

Conceptual Design

Members entity is related to weak entities (Health Metrics, Personal Information, Fitness Goals, Fitness Achievements, Exercise Routines), because all of these tables rely on the member_id to be identified

Members has (Health Metrics and Personal Information) is a one to one relationship and has total participation on both sides because when the user is registered they are automatically added to these tables

Members has (Fitness Goals, Fitness Achievements, Exercise Routines) is a one to many relationship, member can have many goals etc, and is a partial participation on the members part because they don't have to be in any of those tables until they add information, but

Fitness goals, exercise routines, and fitness achievements are full participation because they need a member to exist.

Members pays Billings entity is a one to one relationship with total participation on both sides because a member has to pay, it's a weak entity because it relies on the member_id to be identified

Members can schedule group classes (Group Classes Entity) in a many to many relationship, classes can have multiple members and members can be in multiple classes. They both are partial participation as they do not rely on each other

Members can schedule private sessions (Trainer Schedule) in a many to many relationship, trainers can have multiple members/classes and members can be in multiple private sessions. They both are partial participation as they do not rely on each other

Staff Entity can either represent an administrator or a trainer based on the type attribute, they all have a unique id staff_id

Staff Entity interacts with the Group Class Entity because a trainer can run a class, it is a one to many relationship, because a trainer can run multiple classes and it is partial participation because a trainer does not need to run a class, but is total participation for the class because a class cannot run without a trainer

The admin type from Staff Entity has the same relationships with the Group Class Entity, for the same reasons except that the admin schedules the classes instead of runs it

Each Group Class will have members in it and is a weak identity because it needs the class identity in order to identify it, it's many to many relationship and members in the Class Members entity require full participation because they've signed up for a class, but a class does not need members to be scheduled so it is partial participation.

Staff admins can book rooms, and it's many to many, because rooms can be booked by multiple admins and any admin can book a room, room bookings is total participation because it cannot exist without an admin creating it, but not every admin needs to make a room booking so it's partial participation

Staff admin manages Equipment Entity, equipment entity has a unique identifier equipment id and each equipment has a last maintenance month, it's many to many, equipment has full participation with the fitness club

Health Metrics, Personal Information, and Billings don't have primary keys in the ER because they are one to one relationships and rely on the members_id to uniquely identify the tuples within the table.

Assumptions

We assumed that scheduling classes and private sessions were to happen a day in advance. So members could only ever see the next day's schedule.

We assumed that private sessions and group classes are only 1 hour long and happen on the hour, room bookings as well.

We used military time to represent the hours between 8 AM and 8 PM and classes and private sessions could be booked on the hour.

We assumed that members can track their fitness goals and fitness achievements so they can update their profile when they have a new goal or they want to record an achievement they had

We assumed that equipment will just have their last maintenance time in terms of months and the admin can update the equipment based on the newest month that the equipment was updated, so it tracks the most recent maintenance day

We let the user add their own specific exercises to their routines

We assumed that an admin can book a room for whatever they want and they can manually input the room number they wish to book, due to giving the admin the freedom to choose any room, we did not track all the room numbers in the fitness club

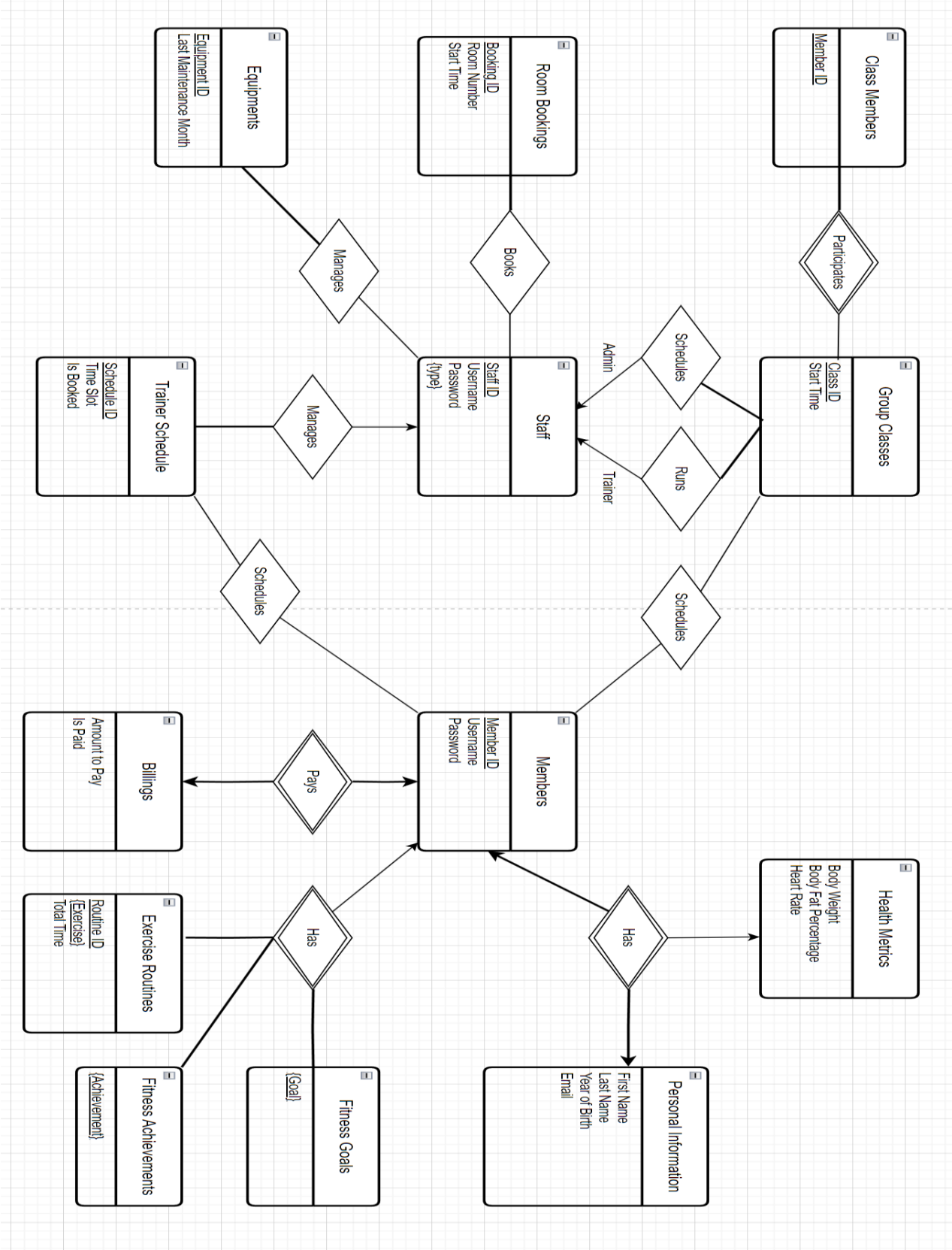
Trainer and admin will share the attribute `staff_id` and based on the type attribute the application will give them their corresponding interface, that way they can't use each other's functionality

When the user registers, they need to manually go into their profile and add their information

Implementation

The code was implemented using a java program using a text interface all into a single file. It connected to the postgresql database and implemented SQL queries from the application.

ER DIAGRAM



Relation Schema



Github Link

https://github.com/kevinismugin/Fitness_Club_Database

YouTube Link

<https://www.youtube.com/watch?v=QxHOemU-SIE>