

BoilerBite

Team 16 - Product Backlog

Ridwan Chowdhury, Isha Mahadalkar, Jeremy Yang, Uday Chaudhary

Problem Statement

Students at Purdue University often have a tough time keeping track of their nutrition. Between rushing to classes and keeping up with homework, there simply isn't time for Purdue students to meticulously plan out their meals and ensure that they meet their own dietary goals. BoilerBite's goal is to make that process easy and accessible for students on Purdue's meal plans. By creating an application where tracking dining court meals takes only a few simple taps, we aim to make nutrition an attainable goal.

Background Info:

Domain and Problem Analysis

Students at Purdue often have a problem with keeping track of their calorie intake during a semester since there is no app to provide them with the knowledge of the food being served at the dining courts. This often causes them undue stress, as they lose or gain more weight than they desire while attempting to keep up with their course load. The best way to combat this would be to develop a time-efficient way for students to track what they eat. With BoilerBite, we want to give students the convenience of spending just a few minutes a day to keep track of all their macronutrients. BoilerBite will be targeting the domain of nutrition apps as it focuses on making a meal plan for students according to their specific calorie requirements.

Targeted Users

This app is for anyone who uses the dining courts and wants to stick to a diet plan that provides a specific amount of calories per day. Students have a lot of work during a semester and therefore, healthy eating is important for students to prevent themselves from falling ill or becoming over/underweight.

Similar Platforms

There are certain apps that inform students of the various foods at specific dining courts during different times of days like Purdue Meals and Purdue Menu. Both these apps only give information about the food items being served at each dining court on a specific day without tailoring it for a specific user. There are also a lot of meal planning apps like PlateJoy and FoodPrint which allow you to create a daily nutrition plan and keep track of your daily calorie intake.

Limitations

One limitation of other apps is that they are not specific to Purdue. The user has to work with a much more complicated interface with having to enter detailed information for every meal that they eat. These apps might also suggest food items that are not available in the dining courts. Other dining apps Purdue students currently use are currently unable to give distinct recommendations based on the calorie specifications given by the user nor can they keep track of each individual's nutrition goals. With BoilerBite we aim to implement the positive aspects of all these applications into one intuitive application designed specifically for Purdue students.

Functional Requirements

1. As a user, I want to be able to create an account.
2. As a user, I want to be able to login.
3. As a user, I want to be able to delete my account when I want to.
4. As a user, I want to be able to input my weight, height, and age into my profile.
5. As a user, I want to be able to input my desired daily calorie intake into my profile.
6. As a user, I want to be able to access my account anytime.
7. As a user, I want to be able to update my profile with a new weight, height, and age.
8. As a user, I want to be able to choose the units used to display my information (lb. vs kg).
9. As a user, I want to be able to compare my current weight to my past weight.
10. As a user, I want to be able to filter the foods recommended to me.
11. As a user, I want to be able to filter out certain dining courts.
12. As a user, I want to be able to check the main ingredients of the items on the menus.
13. As a user, I want to be able to check for allergens in the dishes of the dining courts .
14. As a user, I want to be able to provide feedback/report bugs to the developers.
15. As a user, I want to be able to set a desired body weight as a goal.
16. As a user, I want to be able to set a desired daily calorie intake as a goal.
17. As a user, I want to be able to remove my weight goal.
18. As a user, I want to be able to remove my daily calorie intake goal.
19. As a user, I want to be able to see how far away I am from my desired weight.

20. As a user, I want to be able to see how far away I am currently from my desired daily calorie intake.
21. As a user, I want to be able to check open hours of the dining courts.
22. As a user, I want to be able to view the menu of the dining courts.
23. As a user, I want to be able to record the dishes I eat each meal.
24. As a user, I want to be able to know how many calories I ate each meal, calculated based on the dishes that I recorded.
25. As a user, I want to be reminded when I consume a certain amount of calories over the limit that I desire.
26. As a user, I want to be able to see how many carbs, grams of protein, fat, etc. I consume each meal.
27. As a user, I want to be reminded when I consume a certain amount of calories under the limit that I desire.
28. As a user, I want to be able to turn off notifications about consuming too many calories.
29. As a user, I want to be able to turn off notifications about consuming too few calories.
30. As a developer, I want to be able to calculate how much carbs, protein, fat, etc. the users have consumed each meal.
31. As a developer, I want to be able to recommend users to add certain food to increase their calorie intake if they are not eating enough.
32. As a developer, I want to be able to recommend users to remove certain food to decrease their calorie intake if they are over eating.
33. As a developer, I want to be able to send a notification to the user that they have met the calorie intake for the meal.
34. As a developer, I want to make it easy for the users to navigate through different pages.
35. As a developer, I want to be able to present a visualization of daily/weekly progress of the users (whether or not the users met their goal of the day).
36. As a developer, I want to be able to delete a user's account from the server.
37. (If time allows) As a user, I want to be able to record names, and nutrition information of food items being consumed which are not present/eaten in the dining courts (e.g Ramen, and its nutrition information).
38. (If time allows) As a user, I want to be able to select the new items recorded and add them to my meals.
39. (If time allows) As a user, I want to be able to set a desired daily water intake as a goal.
40. (If time allows) As a user, I want to be able to remove my desired daily water intake goal.

41. (If time allows) As a user, I want to be able to see how far away I am currently from my desired daily water intake.
42. (If time allows) As a developer, I want to be able to send users emails with information regarding the app (server down, server maintenance downtime).
43. (If time allows) As a developer, I want to be able to recommend a meal plan based on the users' information.

Non-functional Requirements:

Architecture and Performance

Our plan is to develop the frontend and backend separately, so as to make it easier to divide up the work. In addition, separating the development in this fashion makes it easier to pinpoint where bugs occur. The frontend will be developed in Swift 5.1, the native language for iOS platforms. Swift is an extremely fast platform built with performance in mind; in fact, Swift is 2.6x faster than Objective-C and 8.4x faster than Python. This is important in order for our user interface to feel very fluid and responsive. The frontend will have the GUI for the application, and will pull data and requests from our backend.

Our backend will be developed using mainly SQL(Structured Query Language) to develop a relational database where the frontend can store and pull information from. SQL is considered the standard language for relational database systems everywhere, and is extremely popular both for its versatility and scalability. Databases using SQL can become very large, containing millions of rows and processing thousands of requests at once, before there is any sort of slowdown. This is extremely necessary for our project, as our goal is to be able to process a maximum of 1000 server requests per minute with an average response time of less than 500ms per request. With these requirements for our server, we believe we can meet our estimated maximum load for the application. As there are many versions of SQL available due to its popularity, we will utilize the open-source MySQL for our project.

Security

We are going to take the security of the application extremely seriously since it is going to contain user sensitive information. We will be implementing a permissions system to make sure a user can only access their own information as well as ensure that the user data is secure. Since, we only have one "type" of users which are the students we do not need to worry about the data being accessed by a different type of user. Additionally, all the requests to API's used will be validated to prevent malicious

requests and all accesses to private resources and data will be verified and forced to comply with security policies. In order to protect the sensitive info stored in the SQL database, we will be encrypting user info to prevent easy access and auditing any logins into the database to ensure no bad actors get access to the sensitive data. Finally, to protect against the threat of SQL injections, we will be implementing stored procedures instead of direct SQL queries to ensure non-compliant requests cannot access the database.

Usability

We want the students using the application to have a streamlined user experience. We want the students to easily navigate the application with just a few simple taps and find information about each food item, like the ingredients, allergens and calories. We want to make sure the users can easily filter, add and delete the food items for each meal and have the information being updated in real time on the screen so that they can create their perfect meal. We would like the app to be usable 24/7 so students can access their info whenever they would like.

Hosting/Deployment

Separation of the frontend and backend of our project allows us to develop, host, and (most importantly) deploy them separately. Once we have finished the initial version of the frontend, we can host the application on the iOS appstore for public download and usage. After development of the backend has completed, we plan to deploy it to a Google Cloud Platform server, where our program will be hosted. Google Cloud Platform has a free tier that allows you to run 1 small server for free; this is what we will be using for our project. Using the Google Cloud Platform will allow us to meet our goal of running our server 24/7.