GENESIS - Learning Outcome & Mini-project Summary Report

LTTS GLOBAL ENGINEERING ACADEMY

L&T Technology Services

Details

Ver. Rel. No.	Release Date	Prepared By	Reviewed By	To Be Approved	Remarks/Revision Details
1.0	17/02/2022	Vaishnavi kotala 40021047			

Contents

Contents

Contents....3
Mini Project : library management system [Individual]....7
Modules :....7
Requirement....7
Introduction....7
Research....7
SWOT Analysis....7
4W's and 1'H....8
Who:....8
What:....8
When:....8
Where:....8
How:....8
Cost and Features and Timeline....8
Defining Our System....9
Hardware requriement....9
Software interface:....9
Communication interface:....9
Detail requirements....9
High Level Requirements....10
Low Level Requirements....10
Design....11
Test Plan....12
High Level Requirements....12
Low Level Requirements....13
Implementation and Summary Git Link:....13
Git Dashboard....13
Summary....14
Git Inspector Summary....14
Individual Contribution & Highlights:....14
Challenges Faced and How Was It Overcome....15

Mini Project 2 – Embedded Seat Heat Monitor and Control System [Individual]....................................15
Modules.. 15
Heat Control System 15
SWOT ANALYSIS... 16
STRENGTHS 16
WEAKNESS... 16
Requirements 17
Design 18
OFF 18
Images and Videos: 20
ON...21
Git Dashboard 23
Mini Project 3 – Scientific Calculator [Team] 24
Modules.. 24
Low Level Design... 31
SEQUENCE DIAGRAM FOR CONVERSI 33
6_Images_And_Videos... 38
Github Link: https://github.com/GENESIS2021Q1/Applied_SDLC-Dec_Team_48.git 39
Learning Resources 40
Individual Contribution and Highlights 40
Mini Project 4 – Calendar Automation [Team]..41
Modules.. 41
Requirements 41
High Level Requirements 41
Low Level Requirements.. 42
Link for standard input template:...42
Requirements for updating Master calendar using Master calendar as input: 42
App deployment:..43
Additional features for V1 to do: 43
Github link: https://github.com/Usharani8/Oopswithpython_Calendar_Automation_Team-47.git 43
Individual contribution and Highlights:.. 43
Mini Project 5 – Ford Project [Team] 44
Modules.. 44
Requirements 44
Introduction... 44
DESIGN..45

High Level Requirements 46
Low Level Requirements 46
Design 47
OUTPUT 48
Team Activity 49
Mini Project 6 – Wiper Control [Team] 50
Modules 50
Requirements 50
4W's and 1'H 50
Who: 50
What: 50
When: 50
Where: 50
How: 50
High Level Requirements 50
Low Level Requirements 52
Design 52
Test Plan 54
High Level Test Plan 54
Low Level Test Plan 54
Implementation and Summary Git Link: 55
Mini Project 7 – LAMBHORGINI[Team] 56
Module: 56
Requirements: 56
Research: 56
SWOT- Strengths, and Weakness, Opportunities Threats: 56
4W's and 1'H: 56
Detail requirements: 57
High Level Requirements: 57
Low Level Requirements: 57
Design: 58
Implementation and Summary: 58
Individual Contribution and Highlights: 58
Role in Project Team: 58
Mini Project 8 – EV Car [Team] Modules 59
Module: 59

Requirements: 59
1.Battery: 59
Battery Features: 59
2.Motor: 59
Motor Features: 59
3.Controller: 59
controller Features: 59
4.Inverter: 60
Inverter features: 60
Implementation and Summary: 60
Individual Contribution and Highlights 60
Role in Project Team: 60
Mini Project 9 –Seatbelt [Individual] 61
Modules 61
High level Requirements 61
Number 61
Requirements 61
Description 61
HLR1 61
Pressure control alarm 61
alarm buzzes when seat belt is not buckled up even the pressure is applied 61
HLR2 61
Open airbag 61
depending upon the tension airbag will be opened 61
HLR3 61
buckles and its types 61
lap belts, diagonal belts, the harness, combination of lap and diagonal belts.... etc. 61
HLR4 61
Height adjustment of seat belt 61
Height adjustment of seat belt 61
Low level requirement 61
Design 62
Implementation and Summary Git Link: 62

Mini Project : library management system [Individual]

Modules :

1. C Programming
2. Git

Requirement

Introduction

The purpose of a library management system is to operate a library with efficiency and at reduced costs. The system being entirely automated streamlines all the tasks involved in operations of the library. The activities of book purchasing, cataloging, indexing, circulation recording and stock checking are done by the software. Such software eliminates the need for repetitive manual work and minimizes the chances of errors.

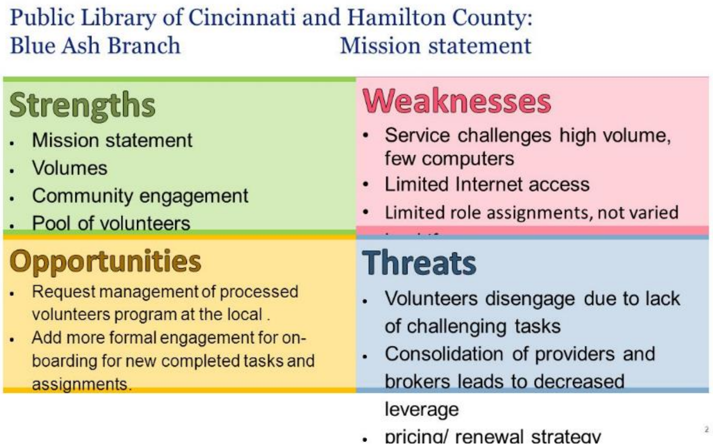
The purpose of this application are as follows:

- The software is for automation of library.
- It provides following facilities to Operator.
- Can enter details related to a particular book.
- Can enter details related to a particular student/user.
- Can read and write information about any member.
- Can update, create, delete the record of membership as per requirement and implementation plants.

Research

- With the advancement of technology, it is imperative to exalt all the systems into a user-friendly manner. The Library Management system (LMS) acts as a tool to transform traditional libraries into digital libraries. In traditional libraries, the students/user has to search for books which are hassle process and there is no proper maintenance of database about issues/fines. The overall progress of work is slow and it is impossible to generate a fast report.
- The librarians have to work allotted for arranging, sorting books in the book sells. At the same time, they have to check and monitor the lend/borrow book details with its fine. It is a tedious process to work simultaneously in different sectors. LMS will assist the librarians to work easily. The LMS supports the librarians to encounter all the issues concurrently. The users need not stand in a queue for a long period to return/borrow a book from the library. The single PC contains all the data's in it. The librarians have to assess the system and provide an entry in it. Through LMS the librarian can find the book in the bookshelves.

SWOT Analysis



Public Library of Cincinnati and Hamilton County:
Blue Ash Branch Mission statement

Strengths

- Mission statement
- Volumes
- Community engagement
- Pool of volunteers

Weaknesses

- Service challenges high volume, few computers
- Limited Internet access
- Limited role assignments, not varied

Opportunities

- Request management of processed volunteers program at the local .
- Add more formal engagement for on-boarding for new completed tasks and assignments.

Threats

- Volunteers disengage due to lack of challenging tasks
- Consolidation of providers and brokers leads to decreased leverage
- pricing/ renewal strategy

2

4W's and 1'H

Who:

- Library management system is also useful for students as well as a librarian to keep the constant track of the availability of all books in a store.

What:

- A library management system is software that is designed to manage all the functions of a library.

When:

- When there had large number of data i.e, huge number of books information.

Where:

- The Library management system is nowadays essential for schools, colleges, private libraries, and other organizations.

How:

- A Library management system is a software that uses to maintain the record of the library

Cost and Features and Timeline

- A modern integrated library management system (LMS).
- Can be scalable to Windows, Linux and Mac OS platform.

- Full catalog, circulation and acquisitions system for library stock management.
- . Simple, clear search interface for all users.
- . Multilingual and multi-user support.
- . Export and import records.
- . Easy way to enter new books.
- . Keep record of complete info of a book like title, author name, publisher name etc.
- . Easy way to check-in and check-out.
- . It is user-friendly software.
- . Keep record of different categories.
- . Classify the books subject wise.
- . Easy to maintain database.
- It is cost effective

Defining Our System

Hardware requriement

- Operating system: window
- Hard disk:40 GB)
- RAM: 256 MB
- Processor: Pentium(R)Dual-core CPU

Software interface:

- c language

Communication interface:

- Window
- The computer's capability depends on the software's efficiency. Provided the database size is big enough, the program will take any amount of inputs. This would be dependent on the memory space available.

Detail requirements

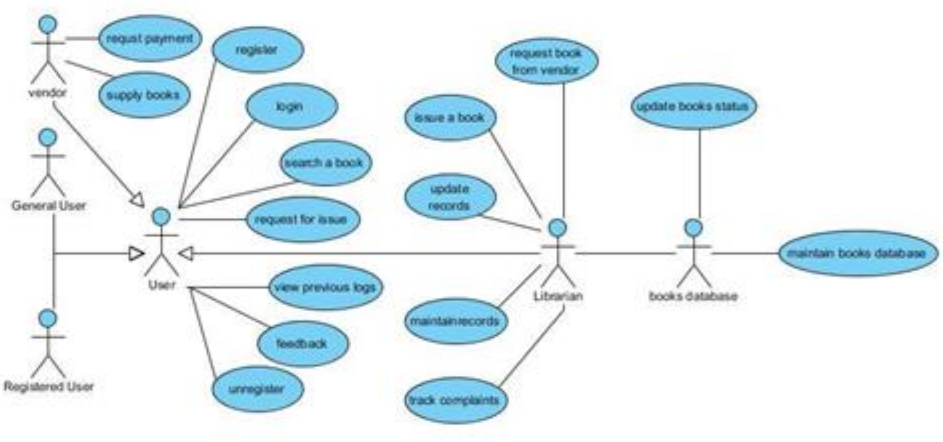
High Level Requirements

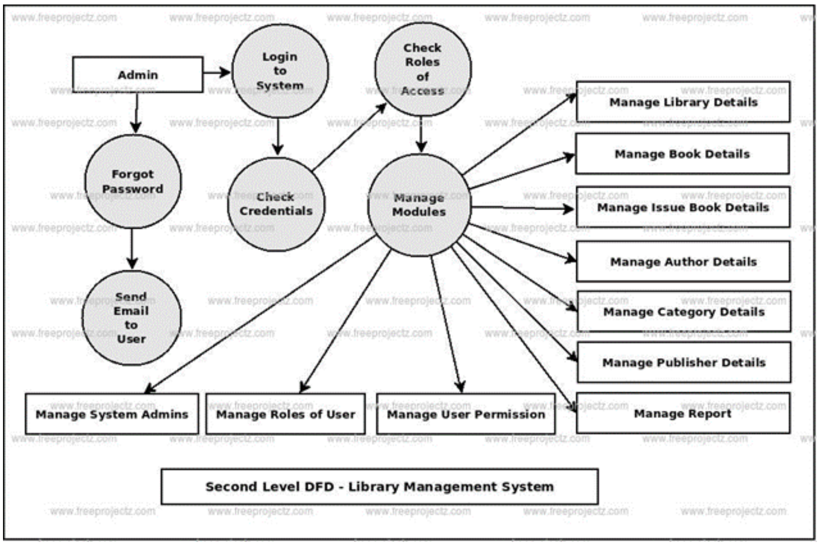
ID	Description	Status
HLR 1	The user must know the number of shelves in the library	Implemented
HLR 2	Same Id's for 2 or more books shall not be allowed.	Implemented
HLR 3	Only authentic user must have the access to the system	Implemented

Low Level Requirements

ID	Description	Status
LLR 1	User must enter issue and return date in database	Implemented
LLR 2	User must have the knowledge about the no of copies of a book	Implemented

Design


vendor
requst payment
supply books
General User
Registered User
User
register
login
search a book
request for issue
view previous logs
feedback
unregister
issue a book
update records
request book from vendor
maintainrecords
track complaints
Librarian
update books status
books database
maintain books database



Second Level DFD - Library Management System

Test Plan

High Level Requirements

Test ID	Description	Exp I/P	Exp O/P	Actual Out	Type Of Test
H_01	Components Testing	6	exit	exit	Requirement Based
H_02	The inputs required is given	7	exit	exit	Scenario Based
H_03	System Testing	one	exit	fail	Boundary Based

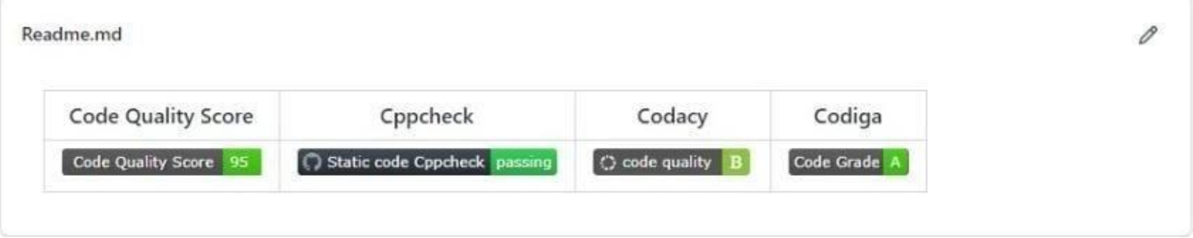
Low Level Requirements

Test ID	Description	Exp I/P	Exp O/P	Actual Out	Type Of Test
L_01	The components required for the project is tested for the further process	2	pass	pass	Requirement Based
L_02	The software used for the sysytem is tested and if it is passed in the test it is used for the further process	Three	pass	fail	Scenario Based
L_03	Overall system is tested and then it is approved for the next process	10	pass	Fails	Boundary Based

Implementation and Summary Git Link:

Link: https://github.com/kotalavaishnavi/M1_App_LibraryManagement

Git Dashboard

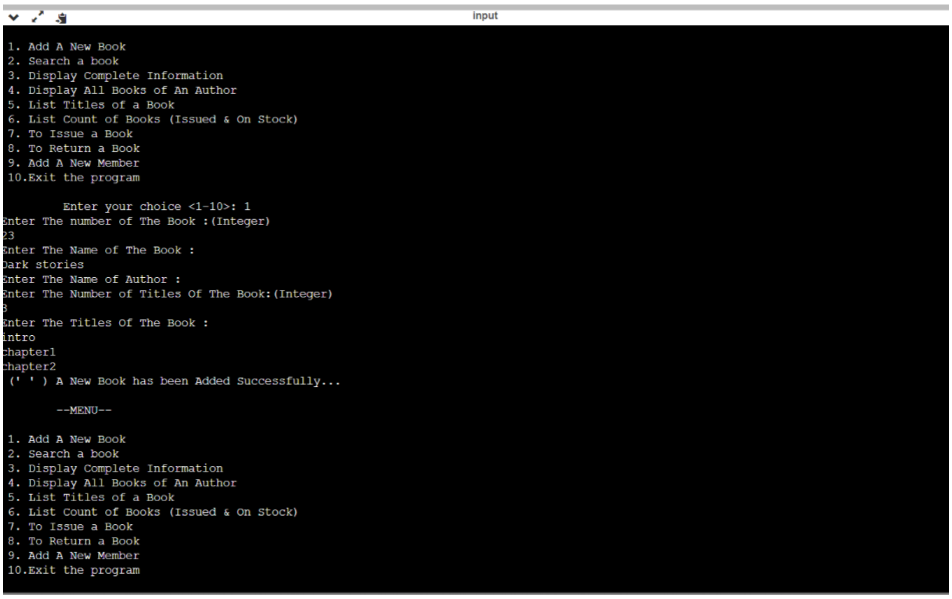


Readme.md

Code Quality Score	Cppcheck	Codacy	Codiga
Code Quality Score 95	Static code Cppcheck passing	code quality B	Code Grade A

Summary

Git Inspector Summary

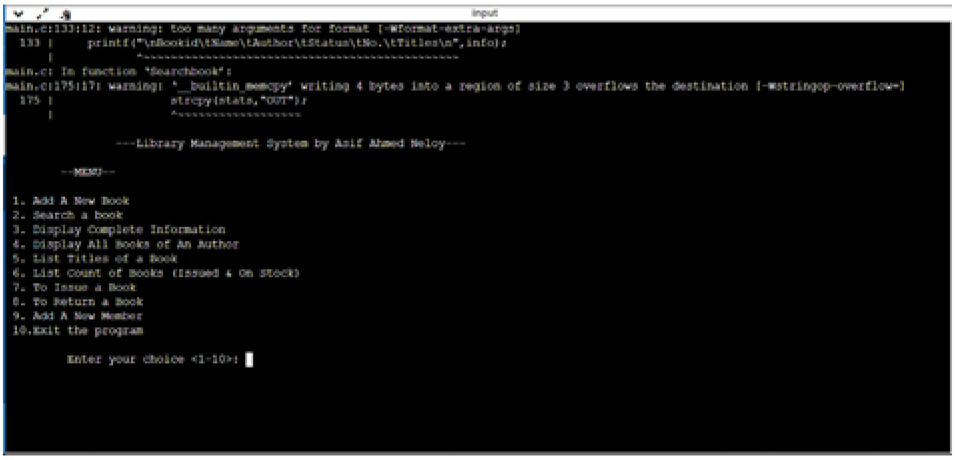


```
1. Add A New Book
2. Search a book
3. Display Complete Information
4. Display All Books of An Author
5. List Titles of a Book
6. List Count of Books (Issued & On Stock)
7. To Issue a Book
8. To Return a Book
9. Add A New Member
10.Exit the program

        Enter your choice <1-10>: 1
Enter The number of The Book :(Integer)
23
Enter The Name of The Book :
Dark stories
Enter The Name of Author :
Enter The Number of Titles Of The Book:(Integer)
3
Enter The Titles Of The Book :
intro
chapter1
chapter2
 (' ' ) A New Book has been Added Successfully...

       --MENU--

1. Add A New Book
2. Search a book
3. Display Complete Information
4. Display All Books of An Author
5. List Titles of a Book
6. List Count of Books (Issued & On Stock)
7. To Issue a Book
8. To Return a Book
9. Add A New Member
10.Exit the program
```



```
main.c:133:12: warning: too many arguments for format [-Wformat-extra-args]
  133 |     printf("\nBookid\tName\tAuthor\tStatus\tNo.\tTitles\n",info);
      |            ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
main.c: In function 'Searchbook':
main.c:175:17: warning: '__builtin_memcpy' writing 4 bytes into a region of size 3 overflows the destination [-Wstringop-overflow=]
  175 |                 strcpy(stats,"OUT");
      |                 ^~~~~~~~~~~~~~~~~~~

              ---Library Management System by Asif Ahmed Neloy---

       --MENU--

1. Add A New Book
2. Search a book
3. Display Complete Information
4. Display All Books of An Author
5. List Titles of a Book
6. List Count of Books (Issued & On Stock)
7. To Issue a Book
8. To Return a Book
9. Add A New Member
10.Exit the program

        Enter your choice <1-10>: ▌
```

Individual Contribution & Highlights:

Summary

- Improved implementation of C programming.
- Implementation of SDLC .
- V model and Agile methodology.
- Source code management using GitHub.

Challenges Faced and How Was It Overcome

1. Working with vsc, learnt from training.
2. Debugging learnt from buddy team.
3. Multifile, got good knowledge from couse material.
4. Building workflow found to be difficult task, learnt from online source.

Mini Project 2 – Embedded Seat Heat Monitor and Control System [Individual]

Modules

1. C Programming
2. Embedded System
3. SimulIDE
4. Git

Heat Control System

Theory

The heat control system is basically used to control the temperature of a car seat. When a user or driver of the car gets seated on a car, the button sensor gets activated. After that, the user gets access to turn on the heater. The temperature sensor keeps monitoring the temperature and sends the analog value to the microcontroller. The microcontroller processes the analog input of the temperature sensor and outputs a temperature value through serial communication. All the activities of the control system are done on a microcontroller called Atmega328.

- In the places where snow falls or temperature is less, that is around zero degree centigrade, people feel comfort if the car seats are warm while travelling.
- In this project Microcontroller ATMega 328 is used to control seat temperature.
- Program is designed like Heater starts operation only if person is sitting on seat and heater switch is closed.
- When both above conditions are true Temperature sensor starts sensing temperature of seat.
- If seat temperature is more than 25 deg centigrade, there will be no supply to heater because seat is warm enough.
- If temperature is less than 25 deg PWM output drives the heater driver circuit.
- PWM output and temperature values are inversely proportional i.e., lesser the temperature more value of PWM signal will drive the heater.
- For different range of temperature values different PWM value signal is generated.
- When temperature rises above 25 deg, PWM signal will be 0(low).
- All this operation is automatic and also temperature range is transmitted to serial output.

Research:

- OBJECTIVES: The aim of the project is to design a CAR SEAT HEAT CONTROL SYSTEM, Climate control is a more sophisticated form of air-conditioning, which allows the temperature of a cars Seats to be accurately controlled. Users can set the required temperature and the system automatically adjusts the speed and amount of cold air introduced into the Seats & cabin

SWOT ANALYSIS

STRENGTHS

- Seats with individual heating systems.
- The temperature range can easily be changed.
- Easy to use.

WEAKNESS

- Usually recommended for countries with extreme cold temperatures
- During this process, the engine needs to be on.

Opportunities:

- In future, more Data such as temperature and seat comfortability can be added apart from the ones which are used here in this project.

Threats:

- Company now send and retrieve data from databases to help better manage inventory and the like, but some companies don't have the right data encryption practices in place. This can often lead to lost data and sometimes, the data easily obtainable by criminals.

Requirements

4W's and 1 H's

Where:

- This problem is surfaced in all the user and manufacture in the world.

Who:

- All the Car Users & Manufactures who has lots of System requirement in their different units and their different services

When:

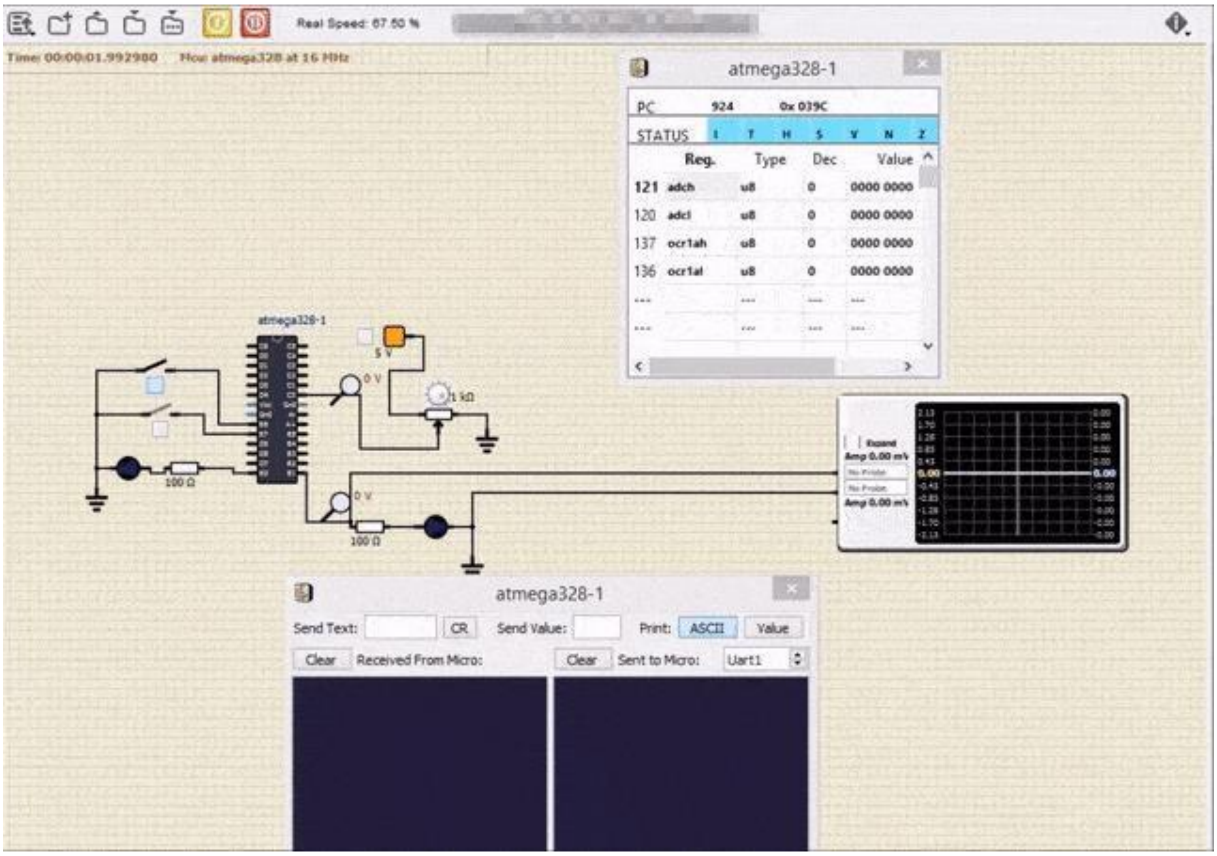
- As the business of the organisation increases number of System users
- With the increases of user end

How:

- This project takes in all the input values and yields out the management parameters.

Design

ON



Real Speed: 67.50 %

Time: 00:00:01.992980 Mcu: atmega328 at 16 MHz

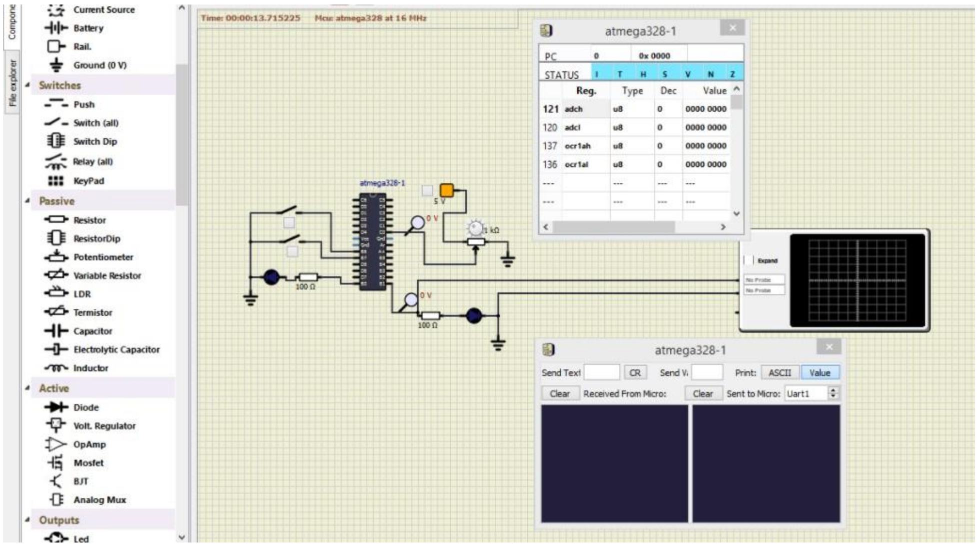
atmega328-1

PC	924	0x 039C

STATUS I T H S V N Z

	Reg.	Type	Dec	Value
121	adch	u8	0	0000 0000
120	adcl	u8	0	0000 0000
137	ocr1ah	u8	0	0000 0000
136	ocr1al	u8	0	0000 0000
---		---	---	---
---		---	---	---

OFF



Time: 00:00:13.715225 Mcu: atmega328 at 16 MHz

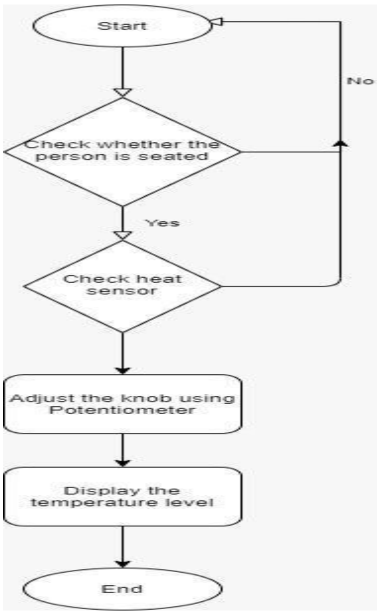
atmega328-1

PC	0	0x 0000

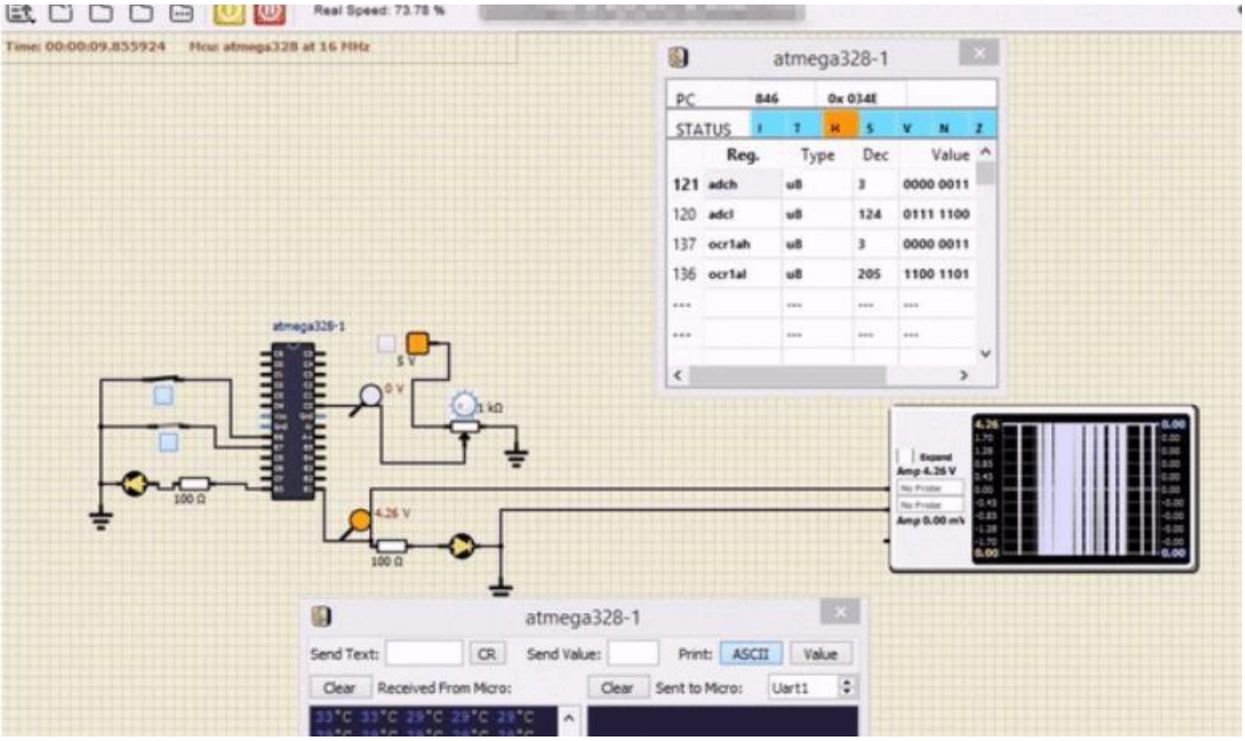
STATUS I T H S V N Z

	Reg.	Type	Dec	Value
121	adch	u8	0	0000 0000
120	adcl	u8	0	0000 0000
137	ocr1ah	u8	0	0000 0000
136	ocr1al	u8	0	0000 0000
---		---	---	---
---		---	---	---

Design: Structure Diagram


Start
No
Check whether the person is seated
Yes
Check heat sensor
Adjust the knob using Potentiometer
Display the temperature level
End

Behaviour Diagram



atmega328-1

PC	846	0x 034E	

STATUS: I T H S V N Z

	Reg.	Type	Dec	Value
121	adch	u8	3	0000 0011
120	adcl	u8	124	0111 1100
137	ocr1ah	u8	3	0000 0011
136	ocr1al	u8	205	1100 1101
---		---	---	---
---		---	---	---

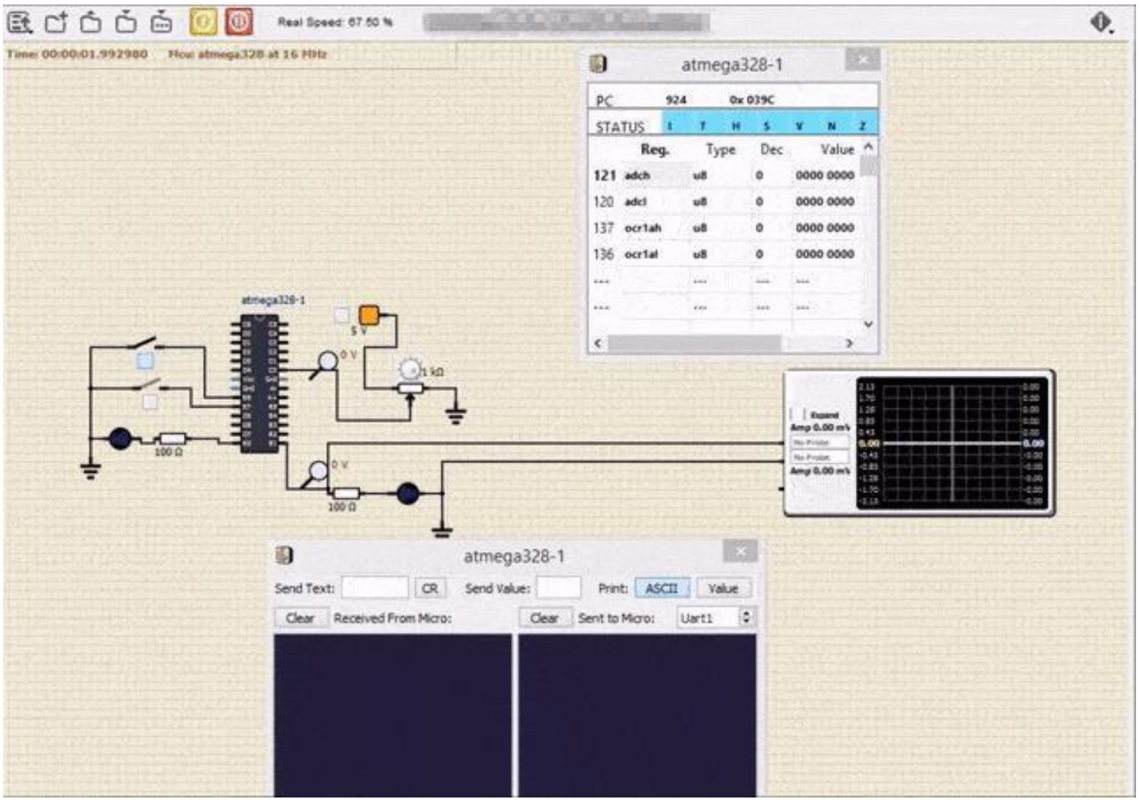
```
33°C 33°C 29°C 29°C 29°C
```

Images and Videos:

Simulation

The functionality of the heat control system is coded in embedded c and the working is demonstrated using simulation in a software called SimulIDE. Below shows two images where in the 1st image shows the status of the simulation when the system is ON and the second image shows the status of the system when it is OFF

ON



Real Speed: 67.50 %

Time: 00:00:01.992980 Mcu: atmega328 at 16 MHz

atmega328-1

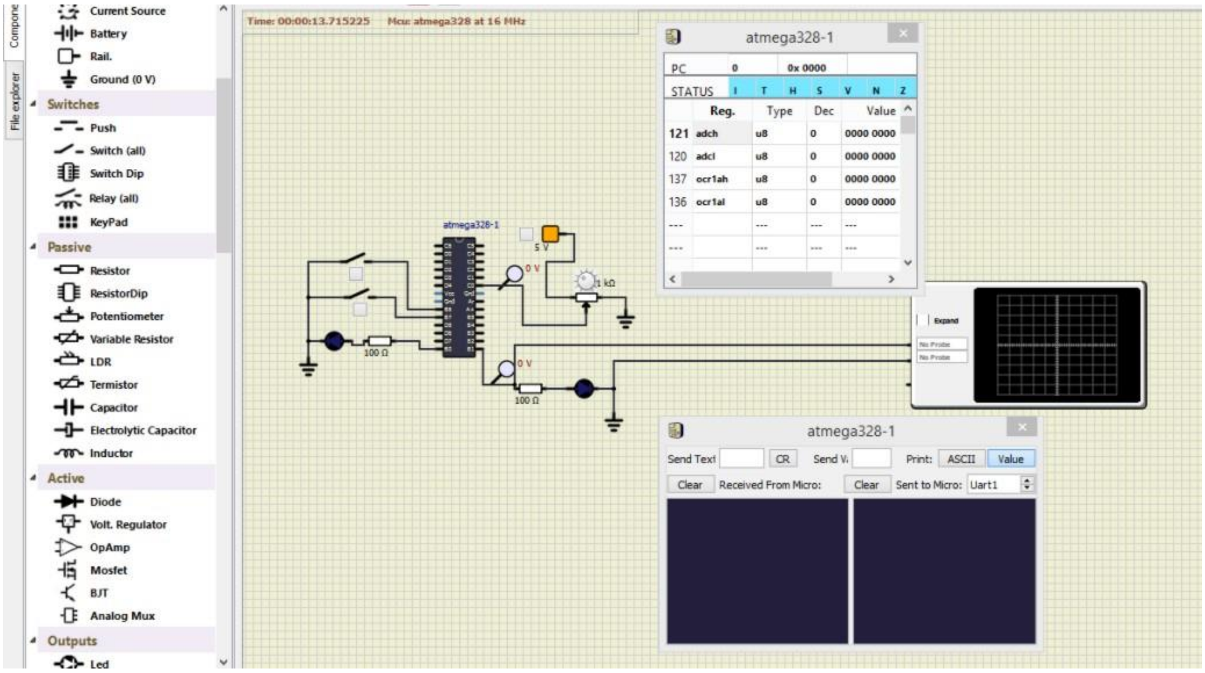
PC	924	0x 039C		
STATUS	I T H S V N Z			
	Reg.	Type	Dec	Value
121	adch	u8	0	0000 0000
120	adcl	u8	0	0000 0000
137	ocr1ah	u8	0	0000 0000
136	ocr1al	u8	0	0000 0000
---		---	---	---
---		---	---	---

atmega328-1

Send Text: CR Send Value: Print: ASCII Value

Clear Received From Micro: Clear Sent to Micro: Uart1

OFF



Time: 00:00:13.715225 Mcu: atmega328 at 16 MHz

atmega328-1

PC	0	0x 0000		
STATUS	I T H S V N Z			
	Reg.	Type	Dec	Value
121	adch	u8	0	0000 0000
120	adcl	u8	0	0000 0000
137	ocr1ah	u8	0	0000 0000
136	ocr1al	u8	0	0000 0000
---		---	---	---
---		---	---	---

atmega328-1

Send Text CR Send V: Print: ASCII Value

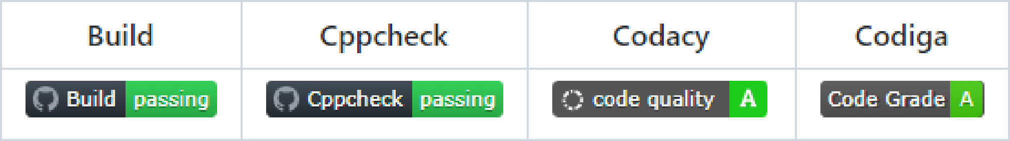
Clear Received From Micro: Clear Sent to Micro: Uart1

Implementation and Summary

Git Link:

Link: https://github.com/kotalavaishnavi/M2-Embedded_SEAT-HEAT-MONITOR-AND-CONTROL-SYSTEM

Git Dashboard



Build	Cppcheck	Codacy	Codiga
Build passing	Cppcheck passing	code quality A	Code Grade A

Mini Project 3 – Scientific Calculator [Team]

Modules

1. SDLC
2. Git

Requirements:

Introduction:

It is an advanced calculator that will allow users to perform operations in mathematics (Arithmetic, Trigonometric, Matrices, Conversion etc). However, the input has to be provided by user. The input values can be from any integer to even a number with decimals. Moreover, this calculator is smart enough to operate all the operations.

Modules:

1. SDLC
2. Git

Features:

- It can do all operations (Addition, Subtraction, Multiplication, Division).
- Logarithmic operations, Exponential operations are also available.
- Power functions, Factorial and Conversions which are helpful for students are added.
- Basic Trigonometric Operations are also available.
- Basic Matrix Operations are also available.
- It has double precision.

Cost v/s Features:

COST

Casio FX (Programmable)	Casio FX (Non-color Programmable)	Casio FX (color Programmable)
Casio FX (240 functions)	Casio FX (300 functions)	Casio FX (522 functions)
10 Digit Casio	12 digit Casio	12 digit robust Casio

FEATURES

SWOT Analysis:

Strengths

- User Friendly
- All basic operations
- Double Precision
- Trigonometric operations
- Matrix Operations

Weakness

- Limited Operations
- Memory Wastage

Opportunities

- It can be expanded by adding additional features like Inverse Trigonometric operations, Equations etc.

Threats

- There are other programmable calculators which may affect our product marketing.

4W's and 1H's:

What: Scientific Calculator.

Where: Used in Provisional Stores and Students.

When: For Complex Calculations.

Why: For Easy and faster Calculations.

How: Operates by using User Inputs.

Detail Requirements:

High level Requirements:

High level Requirement	Description
HLR1	Basic Operations
HLR2	Trigonometric Operations
HLR3	Matrix Operations
HLR4	Conversions
HRL5	Advanced operations

Low level Requirements:

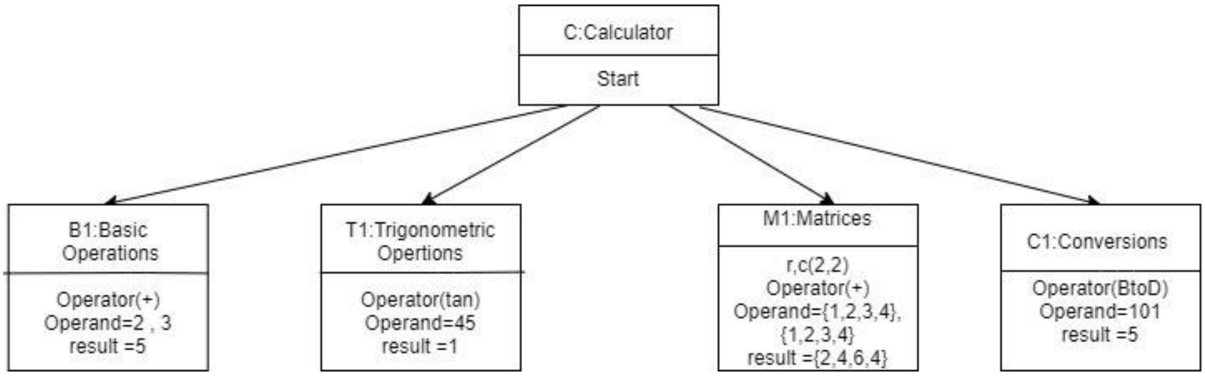
Low Level Requirement	Description	Related HLR
LLR1	Addition	HLR1
LLR2	Subtraction	HLR1
LLR3	Multiplication	HLR1
LLR4	Division	HLR1
LLR5	Sine	HLR2
LLR6	Cos	HLR2
LLR7	Tan	HLR2
LLR8	Cosec	HLR2
LLR9	sec	HLR2
LLR10	Cot	HLR2
LLR11	matrix addition	HLR3

Low Level Requirement	Description	Related HLR
LLR12	matrix subtraction	HLR3
LLR13	matrix multiplication	HLR3
LLR14	Binary - Decimal Conversion	HLR4
LLR15	Decimal - Binary Conversion	HLR4
LLR16	Decimal - Octal Conversion	HLR4
LLR17	Octal - Decimal Conversion	HLR4
LLR18	Length unit Conversion	HLR4
LLR19	Temperature unit Conversion	HLR4
LLR20	Current AC-DC Conversion	HLR4
LLR21	Log	HLR5
LLR22	Exponential	HLR5
LLR23	Modulus (remainder)	HLR5
LLR24	Factorial	HLR5
LLR25	Square root	HLR5
LLR26	Cube root	HLR5
LLR27	LCM	HLR5
LLR28	GCD	HLR5
LLR29	Permutation	HLR5
LLR30	Combination	HLR5

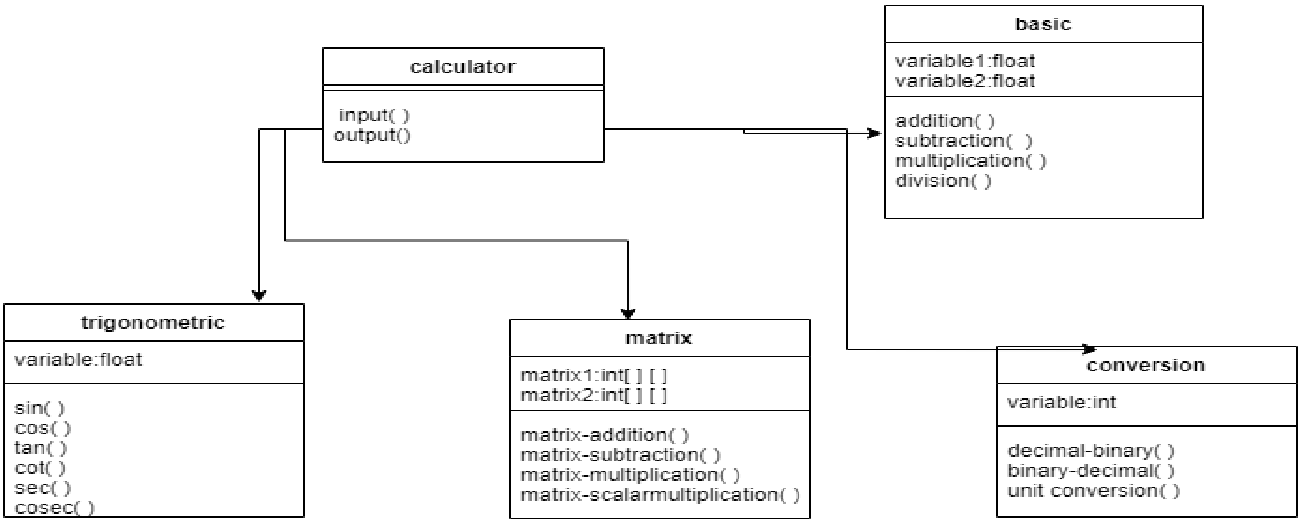
Design:

High Level Design

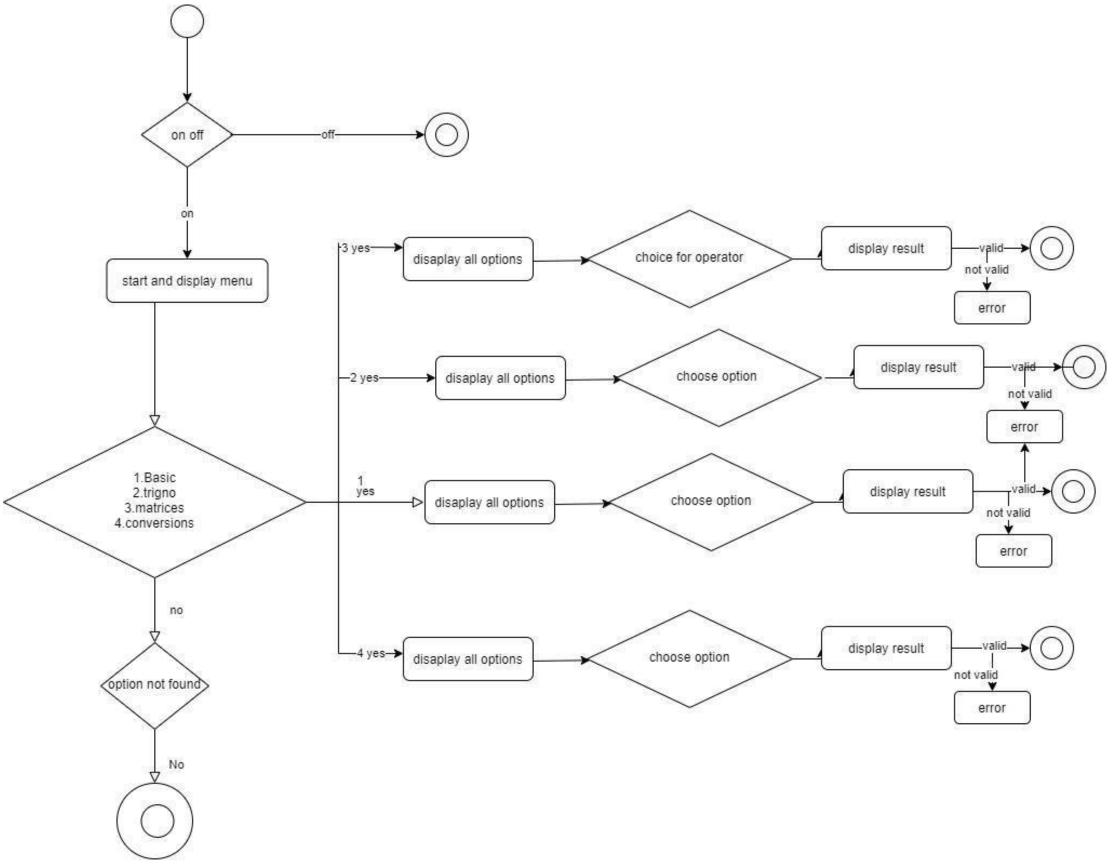
OBJECT DIAGRAM FOR HIGH LEVEL REQUIREMENTS


C:Calculator
Start
B1:Basic Operations
Operator(+)
Operand=2 , 3
result =5
T1:Trigonometric Opertions
Operator(tan)
Operand=45
result =1
M1:Matrices
r,c(2,2)
Operator(+)
Operand={1,2,3,4},
{1,2,3,4}
result ={2,4,6,4}
C1:Conversions
Operator(BtoD)
Operand=101
result =5

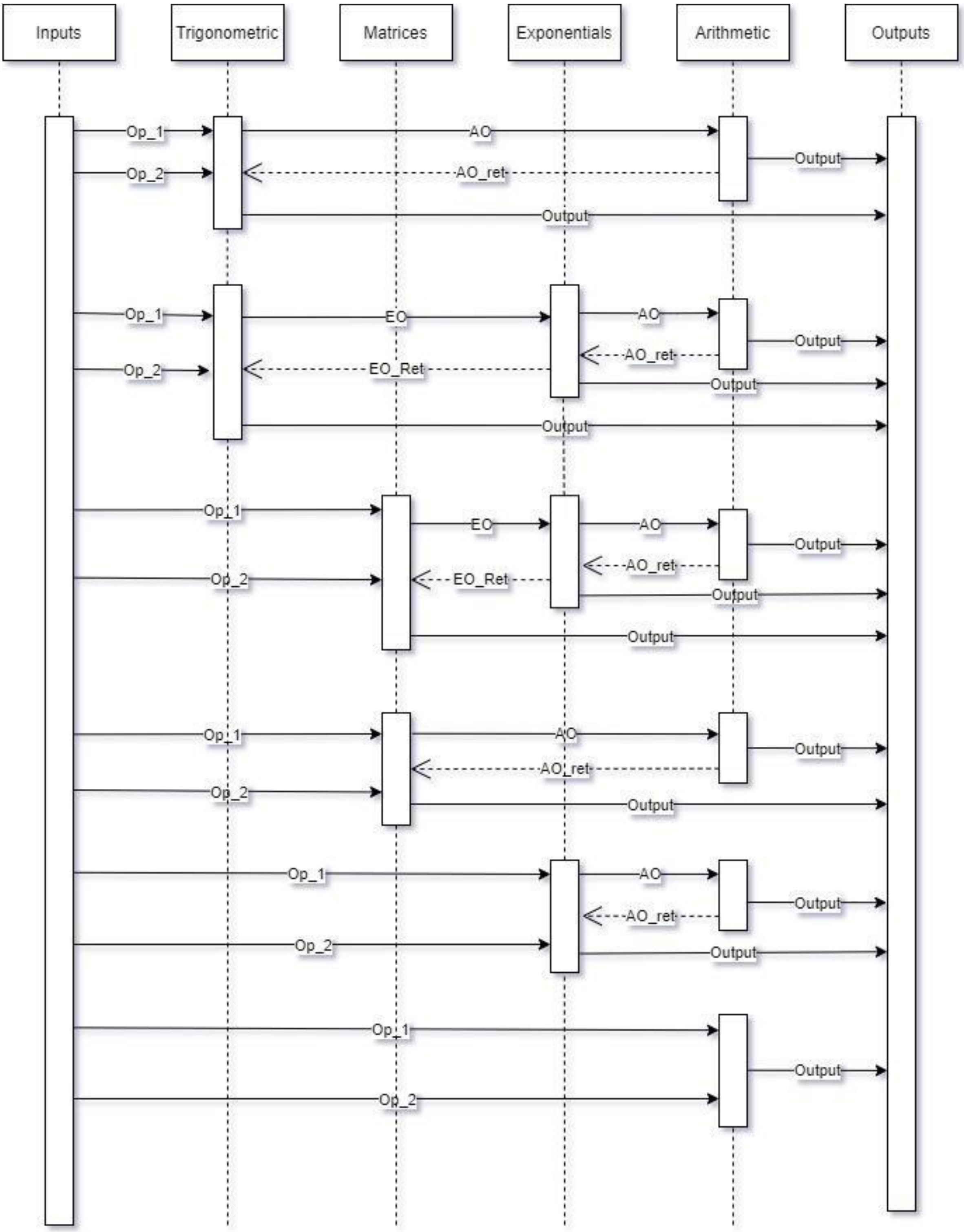
CLASS DIAGRAM FOR HLR



ACTIVITY DIAGRAM for HLR

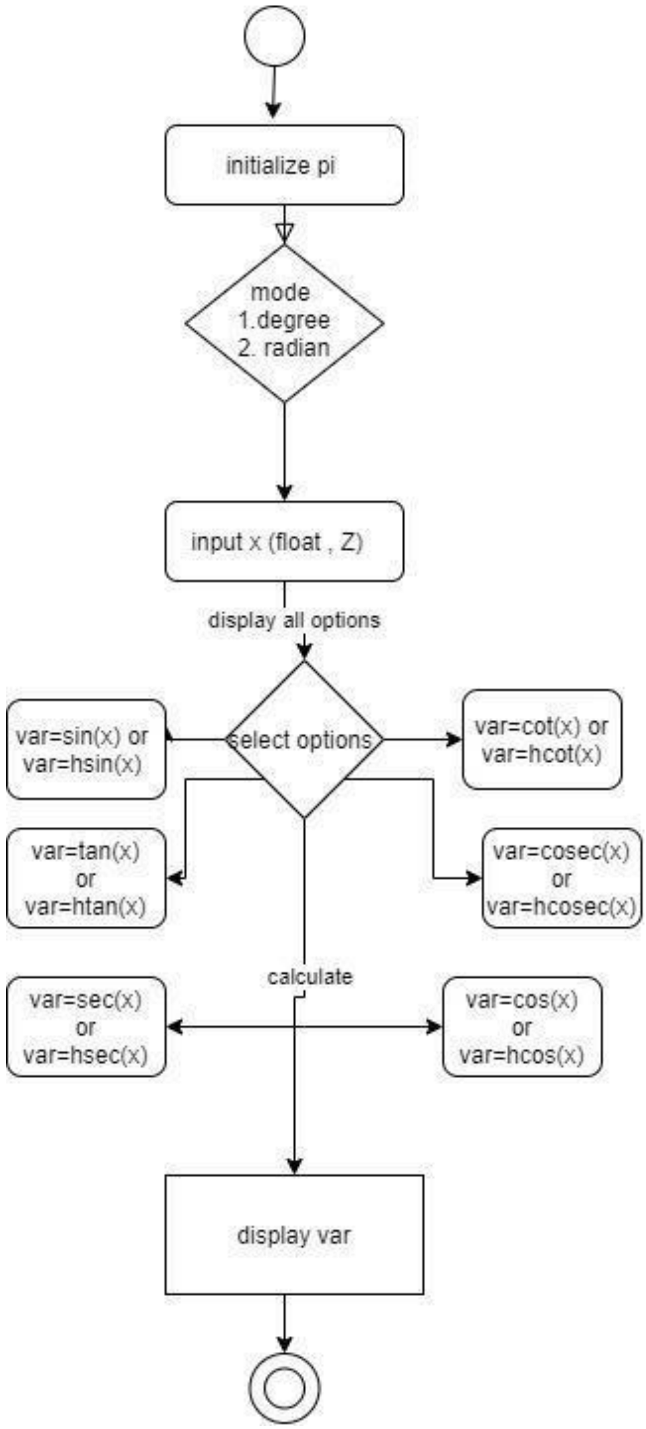


SEQUENCE DIAGRAM FOR HIGH LEVEL REQUIREMENTS

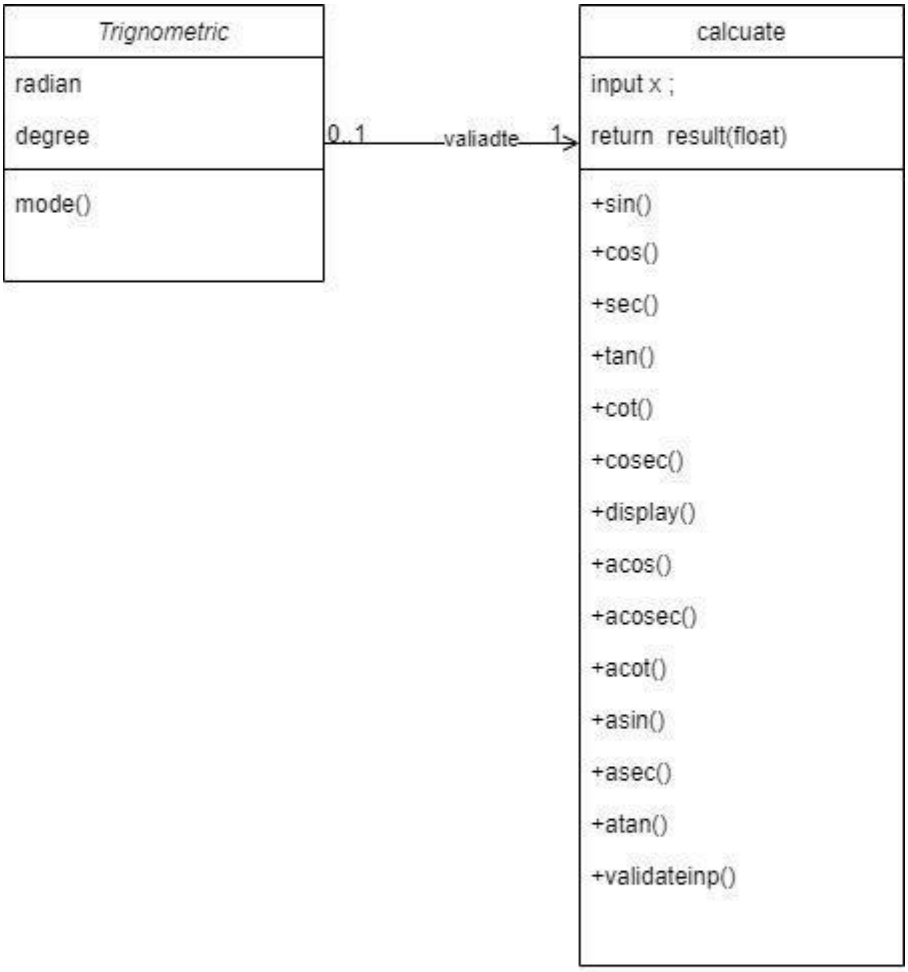


Low Level Design

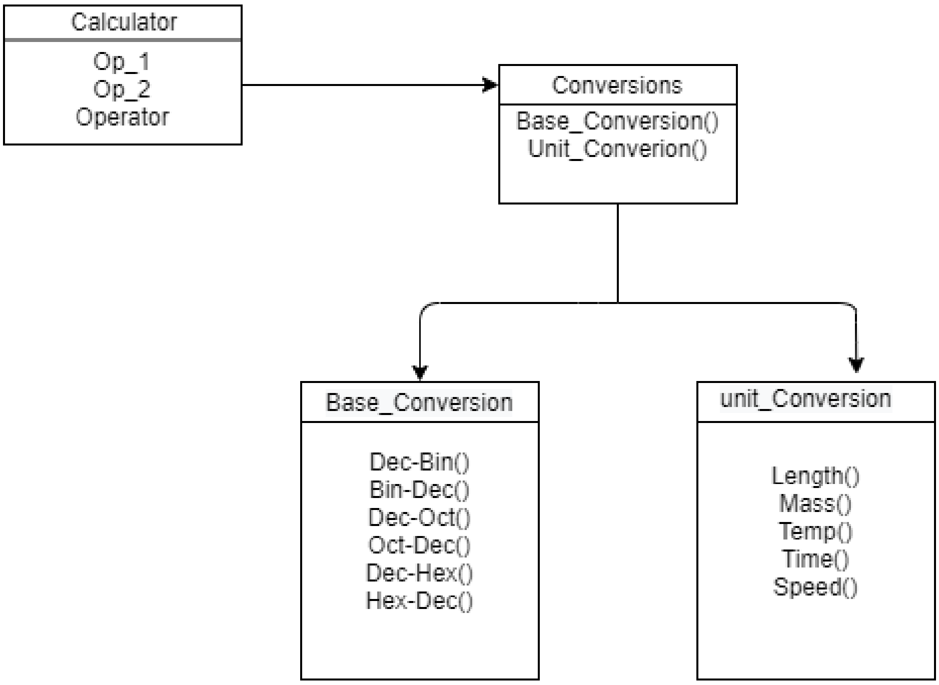
ACTIVITY DIAGRAM for TRIGNOMETRIC

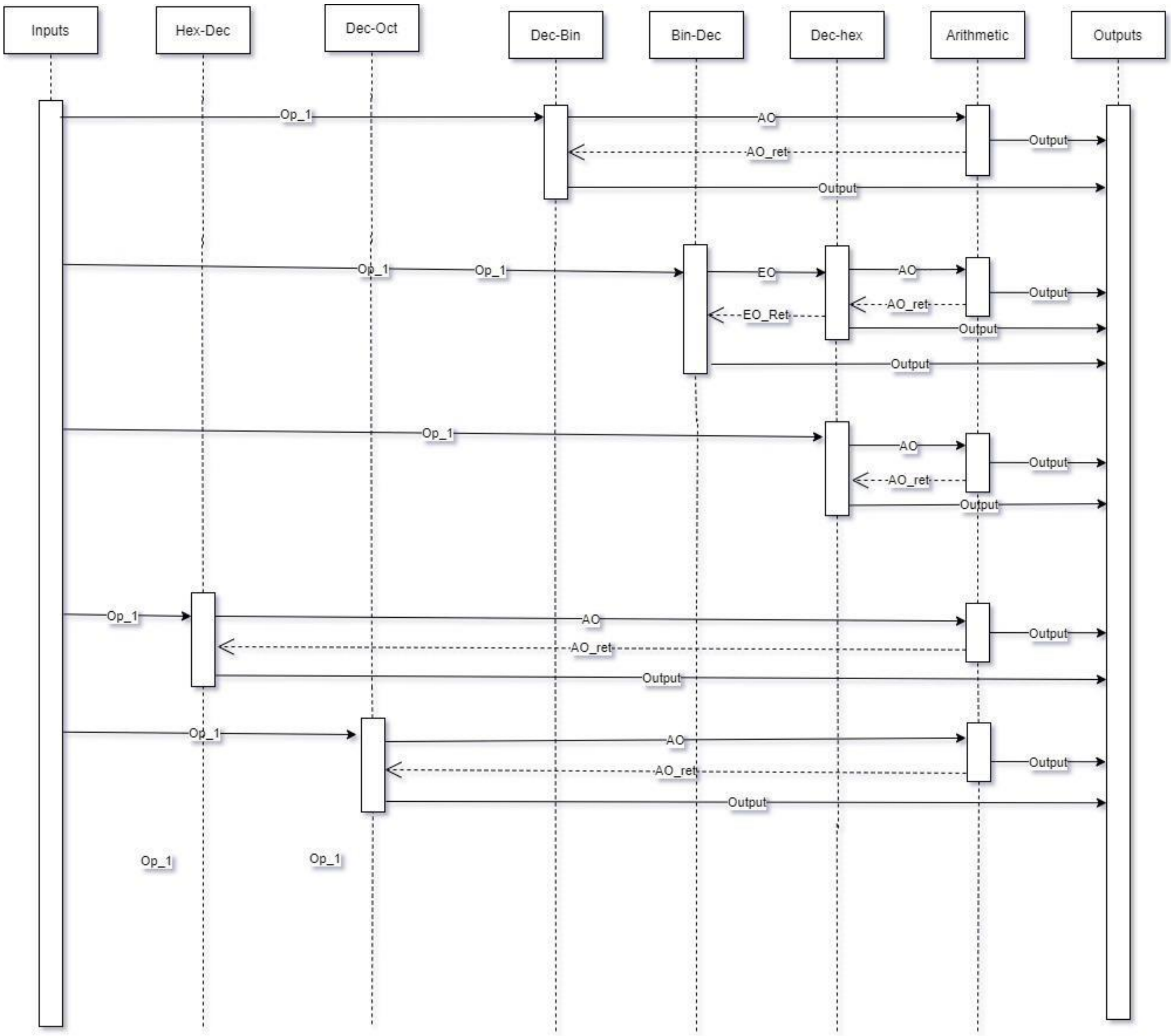


CLASS DIAGRAM FOR TRIGNOMETRIC



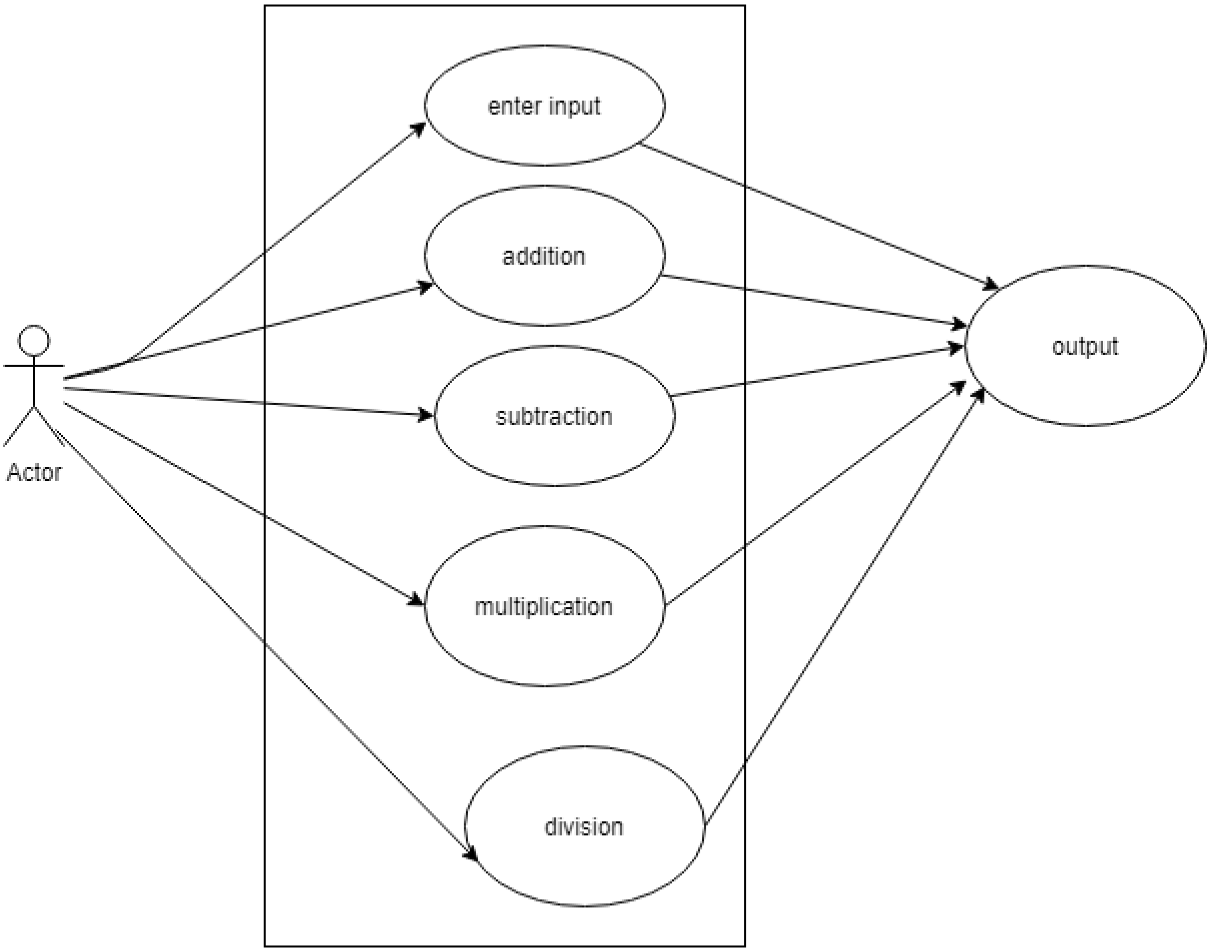
CLASS DIAGRAM FOR CONVERSION



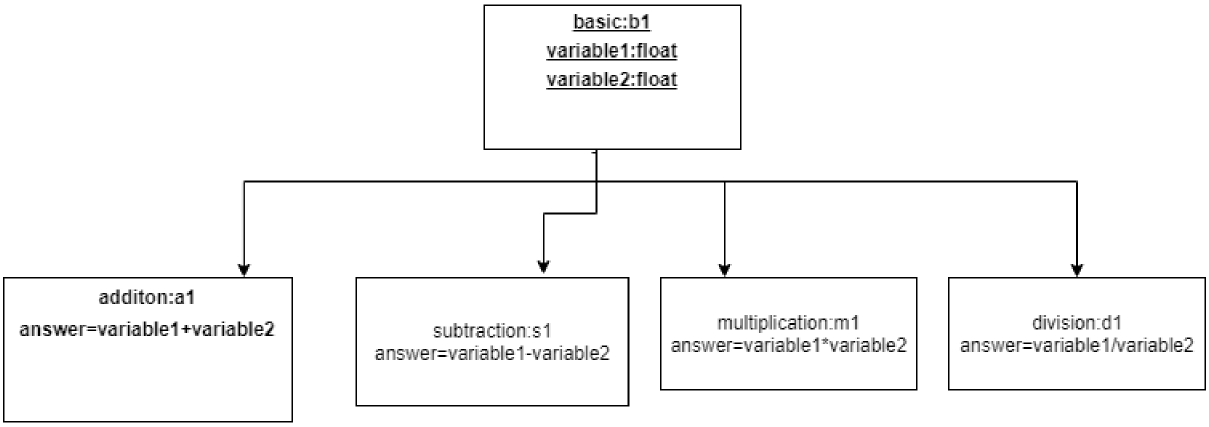


SEQUENCE DIAGRAM FOR CONVERSI

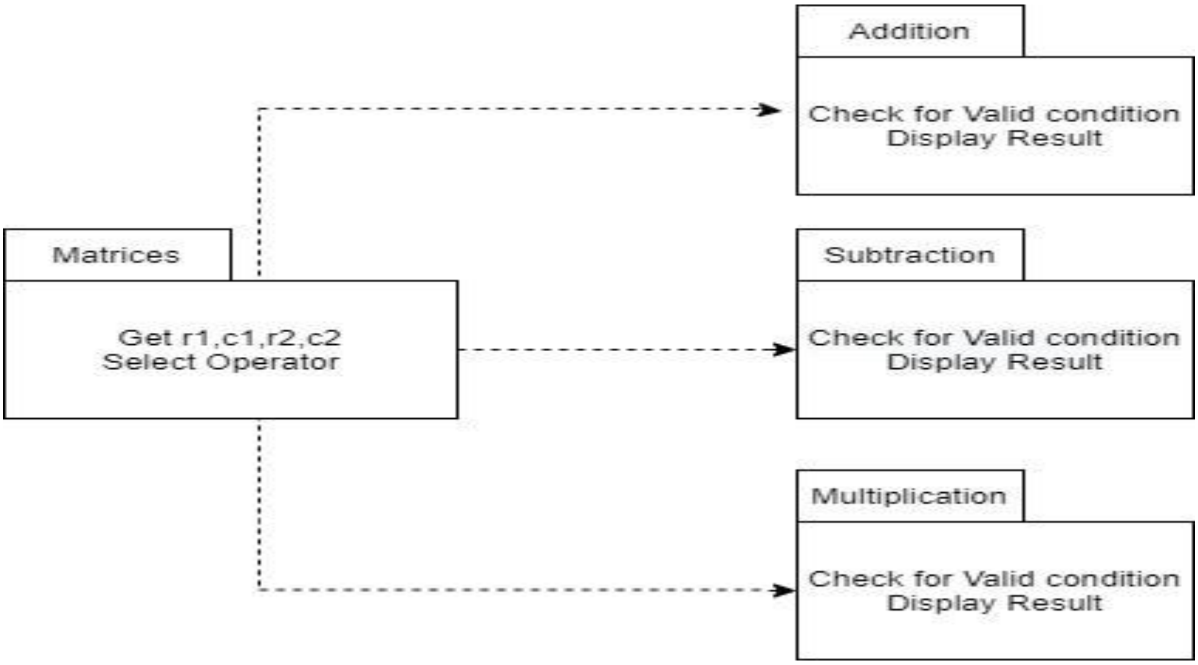
USE CASE DIAGRAM FOR BASIC



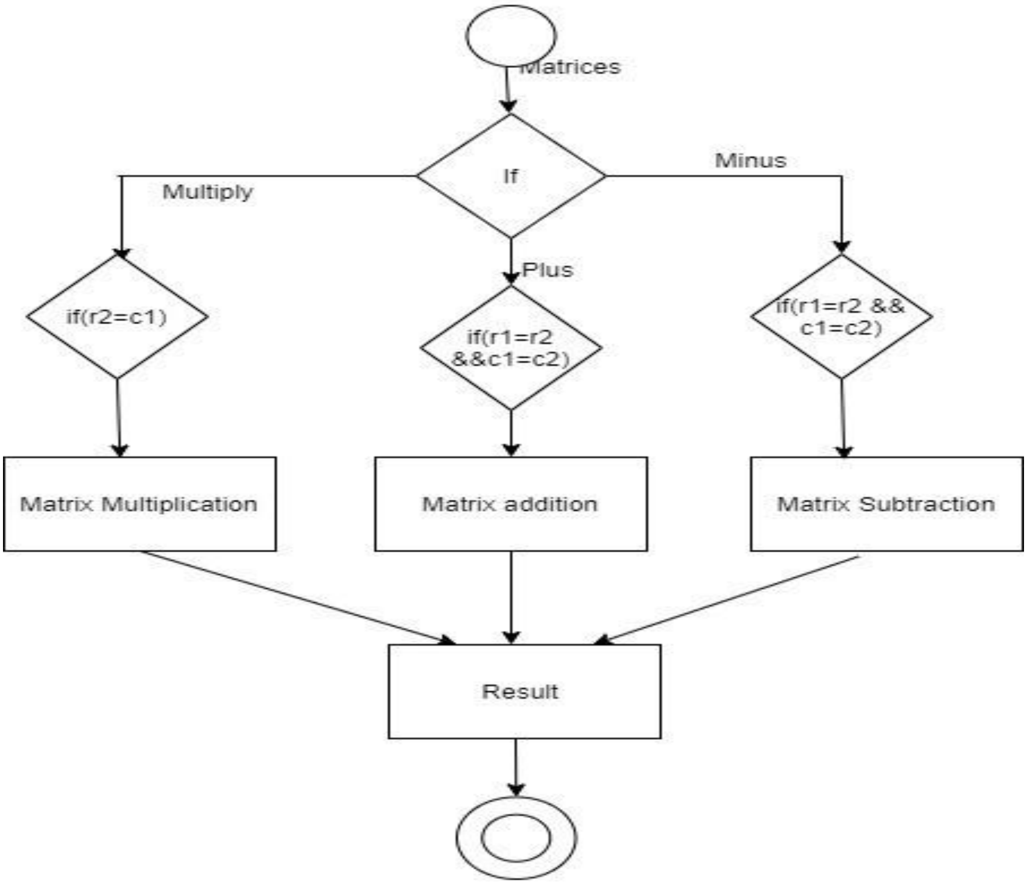
OBJECT DIAGRAM FOR BASIC


basic:b1
variable1:float
variable2:float
additon:a1
answer=variable1+variable2
subtraction:s1
answer=variable1-variable2
multiplication:m1
answer=variable1*variable2
division:d1
answer=variable1/variable2

PACKAGE DIAGRAM FOR MATRIX OPERATIONS



ACTIVITY DIAGRAM FOR MATRIX OPERATION



Test cases:

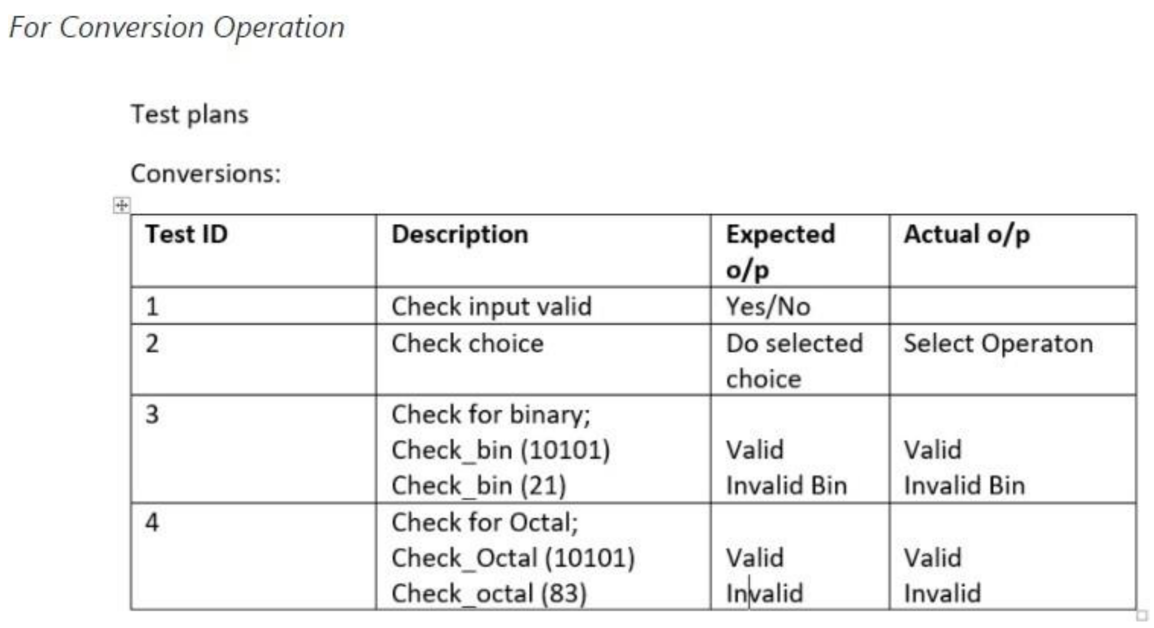
Test Plan for Matrix Operation:

Test ID	Requirement	Expected Output	Actual output
LLR8	R1! =R2 &C1! =C2	Invalid	Invalid
LLR8	R1=R2 &C1=C2	Do Matrix Addition	Do Matrix Addition
LLR9	R1! =R2 &C1! =C2	Invalid	Invalid
LLR9	R1=R2 &C1=C2	Do Matrix Subtraction	Do Matrix Addition
LLR10	R2! =C1	Invalid	Invalid
LLR10	R2=C1	Do Matrix Multiplication	Do Matrix Addition

Test Plan for Trigonometric Operation:

Test ID	Requirement	Expected Output	Type of test case	Actual output
LLR1	sin (30)	-0.988032	Positive Test case	-0.988032
LLR2	cos (30)	0.1542	Positive Test case	0.1542
LLR3	tan (30)	-6.405331	Positive Test case	-6.405331
LLR4	cot (30)	-0.15612	Positive Test case	-0.15612
LLR5	sec (30)	6.4829	Positive Test case	6.4829
LLR6	cosec (30)	-1.012113	Positive Test case	-1.012113
LLR7	sin (60)	-0.304811	Positive Test case	-0.304811
LLR8	cos (60)	-0.9524	Positive Test case	-0.9524

Test ID	Requirement	Expected Output	Type of test case	Actual output
LLR9	tan (60)	0.32004	Positive Test case	0.32004
LLR10	cot (60)	3.124606	Positive Test case	3.124606
LLR11	sec (60)	-1.04996	Positive Test case	-1.04996
LLR12	cosec (60)	-3.280726	Positive Test case	-3.280726

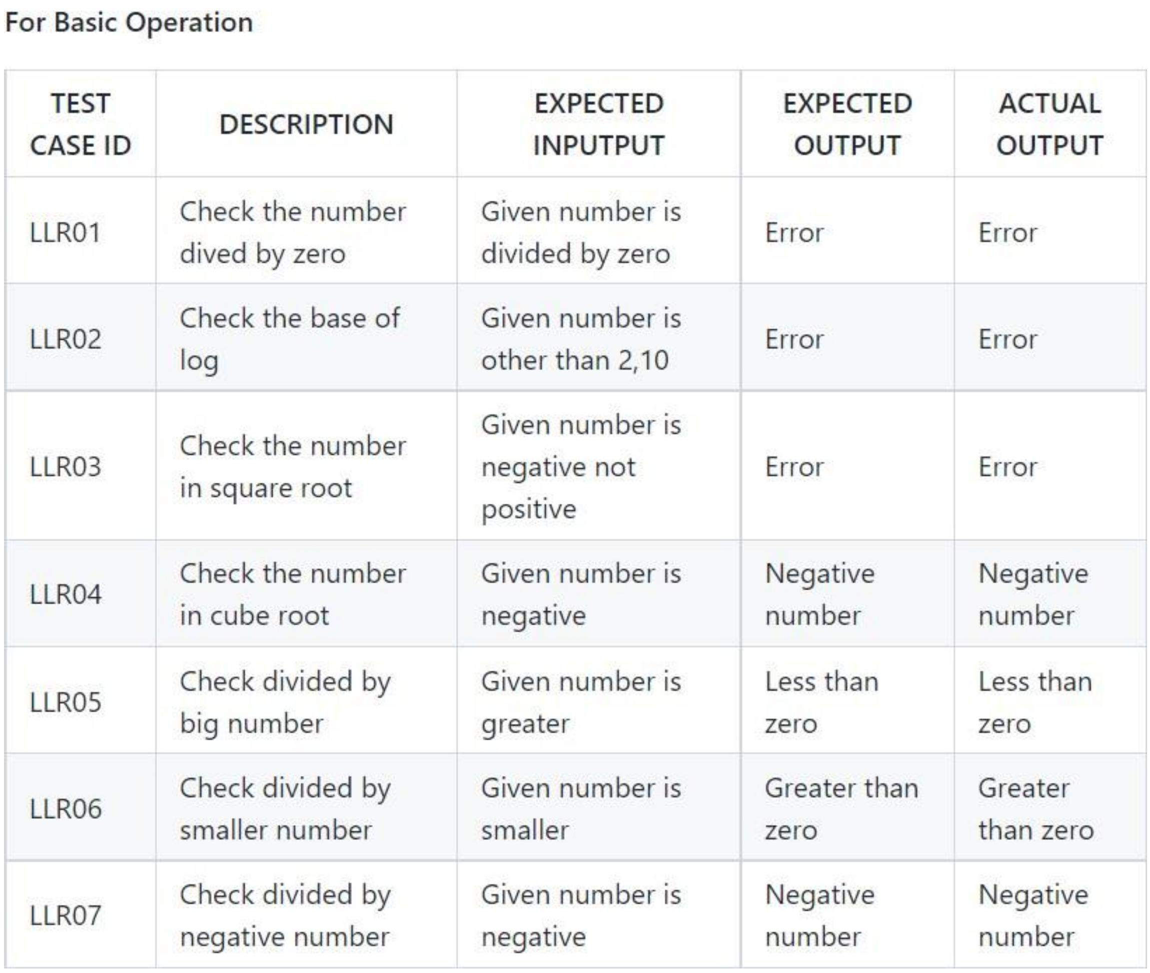


For Conversion Operation

Test plans

Conversions:

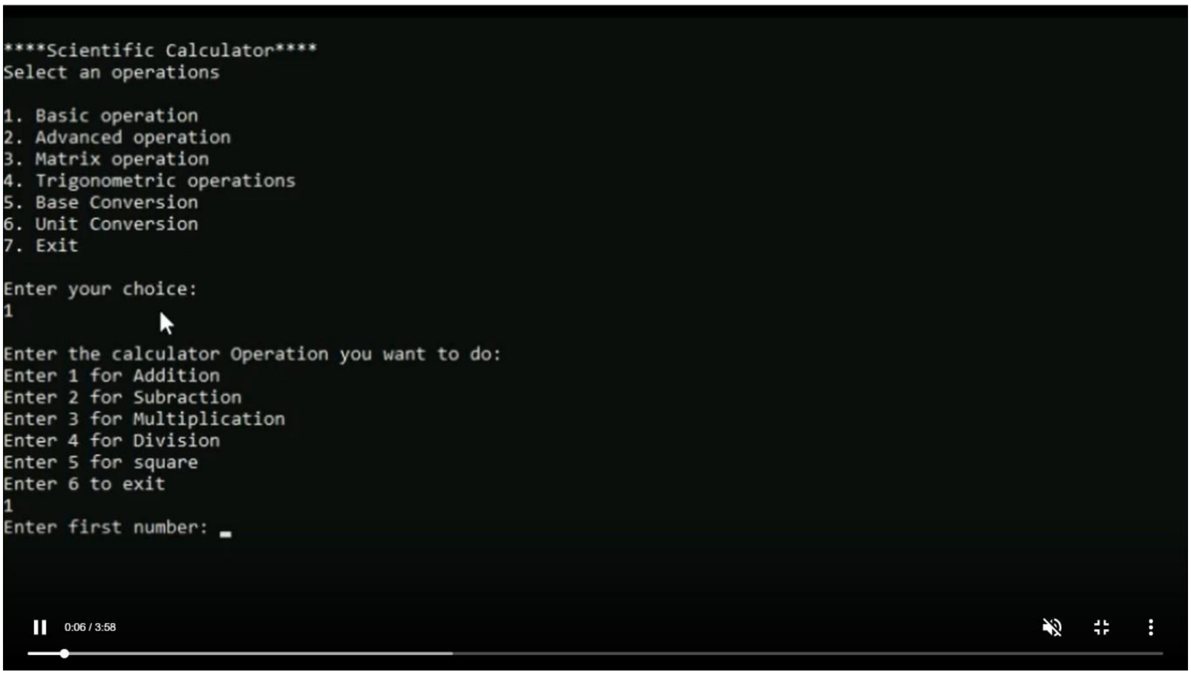
Test ID	Description	Expected o/p	Actual o/p
1	Check input valid	Yes/No	
2	Check choice	Do selected choice	Select Operaton
3	Check for binary; Check_bin (10101) Check_bin (21)	Valid Invalid Bin	Valid Invalid Bin
4	Check for Octal; Check_Octal (10101) Check_octal (83)	Valid Invalid	Valid Invalid



For Basic Operation

TEST CASE ID	DESCRIPTION	EXPECTED INPUTPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
LLR01	Check the number dived by zero	Given number is divided by zero	Error	Error
LLR02	Check the base of log	Given number is other than 2,10	Error	Error
LLR03	Check the number in square root	Given number is negative not positive	Error	Error
LLR04	Check the number in cube root	Given number is negative	Negative number	Negative number
LLR05	Check divided by big number	Given number is greater	Less than zero	Less than zero
LLR06	Check divided by smaller number	Given number is smaller	Greater than zero	Greater than zero
LLR07	Check divided by negative number	Given number is negative	Negative number	Negative number

6_Images_And_Videos



```
****Scientific Calculator****
Select an operations

1. Basic operation
2. Advanced operation
3. Matrix operation
4. Trigonometric operations
5. Base Conversion
6. Unit Conversion
7. Exit

Enter your choice:
1

Enter the calculator Operation you want to do:
Enter 1 for Addition
Enter 2 for Subraction
Enter 3 for Multiplication
Enter 4 for Division
Enter 5 for square
Enter 6 to exit
1
Enter first number: _
```

Implementation and Summary:

Github Link: **https://github.com/GENESIS2021Q1/Applied_SDLC-Dec_Team_48.git**

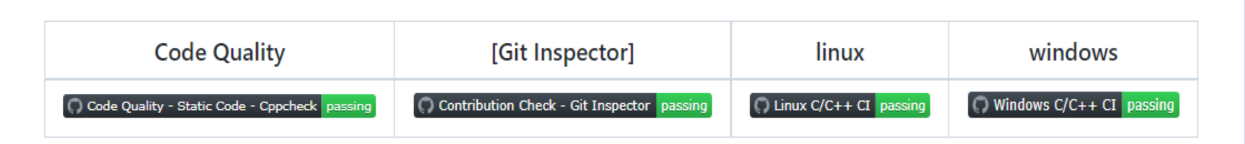
Challenges Faced and How Was It Overcome

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

Learning Resources

1. markdownCheatsheet
2. markdownBasics
3. git inspector
4. github workflow

Git Dashboard:



Code Quality	[Git Inspector]	linux	windows
Code Quality - Static Code - Cppcheck passing	Contribution Check - Git Inspector passing	Linux C/C++ CI passing	Windows C/C++ CI passing

Individual Contribution and Highlights

- Requirements
- Implementation (Add Order in System)
- Created Unity File
- Test cases.

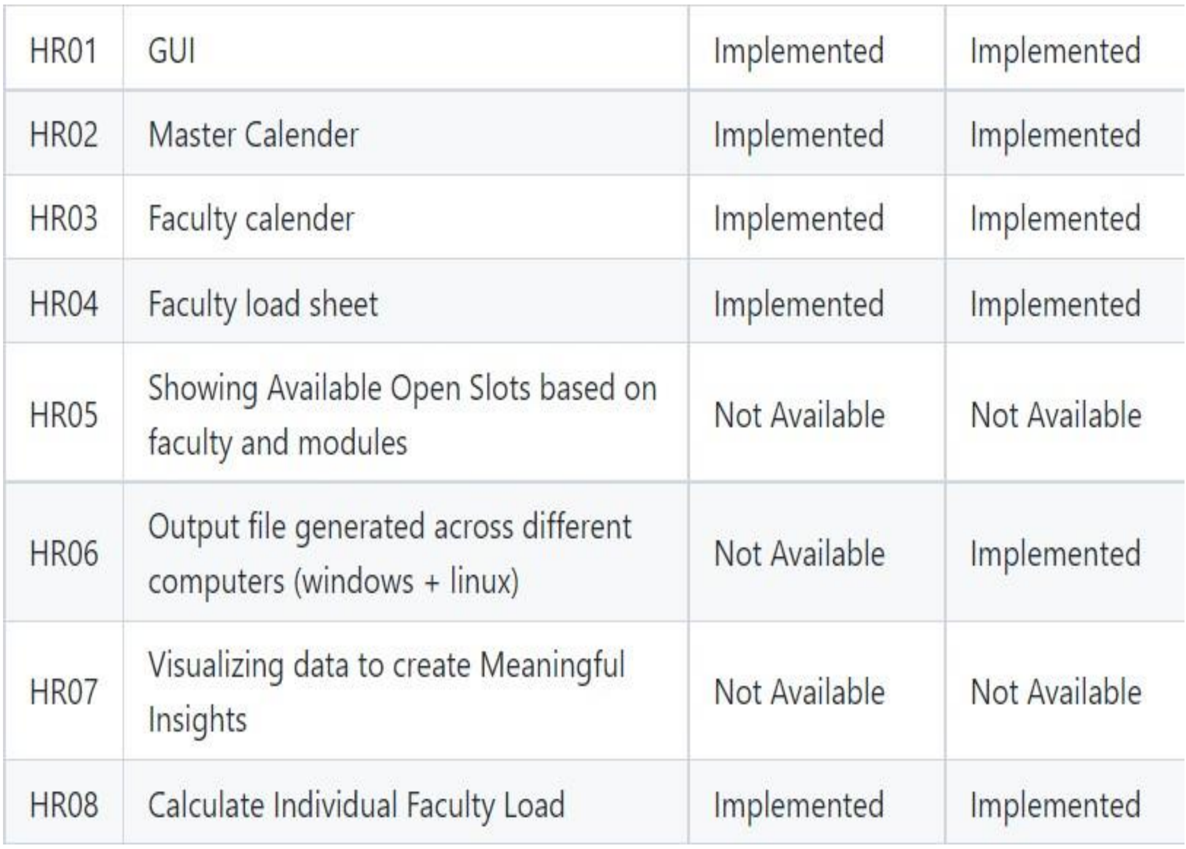
Mini Project 4 – Calendar Automation [Team]

Modules

1. Python
2. Git

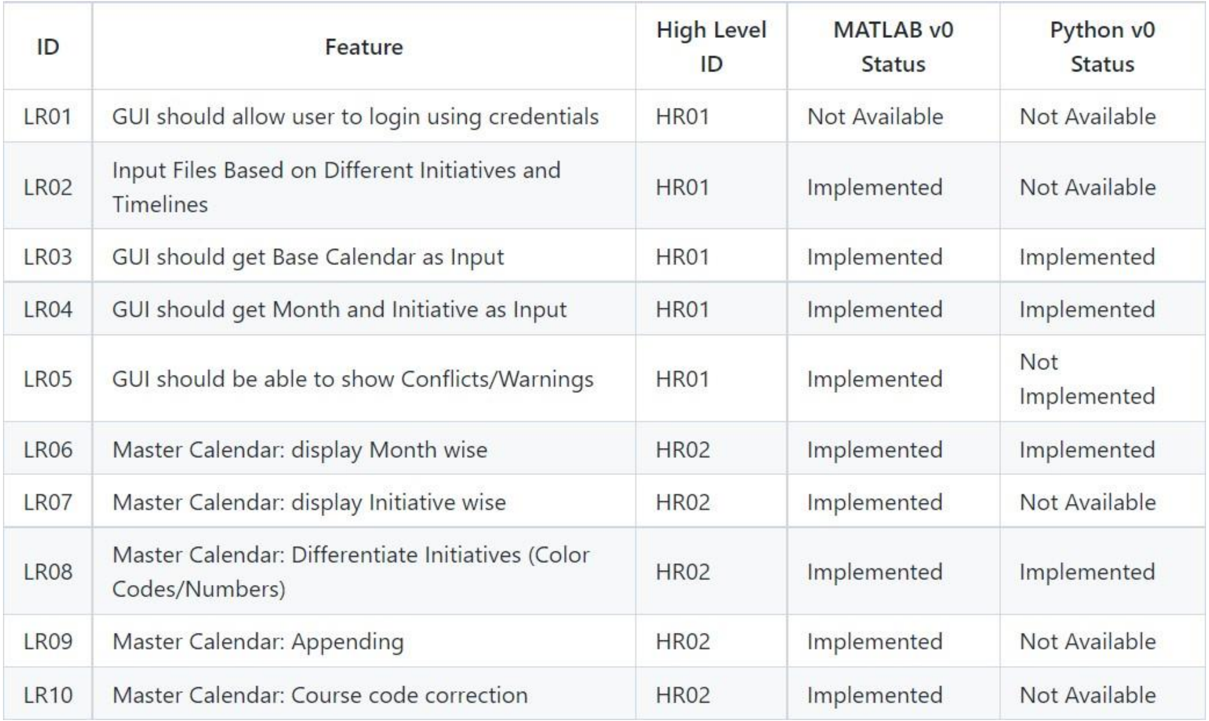
Requirements

High Level Requirements



HR01	GUI	Implemented	Implemented
HR02	Master Calender	Implemented	Implemented
HR03	Faculty calender	Implemented	Implemented
HR04	Faculty load sheet	Implemented	Implemented
HR05	Showing Available Open Slots based on faculty and modules	Not Available	Not Available
HR06	Output file generated across different computers (windows + linux)	Not Available	Implemented
HR07	Visualizing data to create Meaningful Insights	Not Available	Not Available
HR08	Calculate Individual Faculty Load	Implemented	Implemented

Low Level Requirements



ID	Feature	High Level ID	MATLAB v0 Status	Python v0 Status
LR01	GUI should allow user to login using credentials	HR01	Not Available	Not Available
LR02	Input Files Based on Different Initiatives and Timelines	HR01	Implemented	Not Available
LR03	GUI should get Base Calendar as Input	HR01	Implemented	Implemented
LR04	GUI should get Month and Initiative as Input	HR01	Implemented	Implemented
LR05	GUI should be able to show Conflicts/Warnings	HR01	Implemented	Not Implemented
LR06	Master Calendar: display Month wise	HR02	Implemented	Implemented
LR07	Master Calendar: display Initiative wise	HR02	Implemented	Not Available
LR08	Master Calendar: Differentiate Initiatives (Color Codes/Numbers)	HR02	Implemented	Implemented
LR09	Master Calendar: Appending	HR02	Implemented	Not Available
LR10	Master Calendar: Course code correction	HR02	Implemented	Not Available

Link for standard input template:

https://docs.google.com/spreadsheets/d/1EWYp_1iyK2wLMfKGJOiTJAk5WexZusCP/edit?usp=sharing&ouid=113003694561146884677&rtpof=true&sd=true

- Using the template above, training schedule can be added month wise and initiatives wise
- The name of the input excel sheet MUST be named as "Test vector"(as shown in template)
- Along with the Test vector sheet, "Key" sheet MUST be present under the columns assigned as in the template
- The "Key" sheet must contain all times the 6 fixed initiatives with their respective codes and total list of course code and course title in order to refer for corrections while writing to output files
- Appending additional slots for existing courses is possible by adding just the additional slots in the input file for the same course

Requirements for updating Master calendar using Master calendar as input:

Link for template

2 Slots format - M/A
: https://docs.google.com/spreadsheets/d/1jtKnXV12VE1fH20CGDo4B3uNWRTAhQCWz-hHUDWUe3I/edit?usp=sharing

4 Slots format - M1/M2/A1/A2
: https://docs.google.com/spreadsheets/d/1jVheSPZkOtfNKRNoc_858nwk2UaHCe0gExTNZfZ8vxA/edit?usp=sharing

- Any of the two templates can be used for updating Master calendar month wise on to the drive
- The blocked slots must have the corresponding initiative code in the cell according to the key as shown in the sample data in the template
- The name of the sheet must be the name of the month to be updated
- The "Key" sheet must be present with the fixed list of initiatives and initiative code

App deployment:

- The app is deployed on Heroku servers.
- To add/modify new features, you will be required to install HEROKU CLI link
- After installation, open terminal in working directory and enter the following commands:
    - "Heroku git:clone -a Gea calendar"
    - login using Heroku credentials
- After pulling and making changes, enter the following commands to push app and deploy on server
    - Git adds.
    - git commit -m "commit message"
    - git push Heroku master

Additional features for V1 to do:

- Update key sheet by appending new initiatives/courses list
- Check for duplicate course entries in input file
- Using built in libraries to identify number of days in month, current year and highlight weekend and holidays
- Function to remove a course schedule
- Read multiple months data in one sheet as input file (currently takes data one by one month)
- Calculate individual faculty load

Github link: https://github.com/Usharani8/Oopswithpython_Calendar_Automation_Team-47.git

Individual contribution and Highlights:

1. Improved implementation of Python Programming
2. Source code management using Git Hub.

Mini Project 5 – Ford Project [Team]

Modules

1. MATLAB
2. Git

Requirements

Introduction

A sunroof is a movable panel that opens to uncover a window in an automobile roof, allowing light and fresh air to enter the passenger compartment. Sunroofs can be manually operated or motor driven, and are available in many shapes, sizes and styles. While the term sunroof is now used generically to describe any glass panel in the roof, the term "moonroof" was historically used to describe stationary glass panes rigidly mounted in the roof panel over the passenger compartment.

A moonroof has a glass panel that is transparent and usually tinted. Previous terms include Sunshine Roof, Sliding Head and Sliding Roof.

Overview

• It gives the car a stylish interior and exterior, making it look trendy.

• They provide the user with an extra window that enables them to breathe in the more fresh air.

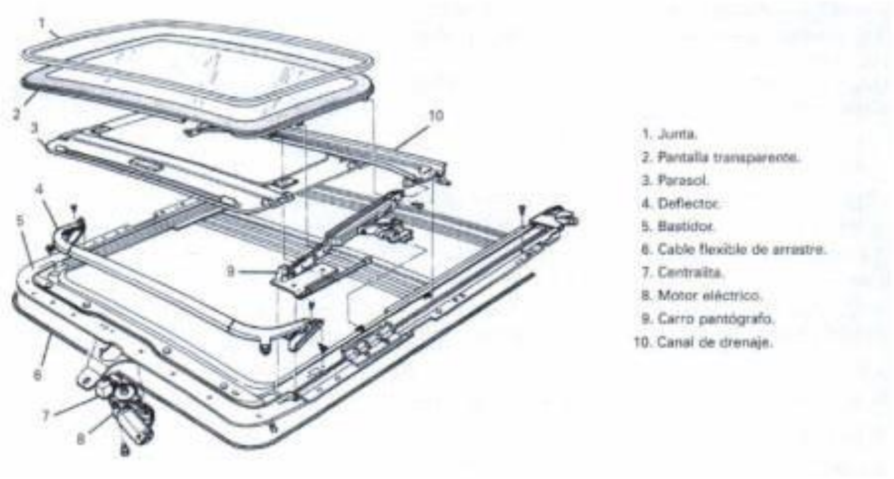
• In traffic, the car sunroofs come in handy as they fairly reduce the noises of vehicles and let the fresh air come in the passenger’s cabinet.

• It gives a beautiful view during the rains as the raindrops fall on the glass of the sunroofs. • In cold climates, these sunroofs enable the sun to throw in bright rays and make the driving experience better.

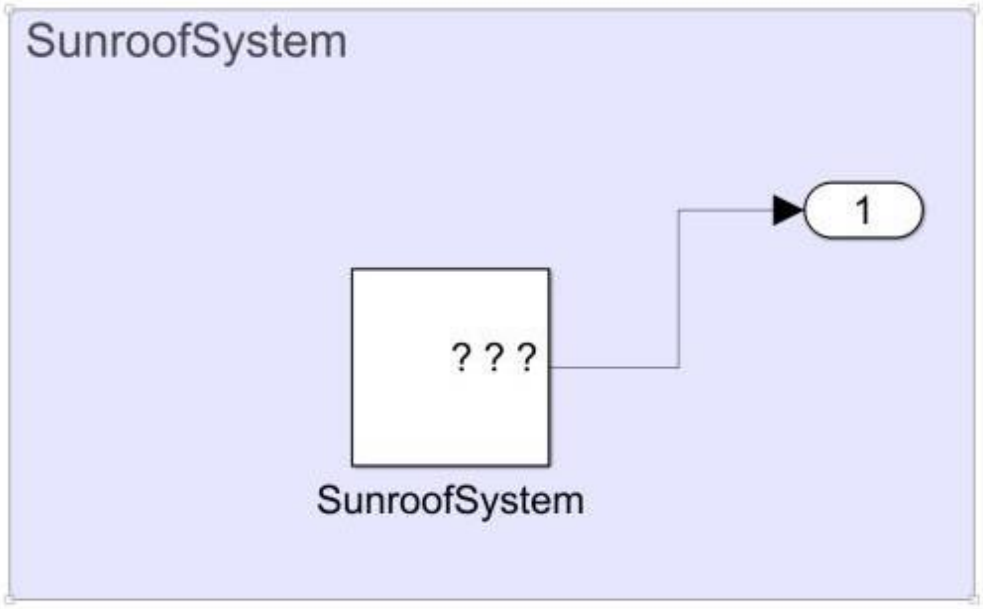
• A Sunroof can operate in dual ways, firstly, the whole roof and the tilt function. With either of the two, one can drastically reduce the car compartment’s warmth. This helps with all the warm days and humidity and is a good feature, especially in India

DESIGN

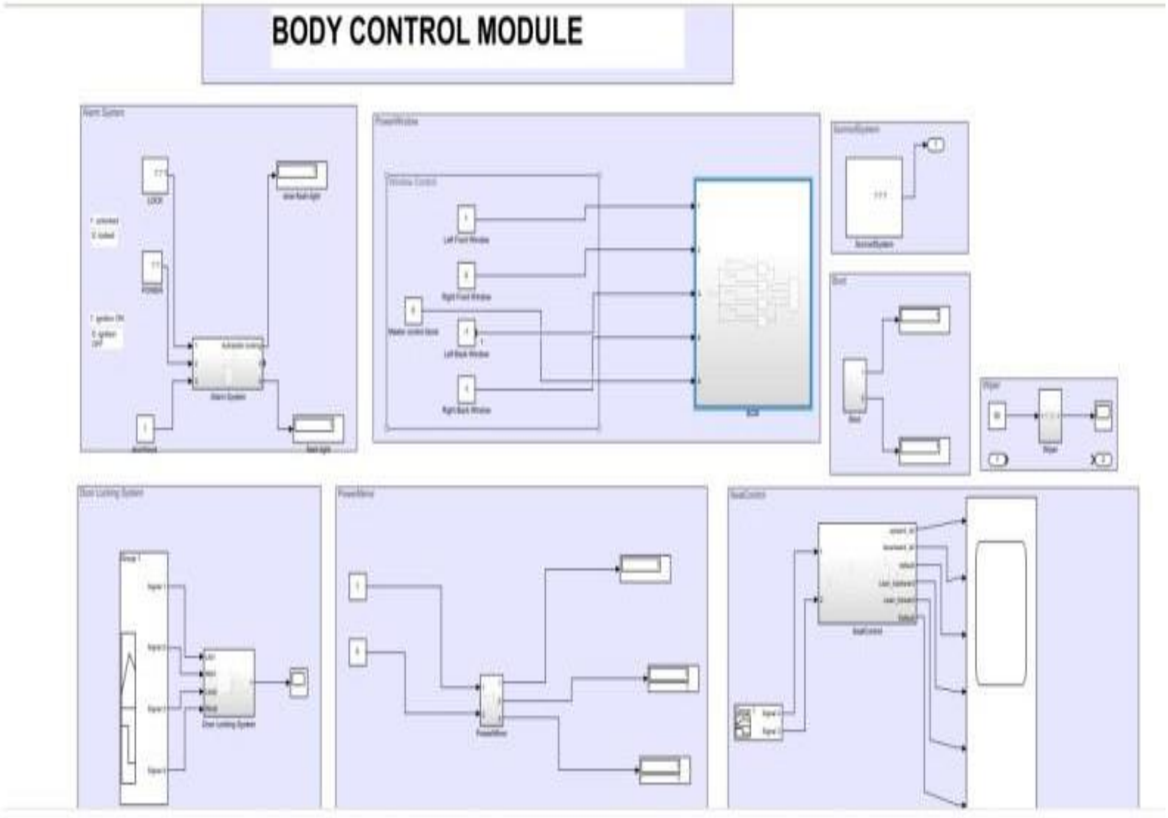
DESIGN



MODELING



Team Activity



High Level Requirements

ID	Description
HLR1	Receive signals from the User.
HLR2	All conditions should pass. One condition should pass at a time.
HLR3	Manual and Automatic lock

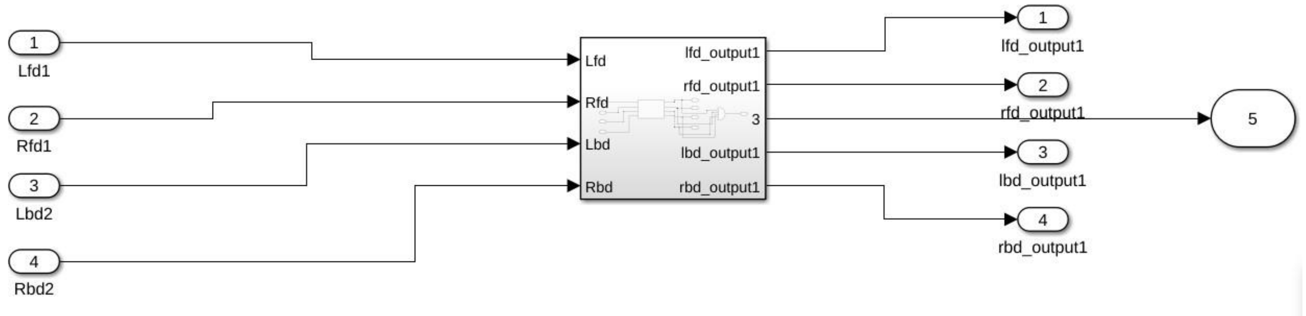
Low Level Requirements

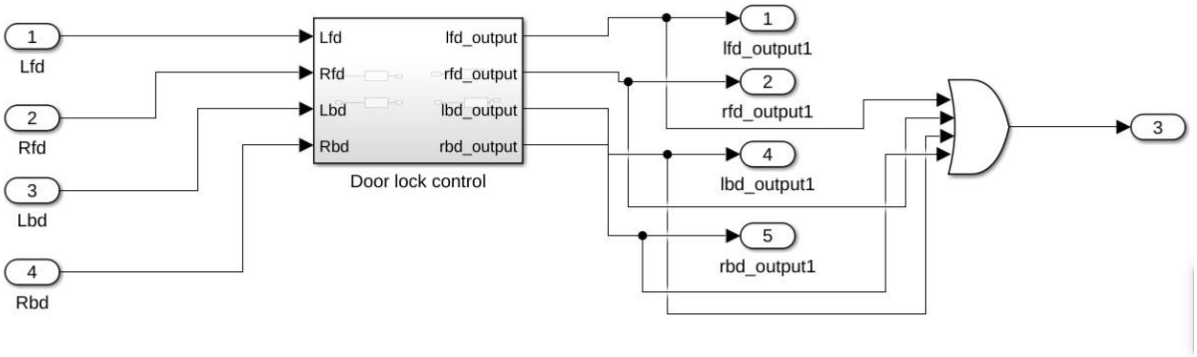
ID	Description
LLR1	Put the input value.

LLR2	It should unlock and lock if the Smart key near to the car.
LLR3	Compare the conditions.

Design

This project was implemented using Matlab.

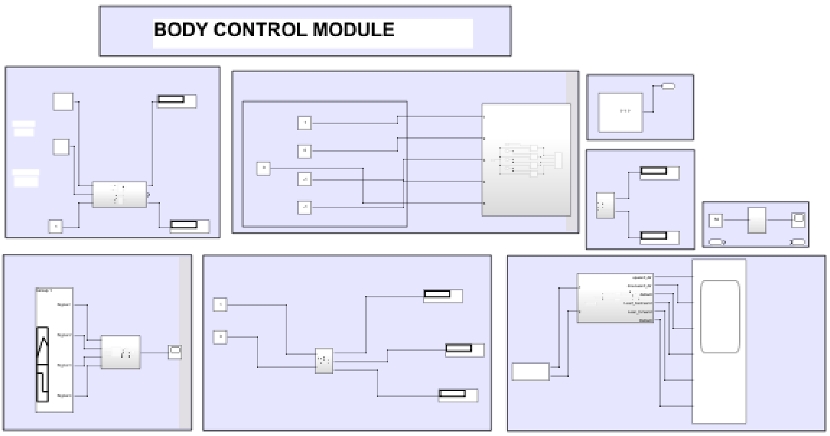




OUTPUT

Running the Simulation in Central Door control Mode

Team Activity



Mini Project 6 – Wiper Control [Team]

Modules

1. C Programming
2. STM32

Requirements

4W's and 1'H

Who:

• A wiper speed control system for an automotive wiper controls the operational speed of a wiper in accordance with rain conditions.

What:

• Vehicles are now available with driver-programmable intelligent (automatic) windscreen wipers that detect the presence and amount of rain using a rain sensor.

When:

• Whenever the water hit the rain sensor that located on windscreen, it will send a signal to move on the wiper motor. Once water is not detected by sensor, the wiper will automatically stop. This will help the driver to give more concentration and reduce the car accident probability.

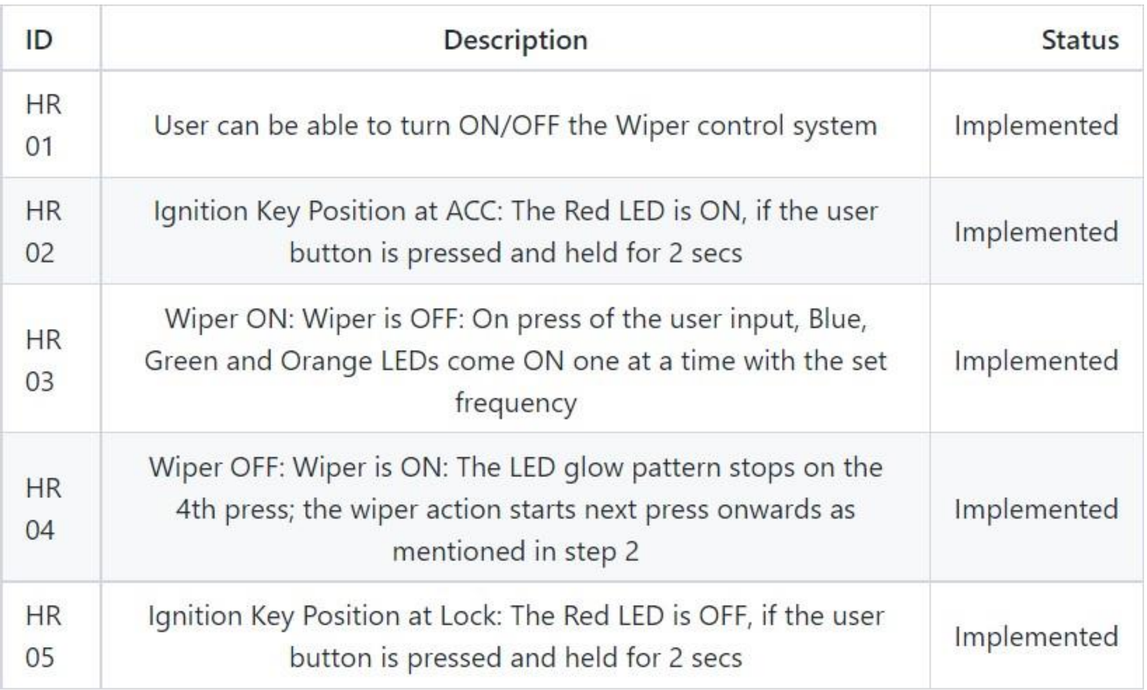
Where:

• It is located on the dashboard, above the brake and accelerator pedal, and is responsible for the complete operation of the windshield wiper system.

How:

• Windshield wipers are controlled by the stalk on the right side of your steering wheel. Simply moving the stalk down will turn your windshield wipers on. Moving the stalk down will turn your you wipers on.

High Level Requirements



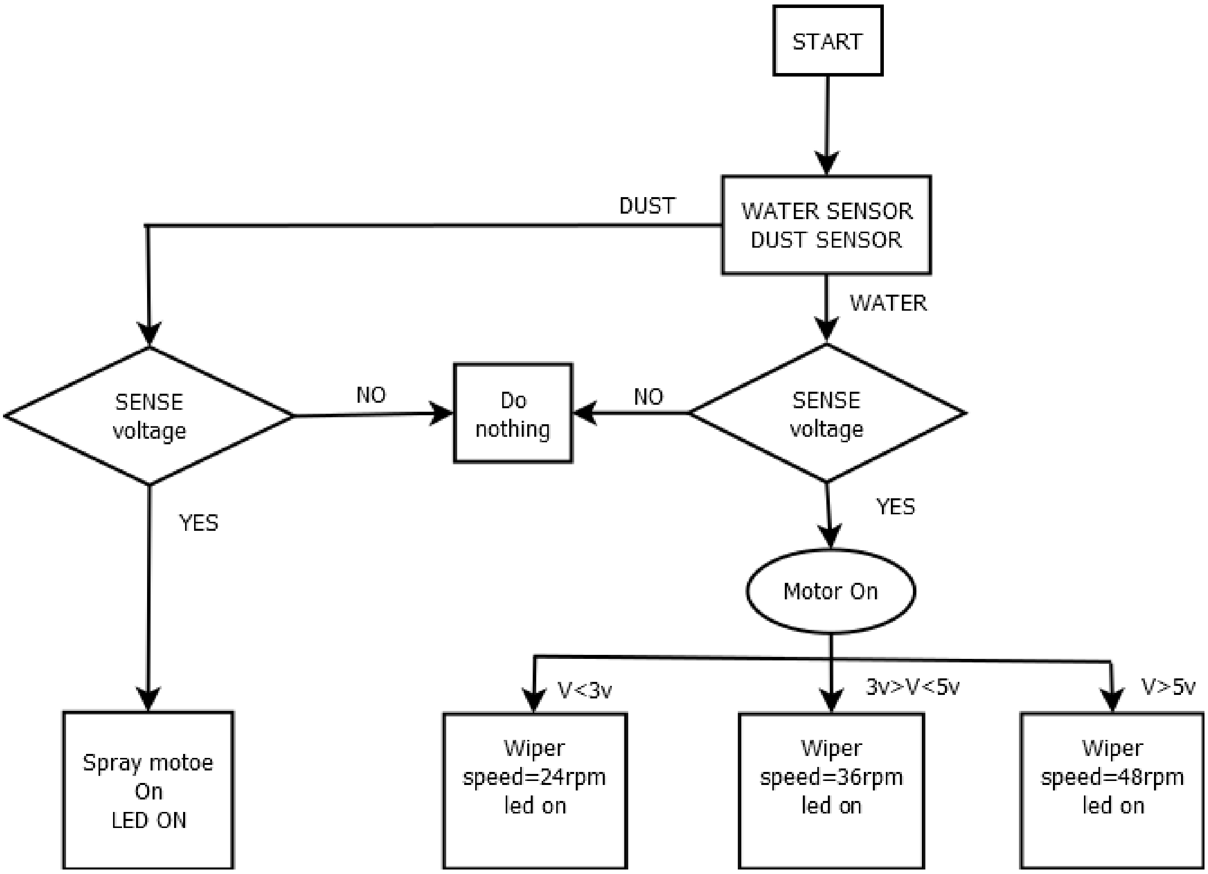
ID	Description	Status
HR 01	User can be able to turn ON/OFF the Wiper control system	Implemented
HR 02	Ignition Key Position at ACC: The Red LED is ON, if the user button is pressed and held for 2 secs	Implemented
HR 03	Wiper ON: Wiper is OFF: On press of the user input, Blue, Green and Orange LEDs come ON one at a time with the set frequency	Implemented
HR 04	Wiper OFF: Wiper is ON: The LED glow pattern stops on the 4th press; the wiper action starts next press onwards as mentioned in step 2	Implemented
HR 05	Ignition Key Position at Lock: The Red LED is OFF, if the user button is pressed and held for 2 secs	Implemented

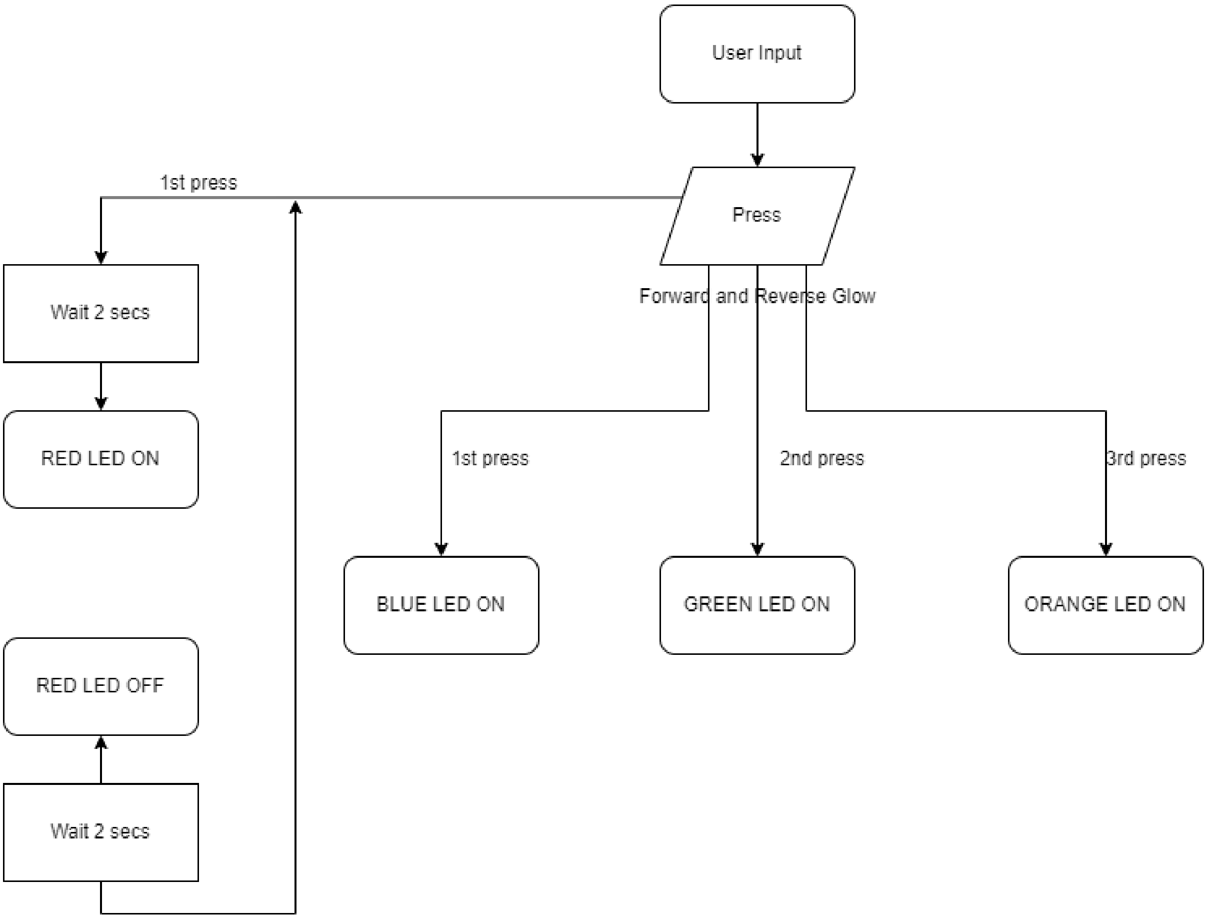
ID	Description	Status
HLR_1	Press and hold the button to put the Ignition key position in ACC mode	Implemented
HLR_2	Different wiper frequencies to be set (1Hz, 4Hz & 8Hz)	Implemented
HLR_3	Hold the button to put the system in Idle state	Implemented

Low Level Requirements

ID	Description	HLTP ID	Status
LLR_1	Hold the button for 2 sec to bring the ignition key position at ACC mode	HLR_1	Implemented
LLR_2	Hold the button for 2 sec to go back to the Idle state	HLR_1, HLR_3	Implemented
LLR_3	Press the button one time to set frequency to 1Hz	HLR_2	Implemented
LLR_4	Press the button second time to set frequency to 4Hz	HLR_2	Implemented
LLR_5	Press the button third time to set frequency to 8Hz	HLR_2	Implemented
LLR_6	Press the button fourth time to turn OFF the wiper action	HLR_2	Implemented
LLR_7	Hold the button for 2 sec to bring ignition key position at Lock state	HLR_3	Implemented

Design


START
WATER SENSOR
DUST SENSOR
DUST
WATER
SENSE voltage
NO
Do nothing
NO
SENSE voltage
YES
YES
Motor On
V<3v
3v>V<5v
V>5v
Spray motoe On LED ON
Wiper speed=24rpm led on
Wiper speed=36rpm led on
Wiper speed=48rpm led on



Test Plan

High Level Test Plan

ID	Description	Output	Type of Test
HLTP_1	Press and hold the button to put the Ignition key position in ACC mode	System Enters ACC State	Requirement Based
HLTP_2	Different wiper frequencies to be set (1Hz, 4Hz & 8Hz)	Responds Based on Input	Requirement Based
HLTP_3	Hold the button to put the system in Idle state	Enters Idle State	Requirement Based

Low Level Test Plan

ID	Description	Output	HLTP ID	Type of Test
LLTP_1	Hold the button for 2 sec to go back to the Idle state	Red LED-OFF	HLTP_1, HLTP_3	Requirement Based
LLTP_2	Hold the button for 2 sec to bring the ignition key position at ACC mode	Red LED-ON	HLTP_1	Requirement Based
LLTP_3	Press the button one time to set frequency to 1Hz	Blue LED-ON	HLTP_2	Requirement Based
LLTP_4	Press the button second time to set frequency to 4Hz	Green LED-ON	HLTP_2	Requirement Based
LLTP_5	Press the button third time to set frequency to 8Hz	Orange LED-ON	HLTP_2	Requirement Based
LLTP_6	Press the button fourth time to turn OFF the wiper action	All LED OFF except Red	HLTP_2	Requirement Based
LLTP_7	Hold the button for 2 sec to bring ignition key position at Lock state	Red LED-OFF	HLTP_3	Requirement Based

Implementation and Summary Git Link:
Link: https://github.com/GENESIS-2022/MasteringMCU-Team46

Mini Project 7 – LAMBHORGINI[Team]

Module:

Automotive Systems

Requirements:

In this project Lamborghini we have taken following features and I have contributed to Parking System Feature

1. Wiper Control System
2. Sunroof control system
3. Power window system
4. Seat belt control

Research:

Automation is a necessity in our day-to-day life because it not only seeks to improve the quality of life for humans at both home and work.

It allows the distribution of both quality products and services to be made available at faster rates, reduces time and human error.

These days the technology is developing at very high-speed innovation and advancements is there in all the sectors.

SWOT- Strengths, and Weakness, Opportunities Threats:

Strengths:

Comfortable, Easy to use, Fast and accuracy operation, Easy transportation

Weakness:

High cost

Opportunities:

it's a modern way of approach, High use in modern days

Threat:

if sensor is damages, operation of the system will stop

4W's and 1'H:

WHO:

Any user who is travelling in the car can use this system.

WHAT:

It’s a Body control model of a car which describes different features

WHEN:

There is a requirement of vehicle these systems are used

WHERE:
Used in automatic cars

HOW:
By developing an embedded system which is user friendly and can be implemented without difficulty.

Detail requirements:

High Level Requirements:

Number	Requirements	Description	Features
HLR1	Rain and Humidity sensor	checks the intensity of rain and humidity	Wiper Control
HLR2	Semi-Automatic control	Operated automatically or manually	Wiper Control
HLR3	Speed control	High, Medium, Low	Wiper Control
HLR4	Dry mode	front and back wiper system	Wiper Control
HLR5	Obstacle Sensor	when obstacle detected the operation is delayed for 10seconds.	Power window
HLR6	Both driver and passenger control	operated with the first received command (either by driver or passenger)	Power window
HLR7	Dust detected sensor	window will be closed when the dust is detected	Power window
HLR8	Temperature and Rain sensor	Which detects intensity of rain and temperature	Sunroof Control
HLR9	Obstacle sensors	when obstacle is detected operation of window will be stopped	Sunroof Control
HLR10	Pressure control alarm	alarm buzzes when seat belt is not buckled up even the pressure is applied	Seatbelt Control
HLR11	Open airbag	depending upon the tension airbag will be opened	Seatbelt Control
HLR12	buckles and its types	lap belts, diagonal belts, the harness, combination of lap and diagonal belts.... etc.	Seatbelt Control
HLR13	Height adjustment of seat belt	Height adjustment of seat belt	Seatbelt Control

Low Level Requirements:

Number	Requirements	Description	Features
LLR1	Length of wipers	Depends on the model of the car	Wiper Control
LLR2	Direction of	Same and Opposite Directions (180	Wiper Control

Number	Requirements	Description	Features
	wipers	degrees)	
LLR3	Semi controlled	manually/automatic	Power window and Sunroof
LLR4	Length of the belt	Depends on the model of the car	Seat Belt control
LLR5	Engine On	Can be operated manually or automatically	wiper control, power window, sunroof control
LLR6	Engine Off	Can be operated manually	Power window and Sunroof control

Design:

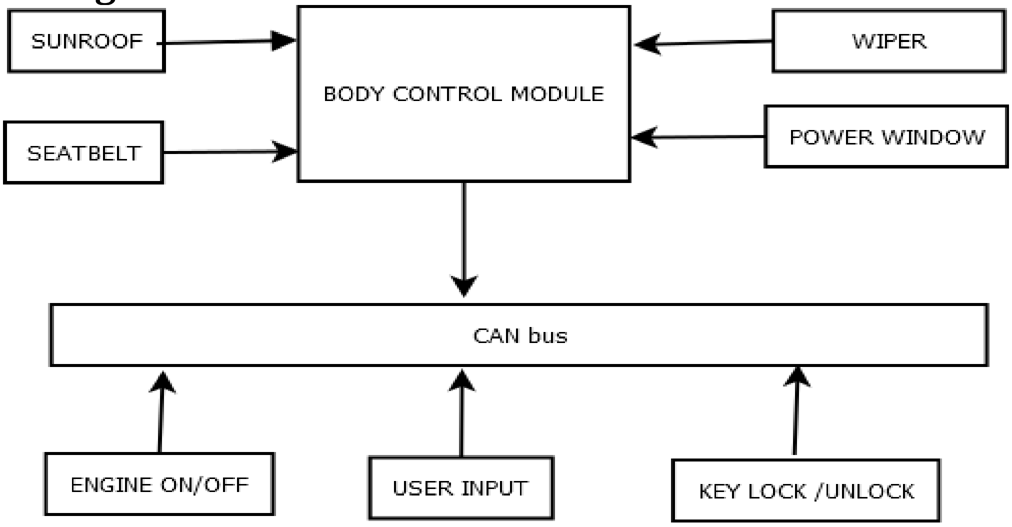


Figure 1 Structure Diagram

Implementation and Summary:

Git Link:

Link: https://github.com/pasumarthiharitha/Automotive_Lamborghini.git

Individual Contribution and Highlights:

1. Seatbelt Case Study
2. Source code management using GitHub

Role in Project Team:

1. Designer: Done Designing for Project
2. Researcher: Done case study for Seatbelt control

Mini Project 8 – EV Car [Team] Modules

Module:

- Applied Control Systems and Vehicle Dynamics
- MATLAB
- MATLAB Script

Requirements:

1.Battery:

1.Battery type used in this EV is Lithium-ion Polymer.
2.Lithium is also the lightest of all metals. However, lithium-ion (Li-ion) batteries contain no lithium metal, they contain ions. For those wondering what an ion is, an ion is an atom or molecule with an electric charge caused by the loss or gain of one or more electrons.

Battery Features:

3.Lithium-ion batteries are one of the most popular forms of energy storage in the world, accounting for 85.6% of deployed energy storage systems.

2.Motor:

1.BLDC motors are used, which have traction characteristics like high starting torque, high efficiency around 95 98%, etc.
2.BLDC motors are suitable for high power density design approach. The BLDC motors are the most preferred motors for the electric vehicle application due to its traction characteristics.

Motor Features:

3.Electronically commutated
4.High-energy, rare-earth magnets used for rotor field
5.Requires speed control with 6-lead connection (3 power,2 Hall Effects, 1 drain)
6.Rated speed 2500 RPM; minimum 150-200 RPM
7.Linear speed torque curves
8.Built-in tach pulse for economical speed readout
9.Encoder options for servo performance.

3.Controller:

1.Programmable BLDC motor controller provides efficient smooth and quite control for electric vehicles which operate 36V/48V battery system.

controller Features:

1.E- Braking will release the motor from the controller as long as brake is applied.
2.Over current protection to protect controller during faulty conditions or short circuit.
3.Low voltage detection ensures Battery life.
4.Pedal assist mode controls the motor speed based on the speed of peddling.
5.Accelerator fault protection to prevent runway.
6.Provision for 120°/ 60° Selection.
7.Speedometer output.
8.Brake inputs with high active and low active provision.
9.Speed limit control provision.

4.Inverter:

1.The traction inverter converts energy from the vehicle's battery in order to drive the motors in the drivetrain. This key component has a direct impact on road performance, driving range and reliability of the vehicle also as a consequence of their weight and size.
2.Subject to all the possible stress found in a road vehicle from heat and vibrations, these converters must be able to handle high power and currents along with the associated Electro Magnetic Compatibility (EMC) challenges as well as provide fail-safe operation to ensure dependability and safety for the driver and passengers.
3.To help developers increase the automotive inverter's power efficiency and reduce size and weight, ST has a wide offer of discrete semiconductors including AEC-Q101 qualified IGBTs and both silicon and silicon-carbide (SiC) MOSFETs and diodes, AEC-Q100 qualified galvanically isolated IGBT and MOSFET gate drivers and SPC5 32-bit automotive microcontrollers for designing scalable, cost-effective and energy-efficient EV traction inverter solutions.

Inverter features:

4.Traction inverters are typically capable of transferring power in the 20 to 100 kW range, with switching voltages in the 200 V to 800 V range and currents in the hundreds of amperes.

Implementation and Summary:

Submission: Submitted in GEA Learn

Individual Contribution and Highlights

1. Done in MATLAB Script

Role in Project Team:

1. Done MATLAB scripting for EV Car
2. Researcher: Done case study for EV Car

Mini Project 9 –Seatbelt [Individual]

Modules

1. AUTOSAR
2. Git

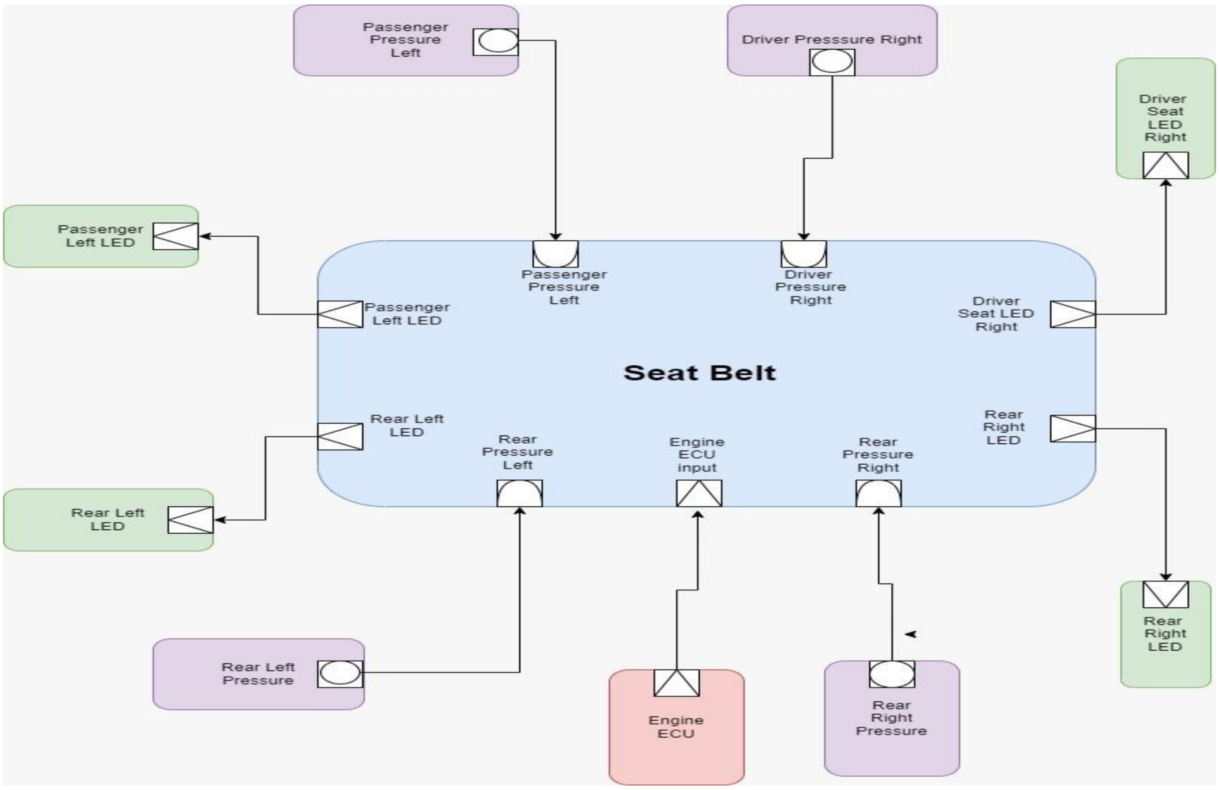
Power Window:
High level Requirements

Number	Requirements	Description
HLR1	Pressure control alarm	alarm buzzes when seat belt is not buckled up even the pressure is applied
HLR2	Open airbag	depending upon the tension airbag will be opened
HLR3	buckles and its types	lap belts, diagonal belts, the harness, combination of lap and diagonal belts.... etc.
HLR4	Height adjustment of seat belt	Height adjustment of seat belt

Low level requirement

ID	Requirements	Description
LLR1	Length of the belt	Depends on the model of the car

Design



Implementation and Summary Git Link:

Link: https://github.com/pasumarthiharitha/Automotive_Lamborghini.git