Software Requirements Specification

for

Brew Day!

**Version 1.7 approved**

**Prepared by**

**Guo Rui 1630013011**

**Ji Jia 1630003023**

**Xie Qizhou 1630003056**

**Chen Mingxuan 1630003002**

**Atanasoff**

**12:34 27/03/2019**

Table of Contents

Table of Contents ii

Revision History iii

1. Introduction (Written by Ji Jia) 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 (Written by Xie Qizhou) 2

2.1 Product Perspective 2

2.2 Product Features 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 3

2.5 Design and Implementation Constraints (Written by Chen Mingxuan) 3

2.6 User Documentation 3

2.7 Assumptions and Dependencies 3

3. System Features 3

3.1 Maintain Recipes 3

3.1.1 Description and Priority (Written by Guo Rui) 3

3.1.2 Stimulus/Response Sequences (Written by Xie Qizhou) 4

3.1.3 Functional Requirements (Written by Chen Mingxuan) 4

3.2 Maintain ingredients 4

3.2.1 Description and Priority (Written by Guo Rui) 4

3.2.2 Stimulus/Response Sequences (Written by Xie Qizhou) 5

3.2.3 Functional Requirements (Written by Chen Mingxuan) 5

3.3 Maintain equipment information 6

3.3.1 Description and Priority (Written by Guo Rui) 6

3.3.2 Stimulus/Response Sequences (Written by Xie Qizhou) 6

3.3.3 Functional Requirements (Written by Chen Mingxuan) 6

3.4 Write note 7

3.4.1 Description and Priority (Written by Guo Rui) 7

3.4.2 Stimulus/Response Sequences (Written by Xie Qizhou) 7

3.4.3 Functional Requirements (Written by Chen Mingxuan) 7

3.5 Recommend recipes 7

3.5.1 Description and Priority (Written by Guo Rui) 7

3.5.2 Stimulus/Response Sequences (Written by Xie Qizhou) 8

3.5.3 Functional Requirements (Written by Chen Mingxuan) 8

4. External Interface Requirements (Written by Guo Rui) 9

4.1 User Interfaces (Written by Chen Mingxuan & Ji jia) 9

4.2 Hardware Interfaces 18

4.3 Software Interfaces 18

4.4 Communications Interfaces 18

5. Other Nonfunctional Requirements (Written by Chen Mingxuan) 18

5.1 Performance Requirements 18

5.2 Safety Requirements 19

5.3 Security Requirements 19

5.4 Software Quality Attributes 19

6. Other Requirements (Written by Guo Rui) 20

Appendix A: Glossary 20

Appendix B: Analysis Models(Written by Ji Jia, Guo Rui, Xie Qizhou) 20

Appendix C: Issues List 21

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Ji Jia, Guo Rui, Xie Qizhou, Chen Mingxuan | 05/03/2019 | The first version of SRS. All the members did it together. | Initial version |
| Ji Jia, Guo Rui, Xie Qizhou, Chen Mingxuan | 12/03/2019 | Update the use case diagram in section 2.2. | Version 1.1 |
| Guo RUI, Ji Jia | 12/03/2019 | Correct errors and add references in section 1. | Version 1.2 |
| Xie Qizhou | 12/03/2019 | Draw the transition diagrams in section 3. | Version 1.3 |
| Chen Mingxuan | 13/03/2019 | Draw the User Interface diagrams in section 4. | Version 1.4 |
| Guo Rui, Xie Qizhou | 19/03/2019 | Add new system features, update transition diagrams in section 3. | Version 1.5 |
| Ji Jia, Guo Rui, Xie Qizhou, Chen Mingxuan | 20/03/2019 | Update User Interface diagrams in section 4. | Version 1.6 |
| Ji Jia, Guo Rui, Xie Qizhou, Chen Mingxuan | 27/03/2019 | Correct problems in use case diagram, transition diagrams and UI.  Draw the sequence diagram and class diagram. | Version 1.7 |

# Introduction (Written by Ji Jia)

## Purpose

The purpose of this document is to build an application, Brew Day, which allows home brewers to maintain an organized database of their beer recipes. The application allows users to create, store and modify recipes, and later on delete them, if the user wishes to do so. The application is intended for "all-grain" brewers only, and thus all recipes are for this kind of brews.

Besides the actual recipes, the application must maintain recipe instances; these instances can be accompanied by notes to refer to issues that may affect the resulting beer and the brewers would like to keep logging. A particular kind of note is the tasting notes, which allow brewers to keep track of opinions on a beer from a particular brew.

Also, the application maintains the list of available ingredients. This allows brewers to be notified about missing ingredients for the next brew.

## Document Conventions

This document follows MLA Format. Bold-faced text has been used to emphasize section and sub-section headings. Highlighting is to point out words in the glossary and italicized text is used to label and recognize diagrams.

## Intended Audience and Reading Suggestions

This document is to be read by the development team members, the project managers, the testers and documentation writers. The SRS has been organized approximately familiar with the SRS.A

## Project Scope

The project implements the features, i.e., creation of recipe instances, supporting for notes on brews, keeping tracks of available ingredients, and the mandatory feature “What should I brew today?” Developers may choose to allow ingredients availability from brews information automatically.

## References

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

[2]. Simon Bennett; Steve McRobb; Ray Farmer, “Object Oriented Systems Analysis and Design Using UML”, (4th Edition), McGraw Hill, 2010.

# Overall Description (Written by Xie Qizhou)

## Product Perspective

This product is basically a small application with only a few functions in it. Meanwhile, this is an application which is new and self-contained product.

## Product Features

## User Classes and Characteristics

User classes:

Member:

* Record his own recipes.
* Share his recipes to other members.
* View on other members' shared recipes.
* Manage his own recipes (modify, delete etc.)
* Make tasting notes.

Staff:

* Manage the memberships of this application.
* Maintaining a list of usable recipes.

## Operating Environment

This system will be and deploy and test on Windows 10.

Page browsing using Chrome (64-bit).

Other environment issue is to be determined.

## Design and Implementation Constraints (Written by Chen Mingxuan)

Java will be used to build this software, our database will based on MySQL, all the programming process will push to the Github.

## User Documentation

User's Guide

Contact number and E-mail address

## Assumptions and Dependencies

* Java SE 11.0.2
* Other factors are still underdetermined.

# System Features

## Maintain Recipes

### Description and Priority (Written by Guo Rui)

**Maintain Recipes (highest priority)**

When user selects “Maintain Recipes”

System will ask: Add, Delete, or Update.

1. If user selects add, the system will allow user to create a new recipes.

User should fill in all the ingredients table and actual amounts, and then click submit button.

System will check if the user’s input is correct, if it is, then upload the new recipes to database, print “Successful!” message, and create a new record of viewing. If not, the system will return an error message.

1. If user select delete, the system will allow user to delete previous recipes.

User should select which one, and click delete button.

The system will ask “Do you want to delete this recipes?”

The user should answer Yes or No (if user don’t want to delete).

If user click “Yes”, then the system will delete this recipes from database, and return a successful message.

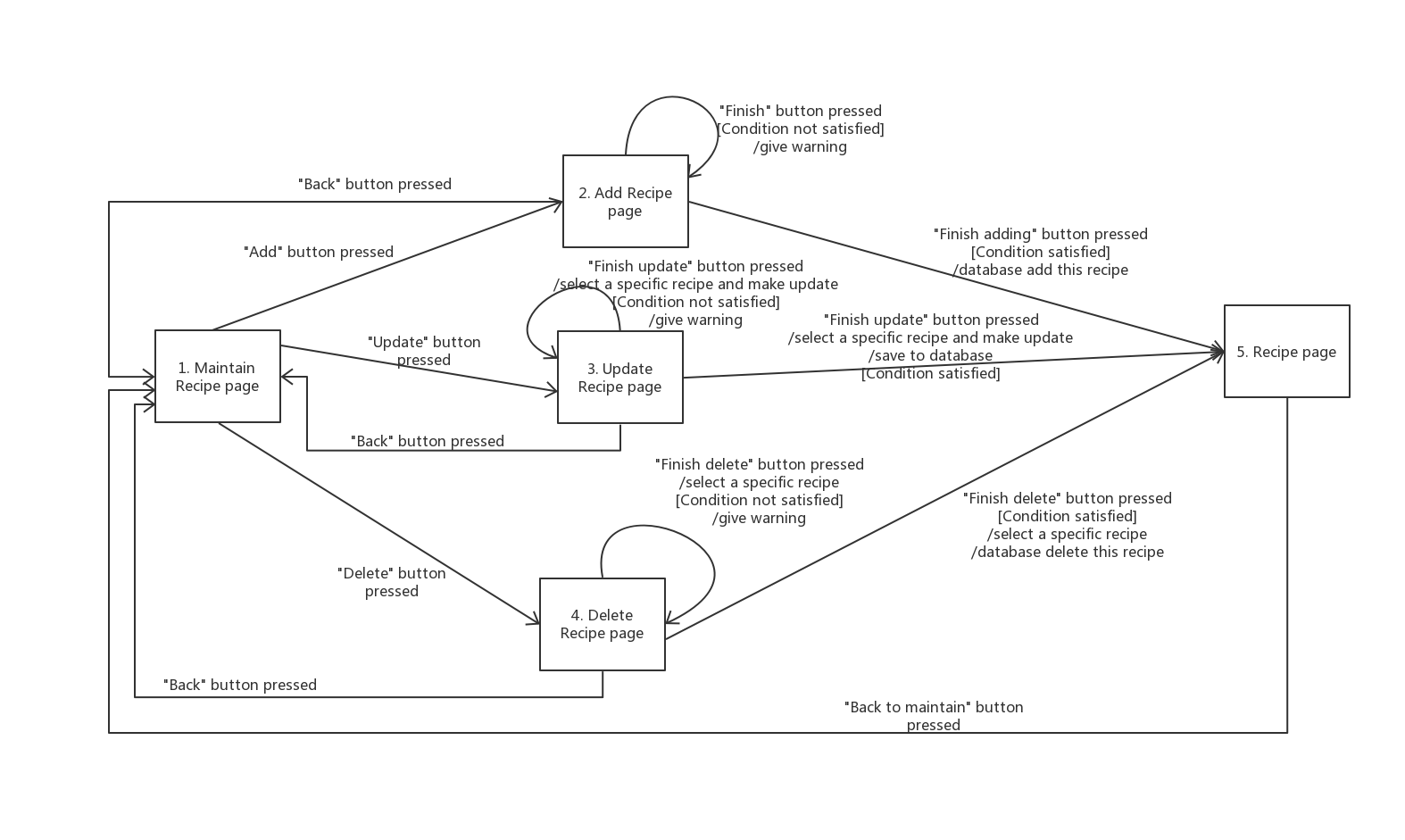
1. If user select update, the system will allow user to update current recipes.

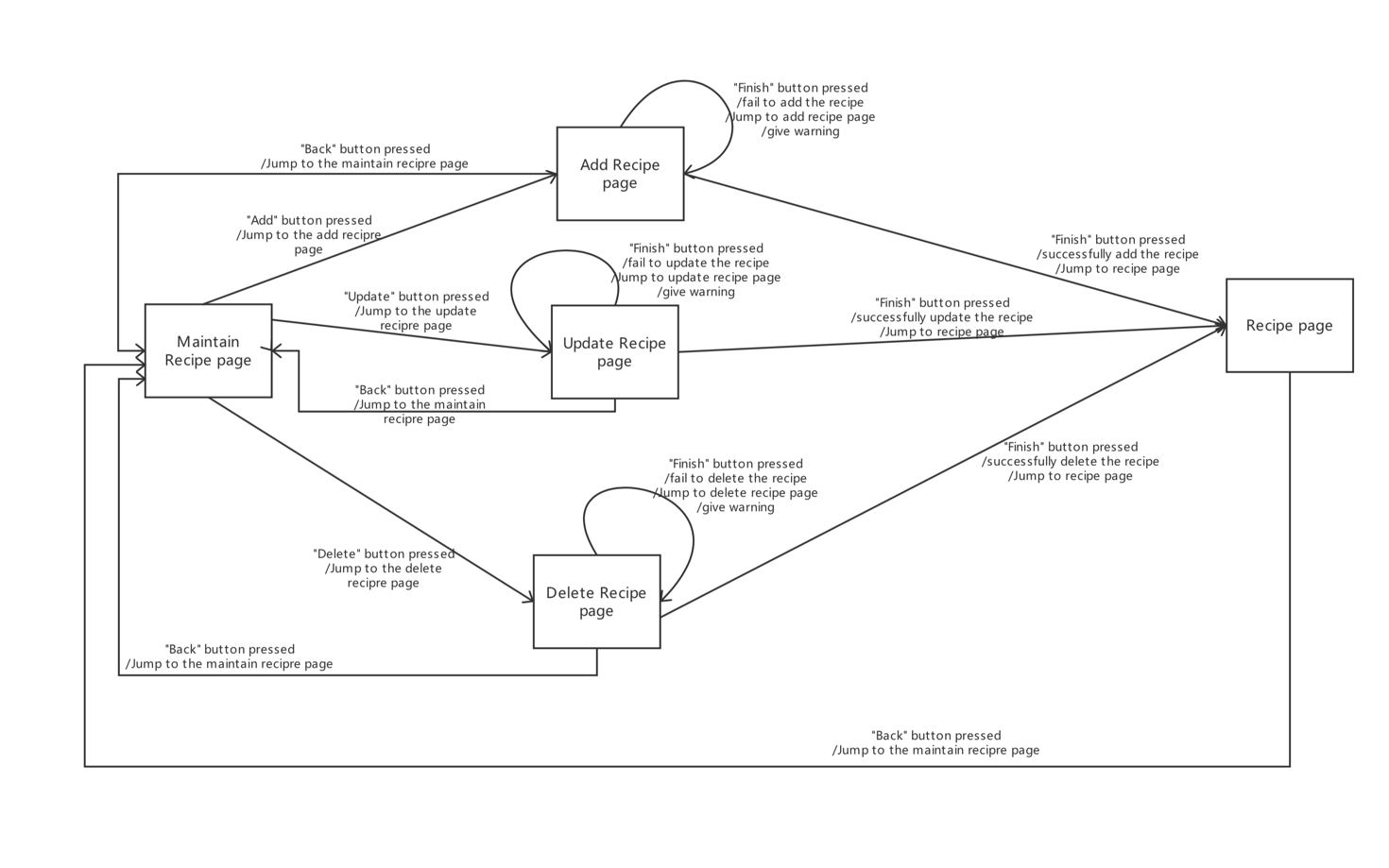
User should change this recipes, and click “finish” button.

The system will check if the user’s input is correct, if it is, then update this recipes from database, and return a successful message. If not, the system will return an error message.

### Stimulus/Response Sequences (Written by Xie Qizhou)

Transition diagrams for recipe page.





### Functional Requirements (Written by Chen Mingxuan)

Most pages require a response, especially after clicking "Finish," so in case of a network problem or a data upload error, we will stop the response and return to the page that initiated the response, warning the user that something is wrong by popover.

## Maintain ingredients

### Description and Priority (Written by Guo Rui)

**Maintain ingredients (highest priority)**

When user selects “Maintain ingredients”.

System will ask Add or Subtract.

1. If user selects add, the system will ask how much liters beer for this recipe.

User should fill in the ingredients table and actual amounts, and then click submit button.

System will check if the user’s input is correct, if it is, then compute the percentage for each ingredient, and show out in the table. And then upload the new ingredients to database, print “Successful!” message, and create a new record of viewing. If not, the system will return an error message.

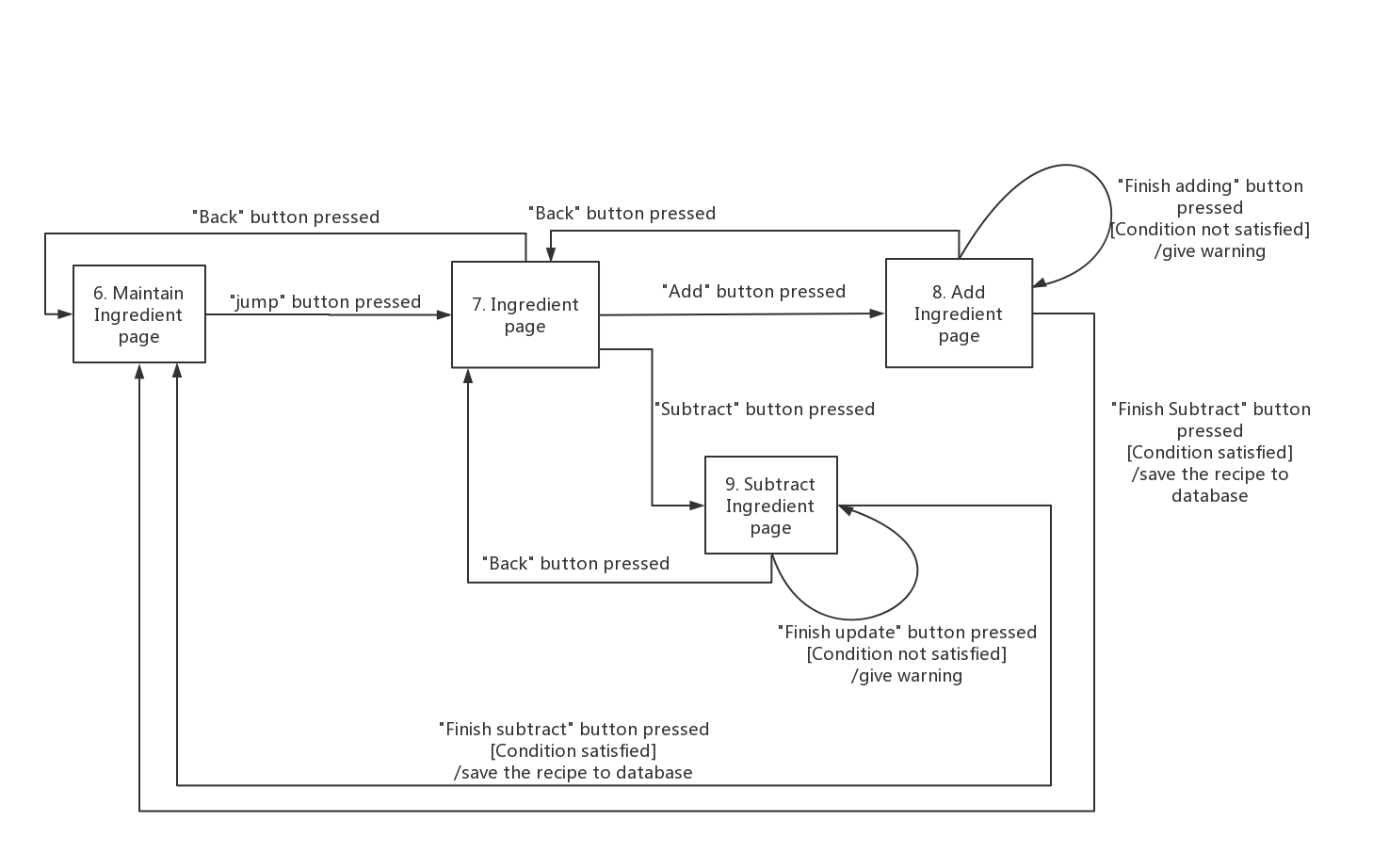
1. If user select subtract, the system will allow to reduce current ingredients.

User should change the ingredients or beer size, and click “finish” button.

The system will check if the user’s input is correct, if it is, then update this ingredients from database, and return a successful message. If not, the system will return an error message.

### Stimulus/Response Sequences (Written by Xie Qizhou)

Transition diagrams for ingredient page.



### Functional Requirements (Written by Chen Mingxuan)

Most pages require a response, especially after clicking "Finish," so in case of a network problem or a data upload error, we will stop the response and return to the page that initiated the response, warning the user that something is wrong by popover.

## Maintain equipment information

### Description and Priority (Written by Guo Rui)

**Maintain equipment information (highest priority)**

When user selects “Maintain equipment information”.

System will ask Add or Update.

1. If user selects add, the system will allow user to add equipment.

User should fill in the equipment items table and actual amount of liters, and then click submit button.

System will check whether the user’s input is correct, if it is, then upload the new equipment table to database, print “Successful!” message, and create a new record of viewing for equipment information. If not, the system will return an error message.

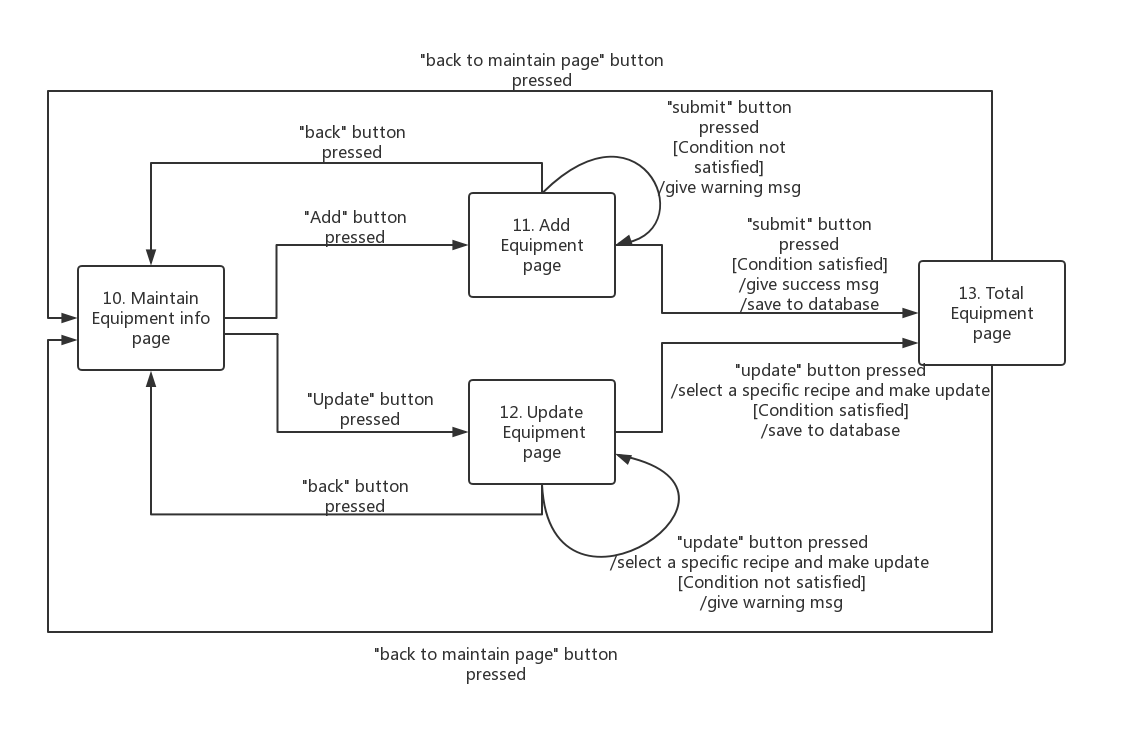
1. If user select update, the system will allow user to update current equipment table.

User should change the equipment table with correct liters, and click “finish” button.

The system will check whether the user’s input is correct, if it is, then update this equipment table from database, and return a successful message. If not, the system will return an error message.

### Stimulus/Response Sequences (Written by Xie Qizhou)

Transition diagrams for maintaining equipment information page.



### 3.3.3 Functional Requirements (Written by Chen Mingxuan)

Most pages require a response, especially after clicking "Finish," so in case of a network problem or a data upload error, we will stop the response and return to the page that initiated the response, warning the user that something is wrong by popover.

## Write note

### Description and Priority (Written by Guo Rui)

For each recipe, if you slide right for the item, then the system will show add note function.

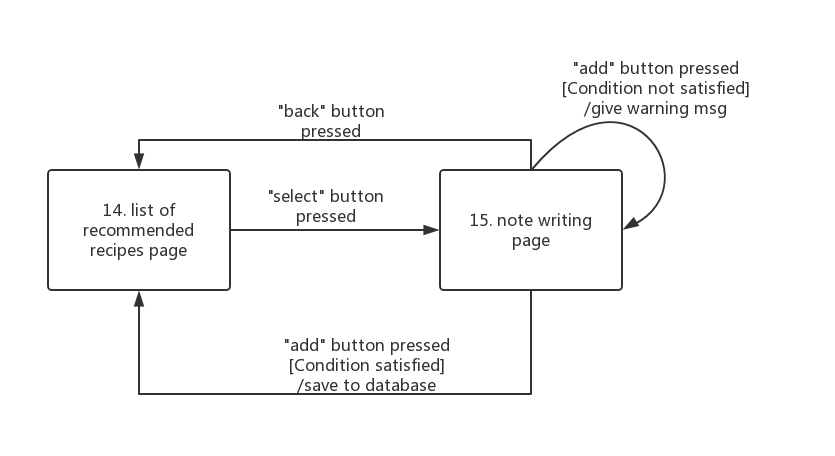
**Write note (middle priority)**

User can click add note button for each recipe in the table. If user click add, user should type some words or sentences in English, and then click submit button.

The system will add this note to database, and return a successful message.

### Stimulus/Response Sequences (Written by Xie Qizhou)

Transition diagrams for writing note page.



### Functional Requirements (Written by Chen Mingxuan)

When “add” button pressed, the database must check if the note successfully added into it or not, when it fail to add, it will give a warning to user.

## Recommend recipes

### Description and Priority (Written by Guo Rui)

**Recommend Recipes (highest priority)**

When user selects “Recommend recipes”

System will ask “How much liters beer do you want?”

The user must input the number of liters and click submit button.

First, the system will check if the number of input is positive value. If not, system will print an error message.

Second, the system will check if the current equipment capacity is bigger than user’s input.

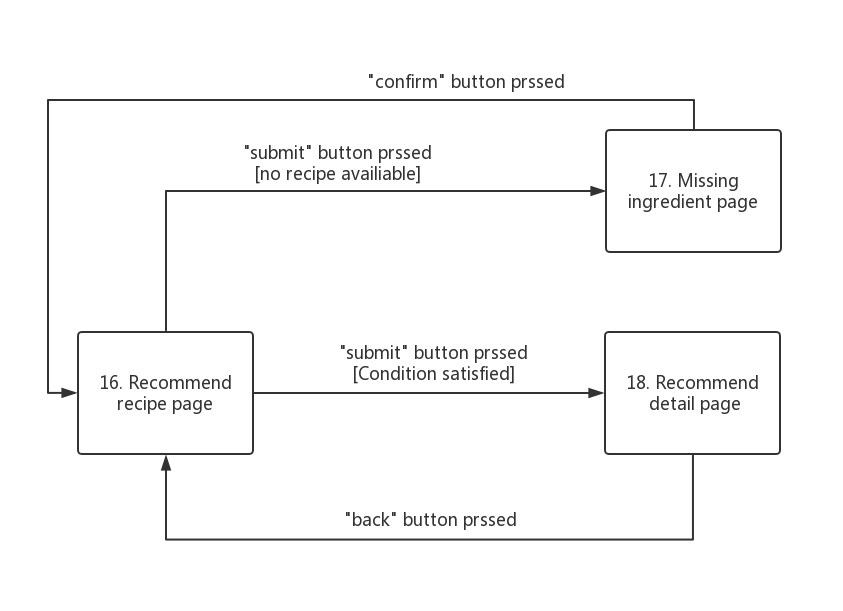
If it is, then the system will calculate and compare each recipes, whether current ingredients is enough for this recipe. If it is enough, then this recipe will be recommend to user. The system will do a loop to check all the recipes. Then show all the recommend recipes for user to choose.

User should select one, and click finish button, then the system will calculate the assumption of ingredients, and call the subtract ingredients function to maintain the ingredients.

If not, then the system will print “The current equipment capacity is not enough!” and show how much ingredients needed for this recipe. User can change the equipment or change the number of batch beer and try again.

### Stimulus/Response Sequences (Written by Xie Qizhou)

Transition diagrams for recommend page.



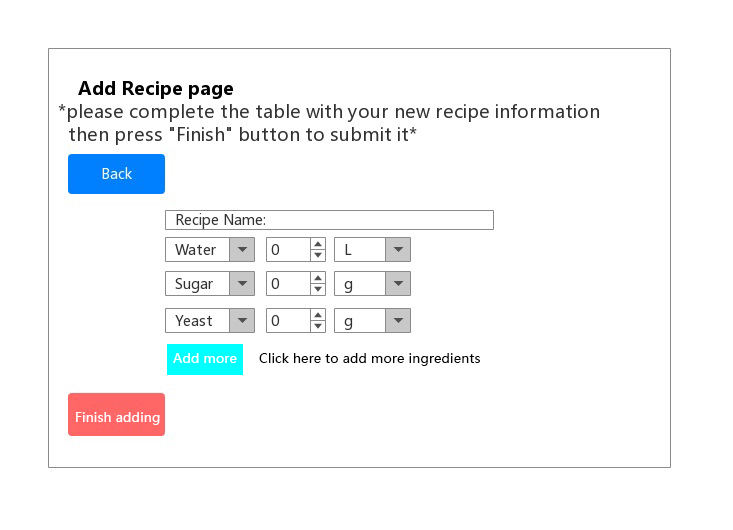
### Functional Requirements (Written by Chen Mingxuan)

Most pages require a response, especially after clicking "submit", so in case of a network problem or a data upload error, we will stop the response and return to the page that initiated the response, warning the user that something is wrong by popover.

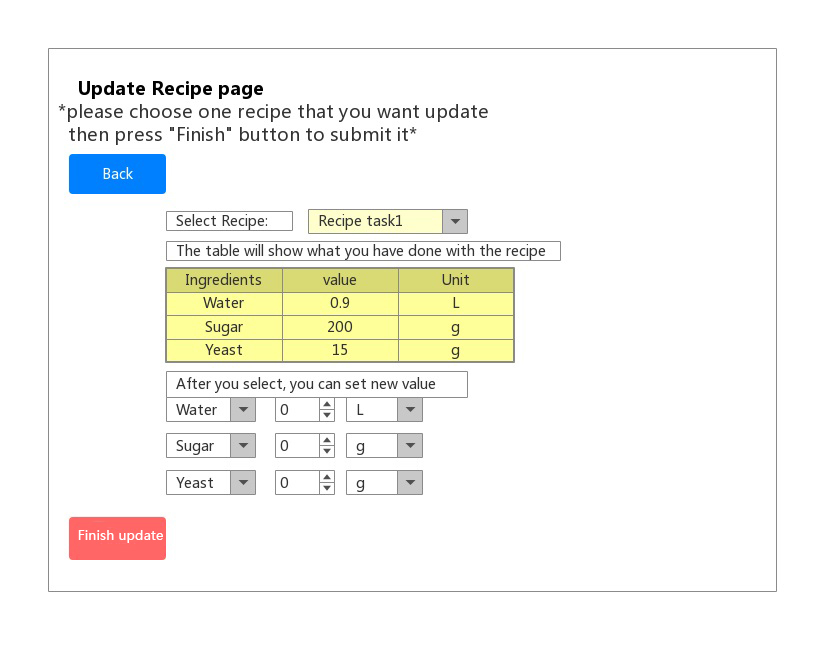
# External Interface Requirements (Written by Guo Rui)

## User Interfaces (Written by Chen Mingxuan & Ji jia)

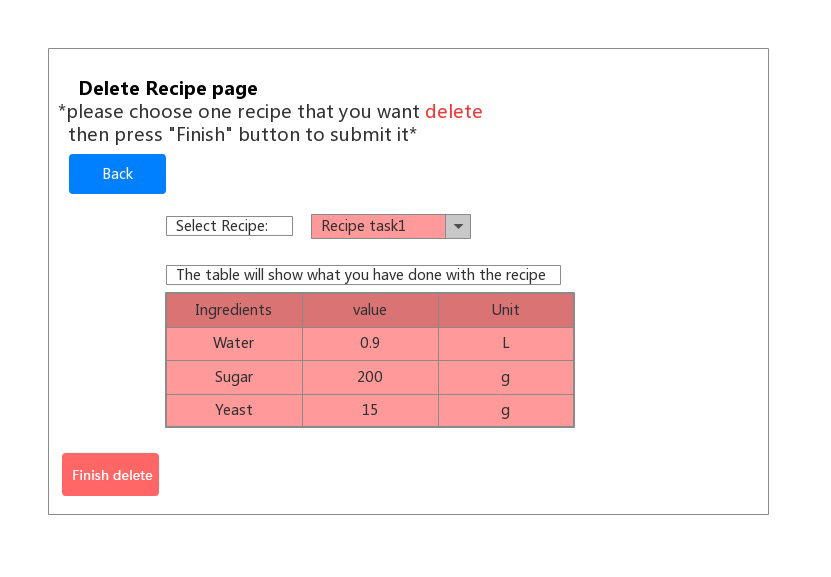
1. Maintain Recipe page



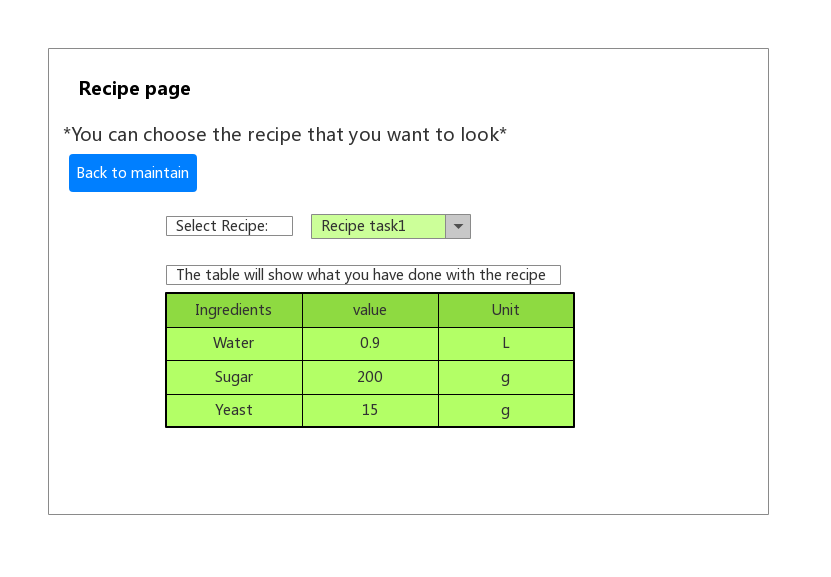
2. Add Recipe page



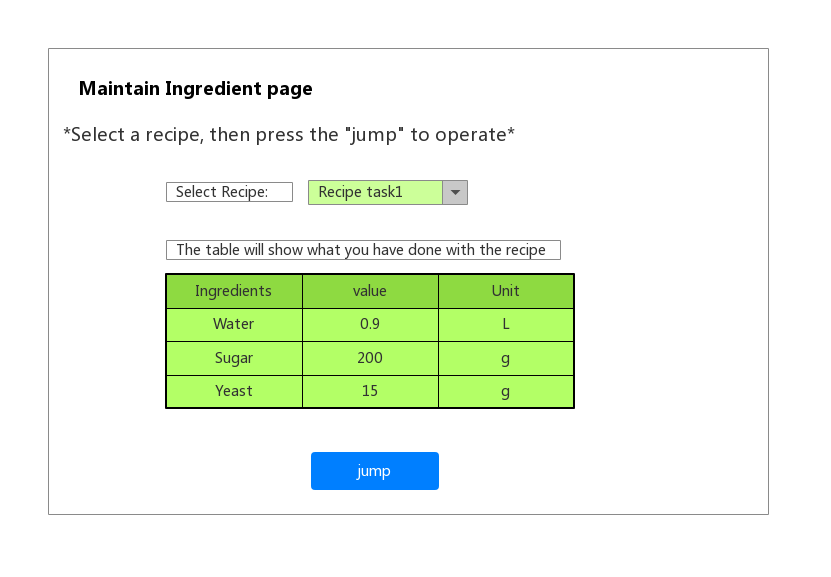
3. Update Recipe page



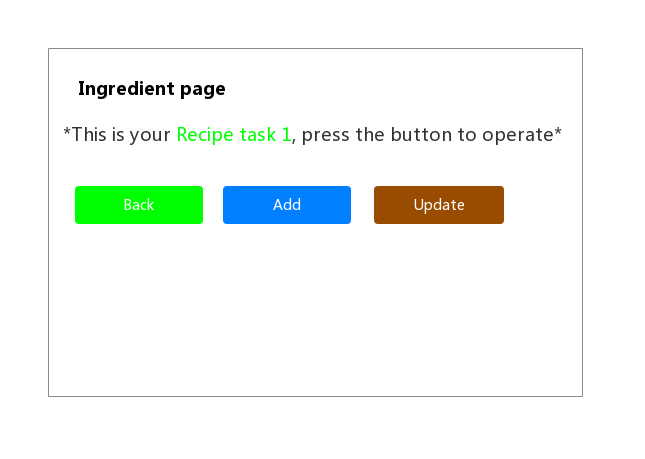
4. Delete Recipe page



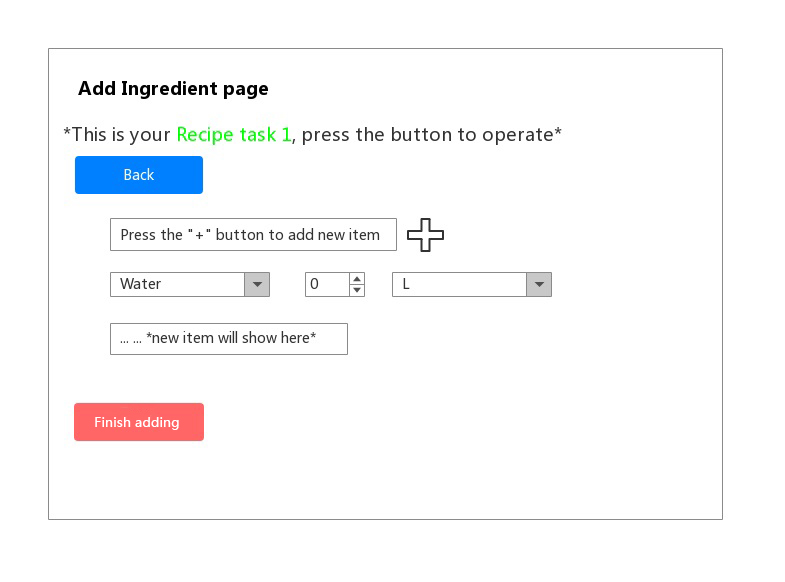
5. Recipe page



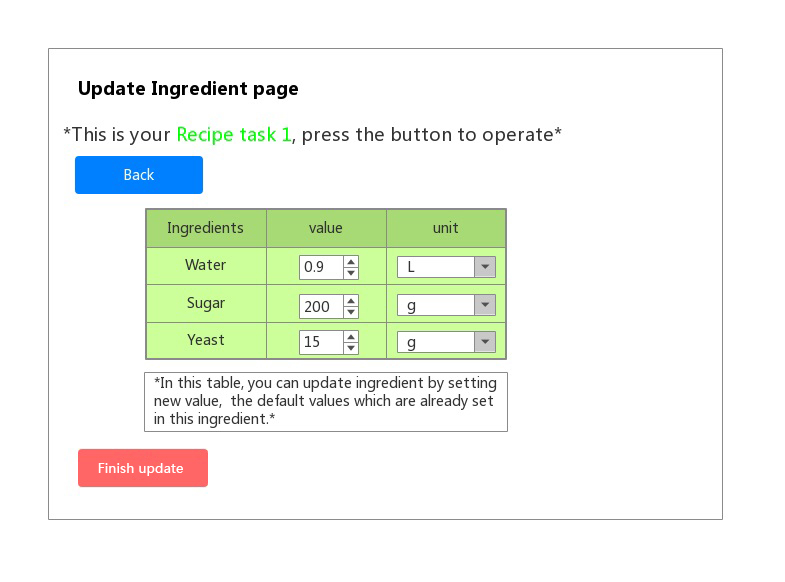
6. Maintain Ingredient page



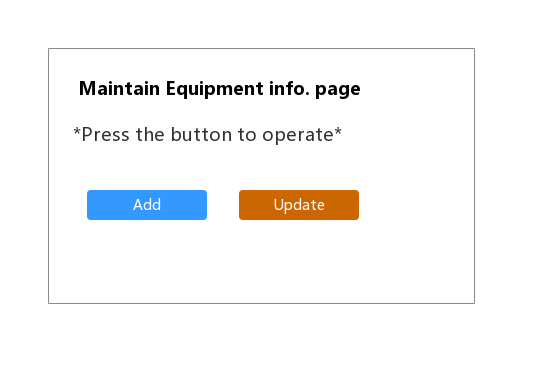
7. Ingredient page



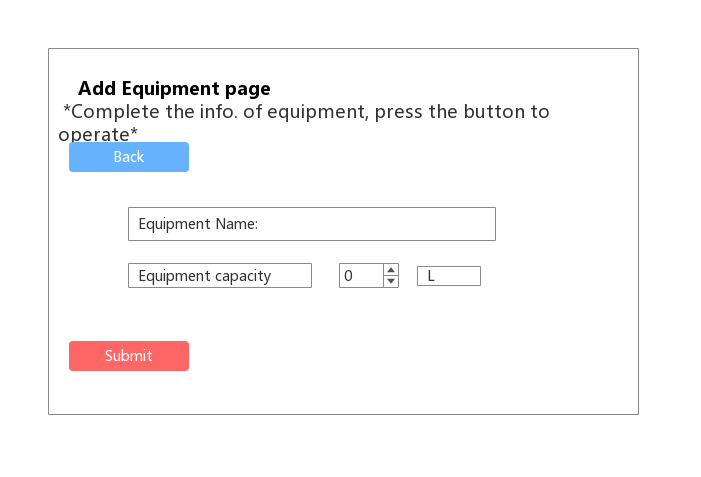
8. Add Ingredient page



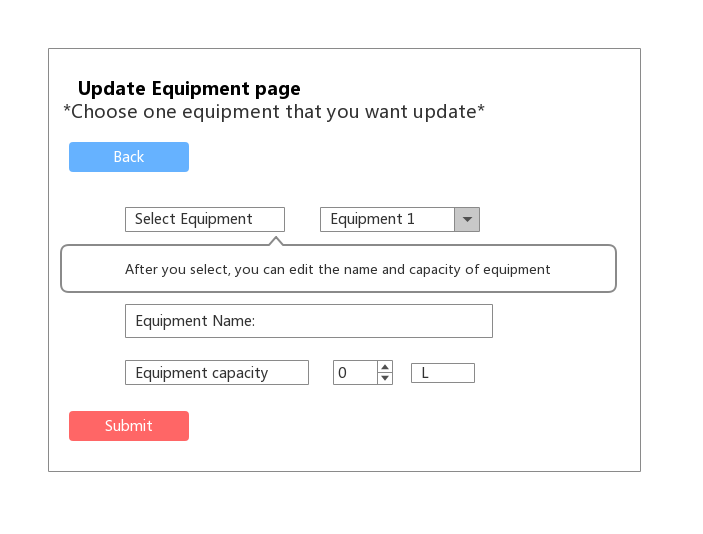
9. Update Ingredient page



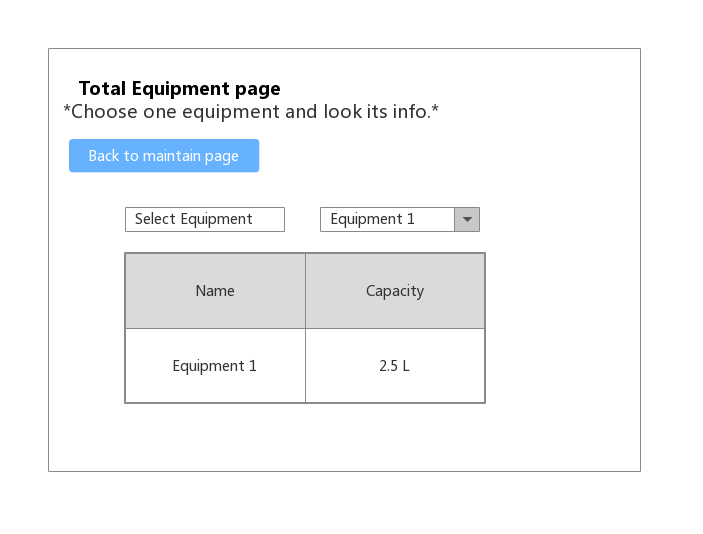
10. Maintain Equipment info page



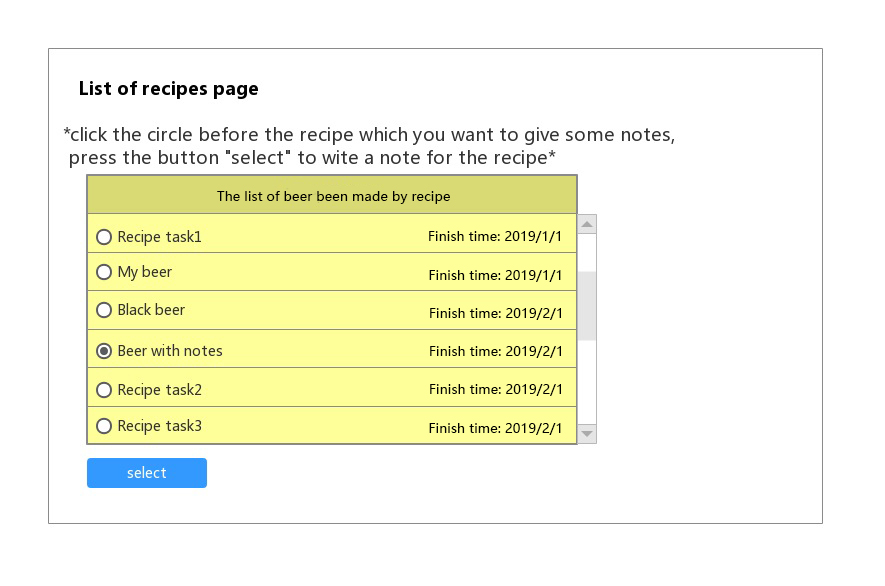
11. Add Equipment page



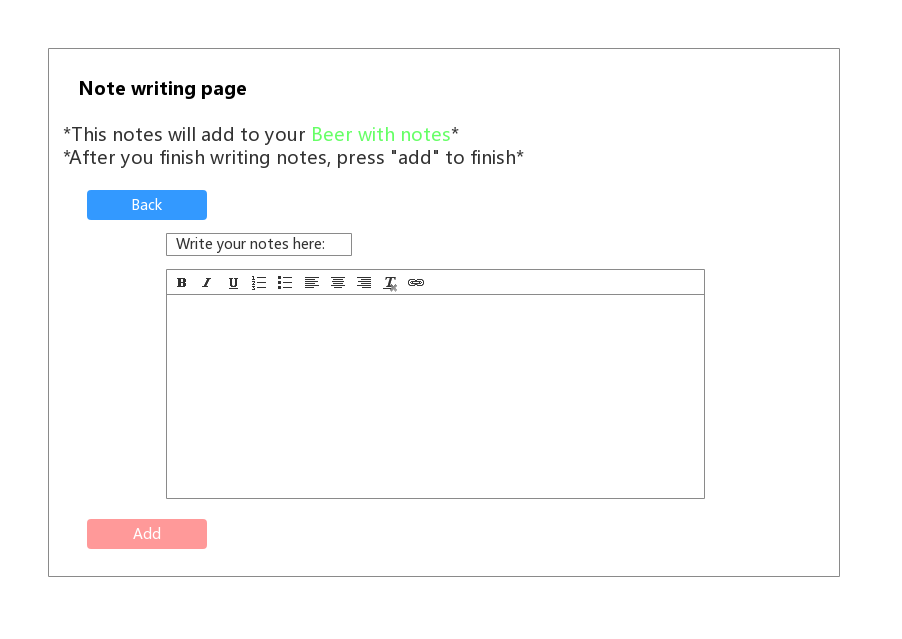
12. Update Equipment page



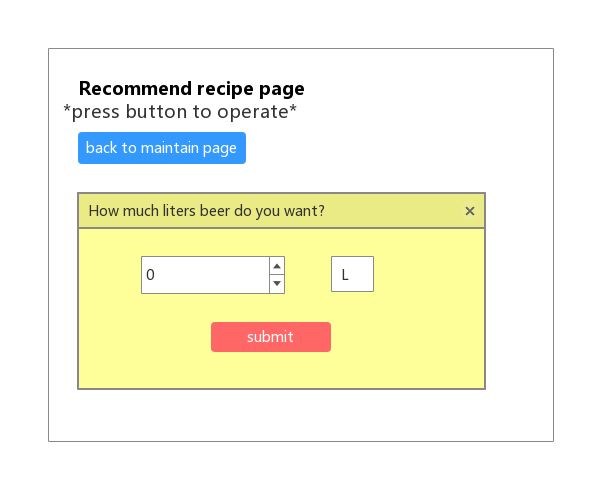
13. Total Equipment page



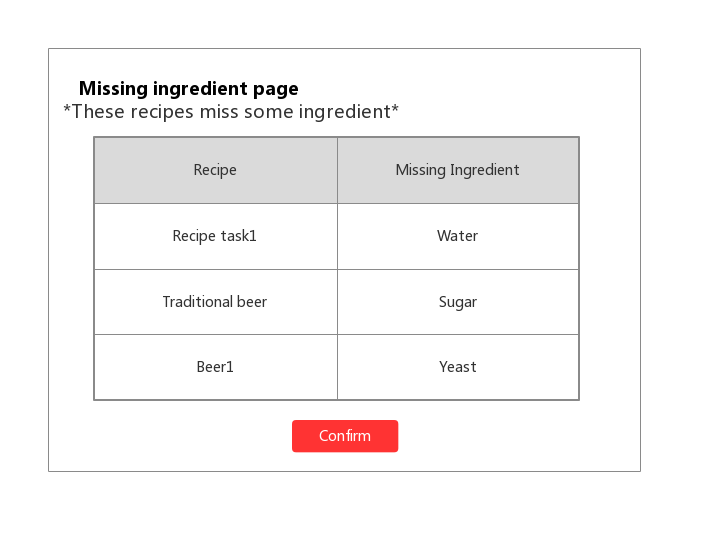
14. List of recipes page



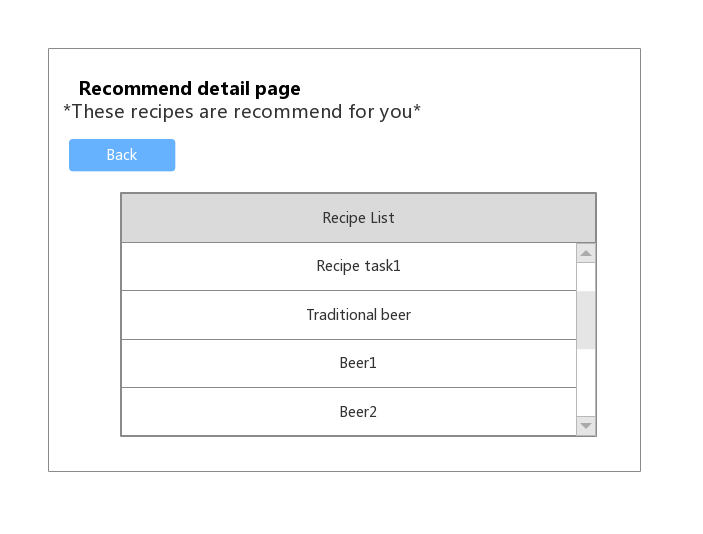
15. Note writing page



16. Recommend recipe page



17. Missing ingredient page



18. Recommend detail page

## Hardware Interfaces

For the software to run, you will need a PC (the operating system must be Windows10), and I/O devices. Since the application must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system.

Hardware device types:

* CPU
* Main memory
* Motherboard
* Graphics card
* Networking card
* Monitor
* Keyboard
* Mouse
* Hard disks( for users to download data from software website )
* USB Flash drives( for transfer user data to another computer)

## Software Interfaces

1. The software must open and run by using the Chrome browser.
2. The system shall connect with the online database, which provides the storage for software users.

## Communications Interfaces

The browser must be Chrome, and all the software UI base on this browser.

The software shall use the HTTP protocol for communication over the internet and for the intranet communication will be through TCP/IP protocol suite.

# Other Nonfunctional Requirements (Written by Chen Mingxuan)

## Performance Requirements

**Response time:**

In 95% of cases, the general time response time does not exceed 1 seconds, and the peak time does not exceed 4 seconds.

When the network is unblocked, it takes no more than 5 seconds for the software to connect to the corresponding brewing equipment.

In the recommended configuration environment: the login response time is within 2 seconds, the refresh response time of the column is within 2 seconds, the response time of the entry page list is refreshed within 2 seconds, the response time of the information entry is opened within 1 second, and the response time of the history task list is refreshed in 2 seconds.

Searching for specific conditions based on number and name during off-peak hours, you can get search results in 3 seconds.

**Business volume:**

The estimated number of users is 1,000, the number of login users per day is about 300, and the bandwidth of the network is 100M bandwidth.

The system can simultaneously satisfy 1,000 user requests and provide browsing capabilities for 5,000 concurrent users.

**System capacity:**

Support 30,000 users and support GB-level data.

The number of database table rows does not exceed 1 million rows, the maximum capacity of the database does not exceed 1000 GB, and the disk space needs at least 40G.

**Accuracy:**

When accessing the system via the Internet, it is expected that the maximum query time <15 seconds when searching for numbers and names.

The accuracy of the calculation is 7 digits after the decimal point.

**Resource usage rate:**

CPU usage <=30%.

Memory usage <=30%.

## Safety Requirements

This software only provides basic functions such as recording formula. Users are requested to abide by relevant regulations and laws in the process of using this software. Users are requested to pay attention to the problems that may arise especially in brewing beer. In case of safety accidents, the software does not bear any responsibility. The information provided in the software is for reference only.

## Security Requirements

Strict access control, after the user is authenticated, he can only access the data within its scope of authority, and can only perform operations within its scope of authority. Different users have different identities and rights. They need to provide trusted authorization management services under the premise of authenticity of users, protect data from illegal/over-authorized access and tampering, and ensure data confidentiality and integrity. Provides operational log management and security auditing to track historical usage of the system.

Network delivery data should be encrypted. It is necessary to ensure that data is not sneaked, stolen or tampered with during collection, transmission and processing. Business data needs to be encrypted at the time of storage to ensure that it is not hackable.

## Software Quality Attributes

**Usability:**

Within 3 months of the introduction of the product, 60% of users should be able to use it to create the first wine record in 45 seconds.

60% of users will realize that this is a beer brewing application within 5 seconds of seeing the product for the first time.

After receiving less than 15 minutes of system introduction training, 80% of users can learn basic application usage, such as logging in, logging out, editing personal data, creating winemaking tasks, viewing history, deleting records, and so on.

**Maintainability:**

After receiving the modification request, the general modification should be completed within 1~2 days; for the evaluation, the major demand or design modification should be completed within 1 week.

90% of the BUG modification time does not exceed 2 working days, and the other does not exceed 3 working days.

Installing the new version must keep all database content and all personal settings unchanged.

The product must provide a tool that can track any database field.

# Other Requirements (Written by Guo Rui)

This software is copyrighted, it is belong to Atanasoff Company. Atanasoff Company is legally entitled to all rights to the software, which provides paid use for commercial software. If not authorized by Atanasoff Company, you don’t have the freedom to copy and change this software. Any infringement will be prosecuted.

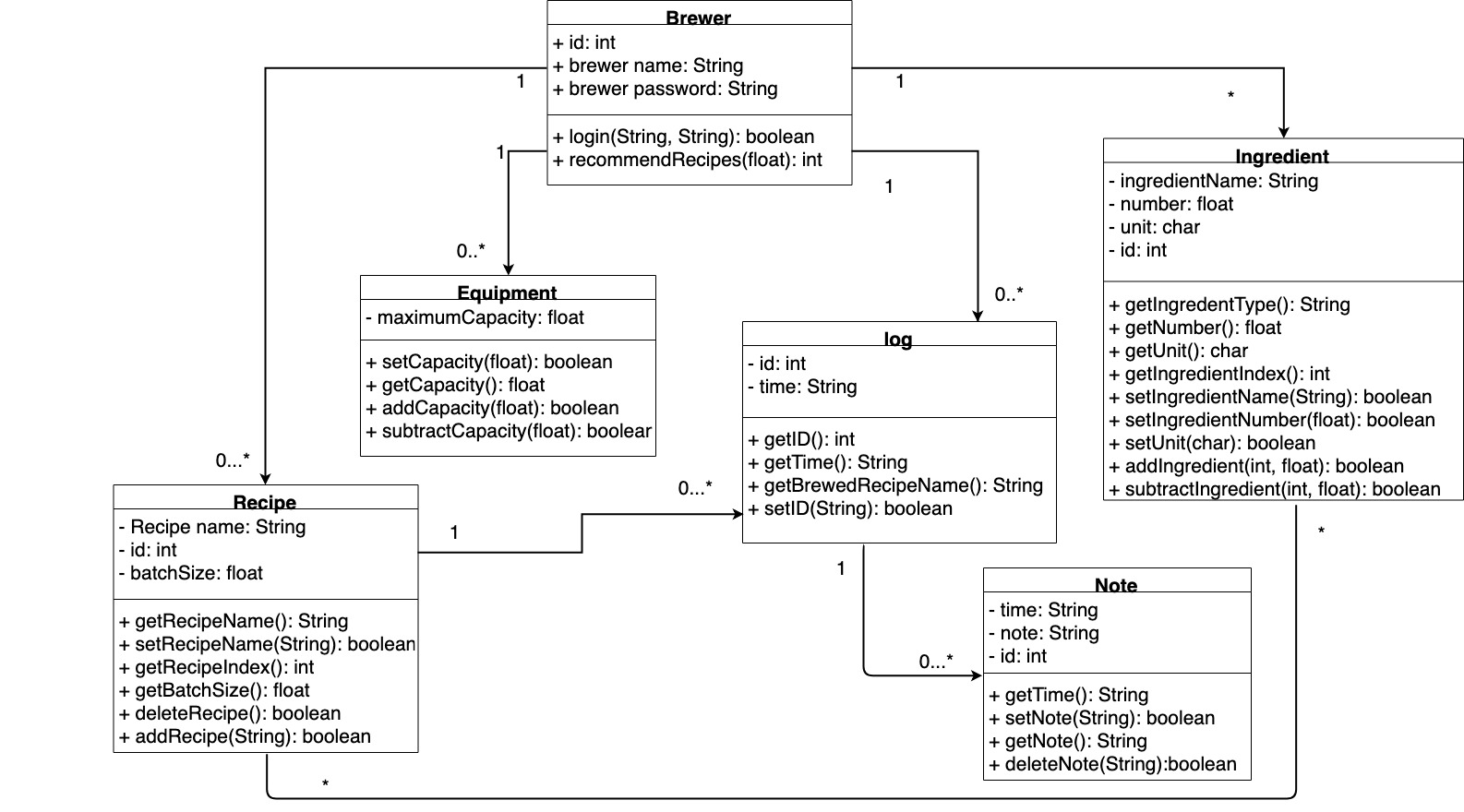
Any individual or company that has the above-mentioned infringement must immediately stop the infringement and bear full responsibility for all the adverse consequences caused by its infringement, and pursue its economic and legal responsibilities in accordance with the Copyright Law, the Computer Software Protection Regulations and other laws.

Appendix A: Glossary

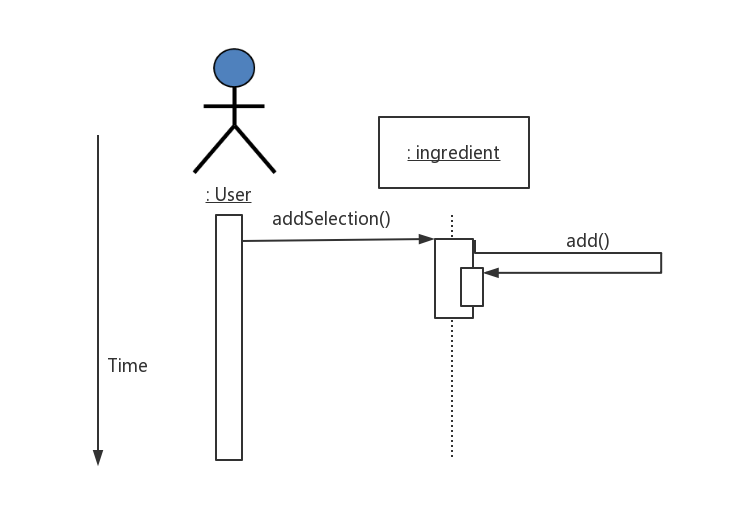
<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models(Written by Ji Jia, Guo Rui, Xie Qizhou)

Class diagram



Eequence diagram for the “Add” use case in maintaining ingredients.



Appendix C: Issues List

< This is a dynamic list of the open requirements issues that remain to be resolved, including TBDs, pending decisions, information that is needed, conflicts awaiting resolution, and the like.>

1Do you need to add photos to the notes？(SRS 2.2)

2Do you need some reminders of the process of brewing? (SRS 2.2)

3Is an online recipes sharing platform(share through WeChat)? (SRS 2.2 and SRS 4.3)

4Do you need a template of the recipes? (SRS 2.2)

5Do you need a system of grading the recipes?(SRS 2.2)

6Do you need a system of the comment of other users?(SRS 2.2)

7Do you want to run on the phone or computer?(SRS 2.4)

8Do you want to have a login password?(SRS 5.3)

9Do you need to record the price of each ingredient?(SRS 2.2)

10Do you need to classify different kinds of beer?(SRS 2.2)

11Do you need to compare the recipes of two different types of beer?(SRS 2.2)