ANDROID CALCULATOR APP

INDEX

OVERVIEW

In this course design we are going to create a Calculator App that can do Arithmetic operations.

It will run on the Android Platform hence we can be able to deploy it and use it in our smartphones.

We will also be following the material design guidelines in order to make our app interface appear appealing to the eye.

PURPOSE OF THE DESIGN

Our Material designed Calculator should be capable of performing a number of arithmetic operations like

- Addition
- Subtraction
- Multiplication
- Division

Create a Graphical input style other than console terminal input.

Also, our calculator will only process integer values and not floats.

TOPIC REQUIREMENTS

1. JAVA JDK
2. ANDROID SDK
3. MATERIAL DESIGN PRINCIPLES
4. XML
5. ANDROID STUDIO

MAIN IDEAS AND METHODS OF SOLVING THE PROBLEM

We will use Material design principles to sketch the app interface before we write any code.

Sketch the app inputs on a paper piece

We will use java to write our principle code and xml to define our app interface.

We will write the code using a development environment called Android studio as it is suitable for developing android apps and deployment and debugging is easier and way faster.

DESIGN EXPLANATION AND THOUGHT.

We will create clickable Text Views with numbers and operations that will help the user to input their operations with ease.

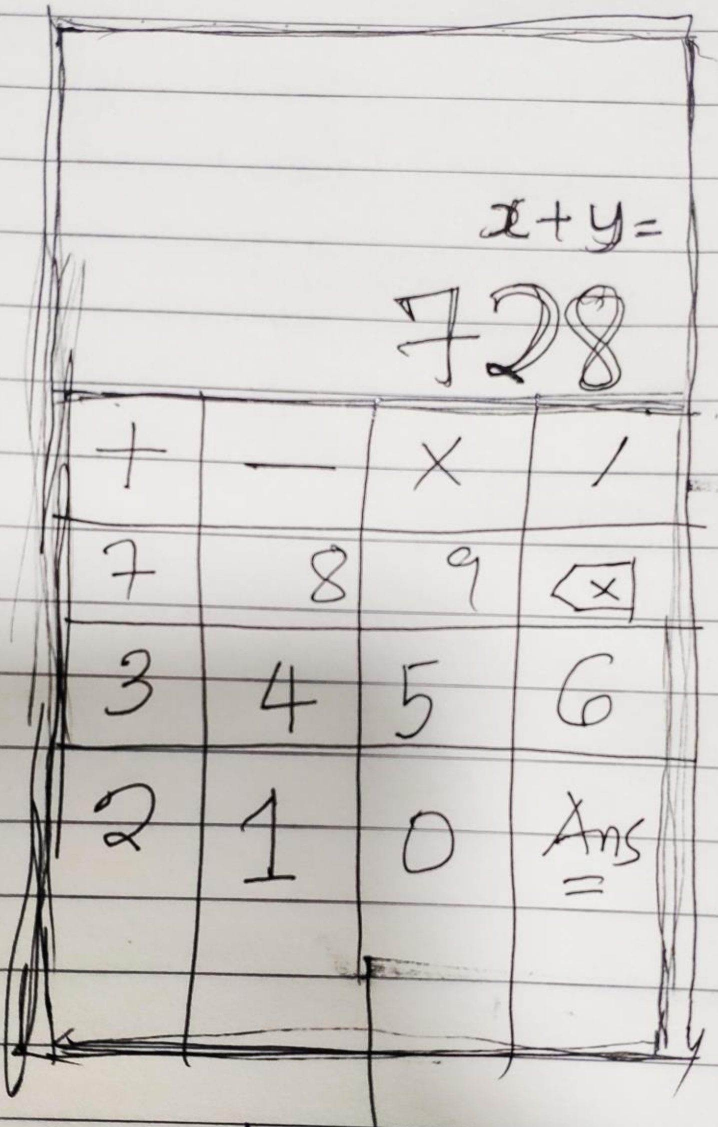
Then we will process the input operation
That the user requested and output an answer.

DETAILED DESIGN

The sketch below demonstrates the basic idea of the app interface.

SKETCH

MEMO NO. ________
DATE / /

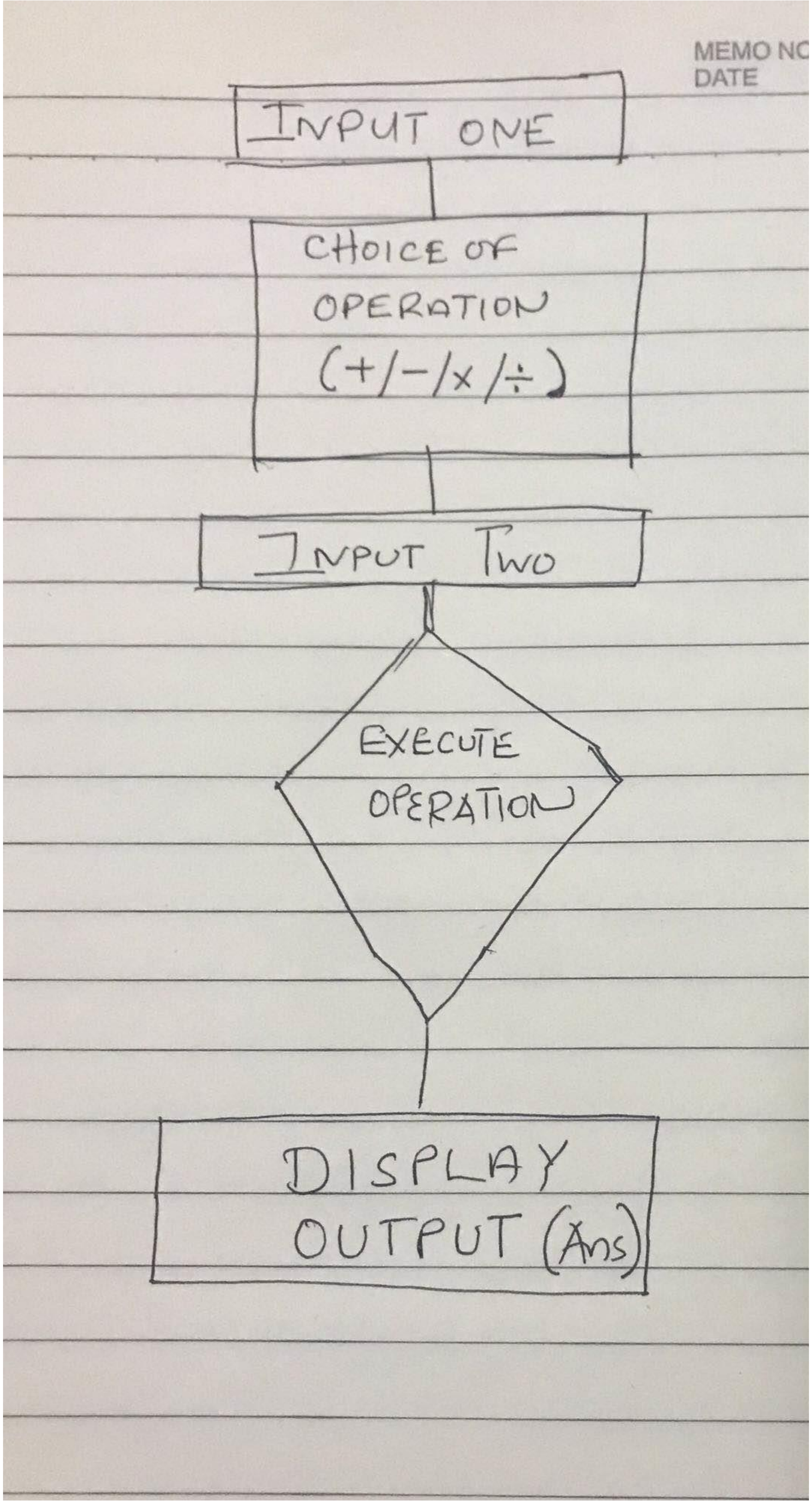


example sketch by:
Nooreelahi Mazumder

From this sketch we should derive a working program.

FLOW CHART

HERE GOES THE FLOW CHART.

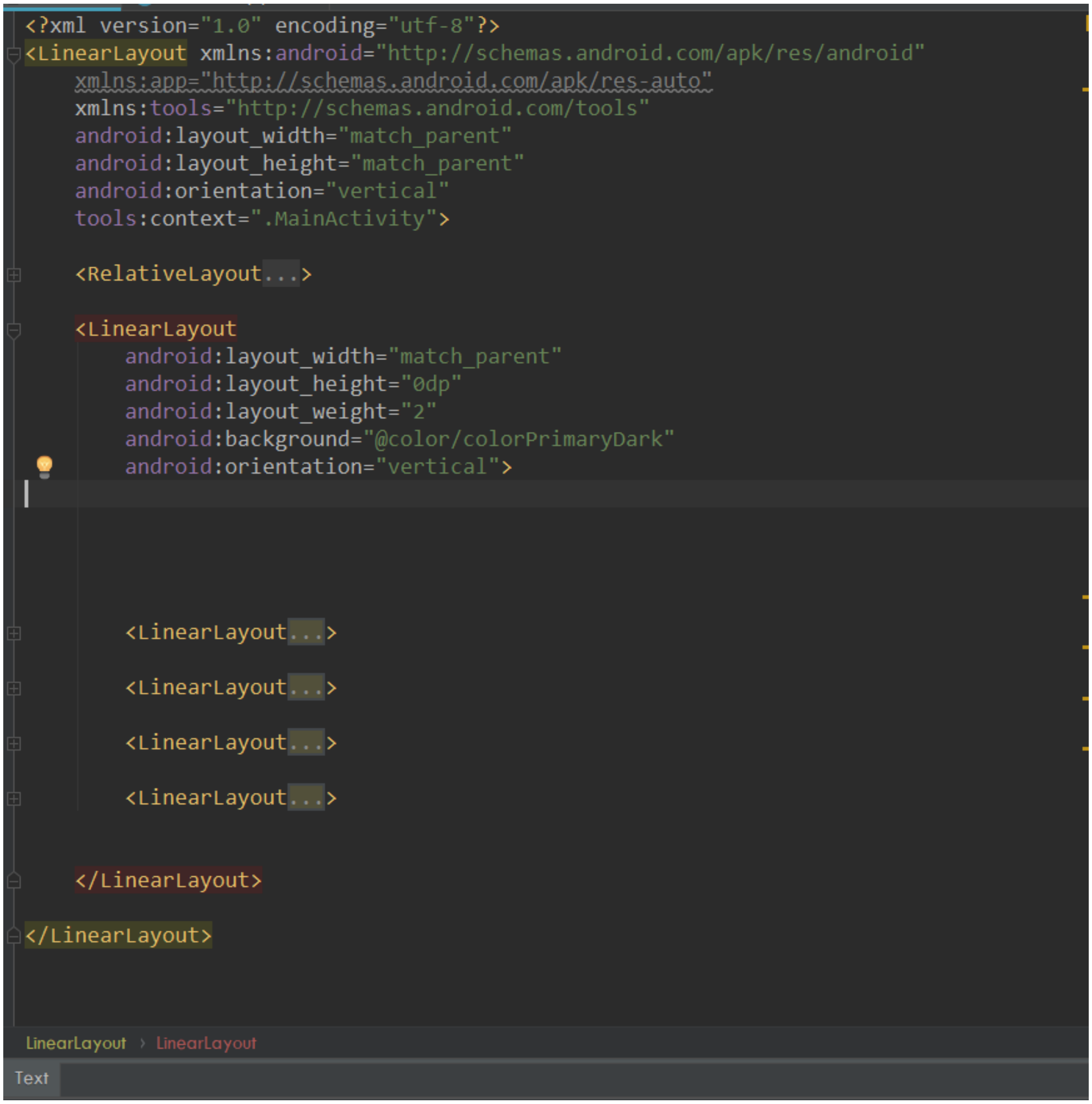

MEMO NO
DATE
INPUT ONE
CHOICE OF OPERATION (+/-/x/÷)
INPUT TWO
EXECUTE OPERATION
DISPLAY OUTPUT (Ans)

PROGRAMMING.

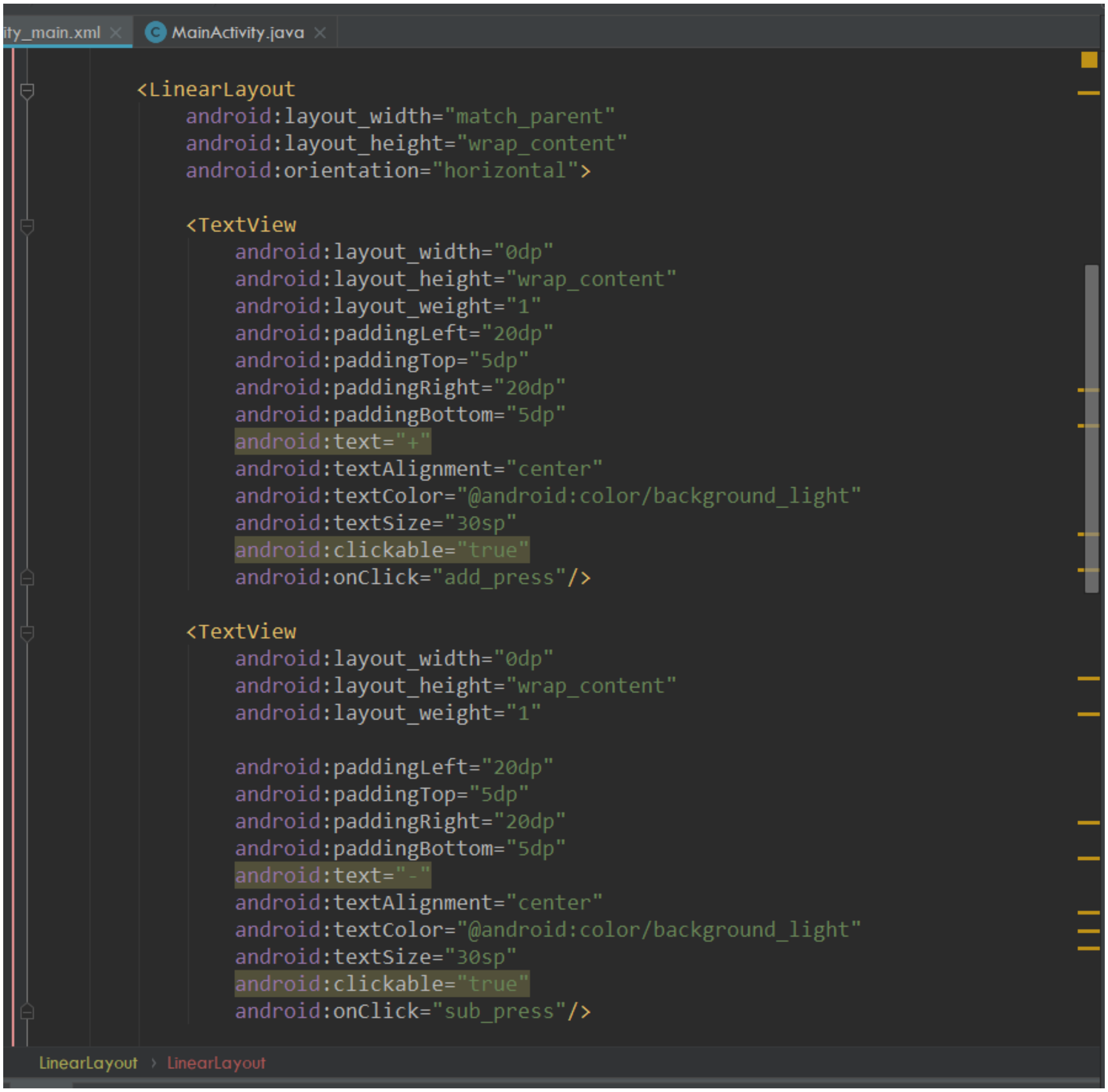
Here are the programs we wrote to create the app.

XML CODE.

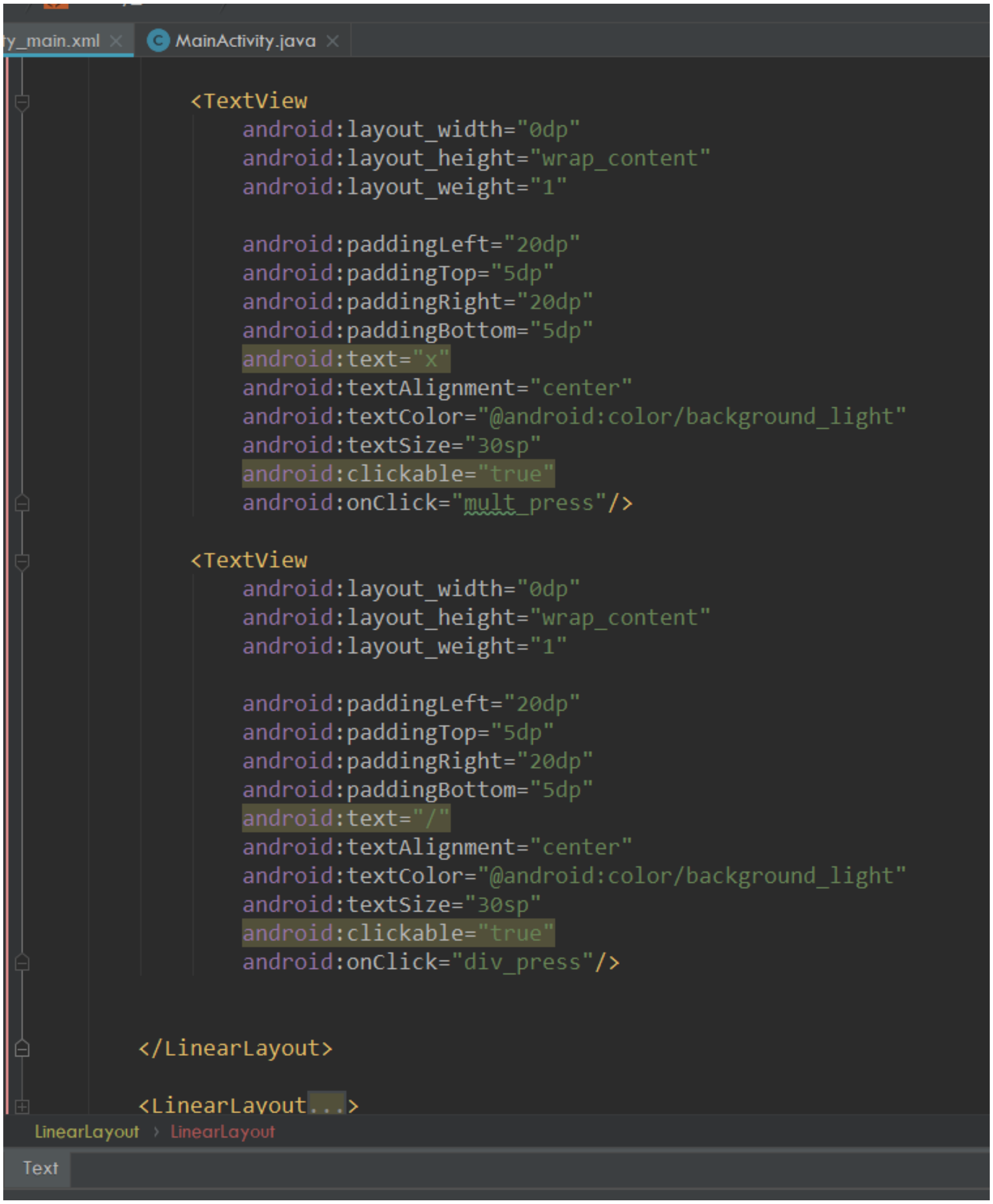
This file defines the graphical interface of the app.



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

    <RelativeLayout...>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="2"
        android:background="@color/colorPrimaryDark"
        android:orientation="vertical">




        <LinearLayout...>

        <LinearLayout...>

        <LinearLayout...>

        <LinearLayout...>


    </LinearLayout>

</LinearLayout>
```

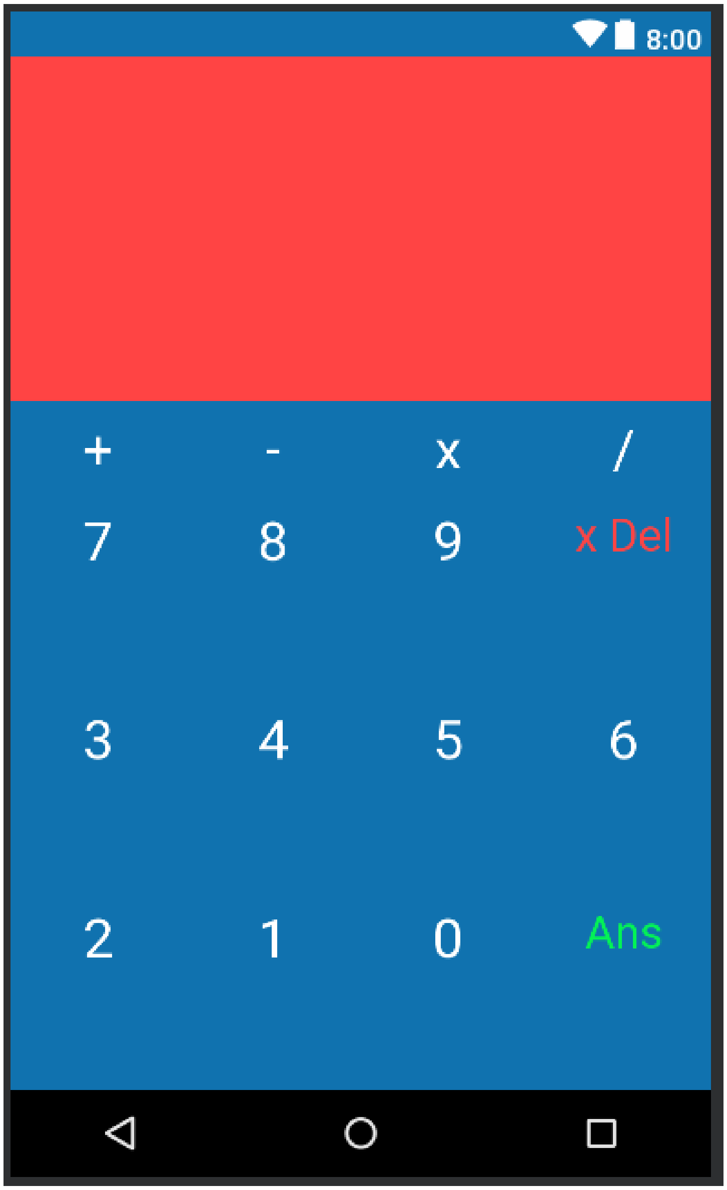


```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <TextView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:paddingLeft="20dp"
        android:paddingTop="5dp"
        android:paddingRight="20dp"
        android:paddingBottom="5dp"
        android:text="+"
        android:textAlignment="center"
        android:textColor="@android:color/background_light"
        android:textSize="30sp"
        android:clickable="true"
        android:onClick="add_press"/>

    <TextView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"

        android:paddingLeft="20dp"
        android:paddingTop="5dp"
        android:paddingRight="20dp"
        android:paddingBottom="5dp"
        android:text="-"
        android:textAlignment="center"
        android:textColor="@android:color/background_light"
        android:textSize="30sp"
        android:clickable="true"
        android:onClick="sub_press"/>
```

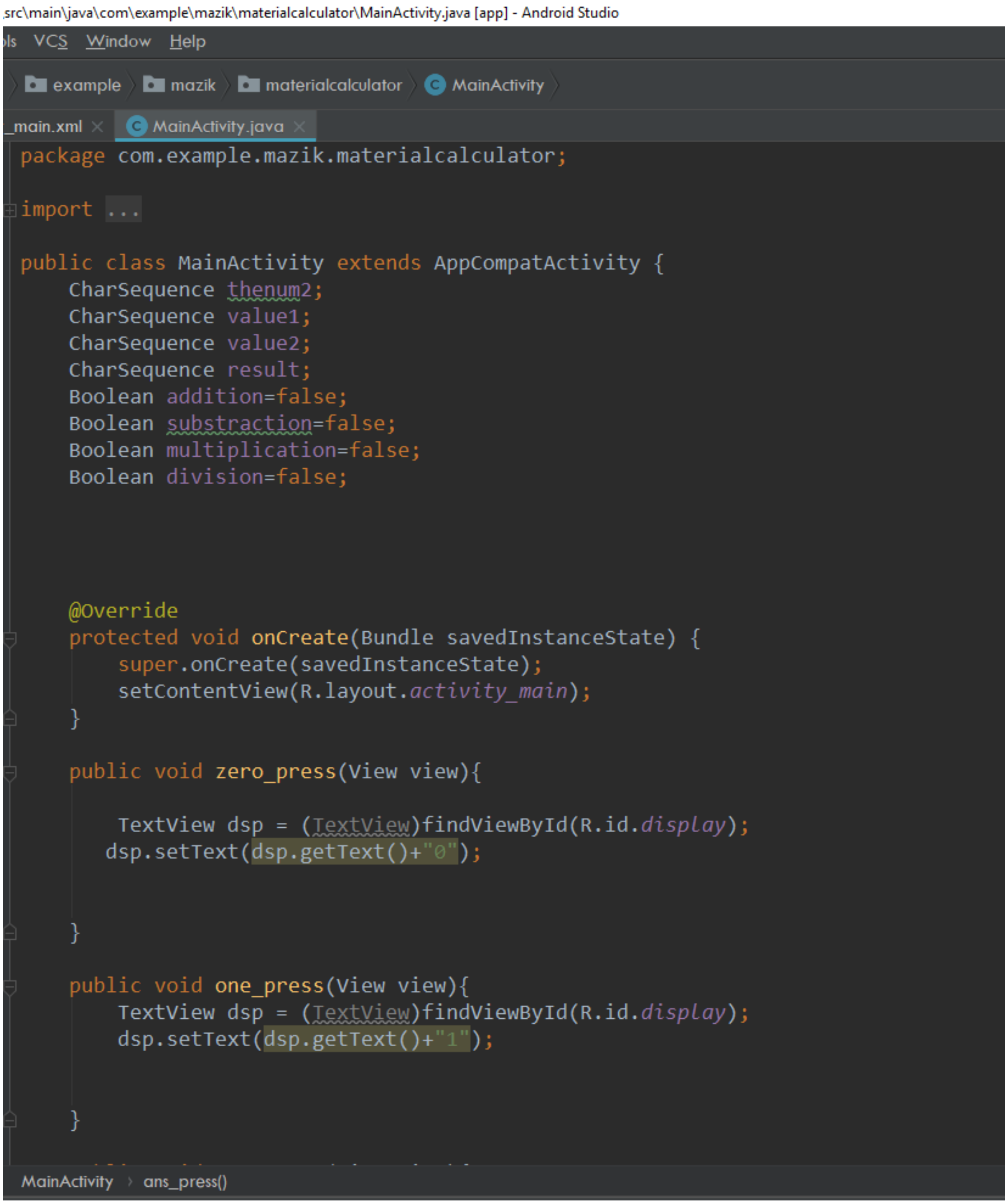


```
        <TextView
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"

            android:paddingLeft="20dp"
            android:paddingTop="5dp"
            android:paddingRight="20dp"
            android:paddingBottom="5dp"
            android:text="x"
            android:textAlignment="center"
            android:textColor="@android:color/background_light"
            android:textSize="30sp"
            android:clickable="true"
            android:onClick="mult_press"/>

        <TextView
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"

            android:paddingLeft="20dp"
            android:paddingTop="5dp"
            android:paddingRight="20dp"
            android:paddingBottom="5dp"
            android:text="/"
            android:textAlignment="center"
            android:textColor="@android:color/background_light"
            android:textSize="30sp"
            android:clickable="true"
            android:onClick="div_press"/>


    </LinearLayout>

    <LinearLayout...>
```

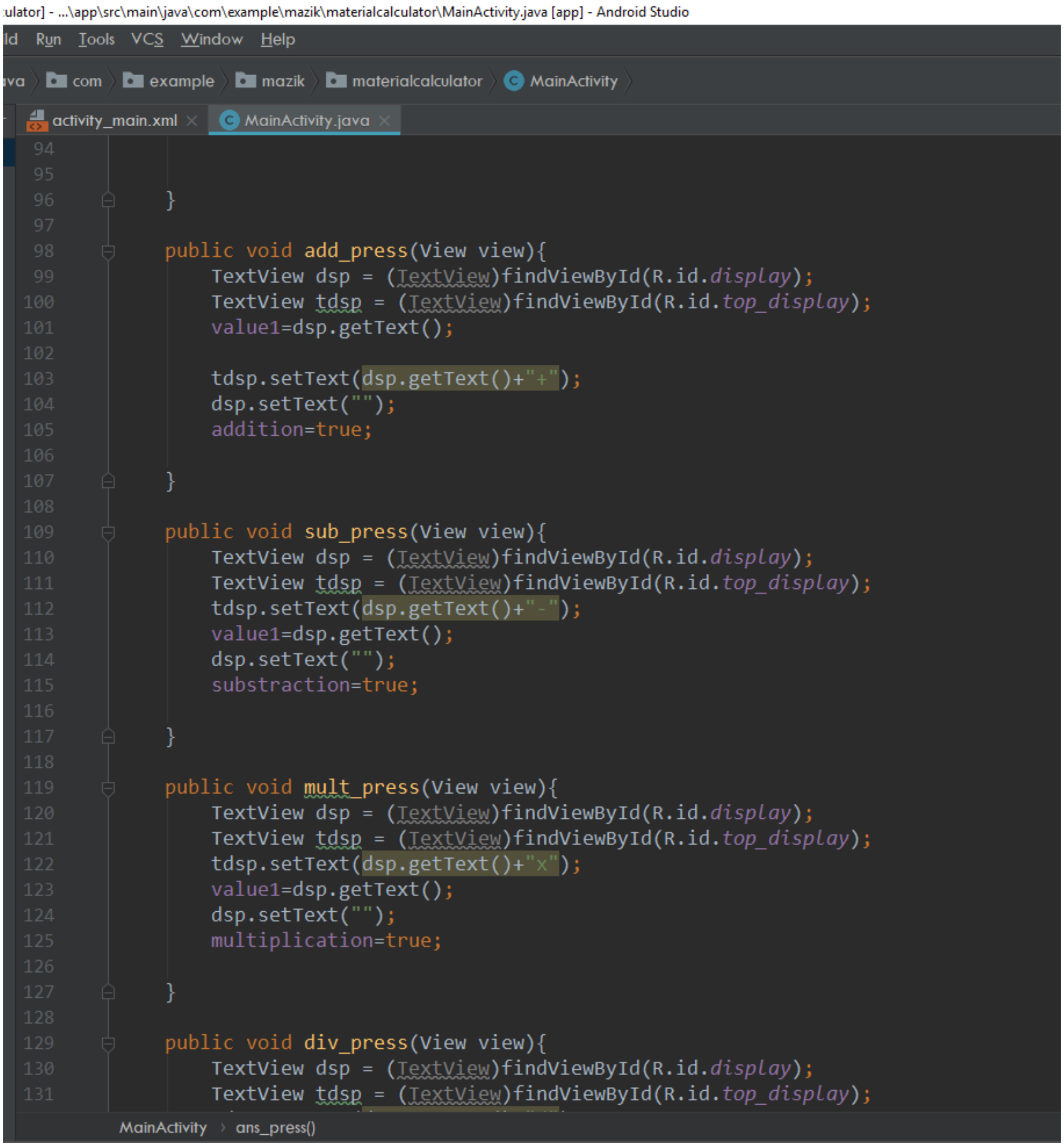
This is the final look of the GRAPHICAL INTERFACE.



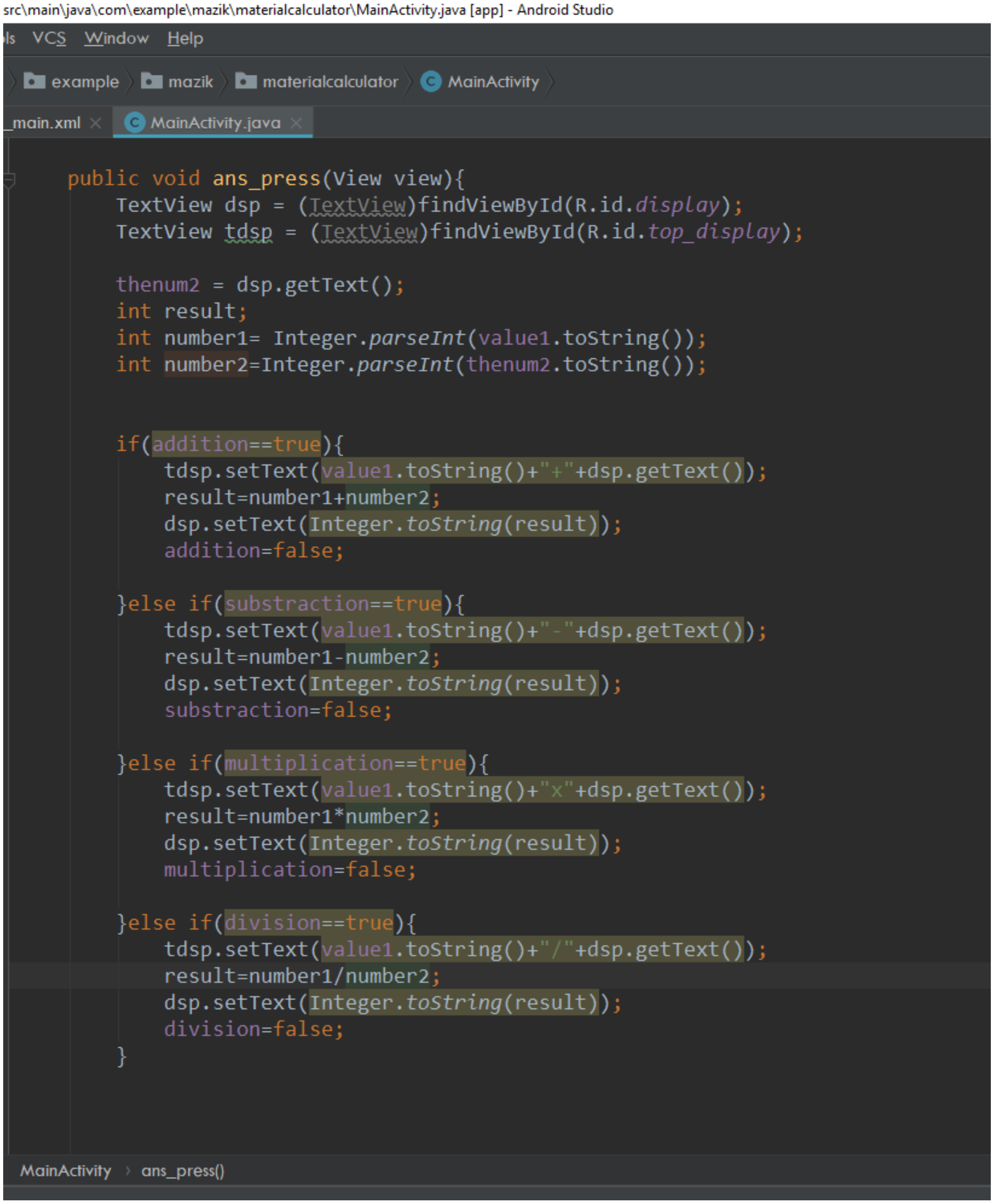
JAVA CODE



```java
package com.example.mazik.materialcalculator;

import ...

public class MainActivity extends AppCompatActivity {
    CharSequence thenum2;
    CharSequence value1;
    CharSequence value2;
    CharSequence result;
    Boolean addition=false;
    Boolean substraction=false;
    Boolean multiplication=false;
    Boolean division=false;




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

    public void zero_press(View view){

        TextView dsp = (TextView)findViewById(R.id.display);
       dsp.setText(dsp.getText()+"0");


    }

    public void one_press(View view){
        TextView dsp = (TextView)findViewById(R.id.display);
        dsp.setText(dsp.getText()+"1");


    }
```


ulator] - ...\app\src\main\java\com\example\mazik\materialcalculator\MainActivity.java [app] - Android Studio
ld Run Tools VCS Window Help
va com example mazik materialcalculator MainActivity
activity_main.xml
MainActivity.java
MainActivity › ans_press()

```


    }

    public void add_press(View view){
        TextView dsp = (TextView)findViewById(R.id.display);
        TextView tdsp = (TextView)findViewById(R.id.top_display);
        value1=dsp.getText();

        tdsp.setText(dsp.getText()+"+");
        dsp.setText("");
        addition=true;

    }

    public void sub_press(View view){
        TextView dsp = (TextView)findViewById(R.id.display);
        TextView tdsp = (TextView)findViewById(R.id.top_display);
        tdsp.setText(dsp.getText()+"-");
        value1=dsp.getText();
        dsp.setText("");
        substraction=true;

    }

    public void mult_press(View view){
        TextView dsp = (TextView)findViewById(R.id.display);
        TextView tdsp = (TextView)findViewById(R.id.top_display);
        tdsp.setText(dsp.getText()+"x");
        value1=dsp.getText();
        dsp.setText("");
        multiplication=true;

    }

    public void div_press(View view){
        TextView dsp = (TextView)findViewById(R.id.display);
        TextView tdsp = (TextView)findViewById(R.id.top_display);
```



src\main\java\com\example\mazik\materialcalculator\MainActivity.java [app] - Android Studio
ls VCS Window Help
example
mazik
materialcalculator
MainActivity
_main.xml
MainActivity.java
MainActivity › ans_press()

```java
    public void ans_press(View view){
        TextView dsp = (TextView)findViewById(R.id.display);
        TextView tdsp = (TextView)findViewById(R.id.top_display);

        thenum2 = dsp.getText();
        int result;
        int number1= Integer.parseInt(value1.toString());
        int number2=Integer.parseInt(thenum2.toString());


        if(addition==true){
            tdsp.setText(value1.toString()+"+"+dsp.getText());
            result=number1+number2;
            dsp.setText(Integer.toString(result));
            addition=false;

        }else if(substraction==true){
            tdsp.setText(value1.toString()+"-"+dsp.getText());
            result=number1-number2;
            dsp.setText(Integer.toString(result));
            substraction=false;

        }else if(multiplication==true){
            tdsp.setText(value1.toString()+"x"+dsp.getText());
            result=number1*number2;
            dsp.setText(Integer.toString(result));
            multiplication=false;

        }else if(division==true){
            tdsp.setText(value1.toString()+"/"+dsp.getText());
            result=number1/number2;
            dsp.setText(Integer.toString(result));
            division=false;
        }
```

TESTING AND DEBUGGING

Now we will test our app in an android device.

We will install it.

First we have to allow installation from unknown sources in our device this will allow us to deploy unsigned apps into our phone.

We will also enable developer options in our phone so as to allow the app to be run and perform on the spot changes to the code we have written and be able to make instant changes to our device without running the whole program.

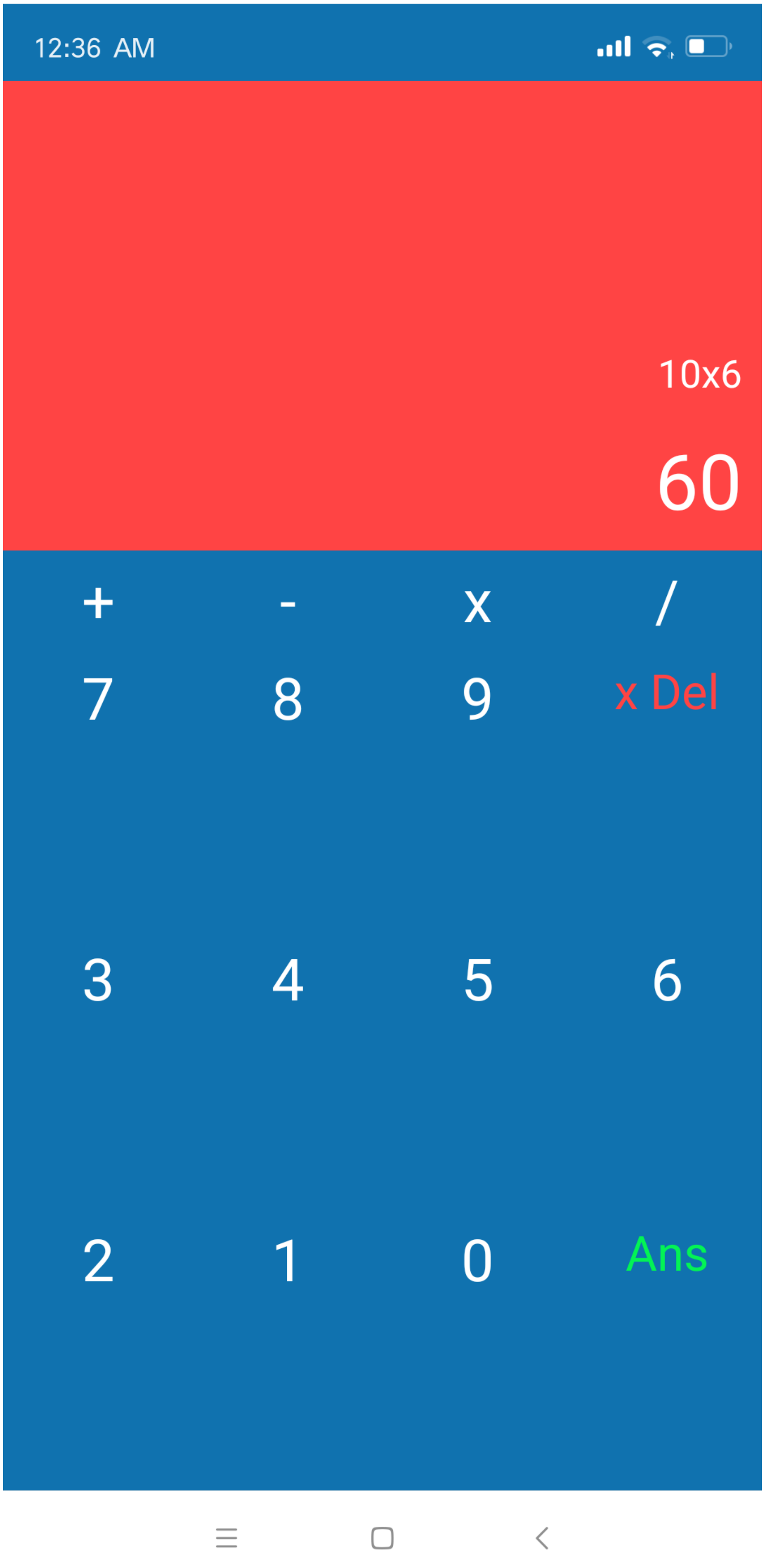
After installation is done.

We will test it on a real device.

For the test we will use

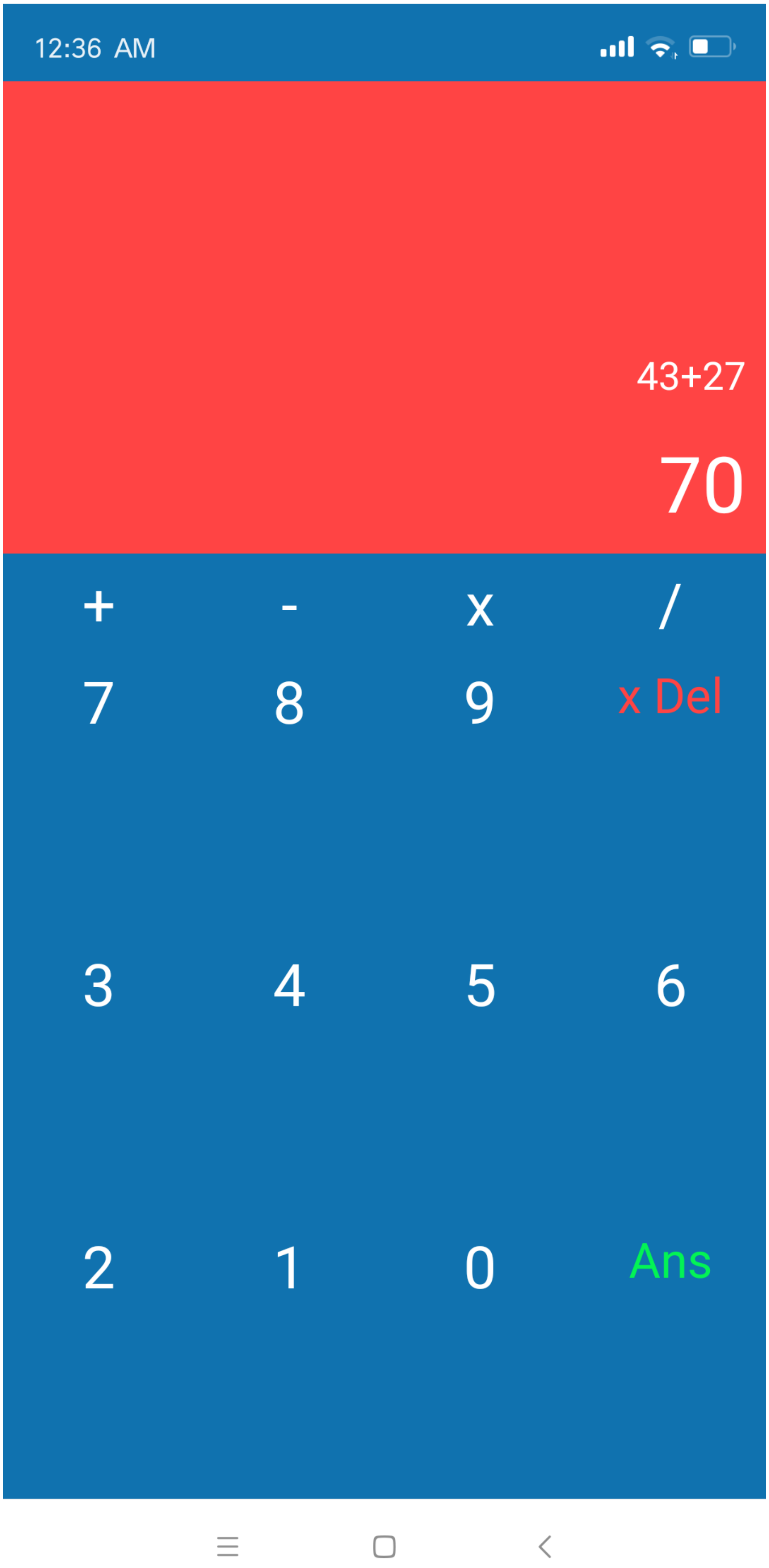
- Xiaomi 8
- Vivo y66

The following are Xiaomi screen shots.



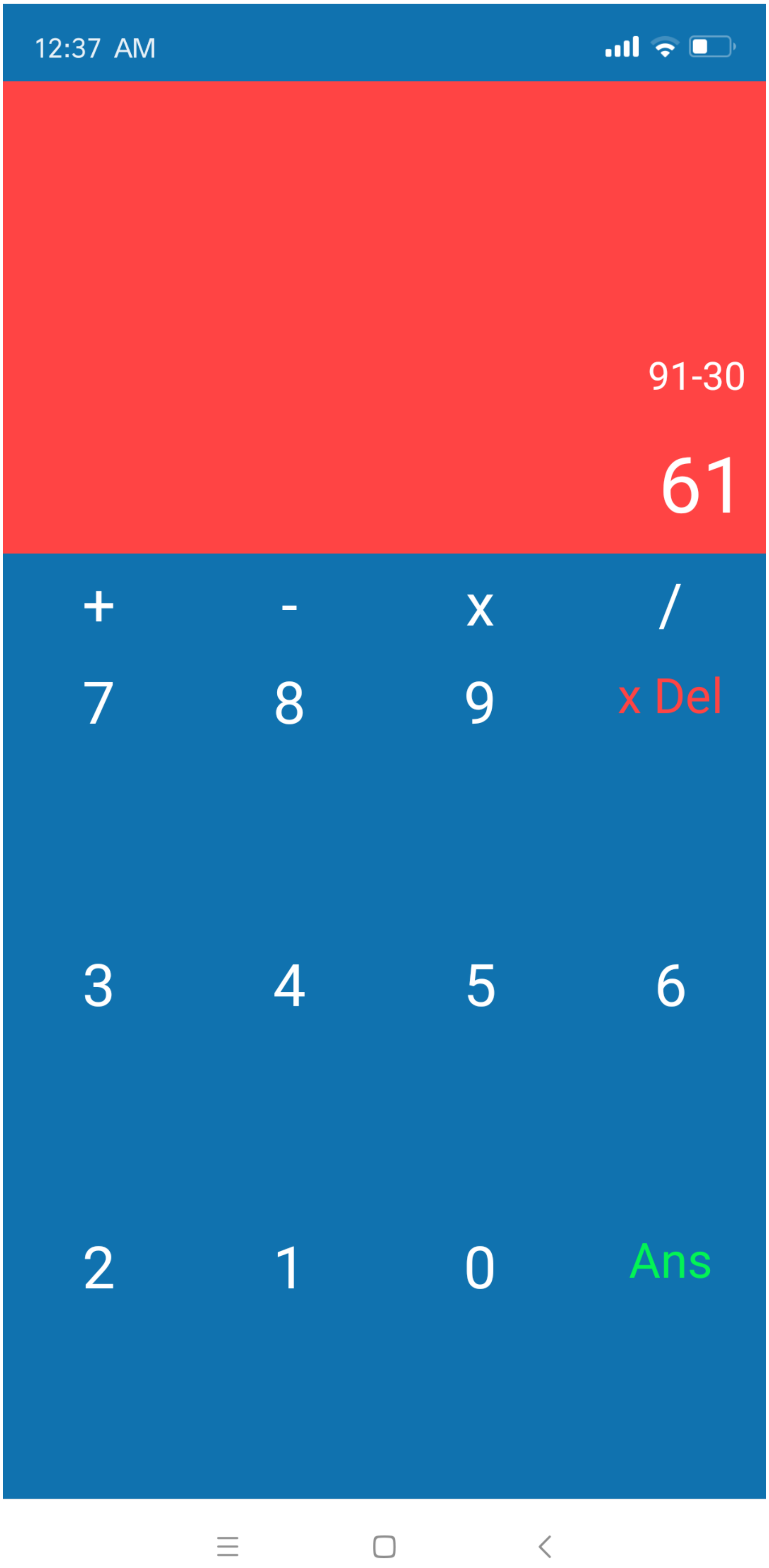
10x6

60



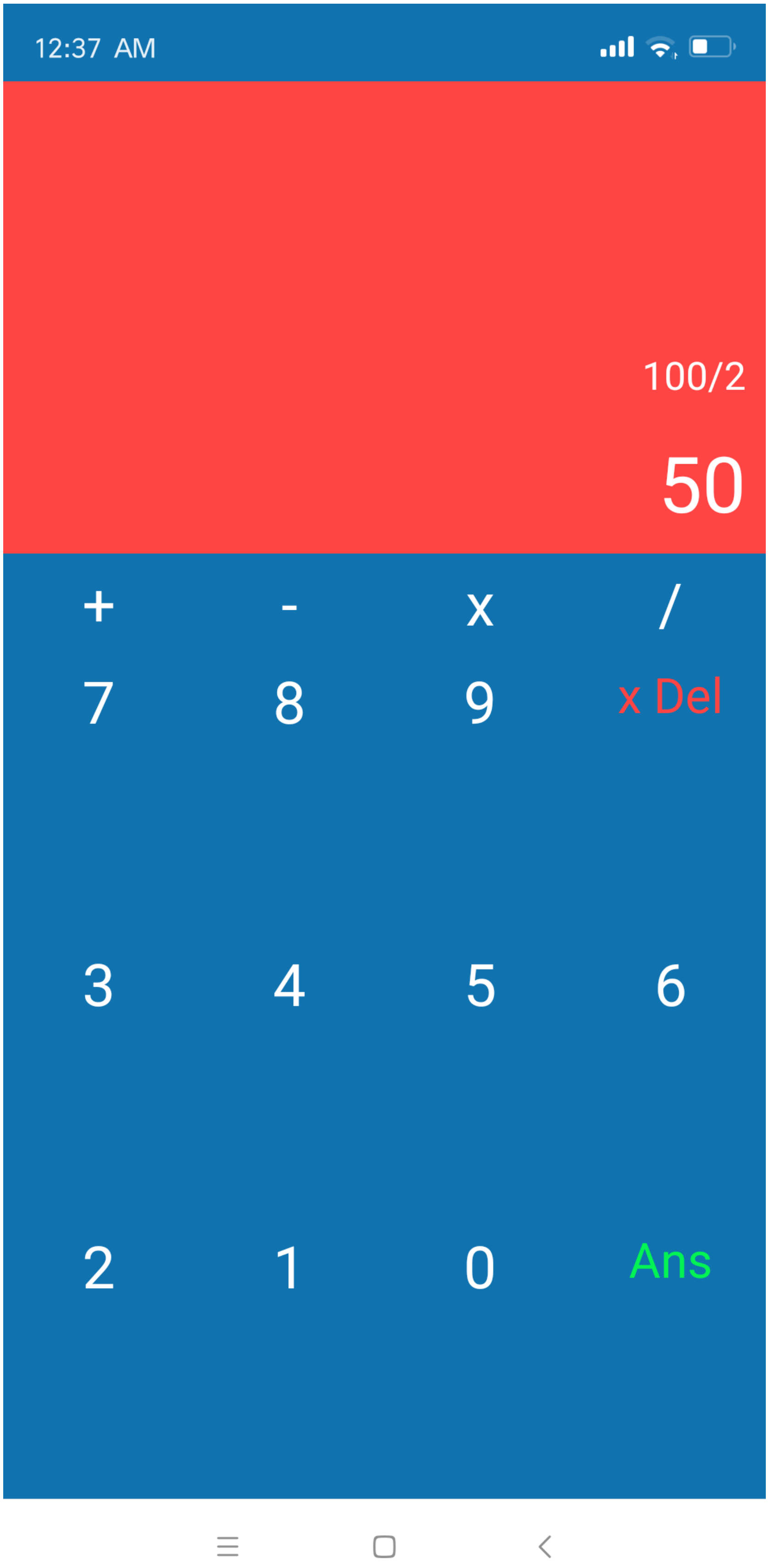
43+27

70



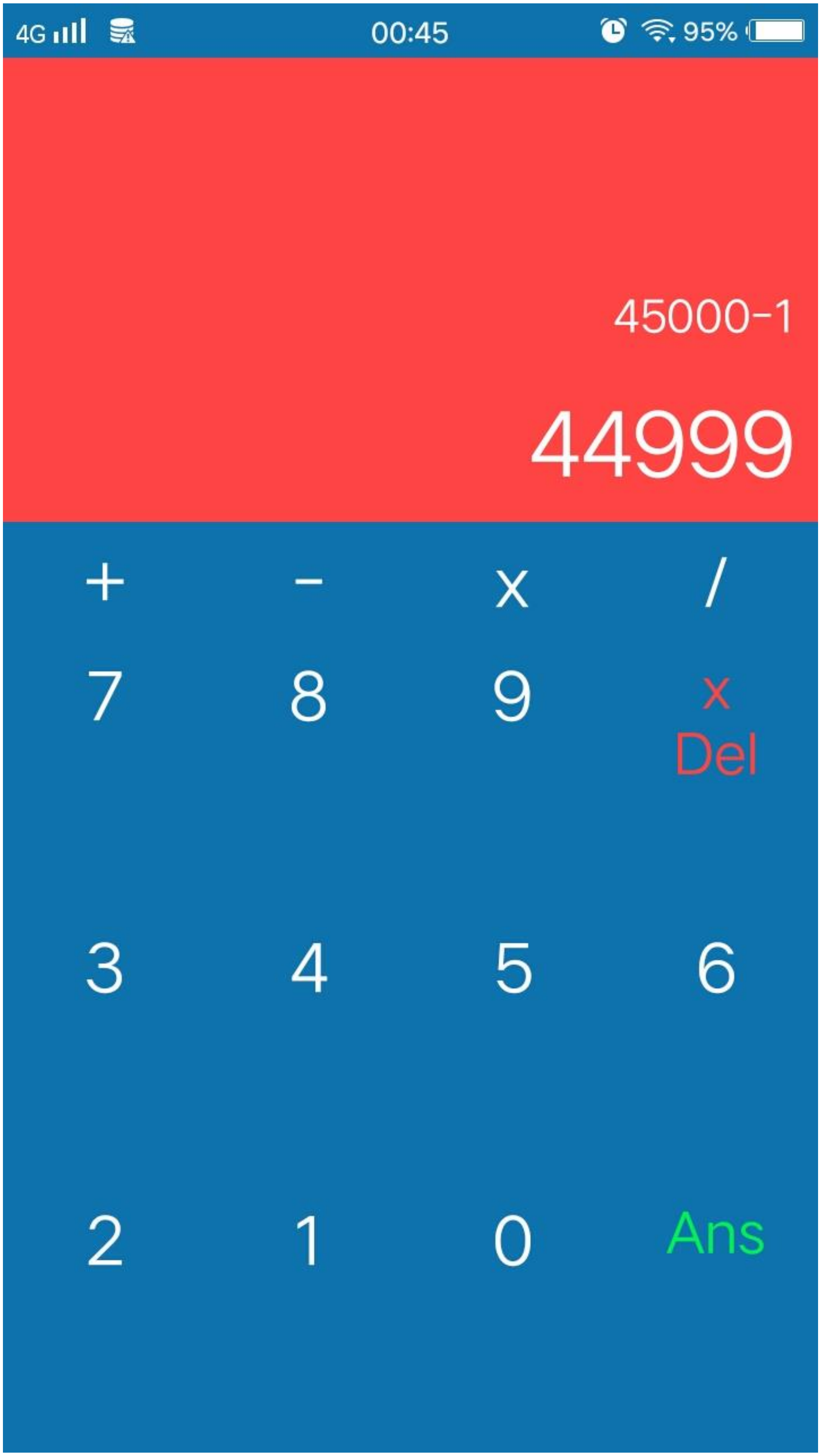
91-30

61

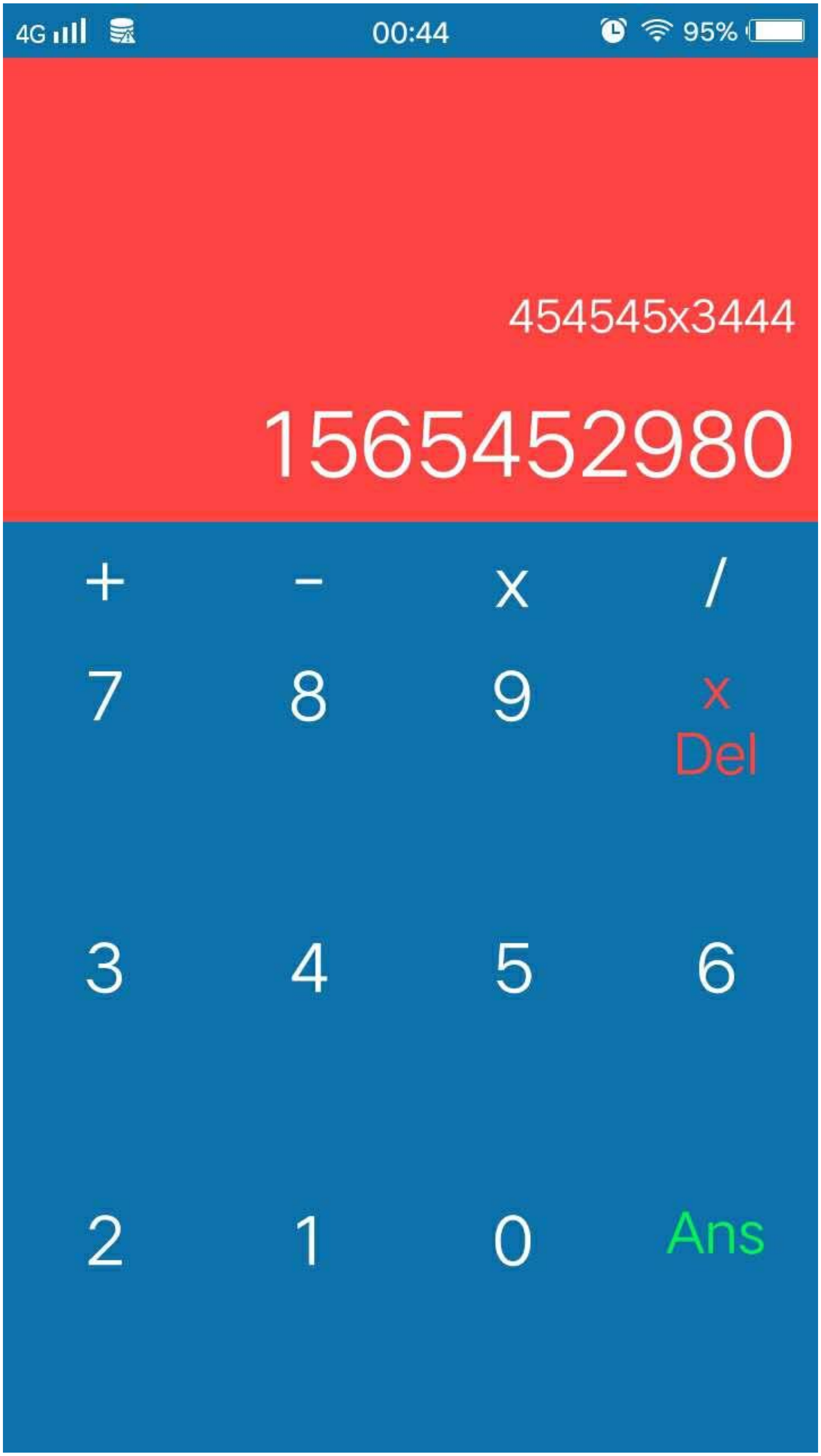


```
100/2

50
```

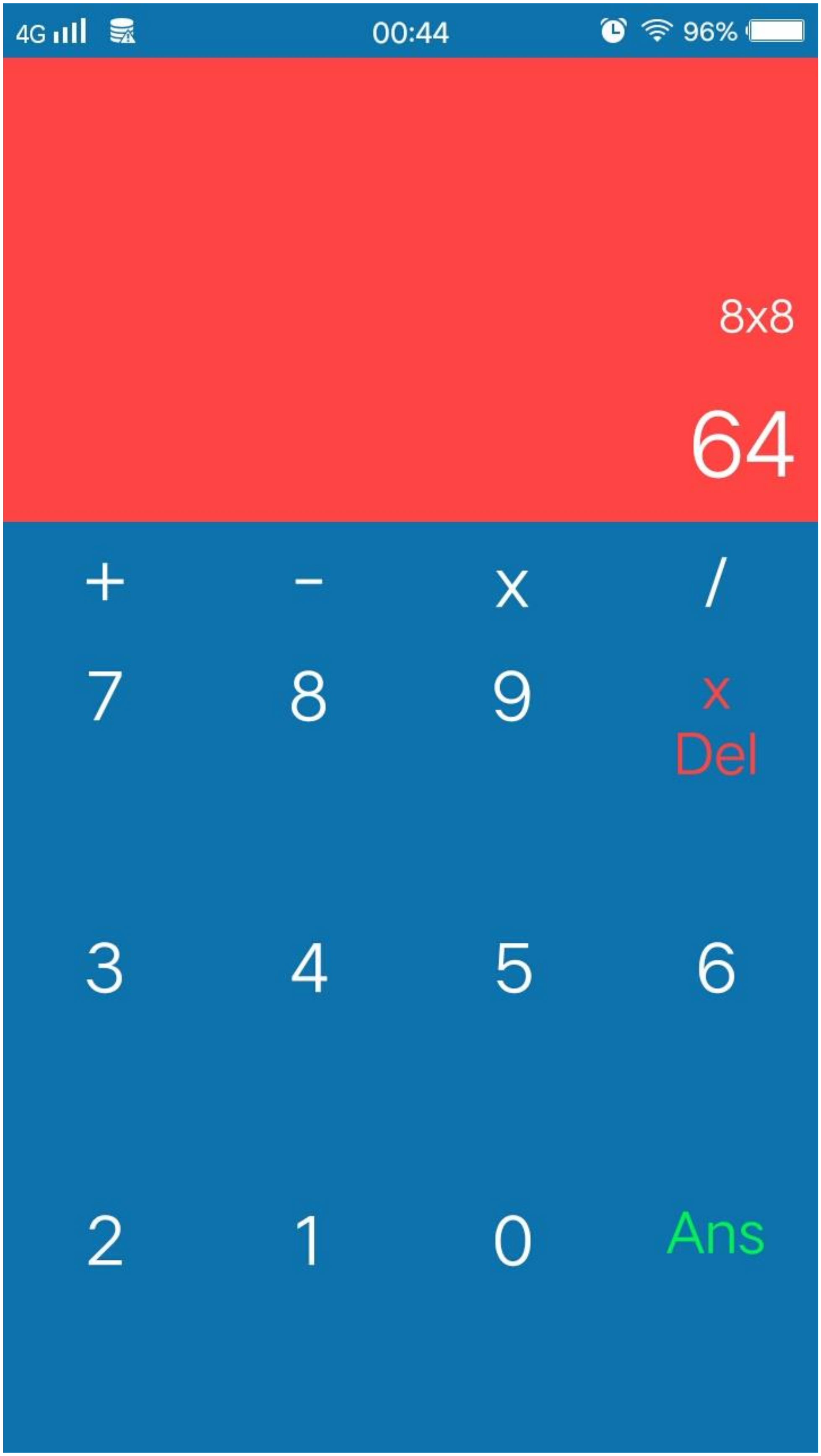
Vivo y66 screenshots.



```
45000-1
44999
```



```
454545x3444
1565452980
```



8x8

64

Summary

Generally the app can solve basic operations
As described on the screenshots.

Its interface helps the user input data easier than using terminal console.

Our app can help you do your calculation problems with ease.

For simplicity we have only showed here the most important parts of the program. There are more files behind the building of this project.

We have pushed the app on to **GITHUB** so as everyone can be able to view the source code and do alterations of their desire to the project.

If you desire to write, alter or learn more about Android via this simple project you can visit

github.com/eighthwnder/MaterialCalculatorApp

To see the whole code and run it in your own computer and smartphone.

REFEERENCES.

Material design principles - material.io/design

Android documentation – developer.android.com

Java – teach yourself java in 24hours book

And a lot of YouTube tutorials.