

Group Number: 203-2

Overview of Project Features:

- Search by Image
 - A way to browse through recipes by letting images be your guide. Will display an image of a recipe along with its' name so you know how delicious it is before even looking at its ingredients.
- Recipe Display
 - Fully expand selected recipe in order to view its' requirements. User will be able to find prep/cook time, health rating, ingredient choice, and instructions for how to make the meal here.
- Favorite Recipes
 - A simple toggle star next to recipe entries that will save a certain recipe to a users' favorites folder. This can be accessed quicker and easier next time they wish to make the same thing; not to mention, saving them the headache of forgetting what the recipe was called in the first place.
- Meal Filter (Breakfast, Lunch, Dinner)
 - Depending on which meal of the day you would like to enjoy (breakfast, lunch, or dinner), you can use the meal filter to look at recipes belonging to that description. For example, if the user chooses breakfast, they will only see breakfast recipes.
- Ingredient Filter (Allergies, Food-Waste)
 - Will allow the user to filter out any ingredients based on food allergies, and gives the user the option of using their own food in the refrigerator to save costs and minimize food waste.
- Back Button
 - Will allow user to revert to previous state of application. For example, if the user had clicked on a recipe and displayed it, will go back to previous search. If in image search, will revert back to main menu with selection criteria.
- Health Filter
 - Will allow the user to find recipes based on the "healthiness" of the meal. Calories, vegetable to fat ratio, and protein used will all be factors to determine healthiness.

Search by Image

Functional requirements:

Acceptance Criteria:

User should be able to scroll through different food choices by looking at pictures.

By tapping the picture the recipe should display.

User Story:

Aaron is a hungry college student, he decides he wants to eat a breakfast item so he clicks on the breakfast option. Aaron sees many different breakfast meals such as omelettes, eggs benedict, ect. Aaron sees a picture of a delicious looking omelette and clicks on the picture. When he clicks on the feature the recipe should display giving him all the ingredients and prep work needed to make the food.

Non-functional requirements:

- Images should load in under 10 seconds.
- At least 3 images should be displayed along with a text showing what the food is.
- Use an API to generate the images with the correlating image

Recipe Display

Functional requirements:

Acceptance Criteria:

Ensure reciMe user is able to...

- Select a recipe from the search results menu
- View how well recipe matches with input requirements
- Easily view ingredients, preparation time, and instructions for cooking the selected meal

User Story:

As a reciMe user, Laura needs to view recipes on the app based on requirements she input (which meal, health stuff, etc). She'll choose whichever one from the search results menu that she likes the most, and will then use the preparation time and ingredient matching features to decide if she actually wants to cook the dish.

Non-functional requirements:

- Pull recipe, prep time, etc. from the included API
- Input requirements pulled from user profile.
- Display should load in under 5 seconds

Favorited Recipes

Functional requirements:

Acceptance Criteria:

A feature that categorizes favorited recipes

User Story:

The favorite feature involves a star icon next to the name of the recipe that is yellow when the recipe is favorited and white when not favorited. Nish would benefit from this as there are certain meals that he favors and will always cook at some point in the week. Nish simply presses the star on a recipe he really likes and then it adds the recipe to his favorites category. If Nish wants to see the recipes that he favorited, he can press the “Favorites” category and scroll down a list of recipes that he favorited. If Nish doesn’t want a recipe in his favorites anymore, he can just tap the star again. The icon then becomes white and the recipe is not in his category anymore. He can see my favorites anytime when he just wants to find something familiar.

Non-functional requirements:

The recipes will be in structs with recipe properties such as ingredients, recipe, meal name, and favorited. The favorited property can be a bool with either a true or false value. When someone taps the star icon, it can trigger a function that toggles the true or false value. If the value is true, then add it to a dynamic array of recipe structs (or a linked list) that stores favorites. If the value is false, it goes through the array and removes the newly unfavorited recipe.

Health Filter

Functional requirements:

Acceptance criteria:

Recipes displayed have correct “health value”

User Story:

Johnny has decided he wants to eat healthier to lose his beer gut. Thankfully reciMe can filter recipes based on health because Johnny isn't very familiar with what is actually healthy. ReciMe filters out the unhealthy recipes because Johnny turned the health slider all the way to the right. Johnny looks at the results and is disappointed to see only salads with vinaigrettes. He wants to eat healthier, but doesn't want to eat only vegetables and salads. Johnny turns the health slide to the middle. Various recipes are displayed with a similar calorie count and healthy proteins. Johnny sees a chicken alfredo pasta dish with many vegetables. He clicks the image of the dish and the recipe is displayed. Johnny goes to the store, gets the ingredients required, and cooks the meal. He is very happy with his meal and is glad that he used reciMe to find a healthy meal.

Non Functional requirements:

- Health factor will be based on a health algorithm.
- Slider must change the health factor filtering results accordingly.

Back Button

Functional requirements:

Acceptance Criteria:

A usable and clickable button that returns the user to the previous state in the application

User Story:

Ian just clicked on a recipe by accident when he meant to click on the **other** mac 'n cheese recipe. After all, he is no simple pleb, and he much prefers truffle mac 'n cheese to regular mac 'n cheese. Because of this, he doesn't want to go start over at the main menu; he already chose all his filters and recipe stipulations. That's where the back button comes in. Ian can now go back to the recipe search page he was just on and choose that truffle mac 'n cheese he wanted the first time around. Not only can Ian do this, but so can any other user of the ReciMe application. This should be one of the first features to aid in bug testing and ease of use.

Non-functional requirements:

- Should be silky smooth, taking you back to the previous page in under 3 seconds.
- Big enough to see but not accidentally click
- Should fit overall ReciMe aesthetic
- Can't get in the way of recipes or overlap with pictures/words

Ingredient Filter

Functional requirements:

Acceptance Criteria:

A filter that allows the user to filter out certain foods, and determine how much waste a meal will have.

User Story:

Mohammed suffers from peanut allergies. He is constantly annoyed by cookbooks with recipes that rely on peanuts, but do not taste good without them. reciMe allows Mohammed to filter out all the recipes that involve peanuts and will not entice him with the delicious images. Finally, Mohammed is frustrated by complex recipes that force him to constantly go to the grocery store. reciMe allows Mohammed to filter the recipes he has in his fridge and figure out what he can make without grocery shopping every other day!

Non-functional requirements:

- Ingredients list can be filtered out, and recipes will appear.
- User will also be able to select the ingredients that they have to minimize waste.

Project Plan:

Link to KanBan board: <https://trello.com/b/wpcrc3NV/recime>

Each color represents a different sprint except blue which is all the milestones. Group members are assigned to each feature to be worked on and the start dates of sprints are displayed on the individual tiles