Elsa Faisal
101237082

Link to Youtube demo: https://youtu.be/91cdcrDCZKI

Link to Github repository: https://github.com/elsaurrrr/Health-and-Fitness-App/tree/main

Conceptual Design (25 marks):

Refer to ER-model.png

A few choices I made while creating the Entity Relationship Model:

Members should be able to register and manage their profiles, establish personal fitness goals (you can determine suitable fitness goals such as weight and time, and members will set the values), and input health metrics. They should have access to a personalized dashboard that tracks exercise routines, fitness achievements, and health statistics.

This is mainly covered by the Member table. Users' health metrics are attributes on that table; they would be able to update their information by updating those corresponding values. It is also covered by the Fitness Goals table, where members are able to establish fitness goals with a precise time frame and metrics. While I considered a separate, summative Health Metric Table, the reason I ultimately did not proceed was its relationship with the fitness goals table; in order to allow the user to have more concise fitness goals, I wanted for them to clearly see a single precise metric pertaining to their goal, rather than a full card of all their health data.

Trainers should have the ability to manage their schedules and view member profiles.

This is accomplished with the Trainer table, and the Schedule table. Trainers have their own schedule, with all of their Personal Training Sessions and Group Fitness Classes, so that they are able to effectively manage their schedule.

Administrative Staff should be equipped with features to manage room bookings, monitor fitness equipment maintenance, update class schedules, oversee billing, and process payments for membership fees, personal training sessions, and other services.

This is covered by the Admin Staff table, as well as the Schedule table, Timeslot table, Bill table, Payments tables, etc. There isn't any cardinality between the Admin and other Entities; rather Admin users have the ability to manage the other Users and services of the app.

A few other choices I made:
- Separate bill and payment entities; in general, this allows for better organization, clarity, and flexibility in managing financial transactions within the Health and Fitness App. This adheres to the principles of normalization, which helps reduce redundancy and improve data integrity– each table contains unique information related to either billing or payment processes. It also make it more scalable. More practically, this accommodates real-life situations where bills are not paid immediately when issued (i.e. insurance, monthly billing setups, etc.) .
- Date and Time information abstracted from the class and PT sessions to the Timeslot Entity to prevent duplicate attributes; centralizing date and time information in the Timeslot entity, ensures consistency across all activities, such as personal training sessions and group fitness classes. This helps avoid discrepancies or conflicts that may arise if each activity manages its own date and time information. Having a separate Timeslot entity also provides flexibility in managing scheduling. It allows for easy adjustment of time slots, such as extending or shortening the duration of a slot, without needing to modify individual sessions or classes; also ensures no overlapping time slots for

different activities, reducing the risk of scheduling conflicts
- Schedule table: serves as a hub where associations between members, trainers, rooms, and specific activities are stored and managed, system gains flexibility in managing different types of activities and their scheduling requirements. New activities can be easily added to the system without significant changes to the database schema, as they can leverage the existing structure of the Schedule table

Overall, I really thought about the extensibility of the app as I was creating the Entity Relationship model

Reduction to Relation Schemas (25 marks):

Refer to Relation-Schema.png

DDL File (10 marks):

Refer to sql/ddl.sql

DML File (10 marks):

Refer to sql/dml.sql

Implementation (30 marks):

N/A (No implementation)

Bonus (30 marks):

- **Bonus Features**:

N/A (No bonus features)

- **Group Bonus**:

1 person (Elsa Faisal)