Software Requirements Specification

for

Brew Day!

Version 2.0 approved

Prepared by ZHANG Zhiyi, LUO Zichen, WANG Yuan, LU Guangxing

Cerf

4 March 2019

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction (team work, everyone contributes every section) 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 2

2. Overall Description (team work, everyone contributes every section) 2

2.1 Product Perspective 2

2.2 Product Features 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 3

2.6 User Documentation 3

2.7 Assumptions and Dependencies 3

3. System Features 3

Maintain Recipes 3

3.1 3

3.2 Maintain Ingredients 4

4. External Interface Requirements (team work, everyone contributes every section) 5

User Interfaces 5

*4.1* 5

4.2 Hardware Interfaces 6

4.3 Software Interfaces 6

4.4 Communications Interfaces 6

5. Other Nonfunctional Requirements (team work, everyone contributes every section) 6

5.1 Performance Requirements 6

5.2 Safety Requirements 6

5.3 Security Requirements 6

5.4 Software Quality Attributes 6

6. Other Requirements (team work, everyone contributes every section) 6

Appendix A: Glossary 7

Appendix B: Analysis Models 7

Appendix C: Issues List 7

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| ZHANG Zhiyi,  LUO Zichen,  LU Guangxing,  WANG Yuan | 3/4/2019 | Initialial SRS Document | 1.0 |
| ZHANG Zhiyi,  LUO Zichen,  LU Guangxing,  WANG Yuan | 3/11/2019 | Edit Section 3 and Section 4.1 | 2.0 |

# Introduction (team work, everyone contributes every section)

## Purpose

This document is going to describe the whole software named Brew Day!, which is now designing. And this is the first version of SRS regarding this software.

## Document Conventions

In this document, we particular use these conventions for clarity and readability:

* **Brew Day!** is the name of the software/system we are developing. And for clarity in context, we describe it in an underline format.
* **SRS document** refers to the specific software requirements specification we write to describe our Brew Day! software, if not specifically pointed out.
* **Software** refers to the Brew Day! software, if not specifically pointed out.
* **TBD** for sections which we have not acquired enough information to fill in, and **N/A** for sections which are temporarily not applicable.

## Intended Audience and Reading Suggestions

We suggest everyone who is related to this software read all parts, but if there will be some difficulty while reading, we suggest:

If you are developers, testers and documentation writers, please read all part of this document carefully.

If you are project managers and marketing staff, please focus on **2. Overall Description** and **5. Other Nonfunctional Requirements.**

If you are users, please read **1. Introduction.**

## Project Scope

This software is developed for home beer brewers, to give them some little help such as record recipes, keep track of ingredients, etc. This software will be desktop-based. A brief user manual will also be provided.

## References

# Overall Description (team work, everyone contributes every section)

## Product Perspective

No context. This is a new independent software. It is the first version.

## Product Features

A basic scenario for “Recommend a recipe” use case:

* The user runs the “Recommend a recipe” function.
* The software selects one recipe from the library, basing on certain criteria.
* The software displays the selected recipe on the interface.

## User Classes and Characteristics

This software is developed mostly for home beer brewers. We prefer users with certain ability to use a computer and knowledge of beer brewing.

## Operating Environment

This software will be operated on desktop

## Design and Implementation Constraints

No constraints.

## User Documentation

User manuals will be provided.

## Assumptions and Dependencies

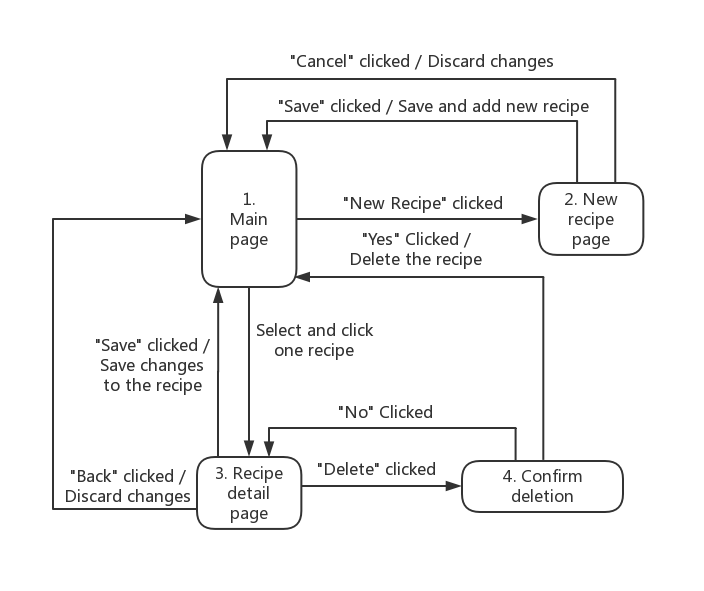
TBD

# System Features

## Maintain Recipes

3.1.1 Description and Priority

Maintain recipes is the fundamental feature of this software, which is also has the highest priority. In this feature, user could add new recipes, delete recipes and update existing recipes.

3.1.2 Stimulus/Response Sequences

3.1.3 Functional Requirements

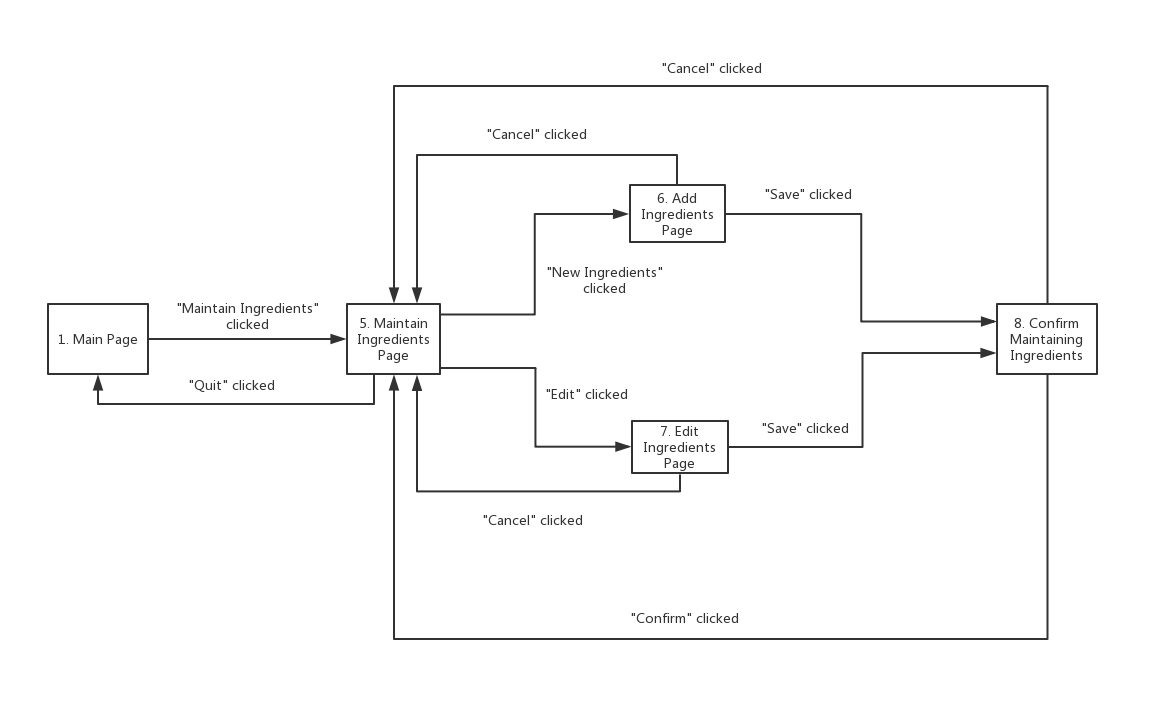
REQ-1:

REQ-2:

## Maintain Ingredients

3.2.1 Description and Priority

For the feature maintain ingredients, there is also the ability to add and update records for ingredients, which is also at a high priority.

3.2.2 Stimulus/Response Sequences

3.2.3 Functional Requirements

There is a confirm window for user to confirm the information about the add and update ingredients

# External Interface Requirements (team work, everyone contributes every section)

## User Interfaces

## Hardware Interfaces

This application is desktop-based and it has GUI displayed on the Monitor. User can use mouse and keyboard to input data.

## Software Interfaces

This software needs to connect to local database to record and load recipes, which will be decided and developed in the furture.

## Communications Interfaces

N/A

# Other Nonfunctional Requirements (team work, everyone contributes every section)

## Performance Requirements

TBD

## Safety Requirements

This software only works as a method for users to store and display their recipes. The software has no ability to check whether the recipe could bring harms to human. User will be solely-responsible for the context of their recipe.

## Security Requirements

The software will not share the recipe or any other user data without the permission of user. Data will be stored locally, so we will not provide any authentication methods for entering our software.

## Software Quality Attributes

TBD

# Other Requirements (team work, everyone contributes every section)

TBD

Appendix A: Glossary

N/A

Appendix B: Analysis Models

N/A

Appendix C: Issues List

TBD