Software Requirements Specification

for

Gym Management System

Version 1.0

Prepared by Zhenghao Wu & Qizhou Xie

GMS Studio(Team 5)

April 2, 2018

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Project Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Features 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. System Features 3

3.1 System Feature 1 3

3.2 System Feature 2 (and so on) 4

4. External Interface Requirements 4

4.1 User Interfaces 4

4.2 Hardware Interfaces 4

4.3 Software Interfaces 4

4.4 Communications Interfaces 4

5. Other Nonfunctional Requirements 5

5.1 Performance Requirements 5

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 6

Appendix C: Issues List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
| Zhenghao Wu | 2/4/2018 | Finish basic framework | V0.1.0 |
| Qizhou Xie | 4/4/2018 | Add basic description | V0.2.0 |
| Zhenghao Wu | 4/4/2018 | Add technical details | V0.3.0 |

# Introduction

## Purpose

The purpose of this document is to show the detail description of the Gym Management System. It will talk about the feature of the system (what the system can do for customers), the interface of the system, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the Supervisor of Software Development Workshop II’s Group Project for approval.

## Document Conventions

This document will be written using two different fonts, Arial for main content. Time New Roman for different levels of titles (Set using styles tool). Main content’s font size should be in 11pt.

## Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

## Project Scope

The targeted user groups are those who set up accounts in this system and staffs of the gymnasium, Customers of the gymnasium and Coach who work for the gymnasium. The aim of the system is to

* Give another way for customers to visit the gymnasium and help in reducing customers' time waste.
* Digitized daily operations, provide an easy way to manage the gym for the worker and manager who work in the gym.
* Help coaches to manage day-to-day appointment.

## References

[1] IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.

[2] Software Requirements Specification - Amazing Lunch Indicator. (n.d.). 1st ed. [ebook] Sweden: Sarah Geagea, etc. Available at: http://www.cse.chalmers.se/~feldt/courses/reqeng/examples/srs\_example\_2010\_group2.pdf [Accessed 5 Apr. 2018].

# Overall Description

## Product Perspective

This project is developed independently using a basic framework and is not a substitute for a product. It is not a secondary development based on previous products.

## Product Features

It will provide a platform for customers to make operations, such as recharging, viewing information of the gymnasium, inquiring their personal account and making appointment with the coach they want. It's just like things they can do at the counter. All customers will be register by the working staff, who own the admin privileges. They can help member to make appointment and recharge quota. The platform also provided functions for coaches, they can check.

## User Classes and Characteristics

* User Classes
  + Member:
    - Can make appointment with coach online
    - Can recharge quota online via WeChat or AliPay
    - Can check information
    - Can search information
  + Staff:
    - Allow access to all module
    - Can help user to make appointment with coach
    - Can create, modify, delete member & coach
    - Can recharge quota
    - Can search information
  + Coach
    - Can check appointment
    - Can search information

## Operating Environment

* This system will be and deploy and test on Windows 10 Pro (1709).
* The web framework Django 2.0+ is running under python 3.6.4
* Database is using MariaDB 10.1.29
* phpMyAdmin 4.7.4 for graphic database management Interface
* PHP related components are running under Apache 2.4.29
* Page browsing using Chrome 65.0.3325.181(64-bit)

## User Documentation

* Deployment Guide
* System Introduction

## Assumptions and Dependencies

* Front-end:
  + Bootstrap v4.0.0
  + tabler v0.0.21(This project is being iterated at high speed and the version may change)
  + sweetalert2 v7.17.0
  + jQuery v3.3.1
  + yue.css 0.4.0
* Back-end:
  + Django 2.0+
  + phpMyAdmin 4.7.4 @ MariaDB 10.1.29
  + Apache 2.4.29

# System Features

## Login

3.1.1 Description and Priority

Because the system does not allow registration, and all functions need to be logged in before allowing use.

The three roles in the system (Member, Worker, Coach) have login privileges.

3.1.2 Stimulus/Response Sequences

* When a not login user accesses the system, a login window is displayed on the page
* The user enters the account password and clicks the login button
* Server verified the account and the password
* If correct: open login session, return to the menu page

3.1.3 Functional Requirements

* Require database Inquiry
* If the username & the password don’t match or don’t exist: return error message, stay on the current page.

## Recharge Quota

3.1.1 Description and Priority

Members can recharge quota through the online payment platform (WeChat Pay and Alipay), The staff can also manually modify the quota for the member.

3.1.2 Stimulus/Response Sequences

* When a not login user accesses the system, a login window is displayed on the page
* The user enters the account password and clicks the login button
* Server verified the account and the password
* If correct: open login session, return to the menu page

3.1.3 Functional Requirements

* Require database Inquiry
* WeChat Pay and Alipay API
* If the payment didn’t finish within 300 seconds. It will automatically halt.

# External Interface Requirements

## User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

## Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

## Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

## Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

# Other Nonfunctional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>