1. Positioning

1.1 Problem statement

The problem of	Not having enough time to eat properly
Affects	College students
The impact of which is	That students are less healthy both physically and mentally.

1.2 Product Position Statement

Product Position Statement

For	College students
Who	Need to track their food intake and financial decisions
The website CalPal	Which is a calorie and fitness tracker
That	Is able to help students consistently eat healthier.
Unlike	MyFitnessPal.
Our product	Our website uses a budgeting program that keeps track of the spending of users.

1.3 Value proposition and customer segment

Value proposition: CalPal is a calorie tracker/fitness website that helps college students eat healthier meals that meet their individual needs and fitness goals.
Consumer segment: College students with busy schedules who may not eat as healthy as they would like.

2. Stakeholders

Northern Arizona University - The development of this website is under the supervision of Northern Arizona University.

MyFitnessPal - The main competitor of our website.

College students - The clients that our website is appealing to.

Rita Bolanos, Isidro Marquez, Logan Bankert, Annaliese Dedmore, Luke Flaker, Nile Ham - The developers of the project.

3. Functional requirements (features)

1. A calorie tracker that allows users to keep track of how many calories they've consumed in a given day.
2. A fitness goal setter that allows users to set specific health goals that affect the calorie tracker's success point.
3. A macros tracker that allows users to track how much protein, fiber, fat, etc. they've consumed on a given day.
4. A cost calculator that calculates the cost of a given meal.
5. An allergy setter that allows a user to specify allergies.
6. A weight tracker that keeps track of the user's entered weight.

4. Non-functional requirements

1. Motivational statements to keep users motivated about their goals.
2. The app shall respond within a 2 second timespan.
3. The app must be able to function without an internet connection.
4. The app shall be accessible on multiple platforms.

5. MVP

Our MVP will have the following features validated:

Calorie tracker: Students are able to track meals to see the caloric intake

Macrostracker: Students are able to track the macronutrients of their meals.

Goal setter: Students will be able to set personal goals and the app will help students meet them.

Budget calculator: Student will be able to save money using the app.

It will be validated by fulfilling the following requirements:

It will update calories, macros, goals in less than two seconds after inputting it. Students will report at least one budget friendly healthier food option. All information gets saved locally, meaning it can be used offline.

6. Use cases

6.1 Use case diagram
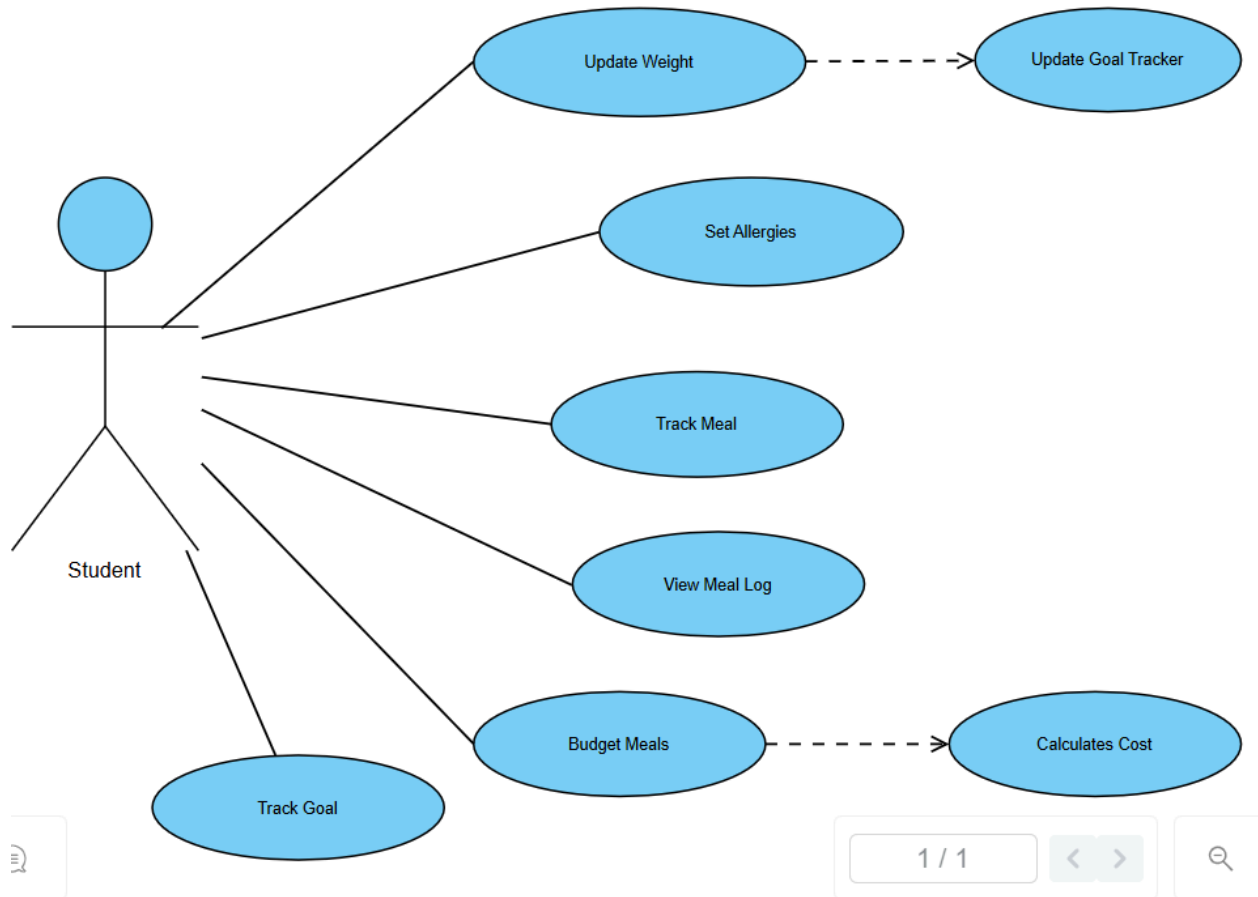
6.2 Use case descriptions and interface sketch

1. Use case #1: Weight is updated.
   Actor: Busy college student.
   Trigger: A user enters a new weight.
   Pre-conditions: A previous weight was entered or is being entered for the first time.
   Post-condition: The weight tracker and fitness goal progress is updated.
   Success Scenario:
      1. The student enters a new weight into the tracker.
      2. The tracker stores the old weight.
      3. The tracker updates the user's current weight.
      4. The tracker updates the user's current fitness goal progress.
   Alternate Scenario 1:
      1. The tracker has no prior weights entered.
      2. The tracker begins tracking the entered weight as a starting point.

Weight Trend

195

190

185

180

175

Mar   Apr   May   Jun   July

Current Weight: 176.3lbs
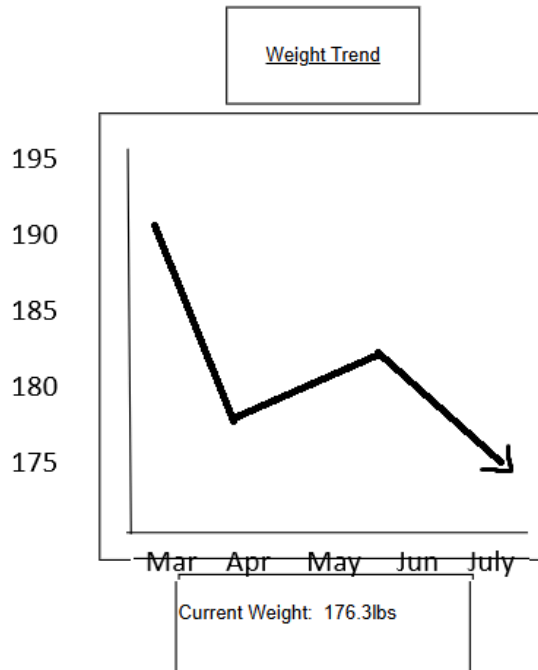
2. Use Case #2: Concern about nutrients.
   Actor: Health conscious college Student.
   Trigger: A user finishes a meal and wants to track that meal.
   Pre-Conditions:The user has access to the app and can select any food item.
   Post-Conditions: The meal is saved to the daily log and macros are updated.
   Success Scenario:
   1. The student opens the application and selects "add a food item".
   2. The student inputs the food number along with how many calories and macros
      there are.
   3. The student saves the food item and daily nutrient numbers are updated.

   Alternate Scenarios:
   1. The app can't find the name of a food item: the app will inform the user.
   2. No internet connection: The app will switch into offline mode and store any data
      locally.
   3. Incomplete entry: The app will highlight which fields of the log are missing and
      inform the user.

```
┌─────────────────┐
│ Nutrient Tracker│
└─────────────────┘

┌──────────────────────────┐
│  ┌────────────────────┐  │
│  │ Ingredient 1:      │  │
│  └────────────────────┘  │
│  ┌────────────────────┐  │
│  │ Ingredient 2:      │  │
│  └────────────────────┘  │
│  ┌────────────────────┐  │
│  │ Add+:              │  │
│  └────────────────────┘  │
│                          │
│                          │
│  ┌────────┐  ┌────────┐  │
│  │Calories│  │ Macros │  │
│  └────────┘  └────────┘  │
└──────────────────────────┘
```

3. Use Case #3: Worry about meals
   Actor: Lazy college student
   Trigger: A user hates worrying about what they're going to eat for each meal.
   Pre-Conditions: The user has meals that are previously logged.
   Post-Conditions: The selected meal shows exactly what ingredients are needed.
   Success Scenario:
   1. The student picks a meal already logged in the app.
   2. The required ingredients and the necessary measurements are listed.
   3. The user confirms they have purchased the ingredients for the meal.
   4. The meal is created.

   Alternative scenario:
   1. There are no meals logged in the journal.
   2. The user does research on potential meals.
   3. The user adds the meal to the app for future reference.
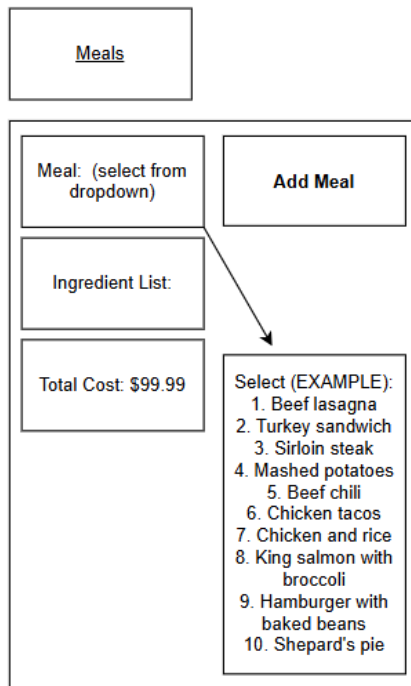
4. Use Case #4: Worried about allergies.
   Actor: Health risk worried student.
   Trigger: A user wants to be able to have health concerns be something they can see.
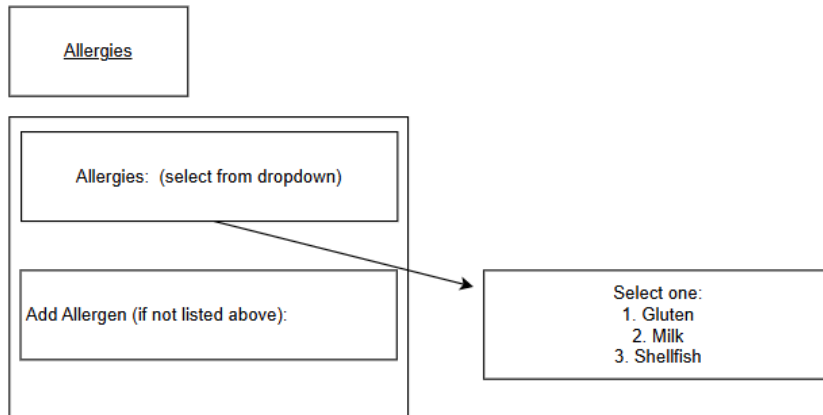   Pre-Conditions: User wants to add an allergen.
   Post-Conditions: The allergen is logged and foods are properly excluded based off chosen allergen.
   Success Scenario:
   1. A user adds an allergy to the allergy tab.
   2. Tracker stores allergy information into a list.
   3. Users are able to see the allergy list even if they leave and come back.
   Alternative Scenario:
   1. A user finds a common allergen that's already in the app.
   2. The user now is able to see which foods contain that allergen.
   3. The user now doesn't have to worry about eating something that may contain that allergen.

```
┌─────────────┐
│  Allergies  │
│             │
└─────────────┘
┌──────────────────────────────┐
│ ┌──────────────────────────┐ │
│ │ Allergies: (select from  │ │
│ │ dropdown)                │ │──────┐        ┌────────────────────┐
│ └──────────────────────────┘ │      └───────▶│   Select one:      │
│                              │               │   1. Gluten        │
│ ┌──────────────────────────┐ │               │   2. Milk          │
│ │ Add Allergen (if not     │ │               │   3. Shellfish     │
│ │ listed above):           │ │               └────────────────────┘
│ └──────────────────────────┘ │
│                              │
└──────────────────────────────┘
```

5. Use Case #5: Motivated to set goals.
   Actor: Motivated student.
   Trigger: The student selects a fitness goal.
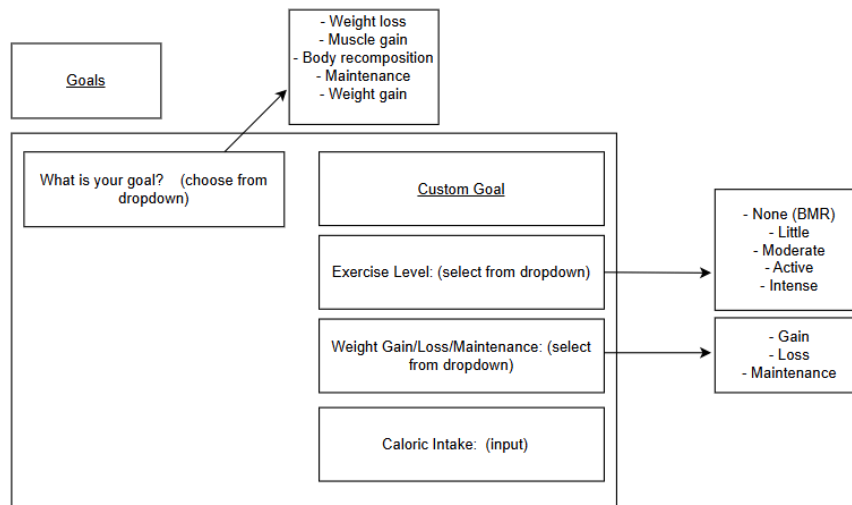   Pre-Conditions:Other goals have been selected or not.
   Post-Conditions:The goal is added to the list of a user's personal goals.
   Success Scenario:
   1. The user selects a goal from the list.
   2. The goal is added to the users personal goals.
   3. The different trackers on the app are updated to reflect the new goal.
   Alternate Scenario:
   1. The user makes their own custom goal.
   2. The user adds the name and qualifications of the goal.
   3. The goal is applied to the users personal goals.
   4. The different trackers on the app are updated to reflect the new goal.

```
                              ┌──────────────────────┐
                              │ - Weight loss        │
                              │ - Muscle gain        │
┌─────────────┐               │ - Body recomposition │
│   Goals     │               │ - Maintenance        │
│             │          ┌───▶│ - Weight gain        │
└─────────────┘          │    └──────────────────────┘
┌────────────────────────┼──────────────────────────────────────────────┐
│ ┌──────────────────────┴──┐   ┌────────────────────────────┐           │
│ │ What is your goal?       │   │      Custom Goal           │           │
│ │ (choose from dropdown)   │   │                            │           │
│ └─────────────────────────┘   └────────────────────────────┘  ┌──────────────────┐
│                               ┌────────────────────────────┐   │ - None (BMR)     │
│                               │ Exercise Level: (select    │──▶│  - Little        │
│                               │ from dropdown)             │   │ - Moderate       │
│                               └────────────────────────────┘   │  - Active        │
│                                                                 │ - Intense        │
│                               ┌────────────────────────────┐   └──────────────────┘
│                               │ Weight Gain/Loss/          │   ┌──────────────────┐
│                               │ Maintenance: (select       │──▶│  - Gain          │
│                               │ from dropdown)             │   │  - Loss          │
│                               └────────────────────────────┘   │ - Maintenance    │
│                                                                 └──────────────────┘
│                               ┌────────────────────────────┐
│                               │ Caloric Intake: (input)    │
│                               └────────────────────────────┘
└───────────────────────────────────────────────────────────────────────┘
```

6. Use Case #6: Conscientious about spending.
   Actor: College student on a budget.
   Trigger: Student logs the meal and system provides budget alternatives.
   Pre-Conditions: A user enters a meal that they would like to cook but budget-friendly.
   Post-Conditions: App recommends cheap options that cater to the user's tastes.
   Success Scenario:
      1. The user selects a meal that is low cost.
      2. The user views the required ingredients and most cost-effective ways of getting required ingredients.
      3. The user selects that meal and adds the cost of ingredients to their budget.
   Alternate Scenario:
1. The user selects ingredients that are cheap to purchase.
2. The user views suggestions from the app regarding meals that include those ingredients.
3. The user selects the respective meal and its ingredients and saves that meal for the future.
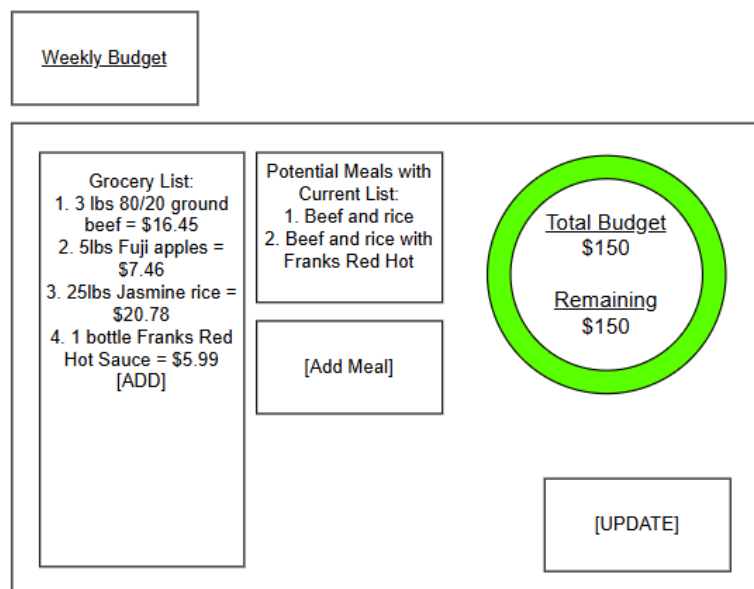
7. User stories

1. As a student athlete, I want to keep track of my carb intake so that I can carb-load more effectively.
2. As a busy student, I want to keep track of my calorie intake so that I eat enough to not be tired during class.
3. As a student, I want to save frequently eaten foods so that I can easily access them.
4. As a budget conscious student, I want to see the cost per food item so that I can stay within my budget.

5. As a lazy student, I want to be able to track my total targeted macros vs. logged macros.
6. As a health conscious student, I want to be able to track all my allergies to make sure I remember what to avoid.
7. As a student committed to improving my personal health, I want to be able to view all of my health details and goals all in one place.
8. As a newly-independent student, I want to be able to track all of my health changes as I adapt to college life.
9. As someone who enjoys eating healthy and saving money, I want to be able to track what I eat and spend at the same time.
10. As someone who frequently does intense workouts, I want to be able to track what I eat throughout the year depending on what my goals are.
11. As a freshman college student with a limited dining plan, I want to be able to juggle my spending on other meals alongside using my dining plan to ensure I have enough until the semester ends.
12. As a busy college student, I want to be able to keep track of what I spend at the grocery so as to not go over my weekly budget.

8. Issue Tracker

[link to the repo issue tracker and a screenshot of it containing every user story as its own issue]
https://github.com/loganb7869/CS386-Project-Repo/issues