**GYM MANAGEMENT SYSTEM  
PROJECT REPORT**

**(2021 Regulation)**

***Submitted by***

**K.HARSHA VARDHAN (RA2211047010095)**

**ATHARV DOBHAL(RA2211047010134)**

**II Year/ IIIrd Semester**

**Academic Year: 2023 -2024**

***Under the Guidance of***

**Dr. SUMATHY G**

**Assistant Professor, Department of Computational Intelligence**

***In partial satisfaction of the requirements for the degree of***

**BACHELOR OF TECHNOLOGY**

**In**

**ARTIFICIAL INTELLIGENCE**

|  |
| --- |
|  |
|  |  |

**SCHOOL OF COMPUTING**

**COLLEGE OF ENGINEERING AND TECHNOLOGY SRM INSTITUTE OF SCIENCE AND TECHNOLOGY KATTANKULATHUR - 603203**

# NOV / DEC 2023

|  |
| --- |
|  |
|  |  |

SRM INSTITUTION OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR-603203

**BONAFIDE CERTIFICATE**

Certified that this Course Project Report titled **“GYM MANAGEMENT SYSTEM”** is the bonafide work done by **K.HARSHA VARDHAN[RA2211047010095], ATHARV DOBHAL[RA22110447010134]** of II Year/ IIIrd Sem B.Tech (AI) who carried out under my supervision for the course **21CSC101T** - **Object Oriented Design and Programming**. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.

|  |  |
| --- | --- |
| **SIGNATURE**  Faculty In-Charge  **Dr. Sumathy G**  Assistant Professor  Department of Computational Intelligence  SRM Institute of Science and Technology  Kattankulathur Campus, Chennai | **HEAD OF THE DEPARTMENT**  **Dr. R Annie Uthra**  Professor and Head,  Department of Computational Intelligence,  SRM Institute of Science and Technology  Kattankulathur Campus, Chennai |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.No** | **CONTENTS** | **PAGE NO** |
|  | Abstract | **1** |
|  | Objectives | **2** |
| 1. | Problem Statement | **3** |
| 2. | Modules of Project | **4** |
| 3. | Diagrams |  |
|  | 3.1. Use case Diagram | **5** |
|  | 3.2. Class Diagram | **6** |
|  | 3.3. Sequence Diagram | **7** |
|  | 3.4. Collaboration Diagram | **8** |
|  | 3.5. State Chart Diagram | **9** |
|  | 3.6. Activity Diagram | **10** |
|  | 3.7. Package Diagram | **11** |
|  | 3.8. Component Diagram | **12** |
|  | 3.9. Deployment Diagram | **13** |
| 4. | Code/Output Screenshots | **15-22** |
| 5. | Conclusion and Results | **23** |
| 6. | References | **24** |

**Abstract:**

The Gym Management System (GMS) is a comprehensive software solution designed to optimize the operations of fitness facilities, providing an integrated platform for gym owners, staff, and members. This system offers a range of features aimed at enhancing overall efficiency and member experience in modern fitness centers.

Key functionalities of the GMS include member registration and management, scheduling and booking of classes or training sessions, tracking equipment maintenance, and financial management. By automating these tasks, gym staff can focus on delivering top-quality services and personal attention to their clients.

Members benefit from a user-friendly mobile app that allows them to book classes, access workout plans, and track their progress. The GMS also supports member communication through notifications and alerts, fostering a strong sense of community and engagement.

The GMS is not only a tool for gym management but also a means to boost revenue through effective member retention and attracting new clients. Data analytics features enable gym owners to gain insights into member preferences, helping tailor offerings to their needs.

In summary, the Gym Management System offers a holistic solution for gym owners and members, fostering efficiency, engagement, and growth within the fitness industry.

**Objective:**

The Gym Management System is designed with the primary objective of offering a comprehensive and efficient software solution to address the needs of gym owners and managers. This system is intended to revolutionize the management and operation of fitness facilities by providing an integrated platform that encompasses a wide range of features and functionalities. With a focus on streamlining administrative tasks, enhancing member experience, and boosting business productivity, this Gym Management System aims to achieve several key objectives.

First and foremost, the system seeks to automate various administrative functions, including membership management, staff scheduling, financial transactions, and reporting. By automating these tasks, gym owners and managers can significantly reduce the time and effort required for day-to-day operations, allowing them to concentrate on other critical aspects of their business.

Moreover, the system intends to enhance member engagement by offering features such as online class reservations, workout tracking, and personalized fitness plans. This not only makes it easier for members to access and utilize gym services but also encourages their long-term commitment to the gym.

Additionally, the Gym Management System strives to facilitate data-driven decision-making by providing detailed insights into membership trends, financial performance, and member preferences. By analyzing this data, gym owners and managers can make informed choices that can lead to improved service quality and increased revenue.

Ultimately, the overarching goal of the Gym Management System is to contribute to the growth and success of the gym facility it serves. By achieving the objectives of automation, member engagement, and data-driven decision-making, the system aims to create a more efficient, member-focused, and financially sustainable gym environment. This, in turn, benefits both gym owners and members alike, fostering a thriving fitness community.

1. **PROBLEM STATEMENT:**

Design and implement a Gym Management System that allows the gym members to create and view their memberships. The system should provide a user-friendly menu that enables the user to choose between creating a new membership, viewing existing memberships, and exiting the system. The system should store the member's name and timing preferences (morning, afternoon, or evening), and provide an option to view the membership details on request. The system should handle incorrect inputs and provide appropriate error messages to the user.

## MODULES OF PROJECT

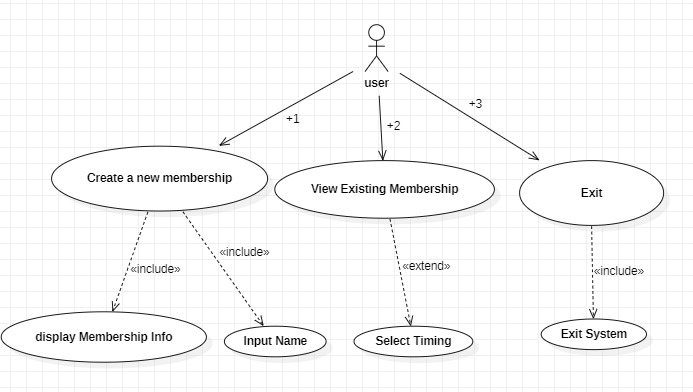
In these chapters a variety of views and development regarding the gym management has been presented. The Gym Management System allows the user for the gym component and to save the user details. This software allows us to store the all the details of the user related to the Gym Management System. The implementation of this system reduces the register books and provides saved reports for the users. Gym Management System helps to easy management of customers, equipment, plans, enquiries etc. The system in which you can login into the system and set your gym plan or program.

1. \*\*Create a New Membership\*\*: This choice enables users to embark on their fitness journey by inputting their name and selecting a session timing (morning, afternoon, or evening). The system leverages the `GymMembership` class to securely store this data and appends it to the `members` vector. The result is an efficient and organized approach to membership creation.

2. \*\*View Existing Memberships\*\*: For both administrators and members, the ability to view current memberships is essential. The program, in response to this choice, provides a comprehensive list of existing memberships. Names and session timings of all registered members are displayed, streamlining the process of accessing this crucial information. Importantly, the system also accommodates scenarios where no memberships exist, ensuring that users are informed.

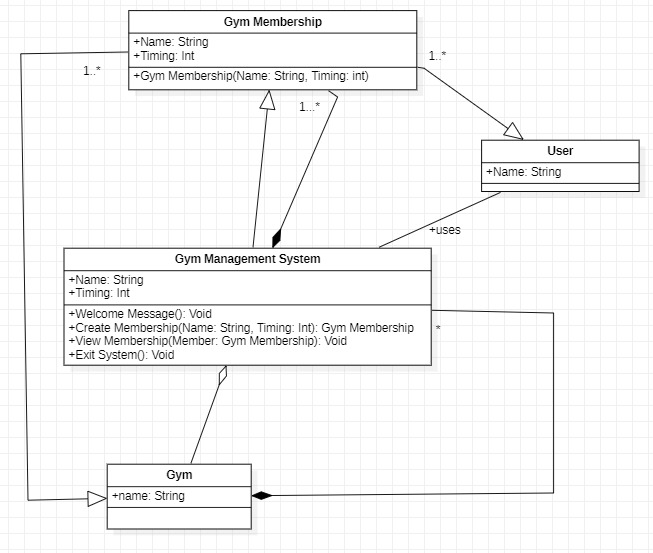
3. \*\*Exit\*\*: When users have accomplished their objectives, they can gracefully exit the program. This choice, though seemingly simple, allows users to conclude their interactions with a sense of satisfaction and gratitude, thanks to a thank-you message.

## USE CASE DIAGRAM

****

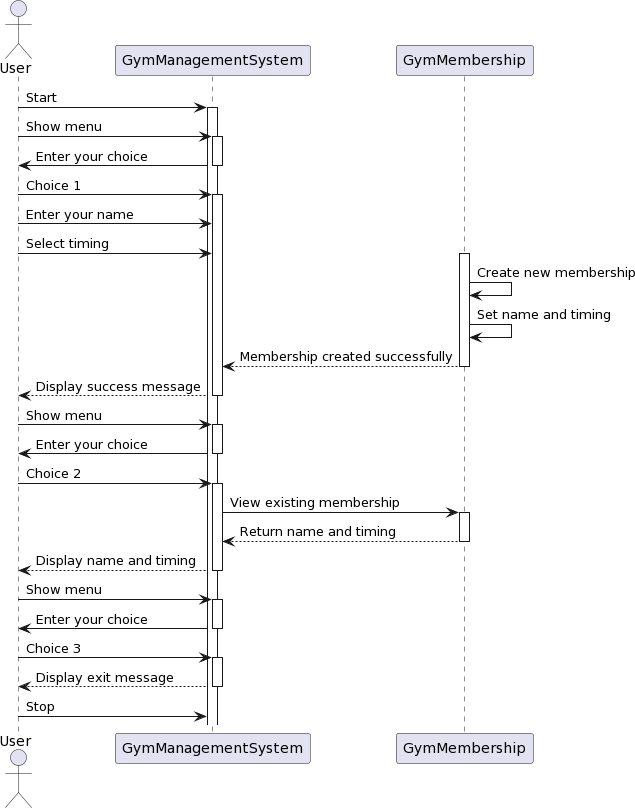
The use case diagram for the given Gym Management System code includes two primary actors, the Gym Manager and the Gym Members. The Gym Manager can create a new membership and view existing memberships. The Gym Members can also view their existing memberships. The system provides a user-friendly menu that enables the users to choose between creating a new membership, viewing existing memberships, and exiting the system. The system stores the member's name and timing preferences and provides an option to view the membership details on request. The use case diagram helps to identify the interactions between the system and its users, and the functionalities provided by the system.

## CLASS DIAGRAM

****

The class diagram for the Gym Management System code consists of a single class named "GymMembership" that represents the membership data, including the member's name and timing preferences. The "main" method is not included in the class diagram as it is not part of the "GymMembership" class. The "GymMembership" class has two attributes, "name" and "timing", and a constructor method that initializes these attributes. The class also has getters and setter’s methods to access and modify the attributes. The class diagram shows the structure of the "GymMembership" class and its relationships with other classes in the system, providing a high-level overview of the system's architecture.

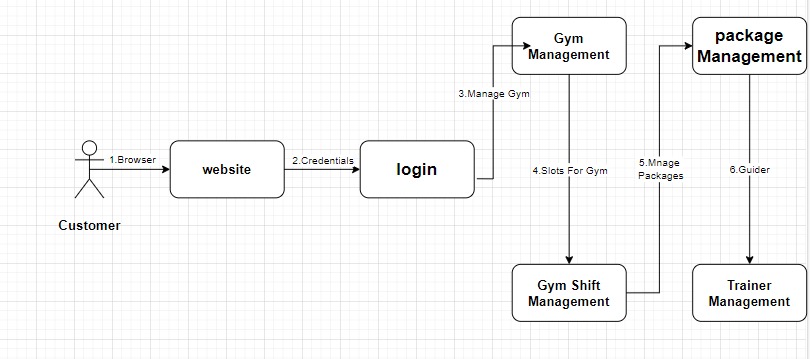
## SEQUENCE DIAGRAM





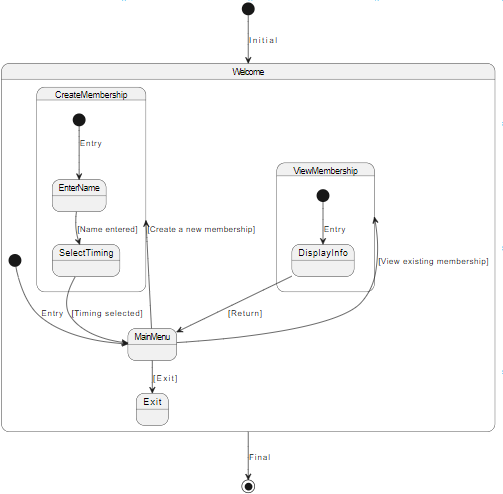
The sequence diagram for the Gym Management System code shows the interactions between the user and the system during the creation and viewing of memberships. The user initiates the process by selecting the "Create a new membership" or "View existing membership" option from the menu. If the user selects the "Create a new membership" option, the system prompts the user to enter their name and timing preferences, and then creates a new membership object with the provided data. If the user selects the "View existing membership" option, the system checks if a membership object exists and retrieves and displays its data if available. The sequence diagram helps to visualize the flow of control and the interactions between the user and the system during the membership creation and viewing processes.

## COLLABORATION DIAGRAM

****

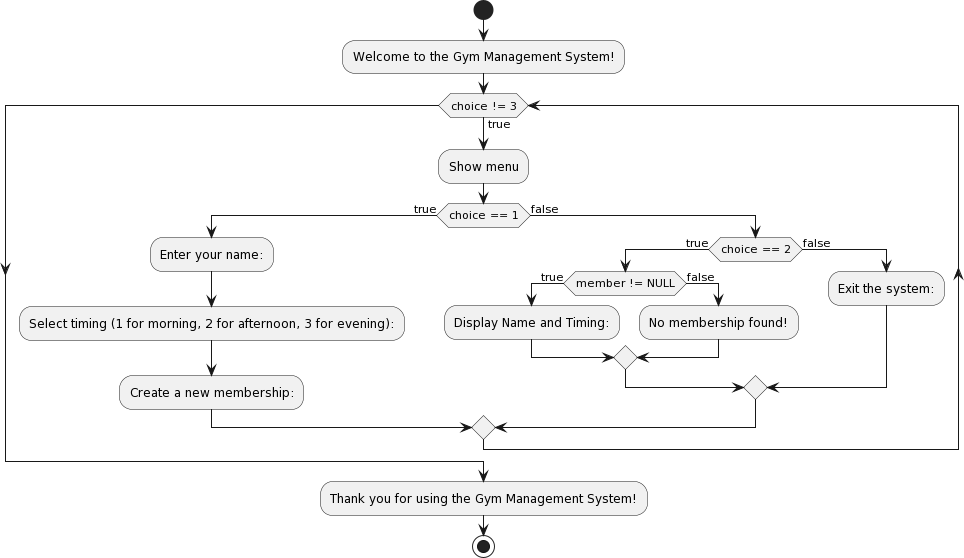
The communication diagram for the Gym Management System code shows the objects and messages exchanged during the membership creation and viewing processes. The diagram consists of two objects, the "user" and the "GymMembership" object. The user sends a message to the system to create a new membership or view an existing membership. If the user selects the "Create a new membership" option, the system creates a new "GymMembership" object with the provided data and sends a confirmation message to the user. If the user selects the "View existing membership" option, the system retrieves the data from the existing "GymMembership" object and sends it back to the user. The communication diagram helps to illustrate the interactions between the user and the "GymMembership" object during the membership creation and viewing processes.

## STATE CHART DIAGRAM



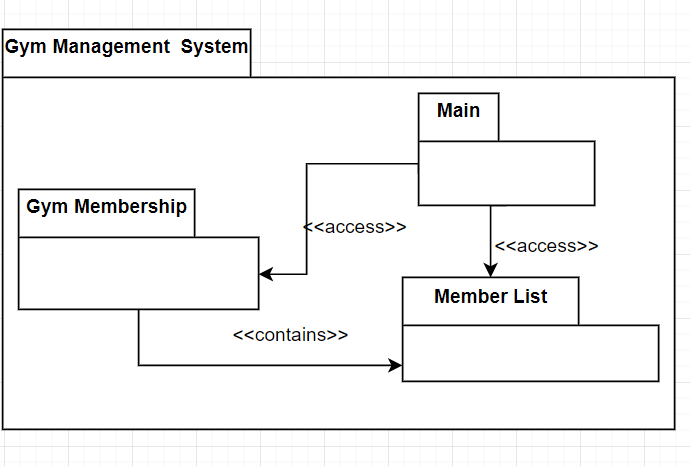
The state chart diagram for the Gym Management System project describes the different states and events that can occur during the system's operation. It includes three states: "Main Menu", "Create Membership", and "View Membership". The "Main Menu" state is the initial state, and the user can transition to the other two states based on the input provided. The "Create Membership" state allows the user to input their name and select a timing for their gym membership. After inputting the required information, the system transitions back to the "Main Menu" state. The "View Membership" state displays the information for the existing membership, and the system also returns to the "Main Menu" state after displaying the information.

## ACTIVITY DIAGRAM



The activity diagram for the Gym Management System code shows the flow of activities that occur during the membership creation and viewing processes. The diagram consists of three activities, "Display menu", "Create membership", and "View membership". The "Display menu" activity presents the user with the menu options and waits for input. The "Create membership" activity prompts the user to enter their name and timing preferences and then creates a new "GymMembership" object with the provided data. The "View membership" activity retrieves the data from the existing "GymMembership" object and displays it to the user. The activity diagram helps to visualize the steps involved in the membership creation and viewing processes, making it easier to understand the system's behavior.

## PACKAGE DIAGRAM



The package diagram for the Gym Management System code consists of a single package named "GymManagementSystem" that contains two classes: "GymMembership" and "main". The "GymMembership" class represents the data of the membership, including the member's name and timing preferences. The "main" class contains the main method of the system that implements the functionalities for creating and viewing memberships. The diagram shows a dependency relationship between the "main" class and the "GymMembership" class, indicating that the "main" class uses the "GymMembership" class to store and retrieve membership data. The package diagram helps to organize the code into meaningful packages and shows the relationships between the classes in the system.

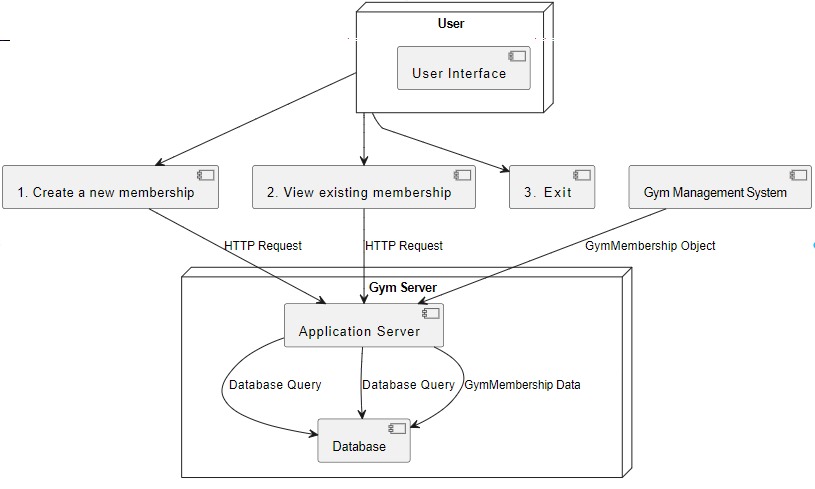
## COMPONENT DIAGRAM

Diagram

Description automatically generated

The component diagram for the Gym Management System code shows the high-level components of the system and their interrelationships. The diagram consists of a single component named "Gym Management System" that represents the entire system. This component is further divided into two sub-components, "User Interface" and "Membership Management". The "User Interface" sub-component interacts with the user to present the menu options and retrieve input. The "Membership Management" sub-component manages the creation and viewing of membership data through the "GymMembership" class. The component diagram shows the modular structure of the system and the relationships between its components, making it easier to understand the system's architecture and organization.

* 1. **DEPLOYMENT DIAGRAM**



A deployment diagram for a Gym Management System illustrates how the software and hardware components of the system are distributed and interconnected in a real-world deployment scenario. In this context, it serves as a crucial tool for visualizing the physical deployment of the system and helps ensure that the software operates effectively in its intended environment. The diagram typically represents the various nodes, which are hardware or software components, such as servers, databases, workstations, and more, along with their relationships. For a Gym Management System, this could include servers hosting the application, the database server storing member and scheduling information, client workstations used by gym staff, and potentially external systems for payment processing. Arrows or connectors show how these components interact, depicting the flow of data and communication pathways. The deployment diagram is instrumental in system architecture planning, infrastructure provisioning, and understanding how the Gym Management System is hosted and accessed, ultimately ensuring its seamless operation in the real-world gym environment.

## SOURCE CODE

## #include <iostream>

## #include <string>

## #include <vector>

## using namespace std;

## // Class to store information about a gym membership

## class GymMembership {

## public:

## string name;

## int timing;

## // Constructor

## GymMembership(string n, int t) {

## name = n;

## timing = t;

## }

## };

## int main() {

## int choice;

## string name;

## int timing;

## vector<GymMembership> members; // Store all created memberships

## cout << "Welcome to the Gym Management System!" << endl;

## do {

## // Show menu

## cout << "-------------------------" << endl;

## cout << "1. Create a new membership" << endl;

## cout << "2. View existing memberships" << endl;

## cout << "3. Exit" << endl;

## cout << "-------------------------" << endl;

## cout << "Enter your choice: ";

## cin >> choice;

## switch (choice) {

## case 1:

## // Create a new membership

## cout << "Enter your name: ";

## cin.ignore(); // Clear the newline character from the previous input

## getline(cin, name);

## cout << "Select timing (1 for morning, 2 for afternoon, 3 for evening): ";

## cin >> timing;

## members.push\_back(GymMembership(name, timing)); // Add the new membership to the vector

## cout << "Membership created successfully!" << endl;

## break;

## case 2:

## // View existing memberships

## if (!members.empty()) {

## cout << "Existing memberships:" << endl;

## for (size\_t i = 0; i < members.size(); ++i) {

## cout << "Membership " << i + 1 << ":" << endl;

## cout << "Name: " << members[i].name << endl;

## cout << "Timing: ";

## switch (members[i].timing) {

## case 1:

## cout << "Morning";

## break;

## case 2:

## cout << "Afternoon";

## break;

## case 3:

## cout << "Evening";

## break;

## default:

## cout << "Invalid timing";

## }

## cout << endl;

## }

## } else {

## cout << "No memberships found!" << endl;

## }

## break;

## case 3:

## // Exit

## cout << "Thank you for using the Gym Management System!" << endl;

## break;

## default:

## cout << "Invalid choice! Please try again." << endl;

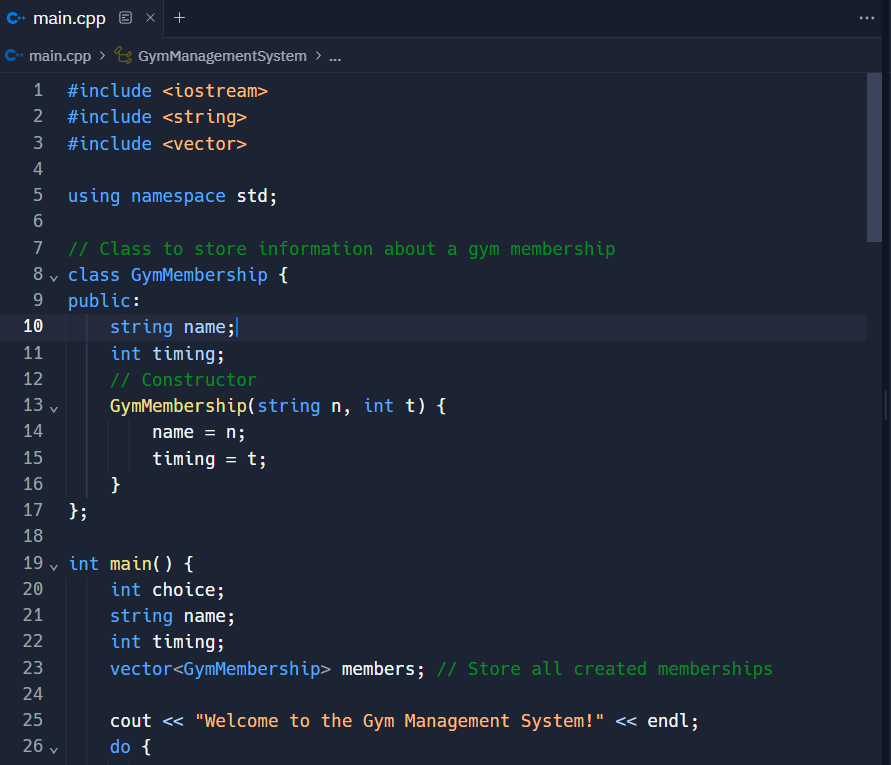
## }

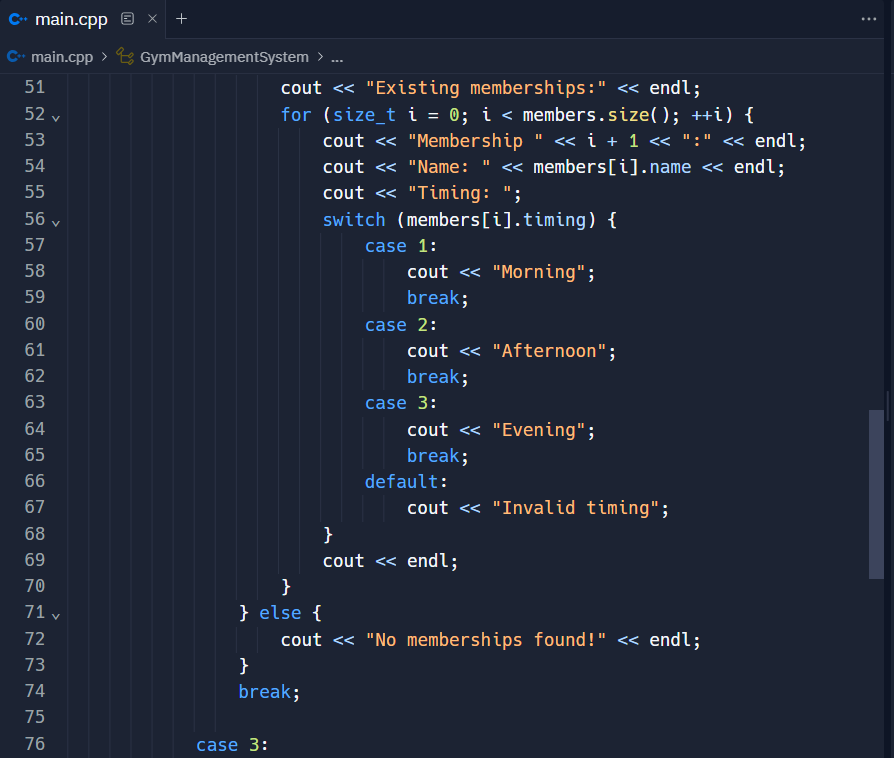
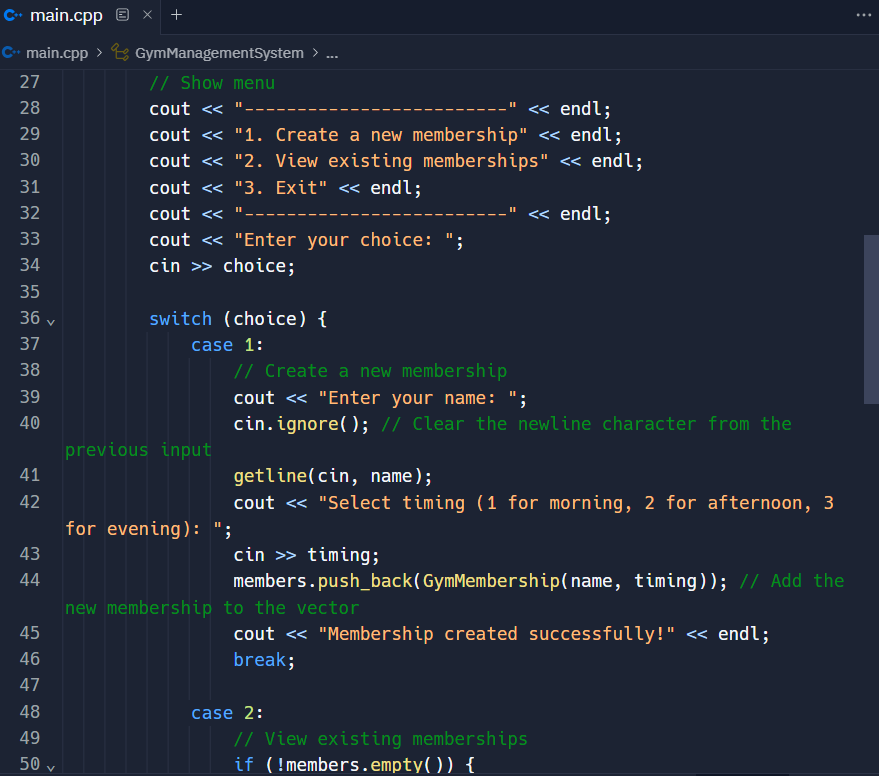
## } while (choice != 3);

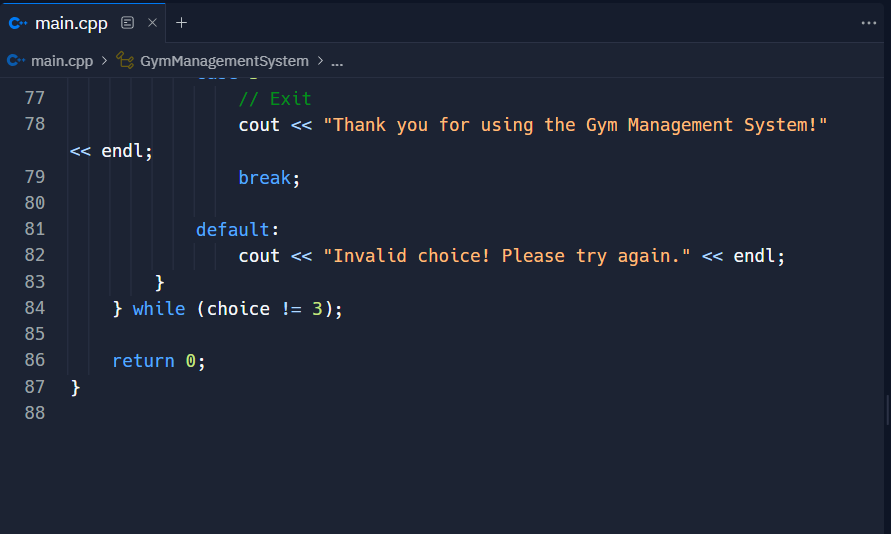
## return 0;

## }

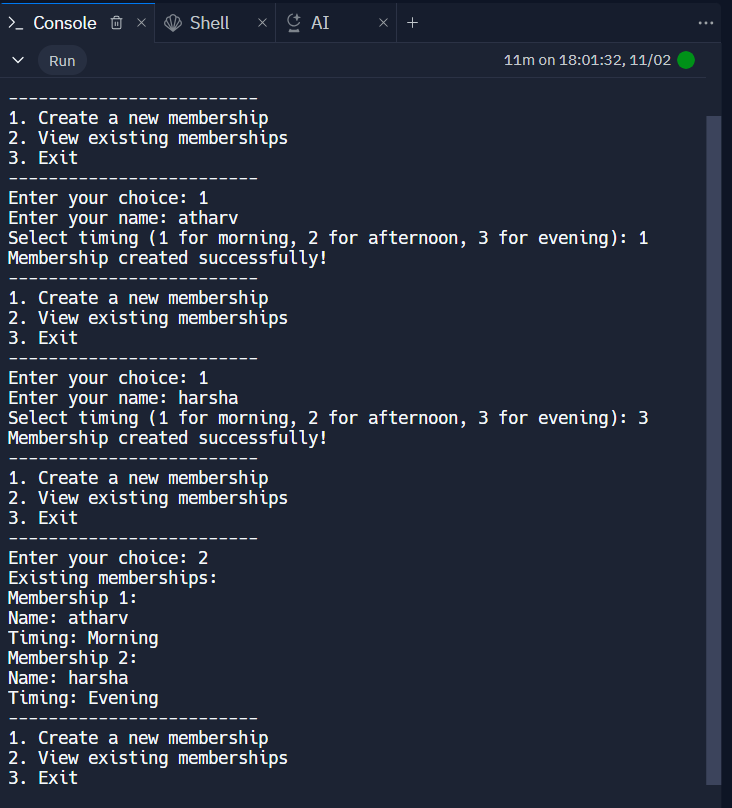
## CODE:

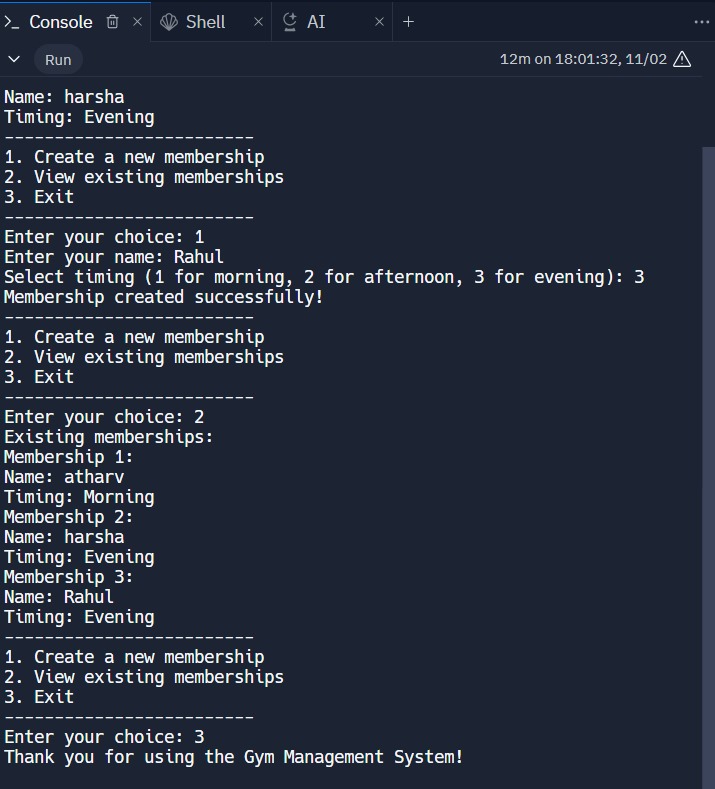


****

****

**OUTPUT:**





## CONCLUSION

The above code implements a simple gym management system that leverages object-oriented programming principles and the C++ programming language. This program allows users to create new gym memberships and view existing ones, providing a foundation for more extensive features to be added in the future. With a focus on modularity and user interaction, this gym management system caters to the basic needs of both gym administrators and members.

At the core of this system is the `GymMembership` class. This class is responsible for storing crucial information related to gym memberships. For each member, it records their name and the timing of their gym sessions. The program efficiently manages this information, making it accessible to the user when needed.

Upon execution, the program presents a user-friendly menu. Users can interact with the program by selecting options from this menu, which is facilitated through a switch statement. The available options include creating a new membership, viewing existing memberships, and exiting the system.

When the user chooses to create a new membership, the program guides them through the process. It collects essential details, such as the member's name and session timing, and then stores this information using the `GymMembership` class. This option is especially useful for gym administrators who need to register new members efficiently.

Furthermore, an automated reminder system for session timings or membership renewals can significantly improve member engagement and retention. This feature can help in sending notifications to members about upcoming sessions or membership expiration, thus enhancing the overall member experience.

## REFERENCES

1. Minghui Liu, Erxiang Chen, “Research and Design on Library Management System Based on Struts and Hibernate Framework,” WASE International Conference on Information Engineering, ICIE, 2009.
2. A. V. Dinesh Kumar, K Bhargav Ram Rayal, M. Saraswathi, “Smart Gym Management System,” IJSRET, Trends Volume 6, Issue 3, May-June-2020.
3. Kavita Gupta, “Biometric Authentication” International Journal of Engineering Research Technology (IJERT), VIMPACT 2017.
4. Mr Akshay Sambare, Dipali Bondre, Sachin Thorat, Miss Archana Vishe, Prof. Ankit Sanghavi “Gym Monitoring Framework for Fitness Management System,” International Journal of Advanced Research in Computer and Communication Engineering Vol. 6, Issue 3, March 2017.
5. Jing Yang, Bei Jing, Leixiao Li, Yan Zhao ‘Combined Framework of Struts and Hibernate‘ Combined Framework of Str6uts and Hibernate,” IEEE-2011.