Functional Specification

By: Corina Geier, Jeffrey Lai, Edward Lou, Sai Muktevi & Andrew Zhou

RecipEat

Table of Contents

Background

Figuring out what to eat based on the food you have in your fridge can be a difficult task for many. This task is made even harder when someone is trying to eat healthy. RecipEat aims to solve this problem by providing healthy recipes that can be made with the food someone has on hand. RecipEat works with a user's dietary constraints and nutritional goals to provide recipe recommendations that are the most relevant to the user. RecipEat also allows for a visual comparison of recipes so that users can quickly decide which recipe they would rather make.

Goals

- Create an enjoyable user experience
- Make it fast and simple to find recipes based on the ingredients a user has at home that meets nutritional targets
- Allow users to keep track of and update the food they have at home using a bag of ingredients
- Construct a user authentication system so user information and preferences are saved

User Profile

Our target audience is anyone who cooks at home and has a desire to eat healthy meals. Our target users would have the ability to browse the web, but not much else technical experience would be needed. Users can have all levels of cooking experience as they can choose recipes that fit their level of experience.

User Personas

1.

	Max 45 years old Working dad
Daily Routine	Get kids ready for school — Go to work — Exercise — Make dinner for family
Goals	• Wants to cook healthy meals for his family so that him and his kids are getting the nutrients they need • Wants to cook meals based on the food in the house due to his busy schedule
Technical Ability	• Can easily browse the web and use a computer • Not as technically savvy on a phone

2.

	Emily 20 years old Student
Daily Routine	Go to school — Socialize — Make dinner and prep food for tomorrow — Study
Goals	• Find easy recipes to make that are healthy • Save money on food by prepping meals at home
Technical Ability	• Very technically savvy

Data Sources

Our data source and the different entities involved will be populated while the user interacts with the application. The main source of data to populate our database will be the Spoonacular API. Several API calls made during the user interactions will follow the database Entity Relationship (ER) structure as shown below.

DATABASE SCHEMA (SQL)

User
user_id: str
name: str
email_id: str
height: int
weight: int
age: int
gender: str
diet: str
intolerences: str

Bag of Ingredients
id: serialize primary key
user_id: str foreign key
ingredient: str
ingredient_name: str
amount: str
units: str

primary key is bold
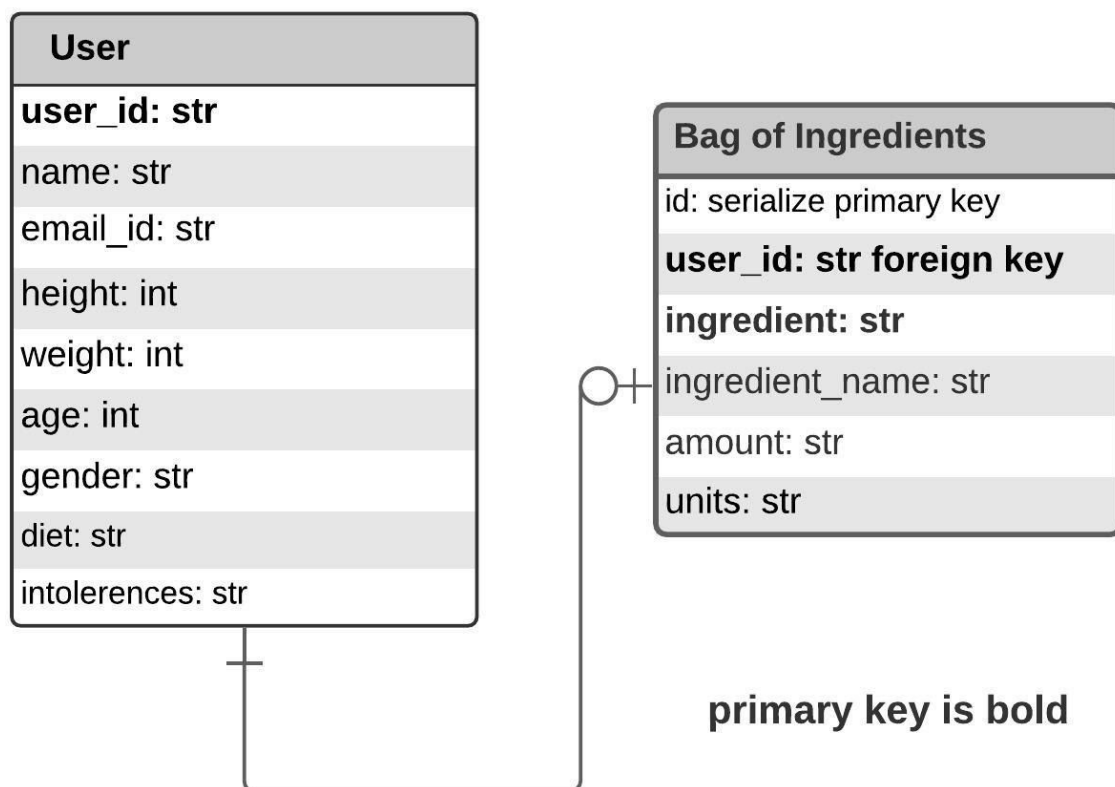
The main entity tables are:

1. **User**: stores details regarding the user
2. **Bag of Ingredients**: maps the ingredients to a User ID,

The database for user management and authentication is hosted on the google cloud service in a Firebase Realtime Database system using Firebase Authentication. As for

the database maintaining both User details and Bag of Ingredients we're using PostgreSQL.

Use Cases

1. Emily wants to make a quick meal with the ingredients she has

Emily will log on to RecipEat and update her bag of ingredients. This information will be saved and she will start looking for recipes that she can make solely with the ingredients she has on hand. RecipEat will return the recipes that match up most closely with the ingredients she has and she'll be able to search through these recipes and choose one to make.

2. Max wants to prepare a nutritious meal to fulfill dietary requirements for him and his kids

Max goes to RecipEat to search for some recipes he can make for dinner that night for him and his kids. Since one of his kids is lactose intolerant he will make sure and mark this dietary constraint in the system. Next, he will input some nutritional target he has, such as the desired level of protein and a maximum amount of sugar. RecipEat will take all of this into consideration and return recipes to Max that do not contain dairy products and are the closest nutritional match.

3. Max wants to view a visual representation of a couple recipes to compare them to find a new recipe to try out.

Now that Max has found a few recipes that fit into his dietary and nutritional constraints he wants an easy way to compare them to decide which recipe to use. Max uses RecipEat's visual comparator to view a side by side display of two recipes he is deciding between. This view will show him the differences in ingredients and in nutrients so he can easily see the tradeoffs between the two recipes. After discovering that one recipe has less sugars than the other and requires fewer ingredients he will decide to cook the recipe of choice.

Appendix

Old Version of our Database Design:

Archived the old database design due to simpler design of the Entity Relationship in our current database.