User Manual

Gym management system

Leyang Zhang

Enxu Wu

CMPSC 431W

12/3/2023

System Environment and Software Requirements

To successfully deploy and run your Spring Boot application, ensure your system environment meets the following requirements:

Operating System: macOS. Ensure your macOS system is up-to-date, or at least compatible with Spring Boot 3.2.0 and MySQL 8.2.0.

MySQL: Version 8.2.0. This is a popular open-source database management system suitable for storing your application data. Download and install this version from the MySQL official website.

Installation and Configuration Guide

Installing IntelliJ idea

- Download IntelliJ IDEA from the JetBrains website.
- Install IntelliJ IDEA on your computer.

Installing MySQL:

- Go to the MySQL official website and download the MySQL 8.2.0 installer for macOS.
- Follow the steps in the installation wizard to complete the installation.
- Set the password for the root user during installation and remember it, as you will need it later when configuring your Spring Boot application.

Step-by-Step Project Deployment Guide

Step 1: Download and Install MySQL

- Visit MySQL Official Website: Navigate to the MySQL download page.
- Select MySQL Version: Choose the appropriate MySQL version for your operating system.
- Install MySQL: Follow the installation guide on the website. Remember to note the root password set during installation, as it will be required later.

Step 2: Download and Install IntelliJ IDEA and DataGrip

- Download IntelliJ IDEA:
  - Go to the IntelliJ IDEA download page.
  - Select the suitable edition (Community or Ultimate) and download it.
  - Follow the installation instructions.
- Download DataGrip:
  - Visit the DataGrip download page.
  - Download and install DataGrip.

Step 3: Import SQL File into MySQL

- Clone the GitHub Repository: Obtain the SQL file from your project's GitHub repository.
- Create a New Database:
  - Launch MySQL using a command line interface or a graphical tool like MySQL Workbench.
  - Execute `CREATE DATABASE your_database_name;` to create a new database.
- Import SQL File:
  - Connect to the new database using DataGrip or MySQL Workbench.
  - Import the SQL file from your GitHub repository into this database.

Step 4: Download and Run the Spring Boot Project

- Clone the Spring Boot Repository: Download the Spring Boot project from GitHub to your local machine.
- Open Project in IntelliJ IDEA:
  - Start IntelliJ IDEA.
  - Open the downloaded project by selecting 'Open' and navigating to its directory.
- Run the Project:
  - Locate the main application class (marked with `@SpringBootApplication`) in IntelliJ IDEA.
  - Run it by right-clicking on the file and selecting 'Run'.

Step 5: Configure Application Settings

- Modify `application.yml`:
    - Open the `application.yml` file in your Spring Boot project.
    - Adjust settings like server port, database username, and password to match your local MySQL setup.
    - Save the changes.

Step 6: Verify Deployment

- Check Application Status:
    - After starting the Spring Boot application, ensure it's running by accessing the assigned port in a web browser (e.g., `http://localhost:8081` if the port is set to 8081).
    - Verify that the application connects to the database and functions as expected.

Step 7:Configure the Database Connection

- Open Application Properties File:
    - Locate and open the `application.properties` file in your Spring Boot project. This file is typically found in the `src/main/resources` directory.
- Database Connection Settings:
    - Inside the `application.properties` file, specify the MySQL database connection details as follows:

```
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://127.0.0.1:3306/CMPSC431W

spring.datasource.username= “Your Username”

spring.datasource.password= “Your Password”

server.port=8081
```

Username and Password: These credentials are required for connecting to your MySQL database. You can modify these values according to your MySQL setup. For instance, if you have set a different username or password for your database user, you should update these values accordingly.
Port: This is the port number on which your Spring Boot application listens. By default, Spring Boot applications typically use port 8080. In this case, it has been set to 8081. If

this port conflicts with other services on your system, or if you have specific port configuration needs, you can change this value.

User System Functionalities

Login:

- Access Account: Start at the login page by entering your email to log into your account.

Registration:

- Create Account: New users can sign up by providing personal details such as email, name, birthday, gender, and phone number.

Main User Interface:

- Navigate: Post-login, users reach a dashboard with multiple options for interaction.

Profile Management:

- Update Profile: Users can modify their personal information and details.

Comments Section:

- Write and View: Users have the ability to post new comments and read existing ones.

Payments:

- Transact: Facilitates the processing of payments for memberships and services offered.

Courses:

- Enrollment and Management: View course lists, check coach assignments, sign up for or withdraw from courses.

Facilities:

- Bookings and Usage: Check facility availability, make reservations, and manage borrowings and returns.

Announcements:

- Stay Informed: Users can read all announcements and updates related to the system.

Course Administration:

- Manage Enrollments: Control course sign-ups and view offerings sorted by name or semester.

Facility Administration:

- Oversee Facilities: Handle reservations, view facility status, and track borrowing or returns.

Financial Interactions:

- Payments and Feedback: Make payments as required and provide feedback through comments.

This organized functionality allows users to efficiently engage with the gym system, maintaining their profile, managing courses and facilities, and participating in the system community through announcements and comments.

Employee System Functionalities

Announcement Management:

- Create: Employees can publish new announcements to the system.
- Update: Existing announcements can be modified as necessary.
- Delete: Remove outdated or incorrect announcements.
- View All: Access a complete list of all system announcements.

Payment Overview:

- View All Payments: Employees can review all transactions within the system.
- User Payments: Check payment details by entering a specific user ID.

Course Management:

- View Courses: Look up all available courses.
- Add/Delete Courses: Introduce new courses to the system or remove obsolete ones.
- Enrollment and Assignment: See who is enrolled in courses and assign coaches to teach.

Employee Management:

- Add Employee: Register new employees into the system.
- Assign Roles: Allocate specific courses or fields to coaches.

User Overview:

- View All Users: Obtain a comprehensive view of all user profiles and their activity.

Comments Management:

- View All Comments: Monitor and manage user comments.

Facility Management:

- Add/Delete Facilities: Update the list of facilities as per the current offerings.
- View Facilities: Oversee all gym facilities and their statuses.
- Usage Records: Keep track of how facilities are being used.

This systematic approach enables employees to effectively oversee various aspects of the gym system, including administrative tasks, financial transactions, course scheduling, user management, and facility maintenance.