Get SWOLE Fitness Club

COMP 3005 Final Project Report

By: Abdullah Soboh (101220742) & Imann Brar (101225891)

Instructors: Ahmed El-Roby and Abdelghny Orogat

Due: April 13, 2024 (11:59 PM)

Executive Summary

This report documents the design and creation of a Health and Fitness Club Management System. The system centralizes the management of member profiles, training schedules, and administrative tasks. Aimed at simplifying the operations of a fitness club, it offers a robust solution for member engagement and administrative efficiency.

System Overview

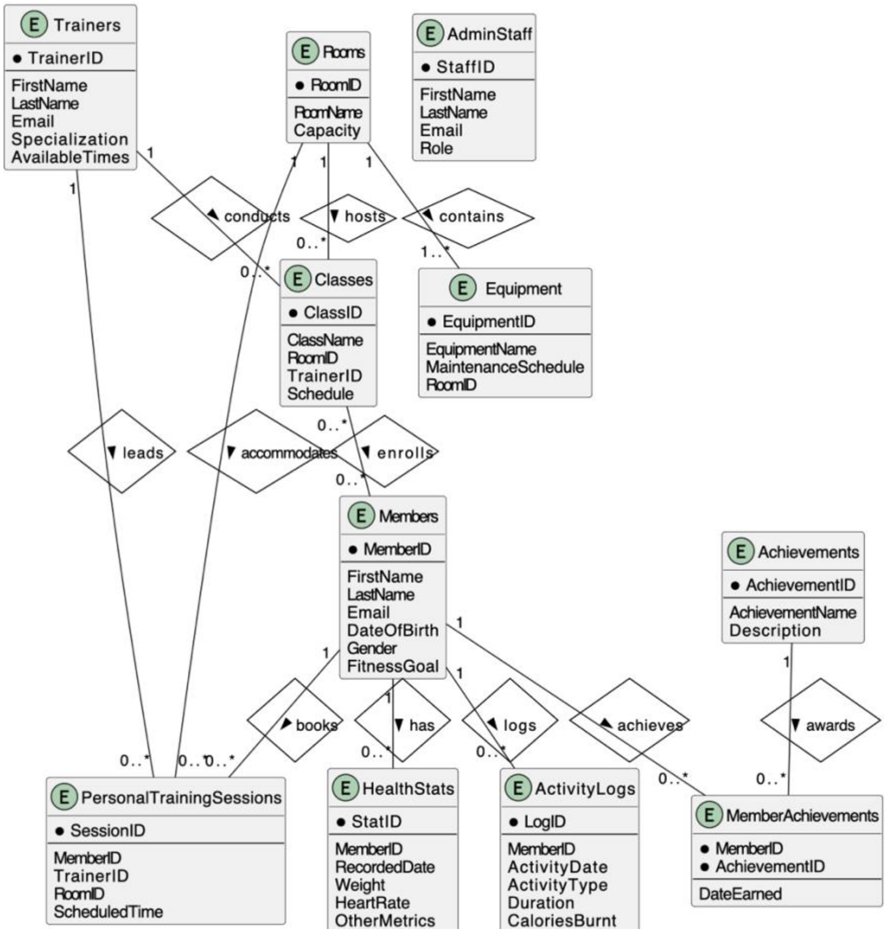
The Health and Fitness Club Management System is a versatile platform developed to meet the diverse requirements of members, trainers, and staff. Members can easily sign up, update their profiles, set personal fitness goals, and track their health and fitness achievements. The system allows for smooth scheduling of personal training sessions and fitness classes. Trainers have access to manage their availability and view member information for tailored services. Administrative staff can oversee room bookings, maintain equipment, update schedules, and manage billing.

Database Design

The system's foundation is a relational database that organizes and stores all necessary data. The database structure was conceived with efficiency and scalability in mind, allowing for easy updates and data retrieval.

2.1 Conceptual Design

The conceptual design of the Get SWOLE Fitness Club’s Management System's database is articulated through an Entity-Relationship (ER) diagram, which visually represents the data and its interconnections. The diagram details the entities involved in the system and their relationships, serving as a blueprint for the database's architecture.
Entities and Their Attributes



The ER diagram encompasses several primary entities:

1. **Members**: Individuals who join the club and use its services.
2. **Trainers**: Certified professionals providing personal training.
3. **AdminStaff**: Operational staff managing the club's administrative tasks.
4. **Rooms**: Physical spaces within the club used for various activities.
5. **Equipment**: Items available for members and trainers to use.
6. **Classes**: Scheduled group fitness activities.
7. **PersonalTrainingSessions**: One-on-one training appointments.
8. **ActivityLogs**: Records of members' workout sessions.
9. **Achievements**: Notable milestones members can achieve.
10. **HealthStats**: Health-related data points tracked over time for each member.

Assumptions:

Trainer Assumptions:

- Trainers provide a monthly updated schedule that defines their availability.
- They manage various types of training sessions and handle numerous members simultaneously, ensuring no overlapping sessions occur.

Administrative and Booking Assumptions:

- Administrative tasks involve managing multiple room bookings and maintaining equipment schedules concurrently.
- Sessions are only scheduled when both the trainer and the room are available, adhering strictly to room capacity limits.
- Manual adjustments are necessary for billing exceptions, although regular billing processes are automated.

Equipment and Maintenance Assumptions:

- Regularly scheduled maintenance is presumed to keep equipment operational, with unscheduled breakdowns managed individually.

Membership Assumptions:

- Membership requires individual registration with personal details, prohibiting account sharing.
- Members can modify their fitness goals over time and are expected to self-report health metrics periodically.

Billing Assumptions:

- All membership fees and financial transactions are processed through an assumed electronic integration with a payment service, ensuring efficient and secure handling of payments.

2.2 Relationships and Cardinalities

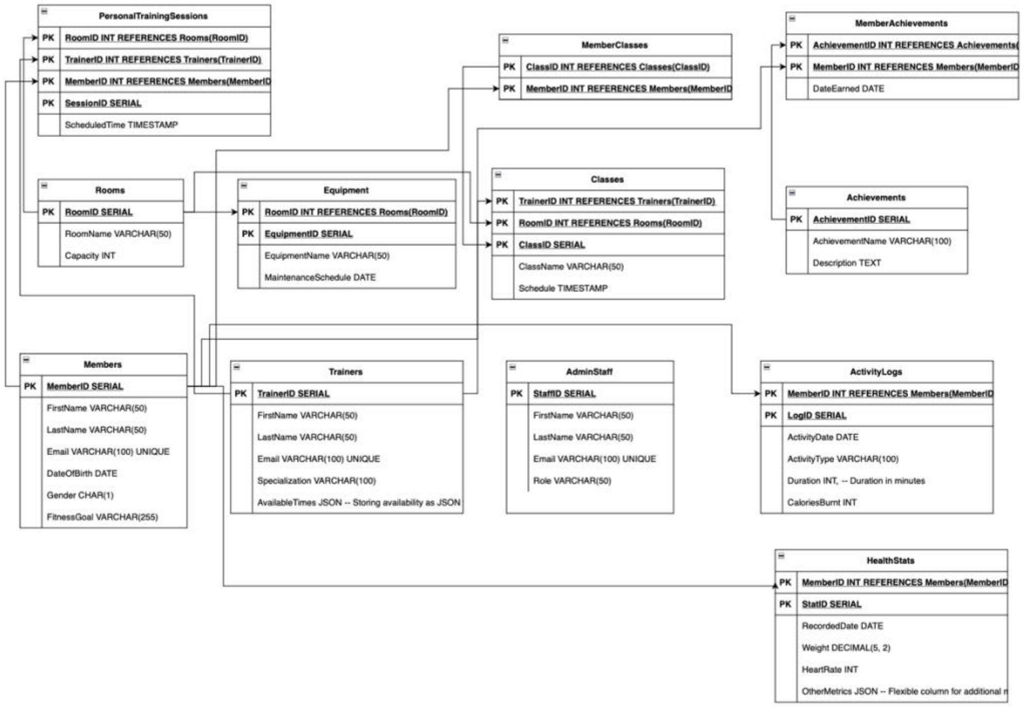
The relationships between these entities are characterized by specific cardinalities:

- **Members to PersonalTrainingSessions**: One-to-many. Each member can book multiple training sessions, but each session is with one member.
- **Members to Classes**: Many-to-many. Members can enroll in multiple classes, and each class can host multiple members.
- **Trainers to Classes**: One-to-many. A trainer can lead multiple classes, but each class is led by one trainer.
- **Trainers to PersonalTrainingSessions**: One-to-many. A trainer can conduct several personal training sessions.
- **AdminStaff to Rooms**: One-to-many. Administrative staff manage bookings for each room, but each room is managed by the club as a whole, not individual staff members.

- **Rooms to Equipment**: One-to-many. Each room can contain multiple pieces of equipment.

Relational Schemas

The ER components were reduced to relational schemas to structure the database effectively. Tables were designed with primary keys, establishing clear connections between different data points, such as Member Achievements and Training Sessions.


PersonalTrainingSessions
PK RoomID INT REFERENCES Rooms(RoomID)
PK TrainerID INT REFERENCES Trainers(TrainerID)
PK MemberID INT REFERENCES Members(MemberID
PK SessionID SERIAL
ScheduledTime TIMESTAMP
MemberClasses
PK ClassID INT REFERENCES Classes(ClassID)
PK MemberID INT REFERENCES Members(MemberID
MemberAchievements
PK AchievementID INT REFERENCES Achievements(
PK MemberID INT REFERENCES Members(MemberID
DateEarned DATE
Rooms
PK RoomID SERIAL
RoomName VARCHAR(50)
Capacity INT
Equipment
PK RoomID INT REFERENCES Rooms(RoomID)
PK EquipmentID SERIAL
EquipmentName VARCHAR(50)
MaintenanceSchedule DATE
Classes
PK TrainerID INT REFERENCES Trainers(TrainerID)
PK RoomID INT REFERENCES Rooms(RoomID)
PK ClassID SERIAL
ClassName VARCHAR(50)
Schedule TIMESTAMP
Achievements
PK AchievementID SERIAL
AchievementName VARCHAR(100)
Description TEXT
Members
PK MemberID SERIAL
FirstName VARCHAR(50)
LastName VARCHAR(50)
Email VARCHAR(100) UNIQUE
DateOfBirth DATE
Gender CHAR(1)
FitnessGoal VARCHAR(255)
Trainers
PK TrainerID SERIAL
FirstName VARCHAR(50)
LastName VARCHAR(50)
Email VARCHAR(100) UNIQUE
Specialization VARCHAR(100)
AvailableTimes JSON -- Storing availability as JSON
AdminStaff
PK StaffID SERIAL
FirstName VARCHAR(50)
LastName VARCHAR(50)
Email VARCHAR(100) UNIQUE
Role VARCHAR(50)
ActivityLogs
PK MemberID INT REFERENCES Members(MemberID
PK LogID SERIAL
ActivityDate DATE
ActivityType VARCHAR(100)
Duration INT, -- Duration in minutes
CaloriesBurnt INT
HealthStats
PK MemberID INT REFERENCES Members(MemberID
PK StatID SERIAL
RecordedDate DATE
Weight DECIMAL(5, 2)
HeartRate INT
OtherMetrics JSON -- Flexible column for additional r

2.3 DDL File

```sql
-- Members Table
CREATE TABLE Members (
   MemberID SERIAL PRIMARY KEY,
   FirstName VARCHAR(50),
   LastName VARCHAR(50),
   Email VARCHAR(100) UNIQUE,
   DateOfBirth DATE,
   Gender CHAR(1),
   FitnessGoal VARCHAR(255)
);

-- Trainers Table
CREATE TABLE Trainers (
   TrainerID SERIAL PRIMARY KEY,
   FirstName VARCHAR(50),
   LastName VARCHAR(50),
   Email VARCHAR(100) UNIQUE,
   Specialization VARCHAR(100),
   AvailableTimes JSON  -- Storing availability as JSON for flexibility
);

-- Administrative Staff Table
CREATE TABLE AdminStaff (
   StaffID SERIAL PRIMARY KEY,
   FirstName VARCHAR(50),
   LastName VARCHAR(50),
   Email VARCHAR(100) UNIQUE,
   Role VARCHAR(50)
);

-- Rooms Table
CREATE TABLE Rooms (
   RoomID SERIAL PRIMARY KEY,
   RoomName VARCHAR(50),
   Capacity INT
);

-- Equipment Table
CREATE TABLE Equipment (
   EquipmentID SERIAL PRIMARY KEY,
   EquipmentName VARCHAR(50),
   MaintenanceSchedule DATE,
   RoomID INT REFERENCES Rooms(RoomID)
);

-- Classes Table
CREATE TABLE Classes (
   ClassID SERIAL PRIMARY KEY,
   ClassName VARCHAR(50),
   RoomID INT REFERENCES Rooms(RoomID),
   TrainerID INT REFERENCES Trainers(TrainerID),
   Schedule TIMESTAMP
);

-- Personal Training Sessions Table
CREATE TABLE PersonalTrainingSessions (
   SessionID SERIAL PRIMARY KEY,
    MemberID INT REFERENCES Members(MemberID),
    TrainerID INT REFERENCES Trainers(TrainerID),
    RoomID INT REFERENCES Rooms(RoomID),
    ScheduledTime TIMESTAMP
);

-- Member Classes Many-to-Many Relationship Table
CREATE TABLE MemberClasses (
    MemberID INT REFERENCES Members(MemberID),
    ClassID INT REFERENCES Classes(ClassID),
    PRIMARY KEY (MemberID, ClassID)
);

-- Activity Logs Table
CREATE TABLE ActivityLogs (
    LogID SERIAL PRIMARY KEY,
    MemberID INT REFERENCES Members(MemberID),
    ActivityDate DATE,
    ActivityType VARCHAR(100),
    Duration INT,  -- Duration in minutes
    CaloriesBurnt INT
);

-- Achievements Table
CREATE TABLE Achievements (
    AchievementID SERIAL PRIMARY KEY,
    AchievementName VARCHAR(100),
    Description TEXT
);

-- Member Achievements Many-to-Many Relationship Table
CREATE TABLE MemberAchievements (
    MemberID INT REFERENCES Members(MemberID),
    AchievementID INT REFERENCES Achievements(AchievementID),
    DateEarned DATE,
    PRIMARY KEY (MemberID, AchievementID)
);

-- Health Stats Table
CREATE TABLE HealthStats (
    StatID SERIAL PRIMARY KEY,
    MemberID INT REFERENCES Members(MemberID),
    RecordedDate DATE,
    Weight DECIMAL(5, 2),
    HeartRate INT,
    OtherMetrics JSON  -- Flexible column for additional metrics
);
```

2.4 DML File

```
-- Insert Members
INSERT INTO Members (FirstName, LastName, Email, DateOfBirth, Gender, FitnessGoal) VALUES
('John', 'Doe', 'johndoe@example.com', '1985-07-10', 'M', 'Increase Activity Level'),
('Jane', 'Smith', 'janesmith@example.com', '1992-11-30', 'F', 'Maintain Health'),
('Alice', 'Johnson', 'alicejohnson@example.com', '1988-05-16', 'F', 'Improve Fitness');

-- Insert Trainers
INSERT INTO Trainers (FirstName, LastName, Email, Specialization, AvailableTimes) VALUES
('Bob', 'Builder', 'bobbuilder@example.com', 'Strength Training', '{"Monday": ["08:00-12:00"], "Wednesday": ["14:00-18:00"]}'),
('Nancy', 'Rogers', 'nancyrogers@example.com', 'Yoga Instructor', '{"Tuesday": ["10:00-14:00"], "Thursday": ["16:00-20:00"]}');

-- Insert Admin Staff
INSERT INTO AdminStaff (FirstName, LastName, Email, Role) VALUES
('Bruce', 'Wayne', 'brucewayne@example.com', 'Manager'),
('Clark', 'Kent', 'clarkkent@example.com', 'Receptionist');

-- Insert Rooms
INSERT INTO Rooms (RoomName, Capacity) VALUES
('Aerobics Room', 20),
('Weight Room', 15);

-- Insert Equipment
INSERT INTO Equipment (EquipmentName, MaintenanceSchedule, RoomID) VALUES
('Treadmill', '2024-04-15', 2),
('Dumbbells', '2024-05-10', 2);

-- Insert Classes
INSERT INTO Classes (ClassName, RoomID, TrainerID, Schedule) VALUES
('Morning Yoga', 1, 2, '2024-06-01 08:00:00'),
('Weight Training', 2, 1, '2024-06-01 09:00:00');

-- Insert Personal Training Sessions
INSERT INTO PersonalTrainingSessions (MemberID, TrainerID, RoomID, ScheduledTime) VALUES
(1, 1, 2, '2024-06-03 10:00:00'),
(2, 2, 1, '2024-06-04 11:00:00');

-- Insert Member Classes (Many-to-Many Relationship)
INSERT INTO MemberClasses (MemberID, ClassID) VALUES
(1, 1),
(2, 2);

-- Insert Activity Logs
INSERT INTO ActivityLogs (MemberID, ActivityDate, ActivityType, Duration, CaloriesBurnt) VALUES
(1, '2024-06-02', 'Running', 30, 300),
(2, '2024-06-02', 'Cycling', 45, 450);

-- Insert Achievements
INSERT INTO Achievements (AchievementName, Description) VALUES
('Marathon Finisher', 'Completed a full marathon'),
('Yoga Master', 'Participated in 100 yoga sessions');

-- Insert Member Achievements
INSERT INTO MemberAchievements (MemberID, AchievementID, DateEarned) VALUES
(1, 1, '2024-06-01'),
(2, 2, '2024-06-01');

-- Insert Health Stats
INSERT INTO HealthStats (MemberID, RecordedDate, Weight, HeartRate, OtherMetrics) VALUES
(1, '2024-06-01', 85.0, 72, '{"blood_pressure": "120/80"}'),
(2, '2024-06-01', 65.0, 68, '{"blood_pressure": "110/70"}');
```

2.4 Implementation

The Get SWOLE Fitness Club Management System is developed as a full-stack application using React for the frontend and Flask/Python for the backend, providing a dynamic and responsive user interface. React facilitates efficient state management and real-time user interaction, while Flask serves as a lightweight backend framework, handling API requests and integrating smoothly with databases. This setup ensures seamless full-stack connectivity and robust data handling. The application's architecture is designed with modularity in mind, structuring the frontend and backend into distinct components that correspond to different system functionalities.

2.4.1) Bonus Implementation

In developing the Get SWOLE Fitness Club Management System, we went beyond the basic requirements by creating a full-stack application using React and Flask. This choice not only streamlined the system's operation but also enhanced its user-friendliness. Additionally, we incorporated extra functionality that provides members with detailed statistics on their fitness progress. This feature allows members to effectively track their performance, identify areas for improvement, and set more targeted fitness goals. By integrating these capabilities, we have significantly boosted the system's efficiency, making it easier for members to engage with and benefit from the platform. This demonstrates our commitment to delivering a practical and effective management tool.

EXTRA)

Features and Functions

The system comprises various functions that enable the operations mentioned in the project requirements. Notably:

Member Functions:

- User Registration: Enables new members to sign up and join the club, providing access to club facilities and services.
- Profile Management: Members can update their personal information, set and adjust fitness goals, and track health metrics to personalize their fitness journey.
- Dashboard Display: Provides a personal dashboard where members can view their exercise routines, track fitness achievements, and monitor health statistics, helping them stay motivated and informed.

- Schedule Management: Allows members to book and manage their participation in personal training sessions or group fitness classes, with system checks to ensure trainer availability.

Trainer Functions:

- Schedule Management: Trainers can manage their schedules, setting available times for sessions, which integrates with the member booking system.
- Member Profile Viewing: Trainers can access and view member profiles by searching by member's name, which aids in personalized training plans.

Administrative Staff Functions:

- Room Booking Management: Facilitates the reservation and scheduling of rooms within the facility for various activities and events.
- Equipment Maintenance Monitoring: Enables the tracking and management of gym equipment maintenance to ensure safety and functionality.
- Class Schedule Updating: Staff can update and modify class schedules to reflect changes in trainer availability or room usage.
- Billing and Payment Processing: Handles financial transactions related to membership fees and services, assuming integration with an external payment service for processing payments, though actual integration is not implemented.

2.6 Conclusion

In conclusion, the Get SWOLE Fitness Club Management System, developed for COMP 3005, represents a robust integration of full-stack technologies with React and Flask driving the front and back ends, respectively. This system significantly enhances the operational efficiency of health and fitness clubs by centralizing the management of member profiles, training schedules, and administrative tasks. Designed with a focus on user engagement and administrative ease, it offers a comprehensive suite of features that cater to the distinct needs of members, trainers, and administrative staff. This modular approach not only ensures seamless functionality and ease of use but also facilitates future scalability and maintenance, positioning the Get SWOLE Fitness Club as a leader in its class for management solutions.