Individual Report – Christian Rose

# Contributions

Throughout the development of our Gym Management System, I contributed to several key areas of the project, especially in designing the console interface and building the full membership management system.

Here are the areas I focused on:

• Console UI Design:  
 - I designed and implemented the full console-based interface using Java's Scanner for input.  
 - I created intuitive, role-based menu systems for Admins, Trainers, and Members.  
 - Each menu includes user-friendly prompts and error handling for invalid inputs.

• Membership System:  
 - I developed the complete membership functionality from scratch, including the `Membership` class.  
 - I implemented the data layer (`MembershipDAO`, `MembershipDAOImpl`) to allow saving and retrieving memberships.  
 - I added logic in `MembershipService` to allow Trainers and Members to purchase memberships.  
 - Admins can view all active memberships and see the total revenue collected.

• Documentation:  
 - I wrote the User Documentation section, which explains the system for users with no technical background.  
 - I included clear step-by-step usage instructions and a simplified class diagram.  
 - I added Javadoc comments to all files to improve code readability.

• GitHub Contributions:  
 - My commits were made under the username `CRose1497`.  
 - I contributed regularly to the project on the shared repository: https://github.com/NSparkes95/java-final-sprint-s3/commits/main  
 - My commits included work on the membership logic, console menus, user doc write-up, and Javadoc additions.

# Challenges Faced

• Managing Scanner Inputs:  
 - One of the early challenges I faced was managing mixed inputs using Scanner.  
 Switching between `nextLine()`, `nextInt()`, and other types often caused bugs if not handled carefully.  
 I solved this by restructuring how inputs were consumed and tested every menu thoroughly.

• Designing for Non-Technical Users:  
 - Writing documentation that is clear and beginner-friendly was harder than expected.  
 I had to break down technical processes into plain English and simplify steps as much as possible.

• Multi-role Logic:  
 - Connecting the same membership logic for both Trainers and Members took some thought.  
 I had to ensure shared logic was reused without duplicating code, while still maintaining role-specific menus.

• Git Collaboration:  
 - Like most group projects, syncing changes on GitHub required coordination.  
 I was careful to pull often, resolve any conflicts early, and avoid overwriting others’ work.