Software Requirements Specification

for

**Advanced Calculator**

**Version 4.0**

Prepared by **Soumen Dutta (17CS10057)**

**& Anshul Choudhary (17CS10005)**

Indian Institute of Technology, Kharagpur

12 April 2019

**Table of Contents**

Contents

[**1.** **INTRODUCTION** 4](#_Toc5995742)

[1.1 Purpose 4](#_Toc5995743)

[1.2 Product Scope 4](#_Toc5995744)

[1.3 Challenges 4](#_Toc5995745)

[**2.** **Overall Description** 5](#_Toc5995746)

[2.1 Product Perspective 5](#_Toc5995747)

[2.2 Product Functions 5](#_Toc5995748)

[2.3 User Classes and Characteristics 5](#_Toc5995749)

[2.4 Operating Environment 5](#_Toc5995750)

[**3.** **Tentative Timeline** 6](#_Toc5995751)

[**4.** **Functional Requirements** 6](#_Toc5995752)

[4.1 User Interface Requirements 6](#_Toc5995753)

[4.2 System Features 6](#_Toc5995754)

[**5.** **Nonfunctional Requirements** 7](#_Toc5995755)

[5.1 Performance 7](#_Toc5995756)

[5.2 Reliability 7](#_Toc5995757)

[5.3 Usability 8](#_Toc5995758)

[**6.** **Estimated Costs** 8](#_Toc5995759)

[**7.** **Use-Case Diagrams** 8](#_Toc5995760)

[**8.** **Class Diagrams** 9](#_Toc5995761)

[**9.** **Sequence Diagrams** 16](#_Toc5995762)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Advanced Calculator | 13 Feb 2019 | - | Version 1 |
| Advanced Calculator | 22 Feb 2019 | Use Case Diagrams added. | Version 2 |
| Advanced Calculator | 29 March  2019 | Class and Sequence diagrams added. | Version 3 |
| Advanced Calculator | 12 April 2019 | Final Revised version | Version 4 |

# **INTRODUCTION**

## Purpose

The purpose of this document is to present a detailed description of the Advanced Calculator Application. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. The application is designed to act like a handheld scientific calculator with the usual standard functions like add, subtract, multiply, divide, memory and high-level mathematics plus algebra. Additionally, it will have the capability of computing trigonometric functions, exponential functions, differentials, integrals, matrix calculation, graph plotting etc. The program is designed to be user friendly.

## Product Scope

This software is a scientific calculator which have more features as compared to the ordinary calculator. The functions which are involved in this type of calculator are scientific notation, floating point values, logarithmic functions, exponential functions, complex numbers, fractions, equation solving, matrix calculations, calculus, conversion of units and physical constants. The scientific calculator is used in various fields for example in astronomy, geology, physics, chemistry, somehow in biology as well for performing the basic functions as well as complex calculations. This software will be useful for students, scientists, and nearly anyone who performs arithmetic in his daily life. The goal of this application is to provide the users with a one stop solution to all their computation needs. The application will be designed to maximize the productivity of students and minimize the time required for mathematical computation.

## Challenges

Designing an Advanced Calculator require a good understanding of higher mathematics as well as programming language. Simple arithmetic programs are not a big task but performing calculus and matrix algebra might pose a problem. For large calculations, calculator may take too much of time, so the implementation must be efficient. The chief challenge is the implementation of all the above mentioned functions in an optimal manner. As for several of these functions are not available in default java packages, so we may need to use other open source packages for the same.

Apart from these the other challenges that we need to overcome include developing a user friendly GUI design and effective exception handling in case of erroneous input.

# **Overall Description**

## Product Perspective

Advanced Calculator is standalone desktop application that can run on Windows and Linux Platforms. This software is written in java programming language along with python and thus will require at least JRE (Java Runtime Environment) for execution. Apart from that it is independent of any other application.

## Product Functions

The key features of advanced calculator include:

* Scientific notation
* Floating point arithmetic
* Logarithmic functions, using both base 10 and base e
* Trigonometric functions, hyperbolic functions and their inverse.
* Exponential functions and roots beyond the square root
* Quick access to constants such as pi and e
* Fractions calculations
* Matrix calculations
* Calculus (also includes symbolic functions)
* Variable creation and storage
* Graph plot for univariable functions
* All the functions mentioned above involves symbolic calculations too.

## User Classes and Characteristics

The principle users of this product are considered to be students pursuing higher education and professionals who require a scientific calculator for their work. It is considered that the user does have the basic knowledge of operating a desktop computer. Also he/she must have a basic knowledge of mathematics to able to use this product.

## Operating Environment

The software is written in standard Java8 with backend in python. It will use javac as compiler (for conversion from source code to Byte code) and JVM interpreter (to run the byte code).

The software shall use standard java packages and python packages as well as other open source packages required for calculus and matrix algebra. The application is designed to work in a desktop environment that supports JVM and python.

# **Tentative Timeline**

* **Week 1**: Creating and Finalizing SRS Document
* **Week 2**: Coding Basic Arithmetic calculation module
* **Week 3**: Coding Exponential and Trigonometric module
* **Week 4**: Coding Calculus module
* **Week 5**: Coding Unit Converter and Equation solver modules
* **Week 6**: GUI Development (Day - I)
* **Week 7**: GUI Development (Day - II)
* **Week 8**: Testing, debugging and module integration

# **Functional Requirements**

## User Interface Requirements

The calculator in default mode has the following keys: 0,1,2,3,4,5,6,7,8,9,0,M,MS,<B,(,),+,-,/,\*,%,Shift,C,AC,+,log,binomial,n!,Abs,ln,^2,^3sin,cos,tan,sinh,cosh,tanhE,pi,Ans,E^,asin,acos,atan,asinh,acosh,atanh,cbrt,sqrt,a,b,c,d,A,B,C,D,EigenValue,Determinant,Inverse,Transpose,Adjoint,w,x,y and z.

In any situation the calculator has to produce a correct result defined by the well-known  
arithmetic rules. If the calculations are impossible the calculator has to display information helping the user to resolve the erroneous situation, like:

* On encountering a division by 0 the display should read "Cannot divide by 0" and typing the key “C” should reset the calculator.
* On calculating the square root value of a negative operand the display should read "Wrong operand".
* On erroneous operand or operation keys the display should read “Reset (C) to continue” or “Clear (AC) to continue” as appropriate. Of course, any situation can be cleared using the main reset key “C”.

## System Features

* **Algebraic Calculator**

The standard calculator input will include a string composed of numerical symbols, operators, functions (trigonometric, exponential, etc.), variables, etc. which will be evaluated according to standard mathematical convention.

The string previously taken as input will be processed. First all the trigonometric and exponential functions will be evaluated separately and the value will be returned to the Calculation function which will then perform the basic mathematical operation (+, \*, -, /) according to the usual precedence rules. All the above operations will be performed in python using suitable packages.

* **Integration**

In this mode user can compute the indefinite and definite integral of a one variable function. This mode will have a separate interface showing three fields for upper, lower limits and the function to be integrated in terms of variable X. The output after computation will be displayed in the separate field.

* **Differentiation**

In this mode user can compute the derivative of a one variable function at a particular on the real line. This mode will have a separate field for variable X and the point at which the derivative is to be evaluated. If the variable field is empty then it will display the differentiated equation in terms of variable x.

* **Matrix Algebra**

In this mode user can compute the Inverse, Adjoint, Determinant, Eigen Value, Transpose, product, sum, difference and any other valid equation containing up to four matrices. It is also capable of performing symbolic calculations.

* **Graph Plotter**

In this mode user can plot can graphs for most univariate functions. It is very useful feature. Along with graph user can compute the value of function at a given point using the scroll bar given at the bottom of graph GUI. The range of x and y domains can be changed accordingly.

# **Nonfunctional Requirements**

## Performance

The system must be interactive and the delays involved must be less. While performing large calculations, the system must not take more than 5 seconds to output the result. This can be ensured using efficient algorithms and JAVA libraries. The python dependencies makes the calculator a bit slow but also make it capable of performing wide range of functionalities.

## Reliability

As the system provide the right tools for problem solving it must be made sure that the system is reliable in its operations and for giving correct results. It will require a lot of error handling due to large number permutation of possible inputs. Since this software is in its initial stage, it requires a lot of debugging and improvements for actual use or publication.

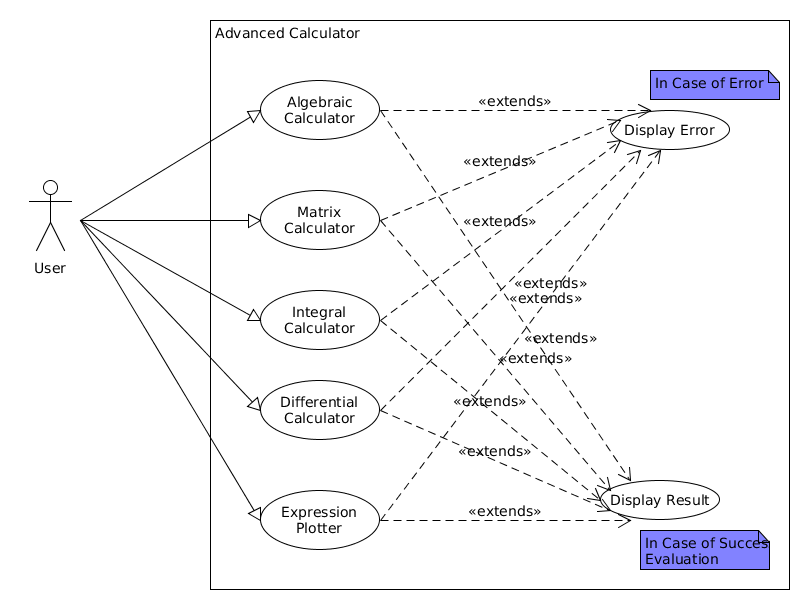
## Usability

The system should be easy to handle and should navigate in the most expected way with no delays. It contains all the necessary operations and some advance operations as well, which increases the usability of the program. The feature of graph plotting really makes it an wonderful software and find its reach to the student community.

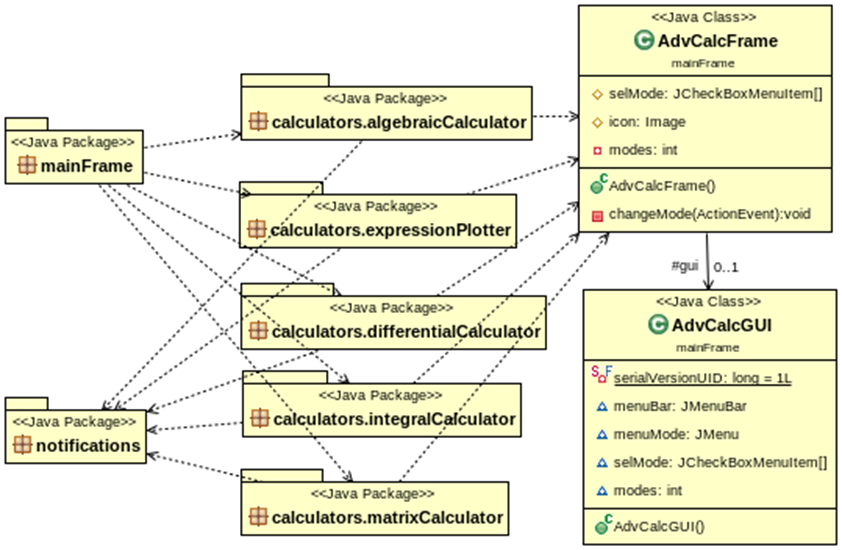
# **Estimated Costs**

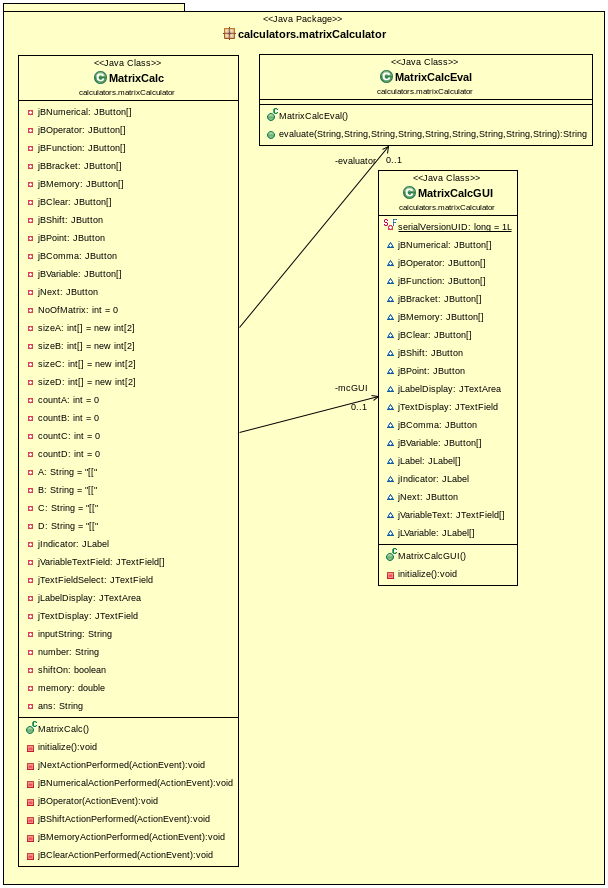
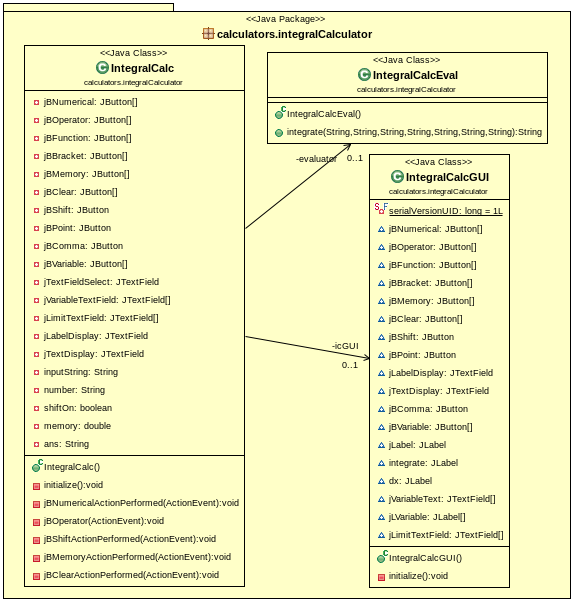
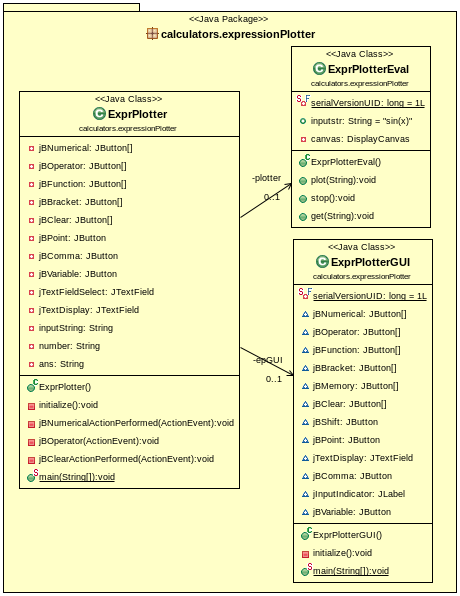
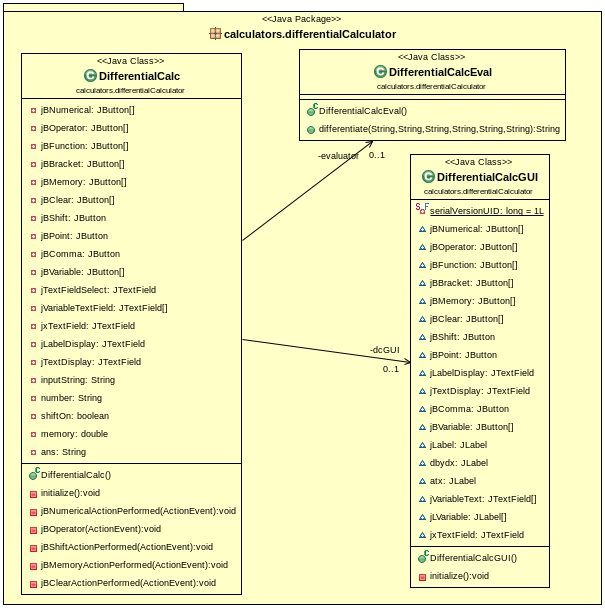
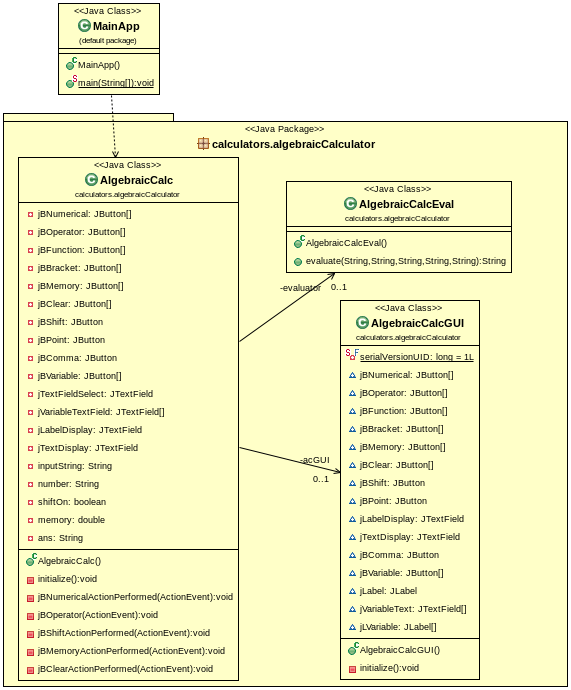
There are several software products available in the market which provide similar features, thus we intend to keep this software as an open-source project. This will allow people to use it freely and contribute to its development. Further, this will allow it to be beneficial for a larger group of people.

# **Use-Case Diagrams**



# **Class Diagrams**





# **Sequence Diagrams**

