# Affamato: Phase Two Report

## Team:

Canvas Group: **Falcon**

**Phase Two Team Lead:** Cameron Clark

## Members:


- Cameron Clark: cameron.clark0821@utexas.edu \\ github.com/cameronclark0821
- Justin Henry: justinhenry@utexas.edu \\ github.com/justinhenry
- Alex Issa: alex.issa32@utexas.edu \\ github.com/alexissa32
- Julia Rebello: julialrebello@utexas.edu \\ github.com/JLRebello
- Samir Riad: sriad123@utexas.edu \\ github.com/sriad123
- Rooshi Patidar: rooshipatidar@utexas.edu \\ github.com/rooshimadethis


**URL** to Github/Gitlab repo and shared Google docs:

- Github: https://github.com/alexissa32/Affamato
- Google Drive: https://drive.google.com/drive/folders/0ANUp-cLx6lnZUk9PVA
- Slack: https://team-falcon-group.slack.com

**Website URL:** https://www.affamato.xyz/

**Phase II Report Contents:**

1. Goals and Accomplishments
2. Design and Requirements
3. Scraping and Database (Description)
4. Scraping and Database (Screenshots)
5. Screens, Features and Functionality (Description)
6. Screens, Features and Functionality (Screenshots)
7. Tools, Software and Frameworks Used
8. Testing
9. Updated Phase Goals

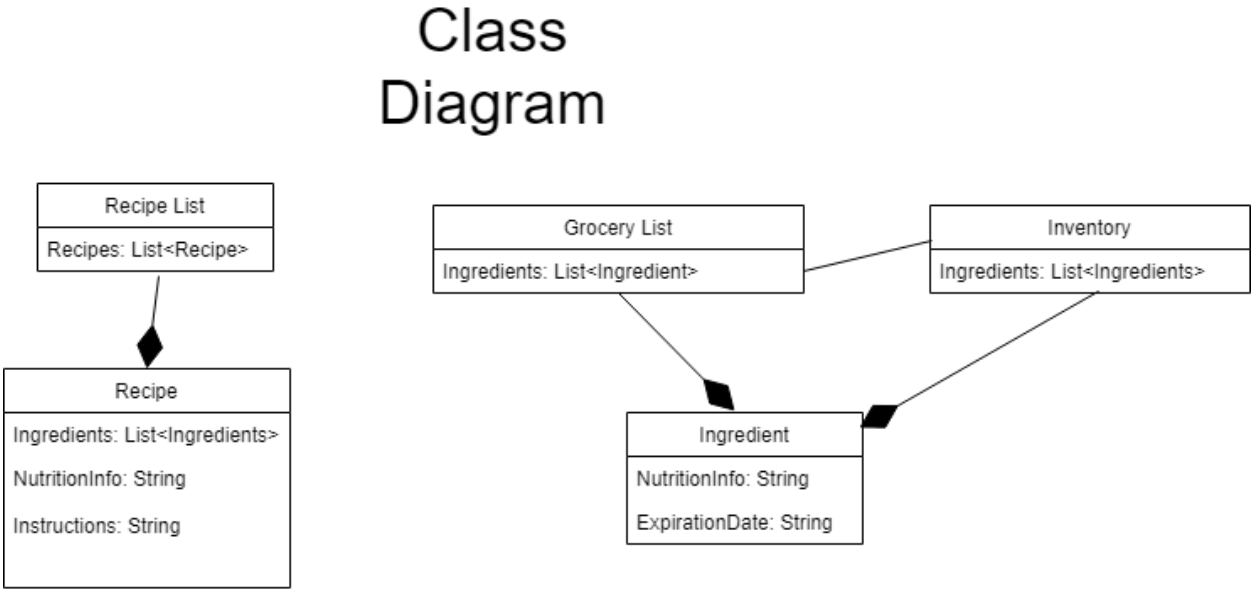
1. Goals and Accomplishments

- At least 5 additional user stories on issue boards (Done)
- Collection of a lot of data from your sources and storing in Database: (Done)
  - Automated scraping to retrieve food items and associated info (Done)
  - Many recipes and food items scraped - 500+ each (Done)
  - JSON for recipes and ingredients parsed out so queries are easier (Done)
- Creating a dynamic website with many pages hosted on GCP: (Done)
  - Conversion from static to dynamic pages on GCP (Done)
  - Login functionality works, but unable to save recipe lists, grocery lists, inventory, etc. because the front end and back end for that are still being worked on (Done)
  - Search and discovery functionality works (On backend), but hasn't been connected to frontend UI yet (Done)
- Connection of frontend and database via a backend using Java Servlets: (Done)
  - Servlets used to connect between JSP pages (Done)
  - Servlets used for search and discover functionality (Done)
  - Servlets used for storing Users (As in their Inventory, Grocery List, and Recipe Lists), ingredients, recipes, etc. in database (Done)
- Testing: (Done)
  - Most testing for this phase was done via code reviews, pair programming, and inspecting different aspects of our AppEngine to make sure it was doing what we wanted it to do (Done)
- Report refinement, updating as we go (Done)

Comments:

- To see our user stories on issue boards, check our Github. To see them in the report, check the "Design and Requirements" section below.
- To see the other major bullet points and their progress, check the relevant sections titles below.
- Our team spent a lot of time developing a front end using React.js and bootstrap, but struggled to be able to deploy it to our App Engine. We switched back to DHTML and JSP as a result, but may try to utilize that design in Phase 3.

2. Design and Requirements




---

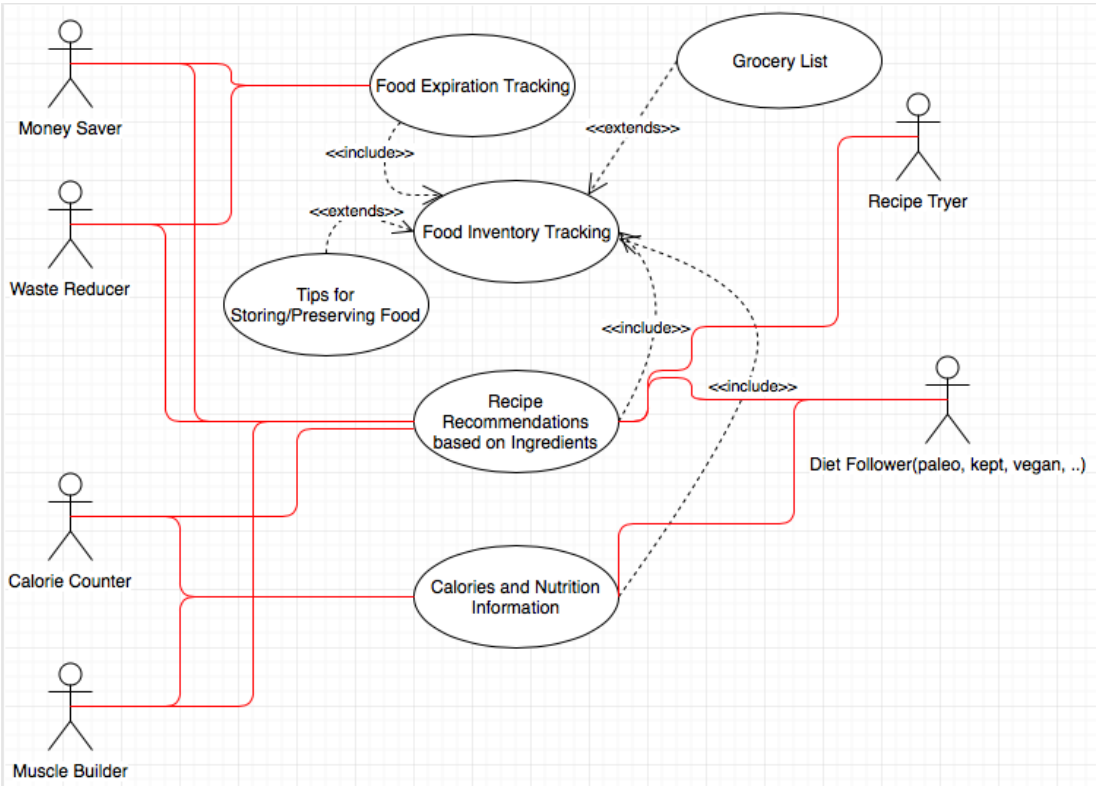
Phase I User Stories:

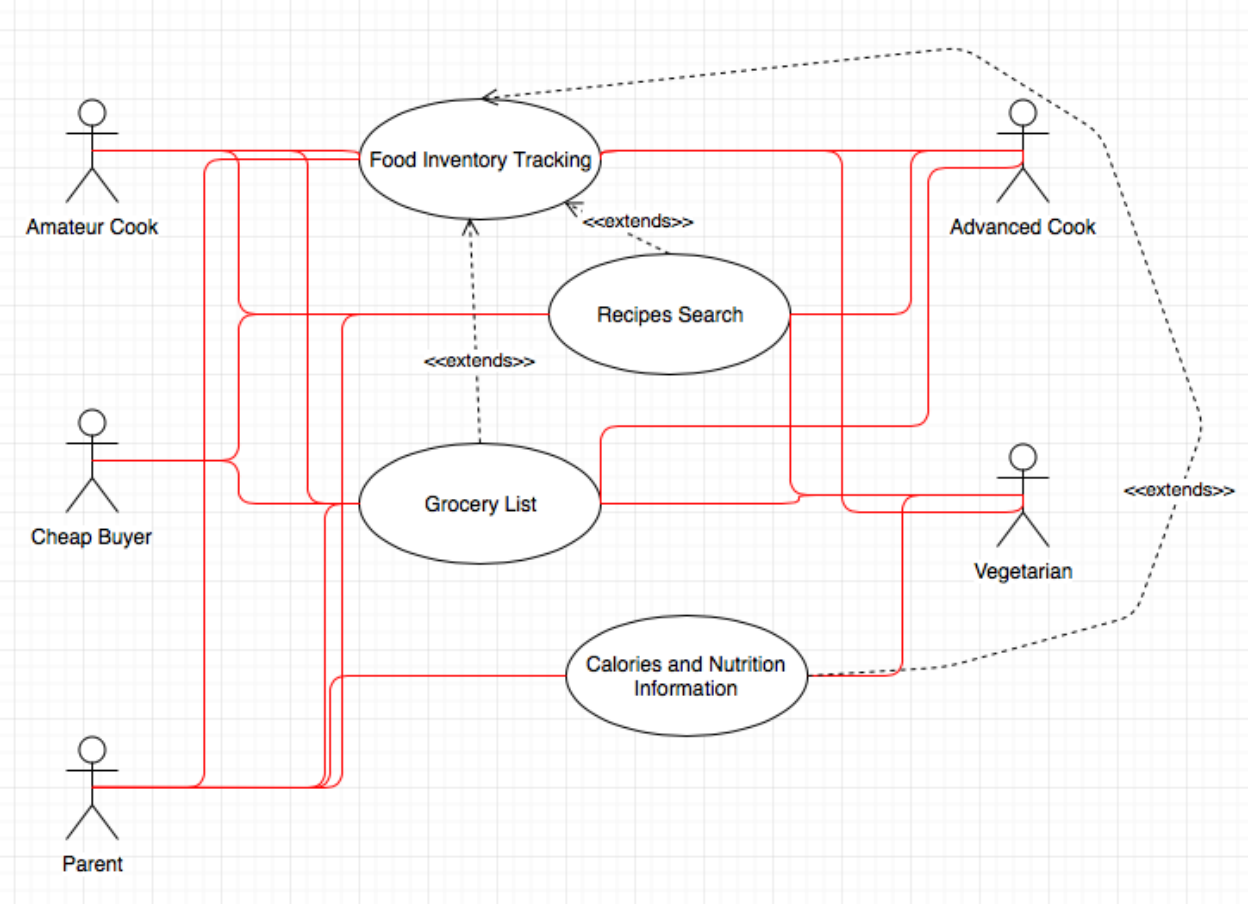
1. Save user's money.
    a. As a thrifty consumer, I would like to cut down on grocery shopping by efficiently using my groceries, so I am not re-purchasing items unnecessarily.
2. Reduce food waste.
    a. As a green consumer, I would like to reduce the amount of food I waste by tracking when items expire so that I may use all of the perishables that I purchase.
3. Calorie counter: lose weight, focusing on calories.
    a. As someone who would like to lead a healthier lifestyle, I would like to have quick access to the nutritional information of the food I buy.
4. Tries new recipes (For example, wanting cuisine type X).
    a. As an adventurous foodie, I would like to try new recipes.
5. Bulking: gain weight, focusing on maximizing volume of protein and healthy fats, getting more with less so one can fit it in the fridge/only need to get groceries once a week.
    a. As a bodybuilder, I would like to increase my muscle mass by consuming the right foods at a great enough volume.
6. Trying to follow a specific diet for health-related reasons
    a. As a someone who has dietary and health restrictions, I would like to have easy access to what foods I can eat, and what recipes I can make with them.
7. User who wants to obtain cooking skills
    a. As a someone who has just left home, I would like to become more independent by learning some easy recipes I can make for myself.
8. User who wants to purchase cheap items and find recipes for them
    a. As a someone who likes to purchase on-sale items and use coupons to shop cheaply, I would like to find recipes that will accomodate my purchases.
9. Vegetarian User

a. As a someone who has chosen to be a vegetarian for personal reasons, I would like to find recipes I can make with ingredients that don't include any meats, poultry, or fish.

10. Advanced cooking-level/chef user who wants to improve skills
    a. As a chef or someone who cooks very often and already have skills in the kitchen, i'd like to improve and advance my skills even further by exploring recipes that include unfamiliar ingredients.
11. Parent User
    a. As a parent or guardian of young children, I would like to find recipes that use kid-friendly ingredients so that I can cook healthy meals my children will want to eat.

Phase II User Diagrams:





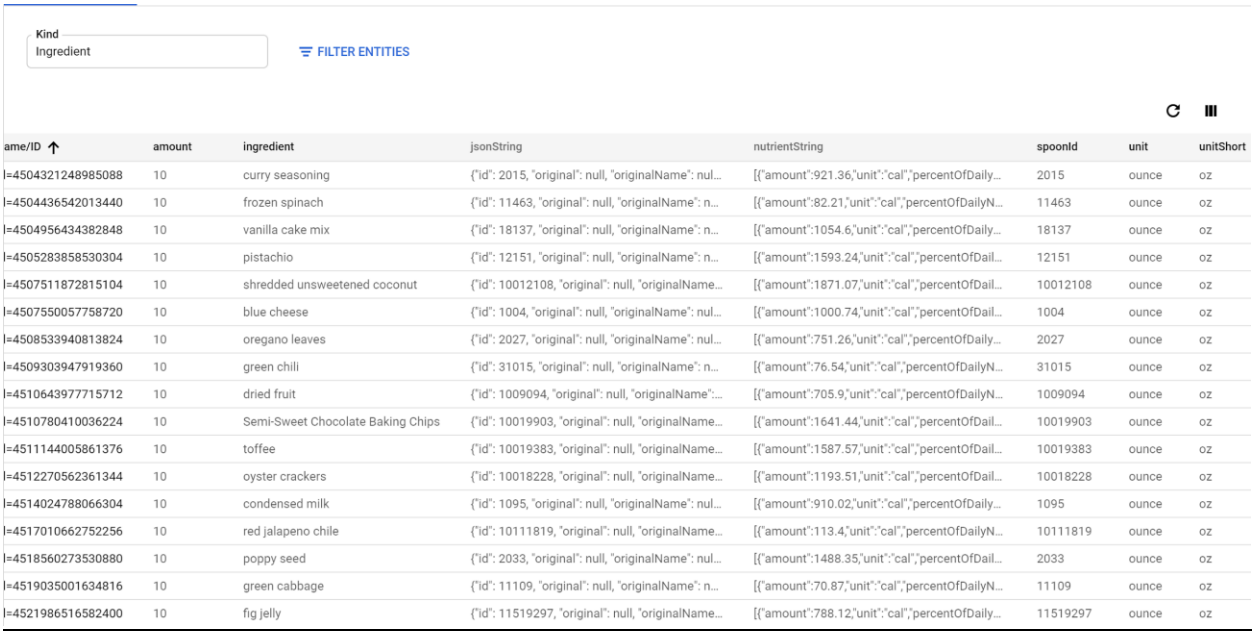

3. Scraping and Database (Description)

For phase two we continued scraping using the Python scripts that we developed in phase one. We did this incrementally using multiple accounts, as spoonacular only allowed 50 API calls per user per day. We continued to do this until we had over 900 ingredients and over 600 recipes. This JSON data was still stored in individual .json files. We created a cron job to wipe the GCP Datastore and construct Ingredient and recipe objects using the .json files. Initially, these objects simply consisted of a string made by converting the JSON data to a string. We later updated the Ingredient and Recipe constructors to include fields that will be useful data. For the Recipes, these include: strings for the recipe title and instructions, integers for the minutes to prepare and cook the recipe, and booleans for if the recipe is vegetarian, vegan, ketogenic, gluten free, and dairy free. For the Ingredients these include: a string for the ingredient name, along for the ingredient ID (provided by spoonacular), a float for the amount of the ingredient, and string for the units of that amount. We also included a string that includes information about the nutrients in the ingredient and how much of each nutrient. These fields are summarized in the following screenshots.

All of our code can be seen on the Github.

## 4. Scraping and Database (Screenshots)

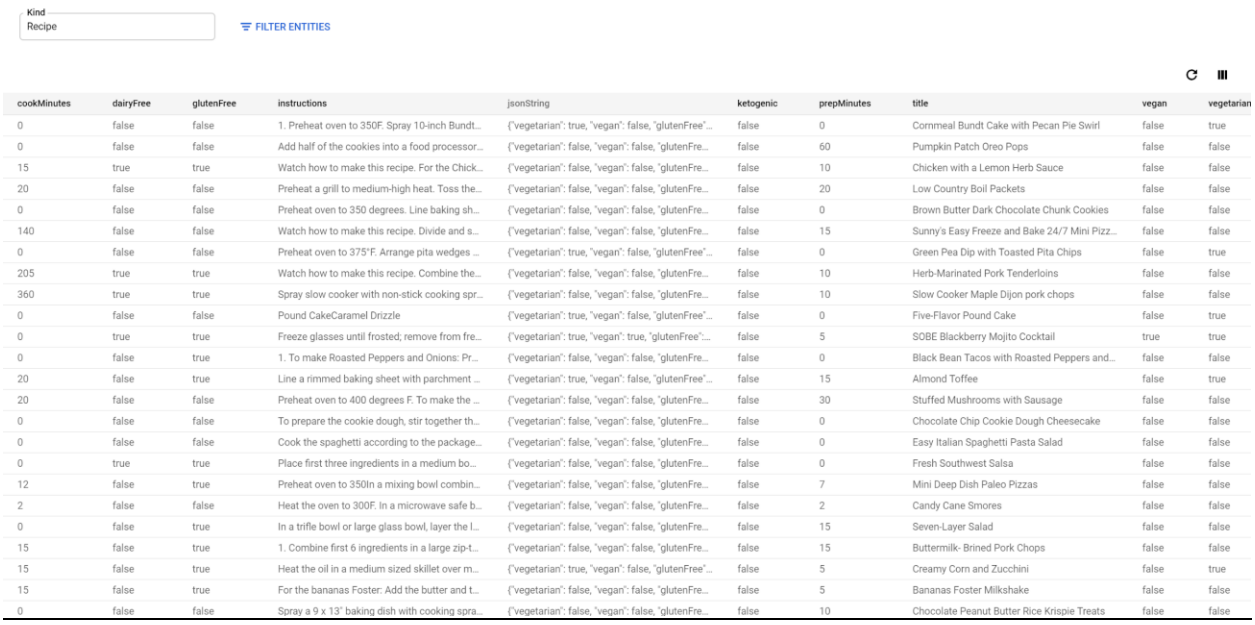
-Extensive List of Ingredients in GCP Datastore:



Kind: Ingredient — FILTER ENTITIES

| ame/ID | amount | ingredient | jsonString | nutrientString | spoonId | unit | unitShort |
|---|---|---|---|---|---|---|---|
| l=4504321248985088 | 10 | curry seasoning | {"id": 2015, "original": null, "originalName": nul... | [{"amount":921.36,"unit":"cal","percentOfDaily... | 2015 | ounce | oz |
| l=4504436542013440 | 10 | frozen spinach | {"id": 11463, "original": null, "originalName": n... | [{"amount":82.21,"unit":"cal","percentOfDailyN... | 11463 | ounce | oz |
| l=4504956434382848 | 10 | vanilla cake mix | {"id": 18137, "original": null, "originalName": n... | [{"amount":1054.6,"unit":"cal","percentOfDaily... | 18137 | ounce | oz |
| l=4505283858530304 | 10 | pistachio | {"id": 12151, "original": null, "originalName": n... | [{"amount":1593.24,"unit":"cal","percentOfDail... | 12151 | ounce | oz |
| l=4507511872815104 | 10 | shredded unsweetened coconut | {"id": 10012108, "original": null, "originalName... | [{"amount":1871.07,"unit":"cal","percentOfDail... | 10012108 | ounce | oz |
| l=4507550057758720 | 10 | blue cheese | {"id": 1004, "original": null, "originalName": nul... | [{"amount":1000.74,"unit":"cal","percentOfDail... | 1004 | ounce | oz |
| l=4508533940813824 | 10 | oregano leaves | {"id": 2027, "original": null, "originalName": nul... | [{"amount":751.26,"unit":"cal","percentOfDaily... | 2027 | ounce | oz |
| l=4509303947919360 | 10 | green chili | {"id": 31015, "original": null, "originalName": n... | [{"amount":76.54,"unit":"cal","percentOfDailyN... | 31015 | ounce | oz |
| l=4510643977715712 | 10 | dried fruit | {"id": 1009094, "original": null, "originalName":... | [{"amount":705.9,"unit":"cal","percentOfDailyN... | 1009094 | ounce | oz |
| l=4510780410036224 | 10 | Semi-Sweet Chocolate Baking Chips | {"id": 10019903, "original": null, "originalName... | [{"amount":1641.44,"unit":"cal","percentOfDail... | 10019903 | ounce | oz |
| l=4511144005861376 | 10 | toffee | {"id": 10019383, "original": null, "originalName... | [{"amount":1587.57,"unit":"cal","percentOfDail... | 10019383 | ounce | oz |
| l=4512270562361344 | 10 | oyster crackers | {"id": 10018228, "original": null, "originalName... | [{"amount":1193.51,"unit":"cal","percentOfDail... | 10018228 | ounce | oz |
| l=4514024788066304 | 10 | condensed milk | {"id": 1095, "original": null, "originalName": nul... | [{"amount":910.02,"unit":"cal","percentOfDaily... | 1095 | ounce | oz |
| l=4517010662752256 | 10 | red jalapeno chile | {"id": 10111819, "original": null, "originalName... | [{"amount":113.4,"unit":"cal","percentOfDailyN... | 10111819 | ounce | oz |
| l=4518560273530880 | 10 | poppy seed | {"id": 2033, "original": null, "originalName": nul... | [{"amount":1488.35,"unit":"cal","percentOfDail... | 2033 | ounce | oz |
| l=4519035001634816 | 10 | green cabbage | {"id": 11109, "original": null, "originalName": n... | [{"amount":70.87,"unit":"cal","percentOfDailyN... | 11109 | ounce | oz |
| l=4521986516582400 | 10 | fig jelly | {"id": 11519297, "original": null, "originalName... | [{"amount":788.12,"unit":"cal","percentOfDaily... | 11519297 | ounce | oz |

-Extensive List of Recipes in GCP Datastore



Kind: Recipe — FILTER ENTITIES

| cookMinutes | dairyFree | glutenFree | instructions | jsonString | ketogenic | prepMinutes | title | vegan | vegetarian |
|---|---|---|---|---|---|---|---|---|---|
| 0 | false | false | 1. Preheat oven to 350F. Spray 10-inch Bundt... | {"vegetarian": true, "vegan": false, "glutenFree"... | false | 0 | Cornmeal Bundt Cake with Pecan Pie Swirl | false | true |
| 0 | false | false | Add half of the cookies into a food processor... | {"vegetarian": false, "vegan": false, "glutenFre... | false | 60 | Pumpkin Patch Oreo Pops | false | false |
| 15 | true | true | Watch how to make this recipe. For the Chick... | {"vegetarian": false, "vegan": false, "glutenFre... | false | 10 | Chicken with a Lemon Herb Sauce | false | false |
| 20 | false | false | Preheat a grill to medium-high heat. Toss the... | {"vegetarian": false, "vegan": false, "glutenFre... | false | 20 | Low Country Boil Packets | false | false |
| 0 | false | false | Preheat oven to 350 degrees. Line baking sh... | {"vegetarian": false, "vegan": false, "glutenFre... | false | 0 | Brown Butter Dark Chocolate Chunk Cookies | false | false |
| 140 | false | false | Watch how to make this recipe. Divide and s... | {"vegetarian": false, "vegan": false, "glutenFre... | false | 15 | Sunny's Easy Freeze and Bake 24/7 Mini Pizz... | false | false |
| 0 | false | false | Preheat oven to 375°F. Arrange pita wedges ... | {"vegetarian": true, "vegan": false, "glutenFree"... | false | 0 | Green Pea Dip with Toasted Pita Chips | false | true |
| 205 | true | true | Watch how to make this recipe. Combine the... | {"vegetarian": false, "vegan": false, "glutenFre... | false | 10 | Herb-Marinated Pork Tenderloins | false | false |
| 360 | true | true | Spray slow cooker with non-stick cooking spr... | {"vegetarian": false, "vegan": false, "glutenFre... | false | 10 | Slow Cooker Maple Dijon pork chops | false | false |
| 0 | false | false | Pound CakeCaramel Drizzle | {"vegetarian": true, "vegan": false, "glutenFree"... | false | 0 | Five-Flavor Pound Cake | false | true |
| 0 | true | true | Freeze glasses until frosted; remove from fre... | {"vegetarian": true, "vegan": true, "glutenFree":... | false | 5 | SOBE Blackberry Mojito Cocktail | true | true |
| 0 | false | true | 1. To make Roasted Peppers and Onions: Pr... | {"vegetarian": false, "vegan": false, "glutenFre... | false | 0 | Black Bean Tacos with Roasted Peppers and... | false | false |
| 20 | false | true | Line a rimmed baking sheet with parchment ... | {"vegetarian": true, "vegan": false, "glutenFree"... | false | 15 | Almond Toffee | false | true |
| 20 | false | false | Preheat oven to 400 degrees F. To make the ... | {"vegetarian": false, "vegan": false, "glutenFre... | false | 30 | Stuffed Mushrooms with Sausage | false | false |
| 0 | false | false | To prepare the cookie dough, stir together th... | {"vegetarian": false, "vegan": false, "glutenFre... | false | 0 | Chocolate Chip Cookie Dough Cheesecake | false | false |
| 0 | false | false | Cook the spaghetti according to the package... | {"vegetarian": false, "vegan": false, "glutenFre... | false | 0 | Easy Italian Spaghetti Pasta Salad | false | false |
| 0 | true | true | Place first three ingredients in a medium bo... | {"vegetarian": false, "vegan": false, "glutenFre... | false | 0 | Fresh Southwest Salsa | false | false |
| 12 | false | true | Preheat oven to 350In a mixing bowl combin... | {"vegetarian": false, "vegan": false, "glutenFre... | false | 7 | Mini Deep Dish Paleo Pizzas | false | false |
| 2 | false | false | Heat the oven to 300F. In a microwave safe b... | {"vegetarian": false, "vegan": false, "glutenFre... | false | 2 | Candy Cane Smores | false | false |
| 0 | false | true | In a trifle bowl or large glass bowl, layer the l... | {"vegetarian": false, "vegan": false, "glutenFre... | false | 15 | Seven-Layer Salad | false | false |
| 15 | false | true | 1. Combine first 6 ingredients in a large zip-t... | {"vegetarian": false, "vegan": false, "glutenFre... | false | 15 | Buttermilk- Brined Pork Chops | false | false |
| 15 | false | true | Heat the oil in a medium sized skillet over m... | {"vegetarian": true, "vegan": false, "glutenFree"... | false | 5 | Creamy Corn and Zucchini | false | true |
| 15 | false | true | For the bananas Foster: Add the butter and t... | {"vegetarian": false, "vegan": false, "glutenFre... | false | 5 | Bananas Foster Milkshake | false | false |
| 0 | false | false | Spray a 9 x 13" baking dish with cooking spra... | {"vegetarian": false, "vegan": false, "glutenFre... | false | 10 | Chocolate Peanut Butter Rice Krispie Treats | false | false |

## 5. Screens, Features and Functionality (Description)

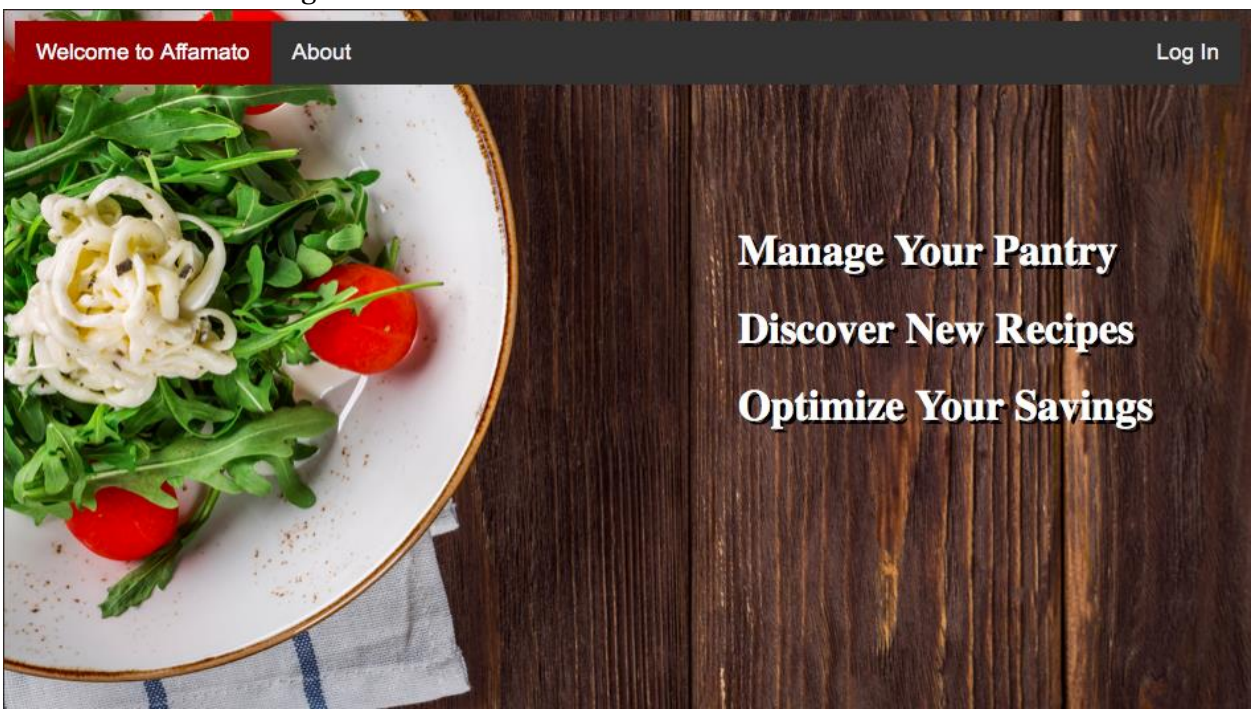
For phase two, we have implemented a dynamic website (In DHTML), which will have the welcome and about pages, where you can log in/out with your gmail account, dashboard page which becomes available once a user is logged in, and selections from the dashboard page which will display grocery lists, food inventory, recipes list, and search bar . The pages are not fully functional; the accounts are not implemented, and interaction with food/recipes is not

functional. Users won't be able to interact with food items or recipes until next phase, as most of our work was on advancing the framework for the website and developing a wider selection within the database. Please go to the link to see our website.

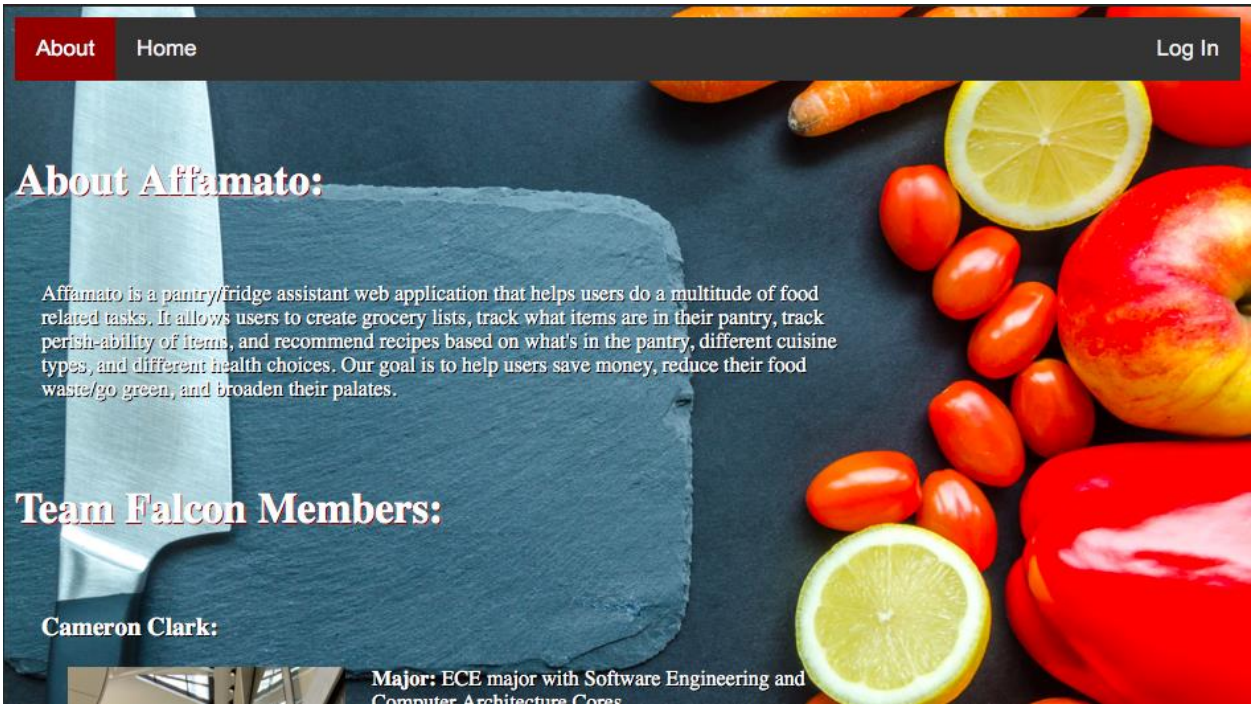
We also plan on further developing the dynamic user interface (with reactjs and bootstrap) which is demonstrated below.

All of our code can be seen on the Github.

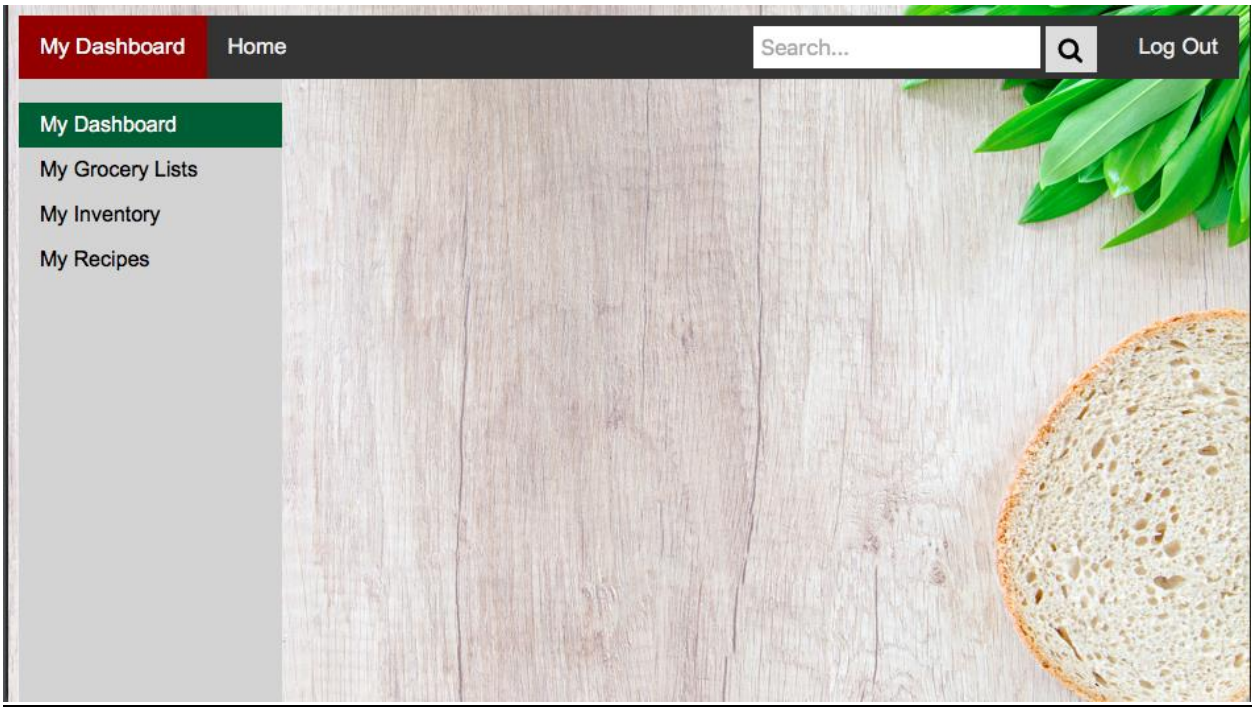
6. Screens, Features and Functionality (Screenshots)
-Affamato Welcome Page:



-Affamato About Page




-Affamato Dashboard



7. Tools, Software, Frameworks Used

In this phase, we used much of the same software tools that we used in last phase. We continued to run the same Python scripts, and didn't develop any new code for scraping. We also continued to develop with DHTML for the frontend and Java for the backend, which uses the Google objectify library and Datastore to store the data. We changed gears from using Java Spring in the backend to using Java Servlets because our team was more familiar with how to implement it with the rest of our project. We also continued to develop using React, but were unable to successfully connect our react frontend with our backend, but we are looking to make that connection in phase 3.

8. Testing

During Phase 2, our team tested the database by inspection, searching through the GCP datastore to look for corrupt data; searching for "null" was the most useful, as it alerted us to fields that were not being populated in some cases. We visually and manually tested our web application to ensure the basic functionality that we accomplished in phase 2. We also did pair programming and code reviews throughout the phase to ensure the quality of our code. We will be moving towards unit testing during phase 3 with JUnit, Mocha, and Selenium.

9. Updated Phase Goals

Phase I - due March 1

- At least 5 user stories on issue boards (Done)
- Beginnings of scraping and database: (Done)
    - Start scraping (Wrote own python scripts) (Done)
    - Food item data scraping functional - over 150 ingredients scraped and saved as JSON files (spoonacular/yummly) (Done)
    - Recipe scraping functional - over 20 recipes scraped and saved as JSON files (spoonacular/yummly) (Done)
    - Basic database running on GCP Datastore (NoSQL) with the information above stored. JSON files saved as Recipe/Ingredient Entities defined in Java (Done)
- Barebones UI built from DHTML, reactjs, and bootstrap: (Done)
    - Static site with at least 5 pages hosted on GCP (Done)
    - Basic pages for: Welcome/login page, grocery list page, my recipes page, my inventory page, about page (Done)
        - NOTE: Will not be fully functional: Users won't be able to interact with food items, recipes, or grocery lists yet, nor will accounts be functioning
    - Stats derived on the About page, dynamically, from GitHub (Done)
    - Progress made on dynamic versions of the page with reactjs and bootstrap (Done)
- Other things:
    - Team needs to learn about basics of APIs, GCP, SQL, NoSQL, Postman, DHTML, reactjs, bootstrap, Selenium, Mocha, Slack and Github (Done)
    - URL from a hostname provider (Namecheap) (Done)
    - Report, updating as we go. (Done)

Phase II - due March 29

- At least 5 additional user stories on issue boards (Done)
- Collection of a lot of data from your sources and storing in Database: (Done)
    - Automated scraping to retrieve food items and associated info (Done)
    - Many recipes and food items scraped - 500+ each (Done)
    - JSON for recipes and ingredients parsed out so queries are easier (Done)
- Creating a dynamic website with many pages hosted on GCP: (Done)
    - Conversion from static to dynamic pages on GCP (Done)
    - Login functionality works, but unable to save recipe lists, grocery lists, inventory, etc. because the front end and back end for that are still being worked on (Done)
    - Search and discovery functionality works (On backend), but hasn't been connected to frontend UI yet (Done)
- Connection of frontend and database via a backend using Java Servlets: (Done)
    - Servlets used to connect between JSP pages (Done)
    - Servlets used for search and discover functionality (Done)
    - Servlets used for storing Users (As in their Inventory, Grocery List, and Recipe Lists), ingredients, recipes, etc. in database (Done)
- Testing: (Done)
    - Most testing for this phase was done via code reviews, pair programming, and inspecting different aspects of our AppEngine to make sure it was doing what we wanted it to do (Done)
- Report refinement, updating as we go (Done)

<u>Phase III - due April 9</u>

- At least 5 more user stories –put them on your issue boards
- Finish data collection- all recipes, food, and relevant info is scraped and stored in the database
- All objects have functional instances for users: food items, pantry, recipes, etc.
- Login page is fully functional but no account creation stored by Affamato (sign in with google -> new account creation would be like making a new google acct. Still holds their recipe list, grocery list, and inventory, etc.)
- Login functionality works, and user can fully use inventory, recipe list, grocery lists via front end
- Add front end pages for search results, discover page
- Testing (Refine and expand):
    - Most unit tests for JS/JSP written with Mocha
    - Most GUI tests using Selenium
    - Most Java Servlet tests using JUnit
- Fully functional search implementation
- Fully functional account creation (username+password) in the database
- Refine your dynamic site with many pages hosted on GCP
    - All pages are in their final form in both appearance and functionality
    - Search algorithm works for complex scenarios
- Add to and refine the technical report, updating as we go

<u>Phase IV - due April 30</u>

- Development is IDEALLY completed in phase III
- Refactor, apply design patterns
- Finish the final project report
- Catch up with something if stuff gets behind
- Expand on calorie/exercise tracking in accordance with our long term vision for a lifestyle app if we have lots of extra time
- Create a presentation and put it on GitHub as a pdf (not required until you present)