./

GENESIS - Learning Outcome & Mini-project Summary Report



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

**Details**

Contents

[Contents 3](#_Toc69294715)

[Mini Project -1 Team SDLC (System Development life cycle) 4](#_Toc69294716)

[1.1 Modules Used 4](#_Toc69294717)

[1.2 Project title: Mini Calculator 4](#_Toc69294718)

[1.3 Topic and Subtopics 4](#_Toc69294719)

[1.4 Objectives & Requirements 4](#_Toc69294720)

[1.5 Design 5](#_Toc69294721)

[1.6 Test Plan 6](#_Toc69294722)

[1.7 Implementation Summary 7](#_Toc69294723)

[1.8 Video Summary 7](#_Toc69294724)

[1.9 Git Link 7](#_Toc69294725)

[https://github.com/99003774/N7\_SDLC\_Calculator.git 7](#_Toc69294726)

[1.10 Git Dashboard 7](#_Toc69294727)

[1.11 Individual Contribution & Highlights 11](#_Toc69294728)

[Summary 12](#_Toc69294729)

[Challenges faced and how were they overcome 12](#_Toc69294730)

[Future Scope (If applicable) 12](#_Toc69294731)

[Miniproject -2 [Team/Individual] 13](#_Toc69294732)

[Module/s 13](#_Toc69294733)

[Topic and Subtopics 13](#_Toc69294734)

[Objectives & Requirements 13](#_Toc69294735)

[Design 13](#_Toc69294736)

[Test Plan 13](#_Toc69294737)

[Implementation Summary 13](#_Toc69294738)

[Git Link 13](#_Toc69294739)

[Git Dashboard 13](#_Toc69294740)

[Summary 13](#_Toc69294741)

[Individual Contribution & Highlights 13](#_Toc69294742)

[Summary 13](#_Toc69294743)

[Challenges faced and how were they overcome 13](#_Toc69294744)

# Mini Project -1 Team SDLC (System Development life cycle)

## 1.1 Modules Used

Modules used in this project are SDLC and C programming.

## 1.2 Project title: Mini Calculator

“Modules linked to the miniproject Ex – Linux, SDLC and C++ or SDLC and HTML etc”

## 1.3 Topic and Subtopics

* The core steps of SDLC is being implemented.
  + The features of Calculator are implemented.
  + The testing has been done for each function.
* Introduction about SDLC
* C Programming
* Code Analysis
  + CPP Check
  + Valgrind
* Testing
  + Unity Testing
* Makefile
* V Model
* Agile Model
* Git Hub

“Briefly list the core topics and subtopics being implemented and how”

## 1.4 Objectives & Requirements

“High level and low level in the template”

High Level Requirements:

|  |  |  |
| --- | --- | --- |
| **ID** | **Description** | **Status** |
| 01 | 14-digit screen input. | Implemented. |
| 02 | Dedicated MRC (Memory Recall and Clear). | Implemented. |
| 03 | Dedicated check keys. | Implemented. |
| 04 | Permutation and combination functions are performed. | Implemented. |
| 05 | Volume for cone, sphere, cylinder. | Implemented. |
| 06 | Area for square and rectangle. | Implemented. |
| 07 | Arithmetic operations are performed. | Implemented. |

Low Level Requirements:

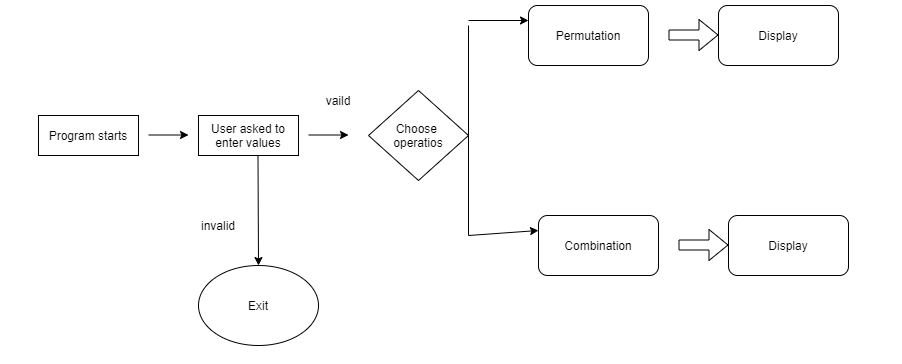
|  |  |  |
| --- | --- | --- |
| **ID** | **Description** | **Status** |
| 01 | Dedicated ON/OFF switch. | Implemented. |
| 02 | Grand total key. | Implemented. |
| 03 | Decimal key. | Implemented. |
| 04 | Basic math operations keys. | Implemented. |

## 1.5 Design

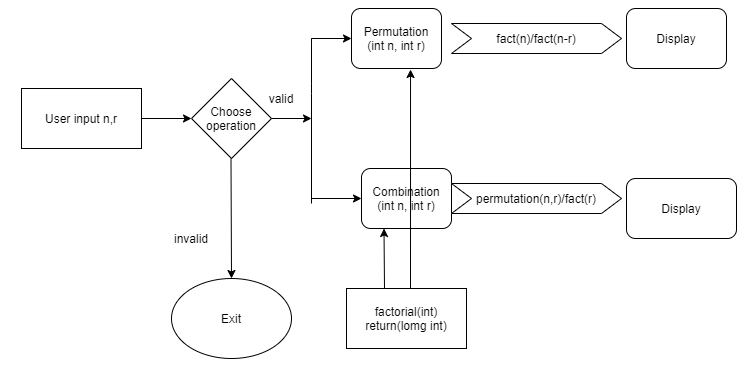
“System Level and subsystem level UMLs – Structural and Behavioral”

1.5.1 Permutation and Combination

High Level Requirement:

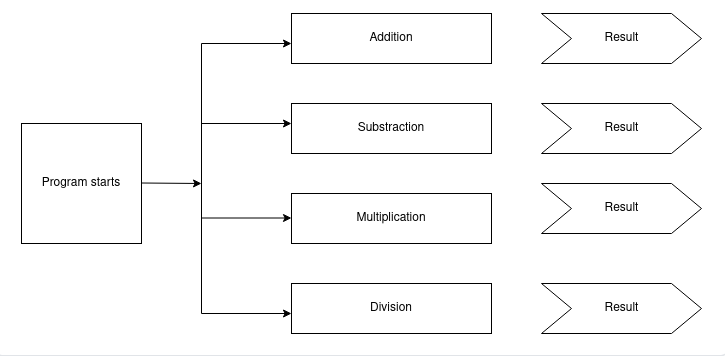


Low Level Requirement:

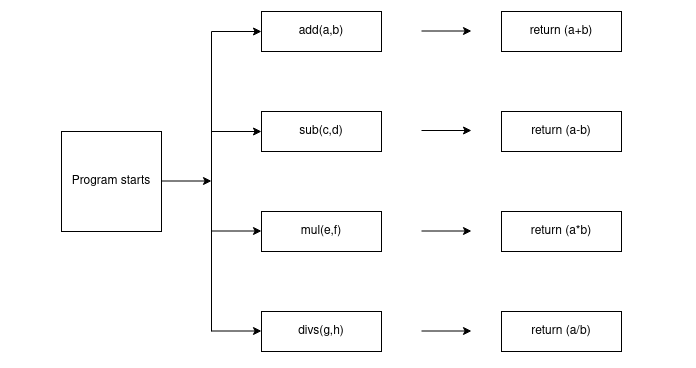


1.5.2 Arithmetic operations:

High Level Requirement:

****

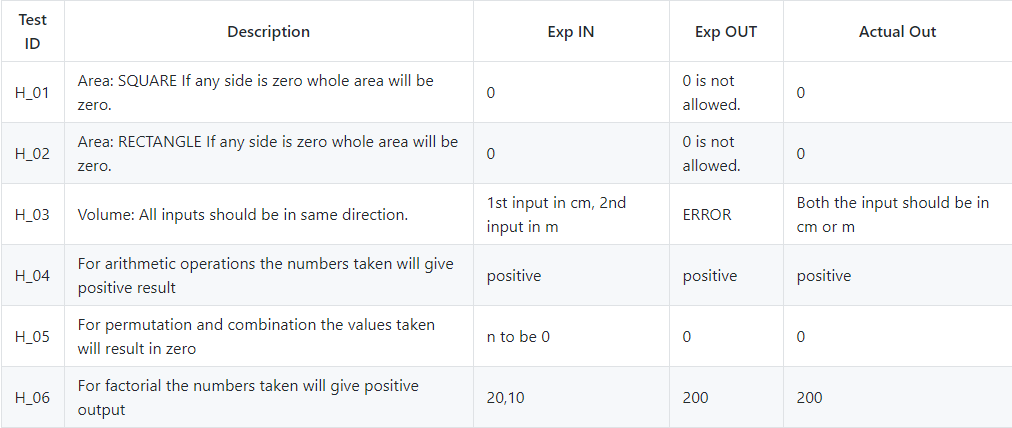
Low Level Requirement:



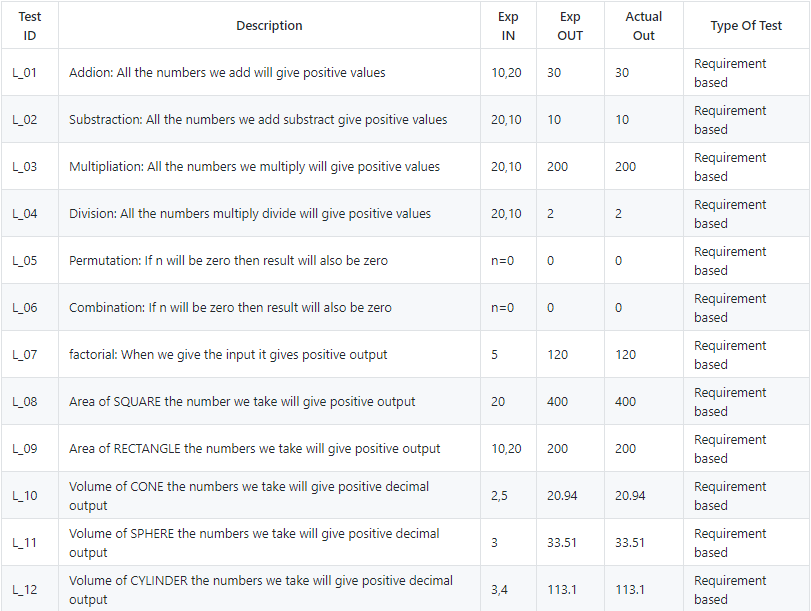
## 1.6 Test Plan

“Integration level and unit level in the template”

1.6.1 High Level Test Plan

****

1.6.2 Low Level Test Plan

****

## 1.7 Implementation Summary

It is a basic calculator that will allow users to perform operations in Mathematics Addition, Subtraction, Multiplication, Division, Trigonometry, Factorial, Area, Volume etc. However, the input has to be in the form "number1 operator1 number2 operator2 number3" (i.e 2+4\*10). The input values can be from any integer to even a number with decimals. Moreover, this calculator is smart enough to operate multiplication/division before addition/subtraction, in another word it is implemented with the order of precedence logic.

### 1.8 Video Summary

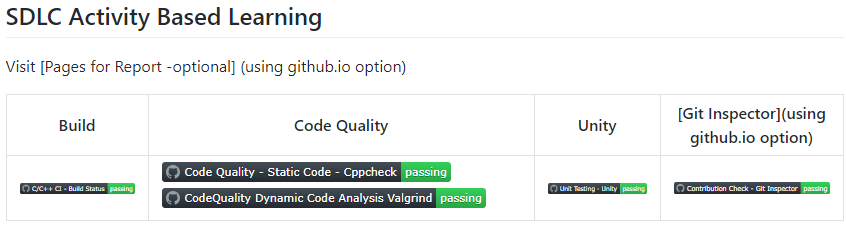
“Please upload a short video on the repo for the walkthrough of the project (Team/Individual) less than 7min and less than 30MB File Size. Start is the Standard opening slide with title of miniproject + Team members followed by the walkthrough”

### 1.9 Git Link

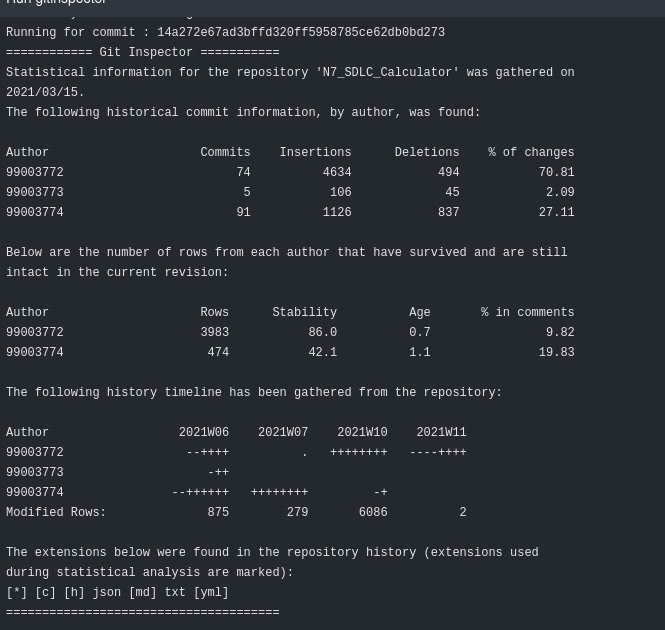
### https://github.com/99003774/N7\_SDLC\_Calculator.git

### 1.10 Git Dashboard

1.10.1 Badges

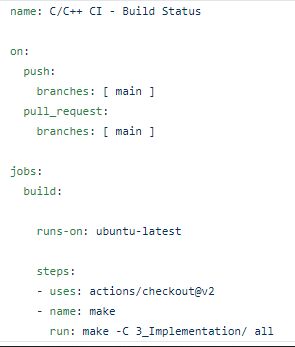


#### 1.10.2 Git inspector summary

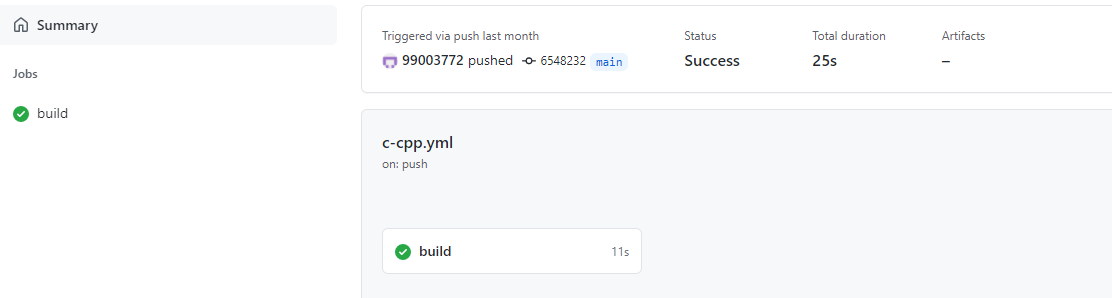


#### 1.10.3 Build

1.10.3.1 Setup for Build



1.10.3.2 Outcome of the Build

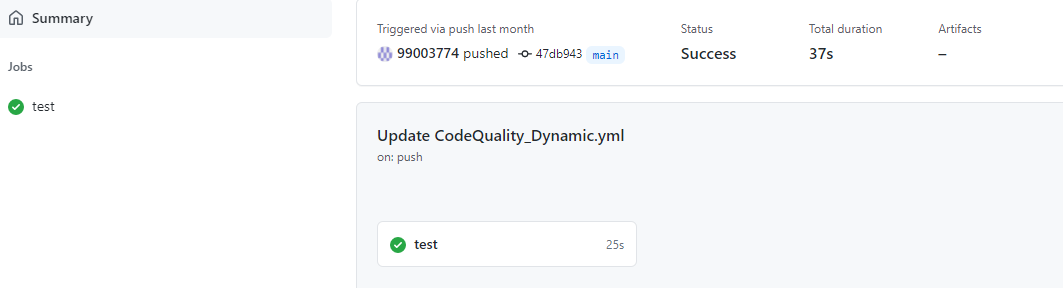


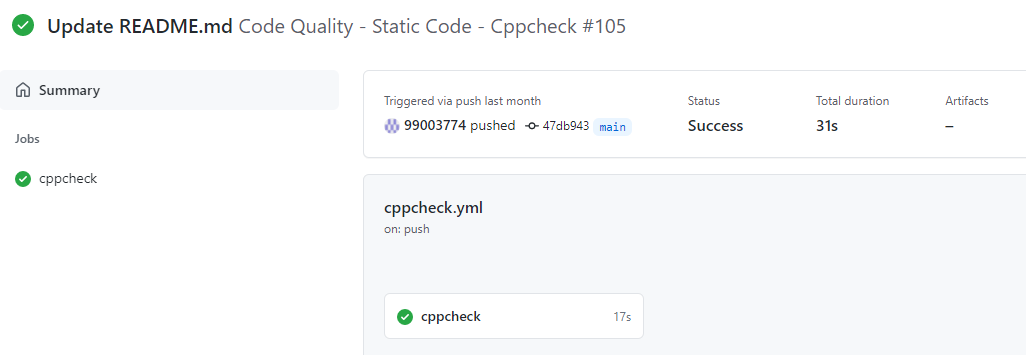
#### 1.10.4 Code quality and Issues or Bug Tracking

1.10.4.1 Setup for Code Quality



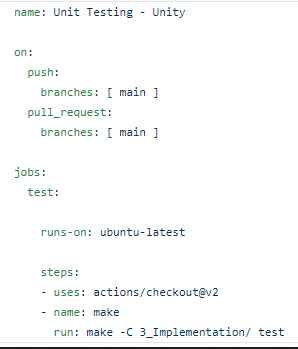
1.10.4.2 Outcome of the Cody Quality



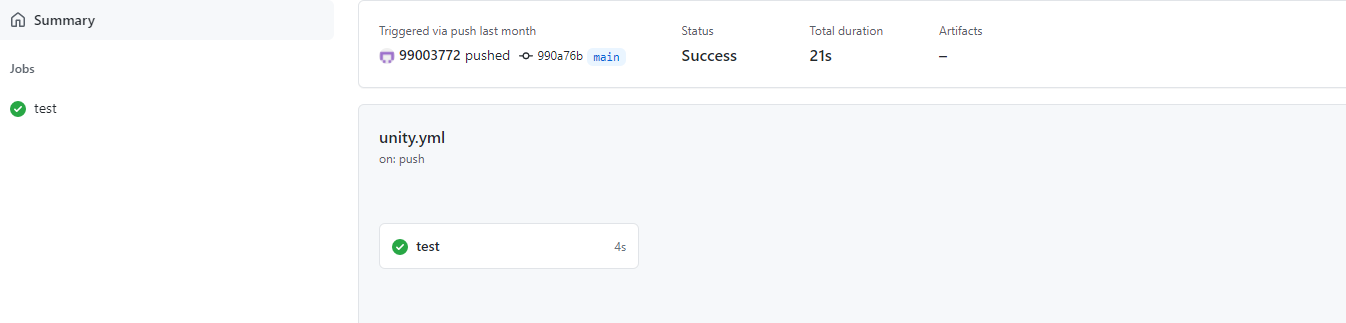


#### 1.10.5 Unit Testing

1.10.5.1 Setup for Unity Testing

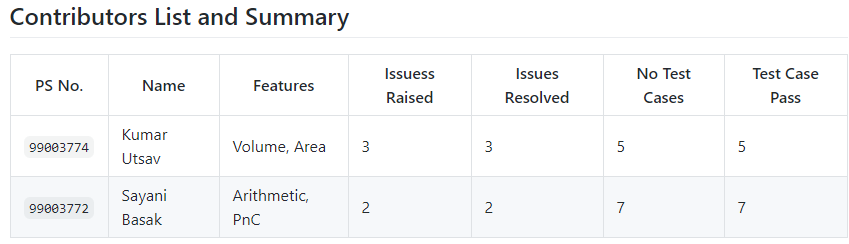


1.10.5.2 Outcome of the Unity Testing



## 1.11 Individual Contribution & Highlights

* Trigonometry and Logical feature is implemented.
* Test case for the same is implemented.
* High level and low-level test cases is implemented for the same.
* Issue raised and the issue was solved.
* Helped during the workflow’s implementation of the project.



### 1.12 Summary

Technical:

* Improved implementation of ‘C’ concepts.
* Practical implementation of SDLC lifecycle.
* Source code management. (Github)

Soft skills:

1. Project management

2. Conflict management.

### 1.13 Challenges faced and how were they overcome

* System issues (crashing and Interfacing).
* Differentiation of high level and low level.
* Committing to GitHub, pull and push in GitHub.
* Converting pictures & tables into readme file.
* Cpp check and Unity testing.

### Future Scope (If applicable)

### 

# Miniproject -2 [Team/Individual]

## Module/s

### Topic and Subtopics

## Objectives & Requirements

## Design

## Test Plan

## Implementation Summary

### Git Link

### Git Dashboard

### Summary

#### Git inspector summary

#### Build

#### Code quality

#### Unit Testing

#### Issues

## Individual Contribution & Highlights

### Summary

### Challenges faced and how were they overcome

# Mini Project -3 Python Programming (System Development life cycle)

## 3.1 Modules Used

Modules used in this project are SDLC and C programming.

## 3.2 Project title: Mini Calculator

“Modules linked to the miniproject Ex – Linux, SDLC and C++ or SDLC and HTML etc”

## 3.3 Topic and Subtopics

* The core steps of SDLC is being implemented.
  + The features of Calculator are implemented.
  + The testing has been done for each function.
* Introduction about SDLC
* C Programming
* Code Analysis
  + PEP8
* V Model
* Agile Model
* Git Hub

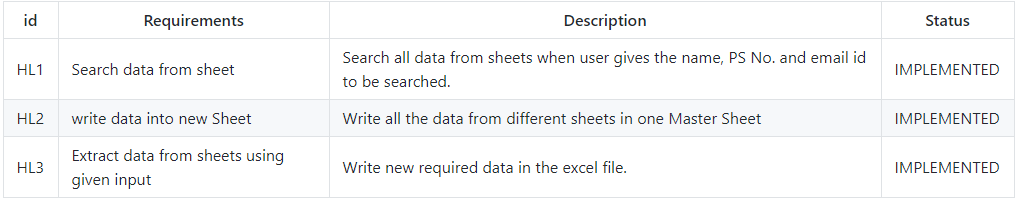
“Briefly list the core topics and subtopics being implemented and how”

## 3.4 Objectives:

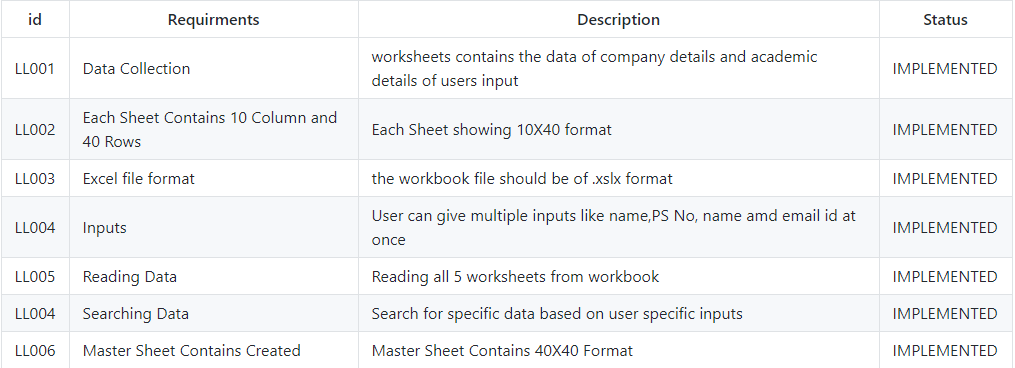
To extract the data present in different spreadsheets in one excel file as required by the user.

## Requirements:

3.4.1 High Level Requirement Analysis:

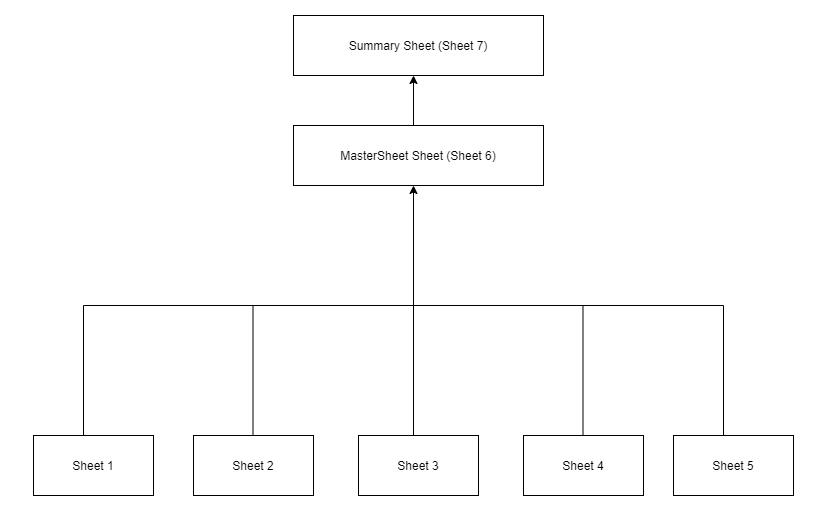


3.4.2 Low Level Requirement Analysis:

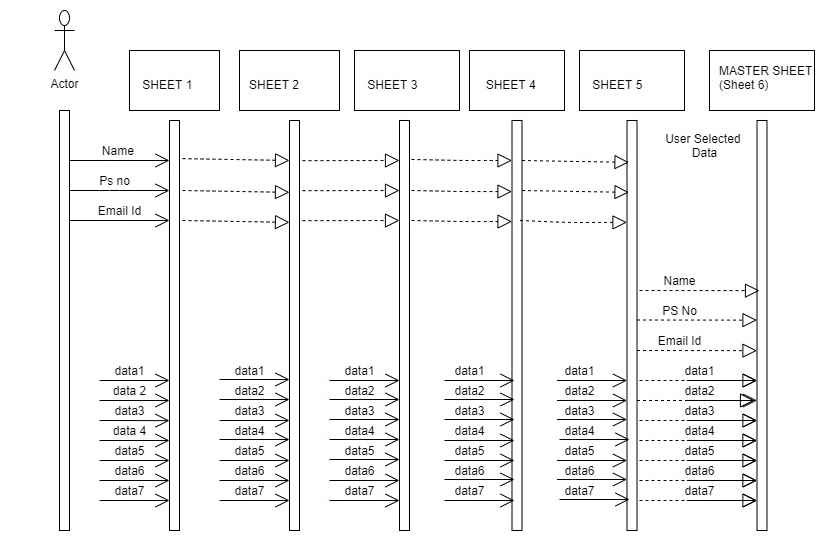


## 3.5 Design

3.5.1 High Level Diagram



3.5.2 Low Level Diagram



## 3.6 Test Plan

## 3.7 Implementation Summary

The aim of the project is to extract the data present in different spreadsheets in one excel file as required by the user. The excel sheet consists of 5 spreadsheets. The user defines the data that needs to be searched on the basis of Name, PS Number and Email ID. The python program then reads the data corresponding to the particular data from different spreadsheets of excel. It then creates a master sheet and adds the data from all the sheets to it. In the end all data will be printed in Data1.xlsx file, all data store in mastersheet present in Data1.xlsx file and the total number of data will also be shown in the summary sheet (data of individual and also the total no. of data present in the master sheet)..

## **3..8** 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”

### 3.9 Git Link

https://github.com/99003772/Python\_miniproject.git

## 3.10 Individual Contribution & Highlights

* Trigonometry and Logical feature is implemented.
* Test case for the same is implemented.
* High level and low-level test cases is implemented for the same.
* Issue raised and the issue was solved.
* Helped during the workflow’s implementation of the project

Snapshot of contribution list and summary is given from git

### 3.10 Summary

Technical:

* Improved implementation of Python concepts.
* Practical implementation of SDLC lifecycle.
* Source code management. (GitHub)

Soft skills:

1. Project management

2. Conflict management.

### 3.11 Challenges faced and how were they overcome

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