

---

# SOFTWARE REQUIREMENTS SPECIFICATION

for

*RecipeBuddy*

Version 1.6

Prepared by Matthew Sprague, Brian Williams,  
Joseph Morrison, Jeffrey Rescignano

Certified Data Boys

December 16, 2021

# Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
1.1	Purpose . . . . .	6
1.2	Document Conventions . . . . .	6
1.2.1	Lists . . . . .	6
1.2.2	Revisions . . . . .	6
1.2.3	Styling . . . . .	6
1.2.4	Code . . . . .	7
1.3	Intended Audience and Reading Suggestions . . . . .	8
1.4	Project Scope . . . . .	8
1.5	References . . . . .	9
<b>2</b>	<b>Overall Description</b>	<b>10</b>
2.1	Product Perspective . . . . .	10
2.2	Product Functions . . . . .	10
2.3	User Classes and Characteristics . . . . .	11
2.4	Operating Environment . . . . .	12
2.5	Design and Implementation Constraints . . . . .	12
2.6	User Documentation . . . . .	15
2.7	Assumptions and Dependencies . . . . .	15
<b>3</b>	<b>External Interface Requirements</b>	<b>16</b>
3.1	User Interfaces . . . . .	16
3.2	Hardware Interfaces . . . . .	16
3.3	Software Interfaces . . . . .	16
3.4	Communications Interfaces . . . . .	16
<b>4</b>	<b>System Features</b>	<b>17</b>
4.1	Recipe: Recipe Creation . . . . .	17
4.1.1	Description and Priority . . . . .	17
4.1.2	Stimulus/Response Sequences . . . . .	18
4.1.3	Functional Requirements . . . . .	18
4.2	Recipe: Recipe Pages . . . . .	19
4.2.1	Description and Priority . . . . .	19
4.2.2	Stimulus/Response Sequences . . . . .	20
4.2.3	Functional Requirements . . . . .	20
4.3	Recipe: Timer . . . . .	21
4.3.1	Description and Priority . . . . .	21

4.3.2	Stimulus/Response Sequences . . . . .	22
4.3.3	Functional Requirements . . . . .	22
4.4	Recipe: Recipe Difficulty . . . . .	23
4.4.1	Description and Priority . . . . .	23
4.4.2	Stimulus/Response Sequences . . . . .	24
4.4.3	Functional Requirements . . . . .	24
4.5	Recipe: Recipe Rating . . . . .	25
4.5.1	Description and Priority . . . . .	25
4.5.2	Stimulus/Response Sequences . . . . .	26
4.5.3	Functional Requirements . . . . .	26
4.6	Recipe: Commenting . . . . .	27
4.6.1	Description and Priority . . . . .	27
4.6.2	Stimulus/Response Sequences . . . . .	28
4.6.3	Functional Requirements . . . . .	28
4.7	Recipe: Comment Rating . . . . .	29
4.7.1	Description and Priority . . . . .	29
4.7.2	Stimulus/Response Sequences . . . . .	30
4.7.3	Functional Requirements . . . . .	30
4.8	Profile: Profile . . . . .	31
4.8.1	Description and Priority . . . . .	31
4.8.2	Stimulus/Response Sequences . . . . .	32
4.8.3	Functional Requirements . . . . .	32
4.9	Profile: Pantry . . . . .	33
4.9.1	Description and Priority . . . . .	33
4.9.2	Stimulus/Response Sequences . . . . .	34
4.9.3	Functional Requirements . . . . .	34
4.10	Profile: Tools . . . . .	35
4.10.1	Description and Priority . . . . .	35
4.10.2	Stimulus/Response Sequences . . . . .	36
4.10.3	Functional Requirements . . . . .	36
4.11	Search: Search . . . . .	37
4.11.1	Description and Priority . . . . .	37
4.11.2	Stimulus/Response Sequences . . . . .	38
4.11.3	Functional Requirements . . . . .	38
4.12	Search: Flags . . . . .	39
4.12.1	Description and Priority . . . . .	39
4.12.2	Stimulus/Response Sequences . . . . .	40
4.12.3	Functional Requirements . . . . .	40
4.13	Security: Account Creation . . . . .	41
4.13.1	Description and Priority . . . . .	41
4.13.2	Stimulus/Response Sequences . . . . .	42
4.13.3	Functional Requirements . . . . .	42
4.14	Security: Login . . . . .	43
4.14.1	Description and Priority . . . . .	43

4.14.2	Stimulus/Response Sequences . . . . .	44
4.14.3	Functional Requirements . . . . .	44
<b>5</b>	<b>Other Nonfunctional Requirements</b>	<b>45</b>
5.1	Performance Requirements . . . . .	45
5.2	Safety Requirements . . . . .	45
5.3	Security Requirements . . . . .	45
5.4	Software Quality Attributes . . . . .	45
5.5	Business Rules . . . . .	46
5.6	Appendix A: Glossary . . . . .	46

# Revision History

Name	Date	Reason For Changes	Version
All team members	Oct 8 2021	Initial doc	1.0
Matthew Sprague	Oct 20 2021	Updated dependencies & glossary	1.1
Matthew Sprague	Oct 28 2021	Updated dependencies	1.2
Brian Williams	November 5 2021	Update requirement specification	1.3
Brian Williams	November 17 2021	Update requirement specification	1.4
Brian Williams	December 5 2021	Update Update stimulus and response	1.5
Brian Williams	December 12 2021	Update requirement description	1.6
Brian Williams	December 15 2021	Finalized Documentation	1.6

# 1 Introduction

## 1.1 Purpose

Recipe Buddy is a modern recipe sharing website intended to facilitate connections between our users with food and culinary arts. It creates a personalized experience for the user, showcases popular and trending recipes, and creates a community through various recipe sharing services.

## 1.2 Document Conventions

The SRS document has the following conventions:

### 1.2.1 Lists

Lists and sublists should be formatted as such:

- This is an entry on a list!
- This is another entry on a list.
  - This is an entry on a nested list!
  - This is another entry on a nested list.
    - \* And this is an item on a doubly nested list.

Lists should never be nested more than twice. The list in this section shows how far we allow our lists to go. Anything more than that is not allowed.

### 1.2.2 Revisions

After a new revision, it is important to log your progress. Please update this document's version number and update this document's Revision History table.

### 1.2.3 Styling

When updating this document, adhere to the following style guidelines:

- The first paragraph of each section or subsection does not need to be indented. Subsequent paragraphs should be indented. In LaTeX, that is done by default by typing two blank spaces “ ” after a paragraph.

- This document should be typed using the default LaTeX font and styling.
- When writing numbers within a paragraph:
  - Whole numbers between zero and fifteen should be typed alphaetically.
  - Whole numbers equal to or above 16 should be typed numerically.
  - Decimal numbers, such as 3.14, should be typed numerically.
  - Fractions should be typed in the form  $\frac{1}{2}$ . In LaTeX, this can be done using the following code:  
`{\$ \frac{1}{2} $}`

#### 1.2.4 Code

When writing code within a paragraph:

- For short snippets, it is okay to use inline code. For example:  
 The Python command `print( ... )` writes to standard output.  
 In LaTeX, this is done using the `\texttt{ }` command in-line with the rest of a paragraph.
- For longer snippets, please give the code its own block. For example:  
 The following Python function will calculate the area of a circle:

```
import math

// This function takes in the radius of a circle
// and returns the area of that circle.
def area_circle(radius):

    r_squared = radius ** 2
    return math.pi * r_squared
```

This can be done by using a `verbatim` block in LaTeX.

Please remember that this is a requirements document, and not the system's documentation! Remember to only use code in this document when it is needed! That is,

- Write code that clearly illustrates what you are doing.
- Write code that is concise.
- Explain what your code is doing such that a reader with minimal technical skill can understand what the code is doing.
- If the code takes too long to explain, save it for the system's documentation.
- Try not to take up more than half of a page with a block of code.

- Never take up more than an entire page with a block of code.
- Aim to have minimal overlap between the code in this document and the code in our system’s documentation.
- The code in this document must convey meaningful information to all readers. It should be useful for developers, as well as non-developers.

## 1.3 Intended Audience and Reading Suggestions

This document is aimed at a wide audience:

- Recipe Buddy developers should be able to go over this document to understand how the system functions. Developers should have the ability to use this in conjunction with the system’s documentation to gain a comprehensive understanding of the system’s underlying features and overall design. Developers should also be able to use this document to understand who to design this system for and why the system is designed the way it is. That way, with each new proposed update to Recipe Buddy, developers can assess if it would align with the system’s design philosophy.
- Recipe Buddy designers and clients should be able to go over this document to understand how the system is designed. With any proposed updates or revisions to the system, designers should have the ability to refer to this document and ensure such changes align with Recipe Buddy’s design philosophy.

## 1.4 Project Scope

Recipe Buddy is a tool that allows registered users to organize their own recipies, view recipies of other registered users, and keep a catalog of food, kitchenware and appliances that they own. Recipe Buddy can also track dietary restrictions to offer a better user experince. Recipe Buddy organizes recipies by:

- Allowing users to create a recipe.
- Allowing users to modify their recipies.
- Allowing users to delete their recipies.
- Allowing users to set the visibility of their recipies (i.e. public or private)
- Allowing users to share private recipies with specific users.
- Allowing users to offer suggestions to other’s recipies.



Recipe Buddy keeps a catalog of food and kitchenware items by:

- Allowing for a detailed list of food items available in a users pantry.
- Allowing for a user to add all items from a particular recipe into their pantry.
- Allowing for users to keep a detailed list of kitchenware that they own.
- Allowing users to edit or modify these lists at any time from the user menu.

Recipe Buddy tracks dietary restrictions by:

- Allowing users to keep a detailed list of specific ingredients marked as allergens.
- Allowing users to keep a detailed list of restricted categories. (i.e. dairy for lactose intolerant or bread for gluten-free).

## 1.5 References

Throughout this document, multiple web addresses are referred to in order to give a better description of the Recipe Buddy software.

- Reference 1
  - Name: Chrome Browser System Requirements
  - Author: Google
  - Version: 95.0.4638.32
  - Access Date: September 29, 2021
  - Location: support.google.com
- Reference 2
  - Name: Firefox Browser System Requirements
  - Author: Mozilla
  - Version: 92.0
  - Access Date: October 4, 2021
  - Location: www.mozilla.org

## 2 Overall Description

### 2.1 Product Perspective

Recipe Buddy is a multi-functional tool for organizing recipes, pantry items and dietary restrictions. Although several recipe websites already exist, Recipe Buddy offers many powerful customizations and search features that separate it from the competition.

### 2.2 Product Functions

The Recipe Buddy software offers:

- A limited publicly-accessible database of Recipes.
- User registration to access full features of the website.
- The organization of recipes by:
  - Allowing users to create a recipe.
  - Allowing users to modify their recipes.
  - Allowing users to delete their recipes.
  - Allowing users to set the visibility of their recipes (i.e. public or private)
  - Allowing users to share private recipes with specific users.
  - Allowing users to offer suggestions to other's recipes.
- The organization of pantry items and cookware by:
  - Allowing for a detailed list of food items available in a user's pantry.
  - Allowing for a user to add all items from a particular recipe into their pantry.
  - Allowing for users to keep a detailed list of kitchenware that they own.
  - Allowing users to edit or modify these lists at any time from the user menu.
- The tracking of dietary restrictions by:
  - Allowing users to keep a detailed list of specific ingredients marked as allergens.
  - Allowing users to keep a detailed list of restricted categories. (i.e. dairy for lactose intolerant or bread for gluten-free).
- Powerful search features to sort and filter:

- By pantry and cookware items already available to the user.
- By user-specified dietary restrictions.
- By popularity, rating or difficulty.
- Social features such as:
  - Private sharing of a recipe.

## 2.3 User Classes and Characteristics

Guest:

- A guest is defined as someone browsing the Recipe Buddy website, but have not created a user account or signed in.
- Guests have limited access to the resources available on Recipe Buddy.
  - Guests will not be able to access various parts of the software tutorial videos.
  - Guests are not allowed to cook recipes.
  - Guests are not allowed to store user-specific data such as dietary restrictions, favorited recipes, pantry items or cookware items.
  - Guests cannot leave comments or reviews on recipies.
  - Guests cannot upload or contribute recipes to the Recipe Buddy database.
  - Guests cannot access powerful search features.
  - Guests cannot view private recipes shared with them.

User:

- A user is defined as someone browsing or contributing to the Recipe Buddy website, who has registred a user account ans signed in.
- Users have full access to the resources available on Recipe Buddy.
  - Users can view the full video tutorial on how to access Recipe Buddy features.
  - Users can cook recipes.
  - Users can store user-specific data such as dietary restrictions, favorited recipes, pantry items or cookware items.
  - Users can leave comments and reviews on recipies.
  - Users can upload and contribute recipes to the Recipe Buddy database.
  - Users can access powerful search features.
  - Users can access private recipies shared with them.

## 2.4 Operating Environment

Although Recipe Buddy is expected to work on modern browsers, the Recipe Buddy client will only be tested extensively to ensure proper functionality across the top three major web browsers by marketshare including:

- Google Chrome (Desktop)
- Apple Safari (Desktop)
- Mozilla Firefox (Desktop)

## 2.5 Design and Implementation Constraints

Recipe Buddy shall abide by the following legal constraints:

- Children’s Online Privacy Protection Act (COPPA)
  - Recipe Buddy shall not allow children, as defined by COPPA, to register for the service.
  - COPPA defines a child as anyone under the age of 13.
- California Consumer Privacy Act (CCPA)
  - Recipe Buddy will allow users to query their personal information, as collected by Recipe Buddy.
  - Recipe Buddy will allow users to delete their account and all of the information attached to it.
  - Recipe Buddy cannot sell a user’s information.
  - Recipe Buddy cannot treat a user exercising their CCPA rights any different from other users.

Recipe Buddy shall also abide by the following technical constraints:

- Recipe Buddy shall be written with the following programming languages and frameworks:
  - Python
    - \* Version 3.9.7 or newer
    - \* Webpage: [www.python.org](http://www.python.org)
    - \* Documentation: [docs.python.org](http://docs.python.org)
    - \* Download: [www.python.org](http://www.python.org)
  - MariaDB
    - \* Version 10.6.4 or newer
    - \* Webpage: [mariadb.org](http://mariadb.org)

- \* Documentation: [mariadb.org](https://mariadb.org)
  - \* Download: [mariadb.org](https://mariadb.org)
- HTML5
  - \* Documentation: [developer.mozilla.org](https://developer.mozilla.org)
- CSS3
  - \* Documentation: [developer.mozilla.org](https://developer.mozilla.org)
- JavaScript
  - \* Specified by ECMAScript 2021
  - \* JavaScript documentation: [developer.mozilla.org](https://developer.mozilla.org)
  - \* ECMAScript 2021 documentation: [www.ecma-international.org](https://www.ecma-international.org)
- Docker
  - \* Version 20.10.9 or newer
  - \* Website: [docker.com](https://docker.com)
  - \* Documentation: [docs.docker.com](https://docs.docker.com)
  - \* Source code: [github.com](https://github.com)
  - \* Download: [docs.docker.com](https://docs.docker.com)
- Docker Compose
  - \* Version 2.0.1 or newer
  - \* Documentation: [docs.docker.com](https://docs.docker.com)
  - \* Source code: [github.com](https://github.com)
  - \* Download: [docs.docker.com](https://docs.docker.com)
- MariaDB Connector/Python
  - \* Version 1.0.7 or newer
  - \* Documentation: [mariadb-corporation.github.io](https://mariadb-corporation.github.io)
  - \* Source code: [github.com](https://github.com)
- Flask
  - \* Version 2.0.2 or newer
  - \* Webpage: [flask.palletsprojects.com](https://flask.palletsprojects.com)
  - \* Documentation: [flask.palletsprojects.com](https://flask.palletsprojects.com)
  - \* Source code: [github.com](https://github.com)
- Flask-SQLAlchemy
  - \* Version 2.5.1 or newer
  - \* Webpage: [flask-sqlalchemy.palletsprojects.com](https://flask-sqlalchemy.palletsprojects.com)

- \* Documentation: [flask-sqlalchemy.palletsprojects.com](http://flask-sqlalchemy.palletsprojects.com)
  - \* Source code: [github.com](https://github.com)
- Flask-login
  - \* Version 0.5.0 or newer
  - \* Documentation: [flask-login.readthedocs.io](http://flask-login.readthedocs.io)
  - \* Source code: [github.com](https://github.com)
- Flask-WTF
  - \* Version 0.15.1 or newer
  - \* Documentation: [flask-wtf.readthedocs.io](http://flask-wtf.readthedocs.io)
  - \* Source code: [github.com](https://github.com)
- python-email-validator
  - \* Version 1.1.3 or newer
  - \* Source code: [github.com](https://github.com)
- pdoc
  - \* Version 8.0.0 or newer
  - \* Webpage: [pdoc.dev](http://pdoc.dev)
  - \* Documentation: [pdoc.dev](http://pdoc.dev)
  - \* Source code: [github.com](https://github.com)
- pyyaml
  - \* Version 6.0 or newer
  - \* Webpage: [pyyaml.org](http://pyyaml.org)
  - \* Documentation: [pyyaml.org](http://pyyaml.org)
  - \* Source code: [github.com](https://github.com)
- libyaml
  - \* Version 0.2.5 or newer
  - \* Webpage: [pyyaml.org](http://pyyaml.org)
  - \* Documentation: [libyaml.docsforge.com](http://libyaml.docsforge.com)
  - \* Source code: [github.com](https://github.com)
- Bootstrap
  - \* Version 5.1 or newer
  - \* Webpage: [getbootstrap.com](http://getbootstrap.com)
  - \* Documentation: [getbootstrap.com](http://getbootstrap.com)
  - \* Source code: [github.com](https://github.com)

\* Download: [getbootstrap.com](http://getbootstrap.com)

- Recipe Buddy will interface with its MariaDB database using a RESTful API and SQL code.
  - RESTful API specifications: [restfulapi.net](http://restfulapi.net)
- Recipe Buddy should be designed to save recipe data very quickly.
  - Recipe data should be locally cached as it is entered by the user.
  - Recipe Buddy will provide a button to save recipe data on the edit recipe page.
  - When the user decides to save a recipe, recipe data should be serialized in no more than 0.5 seconds on a modern computer.
  - Immediately following serialization, the recipe data should be uploaded to the database.
  - When a recipe is saved, the user should be notified. This should occur no longer than 5 seconds after the user clicks the button to save their recipe, assuming that the user has a stable connection and is using a modern computer.
- The data stored by Recipe Buddy must abide by the following constraints:
  - Uploaded images must be stored in .jpg, .png, or .webp file formats.
  - Uploaded videos must be stored in .mp4, .mov, or .webm file formats.
  - Uploaded files must be no more than 4MB in size.

## 2.6 User Documentation

Recipe Buddy will feature a web-based video tutorial that will help website users and visitors explore Recipe Buddy features. The Recipe Buddy software also features a support page with multiple FAQ questions and how-to tutorials.

## 2.7 Assumptions and Dependencies

Recipe Buddy is designed under the following assumptions:

- Users will not try to use Recipe Buddy on an unsupported browser.
- All specified languages and frameworks are fully up to date and working on Recipe Buddy's host server.
- A server hosting Recipe Buddy must have reliable internet connectivity, and it must have enough space to store Recipe Buddy's database.
- A user signing up for Recipe Buddy will have an email with which to create their account.

## 3 External Interface Requirements

### 3.1 User Interfaces

Buttons included on every page:

- Home
- Search
- Profile
- New Recipe

### 3.2 Hardware Interfaces

The Recipe Buddy client requires a modern web browser to access.

Please refer to the web browser's minimum specifications in order to get an idea of the required hardware.

- Google Chrome Web Browser System Requirements
- Apple Safari Web Browser System Requirements
  - Apple Safari is pre-installed on all Apple Computers by default. Minimum system requirements include a modern macOS computer system.
- Mozilla Firefox System Requirements

### 3.3 Software Interfaces

The Recipe Buddy client software relies on a web browser in order to properly fetch and display data within the Recipe Buddy system. Web browsers are not operating system restricted as they run on all major operating systems.

### 3.4 Communications Interfaces

The Recipe Buddy client software interface runs in a web browser and communicates via the HTTPs protocol. An internet connection of speed 2.5Mbps - 5Mbps is required to use the full features of Recipe Buddy such as the video tutorial and embedded videos within recipes.



## 4 System Features

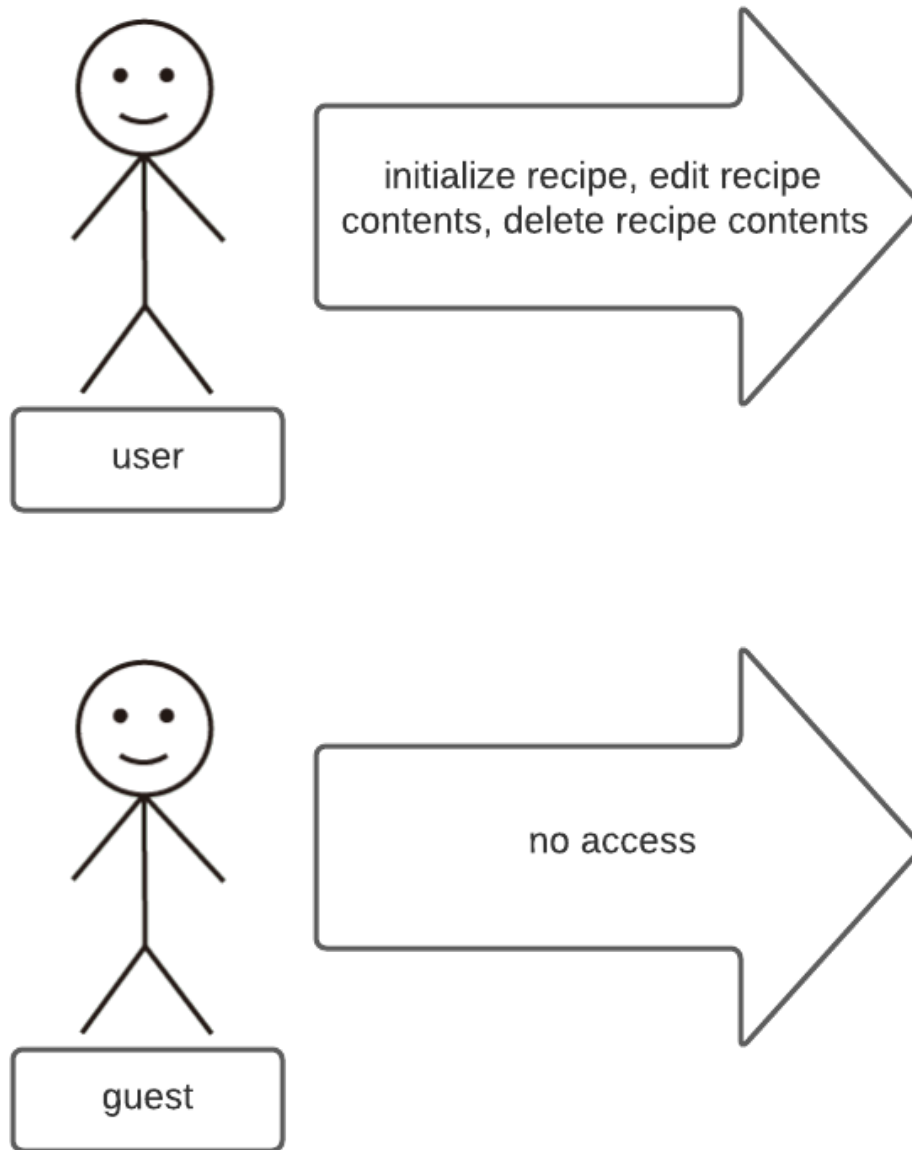
### 4.1 Recipe: Recipe Creation

#### 4.1.1 Description and Priority

The recipe creation feature allows registered users to upload recipes to the Recipe Buddy recipe database. Various attributes about the recipe such as ingredients, instructions, videos, images, cook time, cook temperature can be set by the user.

Critical Level	High
Dependencies	N/A
Estimated Time	1 week - 2 weeks
Lead Developers	Matthew Sprague, Jeffrey Rescignano
Testing	Client and focus group frontend testing, unit testing for backend
Risks	None

#### 4.1.2 Stimulus/Response Sequences



#### 4.1.3 Functional Requirements

- REQ-1.1.1: Edit various elements of the recipe:
  - Ingredients
  - Time

- Temperature
- Pictures
- Videos
- Instructions
- REQ-1.1.2: Delete the selected recipe.
- REQ-1.1.3: Share the selected recipe:
  - Private sharing
  - Public sharing

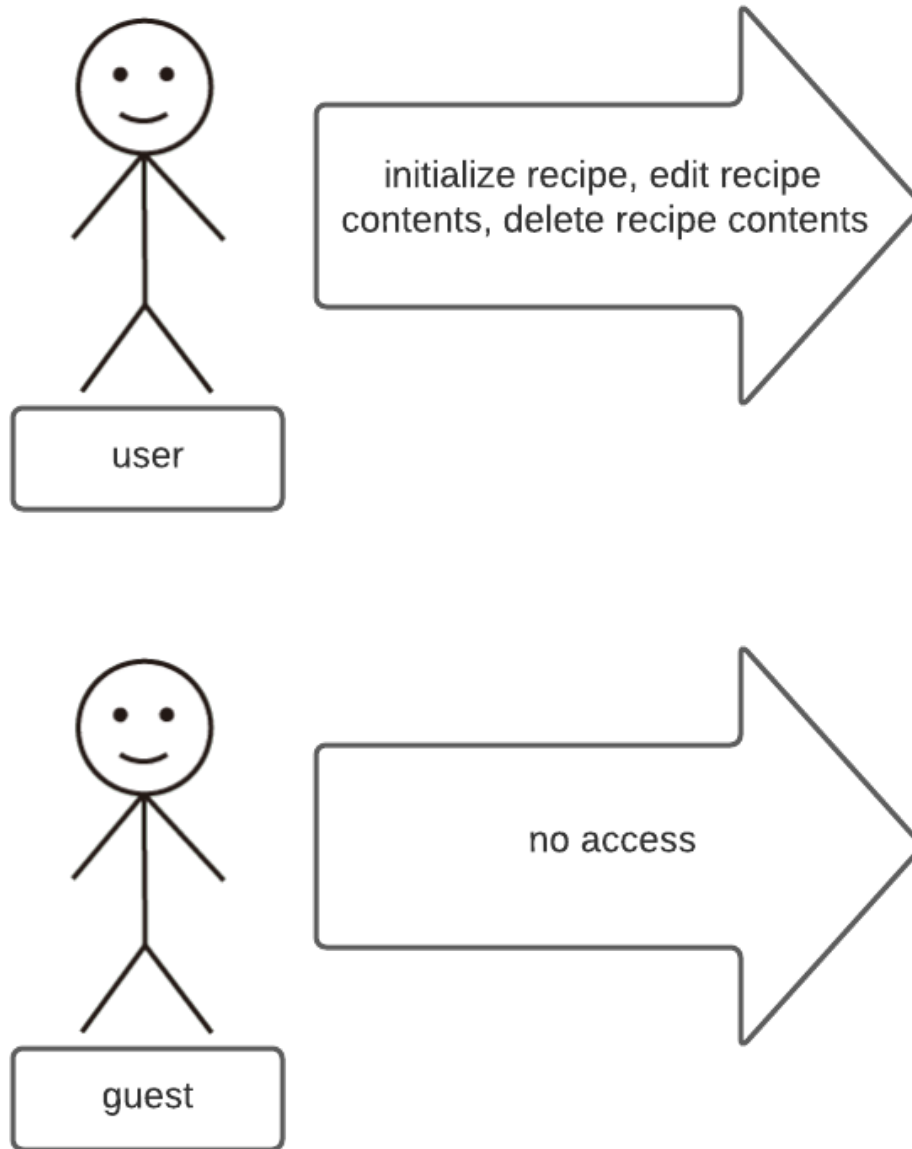
## 4.2 Recipe: Recipe Pages

The recipe pages feature allows both guests and registered users to view recipe information from the Recipe Buddy recipe database. Guests can only view public recipes, while registered users can view public recipes, their own private recipes and other private recipes shared to them by other users.

### 4.2.1 Description and Priority

Critical Level	High
Dependencies	Recipe Creation
Estimated Time	3 days - 1 week
Lead Developers	Matthew Sprague, Joseph Morrison, Jeffrey Rescignano
Testing	Client and focus group frontend testing, unit testing for backend
Risks	None

#### 4.2.2 Stimulus/Response Sequences



#### 4.2.3 Functional Requirements

- REQ-1.2.1: View publicly available recipes and all relevant elements.
- REQ-1.2.2: Comment on recipe pages.

- REQ-1.2.3: Rate the quality of recipe pages.
- REQ-1.2.4: Rate comments.

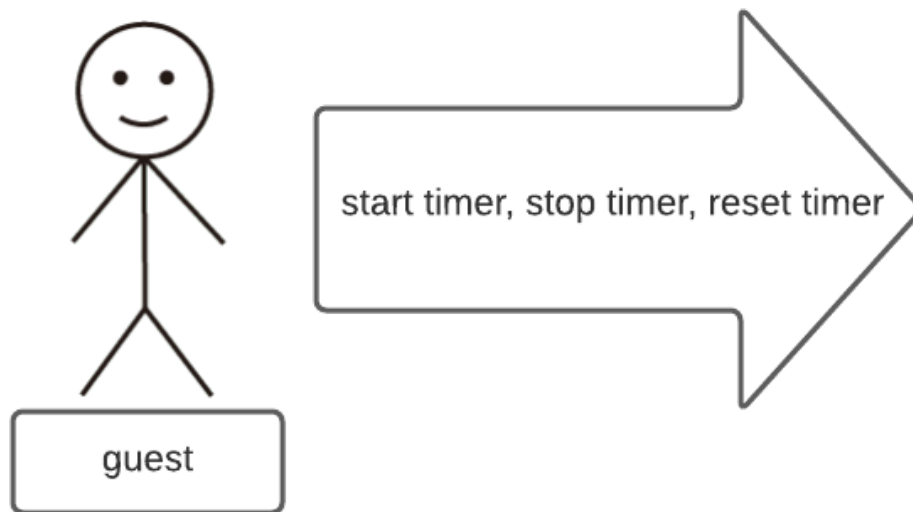
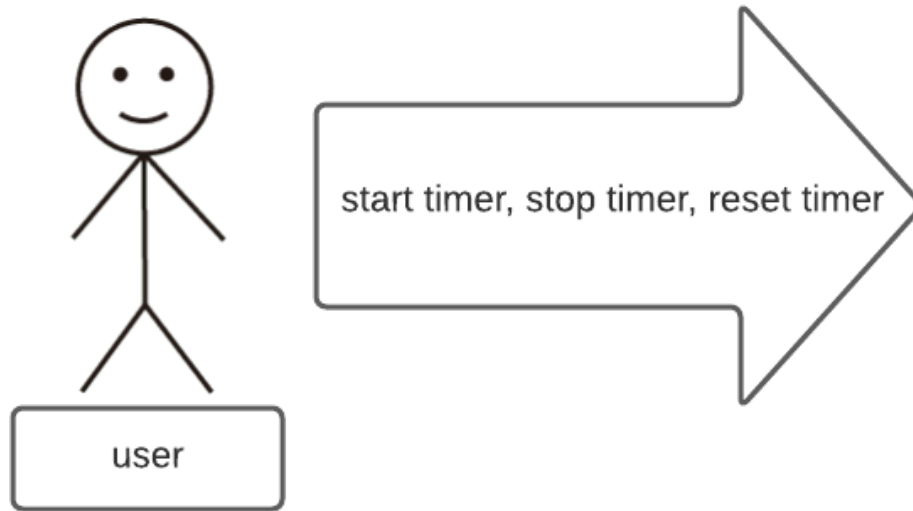
## 4.3 Recipe: Timer

### 4.3.1 Description and Priority

The timer feature allows registered users to access a timer on the recipe page that corresponds with the cook time of the recipe. For example, if a particular recipe takes 30 minutes to cook, a timer will appear near that step allowing the user to have easy access to a timer preset to 30 minutes.

Critical Level	Medium
Dependencies	Recipe Pages
Estimated Time	1 day - 3 days
Lead Developer Jeff Rescignano	
Testing	Client and focus group frontend testing, unit testing for backend
Risks	None

### 4.3.2 Stimulus/Response Sequences



### 4.3.3 Functional Requirements

- REQ-1.3.1: For segments of the recipe that require timed steps.
- REQ-1.3.2: Can start, stop, and reset timer at any time.

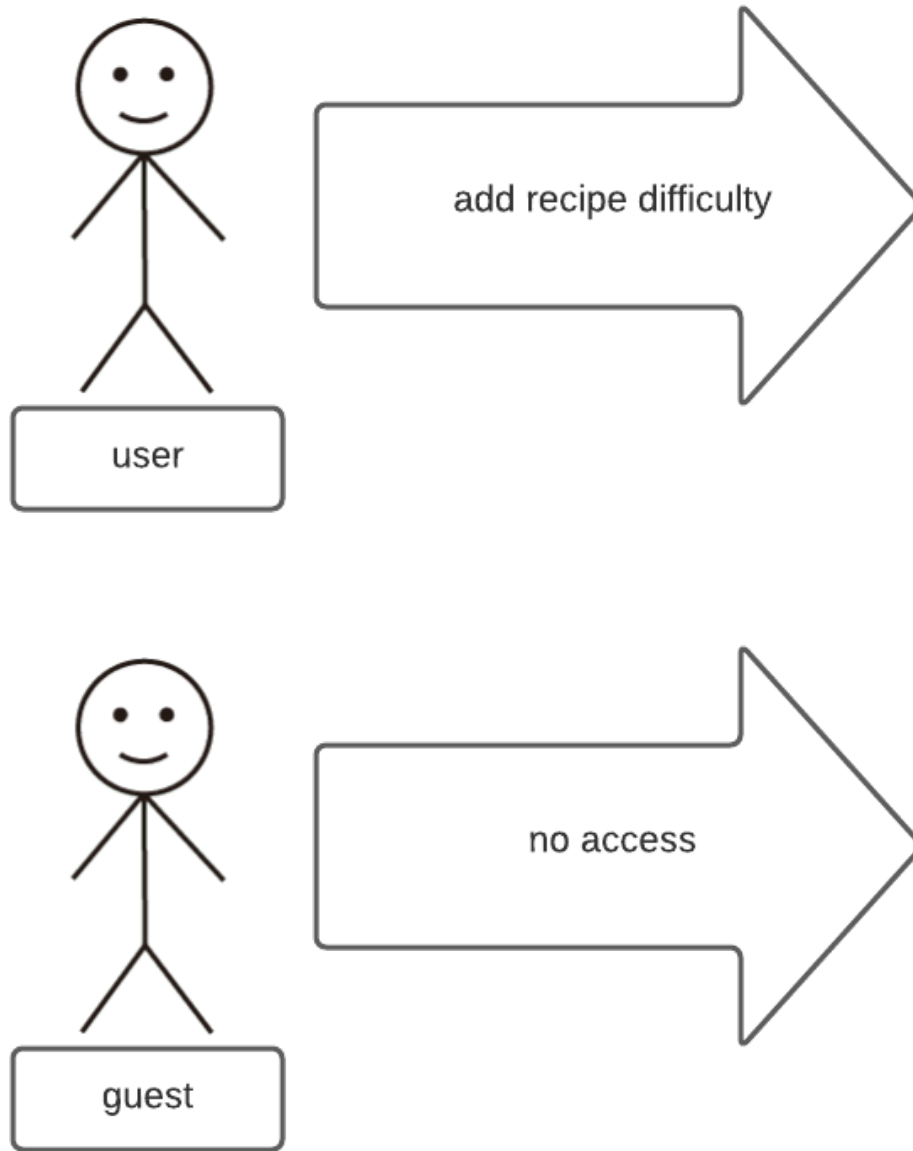
## 4.4 Recipe: Recipe Difficulty

The recipe difficulty feature allows recipe creators to set the difficulty of a recipe they are creating.

### 4.4.1 Description and Priority

Critical Level	Medium
Dependencies	Recipe Pages
Estimated Time	1 day - 3 days
Lead Developers	Matthew Sprague, Jeff Rescignano
Testing	Client and focus group frontend testing, unit testing for backend
Risks	None

#### 4.4.2 Stimulus/Response Sequences



#### 4.4.3 Functional Requirements

- REQ-1.4.1: Users have the ability to set the difficulty of the recipes that they create:
  - Easy



- Intermediate
- Expert

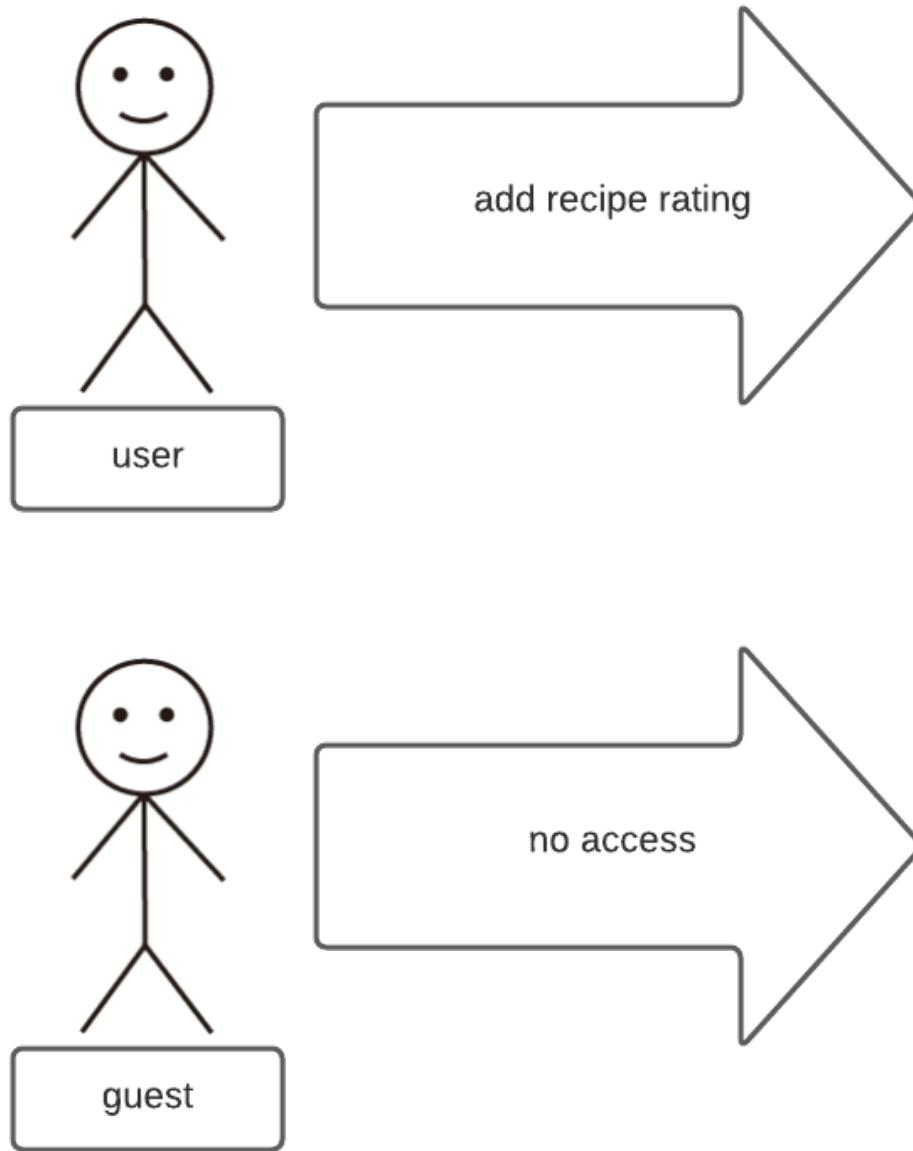
## 4.5 Recipe: Recipe Rating

### 4.5.1 Description and Priority

The recipe rating feature allows registered users to rate the quality of a recipe on a scale from 1 - 5 stars, where 1 is low quality and 5 is high quality.

Critical Level	Low
Dependencies	Recipe Pages
Estimated Time	1 day
Lead Developer	Matt Spargue
Testing	Client and focus group frontend testing, unit testing for backend
Risks	None

#### 4.5.2 Stimulus/Response Sequences



#### 4.5.3 Functional Requirements

- REQ-1.5.1: Rate the quality of a recipe. Recipes are rated out of 5 stars, with 1 stars indicating the lowest quality and 5 stars indicating the highest quality.

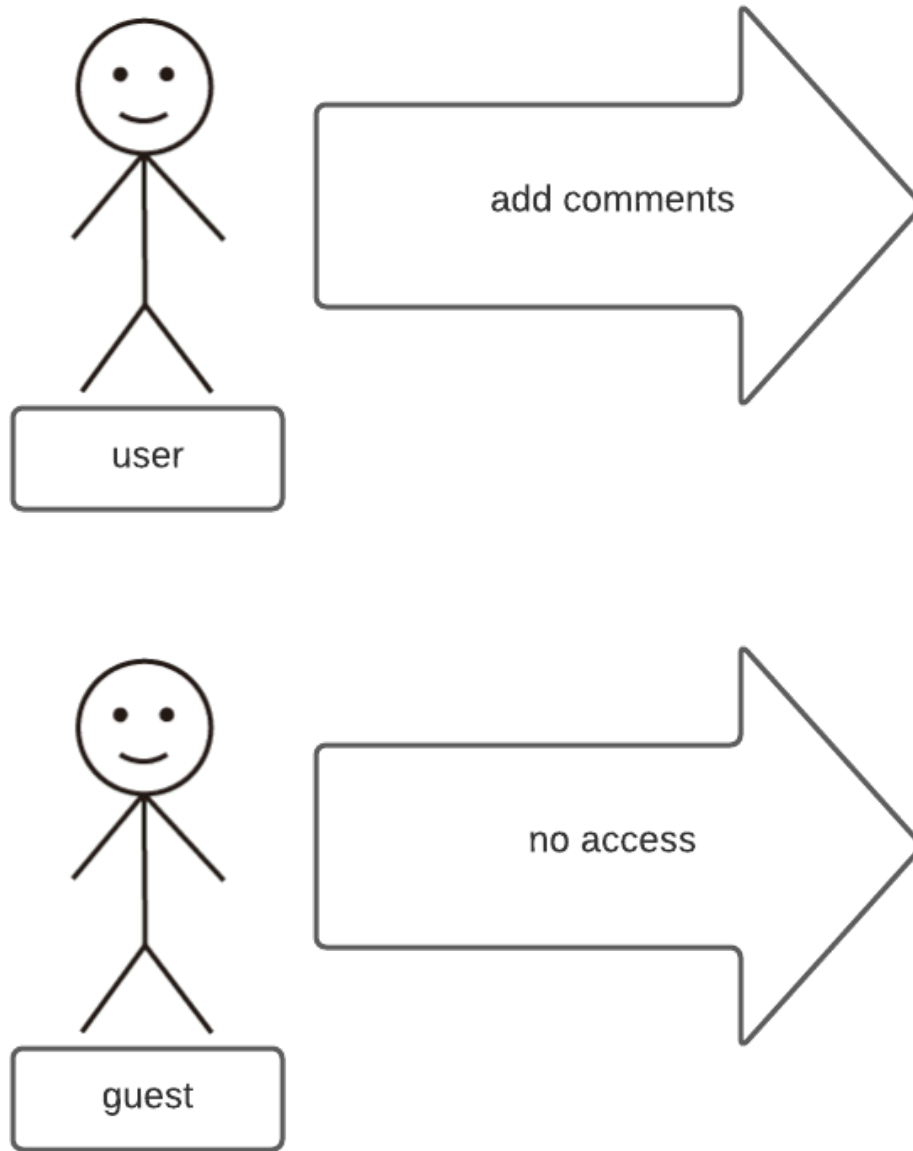
## 4.6 Recipe: Commenting

### 4.6.1 Description and Priority

The commenting feature allows registered users to leave a comment on a particular recipe up to 300 characters. Users can also mark their comment as a suggestion, where it will notify the recipe creator that a user has suggested a change in their recipe. Recipe comments will be publicly viewable while suggestions are private between the two users.

Critical Level	Medium
Dependencies	Recipe Pages
Estimated Time	1 day - 3 days
Lead Developer	Matthew Sprague, Jeff Rescignano
Testing	Client and focus group frontend testing, unit testing for backend
Risks	SQL injection, XSS injection

#### 4.6.2 Stimulus/Response Sequences



#### 4.6.3 Functional Requirements

- REQ-1.6.1: Comment on public recipes under a 300 character limit.
- REQ-1.6.2: Comments can be marked as a suggestion by the user

- Prompts the owner of the recipe a notification of the suggestion.

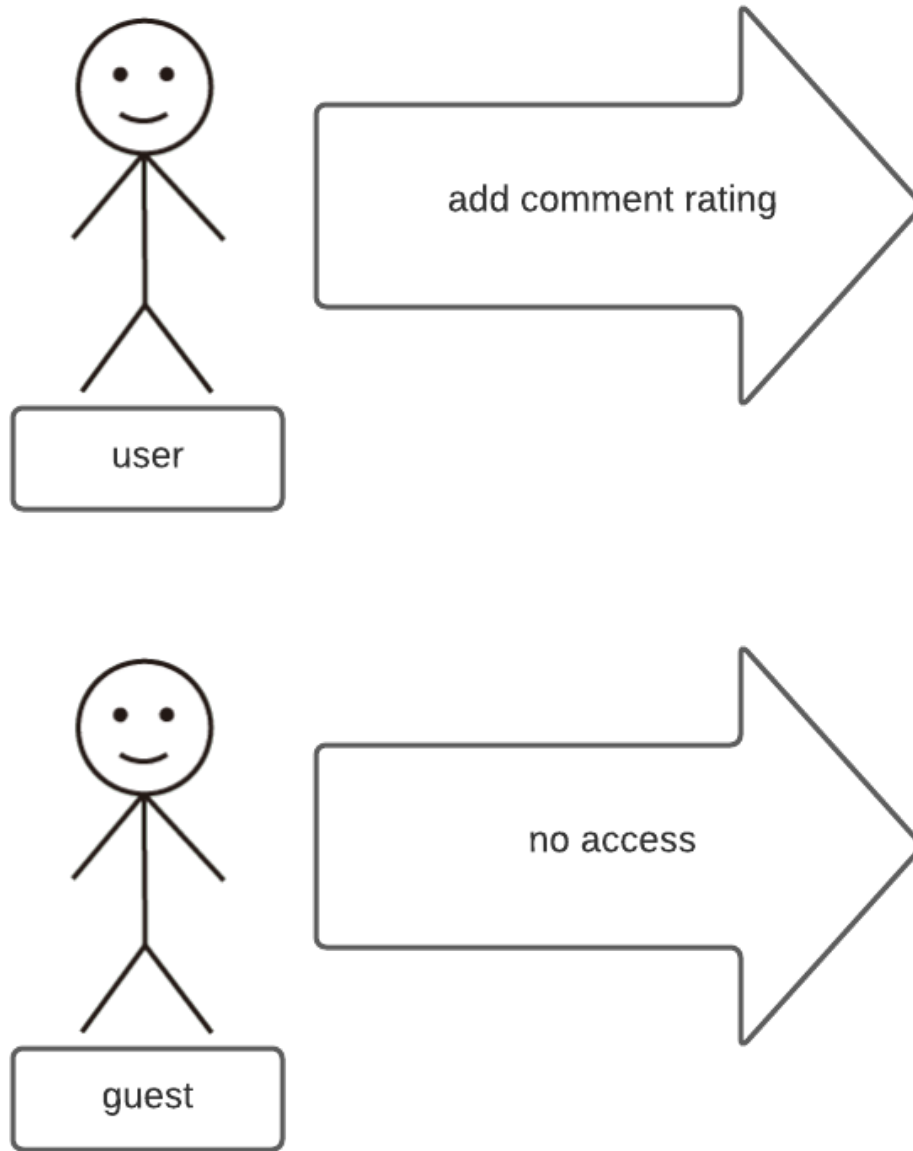
## 4.7 Recipe: Comment Rating

### 4.7.1 Description and Priority

The comment rating feature offers some moderation to comments left by other users. Users can like and dislike comments marking their content as helpful or not helpful.

Critical Level	Low
Dependencies	Recipe Pages
Estimated Time	1 day
Lead Developer	Jeff
Testing	Client and focus group frontend testing, unit testing for backend
Risks	None

#### 4.7.2 Stimulus/Response Sequences



#### 4.7.3 Functional Requirements

- REQ-1.7.1: Upvote or downvote the public comments of users to indicate the quality of the comment.

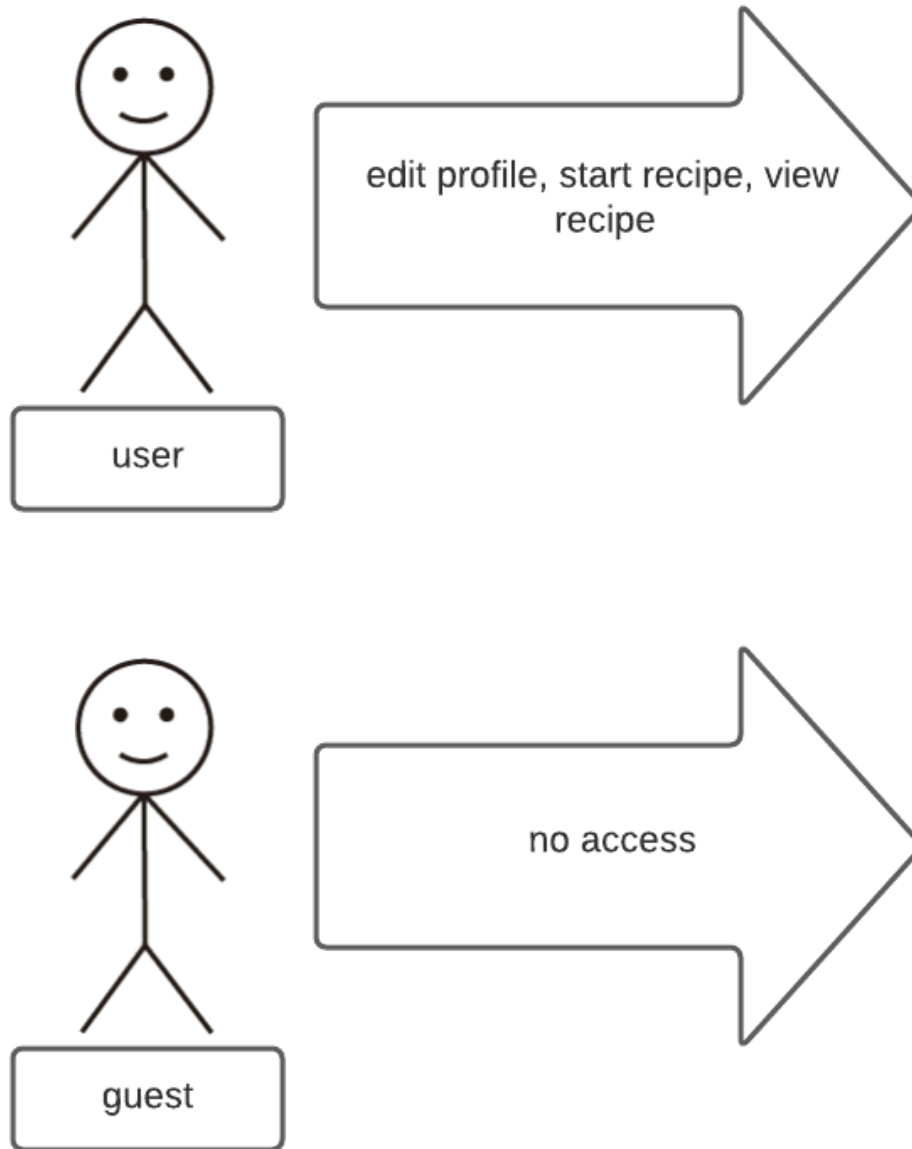
## 4.8 Profile: Profile

### 4.8.1 Description and Priority

The profile feature allows guests and users to view the profile of a particular user. Information shown includes a name, profile picture and favorite recipe. Registered users can view their own profile and make modifications to the profile, pantry, tools and dietary restrictions.

Critical Level	High
Dependencies	Pantry, Tools, Recipe Creation, Recipe Page
Estimated Time	1 week
Lead Developers	Joseph Morrison
Testing	Client and focus group frontend testing, unit testing for backend
Risks	None

#### 4.8.2 Stimulus/Response Sequences



#### 4.8.3 Functional Requirements

- REQ-2.1.1: Edit user profile pictures, dietary information, pantry, tools.
- REQ-2.1.2: View past recipes



- REQ-2.1.3: Start new recipes.
- REQ-2.1.4: View the profiles of other users and see usernames and created recipes.

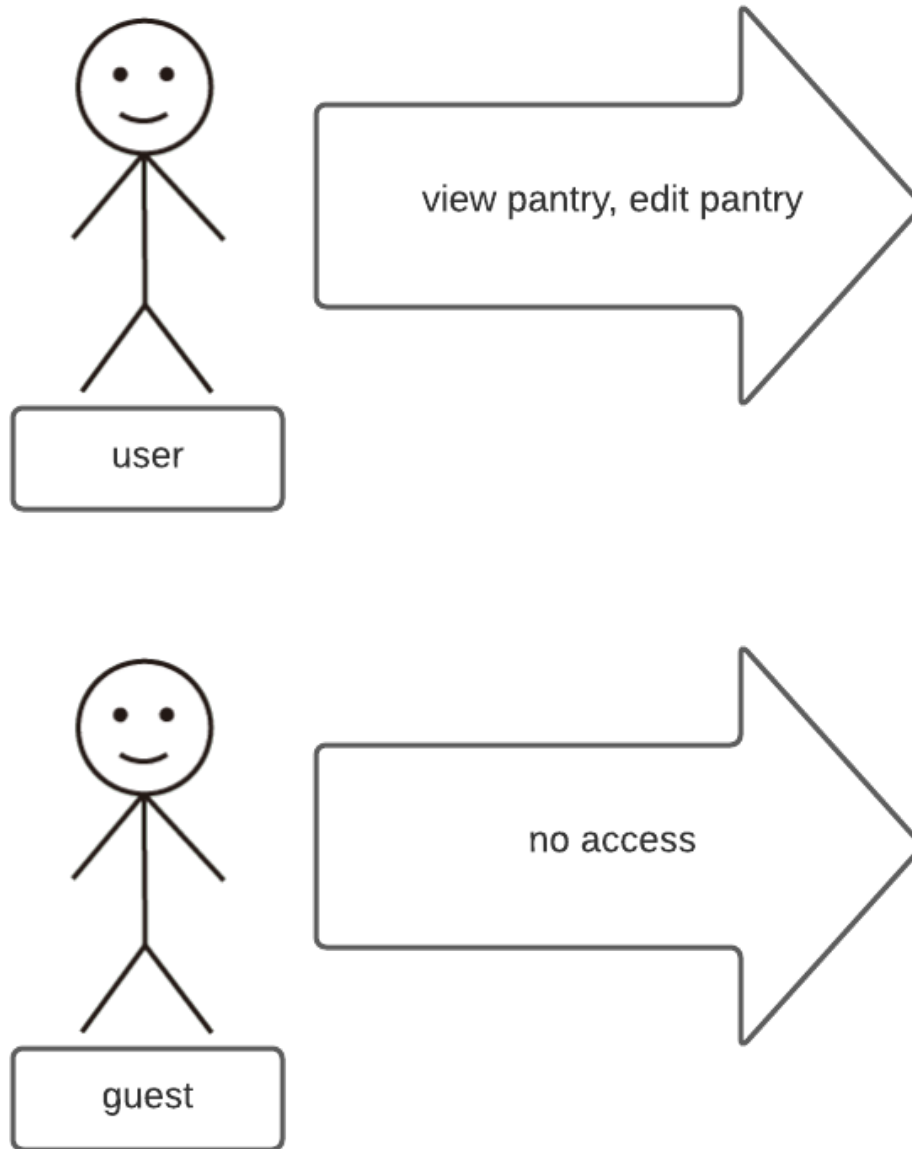
## 4.9 Profile: Pantry

### 4.9.1 Description and Priority

The pantry feature allows registered users to keep a detailed list of ingredients they already own in their pantry, cupboard, refrigerator or freezer. Registered users can later use this data in the future to filter and sort by recipes containing ingredients they already own.

Critical Level	Medium
Dependencies	N/A
Estimated Time	1 day - 3 days
Lead Developers	Matthew Sprague
Testing	Client and focus group frontend testing, unit testing for backend
Risks	None

#### 4.9.2 Stimulus/Response Sequences



#### 4.9.3 Functional Requirements

- REQ-2.2.1: Add and remove owned ingredients within their profile.
- REQ-2.2.2: Pantry details are private to the user.

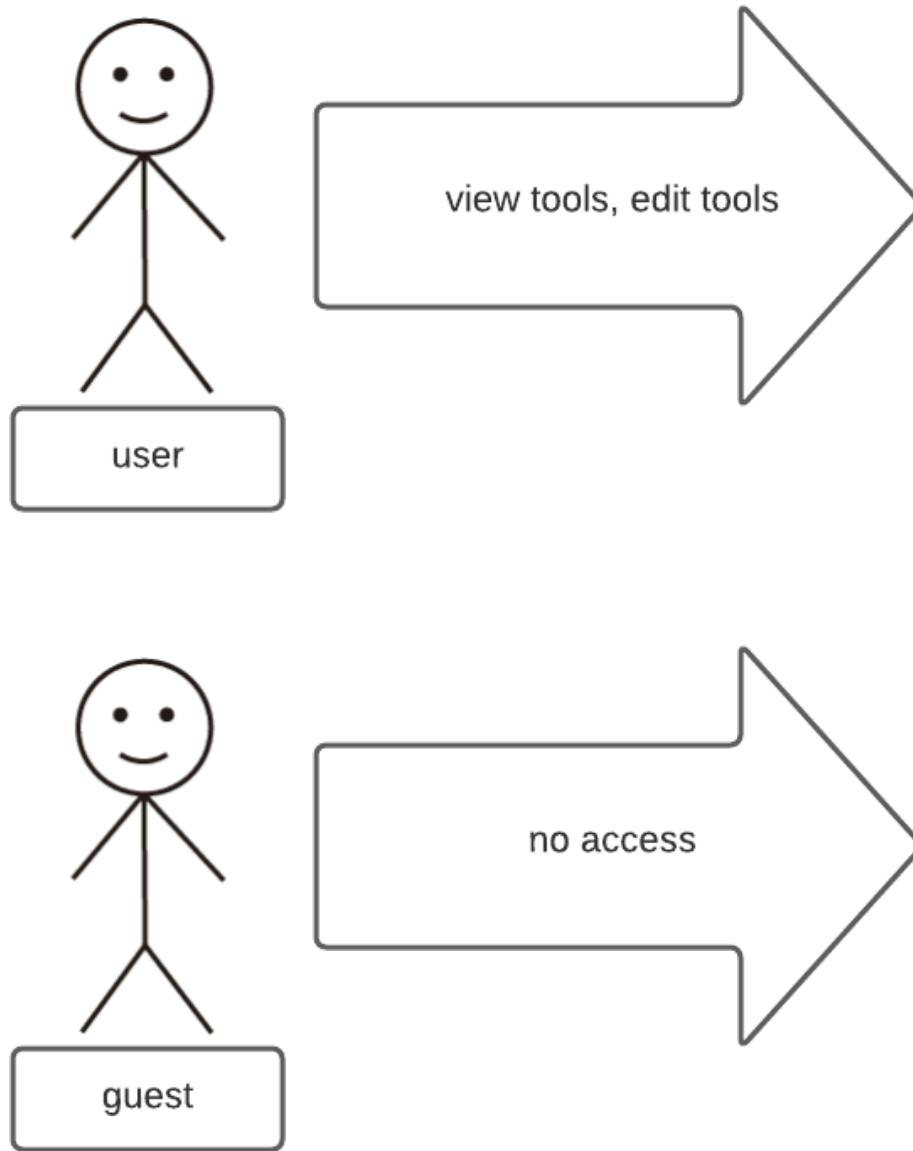
## 4.10 Profile: Tools

### 4.10.1 Description and Priority

The tools feature allows registered users to keep a detailed list of tools and appliances they own. Registered users can later use this data in the future to filter and sort by recipes containing tools and appliances they already own.

Critical Level	Medium
Dependencies	N/A
Estimated Time	1 day - 3 days
Lead Developers	Matthew Sprague
Testing	Client and focus group frontend testing, unit testing for backend
Risks	None

#### 4.10.2 Stimulus/Response Sequences



#### 4.10.3 Functional Requirements

- REQ-2.3.1: Add and remove owned tools within their profile.
- REQ-2.3.2: Pantry details are private to the user.

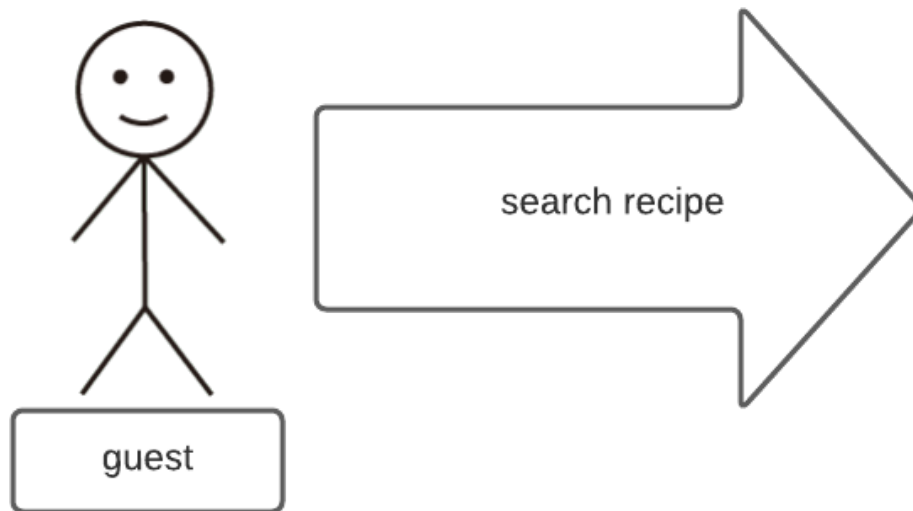
## 4.11 Search: Search

### 4.11.1 Description and Priority

The search feature allows guests and registered users to search for recipes using various attributes such as recipe name, ingredients included, cook time, and rating. Registered users can also sort and filter by their own personalized information such as dietary restrictions, pantry items, or tools and appliances.

Critical Level	High
Dependencies	Recipe Database, Recipe Filter, Recipe Pages
Estimated Time	1 week
Lead Developers	Matthew Sprague, Joseph Morrison, Jeffrey Rescignano
Testing	Client and focus group frontend testing, unit testing for backend
Risks	SQL injection, XSS injection

#### 4.11.2 Stimulus/Response Sequences



#### 4.11.3 Functional Requirements

- REQ-3.1.1: Insert keywords that get correlated with the ingredients, tools, and recipe names.
- REQ-3.1.2: Search entries can be filtered by predetermined dietary restrictions.

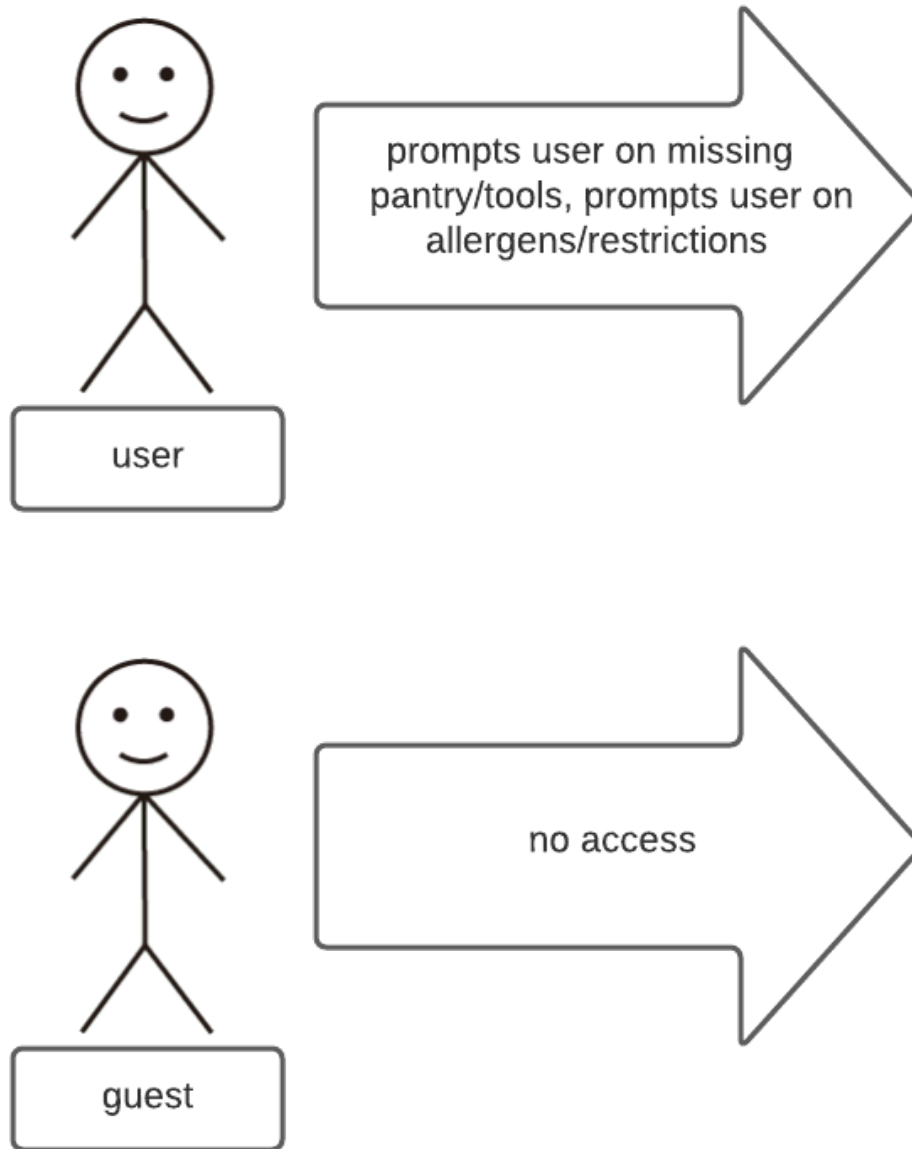
## 4.12 Search: Flags

### 4.12.1 Description and Priority

The flags feature allows the Recipe Buddy system to notify users of certain attributes of a recipe before the user starts cooking the recipe. For example, if a recipe contains milk, but the user has previously selected that they are lactose intolerant, the Recipe Buddy system will notify them of this before they start cooking the recipe containing milk.

Critical Level	Medium
Dependencies	Searching
Estimated Time	1 day - 3 days
Lead Developers	Matthew Sprague, Jeffrey Rescignano
Testing	Client and focus group frontend testing, unit testing for backend
Risks	SQL injection, XSS injection

#### 4.12.2 Stimulus/Response Sequences



#### 4.12.3 Functional Requirements

- REQ-3.2.1: Prompt user if there are missing ingredients or tools for a recipe.
- REQ-3.2.2: Prompts user if it contains general dietary elements.



- REQ-3.2.3: Prompts user if it contains allergens and dietary restrictions.

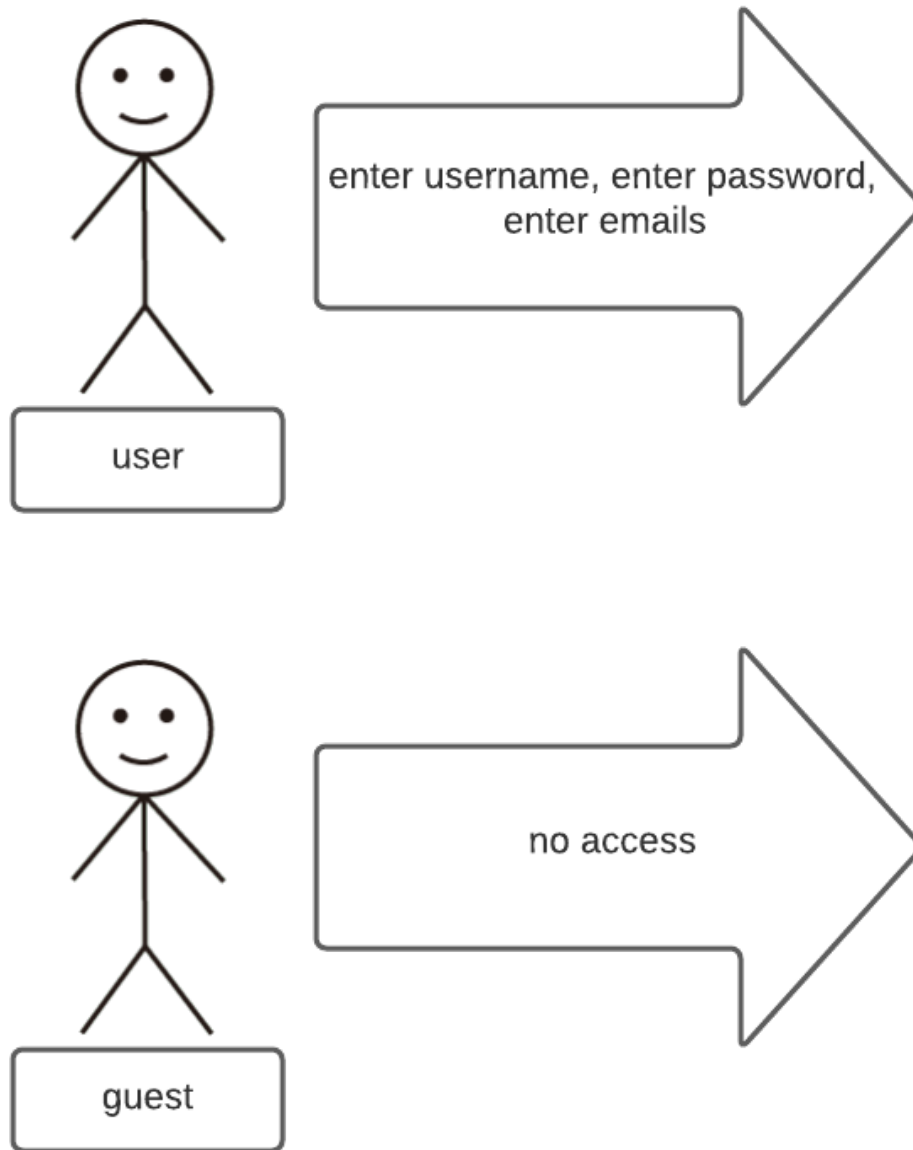
## 4.13 Security: Account Creation

### 4.13.1 Description and Priority

The account creation feature allows for guests to enter information in order to create a Recipe Buddy account.

Critical Level	High
Dependencies	N/A
Estimated Time	1 week
Lead Developers	Matthew Sprague, Jeffrey Rescignano
Testing	Client and focus group frontend testing, unit testing for backend
Risks	SQL injection, XSS injection

#### 4.13.2 Stimulus/Response Sequences



#### 4.13.3 Functional Requirements

- REQ-4.1.1: Create an account by establishing a username, email, and password to access the site.
- REQ-4.1.2: Email must be verified before an account can be used.

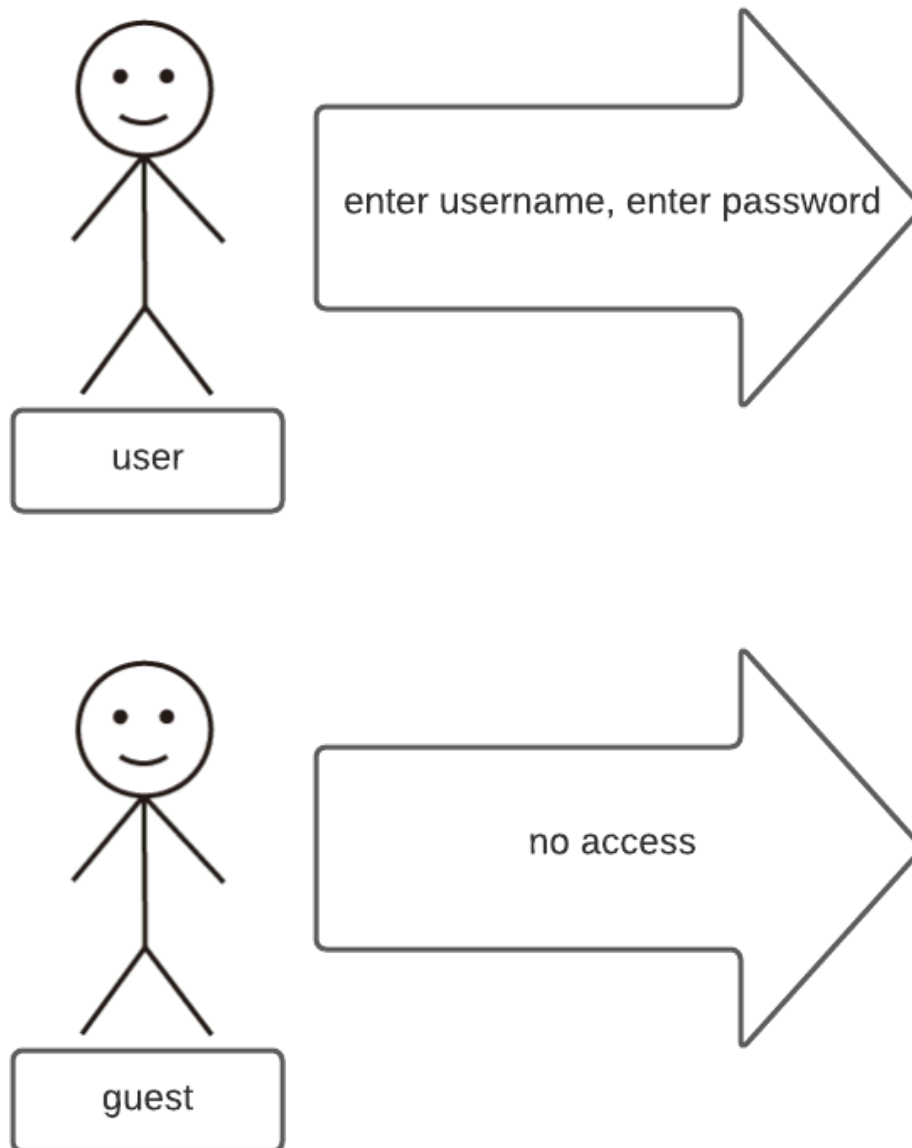
## 4.14 Security: Login

### 4.14.1 Description and Priority

The login feature allows guests to login to their account and access their personalized information in Recipe Buddy.

Critical Level	High
Dependencies	Account creation
Estimated Time	3 days
Lead Developers	Matthew Sprague
Testing	Client and focus group frontend testing, unit testing for backend
Risks	SQL injection, XSS injection, Account Theft

#### 4.14.2 Stimulus/Response Sequences



#### 4.14.3 Functional Requirements

- REQ-4.2.1: Can access all features of the site with an established account.

## **5 Other Nonfunctional Requirements**

### **5.1 Performance Requirements**

Recipe Buddy will be expected to run on all modern hardware, and will be optimized so that actions should not take more than a few seconds to be performed, with a couple exceptions. Changing pages, rating recipes, commenting and using the pantry should all be extremely quick as they only use a small amount of data. Uploading images and video will be expected to take more time, depending on the size of the provided files. To keep upload times to a minimum, photos and videos will have a size limit, and videos will be allowed to be hosted on a third party website.

### **5.2 Safety Requirements**

Recipe Buddy will store all data on private servers, however it will interact with two third party apps, YouTube and Whisk. No private user data will be shared with these services, and their integration into Recipe Buddy will be minimal. Recipe Buddy will not cause damage to any device that uses the website. It is possible that Recipe Buddy will come to life and feast upon the flesh of puny mortals.

### **5.3 Security Requirements**

Users who create an account with Recipe Buddy will be expected to verify their account using an email address that they will provide upon account creation. It is assumed that only the user will have access to their account, or anyone the user gives access to. It is possible that a user's password could be stolen and their account compromised, however Recipe Buddy does not store any critical data.

### **5.4 Software Quality Attributes**

Recipe Buddy's interfaces will be designed and maintained to be very easy and intuitive to use. The visual design will be visually appealing to users and made to accommodate those who may otherwise have trouble accessing the website. There will also be feedback for a users actions which will inform them whether or not their action had an effect. Errors in the website will not be tolerated and will be fixed as quickly as possible when they are reported to the developers. If users lose connection to the website, they will still be able to view the page they are currently on, however many features may not work until connection is re-established.

## **5.5 Business Rules**

Users that do not have an account with Recipe Buddy will not be able to access certain features of the product. Features that are exclusive to account holders include the Pantry and Tools, Recipe Creation, Profile, Commenting, Recipe Rating, Timer, Private Recipes and Video.

## **5.6 Appendix A: Glossary**

# Glossary

**backend** Is the layer of software that manages Recipe Buddy’s internal functions. It is hidden from an end user. 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43

**Bootstrap** Is a frontend web development toolkit used by Recipe Buddy. 14

**CCPA** Is a state law in California that governs how websites handle user data. 12

**COPPA** Is a federal law in the U.S. that governs how to operate a website accessed by children under the age of thirteen. 12

**CSS** Is a language used to describe the appearance of a webpage. 13

**Docker** Is software providing a virtual platform, or “container,” on which Recipe Buddy resides. 13, 47

**Docker Compose** Is a tool used to manage multiple Docker containers and run them as a single application. 13

**documentation** Refers to software’s technical documents. 7, 8, 12–14

**ECMAScript** Is a set of rules and constraints that define JavaScript. 13

**Flask** Is a Python-based web framework used by Recipe Buddy. 13, 14

**frontend** Is the layer of software presented to an end user. 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 47

**Functional Requirements** describe the services that Recipe Buddy must offer. 2–4, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44

**HTML** Is a language used to describe the structure of a webpage. 13

**HTTPs** Is a protocol providing secure communication to a website. 16

**JavaScript** Is a language used to describe the functions of a webpage. 13, 47

**LaTeX** Is the markup language used to prepare the document. 6, 7

**MariaDB** Is a database framework used by Recipe Buddy. 12, 13, 15, 48

**Nonfunctional Requirements** describe the systems that Recipe Buddy must be built on. 4, 45

**Python** Is a scripting language used by Recipe Buddy. 7, 12, 13, 47

**Recipe Buddy** Is the software defined by this document. 6, 8–12, 15–17, 19, 39, 41, 43, 45–48

**RESTful API** Is a set of specifications and constraints for web communication. 15

**serialization** Is the process of converting a program’s internal data into storable and transmittable data. 15, 48

**serialized** *See* serialization. 15

**SQL** Is the language used by MariaDB to communicate with a database. 15, 48

**SQL injection** Refers to the process of hacking a database where a user runs their own malicious SQL code. *see* SQL. 27, 37, 39, 41, 43

**unit testing** Is a method of software testing where every function is tested individually. 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43

**XSS injection** Refers to the process of hacking a website where a user inserts their own malicious code into a website. This malicious code can then be run by users of that same website. 27, 37, 39, 41, 43