Software Requirements Specification

for

Brew Day!

Version 2.0 approved

Deleted: 1

Prepared by Zhenghao Wu, Anqin Zha, Renjie Deng and Ruichao Zhong

Dijkstra

Tuesday, March 12, 2019

Deleted: Monday, March 11, 2019

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Zhenghao Wu, Anqin Zha, Renjie	2019-03-04	The first version of the SRS	1.0

Deleted: 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 3
2.4 Operating Environment 3
2.5 Design and Implementation Constraints 3
2.6 User Documentation 3
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 4
5.1 Performance Requirements 4
5.2 Safety Requirements 4
5.3 Security Requirements 5
5.4 Software Quality Attributes 5
6. Other Requirements 5
Appendix A: Glossary 5
Appendix B: Analysis Models 5
Appendix C: Issues List 5

Deng and Ruichao Zhong			
Zhenghao Wu, Anqin Zha, Renjie Deng and Ruichao Zhong	2019-03-12	Add two feature details and corresponding user interface design Change reference to APA format	2.0

Deleted: ¶

1. Introduction

1.1 Purpose

This document is for product named ‘‘Brew *Day!" which* developed by group *Dijkstra*. The document will be covered Introduction of the document, overall description, main features, external requirements and other non-functional requirements of product named *Brew Day!*

This document is written to help developers to develop the software conveniently, and for client to know what major features this product will provide.

This document is for the whole system of “*Brew Day!"* And no document for the partial function is provided.

1.2 Document Conventions

The title will use Times New Roman as font, for the first level title, the font size is 18, for the second level title, the font size is 14.

The main part of this document will use the font “Arial” and the font size is 11. Additionally, this document will use **bold text** to emphasize the important words. and *italic text* to indicate product names and company names in the main part.

For the acronyms in this document, please refer to Appendix A in part 7.

1.3 Intended Audience and Reading Suggestions

This Document is written for clients, developers and tester.

- For clients, the description of this product in part 2 and main features of this product in part 3 are recommended.
- For developers, the whole document is recommended to read.
- For tester, the content in part 2 and the nonfunctional requirements in part 5 are recommended.

1.4 Project Scope

This product is made for home brewer to brew beers. With this product, home brewer will have the ability to record and modify the information of ingredients relative to brewing beer to create a better brewing experience.

1.5 References

- Brew Target: Main brewtarget source code repository. (2019). GitHub. Retrieved 11 March, 2019, from https://github.com/Brewtarget/brewtarget
- Allaboutbeercom. (2019). All About Beer. Retrieved 11 March, 2019, from http://allaboutbeer.com/learn/glossary/

Deleted: An open-source brewing software https://github.com/Brewtarget/brewtarget

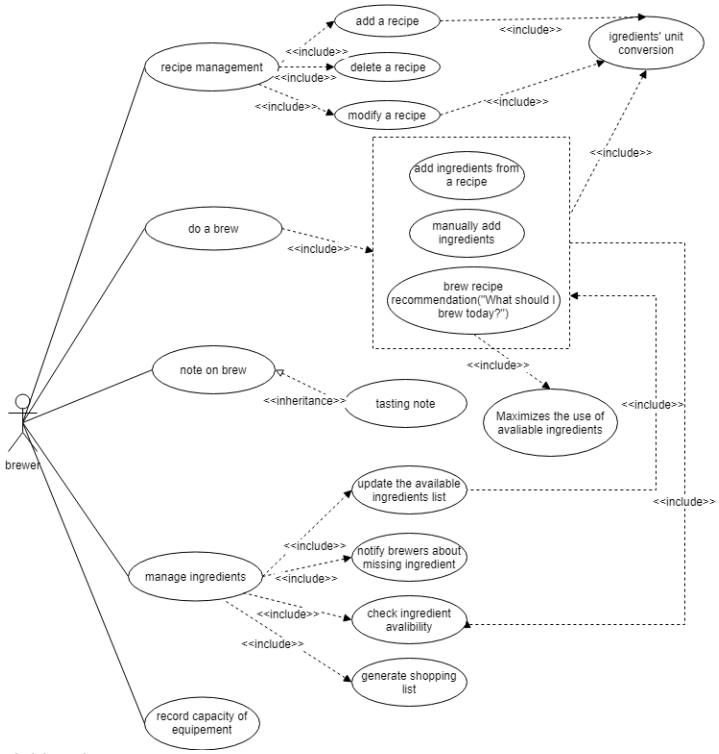
Deleted: <#>Beer Glossary http://allaboutbeer.com/learn/glossary/¶

2. Overall Description

2.1 Product Perspective

This is a new independent system and there is no context in this system.

2.2 Product Features



Scenario 1: Add recipe

- Brewer enter the recipe management and click add recipe
- The system displays a recipe input form
- The Brewer input the recipe information
- The system converts the ingredient unit
- The system save the recipe to database

2.3 User Classes and Characteristics

The only user class is “all-grain” brewers for their home brewing on a small scale (the "extract" brews are not supported). Majority of them are expert in brewing. Additionally, they have some basic knowledge about computer. For example, they know how to use an application in Windows desktop.

2.4 Operating Environment

The application is desktop-based. In addition, the application should be able to run on Microsoft Windows.

2.5 Design and Implementation Constraints

N/A

2.6 User Documentation

An independent user manual will be provided, which will be written in English.

2.7 Assumptions and Dependencies

The user should have a PC running Windows. And user have a Java Runtime Environment 11 to run the program.

3. System Features

3.1 Maintain recipes

3.1.1 Description and Priority

Users can use this feature to view their recipes and manage their own recipes by doing addition, deletion and modification operations to the recipes. This is a high priority feature in the system.

3.1.2 Stimulus/Response Sequences

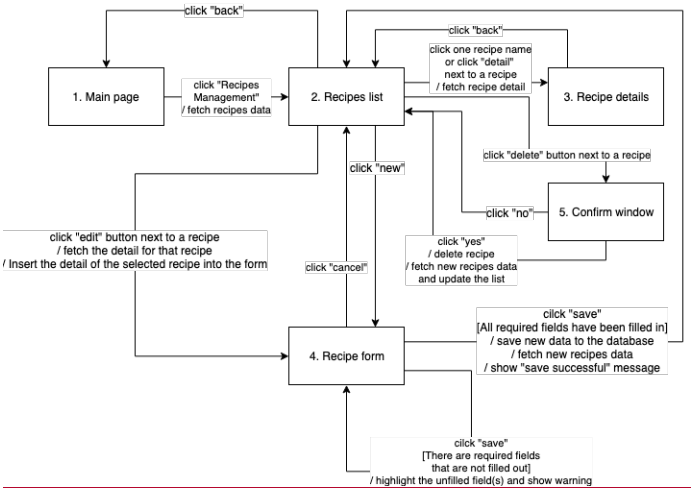
Deleted: <*This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.*>¶
System Feature 1

Deleted: <*Don’t really say “System Feature 1.” State the feature name in just a few words.*>¶
3.1.1→

Formatted: Font: (Default) Arial, 11 pt

Formatted: Normal

Deleted: <*Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).*>¶
3.1.2→



3.1.3 Functional Requirements

REQ-1: When the user creates and modifies the recipe, the system displays a recipe form for filling out. When the user submits the form, the system checks that all required fields of the form have been filled in and the format is correct. If the request is not met, the system rejects the submission of the form, highlights the fields that not met the requirement in the form, and displays a warning message.

3.2 Maintain ingredients

3.2.1 Description and Priority

The feature is to let users manage amount of ingredients in their house. If they have bought a new ingredient, they can use “add” to add it to system. If the ingredient is existing and its amount has changed, we can use “update” to change its amount.

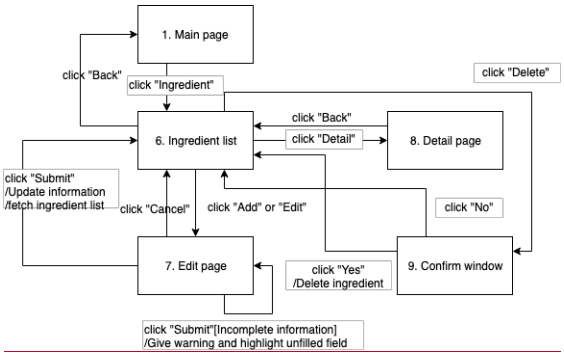
3.2.2 Stimulus/Response Sequences

Formatted: Normal, Centered

Deleted: *<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>*¶
3.1.3→

Formatted: Heading 3

Formatted: Font: Not Italic



3.2.3 Functional Requirements

REQ-2: If any of user's input is empty, the system will ask user to complete the form.
REQ-3: If any of user's input is not match its data type, the system will ask user to fix.

Formatted: Font: Not Italic

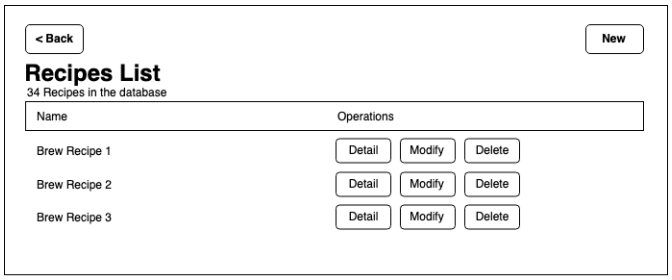
4. External Interface Requirements

Deleted: <*Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.*>¶
¶
<*Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.*>¶
¶
REQ-1:¶
REQ-2:¶
System Feature 2 (and so on)¶

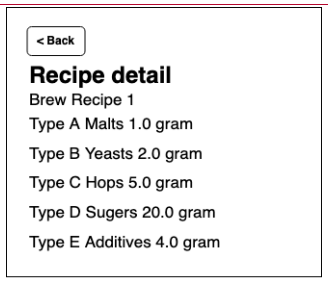
Formatted: Normal, Centered

4.1 User Interfaces

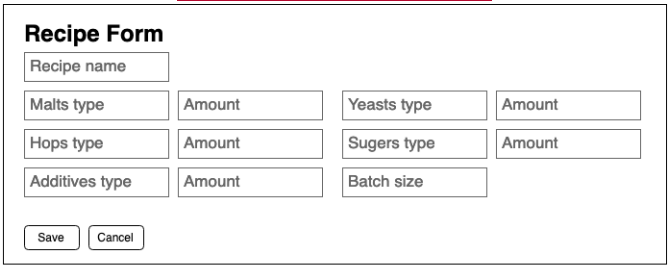
1. Main page



2. Recipes list

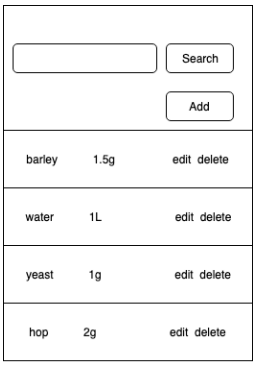


3. Recipe details

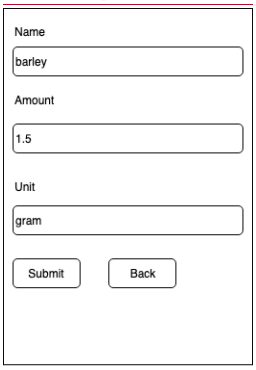


4. Recipe form

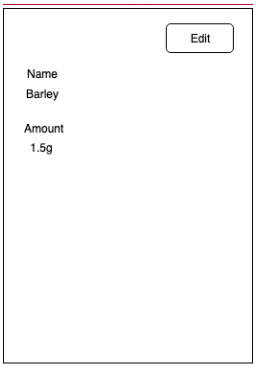
5. Confirm window



6. Ingredient list



7. Edit page



8. Detail page

Formatted: Centered

Formatted: Normal

4.2 Hardware Interfaces

The software does not have the function to control the hardware.

4.3 Software Interfaces

The software queries the database to display the information displayed on the software user interface, and the user can operate the software user interface to perform database addition, query, deletion and modification.

4.4 Communications Interfaces

This software is an offline software which means communication function will not be provided.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The response of search operation should be less than one second, other operations should be less than two seconds.

5.2 Safety Requirements

The system will not check whether the user's recipe is safety or not. Users should be responsible for their own recipe.

5.3 Security Requirements

Password protection is not provided in the software.

5.4 Software Quality Attributes

Maintainability: Detailed documents would be written to ensure the software could be easily maintained in the future. Each method has comments to help developer to understand.

Robustness: The software has a Mean time to failures (MTTF) for 2000 hours per failure.

Usability: The software will display all the functions directly on the window and would be easy for users to learn to use. Each function should be able to access within 10 clicks.

Reusability: With modular design, some functions of the system could be reuse on other software.

6. Other Requirements

TBD

Appendix A: Glossary

- N/A: Not Applicable
- TBD: To be determined

Appendix B: Analysis Models

TBD

Appendix C: Issues List

TBD