./

Learning Report – Applied System Development Life Cycle and Software Testing



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver.Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **To be approved By** | **Remarks/Revision Details** |
| 1 |  | Name/PS No | Name/PS No | Module Owner Name | Comments |
| 2 | 15/02/2021 | Suvradeep Dutta | Pujitha(99003787) | Srinivas, Prithvi |  |
| 3 | 15/02/2021 | Suvradeep Dutta | Souvik Jana (99003784) | Srinivas, Prithvi |  |

**Document History**

CONTENTS :

[ACTIVITY 1: SYSTEM/ SOFTWARE DEVELOPMENT 4](#_Toc53129064)

**research……………………………………………………………………………………………………………………………………………………………...4**

**FEATURES………………………………………………………………………………………………………………………………………………………………7**

[**SWOT ANALYSIS…………………………………………………………………………………………………………………………………………………….**](#_Toc53129067)**7**

**4W & 1H…………………………………………………………………………………………………………………………………………………...8**

[**REQUIREMENTS** **7,8**](#_Toc53129068)

[**DESIGN**……………………………………………………………………………………………………………………………………………………………………….**9**](#_Toc53129069)

[HIGH LEVEL DESIGN………………………………………………………………………………………………………………………………………………..](#_Toc53129070)**11**

[LOW LEVEL DESIGN](#_Toc53129071) **12**

**TEST PLANS………………………………………………………………………………………………………………………………………….12**

[ACTIVITY 2: AGILE METHODOLOGY 13](#_Toc53129074)

[**THEME** **13**](#_Toc53129075)

[**EPIC** **13**](#_Toc53129076)

[**USER STORY** **13**](#_Toc53129077)

## Table of Figures

[Figure 1 DEPLOYMENT DIAGRAM(HIGH LEVEL) 14](#_Toc52177314)

[Figure 2 USE CASE DIAGRAM (LOW LEVEL) 15](#_Toc52177317)

[Figure 3 CLASS DIAGRAM (LOW LEVEL) 16](#_Toc52177318)

[Figure 12 GIT COMMIT](#_Toc52177325) 17

[Figure 13 GIT PROFILE 18](#_Toc52177326)

[Figure 14 GIT ISSUE 19](#_Toc52177327)

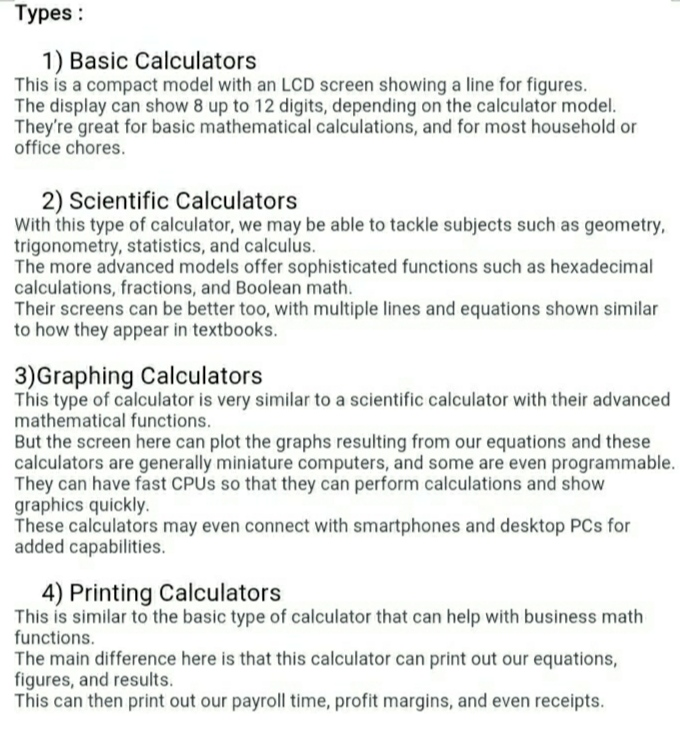
[Figure 16 GIT ACTIONS 20](#_Toc52177329)

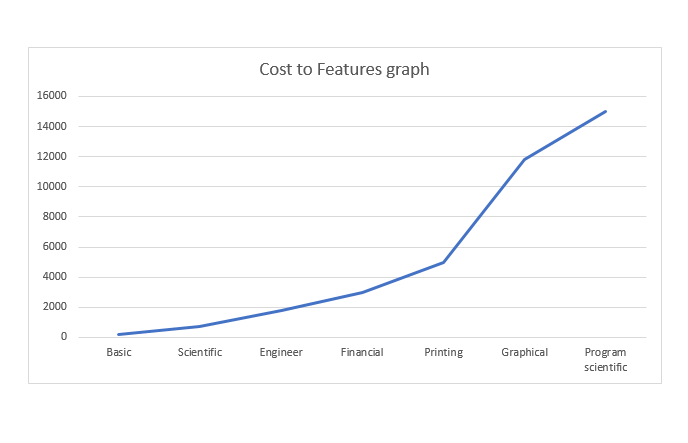
[Figure 20 GIT CODE QUALITY 2](#_Toc52177333)1

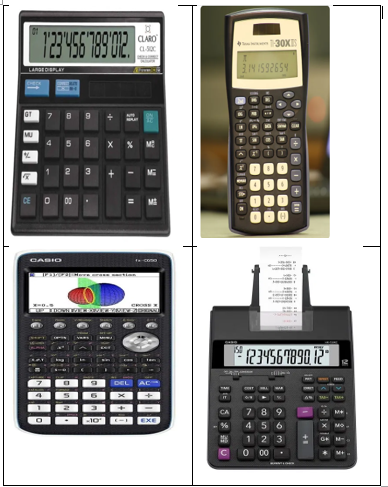
**Activity -1: System/Software Design**

**Link:** [**https://github.com/99003784/N9\_SDLC\_CALCULATOR**](https://github.com/99003784/N9_SDLC_CALCULATOR)( GIthub repository link )

**RESEARCH :**

****

****

****

**OUR PRODUCT**

**FEATURES:**

1. The calculator will solve all basic arithmetic operations. (addition, subtraction, multiplication and division)
2. The calculator will solve all the exponential, logarithmic operations, power of a number, and factorial of a number.
3. The calculator is going to solve all trigonometric and inverse trigonometric operations.
4. The calculator is going to have a clear function which will clear the calculator display.
5. The calculator will have a liquid crystal display i.e. LCD Display.

**SWOT ANALYSIS:**

**STRENGTHS:**

1. Our product is very cost efficient.
2. It is very user friendly since it has limited functions.
3. It will work on solar power and has 1.5v battery as a battery backup and has an automatic power off.

**WEAKNESS:**

1. It has limited operations. People trying to do calculations regarding complex numbers and other things regarding statistics will not be able to do so.
2. The battery inside a solar calculator is mostly alkaline button cell which cannot be recharged.

**OPPORTUNITIES:**

1. The price of the product is less than other products with same features that are available in the market.
2. The product will do very well in shop counters as the calculator is very user friendly and anybody can use it without prior knowledge of that calculator.
3. Students till class 10 will be attracted to this product because of its simplicity in operations and design.
4. The product will also work in banking sectors and other government sectors where they want low price, minimum features, handy products.

**THREATS**

1. Students who are pursuing higher education won’t be interested in this product as it has limited operations.
2. Several other competitors are there who are willing to make the product with more features.
3. One of the threats in the product is its battery which cannot be recharged.

**4W & 1H :**

**WHAT:** It is a simple electronic hardware device or software that are capable of performing the simple calculations such as addition, subtraction, multiplication, division , calculating power of number, exponential function, logarithmic function, permutation and combination, trigonometry ,inverse-trigonometric functions, factorial of a number, binary to decimal conversion etc.

**WHEN:** 1) Useful during exams, for getting complex calculation in very less time.

1. Calculation of bills in malls , shops, and restaurants .

**WHERE:** 1) Exam hall

1. Shop Counters
2. Colleges and schools.
3. Banking sectors

**WHY:** 1) Complex calculations are very tough to calculate in less time , so this calculator is used for simple operations.

1. We get the result quickly and accurately.
2. We save our valuable time by using the product.
3. Saves human power.

**HOW:** 1) Input will be given by the user from the keyboard and the result will be displayed to the screen.

1. Write the code for all the requirements.
2. Use one programming language to code the functions(C).
3. Use Github and visual studio for making and building file for required specification.
4. Check all functionalities.

**High Level Requirements**

## 1) Maximum input digits a user can enter is up to 12 digits.

## 2) Arithmetic Operations

## 3) Exponential and Logarithmic Operations.

## 4) Square roots and powers.

## 5) Permutation and Combinations.

## 6) Finding area of different geometrical shapes like circle, square, rectangle, triangle.

## 7) Finding volume of different geometrical shapes like cube, cylinder, cone, sphere.

## 8) Trigonometric And Inverse Trigonometric Functions.

## 9) Measurements conversion (km, cm, Inch, etc. ).

## 10) weight conversion (kilogram, liter, etc. ).

## 11) Mixed to improper fractions.

## 12) Cube and cube root.

## 13) Temperature conversion.

## 14) Arithmetic operations of fractions. (+,-,\*,/) .

## 15) Calculation of Reminder.

## 16) Calculation of simple interest.

## 17) Binary to decimal conversion.

## **Low Level Requirements**

1) Exponential operation (result= b ^ x where b=base and x=exponent. Input type : integer and float)

Logarithmic operation (result=log(x) where the input type of x is integer or float) .

2) Area of different geometrical shapes (The input data type used in this scenario are integer, float and long).

3) Volume of different geometrical shapes( The input data type used in the scenario are integer and float.)

4) Permutation and Combination (result=nPr; input type : integer and result=nCr; input type: integer) .

5) using float for Decimal values .

6) using integer for Integer values .

7) using Double .

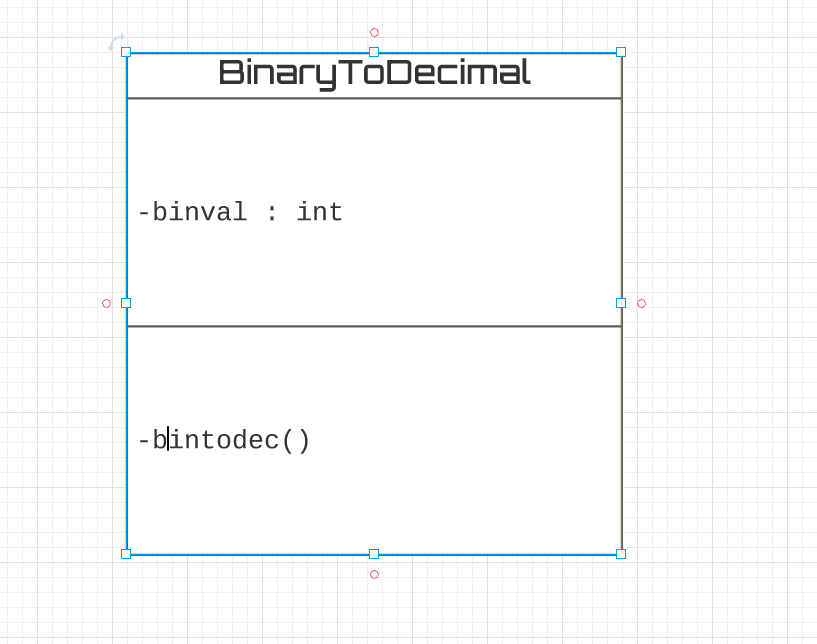
8) calculation of reminder using modulus operator .

9) using arithmetic operators for calculation of simple interest .

**DESIGN:**

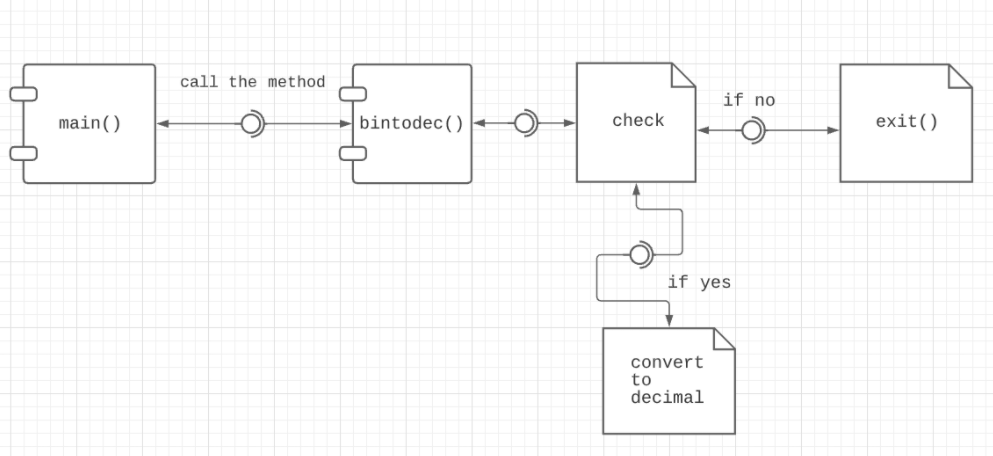
**HLR**\_**UML\_DIAGRAM:**

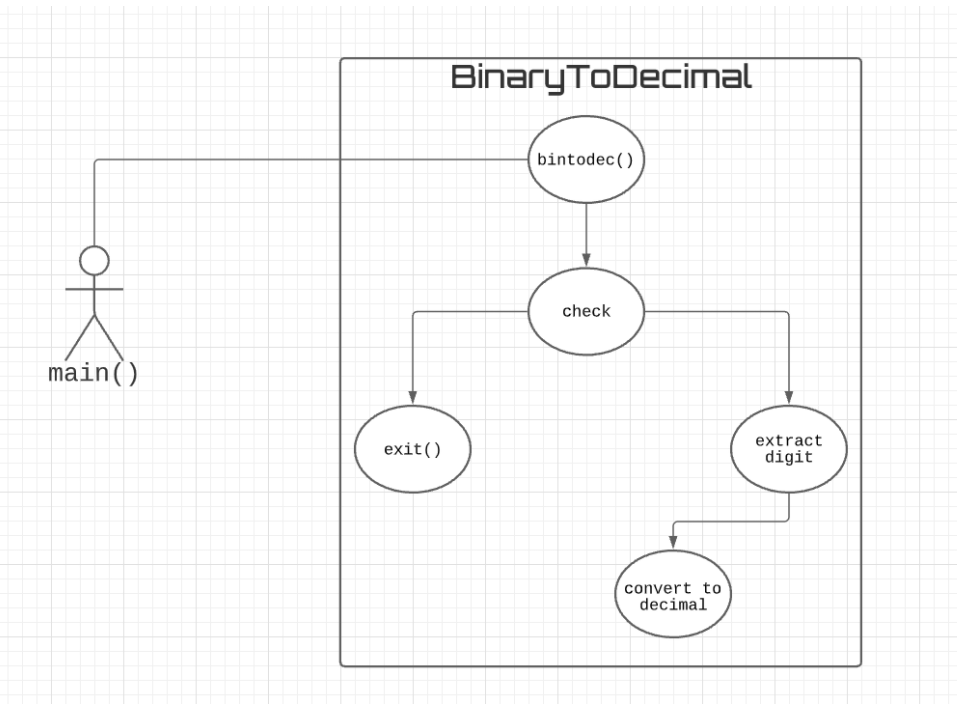
* ***CLASS DIAGRAM*:**



**LLR\_UML\_DIAGRAMS:**

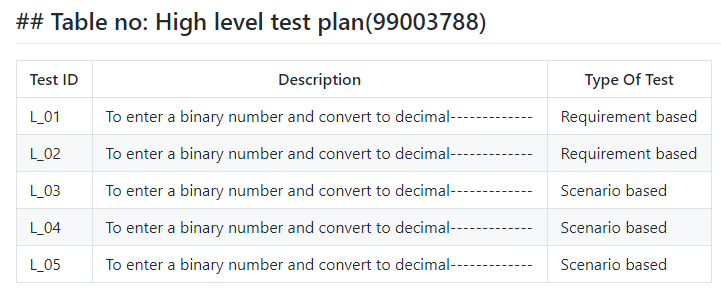
* ***COMPONENT DIAGRAM:***



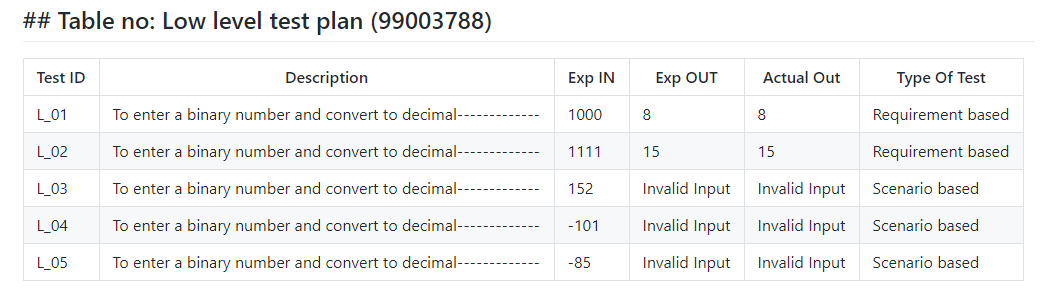
* ***USE CASE DIAGRAM: ***

-

## **Table no: High level test plan**



## **Table no: Low level test plan**



**ACTIVITY 2: AGILE METHODOLOGY:**

**THEME:**

The theme is designing a calculator with certain features according to the specific requirements. The target customers for the calculator are students, shop keepers, banking executives and engineers.

**EPIC:**

Function for converting a Binary number to its Decimal equivalent

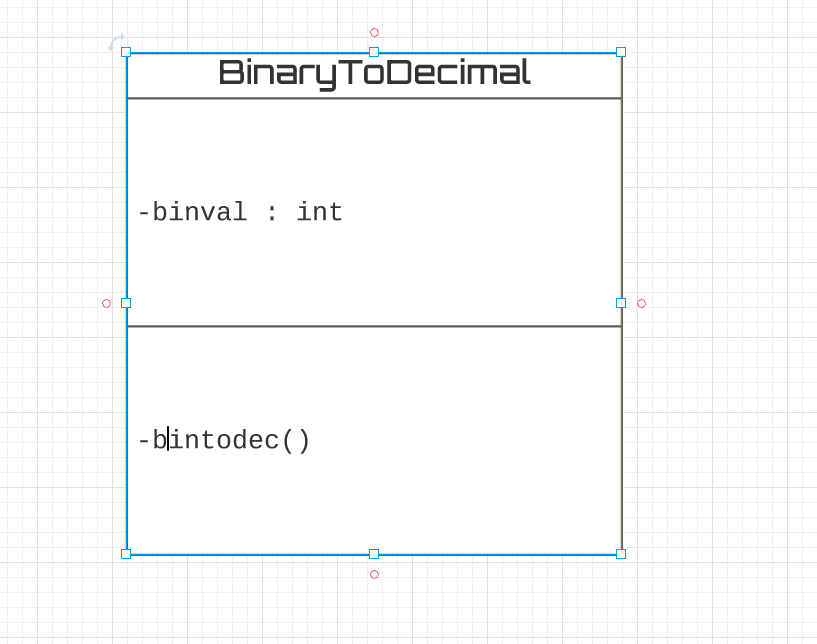
**USER STORY:**

BINARY TO DECIMAL CALCULATION:

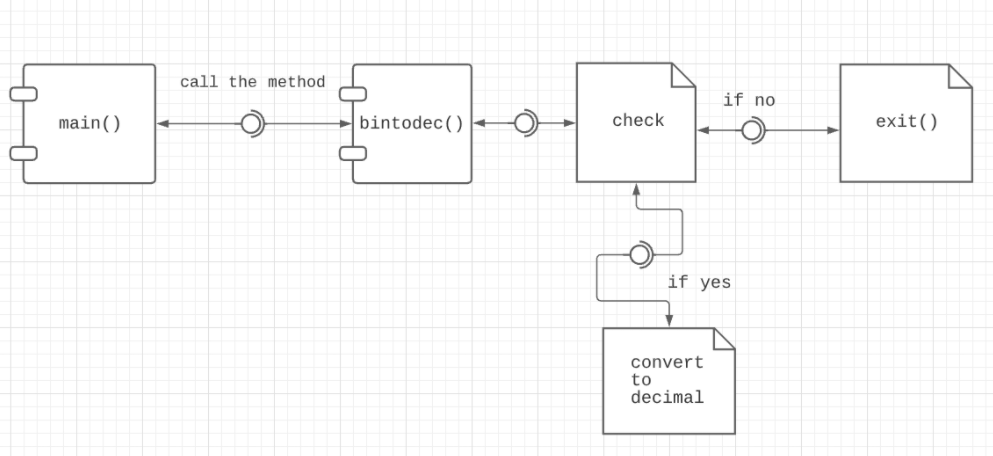
* Enter a number.
* Check if greater than zero.
* If not then print "Invalid Input".
* If yes then check if binary or not.
* If not then print "Invalid Input".
* If yes, then convert to decimal equivalent value.

**TABLE OF FIGURES:**

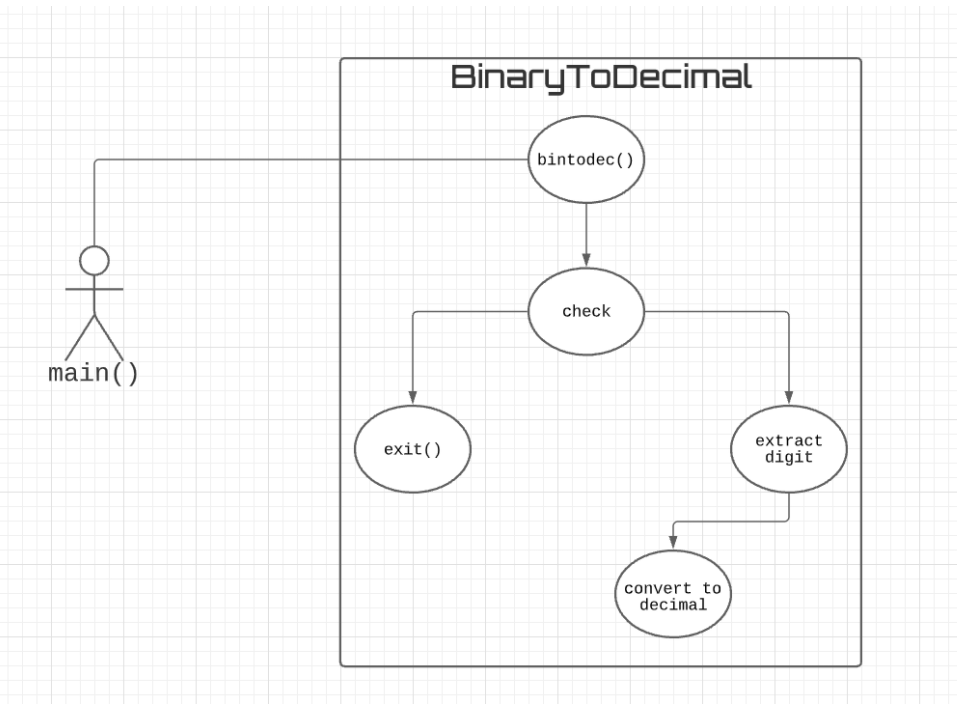
**CLASS DIAGRAM:**



**COMPONENT DIAGRAM:**



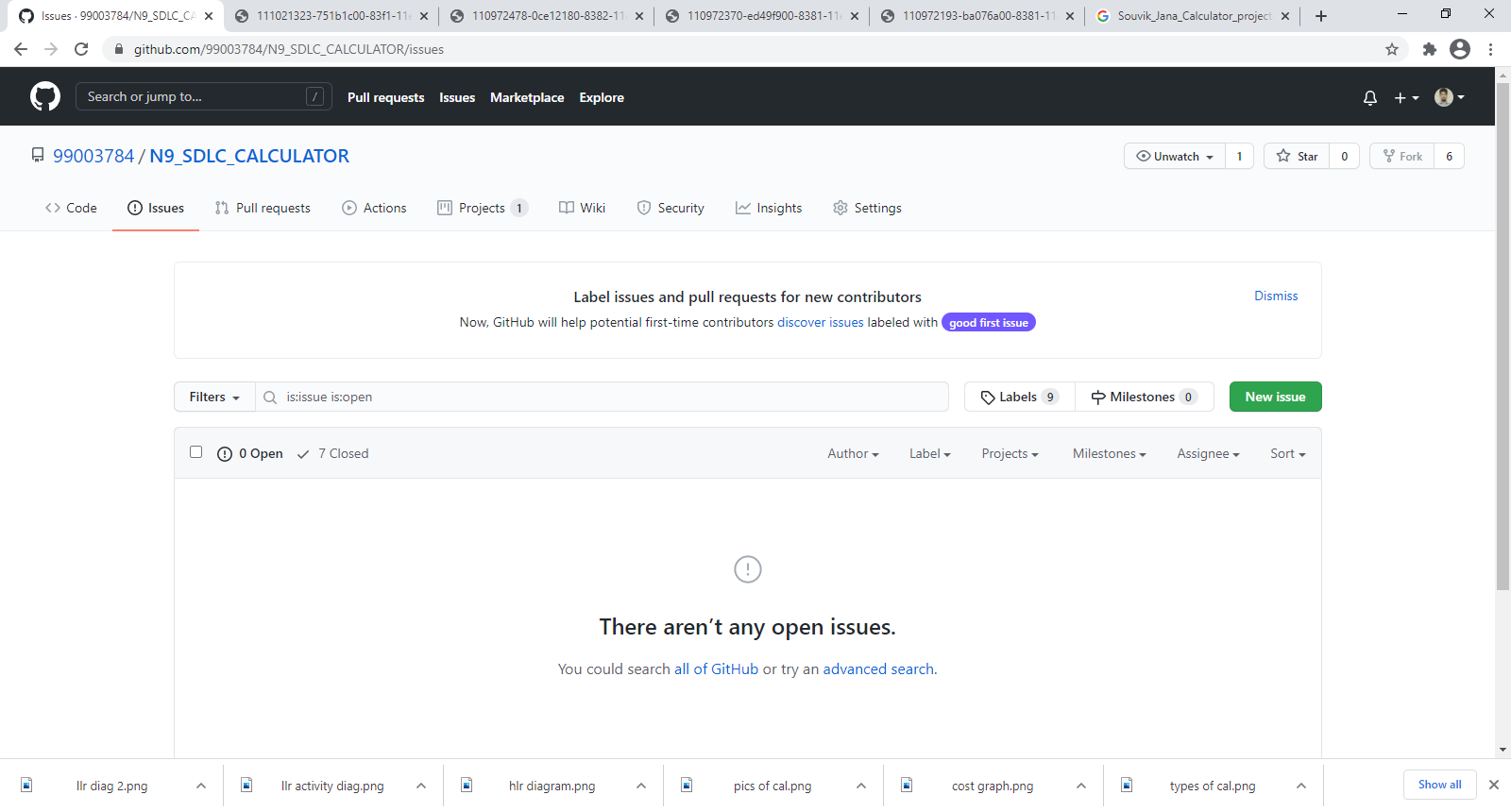
**USE CASE DIAGRAM:**

******

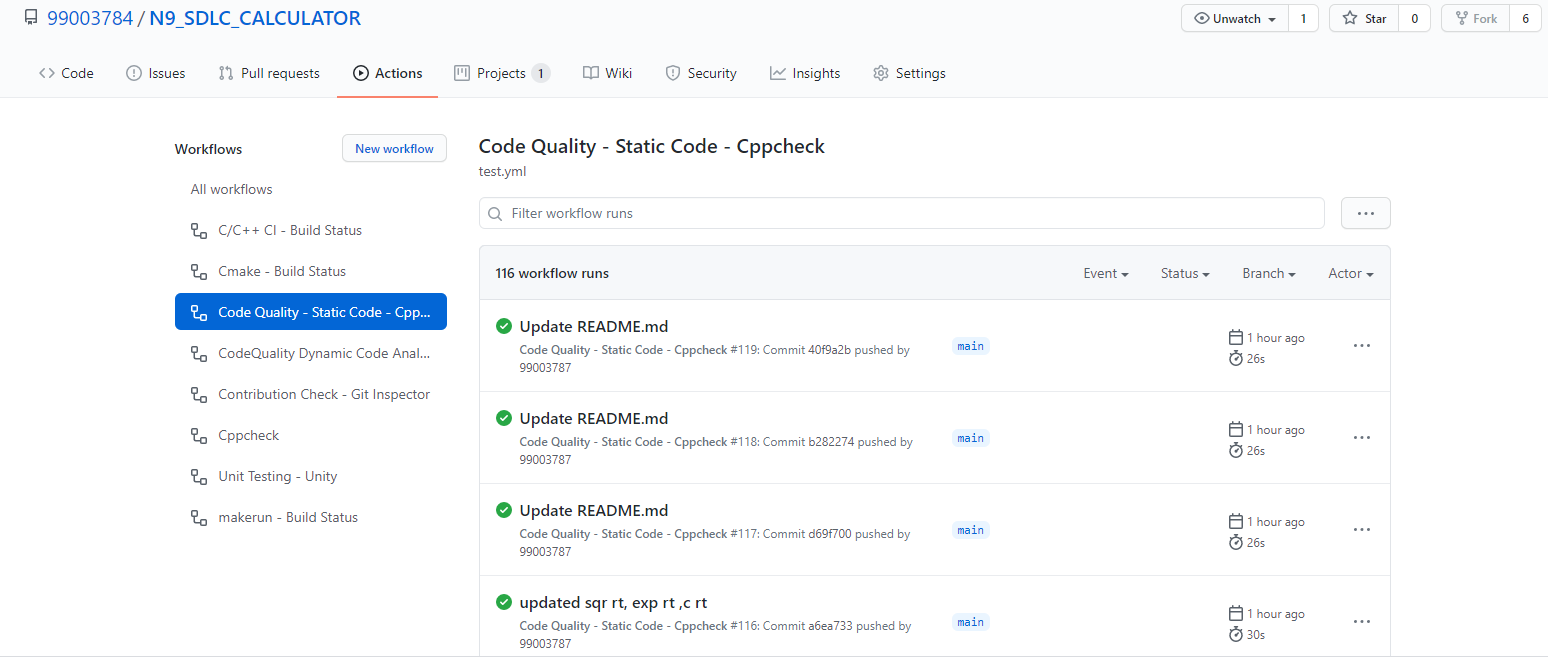
**GIT COMMITS:**

****

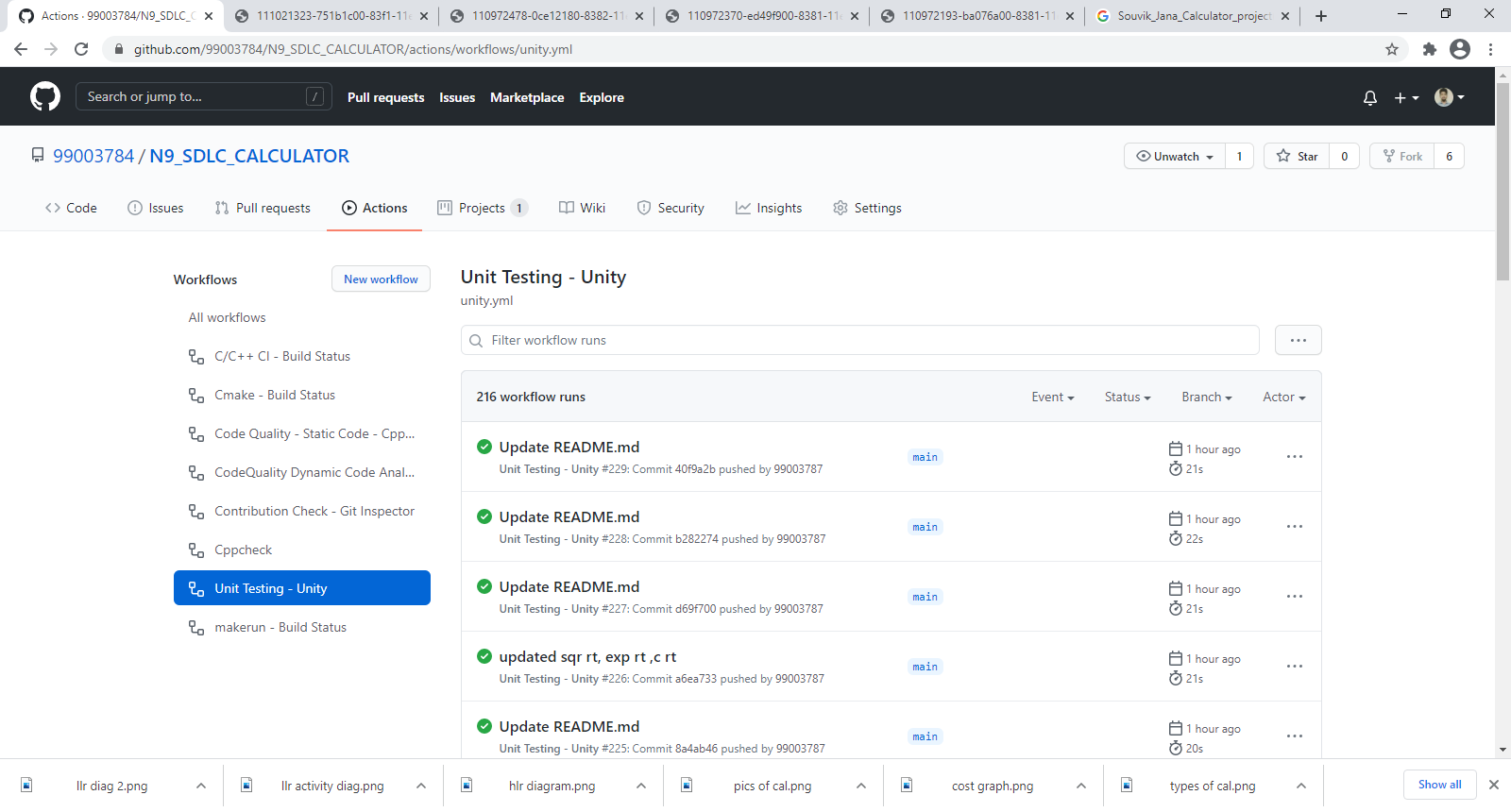
**GIT ISSUES:**

****

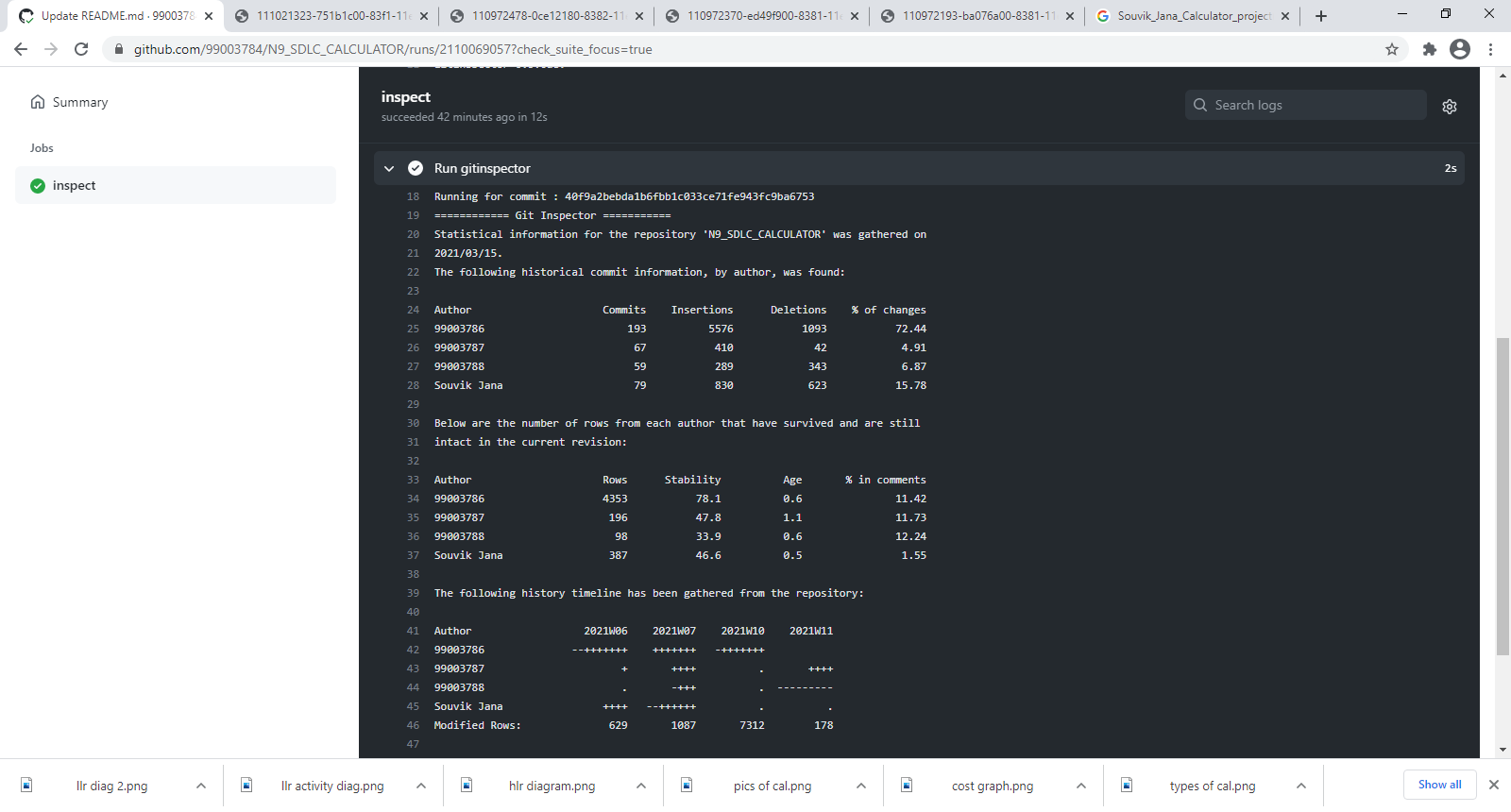
**GIT cpp check :**

****

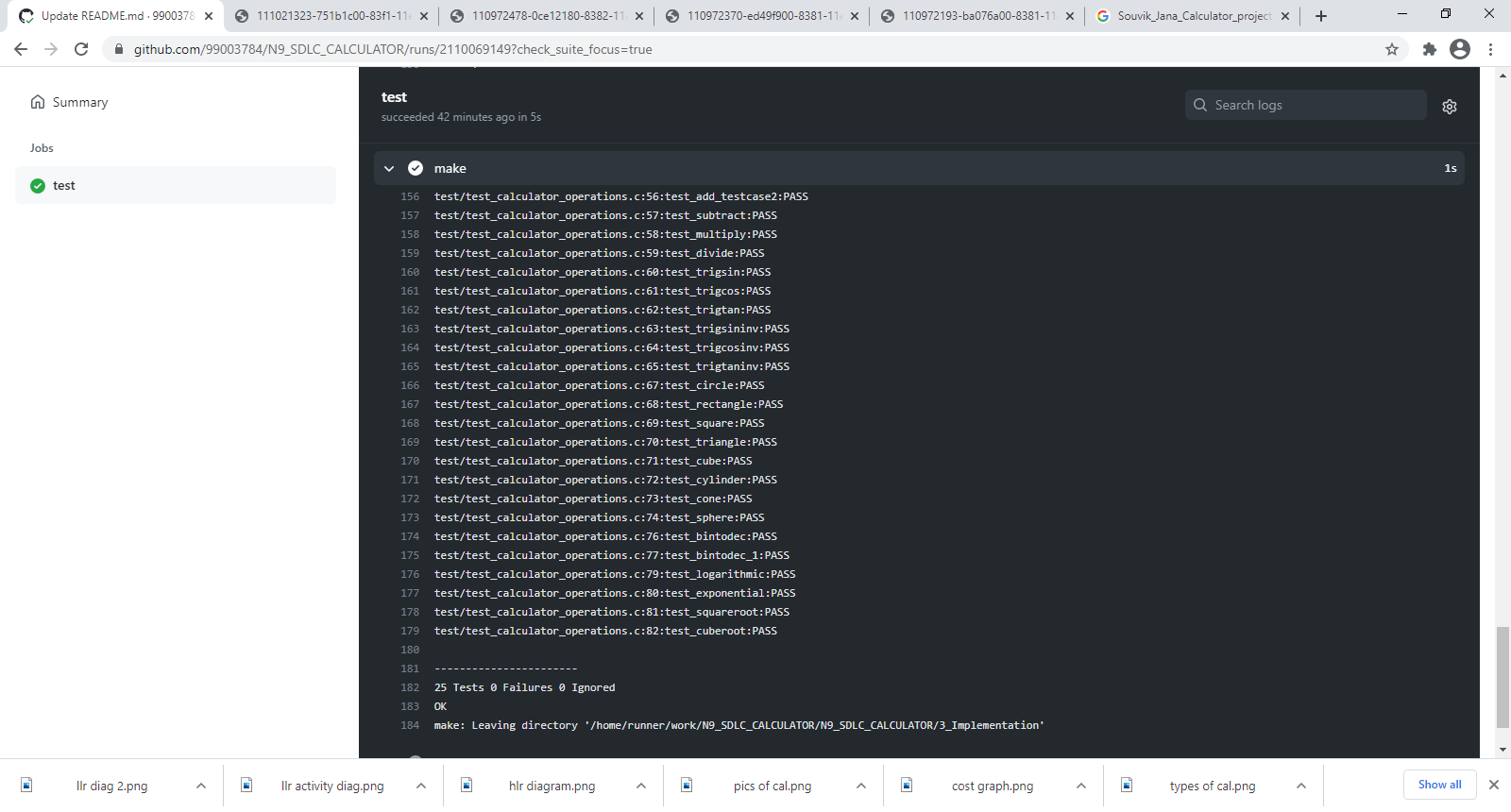
**GIT ACTIONS :**

****

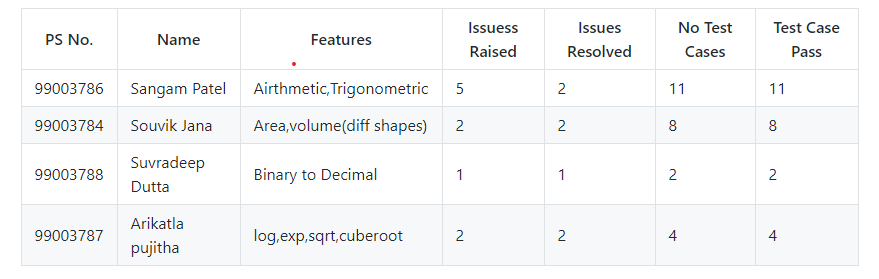
**GIT INSPECTOR:**

****

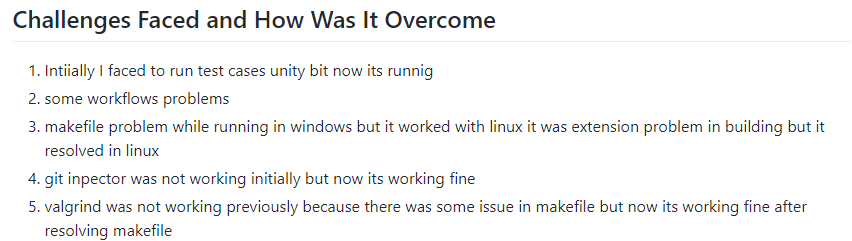
**TEST CASES :**

****

**CONTRIBUTION LIST :**

****

**CHALLENGES FACED AND HOW THEY WERE RESOLVED :**

****