

# Release Plan Template – CSE 115a – Software Engineering

**Product Name:** NutriHall **Team Name:** NutriTech **Release Name:** NutriHall **Release Date:** Dec 10, 2024 **Revision Number:** 1 **Revision Date:** Oct 9, 2024

---

## High-Level Goals

- Implement a neural network to provide personalized meal plan features based on user preference and dietary goals.
  - Use real-time UCSC dining hall menus to display current meal suggestions.
  - Create a nutrition monitoring feature to track daily macro and calorie intake.
  - Be able to regenerate new meal suggestions in real time.
  - Create user profile functionality in order to store the preference of each user.
- 

## User Stories Defining the Scope of the Release

### Sprint 1

#### High Priority

- **User Story 1.1:** As a health-conscious student, I want to be able to log in to the website, view menus, and view all food/drink items live so that I understand what is offered from the dining hall. [3 story points]
  - Developer task
  - poi
- **User Story 1.2:** In Node.js, leveraging Puppeteer.js, write a web scraper that scrapes all menu items from the UCSC dining hall website. [6 story points]
- **User Story 1.3:** Write a script that runs the Puppeteer.js web scraper once a day so that menus remain up-to-date. [2 story points]
- **User Story 1.4:** Using the menu items scraped in User Story 1.2, connect and display them on the front-end. [4 story points]
- **User Story 1.5:** Create a list to choose which college dining hall menu the user wants to display. [2 story points]
- **User Story 1.6:** Build the PostgreSQL database to hold the data web-scraped. [4 story points]

#### Medium Priority - Spike

- **Spike:** Give some team members an opportunity to learn React and familiarize themselves with the front-end framework of choice for implementing the UI. [4 story points]

## **Sprint 2**

**User Story 2:** As a student who has dietary restrictions/allergies, I want to select and deselect preferences and restrictions so that I am not recommended meals I cannot eat.

### **Tasks:**

- Task 1: Design the front-end UI for selecting preferences and restrictions (e.g., allergens, dietary preferences). (3 hours)
- Task 2: Create a database schema to store user preferences and dietary restrictions. (3 hours)
- Task 3: Develop an API endpoint to save and retrieve user preferences from the database. (3 hours)
- Task 4: Implement the recommendation algorithm that filters meals based on user preferences and restrictions. (4 hours)
- Task 5: Integrate the recommendation system into the meal suggestions on the front-end. (3 hours)
- Task 6: Test the functionality to ensure meal suggestions adhere to restrictions. (2)

**User Story 3:** As a student who pays attention to their diet, I want to view past selected meals.

### **Tasks:**

- Task 1: Set up a front-end view to display a history of selected meals. (3 hours)
- Task 2: Create a back-end API endpoint to fetch past meal data from the database. (3 hours)
- Task 3: Update the database schema to store a record of meals selected by the user. (3 hours)
- Task 4: Integrate the API with the front-end to display past meal data. (3 hours)
- Task 5: Perform testing to verify that past meals display correctly. (2 hours)
- 

## **Sprint 3**

### **User Story 4:**

As a student who pays attention to their diet, I want to be able to search the meals available and add them to the list of things eaten so that I can calculate totals for the day.

### **Tasks:**

- Task 1: Create a search bar on the front-end to search meals by name or category. (3 hours)
- Task 2: Build a back-end API to perform searches based on user input. (3 hours)
- Task 3: Implement the functionality for users to add meals to a daily log. (3 hours)
- Task 4: Update the database schema to store daily meal logs for each user. (3 hours)

- Task 5: Create an API endpoint to calculate and return total nutrition values for the day. (3 hours)
- Task 6: Integrate the totals into the front-end view for the daily log. (3 hours)
- Task 7: Test the search functionality and meal logging system. (2 hours)

#### **User Story 5:**

As a student who pays attention to their diet, I want to be suggested meal plans that fit within my restrictions and adhere to my goals.

#### **Tasks:**

- Task 1: Design a front-end UI for meal plan suggestions. (3 hours)
- Task 2: Develop a meal plan algorithm that adheres to user restrictions and nutrition goals. (4 hours)
- Task 3: Create an API endpoint to generate and return meal plans. (3 hours)
- Task 4: Integrate the meal plan suggestions into the front-end interface. (3 hours)
- Task 5: Test meal plan generation and ensure it aligns with user restrictions and goals. (3 hours)

Total: 16 hours

### **Sprint 4**

**User Story 6:** As a student who pays attention to their diet, I want to be able to regenerate suggested meals if I dislike certain portions of the meal.

#### **Tasks:**

- Task 1: Create a "Regenerate" button on the meal suggestions interface. (2 hours)
- Task 2: Modify the meal plan algorithm to accept user feedback and regenerate meals accordingly. (4 hours)
- Task 3: Update the API endpoint to support meal regeneration based on dislikes. (3 hours)
- Task 4: Integrate the meal regeneration feature into the front-end. (2 hours)
- Task 5: Test the regeneration functionality to ensure it works as expected. (2 hours)
- User Story 7: As a student who pays attention to their diet, I want to be able to save my preferences and restrictions.
- 

**User Story 7:** As a student who pays attention to their diet, I want to be able to save my preferences and restrictions to a profile.

#### **Tasks:**

- Task 1: Create a UI to allow users to save or update preferences and restrictions. (3 hours)
- Task 2: Develop a back-end API to save and update preferences in the database. (3 hours)
- Task 3: Create the necessary database tables for storing user preferences. (2 hours)
- Task 4: Test saving and updating preferences functionality. (2 hours)

---

## Sanity Check

- **Total Story Points:** 25
- **Team Capacity:** 80 story points over four sprints
- **Distribution of Work:** Planning is still within team capacity for Sprint 1 so far, and a decent quantity of work has been assigned in such a manner as to ensure things run smoothly. There is even a spike included in this sprint, so some can familiarize themselves with React should UI work come up later in the project.

---

## Product Backlog

- Track users via GPS to capture distance information for bringing more convenient food options to them.
- A working website that scrapes the UCSC Dining Hall webpage for meals and drinks.
- The website will store user data including previous meals consumed, weight, preferred diet, and food restrictions to customize a meal plan for the user.
- The website will list details about the meals/drink items, such as protein, carbs, fats, etc. (nutritional information).
- Adding other restaurants from around Santa Cruz and other colleges to the database.

---

## NutriHall Features

- **Personalized Meal Plans:** NutriHall will auto-suggest meal options, considering dietary preferences and goals for an overall balanced and healthy diet.
- **Current Menus:** In real-time, see what's being served in UCSC dining halls. This allows you to select items from your plan but also gives you a sense of what is currently offered in the dining halls.
- **Nutrient Intake Tracking:** Follow daily calorie and nutrient intake to keep yourself on track with your health goals through a digestible breakdown of each meal.
- **Portion Control and Reduction of Food Waste:** Receive recommendations on portion sizes, which are calculated to maximize nutrition intake and minimize food waste in dining halls.
- **User Profile Creation:** Let user profiles hold user preferences that include diet restrictions, location, and goals.

---

## Roles

- **2 UI Developers**
  - **1 UI to Backend Developer**
  - **1 Backend Developer**
  - **1 Database Developer**
- 

## **Tech Stack**

- **Frontend:** React
- **Backend:** Node.js
- **Webscraper:** Puppeteer.js
- **Database:** PostgreSQL