**My Calculator**

-Sharanmegha K V

PS number-99002786

Contents

[List of figures 3](#_Toc53012543)

[Introduction 4](#_Toc53012544)

[1. Requirements 4](#_Toc53012545)

[1.1 Research: 4](#_Toc53012546)

[1.1.1 Aging 4](#_Toc53012547)

[1.1.2 Costing 4](#_Toc53012548)

[1.2 SWOT 4](#_Toc53012549)

[1.3 High Level Requirements 5](#_Toc53012550)

[1.4 Low level Requirements: 5](#_Toc53012551)

[2. Design 6](#_Toc53012552)

[2.1 Structural UML diagrams 6](#_Toc53012553)

[2.2 Behavioural UML Diagrams 7](#_Toc53012554)

[3. Test Plan 9](#_Toc53012555)

[3.1 Test cases (Boundary,Scenario and Requirement Based) 9](#_Toc53012556)

[4. Implementation 10](#_Toc53012557)

[4.1 GIT 10](#_Toc53012558)

# List of figures

|  |  |  |
| --- | --- | --- |
| **Fig No.** | **Description** | **Page No.** |
| 1 | Component diagram of a calculator | 6 |
| 2 | Block diagram of a Calculator | 7 |
| 3 | Use Case diagram for a calculator | 7 |
| 4 | Sequence diagram of a calculator | 8 |
| 5 | Snapshot of the Github Repository | 10 |

# Introduction

An **electronic calculator** is typically a portable electronic device used to perform calculations, ranging from basic arithmetic to complex mathematics. It can perform operations like addition, subtraction, multiplication and division. In addition to this calculator can also find the square root of a number and checks the factorial of a number.

Arithmetic calculations: As a calculator, it seems obvious that this machine will be able to perform basic arithmetic calculations. Addition, subtraction, multiplication etc. will be done with this machine. Also Higher level (scientific) calculations can be done by the user.

# Requirements

## 1.1 Research:

### 1.1.1 Aging

|  |  |
| --- | --- |
| **TIME** | **DESCRIPTION** |
| Before | Simple and client needed to collaborate more and saw more idea to utilize it (not easy to understand) |
| Now | Advanced and logical |
| Future | Voice empower input or anticipating office, neural understanding information |

### 1.1.2 Costing

|  |  |  |
| --- | --- | --- |
| **Sl.no** | **Calculator type** | **Costs** |
| 1 | Standard Calculator | 3$-5$ |
| 2 | Digital Calculator | 7$-10$ |
| 3 | Scientific Calculator | 11$-15$ |

## 1.2 SWOT

|  |  |  |  |
| --- | --- | --- | --- |
| **Strength** | **Weakness** | **Opportunities** | **Threats** |
| Fast calculation | Damaged under 30mm water | Healthy programming practice | Fault results for invalid inputs |
| Water proof | not robust and axile | chance to get pro skills in c and c++ and git | git Hardware acceleration |
| Large calculation | Basic knowledge needed | Business standard ideas | Imaginary and indefinite number or infinite number display problem |
| Basic arithmetic operations | Need to know prerequisites |  |  |
| Matrix , power and root extra | Continuous inputs |  |  |

## 1.3 High Level Requirements

* Output: This function will display the results of a calculation or memory.
* Results should be displayed under the following conditions:
  1. Whenever the equals sign (=) is input.
  2. Optionally, whenever the ENTER key is pressed

## 1.4 Low level Requirements:

System Interfaces

1) Arithmetic: the system will use the built in ALU to compute all the arithmetic that will be done for the calculator.

2) Upper level Calculations: The programmers of the machine will write in functions that the machine will be able to call to produce upper level calculations

# Design

## Structural UML diagrams

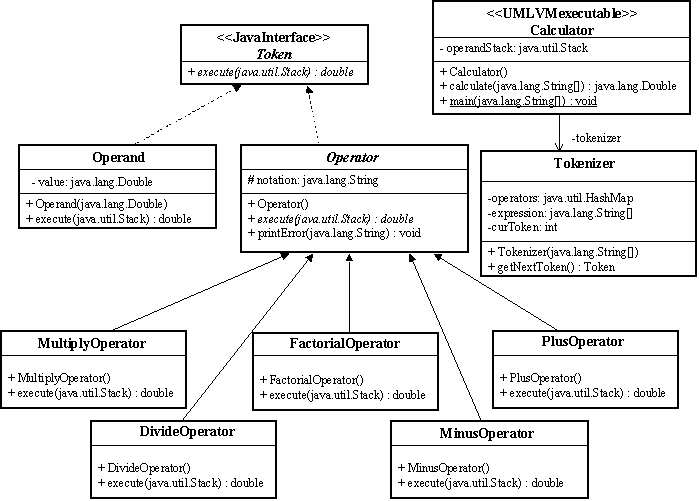
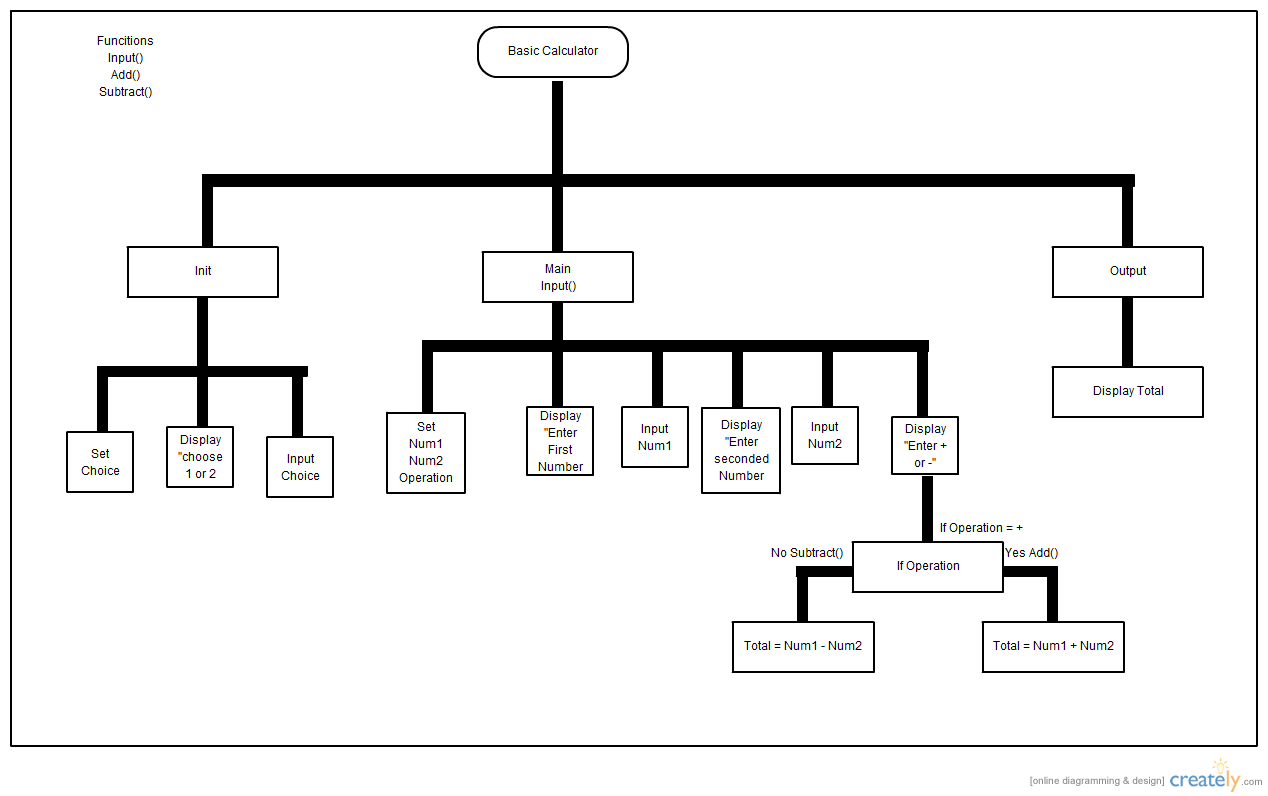


Fig. 1: Component diagram of a calculator

Fig. 2: Block diagram of a Calculator

## 2.2 Behavioural UML Diagrams

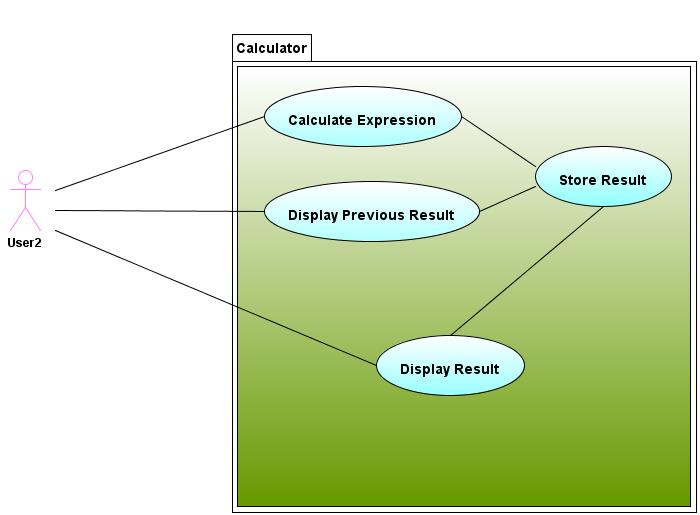


Fig. 3: Use Case diagram for a calculator

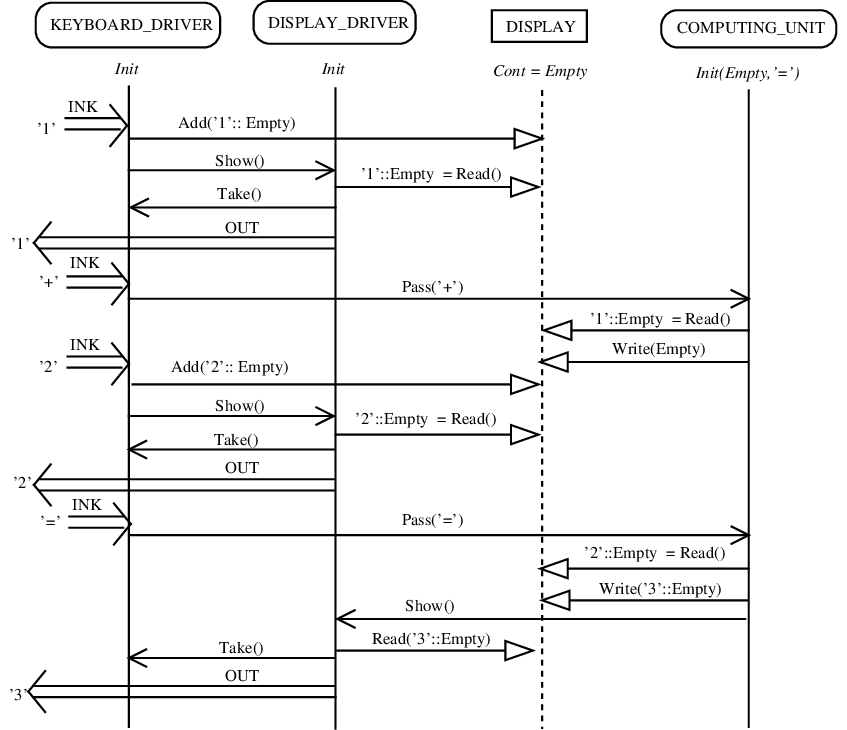


Fig. 4: Sequence diagram of a calculator

# Test Plan

## 3.1 Test cases (Boundary,Scenario and Requirement Based)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **DESCRIPTION** | **PRECONDITION** | **EXPECTED INPUT** | **EXPECTED OUTPUT** | **ACTUAL OUTPUT** | **ACTION** |
| 1 | Verify the addition of two numbers | Addition operator must be used in between the digits | Add 20 and 10 | 30 | expected output | pass |
| 2 | Verify the addition of two numbers | Subtraction operator must be used in between the digits | Subtract 0 and -3 | -3 | Expected output | pass |
| 3 | Verify the addition of two numbers | Subtraction operator must be used in between the digits | Subtract 1000 and 900 | 1 | Not as expected output | Fail |
| 4 | Verify the addition of two numbers | Multiplication operator must be used in between the digits | Multiply 0 and 2 | 0 | expected output | pass |
| 5 | Verify the addition of two numbers | Multiplication operator must be used in between the digits | Multiply 2 and 5 | 2 | Not as expected output | Fail |
| 6 | Verify the addition of two numbers | Division operator must be used in between the digits | Divide 1 and 0 | 0 | As expected output | pass |
| 7 | Verify the addition of two numbers | Division operator must be used in between the digits | Divide 2 and 3 | 3 | Expected output | pass |

# 4. Implementation

## 4.1 GIT

GIT Repository link: <https://github.com/99002786/act2>

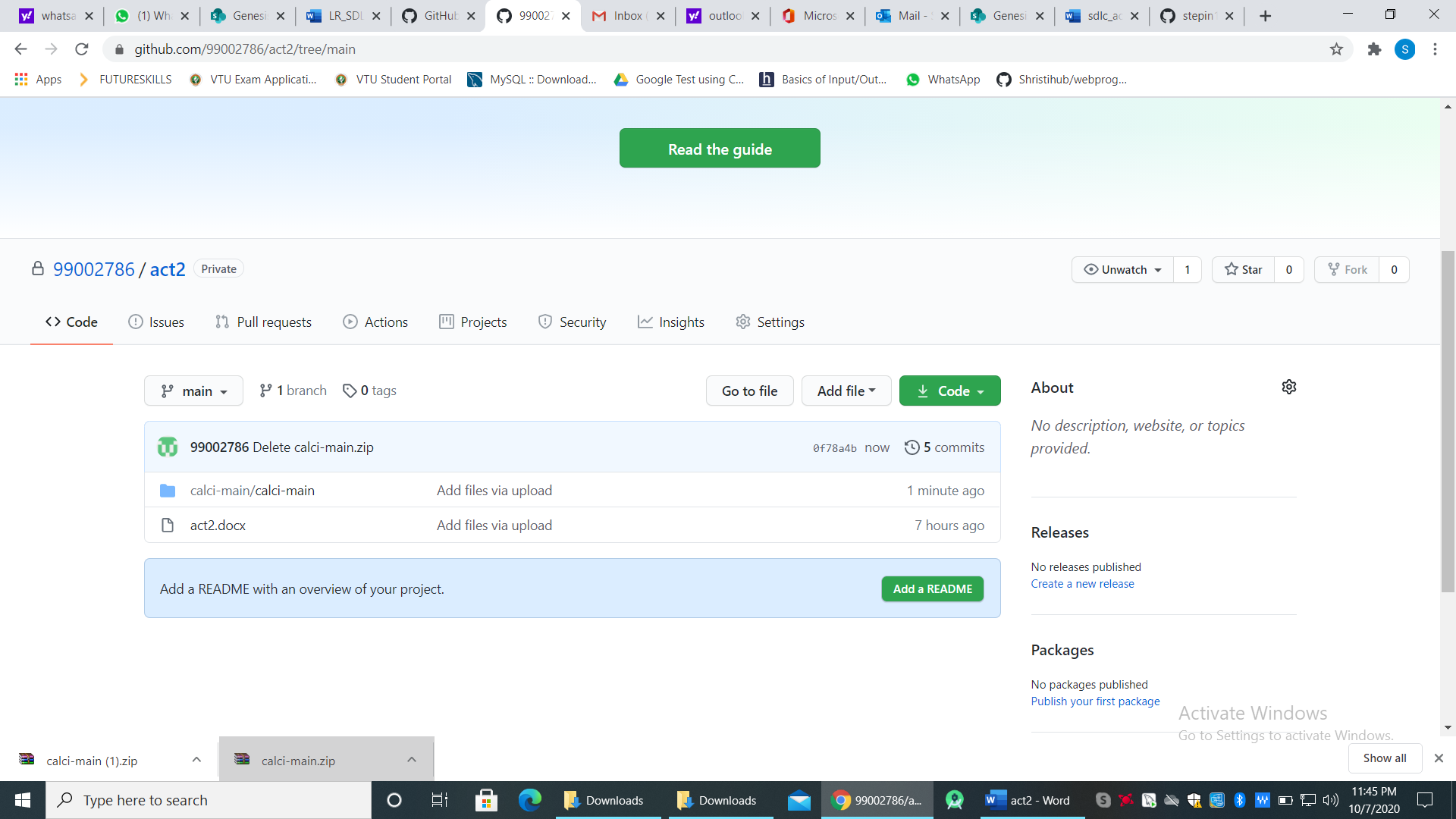


Fig 5: Snapshot of the Github Repository