Cheap Eats Project Report

Team Member Name	Github Username
Abdullahi Husein	Abdullahihus
Alec Wang	wangalec
Christopher Sponza	chsp7258
Mara Backsen	meb321
Nathnael Tewelde	NatTew

Project Description

Cheap Eats is a web application designed for those looking for places to eat in their local area with the best bang for their buck. It provides a platform for users to discover, review, and share their favorite local restaurants. Using personalized features like wishlists and user search, Cheap Eats creates a community of food lovers who want to maximize their food quality to cost ratio.

Key Features:

- User Authentication: Users can register for an account and securely log in to access personalized features.
- Restaurant Search: Users can search for local restaurants and view their average rating based on quality and price.
- Restaurant Reviews: Users can contribute to the community by adding their own reviews to places they've eaten.

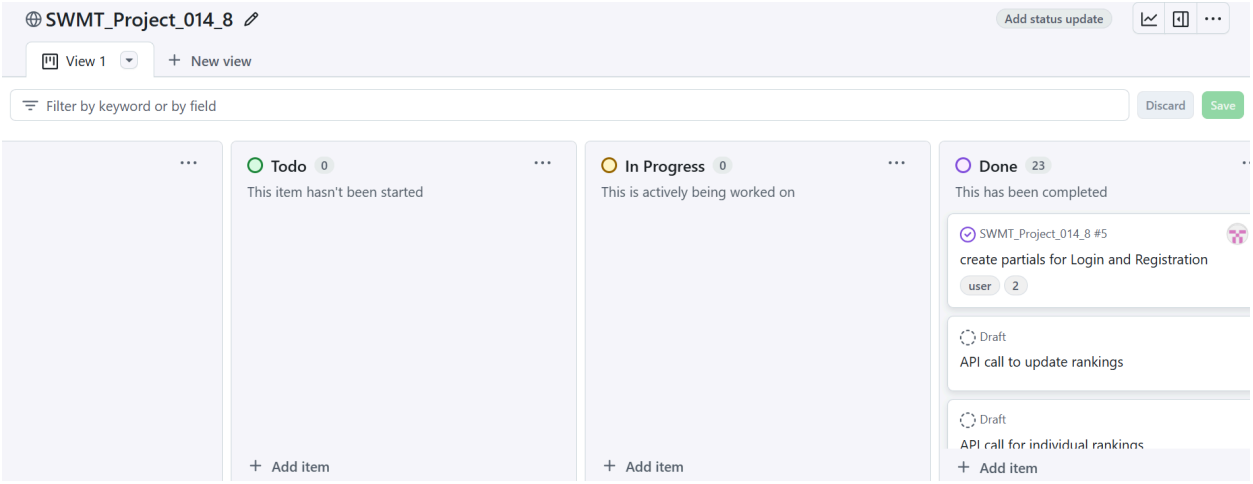
- Wishlist: Users can create a wishlist of restaurants they hope to eat at in the future.
- User Interaction: Users can search for other members to see their reviews and restaurants on their wishlist.

Overall Benefit:

Cheap Eats empowers users to explore local budget-friendly dining options, share their reviews, and track restaurants they would like to visit all while interacting with other people in their area.

Project Tracker:

https://github.com/users/chsp7258/projects/1/views/1



Demonstration Video:

https://drive.google.com/file/d/1pNx4nKzW-qQpIb7kovq-OyxZsA-B5mP5/view?usp=sharing

Version Control:

https://github.com/chsp7258/SWMT_Project_014_8

Individual Contributions:

Abdullahi: I created the front-end for the Explore page and edited Mara's home page to incorporate the wishlist functionality. Additionally, I handled the back-end for the home page to ensure its integration with the application's logic. I developed the API route to Yelp, enabling dynamic data fetching, and worked on the search-restaurant back-end, allowing users to query restaurants effectively. I was responsible for the wishlist implementation, managing both the front-end and back-end functionality for adding and removing restaurants from users' wishlists. Lastly, I contributed to some styling and the integration of everyone's code, ensuring the application functioned smoothly as a cohesive system.

Alec: I worked on a lot of the database design and backend. I wrote a lot of the backend APIs that interact with the database to rate restaurants, get your ratings, discover restaurants, and more. I also wrote tests for our backend. In addition, I helped out on version control and merge conflicts with our team as well as other miscellaneous tasks like sections of the readme.

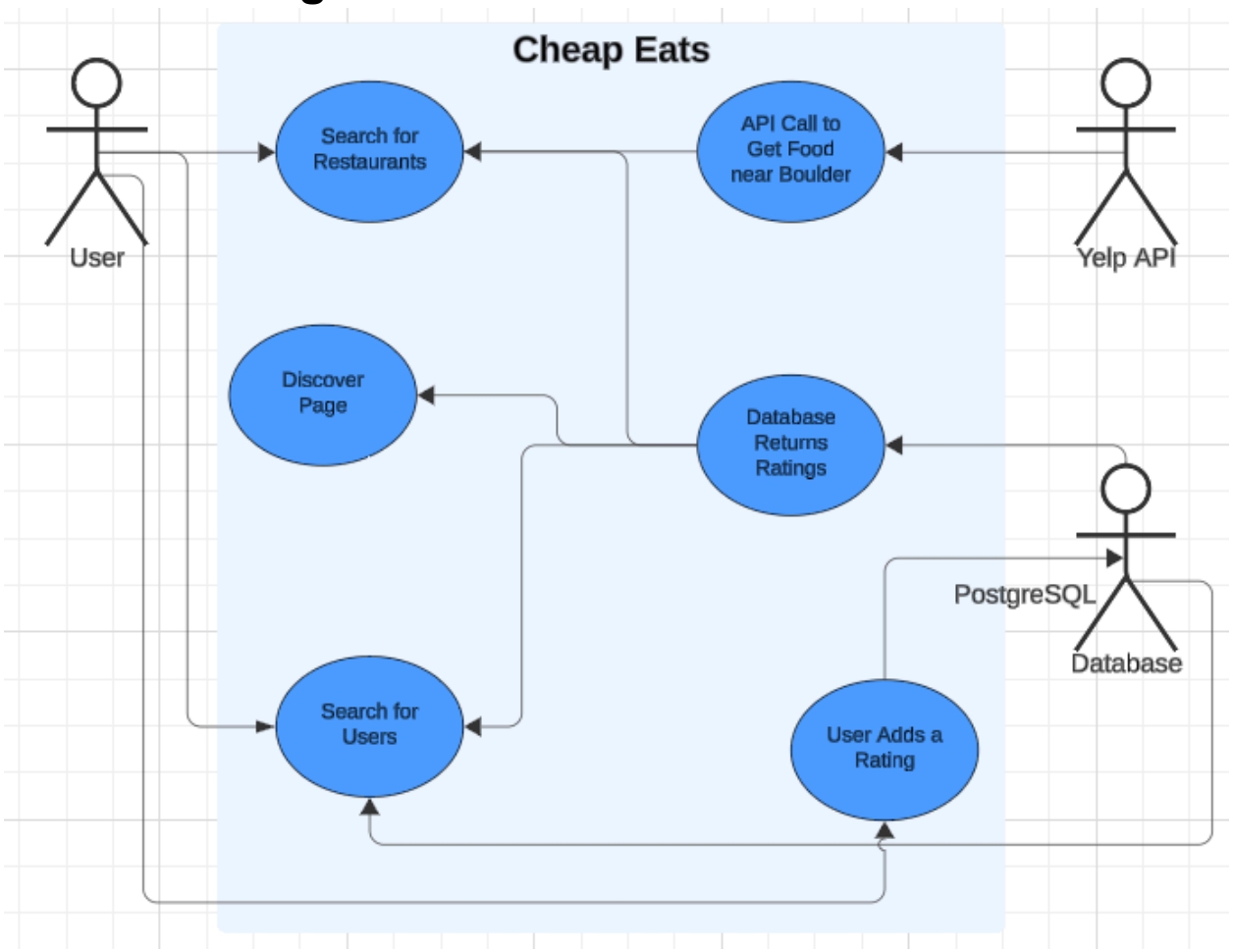
Christopher: I created the Navigation bar and implemented the functionality including a differentiation when logged in vs logged out. I also implemented the search for other users feature including such that the current user can see other users ratings and wishlist. This included the API applicable pages and API routes.

Mara: I did most of the ReadMe. I also created the original home page, which was later added to by my team members. I wrote the API routes for login, register, and logout and laid down the index.js template. I added styling to the discover, home, and add restaurant pages. I did the logout and message partials. I also worked on the discover page, adding cards to make it more readable.

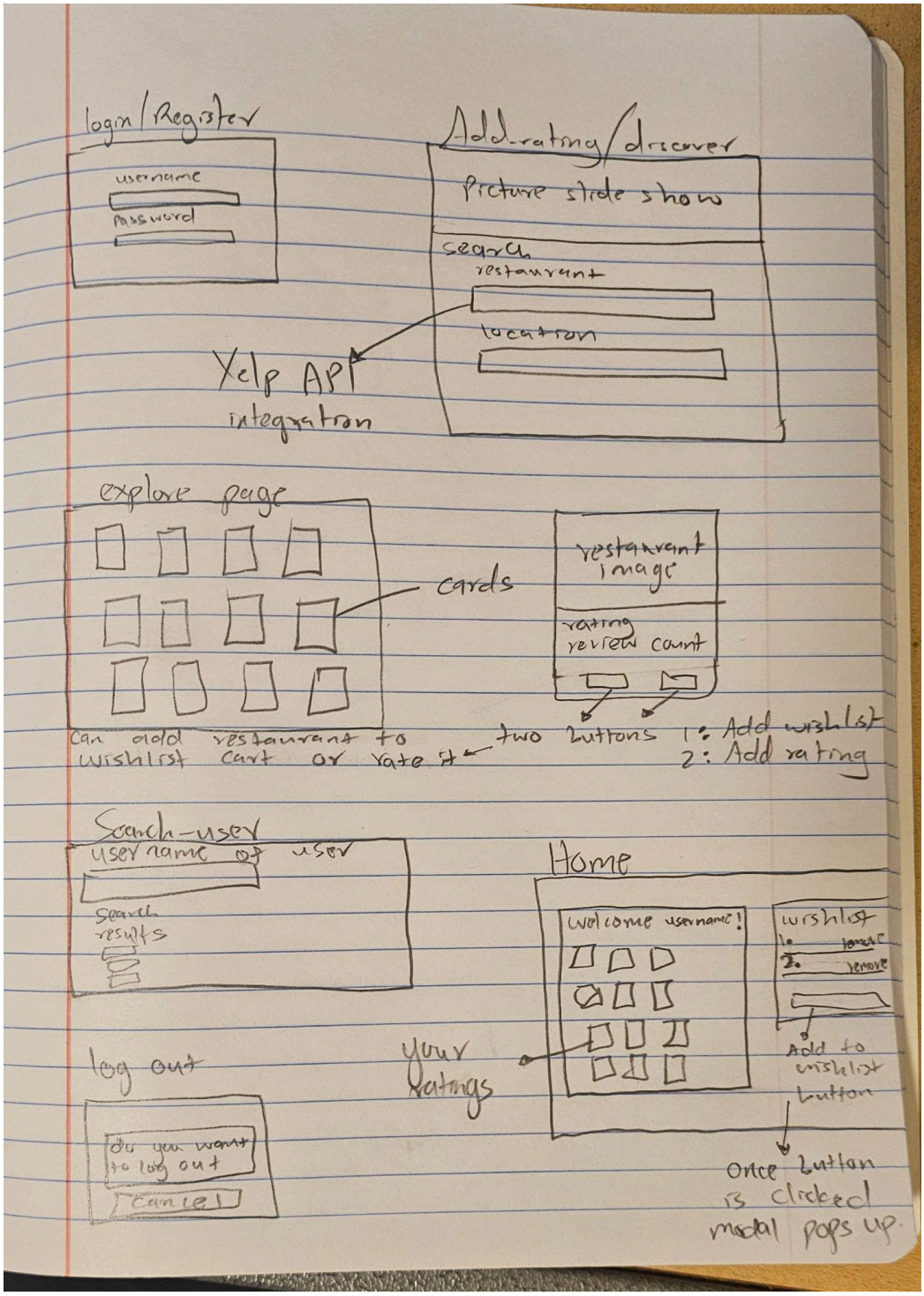
Nathnael: I worked on the database design and implementation, worked on the api routes for the registrations and logout pages. I worked on the add restaurants page, along with the add rating and add restaurant

endpoints. I created the function we used to rate restaurants.

Use Case Diagram:



Wireframes:


login/Register
username
password
Add_rating/discover
Picture slide show
search
restaurant
location
Yelp API
integration
explore page
cards
restaurant
image
rating
review count
Can add restaurant to
wishlist cart or rate it
two buttons 1: Add wishlist
2: Add rating
Search-user
user name of user
search
results
Home
welcome username!
wishlist
remove
remove
Add to
wishlist
button
your
ratings
log out
do you want
to log out
cancel
Once button
is clicked
modal pops up.

Test Plan and Observations:

User Testing Results: Lab 11

Test Cases and Observations

Note: THE DISCOVER PAGE WAS RENAMED THE ADD RATING PAGE ON SITE AFTER CONSIDERING WHAT THE PAGE DOES.

Test Case 1: Successful Restaurant Search and Display of Results

Participants:

- Test User 1: **Alex** (Student, 22 years old, unfamiliar with the application)
- Test User 2: **Sarah** (Professional, 35 years old, experience with similar apps)

Results:

1. **Alex's** Observations:
    - Navigated to the Add rating page (discover) and entered "Buff Restaurant" and "Boulder."
    - Clicked "Search" and reviewed the results.
    - Stated: "The interface is clean, but I wasn't sure if I could leave the location field blank to get nearby results."
    - Found accurate results but wished for more details like operating hours.
2. Outcome: Successful, but user feedback noted for providing more contextual instructions.
3. **Sarah's** Observations:
    - Quickly located the search fields and entered the data.
    - Commented: "The images and ratings were a nice touch, but I had to scroll to find 'Buff Restaurant' in the results."
4. Outcome: Successful; feedback suggests making results for exact matches appear at the top or have less restaurants as output.
5. This was done on a local host and I did lessen the number of results given by the search to 4 just in case the API grabs the wrong restaurant.

Overall: Both users found the results relevant, but minor improvements were suggested for user guidance and result prioritization. Although we still need the location to find the closest exact match in boulder.

Test Case 2: Handling Invalid Restaurant Name

Participants:

- Test User 3: **Farah** (student, 21 years old, casual application user)
- Test User 4: **Mia** (High school student, 16 years old, tech-savvy)

Results:

1. **Farah's** Observations:
    - Entered "djkgdhkl;a" as the restaurant name and "Boulder" as the location.
    - Observed: "Page just refreshes and nothing happened"
    - Suggested: "A message should pop up telling me what happened."
2. Outcome: Error message didn't display as expected.
3. **Mia's** Observations:
    - Entered invalid data and triggered the error message.
    - Stated: "The feedback is fine, but it didn't give me an option to search nearby instead of retrying."
4. Outcome: Successful error handling, but users preferred suggestions for nearby locations or additional search tips.
5. This was done on the local host as well and Mia's suggestion was good however the API will not do error handling so the easiest fix was to redo the entries.

Overall: After encountering the message error problem, a patch was deployed and when Mia tried the application it successfully worked, Although the suggestion couldn't be implemented they were satisfied with the alternative.

Test Case 3: Adding a Rating for a Restaurant

Participants:

- Test User 5: **Brian** (Random, 35 years old, able to use phone)
- Test User 6: **Emily** (College student, 19 years old, first-time tester of this app)

Results:

1. **Brian's** Observations:
    - Searched for "Oak at Fourteenth" suggested by me because he didn't know any restaurants and clicked "Add Rating."
    - Stated: "The pre-filled restaurant name was helpful, but it also looked like the other spots were prefilled which could be confusing."
    - Successfully submitted ratings and confirmed that the success message was visible.
2. Outcome: Successful, but better clarification of what to input is needed and maybe adjust rating to make sense.
3. **Emily's** Observations:

    - Followed the same process and submitted her ratings.
    - Commented: "The flow was intuitive, but I accidentally clicked 'Back' and lost my data before submitting."
4. Outcome: Successful submission after a second attempt; suggested auto-saving form inputs.
5. This was done on localhost suggestions were taken into consideration and the rating algorithm was adjusted to test for taste and expense.

Outcome: rating math was changed to incorporate taste and expense for the person to find a balance between best bang for buck restaurant. **Emily suggested adding a way to see your friends ratings and wishlist**.

Test case 4: Adding and removing wishlist restaurant.

Participants:

- Test User 1: **Braxton**(roomate, 20 years old, civil engineering)
- Test User 2: **Herny** (roomate, 20 years old, moderate app experience, first beta tester)

Results:

1. **Braxton's** Observations:
    - Successfully triggered the modal and added "The Sink" to the wishlist.
    - Commented: "The process was smooth, but the no message pop up"
    - Removed the restaurant from the homepage without issues.
2. Outcome: Successful; suggested making confirmation messages persistent until dismissed.
3. **Henry's** Observations:
    - Clicked "Add Wishlist" but initially thought the modal would auto-fill the restaurant name from search results.
    - Added "The Sink" manually and verified its appearance on the homepage.
    - Commented: "It would be nice to pre-fill the restaurant name if I already searched for it."
    - Removed the restaurant successfully.
4. Outcome: Successful; suggested integrating pre-filled restaurant names into the modal if triggered from search results.
5. Test was done on localhost. Suggestions were integrated to the explore page. Button now adds a restaurant to the wishlist directly from the card with no extra steps.

Outcome: Both said it worked fine and didn't need much to add. Simple and effective is better than overcomplication. Added that maybe adding a way to add restaurants from the explore page directly to the wishlist could be a good feature.

Summary of Changes Based on Testing

1. Improved error message text for invalid or valid entries to pop up.
2. Added logic to prioritize exact matches in search results. Less search results displayed (limited to 4).
3. Incorporated a rating scale description near the input fields and changed how good food tastes as well as how much financial capital needed to eat at a particular restaurant.
4. Added a tab to look for other users and see their ratings and wishlist cart.
5. Planned future implementation of auto-saving form data to prevent loss when navigating away.
6. Lastly for the wishlist to be able to add restaurants from the explore page by clicking a button.

These observations and feedback from diverse participants helped refine the application to ensure better usability and a more user-friendly experience.