

Requirements Document

Preface:

This iteration 1 of the requirements document. This document is expected to be read by the engineers working on the project, the users that should benefit from the project and the engineering team leader.

Introduction:

The need for this system arises from the expanding demand on **gyms**. A management system for such an entity is required to make employees life easier makes it easier for managers to track what is happening and generate reports.

Glossary:

- **User:** The main target user of the system is the **gym manager**.
- **Member:** Refers to the customers that need to be tracked and stored in the **management system** (members of the gym).
- **Employee:** General term refers to the employees in the gym, but they won't have direct access to the system. This include **accountants** and **personal trainers**.
- **Server:** a computer program or device that provides a service to another computer program.
- **Client:** a computer or a program that, as part of its operation, relies on sending a request to another program or a computer hardware or software that accesses a service made available by a server.
- **Database:** a structured set of data held in a computer, especially one that is accessible in various ways.
- **Tuple:** one row in a database.

User requirements definition:

Functional requirements:

1. User shall login to the system and verify his identity.

2. User shall add new members.
3. User shall add new employees.
4. User shall change members data.
5. User shall change employees data.
6. User shall add new memberships.
7. User shall generate monthly income reports.
8. User shall track every member invitations and membership status.
9. User shall be able to search for a specific member.
10. User shall be able to search for a specific employee.
11. User shall manage prices and discounts.

Non-functional requirements:

1. Ease of use.
2. Security.
3. Robustness.
4. Reliability.

System architecture:

We will use **server-client architecture**, server will contain the database and all the information of the members and employees while client will be the link between the user and the server.

System requirements specification:

Functional requirements:

1. Login will make a request to the database to verify the credentials of the manager.
2. User will have a form that he will enter the new member information into and this information shall be add as a tuple in the database.
3. User will have a form that he will enter the new employee information into and this information shall be add as a tuple in the database.

4. User shall search using the member's name, ID or phone number then select the piece of information that he wants to change.
5. User shall search using the employee's name, ID or phone number then select the piece of information that he wants to change.
6. User will have a form that he will enter the information necessary for a new membership (information will be specified in the Database schema).
7. User shall select a month, the member will request the information from the server and the server shall reply with the necessary information for the monthly income report.
8. User shall get notified when a membership expires.
9. User shall search for a member using his name, ID or phone number.
10. User shall search for an employee using his name, ID or phone number.
11. User shall see prices corresponding to the duration of each membership and shall be able to change this prices.

Non-functional requirements:

1. It shall take no longer than **three hours** for the user to learn the system and get to an average of **three misuses** per hour.
2. Credentials of the user should be encrypted.
3. System shall take no longer than **five minutes** to restart after failure.
4. System shall fail **less than three times per day**.

System evolution:

System is designed based on the assumptions that a wider variety of users may use the system including for example employees and it's assumed that the system can be upgraded to a web based app that's what motivated using the client-server architecture.

Appendices:

