Table of Contents

Chapter 1

Introduction

1.1 Background and Objectives

As per the COMP 207 course requirement of the second year, first semester of Computer Science in Kathmandu University, we created an android application for calculating purpose. The calculator features various kinds of normal calculator as addition, subtraction, multiplication and division. Besides these normal features the calculator features scientific functions such as ln, log, square root, modulus, sin, cos, tan, sin-1, cos-1, tan-1, solution of a quadratic equation, an static notepad for the user to write the notes etc.

The objective of the application is to provide the easy calculations for the users and the students in the field of engineering and science so that they can use the application as the calculator for their various purposes like research, projects etc.

We developed the application in order to meet the following objectives:
- To learn about the Android development.
- To learn the different ways methods used for computation of data in calculator.
- Create a user friendly user interface (UI).
- Include mostly used features of the scientific calculator.

1.2 System Overview

The android application is developed by using the Software Development Kit, AVD manager for the emulation, XML for the User Interface(UI) design, and core JAVA for the working of the application.
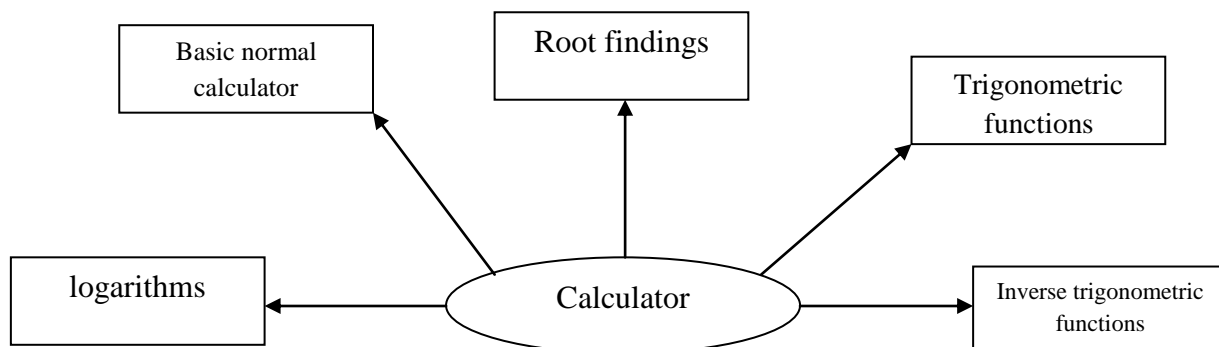
Fig: Calculators Features

1.3 Methodology

The design issues of every project work are a complex part every programmer should face. From our own research and problems, we started with an idea do something simple yet important and useful for all of us. Together we came to a common conclusion to develop calculator for Android. Concerning about how we marched forward in this project has been mentioned below.

Android operating system (OS) is getting a lot of popularity in the world current market as most of the smart phones are powered by the android OS. It is because its lightweight framework that can easily be supported by the mobile phones, tablets and other smart devices. Even the computers can be run using the android OS. Since the android application has a lot of coverage in market, learning to make applications in android can be advantageous. Thus we concluded that we need to develop android applications. So we plan to develop and a simple calculator in android.

1.4 Overview of the report

This report, being the final report, is description of the target work achieved. Till now, the basic features and the objective of the calculator have been described briefly. The report is concerned with the explanation of the android application project completed and its features, and its experimentation. Up to this point, the background and the overview of the system have been discussed briefly along with the methodology.

In background, we discussed the importance of the project. We have planned to keep the application in the Google Play Store. In the upcoming parts, the program will be discussed in detail about what has been done and the features and the facilities of the application. Also, the different parts of the program along with the experiments that have been done will be covered.

Chapter 2

System analysis and design

2.1 Feasibility study

After discussion with the team members and the looking at the other developed applications we came up with a user friendly design. We made design in XML and improved it gradually according to the need. The feasibility can be discussed into two parts.

2.1.1 Technical Feasibility

The application is technically feasible in mathematics, and for scientific uses. It can also be used in research and other computational related problems. The calculator has user friendly interface. The application is useful for students, researchers, mathematicians, businessman's etc.

2.1.2 Social Feasibility

The application is a calculation purpose so it doesn't concerns much about the social feasibility as it can be used by all kinds of users. The application doesn't contains anything that any religion or the members of the society and other people.

2.2 Software requirement specification

2.2.1 Software requirement

- Any device with android OS (API 10 or higher).

2.2.2 Hardware requirement

- Physical Memory (RAM) 256MB minimum.
- Processor
- Screen resolution size(equivalent to NEXUS 1 or higher)

2.3 Problems and Solutions

2.3.1 Problems

The following problems were encountered during our project activities.

- At the beginning we were unfamiliar with the android application development.
- Difficulty in choosing the proper API for the development of the application.
- Problem in downloading the API, SDK (……….) manager, AVD (………) manager as the internet connection was not reliable.
- Embedding the codes and the layouts together.
- There was lack of time due to the earthquakes and the aftershocks continued in 2072/01/12 as we feared to enter our home.
- Problem for electricity as the some of the electric poles were damaged by earthquake.

2.3.2 Solutions:

The solutions are mentioned below:

- We took help from video tutorials from THE NEW BOSTON for android development.
- We took other video tutorials from www.youtube.com.
- We searched in different websites to implement the features we wanted.
- Took reference form the other projects.
- With time embedding the codes became easier.

2.4 Scopes and opportunities

As we all know that in IT industry android OS has taken a wide coverage compared to any other operating system. It is because android can easily be installed in the smart phones and tablets. The features provided by the android OS are reliable. The features like camera, user interface, GPS, voice reorganization etc. Since it is an open source anyone can develop application for android and anyone can use it.

This creates the wide scope of android as we have a lot of users. If we can develop an application and collect lots of users then it is a good opportunity to enhance the application into the android market.

Chapter 3
Discussion

3.1 Java

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture. Java is one of the most popular programming languages in use, particularly for client-server web applications. Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

The java programming language is used by the android to run the core functions. The user interface is developed by using XML and the objects like buttons, textbox etc are linked to the java and operated by using the core java. The applications provide different functions that can be imported. Those functions are also made in java. So java is important for the android development.

3.2 XML

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format which is both human-readable and machine-readable. It is defined by the W3C's XML 1.0 Specification and by several other related specifications, all of which are free open standards.

The design goals of XML emphasize simplicity, generality and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, it is widely used for the representation of arbitrary data structures such as those used in web services.
Several schema systems exist to aid in the definition of XML-based languages, while many application programming interfaces (APIs) have been developed to aid the processing of XML data.

The XML in android application is used for the development of the user interface. The layout of the different view pages are made by using the XML file and then accessed by the core java. The XML is similar to the CSS but there are few differences in the syntax and coding.

3.3 Resources Used

The resources used for the development of this application are mentioned below:

- Android Studio 1.1
- SDK (Software Development Kit)
- AVD manager.

3.4 Program working mechanism

The application has certain algorithm for the implementation of different functions that are mentioned below:

3.4.1 Splash Screen

```
public class Splash extends Activity {
  MediaPlayer ourSong;
  @Override
  protected void onCreate(Bundle SujalKokh) {
    super.onCreate(SujalKokh);
    setContentView(R.layout.splash);
    ourSong = MediaPlayer.create(Splash.this, R.raw.legohouse);
    ourSong.start();
    Thread timer =  new Thread(){
      public void run(){
        try{
          sleep(4000);
        }

        catch(InterruptedException e){
          e.printStackTrace();
        }

        finally{
          IntentopenStartingPoint=new
Intent("com.mythcrew.torresapp.STARTINGPOINT");
          startActivity(openStartingPoint );
        }
      }
    };
    timer.start();
  }

  @Override
```

```java
   protected  void onPause(){
      super.onPause();
      ourSong.release();
      finish();
   }
}
```

3.4.2 Main Content
```java
public class starting_point extends ActionBarActivity {

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


   ArrayList<String> arrayList = new ArrayList<String>();
   String string = "";
   String string1 = "";

   public void onClick1(View v) {
      TextView textView2 = (TextView) findViewById(R.id.textView2);
      Button button = (Button) v;
      string = (String) button.getText().toString();
      if  (!string.contains("+")  &&  !string.contains("-")  &&  !string.contains("*")  &&
!string.contains("/")) {

         string1 = string1 + string;
         if (arrayList.size() > 0) {
            arrayList.remove((arrayList.size() - 1));

         }
         arrayList.add(string1);
      } else {
         arrayList.add(string);
         arrayList.add(string);
         string1 = "";
      }
      textView2.setText(textView2.getText().toString() + string);
```

```java
        //textView2.setText(arrayList.toString());
    }

public void onClick(View v) {
    TextView textView1 = (TextView) findViewById(R.id.textView);
    float calc = 0;
    float c = arrayList.size();

    while (c != 1) {
        if (c > 3) {
            if (arrayList.get(3).contains("*") || arrayList.get(3).contains("/")) {
                if (arrayList.get(3).contains("*")) {
                    calc = Integer.parseInt(arrayList.get(2)) * Integer.parseInt(arrayList.get(4));
                }

                if (arrayList.get(3).contains("/")) {
                    calc = Integer.parseInt(arrayList.get(2)) / Integer.parseInt(arrayList.get(4));
                }
                arrayList.remove(2);
                arrayList.remove(2);
                arrayList.remove(2);
                arrayList.add(2, Integer.toString(2));
                c = arrayList.size();
            } else {
                if (arrayList.get(1).contains("+")) {
                    calc = Integer.parseInt(arrayList.get(0)) + Integer.parseInt(arrayList.get(2));
                }
                if (arrayList.get(1).contains("-")) {
                    calc = Integer.parseInt(arrayList.get(0)) - Integer.parseInt(arrayList.get(2));
                }
                if (arrayList.get(1).contains("*")) {
                    calc = Integer.parseInt(arrayList.get(0)) * Integer.parseInt(arrayList.get(2));
                }
                if (arrayList.get(1).contains("/")) {
                    calc = Integer.parseInt(arrayList.get(0)) / Integer.parseInt(arrayList.get(2));
                }
                arrayList.remove(0);
                arrayList.remove(0);
                arrayList.remove(0);
                arrayList.add(0, Integer.toString(calc));
```

```java
                c = arrayList.size();
            }
        } else {
            if (arrayList.get(1).contains("+")) {
                calc = Integer.parseInt(arrayList.get(0)) + Integer.parseInt(arrayList.get(2));
            }
            if (arrayList.get(1).contains("-")) {
                calc = Integer.parseInt(arrayList.get(0)) - Integer.parseInt(arrayList.get(2));
            }
            if (arrayList.get(1).contains("*")) {
                calc = Integer.parseInt(arrayList.get(0)) * Integer.parseInt(arrayList.get(2));
            }
            if (arrayList.get(1).contains("/")) {
                calc = Integer.parseInt(arrayList.get(0)) / Integer.parseInt(arrayList.get(2));
            }

            arrayList.remove(0);
            arrayList.remove(0);
            arrayList.remove(0);
            arrayList.add(0, Integer.toString(calc));
            c = arrayList.size();
        }
    }
    textView1.setText(Integer.toString(calc));
}

public void clear(View v) {
    TextView textView1 = (TextView) findViewById(R.id.textView);
    TextView textView2 = (TextView) findViewById(R.id.textView2);

    string1 = "";
    string = "";
    textView1.setText("0");
    textView2.setText("");
    arrayList.clear();
}
}
```
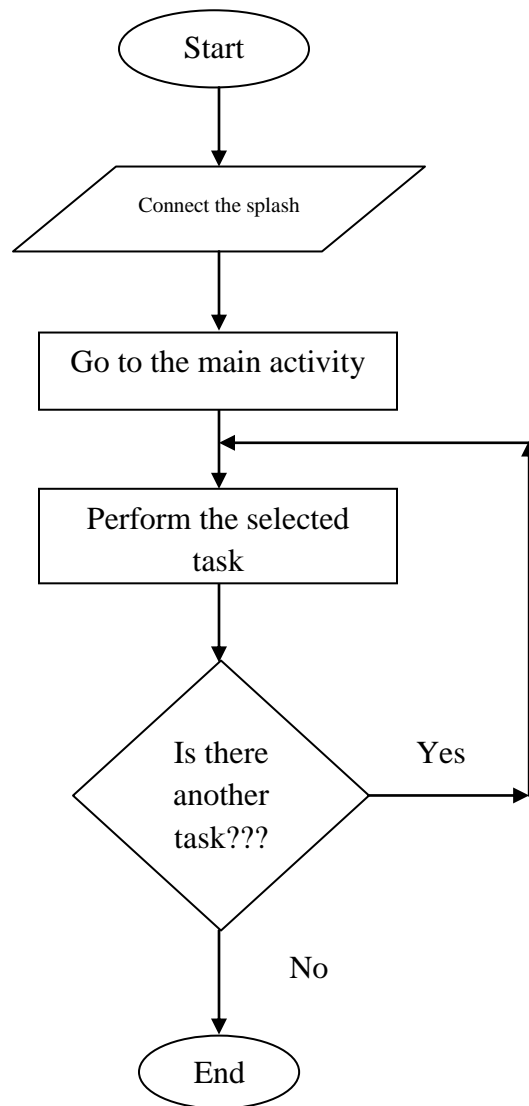
3.5 Program Flowchart

Fig 2: Flowchart of the program

The program starts form the splash screen. The splash screen is a image that is shown when the application starts. The splash screen also has music with it. After 4 seconds the application enters to the main activity.

On the main activity we have a simple calculator that features the functions of the calculator.

Chapter 4
Program Description

4.1 Project implementation

4.1.1 Functions used in splash screen.
The function used in splash screen is mentioned below:

Under the class splash we have

Void run(): This function is for the exception handling.

catch(InterruptedException e): In the exception handling, to catch the error we use this function

void onPause(): This function is to remove the splash screen when we exit the program

4.2.2 Functions used in main activity
Public void clear() :

This function is used on the button 'C' which allows the textboxes to clear the data on it. Putting null value on textbox1 and '0' on textbox.

public void onClick1() :

This function is used on number and operator buttons which allows them to pass their respective values on the textbox1 and order the values on textbox1 and concatenate them while multiple number buttons are clicked.

public void onClick() :

This function is used on the button '=' which allows to calculate the result of the numerical data entered on textbox1 and display the sum on textbox.

References

20th July 2015, examples of calculator, www.android.com

12th July 2015, Array list management, www.androidvibe.com

15th July 2015, Examples of Calculator development, www.stackoverflow.com

16th July 2015, tutorial on android development (basic calculator, www.youtube.com/thenewboston