

Team Agile

Recipe Manager Pitch

User Personas/Stories



Karen Smith, 41

Unemployed

Background

- She is the mom of the year
- She only eats vegan food
- Her slogan is to Eat Your Vegetables

Frustrations

- Having to manually explore through all recipes until finding a vegetarian one.
- She wants to faster recipe manager with a filtering system for vegan, vegetarian, pescatarian food



Ben Liu, 26

Personal Trainer

Background

- Gym rat
- Counts his calories and protein intake
- Needs precise nutrient amounts to stay being a shredded unit

Frustrations

- Having to manually find calorie counts on other websites outside of the recipe manager
- Wants fast recipes, so he can spend more time getting huge
- Wants meals with bigger portions to increase calories



Chad Liu, Age 21

Actor

Background

- Been acting since he was 5
- Eats what other people think is the best
- Average Cooking Experience
- Very busy but would invest time to learn how to cook

Frustrations

- Schedule is always packed
- Most recipe managers are too complex for him
- Hard to find quick top rated recipes
- Wants simplicity but at the same time appealing



Jane Lopez, Age 22

College student

Background

- Born and grew up in Spain
- A beginner food enthusiast
- Wants to add her own creation to manager
- Wants to try new & easy recipes
- Has plenty time to cook & study new recipes

Frustrations

- Many recipe managers are in English
- She finds it really hard to cook while reading instructions

Summary

In general:

- Assuming most (but not all) of the users are beginners
- Users often want the application to be fast and simple
- In addition to the basic CRUD features, users also want:
 - Categorizing/filtering features
 - Language translation
 - A rich amount of information about nutrition of specific food (calories count, etc)

Problem Statement

After considering the frustrations and challenges of the user personas, we can observe a few overlapping issues we want to solve:

- The personas struggle with complex recipe managers that do not serve their needs.
- The personas need a recipe manager to present a simplistic view of recipes and easy-to-understand controls.

In general, cooking can become an inconvenience without the right tools.

Solution and Purpose

- In order to solve this problem, we need need something that can provide the users more flexibility in their cooking, something that can make them love cooking even more.
- While we aim to make the application as simple to use as possible, we still want to make cooking more appealing and satisfying. Thus, styling will still be very important as we keep simplicity in mind.
- Our solution: YouEatWhatYouCook!
 - A simple and fast recipe manager web application that can perform all of the basic CREATE/READ/UPDATE/DELETE operations
 - On top of that, it incorporates the frustrations of our users into the application:
 - Category filters
 - Database operation
 - Language translation
 - Possibly some kind of API
 - Nutrition information
 - Manipulation of fetched data
 - and more!

Features selection based on users' specifications

Domain specific =



Must Have

Add/edit/remove recipes

Add pictures

upload your own recipes

Search Bar (by ingredients and name)

← aiming to have all of these

Second level

Reviews and comments for recipes

Filter by material

Rate and Share Recipes

sharing feature

Starred recipes (favorites tab)

← aiming to have most of these

Third level

Unit conversion:
-Degrees
-Pounds/grams

In-app timer and alarm

auto-scroll

Filter by how many person it's for

Language changer

Calories Counter

text-to-speech instructions

Filter by Diet

the ability to convert proportions in recipe to feed more

Price filters

the ability to see previously viewed recipes (history)

Sort by type of meal:
Breakfast, Lunch, Dinner, Drinks, Dessert, etc

← aiming to have some of these

Brainstorming

Sort by categories:
- Asian
- Mexican
- American

Log-in and user accounts

Filter by time and complexity of recipe

Links to where to buy ingredients or less common ingredients?

Recipes made by many people

Platforms should have accounts that can host various organizations or groups

comparison page for similar recipes

Appetite

- Given that we only have less than 5 weeks to build this application, we need to **prioritize the must-have** (CRUD) features.
- We're planning to perform 2 sprints
 - First sprint will solely focus on getting the CRUD features working
 - Second sprint will be on expanding and building additional features
- Our knowledge of the technology stack might be limited, so we are trying to keep the functionalities simple and find ways to easily implement them.

Project Timeline

— — —

Week 5: Project Proposal, Pitch, and Design

Beginning of Week 6: Get ourselves organized for agile development and make concrete github processes

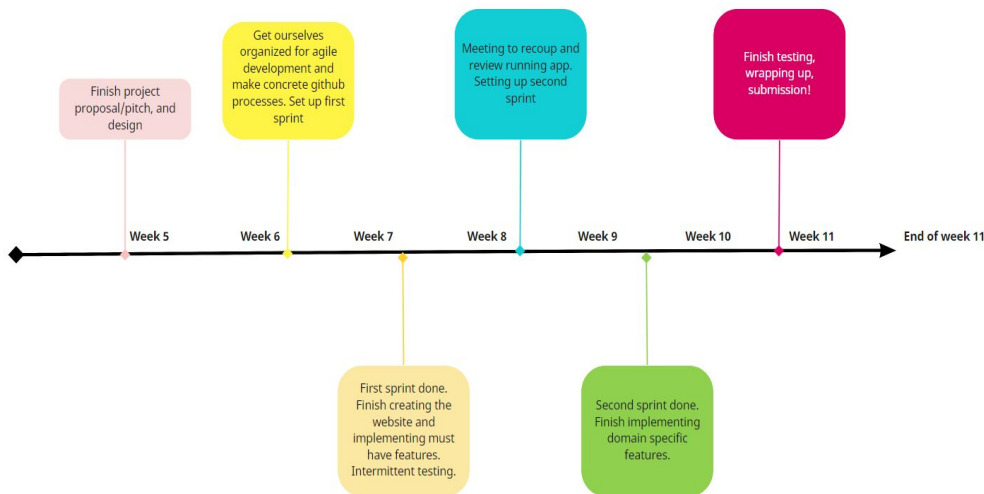
Week 6 - 7: First sprint: Creating the website and implementing must have features. Intermittent testing.

Week 8: Meeting to recoup and review running app. Setting up second sprint

Week 8-9.5: Second sprint: Implementing domain specific features.

Week 9.5-10: Testing. Wrapping up. Retrospective activities.

Project Timeline



Project Roadmap

Project Roadmap

istorm ☆





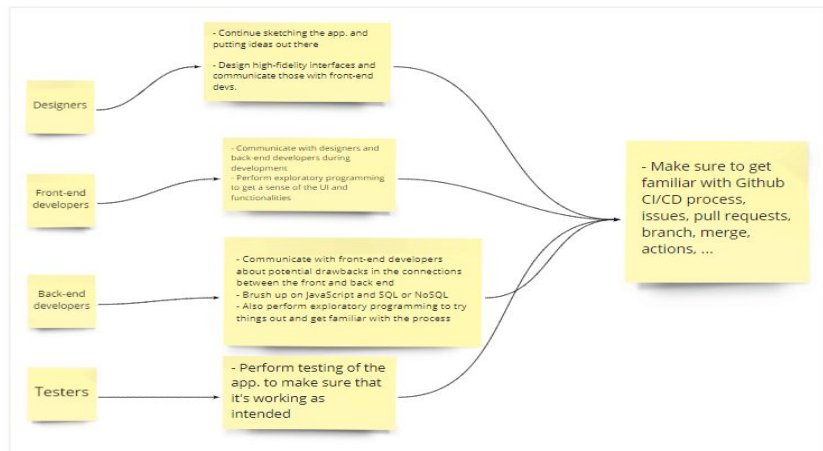








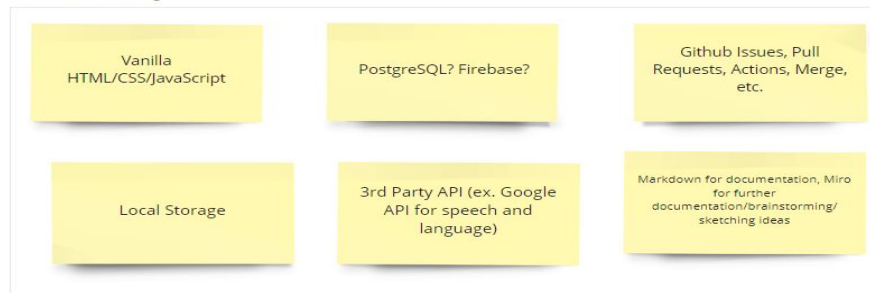

Roles



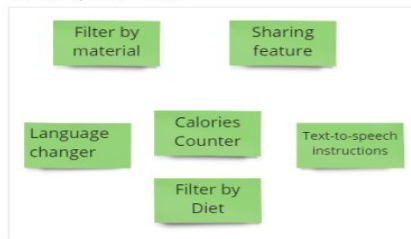
Must-have features



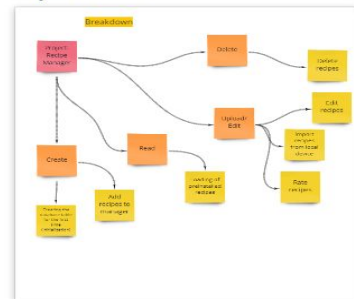
Tools and Technologies



Domain-specific features

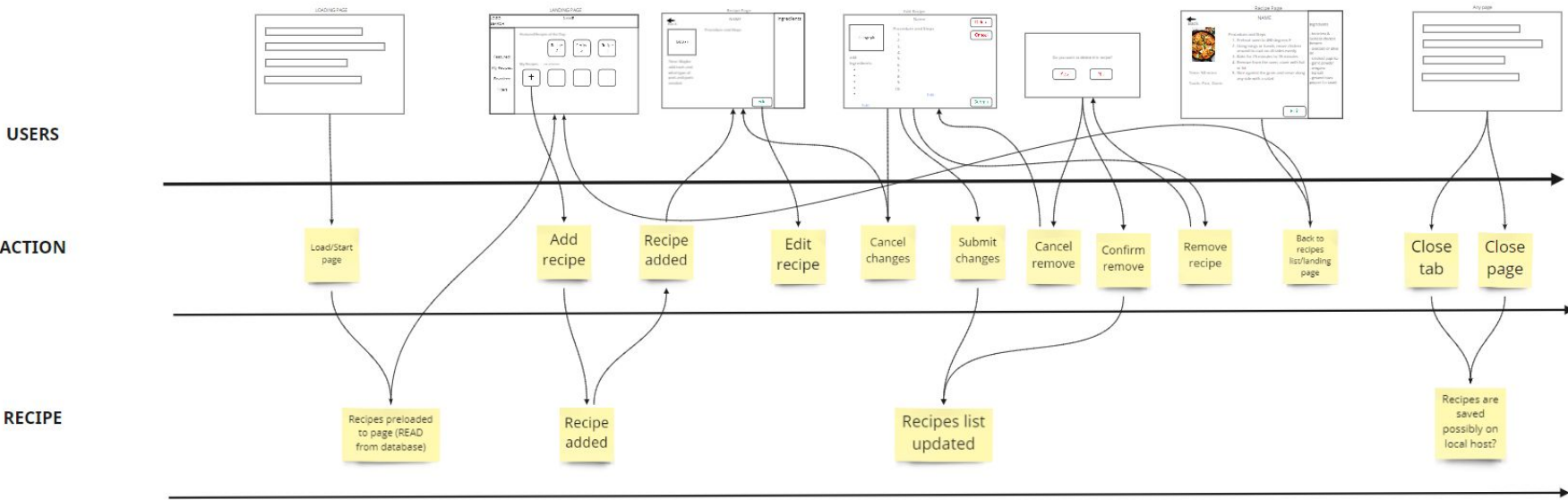


Project Breakdown

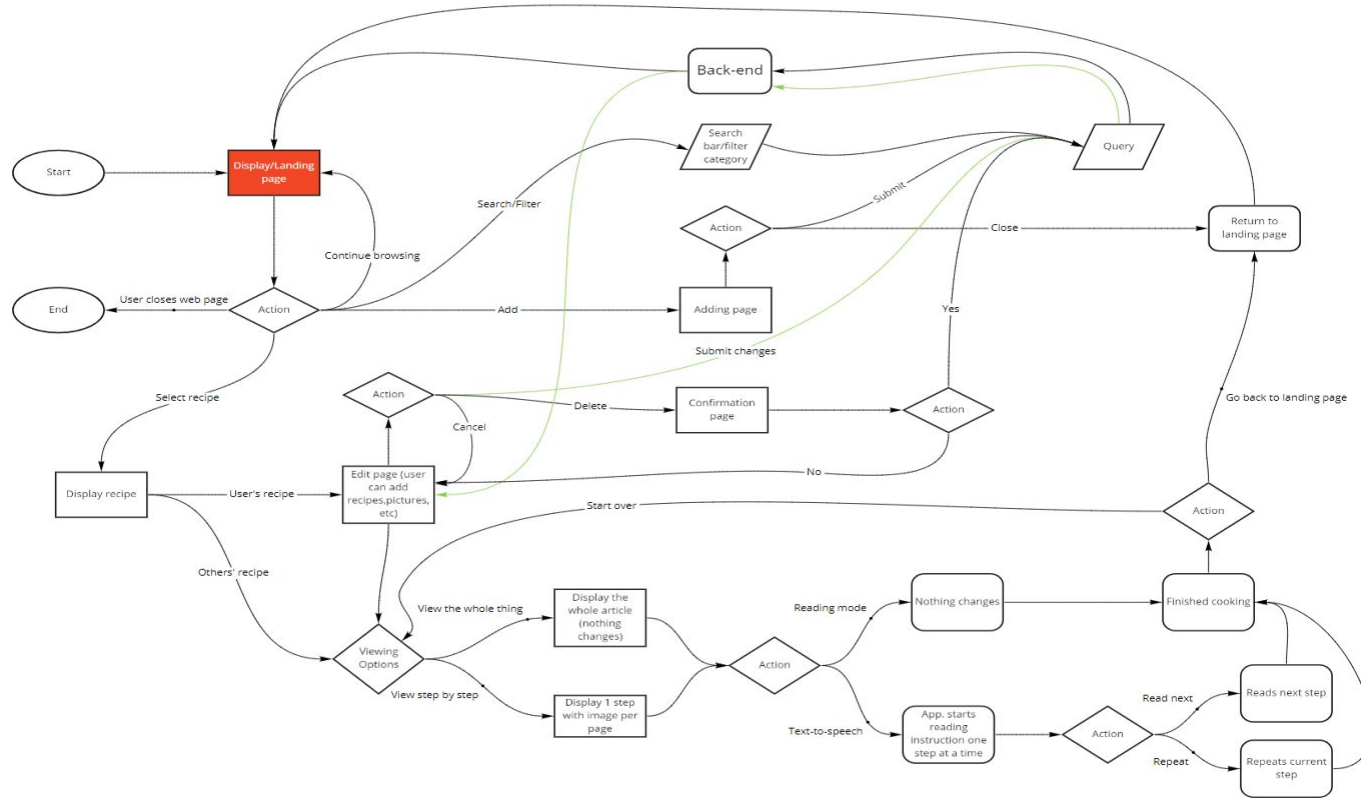


Event Modeling Diagram (User)

Event Modeling Diagram



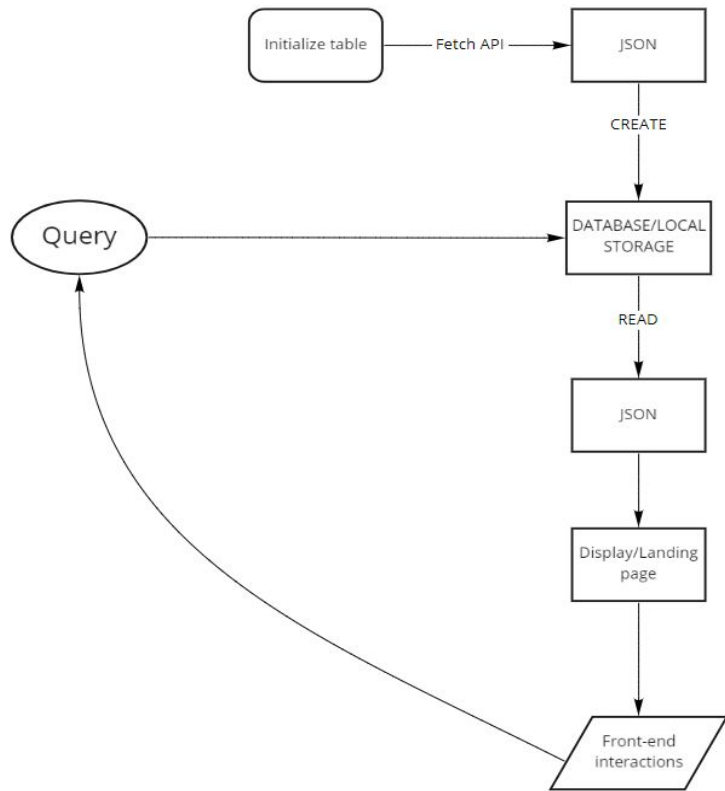
Front-end interactions flow chart (User)



Back-end/System

— — —
Data model (for one recipe):

- **id** => long int
- **name of food** => string
- **total price** => double
- **total calories** => int
- **rating** => float
- **review count** => int
- **category** => array of strings



High Fidelity Diagrams

— — —

Add Recipe

LOGO

NAME

SEARCH

import a pic

add
Ingredients:
•
•
•
•
•
•

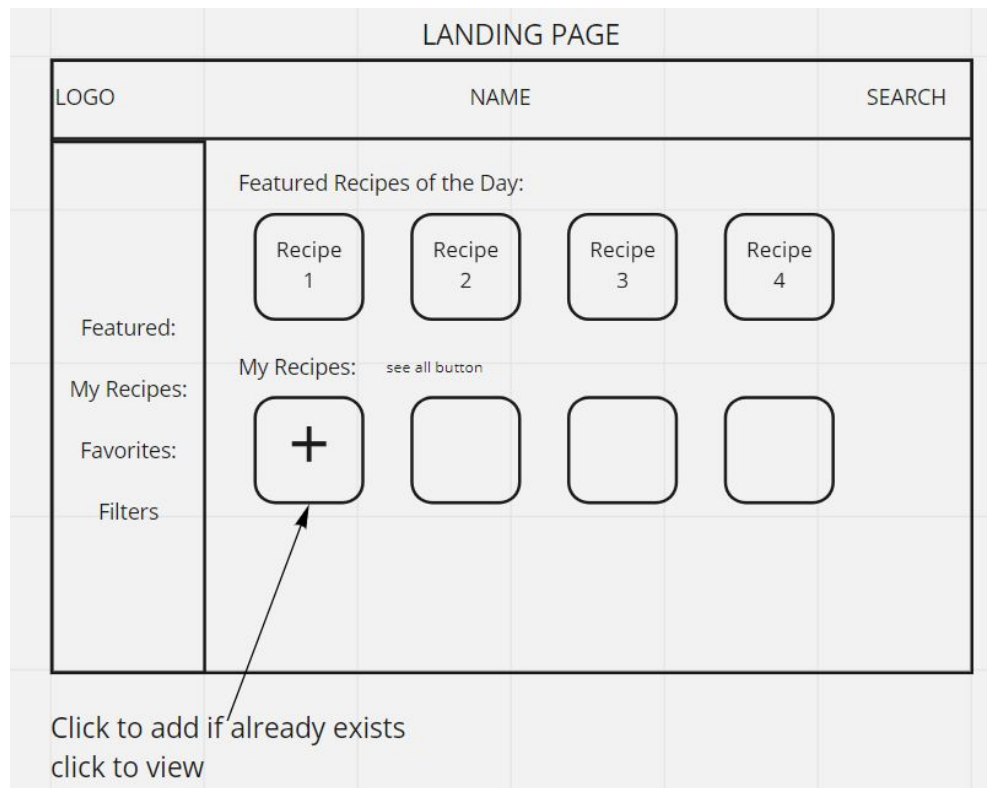
add
Procedure:
1.
2.
3.
4.
5. ...

Cancel

Submit

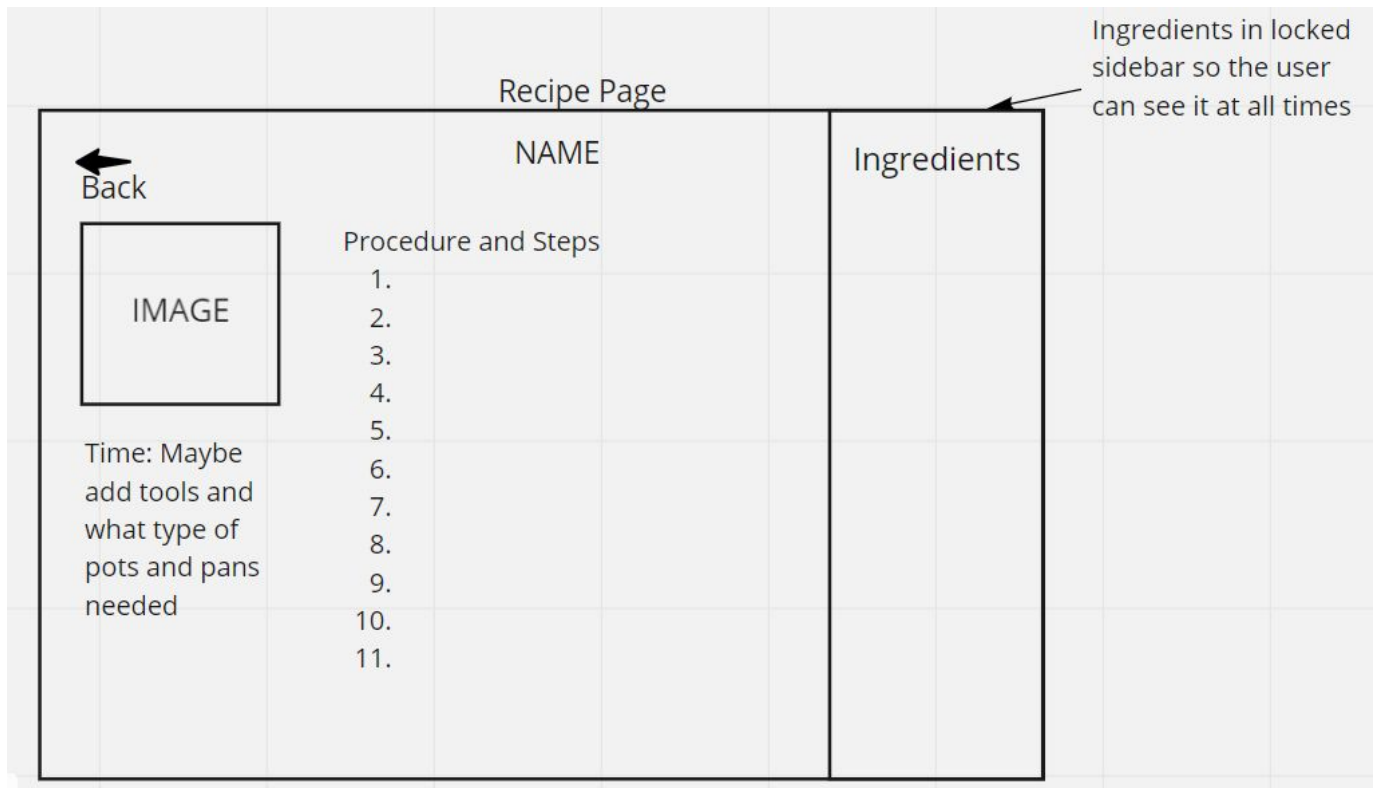
High Fidelity Diagrams

— — —



High Fidelity Diagrams

— — —



High Fidelity Diagrams

— — —

The diagram shows a web form titled "Edit Recipe" on a light gray grid background. The form is a white rectangle with a black border. Inside, the layout is as follows:

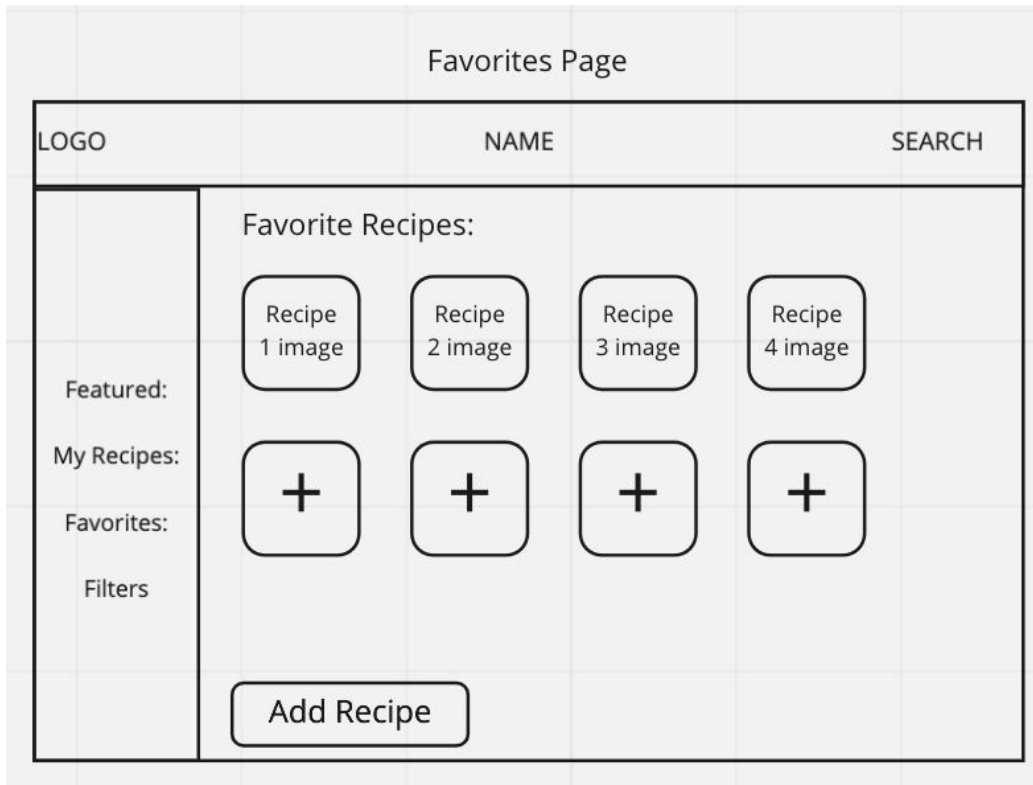
- Title:** "Edit Recipe" is centered at the top of the form.
- Name Field:** A text input field with the placeholder text "Name" is located below the title.
- Procedure and Steps:** A section titled "Procedure and Steps" contains a numbered list from 1. to 10.
- Change Picture:** A box labeled "change pic" is positioned to the left of the procedure list.
- Ingredients:** A section titled "add Ingredients:" is below the procedure list, followed by a bulleted list.
- Buttons:** Three buttons are on the right side: "Delete" (red text), "Cancel" (red text), and "Submit" (green text). There are also two blue "Edit" labels: one at the bottom left and one in the bottom right area of the form.

Annotations with arrows point to the buttons:

- An arrow points to the "Delete" button with the text: "Confirmation that you are deleting".
- An arrow points to the "Cancel" button with the text: "Confirmation you are deleting unsaved work".
- An arrow points to the "Submit" button with the text: "Confirmation you are saving and editing this recipe".

High Fidelity Diagrams

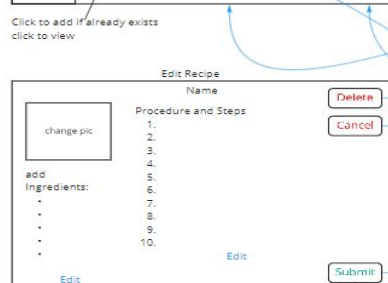
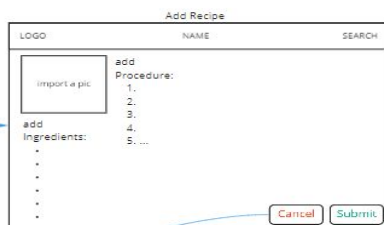
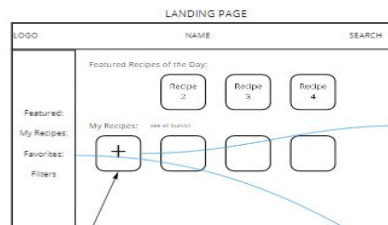
— — —



Wireframe



I used to freelance and make websites and every person that I made a website for always wanted this page to connect to the customer basically



Rabbit Holes / Risks

Risks:

- Having users able to upload their own recipes might require a login and user management, which could be a vulnerability and introduces tons of risks.
- Using external API could lead to unexpected behavior that we have no control over.

Rabbit holes:

- Too much focus on a specific filters
- Too much styling can waste a lot of time
- Trying to decide on how the user will be able to add/edit/read