# **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> |
| <Nobody> | <Nothing> |

# 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 |
|  |  |  |  |

# Table of Contents

[1. System Requirements 6](#_Toc412649646)

[1.1 Functional Requirements 6](#_Toc412649647)

[1.1.1 Requirement Title 6](#_Toc412649648)

[1.2 Non-Functional Requirements 6](#_Toc412649649)

[1.2.1 Requirement Title 6](#_Toc412649650)

[2. System Constraints 7](#_Toc412649651)

[2.1 Tool Constraints 7](#_Toc412649652)

[2.1.1 Requirement Title 7](#_Toc412649653)

[2.2 Language Constraints 7](#_Toc412649654)

[2.2.1 Requirement Title 7](#_Toc412649655)

[2.3 Platform Constraints 7](#_Toc412649656)

[2.3.1 Requirement Title 7](#_Toc412649657)

[2.4 Hardware Constraints 7](#_Toc412649658)

[2.4.1 Requirement Title 7](#_Toc412649659)

[2.5 Network Constraints 7](#_Toc412649660)

[2.5.1 Requirement Title 8](#_Toc412649661)

[2.6 Deployment Constraints 8](#_Toc412649662)

[2.6.1 Requirement Title 8](#_Toc412649663)

[2.7 Transition & Support Constraints 8](#_Toc412649664)

[2.7.1 Requirement Title 8](#_Toc412649665)

[2.8 Budget & Schedule Constraints 8](#_Toc412649666)

[2.8.1 Requirement Title 8](#_Toc412649667)

[2.9 Miscellaneous Constraints 8](#_Toc412649668)

[2.9.1 Requirement Title 8](#_Toc412649669)

[3. Requirements Modeling 10](#_Toc412649670)

[3.1.1 Requirement Title 10](#_Toc412649671)

[4. Evolutionary Requirements 11](#_Toc412649672)

[4.1 Functional Requirements 11](#_Toc412649673)

[4.1.1 Requirement Title 11](#_Toc412649674)

[4.2 Non-Functional Requirements 11](#_Toc412649675)

[4.2.1 Requirement Title 11](#_Toc412649676)

# Table of Tables

# Table of Figures

<Generate table here>

### System Requirements

#### Functional Requirements

##### phone number verifying

|  |  |
| --- | --- |
| Title | <phone number verifying> |
| 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. 2 **Community**

|  |  |
| --- | --- |
| Title | <community> |
| 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 **menu**

|  |  |
| --- | --- |
| Title | < menu> |
| Description | <User can select selecting different services from the menu such as “fitness courses”, 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 | <3.1.3> |

1 .1. 4 **healthy tips**

|  |  |
| --- | --- |
| Title | < healthy tips> |
| Description | <During the initialization loading, there would appear some healthy tips> |
| Priority | <5> |
| Precondition(s) | <the system is in the initialization and loading process> |
| Basic Flow | <1. Users opened Happy Fitting app  2. System is in the initialization process  3. pictures of healthy tips would be showed.  > |
| Postconditions(s) | <System finished initialization and User would navigate to the view of login> |
| Use Case Diagram | <3.1.6> |

1 .1. 5 **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 | <3.1.4> |

#### 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 Scientific and high-performance Algorithm

|  |  |
| --- | --- |
| Title | < scientific and high-performance algorithm> |
| Description | < Including scientific algorithm like calculating Calories to provide users accurate tracking result.> |
| Priority | <2> |
| Applicable FR(s) | <tracking user’s physical activity data and calculating the calorie consumption> |

### System Constraints

#### Tool Constraints

< List all tool constraints in the following example format >

none

#### Language Constraints

< List all language constraints in the following example format >

##### JAVA

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

#### Platform Constraints

< List all platform constraints in the following example format >

##### Android

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

#### Hardware Constraints

< List all hardware constraints in the following example format >

none

#### Network Constraints

< List all network constraints in the following example format >

none

#### Deployment Constraints

< List all deployment constraints in the following example format >

none

#### Transition & Support Constraints

< List all transition & support constraints in the following example format >

none

#### Budget & Schedule Constraints

< Cost for server, database, and propagate >

##### Need money for building database and propagate the software

|  |  |
| --- | --- |
| 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

< List all miscellaneous constraints in the following example format >

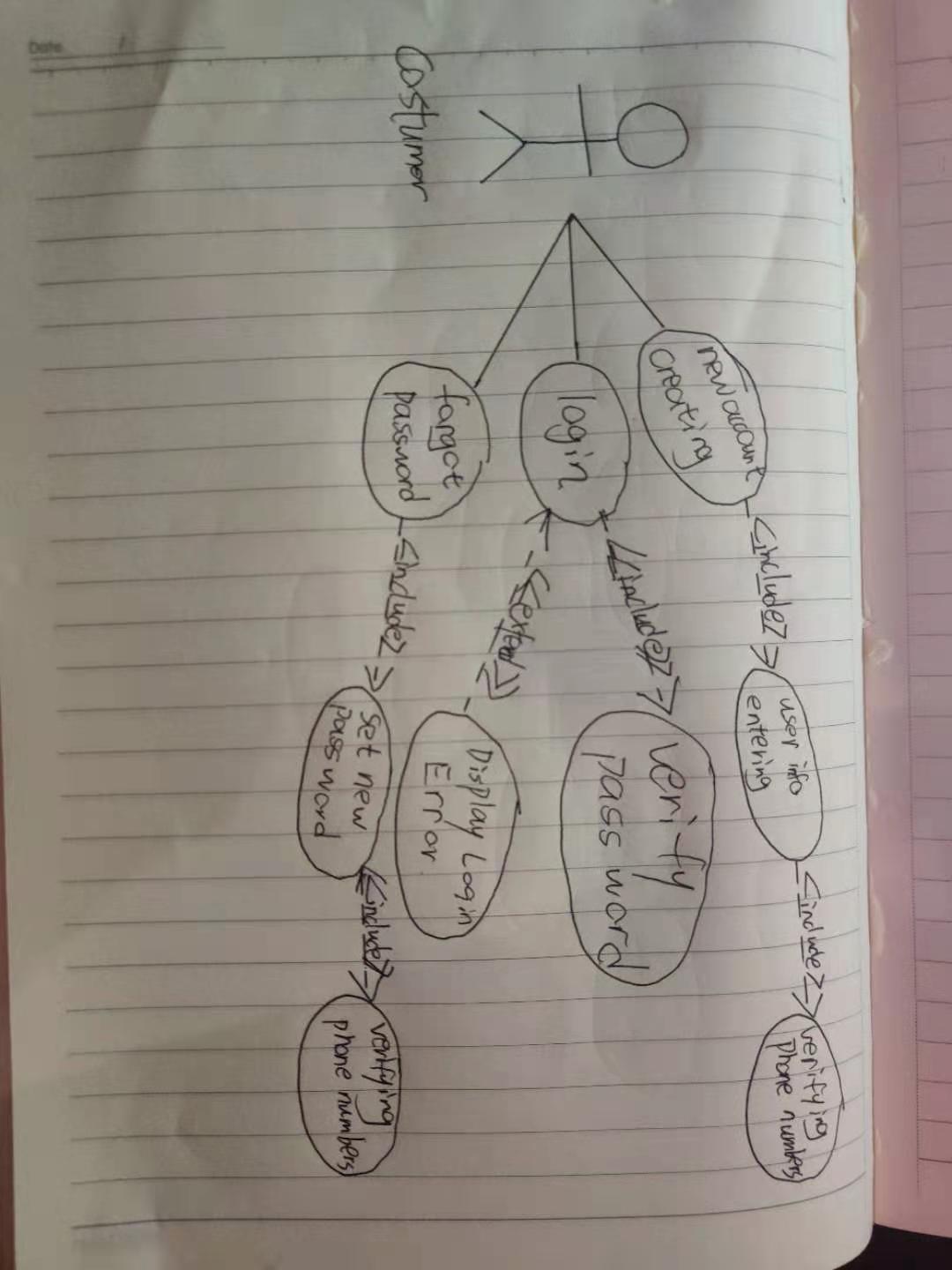
##### deadline of this project

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

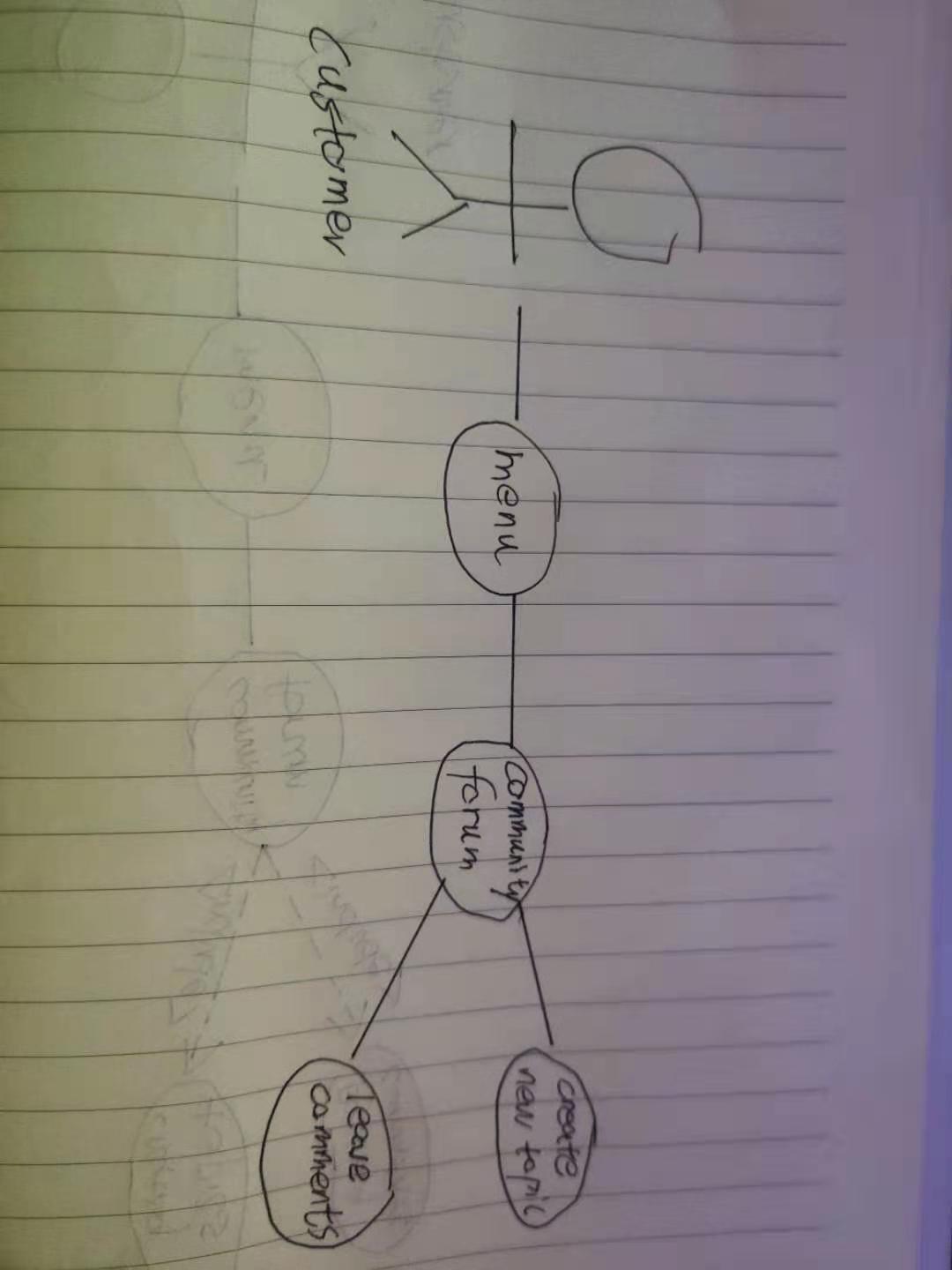
### Requirements Modeling

< List all Use-case diagrams for the functional requirements in the following format>

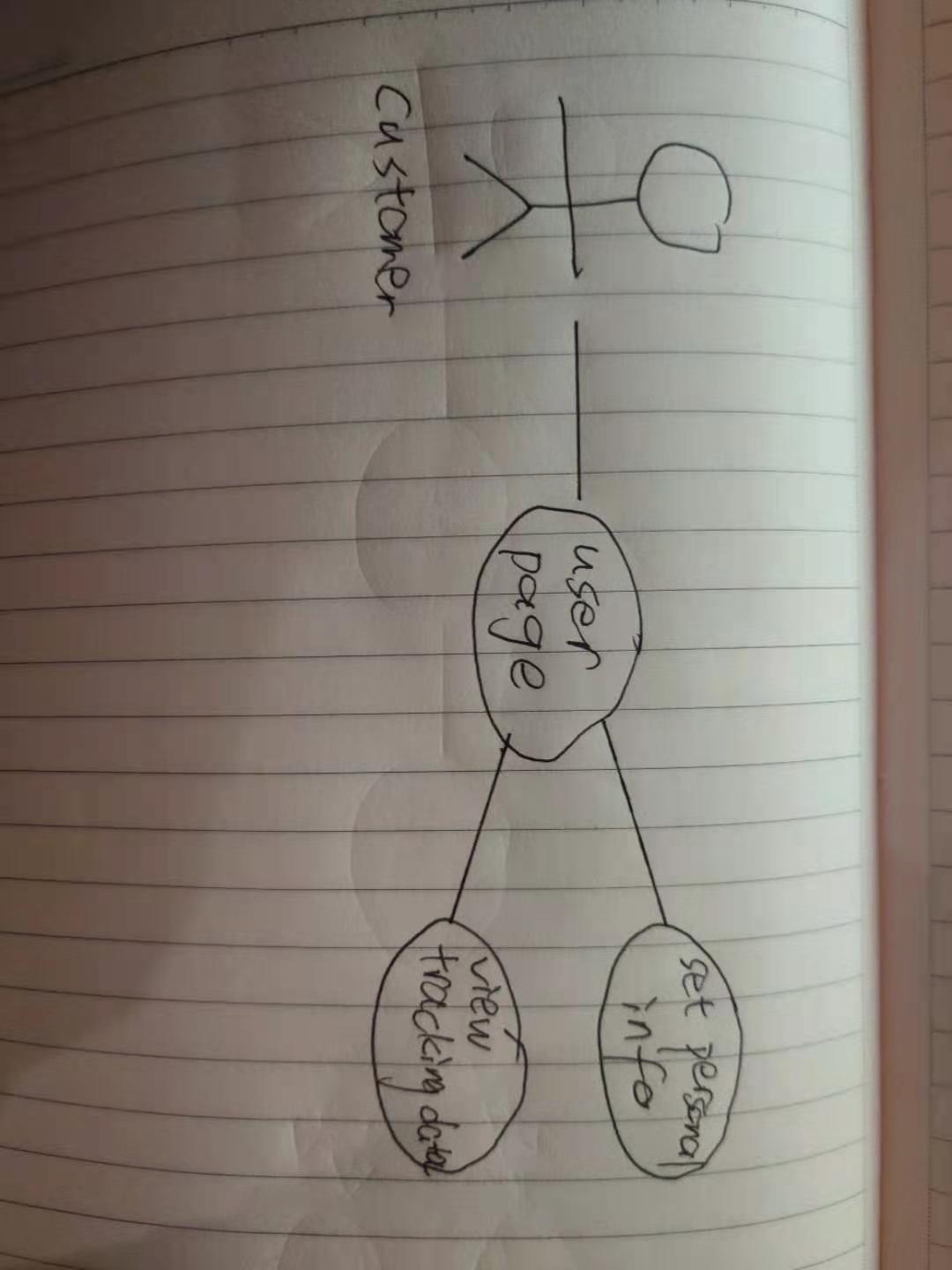
##### Login process



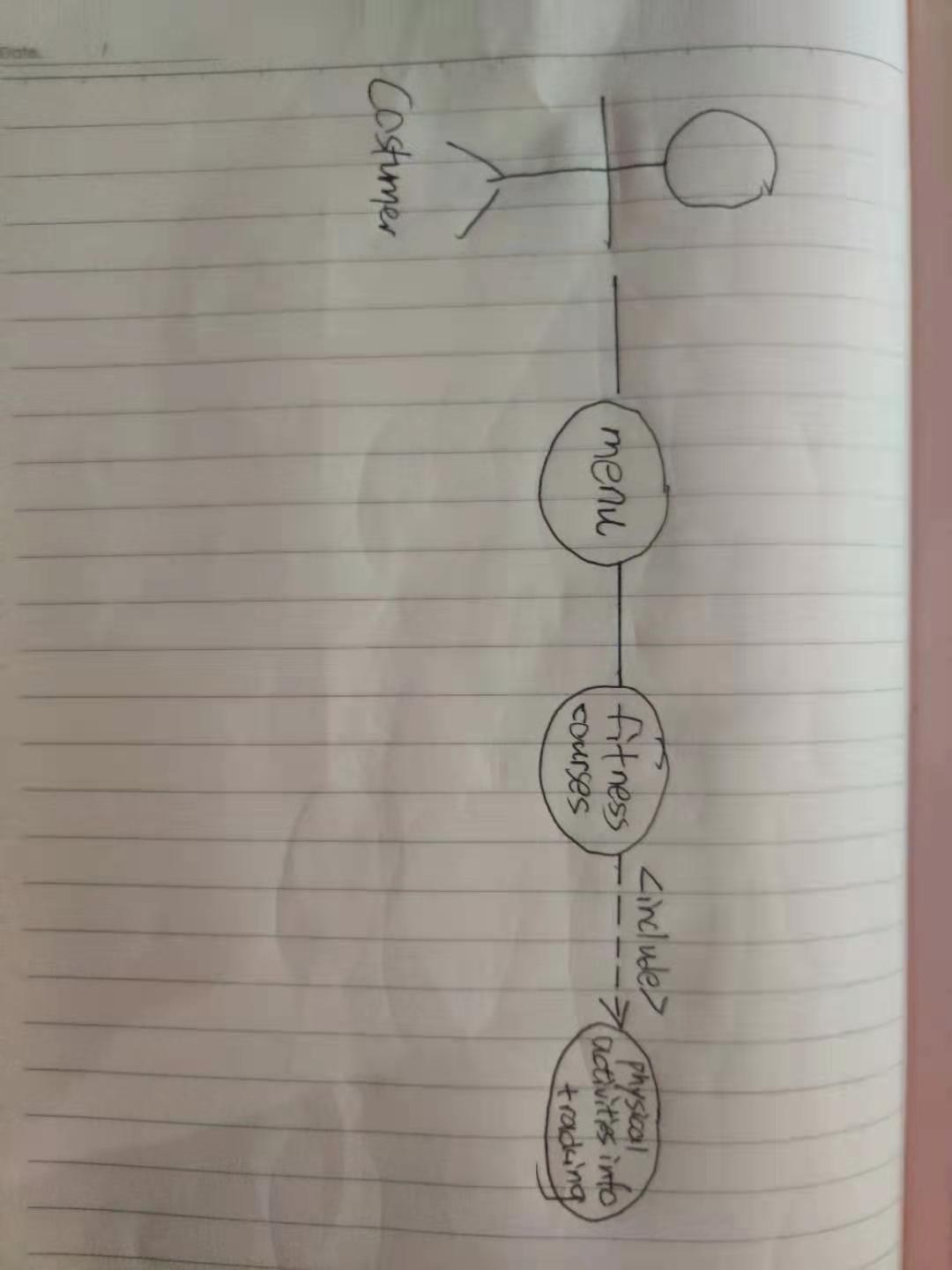
3. 1. 2 community forum



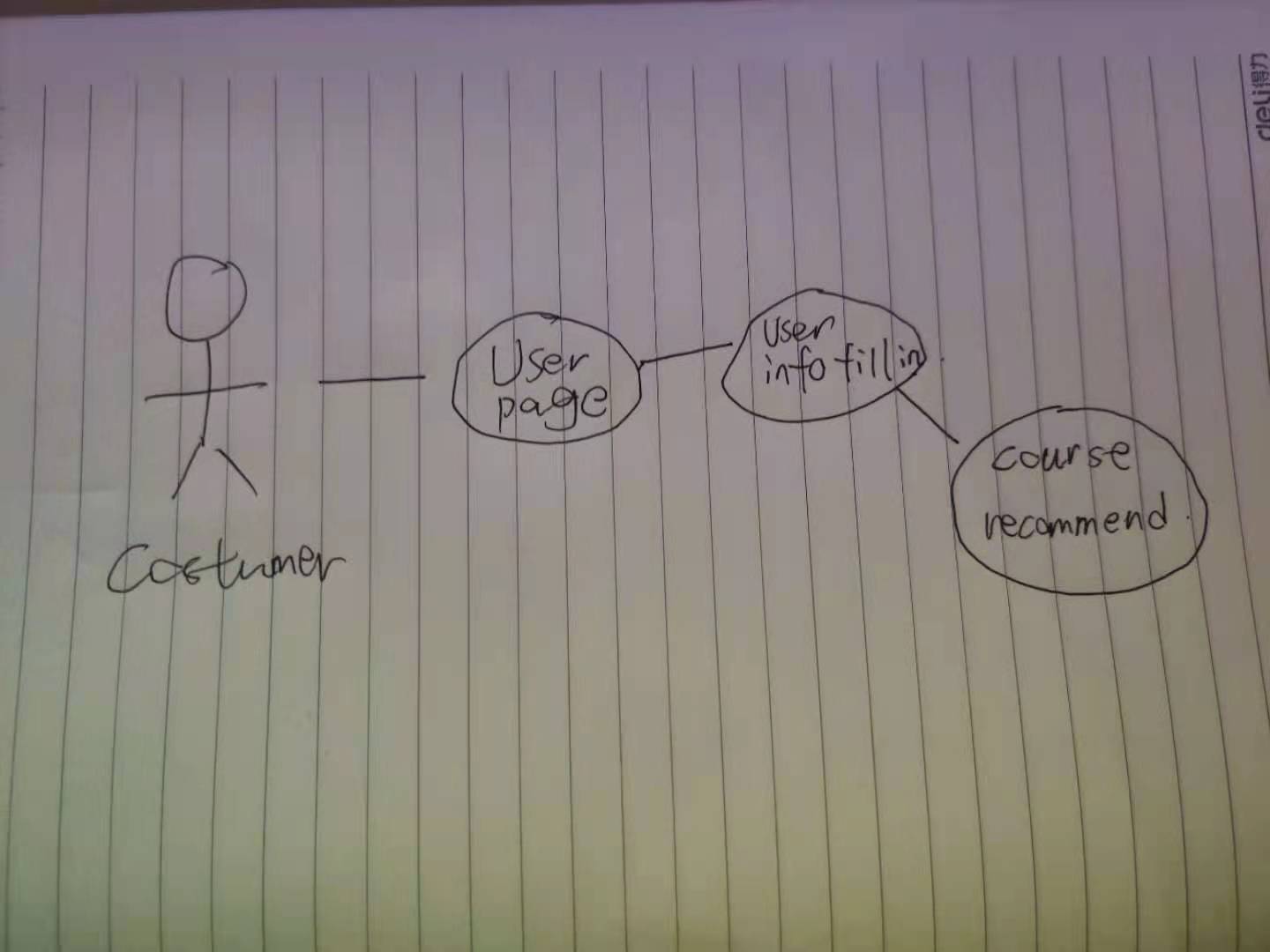
3. 1. 3 UserPage

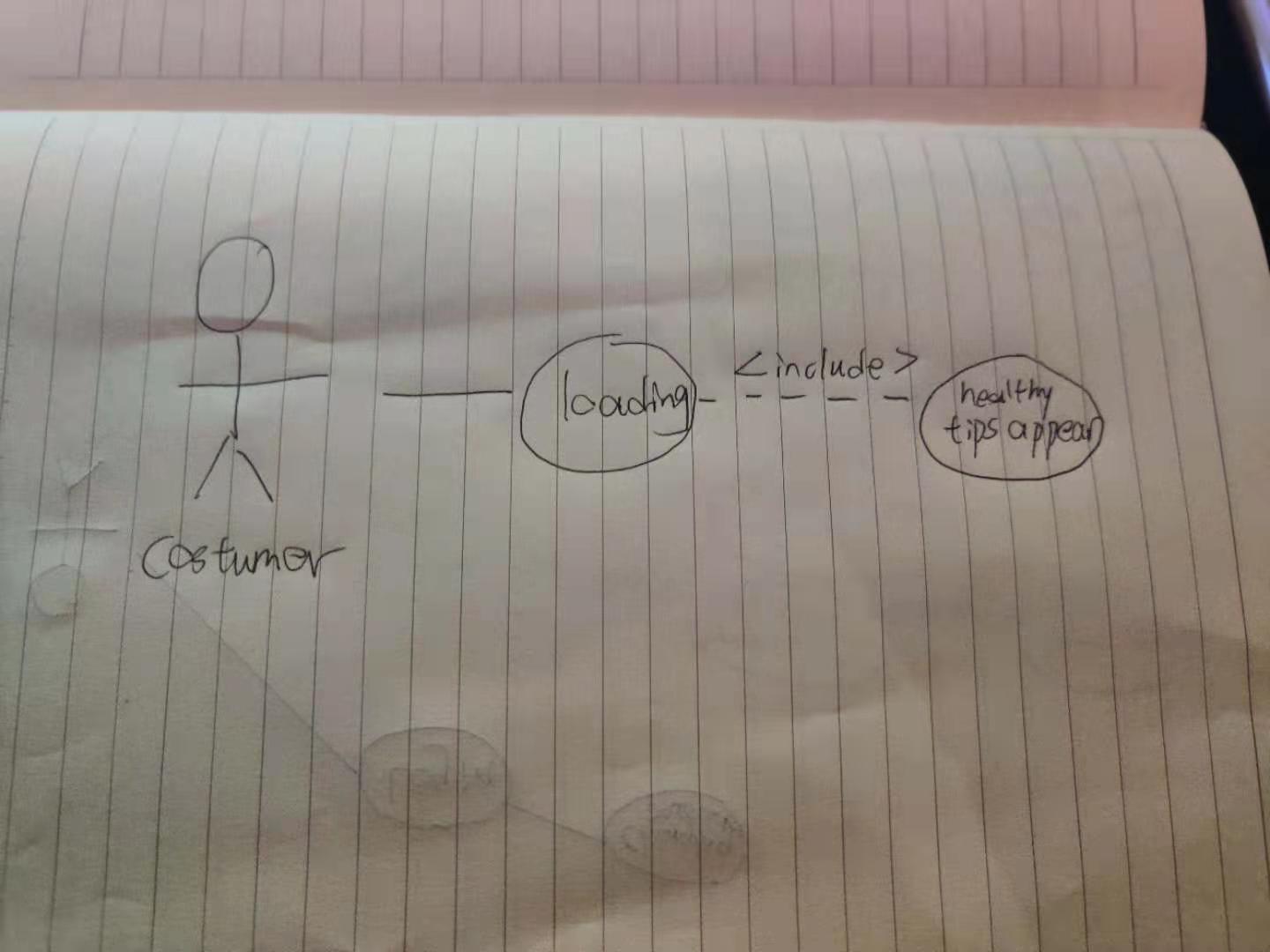


3. 1. 4 Fitness courses & data tracking



3. 1. 5 courses recommend



3. 1. 6 Healthy tips appear during loading

### Evolutionary Requirements

#### Functional Requirements

< Collecting user’s using habits and create the database for each user>

##### 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 habbit> |
| 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 | <3.1.6> |

#### Non-Functional Requirements

< The user can see the product they need most when they are using the entertainment function of the software >

##### larger database

|  |  |
| --- | --- |
| Title | <larger database> |
| Description | <larger database to store more and more info> |
| Priority | <0> |
| Applicable FR(s) | <Collecting user’s using habits> |

##### **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> |