Hydronium: Dua Baig, Aidan Wong, Qianjun Zhou
SoftDev
P01: ArRESTed Development
2024-11-25
Time Spent: 2
TARGET SHIP DATE: 2024-12-16

DESIGN DOCUMENT (VERSION 0)

I. Description

This project is a multifunctional cooking website that contains recipes for every occasion. Users can search a collection of recipes drawn from an API and other users and contribute and save recipes from this collection. They can also edit recipes to add details, clarify a vague concept, or revise an incorrect recipe step. Users also have access to a holiday calendar, which is used to explore recipes for various holidays, and a random recipe generator for days they are unsure what to cook. This website can be viewed without an account but requires one to contribute and save recipes from the recipe collection.

A. Program Components

a. User Accounts:
    i. Creation of accounts and login/logout functionality
    ii. Sessions
b. Routes to different pages of the website using Flask and Python
c. APIs:
    i. Spoonacular: Provides food-related information (recipes, nutrition, etc)
    ii. Calendarific: Generate holidays and related information for a calendar
    iii. SearchAPi: Search for recipes for a given holiday
    iv. Giphy API: Generating related gifs for each recipe
d. SQLite3 Database: Stores data of the user and recipes
e. Jinja Templates:
    i. User dashboard:
        1. Logged In State: Contains recipes that the user has saved, a small image of the user's profile image, and a search bar that allows the user to search for recipes
        2. Logged Out State: Contains a search bar that allows the user to search for recipes, a login button, and a logout button
    ii. Recipe Collection: Contains user-contributed recipes and recently searched recipes

iii. Recipe Viewer: Displays recipe with corresponding image/gif and the user who created it (if it is a user-contributed recipe)

iv. Search Result Page: Displays the results of a user search for recipes

v. Holiday Calendar Page: Displays all the holidays from an API and allows the user to click the holiday to be brought to a page with related recipes

1. Holiday Recipe Results Page: Displays the related recipes to a holiday

B. Program APIs

a. Spoonacular: Provides food-related information (recipes, nutrition, etc)

b. Calendarific: Generate holidays and related information for a calendar

c. SearchAPi: Search for recipes for a given holiday

d. GiphyAPI: Generating related gifs for each recipe

C. Frontend Framework: Foundation

1. Why Foundation?
   a. Easy-to-use and easy-to-follow tutorials
   b. Vast amount of CSS components and JS ones that will make things look and feel good
2. How Foundation?
   a. Grid System: Structure layout of pages like dashboard, recipe pages, and search results (Website will be able to adapt to size changes)
   b. Pre-designed Components: Buttons, Forms, etc
   c. JS Plugins: Modal Windows for confirmation messages

D. Program Component Connections

a. User accounts: Give access to contributing and favoriting recipes (Created through registration, and can be logged in to or out of).

b. Routes + Python: Routes allow users to traverse the website. They connect the different pages (HTML documents) of the website. Python also interacts with APIs and the database.

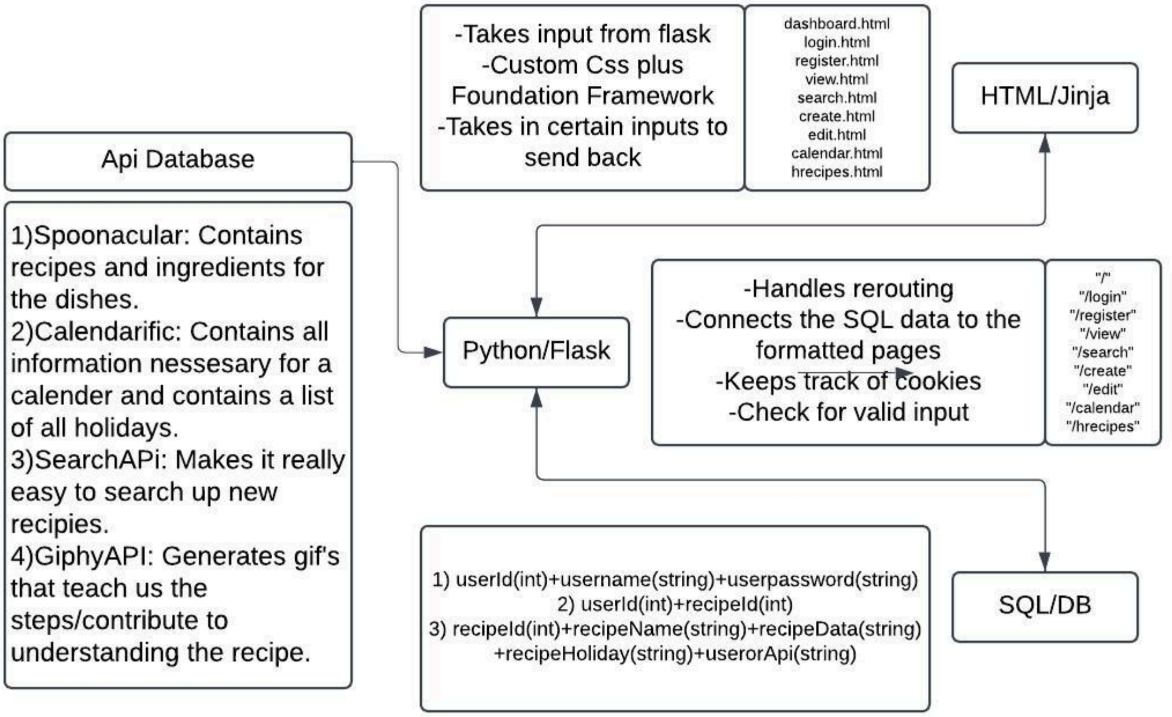
c. Database: Stores information related to the user (ID, permissions, etc) and recipe information (favorited recipes, recipe names, etc)

i. One of the main factors for information exchange between components (has all the data)

d. APIs: Provide information that is stored in the database and displayed on the HTML pages

e. Templates: Allow for dynamic web pages (as they need to update when new stories are created/edited)

E. Component Map



-Takes input from flask
-Custom Css plus Foundation Framework
-Takes in certain inputs to send back

1)Spoonacular: Contains recipes and ingredients for the dishes.
2)Calendarific: Contains all information nessesary for a calender and contains a list of all holidays.
3)SearchAPi: Makes it really easy to search up new recipies.
4)GiphyAPI: Generates gif's that teach us the steps/contribute to understanding the recipe.

-Handles rerouting
-Connects the SQL data to the formatted pages
-Keeps track of cookies
-Check for valid input

1) userId(int)+username(string)+userpassword(string)
2) userId(int)+recipeId(int)
3) recipeId(int)+recipeName(string)+recipeData(string)+recipeHoliday(string)+userorApi(string)

F. Database Organization

Table to store userData

userId (integer)	userName (string)	userPassword (string)

Table to store favRecipe

userId (integer)	recipeId (integer)

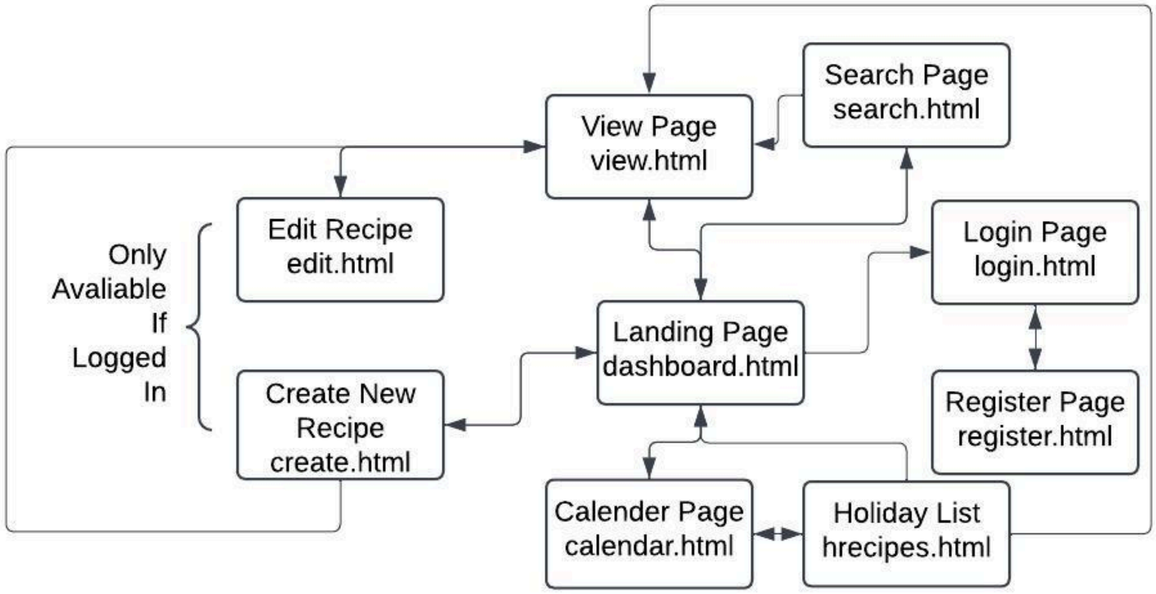
Table to store recipeData

recipeId (integer)	recipeName (string)	recipeData (string)	recipeHoliday (string)	userOrApi (string)

a. User Table
    i. User ID (PK)
    ii. Username (PK)
    iii. Password
b. Favorited Recipes Table (Tracks recipes that registered users have favorited)
    i. User ID (FK)
    ii. Recipe ID (FK)
c. Recipe Collection Table (Tracks all user-generated and API-called recipes)
    i. Recipe ID (PK)
    ii. Recipe Name
    iii. Recipe Data
    iv. Holiday of Recipe
    v. UserorAPi (if the recipe was contributed by a user or not)

Note: FK stands for foreign key (to link tables), PK for primary key (each row value must be unique)

G. Site Map + Descriptions



1. Landing page/Dashboard (/): Dashboard
2. Login (/login): Allows the user to sign in to an account
3. Registration Page (/register): Allows the user to register for an account and logs the user in upon successful account creation

4. Recipe Viewer (/view): Displays recipe information with additional information like nutrition facts, holiday if applicable, etc
5. Search result page (/search): Shows the result of a recipe search by a user
6. Create Recipe Page (/create): Page where user can create recipes (Name and contents)
7. Recipe Editor Page (/edit): Page where user can edit recipes (Making corrections, adding on, etc)
8. Calendar Page (/calendar): Displays a calendar with holidays filled in on their respective dates
9. Holiday Recipes Page (/hrecipes): Displays recipes related to a specific holiday (after clicking on the calendar page)

H. Task Breakdown

1. Dua Baig: Frontend
    a. Create HTML pages with Jinja templating - Includes any forms required for logging in/signing up
    b. Design site style with CSS and front-end framework (Foundation)
2. Aidan Wong: Backend (Database and API functions)
    a. Create SQLite3 database schema
    b. Work on database interaction modules (python) → General operations (ex. Inserting data, creating tables, etc)
    c. Work on API interaction modules (python) → General operations (ex. Retrieving data based on function input, etc)
3. Qianjun Zhou: Backend (Python)
    a. Routing and logic between pages
    b. User session management
    c. Linking database and API functions with site features