

## **COMP 2663 – Software Engineering 1, Fall 2025**

**Group Name: DNW**

**Contributions of each student:**

**Welton: Q1, purpose, scope and definitions, requirements UML use case diagram (Q2) and brainstorming question 1.**

**Nabil: Functional and non-functional requirements, brainstorming and UML-use case diagram**

**David: Brainstorming and Q3**

**Date: Oct 23, 2025.**

### **Project Component 2 (PC2) – Requirements: SRS and Use Cases**

You may complete this in a group of size up to 3. Only submit one PC2 per group needs to be submitted by the group leader. You should include a title page with your group name and group member names for this component.

1. Write the high-level requirements for your course project, following similar form to IEEE830-1993
  - a. What are the requirements identified
  - b. Track the amount of time spent doing this exercise.
  - c. Decide what fraction of the requirements are well understood by the team.
  - d. Estimate how long it would take to obtain 95 percent of the requirements.
  - e. State how the process you used could have been improved. Be specific and provide examples

#### **Purpose**

The purpose of the document is to state the high-level requirements needed for the Danawe fitness trainer app, which allows users to have predefined workout plans with adjustable difficulty, to book workout classes for a gym, to track their progress as well as log their own progress, and to create their own workout plans. The document will also define functional and non-functional requirements that guide the design and implementation of the software.

## Scope

Danawe Fitness will allow the users to manage their memberships, book classes, create their own workout plans, time their workouts, and receive progress reports. The software aims to work as a mobile gym membership, allowing the user to work out from home if unable to head to the gym, book gym classes, access the gym, and manage their membership, being able to cancel from home.

## Core features:

- Membership management and payments
- Class booking and schedule viewing
- Workout and progress tracking
- Trainer management tools
- Notifications and reminders
- Reports and analytics for admins

## Definitions and Acronyms

Member: Registered gym user with booking privileges

Trainer: workout and sessions manager

Admin: User with system control, data access, and report generation permissions

SRS: Software Requirements Specification

UML: Unified Modeling Language

## Overall Description and product Perspective

Danawe Fitness is a mobile application designed to integrate with third-party services for secure payments and notifications. It includes an admin dashboard, trainer portal, and member interface.

## User Characteristics and technical levels

Member: Books classes, tracks workouts (amateur)

Trainer: Schedules classes, logs progress (Medium)

Admin: Manages users, finances, and reports (Hard)

## Constraints

-Response time under 5 seconds per transaction.

## Assumptions

-Payment handled via third-party API (e.g., PayPal, Apple Pay).

- All users possess valid email accounts for verification and notifications.

## Specific Requirements

- Functional Requirements:

- Membership Management
- The system shall allow users to sign up, renew, or cancel memberships.
- The system shall store membership history and renewal dates.
- The system shall disable access for expired memberships
- Class Booking and Scheduling
- The system shall allow members to browse available classes.
- Trainers shall be able to add, edit, or remove class schedules.
- The system shall prevent overbooking by limiting slots.
- Trainers shall mark class attendance.
- Trainers shall design and assign workout plans to members.
- Workout and Progress Tracking
- Members shall log workout details.
- The system shall generate progress analytics (e.g., calories burned, sessions completed).
- Notifications and Reminders
- Members shall receive reminders for upcoming classes.
- The system shall send renewal alerts and payment confirmations.

-Non-Functional Requirements

Type Description

Usability: Interface needs to be intuitive and responsive

Performance: Response time must not exceed 5 seconds.

Security: Data must be encrypted

Reliability: Uptime of 99% or higher.

Maintainability: The system should facilitate easy module updates.

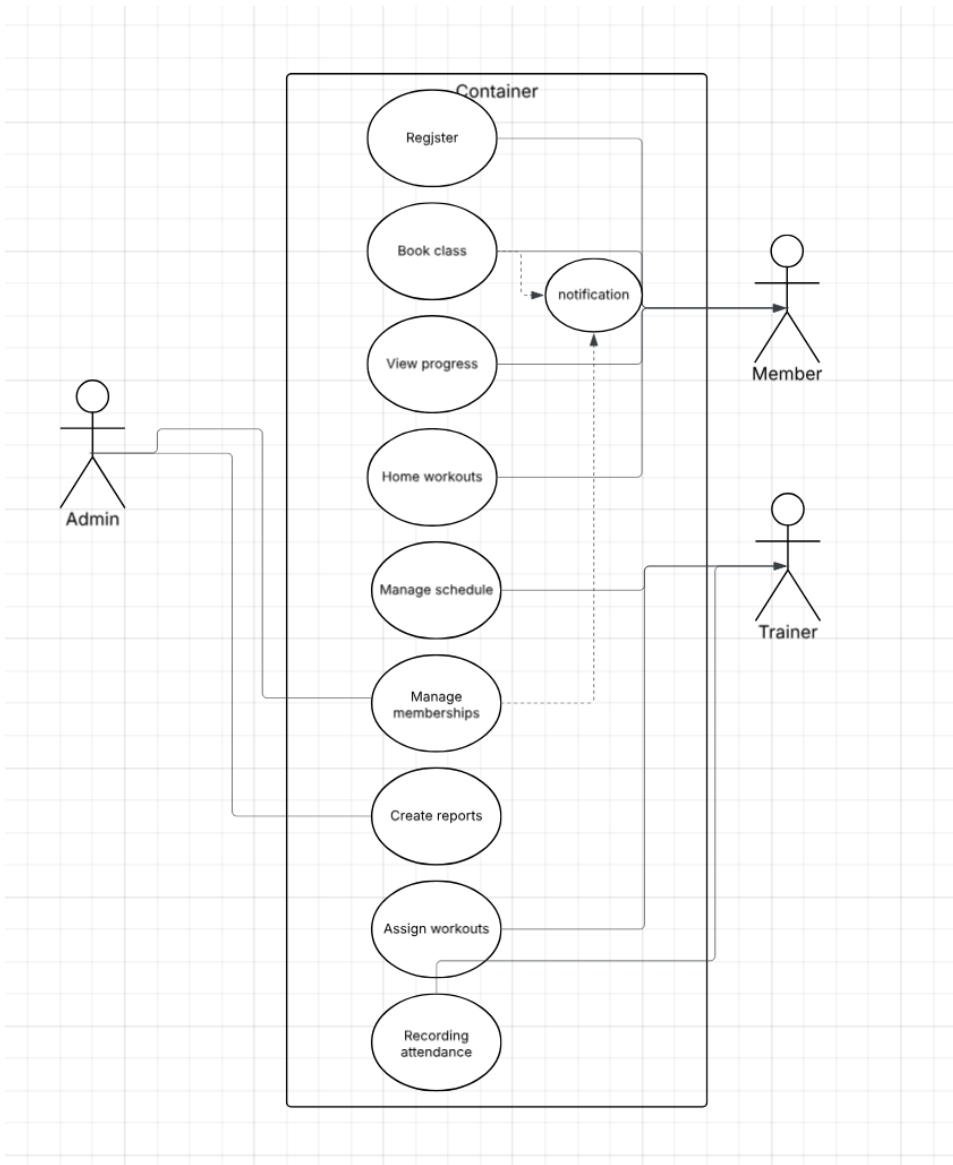
-Process Reflection

Time Spent: 3 hours

Fraction of requirements understood: 70%

Estimated time to reach 95%: +4 hours

2. Create a set of high-level use cases and provide UML use case diagrams that correspond to the essential requirements of your project.



3. Create a (potentially revised) Gantt chart that will be used for the remainder of your project.

Task	Duration (Weeks)	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6
Requirements Finalization	1						
System Design (Architecture & UI)	5						
Backend Development (API, DB)	10						
Frontend Development (Mobile UI)	10						
Feature Integration	5						
Testing & QA	5						
Deployment & Documentation	3						