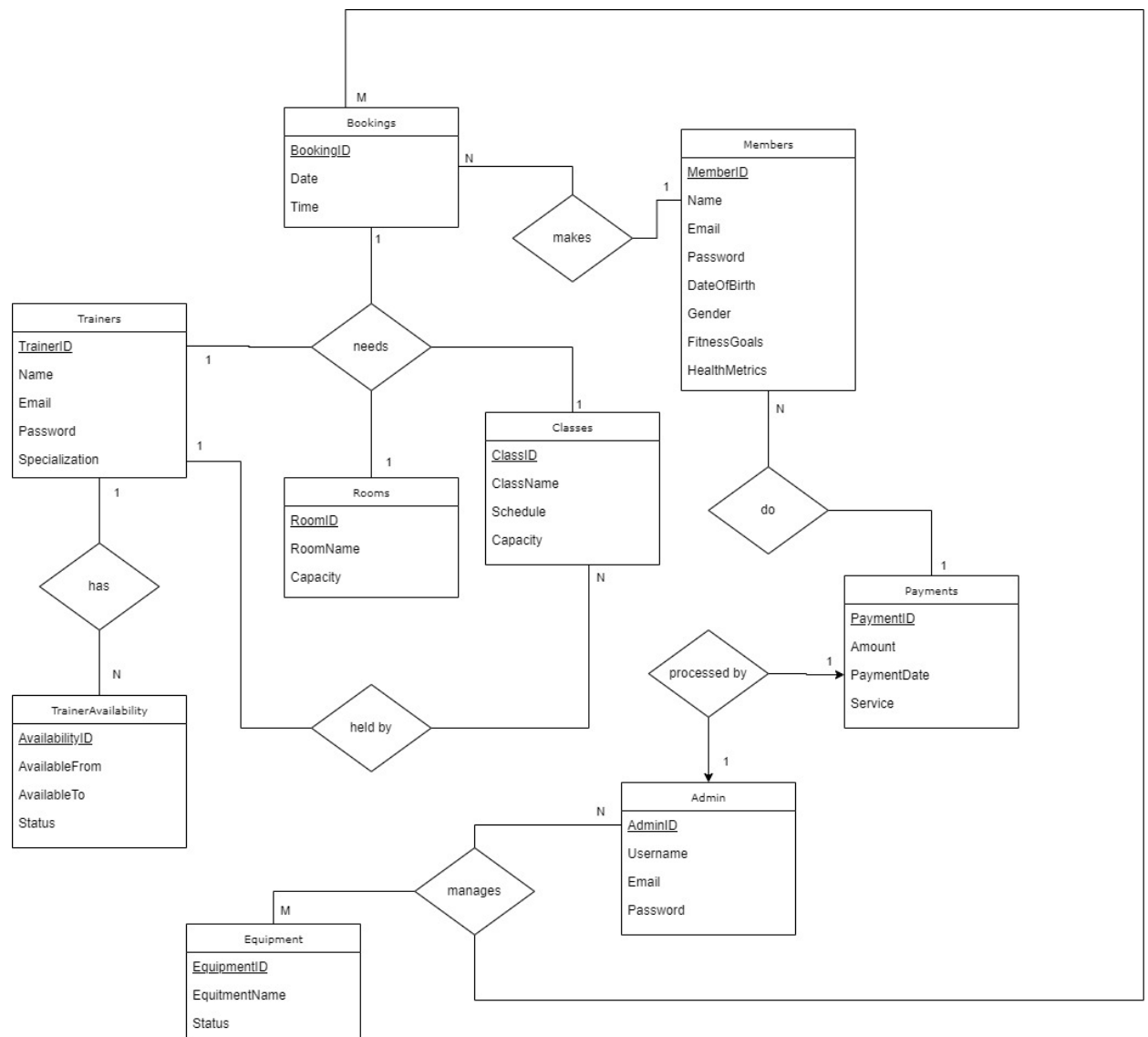COMP3005 Final Project Report

Group Member1: Pak Yin, Kan, 101260592 (Preferred name: Knox Kan)

Group Member2: Muhammed Burhan, 101270831

Conceptual Design with ER Diagram:

The above ER diagram shows the database design for this Gym management program. This program is mainly used by three parties, namely, members, trainers as well as admins, and therefore, those are three main entities in this diagram. For member, they have the ability to make bookings to various types of activities such as personal training and classes and have their responsibilities to make payment and as such this extends to another entities – "Payments", "Bookings",

"Classes" and "Room". Those entities should store all necessary data input from the main entities – "Admin", "Members" or "Trainer".
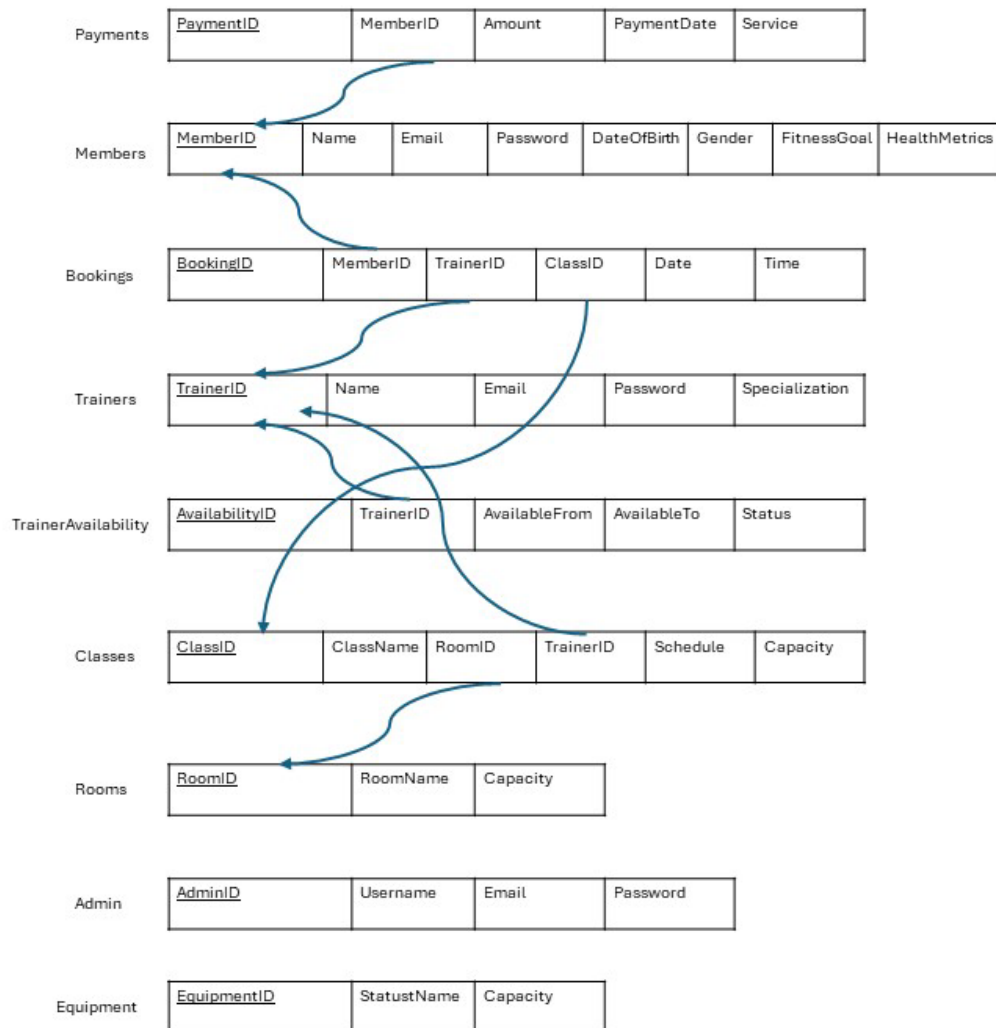
For the Trainer, since they should have the permission to enter their own availability for the personal training session, "TrainerAvailability" entity is created at this point for the trainer to input their availability as many as they can, with referencing to the TrainerID. In this approach, it prevents a ton of duplication record if we merged the availability to the Trainer table.

The "Admin" entity manages the current equipment of the Gym so it has access to the "Equipment" entity. The admin can also monitor and manage classes so they can also manipulate the "Classes" and "Rooms" tables.

Noted that, we acknowledge that when the program expands and especially when new requirements come, the schema needs to be adjusted, the above ER-diagram should cover the basic idea and requirements of this project.

Reduction to Relation Schemas:

With the ER diagram, we can come up with the following relation schemas:

Payments	PaymentID	MemberID	Amount	PaymentDate	Service

Members	MemberID	Name	Email	Password	DateOfBirth	Gender	FitnessGoal	HealthMetrics

Bookings	BookingID	MemberID	TrainerID	ClassID	Date	Time

Trainers	TrainerID	Name	Email	Password	Specialization

TrainerAvailability	AvailabilityID	TrainerID	AvailableFrom	AvailableTo	Status

Classes	ClassID	ClassName	RoomID	TrainerID	Schedule	Capacity

Rooms	RoomID	RoomName	Capacity

Admin	AdminID	Username	Email	Password

Equipment	EquipmentID	StatustName	Capacity

The schema is pretty simple and can be easily understood by breaking down the ER diagram. Members, Trainers, Admin table represents three types of users, and the other tables serves for other functionalities.

DDL File:

Please check the "SQL" directory. In theory, you do not need to run this file as when starts running the program, it will create the tables for the users automatically, but it was provided in case if there is any issue.

Link to DDL file: https://github.com/knoxkpy/COMP3005-Project/blob/main/SQL/DDL.sql

DML File:

Please check the "SQL" directory. In theory, you do not need this file as well since if you follow the instructions of the program, the database should work properly. But this file serves as a "facilitator" for the demo.

Link to DML file: https://github.com/knoxkpy/COMP3005-Project/blob/main/SQL/DML.sql

Implementation:

This program was written using python and utilized psycopg library to connect to PostgreSQL database. For detailed implementation of various functions (performing CURD), please look at the code.

There are few assumptions for the implementation:
- For the health matrix, we go for simplicity. The assumption was made that the user knows how to input a json form of data when registering their account. Wrongly input data may fail to pull the data and display them on the console.
- This goes the same for the date and time format. Instructions were given for the input.

Bonus Features:

Although we decided to use the command-line interface for our program, we added several other features.
- Registration (all members/admin staff/ trainers need to register for the system)
  - For registration, you can choose to register for a member, admin staff or as a trainer. But for trainers and admins, you need to have the invitation code in order to register. This prevents a random user registering the system as an admin or trainer.
- Authorization and authentication.
  - For different types of user (member, trainer, admin), they will have different privileges in this system. For example, the trainer can set their availability, but the member cannot. The member can join the classes but they cannot edit the details of the class and etc...
- Member checking their bills in the system.
  - This is not a required functionality in the specification, but now, once the member is logged in, they have the option to check their

pending bills.
- Trainer can see the members detail who joined their classes and personal training sessions.
    - For Member Profile Viewing functionality in the trainer.py, it is only for viewing the member(s) who has joined the trainer's class or personal training session.
    - The original required "Search by Member's name" is moved to a new functionality called "Search Member Profile by Name" in the same trainer.py.

Collaboration Part:

Pak Yin, Kan (Knox Kan) (101260592):
- System design (overall, database)
- Login functionality (authentication. Authorization)
- ER diagram, ER mapping
- All member functions
- Trainer functions: Schedule Management (Trainer can set the time for which they are available.)
- Project report (this file)
- Demonstration

Muhammed Burhan (101270831):
- System design (Database)
- Trainer functions: Member Profile Viewing (Search by Member's name)
- All administrative Staff functions.
- DDL file preparation
- DML file preparation
- Demonstration

GitHub Repository:

https://github.com/knoxkpy/COMP3005-Project

Youtube Video Demo Link:

https://www.youtube.com/watch?v=aNKNAwOhG-s