

**Team Mango: Aryaman Goenka, Sadid Ethun, Haotian Gan**

SoftDev

P01 -- Design Doc

12/8/21

Time Spent: 5 hours

**Concept:** Food Network Website

- Discover new recipes and restaurants
  - Recipe search bar (**Spoonacular API**)
  - Restaurant Search Bar (**Yelp Fusion API**)
  - Save **Spoonacular API**'s recipes to your personal library, your "cookbook".
  - Each recipe comes with ingredient nutritional information hover-overs
    - Detailed Nutritional Breakdown: (**USDA API**)
  - Social Media-esque site:
    - Each user has a page similar to Instagram's setup
    - Your favorite recipes and restaurants are publicly viewable on your profile
    - You can make posts about what you've made recently

**All APIs Used:**

- **Spoonacular API (Recipe search)**
- **USDA API (Nutritional breakdown for ingredients)**
- **Yelp Fusion API (Restaurant Search, Image of restaurant)**

**Site Components**

- Login
  - Verify user exists using flask form or confirm new username and password
  - Give cookies to keep a session running
  - Client-side verification that username/password meet constraints
  - After, redirect to Main/Landing Page
- Main Page
  - Nav Bar
    - Search page link
    - Personal user posts link
    - Link to page with user's favorited recipes, restaurants, and posts
    - Personal recipe book link
  - A feed with user posts (Will look the same as Instagram)
- Personal User Page
  - Each user has a page that looks like an Instagram page
  - Three sections:
    - User Posts
      - Posts require an image and a description, just like on Instagram.
      - Posts have a comment section.
      - Posts are shown on the home page

- User recipe book
    - The user's "cookbook". Everything that the user has deliberately chosen to save.
  - Favorite restaurants,
    - A list of the user's favorited posts and restaurants
- Search Page (Two sections)
  - Find Recipes (Spoonacular API)
    - Search by:
      - Diet Restrictions
      - Cuisine
      - Arbitrary word query
  - Find Restaurants (Documenu API)
    - Restaurants are automatically suggested to the user based on their saved, searched and posted recipes based on the cuisine.
    - Manual search functionality
      - Cuisine
      - Menu Item
      - Location
        - Key words and cities

### DBs (SQLite3)

- User DB
  - One Entry per User
  - Values: Username, Password
- User Personal Information
  - Foreign Keys Referencing to User DB
  - DB for user personal info
- User's favorite restaurants, recipes, and posts

### Database Organization

#### User DB

| UserID (INTEGER PRIMARY KEY) | Username (TEXT NOT NULL) | Password (TEXT NOT NULL) |
|------------------------------|--------------------------|--------------------------|
| 0                            | user                     | pass                     |
| 1                            | user2                    | pass                     |

#### User Personal Information

| Foreign Key (a | Preferred | Dietary | Display Name | Personal Page |
|----------------|-----------|---------|--------------|---------------|
|----------------|-----------|---------|--------------|---------------|

| reference to user<br>DB UserID) | Cuisine | Restrictions |              | Description   |
|---------------------------------|---------|--------------|--------------|---|
| 0 (user)                        | Italian | Pescatarian  | FeedMeCheese | Avid chef<br>interested in<br>Italian cuisine<br><br>Columbia 22' |

#### User Favorite Restaurants

| Foreign Key (a<br>reference to user DB<br>userID) | Cuisine | Location                                      | Other Documenu API<br>data... |
|---|---------|---|-------------------------------|
| 0 (user)  | Fusion  | 345 Chambers St,<br>Stuyvesant High<br>School | ...                           |

#### User Favorite Recipes

| Foreign Key (a<br>reference to user<br>DB userID) | Cuisine | Description     | Ingredients               | Instructions               |
|---|---------|-----------------|---------------------------|----------------------------|
| 0 (user)  | Indian  | Delicious Pasta | [array of<br>ingredients] | [array of<br>instructions] |

#### User Posts:

| Foreign Key (a<br>reference to user<br>DB's userID) | postID   | Image Link  | Description               |
|---|----------|---|---------------------------|
| 0   | 12321398 | <a href="https://imgur/...">https://imgur/...</a> | Made some apple pie<br>:) |

#### Post Comments:

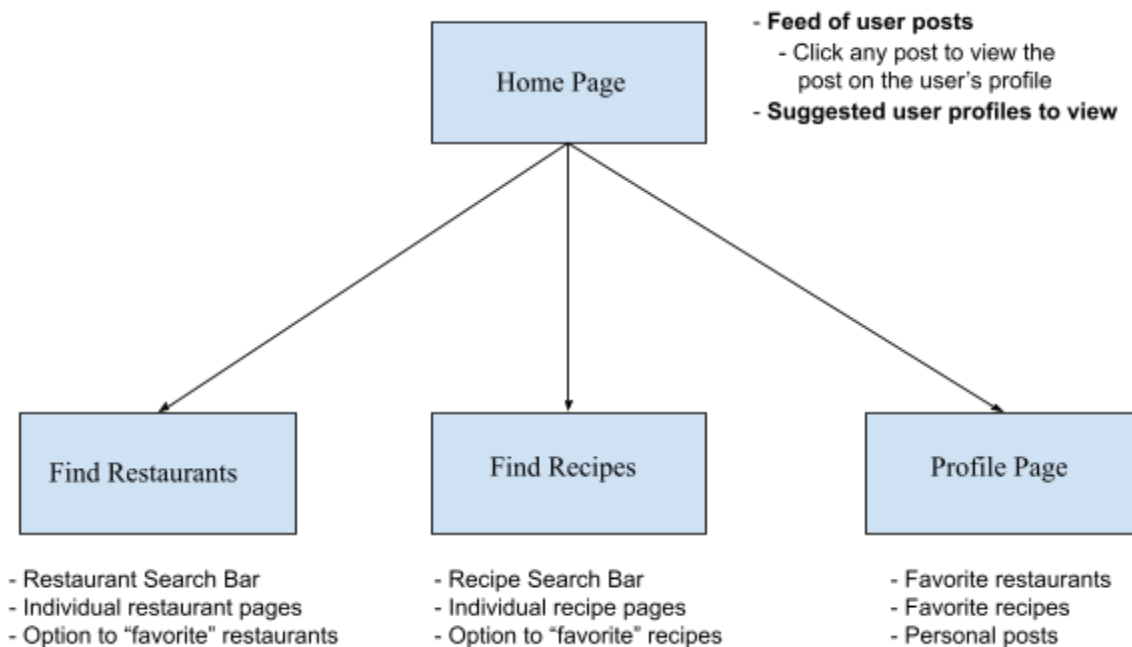
| Foreign Key (a reference to<br>user DB's user ID) (The<br>commenter) | Foreign Key (postID) (the<br>post being commented on) | Comment                      |
|--|---|------------------------------|
| 0  | 12321398  | This is such a great recipe! |

## Code Components

1. Flask
  - a. Hosts and routes the website.
  - b. Interacts with APIs on the backend, allowing for the functionality of the API.
2. Sqlite3
  - a. Stores results from API to minimize calls to API (storing data on the recipes users have favorited)
  - b. Stores User Data
3. CSS/JS (Bootstrap)
  - a. Used for styling and interactive elements on the frontend.

## Roles

- Aryaman Goenka (PM): DB, Flask Routing, Find food facts function
- Sadid Ethun: Html, Bootstrap, Find restaurants function
- Haotian Gan: User pages, restaurant suggestions, recipe search page



Home Page (/) (GET)

- Feed of user posts
- Links to suggested user profiles to view

Login and Registration (/login and /register) (POST and GET)

- Login and registration

Logout (/logout) (GET)

### **User Profiles**

/user/<string:username>/saved\_restaurants

/user/<string:username>/saved\_recipes

/user/<string:username>/user\_posts

### **Create, Update, Delete, Read (POST endpoints)**

/api/update\_user\_info

/api/make\_comment

/api/delete\_comment

/api/update\_user\_profile\_picture

/api/make\_post

/api/delete\_post

/api/is\_recipe\_favorited

/api/favorite\_recipe

/api/unfavorite\_recipe

/api/is\_restaurant\_favorited

/api/favorite\_restaurant

/api/unfavorite\_restaurant

/recipe/<int:recipe\_id>

/restaurants

/restaurants/search

/restaurants/view

/recipes/search

Future features (if time allows)

- Ability for users to create their own custom recipes.

**Target Ship Date - 1/03/22**