Software Requirements Specification

for

< Brew Day!>

Version 3.0 approved

Prepared by <Kexin CAO>

<Changyu SHEN>

<Yuan ZHANG>

<Xianggao GU>

<Frege>

<3/5/2019>

Table of Contents

Table of Contents ii

Revision History iii

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 7

2.4 Operating Environment 7

2.5 Design and Implementation Constraints 7

2.6 User Documentation 7

2.7 Assumptions and Dependencies 7

3. System Features 7

3.1 Maintain Recipes 7

3.2 Maintain Ingredients 9

3.3 Recommend a Recipe 11

3.4 Write Note 12

3.5 Maintain Equipment Information 13

4. External Interface Requirements 14

4.1 User Interfaces 14

4.1.1 Maintain Recipes 14

4.1.2 Maintain Ingredients 15

4.1.3 Recommend a Recipe 15

4.1.4 Write Notes 16

4.1.5 Equipment Information 16

4.2 Hardware Interfaces 16

4.3 Software Interfaces 16

4.4 Communications Interfaces 17

5. Other Nonfunctional Requirements 17

5.1 Performance Requirements 17

5.2 Safety Requirements 17

5.3 Security Requirements 17

5.4 Software Quality Attributes 18

6. Other Requirements 18

Appendix A: Glossary 18

Appendix B: Analysis Models 18

Appendix C: Issues List 18

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Frege (all) | 3/5/2019 | Preliminary manuscript of the project | 1.0 |
| Frege(all) | 3/12/2019 | Preliminary manuscript of the project | 1.1 |
| Frege(all) | 3/19/2019 | Preliminary manuscript of the project | 1.2 |
| Frege(all) | 3/26/2019 | Preliminary manuscript of the project | 2.0 |
| Cao, Shen, Zhang | 4/2/2019 | Preliminary manuscript of the project | 3.0 |

# Introduction

## Purpose

This article is the version 1.0 SRS of an application called “Brew Day!” We will systematically introduce all the plans to help with programming the entire project, and also help customers better understand the features of this application and how testers test it. In this SRS, we will introduce the types of customers that this project applies to; the basic product characteristics, product design and its operating environment, dependencies; the requirements for product performance, safety and security; and the additional help for developers

## Document Conventions

In this SRS, all normal requirements are written in Times font, 12 font-size. For higher-level requirements, we will highlight them and use larger font-size.

## Intended Audience and Reading Suggestions

This SRS applies to developers, project managers and testers. The rest of the article consists of four important parts: Overall Description (Section 2), a brief introduction to the basic information of the project; System Features (Section 3), which detailing the features that the software needs to have; External Interface Requirements (Section 4), which detailing the various interfaces; and Other Nonfunctional Requirements (Section 5), which pointing out the performance that the software needs to achieve. For project managers, we recommend carefully reading 2.4, 2.7 and the entire section 3 (system features) to properly arrange the entire project before starting the program; for developers, we recommend reading the entire section 4 for better development; for testers, we recommend reading the entire section 5 so that they can know what requirement the software needs to accomplish.

## Project Scope

“Brew Day!” is an application which can make home brewers much easier to maintain their beer recipes with an organized database, which is benefit for them. Also, users are allowed to create, store and modify their recipes, even when they wish to delete them later, they can remove recipes conveniently. In addition, customers also need a special function “what should I brew today?” in which home brewers can choose the recipe that maximizes the use of the available ingredients, taking into account the equipment capacity. Optionally, developers may choose to allow ingredients availability manually, as opposed to do it automatically from brews information.

## References

Siebenthal, C. (2019). Learn the Step-by-Step Process Of Brewing

Beer, Explained in Animated GIFs. Retrieved from

https://firstwefeast.com/drink/learn-how-to-brew-beer-with-these-fun-animated-gifs

# Overall Description

## 图片5Product Perspective

This product is a new, self-contained product.

The beer recipe database could save all the recipes. And the recipe management system could adjust the recipes including add, delete, update. The ingredients database saves all the quantities of ingredients. And the number of ingredients could be added and subtracted. The recommendation system could look though the recipe database and according to the ingredient database to give a special recipe. Besides, the recoding system could record the recipe instances and notes after brewing beer saved in note database. Finally, it has a maintenance system to make sure everything goes well.

## Product Features

Manage the recipes: including add, delete, update.

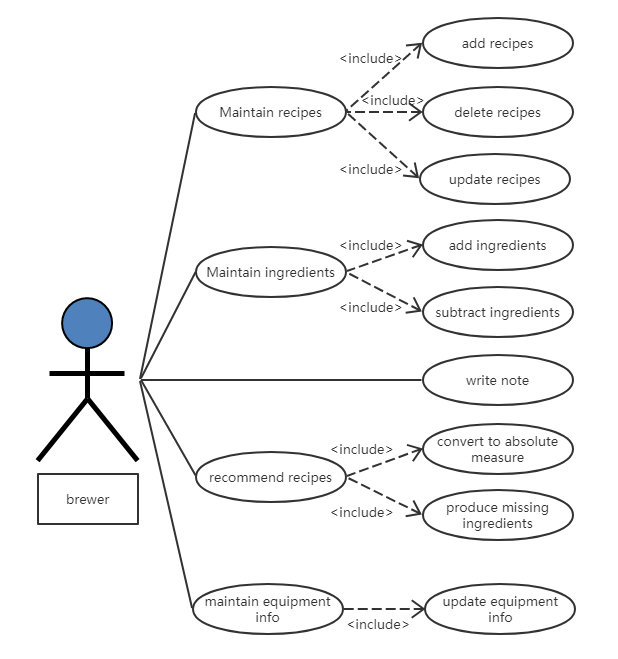
Change the quantity of ingredients: add, subtract.

Recommend recipes by batch size.

Record the recipe instances.

Take notes after brewing beer.

Remind missing ingredients for the next brew.



Basic scenario for “Update recipes”:

-The user chooses one recipe then click “Update”

-The Server displays the detail of recipe

-The user modifies the ingredients and click “Save”

-The Server saves the ingredients of recipes and displays it.

Alternative scenario – Repetitive recipes:

-The user chooses one recipe then click “Update”

-The Server displays the detail of recipe

-The user modifies the information and click “Save”

-The information is already saved

Basic scenario for “Add recipes”:

-The user clicks “Add” to create a new recipe

-The Server displays the information boxes of recipe

-The user fills in the information boxes and click “Save”

-The Server saves the information of recipes and displays it.

Alternative scenario – Repetitive recipes:

-The user clicks “Add” to create a new recipe

-The Server displays the information boxes of recipe

-The user fills in the information boxes and click “Save”

-The information is already saved

Basic scenario for “Delete recipes”:

-The user chooses one recipe then click “Delete”

-The Server displays the detail of recipe

-The user clicks “Delete” to be sure of deleting

-The Server delete the information of recipes and remind it successes.

Alternative scenario – Delete wrong recipes:

-The user chooses one recipe then click “Delete”

-The Server displays the detail of recipe

-The user to be not sure of deleting

-The deleting is canceled

Basic scenario for “Add ingredients”:

-The user enters the ingredient page

-The Server shows quantities of the ingredients

-The user adds the number of ingredients

-The Server adds ingredient amounts and displays the latest

Alternative scenario – Negative ingredient amount:

-The user enters the ingredient page

-The Server shows quantities of the ingredients

-The user add the a negative number of ingredients

- The adding fails because of Negative ingredient amount

Basic scenario for “Subtract ingredients”:

-The user enter the ingredient page

-The Server shows quantities of the ingredients

-The user subtracts the number of ingredients

-The Server subtracts ingredient amounts and displays the latest

Alternative scenario – Negative ingredient amount:

-The user enter the ingredient page

-The Server shows quantities of the ingredients

-The user subtracts the number of ingredients

-The subtracting fails because of Negative ingredient amount

Basic scenario for “Write note”:

-The user chooses one recipe instance.

-The Server displays the note of it.

-The user edits the notes of recipes instance then click “Save”

-The Server saves the notes and displays it.

Basic scenario for “Record recipe instances”:

-The user chooses one recipe and click “Do it”

-The Server displays the boxes of the conditions

-The user enters the conditions of recipes instance then click “Save”

-The Server saves the conditions of recipes and displays it.

Alternative scenario – Incomplete information

-The user chooses one recipe and click “Do it”

-The Server displays the boxes of the conditions

-The user enters the conditions of recipes instance then click “Save”

-The information can’t be saved and remind “Incomplete”

Basic scenario for “Convert to absolute measure”:

-The user chooses “Recommend recipe”.

-The Server displays the recipe and its ingredients according to the ingredients and batch size

-The user clicks “OK”

-The Server convert to absolute measure and show the data

-The user checks the information and clicks “Do it”

-The Server save the information and start brewing.

Alternative scenario – Incomplete information

-The user chooses “Recommend recipe”.

-The Server displays the recipe and its ingredients according to the ingredients and batch size

-The user clicks “OK”

-The Server convert to absolute measure and show the data

-The user checks the information and clicks “Do it”

-The information can’t be saved and remind “Incomplete”

Basic scenario for “Produce missing ingredients”:

-The user chooses one recipe.

-The Server displays the ingredients needs

-The user clicks “Missing”

-The Server remind the missing ingredients and their quantities

Alternative scenario – No missing ingredients

-The user chooses one recipe.

-The Server displays the ingredients needs

-The user clicks “Missing”

-No ingredient is missing

Basic scenario for “Update equipment info”:

-The user enters the equipment info page.

-The Server displays the equipment info

-The user fills the size box and clicks “Update”

-The Server update the info, remind success and show the info

Alternative scenario – Negative equipment amount

-The user enters the equipment info page.

-The Server displays the equipment info

-The user fills the size box and clicks “Update”

-The Server does not change anything and reminds fail then return to the equipment info page.

## User Classes and Characteristics

For people who haven’t brewed beer before, we set up beginner guide steps for them to learn to how to start brewing by this software. And they may prefer the recommend recipe function.

For people who have brewing experience but don’t know how to use software, the guide steps will be more focus on how to use it rather than how to brew beer.

For people who often brew beer and like using applications, they can skip the beginner guide steps and the record function is more important for them. Because they need this software to take notes after brewing beer and want it to maintain the ingredients and the recipes.

## Operating Environment

The software will operate under the standalone Windows operating environment.

## Design and Implementation Constraints

The Internet connection is a constraint for the application. Since the application fetches data from the database over the Internet, it is crucial that there is an Internet connection for the application to function.

## User Documentation

For user documentation and information, please consult attached user manual, and on-line help.

## Assumptions and Dependencies

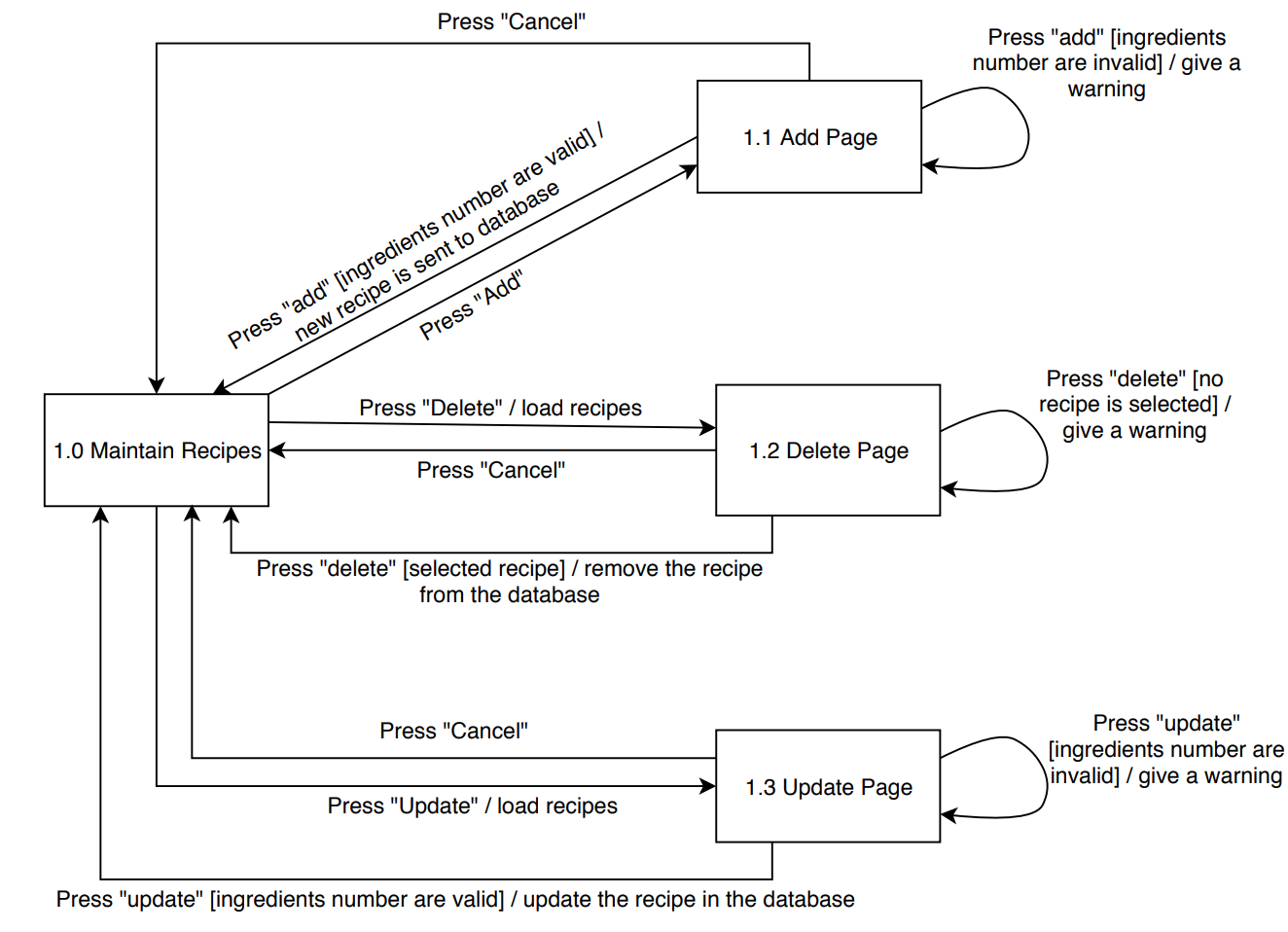
Assumption about the product is that it will always be used on Windows computers that have JRE. If the computer does not JRE, then user will install JRE for use this application.

# System Features

## Maintain Recipes

3.1.1 Description and Priority

This is the main feature provided for customers to maintain the whole recipes. It has three functions, which are “add”, “delete” and “update”. This feature is the most important part (rank 9) which is the basic function for the whole project. It will greatly improve users’ satisfaction about this project.

3.1.2 Stimulus/Response Sequences

3.1.3 Functional Requirements

When user in the “Maintain Recipes” page, there are three bottoms in it. When user click “Add” bottom, it needs to jump into the “Add” page, and in the “Add” page, users can add any recipe they like, or click “Cancel” bottom to return the “Maintain Recipes” page; When users click “Delete” bottom, it needs to jump into the “Delete” page, where user can find the recipe they do not like to remove, or click “cancel” bottom to return back the maintain page. If there is any recipe, show all of them; If there is not any recipe, give a warning only; When users click “Update” bottom, it needs to jump into the “Update” page, where user can find the recipe they need to change to modify, or click “cancel” bottom to return back the maintain page. If there is any recipe, show all of them; If there is not any recipe, give a warning only; If the user now is in the maintain recipes page, they can click “home” bottom to go back to the homepage.

REQ-1: In the “Maintain Recipes” page, “Add”, “Delete” and “Update” bottoms in it. Click “home” bottom can return to the main page.

REQ-2: In the “Add” page, can add recipe or click “cancel” bottom to return the “Maintain Recipes” page.

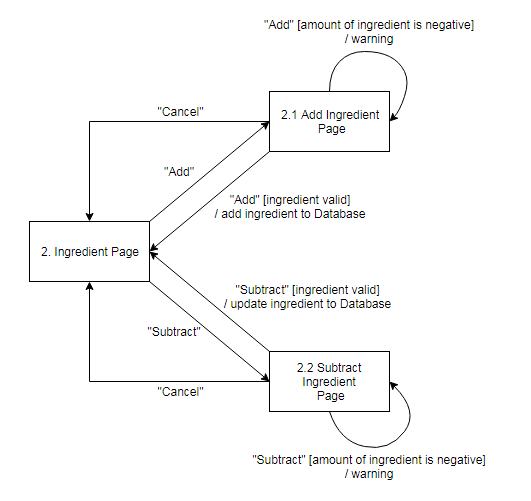
REQ-3: In the “Delete” page, can remove the recipe, or click “cancel” bottom to return back the maintain page. If there is any recipe, show all of them; If there is not any recipe, give a warning only.

REQ-4: In the “Update” page, can update the recipe, or click “cancel” bottom to return back the maintain page. If there is any recipe, show all of them; If there is not any recipe, give a warning only.

## Maintain Ingredients

3.2.1 Description and Priority

This key feature provides customers with the ability to record all ingredients. It has two functions, "Add" and "Subtract". The priority of the feature is the highest (rank 9), and the sub-features, “Add” and “Subtract” are the second highest (rank 8). It is necessary for the complete user use and experiment.

3.2.2 Stimulus/Response Sequences

3.1.3 Functional Requirements

After clicking “Add” button in the Ingredient Page, it will jump to the Add Ingredient Page. Fill in the add box, click the “Add” button in the Add Ingredient Page, it will save the new amount of the ingredient to the database, then it will jump back to Ingredient Page. If the input number is negative, the add action will be failed, then it will give a warning and jump back to Add Ingredient Page. If no input in some frames, the amount of them will not be changed. If click “Cancel” button, it will jump to the Ingredient Page and do not change anything.

After clicking “Subtract” button in the Ingredient Page, it will jump to the Subtract Ingredient Page. Input the positive number of ingredients, click the “Subtract” button in the Subtract Ingredient Page, it will update the amount of ingredient to the database, then it will jump back to Ingredient Page. If the input number is negative or larger the exist amount (the final amount will be negative), the subtract action will be failed, then it will give a warning and jump back to Subtract Ingredient Page. If no input in some frames, the amount of them will not be changed. If click “Cancel” button, it will jump to the Ingredient Page and do not change anything.

REQ-1: In the Ingredient Page, “Add” and “Subtract” bottoms in it. Click “Home” bottom can return to the main page.

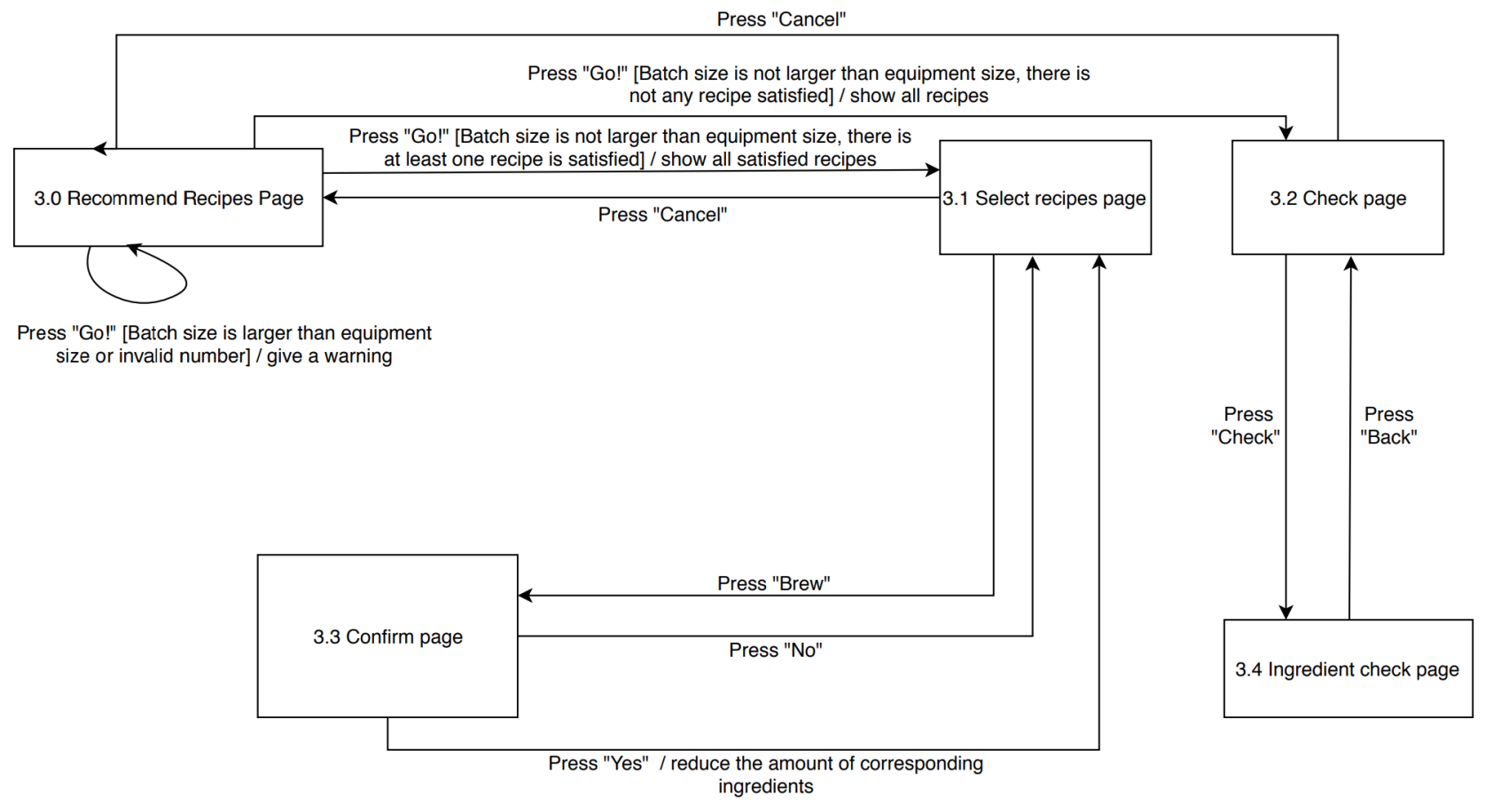
REQ-2: In the Add Page, can add ingredients or click “Cancel” bottom to return the Ingredient Page.

REQ-3: In the Subtract Page, can update the ingredients, or click “Cancel” bottom to return back the Ingredient Page.

## Recommend a Recipe

3.3.1 Description and Priority

“Recommend Recipes” is a high priority features which is the customers’ compulsory request. This function can recommend many kinds of recipes to customers according to the batch size they enter. The more precise it recommends, the more satisfied customers will be.

3.3.2 Stimulus/Response Sequences

3.3.3 Functional Requirement

In the Recommend page, users are asked to input the batch size. After they enter the batch size, the system will check whether this batch size is smaller than the equipment size, if larger, give warning, otherwise, if there is at least one recipe is satisfied, jump into the Select Recipes Page; if there is not any recipe satisfied, jump into the Check page. In the Select Recipes Page, all the satisfied recipes will show in the list. After clicking “Brew” button, the confirm page will appear and user can decide whether he choose “Yes” or “No”; In the Check page, all the recipes will be showed in the list (the recipe which lack the least ingredients will show in the top), and user can click “Check” to check the detail.

REQ-1: In the Recommend page, users are asked to input the batch size.

REQ-2: System checks batch size. If bigger than equipment size, give warning, otherwise, if there is at least one recipe is satisfied, jump into the Select Recipes Page; if there is not any recipe satisfied, jump into the Check page.

REQ-3: In the Select Recipes Page, all the satisfied recipes will show in the list. After clicking “Brew” button, the confirm page will appear and user can decide whether he choose “Yes” or “No”. If user chooses “Yes”, system will reduce the amount of corresponding ingredients; If user chooses “No”, system will jump back to Select recipes page.

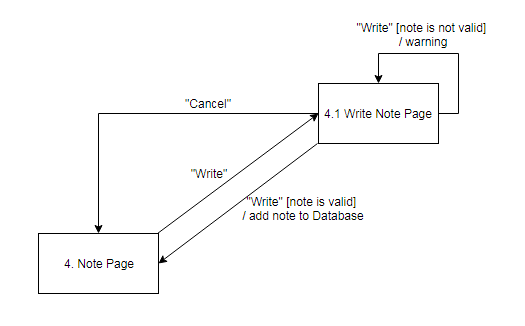
REQ-4: In the Check page, all the recipes will be showed in the list (the recipe which lack the least ingredients will show in the top), and user can click “Check” to check the detail and then click “Back” to go back.

## Write Note

3.4.1 Description and Priority

Provide users with “Write Note” feature. Each time a brew is made, the user creates a new note which is following the recipe instance. This feature has a middle priority. “Write” function can create a new note (rank 8).

3.4.2 Stimulus/Response Sequences



3.4.3 Functional Requirement

After click “Write” button in the Note Page, it will jump to the Write Note Page. Then fill in the date, recipe name and content of note, click “Write” button in the Write Note Page. If the note is valid, it will save to the database and it will jump to the Note Page. If the note is not valid or click the “Cancel” button, it will give a warning and go back to the Note Page without doing anything.

REQ-1: In the Note page, user can see the titles of notes and could know when they created, “Write” button for create a new note and click title could read the specific one.

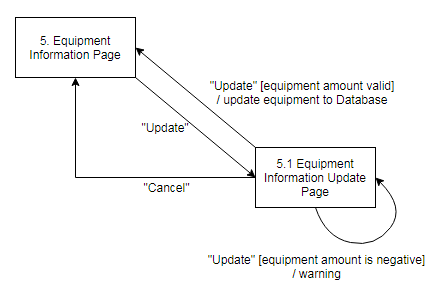
REQ-2: In the Write Note Page, can write a new note or click “Cancel” to jump to the Note Page.

## Maintain Equipment Information

3.5.1 Description and Priority

“Maintain Equipment Information” is a high priority feature. This function can record the name, amount and capacity of the equipment for subsequent user production. It has two functions, “Add” and “Update”. The priority of the feature is the highest (rank 9), “Add” and “Update” are the second highest (rank 8).

3.5.2 Stimulus/Response Sequences



3.5.3 Functional Requirement

After clicking “Update” button in the Equipment Information Page, it will jump to the Equipment Information Update Page. Fill in the add box, click the “Update” button, it will save the updated information to the database, then it will jump back to Equipment Information Page. If there is equipment already existing, the add action will be failed, then it will give a warning and jump back to Equipment Information Update Page. If click “Cancel” button, it will jump to the Equipment Information Page and do not change anything.

REQ-1: In the Equipment Information Page, “Update” bottoms in it. Click “Home” bottom can return to the main page.

REQ-2: In the Update Page, can update the equipment, or click “Cancel” bottom to return back the Equipment Information Page.

# External Interface Requirements

## User Interfaces

### Maintain Recipes

### C:\Users\l630021001\Downloads\1385790983.jpgC:\Users\l630021001\Downloads\541906735.jpgMaintain Ingredients

### Screen%20Shot%202019-03-19%20at%2016.49.23.png Recommend a Recipe

Note:

√: means the ingredient of this recipe are enough

✘: means the ingredient of this recipe is not enough

### Write Notes

### Equipment Information

## Hardware Interfaces

None.

## Software Interfaces

1. The Howe Brewing System shall communication with the beer recipe database to store and to collect users’ beer recipe on cloud.

2. The Howe Brewing System shall communication with the ingredient database to record every user’s remaining ingredients.

3. The Howe Brewing System shall communication with the maintenance system to maintain the recipe instances.

4. The Howe Brewing System shall communication with the ingredient management system for handling the modification of user’s ingredient database each time users consume their ingredients.

5. The Howe Brewing System shall communication with the recipe management system to enable users to change their recipes.

6. The Howe Brewing System shall communication with the recording system to allow users to write some notes for each time they brew beer.

7. The Howe Brewing System shall communication with the note database to record every note written by users.

8. The Howe Brewing System shall communication with the recommendation system to select and give the user the best recipe, which can be brewed with the available ingredients, maximizing the use of the ingredients, and the batch size.

## Communications Interfaces

None.

# Other Nonfunctional Requirements

## Performance Requirements

The customer of this product is family-based, which means that an application has a private database. Not a large-scale software. In order to provide a good user experience, the system response time should be within 1 seconds. In case of opening databases, sorting questions and computing there are no delays and the operation is performed in less than 1 second for opening, sorting, computing > 95% of the files. UI elements that do not involve outside connections to social media resources will respond to user input immediately with no noticeable lag.

## Safety Requirements

The software background will have backup data. Once the program crashes, opening the software again will call the backup data to protect the user's operation and modification.

## Security Requirements

This application involves personal privacy and property security. The user's material inventory will not be easily leaked. User names and passwords for access to our application (Brew Day!) should be stored (encrypted) securely. Account login is performed account API’s following the specification given by the API documentation to maintain account security. Our application will not monitor or collect data on user information.

## Software Quality Attributes

**Adaptability**: Checking that the application always has something to function and always pop up error messages in case of component failure. In this case, the error messages appear when something goes wrong so to prevail availability problems.

**Maintainability**: This application code is detailed in detail, clearly defined, easy to find problems, and convenient to maintain and modify.

**Reusability**: The code for its inventory system can be applied to all other inventory management applications.

**Robustness**: If the application is unresponsive, it will automatically close, automatically retain the user state before the shutdown, and the information edited by the user is automatically saved as a draft file. If the application is interrupted while increasing/decreasing the material inventory, this operation will not be saved, and the accuracy of the inventory quantity is guaranteed.

**Usability**: The interface is simple and beautiful, and the user manual is detailed. The operation is simple and easy to use.

# 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.>

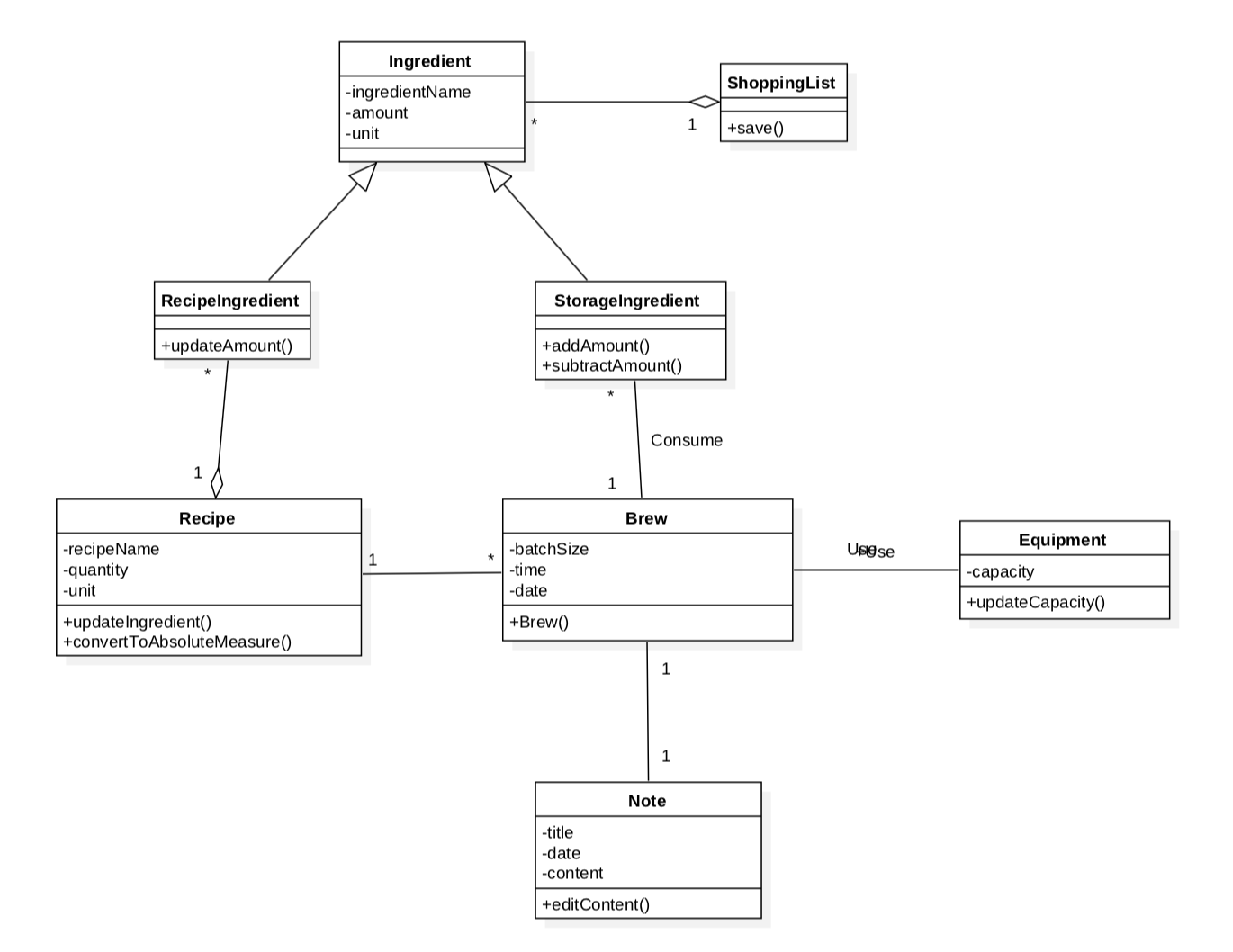
<TBD>

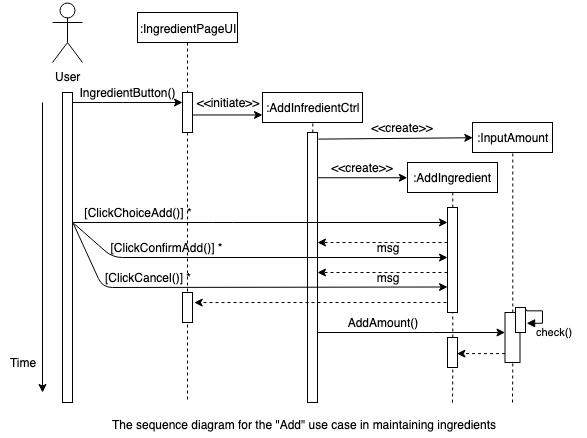
Appendix A: Glossary

<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.>

<TBD>

Appendix B: Analysis Models





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.>

<TBD>