Software Requirements Specification

for

Cooking Companion

Version 1.0 approved

Prepared by Cooking Code-panions

CCP Inc

February 19, 2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Tyler/Heath/Kass /Nick/Tristan	02/19/23	Initial document creation	1.0
Tyler/Heath/Kass /Nick/Tristan	03/03/23	Updated based on deliverable 1 feedback	1.1

1. Introduction

1.1 Purpose

Cooking Companion is a web application that enables people to cook, share, and save recipes with ease. The foundation of the application is a collection of recipes supplied by users. Users will be able to cultivate a list of personal recipes and search for new recipes by name, specific ingredients, or dietary restrictions. It will be able to produce a detailed shopping list for all ingredients and cookware needed to complete the recipe. Once the users have everything needed to begin cooking, the application will provide step-by-step instructions to guide the cooking process.

1.2 Document Conventions

Every requirement statement is to have its own priority. Standards and typographical conventions will be updated at a later time.

1.3 Intended Audience and Reading Suggestions

This document is intended for developers, marketing staff, testers, and documentation writers. This SRS contains information regarding the Cooking Companion web application. Including a product description that focuses on the product perspective, product functions, user classes/characteristics, operating environment, design/implementation constraints, user documentation, as well as assumptions and dependencies.

The reader is suggested to read this document by proceeding through the sections that are most pertinent to the individual.

1.4 Product Scope

Cooking Companion is a web application that enables people to cook, share, and save recipes with ease. The foundation of this application will be a collection of recipes supplied by users. Users will be able to cultivate a list of personal recipes. This will provide the user with a space to store their collection or recipes.

1.5 References

https://github.com/njacobs2019/Cooking-Code-panions

2. Overall Description

2.1 Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

Cooking Companion offers a range of features and capabilities comparable to those found on popular food websites, such as Allrecipes and Betty Crocker, as well as mobile applications like Whisk and Recipe Keeper. Cooking Companion is a stand alone product. It will function as a recipe database and sharing platform, implementing and improving many features present in its family-members.

2.2 Product Functions

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

Users are separated into two categories: registered users have signed into an account authenticated for the platform, and non-registered who have not.

Non-registered users will still have access to basic functionalities:

- Search publicly visible recipes
- Filter searches based on tags: ingredients, kitchenware, creator-specified adjectives.
- Receive step-by-step instructions
- Share recipes with a link to its page

Registered users will be able to use their account as a personal online recipe book, or to share their recipes with other users by making them publicly visible:

- Search publicly visible recipes
- Create recipe pages for storing a list of required ingredients, kitchenware, and instructions.
- Set recipe visibility to public or private
- Filter searches based on tags: ingredients, kitchenware, creator-specified adjectives.
- Receive step-by-step instructions
- Share recipes with a link

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

The application will differentiate between registered and unregistered users. Registered users will have access to the full functionality of the program, while non registered users will receive reduced functionality. See functionality outlined in Section 2.2 for permitted functionalities of these roles.

Developers will be permitted to deploy new features and change aspects of the user interface.

Administrators will have permission to view all pages, public and private, for customer service and maintenance purposes. They will also have the ability to alter or delete content, but remain compliant with U.S. privacy requirements.

2.4 Operating Environment

The Cooking Companion app will run on modern web browsers.

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

2.5 Design and Implementation Constraints

This section will be updated as we continue development. The development constraints of this application will be addressed as they become relevant.

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

2.6 User Documentation

https://github.com/njacobs2019/Cooking-Code-panions
https://docs.google.com/document/d/1Tv3_NzzIQjxdKin72MFRWbYPem2j1ovpuWA60yGnsEo/edit?usp=sharing

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

2.7 Assumptions and Dependencies

This section will be updated as we continue development. Assumptions and dependencies will be addressed as needed.

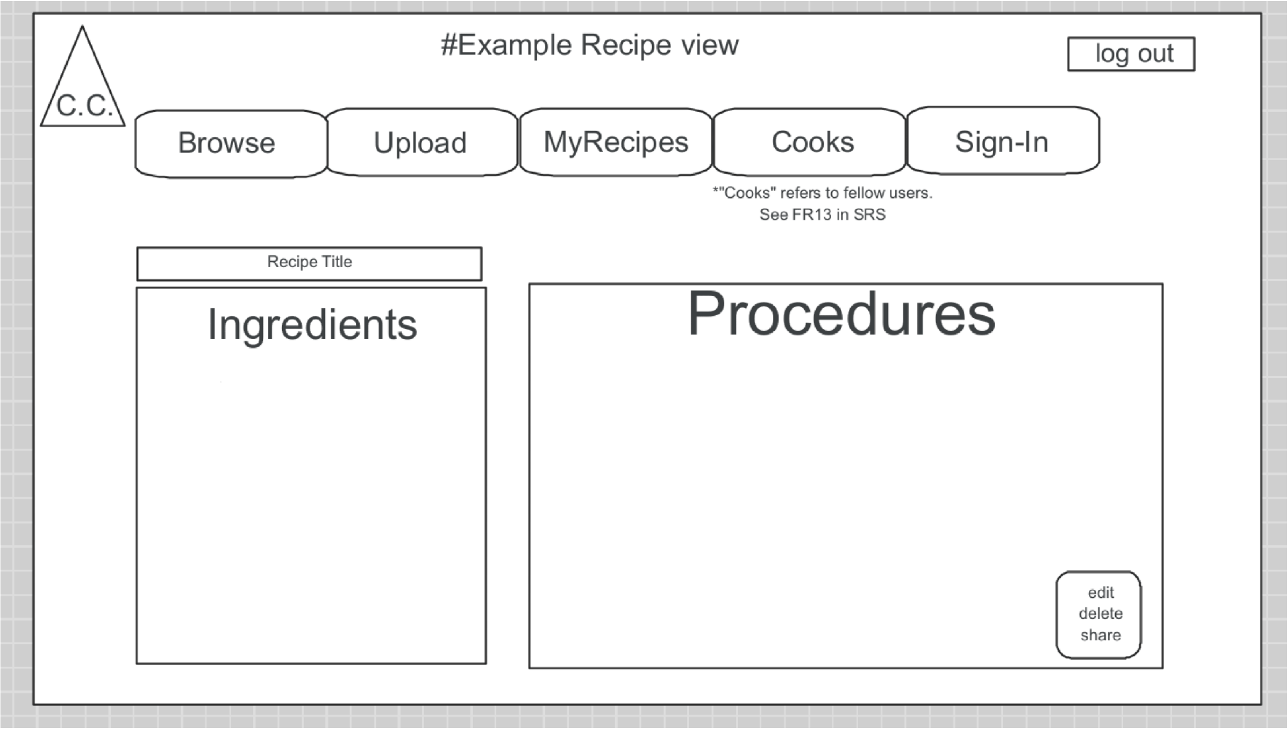
<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

3. External Interface Requirements

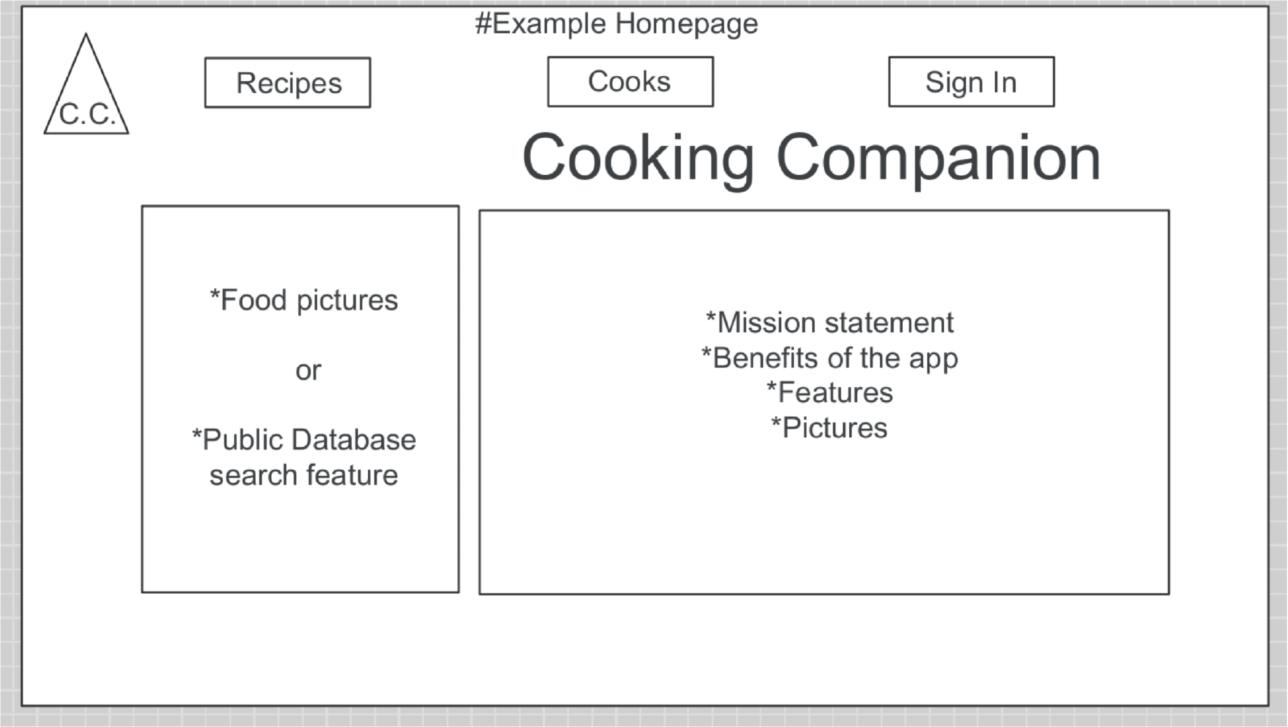
3.1 User Interfaces

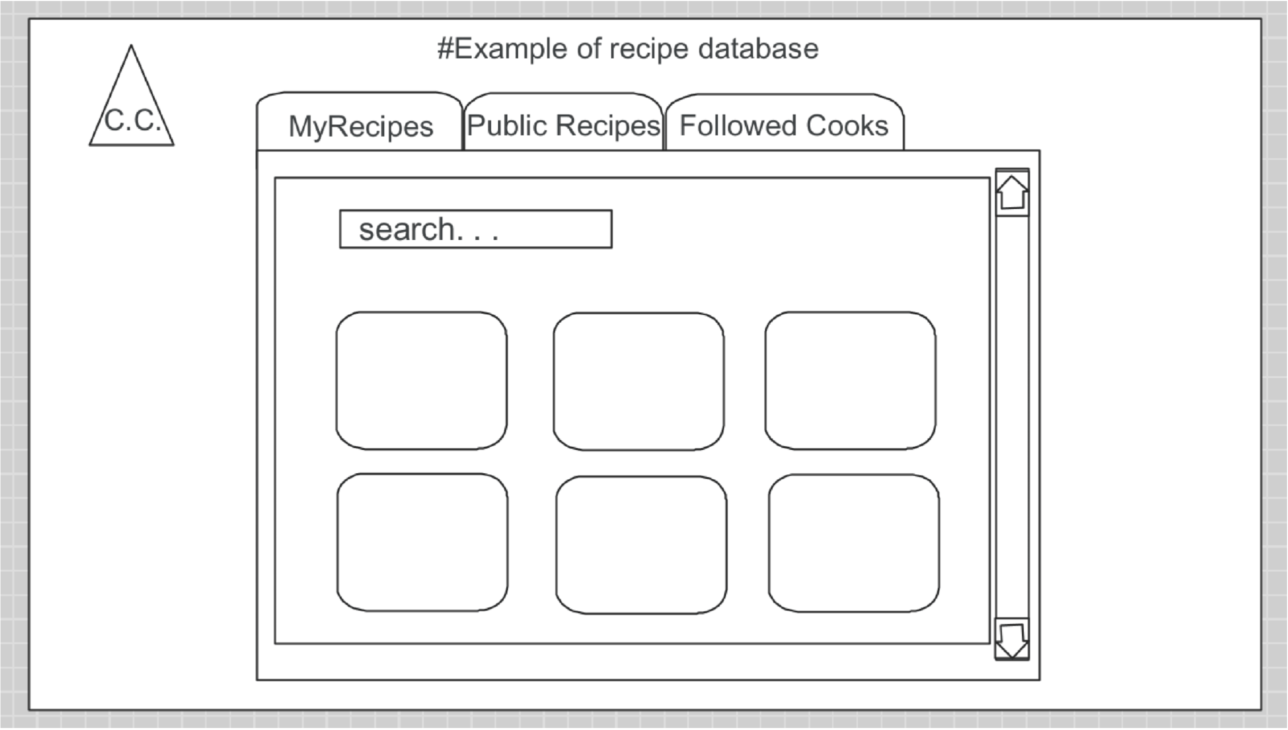
See user interface designs below:

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>



*"Cooks" refers to fellow users.
See FR13 in SRS


#Example Homepage
C.C.
Recipes
Cooks
Sign In
Cooking Companion
*Food pictures
or
*Public Database search feature
*Mission statement
*Benefits of the app
*Features
*Pictures


#Example of recipe database
C.C.
MyRecipes
Public Recipes
Followed Cooks
search. . .

4. Functional Requirements

1. The application shall allow the user to store recipes in a personal collection.
2. The application shall allow the user to view recipes in their personal collection.
3. The application shall allow the user to view recipes in the public collection.
4. The application shall allow the user to create an account.
5. The application shall allow the user to filter for only recipes with certain tags.
6. The application shall allow the user to omit recipes with certain tags.
7. The application shall allow the user to change the publicity setting of a recipe they have uploaded.
8. The application shall allow the user to edit recipes they have uploaded.
9. The application shall allow the user to log into their account.
10. The application shall allow the user to view the tags of a recipe.
11. The application shall allow the user to view a recipe step-by-step.
12. The application shall allow the user to share a link to a recipe.
13. The application shall allow the user to follow other users.
14. The application shall allow the user to delete a recipe they have uploaded.
15. The application shall allow the user to save public recipes to their personal collection.
16. The application shall allow the user to view the public recipes uploaded by a specific user.
17. The application shall allow the user to change their password.
18. The application shall allow the user to add images to their recipes.
19. The application shall allow the user to search a collection for keywords.
20. The application shall allow the user to log out of their account.
21. The application shall only allow the user who uploaded a recipe to edit it.

5. Other Nonfunctional Requirements

1. The application shall protect the database(s) from malicious attacks.
2. The application shall require the user to log in for complete functionality
3. The application shall add uploaded recipes to the collection(s) within 5 seconds 95% of the time.
4. The application shall not display private recipes publicly.
5. The application shall fetch and render a page within 5 seconds when there are less than 5000 simultaneous users 95% of the time.
6. The application shall respond to recipe posts within 5 seconds of the user submitting the request 95% of the time.
7. The application shall be capable of handling 1000 simultaneous requests 90% of the time.
8. The application shall comply with U.S. privacy requirements.
9. The application shall comply with the World Wide Web Consortium's Web Content Accessibility Guidelines.
10. The application shall be compatible with popular web browsers on desktop and mobile.