# **Gym Management System - User Guide**

**Welcome!**

Thank you for purchasing the Gym Management System! This guide will help you get started and use the system effectively to manage your gym.

**What is the Gym Management System?**

Think of this system as your digital gym assistant. It helps you keep track of all the important information about your gym in one place, making it easier to manage your members, memberships, and workout classes.

**What can you do with this system?**

* **Manage Members:** Add new members, update their information, and keep track of their contact details.
* **Manage Memberships:** Create different membership plans (like Basic, Premium, etc.) and assign them to your members.
* **Manage Workout Classes:** Create and schedule workout classes and assign trainers to them.
* **Keep Everything Organized:** Say goodbye to messy spreadsheets and paper records! This system keeps everything organized and easy to find.

**Key Things to Know**

* **Members:** These are the people who have joined your gym. Each member has a profile with their personal information and details about their membership.
* **Memberships:** These are the different plans you offer at your gym. Each membership has a name, a price, and a description of what it includes (e.g., access to the gym, access to classes, etc.).
* **Workout Classes:** These are the exercise sessions you offer at your gym. Each class has a name, a description, and a trainer assigned to it.
* **Roles:** Different people using the system might have different roles. For example, an "Admin" can do everything, while a "Trainer" might only be able to manage workout classes.

**How the System Works**

Imagine the system as having three main sections:

[Members] <-----> [Memberships]

^

|

|

|

[Workout Classes]

* Members can have Memberships.
* Workout Classes are taught by Trainers (who are also Members).

**Getting Started**

1. **Open the Program:** Find the "Gym Management System" icon on your computer and double-click it to open the program.
2. **Main Menu:** You'll see a main menu with a list of options.

**Using the System**

The system uses menus to guide you. Here's how to navigate:

* **Select an Option:** Type the number next to the option you want to select and press Enter.
* **Go Back:** To go back to the previous menu, look for an option like "Back to Main Menu" or "Return" and select it.

**Common Tasks**

* **Adding a New Member:**
  1. From the Main Menu, select "User Management" (usually by typing '1' and pressing Enter).
  2. Select "Create User".
  3. Enter the member's information when prompted (username, password, email, phone number, address).
  4. Select a role for the member (Admin, Trainer, or Member) by typing the corresponding number and pressing Enter.
  5. The system will confirm that the member has been added.
* **Managing Memberships:**
  1. From the Main Menu, select "Membership Management" (usually by typing '2' and pressing Enter).
  2. Follow the on-screen prompts to create, update, or delete memberships.
* **Managing Workout Classes:**
  1. From the Main Menu, select "Workout Class Management" (usually by typing '3' and pressing Enter).
  2. Follow the on-screen prompts to create, update, or delete workout classes.
* **Exiting the System:**
  1. From the Main Menu, select "Exit" (usually by typing '4' and pressing Enter).

I hope this guide helps you get started with the Gym Management System. Enjoy managing your gym with ease!

**Development Documentation**

This section provides a brief overview of the technical aspects of the Gym Management System for developers or those interested in the system's inner workings. It's intended to give a high-level understanding of the code structure and technologies used.

**Directory Structure**

gym-management/

├── pom.xml (Maven project file)

└── src/

├── main/

│ ├── java/

│ │ └── com/example/

│ │ ├── config/ (Database configuration)

│ │ │ └── DBConfig.java

│ │ ├── dao/ (Data Access Objects)

│ │ │ ├── PostgresUserDAO.java

│ │ │ ├── PostgresMembershipDAO.java

│ │ │ └── PostgresWorkoutClassDAO.java

│ │ ├── enums/ (Enumerations)

│ │ │ └── Role.java

│ │ ├── model/ (Data models)

│ │ │ ├── User.java

│ │ │ ├── Membership.java

│ │ │ └── WorkoutClass.java

│ │ ├── ConsoleUI.java (User interface)

│ │ └── App.java (Main application class)

│ └── resources/ (Optional: Configuration files, etc.)

└── test/

└── java/

└── com/example/

└── AppTest.java (Unit tests)

**Class Diagram**

A screenshot of a computer

AI-generated content may be incorrect.

**Key Technologies Used:**

* + **Java:** "The system is written in Java, a popular programming language known for its portability and object-oriented features."
  + **PostgreSQL:** "We use PostgreSQL as our database to store and manage the gym's data. PostgreSQL is a robust and reliable open-source relational database management system."
  + **JDBC:** "Java Database Connectivity (JDBC) is the Java API we use to connect to and interact with the PostgreSQL database."
  + **BCrypt:** "BCrypt is a password hashing algorithm used to securely store user passwords in the database."
  + **Maven:** "Maven is a build automation tool that helps us manage the project's dependencies and build process."

The project is organized into several key folders.

* + - **com.example.model:** "This folder contains the Java classes that represent the data in our system, such as User, Membership, and WorkoutClass. These classes define the structure of the data we store in the database."
    - **com.example.dao:** "This folder contains the Data Access Objects (DAOs), which are responsible for interacting with the database. The DAOs contain the code that performs the Create, Read, Update, and Delete (CRUD) operations on the database tables."
    - **com.example:** "This folder contains the main application logic, including the ConsoleUI class, which handles the user interface."
    - **com.example.config:** "This folder contains the DBConfig class, which stores the database connection details."
* **Data Access Layer:**
  + The Data Access Layer (DAL) is the part of the system that interacts with the database. It consists of the DAO interfaces and their implementations.
  + The DAO interfaces define the methods that can be used to access the database (e.g., createUser(), getMembershipById(), updateWorkoutClass()).
  + The DAO implementations (e.g., PostgresUserDAO, PostgresMembershipDAO, PostgresWorkoutClassDAO) contain the actual code that connects to the database and executes the SQL queries.
* **User Interface:**
  + "The user interface is a console-based application that allows users to interact with the system. It presents menus and prompts to guide the user through different tasks."
  + "The GymUI class handles the user interface logic, including displaying menus, getting user input, and calling the appropriate DAO methods."
* **Build and Run Instructions:**
  + "To build and run the system, you'll need to have Java and Maven installed on your computer."
  + "Open a command prompt or terminal and navigate to the project directory."
  + "Run the command mvn clean install to build the project."
  + "Run the command java com.example.GymUI to start the application."
* **Database Setup:**
  + "To set up the database, you'll need to have PostgreSQL installed on your computer."
  + "Create a new database named gym\_management (or any name you prefer)."
  + "Create the tables in the database using the SQL scripts provided in the database folder."
  + "Update the DBConfig.java file with the correct database connection details."

We hope this guide helps you get started with the Gym Management System. Enjoy managing your gym with ease!