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:

Ex. No:1 **Date:**26/8/2023

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.

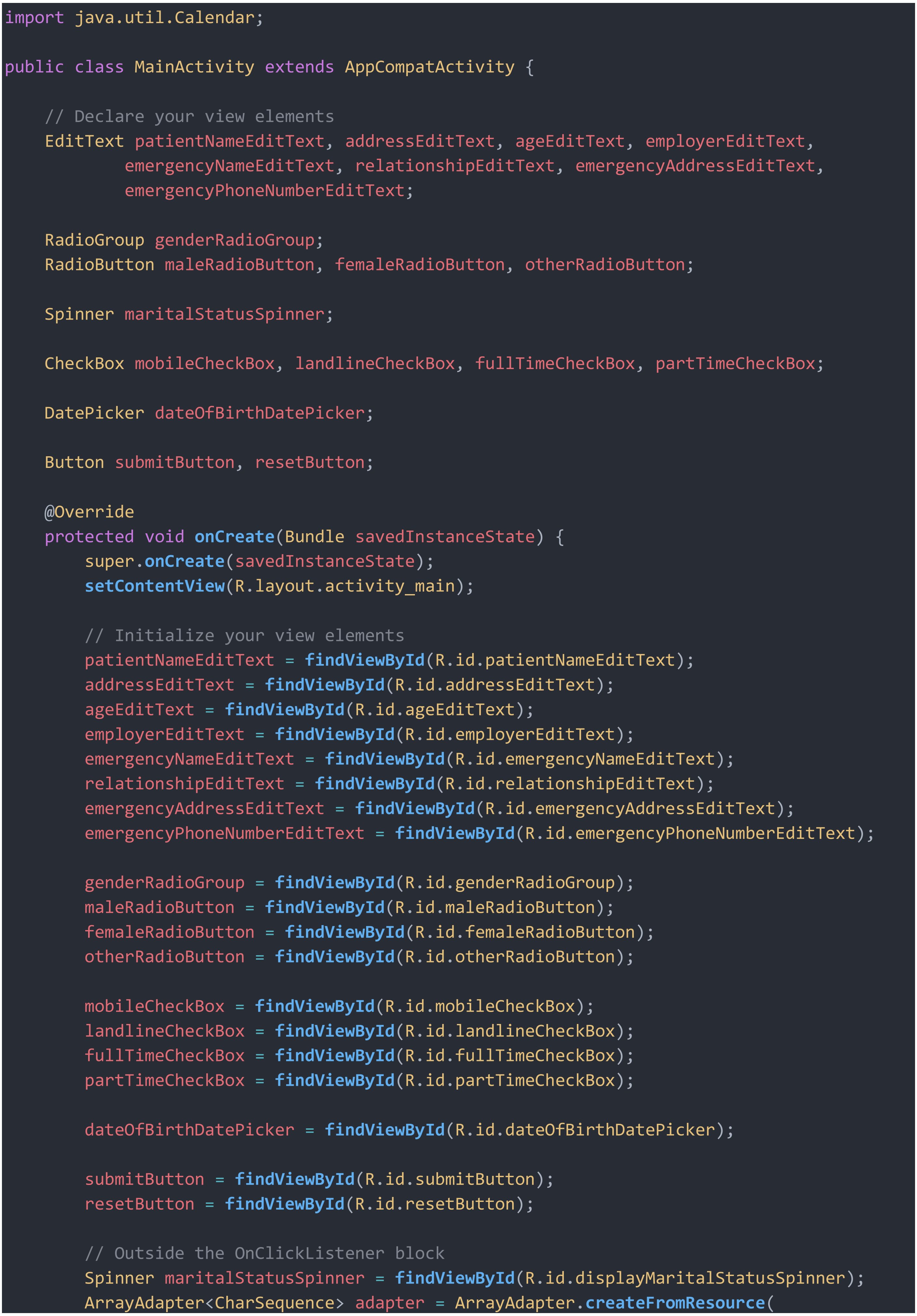
Features used:

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.

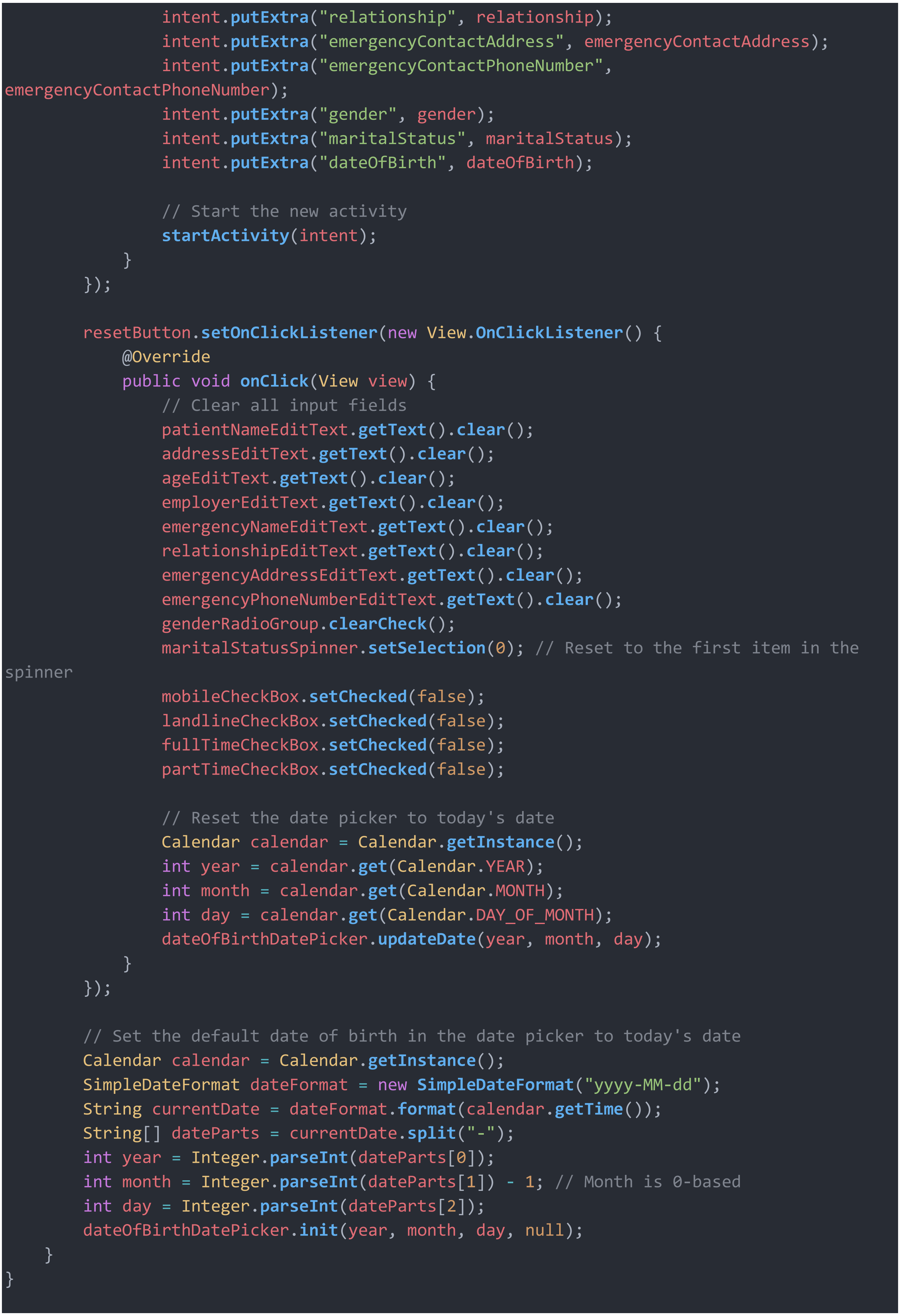
Source code:

- `MainActivity.java`

```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;
```



```java
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(
                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"/>
```

```
                <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>
```

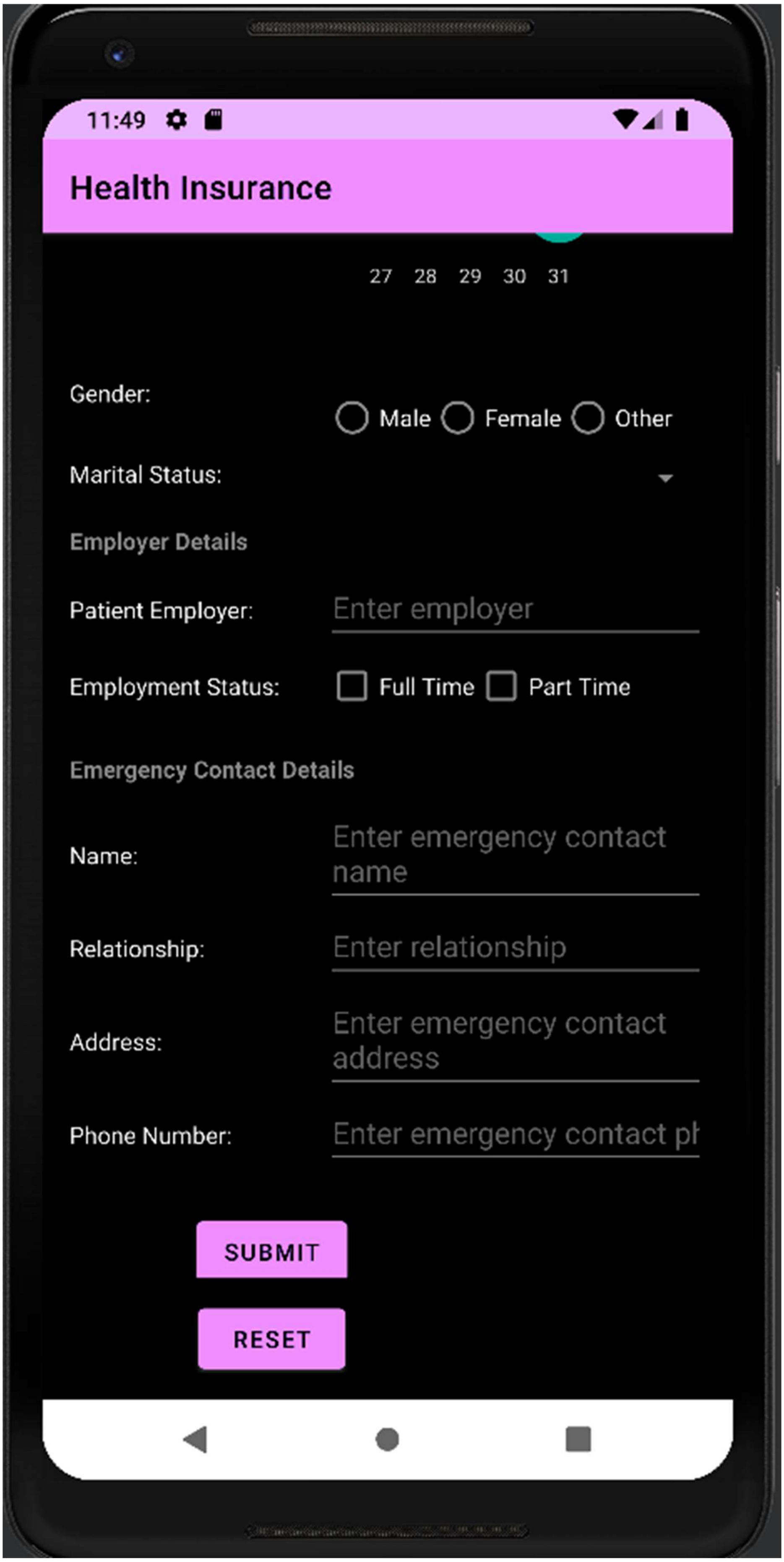
- DisplayDataActivity.java

```
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:



11:49

Health Insurance

27 28 29 30 31

Gender: ○ Male ○ Female ○ Other

Marital Status:

Employer Details

Patient Employer: Enter employer

Employment Status: □ Full Time □ Part Time

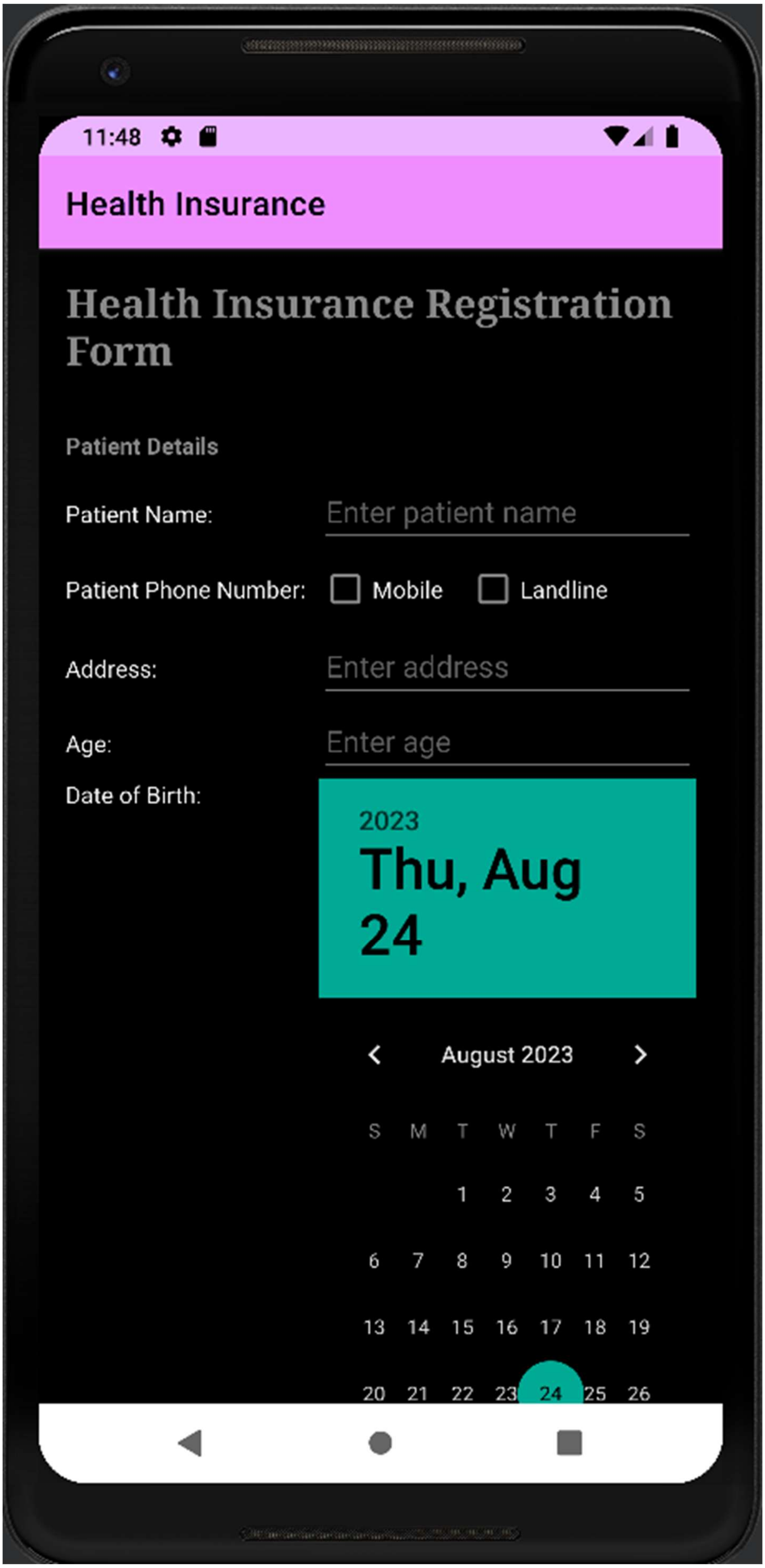
Emergency Contact Details

Name: Enter emergency contact name

Relationship: Enter relationship

Address: Enter emergency contact address

Phone Number: Enter emergency contact pl



Health Insurance

Health Insurance Registration Form

Patient Details

Patient Name: Enter patient name

Patient Phone Number: □ Mobile □ Landline

Address: Enter address

Age: Enter age

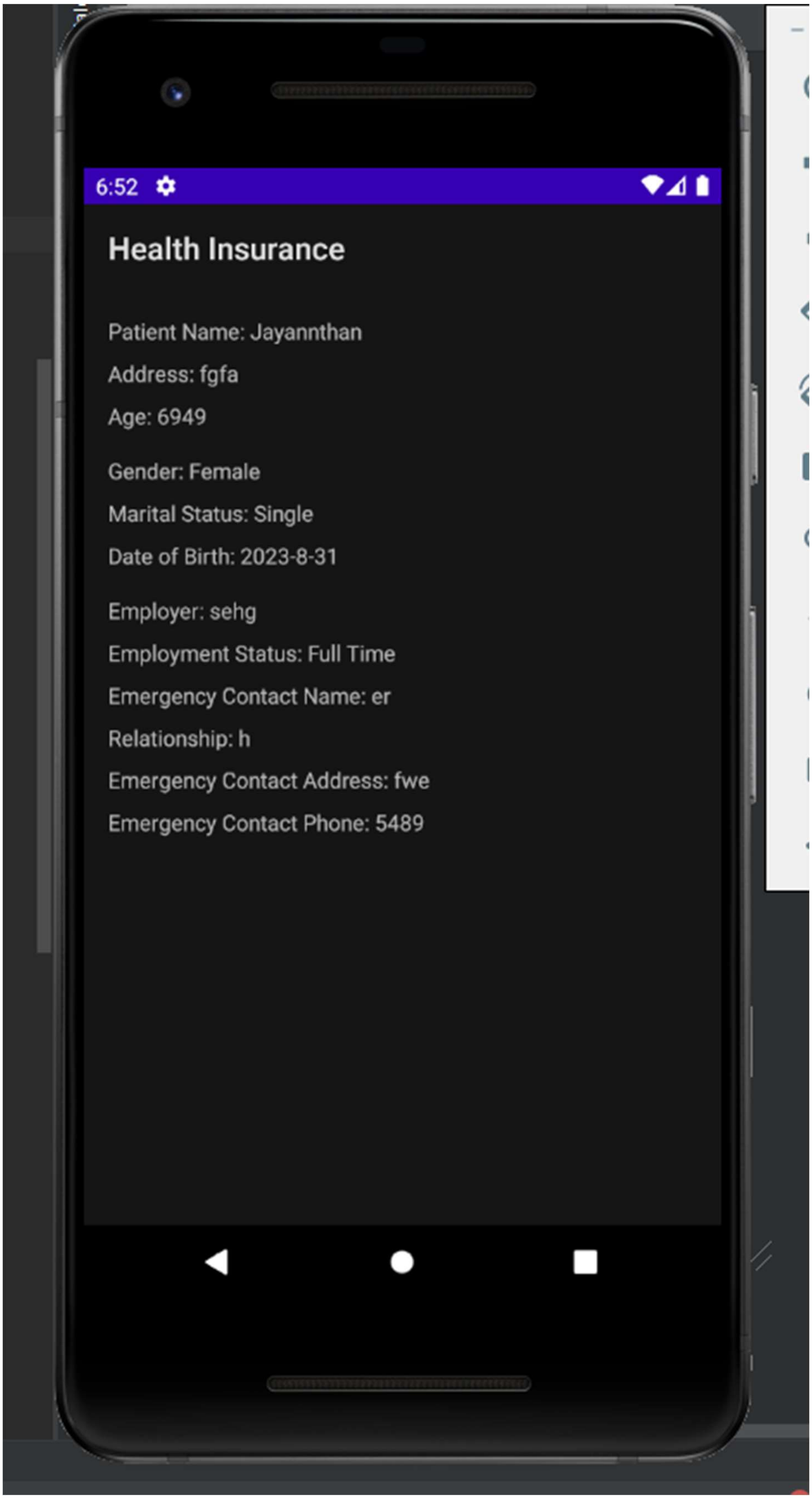
Date of Birth:

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



6:52

Health Insurance

Patient Name: Jayannthan

Address: fgfa

Age: 6949

Gender: Female

Marital Status: Single

Date of Birth: 2023-8-31

Employer: sehg

Employment Status: Full Time

Emergency Contact Name: er

Relationship: h

Emergency Contact Address: fwe

Emergency Contact Phone: 5489

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

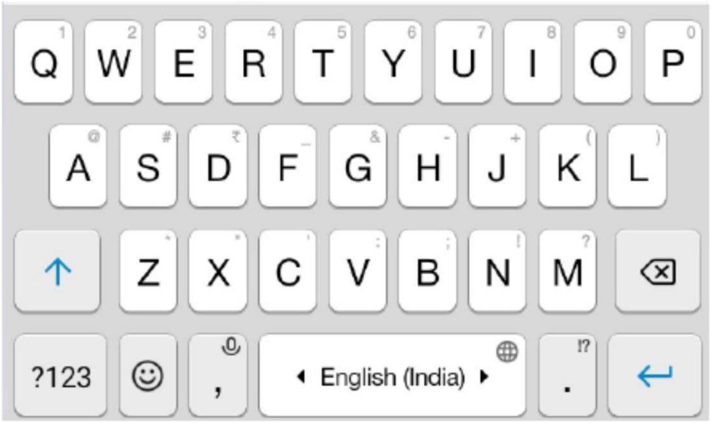
Name: Jayannthan P T *Dept: CSE 'A'* *Roll No.: 205001049*

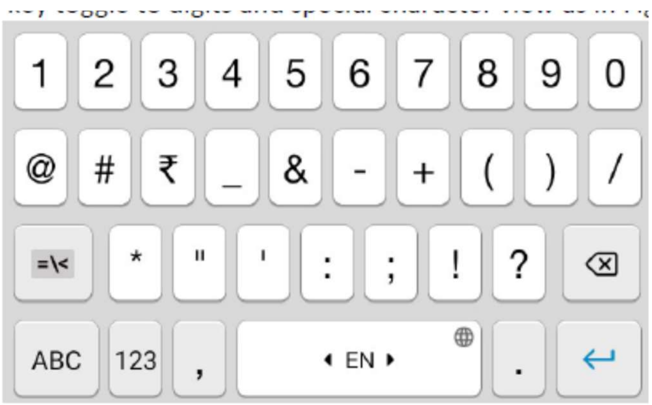
Create an Android mobile application which simulates a virtual keyboard with following features use intents to navigate between multiple activities.

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.

Ex. No:2 **Date:**5/9/2023

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.

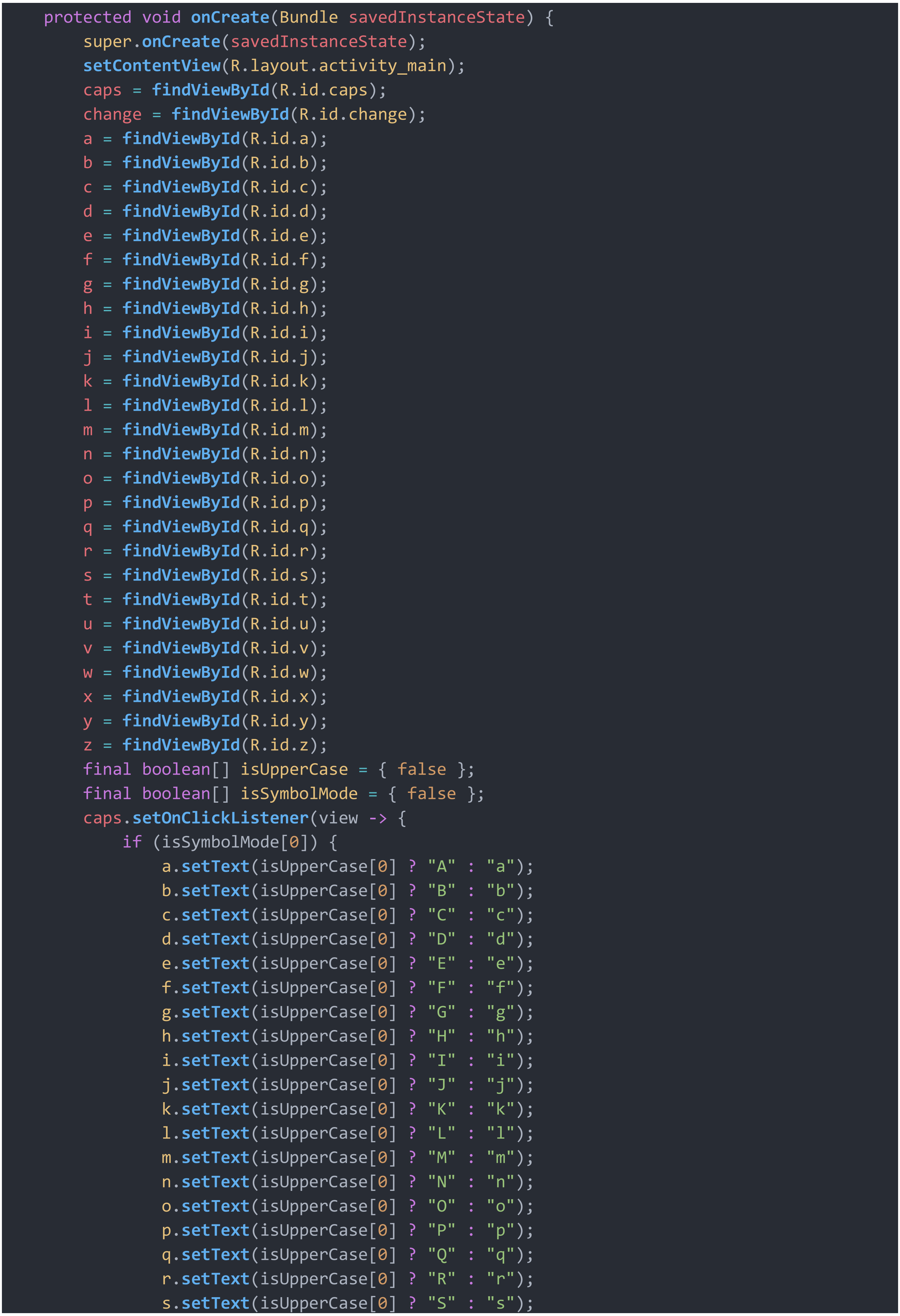
Features used:

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.

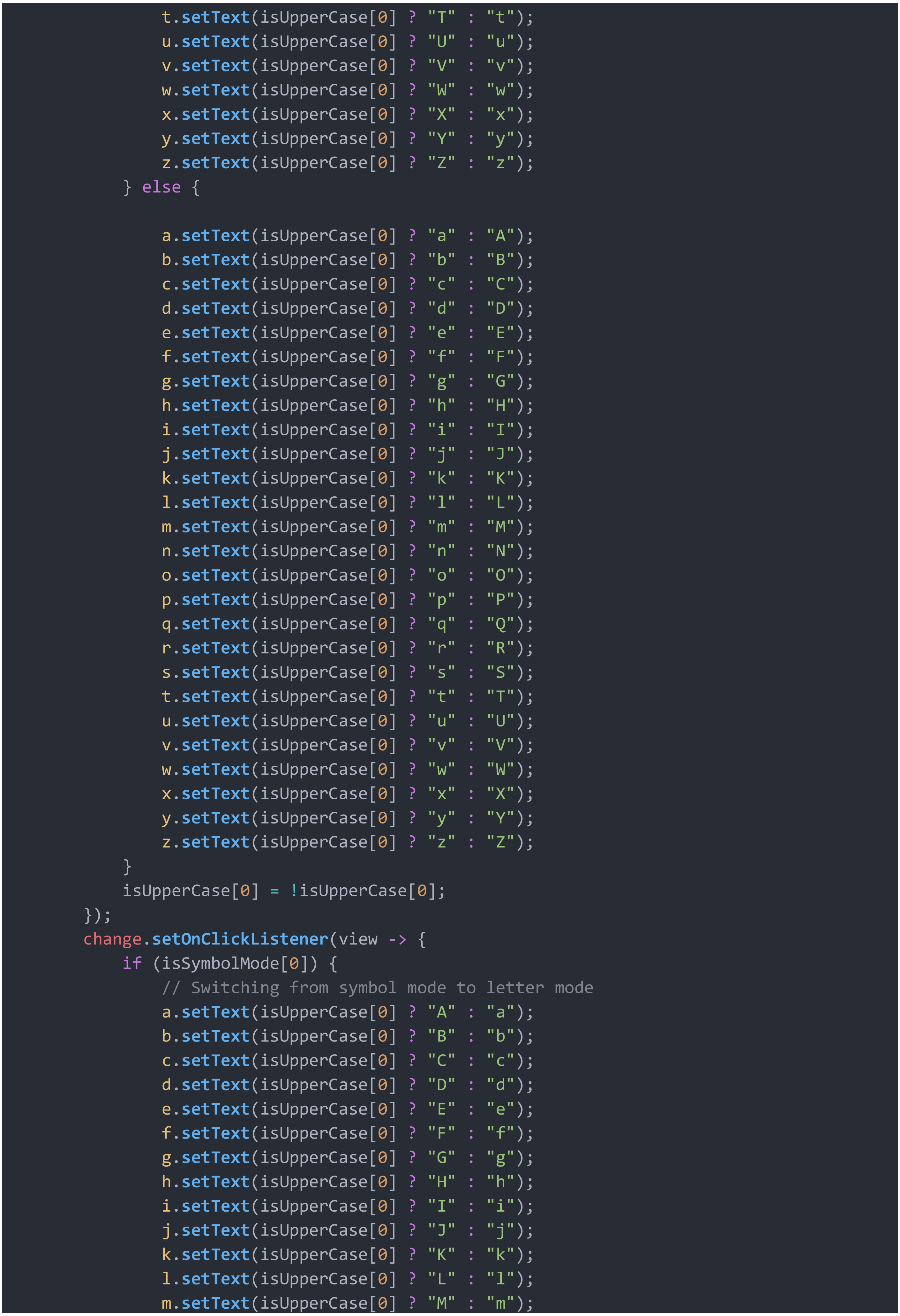
Source code:

- `MainActivity.java`

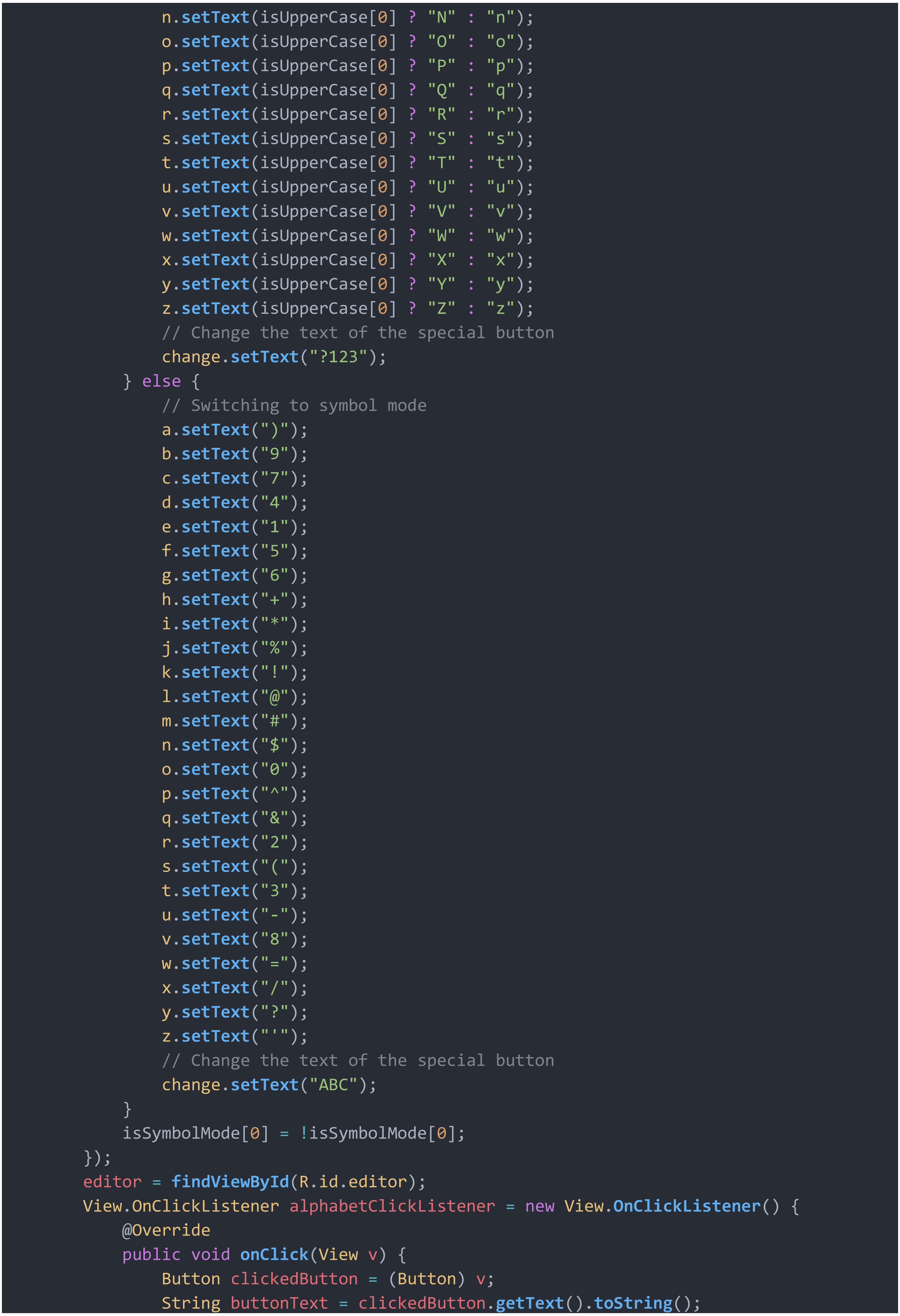
```java
package com.example.keyboard;

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
```



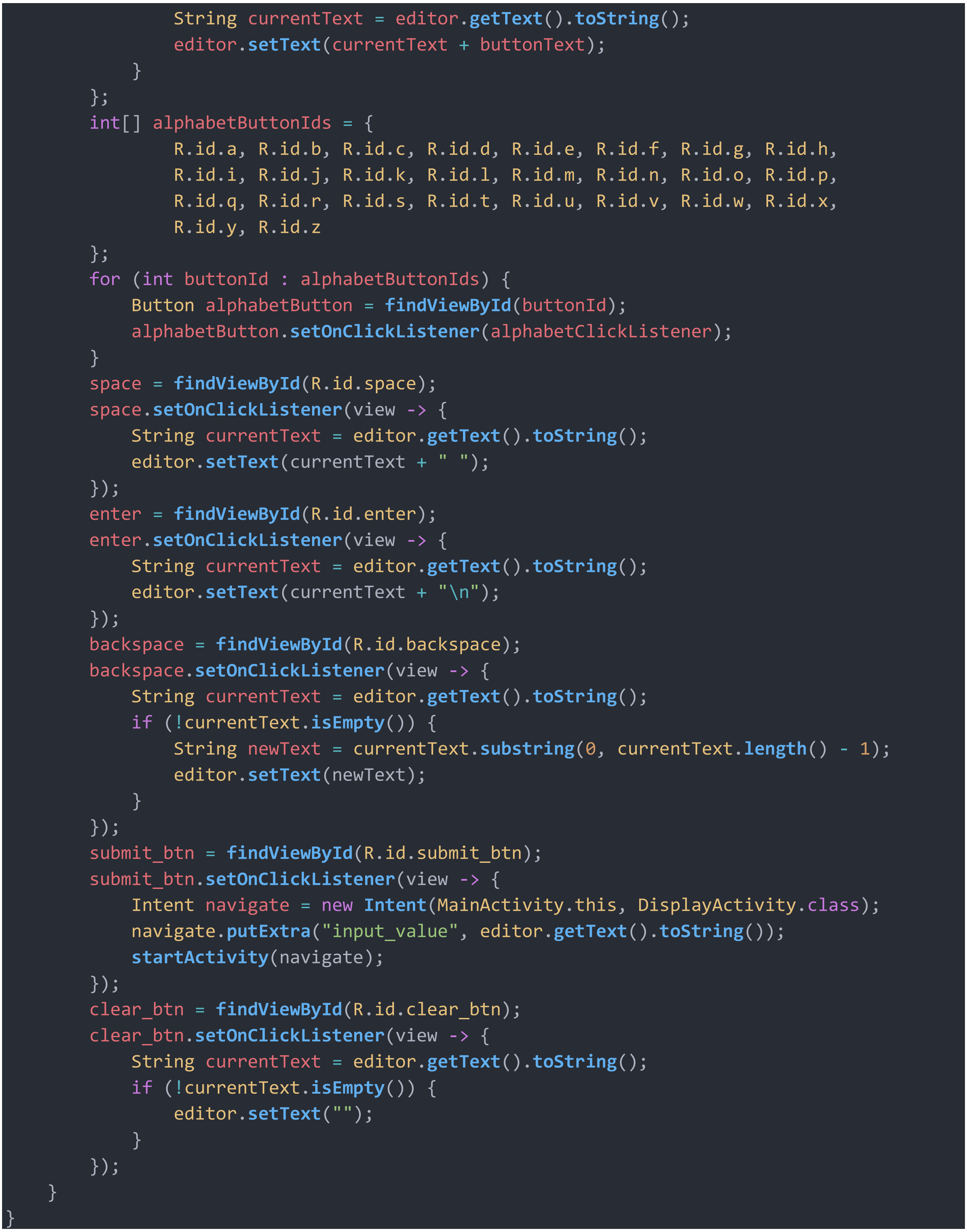
```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        caps = findViewById(R.id.caps);
        change = findViewById(R.id.change);
        a = findViewById(R.id.a);
        b = findViewById(R.id.b);
        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");
```



```
                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");
                // Change the text of the special button
                change.setText("?123");
            } else {
                // Switching to symbol mode
                a.setText(")");
                b.setText("9");
                c.setText("7");
                d.setText("4");
                e.setText("1");
                f.setText("5");
                g.setText("6");
                h.setText("+");
                i.setText("*");
                j.setText("%");
                k.setText("!");
                l.setText("@");
                m.setText("#");
                n.setText("$");
                o.setText("0");
                p.setText("^");
                q.setText("&");
                r.setText("2");
                s.setText("(");
                t.setText("3");
                u.setText("-");
                v.setText("8");
                w.setText("=");
                x.setText("/");
                y.setText("?");
                z.setText("'");
                // Change the text of the special button
                change.setText("ABC");
            }
            isSymbolMode[0] = !isSymbolMode[0];
        });
        editor = findViewById(R.id.editor);
        View.OnClickListener alphabetClickListener = new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Button clickedButton = (Button) v;
                String buttonText = clickedButton.getText().toString();
```



```
                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" />
```

```
                <Button android:id="@+id/s" 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="s" />
                <Button android:id="@+id/d" 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="d" />
                <Button android:id="@+id/f" 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="f" />
                <Button android:id="@+id/g" 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="g" />
                <Button android:id="@+id/h" 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="h" />
                <Button android:id="@+id/j" 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="j" />
            </TableRow>
            <TableRow android:layout_gravity="center">
                <Button android:id="@+id/z" 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="z" />
                <Button android:id="@+id/x" 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="x" />
                <Button android:id="@+id/c" 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="c" />
                <Button android:id="@+id/v" 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="v" />
                <Button android:id="@+id/b" 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="b" />
                <Button android:id="@+id/n" 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="n" />
                <Button android:id="@+id/m" 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="m" />
```

```
            </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>
```

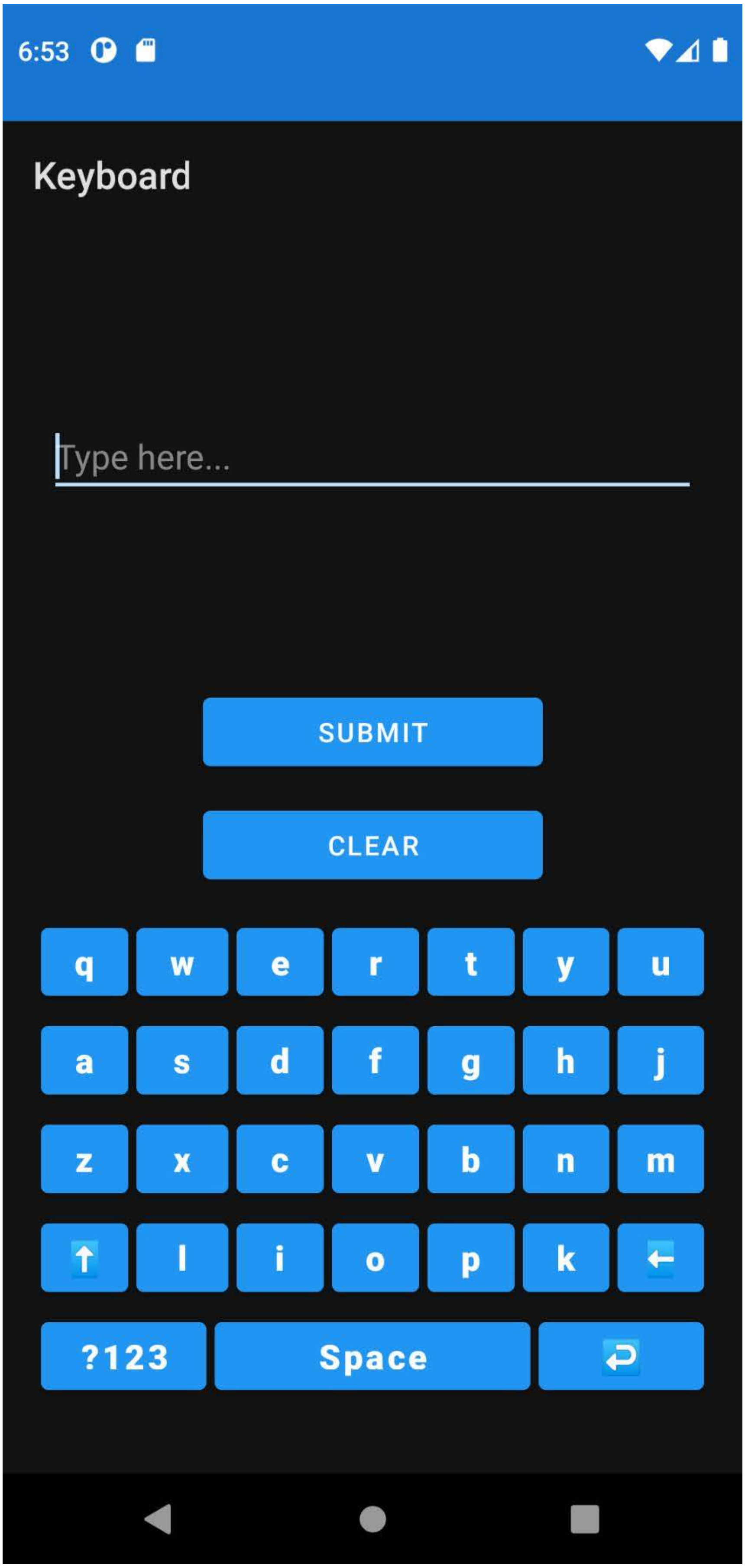
- DisplayDataActivity.java

```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);
    }
}
```

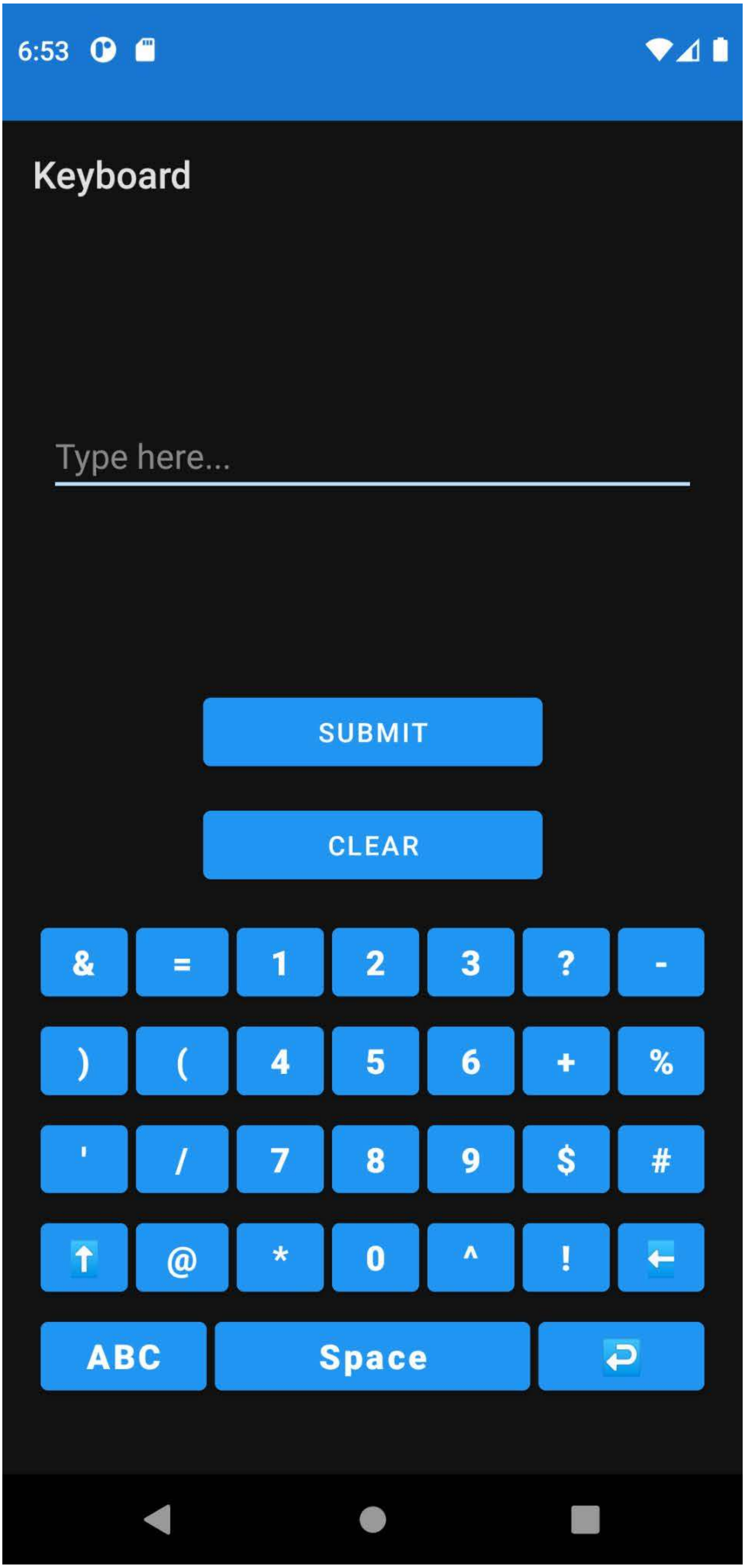
- activity_display_data.xml

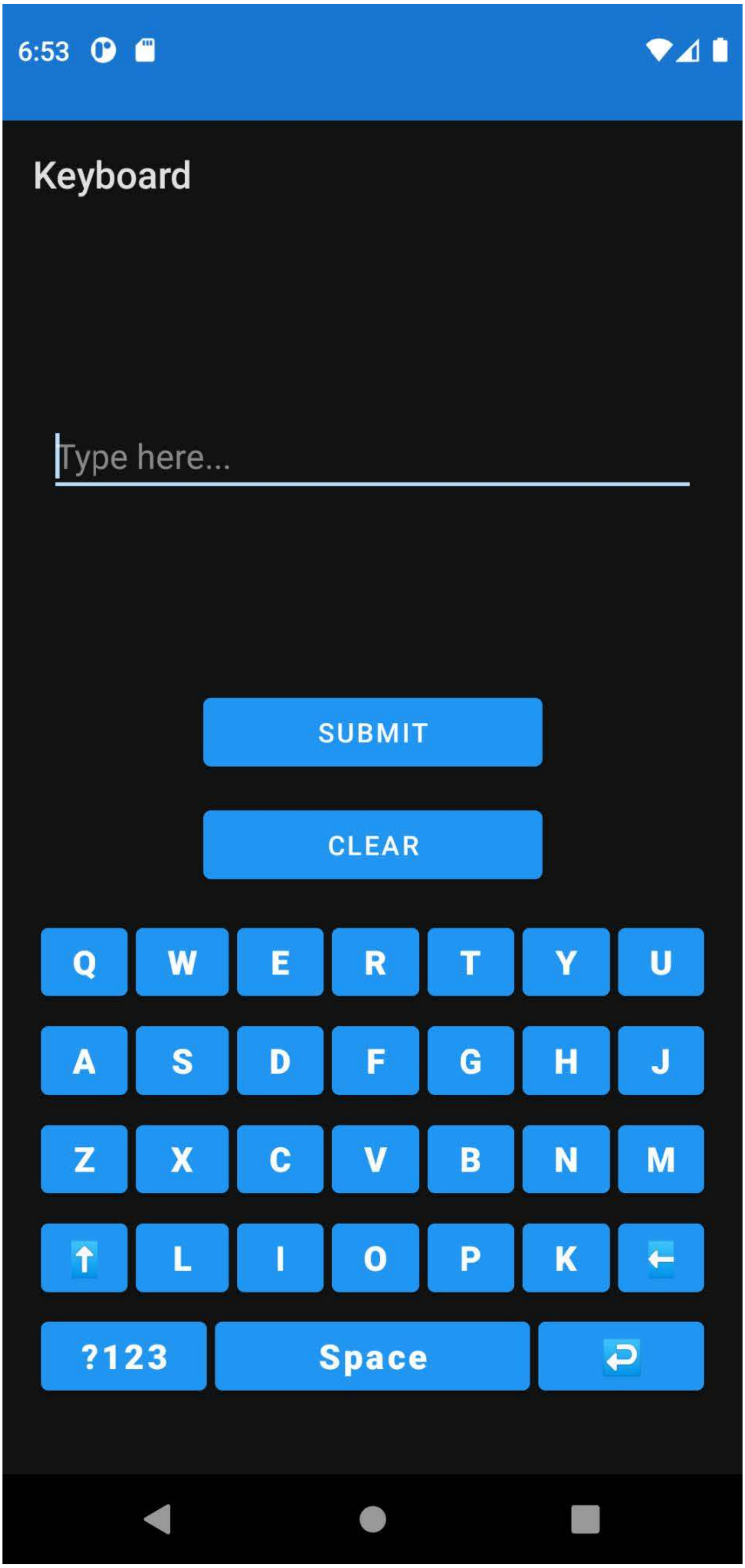
```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent" android:layout_height="match_parent">
    <TextView android:id="@+id/title"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


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.

Ex. No:3 **Date:**16/9/2023

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.

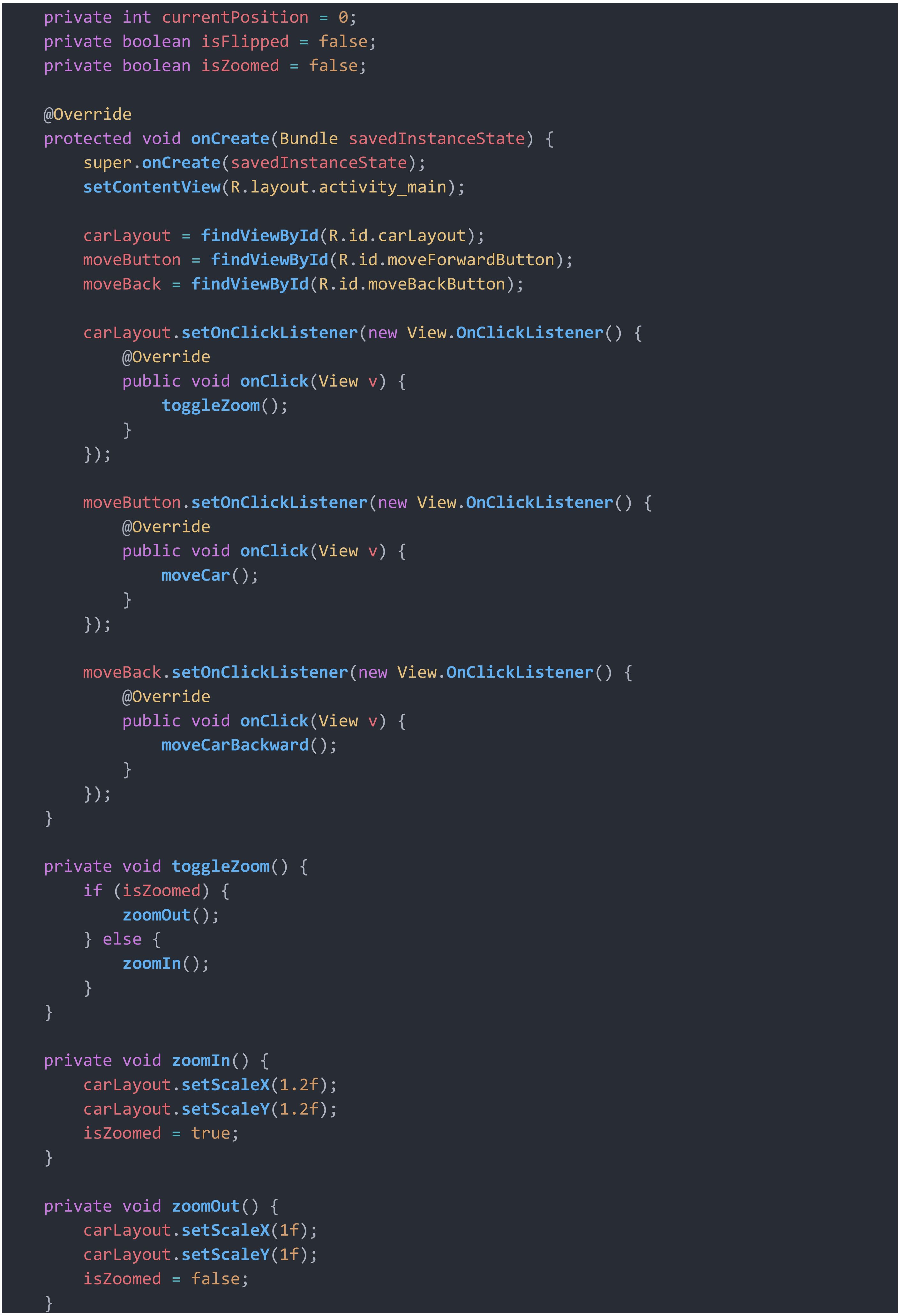
Features used:

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`

```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;
```



```java
    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>
```

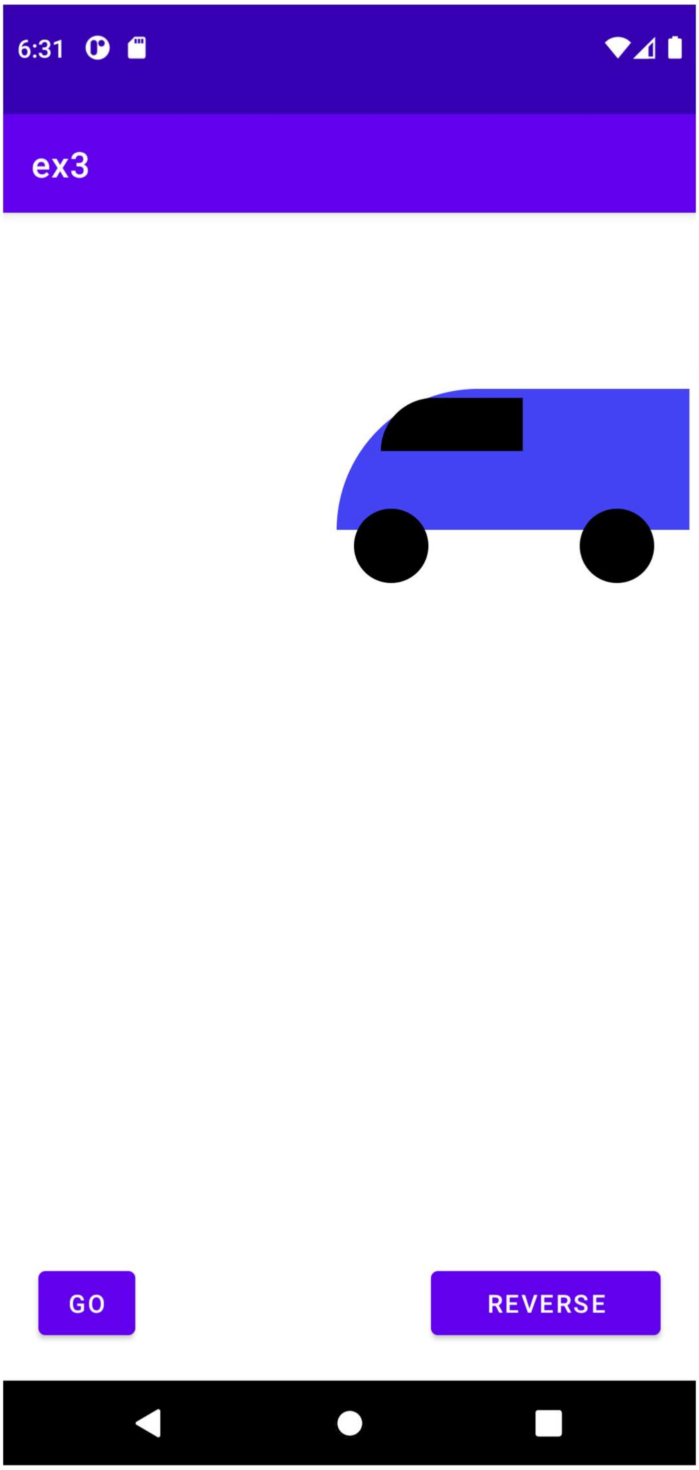
```
<shape xmlns:android="http://schemas.android.com/apk/res/android"
android:shape="rectangle">
    <gradient android:drawable="@drawable/car_body_gradient" android:type="linear"
android:angle="0" />
    <solid android:color="#4444f4" />
    <size android:width="200dp" android:height="80dp" />
    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

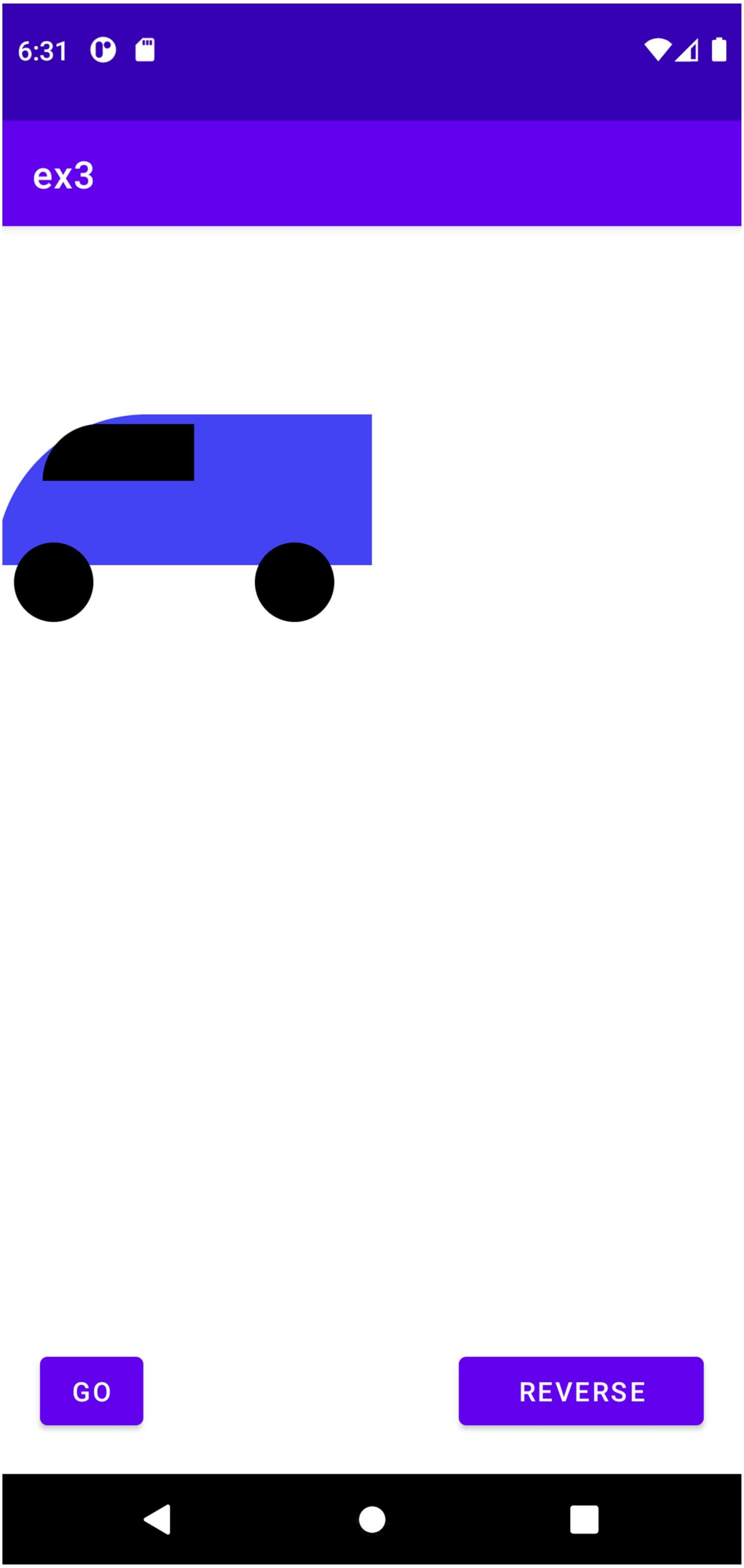
```
<shape xmlns:android="http://schemas.android.com/apk/res/android" android:shape="oval">
    <solid android:color="#000000" />
    <size android:width="42dp" android:height="42dp" />
```

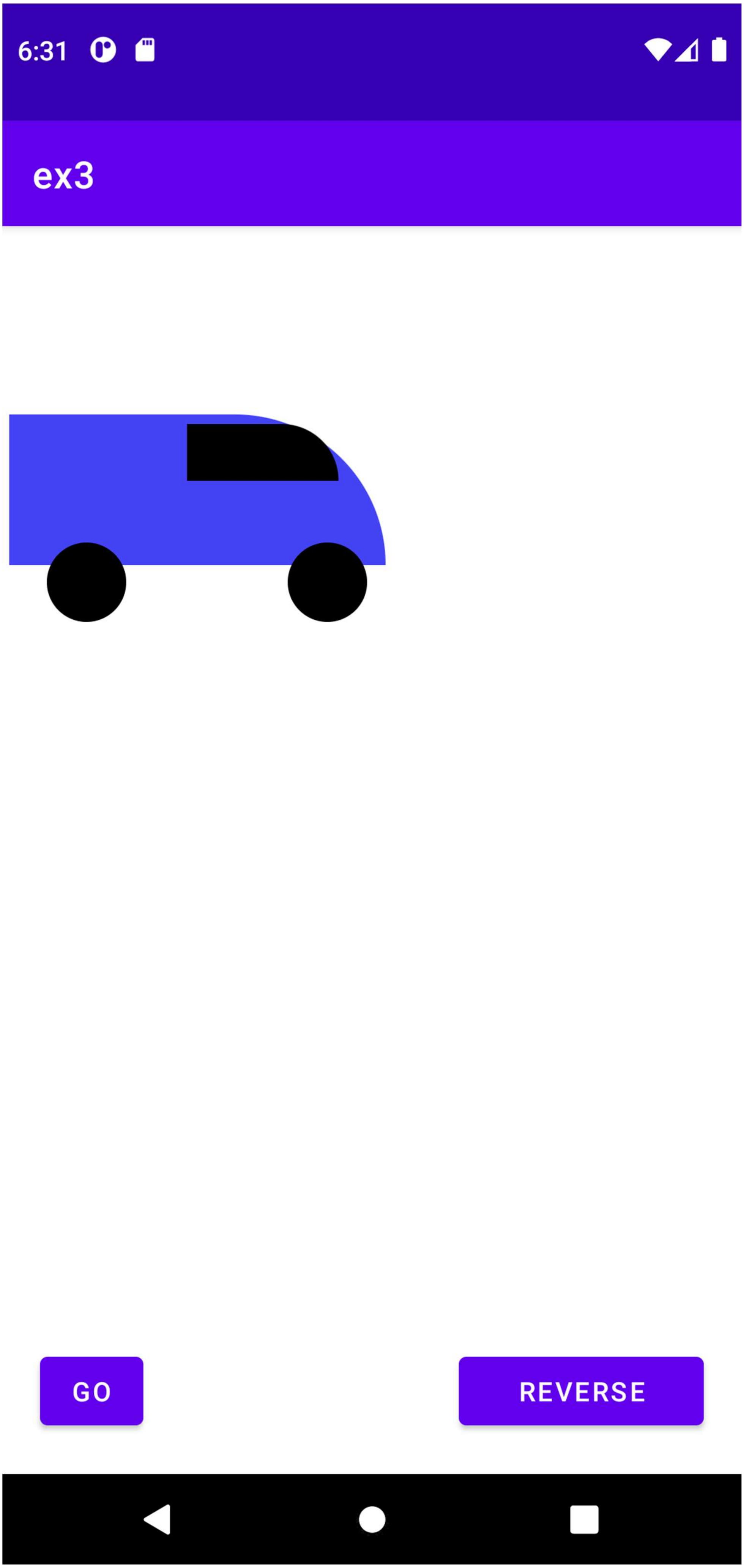
```
</shape>
```

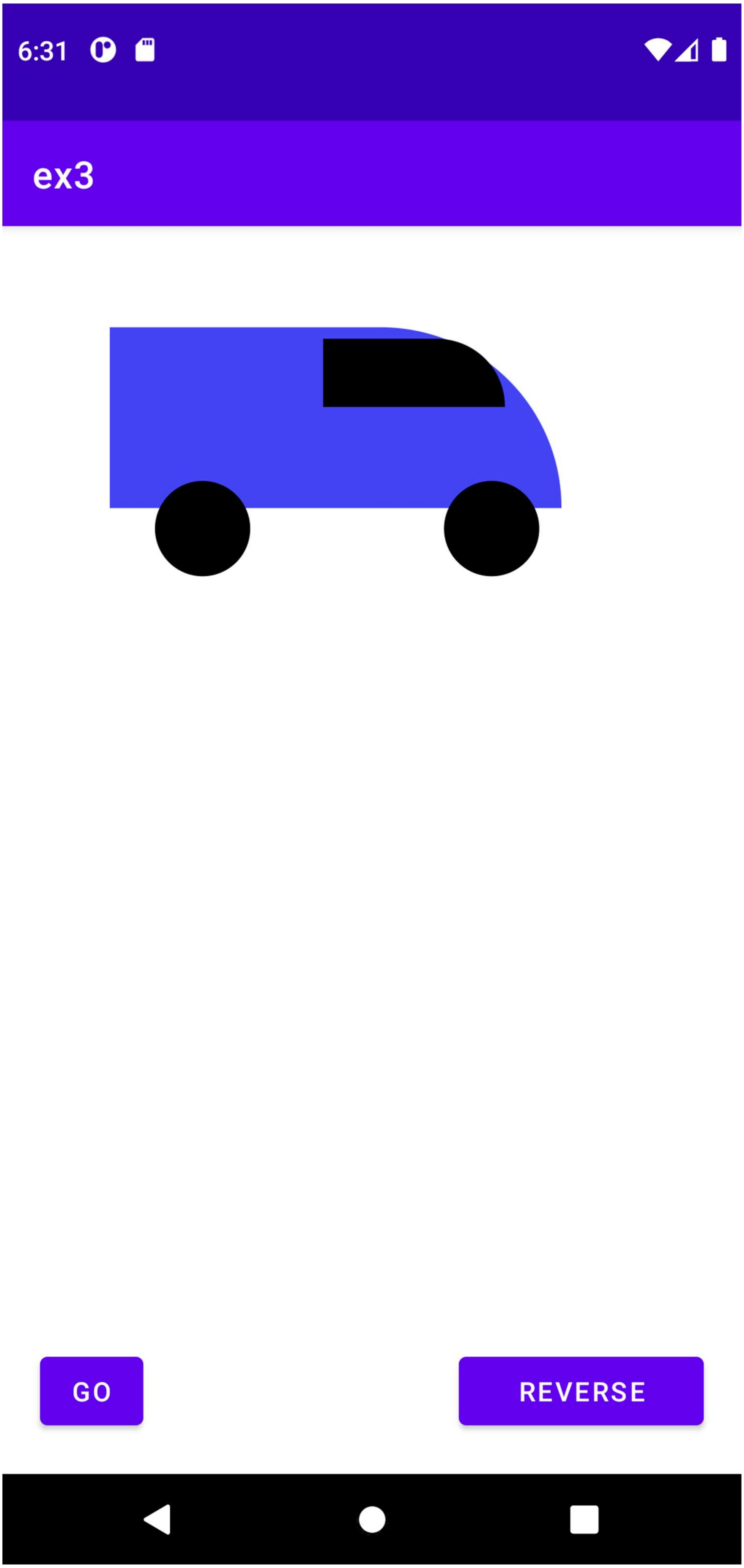
```
<shape xmlns:android="http://schemas.android.com/apk/res/android"
android:shape="rectangle">
    <solid android:color="#000000" />
    <size android:width="160dp" android:height="60dp" />
    <corners android:topRightRadius="70dp" />
</shape>
```

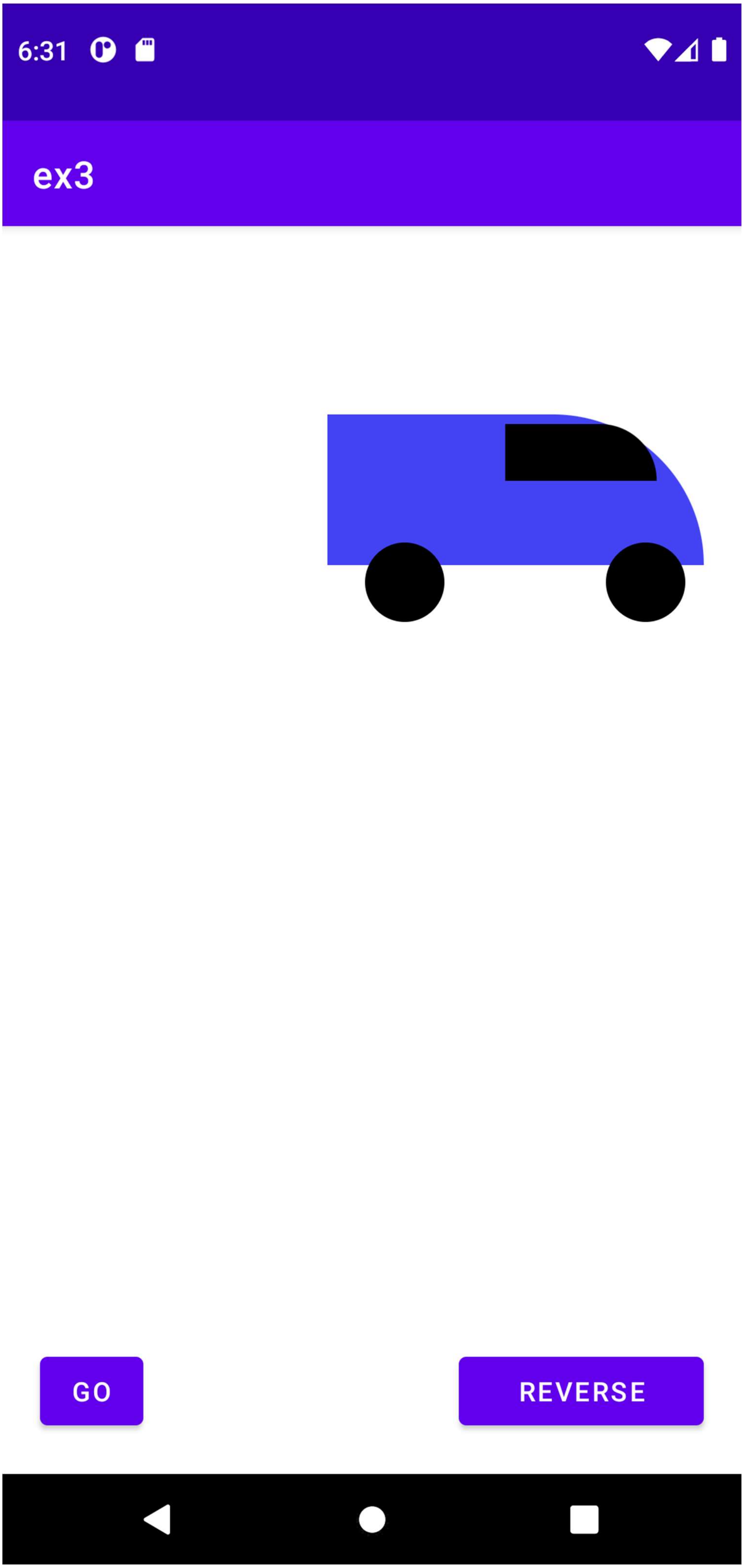
Output:




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.