Health and Fitness Club Management System - Project Report

COMP3005 - Dr. Abdelghny Orogot

Abaseen Ahmadzai - 101228053 & Talha Ather - 101230176

April 13, 2024

2.1 Conceptual Design of the system:

For this system, we took a more terminal based approach as to how we wanted to implement this, and due to this, it shaped the way our classes would be, as well as the number of classes. We knew that if we wanted a terminal where it would display everything according to whichever user is logged in, the classes must also conform to that and establish a way for us to utilize them to the best of their abilities, to continue with good design and good database implementation practices.

Project Requirements:

The Health and Fitness Club Management System is a database-driven application which manages basically the operation within any well established fitness club. This means that all the different users, in our case 3: members, trainers and administrative staff, can manage different aspects of their fitness goals/ what a fitness facility needs in order to be run properly. This includes training sessions, room bookings and billings. Detailed below is a further in-depth functionality for each type of user.

Members

- Members have the ability to register themselves inside the fitness system, what this means is they must have a unique username to register. They also can join either solo or group sessions with a trainer, display their fitness goals as well as their health metrics

Trainers

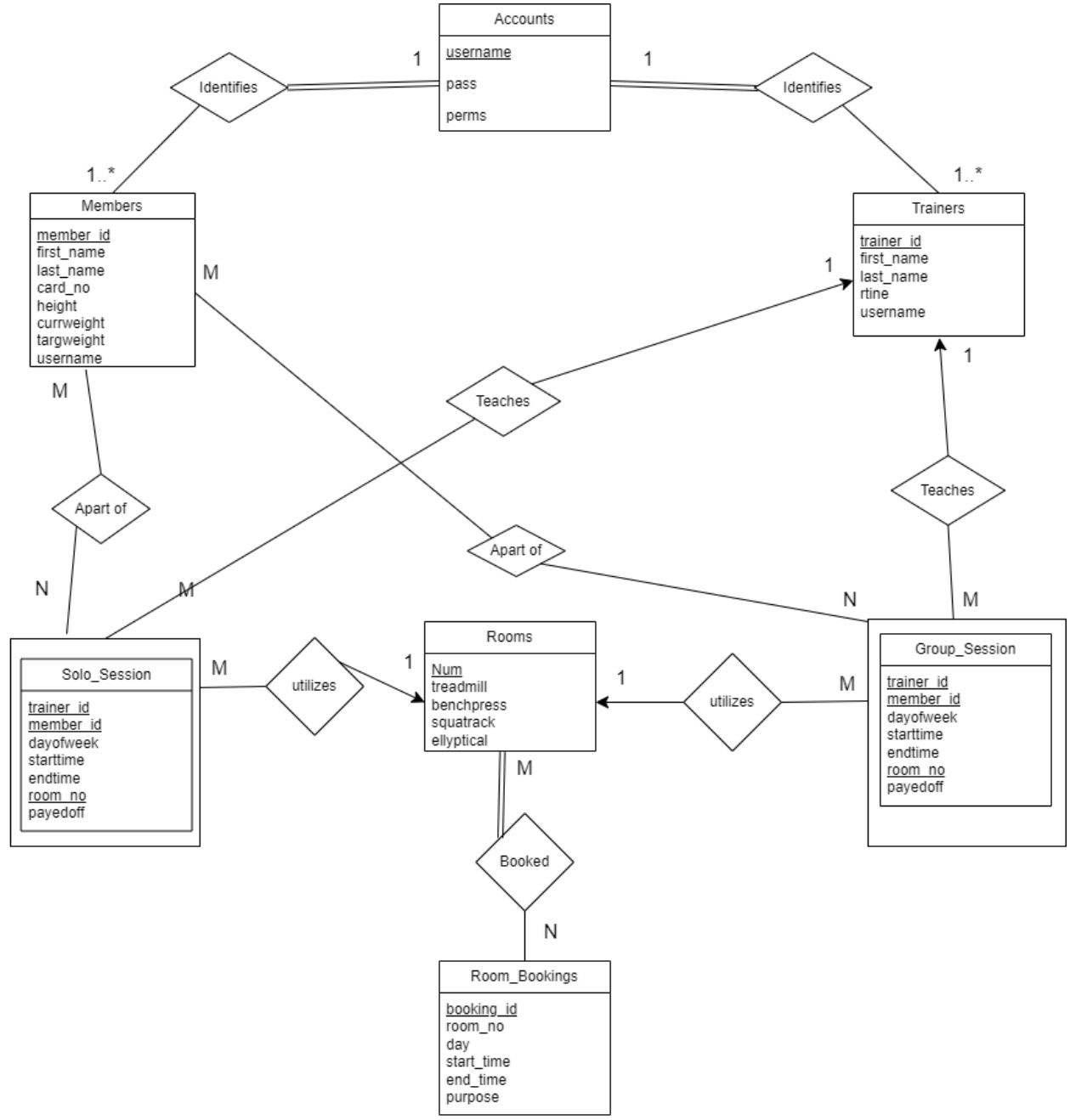
- Trainers can log their class availability in the system to let members know when they are available for sessions, every session is taught by one trainer. They are also able to view the members either in sessions or throughout.

Administrative Staff

- The staff maintains the internal system of the club, by managing equipment, room bookings, and being able to view all the available sessions, as well as billing members for sessions

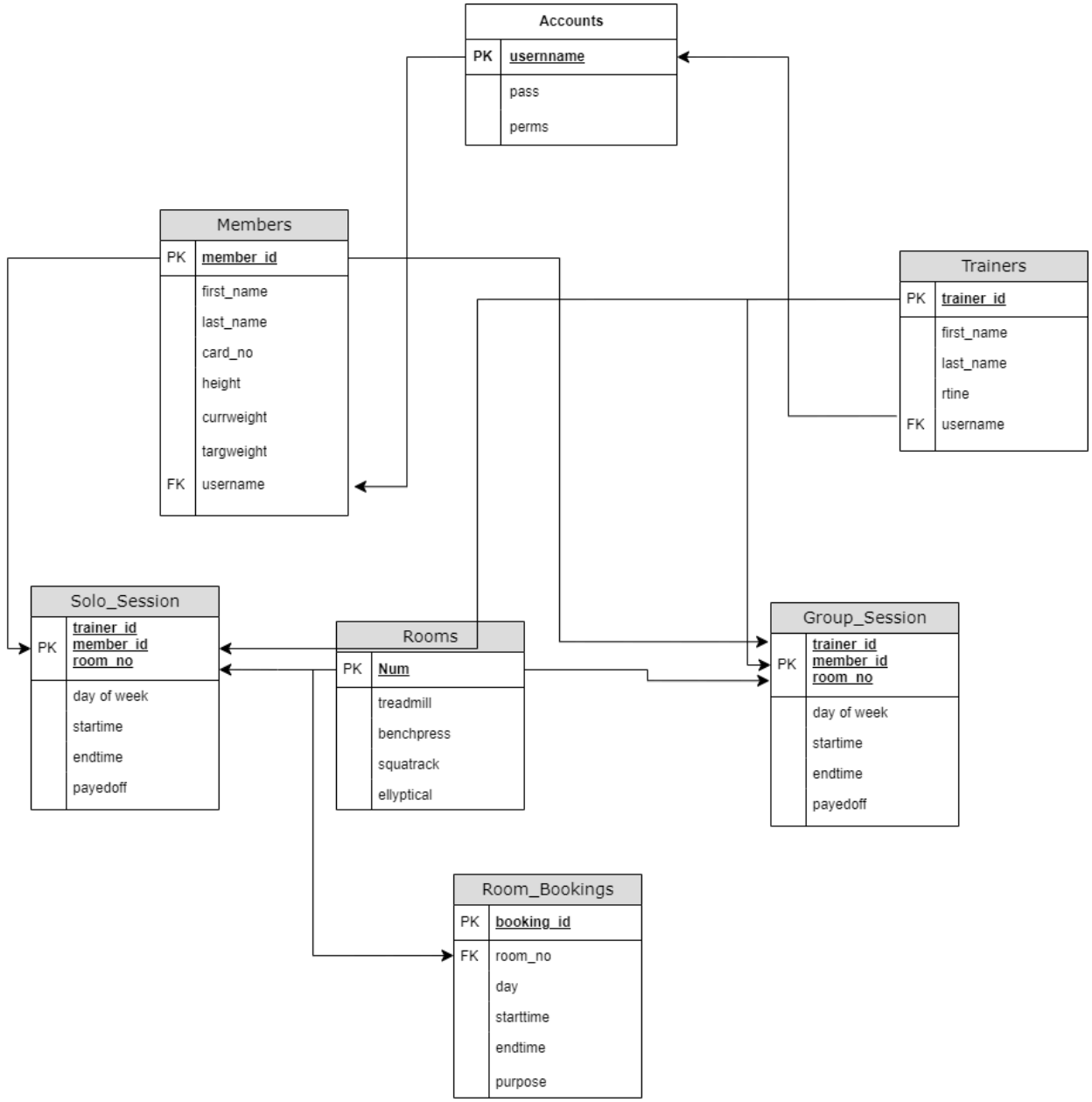
2.2 Reduction to Relation Schemas

ER Model:



(should be a 1 instead of N above the Room_Bookings)

Reduction to Relation Schema:



Assumptions and Explanations of ER Model:

Accounts identifies Members relation:

- This is a one-to-many relationship, seeing as we assumed it would make sense that each member only has one account, but there are many member accounts in the system, as well as accounts that aren't members (trainers and staff).
- There is full participation since every member (if logged in which we assume) has an account or registers for one

Accounts identifies Trainers relation:

- This is a one-to-many relationship, seeing as we assumed it would make sense that each trainer only has one account, but there are many trainer accounts in the system, as well as accounts that aren't trainers (members and staff).
- THere is full participation since every trainer (assumed to be logged in) would have an account made for them by the admin staff

Members are Apart of Solo_Session relation:

- This is a many-to-many relationship due to how there can be many solo-sessions and members can be apart of as many solo sessions as are available
- There is partial participation since if a session exists that it can be made up of a member and a trainer
- The class Solo_session is considered a weak entity as it doesn't have a unique identifier and these primary keys can be used to identify other similar sessions

Members are Apart of Group_Session relation:

- This is a many-to-many relationship due to how there can be many group-sessions and members can be apart of as many group sessions as they want
- There is partial participation since if a session exists then it can be made up of a member and a trainer
- The class Group_session is considered a weak entity as it doesn't have a unique identifier and these primary keys can be used to identify other similar sessions

Trainers Teaches Solo_Session relation:

- THis is a One-to-many relationship since a trainer can teach as many solo_sessions as they desire, but each solo session will only have one trainer in it

Trainers Teaches Group_Session relation:

- THis is a One-to-many relationship since a trainer can teach as many group_sessions as they desire, but each group session will only have one trainer in it

Solo_Session utilizes Rooms relation:

- This is a Many-to-One relationship seeing as a solo session is only in one room however a room can have multiple solo sessions
- Partial participation because a room doesn't only have solo sessions it can also have group sessions

Group_Session utilizes Rooms relation:

- This is a Many-to-One relationship seeing as a group session is only in one room however a room can have multiple group sessions
- Partial participation because a room doesn't only have group sessions it can also have solo sessions

Rooms Booked with Room_bookings relation:

- This is a many to one relationship as a room can be booked multiple times however not every room is booked
- There is full participation from the room_bookings as every room in that booking is booked, however there is partial participation from the rooms because not every room is booked

General assumptions made within the fitness system:

- To make transactions simpler, we ask the members when registering, to input their card details so the admin staff can charge them (when it is appropriate) whenever.

- Since admin only has one user in the accounts table, nobody can add another admin account, we assumed admin oversees all, however they can add trainers, which isn't done in the login page
- Since usernames are authentic, that is the only piece of account information being checked when a user or anybody (trainer/admin) is logging in

2.3 DDL File
https://github.com/Abaseen123/COMP3005_fitnessClubFinalProject/tree/main/SQL

2.4 DML File
https://github.com/Abaseen123/COMP3005_fitnessClubFinalProject/tree/main/SQL

2.5 Implementation
https://github.com/Abaseen123/COMP3005_fitnessClubFinalProject/blob/main/fitness.py

See 2.1 also

We utilized psycopg to connect to the database. Since we were going with a terminal approach, python seemed like a good choice as the syntax with utilizing the pgadmin database is quite simple and straightforward.

Anytime In the application when anybody is logged in, a key feature that is present is the "0. Exit" feature, where if you type 0, its a while loop running, so it brings you back to the previous menu, this feature is much like that of a "back" button on an actual app, which is crucial for any application to run.

2.6 Bonus Features
NA

2.7 Github Repository

https://github.com/Abaseen123/COMP3005_fitnessClubFinalProject

VIDEO LINK:

https://www.youtube.com/watch?v=jUDAkYTzInY