Cook Book

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

Course CS4850

Spring 2023

Professor Perry

2/12/23

Team:

Madelyn McAmish

Irfan Khan

Amir Davis

Tyrell Clarke

Ryeon Naderi

Table of Contents

1.0 Introduction **3**

1.1 Overview 3

1.2 Project Goals 3

1.3 Definitions and Acronyms 3

1.4 Assumptions 3

2.0 Design Constraints **3**

2.1 Environment 3

2.2 User Characteristics 3

2.3 System 4

3.0 Functional Requirements **4**

4.0 Non-Functional Requirements **4**

4.1 Security 4

4.2 Capacity 4

4.3 Usability 4

4.4 Other 4

5.0 External Interface Requirements **4**

5.1 User Interface Requirements 4

5.2 Hardware Interface Requirements 5

5.3 Software Interface Requirements 5

5.4 Communication Interface Requirements 5

APPENDICES **5**

1.0 Introduction

You’re human, and you’re hungry. The question is how hungry are you? How often do you come across a picture or video online of food that you want to make? Wouldn’t it be convenient to have an app customized to finding and sharing recipes? New foods, new ways to make them. Recipes from far and wide. It is completely interactive, and a great way to spend time with friends and family. Introducing (Cook Book, your stomachs’ best friend) (Cook Book, the hub to your hungry) (IDK which sounds better)

1.1 Overview

Our mobile app “Cook Book” will keep you organized by allowing you to share the pictures and recipes you find with your family and friends with a click of a button. It will also offer an “add to list” feature with each post, allowing you to add the ingredients you see in a recipe to your list. This list can be shared with family and friends in a separate tab, allowing you to stay organized and quickly track all of the ingredients you need to buy at the grocery store.

1.2 Project Goals

Our goal is to make the most reliable and interactive recipe + grocery shopping app, on the market.

1.3 Definitions and Acronyms

“Cook Book” - Cook + Book

“What’s Cooking” - the name of the trending recipes page, which will incorporate images and descriptions of the current and most popular recipes on the app.

1.4 Assumptions

It is assumed that the app will be made available for the project in some phase of its completion., and will be used to provide a demo for presentations. It is assumed that the user knows how to use mobile apps on mobile devices.

2.0 Design Constraints

2.1 Environment

2.2 User Characteristics

- Users should be able to login with username and password
- Users should be prompted for authentication
- Users should be logged into a homepage and navigation should be simple
- Users should be able to create new grocery lists.
- Users should be able to add items to their lists.
- Users should be able to delete items from their lists.
- Users should be able to mark items as purchased.

- Users should be able to search for items within their lists.
- Users should be able to sort items within their lists.
- Users should be able to filter items within their lists.
- Users should be able to view their lists by store location or category.

2.3 System

- The app should be developed using a modern programming language, such as Python or JavaScript.
- The app should use a database to store user data, such as SQLite or MySQL.
- The app should use a web framework to handle user requests and responses.
- The app should use a responsive front-end framework, to ensure a consistent and user-friendly interface across different devices and platforms.

3.0 Functional Requirements

- List creation and management: The app should allow users to create and manage grocery lists easily, add new items, edit or delete existing items, and organize items based on categories or sections.
- Synchronization and backup: The app should synchronize the grocery list across multiple devices, such as smartphones, tablets, or laptops, and provide backup and restore functionalities to prevent data loss or corruption.
- Search and filter: The app should allow users to search and filter items in the grocery list based on keywords, categories, or nutritional information, and provide suggestions or recommendations based on previous purchases or user preferences.
- Barcode scanning and recognition: The app should be able to scan barcodes of products using the smartphone camera and automatically add the product to the grocery list, display nutritional information or allergen warnings, or suggest alternative products.
- Integration with external services: The app should integrate with external services or platforms, such as online grocery stores, recipe websites, or delivery services, to provide users with additional functionalities, such as price comparison, meal planning, or delivery tracking.

4.0 Non-Functional Requirements

The following are the non-functional requirements of the dynamic grocery list app:

- The app should be responsive and performant, with minimal lag or delay when adding or editing items.
- The app should be easy to use and intuitive, with clear and concise instructions for users.
- The app should be secure and protect user data from unauthorized access or manipulation.

- The app should be compatible with multiple devices and platforms, including mobile phones, tablets, and desktop computers.
- The app should be accessible to users with disabilities, such as those who are visually impaired or have limited mobility.
- The system should be scalable and able to handle a large number of users and requests.
- The system should be maintainable and easy to update or modify as needed.
- The system should be secure and protect user data from unauthorized access or manipulation.
- The system should be reliable and able to recover quickly from errors or failures.

4.1 Security

The app will be designed with security in mind, including data encryption, secure login/authentication, and protection against hacking and data breaches. Authentication and authorization: The app should provide secure authentication and authorization mechanisms to ensure that only authorized users can access and modify their own grocery lists, and prevent unauthorized access or data breaches.

4.2 Capacity

- Scalability: The app should be scalable to accommodate a growing number of users and lists, and be able to handle high traffic or peak usage periods, such as holidays or weekends.
- Performance: The app should be able to provide a fast and responsive user interface, with minimal latency or delays, even under heavy load or slow network conditions.
- Data storage: The app should be able to store large amounts of data, including user information, grocery lists, product information, and transaction history, and provide reliable and secure data access and retrieval

4.3 Usability

The app should be designed to be user-friendly and intuitive, with clear instructions and visual cues to guide users through the app's features. The app could be designed to be accessible to all users, including those with disabilities, by following accessibility guidelines and best practices. This includes features such as support for screen readers, adjustable font sizes, and high-contrast modes.

4.4 Other

5.0 External Interface Requirements

5.1 User Interface Requirements

- The app should have a clean and modern interface with a simple and intuitive design.
- The app should use a consistent color scheme throughout the app.
- The app should use clear and legible typography for all text.
- The app should use appropriate icons and graphics to help users understand actions and features.
- The app should be designed to be accessible for users with disabilities, following accessibility guidelines.

5.2 Hardware Interface Requirements

- Mobile devices: The app should be designed to run on smartphones and tablets, Android platforms, with support for various screen sizes and resolutions.
- Internet connection: The app should require an internet connection to access data and synchronize changes between devices.
- Camera: The app should allow users to scan barcodes or take photos of items to add to their lists, requiring access to the device's camera.
- GPS: The app should allow users to search for nearby stores and map out their shopping routes, requiring access to the device's GPS.
- Notification system: The app should be able to send push notifications to users' devices to remind them of items on their lists, requiring access to the device's notification system.

5.3 Software Interface Requirements

- User-friendly interface: The app should have an intuitive and easy-to-use interface that allows users to quickly add items to their grocery list, edit or delete items, and view their list at a glance.
- Dynamic and real-time updates: The app should be able to update the grocery list in real-time as users add, remove or modify items, and should also automatically update the list based on factors such as location, time of day, or user preferences.
- Multiple list management: The app should allow users to create and manage multiple grocery lists, such as a weekly meal plan, a pantry inventory, or a shopping list for a specific store.
- Integration with external platforms: The app should be able to integrate with other platforms or services, such as online grocery stores, recipe websites, or social media, to provide users with additional functionalities and personalized recommendations.
- Accessibility and customization: The app should be accessible to users with disabilities and should provide options for customization, such as the ability to change font size, color schemes, or language settings, to meet the needs of different users.

5.4 Communication Interface Requirements

(POSSIBLE IMPLEMENTATION)

- Integration with messaging apps: The app should allow users to share their grocery list with friends or family members via popular messaging apps, to facilitate collaboration and coordination.
- Push notifications: The app should be able to send push notifications to users to remind them of upcoming tasks or deadlines related to their grocery list, such as when to start preparing a meal or when to go shopping.
- Voice assistants integration: The app should be compatible with popular voice assistants, to allow users to add items to their grocery list or update it using voice commands.
- Email and SMS notifications: The app should be able to send notifications to users via email or SMS, for example, to confirm the creation of a new grocery list, notify them of special deals or promotions, or provide personalized recommendations based on their shopping history.
- API integration: The app should provide an API (application programming interface) that allows third-party developers to integrate their own services or apps with the grocery list app, for example, to provide additional functionalities, access data, or improve user experience.

APPENDICES

Recipe database: An appendix that provides a searchable database of recipes that users can browse or search for based on ingredients, nutritional information, or dietary preferences. The appendix can also include user-submitted recipes and reviews.

Nutritional information: An appendix that provides detailed nutritional information for common grocery items, such as calories, fat, protein, and carbohydrates, along with recommended daily intake and health tips.

Barcode scanner guide: An appendix that provides a guide on how to use the barcode scanner feature of the app, including tips on how to scan barcodes effectively, troubleshoot common issues, and understand barcode symbologies.

Shopping list templates: An appendix that provides pre-made shopping list templates that users can use as a starting point for their grocery lists, based on common categories, occasions, or dietary restrictions.

Price comparison tool: An appendix that allows users to compare prices of grocery items across different stores and retailers, to help them save money and find the best deals.

Grocery store locator: An appendix that provides a list of nearby grocery stores and their hours, location, and contact information, based on the user's location, along with user reviews and ratings. The appendix can also include a map and directions.

Product review section: An appendix that allows users to leave reviews and ratings for grocery items they have purchased, and read reviews from other users, to help them make informed purchasing decisions.