VIBING:

INTRODUCTION:

This document provides the complete documentation for our **AI-First Software Engineering** course project. It includes the project requirements, UML-based design, and optimization steps, along with all prompts and AI chat logs.

REQUIREMENTS ELICITATION:

Prompt:

"according to this description:

Vibing allows gym owners to manage classes, schedules, and user accounts smartly and securely. Members can browse classes, see gym hours, and receive reminders before their sessions, as well as see their subscription status.

and these features:

Add/edit/remove gym classes (e.g., Boxing, Yoga)

Define opening and closing hours

Manage user accounts (members/trainers)

Notifications for upcoming classes

can you give me 5 user stories with acceptance criteria , and 5 nonfunctional requirement and 3 suggested priority "

• output:



1. As a gym owner, I want to **add, edit, and remove gym classes** so that I can keep the class list updated.

Acceptance Criteria:

- Owner can create a new class by entering its name, time, trainer, and duration.
- Owner can edit class details and save changes.
- Owner can delete a class, and it will no longer appear to members.

2. As a gym owner, I want to set and modify the gym's opening and closing hours so that members know when the gym is available.

🔽 Acceptance Criteria:

- Owner can input daily opening and closing times.
- Updated hours are reflected instantly on the member interface.
- System prevents setting overlapping or invalid hours (e.g., closing before opening).
- **3.** As a member, I want to view available classes and gym hours so that I can plan my visits easily.

🗸 Acceptance Criteria:

- Member dashboard displays a list of upcoming classes with times and trainers.
- Gym hours are visible and updated in real time.
- Only currently available classes are shown.

4. As a member, I want to receive notifications before my scheduled classes so that I don't miss my sessions.

🔽 Acceptance Criteria:

- Notifications are sent at a configurable time before each class (e.g., 30 min).
- Members can enable/disable notifications in their settings.
- Notification includes class name, time, and trainer.
- **5. As an admin**, I want to **manage user accounts (members/trainers)** so that the system stays organized and secure.

🔽 Acceptance Criteria:

- Admin can create, update, deactivate, or delete user accounts.
- Passwords are securely stored and encrypted.
- User roles (member, trainer, admin) are clearly defined and restricted.

Nonfunctional Requirements (NFRs)

- Performance: The system should load class schedules within 2 seconds of user request.
- 2. **Security:** All user data must be encrypted using **SSL/TLS** and stored securely in the database.
- 3. **Scalability:** The platform should support up to **10,000 concurrent users** without performance degradation.
- 4. **Availability:** The system should have **99.5% uptime** to ensure members can access schedules anytime.
- 5. **Usability:** The interface should be **mobile-friendly** and accessible on both web and mobile devices.

Suggested Priority Levels

Priority	Description	Example Features
High	Core functionality needed for MVP	Class management, user accounts
Medium	Enhances user experience	Notifications, gym hours display
Low	Additional polish or optimization	Advanced analytics, design themes

METHODOLOGY SELECTION:

• Prompt:

"Choose an Agile flavor (Scrum or Kanban) for incremental delivery — document why (team size, deadlines, risk). For small teams: Scrum (2-week sprints) is recommended; for continuous delivery or frequent changes: Kanban. Cite rationale in the proposal.

Use AI to help create sprint plans and estimate effort (T-shirt sizing or story points), but human judgement must validate estimates."

Output:

6 Agile Methodology Choice

Chosen Approach: Scrum

Rationale:

The **Scrum** framework is chosen because the **Vibing** project is being developed by a **small team** with clearly defined roles (e.g., developer, designer, tester). Scrum's **2-week sprints** enable structured progress through planning, development, and review cycles. This helps the team focus on **incremental delivery**, ensuring that high-priority features like class management and user accounts are built, tested, and improved early.

Scrum also provides regular feedback through sprint reviews and retrospectives, which is beneficial for reducing **risk** and aligning development with changing requirements (e.g., adding new class types or modifying notification logic). The defined sprint rhythm keeps deadlines predictable while maintaining flexibility.

If the system required **continuous updates** (e.g., daily deployment of small fixes), **Kanban** could have been a better fit. However, for this project's **structured**, **time-boxed** development with moderate risk and a small team, Scrum is the most suitable choice.

Sprint Planning and Effort Estimation

Al tools will be used to **assist** in sprint planning — such as suggesting sprint goals, estimating user story effort using **T-shirt sizing (S, M, L, XL)** or **story points**, and organizing backlog priorities.

However, **human judgement** remains essential for validating these estimates to reflect the team's actual capabilities and workload.

Example:

User Story	Effort Estimate	Size	Sprint
Add/edit/remove gym classes	8 story points	L	Sprint 1
Set opening and closing hours	5 story points	М	Sprint 1
Manage user accounts	13 story points	XL	Sprint 2
Send notifications	5 story points	M	Sprint 2
Display member subscription status	3 story points	S	Sprint 3

DESIGN (UML):

• Prompt:

"Given these user stories: **1. As a gym owner**, I want to **add, edit, and remove gym classes** so that I can keep the class list updated.

🔽 Acceptance Criteria:

- Owner can create a new class by entering its name, time, trainer, and duration.
- Owner can edit class details and save changes.
- Owner can delete a class, and it will no longer appear to members.
- 2. As a gym owner, I want to set and modify the gym's opening and closing hours so that members know when the gym is available.

🔽 Acceptance Criteria:

- Owner can input daily opening and closing times.
- Updated hours are reflected instantly on the member interface.
- System prevents setting overlapping or invalid hours (e.g., closing before opening).
- **3. As a member**, I want to **view available classes and gym hours** so that I can plan my visits easily.

🔽 Acceptance Criteria:

- Member dashboard displays a list of upcoming classes with times and trainers.
- Gym hours are visible and updated in real time.
- · Only currently available classes are shown.
- **4.** As a member, I want to receive notifications before my scheduled classes so that I don't miss my sessions.

🔽 Acceptance Criteria:

- Notifications are sent at a configurable time before each class (e.g., 30 min).
- Members can enable/disable notifications in their settings.
- Notification includes class name, time, and trainer.

5. As an admin, I want to **manage user accounts (members/trainers)** so that the system stays organized and secure.

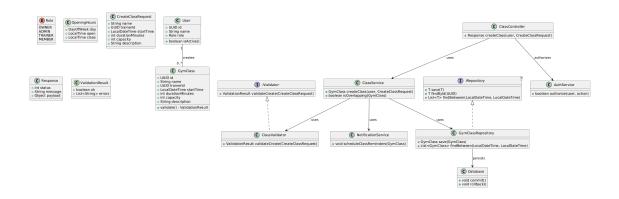
Acceptance Criteria:

- Admin can create, update, deactivate, or delete user accounts.
- Passwords are securely stored and encrypted.
- User roles (member, trainer, admin) are clearly defined and restricted.,

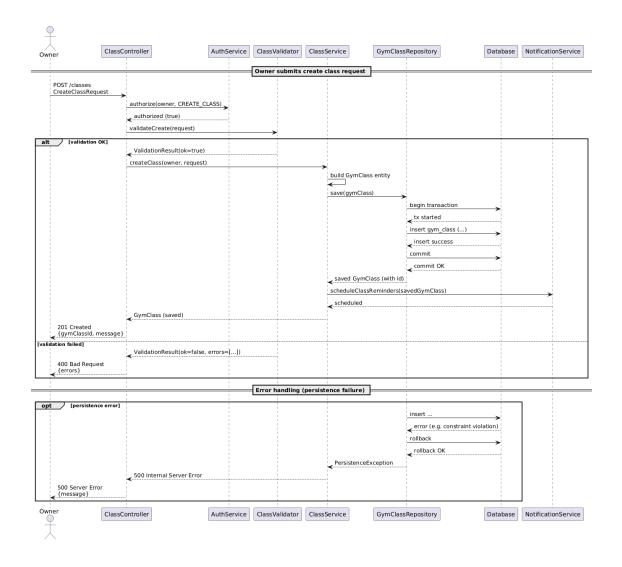
produce PlantUML class, sequence and use case diagrams for the 'create task' flow including validation and persistence steps."

• Output:

class diagram:



sequence diagram:



use case diagram:

