# Bazaar

Product Backlog

## Problem Statement:

Meal planning and college students don't go together. It's usually a mammoth task for students to plan their meals for the day, let alone plan for an entire week. With Bazaar we plan to make it easier to plan meals for a day or a week. Our goal is simple; we provide a list of common recipes that college students can make easily by allowing them to create their own recipes or select from our list of recipes, checking of ingredients that they already have and giving them a shopping list of items they will need and also providing the cheapest store they can buy it at near them. Bazaar will also use machine learning to suggest recipes to the user based on previous choices/preferences.

## Background Information:

As college students, all members of our team have at one point experience the problem of not knowing what we want to make for a meal. Whether it is a quick meal or a weekend feast, college student's usually have a problem making meals. Currently there are different recipe hub websites, but we want to create a hub combined with a shopping list, and a machine learning algorithm to offer an insight on whether or not the user would like the dish they are viewing. This app is not only limited to college students, because everyone faces the issue of making meals for themselves, Bazaar will be able to help all people with their recipe needs.

## Environment:

We will use HTML for the static pages and ReactJS for user interaction of the front end of the web application and NodeJS for the back end of the web application. We are using firebase for authentication, and we are also using it for our database to hold user information such as user preferences, saved recipes, etc.

## **Background Information:**

Functional

| Backlog Id | Functional Requirements | Hours | Status |
|---|---|---|---|
| 1 | As a user, I would like to be able to sign in to Bazaar using my Google Account. | 4 | Planned for Sprint 1 |
| 2 | As a user, I would like to change my username. | 2 | Planned for Sprint 1 |
| 3 | As a user, I would like to search for recipes on the application. | 10 | Planned for Sprint 1 |
| 4 | As a user, I would like to save my favorite recipes. | 2 | Planned for Sprint 1 |
| 5 | As a user, I would like to add or remove ingredients to a shopping list. | 3 | Planned for Sprint 2 |
| 6 | As a user, I would like to share recipes on social media. | 2 | Planned for Sprint 2 |
| 7 | As a developer, I would like to inspect my code, unit inspect my code, and inspect the design of my code. | 100 | Planned for Sprint 1 and 2 |
| 8 | As a user, I would like to see a prediction of my liking towards the recipe | 15 | Planned for Sprint 2 |
| 9 | As a user, I would like to add my dish preferences. | 5 | Planned for Sprint 1 |
| 10 | As a user, I would like to sign out out of my Google account | 1 | Planned for Sprint 2 |
| 11 | As a user, I would like to see | 3 | Planned for |

| | | | |
|---|---|---|---|
| | youtube videos of the dish I am making. | | Sprint 2 |
| 12 | As a user, I would like to create recipes for others to view | 5 | Planned for Sprint 2 |
| 13 | As a user, I would like to upvote or downvote recipes. | 2 | Planned for Sprint 1 |
| 14 | As a user, I would like to comment on recipes | 2 | Planned for Sprint 2 |
| 15 | As a user, I would like to have a calendar where I can view all the meals for a week. | 8 | Planned for Sprint 2 |
| 16 | As a user, I would like to plan my meals for the week | 10 | Planned for Sprint 2 |
| 17 | As a user, I would like to get an email everyday about my meals for that day | 3 | Planned for Sprint 2 |
| 18 | As a user, I would like to edit how often I get emails. | 1 | Planned for Sprint 2 |
| 19 | As a user, I would like the option to receive text messages instead of emails. | 3 | Planned for Sprint 2 |
| 20 | As a user, I would like the app to keep track of the calories I intake from the meals I have. | 4 | Planned for Sprint 2 |

## Non-Functional

| Backlog Id | Non-Functional Requirements | Hours | Status |
|---|---|---|---|
| 21 | As a user, I would like personal information to remain secure | 4 | Planned for Sprint 2 |

| | | | |
|---|---|---|---|
| | and private | | |
| 22 | As a developer, I would like to learn how to use ReactJS | 10 | Planned for Sprint 1 |
| 23 | As a developer, I would like to learn how to use Firebase | 3 | Planned for Sprint 1 |
| 24 | As a developer, I would like to learn how to setup a local test environment | 2 | Planned for Sprint 1 |
| 25 | As a developer, I would like to learn how to use NodeJS. | 10 | Planned for Sprint 1 |
| 26 | As a developer, I would like to learn how to unit test in ReactJS. | 8 | Planned for Sprint 2 |
| 27 | As a developer, I would like to learn how to unit test in NodeJS. | 8 | Planned for Sprint 2 |

# Use Cases:

**Case: Sign in using my Google Account**

| **Action** | **System Response** |
|---|---|
| 1. Click the Google Sign In button | 2. Opens Google login popup |
| 3. Input username and password | 4. Login user and obtain relevant information |
| | 5. Close Google login popup after auth is complete |

**Case: Change username**

| **Action** | **System Response** |
|---|---|

| | |
|---|---|
| 1. Return to preferences page | |
| 2. Enter new username in username text field | |
| 3. Press submit button | 4. Update username in Firebase |
| | 5. Alert user that update was successful |

**Case: Search for recipes**

| **Action** | **System Response** |
|---|---|
| 1. Go to search bar and enter dish name | |
| 2. Click search | 3. Server retrieves a list from the database |
| | 4. Front end renders the JSON list |

**Case: Add Favorites**

| **Action** | **System Response** |
|---|---|
| 1. Go to recipe page | |
| 2. Click the “Save” button | 3. Server adds recipe to database under user |
| | 4. Front end alerts user that recipe is saved |

**Case: Add items from shopping list**

| **Action** | **System Response** |
|---|---|
| 1. Go to recipe page | |
| 2. Next to each ingredient click Add or Add All Ingredients | 3. Server adds ingredients to ingredient list under specific user |

| | |
|---|---|
| | 4. Front end alerts user that ingredient is saved |

**Case: Remove items from shopping list**

| **Action** | **System Response** |
|---|---|
| 1. Go to shopping list page | |
| 2. Next to each ingredient click Remove Ingredient or Remove All Ingredients | 3. Server removes ingredients to ingredient list under specific username |
| | 4. Front end alerts user that ingredient has been successfully removed |

**Case: Share on Facebook**

| **Action** | **System Response** |
|---|---|
| 1. Click on share recipe button | 2. Facebook API handles sharing |
| | 3. Front end alerts user that recipe has been successfully saved |

**Case: Prediction of liking a recipe**

| **Action** | **System Response** |
|---|---|
| 1. Go to recipe page | 2. Server uses recipe details and user preferences to predict if the user will like the recipe or not |
| | 3. Front end loads like or dislike element next to recipe |

**Case: Add dish preference**

| **Action** | **System Response** |
|---|---|
| 1. Either access from sign up page or go to the Preference page | 2. Server loads current list of preferences |
| 3. Allows you to change preferences | 4. Server updates preference list in database |
| | 5. Front end alerts user that ingredient has been successfully added |

**Case: Logout from Google Account**

| **Action** | **System Response** |
|---|---|
| 1. Click on logout button on navigation bar | 2. Server redirects back to sign-in page |

**Case: Access youtube videos of dish**

| **Action** | **System Response** |
|---|---|
| 1. Go to a recipe page | 2. Server uses Youtube API to search with recipe name for relevant videos |
| | 3. Server retrieves video code and embeds videos on frontend |
| 4. User can click play on any video loaded | 5. Video starts to play |

**Case: Create Recipes**

| **Action** | **System Response** |
|---|---|
| 1. Go to Create Recipe page | |
| 2. Fill out form that contains ingredients, calories, instructions,cooking time, tags. | |

| | |
|---|---|
| 3. Click submit recipe button | 5. Server adds recipe to database and alters user that recipe has been created |

**Case: Upvote or Downvote**

| **Action** | **System Response** |
|---|---|
| 1. Go to recipe page | |
| 2. Click upvote or downvote button | 3. Server takes response and saves it under user preferences |
| 3. Click submit recipe button | 4. Server alerts user that response has been recorded and updates public count of upvotes and downvotes |

**Case: Commenting on recipes**

| **Action** | **System Response** |
|---|---|
| 1. Go to recipe page | |
| 2. Type out comment in the comment box | |
| 3. Click submit comment button | 4. Disqus API handles comment request and updates comment section |

**Case: Calendar**

| **Action** | **System Response** |
|---|---|
| 1. Go to calendar page | 2. Server loads calendar for user's google account |
| | 3. Front-end loads Google's embedded calendar |

**Case: Upvote or Downvote**

| **Action** | **System Response** |
|---|---|

| | |
|---|---|
| 1. Go to recipe page | |
| 2. Click upvote or downvote button | 3. Server takes response and saves it under user preferences |
| 3. Click submit recipe button | 4. Server alerts user that response has been recorded and updates public count of upvotes and downvotes |

**Case: Planning meals**

| **Action** | **System Response** |
|---|---|
| 1. Users goes to meal planning page | |
| 2. User selects from saved meals or can select from a recipe page and selects a day and time for selected meal | 3. Server takes meal and time and use Google calendar API to add an event to the calendar |

**Case: Meal emails**

| **Action** | **System Response** |
|---|---|
| 1. Sign up for email alerts | 2. Sends email alerts based on user preferences |

**Case: Email preferences**

| **Action** | **System Response** |
|---|---|
| 1. Go to preferences page, under email preferences, select desired email frequency. | 2. Server takes user response and updates user preference in database |

**Case: Text messages**

| **Action** | **System Response** |
| --- | --- |
| 1. Go to preferences page, select text alerts instead of email alerts. | 2. Server takes user response and updates user preferences in database |

**Case: Calorie tracker**

| **Action** | **System Response** |
| --- | --- |
| 1. Go to the calendar page | 2. Server will retrieve caloric value of each meal along with totaling up the total calories for day and week. |
| | 3. Front-end will display caloric value |