./

GENESIS - Learning Outcome & Mini-project Summary Report



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **To be Approved** | **Remarks/Revision Details** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Details**

Contents

[Contents 3](#_Toc55470819)

[Miniproject -1 [Team/Individual] 4](#_Toc55470820)

[Module/s 4](#_Toc55470821)

[Topic and Subtopics 4](#_Toc55470822)

[Objectives & Requirements 4](#_Toc55470823)

[Design 4](#_Toc55470824)

[Test Plan 4](#_Toc55470825)

[Implementation Summary 4](#_Toc55470826)

[Video Summary 4](#_Toc55470827)

[Git Link 4](#_Toc55470828)

[Git Dashboard 4](#_Toc55470829)

[Summary 4](#_Toc55470830)

[Individual Contribution & Highlights 5](#_Toc55470831)

[Summary 5](#_Toc55470832)

[Challenges faced and how were they overcome 5](#_Toc55470833)

[Future Scope (If applicable) 5](#_Toc55470834)

[Miniproject -2 [Team/Individual] 6](#_Toc55470835)

[Module/s 6](#_Toc55470836)

[Topic and Subtopics 6](#_Toc55470837)

[Objectives & Requirements 6](#_Toc55470838)

[Design 6](#_Toc55470839)

[Test Plan 6](#_Toc55470840)

[Implementation Summary 6](#_Toc55470841)

[Git Link 6](#_Toc55470842)

[Git Dashboard 6](#_Toc55470843)

[Summary 6](#_Toc55470844)

[Individual Contribution & Highlights 6](#_Toc55470845)

[Summary 6](#_Toc55470846)

[Challenges faced and how were they overcome 6](#_Toc55470847)

# Mini Project -1 [Individual]

## **Module/s**

“Modules linked to the Mini project are Linux, C++ and SDLC”

### Topic and Subtopics

* **Core topics implemented**:

1. Classes and Objects

2. Inheritance

3. Vectors and list

4. threads

* **Sub topics** **implemented**

1. strings

2. constructors and destructors etc.

* **Linux Concepts used:**

1. Make files for the execution of the project.
2. Static and Dynamic libraries for the faster execution and quicker run time.

## Objectives & Requirements

**Objective**: Smart Library Management System comprising of Books Management, Members Management, Store Management and Utilities Services.

**Requirements:**

**High Level Requirements:**

|  |  |
| --- | --- |
| ID | Description |
| H1 | Management of Book Information |
| H2 | Management of Library Member Information |
| H3 | Store Management |
| H4 | Provide library Utilities to Members |

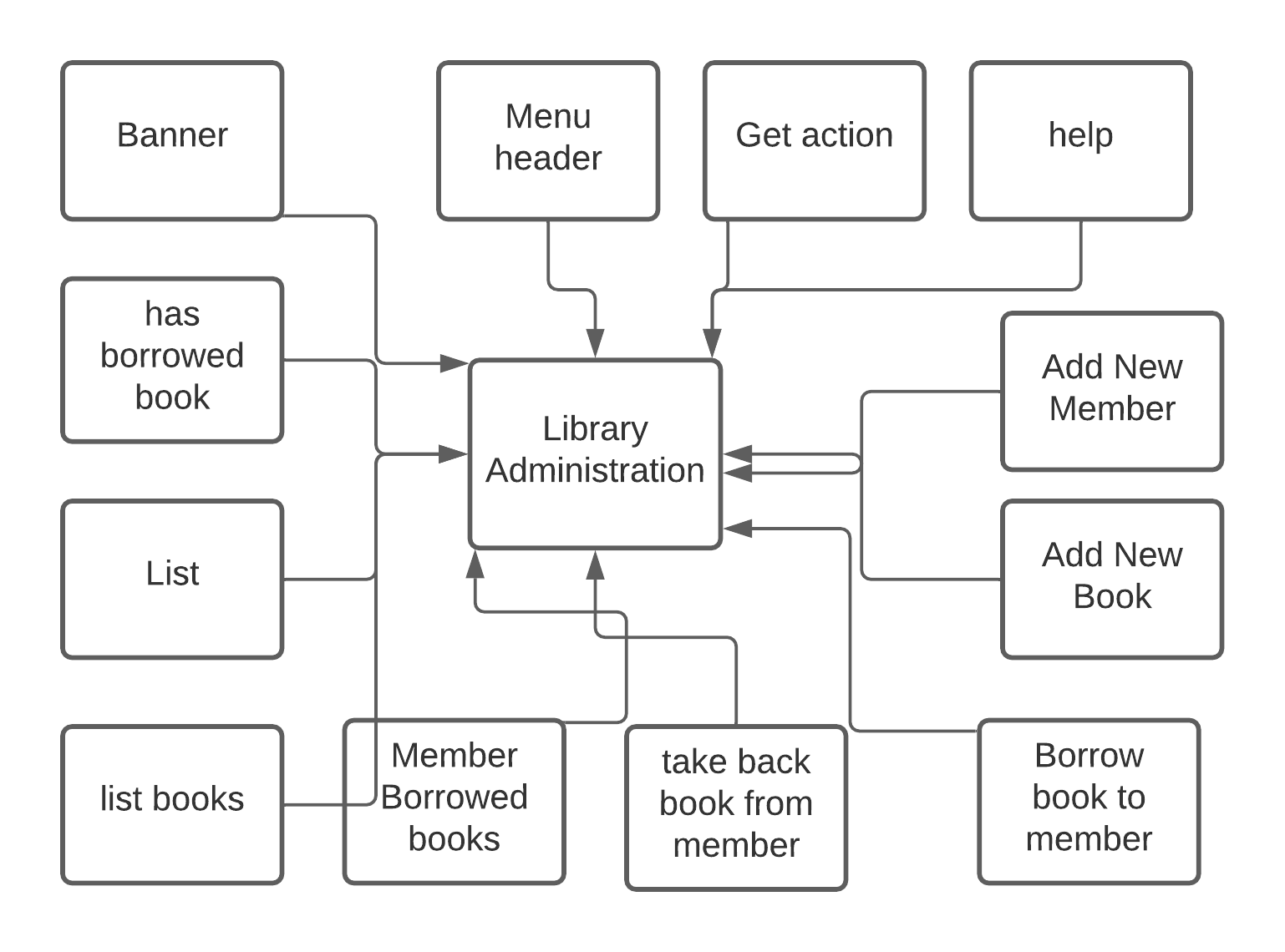
**Low Level Requirements:**

|  |  |
| --- | --- |
| ID | Description |
| L1 | Get the information about book |
| L2 | List out books which can be borrowed to member. |
| L3 | Set the information about book. |
| L4 | Get Member information |
| L5 | Set Member Information |
| L6 | Get Member borrowed books count |
| L7 | Check whether Member has borrowed books |
| L8 | Find Member by id |
| L9 | Print New Member information |
| L10 | Print Borrowed book information |

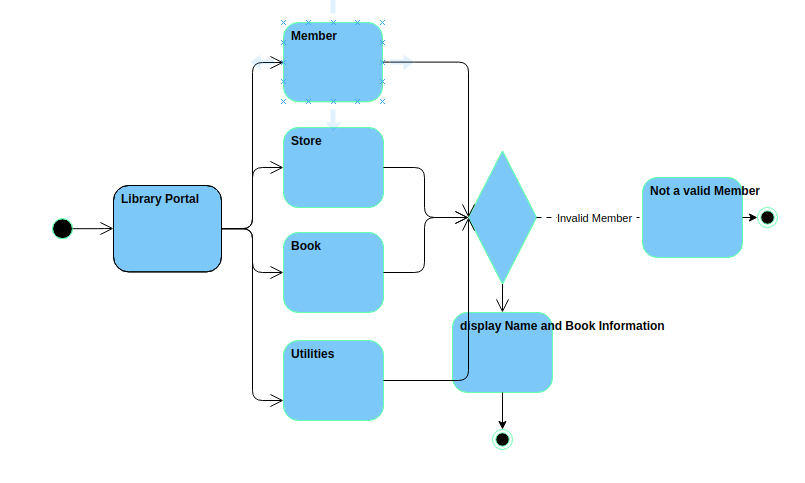
## **Design**

Behavioral Diagram:

1. Component Diagram

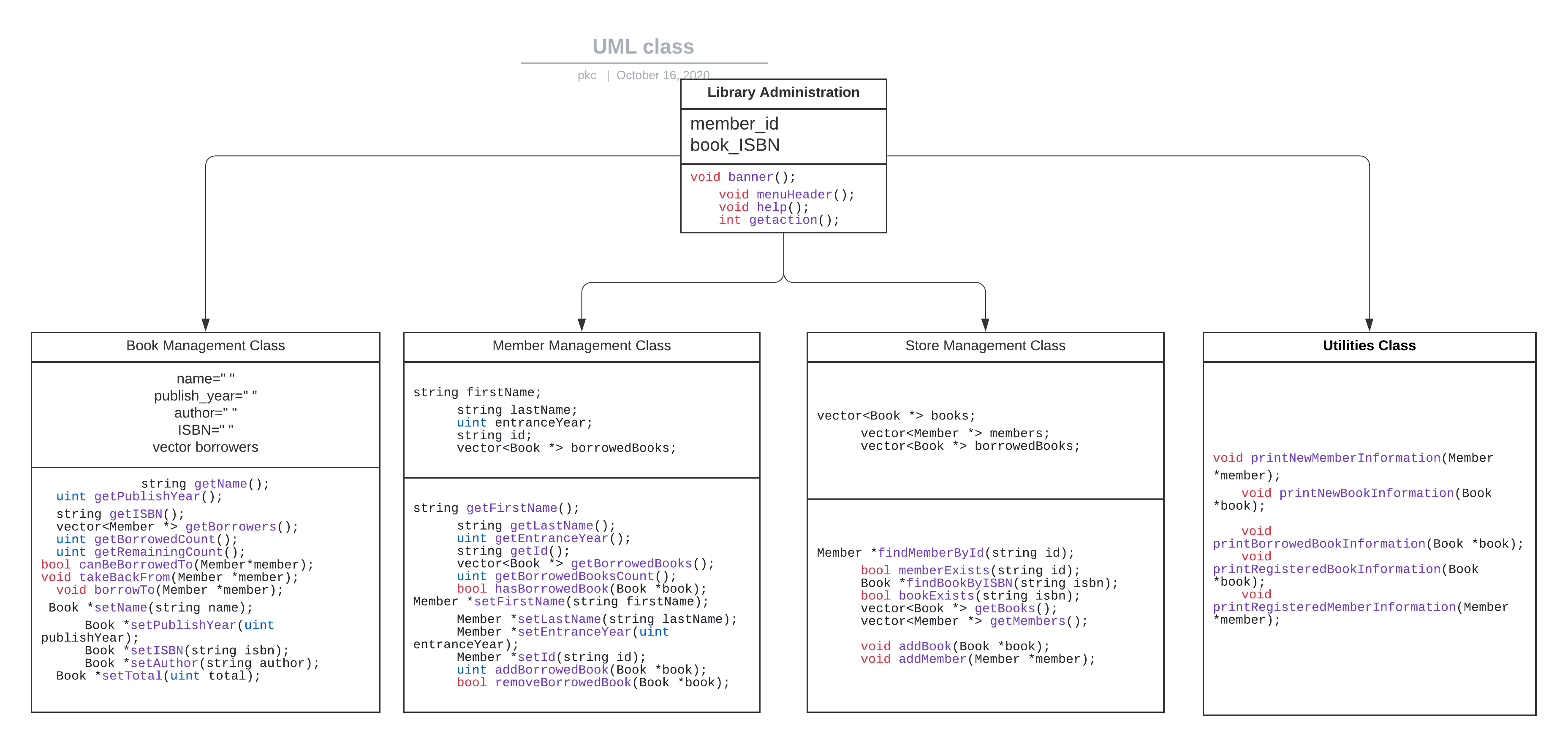
****

1. Activity Diagram



Structural Diagram

1. Class Diagram



## Test Plan

High Level Test Plan (Integration Test Plan)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Description** | **Precondition** | **Expected IO** | **Expected OP** | **Actual OP** |
| H1 | Management of Library book information | Set name of author, ISBN, book, publish year etc. | Set using inputs to the console | Successful set of Book details | Successful set of Book details |
| H2 | Management of Library Member Information | Set name of member, entrance year, id, borrowed books | Set using inputs to the console | Successful set of Member details | Successful set of Member details |
| H3 | Finding member and books using id | Get the id and ISBN set from the user | Set using inputs to the console | Find member and books using id and ISBN | Find member and books using id and ISBN |
| H4 | Provide library Utilities to Members | Get the command of utilities from user | Get the command of utilities from user | Successful completion of user command | Successful completion of user command |

Low Level Test Plan (Unit Test Plan)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Description** | **Precondition** | **Expected IO** | **Expected OP** | **Actual OP** |
| L1 | Get the information about book | Set ISBN and author name | Set using input Console | Successful information about book | Successful information about book |
| L2 | List out books which can be borrowed to member. | Enter member id | Set using input Console | Members remaining book count | Members remaining book count |
| L3 | Set the information about book. | Name of Book, Author, ISBN, publish year | Set using input Console | Successful setting of book information | Successful setting of book information |
| L4 | Get Member information | Enter member id | Set using input Console | Successful information about Member | Successful information about Member |
| L5 | Set Member Information | Name, entrance year, id, Borrowed book list | Set using input Console | Successful setting of Member information | Successful setting of Member information |
| L6 | Get Member borrowed books count | Get Borrowed list data from book class using ISBN | Set using input Console | Get the number of books member has borrowed | Get the number of books member has borrowed |
| L7 | Check whether Member has borrowed books | Get ISBN of borrowed books | Set using input Console | Return true if ISBN matches | Return true if ISBN matches |
| L8 | Find Member by id | Member ID | Set using input Console | True if ID matches | True if ID matches |
| L9 | Print New Member information | Member name, id, entrance year, | Set using input Console | Display Member Information | Display Member Information |
| L10 | Print Borrowed book information | Get book name, author, ISBN, publish year and book count | Set using input Console | Display Borrowed book information | Display Borrowed book information |

## Implementation Summary

Library Management System is working with required functionalities.

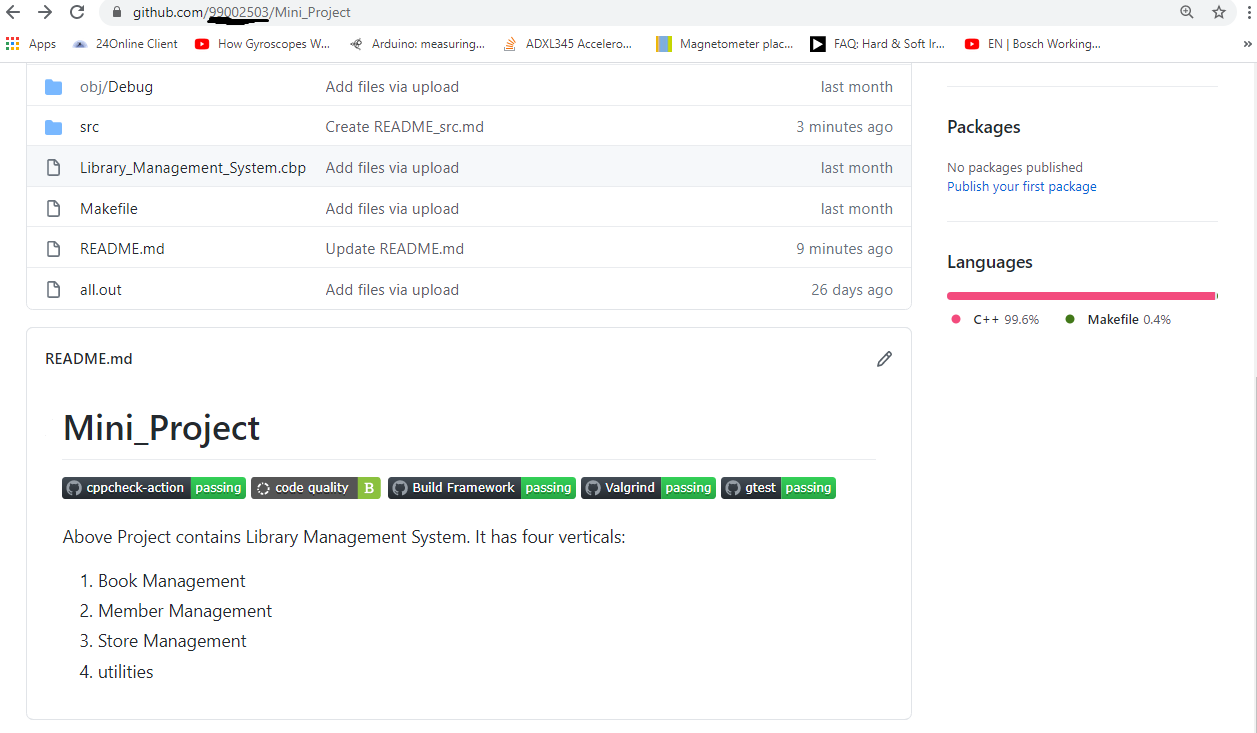
### Video Summary

“Please upload a short video on the repo for the walkthrough of the project (Team/Individual) less than 7min and less than 30MB File Size. Start is the Standard opening slide with title of Mini Project + Team members followed by the walkthrough”

### Git Link

### <https://github.com/99002503/Mini_Project>

### Git Dashboard



#### Git inspector summary

“In linux install gitinspector and Run the command –

gitinspector -H -l -m -T -w -r --grading --format=html > gitinsp.html

and upload the same to your repo and paste the snapshot in the report”

#### Build

Make file and all.out file was created.

#### Code quality and Issues or Bug Tracking

“Brief on code quality, errors and warnings flagged (issues created) and fixed”

Code quality: B

Warnings Flagged: Improve Code quality from B to A.

Status: should be fixed.

#### Unit Testing

“Unit Testing setup alignment with test plans and summary of outcome”

## **Individual Contribution & Highlights**

It is an Individual project.

### Challenges faced and how were they overcome

Usage of Collection was challenge. Self-learn and practice helped to overcome.

### Future Scope

Integration with Database using MySQL.

### 

# Mini Project -2 [Team]

“Modules linked to the Mini project are Java”

### Topic and Subtopics

Objectives: To create a program which will l maintain the details of all club members who have registered for the membership. Based on membership type, club member will be able to avail various facilities  
in the club. Membership type can be Regular Members, Premium members and Gold members. Validity  
of the membership will be based on the type of membership. A person can register for the membership and can update their details. On registration, a membership id will be generated for the person who have successfully registered for the club. Admin can manage types of memberships such as adding new membership schemes, modify existing schemes and remove any membership schemes. A member can also view the expiry date of membership and can also check the number of days or month left.

## Requirements:

**High Level Requirements:**

|  |  |
| --- | --- |
| ID | Description |
| H1 | Application will maintain the details of all club members who have registered for the membership. |
| H2 | Based on Membership type club members will be able to avail various facilities in the club. |
| H3 | A person can register for the membership and can update their details. |
| H4 | Admin can manage types of memberships |

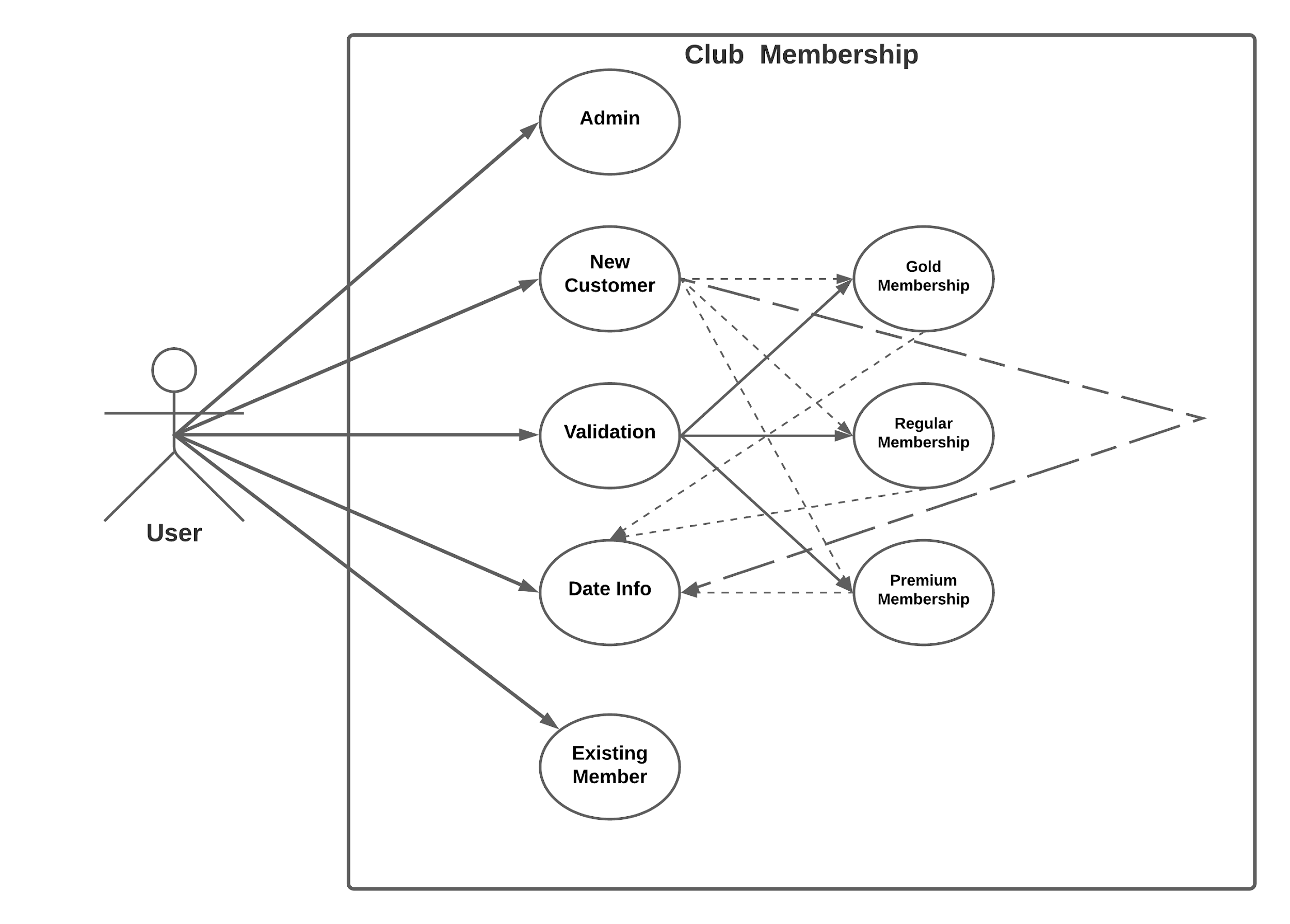
**Low Level Requirements:**

|  |  |
| --- | --- |
| ID | Description |
| L1 | Get the information about book |
| L2 | List out books which can be borrowed to member. |
| L3 | Set the information about book. |
| L4 | Get Member information |
| L5 | Set Member Information |

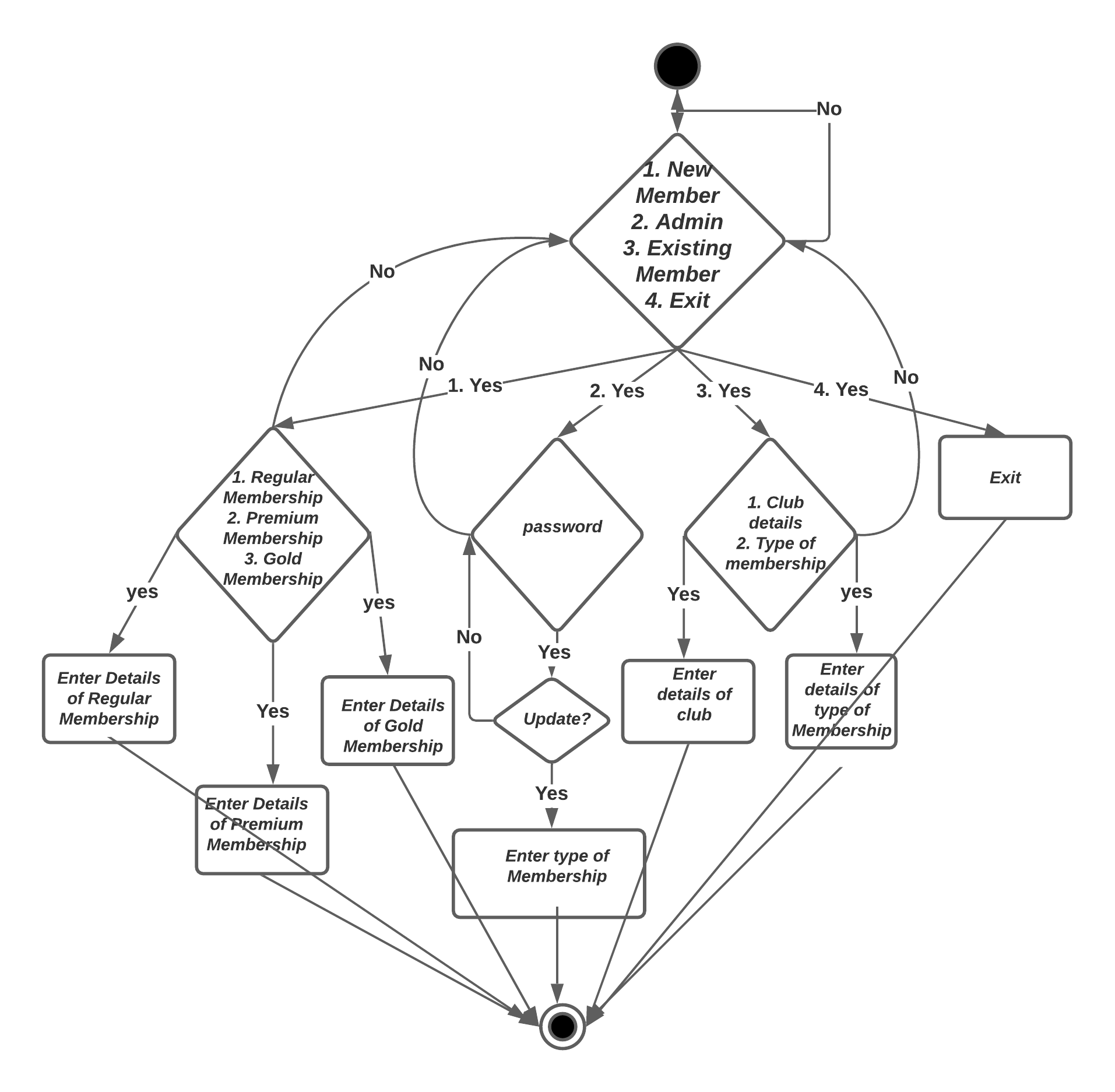
## Design

High Level Design:

1. Use Case Diagram

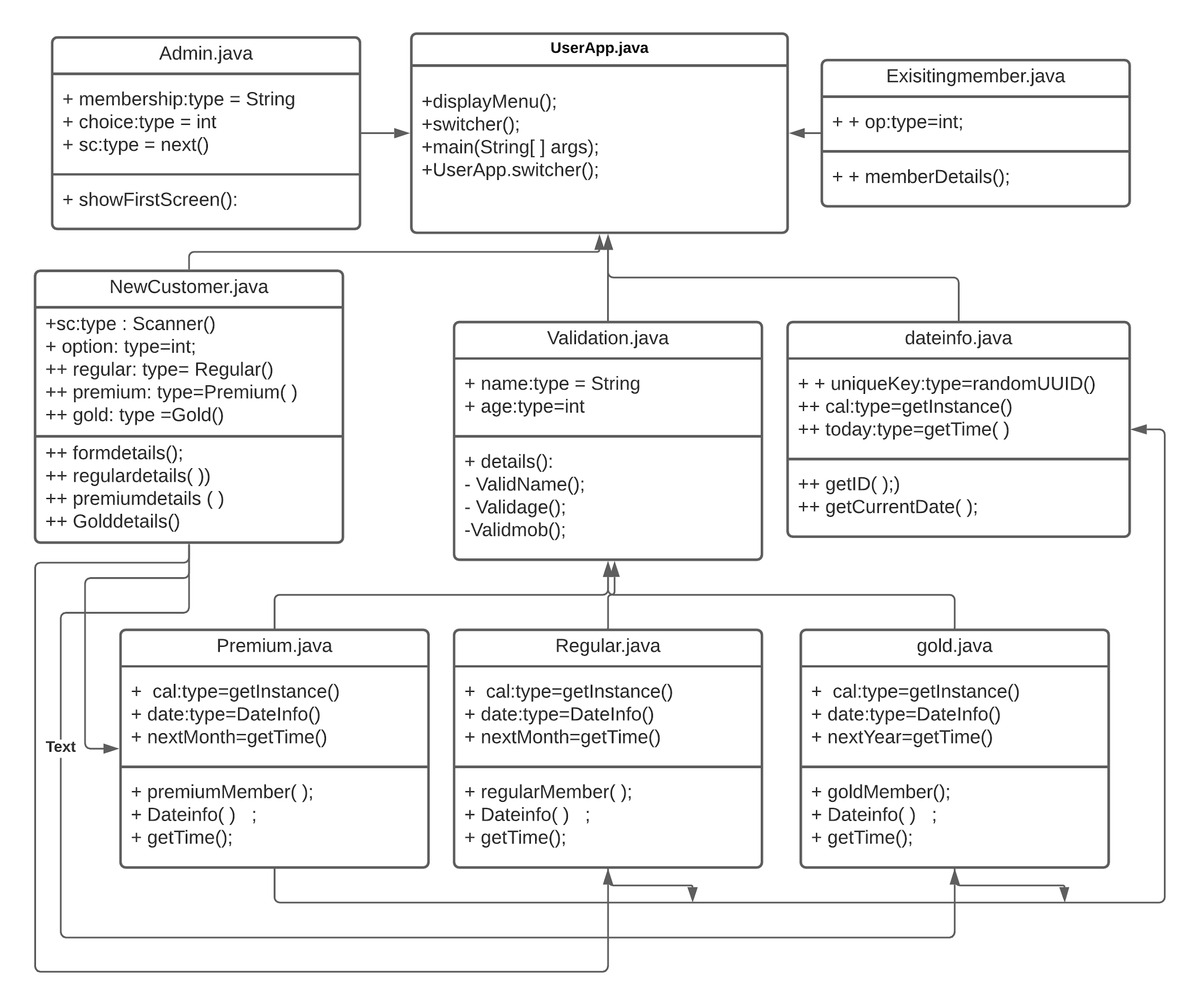


1. Activity Diagram



Low Level Design

1. Class Diagram



## Test Plan

## Test Plan:

### Functional Test plan:

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case** | **Input** | **Expected output** | **Actual output** |
| Adding new regular membership | Name, address, age, mobile number | Should show registration date, expiration date and membership id | - |
| Adding new premium membership | Name, address, age, mobile number | Should show registration date, expiration date and membership id | - |
| Adding new gold membership | Name, address, age, mobile number | Should show registration date, expiration date and membership id | - |
| Entering in Admin section | Correct password | Login successfully | - |
| Entering in Admin section | Wrong password | Invalid password | - |
| Checking validity of regular member | Date: 1 Sep 2020 | End date: 1 Oct 2020  Valid for 1 month and amount 300rs | - |
| Checking validity of premium member | Date: 1 Sep 2020 | End date: 1 Dec 2020  Valid for 3 months and amount 700rs | - |
| Checking validity of gold member | Date: 1 Sep 2020 | End date: 1 Sep 2021  Valid for 1 year and amount 1700rs | - |
| Validation of phone number | If phone number length is 10 digits | Should go to the next step | - |
| Validation of age | If age between 15-99 | Should go to the next step | - |
| Validation of name | If name provided | Should go to the next step |  |
| Existing from club | Exit option | Terminate the execution | - |

### Scenarios test plan:

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case** | **Input** | **Expected output** | **Actual output** |
| Validation of phone number | If phone number length is less than 10 digits | Phone number length should be 10 digits | - |
| Validation of age | 100 | Age should be in between 15-99 | - |
| Validation of name | If name not provided | Name should be there | - |

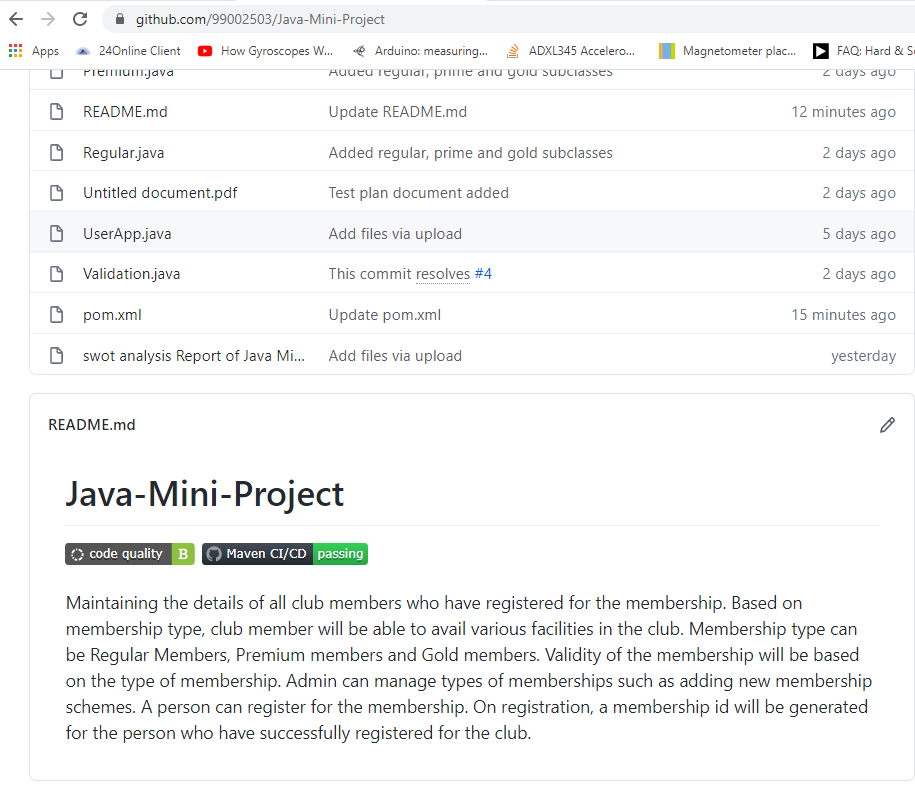
## Implementation Summary

Implemented using Java in VS code.

### Git Link

<https://github.com/99002503/Java-Mini-Project>

### Git Dashboard



### Summary

#### Git inspector summary

#### Build

#### Code quality

Code quality: B

Warnings Flagged: Improve Code quality from B to A.

Status: should be fixed.

#### Unit Testing

Unit testing was not required for this.

#### Issues

## Individual Contribution & Highlights

Worked on coding of New Customer, Existing Member and Date Info. Also worked on Architecture and creation of badge.

### Challenges faced and how were they overcome

Database Integration. Could not overcome as there is no derby file supported by JDK from 2018.