

**Stage 1**

**CSCI-2210**

**Gym Management System**

**By Raul Sena, Jacey Tsosie, Angel Garcia Vega**

# GYM MANAGEMENT SYSTEM – HIIT LABS

*Description: This program should manage all operations and handle all aspects of a gym called HIIT-LABS*

## 1. Gym Management class (Class Gym)

Handle the System Components: The Gym class administers and manages the system. It coordinates Classes, Memberships, Staff, Equipment, Members, and Facility. It is the Main method of managing and initializing the gym system.

## 2. Member Management (Class Member)

Create new members, delete members, and update the members information, assign member to membership: The system must allow creating new members, update information of existing members, and removing members. Becoming a member involves enrolling them into their desired tier of membership; default membership shall be the lowest.

Members' Information stored: Members' details such as age, name contact, memberID, type of Membership, SetPin (Check-ins) and Billing information, should be allowed to be modified, updated, or erased.

*Example: Member M1 = new Member (###);*

*38: Age*

*Jhon Cena: Name*

*111-222-3333: Contact*

*00001: MemberID*

*Basic: Type of Membership // default lowest tier*

*1234: SetPin*

*1111 2222 3333 4444: Billing Information*

*Jhon E Cena*

*10/27*

### 3. Fitness Classes Management (Class fitClass) Modify Info

Create new sessions, delete sessions, assign instructor for sessions, assign name of session, remove instructor for the session, create and assign classID, create and assign date (DD/MM/YR), create and assign time Start: End (military time 24/hrs.), create and assign capacity: The system must allow the creation, deletion, scheduling and rescheduling of sessions. Not only that but should also allow the assigning or removal of an instructor, system should also assign capacity of class, classID (#### 4 digits), class date and class time.

*Example: fitClass class1 = new fitClass();*

*Jeniffer Hall: fitness Instructor Str*

*Pilates: Name of session Str*

*3333: ClassID Int*

*02/29/2024: Date Int*

*11:00: Start Int*

*12:30: End Int*

*25: Capacity Int*

### 4. Membership Management (Class Membership)

Create new memberships, delete memberships, modify memberships, add or remove members from memberships, upgrade or downgrade members memberships: the system must allow the creation, deletion and modifications of memberships, not only that but members who would like to end, upgrade or downgrade their memberships should be allowed to do so. Depending on the Membership, members will have access to Book certain activities such as specialized equipment or fitClass.

*ROUGH DRAFT*

*Membership basic = new Membership ();*

*Membership.basic(M1);*

*M1.upgrade(M1);*

*M1.downgrade(M1);*

## **5. Staff Management (Class StaffMember) alias General Staff**

Create and assign default position, Create new staff, delete new staff, create and assign information about staff, update, modify or delete information about staff(age, contact, availability), create and assign StaffID: The system must allow the creation of new staff, deletion of staff (only if approved by Manager), update and modify staff information.

Staff Member Information Stored: Staff's information such as age, availability, contact, and StaffID, should be allowed to be modified, updated, or erased.

## **6. Manager Management (Class Manager)**

Manager Class will Inherit from General Staff and attain the ability to: Create and assign default position, Create new staff, delete new staff, create and assign information about staff, update, modify or delete information about staff(age, contact, availability), create and assign StaffID. Management Class will also have the ability to assign staff duties, delete staff, assign staff positions (Manager, Trainer, general), demote or promote staff: System must assign staff duties, assign positions such as management, front office, janitor, general staff, etc. System must allow the promotion or demotion of staff. Manager Class will also store the general information (same information as Staff Class).

## **7. Trainer Management (Class Trainer)**

Trainer Class will Inherit from General Staff: and be assigned a fitclass, Trainer class will create, assign and return Trainer schedule and what classes they teach, Trainer will have the ability to modify classes if they would like to do so (cancel, edit info, or close). Trainer Class will also store the general information (same information as Staff Class).

## **8. Equipment Management (Class Equipment) alias General Equipment**

Create and assign average life of equipment, Create and return a list of equipment in the facility, return information about specific equipment piece, create new equipment, Create and assign equipmentID Add or remove equipment, replace equipment, delete equipment, Modify, edit or delete information about equipment piece: The system must allow the creation, addition, replacement, removal and deletion of equipment, it must also return the information of the following equipment piece in a report: date of purchase of equipment, average life of equipment, how long the equipment has been in the facility, dates of maintenance of equipment. Replace equipment only if the average life of equipment is past due.

Equipment Information Stored: date of purchase of equipment, average life of equipment, how long the equipment has been in the facility, dates of maintenance of equipment logged, System should be able to return information listed and should allow the information to be modified, updated and erased.

## **9. Specialized Equipment Management (Class SpecialEquip) Modify Info**

The system must allow the assignment of specialized equipment: specialized equipment will Inherit the same features as regular equipment, only difference will be the assignment of the equipment through the Booking Class. Class Inherits creation, addition, removal, and deletion of Specialized equipment.

When a new specialized piece of equipment is created a specializedequipID is assigned, staff create, assign or modify date available (DD/MM/YR), create, assign or modify time Start: End (military time 24/hrs.), to let members know when specialized equipment is available, This will also be shown through Booking Schedule and Facility class information. This will loop every week unless changed by staff.

*Inherit Everything from Parent Class (Equipment)  
But also add info about availability of SpecialEquip*

*Example:*

```
SpecialEquip Se1 = new SpecialEquip();  
Se1.name = Under Water Treadmill
```

*Availability: Shown through booking and Facility class*

```
Under Water Treadmill 1122  
09/20/2025 14:00 to 15:00  
1 out of 1
```

*//Repeat Until changed, will always be available during these times  
until changed by staff*

## **10. Facility Management (Class Facility)**

Allows access to information about the facility, such as rooms, facilities, Equipment/Specialized Equipment and available classes (fitnessClasses) that can be booked through the **Booking Class**.

**Returns information** about each section of the facility to members, traffic of each section (capacity).

**Staff** have access to **modify, delete, pause** sections (indefinitely close sections for cleaning or other), **add** new sections and **remove** sections of the facility.

## **11. Booking Management (Class Booking)**

The System must **return the sessions available within the week, add or remove members from session or Specialized Equipment** (a schedule of available classes and specialized equipment available), **Sessions will be received and added to schedule accordingly to times and dates**(schedule sessions, reschedule sessions), members will book; sign up through Booking Class, if member meets membership criteria and capacity from fitness Classes Class is not max, Booking will **add or remove members for sessions and specialized equipment**, **Changes** will be shown in **Booking Class Schedule** and **Facility Class Information**, if class is full Booking will not be permitted. Special equipment has the capacity of only one member per piece of specialized equipment.

*Booking Information Stored: System Must store members' information, those who have signed up for those booked sessions (Specialized Equipment and fitClass).*

*Example: of Booking Management*

*02/17/2026 13:00 to 14:00 <---- Date and time  
Zumba 1234 <--- Class Name, Class ID  
Instructor: Jhon Deer <--- Instructor  
10 out of 20 <--- Capacity = Capacity +1*

*Tanning bed 1212 <--Special Equipment Name and ID  
02/09/2026 13:00 to 14:00 <-- time and date  
1 out of 1 <-- Capacity Full*

## **12. Payment Management Class (Payment Class)**

The System must **verify that the credit card information is correct** (is True) and valid, System must allow the **modification of payment**, System must **create and return** an itemized receipt for time frame: **Start of Membership to End of Membership** (how much member paid, the total). **If member(s) misses' payment membership will be automatically revoked, and member will attain a negative balance which will not allow member to enter the gym, unless balance is paid, he will no longer be a member of the gym and access information about the gym. If a member ends his membership while having a negative balance, member must pay negative balance, or he will be continuously charged until done so.**

*Example:*

*Member: Jhon Cena  
Membership: Premium*

*January  
Premium Membership 10.99 + tax  
Total 11.89  
Charged to Card Ending in 1234  
And so on...*

### 13. Check-in Class (Check-in Class)

Allows members to Check in with a SetPin (Created when putting information of the Specific Member), Check-in class tracks attendance of Member to the Gym and can return attendance of member for a specific month, Check-in also allows members to modify their SetPin if they wish to do so.

Would like to have a Stat tracker (diagram) that monitors traffic with using the check in information

February Attendance

Monday 02/01/27	Tuesday 02/02/27	Wednesday 02/03/27	Thursday 02/04/27	Friday 02/05/27
Check in: 12:49	** if no attendance leaves empty**	Check in: 18:32	Check in: 7:27	**Empty**