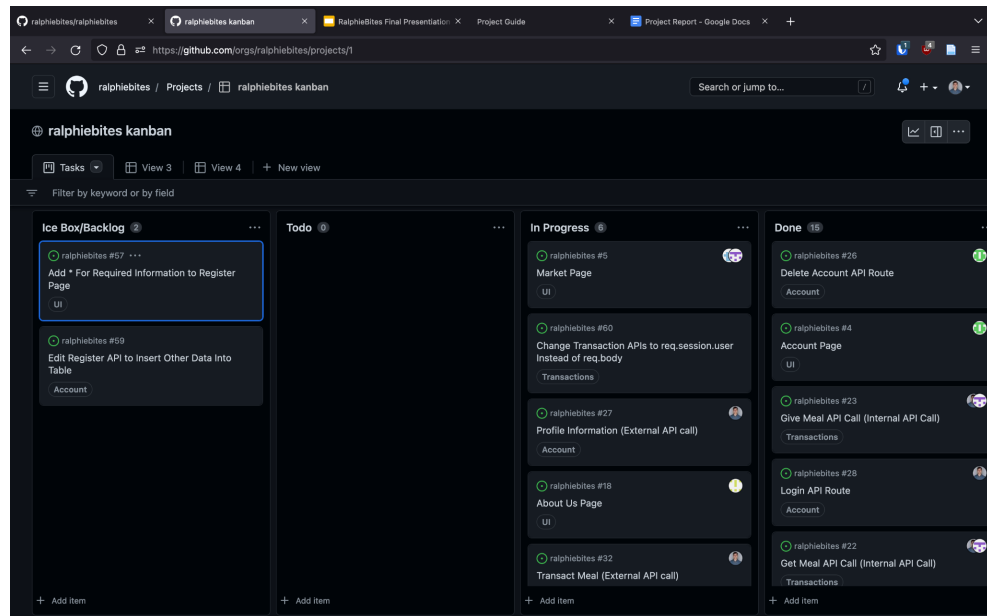


## Project Report | Team 015-06

- **Title:** RalphieBites
- **Who:** Annie Cai, Sanjana Kumpati, Vraj Patel, Ian Draves, Kavin Ramesh, Sriyans Pattanaik
- **Project Description**
  - RalphieBites is a web app that allows CU students to donate meals from their meal plan to students in need of a meal. After signing up for the service, students are able to easily donate as many meals as are currently available in their student account with a simple click of a button. Similarly, students who are in need of a meal can simply sign up and, within a single click, request a meal. Automatically, RalphieBites connects students looking to donate a meal with students requesting a meal and completes the transaction as soon as a pair is found. We believe RalphieBites is an excellent compromise to the otherwise hundreds of unused meal swipes that accumulate each week. Indeed, there is a significant surplus of meal swipes since every freshman is required to have a meal plan and many fail to use all their meals in a given week. As a team, we have all personally experienced this problem, having left over meal swipes at the end of many weeks during our freshman year. Ultimately, RalphieBites provides a technical solution to a practical problem, alleviating wasted meal swipes and aiding food insecure students.
- **Project Tracker - GitHub project board:**
  - Project Board: <https://github.com/orgs/ralphiebites/projects/1>



- **Video:**  
<https://drive.google.com/file/d/11eM4HLIVWFSeZvYrediRelK3XQu5A1F5/view?usp=sharing>
- **VCS**
  - Source Code: <https://github.com/ralphiebites/ralphiebites>

- Test Cases:  
[https://github.com/ralphiebites/ralphiebites/blob/main/milestone\\_submissions/Lab%2011.pdf](https://github.com/ralphiebites/ralphiebites/blob/main/milestone_submissions/Lab%2011.pdf)

**First Registration Test Case: {**

**First Name: Sriyans**  
**Last Name: Pattanaik**  
**Email: srpa3716@colorado.edu**  
**Username: sripat**  
**Password: January\$26**  
**}**

*First Test Case should be valid*

- **Second Registration Test Case: {**

**First Name: Kav1n**  
**Last Name: Ramesh**  
**Email: [kara9783@coforado.edu](mailto:kara9783@coforado.edu)**  
**Username: wr0ng\_us3rn4me**  
**Password: H4Ppy-B1RTHday**  
**}**

*Second Test Case should fail not colorado.edu email*

- **Third Registration Test Case: {**

**First Name:**  
**Last Name:**  
**Email:**  
**Username:**  
**Password:**  
**}**

*Third Test Case is empty case (should fail)*

- **First Login Test Case: {**

**Username: sripat**  
**Password: January\$26**  
**}**

*First Test Case is valid (should pass)*

- **Second Login Test Case: {**

**Username: wr0ng\_us3rn4me**  
**Password: H4Ppy-B1RTHday**  
**}**

*Second Test Case has incorrect username (should fail)*

- **Third Login Test Case: {**

**Username:**  
**Password:**  
**}**

*Second Test Case has empty case (should fail)*

- **First Access Test Case:**

```
{  
  User logged in: true  
  URL: localhost:3000/market  
}
```

*First Test Case is correct.*

- **Second Access Test Case:**

```
{  
  User logged in: false  
  URL: localhost:3000/market  
}
```

*Second Test Case is incorrect.*

- Video demo:

<https://drive.google.com/file/d/11eM4HLIVWFSeZvYrediRelK3XQu5A1F5/view?usp=sharing>

- README.md in GitHub:

<https://github.com/ralphiebites/ralphiebites/blob/main/README.md>

- Project documentation – In README

- Project Board: <https://github.com/orgs/ralphiebites/projects/1>

- **Contributions:**

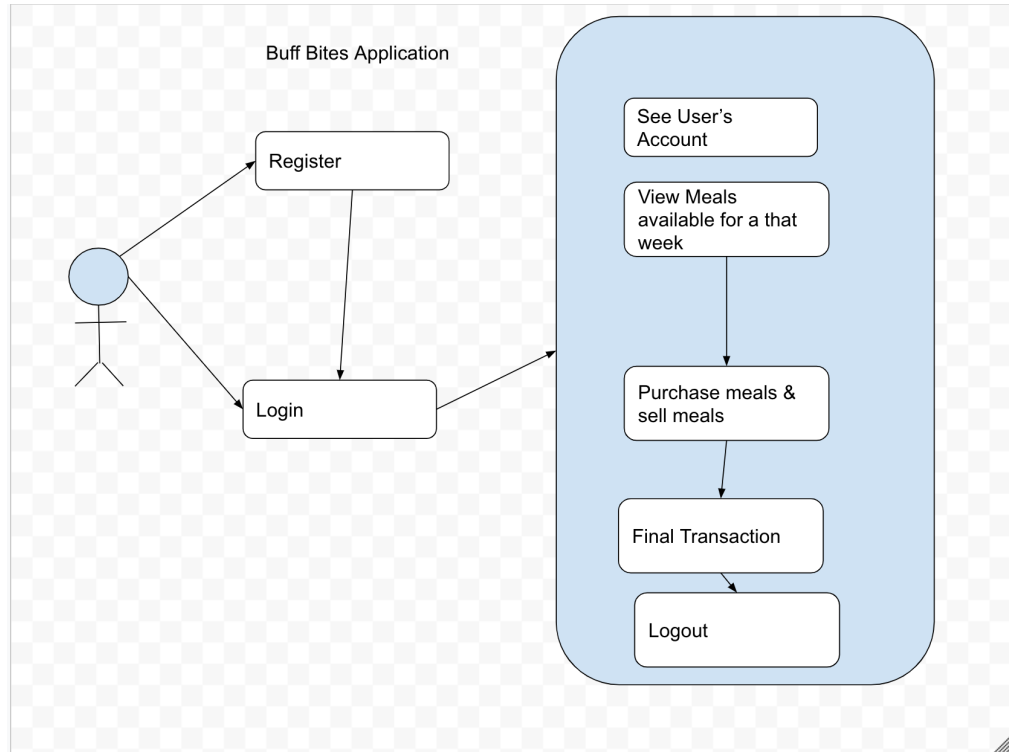
- Annie

- I worked on the front end and the back end of the transactions section. I designed the layout of the market page and figured out the additional information needed. I also designed the transactions APIs, /claim and /donate, for the transactions backend along with Sanjana and Ian. The transaction APIs get the active user's username before using the donateMeal(), claimMeal(), and getDonations() functions to process the meal request and/or donations.

- Sanjana

- I worked on the front end and back end using VSC. I designed the registration page to have the user enter first name, last name, @colorado.edu email, username, password, and confirm password. Login page I had the user enter username and password. For the back end I wrote an api call for registration in which I checked if password and confirm password were the same and if the email was a @colorado.edu. If not there will be an error message. Another contribution was the market page api call for get meals. Allowing the user to add a meal.

- Vraj
  - I worked on the front end and back end using VSC. I designed the account page which takes the users information entered in registration and displays current information (username, first & last name, meals). At the bottom of their information, I added a delete button. This connects to the back-end side and deletes the user information from the database. Therefore, they can no longer login with their previous credentials.
- Ian
  - I worked primarily on the backend, writing helper functions and API routes to make the transaction functionality work properly, setting up the initial server architecture, and integrating meal tracking on the “Market” and “Account” pages. Alongside Sanjana and Annie, I created the “/donate” and “/claim” API routes. I personally implemented all of the helper functions which those API routes use, including the “claimMeal()”, “donateMeal()”, and “getDonations()” functions. Aside from that, I updated the user interface for every page on the website once everyone had pushed their work to use Bootstrap with consistent theming across the site.
- Kevin
  - I worked on the front end and designed the about us/home page. This page uses bootstrap to design and work the carousel object which showcases our group and the members. Each “slide” has its own image and description which were formatted properly to be consistent on each slide regardless of the size of the image. I also worked a little on the styling of the page and making it look simple yet pleasing.
- Sriyans
  - I worked on the front end primarily. I designed the help page, which essentially just shows the user how to use basic features in our application. Most help web pages on websites use collapsibles to portray an FAQ system. The user can go there to see if they have any issues with accessing things on the website. I used bootstrap for the collapsibles on the help page. I added a few FAQs and instructions for the user to be able to see how to access their meals, transfer their meals to another user, and how to access settings, etc.
- **Use Case Diagram**



- **Test results:** In Lab 11, you created a Test Plan. You need to include the test results and observations in the project report. Refer to [this](#) for more information

Some test cases we wanted to check:

**First Registration Test Case: {**

**First Name: Sriyans**  
**Last Name: Pattanaik**  
**Email: srpa3716@colorado.edu**  
**Username: sripat**  
**Password: January\$26**  
**}**

*First Test Case should be valid*

For this first test result we got a successful registration. It directly went into the login page with no errors.

- **Second Registration Test Case: {**

**First Name: Kav1n**  
**Last Name: Ramesh**  
**Email: [kara9783@coforado.edu](#)**  
**Username: wr0ng\_us3rn4me**  
**Password: H4Ppy-B1RTHday**  
**}**

*Second Test Case should fail not colorado.edu email*

For this second test result we got an unsuccessful registration. It is promoted with an error message notifying the user that they will have to use a colorado.edu email.

- **Third Registration Test Case: {**

**First Name:**

**Last Name:**

**Email:**

**Username:**

**Password:**

**}**

*Third Test Case is empty case (should fail)*

The third registration test failed since nothing was entered. It notifies the user that these need to be filled out.

- **First Login Test Case: {**

**Username: sripat**

**Password: January\$26**

**}**

*First Test Case is valid (should pass)*

The first login test case passed since the correct username and password was entered.

- **Second Login Test Case: {**

**Username: wr0ng\_us3rn4me**

**Password: H4Ppy-B1RTHday**

**}**

*Second Test Case has incorrect username (should fail)*

The second login test failed since there was an incorrect username. It notified the user that the username or password is incorrect.

- **Third Login Test Case: {**

**Username:**

**Password:**

**}**

*Second Test Case has empty case (should fail)*

The third login test failed since there was an incorrect username and password since nothing was entered. It notifies the user that these need to be filled out.

All transaction test cases in the link above are not relevant since we changed our design. We added some more test cases on access.

- **First Access Test Case:**  
{  
  **User logged in: true**  
  **URL: localhost:3000/market**  
}

*First Test Case is correct.*

The first login test successful since there is a correct path to go to the market.

- **Second Access Test Case:**  
{  
  **User logged in: false**  
  **URL: localhost:3000/market**  
}

*Second Test Case is incorrect.*

The second login test was unsuccessful since there is not a correct path to go to the market.

- **Deployment environment:** <http://csci3308.int.colorado.edu:3000>