./

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 | 15/02/21 | Himanshu Kumar/99003721 |  |  |  |
| 3 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Document History**

Table of Contents

[ACTIVITY 1: SYSTEM/ SOFTWARE DEVELOPMENT 3](#_Toc53129064)

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

[**MY PRODUCT: “Name ”** 5](#_Toc53129066)

[**SWOT ANALYSIS** 6](#_Toc53129067)

[**REQUIREMENTS** 6](#_Toc53129068)

[**DESIGN** 7](#_Toc53129069)

[HIGH LEVEL DESIGN 8](#_Toc53129070)

[LOW LEVEL DESIGN 9](#_Toc53129071)

[**TEST PLANS** 10](#_Toc53129072)

[**REFERENCES** 10](#_Toc53129073)

[ACTIVITY 2: AGILE METHODOLOGY 11](#_Toc53129074)

[**THEME** 11](#_Toc53129075)

[**EPIC** 12](#_Toc53129076)

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

[**REFERENCES** 13](#_Toc53129078)

[APPENDIX: 14](#_Toc53129079)

GitHub Repo Link - <https://github.com/99003721/AppliedSDLC_Calculator_N1>

**INTRODUCTION**

This is an attempt to build the software development cycle of a semi-engineering calculator. It includes the numerous high-level and low-level requirements of the various features involved in semi-engineering such as log, antilog, exponents roots and polynomial functions. It then explores about the various test cases that should be taken in consideration such that all requirements are taken care of. In the end, we build the codes that follow all the requirements and design efficiently as per them.

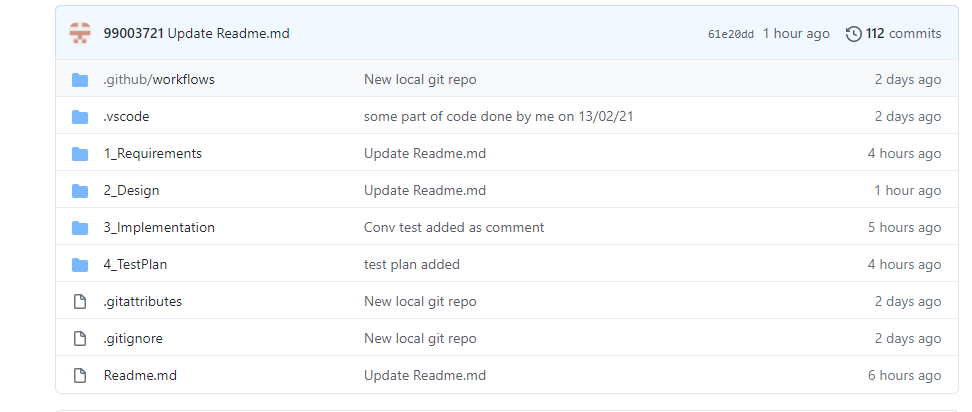
**MY PRODUCT: “Name ”**

**Semi-engineering Calculator**

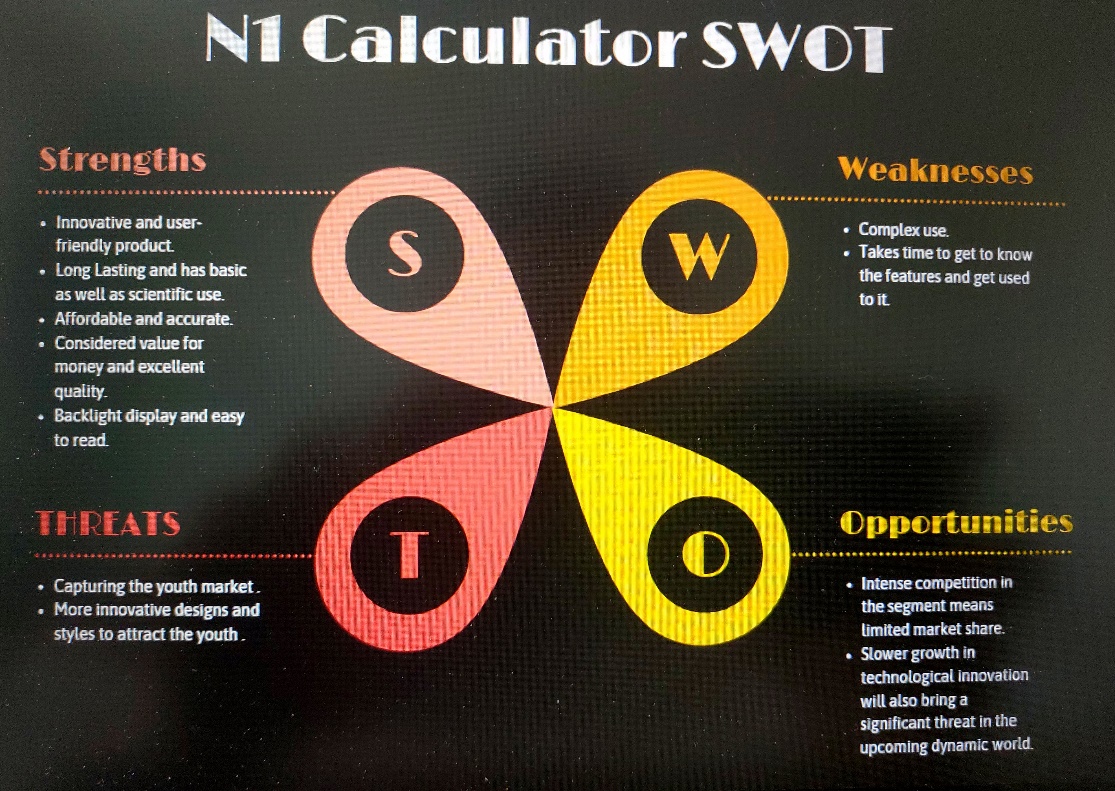
**Git Resource**

****

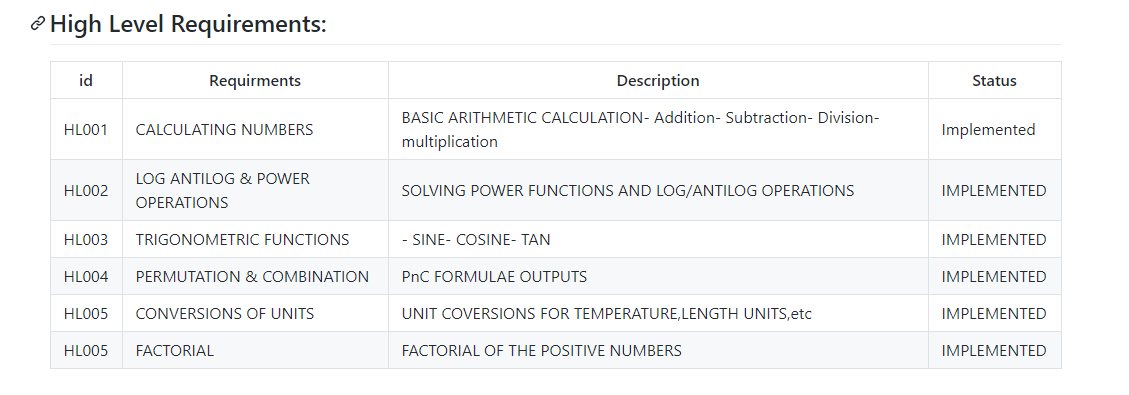
****

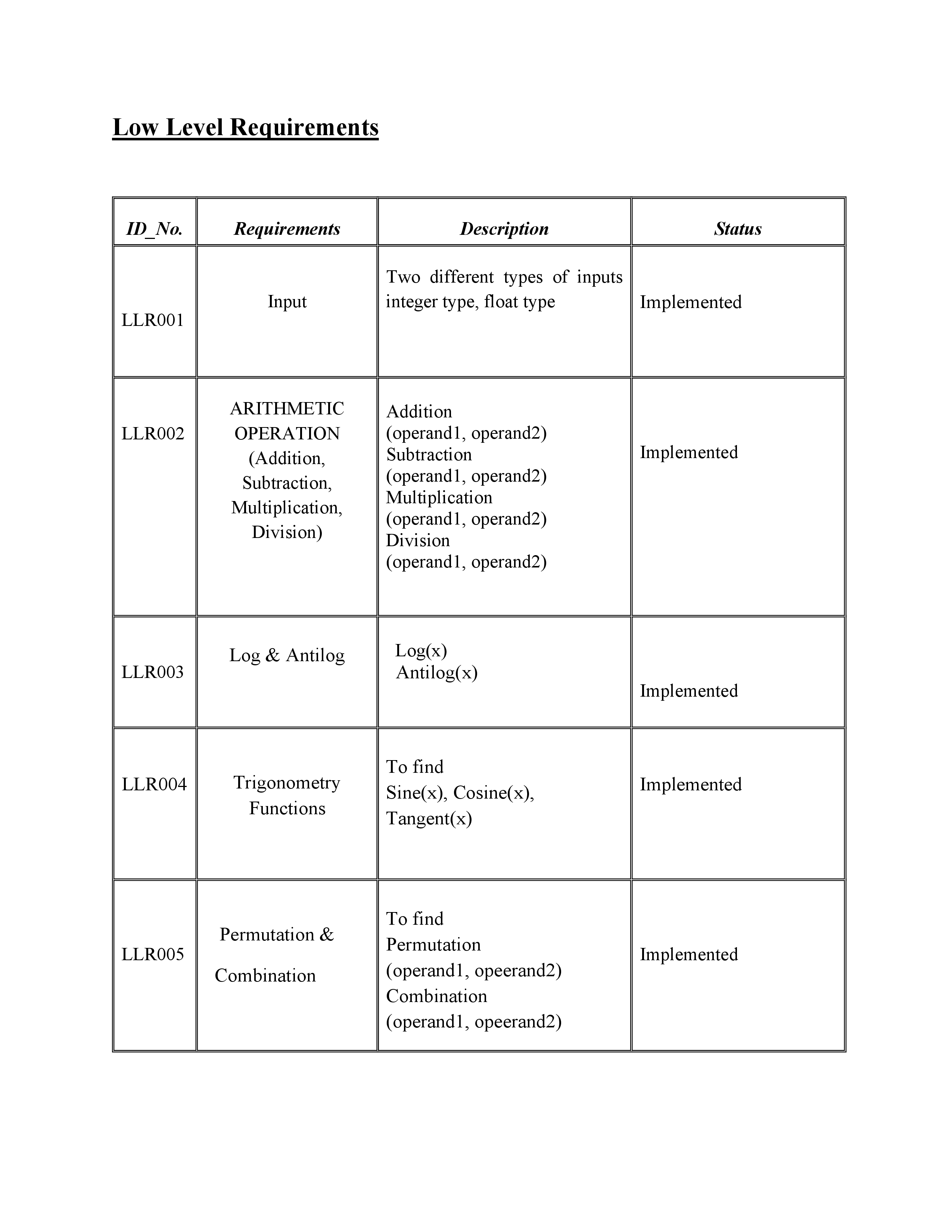
****

**SWOT ANALYSIS**

****

**REQUIREMENTS**

**HLR –**

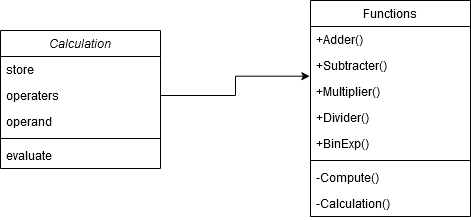
**LLR –**

**DESIGN –**

**HIGH LEVEL DESIGN –**

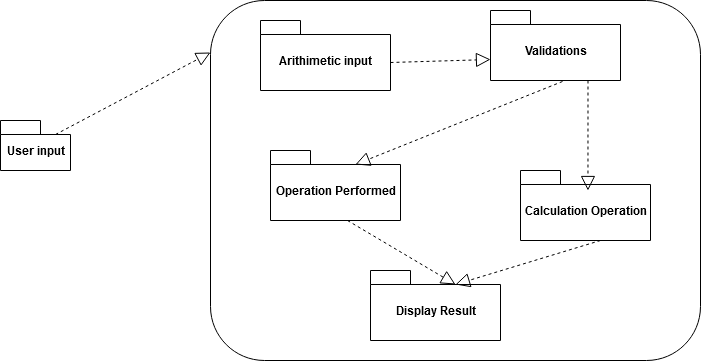
**Figure 1 –**

**CLASS DIAGRAM(HIGH LEVEL)**

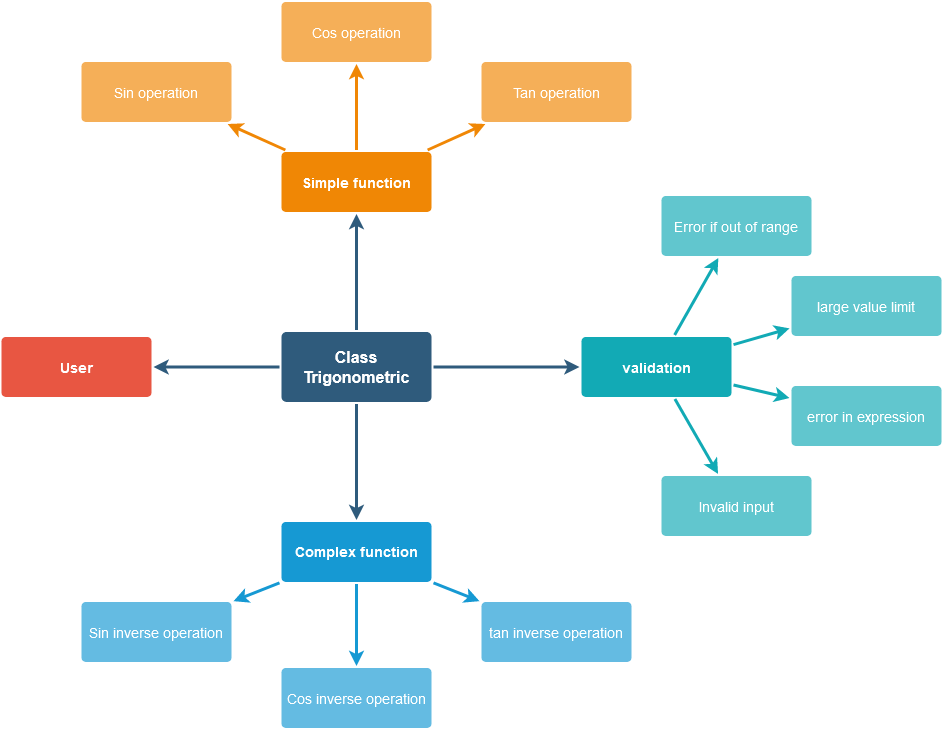
****

**LOW LEVEL DESIGN**

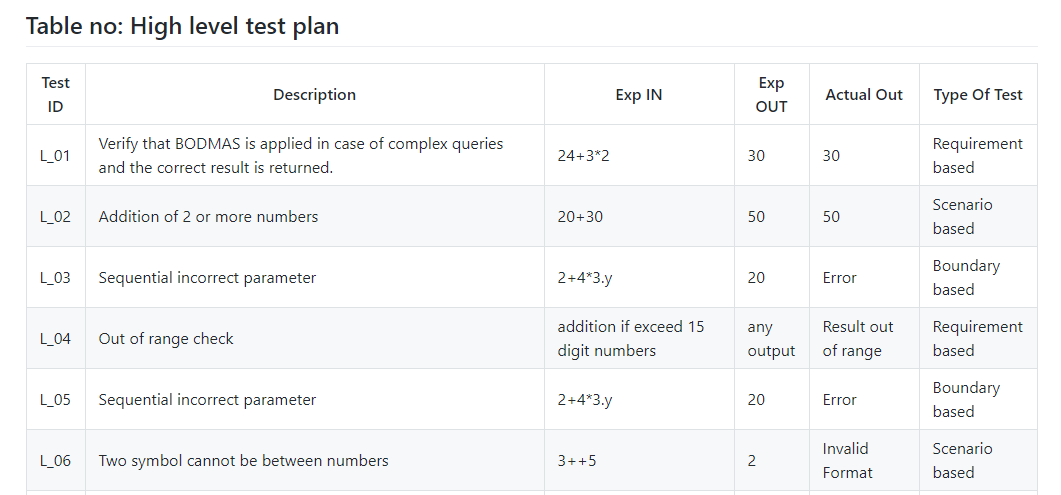
Package DIAGRAM (HIGH LEVEL)

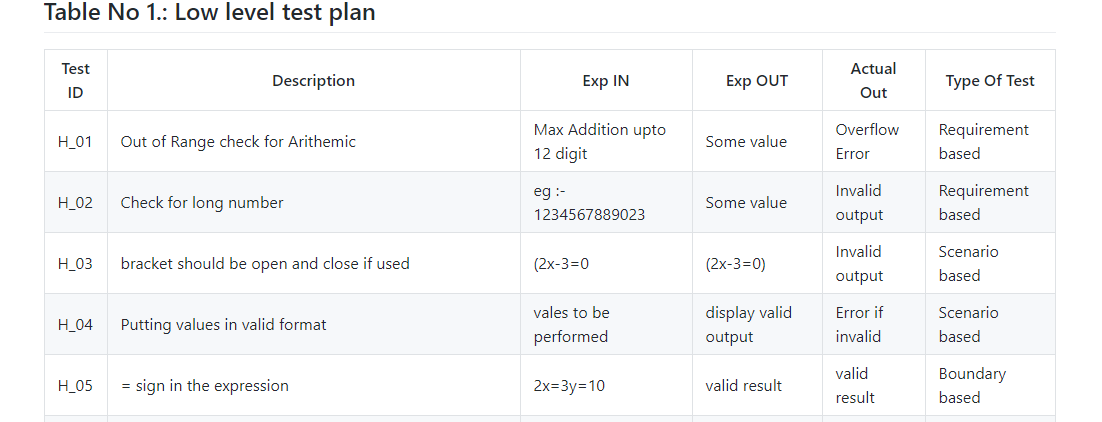
****

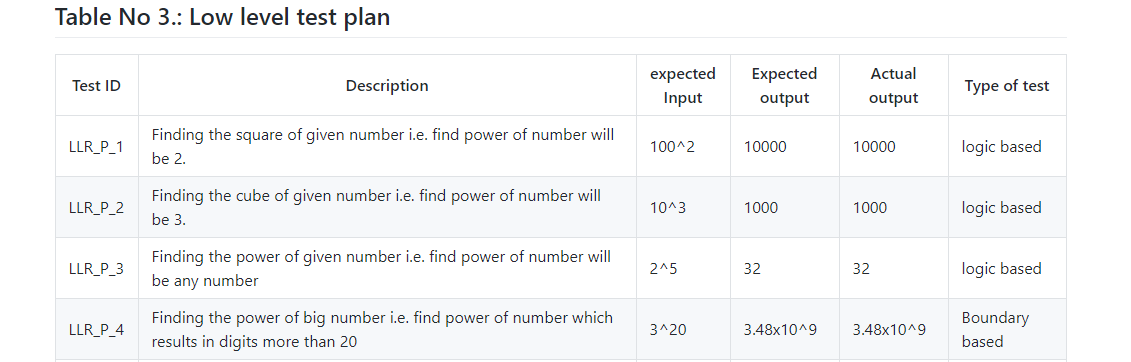
COMMUNICATION DIAGRAM (LOW LEVEL)



**TEST PLANS –**

****

****

****

**REFERENCES –**

**UML Diagram -** <http://draw.io/>

<https://www.javatpoint.com/software-development-life-cycle>

<https://www.tutorialspoint.com/cprogramming/index.htm>

GitHub Repo Link - <https://github.com/99003721/AppliedSDLC_Calculator_N1>

**AGILE METHODOLOGY**

**THEME –**

We have majorly focus to provide a efficient and simple Semi-engineering calculator for the students. It explore most such basic function in requirements. Our theme is to made a quick and friendly operational calculator. The range we have chosen is for medium range cost and features. Its similar to the scientific calculator and mostly used in calculation.

**EPIC –**

* Factorial
* Cube Root
* Exponential

**USER STORY –**

**Exponential** –

* Power should be valid input
* Exponential F(x) =a(b^x) Where b is positive real no. & b not equal to 1

**Factorial** –

* Finding the factorial of numbers
* Input\_number! Means n(n-1)(n-2)….3,2,1.
* First check the given number is positive. If the given integer is a negative the output is not defined
* Submit integer value at lower boundary [0]

**Cube root** –

* Finding the cube of given number i.e. find power of number will be 3
* Finding the power of given number i.e. find power of number will be any number
* Finding the power of big number i.e. find power of number which results in digits more than 20

**REFERENCES –**

<https://www.programiz.com/>

<https://www.javatpoint.com/c-programming-language-tutorial>

<https://creately.com/lp/uml-diagram-tool/>

<https://www.lucidchart.com/pages/examples/uml_diagram_tool>

GitHub Repo Link - <https://github.com/99003721/AppliedSDLC_Calculator_N1>