––./

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](#_Toc69466954)

[Mini Project -1 SDLC (System Development life cycle) =>[Team] 5](#_Toc69466955)

[1.1 Modules Used 5](#_Toc69466956)

[1.2 Project title: Mini Calculator 5](#_Toc69466957)

[1.3 Topic and Subtopics 5](#_Toc69466958)

[1.4 Objectives 5](#_Toc69466959)

[SWOT Analysis 5](#_Toc69466960)

[4W and 1H 6](#_Toc69466961)

[1.5 Requirements: 6](#_Toc69466962)

[High Level Requirements: 6](#_Toc69466963)

[Low Level Requirements: 6](#_Toc69466964)

[1.5 Design 7](#_Toc69466965)

[1.5.1 High Level Diagram of the Calculator 7](#_Toc69466966)

[1.5.2 Permutation and Combination 8](#_Toc69466967)

[1.5.3 Arithmetic operations: 9](#_Toc69466968)

[1.6 Test Plan 10](#_Toc69466969)

[1.6.1 High Level Test Plan 10](#_Toc69466970)

[1.6.2 Low Level Test Plan 10](#_Toc69466971)

[1.7 Implementation Summary 10](#_Toc69466972)

[1.8 Video Summary 11](#_Toc69466973)

[1.9 Git Link 11](#_Toc69466974)

[1.10 Git Dashboard 11](#_Toc69466975)

[1.10.1 Badges 11](#_Toc69466976)

[1.10.2 Git inspector summary 12](#_Toc69466977)

[1.10.3 Build 13](#_Toc69466978)

[1.10.4 Code quality 14](#_Toc69466979)

[1.10.5 Git issues 16](#_Toc69466980)

[1.10.6 Unit Testing 16](#_Toc69466981)

[1.11 Individual Contribution & Highlights 18](#_Toc69466982)

[1.12 Summary 18](#_Toc69466983)

[1.13 Individual Contribution 19](#_Toc69466984)

[1.14 Challenges faced and how were they overcome 19](#_Toc69466985)

[1.14 Future Scope (If applicable) 19](#_Toc69466986)

[Mini project -3 Embedded C => [Team] 20](#_Toc69466987)

[Modules Used: 20](#_Toc69466988)

[Topics and Subtopics: 20](#_Toc69466989)

[Objectives & Requirements: 20](#_Toc69466990)

[Components Used: 20](#_Toc69466991)

[Individual Contribution & Highlights 23](#_Toc69466992)

[Summary 23](#_Toc69466993)

[Challenges faced and how were they overcome 23](#_Toc69466994)

[Mini Project -3 Python Programming (Individual) 24](#_Toc69466995)

[3.1 Modules Used 24](#_Toc69466996)

[3.2 Project title: Mini Calculator 24](#_Toc69466997)

[3.3 Topic and Subtopics 24](#_Toc69466998)

[3.4 Objectives: 24](#_Toc69466999)

[3.4.1 Introduction 24](#_Toc69467000)

[3.4.2 4W & 1H 24](#_Toc69467001)

[3.4.3 SWOT Analysis 25](#_Toc69467002)

[3.5 Requirements: 26](#_Toc69467003)

[3.5.1 High Level Requirement Analysis: 26](#_Toc69467004)

[3.5.2 Low Level Requirement Analysis: 26](#_Toc69467005)

[3.6 Design 27](#_Toc69467006)

[3.6.1 High Level Diagram 27](#_Toc69467007)

[3.6.2 Low Level Diagram 27](#_Toc69467008)

[3.7 Implementation Summary: 28](#_Toc69467009)

[3.8 Summary: 28](#_Toc69467010)

[3.9 Challenges faced and how were they overcome: 28](#_Toc69467011)

[Mini Project -4 Kernel Programming and Device Drivers(Individual) 29](#_Toc69467012)

[4.1 Modules Used 29](#_Toc69467013)

[3.2 Project title: Mini Calculator 29](#_Toc69467014)

[3.3 Topic and Subtopics 29](#_Toc69467015)

[3.4 Objectives: 30](#_Toc69467016)

# Mini Project -1 SDLC (System Development life cycle) =>[Team]

## 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

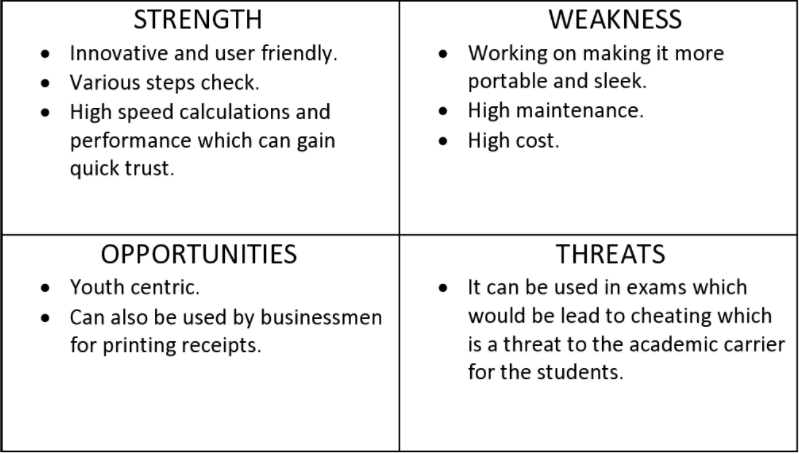
* The core steps of SDLC is being implemented.
  + The features of Calculator are implemented.
  + The testing has been done for each function.
* Introduction about SDLC
* C Programming
* Code Analysis
  + CPP Check
  + Valgrind
* Testing
  + Unity Testing
* Makefile
* V Model
* Agile Model
* Git Hub

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

## 1.4 Objectives

Designing a basic calculator that performs basic functions as well as some specific functions as per requirements.

## SWOT Analysis



### 4W and 1H

**Who:** Student and businessman.

**What:** Smart Scientific Calculator.

**When:** For fast and effective way to complete calculations.

**Where:** Statics comparing the previous data with present data.

**How:** Easy to user interface.

## 1.5 Requirements:

### 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. |

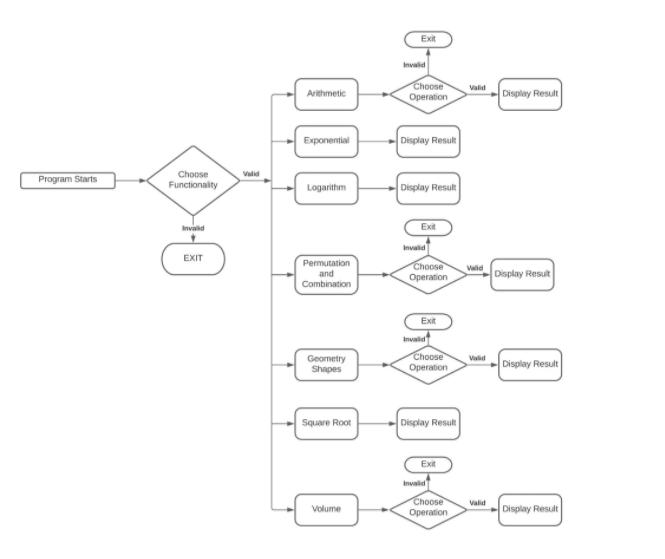
### Low Level Requirements:

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

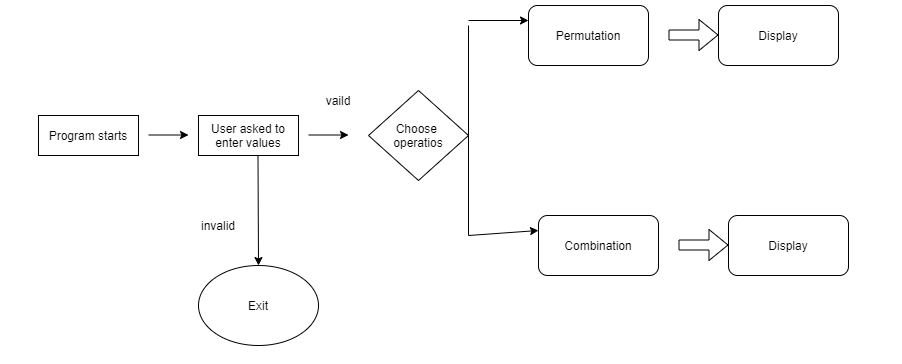
“System Level and subsystem level UMLs – Structural and Behavioral”

### 1.5.1 High Level Diagram of the Calculator

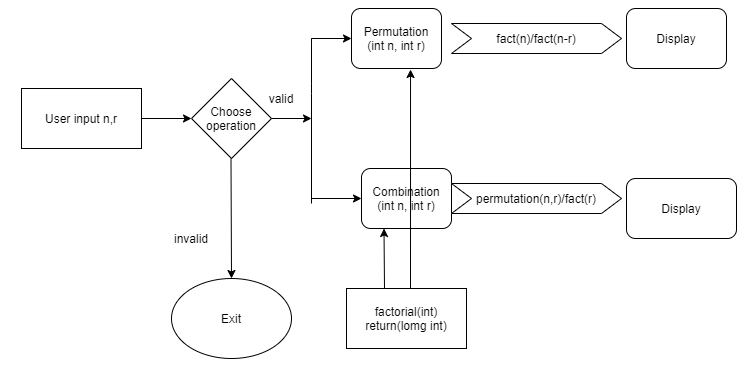


### 1.5.2 Permutation and Combination

High Level Requirement:

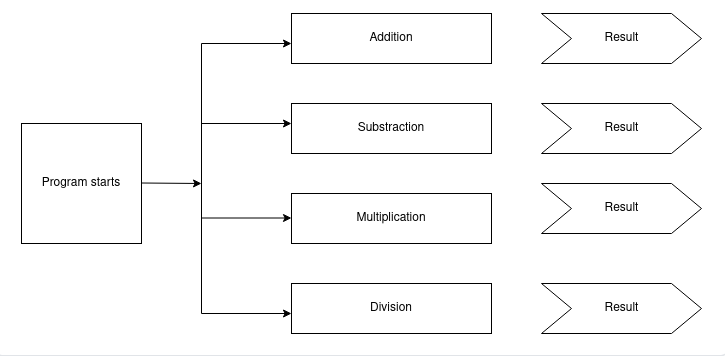


Low Level Requirement:

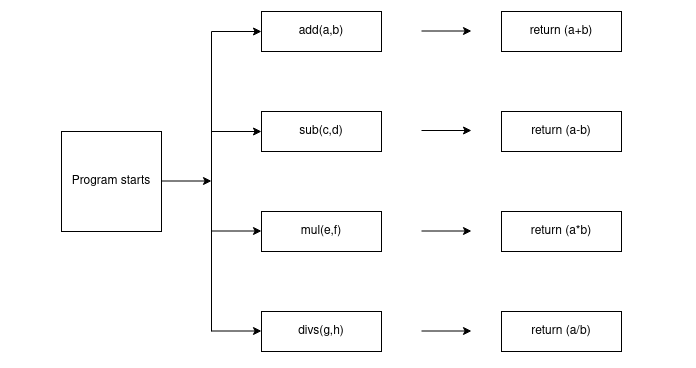


### 1.5.3 Arithmetic operations:

High Level Requirement:

****

Low Level Requirement:



## 1.6 Test Plan

### 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 | Subtraction: All the numbers we add subtract 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.

Implementation folder had all source files, header files, test files for different features of the calculator such as Basic Arithmetic, Square root, cube root, exponent, logarithm, etc.

Here, **inc** folder holds all the header files with “.h” extension which contains prototype of all functions, structure definition, macro definition and definition of all the enumerators.

The **src** folder holds all the source files with “.c” extension which has definitions of all the functions whose prototype is defined in header files.

Other than these folders, there is also a **unity** folder which holds prototypes and definition of the standard unity test case functions.

Also, there is a **Makefile** which builds, debugs using valgrind, check static and dynamic code quality, performs overall unit testing for all the codes together with the execution of single commands based on different defined targets.

## 1.8 Video Summary

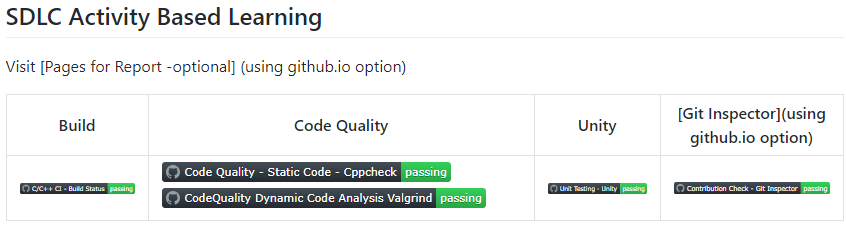
“Please upload a short video on the repo for the walkthrough of the project (Team/Individual) less than 7min and less than 30MB File Size. Start is the Standard opening slide with title of miniproject + Team members followed by the walkthrough”

## 1.9 Git Link

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

## 1.10 Git Dashboard

### 1.10.1 Badges



**Summary**

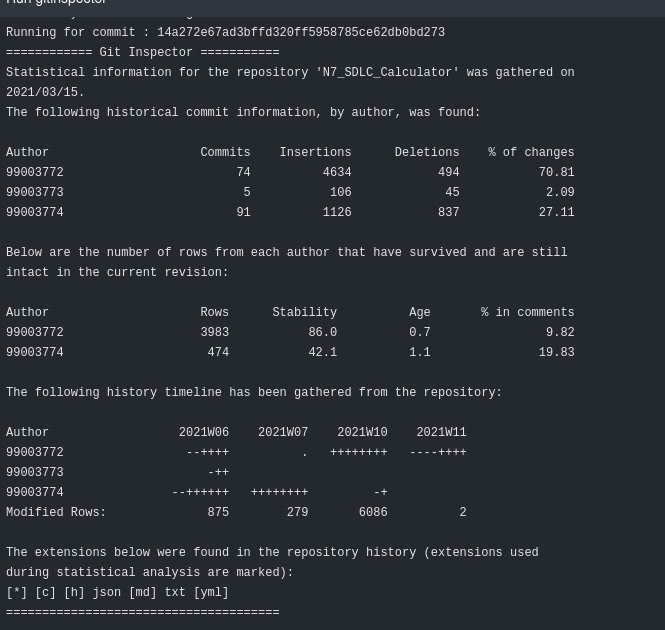
In this project, we mainly-focused on how to design a calculator which is bit different and cost-effective as compared to other calculators present in the market.

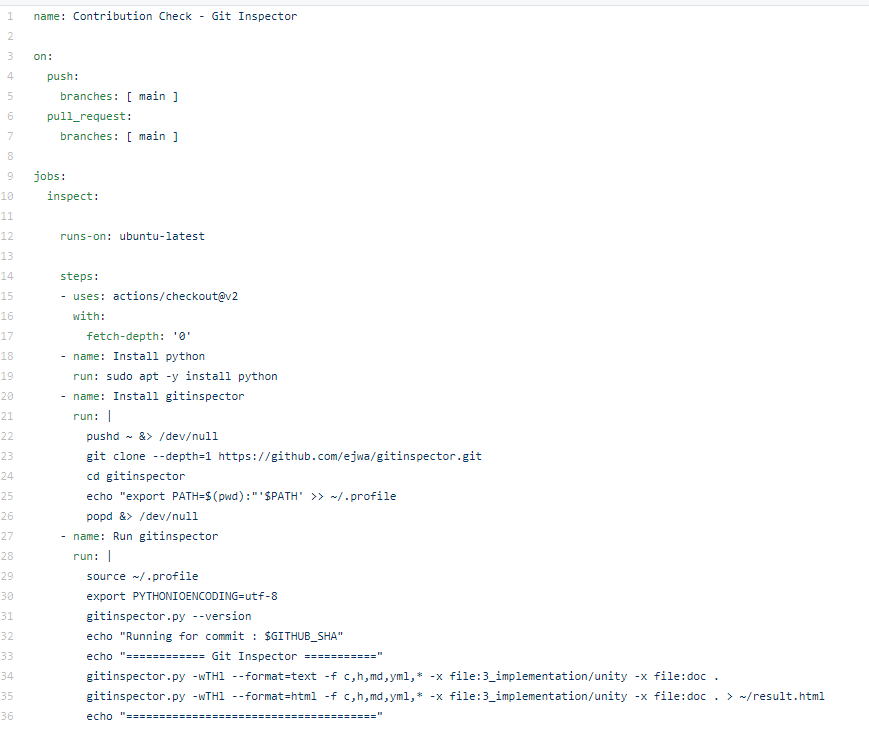
So, for this we first analyzed all the other calculators from low-end feature low cost calculator to high-end feature high-cost calculator and prepared a list of features to include in our modified cost-effective more featured calculator.

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.

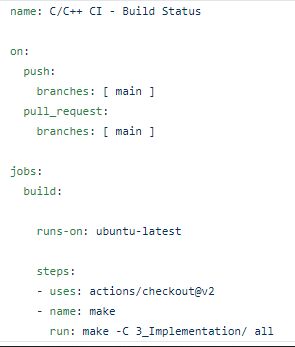
### 1.10.2 Git inspector summary



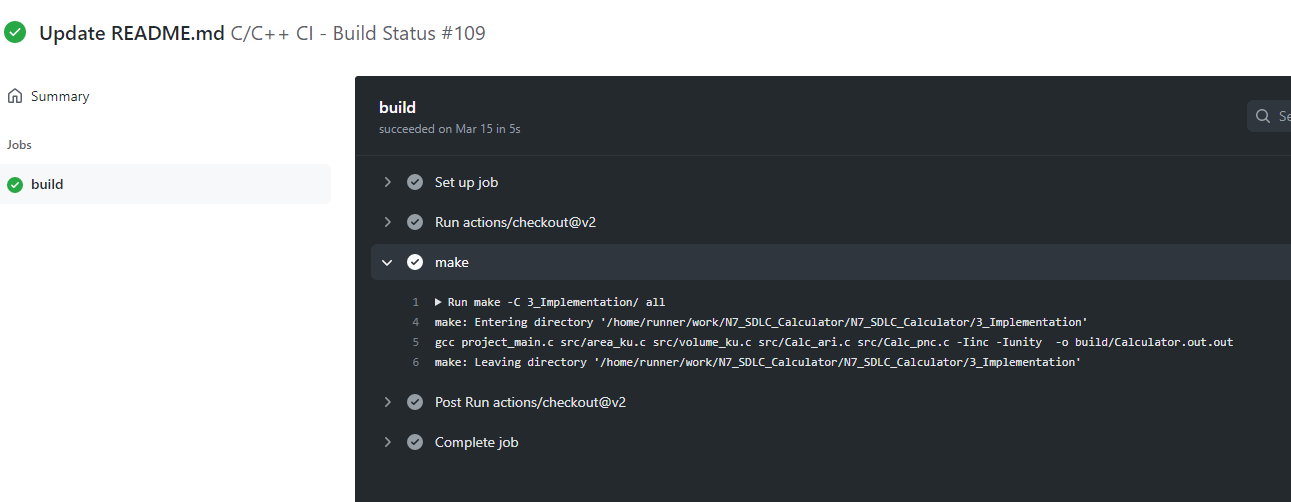


### 1.10.3 Build

#### 1.10.3.1 Setup for Build



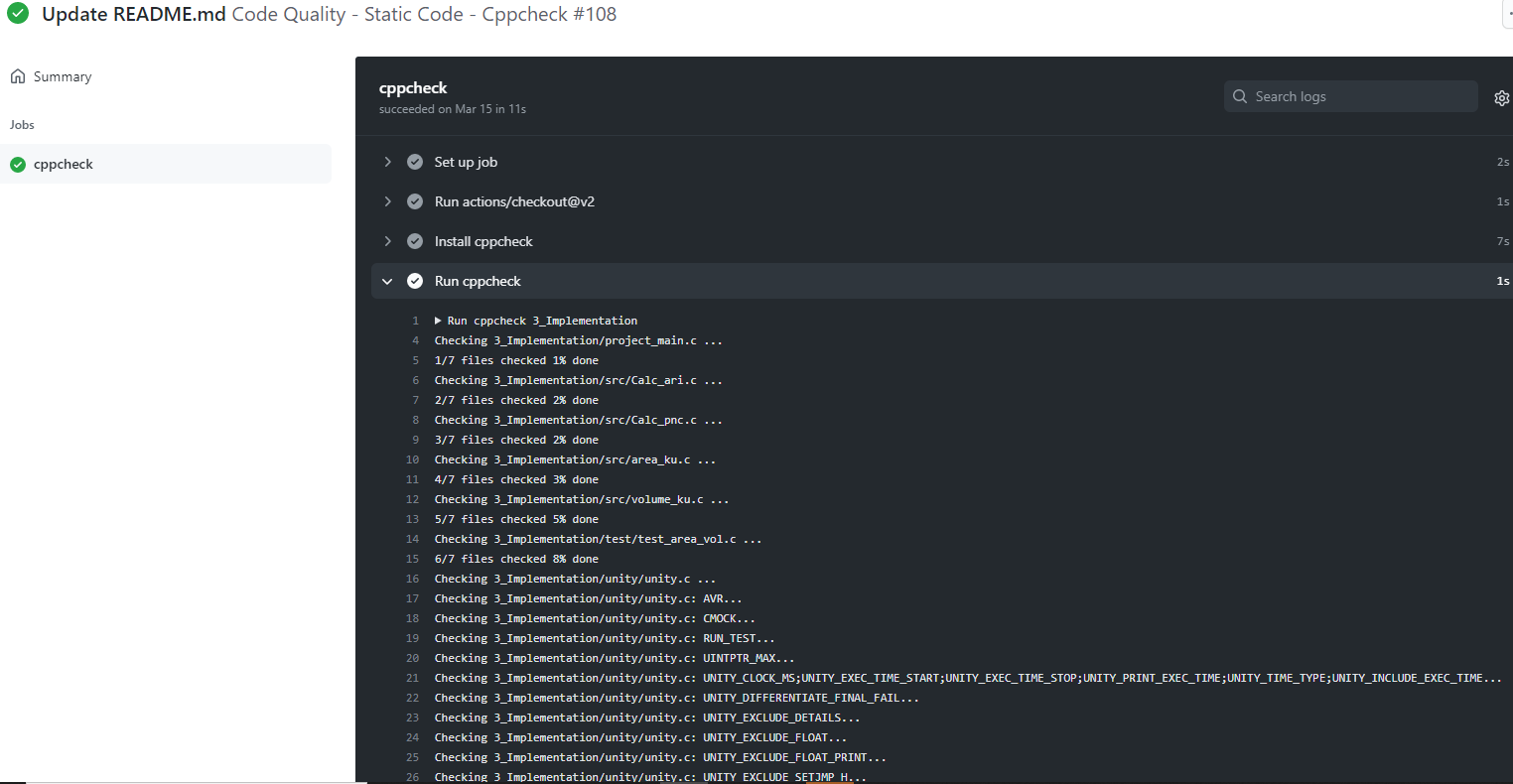
#### 1.10.3.2 Outcome of the Build



### 1.10.4 Code quality

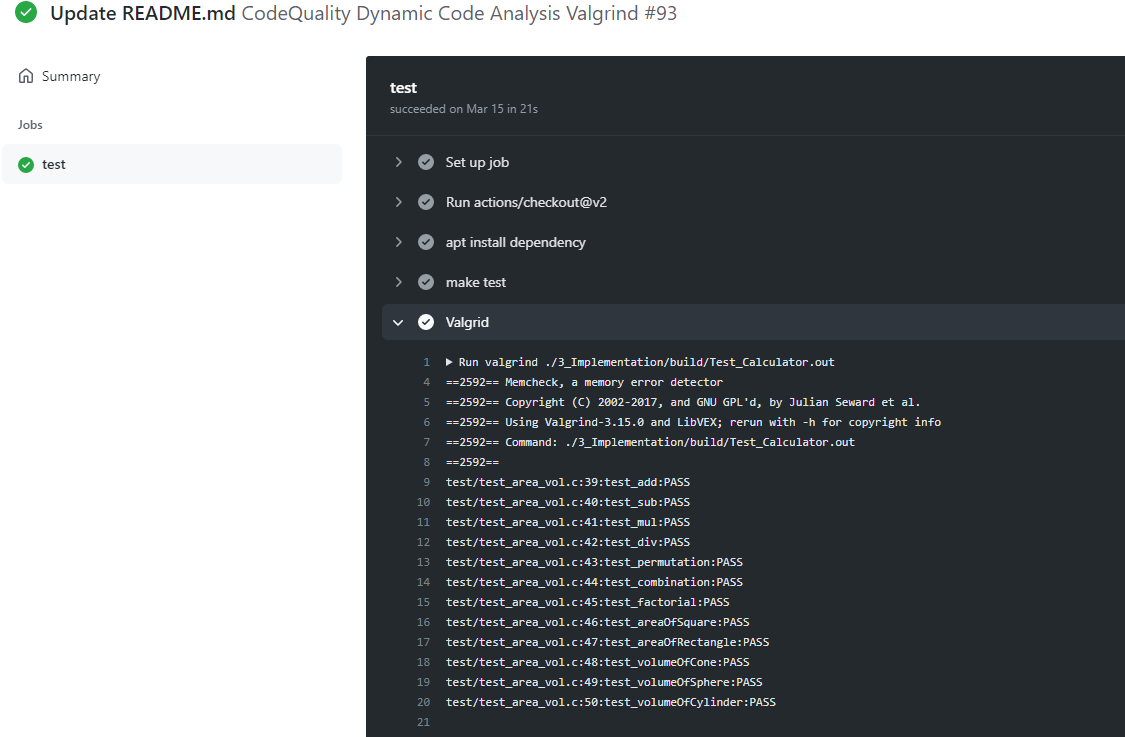
#### 1.10.4.1 Setup for Static Code Quality



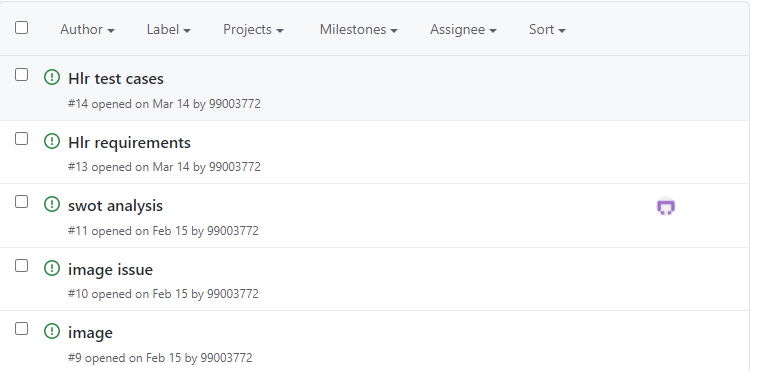


#### 1.10.4.2 Outcome of the Dynamic Cody Quality



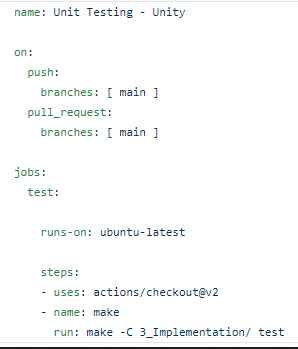


### 1.10.5 Git issues

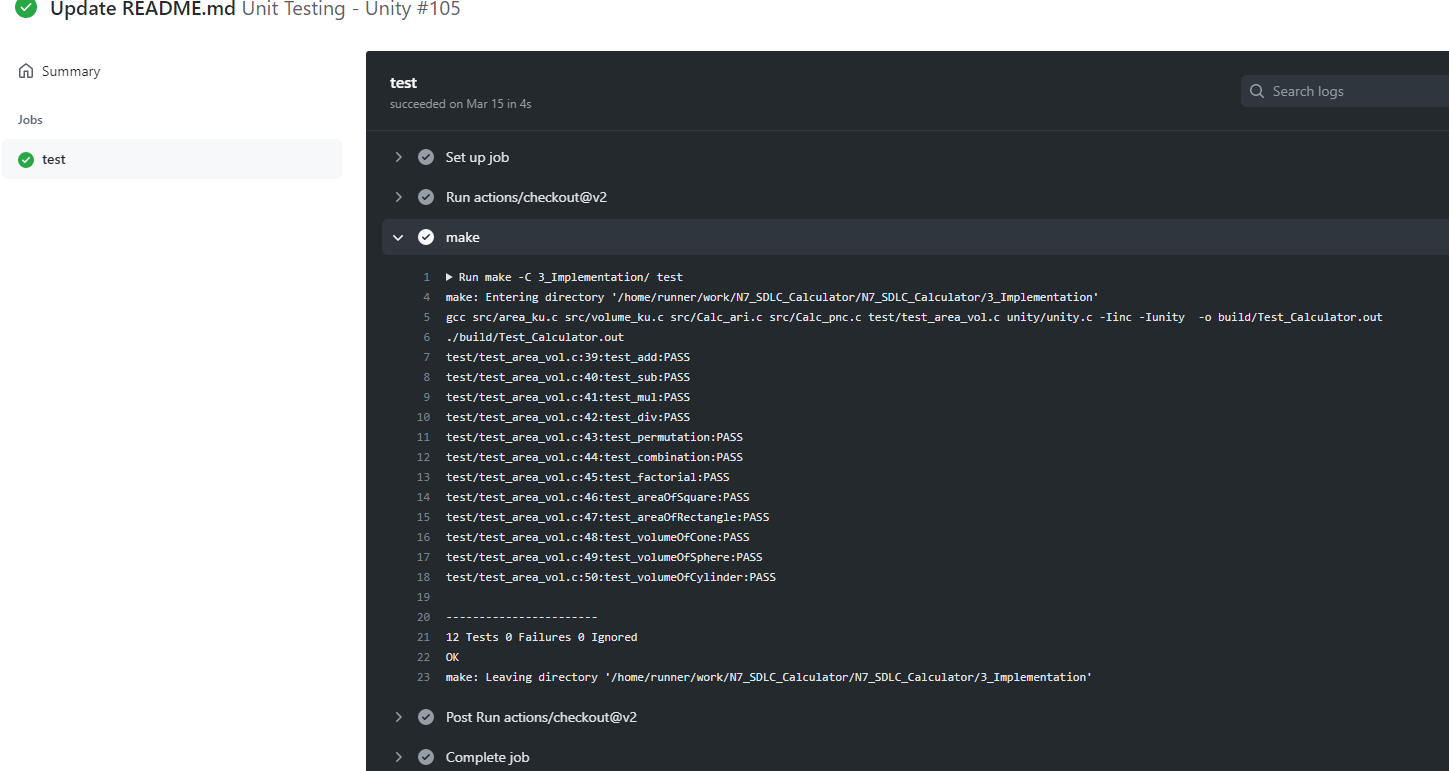


### 1.10.6 Unit Testing

#### 1.10.6.1 Setup for Unity Testing



#### 1.10.6.2 Outcome of the Unity Testing

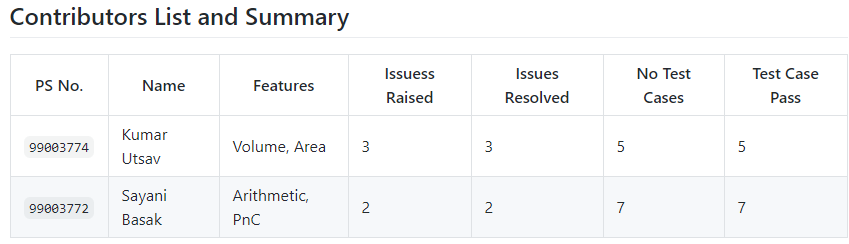


## 1.11 Individual Contribution & Highlights



* 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 Individual Contribution

My Contribution to this project is that I have implemented the features of finding area and perimeter of geometrical shapes. I have also implemented the features of Arithmetic Operations.

I have written my code in main.c in src folder under implementation. Accordingly, I have added main.h in inc folder under implementation. I have modified the test\_calculator\_operations.c where I have written my test cases.

I have modified Makefile so that program should be able to build as well as it is able to perform unity based unit testing.

## **1.14** Challenges faced and how were they overcome

* Initially I was unable to run the test cases but now it’s running.
* Differentiation of high level and low level.
* Committing to GitHub, pull and push in GitHub.
* Cpp check and Unity testing.
* Some workflow problems were there but now it has been solved.
* Makefile problem while running in windows but it worked with linux as there persisted an existence problem but it got resolved in linux.
* Git inspector was not working initially but now it’s working.
* Valgrind was not working previously because there was some issue in makefile but now it’s working fine after resolving makefile.

## 1.14 Future Scope (If applicable)

1) Features like matrix operations can be added.

2) Features such as Physics operations (Acceleration, capacitive reactance, circular velocity, Coulomb’s Law, Gravitational force, projectile motion) can be added.

3) Features like AP, GP sum (AP sum, GP sum) can be implemented in the future.

4) The price of the product is less than other products with same features that are available in the market.

5) The product will also work in banking sectors and other government sectors where they want low price, minimum features, handy products.

### 

# Mini project -3 Embedded C => [Team]

## Modules Used:

Modules used in this project are Embedded Systems and Embedded C Programming and was implemented on the hardware STM32.

## Topics and Subtopics:

* Driver API Development
* GPIO
* ADC
* SPI, UART, I2C
* External interrupt
* Debugging using STM Board
* Driver Development (Hardware Abstraction Level- HAL)

* + GPIO
  + ADC
  + External Interrupt
  + Debugging using STM Board

## Objectives & Requirements:

To implement different CAR Module features using STM32f407VG Microcontroller featuring 32-bit ARM-M4 with FPU core.

* Car Module Features:
* Power window feature [ Done by me]
* Sunroof control feature [ Done by me]
* Interior Lighting using PIR Motion detection sensor
* Door Lock feature
* Seat control feature
* Wiper Control feature

## Components Used:

* STM32f407VG Microcontroller
* Breadboard
* LED
* LDR Sensor
* Soil Sensor
* PIR Motion detection sensor
* RGB Color Sensor
* Potentiometer Sensor
* Ring Buzzer Sensor
* Jumper Wires

Requirements

High Level Requirements:

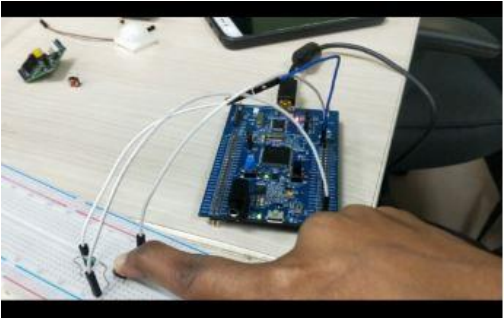
|  |  |  |  |
| --- | --- | --- | --- |
| **Serial No.** | **Requirements** | **Description** | **Status** |
| 1 | Power window | User manually opens or closes the window through a button press. | Implemented |
| 2 | Sunroof control |  |  |

Low Level Requirements:

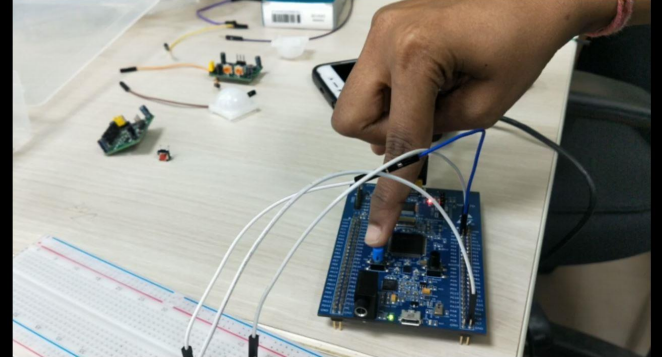
|  |  |  |  |
| --- | --- | --- | --- |
| **Serial No.** | **Requirements** | **Description** | **Status** |
| 1 | When car window(s) is/are open. | GPIO pin PD13 is set and the LED on the discovery board glows. | Implemented |
| 2 | When car window(s) is/are closed. | GPIO pin PD13 is reset and the LED on the discovery board is off. | Implemented |
| 3 |  |  |  |
| 4 |  |  |  |

Design:

Power Window



Sunroof Control



Test Plan

High Level Test Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Serial No.** | **Description** | **Expected Input** | **Expected Output** | **Actual Output** | **Type of Test** |
| 1 | Power Window | Window is open | Led in the car starts glowing. | Green LED on the discovery board starts glowing | Requirement based |
| 2 |  |  |  |  |  |

Low Level Test Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Serial No.** | **Description** | **Expected Input** | **Expected Output** | **Actual Output** | **Type of Test** |
| 1 | Power Window | Window is open | Led in the car starts glowing. | GPIO PIN PD 13 is set and the green LED on the discovery board glows. | Requirement based |
| 2 | Power Window | Window is close | Led in the car is off. | GPIO PIN PD 13 is reset and the green LED on the discovery board stops glowing. | Requirement based |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |

## Implementation Summary

Multiple features of the car using STM32f407 discovery board has been implemented. Here we have assigned certain pins for GPIO input and output by board configuration in STM32CubeIDE. Then we have generated a high-level abstraction code and then we use certain GPIO HAL functions, ADC HAL functions and configured.

We have also used the same LED’s for different feature indications.

Features that I have implemented and pin configurations are as follows: -

* Power Window:
* PD13: Denotes the window(s) status (SET (1): Open, RESET (0): Closed)
* Sunroof Control:
* PD14: Denotes the window(s) status (SET (1): Close, RESET (0): Open)

### Summary

In this project, the features that has been selected are namely:

* Power window
* Sunroof Control
* Interior Lighting
* Door Lock/Unlock
* Seat Control, and
* Wiper Control

Out of these six, the features that I have implemented are:

* Power Window, and
* Door lock/Unlock

For power window control, I have used a switch along with the discovery board. When the switch is pressed GPIO PIN PD13 is set and the green LED on the discovery board turns on indicating that the car window(s) is/are open. Similarly, when the switch is released the GPIO PIN PD 13 is reset and the green LED on the discovery board turns off indicating that the window(s) is/are close.

For door lock/unlock feature, I have used the discovery board along with a buzzer and a switch. When the switch is pressed, the GPIO PIN PD 14 is reset and the buzzer starts buzzing indicating that the door(s) is/are open. Similarly, when the switch is released, the GPIO PIN PD 14 is set and the buzzer stops buzzing indicating the door(s) is/are closed.

### Challenges faced and how were they overcome:

* At first, we were facing issues with the STN32 Discovery Board because of which we needed to change the board.
* There were even few issues with our code even which were eliminated after going through the entire code properly and debugging.
* The makefile that we created was not building the code but that was also eliminated after proper debugging

# Mini Project -3 Python Programming (Individual)

## 3.1 Modules Used

Modules used in this project is Python.

## 3.2 Topic and Subtopics

### Core-Topics

* Basic Python

### Sub-Topics

* Data Types
* Arithmetic operations
* String operations
* Control structures

### Sub-Topics

* If-else statements
* While loops
* For loops
* Nested Loops
* Functions

### Sub-Topics

* Defining custom functions
* Pass by value
* Pass by reference
* Introduction to Library functions
* Data Structures

### Sub-Topics

* List
* Tuple
* Set
* Dictionary
* Exceptional Handling

### Sub-Topics

* Try, Except
* Finally
* pass keyword
* Date and time Library

### Sub-Topics

* Date comparison
* Execution time analysis
* Excel file library (openpyxl)

### Sub-Topics

* Open multiple excel files
* Reading multiple excel files sheet by sheet.
* Mastersheet creation
* Writing to excel file

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.

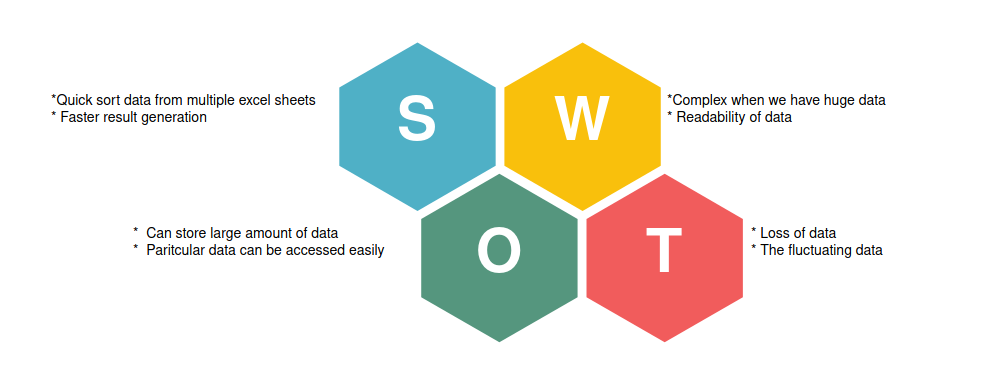
**How:**

\* 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 |

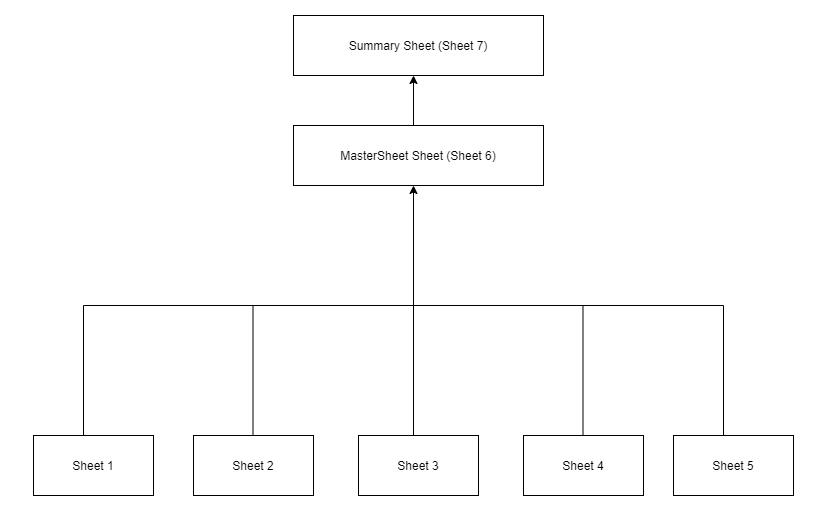
### 3.5.2 Low Level Requirement Analysis:

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

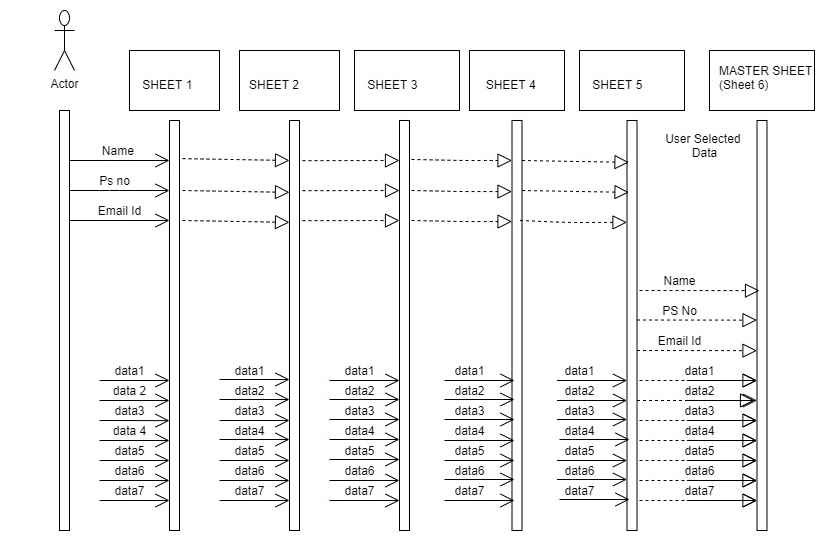
## 

## 3.6 Design

### 3.6.1 High Level Diagram



### 3.6.2 Low Level Diagram



## 

## 3.7 Test Plan:

### 3.7.1 High Level Testing:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Id** | **Description** | **Expected Input** | **Expected Output** | **Actual Output** | **Type of Test** |
| 1 | To Access multiple workbooks stored in different locations | To Access multiple workbooks stored in different locations | Choice:2  Number of worksheets=2, C:\Users\99003731\Desktop\apps\New folder2\share\_axis.xlsx  C:\New folder1\share\_asian.xlsx | ‘<Workbook 1>’  ‘<Workbook 2>’ | ‘<Workbook1>’  ‘<Workbook2>’ |
| 2 | To Read multiple worksheets of a workbook | To Read multiple worksheets of a workbook | Choice:2  Number of worksheet =1,  Enter file path: D:\New folder\share\_adani.xlsx | ‘<ADANIPORTS>,  <Sheet 1>’ | ‘<ADANIPORTS>,  <Sheet 1>’ |
| 3 | To Search by date | Choice:1 (to take excel files path from file)  Choice1 :1 (To search by date)  Number of dates:2  Date :2/2/200  Date: 3/1/2000 | All the Data of 2/2/2000 and 3/1/2000 copied to master-workbook | All the data of 2/2/2000 and 3/1/2000 copied to master-workbook | Requirement based |

### 3.7.2 Low Level Testing:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Id** | **Description** | **Expected Input** | **Expected Output** | **Actual Output** | **Type of Test** |
| 1 | To access multiple worksheets by provide their path one by one | Choice: 2  Number of worksheets = 3  Enter file path: C:\Users\99003731\Desktop\apps\New folder2\share\_axis.xlsx  Enter File path:  C:\New folder1\share\_asian.xlsx  Enter File path:  D:\New folder\share\_adani.xlsx   |  | | --- | |  | | ‘<Workbook 1>’  ‘<Workbook 2>’  ‘<Workbook 3>’ | ‘<Workbook1>’  ‘<Workbook2>’  ‘<Workbook 3>’ | Requirement based |
| 2 | To access multiple worksheets whose paths are present in text file | Choice:1  File.txt | Starts reading every worksheet | Starts reading every worksheet | Scenario based |
| 3 | Searching data by only dates | Choice:1 (to take excel files path from file)  Choice1 :1 To search by date  Number of dates:2   1. 29/2/200   31/1/2000 | All the Data of 29/2/2000 and 31/1/2000 copied to master-workbook | All the data of 29/2/2000 and 31/1/2000 copied to master-workbook | Requirement based |
| 4 | Search Data by only ShareID | Choice:1 (to take excel files path from file)  Choice1 :2 To search by ShareID  Number of shares:2   1. HD855   AP575 | All the Data related to ShareID’s HD855, AP575 is copied to master-workbook | All the Data related to ShareID’s HD855, AP575 is copied to master-workbook | Requirement based |
| 5 | Search data by both date and ShareID | Choice:1 (to take excel files path from file)  Choice1 :3 To search by both ShareID  Number of dates: 2   1. 4/2/2000 2. 17/1/2000   Number of shares:1  HD855 | All the data on 4/2/2000,  17/1/2000 for  ShareID HD855 is copied to master-workbook | All the data on 4/2/2000,  17/1/2000 for  ShareID HD855 is copied to master-workbook | Requirement based |
| 6 | To Skip date which is not present in the sheets and search for all the other data | Choice:1 (to take excel files path from file)  Choice1 :1 To search by date  Number of dates:2   1. 29/2/200   26/1/2000 (Not a valid date) | Only data of 29/2/2000 copied to master-workbook | Only data of 29/2/2000 copied to master-workbook | Scenario based |
| 7 | To Skip ShareID which is not present in the sheets and search for all the other data | Choice:1 (to take excel files path from file)  Choice1 :2 To search by ShareID  Number of shares:2   1. AX700 (Invalid ShareID)   AP575 | Only data of AP575 copied to master-workbook | Only data of AP575 copied to master-workbook | Scenario based |
| 8 | To Skip date/ShareID which is not present in the sheets and search for all the other data | Choice:1 (to take excel files path from file)  Choice1 :3 To search by both ShareID  Number of dates: 2   1. 4/2/2000 2. 17/1/2000   Number of shares:1  AB878 | Empty Master-workbook | Empty Master-workbook | Scenario based |

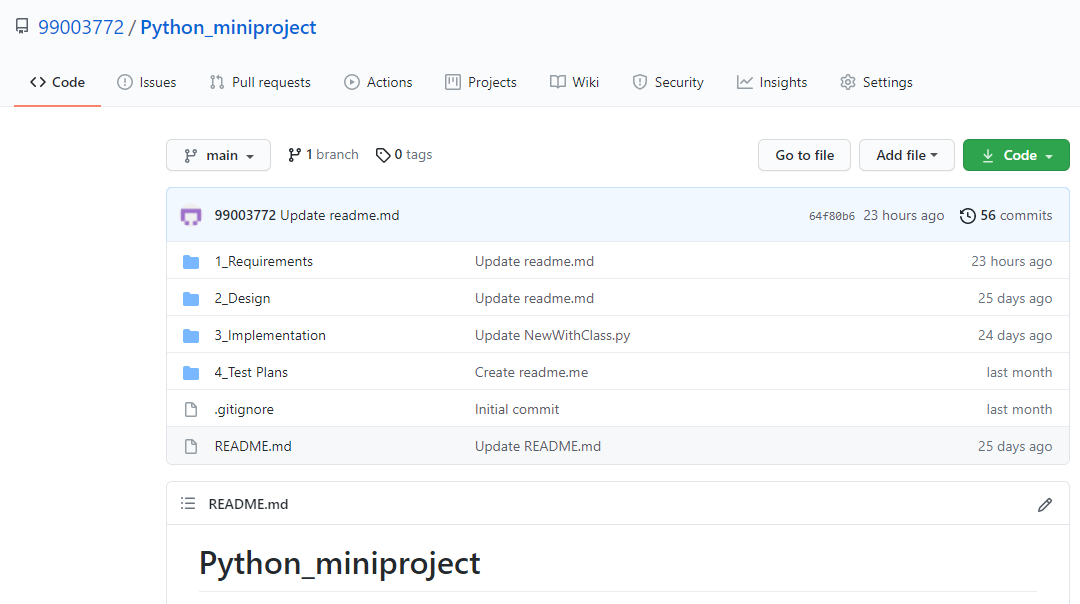
## 3.8 Implementation Summary

Aim of this project is to extract the data present in 5 different spreadsheets in one excel file as required by the user. The excel sheet consists of 5 different spreadsheets. The user defines the data that needs to be searched based on Area Name. The python program then reads the data corresponding to the 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, it will create a summary sheet which will count the individual number of date and the total number of data.

## 3.9 Git Link

<https://github.com/99003772/Python_miniproject.git>

## 3.10 Git Dashboard



## 3.11 Summary

Python is one of the very powerful language present in the industry. However, despite the number of cons developer still use python to develop end-to-end, quality and robust web application. It is a great programming language with some excellent advantage such as easy use code lines, smooth maintenance and easy debugging.

Python is also used of automation and this is reflected in the mini project also. An excel file with 5 data sheets is given and based upon user input the data corresponding to the input will be searched from all 5-different sheet and will be written in the master sheet, and in the end, it will create a summary sheet which will count the individual number of date and the total number of data.

## 3.12 Challenges faced and how were they overcome

Some of the challenges faced are as follow:

* Working with, and handling excel sheet is a new topic and there were lots of issues initially.
* Understanding the concept of pandas and openpyxl modules and what module to use was challenging in the beginning, as many times excel sheet was getting corrupted.
* Writing the extracted data into the master sheet was also challenging.

# Mini Project -4 Kernel Programming and Device driver =>[Individual]

## 4.1 Modules Used

The modules used in this are Linux and Kernel Device Drivers.

## Topic and Subtopics

* Basic Linux commands.
* Qemu Based Emulation.
* Creation of SD card.
* Building custom Kernel.
* Cross Compilation.
* Static and dynamic libraries.
* System calls.
* Adding system calls in kernel space.
* Invoking system calls from user space.
* Kernel modules.
* In-Tree modules: Dynamic.
* In-tree modules: static.
* Basics of Kernel Device Drivers
* Registering Char Driver.
* Kernel Data Structure.
* Kfifo API.
* List API.
* IPC Kernel
* Concurrency.
* Kernel Threads.
* Locking and Synchronization.
* Semaphores.
* Mutex.
* Spinlocks.
* Wait queues.
* IOCTL.
* Driver model.

## Objectives

The main objective of this module is to apply the concepts of Linux kernel, kernel device drivers to develop:

* Custom kernel.
* Create char drivers.
* Developing cross compiled code for target qemu.
* Creating own system calls.
  1. Requirements
* Basic Linux commands.
* Programming in Linux Environment.
* Custom kernel.
* zImage
* vexpress-v2p-ca9.dtb
* rootfs.img
* Operating system Basics.
* IPC concepts.
* Concurrency.
* File handling using system calls.
* Virtual Memory concept.

## 4.5 Implementation Summary:

### 4.5.1 Hands-on Activity that are implemented are as follow:

* Register char driver
* Register file operations
* Device Create, Class Create
* Read, write operations using global buffer
* Read, write operations using kfifo
* ioctl operations, returning length/remaining space, reset operation
* ioctl operations - filling length/remaining space in structure
* synchronization in char driver - using wait queue

### 4.5.2 User space code:

* simple read, write
* multiple read, multiple write
* Userspace code for IOCTL operations

### 4.5.3 kthread examples:

* simple two threads
* Race condition scenarios
* Mutual exclusion using semaphore, mutex, spinlock
* Synchronization using semaphores, wait queues
* Device Tree based platform driver code -- dummy UART

### 4.5.4 Activity that are implemented are as follow:

* System calls -- echo back the given string.
* System calls—traverse process list print pid and ppid.
* System calls—length of string.
* System calls—taking simple parameter.
* IOCTL operation traverse the list.

## 4.6 Git Link:

<https://github.com/99003763/Embedded-linux-and-Kernel-Programming.git>

## 4.7 Summary

4.8 Challenges faced and how were they overcome