./

GENESIS - Learning Outcome & Mini-project Summary Report



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **To be Approved** | **Remarks/Revision Details** |
| 1. | 14/04/2021 | Jasleen Kaur Sethi/99003720 |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Details**

Contents

[Contents 3](#_Toc69458865)

[Applied SDLC – Building a Calculator [Team] 5](#_Toc69458866)

[Module 5](#_Toc69458867)

[Topic and Subtopics 5](#_Toc69458868)

[Objectives & Requirements 6](#_Toc69458869)

[Design 10](#_Toc69458870)

[High Level Design 10](#_Toc69458871)

[Test Plan 16](#_Toc69458872)

[High Level Test Plan 16](#_Toc69458873)

[Implementation Summary 24](#_Toc69458874)

[Video Summary 24](#_Toc69458875)

[Git Link 24](#_Toc69458876)

[Git Dashboard 24](#_Toc69458877)

[Summary 25](#_Toc69458878)

[Individual Contribution & Highlights 33](#_Toc69458879)

[Summary 33](#_Toc69458880)

[Challenges faced and how were they overcome 33](#_Toc69458881)

[Future Scope 33](#_Toc69458882)

[Creating a data sheet using Python [Individual]: 34](#_Toc69458883)

[Modules 34](#_Toc69458884)

[Topic and Subtopics 34](#_Toc69458885)

[Objectives & Requirements 34](#_Toc69458886)

[Design 35](#_Toc69458887)

[Test Plan 36](#_Toc69458888)

[Implementation Summary 38](#_Toc69458889)

[Git Link 38](#_Toc69458890)

[Git Dashboard 38](#_Toc69458891)

[Summary 39](#_Toc69458892)

[Individual Contribution & Highlights 39](#_Toc69458893)

[Summary 39](#_Toc69458894)

[Challenges faced and how were they overcome 40](#_Toc69458895)

[Miniproject -3 [Team]: Embedded C 40](#_Toc69458896)

[Module/s 40](#_Toc69458897)

[Topic and Subtopics 40](#_Toc69458898)

[Objectives & Requirements 40](#_Toc69458899)

[ID 41](#_Toc69458900)

[Requirements 41](#_Toc69458901)

[Description 41](#_Toc69458902)

[Status 41](#_Toc69458903)

[EmL0 41](#_Toc69458904)

[The module is enabled 41](#_Toc69458905)

[Module is enabled when GPIO Pin 14 is set 41](#_Toc69458906)

[IMPLEMENTED 41](#_Toc69458907)

[EmL1 41](#_Toc69458908)

[LED glows when seat Control module is enabled. 41](#_Toc69458909)

[LDR senses darkness. Orange LED when the control is set. 41](#_Toc69458910)

[IMPLEMENTED 41](#_Toc69458911)

[EmL2 41](#_Toc69458912)

[Buzzer beeps and LED glows when Door Control module is enabled. 41](#_Toc69458913)

[Green LED is lit and Buzzer beeps when the control is set. 41](#_Toc69458914)

[IMPLEMENTED 41](#_Toc69458915)

[EmL3 41](#_Toc69458916)

[LED glows when seat Control module is enabled. 41](#_Toc69458917)

[Green LED when the control is set. 41](#_Toc69458918)

[IMPLEMENTED 41](#_Toc69458919)

[EmL4 41](#_Toc69458920)

[Moisture sensor is enabled when Wiper control module is enabled. 41](#_Toc69458921)

[Moisture sensor senses the water on windshield & Green LED is lit if Wiper control module is set. 41](#_Toc69458922)

[IMPLEMENTED 41](#_Toc69458923)

[EmL5 41](#_Toc69458924)

[RGB is enabled when Power window module is enabled and window opens 41](#_Toc69458925)

[RGB turns pink when the control is set. 41](#_Toc69458926)

[IMPLEMENTED 41](#_Toc69458927)

[EmL6 41](#_Toc69458928)

[LED glows when AC Conditioning Control module is enabled. 41](#_Toc69458929)

[Blue LED when the control is set 41](#_Toc69458930)

[IMPLEMENTED 41](#_Toc69458931)

[Design 41](#_Toc69458932)

[Test Plan 41](#_Toc69458933)

[Implementation Summary 43](#_Toc69458934)

[Git Link 44](#_Toc69458935)

[Git Dashboard 44](#_Toc69458936)

[Individual Contribution & Highlights 44](#_Toc69458937)

[Summary 44](#_Toc69458938)

[Challenges faced and how were they overcome 45](#_Toc69458939)

[Linux and OS programming & Kernel [Individual] 45](#_Toc69458940)

[Modules 45](#_Toc69458941)

[Topic and Subtopics 45](#_Toc69458942)

[Objectives 45](#_Toc69458943)

[Requirements 45](#_Toc69458944)

[Test Plan 47](#_Toc69458945)

[Implementation Summary 48](#_Toc69458946)

[Git Link 48](#_Toc69458947)

[Git Dashboard 48](#_Toc69458948)

[Summary 48](#_Toc69458949)

[Summary 49](#_Toc69458950)

[Challenges faced and how were they overcome 49](#_Toc69458951)

# Applied SDLC – Building a Calculator [Team]

## **Module**

The modules linked to the mini project is SDLC and C Programming

### Topic and Subtopics

The core topics and subtopics being implemented are:

**Software Development Life Cycle**

The basics of the Software Development Life Cycle. It includes the flow of development in a product in the industry.

* It starts with finding the various high level and low-level requirements that are required for building the product. Then the high level and low-level design is built for brief visual understanding of the working of the product.
* The software for the product is then implemented in sync with the requirements.
* The implemented code is then tested with the method of unit testing and integration testing.
* After all the implementation and testing, it is finally deployed in the market.

**Unified Modelling Language(UML) Diagrams**

The various structural and Behavioral diagrams were studied and implemented for designing the product according to the requirements provided.

The various structural and Behavioral diagrams learnt were:

Structural Diagrams:

* + - Class Diagram
    - Component Diagram
    - Object Diagram
    - Deployment Diagram
    - Package Diagram
    - Profile Diagram
    - Composite Structure Diagram

Behavioral Diagrams:

* + - Use Case Diagram
    - Activity Diagram
    - State Machine Diagram
    - Sequence Diagram
    - Communication Diagram
    - Interaction Overview Diagram
    - Timing Diagram

## **Objectives & Requirements**

**Objective:**

 To build the software development cycle of a semi-engineering calculator. It fulfils the numerous high-level and low-level requirements of the various features involved in semi-engineering such as log, antilog, exponents, roots, LCM and GCD functions. It then explores about the various test cases that should be taken in consideration such that all requirements are taken care of. In the end, we build the codes that follow all the requirements and design efficiently as per them.

**Requirements:**

**High Level Requirements:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirements** | **Description** | **Status** |
| HL001 | ARITHMETIC OPERATIONS | BASIC ARITHMETIC CALCULATION – Addition, Subtraction, Division and multiplication | IMPLEMENTED |
| HL002 | LOG ANTILOG FUNCTIONS | SOLVING POWER FUNCTIONS AND LOG/ANTILOG OPERATIONS | IMPLEMENTED |
| HL003 | TRIGONOMETRIC FUNCTIONS | SINE, COSINE, TAN FUNCTIONS | IMPLEMENTED |
| HL004 | PERMUTATION & COMBINATION | PERMUTATION AND COMBINATION FORMULAE OUTPUTS | IMPLEMENTED |
| HL005 | CONVERSIONS OF UNITS | UNIT COVERSIONS FOR TEMPERATURE AND VARIOUS LENGTH UNITS | IMPLEMENTED |
| HL006 | FACTORIAL & POWER | FACTORIAL AND POWER OF THE POSITIVE NUMBERS | IMPLEMENTED |
| HL007 | LCM & GCD | LCM and GCD OF THE POSSIBLE REAL NUMBERS | IMPLEMENTED |
| HL008 | PRECEDENCE | CALCULATION ACCORDING TO THE PRECEDENCE OF THE FUNCTIONS | IMPLEMENTED |
| HL009 | STORE AND RECALL | CALCULATOR TO STORE AND RECALL RESULTS | IMPLEMENTED |
| HL0010 | PRECISION | RESULTS SHOULD BE ACCORDING TO PRECISION | IMPLEMENTED |

**Low Level Requirements:**

**Arithmetic Operations:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirements** | **Description** | **Status** |
| LL001 | Addition | Taking two numbers as input and adding them | Implemented |
| LL002 | Subtraction | Taking two numbers as input and subtracting them | Implemented |
| LL003 | Multiply | Taking two numbers as input and multiplying them | Implemented |
| LL004 | Divide | Taking two numbers as input and dividing them | Implemented |

**Log and Antilog operations**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirements** | **Description** | **Status** |
| LL003 | Logarithm | Taking one number as input with base as 10 and finding its log output | Implemented |
| LL004 | Anti-logarithm | Taking one number as input with base as 10 and finding its antilog output | Implemented |

**Trigonometric Operations**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirements** | **Description** | **Status** |
| LL005 | Sine Functions | Finding the sine of a number | Implemented |
| LL006 | Cos Functions | Finding the cos of a number | Implemented |
| LL007 | Tan Functions | Finding the tan of a number | Implemented |

**Permutation and Combination**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirements** | **Description** | **Status** |
| PER\_1 | n<R | shows error when N<R | Implemented |
| PER\_2 | Formulae calculation Permutation | finds permutation using the formulae | Implemented |
| COM\_1 | n<R | shows error when N<R | Implemented |
| COM\_2 | Formulae calculation combination | finds combination using the formulae | Implemented |

**Conversion of Units**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirements** | **Description** | **Status** |
| COU\_1 | Length conversion | converts input from cm to m &km | Implemented |
| COU\_2 | Temperature conversion | converts user input from Celsius to Fahrenheit | Implemented |
| COU\_3 | Degree to radian conversion | converts one user input from degree to rad at a time | Implemented |

**Factorial and Power**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirements** | **Description** | **Status** |
| Fac\_1 | Factorial | Finding factorial of the given number | Implemented |
| pre\_1 | Library | Math library file will be required in coding section. | Implemented |
| pre\_2 | power function | Requirement of power function while executing this section of program. | Implemented |
| pre\_3 | Data type | Must contain best suited data types else show | Implemented |
| pre\_3 | Boundary limit | Limit exceeds. | Implemented |

**LCM and GCD Functions**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirements** | **Description** | **Status** |
| LL0011 | LCM | Finding the least common multiple of the possible real numbers | Implemented |
| LL0012 | GCD | Finding the greatest common devisor of the possible real numbers | Implemented |

**Precedence**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirements** | **Description** | **Status** |
| BOD\_1 | Bracket Type | For more than 1 operation, brackets ((),[],{}) should be identified and executed first. | Implemented |
| BOD\_2 | Bracket order 1 | The bracket order (), {}, [] is to be followed. | Implemented |
| BOD\_3 | Bracket order 2 | For more than 1 operation, powers and roots should be identified and executed after brackets. | Implemented |
| BOD\_4 | Bracket order 3 | For more than 1 operation, Division should be identified and executed after exponents and roots. | Implemented |
| BOD\_5 | Bracket order 4 | For more than 1 operation, Multiplication should be identified and executed after Division. | Implemented |
| BOD\_6 | Bracket order 5 | For more than 1 operation, Multiplication should be identified and executed after Division. | Implemented |
| BOD\_7 | Bracket order 6 | For more than 1 operation, Addition should be identified and executed after Multiplication. | Implemented |
| BOD\_8 | Bracket order 7 | For more than 1 operation, subtraction should be identified and executed after Addition. | Implemented |

**Store and Recall Functions**

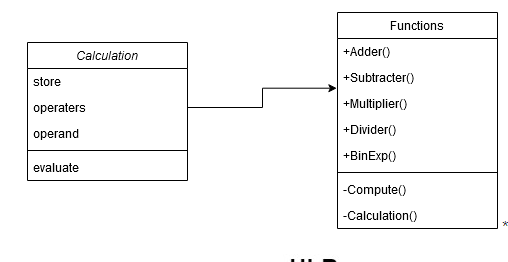
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirements** | **Description** | **Status** |
| BOD\_1 | Stored value | - Capable of storing values for 50 days | Implemented |
| BOD\_2 | MC function | - Pressing ‘MC’ and 0 clears the memory | Implemented |
| BOD\_3 | Memory storage | - Stores up to 5 memories. | Implemented |
| BOD\_4 | Memory style | - Show all memories in a first in last out style after every click | Implemented |

**Precision**

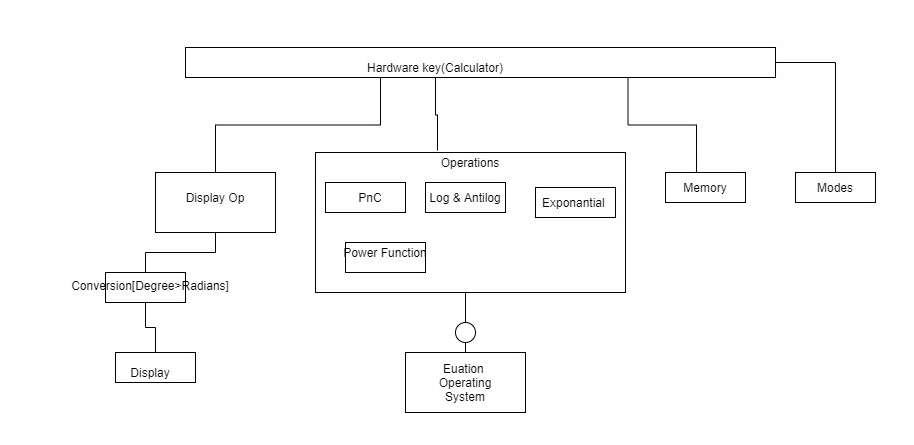
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirements** | **Description** | **Status** |
| pre\_1 | Multiplication power | When multiplication of 10 digit it should show in power of ten | Implemented |
| pre\_2 | divided by 0 | When value is divided by zero then zero then it should through error. | Implemented |
| pre\_3 | Boundary Precision | user puts value of more than 10 then it must not except more than that and print limit exceeds as an output. | Implemented |
| COM\_3 | Formulae calculation combination | finds combination using the formulae. | Implemented |
| COM\_1 | n<R | shows error when N<R | Implemented |

## **Design**

## **High Level Design**



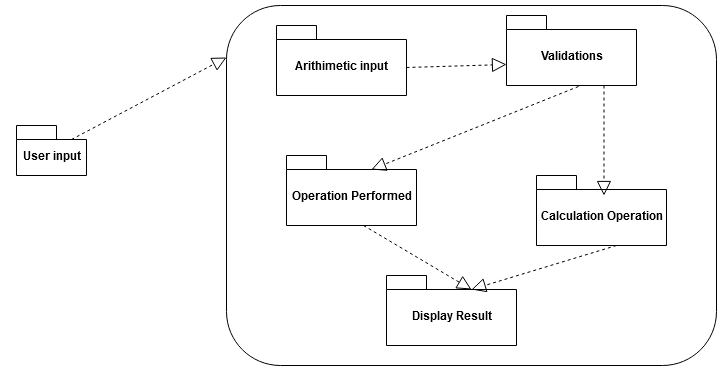
**Structural – Class Diagram**



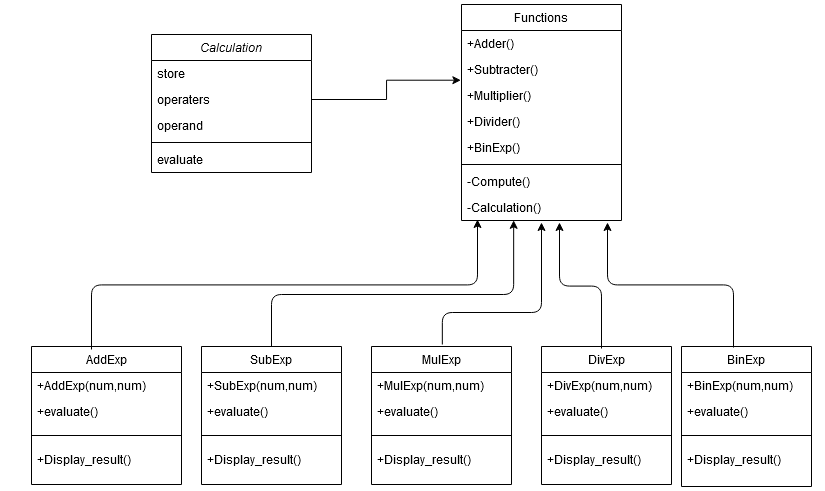
**Structural – Component Diagram**

**Low Level Design**

**Arithmetic Operations**

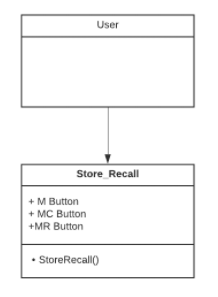


**Structural - Package Diagram**

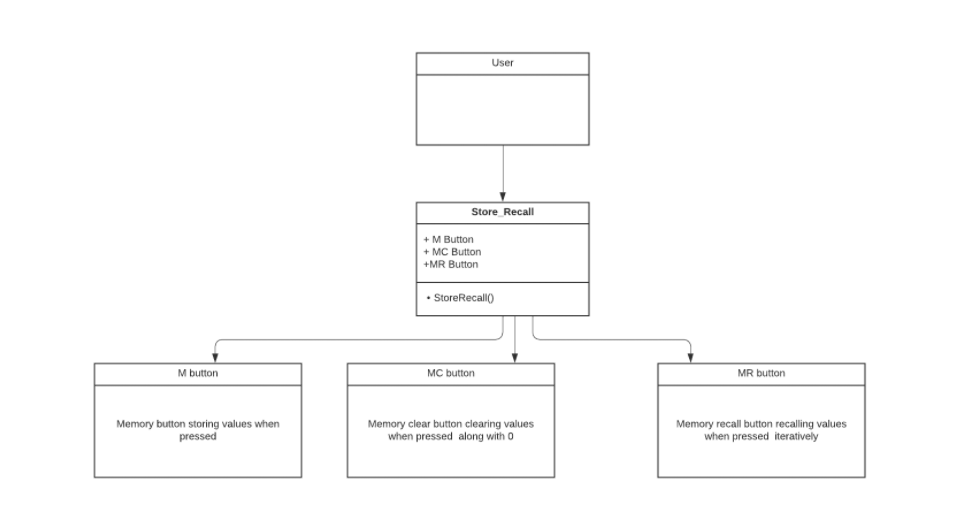


**Behavioral – Object Diagram**

**Store and Recall:**

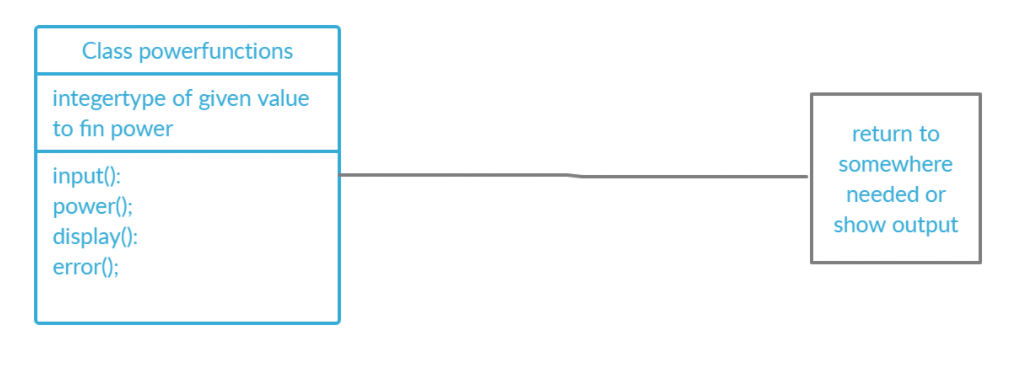


**Structural – Class Diagram**



**Behavioral – Object Diagram**

**Power Function**

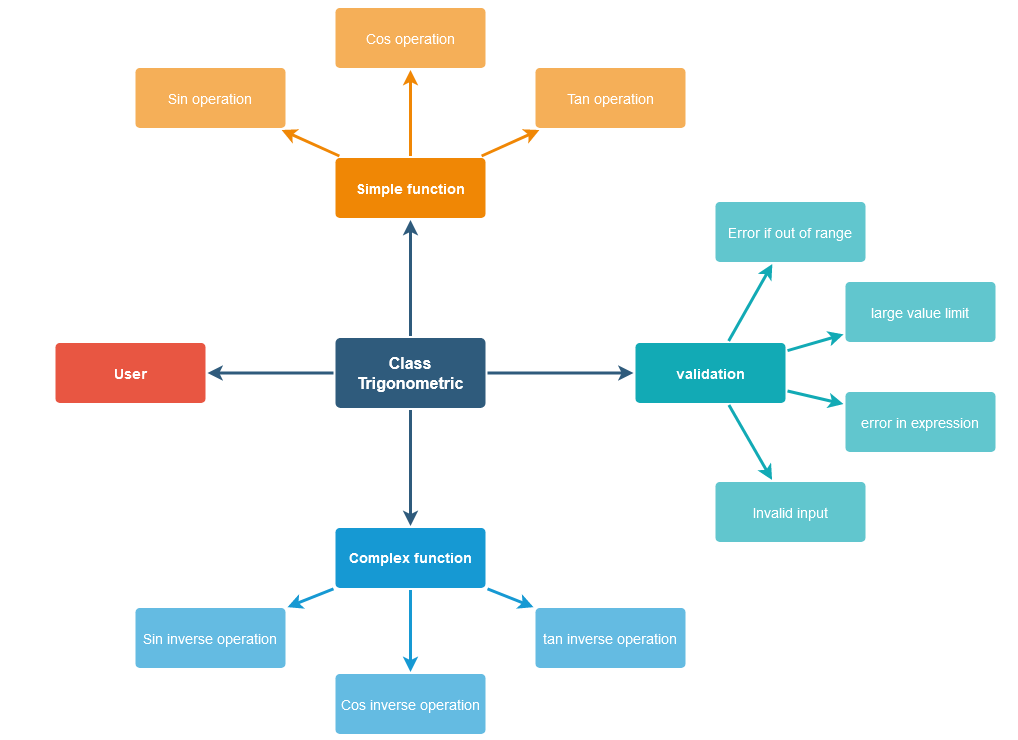


**Structural – Class Diagram**



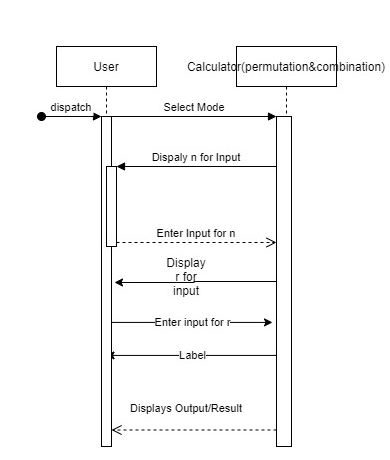
**Behavioral – Object Diagram**

**Trigonometric Functions**



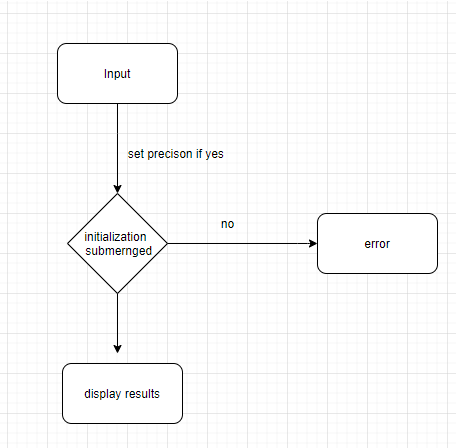
**Structural – Component Diagram**

**Permutation and Combination**



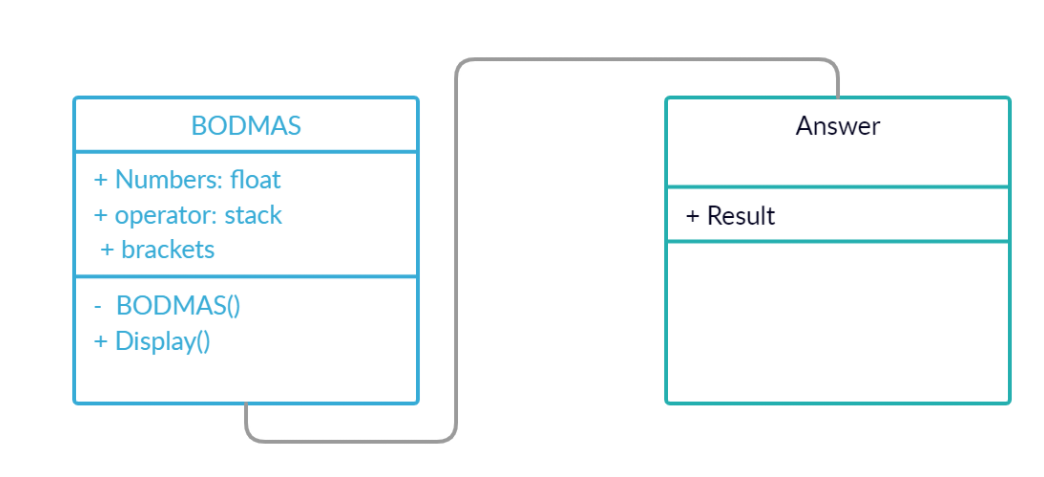
**Behavioral – Sequence Diagram**

**Precision**

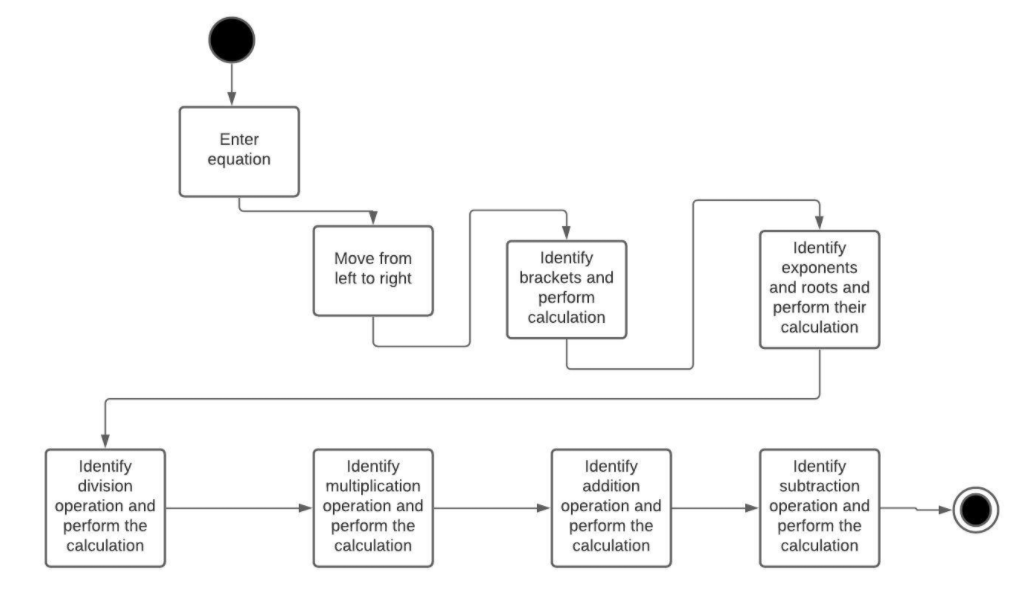


**Behavioral – Activity Diagram**

**Precedence**



**Structural – Class Diagram**



**Behavioral – Activity Diagram**

## **Test Plan**

## **High Level Test Plan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Description | Expected Input | Expected Output | Actual Output | Type of test |
| L\_01 | Verify that BODMAS is applied in case of complex queries and the correct result is returned. | 24+3\*2 | 30 | 30 | Requirement based |
| L\_02 | Addition of 2 or more numbers | 20+30 | 50 | 50 | Scenario based |
| L\_03 | Sequential incorrect parameter | 2+4\*3. y | 20 | Error | Boundary based |
| L\_04 | Out of range check | addition if exceed 15 digit numbers | any output | Result out of range | Requirement based |
| L\_05 | Sequential incorrect parameter | 2+4\*3. y | 20 | Error | Boundary based |
| L\_06 | Two symbols cannot be between numbers | 3++5 | 2 | Invalid Format | Scenario based |
| L\_07 | Trigonometry Sine value | sin(x) | sin x | sin x | Boundary based |

**Low Level Test Plan**

**Arithmetic Operations**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Description | Expected Input | Expected Output | Actual Output | Type of test |
| L\_0211 | Adding two negative numbers | num1=-750, num2=-7500 | result=-8250 | result=-8250 | Scenario based |
| L\_0212 | Adding one negative number and one positive number | num1=10, num2=-20 | result=-10 | result=-10 | Scenario based |
| L\_0213 | Adding one positive number and one negative number | num1=-98, num2=20 | result=-78 | result=-78 | Scenario based |
| L\_0221 | Subtracting two positive numbers | num1=13, num2=4 | result=9 | result=9 | Scenario based |
| L\_0222 | Subtracting positive number from a negative number | num1=-45, num2=6 | result=-51 | result=-51 | Scenario based |
| L\_0223 | Subtracting two negative numbers | num1=-7985, num2=-7745 | result=-240 | result=-240 | Scenario based |
| L\_0231 | Multiplying one positive number with one negative number | num1=-20, num2=3 | result=-60 | result=-60 | Scenario based |
| L\_0232 | Multiplying one negative number with one positive number | num1=8745, num2=-83 | result=-725835 | result=-725835 | Scenario based |
| L\_0233 | Multiplying two negative numbers | num1=-99, num2=-999 | result=98901 | result=98901 | Scenario based |
| L\_0241 | Dividing by zero | num1=1, num2=0 | result=-1(for error) | result=-1 | Scenario based |
| L\_0242 | Dividing two negative number | num1=-135, num2=-3 | result=45 | result=45 | Scenario based |
| L\_0243 | Dividing negative number by positive number | num1=-28, num2=14 | result=-2 | result=-2 | Scenario based |
| L\_0244 | Dividing positive number by negative number | num1=96, num2=-12 | result=-8 | result=-8 | Scenario based |
| L\_0311 | Multiplying two 8 digit numbers | num1=100000000, num2=99999999 | result=-1 | result=-1(for Error) | Boundary based |
| L\_0232 | Multiplying one negative number with one positive number | num1=8745, num2=-83 | result=-725835 | result=-725835 | Scenario based |
| L\_0233 | Multiplying two negative numbers | num1=-99, num2=-999 | result=98901 | result=98901 | Scenario based |
| L\_0241 | Dividing by zero | num1=1, num2=0 | result=-1(for error) | result=-1 | Scenario based |
| L\_0242 | Dividing two negative number | num1=-135, num2=-3 | result=45 | result=45 | Scenario based |

**Precedence**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Description | Expected Input | Expected Output | Actual Output | Type of test |
| LLR\_P\_1 | First, the operators within brackets are executed in order of multiplication and subtraction order and then add operator outside bracket. | 14 + (8 – 2 × 3) | 16 | 16 | Logical test |
| LLR\_P\_2 | The terms inside {} are simplified followed by () and then operated with terms outside the bracket. | 1800÷10{(12−6)+(24−12)} | 3240 | 3240 | logical |
| LLR\_P\_3 | The terms inside () are simplified followed by {}, then [] and then terms are operated with the terms outside the bracket. | 1/2[{−2(1+2)}10] | -30 | -30 | Logical |
| LLR\_P\_4 | The terms inside () are simplified first followed by exponent and then terms are operated with the terms outside the bracket. | (2^9\*2) + 9 / (e^3) | 1204.76983 | 1204.76983 | Logical |
| LLR\_P\_5 | An equation of the result of more than 8 digits is given by the user. | (10\*5+100) \*100 | Error: Out of range | Error: Out of range | Boundary Based |
| LLR\_P\_6 | Values up to 9 digits | 3+(3\*4+5) \*3/10+1 | Error: Out of range | Error: Out of range | Requirement based |

**Roots**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Description | Expected Input | Expected Output | Actual Output | Type of test |
| LLR\_P\_1 | Finding the square of given number i.e. find power of number will be 2. | 100^2 | 10000 | 10000 | logic based |
| LLR\_P\_2 | Finding the cube of given number i.e. find power of number will be 3. | 10^3 | 1000 | 1000 | logic based |
| LLR\_P\_3 | Finding the power of given number i.e. find power of number will be any number | 2^5 | 32 | 32 | logic based |
| LLR\_P\_4 | Finding the power of big number i.e. find power of number which results in digits more than 20 | 3^20 | 3.48x10^9 | 3.48x10^9 | Boundary based |
| LLR\_P\_5 | Finding the power of other then the numbers i.e. finding power of any operator or any other buttons | [^20Or +^60 | Error: Out of range | Error: Out of range | Requirement based |

**LCM and GCD**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Description | Expected Input | Expected Output | Actual Output | Type of test |
| LLR\_P\_1 | finding LCM of two positive numbers | 5,15 | 75 | 75 | logic based |
| LLR\_P\_2 | finding LCM of any one negative number | -72,120 | 360 | 360 | logic based |
| LLR\_P\_3 | finding LCM of one number as zero | 0,10 | Error | Error | logic based |
| LLR\_P\_4 | finding LCM of two large numbers | 100000,100000 | Out of range | Out of range | Boundary based |
| LLR\_P\_1 | finding GCD of two positive numbers | 5,15 | 5 | 5 | logic based |
| LLR\_P\_2 | Finding GCD of any one negative number | -72,120 | 20 | 20 | logic based |
| LLR\_P\_3 | finding GCD of one number as zero | 0,10 | Error | Error | logic based |
| LLR\_P\_4 | finding GCD of two large numbers | 100000,100000 | Out of range | Out of range | Boundary based |

**Permutation and Combination**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Description | Expected Input | Expected Output | Actual Output | Type of test |
| LLR\_P\_1 | find permutation using the formula | 3,2 | 6 | 6 | logic based |
| LLR\_P\_2 | finding permutations of large numbers | 180,179 | Error: large result | Error: large Result | Boundary based |
| LLR\_P\_3 | Permutation of numbers when n<r | 10,11 | Error | Error | requirement based |
| LLR\_P\_4 | find combination using the formula | 6,4 | 15 | 15 | logic based |
| LLR\_P\_5 | finding combination of large numbers | 180,179 | Error: large result | Error: large Result | Boundary based |
| LLR\_P\_6 | Combination of numbers when n<r | 4,6 | Error | Error | requirement based |

**Trigonometric Functions**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Description | Expected Input | Expected Output | Actual Output | Type of test |
| LLR\_P\_1 | Trigonometric Sine value of 0 | 0 | 0 | 0 | math.h file & Boundary based |
| LLR\_P\_2 | Trigonometric Cosine value of 0 | 0 | 1 | 1 | math.h file & Boundary based |
| LLR\_P\_3 | Trigonometric Tan value of 0 | 0 | 0 | 0 | math.h file & Boundary based |
| LLR\_P\_4 | Trigonometric Sine value of 90 | 90 | 1 | 1 | math.h file & Boundary based |
| LLR\_P\_5 | Trigonometric Cosine value of 90 | 90 | 0 | 0 | math.h file & Boundary based |
| LLR\_P\_6 | Trigonometric Tan value of 90 | 90 | not defined | not defined | math.h file & Boundary based |
| LLR\_P\_6 | Other than numeric values in trigonometric ratios | /,=,,&,;,', | error | error | math.h file & Boundary based |

**Store and Recall Functions**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Description | Expected Input | Expected Output | Actual Output | Type of test |
| LLR\_P\_1 | M button after 50 days | M button after 50 days | No memory found | No memory found | requirement based |
| LLR\_P\_2 | MC button clicked to select the value to clear and 0 pressed on 3rd value to clear it | MC button clicked and 0 pressed on 3rd value | Displays values and then clears the 3rd value | Displays values and then clears the 3rd value | logic based |
| LLR\_P\_3 | M clicked after 5 stored values already | entering the 6th value to store in the memory | Memory full | Memory full | requirement based |
| LLR\_P\_4 | MR button is pressed | Displays values iteratively | Displays values iteratively | Out of range | Boundary based |

## **Implementation Summary**

The implementation part of the project consists of the software implementation of the product, Calculator, as documented in the requirements and design section of the life cycle. It consists of various C files in the and header files that link the multiple files. It also consists of the Makefile which is used to build run and clean all the multiple files and check for the test cases that are formulated according to the requirement specified.

The source files contain the implementation of the various high-level requirements specified including the following functions:

* + - Arithmetic Operations
    - Trigonometric Operations
    - Roots
    - Log and Antilog functions
    - Permutation and combination
    - Conversion of units
    - Factorial and power functions

Note: The GitHub private repo should be documented (Readme.md files at each folder level)

Ensure code quality and clean code and description practices

Mandatory: To add the GitHub user - **stepin654321** as a contributor to the repo”

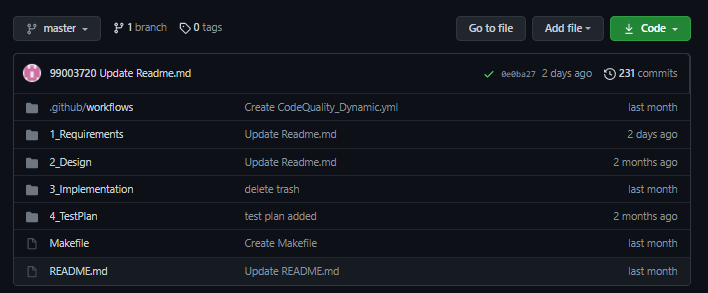
### Video Summary

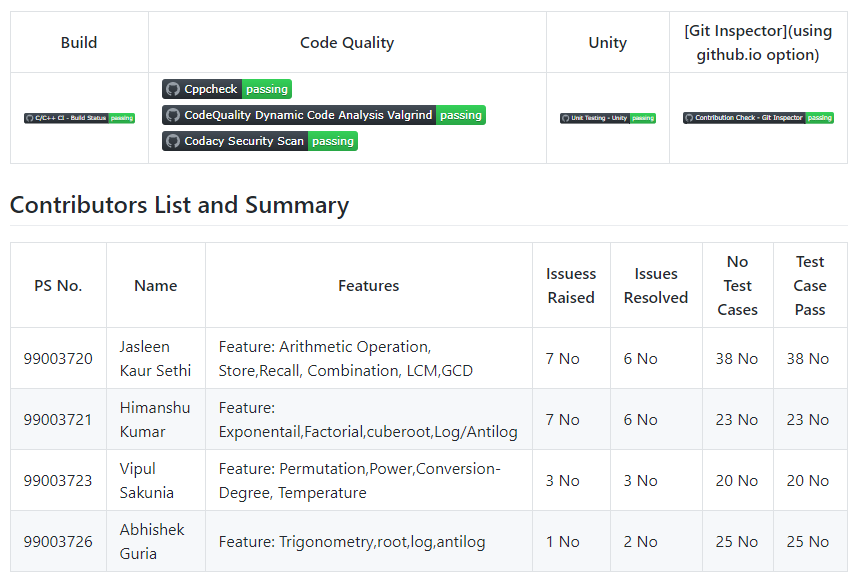
“Please upload a short video on the repo for the walkthrough of the project (Team/Individual) less than 7min and less than 30MB File Size. Start is the Standard opening slide with title of miniproject + Team members followed by the walkthrough”

### Git Link

<https://github.com/99003721/AppliedSDLC_Calculator_N1>

### Git Dashboard





### Summary

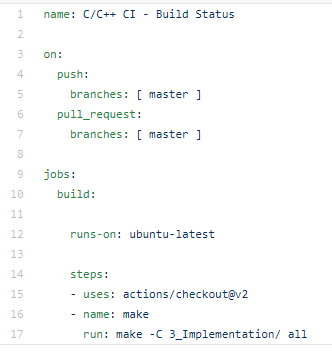
This project started with the research on the various calculators in the market, its age and its features. Then the requirements for our custom calculator was documented in the form of high level and low-level requirements. After the formulation of requirements, the design was made using the UML diagrams for the visual understanding of how the calculator software must be executed. After that, all the features required by the calculator are created in the form of C codes. The codes are then tested with all the possible test cases and error detection and correction is done. All the multiple files along with the headers are then executed and run with the help of a Makefile. Various building, CPP Checks and Unity testing is then done on the codes and badges for these workflows are finally attained.

#### Git inspector summary

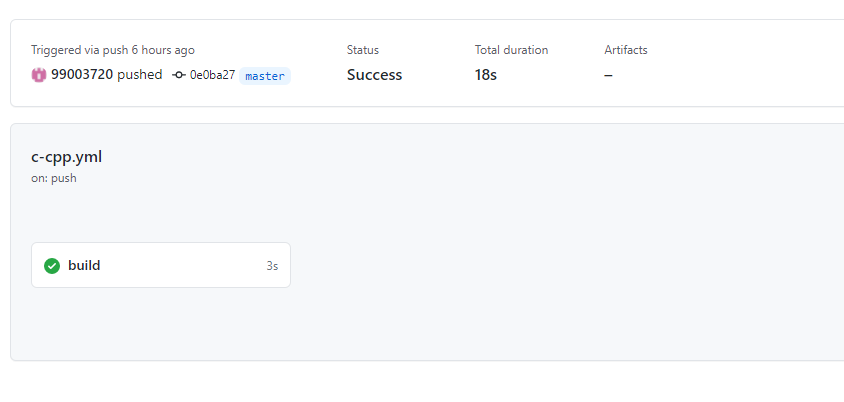
#### 

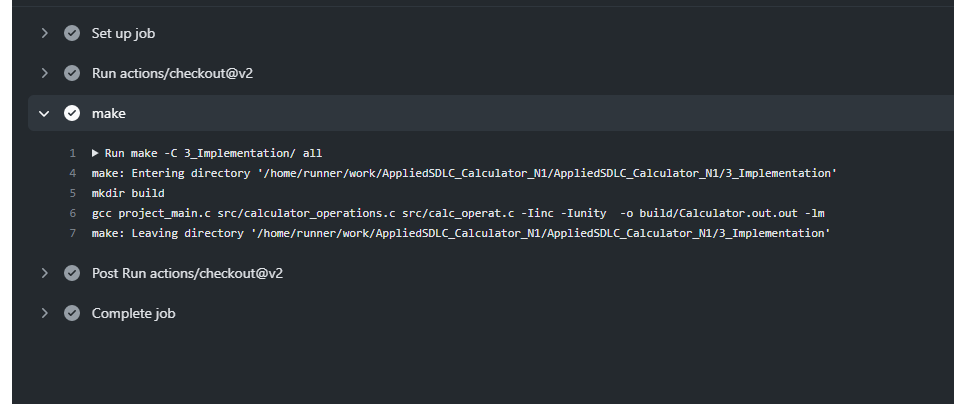
#### Build

**Setup done**



**Outcome**



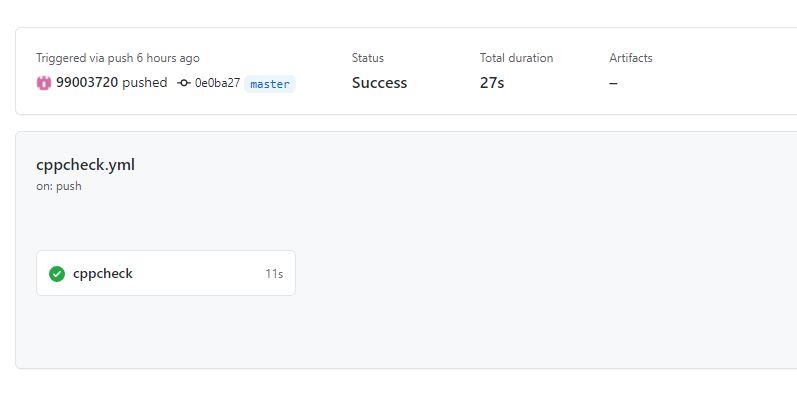


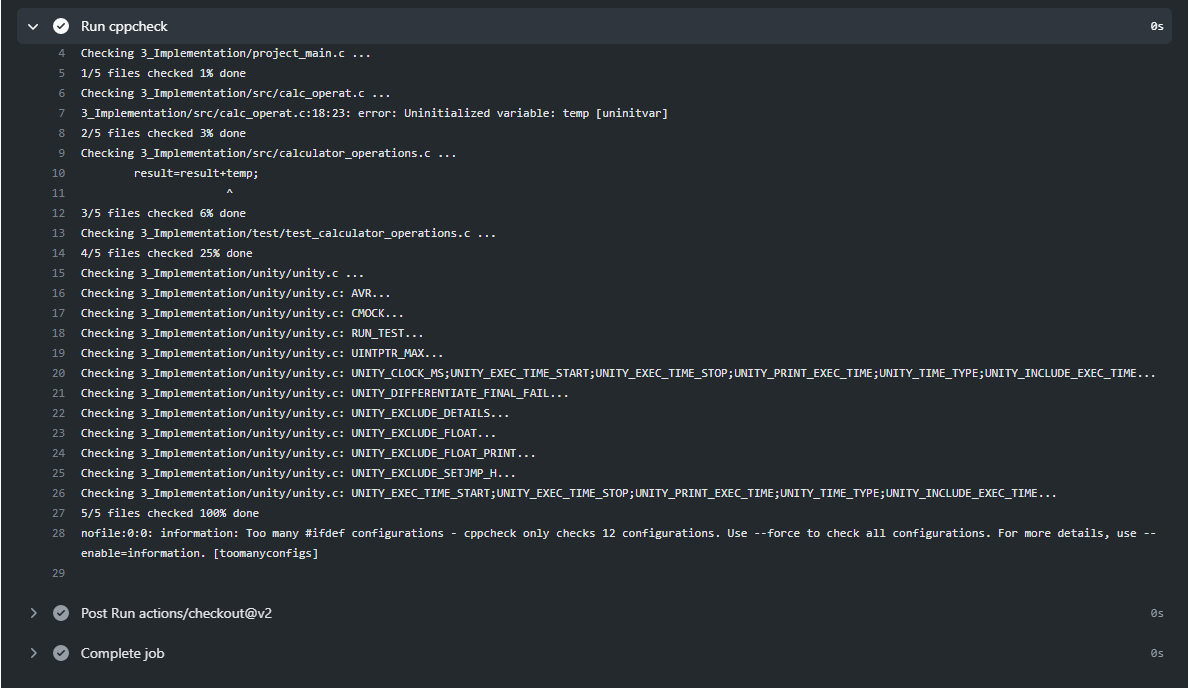
#### Code quality and Issues or Bug Tracking

**Setup Alignment**

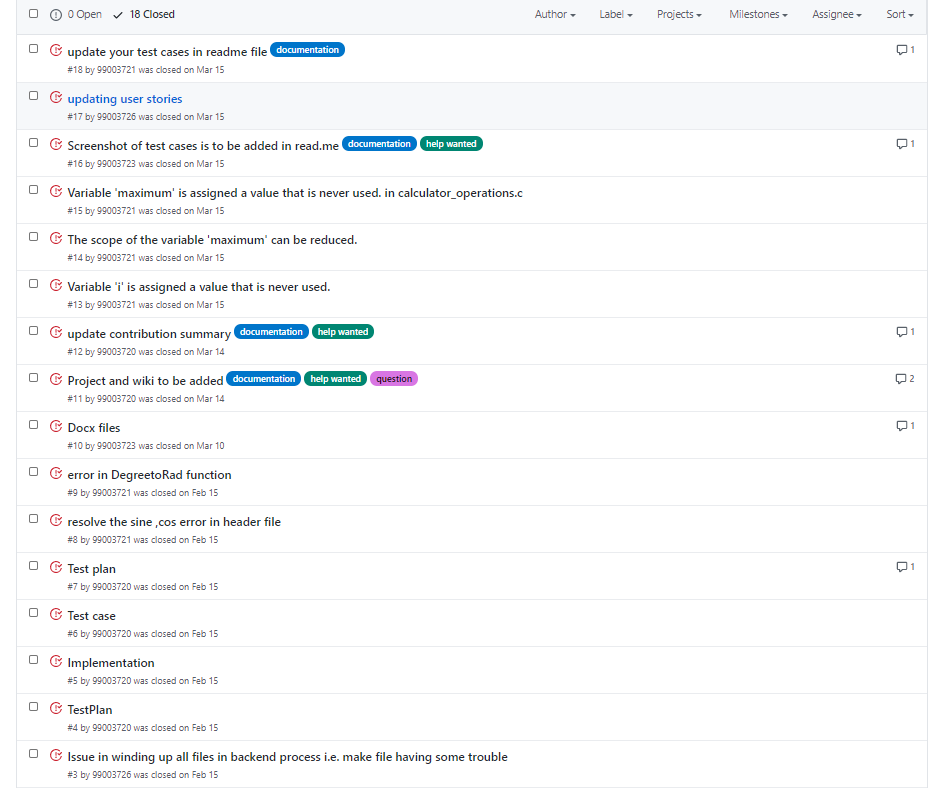


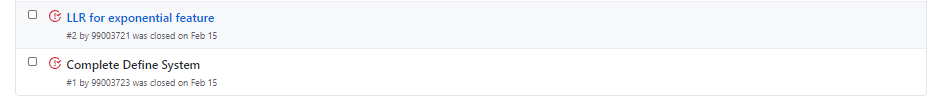
**Outcome**





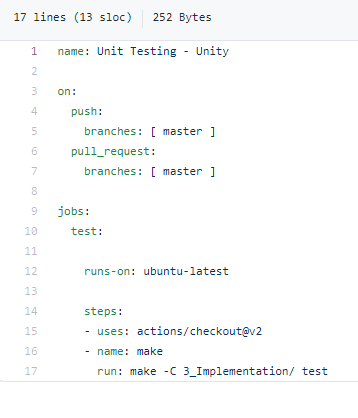
**Issues**



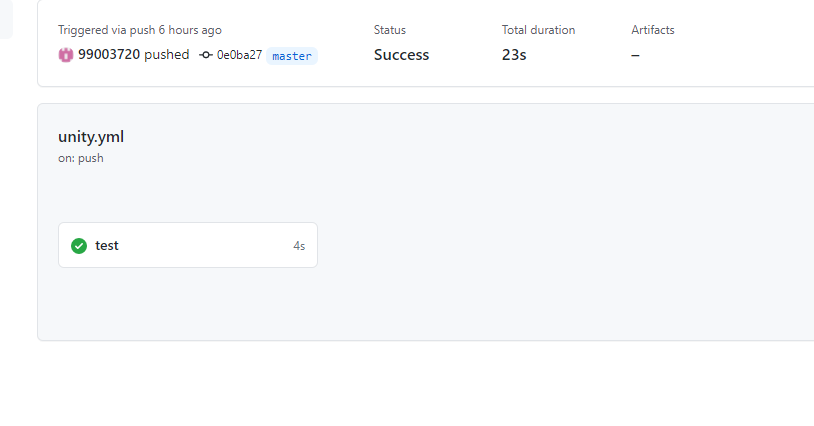


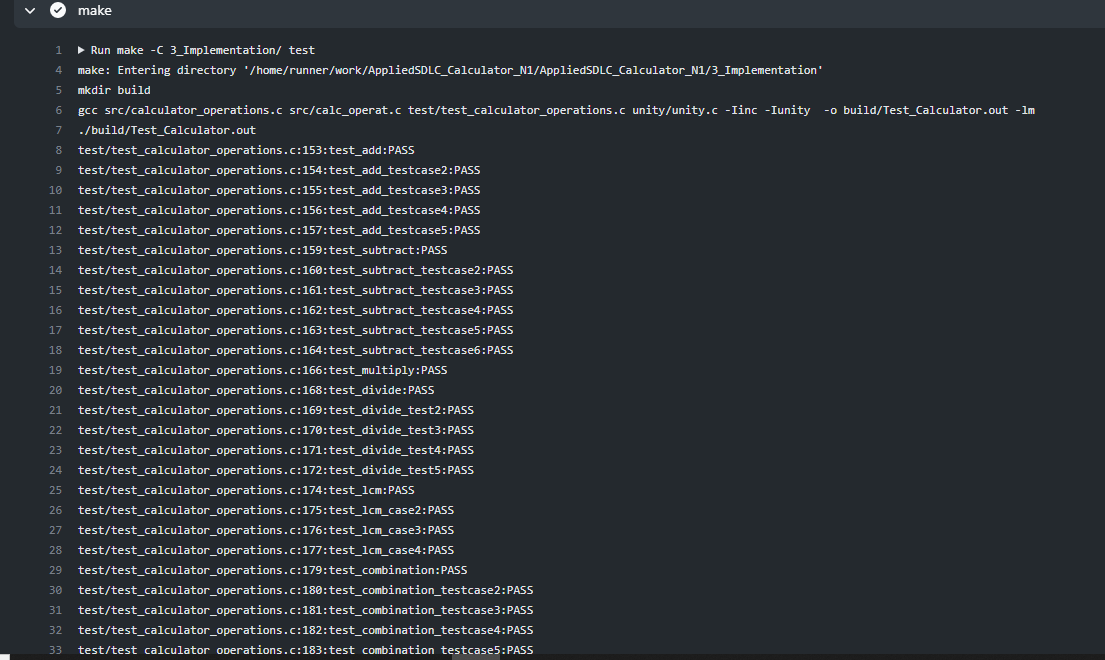
#### Unit Testing

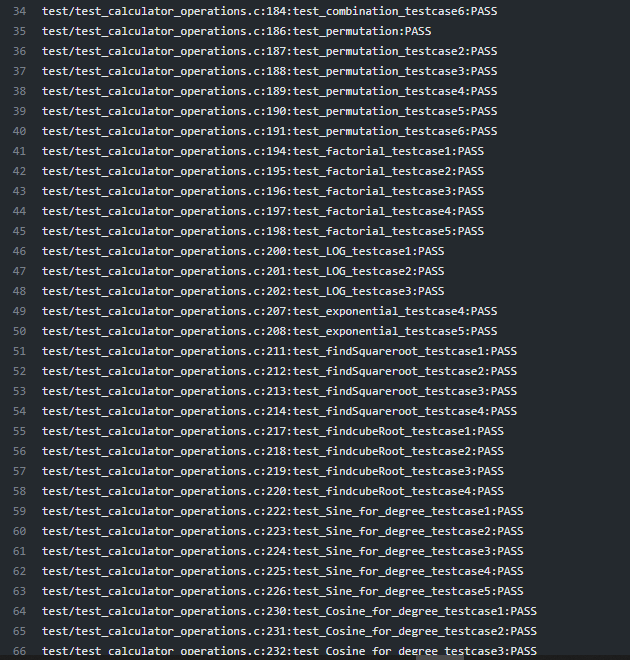
**Setup Alignment**

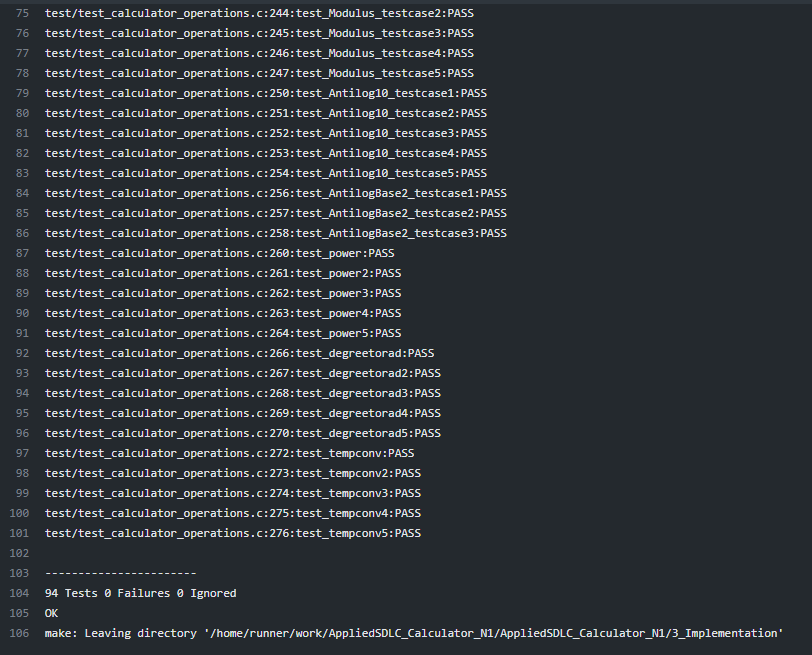


**Outcome**









## **Individual Contribution & Highlights**

The features implemented by me are:

Arithmetic Operations

* Requirements on various arithmetic operations as addition, subtraction, multiple and division.
* The low-level design for the operations
* Various test cases for the operations including all kinds of requirements

Permutation and Combination

* + - Requirements and design for the permutation and combination functions
    - Implemented code for permutation and combination of two positive numbers

LCM and GCD

* + - Requirements and design for the LCM and GCD functions
    - Implemented code for LCM and GCD of two positive numbers

### Summary

* Through this project, I learnt about team work. Communication between the team members was the most essential part of working in a team.
* I improved my C programming coding skills with number of different functions implemented by me and my team members.
* I improved my leadership skills by ownership and accountability of work done.

### Challenges faced and how were they overcome

1. Running the make file as its resolved by defining its correct path (.out for Linux and -lm for math functions)
2. Synchronizing the VS code to GitHub, colleague help to resolve the issue
3. Making the function call in correct path
4. Open git log while committing, thus went to GitHub desktop and pulled origin and then pushed origin.
5. Test case code for the boundary problem. Added code with the help of internet

### Future Scope

Every project can be well executed and properly defined when it is designed and deployed as per the Software Development Life Cycle.

# Creating a data sheet using Python [Individual]:

## **Modules**

The modules linked to the mini project is SDLC and Python.

### Topic and Subtopics

**Python Programming**

* + - Strings
    - Loops
    - Exception Handling
    - Classes and Objects

**Libraries**

* Pandas
* OpenPyxl
* OS

## Objectives & Requirements

Objective:

The objective of the project is to extract the data present in different spreadsheets in different directories as required by the user by different paths given by him. The program scrolls through all the spreadsheets with the following data common in all the sheets:

* Name
* PS Number
* Email Id

The corresponding data must then be printed on the console as well as written to the excel sheet.

**Requirements:**

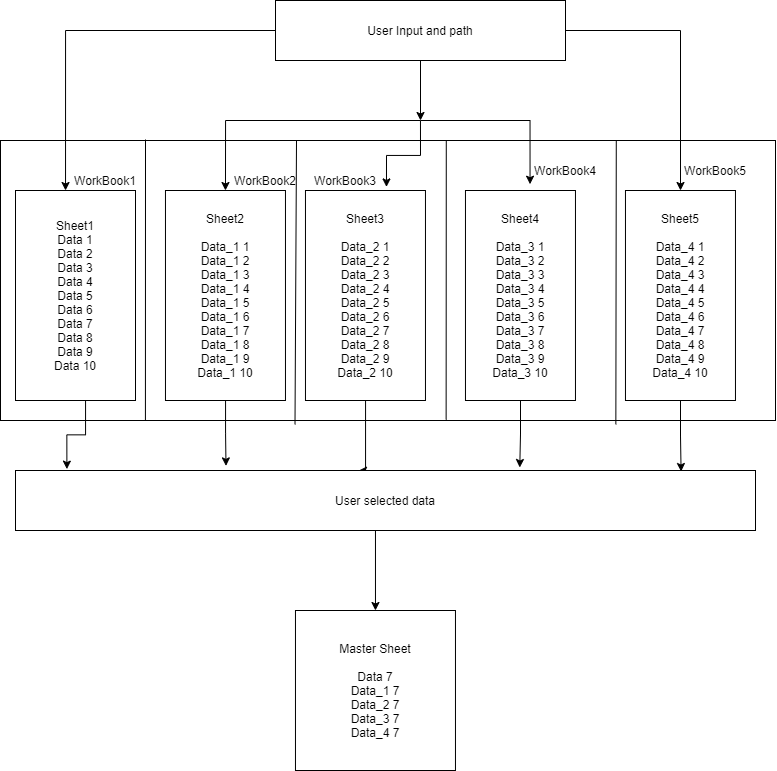
**High Level Requirements:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirements** | **Description** | **Status** |
| HL001 | Providing Path | Path for the excel data is provided by the user | IMPLEMENTED |
| HL002 | Searching Data | Search all data from 5 sheets when user defines the name, PS number and email ID to be searched. | IMPLEMENTED |
| HL003 | writing to excel | Write all the data from different spreadsheet in one master sheet in a new excel sheet. | IMPLEMENTED |
| HL004 | extracting user defined data | Write new user required data in the excel file. | IMPLEMENTED |

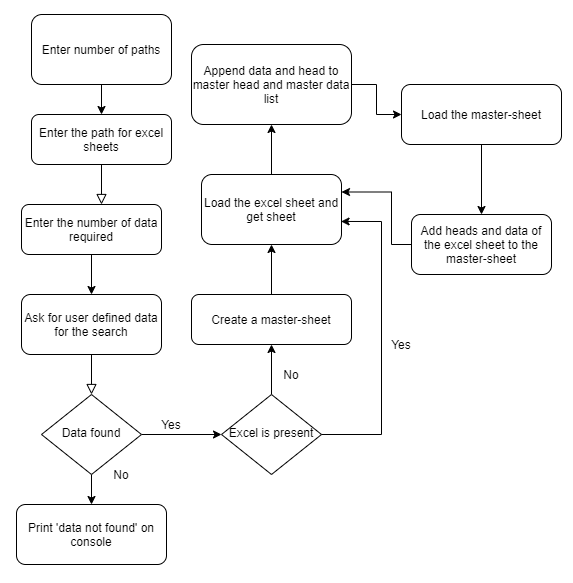
**Low Level Requirements:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirements** | **Description** | **Status** |
| LL001 | Searching Data by user | The data to be searched is defined by the user. | IMPLEMENTED |
| LL002 | Searching Data according to the path provided | Data is present in different excel files present in different directories | IMPLEMENTED |
| LL003 | Search Parameters | The user defines the Name, email ID and PS Number of the data to be searched in 4\*10 rows and columns per sheet | IMPLEMENTED |
| LL004 | Extracting user defined data | Data defined by user must be extracted from 5 different spreadsheets and put into one master sheet. | IMPLEMENTED |
| LL005 | Printing extracted data | 10 different data to be printed to the console. | IMPLEMENTED |

## **Design**



**Structural – Interaction Overview Diagram**



**Behavioral – Activity Diagram**

## **Test Plan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Description** | **Expected Input** | **Expected Output** | **Actual Output** |
| PY01 | The user defines the name in the common data for segregation of data required | Evan | The data corresponding to Evan in all excel sheets to a master sheet | The data corresponding to Evan in all excel sheets to a master sheet |
| PY02 | The user defines the PS Number in the common data for segregation of data required | 99003720 | The data corresponding to 99003720 in all excel sheets to a master sheet | The data corresponding to 99003720 in all excel sheets to a master sheet |
| PY03 | The user defines the Email ID in the common data for segregation of data required | evan.zigomalas@gmail.com | The data corresponding to Email ID in all excel sheets to a master sheet | The data corresponding to Email ID in all excel sheets to a master sheet |
| PY04 | The user defines the name and PS Number in the common data for segregation of data required | Evan and 99003720 | The data corresponding to name and PS Number in all excel sheets to a master sheet | The data corresponding to name and PS Number in all excel sheets to a master sheet |
| PY05 | The user defines the name and Email ID in the common data for segregation of data required | Evan and evan.zigomalas@gmail.com | The data corresponding to Evan in all excel sheets to a master sheet | The data corresponding to Evan in all excel sheets to a master sheet |
| PY06 | The user defines the PS Number and Email ID in the common data for segregation of data required | 99003720 and evan.zigomalas@gmail.com | The data corresponding to PS Number and Email ID in all excel sheets to a master sheet | The data corresponding to PS Number and Email ID in all excel sheets to a master sheet |
| PY07 | The user defines all the three, Name, PS Number and Email ID in the common data for segregation of data required | [evan.zigomalas@gmail.com](mailto:evan.zigomalas@gmail.com), 99003720 and Evan | The data corresponding to Name, PS Number and Email ID in all excel sheets to a master sheet | The data corresponding to Name, PS Number and Email ID in all excel sheets to a master sheet |
| PY08 | Incorrect path is provided by the user. | C:\Users\99003720\Desktop\pythonProject | Path not found | Path not found |
| PY09 | Incorrect common data provided | Jasleen and 99003720 | DATA NOT FOUND | DATA NOT FOUND |

## **Implementation Summary**

The user defines the common data that he can provide for the data to be searched. This data can be:

* Name
* PS Number
* Email ID

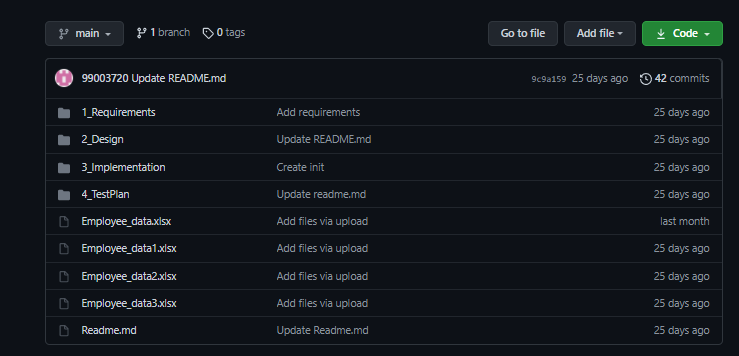
The python program then reads the data corresponding to the data from different spreadsheets of excel present according to the path of the main excel sheet provided by the user. It then creates a master sheet in another excel file and adds the data from all the sheets to it. In the end, the data to be provided to the user is printed to the console as well as written to the excel.

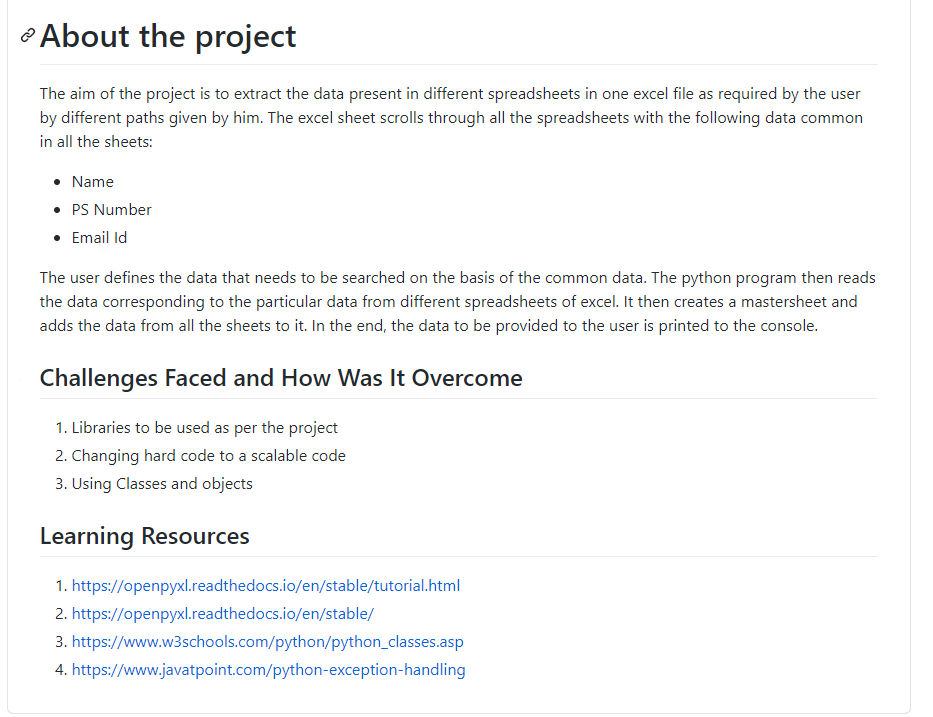
### 

### Git Link

https://github.com/99003720/Python\_mini\_project

### Git Dashboard





### Summary

In this python project, 3 spreadsheets contain employee data in 40 rows and 10 columns. The python program asks for the number of data that needs to be put to the master sheet. The user provides one, two or three of the common data – Name, PS Number and Email ID. He also provides the path of all the spreadsheets that has all the data. The python program then searches the data according to the data provided. It then creates a master sheet and adds the corresponding data to the master sheet until the number of data to be written are over. All the data is finally printed to the console and written to the excel sheet.

### Challenges faced and how were they overcome

1. Libraries that can be used for implementation of the project. Referred internet to know the best two libraries, Pandas and OpenPyxl
2. Changing the hard code to an easily scalable code. Referred internet to add different functions and classes to make it scalable.
3. Using classes and objects for a scalable function.

# Embedded C [Team]

## Module

The module linked to the mini project is Embedded C.

### Topic and Subtopics

GPIO Programming

ADC, USART, SPI and I2C Protocols

Hardware Abstraction Layer Application interface

## **Objectives & Requirements**

**Objective:**

The objective of this project is to implement the features of the BCM Module using STM32F407VG microcontroller with 32-bit ARM-cortex-M4 FPU core.

**Requirements:**

**High Level Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirements** | **Description** | **Status** |
| EmC1 | Module enable | Module is enabled when set | IMPLEMENTED |
| EmC2 | Implementation of Door Control module | LED on and Buzzer beeps | IMPLEMENTED |
| EmC3 | Implementation of seat Control module | Led is lit when control is set | IMPLEMENTED |
| EmC4 | Implementation of Power Window module | RGB turns pink when control is set | IMPLEMENTED |
| EmC5 | Implementation of Wiper control module | Moisture sensor senses water in wiper and LED is lit | IMPLEMENTED |
| EmC6 | Implementation of Headlight control module | Led is lit and LDR senses darkness when control is set. Headlight turns on. | IMPLEMENTED |
| EmC7 | Implementation of AC Conditioning Control module | Led is lit when control is set | IMPLEMENTED |

**Low Level Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirements** | **Description** | **Status** |
| EmL0 | The module is enabled | Module is enabled when GPIO Pin 14 is set | IMPLEMENTED |
| EmL1 | LED glows when seat Control module is enabled. | LDR senses darkness. Orange LED when the control is set. | IMPLEMENTED |
| EmL2 | Buzzer beeps and LED glows when Door Control module is enabled. | Green LED is lit and Buzzer beeps when the control is set. | IMPLEMENTED |
| EmL3 | LED glows when seat Control module is enabled. | Green LED when the control is set. | IMPLEMENTED |
| EmL4 | Moisture sensor is enabled when Wiper control module is enabled. | Moisture sensor senses the water on windshield & Green LED is lit if Wiper control module is set. | IMPLEMENTED |
| EmL5 | RGB is enabled when Power window module is enabled and window opens | RGB turns pink when the control is set. | IMPLEMENTED |
| EmL6 | LED glows when AC Conditioning Control module is enabled. | Blue LED when the control is set | IMPLEMENTED |

## **Design**

## **Test Plan**

**High Level Test Plan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Description** | **Expected Input** | **Expected Output** | **Actual Output** |
| TestEm1 | Module is enabled | Module is turned on | Module gets enabled | Module gets enabled |
| TestEm2 | Buzzer beeps and LED glows when door is open and turns off when door closes | Push button pressed indicating door is open | Green LED glows | Green LED glows |
| TestEm3 | LED glows when seat is shifted front or back and turns off when the module disables | Push button pressed indicating seat is shifted front or back | Buzzer beeps and green LED glows | Buzzer beeps and green LED glows |
| TestEm4 | Moisture sensor senses and wiper control enables when water turns on, on top of the windshield. | Push button pressed indicating wiper is enabled | When water is on windshield, moisture sensor senses and turns LED on indicating wiper control module is enabled | When water is on windshield, moisture sensor senses and turns LED on indicating wiper control module is enabled |
| TestEm5 | Orange LED glows when headlight is on as LDR senses darkness and turns off when headlight turns off | Push button pressed indicating headlight is on | LDR senses darkness and orange LED turns on when headlight turns on | LDR senses darkness and orange LED turns on when headlight turns on |
| TestEm6 | LED glows when AC turns on and turns off when AC is off | Push button pressed indicating seat is AC is turned on | Blue LED turns on when AC is on | Blue LED turns on when AC is on |
| TestEm7 | LED glows when power window is enabled | Window is open | RGB Turns Pink when window is open | RGB Turns Pink when window is open |

**Low Level Test Plan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Description** | **Expected Input** | **Expected Output** | **Actual Output** |
| LteEm1 | Module is enabled when GPIO Pin 6 is set | GPIO Pin 6 is set |  |  |
| LteEm2 | Green LED turns on and buzzer beeps when door module is enabled | Push button is pressed indicating Door is open | Green LED is On and Buzzer starts beeping. | Green LED is open |
| LteEm3 | LED turns off and buzzer stops beeping when door closes | Push button is pressed indicating Door is closed | Green LED turns off and buzzer stops beeping | Green LED turns off and buzzer stops beeping |
| LteEm4 | Green LED turns on when seat is adjusted front or back | Push button is pressed indicating seat shifts front or back | Green LED Glows. | Green LED Glows. |
| LteEm5 | LED stops glowing when the seat is adjusted | Push button is released indicating seat is adjusted | Green LED is turned off. | Green LED is turned off. |
| LteEm6 | Moisture sensor senses water when the wiper control module is enabled and water starts on the windshield. LED glows. | Moisture sensor  To sense water. | Green LED Glows. | Green LED Glows. |
| LteEm7 | LED turns off when the water stops on the windshield. | Moisture sensor  To sense water. | Green LED is turned off. | Green LED is turned off. |
| LteEm8 | RGB turns pink indicating the headlight is on | Push button is pressed indicating Window is closed | Two LEDs glow in the RGB module. | Two LEDs glow in the RGB module. |
| LteEm9 | RGB stops when the headlight turns off | Push button is released indicating Window is closed | LEDs turned off. | LEDs turned off. |
| LteEm10 | LED glows when AC is on | Push button is pressed indicating AC is ON. | Blue LED Glows. | Blue LED Glows. |
| LteEm11 | LED turns off when AC is off | Push button is pressed indicating AC is OFF. | Blue LED is turned off. | Blue LED is turned off. |

## **Implementation Summary**

In this project, we have implemented six functions of a BCM on STM board.

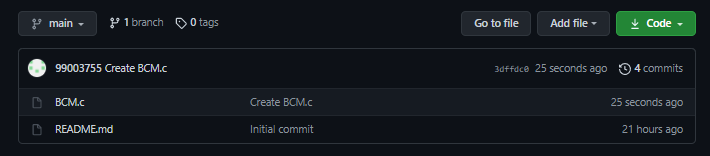
Red LED – On, is used as a Default state which indicates that the Engine is On.

1. Seat Control: In this module GPIO Pin 0 of port A, connected to a push button is used to indicate the status of the seat. If the push button is pressed it signifies that the seat can move left or right and correspondingly Green LED glows. When switch is released the Green LED is Turned Off.
2. Door Control: A push button connected to a GPIO pin, port A pin 0 is used to indicate the Door Status. A Buzzer (connected to port D pin 11) & LED is used to notify about the Door Status. When the Door is open i.e. The push button is pressed, The Green LED is turned on and the buzzer starts to beep. When the push button is released the buzzer and LED is turned off.
3. Power Window: A push button connected to a GPIO pin is used to indicate the Power Window Status. Once the button is pressed two LED’s in RGB module (Connected to port C, pin 7 & 8) is turned ON. When the button is released the LED is turned off.
4. Wiper Control: In this module, a Moisture sensor is used as input. It is connected to port C pin 11. When water is sensed by the sensor i.e. Port C pin 11 value reads value 0, Green LED is turned on which indicates that the Wiper is turned ON. When the value is 1, the LED is turned OFF.
5. Headlight Control: In this module, a LDR sensor (connected to port D, pin 10) is used. When the LDR sensor does not get any light, it indicates that the headlight must be turned on, here the Blue LED is turned on to indicate the headlight. When LDR sensors detects light, Blue LED is turned off.
6. AC Control: In this module, a push button is used to indicate the input to AC module. When the push button is pressed Orange LED glows indicating that the AC is ON. When push button is released the LED is turned off, which indicates that the AC is Off.

### Git Link

<https://github.com/99003755/Embedded_C.git>

### Git Dashboard



## 

## Individual Contribution & Highlights

The features implemented by me are:

Power Window

AC Conditioning

I worked in integrating all the six modules and the hardware implementation for the same.

### Summary

In the project, we have replicated the BCM module using STM32F407VG microcontroller with 32-bit ARM-cortex-M4 FPU core. We have implemented and integrated 6 features:

* Seat Control Module
* Door Control Module
* Power Window
* AC Conditioning
* Headlight Control Module
* Wiper Control Module

The functioning of these modules is replicated using indicators connected to the board such as buzzer, RGB, Moisture sensor and LEDs of different colors.

### Challenges faced and how were they overcome

1. Accessing the pin configuration for the STM32F407VG
2. Integrating all the features on one STM Board

# Linux and OS programming & Kernel [Individual]

## **Modules**

The modules linked to the mini project is Linux and OS programming & Kernel Device Drivers

### Topic and Subtopics

## **Objectives**

The objective of this project is to implement the learning of the module in the following activities:

* Making a System call to echo back the given string
* Making a System call to traverse process list and print PID and PPID of a process.
* Making a System call to retrieve the attributes of calling process
* Implementing IOCTL to echo back the string, implement two operations such that the string passed by one operation is retrieved by the other operation
* Implementing IOCTL code such that one operation receives an integer, create a list node and append, the other operation traverses the list. The exit method will clean up the nodes.
* Implementing three kernel threads where the first two threads will write N chars each into Kfifo. The third thread waits for first two threads and then retrieves data from Kfifo
* Applying mutual exclusion between first two threads by implementing three kernel threads. First two threads will write N nodes into a kernel list. The third thread waits for the first two threads and traverses the list while mutual exclusion is applied between the first two threads. Exit method cleanup the list nodes

## **Requirements**

**High Level Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirements** | **Description** | **Status** |
| KDD1 | System Call to echo back string | The system call echoes back given string from user space | IMPLEMENTED |
| KDD2 | System Call to traverse process list and print PID and PPID | The System Call traverse the process list and prints PID and PPID on Qemu | IMPLEMENTED |
| KDD3 | System Call to traverse process list and retrieve attributes of the calling process | The system call retrieves the attributes of the process to Qemu | IMPLEMENTED |
| KDD4 | IOCTL to echo back string using two operations | Use IOCTL to echo back string by implementing 2 operations, one for passing string and other of for retrieving it. | IMPLEMENTED |
| KDD5 | IOCTL to create list, traverse and cleanup nodes | Implementing IOCTL to create list of N nodes and traversing the list. Using exit method to clean up the nodes. | IMPLEMENTED |
| KDD6 | Threads implementation of 3 threads where 3rd third retrieves data for first 2 threads | 2 threads will write N data to the list. Third waits for the 2 threads and then retrieves data from Kfifo | IMPLEMENTED |
| KDD7 | Threads implementation of 3 threads where 3rd third retrieves data for first 2 threads with list of N nodes and its traversal | Applying mutual exclusion to implement 3 threads where 2 write N nodes to threads and 3rd traverses the list. Exit module cleans up the node. | IMPLEMENTED |

**Low Level Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirements** | **Description** | **Status** |
| LKDD1 | The string is echoed back to Qemu by system call | The string “Hi” gets echoed to Qemu by creating a system call invoked by the user space. | IMPLEMENTED |
| LKDD2 | The process is traversed and PID and PPID is retrieved by system Call | User invokes a system call to get PID and PPID of the process. | IMPLEMENTED |
| LKDD3 | Attributes of a process is retrieved by system call | User invokes a system call to retrieve the attributes of the process. | IMPLEMENTED |
| LKDD4 | IOCTL used to echo back string with 2 operations | IOCTL to echo back string using 2 operations one to pass the argument and other to retrieve data and display it on Qemu. | IMPLEMENTED |
| LKDD5 | IOCTL to create, traverse and clean up nodes of list | IOCTL to using 2 operations one to create list of N nodes and other to traverse it and display it on Qemu. Exit method to clean up the nodes. | IMPLEMENTED |
| LKDD6 | 3 thread implementations where 3rd third retrieves data for first 2 threads in KFIFO | first two threads will write N chars each into Kfifo. The third thread waits for first two threads and then retrieves data from Kfifo | IMPLEMENTED |
| LKDD7 | Mutual exclusion of three threads to traverse, list and clean up nodes | First two threads will write N nodes into a kernel list. The third thread waits for it and traverses the list. Mutual exclusion applied between the first two threads. Exit method cleanup the list nodes | IMPLEMENTED |

## **Test Plan**

**High Level Test Plan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Description** | **Expected Input** | **Expected Output** | **Actual Output** |
| KDT1 | The string passed by the user is echoed to Qemu when invoked | “Hi” as string passed in userspace | “Hi” is printed in Qemu | “Hi” is printed in Qemu |
| KDT2 | The PID and PPID of the process is retrieved in Qemu when user space code is invoked | User space invocation for the system call | PID and PPID is printed in Qemu | PID and PPID is printed in Qemu |
| KDT3 | The attributes of the process are retrieved in Qemu when user space code is invoked | User space invocation for the system call | Process PID, Process State, PPID, Priority, RT Priority, Static Priority and Normal priority are printed in Qemu | Process PID, Process State, PPID, Priority, RT Priority, Static Priority and Normal priority are printed in Qemu |
| KDT4 | Two operations are invoked by IOCTL. String invoked by the user in 1st operation is echoed in the 2nd operation. | User space invocation of the IOCTL functions | String is printed in qemu by invoking IOCTL functions | String is printed in qemu by invoking IOCTL functions |
| KDT5 | Two operations are invoked by IOCTL. List is created by the user in 1st operation and traversed in the 2nd operation. Nodes are cleaned up in the exit method. | User space invocation of the IOCTL functions | List is created of N nodes when init method is executed. Exit Method cleans up the nodes and displays "Buffer is empty” | List is created of N nodes when init method is executed. Exit Method cleans up the nodes and displays "Buffer is empty” |
| KDT6 | 3 threads are implemented. 1st and 2nd thread creates a list of N nodes, 3rd traverses the list after waiting for 2 threads to complete. | Invocation of the code | 3rd thread waits for 1st 2 threads to create a list through N nodes. 3rd thread then traverses the list and adds in Kfifo. | 3rd thread waits for 1st 2 threads to create a list through N nodes. 3rd thread then traverses the list and adds in Kfifo. |
| KDT7 | 3 threads are implemented. 1st and 2nd thread creates a list of N nodes, 3rd traverses the list after waiting for 2 threads to complete by mutual exclusion. | Invocation of the code | 3rd thread waits for 1st 2 threads to create a list through N nodes by mutual exclusion. 3rd thread then traverses the list. Exit method cleans up all the nodes. | 3rd thread waits for 1st 2 threads to create a list through N nodes by mutual exclusion. 3rd thread then traverses the list. Exit method cleans up all the nodes. |

## **Implementation Summary**

### Git Link

[99003720/Kernal\_Programming (github.com)](https://github.com/99003720/Kernal_Programming)

### Git Dashboard

### Summary

#### Git inspector summary

#### Issues

### Summary

### Challenges faced and how were they overcome