SW Engineering CSC648/848 Summer 2024

ScholarEats

Team 4:

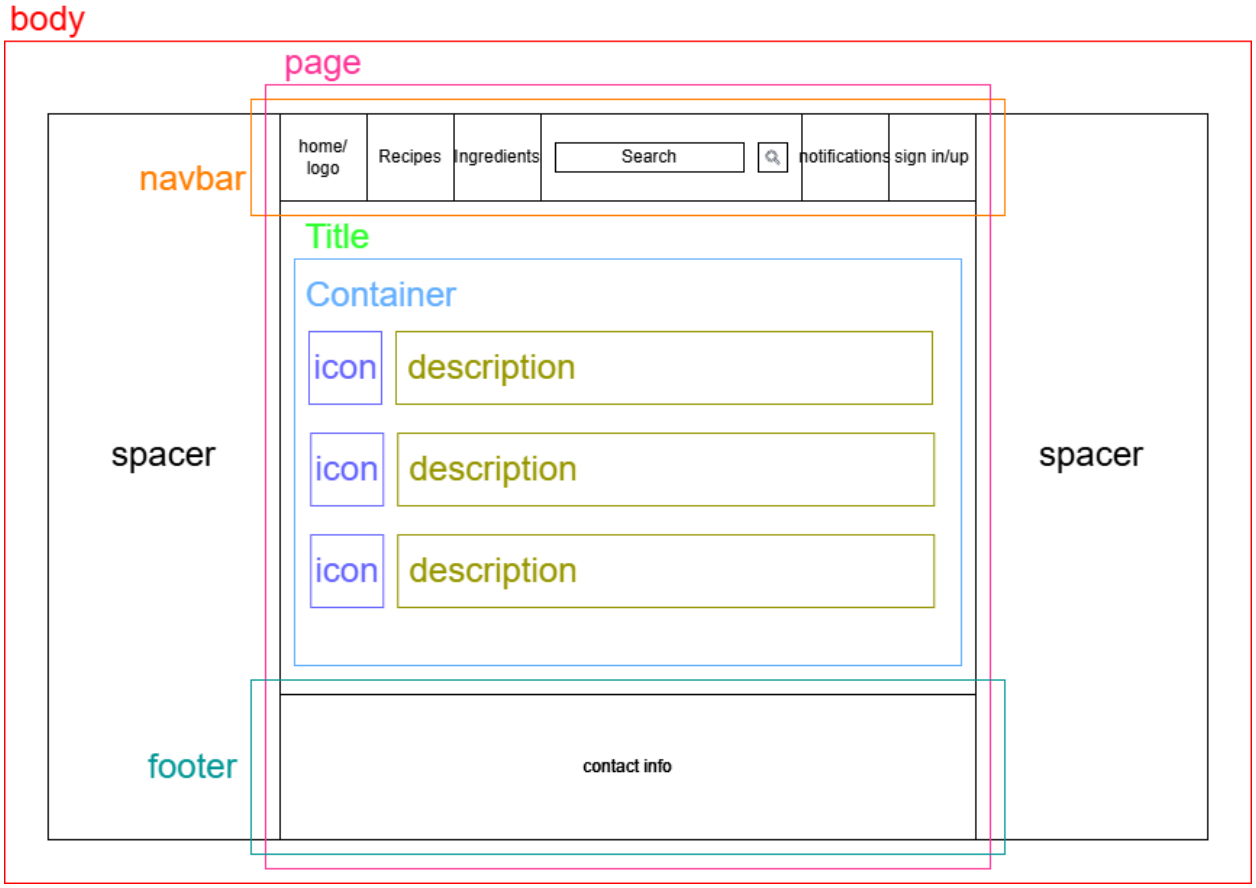
Angelo Arriaga (Team Lead - aarriaga1@sfsu.edu)
Donovan Taylor (Front-End Lead)
Hancun Guo (Front-End)
Tina Chou (Front-End)
Edward McDonald (Back-End Lead)
Karl Carsola (Back-End)
Sai Bavisetti (Database)
Maeve Fitzpatrick (Docs Editor)
Sabrina Diaz-Erazo (GitHub Master)

Milestone 3

07/23/24

History (Revisions)	
07/23/2024	Milestone 3 Part 2 feedback added
07/23/2024	Milestone 3 Version 1 Submission
07/22/2024	Milestone 2 Version 2 Submission
07/09/2024	Milestone 2 Version 1 Submission
06/26/2024	Milestone 1 Version 1 Re-Submission
06/20/2024	Milestone 1 Version 1 Submission

1. Data Definitions



Database:

activity_log	log_id
	user_id
	action
	timestamp

admin	admin_id
	uuid
	email
	username
	password_hash
	verification_status

	accessibility

email_log	email_log_id
	user_id
	email
	timestamp
	action

expired_products	ingredient_id
	Name
	expiration_date
	quantity
	store_id

has	user_id
	university_id

ingredient	ingredient_id
	name
	img_src

ingredient_copy	ingredient_id
	name

manages	admin_id
	user_id

notifications	notification_id

	user_id
	message
	is_read
	timestamp

recipe_ingredient	recipe_id
	ingredient_id

recipe_ingredient_copy	recipe_id
	ingredient_id

recipes	Unnamed: 0
	recipe_name
	prep_time
	cook_time
	total_time
	servings
	yield
	ingredients
	directions
	rating
	cuisine_path
	nutrition
	timing
	img_src
	dietary restrictions
	calories

	protein
	fat
	fiber
	difficulty
	cooking tip
	user_id
	MyUnknownColumn

roles	role_id
	role_name

sessions	session_id
	user_id
	session_start
	session_end
	expires
	data
	user_agent

store	store_id
	Name
	ingredient_id
	user_id
	quantity
	expiration_date
	university_id

university	university_id
	name
	email_suffix

user_info	user_info_id
	user_id
	profile_photo
	bio
	pronouns
	allergies
	dietary_restrictions
	favorited_recipes
	favorited_ingredients
	university
	modes

user_recipes	user_recipe_id
	user_id
	recipe_name
	prep_time
	cook_time
	total_time
	yield
	directions
	img_source
	nutrition

	servings
	dietary_restrictions
	calories
	fat
	protein
	fiber
	difficulty
	cooking_tip
	accessibility
	is_approved
	admin_id

Users	user_id
	uuid
	email
	username
	password_hash
	verification_status
	accessibility
	blacklist
	role_id
	university

2. Functional Requirements

PRIORITY 1 - Critical

- Administrators
    1. ADMINISTRATORS SHALL BE ABLE TO AUTHENTICATE THE USERS.
    2. ADMINISTRATORS SHALL BE ABLE TO BLACKLIST/ WHITELIST A USER
    3. ADMINISTRATORS SHALL BE ABLE TO EDIT THE INGREDIENT LIST
    4. ADMINISTRATORS SHALL BE ABLE TO REMOVE USER ACCOUNTS
    5. ADMINISTRATORS SHALL BE ABLE TO FOLLOW THE EXPIRATION DATE CLOSELY AND TAKE THE NECESSARY ACTIONS.
- System
    1. SYSTEM SHALL AUTOMATICALLY ENROLL USERS IN CORRESPONDING UNIVERSITY FOOD PROGRAMS
    2. SYSTEM SHALL BE ABLE TO GENERATE RECOMMENDED RECIPES
    3. SYSTEM SHALL TELL ADMINISTRATORS WHEN AN INGREDIENT HAS RUN OUT OF STOCK
    4. SYSTEM SHALL TELL ADMINISTRATORS WHEN FOOD HAS SPOILED
- Users
    1. USERS SHALL BE ABLE TO CHANGE THEIR PASSWORD
    2. USERS SHALL BE ABLE TO EDIT THEIR USERNAME
    3. USERS SHALL BE ABLE TO FILTER RECIPES BY COOKING AIDS REQUIRED
    4. USERS SHALL BE ABLE TO FILTER RECIPES BY DIETARY RESTRICTIONS
    5. USERS SHALL BE ABLE TO FILTER RECIPES BY DIFFICULTY
    6. USERS SHALL BE ABLE TO FILTER RECIPES BY INGREDIENTS
    7. USERS SHALL BE ABLE TO MAKE ACCOUNTS WITH VALID UNIVERSITY EMAIL.
    8. USERS SHALL BE ABLE TO SET ALLERGIES
    9. USERS SHALL BE ABLE TO SET DIETARY RESTRICTIONS
    10. USERS SHALL BE ABLE TO SORT RECIPES BY CALORIES
    11. USERS SHALL BE ABLE TO SORT RECIPES BY FAT
    12. USERS SHALL BE ABLE TO SORT RECIPES BY FIBER
    13. USERS SHALL BE ABLE TO SORT RECIPES BY PROTEIN
    14. USERS SHALL BE ABLE TO VIEW AVAILABLE INGREDIENTS
    15. USERS SHALL BE ABLE TO VIEW RECIPES

PRIORITY 2 - Desired

- Administrators
    1. ADMINISTRATORS SHALL BE ABLE TO REMOVE RECIPES OR REVIEWS
    2. ADMINISTRATORS SHALL BE ABLE TO MAKE UNIVERSITY-WIDE ANNOUNCEMENTS
- Systems
    1. SYSTEM SHALL ENSURE RECIPES ARE AVAILABLE FOR ALL (MAJOR) DIETARY RESTRICTIONS

- Users
    1. USERS SHALL BE ABLE TO FAVORITE INGREDIENTS
    2. USERS SHALL BE ABLE TO FAVORITE RECIPES
    3. USERS SHALL BE ABLE TO REVIEW/RATE RECIPES
    4. USERS SHALL BE ABLE TO SORT RECIPES BY RATING
    5. USERS SHALL BE ABLE TO VIEW OWN FAVORITE RECIPES
    6. USERS SHALL BE ABLE TO ADD THEIR PREFERRED PRONOUNS
    7. USERS SHALL BE ABLE TO ALLOW NOTIFICATIONS
    8. USERS SHALL BE ABLE TO EDIT PROFILE BIO
    9. USERS SHALL BE ABLE TO SHARE RECIPES
    10. USERS SHALL BE ABLE TO SWITCH BETWEEN LIGHT AND DARK MODE
    11. USERS SHALL BE ABLE TO VIEW HISTORY OF RECEIVED RECIPES

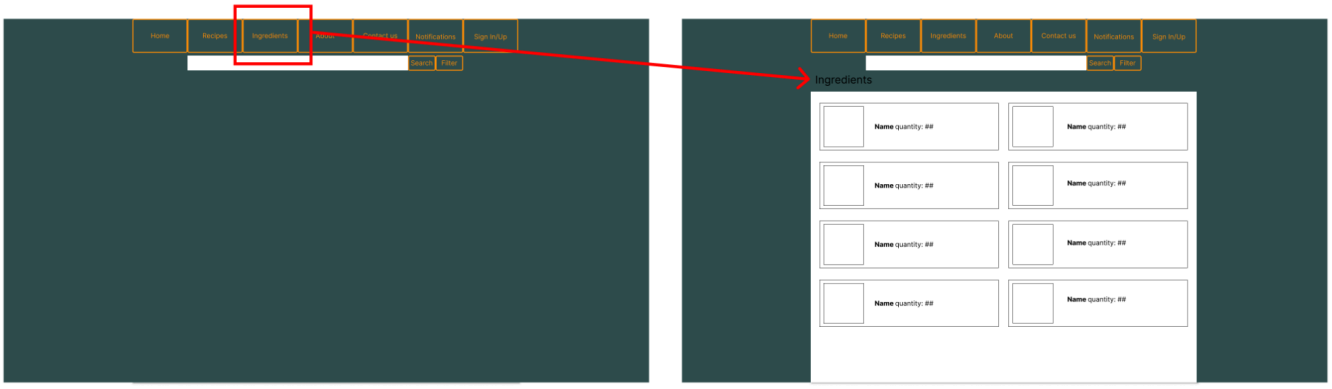
PRIORITY 3 - Opportunistic

- Administrators
    1. ADMINISTRATORS SHALL BE ABLE TO DELETE USER POSTED RECIPES
    2. ADMINISTRATORS SHALL BE ABLE TO PROMOTE RECIPES
    3. ADMINISTRATORS SHALL BE ABLE TO VIEW ALL USER-REPORTS
- Users
    1. USERS SHALL BE ABLE TO BLOCK OTHER USERS
    2. USERS SHALL BE ABLE TO CHANGE PROFILE PHOTO
    3. USERS SHALL BE ABLE TO REPORT PHOTOS OR REVIEWS
    4. USERS SHALL BE ABLE TO REPORT USER ACCOUNTS
    5. USERS SHALL BE ABLE TO SUBMIT RECIPES
    6. USERS SHALL BE ABLE TO TRANSFER ENROLLMENT IN UNIVERSITY FOOD PROGRAMS
    7. USERS SHALL BE ABLE TO UPLOAD PHOTOS OF THE RECIPES THEY COOK FOR REVIEWS
    8. USERS SHALL BE ABLE TO VIEW OTHER USER’S FAVORITE INGREDIENTS
    9. USERS SHALL BE ABLE TO VIEW OTHER USER’S FAVORITE RECIPES

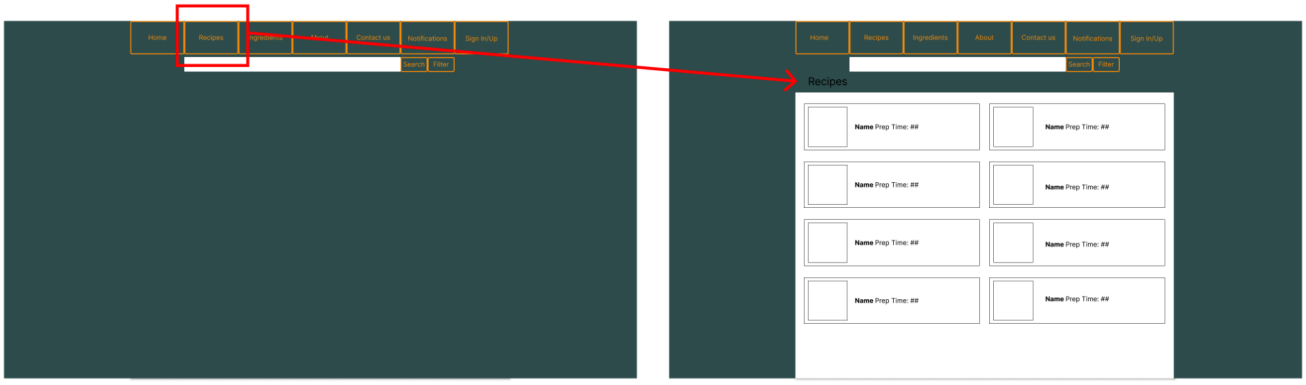
3. Wireframes Based on your Mockups/Storyboards (detailed)

[Figma Link]

1. Meal Planning
    a. Since the student is already using his university's grocery program, all he needs to do is go online and look at the recipes that are provided for his school's inventory. Once he signs up, he can browse a myriad of recipes using ingredients supplied by Gator Groceries.

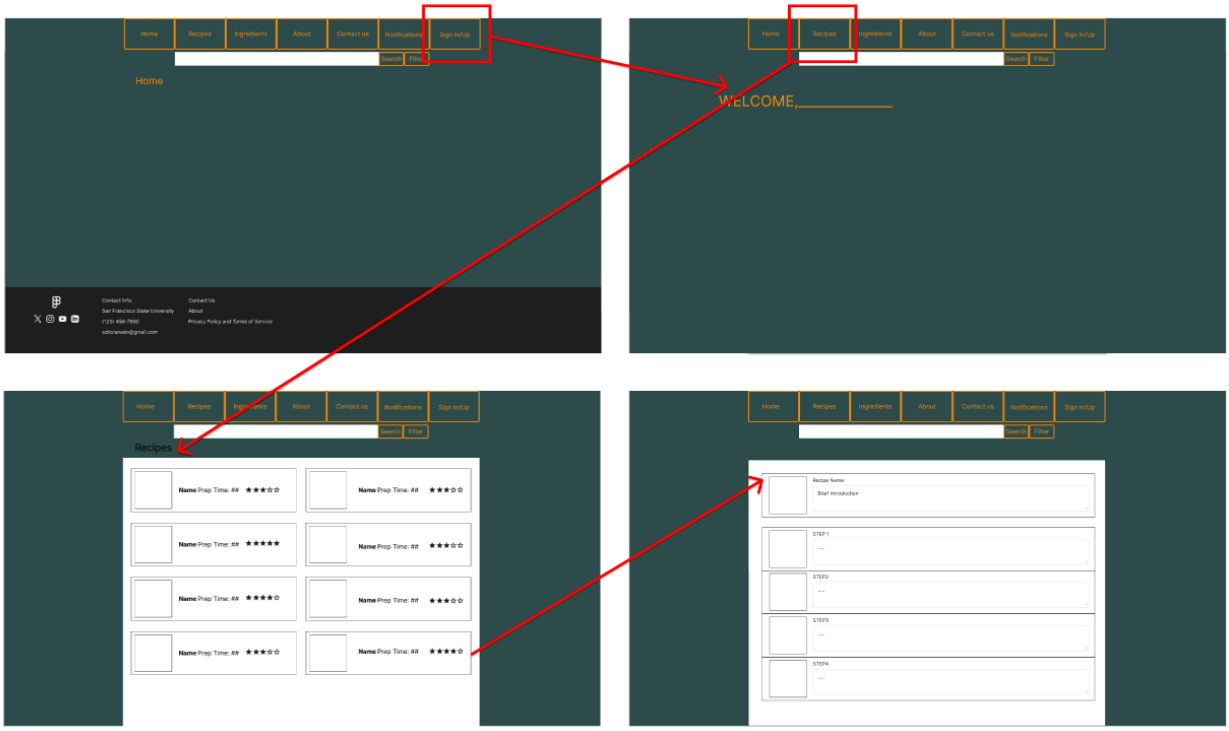


2. Money Saving
    a. Jane utilizes an online service (ScholarEats), which shows her the food available this week at her school's free food program. Along with this information, she also receives a few auto-generated recipes. This further encourages her to go to her university's free food program.



3. Waste Reduction
    a. Julie then finds out about ScholarEats and accesses it on her phone. Julie is amazed with the variety of recipes that are available and how every ingredient from Gator Groceries is utilized in some way. Now, when she goes to Gator Groceries, she can get exactly the type and amount of ingredients she needs without having to worry about any of them going to waste. She also likes that there is an inventory feature so she does not have to worry about showing up to Gator Groceries and having to find out that everything is gone.

4. Health Consciousness
    a. Jack logged in and checked available ingredients. He selected a low-carb recipe. The platform provided necessary ingredients and step-by-step cooking instructions. Jack picked up the ingredients from campus grocery hub and cooked his meal. This program helped him maintain a healthy diet.



5. Student Engagement
    a. Joe looks into his automatically-enrolled account, and opts in for the next week's groceries. After opting in himself, Joe posts about the program on social media, which catches their friend Dwight's attention; he is intrigued by the program, looks into it himself and also opts in for the groceries.

6. Time Saving
    a. Jessie, a student who is currently attending full-time at her university and a part-time worker at a convenience store, has a hard time balancing her school and work life. Despite the many efforts that she has made to manage her time properly she always falls short with the time she has for herself. Luckily, she found that she can save more time with the use of ScholarEats where Jessie can easily get recipes based on the groceries that her university has. With this, she is able to be stress free and able to spend less time on finding what to make.

7. Cooking Education
    a. Olivia can use this app that her school manages locally to see which recipes have ingredients readily available within the university's food pantry. Olivia confirms her intention of picking up a set of ingredients earlier in the day and gets notified when it is ready to pick up. Olivia picks up the ingredients. Olivia goes to her dorm and the app gives her step by step instructions on how to cook.

8. Community Recipe Submission
    a. Motivated by a desire to engage with her peers and enrich the campus dining experience, she logs into the ScholarEats platform, eager to introduce others to her culinary creation. The platform's user-friendly interface for the recipe submission process includes boxes for the ingredient list, step-by-step instructions, and a place to submit optional photos of the dish. It is well organized and easy to navigate. She appreciates the option to preview her entry before finalizing it. Upon submission, the ScholarEats system quickly processes the recipe, displaying a confirmation message that reassures Emily that her contribution has been received and is under review. Emily receives a notification when her recipe is live, adding a sense of accomplishment and fostering a deeper sense of community involvement.

We are not utilizing Use Case 8 anymore because it takes away from the main purpose of the application, which is compiling recipes that utilize the available ingredients that the university is offering. Allowing recipe submissions from students would introduce recipes that may require a whole lot more ingredients than what the user has and/or can afford.

4. High level database Architecture and organization (detailed)

Database-Related Functional Requirements:

1. ADMINISTRATORS SHALL BE ABLE TO EDIT THE INGREDIENT LIST
2. ADMINISTRATORS SHALL BE ABLE TO REMOVE USER ACCOUNTS
3. ADMINISTRATORS SHALL BE ABLE TO FOLLOW THE EXPIRATION DATE CLOSELY AND TAKE THE NECESSARY ACTIONS.
4. SYSTEM SHALL BE ABLE TO GENERATE RECOMMENDED RECIPES
5. SYSTEM SHALL TELL ADMINISTRATORS WHEN AN INGREDIENT HAS RUN OUT OF STOCK
6. SYSTEM SHALL TELL ADMINISTRATORS WHEN FOOD HAS SPOILED
7. USERS SHALL BE ABLE TO FILTER RECIPES BY DIETARY RESTRICTIONS
8. USERS SHALL BE ABLE TO SET ALLERGIES
9. USERS SHALL BE ABLE TO SET DIETARY RESTRICTIONS
10. USERS SHALL BE ABLE TO VIEW RECIPES

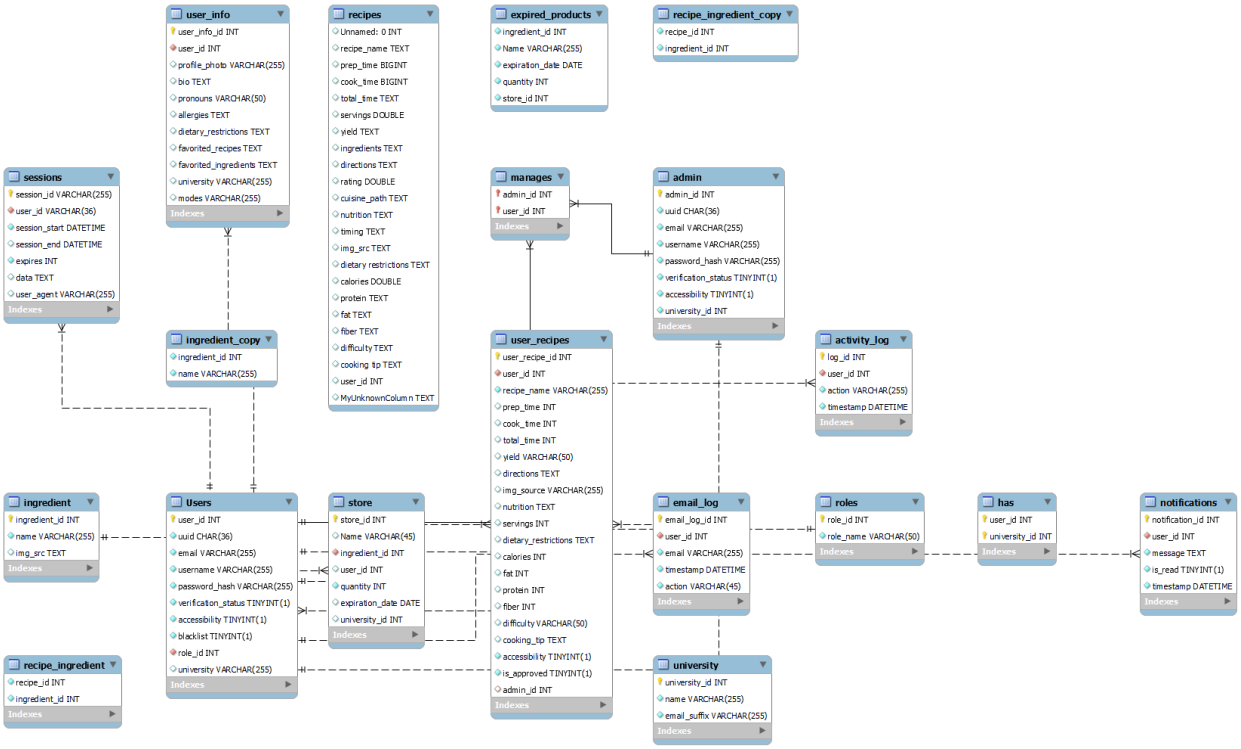
DBMS Definition:

The Database Management System used for this project is MySQL. MySQL is an open-source relational database management system known for its reliability, ease to use and performance. It is widely used for web type applications and supports SQL for managing the data.

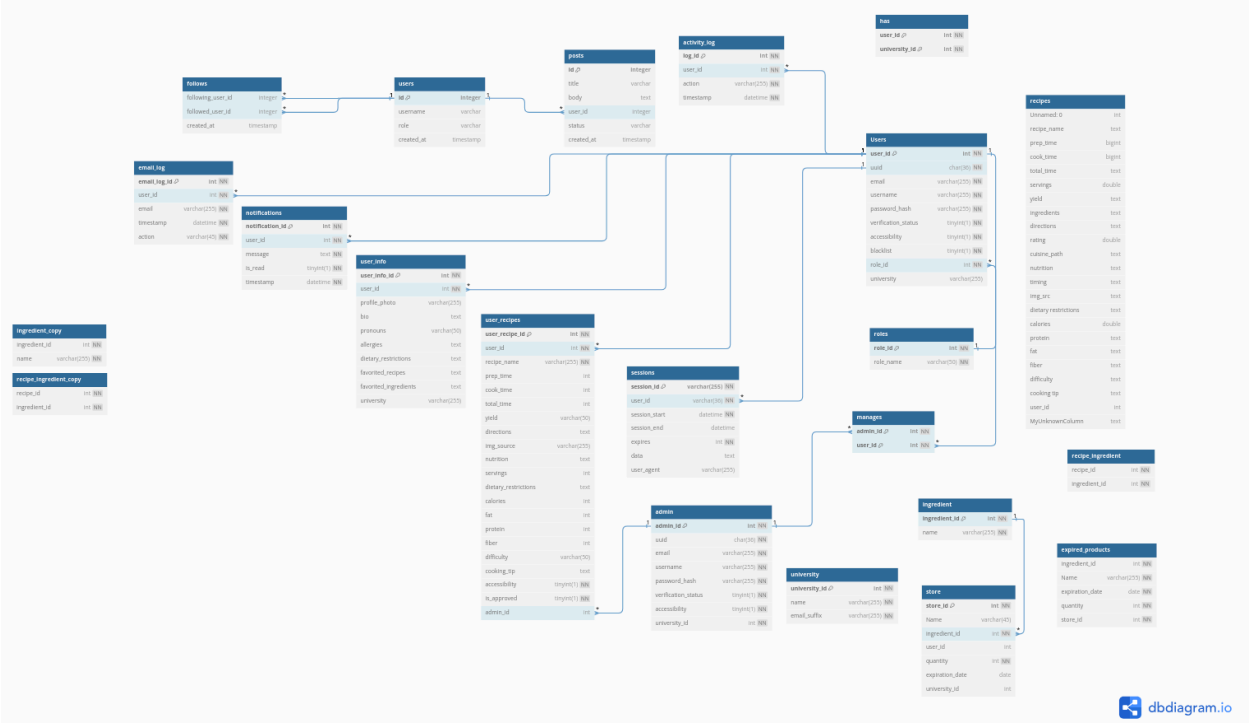
Entity Descriptions:

1. Users: Contains user information such as email, username, password hash, and other personal details.
2. Recipes: Contains recipe details such as recipe name, prep time, cook time, ingredients and nutritional information.
3. Ingredients: Stores information about ingredients used in recipes.
4. Recipe_Ingredient: A join table that links recipes and ingredients.
5. Store: Manages the ingredients a user has in their store, including quantity and expiration date.
6. User_Recipes: Links users to their favorite or created recipes.
7. Activity_Log: Records user activities such as logins and specific actions within the application.
8. Notifications: Stores notifications sent to users.
9. Roles: Manages user roles and their permissions.
10. University: Contains information about universities, which users can be associated with.
11. Email_Log: Logs emails sent to users for activities like verification and notifications.
12. Manages: Links admin users to the users they manage.
13. Admin: Contains information about admin users.
14. Sessions: Tracks user sessions for login and activity tracking.
15. Expired_Products: Tracks products that have expired in the user's store.

EER Diagram:

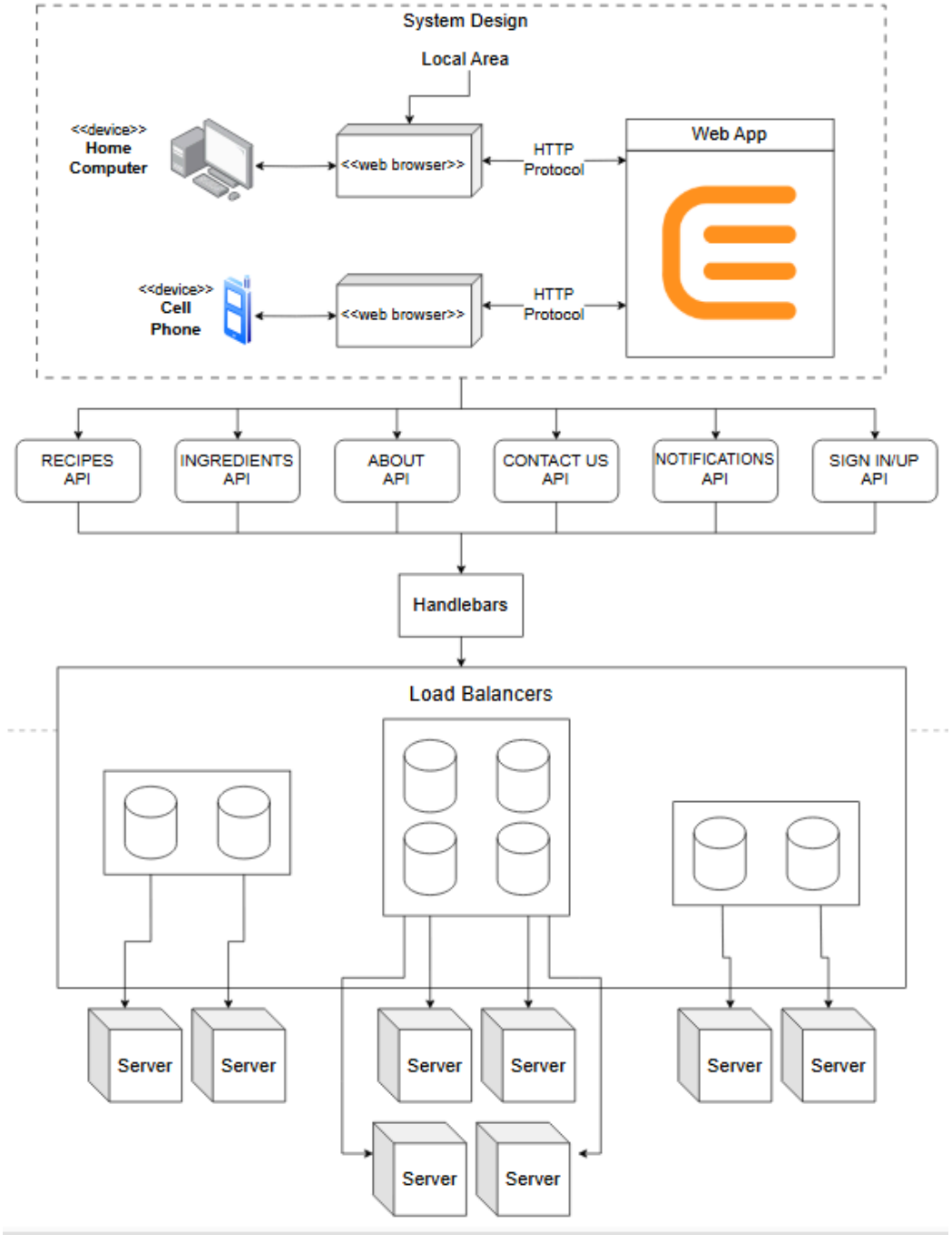

user_info
user_info_id INT
user_id INT
profile_photo VARCHAR(255)
bio TEXT
pronouns VARCHAR(50)
allergies TEXT
dietary_restrictions TEXT
favorited_recipes TEXT
favorited_ingredients TEXT
university VARCHAR(255)
modes VARCHAR(255)
Indexes
recipes
Unnamed: 0 INT
recipe_name TEXT
prep_time BIGINT
cook_time BIGINT
total_time TEXT
servings DOUBLE
yield TEXT
ingredients TEXT
directions TEXT
rating DOUBLE
cuisine_path TEXT
nutrition TEXT
timing TEXT
img_src TEXT
dietary restrictions TEXT
calories DOUBLE
protein TEXT
fat TEXT
fiber TEXT
difficulty TEXT
cooking tip TEXT
user_id INT
MyUnknownColumn TEXT
expired_products
ingredient_id INT
Name VARCHAR(255)
expiration_date DATE
quantity INT
store_id INT
recipe_ingredient_copy
recipe_id INT
ingredient_id INT
sessions
session_id VARCHAR(255)
user_id VARCHAR(36)
session_start DATETIME
session_end DATETIME
expires INT
data TEXT
user_agent VARCHAR(255)
Indexes
ingredient_copy
ingredient_id INT
name VARCHAR(255)
manages
admin_id INT
user_id INT
Indexes
admin
admin_id INT
uuid CHAR(36)
email VARCHAR(255)
username VARCHAR(255)
password_hash VARCHAR(255)
verification_status TINYINT(1)
accessibility TINYINT(1)
university_id INT
Indexes
user_recipes
user_recipe_id INT
user_id INT
recipe_name VARCHAR(255)
prep_time INT
cook_time INT
total_time INT
yield VARCHAR(50)
directions TEXT
img_source VARCHAR(255)
nutrition TEXT
servings INT
dietary_restrictions TEXT
calories INT
fat INT
protein INT
fiber INT
difficulty VARCHAR(50)
cooking_tip TEXT
accessibility TINYINT(1)
is_approved TINYINT(1)
admin_id INT
Indexes
activity_log
log_id INT
user_id INT
action VARCHAR(255)
timestamp DATETIME
Indexes
ingredient
ingredient_id INT
name VARCHAR(255)
img_src TEXT
Indexes
Users
user_id INT
uuid CHAR(36)
email VARCHAR(255)
username VARCHAR(255)
password_hash VARCHAR(255)
verification_status TINYINT(1)
accessibility TINYINT(1)
blacklist TINYINT(1)
role_id INT
university VARCHAR(255)
Indexes
store
store_id INT
Name VARCHAR(45)
ingredient_id INT
user_id INT
quantity INT
expiration_date DATE
university_id INT
Indexes
email_log
email_log_id INT
user_id INT
email VARCHAR(255)
timestamp DATETIME
action VARCHAR(45)
Indexes
roles
role_id INT
role_name VARCHAR(50)
Indexes
has
user_id INT
university_id INT
Indexes
notifications
notification_id INT
user_id INT
message TEXT
is_read TINYINT(1)
timestamp DATETIME
Indexes
recipe_ingredient
recipe_id INT
ingredient_id INT
Indexes
university
university_id INT
name VARCHAR(255)
email_suffix VARCHAR(255)
Indexes

ERD Diagram:

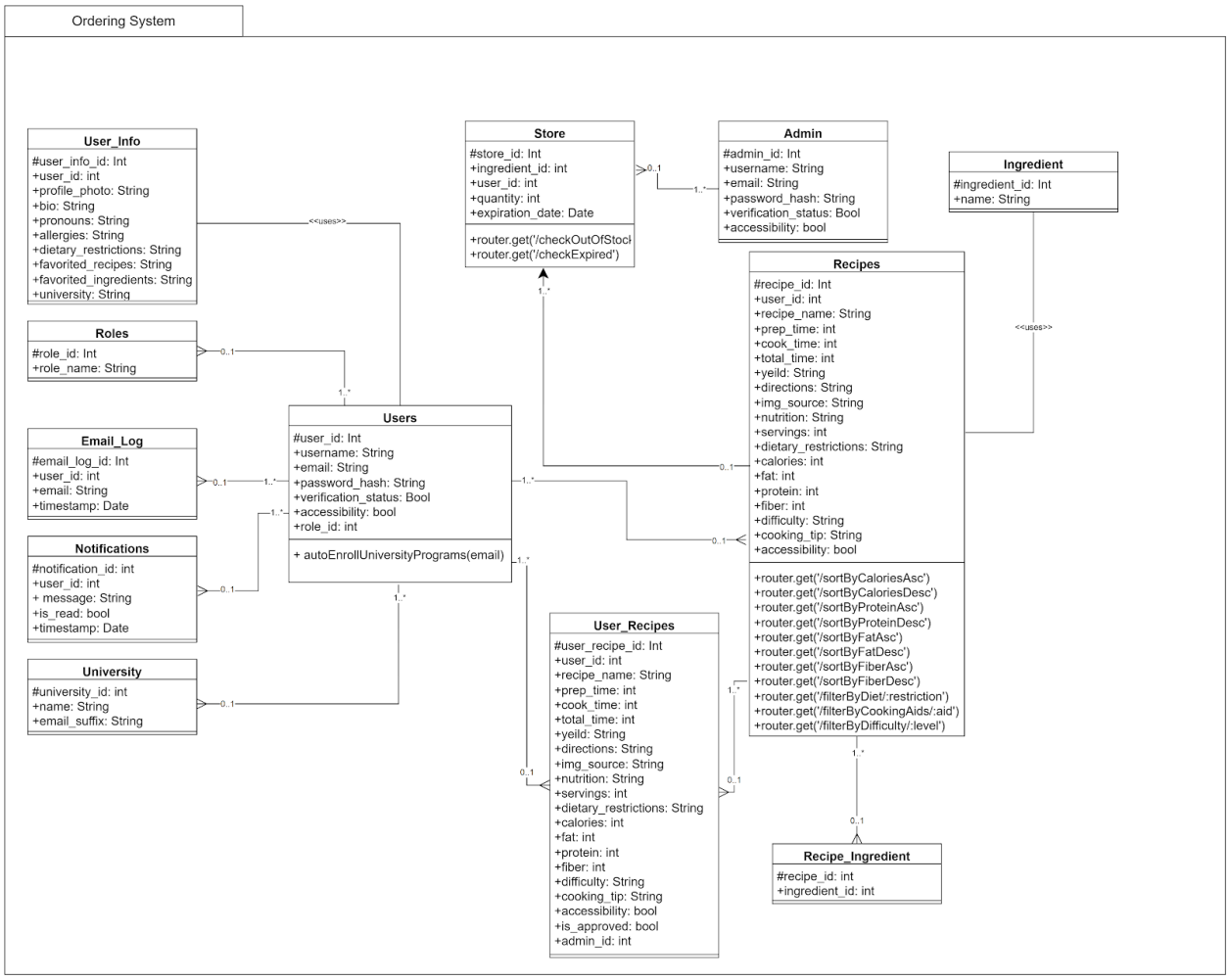

follows
following_user_id integer
followed_user_id integer
created_at timestamp
users
id integer
username varchar
role varchar
created_at timestamp
posts
id integer
title varchar
body text
user_id integer
status varchar
created_at timestamp
activity_log
log_id int NN
user_id int NN
action varchar(255) NN
timestamp datetime NN
has
user_id int NN
university_id int NN
email_log
email_log_id int NN
user_id int NN
email varchar(255) NN
timestamp datetime NN
action varchar(45) NN
notifications
notification_id int NN
user_id int NN
message text NN
is_read tinyint(1) NN
timestamp datetime NN
user_info
user_info_id int NN
user_id int NN
profile_photo varchar(255)
bio text
pronouns varchar(50)
allergies text
dietary_restrictions text
favorited_recipes text
favorited_ingredients text
university varchar(255)
Users
user_id int NN
uuid char(36) NN
email varchar(255) NN
username varchar(255) NN
password_hash varchar(255) NN
verification_status tinyint(1) NN
accessibility tinyint(1) NN
blacklist tinyint(1) NN
role_id int NN
university varchar(255)
recipes
Unnamed: 0 int
recipe_name text
prep_time bigint
cook_time bigint
total_time text
servings double
yield text
ingredients text
directions text
rating double
cuisine_path text
nutrition text
timing text
img_src text
dietary restrictions text
calories double
protein text
fat text
fiber text
difficulty text
cooking tip text
user_id int
MyUnknownColumn text
ingredient_copy
ingredient_id int NN
name varchar(255) NN
recipe_ingredient_copy
recipe_id int NN
ingredient_id int NN
user_recipes
user_recipe_id int NN
user_id int NN
recipe_name varchar(255) NN
prep_time int
cook_time int
total_time int
yield varchar(50)
directions text
img_source varchar(255)
nutrition text
servings int
dietary_restrictions text
calories int
fat int
protein int
fiber int
difficulty varchar(50)
cooking_tip text
accessibility tinyint(1) NN
is_approved tinyint(1) NN
admin_id int
sessions
session_id varchar(255) NN
user_id varchar(36) NN
session_start datetime NN
session_end datetime
expires int NN
data text
user_agent varchar(255)
roles
role_id int NN
role_name varchar(50) NN
manages
admin_id int NN
user_id int NN
recipe_ingredient
recipe_id int NN
ingredient_id int NN
admin
admin_id int NN
uuid char(36) NN
email varchar(255) NN
username varchar(255) NN
password_hash varchar(255) NN
verification_status tinyint(1) NN
accessibility tinyint(1) NN
university_id int NN
ingredient
ingredient_id int NN
name varchar(255) NN
university
university_id int NN
name varchar(255) NN
email_suffix varchar(255) NN
store
store_id int NN
Name varchar(45)
ingredient_id int NN
user_id int
quantity int NN
expiration_date date
university_id int
expired_products
ingredient_id int NN
Name varchar(255) NN
expiration_date date NN
quantity int NN
store_id int NN
dbdiagram.io

5. High Level Diagrams (detailed)

Our system architecture involves every user utilizing either a computer or mobile device that has access to a web browser. That web browser accesses our web app, ScholarEats, over HTTP protocol. The web app interacts with several APIs, including the following tabs: Recipes, Ingredients, About, Contact Us, Notifications and Sign In/Up. The API requests are processed by Handlebars, which then sends the requests to the different load balancers. The load balancers distribute to whatever server is available.



UML Diagram:



Summary of UML Diagram:

The main entities are ‘Users’, ‘User_Info’, ‘Email_Log’, ‘University’, ‘Store’, ‘Admin’, ‘Recipes’, and ‘Ingredient’.

The ‘User_Info’ class stores detailed user information, such as ‘profile photo’, ‘bio’, ‘pronouns’, ‘allergies’, ‘dietary restrictions’, ‘favorited ingredients’, ‘favorited recipes’, and the ‘university’ the user belongs to. It has a one-to-one association with the ‘Users’ class.

The ‘Users’ class contains basic user information, including ‘username’, ‘email’, ‘password hash’, ‘verification status’, and ‘role ID’. It also has a method to automatically enroll users in university programs based on their email domain. The decision to make the User and User_Info entities separate classes was made in order to simplify maintenance.

The Roles class assigns a role to each user, which is linked through the ‘role_id’. Having Roles as its own class will help with scalability if future roles are ever created.

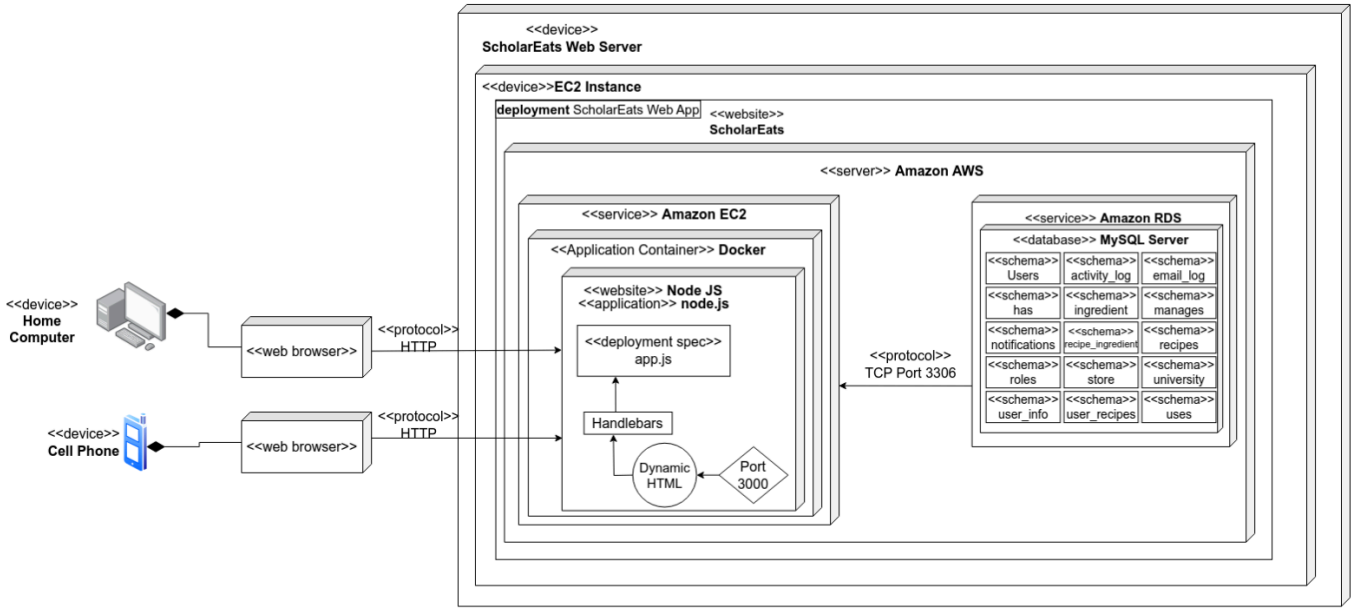
The ‘University’ class contains details about different universities, including their ‘names’ and ‘email suffixes’, which are used for auto-enrollment. The ‘Admin’ class stores admin details such as ‘username’, ‘email’, ‘password hash’, ‘verification status’, and ‘accessibility’. By keeping these entities in separate classes, this design helps to maintain data integrity. Also, making the Admin entity separate from a regular User entity was chosen for security reasons.

The ‘Store’ class manages inventory, including ‘ingredient IDs’, ‘quantities’, and ‘expiration dates’. It contains methods to check for out-of-stock items and expired ingredients and has a one-to-many relationship with the ‘Users’ class. This class is intended to help maintain the inventory in a centralized and consistent way.

The ‘Recipes’ class captures recipe details like ‘recipe name’, ‘preparation time’, ‘cook time’, ‘total time’, ‘yield’, ‘directions’, ‘image source’, ‘nutrition’, ‘servings’, ‘dietary restrictions’, ‘calories’, ‘fat’, ‘protein’, ‘fiber’, ‘difficulty’, ‘cooking tips’, and ‘accessibility’. It includes several methods for sorting and filtering recipes based on these attributes. The ‘User_Recipes’ class links users to their recipes, capturing details such as ‘servings’, ‘directions’, ‘ingredients’, and ‘admin approval status’. Having these various filters and sort options helps to optimize data retrieval.

The ‘Ingredient’ class lists possible ingredients, while the ‘Recipe_Ingredient’ class maps ingredients to recipes, forming a many-to-many relationship between ‘Recipes’ and ‘Ingredient’. By separating these classes, we can include information, like preparation instructions or quantity for a particular ingredient in a particular recipe without affecting the definition/function of the Recipe or Ingredient as a whole.

Application Network and Deployment Design:



6. List of Contributions in this milestone (detailed including contributions to the horizontal prototype)

- Donovan: (10 /10)
    - Adjusted Routing for the individual recipes in the database to have their own pages
    - Adjusted UI to incorporate filtering within search results
    - Notifications Page
    - Site-wide .css adjustments
    - Restructured handlebar's structure to improve ease of development
    - Iterated on Navigation bar to follow coding convention
    - Led a meeting regarding the Wireframes for the Use-cases
- Hancun: (8 /10)
    - Admin Tools Page
    - Created default image asset for recipes
    - Created default image asset for ingredients
    - Contributed to .css adjustments
    - Assisted in site-wide improvements
- Tina: (7.5 /10)
    - Added fields to Account Management to account for allergies, website theme and more
    - Drafted the handlebars view for the Individual Recipe page
    - Drafted the handlebars view for the site's Landing page
    - Contributed to the updating of the Use-Cases' Wireframes
- Edward: (10 /10)
    - Configured possibility of filtering within a given search result.
    - Created ways to check if user was logged in between pages.
    - Led the adjustments to Section 4 and 5 following the return of feedback from Milestone 2 Version 2
    - Led meetings to iterate on the back-end and set the groundwork for upcoming features
- Karl: (8.5 /10)
    - Connected Allergies to Database
    - Connected Dietary Restrictions to Database
    - Created ways to check if a user was logged in between pages.
    - Adjusted various validations
    - Restructured session management
- Sai: (7.5 /10)
    - Implemented methods for User Authentication
    - Implemented methods for Inventory Management
    - Populated various database tables
    - Admin Authorization Table
    - Updated the default values of the recipes and ingredients to point to Hancun's new assets for their image source (img_src column)

- Maeve: (8 /10)
    - Contributed to the updating of the Use-Cases' Wireframes
    - Managed and oversaw the move to documenting on Notion
    - Managed and oversaw the move back FROM Notion after our storage requirements changed
- Sabrina: (8 /10)
    - Contributed to the updating of the Use-Cases' Wireframes
    - Assisted in the UI iterations
    - Assisted in implementation of user preferences for the site
    - Drafted CSS for Dark Mode
    - Adjusted Account Management options

Prototype Feedback

- Simple but good mockups/use cases
- Should make a separation/space between the header components - it's too easy to accidentally click on the wrong button
- put a placeholder like "Search for recipes" in search bar - currently unclear what user should search for
- Must clarify that the results are recommendations when there are no results - ex. "No matches - here are some recommendations"
- Tagline itself is really good -says exactly what the website is about - but font size should be increased and should be formatted to not be in a single sentence
- Distinguish coloring of header and footer from the main content - currently confusing
- Registration is not good/doesn't work – make it look like Login
- WE USED ARROWS IN OUR WEB FRAMES YAY
- Make sure site matches the use cases - specifically header buttons
- User doesn't know if logged in – "Welcome" isn't good enough - add other signifier (ex. Hi _____ or Default Profile Picture)

Following the feedback given during class the team immediately met to discuss how to apply the feedback and correct any deficiencies that we saw. We immediately tasked members on site-wide .css adjustments involving the colors chosen on the site and adjust the UI for users to have a better experience traversing the site.

Additionally the team has moved to incorporate changes that allow for clearer error messaging for the user to understand why they are seeing certain results or why a search reveals recommendations instead of what they searched for. This will also expand to better showing that the user has logged in instead of going to another page without seemingly any response.