(2024A) COMP-5450-AA - Mobile Midterm

Aayush Parekh |1215791

https://github.com/aayushparekhh/Midterm

QUESTION 1 – CALCULATOR APP

I. UI XML Code

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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"
    android:layout_margin="0dp"
    android:padding="0dp"
    tools:context=".MainActivity">

    <TableLayout
        android:id="@+id/tableLayout"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:stretchColumns="0,1,2,3"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent">

        <TableRow
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <Button
                android:id="@+id/button_clear"
                style="@style/Widget.AppCompat.Button.Borderless"
                android:layout_width="@dimen/box_size"
                android:layout_height="@dimen/box_size"
                android:background="@color/colorAccent"
                android:onClick="onClearClick"
                android:text="@string/button_clear"

android:textAppearance="@style/TextAppearance.AppCompat.Large" />
```

```
            <Button
                android:id="@+id/button_root"
                style="@style/Widget.AppCompat.Button.Borderless"
                android:layout_width="@dimen/box_size"
                android:layout_height="@dimen/box_size"
                android:background="@color/colorAccent"
                android:onClick="onOperatorClick"
                android:text="@string/button_root"

android:textAppearance="@style/TextAppearance.AppCompat.Large" />

            <Button
                android:id="@+id/button_pow"
                style="@style/Widget.AppCompat.Button.Borderless"
                android:layout_width="@dimen/box_size"
                android:layout_height="@dimen/box_size"
                android:background="@color/colorAccent"
                android:onClick="onOperatorClick"
                android:text="@string/button_pow"

android:textAppearance="@style/TextAppearance.AppCompat.Large" />

            <Button
                android:id="@+id/button_plus"
                style="@style/Widget.AppCompat.Button.Borderless"
                android:layout_width="@dimen/box_size"
                android:layout_height="@dimen/box_size"
                android:background="@color/colorAccent"
                android:onClick="onOperatorClick"
                android:text="@string/button_plus"

android:textAppearance="@style/TextAppearance.AppCompat.Large" />
        </TableRow>

        <TableRow
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <Button
                android:id="@+id/button_seven"
                style="@style/Widget.AppCompat.Button.Borderless"
                android:layout_width="@dimen/box_size"
                android:layout_height="@dimen/box_size"
                android:background="@drawable/button_bg_stroke"
                android:onClick="onNumClick"
                android:text="@string/button_7"

android:textAppearance="@style/TextAppearance.AppCompat.Large" />

            <Button
                android:id="@+id/button_eight"
                style="@style/Widget.AppCompat.Button.Borderless"
```

```
                android:layout_width="@dimen/box_size"
                android:layout_height="@dimen/box_size"
                android:background="@drawable/button_bg_stroke"
                android:onClick="onNumClick"
                android:text="@string/button_8"

android:textAppearance="@style/TextAppearance.AppCompat.Large" />

            <Button
                android:id="@+id/button_nine"
                style="@style/Widget.AppCompat.Button.Borderless"
                android:layout_width="@dimen/box_size"
                android:layout_height="@dimen/box_size"
                android:background="@drawable/button_bg_stroke"
                android:onClick="onNumClick"
                android:text="@string/button_9"

android:textAppearance="@style/TextAppearance.AppCompat.Large" />

            <Button
                android:id="@+id/button_minus"
                style="@style/Widget.AppCompat.Button.Borderless"
                android:layout_width="@dimen/box_size"
                android:layout_height="@dimen/box_size"
                android:background="@color/colorAccent"
                android:onClick="onOperatorClick"
                android:text="@string/button_minus"

android:textAppearance="@style/TextAppearance.AppCompat.Large" />
        </TableRow>

        <TableRow
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <Button
                android:id="@+id/button_four"
                style="@style/Widget.AppCompat.Button.Borderless"
                android:layout_width="@dimen/box_size"
                android:layout_height="@dimen/box_size"
                android:background="@drawable/button_bg_stroke"
                android:onClick="onNumClick"
                android:text="@string/button_4"

android:textAppearance="@style/TextAppearance.AppCompat.Large" />

            <Button
                android:id="@+id/button_five"
                style="@style/Widget.AppCompat.Button.Borderless"
                android:layout_width="@dimen/box_size"
                android:layout_height="@dimen/box_size"
                android:background="@drawable/button_bg_stroke"
```

```
                android:onClick="onNumClick"
                android:text="@string/button_5"

android:textAppearance="@style/TextAppearance.AppCompat.Large" />

            <Button
                android:id="@+id/button_six"
                style="@style/Widget.AppCompat.Button.Borderless"
                android:layout_width="@dimen/box_size"
                android:layout_height="@dimen/box_size"
                android:background="@drawable/button_bg_stroke"
                android:onClick="onNumClick"
                android:text="@string/button_6"

android:textAppearance="@style/TextAppearance.AppCompat.Large" />

            <Button
                android:id="@+id/button_multiply"
                style="@style/Widget.AppCompat.Button.Borderless"
                android:layout_width="@dimen/box_size"
                android:layout_height="@dimen/box_size"
                android:background="@color/colorAccent"
                android:onClick="onOperatorClick"
                android:text="@string/button_mul"

android:textAppearance="@style/TextAppearance.AppCompat.Large" />
        </TableRow>

        <TableRow
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <Button
                android:id="@+id/button_one"
                style="@style/Widget.AppCompat.Button.Borderless"
                android:layout_width="@dimen/box_size"
                android:layout_height="@dimen/box_size"
                android:background="@drawable/button_bg_stroke"
                android:onClick="onNumClick"
                android:text="@string/button_1"

android:textAppearance="@style/TextAppearance.AppCompat.Large" />

            <Button
                android:id="@+id/button_two"
                style="@style/Widget.AppCompat.Button.Borderless"
                android:layout_width="@dimen/box_size"
                android:layout_height="@dimen/box_size"
                android:background="@drawable/button_bg_stroke"
                android:onClick="onNumClick"
                android:text="@string/button_2"
```

```
android:textAppearance="@style/TextAppearance.AppCompat.Large" />

            <Button
                android:id="@+id/button_three"
                style="@style/Widget.AppCompat.Button.Borderless"
                android:layout_width="@dimen/box_size"
                android:layout_height="@dimen/box_size"
                android:background="@drawable/button_bg_stroke"
                android:onClick="onNumClick"
                android:text="@string/button_3"

android:textAppearance="@style/TextAppearance.AppCompat.Large" />

            <Button
                android:id="@+id/button_divide"
                style="@style/Widget.AppCompat.Button.Borderless"
                android:layout_width="@dimen/box_size"
                android:layout_height="@dimen/box_size"
                android:background="@color/colorAccent"
                android:onClick="onOperatorClick"
                android:text="@string/button_div"

android:textAppearance="@style/TextAppearance.AppCompat.Large" />
        </TableRow>

        <TableRow
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <Button
                android:id="@+id/button_dot"
                style="@style/Widget.AppCompat.Button.Borderless"
                android:layout_width="@dimen/box_size"
                android:layout_height="@dimen/box_size"
                android:background="@drawable/button_bg_stroke"
                android:onClick="onNumClick"
                android:text="@string/button_dot"

android:textAppearance="@style/TextAppearance.AppCompat.Large" />

            <Button
                android:id="@+id/button_zero"
                style="@style/Widget.AppCompat.Button.Borderless"
                android:layout_width="@dimen/box_size"
                android:layout_height="@dimen/box_size"
                android:background="@drawable/button_bg_stroke"
                android:onClick="onNumClick"
                android:text="@string/button_0"

android:textAppearance="@style/TextAppearance.AppCompat.Large" />
```

```
            <Button
                android:id="@+id/button_equals"
                style="@style/Widget.AppCompat.Button.Borderless"
                android:layout_width="@dimen/box_size"
                android:layout_height="@dimen/box_size"
                android:background="@color/colorPrimaryDark"
                android:onClick="onEqualsClick"
                android:text="@string/button_equals"

android:textAppearance="@style/TextAppearance.AppCompat.Large"
                android:textColor="@android:color/white" />

            <Button
                android:id="@+id/button_mod"
                style="@style/Widget.AppCompat.Button.Borderless"
                android:layout_width="@dimen/box_size"
                android:layout_height="@dimen/box_size"
                android:background="@color/colorAccent"
                android:onClick="onOperatorClick"
                android:text="@string/button_mod"

android:textAppearance="@style/TextAppearance.AppCompat.Large" />
        </TableRow>
    </TableLayout>

    <ImageButton
        android:id="@+id/button_backspace"
        style="@style/Widget.AppCompat.Button.Borderless"
        android:layout_width="52dp"
        android:layout_height="40dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:contentDescription="@string/backspace"
        android:onClick="onBackspaceClick"
        android:src="@drawable/ic_backspace_black_24dp"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        app:layout_constraintBottom_toTopOf="@+id/tableLayout"
        app:layout_constraintEnd_toEndOf="parent" />

    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:gravity="end"
        android:orientation="horizontal"
        app:layout_constraintBottom_toTopOf="@+id/button_backspace"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent">

        <TextView
            android:id="@+id/input_value_1"
```

```
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginBottom="8dp"
            android:layout_marginEnd="8dp"
            android:baselineAligned="false"
            android:ellipsize="end"
            android:maxLength="12"
            android:maxLines="1"
            android:textAlignment="textEnd"

android:textAppearance="@style/TextAppearance.AppCompat.Display1"
            android:textColor="@android:color/black"

app:layout_constraintBottom_toTopOf="@+id/button_backspace"
            app:layout_constraintEnd_toStartOf="@+id/input_operation"
/>

        <TextView
            android:id="@+id/input_operation"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginBottom="8dp"
            android:layout_marginEnd="8dp"
            android:baselineAligned="false"
            android:ellipsize="end"
            android:maxLines="1"
            android:maxLength="1"
            android:paddingLeft="2dp"
            android:paddingRight="2dp"
            android:textAlignment="center"

android:textAppearance="@style/TextAppearance.AppCompat.Display1"
            android:textColor="@color/colorPrimaryDark"

app:layout_constraintBottom_toTopOf="@+id/button_backspace"
            app:layout_constraintEnd_toStartOf="@+id/input_value_2" />

        <TextView
            android:id="@+id/input_value_2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginBottom="8dp"
            android:layout_marginEnd="24dp"
            android:baselineAligned="false"
            android:ellipsize="end"
            android:gravity="end"
            android:maxLength="12"
            android:maxLines="1"
            android:textAlignment="textEnd"

android:textAppearance="@style/TextAppearance.AppCompat.Display1"
            android:textColor="@android:color/black"
```

```
app:layout_constraintBottom_toTopOf="@+id/button_backspace"
            app:layout_constraintEnd_toEndOf="parent" />

    </LinearLayout>

    <LinearLayout
        android:id="@+id/linearLayout"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_weight="1"
        android:gravity="end"
        android:orientation="vertical"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <TextView
            android:id="@+id/textView_result"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginEnd="8dp"
            android:layout_marginTop="8dp"
            android:gravity="end"
            android:maxLines="1"
            android:textAlignment="viewEnd"

android:textAppearance="@style/TextAppearance.AppCompat.Display2"
            android:textColor="@android:color/black"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintTop_toTopOf="parent" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:gravity="end"
        android:orientation="vertical"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/linearLayout">

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

```
            android:layout_marginEnd="8dp"
            android:layout_marginTop="8dp"
            android:gravity="end"
            android:maxLines="1"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/textView_result"
/>
    </LinearLayout>

</android.support.constraint.ConstraintLayout>aa</androidx.constraintl
ayout.widget.ConstraintLayout>
```

II. Main Activity Kotlin File

```
package me.mahakagg.calculator;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;

/*
 *  Author: Mahak Aggarwal
 *  Last date modified: 26-09-2018
 *  (C), All rights reserved, Mahak Aggarwal 2018
 *
 *  App Icon -
 *  credits to Font Awesome and MDBootstrap for the application logo
 *  Link - https://mdbootstrap.com/tools/logo-generator-icons/
 * */

public class MainActivity extends AppCompatActivity {
    private Calculator mCalculator; // object of Calculator class
    private TextView mInputValue1TextView; // for TextView ID -
input_value_1
    private TextView mInputValue2TextView; // for TextView ID -
input_value_2
    private TextView mOperatorTextView; // for TextView ID -
input_operation
    private TextView mFinalResultTextView; // for TextView ID -
textView_result
```

```
    private TextView mCompleteOperation; // for TextView ID -
complete_operation
    private double number_one; // first number
    private double number_two; // second number
    private String operation_string; // current operation
    private static int MAX_CHARACTERS = 10;

    private enum operator {
        ADD, SUB, MUL, DIV, MOD, ROOT, POW, NULL
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mCalculator = new Calculator();
        mInputValue1TextView = findViewById(R.id.input_value_1); //
first number
        mInputValue2TextView = findViewById(R.id.input_value_2); //
second number
        mOperatorTextView = findViewById(R.id.input_operation);
//operation
        mFinalResultTextView = findViewById(R.id.textView_result); //
final result
        mCompleteOperation = findViewById(R.id.complete_operation); //
string containing the numbers and the operation
        operation_string = operator.NULL.name();

//        Implementation of saved instance state
        if (savedInstanceState != null) {

mInputValue1TextView.setText(savedInstanceState.getString("First_numbe
r", ""));

mInputValue2TextView.setText(savedInstanceState.getString("Second_numb
er", ""));

mOperatorTextView.setText(savedInstanceState.getString("Operation",
""));

mFinalResultTextView.setText(savedInstanceState.getString("Final_resul
t", ""));

mCompleteOperation.setText(savedInstanceState.getString("Complete_oper
ation", ""));
            operation_string =
savedInstanceState.getString("Operation_string",
operator.NULL.name());
            number_one = savedInstanceState.getDouble("Number_one",
0);
            number_two = savedInstanceState.getDouble("Number_two",
0);
```

```
        }
    }

    //    implementation of Saved instance state
    @Override
    protected void onSaveInstanceState(Bundle outState) {
        super.onSaveInstanceState(outState);
        outState.putString("First_number",
mInputValue1TextView.getText().toString());
        outState.putString("Operation",
mOperatorTextView.getText().toString());
        outState.putString("Second_number",
mInputValue2TextView.getText().toString());
        outState.putString("Final_result",
mFinalResultTextView.getText().toString());
        outState.putString("Complete_operation",
mCompleteOperation.getText().toString());
        outState.putString("Operation_string", operation_string);
        outState.putDouble("Number_one", number_one);
        outState.putDouble("Number_two", number_two);
    }

    //    method to check where to append the numbers (first or second
number)
    private void selectTextViewToAppend(String number) {
        if (operation_string.equals(operator.NULL.name())) {
            if
(mInputValue1TextView.getText().toString().contains(".")) {
                MAX_CHARACTERS++;
            }
            if (mInputValue1TextView.getText().length() <
MAX_CHARACTERS) {
                mInputValue1TextView.append(number);
                MAX_CHARACTERS = 10;
            } else {
                MAX_CHARACTERS = 10;
                Toast.makeText(this, "Cannot have more than 10
numbers", Toast.LENGTH_LONG).show();
            }
        } else {
            if
(mInputValue1TextView.getText().toString().contains(".")) {
                MAX_CHARACTERS++;
            }
            if (mInputValue2TextView.getText().length() <
MAX_CHARACTERS) {
                mInputValue2TextView.append(number);
                MAX_CHARACTERS = 10;
            } else {
                MAX_CHARACTERS = 10;
                Toast.makeText(this, "Cannot have more than 10
numbers", Toast.LENGTH_LONG).show();
```

```
            }

        }
    }

    //            handle operations for numbers
    public void onNumClick(View view) {
        if (!mFinalResultTextView.getText().toString().equals("")) {
            onClearClick(view);
        }
        switch (view.getId()) {
            case R.id.button_one:
                selectTextViewToAppend("1");
                break;
            case R.id.button_two:
                selectTextViewToAppend("2");
                break;
            case R.id.button_three:
                selectTextViewToAppend("3");
                break;
            case R.id.button_four:
                selectTextViewToAppend("4");
                break;
            case R.id.button_five:
                selectTextViewToAppend("5");
                break;
            case R.id.button_six:
                selectTextViewToAppend("6");
                break;
            case R.id.button_seven:
                selectTextViewToAppend("7");
                break;
            case R.id.button_eight:
                selectTextViewToAppend("8");
                break;
            case R.id.button_nine:
                selectTextViewToAppend("9");
                break;
            case R.id.button_zero:
                selectTextViewToAppend("0");
                break;
            case R.id.button_dot:
                if (operation_string.equals(operator.NULL.name())) {
                    if
(mInputValue1TextView.getText().toString().contains(".")) {
                        Toast.makeText(this, "Cannot have more than
one decimal point in a number", Toast.LENGTH_LONG).show();
                    } else {
                        mInputValue1TextView.append(".");
                    }
                } else {
```

```
                    if
(mInputValue2TextView.getText().toString().contains(".")) {
                        Toast.makeText(this, "Cannot have more than
one decimal point in a number", Toast.LENGTH_LONG).show();
                    } else {
                        mInputValue2TextView.append(".");
                    }
                }
                break;
            default:
                break;
        }
    }

    //            handle operations for operators
    public void onOperatorClick(View view) {
        if (!mInputValue1TextView.getText().toString().equals("")) {
            switch (view.getId()) {
                case R.id.button_plus:
                    operation_string = operator.ADD.name();
                    mOperatorTextView.setText("+");
                    break;
                case R.id.button_minus:
                    operation_string = operator.SUB.name();
                    mOperatorTextView.setText("-");
                    break;
                case R.id.button_multiply:
                    operation_string = operator.MUL.name();
                    mOperatorTextView.setText("x");
                    break;
                case R.id.button_divide:
                    operation_string = operator.DIV.name();
                    mOperatorTextView.setText("/");
                    break;
                case R.id.button_mod:
                    operation_string = operator.MOD.name();
                    mOperatorTextView.setText("%");
                    break;
                case R.id.button_pow:
                    operation_string = operator.POW.name();
                    mOperatorTextView.setText("^");
                    break;
                case R.id.button_root:
                    operation_string = operator.ROOT.name();
                    mOperatorTextView.setText("√");
                    break;
                default:
                    operation_string = operator.NULL.name();
                    break;
            }

        }
```

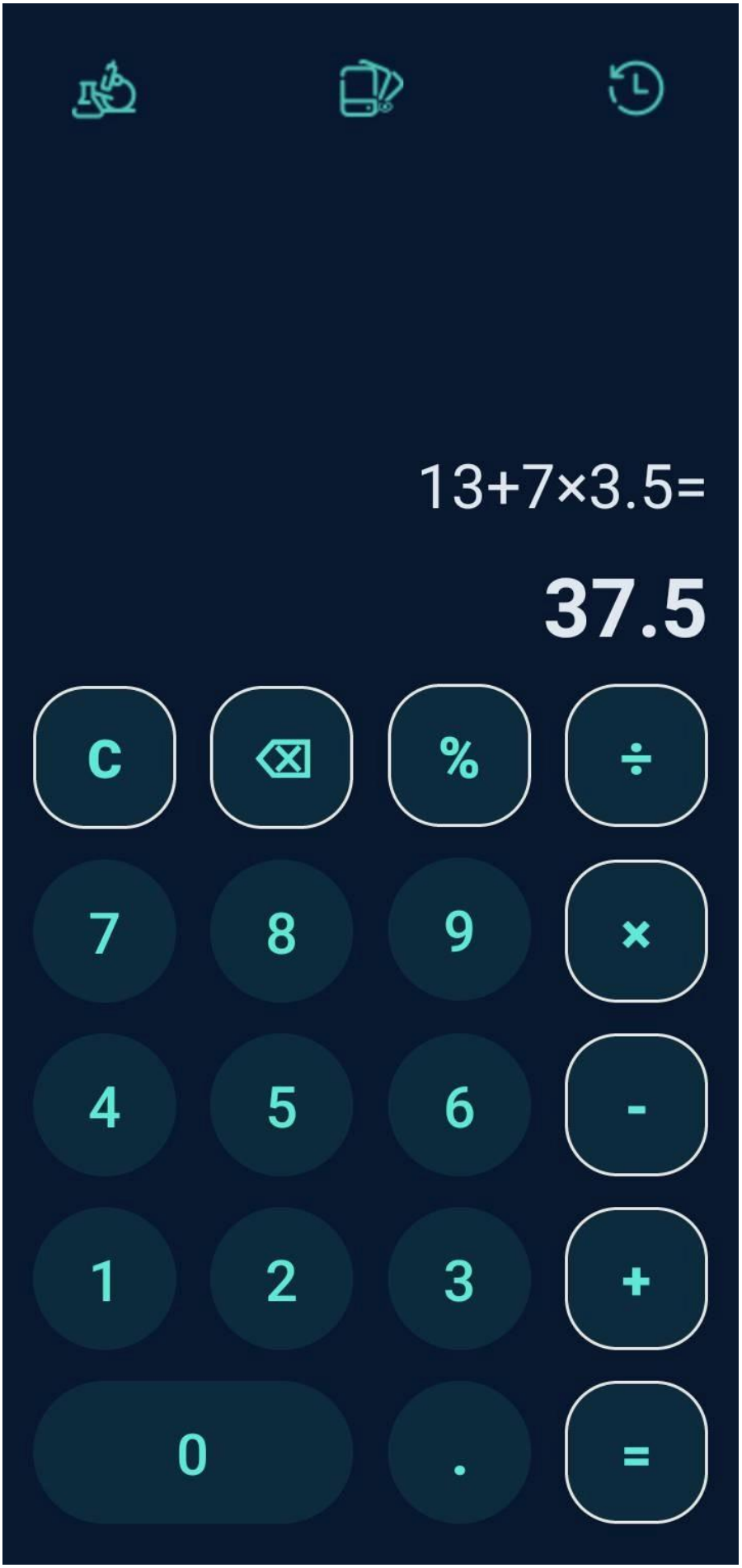
```
        else if (view.getId() == R.id.button_root) {
            mInputValue1TextView.setText("1");
            operation_string = operator.ROOT.name();
            mOperatorTextView.setText("√");
        }
        else {
            Toast.makeText(this, "Enter first number before
operation", Toast.LENGTH_LONG).show();
        }
    }

    public void onEqualsClick(View view) {
//        handle equals click
        if (mInputValue1TextView.getText().toString().equals("") ||
mOperatorTextView.getText().toString().equals("") ||
mInputValue2TextView.getText().toString().equals("")) {
            Toast.makeText(this, "Enter the numbers and the
operation", Toast.LENGTH_LONG).show();
        } else {
            number_one =
Double.parseDouble(mInputValue1TextView.getText().toString());
            number_two =
Double.parseDouble(mInputValue2TextView.getText().toString());
            String operation;

            switch (operator.valueOf(operation_string)) {
                case ADD:

mFinalResultTextView.setText(String.valueOf(mCalculator.addition(numbe
r_one, number_two)));
                    operation =
mInputValue1TextView.getText().toString() +
getString(R.string.button_plus) +
mInputValue2TextView.getText().toString();
                    break;
                case SUB:

mFinalResultTextView.setText(String.valueOf(mCalculator.subtraction(nu
mber_one, number_two)));
                    operation =
mInputValue1TextView.getText().toString() +
getString(R.string.button_minus) +
mInputValue2TextView.getText().toString();
                    break;
                case MUL:

mFinalResultTextView.setText(String.valueOf(mCalculator.multiplication
(number_one, number_two)));
                    operation =
mInputValue1TextView.getText().toString() +
getString(R.string.button_mul) +
mInputValue2TextView.getText().toString();
```

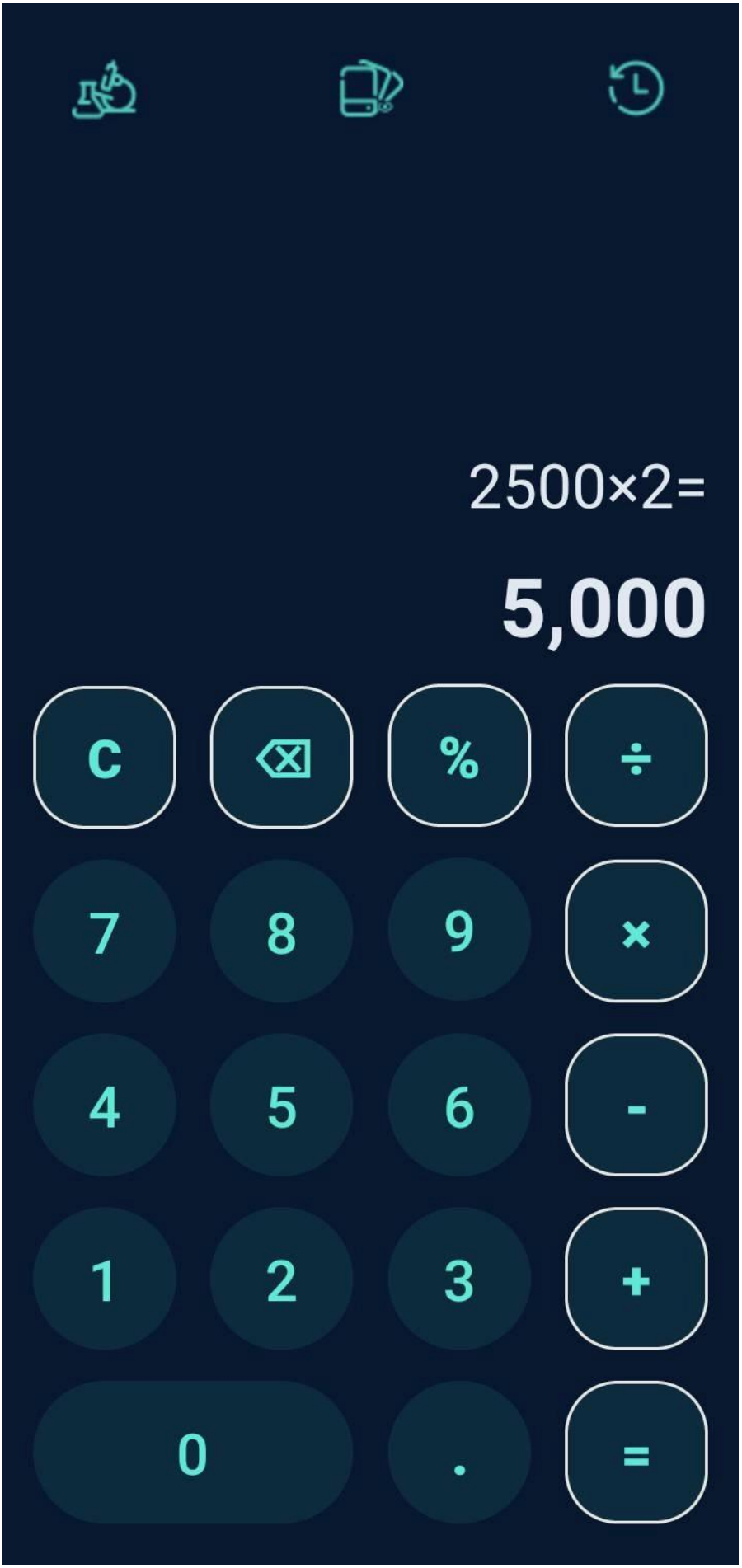
```
                    break;
                case DIV:
                    try {

mFinalResultTextView.setText(String.valueOf(mCalculator.division(numbe
r_one, number_two)));
                        operation =
mInputValue1TextView.getText().toString() +
getString(R.string.button_div) +
mInputValue2TextView.getText().toString();
                    } catch (IllegalArgumentException e) {
                        Toast.makeText(this,
getString(R.string.error), Toast.LENGTH_LONG).show();
                        operation = "";
                        clearAll();
                    }
                    break;
                case MOD:

mFinalResultTextView.setText(String.valueOf(mCalculator.modulus(number
_one, number_two)));
                    operation =
mInputValue1TextView.getText().toString() +
getString(R.string.button_mod) +
mInputValue2TextView.getText().toString();
                    break;
                case ROOT:

mFinalResultTextView.setText(String.valueOf(mCalculator.squareRoot(num
ber_one, number_two)));
                    operation =
mInputValue1TextView.getText().toString() +
getString(R.string.button_root) +
mInputValue2TextView.getText().toString();
                    break;
                case POW:

mFinalResultTextView.setText(String.valueOf(mCalculator.power(number_o
ne, number_two)));
                    operation =
mInputValue1TextView.getText().toString() +
getString(R.string.button_pow) +
mInputValue2TextView.getText().toString();
                    break;
                case NULL:

mFinalResultTextView.setText(getString(R.string.error));
                    operation = "";
                    break;
                default:
                    operation = "";
                    break;
```

```
            }
            mCompleteOperation.setText(operation);
            clearAll();
        }
    }

    //    handle clear click
    public void onClearClick(View view) {
        clearAll();
        mFinalResultTextView.setText("");
        mCompleteOperation.setText("");
    }

    //    clearing most values (needed many times, so created a method
to reduce code duplication
    public void clearAll() {
        mInputValue1TextView.setText("");
        mOperatorTextView.setText("");
        mInputValue2TextView.setText("");
        number_one = 0;
        number_two = 0;
        operation_string = operator.NULL.name();
    }

    //    handle backspace click (the ImageButton) in the layout
    public void onBackspaceClick(View view) {
        view.setOnLongClickListener(new View.OnLongClickListener() {
            @Override
            public boolean onLongClick(View v) {
                clearAll();
                return false;
            }
        });
        if (!mInputValue2TextView.getText().toString().equals("")) {
            backspaceImplementation(mInputValue2TextView);
        } else {
            if (!mOperatorTextView.getText().toString().equals("")) {
                backspaceImplementation(mOperatorTextView);
            } else {
                if
(!mInputValue1TextView.getText().toString().equals("")) {
                    backspaceImplementation(mInputValue1TextView);
                }
            }
        }
    }

    private void backspaceImplementation(TextView view) {
        String backspace = view.getText().toString();
        backspace = backspace.substring(0, backspace.length() - 1);
        view.setText(backspace);
    }
```

```
}
```

III. Screenshot


13+7×3.5=
37.5
C
%
÷
7
8
9
×
4
5
6
-
1
2
3
+
0
.
=


2500×2=
5,000
C
%
÷
7
8
9
×
4
5
6
-
1
2
3
+
0
.
=

QUESTION 2

IV. UI XML Code

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
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">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">

        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/AppTheme.PopupOverlay"
            app:title="@string/app_name"
            />

    </com.google.android.material.appbar.AppBarLayout>

    <include layout="@layout/content_main" />

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

V. Main Activity Kotlin File

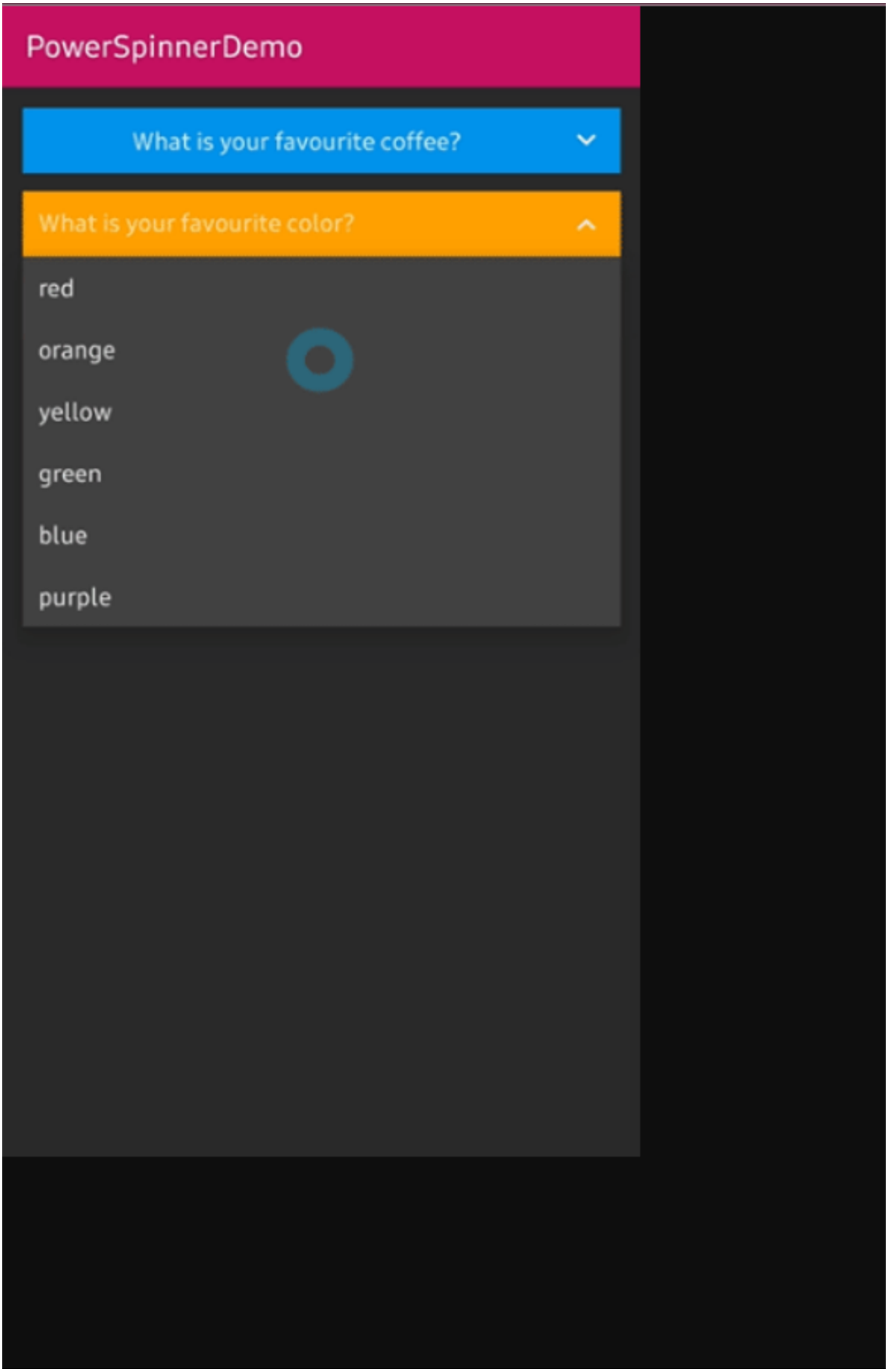
```
package com.cvaghela.spinner.searchablespinner

import android.os.Bundle
import android.view.View
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import androidx.lifecycle.MutableLiveData
import
com.cvaghela.spinner.searchablespinner.interfaces.OnItemSelectedListen
er
import com.google.gson.Gson
import com.google.gson.reflect.TypeToken

class MainActivity : AppCompatActivity() {

    lateinit var searchableSpinner: SearchableSpinner
    lateinit var searchableSpinner1: SearchableSpinner
    lateinit var searchableSpinner2: SearchableSpinner
    lateinit var searchableSpinner3: SearchableSpinner

    lateinit var adapter: SimpleArrayListAdapter
    lateinit var adapterTag: SimpleArrayWithTagAdapter

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        searchableSpinner = findViewById(R.id.searchableSpinner)
        searchableSpinner1 = findViewById(R.id.searchableSpinner1)
        searchableSpinner2 = findViewById(R.id.searchableSpinner2)
        searchableSpinner3 = findViewById(R.id.searchableSpinner3)

        getListData().observeForever {

            if (it.size > 0) {

                adapter = SimpleArrayListAdapter(this, it)
                adapterTag = SimpleArrayWithTagAdapter(this, it)
                searchableSpinner.setAdapter(adapter)
                searchableSpinner.setOnItemSelectedListener(object :
OnItemSelectedListener {
                    override fun onItemSelected(view: View?, position:
Int, id: Long) {
                        Toast.makeText(
                            this@MainActivity,
```

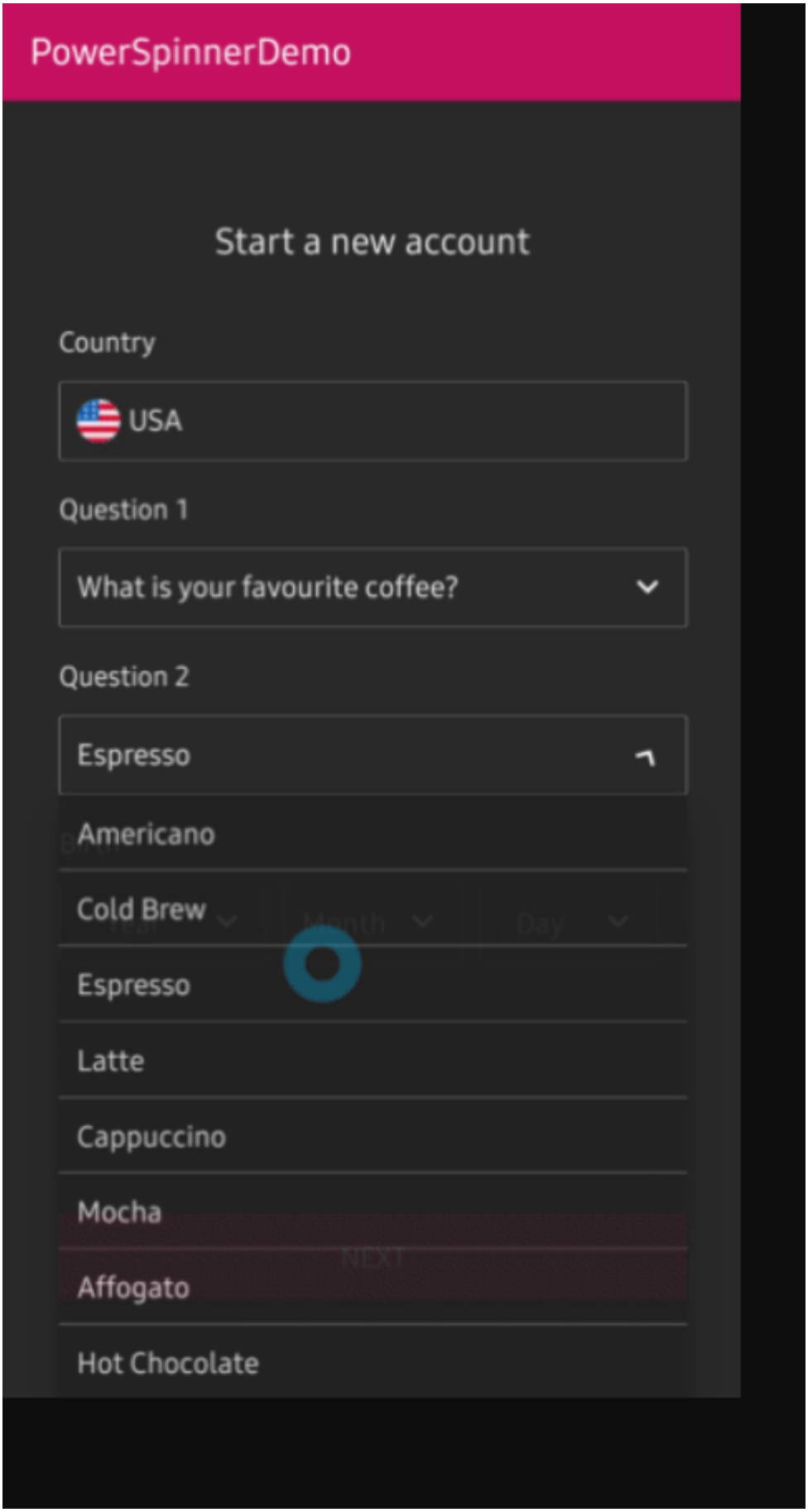
```
                            "Item on position " + position + " : " +
adapter.getItem(
                                position
                            ) + " Selected",
                            Toast.LENGTH_SHORT
                        ).show()
                    }

                    override fun onNothingSelected() {
                        Toast.makeText(this@MainActivity, "Nothing
Selected", Toast.LENGTH_SHORT)
                            .show()
                    }

                })


                searchableSpinner1.setAdapter(adapter)
                searchableSpinner1.setOnItemSelectedListener(object :
OnItemSelectedListener {
                    override fun onItemSelected(view: View?, position:
Int, id: Long) {
                        Toast.makeText(
                            this@MainActivity,
                            "Item on position " + position + " : " +
adapter.getItem(
                                position
                            ) + " Selected",
                            Toast.LENGTH_SHORT
                        ).show()
                    }

                    override fun onNothingSelected() {
                        Toast.makeText(this@MainActivity, "Nothing
Selected", Toast.LENGTH_SHORT)
                            .show()
                    }

                })

                searchableSpinner2.setAdapter(adapterTag)
                searchableSpinner2.setOnItemSelectedListener(object :
OnItemSelectedListener {
                    override fun onItemSelected(view: View?, position:
Int, id: Long) {
                        Toast.makeText(
                            this@MainActivity,
                            "Item on position " + position + " : " +
adapterTag.getItem(
                                position
                            ) + " Selected",
                            Toast.LENGTH_SHORT
```

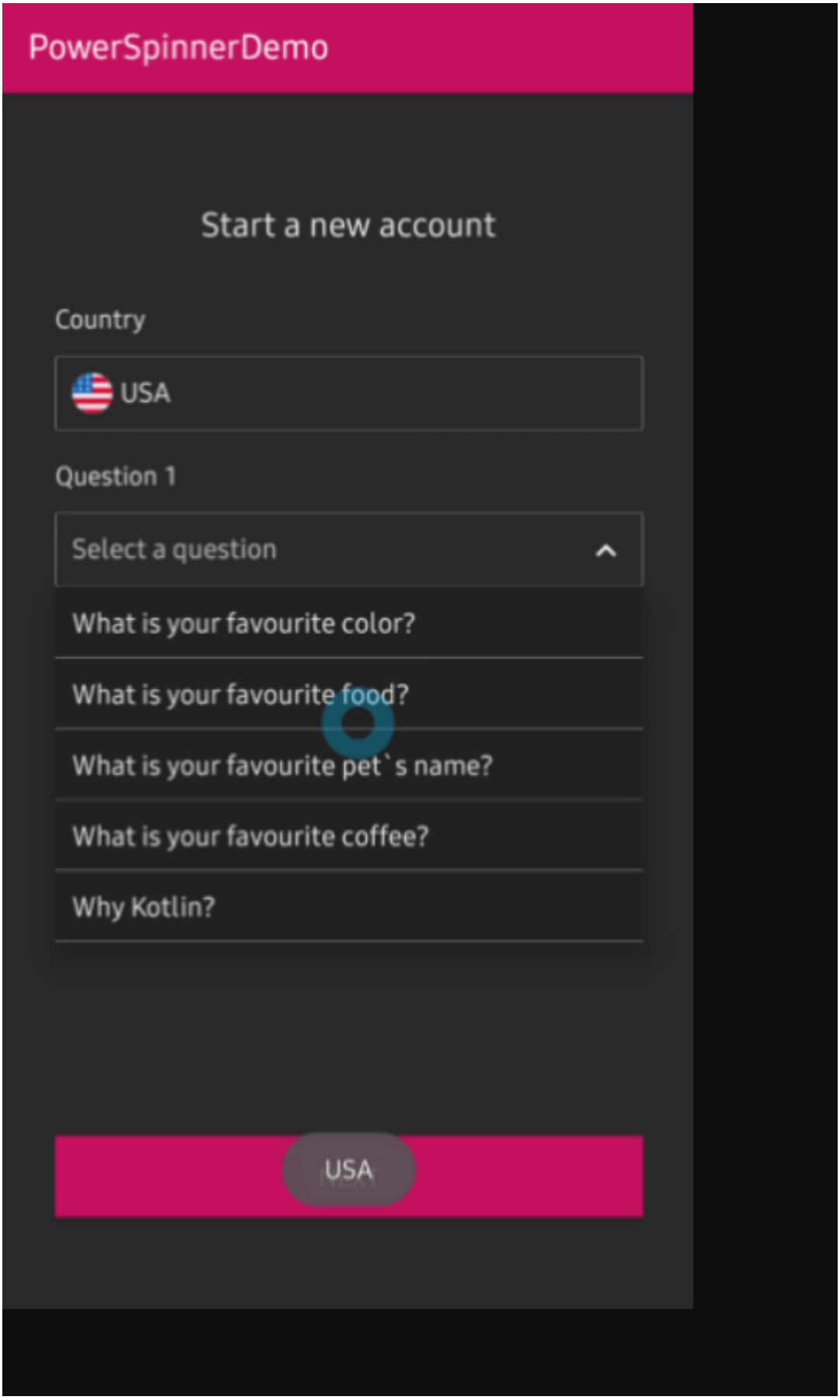
```
                        ).show()
                    }

                    override fun onNothingSelected() {
                        Toast.makeText(this@MainActivity, "Nothing
Selected", Toast.LENGTH_SHORT)
                            .show()
                    }

                })

                searchableSpinner3.setAdapter(adapter)
                searchableSpinner3.setOnItemSelectedListener(object :
OnItemSelectedListener {
                    override fun onItemSelected(view: View?, position:
Int, id: Long) {
                        Toast.makeText(
                            this@MainActivity,
                            "Item on position " + position + " : " +
adapter.getItem(
                                position
                            ) + " Selected",
                            Toast.LENGTH_SHORT
                        ).show()
                    }

                    override fun onNothingSelected() {
                        Toast.makeText(this@MainActivity, "Nothing
Selected", Toast.LENGTH_SHORT)
                            .show()
                    }

                })

            }

        }

    }


    fun getListData(): MutableLiveData<ArrayList<String>> {
        val encodedString = MutableLiveData<ArrayList<String>>()

        val jsonFileString = getJsonDataFromRaw(
            this,
            R.raw.array_item
        )
        val type = object : TypeToken<ArrayList<String>>() {}.type
        encodedString.postValue(Gson().fromJson(jsonFileString, type))

        return encodedString
```

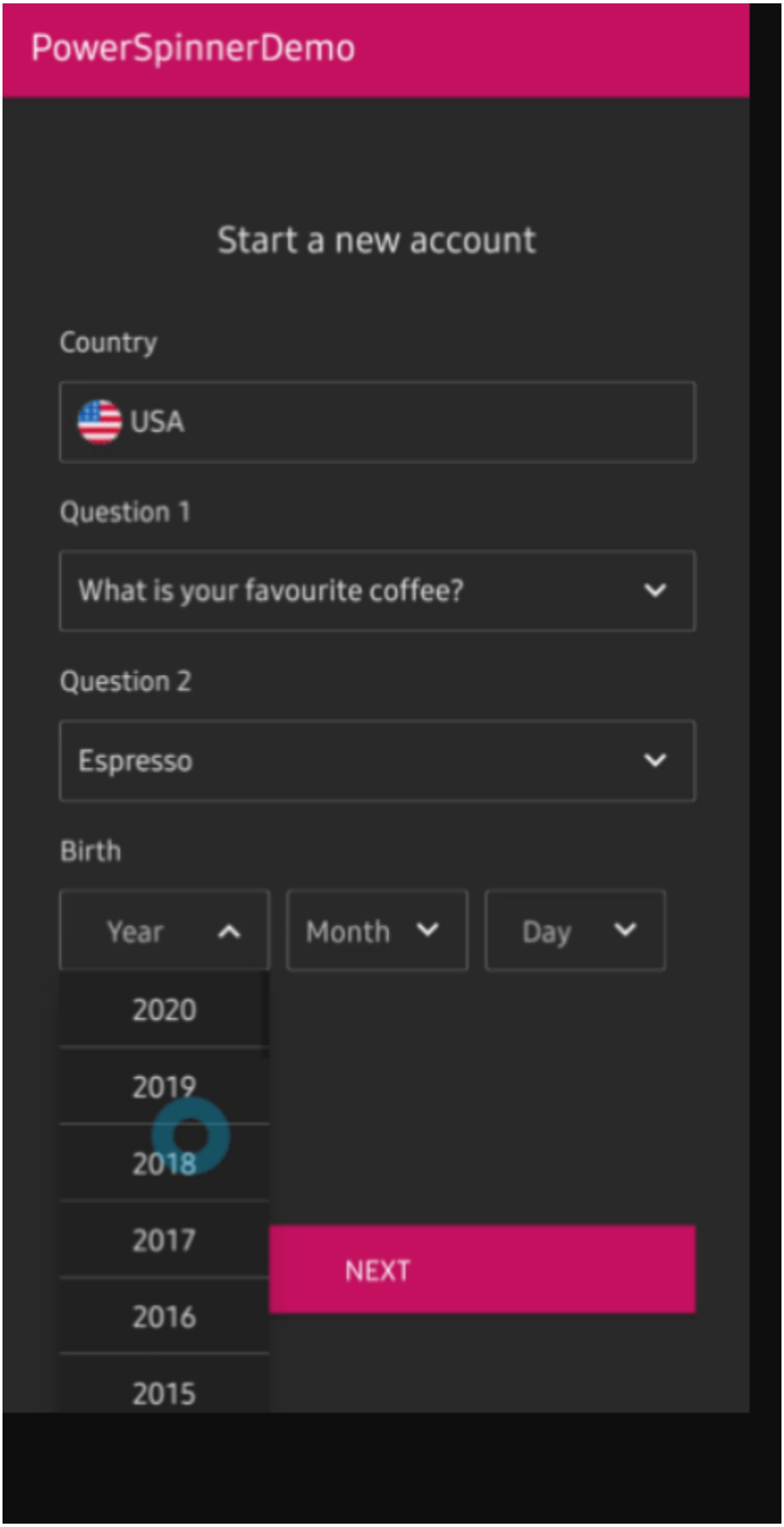
```
        }
}
```

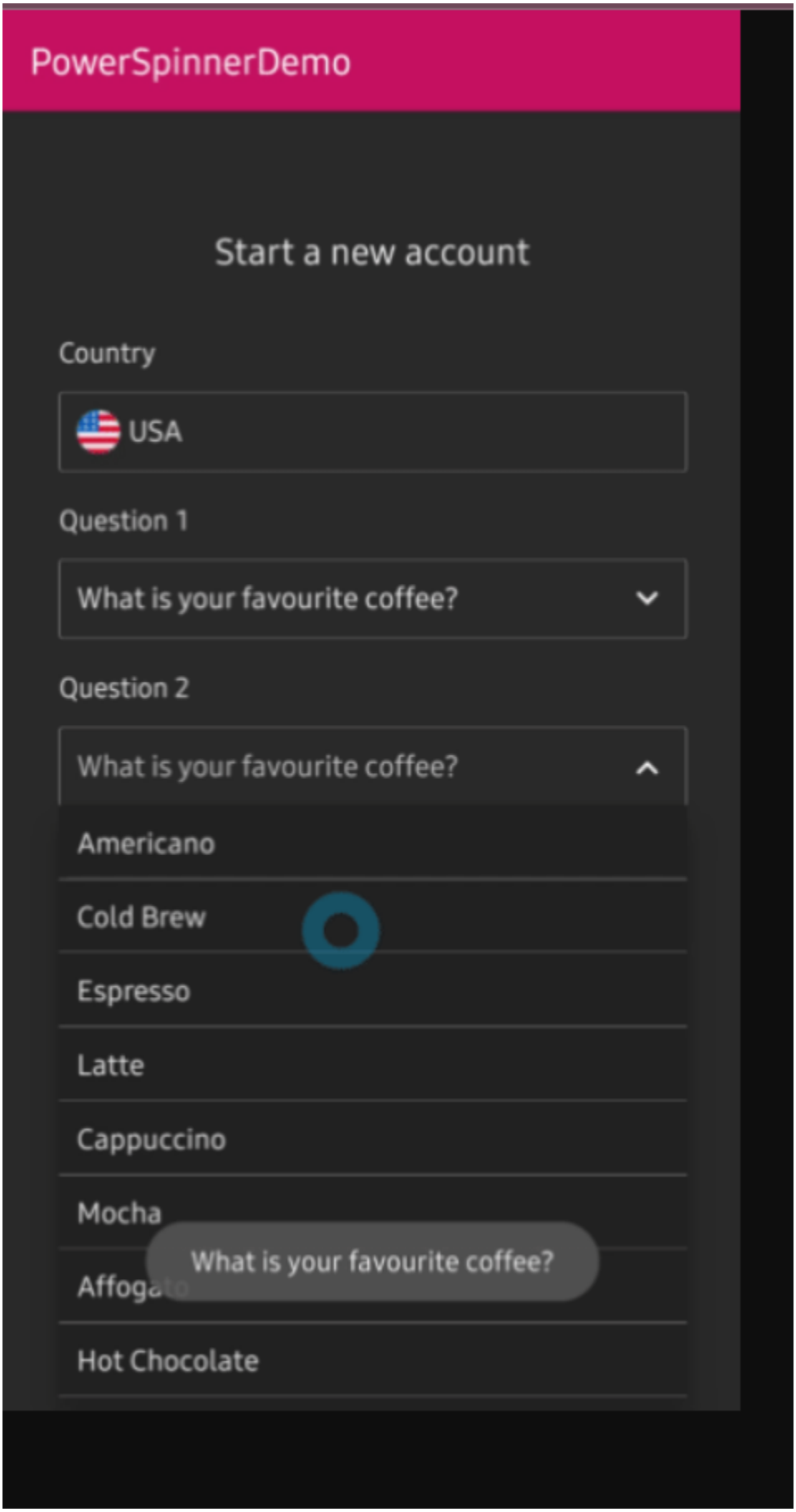
VI. Screenshot

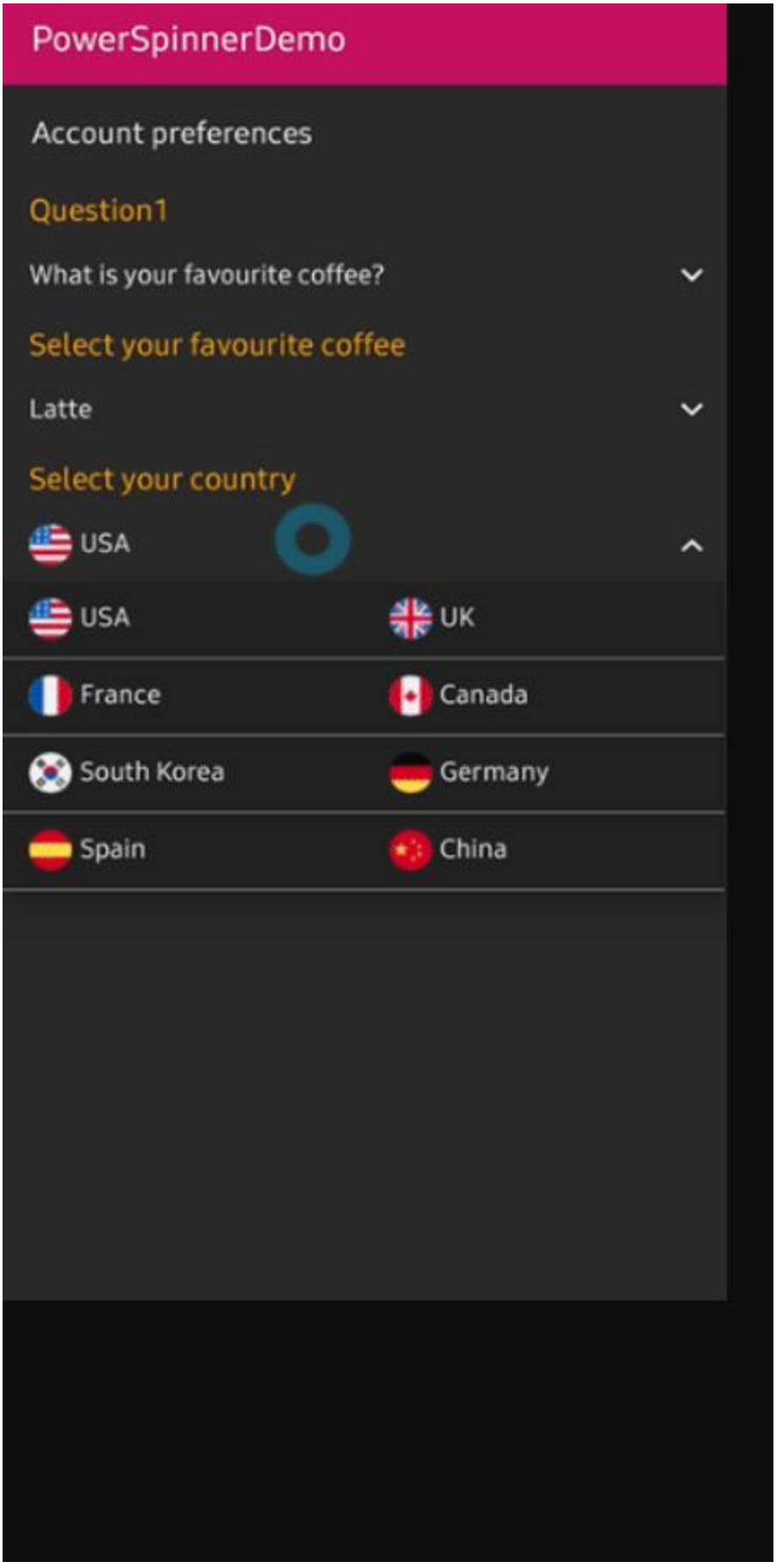

PowerSpinnerDemo
What is your favourite coffee?
What is your favourite color?
red
orange
yellow
green
blue
purple

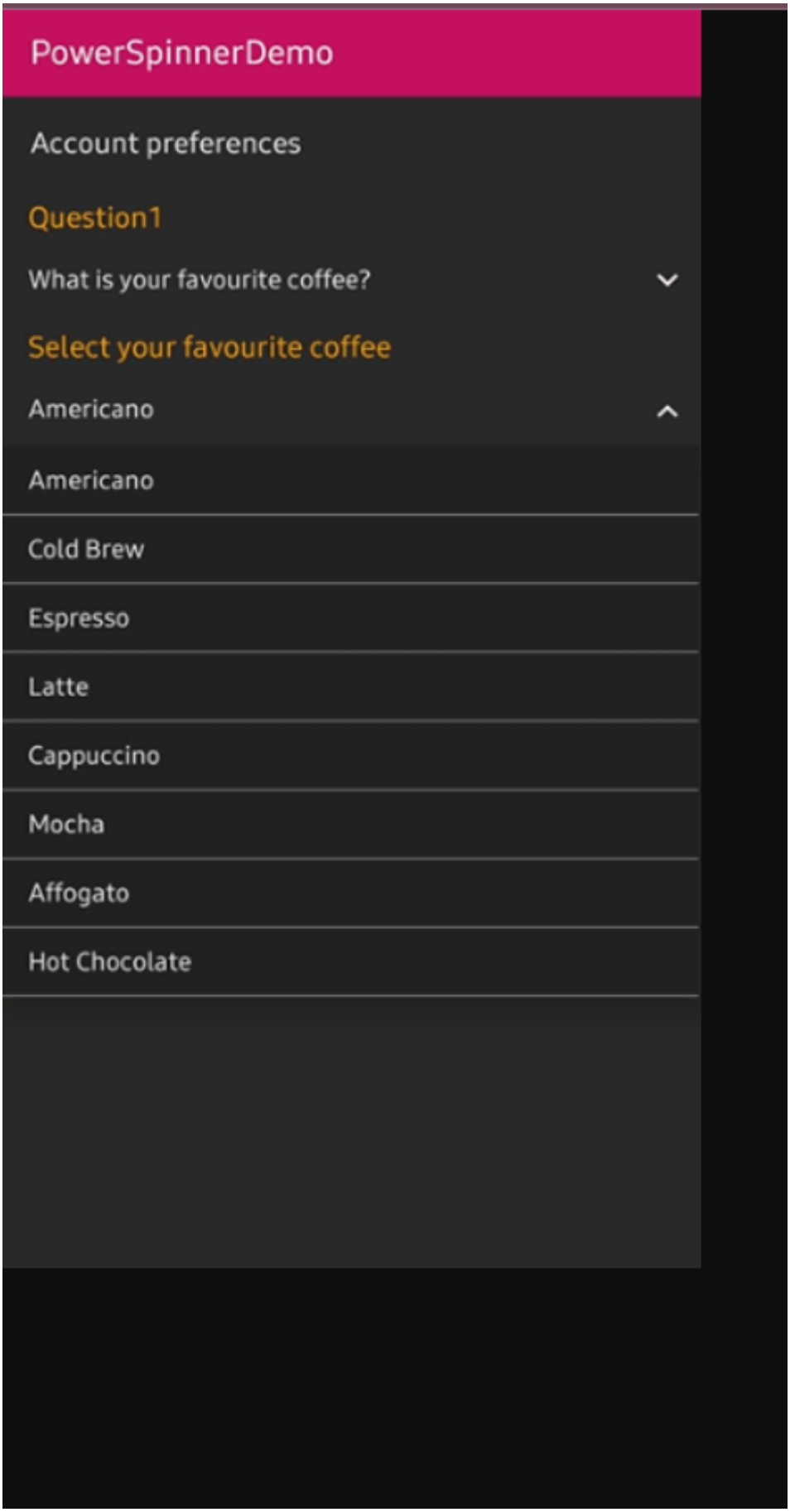

PowerSpinnerDemo
Start a new account
Country
USA
Question 1
What is your favourite coffee?
Question 2
Espresso
Americano
Cold Brew
Espresso
Latte
Cappuccino
Mocha
Affogato
Hot Chocolate

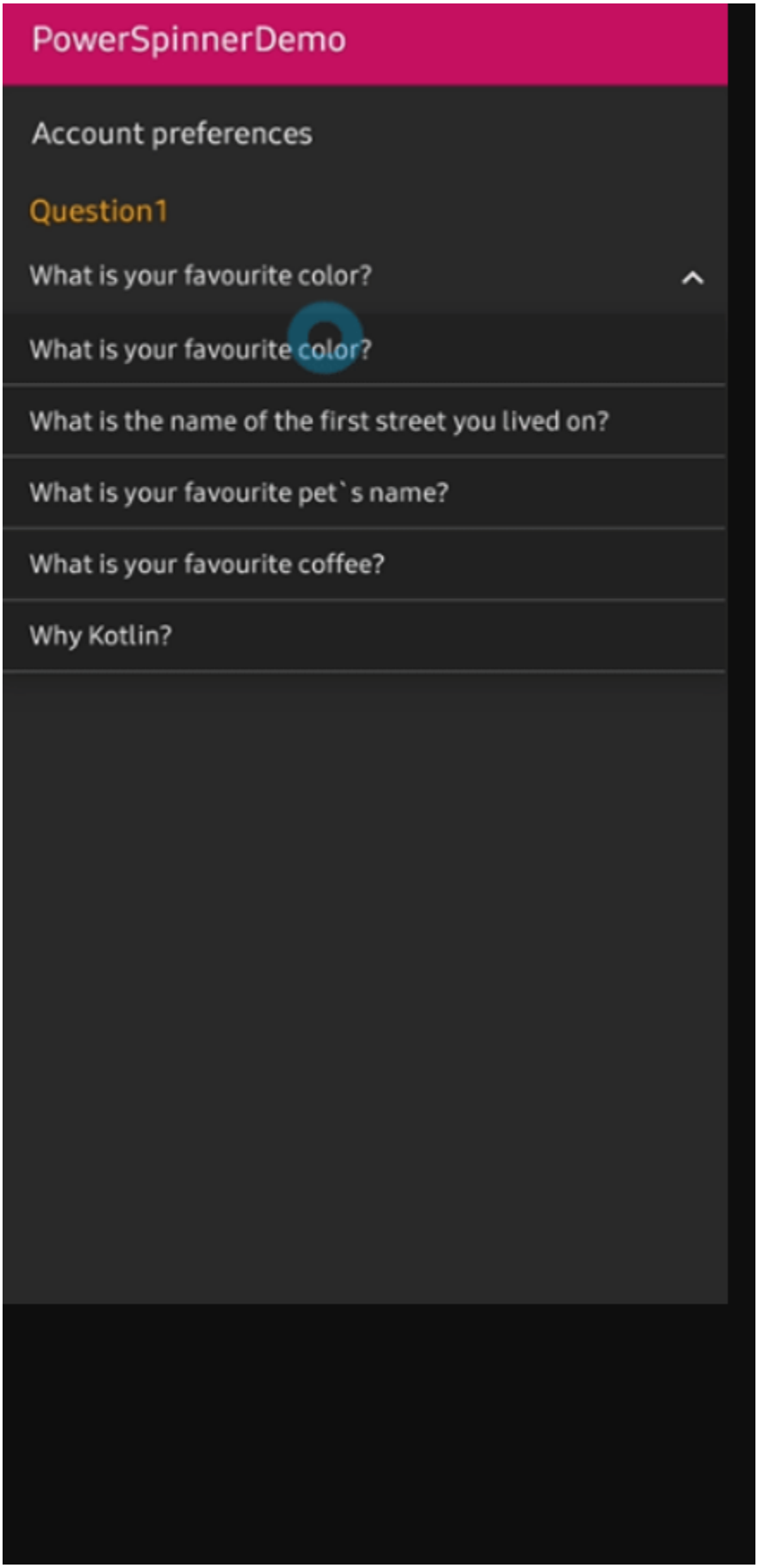

PowerSpinnerDemo
Start a new account
Country
USA
Question 1
Select a question
What is your favourite color?
What is your favourite food?
What is your favourite pet`s name?
What is your favourite coffee?
Why Kotlin?
USA


PowerSpinnerDemo
Start a new account
Country
USA
Question 1
What is your favourite coffee?
Question 2
Espresso
Birth
Year
Month
Day
2020
2019
2018
2017
2016
2015
NEXT


PowerSpinnerDemo
Start a new account
Country
USA
Question 1
What is your favourite coffee?
Question 2
What is your favourite coffee?
Americano
Cold Brew
Espresso
Latte
Cappuccino
Mocha
What is your favourite coffee?
Affogato
Hot Chocolate


PowerSpinnerDemo
Account preferences
Question1
What is your favourite coffee?
Select your favourite coffee
Latte
Select your country
USA
USA
UK
France
Canada
South Korea
Germany
Spain
China


PowerSpinnerDemo
Account preferences
Question1
What is your favourite coffee?
Select your favourite coffee
Americano
Americano
Cold Brew
Espresso
Latte
Cappuccino
Mocha
Affogato
Hot Chocolate



QUESTION 3 – Notes app that uses Putrxtra

and memory sharing

VII. UI XML Code

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
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/layout_coordinator"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:focusableInTouchMode="true"
    tools:context="com.rafapps.simplenotes.NoteActivity">

    <ScrollView
        android:id="@+id/scroll_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:fillViewport="true"
        tools:context="com.rafapps.simplenotes.NoteActivity"
        tools:showIn="@layout/activity_note">

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

            <EditText
                android:id="@+id/et_title"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:ems="10"
                android:hint="@string/title"
                android:imeOptions="flagNoExtractUi"
                android:inputType="textCapWords"
                android:maxLength="36"
                android:paddingEnd="8dp"
                android:paddingStart="8dp"
                android:textCursorDrawable="@null"
                android:textStyle="bold"
                app:layout_constraintLeft_toLeftOf="parent"
                app:layout_constraintTop_toTopOf="parent" />

            <EditText
                android:id="@+id/et_note"
                android:layout_width="match_parent"
                android:layout_height="0dp"
                android:layout_weight="1"
```

```
                        android:autoLink="web|email"
                        android:ems="10"
                        android:gravity="top"
                        android:hint="@string/note"
                        android:imeOptions="flagNoExtractUi"

            android:inputType="textCapSentences|textMultiLine"
                        android:linksClickable="true"
                        android:paddingEnd="8dp"
                        android:paddingStart="8dp"
                        android:textCursorDrawable="@null"
                        app:layout_constraintLeft_toLeftOf="parent"

            app:layout_constraintTop_toBottomOf="@+id/title" />

                </LinearLayout>

            </ScrollView>

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

VIII. Main Activity Kotlin File

```
package com.rafapps.simplenotes;


import android.content.Context;

import android.content.DialogInterface;

import android.content.Intent;

import android.content.SharedPreferences;

import android.content.res.ColorStateList;
```

```
import android.graphics.Color;

import android.graphics.drawable.ColorDrawable;

import android.os.Bundle;

import android.preference.PreferenceManager;

import android.text.TextUtils;

import android.view.Menu;

import android.view.MenuItem;

import android.widget.EditText;


import androidx.annotation.ColorInt;

import androidx.appcompat.app.AlertDialog;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.content.ContextCompat;

import androidx.core.graphics.ColorUtils;


import
com.joaomgcd.taskerpluginlibrary.condition.TaskerPluginRunnerCondition
;


public class NoteActivity extends AppCompatActivity {


    private static final String EXTRA_NOTE_TITLE = "EXTRA_NOTE_TITLE";


    private boolean colourNavbar;

    private String title, note;

    private EditText noteText, titleText;

    private AlertDialog dialog;
```

```
    private @ColorInt
    int colourPrimary, colourFont, colourBackground;

    public static Intent getStartIntent(Context context, String title)
{
        Intent intent = new Intent(context, NoteActivity.class);
        intent.putExtra(EXTRA_NOTE_TITLE, title);
        return intent;
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_note);
        titleText = findViewById(R.id.et_title);
        noteText = findViewById(R.id.et_note);

        Intent intent = getIntent();
        String action = intent.getAction();
        String type = intent.getType();

        // If activity started from a share intent
        if (Intent.ACTION_SEND.equals(action) && type != null) {
            if ("text/plain".equals(type)) {
                String sharedText =
intent.getStringExtra(Intent.EXTRA_TEXT);
```

```
                noteText.setText(sharedText);
                note = sharedText;
                title = "";
            }
        } else { // If activity started from the notes list
            title = intent.getStringExtra(EXTRA_NOTE_TITLE);
            if (title == null || TextUtils.isEmpty(title)) {
                title = "";
                note = "";
                noteText.requestFocus();
                if (getSupportActionBar() != null)
getSupportActionBar().setTitle(getString(R.string.new_note));
            } else {
                titleText.setText(title);
                note = HelperUtils.readFile(NoteActivity.this, title);
                noteText.setText(note);
                if (getSupportActionBar() != null)
                    getSupportActionBar().setTitle(title);
            }
        }

getSettings(PreferenceManager.getDefaultSharedPreferences(NoteActivity
.this));
        applySettings();
    }
```

```
@Override
public void onRestart() {
    super.onRestart();
    note = noteText.getText().toString().trim();
    if (getCurrentFocus() != null)
        getCurrentFocus().clearFocus();
}

@Override
public void onPause() {
    if (!isChangingConfigurations()) {
        saveFile();
    }
    if (dialog != null && dialog.isShowing())
        dialog.dismiss();
    dialog = null;
    super.onPause();
}

@Override
public void onBackPressed() {
    super.onBackPressed();
}

@Override
```

```
    public boolean onSupportNavigateUp() {
        onBackPressed();
        return true;
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_note, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        if (id == R.id.btn_undo) {
            noteText.setText(note);
            noteText.setSelection(noteText.getText().length());
            return (true);
        }

        if (id == R.id.btn_share) {
            Intent sendIntent = new Intent();
            sendIntent.setAction(Intent.ACTION_SEND);
            sendIntent.putExtra(Intent.EXTRA_TEXT,
noteText.getText().toString());
            sendIntent.setType("text/plain");
```

```
            startActivity(Intent.createChooser(sendIntent,
getString(R.string.share_to)));

            return (true);

        }



        if (id == R.id.btn_delete) {

            dialog = new AlertDialog.Builder(NoteActivity.this,
R.style.AlertDialogTheme)

                    .setTitle(getString(R.string.confirm_delete))


.setMessage(getString(R.string.confirm_delete_text))

                    .setPositiveButton(getString(R.string.yes), new
DialogInterface.OnClickListener() {

                        public void onClick(DialogInterface dialog,
int which) {

                            if
(HelperUtils.fileExists(NoteActivity.this, title)) {

                                deleteFile(title +
HelperUtils.TEXT_FILE_EXTENSION);

                            }

                            title = "";

                            note = "";

                            titleText.setText(title);

                            noteText.setText(note);

                            finish();

                        }

                    })

                    .setNegativeButton(getString(R.string.no), new
DialogInterface.OnClickListener() {
```

```
                        public void onClick(DialogInterface dialog,
int which) {


                        }

                    })

.setIcon(ContextCompat.getDrawable(getApplicationContext(),
R.drawable.ic_delete_white_24dp))

                    .show();

            if (dialog.getWindow() != null) {

dialog.getWindow().getDecorView().setBackgroundColor(colourPrimary);

            }

dialog.getButton(DialogInterface.BUTTON_POSITIVE).setTextColor(Color.W
HITE);

dialog.getButton(DialogInterface.BUTTON_NEGATIVE).setTextColor(Color.W
HITE);

            return (true);

        }



        return (super.onOptionsItemSelected(item));

    }



    private void getSettings(SharedPreferences preferences) {

        colourPrimary =
preferences.getInt(HelperUtils.PREFERENCE_COLOUR_PRIMARY,
ContextCompat.getColor(NoteActivity.this, R.color.colorPrimary));

        colourFont =
preferences.getInt(HelperUtils.PREFERENCE_COLOUR_FONT, Color.BLACK);
```

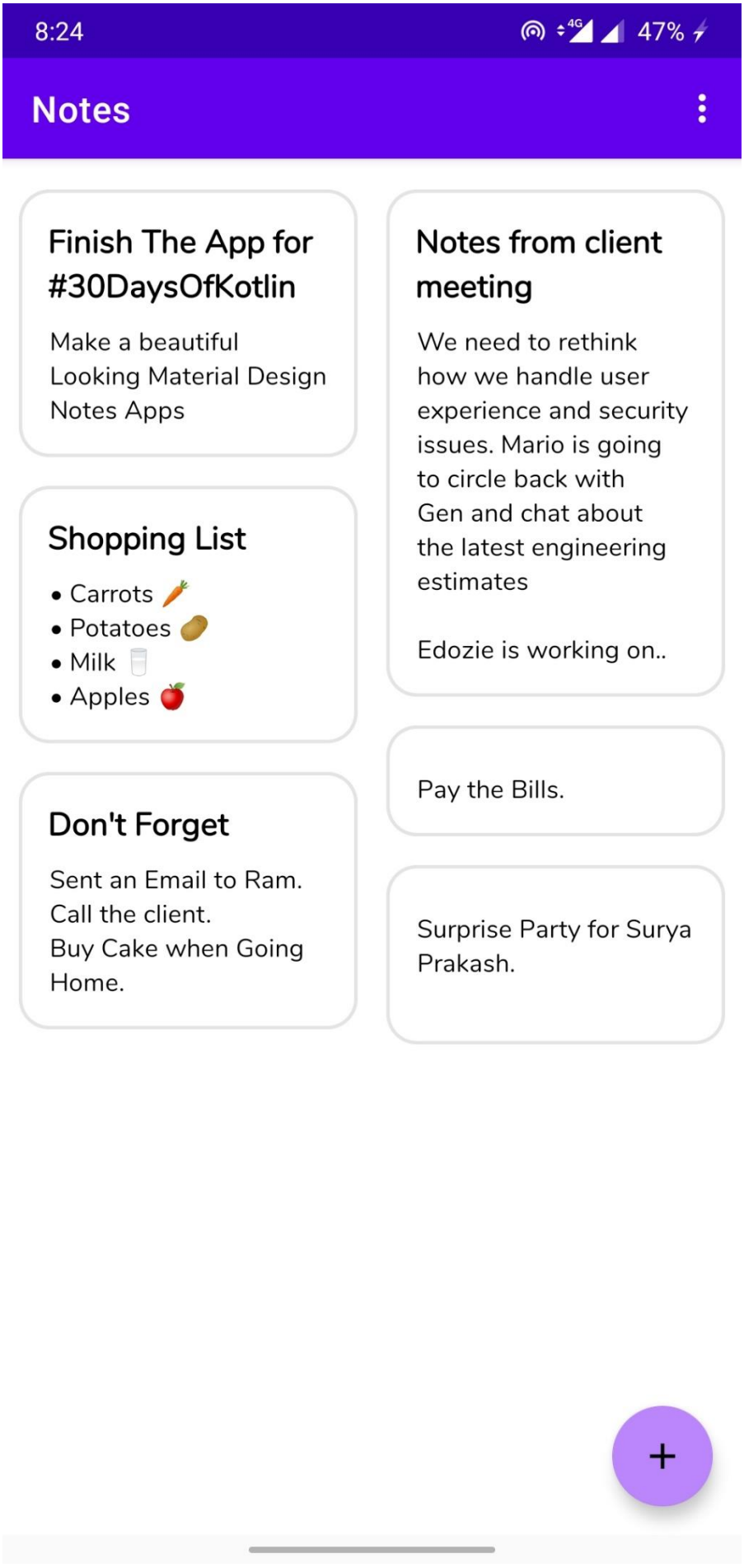
```
        colourBackground =
preferences.getInt(HelperUtils.PREFERENCE_COLOUR_BACKGROUND,
Color.WHITE);

        colourNavbar =
preferences.getBoolean(HelperUtils.PREFERENCE_COLOUR_NAVBAR, false);

    }



    private void applySettings() {

        HelperUtils.applyColours(NoteActivity.this, colourPrimary,
colourNavbar);



        // Set text field underline colour


noteText.setBackgroundTintList(ColorStateList.valueOf(colourPrimary));


titleText.setBackgroundTintList(ColorStateList.valueOf(colourPrimary))
;



        // Set actionbar and background colour


findViewById(R.id.scroll_view).setBackgroundColor(colourBackground);

        if (getSupportActionBar() != null)

            getSupportActionBar().setBackgroundDrawable(new
ColorDrawable(colourPrimary));



        // Set font colours

        titleText.setTextColor(colourFont);

        noteText.setTextColor(colourFont);



        // Set hint colours
```

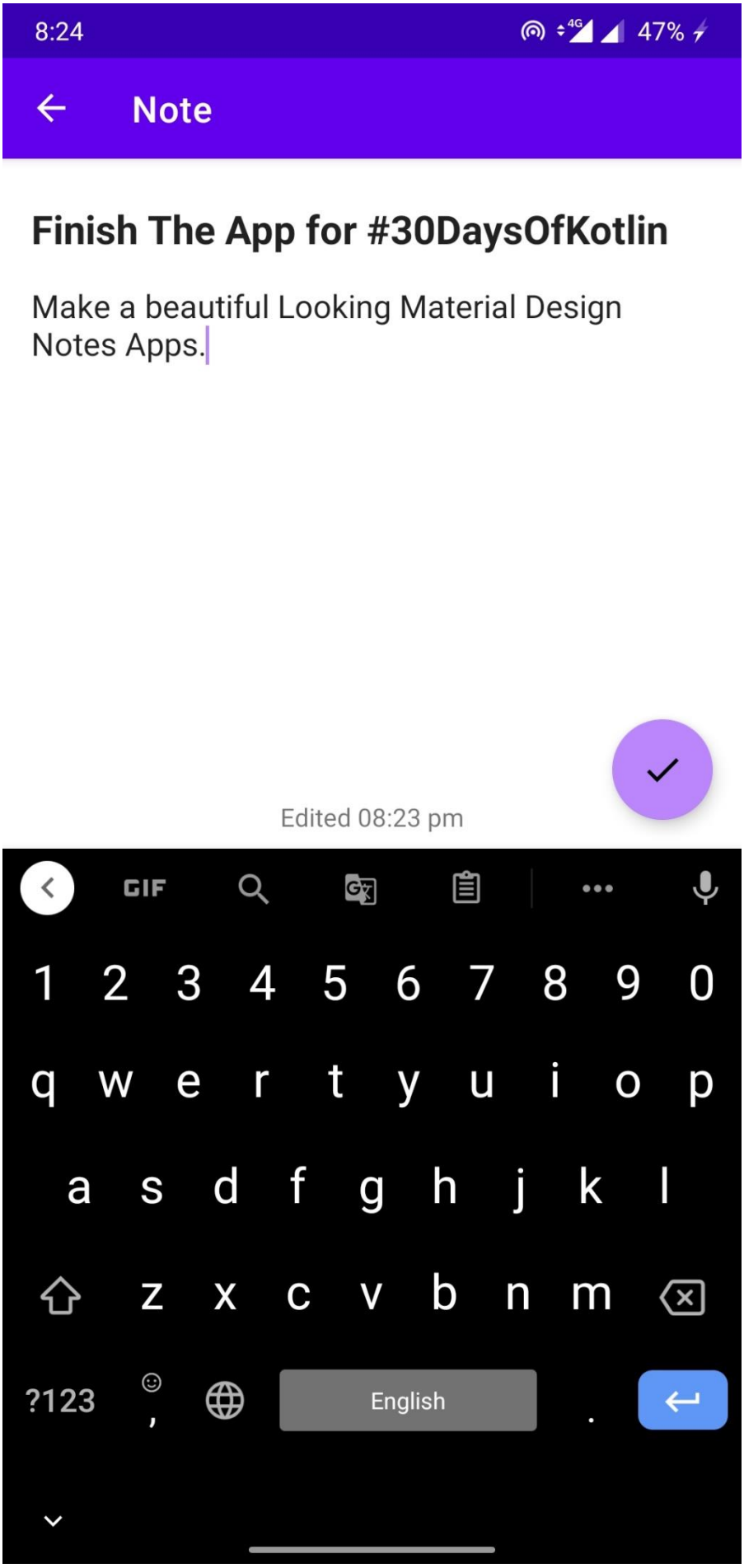
```
titleText.setHintTextColor(ColorUtils.setAlphaComponent(colourFont,
120));


noteText.setHintTextColor(ColorUtils.setAlphaComponent(colourFont,
120));

    }



    private void saveFile() {

        // Get current title and note

        String newTitle =
titleText.getText().toString().trim().replace("/", " ");

        String newNote = noteText.getText().toString().trim();



        // Check if title and note are empty

        if (TextUtils.isEmpty(newTitle) && TextUtils.isEmpty(newNote))
{

            return;

        }



        // Check if title and note are unchanged

        if (newTitle.equals(title) && newNote.equals(note)) {

            return;

        }



        // Get file name to be saved if the title has changed or if it
is empty

        if (!title.equals(newTitle) || TextUtils.isEmpty(newTitle)) {

            newTitle = newFileName(newTitle);
```

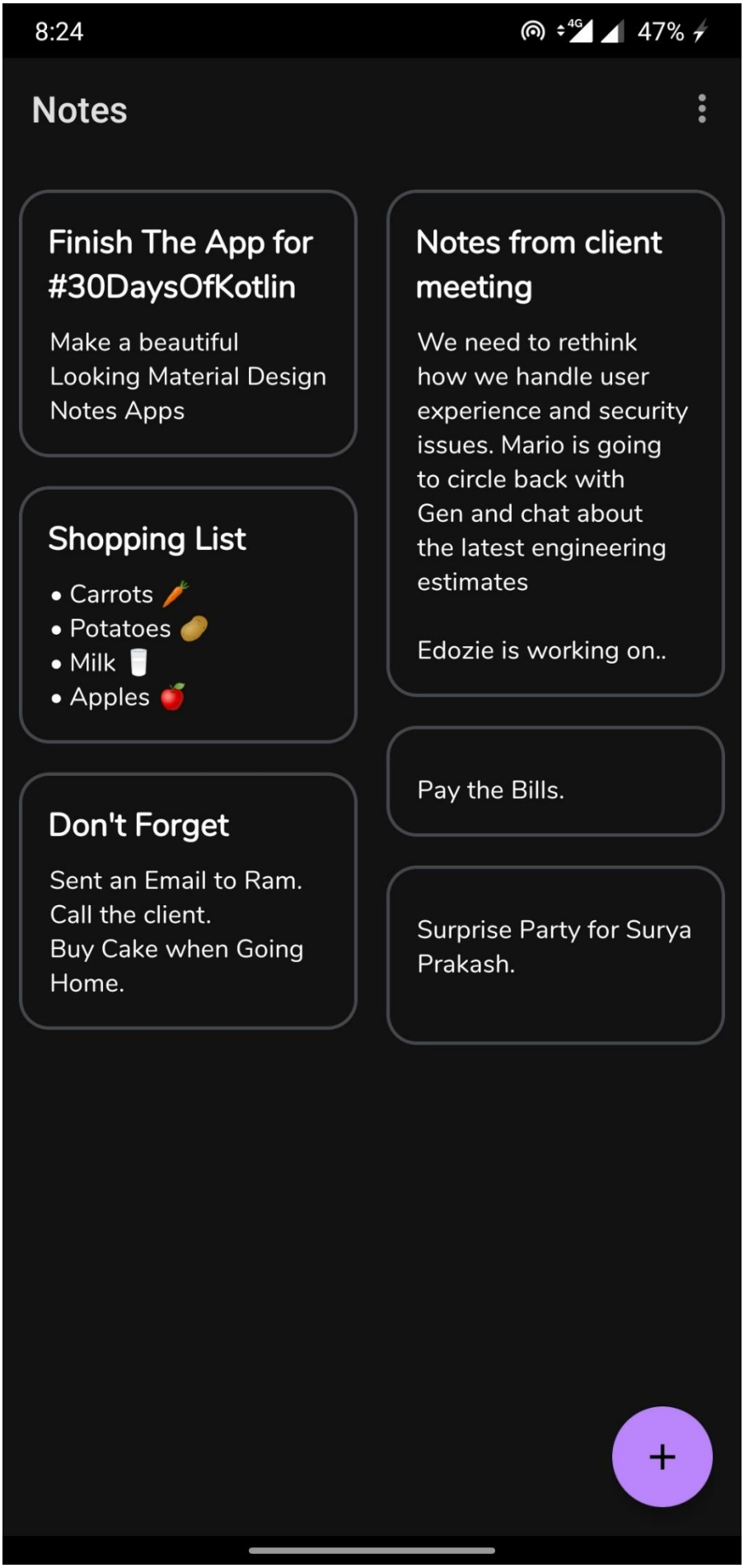
```
            titleText.setText(newTitle);
        }

        // Save the file with the new file name and content
        HelperUtils.writeFile(NoteActivity.this, newTitle, newNote);

        // If the title is not empty and the file name has changed
then delete the old file
        if (!TextUtils.isEmpty(title) && !newTitle.equals(title)) {
            deleteFile(title + HelperUtils.TEXT_FILE_EXTENSION);
        }

        // Set the title to be the new saved title for when the home
button is pressed
        title = newTitle;

        // Send Tasker event
        TaskerPluginRunnerCondition.Companion.requestQuery(
                this,
                TaskerEventNoteUpdateActivity.class,
                new NoteOutput(newTitle, newNote)
        );
    }

    private String newFileName(String name) {
        // If it is empty, give it a default title of "Note"
        if (TextUtils.isEmpty(name)) {
```

```
            name = getString(R.string.note);
        }
        // If the name already exists, append a number to it
        if (HelperUtils.fileExists(NoteActivity.this, name)) {
            int i = 1;
            while (true) {
                if (!HelperUtils.fileExists(NoteActivity.this, name +
" (" + i + ")") || title.equals(name + " (" + i + ")")) {
                    name = (name + " (" + i + ")");
                    break;
                }
                i++;
            }
        }
        return name;
    }

}
```

IX. Screenshot


8:24
47%
Notes
Finish The App for #30DaysOfKotlin
Make a beautiful Looking Material Design Notes Apps
Shopping List
• Carrots
• Potatoes
• Milk
• Apples
Don't Forget
Sent an Email to Ram.
Call the client.
Buy Cake when Going Home.
Notes from client meeting
We need to rethink how we handle user experience and security issues. Mario is going to circle back with Gen and chat about the latest engineering estimates
Edozie is working on..
Pay the Bills.
Surprise Party for Surya Prakash.
+


8:24
47%
Note
Finish The App for #30DaysOfKotlin
Make a beautiful Looking Material Design Notes Apps.
Edited 08:23 pm
GIF
1 2 3 4 5 6 7 8 9 0
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
?123
English


8:24
47%
Notes
Finish The App for #30DaysOfKotlin
Make a beautiful Looking Material Design Notes Apps
Shopping List
• Carrots
• Potatoes
• Milk
• Apples
Don't Forget
Sent an Email to Ram.
Call the client.
Buy Cake when Going Home.
Notes from client meeting
We need to rethink how we handle user experience and security issues. Mario is going to circle back with Gen and chat about the latest engineering estimates
Edozie is working on..
Pay the Bills.
Surprise Party for Surya Prakash.
+