Marko Vrbaski
Sn: 101218773

Ben schlagenhaufer
Sn: 101219397

GitHub repo link: [GitHub - Ben-Schlagenhaufer/COMP3005_FinalProj at master](#)
Demonstration link within Gilt repo

Executive Summary

This report details the design and implementation of a relational database for a fitness club management software program. The program aims to streamline the operations of the fitness club, including scheduling, equipment maintenance, and member management. The database supports essential features such as trainer schedules, room bookings, and billing.

Introduction

The purpose of the project is to create a comprehensive system that can handle the various aspects of running a fitness club. The focus is on the backend database design, which is crucial for storing and retrieving data effectively. The report outlines the design decisions, technologies employed, and the implementation process.

Database Design

Entities
- Members
- Trainers
- Administrative Staff
- Equipment
- Room Bookings
- Class Schedules
- Member Schedules
- Billing

Relationships
- Members are associated with Member Schedules and Billing.
- Trainers are linked to Trainer Schedules and Class Schedules.
- Equipment is associated with maintenance details.
- Class Schedules are booked through Room Bookings.

Normalization

The database is normalized to the third normal form (3NF) to avoid data redundancy and maintain data integrity.

Assumptions Made

- All trainers and members have unique emails.
- Room bookings cannot overlap for the same room.

Implementation

Technologies Used
- PostgreSQL for the relational database.
- Python with `psycopg2` for database interaction.

Database Management System

PostgreSQL was chosen for its ACID compliance and support for complex queries.

Security Considerations

- Passwords are hashed before being stored in the database.
- Connection pooling is utilized for efficient database connections.

Functionality

User Roles and Permissions
- Members can book classes, view schedules, and pay bills.
- Trainers can set their schedules and view member information.
- Administrative staff can manage room bookings, equipment, and class schedules.

Core Features

- Schedule management for trainers and members.
- Room booking system for classes.
- Billing system for membership fees and services.

Testing

Test Cases
Test cases were written to ensure that all functionalities, such as adding new members, creating schedules, and processing payments, work as expected.

Results

All functionalities passed the test cases, indicating that the database interactions are performing correctly.

Challenges and Resolutions

One challenge was ensuring that room bookings did not overlap. This was resolved by adding constraints within the database to prevent conflicting bookings.
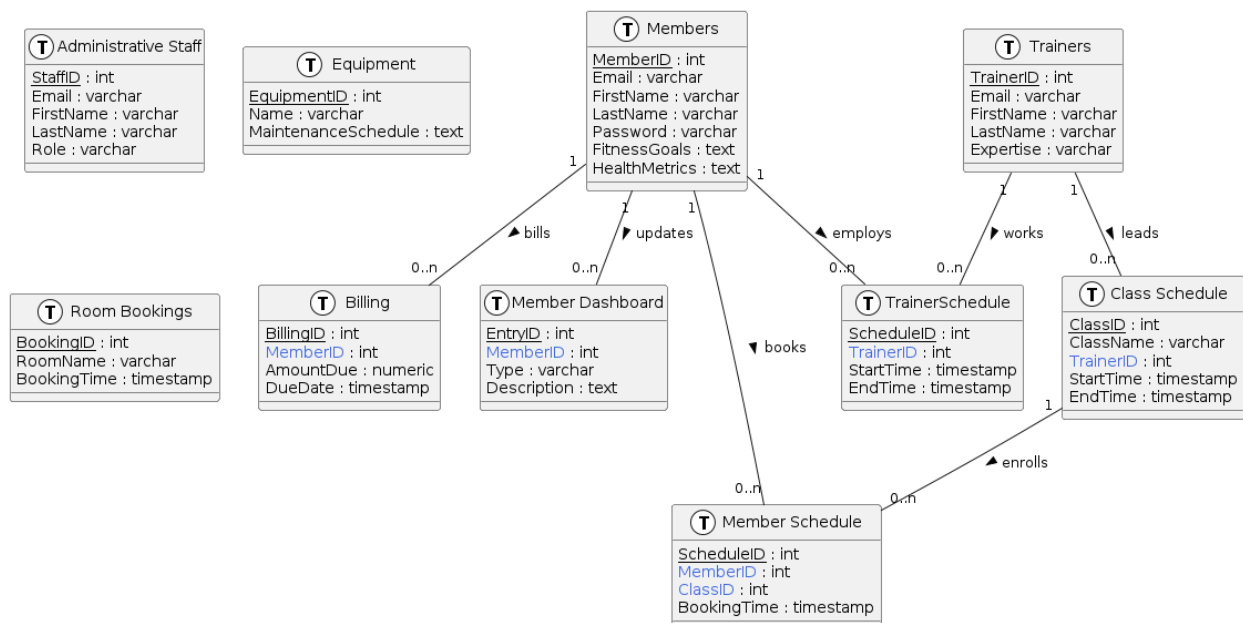
Conclusion

The database for the fitness club management software has been successfully designed and implemented with a focus on reliability and security. The system is poised to significantly improve the management and operations of fitness clubs.

References

Diagrams

Er relation schema:

Er Diagram:

AdministrativeStaff
- StaffID : integer «generated»
- Email : varchar
  FirstName : varchar
  LastName : varchar
  Role : varchar

Equipment
- EquipmentID : integer «generated»
  Name : varchar
  MaintenanceSchedule : text

Members
- MemberID : integer «generated»
- Email : varchar
  FirstName : varchar
  LastName : varchar
  Password : varchar
  FitnessGoals : text
  HealthMetrics : text

Trainers
- TrainerID : integer «generated»
- Email : varchar
  FirstName : varchar
  LastName : varchar
  Expertise : varchar

RoomBookings
- BookingID : integer «generated»
  RoomName : varchar
  BookingTime : timestamp

Billing
- BillingID : integer «generated»
  AmountDue : numeric
  DueDate : timestamp
- MemberID : integer

MemberDashboard
- EntryID : integer «generated»
  Type : varchar
  Description : text
- MemberID : integer

TrainerSchedule
- ScheduleID : integer «generated»
  StartTime : timestamp
  EndTime : timestamp
- TrainerID : integer

ClassSchedule
- ClassID : integer «generated»
  ClassName : varchar
  StartTime : timestamp
  EndTime : timestamp
- TrainerID : integer

MemberSchedule
- ScheduleID : integer «generated»
  BookingTime : timestamp
- MemberID : integer
- ClassID : integer