The Cupcakes



Project Pitch

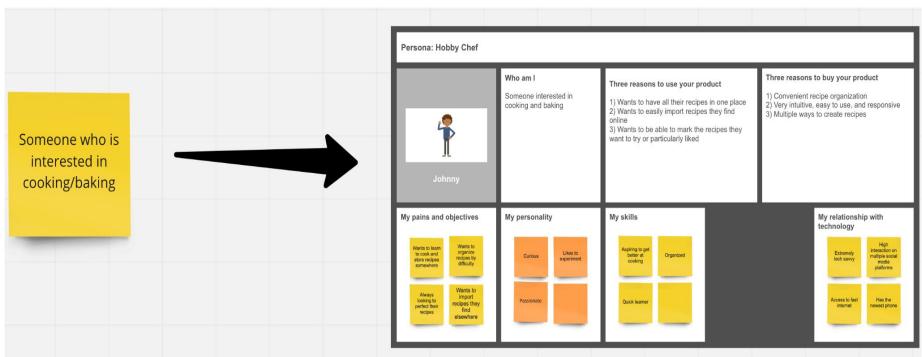
Problem Statement



The purpose of our app is to provide busy adults and people just starting to cook an intuitive and hassle-free way to organize their cooking - whether that be storing their favorite recipes or exploring new ones! We want to make cooking as easy as possible for our users.

The Users





The Users





Other Apps - User Want's

From our research into similar applications, many users emphasized that they like having filtering options, and they appreciated fast, responsive, and simple interfaces. Many users also expressed appreciation for explore features.

From this data, we want to prioritize making our app responsive, intuitive, and very user-friendly. We would also like to incorporate filtering options that make sense, as well as an explore page for users to find new recipes.

The Appetite - Our Roadmap





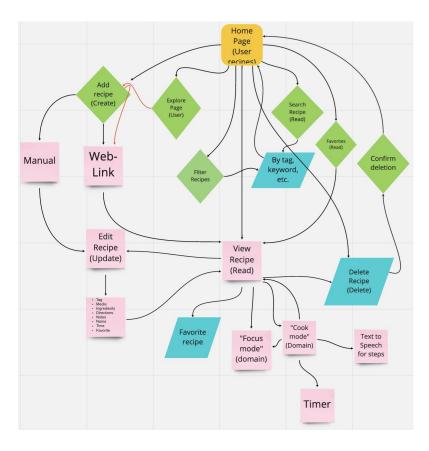
Solution - Tools & Technologies



- Design
 - Miro and Figma
- Front-End
 - JavaScript, CSS, HTML
- Back-End
 - Data storage
 - Local
 - API to get recipes for Explore page
 - Looking at spoonacularapi
 - Likely going to be using JSON ld data format (lab 6)
 - JavaScript
- QA
 - CI/CD pipeline with Github Actions
 - Cypress testing framework
 - Postman for API testing

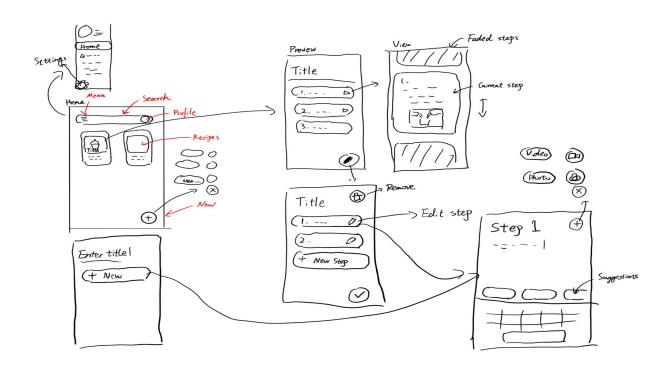
Solution - Feature Flow





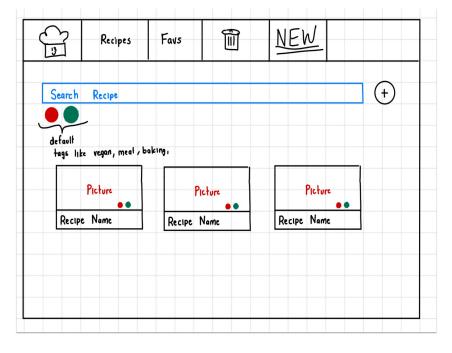
Solution - Feature Flow

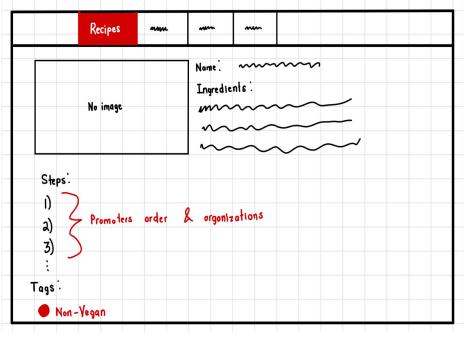




Solution - Low Fidelity

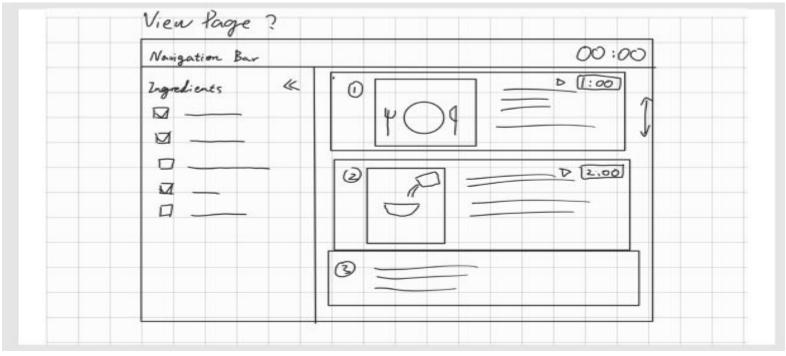






Solution - Low Fidelity





Solution - High Fidelity Prototype

https://www.figma.com/file/OcxbXEZCtuY5VOgeYYeRFY/Prototype?node-id=0 %3A1

Rabbit Holes



 Iterating on the same feature(s) week-to-week, need to make sure that we aren't going overboard on complexity, as well as making sure we thoroughly test as we go - see next slide for more details.

 General lack of knowledge/experience, no one on the team has done anything like this before, so we need to make sure that we are flexible and organized to tackle any unexpected obstacles that comes our way

Avoiding Rabbit Holes - Testing

Test Plan Template

- Type of test
 - functional, regression, end-to-end, accessibility, error condition etc.
- scope: what is covered in this single test
- · out of scope: what is not covered
- detail steps
 - how to reproduce the test
- assumption
- result
- screenshot
- priority level
- assignee

Functionality Report

For every new features added to the current version, developers would add a short but comprehensive description about the functionality and the boundary conditions for this feature.

- · developer name
- date(spring cycle)
 - · start date, end date, submit date
- · feature description
 - functionality
 - error condition
 - accessibility
- · components involved
- ideal behavior
- · constrains
- · essential level

No-Go's / Stretch Features



- Grocery List
 - While it would be nice to have, it was too far down on our priority list to justify the work it would require
- Voice Assistant Control (ala Siri)
 - While this is a great idea for hectic cooking sessions, we felt that it is too complex a problem to solve in the time we have.
- Meal Planner
 - We heavily considered integrating a meal planner into our app, but for now we don't think
 it will be doable in the time we have and we would rather prioritize making our CRUD
 features as robust as possible. If time permits, this will be the first stretch feature we
 implement.
- Online Database storage
 - This would be a super nice feature to have, but there are a lot of unknowns with it so we are going to use local storage for now.