

# **HEALTH AND FITNESS CLUB MANAGEMENT SYSTEM**

**COMP 3005 A - Final Report**

**Group 4**

*Completed by:*

*Rimsha Atif - 101187263*

*Nermeen El-Sherbiny - 101263616*

*Vishrutha Gopa - 101267228*

## 2.1 CONCEPTUAL DESIGN

The ER Model and assumptions table below explains the design of our database, including the assumptions, entities, cardinalities, and participation types.

### ER Model Link

- <https://online.visual-paradigm.com/share.jsp?id=333131383230342d38>
- <https://github.com/VishruthaGopa/Health-and-Fitness-Club-Management-System/blob/main/Diagrams/Entity%20Relationship%20Diagram.jpg>

**Table 1: Mapping and Assumption Table**

Dotted line represents partial participation

Solid line represents the total participation

Representation in ER Model	Extra Information
<b>User relation:</b> A User relation used to store the user_id(PK), username, password and the role used to differentiate between the roles and responsibilities of the different users	This will mainly be for the login and registration page, which we will use to delegate certain accessibility features depending on their role
<b>Member relation:</b> stores member_id (PK), first_name, last_name, gender, email, date_of birth, address, phone_number, start_date, Payment_status.	This keeps track of member information. The payment status is used to keep track of if a member paid the starting fee. Members are the only users allowed to create an account.
<b>Member_Health relation:</b> Stores member_id (PK), start_weight, current_weight, height, age, fitness_goal(fitness_id, weight_goal, time_goal, diet_goal, form_of_exercise)	This keeps track of individual member goals and current health. Members are allowed to update these, and based on this we can show them their progress based on how far along they are. The form of exercise includes things like cardio, strength training, etc.
<b>Exercise_Routine relation:</b> Stores Routine_id (PK), Member_id (FK), Routine_name, description, duration, date_created	This allows a member to create a personalized workout routine so that they can stay on track for their goals.

<b>Admin Relation:</b> Stores admin_id (PK), first_name, last_name, email, phone_number	
<b>Trainer relation:</b> Stores trainer_id (PK), first_name, last_name, email, phone_number, specialisations	This stores the trainer's information. The specialization is a multi-valued attribute that will allow us to pair the right trainers with the right members.
<b>Personal_training_session relation:</b> Stores: Session_id (PK), Trainer_id (FK), Member_id (FK), session_date, session_time, duration, Room_id (FK), payment(Price, payment_status), Booking Status.	For personal training sessions, personal trainers can add sessions based on their availability. Trainers choose the date, time, room, and price for each session. Member IDs are initially set to a default value of null until a member signs up and pays. Trainers can cancel sessions. Admins have the authority to cancel sessions and update room assignments. Members must complete payment before they are approved for the session.
<b>Group_fitness_classes relation:</b> Stores Class_id (PK), Trainer_id (FK), Room_id, class_name, description, Session_date, Session_time, Member_ids, price	Similar to the idea of personal training, trainers can add classes based on their availability, selecting the date, time, room, and class name with a description. These classes are free for members and included in the membership fee, Members can sign up for a class and a list of who has signed up is stored in the member_ids attribute which is multi-valued. Trainers can cancel sessions.
<b>Room_bookings relation:</b> Stores Room_id (PK), Room_name, Room_location, booked(timing, duration, date, session_id/class_id)	For room bookings, all available rooms for 3 days from the time 4 pm-9 pm with intervals of 1 hour, and the status of their booking. Trainers can pick a room for their classes from the available slots.

<p><b>Equipment_maintenance relation:</b></p> <p>Stores equipment _id (PK), Equipment_name, maintained_date, next_maintenace, performed_by</p>	
<p><b>Equipment maintenance relationship:</b></p> <p>Equipment is maintained by admin, and all equipment must be maintained, but not all admin must maintain equipment. It is an N:M relationship because an employee can maintain many pieces of equipment and a piece of equipment can be maintained by many employees.</p>	<p>Admin can update, delete and create new equipment and manage the maintenance of the equipment.</p>
<p><b>Room management relationship:</b></p> <p>Admins can manage and oversee room bookings. It's an N:M relationship and total participation on the side of the rooms because all rooms must be managed but not all admin have to manage them.</p>	<p>Admins can free up a room by cancelling a class or switching the rooms in which classes are taking place</p>
<p><b>Trainer book relationship:</b></p> <p>Trainers book rooms for their personal and group sessions. It's a 1:N relationship, with total participation on both sides. Only one trainer per room, but a trainer can book multiple rooms.</p>	
<p><b>Members register relationship:</b></p> <p>Members register for personal training/group class sessions. Not all members have to</p>	

<p>register and all classes have to be registered for or will be cancelled. One member can sign up for many personal fitness/group training sessions, but only one person can sign up for a personal training session and many people can sign up for one group fitness class..</p>	
<p><b>Member create exercise_routine relationship:</b></p> <p>Each member can create many exercise routines and each exercise routine is created by one member, thus, it's a 1:N relationship. Also, all exercise routines must be created by members, but not all members must create an exercise routine, thus a partial participation of the members</p>	
<p><b>Member update member_health relationship:</b></p> <p>1:1 relationship and total participation on both sides, because every member must update their health, and goals and can only do one “profile”, which consists of a goal, and all health is updated by one member and must be created by a member.</p>	
<p><b>Trainer lead group_fitness_class:</b></p> <p>Every trainer can lead many group fitness classes and every fitness class is led by one trainer, thus a 1:N relationship. Total participation on both sides</p>	

<p><b>Trainer conduct personal_fitness_classes:</b></p> <p>Every trainer can conduct many personal fitness classes and every fitness class is led by one trainer, thus a 1:N relationship. Total participation on both sides.</p>	
<p><b>Admin updates classes:</b></p> <p>Admin has the power to update the classes as they see fit, which usually just means cancelling and changing the location for the classes. Not all classes have to be updated and not all admins have to update, but only one admin can cancel a class and they can cancel multiple classes</p>	

## 2.2 RELATIONAL DATABASE

The following is a link to our relational database schema. This schema is derived from our entity diagram from section 2.1 of the report.

- <https://online.visual-paradigm.com/share.jsp?id=333135323634362d32>
- <https://github.com/VishruthaGopa/Health-and-Fitness-Club-Management-System/blob/main/Diagrams/Relational%20Database%20Schema.vpd.png>

## 2.3 DDL FILE

The DDL used to create our database is available in the SQL directory of our GitHub repository. Access it here:

<https://github.com/VishruthaGopa/Health-and-Fitness-Club-Management-System/blob/main/SQL/ddl.sql>

## 2.4 DML FILE

The DDL used to create our database is available in the SQL directory of our GitHub repository. Access it here:

<https://github.com/VishruthaGopa/Health-and-Fitness-Club-Management-System/blob/main/SQL/dml.sql>

## **2.5 IMPLEMENTATION**

The Health and Fitness Club Management System is a web-based application that meets the needs of members, trainers, and admin of the club. The application was built using the Django framework linked to a PostGres database to store application information. For the backend, python language is used to program all the logic of the application and the front end of the application is designed using HTML templates and CSS styles to create an inviting and user-friendly interface.

## **2.6 BONUS FEATURES**

The bonus feature provided by our web application is the recommendation system in our personal training section. Members have the option to create a current fitness goal in their dashboard. One of the attributes of the fitness goals is a 'form of exercise'. Based on the form of exercise specified for the current fitness goal, the member can be recommended a trainer who specializes in the form of exercise.

## **2.7 GITHUB REPOSITORY**

This is a link to our GitHub repository, containing all the source code, SQL files, diagrams, and the README for our project:

<https://github.com/VishruthaGopa/Health-and-Fitness-Club-Management-System>