**Software Requirements Specification for**

**Gym Management Web App**

**Version 1.0**

**Members: Tabacu Bianca Oana, Pirlac Mihai Cristian, Tudor Andrei Catalin, Vieru Bogdan**

**March 9, 2020**

**Table of Contents**

1. **Introduction** 
   1. **Purpose**
   2. **Product Scope**
   3. **Intended Audience and Reading Suggestions**
   4. **Definitions, acronyms, and abbreviations**
   5. **References**
2. **Overall Description** 
   1. **Product Perspective**
   2. **Product Function**
   3. **User Classes and Characteristics**
   4. **Operating Environment**
   5. **Design and Implementation Constraints**
   6. **User Documentation**
   7. **Assumptions and Dependencies**
3. **External Interface Requirements** 
   1. **User Interfaces**
   2. **Hardware Interfaces**
   3. **Software Interfaces**
   4. **Communications Interfaces**
4. **System Features** 
   1. **Account Login**
   2. **Account Logout**
   3. **Admin Add Member**
   4. **Admin Edit Member**
   5. **Admin Add Invoice**
   6. **Admin Generate Reports**
   7. **Admin Add Pool**
5. **Other Non-functional Requirements** 
   1. **Performance Requirements**
   2. **Safety Requirements**
   3. **Security Requirements**
   4. **Software Quality Attributes**
   5. **Business Rules**
6. **Other Requirements**

**1.Introduction**

**1.1 Purpose**

This document describes the requirements for handling the activities of a gym. The system will have a user interface, a database server in which will store information about the activity of the gym application server.

* 1. **Product Scope**

This software system will be a web application for a gym. This system will be designed to ease the client's coming to a gym. Now, they will easily track their progress or regress, they can buy their membership online and they can choose which trainor they want to choose.

## Intended Audience and Reading Suggestions

This document is for developers, project managers and users.

In the first chapter you will find the purpose and references used to make product.

In the subchapter 2.2 are enumerate the functions realized by the product and o short description of each function, information about the components of each class (tenants, administrator, employees of administration etc.). The attribute and operations of each are presented in subchapter 2.3 through an image.

Chapter 4 describes the functional requirements.

* 1. **Definitions, acronyms, and abbreviations**

This Document was created based on the IEEE template for System Requirement Specification Documents.

N.A – Not available

Font – Arial

Dimension – 11

The document is structured on chapters and each chapter in more subchapters.

The language used for this document is English.

**1.5 References**

IEEE. IEEE Std 830-IEEE Recommended Practice for Software Requirements

<https://bitsbucket.com/>

<https://www.sourcetreeapp.com/>

<https://angular.io/>

<https://material.angular.io/>

<https://dotnet.microsoft.com/apps/aspnet>

<https://www.microsoft.com/en-us/sql-server/sql-server-downloads>

**2. Overall Description**

**2.1 Product Perspective**

The “Aplicatie web cu baza de date pentru membrii unei sali de fitness” is a virtually self-contained managing system; however, it will require users to have access to a web browser on their workstation computer. This means that the users of the system do not need to invest in any other software to get the most out of this software system.

**2.2 Product Function**

* **Login**
* **User/client interface**
* **Administrator interface**
* **Book a class**
* **Apply for a job**
* **Track client’s progress by introducing data(i.e : weight, height, workout informations)**

**2.3 User Classes and Characteristics**

The application users will be company employees which will have administrator privileges and members which will have regular member accounts.

**2.4 Operating Environment**

As mentioned before the app will require users to have access to a web browser on their workstation computer like Google Chrome or any other modern browser.

**2.5 Design and Implementation Constraints**

Creating a user interface which is both effective and easily navigable will pose a difficult challenge. Other constraints such as a weak web-host and slow internet browsers (“Internet Explorer”) are also worth considering. Our app is meant to be quick and responsive, even when dealing with large groups and transactions, so each feature must be designed and implemented with efficiency in mind.

Backend language: C#

Frontend: HTML, CSS, Typescript

Framework: ASP.NET Core

**2.6 User Documentation**

NA

**2.7 Assumptions and Dependencies**

The web application assumes that the user has a computer with an Internet connection and a web browser to access the app. The system may not behave correctly when used with internet browsers other than Firefox and Google Chrome.

**3. External Interface Requirements**

**3.1 User Interfaces**

The layout should be as follows:

⦁ The header which displays:

⦁ the logo

⦁ the currently logged user

⦁ the “Logout” and “Manage Account” menu

⦁ the menu, immediately under the header

⦁ The body, which displays

⦁ the content

⦁ The footer, which displays

⦁ contact

There will be two different user interfaces that will accompany this website: one for the users and one for the administrators.

The administrators will be able to add users and remove them.

**3.2 Hardware Interfaces**

NA

**3.3 Software Interfaces**

The application is connected to a database (SQL Server). The database will communicate with the server, and the server will send the information to the client application. This a mock-up picture of the databases.

**3.4 Communications Interfaces**

HTTP will be the communication protocol that our app will use.

**4. System Features**

**4.1 Account Login**

4.1.1 Functional Requirements

• The system allow a user to access an account

• The system allow the user to change or reset his password

• The system store the information in the database.

All fields are mandatory and are to be written in the text boxes offered by the login interface. The user have access to the app only if the username and password are correct. If the username and password doesn’t corespond the user can reset the password by entering phone number.

**4.2 Account Logout**

4.2.1 Stimulus: Click "Logout" Button

4.2.2 Functional Requirements

•The system shall allow the registered users to exit account, then access to operations are restricted.

**4.3 Admin Add Member**

4.3.1 Stimulus: Click "Add Member" Button

4.3.2 Functional Requirements

•The system allow a logged-in admin to create new member accounts and new admin accounts.

* Only admins can add new admins and members.

**4.4 Admin Edit User**

4.4.1 Stimulus: Click "Edit User" Button

4.4.2 Functional Requirements

• The system allow a logged-in admin to edit a registered member

• The system display all the information that user can modify, such as user, password, privileges.

• Members are informed with the modified details

**4.5 Admin Generate Reports**

4.5.1 Stimulus: Click "Generate Reports" Button

4.5.2 Functional Requirements

* The system allow a logged-in admin to generate a report.
* The admin will be able to select between many types of reports: new accounts, old accounts, closed accounts
* Each type of report will display the corresponding data:
  + The new accounts reports will display the name of the new users
  + The old accounts reports will display the name of the first users on the page
  + The closed accounts reports will display all the users that closed their accounts

**4.6 Admin Create Pool**

4.6.1 Stimulus: Click "Create Pool" Button

4.6.2 Functional Requirements

• The system allow a logged-in admin to create a new event (classes)

• The user will have to check in at the current event or vote for a different date.

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements**

The product shall take initial load time depending on the internet connection strength which also depends on the media-device from which the product is run. The performance shall depend upon hardware components of the client/customer.

**5.2 Safety Requirements**

Information transmission should be securely transmitted to the server without any changes in information.

**5.3 Security Requirements**

The system shall use secure sockets in all transactions that include any confidential customer information.

**5.4 Software Quality Attributes**

Availability: the system is available 24 hours a day.

Portability: a user can log in to the system at any time.

Reliability: the system can be used by multiple users concurrently. Any user can access the system using even a low performance PC.

Rule 1(required) Source code should have comments.

Rule 2(advisory) No identifier name should be reused.

Rule 3(required) All automatic variables shall have been assigned a value before being used.

Rule 4(advisory) The increment (++) and decrement (--) operators should not be mixed with other operators in an expression.

Rule 5(advisory) Write only one declaration per line.

Rule 6(advisory) Use meaningful names for variables

Rule 7(required) All if … else if constructs shall be terminated with an else clause.

Rule 8(required) All constants must be declared with capital letters and underline between words.

Rule 9(required) The operands of a logical && or || shall be primary-expressions.

Rule 10(required) There shall be no unreachable code.