Software Requirements Specification

for

Recipe Buddy

Version 1.0 approved

Prepared by Andy Mahoney, Austin Dykeman, Michael Shea, Noah Lampron, & Tyler Neese

gOatmeal™

Oct. 6, 2019

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Tyler	9/16/19	Started a rough draft of the requirement document	1.0
Tyler	9/18/19	Added more to section one of the document	1.0.1
Tyler	9/23/19	Added some info to section 2 and added some descriptions of features in section 3	1.0.2
Tyler	9/23/19	Filled out parts of section 4 and started section 5	1.0.3
Austin	9/28/19	Filled out all of section 2	1.0.4
Tyler	9/30/19	Added more of the core features in section 3	1.0.5
Austin	10/05/19	Parts of 4, and 5 & Sections 7, and 9	1.0.6
Noah	10/06/19	Filled out section 8	1.0.7
Noah	12/14/19	Edits to Section 1, Section 3, and Section 4	1.0.8

1. Introduction

Purpose

Recipe Buddy is a website whose main function is to allow users to create, view, and follow other recipes on the website. The website will also be able to search through a database of recipes by tags indicating certain characteristics about the recipes and will also be able to store a grocery list of items that the users would need in order to make their recipes.

Document Conventions

The document is written in Arial font size 11. Sections are made bold in order to make them stand out. Sections are denoted by the specified number, followed by the bold section name. Subsections aren't given a number, but are also bolded.

Intended Audience and Reading Suggestions

This document is intended to be used by developers of the software as well as users and testers of the software. This SRS contains the standard format and includes a overall description of the software, requirements it needs to meet, key milestones in the development process as well as a documentation of the edits made along the way.

Section 2 covers an overall description of the Recipe Buddy software. This includes features of the product. This section contains valuable information for anybody interested in the Recipe Buddy software.

Section 3 covers more specific information on the functions described in section 2. This is a lower level view that explains interactions more. This would be interesting for developers/maintainers of the software in addition to anybody needing to look closer into the functionality of the software.

Section 4 covers hardware requirements. This could be helpful for the client or for people maintaining the code. Due to the limited scope of the project, and it's very limited software, there isn't much required. This makes it an easy read for anybody who is having trouble with the hardware.

Section 5 covers nonfunctional requirements of the software. This includes Performance/Safety/Security requirements. This is a helpful read for anybody on the clients side in addition to the developers/maintainers of the software.

Sections 6-10 Covers milestones/goals/timing of the project. This will be helpful for the developers to know the timetable of everything. This section is definitely not needed for regular users.

Project Scope

Recipe Buddy is a website that aims to improve the cooking experience of its users by making viewing and creating recipes fluid and easily accessible. Recipe Buddy attempts to seamlessly integrate viewing many other users recipes and following along with recipes a user makes themselves or finds online. Users should also be able to sort and navigate the website easily in order to find recipes that appeal to their specific needs.

References

Our Front-End will consist of HTML, CSS, HBS.

Our Back-End will consist of MongoDB, Express

To connect the front and back end we will be using NodeJS

2. Overall Description

Product Perspective

RecipeBuddy is a self-contained website that is free to use. It will be a brand-new website that is not a part of a larger project. Our goal is to provide users with the ability to find and use recipes. RecipeBuddy has a database to store recipes in.

Product Features

The main features and functionalities of RecipeBuddy are listed below:

- Adding a recipe. (Added recipes will enter the database for all to see).
- Editing a recipe.
- Deleting a recipe.
- Step-by-step instruction for cooking, called the Cook mode.
- Timer feature available during Cook mode, use is optional to user.
- Shopping list of ingredients.
- Ability to see all information on recipe.
- Upload pictures to recipes.
- Infinite Scrolling.
- Tag's are preloaded to recipes, and users have the ability to add their own tags.
- Sort through recipes by name, tag, ingredients, calories, and more.

Any of these features can be used in the website.

For more detailed information, see Part 3 of the document.

User Classes and Characteristics

RecipeBuddy anticipates a couple of different types of users:

1. General users who can use the software casually. These users want to find something to cook. The following features are what we believe these users will us the most:
    - Sort and search
    - Cook mode, with timer
    - Adding tags to recipes

2. Professional users who expect easy to use features. These users will add the most to the recipe list. The following features are what we believe these users will us the most:
    - Add, edit, and delete
    - Cook mode, with timer
    - Uploading pictures
    - Adding tags to recipes

The professional users are more favored because they will be adding to the recipe database, while casual users will be the bulk of the traffic for RecipeBuddy. As such, most features are tailored

toward the professional users. Though we expect not both groups to want ease of use for most features.

Operating Environment

RecipeBuddy is a website designed the desktop version of Firefox, specifically version 69.0.1. Since this is the case RecipeBuddy will work on most OS's, while the only hardware requirements are the ability to open and view websites. Also, a working gpu, to see images.

Design and Implementation Constraints

RecipeBuddy is written in html, CSS, and JavaScript for the website, and SQL for the database. The database will be used to store recipes.

User Documentation

There is no user documentation. RecipeBuddy is designed to be as simple to use as possible.

Assumptions and Dependencies

A running Firefox browser and connection to the internet.

3. System Features

Ability to Add Recipes

3.1.1 Description and Priority

This feature will allow users to add new recipes to the website. It is of High priority and is one of the required features of the finished website. This feature needs to be implemented for the website to function correctly, so it is one of the first features to add.

3.1.2 Stimulus/Response Sequences

The user will press button on the website which will allow them to create a new recipe on a new page. The website will give several blank boxes for the user to fill out, labeled with the category the user is filling out. Once the user is finished with their recipe, they will be able to push another button in order to upload the new recipe to the site.

3.1.3 Functional Requirements

There needs to be a button on the website that allows the user to publish a new recipe. It should bring them to another page that contains several fields detailing what information the user should enter into the fields in order to complete there recipes.

3.1.4 Notes

Add should be able to validate data with the back-end to make sure it is of the correct type. Users should be allowed to have a recipe with only a name and no information. They should be able to search by name, and by tag.
REQ-1: Add Recipes

Infinite Scrolling.

3.2.1 Description and Priority

This feature will allow the user to view new recipes infinitely by scrolling to the bottom of the page over and over again and continuing to have new recipes load at the bottom of the page. This is a feature of high importance and was specifically requested by the customer. This feature doesn't effect the success of any of the other features but it is still of high importance.

3.2.2 Stimulus/Response Sequences

The user would view all the recipes currently displayed by the screen by scrolling to the bottom of the page, then once the website sees that the user is at the bottom of the page, it will automatically load more recipes allowing the user to continue scrolling and viewing more recipes.

3.2.3 Functional Requirements

The website needs to be able to continuously detect if the user is at the bottom of the page or not and if they are, then automatically load more recipes. This might require the use of libraries in order to make this easier. The website needs to also retrieve these recipes from a database that stores all recipes entered on the site.

REQ-2: Infinite Scrolling

Ability to Delete Recipe

3.3.1 Description and Priority

This feature will allow the user to delete recipes that they have created. This feature is of high priority because it is required to make the website function smoothly. The users should be able to select a button that will allow them to delete recipes that they have created permanently from the database. This feature has a benefit of 9, a penalty of 1, a cost of 3 and a risk of 1.

3.3.2 Stimulus/Response Sequences

The user should be able to press a button that will allow them to delete the recipe they are currently viewing. Once they select the delete button there should be a pop-up window that confirms that they are sure they want to permanently delete the recipe.

3.3.3 Functional Requirements

There needs to be a button on the website that is clearly marked “Delete” and once pressed allows the user to delete a recipe permanently from the website. There should be a confirmation pop up box that allows users to confirm that they want to delete the selected recipe and once they have confirmed it, there must be code that allows the website to find the selected recipe in the database and permanently remove it.

REQ-3: Delete Function

Ability to Edit Recipe

3.4.1 Description and Priority

This feature should allow the user to edit all parts of an already created recipe then upload the changes to the database. The user should be able to change the descriptions of steps, add or delete new steps, as well as change timers in the recipe. This is a medium priority for the project.

3.4.2 Stimulus/Response Sequences

The user should be able to press a button on the website that allows them to edit a recipe. It should then show the user a outline of the recipe and allow them to edit the descriptions of each step as well as timers and tags on the recipe.

3.4.3 Functional Requirements

The website should have some form of gui that allows the user to see there completed recipe as well as alter the features of it. The website should then take the new data and add it to the database while removing the old data. The website needs to be able to add and remove data from an existing recipe in the database.

3.4.4 Notes

Should validate that there is at least the name of the recipe. All other fields may be blank or have valid input. They may not have invalid input for the database.

4. External Interface Requirements

User Interfaces

The website will have modern and minimalistic look to it. It will have the name RecipeBuddy at the top of the screen as well as a small toolbar that has the options to add, delete, view and edit recipes as well as access other features such as a virtual shopping cart built into the website. Below the toolbar, the website will display other users' recipes and allow the user to infinitely scroll and continue to view recipes. If the user clicks on a recipe it will pop out and show a textbox with a brief description of the recipe. Below each of the recipes there will be some small bubbles that are filled with tags that the users entered in order to describe their recipes to others.

Home Page - This will contain a button accessing the shopping cart, a search bar that self fills to the word 'Search'. Add recipe button that will allow the user to create a new recipe. It will also populate itself with all of the recipes that are currently found in the database in blocks. Each recipe block will have access to a cook button, which will bring you to the **Cook Page** an edit button which will bring you to the **Edit Page**, and a delete button that will delete the recipe. Each recipe will also have tags that can be found underneath the picture.

Add Page - Like all other pages, there will be a gOatmeal logo that will go to the **Home Page**, the cook button, edit button, and delete button that lead to their respective page. Add page will have a box to link an image file. It will also contain a field for recipe name, ingredient(Includes name, quantity and units). The ingredients section will also have an add ingredient button that adds a new empty ingredient group of fields. Underneath the ingredients section, we will have a cooking steps section. This section will includes a description for the step, and how long the step will take. It will of course also have a button to add a new step. Lastly, it will have a field where you can put in the tags for the recipe.

Edit Page - Very similar to the add page, with the exception that you are able to delete ingredients, and delete cooking steps.

Shopping Cart - The shopping cart will have the gOatmeal logo on the top just like every other page. In addition to that it will have an Add Ingredients button that should allow the user to add an ingredient to the shopping cart. There is also a delete button next to the ingredient being listed that allows it to be deleted. At the bottom of the page there should be a save cart button that saves the cart contents to the database.

Cook Page - The cook page will also have the logo on the top of the screen that takes you to the

Home Page. It will also have the shopping cart button that goes to the **Shopping Cart** page and the delete button that will delete the button. It will have a picture of the recipe on the left that displays from the URL given. In the ingredients column it will have each ingredient as listed in the recipe, with the form ('name', 'amount', 'calories'). Underneath the ingredients column there is a cooking steps column. This is visually very similar to the ingredients column except with a description of the step followed by how long the step should take to complete. At the end there will be a complete key that finishes the recipe and sends you back to the home page.

Hardware Interfaces

A working computer that can support the latest versions of Firefox browser.

Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

Communications Interfaces

This website should be designed to work on the Mozilla Firefox browser and does not need to send data to other websites. It should only communicate with the users browsers as well as its own database.

5. Other Nonfunctional Requirements

Performance Requirements

The user's machine must be able to run a modern website that does not contain any extremely hardware reliant software. The website should be accessible to everyday normal people that enjoy cooking and easily searching for new recipes. It should not be difficult or counter intuitive to navigate and should be execrable to people that are not tech savvy.

Safety Requirements

There are currently no safety requirements.

Security Requirements

There are currently no security requirements.

Software Quality Attributes

The website software should run smoothly without giving any reason for concern to the user which includes minimizing bugs and other issues. It should also be intuitive and easy to use for anyone even people that are not great at using computers. It should continue to run without needing to be monitored every day and should continue to allow users to add recipes and view new recipes in the future.

Business Rules

There is no need to restrict access to any functions. All users can currently use any function they desire.

6. Key Milestones

#	Milestone	Target Completion Date	Comments
1.	Front-End done	11/15	Front end not including connection to MongoDB
2.	Database complete	11/20	None of us have experience with the database so it will take a while
3.	Front-End connected to DB	12/10	Using HBS to connect MongoDB to the front-end. Also requires MongoDB knowledge so we are not well prepared to tackle this.

7. Key Resource Requirements

Major Project Activities	Skill/Expertise Required	Internal Resource	External Resource	Issues/Constraints
Add/Delete Features	HTML/CSS/HBS/MongoDB		Internet	For all major activities, MongoDB experience is required. Something our team entirely lacks.
Edit Feature	HTML/CSS/HBS/MongoDB		Internet	“
Cook Feature	HTML/CSS/HBS/MongoDB		Internet	
Upload Pictures	HTML/CSS/HBS/MongoDB		Internet	
Sort Recipes	HTML/CSS/HBS/MongoDB		Internet	
Shopping Cart	HTML/CSS/HBS/MongoDB		Internet	
Recipe Tags	HTML/CSS/HBS/MongoDB		Internet	
Search Feature	HTML/CSS/HBS/MongoDB		Internet	
Database	MongoDB	Noah is learning it	Internet	
Infinite Scrolling	HTML/CSS/HBS/MongoDB			

8. 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.>

A database for Recipe Buddy requires a server-side implementation that will store recipes and use information for the website. It will be constructed using MySQL. It should be able to store the following information

- **Recipes:** ID, Name, Steps, Ingredients, Calories, Time to cook, Date last accessed

The server will be running on a Linux platform. The interaction between the database and the website will be handled using PHP.

9. Requirement Change Management

Changes will be decided based on the customers demands, and when 3 or more team members make agree with a change.

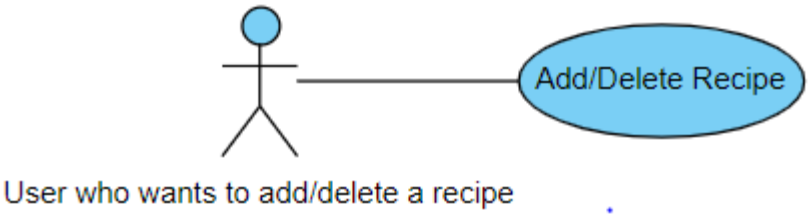
10. Restrictions, Limitations, and Constraints

There are no special restrictions, limitations, or constraints on the program that has not been previously discussed.

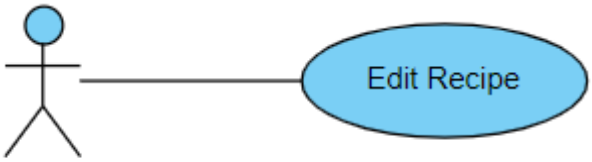
Appendix A: Glossary

<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.>

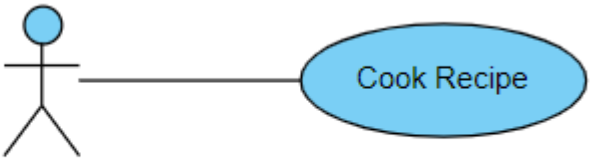
Appendix B: Analysis Models



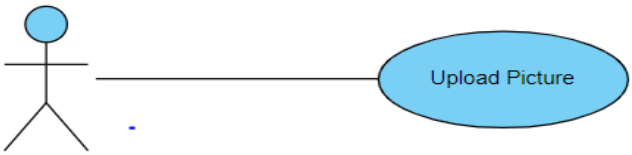
User who wants to add/delete a recipe



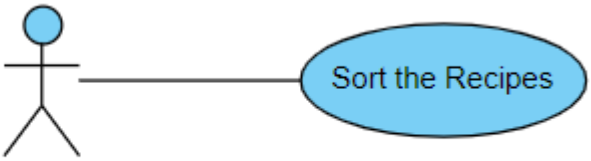
User who wants to edit a recipe



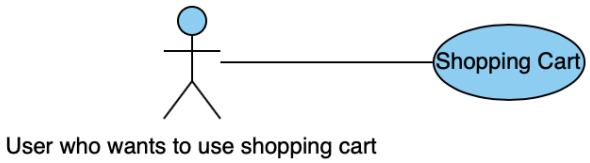
User who wants to Cook a recipe

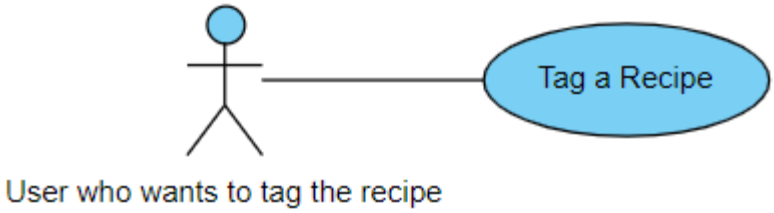


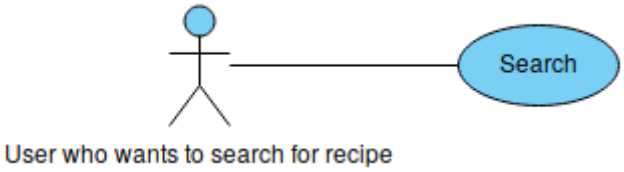
User who wants to upload a picture



User who wants to sort the recipes







Appendix C: Issues List

Still are having troubles with the Database. With no prior experience the group has to learn everything needed on our own time.

Appendix D: System Traceability Matrix

<A matrix that traces stated software requirements back to the system specification.>