./

GENESIS - Learning Outcome & Mini-project Summary Report



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **To be Approved** | **Remarks/Revision Details** |
| 1. |  | Nitin N Shetty |  |  |  |
|  |  |  |  |  |  |

**Details**

Contents

[Contents 3](#_Toc69361300)

[Miniproject -1 [Team] 4](#_Toc69361301)

[Modules used 4](#_Toc69361302)

[Topic and Subtopics 4](#_Toc69361303)

[Objectives & Requirements 5](#_Toc69361304)

[Low Level Requirement 6](#_Toc69361305)

[Complex Number 7](#_Toc69361306)

[9](#_Toc69361307)

[Design 11](#_Toc69361308)

[Test Plan 26](#_Toc69361309)

[Implementation Summary 28](#_Toc69361310)

[Video Summary 28](#_Toc69361311)

[Git Link 28](#_Toc69361312)

[https://github.com/99003738/AppliedSDLC\_N3.git 28](#_Toc69361313)

[Git Dashboard 28](#_Toc69361314)

[Summary 29](#_Toc69361315)

[Summary 35](#_Toc69361316)

[Challenges faced and how were they overcome 35](#_Toc69361317)

[Miniproject -2 [Individual] 37](#_Toc69361318)

[Module:- Python 37](#_Toc69361319)

[Topic and Subtopics 37](#_Toc69361320)

[Objectives & Requirements 37](#_Toc69361321)

[Design 38](#_Toc69361322)

[Low Level Design 39](#_Toc69361323)

[Implementation Summary 40](#_Toc69361324)

[Git Link 40](#_Toc69361325)

[Git Dashboard 40](#_Toc69361326)

[Summarys 40](#_Toc69361327)

[Individual Contribution & Highlights 40](#_Toc69361328)

[Summary 40](#_Toc69361329)

[Challenges faced and how were they overcome 40](#_Toc69361330)

# Miniproject -1 [Team]

## Modules used

SDLC (Software Development Life Cycle) and C Programming Modules are used in the project.

Project Title

Mini Calculator

### Topic and Subtopics

* Introduction about SDLC
* C Programming
* Git Hub
* Code Analysis

Valgrind

CPP check

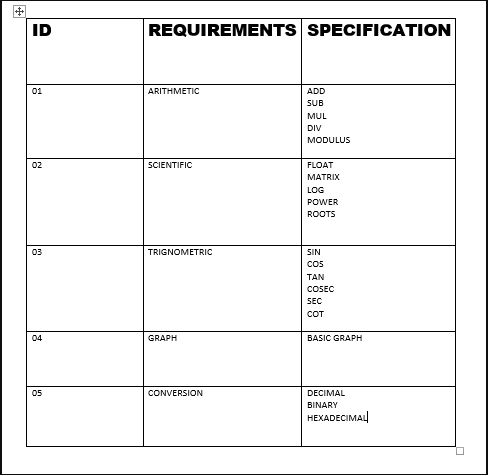
* Testing
* Unity Testing
* Features of Calculator
* Core Steps of SDLC
* Testing has been done for each function
* Makefile
* V Model
* Agile Model

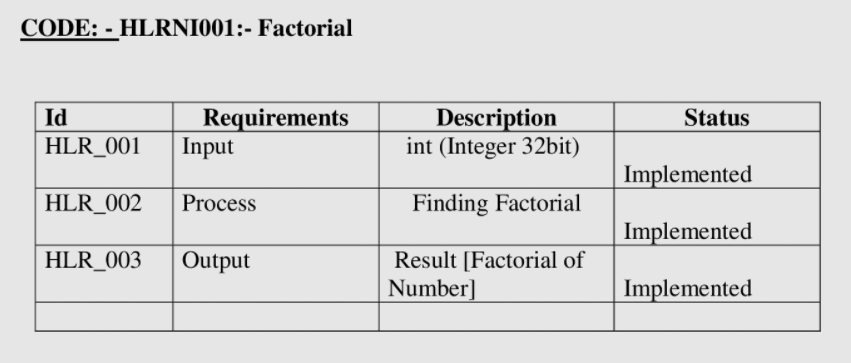
## Objectives & Requirements

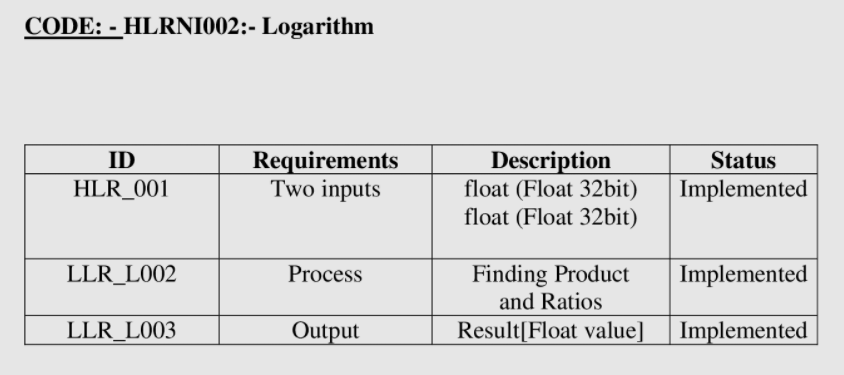
Objective: -  The simplest calculators can do only addition, subtraction, multiplication, and division. More sophisticated calculators can handle exponential operations, roots, logarithms, trigonometric functions, and hyperbolic functions So we are designing a calculator which contains all the functionalities

Of simplest calculator and sophisticated calculator.

High Level Requirement

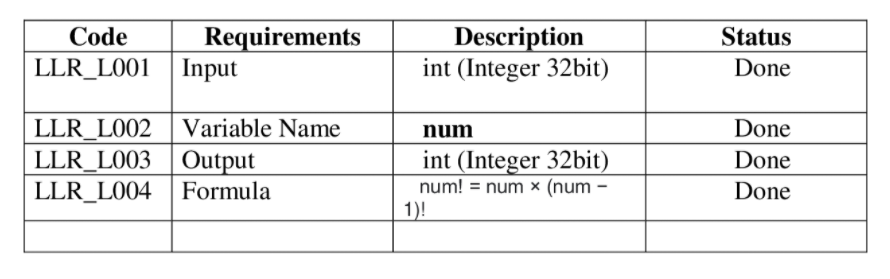




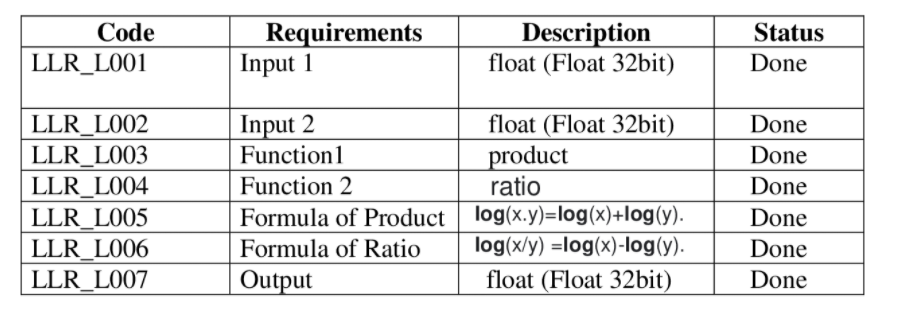


## Low Level Requirement

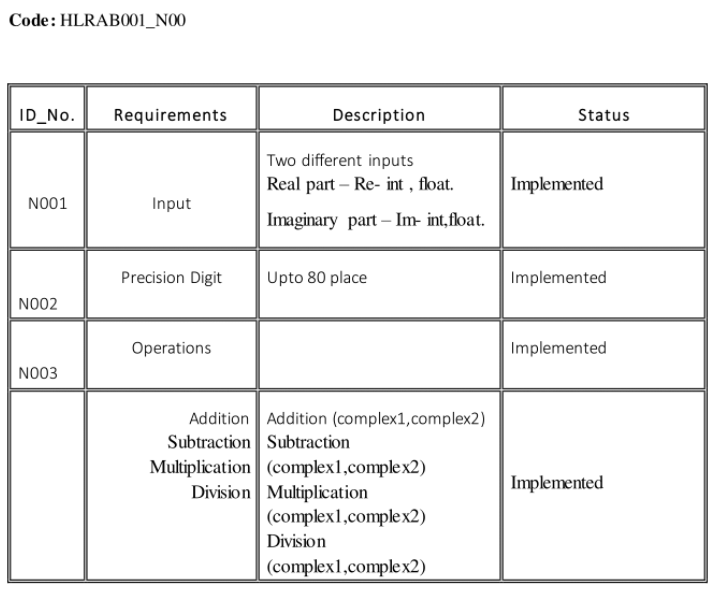
Factorial



Logarithm

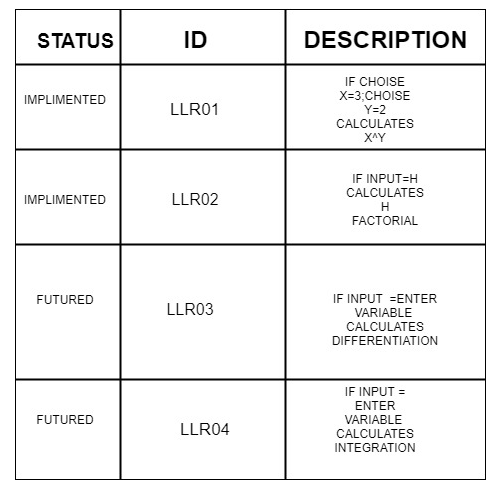


## Complex Number

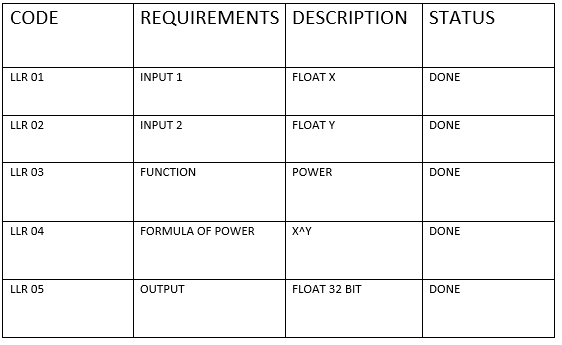


Arithmetic Operation

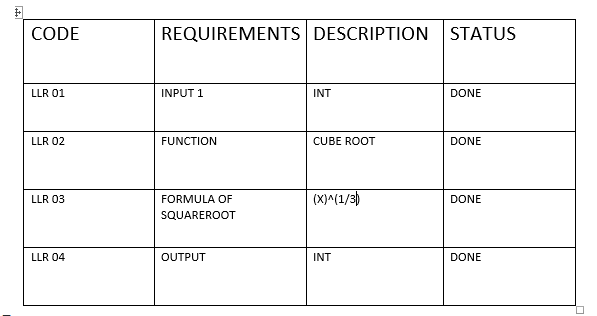
## 



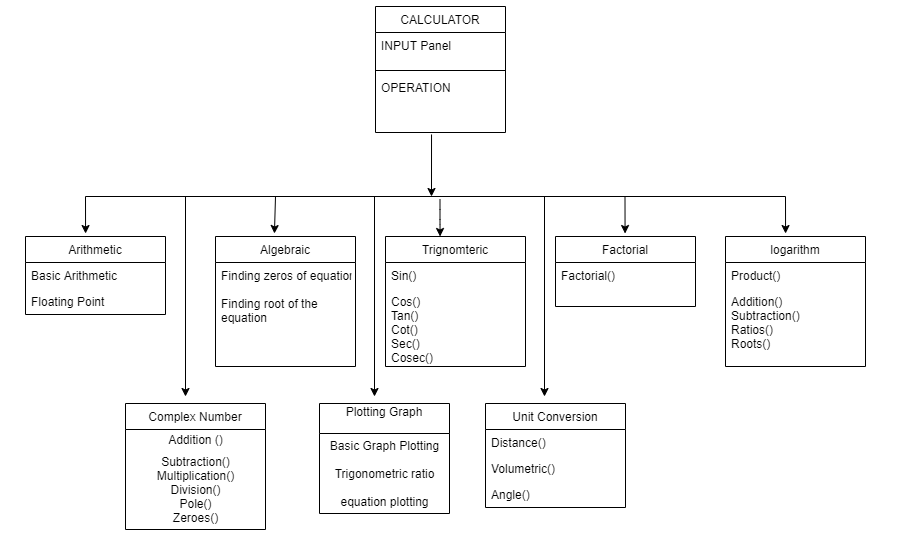
Square Root



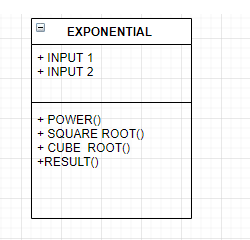
Cube Root



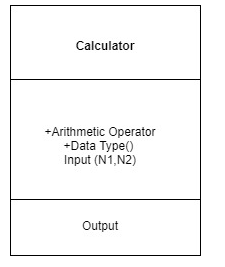
Structural Diagram of Calculator



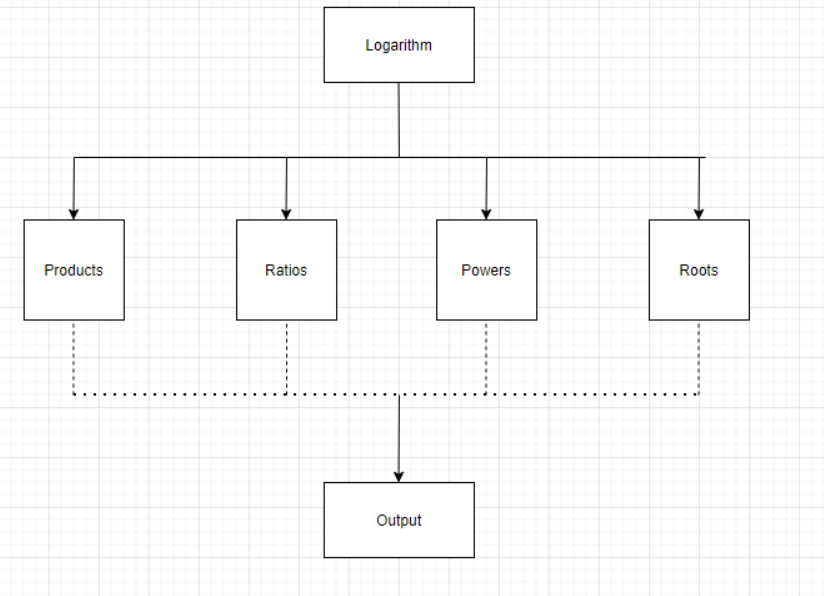
LLR Diagram(Exponential)



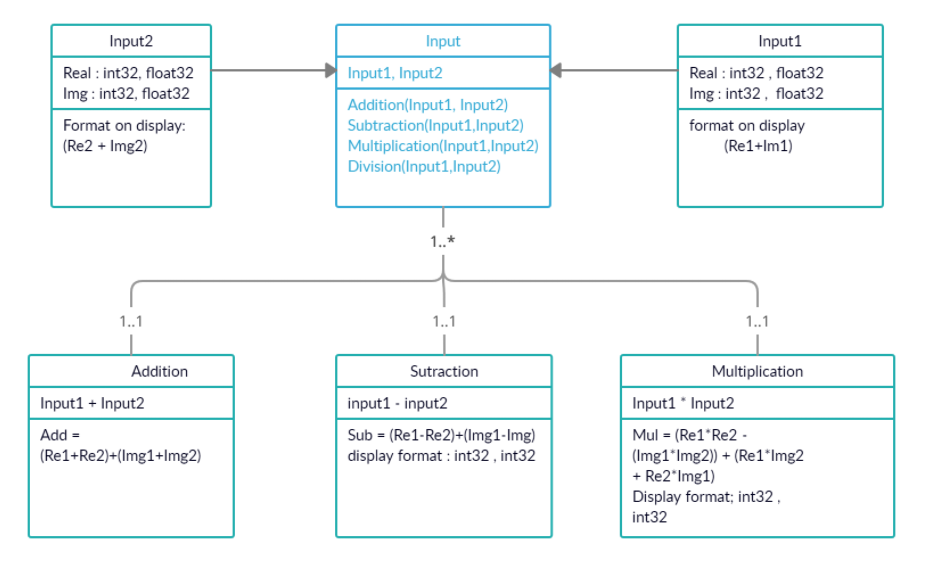
LLR Diagram (Calculator)



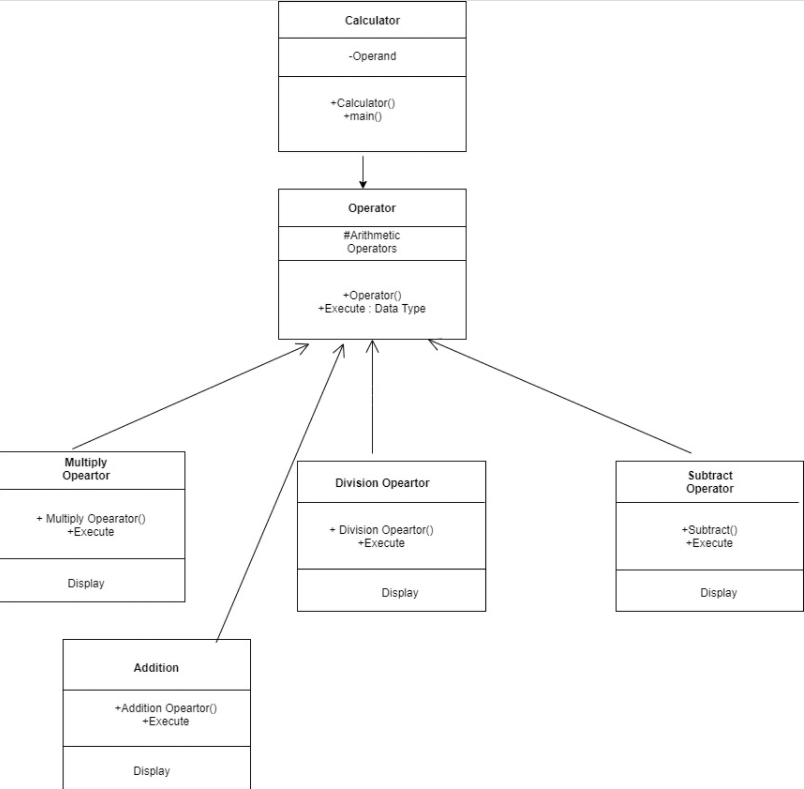
HLR Diagram(Logarithm)



LLR Diagram (Complex Number)

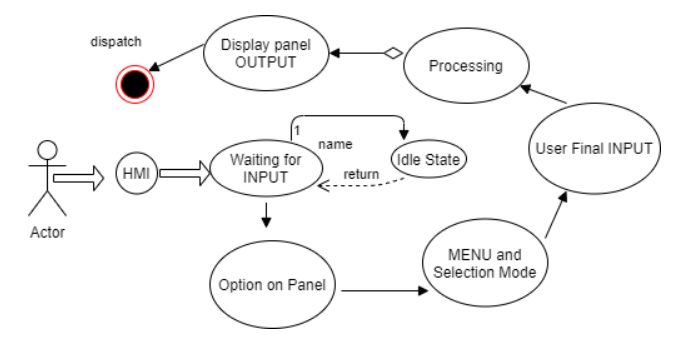


LLR Diagram(Arithmetic Operations)



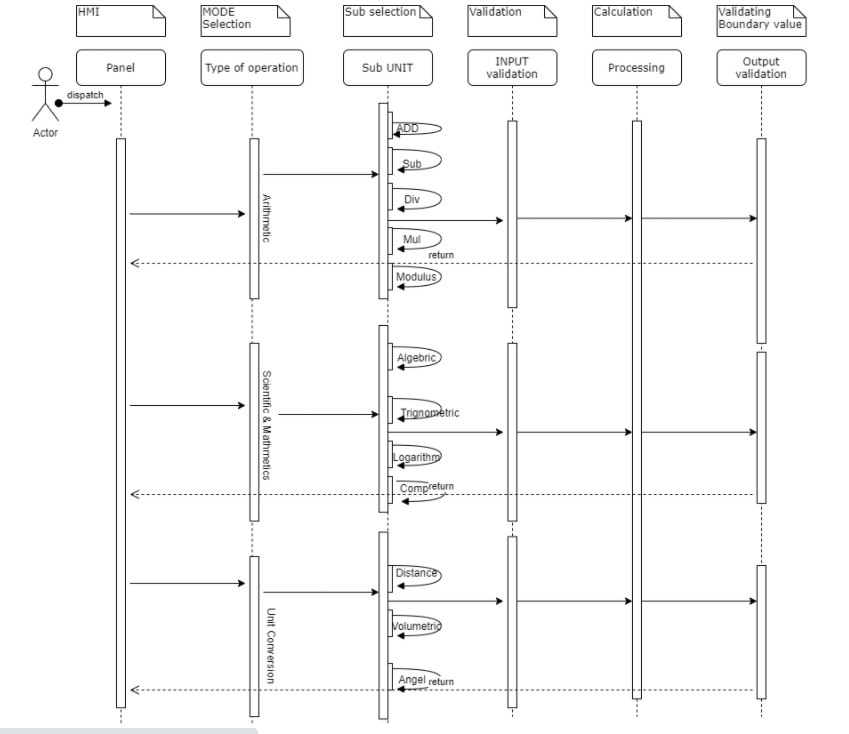
Behavioral Diagram

HLR Calculator Diagram (Flow control)

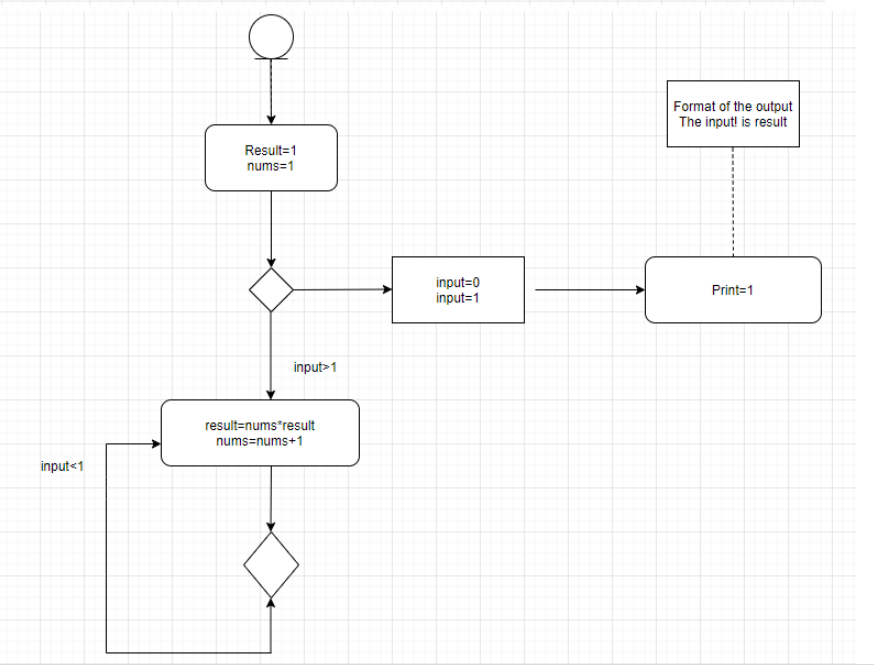


L

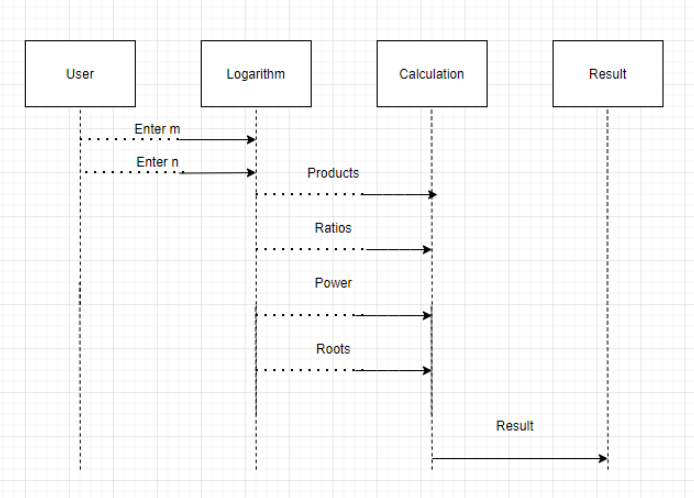
HLR Calculator Diagram (Sequence Diagram)



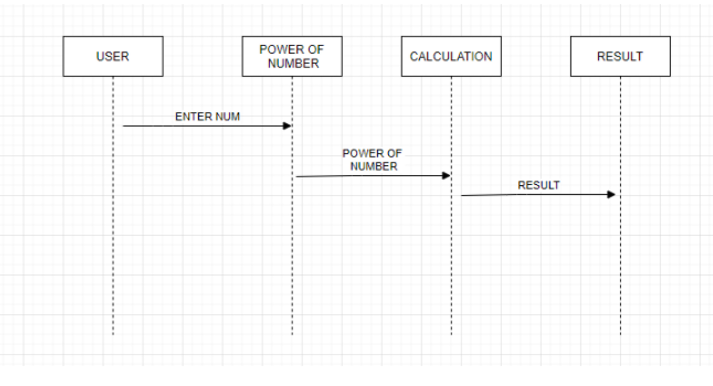
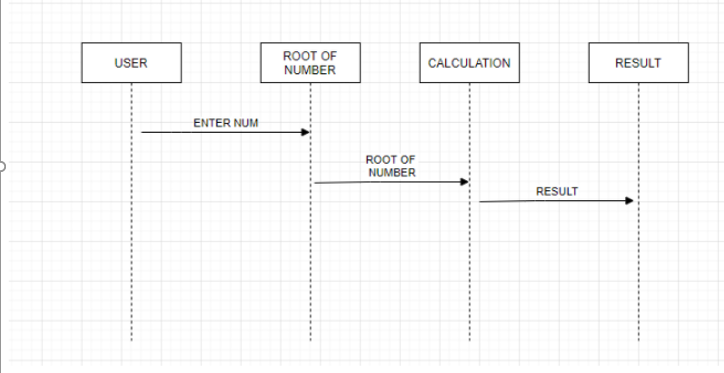
LLR Diagram (Factorial)



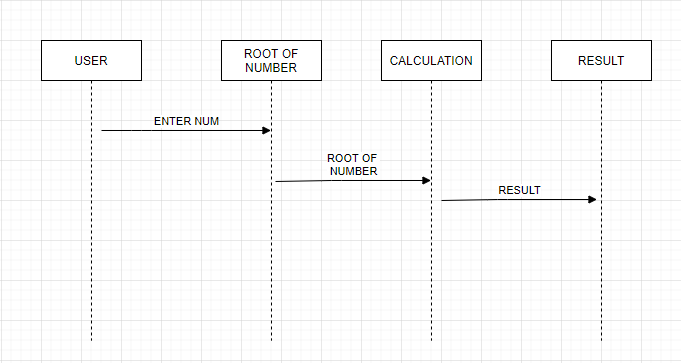
LLR Diagram (Logarithm)



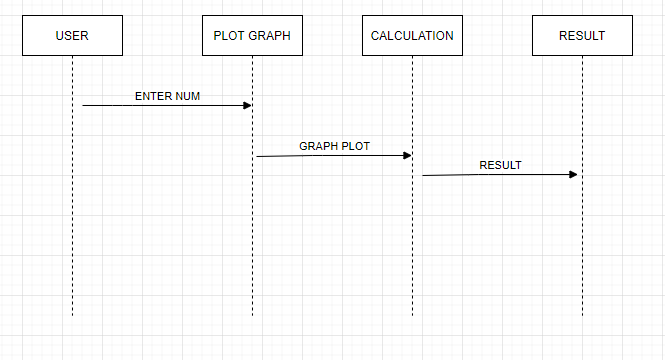
LLR Diagram (Finding a Power)



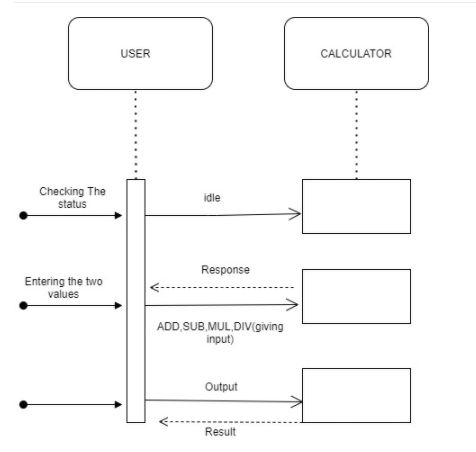
LLR Diagram (Finding Root of Number)



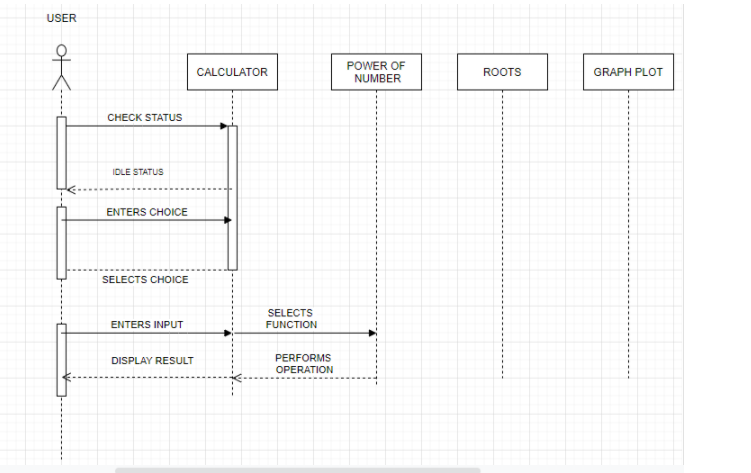
LLR Diagram (Plotting Graph)



HLR Diagram (Arithmetic Operation)



HLR Diagram ( Power, Root, Graph)



## 

## Test Plan

Complex Number

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test\_ID | Description | Expected Input | Expected Output | Actual Output | Type of Test |
| L001 | Giving right value to check the function | (1,1) (2,1) | Add-(3,2) | Add-(3,2) | Requirement Based |
| L002 | Checking the function subtraction | (3,6) (2,4) | Sub-(1,2) | Sub-(1,2) | Requirement Based |
| L003 | Subtraction  Input1<Input2 | (4,5) (6,7) | (-2,-7) | (-2,-7) | Scenario Based |

Factorial

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test\_ID | Description | Expected Input | Expected Output | Actual Output | Type of Test |
| TCN001 | Standard Input | 5! | 120 | 120 | Requirement Based |
| TCN002 | Standard Input | 1! | 1 | 1 | Scenario Based |
| TCN003 | Rational Number | 1/2 | Error | Error | Scenario Based |

Logarithm

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test\_ID | Description | Expected Input | Expected Output | Actual Output | Type of Test |
| TCN001 | Standard Input | m=5  n=6 | Product=116  Ratio=300 | Product=116  Ratio=300 | Requirement Based |
| TCN002 | Standard Input | m=9  n=10 | Product=456  Ratio=554 | Product=456  Ratio=554 | Requirement Based |
| TCN003 | Particular Input | m=0  n=5 | Error | Error | Scenario Based |

Basic Arithmetic Operations

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test\_ID | Description | Expected Input | Expected Output | Actual Output | Type of Test |
| AO\_001 | Standard Input  (ADD) | n1=2  n2=5 | 7 | 7 | Requirement Based |
| AO\_002 | Standard Input  (SUB) | n1=8  n2=3 | 5 | 5 | Requirement Based |
| AO\_003 | Standard Input  (DIV) | n1=1  n2=0 | Error | Error | Scenario Based |

Power

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test\_ID | Description | Expected Input | Expected Output | Actual Output | Type of Test |
| TC\_001 | Standard Input | Input1: 2  Input2: 3 | 8 | 8 | Requirement Based |
| TC\_002 | Standard Input | Input1: 5  Input2: 2 | 25 | 25 | Requirement Based |
| TC\_003 | Standard Input | Input1: 10  Input2: 2 | 100 | 100 | Requirement Based |

Roots

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test\_ID | Description | Expected Input | Expected Output | Actual Output | Type of Test |
| TC\_004 | Standard Input | 4 | 2 | 8 | Requirement Based |
| TC\_005 | Standard Input | 0.25 | 0.5 | 0.5 | Scenario Based |
| TC\_006 | Standard Input | 16 | 4 | 100 | Requirement Based |

## Implementation Summary

Now days calculator is contributing in each one of us life, some of us using for very basic arithmetic calculation and some of using for calculating such a complex problem which take so many hours to calculate by manually. Hence in market there are various categories of calculator available based on your requirements. Some of them are made to be very specific in term of their using and some of them are used by different-different class of people who are using it. Like students, graduate students, business man, local shops and etc. If define this device in very short then we can say it is a device that performs arithmetic operations on numbers. Implementation folder has all the source files, header files, test files for different features of the calculator.

Here inc folder contains all header files with ‘.h’ extension. It contains prototype of all functions.

* The src folder holds all the source file with ‘.c’ extension. It has definition of all function whose prototype is define in inc folder.
* The test folder contain test.c file for testing of source code based on requirement, scenario, and boundary.
* The unity folder contains file which holds prototype and definition of the standard unity test case functions.
* And then there is Makefile

### Video Summary

“Please upload a short video on the repo for the walkthrough of the project (Team/Individual) less than 7min and less than 30MB File Size. Start is the Standard opening slide with title of miniproject + Team members followed by the walkthrough ”

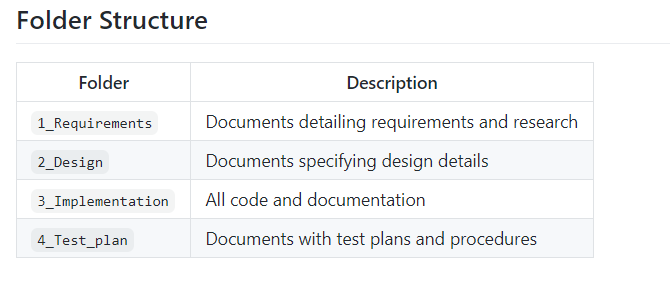
### Git Link

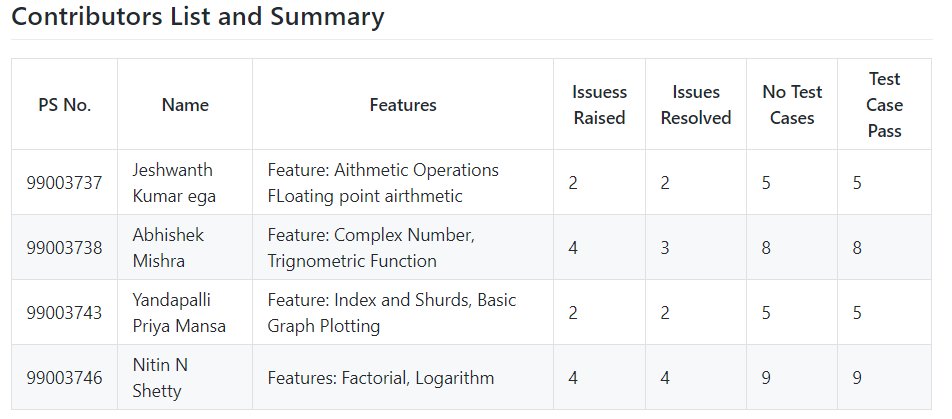
### <https://github.com/99003738/AppliedSDLC_N3.git>

### Git Dashboard

Badges:-







### Summary

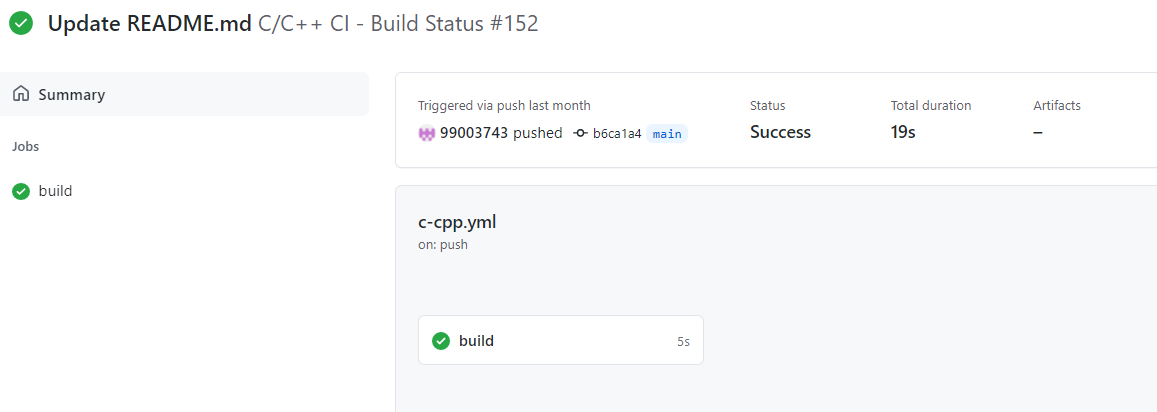
It is a simple electronic hardware/software device that is capable of performing the simple calculations such as addition, subtraction, multiplication, division, calculating power of number, exponential function, logarithmic function, permutation and combination, trigonometry, inverse-trigonometric functions,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

#### Git inspector summary

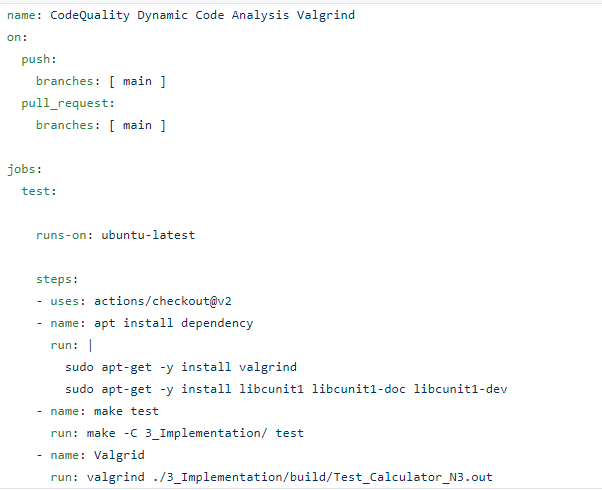
#### 

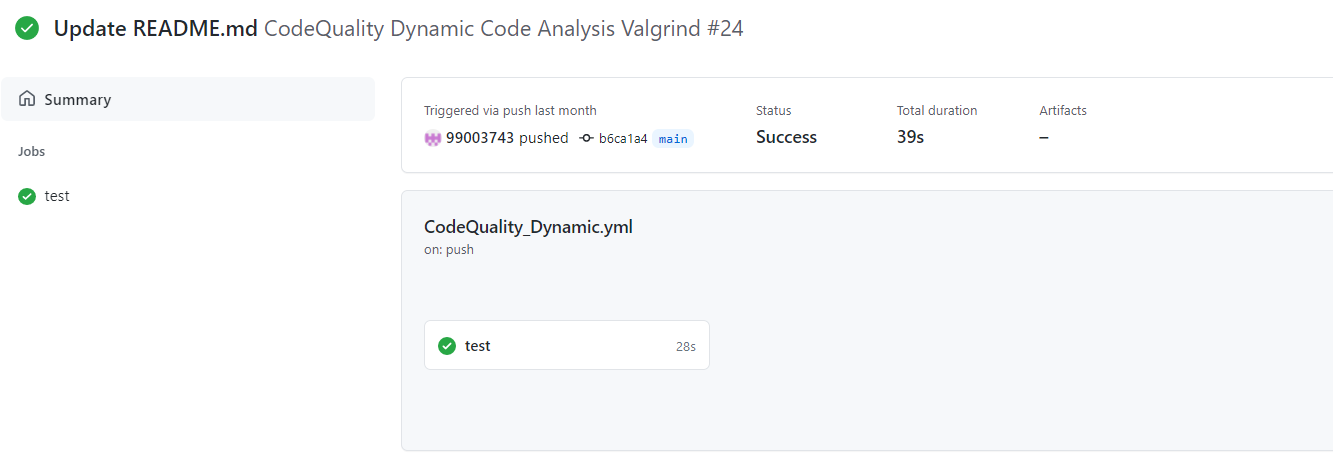
#### Build



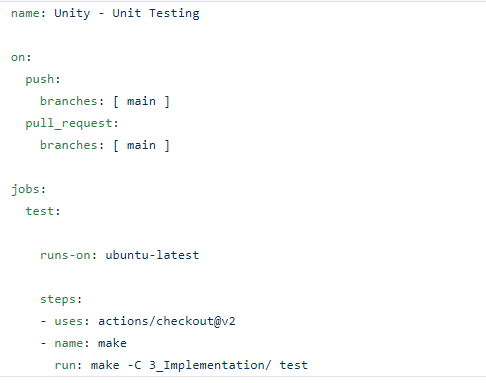


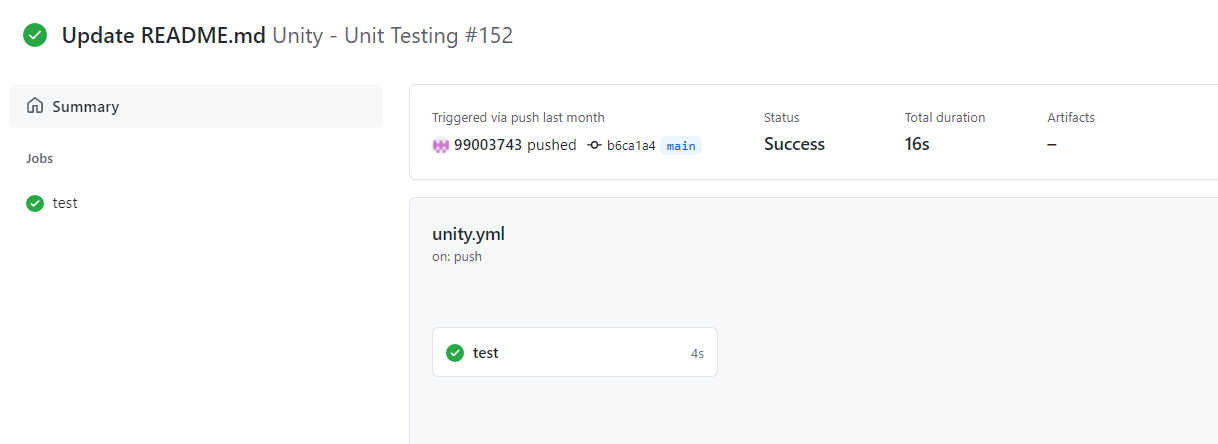
#### Code quality and Issues or Bug Tracking





#### Unit Testing





#### Individual Contribution & Highlights

* In Calculator Project, I have done Fraction and Logarithm Feature
* Implemented both the feature in C language
* Implemented all possible test cases for the same
* High level and Low-Level Requirements are listed
* Plotted UML (Structural and Behavioral) diagrams for High level and Low-Level Requirements
* Listed 4W1H for the Project
* Issues are raised and respective issues are solved.
* Contributed in overall workflow and in project implementation.

### Summary

### The main motto is to design a calculator with certain features according to the specific requirements. The target customers for the designed calculator are students, shopkeepers, banking executives and engineers.

Technical:

* Improved implementation of C Concepts
* Practical Implementation of SDLC life Concepts
* Source code management(Github)

Soft skills:

* Project management
* Conflict management

### Challenges faced and how were they overcome

* Running the make file as its resolved by defining its correct path(.out for linux and -lm for math functions)
* Synchronizing the VS code to github, colleague help to resolve the issue
* Making the function call in correct path
* Open git log while committing, thus went to github desktop and pulled origin and then pushed origin.
* Test case code for the boundary problem. Added code with the help of internet
* Integration problem since lots of header file was there. Changes made in header file to remove the multiple occurrence error.

### 

# Miniproject -2 [Individual]

## Module Used

In this Module we used Advanced python for project.

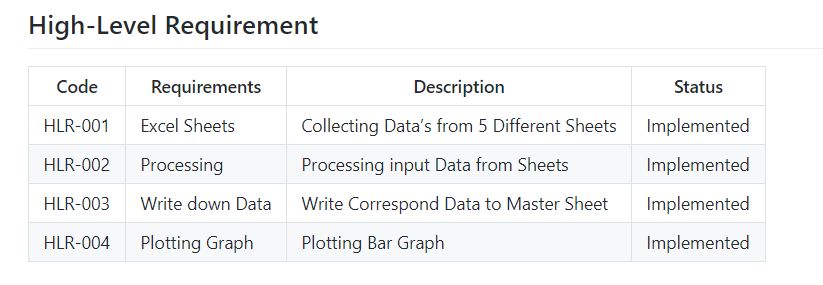
Project title : Retrieving particular data from different sheets and adding retrieved data in Master Sheet

### Topic and Subtopics

## Objectives

## I am having collection data in Different Sheets.This program takes a user input as a keyword and search the occurrence of the word in the Excel file and assembles all corresponding data from 5 sheets, and store these details in a Master Sheet.

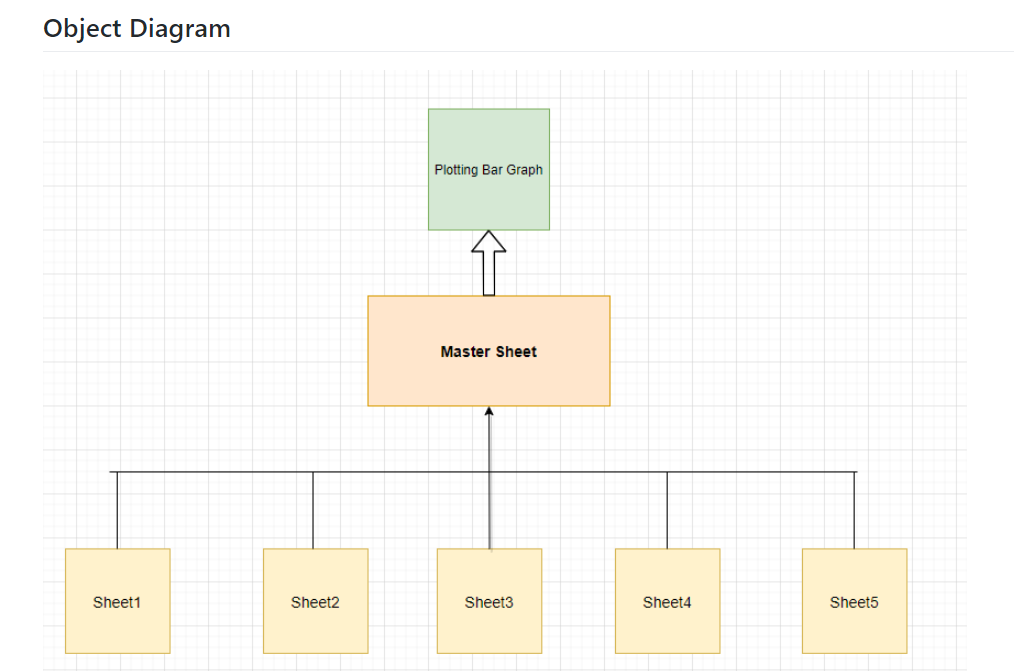
## Requirements



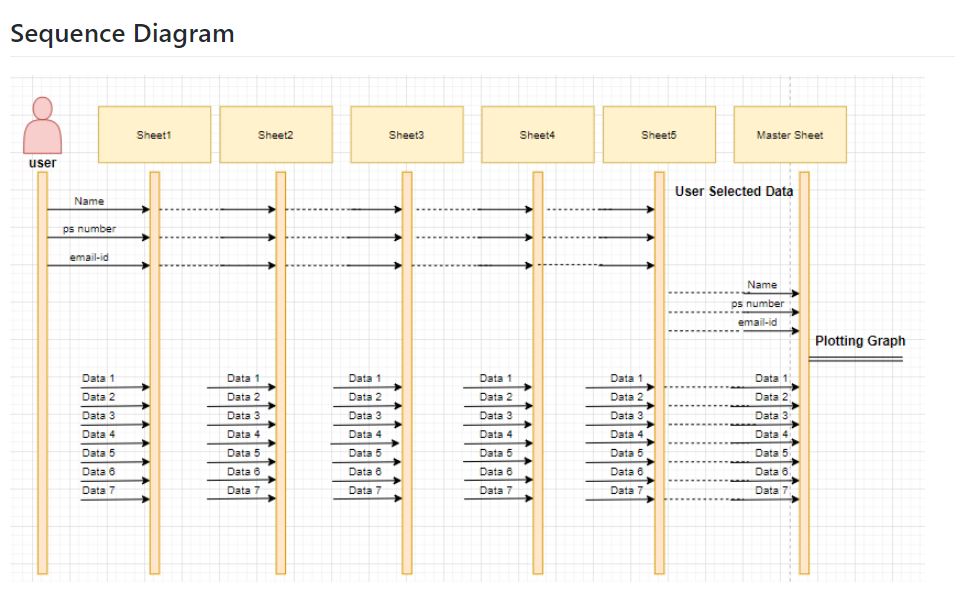
## 

## Design

High Level Design



## Low Level Design



## 

## Implementation Summary

I am having collection of students marks data from Sem 1 to Sem 5 in 5 Different Sheets. This program takes a user input as a keyword and search the occurrence of the word in the Excel file and assembles all corresponding data from 5 sheets, and store these details in a Master Sheet.

1. There are 10 columns in each sheet, All Sheets have 3 common data i.e. Name, PS Number & Email. The remaining columns are unique to each sheet.
2. The user provides these 3 common values to search for a Data. Using these 3 values the record is searched in all available sheets.
3. User can also search repeated Name, PS-no and also email-id
4. This record from all the sheets is appended in a list and is Written in the Master Sheet in a single row, against the name.
5. The recent data from the Master Sheet is Represented by Plotting Graph.

### Git Link

<https://github.com/99003746/Mini_Project_Python.git>

### Summary

Technical:

Improved implementation of Python Concepts

* Practical Implementation of Python Concepts
* Source code management
* Used predefined modules ( openpyxl, pandas)

Soft skills:

* Project management
* Conflict management

### Challenges faced and how were they overcome

* Differentiation of high level and low level.
* Committing to GitHub, pull and push in GitHub.
* Converting pictures and tables into readme file.
* System issues (crashing and Interfacing)
* Understanding the modules of openpyxl and pandas.