

Mobile Application Development PROJECT (102261)

FACULTY: Mr.S.Prabhakaran

PROJECT NAME: CALCULATOR APP

CLASS: 406

SUBMITTED BY:

Shubham Panday(RA1711003010003)

Shivam Bhosale(RA1711003010257)

Rishma Mitra Dhar(RA1711003010065)

Soumya Awasthi(RA1711003010233)

TABLE OF CONTENTS

- 1. TITLE**
- 2. ACKNOWLEDGEMENT**
- 3. ABSTRACT**
- 4. INTRODUCTION**
- 5. WALK THROUGH THE APP**
- 6. SYSTEM REQUIREMENTS**
- 7. CODE**
- 8. FLOW CHART**
- 9. SCREEN SHOTS**
- 10. CONCLUSION**
- 11. REFERENCES**

TITLE : CALCULATOR APPLICATION

**USED FOR PERFORMING BASIC
MATHEMATICAL OPERATIONS**

Acknowledgment

It gives immense pleasure in bringing out this synopsis of the project entitled
“The Calculator App”

Firstly we would like to thank our Professor Mr. Prabhakaran Soundar who gave us his valuable suggestions and ideas when we were in need of them. He encouraged us to work on this project.

We are also grateful to our Institution for giving us the opportunity to work with them and providing us the necessary resources for the project. We would also thank all of them who helped us to complete this project.

We are immensely grateful to all involved in this project as without their inspiration and valuable suggestion it would not have been possible to develop the project within the prescribed time.

With Sincere Thanks,

Shubham Panday(RA1711003010003)

Shivam Bhosale(RA1711003010257)

Rishma Mitra Dhar(RA1711003010065)

Soumya Awasthi(RA1711003010233)

Abstract

This Calculator app is an application developed for android cells phones and tablets. It enables the users to perform fundamental mathematical operations such as addition, subtraction, multiplication, division and modulo on their phone. Being installed on phones, it always remains with the user, helps in daily life calculations and works as a typical android calculator app.

Comparison:

The comparison between existing traditional calculators and the proposed Calculator app for Android can be done in the following points:

Size: The separate calculators are larger in size due to use of their own circuit board whereas an android calculator is installed in cell phones.

Power: A small battery power of a cell phone is enough for running an android calculator but traditional calculators consume comparatively higher power.

Accuracy: The accuracy of both the calculators depends upon their programming. Calculators can be designed for the desired accuracy.

Cost: Android Calculator, being an application, can be downloaded free of cost and installed on phone. The traditional calculators have their own hardware. As a result of this, they need to be purchased.

Ease and comfort: Most of the android cell phones are of touch screen type. So, the calculator app Android is easy and comfortable to use.

Introduction

Fundamental mathematical calculations are required to be performed in every step of human life. In most of the places such as in shops, hotels, medicals, schools etc., separate calculators are used for calculation. But, these calculators, being larger in size and needing extra power, can't be carried all the time with us. This creates the necessity of a mobile calculator which is always with us.

Since android applications are the most compatible ones and easily available, android cell phones and tablets are the most widely used devices all over the world. The application gives fast and accurate results.

This application consumes very small power of your cell phone. They offer several kinds of facilities and one of them is a calculator which is a fundamental application of each phone. Looking at the growing use of android phones, it can be said that a Calculator app as Android Project has good scope in today's technological world.

Features:

- It is easy and comfortable to use.
- The application can be easily downloaded and installed on any android cell phone and tablet.
- It provides the facility to perform fundamental mathematical operations.
- The application gives fast and accurate results.
- This application consumes very small power of your cell phone.

A Walk Through the App

- The Android Application has an interface which can be easily accessible even by a lay-man.
- The first row consists of an input line followed by another where the user is supposed to give the input numbers for the operation to be carried out within them.
- The next five rows consists of different options for operation to be performed.
- Following operations can be performed:-
 1. Addition
 2. Subtraction
 3. Multiplication
 4. Division
 5. Modulos
- After entering the input numbers the user can choose to perform any of the above operations, Simply by clicking on it.
- The result of the same would be declared just below the input lines.
- Therefore, the user can perform all the necessary calculations using this application.

System Requirements

We are creating a calculator app and for making this app and for it to be fully functional we will be using Android Studio for developing the app from scratch.

Android Studio is Google's officially supported IDE for developing Android apps. This IDE is based on IntelliJ IDEA, which offers a powerful code editor and developer tools. Android Studio 3.2.1 includes the following features:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Instant Run to push changes to your running app without building a new APK
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to help you catch performance, usability, version compatibility, and other problems
- C++ and NDK support.
- Plugin Architecture for extending Android Studio via plugin.

Google provides Android Studio for Windows, Mac OS and Linux based operating systems. You can download Android Studio from the Android Studio home page, but before downloading make sure you have the following system requirements.

Windows

- Microsoft Windows 7/8/10 (32-bit or 64-bit).
- 3 GB RAM minimum, 8 GB RAM recommended (plus 1 GB for the Android Emulator).
- 2 GB of available disk space minimum, 4 GB recommended (500 MB for IDE plus 1.5 GB for Android SDK and emulator system image).
- 1280 x 800 minimum screen resolution.

Mac OS

- Mac OS X 10.10 (Yosemite) or higher, up to 10.13 (High Sierra).
- 3 GB RAM minimum, 8 GB RAM recommended (plus 1 GB for the Android Emulator).
- 2 GB of available disk space minimum, 4 GB recommended (500 MB for IDE plus 1.5 GB for Android SDK and emulator system image).
- 1280 x 800 minimum screen resolution.

Linux OS

- GNOME or KDE desktop. Tested on Ubuntu 14.04 LTS, Trusty Tahr (64-bit distribution capable of running 32-bit applications).
- 64-bit distribution capable of running 32-bit applications.
- GNU C Library (glibc) 2.19 or later.
- 3 GB RAM minimum, 8 GB RAM recommended (plus 1 GB for the Android Emulator).
- 2 GB of available disk space minimum, 4 GB recommended (500 MB for IDE plus 1.5 GB for Android SDK and emulator system image).
- 1280 x 800 minimum screen resolution.

System Requirements for Running the Application

After creating the app in order to run the app we need to download the emulator on Android Studio itself. In this we have to download the virtual machine part of a phone and have to run the app on that virtual machine.

The minimum requirements for running the app are mainly the android version on the phone and that should be equal to or above Android 4.0 ie the ice cream sandwich version of android. If the phone has android 4.0 then it will automatically have the specifications to run the calculator app.

CODE

activity_xml file

This file contains the layout of the calculator app. In this particular app we have used Linear Layout and have set the orientation to vertical, by doing this we can place all the Text Views and Buttons easily under one another.

```
<?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"
    android:background="@drawable/gradient1">

    <EditText

        android:id="@+id/number1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```

        android:ems="10"

        android:hint="Number 1"

        android:inputType="numberSigned" />
<EditText

        android:id="@+id/number2"

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:ems="10"

        android:hint="Number2"

        android:inputType="numberSigned" />
<TextView

        android:id="@+id/Result"

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:text="Result"

        android:textSize="25sp" />
<Button

        android:id="@+id/add"

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

```

```

        android:text="ADD" />
<Button
    android:id="@+id/subtract"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="SUBTRACT" />
<Button
    android:id="@+id/multiply"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="MULTIPLY" />
<Button
    android:id="@+id/divide"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="DIVIDE" />
<Button
    android:id="@+id/modulo"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"

```

```
        android:text="MODULO" /></LinearLayout>
```

MainActivity.java

In the MainActivity.java file we create the functions of the layout we created in the activity_xml file. We describe the function of each TextView and Button here.

```
package com.example.calculator1;

import android.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends
AppCompatActivity {

    TextView result;

    EditText number1, number2;

    Button add,subtract,multiply,divide,modulo;

    float result_num;

    int num1,num2;

    @Override
```

```

        protected void onCreate(Bundle
savedInstanceState) {

            super.onCreate(savedInstanceState);

            setContentView(R.layout.activity_main);

            result =
(TextView) findViewById(R.id.Result);

            number1 =
(EditText) findViewById(R.id.number1);

            number2 =
(EditText) findViewById(R.id.number2);

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

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

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

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

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


            add.setOnClickListener(new
View.OnClickListener() {

                @Override

                public void onClick(View v){

```

```
num1=Integer.parseInt(number1.getText().toString())  
);
```

```
num2=Integer.parseInt(number2.getText().toString())  
);
```

```
        result_num=num1+num2;
```

```
result.setText(String.valueOf(result_num));
```

```
    }
```

```
});
```

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

```
            @Override
```

```
            public void onClick(View v){
```

```
num1=Integer.parseInt(number1.getText().toString())  
);
```

```
num2=Integer.parseInt(number2.getText().toString())  
);
```

```
        result_num=num1-num2;
```



```

result.setText(String.valueOf(result_num));

        }

    });

    multiply.setOnClickListener(new
View.OnClickListener() {

        @Override

        public void onClick(View v){

num1=Integer.parseInt(number1.getText().toString()
);

num2=Integer.parseInt(number2.getText().toString()
);

        result_num=num1*num2;

result.setText(String.valueOf(result_num));

        }

    });

    divide.setOnClickListener(new
View.OnClickListener() {

        @Override

```

```

        public void onClick(View v){

num1=Integer.parseInt(number1.getText().toString()
);

num2=Integer.parseInt(number2.getText().toString()
);

        result_num= num1/num2;

result.setText(String.valueOf(result_num));

        }

});

modulo.setOnClickListener(new
View.OnClickListener() {

        @Override

        public void onClick(View v){

num1=Integer.parseInt(number1.getText().toString()
);

num2=Integer.parseInt(number2.getText().toString()
);

```

```

        result_num= num1%num2;

        result.setText(String.valueOf(result_num));

    }

});

}

}

```

gradient.xml

This is an extra file we have added for the background gradient in the app.

```

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

<selector android:shape="rectangle"

xmlns:android="http://schemas.android.com/apk/res/android">

    <item>

        <shape>

            <gradient

                android:startColor="#FFC400"

                android:endColor="#FF6F00"

                android:angle="90"

            />

```

```
        </shape>
```

```
    </item>
```

```
</selector>
```

AndroidManifest.xml

This file contains information of the packages, including the components of the application.

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

```
<manifest
```

```
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    package="com.example.calculator1">
```

```
        <application
```

```
            android:allowBackup="true"
```

```
            android:icon="@mipmap/ic_launcher"
```

```
            android:label="@string/app_name"
```

```
            android:roundIcon="@mipmap/ic_launcher_round"
```

```
            android:supportsRtl="true"
```

```
            android:theme="@style/AppTheme">
```

```
                <activity android:name=".MainActivity">
```

```
                    <intent-filter>
```

```
        <action
android:name="android.intent.action.MAIN" />

        <category
android:name="android.intent.category.LAUNCHER" />

    </intent-filter>

</activity>

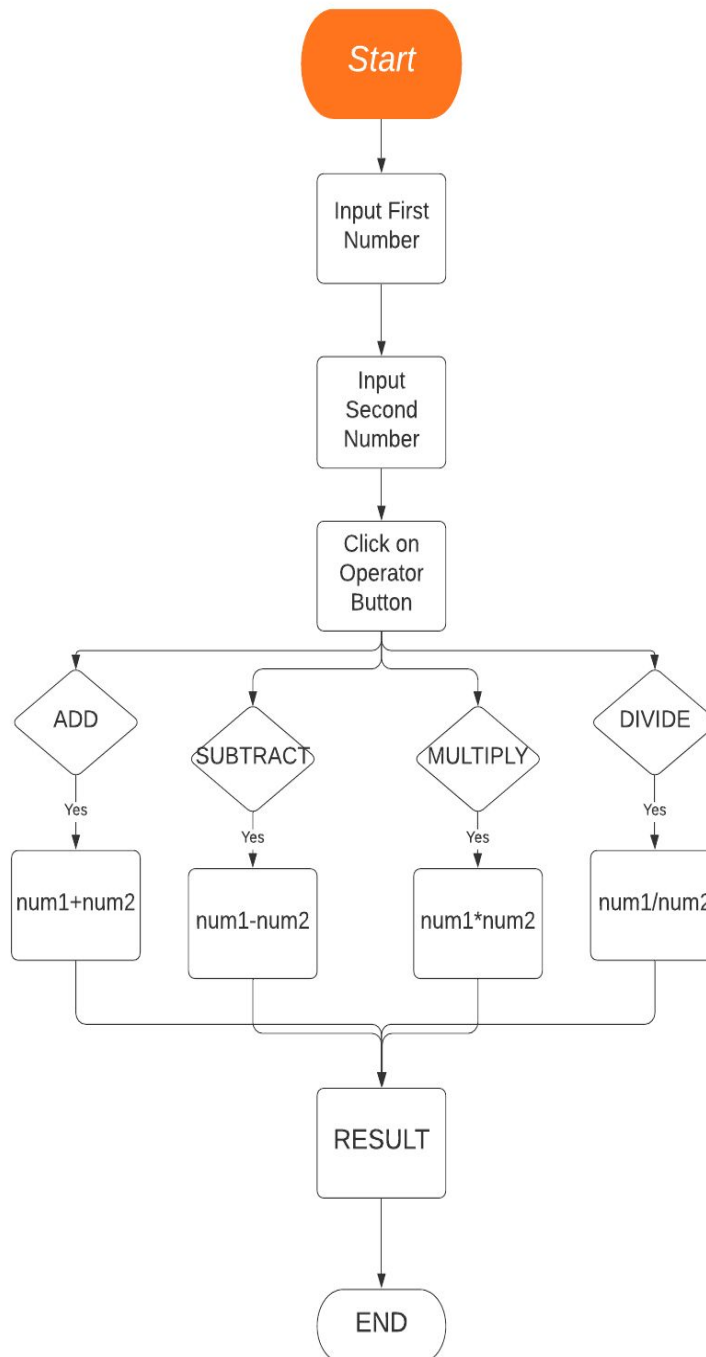
</application>

</manifest>
```

FlowChart

CalcFlowchart

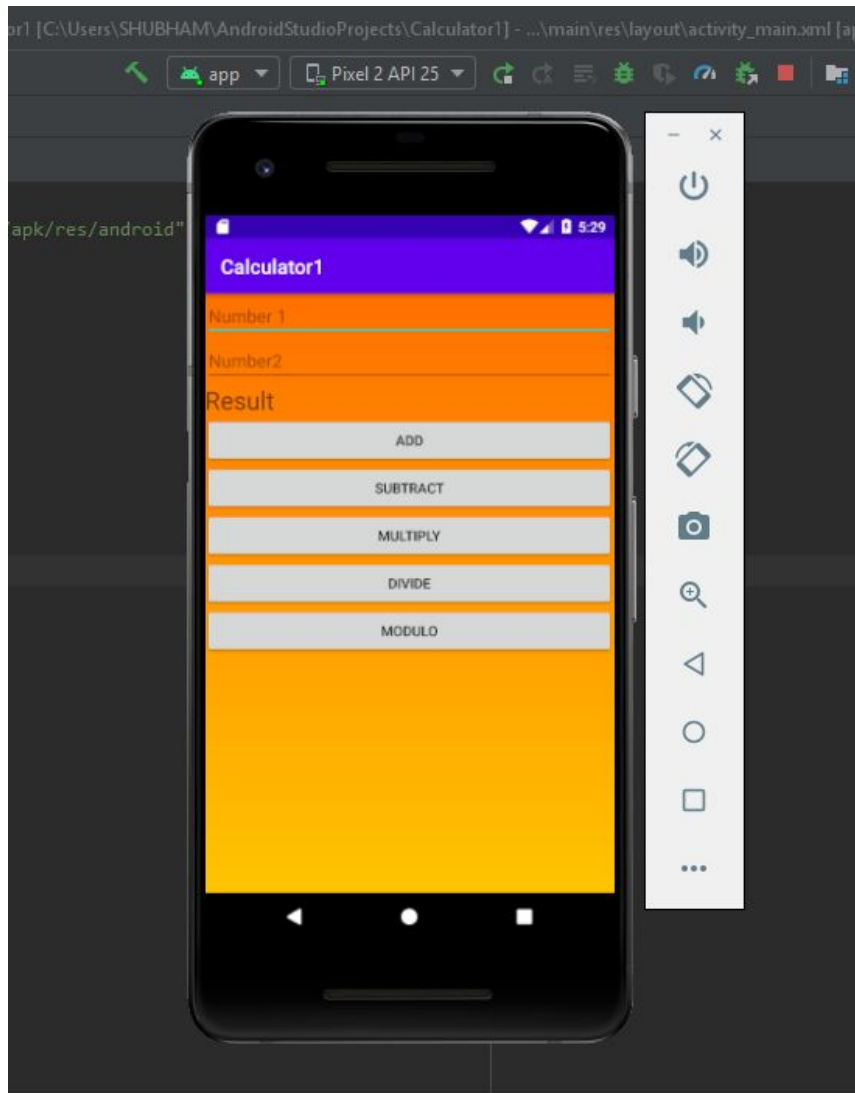
Shubham Pandey - | April 15, 2020



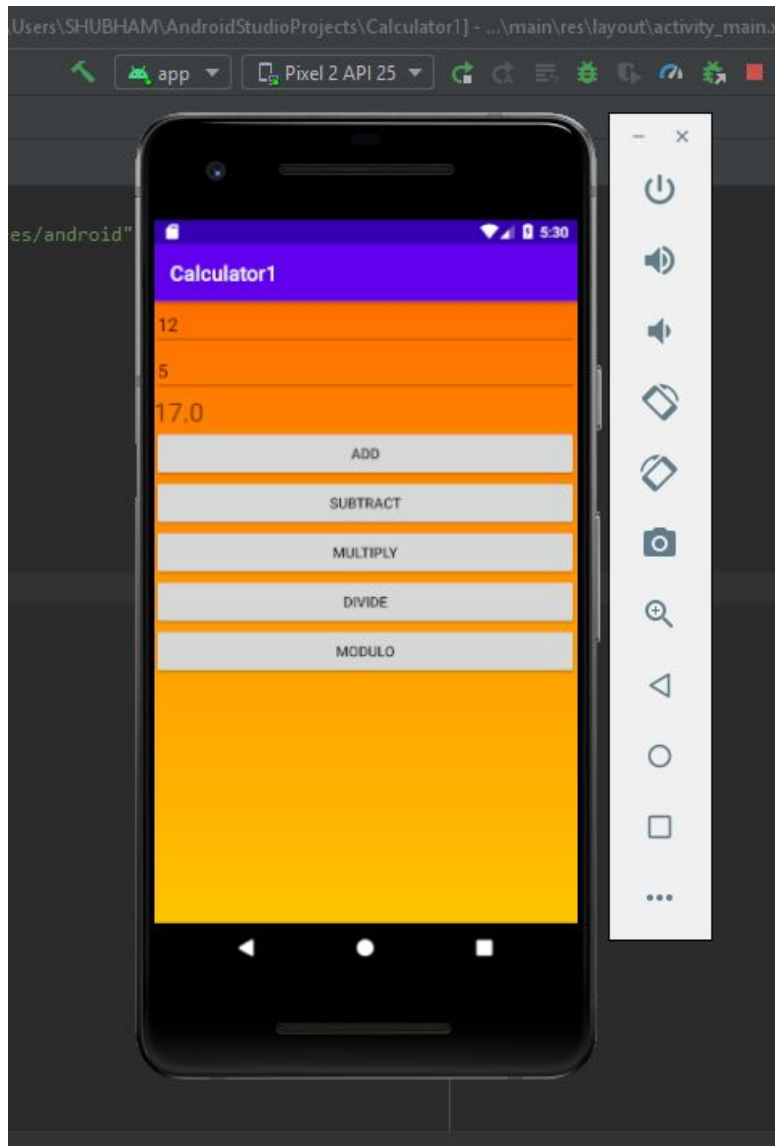
Screenshots

Below are the screenshots of the app which was run on Android Studio. Here we have used Pixel2 emulator.

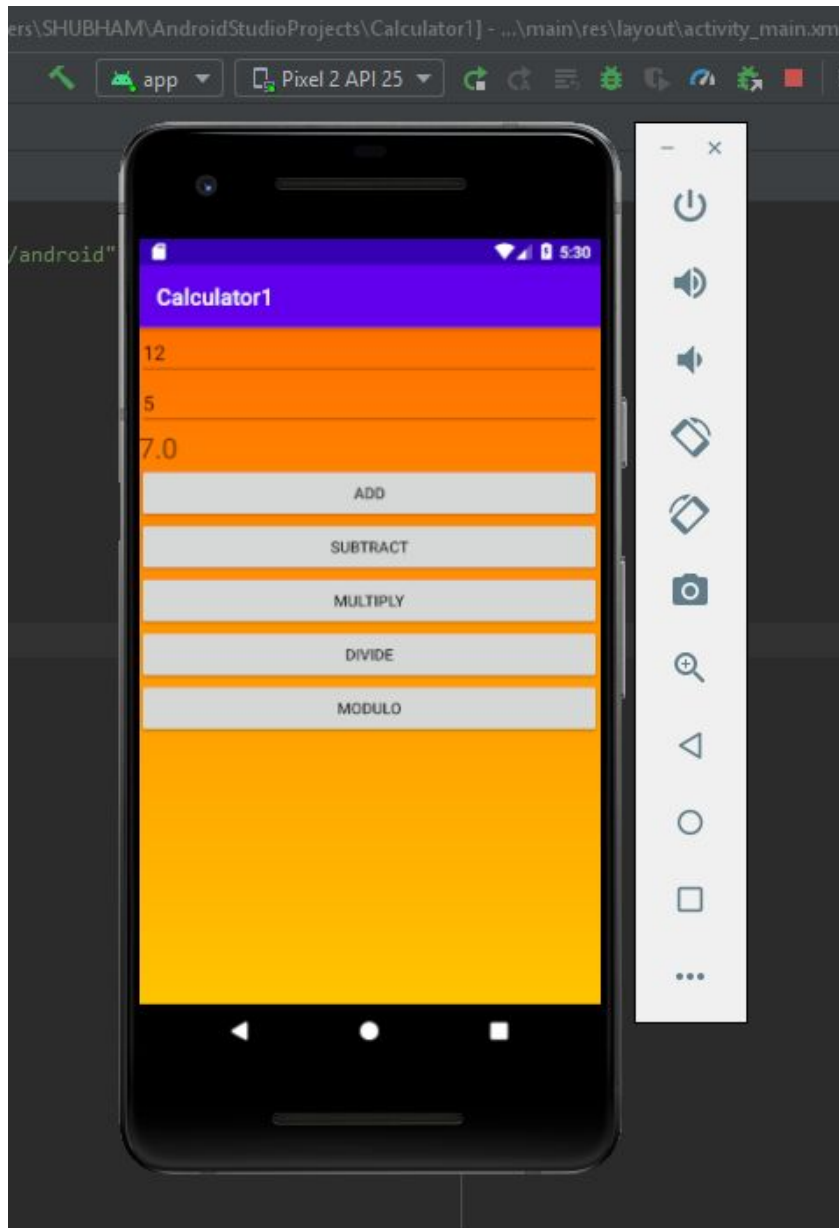
Main Screen of the APP



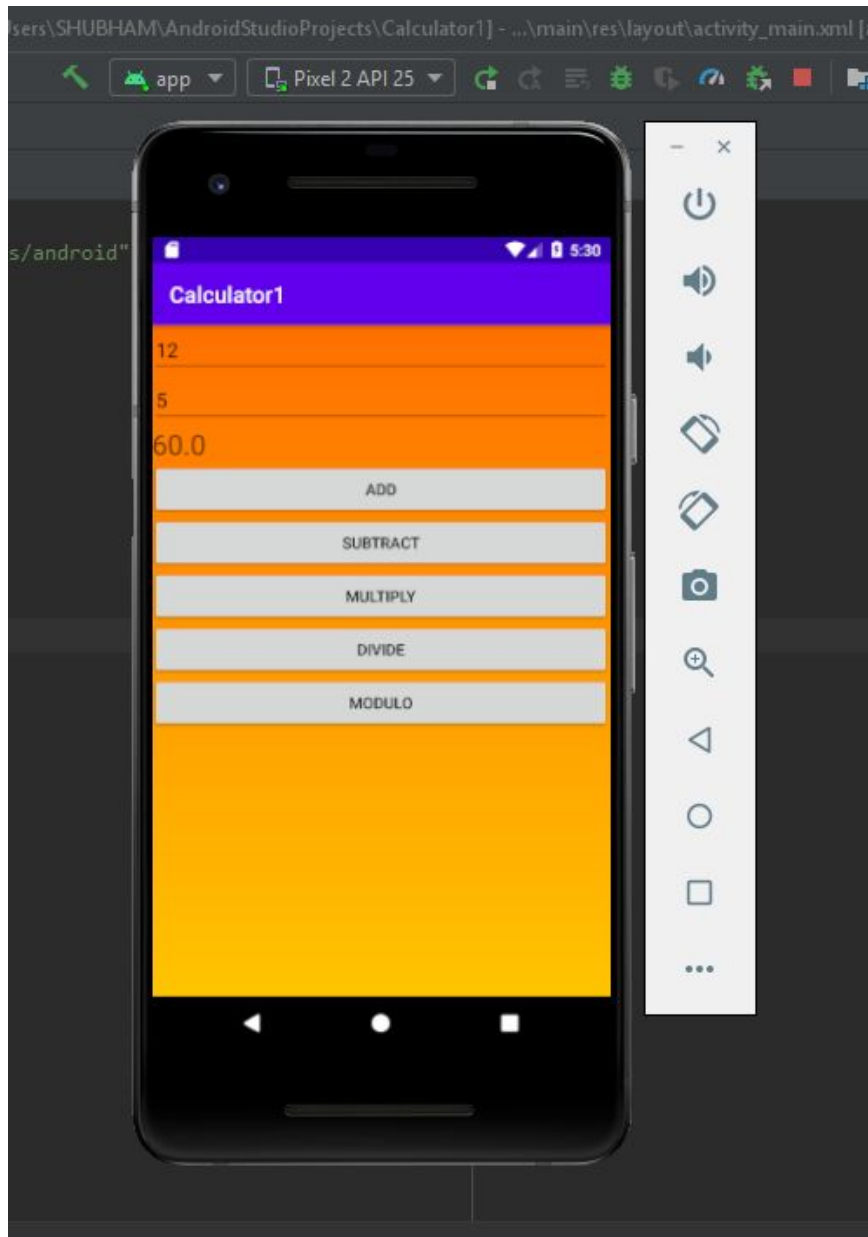
ADDING two numbers



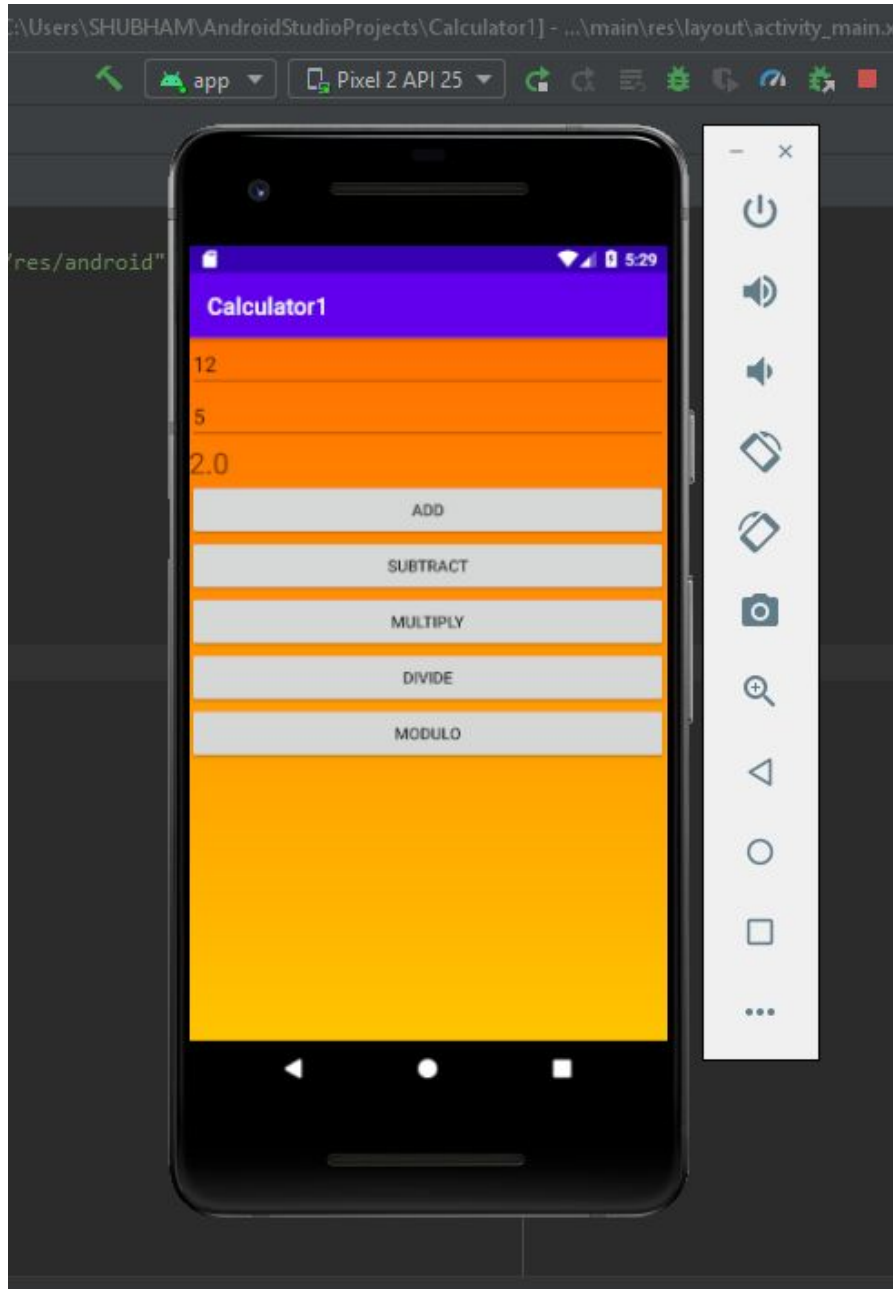
SUBTRACTING two numbers



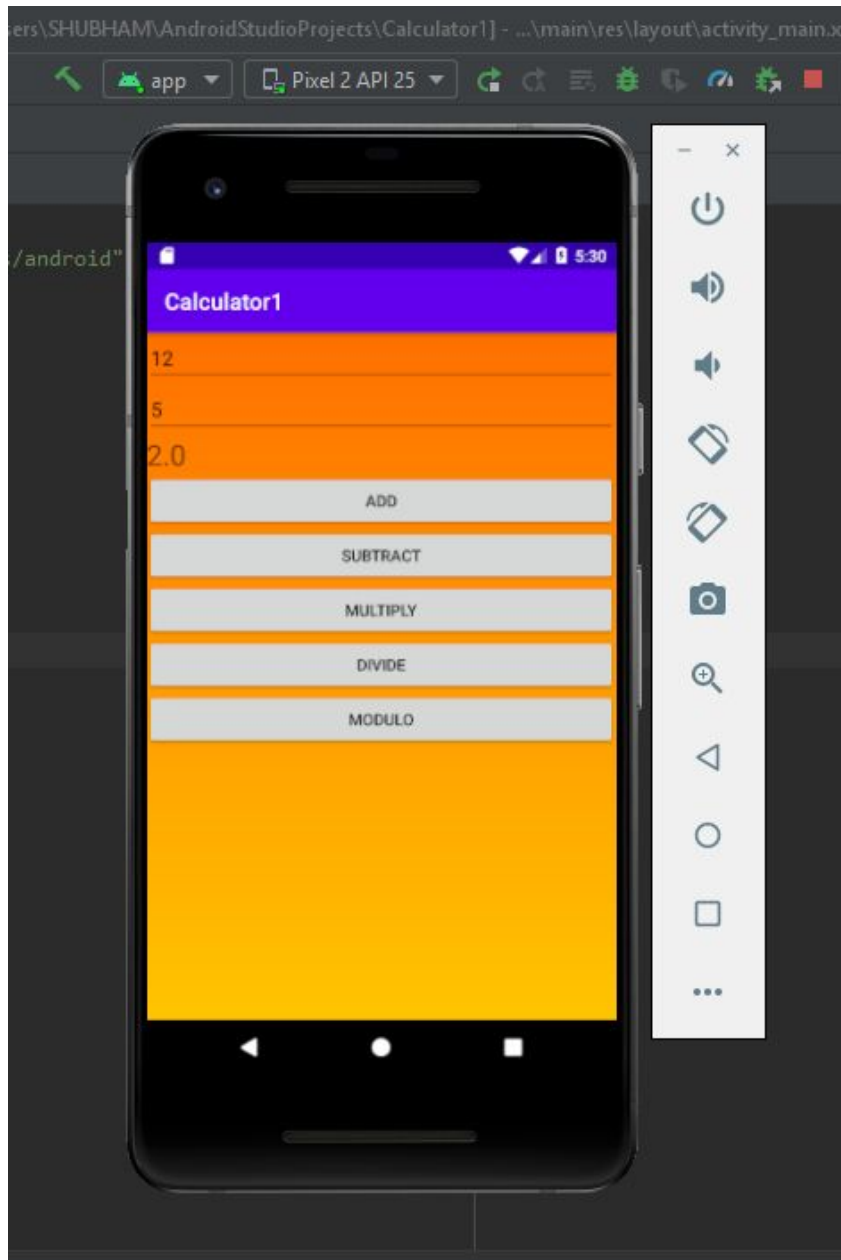
MULTIPLYING two numbers



DIVIDING two numbers



MODULO of two numbers



Conclusion

This report has discussed the development of Fundamental mathematical calculations. The objectives of this project were to develop the necessary software to have the calculators, being larger in size and needing extra power.

All objectives were tried to be met. By keeping track of the calculations the Android Application has an interface which can be easily accessible even by a lay-man. Also, a Calculator app as Android Project has good scope in today's technological world. This project introduced us to the important topics of Android Studio.

In the project, the calculator creates a calculator app and for making this app and for it to be fully functional we will be using Android Studio for developing the app from scratch. Besides this Android Studio is Google's officially supported IDE for developing Android apps. For example, in an Application driven world, the phase of calculations are continually measured by using Android Studio in Mobile App Development.

It's because Android doesn't release 1 phone from 1 company with 1 new OS every year, but countless phones from numerous companies, adding their own twist, throughout the year, developing gradually day-by-day. I am not one to say that Android is better or worse than one OS, but is unique and incomparable to other mobile operating systems.

References

Programmer Web Resources:

1. Introduction to Android: <http://developer.android.com/guide/index.html>.
2. Android API: <http://developer.android.com/reference/packages.html>
3. Java 6 API: <http://docs.oracle.com/javase/6/docs/api/>
4. Android Fundamentals:
<http://developer.android.com/guide/components/fundamentals.html>
5. The Java Tutorials: <http://docs.oracle.com/javase/tutorial/>
6. Android User Interfaces:
<http://developer.android.com/guide/topics/ui/index.html>
7. Layout: <http://developer.android.com/guide/topics/ui/declaring-layout.html>
8. Common Tasks:
<http://developer.android.com/guide/appendix/faq/commontasks.html>
9. Android Training: <http://developer.android.com/training/index.html>.
10. Android Developer's Blog: <http://android-developers.blogspot.com/>
11. Developer FAQ: <http://developer.android.com/resources/faq/>
12. Developer Forums:
<http://developer.android.com/resources/community-groups.html>
13. Android Developer's Group:
<http://groups.google.com/group/android-developers?lnk=>

General Android Information:

1. Android Power: <http://blogs.computerworld.com/raphael>
2. Phandroid: <http://phandroid.com/>
3. Android Guys: <http://www.androidguys.com/>