

ProjectOI: ArRESTed Development

TNPG: APInsomniacs

Roster: Amanda Tan, Naomi Lai, Jacob Lukose, Kishi Wijaya

TARGET SHIP DATE: 2024-12-20

Ship Date: 2024-12-17

DESIGN DOCUMENT v.1

A. Description

- I. This project implements a website all about food, with the following:
2. Visitors will be able to access:
 - a. Catalog of foods with nutritional information, and product searches
 - b. Catalog of recipes
 - c. Page of recent food news
3. After registration, logged in users will be able to:
 - a. Find grocery stores near user
 - b. Favorite catalog items
 - c. Write notes on catalog items visible to user

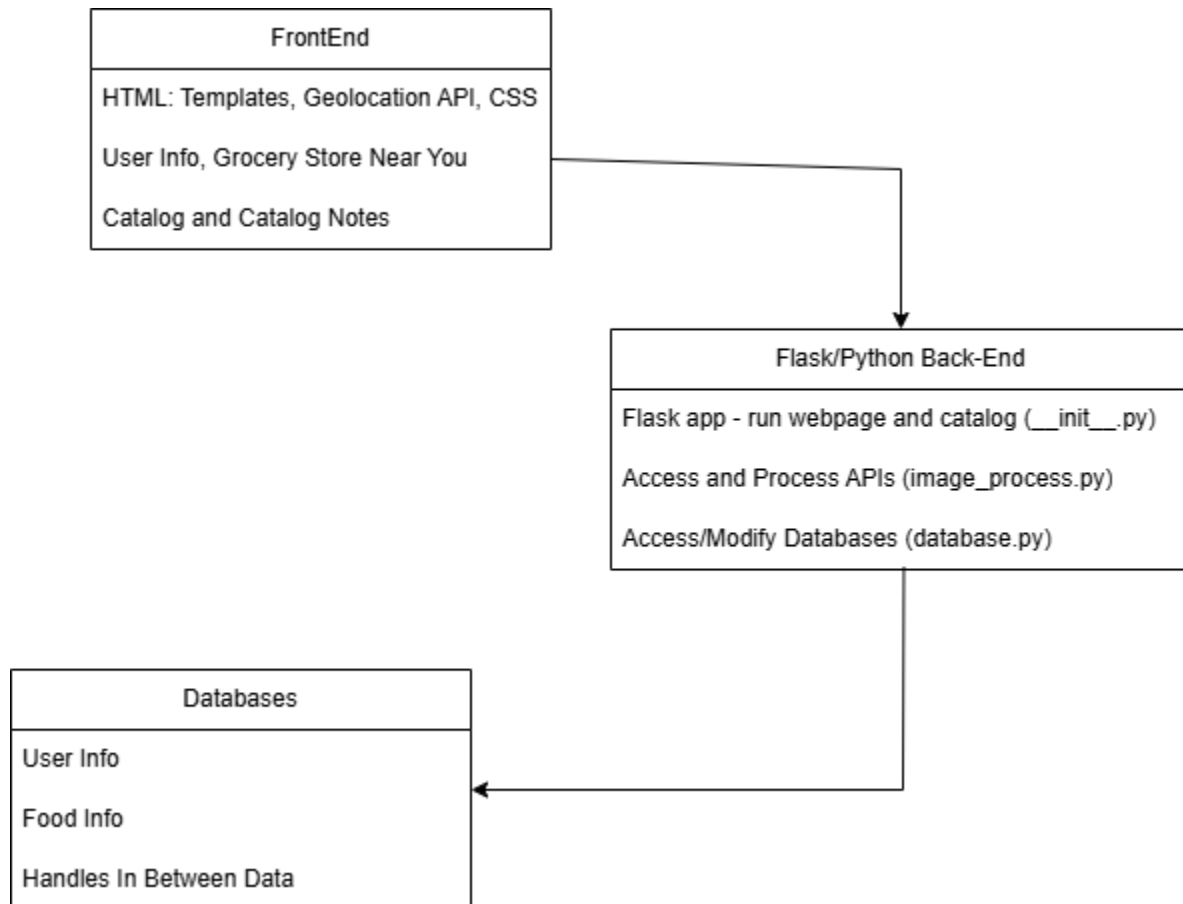
B. Functionalities

- I. Management of Individual Favorites Page:
 - a. Ability to favorite and unfavorite information on foods, recipes, news article entries
 - b. User ability to leave notes to personal reference
2. Home Page: Page with all the existing food entries
 - a. Uses links to bring to each page

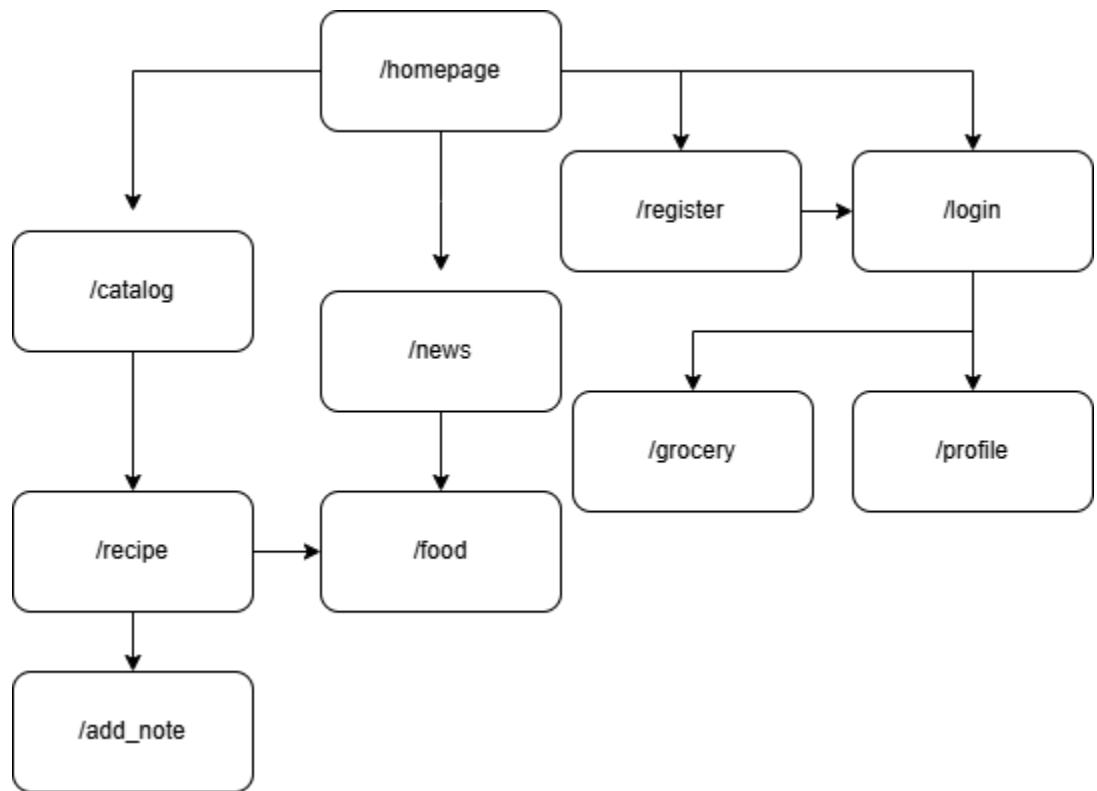
C. Program Components

- I. Flask/Python:
 - a. Serves the websites, handles HTTP requests, manages user sessions and authentication (login)
 - b. Interacts with various tables within the database file to fetch and store user and API data
 - c. Coordinates Flask routes (between HTML pages) and middleware functions (ex. Module for storing data into tables, creating user sessions, etc) to control application flow
2. SQLite3 (Database):
 - a. Stores user information (usernames, passwords, etc)
 - b. Stores data about each food page in a unique table
 - c. Data used by middleware functions and HTML templates

3. HTML / Jingo: Provides user interface with links to access different pages
 - a. Renders content passed by Flask routes/database
 - b. Displays form for user interactions and works with Python to create the various actions
 - c. Used by Flask routes to have pages with default formatting (keeps things consistent and makes it easy to create pages for new content)
4. CSS / Bootstrap: Create a consistent theme for pages across the board
 - a. Works with HTML documents



D. Site Map



- I. /: (homepage) displays links and previews of related content
2. /catalog: (catalog of recipes) includes previews of recipes + image
3. /news: previews of articles + images
4. /grocery: nearest grocery store
5. /food: individual food item displaying nutrition information etc.
6. /recipe: individual recipe instructions etc. + user notes + option to favorite item
7. /add_note: user's notes of individual recipe
8. /article: content of article
9. /login: user login
10. /register: create unique username + password
- II. /profile: user's favorited items

E. Database Organization

- I. users

id	username	password
----	----------	----------

INTEGER PRIMARY KEY AUTOINCREMENT	TEXT UNIQUE NOT NULL	TEXT NOT NULL
--------------------------------------	-------------------------	---------------

2. recipes

id	ingredient	content
INTEGER PRIMARY KEY AUTOINCREMENT	TEXT NOT NULL	TEXT NOT NULL

3. food

id	name	image	recipes
INTEGER PRIMARY KEY AUTOINCREME NT	TEXT NOT NULL	BLOB	REFERENCE recipes(content)

4. news

id	title	content
INTEGER PRIMARY KEY AUTOINCREMENT	TEXT UNIQUE NOT NULL	TEXT NOT NULL

5. favorite_recipes (each user gets table)

user	recipe
REFERENCE users(username)	REFERENCE recipes(id)

F. APIs

- I. [Spoonacular](#)
 - a. This food API provides information on ingredients, recipes, nutrition, allergens, and more.
2. [New York Times API](#)
 - a. This API provides information such as monthly statistics on stories, NYT article database, and bestselling books.

- b. We plan to use the 'Top Stories' section and filter for the food category to access recent food-related news.
- 3. [Location API](#)
 - a. This API returns information about the location and details of various Kroger grocery stores, including department and store hours.
- 4. [Geolocation API](#)
 - a. The Geolocation API provides the user's location for web application use.
 - b. We will use this API in combination with the Location API to find the user's nearby grocery stores.

G. Front-end Framework

- 1. We will use Bootstrap as our front-end framework because of its aesthetics and simplicity in achieving desired appearance. It is intuitive to use.
- 2. We plan to utilize Bootstrap's responsive grids to easily format articles, recipes, and information. We are also interested in customizing Bootstrap's default colors and fonts to suit our design.

H. Task Breakdown

- 1. Amanda Tan: Project Lead
 - a. Manage overall system/file structure: Work with backend and frontend to make sure everything works smoothly
 - b. Pick up any work that isn't explicitly delegated/falling behind schedule
 - c. Assist with roadblocks in any area
- 2. Naomi Lai:
 - a. Make sure routing flows smoothly and web page reflects accurate API information
 - b. Organize individual favorites page
- 3. Jacob Lukose:
 - a. Create databases based on Food API and News API.
 - b. Login page and database.
- 4. Kishi Wijaya: HTML & CSS
 - a. Create HTML templates
 - b. Create CSS templates
 - c. Work with the Geolocation API and the Location API