# Software Requirements Documentation for Recipe Finder

GiveUsA+PlsTy

Brodrick Grimm, Kalob Reinholz, Heng Tan

December 1, 2020

We have abided by the UNCG Academic Integrity Policy on this assignment.

**Table of Contents:**

1. Introduction
    1.1. Purpose: Help users find recipes using information they provide or provide them with a random recipe.
    1.2. Document Conventions:
        1.2.1. Section 1 is for introducing the overall basic information.
        1.2.2. Section 2 is for overall description, which describes the characteristics.
        1.2.3. Section 3 is for the functional requirements and the main functionalities.
        1.2.4. Section 4 is the technical requirements and the technical functionalities.
        1.2.5. Section 5 is the nonfunctional requirements, this section is for the requirements of the program.
    1.3. Intended Audience: General population and home cooks
    1.4. Definitions/Jargon: N/A
    1.5. Project scope: To allow users to select a cuisine and include or exclude ingredients and provide them with a recipe depending on their choices. If the user does not know what they want they have the option to select a random recipe. The user will also have the option to save the recipe if they like the recipe and want to cook it again.
    1.6. Technical Challenges: The first challenge we faced was learning how to connect to an API. Our second challenge was learning how to use JavaFX. The last challenge we faced was learning how to read and write to a file and then display the information to a scene in JavaFx.

1.7. References: Spoonacular API: https://spoonacular.com/food-api/docs

2. Overall Description

2.1. Product Features: Our application will allow users to save recipes, delete previously saved recipes, find a random recipe, and find a target recipe based on the users' choices.

2.2. User Characteristics: Some user characteristics include having difficulties deciding on what to cook or looking for new recipes to cook.

2.3. Operating Environment: The targeted operating environment for the Spoonacular application is at any general person's home. It should be accessible, quick and easy to run and find a brand new recipe with what criteria they want.

2.4. Design and Implementation Constraints: Our group's original intention was to allow users to enter ingredients they currently had and find the recipes based on those ingredients. However, the API we used made it too easy and we felt like the project would have not been completely fulfilled. Therefore, we decided to ask users what type of food they would want to make based on ingredients they want to include and exclude.

2.5. Assumptions and Dependencies: One assumption that our group made was that the user would already have a computer that has Java downloaded. Our application is dependent on an API that requires an internet connection.

3. Functional Requirements

3.1. The primary function of Recipe Finder will be for a user to input ingredients that they want to include or exclude, and decide on the type of cuisine that they want. Then the application will search for recipes that align with the users' selections.

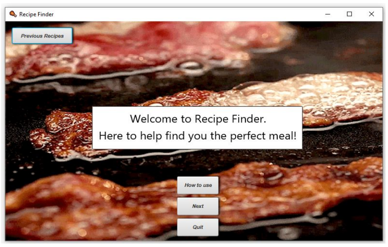
3.2. The secondary function of Recipe Finder is to find a random recipe for the user. Another secondary function is to allow the user to save recipes and delete them later if they want to.

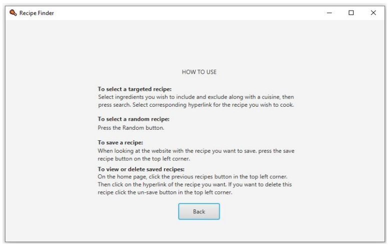
4. Technical Requirements

4.1. Operating Systems/Compatibility: Any computer that has Java downloaded should be compatible with the Recipe Finder application.

4.2. Interface Requirements

4.2.1. User Interface:


Recipe Finder
Previous Recipes
Welcome to Recipe Finder.
Here to help find you the perfect meal!
How to use
Next
Quit


Recipe Finder

HOW TO USE

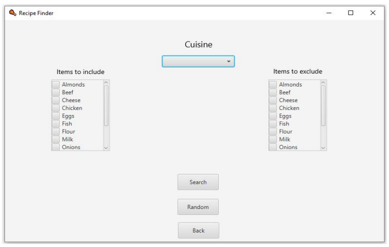
**To select a targeted recipe:**
Select ingredients you wish to include and exclude along with a cuisine, then press search. Select corresponding hyperlink for the recipe you wish to cook.

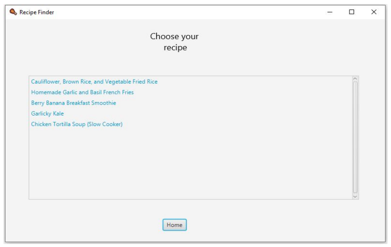
**To select a random recipe:**
Press the Random button.

**To save a recipe:**
When looking at the website with the recipe you want to save. press the save recipe button on the top left corner.

**To view or delete saved recipes:**
On the home page, click the previous recipes button in the top left corner. Then click on the hyperlink of the recipe you want. If you want to delete this recipe click the un-save button in the top left corner.

Back

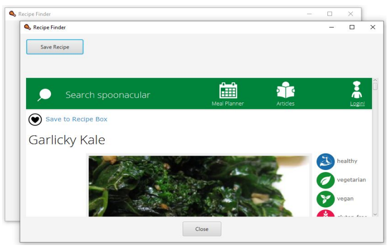


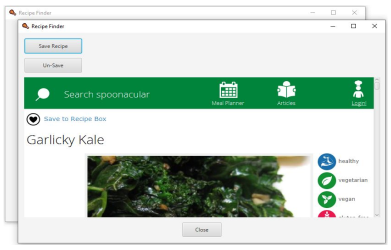

Recipe Finder

Choose your recipe

Cauliflower, Brown Rice, and Vegetable Fried Rice
Homemade Garlic and Basil French Fries
Berry Banana Breakfast Smoothie
Garlicky Kale
Chicken Tortilla Soup (Slow Cooker)

Home





When a user starts our application, the first thing they should see is the home page (top left). If the user does not know how to use the application they can click the “how to use” button taking them to the how to use view (top right). After the user reads the how to use view they would click the back button taking them to the home page. They would then click the next button taking them to the food information view where they would choose a cuisine and ingredients they want to include and exclude, or they could click the “random” button (middle left). Once the user selects their cuisine and the ingredients they want to include and exclude, they can click the “search” button taking them to the recipe choice view (middle right). When the user selects a recipe they want to cook from the list provided, or they clicked the “random” button from before, a new window will be created which will show them a website with the recipe they clicked (bottom left). If the user saves the recipe they can go back to the home page and click the “previous recipes” button and view all their saved recipes (middle right). When the user selects one of their saved recipes a new window will be created showing them a website with the recipe they clicked and an “un-save” button that un-saves the recipe from their saved recipes (bottom right).

4.2.2. Hardware Interface: Mouse/Keyboard

4.2.3. Software Interface: Java and/or Java IDE

4.2.4. Communications Interface: Our application communicates with the Spoonacular API to obtain all the recipes.

5. Nonfunctional Requirements

5.1. Performance Requirements: Performance requirement is any computer that has an updated version of Java downloaded.

5.2. Safety/Recovery Requirements: If the RecipeSaver.txt is tampered with, it could cause the application to crash. If this happens the user would have to reformat the text file or delete the contents of the text file and run the application again.

5.3. Security Requirements: N/A

5.4. Policy Requirements: N/A

5.5. Software Quality Attributes

5.5.1. Availability: Our application is available on Github, making this program publicly available.

5.5.2. Correctness: Some of the ingredients could be further defined such as fish could be labeled with Salmon, Tuna, Sardine, Mackerel, etc.

5.5.3. Maintainability: The application will be maintained by the creators to keep it up to date and running smoothly.

5.5.4. Reusability: The application can be reused if a person wants to add more features like including or excluding more ingredients or adding more features like max protein, min protein, etc. Further down the line, there could be more cuisines available as well.

5.5.5. Portability: The program should be installed and run on a computer.

5.6. Process Requirements

5.6.1. Development Process Used: The Quigley Method, on a serious note we had a scrum every week.

5.6.2. Time Constraints: August 2020 to December 2020.

5.6.3. Cost and Delivery Date: Cost is free and Delivery date is December 1, 2020.