Details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ver. Rel. No. | Release Date | Prepared By | Reviewed By | To Be Approved | Remarks/Revision Details |
| 1.0 | 16/02/2022 | Triveni B  40021070 |  |  |  |

Contents

[Miniproject – 1: Pacman Game [Individual] 4](#_Toc95931104)

[Modules:](#_Toc95931105) 4

[Requirements 5](#_Toc95931106)

[High Level Requirements](#_Toc95931107) 5

[Low Level Requirements](#_Toc95931108) 5

[Design](#_Toc95931109) 6

[Test Plan](#_Toc95931110) 7

[High Level Test Plan 8](#_Toc95931111)

[Low Level Test Plan](#_Toc95931112) 9

Implementation and Summery……………………………………………………………………...10

Summary……………………………………………………………………………………..10

Git Inspector Summary………………………………………………………………………10

[Miniproject 2 – Virtual Costume Adivisor [Team] 11](#_Toc95931118)

[Modules 11](#_Toc95931119)

[Requirements 11](#_Toc95931120)

[High Level Requirements 11](#_Toc95931121)

[Low Level Requirements 12](#_Toc95931122)

[Design 12](#_Toc95931123)

[Test Plan 14](#_Toc95931124)

[High Level Test Plan 1](#_Toc95931125)5

[Low Level Test Plan 15](#_Toc95931126)

[Implementation and Summary 1](#_Toc95931127)6

[Git Link: 1](#_Toc95931128)7

[Miniproject 3 –Wiper Control System [Team] 1](#_Toc95931130)8

[Modules 1](#_Toc95931131)8

[Requirements 1](#_Toc95931132)8

[High Level Requirements 1](#_Toc95931133)9

[Low Level Requirements 19](#_Toc95931134)

[Design](#_Toc95931135) 20

[Summary 2](#_Toc95931139)1

[Git Link: 21](#_Toc95931140)

[Miniproject 4 – Calendar Automation[Team] 2](#_Toc95931143)2

[Modules 22](#_Toc95931144)

[Requirements 22](#_Toc95931145)

[High Level Requirements 22](#_Toc95931146)

[Low Level Requirements 23](#_Toc95931147)

[Summary 23](#_Toc95931151)

[Git Link: 23](#_Toc95931152)

[Miniproject 5 – Jaguar Project [Team] 24](#_Toc95931156)

[Modules 24](#_Toc95931157)

[Requirements 25](#_Toc95931158)

[Design 25](#_Toc95931159)

[Miniproject 6 – EV Bike[Team] 26](#_Toc95931160)

[Modules 26](#_Toc95931161)

[Requirements 27](#_Toc95931162)

[High Level Requirements 27](#_Toc95931163)

[Low Level Requirements 28](#_Toc95931164)

[Design 28](#_Toc95931165)

[Git Link: 29](#_Toc95931170)

[Miniproject 7 – Seat belt control [Individual] 30](#_Toc95931172)

[Modules 30](#_Toc95931173)

[Requirements 30](#_Toc95931174)

[Design 31](#_Toc95931175)

[Implementation and Summary 31](#_Toc95931176)

[Git Link: 3](#_Toc95931177)**1**

## List of Figures

[Figure 1 Behavior Diagram 6](#_Toc95933213)

[Figure 2 Structure Diagram 7-8](#_Toc95933214)

[Figure 3 Git Dashboard 10](#_Toc95933215)

[Figure 4 Git Inspector Summary 10](#_Toc95933216)

[Figure 5 Behavior Diagram……………………………………………………………….13](#_Toc95933217)-14

[Figure 6 Structure Diagram 15-16](#_Toc95933218)

[Figure 7 Behavior Diagram 20](#_Toc95933219)

[Figure 8 Structural Diagram 21](#_Toc95933220)

[Figure 9 Git Dashboard 23](#_Toc95933221)

[Figure 10 Structure Diagram 25](#_Toc95933222)

[Figure 11 Simulation Diagram 29](#_Toc95933223)

[Figure 12 Power Window Diagram 31](#_Toc95933224)

# 

# Miniproject – 1: Pacman Game [Individual]

## Modules:

1. C Programming
2. Git

### Requirements

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

**Why:**

1. This game mainly used for Entertainments and decrease stress.
2. This game for open source any one can uses it anywhere like iPhones, smart phones and also tabs, laptops.
3. This game easy to play any one.

**Where:**

1. iPhone
2. Smartphones
3. Tabs
4. Laptops
5. Android Phones

**What:**

1. Pac-Man is an action maze chase video game.
2. The player controls the eponymous characters through an enclosed maze.
3. The objective of the game is to eat all of the dots placed in the maze while avoiding four colored ghosts.

**When:**

1. If you download your personal phones and laptops, it was easy to use any one freely.

**How:**

1. This open-source game. And then it’s given more entertainment

### High Level Requirements

| **ID** | **Description** | **Status** |
| --- | --- | --- |
| HLR\_1 | OS Windows 10 | 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

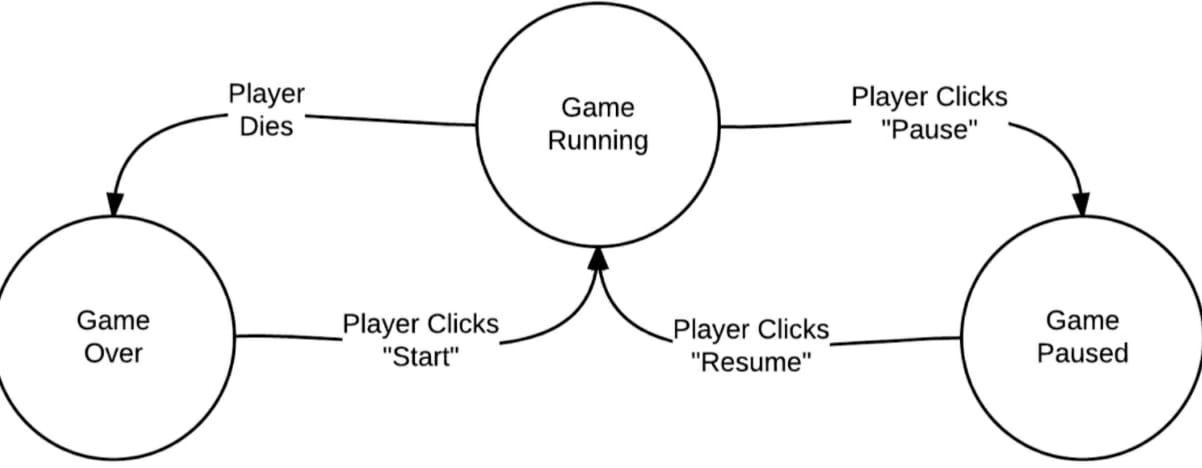


Figure 1- Behaviour Diagram1

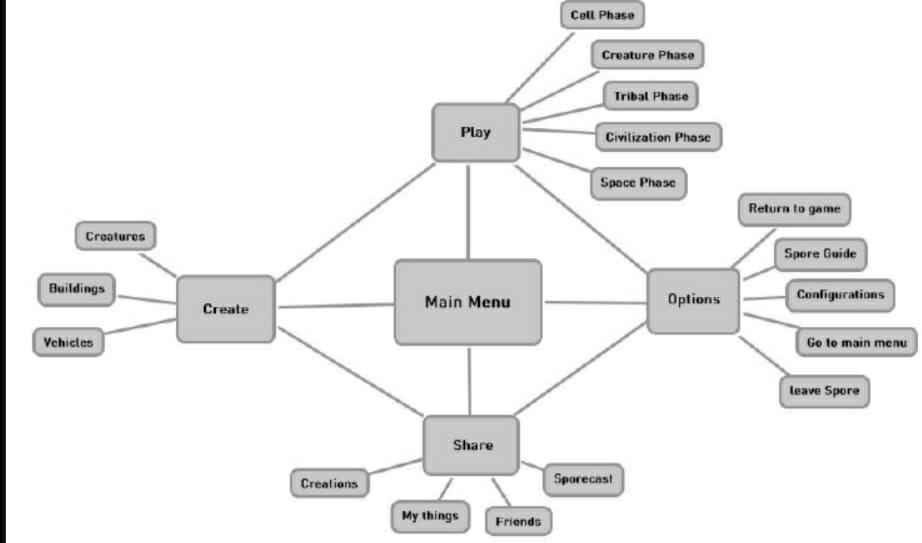


Figure 1- Behaviour Diagram2

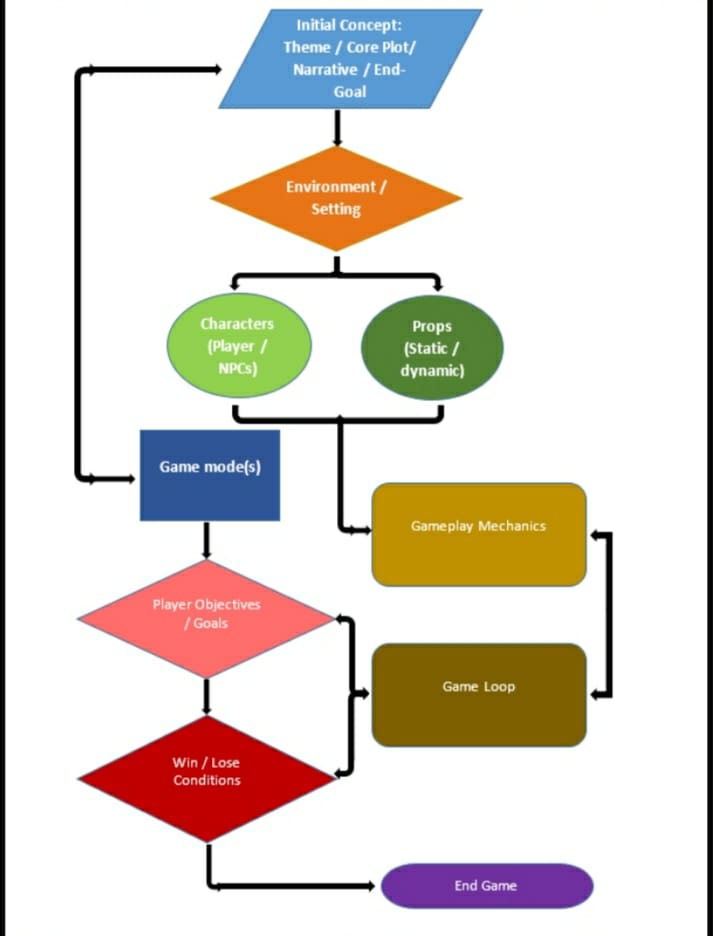


Figure 2-Structure Diagram1

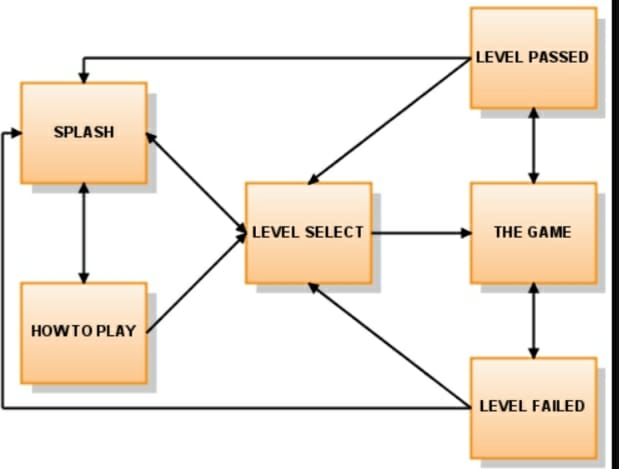


Figure 2-Structure Diagram2

## Test Plan

### High Level Test Plan

| **ID** | **Description** | **Expected I/P** | **Expected O/P** | **Actual O/P** | **Type Of Test** |
| --- | --- | --- | --- | --- | --- |
| HLTP\_1 | Check and verify moving the game or not | Enter the selected placeless moving | displayed moving place | displayed moving place | Requirement Based |
| HLTP\_2 | Check whether game is displayed correctly or not | Enter moving place | game is displayed | game is displayed | Requirement Based |
| HLTP\_3 | Check whether entered game is correct | user name | Display game name | Display game name | Requirement Based |

### Low Level Test Plan

| **ID** | **Description** | **Expected I/P** | **Expected O/P** | **Actual O/P** | **Type Of Test** |
| --- | --- | --- | --- | --- | --- |
| LLTP\_1 | To check each moving path | path ID | Path displayed with Equal dots | path displayed with equal dots | Requirement Based |
| LLTP\_2 | To check the with not eat the dots | path ID | Display path with person | Display Path with person | Requirement Based |
| LLTP\_3 | To check when path is given equal number of lines | user id quality | single line to moving | single line to moving | Requirement Based |

## 

## Implementation and Summary

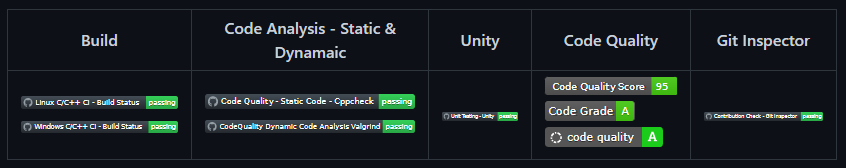
Git Dashboard 

Figure 3-Git Dashboard

# Summary

Git Inspector Summary:

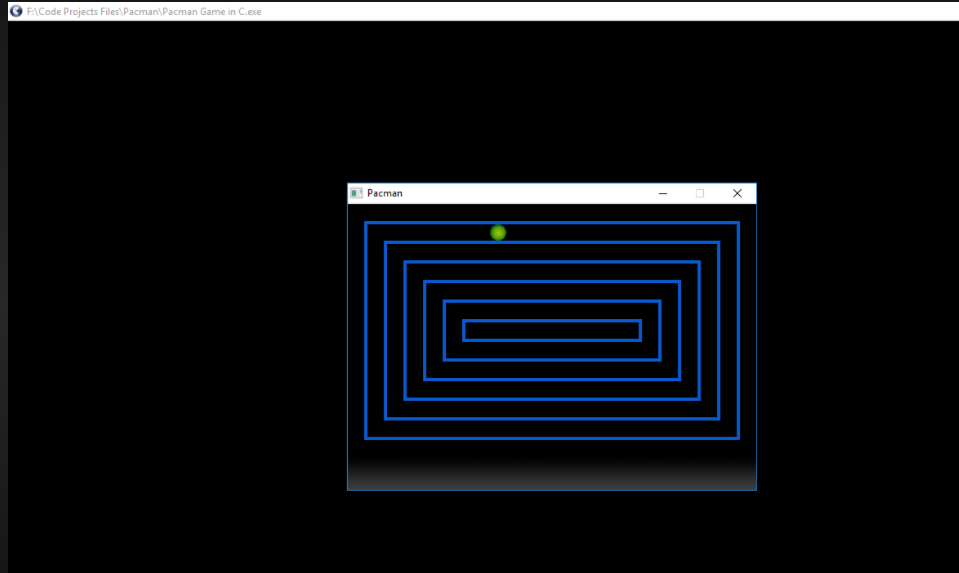


Figure 4-Git Inspector Summery

# 

# Miniproject 2 – Virtual Costume Advisor [Team]

## Modules

1. C Programming
2. Git
3. Visual Studio Code

### Requirements

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

**What:**

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

**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.

**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 | Status |
| --- | --- | --- |
| 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

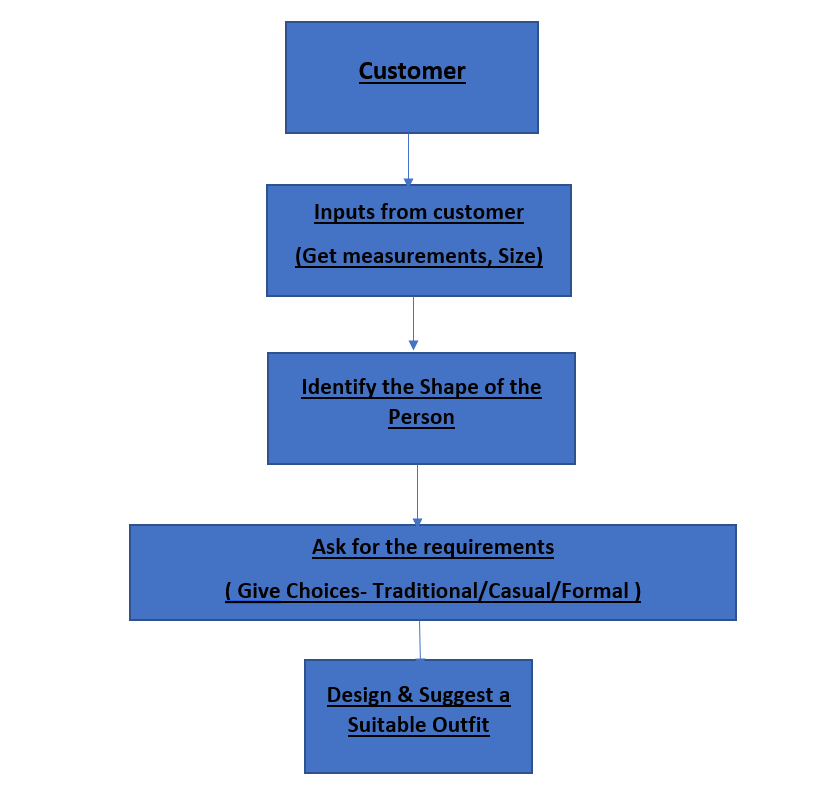


Figure 5- Behaviour Diagram Low Level

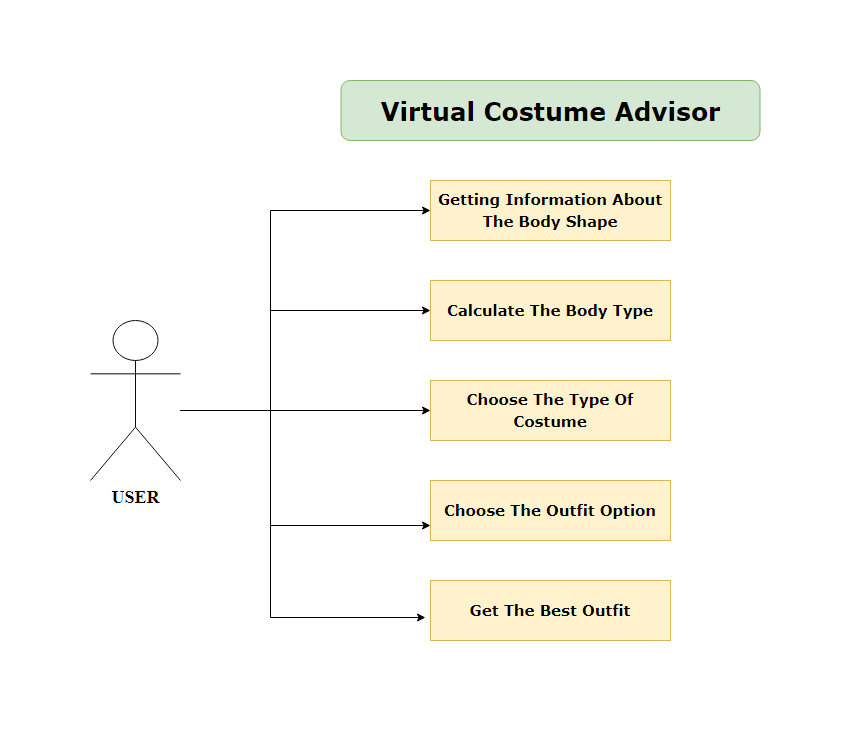


Figure 6- Behaviour Diagram High Level

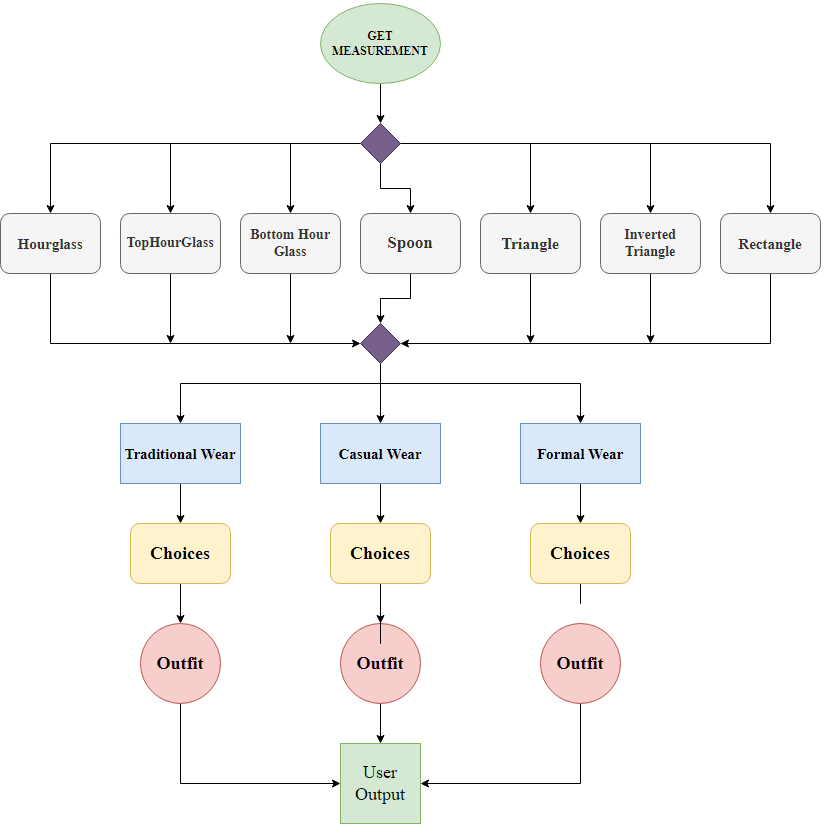


Figure 7- Structural Diagram High Level

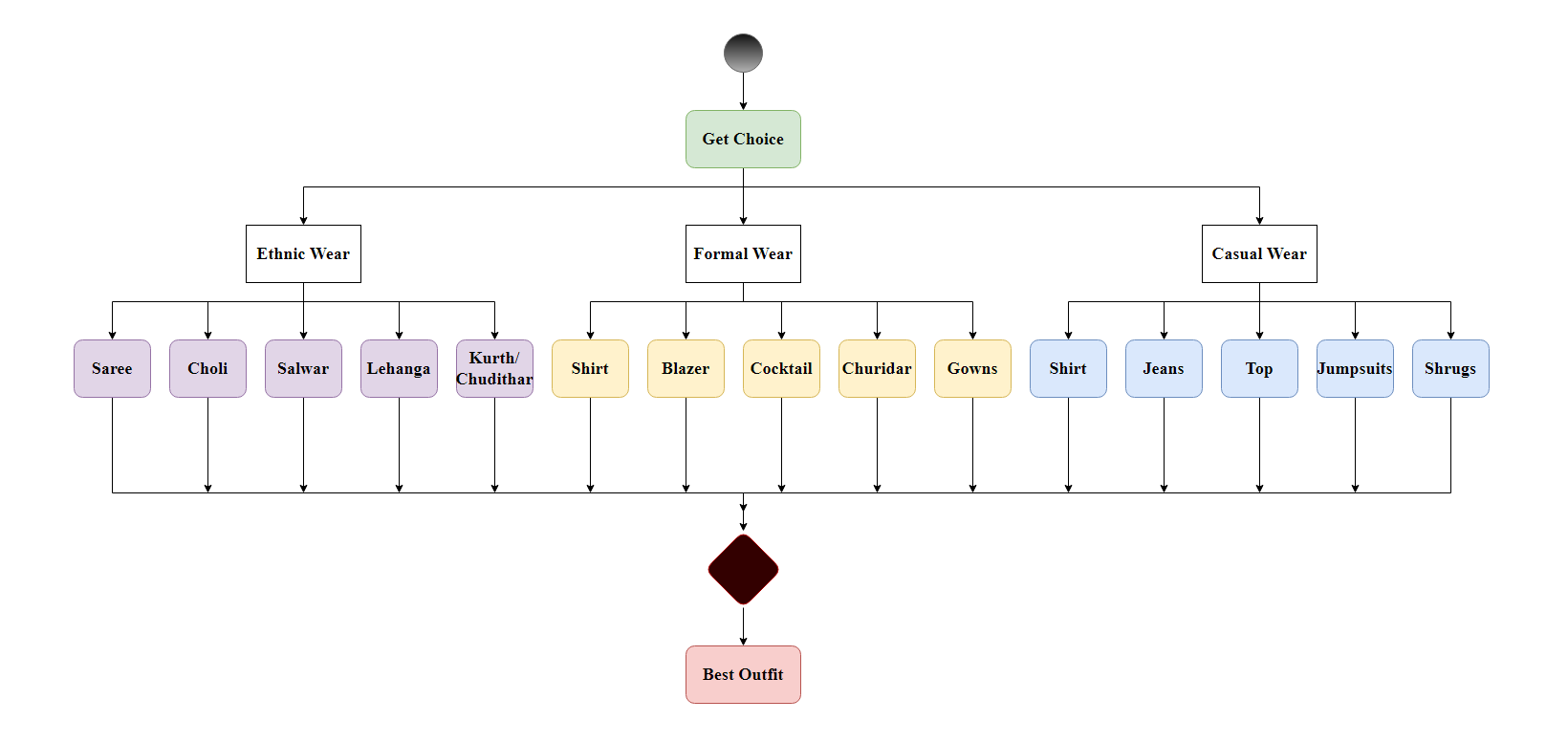


Figure 8- Structural Diagram Low Level

## Test Plan

### High Level Test Plan

| ID | Description | Expected I/P | Expected O/P | Type Of Test |
| --- | --- | --- | --- | --- |
| HLTP\_1 | Check and verify all the measurement values are entered | Enter the proper measurements according to the description | Body shape is displayed | Requirement Based |
| HLTP\_2 | Check and verify all the measurements entered are in centimetre | Enter the measurements in centimetre | Body shape is displayed | Scenario Based |
| HLTP\_3 | Check whether all entered measurements is displayed correctly | Enter the measurements | Body shape is displayed | Scenario Based |
| HLTP\_4 | Check whether choices of outfit types are displayed correctly | Enter the choice | Another set of choices displayed | Scenario Based |

### Low Level Test Plan

| ID | Description | Expected I/P | Expected O/P | Type Of Test |
| --- | --- | --- | --- | --- |
| LLTP\_1 | To check if the measurements give the proper body shape | Measurements | Body Type | Requirement Based |
| LLTP\_2 | To check if the choices give the proper body shape | Choice | The required outfit | Requirement Based |
| LLTP\_3 | To check if the calculation is properly done to give proper output | Body Type | Scenario Based | Requirement Based |
| LLTP\_4 | To check if all of the four required measurements are entered | Bust, Waist, High hip, Hip sizes | Display Body shape | Scenario Based |
| LLTP\_5 | To check if required choice of outfit type is entered | Choices | Display the best outfits | Scenario Based |

## Implementation and Summary

### Git Link:

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

# 

# Miniproject 3: Wiper Control System [Team]

## Modules

1. Microcontrollers
2. Git

### Requirements

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

**Why:**

1. The main purpose of the wiper system is to clean the windscreen sufficiently to provide suitable visibility at all times.

**Where:**

1. On most vehicles, the windshield wipers can be activated by a lever located to the right of the steering wheel.

**What**

1. Now, virtually all automotive wipers are controlled by a microprocessor.
2. Many wiper systems in cars today use a rain sensor to detect the speed at which the raindrops are falling on the windshield.

**When:**

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

**How:**

1. Windshield wipers are controlled by the stalk on the right side of your steering wheel.
2. 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 |
| --- | --- | --- |
| 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 |

## 

## Design

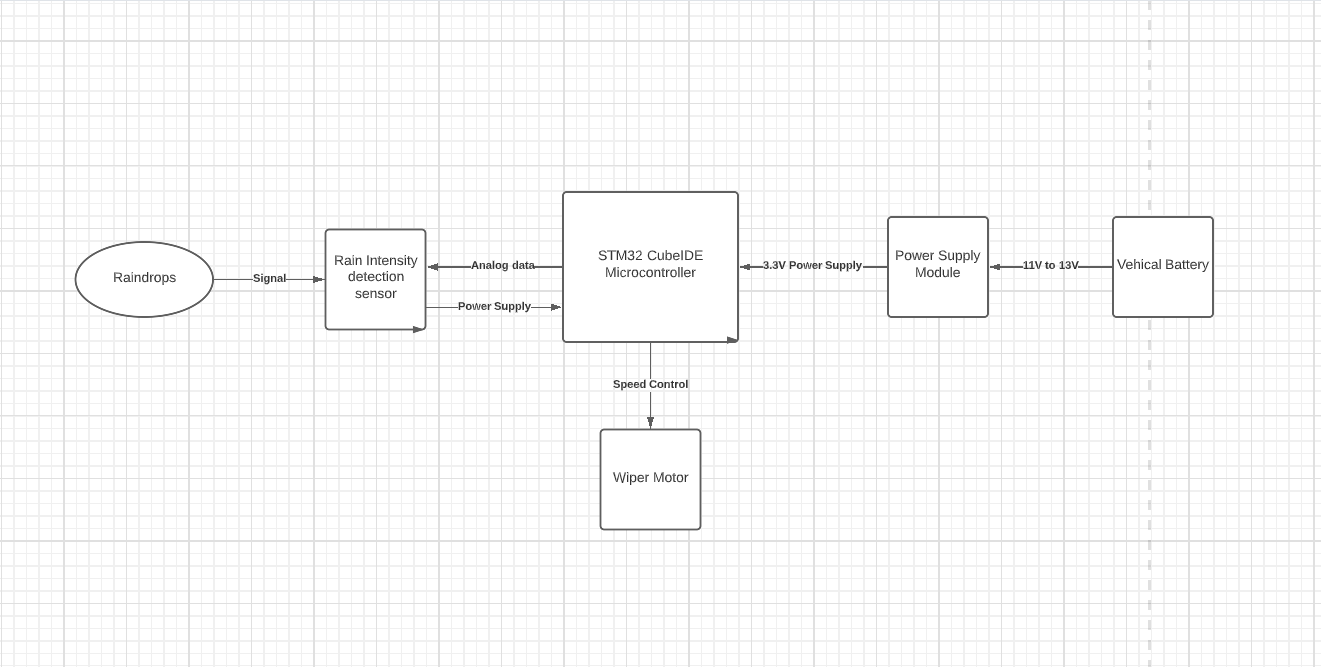


Figure 9- Behaviour Diagram

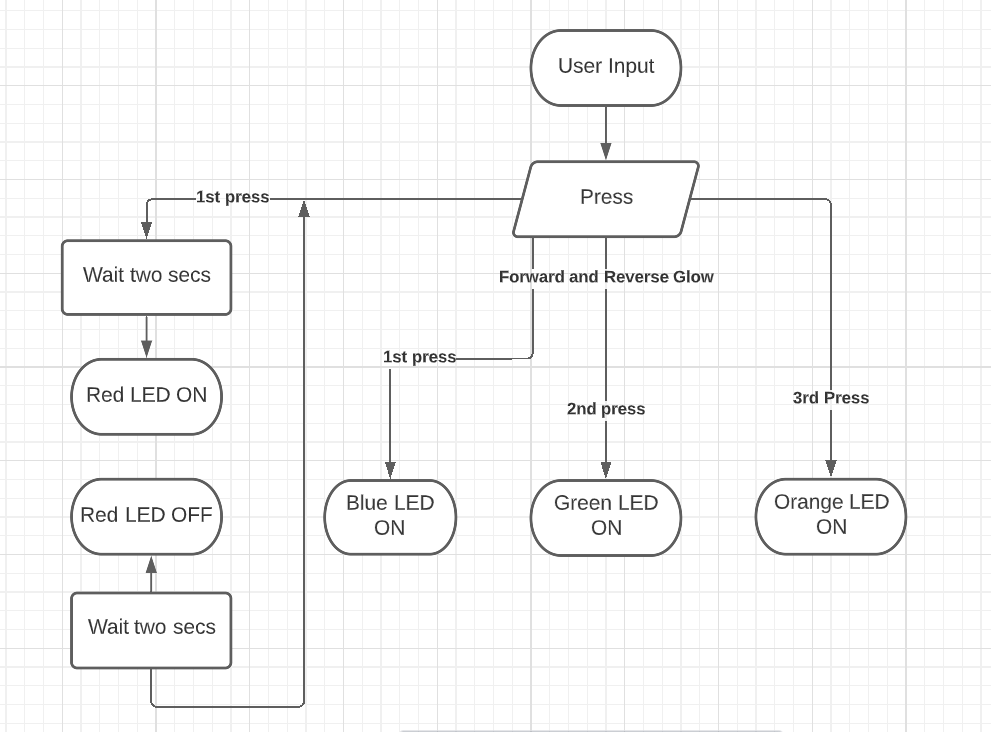


Figure 10- Structure Diagram

## Implementation and Summary

### Git Link:

Link: https://github.com/GENESIS-2022/MasteringMCU-Team76.git

# 

# Miniproject 4 – Calendar Automation [Team]

## Modules

1. Python
2. Git

### Requirements

### High Level Requirements

| ID | Feature | Malab v0 Status | Python v0 Status |
| --- | --- | --- | --- |
| HLR\_01 | GUI | Implemented | Implemented |
| HLR\_02 | Master Calendar | Implemented | Implemented |
| HLR\_03 | Faculty calendar | Implemented | Implemented |
| HLR\_04 | Faculty load sheet | Implemented | Implemented |
| HLR\_05 | Showing Available Open Slots based on faculty and modules | Not Implemented | Not Implemented |

### Low Level Requirements

| ID | Feature | High Level ID | Malab v0 Status |
| --- | --- | --- | --- |
| LLR\_01 | GUI should allow user to enter inputs | HLR\_01 | Not Implemented |
| LLR\_02 | Input Files Based on Different Initiatives and Timelines | HLR\_01 | Not Implemented |
| LLR\_03 | GUI should get Base Calendar as Input | HLR\_01 | Implemented |
| LLR\_04 | GUI should get Month and Initiative as Input | HLR\_01 | Implemented |
| LLR\_05 | GUI should be able to show Conflicts/Warnings | HLR\_01 | Implemented |
| LLR\_06 | Master Calendar: display Month wise | HLR\_02 | Implemented |

## Implementation and Summary

### Git Dashboard

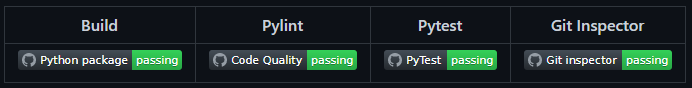


Figure 11- Git Dashboard

# 

# 

# Miniproject 5 – Jaguar Project [Team]

## Modules

1. Automotive Systems
2. 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. Sideview 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

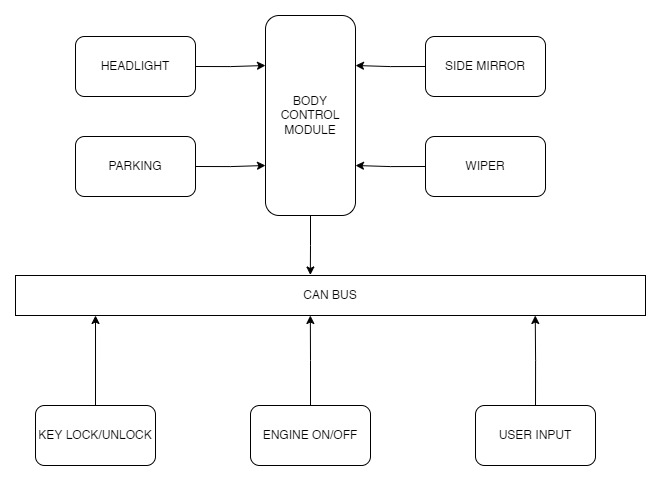


Figure 12- Structure Diagram

# 

# Miniproject 6 – EV Bike [Team]

## Modules

1. Matlab
2. Matlab Script

### Requirements

**Motor Performance:**

1. Our Arrow M1 has a Mid Drive IPM motor which can produce 7.2 kW power and 40 Nm torque. We find these figures to be a nice balance of drivability and efficiency.
2. Arrow M1 has an acceleration time from 0 to 60 km/hr of 6.5 seconds.
3. Top speed of our Arrow M1 is 100 km/hr

**Battery Performance:**

1. We are using a Lithium polymer battery to reduce weight and thereby increase fuel efficiency, performance and handling.
2. A range of 220 km is class leading due to our battery being the biggest at 4.6 kWh.
3. Charging times of our Arrow M1 is higher than the competition at 7.15 hours but we make up for it in the range section.
4. We also offer fast charging.

**Braking Performance:**

1. Our Arrow M1 also uses combi braking system and use disc brakes for both front and back wheels.
2. Braking performance is on par with the competition.

**Wheel Performance:**

1. Our Arrow M1 uses Alloy wheels at 12 inches diameter.
2. We use a 90 section, 90 profile tire for a balance between grip, efficiency and ride quality.

**Suspension Performance:**

1. We use Mono shocks for rear and single fork for front.

**Dimensions:**

1. Our kerb weight is 110 kg which is just 2 kg heavier than the Ather 450X while having a substantially bigger battery and more powerful motor.
2. Length, Height and Weight are all comparable to the competition.
3. Wheelbase is 1370 mm is the longest in the segment.
4. With a seat height of 782 mm it is accessible for a wide range of people in terms of height.

## 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 Bike
2. Researcher: Done case study for EV Bike

# Miniproject 7 – Seat Belt control [Individual]

## Modules

1. Autosar
2. Git

### High Level Requirements

| S.NO | Requirements | Description | Features |
| --- | --- | --- | --- |
| 1 | Pressure control alarm | Alarm buzzes when seat beltis not buckled up even the pressure is applied. | Seat belt control |
| 2 | Open airbag | Depending upon the tension air bag will opened | Seat belt control |
| 3 | Height adjustment of seat belt | Height adjustment of seat belt | Seat belt control |

**Low Level Requirements**

| **S.NO** | **Requirements** | **Description** | **Features** |
| --- | --- | --- | --- |
| 1 | Length of seat belt | Depends upon the model of the car | Seat belt control |
| 2 | Buckles and its types | Lap belts,diagnol belt | Seat belt control |

## 

## Design

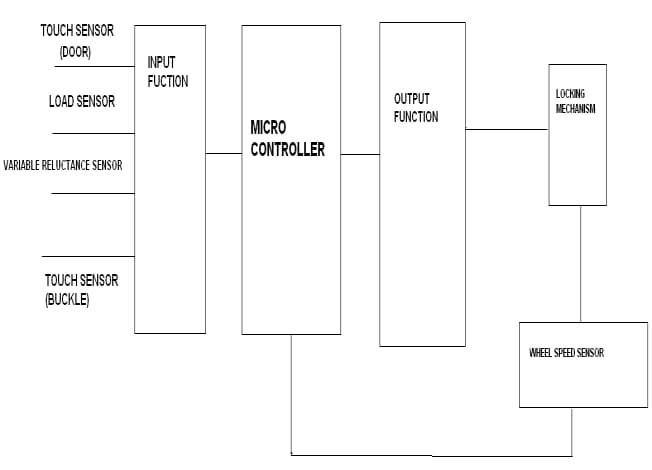


Figure 14 - Power System Diagram

## Implementation and Summary

### Git Link:

https://github.com/BTRIVENI/Autosar\_Triveni\_40021070\_TRN.git