**GYM MANAGEMENT SYSTEM**

A CAPSTONE PROJECT REPORT

# (Object Oriented Programming with C++ Using Encapsulation- DSA0199)

***Submitted to***

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES**

***In partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING IN COMPUTER SCIENCE & ENGINEERING**

***By***

1. **Madhusudhan (192210035),**
2. **Akash (192210443)**

***Course Faculty***

**Dr. Jayanthi**



**SAVEETHA SCHOOL OF ENGINEERING, SIMATS, CHENNAI - 602105**

**MARCH-2024**

**SAVEETHA SCHOOL OF ENGINEERING**

**SAVEETHA INSTITUE OF MEDICAL AND TECHNICAL SCIENCES,CHENNAI - 602105**

**BONAFIDE CERTIFICATE**

Certified that this project report **“GYM MANAGEMENT SYSTEM”** is the Bonafide work of **“K. Madhusudhan, D. Akash”** who carried out the project work under my supervision.

**Submitted to**

**Dr. Jayanthi**

(Course Faculty)

Department of Deep Learning, Saveetha School of Engineering, SIMATS

## SIGNATURE of Course Faculty

**ACKNOWLEDGEMENT**

This project work would not have been possible without the contribution of many people. It gives me immense pleasure to express my profound gratitude to our Honorable Chancellor **Dr. N. M. Veeraiyan**, Saveetha Institute of Medical and Technical Sciences, for his blessings and for being a source of inspiration. I sincerely thank our Director of Academics **Dr. Deepak Nallaswamy,** SIMATS, for his visionary thoughts and support. I am indebted to extend my gratitude to our Director **Dr. Ramya Deepak,** Saveetha School of Engineering, for facilitating us all the facilities and extended support to gain valuable education and learning experience.

I register my special thanks to **Dr. B. Ramesh,** Principal, Saveetha School of Engineering for the support given to me in the successful conduct of this project. I wish to express my sincere gratitude to my Course faculty **Ms.K.Divya**, for his inspiring guidance, personal involvement and constant encouragement during the entire course of this work.

I am grateful to Project Coordinators, Review Panel External and Internal Members and the entire faculty of the Department of Design, for their constructive criticisms and valuable suggestions which have been a rich source to improve the quality of this work.

**STUDENT NAME’s**

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **CHAPTER**  **NO** | **TITLE** |
| **1** | **Introduction** |
| **2** | **Project Description and Goals:** |
| **3** | **Technical Specifications:** |
| **4** | **Design Approach and Details** |
| **5** | **Schedule, Tasks, and Milestones:** |
| **6** | **Project Demonstration:** |
| **7** | **Cost Analysis:** |
| **8** | **Result:** |
| **9** | **Discussion:** |
| **10** | **Conclusion:** |

**Introduction:** The **Gym Management System Application** is a state-of-the-art solution designed to streamline gym operations, improve member experience, and manage daily tasks efficiently. Built on a foundation of object-oriented programming, the system leverages the power of classes to handle various functionalities such as member registration, trainer management, class scheduling, and billing. The system provides a secure login page with tailored roles for admins, trainers, and members, ensuring the right access level for each user.

The application also facilitates seamless management of workout plans, personal training sessions, membership renewals, and real-time tracking of gym activities. With the view schedule functionality, members and trainers can monitor their fitness sessions and class schedules, while the billing feature automates invoicing for memberships and additional services. By integrating these essential features into a cohesive platform, the Gym Management System allows gyms to optimize operations, boost member engagement, and enhance the overall fitness experience. In a competitive fitness industry, this system offers gyms the tools needed to stay ahead of trends, attract more members, and deliver outstanding fitness services.

**Project Description and Goals:** The **Gym Management System** is a comprehensive software solution tailored to optimize gym operations. Key features include a login page, admin panel, member page, trainer management, workout and class scheduling, and billing. The primary goal is to create a robust, user-friendly platform that enhances the management of gym resources and boosts the overall member experience.

**Functionality:** The system provides user authentication, member and trainer management, class and session scheduling, workout tracking, and automated billing functionalities.

**User-Friendly Interface:** The system will feature an intuitive, user-friendly interface that ensures easy navigation for gym staff, trainers, and members, providing clear access to essential features such as booking classes or reviewing workout progress.

**Accurate Calculation:** Precise algorithms will calculate membership fees, additional charges for personal training sessions, discounts, and other transactions to ensure accurate and seamless billing.

**Error Handling:** The system will include robust error handling mechanisms to detect and manage issues such as failed login attempts, invalid input, and scheduling conflicts, providing helpful feedback for users to resolve problems.

**Cross-Platform Compatibility:** To ensure accessibility across devices, the system will be developed with cross-platform compatibility in mind, allowing it to function seamlessly on desktops, tablets, and mobile devices.

**Documentation and Support:** Comprehensive user manuals, FAQs, and troubleshooting guides will be provided to assist users. Support channels such as email and chat support will be available to address any concerns or issues that arise.

**Testing and Validation:** Thorough testing of each system component will be conducted to ensure performance, security, and usability. The system will undergo validation against test cases to verify that it meets the functional and non-functional requirements.

**Technical Specifications:** Appropriate programming languages, frameworks, and databases will be chosen based on project needs. Data structures and class hierarchies will be designed to represent the various entities such as members, trainers, and workout plans, ensuring efficient handling of the system’s operations.

**Platform Compatibility:** The system will be tested for compatibility with common operating systems (Windows, macOS, Linux) and web browsers (Chrome, Firefox, Safari). Testing will also cover different devices to ensure cross-platform operability.

****Design Approach and Details**:** A modular and scalable design approach will be used to allow for future expansions or modifications. Object-oriented design principles will promote reusability and ease of maintenance, ensuring the system can evolve as the gym’s needs grow.

**Schedule, Tasks, and Milestones:**

**Planning Phase:**

* Define project scope, objectives, and requirements.
* Identify stakeholders, including gym managers, trainers, and members.
* Create a detailed project plan with timelines, milestones, and resource allocation.

**Design Phase:**

* Design the system architecture and database schema for storing member data, schedules, and billing information.
* Develop wireframes and mockups for the user interface.
* Define the class structures for users (members, trainers, admins), workout plans, and billing modules.

**Project Demonstration:** A system demonstration will be conducted, showcasing key features such as membership management, class scheduling, and automated billing. Feedback will be gathered for further refinement.

**Cost Analysis:** Development costs, including licenses, infrastructure, and support, will be calculated and compared against the expected return on investment from improved gym operations and member retention.

**Result:** The **Gym Management System** offers a complete solution for managing gym operations, including membership registration, class scheduling, trainer assignments, and billing. By automating key tasks, the system helps gyms improve efficiency, reduce manual errors, and provide a better experience for members and staff alike.

**Discussion:** The development process involved close collaboration with gym owners and staff to ensure the system met real-world needs. Continuous feedback was collected to make improvements throughout the design and implementation phases. The system’s adaptability ensures that gyms can scale operations as their member base grows or new services are introduced. Looking forward, ongoing support and maintenance will be critical to addressing emerging needs in the evolving fitness industry.

**Summary:** The **Gym Management System** provides a powerful, user-friendly platform for managing gym operations, from membership registration to personal training and class schedules. With its automated billing and accurate workout tracking, gyms can enhance the member experience while optimizing internal operations.

**Conclusion:** The **Gym Management System** is a vital tool for modern fitness centers seeking to optimize their operations, attract and retain members, and streamline billing and scheduling. In the ever-evolving fitness industry, adopting technology-driven solutions like this system is crucial for long-term success and growth.

**Code:**

.

#include <iostream>

#include <string>

#include <vector>

#include <limits>

#include <algorithm>

using namespace std;

class GymMembership {

public:

string name;

int timing;

GymMembership(string n, int t) : name(n), timing(t) {}

};

bool compareByTiming(const GymMembership& a, const GymMembership& b);

int main() {

int choice;

string name;

int timing;

vector<GymMembership> members;

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

do {

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

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

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

cout << "3. Modify a membership" << endl;

cout << "4. Cancel a membership" << endl;

cout << "5. Search for a membership" << endl;

cout << "6. Exit" << endl;

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

cout << "Enter your choice: ";

while (!(cin >> choice)) {

cout << "Invalid input. Please enter a number: ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

switch (choice) {

case 1:

cout << "Enter your name: ";

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

getline(cin, name);

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

while (!(cin >> timing) || (timing < 1 || timing > 3)) {

cout << "Invalid timing. Please enter 1, 2, or 3: ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

members.push\_back(GymMembership(name, timing));

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

break;

case 2:

if (!members.empty()) {

cout << "Existing memberships (in order of timing):" << endl;

sort(members.begin(), members.end(), compareByTiming);

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:

if (!members.empty()) {

int index;

cout << "Enter the membership number to modify: ";

while (!(cin >> index) || (index < 1 || index > members.size())) {

cout << "Invalid membership number! Please enter a valid number: ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

cout << "Enter new name: ";

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

getline(cin, members[index - 1].name);

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

while (!(cin >> timing) || (timing < 1 || timing > 3)) {

cout << "Invalid timing. Please enter 1, 2, or 3: ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

members[index - 1].timing = timing;

cout << "Membership modified successfully!" << endl;

} else {

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

}

break;

case 4:

if (!members.empty()) {

int index;

cout << "Enter the membership number to cancel: ";

while (!(cin >> index) || (index < 1 || index > members.size())) {

cout << "Invalid membership number! Please enter a valid number: ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

members.erase(members.begin() + index - 1);

cout << "Membership canceled successfully!" << endl;

} else {

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

}

break;

case 5:

if (!members.empty()) {

string searchName;

cout << "Enter the name to search: ";

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

getline(cin, searchName);

bool found = false;

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

if (members[i].name == searchName) {

found = true;

cout << "Membership found at index " << 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;

}

}

if (!found) {

cout << "Membership not found for the given name." << endl;

}

} else {

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

}

break;

case 6:

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

break;

default:

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

}

} while (choice != 6);

return 0;

}

bool compareByTiming(const GymMembership& a, const GymMembership& b) {

return a.timing < b.timing;

}

**Output:**

