

Let's Cook!

Amanda Hoelting, Ellison Largent, Lily Logan, Will Marceau

Software Design Specification

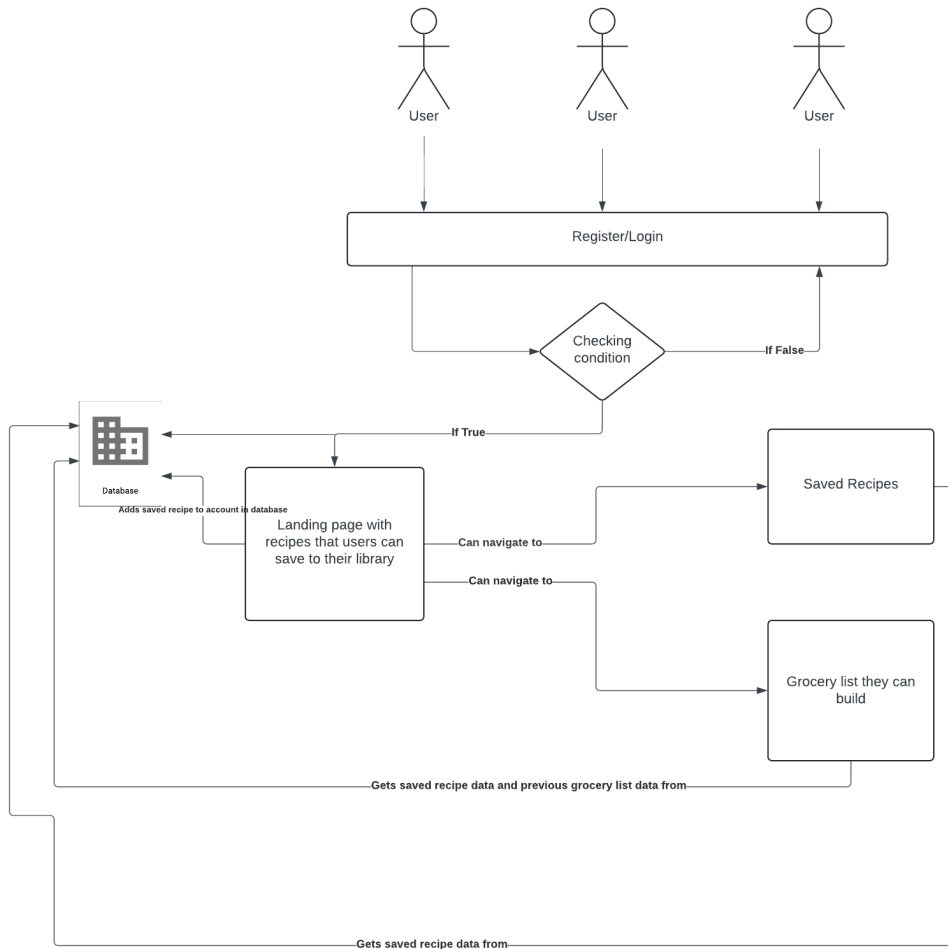
1. System Overview

1.1. Description

Choosing what meal to make is a common challenge, often leading to unnecessary takeout and overspending on food. *Let's Cook!* is a web application designed to simplify meal planning by offering a streamlined, intuitive solution to recipe selection. Rather than flipping through traditional cookbooks or endlessly scrolling through online recipes, users can respond with a simple “yes,” “no,” or “maybe” to suggested meals. This quick decision-making process not only saves time but also automatically tracks the ingredients needed for each selected meal. By swiping through options, users can easily curate a meal plan while concurrently building a personalized grocery list, all within one cohesive platform.

Let's Cook! consists of six main components: user account information, a login screen, recipe landing page, saved recipes, meal plans, and grocery list. The user account and login screen manage profiles for individual users, storing essential data such as email, phone number, and name to maintain the user's active account. The recipe landing page is where users interact with meal suggestions, determining whether to choose, discard, or consider recipes. This decision-making process sends selected recipes to the user's saved recipes section and updates the meal plan with ingredients automatically added to the grocery list. These interconnected components work together to provide a seamless experience, with real-time updates between recipe decisions and meal planning, ensuring users can effortlessly manage their cooking and grocery needs.

1.2. Design Description



List of components:

Login page - Provides a login page where users can register for an account or log in with their existing username and password.

Landing page - A page that will have one recipe on it and will allow users to swipe left if they do not want to save it to their saved recipes menu or swipe right if they do want to save it. Once they swipe, a new recipe will pop up afterwards.

Saved recipes - A page that will have a list of all recipes the user has saved. The user also has the option to create and add their own recipe here. There will be search and filter options to allow the user to find exactly the recipe they are looking for.

Grocery list - A page that will allow users to add saved recipes to their weekly meal plan and then will add their groceries to a list where they can check off the ingredients as they get them at the store.

2. Software Architecture

2.1. *coming soon*

3. Software Modules

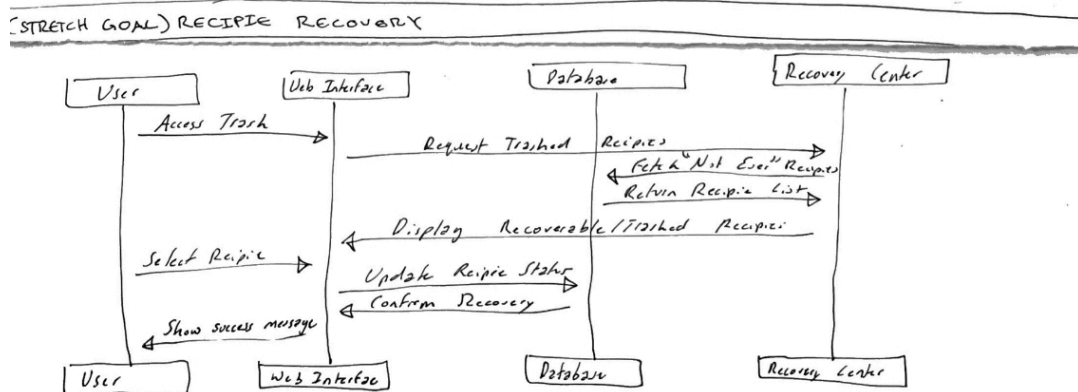
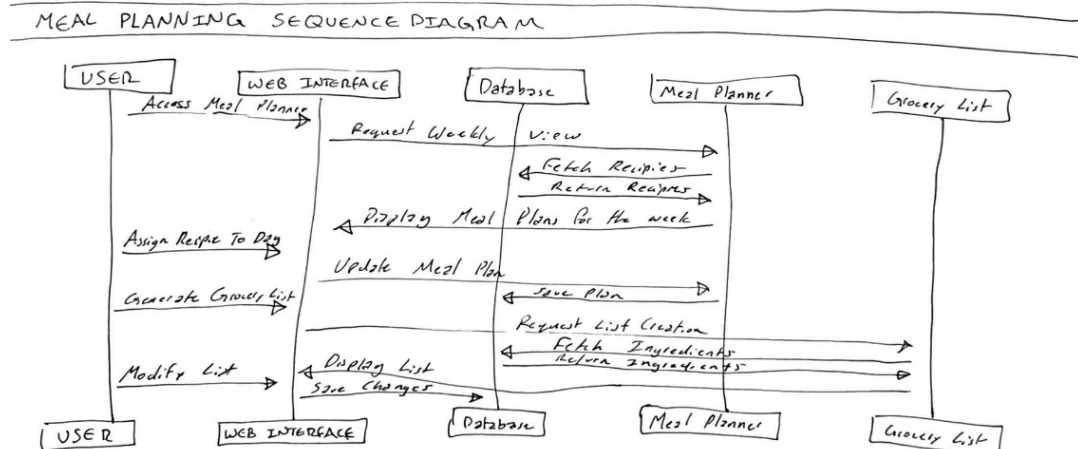
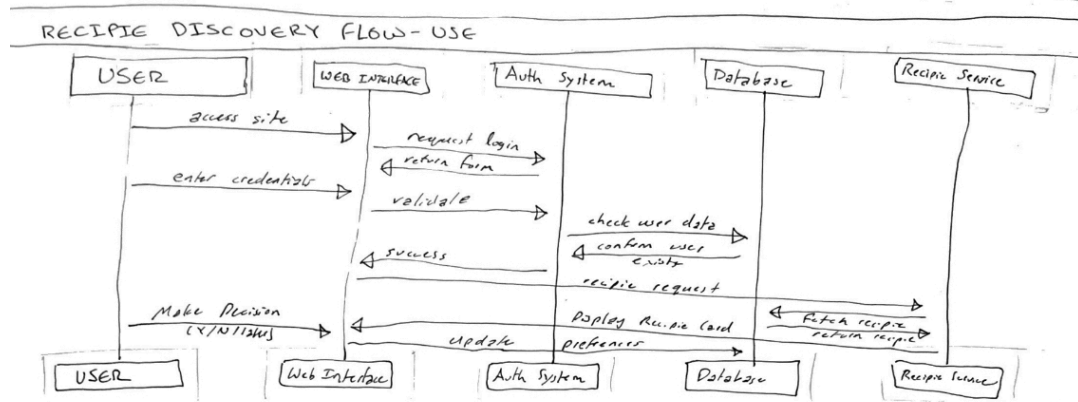
3.1. *coming soon*

3.2. <Module Name> (Include one subsection for each module.)

3.2.1. *coming soon*

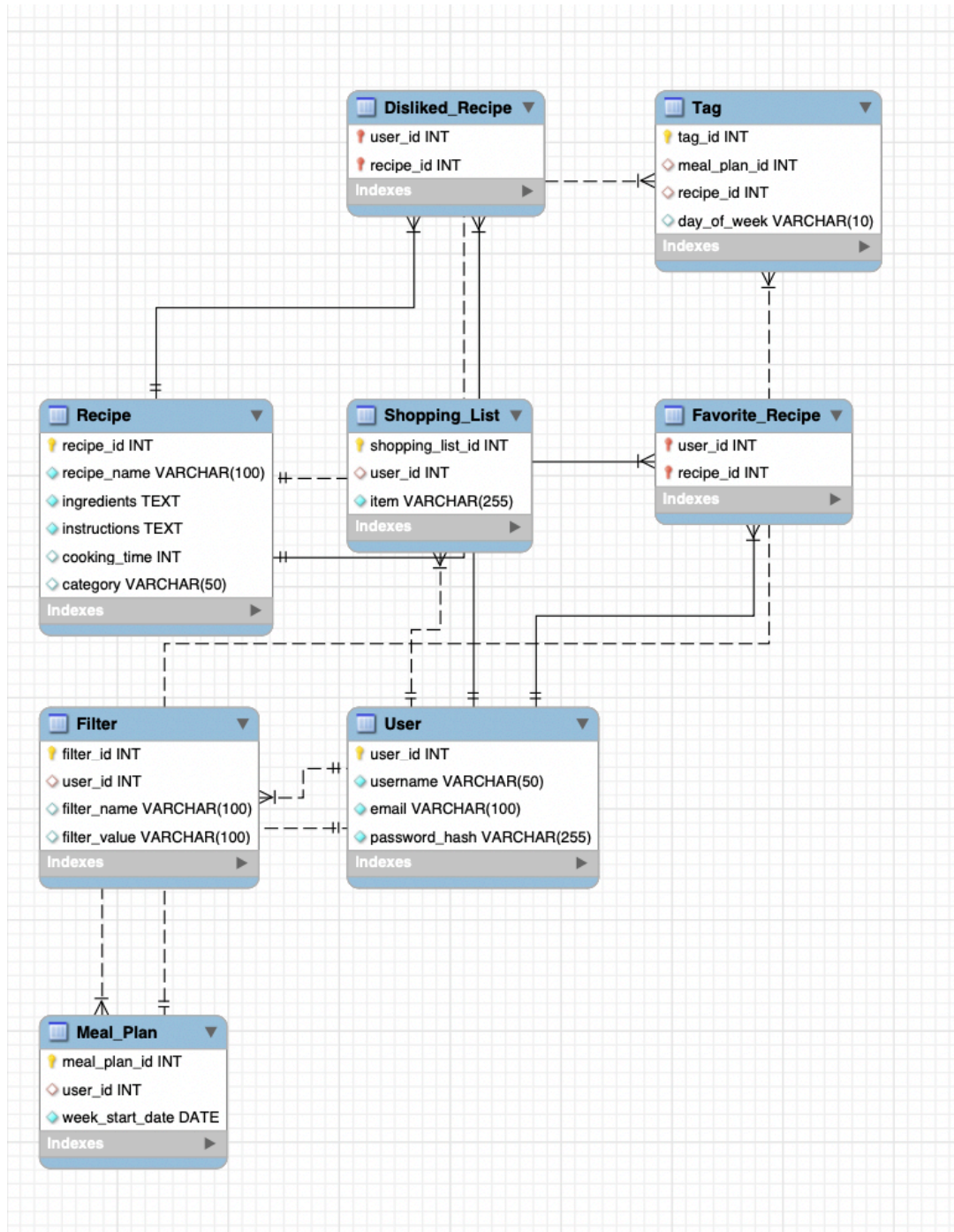
4. Dynamic Models of Operational Scenarios (Use Cases)

MAJOR USE CASES ~ UML DIAGRAMS



5. Design Underlying Database

5.1. ER Diagram of SQL Database References



5.2. Design Rationale

The design of the "Let's Cook" app database works to simplify the meal planning process by organizing user interactions with recipes in a more efficient and effective manner. The model centers around users and their ability to interact with the recipes, marking them as favorites or dislikes, tagging them for specific days of the week, and adding ingredients to their shopping list. This many-to-many relationship between users and recipes is efficiently handled through junction tables (e.g., Favorite_Recipe and Disliked_Recipe), allowing each user to have a personalized experience while minimizing data redundancy.

Additionally, the inclusion of meal planning and shopping list tables provides a practical mechanism for users to organize their weekly meals and streamline grocery shopping. Each user can create weekly meal plans, and within these plans, they can tag recipes to specific days, aligning with the goal of making decision-making easier. The schema design enforces relationships through foreign keys and cascading deletes, ensuring data integrity by automatically removing associated records if a user is deleted. By leveraging this relational model, the app maintains flexibility, scalability, and clarity, allowing possible stretch goal expansion in the future with features like custom recipe creation and advanced filtering without requiring major structural changes.

6. References

7. Acknowledgments