# **CSC 431** <HF-Happy Fitting> Software Requirements Specification (SRS)

**<Yi Rong, Zhenyang Guo, Zipei Chen>**

|  |  |
| --- | --- |
| <Zhenyang Guo> | <Documenter> |
| <Zipei Chen> | <Coding> |
| <Yi Rong> | <Market Analyzing> |

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author(s) | Change Comments |
| 1.0.1 | 2.11 | The whole team | Make the speed for calculating faster and animation more smooth |
| 1.0.2 | 2.21 | The whole team | User can edit their habit for more accurate recommendation |
| 2.0 | 3.9 | The whole team | everything is updated |
| 3.0 | 5.5 | The  Whole  team | Everything is updated |

# Table of Contents

[1. System Requirements 6](#_Toc71150423)

[1.1 Functional Requirements 6](#_Toc71150424)

[1.1.1 Login process 6](#_Toc71150425)

[1.1.1.0 login 6](#_Toc71150426)

[1.1.1.1 reset password 6](#_Toc71150427)

[1.2 Non-Functional Requirements 8](#_Toc71150428)

[1.2.1 clear and organized UI 8](#_Toc71150429)

[1.2.3 large database 8](#_Toc71150430)

[2. System Constraints 10](#_Toc71150431)

[2.1 Tool Constraints 10](#_Toc71150432)

[2.2 Language Constraints 10](#_Toc71150433)

[2.2.1 JAVA 10](#_Toc71150434)

[2.3 Platform Constraints 10](#_Toc71150435)

[2.3.1 Android 10](#_Toc71150436)

[2.4 Hardware Constraints 10](#_Toc71150437)

[2.5 Network Constraints 11](#_Toc71150438)

[2.6 Deployment Constraints 11](#_Toc71150439)

[2.7 Transition & Support Constraints 11](#_Toc71150440)

[2.8 Budget & Schedule Constraints 11](#_Toc71150441)

[2.8.1 Database and software budget constraints 11](#_Toc71150442)

[2.9 Miscellaneous Constraints 12](#_Toc71150443)

[2.9.1 Schedule constraint 12](#_Toc71150444)

[3. Requirements Modeling 13](#_Toc71150445)

[3.1.1 Login process 13](#_Toc71150446)

[3.2 Functional Requirements 14](#_Toc71150447)

[3.2.1 recommend courses 14](#_Toc71150448)

[3.3 Non-Functional Requirements 15](#_Toc71150449)

[3.3.1 larger database 15](#_Toc71150450)

[3.3.2 better algorithms 15](#_Toc71150451)

# Table of Tables

1.1.1 login process 6

1.1.2 reflect actions 7

2.1.1 better ui 8

2.1.2 speed 8

2.1.3 large database 8

2.2.1 Java 10

2.3.1 Android 10

2.8.1 Database and software budget constraints 11

2.9.1 Schedule constraints 12

3.2.1 recommend courses 14

3.3.1 larger database 15

3.3.2 better algorithm 15

# Table of Figures

3.1.1 login process 13

3.1.2 reflect actions 14

### System Requirements

#### Functional Requirements

##### **Login process**

##### **1.1.1.0 login**

|  |  |
| --- | --- |
| Title | <login> |
| Description | <users could login using their accounts> |
| Priority | <0> |
| Precondition(s) | <Users already had their accounts created> |
| Basic Flow | <1.Users enter their accounts and passwords.  2.verifying the entered passwords.  3.if it is right, login successfully.  3.else, error message would be displayed. |
| Postconditions(s) | <successfully jump to the menu page> |
| Use Case Diagram | <3.1.1> |

##### **1.1.1.1 reset password**

|  |  |
| --- | --- |
| Title | <reset password> |
| Description | <system should send verify message to the phone number users used to create their accounts when users forgot their password and want to change it> |
| Priority | <0> |
| Precondition(s) | <Users have already had their accounts created> |
| Basic Flow | <1. User selects “ Forgot your password?” feature from view.  2. User should receive an message of 6digits numbers from system.  3.User would be asked to enter received numbers.  4.If the numbers users entered is correct, then Users would be asked to enter new password.  5.else if the numbers users entered is incorrect, then Users would be asked to reenter the correct numbers.  6.the new password has already been stored into the database.> |
| Postconditions(s) | <User could then use the new password to login their accounts successfully> |
| Use Case Diagram | <3.1.1> |

1.1.1.2 account creating

|  |  |
| --- | --- |
| Title | <account creating> |
| Description | <new users could create their own account> |
| Priority | <0> |
| Precondition(s) | <Users don’t had their accounts created> |
| Basic Flow | <1. User selects “ new account” feature from view.  2. User should enter the required information.  3.their phone numbers would be verified and user would be asked to enter the received 6 digits verify code  4.If the numbers users entered is correct, then new account is created successfully and relate information should be stored into the database.  5.else if the numbers users entered is incorrect, an error message would be displayed.> |
| Postconditions(s) | <User could then use this new account to login their accounts successfully> |
| Use Case Diagram | <3.1.1> |

1 .1. 2 reflect actions

1.1.2.1 reflect user’s action of selecting services

|  |  |
| --- | --- |
| Title | < reflect user’s action of selecting services > |
| Description | <User can select selecting different services from the menu such as “course list”, then user would navigate to the corresponding view> |
| Priority | <0> |
| Precondition(s) | <User should have already logged in> |
| Basic Flow | <1. User select the services from the menu  2. User would navigate to a new view  > |
| Postconditions(s) | <the new view is the corresponding view> |
| Use Case Diagram | <3.1.2> |

1.1.2.2 reflect user’s action of taking courses

|  |  |
| --- | --- |
| Title | < reflect user’s action of taking courses > |
| Description | <User can select different courses from the course list and then start their fitting> |
| Priority | <0> |
| Precondition(s) | <User should have already navigated to the course list view> |
| Basic Flow | <1. User select the course from the course list  2. User would navigate to a new view of according course page  3.User could start their courses.  > |
| Postconditions(s) | <the new view is the corresponding view> |
| Use Case Diagram | <3.1.2> |

1 .1. 2. 3 reflect actions of posting comments on blogs

|  |  |
| --- | --- |
| Title | < reflect actions of posting comments on blogs > |
| Description | <User should be able to post their comments> |
| Priority | <3> |
| Precondition(s) | <There is a comments bar available in the view> |
| Basic Flow | < 1. User enter words in the comments bar  2. User selects “Done” feature to post their comments  > |
| Postconditions(s) | <User’s comments should be posted successfully and available for other users to view > |
| Use Case Diagram | <3.1.2> |

1 .1. 3 User information tracking

|  |  |
| --- | --- |
| Title | <User information tracking> |
| Description | <After User choose the service of “fitness courses” from the menu and start one of these courses(each courses is a series of fitness tutorial videos), System would automatically tacking the User’s exercise length, Calorie consumption based on the lessons user took.> |
| Priority | <3> |
| Precondition(s) | < User have started one of the online lessons > |
| Basic Flow | <1. tracking the exercise length after users start lessons  2. computing the calorie consumptions based on the type physical activities.  3.store these information in the database  > |
| Postconditions(s) | <User should be able to see these information in their user page> |
| Use Case Diagram | <3.1.2> |

#### Non-Functional Requirements

##### clear and organized UI

|  |  |
| --- | --- |
| Title | <clear and organized UI> |
| Description | <clear and organized UI in order to make users feel comfortable and easier to use this app> |
| Priority | <3> |
| Applicable FR(s) | <organized Menu for user to choose> |

1.2.2. Speed

|  |  |
| --- | --- |
| Title | <Speed> |
| Description | <Accelerate the overall speed to make users using this app smoothly> |
| Priority | <2> |
| Applicable FR(s) | <overall quality of this app, including the faster initialization process> |

##### 1.2.3 large database

|  |  |
| --- | --- |
| Title | <large database> |
| Description | <large database could let happy fitting be able to meet a large number of users> |
| Priority | <0> |
| Applicable FR(s) | < large database used to store courses, user’s information regarding their basic personal information of name, age, weight, their daily calorie lost, total exercise time and blog information > |

### System Constraints

#### Tool Constraints

none

#### Language Constraints

##### JAVA

|  |  |
| --- | --- |
| Title | <JAVA> |
| Description | <this is an android based application and use Java for programming.> |
| Priority | <0> |

#### Platform Constraints

##### Android

|  |  |
| --- | --- |
| Title | <Android> |
| Description | <This application could only be running on android based devises.> |
| Priority | <0> |

#### Hardware Constraints

none

#### Network Constraints

none

#### Deployment Constraints

none

#### Transition & Support Constraints

none

#### Budget & Schedule Constraints

##### **Database and software budget constraints**

|  |  |
| --- | --- |
| Title | <Need money for building database and propagate the software> |
| Description | <The propagate needs the money and if no one knows the software, it’s hard to make money.  The database is necessary for completing the software> |
| Priority | <0> |

#### Miscellaneous Constraints

##### Schedule constraint

|  |  |
| --- | --- |
| Title | <deadline of this project> |
| Description | <There is a time limit in designing this application> |
| Priority | <0> |

### Requirements Modeling

##### Login process

Diagram

Description automatically generated

3. 1. 2 reflect users’ actions

Chart, diagram, bubble chart

Description automatically generated

#### Functional Requirements

##### recommend courses

|  |  |
| --- | --- |
| Title | < recommend courses> |
| Description | <Collect what user like do most and save in a database which also visible and changeable by the user, the software can automatically recommend the courses similar to the user’s habbits> |
| Priority | <3> |
| Precondition(s) | <User have filled the personal information regarding their preference> |
| Postconditions(s) | <System would recommend courses best fit user’s preference > |
| Use Case Diagram | <3.1.2> |

#### Non-Functional Requirements

##### larger database

|  |  |
| --- | --- |
| Title | <larger database> |
| Description | <larger database to meet the requirement of storing more and more information of both users and courses> |
| Priority | <0> |
| Applicable FR(s) | <1.Collecting user’s using habits and put them into the machine learning algorithms to predict their preference. 2. Storing more courses, more user’s information regarding their basic personal information of name, age, weight, their daily calorie lost, total exercise time and blog information > |

##### **better algorithms**

|  |  |
| --- | --- |
| Title | <better algorithms> |
| Description | <using machine learning algorithm do data analysis and prediction> |
| Priority | <3> |
| Applicable FR(s) | < recommend courses based on user’s preference> |