RAJALAKSHMI ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

CS23332 DATABASE MANAGEMENT SYSTEM

GYM MANAGEMENT SYSTEM

MINI PROJECT REPORT

Submitted by

JAGDISH KHANNA KP	231501064
PRATHISHA R	231501119
PREESHAA BV	231501120

In partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

RAJALAKSHMI ENGINEERING COLLEGE
(AUTONOMOUS)

THANDALAM

CHENNAI-602105

2024 – 25

BONAFIDE CERTIFICATE

Certified that this project report "**GYM MANAGEMENT SYSTEM" is** the Bonafide work of "JAGDISH KHANNA KP (231501119), PRATHISHA R (231501119), PREESHAA BV (231501120)".

who carried out the project work under my supervision.

Submitted for the Practical Examination held on ____________________

SIGNATURE

Mr. U.Kumaran

Assistant Professor(SS)

AIML

Rajalakshmi Engineering college

(Autonomous)

Thandalam, Chennai - 602 105

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

A gym management system is an integrated software solution designed to streamline and automate the operations of fitness centres, gyms, and wellness facilities. It facilitates efficient member registration, payment tracking, attendance monitoring, and scheduling of fitness classes or personal training sessions. The system enhances user experience by offering features like online membership management, automated reminders, and real-time updates on gym events or promotions. For gym administrators, it provides tools for staff management, inventory tracking, and financial reporting, enabling better decision-making and resource allocation. With secure data storage and user-friendly interfaces, the gym management system ensures seamless communication between staff and members while fostering a healthier and more organized fitness environment.

TABLE OF CONTENTS

INTRODUCTION

1.1 INTRODUCTION

This report outlines the development of a comprehensive Gym Management System aimed at streamlining and automating the operations of fitness centres and gyms. With the growing demand for efficient fitness facility management and enhanced member experiences, this system provides a structured and systematic solution to meet the needs of gym administrators and members. Leveraging modern software development principles and a user-centric design approach, the system ensures seamless management of daily operations while enhancing overall productivity.

The Gym Management System addresses common challenges faced by fitness facilities, such as managing member registrations, scheduling classes, tracking attendance, and handling payments. Through a web-based interface, the system provides users with convenient access to key functionalities, including membership management, personalized workout plans, and real-time communication with gym staff. Designed to cater to a broad range of users, from fitness enthusiasts to professional trainers, the system emphasizes ease of use, accessibility, and operational efficiency.

1.2 Objectives

The primary objectives of the Gym Management System are:

- Streamlined Membership Management: Develop a platform that allows easy registration, renewal, and tracking of memberships, ensuring a hassle-free experience for gym members and staff.
- Class and Schedule Management: Provide tools to efficiently organize and manage fitness classes, personal training sessions, and resource allocation, ensuring optimal scheduling.
- Attendance and Performance Tracking: Enable members and trainers to monitor attendance and track fitness progress, fostering motivation and accountability.
- Integrated Payment System: Implement a secure and efficient payment gateway to handle membership fees, class bookings, and other transactions seamlessly.
- User-Friendly Interface: Ensure the system is intuitive and easy to navigate for users with varying levels of technical expertise.

Additionally, the system aims to enhance member engagement by incorporating features such as automated reminders, personalized recommendations, and real-time notifications for gym events and offers.

1.4 MODULE

The Gym Management System consists of several key modules designed to efficiently handle various tasks in gym administration. These modules include:

1. Member Management: Handling member registrations, updates, and viewing of member details.
2. Trainer Management: Managing trainer profiles, schedules, and assignments.
3. Payment Management: Keeping track of payments and membership fees.
4. Login System: Admin login functionality for secure access to the system.

Each module is integrated to provide a seamless experience for both gym staff and management, enhancing operational efficiency.

SURVEY OF TECHNOLOGY

2.1 SOFTWARE DESCRIPTION

The Gym Management System is a Java-based software application designed to manage and automate the administrative tasks of a gym. It integrates various modules such as member management, trainer schedules, payment processing, and system login. The software uses Java Swing for the graphical user interface (GUI) and MySQL for database management. This system provides an easy-to-use platform for gym owners and staff to efficiently manage their operations, improve organization, and track important data related to gym memberships, trainers, and payments.

2.2 LANGUAGES

2.2.1 MY SQL

In the Gym Management System, MySQL is used to manage and store the data related to gym operations. It stores essential information such as member details, trainer profiles, payment records, and other gym-related data. The system communicates with the MySQL database through JDBC (Java Database Connectivity), which enables Java to send queries to the database and retrieve the required information. SQL queries are used for performing CRUD (Create, Read, Update, Delete) operations on the data, ensuring proper storage and retrieval within the system**.**

2.2.2 JAVA

In the Gym Management System, **Java** is used for the core functionality, handling system logic and user interactions. **Java Swing** is employed to create the graphical user interface (GUI), allowing users to interact with the system through visually appealing forms. The language manages actions such as registering members, updating trainer schedules, processing payments, and handling system operations. Additionally, **JDBC** (Java Database Connectivity) enables the application to connect to a **MySQL** database for data storage and retrieval, supporting CRUD operations across the system.

REQUIREMENT AND ANALYSIS

3.1 REQUIREMENTS SPECIFICATION

The Gym Management System requires the following:

1. **Hardware:**
   - A computer with at least 4GB RAM.
   - A MySQL-compatible database server.
2. **Software:**
   - Java Development Kit (JDK), version 8 or higher.
   - NetBeans or any Java IDE for development.
   - MySQL Database for storing data.
   - Java Swing for building the graphical user interface.
3. **System Requirements:**
   - Operating system: Windows, macOS, or Linux.
   - JDBC Connector for Java-MySQL integration.

These specifications ensure the system functions smoothly, handling data management and user interface tasks.

3.2 HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Requirements:

- A computer with at least 4GB of RAM.
- Sufficient storage for the MySQL database and application files.
- A stable internet connection for database access and updates.

Software Requirements:

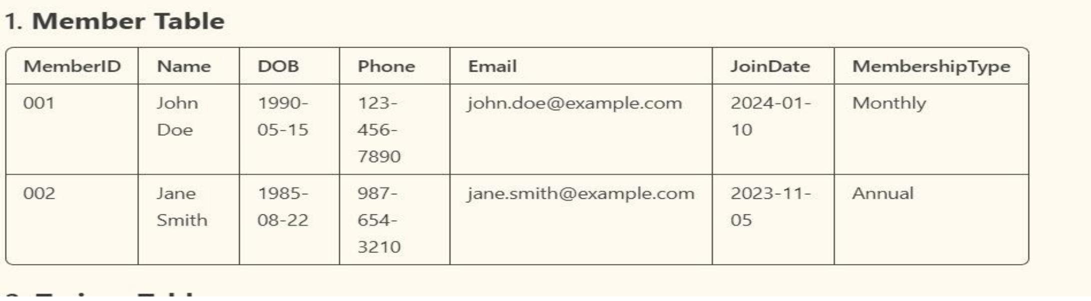
- Java Development Kit (JDK) 8 or higher**.**
- NetBeans IDE or any Java-compatible IDE.
- MySQL for database management.
- JDBC Connector to enable Java-MySQL communication.
- Java Swing for creating the user interface.

3.3 DATA FLOW DIAGRAM

A **Data Flow Diagram (DFD)** is used to represent how data moves within the Gym Management System. It shows how input data (e.g., member information, payment details) flows through various processes (e.g., registering members, processing payments) and how output is generated (e.g., member details, payment records).

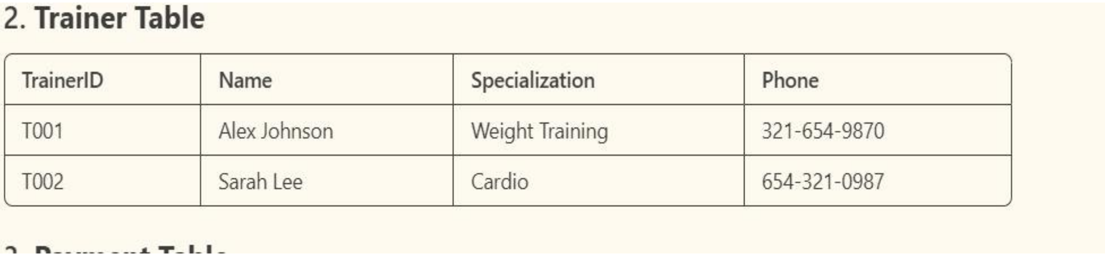
- **Level 0 DFD**: Depicts the overall system as a single process interacting with external entities such as users and the database.
- **Level 1 DFD**: Breaks down the main process into sub-processes, showing how data flows between different modules like member management, trainer management, and payment processing.
- A data-flow diagram is a way of representing a flow of data through a process or a system.
- The DFD also provides information about the outputs and inputs of each entity and the process itself.

3.4 DATA DICTIONARY



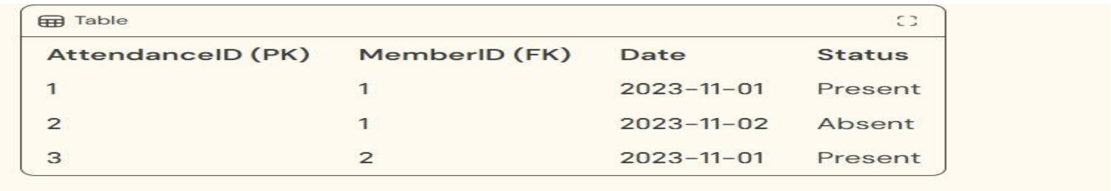
1. **Member Table**

MemberID	Name	DOB	Phone	Email	JoinDate	MembershipType
001	John Doe	1990-05-15	123-456-7890	john.doe@example.com	2024-01-10	Monthly
002	Jane Smith	1985-08-22	987-654-3210	jane.smith@example.com	2023-11-05	Annual



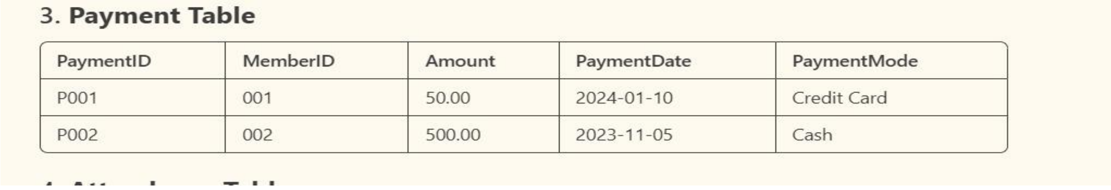
2. **Trainer Table**

TrainerID	Name	Specialization	Phone
T001	Alex Johnson	Weight Training	321-654-9870
T002	Sarah Lee	Cardio	654-321-0987



Table

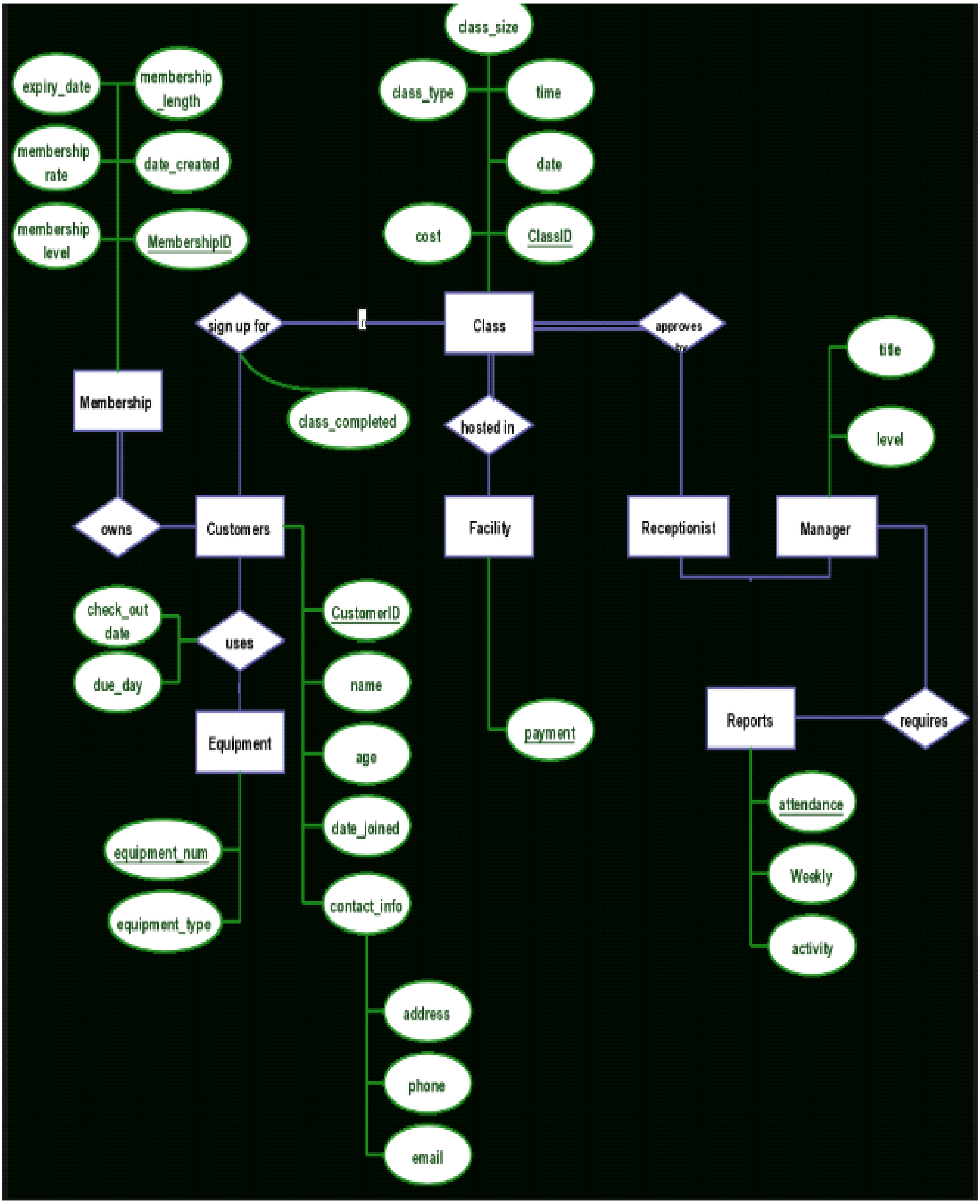
AttendanceID (PK)	MemberID (FK)	Date	Status
1	1	2023-11-01	Present
2	1	2023-11-02	Absent
3	2	2023-11-01	Present

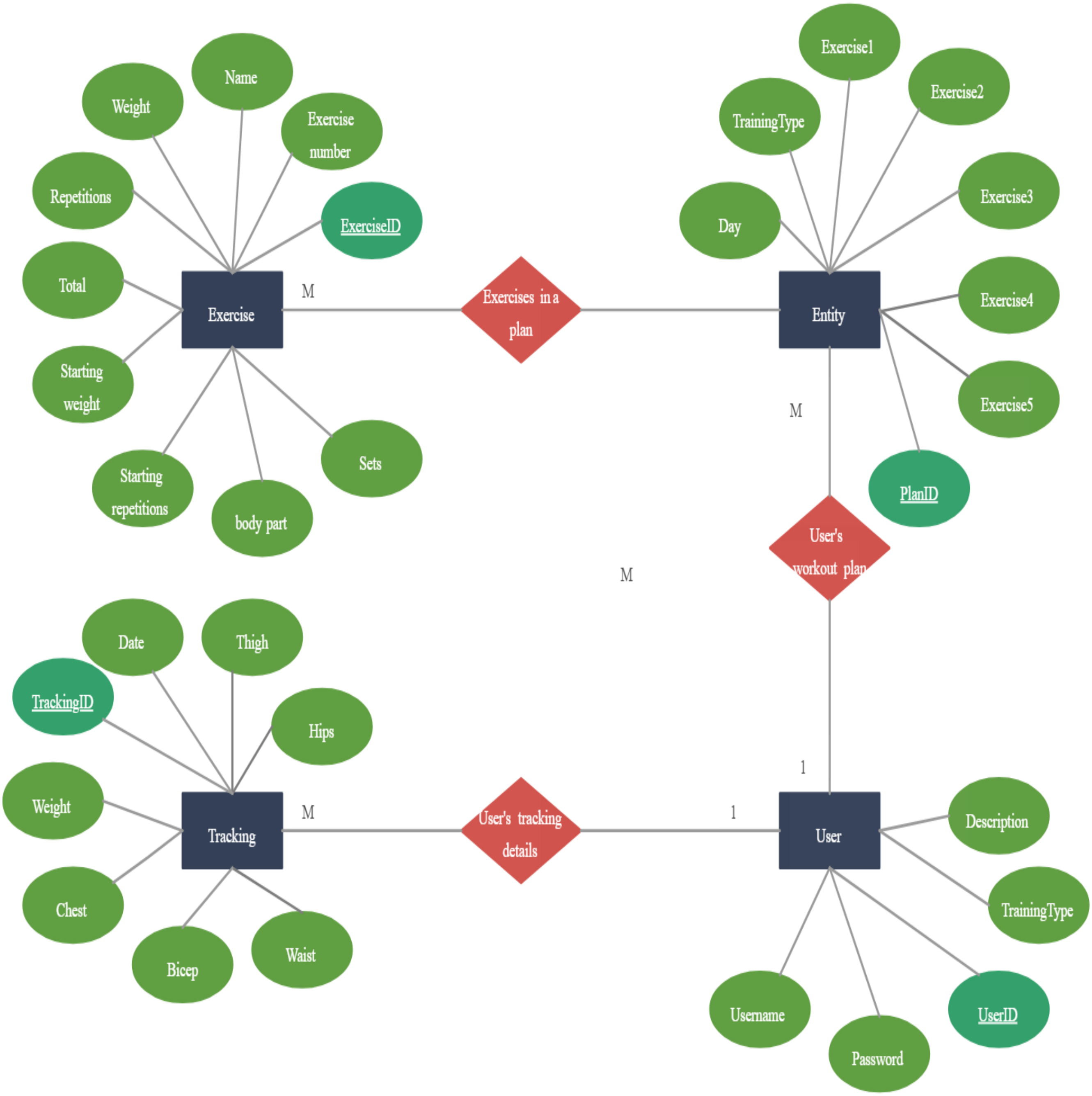


3. **Payment Table**

PaymentID	MemberID	Amount	PaymentDate	PaymentMode
P001	001	50.00	2024-01-10	Credit Card
P002	002	500.00	2023-11-05	Cash

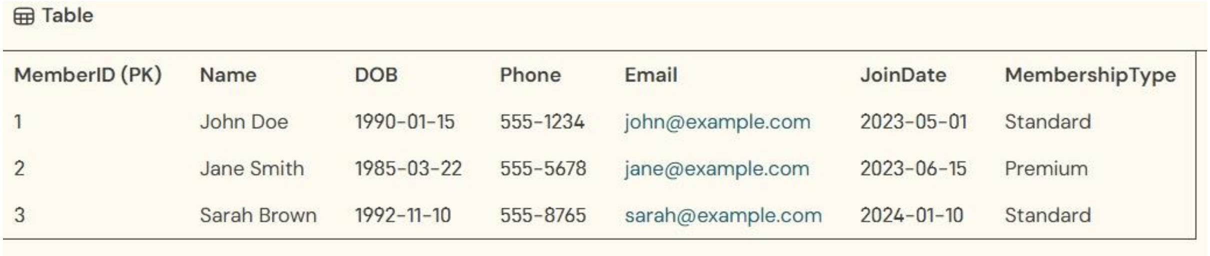
3.5 E-R DIAGRAM



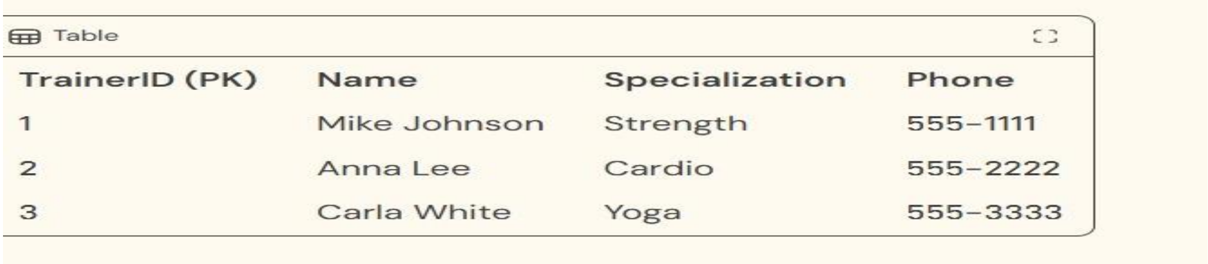

Name
Weight
Exercise number
Repetitions
ExerciseID
Total
Exercise
M
Starting weight
Starting repetitions
body part
Sets
Exercises in a plan
Exercise1
TrainingType
Exercise2
Day
Exercise3
Entity
Exercise4
Exercise5
M
PlanID
User's workout plan
M
Date
Thigh
TrackingID
Hips
Weight
Tracking
M
User's tracking details
1
1
User
Description
TrainingType
Chest
Bicep
Waist
Username
Password
UserID

3.6 NORMALIZATION

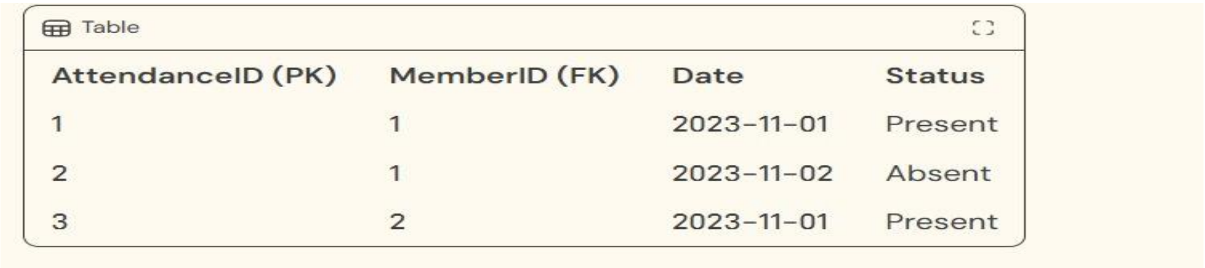
1NF TABLE



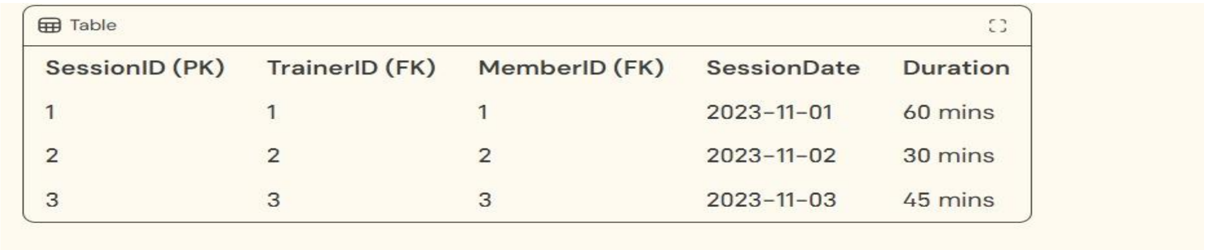
MemberID (PK)	Name	DOB	Phone	Email	JoinDate	MembershipType
1	John Doe	1990-01-15	555-1234	john@example.com	2023-05-01	Standard
2	Jane Smith	1985-03-22	555-5678	jane@example.com	2023-06-15	Premium
3	Sarah Brown	1992-11-10	555-8765	sarah@example.com	2024-01-10	Standard



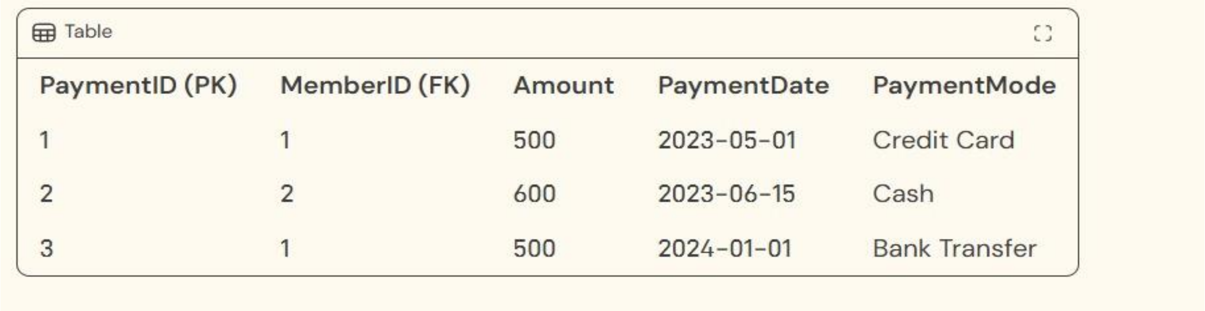
TrainerID (PK)	Name	Specialization	Phone
1	Mike Johnson	Strength	555-1111
2	Anna Lee	Cardio	555-2222
3	Carla White	Yoga	555-3333



AttendanceID (PK)	MemberID (FK)	Date	Status
1	1	2023-11-01	Present
2	1	2023-11-02	Absent
3	2	2023-11-01	Present

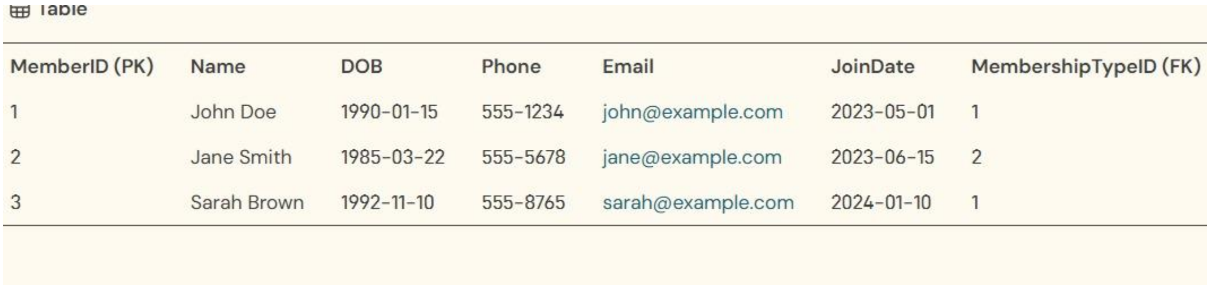


SessionID (PK)	TrainerID (FK)	MemberID (FK)	SessionDate	Duration
1	1	1	2023-11-01	60 mins
2	2	2	2023-11-02	30 mins
3	3	3	2023-11-03	45 mins

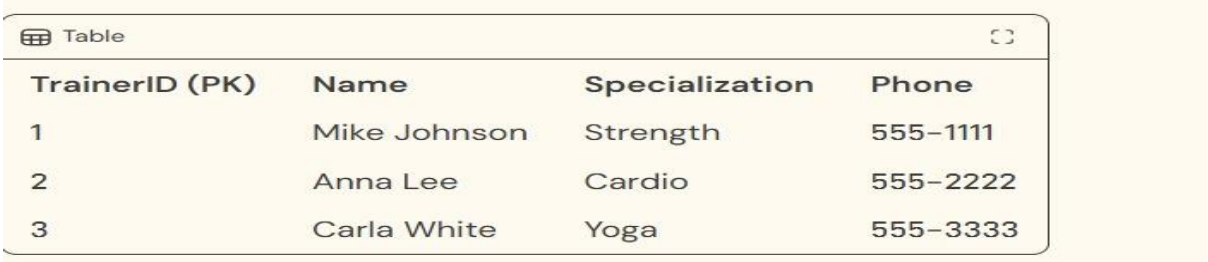


PaymentID (PK)	MemberID (FK)	Amount	PaymentDate	PaymentMode
1	1	500	2023-05-01	Credit Card
2	2	600	2023-06-15	Cash
3	1	500	2024-01-01	Bank Transfer

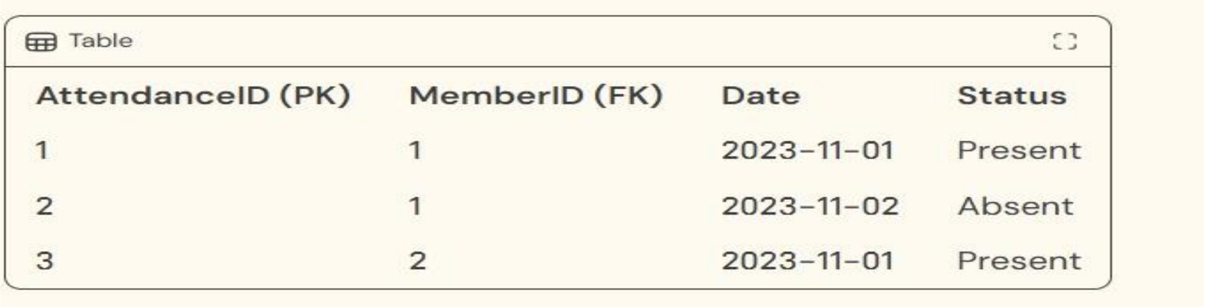
2NF TABLE



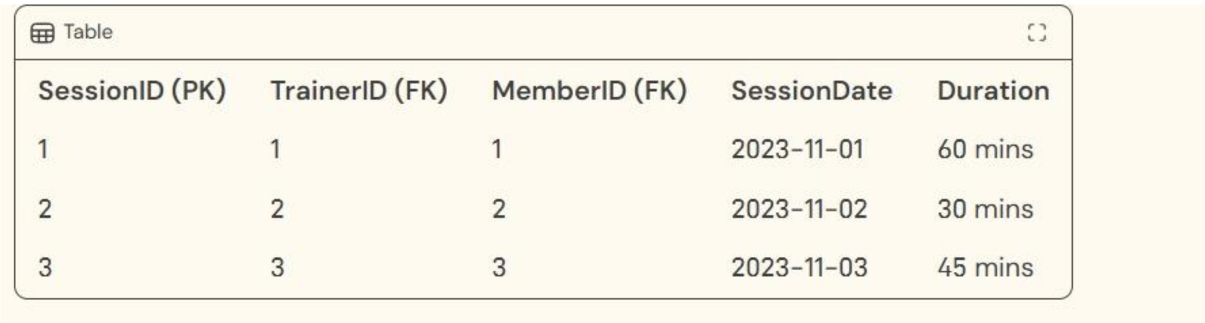
MemberID (PK)	Name	DOB	Phone	Email	JoinDate	MembershipTypeID (FK)
1	John Doe	1990-01-15	555-1234	john@example.com	2023-05-01	1
2	Jane Smith	1985-03-22	555-5678	jane@example.com	2023-06-15	2
3	Sarah Brown	1992-11-10	555-8765	sarah@example.com	2024-01-10	1



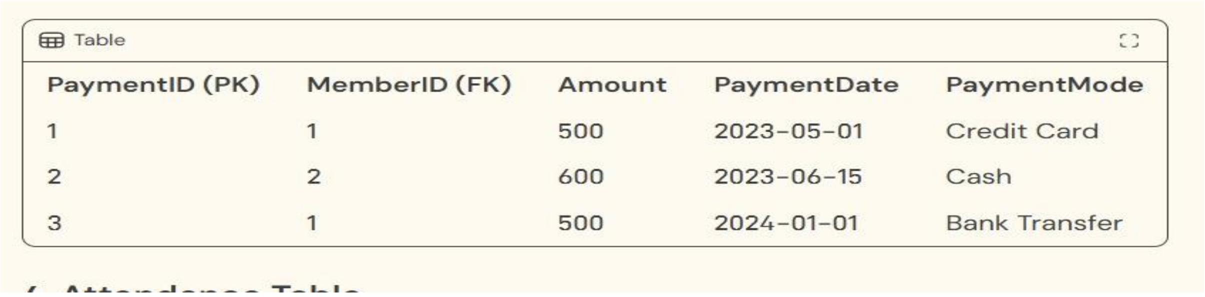
TrainerID (PK)	Name	Specialization	Phone
1	Mike Johnson	Strength	555-1111
2	Anna Lee	Cardio	555-2222
3	Carla White	Yoga	555-3333



AttendanceID (PK)	MemberID (FK)	Date	Status
1	1	2023-11-01	Present
2	1	2023-11-02	Absent
3	2	2023-11-01	Present

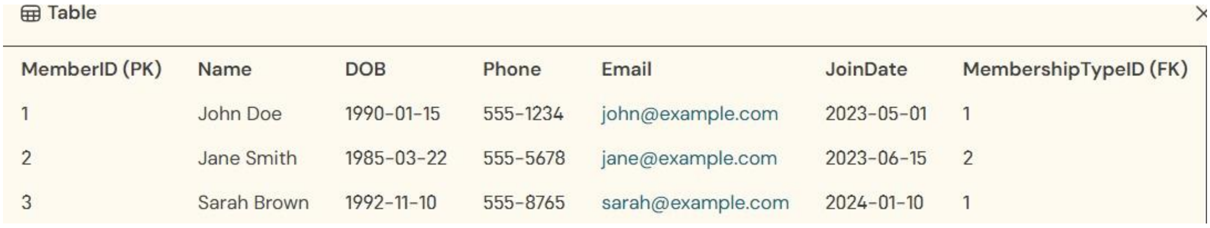


SessionID (PK)	TrainerID (FK)	MemberID (FK)	SessionDate	Duration
1	1	1	2023-11-01	60 mins
2	2	2	2023-11-02	30 mins
3	3	3	2023-11-03	45 mins

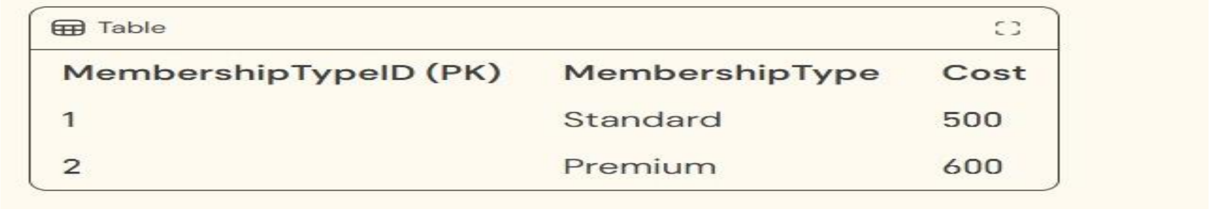


PaymentID (PK)	MemberID (FK)	Amount	PaymentDate	PaymentMode
1	1	500	2023-05-01	Credit Card
2	2	600	2023-06-15	Cash
3	1	500	2024-01-01	Bank Transfer

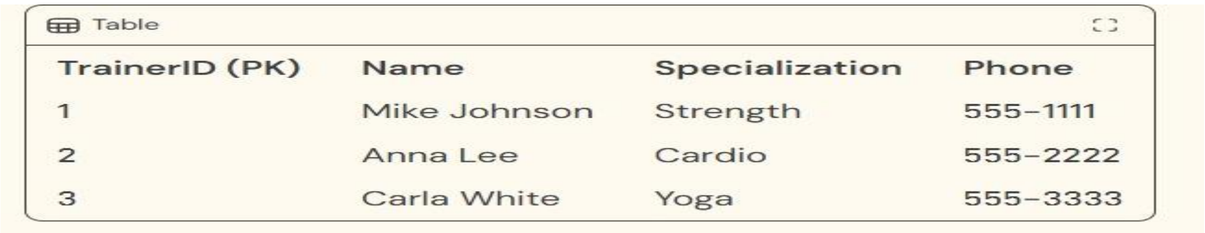
3NF



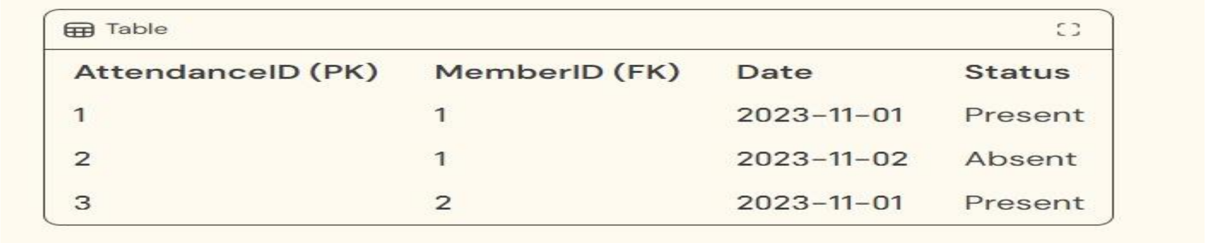
MemberID (PK)	Name	DOB	Phone	Email	JoinDate	MembershipTypeID (FK)
1	John Doe	1990-01-15	555-1234	john@example.com	2023-05-01	1
2	Jane Smith	1985-03-22	555-5678	jane@example.com	2023-06-15	2
3	Sarah Brown	1992-11-10	555-8765	sarah@example.com	2024-01-10	1



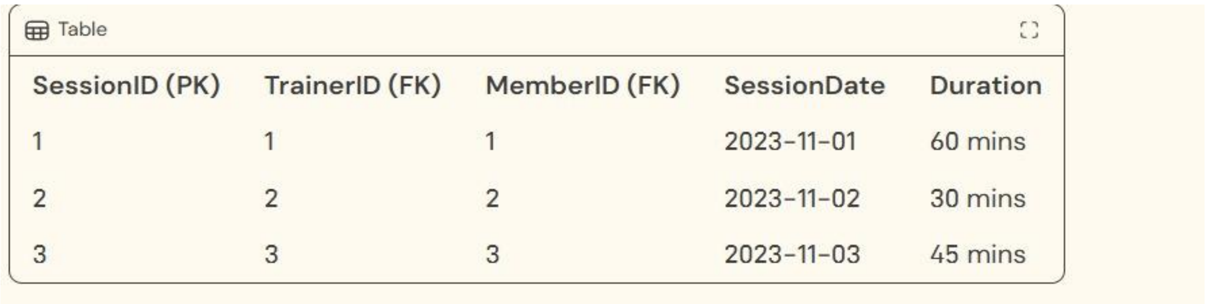
MembershipTypeID (PK)	MembershipType	Cost
1	Standard	500
2	Premium	600



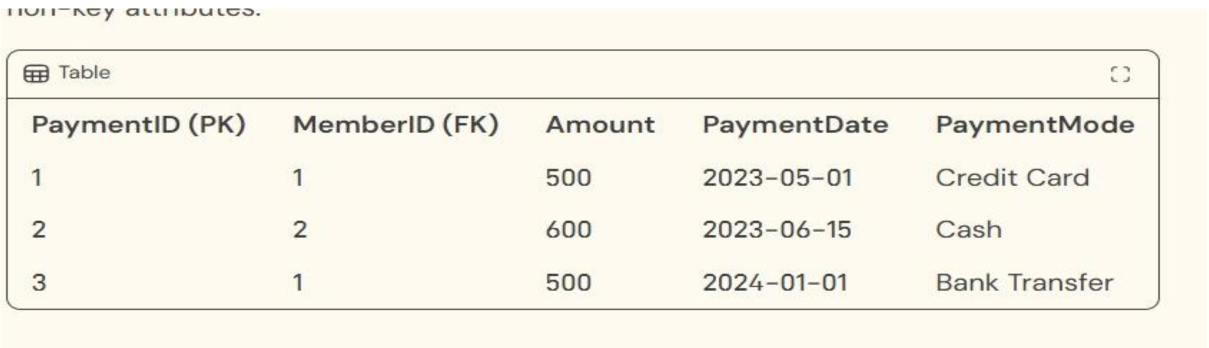
TrainerID (PK)	Name	Specialization	Phone
1	Mike Johnson	Strength	555-1111
2	Anna Lee	Cardio	555-2222
3	Carla White	Yoga	555-3333



AttendanceID (PK)	MemberID (FK)	Date	Status
1	1	2023-11-01	Present
2	1	2023-11-02	Absent
3	2	2023-11-01	Present



SessionID (PK)	TrainerID (FK)	MemberID (FK)	SessionDate	Duration
1	1	1	2023-11-01	60 mins
2	2	2	2023-11-02	30 mins
3	3	3	2023-11-03	45 mins


non-key attributes.

PaymentID (PK)	MemberID (FK)	Amount	PaymentDate	PaymentMode
1	1	500	2023-05-01	Credit Card
2	2	600	2023-06-15	Cash
3	1	500	2024-01-01	Bank Transfer

PROGRAM CODE

4.1 Code Details and Code Efficiency

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.mycompany.gymmanagementsystem;
import database.ConnectionProvider;
import java.awt.Color;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
/**
 *
 * @author quach
 */
public class benefitsPanel extends javax.swing.JPanel {
/ /**
     * Creates new form NewJPanel
     */
   public benefitsPanel() {
       initComponents();
       basic1.setSelected(false);
       basic2.setSelected(false);
       plus1.setSelected(false);
```

```
        plus2.setSelected(false);
        plus3.setSelected(false);
        premium1.setSelected(false);
        premium2.setSelected(false);
        premium3.setSelected(false);
        premium4.setSelected(false);
    }
    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents
    private void initComponents() {
        jLabel1 = new javax.swing.JLabel();
        plus1 = new javax.swing.JCheckBox();
        basic1 = new javax.swing.JCheckBox();
        basic2 = new javax.swing.JCheckBox();
        plus2 = new javax.swing.JCheckBox();
        plus3 = new javax.swing.JCheckBox();
        premium1 = new javax.swing.JCheckBox();
        premium2 = new javax.swing.JCheckBox();
        premium3 = new javax.swing.JCheckBox();
        premium4 = new javax.swing.JCheckBox();
        jComboBox1 = new javax.swing.JComboBox<>();
        setPreferredSize(new java.awt.Dimension(451, 571));
        jLabel1.setFont(new java.awt.Font("Tahoma", 1, 36)); // NOI18N
        jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/icons/benefiticon.png"))); // NOI18N
        jLabel1.setText("BENEFITS");
        plus1.setBackground(new java.awt.Color(204, 204, 204));
        plus1.setText("ALL GROUP EXERCISE CLASSES");
        basic1.setBackground(new java.awt.Color(204, 204, 204));
        basic1.setText("USE OF ALL STRENGTH EQUIPMENT");
        basic2.setBackground(new java.awt.Color(204, 204, 204));
        basic2.setText("USE OF ALL CARDIO EQUIPMENT");
        plus2.setBackground(new java.awt.Color(204, 204, 204));
        plus2.setText("1 PERSONAL TRAINING SESSION");
        plus3.setBackground(new java.awt.Color(204, 204, 204));
        plus3.setText("1 GROUP TRAINING SESSION");
        premium1.setBackground(new java.awt.Color(204, 204, 204));
        premium1.setText("USE OF BASKETBALL COURTS");
        premium2.setBackground(new java.awt.Color(204, 204, 204));
        premium2.setText("USE OF RACQUETBALL COURTS");
        premium3.setBackground(new java.awt.Color(204, 204, 204));
        premium3.setText("UNLIMITED STUDIO CYCLING");
        premium4.setBackground(new java.awt.Color(204, 204, 204));
        premium4.setText("UP TO TWO GUESTS PER VISIT");
        jComboBox1.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
    "Basic", "Plus", "Premium" }));
        jComboBox1.addItemListener(new java.awt.event.ItemListener() {
          public void itemStateChanged(java.awt.event.ItemEvent evt) {
            jComboBox1ItemStateChanged(evt);
          }
        });
        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);
        this.setLayout(layout);
        layout.setHorizontalGroup(
          layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

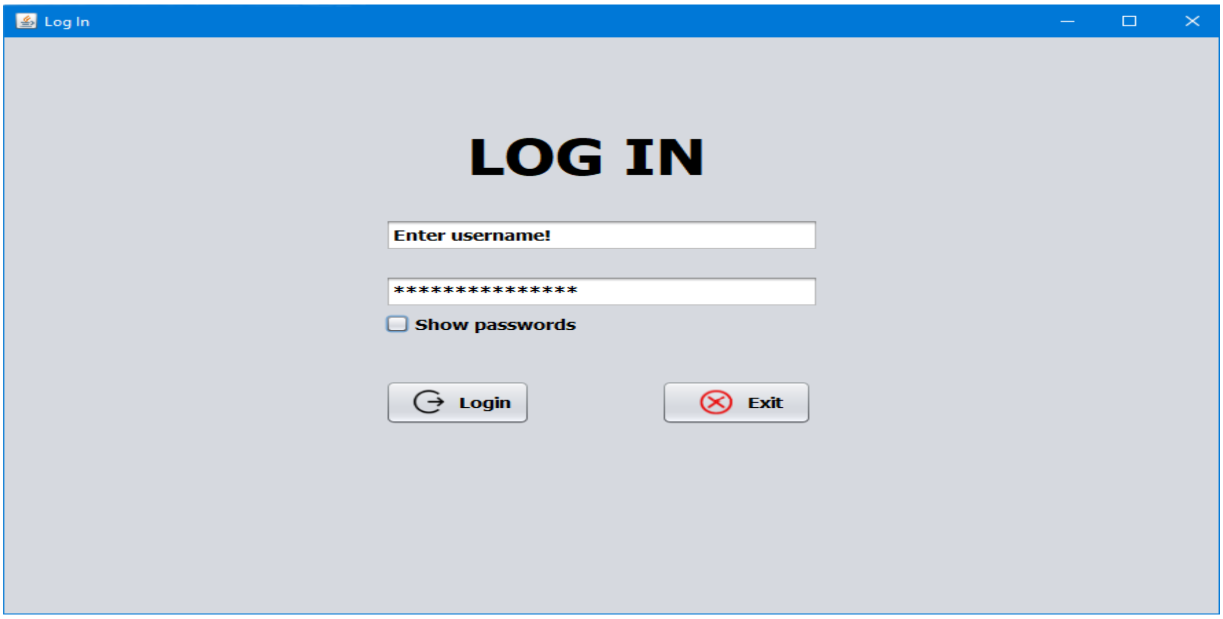
```
          .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()

             .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

      .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)

                .addComponent(basic1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                .addComponent(plus1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                .addComponent(basic2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                .addComponent(plus2, javax.swing.GroupLayout.DEFAULT_SIZE,

             .addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

             .addGap(29, 29, 29)

             .addComponent(basic1)

             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

             .addComponent(basic2)

             .addGap(18, 18, 18)

             .addComponent(plus1)

             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

             .addComponent(plus2)

             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

             .addComponent(plus3)

             .addGap(18, 18, 18)

             .addComponent(premium1)

             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

             .addComponent(premium2)

             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

             .addComponent(premium3)

             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

             .addComponent(premium4)
```

```
            .addContainerGap(151, Short.MAX_VALUE))
    );
}// </editor-fold>//GEN-END:initComponents
```

RESULT AND DISCUSSION

LOG IN

- The user will be an admin by default, and the username and password will be {admin, admin}



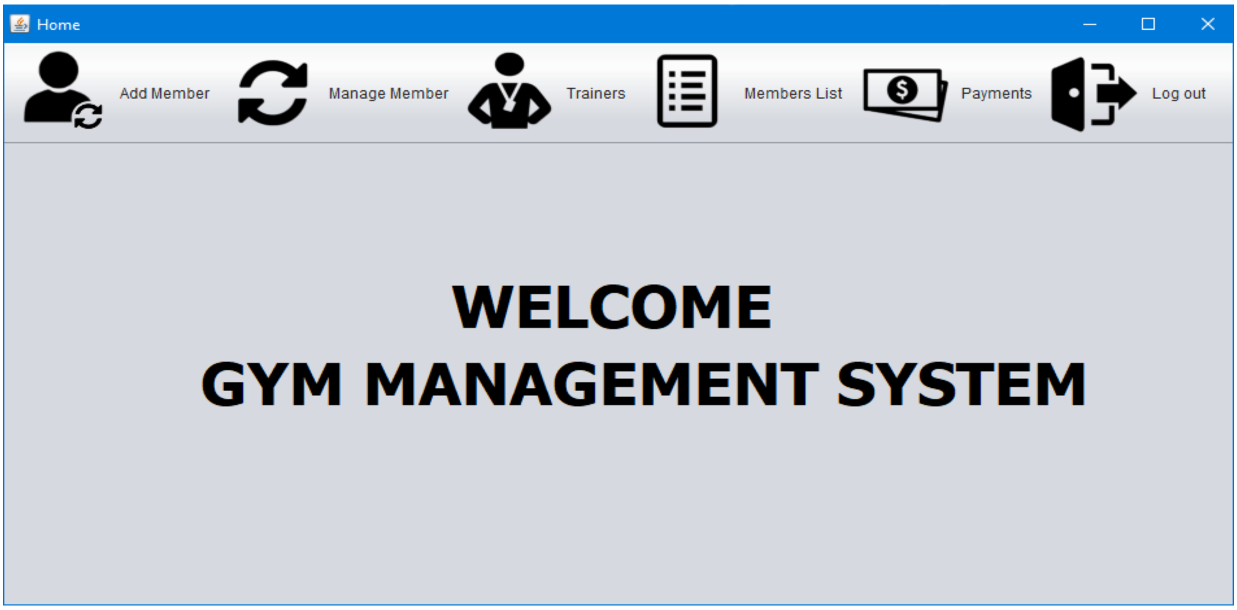
Enter username!

****

□ Show passwords

Homepage

- The admin menu as below, consists of Add Member, Manage Member, Trainers, Members List, Payments, and Log out

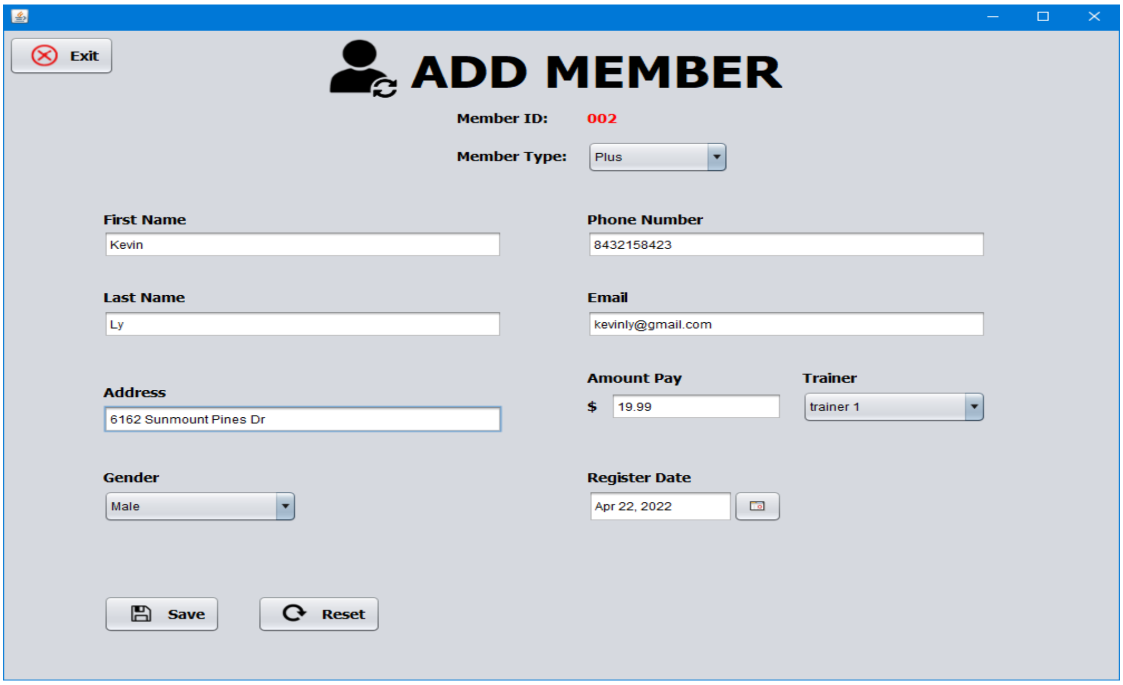


WELCOME

GYM MANAGEMENT SYSTEM

Add Member page

- To add member, click on the option "Add Members" from admin menu



Member ID: 002

Member Type: Plus

First Name: Kevin

Phone Number: 8432158423

Last Name: Ly

Email: kevinly@gmail.com

Amount Pay: $ 19.99

Trainer: trainer 1

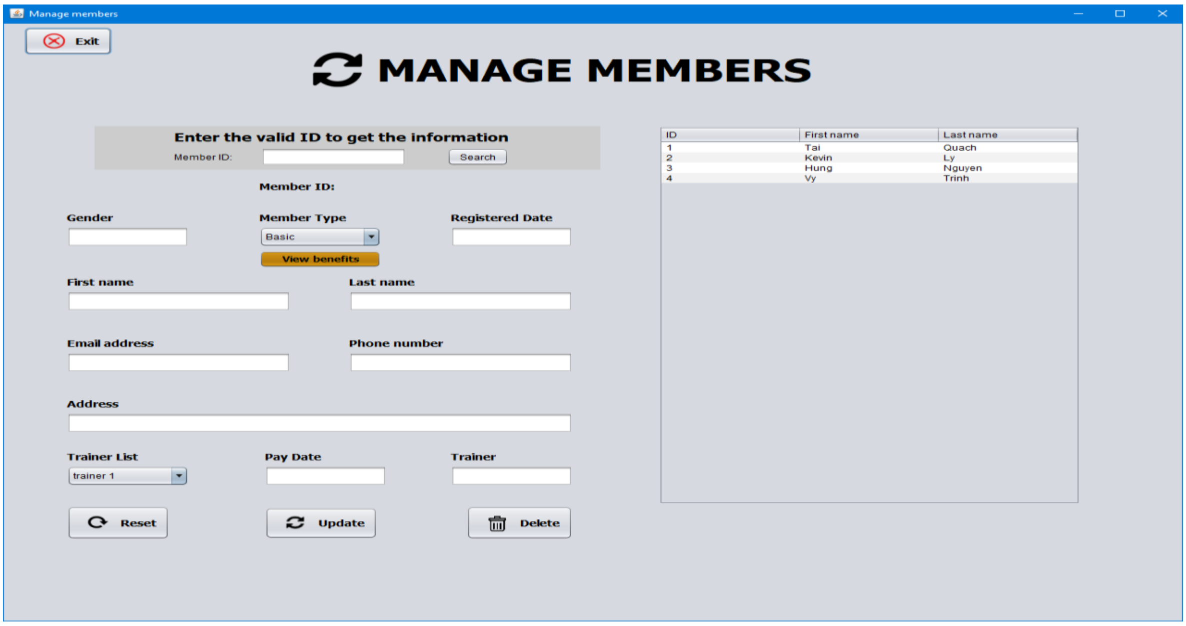
Address: 6162 Sunmount Pines Dr

Gender: Male

Register Date: Apr 22, 2022

Manage Member page

- To manage member, click on the option "Manage Members" from admin menu



Enter the valid ID to get the information

Member ID:

Member ID:

Gender

Member Type: Basic

Registered Date

First name

Last name

Email address

Phone number

Address

Trainer List: trainer 1

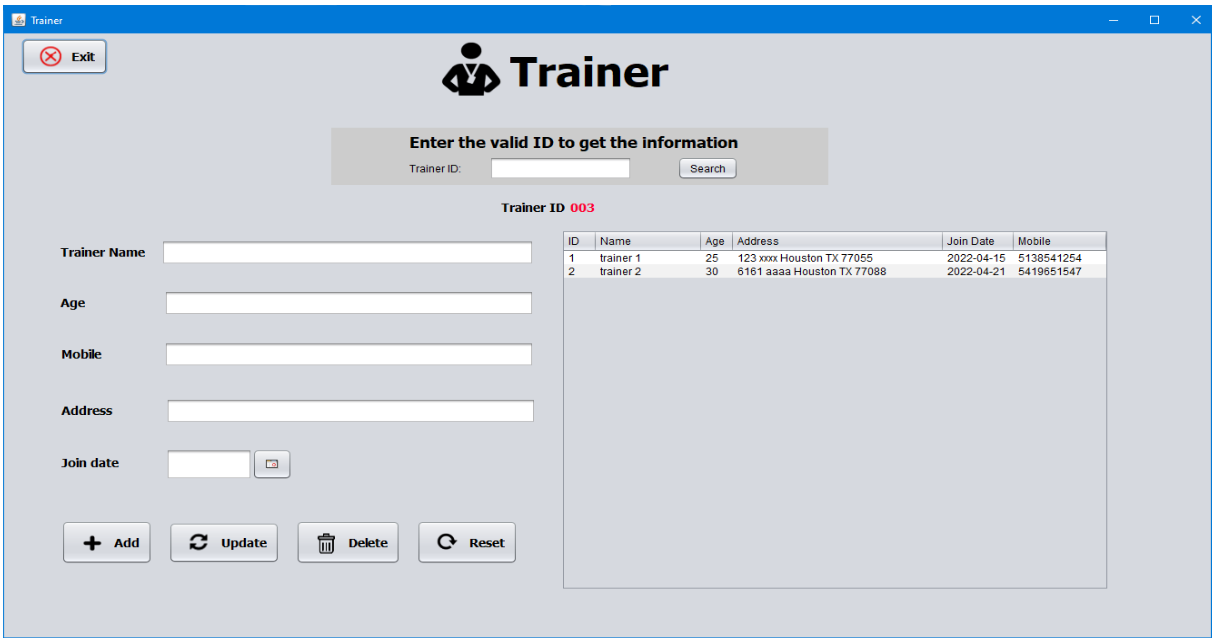
Pay Date

Trainer

ID	First name	Last name
1	Tai	Quach
2	Kevin	Ly
3	Hung	Nguyen
4	Vy	Trinh

Trainers page

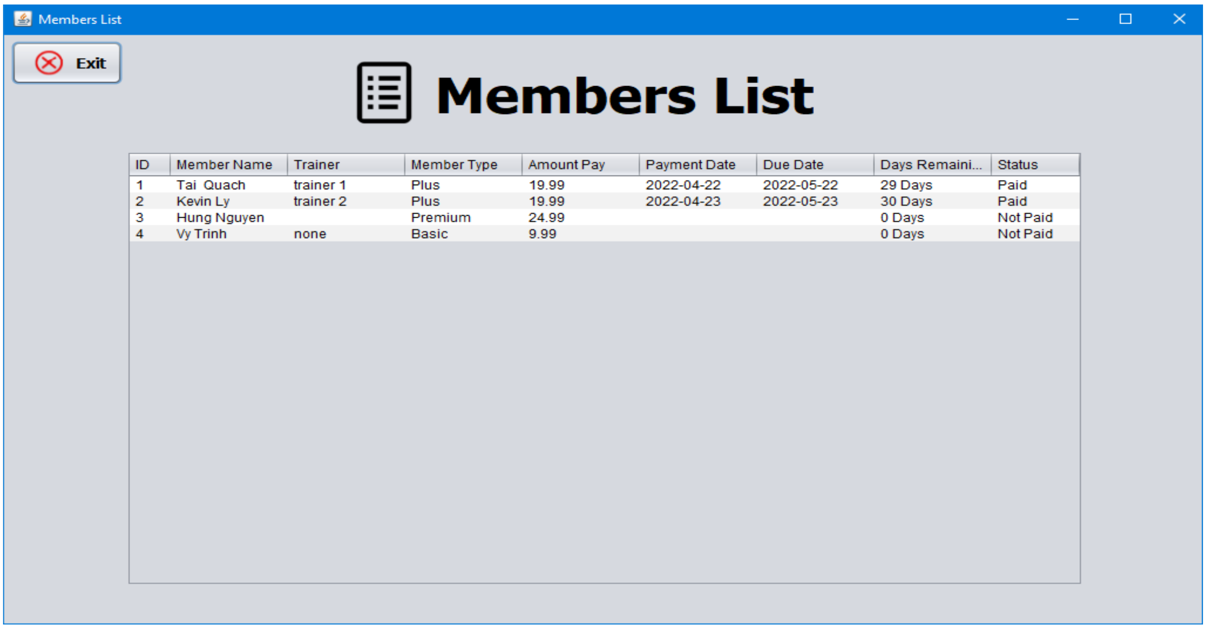
- To add/manage trainer, click on the option "Trainers" from admin menu



ID	Name	Age	Address	Join Date	Mobile
1	trainer 1	25	123 xxxx Houston TX 77055	2022-04-15	5138541254
2	trainer 2	30	6161 aaaa Houston TX 77088	2022-04-21	5419651547

Members List page

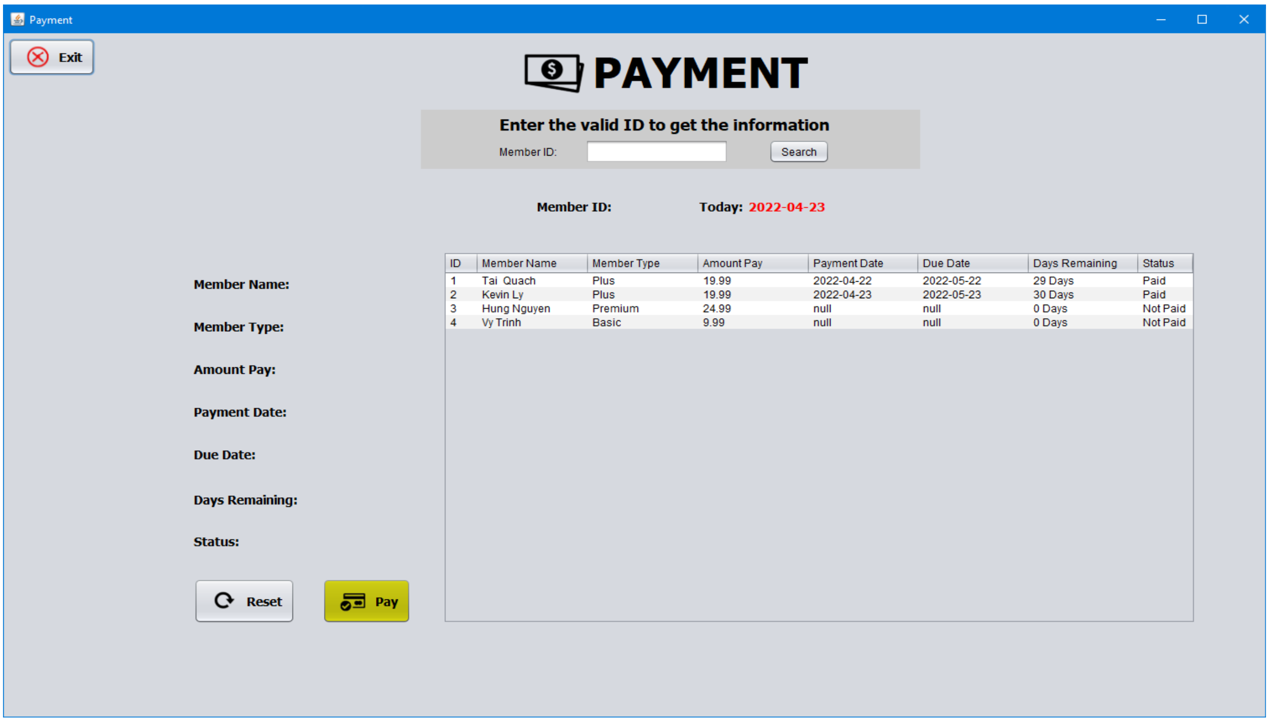
- To view the members, click on the option "Members List" from admin menu



ID	Member Name	Trainer	Member Type	Amount Pay	Payment Date	Due Date	Days Remaini...	Status
1	Tai Quach	trainer 1	Plus	19.99	2022-04-22	2022-05-22	29 Days	Paid
2	Kevin Ly	trainer 2	Plus	19.99	2022-04-23	2022-05-23	30 Days	Paid
3	Hung Nguyen		Premium	24.99			0 Days	Not Paid
4	Vy Trinh	none	Basic	9.99			0 Days	Not Paid

Payments page

- To manage payments, click on the option "Payments" from admin menu



Enter the valid ID to get the information

Member ID:

Member ID:

Today: 2022-04-23

Member Name:

Member Type:

Amount Pay:

Payment Date:

Due Date:

Days Remaining:

Status:

ID	Member Name	Member Type	Amount Pay	Payment Date	Due Date	Days Remaining	Status
1	Tai Quach	Plus	19.99	2022-04-22	2022-05-22	29 Days	Paid
2	Kevin Ly	Plus	19.99	2022-04-23	2022-05-23	30 Days	Paid
3	Hung Nguyen	Premium	24.99	null	null	0 Days	Not Paid
4	Vy Trinh	Basic	9.99	null	null	0 Days	Not Paid

TESTING

In a Gym Management System, testing strategies are critical to ensure the system functions efficiently, providing a seamless experience for users (members and staff). A well-structured testing approach, including test cases and results, helps identify bugs and areas for improvement. Here's an overview of testing strategies, test cases, and common bug fixes and improvements that could be applied to a Gym Management System:

6.1 Unit Testing

Unit tests focus on individual components or functions of the system, ensuring that each part behaves as expected in isolation. For a Gym Management System, this includes testing individual modules like member registration, subscription management, payment processing, and session scheduling.

6.2 Integration Testing

Integration testing checks how well different modules interact with each other. For example, it would verify that the registration module integrates smoothly with the payment system or that membership plans are correctly linked to member profiles.

6.3 System Testing

System testing involves testing the entire Gym Management System as a whole. The goal is to ensure that the complete system meets requirements and works properly under different conditions, such as multiple users or high server loads.

6.4 Acceptance Testing

This type of testing is performed to validate whether the system meets the user’s needs. User acceptance testing (UAT) is typically done by gym staff or a sample of end-users to ensure it meets their expectations.

CONCLUSION

The Gym Management System offers an efficient and user-friendly solution for streamlining gym operations. By integrating member management, class scheduling, payment tracking, and performance monitoring into a single platform, it enhances the overall efficiency and user experience for both gym administrators and members. Built on a scalable and robust technical foundation, the system is adaptable for future enhancements such as personalized fitness recommendations, automated reminders, and analytics-driven insights. This makes it a valuable tool for promoting fitness engagement and operational excellence in any gym or fitness centre.

REFERENCES

- Database Design and Management – "Database Management Systems" by Raghu Ramakrishnan and Johannes Gehrke. McGraw-Hill, 2003.
- Java Programming – "Head First Java" by Kathy Sierra and Bert Bates. O'Reilly Media, 2005.
- MySQL Documentation – MySQL Reference Manual. Available at: https://dev.mysql.com/doc/
- Java Swing – "Java Swing" by Marc Loy, Robert Eckstein, and Dave Wood. O'Reilly Media, 2000.
- Data Flow Diagrams – "Systems Analysis and Design" by Alan Dennis, Barbara Haley Wixom, and David Tegarden. Wiley, 2015.
- Normalization in Databases – "Database Systems: The Complete Book" by Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom. Prentice Hall, 2008.
- GitHub Repository – "Gym Management System" by Quach Thien Tai. Available at: https://github.com/quachthientai/Gym-Management-System.