

Project Architecture Document for NutriCost Analyzer

Student Names: itay derazon

Project Code: 87

Advisor: Ariel Roth

Abstract

"NutriCost Analyzer" is a web-based application designed to help users optimize their grocery shopping by creating personalized meal plans that meet specific nutritional targets. The app integrates real-time pricing data from various supermarkets and comprehensive nutritional information to ensure that meal plans are not only healthy but also cost-effective

1. Introduction

As interest in health and nutrition grows, consumers increasingly seek tools that can simplify the management of their dietary needs without compromising their budgets. "NutriCost Analyzer" addresses this need by combining advanced data analytics with user-friendly interfaces, offering a seamless solution for meal planning and grocery shopping

2. Problem Statement

Many consumers find it challenging to balance nutritional value and cost when planning meals, often due to a lack of integrated tools that can offer real-time data on both nutritional content and grocery pricing. This app aims to fill that gap by providing a comprehensive tool that simplifies decision-making for health-conscious budget shoppers

3. Objectives

1. Develop a web application that users can use to input their nutritional goals and receive tailored meal plans
2. Integrate real-time supermarket pricing data to provide cost-effective shopping options
3. Design a user-friendly interface that makes meal planning and shopping simple and enjoyable
4. Ensure data security and privacy through robust cloud services

4. User Stories

1. **As a user, I want to see daily meal suggestions based on standard nutritional guidelines, so I can follow a balanced diet without needing personalized settings**
2. **As a user, I want to easily view the cheapest options for ingredients from a standard shopping list, so I can save money on my groceries**
3. **As a user, I want to filter meal plans by major dietary restrictions (e.g., vegetarian, gluten-free), so I can avoid foods that don't meet my dietary needs**

4. **As a user, I want to save my favorite meals for easy access, so I can quickly find recipes I enjoy**
5. **As a user, I want to generate a basic shopping list from my meal plan, so I can ensure I buy all necessary ingredients**
6. **As a user, I want to adjust portion sizes for recipes, so they can match my personal or family's needs**
7. **As a user, I want a simple way to replace ingredients I don't like or have allergies to, so I can personalize meals to my taste and needs**
8. **As a user, I want to view nutritional information for each meal, so I can make informed dietary choices.**
9. **As a user, I want to access recipes with no more than five ingredients, so I can prepare simple and quick meals.**
10. **As a user, I want to reset my meal plan each week, so I can start fresh with new meal options.**

Sample Test Code for User Stories:

User Story 4: Save Favorite Meals

```
def test_save_favorite_meal(user_id, meal_id):  
  
    response = requests.post(f"http://localhost:5000/api/users/{user_id}/save_favorite",  
                             json={"meal_id": meal_id})  
  
    assert response.status_code == 200  
  
    assert response.json()["status"] == "Favorite saved"
```

User Story 5: Generate Shopping List

```
def test_generate_shopping_list(user_id):  
  
    response = requests.get(f"http://localhost:5000/api/users/{user_id}/shopping_list")  
  
    assert response.status_code == 200  
  
    assert "items" in response.json()
```

User Story 7: Replace Ingredients

```
def test_replace_ingredient(meal_id, old_ingredient, new_ingredient):  
  
    response = requests.put(f"http://localhost:5000/api/meals/{meal_id}/replace_ingredient",  
                             json={"old": old_ingredient, "new": new_ingredient})  
  
    assert response.status_code == 200
```

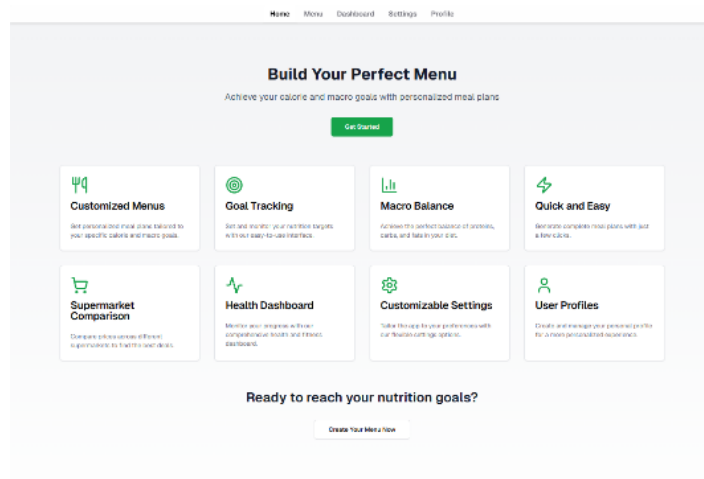
```
assert new_ingredient in response.json()["ingredients"]
```

User Story 10: Reset Meal Plan

```
def test_reset_meal_plan(user_id):  
  
    response = requests.post(f"http://localhost:5000/api/users/{user_id}/reset_meal_plan")  
  
    assert response.status_code == 200  
  
    assert response.json()["status"] == "Meal plan reset"
```

5. Application Screenshots

- **Home Screen:** The landing page of the web
- **Meal Planning Screen:** Displays suggested daily meal plans based on user input
- **Price Comparison Screen:** Shows pricing options from various supermarkets
- **Dashboard Screen:** Track and manage daily water, calorie, and macronutrient intake with customizable goals and visual progress indicators
- **User Profile or Login Screen:** Where users can log in and manage their profiles



The "Create Your Menu" screen has a navigation bar and a "Set Your Goals" section with input fields for Calories (2000), Protein (150), Carbs (250), Fat (55), and Number of Meals (3). A "Generate Menu" button is at the bottom of this section. Below is the "Your Personalized Menu" section, which displays two meal plans. Meal 1 includes Chicken Breast (150g), Brown Rice (100g), Lean Beef (100g), Avocado (100g), and Greek Yogurt (100g), with a detailed list of nutrients: 1000 cal (P: 77.4g) (C: 100g) (F: 100g), 120 kcal (P: 4.4g) (C: 30.8g) (F: 1.4g), 220 kcal (P: 22.0g) (C: 50g) (F: 10.2g), 20 kcal (P: 3.3g) (C: 3.3g) (F: 0.4g), and 80 kcal (P: 18.3g) (C: 8.3g) (F: 0.8g). Meal 2 is partially visible below.

HomeMenuDashboardSettingsProfile

Break: Yogurt 100g

55 cal | F: 10.0g | C: 5.2g | F: 0.0g

Meal 2

Chicken breast 100g

165 cal | F: 3.0g | C: 30.0g | F: 3.0g

Brown Rice 100g

110 cal | F: 0.4g | C: 23.0g | F: 1.4g

Leafy Greens 100g

20 cal | F: 0.5g | C: 0.0g | F: 1.2g

Almonds 50g

100 cal | F: 9.0g | C: 3.0g | F: 0.0g

Break: Yogurt 100g

55 cal | F: 10.0g | C: 5.2g | F: 0.0g

Meal 3

Chicken breast 100g

165 cal | F: 3.0g | C: 30.0g | F: 3.0g

Brown Rice 100g

110 cal | F: 0.4g | C: 23.0g | F: 1.4g

Leafy Greens 100g

20 cal | F: 0.5g | C: 0.0g | F: 1.2g

Almonds 50g

100 cal | F: 9.0g | C: 3.0g | F: 0.0g

Break: Yogurt 100g

55 cal | F: 10.0g | C: 5.2g | F: 0.0g

Nutritional Summary

Calories

1000 / 2000

Protein

100 / 150

Carbs

150 / 250

Fat

50 / 65

Supermarket Comparison

Supermarket Comparison

superA

\$44.95

superB

\$45.94

superC

\$43.37

Best option: superC

HomeMenuDashboardSettingsProfile

Health Dashboard

Water Intake

0 / 8 glasses

Daily Steps

0

Goal: 10,000 steps

Calories

0 / 2000

Protein

0g / 150g

Fat

0g / 65g

Carbs

0g / 250g

HomeMenuDashboardSettingsProfile

User Profile

Personal Information

Name

Email

example@gmail.com

Age

30

Height (cm)

170

Weight (kg)

70

Save Changes

6. Typical User Flows

1. Meal Planning Flow:

- User logs into the app → Inputs daily nutritional goals → Receives a personalized meal plan → Reviews and customizes the plan if necessary.

2. Grocery Shopping Optimization Flow:

- User reviews the meal plan → Navigates to the price comparison → Chooses products based on the best prices → Generates a shopping list based on selections.

7. Project Entity Diagram



8. Implementation

- **Backend:** Python with Flask to manage server-side logic.
- **Frontend:** React to create a responsive web interface.
- **Database:** MySQL for storing user data and meal plans.
- **Cloud Services:** AWS services for data storage and handling API requests securely