

Group 4 - Milestone 1 - Project Planning Report

PlanYourPlate - Meal Planning System

GitHub Repository: <https://github.com/CMPT-276-FALL-2024/project-10-streams>

Ines Machinho Rodrigues (301426807)

Owen Smith (301569950)

Sara Magdalinski (301465986)

Wonchan Kim (301449586)

November 1st, 2024

Table of Contents

PlanYourPlate API's.....	1
API 1: Edamam Recipe Search API.....	1
Feature 1: Searching Recipes by Allergies.....	1
Feature 2: Searching Recipes by Nutrient Amounts.....	1
Feature 3: Searching Recipes by Calorie Limit.....	2
API 2: The MealDB API.....	2
Feature 1: Searching by Area.....	2
Feature 2: Search by Main Ingredient.....	3
Feature 3: Show a Random Meal.....	3
Mid-Fidelity Prototype.....	4
Software Development Life Cycle (SDLC).....	6
Work Breakdown Structure.....	7
Milestones.....	10
Risk Assessment:.....	11
Low Risks.....	11
Medium Risks.....	12
High Risk.....	12
Data Flow Diagrams.....	13
MVC.....	14
Model.....	14
View.....	14
Controller.....	14

PlanYourPlate API's

PlanYourPlate will make use of two APIs, Edamam Recipe Search and MealDB. These APIs provide a variety of ways to search for meals based on user preferences and dietary specifications as well as more generalized options for those who are indecisive. These APIs will help to ensure that PlanYourPlate can find meals for any user's needs.

API 1: Edamam Recipe Search API

Edamam Recipe Search API can be used to find recipes based on a variety of constraints from its database of over 2.3 million recipes. It contains recipes from over 500 sources and organizes the recipes in order of quality. It contains over 80 filters to allow for personally tailored recipes. It is equipped with a Natural Language Processing (NLP) engine to automate nutrition analysis. PlanYourPlate will use this API to provide recipe suggestions based on allergies, calorie limits and nutrient amounts. The vast database provided by Edamam Recipe Search API will be useful for finding a large number of recipes for any user specifications.

Feature 1: Searching Recipes by Allergies

Edamam Recipe Search API can provide recipes that avoid allergens. For example, it can provide recipes that are gluten-free, soy-free, peanut-free and almost any other common allergen. Users can select one or multiple allergens to exclude, ensuring that suggested recipes are safe and suitable for their needs. It can also incorporate requested ingredients into allergen-free recipes. This feature works by having a user simply select allergens they wish to avoid and enter any items they wish to include (or even what meal they would like to have). The API will then provide a list of recipes that suit the user's needs. This feature is extremely useful for users who have allergies or dietary restrictions or users who are preparing meals for individuals with dietary restrictions. This feature will allow users to find recipes that are well-suited for them without having to look through the ingredient list of each recipe before deciding on a meal. This will make meal planning quicker and more convenient.

Feature 2: Searching Recipes by Nutrient Amounts

Edamam Recipe Search API can provide recipes with desired amounts of nutrients such as fat, sugars, protein, fibre, sodium and more. Users can enter the minimum and maximum amount (in grams) of any of these nutrients that they would like to include in their meal, along with any ingredients they would like included, and the API will provide a list of recipes meeting this

criterion. This feature will be useful for individuals who are trying to meet certain nutrient goals. For example, a bodybuilder may wish to find recipes that are very high in protein and can search for meals that have between 50g and 150g of protein. This feature will allow users to browse a large selection of meals that align with their nutrition goals without having to manually calculate nutritional values for each recipe they visit. This will allow users to explore a wide variety of meals effortlessly, saving them time and energy while helping them achieve their goals.

Feature 3: Searching Recipes by Calorie Limit

Edamam Recipe Search API can provide recipes that fit within a certain amount of calories. Given an inputted list of ingredients and a maximum amount of calories, it will provide meal suggestions that contain the given ingredients and are less than the specified calories. For example, inputting “chicken, bread, cucumber” and “250kcal” provides recipes such as a “Sesame Ginger Chicken Sandwich” which contains 211 kcal per serving. This feature will be extremely useful for users who are trying to consume a certain amount of calories per day. By using this feature, a user will be able to choose from a wide range of meals without having to put effort into calculating the calories in their meal. This will allow users to stick to their goals by providing a convenient method to find new recipes that meet their needs.

API 2: The MealDB API

The MealDB API is a robust tool for searching for meal recipes from an extensive database, allowing users to filter and find recipes based on a variety of preferences. The API allows users to find a meal suggestion based on many different criteria, such as a preferred cuisine or main ingredient. This API can also suggest a random meal idea, making it useful for users who have no specific preferences for their meals.

Feature 1: Searching by Area

The MealDB API provides a wide range of recipes from various cuisines, enabling users to explore international dishes by filtering meals based on their country. This feature works by simply entering a cuisine such as “Canadian” or “Indian” and the API will return a list of recipes from that cuisine. This feature is particularly valuable for individuals from diverse cultural backgrounds as it allows them to quickly find recipes with flavours reminiscent of home. It is

also useful for individuals trying to find more variety in their meals, allowing them to find recipes from new cuisines. By providing easy access to recipes from specific cuisines, this feature streamlines meal planning and makes it easier for users to experiment with global flavours.

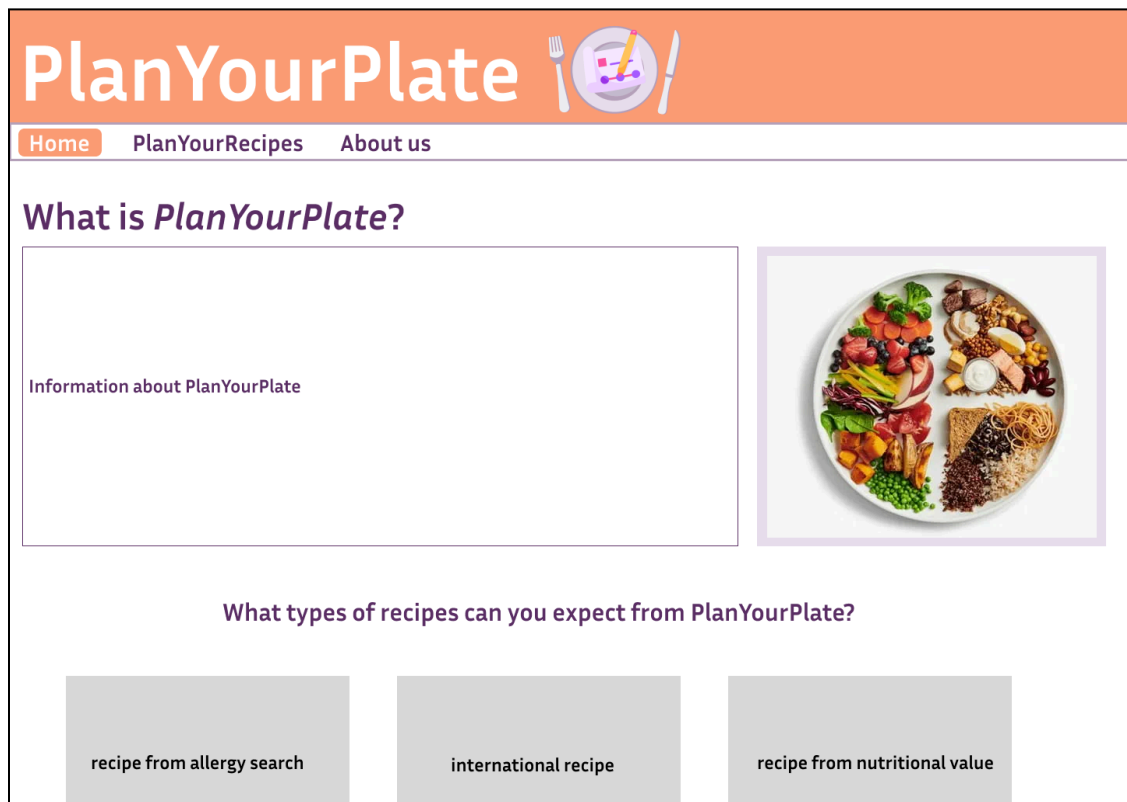
Feature 2: Search by Main Ingredient

The MealDB API allows users to search for recipes by specifying a main ingredient, making it easy to find dishes based on what is available in the kitchen. A user can enter one ingredient they want their dish to revolve around, and this API will return a list of meals based on that ingredient. This feature is particularly useful for reducing food waste and accommodating personal preferences or dietary needs. By focusing on a specific ingredient, users can discover new recipes that highlight or make the most of what they already have. For example, if someone has an ingredient that is about to expire, they could use this feature to generate meal ideas based on that ingredient. This feature could also be used by someone who only enjoys certain foods to find recipes that work off of their preferences.

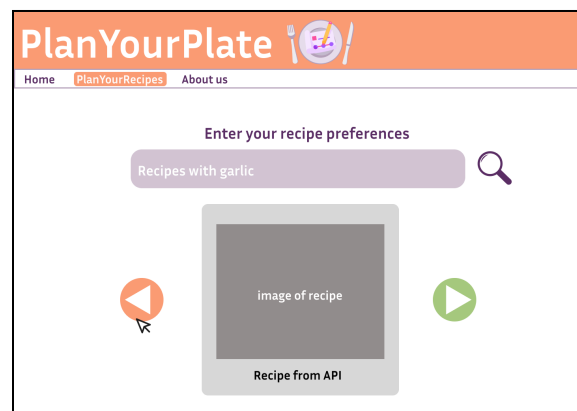
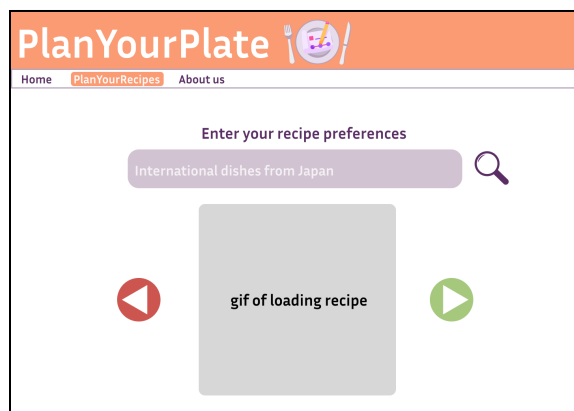
Feature 3: Show a Random Meal

The MealDB API offers a feature that displays a random meal suggestion to users. With the click of a button, a random meal idea will be generated. This can be particularly helpful for individuals who are short on time and need quick meal ideas without having to spend time planning. Additionally, it is beneficial for beginners or those with limited cooking experience, as it provides them with simple, randomized meal options to try without the pressure of making decisions about what to cook. This feature can greatly speed up the process of deciding what to make, removing some of the mental strain of decision-making

Mid-Fidelity Prototype

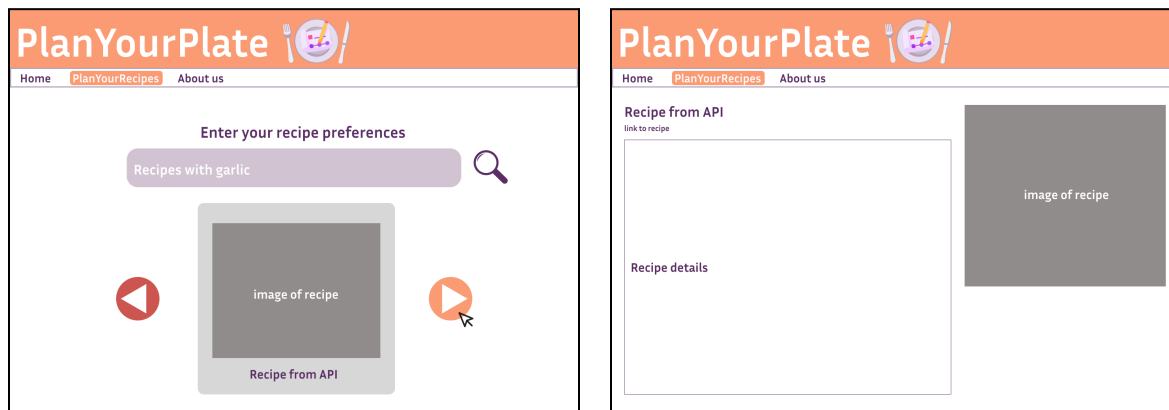


The Home page consists of a description of PlanYourPlate and examples of recipes showcasing how PlanYourPlate works. By using a variety of images, users are able to quickly see how it works and be eager to try it if the examples appeal to them. Our large header as well as our logo captures the user's attention, drawing them below to the navigation bar where the user can explore our pages such as PlanYourRecipes where the user can try out our feature.

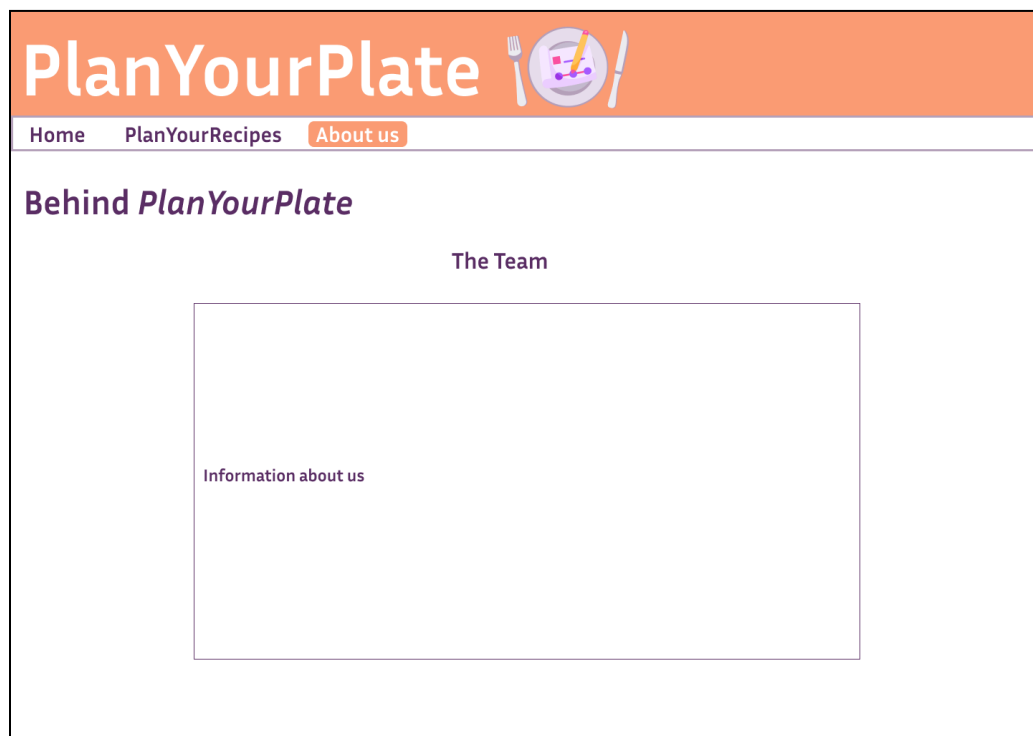


The PlanYourRecipes page consists of a large search bar with instructions above, as well as an example in the search bar to show users what they could type to get their desired recipes. Reinforcing the recognition over recall usability heuristic, a magnifying glass icon is used on the side of the search bar to let users know how to use it. The minimalist design also helps new users not become overwhelmed.

If a user is unhappy with the recipe generated, they can decline by pressing the red arrow button on the left. Hovering either arrow button will turn it orange, to provide visual feedback to the user, adhering to the visibility of system status usability heuristic. This will cause the user to return to the original screen to update their search.



If a user is happy with the recipe generated, they can click the green arrow button instead which will display the full recipe, including a link to the original recipe. At any point in this process, users are able to click any of the other pages in the navigation bar, in order to promote user control and freedom.



For the About us page, information about the team will be included, along with inspiration for the project.

Throughout the pages, the prototype maintains consistency with page layout, and colour schemes in order to give the user a pleasant experience. Since our target audience is primarily users who are in a hurry, it was important to prioritize ease of learning at first use to achieve the goal of not compromising on your meals because of lack of time.

Link to interactive prototype:

<https://www.figma.com/proto/n99rbL6B0c1HEaG7xbydhz/PlanYourPlate-Mid-Fi?node-id=0-1&t=IOtJCi68HqhTFEqW-1>

Software Development Life Cycle (SDLC)

For this project, our group has decided to follow the Agile SDLC model, specifically the Scrum framework. One of the main reasons we have chosen this method is the emphasis on responding to change over following a plan. Since this is the first major project anyone in our group has undertaken, we expect to have some conflicts with our original design during the development and testing phase. The flexibility provided by this model will allow us to revise our design quickly and efficiently when needed. Another factor in our decision to follow this model is the consistent collaboration involved. By having daily standups and retrospective meetings, our group will have constant communication assuring that everything is running smoothly.

Furthermore, since we have a relatively short time frame for the project, splitting the work up into sprints will make it more manageable. By having sprints, we will be able to set many smaller goals allowing us to focus on one task at a time. Finally, using this methodology will allow us to test our product frequently, ensuring that all issues are caught as soon as possible giving us as much time as possible to fix them. Overall, we believe that Agile: Scrum will allow us to produce a polished and functional final product.

Work Breakdown Structure

Task#	TASK	Assigned to	Estimated Hours
1	Requirement Gathering	Team	Done In M1-M1.5
2	CI/CD Pipeline Setup		
2.1	Set up Github Actions	Sara	
2.1.1	Linting & Unit tests include	Sara	2
2.1.2	Complete Github Action Setting		2
2.2	Setup Netlify	Sara	
2.2.1	Configure automatic deployment	Sara	2
2.2.2	Set up Staging Environment for testing	Sara	2
2.2.3	Complete Netlify Setup		4
2.3	Complete CI/CD Pipeline Setup		6
3	API Integration	Team	
3.1	API 1 (Edamam Recipe) Integration	Team	
3.1.1	Allergy-Based Recipe Search Integration		1
3.1.2	Calorie Limit Search	Team	1
3.1.3	Nutrient-Specific Recipe Search	Team	1
3.1.4	Complete API 1 Integration		3
3.2	API 2(MealDB API) Integration		
3.2.1	International Dish Search	Team	1
3.2.2	Main Ingredient Filter	Team	1
3.2.3	Random Meal Suggestion	Team	1
3.2.4	Complete API 2 Integration	Team	3
3.3	Complete API Integration		6

4	UI/UX Design	Ines & Wonchan	
4.1	Mid-fidelity Prototype	Ines	Done in M1.5
4.2	UI Implementation with Tailwind CSS		
4.2.1	Convert prototype design into functional UI	Ines&Wonchan	5
4.2.2	Integrate Tailwind CSS for styling	Ines&Wonchan	3
4.2.3	Add responsiveness and ensure accessibility	Ines&Wonchan	3
4.2.4	Develop components of prototype for API1	Ines&Wonchan	4
4.2.5	Develop components of prototype for API2	Ines&Wonchan	4
4.2.6	Complete UI Implementation		19
4.3	Testing Before Deployment		
4.3.1	Main UI testing	Owen	3
4.3.2	API1 Testing	Owen	3
4.3.3	API2 Testing	Wonchan	3
4.3.4	Team Feedback & Accommodation	Team	5
4.3.5	Complete Testing		14
4.4	Complete UI/UX Design		33
5	Main Features Development		
5.1	Implement Local Storage using React	Team	3
5.2	Develop functionality to save User Input	Team	3
5.3	Develop functionality to save Suggested Recipe	Team	3
5.4	Develop functionality to access viewed Recipe	Team	3
5.5	Feature-1 Searching Recipes by Allergies		
5.5.1	Integrate API 1 with user input	Team	3
5.5.2	Testing	Owen	1
5.5.3	Complete Feature 1		4

5.6	Feature-2 Searching Recipes by Nutrient		
5.6.1	Integrate API 1 with user input	Team	3
5.6.2	Testing	Owen	1
5.6.3	Complete Feature 2		4
5.7	Feature-3 Searching Recipes by Calorie		
5.7.1	Integrate API 1 with user Input	Team	3
5.7.2	Testing	Owen	1
5.7.3	Complete Feature 3		4
5.8	Feature-4 Searching by Area		
5.8.1	Integrate API 2 with user input	Team	3
5.8.2	Testing	Owen	1
5.8.3	Complete Feature 4		4
5.9	Feature-5 Searching Recipe by main Ingredient		
5.9.1	Integrate API2 with user input	Team	3
5.9.2	Testing	Wonchan	1
5.9.3	Complete Feature 5		4
5.10	Feature-6 Show a Random Meal		
5.10.1	Integrate API2	Team	3
5.10.2	Testing	Wonchan	1
5.10.3	Complete Feature 6		4
5.11	Team Feedback	Team	4
5.12	Complete Main Feature Development		40
6	Application Testing & Quality Assurance		
6.1	Unit testing		
6.1.1	Write unit tests on each feature	Owen & Wonchan	4

6.1.2	Run test	Owen & Wonchan	2
6.1.3	Team Feedback	Team	3
6.1.4	Unit testing complete		9
6.2	Integration Testing		
6.2.1	Test integration between Inputs & API	Owen & Wonchan	3
6.2.2	Test Feature implementation	Owen & Wonchan	4
6.2.3	Team Feedback	Team	2
6.2.4	Complete Integration Testing		9
6.3	Complete Application Testing		18
7	Documentation & Reporting		
7.1	Feature & API Documentation	Team	Done in M1.5
7.2	Write Final Report	Team	10
7.3	Make demo video	Team	5
7.4	Complete Documentation		15
8	Final Presentation	Team	

Milestones

Milestone	Tasks/Features	Deadline
Initial setup & Requirement Gathering	Requirement Gathering, Preliminary Planning (Done in M1-M1.5)	Completed
CI/CD pipeline Setup	Set up Github Actions & Netlify	November 4th
API Integration	API 1 Integration	November 7th
	API 2 Integration	November 10th

UI/UX Development	UI implementation with Tailwind	November 15th
	Testing & Feedback	November 18th
Main Feature Deployment	Implement Local Storage	November 19th
	Implement Feature 1-6	November 21th
	Team Feedback	November 22th
Application Testing & QA	Unit Testing	November 23th
	Integration Testing	November 24th
	Application Testing	November 25th
Documentation	Feature & API documentation	November 23th
	Final Report	November 25th
	Demo video	November 26th
	Complete Reporting	November 26th

Risk Assessment:

Low Risks

- The APIs that we use could potentially go down, considering that we are using them to access a database for recipes. They play a huge role in our design, since the way to avoid this would be to create our own, the best we can do is have clear error messages to satisfy the ten heuristics.
- Some browser compatibility issues may arise, the website may not render consistently across all browsers. To mitigate, we will have to test on all major browsers(chrome, safari, edge) during development, and use accommodating software to maximize compatibility.
- Since we are outsourcing APIs, we cannot guarantee that what is received from the API is fulfilling to the user. We also can't guarantee the user's input to be applicable enough for a good response. The APIs will have an error message for certain inputs, but we will put

in our own filters before the user's input goes into the API to help efficiency in the website.

- Collecting user feedback on our UI or site in general can slow improvements if we don't plan ahead. If we schedule frequent, minimal but informative feedback sessions early in the project to fix and iterate on the design, then updates will be more efficient
- We cannot control the internet speeds of the people using our site and this could affect our loading times and therefore the user's experience. While this is mostly out of our control, we can help mitigate this by using lightweight access such as compressed images and maybe implementing lazy loading(wait on loading for certain things such as images).

Medium Risks

- High volume of users could exceed free API limits and affect its functionality, a key part of our site. This is mostly out of our control, but we can make it very clear to the users what is happening and include fallback content for cases when the API fails. While adhering to the ten heuristics.
- Inaccurate meal suggestions mismatched to the user's preferences. There is no guarantee that the user is fulfilled by the initial feedback from the API, so to mitigate this, we add an option to adjust. Allow the user to make adjustments to their meal plans or recipes through a loop(if the user prompts again).
- Mobile device compatibility is no guarantee either and this is important incase user's don't have immediate access to a monitor, it cuts the user base down a lot. So to mitigate we will have to have responsive design principles tested on common devices. This will hopefully optimize for mobile users to ensure usability.

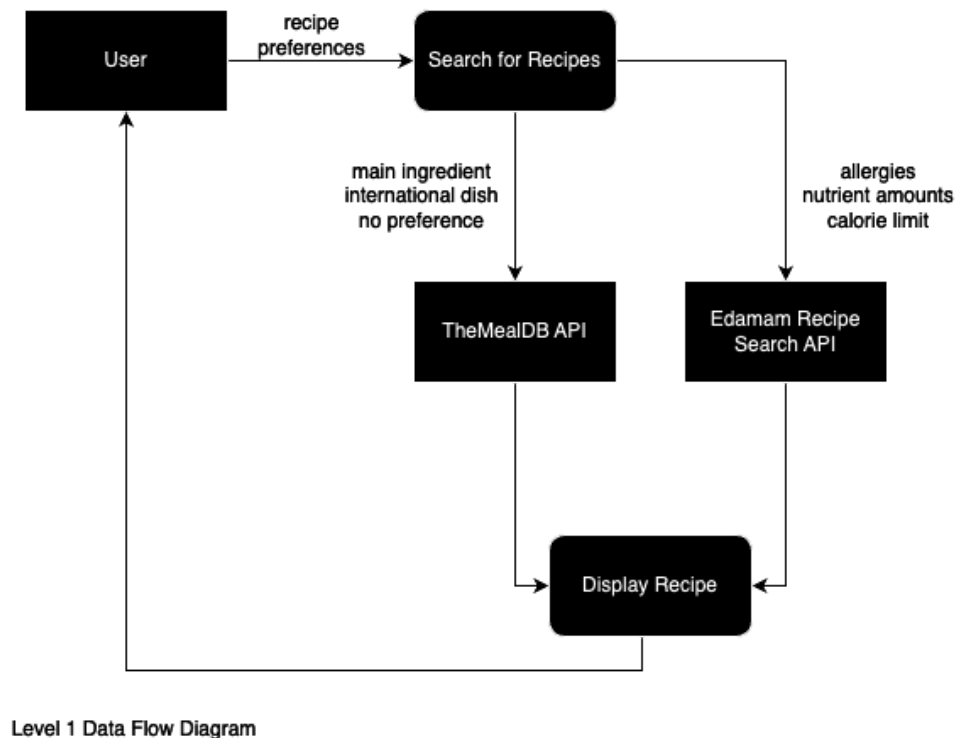
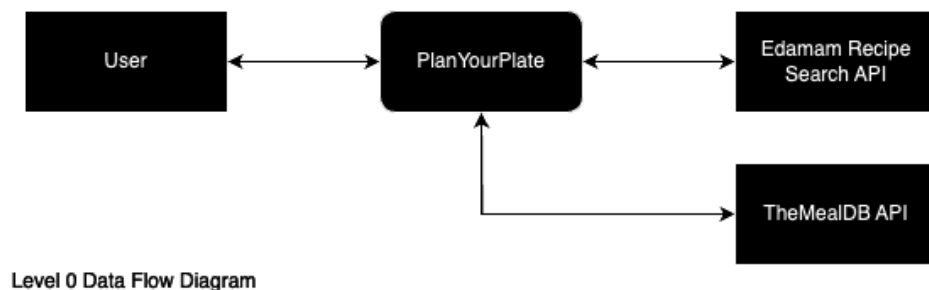
High Risk

- Data privacy is always important, especially on our site which will receive private information from the user. User's will have to input allergies and dietary restrictions to which they may find offensive if found that the data was not private. So we will strictly prohibit the outsourcing of the data, but we will also clearly communicate the privacy practices to ensure the user's confidence in our site. We can also avoid storing the data

long-term, although it can be used temporarily to aid the user in the search for a meal plan, the privacy impact will be consistent.

- API failure or discontinuation is plausible. In the search for applicable APIs, multiple sites are simply not operating anymore and there's no guarantee that the one we're using will stay up. Unfortunately there is not a lot we can do to accommodate this potential flaw except to have replacements on standby. Researching alternative APIs that could provide the site the same system functionality and structure that would allow the site to continue functioning.

Data Flow Diagrams



When using the PlanYourRecipes feature, a user will enter their preferences, such as ingredients, allergies, or simply no preference. The feature will in turn pass the preference to one of the API's depending on the nature of the preference. Upon finding a recipe that matches the preference, our feature will display a sample of the recipe to the user.

MVC

Model

The model will handle the processing and handling with our APIs. It will process any data to and from these APIs to fit the website's format and will handle the generated outcomes from the API before sending them to the controller. The handling will be any checking or adjusting the data received needs. This layer could also act as temporary storage to tailor for the user's experience as they navigate the site.

View

The view includes the frontend components for the UI. Presenting the meal plans and recipes to the user in an easy to view way. It will also display suggestions, nutritional information and other details, based on the users initial input and secondary choices. Interactive elements will allow the users to expand the appropriate data on their meal from dietary preferences, recipe detail and nutritional information. The view will update as the user navigates through the site.

Controller

The controller will serve as the "bridge" between the model and the view. It will process the user input, such as dietary preferences, or nutritional mandates, send this data to the model, and call the APIs. The Model will retrieve and organize the data, then the controller will update the view. The controller will also be responsible for making API calls through the model to optimize load times to hopefully ensure a positive user experience.

Potential class representation with MVC characteristics/structure:

