**Details**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ver. Rel. No. | Release Date | Prepared By | Reviewed By | To Be Approved | Remarks/Revision Details |
| 1.0 | 21/02/2022 | Vinaya Ranade  40021134 |  |  |  |

Contents

[List of Figures 4](#_Toc96351336)

[Miniproject – 1: Airline booking system [Individual] 5](#_Toc96351337)

[Modules: 5](#_Toc96351338)

[Requirements 5](#_Toc96351339)

[High Level Requirements 6](#_Toc96351340)

[Low Level Requirements 6](#_Toc96351341)

[Design 7](#_Toc96351342)

[Test Plan 9](#_Toc96351343)

[High Level Test Plan 9](#_Toc96351344)

[Low Level Test Plan 10](#_Toc96351345)

[Implementation and Summary 10](#_Toc96351346)

[Git Link: 10](#_Toc96351347)

[Mini project 2 – Embedded Calculator [Individual] 11](#_Toc96351348)

[Modules 11](#_Toc96351349)

[Requirements 11](#_Toc96351350)

[High Level Requirements 12](#_Toc96351351)

[Low Level Requirements 12](#_Toc96351352)

[Design 13](#_Toc96351353)

[Git Link: 14](#_Toc96351354)

[Miniproject 3 – Diary Entry System [Team] 15](#_Toc96351355)

[Modules 15](#_Toc96351356)

[Requirements 15](#_Toc96351357)

[High Level Requirements 16](#_Toc96351358)

[Low Level Requirements 16](#_Toc96351359)

[Design 17](#_Toc96351360)

[Figure 14. Component structure diagram 19](#_Toc96351361)

[Figure 16. Case diagram 20](#_Toc96351362)

[Test Plan 20](#_Toc96351363)

[High Level Test Plan 20](#_Toc96351364)

[Low Level Test Plan 21](#_Toc96351365)

[Implementation and Summary 21](#_Toc96351366)

[Git Link: 21](#_Toc96351367)

[Individual Contribution and Highlights 21](#_Toc96351368)

[Miniproject 4 – Calendar Automation [Team] [Team] 22](#_Toc96351369)

[Modules 22](#_Toc96351370)

[Requirements 22](#_Toc96351371)

[High Level Requirements 22](#_Toc96351372)

[Low Level Requirements 23](#_Toc96351373)

[Link for template standard input template: 24](#_Toc96351374)

[Requirements for updating Master calendar using Master calendar as input 24](#_Toc96351375)

[Link for template 24](#_Toc96351376)

[App deployment 25](#_Toc96351377)

[Additional features for V1 to do 25](#_Toc96351378)

[Git Link: 25](#_Toc96351379)

[Miniproject 5 – Team BMW [Team] 26](#_Toc96351380)

[Modules 26](#_Toc96351381)

[Miniproject 6 – Wiper Control[Team] 32](#_Toc96351382)

[Modules 32](#_Toc96351383)

[Requirements 32](#_Toc96351384)

[Low Level Requirements 32](#_Toc96351385)

[Design 34](#_Toc96351386)

[Test Plan 35](#_Toc96351387)

[High Level Test Plan 35](#_Toc96351388)

[Low Level Test Plan 36](#_Toc96351389)

[Implementation and Summary 37](#_Toc96351390)

[Git Link: 37](#_Toc96351391)

[Miniproject 7 – Ford Project [Team] 38](#_Toc96351392)

[Modules 38](#_Toc96351393)

[Requirements 38](#_Toc96351394)

[Ford Aspire 38](#_Toc96351395)

[Body Control Module 38](#_Toc96351396)

[Features: 38](#_Toc96351397)

[4W's and 1H 39](#_Toc96351398)

[What : 39](#_Toc96351399)

[Who : 39](#_Toc96351400)

[Where : 39](#_Toc96351401)

[When : 39](#_Toc96351402)

[How : 39](#_Toc96351403)

[Design 40](#_Toc96351404)

[Implementation and Summary 40](#_Toc96351405)

[Git Link: 40](#_Toc96351406)

[Individual Contribution and Highlights 41](#_Toc96351407)

[Mini project 8 – EV Bike[Team] 42](#_Toc96351408)

[Modules 42](#_Toc96351409)

[Requirements 42](#_Toc96351410)

[Implementation and Summary 47](#_Toc96351411)

[Individual Contribution and Highlights 47](#_Toc96351412)

[Miniproject 9 – Door Locking System[Individual] 48](#_Toc96351413)

[Modules 48](#_Toc96351414)

[Requirements 48](#_Toc96351415)

[High level Requirement 48](#_Toc96351416)

[Low level Requirement 49](#_Toc96351417)

[Design 49](#_Toc96351418)

[Implementation and Summary 49](#_Toc96351419)

[Git Link: 49](#_Toc96351420)

[Individual Contribution and Highlights 50](#_Toc96351421)

## List of Figures

[Figure 1 Behavior Diagram 8](#_Toc95933213)

Figure 2 Feature level Behavioural Diagram………………………………………………….9

Figure 3 Feature level Behavioural Diagram………………………………………………...11

Figure 4 Feature level Behavioural Diagram………………………..………………………12

Figure 5 Behavioural Diagram…..………………………………………………………….12

Figure 6 Circuit Diagram …………………………………………………………………..14

Figure 10.Behavirioul Diagram…………………………………………………………….16

Figure 12 Delete Record Feature Diagram………………………………………………….20

Figure 13 .Add Record feature level diagram………………………………………………22

Figure 14. Component structure diagram………………………..…………………………24

Figure 15. Flow structure diagram………………………………………………………….25

Figure 16. Case diagram……………………………………………………………………27

Figure 17. Structural Diagram………………………………………………………………30

Figure 18.Block Diagram …………………………………………………………………..33

Figure 19.Model Design……………………………………………………………….35

Figure 20. Model Design ……………………………………………………………..37

Figure 21.Model Design………………………………………………………………42

Figure 22.Model Simulation……………………………………….…………………45

Figure 23..Model Simulation…………….…………………………………………..47

Figure 24. Block Diagram……………………………………………………………47

Figure 25. Block Diagram…………………………………………………………….48

Figure 26. Class Diagram……………………………………………………………..49

Figure 27 .Deployment Diagram……………………………………………………..49

Figure 30 .VFB Diagram…………………………………………………………….50

# Miniproject – 1: Airline booking system [Individual]

## Modules:

1. C Programming
2. Git

### Requirements

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

**Why:**

* Air travel is one of the first choice among most of the customers to save time
* The main objective is to reduce time by simplifying the reservation process for customers who do not want to visit travel agents or airports for booking tickets.

**Where:**

* The system can be accessed easily by users anywhere anytime at their own comfort rather than visiting airport booking counters.

**Who:**

* This system will help the airline companies to provide an easy booking service to their customers directly without the need of any travel agents or separate booking counters.

**When:**

* The need for airline reservation system was realized since the growth in number of airline companies all with multiple prices, aircrafts and schedules

**How:**

* Airline Booking will ask for the passenger’s statistics which include name, passport no, email id. After these procedures, a reservation is done.
* As a reservation proof, the system provides seat number.
* In order to check tickets, the person has to offer seat no. then the system checks for the respective tickets in and shows a result.
* Cancelling a flight reservation is easy thru the system, the user simply has to provide their seat number.

### High Level Requirements

| **ID** | **Description** | **Status** |
| --- | --- | --- |
| HLR 1 | It should make a reservation and display seat number to the customer. | Implemented |
| HLR 2 | It should display all the reservation details of reserved seats . | Implemented |
| HLR 3 | It should cancel a seat and make it available for reservation as soon as cancelled. | Implemented |

### Low Level Requirements

| **ID** | **Description** | **Status** |
| --- | --- | --- |
| LLR 1 | It should take required details from the customer before making a reservation | Implemented |
| LLR 2 | It should check if the entered details are valid and send error in case of invalid | Implemented |
| LLR 3 | It should generate a unique seat number for every reservation | Implemented |
| LLR 4 | It should take input seat details before cancellation | Implemented |
| LLR 5 | Take input details of customer and display from existing records | Implemented |

## Design

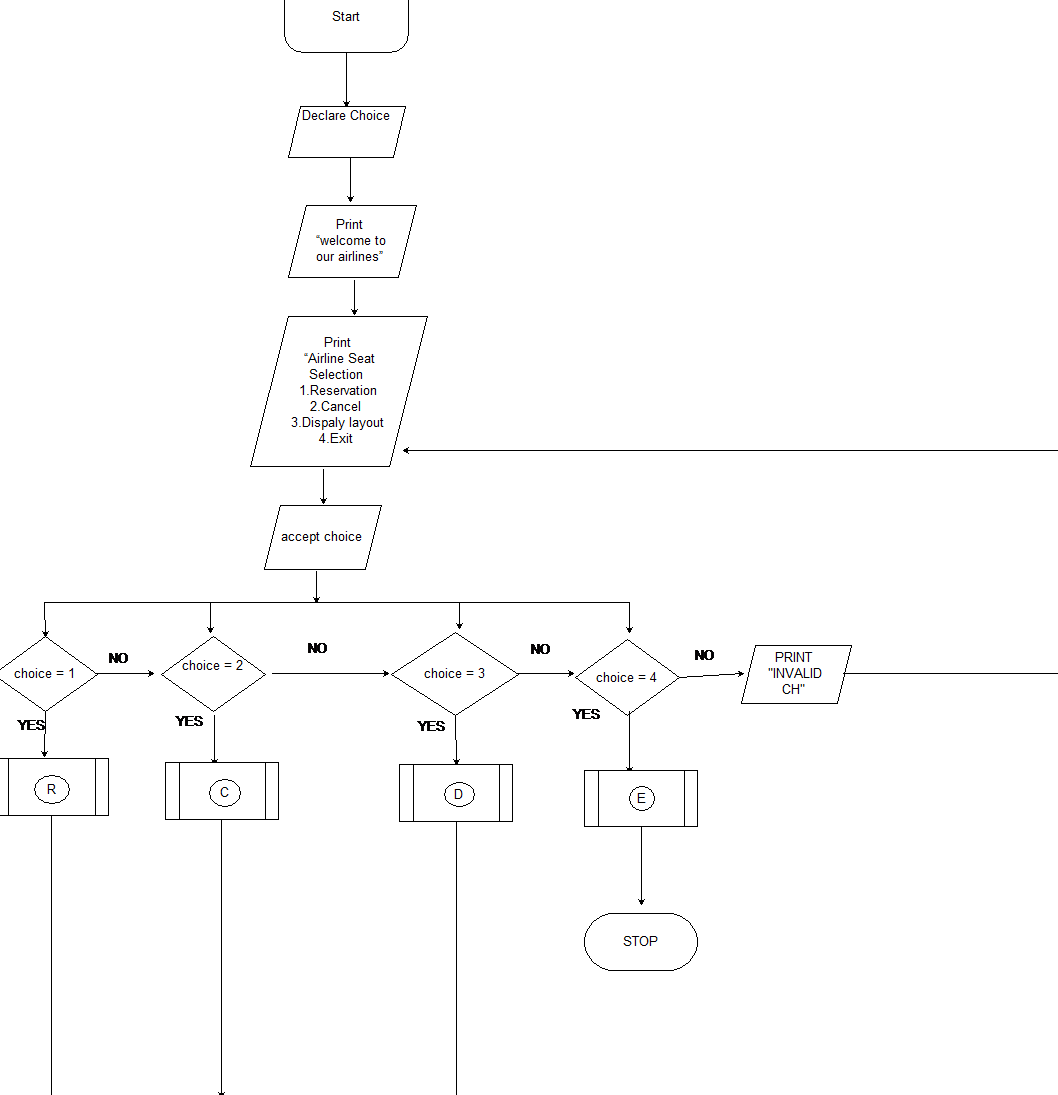


Figure 1 Behavioural Diagram

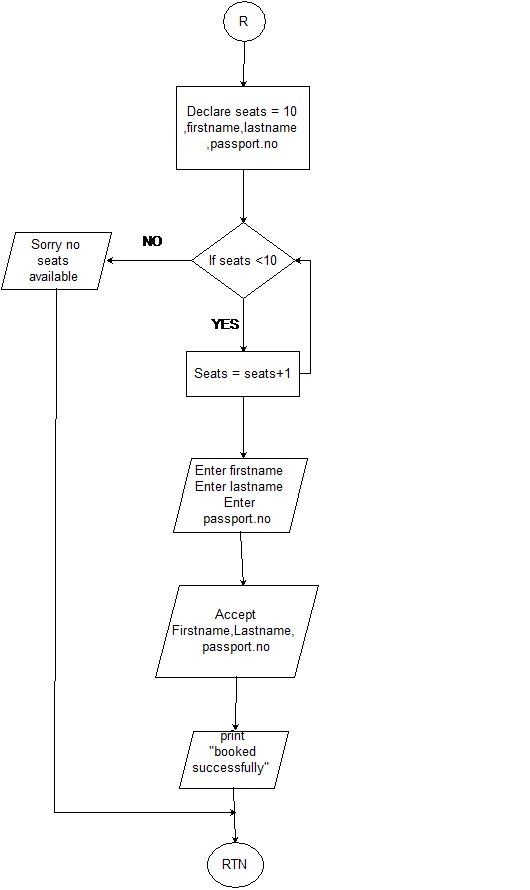


Figure 2 Feature level Behavioural Diagram

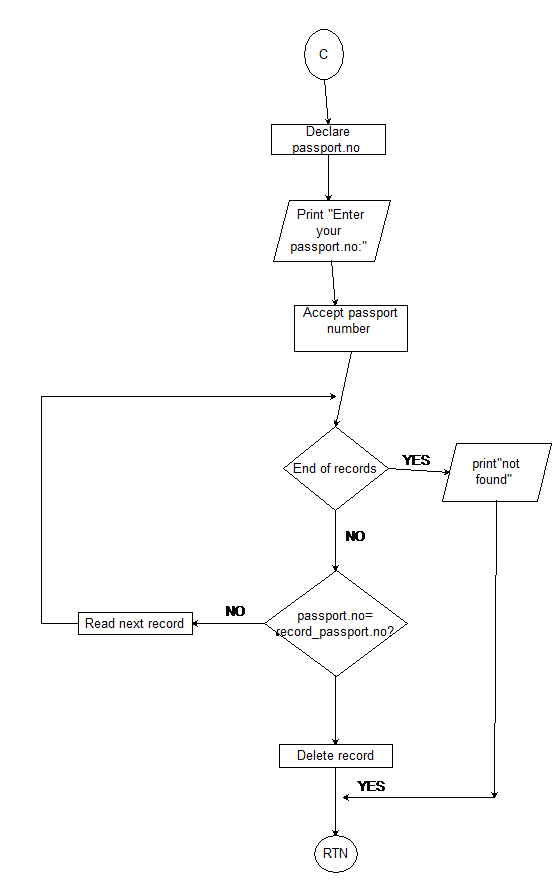


Figure 3 Feature level Behavioural Diagram

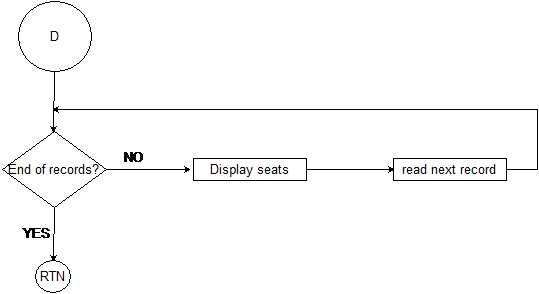


Figure 4 Feature level Behavioural Diagram

## Test Plan

### High Level Test Plan

| **Test Id** | **Description** | **Expected Output** | **Actual Output** | **Type of test** |
| --- | --- | --- | --- | --- |
| H\_01 | Make a reservation with passport no 45521 | Successful and seat number | Successful with seat number A\_1 | Requirment based |
| h\_02 | Make a reservation with already entered passport number | Seat number A\_2 | Sat number A-2 | Scenario based |
| H-03 | Enter a 9 digit passport number | Successful | Successful | Requirement based |
| H\_04 | Delete reservation after entering passport number | Booking Deleted | Booking Deleted | Requirment based |
| H\_05 | Select Delete and Enter 0 as passport number | Passport Number is wrong | Passport Number is wrong | Scenario based |
| H\_06 | Select Display record enter A\_1 | Successfully print all details of A\_1 | Successfully print all details of A\_1 | Requirment based |

### Low Level Test Plan

| **Test Id** | **Description** | **Expected Output** | **Actual Output** | **Type of test** |
| --- | --- | --- | --- | --- |
| L\_01 | Select Reservation and enter details simultaneously | Successful and seat number | Successful and seat number | Scenario based |
| L\_02 | Select delete and enter invalid passport number | Passport number is wrong | Passport number is wrong | Scenario based |
| L\_03 | Select display without any reservations | Back to option select window | Back to option select window | Scenario based |
| L\_04 | Enter invalid choice from select menu | Sorry Invalid choice | Sorry Invalid choice | Scenario based |

## Implementation and Summary

### Git Link:

Link: <https://github.com/vinayaranade/M1_Application_Airlinebookingsystem.git>

# Mini project 2 – Embedded Calculator [Individual]

## Modules

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

### Requirements

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

**Why:**

1. The main objective of this project is to perform basic mathematical calculations such as addition, multiplication, subtraction and division with the use of Microcontroller, Key pad and LCD .

**Where:**

1. It can be implemented in any places like our Houses, Institutions, Banks etc

**Who:**

1. The project also exhibits low cost system which is widely needed in our daily life.
2. The system is user friendly and the cost of the system is low it can be used by everyone in their everyday life.

**When:**

1. The use of this calculator is necessary in everyday life and digital calculator is beneficial to provide automated mistake free and automated mathematical solutions to the users.

**How:**

1. This design of calculator using an Atmega328 microcontroller can be used to perform basic mathematical operations such as addition, subtraction, multiplication and division and display the results on a LCD screen.

### High Level Requirements

| **ID** | **Description** | **Status** |
| --- | --- | --- |
| HLR 1 | System should display enter expression. | Implemented |
| HLR 2 | It should display the digits entered by user | Implemented |
| HLR 3 | It should perform basic operation and display the result on LCD Screen . | Implemented |

### Low Level Requirements

| **ID** | **Description** | **Status** |
| --- | --- | --- |
| LLR 1 | Take input from user from the keypad input | Implemented |
| LLR 2 | Take the operation to be performed from the user | Implemented |
| LLR 3 | Perform the operation and display result on 16 x 2 LCD | Implemented |

## Design

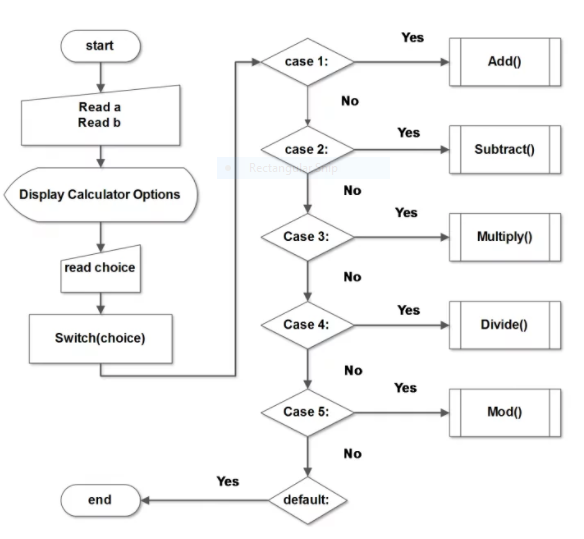


Figure 5 Behavioural Diagram

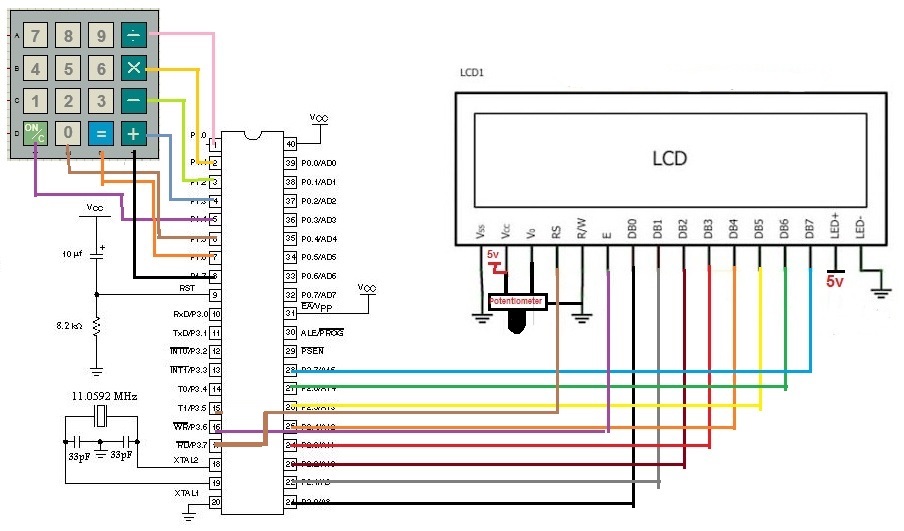


Figure 6. Circuit Diagram

### Git Link:

Link: <https://github.com/vinayaranade/M2-Embedded_Calculator.git>

# Miniproject 3 – Diary Entry System [Team]

## Modules

1. SDLC
2. Git

### Requirements

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

**Why:**

* Makes a difference the client to effectively include their imperative meetings, presentation records, additionally can be altered.

**Where:**

* It can be used by Travel Specialists or doctors in fact anybody can utilize it to keep their records safe.

**Who**

* Anybody can use it.

**When:**

* At whatever point the client needs to keep his individual records secure at a place.

**How:**

* It will keep your all-individual records securely at one place.

### High Level Requirements

| **ID** | **Description** | **Implementation** |
| --- | --- | --- |
| HLR1 | Add the inputs to add records, edit, view and delete. | Implemented |
| HLR2 | Users can add the record in the system. | Implemented |
| HLR3 | Can view that record for further. | Implemented |
| HLR4 | User can Edit the added record and can make some changes in it. | Implemented |
| HLR5 | Can delete the record permanently if not needed. | Implemented |
| HLR6 | User can edit the Password for security purpose. | Implemented |

### Low Level Requirements

| **ID** | **Description** |  | **Implementation** |
| --- | --- | --- | --- |
| LLR\_1 | Login page of Diary Entry system. |  | Implemented |
| LLR\_2 | The system will ask password to view and Edit the records. |  | Implemented |
| LLR\_3 | Edit data. |  | Implemented |
| LLR\_4 | enter username and password. |  | Implemented |
| LLR\_5 | Newly added details should be recorded. |  | Implemented |

## Design

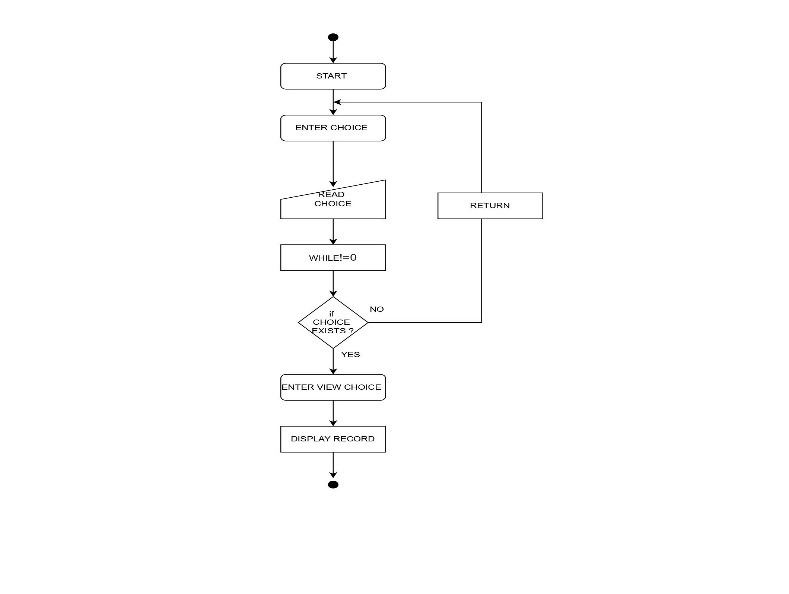


Figure 10.Behavirioul Diagram

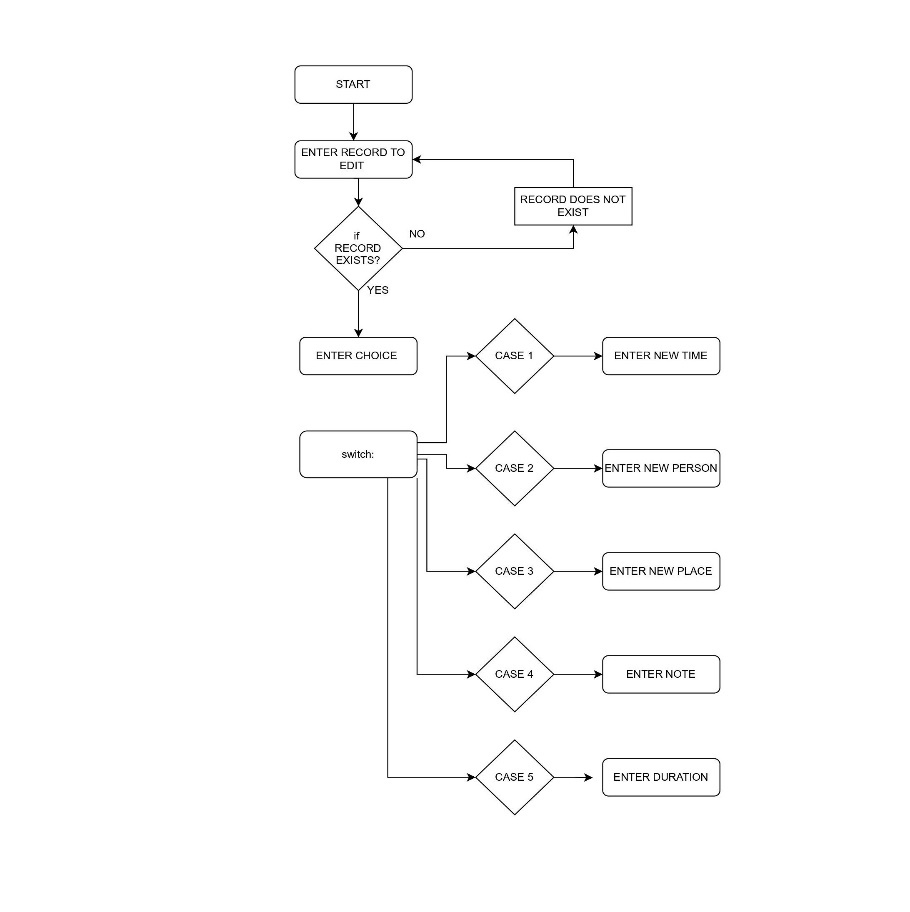


Figure 11. Behavirioul diagram

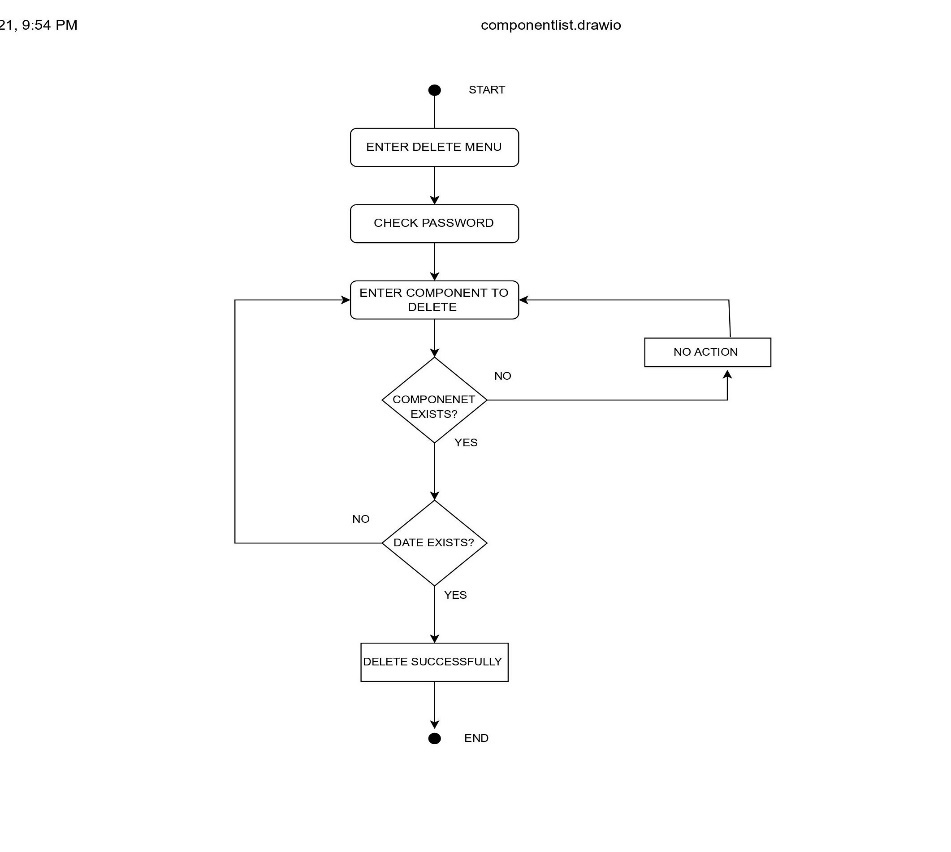


Figure 12 Delete Record Feature Diagram

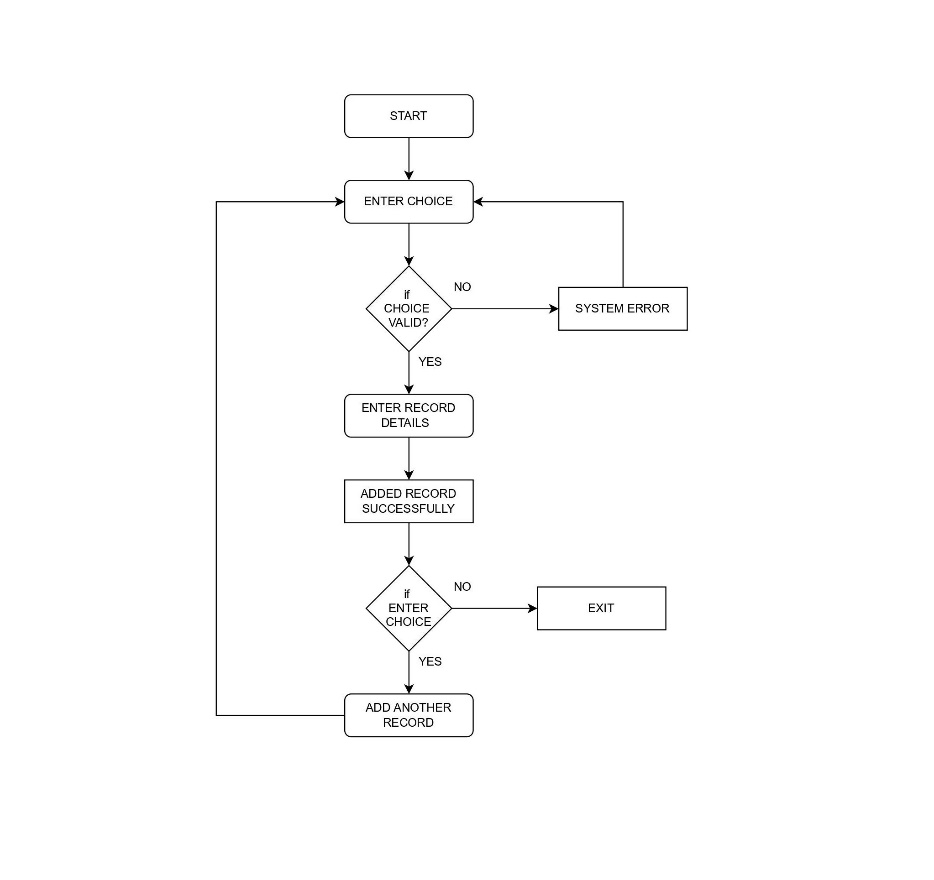
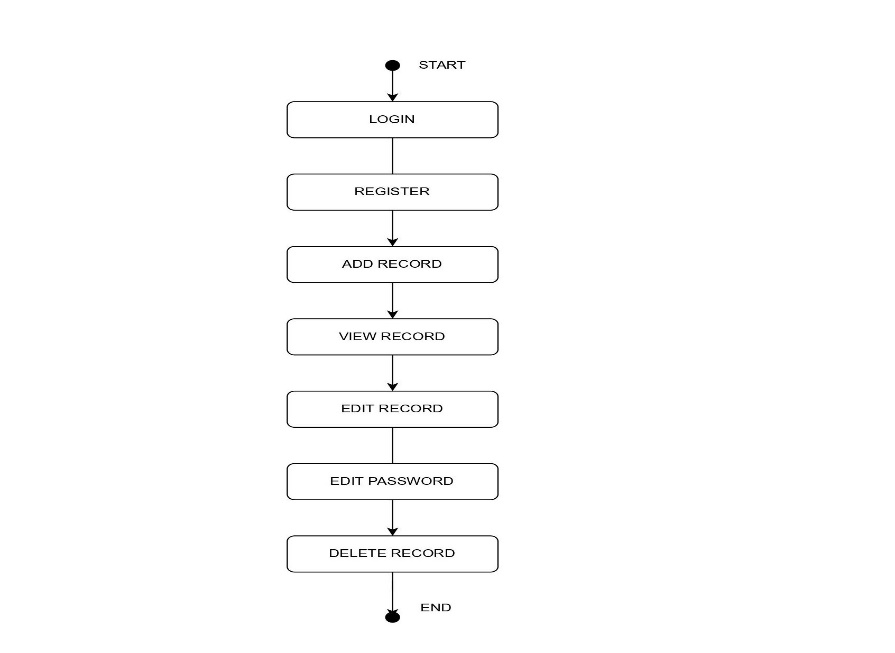


Figure 13 .Add Record feature level diagram



## Figure 14. Component structure diagram

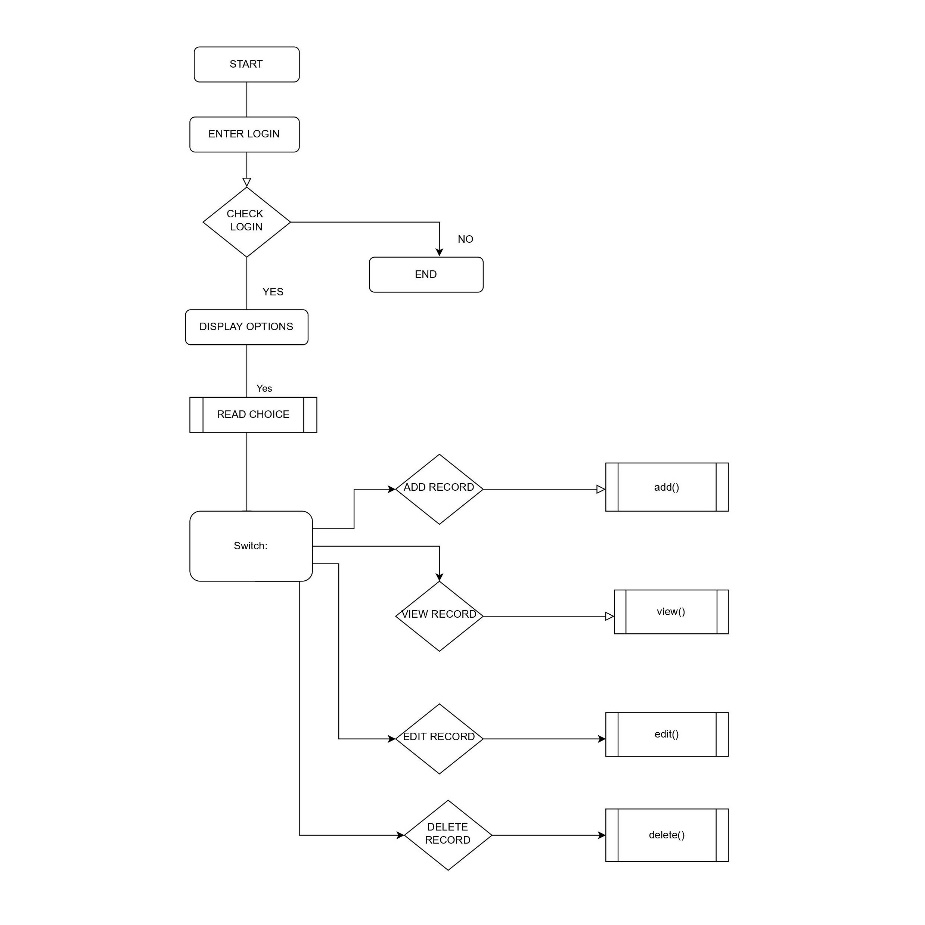
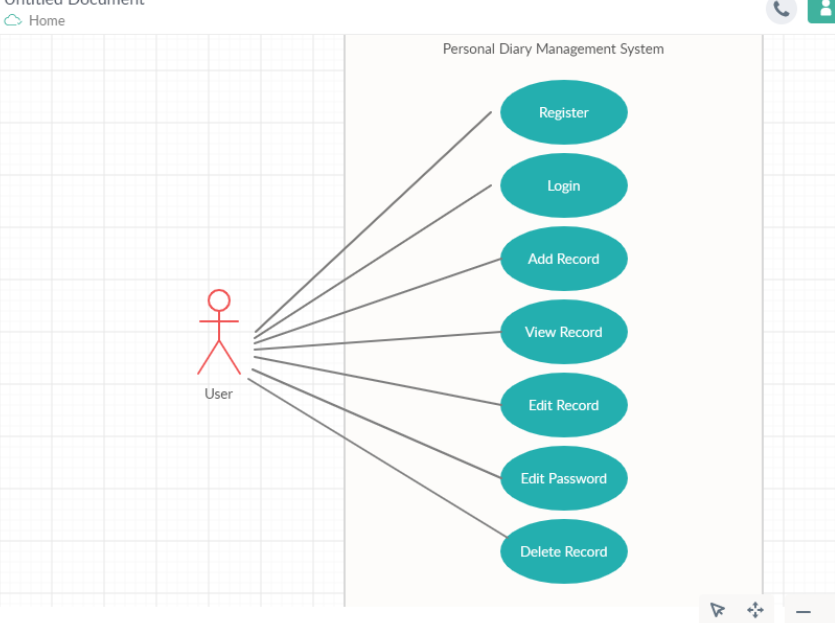


Figure 15. Flow structure diagram



## Figure 16. Case diagram

## Test Plan

### High Level Test Plan

| **Test ID** | **Description** | **Expected Output** | **Actual Output** | **Pass/fail(Result)** |
| --- | --- | --- | --- | --- |
| H\_01 | Check if the record is viewed or not | SUCCESS | SUCCESS | PASS |
| H\_02 | Check if the record information is added or not | SUCCESSS | SUCCESS | PASS |
| H\_03 | Check if the record is edited | SUCCESS | SUCCESS | PASS |
| H\_04 | Check if the password is edited or modified | SUCCESSS | SUCCESS | PASS |
| H\_05 | Check if the record is deleted or not | SUCCESS | SUCCESS | PASS |

### Low Level Test Plan

| **Test ID** | **Description** | **Exp IN** | **Exp OUT** | **Actual Out** |
| --- | --- | --- | --- | --- |
| L\_01 | Check if record information is properly added | Data and information | SUCCESS | SUCCESS |
| L\_02 | if the data is collected from diary during when the user needed | Datas | SUCCESS | SUCCESS |
| L\_03 | If the record data is delete | Diary datas | SUCCESS | SUCCESS |

## Implementation and Summary

### Git Link:

Link: <https://github.com/GENESIS2021Q1/Applied_SDLC-Dec_Team_49.git>

### Individual Contribution and Highlights

Role in Project Team

1. Worked on project ideas and research of project ideas by other members

1. Helped in the implementation and research of features
2. Worked on UML diagrams
3. Added High level and low level diagrams for the Design part
4. Worked on modifications of Design according to new suggesstions and worked on report

# Miniproject 4 – Calendar Automation [Team] [Team]

## Modules

1. Python
2. Git

### Requirements

### High Level Requirements

| **ID** | **Feature** | **MATLAB v0 Status** | **Python v0 Status** |
| --- | --- | --- | --- |
| HR01 | GUI | Implemented | Implemented |
| HR02 | Master Calender | Implemented | Implemented |
| HR03 | Faculty calender | Implemented | Implemented |
| HR04 | Faculty load sheet | Implemented | Implemented |
| HR05 | Showing Available Open Slots based on faculty and modules | Not Available | Not Available |
| HR06 | Output file generated across different computers (windows + linux) | Not Available | Implemented |
| HR07 | Visualizing data to create Meaningful Insights | Not Available | Not Available |
| HR08 | Calculate Individual Faculty Load | Implemented | Implemented |

### Low Level Requirements

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

# Link for template standard input template:

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

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

## Requirements for updating Master calendar using Master calendar as input

# Link for template

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

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

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

## App deployment

* The app is deployed on heroku servers.
* To add/modify new features, you will be required to install HEROKU CLI [link](https://devcenter.heroku.com/articles/getting-started-with-python#set-up)
* After installation, open terminal in working directory and enter the following commands:
  + "heroku git:clone -a gea calendar"
  + login using heroku credentials
* After pulling and making changes, enter the following commands to push app and deploy on server
  + Git add.
  + git commit -m "commit message"
  + git push heroku master

### Additional features for V1 to do

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

### Git Link:

Link: <https://github.com/tlnsnani/OopsWithPython_Calendar_Automation_Team-48.git>

# Miniproject 5 – Team BMW [Team]

## Modules

1. Matlab
2. Git

**Module: - Applied Model Based Design Module**

**Individual Topic:** ALARM CONTROL SYSTEM(BMW)

**Introduction**

An alarm system is necessary in all car and automobiles for protection against theft. The alarm control systems need to be incorporated as an integral part of vehicle electronics. The alarm control systems use electric sirens and generate sound when triggered. The alarms are triggered when entry point is reached, they are turned on when doors are opened and switch off automatically after a few minutes. The alarm system consists of two parts control unit and a siren part. The system consists of flash lights which draw current of about 5 A and are turned on when the system is triggered.

Three types of intruder alarms are used in the control system

1.Switch operated

2.Battery voltage sensed

3.Volumetric sensed

**Overview**

An automobile alarm system that protects vehicles against window breaks and door openings and has wireless transmitters for enabling and disabling the alarm system, with a unique and less obnoxious siren sound than current ones. The current system has frequency with which traditional car alarm systems sound, they are ineffectual and inefficient, providing little protection or security to vehicle owners. The primary goal of this feature is to improve user situational awareness during vehicle security threats and provide a system built inside the automobile.

**System Analysis**

The system needs to be able to work with low power consumption while providing high reliability and fast response time. The microcontroller must manage efficiently the turning on and off of the wireless communication when not needed to save power. The system also needs to be able to run off of battery back-up in case the battery is unplugged by the thief. We will test the time required for the microcontroller to accomplish both easy and difficult tasks under varied supply voltages to determine the effect on performance.

**Design**

**Structural level diagram**

Flowchart

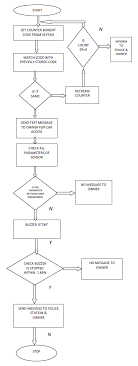


Figure 17. Structural Diagram

**Block diagram**

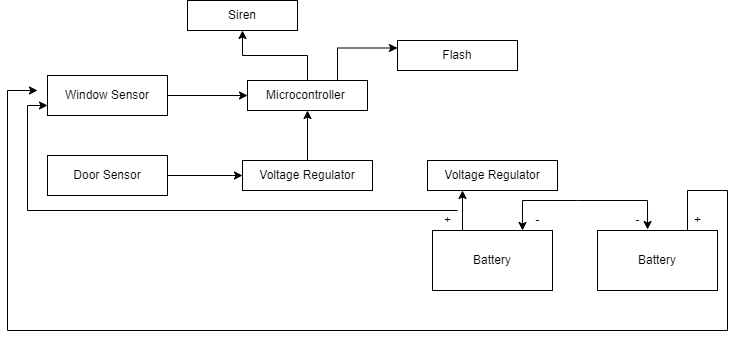


Figure 18. Block diagram

**Requirements**

**High Level Requirements**

|  |  |
| --- | --- |
| ID | Description |
| HLR\_1 | The alarm system must transition from disabled to enabled |
| HLR\_2 | The door open must be detected by the system and siren should be triggered |
| HLR\_3 | The flash should turn on when the system is triggered that is door is opened |
| HLR\_4 | The siren should turn off automatically after few minutes |
| HLR\_5 | System should detect the window open and trigger alarm |
| HLR\_6 | The alarm and sensors are turned off when door is shut. |

**Low-Level Requirements**

|  |  |
| --- | --- |
| ID | Description |
|  |  |
| LLR\_1 | The automatic lock system must get activated when system is triggered |
| LLR\_2 | Indication of open door with slow and flash light |
| LLR\_3 | Alarm disarms when doors are closed |
| LLR\_4 | System transition from enabled to disabled when no input from sensor |

**Modeling**

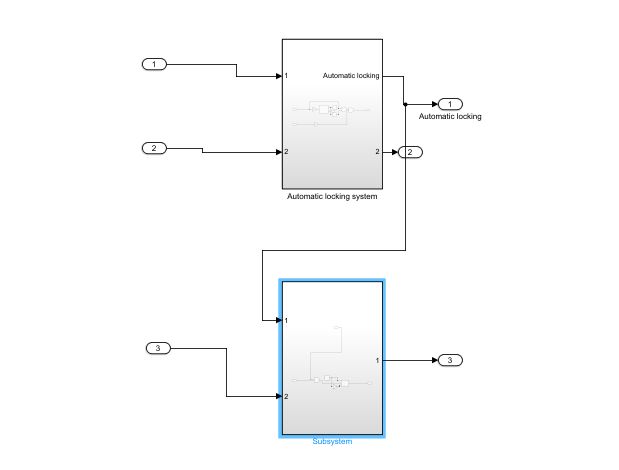


Figure 20.Model design

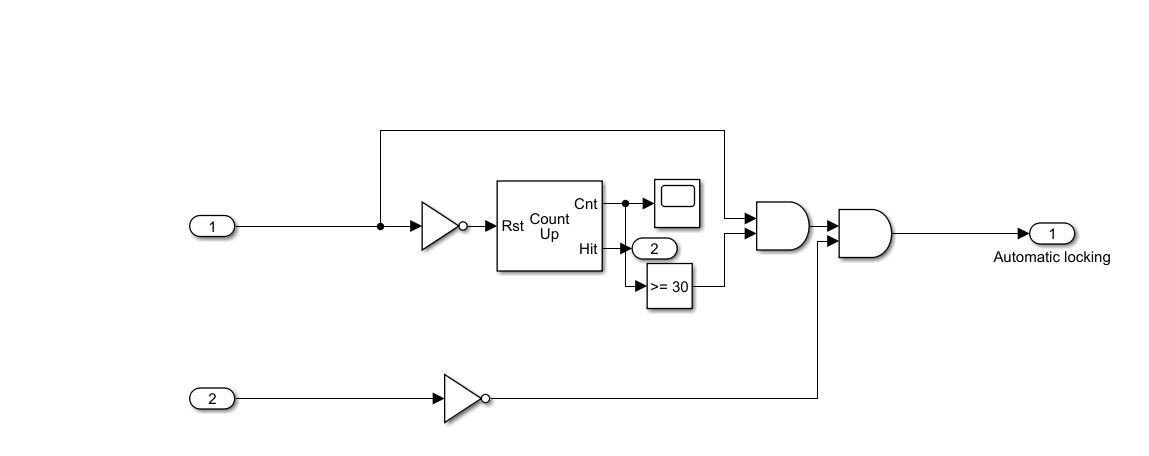


Figure 21. Model Design

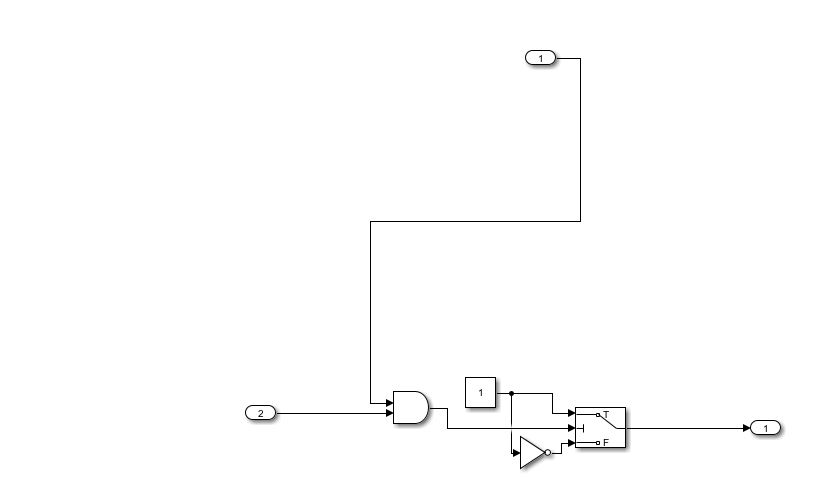


Figure 22. Model Simulation

**Conclusion**

The primary goal of ensuring the safety of the car owner has been met. Window breaks, door opens, and starting the car are all protected by the alarm system. The horn is only designed to go for a few minutes, so it accomplishes the second purpose of limiting noise to a minimum. A car alarm system may not prevent injuries, but it does give instant protection against loss of property by deterring car thefts. AN auto alarm system, on the other hand, provides indirect safety to a neighbourhood. Car alarms discourage thieves from breaking into and stealing vehicles.

**Merging with BMW (BCM MODULE)**

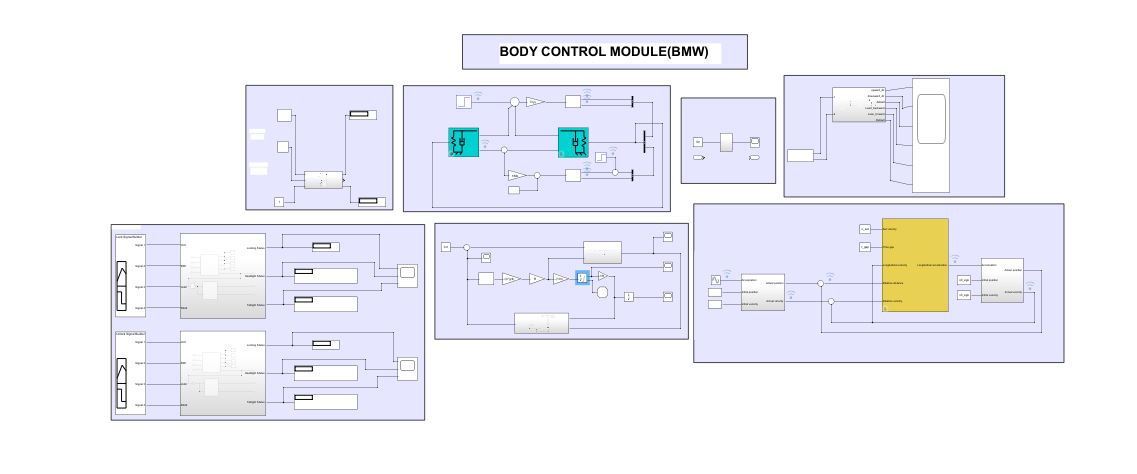


Figure 23. Model Simulation

# Miniproject 6 – Wiper Control[Team]

## Modules

1. C Programming
2. STM32

## Requirements

**High Level Requirements**

A windscreen wiper or windshield wiper is a device used to remove rain and dirt from a windscreen. Almost every motor vehicle, including trains, planes, and boats, has wipers, which are almost always required. The wiper control system contains a controller that takes in the input from the user and controls the operation of the wiper based on those inputs.

### Low Level Requirements

The aim of this project is to help reduce accidents that happen as a result of the driver intending to clean the front screen thereby taking the attention of the driver off the road. This is also known as wiper delay module. The wiper control system checks for the ignition key position and if user button is pushed for more than 2 sec the wiper system is on indicated by red Led on similarly for ignition key position at lock wiper is off and red led off.

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

**Who:**

The system is beneficial for the common economic car manufactures as it provides an efficient and reliable system for wiper control at a reduced cost.

**What:**

Drivers are exposed to an ever-increasing number of distractions. Wiper control system become an appealing feature as they minimize the distraction by providing a clear front view

**When:**

The need for wiper control system was realized since the automotive industry has been researching ways to exploit modern computing and electronic advances in the development of safety, reliability, and entertainment technologies.

**Where:**

Wiper control system is used to remove rain and dirt from windscreen and is required in every motor vehicle such as train, plane, boat and automobiles.

**How:**

The wiper control system checks for the ignition key position and if user button is pushed for more than 2 sec the wiper system is on indicated by red Led on similarly for ignition key position at lock wiper is off and red led off.

**SWOT Analysis**

Strengths:

* The present automobiles have a large number of features with high equipped technology but the cost of these systems is very high
* Thus, an effort has been made in this this wiper control system to reduce the cost so that it can be implemented in the common economic cars where a common man can enjoy the benefits.
* It helps the driver to concentrate on the road while driving by providing a clean front view hence allowing the driver to focus on the road and driving.

Weakness:

* The position of wiper needs to be operated manually.
* The driver may feel the need of wiper but may not be able to turn it due to his or her attention being busy in traffic or other factors.
* The speed of wipers needs to be adjusted manually by the user according to his or her need.

Opportunities:

* This wiper system makes wiper operation easier and more comfortable for the driver.
* It will provide a new level of comfort and assistance to drivers who work at night or in congested regions, where they already have to focus on brakes and clutches.

Threats:

* This system allows only a few intermittent or variable speed operations.
* It requires driver’s constant attention to wiper speed and adjustment.
* The manual adjustment of the wiper distracts driver's attention, which may be a direct cause accident.

## Design

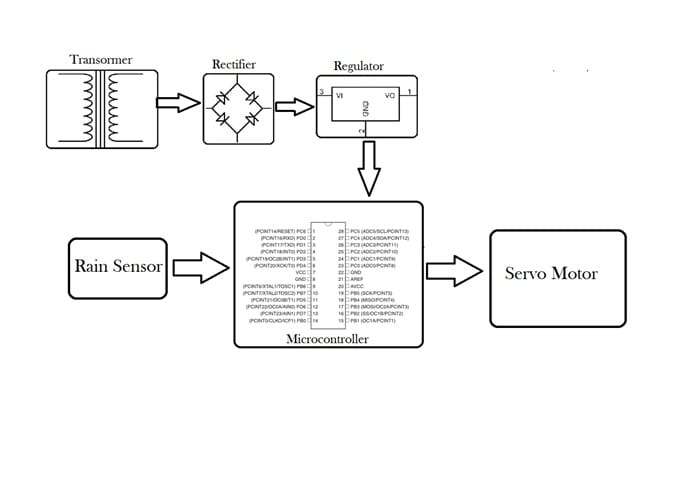


Figure 24 Block Diagram

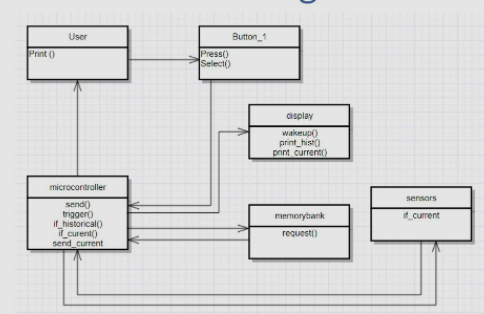


Figure 26. Class Diagram

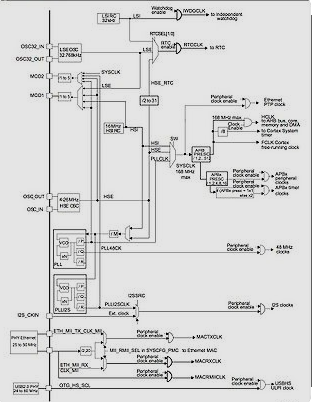


Figure 27. Deployment Diagram

## Test Plan

### High Level Test Plan

| **ID** | **Description** | **Expected O/P** | **Actual O/P** | **Type of Test** |
| --- | --- | --- | --- | --- |
| H\_01 | Detect Raindrops/snow | Detecting | SUCCESS | Requirement |
| H\_02 | Send Signal to Wiper Motor | Motor ON | SUCCESS | Scenario |
| H\_03 | If Rain sensor got failure it won't send SIGNAL | failed | Work processing | Future |
| H\_04 | sensor failure | failed | Work processing | Future |
| H\_05 | If Its Heavy rainy Wiper control should Work every 3 min Activate Timer to wipe frequently | Automation | SUCCESS | Future |
| H\_06 | Process Off | stop | SUCCESS | Implemented |

### Low Level Test Plan

| **ID** | **Description** | **Expected O/P** | **Actual O/P** | **Type of Test** |
| --- | --- | --- | --- | --- |
| L\_01 | Identify and sense Raindrops/Snow on windshield | Rain sensor Detected | SUCCESS | Requirement |
| L\_02 | Wiper Motor ON | Wiper Controller starts wiping windshield | SUCCESS | Scenario |
| L\_03 | When No Rain Motor should OFF | Detecting Raindrops on shield | SUCCESS | Scenario |
| L\_04 | No rain | sensor off | SUCCESS | Scenario |
| L\_05 | While implementing Wiper controller should set for particular time, It shold be adjustable | Heart beat rate high/low | SUCCESS | Scenario |
| L\_06 | if every sensor activate and results Not found | Automatically stop vehicle | SUCCESS | Future Implementation |

## Implementation and Summary

### Git Link:

Link: <https://github.com/GENESIS-2022/MasteringMCU-Team65.git>

**Individual Contribution and Highlights**

1. Requirements of Wiper System
2. Design and implementation

# Miniproject 7 – Ford Project [Team]

## Modules

1. Automotive Systems
2. Git

### Requirements

# Ford Aspire

The Ford Aspire nameplate has been used by the American automobile manufacturer Ford for the following cars, in the following markets: Ford Festiva, in North America from 1993 to 1997. The sedan version of the Ford Figo, a rebadged third generation Ford Ka in India since 2015.

# Body Control Module

## Features:

* Door Lock System
* Interior Light Control
* Power Mirror
* Power Window

**Individual Feature**: **- Door Lock System**

One of the features that most new entries into the Indian vehicle industry have is locking. New cars are typically equipped with a power-operated locking mechanism that is linked to the vehicle's alarm system. Each automobile key includes a unique code that synchronises with the vehicle's alarm system. The engine immobiliser is turned off after the automobile is locked using the key. The system will be implemented back up only when the same code is received from the car key.

The locking system in a vehicle must grant access only to authorised persons. It is the means via which the vehicle doors and boot lid are locked and unlocked and the engine is started.The locking system is operated with a key or remote control. Keyless systems are increasingly being used to start engines. In a keyless system, a transmitter – which usually also houses the controller for the central locking – is inserted into a reader in the vehicle and the engine is then started by pressing a button. A more recent development has seen the use of systems that work without any contact at all.

## 4W's and 1H

### What :

The most important advantage of having a central locking system in your vehicle is the increased security it provides. The mechanism allows you to quickly lock all doors by operating the driver's lock. There are no risks associated with leaving the door unlocked. Small children will be safe in the back seat because the rear doors can be easily locked.

### Who :

The door lock system provides the advantage to the user for high security . It can be used to locate the car in a huge parking lot that houses hundreds of vehicles. The remote of the central locking system basically works on radio waves that are transmitted to the device fitted in the car.

### Where :

The central locking system may not be a standard feature in your car, but it might be offered as an optional one. One can easily fit this system in their cars from workshops and they aren’t expensive or difficult to install.

### When :

Until the 1960s, some ford cars still had different keys for the doors and the ignition. A combined key for all purposes, the precursor of today’s locking systems, gained popularity as the decade wore on. There are several types of central locking systems available today. It is possible to use a remote from a distance and operate the locks of the car. Some of the remotes can also set off the alarm of the vehicle.

### How :

To secure the automobile from the inside, the driver simply locks one of the front doors with the key, and all of the doors automatically lock. By manually locking or unlocking the driver's side door lock or pushing the lock and unlock buttons on the key fob or dashboard, the driver can lock or unlock all doors.

## Design

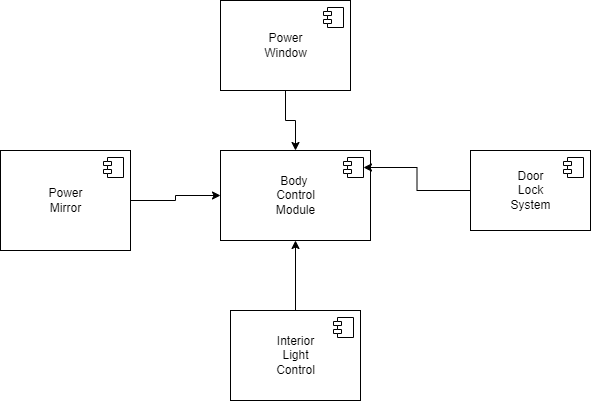


Figure 28. Body Control Module

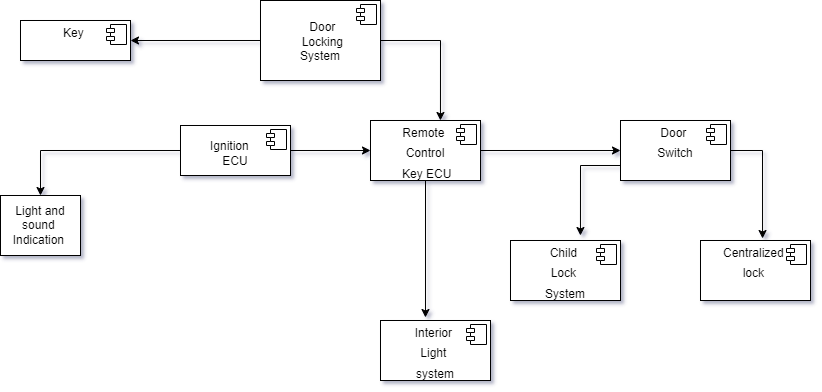


Figure 29. Structure Diagram

## Implementation and Summary

### Git Link:

Link: <https://github.com/Pradnya579/Automotive_Ford_Team.git>

### Individual Contribution and Highlights

1. Door Lock System

Role in Project Team

1. Designer: Done Designing for Project
2. Researcher: Done case study for Body Control Module of Ford.

# Mini project 8 – EV Bike[Team]

## Modules

1. Matlab
2. Matlab Script

### Requirements

Requirements

**1.Battery:**

1.Battery type used in this EV is Lithium-ion Polymer.

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

**Battery Features:**

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

**2.Motor:**

1.BLDC motors are used, which have traction characteristics like high starting torque, high efficiency around 95 98%,etc.

2.BLDC motors are suitable for high power density design approach. The BLDC motors are the most preferred motors for the electric vehicle application due to its traction characteristics.

**Motor Features:**

3.Electronically commutated

4.High energy, rare earth magnets used for rotor field

5.Requires speed control with 6-lead connection (3 power,2 Hall Effects, 1 drain)

6.Rated speed 2500 RPM; minimum 150-200 RPM

7.Linear speed torque curves

8.Built-in tach pulse for economical speed readout

9.Encoder options for servo performance.

**3.Controller:**

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

**Controller Features**:

1.E- Braking will release the motor from the controller as long as brake is applied.

2.Over current protection to protect controller during faulty conditions or short circuit.

3.Low voltage detection ensures Battery life.

4.Pedal assist mode controls the motor speed based on the speed of peddling.

5.Accelerator fault protection to prevent runway.

6.Provision for 120°/ 60° Selection.

7.Speedometer output.

8.Brake inputs with high active and low active provision.

9.Speed limit control provision.

**4.Inverter :**

1.The traction inverter converts energy from the vehicle's battery in order to drive the motors in the drivetrain. This key component has a direct impact on road performance, driving range and reliability of the vehicle also as a consequence of their weight and size.

2.Subject to all the possible stress found in a road vehicle from heat and vibrations, these converters must be able to handle high power and currents along with the associated Electro Magnetic Compatibility (EMC) challenges as well as provide fail-safe operation to ensure dependability and safety for the driver and passengers.

3.To help developers increase the automotive inverter's power efficiency and reduce size and weight, ST has a wide offer of discrete semiconductors including AEC-Q101 qualified IGBTs and both silicon and silicon-carbide (SiC) MOSFETs and diodes, AEC-Q100 qualified galvanically isolated IGBT and MOSFET gate drivers and SPC5 32-bit automotive microcontrollers for designing scalable, cost-effective and energy-efficient EV traction inverter solutions.

**Inverter features :**

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

**Analysis**

| **Specification** | **MG ZS EV** | **Hyundai Kona Electric** |
| --- | --- | --- |
| Model Price | 21.08 lakh Rs | 23.79 lakh Rs |
| Gross combination weight | 1609 kg | 2020 kg |
| Range | 419 km/Charge | 452 Km/Charge |
| Battery type | lithium-ion | Lithium-ion Polymer |
| Max torque | 350Nm | 395 Nm |
| Power | 140.8bhp | 134.14bhp |
| Charging time | 18-19 hrs | 7 h 30 min |
| Electric range | 340 km | 305 km |
| Wheel base | 2585 mm | 2600 mm |

**Research**

Model 1 MG ZS EV

Specifications

• Model Price- 21.08 lakh Rs

• Gross combination weight -1609 kg

• Range - Certified range of 419 km

• Battery type used -lithium-ion

• Battery - 44.5 kWh 394 V lithium-ion

• Max torque - 350Nm@5000rpm

• Power – 140.8bhp

• Charging time - 18-19 hrs for fully charge

• Drive line- ZS EV is front wheel drive and can accelerate from 0 to 62 miles per hour in 8.2 seconds

• Motor- Permanent Magnet Synchronous Electric Motor (EV)

• Engine -72kWh battery and single e-motor ,154bhp,206lb ft

• Transmission -5-speed manual 4-speed automatic 6-speed automatic 6-speed DCT CVT e-CVT (EV)

• Electric range -211 mi (340 km) (claimed) 163 mi (262 km) (WLTP)

• Rear suspension -Torsion beam

• Front suspension -MacPherson Strut

• Wheel base – 2585 mm

Hyundai Kona Electric

Specifications

• Model price – 23.79 lakh Rs

• Gross combination weight – 2020 kg

• Range - 452 Km/Charge

• Battery type used – Lithium-ion Polymer

• Battery - 39.2 kWh 327 V lithium polymer

• Max torque - 395 Nm (40.27 Kgm)

• Power - 134.14bhp

• Charging time - 7 h 30 min for full charge

• Drive line – trim has front-wheel drive and the same powertrain

• Motor - Permanent Magnet Synchronous Motor (PMSM)

• Engine – a single-speed transmission and a 201-horsepower 150-kW electric motor.

• Transmission - a single-speed transmission and Automatic Single Speed Reduction Gear

• Electric range - Maximum range 305 km for the 39.2 kWh battery version

• Rear suspension - McPherson Strut

• Front suspension - multi-Link

• Wheel base – 2600 mm

Model 3 EV car

• Model weight – 1700-1800 kg

• Range – 450-490 Kg/Charge

• Battery type used – Lithium-ion Polymer

• Battery -44.5 kWh

• Max torque - 395 Nm

• Charging time - 7 h 30 min for full charge

• Motor- Permanent Magnet Synchronous Electric Motor (EV)

• Rear suspension -Torsion beam

• Front suspension -MacPherson Strut

• Wheel base – 2585 mm

## 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 9 – Door Locking System[Individual]

## Modules

1. Autosar
2. Git

### Requirements

### High level Requirement

| **Requirement Id** | **Description** |
| --- | --- |
| HLR\_1 | Remote control locks are used Instead of using keys to control car locks & components |
| HLR\_2 | Door locks & components inside the vehicle authorize access to owners only |
| HLR\_3 | When the velocity is 0 and the battery is on their will be an indication without any sound |
| HLR\_4 | When the car is in motion the door lock should be firmly closed |
| HLR\_5 | The door open indication will be off only when all seat belts are locked . |
| HLR\_6 | If the car is in motion and any door is open the system will display an indication |

### Low level Requirement

| **Requirement Id** | **Description** |
| --- | --- |
| LLR\_1 | When unauthorized person tries to get into your vehicle by breaking the door their will be a beeping sound |
| LLR\_1 | Beeping sound will stop when the doors are closed. |
| LLR\_2 | When the doors are open there will be individual light indication. |
| LLR\_3 | When the boot is open there will be indication on the dashboard. |

## Design

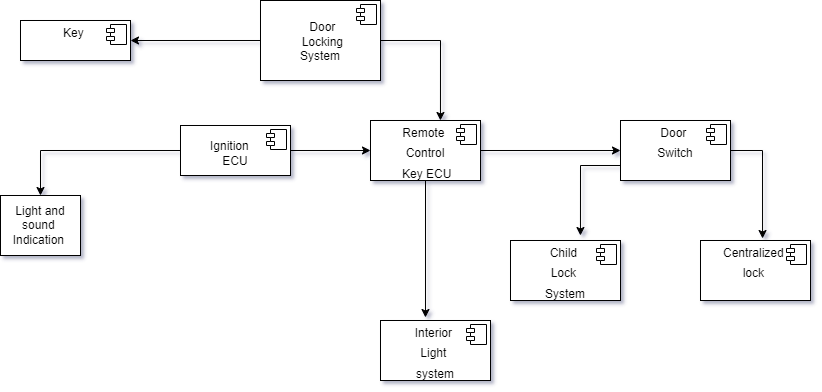


Figure 30. VFB Diagram

## Implementation and Summary

### Git Link:

Link: <https://github.com/Pradnya579/Automotive_Ford_Team.git>

### Individual Contribution and Highlights

1. Door Locking system
2. Source code management using GitHub
3. AtomicSwComponent
4. SWCInternalBehavior
5. SWCImplementation