
🏋️ Final Project – Gym Management System

Course: Java Final Sprint – Winter 2025
Developer: May Basalo

---

i. User Documentation

💡 Overview

The **Gym Management System** is a Java-based console application built to help gyms manage users, workout classes, and memberships efficiently. It features three distinct user roles:

- 👑 **Admin** – Manage users and track revenue
- 👩‍🏫 **Trainer** – Handle workout class assignments
- 🙋 **Member** – View and purchase memberships

All data is stored securely using **PostgreSQL**, with passwords hashed using **BCrypt** for maximum security.

🚀 Key Features

- 🔐 Role-based login (Admin, Trainer, Member)
- 🔒 Secure password hashing using BCrypt
- 🔄 Full CRUD operations for classes and memberships
- 💵 Admin dashboard to track revenue
- 🗃️ PostgreSQL + JDBC for reliable data persistence

🔧 How It Works

1. User **registers** or **logs in** via the console
2. Based on their role, a **custom menu** appears

3. All data operations are handled and saved via DAO classes
4. Passwords are hashed before being stored in the database
5. The system supports data retrieval and summaries like total revenue

---

📦 Class Architecture

- User – Abstract base class for Admin, Trainer, and Member
- UserService – Handles login, registration, and role assignment
- UserDAO – Manages database access and queries
- WorkoutClass, Membership – Core models representing system entities
- WorkoutClassService, MembershipService – Business logic handlers
- GymApp – The console app entry point with user interaction

📊 Class Diagram

This diagram outlines the inheritance structure of users and how they interact with WorkoutClass and Membership.

▶️ How to Run the Application

```
git clone https://github.com/MayBasalo/Java-Final-Sprint.git
cd Java-Final-Sprint
mvn clean install
mvn exec:java -Dexec.mainClass="org.gym.GymApp"
```

---

ii. Developer Documentation

🗂️ Project Structure

Your project follows a clean, modular Maven structure.

📁 **org.gym.user → User models & services**
📁 **org.gym.membership → Membership models & DAO**
📁 **org.gym.workoutclasses → Workout class models & logic**
📁 **org.gym → Entry point: GymApp.java**

📷 ***See project-structure.png for visual breakdown.***

✍️ **Sample Javadoc**

```
/**
 * Authenticates a user using BCrypt.
 * @param username The entered username
 * @param password Raw password input
 * @return Authenticated User or null if login fails
 */
```

```
/**
 * Displays all classes assigned to the current trainer.
 */
public void showAllClasses() { ... }
```

Full Javadoc comments are available throughout the codebase for clarity.

📐 **Build Instructions**

1. **Install Java 17+**
2. **Install Maven**
3. **Run:**

```
mvn clean install
mvn exec:java -Dexec.mainClass="org.gym.GymApp"
```

📦 **Dependencies**

- **jbcrypt – Password hashing**
- **PostgreSQL JDBC Driver – Database connection**
- **Maven – Build tool and dependency manager**

🗄️ **Database Setup**

- **Use schema.sql to create required tables**
- **Use scripts.sql to insert test data and queries**
- **Run both files in your PostgreSQL environment**

---

iii. Individual Report

✅ **My Contributions**

- **Developed all core classes and services from scratch**
- **Designed and implemented the database schema**
- **Built a role-based UI with menu branching logic**
- **Implemented DAO patterns and handled PostgreSQL connections**
- **Documented the entire project and managed source control via Git**

⚠️ **Challenges I Overcame**

- **Getting BCrypt hashing and login flow working smoothly**
- **Resolving Maven build issues on macOS**
- **Merging branches without losing functionality**
- **Structuring packages and maintaining imports in VS Code**

🙏 **Thanks for checking out my project!**