

**SW Engineering CSC648/848**

**“FoodsOnly”**

**Section 01 Team 06**

**Issac Moreno (Team Lead)**

**Noah James Yamsuan (Scrum Master)**

**Anshav Upadhyay Nepal (Front End Lead)**

**Terrell Enoru (Back End Lead)**

**Kayla Young (GitHub Master)**

**Karl Xavier Layco**

**Milestone 2**

**3/18/2024**

## **1. Part I:**

### **Data Definitions V2:**

Recipe Entity- An entity that stores information on a recipe

- Title-name of a recipe (string)
- Ingredients-ingredients used in recipe(list of strings)
- Macros- macronutrients of the completed recipe(strings)
  - Calories (integer)
  - Protein (integer)
  - Fat (integer)
  - Carbohydrates (integer)
  - and others.
- Description- Description on what the finished dish is (string)
- Instructions- Instructions on how to make the recipe (list of strings)
- Rating- User submitted rating of the recipe (integer)

User Entity- A user is an entity that stores information about unregistered users

- Name- name of user (string)
- Dietary restrictions- restrictions on what users can eat based on diet (list of strings)
- Search history- a history of searches by the user (list of strings)

Account Entity- An account is an entity that stores information about registered user accounts

- Username-username of user. Used for login (string)
- Password- Password of user. Used for login, and will be hashed. (string)
- Email- email of user. Used for login (string)
- Userid- Userid used to identify an account. (integer)
- Profile- A collection of the public information associated with an account (list of strings)

Profile Entity- A profile is an entity that stores information about accounts public information

- Social media links- links to the social media that an account has (dictionary)
  - Facebook (string)
  - Instagram (string)
  - Twitter (string)
- Recipes- a collection of recipes made or saved by an account (list of strings)
- Allergies- Allergies of a user put in their account (list of strings)

- Restaurant affiliations- Marker of whether or not the account is associated with a restaurant (list of strings)
- Username-Username of account holder

Post Entity- A post is an entity that stores information about a recipe post

- Recipe- The recipe associated with the post. (list of strings)
- Recipe image- Image of completed recipe (image)
  - Maximum size - 10 mb for images, 50 mb for videos
- Tags- Tags are searchable terms used to identify aspects of a post (list of strings)
- Comments- Comments are user submitted comments on a post (list of strings)

Restaurant Entity- A restaurant is an entity that stores information about a recipe

- Restaurant name- Name of a restaurant (string)
- Location-Location of a restaurant (string)
- Food type- Type of food made in the restaurant (string)
- Price range- The average price of food in the restaurant i.e. from high end to fast food \$ to \$\$\$ (string)
- Contact info- Information on how to contact a restaurant i.e Phone number (list of strings)
  - Phone (string)
  - Email (string)
  - Name (string)
- Reviews- User submitted reviews of the restaurant. (list of strings)

## **1.1 User Privileges:**

### **Unregistered Users:**

- Can view recipes and restaurant information.
- Can search for recipes based on criteria.
- Cannot save recipes or create posts.

### **Registered Users:**

- Have all privileges of unregistered users.
- Can create and save recipes.
- Can create posts and upload images/videos.

- Can review recipes and restaurants.
- Can follow other users and restaurants.

## **2. Functional Requirements V2:**

### **1. Users accounts: Priority 1**

- Users shall be able to make accounts by entering their email and password.
- Accounts shall have an existing email, password, full name, birthdate, and username.
- Guests or Non-Users shall be required to make an account to make a post, make a comment on the posts, find profiles of other users, and access the restaurant finder.

### **2. User Login: Priority 1**

- Users shall be able to login using their email or username, and password.
- Users shall be able to reset their password by clicking on “Forgot Password.”
- An account shall only be created with one unique email.

### **3. Home Page: Priority 1**

- The body of the home page shall display 15 posts of other registered users per page.
- The home page shall have a section displaying the trending recipe of the week or month.

### **3. Posting: Priority 2**

- A post shall consist on
- Each recipe is required to include details such as ingredients, step-by-step instructions, nutritional information.
- Users will be able to add any special notes to their recipes if they wish to.
- Users shall be able to include images such as jpgs, pngs, or gifs to their posts.
- Posts shall have tags to help with searching and filtering.

### **4. Search: Priority 1**

- Users shall be able to assign categories/tags when making their posts. If the tag does not exist, users shall be able to make a new one.
- Users shall be able to interact with our AI, using ChatGPT API, to ask for good food suggestions such as “What kind of Asian food do you suggest?” or “Can you show me a cheap recipe that is also low in calories?”

### **5. Filtering: Priority 2**

- Users shall be able to filter searches using categories/tags, nutritional details, or recipe preference.

## 6. Profiles **Priority 1**

- Profiles shall be able to show food or restaurant recommendations of the user.
- Users will be able to bookmark recipes that they like or want to remember.
- The profile page shall be able to display bookmarked posts. Only the user of the profile can view their bookmarks.
- Users shall have an option to display food preference, allergies or diet onto their profile page.
- Users will be able to create, customize, and follow meal plans using recipes and posts.
- For meal plans the user will be able to set a weekly schedule, be able to add posts and recipes for others to follow in their meal plans, and add notes to their plan.
- Users shall have an option to add recipe preference to their profile.

## 7. Nutrition **Priority 3**

- Users shall be able to include their diets into their profiles.
- Posts shall display total calories of the food/recipe using a FoodAPI to pull values and print them based on what kind of food the user put in their recipes.

## 9. Alternatives **Priority 3**

- Users shall have an option to search for posts with only healthy tags.
- The program shall have an AI where it can suggest recipe/food alternatives to the user.
- When the user asks “Do you have any other cheaper alternatives?” then the AI will search through our database and offer a cheaper solution.

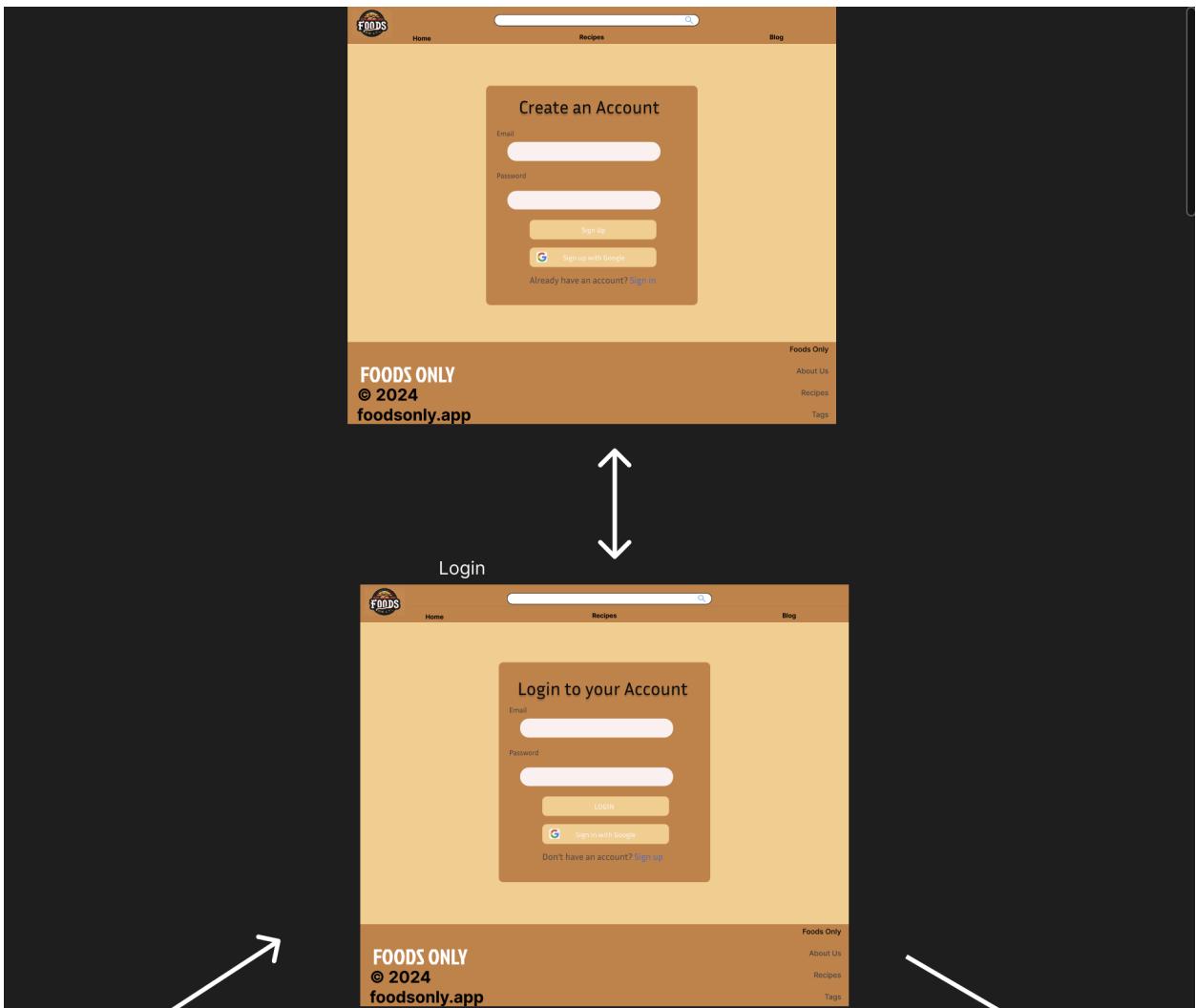
## 10. Cookies: **Priority 3**

- The home page shall display posts that are relevant to the user's search history.

## **3. UI Mockups and UX FLow:**

**Link to full flow:** [Link to Figma Flow](#)

## Login Page/Create Account page



**Useful:** Logging into an account is extremely useful for a user if they want to post a recipe, save a recipe, or create a personalized meal plan.

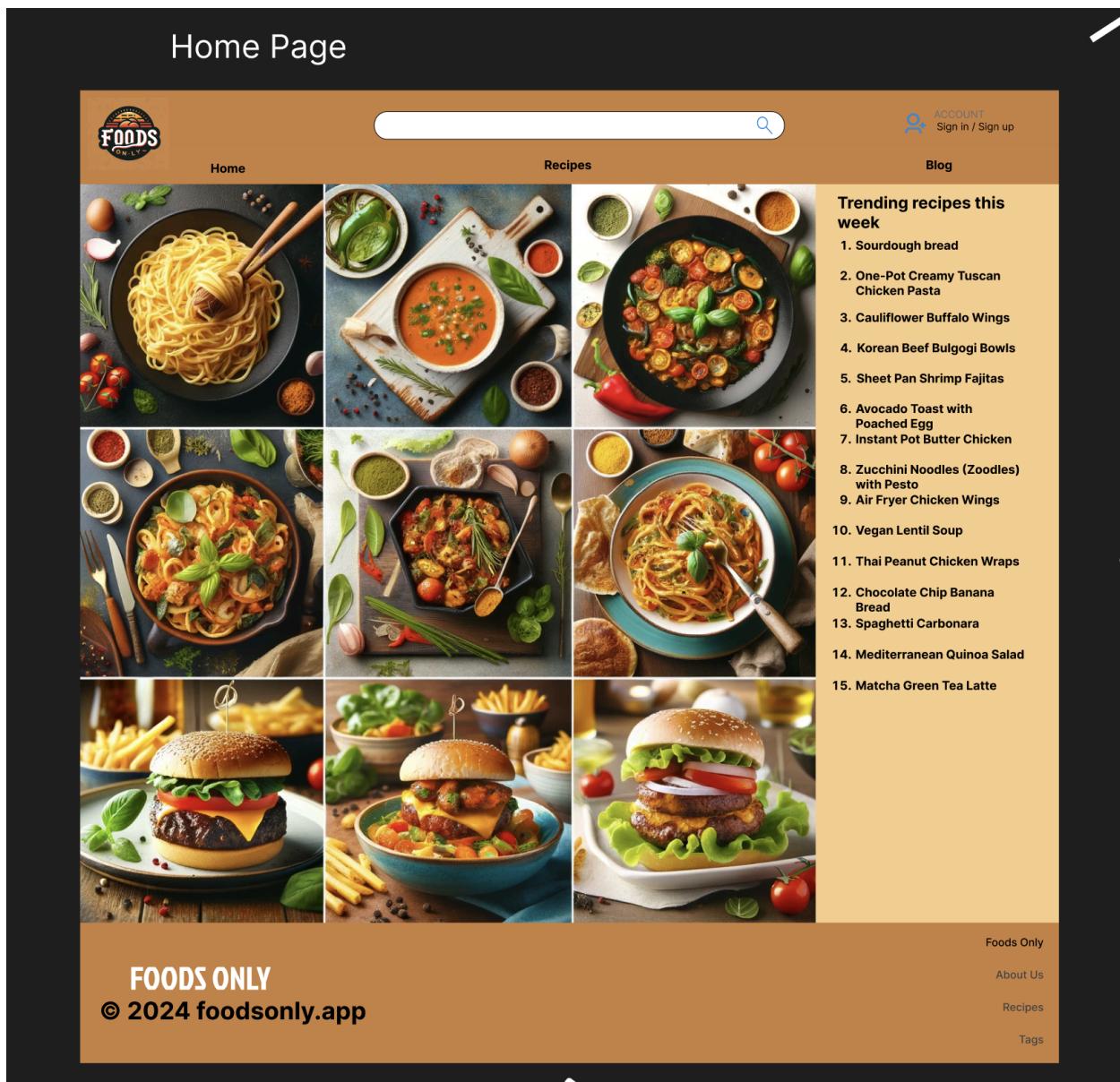
**Desirable:** It is desirable to want to create an account/login to access the full features of our product. (User privileges restrict some features like bookmarking).

**Accessible:** The login/account creation page is accessible because it is found on the navigation bar throughout the website.

**Feasible:** Logging/creating an account is feasible but not absolutely necessary for our website since some features allow unregistered users to browse.

**Credible:** Accounts are credible because they are stored safely and ensure that they are unique to their own.

## Home Page:



**Useful:** The home page is very useful because it displays key interests for the user like, new recipes, searching, trending recipes, and account access.

**Desirable:** It is desirable to have a home page so that new and existing users have a welcoming page to view upon visiting our sight.

**Accessible:** The home page is accessible throughout the entire website by clicking on either the logo icon or the “Home” option.

**Feasible:** A home page is feasible because it truly acts as the “face” of the website.

**Credible:** A home page is credible as long as it displays current and true data to the user. For example, real trending recipes and new data from the database.

## Recipe Creation:

The screenshot shows the 'Recipe Creation' page of the Foods Only website. At the top, there is a navigation bar with a logo, a search bar, and user account options (Blog and Logout). Below the navigation, the main form area is divided into several sections:

- Recipe Details**: Fields for Title and Description.
- Ingredients**: A single field for Ingredients.
- Instruction**: A large text area with a blue border and an 'Add Step' button.
- Nutritional Information**: Fields for Calories, Fat, Protein, and Carbohydrates.
- Special Notes**: A large text area.
- Images**: An 'Add Image' button.
- Tags**: An 'Add Tags' button.

A prominent green 'Post Recipe' button is located at the bottom center of the form area. At the very bottom of the page, there is a footer with the 'FOODS ONLY' logo, copyright information (© 2024), and a website URL (foodsonly.app). To the right of the footer, there is a sidebar with links to 'Foods Only', 'About Us', 'Recipes', and 'Tags'.

CONTD:

The screenshot shows a mobile application interface for 'FOODS ONLY'. At the top, there is a navigation bar with a logo on the left, a search bar in the center, and a user profile icon on the right. Below the navigation bar, a post is displayed with a user's profile picture, their handle '@username', and a placeholder text 'Short User Description'. The main content area features a title 'Title of Post' above a grid of nine food images. The grid contains various dishes such as spaghetti, soups, and burgers. To the right of the grid is a vertical 'Side Bar' with a grey background. At the bottom of the screen, there is a footer bar with the 'FOODS ONLY' logo, the copyright notice '© 2024 foodsonly.app', and links for 'Foods Only', 'About Us', 'Recipes', and 'Tags'.

**Useful:** It is useful to have a recipe creation page because it allows us to have new and (hopefully) unique content that is created by our users.

**Desirable:** Creating recipes is desirable because it ensures that we have a user-centered application by allowing users to upload their own content.

**Accessible:** Creating recipes is accessible upon navigating to the recipes section. The current format of the form is less accessible but will be updated to be more user-friendly.

**Feasible:** Recipe creation is a feasible feature because it is a key component of a food blog.

**Credible:** Due to the nature of creating content, our recipes may not always be credible when added by a user. Therefore, this function decreases our credibility.

## Profile Viewing:

User's view of other profiles

Logout

@username

A description of the user

Food Preferences

FOODS ONLY  
© 2024 foodsonly.app

User's view of their own profile

Edit profile

Logout

@username

A description of the user

Bookmarks Food Preferences Meal Plans

FOODS ONLY  
© 2024 foodsonly.app

CONTD.

Bookmark

Logout

Bookmarks

FOODS ONLY  
© 2024 foodsonly.app

Edit Profile

Logout

Change photo

Username

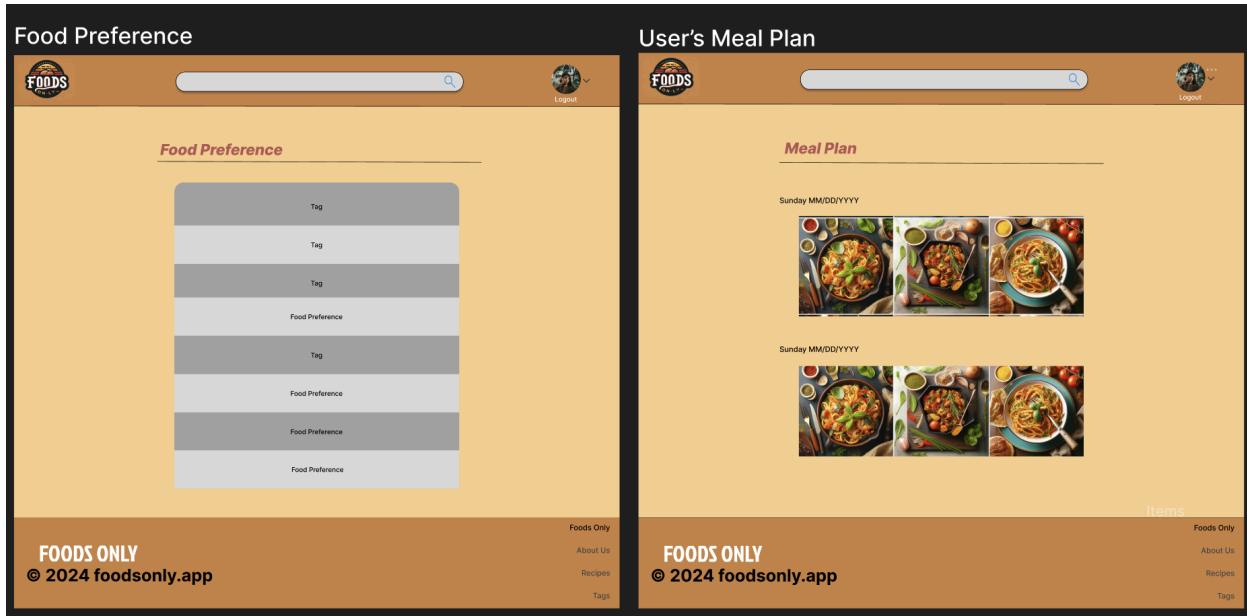
Gender

Bio

Submit Change

FOODS ONLY  
© 2024 foodsonly.app

CONTD.



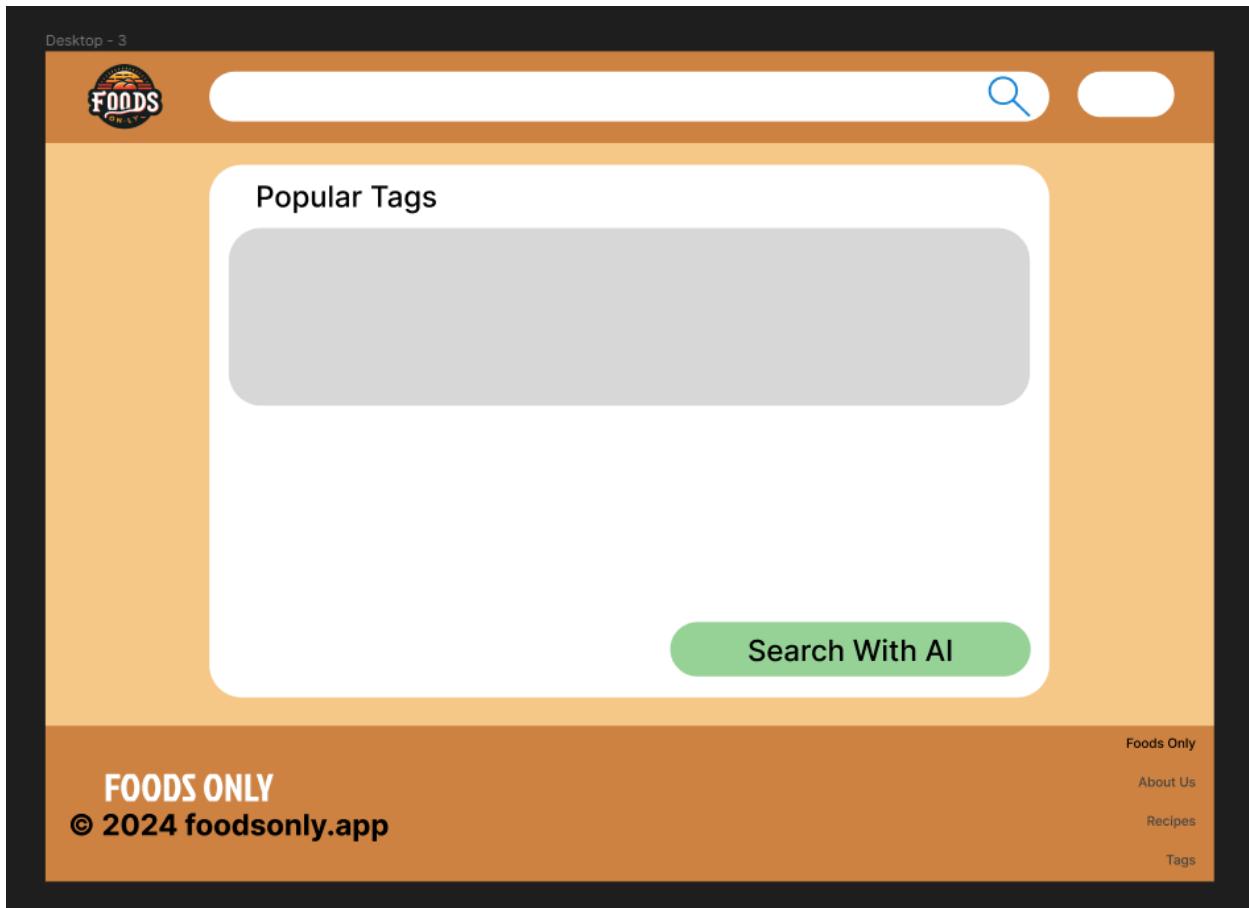
**Useful:** The usefulness of adding profile viewing/editing creates many different factors. From this feature, we have built-in bookmarks, editing, and meal plans.

**Desirable:** Having a viewing profile UI makes our website more user-centered because users can interact with each other, and potentially share recipes by bookmarking each other's recipes.

**Accessible:** These features are accessible, but a little hidden from the main flow of the website. Organizing these features in a tucked-away fashion adds importance to browsing recipes.

**Feasible:** It is feasible to include these features because a user can act as an object that possesses different structures like a meal plan or bookmarks.

**Credible:** It is credible since we require users to login to their accounts before they can view other profiles, and we will require users to use their passwords for authorization.



**Useful:** The usefulness of doing a search function is for users to be able to search specific items they want to have and filter out items they do not want.

**Desirable:** The search function will make the program desirable to use since items can be easily search accessed.

**Accessible:** The usefulness of searching for tags using artificial intelligence is self-evident because it increases the speed at which users can find what they want.

**Feasible:** It is feasible to include these features because the recipes can fetch from the database and show to the front end.

**Credible:** This feature is credible because as seen from chat gpt ai assisted searching is very useful and convenient

#### 4. High Level Architecture, Database Organization:

Recipe Schema {

```
Title:String  
Macro:String  
Description:String  
Instruction: String  
Rating:Integer  
}
```

```
User Schema{  
Name: String  
Dietary Restrictions: String  
Rating:Integer  
}
```

```
Account Schema{  
Username:String  
Password:String  
Email:String  
Userid:integer  
Profile: array  
}
```

```
Profile Schema{  
Social Media: Array  
Recipes: Array  
Allergies:Array  
Restaurant affiliation: String  
}
```

```
Post Schema{  
Recipe: String  
Recipe Image: image  
Tags:array  
Comments:array  
}
```

```
Restaurant Schema{
  Restaurant Name:String
  Location:String
  Food Type:String
  Price Range:String
  Contact Info:Array
  Reviews:Array
```

- a. Add/Delete/Search: Specify which operations are permitted per collection.  
(ie. which entries will be add/searched/deleted/displayed/etc)  
In the recipe table all operations will be allowed on all the entries. In the User collection all operations will be allowed on all entries. On the Account collection it will be possible to add a username, password, or email if none exist or delete any of them if they do exist however another one must be created. UserId shall be unmodifiable. Profile shall be displayed added and searched. On the Post collection all operations shall be allowed on the recipe, recipe image, and comments whereas on the tag entry only add or delete, and search operations are allowed. On the restaurant collection all operations shall be allowed on every entry.

On the profile collection all operations shall be allowed on all entries.

- b. Define and describe API used to communicate frontend to backend.  
INCLUDING our 3rd party APIs.  
The three apis that we will be using are spoonacular and google places.  
Spoonacular is an api top get the recipes from and google places is an api we will use to get location based restaurant recommendations. We will also be creating a simple crud api for interaction with the user, account, post, and profile schema.

## 6. Key Risks:

1. Skill risks and mitigation plan:

For our team, there are medium skills risks for our chosen frameworks. This is considered a medium risk because we are fairly new to the products. However, our mitigation plan is to actively learn, study, and share our findings with the rest of the team.

2. Schedule risks:

For our team, scheduling poses a low risk for progress. I claim that it is a low risk because, thus far, we have been able to meet 2-3 times per week and actively engage in each milestone.

3. Teamwork risks:

Our teamwork and effort poses medium risks for our progress. Teamwork is a medium risk because most members of our teams are also enrolled in other workload intensive courses. Therefore, some of us may fall behind due to workload. However, our plan is to pick up our teammates' slack and ensure we are always on the same page.

4. Legal risks:

Legal risks for our project are low. The chosen products and frameworks for our project are open source and allow us to implement its usage within licensing agreement. As long as we do not steal other people's ideas or use our project for profit, we are within legal measures.

## 7. Project Management:

When it comes to project management, Issac (team lead) leads the progress and planning of each milestone. After each milestone is announced, we meet during the week to break up the milestone into parts and work on each part consecutively. Furthermore, Issac also ensure that the workload is split between members and assign tasks that may be specific to a member's role (eg. backend). Scrum master is sharing progress with each member per meeting and ensures we are able to meet each week. Scrum master also keeps track of meetings times and availability. Our tool to track progress is Notion. We recently setup our Notion workspace and are still getting use to its features. However, as team lead, Issac creates each sprint and assign tasks (as mentioned before) with included details like priority, status, assignee, etc. Notion allows us to visualize our progress per milestone and ensure we are always on the same page. Furthermore, we also utilize Notion for file sharing and keeping our crucial website links in a tracker.