./

GENESIS - Learning Outcome & Mini-project Summary Report



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **To be Approved** | **Remarks/Revision Details** |
| 1 | 07/12/2020 | Sushma SM  A Joseph Vijay Kumar  Androthu Murali Satya Pavan |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Details**

Contents

[Contents 3](#_Toc58228014)

MINIPROJECT [TEAM]…………………………………………………………………………………………………………………………………………….5-17

SMART CALCULTOR USING C……………………………………………………………………………………………………………………………5

1. INTRODUCTION……………………………………………………………………………………………………………………………………………5

2. REQUIREMENTS………………………………………………………………………………………………………………………………………..5-7

a. Aging……………………………………………………………………………………………………………………………………………………………5

b. Costing…………………………………………………………………………………………………………………………………………………………6

c. SWOT Analysis……………………………………………………………………………………………………………………………………………..6

d. Requirements Analysis…………………………………………………………………………………………………………………………………7

3. DESIGN………………………………………………………………………………………………………………………………………………………..8

a. High Level Design…………………………………………………………………………………………………………………………………………8

b. Low Level Design…………………………………………………………………………………………………………………………………………8

4. UML DIAGRAMS……………………………………………………………………………………………………………………………………..8-12

a. Structural Diagram……………………………………………………………………………………………………………………………………8-9

i. Deployment Diagram……………………………………………………………………………………………………………………………………9

ii. Object Diagram..………………………………………………………………………………………………………………………………………….9

b. Behavioral Diagram………………………………………………………………………………………………………… ………………….10-12

i. High Level Use Case Diagram………………………………………………………………………………………………………………………10

ii. Low Level Use Case Diagram………………………………………………………………………………………………………………………10

iii. Activity Diagram………………………………………………………………………………………………………………………………………..11

iv. Sequence Diagram…………………………………………………………………………………………………………………………………….12

5. TEST PLAN…………………………………………………………………………………………………………………………………………….12-14

5.1 Unit Testing……………………………………………………………………………………………………………………………………………..14

5.2 Integration Testing…………………………………………………………………………………………………………………………………..14

6. GITHUB ACTIONS……………………………………………………………………………………………………………………………………….15

7. REFERNCES………………………………………………………………………………………………………………………………………………..15

8. AGILE ASPECTS……………………………………………………………………………………………………………………………………..15-16

8.1 Theme……………………………………………………………………………………………………………………………………………………..15

8.2 Epic………………………………………………………………………………………………………………………………………………………….15

8.3 User Stories……………………………………………………………………………………………………………………………………………..16

9. SUMMARY…………………………………………………………………………………………………………………………………………………16

CHALLENGES FACED AND OVERCOMMED……………………………………………………………………………………………………..17

**LIST OF FIGURES**

Figure 1 Aging of smart calculator

Figure 2 Costing of smart calculator

Figure 3 Deployment diagram of smart calculator

Figure 4 Object diagram of smart calculator

Figure 5 High level use case diagram of smart calculator

Figure 6 Low level use case diagram of smart calculator

Figure 7 Activity diagram of smart calculator

Figure 8 Sequence diagram of smart calculator

Figure 9 Github actions flow of smart calculator

**LIST OF TABLES**

Table 1 Aging of smart calculator

Table 2 Costing of smart calculator

Table 3 SWOT analysis of smart calculator

Table 4 Requirement analysis of smart calculator

Table 5 High level design of smart calculator

Table 6 Low level design of smart calculator

Table 7 Test plan of smart calculator

Table 8 Unit testing of smart calculator

Table 9 Integration testing of smart calculator

# Miniproject [Team]

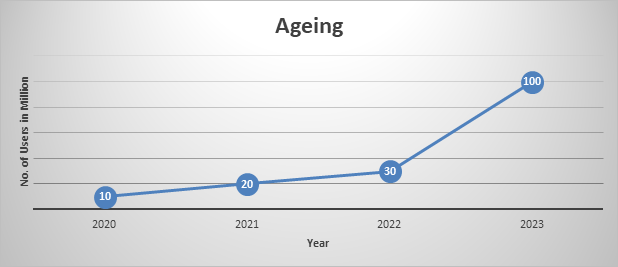
**SMART CALCULATOR USING C**

1. **INTRODUCTION**

Scientific calculator is a device which performs arithmetic operations on numbers, trigonometric operations and other conversions such as factorial, square, temperature from centigrade to Fahrenheit of a number and so on. The simplest calculators can do only addition, subtraction, multiplication, and division. More sophisticated calculators can handle exponential operations, roots, logarithms, trigonometric functions, and hyperbolic functions. Calculator system is a tool to change mathematical knowledge and sophisticated problem-solving strategies. The calculator will also be able to compute the power function. The calculator is an electronic device, which is used to calculate anything from anywhere, anytime.

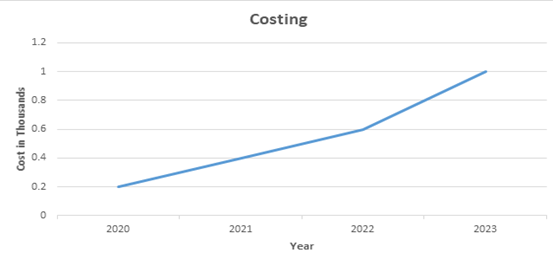
1. **REQUIREMENTS**
2. **Aging:**

|  |  |
| --- | --- |
| **Year** | **Description** |
| 1978 A.D. | The world's first solar powered calculators were released |
| 1978 A.D. | Casio releases its first miniature calculator |
| 1993 A.D. | Apple releases the first personal digital assistant the Newton |
| 2001 A.D. | Microsoft develops Power Calculator for Windows XP |
| 2005 A.D. | Microsoft releases Windows Calculator Plus |
| 2009 A.D. | Microsoft launches Windows 7 with the newest version of Windows calculator |
| 2011 A.D. | Apple releases ios5 that brings modern calculators to smartphone users |



1. **Costing:**

|  |  |
| --- | --- |
| **Model Name** | **Price** |
| TEAM\_5 Mk 8 | $300 |
| Sharp CS10A | $1,490 |
| Casio Mini Card LC-78 | 15,95$ |
| Apple newton | 70$ |
| Apple ios5 | 1,199 ₹ |



1. **SWOT ANALYSIS**

|  |  |
| --- | --- |
| **Strengths**   * Many functions in one system * Provides instance solution to given numbers based on operations * It gives an accurate result * Simple design * User friendly interface | **Weakness**   * Data cannot be stored * Can’t perform multiple operations at a time * Limits the knowledge of users * Limited number of operations * Accepts only 2 or 1 operand only |
| **Opportunities**   * Increases business by implementing new functions * Technology advances | **Threats**   * May not give expected output due to the precedence defined * For longer digit numbers operations cannot be performed |

1. **Requirement Analysis**

|  |  |
| --- | --- |
| **Product ID** | **Description** |
| HL\_01 | Speed and accuracy should be more |
| HL\_02 | Processor should be multifunctional |
| HL\_03 | System should be compact |
| HL\_04 | Battery lifetime should be more |
| HL\_05 | Precision should be high |
| H\_01\_L\_01 | Speed should be in picoseconds |
| H\_02\_L\_02 | The functions should be selected and should produce correct output. |
| H-03\_L\_03 | Precision value should be 10power -15 |
| L\_04 | Process time should be less than nanoseconds |

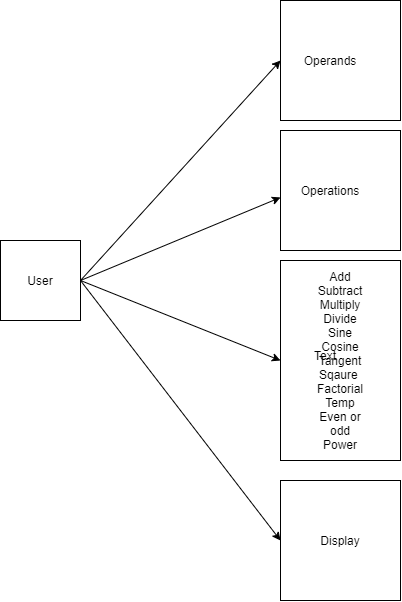
1. **DESIGN**
   1. **High Level Design**

|  |  |
| --- | --- |
| **ID** | **Description** |
| HL\_01 | Operations are provided to the user |
| HL\_02 | User provides the operand and operator to perform specific operations |
| HL\_03 | User can reiterate the operation with the previously computed results |
| HL-04 | user should be displayed with the result after every single computation and later it can be chosen to continue the same operation |

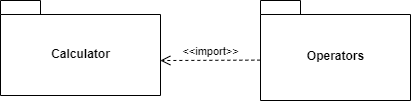
* 1. **Low Level Design**

|  |  |
| --- | --- |
| **ID** | **Description** |
| LL\_01 | User can choose any of the operations to be performed i.e., addition, subtraction, multiplication, division, sine, cosine, tangent, square root, power, factorial, temperature and even or odd |
| LL\_02 | User provides the operands i.e., num1 and num2 or num to be computed by the selected operator |
| LL\_03 | When the user chooses, 0 functionality from the menu . He/she should be taken out from the system. |
| LL\_04 | The user can continue to compute new operations by opting 1 |

1. **UML DIAGRAMS**
   1. **Structural Diagrams**
      1. **Deployment Diagram**

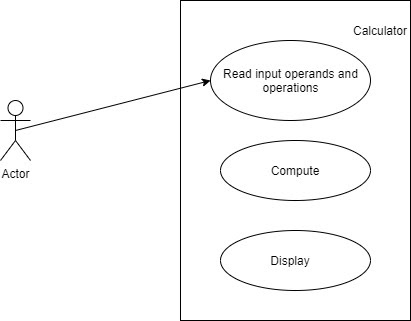
****

* + 1. **Object Diagram**

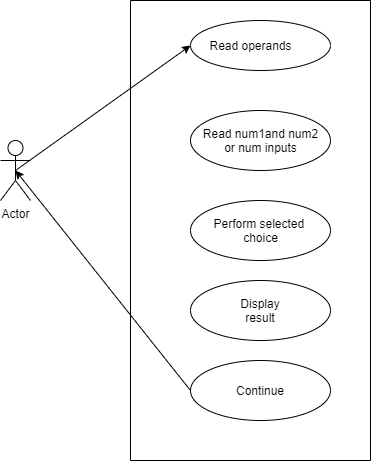
****

* 1. **Behavioral Diagrams**

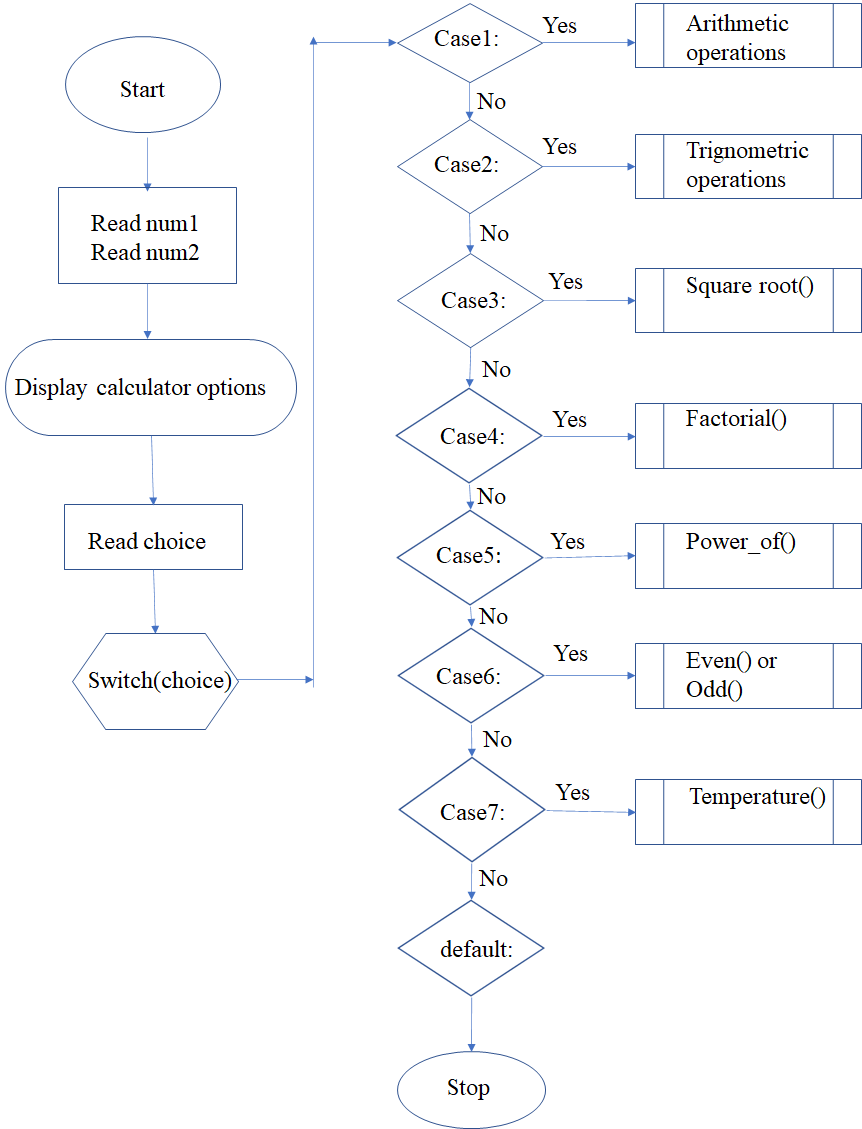
## **High Level Use Case Diagram**

****

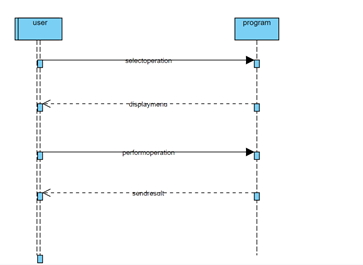
* + 1. **Low Level Use Case Diagram**

****

* + 1. **Activity Diagram**

****

* + 1. **Sequence Diagram**

****

1. **TEST PLAN**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Precondition** | **Expected I/p** | **Expected o/p** | **Actual o/p** |
| TC\_HL\_01 | Input from user | Accept the input | Provide the numbers and operation to be performed | Will show the inputs in display | Same as expected |
| TC\_HL\_02 | Maths operations | Correct Calculation | Correct input symbol as pressed by the user. | Will do correct calculation | Same as expected |
| TC\_HL\_03 | display | Display output | Proper display of output which can be interpreted by humans. | will display results clear visibility |  |
| TC\_LL\_01 | Format of result | Format of the output can be more than one type | No expected input | Will show error or fatal error | Operation has performed |
| TC\_LL\_02 | Divide by zero | If the numerator is zero | decimal with number 0 | 0 | 0 |
| EX-01 | Addition of 2 numbers | User input | 7 and 3 | 10 | 10 |
| EX-02 | Subtraction of 2 numbers | User input | 5 and 2 | 3 | 3 |
| EX-03 | Division of 2 number s | User input | 4 and 2 | 2 | 2 |
| EX-04 | Multiplication of 2 numbers | User input | 5 and 7 | 35 | 35 |
| EX-05 | Square of a number | User input | 25 | 5 | 5 |
| EX-06 | sine | User input | 30 | -0.98 | -0.98 |
| EX-07 | cosine | User input | 0 | 1 | 1 |
| EX-08 | Factorial of a number | User input | 3 | 6 | 6 |
| Ex-09 | tangent | User input | 45 | 1.61 | 1.61 |
| EX-10 | Power of number | User input | 4 and 2 | 16 | 16 |

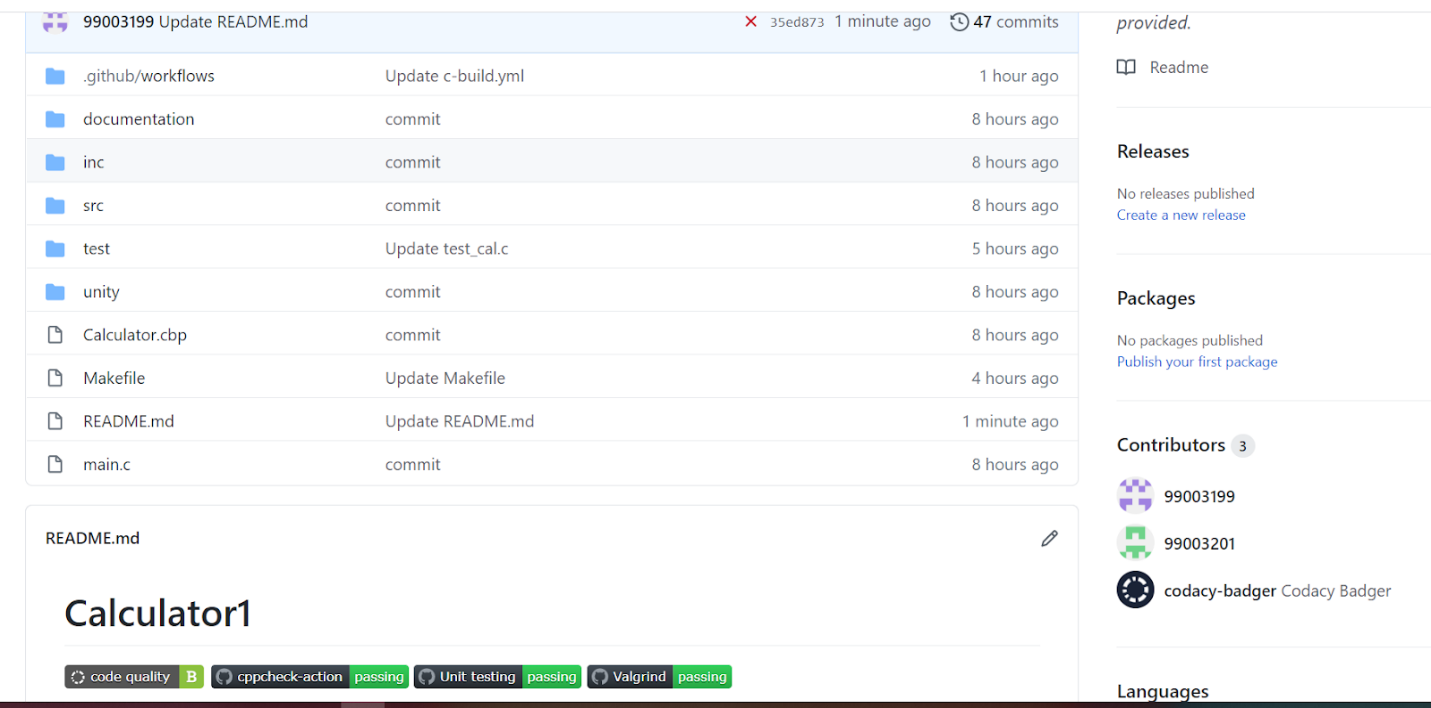
**5.1 Unit Testing:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Description** | **Expected input** | **Expected output** | **Actual output** |
| T01 | Beginning of application | Start | Display of different types of operations | Same as expected |
| T02 | Selecting operations from menu | Choosing a user desired operation | Respective operation will be performed | Same as expected |
| T03 | Quit from the application | Choose 1 option | Exiting from application | Same as expected |

**5.2 Integration testing:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Expected input** | **Expected output** | **Actual output** |
| L01 | Inputting the  operands | Input operands based on operation | Respective numbers will be displayed | Same as expected |
| L02 | Performing multiple operation | Choosing continue after first operation | Select the operation to be performed | Same as expected |

1. **GITHUB ACTIONS**



**Github link**: <https://github.com/99003199/Calculator1>

1. **REFERENCES**

<https://fresh2refresh.com/c-programming/c-programs/c-code-for-calculator-application/>

<https://www.programiz.com/c-programming/examples/calculator-switch-case>

1. **AGILE ASPECTS**

* Theme
* Epic
* User Story

**8.1 Theme**

To develop a smart calculator which displays 10 types of operations based on the option selected by the user. The user selects the option through keypad facility and the calculator performs the selected operation and finally displays an option whether the user as to continue or not.

**8.2 Epic**

* The product should perform the selected choice when user clicks the button
* The product should stop performing the operation when user opts 0
* The product should display the result of the operation performed by the user

**8.3 User Stories**

* As a user, I want the product to perform the operation user

Acceptance Criteria:

1. The product should perform operation when user clicks on the keypad.
2. The product should continuously perform the operation when user selects 1 to continue.
3. If option selected by user is not available then it should display invalid option.

* As a user, I want the product to stop performing the operation when 0 is selected.

Acceptance Criteria:

1. The product should perform the operation until the user wishes to stop.
2. The product should stop automatically when 0 is selected.

* As a user, I want the product to perform the operations based on the conditions provided

Acceptance Criteria:

1. The product should take inputs either 1 or 2 based on the operation to be performed.
2. The product should display the results in appropriate proportions.
3. **SUMMARY**

As a team of three we are assigned with a miniproject i.e. Smart Calculator using C which includes documentation (which includes introduction, requirements, aging, costing, SWOT analysis, design, UML diagrams, test plan, references, git actions, using a concept of agile we have to write epic, theme and user stories), committing the code to Git and creating badges with help of workflows.

I, Sushma S M, my work in the project is I have chosen trigonometry functions (such as sine, cosine, tangent) and to convert the temperature from centigrade to Fahrenheit. Done with coding and testing with help of codeblocks for the functions which I have chosen. While coming to documentation my work is high level and low level use case diagram, references, Github actions, agile aspects (such as epic, theme, 1 user story). Finally I combined all the functions into one code and committed to git repository and created badges for the workflows.

A Joseph Vijay Kumar has opted arithmetic operations such as addition, subtraction, multiplication, division. He wrote the functions for the operations which he has opted and done with testing part also. While in documentation, his part of work is test plan, activity diagram, sequence diagram and user story. And he committed the requirements and architecture to the Git repository.

Androthu Murali Satya Pavan has opted finding square root, factorial, even or odd and power of given operands. He wrote the functions for the operations which he has opted and done with testing part. While in documentation, his part of work is introduction, aging, costing, requirements, object deployment diagram and user story. And he committed the test plan to the Git repository.

**CHALLENGES FACED AND OVERCOMMED**

* As we are new to the CodeBlocks IDE, we referred a lot of videos; websites and Stepin connect session’s recordings to setup the tool and how to use it.
* To do the unit testing, we struggled a lot as this is new to us. To overcome this we referred the connect session and also a few websites.
* As we are new to use Git, firstly we just uploaded a code directly to the Git account; we faced problems in creating badges for workflows. Later on, we came to know that if we push the code using Git Bash then we create a badges using workflow. This problem has been overcommed with help of friends.
* To write the flows, we referred a Stepin654321 git account; even after referring it we were not able to generate badges. Later we referred our friend’s workflows and created badges.