./

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](#_Toc55470819)

[Miniproject -1 [Team/Individual] 4](#_Toc55470820)

[Module/s 4](#_Toc55470821)

[Topic and Subtopics 4](#_Toc55470822)

[Objectives & Requirements 4](#_Toc55470823)

[Design 4](#_Toc55470824)

[Test Plan 4](#_Toc55470825)

[Implementation Summary 4](#_Toc55470826)

[Video Summary 4](#_Toc55470827)

[Git Link 4](#_Toc55470828)

[Git Dashboard 4](#_Toc55470829)

[Summary 4](#_Toc55470830)

[Individual Contribution & Highlights 5](#_Toc55470831)

[Summary 5](#_Toc55470832)

[Challenges faced and how were they overcome 5](#_Toc55470833)

[Future Scope (If applicable) 5](#_Toc55470834)

[Miniproject -2 [Team/Individual] 6](#_Toc55470835)

[Module/s 6](#_Toc55470836)

[Topic and Subtopics 6](#_Toc55470837)

[Objectives & Requirements 6](#_Toc55470838)

[Design 6](#_Toc55470839)

[Test Plan 6](#_Toc55470840)

[Implementation Summary 6](#_Toc55470841)

[Git Link 6](#_Toc55470842)

[Git Dashboard 6](#_Toc55470843)

[Summary 6](#_Toc55470844)

[Individual Contribution & Highlights 6](#_Toc55470845)

[Summary 6](#_Toc55470846)

[Challenges faced and how were they overcome 6](#_Toc55470847)

# Miniproject -1 [Team]

## **Module:**

“SDLC (System Development Life Cycle)”

### Topic and Subtopics:

* ***V-Model***
* C Programming
* Makefile
* Unit Testing
* Version Control (via GitHub)
* ***Agile-Methodology***
* Theme
* Epic
* User- Story

All these topics had been implemented using V-model.

## **Objective:**

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

## **Requirements:**

***High Level Requirement:***

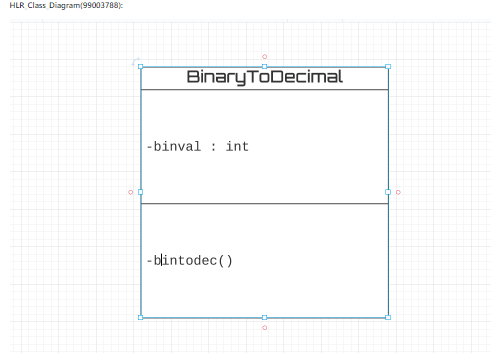
1. Maximum input digits a user can enter is up to 12 digits.
2. Arithmetic Operations
3. Exponential and Logarithmic Operations.
4. Square roots and powers.
5. Permutation and Combinations.
6. Finding area of different geometrical shapes like circle, square, rectangle, triangle.
7. Finding volume of different geometrical shapes like cube, cylinder, cone, sphere.
8. Trigonometric and Inverse Trigonometric Functions.
9. Measurements conversion (km, cm, inch, etc.).
10. weight conversion (kilogram, liter, etc.).
11. Mixed to improper fractions.
12. Cube and cube root.
13. Temperature conversion.
14. Arithmetic operations of fractions. (+,-,\*,/) .
15. Calculation of Reminder.
16. Calculation of simple interest.
17. Binary to decimal conversion

***Low Level Requirement:***

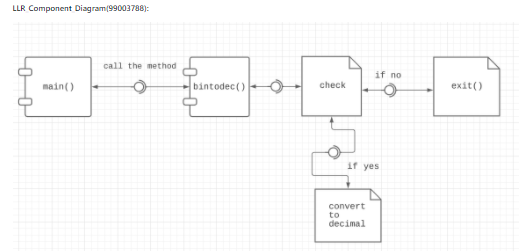
1. Exponential operation (result=b^x where b=base and x=exponent. Input type: integer and float) Logarithmic operation (result=log(x) where the input type of x is integer or float) .
2. Area of different geometrical shapes (The input data type used in this scenario are integer, float and long).
3. Volume of different geometrical shapes (The input data type used in the scenario are integer and float.)
4. Permutation and Combination (result=nPr; input type: integer and result=nCr; input type: integer).
5. Using float for Decimal values.
6. Using int for Integer values.
7. Using Double.
8. Calculation of reminder using modulus operator.
9. Using arithmetic operators for calculation of simple interest.

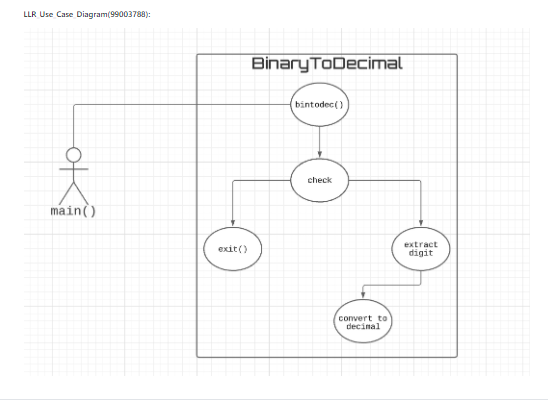
## **Design**

* High Level UML Diagram:

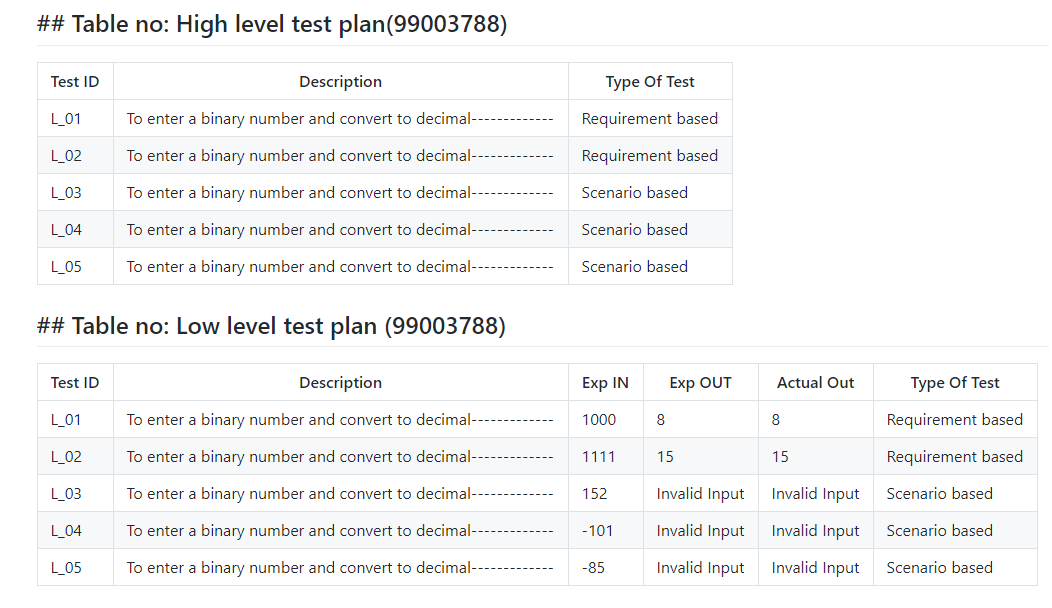


* Low Level UML Diagram:





## Test Plan



## Implementation Summary

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.

The **test** folder holds the ***test\_calculator\_operations.c*** file for cumulative testing of the source codes based on requirements, scenario and boundary.

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.

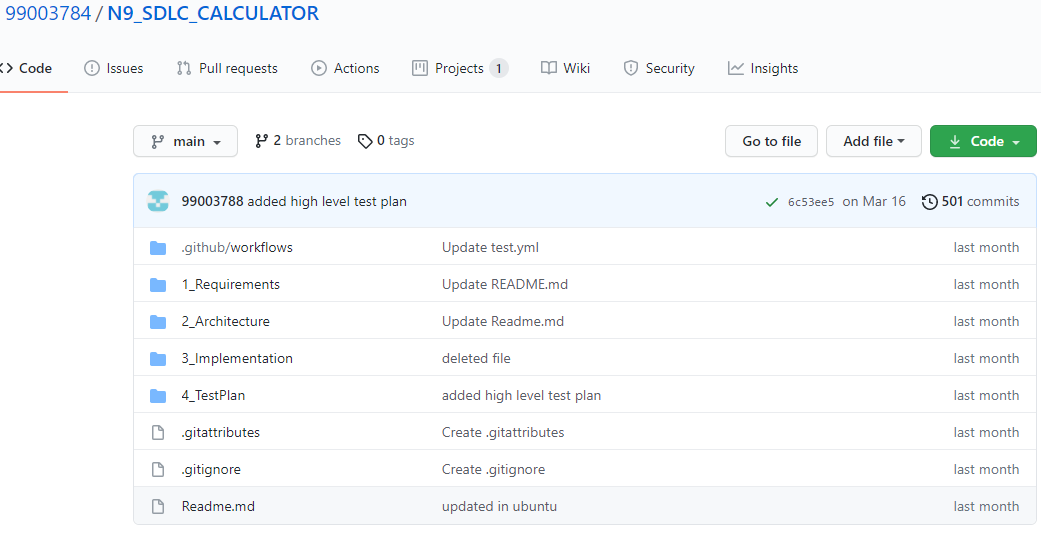
### Video Summary

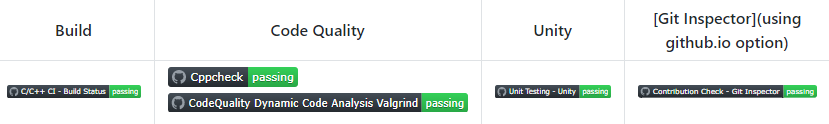
“Please upload a short video on the repo for the walkthrough of the project (Team/Individual) less than 7min and less than 30MB File Size. Start is the Standard opening slide with title of miniproject + Team members followed by the walkthrough”

### Git Link

<https://github.com/99003784/N9_SDLC_CALCULATOR.git>

### Git Dashboard

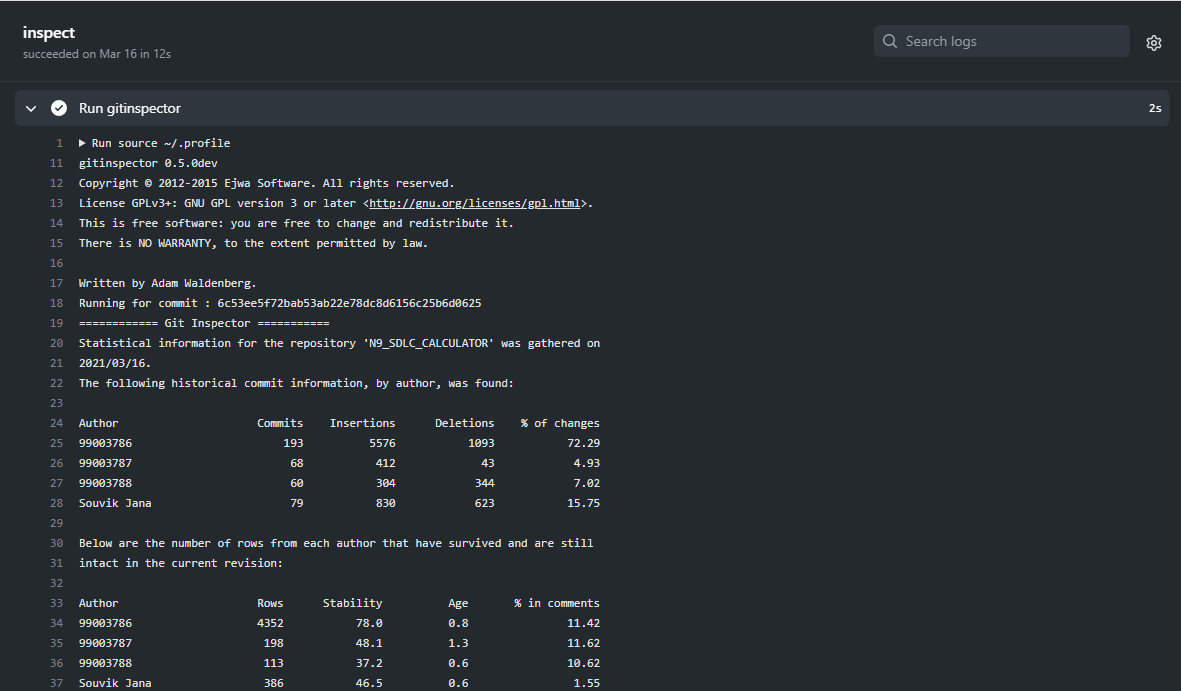
* ***GitHub Repo***
* ***Badges***

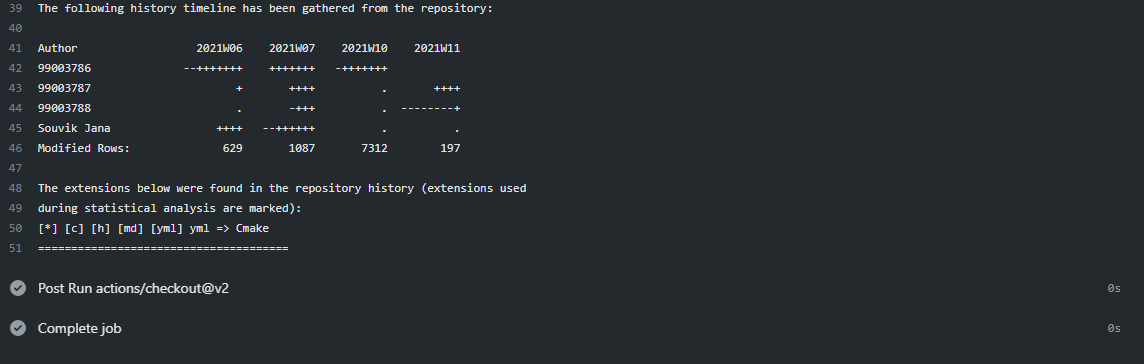
******

### Summary

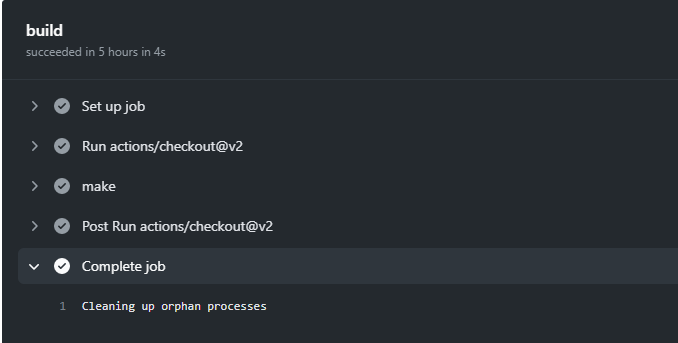
It is a simple electronic hardware/software device that can perform the simple calculations such as addition, subtraction, multiplication, division, calculating power of number, exponential function, logarithmic function, permutation and combination, trigonometry, inverse-trigonometric functions, factorial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

#### Git inspector summary

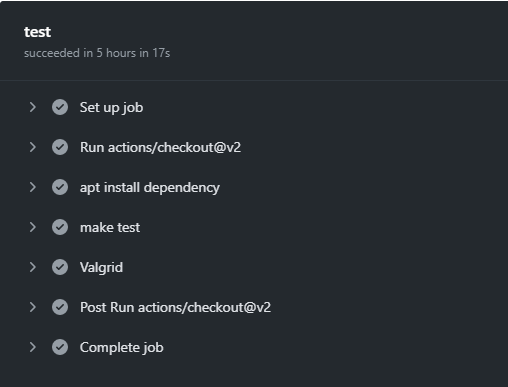


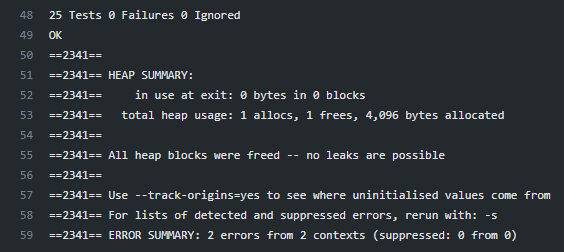


#### Build



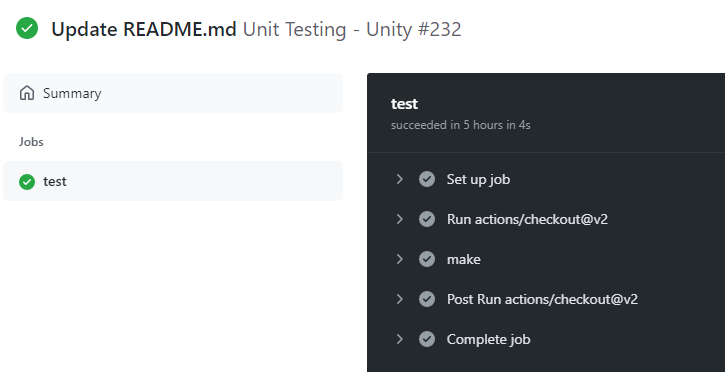
#### Code quality and Issues or Bug Tracking

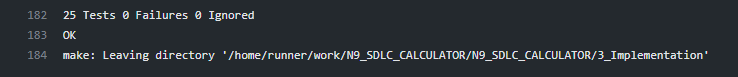
******



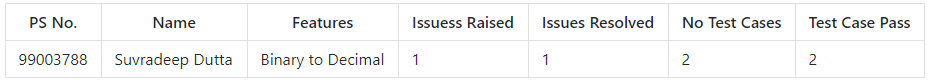
#### Unit Testing

“Unit Testing setup alignment with test plans and summary of outcome”





## Individual Contribution & Highlights

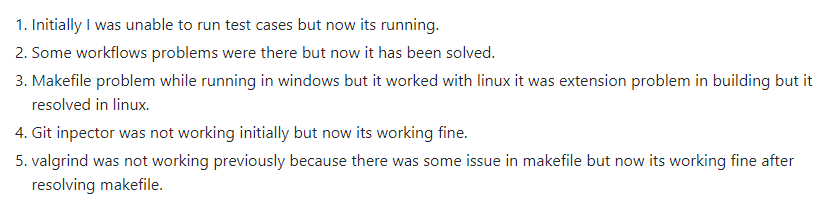


The function that I have implemented is conversion form binary to decimal. Here, first we need to accept a number and check whether it is greater than equal to 0 or not. If not then print “Invalid input”. If yes then the next step is to check whether the number is binary or not. If not the print “invalid Input”. If the number is binary, covert the binary number to its decimal equivalent. Finally print the decimal value.

### 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.

### Challenges faced and how were they overcome



### Future Scope (If applicable)

# Miniproject -2 [Team]

## Module

Embedded C

### Topic and Subtopics

* Driver API Development (Custom)

##### Sub-Topics

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

##### Sub-Topics

* GPIO
* ADC
* External interrupt.
* Debugging using STM-Board.

## Objectives

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

CAR-Module Features: -

* Power Window
* Door Lock/Unlock

## Requirements

##### High-Level Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Serial Number** | **Requirements** | **Description** | **Status** |
| 1 | Power Window | User manually opens or closes the window through a button press. | Implemented |
| 2 | Door Lock/Unlock | Indicates whether the doors of the cars are open or closed. | Implemented |

Table 5. High Level Requirements

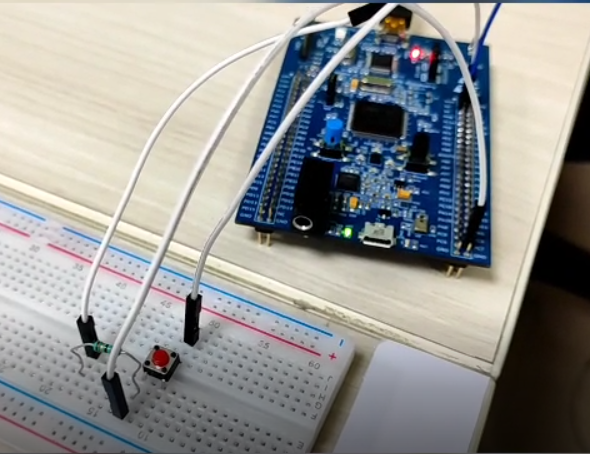
##### Low-Level Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Serial Number** | **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 | When the door(s) of the car is/are open. | GPIO pin PD14 is reset and buzzer buzzes. | Implemented |
| 4 | When the door(s) of the car is/are close | GPIO pin PD14 is set and the buzzer is silent. | Implemented |

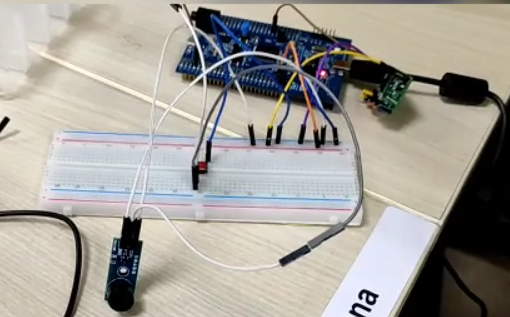
Table 6. Low-Level Requirements

## Design

* Power Window



* Door Lock/Unlock



## Test Plan

#### High-Level Test Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Serial Number** | **Description** | **Expected Input** | **Expected Output** | **Actual Out** | **Type of Test** |
| 1 | Door lock/unlock | Door is open | Buzzer in the car should start buzzing | Buzzer connected to the discovery board start buzzing. | Requirement based |
| 2 | Power window | Window is open | Led in the car starts glowing. | Green LED on the discovery board starts glowing | Requirement based |

Table 7. High-Level Testing

#### Low-Level Test Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Serial Number** | **Description** | **Expected Input** | **Expected Output** | **Actual Out** | **Type of Test** |
| 1 | Door lock/unlock | Door is open | Buzzer in the car should start buzzing | GPIO Pin PD 14 is reset and the buzzer connected to the discovery board start buzzing | Requirement based |
| 2 | Door lock/unlock | Door is close | Buzzer in the car is silent. | GPIO Pin PD 14 is set and the buzzer connected to the discovery board is silent | Requirement based |
| 3 | 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 |
| 4 | 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 |

Table 8. Low -Level Testing

## 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)
* Door Lock/Unlock:
* 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

# Miniproject -3 [Individual]

## Module

Python Programming

### 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

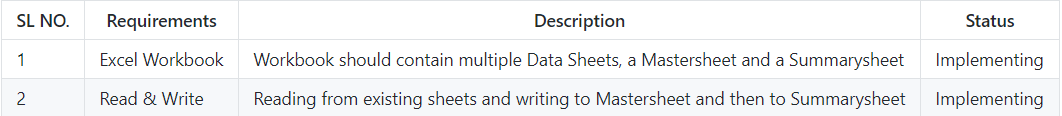
* Open multiple excel files
* Reading multiple excel files sheet by sheet.
* Mastersheet creation
* Writing to excel file

## Objectives

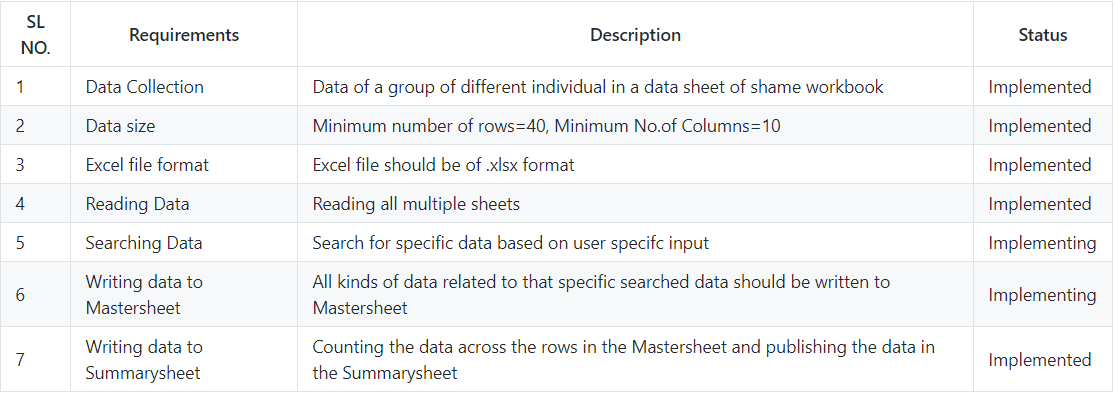
Analyzing data from 5 different excel spreadsheets and write specific data from all the excel sheets on a mastersheet and then counting number of entries in the mastersheet and write it in a summarysheet using Python Programming.

## Requirements

* High Level Requirements:

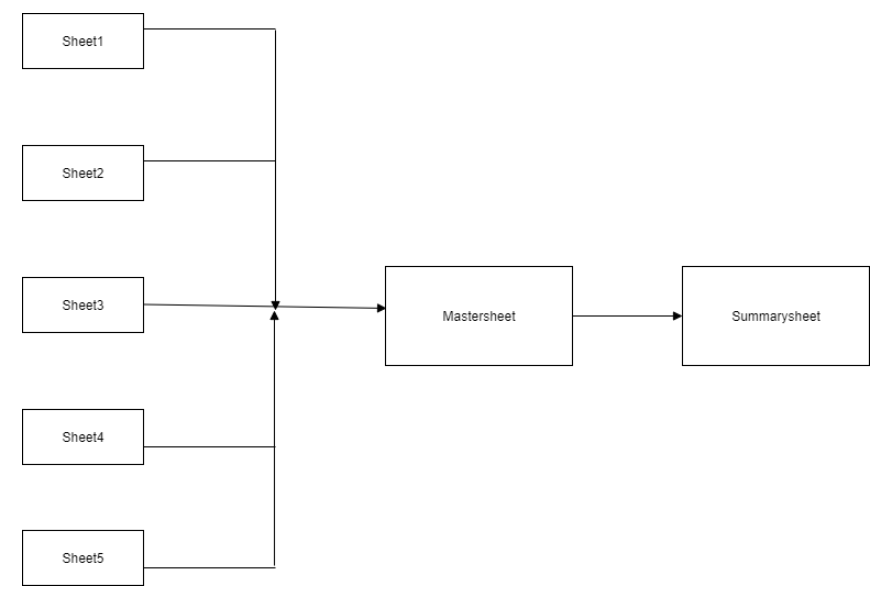


* Low Level Requirements:

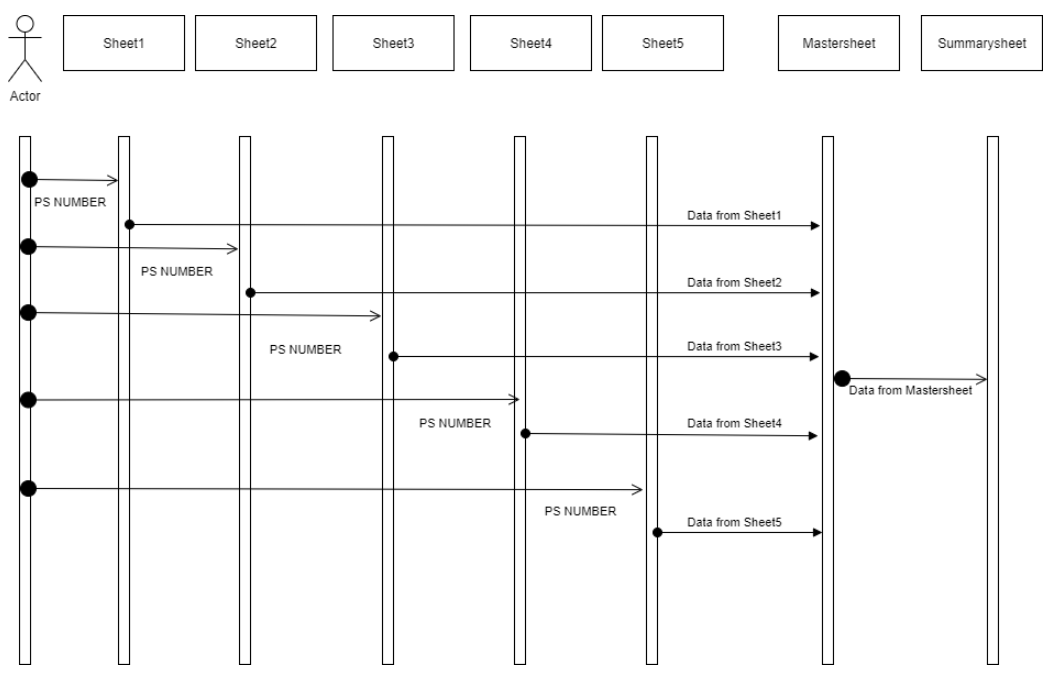


## Design

* High Level UML Diagram



* Low Level UML Diagram



## Test Plan:

#### High level Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Description** | **Expected Input** | **Expected Output** | **Actual Out** | **Type of Test** |
| HLR\_01 | 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>’ | Requirement based |
| HLR\_02 | 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>’ | Requirement based |
| HLR\_03 | 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 |

Table 11.High-Level Testing

#### Low Level Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Description** | **Expected Input** | **Expected Output** | **Actual Out** | **Type of Test** |
| HLR01-LR01 | 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 |
| HLR01-LR02 | 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 |
| HLR02-LR01 | 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 2. 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 |
| HLR02-LR02 | 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 2. 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 |
| HLR02-LR03 | 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   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 |
| HLR03-LR01 | 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 2. 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 |
| HLR03-LR02 | 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) 2. AP575 | Only data of AP575 copied to master-workbook | Only data of AP575 copied to master-workbook | Scenario based |
| HLR02-LR03 | 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   1. AB878 | Empty Master-workbook | Empty Master-workbook | Scenario based |

Table 12.Low-Level Testing

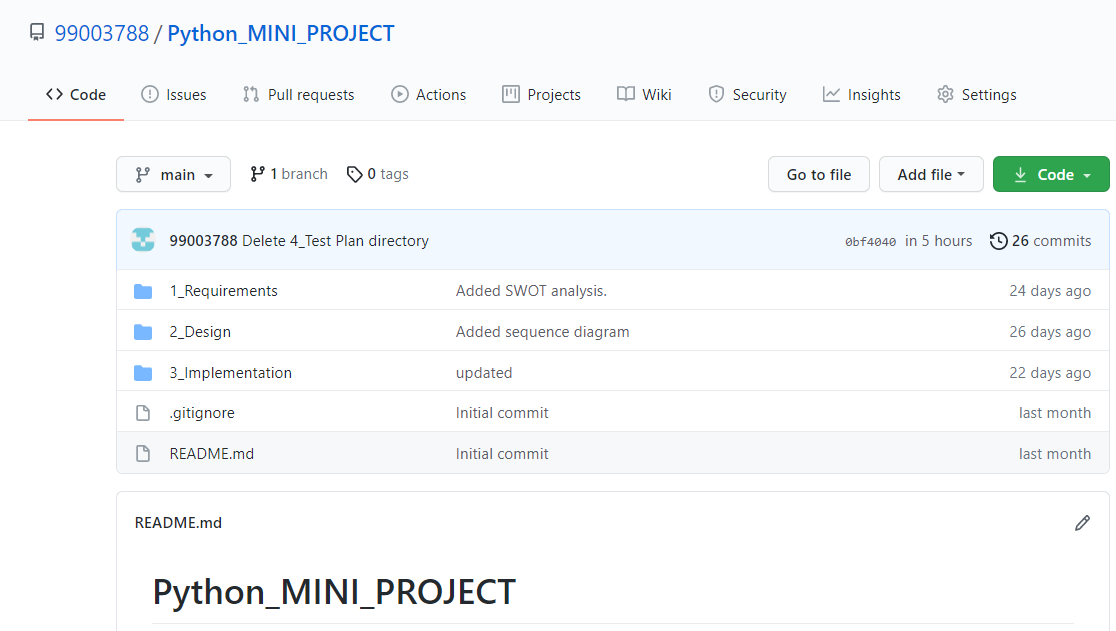
## Implementation Summary

Implementation folder has the python file with extension “.py” containing the code. It also contains the excel file which consists of 5 sheets.

### Git Link

<https://github.com/99003788/Python_MINI_PROJECT.git>

### Git Dashboard



### Summary

As per the objective, firstly I created an excel file that has five sheets. Then I started populating the sheets with data like NAME, PS NUMBER, Email, D.O.B, etc. The PS NUMBER column is common throughout all the five sheets.

Now coming to the code, here I have used openpyxl library. I have also followed Object Oriented Programming approach and used two classes. In the first class named ***accept\_value*** I have created a method ***take\_vlaue ()*** that gives the user a choice to select number of input and then asking him/her to enter the input(s). In the next class named ***main*** that inherits the ***accept\_value*** class***,*** am creating a method ***my\_func ().*** In this method, firstly am loading the excel file. Then making use of the PS NUMBER(S) entered by the user am searching all the data related to that PS NUMBER(S) and the dumping the entire date collected from the 5 sheets into a existing sheet named ***mastersheet***. After that I am counting the number of rows and columns in the mastersheet and then reflecting the result into a sheet named summarysheet.

### Summary

### Challenges faced and how were they overcome

# Miniproject -4 [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