

SSN COLLEGE OF ENGINEERING, KALAVAKKAM
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UCS1711 - MOBILE APPLICATION DEVELOPMENT LAB

Assignment 1

Name: Jayannthan P T

Dept: CSE 'A'

Roll No.: 205001049

Generate a Health Insurance registration form to register the patient details under each group.

Patient Details

- a. Assign a title for the registration form(TextView- textSize, textStyle , typeface)

Group1 Patient Details

- b. Patient Name. Specify some font and colour. (use TextView, EditText-standard)

- c. Patient Phone Number (Mobile or Landline use Checkbox)

- d. Address (use TextView, EditText-standard)

- e. Age (TextView, EditText)

- f. Date of Birth (DatePicker)

- g. Gender (RadioButton)

- h. Marital Status (Spinner)

Employer Details

- a) Patient Employer

- b) Employment Status (full time, parttime, unemployed, retired, student, other Checkbox)

Emergency contact Details

- a) Name (EditText)

- b) Relationship (EditText)

- c) Address (Textarea)

- d) Phone Number (EditText- inputType, phoneNumber)

Use Submit (Button) to submit the details and display the contents. Use Reset button to clear the form.

Additional: Display using Table layout create an output window using OPENGL and to draw the following basic output primitives:

Title of the Program: Generate a Health Insurance registration form to register the patient details under each group

Objective:

The objective of the Health Insurance Android App project is to create a user-friendly application that allows users to input their personal details, employment information, and emergency contact details. The entered data is then displayed in a well-organized manner for the user's review.

Algorithm:

1. Create the main activity layout (**activity_main.xml**) with input fields for patient details, employment information, and emergency contact details.
2. Implement the logic to retrieve user input from the main activity, including handling radio buttons, checkboxes, and date picker.
3. Design the display data activity layout (**activity_display_data.xml**) using a **ScrollView**, **TableLayout**, and **TableRow** structure to display the entered data in a tabular format.
4. Pass the collected data from the main activity to the display data activity using an intent with extras.
5. In the display data activity, retrieve the data from the intent and populate the appropriate **TextView** elements in the layout.

Features used:

1. **EditText** widgets for user input.
2. **RadioGroup** and **RadioButton** widgets for gender selection.
3. **Spinner** widget for selecting marital status.
4. **CheckBox** widgets for selecting multiple options (employment status).
5. **DatePicker** widget for selecting the date of birth.
6. Passing data between activities using intents with extras.

Source code:

- `MainActivity.java`

```
package com.example.healthinsurance;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Spinner;

import java.text.SimpleDateFormat;
```

```
import java.util.Calendar;

public class MainActivity extends AppCompatActivity {

    // Declare your view elements
    EditText patientNameEditText, addressEditText, ageEditText, employerEditText,
        emergencyNameEditText, relationshipEditText, emergencyAddressEditText,
        emergencyPhoneNumberEditText;

    RadioGroup genderRadioGroup;
    RadioButton maleRadioButton, femaleRadioButton, otherRadioButton;

    Spinner maritalStatusSpinner;

    CheckBox mobileCheckBox, landlineCheckBox, fullTimeCheckBox, partTimeCheckBox;

    DatePicker dateOfBirthDatePicker;

    Button submitButton, resetButton;

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

        // Initialize your view elements
        patientNameEditText = findViewById(R.id.patientNameEditText);
        addressEditText = findViewById(R.id.addressEditText);
        ageEditText = findViewById(R.id.ageEditText);
        employerEditText = findViewById(R.id.employerEditText);
        emergencyNameEditText = findViewById(R.id.emergencyNameEditText);
        relationshipEditText = findViewById(R.id.relationshipEditText);
        emergencyAddressEditText = findViewById(R.id.emergencyAddressEditText);
        emergencyPhoneNumberEditText = findViewById(R.id.emergencyPhoneNumberEditText);

        genderRadioGroup = findViewById(R.id.genderRadioGroup);
        maleRadioButton = findViewById(R.id.maleRadioButton);
        femaleRadioButton = findViewById(R.id.femaleRadioButton);
        otherRadioButton = findViewById(R.id.otherRadioButton);

        mobileCheckBox = findViewById(R.id.mobileCheckBox);
        landlineCheckBox = findViewById(R.id.landlineCheckBox);
        fullTimeCheckBox = findViewById(R.id.fullTimeCheckBox);
        partTimeCheckBox = findViewById(R.id.partTimeCheckBox);

        dateOfBirthDatePicker = findViewById(R.id.dateOfBirthDatePicker);

        submitButton = findViewById(R.id.submitButton);
        resetButton = findViewById(R.id.resetButton);

        // Outside the OnClickListener block
        Spinner maritalStatusSpinner = findViewById(R.id.displayMaritalStatusSpinner);
        ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(
            R.array.marital_status_array, R.layout.spinner_item)
```

```
        MainActivity.this, R.array.marital_status_options,
        android.R.layout.simple_spinner_item);
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    maritalStatusSpinner.setAdapter(adapter);

    // Inside the OnClickListener block
    submitButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String patientName = patientNameEditText.getText().toString();
            String address = addressEditText.getText().toString();
            String ageStr = ageEditText.getText().toString();
            int age = 0;
            if (!ageStr.isEmpty()) {
                age = Integer.parseInt(ageStr);
            }
            String employer = employerEditText.getText().toString();
            String employmentStatus = "";
            if (fullTimeCheckBox.isChecked()) {
                employmentStatus = "Full Time";
            } else if (partTimeCheckBox.isChecked()) {
                employmentStatus = "Part Time";
            }

            String maritalStatus = maritalStatusSpinner.getSelectedItem().toString();

            String emergencyContactName = emergencyNameEditText.getText().toString();
            String relationship = relationshipEditText.getText().toString();
            String emergencyContactAddress =
emergencyAddressEditText.getText().toString();
            String emergencyContactPhoneNumber =
emergencyPhoneNumberEditText.getText().toString();

            // Get selected radio button from genderRadioGroup
            int selectedGenderId = genderRadioGroup.getCheckedRadioButtonId();
            RadioButton selectedGenderRadioButton = findViewById(selectedGenderId);
            String gender = selectedGenderRadioButton.getText().toString();

            // Get selected date from dateOfBirthDatePicker
            int year = dateOfBirthDatePicker.getYear();
            int month = dateOfBirthDatePicker.getMonth() + 1; // DatePicker month is
0-based
            int day = dateOfBirthDatePicker.getDayOfMonth();
            String dateOfBirth = year + "-" + month + "-" + day;

            Intent intent = new Intent(MainActivity.this, DisplayDataActivity.class);

            // Pass data to the intent using extras
            intent.putExtra("patientName", patientName);
            intent.putExtra("address", address);
            intent.putExtra("age", age);
            intent.putExtra("employer", employer);
            intent.putExtra("employmentStatus", employmentStatus);
            intent.putExtra("emergencyContactName", emergencyContactName);
```

```
        intent.putExtra("relationship", relationship);
        intent.putExtra("emergencyContactAddress", emergencyContactAddress);
        intent.putExtra("emergencyContactPhoneNumber",
emergencyContactPhoneNumber);
        intent.putExtra("gender", gender);
        intent.putExtra("maritalStatus", maritalStatus);
        intent.putExtra("dateOfBirth", dateOfBirth);

        // Start the new activity
        startActivity(intent);
    }
});

resetButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // Clear all input fields
        patientNameEditText.getText().clear();
        addressEditText.getText().clear();
        ageEditText.getText().clear();
        employerEditText.getText().clear();
        emergencyNameEditText.getText().clear();
        relationshipEditText.getText().clear();
        emergencyAddressEditText.getText().clear();
        emergencyPhoneNumberEditText.getText().clear();
        genderRadioGroup.clearCheck();
        maritalStatusSpinner.setSelection(0); // Reset to the first item in the
spinner
        mobileCheckBox.setChecked(false);
        landlineCheckBox.setChecked(false);
        fullTimeCheckBox.setChecked(false);
        partTimeCheckBox.setChecked(false);

        // Reset the date picker to today's date
        Calendar calendar = Calendar.getInstance();
        int year = calendar.get(Calendar.YEAR);
        int month = calendar.get(Calendar.MONTH);
        int day = calendar.get(Calendar.DAY_OF_MONTH);
        dateOfBirthDatePicker.updateDate(year, month, day);
    }
});

// Set the default date of birth in the date picker to today's date
Calendar calendar = Calendar.getInstance();
SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
String currentDate = dateFormat.format(calendar.getTime());
String[] dateParts = currentDate.split("-");
int year = Integer.parseInt(dateParts[0]);
int month = Integer.parseInt(dateParts[1]) - 1; // Month is 0-based
int day = Integer.parseInt(dateParts[2]);
dateOfBirthDatePicker.init(year, month, day, null);
}
})
```

- activity_main.xml

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

<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent" android:layout_height="wrap_content">

    <LinearLayout android:layout_width="match_parent" android:layout_height="wrap_content"
        android:orientation="vertical" android:padding="16dp">

        <TextView android:layout_width="wrap_content" android:layout_height="wrap_content"
            android:text="@string/health_insurance_registration_form" android:textSize="24sp"
            android:textStyle="bold" android:typeface="serif" android:layout_gravity="center"
            android:layout_marginBottom="16dp"/>

        <TableLayout android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <!-- Group: Patient Details -->
            <TableRow>
                <TextView android:layout_width="wrap_content"
                    android:layout_height="wrap_content" android:text="Patient Details"
                    android:textStyle="bold" android:layout_span="2" android:paddingBottom="8dp"
                    android:paddingTop="16dp"/>
                </TableRow>

                <TableRow>
                    <TextView android:layout_width="wrap_content"
                        android:layout_height="wrap_content" android:text="Patient Name:"
                        android:textColor="#000000" android:paddingEnd="8dp"/>
                    <EditText android:id="@+id/patientNameEditText" android:layout_width="0dp"
                        android:layout_height="wrap_content" android:hint="Enter patient name"
                        android:layout_weight="1"/>
                </TableRow>

                <TableRow>
                    <TextView android:layout_width="wrap_content"
                        android:layout_height="wrap_content" android:text="Patient Phone Number:"
                        android:textColor="#000000" android:paddingEnd="8dp"/>
                    <CheckBox android:id="@+id/mobileCheckBox"
                        android:layout_width="wrap_content" android:layout_height="wrap_content"
                        android:text="Mobile"/>
                    <CheckBox android:id="@+id/landlineCheckBox"
                        android:layout_width="wrap_content" android:layout_height="wrap_content"
                        android:text="Landline"/>
                </TableRow>

                <TableRow>
                    <TextView android:layout_width="wrap_content"
                        android:layout_height="wrap_content" android:text="Address:"
                        android:textColor="#000000" android:paddingEnd="8dp"/>
                </TableRow>
            
```

```
<EditText android:id="@+id/addressEditText" android:layout_width="0dp"
    android:layout_height="wrap_content" android:hint="Enter address"
    android:layout_weight="1"/>
</TableRow>

<TableRow>
    <TextView android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="Age:" android:textColor="#000000"
        android:paddingEnd="8dp"/>
    <EditText android:id="@+id/ageEditText" android:layout_width="0dp"
        android:layout_height="wrap_content" android:hint="Enter age" android:inputType="number"
        android:layout_weight="1"/>
</TableRow>

<TableRow>
    <TextView android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="Date of Birth:"
        android:textColor="#000000" android:paddingEnd="8dp"/>
    <DatePicker android:id="@+id/dateOfBirthDatePicker"
        android:layout_width="0dp" android:layout_height="wrap_content"
        android:layout_weight="1"/>
</TableRow>

<TableRow>
    <TextView android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="Gender:" android:textColor="#000000"
        android:paddingEnd="8dp"/>
    <RadioGroup android:id="@+id/genderRadioGroup" android:layout_width="0dp"
        android:layout_height="wrap_content" android:orientation="horizontal"
        android:layout_weight="1">
        <RadioButton android:id="@+id/maleRadioButton"
            android:layout_width="wrap_content" android:layout_height="wrap_content"
            android:text="Male"/>
        <RadioButton android:id="@+id/femaleRadioButton"
            android:layout_width="wrap_content" android:layout_height="wrap_content"
            android:text="Female"/>
        <RadioButton android:id="@+id/otherRadioButton"
            android:layout_width="wrap_content" android:layout_height="wrap_content"
            android:text="Other"/>
    </RadioGroup>
</TableRow>

<TableRow>
    <TextView android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="Marital Status:"
        android:textColor="#000000" android:paddingEnd="8dp"/>
    <Spinner android:id="@+id/displayMaritalStatusSpinner"
        android:layout_width="0dp" android:layout_height="wrap_content"
        android:layout_weight="1"/>
</TableRow>

<!-- Group: Employer Details -->
<TableRow>
```

```
<TextView android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:text="Employer Details"
    android:textStyle="bold" android:layout_span="2" android:paddingBottom="8dp"
    android:paddingTop="16dp"/>
</TableRow>

<TableRow>
    <TextView android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="Patient Employer:"
        android:textColor="#000000" android:paddingEnd="8dp"/>
        <EditText android:id="@+id/employerEditText" android:layout_width="0dp"
        android:layout_height="wrap_content" android:hint="Enter employer"
        android:layout_weight="1"/>
</TableRow>

<TableRow>
    <TextView android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="Employment Status:"
        android:textColor="#000000" android:paddingEnd="8dp"/>
        <CheckBox android:id="@+id/fullTimeCheckBox"
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:text="Full Time"/>
        <CheckBox android:id="@+id/partTimeCheckBox"
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:text="Part Time"/>
        <!-- Add more checkboxes for other options -->
</TableRow>

<!-- Group: Emergency Contact Details -->
<TableRow>
    <TextView android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="Emergency Contact Details"
        android:textStyle="bold" android:layout_span="2" android:paddingBottom="8dp"
        android:paddingTop="16dp"/>
</TableRow>

<TableRow>
    <TextView android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="Name:" android:textColor="#000000"
        android:paddingEnd="8dp"/>
        <EditText android:id="@+id/emergencyNameEditText"
        android:layout_width="0dp" android:layout_height="wrap_content" android:hint="Enter
        emergency contact name" android:layout_weight="1"/>
</TableRow>

<TableRow>
    <TextView android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="Relationship:"
        android:textColor="#000000" android:paddingEnd="8dp"/>
        <EditText android:id="@+id/relationshipEditText"
        android:layout_width="0dp" android:layout_height="wrap_content" android:hint="Enter
        relationship" android:layout_weight="1"/>
</TableRow>
```

```

<TableRow>
    <TextView android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:text="Address:" android:textColor="#000000"
    android:paddingEnd="8dp"/>
        <EditText android:id="@+id/emergencyAddressEditText"
    android:layout_width="0dp" android:layout_height="wrap_content" android:hint="Enter
    emergency contact address" android:layout_weight="1"/>
    </TableRow>

    <TableRow>
        <TextView android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:text="Phone Number:"
    android:textColor="#000000" android:paddingEnd="8dp"/>
            <EditText android:id="@+id/emergencyPhoneNumberEditText"
    android:layout_width="0dp" android:layout_height="wrap_content" android:hint="Enter
    emergency contact phone number" android:inputType="phone" android:layout_weight="1"/>
    </TableRow>

    <TableRow>
        <Button android:id="@+id/submitButton" android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:text="Submit" android:layout_gravity="center"
    android:layout_span="2" android:layout_marginTop="16dp"/>
    </TableRow>

    <TableRow>
        <Button android:id="@+id/resetButton" android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:text="Reset" android:layout_gravity="center"
    android:layout_span="2" android:layout_marginTop="8dp"/>
    </TableRow>
    </TableLayout>
</LinearLayout>
</ScrollView>

```

- DisplayDataActivity.java

```

package com.example.healthinsurance;

import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class DisplayDataActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_display_data);

        // Retrieve data from the intent
        Intent intent = getIntent();
        String patientName = intent.getStringExtra("patientName");
        String address = intent.getStringExtra("address");
    }
}

```

```
int age = intent.getIntExtra("age", 0);
String employer = intent.getStringExtra("employer");
String employmentStatus = intent.getStringExtra("employmentStatus");
String emergencyContactName = intent.getStringExtra("emergencyContactName");
String relationship = intent.getStringExtra("relationship");
String emergencyContactAddress = intent.getStringExtra("emergencyContactAddress");
String emergencyContactPhoneNumber =
intent.getStringExtra("emergencyContactPhoneNumber");
String gender = intent.getStringExtra("gender");
String maritalStatus = intent.getStringExtra("maritalStatus");
String dateOfBirth = intent.getStringExtra("dateOfBirth");

// Find TextViews in the layout and set their text
TextView displayPatientName = findViewById(R.id.displayPatientName);
TextView displayAddress = findViewById(R.id.displayAddress);
TextView displayAge = findViewById(R.id.displayAge);
TextView displayEmployer = findViewById(R.id.displayEmployer);
TextView displayEmploymentStatus = findViewById(R.id.displayEmploymentStatus);
TextView displayEmergencyContactName =
findViewById(R.id.displayEmergencyContactName);
TextView displayRelationship = findViewById(R.id.displayRelationship);
TextView displayEmergencyContactAddress =
findViewById(R.id.displayEmergencyContactAddress);
TextView displayEmergencyContactPhoneNumber =
findViewById(R.id.displayEmergencyContactPhoneNumber);
TextView displayGender = findViewById(R.id.displayGender);
TextView displayMaritalStatus = findViewById(R.id.displayMaritalStatus);
TextView displayDateOfBirth = findViewById(R.id.displayDateOfBirth);

displayPatientName.setText("Patient Name: " + patientName);
displayAddress.setText("Address: " + address);
displayAge.setText("Age: " + age);
displayEmployer.setText("Employer: " + employer);
displayEmploymentStatus.setText("Employment Status: " + employmentStatus);
displayEmergencyContactName.setText("Emergency Contact Name: " +
emergencyContactName);
displayRelationship.setText("Relationship: " + relationship);
displayEmergencyContactAddress.setText("Emergency Contact Address: " +
emergencyContactAddress);
displayEmergencyContactPhoneNumber.setText("Emergency Contact Phone: " +
emergencyContactPhoneNumber);
displayGender.setText("Gender: " + gender);
displayMaritalStatus.setText("Marital Status: " + maritalStatus);
displayDateOfBirth.setText("Date of Birth: " + dateOfBirth);
}

}
```

- activity_display_data.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">

    <TextView android:id="@+id/displayPatientName" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="" android:paddingBottom="8dp"/>

    <TextView android:id="@+id/displayAddress" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="" android:paddingBottom="8dp"/>

    <TextView android:id="@+id/displayAge" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="" android:paddingBottom="16dp"/>

    <TextView android:id="@+id/displayGender" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="" android:paddingBottom="8dp"/>

    <TextView android:id="@+id/displayMaritalStatus" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="" android:paddingBottom="8dp"/>

    <TextView android:id="@+id/displayDateOfBirth" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="" android:paddingBottom="16dp"/>

    <TextView android:id="@+id/displayEmployer" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="" android:paddingBottom="8dp"/>

    <TextView android:id="@+id/displayEmploymentStatus"
        android:layout_width="wrap_content" android:layout_height="wrap_content" android:text=""
        android:paddingBottom="8dp"/>

    <TextView android:id="@+id/displayEmergencyContactName"
        android:layout_width="wrap_content" android:layout_height="wrap_content" android:text=""
        android:paddingBottom="8dp"/>

    <TextView android:id="@+id/displayRelationship" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="" android:paddingBottom="8dp"/>

    <TextView android:id="@+id/displayEmergencyContactAddress"
        android:layout_width="wrap_content" android:layout_height="wrap_content" android:text=""
        android:paddingBottom="8dp"/>

    <TextView android:id="@+id/displayEmergencyContactPhoneNumber"
        android:layout_width="wrap_content" android:layout_height="wrap_content" android:text=""
        android:paddingBottom="16dp"/>

</LinearLayout>
```

Output:

A screenshot of a smartphone displaying a mobile application for health insurance. The app has a pink header bar with the title "Health Insurance". The status bar shows the time as 11:49 and battery level. Below the header, there is a date picker showing days 27, 28, 29, 30, and 31. The main form area contains the following fields:

Gender: Male Female Other

Marital Status: (dropdown menu)

Employer Details

Patient Employer:

Employment Status: Full Time Part Time

Emergency Contact Details

Name:

Relationship:

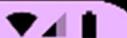
Address:

Phone Number:

SUBMIT **RESET**

The bottom of the screen shows a navigation bar with back, home, and recent apps icons.

11:48



Health Insurance

Health Insurance Registration Form

Patient Details

Patient Name:

Patient Phone Number: Mobile Landline

Address:

Age:

Date of Birth:

2023
Thu, Aug
24

< August 2023 >

S M T W T F S

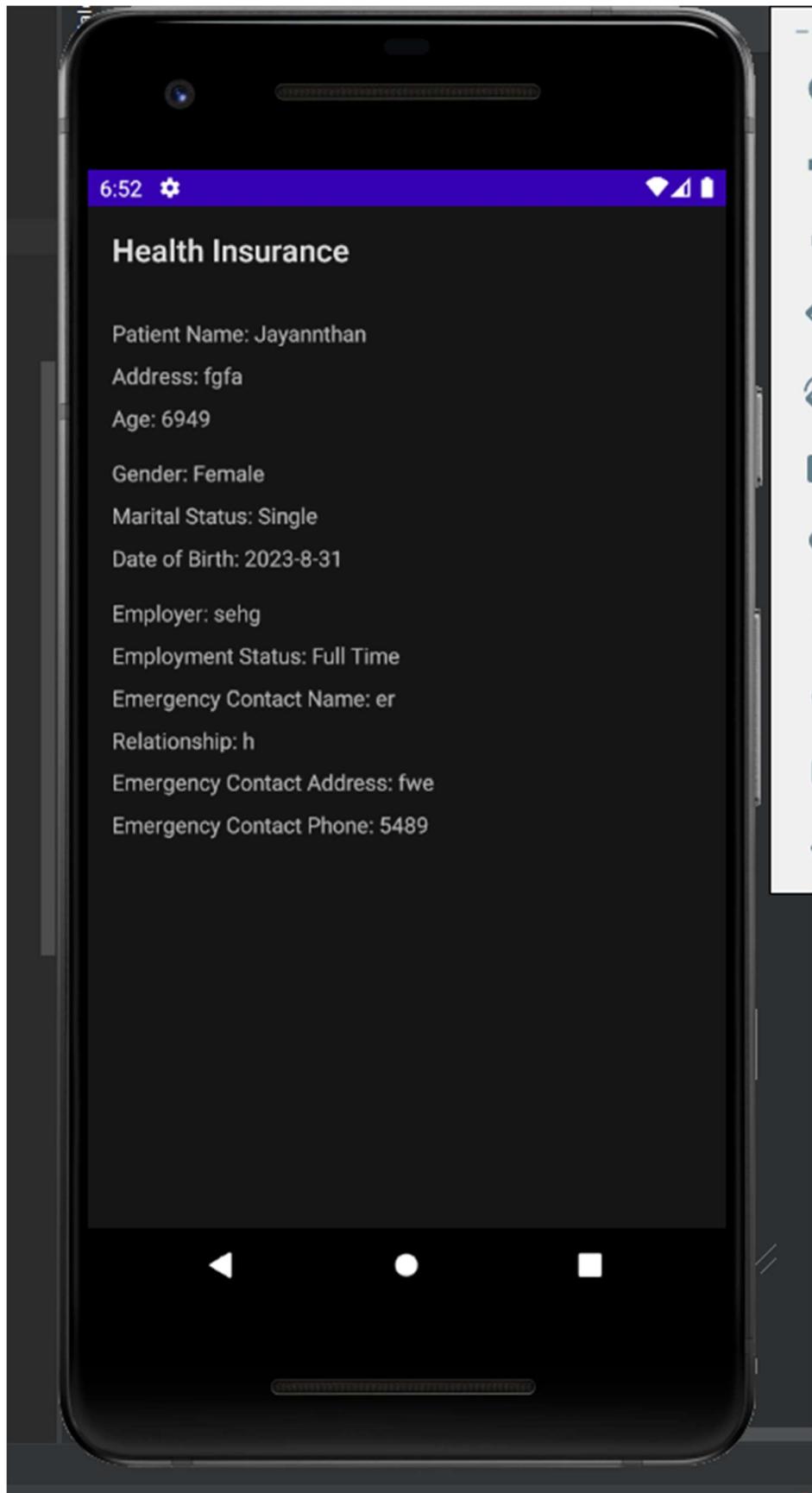
1 2 3 4 5

6 7 8 9 10 11 12

13 14 15 16 17 18 19

20 21 22 23 24 25 26





Result:

The mobile application was completed successfully

Best Practices:

1. Use appropriate variable naming conventions for readability.
2. Implement input validation to ensure data accuracy.
3. Utilize layout resources for UI consistency and responsiveness.
4. Organize the code with comments and logical structure.
5. Implement proper error handling to prevent crashes.

Learning Outcomes:

1. Designing Android user interfaces using XML layouts.
2. Data passing between activities.
3. Handling diverse user input elements (text fields, radio buttons, checkboxes, spinners, date pickers).
4. Creating organized layouts with ScrollView, TableLayout, and TableRow.
5. Enhancing Java coding skills for Android development.
6. Managing user interactions and events.

SSN COLLEGE OF ENGINEERING, KALAVAKKAM
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UCS1711 - MOBILE APPLICATION DEVELOPMENT LAB

Assignment 2

Name: Jayannthan P T

Dept: CSE 'A'

Roll No.: 205001049

Create an Android mobile application which simulates a virtual keyboard with following features use intents to navigate between multiple activities.

- a. Use Grid Layout with each cell containing individual characters / numbers / special characters
- b. A TextView (multiple lines) to display the typed content.
- c. Design the keyboard as shown in Fig. 1.



- d. On press number key toggle to digits and special character view as in Fig. 2.
- e. On pressing each symbol, display the corresponding character in the textView.
- f. On pressing enter, move to newline in textView.
- g. On pressing backspace, erase a character in the textView.
- h. On pressing spacebar button provide single space

Additional: On press Up arrow, toggle between uppercase and lowercase letters.

Title of the Program: Create an Android mobile application which simulates a virtual keyboard with following features use intents to navigate between multiple activities.

Objective:

The objective of the Keyboard Android App project is to create a user-friendly keyboard application that allows users to input text efficiently using a virtual keyboard. Users can switch between letter and symbol modes, type letters and symbols, use capitalization, handle space, backspace, and enter, and eventually submit the typed text to be displayed in another activity..

Algorithm:

1. Create the main activity layout (activity_main.xml) with buttons for letters, symbols, and special functions like capitalization, space, backspace, and enter.
2. Implement the logic to handle button clicks for both letter and symbol modes.
3. Toggle between letter and symbol modes when the "Change" button is clicked.
4. Implement capitalization logic when the "Caps" button is clicked.
5. Handle space, backspace, and enter button clicks to modify the text in the EditText field.
6. Create a separate display activity layout (activity_display_data.xml) to display the typed text.
7. Pass the typed text from the main activity to the display activity using an intent with extras.
8. In the display activity, retrieve the text from the intent and display it in a TextView.

Features used:

1. Buttons for letter and symbol input.
2. Button for switching between letter and symbol modes.
3. Button for capitalization.
4. Buttons for space, backspace, and enter.
5. EditText widget for displaying typed text.
6. Passing data between activities using intents.

Source code:

- MainActivity.java

```
package com.example.keyboard;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {
    Button a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z;
    Button caps, backspace, space, enter, change;
    Button submit_btn, clear_btn;
    EditText editor;

    @Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    caps = findViewById(R.id.caps);
    change = findViewById(R.id.change);
    a = findViewById(R.id.a);
    b = findViewById(R.id.b);
    c = findViewById(R.id.c);
    d = findViewById(R.id.d);
    e = findViewById(R.id.e);
    f = findViewById(R.id.f);
    g = findViewById(R.id.g);
    h = findViewById(R.id.h);
    i = findViewById(R.id.i);
    j = findViewById(R.id.j);
    k = findViewById(R.id.k);
    l = findViewById(R.id.l);
    m = findViewById(R.id.m);
    n = findViewById(R.id.n);
    o = findViewById(R.id.o);
    p = findViewById(R.id.p);
    q = findViewById(R.id.q);
    r = findViewById(R.id.r);
    s = findViewById(R.id.s);
    t = findViewById(R.id.t);
    u = findViewById(R.id.u);
    v = findViewById(R.id.v);
    w = findViewById(R.id.w);
    x = findViewById(R.id.x);
    y = findViewById(R.id.y);
    z = findViewById(R.id.z);
    final boolean[] isUpperCase = { false };
    final boolean[] isSymbolMode = { false };
    caps.setOnClickListener(view -> {
        if (isSymbolMode[0]) {
            a.setText(isUpperCase[0] ? "A" : "a");
            b.setText(isUpperCase[0] ? "B" : "b");
            c.setText(isUpperCase[0] ? "C" : "c");
            d.setText(isUpperCase[0] ? "D" : "d");
            e.setText(isUpperCase[0] ? "E" : "e");
            f.setText(isUpperCase[0] ? "F" : "f");
            g.setText(isUpperCase[0] ? "G" : "g");
            h.setText(isUpperCase[0] ? "H" : "h");
            i.setText(isUpperCase[0] ? "I" : "i");
            j.setText(isUpperCase[0] ? "J" : "j");
            k.setText(isUpperCase[0] ? "K" : "k");
            l.setText(isUpperCase[0] ? "L" : "l");
            m.setText(isUpperCase[0] ? "M" : "m");
            n.setText(isUpperCase[0] ? "N" : "n");
            o.setText(isUpperCase[0] ? "O" : "o");
            p.setText(isUpperCase[0] ? "P" : "p");
            q.setText(isUpperCase[0] ? "Q" : "q");
            r.setText(isUpperCase[0] ? "R" : "r");
            s.setText(isUpperCase[0] ? "S" : "s");
        }
    });
}
```

```
t.setText(isUpperCase[0] ? "T" : "t");
u.setText(isUpperCase[0] ? "U" : "u");
v.setText(isUpperCase[0] ? "V" : "v");
w.setText(isUpperCase[0] ? "W" : "w");
x.setText(isUpperCase[0] ? "X" : "x");
y.setText(isUpperCase[0] ? "Y" : "y");
z.setText(isUpperCase[0] ? "Z" : "z");
} else {

    a.setText(isUpperCase[0] ? "a" : "A");
    b.setText(isUpperCase[0] ? "b" : "B");
    c.setText(isUpperCase[0] ? "c" : "C");
    d.setText(isUpperCase[0] ? "d" : "D");
    e.setText(isUpperCase[0] ? "e" : "E");
    f.setText(isUpperCase[0] ? "f" : "F");
    g.setText(isUpperCase[0] ? "g" : "G");
    h.setText(isUpperCase[0] ? "h" : "H");
    i.setText(isUpperCase[0] ? "i" : "I");
    j.setText(isUpperCase[0] ? "j" : "J");
    k.setText(isUpperCase[0] ? "k" : "K");
    l.setText(isUpperCase[0] ? "l" : "L");
    m.setText(isUpperCase[0] ? "m" : "M");
    n.setText(isUpperCase[0] ? "n" : "N");
    o.setText(isUpperCase[0] ? "o" : "O");
    p.setText(isUpperCase[0] ? "p" : "P");
    q.setText(isUpperCase[0] ? "q" : "Q");
    r.setText(isUpperCase[0] ? "r" : "R");
    s.setText(isUpperCase[0] ? "s" : "S");
    t.setText(isUpperCase[0] ? "t" : "T");
    u.setText(isUpperCase[0] ? "u" : "U");
    v.setText(isUpperCase[0] ? "v" : "V");
    w.setText(isUpperCase[0] ? "w" : "W");
    x.setText(isUpperCase[0] ? "x" : "X");
    y.setText(isUpperCase[0] ? "y" : "Y");
    z.setText(isUpperCase[0] ? "z" : "Z");
}
isUpperCase[0] = !isUpperCase[0];
});
change.setOnClickListener(view -> {
    if (isSymbolMode[0]) {
        // Switching from symbol mode to letter mode
        a.setText(isUpperCase[0] ? "A" : "a");
        b.setText(isUpperCase[0] ? "B" : "b");
        c.setText(isUpperCase[0] ? "C" : "c");
        d.setText(isUpperCase[0] ? "D" : "d");
        e.setText(isUpperCase[0] ? "E" : "e");
        f.setText(isUpperCase[0] ? "F" : "f");
        g.setText(isUpperCase[0] ? "G" : "g");
        h.setText(isUpperCase[0] ? "H" : "h");
        i.setText(isUpperCase[0] ? "I" : "i");
        j.setText(isUpperCase[0] ? "J" : "j");
        k.setText(isUpperCase[0] ? "K" : "k");
        l.setText(isUpperCase[0] ? "L" : "l");
        m.setText(isUpperCase[0] ? "M" : "m");
    }
}
```



```
        String currentText = editor.getText().toString();
        editor.setText(currentText + buttonText);
    }
};

int[] alphabetButtonIds = {
    R.id.a, R.id.b, R.id.c, R.id.d, R.id.e, R.id.f, R.id.g, R.id.h,
    R.id.i, R.id.j, R.id.k, R.id.l, R.id.m, R.id.n, R.id.o, R.id.p,
    R.id.q, R.id.r, R.id.s, R.id.t, R.id.u, R.id.v, R.id.w, R.id.x,
    R.id.y, R.id.z
};
for (int buttonId : alphabetButtonIds) {
    Button alphabetButton = findViewById(buttonId);
    alphabetButton.setOnClickListener(alphabetClickListener);
}
space = findViewById(R.id.space);
space.setOnClickListener(view -> {
    String currentText = editor.getText().toString();
    editor.setText(currentText + " ");
});
enter = findViewById(R.id.enter);
enter.setOnClickListener(view -> {
    String currentText = editor.getText().toString();
    editor.setText(currentText + "\n");
});
backspace = findViewById(R.id.backspace);
backspace.setOnClickListener(view -> {
    String currentText = editor.getText().toString();
    if (!currentText.isEmpty()) {
        String newText = currentText.substring(0, currentText.length() - 1);
        editor.setText(newText);
    }
});
submit_btn = findViewById(R.id.submit_btn);
submit_btn.setOnClickListener(view -> {
    Intent navigate = new Intent(MainActivity.this, DisplayActivity.class);
    navigate.putExtra("input_value", editor.getText().toString());
    startActivity(navigate);
});
clear_btn = findViewById(R.id.clear_btn);
clear_btn.setOnClickListener(view -> {
    String currentText = editor.getText().toString();
    if (!currentText.isEmpty()) {
        editor.setText("");
    }
});
}
```

- activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" tools:context=".MainActivity">
    <androidx.constraintlayout.widget.ConstraintLayout android:id="@+id/key_holder"
        android:layout_width="match_parent" android:layout_height="280dp"
        android:layout_margin="18dp" app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent" app:layout_constraintStart_toStartOf="parent">
        <TableLayout android:layout_width="wrap_content"
            android:layout_height="match_parent" app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent" app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" android:stretchColumns="*">
            <TableRow android:layout_gravity="center">

                <Button android:id="@+id/q" android:layout_margin="2dp"
                    android:textAllCaps="false" android:textColor="@color/button_text" android:textSize="18sp"
                    android:textStyle="bold" android:backgroundTint="@color/button_background"
                    android:layout_weight="1" android:text="q" />
                <Button android:id="@+id/w" android:layout_margin="2dp"
                    android:textAllCaps="false" android:textColor="@color/button_text" android:textSize="18sp"
                    android:textStyle="bold" android:backgroundTint="@color/button_background"
                    android:layout_weight="1" android:text="w" />
                <Button android:id="@+id/e" android:layout_margin="2dp"
                    android:textAllCaps="false" android:textColor="@color/button_text" android:textSize="18sp"
                    android:textStyle="bold" android:backgroundTint="@color/button_background"
                    android:layout_weight="1" android:text="e" />
                <Button android:id="@+id/r" android:layout_margin="2dp"
                    android:textAllCaps="false" android:textColor="@color/button_text" android:textSize="18sp"
                    android:textStyle="bold" android:backgroundTint="@color/button_background"
                    android:layout_weight="1" android:text="r" />
                <Button android:id="@+id/t" android:layout_margin="2dp"
                    android:textAllCaps="false" android:textColor="@color/button_text" android:textSize="18sp"
                    android:textStyle="bold" android:backgroundTint="@color/button_background"
                    android:layout_weight="1" android:text="t" />
                <Button android:id="@+id/y" android:layout_margin="2dp"
                    android:textAllCaps="false" android:textColor="@color/button_text" android:textSize="18sp"
                    android:textStyle="bold" android:backgroundTint="@color/button_background"
                    android:layout_weight="1" android:text="y" />
                <Button android:id="@+id/u" android:layout_margin="2dp"
                    android:textAllCaps="false" android:textColor="@color/button_text" android:textSize="18sp"
                    android:textStyle="bold" android:backgroundTint="@color/button_background"
                    android:layout_weight="1" android:text="u" />
            </TableRow>
            <TableRow android:layout_gravity="center">
                <Button android:id="@+id/a" android:layout_margin="2dp"
                    android:textAllCaps="false" android:textColor="@color/button_text" android:textSize="18sp"
                    android:textStyle="bold" android:backgroundTint="@color/button_background"
                    android:layout_weight="1" android:text="a" />
            
```



```
</TableRow>
<TableRow android:layout_gravity="center">
    <Button android:id="@+id/caps" android:layout_margin="2dp"
    android:layout_weight="1" android:textAllCaps="false"
    android:textColor="@color/button_text" android:textSize="18sp" android:textStyle="bold"
    android:backgroundTint="@color/button_background" android:text="↑" />
    <Button android:id="@+id/l" android:layout_margin="2dp"
    android:layout_weight="1" android:textAllCaps="false"
    android:textColor="@color/button_text" android:textSize="18sp" android:textStyle="bold"
    android:backgroundTint="@color/button_background" android:text="l" />
    <Button android:id="@+id/i" android:layout_margin="2dp"
    android:layout_weight="1" android:textAllCaps="false"
    android:textColor="@color/button_text" android:textSize="18sp" android:textStyle="bold"
    android:backgroundTint="@color/button_background" android:text="i" />
    <Button android:id="@+id/o" android:layout_margin="2dp"
    android:layout_weight="1" android:textAllCaps="false"
    android:textColor="@color/button_text" android:textSize="18sp" android:textStyle="bold"
    android:backgroundTint="@color/button_background" android:text="o" />
    <Button android:id="@+id/p" android:layout_margin="2dp"
    android:layout_weight="1" android:textAllCaps="false"
    android:textColor="@color/button_text" android:textSize="18sp" android:textStyle="bold"
    android:backgroundTint="@color/button_background" android:text="p" />
    <Button android:id="@+id/k" android:layout_margin="2dp"
    android:layout_weight="1" android:textAllCaps="false"
    android:textColor="@color/button_text" android:textSize="18sp" android:textStyle="bold"
    android:backgroundTint="@color/button_background" android:text="k" />
    <Button android:id="@+id/backspace" android:layout_margin="2dp"
    android:layout_weight="1" android:textAllCaps="false"
    android:textColor="@color/button_text" android:textSize="18sp" android:textStyle="bold"
    android:backgroundTint="@color/button_background" android:text="←" />
</TableRow>
<TableRow>
    <Button android:id="@+id/change" android:layout_margin="2dp"
    android:backgroundTint="@color/button_background" android:text="?\123"
    android:textAllCaps="false" android:textColor="@color/button_text" android:textSize="18sp"
    android:textStyle="bold" />
    <Button android:id="@+id/space" android:layout_margin="2dp"
    android:layout_weight="6" android:textAllCaps="false"
    android:textColor="@color/button_text" android:textSize="18sp" android:textStyle="bold"
    android:backgroundTint="@color/button_background" android:text="Space" />
    <Button android:id="@+id/enter" android:layout_margin="2dp"
    android:textAllCaps="false" android:textColor="@color/button_text" android:textSize="18sp"
    android:textStyle="bold" android:backgroundTint="@color/button_background"
    android:text="↲" />
</TableRow>
</TableLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
<EditText android:id="@+id/editor" android:layout_width="match_parent"
    android:layout_height="wrap_content" android:layout_margin="24dp" android:hint="Type
    here..." android:text="" app:layout_constraintBottom_toTopOf="@+id/submit_btn"
    app:layout_constraintEnd_toEndOf="parent" app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
    <Button android:id="@+id/submit_btn" android:layout_width="180dp"
    android:layout_height="wrap_content" android:layout_margin="12dp" android:text="Submit"
```

```
app:layout_constraintBottom_toTopOf="@+id/clear_btn"
app:layout_constraintEnd_toEndOf="parent" app:layout_constraintStart_toStartOf="parent" />
    <Button android:id="@+id/clear_btn" android:layout_width="180dp"
    android:layout_height="wrap_content" android:layout_margin="12dp" android:text="Clear"
    app:layout_constraintBottom_toTopOf="@+id/key_holder"
    app:layout_constraintEnd_toEndOf="parent" app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

- DisplayDataActivity.java

```
package com.example.keyboard;

import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

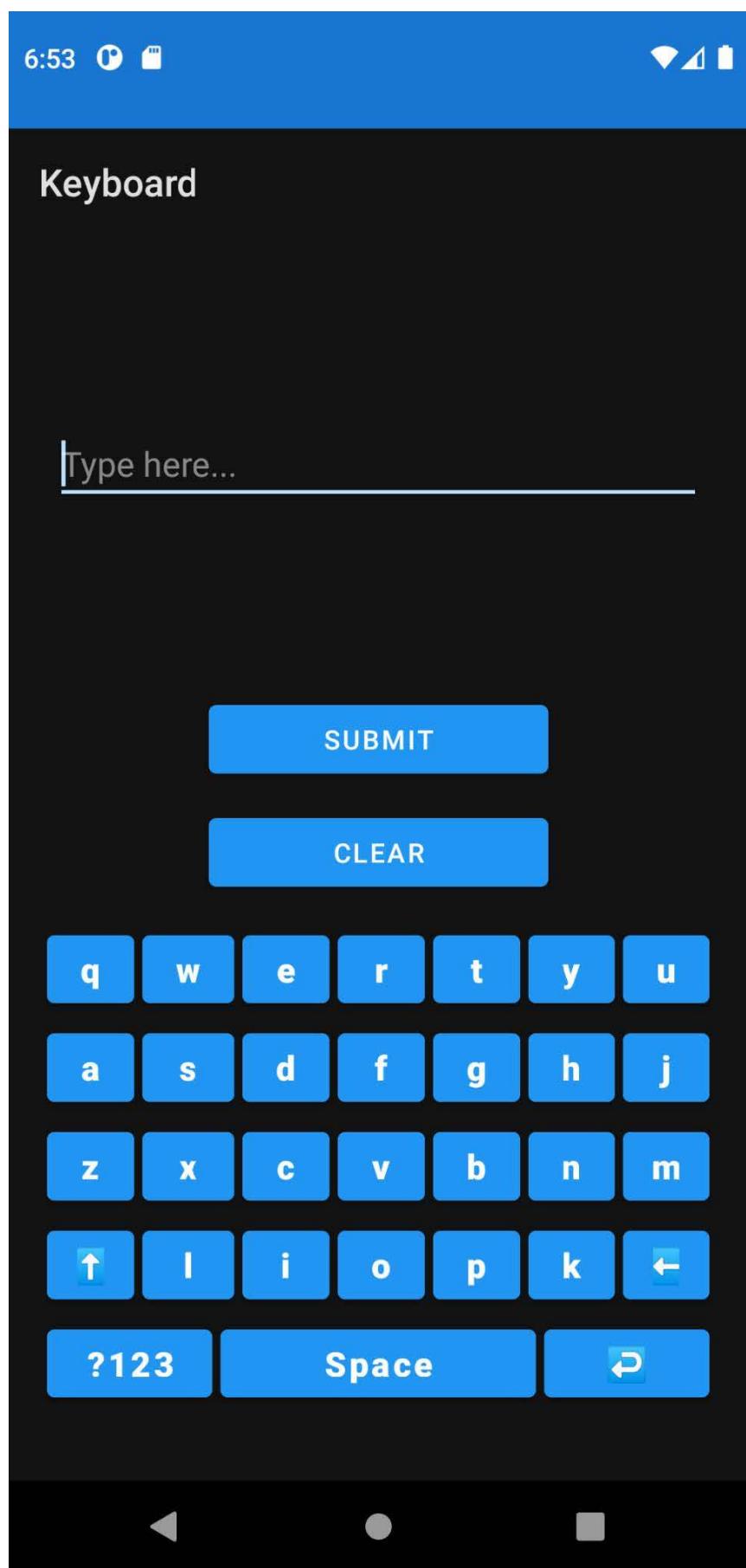
public class DisplayActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_display);
        Intent intent = getIntent();
        String editText = intent.getStringExtra("input_value");
        TextView preview = findViewById(R.id.preview_text);
        preview.setText(editText);
    }
}
```

- activity_display_data.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent" android:layout_height="match_parent">
    <TextView android:id="@+id/title" android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:layout_marginTop="36dp" android:text="You
    typed..." android:textSize="24sp" android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent" app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

    <TextView android:id="@+id/preview_text" android:layout_width="match_parent"
    android:layout_height="wrap_content" android:layout_margin="36dp" android:gravity="center"
    android:text="" app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/title" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Output:



6:53



Keyboard

Type here...

SUBMIT

CLEAR

& = 1 2 3 ? -

) (4 5 6 + %

' / 7 8 9 \$ #

↑ @ * 0 ^ ! ←

ABC

Space



6:53



Keyboard

Type here...

SUBMIT

CLEAR

Q W E R T Y U

A S D F G H J

Z X C V B N M

↑ L I O P K ←

?123

Space



Result:

The mobile application was completed successfully

Best Practices:

1. Use meaningful variable names and comments for better code readability.
2. Implement input validation to ensure that the user interface behaves correctly.
3. Utilize layout resources (XML) for UI consistency and responsiveness.
4. Organize the code with comments and logical structure to enhance maintainability.
5. Implement proper error handling to prevent crashes and provide a smooth user experience.

Learning Outcomes:

1. Designing Android user interfaces using XML layouts for a virtual keyboard.
2. Implementing interactive elements like buttons and EditText fields.
3. Handling user input events, including button clicks and text modifications.
4. Passing data between activities using intents with extras.
5. Enhancing Java coding skills for Android app development.
6. Managing user interactions and events effectively within the app.

SSN COLLEGE OF ENGINEERING, KALAVAKKAM
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UCS1711 - MOBILE APPLICATION DEVELOPMENT LAB

Assignment 3

Name: Jayannthan P T

Dept: CSE 'A'

Roll No.: 205001049

Application using graphical primitives

Design a CAR using Shape drawable with the help of relevant shapes such as Line, Circle, Rectangle and Arc.

- a. Move the car forward by pressing forward button so that car moves from a predefined starting point to the predefined endpoint.
- b. On pressing backward button, rotate the car to 180 degrees from the current point to the starting point.
- c. Implement a Tap-to-zoom animation on any image
- d. Implement the Card flipping animation.

Title of the Program: Design a CAR using Shape drawable with the help of relevant shapes such as Line, Circle, Rectangle and Arc.

Objective:

The objective of the Keyboard Android App project is to create a user-friendly keyboard application that allows users to input text efficiently using a virtual keyboard. Users can switch between letter and symbol modes, type letters and symbols, use capitalization, handle space, backspace, and enter, and eventually submit the typed text to be displayed in another activity.

Algorithm:

1. Create the main activity layout (activity_main.xml) with buttons for moving the car forward and backward, and a RelativeLayout to hold the car image.
2. Implement the logic to handle button clicks for moving the car forward and backward.
3. Implement logic to handle clicks on the car image to toggle zooming in and out.
4. Create XML shape resources for defining the car body, wheels, and other components.
5. Initialize the initial state of the car (position, zoom level, and rotation) in the MainActivity.
6. Implement methods for zooming in, zooming out, moving the car, and flipping the car.
7. Calculate the maximum translation distance to prevent the car from moving off the screen.
8. Use ObjectAnimator to animate the car's movement and rotation.
9. Update the current position and rotation values as the car moves and rotates.

Features used:

1. Buttons for moving the car forward and backward.
2. Button for toggling zoom on the car image.
3. Use of XML shape resources to define the car's appearance.
4. ObjectAnimator for animating car movement and rotation.
5. Handling clicks events on the car image.
6. Calculation of the maximum translation distance.

Source code:

- MainActivity.java

```
package com.example.ex3;

import android.animation.ObjectAnimator;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.RelativeLayout;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private RelativeLayout carLayout;
    private Button moveButton;
    private Button moveBack;
```

```
private int currentPosition = 0;
private boolean isFlipped = false;
private boolean isZoomed = false;

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

    carLayout = findViewById(R.id.carLayout);
    moveButton = findViewById(R.id.moveForwardButton);
    moveBack = findViewById(R.id.moveBackButton);

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

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

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

private void toggleZoom() {
    if (isZoomed) {
        zoomOut();
    } else {
        zoomIn();
    }
}

private void zoomIn() {
    carLayout.setScaleX(1.2f);
    carLayout.setScaleY(1.2f);
    isZoomed = true;
}

private void zoomOut() {
    carLayout.setScaleX(1f);
    carLayout.setScaleY(1f);
    isZoomed = false;
}
```

```

private void moveCar() {
    int start = currentPosition;
    int maxTranslation = getMaxTranslation();

    int end;
    if (isFlipped) {
        end = Math.max(start - 100, -maxTranslation);
    } else {
        end = Math.min(start + 100, maxTranslation);
    }

    ObjectAnimator animator;
    animator = ObjectAnimator.ofFloat(carLayout, "translationX", start, end);
    animator.setDuration(100);
    animator.start();

    currentPosition = end;
}

private int getMaxTranslation() {
    int screenWidth = getResources().getDisplayMetrics().widthPixels;
    int carWidth = (int) (200 * getResources().getDisplayMetrics().density);

    return (screenWidth - carWidth) / 2;
}

private void moveCarBackward() {
    ObjectAnimator rotationAnimator;
    int rotation;
    if (isFlipped == true) {
        rotation = 180;
        rotationAnimator = ObjectAnimator.ofFloat(carLayout, "rotationY", rotation,
0);
        isFlipped = false;
    } else {
        rotation = 0;
        rotationAnimator = ObjectAnimator.ofFloat(carLayout, "rotationY", rotation,
180);
        isFlipped = true;
    }
    rotationAnimator.setDuration(100);
    rotationAnimator.start();
}
}

```

- activity_main.xml

```

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

```

```

<!-- Buttons for moving forward and backward -->

<RelativeLayout android:id="@+id/carLayout" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView android:id="@+id/car" android:layout_width="200dp"
        android:layout_height="80dp" android:layout_marginLeft="100dp"
        android:layout_marginTop="100dp" android:src="@drawable/car" />

    <ImageView android:id="@+id/carwin" android:layout_width="89dp"
        android:layout_height="30dp" android:layout_marginLeft="190dp"
        android:layout_marginTop="105dp" android:src="@drawable/carwin" />
        android:layout_marginTop="168dp"

    <ImageView android:id="@+id/rightWheel" android:layout_width="42dp"
        android:layout_height="42dp" android:layout_marginLeft="248dp"
        android:layout_marginTop="168dp" android:src="@drawable/carwheel" />

    <ImageView android:id="@+id/leftWheel" android:layout_width="42dp"
        android:layout_height="42dp" android:layout_marginLeft="120dp"
        android:layout_marginTop="168dp" android:src="@drawable/carwheel" />

</RelativeLayout>

<Button android:id="@+id/moveForwardButton" android:layout_width="55dp"
    android:layout_height="wrap_content" android:layout_gravity="bottom|start"
    android:layout_marginStart="20dp" android:layout_marginBottom="20dp" android:text="GO" />

<Button android:id="@+id/moveBackButton" android:layout_width="130dp"
    android:layout_height="wrap_content" android:layout_gravity="bottom|end"
    android:layout_marginEnd="20dp" android:layout_marginBottom="20dp" android:text="REVERSE"
/>

</FrameLayout>

```

```

<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <gradient android:drawable="@drawable/car_body_gradient" android:type="linear"
        android:angle="0" />
    <solid android:color="#4444f4" />
    <size android:width="200dp" android:height="80dp" />
    <corners android:topRightRadius="90dp" />
</shape>

```

```

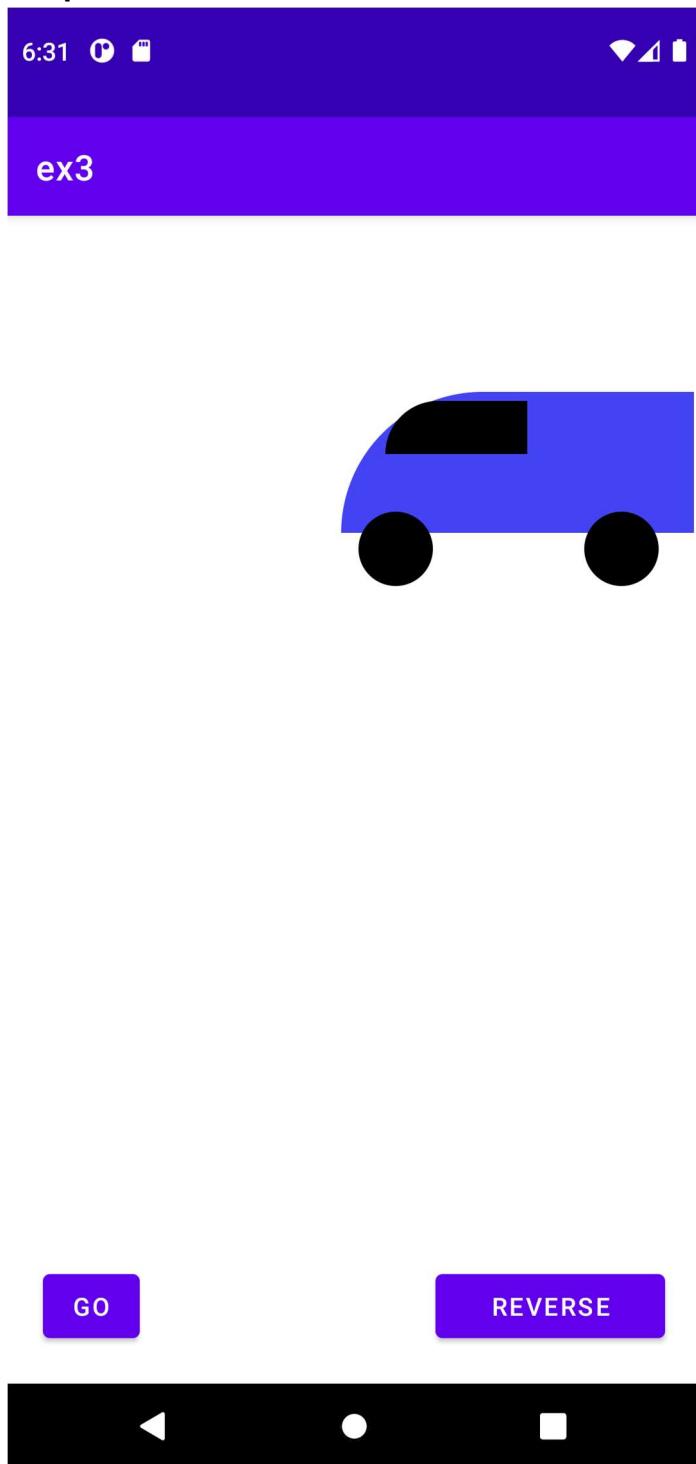
<shape xmlns:android="http://schemas.android.com/apk/res/android" android:shape="oval">
    <solid android:color="#000000" />
    <size android:width="42dp" android:height="42dp" />

```

```
</shape>
```

```
<shape xmlns:android="http://schemas.android.com/apk/res/android"  
    android:shape="rectangle">  
    <solid android:color="#000000" />  
    <size android:width="160dp" android:height="60dp" />  
    <corners android:topRightRadius="70dp" />  
</shape>
```

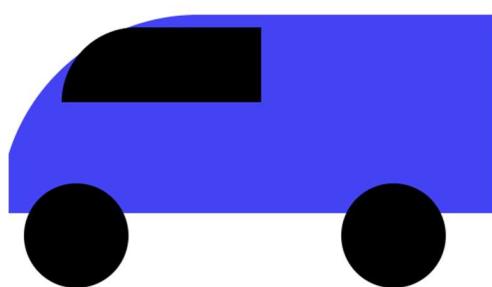
Output:



6:31



ex3



GO

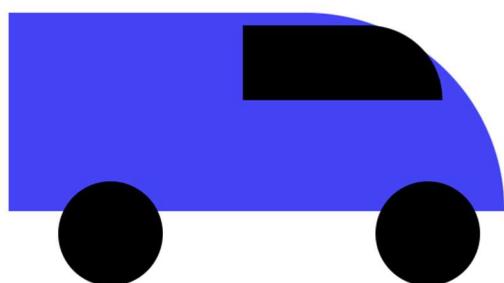
REVERSE



6:31



ex3



GO

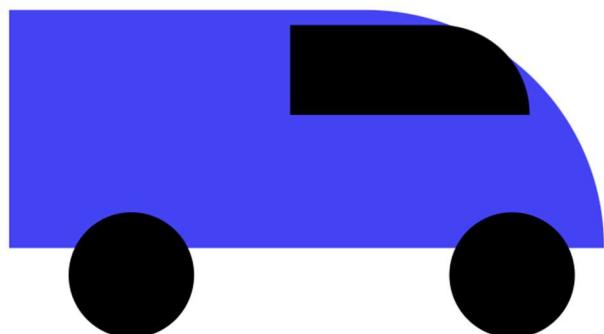
REVERSE



6:31



ex3



GO

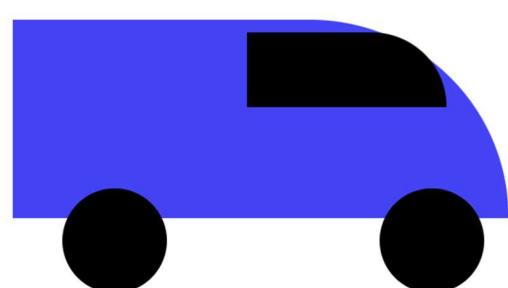
REVERSE



6:31



ex3



GO

REVERSE



Result:

The mobile application was completed successfully

Best Practices:

1. Use meaningful variable names and comments to improve code readability.
2. Utilize XML layouts for defining the UI components to maintain consistency.
3. Organize code logically and use comments for better code maintenance.
4. Implement error handling to handle unexpected scenarios gracefully.
5. Ensure smooth user interactions and animations for a better user experience.
6. Follow Android coding conventions and design guidelines.

Learning Outcomes:

1. Designing Android user interfaces using XML layouts.
2. Implementing interactive elements like buttons and click event handling.
3. Performing animations using ObjectAnimator to enhance user experience.
4. Handling user interactions effectively within the app.
5. Utilizing XML shape resources to define drawable components.
6. Enhancing Java coding skills for Android app development.
7. Understanding and implementing best practices for Android app development.

SSN COLLEGE OF ENGINEERING, KALAVAKKAM
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UCS1711 - MOBILE APPLICATION DEVELOPMENT LAB

Assignment 4

Name: Jayannthan P T

Dept: CSE 'A'

Roll No.: 205001049

Develop a Product information application in Android that enables to perform CRUD operations on data stored in SQLite Database.

In main activity display the following buttons: Create, Insert, Update, Delete, Retrieve, Retrieve All

1. On clicking Create Button, create a new database to store the Product details. (Use SQLite Database)
 - a. Product ID (Make this field as primary key)
 - b. Product Name
 - c. Product Brand
 - d. Description
 - e. Product Price
2. On Clicking Insert, move to a new view which contains the following details: (Insert new Product to the database)
 - a. Product ID (EditText-Validation checking- 4 digit Numbers)
 - b. Product Name (Spinner)
 - c. Product Brand (RadioButton)
 - d. Description (EditText-Alphanumeric characters)
 - e. Product Price (EditText-Validation checking Numbers)
 - f. Submit (Button) – On press, Insert the data into database.
3. On clicking Update, move to a new view which contains above details and Update Product Price using Product ID.
4. On clicking Delete, Delete the whole row in the table by Product ID.
5. On clicking Retrieve, Retrieve the product using Product ID.
6. On clicking Retrieve All, retrieve the details of all the products in a particular brand.

Title of the Program: Android Application Development using Database

Objective:

The objective of the provided Android application is to manage a list of products. Users can perform various operations such as inserting new products, updating product prices, deleting products, searching for product details by ID, and viewing products grouped by brand.

Algorithm:

1. MainActivity:
 - Set up buttons for inserting, deleting, updating, searching, and viewing products.
 - On button click, navigate to the respective activities.
2. InsertActivity:
 - Provide input fields for product ID, name, brand, description, and price.
 - On the "INSERT" button click:
 - Validate user input.
 - Insert a new product into the database.
3. UpdateActivity:
 - Allow users to input a product ID and a new price.
 - On the "UPDATE" button click:
 - Validate user input.
 - Update the price of the specified product in the database.
4. DeleteActivity:
 - Allow users to input a product ID for deletion.
 - On the "DELETE" button click:
 - Validate user input.
 - Delete the specified product from the database.
5. SearchActivity:
 - Enable users to search for product details by entering a product ID.
 - On the "SEARCH" button click:
 - Retrieve and display product details if found.
6. ViewActivity:
 - Display a list of product brands in a ListView.
 - On selecting a brand:
 - Retrieve and display a list of products for that brand.
7. DatabaseHelper:
 - Manage the SQLite database for storing product information.
 - Define database schema and operations for creating, querying, updating, and deleting records.

Features used:

- Android activities for user interaction.
- SQLite database for data storage.
- Input validation for user data.

- Navigation between activities.
- ListView for displaying lists of items.
- Custom XML layouts for UI design.

Source code:

- MainActivity.java

```
package com.example.products;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    Button insertButton;
    Button deleteButton;
    Button updateButton;
    Button searchButton;
    Button viewButton;

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

        // Initialize buttons inside the onCreate method
        insertButton = findViewById(R.id.insert);
        deleteButton = findViewById(R.id.delete);
        updateButton = findViewById(R.id.update);
        searchButton = findViewById(R.id.search);
        viewButton = findViewById(R.id.view);

        insertButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, InsertActivity.class);
                startActivity(intent);
            }
        });

        deleteButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, DeleteActivity.class);
                startActivity(intent);
            }
        });

        updateButton.setOnClickListener(new View.OnClickListener() {
```

```

        @Override
        public void onClick(View v) {
            Intent intent = new Intent(MainActivity.this, UpdateActivity.class);
            startActivity(intent);
        }
    });

    searchButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(MainActivity.this, SearchActivity.class);
            startActivity(intent);
        }
    });

    viewButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(MainActivity.this, ViewActivity.class);
            startActivity(intent);
        }
    });
}
}

```

- activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" tools:context=".MainActivity"
    tools:ignore="ExtraText">

    <TextView android:id="@+id/Heading" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="Product Database"
        app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.06" android:textSize="35sp" />

    <Button android:id="@+id/insert" android:layout_width="122dp"
        android:layout_height="55dp" android:text="Insert"
        app:layout_constraintTop_toBottomOf="@+id/Heading"
        app:layout_constraintStart_toStartOf="parent" app:layout_constraintEnd_toEndOf="parent"
        android:layout_marginTop="16dp" android:textSize="20sp" />

```

```

        <Button android:id="@+id/delete" android:layout_width="122dp"
    android:layout_height="55dp" android:text="Delete"
    app:layout_constraintTop_toBottomOf="@+id/insert"
    app:layout_constraintStart_toStartOf="parent" app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="16dp" android:textSize="20sp" />

        <Button android:id="@+id/update" android:layout_width="122dp"
    android:layout_height="55dp" android:text="Update"
    app:layout_constraintTop_toBottomOf="@+id/delete"
    app:layout_constraintStart_toStartOf="parent" app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="16dp" android:textSize="20sp" />

        <Button android:id="@+id/search" android:layout_width="122dp"
    android:layout_height="55dp" android:text="Search"
    app:layout_constraintTop_toBottomOf="@+id/update"
    app:layout_constraintStart_toStartOf="parent" app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="16dp" android:textSize="20sp" />

        <Button android:id="@+id/view" android:layout_width="122dp"
    android:layout_height="55dp" android:text="View"
    app:layout_constraintTop_toBottomOf="@+id/search"
    app:layout_constraintStart_toStartOf="parent" app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="16dp" android:textSize="20sp" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

- InsertActivity.java

```

package com.example.products;

import android.content.ContentValues;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class InsertActivity extends AppCompatActivity {

    private EditText productIdEditText, productNameEditText, productBrandEditText,
    productDescriptionEditText, productPriceEditText;
    private Button submitButton;

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

        // Initialize UI elements
        productIdEditText = findViewById(R.id.id);

```

```
productNameEditText = findViewById(R.id.name);
productBrandEditText = findViewById(R.id.brand);
productDescriptionEditText = findViewById(R.id.desc);
productPriceEditText = findViewById(R.id.price);
submitButton = findViewById(R.id.submit);

// Set click listener for the Submit button
submitButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // Get user input
        String productId = productIdEditText.getText().toString();
        String productName = productNameEditText.getText().toString();
        String productBrand = productBrandEditText.getText().toString();
        String productDescription =
productDescriptionEditText.getText().toString();
        String productPriceStr = productPriceEditText.getText().toString();

        // Check if any of the fields are empty
        if (productId.isEmpty() || productName.isEmpty() || productBrand.isEmpty()
|| productDescription.isEmpty() || productPriceStr.isEmpty()) {
            Toast.makeText(InsertActivity.this, "Please fill in all fields",
Toast.LENGTH_SHORT).show();
            return;
        }

        // Convert product price to double
        double productPrice = Double.parseDouble(productPriceStr);

        // Insert data into the database
        insertProduct(productId, productName, productBrand, productDescription,
productPrice);

        // Show success message
        Toast.makeText(InsertActivity.this, "Product inserted successfully",
Toast.LENGTH_SHORT).show();

        // Clear input fields
        productIdEditText.getText().clear();
        productNameEditText.getText().clear();
        productBrandEditText.getText().clear();
        productDescriptionEditText.getText().clear();
        productPriceEditText.getText().clear();
    }
});

private void insertProduct(String productId, String productName, String productBrand,
String productDescription, double productPrice) {
    // Create or open the database for writing
    DatabaseHelper dbHelper = new DatabaseHelper(this);
    SQLiteDatabase db = dbHelper.getWritableDatabase();

    // Create a new map of values, where column names are the keys
```

```

        ContentValues values = new ContentValues();
        values.put(DatabaseContract.ProductEntry.COLUMN_PRODUCT_ID, productId);
        values.put(DatabaseContract.ProductEntry.COLUMN_PRODUCT_NAME, productName);
        values.put(DatabaseContract.ProductEntry.COLUMN_PRODUCT_BRAND, productBrand);
        values.put(DatabaseContract.ProductEntry.COLUMN_PRODUCT_DESCRIPTION,
productDescription);
        values.put(DatabaseContract.ProductEntry.COLUMN_PRODUCT_PRICE, productPrice);

        // Insert the new row, returning the primary key value of the new row
        long newRowId = db.insert(DatabaseContract.ProductEntry.TABLE_NAME, null, values);

        // Close the database
        db.close();
    }
}

```

- Insert.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools" android:id="@+id/linearLayout"
    android:layout_width="match_parent" android:layout_height="match_parent">

    <TextView android:id="@+id/Heading" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="Insert" android:textSize="35sp"
        app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.06" />

    <TableLayout android:layout_width="0dp" android:layout_height="0dp"
        app:layout_constraintTop_toBottomOf="@+id/Heading"
        app:layout_constraintStart_toStartOf="parent" app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintBottom_toBottomOf="parent" android:layout_marginTop="16dp">

        <!-- First Row -->
        <TableRow android:layout_height="wrap_content">
            <TextView android:layout_width="0dp" android:layout_height="wrap_content"
                android:layout_weight="1" android:text="Product ID" android:gravity="center" />

            <EditText android:id="@+id/id" android:layout_width="0dp"
                android:layout_height="wrap_content" android:layout_weight="1" android:ems="10"
                android:inputType="text" android:text="" />
        </TableRow>

        <!-- Second Row -->
        <TableRow android:layout_height="wrap_content">
            <TextView android:layout_width="0dp" android:layout_height="wrap_content"
                android:layout_weight="1" android:text="Product Name" android:gravity="center" />

```

```

        <EditText android:id="@+id/name" android:layout_width="0dp"
        android:layout_height="wrap_content" android:layout_weight="1" android:ems="10"
        android:inputType="text" android:text="" />
    </TableRow>

    <!-- Third Row (and so on) -->
    <TableRow android:layout_height="wrap_content">
        <TextView android:layout_width="0dp" android:layout_height="wrap_content"
        android:layout_weight="1" android:text="Product Brand" android:gravity="center" />

        <EditText android:id="@+id/brand" android:layout_width="0dp"
        android:layout_height="wrap_content" android:layout_weight="1" android:ems="10"
        android:inputType="text" android:text="" />
    </TableRow>

    <TableRow android:layout_height="wrap_content">
        <TextView android:layout_width="0dp" android:layout_height="wrap_content"
        android:layout_weight="1" android:text="Product Description" android:gravity="center" />

        <EditText android:id="@+id/desc" android:layout_width="0dp"
        android:layout_height="wrap_content" android:layout_weight="1" android:ems="10"
        android:inputType="text" android:text="" />
    </TableRow>

    <TableRow android:layout_height="wrap_content">
        <TextView android:layout_width="0dp" android:layout_height="wrap_content"
        android:layout_weight="1" android:text="Product Price" android:gravity="center" />

        <EditText android:id="@+id/price" android:layout_width="0dp"
        android:layout_height="wrap_content" android:layout_weight="1" android:ems="10"
        android:inputType="numberDecimal" android:text="" />
    </TableRow>

    <TableRow android:layout_height="wrap_content">
        <Button android:id="@+id/submit" android:layout_width="0dp"
        android:layout_height="wrap_content" android:layout_weight="1" android:text="INSERT"
        android:background="@color/black" android:textColor="@color/white"/>
    </TableRow>
</TableLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

```

- UpdateActivity.java

```

package com.example.products;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

```

```
import androidx.appcompat.app.AppCompatActivity;

public class UpdateActivity extends AppCompatActivity {

    private EditText productIdEditText;
    private EditText newPriceEditText;
    private Button updateButton;
    private DatabaseHelper dbHelper;

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

        productIdEditText = findViewById(R.id.id);
        newPriceEditText = findViewById(R.id.price);
        updateButton = findViewById(R.id.submit);
        dbHelper = new DatabaseHelper(this);

        updateButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                // Get the product ID and new price from the EditText fields
                String productIdText = productIdEditText.getText().toString().trim();
                String newPriceText = newPriceEditText.getText().toString().trim();

                if (productIdText.isEmpty() || newPriceText.isEmpty()) {
                    // Product ID or new price field is empty, show an error message
                    Toast.makeText(UpdateActivity.this, "Product ID and new price are required", Toast.LENGTH_SHORT)
                        .show();
                } else {
                    try {
                        // Parse the product ID and new price as needed (integer and double)
                        int productId = Integer.parseInt(productIdText);
                        double newPrice = Double.parseDouble(newPriceText);

                        // Update the product price in the database
                        boolean isUpdated = dbHelper.updateProductPrice(productId, newPrice);

                        if (isUpdated) {
                            // Show a success message
                            Toast.makeText(UpdateActivity.this, "Price updated successfully", Toast.LENGTH_SHORT)
                                .show();
                        } else {
                            // Show an error message if the update failed
                            Toast.makeText(UpdateActivity.this, "Failed to update price. Product not found.", Toast.LENGTH_SHORT).show();
                        }
                    } catch (NumberFormatException e) {
                        // Handle parsing errors
                        Toast.makeText(UpdateActivity.this, "Invalid input: Please enter valid integer values for Product ID and New Price.", Toast.LENGTH_SHORT).show();
                    }
                }
            }
        });
    }
}
```

```
// Clear the EditText fields
productIdEditText.setText("");
newPriceEditText.setText("");
} catch (NumberFormatException e) {
    // Handle invalid input (non-integer or non-double)
    Toast.makeText(UpdateActivity.this, "Invalid Product ID or new
price", Toast.LENGTH_SHORT)
        .show();
}
}
});
```

- update.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools" android:id="@+id/linearLayout"
    android:layout_width="match_parent" android:layout_height="match_parent">

    <TextView android:id="@+id/Heading" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="Update" android:textSize="35sp"
        app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.06" />

    <TableLayout android:id="@+id/searchTable" android:layout_width="0dp"
        android:layout_height="wrap_content" app:layout_constraintTop_toBottomOf="@+id/Heading"
        app:layout_constraintStart_toStartOf="parent" app:layout_constraintEnd_toEndOf="parent"
        android:layout_marginTop="16dp">

        <!-- Row for Product ID Search -->
        <TableRow android:layout_height="wrap_content">
            <TextView android:layout_width="0dp" android:layout_height="wrap_content"
                android:layout_weight="1" android:text="Product ID" android:gravity="center" />

            <EditText android:id="@+id/id" android:layout_width="0dp"
                android:layout_height="wrap_content" android:layout_weight="1" android:ems="10"
                android:inputType="text" android:text="" />
        </TableRow>
        <TableRow android:layout_height="wrap_content">
            <TextView android:layout_width="0dp" android:layout_height="wrap_content"
                android:layout_weight="1" android:text="Product Price" android:gravity="center" />

            <EditText android:id="@+id/price" android:layout_width="0dp"
                android:layout_height="wrap_content" android:layout_weight="1" android:ems="10"
                android:inputType="numberDecimal" android:text="" />
        </TableRow>
    </TableLayout>

```

```
        </TableRow>
        <TableRow android:layout_height="wrap_content">
            <Button android:id="@+id/submit" android:layout_width="0dp"
android:layout_height="wrap_content" android:layout_weight="1" android:text="UPDATE PRICE"
android:background="@color/black" android:textColor="@color/white"
app:layout_constraintTop_toBottomOf="@id/price" />
        </TableRow>
    </TableLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

- DeleteActivity.java

```
        int productId = Integer.parseInt(productIdText);

        // Attempt to delete the product by ID
        boolean isDeleted = dbHelper.deleteProductById(productId);

        if (isDeleted) {
            // Product deleted successfully
            Toast.makeText(DeleteActivity.this, "Product deleted
successfully", Toast.LENGTH_SHORT).show();

            // Clear the product ID EditText
            productIdEditText.setText("");
        } else {
            // Product not found, show an error message
            Toast.makeText(DeleteActivity.this, "Product not found for
deletion", Toast.LENGTH_SHORT).show();
        }
    } catch (NumberFormatException e) {
        // Handle invalid input (non-integer)
        Toast.makeText(DeleteActivity.this, "Invalid Product ID",
Toast.LENGTH_SHORT).show();
    }
}
});
```

- delete.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools" android:id="@+id/linearLayout"
    android:layout_width="match_parent" android:layout_height="match_parent">

    <TextView android:id="@+id/Heading" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="Delete" android:textSize="35sp"
        app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.06" />

    <TableLayout android:id="@+id/searchTable" android:layout_width="0dp"
        android:layout_height="wrap_content" app:layout_constraintTop_toBottomOf="@+id/Heading"
        app:layout_constraintStart_toStartOf="parent" app:layout_constraintEnd_toEndOf="parent"
        android:layout_marginTop="16dp">

        <!-- Row for Product ID Search -->
        <TableRow android:layout_height="wrap_content">
```

```

        <TextView android:layout_width="0dp" android:layout_height="wrap_content"
        android:layout_weight="1" android:text="Product ID" android:gravity="center" />

        <EditText android:id="@+id/id" android:layout_width="0dp"
        android:layout_height="wrap_content" android:layout_weight="1" android:ems="10"
        android:inputType="text" android:text="" />
    </TableRow>

    <TableRow android:layout_height="wrap_content">
        <Button android:id="@+id/submit" android:layout_width="0dp"
        android:layout_height="wrap_content" android:layout_weight="1" android:text="DELETE"
        android:background="@color/black" android:textColor="@color/white"
        app:layout_constraintTop_toBottomOf="@id/id" />
    </TableRow>
</TableLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

```

- SearchActivity.java

```

package com.example.products;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TableLayout;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class SearchActivity extends AppCompatActivity {

    private EditText productIdEditText;
    private TableLayout outputTable;
    private TextView nameTextView;
    private TextView brandTextView;
    private TextView descriptionTextView;
    private TextView priceTextView;

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

        // Initialize UI elements
        productIdEditText = findViewById(R.id.id);
        outputTable = findViewById(R.id.outputTable);
        nameTextView = findViewById(R.id.name);
        brandTextView = findViewById(R.id.brand);
        descriptionTextView = findViewById(R.id.desc);
        priceTextView = findViewById(R.id.price);
    }
}

```

```

        Button searchButton = findViewById(R.id.submit);

        searchButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                searchProduct();
            }
        });
    }

    private void searchProduct() {
        // Get the product ID entered by the user
        String productIdText = productIdEditText.getText().toString();

        if (productIdText.isEmpty()) {
            // Product ID field is empty, show an error message or handle it as needed
            Toast.makeText(this, "Please enter a Product ID", Toast.LENGTH_SHORT).show();
            return;
        }

        int productId = Integer.parseInt(productIdText);

        // Use the DatabaseHelper to search for the product by ID
        DatabaseHelper databaseHelper = new DatabaseHelper(this);
        Product product = databaseHelper.searchProductById(productId);

        if (product != null) {
            // Display the product information in the outputTable
            nameTextView.setText(product.getName());
            brandTextView.setText(product.getBrand());
            descriptionTextView.setText(product.getDescription());
            priceTextView.setText(String.valueOf(product.getPrice()));

            // Make the outputTable visible
            outputTable.setVisibility(View.VISIBLE);
        } else {
            // Product not found, you can display an error message or handle it as needed
            Toast.makeText(this, "Product not found", Toast.LENGTH_SHORT).show();

            // For now, let's hide the outputTable
            outputTable.setVisibility(View.GONE);
        }
    }
}

```

- search.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"

```

```
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools" android:id="@+id/linearLayout"
android:layout_width="match_parent" android:layout_height="match_parent">

    <!-- Heading TextView -->
    <TextView android:id="@+id/Heading" android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:text="Search" android:textSize="35sp"
    app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.06" />

    <TableLayout android:id="@+id/searchTable" android:layout_width="0dp"
    android:layout_height="wrap_content" app:layout_constraintTop_toBottomOf="@+id/Heading"
    app:layout_constraintStart_toStartOf="parent" app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="16dp">

        <!-- Row for Product ID Search -->
        <TableRow android:layout_height="wrap_content">
            <TextView android:layout_width="0dp" android:layout_height="wrap_content"
            android:layout_weight="1" android:text="Product ID" android:gravity="center" />

            <EditText android:id="@+id/id" android:layout_width="0dp"
            android:layout_height="wrap_content" android:layout_weight="1" android:ems="10"
            android:inputType="text" android:text="" />
        </TableRow>

        <!-- Row for Search Button -->
        <TableRow android:layout_height="wrap_content">
            <Button android:id="@+id/submit" android:layout_width="0dp"
            android:layout_height="wrap_content" android:layout_weight="1" android:text="Search"
            android:background="@color/black" android:textColor="@color/white"
            app:layout_constraintTop_toBottomOf="@+id/id" />
        </TableRow>
    </TableLayout>

    <TableLayout android:id="@+id/outputTable" android:layout_width="0dp"
    android:layout_height="0dp" app:layout_constraintTop_toBottomOf="@+id/searchTable"
    app:layout_constraintStart_toStartOf="parent" app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintBottom_toBottomOf="parent" android:layout_marginTop="16dp"
    android:visibility="gone">

        <TableRow android:layout_height="wrap_content">
            <TextView android:layout_width="0dp" android:layout_height="wrap_content"
            android:layout_weight="1" android:text="Product Name" android:gravity="center" />

            <EditText android:id="@+id/name" android:layout_width="0dp"
            android:layout_height="wrap_content" android:layout_weight="1" android:ems="10"
            android:inputType="text" android:text="" />
        </TableRow>

        <TableRow android:layout_height="wrap_content">
            <TextView android:layout_width="0dp" android:layout_height="wrap_content"
            android:layout_weight="1" android:text="Product Brand" android:gravity="center" />
        </TableRow>
    </TableLayout>

```

```

        <EditText android:id="@+id/brand" android:layout_width="0dp"
android:layout_height="wrap_content" android:layout_weight="1" android:ems="10"
android:inputType="text" android:text="" />
    </TableRow>

    <TableRow android:layout_height="wrap_content">
        <TextView android:layout_width="0dp" android:layout_height="wrap_content"
android:layout_weight="1" android:text="Product Description" android:gravity="center" />

        <EditText android:id="@+id/desc" android:layout_width="0dp"
android:layout_height="wrap_content" android:layout_weight="1" android:ems="10"
android:inputType="text" android:text="" />
    </TableRow>

    <TableRow android:layout_height="wrap_content">
        <TextView android:layout_width="0dp" android:layout_height="wrap_content"
android:layout_weight="1" android:text="Product Price" android:gravity="center" />

        <EditText android:id="@+id/price" android:layout_width="0dp"
android:layout_height="wrap_content" android:layout_weight="1" android:ems="10"
android:inputType="numberDecimal" android:text="" />
    </TableRow>
</TableLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

```

- ViewActivity.java

```

package com.example.products;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;

import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.TextView;
import java.util.List;

public class ViewActivity extends AppCompatActivity {
    private DatabaseHelper databaseHelper;
    private LinearLayout rightSideLayout;

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

        ListView brandListView = findViewById(R.id.brandListView);

```

```
rightSideLayout = findViewById(R.id.rightSideLayout);

databaseHelper = new DatabaseHelper(this);

List<String> brandList = databaseHelper.getBrandList();

ArrayAdapter<String> adapter = new ArrayAdapter<>(this,
android.R.layout.simple_list_item_1, brandList);

brandListView.setAdapter(adapter);

brandListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, android.view.View view, int position, long id) {
        String selectedBrand = brandList.get(position);

        List<Product> productList =
databaseHelper.getProductsByBrand(selectedBrand);

        // Clear the existing product information
        rightSideLayout.removeAllViews();

        // ...

        if (!productList.isEmpty()) {
            for (Product product : productList) {
                // Create a TextView for each product
                TextView productTextView = new TextView(ViewActivity.this);
                productTextView.setLayoutParams(new ViewGroup.LayoutParams(
                    ViewGroup.LayoutParams.MATCH_PARENT,
                    ViewGroup.LayoutParams.WRAP_CONTENT // Set height to
wrap_content
                ));
                productTextView.setText("Product ID: " + product.getId() + "\n" +
                    "Product Name: " + product.getName() + "\n" +
                    "Product Price: " + product.getPrice());

                // Add the TextView to the rightSideLayout
                rightSideLayout.addView(productTextView);

                // Add space (spacer TextView)
                TextView spacer = new TextView(ViewActivity.this);
                spacer.setLayoutParams(new ViewGroup.LayoutParams(
                    ViewGroup.LayoutParams.MATCH_PARENT,
                    50
                ));
                rightSideLayout.addView(spacer);
            }
        } else {
            // Display a message if no products are available for the selected
            brand
            TextView noProductsTextView = new TextView(ViewActivity.this);
            noProductsTextView.setLayoutParams(new ViewGroup.LayoutParams(

```

```
        ViewGroup.LayoutParams.MATCH_PARENT,  
        ViewGroup.LayoutParams.WRAP_CONTENT  
    ));  
    noProductsTextView.setText("No products available for this brand.");  
  
    // Add the message to the rightSideLayout  
    rightSideLayout.addView(noProductsTextView);  
}  
  
}  
});  
  
}  
}
```

- view.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.core.widget.NestedScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools" android:id="@+id/linearLayout"
    android:layout_width="match_parent" android:layout_height="match_parent">

    <LinearLayout android:layout_width="match_parent" android:layout_height="wrap_content"
        android:orientation="vertical">

        <!-- Heading TextView -->
        <TextView android:id="@+id/Heading" android:layout_width="wrap_content"
            android:layout_height="wrap_content" android:text="View" android:textSize="35sp"
            android:layout_gravity="center" android:layout_marginTop="16dp"
            android:layout_marginBottom="16dp" />

        <LinearLayout android:layout_width="match_parent" android:layout_height="600dp"
            android:orientation="horizontal">

            <!-- Left side: List of Brands -->
            <ListView android:id="@+id/brandListView" android:layout_width="0dp"
                android:layout_height="wrap_content" android:layout_weight="1"
                android:divider="@android:color/darker_gray" android:dividerHeight="1dp" />

            <View android:layout_width="1dp" android:layout_height="match_parent"
                android:background="@android:color/darker_gray" />

            <LinearLayout android:id="@+id/rightSideLayout" android:layout_width="0dp"
                android:layout_height="wrap_content" android:orientation="vertical"
                android:layout_weight="1">

                </LinearLayout>
    
```

```
</LinearLayout>
</LinearLayout>
</androidx.core.widget.NestedScrollView>
```

- DatabaseHelper.java

```
// DatabaseHelper.java
package com.example.products;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

import java.util.ArrayList;
import java.util.List;

public class DatabaseHelper extends SQLiteOpenHelper {
    private static final String DATABASE_NAME = "products";
    private static final int DATABASE_VERSION = 1;

    public static final String TABLE_NAME = "products";

    public static final String COLUMN_PRODUCT_ID = "product_id";
    public static final String COLUMN_PRODUCT_NAME = "product_name";
    public static final String COLUMN_PRODUCT_BRAND = "product_brand";
    public static final String COLUMN_PRODUCT_DESCRIPTION = "product_description";
    public static final String COLUMN_PRODUCT_PRICE = "product_price";

    private static final String SQL_CREATE_PRODUCT_TABLE =
        "CREATE TABLE " + TABLE_NAME + " (" +
            COLUMN_PRODUCT_ID + " INTEGER PRIMARY KEY AUTOINCREMENT," +
            COLUMN_PRODUCT_NAME + " TEXT," +
            COLUMN_PRODUCT_BRAND + " TEXT," +
            COLUMN_PRODUCT_DESCRIPTION + " TEXT," +
            COLUMN_PRODUCT_PRICE + " REAL)";

    private static final String SQL_DELETE_PRODUCT_TABLE =
        "DROP TABLE IF EXISTS " + TABLE_NAME;

    public DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(SQL_CREATE_PRODUCT_TABLE);
    }
}
```

```
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    db.execSQL(SQL_DELETE_PRODUCT_TABLE);
    onCreate(db);
}

public Product searchProductById(int productId) {
    SQLiteDatabase db = this.getReadableDatabase();

    String[] projection = {
        COLUMN_PRODUCT_NAME,
        COLUMN_PRODUCT_BRAND,
        COLUMN_PRODUCT_DESCRIPTION,
        COLUMN_PRODUCT_PRICE
    };

    String selection = COLUMN_PRODUCT_ID + " = ?";
    String[] selectionArgs = {String.valueOf(productId)};

    Cursor cursor = db.query(
        TABLE_NAME,
        projection,
        selection,
        selectionArgs,
        null,
        null,
        null
    );

    Product product = null;
    if (cursor != null && cursor.moveToFirst()) {
        String name =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_PRODUCT_NAME));
        String brand =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_PRODUCT_BRAND));
        String description =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_PRODUCT_DESCRIPTION));
        double price =
cursor.getDouble(cursor.getColumnIndexOrThrow(COLUMN_PRODUCT_PRICE));

        product = new Product(productId, name, brand, description, price);
        cursor.close();
    }

    db.close();
    return product;
}
public boolean deleteProductById(int productId) {
    SQLiteDatabase db = this.getWritableDatabase();
    String whereClause = COLUMN_PRODUCT_ID + " = ?";
    String[] whereArgs = {String.valueOf(productId)};

    int rowsDeleted = db.delete(TABLE_NAME, whereClause, whereArgs);
```

```
// Close the database
db.close();

// Check if any rows were deleted (deletion successful)
return rowsDeleted > 0;
}

public boolean updateProductPrice(int productId, double newPrice) {
    SQLiteDatabase db = this.getWritableDatabase();

    ContentValues values = new ContentValues();
    values.put(COLUMN_PRODUCT_PRICE, newPrice);

    String whereClause = COLUMN_PRODUCT_ID + " = ?";
    String[] whereArgs = {String.valueOf(productId)};

    int rowsUpdated = db.update(TABLE_NAME, values, whereClause, whereArgs);

    // Close the database
    db.close();

    // Check if any rows were updated (update successful)
    return rowsUpdated > 0;
}

public List<String> getBrandList() {
    List<String> brandList = new ArrayList<>();
    SQLiteDatabase db = this.getReadableDatabase();

    // SQL query to select distinct brands
    String query = "SELECT DISTINCT " + COLUMN_PRODUCT_BRAND + " FROM " + TABLE_NAME;

    Cursor cursor = db.rawQuery(query, null);

    if (cursor != null && cursor.moveToFirst()) {
        do {
            String brand = cursor.getString(0); // 0 is the column index for brand
            brandList.add(brand);
            Log.d("DatabaseHelper", "Brand: " + brand); // Add this line for debugging
        } while (cursor.moveToNext());
    }

    cursor.close();
}

db.close();
return brandList;
}

// Add a method to select all items in a brand
public List<Product> getProductsByBrand(String brandName) {
    List<Product> productList = new ArrayList<>();
    SQLiteDatabase db = this.getReadableDatabase();

    String[] projection = {
```

```

        COLUMN_PRODUCT_ID,
        COLUMN_PRODUCT_NAME,
        COLUMN_PRODUCT_PRICE
    };

    String selection = COLUMN_PRODUCT_BRAND + " = ?";
    String[] selectionArgs = {brandName};

    Cursor cursor = db.query(
        TABLE_NAME,
        projection,
        selection,
        selectionArgs,
        null,
        null,
        null
    );

    if (cursor != null && cursor.moveToFirst()) {
        do {
            int productId =
cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN_PRODUCT_ID));
            String name =
cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_PRODUCT_NAME));
            double price =
cursor.getDouble(cursor.getColumnIndexOrThrow(COLUMN_PRODUCT_PRICE));

            Product product = new Product(productId, name, brandName, null, price);
            productList.add(product);
        } while (cursor.moveToNext());
    }

    cursor.close();
}

db.close();
return productList;
}
}

```

- DatabaseContract.java

```

// DatabaseContract.java
package com.example.products;

import android.provider.BaseColumns;

public final class DatabaseContract {
    private DatabaseContract() {}

    public static class ProductEntry implements BaseColumns {
        public static final String TABLE_NAME = "products";
    }
}

```

```
    public static final String COLUMN_PRODUCT_ID = "product_id";
    public static final String COLUMN_PRODUCT_NAME = "product_name";
    public static final String COLUMN_PRODUCT_BRAND = "product_brand";
    public static final String COLUMN_PRODUCT_DESCRIPTION = "product_description";
    public static final String COLUMN_PRODUCT_PRICE = "product_price";
}
}
```

- Product.java

```
// Product.java
package com.example.products;

public class Product {
    private int id;
    private String name;
    private String brand;
    private String description;
    private double price;

    public Product(int id, String name, String brand, String description, double price) {
        this.id = id;
        this.name = name;
        this.brand = brand;
        this.description = description;
        this.price = price;
    }

    public int getId() {
        return id;
    }

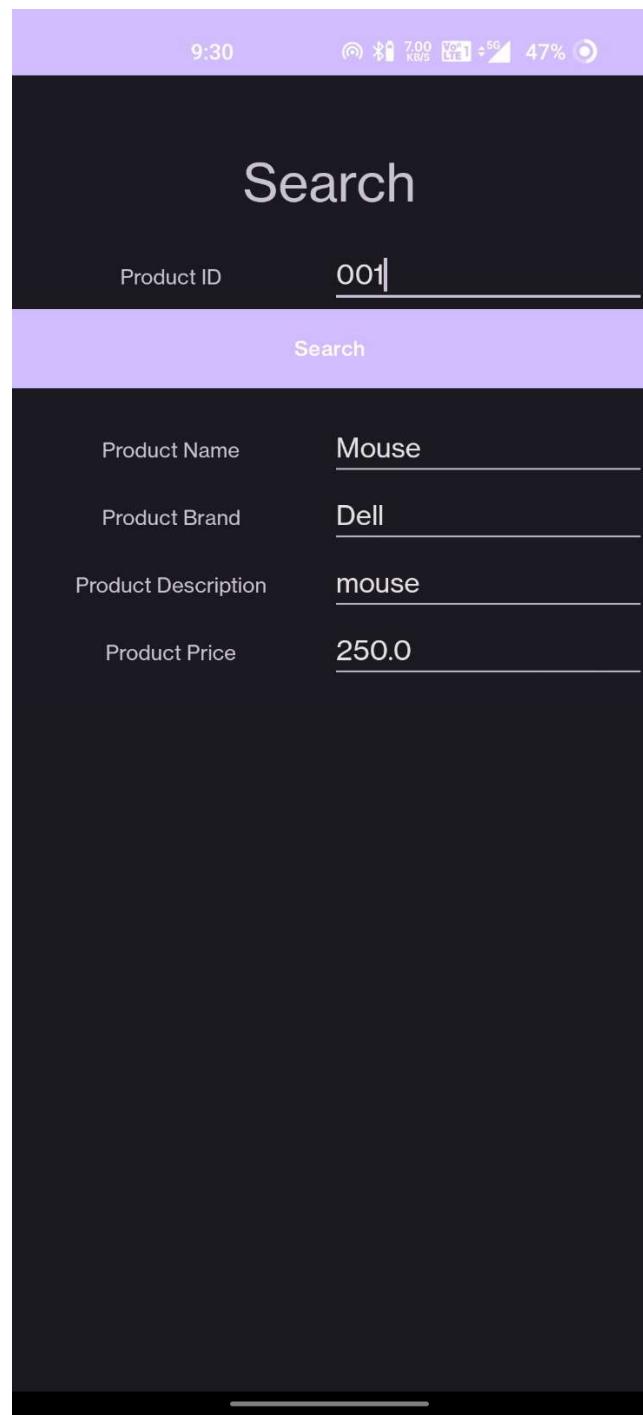
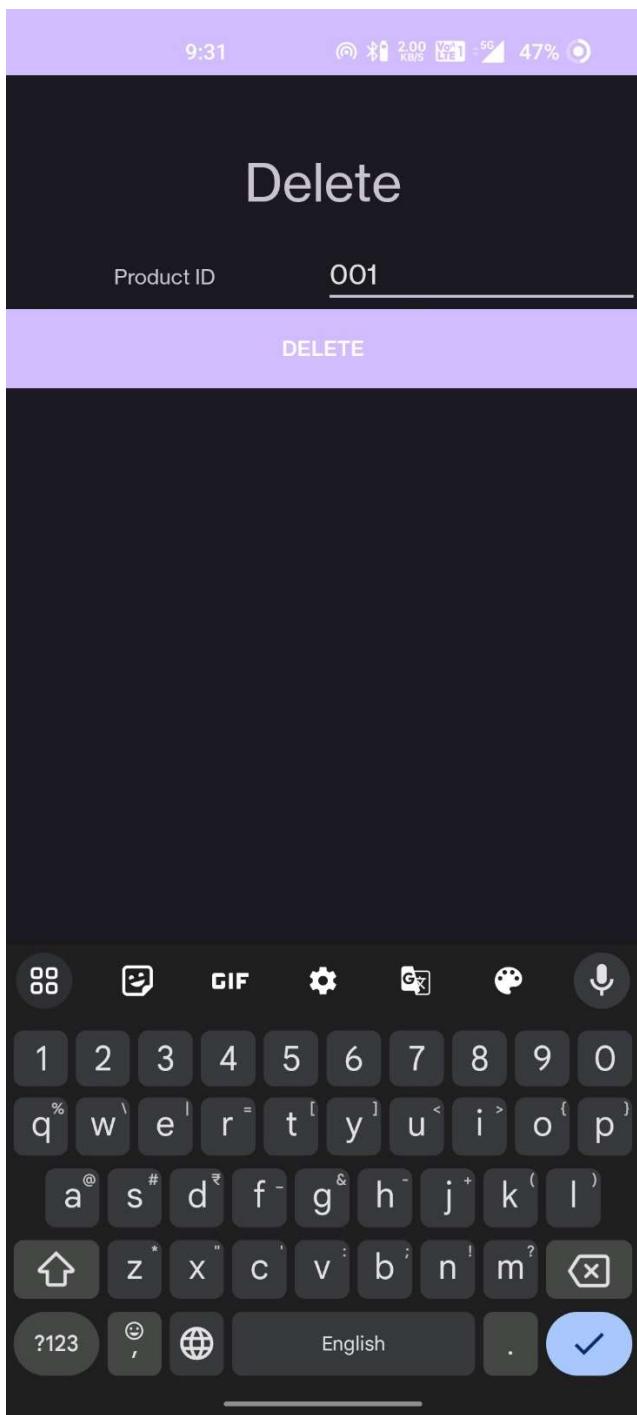
    public String getName() {
        return name;
    }

    public String getBrand() {
        return brand;
    }

    public String getDescription() {
        return description;
    }

    public double getPrice() {
        return price;
    }
}
```

Output:



9:31

0.03 KB/S VoIP 1 5G 47%

Search

Product ID

001

Search

Product Name

Mouse

Product Brand

Dell

Product Description

mouse

Product Price

600.0

9:30

0.33 KB/S VoIP 1 5G 47%

Update

Product ID

Product Price

UPDATE PRICE

1

2

3

-

4



Price updated successfully

7

8

9

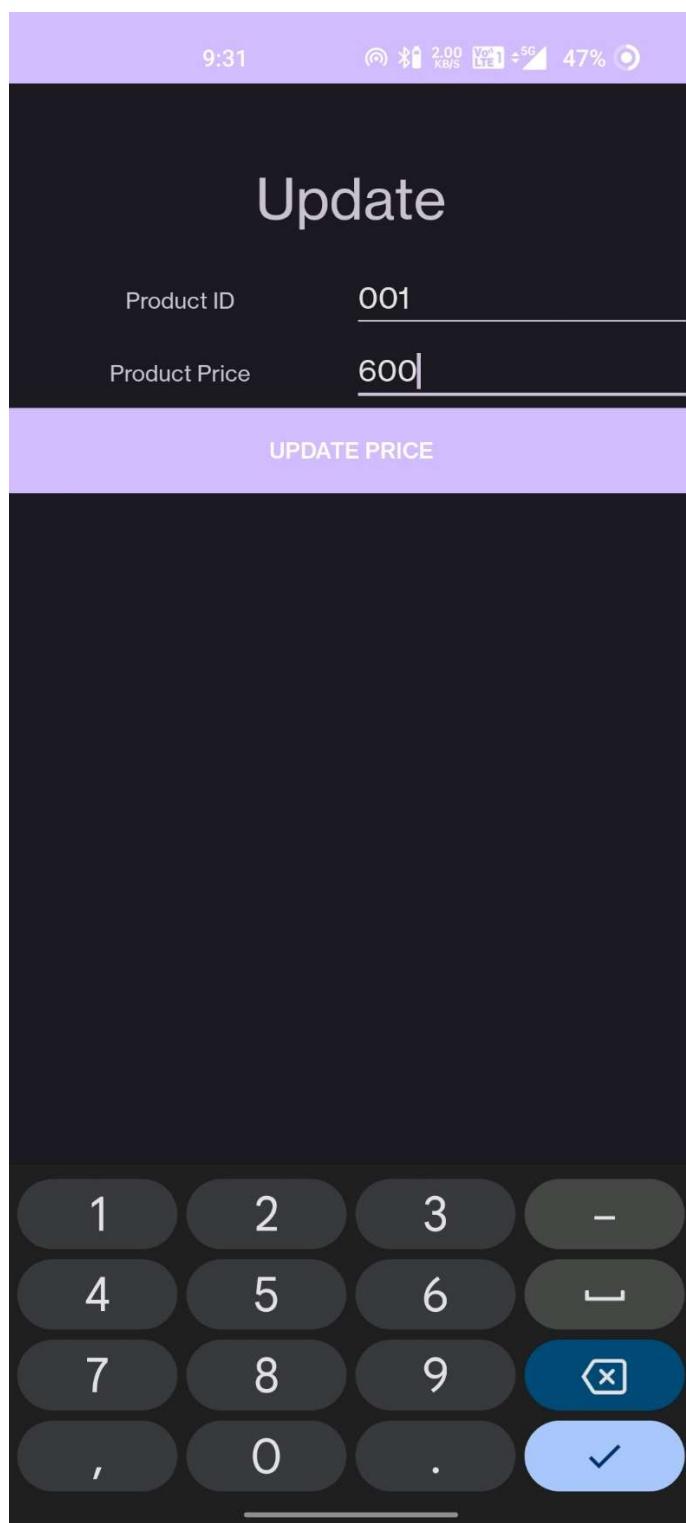
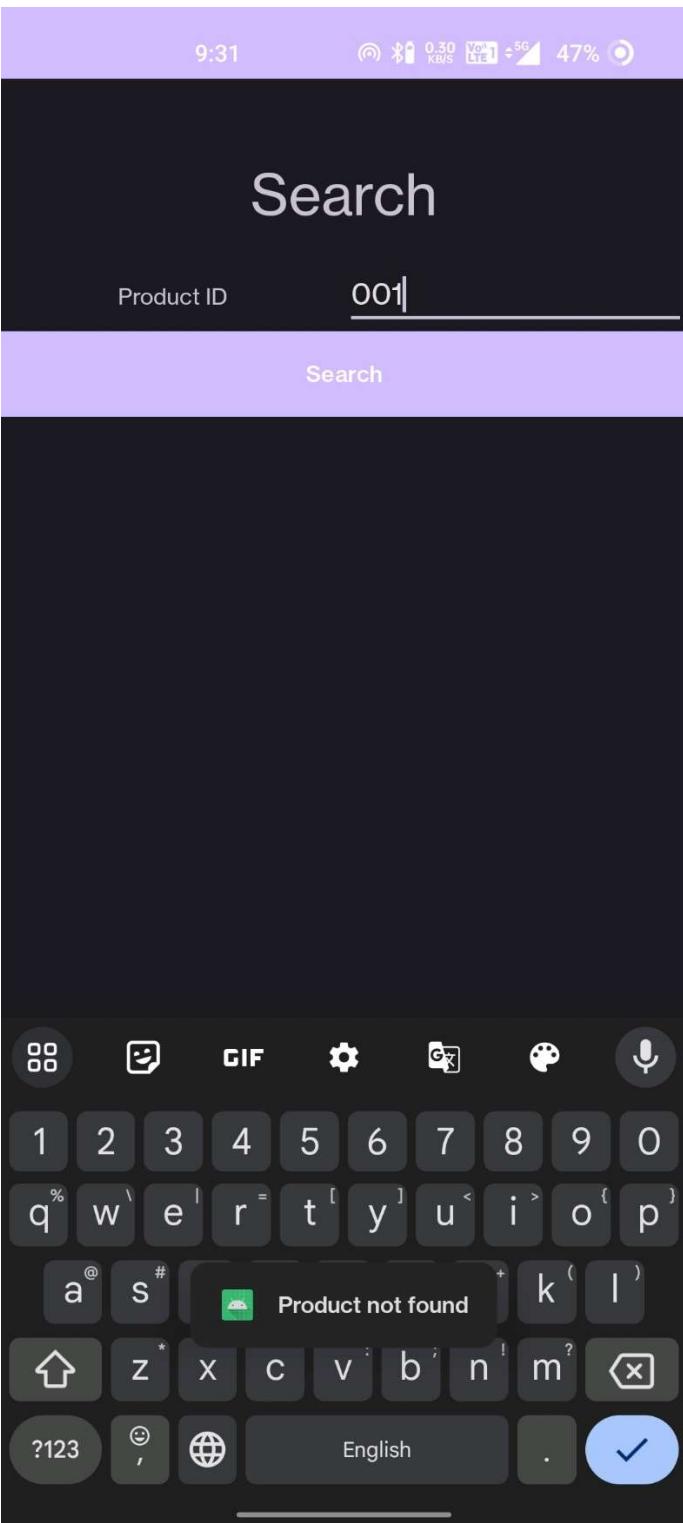


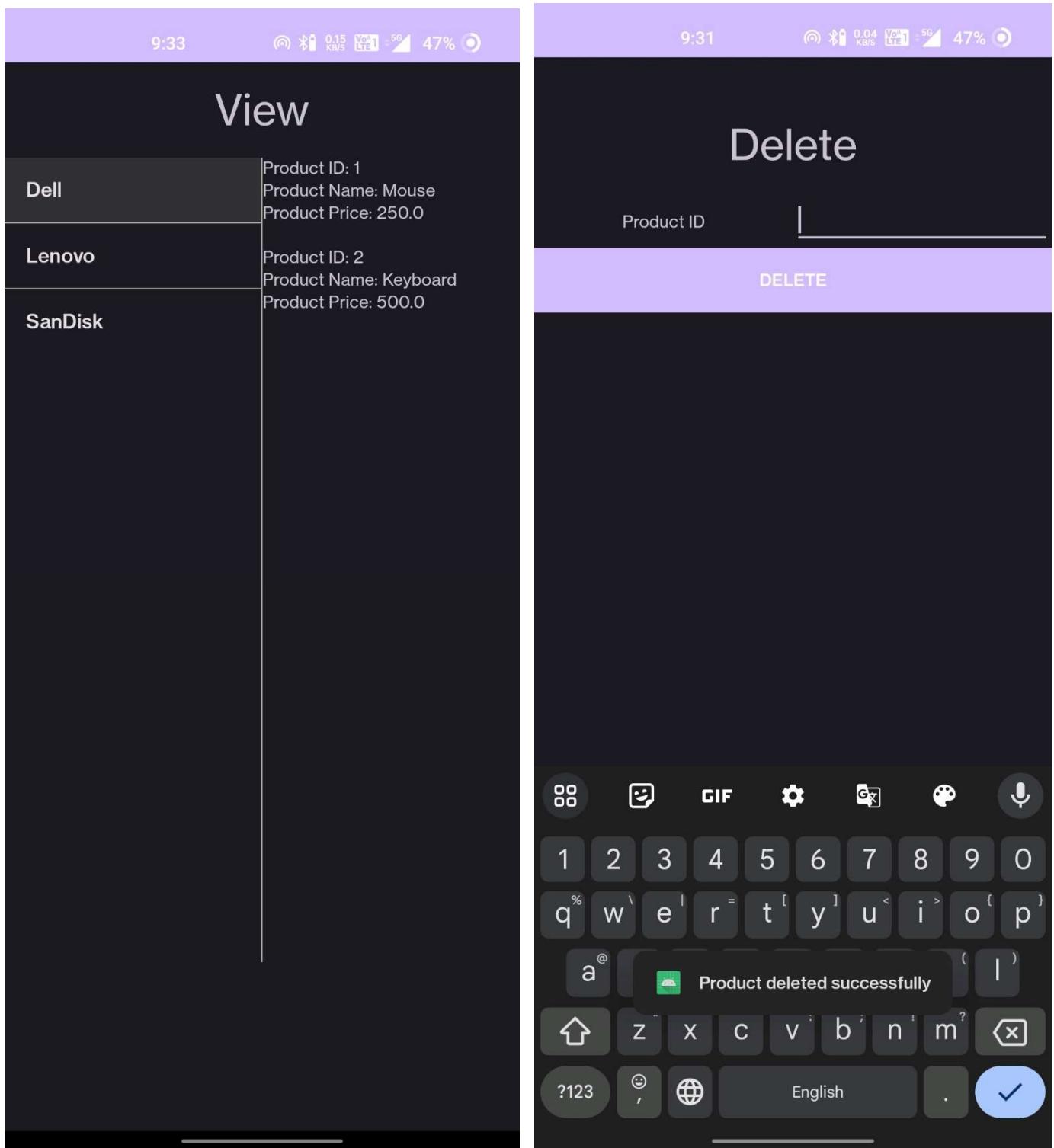
,

0

.







Result:

The mobile application was completed successfully

Best Practices:

- Using SQLite for local data storage.
- Input validation to ensure data integrity.
- Separation of concerns by using different activities for various operations.
- Reusable database helper class.

Learning Outcomes:

- Creating and managing Android activities.
- Working with SQLite databases in Android.
- Implementing input validation in Android applications.
- Navigating between different activities.
- Designing user interfaces with XML layout files.
- Handling user input and performing database operations.

SSN COLLEGE OF ENGINEERING, KALAVAKKAM
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UCS1711 - MOBILE APPLICATION DEVELOPMENT LAB

Assignment 5

Name: Jayannthan P T

Dept: CSE 'A'

Roll No.: 205001049

ANDROID APPLICATION USING MULTITHREADING

Ex. No:5

Title of the Program: Develop an android application to perform multithreading. Define 3 threads to run concurrently when "start" button is clicked.

Objective:

The objective of the Multithreading Android App project is to create an application that demonstrates the use of multithreading in Android. It defines three threads that run concurrently when the "Start" button is clicked. These threads are responsible for changing text color, moving a banner, and incrementing a counter.

Algorithm:

1. Create the main activity layout (activity_main.xml) containing TextViews for displaying various effects and buttons for controlling the threads.
2. Implement three Runnable objects, each representing a separate thread:
 - Runnable for changing text color: It changes the color of the "change_color" TextView in a loop.
 - Runnable for moving a banner: It uses ObjectAnimator to move the "moving_banner" TextView horizontally.
 - Runnable for counting: It updates the "counter" TextView and resets the count when it reaches a certain limit.
3. Create and start three threads (thread1, thread2, thread3) corresponding to the three Runnable objects.
4. Implement click event handlers for the "Start," "Resume," and "Stop" buttons:
 - "Start" button starts all three threads.
 - "Resume" button resumes the paused threads by notifying them.
 - "Stop" button stops the threads by setting a shared boolean variable (running) to false.
5. Use synchronization (lock1, lock2, lock3) to control the execution of threads and ensure proper thread communication.

6. Update UI components using runOnUiThread for UI-related actions within threads.

Features used:

- Threads: Creation and management of three concurrent threads.
- Runnable: Implementing Runnable objects for thread tasks.
- ObjectAnimator: Animating the movement of a TextView.
- Synchronization: Using locks to control thread execution and communication.
- UI updates: Updating UI components within threads using runOnUiThread.
- Button click event handling: Implementing click event handlers for buttons.

Source code:

- MainActivity.java

```
package com.example.exercise6;
import androidx.appcompat.app.AppCompatActivity;
import android.animation.ObjectAnimator;
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.view.animation.TranslateAnimation;
import android.widget.Button;
import android.widget.TextView;
public class MainActivity extends AppCompatActivity {
    public static Boolean running = true;
    public static Object lock1 = new Object();
    public static Object lock2 = new Object();
    public static Object lock3 = new Object();
    public int count = 0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        TextView change_color = findViewById(R.id.change_color);
        TextView moving_banner = findViewById(R.id.moving_banner);
        TextView counter = findViewById(R.id.counter);
        Runnable runnable1 = new Runnable() {
            @Override
            public void run() {
                while (true) {
                    synchronized(lock1) {
                        if (!running) {
                            try {
                                lock1.wait();
                            } catch (InterruptedException e) {
                                e.printStackTrace();
                            }
                        }
                    }
                    runOnUiThread(new Runnable() {
                        @Override
                        public void run() {
                            change_color.setTextColor(Color.parseColor("#00FF00"));
                        }
                    });
                }
            }
        };
    }
}
```

```
        }
    });
    try {
        Thread.sleep(500);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            change_color.setTextColor(Color.parseColor("#FFFF00"));
        }
    });
    try {
        Thread.sleep(500);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            change_color.setTextColor(Color.parseColor("#FF0000"));
        }
    });
    try {
        Thread.sleep(500);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}
}
};

Thread thread1 = new Thread(runnable1);
Runnable runnable2 = new Runnable() {
    @Override
    public void run() {
        while (true) {
            synchronized(lock2) {
                if (!running) {
                    try {
                        lock2.wait();
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                }
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        ObjectAnimator left_to_right =
                            ObjectAnimator.ofFloat(moving_banner, "translationX",
-200 f, 200 f);
                        left_to_right.setDuration(2000);
                        left_to_right.start();
                    }
                });
            }
        }
    }
};
```

```
        }
    });
    try {
        Thread.sleep(500);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}
}
};

Thread thread2 = new Thread(runnable2);
Runnable runnable3 = new Runnable() {
    @Override
    public void run() {
        while (true) {
            synchronized(lock3) {
                if (!running) {
                    try {
                        lock3.wait();
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                }
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        counter.setText("Counter: " + count);
                        count++;
                        if (count > 1000) count = 0;
                    }
                });
            }
            try {
                Thread.sleep(500);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
};

Thread thread3 = new Thread(runnable3);
Button start = findViewById(R.id.start);
start.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        thread1.start();
        thread2.start();
        thread3.start();
    }
});
Button resume = findViewById(R.id.resume);
resume.setOnClickListener(new View.OnClickListener() {
    @Override
```

```

        public void onClick(View view) {
            running = true;
            synchronized(lock1) {
                lock1.notify();
            }
            synchronized(lock2) {
                lock2.notify();
            }
            synchronized(lock3) {
                lock3.notify();
            }
        }
    });
Button stop = findViewById(R.id.stop);
stop.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        running = false;
    }
});
}
}
}

```

- activity_main.xml

```

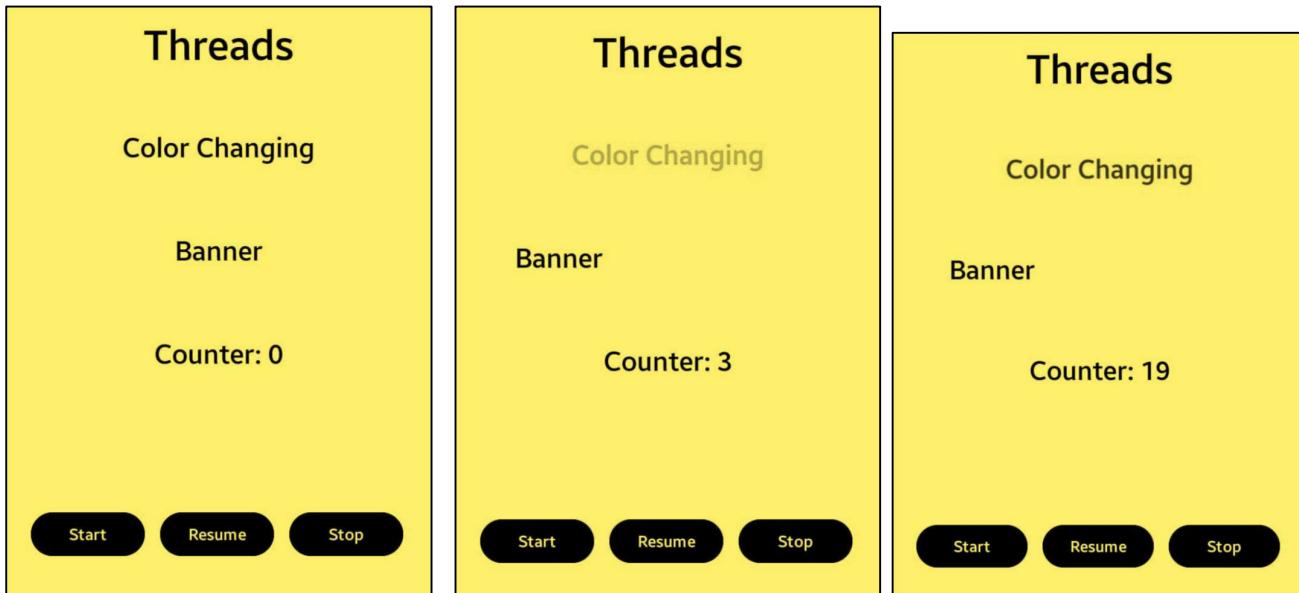
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:layout_margin="24dp"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/title"
        android:layout_width="match_parent"
        android:layout_height="100dp"
        android:gravity="center|center_vertical|center_horizontal"
        android:textSize="40dp"
        android:text="Threads"
        android:textStyle="bold"
        android:typeface="monospace"
        android:textColor="#000000" />
    <TextView android:id="@+id/change_color"
        android:layout_width="match_parent"
        android:layout_height="100dp"
        android:gravity="center|center_vertical|center_horizontal"
        android:textSize="24sp"
        android:text="Color Changing"
        android:textStyle="bold"
        android:typeface="monospace" />

```

```
        android:textColor="#000000" />
    <TextView
        android:id="@+id/moving_banner"
        android:layout_width="match_parent"
        android:layout_height="100dp"
        android:gravity="center|center_vertical|center_horizontal"
        android:textSize="24sp"
        android:text="Banner"
        android:textStyle="bold"
        android:typeface="monospace"
        android:textColor="#000000" />
    <TextView
        android:id="@+id/counter"
        android:layout_width="match_parent"
        android:layout_height="100dp"
        android:gravity="center|center_vertical|center_horizontal"
        android:textSize="24sp"
        android:text="Counter: 0"
        android:textStyle="bold"
        android:typeface="monospace"
        android:textColor="#000000" />
    <TableLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <TableRow
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginTop="100dp">
            <Button
                android:id="@+id/start"
                android:layout_width="110dp"
                android:layout_height="50dp"
                android:gravity="center|center_vertical|center_horizontal"
                android:textSize="14sp"
                android:text="Start"
                android:textStyle="bold"
                android:typeface="monospace"
                android:textColor="#ffffffff"
                android:backgroundTint="#000000" />
            <Button
                android:id="@+id/resume"
                android:layout_width="110dp"
                android:layout_height="50dp"
                android:gravity="center|center_vertical|center_horizontal"
                android:layout_marginStart="15dp"
                android:textSize="14sp"
                android:text="Resume"
                android:textStyle="bold"
                android:typeface="monospace"
                android:textColor="#ffffffff"
                android:backgroundTint="#000000" />
            <Button
                android:id="@+id/stop"
                android:layout_width="110dp"
```

```
        android:layout_height="50dp"
        android:gravity="center|center_vertical|center_horizontal"
        android:layout_marginStart="15dp"
        android:textSize="14sp"
        android:text="Stop"
        android:textStyle="bold"
        android:typeface="monospace"
        android:textColor="#ffffffff"
        android:backgroundTint="#000000" />
    </TableRow>
</TableLayout>
</LinearLayout>
```

Output:



Result:

The mobile application was completed successfully

Best Practices:

- Use meaningful variable and method names to improve code readability.
- Employ synchronization mechanisms to prevent race conditions and ensure proper thread communication.
- Implement proper error handling and exception catching for robust code.
- Separate UI-related code from background thread code using `runOnUiThread`.
- Follow Android coding conventions and design guidelines for a consistent and user-friendly app.
- Consider using constants or resources for hard-coded values like color codes and duration.

Learning Outcomes:

- Demonstrating multithreading in an Android application.
- Creating and managing threads to perform concurrent tasks.
- Implementing Runnable objects to define the behavior of threads.
- Using ObjectAnimator for animations within Android apps.
- Utilizing synchronization to control thread execution and communication.
- Handling button click events for user interaction.
- Gaining insights into proper multithreading practices and error handling in Android development.

SSN COLLEGE OF ENGINEERING, KALAVAKKAM
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UCS1711 - MOBILE APPLICATION DEVELOPMENT LAB

Assignment 6

Name: Jayannthan P T

Dept: CSE 'A'

Roll No.: 205001049

Finding geo coordinates of a location and Reverse Geocoding

a) Develop an android application to find the latitude and longitude of current location and the selected location in a google map using anyone of the below options:

- 1) Location Manager
- 2) Network Provider
- 3) GPS Provider

b) Also perform Reverse Geocoding i.e., given a latitude and longitude of a location, app should display the location name or given a location name it should display the latitude and longitude of that place.

Ex. No:6

Title of the Program: Create an Android mobile application which finds geo coordinates of a location and Reverse Geocoding.

Objective:

The objective of the Location Finder Android App project is to develop an application that enables users to find location coordinates (latitude and longitude) based on a provided address. The app utilizes the device's GPS functionality and geocoding to convert an address into corresponding geographical coordinates.

Algorithm:

1. Request runtime permission for accessing fine location to enable GPS functionality.
2. Set up a button click listener to trigger the location-finding process.
3. Create a LocationManager instance and request location updates from the GPS provider.
4. Implement the LocationListener interface to receive updates when the device's location changes.
5. Upon receiving a location update, retrieve latitude and longitude values.
6. Use Geocoder to convert a user-provided address (from an EditText) into location coordinates.
7. Display the latitude, longitude, and address details on the UI.

Features used:

1. Runtime permission handling for accessing fine location.
2. Utilization of LocationManager for obtaining location updates.
3. Implementation of LocationListener to respond to location changes.
4. Geocoding with Geocoder to convert an address into coordinates.
5. UI components such as TextViews and EditText for displaying and inputting information.
6. Button click handling to initiate the location-finding process.

Source code:

- **MainActivity.java**

```
package com.example.exercise5;

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

import android.Manifest;
import android.annotation.SuppressLint;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import java.util.List;
import java.util.Locale;

public class MainActivity extends AppCompatActivity implements LocationListener {

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

        if (ContextCompat.checkSelfPermission(MainActivity.this,
                Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(MainActivity.this, new String[] {
                Manifest.permission.ACCESS_FINE_LOCATION
            }, 100);
        }
        ;

        Button find = (Button) findViewById(R.id.find);
        find.setOnClickListener(new View.OnClickListener() {
            @Override
```

```

        public void onClick(View view) {
            findLocation();
        }
    });

}

@SuppressWarnings("MissingPermission")
private void findLocation() {
    try {
        LocationManager locationManager = (LocationManager)
this.getSystemService(LOCATION_SERVICE);
        locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
                5000, 5, MainActivity.this);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

@Override
public void onLocationChanged(@NonNull Location location) {
    TextView latitude = (TextView) findViewById(R.id.latitude);
    TextView longitude = (TextView) findViewById(R.id.longitude);
    String get_latitude = latitude.getText().toString();
    latitude.setText(get_latitude + location.getLatitude());
    String get_longitude = longitude.getText().toString();
    longitude.setText(get_longitude + location.getLongitude());
    try {
        Geocoder geocoder = new Geocoder(MainActivity.this, Locale.getDefault());
        List<Address> addresses = geocoder.getFromLocation(location.getLatitude(),
location.getLongitude(), 1);
        String address = addresses.get(0).getAddressLine(0);

        TextView current_address = (TextView) findViewById(R.id.address);
        current_address.setText(address);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}
}

```

- **activity_main.xml**

```

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

    <TextView android:layout_width="wrap_content" android:layout_height="wrap_content"
    android:layout_margin="40dp" android:gravity="center|center_horizontal|center_vertical"
    android:layout_gravity="center|center_horizontal|center_vertical" android:text="LOCATION"

```

```

FINDER" android:textSize="48dp" android:typeface="monospace" android:textStyle="bold"
android:textColor="#000000"/>

    <TextView android:id="@+id/latitude" android:layout_width="wrap_content"
    android:layout_height="50dp" android:layout_marginStart="40dp"
    android:layout_marginTop="50dp" android:gravity="left" android:text="Latitude: "
    android:textSize="32dp" android:typeface="monospace" android:textStyle="bold"
    android:textColor="#55bb22"/>

    <TextView android:id="@+id/longitude" android:layout_width="wrap_content"
    android:layout_height="50dp" android:layout_marginStart="40dp"
    android:layout_marginTop="30dp" android:gravity="left" android:text="Longitude: "
    android:textSize="32dp" android:typeface="monospace" android:textStyle="bold"
    android:textColor="#55bb22"/>

    <TextView android:id="@+id/address" android:layout_width="wrap_content"
    android:layout_height="50dp" android:layout_marginStart="40dp"
    android:layout_marginTop="30dp" android:gravity="left" android:text="Address: "
    android:textSize="32dp" android:typeface="monospace" android:textStyle="bold"
    android:textColor="#55bb22"/>

    <Button android:id="@+id/find" android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:layout_gravity="center"
    android:layout_marginTop="100dp" android:gravity="center" android:text="Find"
    android:textColor="#55bb22" android:textSize="32dp" android:textStyle="bold"
    android:typeface="monospace" android:backgroundTint="#000000"/>

</LinearLayout>

```

Output:

LOCATION FINDER

Latitude: 12.976810

Longitude: 80.221490

Address: Tulive Urbav

Find

Source code:

- **MainActivity.java**

```
package com.example.exercise6b;

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

import android.Manifest;
import android.annotation.SuppressLint;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import java.util.List;
import java.util.Locale;

public class MainActivity extends AppCompatActivity implements LocationListener {

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

        if (ContextCompat.checkSelfPermission(MainActivity.this,
                Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(MainActivity.this, new String[] {
                Manifest.permission.ACCESS_FINE_LOCATION
            }, 100);
        }
        ;

        Button find = (Button) findViewById(R.id.find);
        find.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                findLocation();
            }
        });
    }

    @SuppressLint("MissingPermission")
    private void findLocation() {
        try {
```

```

        LocationManager locationManager = (LocationManager)
this.getSystemService(LOCATION_SERVICE);
        locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 5000, 5,
MainActivity.this);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

@Override
public void onLocationChanged(@NonNull Location location) {
    TextView latitude = (TextView) findViewById(R.id.latitude);
    TextView longitude = (TextView) findViewById(R.id.longitude);
    try {
        Geocoder geocoder = new Geocoder(MainActivity.this, Locale.getDefault());

        TextView current_address = (TextView) findViewById(R.id.address);
        String address_name = current_address.getText().toString();

        List<Address> addresses = geocoder.getFromLocationName(address_name, 1);

        Address address_details = addresses.get(0);

        Double get_latitude = (double) Math.round(address_details.getLatitude() * 100)
/ 100;
        Double get_longitude = (double) Math.round(address_details.getLongitude() *
100) / 100;

        latitude.setText(String.valueOf(get_latitude));
        longitude.setText(String.valueOf(get_longitude));
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}
}

```

- **activity_main.xml**

```

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

    <TextView android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:layout_margin="40dp" android:gravity="center|center_horizontal|center_vertical"
        android:layout_gravity="center|center_horizontal|center_vertical" android:text="LOCATION
FINDER" android:textSize="48dp" android:typeface="monospace" android:textStyle="bold"
        android:textColor="#000000"/>

    <EditText android:id="@+id/address" android:layout_width="325dp"
        android:layout_height="50dp" android:layout_marginStart="40dp"

```

```

        android:layout_marginTop="30dp" android:gravity="left" android:text="Address"
        android:textSize="32dp" android:typeface="monospace" android:textStyle="bold"
        android:textColor="#55bb22"/>

        <TextView android:id="@+id/latitude" android:layout_width="wrap_content"
        android:layout_height="50dp" android:layout_marginStart="40dp"
        android:layout_marginTop="50dp" android:gravity="left" android:text="Latitude: "
        android:textSize="32dp" android:typeface="monospace" android:textStyle="bold"
        android:textColor="#55bb22"/>

        <TextView android:id="@+id/longitude" android:layout_width="wrap_content"
        android:layout_height="50dp" android:layout_marginStart="40dp"
        android:layout_marginTop="30dp" android:gravity="left" android:text="Longitude: "
        android:textSize="32dp" android:typeface="monospace" android:textStyle="bold"
        android:textColor="#55bb22"/>

        <Button android:id="@+id/find" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:layout_gravity="center"
        android:layout_marginTop="100dp" android:gravity="center" android:text="Find"
        android:textColor="#55bb22" android:textSize="32dp" android:textStyle="bold"
        android:typeface="monospace" android:backgroundTint="#000000"/>

</LinearLayout>

```

Output:



Result:

The mobile application was completed successfully

Best Practices:

1. Standard naming conventions

2. Suitable comments
3. Proper indentation
4. Proper user interface which is understandable and easy to navigate
5. Use of modularity and functions

Learning Outcomes:

1. Utilizing the LocationManager and LocationListener for GPS functionality in Android.
2. Requesting and handling runtime permissions for accessing device features.
3. Implementing geocoding to convert addresses into geographical coordinates.
4. Working with UI components such as TextViews, EditText, and Buttons.

SSN COLLEGE OF ENGINEERING, KALAVAKKAM
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UCS1711 - MOBILE APPLICATION DEVELOPMENT LAB

Assignment 7

Name: Jayannthan P T

Dept: CSE 'A'

Roll No.: 205001049

Writing to and Reading from the SD Card

Ex. No:7

Title of the Program:

Develop an android application to read the text from the SD Card and Write into the SD Card. To perform this, create two TextViews one for writing the text and save the text into the SD Card once submit button is clicked and another one for Displaying the text that is retrieved from the SD Card.

Objective:

The objective of the FileReadWrite Android App project is to create an application that allows users to write content to a file and read the contents from the file. The app provides a simple user interface with text fields for entering file contents and file names. It also includes buttons to write to and read from files, along with a separate activity to display the contents read from the file.

Algorithm:

1. Create an Android app with two activities: MainActivity and ReadActivity.
2. MainActivity includes EditTexts for file contents and file name, along with buttons for writing to and reading from files.
3. Implement a method (writeToFile) to write user-provided contents to a file specified by the user.
4. Implement a method (readFromFile) to read contents from a file specified by the user.
5. Display a toast message upon successful writing to a file.
6. Transfer the file name to ReadActivity using an Intent.
7. Read the file contents in ReadActivity and display them in a TextView.

Features used:

1. EditText for user input of file contents and file name.
2. Buttons for triggering write and read operations.
3. File I/O operations for writing and reading files.
4. Toast message for user feedback after writing to a file.
5. Intent to pass data between MainActivity and ReadActivity.
6. TextView for displaying file contents in ReadActivity.

Source code:

- **MainActivity.java**

```
package com.example.exercise7;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        EditText fileContents = (EditText) findViewById(R.id.fileContents);
        EditText fileName = (EditText) findViewById(R.id.fileName);

        Button write = (Button) findViewById(R.id.write);
        write.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String writeFileContents = fileContents.getText().toString();
                String writeFileFileName = fileName.getText().toString();
                writeToFile(writeFileName, writeFileContents);
            }
        });

        Button read = (Button) findViewById(R.id.read);
        read.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String readFileName = fileName.getText().toString();
                Intent i = new Intent(MainActivity.this, ReadActivity.class);
                i.putExtra("fileName", readFileName);
                startActivity(i);
            }
        });
    }
}
```

```

private void writeToFile(String writeFileName, String writeFileContents) {
    File path = getApplicationContext().getFilesDir();

    try {
        FileOutputStream writer = new FileOutputStream(new File(path, writeFileName));
        writer.write(writeFileContents.getBytes());
        writer.close();
        Toast.makeText(getApplicationContext(), "Written to file: " + path + "/" +
writeFileName,
                Toast.LENGTH_SHORT).show();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

- **activity_main.xml**

```

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

    <EditText android:id="@+id/fileContents" android:layout_width="300dp"
    android:layout_height="50dp" android:layout_gravity="center"
    android:layout_marginTop="150dp" android:text="File Contents" android:textSize="24sp"
    android:typeface="monospace" android:textStyle="bold" android:textColor="#000000" />
    <EditText android:id="@+id/fileName" android:layout_width="300dp"
    android:layout_height="50dp" android:layout_gravity="center"
    android:layout_marginTop="50dp" android:text="File Name" android:textSize="24sp"
    android:typeface="monospace" android:textStyle="bold" android:textColor="#000000" />

    <Button android:id="@+id/write" android:layout_width="250dp"
    android:layout_height="50dp" android:layout_gravity="center"
    android:layout_marginTop="75dp" android:text="Write to File" android:textSize="20sp"
    android:typeface="monospace" android:textStyle="bold" />

    <Button android:id="@+id/read" android:layout_width="250dp"
    android:layout_height="50dp" android:layout_gravity="center"
    android:layout_marginTop="50dp" android:text="Read from File" android:textSize="20sp"
    android:typeface="monospace" android:textStyle="bold" />

</LinearLayout>

```

- **ReadActivity.java**

```

package com.example.exercise7;

import androidx.appcompat.app.AppCompatActivity;

```

```

import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;

import java.io.File;
import java.io.FileInputStream;

public class ReadActivity extends AppCompatActivity {

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

        TextView contentsFromFile = (TextView) findViewById(R.id.contentsFromFile);

        Intent i = getIntent();

        String readFileName = i.getStringExtra("fileName");
        String readFileContents = readFromFile(readFileName);
        contentsFromFile.setText(readFileContents);
    }

    private String readFromFile(String readFileName) {
        File path = getApplicationContext().getFilesDir();
        File readFrom = new File(path + "/" + readFileName);
        byte[] content = new byte[(int) readFrom.length()];

        try {
            FileInputStream reader = new FileInputStream(readFrom);
            reader.read(content);
            return new String(content);
        } catch (Exception e) {
            e.printStackTrace();
            return e.toString();
        }
    }
}

```

- `activity_read_data.xml`

```

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

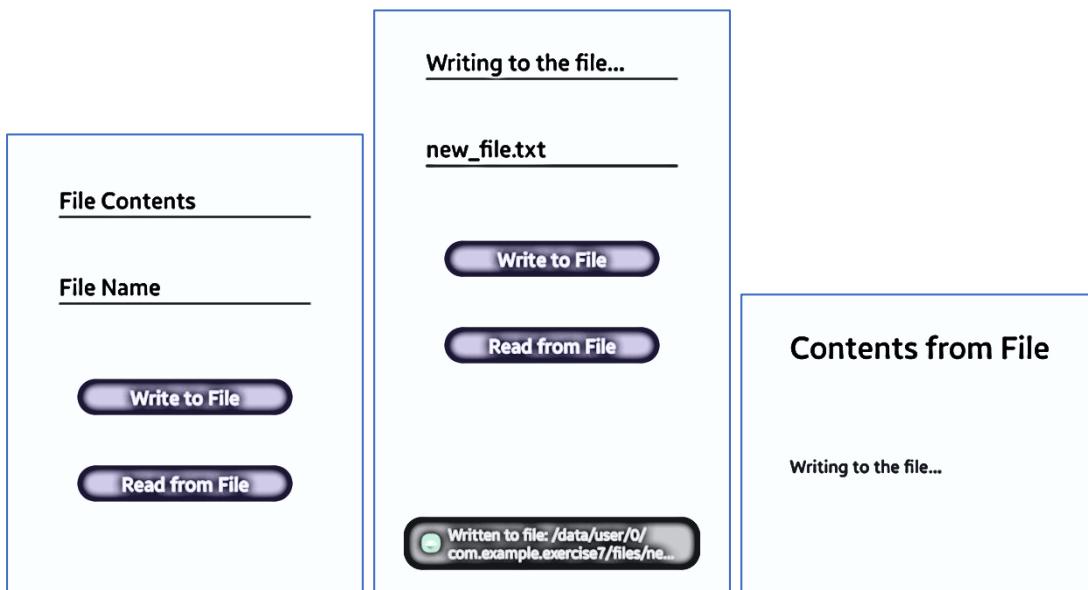
    <TextView android:layout_width="400dp" android:layout_height="50dp"
        android:layout_gravity="center" android:layout_marginTop="100dp" android:text="Contents
        from File" android:gravity="center" android:textSize="32sp" android:typeface="monospace"
        android:textStyle="bold" android:textColor="#000000" />

```

```
<TextView android:id="@+id/contentsFromFile" android:layout_width="300dp"
    android:layout_height="200dp" android:layout_gravity="center"
    android:layout_marginTop="100dp" android:text="" android:textSize="18sp"
    android:typeface="monospace" android:textStyle="bold" />

</LinearLayout>
```

Output:



Result:

The mobile application was completed successfully

Best Practices:

1. Use meaningful variable names and comments for code clarity.
2. Implement error handling for file operations to address potential exceptions.
3. Provide user-friendly toast messages for successful or unsuccessful file operations.
4. Utilize separate activities for distinct functionalities to maintain a clean code structure.
5. Follow Android coding conventions and design guidelines for consistency.
6. Test the application on various devices and Android versions to ensure compatibility.
7. Consider implementing additional error handling and user prompts for improved robustness.

Learning Outcomes:

1. Understanding file read and write operations in Android.
2. Implementing activities, intents, and passing data between activities.
3. Handling user input through EditText and triggering actions with buttons.
4. Displaying information using TextView.
5. Gaining familiarity with Android layout design using LinearLayout.
6. Handling exceptions related to file I/O operations.
7. Implementing best practices for a clean and maintainable codebase.

SSN COLLEGE OF ENGINEERING, KALAVAKKAM
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UCS1711 - MOBILE APPLICATION DEVELOPMENT LAB

Assignment 8

Name: Jayannthan P T

Dept: CSE 'A'

Roll No.: 205001049

SMS Sending and Notification

Ex. No:8

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

Objective:

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

Algorithm:

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

Features used:

1. SMSManager to send SMS messages.

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

Source code:

- **MainActivity.java**

```
package com.example.exercise8;

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

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

public class MainActivity extends AppCompatActivity {

    IntentFilter intentFilter;

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

```
};

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

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

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

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

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

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

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

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

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

@Override
protected void onResume() {
```

```

        registerReceiver(intentReceiver, intentFilter);
        super.onResume();
    }

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

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

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

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

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

```

- **activity_main.xml**

```

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

```

```

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

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

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

            </LinearLayout>

        </ScrollView>

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

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

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

        </LinearLayout>

    </LinearLayout>

```

- **ReceiveActivity.java**

```

package com.example.exercise8;

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

public class MessageReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        Bundle bundle = intent.getExtras();

```

```

SmsMessage[] messages;
String str = "";

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

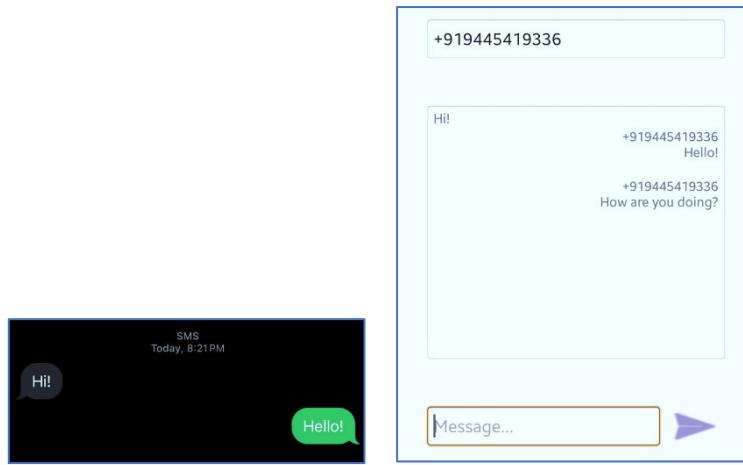
Toast.makeText(context, str, Toast.LENGTH_SHORT).show();

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

```

Output:





Result:

The mobile application was completed successfully

Best Practices:

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

Learning Outcomes:

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