

# Assignment No. B3

## Aim

Mobile Application for Calculator

## Problem Definition

A mobile application needs to be designed for using a Calculator (+, -, \*, /, Sin, Cos, sq-root) with Memory Save/Recall using Extended precision floating point number format. Give the Required modeling, Design and Positive-Negative test cases.

## Learning Objectives

- Learn how to develop Mobile Applications.
- Implement Mobile Application for Calculator and learn how to test the Mobile Application.
- Learn about USE-CASE modeling in Modelio Software.

## Learning Outcome

- Learnt about developing Mobile Applications.
- Implemented Mobile Application for Calculator.
- Learnt USE-CASE modeling in Modelio Software.

## Software And Hardware Requirements

- Latest 64-BIT Version of Linux Operating System
- Android Studio
- Modelio Software

## Mathematical Model

$S = \{s, e, I, o, f, DD, NDD, success, failure\}$

$s = \{ \text{start android sdk} \}$

$e = \text{end of system}$

$I = \{R+, op\}$

$op = \{+, -, *, /\}$

$O = \{I1 \text{ op } I2\}$

$DD = \{ \text{input numbers} \}$

$NDD = \{ \text{Textual Data} \}$

$f = \{add(), sub(), mul(), div(), sin(), cos(), tan(), clearall(), view \text{ calculations}()\}$

$add() = \{I1 + I2\}$

$sub() = \{I1 - I2\}$

$mul() = \{I1 * I2\}$

$div() = \{I1 / I2\}$

$mod() = \{I1 \% I2\}$

$sin() = \{ \sin(I1^0) \}$

$cos() = \{ \cos(I1^0) \}$

$tan() = \{ \tan(I1^0) \}$

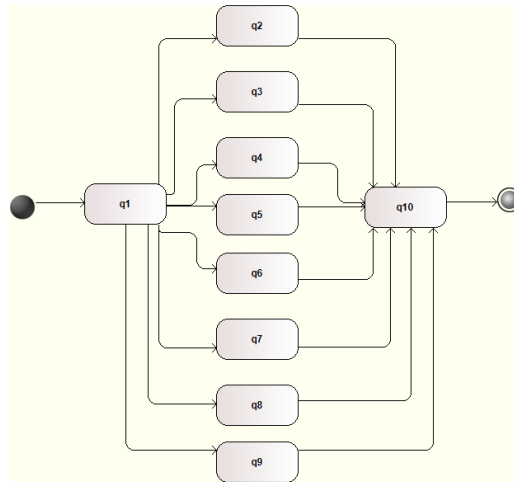
$clearall() = \{ \text{Clear all the calculations} \}$

$view \text{ calculations}() = \{ \text{View calculations till date} \}$

$success = \text{Correct output is displayed.}$

$failure = \text{Incorrect output is displayed.}$

## State Diagram



where,

q1 = function f1  
q2 = function f2  
q3 = function f3  
q4 = function f4  
q5 = function f5  
q6 = function f6  
q7 = function f7  
q8 = function f8  
q9 = function f9  
q9 = function f10  
q10 = function f10

## Theory

### Mobile Applications

A mobile app is a computer program designed to run on mobile devices such as smartphones and tablet computers. Most such devices are sold with several apps bundled as pre-installed software, such as a web browser, email

client, calendar, mapping program, and an app for buying music or other media or more apps. Some pre-installed apps can be removed by an ordinary uninstall process, thus leaving more storage space for desired ones. Where the software does not allow this, some devices can be rooted to eliminate the undesired apps. Mobile native apps stand in contrast to software applications that run on desktop computers, and to web applications which run in mobile web browsers rather than directly on the mobile device.

Android is an open source and Linux-based Operating System for mobile devices such as smartphones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies. Android offers a unified approach to application development for mobile devices which means developers need only develop for Android, and their applications should be able to run on different devices powered by Android. The first beta version of the Android Software Development Kit (SDK) was released by Google in 2007 where as the first commercial version, Android 1.0, was released in September 2008.

## **Android**

Android is a mobile operating system (OS) currently developed by Google, based on the Linux kernel and designed primarily for touchscreen mobile devices such as smartphones and tablets. Android's user interface is mainly based on direct manipulation, using touch gestures that loosely correspond to real-world actions, such as swiping, tapping and pinching, to manipulate on-screen objects, along with a virtual keyboard for text input.

Applications ("apps"), which extend the functionality of devices, are written using the Android software development kit (SDK) and, often, the Java programming language that has complete access to the Android APIs. The SDK includes a comprehensive set of development tools, including a debugger, software libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Initially, Google's supported integrated development environment (IDE) was Eclipse using the Android Development Tools (ADT) plugin; in December 2014, Google released Android Studio, based on IntelliJ IDEA, as its primary IDE for Android application development.

## **Android Studio**

Android Studio is the official integrated development environment (IDE) for Android platform development. It was announced on May 16, 2013 at the Google I/O conference. Android Studio is freely available under the Apache License 2.0. Based on JetBrains' IntelliJ IDEA software, Android Studio is designed specifically for Android development. It is available for download on Windows, Mac OS X and Linux, and replaced Eclipse Android Development Tools (ADT) as Google's primary IDE for native Android application development.

## Program

Main Activity.java

```
package com.example.dbsl.calculator;

import android.content.SharedPreferences;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import java.util.HashMap;
import java.util.Map;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {

    String mem_store = "";
    String expression = "";
    String op = "0";
    String num1 = "";
    String num2 = "";
    String prefs = "history";
    SharedPreferences preferences ;
    boolean dec = false;
    boolean flag = false;
    TextView display,memstore;
    Button mem,zero,one,two,three,four,five,six,seven,eight,nine,decimal_point,
    equals,sine,cos,percent,clear,add,subtract,divide,multiply,reset,open,close;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        preferences = getSharedPreferences(prefs,MODE_PRIVATE);
        display = (TextView)findViewById(R.id.display);
        memstore = (TextView)findViewById(R.id.memstore);
        try {
            mem = (Button) findViewById(R.id.memory);
            mem.setOnClickListener(this);
```

```

open = (Button) findViewById(R.id.open);
open.setOnClickListener(this);
close = (Button) findViewById(R.id.close);
close.setOnClickListener(this);
reset = (Button) findViewById(R.id.reset);
reset.setOnClickListener(this);
zero = (Button) findViewById(R.id.zero);
zero.setOnClickListener(this);
one = (Button) findViewById(R.id.one);
one.setOnClickListener(this);
two = (Button) findViewById(R.id.two);
two.setOnClickListener(this);
three = (Button) findViewById(R.id.three);
three.setOnClickListener(this);
four = (Button) findViewById(R.id.four);
four.setOnClickListener(this);

five = (Button) findViewById(R.id.five);
five.setOnClickListener(this);
six = (Button) findViewById(R.id.six);
six.setOnClickListener(this);
;
seven = (Button) findViewById(R.id.seven);
seven.setOnClickListener(this);
eight = (Button) findViewById(R.id.eight);
eight.setOnClickListener(this);
nine = (Button) findViewById(R.id.nine);
nine.setOnClickListener(this);

decimal_point = (Button) findViewById(R.id.decimal);
decimal_point.setOnClickListener(this);
equals = (Button) findViewById(R.id.equals);
equals.setOnClickListener(this);
sine = (Button) findViewById(R.id.sine);
sine.setOnClickListener(this);
cos = (Button) findViewById(R.id.cosine);
cos.setOnClickListener(this);

percent = (Button) findViewById(R.id.percent);
percent.setOnClickListener(this);
clear = (Button) findViewById(R.id.clear);
clear.setOnClickListener(this);

add = (Button) findViewById(R.id.add);
add.setOnClickListener(this);
subtract = (Button) findViewById(R.id.subtract);

```

```

        subtract.setOnClickListener(this);
        divide = (Button) findViewById(R.id.divide);
        divide.setOnClickListener(this);
        multiply = (Button) findViewById(R.id.multiply);
        multiply.setOnClickListener(this);
    }
    catch (NullPointerException e)
    {
        Log.d("ABC","Null pointer exception encountered");
    }
}

@Override
public void onClick(View v) {
    switch (v.getId())
    {
        case R.id.zero:
            expression += "0";
            mem_store += "0";
            display.setText(expression);
            // //calculate("0",flag,"a");
            break;
        case R.id.one:
            expression += "1";
            mem_store += "1";
            display.setText(expression);
            ///calculate("1",flag,"a");
            break;
        case R.id.two:
            mem_store += "2";
            expression += "2";
            display.setText(expression);
            ///calculate("2",flag,"a");
            break;
        case R.id.three:
            mem_store += "3";
            expression += "3";
            display.setText(expression);
            // //calculate("3",flag,"a");
            break;
        case R.id.four:
            mem_store += "4";
            expression += "4";
            display.setText(expression);
            // //calculate("4",flag,"a");
    }
}

```

```

        break;
    case R.id.five:
        mem_store += "5";
        expression += "5";
        display.setText(expression);
        // //calculate("5",flag,"a");
        break;
    case R.id.six:
        mem_store += "6";
        expression += "6";
        display.setText(expression);
        // //calculate("6",flag,"a");
        break;
    case R.id.seven:
        mem_store += "7";
        expression += "7";
        display.setText(expression);
        // //calculate("7",flag,"a");
        break;
    case R.id.eight:
        mem_store += "8";
        expression += "8";
        display.setText(expression);
        // //calculate("8",flag,"a");
        break;
    case R.id.nine:
        mem_store += "9";
        expression += "9";
        display.setText(expression);
        // //calculate("9",flag,"a");
        break;
    case R.id.decimal:
        if(!dec)
        {
            mem_store += ".";
            expression += ".";
            display.setText(expression);
            dec = true;
            // //calculate(".",flag,"a");
        }
        else {
            Toast.makeText(getApplicationContext(),
                "Can't add another decimal.",Toast.LENGTH_SHORT).show();
        }
        break;
    case R.id.open:

```



```

        mem_store += "(";
        expression += " ( ";
        display.setText(expression);
        break;
case R.id.close:
    mem_store += ")";
    expression += " ) ";
    display.setText(expression);
case R.id.equals:
    calculate();
    break;
case R.id.sine:
    mem_store += "sin ";
    display.setText("sin ");
    expression = "";
    num1 = "";
    num2 = "";
    op = "1";
    dec= false;
    flag = false;
    expression = "sin ";
    //calculate("SIN",flag,"TRIG");
    break;
case R.id.cosine:
    mem_store += "cos ";
    display.setText("cos ");
    expression = "";
    num1 = "";
    num2 = "";
    dec= false;
    flag = false;
    op = "2";
    expression = "cos ";
    //calculate("COS",flag,"TRIG");
    break;
case R.id.percent:
    break;
case R.id.clear:
    break;
case R.id.add:
    mem_store += "+";
    expression += " + ";
    display.setText(expression);
    dec = false;
    //      flag = true;
    //      op = "+";

```

```

//          //calculate("ADD",flag,op);
//          break;
case R.id.subtract:
    mem_store += "-";
    dec = false;
    expression += " - ";
    display.setText(expression);
//          flag = true;
//          op = "-";
//          //calculate("SUB",flag,op);
//          break;
case R.id.divide:
    mem_store += "/";
    dec = false;
    expression += " / ";
    display.setText(expression);
//          flag = true;
//          op = "/";
//          //calculate("DIV",flag,op);
//          break;
case R.id.multiply:
    mem_store += "*";
    dec = false;
    expression += " * ";
    display.setText(expression);
//          flag = true;
//          op = "*";
//          //calculate("MUL",flag,op);
//          break;
case R.id.memory:
    Map<String, ?> stored = preferences.getAll();
    String temp = "";
    for(int i=1;i<=stored.size();i++)
    {
        temp += preferences.getString(""+i,"nothing to show");
        temp += "\n";
    }
    memstore.setText(""+temp);
    break;
case R.id.reset:
    if(op.equals("0")) {
        mem_store += " = ";
        mem_store += expression;
    }
    Log.d("ABC","Full expression is "+mem_store);
    if(!mem_store.equals(" = "))

```

```

        {
            SharedPreferences.Editor editor = preferences.edit();
            editor.putString(""+(preferences.getAll().size()+1),mem_store);
            editor.commit();
        }
        op = "0";
        mem_store = "";
        expression = " ";
        dec = false;
        display.setText(expression);

        break;
    }
}
public void calculate()
{
    if(op.equals("1"))
    {
        mem_store += " = ";
        mem_store += Math.sin(Double.parseDouble
(expression.substring(4))*(Math.PI/180));
        display.setText(""+Math.sin(Double.parseDouble
(expression.substring(4))*(Math.PI/180)));
    }else if(op.equals("2")) {
        mem_store += " = ";
        mem_store += Math.cos(Double.parseDouble
(expression.substring(4)) * (Math.PI / 180));
        display.setText(""+Math.cos(Double.parseDouble
(expression.substring(4))*(Math.PI/180)));
    }
    else
    {
        expression = "" + EvaluateString.evaluate(expression);
        display.setText(expression);
    }
}
}
}

```

activity<sub>main</sub>.xml

```

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

```

```

xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.example.dbsl.calculator.MainActivity">

<TextView
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:id="@+id/display"
    android:layout_alignParentTop="true"
    android:layout_alignParentStart="true" />

<GridLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/display"
    android:id="@+id/grid"
    android:rowCount="7"
    android:columnCount="4">
    <Button
        android:id="@+id/zero"
        android:text="0"/>
    <Button
        android:id="@+id/one"
        android:text="1"/>
    <Button
        android:id="@+id/two"
        android:text="2"/>
    <Button
        android:id="@+id/three"
        android:text="3"/>
    <Button
        android:id="@+id/four"
        android:text="4"/>
    <Button
        android:id="@+id/five"
        android:text="5"/>
    <Button
        android:id="@+id/six"
        android:text="6"/>
    <Button
        android:id="@+id/seven"
        android:text="7"/>
    <Button
        android:id="@+id/eight"
        android:text="8"/>

```

```

<Button
    android:id="@+id/nine"
    android:text="9"/>
<Button
    android:id="@+id/decimal"
    android:text="."/>
<Button
    android:id="@+id/equals"
    android:text="="/>
<Button
    android:id="@+id/sine"
    android:text="sin"/>
<Button
    android:id="@+id/cosine"
    android:text="cos"/>
<Button
    android:id="@+id/percent"
    android:text=""/>
<Button
    android:id="@+id/clear"
    android:text="C"/>
<Button
    android:id="@+id/add"
    android:text="+"/>
<Button
    android:id="@+id/subtract"
    android:text="-"/>
<Button
    android:id="@+id/divide"
    android:text="/"/>
<Button
    android:id="@+id/multiply"
    android:text="*/>
<Button
    android:id="@+id/open"
    android:text="("/>
<Button
    android:id="@+id/close"
    android:text=")"/>
<Button
    android:id="@+id/reset"
    android:text="AC"/>
<Button
    android:id="@+id/memory"
    android:text="MEM"/>
</GridLayout>

```

```

<ScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@id/grid">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/memstore"
        android:layout_below="@+id/grid"
        android:layout_alignParentStart="true" />
</ScrollView>

</RelativeLayout>

```

EvaluateString.java

```

package com.example.dbsl.calculator;

import java.util.Stack;

public class EvaluateString
{
    public static double evaluate(String expression)
    {
        char[] tokens = expression.toCharArray();

        // Stack for numbers: 'values'
        Stack<Double> values = new Stack<Double>();

        // Stack for Operators: 'ops'
        Stack<Character> ops = new Stack<Character>();

        for (int i = 0; i < tokens.length; i++)
        {
            // Current token is a whitespace, skip it
            if (tokens[i] == ' ')
                continue;

            // Current token is a number, push it to stack for numbers
            if ((tokens[i] >= '0' && tokens[i] <= '9') || (tokens[i] == '.'))
            {
                StringBuffer sbuf = new StringBuffer();
                // There may be more than one digits in number
            }
        }
    }
}

```

```

        while (i < tokens.length &&
            (((tokens[i] >= '0' && tokens[i] <= '9')) || (tokens[i] == '.')))
            sbuf.append(tokens[i++]);
        values.push(Double.parseDouble(sbuf.toString()));
    }

    // Current token is an opening brace, push it to 'ops'
    else if (tokens[i] == '(')
        ops.push(tokens[i]);

    // Closing brace encountered, solve entire brace
    else if (tokens[i] == ')')
    {
        while (ops.peek() != '(')
            values.push(applyOp(ops.pop(), values.pop(), values.pop()));
        ops.pop();
    }

    // Current token is an operator.
    else if (tokens[i] == '+' || tokens[i] == '-' ||
        tokens[i] == '*' || tokens[i] == '/')
    {
        // While top of 'ops' has same or greater precedence to current
        // token, which is an operator. Apply operator on top of 'ops'
        // to top two elements in values stack
        while (!ops.empty() && hasPrecedence(tokens[i], ops.peek()))
            values.push(applyOp(ops.pop(), values.pop(), values.pop()));

        // Push current token to 'ops'.
        ops.push(tokens[i]);
    }
}

// Entire expression has been parsed at this point, apply remaining
// ops to remaining values
while (!ops.empty())
    values.push(applyOp(ops.pop(), values.pop(), values.pop()));

// Top of 'values' contains result, return it
return values.pop();
}

// Returns true if 'op2' has higher or same precedence as 'op1',
// otherwise returns false.
public static boolean hasPrecedence(char op1, char op2)
{

```

```

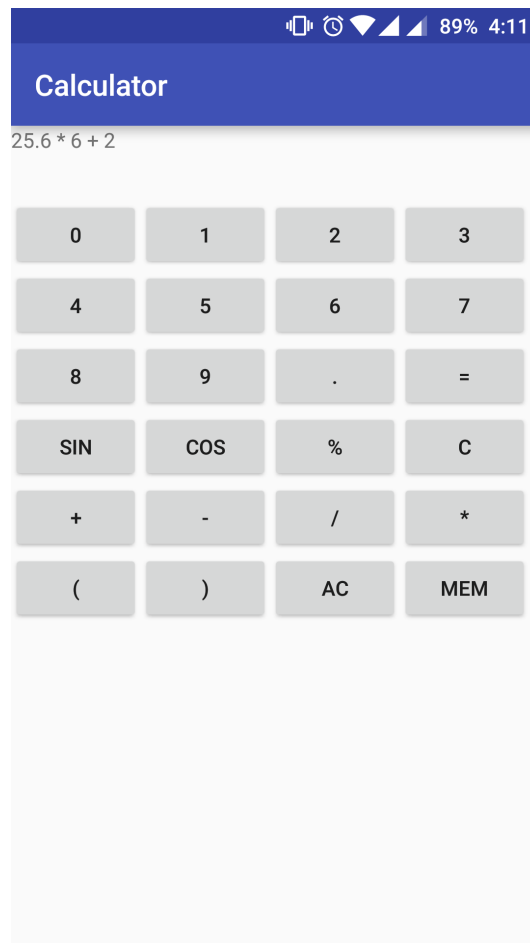
        if (op2 == '(' || op2 == ')')
            return false;
        if ((op1 == '*' || op1 == '/') && (op2 == '+' || op2 == '-'))
            return false;
        else
            return true;
    }

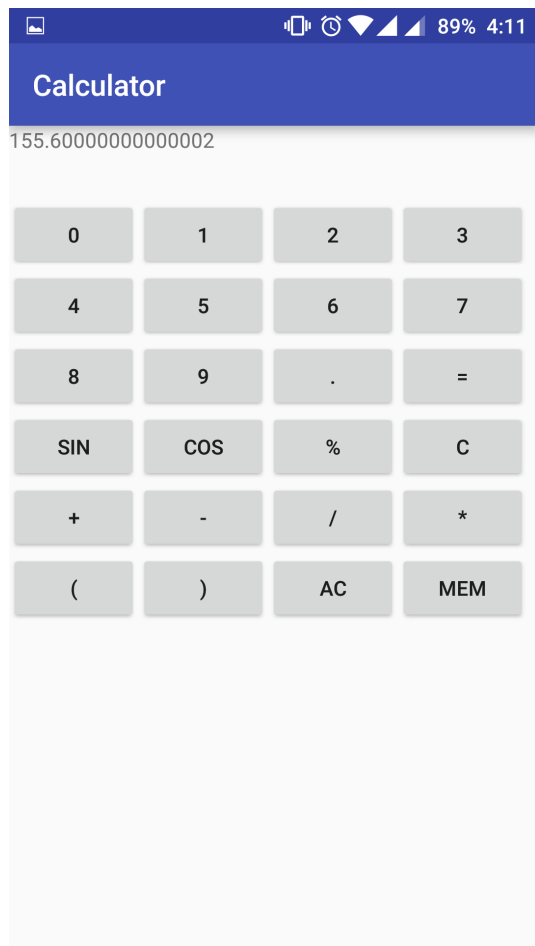
    // A utility method to apply an operator 'op' on operands 'a'
    // and 'b'. Return the result.
    public static double applyOp(char op, double b, double a)
    {
        switch (op)
        {
            case '+':
                return a + b;
            case '-':
                return a - b;
            case '*':
                return a * b;
            case '/':
                if (b == 0)
                    throw new
                        UnsupportedOperationException
                        ("Cannot divide by zero");
                return a / b;
        }
        return 0;
    }
}

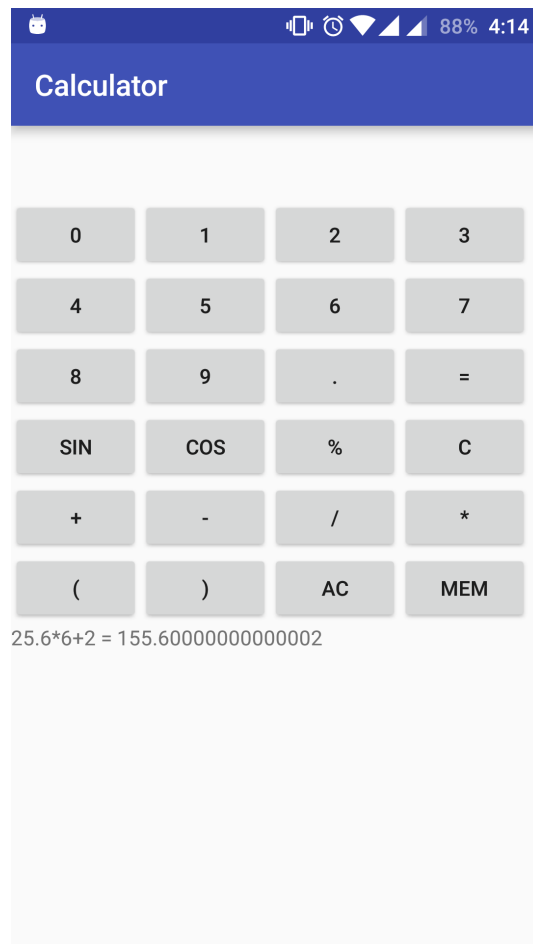
```



## Output







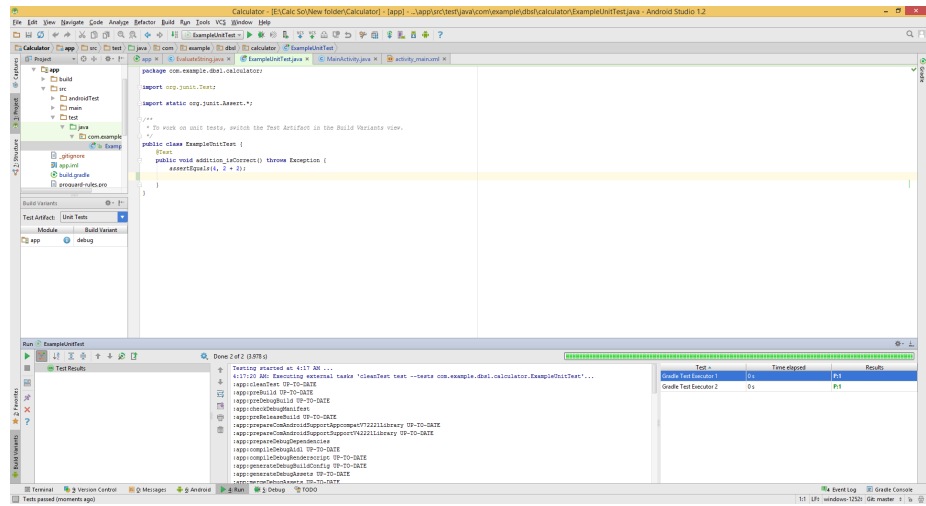
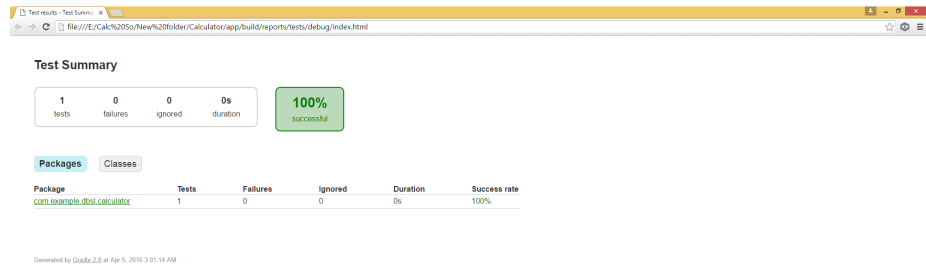
## Testing

```
import org.junit.Test;

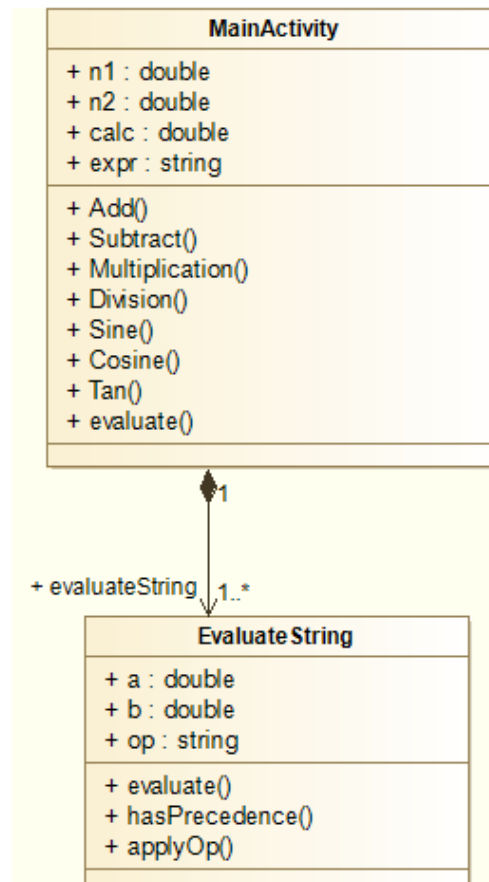
import static org.junit.Assert.*;

/**
 * To work on unit tests, switch the Test Artifact in the Build Variants view.
 */
public class ExampleUnitTest {
    @Test
    public void addition_isCorrect() throws Exception {
        assertEquals(4, 2 + 2);
    }
}
```

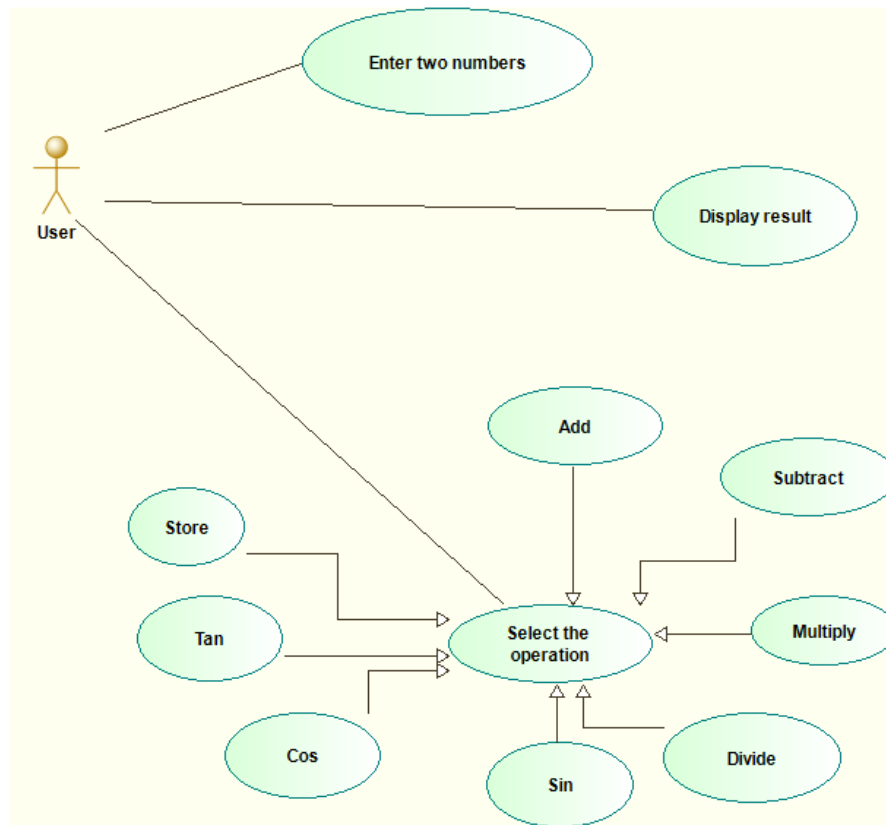
}



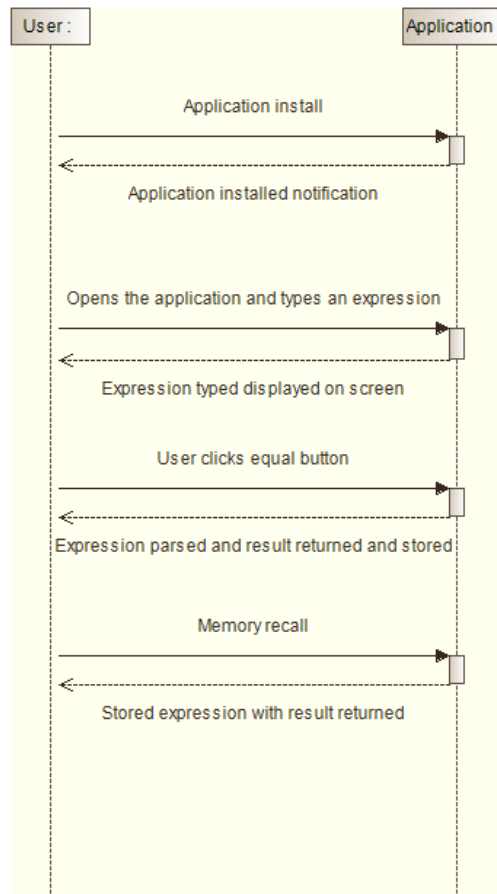
## Class Diagram



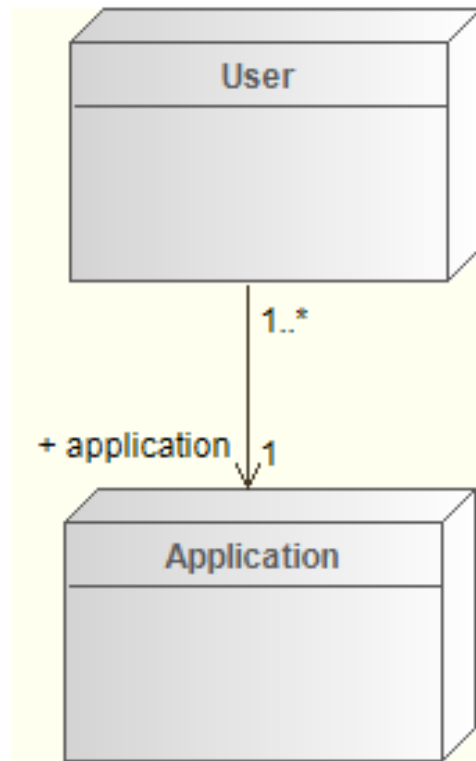
## Use-Case Diagram



## Sequence Diagram



## Deployment Diagram





## Positive Testing

Sr No.	Test Condition	Steps to be executed	Expected Result	Actual Result
1.	Enter value of which trigonometric function is to be calculate	Press Enter	Result of trigonometric function	Same as Expected Result
2.	Enter the button “=”	Click on the equal to Button	Display the final Result	Same as Expected Result
3.	1/0	Click on “=” button	Display Result “infinity”	Same as Expected Result

## Negative Testing

Sr No.	Test Condition	Steps to be executed	Expected Result	Actual Result
1.	Enter Imaginary value	Press Enter	Error Messages	Same as Expected Result
2.	Without “(, ”)	Press Enter	Error	Same as Expected Result
3.	Press “=” without any input	Press “=”	undefined	Same as Expected Result

## Conclusion

Thus we have successfully implemented a mobile application for Calculator on Android platform