;
19     private EditText msg;
20     private TextView status;
21     @Override
22     protected void onCreate(Bundle savedInstanceState) {
23         super.onCreate(savedInstanceState);
24         setContentView(R.layout.activity_main);
25         num = findViewById(R.id.num);
26         msg = findViewById(R.id.msg);
27         status = findViewById(R.id.status);
28         Button send = findViewById(R.id.send);
29         send.setOnClickListener(new View.OnClickListener() {
30             @Override
31             public void onClick(View v) {
32                 if (ContextCompat.checkSelfPermission(MainActivity.
                       this,
33                            Manifest.permission.SEND_SMS) !=
                              PackageManager.PERMISSION_GRANTED) {
34                     ActivityCompat.requestPermissions(MainActivity.
                           this, new
```

```
                                String[]{Manifest.permission.SEND_SMS},
                                    REQUEST_SEND_SMS);
                } else {
                    sendSMS();
                }
            }
        });
    }
    @SuppressLint("SetTextI18n")
    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();
        }
    }
    @SuppressLint("SetTextI18n")
    @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.");
            }
        }
    }
}
```

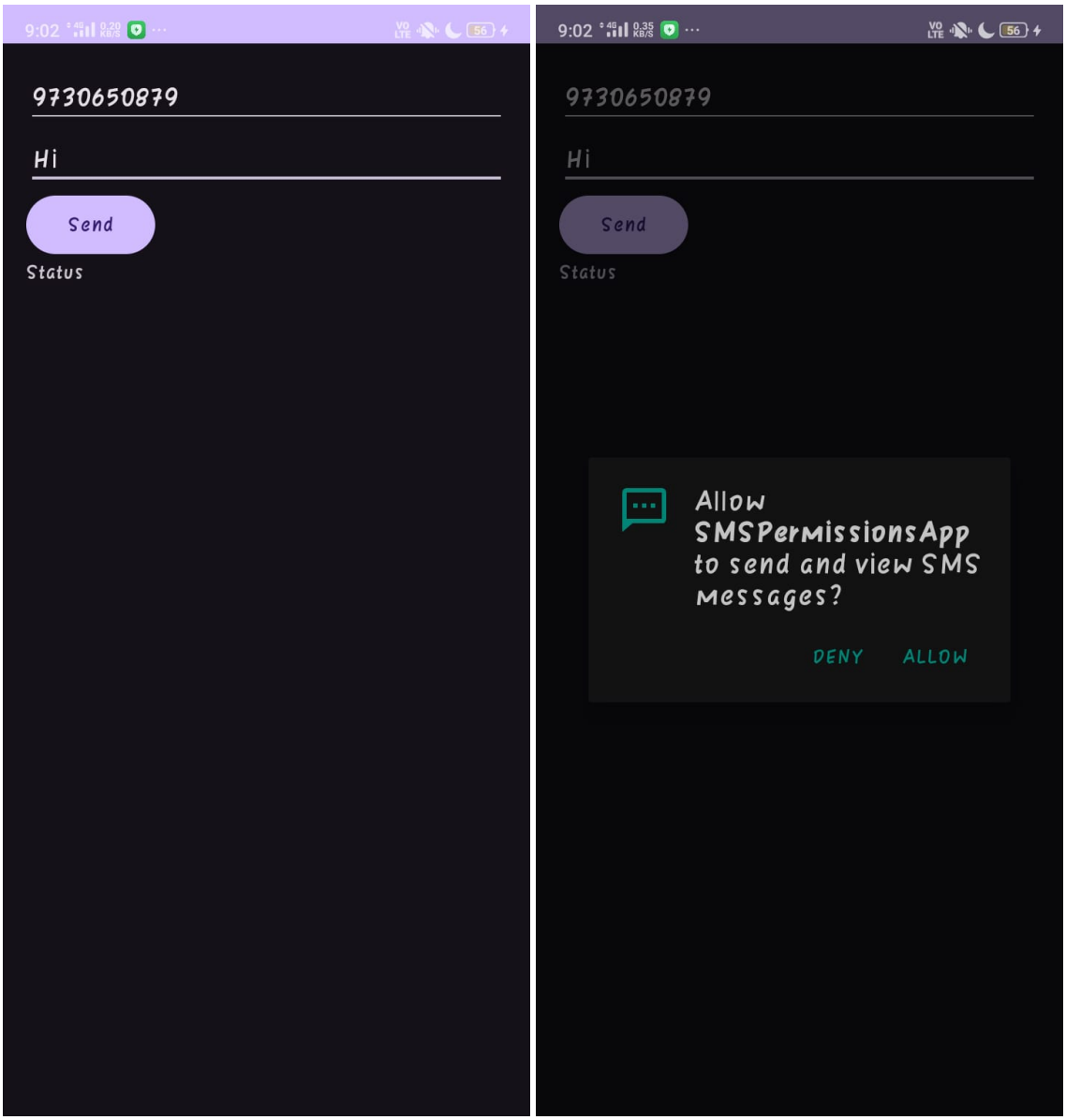
---

activity_main.xml

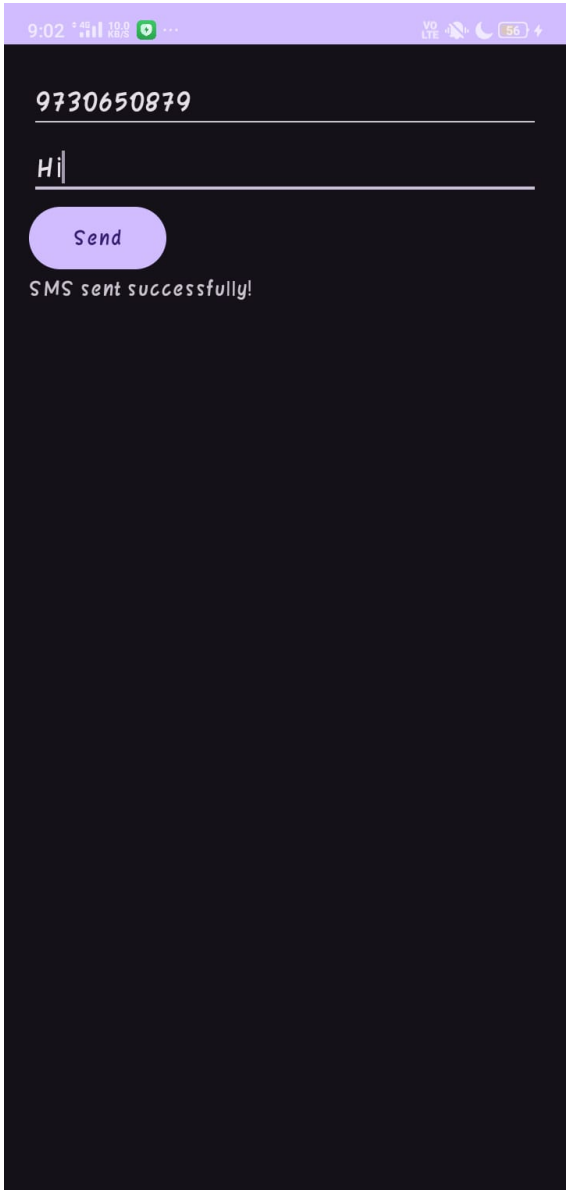
```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/
    android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">
```

```
    <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>
```

Output



(a) Main Screen (b) Image Notification



(a) Main Screen

Q7. 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.

Solution

`MainActivity.java`

```
1 package com.darshan.photosapp;
2
```

```
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,
```

```
48                                  Manifest.permission.
                                        WRITE_EXTERNAL_STORAGE},
                                        REQUEST_PERMISSIONS);
49                      } else {
50                          dispatchTakePictureIntent();
51                      }
52                  }
53              });
54      }
55      private void dispatchTakePictureIntent() {
56          Intent takePictureIntent = new Intent(MediaStore.
                ACTION_IMAGE_CAPTURE);
57          if (takePictureIntent.resolveActivity(getPackageManager()) !=
                null) {
58              File photoFile = null;
59              try {
60                  photoFile = createImageFile();
61              } catch (IOException ex) {
62                  Log.e("MainActivity", "Error occurred while creating
                        the file: " + ex.getMessage());
63              }
64              if (photoFile != null) {
65                  Uri photoURI = FileProvider.getUriForFile(this,
66                          "com.example.android.fileprovider", photoFile
                                );
67                  takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT,
                        photoURI);
68                  startActivityForResult(takePictureIntent,
69                          REQUEST_IMAGE_CAPTURE);
70              }
71          }
72      }
73      private File createImageFile() throws IOException {
74          String timeStamp = new SimpleDateFormat("yyyyMMdd_HHmmss",
75                  Locale.getDefault()).format(new Date());
76          String imageFileName = "JPEG_" + timeStamp + "_";
77          File storageDir = getExternalFilesDir(Environment.
                DIRECTORY_PICTURES);
78          File image = File.createTempFile(imageFileName, ".jpg",
                storageDir);
79          currentPhotoPath = image.getAbsolutePath();
80          return image;
81      }
82      @Override
83      protected void onActivityResult(int requestCode, int resultCode,
            Intent
84              data) {
85          super.onActivityResult(requestCode, resultCode, data);
86          if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode ==
                RESULT_OK) {
87              setPic();
```

```
88          }
89      }
90      private void setPic() {
91          int targetW = photo.getWidth();
92          int targetH = photo.getHeight();
93          BitmapFactory.Options bmOptions = new BitmapFactory.Options()
                ;
94          bmOptions.inJustDecodeBounds = true;
95          BitmapFactory.decodeFile(currentPhotoPath, bmOptions);
96          int photoW = bmOptions.outWidth;
97          int photoH = bmOptions.outHeight;
98          int scaleFactor = Math.min(photoW / targetW, photoH / targetH
                );
99          bmOptions.inJustDecodeBounds = false;
100         bmOptions.inSampleSize = scaleFactor;
101         Bitmap bitmap = BitmapFactory.decodeFile(currentPhotoPath,
                bmOptions);
102         photo.setImageBitmap(bitmap);
103     }
104     @Override
105     public void onRequestPermissionsResult(int requestCode, @NonNull
            String[]
106             permissions, @NonNull int[] grantResults) {
107         super.onRequestPermissionsResult(requestCode, permissions,
108                 grantResults);
109         if (requestCode == REQUEST_PERMISSIONS) {
110             if (grantResults.length > 0 && grantResults[0] ==
111                     PackageManager.PERMISSION_GRANTED) {
112                 dispatchTakePictureIntent();
113             } else {
114                 Toast.makeText(this, "Permission denied",
115                         Toast.LENGTH_SHORT).show();
116             }
117         }
118     }
119 }
```

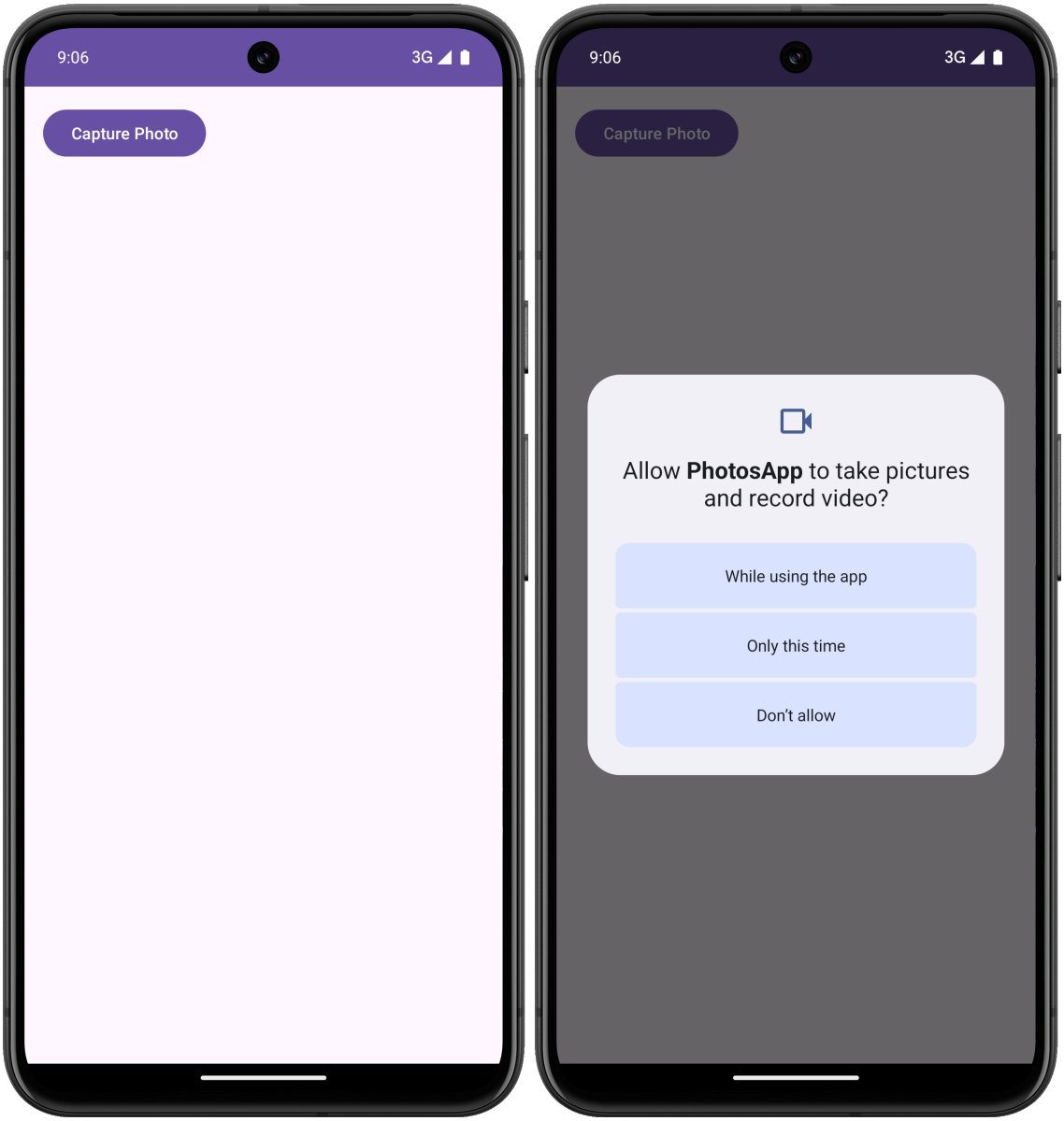
---

activity_main.xml

```
1  <?xml version="1.0" encoding="utf-8" ?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/
       android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      android:orientation="vertical"
6      android:padding="16dp">
7      <Button
8          android:id="@+id/capture"
9          android:layout_width="wrap_content"
```

```
10          android:layout_height="wrap_content"
11          android:text="Capture Photo" />
12      <ImageView
13          android:id="@+id/photo"
14          android:layout_width="match_parent"
15          android:layout_height="wrap_content"
16          android:scaleType="centerCrop" />
17  </LinearLayout>
```

Output



(a) Heads Up Notification (b) Heads Up Notification

Q8. 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.

Solution

`MainActivity.java`

```java
package com.darshan.calllog;

import android.Manifest;
import android.annotation.SuppressLint;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.telephony.PhoneStateListener;
import android.telephony.TelephonyManager;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {
    private static final int REQUEST_PERMISSION_CODE = 1;
    private final ArrayList<String> callLogs = new ArrayList<>();
    private CallLogAdapter adapter;

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

        RecyclerView recyclerView = findViewById(R.id.recyclerView);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));
        adapter = new CallLogAdapter(callLogs);
        recyclerView.setAdapter(adapter);

        if (ActivityCompat.checkSelfPermission(this, Manifest.
           permission.READ_PHONE_STATE) != PackageManager.
           PERMISSION_GRANTED ||
                ActivityCompat.checkSelfPermission(this, Manifest.
                   permission.READ_CALL_LOG) != PackageManager.
                   PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, new String[]{
                    Manifest.permission.READ_PHONE_STATE,
                    Manifest.permission.READ_CALL_LOG},
                       REQUEST_PERMISSION_CODE);
        } else {
            monitorCalls();
        }
    }

    private void monitorCalls() {
        TelephonyManager telephonyManager = (TelephonyManager)
           getSystemService(TELEPHONY_SERVICE);
        telephonyManager.listen(new PhoneStateListener() {
            private long startTime;
```

```
46
47              @SuppressLint("NotifyDataSetChanged")
48              @Override
49              public void onCallStateChanged(int state, String
                    phoneNumber) {
50                  if (state == TelephonyManager.CALL_STATE_RINGING) {
51                      callLogs.add("Incoming: " + phoneNumber);
52                      adapter.notifyDataSetChanged();
53                  } else if (state == TelephonyManager.
                        CALL_STATE_OFFHOOK) {
54                      startTime = System.currentTimeMillis();
55                  } else if (state == TelephonyManager.CALL_STATE_IDLE)
                        {
56                      long duration = (System.currentTimeMillis() -
                            startTime) / 1000;
57                      callLogs.add("Duration: " + duration + " sec");
58                      adapter.notifyDataSetChanged();
59                  }
60              }
61          }, PhoneStateListener.LISTEN_CALL_STATE);
62      }
63
64      @Override
65      public void onRequestPermissionsResult(int requestCode, @NonNull
            String[] permissions, @NonNull int[] grantResults) {
66          super.onRequestPermissionsResult(requestCode, permissions,
                grantResults);
67          if (requestCode == REQUEST_PERMISSION_CODE && grantResults.
                length > 0 &&
68                  grantResults[0] == PackageManager.PERMISSION_GRANTED
                        && grantResults[1] == PackageManager.
                        PERMISSION_GRANTED) {
69              monitorCalls();
70          } else {
71              Toast.makeText(this, "Permissions required", Toast.
                    LENGTH_SHORT).show();
72          }
73      }
74  }
```

CallAdapter.java

```
1  package com.darshan.calllog;
2
3  import android.view.LayoutInflater;
4  import android.view.View;
5  import android.view.ViewGroup;
6  import android.widget.TextView;
7  import androidx.annotation.NonNull;
```

```
import androidx.recyclerview.widget.RecyclerView;
import java.util.List;

public class CallLogAdapter extends RecyclerView.Adapter<
    CallLogAdapter.ViewHolder> {
     private List<String> callLogs;

     public CallLogAdapter(List<String> callLogs) {
         this.callLogs = callLogs;
     }

     @NonNull
     @Override
     public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent,
        int viewType) {
         View view = LayoutInflater.from(parent.getContext()).inflate(
            android.R.layout.simple_list_item_1, parent, false);
         return new ViewHolder(view);
     }

     @Override
     public void onBindViewHolder(@NonNull ViewHolder holder, int
        position) {
         holder.textView.setText(callLogs.get(position));
     }

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

     static class ViewHolder extends RecyclerView.ViewHolder {
         TextView textView;

         ViewHolder(@NonNull View itemView) {
             super(itemView);
             textView = itemView.findViewById(android.R.id.text1);
         }
     }
}
```

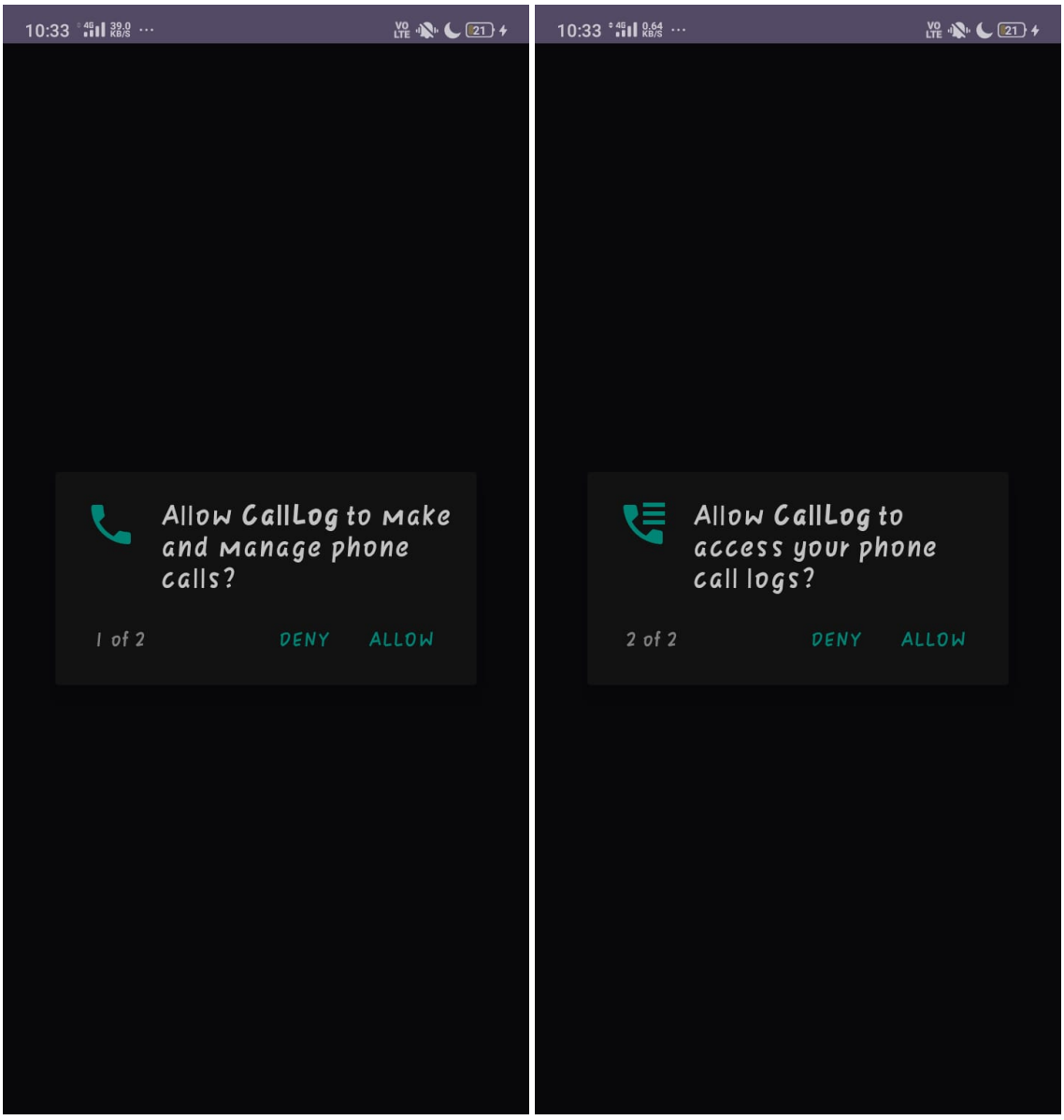
---

activity_main.xml

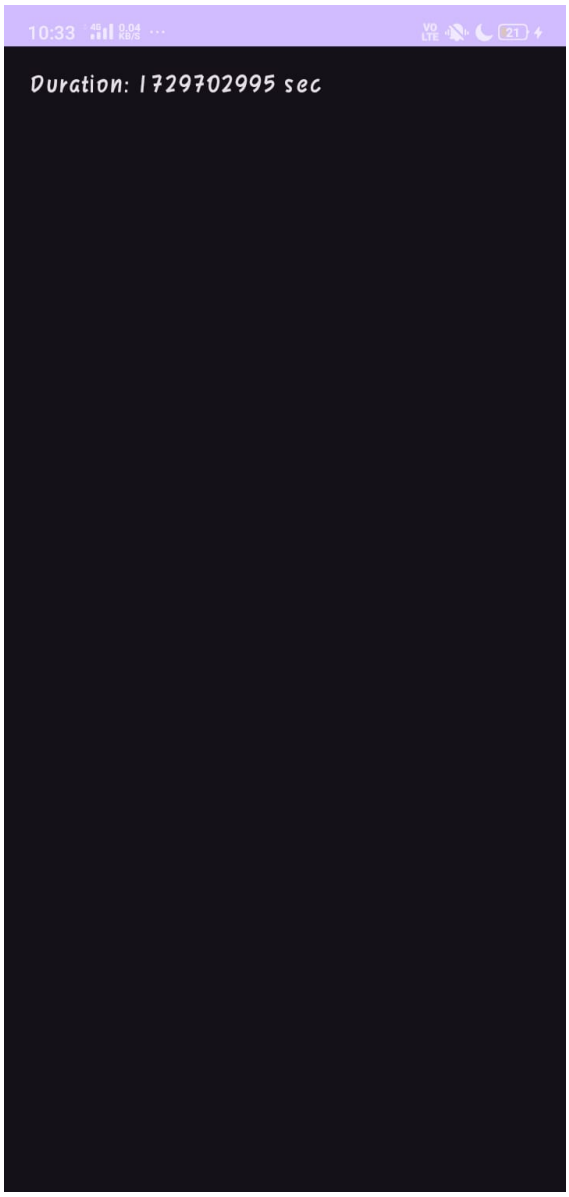
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/
    android"
     android:layout_width="match_parent"
     android:layout_height="match_parent"
     android:orientation="vertical">
```

```
6
7       <RecyclerView
8           android:id="@+id/recyclerView"
9           android:layout_width="match_parent"
10          android:layout_height="match_parent" />
11  </LinearLayout>
```

Output



(a) Main Screen (b) Options Menu



(a) Options Menu

Q9. 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.

Solution

`MainActivity.java`

```
1 package com.darshan.ttsapp;
2 import android.Manifest;
```

```
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.speech.RecognizerIntent;
import android.speech.tts.TextToSpeech;
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_CODE_SPEECH_INPUT = 1;
    private static final int REQUEST_CODE_AUDIO_PERMISSION = 2;
    private TextView recognizedText;
    private TextToSpeech textToSpeech;

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

        recognizedText = findViewById(R.id.textView);
        Button startRecognitionButton = findViewById(R.id.
           recognizeButton);
        Button speakTextButton = findViewById(R.id.speakButton);

        // Initialize Text-to-Speech
        textToSpeech = new TextToSpeech(this, status -> {
            if (status != TextToSpeech.ERROR) {
                textToSpeech.setLanguage(Locale.US);
            }
        });

        // Request audio permission if not granted
        if (ContextCompat.checkSelfPermission(this, Manifest.
           permission.RECORD_AUDIO)
                != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this,
                    new String[]{Manifest.permission.RECORD_AUDIO},
                    REQUEST_CODE_AUDIO_PERMISSION);
        }

        // Start Speech Recognition Button
        startRecognitionButton.setOnClickListener(v ->
           startSpeechRecognition());

```

```
51          // Text-to-Speech Button
52          speakTextButton.setOnClickListener(v -> {
53              String text = recognizedText.getText().toString();
54              if (!text.isEmpty()) {
55                  textToSpeech.speak(text, TextToSpeech.QUEUE_FLUSH,
                        null, null);
56              } else {
57                  Toast.makeText(MainActivity.this, "No text to speak",
                        Toast.LENGTH_SHORT).show();
58              }
59          });
60      }
61
62      // Method to start speech recognition
63      private void startSpeechRecognition() {
64          Intent intent = new Intent(RecognizerIntent.
                ACTION_RECOGNIZE_SPEECH);
65          intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
                RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
66          intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.
                getDefault());
67          intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Speak
                something...");
68
69          try {
70              startActivityForResult(intent, REQUEST_CODE_SPEECH_INPUT)
                    ;
71          } catch (Exception e) {
72              Toast.makeText(this, "Speech recognition is not supported
                    on this device", Toast.LENGTH_SHORT).show();
73          }
74      }
75
76      // Handle speech recognition result
77      @Override
78      protected void onActivityResult(int requestCode, int resultCode,
            Intent data) {
79          super.onActivityResult(requestCode, resultCode, data);
80          if (requestCode == REQUEST_CODE_SPEECH_INPUT && resultCode ==
                RESULT_OK && data != null) {
81              ArrayList<String> result = data.getStringArrayListExtra(
                    RecognizerIntent.EXTRA_RESULTS);
82              if (result != null && !result.isEmpty()) {
83                  recognizedText.setText(result.get(0));
84              }
85          }
86      }
87
88      @Override
89      protected void onDestroy() {
90          if (textToSpeech != null) {
```

```
            textToSpeech.shutdown();
        }
        super.onDestroy();
    }

    // Handle permission request result
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull
       String[] permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
           grantResults);
        if (requestCode == REQUEST_CODE_AUDIO_PERMISSION) {
            if (grantResults.length > 0 && grantResults[0] !=
               PackageManager.PERMISSION_GRANTED) {
                Toast.makeText(this, "Audio permission is required
                   for speech recognition", Toast.LENGTH_LONG).show()
                   ;
            }
        }
    }
}
```

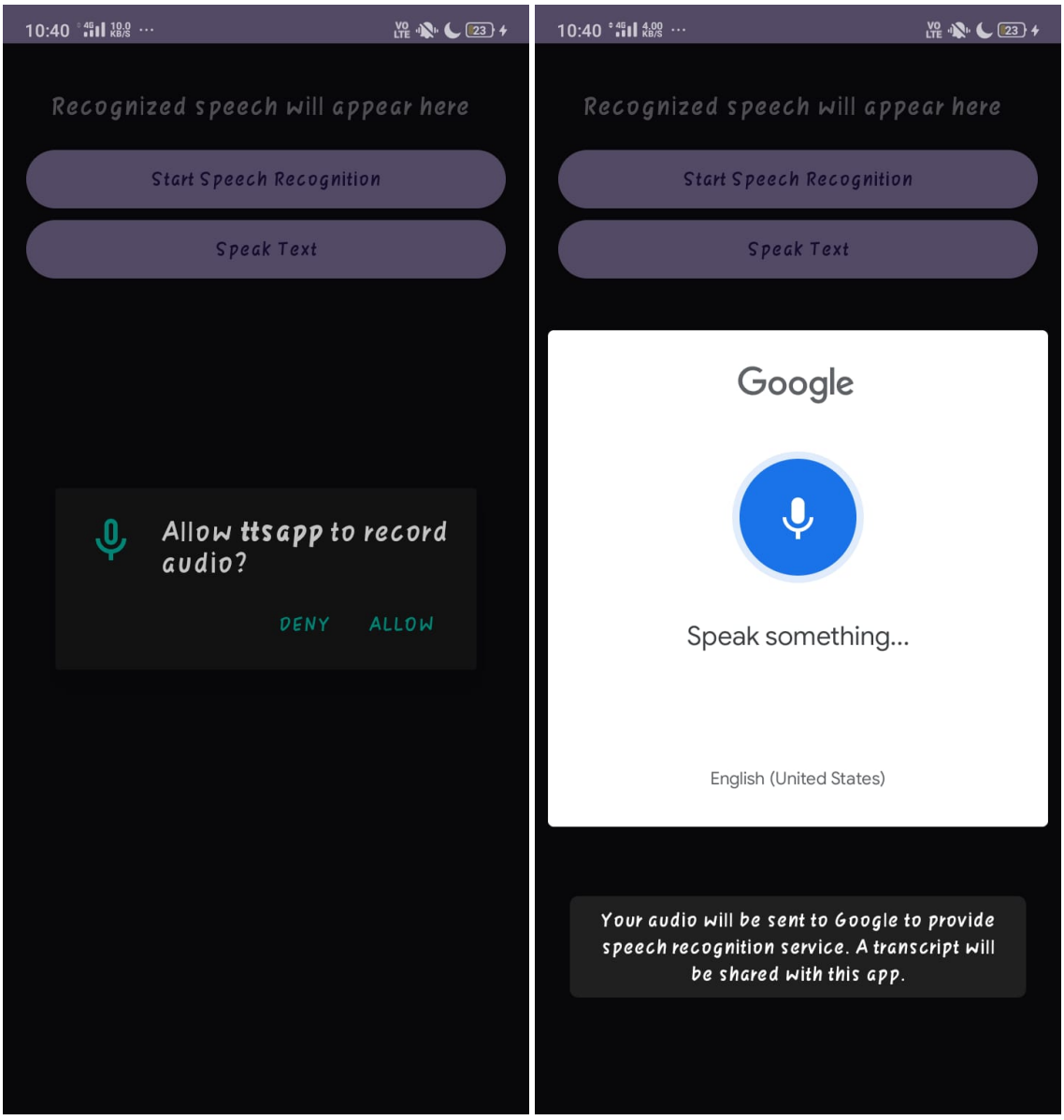
---

activity_main.xml

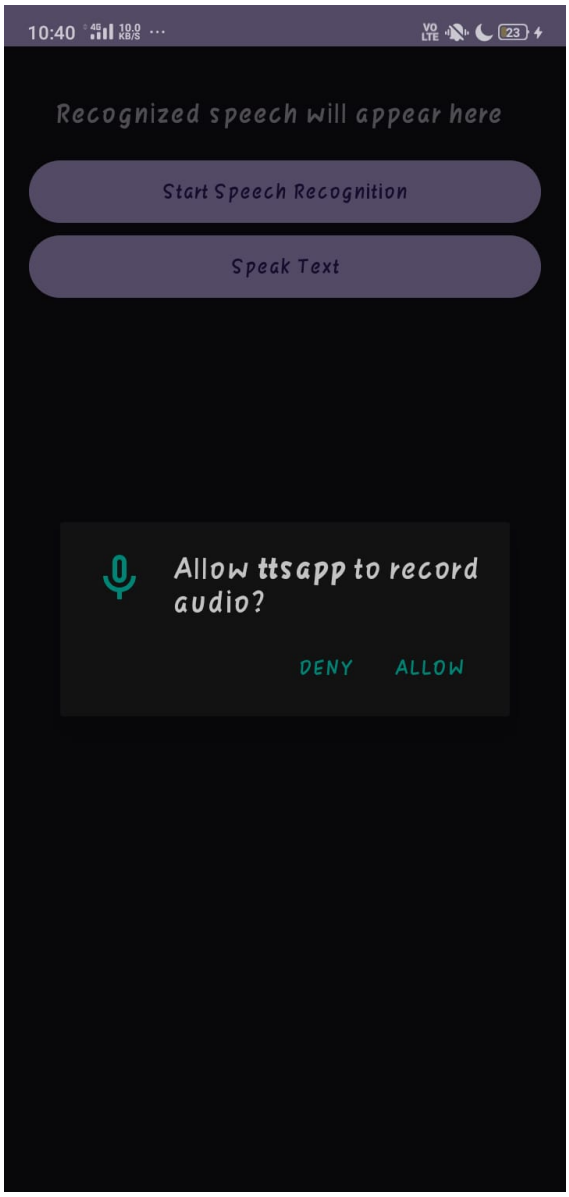
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/
   android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Recognized speech will appear here"
        android:textSize="18sp"
        android:padding="16dp"/>

    <Button
        android:id="@+id/recognizeButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Start Speech Recognition"/>

    <Button
        android:id="@+id/speakButton"
        android:layout_width="match_parent"
```

```
25          android:layout_height="wrap_content"
26          android:text="Speak Text"/>
27  </LinearLayout>
```

Output



(a) Main Screen (b) Success Page



(a) Main Screen

Q10. 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.

Solution

`MainActivity.java`

```
1 package com.darshan.tracker;
2
```

```
import android.Manifest;
import android.content.pm.PackageManager;
import android.location.Location;
import android.os.Bundle;
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_CODE_LOCATION_PERMISSION = 1;
    private FusedLocationProviderClient fusedLocationClient;
    private LocationCallback locationCallback;
    private Location startLocation;
    private TextView distanceTextView;

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

        distanceTextView = findViewById(R.id.distanceTextView);
        fusedLocationClient = LocationServices.
           getFusedLocationProviderClient(this);

        // Request location permission
        if (ContextCompat.checkSelfPermission(this, Manifest.
           permission.ACCESS_FINE_LOCATION)
                != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this,
                    new String[]{Manifest.permission.
                       ACCESS_FINE_LOCATION},
                    REQUEST_CODE_LOCATION_PERMISSION);
        } else {
            startLocationUpdates();
        }
    }

    // Start location updates
    private void startLocationUpdates() {
        LocationRequest locationRequest = LocationRequest.create();
        locationRequest.setInterval(5000); // 5 seconds interval
        locationRequest.setFastestInterval(2000); // 2 seconds
           fastest interval
```

```
50          locationRequest.setPriority(LocationRequest.
               PRIORITY_HIGH_ACCURACY);
51
52          locationCallback = new LocationCallback() {
53              @Override
54              public void onLocationResult(LocationResult
                   locationResult) {
55                  if (locationResult == null) return;
56                  Location currentLocation = locationResult.
                       getLastLocation();
57
58                  if (startLocation == null) {
59                      startLocation = currentLocation;
60                  }
61
62                  float distance = startLocation.distanceTo(
                       currentLocation);
63                  distanceTextView.setText("Distance: " + distance + "
                       meters");
64              }
65          };
66
67          fusedLocationClient.requestLocationUpdates(locationRequest,
               locationCallback, null);
68      }
69
70      // Handle location permission result
71      @Override
72      public void onRequestPermissionsResult(int requestCode, @NonNull
           String[] permissions, @NonNull int[] grantResults) {
73          super.onRequestPermissionsResult(requestCode, permissions,
               grantResults);
74          if (requestCode == REQUEST_CODE_LOCATION_PERMISSION) {
75              if (grantResults.length > 0 && grantResults[0] ==
                   PackageManager.PERMISSION_GRANTED) {
76                  startLocationUpdates();
77              } else {
78                  Toast.makeText(this, "Location permission is required
                       to track distance", Toast.LENGTH_LONG).show();
79              }
80          }
81      }
82
83      @Override
84      protected void onDestroy() {
85          if (fusedLocationClient != null && locationCallback != null)
               {
86              fusedLocationClient.removeLocationUpdates(
                   locationCallback);
87          }
88          super.onDestroy();
```

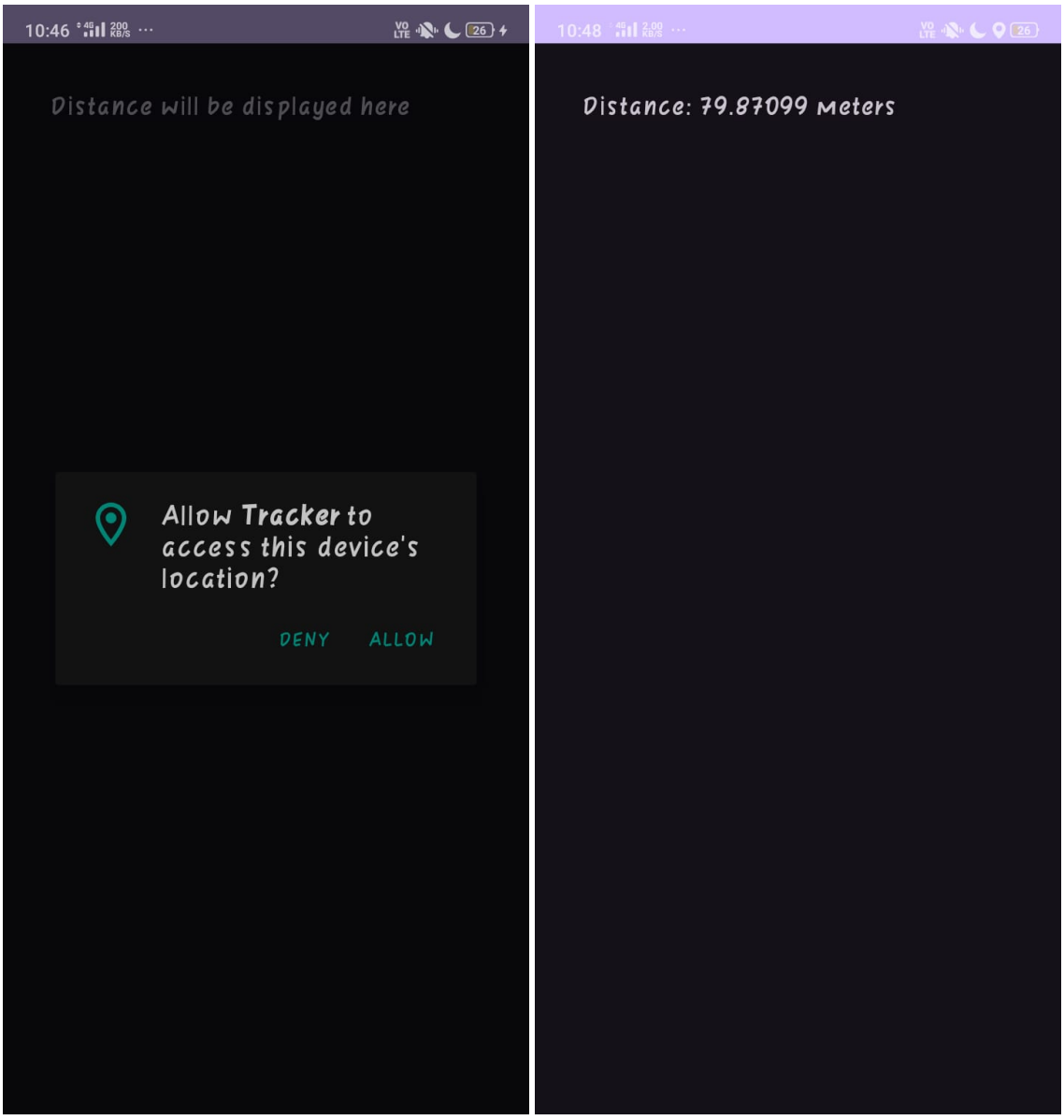
```
89      }
90  }
```

---

activity_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/
       android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      android:orientation="vertical"
6      android:padding="16dp">
7
8      <TextView
9          android:id="@+id/distanceTextView"
10         android:layout_width="match_parent"
11         android:layout_height="wrap_content"
12         android:text="Distance will be displayed here"
13         android:textSize="18sp"
14         android:padding="16dp"/>
15
16 </LinearLayout>
```

Output



(a) Main Screen (b) Form Page