Milestone#3

Project Report

CS2005 :Database Systems

FLEX TRAINER

Abdullah Zubair	22i1077	CS-F
Saffi Muhammad Hashir	22i1293	CS-G
Tauha Imran	22i1239	CS-G

CONTENTS

OVERVIEW

This semester project was aimed to build a Flex Trainer management system as a desktop application for gyms and health membership systems. Flex Trainer is designed to facilitate the gym owners, trainers, and members to automate the operations of keeping records and store them in the form of a large and user-friendly database, The objective of the Flex Trainer is to provide a system that handles the information of the members using the application and maintaining their health care.

SPECIFICATIONS

Every user involved in the system will have their own User Interface. We have created four interfaces:

- Member interface.
- Trainer Interface
- Gym Owner
- Admin Interface

MILESTONES

This Project was spread throughout three milestones

milestone#1

Designing the ERD , EERD and RDM . These served as our basis for developing the whole application.

milestone#2

The Front end , linked and tested interfaces of the whole project

milestone#3

The database creation , linking and development of the entire backend leading to a complete project.

(Enhanced) ENTITY RELATIONSHIP DIAGRAM - EERD

The (enhanced) Entity relationship Diagram developed for the project contains 15 entities and 4 weak entities . There is a super class of **‘USER’** with disjoint sub-classes of **‘Member’** , **‘Trainer’** , **‘GymOwner’** and **‘Admin’** . Along with there is an entity **‘Gym’** for Gyms. Along with this we have **‘Diet Plan’** and **‘Workout Plan’** in this diagram , making up the crux of the EERD.

We used these entities to Map out all the relationships found in the case study.

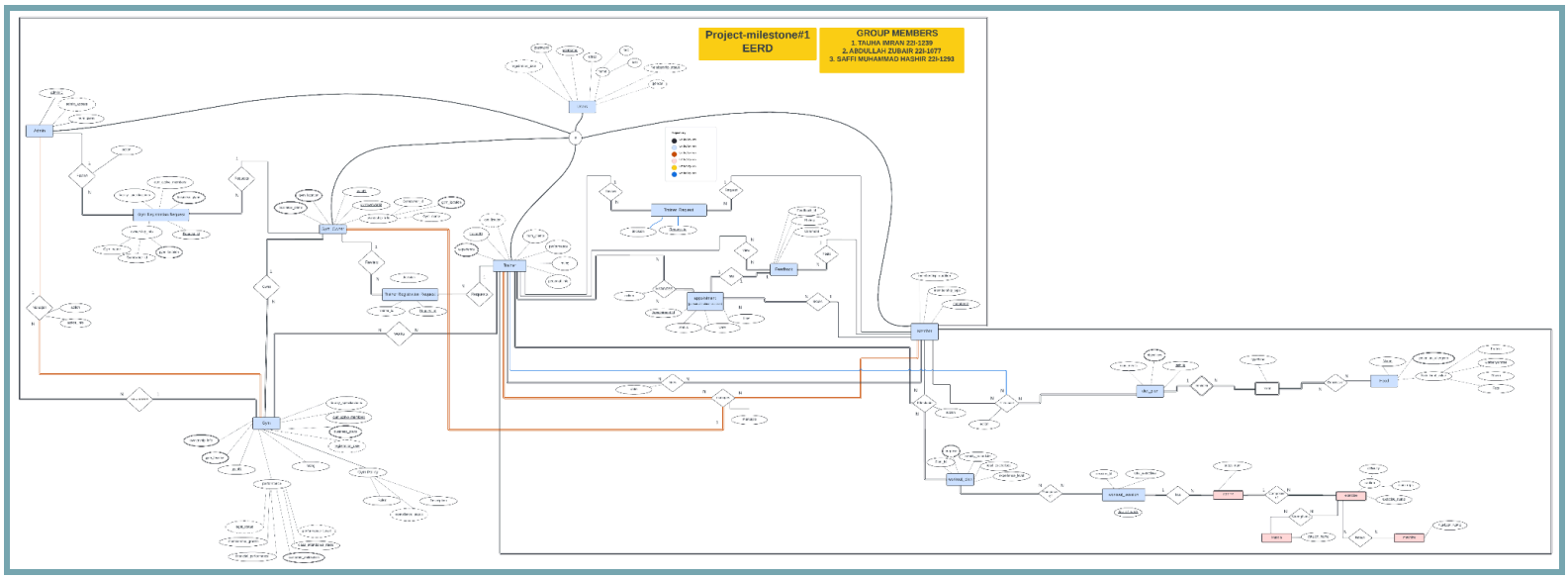
The list of entities is as follows

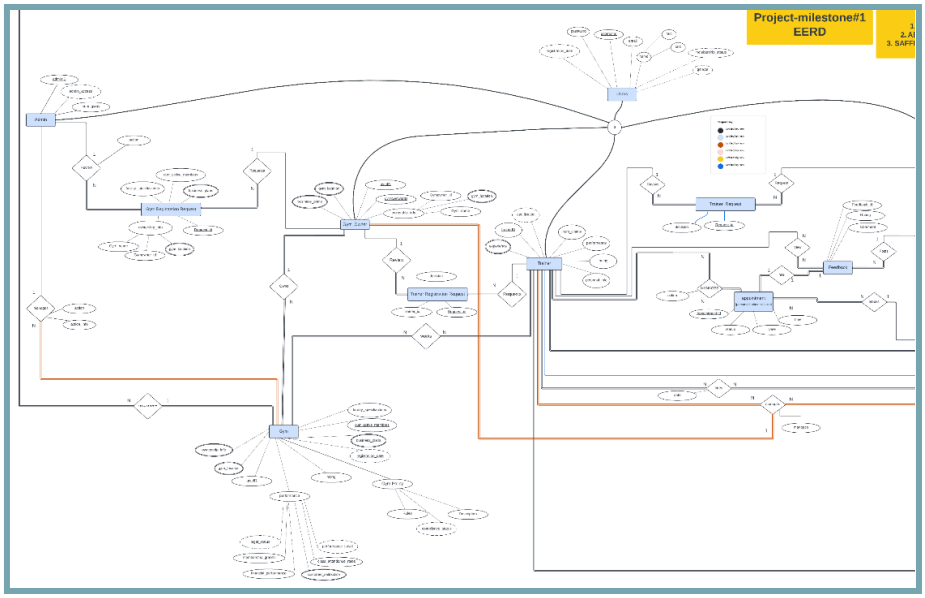
Entities

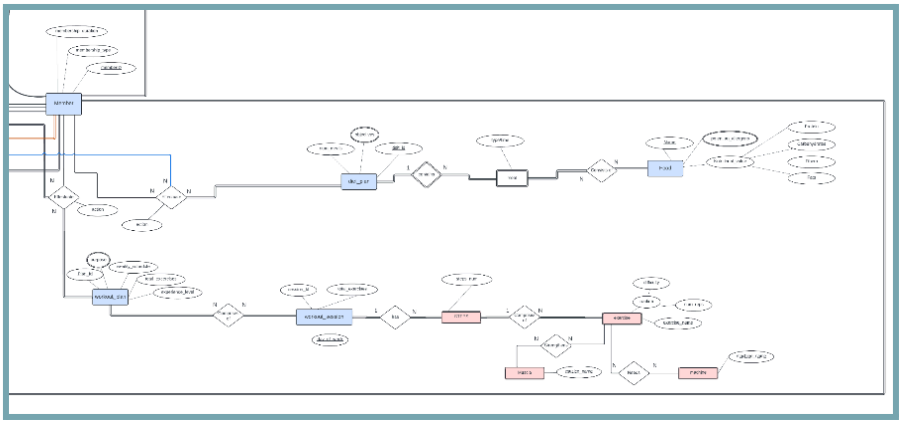
- Users - *superclass*
- Member - *disjoint subclass*
- Trainer - *disjoint subclass*
- Admin - *disjoint subclass*
- Gym_Owner - *disjoint subclass*
- Gym Registration Request
- Trainer Registration Request
- Gym
- Trainer Request
- Appointment
- Feedback
- Workout_plan
- Workout_session
- Diet_plan
- Food

Weak Entities

- Steps
- Exercise
- Muscle
- Machine
- Meal *(dropped in final design)*


Project-milestone#1
EERD
GROUP MEMBERS
1. TRIJNA IMRAN 221-1239
2. ABDULLAH ZUBAIR 221-1077
3. SAFFI MUHAMMAD MASHIR 221-1293

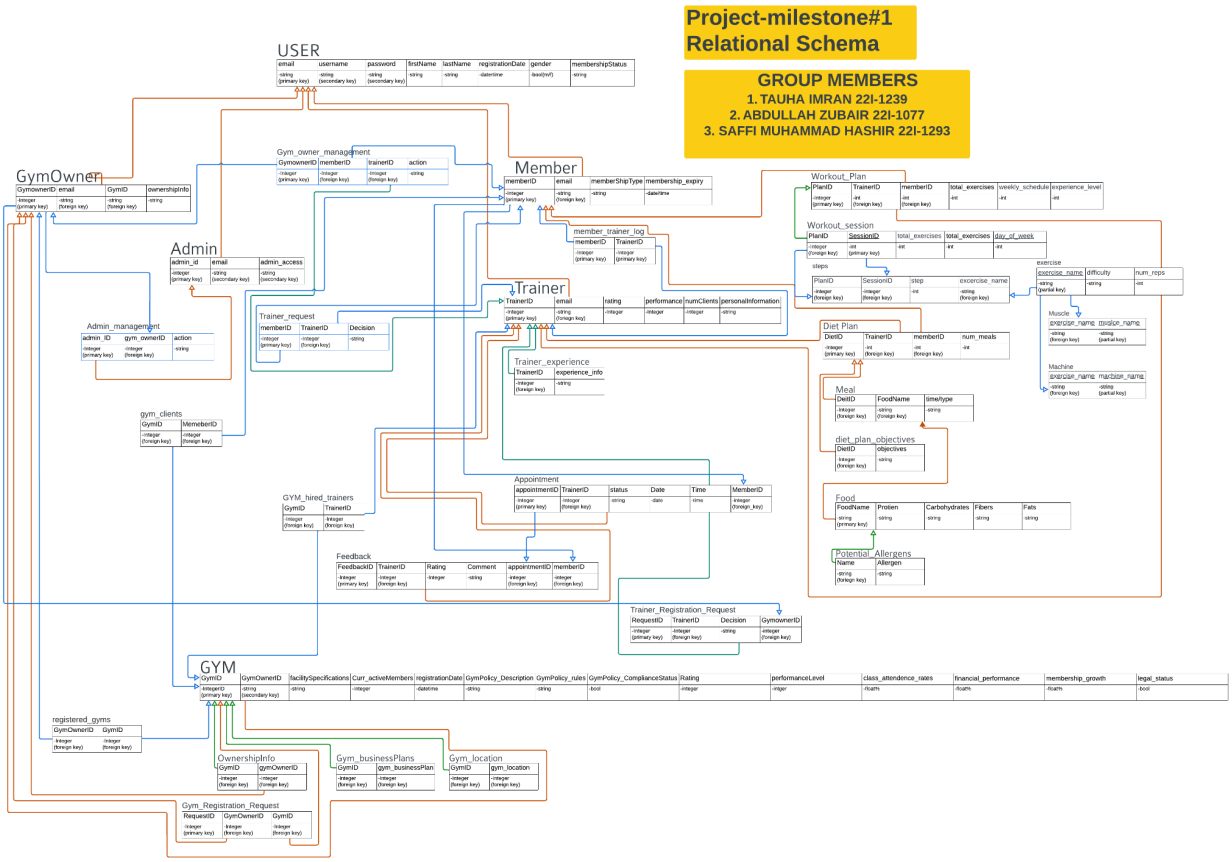

Project-milestone#1
EERD



RELATIONAL DATA MODEL - RDM

The Relational Data Model was made by citing the EERD and using the concepts of normalization

There are 29-30 Tables made in the schema



The first step involves analyzing the schema to identify entities, attributes, and their interconnections. Next, the database designer selects an appropriate relational model, determining tables, fields, and data types. Normalization techniques are then applied to minimize redundancy and ensure data integrity. Indexes and constraints are implemented to enhance performance and enforce business rules. Finally, the RDM system is implemented, tested, and refined to meet the specific requirements of the report, ensuring efficient data storage, retrieval, and manipulation.

FRONT END RECAP & UPDATES

The Front End is majorly the same . The Names of all the Pages are as follows. There are changes to notify about.

- Sign up Page
- Log in Page
- Member Dashboard
    - Main Page
    - Profile Page
    - Flex Workouts
    - Flex Diet
    - Trainer Sessions
    - Gym Owner Dashboard
    - My profile
    - My Trainers
    - My Members
    - My Reports
- Trainer Dashboard
    - My profile
- Trainees
    - Workout
    - Diet Plans
    - Appointment
    - Feedback
    - Reports Manager
- Admin Dashboard
    - Active Users
    - Registered Gyms
    - Gym Registration Requests
    - Reports Manager
- Group Members

DATABASE CREATION IN SQL

We made the Database as ***'FlexTrainer.bak'*** There any many tables , with a no major changes from the designed RDM . The only thing to be noted is that **'meal'** was dropped and **workout_Member** and **diet_Member** were added to help with operations

1. **Admin:** Stores information about gym management system administrators.
2. **Admin_management:** Manages admin accounts within the system.
3. **Appointment:** Tracks appointments made by members with trainers.
4. **audit_trail:** Logs activity within the system, such as user logins and data changes.
5. **diet_Member:** Stores information about members following a diet plan.
6. **Diet_plan:** Details the different diet plans offered by the gym.
7. **diet_plan_objectives:** Defines the goals of the different diet plans.
8. **exercise:** Lists the different exercises available at the gym.
9. **Feedback:** Captures feedback from members about the gym.
10. **Food:** Tracks information about foods members can include in their diet plans.
11. **Gym:** Stores information about the gyms managed by the system.
12. **Gym_business Plans:** Details the different business plans offered by the gym.
13. **gym_clients:** Manages information about members signed up at the gym.
14. **Gym_hired_trainers:** Tracks information about trainers employed by the gym.
15. **Gym_location:** Stores information about the location of the gyms.
16. **Gym_owner_management:** Manages gym owners within the system.
17. **Gym_registration_request:** Tracks requests for gym membership.
18. **GymOwner:** Stores information about the owners of the gyms.
19. **Machine:** Lists the exercise machines available at the gym.
20. **Member:** Manages information about the gym members.
21. **member_trainer_log:** Tracks interactions between members and trainers.
22. **Muscle:** Lists the different muscle groups targeted by exercises.
23. **OwnershipInfo:** Stores information about gym ownership.
24. **Potential_Allergens:** Tracks potential allergens in foods.
25. **registered_gyms:** Manages information about gyms registered with the system.
26. **steps:** Tracks information about steps taken by members, possibly in relation to a fitness tracker.
27. **Trainer:** Manages information about registered trainers.
28. **Trainer_registration_request:** Tracks requests for trainer registration.
29. **Trainer_request:** Handles requests made by trainers, possibly for scheduling or resources.

30. **TrainerExperience:** Stores information about the experience of the trainers.
31. **USERS:** Manages information about the general users of the system.
32. **workout_Member:** Stores information about members following a workout plan.
33. **Workout_Plan:** Details the different workout plans offered by the gym.
34. **Workout_session:** Tracks information about workout sessions completed by members

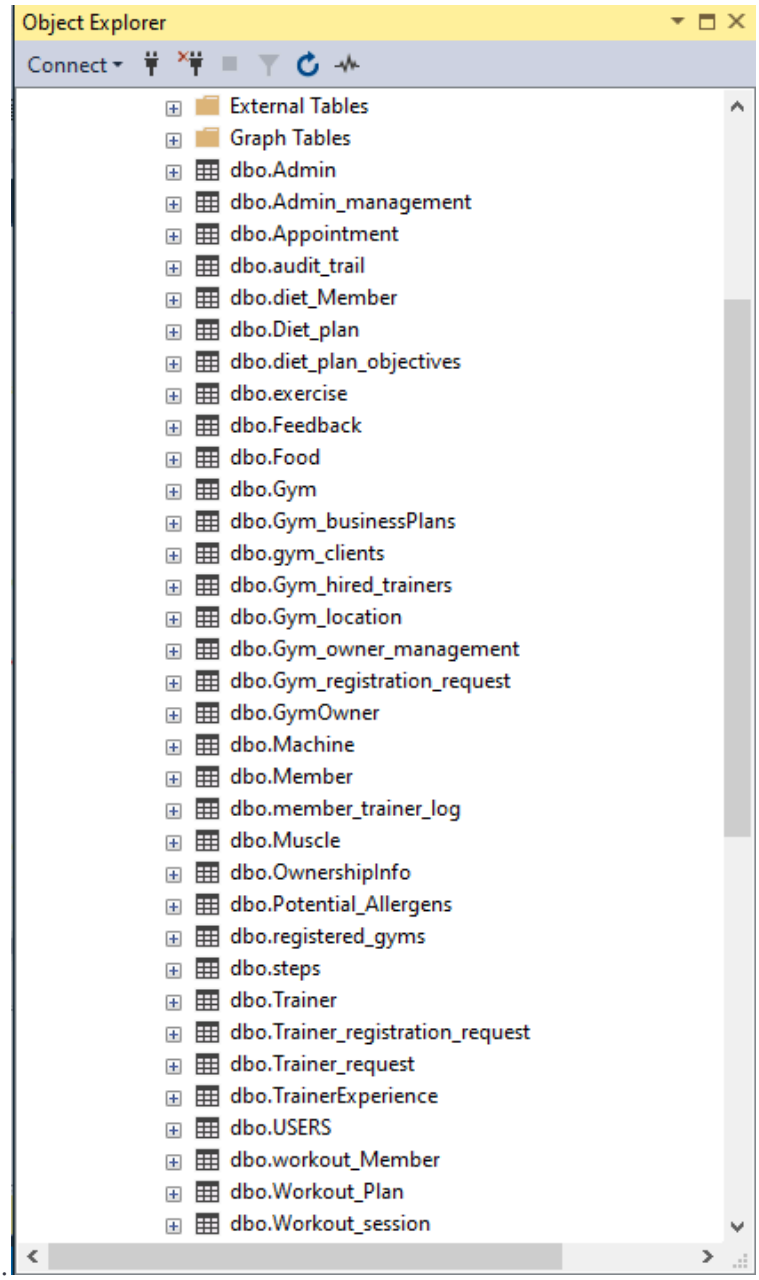


Fig - screenshot of all the tables in the database

DATA HANDLING

We Used triggers for all required procedures to handle the data and used multiple Excel files to track and import dummy data.

1. **Admin_ added**: Track actions related to gym administrators, such as adding, removing, updating, or logging management activities.
2. **Appointment_logged:** Indicates an appointment between a member and trainer was created.
3. **Diet_plan_** entries: Track changes to diet plans, including adding, removing, updating objectives, or modifying the plan itself.
4. **exercise_added:** A new exercise was added to the system.
5. **Feedback_logged:** A member submitted feedback about the gym.
6. **Food_added:** A new food item was added to the system.
7. **Gym_** entries: Track changes to gyms, including adding, removing, updating business plans, adding/removing members or trainers, managing locations, ownership, and registration requests.
8. **GymOwner_** entries: Track actions related to gym owners, such as adding, removing, or updating their information.
9. **Machine_added:** A new exercise machine was added to the system.
10. **Member_** entries: Track changes to member information, including adding, removing, updating details, or logging trainer interactions.
11. **Muscle_added:** A new muscle group was added to the system.
12. **OwnershipInfo_** entries: Track changes to gym ownership information, such as adding, removing, or updating details.
13. **Potential_Allergens_added:** A new potential allergen was added to the food tracking system.
14. **registered_gyms_** entries: Track changes to registered gyms, such as adding, removing, or updating information.
15. **Trainer_** entries: Track changes to trainer information, including adding, removing, updating details, handling registration requests, or logging their requests.
16. **TrainerExperience_** entries: Track changes to trainer experience information, such as adding, removing, or updating details.
17. **user_** entries: Track changes to general user information, including adding, removing, or updating user accounts.
18. **Workout_Plan_** entries: Track changes to workout plans, including adding, removing, or updating plans.

19. **Workout_session_** entries: Track changes to workout sessions, including adding, removing, or updating information about member workouts.

REPORTS & QUERIES

For our Reports and Queries we have used stored procedures. Which have been used in the C# backend. The stored procedures made in MySQL for SQL Server / SSMS are as follows

- **R1_members_of_spec_gymXtrainer**
    - Details of members of one specific gym that get training from 1 specific trainer.
- **R2_members_of_spec_gymXdeitplan (Duplicate)**
    - Details of members from one specific gym that follow a specific diet plan.
- **R3_specific_diet_plan**
    - Details of members across all gyms of a specific trainer that follow a specific diet plan.
- **R4_count_of_members**
    - Count of members who will be using specific machines on a given day in a specific gym.
- **R5_dietplans_less_than_calories (Duplicate)**
    - List of Diet plans that have less than 500 calorie meals as breakfast.
- **R6_dietplans_less_than_total_carbs**
    - List of diet plans in which total carbohydrate intake is less than 300 grams.
- **R7_workoutplan_no_machine (Duplicate)**
    - List of workout plans that don't require using a specific machine.
- **R7_workoutplan_no_machineX (Duplicate)**
    - List of workout plans that don't require using a specific machine.
- **R8_dietplan_no_allergens**
    - List of diet plans which doesn't have peanuts as allergens.
- **R9_membership_data_last_X_months**
    - New membership data in last 3 months (Gym Owner).
- **R10_total_members_gyms_past_X_months**
    - Comparison of total members in multiple gyms, in the past 6 months.
- **R11_trainers_in_spec_gym**
    - Data about trainers in specific Gym .
- **R12_gyms_of_spec_trainer**
    - Data about Gyms a specific trainer.
- **R13_gyms_with_spec_num_clients**
    - Gyms with a specific number of clients.

- **R14_gyms_with_spec_owner**
    - Data of Gyms Owned By a specific owner.
- **R15_trainers_over_rating**
    - Trainers with a specific rating.
- **R16_gyms_acc_num_members**
    - Gyms with specific number of clients.
- **R17_pop_gyms_of_spec_location**
    - popular gyms (of a specific place).
- **R18_pop_workoutplans_of_spec_trainer**
    - popular workouts (of a specific trainer).
- **R19_pop_dietplans_of_spec_trainer**
    - popular diet_plans (of a specific trainer).
- **R20_Admin_actions_on_Gym_owner**
    - all data of admin actions done on a specific Gym owner.
- **R21_GymOwner_actions_on_specific_Trainer**
    - all data of GymOwner actions done on a specific Trainer.
- **R22_GymOwner_actions_on_specific_Member**
    - all data of GymOwner actions done on a specific Member.

WORK DISTRIBUTION

Abdullah Zubair (22i1077)

- Front end in C# for Member and Gym Owner
- Backend in C# for Member and Gym Owner
- Reports in C# backend and front end
- Project merging and linking
- Testing and project-reports

Saffi Muhammad Hashir (22i1293)

- Designing EERD and RDM
- Front end in C# for Trainer and Admin
- Backend in C# for Trainer and Admin
- Project merging and linking
- Testing and project-reports

Tauha Imran (22i1239)

- Designing EERD and RDM
- Database creation in SQL
- Excel data creation and importing into database
- SQL queries for Triggers , Reports and Alterations
- Testing and project-reports

CONCLUSION

In conclusion, this project successfully designed and implemented a database system for managing a gym.

This project stands as a testament to the dedication and collaboration of our 3-person team. While designing and implementing this comprehensive gym management system, we faced challenges in ensuring data integrity across various functionalities. Through careful planning, rigorous testing, and open communication, we successfully overcame these hurdles, resulting in a robust database system.

The system efficiently stores and retrieves information about members, trainers, gym locations, workout plans, diet plans, and more. Stored procedures facilitate various functionalities, including finding members by gym or trainer, managing diet plans based on calorie or allergen restrictions, and tracking actions taken by administrators and gym owners. This system provides a comprehensive solution to streamline gym operations and enhance member experience.