

REVIEWED

By Simon at 2:33 pm, Oct 14, 2021

Little Bartender

Team 3

Rian Bennett, Alexander Walsh, Tyler Vo

Software Requirements Specification Document

95

Version: (1.0)

Date: (10/13/2021)

Table of Contents

1 Purpose	3
2 Scope	3
3 User characteristics	3
3.1 Key users	3
3.2 Secondary users	4
3.3 Unimportant users	4
4 Product perspective	4
4.1 System Context	4
4.2 User interfaces	5
4.3 Software interfaces	6
4.4 Hardware interfaces and Memory constraints	6
4.5 Deployment requirements	6
5 Assumptions and Dependencies	6
6 Specific requirements	6
6.1 System Functional Requirements	6
6.2 Logical Database Requirements	7
6.3 Software System Attributes	8
6.3.1 Usability	8
6.3.2 Performance	8
6.3.3 Reliability/Dependability	8
6.3.4 Security	8
6.3.5 Maintainability	8

1 Purpose

Everyone loves a tasty drink, and for a lot of adults, they love a tasty alcoholic drink. Sometimes people might prefer something simple out of a bottle or can, but other times they might want to shake it up a little and experiment with different combinations. *Little Bartender* allows users to access a variety of alcoholic mixed drink or cocktail recipes for them to explore and try, as well as providing an inventory of ingredients they may or may not have. In addition to this, this program allows users to create and input their own recipes. After buying supplies in varying bulk often leaves clients with ingredients without their favorite counterpart, we aim to help them find a new use for those ingredients that just don't get used enough. After a customer tells us what ingredients they currently have, all their data must then get cross referenced with our growing database that users themselves can also add-to.

2 Scope

Within this new program, named *Little Bartender*, along with providing the user with a plethora of recipes, it should also allow them to find specific recipes quickly by inputting its name and searching through the entire list until a match is found. Furthermore, *Little Bartender* provides an interface where users can input what ingredients they currently have, including amounts. The back end uses this data to compare it to a pre-existing database and find any recipes that include said ingredients. Lastly, this program should allow users to create their own recipes that will be added to the current list by inputting their own ingredients and serving sizes.

3 User characteristics

The goal of *Little Bartender* is to provide an easier way of accessing a list of mixed drink recipes, which could be narrowed down via having the user list which ingredients they already have in stock. Therefore, intended users are people that are at or above drinking age and are able to read and write in English.

3.1 Key users

The key users of this application would be bartenders or aspiring bartenders. Also, another key group of users could be bar owners who can use this application as a quick guide to help train their new bartenders to learn their drinks. Making it easier to train new staff, it would hopefully lead to bar owners supporting this development.

Key User Characteristics:

- **User role responsibilities:** The user will be responsible for using the most updated version of the program to get the most use out of it, as well as being responsible for providing an accurate count of their inventory of ingredients.
- **Subject matter experience:** Primary users include people whose knowledge range from having no experience in drink mixing to experts in this field.
- **Technological experience:** The user should be able to know how to search for or input data on a basic computer.
- **Other user characteristics:** The user must be at or above legal drinking age, as well as being able to read and write in English.

3.2 Secondary users

Secondary users are those individuals who aren't interested in becoming a bartender, nor are they bar owners, or anything related to alcoholic businesses. However, they can still be interested in learning about new drinks to make for themselves or others.

Secondary User Characteristics:

- **User role responsibilities:** Secondary users will use the most updated version of the program and will view recipes already provided by *Little Bartender*.
- **Subject matter experience:** These users may have beginner-level knowledge of drink mixing or no knowledge at all. They may know another person with greater experience.
- **Technological experience:** The user should be able to know how to search for or input data on a basic computer.
- **Other user characteristics:** The user must be at or above legal drinking age, as well as being able to read and write in English. In addition, pictures will be provided to give the user a better understanding of the drink and its ingredients.

3.3 Unimportant users

Unimportant users will include individuals who do not like making or drinking cocktails. They will infrequently use *Little Bartender* because they have no interest in viewing the list of mixed drinks.

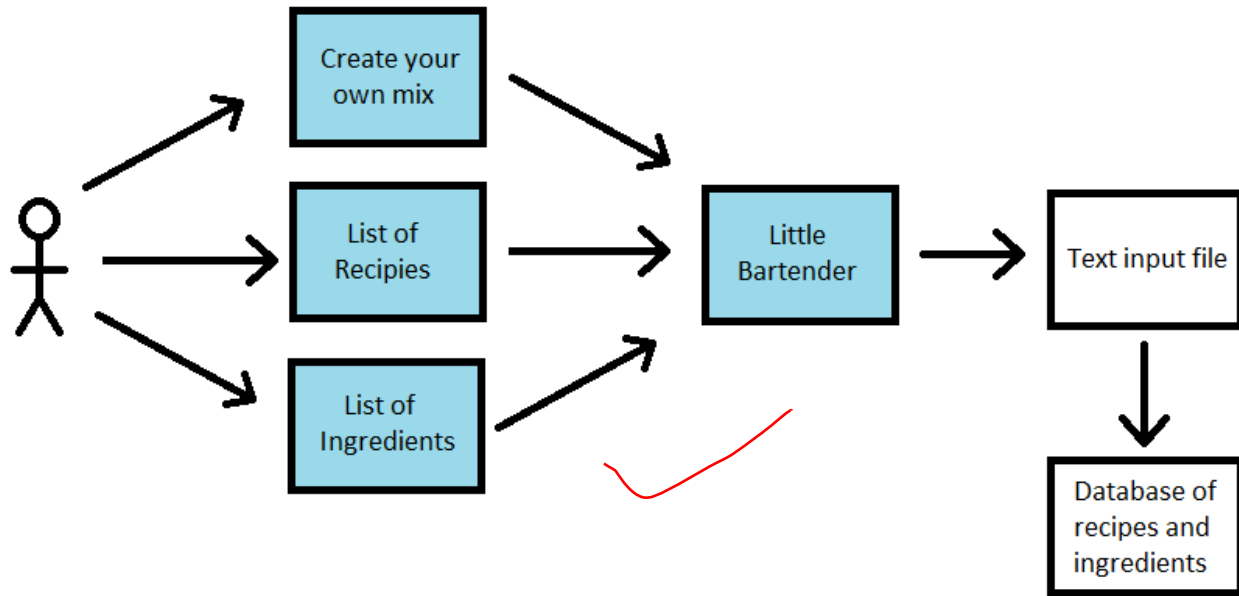
Unimportant User Characteristics:

- **User role responsibilities:** Unimportant users will have no responsibility in providing a list of ingredients or viewing pre-existing recipes.
- **Subject matter experience:** The user's experience with mixing drinks is presumed to be novice-level or non-existent.
- **Technological experience:** Technological experience may vary.
- **Other user characteristics:** The user may be below legal drinking age, preventing them from using the program. They might also not be fluent enough in English to use the program properly.

4 Product perspective

4.1 System Context

Little Bartender will have a UI that will allow a user to display a list of recipes, display a list of ingredients that the user may or may not have in stock, and an interface that would allow them to enter their own recipe with their own or pre-existing ingredients. For the base list of recipes and ingredients, the program will pull data from a text input file.



4.2 User interfaces

The user interface for **List of Recipes** requires the following:

1. It is required that the system neatly compiles the current list of recipes to be displayed to the user.
2. It is required that the system allows users to search for a specific recipe given an inputted keyword.
3. It is required that the system will display a picture of the specific recipe to the user.
4. It is required that the system will allow the user to modify the list of recipes, including deleting, at their own discretion.

The user interface for **List of Ingredients** requires the following:

1. It is required that all ingredients provided by *Little Bartender* are compiled neatly to be displayed to the user.
2. It is required that the system will allow the user to modify the list of ingredients, including adding and deleting, at their own discretion.

The user interface for **Create Your Own Mix** requires the following:

1. It is required that the system will allow the user to input a string of information when prompted, which fills in information regarding lists of ingredients, ABV percentages, and other related things.
2. It is required that the system will insert the created recipe into the current list, which will then be displayed along with it.

4.3 Software interfaces

Little Bartender is intended to be used with Java, and will thus need Java software installed on their computer.

4.4 Hardware interfaces and Memory constraints

~~*Little Bartender* is intended to be used on personal computers with either Windows or MacOS operating systems.~~

4.5 Deployment requirements

Little Barista is intended to be installed and run on both personal computers and smartphones running on a Java-compatible OS. Little Barista should be able to be run without an Internet connection.

5 Assumptions and Dependencies

- It is assumed that the users are able to legally drink and sell alcohol in their local jurisdictions.
- It is assumed that users have knowledge to interact with the interface, particularly through either a computer and mouse or through a touchscreen.
- It is assumed that users are able to read and follow directions.
- It is assumed that users understand the local alcohol laws in their local jurisdiction and take full responsibility for all actions when using the software.
- It is assumed that all users are aware of any dietary restrictions, allergies, or health issues that must be considered in recipe selection.
- It is assumed that the database is not tampered with through external means.

6 Specific requirements

6.1 System Functional Requirements

R.1.1.1 As a first time user, I want to be able to enter the ingredients that I have at home so that I will be able to know what drinks I will be able to make with them.

R.1.1.2 Given that I am a minor, when I enter my birthday, then the application should deny me access.

R.1.1.3 As a moderate experienced user, I want to be able to look up specific recipes by name so that I will not have to spend more time looking through a long list.

R.1.1.4 As an experienced user, I want to be able to add my own recipes into the database so I can refer to them when I need them later..

R.1.1.5 As a user, when I search for recipes, I want the recipes to be organized in ascending order of best fit so that I can select the recipe that fits the given parameters.

R.1.1.6 As a user, I want to have a record of my past recipe selections in a section labeled "history" so that I do not have to re-search recipes for which I regularly have the necessary ingredients.

R.1.1.7 As a user, I want to be able to set filters through a list of recipes based on keywords, so that I can narrow my recipes based on the my tastes or the tastes of others

R.1.1.8 As a user, I want to be able to add, change, or delete components that I own so that I can accurately reflect the components I own.

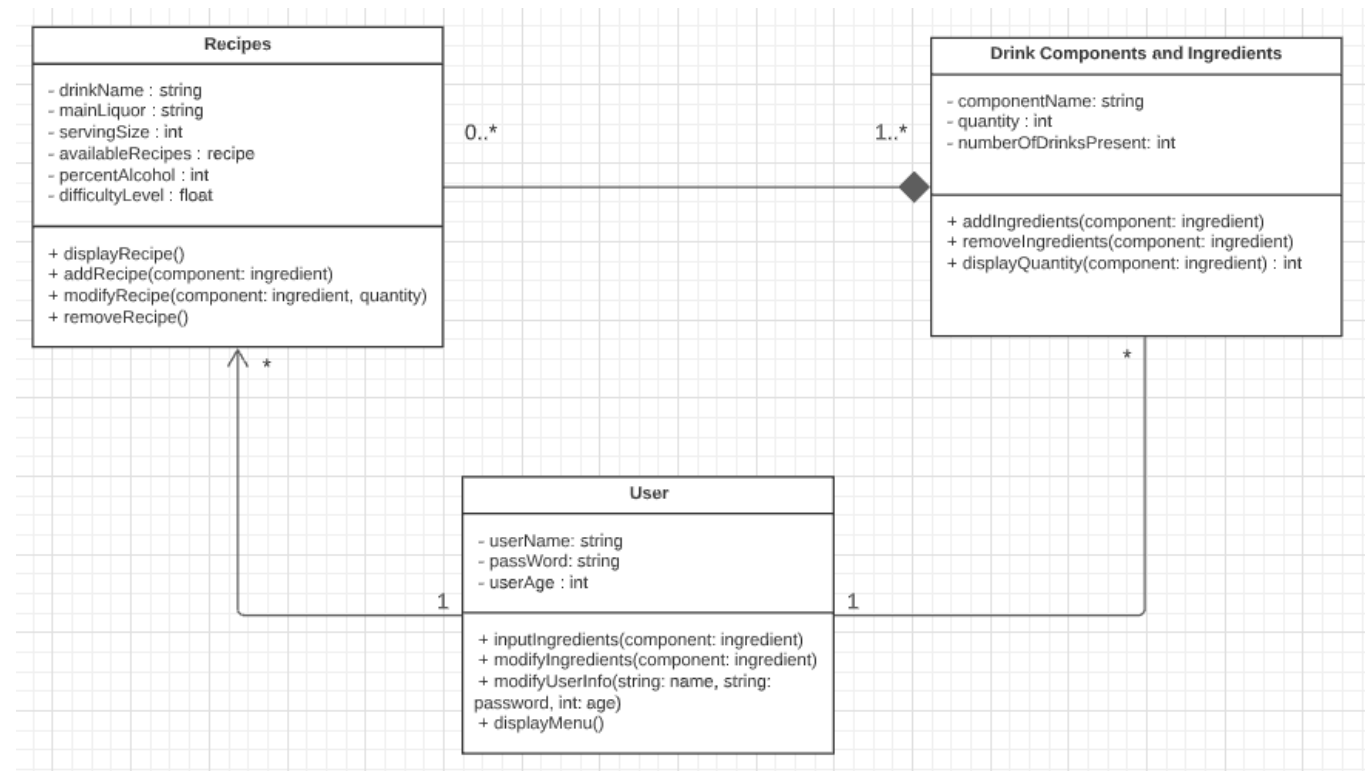
R.1.1.9 As a new user, I want the application to choose me a random drink so that I can have a drink to start learning about the basics

R.1.1.10 As a user, I want to be able to see the image of the finished drink so that I can compare my results with the expected result.

R.1.1.11 As a user, I want to view the difficulty of each individual drink so that I am able to expect the amount of work needed to create a specific drink.

R.1.1.12 As a user, I want to have a record of my marked drinks in a section labeled “favorite drinks” so that I can pull up my favorite recipes.

6.2 Logical Database Requirements



6.3 Software System Attributes

6.3.1 Usability

R3.1.1 The software system should be able to handle any amount of users at any given time.

R3.1.2 The system should have no down time except during planned maintenance, being accessible to users at all times.

R3.1.3 The system should not be used by anybody under the legal drinking age, and thus must not be under the age of 21.

R3.1.4 The software should be easily understood by both the average person, and people with advanced domain knowledge like bartenders.

R3.1.5 The user should be able to understand fluent English, as this is the primary language of the application.

6.3.2 Performance

R3.2.1 When the system displays the recipe list with pictures, they should load within 4 seconds.

R3.2.2 The system should generate suggestions within 5 seconds of submitting the available ingredients list.

R3.2.3 The system should not record or save any extremely personal information, like credit cards, in case of security breach.

R3.2.4 The software should be able to be run without an internet connection.

6.3.3 Reliability/Dependability

R3.3.1 The software should confirm or deny any entry of data into the database with an interface notification for the user.

R3.2.2 The system should notify the user of any incorrect information or unrecognized formats through an interface option.

R3.2.3 The system should not crash during any normal operations, and if it does, throw a message in the console or to the user as to the reason that it crashed.

6.3.4 Security

R3.4.1 The system should check the user's age and determine whether or not they are at or above drinking age.

R3.4.2 The system should verify user information such as their username and password.

6.3.5 Maintainability

R.3.5.1 Ensure that all product code is appropriately documented and commented, allowing avid explanation of each programming style.

R.3.5.2 Ensure that code is in the proper files and locations so that modularity is achievable on all executable files.

R.3.5.3 The database should store data in a simple, secure manner and shall be easy to maintain at all times.

R.3.5.4 Repeated code shall be turned into methods when appropriate.