# Planning-spec

## General description of the project

- This app can help users easily calculate and track their daily calorie intake ~~and provide personalized meal plan recommendations based on their health goals, basic metabolism, dietary preferences, and nutritional goals~~. ***(I will abandon this feature because I don't know where to get these meal plans, especially for people from different countries and cultures.)***
- They can input their meals and calories manually, and then the app will analyze the user's eating habits and design diet plans to lose weight, gain muscle, or maintain weight. ***(As above, maybe I can just provide a recommended calorie goal.)***
- External:

  -Data resource: https://www.edamam.com/

  -Possible use of APIs to fetch the latest nutritional data.

  -Optional integration with fitness trackers or health apps for more comprehensive health monitoring.
- GUI:

  -A mobile application with a simple and clean interface.

  -Start as a web-based app with a user-friendly design.

  -The app's core functionality can be tested using a command-line interface (CLI).
- Possible Enhancements:

  -Include advanced features such as ~~recipe recommendations~~, deeper integration with fitness tracking apps, and additional metrics like macronutrient distribution.

  -Include the feature of the hydration track. ***(If we have enough time for this.)***

  -Include the feature of daily steps (goal: 10k steps). ***(Same as above.)***

  -A feature that allows users to choose their preferred unit of measurement.

  -A slider to adjust meal portion sizes, making calorie tracking more flexible.

  -An API for remote integration, enabling other applications to use the personalized diet and calorie tracking functionalities.

  -Add a food list. When users add a food item, weight, calories, these inforation will be shown underneath the button making intaken food information obvious.

  -Circular Progress Ring and Calorie Limit Alert. Implement a circular progress ring to visually represent the user's remaining daily calorie allowance, and trigger an alert if the user exceeds their calorie limit.

  ***(Something new!!)***

## Task Vignettes (User activity "flow")

- Task 1: Input personal information and health goals

  -age, gender

  -weight, height

  -activity level (low, medium, high)

  -health goal (weight loss, muscle gain, weight maintenance)

  -dietary preference (normal, vegetarian, low-carb)
- Task 1 user activity:

  -open the app and input their personal information (age, gender, weight, height)

  -select their activity level, dietary preferences from a predefined list

-choose health goal(e.g., weight loss) and input the target weight
-analyze the information (e.g., check if the target weight is realistic)
-calculate basic metabolism and daily calorie needs based on activity level and health goal

- Task 2: Log daily meals

-food items (input manually or select from the list)
-quantity (grams, servings…)
-which meal (breakfast, lunch, dinner, snack)

- Task 2 user activity:

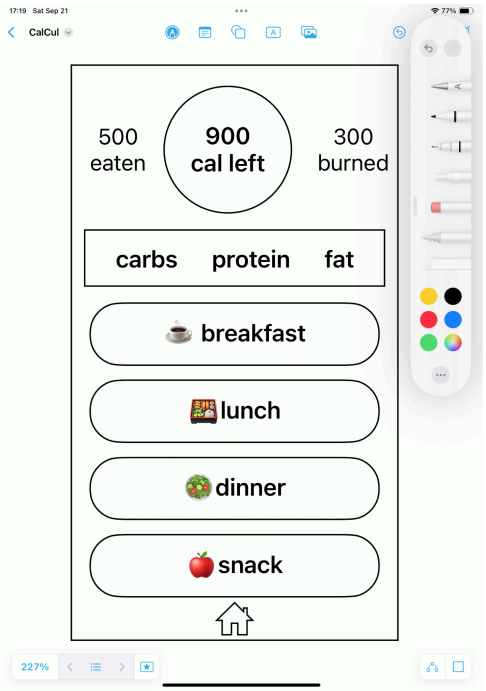
-click a button from those of breakfast, lunch, dinner, or snack, then open a meal entry form
-search food items, select portion
-calculate the total calories of those items selected and display the proportion on the circle or a bar

- ~~Task 3: Generate a personalized meal plan~~

~~-meal preferences (e.g., vegetarian, low-carb, high-protein)~~
~~-allergy or food restrictions~~

- ~~Task 3 user activity:~~

~~-select meal preferences and restrictions in a page~~
~~-generate a personalized meal plan after analyzing meals~~
~~-accept the plan or change something~~




**Technical 'flow'**

- Data structures:

-user personal information (dict/object)
-food database (API, panda dataframe)
-meal log (list of dict/objects)

-daily calorie summary (dict)
-personalized meal plan (list of dict/objects)

- Core functions:

# get_user_information():
-input age, gender, activity…
-output a dictionary of user's personal inofrmation
Flow: user input→get_user_information()→ Data stored in the user profile object

# get_food_data(food_item):
-input food item name
-output food calories and macronutrition from API

# log meal(food_item, portion_size):
-input food item and portion size
-output the meal log
Flow: User input → get_food_data() → Food data from API → log_meal() → Meal log updated

# calculate_daily_summary():
-input logged meals, calorie target
-output the total calorie consumption and remaining calorie budget
Flow: Meal log data → calculate_daily_summary() → Output displayed on UI.

~~# generate_personalized_plan():~~
~~-input user profile, food preferences, health goal, ect…~~
~~-output personalized meal suggestions~~
~~Flow: User profile & meal log → generate_personalized_plan() → Meal plan suggestions displayed~~

**Final (self) assessment**

- Biggest unexpected change:
  Personalized diet plan. I realized it's hard to separate data flow from user inputs and external API. Besides, people from different countries have diverse tastes, so it's hard to generate cuisines from different countries. For example, China. I love Chinese food!!!
- Confidence in implementation:
  Honestly, I'm confused about the whole procedure, but I will try my best to follow this course step by step. I can do it!
- Biggest potential problem and least familiar parts:
  Integrating external APIs into Flask and managing real-time data flows.