Details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ver. Rel. No. | Release Date | Prepared By | Reviewed By | To Be Approved | Remarks/Revision Details |
| 1.0 | 16/02/2022 | Yarasani Vedavathi  40021035 |  |  |  |

Contents

[Miniproject – 1: Pacman Game [Individual] 5](#_Toc95931104)

[Modules: 5](#_Toc95931105)

[Requirements 5](#_Toc95931106)

[High Level Requirements 6](#_Toc95931107)

[Low Level Requirements 6](#_Toc95931108)

[Design 7](#_Toc95931109)

[Test Plan 8](#_Toc95931110)

[High Level Test Plan 8](#_Toc95931111)

[Low Level Test Plan 9](#_Toc95931112)

Implementation and Summery

Summary

Git Inspector Summary

[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 14](#_Toc95931125)

[Low Level Test Plan 15](#_Toc95931126)

[Implementation and Summary 15](#_Toc95931127)

[Git Link: 15](#_Toc95931128)

[Miniproject 3 –Wiper Control System [Team] 16](#_Toc95931130)

[Modules 16](#_Toc95931131)

[Requirements 16](#_Toc95931132)

[High Level Requirements 17](#_Toc95931133)

[Low Level Requirements 17](#_Toc95931134)

[Design 18](#_Toc95931135)

[Summary 20](#_Toc95931139)

[Git Link: 20](#_Toc95931140)

[Miniproject 4 – Calendar Automation[Team] 21](#_Toc95931143)

[Modules 21](#_Toc95931144)

[Requirements 21](#_Toc95931145)

[High Level Requirements 21](#_Toc95931146)

[Low Level Requirements 21](#_Toc95931147)

[Summary 23](#_Toc95931151)

[Git Link: 23](#_Toc95931152)

[Miniproject 5 – Jaguar Project [Team] 25](#_Toc95931156)

[Modules 25](#_Toc95931157)

[Requirements 25](#_Toc95931158)

[Design 25](#_Toc95931159)

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

[Modules 26](#_Toc95931161)

[Requirements 26](#_Toc95931162)

[High Level Requirements 26](#_Toc95931163)

[Low Level Requirements 27](#_Toc95931164)

[Design 28](#_Toc95931165)

[Git Link: 31](#_Toc95931170)

[Miniproject 7 – Power Window [Individual] 32](#_Toc95931172)

[Modules 32](#_Toc95931173)

[Requirements 32](#_Toc95931174)

[Design 33](#_Toc95931175)

[Implementation and Summary 33](#_Toc95931176)

[Git Link: 33](#_Toc95931177)

## List of Figures

[Figure 1 Behavior Diagram 8](#_Toc95933213)

[Figure 2 Structure Diagram 9](#_Toc95933214)

[Figure 3 Git Dashboard 11](#_Toc95933215)

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

[Figure 5 Behavior Diagram 13](#_Toc95933217)

[Figure 6 Structure Diagram 14](#_Toc95933218)

[Figure 7 Block Diagram 14](#_Toc95933219)

[Figure 8 Simulation 15](#_Toc95933220)

[Figure 9 Git Dashboard 16](#_Toc95933221)

[Figure 10 Behavior Diagram 19](#_Toc95933222)

[Figure 11 UserFlow Diagram 19](#_Toc95933223)

[Figure 12 Structure Diagram 20](#_Toc95933224)

[Figure 13 Git Dashboard 25](#_Toc95933225)

[Figure 14 Git Inspector Summary 25](#_Toc95933226)

[Figure 15 Structure Diagram 29](#_Toc95933227)

[Figure 16 Behavior Diagram 30](#_Toc95933228)

[Figure 17 Structure Diagram 34](#_Toc95933229)

[Figure 18 VFB Diagram 37](#_Toc95933230)

# 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

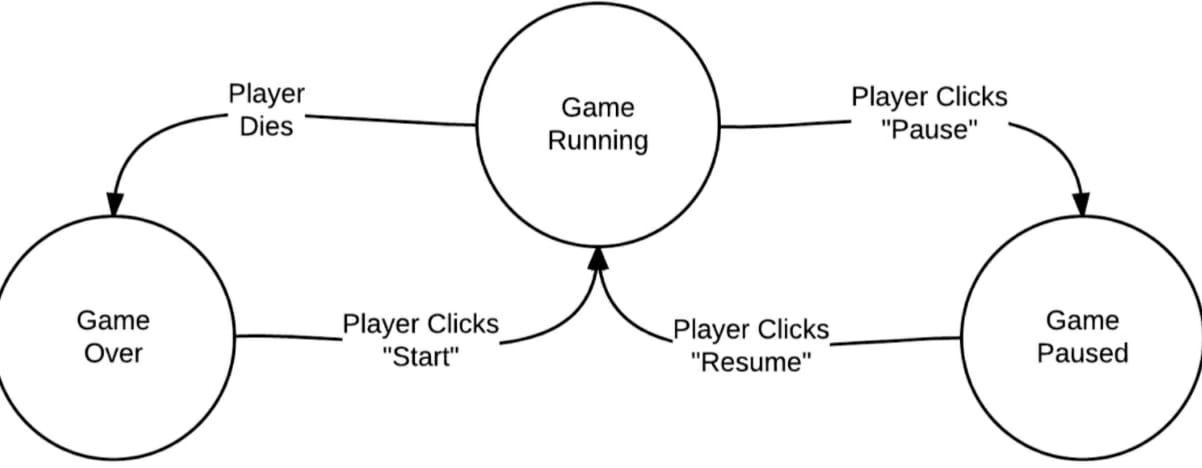


Figure 1- Behaviour Diagram1

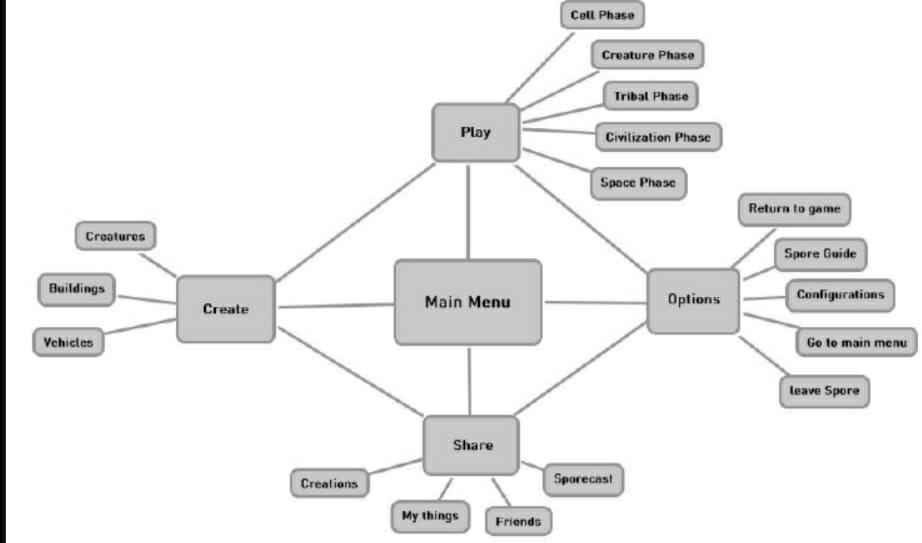


Figure 1- Behaviour Diagram2

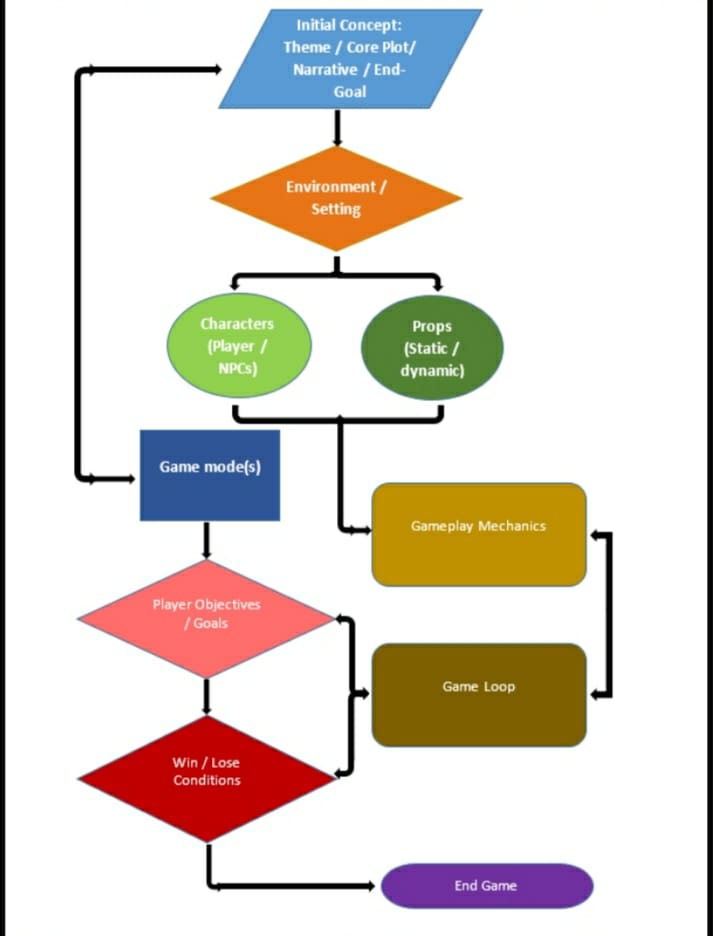


Figure 2-Structure Diagram1

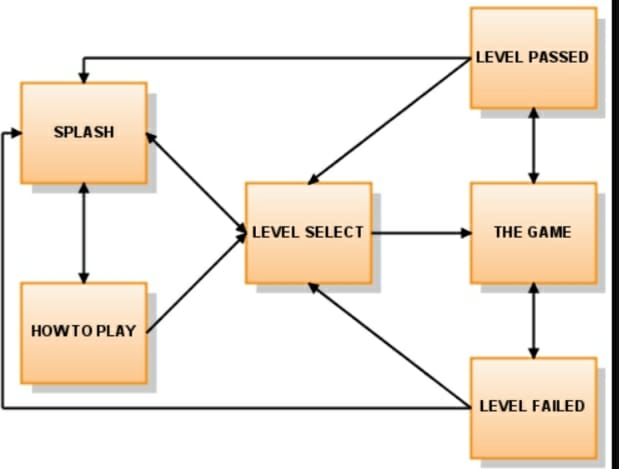


Figure 2-Structure Diagram2

## 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 Link:

Link: <https://github.com/18125a0525/M1_Game_Pacman.git>

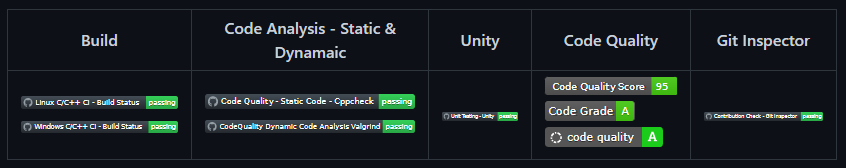
Git Dashboard 

Figure 3-Git Dashboard

# Summary

Git Inspector Summary:

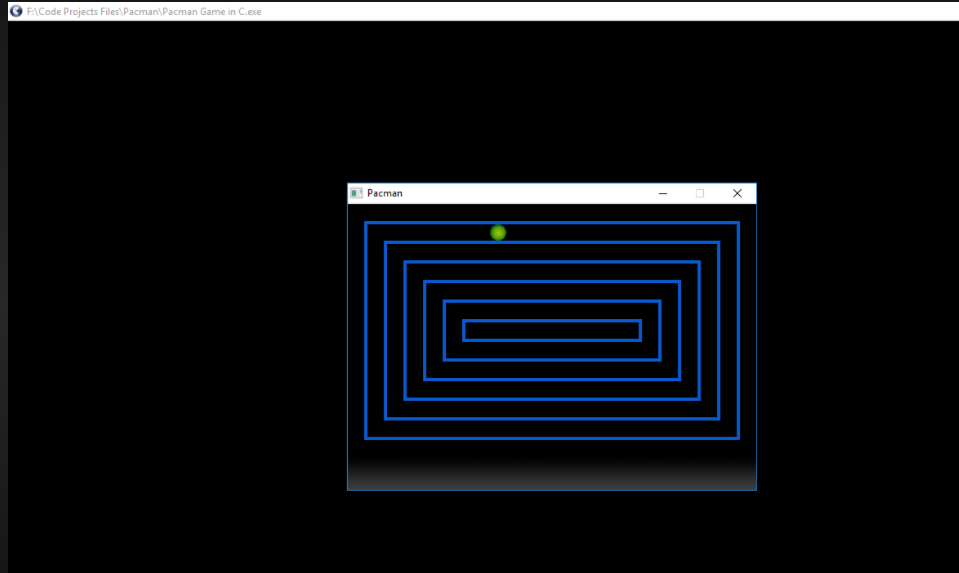


Figure 4-Git Inspector Summery

# 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

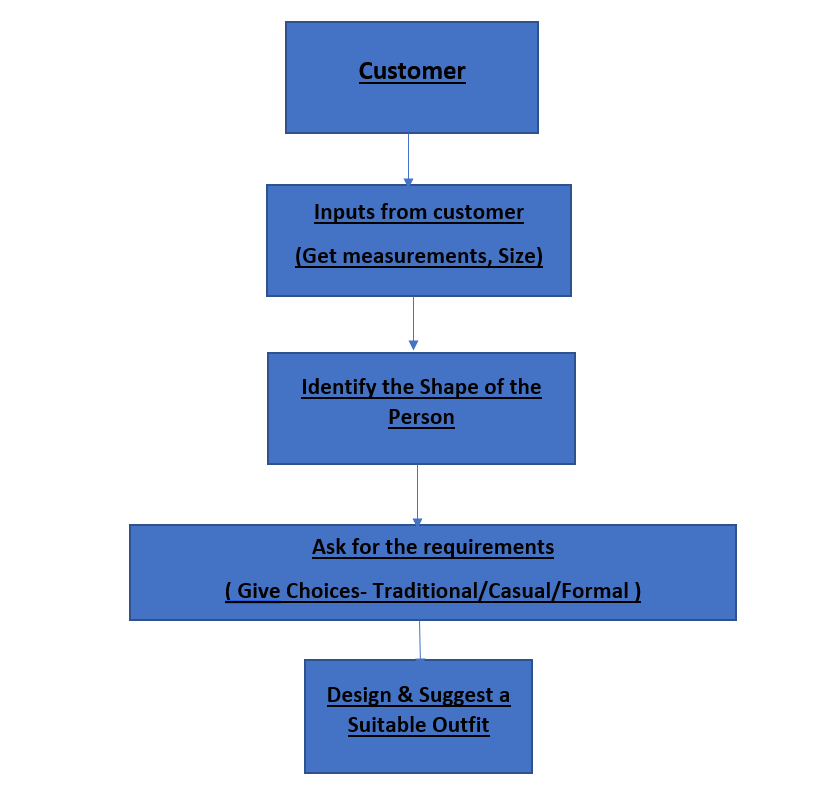


Figure 5- Behaviour Diagram Low Level

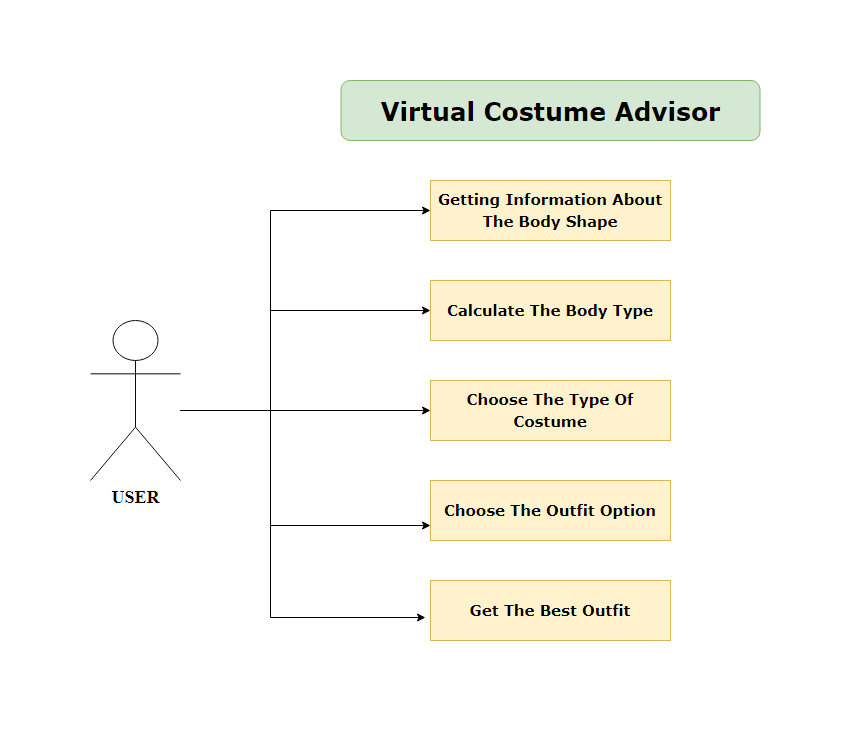


Figure 6- Behaviour Diagram High Level

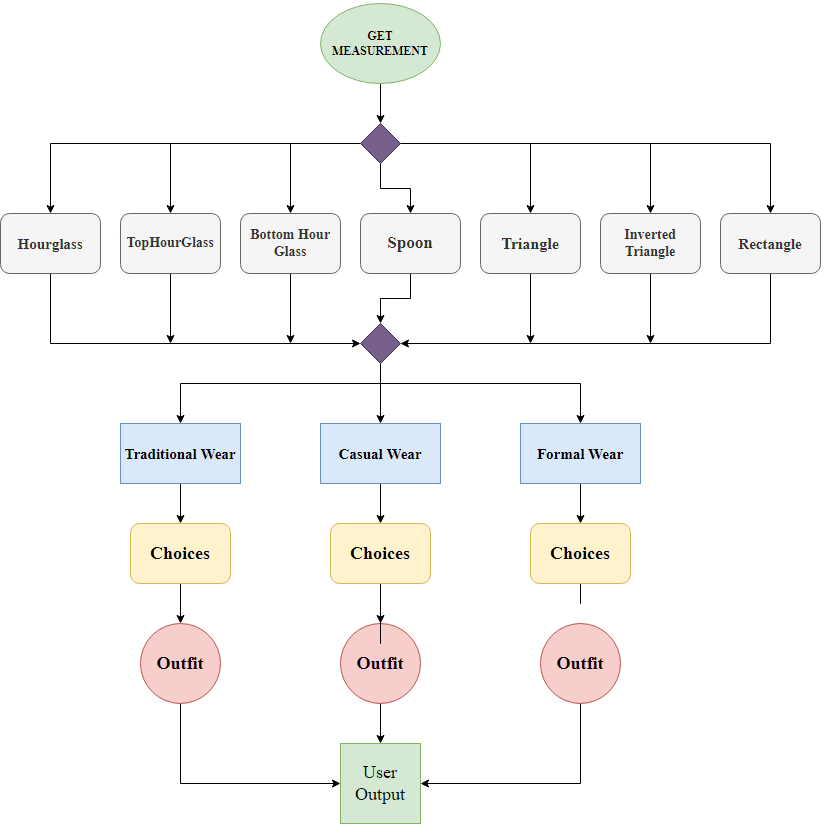


Figure 7- Structural Diagram High Level

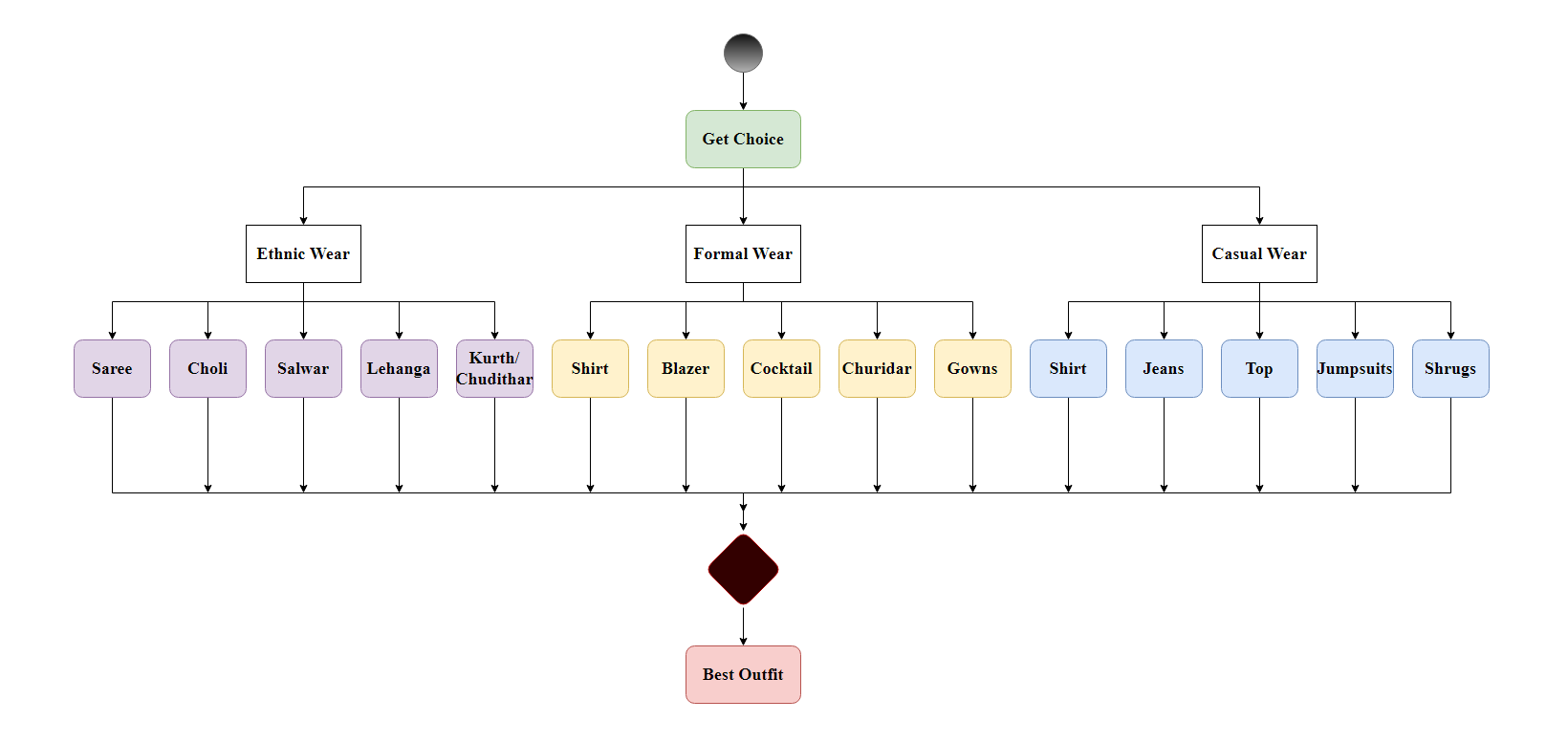


Figure 8- Structural Diagram Low Level

## 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

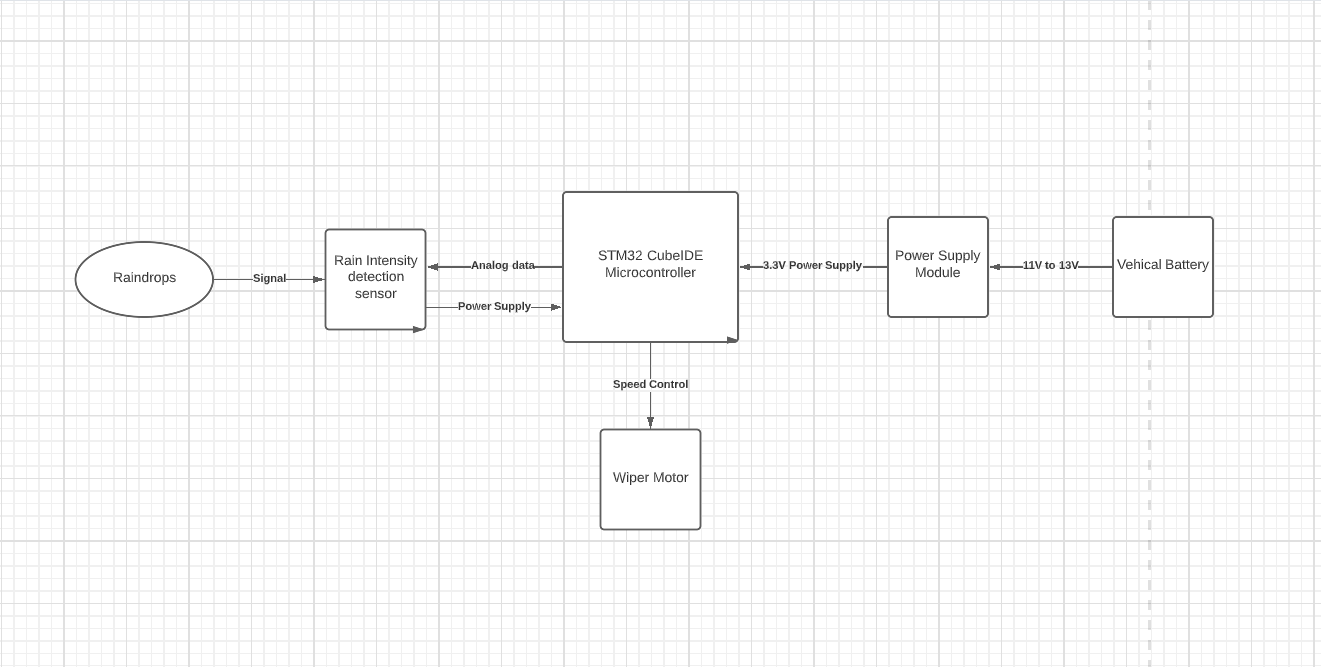


Figure 9- Behaviour Diagram

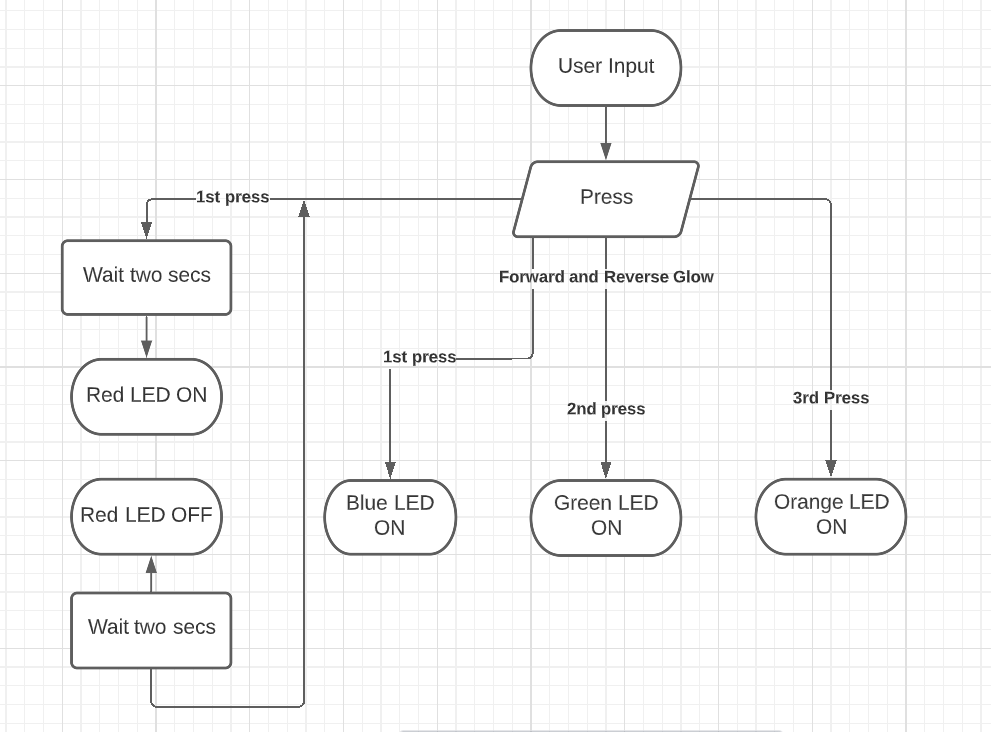


Figure 10- Structure Diagram

## Implementation and Summary

### Git Link:

Link: <https://github.com/GENESIS-2022/MasteringMCU-Team70.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 Link:

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

### Git Dashboard

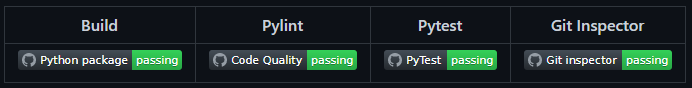


Figure 11- Git Dashboard

# 

# 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

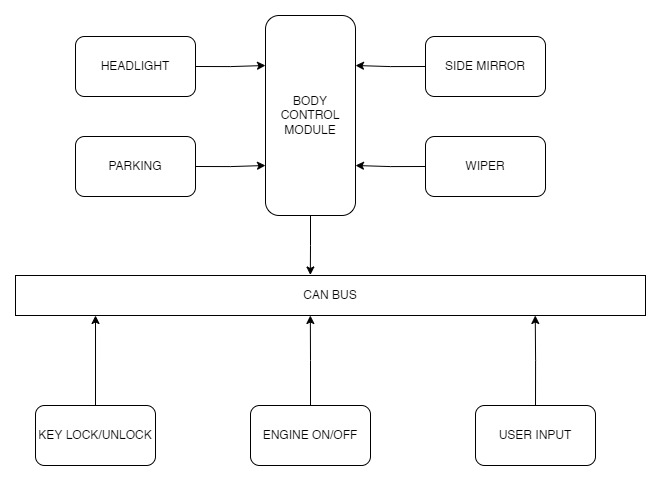


Figure 12- Structure Diagram

# 

# Miniproject 6 – EV Truck [Team]

## Modules

1. Matlab
2. Matlab Script

### Requirements:

### Introduction:

1. The new Volvo FE Electric will be offered in several variants for different types of transport assignment.
2. For instance, with Volvo’s low-entry cab, which makes it easier to enter and exit the cab and gives the driver a commanding view of surrounding traffic.
3. The working environment improves too as a result of the low noise level and vibration-free operation.
4. Battery capacity can be optimised to suit individual needs, and charging takes place either via the mains or via quick-charge stations.
5. The Volvo FE is easy to handle even in the most congested spaces.
6. It is a vehicle that allows for easy manoeuvring and smooth acceleration, as well as providing an array of instruments to help you navigate every situation.
7. The Volvo FE can be specified to meet your superstructure demands.
8. The FE 6×2 rigid with its low noise level, excellent all-round visibility and easy manoeuvring makes it the perfect choice for refuse transporting operations.
9. Frequent stops and starts are aided by the D8K engine with power ratings up to 350 hp, combined with automatic transmission or I-Shift.
10. When super low emissions are required, there’s a CNG powered driveline option available.
11. The low instep offers a convenient entry with anti-slip steps, and the cabs offer among the best close up and all-round visibility there is in the industry.
12. The Volvo FE engine range ensures performance for city and regional distribution as well as construction duties – while at the same time offering alternatives for sensitive environments with stringent emissions regulations.
13. An extended output range stretching from 250 to 350 hp including synthetic diesel, biodiesel and gas alternatives enables you to tailor the driveline for your needs.
14. This is part of our long-term ambition to move towards zero emissions.

## BENEFITS OF ELECTRIC TRUCKS

1.It is always cheaper to charge your electric truck than spending money on gas.

2. Electric trucks provide businesses with many benefits that primarily aim for the long run.

3.EV trucks do not require fuel, which is already one of the biggest advantages, due to fuel cost and effect on nature.

4.Driving electric trucks reduces CO2 emissions and actually offers better performance for diverse trucks also have less parts, which should lead to less damage and lower maintenance.

5.However, this depends on a truck model and its usage. While driving within urban areas with frequent stops and speeding, it is way more efficient to drive electric truck than diesel truck.

### **The specification is given below**:

# Battery:

1. Power and torque.
2. The G9K engine delivers a full 320 hp
3. and 1356 Nm of torque, and has been
4. primarily developed for operations
5. involving short driving cycles with frequent starts and stop.

# Automatic transmission:

1. The 6-speed automatic transmission
2. makes the Volvo FE CNG an easy to use
3. productive tool for urban transports.

# Operating range:

1. Choose between 2×4 or 2×3 gas tanks
2. fitted on the chassis. This gives an
3. operating range of up to 400 km in light
4. distribution and up to 250 km in severe
5. refuse collection.

## Lithium-ion Polymer

* 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 a an atom or molecule with an electric charge caused by the loss or gain of one or more electrons.
  3. Battery Features:
  4. 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

## 

# Controller:

PID Controller:

- Proportional action is the “universal” control action, capable of providing at least marginal control quality for any process.

- Integral action is useful for eliminating offset caused by load variations and process self-regulation.

- Derivative action is useful for cancelling lags, but useless by itself.

- Limitations of each action

- Proportional action will cause oscillations if sufficiently aggressive, in the presence of lags and/or dead time.

- transportation conditions are not subject to regulatory control.

- Gain and phase shift of each action

- Proportional action acts on the present, adding no phase shift to a sinusoidal signal. Its gain is constant for any signal frequency.

Motor:

Permanent Magnet Synchronous Motor (PMSM)

## MODEL DESIGN

- COST – ₹ 15.29 Lakh to ₹ 16.82 Lakh

- GROSS COMBINATION WEIGHT – 7300 kg

- RANGE – More than 100 KMs

- BATTERY - nickel-metal hydride

- BATTERY CAPACITY – 60–120 wH/kg

- CHARGING TIME (Full Charge) – 3-4hrs

- DRIVELINE/MOTOR – Permanent Magnet Synchronous Motor (PMSM)

- PERFORMANCE – 100hp

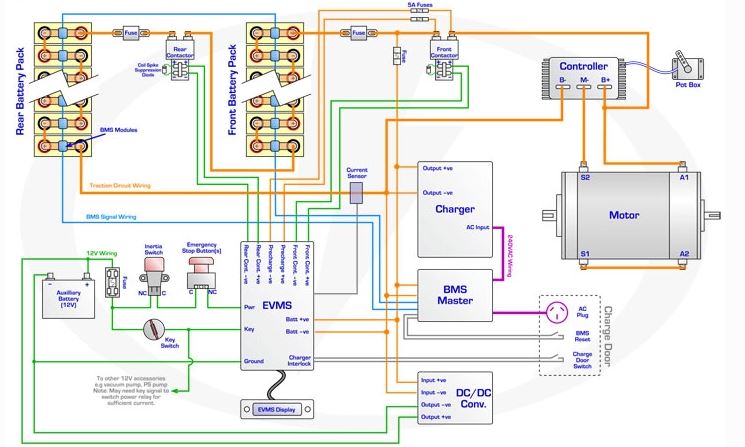
- ELECTRIC MOTOR TORQUE PTO (peak/continuous)- 180Nm

- INVERTER – Grid-tied string

- WHEEL BASE - 3310mm

- CONTROLLER – PID Controller# MODEL DESIGN

## SIMULATION DESIGN



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

Git Link: <https://github.com/raginibhandare/Team11_EV_TRUCK.git>

# Miniproject 7 – Power Window [Individual]

## Modules

1. Autosar
2. Git

### High Level Requirements

| S.NO | Requirements | Description | Features |
| --- | --- | --- | --- |
| 1 | Obstacle Sensor | when obstacle detected the operation is delayed for 10seconds | Power window |
| 2 | Both driver and passenger control | operated with the first received command (either by driver or passenger) | Power window |
| 3 | Dust detected sensor | window will be closed when the dust is detected | Power window |

**Low Level Requirements**

| **S.NO** | **Requirements** | **Description** | **Features** |
| --- | --- | --- | --- |
| 1 | Semi controlled | manually/automatic | Power window |
| 2 | Engine Off | Can be operated manually | Power window |

## Design

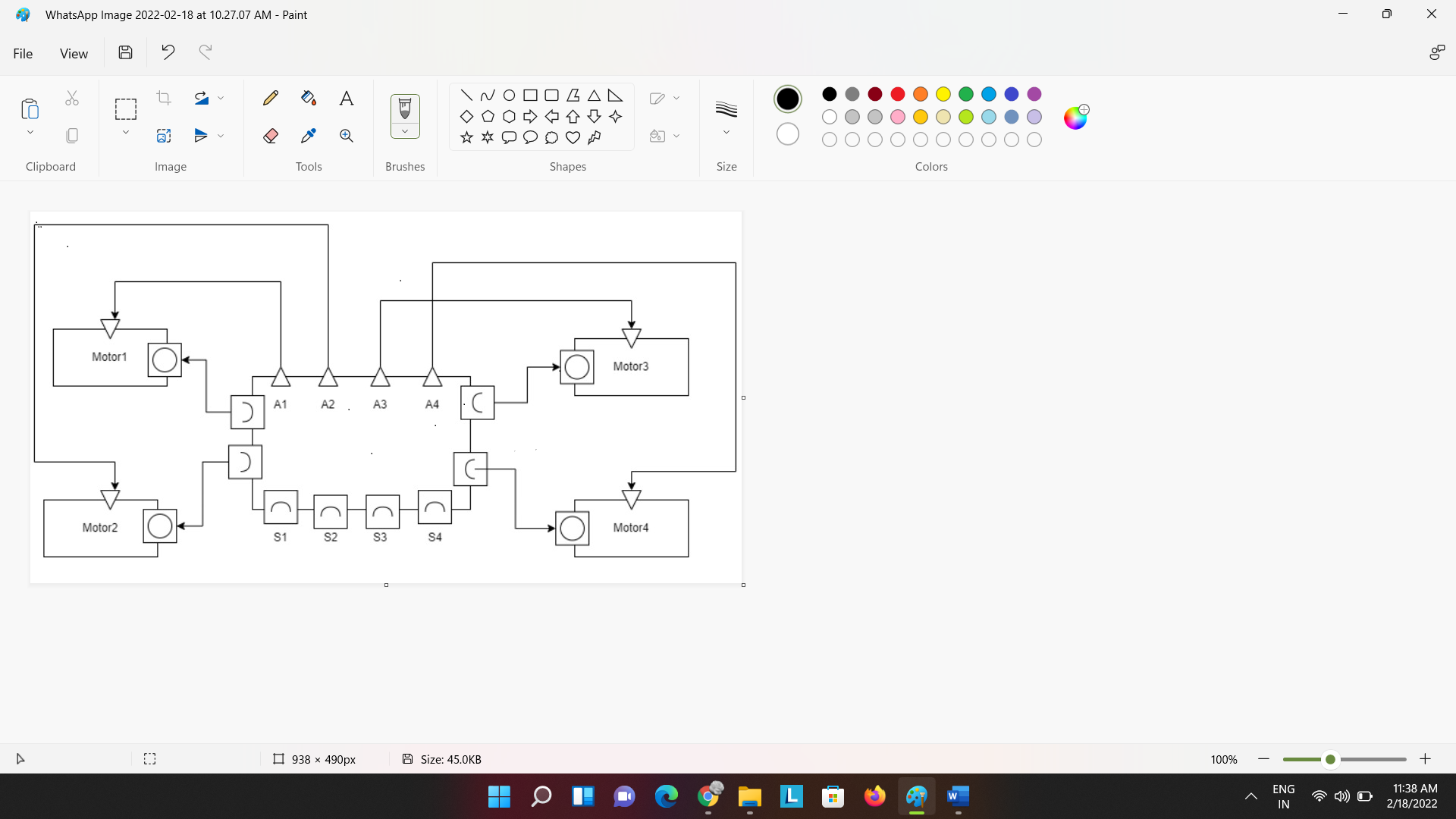


Figure: Power System Diagram

## Implementation and Summary

### Git Link:

Link: <https://github.com/18125a0525/Autosar_Yarasani-Vedavathi_40021035_TRN.git>