

GENESIS - Learning Outcome & Mini-project Summary Report



L&T Technology Services

Details

Ver. Rel. No.	Release Date	Prepared By	Module Name	To Be Approved	Remarks/Revision Details
1.0	17/02/2022	Vishnu Prasath R K 40020535	C Programming On Multiple Platforms		
1.0	17/02/2022	Vishnu Prasath R K 40020535	Essentials of Embedded Systems		
1.0	17/02/2022	Vishnu Prasath R K 40020535	Applied SDLC and Software Testing		
1.0	17/02/2022	Vishnu Prasath R K 40020535	Applied Model Based Design Module		
1.0	17/02/2022	Vishnu Prasath R K 40020535	OOPS with Python		
1.0	17/02/2022	Vishnu Prasath R K 40020535	Mastering Microcontrollers with Embedded Driver Development Module		
1.0	17/02/2022	Vishnu Prasath R K 40020535	Overview of Automotive Systems		
1.0	17/02/2022	Vishnu Prasath R K 40020535	Applied Control Systems and Vehicle Dynamics		
1.0	17/02/2022	Vishnu Prasath R K 40020535	Classic AUTOSAR Basic to Intermediate		

Contents

LIST OF FIGURES:	6
Miniproject – 1: Library Management System [Individual]	7
Modules:	7
Requirements.....	7
4W's and 1 H's.....	7
High Level Requirements	8
Low Level Requirements	8
Design.....	9
Test Plan.....	10
High Level Test Plan	10
Low Level Test Plan	11
Implementation and Summary	11
Git Link:	11
Git Dashboard	11
Miniproject 2 – Seat Heater [Individual]	12
Modules	12
Requirements.....	12
4W's and 1 H's.....	12
Design.....	13
Test Plan.....	13
High Level Test Plan	13
Low Level Test Plan	14
Implementation and Summary	14
Git Link:	14
Git Dashboard	14
Miniproject 3 – NFT Marketplace [Team]	15
Modules	15
Requirements.....	15
4W's and 1 H's.....	15
High Level Requirements	16
Low Level Requirements	16
Design.....	17
Test Plan.....	18
High Level Test Plan	18

Low Level Test Plan	19
Implementation and Summary	19
Git Link:	19
Individual Contribution and Highlights	19
Summary	19
Miniproject 4 – Attendance Automation [Team]	20
Modules	20
Requirements	20
High Level Requirements	20
Low Level Requirements	20
Test Plan	21
High Level Test Plan	21
Low Level Test Plan	21
Implementation and Summary	22
Git Link:	22
Git Dashboard	22
Git Inspector Summary	23
Individual Contribution and Highlights	23
Role in Project Team:	23
Miniproject 5 – Team Tesla [Team]	24
Modules	24
Requirements	24
Design	24
GitHub Link:	25
Miniproject 6 – Wiper Control [Team]	26
Modules	26
Requirements	26
4W's and 1'H	26
High Level Requirements	26
Low Level Requirements	27
Design	27
LOW LEVEL DIAGRAM	27
LEVEL 1	27
LEVEL 2	28
LEVEL 3	28

HIGH LEVEL DIAGRAM.....	29
Test Plan.....	29
High Level Test Plan	29
Implementation and Summary	30
Git Link:	30
Individual Contribution and Highlights	30
Miniproject 7 – Project Volkswagen [Team].....	31
Modules	31
REQUIRMENTS	31
INTRODUCTION.....	31
FEATURES.....	31
COMPONENTS.....	31
High Level Requirements	32
Low Level Requirements.....	32
Design.....	33
Role in Project Team	33
Miniproject 8 – EV Car [Team].....	34
Modules	34
Requirements.....	34
Introduction	34
Implementation and Summary	34
Individual Contribution and Highlights	34
Miniproject 9 – Defogger [Individual]	35
Modules	35
Requirements.....	35
Design.....	35
Implementation and Summary	36
Git Link:	36

LIST OF FIGURES:

Figure 1 Behavior Diagram	9
Figure 2 Structure Diagram	10
Figure 3 Git Dashboard.....	11
Figure 4 Behavior Diagram	13
Figure 5 Git Dashboard.....	14
Figure 6 Behavior Diagram	17
Figure 7 UserFlow Diagram	17
Figure 8 Structure Diagram	18
Figure 9 Git Dashboard.....	22
Figure 10 Git Inspector Summary.....	23
Figure 11 Structure Diagram	27
Figure 12 Behavior Diagram	29
Figure 13 Structure Diagram	33
Figure 14 VFB Diagram	35

Miniproject – 1: Library Management System [Individual]

Modules:

1. C Programming
2. Git

Requirements

4W's and 1 H's

Why:

Adding, deleting, viewing and searching of books can be done by this system without manual efforts of the librarian.

Where:

The Library management system is nowadays essential for schools, colleges, private libraries, and other organizations. They can use this software as the purpose of books issuing and returning for renewal.

Who:

The project can be used almost by all the particular organization people. At the end, user satisfaction is the goal of the project

When:

The project can be used when the book needs to be added, issued or removed.

How:

System design is a solution for how to approach to the creation of a new system. It translates system requirements into ways by which they can be made operational. It is a translational from a user oriented document to a document oriented programmers. For that, it provides the understanding and procedural details necessary for the implementation. Here UML diagrams are used to supplement the working of the new system. The system thus made should be reliable, durable and above all should have least possible maintenance costs.

High Level Requirements

ID	Description	Status
HR_01	Operating System (Windows 10/Linux)	Implemented
HR_02	C language	Implemented
HR_03	Pentium IV Processor	Implemented
HR_04	RAM(512MB)	Implemented
HR_05	Hard Disk(2GB)	Implemented

Low Level Requirements

ID	Description	Status
LR_01	Functions to Add book	Implemented
LR_02	Functions to View books available	Implemented
LR_03	Functions to Search the book by ID	Implemented
LR_04	Functions to Update status of the book	Implemented
LR_05	Functions to Delete the book by ID	Implemented

Design

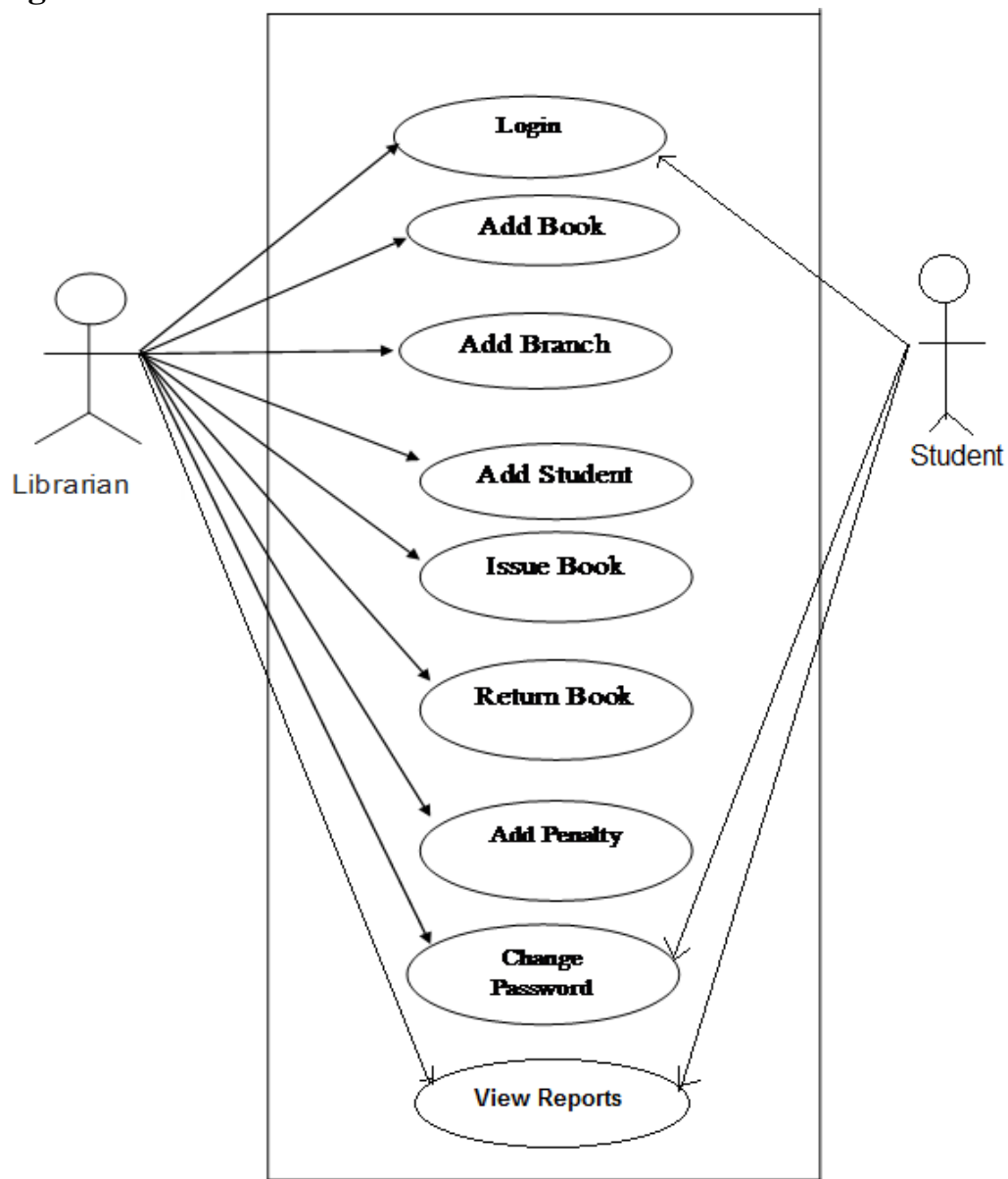


Figure 1 Behavior Diagram

LIBRARY MANAGEMENT SYSTEM

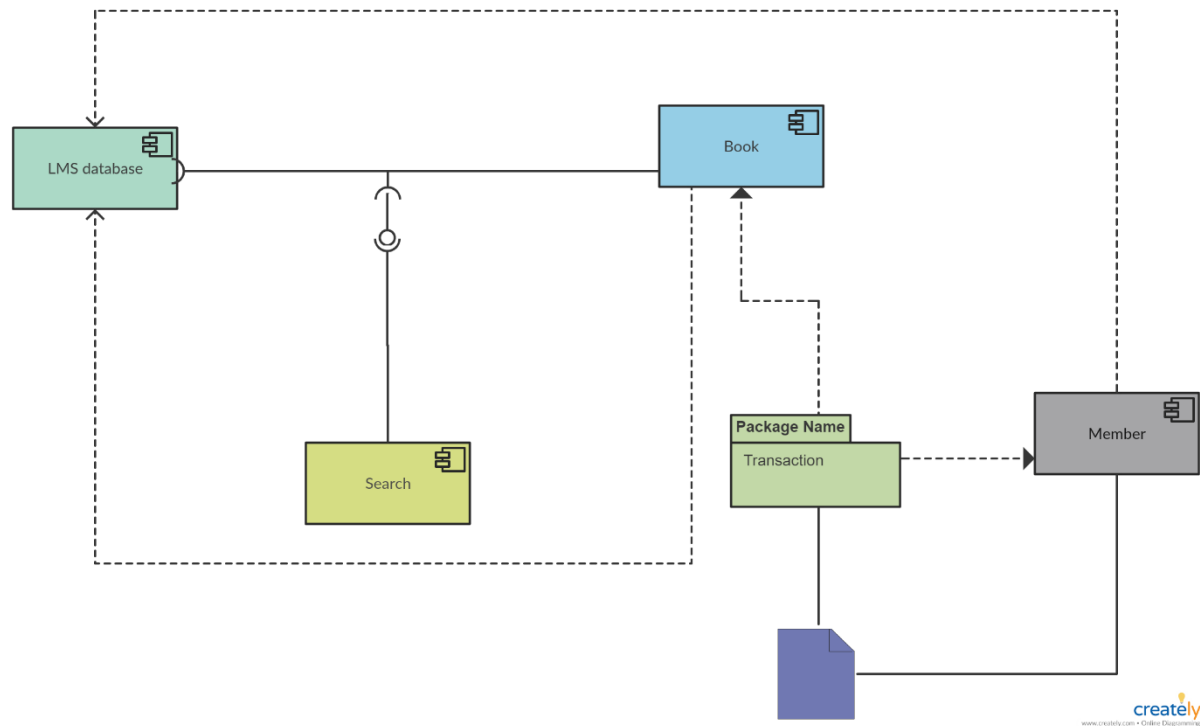


Figure 2 Structure Diagram

Test Plan

High Level Test Plan

Test ID	Description	Exp I/P	Exp O/P	Actual Out	Type Of Test
H_01	Add book	1234,"cs"	1	1	Scenario based
H_02	View book	1234,"cs"	1	1	Scenario based
H_03	Search book	1234	1	1	Scenario based
H_04	Update book	1234, "issued", "16/02/21", "31/03/21", "Sunil","R", 900	1	1	Scenario based

Test ID	Description	Exp I/P	Exp O/P	Actual Out	Type Of Test
H_05	Delete book	1234	1	1	Scenario based

Low Level Test Plan

Test ID	Description	Exp I/P	Exp O/P	Actual Out	Type Of Test
L_01	Add the details of new book	1234,"cs"	1	1	Scenario based
L_02	View the details of all books	1234 ,"cs"	1	1	Scenario based
L_03	Search the details of entered book by giving "ID" and "Name"	1234	1	1	Scenario based
L_04	update the book status	1234, "issued", "16/02/21", "31/03/21", "Sunil","R", 900	1	1	Scenario based
L_05	Delete the book by entering the "ID" of the book to delete	1234	1	1	Scenario based

Implementation and Summary

Git Link:

Link: https://github.com/Vishnu-prasath/M1_Application_Library_Management_System.git

Git Dashboard

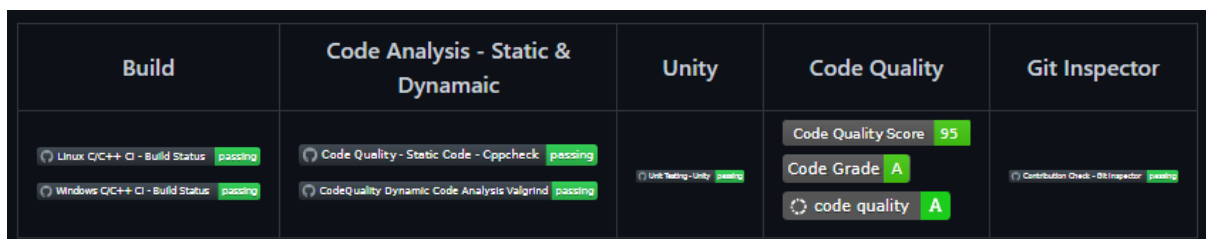


Figure 3 Git Dashboard

Miniproject 2 – Seat Heater [Individual]

Modules

1. C Programming
2. Embedded System
3. Simul IDE
4. Git

Requirements

Heated seats are powered by a heating element, a long strip of material that functions as a resistor. A resistor resists the flow of electricity. When electric current flows through it, the energy is turned into heat, which flows through the seat.

4W's and 1 H's

Why:

1. To maintain the temperature inside the car for not to catch cold.
2. To warm quickly from outside of the car.

Where:

1. This can be used in our cars.

Who:

1. Can be used by the drivers of the car

When:

1. When outside of the car is too cold
2. This project is used to prevent people from high cold.

How:

1. It sense the temperature by sensor and heat the seat through resistor.

.

Design

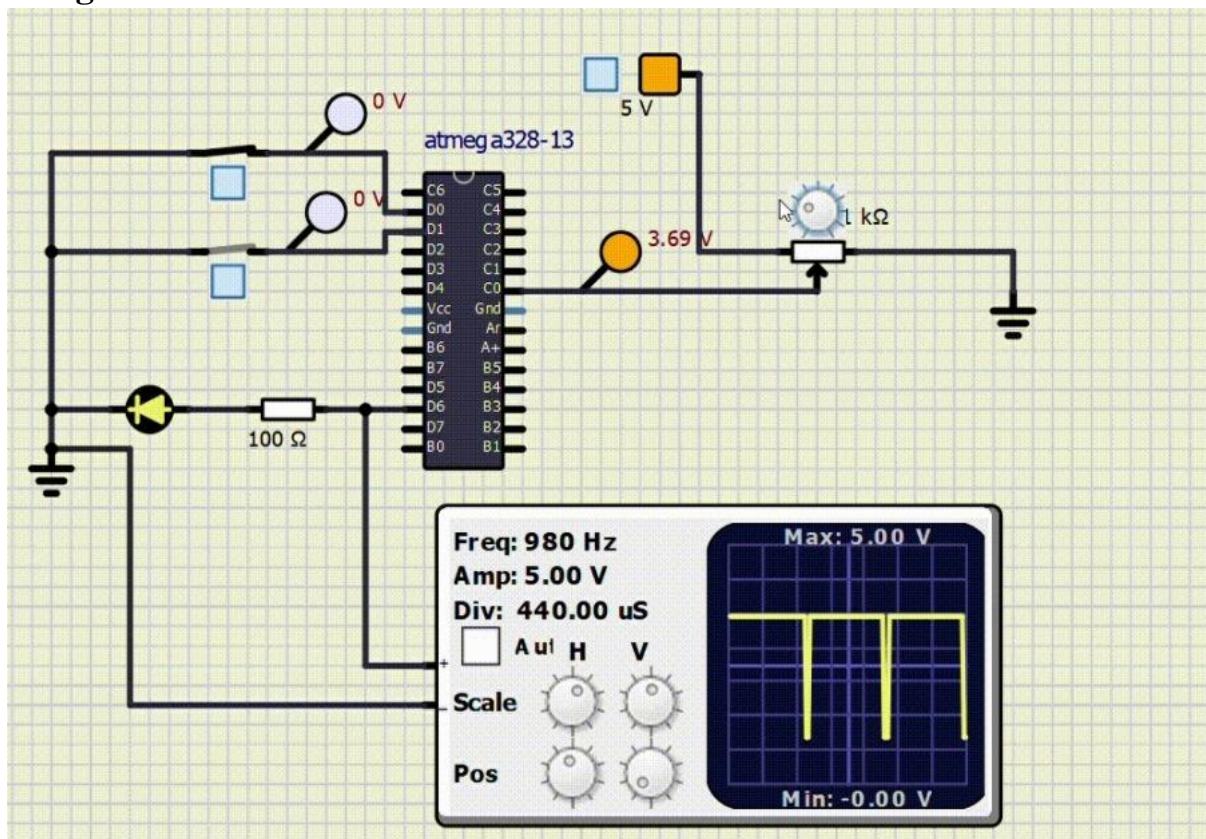


Figure 4 Behavior Diagram

Test Plan

High Level Test Plan

ID	Description	Expected I/P	Expected O/P	Actual O/P	Type Of Test
HLTP_1	Switch On	High	The Program Should Start	SUCCESS	Requirement Based
HLTP_2	Taking Input From The User	Value Input	Should Give Output To The User	SUCCESS	Scenario Based
HLTP_3	Giving Output From The Input	Value Input From The User	Displays Output	SUCCESS	Boundary Based

Low Level Test Plan

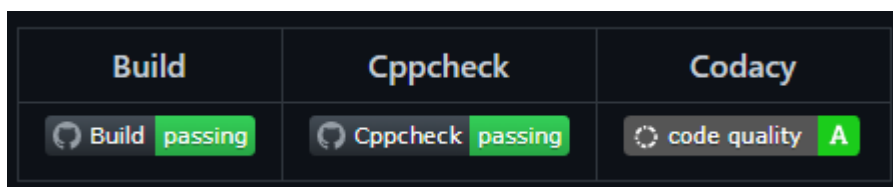
ID	Description	Expected I/P	Expected O/P	Actual O/P	Type Of Test
LLTP_1	Performing Arithmetic operation '+'	(600, 100)	700	700	Requirement Based
LLTP_2	Performing Arithmetic operation '-'	(800, 100)	700	700	Requirement Based
LLTP_3	Performing Arithmetic operation '*'	(10, 7)	70	70	Requirement Based
LLTP_4	Performing Arithmetic operation '/'	(10, 2)	5	5	Requirement Based

Implementation and Summary

Git Link:

Link: https://github.com/Vishnu-prasath/M2_Embedded_Seat_Heater.git

Git Dashboard



Badges

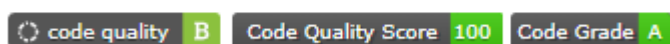


Figure 5 Git Dashboard

Miniproject 3 – NFT Marketplace [Team]

Modules

1. SDLC
2. Git

Requirements

4W's and 1 H's

Why:

1. It can be used by anyone at any place.
2. Digital Items are the Future
3. Individual Creators can use this platform to sell the Digital products.

Where:

1. This can be used in our daily lives.
2. Can be used for international transactions

Who

1. It can be used by anyone.
2. It can be used as a reference for marketplace.

When:

1. One can buy, sell or create anytime.
2. The project can be used when the anyone wants to buy an NFT.
3. Can be used without any centralised authority

How:

1. By using a crypto wallet anyone can Buy or Bid on NFT.
2. It will be helpful for Digital Creators.

High Level Requirements

ID	Description	Status
HLR_1	Create NFT	Implemented
HLR_2	Sell NFT	Implemented
HLR_3	Bid NFT	Implemented
HLR_4	Buy NFT	Implemented
HLR_5	Contact	Implemented

Low Level Requirements

ID	Description	Status
LLR_1	Sign In	Implemented
LLR_2	Register	Implemented
LLR_3	Connect Wallet	Implemented
LLR_4	Activity	Implemented
LLR_5	Forgot password	Implemented
LLR_6	Signup	Implemented

Design

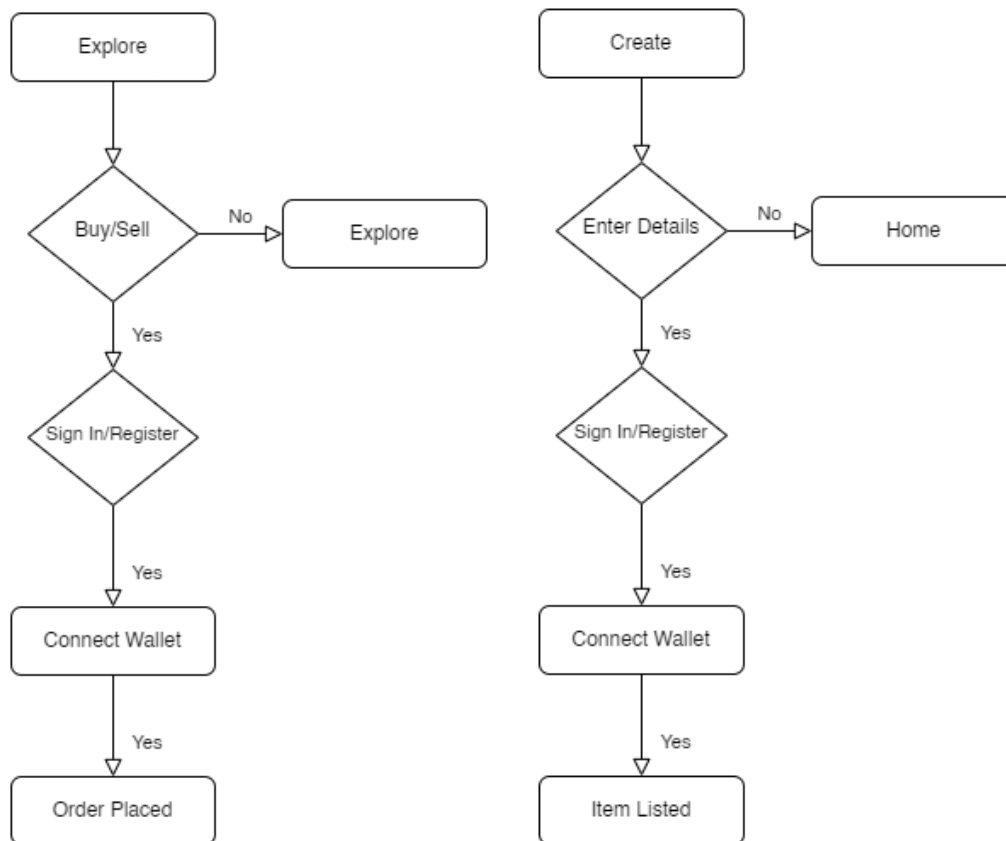


Figure 6 Behavior Diagram

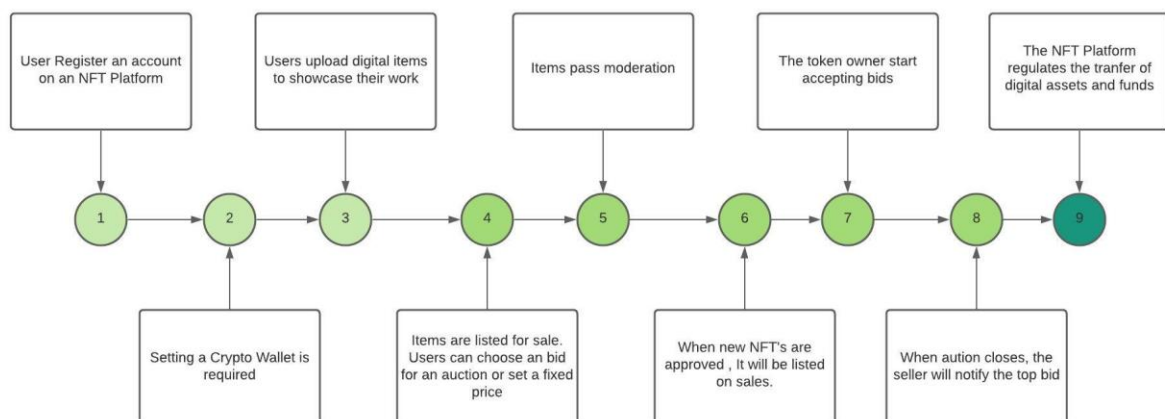


Figure 7 UserFlow Diagram

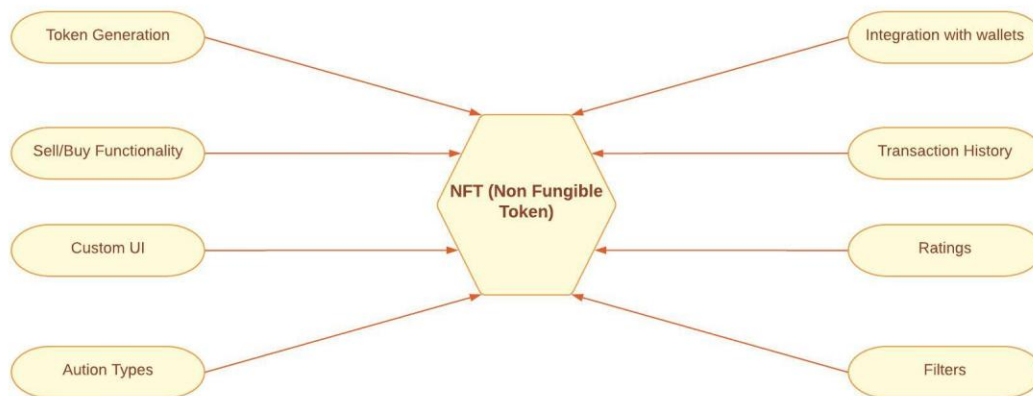


Figure 8 Structure Diagram

Test Plan

High Level Test Plan

ID	Description	Expected I/P	Expected O/P	Actual O/P	Type Of Test
HLTP_1	Create NFT	Click	SUCCESS	SUCCESS	Requirement Based
HLTP_2	Sell NFT	Click	SUCCESS	SUCCESS	Requirement Based
HLTP_3	Bid NFT	Click	SUCCESS	SUCCESS	Requirement Based
HLTP_4	Buy NFT	Click	SUCCESS	SUCCESS	Requirement Based
HLTP_5	Contact	Click	SUCCESS	SUCCESS	Requirement Based
HLTP_6	Sign In	Click	SUCCESS	SUCCESS	Requirement Based
HLTP_7	Register	Click	SUCCESS	SUCCESS	Requirement Based

Low Level Test Plan

ID	Description	Expected I/P	Expected O/P	Actual O/P	Type Of Test
LLTP_1	Connect Wallet	Click	SUCCESS	SUCCESS	Requirement Based
LLTP_2	Activity	Click	SUCCESS	SUCCESS	Requirement Based
LLTP_3	Forgot password	Click	SUCCESS	SUCCESS	Requirement Based
LLTP_4	To check whether none of the fields should be empty	Empty value in the input module	Prompt message mandatory field missing	SUCCESS	Requirement Based
LLTP_5	E-mail ID should be in the perfect format i.e. group2@gmail.com	group2@gmail.com	Prompt message invalid E-mail ID	SUCCESS	Requirement Based

Implementation and Summary

Git Link:

Link: https://github.com/GENESIS2021Q1/Applied_SDLC-Dec_Team_2

Individual Contribution and Highlights

Summary

1. Designed Homepage
2. Search Option
3. Header
4. Footer
5. Integrating All Pages Together

Role in Project Team

1. Designer: Designed Webpages Using HTML, CSS, JavaScript
2. Integrator: Integrated All the Pages Together
3. Tester: Testing the Webpage Performance and Bugs

Miniproject 4 – Attendance Automation [Team]

Modules

1. Python
2. Git

Requirements

High Level Requirements

ID	Feature	Status
HLR_01	GUI	Not Implemented
HLR_02	Attendance Status	Implemented
HLR_03	User Details	Implemented
HLR_04	User load sheet	Implemented
HLR_05	Output file generation	Implemented

Low Level Requirements

ID	Feature	High Level ID	Status
LLR_01	GUI should allow user to enter inputs	HLR_01	Not Implemented
LLR_02	Input Files For Different Sessions	HLR_01	Not Implemented
LLR_03	User can get the Attendance Status	HLR_02	Implemented
LLR_04	User can enter status input to get the Attendance Status	HLR_02	Implemented
LLR_05	User can get the user details	HLR_03	Implemented
LLR_06	User will get the details after the successfull attendance entry	HLR_03	Implemented
LLR_07	User can load different sheets	HLR_04	Implemented

ID	Feature	High Level ID	Status
LLR_08	User can also modify the existing sheets as it is dynamic	HLR_04	Implemented
LLR_09	Output file gets generated	HLR_05	Implemented
LLR_10	Multiple files can be generated with different inputs	HLR_05	Implemented

Test Plan

High Level Test Plan

ID	Description	Expected I/P	Expected O/P	Actual O/P	Type Of Test
HLTP_01	Attendance Status	User Input	SUCCESS	SUCCESS	Requirement Based
HLTP_02	User details	User Input	SUCCESS	SUCCESS	Requirement Based
HLTP_03	User load sheet	User Input	SUCCESS	SUCCESS	Requirement Based
HLTP_04	Output file generation	User Input	SUCCESS	SUCCESS	Requirement Based

Low Level Test Plan

ID	HLTP ID	Description	Expected I/P	Actual O/P	Type Of Test
LLTP_01	HLTP_01	User can get Attendance Status	SUCCESS	SUCCESS	Requirement Based
LLTP_03	HLTP_02	User can get the User details	SUCCESS	SUCCESS	Requirement Based

ID	HLTP ID	Description	Expected I/P	Actual O/P	Type Of Test
LLTP_04	HLTP_02	User will get the details after the successful attendance	SUCCESS	SUCCESS	Requirement Based
LLTP_05	HLTP_03	User can load different sheets	SUCCESS	SUCCESS	Requirement Based
LLTP_06	HLTP_03	User can also modify the existing sheets as it is dynamic	SUCCESS	SUCCESS	Requirement Based
LLTP_07	HLTP_04	Output file gets generated	SUCCESS	SUCCESS	Requirement Based
LLTP_08	HLTP_04	Multiple files can be generated with different inputs	SUCCESS	SUCCESS	Requirement Based

Implementation and Summary

Git Link:

Link: https://github.com/alrichroshan/Attendance_Automation_Team_14.git

Git Dashboard

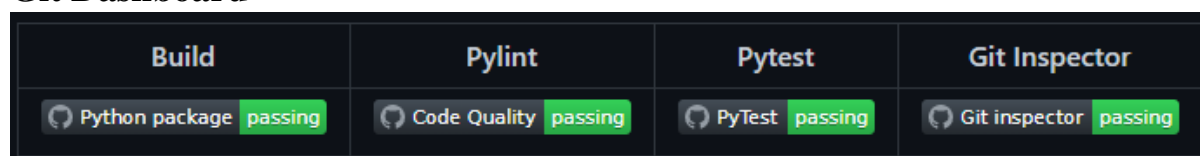


Figure 9 Git Dashboard

Git Inspector Summary

23	Author	Commits	Insertions	Deletions	% of changes
24	Alrich Roshan	4	190	1	35.83
25	Dileep Kumar Varadar	1	1	1	0.38
26	Sreenithy Thayanithy	3	53	20	13.70
27	Vishnu-prasath	3	2	2	0.75
28	alrichroshan	9	105	63	31.52
29	cedricxavi	2	25	3	5.25
30	gulamsuhail00	2	9	9	3.38
31	lokesh4309	1	2	2	0.75
32	muthupbalag1310	1	5	5	1.88
33	subramanikeerthana	1	14	12	4.88
34	vanisreekathirvel	5	8	1	1.69
35					
36	Below are the number of rows from each author that have survived and are still				
37	intact in the current revision:				
38					
39	Author	Rows	Stability	Age	% in comments
40	Alrich Roshan	88	46.3	0.1	0.00
41	Dileep Kumar Varadar	1	100.0	0.1	0.00
42	Sreenithy Thayanithy	7	13.2	0.1	0.00
43	alrichroshan	55	52.4	0.1	0.00
44	cedricxavi	15	60.0	0.1	0.00
45	gulamsuhail00	2	22.2	0.1	0.00
46	subramanikeerthana	14	100.0	0.0	0.00
47	vanisreekathirvel	1	12.5	0.1	0.00

Figure 10 Git Inspector Summary

Individual Contribution and Highlights

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

Role in Project Team:

1. Programmer: Done Programming for Attendance Automation
2. Integrator: Integrated all the codes
3. Tester: Writing Testcases and testing the integrated code

Miniproject 5 – Team Tesla [Team]

Modules

1. Matlab
2. Git

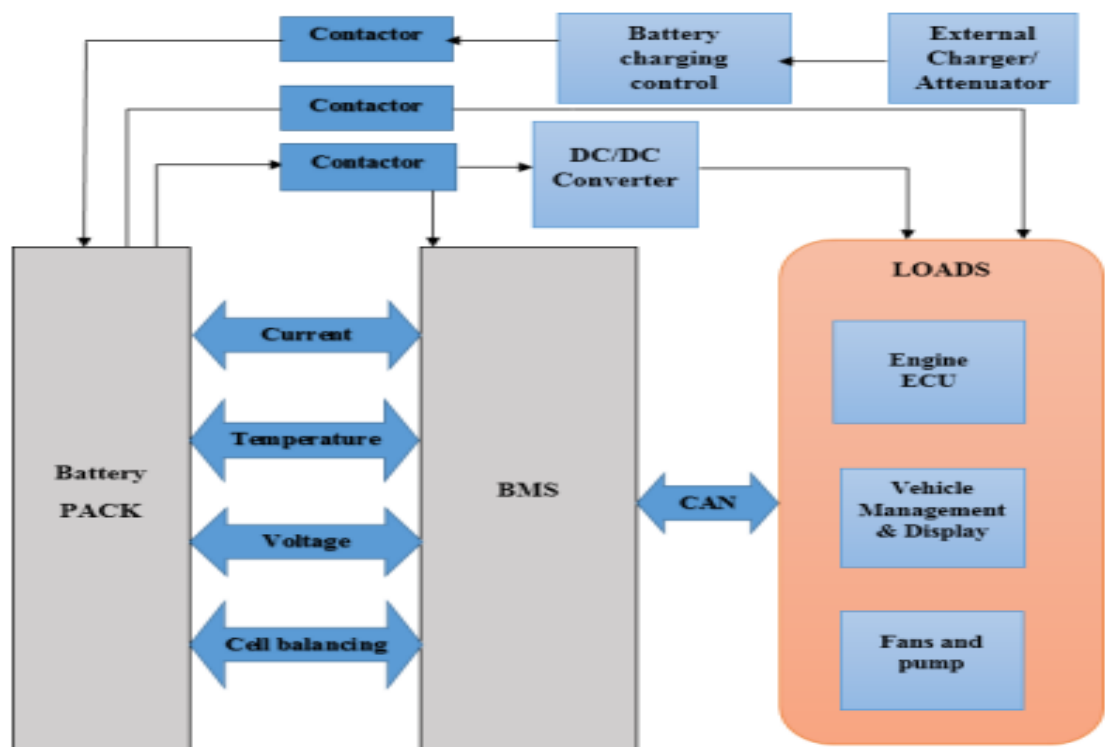
Requirements

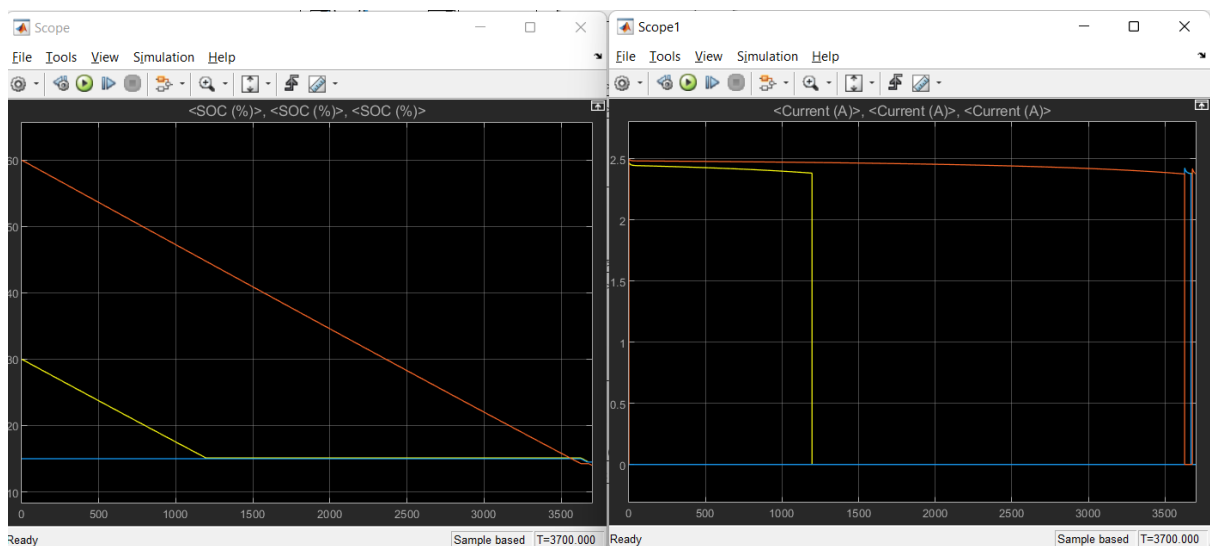
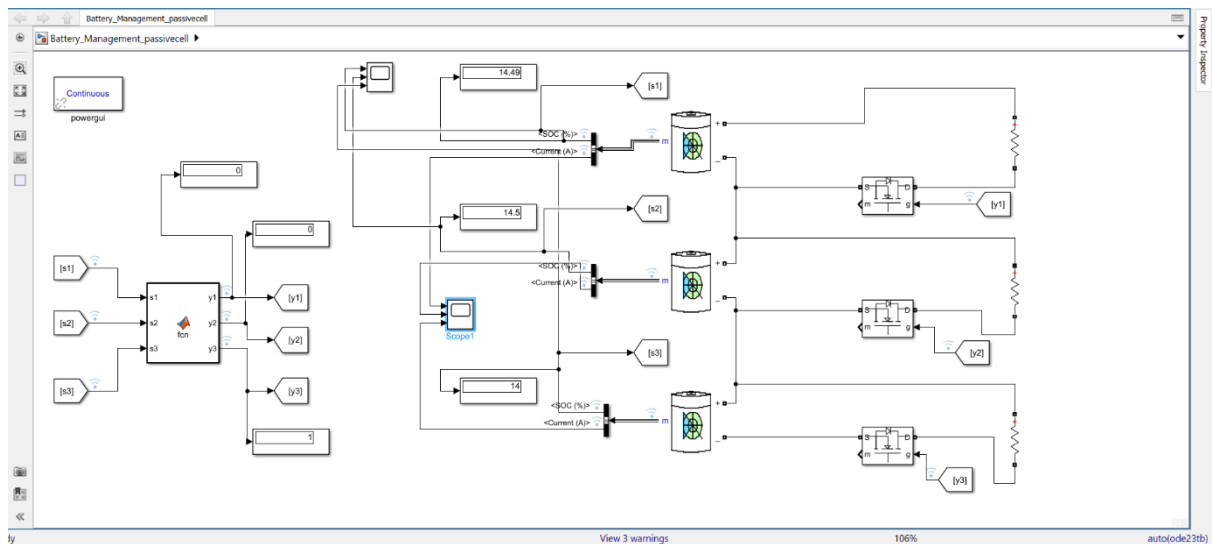
We have implemented following features

1. Adaptive Cruise Control System
2. Anti Lock Braking System
3. Automatic Transmission Control System
4. Door Locking system
5. Engine Braking System
6. Lane Assist System
7. Power Window

Design

This project was implemented using Matlab.





GitHub Link:

Link: https://github.com/sunilkora31/TeamTesla_Applied_MBD.git

Miniproject 6 – Wiper Control [Team]

Modules

1. C Programming
2. STM32

Requirements

4W's and 1'H

Who:

Everybody who knows to drive

What:

System is used for cleaning the windscreen

When:

A car wiper is a device which is used to remove droplets of rainwater from a windshield of a car

.Where:

At the time of car driving driver need a wiper to clean the windshield

How:

Ignition Key Position at ACC: The Red LED is ON, if the user button is pressed and held for 2 secs

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, The frequency changes on every alternate key press, 3 frequency levels with 1, 4 and 8 Hz

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

Ignition Key Position at Lock: The Red LED is OFF, if the user button is pressed and held for 2 secs

High Level Requirements

ID	Description
HLR_1	Receive signals from the user
HLR_2	All conditions should pass. One condition should pass at a time
HLR_3	Run a system diagnostic test sequence at ignition and determine if any errors are present in the system

Low Level Requirements

ID	Description
LLR_1	Put the input value
LLR_2	The system test will engage when the car
LLR_3	Compare the conditions

Design

LOW LEVEL DIAGRAM

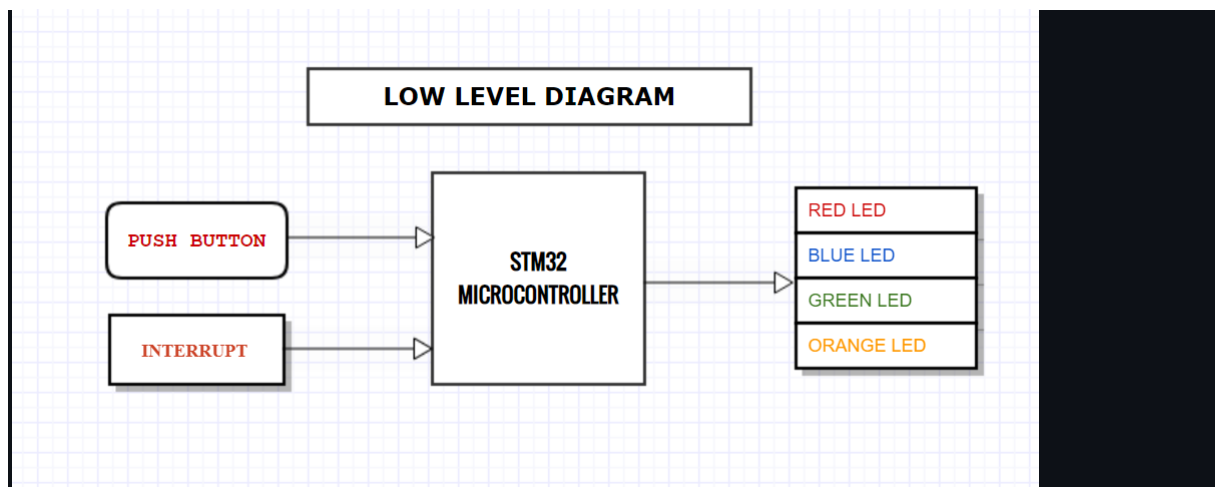
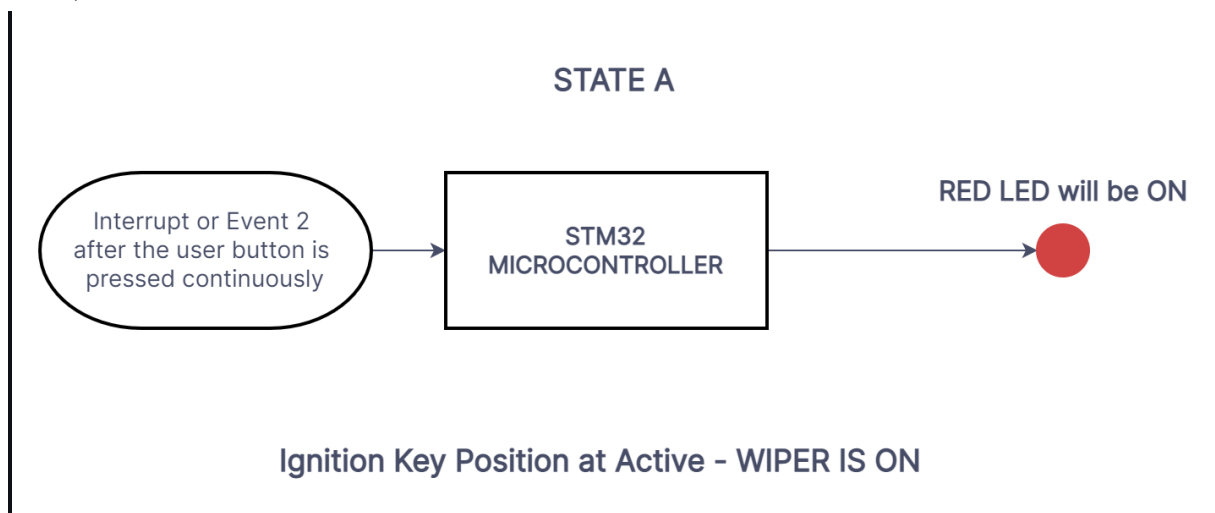
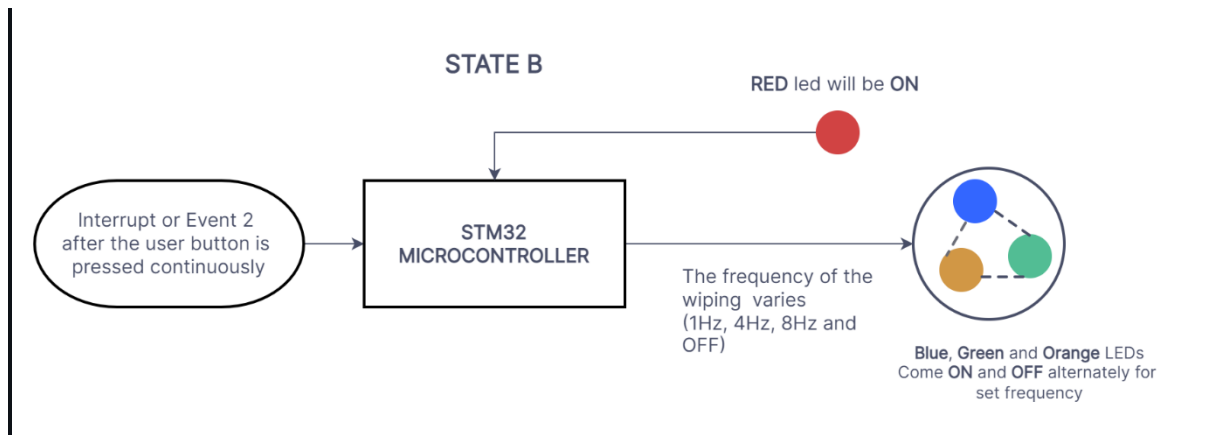


Figure 11 Structure Diagram

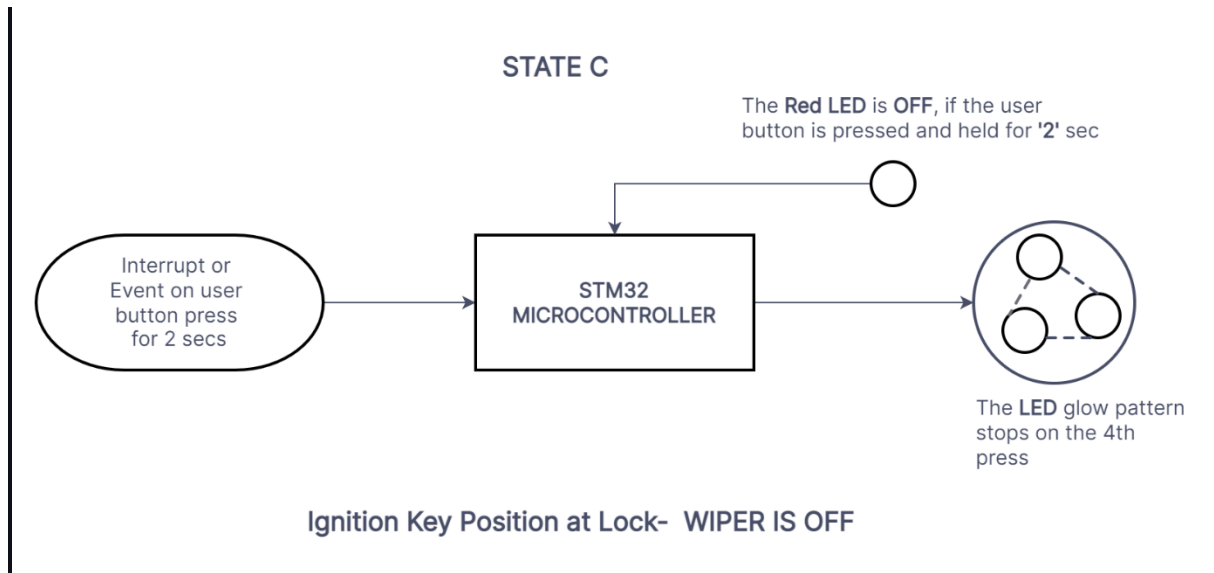
LEVEL 1



LEVEL 2



LEVEL 3



HIGH LEVEL DIAGRAM

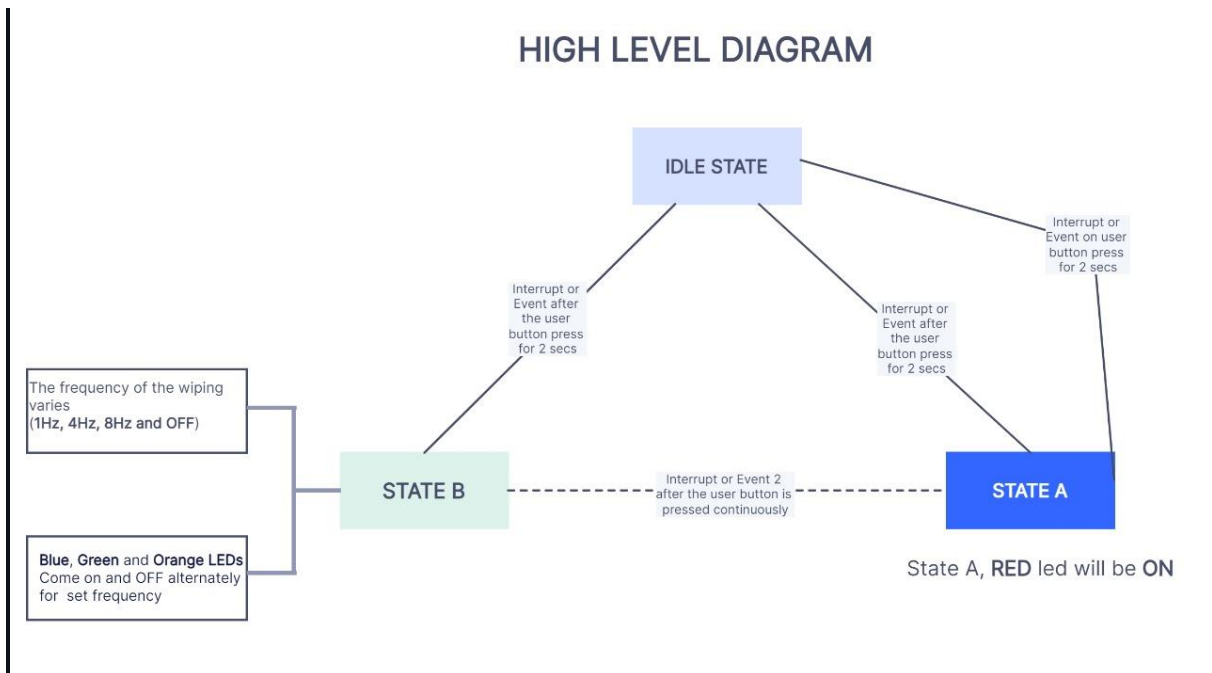


Figure 12 Behavior Diagram

Test Plan

High Level Test Plan

Test ID	Description	Exp I/P	Exp O/P	Actual I/P	Actual O/P
HLT1	checking all the functions are working correctly	Call the functions	All functions execute correctly	All functions are executed correctly	Requirement based
HLT2	Checking whether called functions are executed	Call a specific function	Call function execute	Called function is executed	Scenario based
HLT3	Check for features other than specified	Choosing other values	No output displayed	No output is displayed	Boundary based

Implementation and Summary

Git Link:

Link: <https://github.com/GENESIS-2022/MasteringMCU-Team6>

Individual Contribution and Highlights

1. Wiper System using C Programming
2. Source code management using GitHub

Role in Project Team

1. Programmer: Done Programming for Wiper System
2. Integrator: Integrated all the codes
3. Tester: Writing Testcases and testing the integrated code

Miniproject 7 – Project Volkswagen [Team]

Modules

1. Automotive Systems
2. Git

REQUIRMENTS

INTRODUCTION

Our integrated circuits and reference designs for automotive exterior rearview mirror modules help you accelerate design through accurate light sensors and LED lighting solutions with high precision in controlling and driving electrochromic mirrors and high integration for human interface (HMI) control. Modern rearview mirror modules require: Accurate, stable control of large capacitive load for electrochromic auto dimming. Precise sensing of ambient light to allow effective auto dimming.

FEATURES

BLIND SPOT DETECTION

Blind Spot Detection tracks traffic just behind you. The alert stays active until the car in the adjacent lane is in front of you, or at least directly alongside and you'd have to be blind not to see it. Blind Spot Detection uses ultrasonic or radar sensors on the side and rear of the car.

DEFOGGERS

Defoggers are used to defog the rear window, and to remove raindrops, dew and frost from the outside rear view mirrors. The operation time changes according to the ambient temperature and vehicle speed, It may be 10 minutes to 25 minutes.

COMPONENTS

DC MOTOR

Each rear-view mirror has two DC motors. One DC motor operates the up/down function while the other DC motor operates the left/right function. Both switches inside the power mirror switch are constantly connected to the vehicle's electrical ground circuit with the switch at rest.

INFRARED SENSOR

An infrared sensor (IR sensor) is a radiation-sensitive optoelectronic component with a spectral sensitivity in the infrared wavelength range 780 nm ... 50 µm. IR sensors are now widely used in motion detectors, which are used in building services to switch on lamps or in alarm system.

RADAR

Radar is a electromagnetic sensor used for detecting, locating, tracking, and recognizing objects of various kinds at considerable distances.

High Level Requirements

	TITLE	MODULES	DESCRIPTION
SYS_1	Requirments	DC MOTOR	The Side Rear View Mirror should Open when the Engine Started.
SYS_2	Requirments	DIGITAL CAMERA	The Main Camera Present in the Mirror gives the precise view of the mirror in an LCD Display inside the car when the weather outside is not good.
SYS_3	Requirments	DC MOTOR	When the Engine Turns off, the Mirror Should Close Automatically.
SYS_1	Requirments	DC MOTOR	The Side Rear View Mirror should Open when the Engine Started.

Low Level Requirements

	TITLE	MODULE	DESCRIPTION
SYS_1	Requirements	H-MOD	The Defogger System should connect with the Aircon System to generate the heat and make the mirror crystal clear for the driver.
SYS_2	Requirements	INFRARED SENSOR	The Sensor in the side mirror which can detect the chasing vehicle near it and gives a signal to the driver to be cautious.
SYS_3	Requirements	RADAR	To detect the Blind spots of the DRIVER, up to 6 cameras have been placed in the rear mirror to cover the entire blind spot.

Design

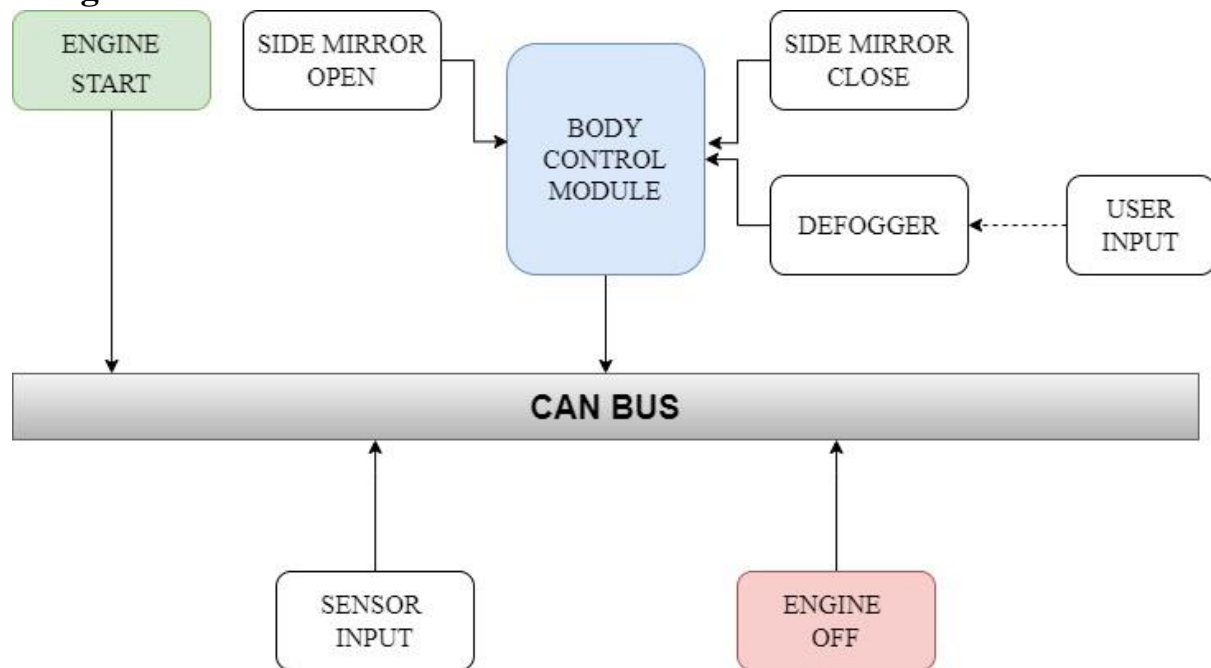


Figure 13 Structure Diagram

Implementation and Summary

Git Link:

Link: https://github.com/Vishnu-prasath/Automotive_Volkswagen.git

Individual Contribution and Highlights

1. Power Window Case Study
2. Source code management using GitHub

Role in Project Team

1. Designer: Done Designing for Project
2. Researcher: Done case study for Side Mirror

Miniproject 8 – EV Car [Team]

Modules

1. Matlab
2. Matlab Script

Requirements

Introduction

An electric car is a one powered by an electric motor rather than a traditional petrol/diesel engine. This electric motor is powered by rechargeable batteries that can be charged by common household electricity. There are two types of EV technology: hybrid and pure electric.

FULLY ELECTRIC CAR

The main parts in an electric car are a rechargeable battery, controller and electric motor. First, the battery is powered. Then the controller converts the current from DC-AC so that it can be used by the motor. The motor converts electrical energy to mechanical energy.

HYBRID CAR

The same technology exists in hybrid cars, alongside a small gasoline engine running a generator. This powers the car at cruising speed, and batteries provide extra power when accelerating. Batteries can recharge themselves when the car is decelerating or standing still. Hybrid technology means that your petrol goes much further, saving you money and reducing environmental impacts.

Implementation and Summary

Submission: Submitted in GEALearn

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.

Miniproject 9 – Defogger [Individual]

Modules

1. Autosar
2. Git

Requirements

S.NO	Function	Description
1	Engine	The Engine Should be ON to turn the mirror on
2	Display	Car Should be in Running mode to Enable display system
3.	Sensor	Sensor senses the Input and give the output
4.	Defogger	Clears the fog in the

Design

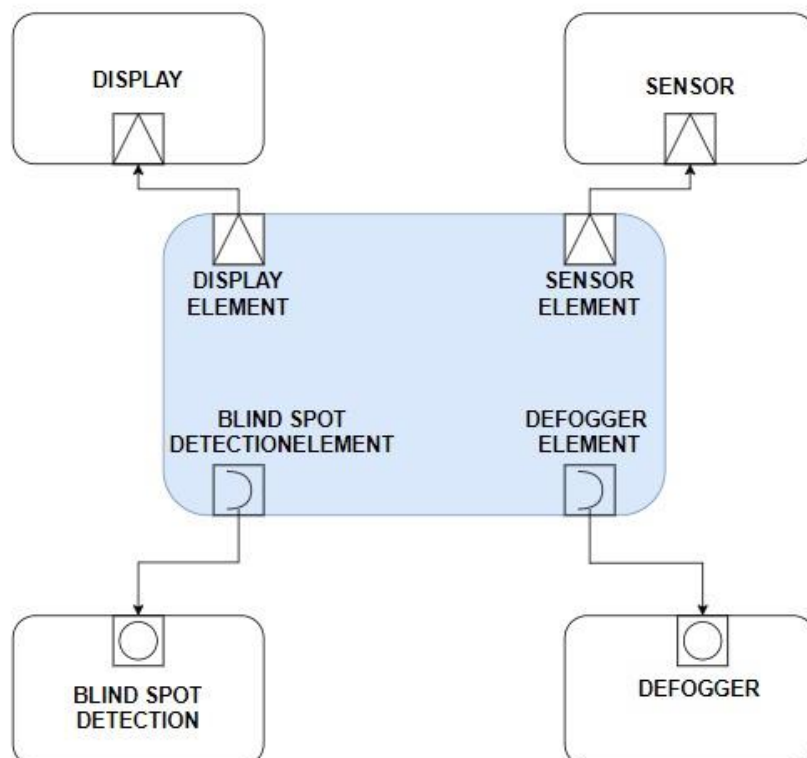


Figure 14 VFB Diagram

Implementation and Summary

Git Link:

Link: https://github.com/Vishnu-prasath/Defogger_40020535_DPS.git