Group: 014-01

Title: CalTracker+

Who:

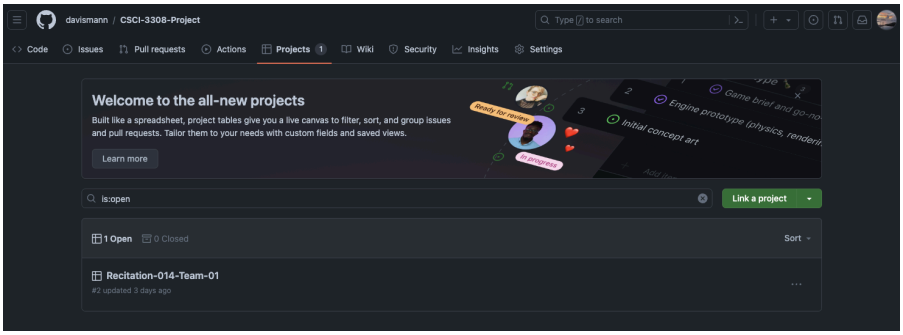
- Davis Mann
- Alex Stelzer
- Andrew Cook
- Sam Engelbert
- Eric Fisher
- Idir Dahmouh

Project Description:

In the world of fitness, many people struggle in finding a place to start with their nutrition goals. We wanted to take the stress out of nutrition. The release of CalTracker+ does just that. CalTracker+ provides the user with a calorie goal dependent on their body composition and goals. This takes the guesswork out of finding the right amount of daily calories that will allow for constant progression towards those goals. Once the user knows how many calories they need to eat, they can search the recipes page based on desired search filters and the user will be provided a link to recipes that match the filters. This allows the user to easily find delicious meals to eat while simultaneously pushing towards their goal. The recipes are provided with the number of calories in the meal which can be entered into the tracking page where food history and calorie count can be viewed. Each day the user can reset their foods eaten and their calorie count for a fresh day of tracking. Over time, the users may be different weights, or if the user is young they may have grown. The users can change these attributes in their profile. When this page is updated, the calorie goal will also be updated to ensure progress is being made efficiently.

Project Tracker - GitHub project board:

- https://github.com/users/davismann/projects/2



Video:

https://youtu.be/ezALJoRG8-

VCS:

- https://github.com/davismann/CSCI-3308-Project.git

Contributions:

Davis Mann: I worked on implementing the login and registration page as well as the tracker page. I made sure to throw error messages relating to the login page or registration page requirements. On the tracker page, I helped create the tracker GET and POST routes that allow the session variables, and imputed variables, to be properly saved into the log or total calories. Additionally, I went over the entire software and made sure everything was working and the style was the same throughout.

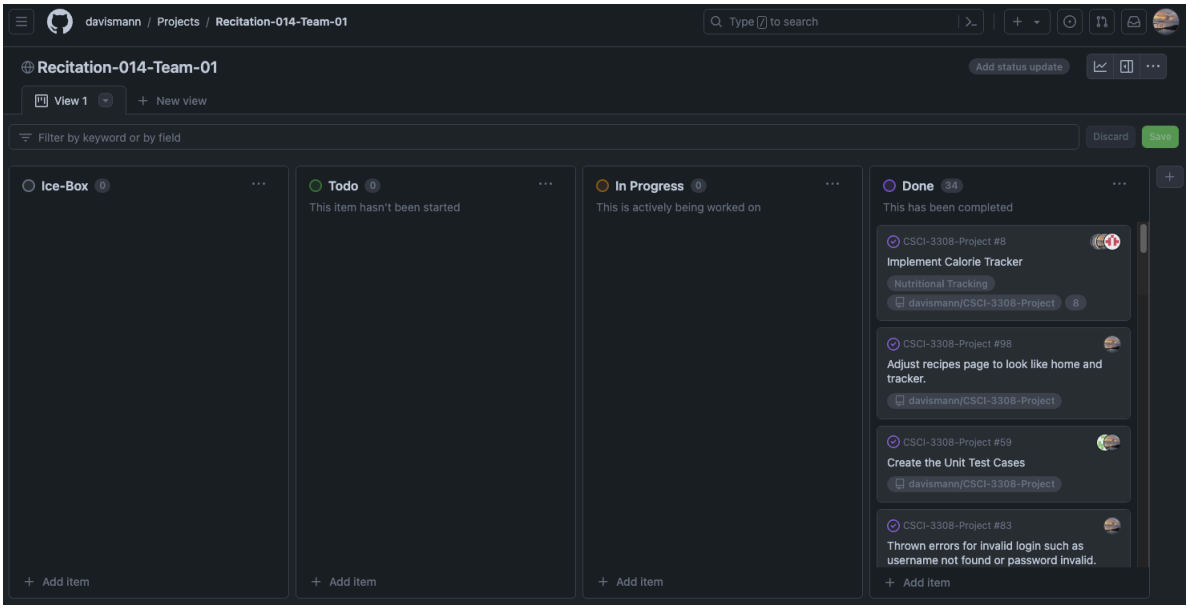
Alex Stelzer: I worked on implementing the SQL as well as the tracker page. I helped out with the registration, login and home page in making sure the registration was saved. I created the base layout for the tracker page to be edited and created as well as helping out with the GET and POST routes so that our tables would properly work and display information. I did a majority of the lab work for lab 11 through lab 13 in creating unit test cases and the deployment environment.

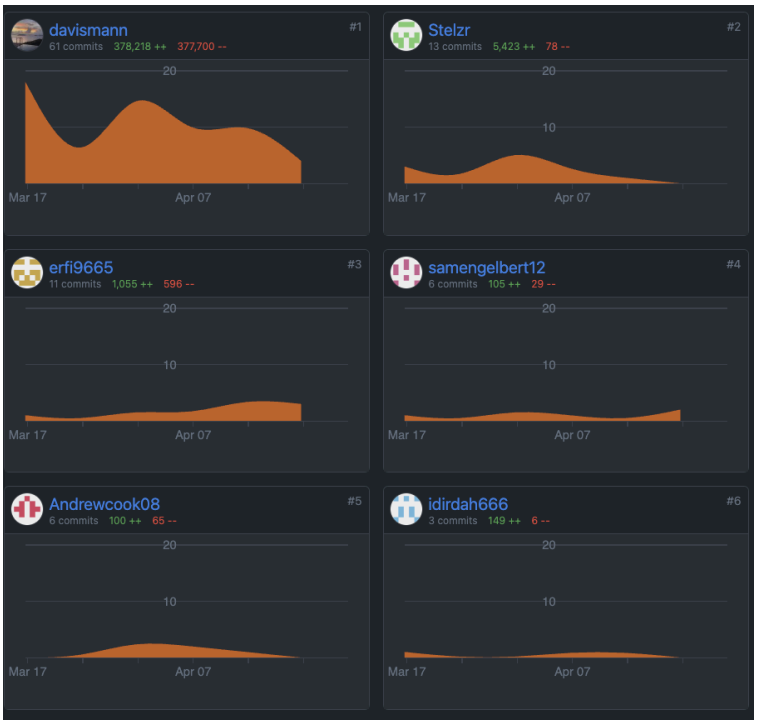
Sam Engelbert: I helped work on both the registration page, home page, and the edit page. On the register page, I added a feature to the POST route which calculates the initial calorie requirement for the user based on their inputs. On the home page, I helped to enable the functionality to allow for the user's calorie data and signup data to be shown on screen after successful login. On the edit page, I made it so that the user's initial signup options show as the default values in each field.

Eric Fisher: I created the recipes search page, along with all the functionality for this. In this, I interfaced with the Edamam Recipe Search API in order to grab filters from the user and search for recipes based on those. These filters included a calorie range along with a keyword, meal type, and ranges for protein, fat, and carbs. I created the .hbs page, GET and POST requests, and script necessary in order to make this feature functional.

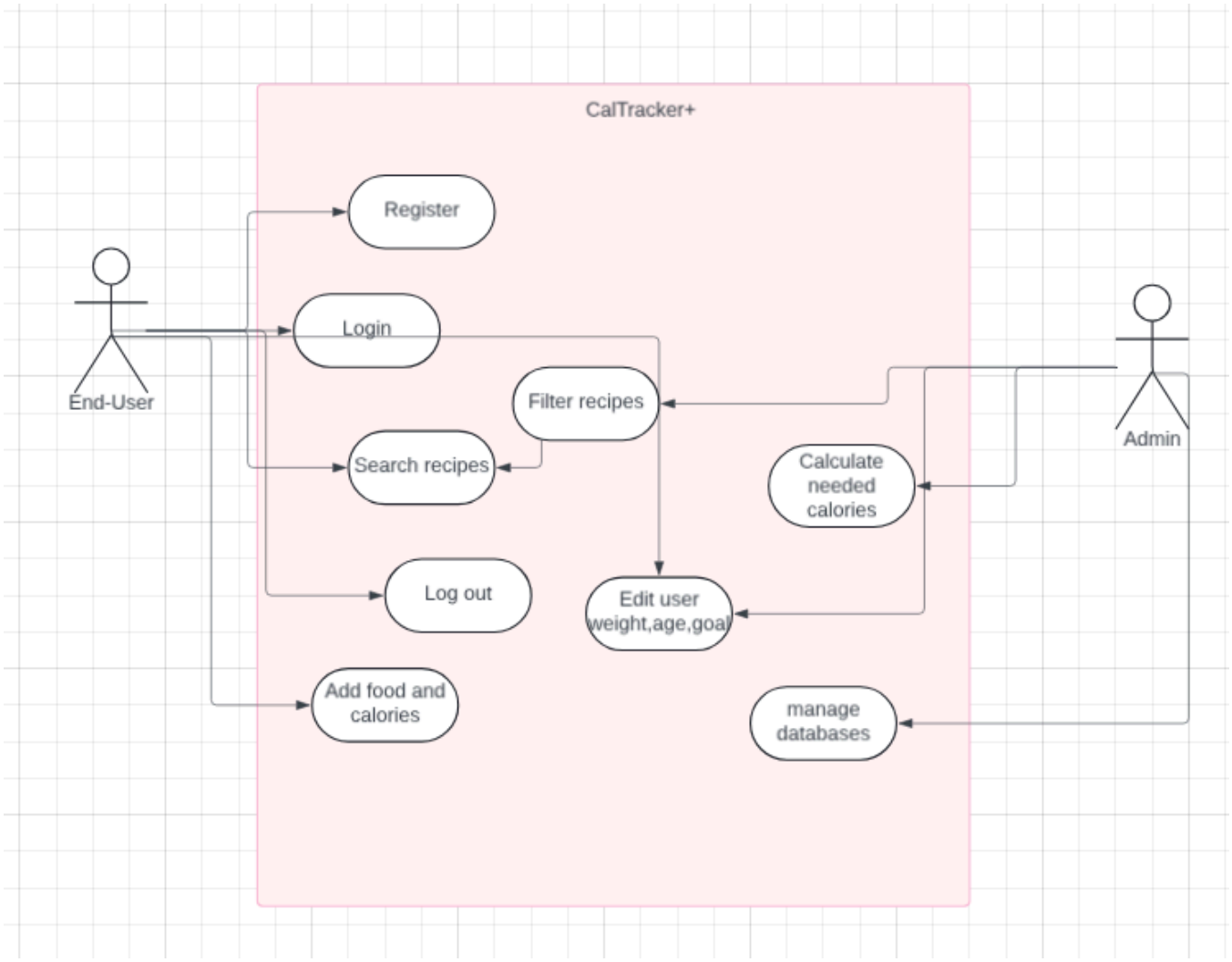
Idir Dahmouh: In this project, I focused on implementing core functionalities including the initial registration and login pages. I integrated the edit function, allowing users to update their details after registering. A significant part of my work involved debugging, where I identified a critical oversight where gender input, crucial for accurate calorie calculations, was not being requested. Additionally, I corrected all spelling errors and enhanced the application by adding units for clarity in data presentation.

Andrew Cook: In this project, I focused mostly on the tracker feature. I worked on the html css and handlebars code for the actual tracker form making sure the route calls were in sync with the routes Davis wrote, and the data was appearing as it should. For example, the calorie total was not showing up so I debugged and fixed it. As Idir mentioned he realized shortly before production time that there was no input for gender on the register page which made the calorie goal inaccurate. I fixed this issue using javascript, html and css by adding the missing fields and updating session variables as needed.

- 

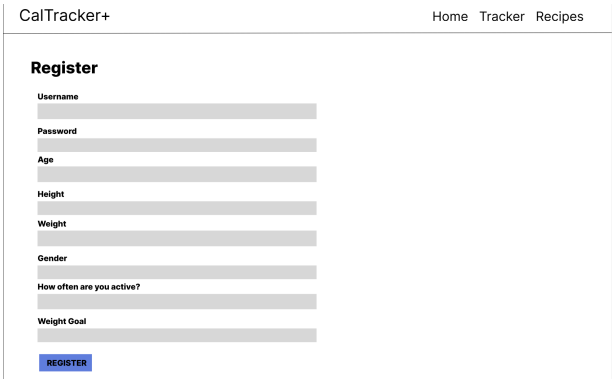
- 

Use Case Diagram:

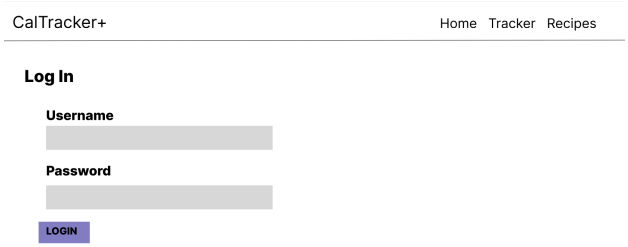

CalTracker+
Register
Login
Filter recipes
Search recipes
Log out
Add food and calories
Edit user weight,age,goal
Calculate needed calories
manage databases
End-User
Admin

Wireframes:

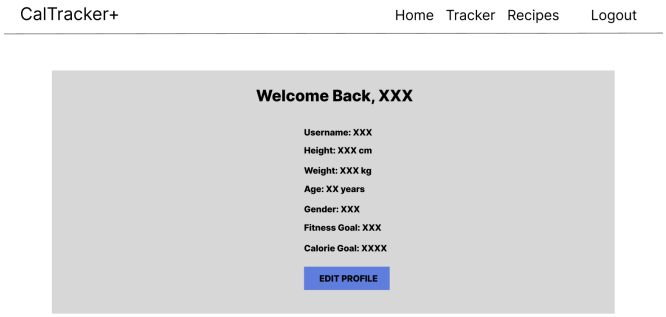
Register



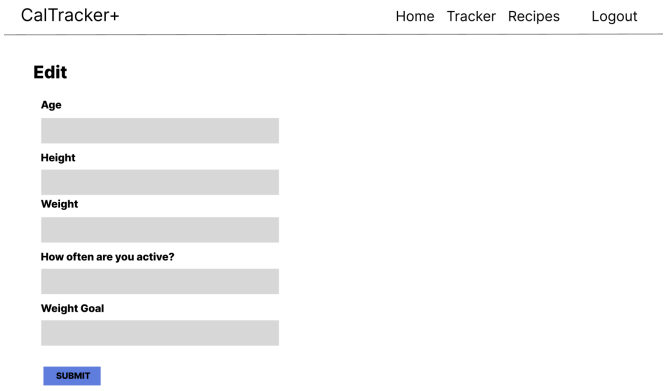
Login



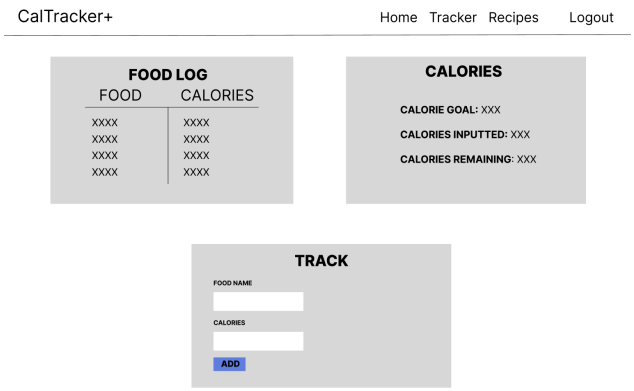
Home



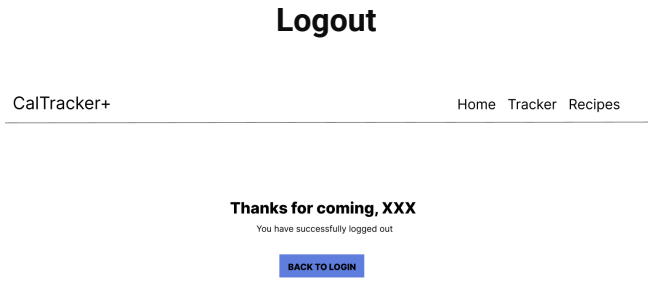
Edit



Tracker



Recipes



Test results:

Feature 1 Results:

- The user was able to effectively register using the valid credentials. All of the variables imputed were within the allowed range. When the user submitted the registration form they were effectively promoted to the login page. When doing this test, we noticed that the tester we established accidentally typed 670 cm into the height which invoked an error message to them.

Feature 2 Results:

- The testing client was able to login with the proved account they created from feature 1 testing. They imputed two fields: username and login, and were able to effectively load into the home page. Additionally, the home page showed all the valid credentials they entered from when they registered. We also had the test user try and login with a different password or invalid username. This provoked an error message that we established in the *index.js* routes.

Feature 3 Results:

- The user now was prompted to test the track method which inputs a specific food and calories into the log and total calorie goal. Upon creating the test case, we were unsure what fields we would need for the tracking mechanism. The test case prompted serving size, which we ultimately decided to leave out. So we instead prompted the testing user

to input the food name: “Apple” and the calories: “95.” The input worked and the food name and calories were stored in the log, and the total calories was increased by 95.

Deployment:

You can access the app from the following link:

http://recitation-14-team-01.eastus.cloudapp.azure.com:3000/login

The app was deployed through Microsoft Azure. You first have to set up your Linux Virtual Machines and your SSH public key. Next, I created a DNS name for the machine called recitation-14-team-01 within the resources of the virtual machine. Then I created an inbounding port rule with the destination port range as 3000. This is where you can actually connect to the virtual machine. In order to do this I had to move my .pem file into the .ssh file, and then use chmod 400 to make sure I have read-only access. Then you connect as azureuser@csci3308. Finally, you install docker and docker compose as the azure user, and run your web solution by cloning the HTTPS URL and running sudo docker compose up -d.