**Details**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ver. Rel. No. | Release Date | Prepared By | Reviewed By | To Be Approved | Remarks/Revision Details |
| 1.0 | 16/02/2022 | Saripalli Sai Manvitha  40021066 |  |  |  |

Contents

[Miniproject – 1 Snake and ladder [Individual] 5](#_Toc95931104)

[Modules: 5](#_Toc95931105)

[Requirements 5](#_Toc95931106)

[High Level Requirements 5](#_Toc95931107)

[Low Level Requirements 6](#_Toc95931108)

Design…………………………………………………………………………………………7

[Test Plan 8](#_Toc95931110)

[High Level Test Plan 8](#_Toc95931111)

Low Level Test Plan………………………………………………...................................8

[Implementation and Summary 9](#_Toc95931113)

Summary…………………………………………………………………………………........9

[Git Inspector Summary…………………………………………………………………….. ..10](#_Toc95931117)

Miniproject – 2 Virtual Costume Advisor [Team]...................................................................10

Modules………………………………………………………………………………………10

Requirements…………………………………………………………………………………11

High Level Requirements………………………………………………………………..11

Low Level Requirements………………………………………………………………..11

Design………………………………………………………………………………………..12

Test Plan……………………………………………………………………………………...15

High Level Test Plan……………………………………………………………………15

Low Level Test Plan…………………………………………………………………….16

Implementation and Summary……………………………………………………………….17

Git Link………………………………………………………………………………………17

[Miniproject 3 – Wiper Control System[Team] 1](#_Toc95931143)7

[Modules: 1](#_Toc95931144)7

[Requirements 1](#_Toc95931145)8

[High Level Requirements 1](#_Toc95931146)8

[Low Level Requirements 1](#_Toc95931147)8

Design………………………………………………………………………………………..19

Test Plan……………………………………………………………………………………...21

[High Level Test Plan 2](#_Toc95931149)1

[Low Level Test Plan 22](#_Toc95931150)

Implementation and Summary……………………………………………………………….22

Git Link……………………………………………………………………………………....22

Summary……………………………………………………………………………………..22

Git Inspector Summary………………………………………………………………………23

[Miniproject 4 – Calendar Automation[Team] 24](#_Toc95931179)

[Modules: 24](#_Toc95931180)

[Requirements 24](#_Toc95931181)

High Level Requirements……………………………………………………………… 24

Low Level Requirements……………………………………………………………….25

[Implementation](#_Toc95931182) and Summary……………………………………………………………….26

Git Link………………………………………………………………………………………26

Git Dashboard………………………………………………………………………………. 26

Miniproject 5 - Jaguar Project[Team]……………………………………………………......27

Modules………………………………………………………………………………………27

Requirements…………………………………………………………………………………27

[Design………………………………………………………………………………………...28](#_Toc95931187)

Miniproject 6 - Golf Cart [Team]…………………………………………………………….29

Modules………………………………………………………………………………………29

Requirements…………………………………………………………………………………29

Range…………………………………………………………………………………………32

Report………………………………………………………………………………………...33

Implementation and Summary……………………………………………………………….33

Individual Contribution and Highlights……………………………………………………...33

Miniproject 7 - Wiper Control System [Individual]………………………………………….34

Modules………………………………………………………………………………………34

Requirements…………………………………………………………………………………34

Design………………………………………………………………………………………...35

Implementation and Summary……………………………………………………………….36

Git Link………………………………………………………………………………………36

Individual Contribution and Highlights……………………………………………………...36

## List of Figures

[Figure 1 Class Diagram 8](#_Toc95933213)

[Figure 2 Use case Diagram 9](#_Toc95933214)

[Figure 3 Flow Chart 11](#_Toc95933215)

[Figure 4 Git Inspector Summary 11](#_Toc95933216)

[Figure 5 Behavior Diagram Low Level 13](#_Toc95933217)

[Figure 6 Behavior Diagram High Level 14](#_Toc95933218)

[Figure 7 Structural Diagram High Level 14](#_Toc95933219)

[Figure 8 Structural Diagram Low Level 15](#_Toc95933220)

[Figure 9 Behavior Diagram High Level 16](#_Toc95933221)

[Figure 10 Use case Diagram 19](#_Toc95933222)

[Figure 11 Block Diagram 19](#_Toc95933223)

[Figure 12 Git Inspector Summary 20](#_Toc95933224)

[Figure 13 Git Dashboard 25](#_Toc95933225)

[Figure 14 Structure Diagram 25](#_Toc95933226)

Figure 15 Flow Chart...............................................................................................................36

# 

# 

# Mini project – 1: Snake and Ladder [Individual]

# Modules:

1. C Programming
2. Git

### Requirements

**4W's and 1 H**.

**Who:**

1. Anyone can play the game.

**When:**

1. Whenever the player wants to get relived from stress.

2. To increase fun.

**What:**

1. It is played by two or more players.

**Where:**

1. We can play manually or by using electronic devices.

2. Available in Visual studio code.

**How:**

1. Using C programming Language.

### 

### High Level Requirements

| **ID** | **Description** | **Status** |
| --- | --- | --- |
| HLR\_1 | OS : Windows 8.1 | Implemented |
| HLR\_2 | Architecture : x84,x64,ARM,ARM64 | Implemented |
| HLR\_3 | Language : C Programming | Implemented |

### 

### Low Level Requirements

| **ID** | **Description** | **Status** |
| --- | --- | --- |
| LLR\_1 | List of operations displayed | Implemented |
| LLR\_2 | Input from the user | Implemented |
| LLR\_3 | Exit the program | Implemented |

## 

## Design

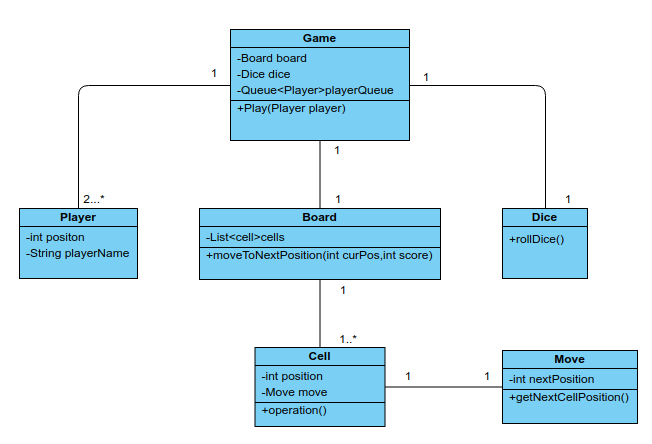
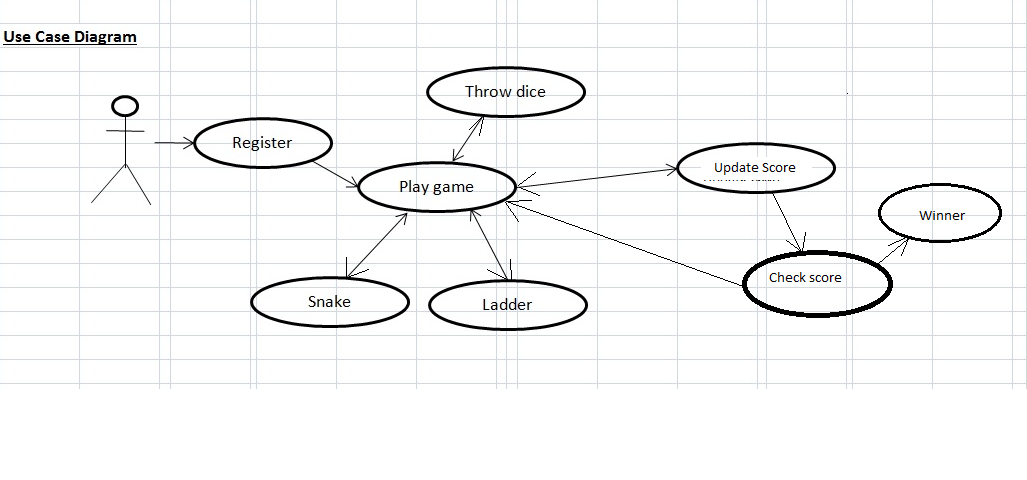


Figure 1 Class Diagram

 Figure 2 Use case Diagram

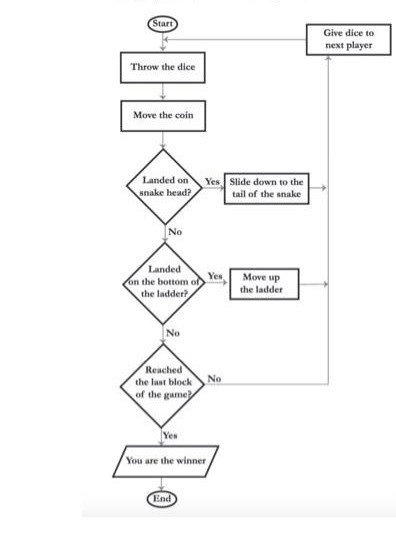


Figure 3 Flow Chart

## Test Plan

### High Level Test Plan

| **ID** | **Description** | **Expected I/P** | **Expected O/P** | **Actual O/P** | **Type Of Test** |
| --- | --- | --- | --- | --- | --- |
| H\_01 | Start the game | Choice | SUCCESS | SUCCESS | Requirement Based |
| H\_02 | Player1 | Choice | SUCCESS | SUCCESS | Requirement Based |
| H\_03 | Player2 | Choice | SUCCESS | SUCCESS | Requirement Based |
| H\_04 | Snakes | Choice | SUCCESS | SUCCESS | Requirement Based |
| H\_05 | Ladders | Choice | SUCCESS | SUCCESS | Requirement Based |
| H\_06 | Game Over | Choice | SUCCESS | SUCCESS | Requirement Based |
| H\_07 | Exit | Choice | SUCCESS | SUCCESS | Requirement Based |

### Low Level Test Plan

| **ID** | **Description** | **Expected I/P** | **Expected O/P** | **Actual O/P** | **Type Of Test** |
| --- | --- | --- | --- | --- | --- |
| L\_01 | Display Data | Success | Success | Success | Requirement Based |
| L\_02 | Sore Player1 | Winner | Winner | Runner | Requirement Based |
| L\_03 | Sore Player2 | Runner | Runner | Winner | Requirement Based |

## 

## Implementation and Summary

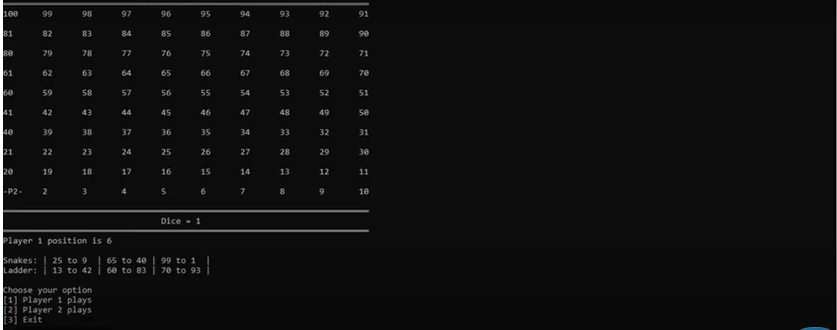
### Git Link:

Link: <https://github.com/Manvitha-123/M1_Game_Snake-and-Ladder.git>

**Summary**

### Git Inspector Summary





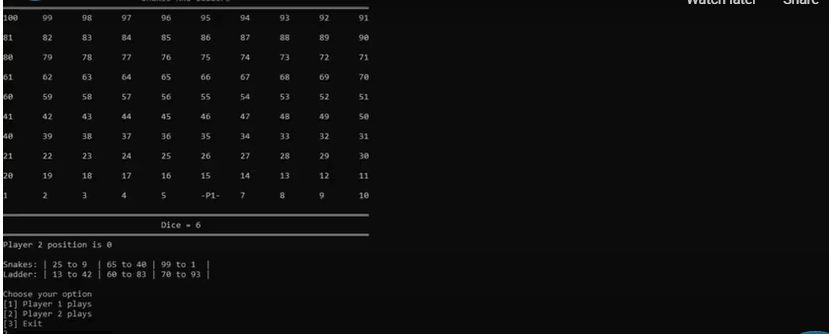


Figure 4 Git Inspector Summary

**Mini project 2 – Virtual Costume Advisor [Team]**

## Modules

1. C Programming
2. Git
3. Visual Studio

### Requirements

**4W's and 1 H**

**Where:**

1. In the Application/system which has this program.

**Who:**

1. People who want to look good by getting targeted outfit ideas for their body shape.

**What:**

1. Calculates the body shape and occasion they are addressing then suggests them the best suitable outfit for their body.

**When:**

1. Anytime they want to get themselves dressed well for particular occasions.

**How:**

1. By entering the measurements of the individuals bust size, waist size, high hip size, hip size.

### 

### High Level Requirements

| **ID** | **Description** | **Platform** |
| --- | --- | --- |
| HLR\_1 | Getting the measurements from the user | Application |
| HLR\_2 | Calculating the body type | Vs code |
| HLR\_3 | Getting the choice of outfit type from the user | Application |
| HLR\_4 | Getting the choice of listed costume from the user | Application |

### Low Level Requirements

| **ID** | **Description** | **Platform** |
| --- | --- | --- |
| LLR\_1 | The measurements should be properly taken and entered correctly by the user | Application |
| LLR\_2 | Coding formula to calculate body type should be accurate | Vs code |
| LLR\_3 | The Choice of outfit type should be properly Chosen and entered correctly by the user | Application |
| LLR\_4 | The Choice of costume should be properly taken and given correctly by the user | Application |

## Design

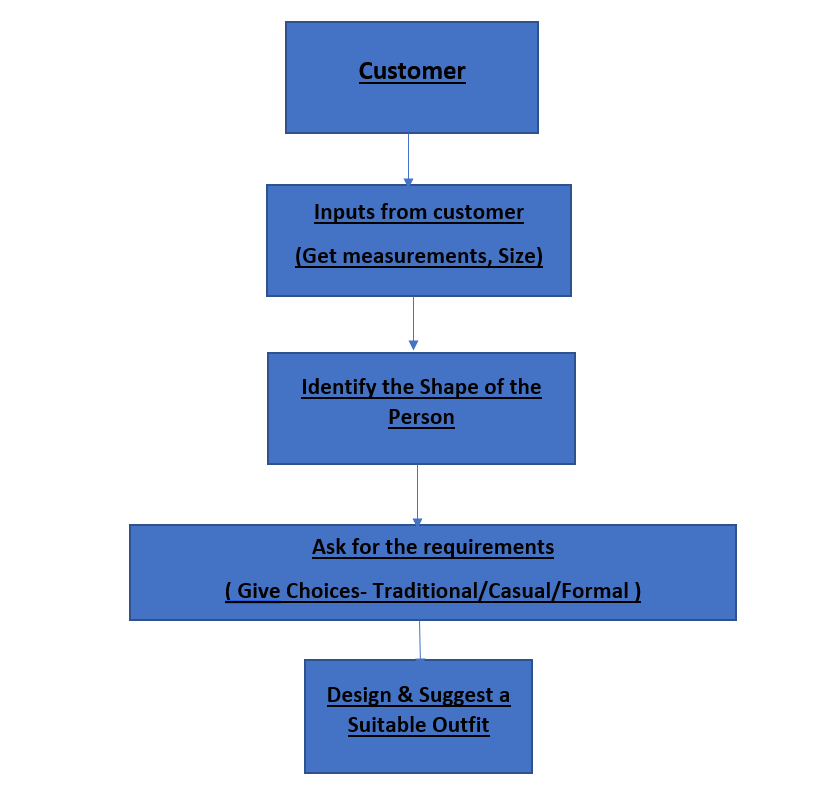


Figure 5 Behaviour Diagram Low Level

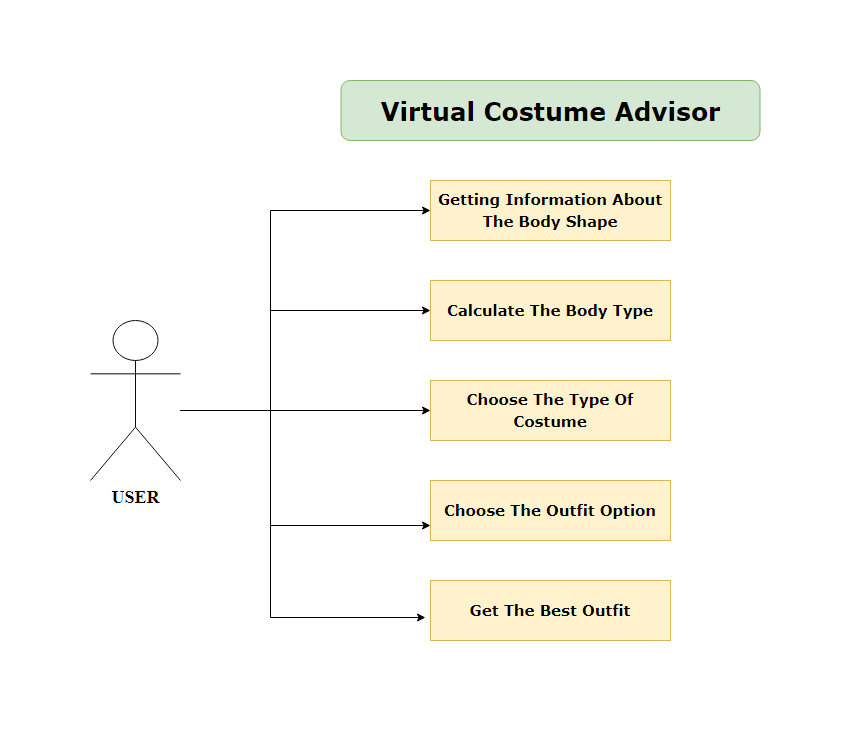


Figure 6 Behaviour Diagram High Level

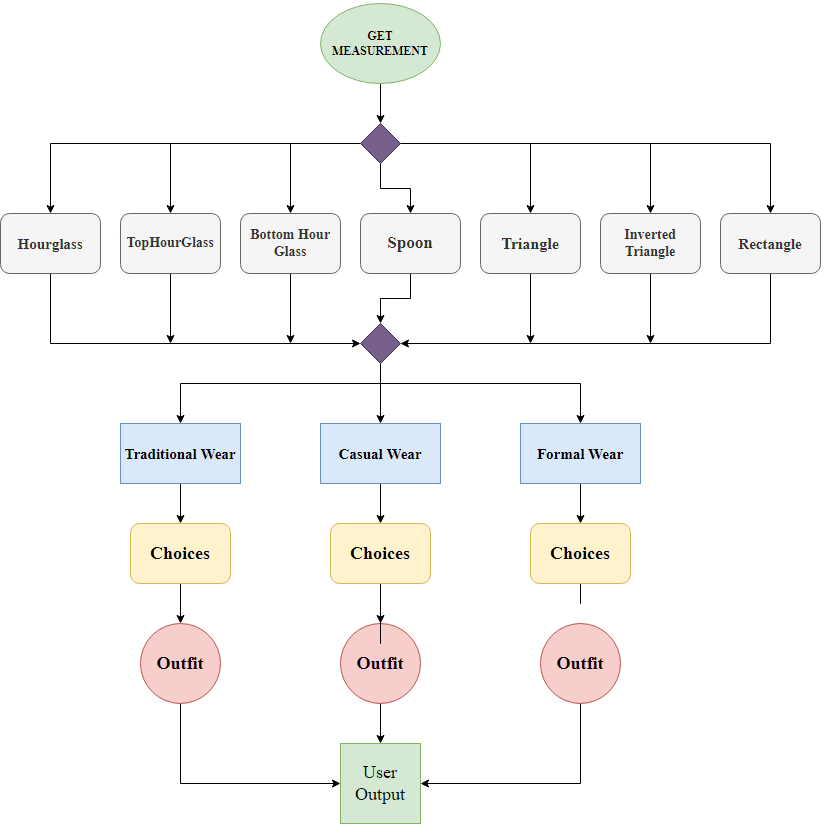


Figure 7 Structural Diagram High Level

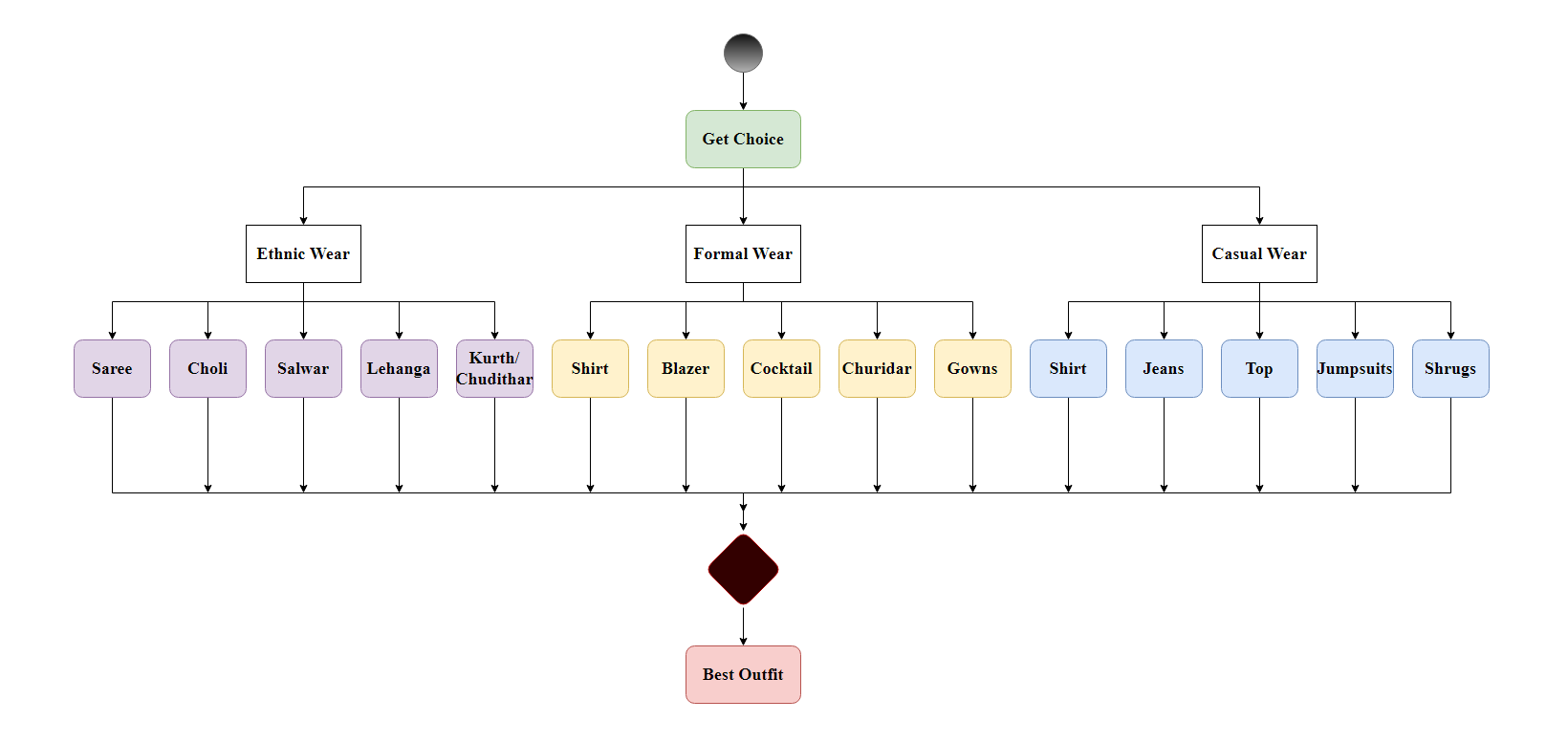


Figure 8 Structural Diagram Low Level

## 

## Test Plan

### High Level Test Plan

| **ID** | **Description** | **Expected I/P** | **Expected O/P** | **Type Of Test** |
| --- | --- | --- | --- | --- |
| H\_01 | Check and verify all the measurement values are entered | Enter the proper measurements according to the description | Body shape is displayed | Requirement Based |
| H\_02 | Check and verify all the measurements entered are in centimetre | Enter the measurements in centimetre | Body shape is displayed | Requirement Based |
| H\_03 | Check whether all entered measurements is displayed correctly | Enter the measurements | Body shape is displayed | Scenario Based |
| H\_04 | Check whether choices of outfit types are displayed correctly | Enter the choice | Another set of choices displayed | Scenario Based |
| H\_05 | Check whether choices of costumes are displayed correctly | Enter the choice | The output of desired choice displayed | Scenario Based |
| H\_06 | Check whether entered choice of outfit type is correct | Enter the Choice number | Display Choices | Boundary Based |
| H\_07 | Check whether entered choice of costumes is correct | Enter the Choice number | Display Choices | Boundary Based |

### Low Level Test Plan

| **ID** | **Description** | **Expected I/P** | **Expected O/P** | **Type Of Test** |
| --- | --- | --- | --- | --- |
| L\_01 | To check if the measurements give the proper body shape | Measurements | Body Type | Requirement Based |
| L\_02 | To check if the choices give the proper body shape | Choice | The required outfit | Requirement Based |
| L\_03 | To check if the calculation is properly done to give proper output | Body Type | Scenario Based | Scenario Based |
| L\_04 | To check if all of the four required measurements are entered | Bust, Waist, High hip, Hip sizes | Display Body shape | Scenario Based |
| L\_05 | To check if required choice of outfit type is entered | Choices | Display the best outfits | Scenario Based |
| L\_06 | To check if required choice of costume is entered | Choices | Display the best costumes of the desired outfit | Scenario based |

## Implementation and Summary

### Git Link:

### Link: <https://github.com/GENESIS2021Q1/Applied_SDLC-Dec_Team_50.git>

### Mini project 3 – Wiper Control System [Team]

# Modules

1. Git
2. Microcontrollers

### Requirements

**4W's and 1 H's**

**Why:**

1. This project helps the users to achieve the clear path when there is a change of weather.

**When:**

1. When there is a change in the weather, the wipers work automatically.

**What:**

1. Rain sensor turns off irrigation system within minutes when rain water falls. Here it consists of water absorbing expansion disks which swell in presence of rain drops. After some amount of rain falls, electrical switch gets depressed.
2. As a result, regular cycle of irrigation is bypassed and automatic irrigation system is switched off. Disks shrink when dry condition is reached and reverse action takes place.

**How:**

1. So here we propose an automatic wiper system that automatically switches 'ON' on detecting rain and stops when rain stops and dust sensor is also added, when it detects dust it will wipe. Our project brings forward this system to automate the wiper system having no need for manual intervention. For this purpose we use rain sensor and dust sensor along with microcontroller to drive the wiper motor. Our system uses rain sensor to detect rain and dust sensor to detect the dust, this signal is then processed by microcontroller to take the desired action.
2. The rain sensor works on the principle of using water for completing its circuit, so when rain falls on it it’s circuit gets completed and sends out a signal to the microcontroller, similarly when dust is appearing in the car glass surface it will detect, and gives signal to the microcontroller. The microcontroller now processes this data and controls the motor. This system is equally useful version

### 

### High Level Requirements

| **ID** | **Description** |
| --- | --- |
| HLR\_1 | These systems detect droplets of rain on the windshield and automatically turn on and adjust the wiper system in accordance to the level of precipitation. |
| HLR\_2 | A windscreen wiper or windshield wiper is a device used to remove rain, snow, ice and debris from a windscreen or windshield. |
| HLR\_3 | Quality and reliability wiper systems meet the highest technical requirements and are the basis for vehicles with sophisticated features. |
| HLR\_4 | Almost all motor vehicle, including trains, aircraft and watercraft, are equipped with such wipers, which are usually an essential requirement. |
| HLR\_5 | Our project brings forward this system to automate the wiper system having no need for manual intervention. |

### Low Level Requirements

| **ID** | **Description** |
| --- | --- |
| LLR\_1 | A new mechatronic reversing system can now be used to clean the windshield with two wiper arms, whereby one wiper arm is powered directly and the other indirectly using a connection link. |
| LLR\_2 | Wiper motor is automatically ON during the time of rainfall. |
| LLR\_3 | Existing system manually used control stalk to activate wiper and the process of pulling up wiper is difficult to be handled. |
| LLR\_4 | Lower level parsing. Under the hood, the Requirement class does most of the heavy lifting. Class requirements. |
| LLR\_5 | These systems detect droplets of rain on the windshield and automatically turn on and adjust the wiper system. |

## 

## Design

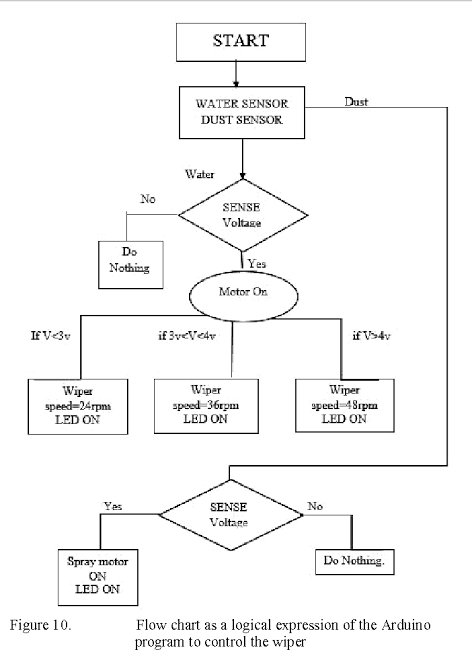


Figure 9 Behaviour Diagram High Level

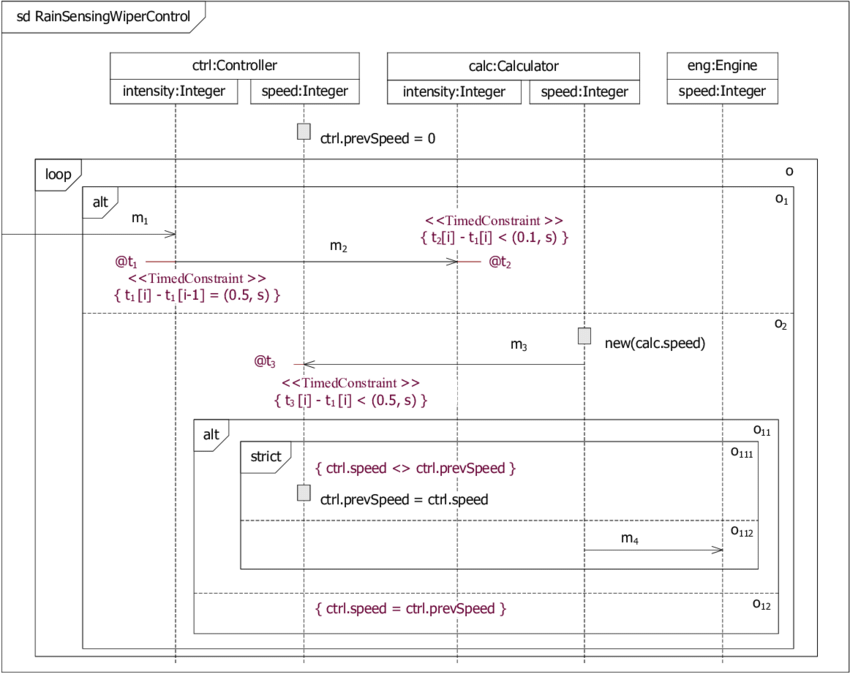


Figure 10 Use case Diagram

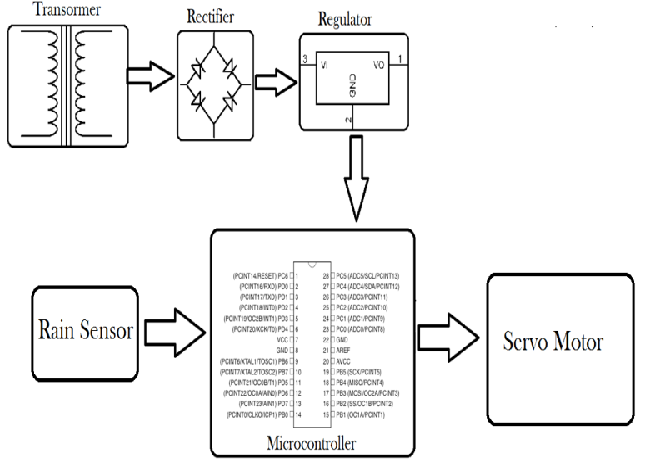


Figure 11 Block Diagram

## Test Plan

### High Level Test Plan

| **ID** | **Description** | **Expected I/P** | **Expected O/P** | **Actual O/P** | **Status** |
| --- | --- | --- | --- | --- | --- |
| H\_01 | To monitor surroundings temperature | DHT 11 Temperature sensor is used to detect temperature | According to Temperature values wiper is operated | According to Temperature values automatic wiper is operated | Passed |
| H\_02 | To monitor Rain condition | Rain Sensor is used to detect the rain condition | According to Rain condition wiper is operated | According to Rain condition wiper is operated | Passed |
| H\_03 | To monitor temperature and Rain conditions | Sensor values are detected | 4 LED lights are operated accordingly | 4 LED lights are operated accordingly | Passed |

### Low Level Test Plan

| **ID** | **Description** | **Expected I/P** | **Expected O/P** | **Actual O/P** | **Status** |
| --- | --- | --- | --- | --- | --- |
| L\_01 | To get displayed in LCD | LCD is connected to STM32 Microcontroller | Displays the wiper movement conditions | Displays the wiper movement conditions | Passed |
| L\_02 | To make wiper movements | Servomotor is connected to STM32 Microcontroller | Wiper is moved accordingly | Wiper is moved accordingly | Passed |

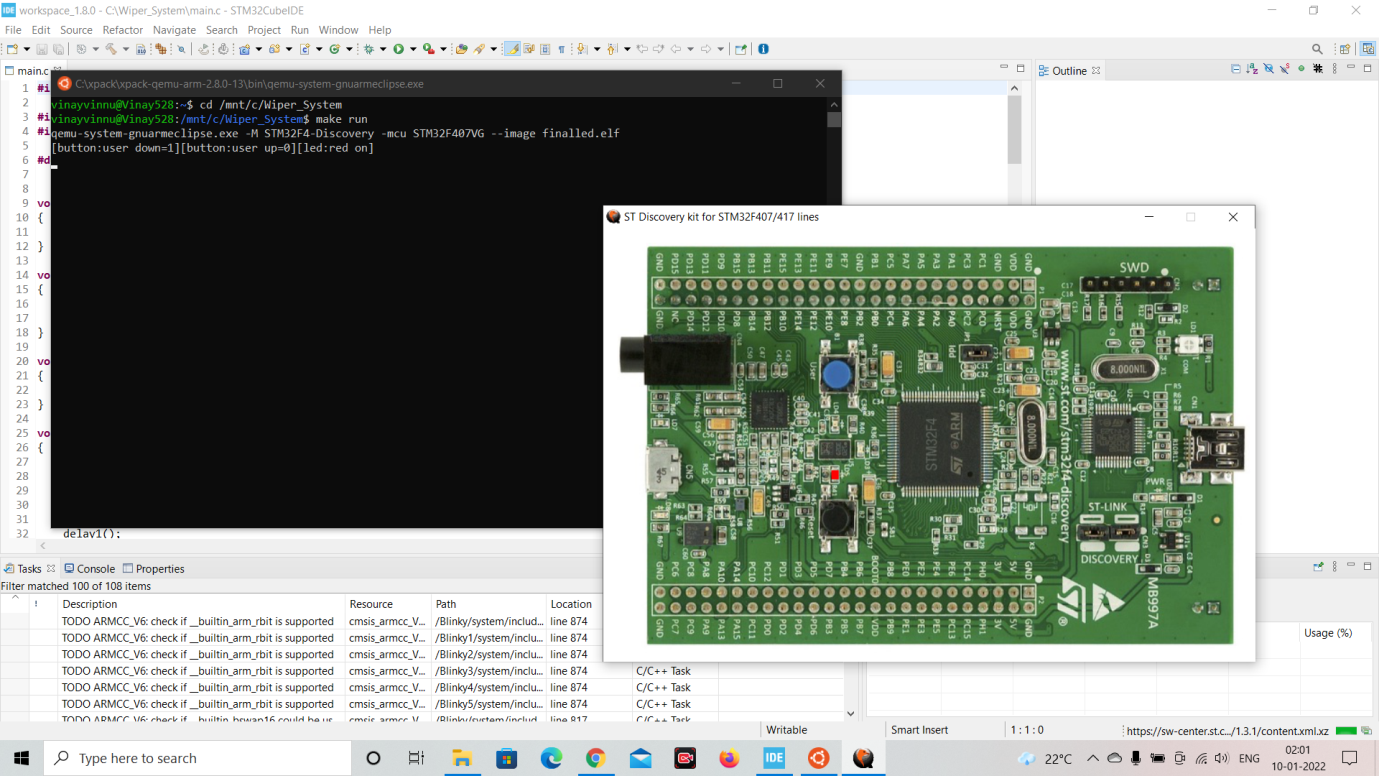
## Implementation and Summary

### Git Link:

Link: <https://github.com/GENESIS-2022/MasteringMCU-Team66.git>

**Summary**

### Git Inspector Summary



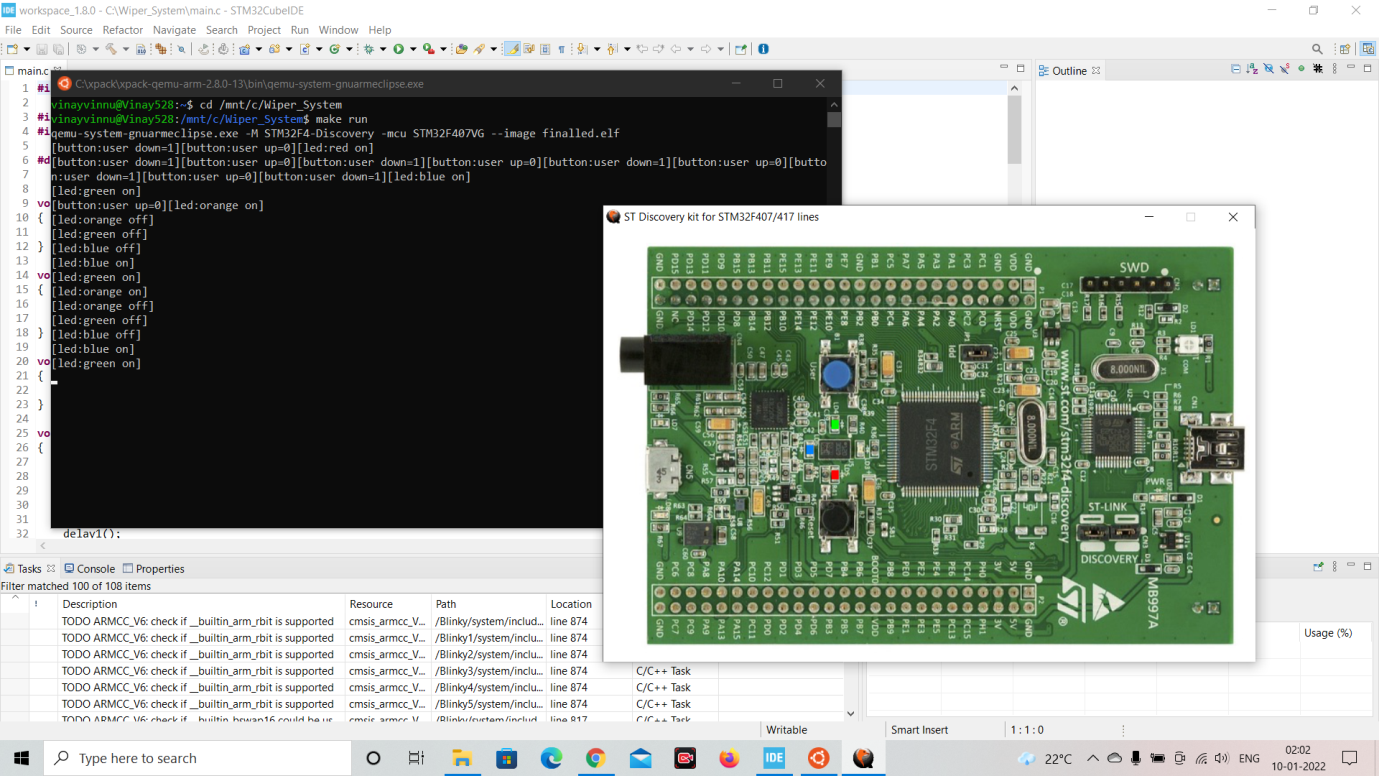


Figure 12 Git Inspector Summary

**Mini project 4 – Calendar Automation [Team]**

**Modules**

1. Git

2. Python

### 

### Requirements

**Software Requirements**

Py Drive==1.3.1

Matplotlib==3.4.3

Openpyxl==3.0.7

pandas==1.3.0

streamlit==0.85.1

xlsx2html==0.4.0

numpy==1.21.1

### 

### High Level Requirements

| **ID** | **Feature** | **Mat lab v0 Status** | **Python v0 Status** |
| --- | --- | --- | --- |
| HR01 | GUI | Implemented | Implemented |
| HR02 | Master Calendar | Implemented | Implemented |
| HR03 | Faculty calendar | 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 |

### Low Level Requirements

| **ID** | **Feature** | **High Level ID** | **Mat lab 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 (Colour Codes/Numbers) | HR02 | Implemented | Implemented |
| LR09 | Master Calendar: Appending | HR02 | Implemented | Not Available |
| LR010 | Master Calendar: Course code correction | HR02 | Implemented | Not Available |
| LR011 | Master Calendar: Course title correction | HR02 | Not Available | Not Available |
| LR012 | Master Calendar: display the dates that were not analysed | HR02 | Implemented | Not Available |
| LR013 | Faculty Calendar: display Month wise | HR03 | Implemented | Implemented |
| LR014 | Faculty Calendar: display Initiative wise | HR03 | Implemented | Not Available |
| LR015 | Faculty Calendar: Appending | HR03 | Implemented | Not Available |

## 

## Implementation and Summary

### Git Link:

Link: <https://github.com/Ramki17/Calendar_Automation-Genesis21_Team49.git>

### 

### Git Dashboard

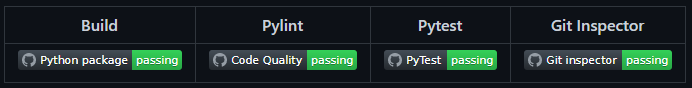


Figure 13 Git Dashboard

# 

# 

# Mini project 5 – Jaguar Project [Team]

## Modules

1. Git

### Requirements

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

1. Parking System
2. Headlight Control
3. Side view Mirror Control
4. Wiper Control System

| **S.NO** | **Function** | **Description** |
| --- | --- | --- |
| 1 | Engine | The Engine Should be ON to Use the Parking System |
| 2 | Reverse Gear | Car Should be in Reverse Gear to Enable Parking System |
| 3 | Assistive Mode | Assistive Mode Can Be Enabled or Disabled |
| 4 | Assistive Mode ON | If Assistive Mode is ON the Sensor Assists |
| 5 | Assistive Mode OFF | If Assistive Mode is OFF the Sensor Will not Assist |
| 6 | Sound Frequency | Sound Frequency Varies Based On The Object Distance |

## Design

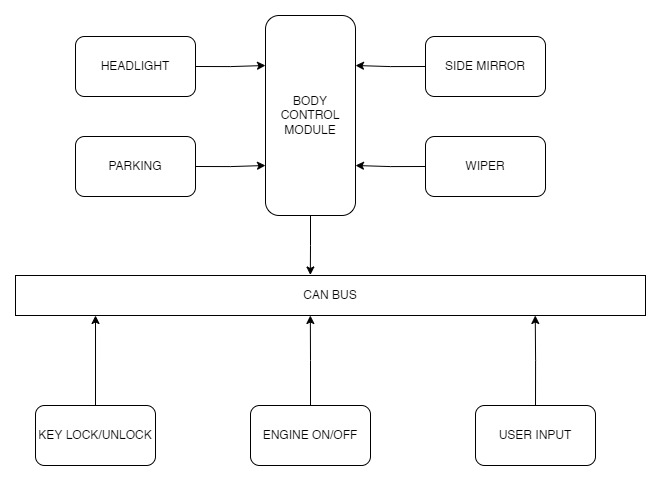


Figure 14 Structure Diagram

**Mini project 6 – Golf Cart [Team]**

## Modules

1. Mat lab
2. Applied Vehicle dynamics

### Requirements

**EZGO marathon Vs YAMAHA UMAX rally 2+2:**

**Motor Specifications:**

|  |  |  |
| --- | --- | --- |
| **Component** | **EZGO marathon** | **Y- UMAX rally 2+2** |
| Engine Type | 350cc twin cylinder unit  18 cubic inches | 402cc low-emission single cylinder 60 degree incline OHV |
| Fuel Tank Capacity | Twin cylinder unit | 5.2US GAL (20 LITERS) |
| Top speed | 12-14 Nm | 15 mph (24.1 km/hr) |
| Minimum Turning Radius | 4.24m | 3.98m |
| Maximum Forward Speed | 12mph (19.3 km/h) | 15mph (24.1 km/h) |

**Battery Specifications:**

|  |  |  |
| --- | --- | --- |
| **Component** | **EZGO marathon** | **Y- UMAX rally 2+2** |
| Battery Type | Works on gas cylinder | Lithium-ion |
| Range | 25-30 miles | 35 miles |
| Battery Charging Time | Works on gas cylinder | 4 Hours |
| Battery Capacity | Works on gas cylinder | 2 kWh |
| No of Cells | Works on gas cylinder | 20,500 |

**Wheel Specifications:**

|  |  |  |
| --- | --- | --- |
| **Wheel Type** | **EZGO marathon** | **Y- UMAX rally 2+2** |
| Front Wheel Size | 10 inches | 12 inches |
| Rear Wheel Size | 11 inches | 12 inches |
| Front Tyre Size | 22\*9-11\*10.5 - ply monitor k272 | 23\*10.5-12\*4- ply monitor k389 |







**Range**

Vehicle Model Buggy

Colour Wide range of Colour options

Usage/Application Eco-friendly drive within enclosed campuses.

Seating Capacity 4-12 persons

Running Distance 40 Km/Charge

Maximum Speed 11KM/Hour

Brand Maini

Material Powder coated tubular steel Chassis,

Body frame dent proof ABS

Number Of Battery 6 no, 8 Volt each, 150AH

Voltage 48V

Motor Rated 48 Volt AC Motor

Power 4 Kw Continuous

Speed 11KM/Hour

Seater 2-14 seater

Capacity Kg 200-1100 KG

**Battery**

Brand Trojan

Capacity @20Hr - 170Ah

Model Name/Number Trojan Motive T-875 with Bayonet Cap

Voltage 8V

Battery Type Deep-Cycle Flooded/Wet Lead-Acid Battery

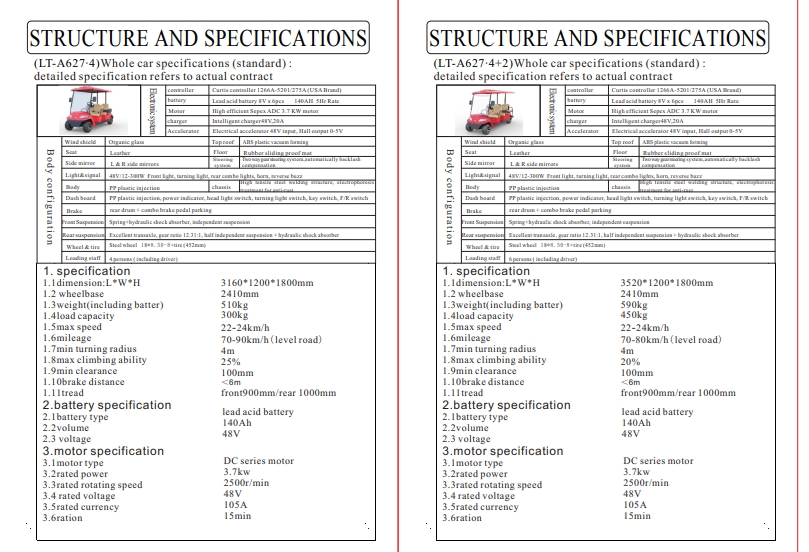
Dimensions 10.27 x 7.10 x 11.14 Inches

Weight 29 Kg

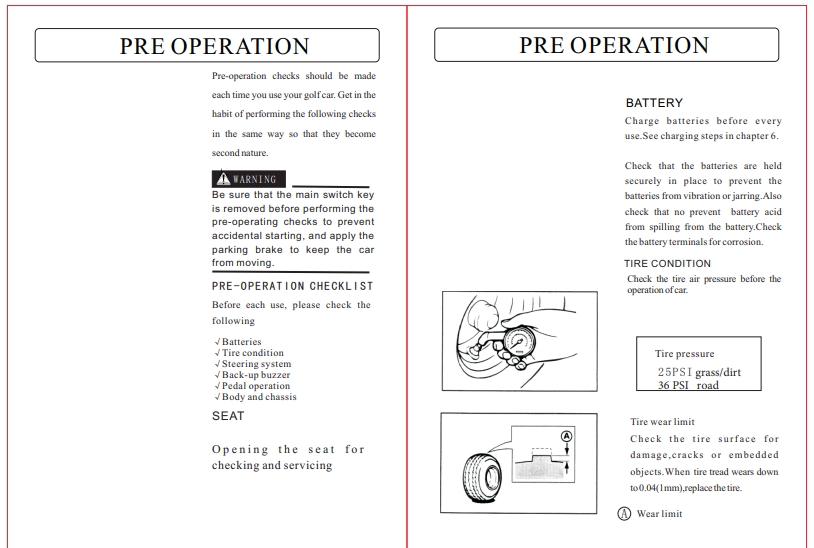
Material Polypropylene

Application/Usage Golf cart, Low Speed Electric Vehicle

**Report**



## 



## 

## Implementation and Summary

Submission: Submitted in GEA Learn

### Individual Contribution and highlights

### 1. Done in Matlab Script

# Mini project 7 – Wiper Control System [Individual]

## Modules

1. Autosar
2. Git

### Requirements

| **Id** | **Requirements** | **Description** |
| --- | --- | --- |
| HLR1 | Wiper Motor | The wiper motor is one part of the wiper system that is useful for moving the wiper automatically. The wiper motor that is often used recently is the ferrite magnet type because this type of wiper has a relatively small size and has lightweight, so it is very suitable to be attached to any type of car. |
| HLR2 | Wiper Link | The wiper link is often referred to as the wiper lever which is useful for changing the rotation motion of the wiper motor into a movement that goes back and forth from the left and right. |
| HLR3 | Wiper Arm | The wiper arm usually consists of a head that is used to tie the wiper arm to the wiper shaft. |
| HLR4 | Wiper Blade | The wiper blade usually consists of a rubber which is useful for cleaning the surface of the car glass. |

## 

## Design

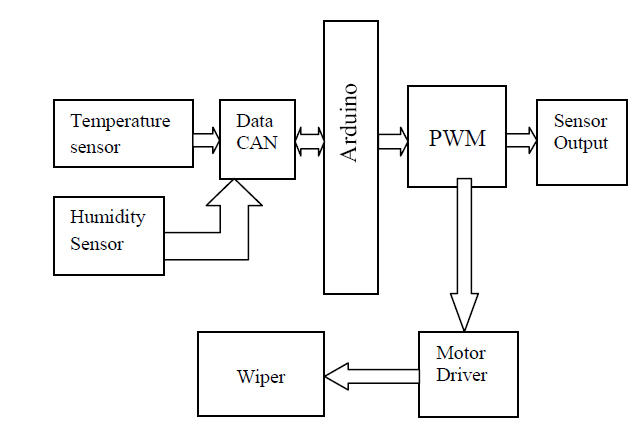


Figure 15 Flow Chart

## Implementation and Summary

### Git Link:

Link: <https://github.com/Manvitha-123/Autosar_Manvitha_40021066_TRN.git>

### Individual Contribution and highlights

1. Done in Autosar artop