
Ardelean Eugen-Richard
30433

Gym Management System
Supplementary Specification

Version <1.0>

Gym Management System	Version: <1.0>
Supplementary Specification	Date: <19/Mar/18>
<document identifier>	

Revision History

Date	Version	Description	Author
19/Mar/18	1.0	First version of the filled-in document	Ardelean Eugen-Richard

Gym Management System	Version: <1.0>
Supplementary Specification	Date: <19/Mar/18>
<document identifier>	

Table of Contents

1.	Introduction	4
2.	Non-functional Requirements	4
2.1	Availability	4
2.2	Performance	4
2.3	Security	5
2.4	Testability	5
2.5	Usability	5
3.	Design Constraints	5

Gym Management System	Version: <1.0>
Supplementary Specification	Date: <19/Mar/18>
<document identifier>	

Supplementary Specification

1. Introduction

The Supplementary Specification captures the system requirements that are not readily captured in the use cases of the use-case model. Such requirements include:

Legal and regulatory requirements, including application standards.

Quality attributes of the system to be built, including usability, reliability, performance, and supportability requirements.

Other requirements such as operating systems and environments, compatibility requirements, and design constraints.

The system I am working on, will be an application that will allow the user to live a healthy life, by giving the user the ability to calculate the amount of calories he eats, by talking to a specialist and by creating a custom program for him by one of the administrators.

2. Non-functional Requirements

This section treats the main non-functional requirements of the system and how they are incorporated in the system, as well as what is the standard for each one of them and how it may or may not be accomplished upon delivering the final version of the product.

2.1 Availability

Availability is the ratio of time a system or component is functional to the total time it is required or expected to function.

The application must have high availability since users must be able to use it any time of the week, for maintenance purposes Sunday has been chosen as the offline day, so the application will be available 6/7 days of the week

2.2 Performance

Performance is the amount of work accomplished by the application and can be expressed using many other notions such as: response time (needs to be short), utilization of computer resources (how much memory is needed to store intermediate results of the variables). Performance can be measured, informally, by choosing one of the most complex use case (such as report generation) as presented below.

The system will be designed to have a small response time, so that the user won't have to wait too much when using it.

- Performance is the amount of resources needed to perform an operation
- Source of stimulus: user
- Stimulus: user clicking a button that performs an operation
- Environment: the application performs the operation
- Artifact: modification of database, and/or use of processor for calculations
- Response: operation is performed
- Response measure: the time it takes to complete the operation

Gym Management System	Version: <1.0>
Supplementary Specification	Date: <19/Mar/18>
<document identifier>	

2.3 Security

Software security is the idea of engineering software so that it continues to function correctly under malicious attack

The security of such an application is directly related to the ease by which a person could access the account of another person or that of an admin. For this, all accounts are protected by a password, that is chosen by the user.

Modifications to the application can be done only by an admin, to prevent damage.

2.4 Testability

Software testability is the degree to which a software artifact (i.e. a software system, software module, requirements- or design document) supports testing in a given test context. If the testability of the software artifact is high, then finding faults in the system (if it has any) by means of testing is easier. The testability of this application needs to be as good as possible, therefore a modular approach will be used in development such that the application will allow tests ranging from unit testing up to integration or even acceptance testing.

2.5 Usability

Usability is the ease of use and learnability of a human-made object such as a tool or device. In software engineering, usability is the degree to which a software can be used by specified consumers to achieve quantified objectives with effectiveness, efficiency, and satisfaction in a quantified context of use.

Regarding the interface with the user, the system aspires to be as easy as possible, providing classical components such as buttons, text fields and labels for the user to fill in the desired information. From the graphical user interface point of view, the system will be as usable as possible.

3. Design Constraints

a. Client-server architecture

The main design constraint of the application is the architectural pattern: Client-server architecture. The client-server model is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. The server will deal with the main operations on the database and will retrieve the information to the requesters.

b. Browser

The application will run on most browsers, from Microsoft Edge to Google Chrome.

c. Java Compatibility

The System will be created using Java 1.8, because it allows the use of lambda expressions and stream processing. The IDE can be either IntelliJ or Eclipse since they both allow the development of Java applications in an environment where compilation and debugging is more facile.

d. Hardware Requirements

The Gym Management System should be able to run, on any computer that has an internet connection and a browser. Though if the memory and/or processing unit are very old, it may do so more slowly.