./

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 | 15.02.2021 | SOURAV DEY | AMAN SRIVASTAVA | SRINIVAS  KAPALA  PRITHVI SEKHAR |  |
| 2 | 15.02.2021 | SOURAV DEY | KOPPARAPU JYOTHI SWAROOPA RANI | SRINIVAS  KAPALA  PRITHVI SEKHAR |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Document History**

Table of Contents

[Table of Figures 3](#_Toc53129062)

[Table of Tables 4](#_Toc53129063)

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

[**INTRODUCTION** 4](#_Toc53129065)

[**MY PRODUCT: “Name ”** 6](#_Toc53129066)

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

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

[**DESIGN** 8](#_Toc53129069)

[HIGH LEVEL DESIGN 8](#_Toc53129070)

[LOW LEVEL DESIGN 11](#_Toc53129071)

[**TEST PLANS** 13](#_Toc53129072)

[**REFERENCES** 15](#_Toc53129073)

[ACTIVITY 2: AGILE METHODOLOGY 15](#_Toc53129074)

[**THEME** 15](#_Toc53129075)

[**EPIC** 15](#_Toc53129076)

[**USER STORY** 16](#_Toc53129077)

[**REFERENCES** 17](#_Toc53129078)

[APPENDIX: 17](#_Toc53129079)

## Table of Figures

Figure 1 ACTIVITY DIAGRAM (HIGH LEVEL) ………………………………………………………………………………………………..….09

[Figure 2 USE CASE DIAGRAM (LOW LEVEL)…………………………………………………………………………………………………….1](#_Toc52177315)0

Figure 3 CLASS DIAGRAM (LOW LEVEL) …………………………………………………………………………………………………………11

Figure 4 TEST PALN………………………………………………………………………………………………………………………………………..12

[Figure 5 GIT](#_Toc52177325) 16

[Figure 6 GIT ISSUES](#_Toc52177326) 17

[Figure 7 GIT COMMITS 1 18](#_Toc52177327)

[Figure 8 GIT COMMIT 2 19](#_Toc52177328)

## Table of Tables

[Table 3 REQUIREMENTS (HLR AND LLR) 06](#_Toc52177306)

[Table 4 HIGH LEVEL TEST PLAN 15](#_Toc52177307)

[Table 5 LOW LEVEL TEST PLAN 16](#_Toc52177308)

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

**Product Definition**

Calculator is a small, portable electronic device used to perform basic and complex operations of arithmetic. The common components of calculator are keypad, LCD Display, Battery or solar cell.

Basic Calculators: These are the most common and basic calculators for simple arithmetic calculation like adding, subtracting, multiplication, division, percentage calculation. Some of the calculators have memory and check function as advanced feature. There are two types in Basic Calculator:

1.Pocket Calculator

2.Desktop Calculator

These calculators most commonly come in LCD display capacity of 8 digits to 12 digits. Basic Calculator priced from Rs. 100 to Rs. 400 depending on features and brands. Basic calculator generally used in Home and Office.

Scientific Calculators: These calculators can perform higher arithmetic and trigonometric calculations. It is generally used by higher education students and engineers. 240+ functions make it easy to perform all types of scientific calculations. Their range varies between Rs. 500 to Rs. 1000 depending on the features and brands.

Graphing Calculators: This is to perform graphical calculation by plotting a graph and has some functions of the scientific calculator. It has quiet a big display and resolution than the others. Their range varies between Rs. 10000 to Rs. 20000 according to features and different brands.

Financial Calculators: A financial calculator is a calculator that performs financial functions commonly needed in business. For many direct financial calculations, it has standalone keys that differentiate it from standard calculators. It may allow the user to program functions that the manufacturer has not provided by default. Their range varies from Rs. 3000 to Rs. 5000 in accordance to the brands and features.

Printing Calculators: These are like basic calculator that makes basic arithmetic calculation but has in-built printer to print the data and results in structured way. It is generally used in businesses where immediate print copy must be made. The rang varies from Rs. 1000 to Rs. 5000 according to the size, features and brands.

**SWOT Analysis**

Strengths:

* Dual Power source
* Pocket Friendly
* Memory storage
* Accuracy and speed
* Attractive

Weakness:

* Less number of complex functions.
* Display digits are limited.

Opportunity:

* Can be used by a wider age group.
* As education is an important aspect in India, it will have a greater market value.

Threats:

* Competitive market.
* Application of smartphones reduces the usage of calculators.
* Running ahead of the market.

**Requirements**

High Level Requirements:

|  |  |
| --- | --- |
| ID | DESCRIPTION |
| HLR\_1 | The designed calculator must provide accurate output of the given Arithmetic Operation:  Additions  Subtraction  Multiplication  Division  Modulus |
| HLR\_2 | The designed calculator must display the last five  Results stored. |

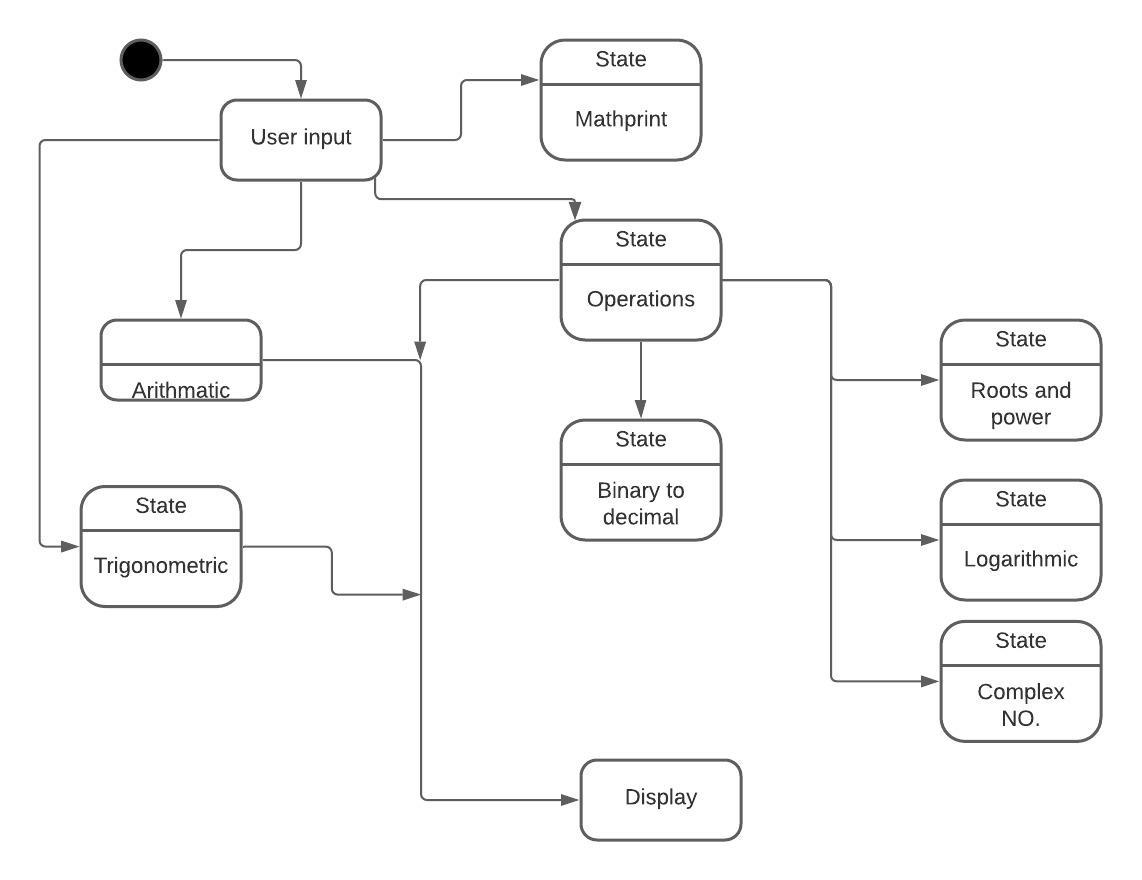
Low Level Requirements:

|  |  |
| --- | --- |
| ID | DESCRIPTION |
| LLR\_1 (Arithmetic Operations) | Addition:  Input validation: Check the ASCII value range of the user input numbers.  Input type: integer, float.  Operations: Take two inputs from the user and check the data type. If the inputs are in float data type; the results will be in floating point. If the inputs are in integer data type; the result will be in integer data type. If the inputs are in combination of integer as well as floating type; then the result should be in floating type.   |  | | --- | |  | |  |  | |
|  | Subtraction:  Input validation: Check the ASCII value range of the user input numbers.  Input type: integer, float.  Operation: Take two inputs from the user and check the data type. Sign of both the input values must be considered and accordingly the subtraction operation must be performed.  If the inputs are in float data type; the results will be in floating point. If the inputs are in integer data type; the result will be in integer data type. If the inputs are in combination of integer as well as floating type; then the result should be in floating type. |
|  | Multiplication:  Input validation: Check the ASCII value range of the user input numbers. Also check the sign of the user input numbers.  Input type: integer, float.  Operation: Take two inputs from the user and check the data type. Sign of both the input values must be considered and accordingly the multiplication operation must be performed.  If the inputs are in float data type; the results will be in floating point. If the inputs are in integer data type; the result will be in integer data type.   |  | | --- | |  |   If the inputs are in combination of integer as well as floating type; then the result should be in floating type. |
|  | Division:  Input Validation: Check the ASCII value range of the user input numbers. Also check the sign of the user input numbers. Divide by zero is not possible.  Input type: integer, float.  Operation: Take two inputs from the user and check the data type. Sign of both the input values must be considered and accordingly the division operation must be performed.  If the inputs are in float data type; the results will be in floating point. If the inputs are in integer data type; the result will be in integer data type.  If the inputs are in combination of integer as well as floating type; then the result should be in floating type. |
|  | Modulus:  Input Validation: Check the ASCII value range of the user input numbers. Also check the sign of the user input numbers. Divide by zero is not possible.  Input type: integer, float.  Operation: Take two inputs from the user and check the data type. Sign of both the input values must be considered and accordingly the division operation must be performed. |
| LLR\_2 (Memory Storage) | A history button is created which shows the last five stored results.  When the user hits the HISTORY button it will display the last five stored value.  Operation: Works with arrays. |

**Design**

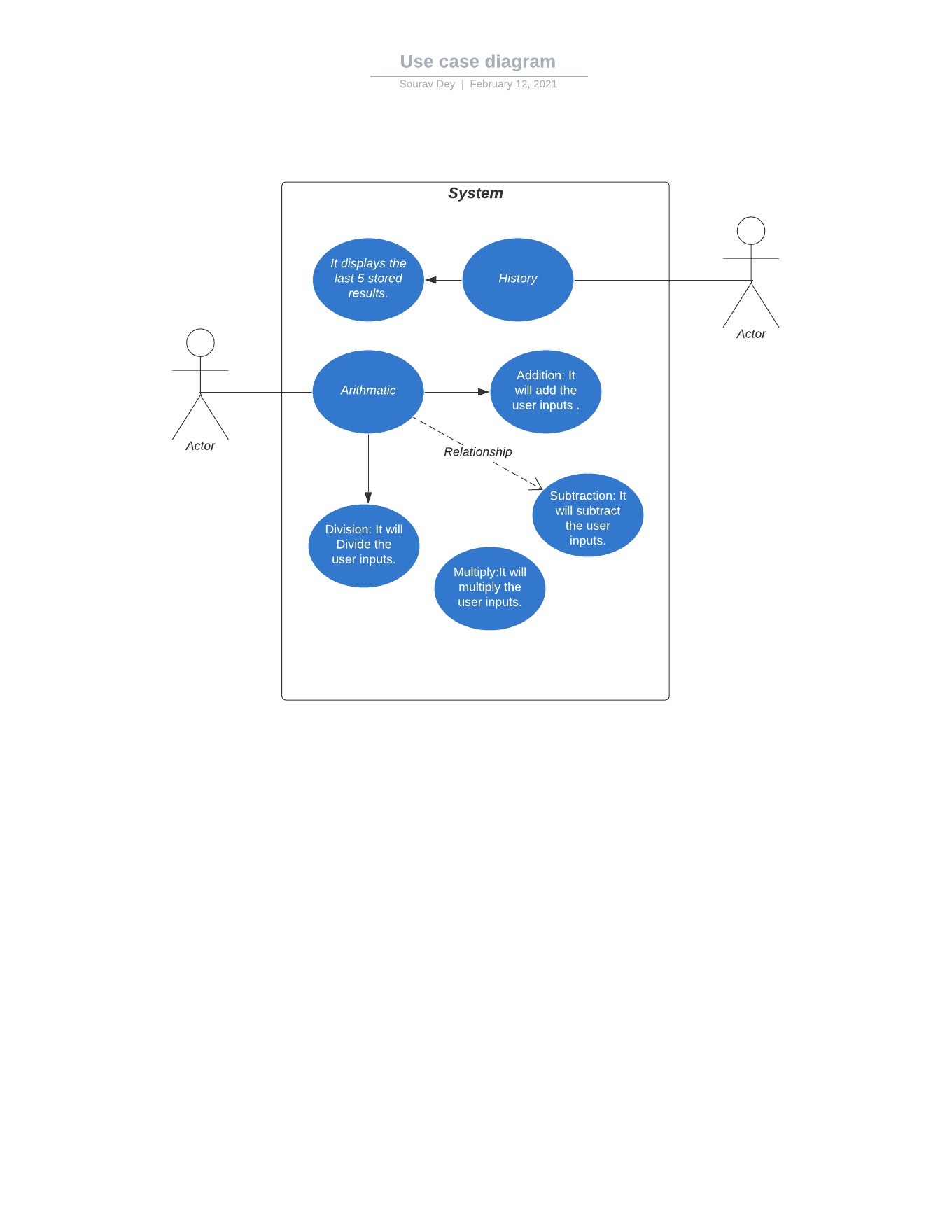
**HLR**

**Activity Diagram**

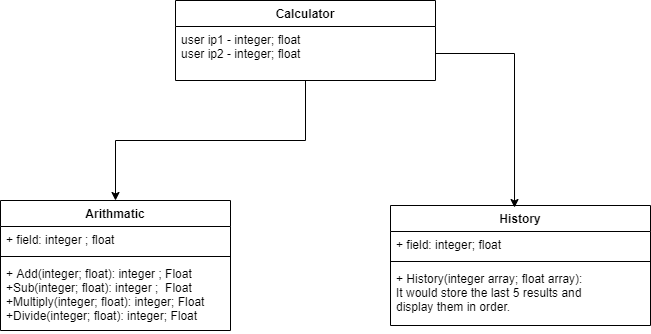


**LLR**

**Use case diagram**

****

**Class diagram**



**Test Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| Test\_id | Description | Expected input | Expected output |
| LLR\_1 \_ Arithmetic Operations | It contains all the basic arithmetic operations. |  |  |
|  | Addition:  1)The user input must be validated. The sign of the user input must also be validated. | Integer, integer | integer |
|  | 2) The floating-point input must provide a floating-point result. | Float, Float | Float |
|  | 3) A combination of floating point input and integer input must provide a floating-point output. | Float, integer  Or  Integer, float | Float |
|  | 4) If result exceeds by 14 digits then display unit must give the result as out of bound or out of range. | Input 1=10 digits  Input 2=6 digits  Or  Input 1=6 digits  Input 2=10 digits | Out of range  Or  Out of bound. |
|  | 5) If the first input is a negative number and second input is positive number or vice-versa.  If the negative input is greater than the positive input then the output must be negative. | Input1= -ve greater  Input2= +ve smaller  Or  Input1= +ve smaller  Input2= -ve greater | Negative  Negative |
|  | Subtraction:  1)The user input must be validated. The sign of the user input must be validated. | Integer or floating-point input.  Alphanumeric input | Pass  Error |
|  | 2) If both the input is of integer type or floating type then the output must be integer or floating type. | Integer, integer  Or  Float, float | Integer  float |
|  | 3)If both the input sign is negative then the output must be the additive of both the values. | Input1= -ve  Input2= -ve | Output=  -(input1+input2) |
|  | 4)If the result exceeds 14 digits then the display unit must show out of bound or out of range | Input1=more than 14 digits  Input2= more than 14 digits | Output= result out of bound. |
|  | Multiplication:  1) The user input must be validated. The sign of the user input must be validated. | Integer or floating-point input.  Alphanumeric input | Pass  Error |
|  | 2) If both the input is of integer type or floating type then the output must be integer or floating type. | Integer, integer  Or  Float, float | Integer  float |
|  | 3)If both the values are negative the output must have a positive sign.  If one input is positive and other one is negative then the resultant must have negative sign. | Input1=-ve  Input2=-ve  Input1=+ve  Input2=-ve | Output=+ve  Output=-ve |
|  | Division:  1) The user input must be validated. The sign of the user input must be validated. | Integer or floating-point input.  Alphanumeric input | Pass  Error |
|  | 2) If both the input is of integer type or floating type then the output must be integer or floating type. | Integer, integer  Or  Float, float | Integer  float |
|  | 3) If both the values are negative the output must have a positive sign.  If one input is positive and other one is negative then the resultant must have negative sign. | Input1=-ve  Input2=-ve  Input1=+ve  Input2=-ve | Output=+ve  Output=-ve |
|  | 4) If the denominator is zero then the display unit must show error.  If the numerator is zero it must display infinite. | Input1 = digit  Input2 = zero  Input1=zero  Input2= digit | Error  infinite |
| LLR\_6\_Memory\_Storage | 1)It must display the last five results when the user hits the history button. | History | Last five results |
|  | 2) The history operation starts storing the results from first after switching on the calculator. | OFF  ON  History | No result  All the results are removed. |

**Activity -2: Agile Methodology**

**Theme**

Calculator: Calculator is a small, portable electronic device used to perform basic and complex operations of arithmetic. The common components of calculator are keypad, LCD Display, Battery or solar cell.

**Epic**

1. Arithmetic Operations:

This feature does the basic five operations as given below:

* Addition
* Subtraction
* Multiplication
* Division
* Modulus

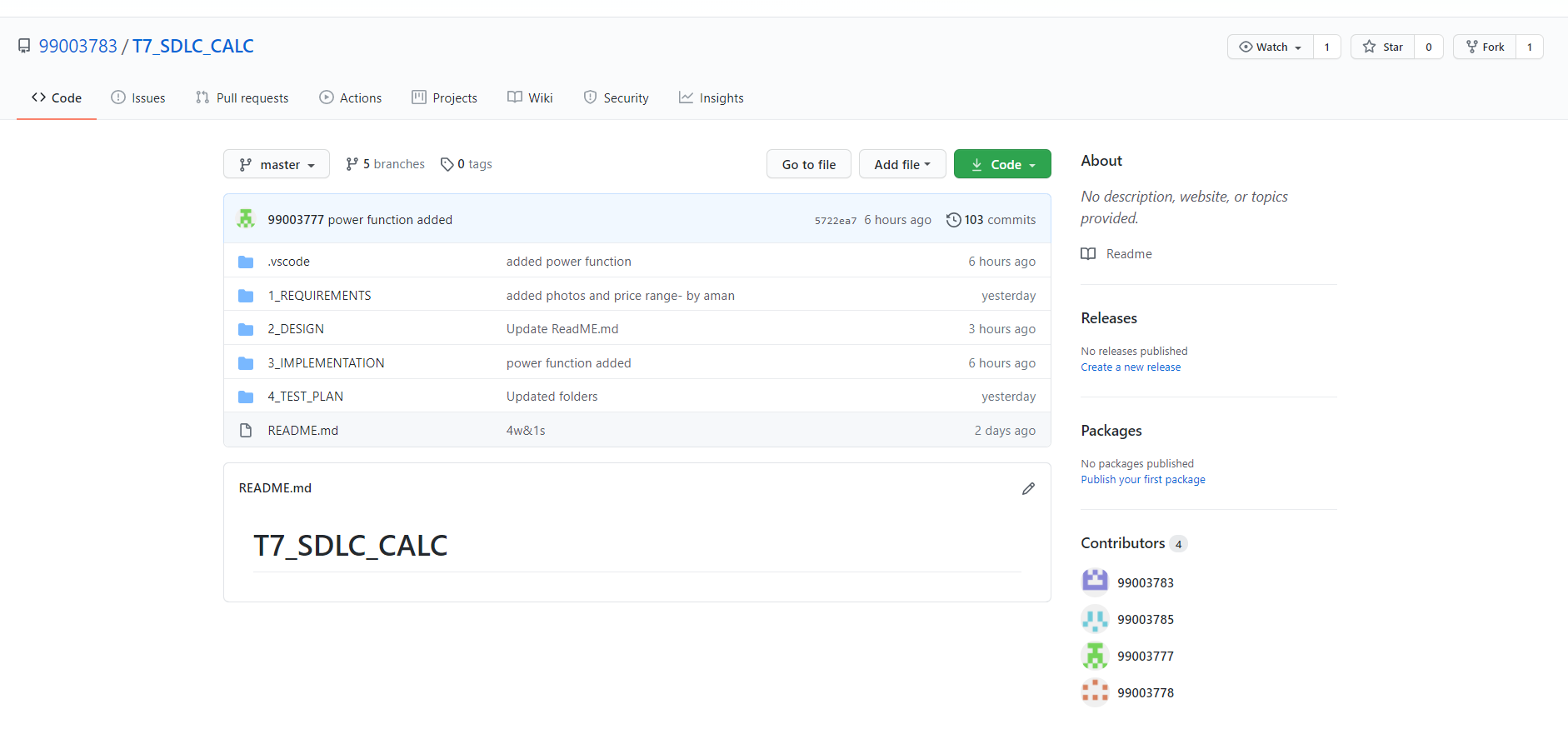
1. Memory Storage: This feature displays the last five stored results on the screen.

**User stories**

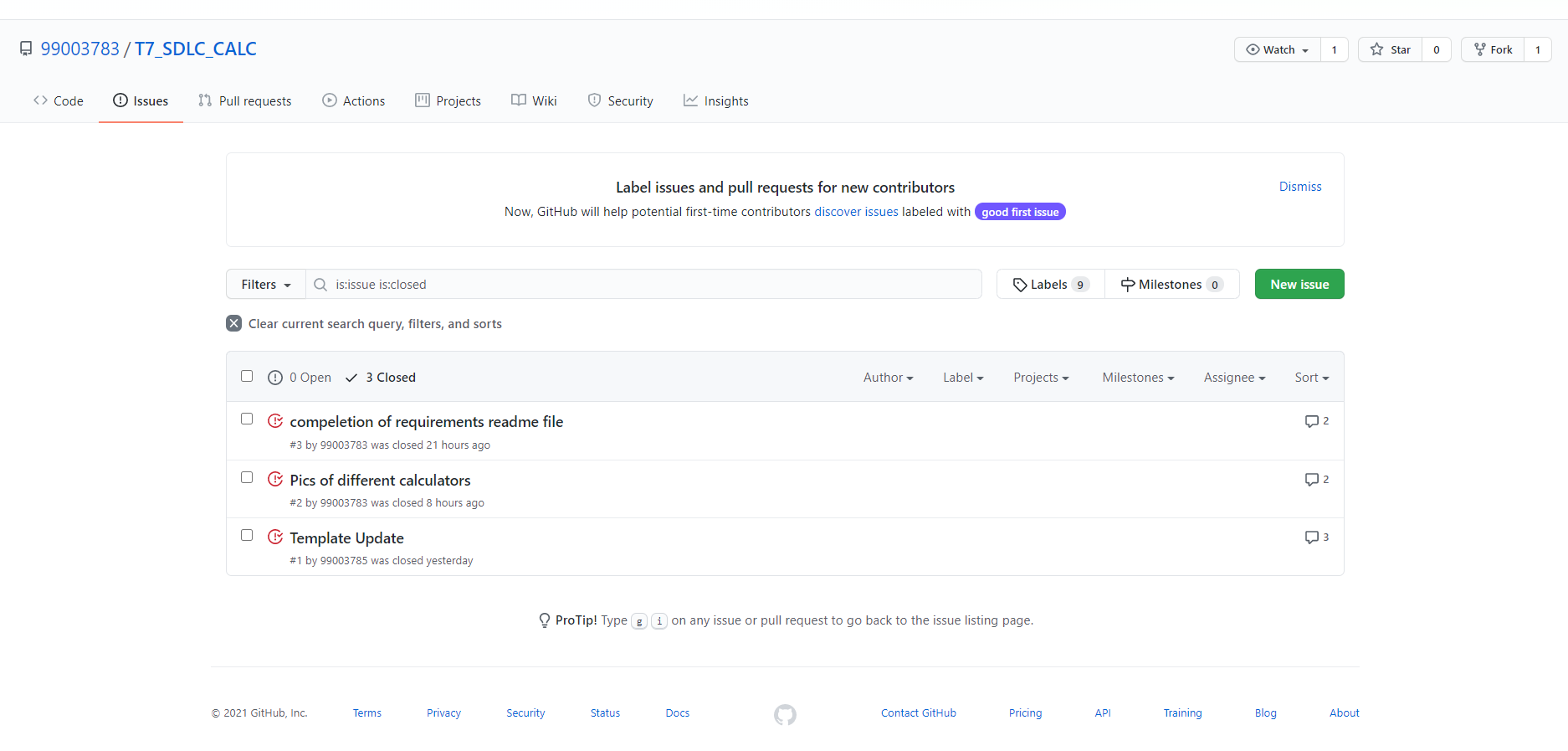
1. Accepting the user inputs.
2. Validating the user inputs according to the constrains like length of digits, ASCII value etc.
3. Validating the operations provided by the user like addition, subtraction, multiply, divide, modulus and memory storage.
4. Computing the operations i.e. implementation of code and giving the desired output.
5. Validating the result and displaying them on the screen.
6. Displaying of errors if present.

**Continuous Integration and Continuous Delivery**

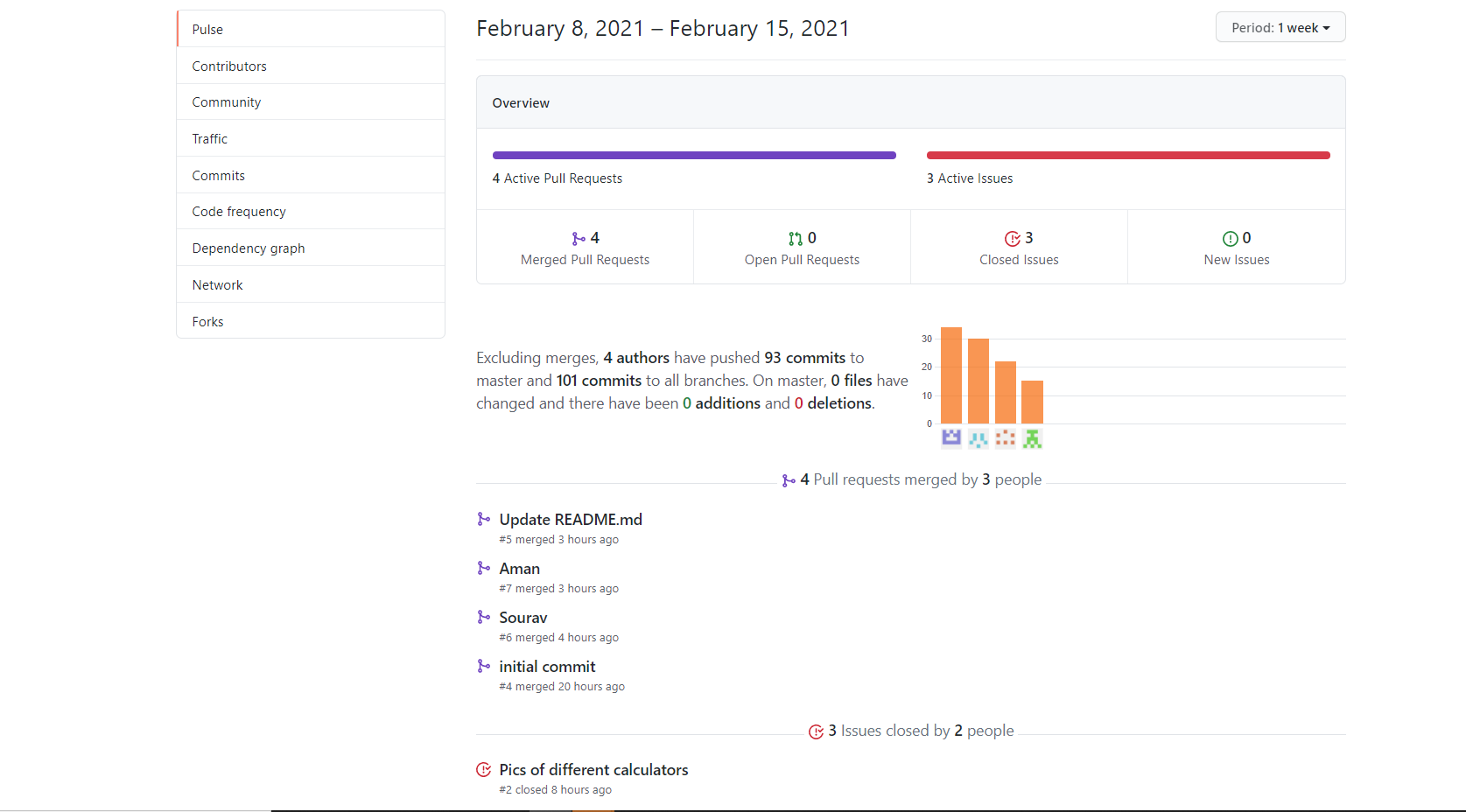
**Git:**

****

**Git Issues:**

****

**Git Commit:**

****

