––./

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](#_Toc69367039)

[Mini Project -1 Team SDLC (System Development life cycle) 5](#_Toc69367040)

[1.1 Modules Used 5](#_Toc69367041)

[1.2 Project title: Mini Calculator 5](#_Toc69367042)

[1.3 Topic and Subtopics 5](#_Toc69367043)

[1.4 Objectives & Requirements 5](#_Toc69367044)

[1.5 Design 6](#_Toc69367045)

[1.5.1 High Level Diagram of the Calculator: 6](#_Toc69367046)

[1.5.2 Permutation and Combination 7](#_Toc69367047)

[1.5.3 Arithmetic operations: 8](#_Toc69367048)

[1.6 Test Plan 9](#_Toc69367049)

[1.6.1 High Level Test Plan 9](#_Toc69367050)

[1.6.2 Low Level Test Plan 9](#_Toc69367051)

[1.7 Implementation Summary 10](#_Toc69367052)

[1.8 Video Summary 10](#_Toc69367053)

[1.9 Git Link 10](#_Toc69367054)

[1.10 Git Dashboard 10](#_Toc69367055)

[1.10.1 Badges 10](#_Toc69367056)

[1.10.2 Git inspector summary 11](#_Toc69367057)

[1.10.3 Build 12](#_Toc69367058)

[1.10.4 Code quality 13](#_Toc69367059)

[1.10.5 Git issues 15](#_Toc69367060)

[1.10.6 Unit Testing 15](#_Toc69367061)

[1.11 Individual Contribution & Highlights 17](#_Toc69367062)

[1.12 Summary 17](#_Toc69367063)

[1.13 Challenges faced and how were they overcome 18](#_Toc69367064)

[1.14 Future Scope (If applicable) 18](#_Toc69367065)

[Mini project -2 [Team/Individual] 19](#_Toc69367066)

[Module/s 19](#_Toc69367067)

[Topic and Subtopics 19](#_Toc69367068)

[Objectives & Requirements 19](#_Toc69367069)

[Design 19](#_Toc69367070)

[Test Plan 19](#_Toc69367071)

[Implementation Summary 19](#_Toc69367072)

[Git Link 19](#_Toc69367073)

[Git Dashboard 19](#_Toc69367074)

[Summary 19](#_Toc69367075)

[Individual Contribution & Highlights 19](#_Toc69367076)

[Summary 19](#_Toc69367077)

[Challenges faced and how were they overcome 19](#_Toc69367078)

[Mini Project -3 Python Programming (System Development life cycle) 20](#_Toc69367079)

[3.1 Modules Used 20](#_Toc69367080)

[3.2 Project title: Mini Calculator 20](#_Toc69367081)

[3.3 Topic and Subtopics 20](#_Toc69367082)

[3.4 Objectives: 20](#_Toc69367083)

[3.4.1 Introduction 20](#_Toc69367084)

[3.4.2 4W & 1H 20](#_Toc69367085)

[3.4.3 SWOT Analysis 21](#_Toc69367086)

[3.5 Requirements: 22](#_Toc69367087)

[3.5.1 High Level Requirement Analysis: 22](#_Toc69367088)

[3.5.2 Low Level Requirement Analysis: 22](#_Toc69367089)

[3.6 Design 23](#_Toc69367090)

[3.6.1 High Level Diagram 23](#_Toc69367091)

[3.6.2 Low Level Diagram 23](#_Toc69367092)

[3.7 Test Plan 24](#_Toc69367093)

[3.8 Implementation Summary 24](#_Toc69367094)

[3.9 Video Summary 24](#_Toc69367095)

[3.10 Git Link 25](#_Toc69367096)

[3.11 Individual Contribution & Highlights 25](#_Toc69367097)

[3.12 Summary 25](#_Toc69367098)

[3.13 Challenges faced and how were they overcome 25](#_Toc69367099)

# 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 mini project 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

All the core-topics and sub-topics are implemented through V-Model.

## 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. |

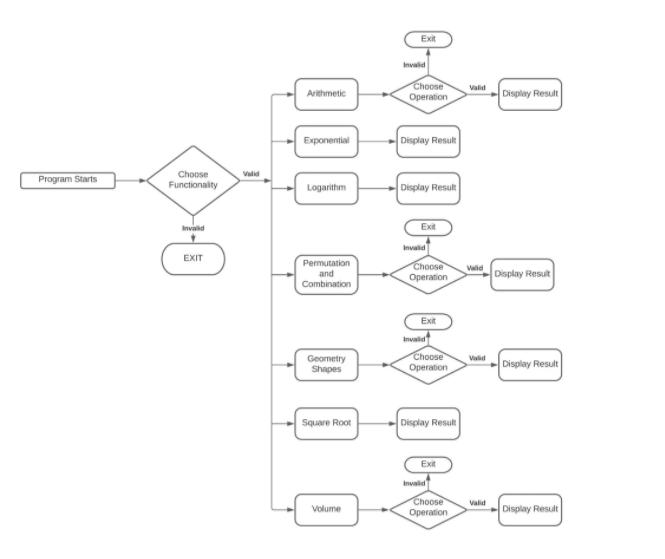
Low Level Requirements:

|  |  |  |
| --- | --- | --- |
| **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

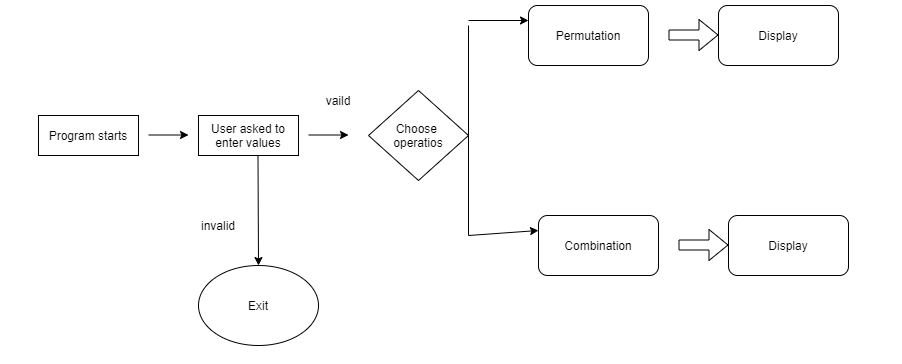
“System Level and subsystem level UMLs – Structural and Behavioral”

### 1.5.1 High Level Diagram of the Calculator:

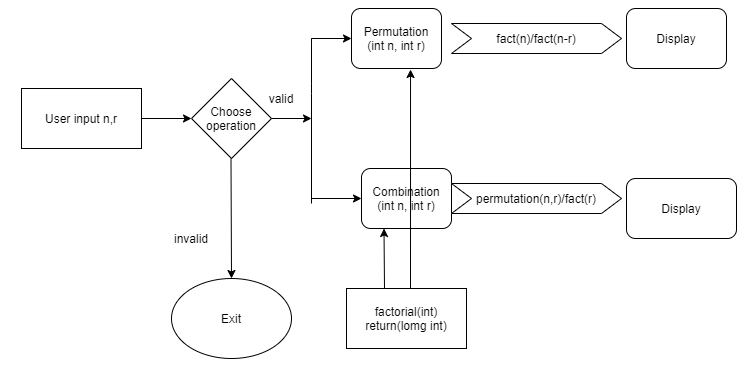


### 1.5.2 Permutation and Combination

High Level Requirement:

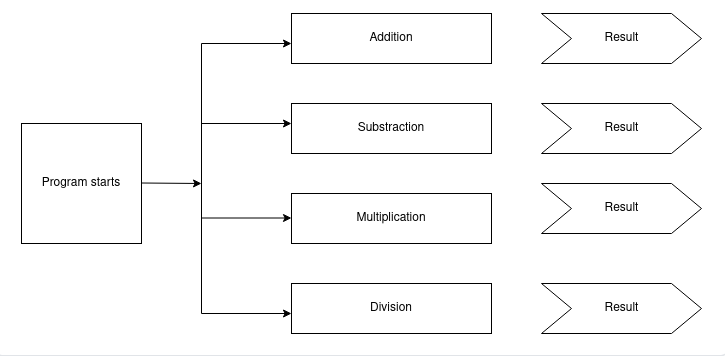


Low Level Requirement:

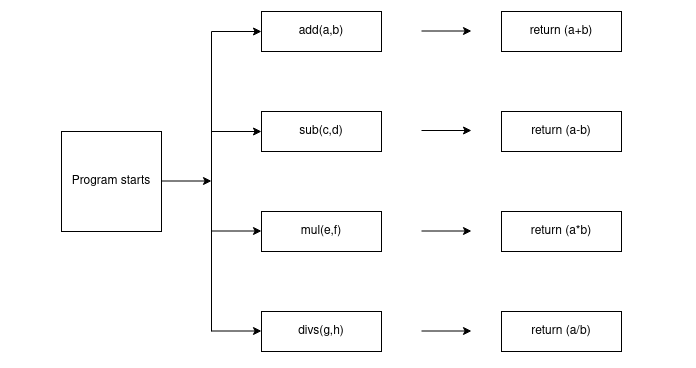


### 1.5.3 Arithmetic operations:

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Exp IN** | **Exp OUT** | **Actual OUT** |
| H\_01 | For arithmetic operations, the numbers taken will give positive result | Positive | Positive | Positive |
| H\_02 | For permutation and combination, the values taken will result in zero | N to 0 | 0 | 0 |
| H\_03 | For factorial, the numbers taken will give positive output | 20,10 | 200 | 200 |

### 1.6.2 Low Level Test Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Exp IN** | **Exp OUT** | **Actual OUT** | **Type of tests** |
| L\_01 | Addition: All the numbers we add will give positive values | 10, 20 | 30 | 30 | Requirement based |
| L\_02 | Substraction: All the numbers we add substract give positive values | 20, 10 | 10 | 10 | Requirement based |
| L\_03 | Multiplication: All the numbers we multiply will give positive values | 20, 10 | 200 | 200 | Requirement based |
| L\_04 | Division: All the numbers multiply divide will give positive values | 20,10 | 2 | 2 | Requirement based |
| L\_05 | Permutation: If n will be zero then result will also be zero | N=0 | 0 | 0 | Requirement based |
| L\_06 | Combination: If n will be zero then result will also be zero | N=0 | 0 | 0 | Requirement based |
| L\_07 | Factorial: When we give the input, it gives positive output | 5 | 120 | 120 | Requirement based |

## 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

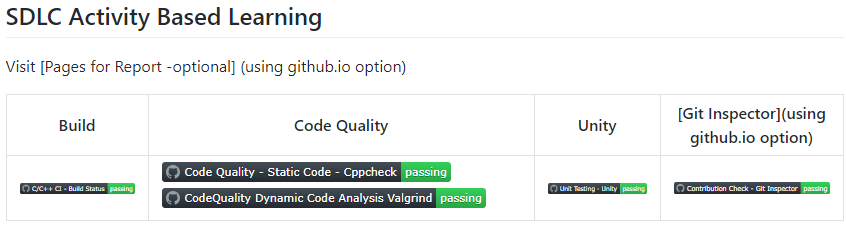
“Please upload a short video on the repo for the walkthrough of the project (Team/Individual) less than 7min and less than 30MB File Size. Start is the Standard opening slide with title of miniproject + Team members followed by the walkthrough”

## 1.9 Git Link

<https://github.com/99003774/N7_SDLC_Calculator.git>

## 1.10 Git Dashboard

### 1.10.1 Badges



**Summary**

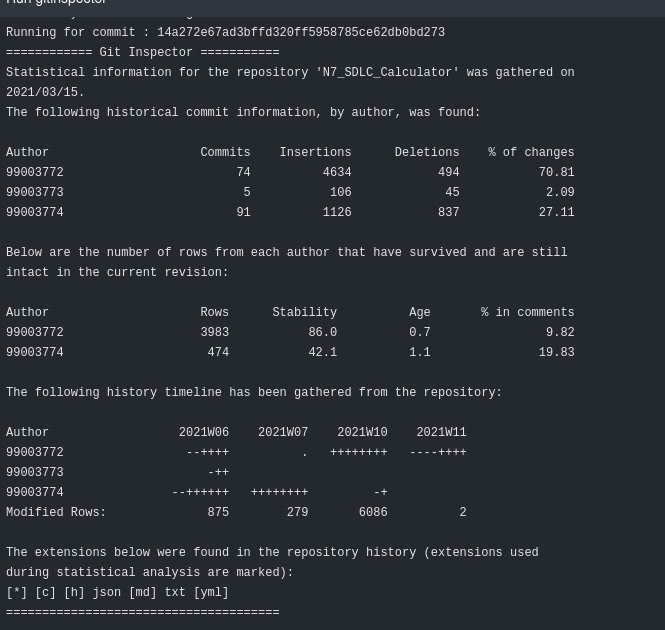
In this project, we mainly-focused on how to design a calculator which is bit different and cost-effective as compared to other calculators present in the market.

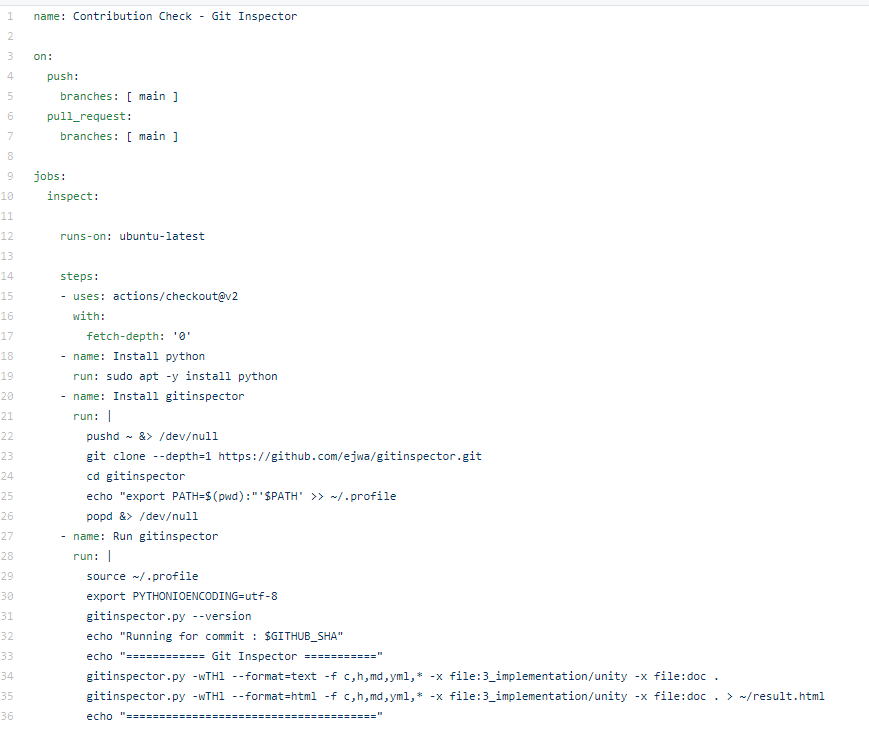
So, for this we first analyzed all the other calculators from low-end feature low cost calculator to high-end feature high-cost calculator and prepared a list of features to include in our modified cost-effective more featured calculator.

Features included in the calculator are basic arithmetic operations (addition, subtraction, multiplication, division), Permutation and Combination, Geometric Calculations.

This calculator is implemented through C programming. This calculator will be mainly used by the school, college students, businessman, engineers for various purpose. Also, this calculator will be cost effective and if implemented over hardware it cost around 800 – 1000 INR.

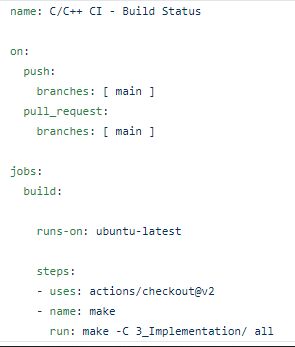
### 1.10.2 Git inspector summary



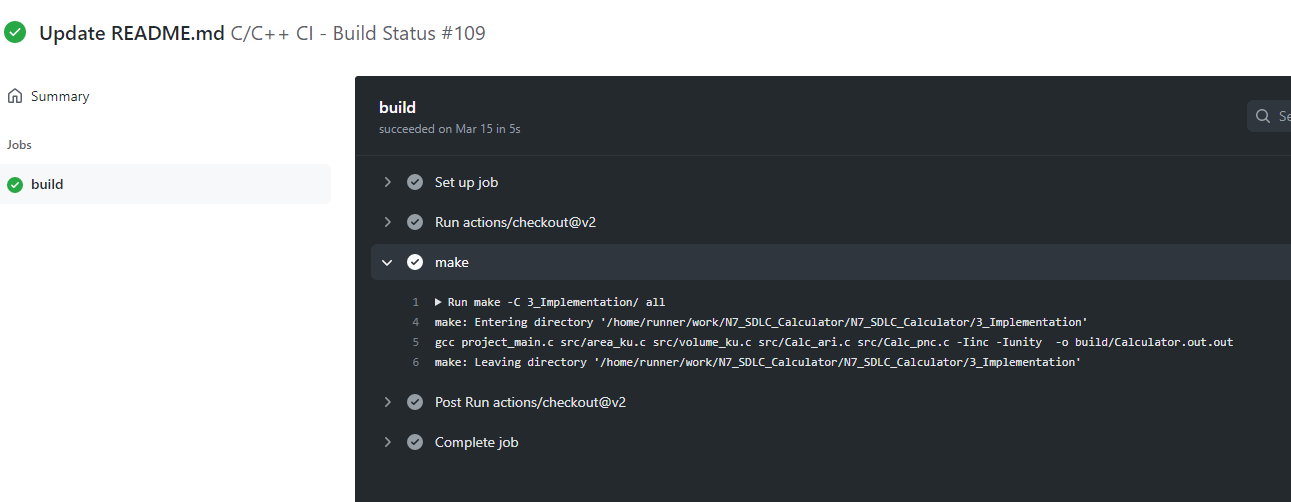


### 1.10.3 Build

#### 1.10.3.1 Setup for Build



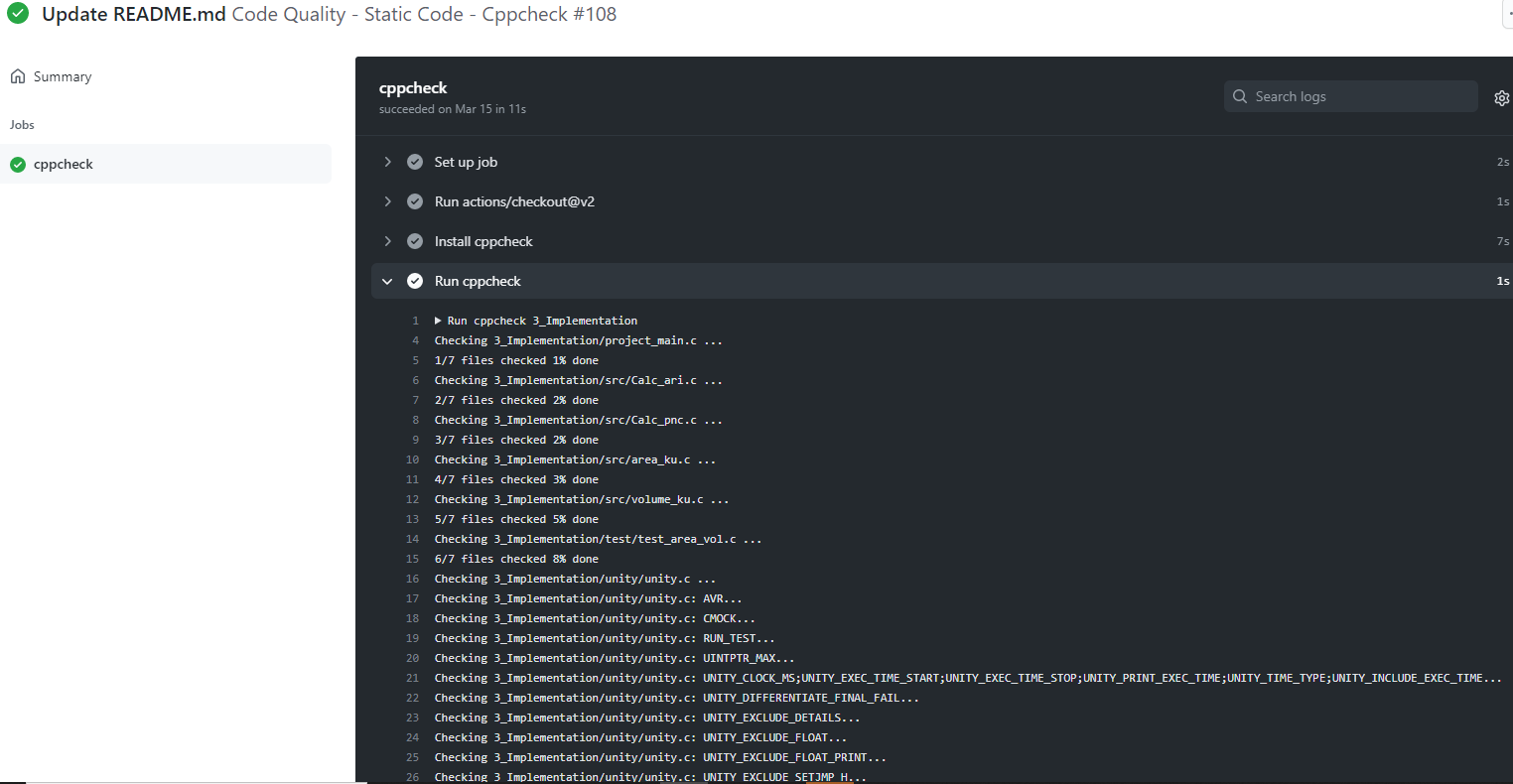
#### 1.10.3.2 Outcome of the Build



### 1.10.4 Code quality

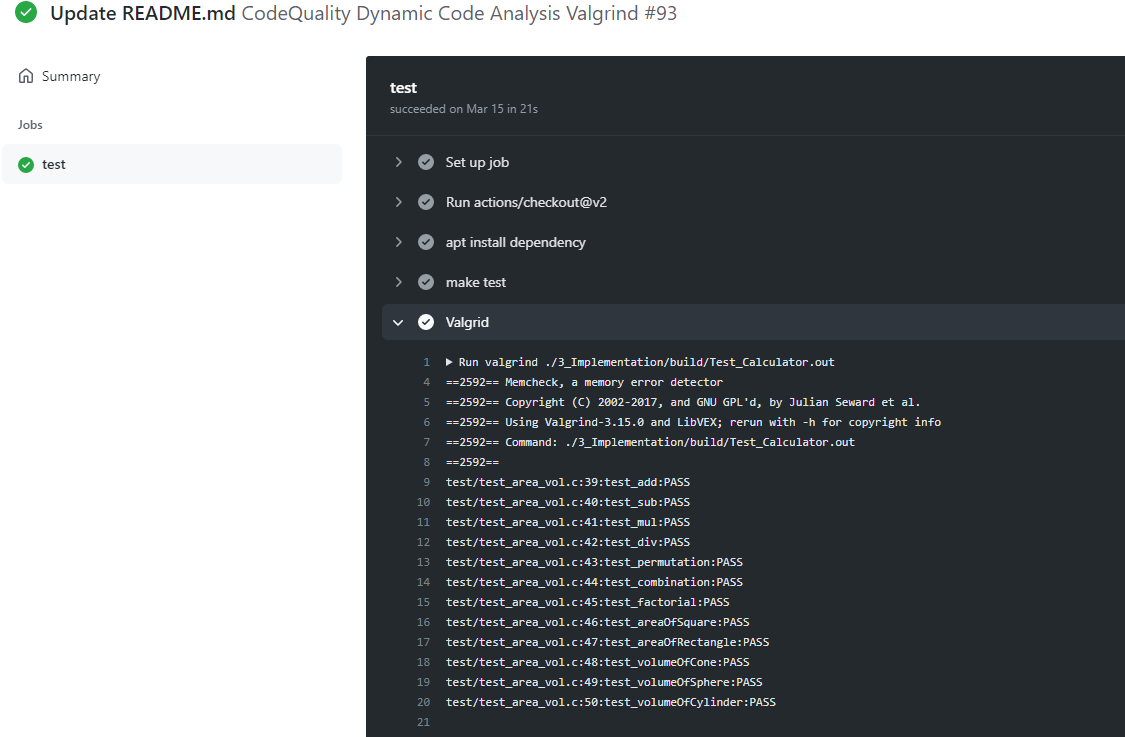
#### 1.10.4.1 Setup for Static Code Quality



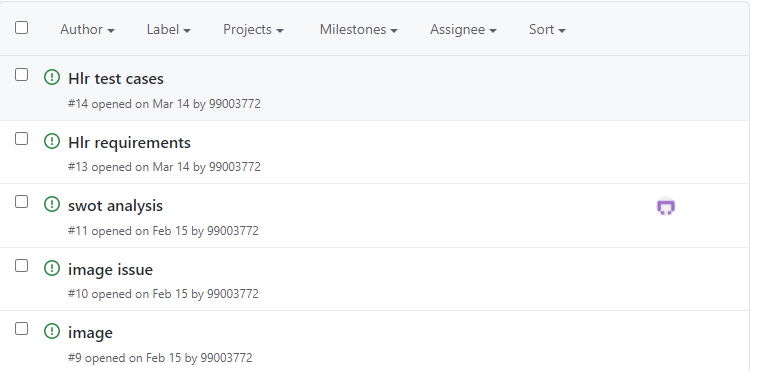


#### 1.10.4.2 Outcome of the Dynamic Cody Quality



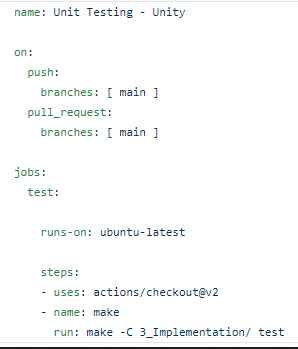


### 1.10.5 Git issues

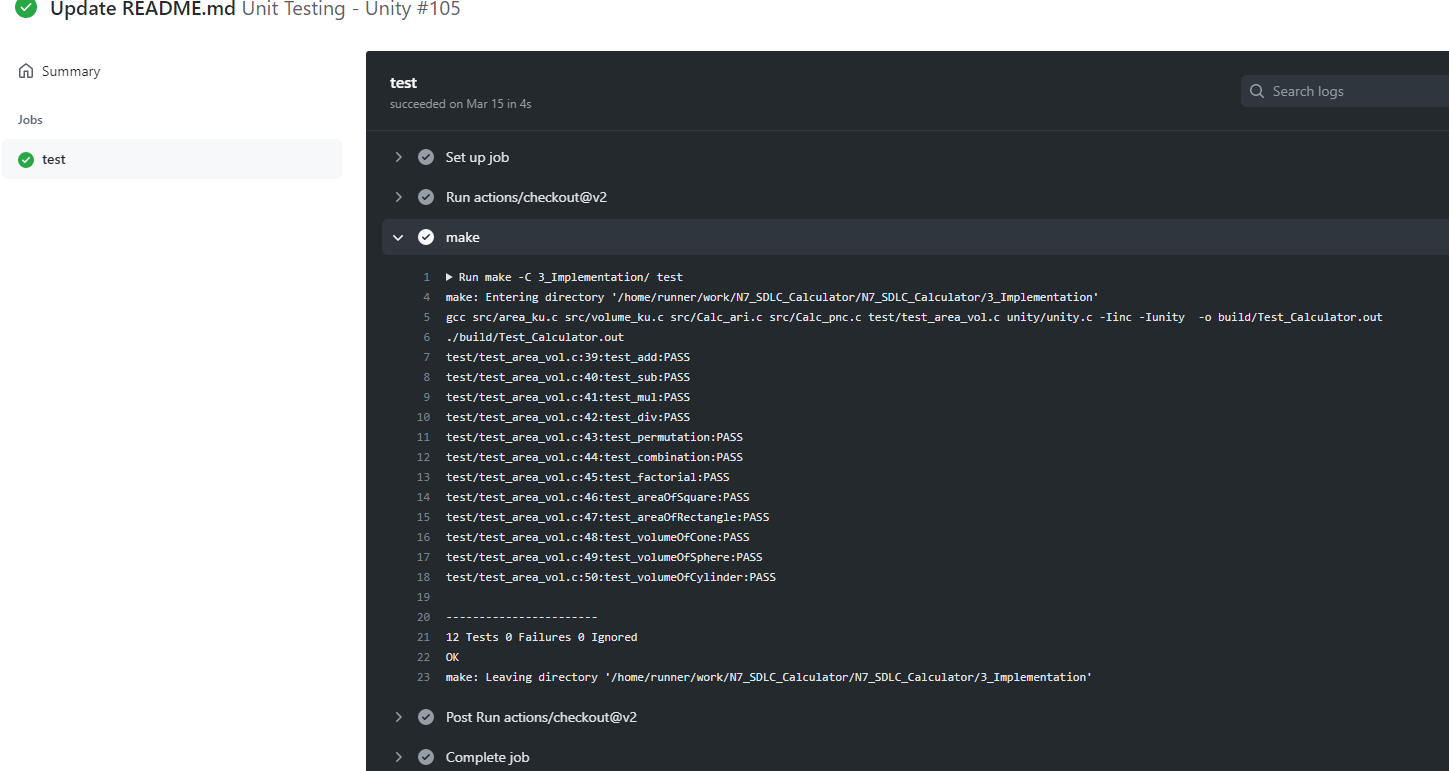


### 1.10.6 Unit Testing

#### 1.10.6.1 Setup for Unity Testing



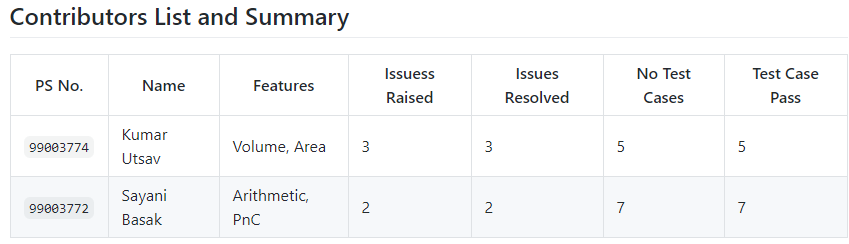
#### 1.10.6.2 Outcome of the Unity Testing



## 1.11 Individual Contribution & Highlights

* Arithmetic operations, Permutation and Combination features are 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.

Highlights



## 1.12 Summary

This project was our first step to work in coordination as a team while working towards our development of our individual skills.

Also, we’re unorganized at the beginning but after planning through V-Model we’re able to analyze and design each scenario and perfectly implemented whole project in the given duration. So, we also learned a great skill called time-management.

* Technical skills developed: -
  + Advanced C programming
  + Make File
  + Unit Testing through Unity
  + Version Control using GitHub
* Soft-Skills Developed: -
  + Team-work
  + Team-management
  + Time Management
  + Assertiveness

## **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.

## 1.14 Future Scope (If applicable)

Array based Matrix operations were not implemented, so that can also be designed. More graphical calculations and polynomial based functions can be added as the features in the calculator.

### 

# Mini project -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 is Python.

## 3.2 Project title: Mini Calculator

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

## 3.3 Topic and Subtopics

* The core steps of SDLC is being implemented.
* Introduction about SDLC
* Code Analysis
  + PEP8
* V Model
* Agile Model
* Git Hub

All the core-topics and sub-topics are implemented through V-Model.

## 3.4 Objectives:

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

### 3.4.1 Introduction

It is a Data Set problem that will allow users to retrieve data from data set. However, the input is in the form of the PS number, name, email id. The required output is all the candidate data.

### 3.4.2 4W & 1H

**What:**

\* We are preparing the master excel sheet to search and retrieve data from all the 5 excel sheets that are created.

\* It is used for easy search of a particular cell or data of a person

\* It provides information of every person details like bio, academics, health and personal details.

**When:**

\* Searching for person information

\* To get the contact information

\* To get the required details of that person educational qualification.

**Why:**

\* We are using to retrieve the data of an individual candidate from the excel workbook of 5 sheets where all the relevant data of 40 candidates is present.

\* We can easily access the details of that individual by giving some input such as name, Ps no and email id.

**Where:**

\* To check the information and bio of a person

\* Very useful during emergency times like health issues

\* We can also use it for knowing that person's bank details and other details related to his or her educational qualification.

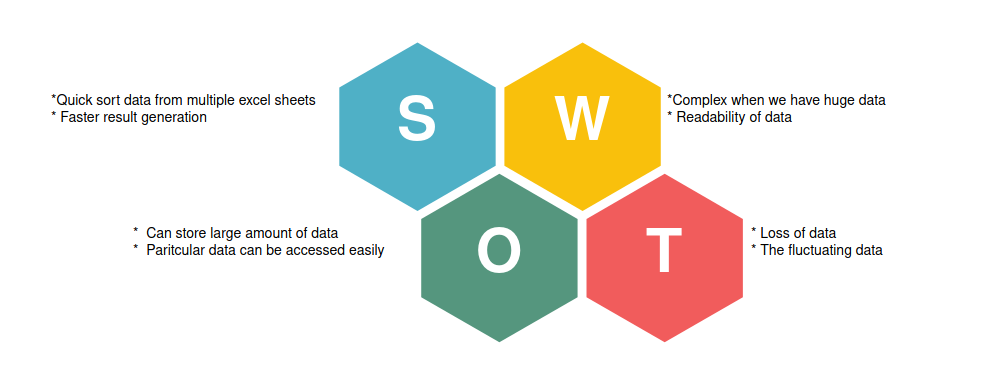
**How:**

\* Input: - We need to give 3 inputs such as Name, Ps No and Email Id.

\* Output: -We will get all the relevant information of that person whose name, Ps no and email id is given.

\* source: -All the relevant data will get copied in master sheet.

### 3.4.3 SWOT Analysis



## 3.5 Requirements:

### 3.5.1 High Level Requirement Analysis:

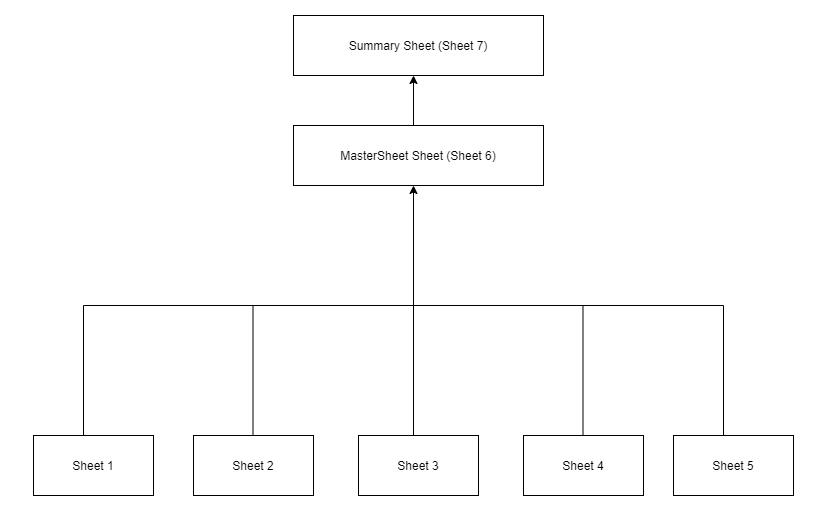
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirements** | **Description** | **Status** |
| H\_01 | Search data from sheet | Search all data from sheets when user gives the name, PS No. and email id to be searched | IMPLEMENTED |
| H\_02 | write data into new Sheet | Write all the data from different sheets in one Master Sheet | IMPLEMENTED |
| H\_03 | Extract data from sheets using given input | Write new required data in the excel file | IMPLEMENTED |

### 3.5.2 Low Level Requirement Analysis:

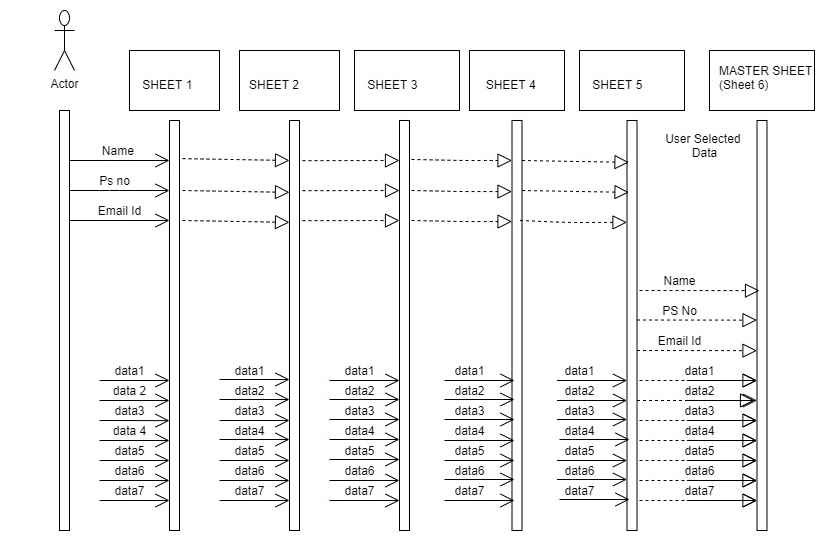
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirements** | **Description** | **Status** |
| L\_01 | Data Collection | worksheets contain the data of company details and academic details of users input | IMPLEMENTED |
| L\_02 | Each Sheet Contains 10 Column and 40 Rows | Each Sheet showing 10X40 format | IMPLEMENTED |
| L\_03 | Excel file format | the workbook file should be of. xslx format | IMPLEMENTED |
| L\_04 | Inputs | User can give multiple inputs like name, PS No, name and email id at once | IMPLEMENTED |
| L\_05 | Reading Data | Reading all 5 worksheets from workbook Search for specific data based on user specific inputs | IMPLEMENTED |
| L\_06 | Searching Data | Search for specific data based on user specific inputs | IMPLEMENTED |
| L\_07 | Master Sheet Contains Created | Master Sheet Contains 40X40 Format | IMPLEMENTED |

## 3.6 Design

### 3.6.1 High Level Diagram



### 3.6.2 Low Level Diagram



3.7 Test Plan

## 3.8 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 master sheet 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).

## **3.9** 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 mini project + Team members followed by the walkthrough”

## **3.10** Git Link

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

## 3.11 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

Snapshot of contribution list and summary is given from git

## **3.12** Summary

Technical:

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

Soft skills:

* Project management
* Conflict management.
* Time Management
* Assertiveness.

## 3.13 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.