1./

GENESIS – Calculator-Mini Project Report



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **To be Approved** | **Remarks/Revision Details** |
| 1 | 07/12/2020 | Ramigani Usharani |  |  |  |
| 2 | 07/12/2020 | Punyashree R |  |  |  |
| 3 | 07/12/2020 | Damera Sowmya |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Details**

Contents

[Contents 3](#_Toc58187397)

[Mini Project – Calculator 4](#_Toc58187398)

[1. Introduction 4](#_Toc58187399)

[**2.** **Requirements** 4](#_Toc58187400)

[Costing 4](#_Toc58187401)

[Aging 4](#_Toc58187402)

[2.2 SWOT Analysis 4](#_Toc58187403)

[2.3 4 W 1 H 5](#_Toc58187404)

[2.4 High Level Requirements 5](#_Toc58187405)

[2.5 Low Level Requirements 5](#_Toc58187406)

[3. Design 6](#_Toc58187407)

[4. Test plan 8](#_Toc58187408)

[4.1 High level 8](#_Toc58187409)

[4.2 Low level 8](#_Toc58187410)

[5. Github Dashboard: 9](#_Toc58187411)

[6. Github Link 11](#_Toc58187412)

[References 11](#_Toc58187413)

# Mini Project – Calculator

# Introduction

A Calculator app is a important application which can be used every day. We have to deal with calculations every day and calculator app allows to perform calculations on the go. A simple calculator app may not be sufficient for many complex equations.

The calculator we have designed will have

* Simple arithmetic calculations like addition, subtraction, multiplication and division.
* Trigonometric functions like sin, cos, tan.
* Scientific operations like square of a given number, square root of a given number and cube of a given number.

## **Requirements**

* 1. Research – Costing and Aging

# Costing

|  |  |  |  |
| --- | --- | --- | --- |
|  | Standard Calculator | Digital Calculator | Scientific Calculator |
| Costing | 3$ - 5$ | 5$ - 7$ | 10$ - 15 $ |

Table 1. Costing

# Aging

|  |  |  |
| --- | --- | --- |
|  | Past | Present |
| Aging | Analog and user had to interact more and performed only some simple operations. | Digital and scientific which performs complex operations. |

Table 2. Aging

# 2.2 SWOT Analysis

|  |  |
| --- | --- |
| **Strengths**   * User Friendly * Fast Calculation * Can be used anywhere and every time | **Weakness**   * Does not perform all mathematical operations * Continuous input is required for each operation |
| **Threats**   * Program may end when wrong input is given. * Fault results for invalid inputs | **Opportunities**   * Saves Human power. * Attracts user with higher efficiency |

Table3. SWOT Analysis

# 2.3 4 W 1 H

* **What** – A calculator is a application which is used to perform mathematical calculations.
* **When** – A calculator is used when the calculations need to be performed are complex.
* **Why** – A calculator saves human power and provides accurate results.
* **Where** – It is used where the calculations are complex.
* **How** – The calculator application takes the input from the user and produces the output.

# 2.4 High Level Requirements

* Calculator must produce the correct result defined by the arithmetic rules.
* If the invalid inputs are given the calculator must display information to resolve the situation.

# 2.5 Low Level Requirements

* All functions are expected to accept only real numbers.
* Prevent user from divide by zero error.

# Design

1. **UML Diagram**

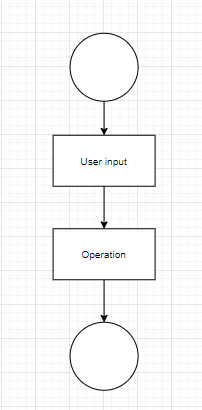


Fig1. UML Diagram

1. **Use Case Diagram**

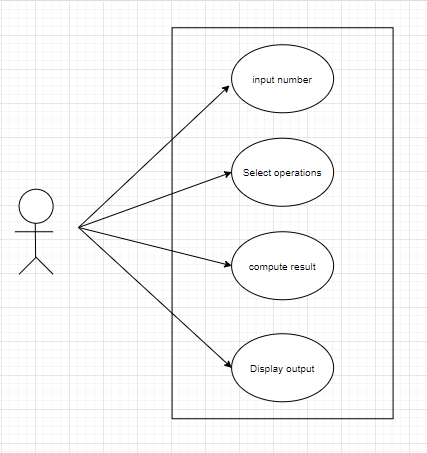


Fig2. Use case diagram

1. **Class diagram**

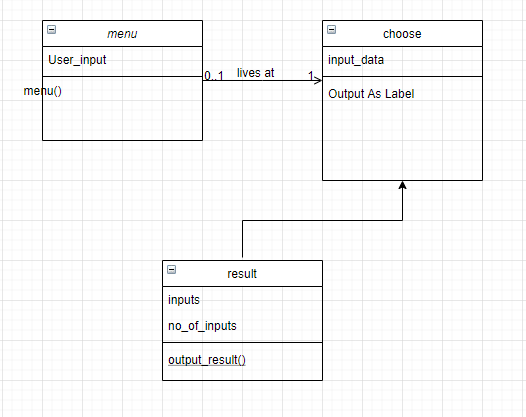


Fig3. Class diagram

1. **Activity diagram**

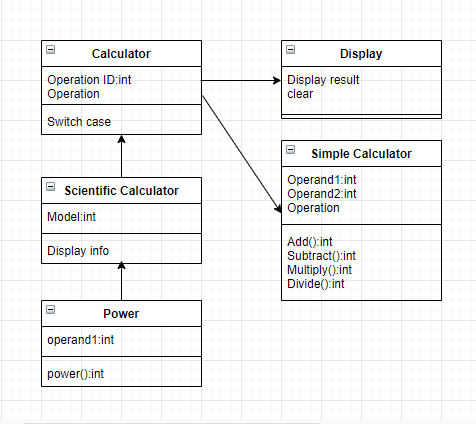


Fig4. Activity Diagram

1. **Sequence diagram**

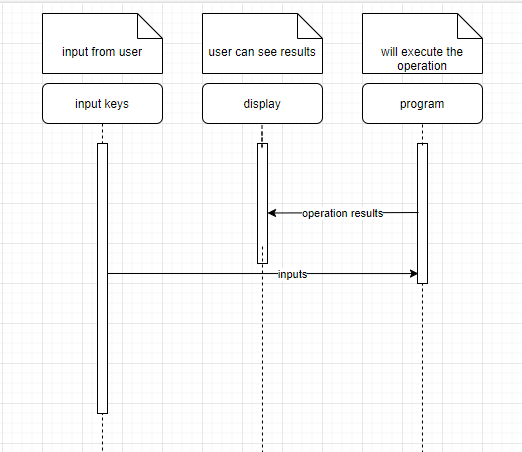


Fig5. Sequence diagram

1. **Flow chart**

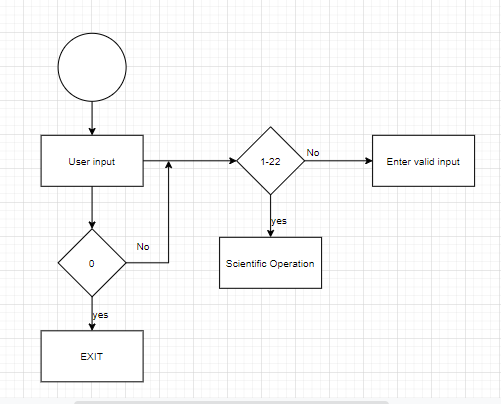


Fig6. Flow chart for simple calculator

# Test plan

# 4.1 High level

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Description** | **Pre-condition** | **Expected input** | **Expected output** | **Actual output** |
| 1 | Get input method | Accept the input from key | Correct input | Will show the inputs in display |  |
| 2 | Mathematical operations | Correct calculation | Correct input | Will do correct operation |  |
| 3 | Display | Display | Proper display | Will display |  |

Table4. High Level test plan

# 4.2 Low level

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case#** | **Description** | **Precondition** | **Expected** **input** | **Expected output** | **Actual output** |
| 1 | Arithmetic Operations | Perform basic arithmetic operations like addition, subtraction, multiplication, division. | 20 + 30  30 - 20  2 \* 3  4 / 2 | 50  10  6  2 |  |
| 2 | Trigonometric Operations | Performs basic trigonometric operations like sin and cos | Sin 30  Cos 45 | 1.732  0.52 |  |
| 3 | Scientific Operations | Performs scientific operations like Modulus, Factorial, power, Square root. | 7 % 2  5!  2^2  Sqrt(144) | 1  120  4  12 |  |

Table5. Low level test plan

# Github Dashboard:

1. **Badges**

Graphical user interface, text, application

Description automatically generated

Fig 7. Badges in Github

1. **Valgrind**

**Graphical user interface, text, application

Description automatically generated**

Fig 8. Valgrind check in Github

1. **Unit testing**

**Graphical user interface, application

Description automatically generated**

Fig9. Unit Testing in Github

1. **Ci framework**

**Graphical user interface, text, application, email

Description automatically generated**

Fig 10. Ci Framework in Github

1. **Cpp check**

**Graphical user interface, text, application, email

Description automatically generated**

Fig 11. Cpp Check in Github

1. Github Link**:**

Link for Github repository-<https://github.com/99003184/calculator>

# References

1. [www.codewithc.com](http://www.codewithc.com)
2. [www.programiz.com](http://www.programiz.com)
3. www.Codeprojects.org