

---

# **Software Requirements Specification**

**for**

# **RecipeBuddy**

**Version 1.0 approved**

**Prepared by Sydney DeCyllis and Henry Rausch**

***C•H•S•D•N Co-op***

**1-12-2023**

# Table of Contents

Revision History .....	Error! No bookmark name given.
<b>1. Introduction.....</b>	<b>3</b>
Purpose.....	3
Intended Audience and Reading Suggestions.....	Error! No bookmark name given.
Project Scope .....	Error! No bookmark name given.
References.....	Error! No bookmark name given.
<b>2. Overall Description.....</b>	<b>Error! No bookmark name given.</b>
Product Perspective.....	3
Product Features.....	Error! No bookmark name given.
User Classes and Characteristics .....	Error! No bookmark name given.
Operating Environment.....	Error! No bookmark name given.
Design and Implementation Constraints.....	Error! No bookmark name given.
Assumptions and Dependencies .....	Error! No bookmark name given.
<b>3. System Features .....</b>	<b>Error! No bookmark name given.</b>
Add Recipe.....	6
Remove Recipe .....	6
Edit Recipe.....	7
Cook Recipe.....	7
Pantry .....	8
Users .....	9
Search.....	10
<b>4. External Interface Requirements .....</b>	<b>11</b>
User Interfaces .....	11
Hardware Interfaces .....	Error! No bookmark name given.
Software Interfaces .....	Error! No bookmark name given.
<b>5. Other Nonfunctional Requirements.....</b>	<b>Error! No bookmark name given.</b>
Performance Requirements.....	13
Safety Requirements .....	Error! No bookmark name given.
Security Requirements .....	Error! No bookmark name given.
Software Quality Attributes .....	Error! No bookmark name given.
<b>6. Key Milestones .....</b>	<b>Error! No bookmark name given.</b>
<b>7. Key Resource Requirements.....</b>	<b>Error! No bookmark name given.</b>
<b>8. Requirement Change Management.....</b>	<b>Error! No bookmark name given.</b>

# 1. Introduction

## Purpose

Version 1.0

*RecipeBuddy is an Android app designed to help users create, store, and cook recipes. The purpose of the app is to provide a convenient and user-friendly way for individuals to manage their recipe collections. RecipeBuddy aims to streamline the recipe creation process, allowing users to easily create and save recipes and customize ingredients and instructions. By providing a comprehensive solution for recipe management, RecipeBuddy seeks to make cooking at home more accessible and enjoyable for individuals of all skill levels.*

## Intended Audience and Reading Suggestions

*Given that RecipeBuddy is created by a small team of 5 developers, this document is primarily created for those 5 developers. As such, the whole document will likely be relevant, although each reader will have their own focus on a certain system. For those not on the development team, starting at Section 1 will give you a general idea of the document, and Section 2 will provide a general idea of the product.*

## Project Scope

*This SRS describes the software requirements for RecipeBuddy, a mobile platform for managing a cookbook. The system includes features for creation, storage, and cooking of recipes. This SRS covers the entire system, including both front-end and back-end functionality.*

## References

*There are no direct references to other products in this document, but other information about RecipeBuddy can be found at <http://cs350-chsdn.cosi.clarkson.edu/index.html>.*

# 2. Overall Description

## Product Perspective

*RecipeBuddy is a new, self-contained product designed to allow a user to create, store, and cook recipes. The software is designed to run on Android devices and is intended for use by home cooks who want to keep track of their favorite recipes and discover new ones.*

*RecipeBuddy is not part of a larger product family or a replacement for existing systems. Instead, it is a standalone product with a clear focus on recipe management and cooking. The product is entirely self contained, so there are no external interfaces other than the screen.*

## **Product Features**

*The product is designed around the following key features*

- *Create and store recipes*
- *Access and cook recipes*
- *Allow for users to login and access the product*

## **User Classes and Characteristics**

*RecipeBuddy is designed to be used by a variety of users, but primarily home cooks. A full list of expected user types follows:*

*Novice Cooks: These users have little to no cooking experience and may require step-by-step guidance in creating and executing recipes. They may also be less familiar with cooking terminology and techniques.*

*Experienced Cooks: These users have some cooking experience and are comfortable with basic cooking terminology and techniques. They will likely require less guidance in cooking concepts.*

*Professional Chefs: These users are experienced and knowledgeable about cooking and require a high level of precision and detail in their recipes. They may require advanced features such as recipe scaling and ingredient substitutions.*

*Certain requirements may pertain only to certain user classes. For example, novice cooks may require more guidance and simpler recipes, while professional chefs may require more advanced features and the ability to scale recipes. The favored user classes for RecipeBuddy are novice and experienced cooks, as they represent the largest user base and are likely to use the app on a regular basis. However, all user classes are important to satisfy in order to ensure that RecipeBuddy is a useful and valuable tool for a wide range of users.*

## **Operating Environment**

*The product will operate on systems running at least Android 21, but is designed for Android 32*

## **Design and Implementation Constraints**

*Our product is delivered “as-is,” with no expectation of continued development after delivery. The product is to be self contained, and will operate normally within an Android system upon installation of the program. Strict requirements regarding storage and other usages are detailed in the requirements section.*

## **Assumptions and Dependencies**

*Our only dependency is Jetpack, which is a standard part of the Android development package. As a result, we can be strongly certain we will not encounter any logical or legal issues regarding usage or reference of external resources.*

## 3. System Features

### Add Recipe

#### 3.1.1 Description and Priority

*The system shall allow a user to add a new recipe to their personal storage. This feature has a high priority.*

#### 3.1.2 Stimulus/Response Sequences

1. User selects "Add Recipe" option from the main menu.
2. System presents a form for the user to enter recipe information.
3. User enters recipe details such as title, image(s), individual ingredients and quantity, utensils used, steps in creation, number of people served, scalable by user, allergens, 1-5 star ratings, comments from previous usages, timer(s), and tags.
4. User clicks "Submit" button.
5. System stores the recipe in the user's personal storage and presents a confirmation message.

#### 3.1.3 Functional Requirements

*REQ-1: The system shall provide a form for the user to enter recipe information.*

*REQ-2: The system shall allow the user to add an image to a recipe.*

*REQ-3: The system shall allow the user to enter individual ingredients and quantity for the recipe.*

*REQ-4: The system shall allow the user to enter utensils used in the recipe.*

*REQ-5: The system shall allow the user to enter steps in creation for the recipe.*

*REQ-6: The system shall allow the user to enter the number of people served by the recipe.*

*REQ-7: The system shall allow the user to scale the recipe by adjusting the quantity of ingredients.*

*REQ-8: The system shall allow the user to rate the recipe using a 1-5 star rating system.*

*REQ-9: The system shall generate tags for the recipe based on user input.*

*REQ-10: The system shall allow the user to add comments to the recipe.*

*REQ-11: The system shall allow the user to set a timer for the recipe.*

*REQ-12: The system shall store the recipe in the user's personal storage.*

### Remove Recipe

#### 3.2.1 Description and Priority

*The "Remove Recipe" feature allows users to remove a recipe from their storage. This feature has a medium priority.*

#### 3.2.2 Stimulus/Response Sequences

1. User selects the recipe they want to remove from their recipe storage.
2. System prompts the user to confirm whether they want to remove the recipe.
3. User confirms that they want to remove the recipe.
4. System removes the recipe from the user's storage and displays a success message.

### 3.2.3 Functional Requirements

*REQ-1: The system must display a list of the user's saved recipes.*

*REQ-2: The user must be able to select a recipe to remove.*

*REQ-3: The system must prompt the user to confirm the removal of the recipe.*

*REQ-4: The user must confirm the removal of the recipe.*

*REQ-5: The system must remove the recipe from the user's storage.*

*REQ-6: The system must display a success message after the recipe is removed.*

## Edit Recipe

### 3.3.1 Description and Priority

*This feature allows users to modify existing recipes in their storage. It is of high priority.*

### 3.3.2 Stimulus/Response Sequences

1. *User selects the recipe they want to edit from their recipe storage.*
2. *System displays the recipe and allows the user to modify the title, ingredients, steps, timer, and pictures.*
3. *User makes desired modifications.*
4. *System saves the changes and displays a success message.*

### 3.3.3 Functional Requirements

*REQ-1: The system must display a list of the user's saved recipes.*

*REQ-2: The user must be able to select a recipe to edit.*

*REQ-3: The system must display the selected recipe and allow the user to modify the title, ingredients, steps, timer, and pictures.*

*REQ-4: The user must be able to modify the recipe's title.*

*REQ-5: The user must be able to modify the recipe's ingredients.*

*REQ-6: The user must be able to modify the recipe's steps.*

*REQ-7: The user must be able to modify the recipe's timer.*

*REQ-8: The user must be able to modify the recipe's pictures.*

*REQ-9: The system must save the changes made by the user.*

*REQ-10: The system must display a success message after the changes are saved.*

## Cook Recipe

### 3.4.1 Description and Priority

*The "Cook" feature allows users to view and follow the steps for cooking a recipe. This feature has a high priority.*

### 3.4.2 Stimulus/Response Sequences

1. *User selects a recipe to cook from their saved recipes.*
2. *System displays the recipe's information, including ingredients, steps, and timer.*
3. *User selects the "Cook" button to begin cooking.*
4. *System displays the recipe instructions and timer*

### 3.4.3 Functional Requirements

*REQ-1: The system must display the recipe's ingredients, steps, and timer.*

*REQ-2: The system must allow the user to start cooking the recipe.*

*REQ-3: The system must display the recipe steps effectively.*

*REQ-4: The system must have a useable timer that persists into the background*

## Pantry

### 3.5.1 Description and Priority

*The "Pantry" feature allows users to add and manage the items in their pantry. This feature has a high priority.*

### 3.5.2 Stimulus/Response Sequences

1. *User selects the "Pantry" feature from the main menu.*
2. *System displays the user's current pantry inventory.*
3. *User selects the "Add" button to add a new item to the pantry.*
4. *System prompts the user to enter the details of the new pantry item, including the name, quantity, and expiration date.*
5. *User enters the details of the new pantry item and confirms the addition.*
6. *System adds the new item to the pantry inventory and displays a success message.*
7. *User selects an existing pantry item and clicks the "Edit" button.*
8. *System displays the details of the selected pantry item.*
9. *User modifies the details of the pantry item and confirms the changes.*
10. *System updates the pantry inventory with the modified details and displays a success message.*
11. *User selects an existing pantry item and clicks the "Remove" button.*
12. *System removes the selected pantry item from the pantry inventory and displays a success message.*

### 3.5.3 Functional Requirements

*REQ-1: The system must display a list of the user's current pantry inventory.*

*REQ-2: The user must be able to add a new item to the pantry.*

*REQ-3: The system must prompt the user to enter the details of the new pantry item, including the name and quantity.*

*REQ-4: The user must confirm the addition of the new pantry item.*

*REQ-5: The system must add the new item to the pantry inventory.*

*REQ-6: The system must display a success message after the new item is added to the pantry inventory.*

*REQ-7: The user must be able to modify the details of an existing pantry item.*

*REQ-8: The system must display the details of the selected pantry item.*

*REQ-9: The user must be able to modify the details of the pantry item.*

*REQ-10: The user must confirm the changes to the pantry item details.*

*REQ-11: The system must update the pantry inventory with the modified details.*

*REQ-12: The system must display a success message after the pantry item details are updated.*

*REQ-13: The user must be able to remove an existing pantry item.*



*REQ-14: The system must prompt the user to confirm the removal of the selected pantry item.*

*REQ-15: The system must display a success message after the pantry item is removed from the pantry inventory.*

## Users

### 3.6.1 Description and Priority

*The "Users" feature allows users to create accounts, log in, and manage their account information. This feature has a high priority.*

### 3.6.2 Stimulus/Response Sequences

1. *User selects the "Create Account" button.*
2. *System prompts the user to enter a username and password.*
3. *User enters a unique username and a strong password.*
4. *System verifies that the username is available and the password meets the requirements.*
5. *System creates a new account with the provided username and password and displays a success message.*
6. *User enters their username and password and selects the "Log In" button.*
7. *System verifies the username and password.*
8. *System grants the user access to their storage and displays their saved recipes.*
9. *Logging Out:*
10. *User selects the "Log Out" button.*
11. *System logs the user out and displays a success message.*

### 3.6.3 Functional Requirements

*REQ-1: The system must provide a "Create Account" button.*

*REQ-2: The system must prompt the user to enter a username and password when creating a new account.*

*REQ-3: The system must verify that the username is available and the password meets the requirements when creating a new account.*

*REQ-4: The system must create a new account with the provided username and password when the user submits the account creation form.*

*REQ-5: The system must display a success message after a new account is created.*

*REQ-6: The system must provide a "Log In" button.*

*REQ-7: The system must prompt the user to enter their username and password when logging in.*

*REQ-8: The system must verify the username and password when logging in.*

*REQ-9: The system must grant the user access to their storage and display their saved recipes when the user successfully logs in.*

*REQ-10: The system must provide a "Log Out" button.*

*REQ-11: The system must log the user out and display a success message when the user selects the "Log Out" button.*

## Search

### 3.7.1 Description and Priority

*The "Search" feature allows users to search for recipes based on various criteria. This feature has a high priority.*

### 3.7.2 Stimulus/Response Sequences

12. *User enters search terms and selects desired filters.*
13. *System searches for recipes that match the entered criteria.*
14. *System displays a list of matching recipes.*
15. *User selects a recipe from the list.*
16. *System displays the recipe details.*

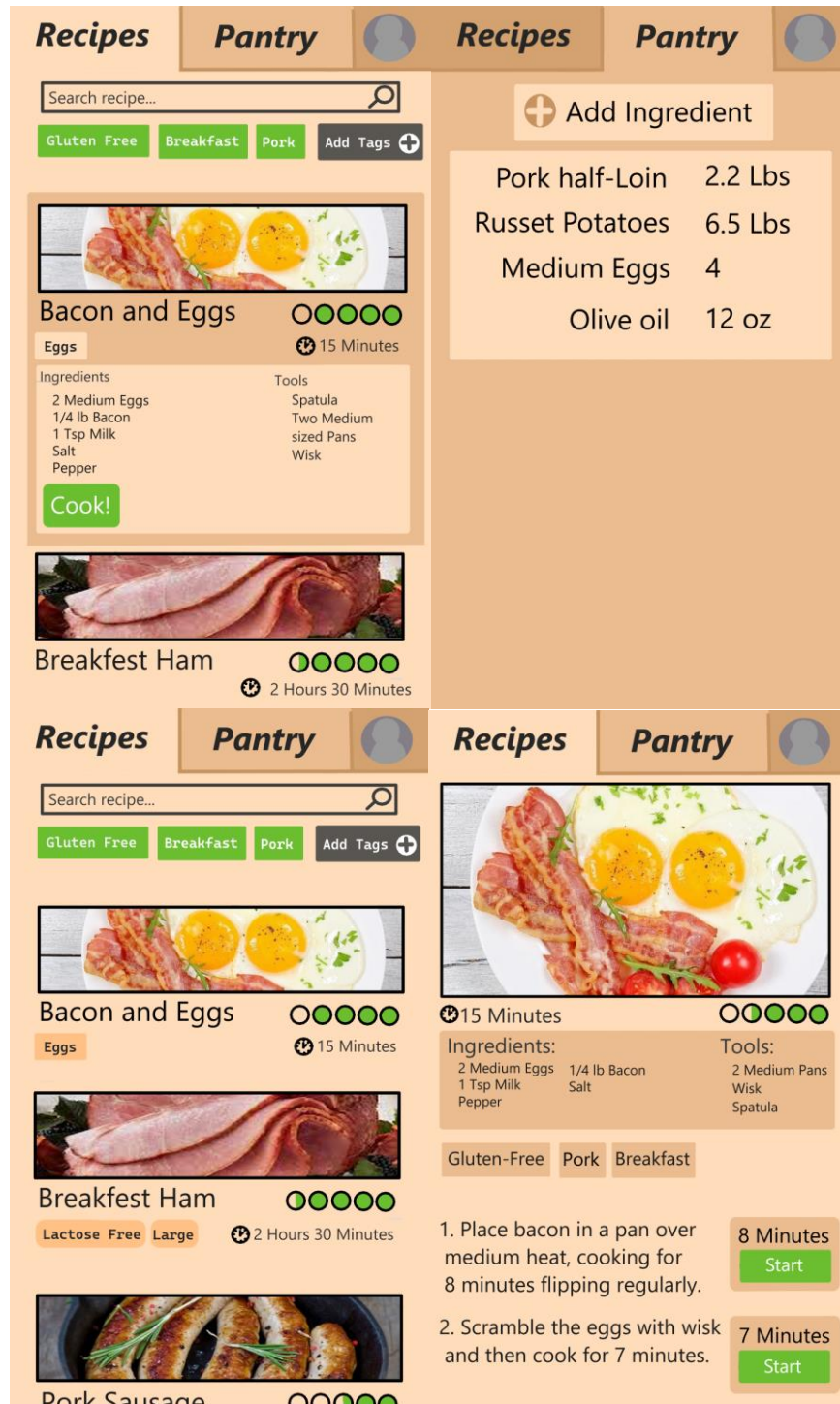
### 3.7.3 Functional Requirements

- REQ-1: The system must allow users to enter search terms.*
- REQ-2: The system must allow users to select filters for their search.*
- REQ-3: The system must search for recipes that match the entered criteria.*
- REQ-4: The system must display a list of matching recipes.*
- REQ-5: The user must be able to select a recipe from the list.*
- REQ-6: The system must display the details of the selected recipe.*

## 4. External Interface Requirements

### User Interfaces

The UI will match the following templates:



## **Hardware Interfaces**

*The only hardware interactions will be with any external device storage and the touch screen.*

## **Software Interfaces**

*The software product will use the Room database library for persistent storage of recipe and pantry data. The app will be developed using the Kotlin programming language and will use Android Studio as the primary development tool. The app will be compatible with Android version 21 and higher.*

## 5. Other Nonfunctional Requirements

### Performance Requirements

The product will be designed to meet the following requirements:

*Will scale to user's screen size without delay*

*User will be able to navigate to any primary function of the app in 3 clicks (Primary function defined as view, add, edit, etc)*

*User will be able to log in in less than a minute.*

*Saving a recipe will take less than one second*

### Safety Requirements

*There are not many expected safety issues with the product, but data is expected to be preserved in the event of a crash.*

### Security Requirements

*Recipe Buddy avoids the need for most security measures by not hosting any critical or personal info from the user. Recipe Buddy uses a local user system specifically to avoid the issue of storing anything sensitive, and does not allow for duplicate usernames to prevent issues regarding accessing another account.*

*We do not encrypt our file system because the stored recipes has been deemed insensitive enough to be unencrypted and to assist in the cross use of stored documents between users and apps.*

### Software Quality Attributes

*Usability: The app should be easy to use for users of all levels of technical expertise. This can be measured by the average time it takes for a user to complete a task, the number of clicks required to complete a task, and the overall user satisfaction rating.*

*Reliability: The app should be reliable and perform consistently under varying conditions. This can be measured by the number of crashes or bugs reported by users, the app's uptime, and the app's response time.*

*Maintainability: The app should be easy to maintain and update for developers. This can be measured by the time it takes to make updates or bug fixes, the ease of integrating new features, and the level of code complexity.*

*Portability: The app should be able to run on multiple devices and operating systems without significant modifications. This can be measured by the number of platforms the app can run on and the level of effort required to make the app compatible with a new platform.*

*Security: The app should have robust security measures to protect user data and prevent unauthorized access. This can be measured by the number of security vulnerabilities found and the level of encryption used to protect user data.*

*Performance: The app should perform well and load quickly, even under heavy user load. This can be measured by the app's response time and the app's resource usage.*

## 6. Key Milestones

#	Milestone	Target Completion Date	Comments
1.	UI and program Design outline completion.		Will likely need further development as the product advances, to account for any unforeseen UI or Program needs.
2.	Front end Dev Completion		Same as milestone 1
3.	Back end Dev Completion		
4.	Integration and Testing	5-3-23	

## 7. Key Resource Requirements

Major Project Activities	Skill/Expertise Required	Project Member	External Resource	Issues/Constraints
UI Design	DA experience, Workflow understanding,	Sydney	N/a	Has to be designed for an android phone.
Front-end Design and implementation	Kotlin, Jetpack; cs experience; Workflow understanding	Ngaio, Carter, Derryk	N/a	UI Design needs to be completed.
Back-end Design and implementation	MySQL; Kotlin:	Henry	N/a	
Front-end and Back-end Integration	MySQL; Kotlin: Jetpack	Henry, Carter, Derryk, Ngaio	N/a	Both Front-end and Back-end need to be almost completely done.
System Requirements Documentation		Henry, Sydney	N/a	
Testing and Testing Plan	Kotlin: Jetpack	Ngaio, Derryk, Carter	N/a	Integration needs to be complete for this too begin.
Web Design and Dev	html; Css; DA experience.	Sydney	N/a	Needs to be hosted somewhere.



## **8. Requirement Change Management**

In order to update the document, a consensus must be reached by all members of the team. A proposed update may arise at any group meeting, if a member wishes to bring forth a vote. If a majority (3 out of 5) team members vote yes to a proposed change, it will be implemented and documented accordingly.