

ECE 49595 Senior Design Proposal

Group

Tony Ni

ni86@purdue.edu

Javon Timmerberg

jtimmerb@purdue.edu

Varun Dengi

vdengi@purdue.edu

Goals

Our goal is to design an application that will help users learn more recipes that they can cook at home and simultaneously make either healthier or dietary choices that are more inline with their lifestyle goals. One of the main appeals of being able to order takeout is the wide variety of options available that a customer can scroll through, and we hope that giving customers the option to visualize more options that they can make at home, they will be more motivated to cook at home and spend less by ordering takeout less.

Idea

We are proposing a web/phone application that prompts a user for desired ingredients and returns back a list of recipes that match the highest amount of ingredients. We store the recipes in a database and match users with potential recipes for a given list of desired ingredients. The user can then swipe for desired recipes that are then dropped into a list of recipes that they can view at any point. A user can then choose a specific recipe they swiped on previously and begin tuning the amount of ingredients needed to cook the recipe for a specific number of meals. After choosing the meal size, a user can change the inclusion of specific ingredients and compare the calorie and macromolecule amounts in real time if they desire. Finally, the program will give the user the full recipe with instructions scraped from the internet.

Proposed Frameworks and Components

We will use TypeScript to build the Scraper, the Backend Service, and the Frontend. Typescript is a typed language based on Javascript, giving us the ability to have precise definition of any structure we create. Having strict types will allow us to build and refactor our code easier because the shape of all objects are defined.

Frontend

For the frontend we will use Figma to blueprint our UI/UX design and iterate through different designs to choose one that will represent the product the best. We will use the React Framework to build the frontend and we will use TypeScript as our language. We will use React because it is the most popular frontend framework in the industry and the most beginner friendly and we will use TypeScript because it is JavaScript but with typing which can help us build better coding semantics.

Backend

For the Backend Service we plan to build it using ExpressJS and NodeJS and with TypeScript. ExpressJS and NodeJS are the most popular frameworks for setting up API's because it is fast and easy for beginners to learn and set up. We are using TypeScript again because it gives us the option of typing on top of JavaScript's set of features.

Database

For the Database we are planning to use AWS DynamoDB to store our information. DynamoDB is a NoSQL document store database which gives us flexibility in storing our data. Furthermore DynamoDB has support built by AWS which will make setting up much easier for us. This will be the data types in our database:

Data types

User

1. User id (PK): string
2. Name : string
3. Liked recipes : string[] of recipes id (FK)

Recipe

1. Recipe ID (PK): string
2. Name : string
3. Ingredients : string {} ingredient ID (FK) is key, value is amount per ingredient
4. Recipe steps : string []
5. Number of Meals: num

Ingredient

1. Ingredient ID (PK)
2. Name: string
3. Macros : ND[]

Data structures

Nutritional Data (ND)

1. Cal
2. Protein
3. Carbs
4. Fat

Deliverables

- Database schema for storing recipes and user information
 - Initial set of recipes, ingredients, and users for development
- Set of basic APIs to be able to reliably communicate with backend database
- Frontend user interface

User Experience

1. User launches the application.
2. User is asked to sign up or login.
3. User goes to the home page.
4. User has a view of all their recipes that they've previously liked.
 - a. This is separate from recipes that they have swiped on in previous queries.
5. User has a page to enter in ingredients.
6. User has a button to start swiping on recipes.
7. After swiping on recipes, the user can view a list of all of the recipes they swiped on in this current query.
8. The user can then select a recipe and adjust the portion size of it.
9. User has the ability to see the macros of the recipes and adjust specific ingredients.