SW Engineering CSC 648-05 Spring 2023

RecipeReel

T03 Milestone 5

Team lead: Yueling Liu yliu50@sfsu.edu

Front end lead: Duncan Herington

Document lead: Marcel Azouri

GitHub & Database & Backend: Nathan Le Howland

History table

M5V1	May 25, 2023
M4V1	May 18, 2023
M3V2	May 18, 2023
M3V1	April 27, 2023
M2V2	April 27, 2023
M2V1	April 3, 2023
M1V2	April 3, 2023
M1V1	March 2, 2023

1. Table of Contents

2. Product Summary

- Product Name: RecipeReel
- Selling our product:

  Welcome to the future of recipe sharing !RecipeReel is not just an average recipe website; it's a dynamic and engaging social platform designed to connect food enthusiasts from all walks of life. At RecipeReel, we believe that cooking is more than just following instructions; it's an art form that should be shared, and enjoyed together. Our platform goes beyond the traditional recipe website and it creates a vibrant and interactive space where food lovers can connect, inspire, and be inspired with one another. It provides a user-friendly interface that feels like a social media platform, users can browse recipes, they can rate a recipe,leaving a comment to engage with the author who posted the recipes when they are signed up and logged in. We are aiming to make cooking a social experience, building connections with like-minded individuals who share their love for cooking.

- Final Priority 1 Functions
  - General User:
    - A general user shall be able to register for an account on RecipeReel using a username, password, and email
    - A general user shall be able to upload their profile picture while registering an account
    - A general user shall be able to search using a keyword or terms that match a recipe’s ingredients
    - A general user shall be able to review the details, including instructions,title, difficulty level, ingredients and comments.
    - A general user shall be able to see the followers and who are following the fellow posted recipe user.
  - Registered User:
    - A registered user shall be able to log in with an email and password.
    - A registered user shall be able to leave comments on recipes.

        - A registered user shall be able to delete their own comments
        - A registered user shall be able to post recipes to RecipeReel
        - A registered user shall be able to log out.
        - A registered user shall be able to follow other registered users.
        - A registered user shall be able to unfollow other registered users.
        - A registered user shall be able to delete a recipe they posted to RecipeReel.
        - A registered user shall be able to save recipes to their favorite recipes collection
        - A registered user shall be able to remove a saved recipe from their favorite recipe collection

- What makes RecipeReel unique?

  RecipeReel believes that cooking is not just about the food, but about community. In order to reflect this, RecipeReel behaves more like a social media platform rather than a traditional recipe website. This could help others discover the best recipe and create a fun and engaging and interactive space where food enthusiasts can come together to share their passion for cooking.

- URL to RecipeReel: https://master.d3u7tlcu99u8gi.amplifyapp.com/

3. Milestone documents – M1-M4

SW Engineering CSC 648-05 Spring 2023

RecipeReel

T03 Milestone 1

Team lead: Yueling Liu yliu50@sfsu.edu

Backend lead: Duncan Herington, Marcel Azouri

Frontend lead: Priya Pradeep

GitHub lead: Nathan Le Howland

Database lead: Samuel Elias

Document lead: Yasson Haddish

History table

M1V2	April 3, 2023
M1V1	March 2, 2023

Table of Contents

1. Executive Summary

Consider for a second that you are at home, perusing social media while attempting to determine what to make for dinner. You come across a picture of a dish that looks really scrumptious, and by simply glancing at it, you could almost taste the flavor. But, you encounter a frustrating situation when attempting to locate the recipe. Just to find the recipe itself, you have to navigate through multiple links, lengthy blog entries, and pop-up advertisements. Even then, the recipe can be unclear or confusing to follow. One of the reasons we're creating a recipe website is because this is a frustrating experience that many people can relate to. We believe that there’s a better way to share and discover new recipes, one that prioritizes ease of use, community engagement, and high-quality content.

RecipeReel is motivated by the need to make things easier for home cooks to find and make connections with one another. RecipeReel aims to create a community of food lovers who can share their knowledge, expertise, and passion for cooking in a supportive and engaging environment. It provides users with an opportunity to rate and comment on recipes, giving them a possibility to help each other find the best recipes for their needs. Users can upload photos, cooking instructions, and ingredients, making it easy for other users to recreate the recipe at home. Recipes can be organized by categories such as cuisine type, and dietary restrictions,and it allows users to find the perfect recipe for any occasion.

We believe that RecipeReel has the potential to become a leading platform in the food industry, facilitating the sharing of recipes and the exchange of cooking tips and ideas. We aim to create a community of food lovers who are passionate about exploring and creating new dishes, and who value the connections, and relationships that can be formed through food.

2. Use Cases

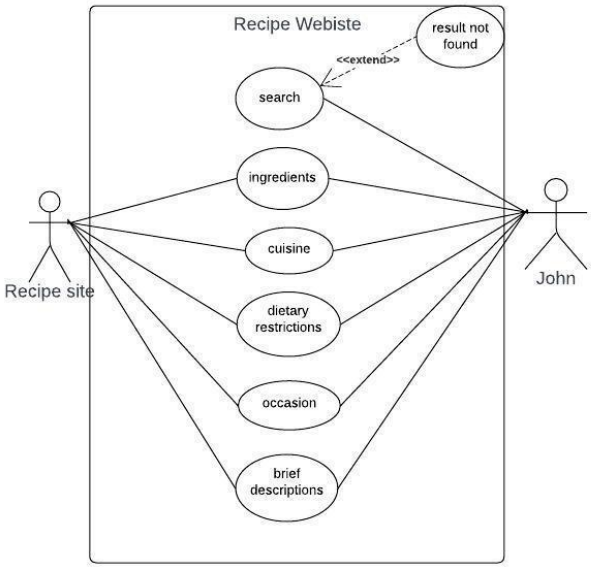
2.1 Dining out is too expensive for a birthday meal

Actor: John, student

Description:

John is a college student at SFSU, he is overwhelmed because his birthday is coming up soon and he was planning to celebrate by going out for a nice meal with friends. However, he has a tight budget, dining out can be a luxury that he as a college student simply cannot afford. To make the most of his special day, he decided to search online for recipes and cook a nice meal to host his friends. He has been browsing through different websites and social media platforms, looking for recipes that are easy to follow and use ingredients that are readily available and affordable.

RecipeReel allows John to search for recipes based on various criteria, such as ingredients, cuisine, dietary restrictions, or occasion. The website provides search results with recipe names and brief descriptions that match the user's search criteria. After browsing several websites, John finally settles on a recipe that looks delicious and within his budget. He makes a list of all the ingredients he needs, heads to the grocery store, and begins preparing the meal. As he cooks, he realizes that he is having a great time and feels a sense of accomplishment knowing that he is preparing his own birthday meal.



(Diagram 2.1, Use Case)

2.2 Bored of local foods

Actor: General User (Mark): Searching to find new foods to try.

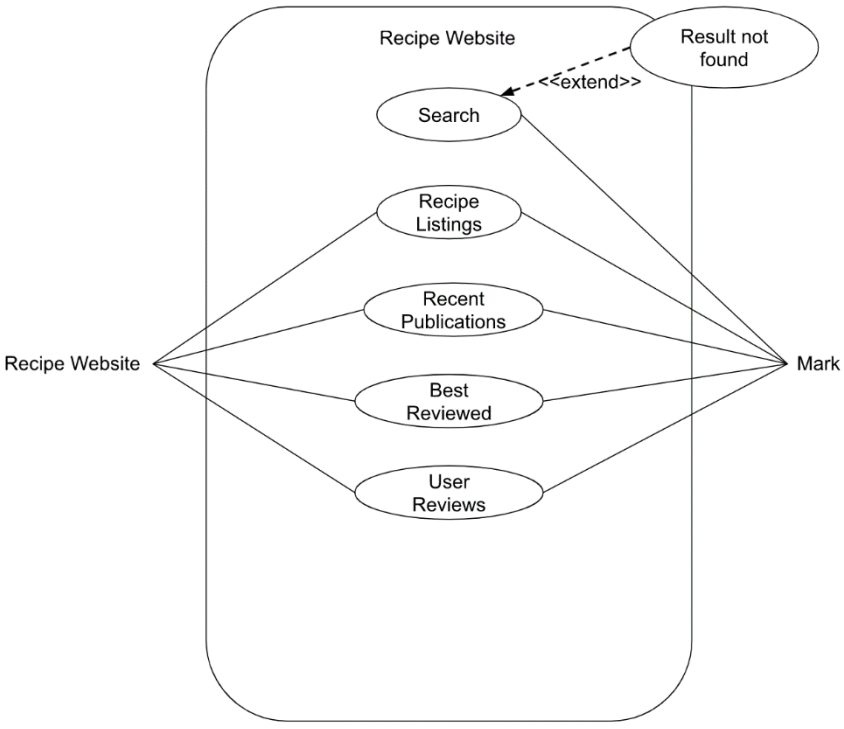
Description:

Mark is an introverted fast-food employee who barely makes above minimuM wage. He often works in the kitchen due to him not being comfortable enough to work the register and interact with customers he knows nothing about. Though he hopes to find a better paying job, he finds himself eating from the restaurant for breakfast and lunch, and then taking home leftovers for dinner to save money while he searches for a career he is interested in pursuing.

As time goes on though, he has found himself getting tired, even loathing the idea of having to eat another meal from his workplace. The unfortunate fact of the matter though is that Mark lives in a small rural town, and the only local restaurants seem to either be other fast-food establishments, or they simply sell higher quality versions of what his fast-food establishment already provides. He needs a change of pace; he wants to taste food from somewhere he would be unable to visit with the income he is currently making. As a result, Mark dips a little into his savings in order to treat himself and starts to research recipes from RecipeReel, where he scours

the new recipe section, as well as the top-rated recipe section in hopes of finding something to satiate his hunger for something new.

The new section of RecipeReel allows Mark to easily search through the website's listing of recipes by latest publications. Though because of how new they are, he may find himself worried about the quality of the recipes and if they are even worth the time and money, he puts in to try them out. To circumvent that, there is also the top-rated section, where Mark can see a catalog of the best rated recipes the website has to offer. Being able to make an educated guess on the quality of a recipe, as well as the consensus of the taste of the meal itself allows for him to feel more at ease when putting his hard-earned money into buying the necessary ingredients to cook it himself. With these tools at Mark's disposal, he will surely be able to find something to prepare for himself that he would be unable to otherwise find in his small rural town.



(Diagram 2.2, Use Case)

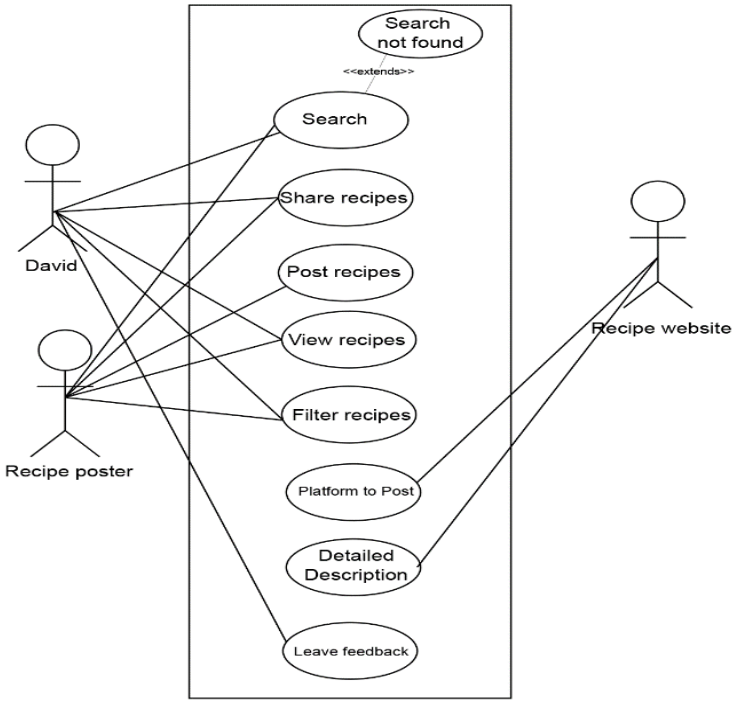
2.3 Cook with what you have

Actor: Construction worker (David)

Description:

David works overtime at a construction company, and most of his work involves hard labor work. After a long working day, David must do groceries on his way home and must prepare his own meal. Since David doesn't have much experience on how to cook, he rather makes quick similar meals that he eats every night. Even though he made a wise decision to make his meal at home and he saves a lot of money from spending unnecessary expenses at restaurants, he always wants to eat good and healthy cuisine from all around the world. If only David knows the food ingredients that he has at home can make different types of dishes from all around the world.

RecipeReel is a great social platform that allows users to share their favorite meals with enough details on how to cook them. Once any user finishes cooking their favorite meal, it gives a platform to post a picture of their masterpiece and lets them share the process of making it, so the whole world can appreciate their art. Now David shall scroll RecipeReel and learn how to make different meals from different countries, with unique flavors with little of what he has at home just by filtering his ingredients.



(Diagram 2.3, Use Case)

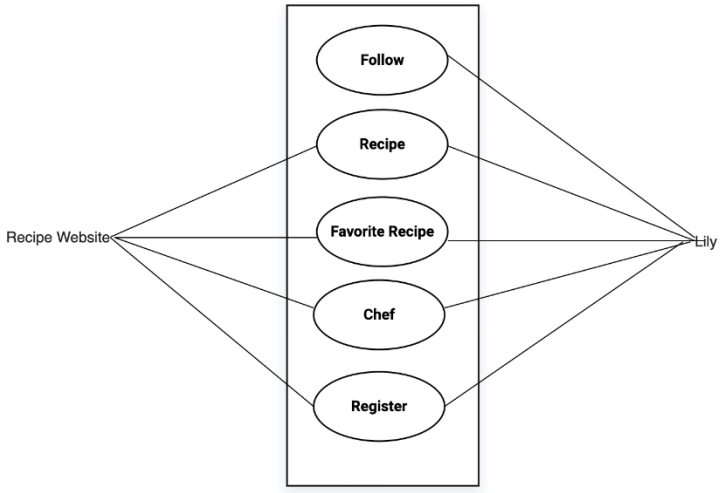
2.4 Home cooks can follow their favorite users

Actor: Lily, Mary

Description:

Lily is an avid home cook who loves everything about food. She loves to go out and eat. With the current state of the economy, going out is a luxury that is becoming less affordable for her. When she can afford to go out, she goes to her favorite restaurant. As her meal progresses, she is loving the food. She tells her waitress to compliment the chef. After the waitress comes back, she tells Lily the executive chef, Mary, is using RecipeReel to share some recipes of the most popular items on the menu. This includes some of the entree that Lily ordered tonight. Lily is determined to eat this well at least once a week. She downloads the app, creates a new account, follows Mary, and learns to cook the entree that she had that night. She learns that chefs from some of her other favorite restaurants are also using RecipeReel and she follows them and learns to cook some more of her favorite dishes.

Lily now eats well affordably; she can make her own spin of dishes that she has had at some of her favorite restaurants. She still goes out to eat occasionally, but she eats something that she can't cook at home. She is happy that she is self-sufficient and can cook tasty food. She even hosts dinner parties now.



(Diagram 2.4, Use Case)

2.5 Homestay owner cooks a variety food for international students

Actor: Lucy Kim, Yu-Jun (Korean student), Márcia (Brazilian student)

Description:

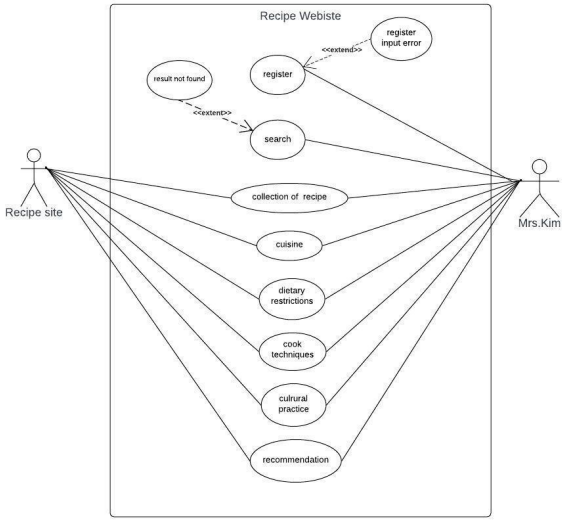
Mrs. Kim is a homestay owner who lives in the United States. She loves taking care of international students who come to study in the country. For Mrs. Kim, it was more than just a job - it was a passion. One of the things that Mrs. Kim takes pride in is providing delicious meals for her students. She knew that food was an essential part of every culture, and she wanted her students to experience the best of American cuisine, as well as the dishes from their own countries.

Mrs. Kim's homestay hosted students from all over the world - China, Korea, Japan, Brazil, and many more. Every day, she would make different meals to cater to the diverse preferences of her students. She was always on the lookout for new recipes and ideas to make their meals more exciting and authentic. To make sure she was creating the best meals for her students, Mrs. Kim would often go online and research popular foods from each student's

country. She would watch cooking videos, read blogs and recipes, and even connect with local people to learn more about the authentic dish

RecipeReel was the website that caught her fancy that offers a vast collection of recipes, all categorized by cuisine, dietary restrictions, and cooking techniques, Mrs. Kim registered an account, as she searched a variety of food, she got a lot recommendation of food from Flavor Frenzy, and she learned about new cooking techniques, and cultural practices. She cooked a variety of Korean dishes, such as kimchi stew, bibimbap, and bulgogi, which Yu-Jun loved. Mrs. Kim also cooked feijoada, churrasco, and brigadiers, which Márcia felt like her mom 's food. Mrs. Kim realized that food was not just about satisfying hunger, but also about creating connections and memories.

As Mrs. Kim tucked her students into bed that night, she knew that she had done her job well. She had not only fed their stomachs but also their souls. Her students were happy, and so was she. She was grateful for the opportunity to make a difference in their lives, one meal at a time.



(Diagram 2.5, Use Case)

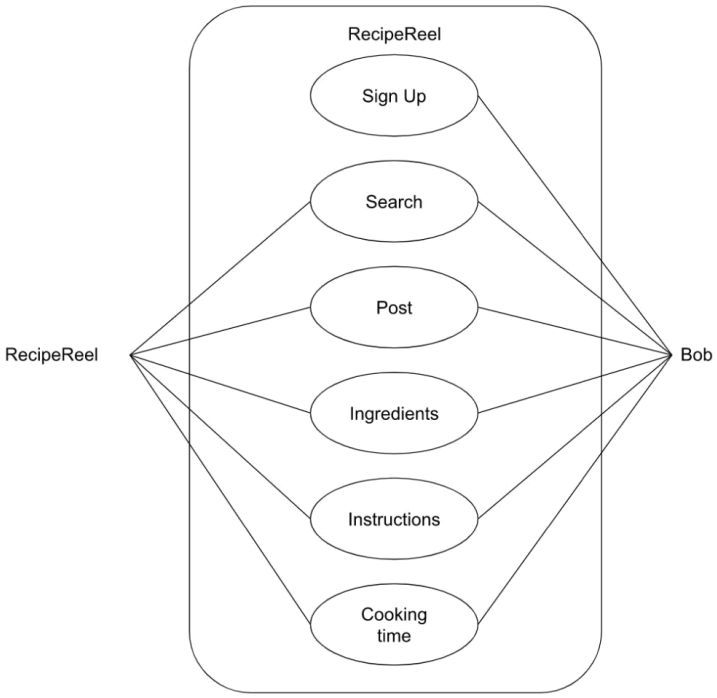
2.6 Share Special Ingredients with the community

Actor: Dan

Description:

Dan is an African citizen currently living in America. He has been preparing meals for himself using a specific ingredient from his own country that gives his dishes a very delicious flavor. However, he recently discovered that many American citizens are not familiar with this ingredient and do not know how to use it in their cooking. As a passionate cook and food enthusiast, Sam wants to share his knowledge and expertise with others by creating a post on this food sharing platform, explaining what the special ingredient is, how it can be used in different dishes, and where to buy it. He wants to create a complete guide that will help others discover the magic of this ingredient and use it into their own cooking. Additionally, he hopes to start a discussion on the website where others can share their own tips and recipes for using the ingredient, creating a community of cooks who are passionate about exploring new flavors and ingredients.

Also, by sharing this special ingredient with the community, he hopes to introduce more people to the ingredient and traditions of his home country, and to promote greater understanding and appreciation between different cultures.



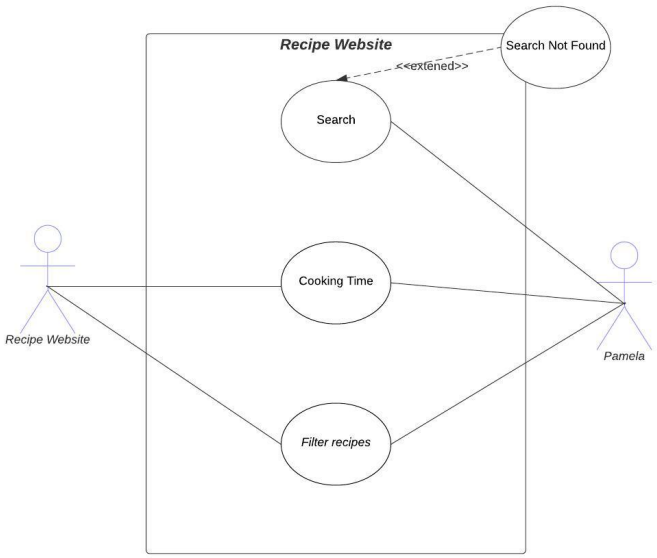
(Diagram 2.6, Use Case)

2.7 Finding recipes based on cooking time

Actor: A working mom (Pamela)

Description:

Pamela is an event planner, and her job requires her to always be on the go. She is also a mom of two toddlers. Her job requires her to be active throughout the day, sometimes even during odd timings. She often doesn't find the time to cook and relies on takeouts, as it's hard to choose what to cook and find the time out of her busy schedule. She wants to find recipes that will help her cook quick and healthy meals. Hence she found RecipeReel, where she can search for recipes and view the cook times of each for each meal she finds interesting. She can scroll through and look for recipes that fit her schedule.



(Diagram 2.7, Use Case)

2.8 Recipes based on dietary restrictions

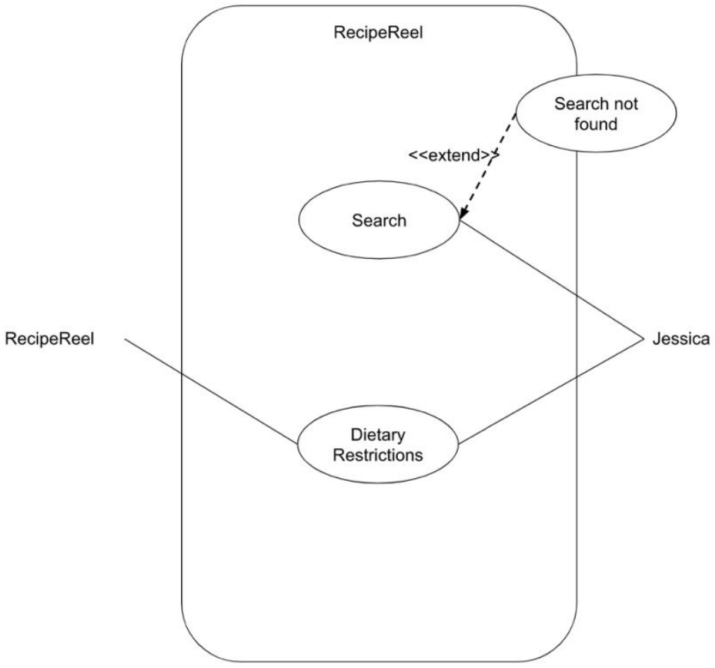
Actor: Christine, Jessica (mom)

Description:

Christine is a 9th grade student, who spends more than half the day at school. She is a track student and is part of debates in her school business club. While at school, she opts to get food from the cafeteria, which is often not a healthy choice. Currently Christine has been advised to follow some dietary restrictions by her doctor. And Jessica, her mother, has no clue what and how to make meals that follow Christine's dietary restrictions. Hence she decides to try different recipes at home with the help of RecipeReel.

On RecipeReel, Jessica uses the categories tab to look up recipes based on dietary requirements, she also looks for recipes that are healthy and would help Christine get all the

nutrients she needs to get through her food.



(Diagram 2.8, Use Case)

3. List of main data items and entities

- Search: It is an entity that allows users to search for recipes using specific keywords. These keywords can be (but not limited to) ingredients, users, or cultural cuisine. General users have access to search because we want them to be able to look up recipes that they would like to cook.
- Ingredients: It is an entity that users shall search for recipes based on the ingredients they have on hand. The ingredients entity shall include various items that are used in cooking recipes, such as chicken, fish, pork, beef, vegetables, and others. These ingredients can be represented as an enumeration or a list and the search shall return recipes that can be made with those ingredients.
- Cuisine: It is an entity that users can use to search for recipes based on specific cuisines. Users can select the cuisine they are interested in, such as American, African, Mexican, or Asian and the search shall return recipes that fall under the cuisine type.
- Dietary Restrictions: It is an entity that users can use to find recipes based on diet specifications. This will allow users to have an option to find their favorite recipes based on their diet restrictions. Such as keto, gluten-free, vegan, dairy free, halal,etc.,. and the search shall return based on their restrictions.
- Occasion: It is an entity that users shall search for recipes based on the occasion or event. Users can select the type of occasion they are preparing for, such as a holidays, Birthday party, and the search shall return all recipes that are suitable or categorized for that occasion.

- Brief Description: It is an entity that will provide the user with a short summary of the dish that highlights its main ingredients and background. Brief descriptions also interact with other entities such as, allergy or eating restrictions
- Follow: It is an entity that refers to a registered user to follow another user's content on the website. When a user follows another user can view the updates about the new contents that the following user has posted. The purpose of this entity is to build connections between users, help users to find similar interests in recipes and discover new recipes and content with other users.
- Unfollowing: It is an entity that allows a registered user to unfollow certain users. The purpose of this entity is to allow users to see only recipes from specific users that they are interested to follow.
- Recipe: Recipe is an entity that will provide users with information about a specific recipe. It includes other entities such as, recipe title, which is the name of the dish; A brief description of this recipe; Ingredient list which informs the user all the ingredients needed to make the dish; Cooking time, which tells the user the amount of time required to cook the dish; Recipe images that user posted, and comments from other users etc.
- Post: It is an entity that allows a registered user to post personal recipes, allowing them to help spread new varieties and food ideas. The post contains 1-6 images per post and with a maximum 2000 characters.
- Category: It is an entity that shall allow users to easily navigate and filter through the available recipes based on their preferences. It is an entity that shall be used as a container for organizing and grouping various sub-data items such as cuisine, ingredient, occasion, and dietary restriction. The Category entity will provide an efficient and useful way for users to find the recipes they are looking for.

- Latest Post: It is an entity that tracks and stores when a user publishes a new post. This entity may be used to notify other users with the recent post and a message shall be displayed on the notification bar. Additionally, if the recent publisher is in the best reviewed publisher category, they shall automatically be recommended to several users on their feed. Recent publications shall also be used on users' feeds regardless of their rating, so new and unrecognized users can get a chance to share their cuisine based on this entity.
- Top Rated: It is an entity that stores users ratings and feedback on its database, then it shall manage to categorize among best reviewed, good received or bad reviewed. By doing this, it will benefit the admin on recommending other users based on their reviews. The best reviewed recipe shall be noticeable and visible by all users so other users can benefit from its best reviewed recipes. Our database system shall be able to store and process user data and use it and use that data to recommend on the home page. To access the best reviewed cousins, our system shall have a system that will help to categorize based on their rating. By searching or filtering users can easily access the best reviewed cuisine.
- Register : It is an entity that allows a general user to register/sign up for an account so they can access more functions of the site. The purpose of this entity is to allow a general user to create an account and start using the social features of the site.
- Cooking time: Cooking time is an entity that will provide users with an estimation of the total time required to cook a specific dish. The purpose of this entity is to allow users to plan their meal preparation and manage their time. The format of this entity will be displayed as hours and minutes. Cooking time also interacts with other entities, such as

preparation time, which tells the user the amount of time required for preparing the dish and resting time for how long to let the food sit before eating.

- Instructions: Instructions are an entity that will provide the user with step-by-step guidance on how to prepare and cook a dish. Instructions also interact with other entities, such as ingredient list which informs the user all the ingredients needed to make the dish and the cooking method which informs the user if they need a grill or stove to cook a certain meal.

4. Initial list of functional requirements

4.1 Functional requirements for general users

4.1.1 A general user shall be able to register.

4.1.2 A general user shall be able to register using a phone number.

4.1.3 A general user shall be able to verify phone numbers.

4.1.4 A general user shall be able to verify using 2FA.

4.1.5 A general user shall be able to search.

4.1.6 A general user shall be able to scroll through posts in a feed like manner.

4.1.7 A general user shall be able to search the website by newest.

4.1.8 A general user shall be able to search the website by best reviewed.

4.1.9 A general user shall be able to search different categories of food.

4.1.10 A general user shall be able to search for recipes based on an ingredient.

4.1.11 A general user shall be able to search for recipes based on dietary restrictions.

4.1.12 A general user shall be able to search for recipes based on a max calorie count.

4.1.13 A general user shall be able to search for recipes based on cooking time.

4.1.14 A general user shall be able to sort search based on lowest or highest calorie count.

4.1.15 A general user shall be able to search for recipes by cuisine type.

4.1.16 A general user shall be able to view recipe details such as ingredients.

4.1.17 A general user shall be able to view recipe details such as cooking time.

4.1.18 A general user shall be able to view recipe details such as difficulty level.

4.2 Functional requirements for registered users

4.2.1 A registered user shall be able to log in.

4.2.2 A registered user shall be able to log out.

4.2.3 A registered user shall be able to edit a profile.

4.2.4 A registered user shall be able to share a post of recipe.

4.2.5 A registered user shall be able to get recommendations when they are not sure what to cook.

4.2.6 A registered user shall be able to download the recipes from the website.

4.2.7 A registered user shall be able to filter by ingredients.

4.2.8 A registered user shall be able to search by ingredients.

4.2.9 A registered user shall be able to filter by dietary restrictions.

4.2.10 A registered user shall be able to change their password.

4.2.11 A registered user shall be able to change their email.

4.2.12 A registered user shall be able to delete their post.

4.2.13 A registered user shall be able to archive their post.

4.2.14 A registered user shall be able to not see promotions.

4.2.15 A registered user shall be able to view recipe details such as ingredients.

4.2.16 A registered user shall be able to view recipe details such as preparation time.

4.2.17 A registered user shall be able to view recipe details such as difficulty level.

4.2.18 A registered user shall be able to follow other users to receive updates and see their recipe collections.

4.2.19 A registered user shall be able to leave comments or feedback on recipes to ask questions or provide suggestions.

4.2.20 A registered user shall be able to rate recipes.

4.3 Functional requirements for admins

4.3.1 An admin shall be able to verify using email.

4.3.2 An admin shall be able to verify using password.

4.3.3 An admin shall be able to verify using a phone number.

4.3.4 An admin shall be able to verify using Two - factor authentication.

4.3.5 An admin shall be able to get a new password if forgotten.

4.3.6 An admin shall be able to change passwords.

4.3.7 An admin shall not be able to change email.

4.3.8 An admin shall be able to log in.

4.3.9 An admin shall be able to log out.

4.3.10 An admin shall be able to share a post of a recipe.

4.3.11 An admin shall be able to delete a post of a recipe.

4.3.12 An admin shall be able to save a post.

4.3.13 An admin shall be able to like a post.

4.3.14 An admin shall be able to dislike a post.

4.3.15 An admin shall be able to leave reviews on recipes.

4.3.16 An admin should be able to follow other authenticated users.

4.3.17 An admin shall be able to change their password.

4.3.18 An admin shall be able to change their linked email address.

4.3.19 An admin shall be able to change their usernames.

4.3.20 An admin shall be able to change their pass.

4.3.21 An admin shall be able to filter recipes by who they're following.

5. List of non- functional requirements

5.1 System requirements

5.1.1 The website shall be able to use client – server.

5.1.2 The website shall be able to handle when visitor scale is high.

5.1.3 The website shall be able to support both cellular and Wi-Fi networks.

5.1.4 The website shall be able to support any user using any web browser.

5.1.5 The database host shall be able to have a web server to host.

5.1.6 The database host network shall be able to have good security.

5.2 Performance requirements

5.2.1 The website reload shall be able to provide fast service.

5.2.2 The website shall be able to have minified code to reduce file size.

5.2.3 The website shall be able to optimize image quick.

5.2.4 The website shall be able to use a performance monitoring tool to catch bottlenecks.

5.2.5 The website database server shall be able to expect multiple users without any delay.

5.2.6 The website database server shall be able to expect multiple requests.

5.3 Privacy

5.3.1 The website shall be able to provide clear and concise policies.

5.3.2 Users shall be able to agree on how their openly information may be shared.

5.3.3 The website shall be able to collect data like name, date of birth, email, and more personal information.

5.3.4 The website shall be able to avoid collecting unnecessary data.

5.3.5 The collected data shall be able to be used for improvement of the website and user experiences.

5.3.6 All confidential user data shall be able store encrypted data.

5.3.7 For more protection firewalls and protection systems shall be able be used.

5.3.8 Users shall be able to be trained not to share sensitive information.

5.4 Storage

5.4.1 The website shall be able to have a good amount of storage that can handle contests like food images, food recipes, food description and more.

5.4.2 The website shall be able to support all file formats and sizes for better and effective performance.

5.4.3 The website shall be able to have a limit on the maximum file size that can be uploaded to prevent excessive resource usage.

5.4.4 The website shall be able to have another storage for emergency backups.

5.5 Security

5.5.1 The website shall be able to use HTTPS. (Secure Transfer protocol).

5.5.2 The website shall be able to implement security measures to protect user data.

5.5.3 The website shall be able to have a strong password policy.

5.5.4 The website shall be able to use a password transcription method.

5.5.5 The website shall be able to offer multi-factor authentication to prevent unauthorized access.

5.6 Marketing and Legal requirements

5.6.1 Popular social media sites shall be able to be used to promote websites.

5.6.2 The website shall be able to be user friendly so any user can browse comfortably.

5.6.3 The information provided on the website shall be able to be clear for anyone to understand.

5.6.4 The website shall be able to select a specific target audience for marketing.

5.6.5 Links of the website shall be able to be pushed through emails and other forms.

5.6.6 The website shall be able to be branded.

5.6.7 The website shall be able to have a name and logo.

5.6.8 The website shall be able to accept feedback.

5.6.9 The website shall be able to protect itself-from any legal claims.

5.7 Content

5.7.1 The website will be using a clean and simple sans-serif font as it is easy to read and doesn't detract from the content.

5.7.2 The website will be using a clear hierarchy for headings, subheadings, and other important information. Additionally, using bullet points or numbered lists for ingredients and instructions can make the recipe easier to follow.

6. Competitive Analysis

6.1 Competitors analysis

Feature/Company	Competitor 1 Sidechef	Competitor 2 Yummly	Competitor 3 nytime.com	Competitor 4 BigOven	Competitor 5 epicurious
Weaknesses	-Recipes shared by company itself -Most features are available only for premium users	- Must create an account for better use. It is not a social media site for food. Users don't share their own recipes. original recipe made by special chefs only. has a lot of options and tools. And this can be confusing to navigate.	-Not usable unless paid for	-Advertisements -Must create an account to access some of the basic stuff	- Can't be sorted by author - Unlimited usage locked behind paywall

Strengths	-Step by step guided tutorials -Allows you to search based on dietary restrictions, occasion, cuisine, meal type, etc. -Partnered with Walmart for grocery shopping & delivery -Meal planner	- The website is easy to navigate. -It has a lot of options and tools. -It has a vast collection of recipes. -It allows users to search for recipes based on (ingredients, cuisine, cooking time). -It has extra gadgets for sale. (Smart Thermometer). -Helps shopping and partnered with delivery services.	-Premium users get curated recipes, suggestions, and a digital space to store their favorite recipes	-Find recipes to use your leftovers and avoid wastage -Smooth user-interface	- Easy to navigate user interface - Above and beyond ways to search for recipes - Allows you to save recipes
Pricing	Basic: Free Premium: $4.99/month or $49.99/year	$4.99/month	Premium: $5/month or $40/yr.	Basic: Free Premium: $2.99/month or $24.99/year.	Premium: $5 / month $40 / year
Social Media	YouTube, Pinterest, twitter, Instagram, LinkedIn, Facebook	It is available in every social media.	YouTube, Pinterest, twitter, Instagram, LinkedIn, Facebook	YouTube, Pinterest, twitter, Instagram, LinkedIn,	Facebook, Twitter, Instagram, Pinterest, YouTube, Google

				Facebook, TikTok	+, Tumblr, RRS feeds
Onboarding Experience	Side Chef is a very user-friendly website and app that not only helps you find recipes of your choice but also helps do your grocery shopping for the meal. There are lots of filters to choose from to find your favorite meal plan.	Yummly is a cool web-app. That allows users to create an account for better use. Users are also allowed to select their dietary preferences and cooking skills, to personalize the user experience and recommend relevant recipes. It is an app where people look at food based on their needs, find, and cook meals prepared by professional chefs.	There aren't many options because it is a subscription service. Subscribers can access the digital cookbook and culinary guide across all platforms, which aid home cooks of all skill levels in finding, saving, and organizing the best recipes in the world.	Big Oven is a user-friendly website and app that allows you to take your recipes with you wherever you go, create grocery lists, cut down on food waste, and quickly share your favorite dishes with friends and family.	Epicurious is user-friendly, giving users a plethora of search options and an easy to navigate user interface. The recipes themselves provide an approximate time to prepare the meal, as well as a clear list of ingredients, and step by step instructions for users to follow.

6.2 Competitive feature analysis

Feature	Competitive 1	Competitive 2	Competitive 3	Competitive 4	Our feature
Text search	+	+	+	+	+
browse	+	+	+	+	+
post	+	-	-	-	+
comment	+	+	+	+	+
download	-	-	-	-	+

Note: + feature exists, ++ superior, - doesn't exist

7. System architecture and technologies

- Server Host: AWS EC2 1 vCPU 2GiB mem
- Operating System: Ubuntu Server 22.04 LTS
- Database: PostgreSQL 14.6
- Web Server: NGINX
- Server-Side Language: JavaScript / Node
- Web Framework: React
- IDE: VS studio,
- Web Analytics: Google Analytics
- SSL Cert: Lets Encrypt (Cert Bot) SASS: 3.

8. Checklist

- Team found a time slot to meet outside of the class GitHub master chosen – Done
- Team decided and agreed together on using the listed SW tools and deployment server – Done
- Team ready and able to use the chosen back and front-end frameworks and those who need to learn are working on learning and practicing – Done
- Team lead ensured that all team members read the final M1 and agree/ understand it before submission – Done
- GitHub is organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.) — Done
- Executive Summary – Done
- Functional requirements – Done
- Team members finished their assigned Non-Functional Requirements – Done
    a. System requirements – Done
    b. Performance requirements - Done
    c. Storage, security, environmental requirements - Done
    d. Privacy – Done
    e. Marketing and legal requirements – Done
    f. Content –Done
- Team chooses a name for the project – Done
- Team members finished their competitor analysis – Done
- Team members finished their introduction on About us page - Done

9. Team member contributions

Team member	Contributions/Activities	Contribution score
Team lead: Yueling Liu	● Executive Summary ● User case and use cases' diagram ● Functional requirements ● Privacy and content non-functional requirements ● Data description for general user ● Data description for authenticated user, author, admin, public photos, items ● Edited and reviewed documentation ● Set up meetings, assigned tasks to each team members ● Revised documentation	
Backend lead: Duncan Herington	● Executive Summary ● User case ● Defined functional requirements ● Defined non-functional requirements for performance requirements ● Competitive feature analysis ● Assisted front-end of About us page	7/10

Marcel Azouri	• Use case • use case diagram • Defined entities • Defined functional requirements • Defined non-functional requirements • Contributed one competitor analysis • Proofread documentation and provided feedbacks	7/10
Priya Pradeep	• Contributed two use cases • Defined functional requirements • Contributed three competitor analysis • Helped revise M1V2 • Redefined two use case	6/10
Nathan Le Howland	• Use case • Use case diagram, • Defined entities • Defined non-functional requirements • Chose techniques stacks • Set up servers • Creating GitHub branches	7/10
Samuel Elias	• Use case • Defined functional requirements • Storage description, • Defined non-functional requirements • Contribute one competitor analysis	5

Yasson Haddish	● Use case ● Defined functional requirements ● Defined non-functional requirements for System requirement, Marketing and Legal ● Editing and revising documentation	5

SW Engineering CSC 648-05 Spring 2023

RecipeReel

T03 Milestone 2

Team lead: Yueling Liu yliu50@sfsu.edu

Backend lead: Duncan Herington, Marcel Azouri

Frontend lead: Yasson Haddish

GitHub lead: Nathan Le Howland

Database lead: Samuel Elias

Document lead: Priya Pradeep

History Table

M2V2	April 27, 2023
M2V1	April 3, 2023
M1V2	April 3, 2023
M1V1	March 2, 2023

Table of Contents

1. Data Definitions

1. General user: This entity is a user that accesses the site with no registered account. A general user will have access to a limited selection of features of the site. A general user is allowed to browse and search recipes but cannot interact with posts such as leaving a comment, saving a post, or giving a like to a post.

2. Register an account: It is an entity that allows a general user to register/sign up for an account so they can access more functions of the site. The purpose of this entity is to allow a general user to create an account and start using the social features of the site.

3. Search: It is an entity that allows users to search for recipes by specific keywords. These keywords can be (but not limited to) ingredients, users, or cultural cuisine. General users have access to search because we want them to be able to look up recipes that they would like to cook.

4. Registered user: It is an entity that created an account and can sign in to the website from different devices. This user will have a unique email and a password.

    4.1 Username

    It is an entity that is typically a string of characters that was chosen by the user to identify themselves. The length of the username shall be limited to a minimum of 6 characters and a maximum of 20 characters. The username is required when logging in on the website. It's paired with a password.

    4.2 Password

    It is an entity that refers to a combination of characters, strings, numbers, or letters that was chosen by users to secure their account and prevent unauthorized access. The password shall be a minimum of 6 characters and a maximum of 20 characters long and it must have a special character. This entity is required when logging into the website. When both the username and password are correctly entered, the user will be granted access to their account.

    4.3 Following

    It is an entity that refers to a registered user who follows another user's content on the website. When a user follows another user, they can view the updates about the new contents that the following user has posted. The purpose of this entity is to build connections between users, help users find similar interests in recipes, and discover new recipes and content with other users.

4.4 Unfollowing

It is an entity that allows a registered user to unfollow certain users. The purpose of this entity is to allow users to see only recipes from specific users that they are interested in following.

4.5 Posts

It is an entity that allows a registered user to post personal recipes, allowing them to help spread new varieties and food ideas. The post contains 1-6 images per post, with a maximum of 2000 characters.

4.6 Delete posts

It is an entity that allows a registered user to delete or remove their own post or the content of the recipe. When a user deletes a recipe post, the content will be removed from the website, and all the comments from other users are no longer visible on the website. The purpose of this entity is to give users the ability to manage their accounts. It also can be helpful if a user accidentally posts something; they can quickly delete it by being seen by other users.

4.7 Like

It is an entity that allows users to express their positive reaction or appreciation for a particular recipe post. The purpose of a “like” entity is to provide a simple and quick way for users to show their support or approval without having to write a comment or engage in a more extended conversation. This also allows users to see how popular or well-received a particular recipe is.

4.8 Dislike

It is an entity that allows users to express negative sentiment or dissatisfaction. The purpose of a “dislike” entity is to provide users with a way to give feedback on a recipe that they don’t enjoy or find useful without having to write a comment or engage in an extended conversation.

4.9 Comment

It is an entity that allows users to engage in a conversation or provide feedback on a particular recipe post. Users can write a text-based response, which they can then share publicly for other users to see and respond to. This can be used to communicate thoughts, views, or ideas in a comment entity in a more extensive and detailed manner, as well as to interact with other users.

4.10 Rate

It is an entity that allows users to provide a way to express their opinions and experiences with a particular recipe and help other users make informed decisions. The rating would help provide feedback to the particular user who posted the recipe and help other users decide whether or not to try it out.

5. Top Rated**:** Top rated is an entity that will provide users with ratings based on several factors such as the quality and uniqueness of the recipes, the level of detail in the instructions, the variety of recipes offered, and difficulty of cooking.

6. Latest Post**:** Latest post entity for a recipe website would be the most recently published recipe on the website. This would be a new recipe that has just been added to the website's collection. The latest post entity is important as it keeps the website fresh and up-to-date, and provides users with new content to explore and engage with.

7. Recipe **:** Recipe is an entity that will provide users information about a specific recipe. It includes other entities such as, recipe title, which is the name of the dish; A brief description of this recipe; Ingredient list which informs the user all the ingredients needed to make the dish; Cooking time, which tells the user the amount of time required to cook the dish; Recipe images that user posted, and comments from other users etc.

   7.1 Title

   It is an entity that will provide the user with a name for a particular recipe. It provides the user with information or a description that accurately reflects the content of the recipe.

   7.2 Description

   It is an entity that will provide the user with a short summary of the dish that highlights its main ingredients and background. The description may also provide additional details such as serving size, cooking time, and any special equipment or techniques that are needed.

   7.3 Instruction

   Instructions are an entity that will provide the user with step-by-step guidance on how to prepare and cook a dish. Instructions also interact with other entities, such as the ingredient list, which informs the user of all the ingredients needed to make the dish, and the cooking method, which informs the user if they need a grill or stove to cook certain meals.

   7.4 RecipeImages

It is an entity that users post on the website; a registered user can post 1-6 images that are related to the recipe. Some good photographs can also make the recipe more appealing and attractive, which can increase engagement and interest among users.

7.5 Category

It is an entity that shall allow users to easily navigate and filter through the available recipes based on their preferences. It is an entity that serves as a container for organizing and grouping various sub-data items such as cuisine, ingredient, occasion, and dietary restrictions. The category entity will provide an efficient and useful way for users to find the recipes they are looking for.

7.5.1 Dietary restrictions

It is a sub-data item that users can use to find recipes based on diet specifications. This will allow users to have the option of finding their favorite recipes based on their dietary restrictions. Such as keto, gluten-free, vegan, dairy-free, halal, etc., and the search shall return based on their restrictions.

7.5.2 Occasion

It is a sub-data item that users search for recipes based on the occasion or event. Users can select the type of occasion they are preparing for, such as a holiday or birthday party, and the search will return all recipes that are suitable for or categorized for that occasion.

7.5.3 Cuisine

It is a sub-data item that users can use to search for recipes based on specific cuisines. Users can select the cuisine they are interested in, such as American, African, Mexican, or Asian, and the search shall return recipes that fall under the selected cuisine type.

7.6 Ingredients

It is an entity that users shall search for recipes based on the ingredients they have on hand. The ingredients shall include various items that are used in cooking recipes, such as chicken, fish, pork, beef, vegetables, and others. These ingredients can be represented as an enumeration or a list, and the search shall return recipes that can be made with those ingredients.

7.7 Cooking Time

Cooking time is an entity that will provide users with an estimation of the total time required to cook a specific dish. The purpose of this entity is to allow users to plan their meal preparation and manage their time. The format of this entity will be displayed

as hours and minutes. Cooking time also interacts with other entities, such as preparation time, which tells the user the amount of time required for preparing the dish, and resting time, which tells the user how long to let the food sit before eating.

7.8 Difficulty level

Difficulty level is an entity that will provide the user with an idea of how hard the dish will be to make for themselves. By having this feature, it allows the user to find dishes that suit their level of cooking skills.

2. Prioritized Functional Requirements

2.1 Must Have Functionalities

2.1.1 General Users

- A general user shall be able to register for an account of RecipeReel using a username, password and email.
- A general user shall be able to view recipes posted on the website using a feed.
- A general user shall be able to search or filter recipes using predefined categories/cuisines.
- A general user shall be able to search or filter recipes using predefined ingredients.
- A general user shall be able to search using cooking time.
- A general user shall be able to search using a keyword or term that matches a recipe's category (cuisine), ingredient, cooking time, title, and difficulty.
- A general user shall be able to view recipes, details, and comments.

2.1.2 Registered Users

- A registered user shall be able to log in with a username and password.
- A registered user shall be able to save their favorite recipes.
- A registered user shall be able to leave comments on recipes.
- A registered user shall be able to like comments on recipes.
- A registered user shall be able to dislike comments.
- A registered user shall be able to post recipes to ReciperReel.
- A registered user shall be able to delete a recipe they posted to RecipeReel.
- A registered user shall be able to follow other registered users.
- A registered user shall be able to unfollow other registered users.
- A registered user shall be able to log out.
- A registered user shall be able to view a feed of recipes(posts) based on other registered users they follow.
- A registered user shall be able to rate a recipe using a 1-5 scale.

2.1.3 Admin

- An admin shall be able to use tools such as PGAdmin and AWS tools to monitor and update all site contents.
- An admin shall be able to manage website functionalities using source code.
- An admin shall be able to analyze site performance using AWS amplify console.
- Admin shall keep user information and data stored safely using a postgres DB.
- An admin shall be able to manage the amount of storage the website has in order to make sure there's enough to store contents such as images, recipes, descriptions, and more.
- An admin shall manage the amount of storage the website used using AWS RDS console.

2.2 Desired Functionalities

2.2.1 General Users

- A general user shall be able to register using a phone number.
- A general user shall be able to filter by author.
- A general user shall be able to filter recipes by date added.
- A general user shall be able to filter by highest review.
- A general user shall be able to filter based on calorie count.
- A general user shall be able to filter by difficulty level.
- A general user shall be able to search for recipes that do not include ingredients.

2.2.2 Registered Users

- A registered user shall be able to verify with their phone number.
- A registered user shall be able to enable 2FA using a 2FA app.
- A registered user shall be able to edit their profiles.
- A registered user shall be able to download a pdf of a recipe.
- A registered user shall be able to archive their recipes.
- A registered user shall be able to edit their security information.
- A registered user shall be able to save posts.
- A registered user shall be able to edit their recipes.
- A registered user shall be able to delete their recipes.

2.2.3 Admin

- An admin user shall be verified using two-factor authentication.
- An admin user shall be able to verify using their phone number.

2.3 Opportunistic functionalities

2.3.1 General Users

- A user shall have access to a meal planner tool for their meal criteria.

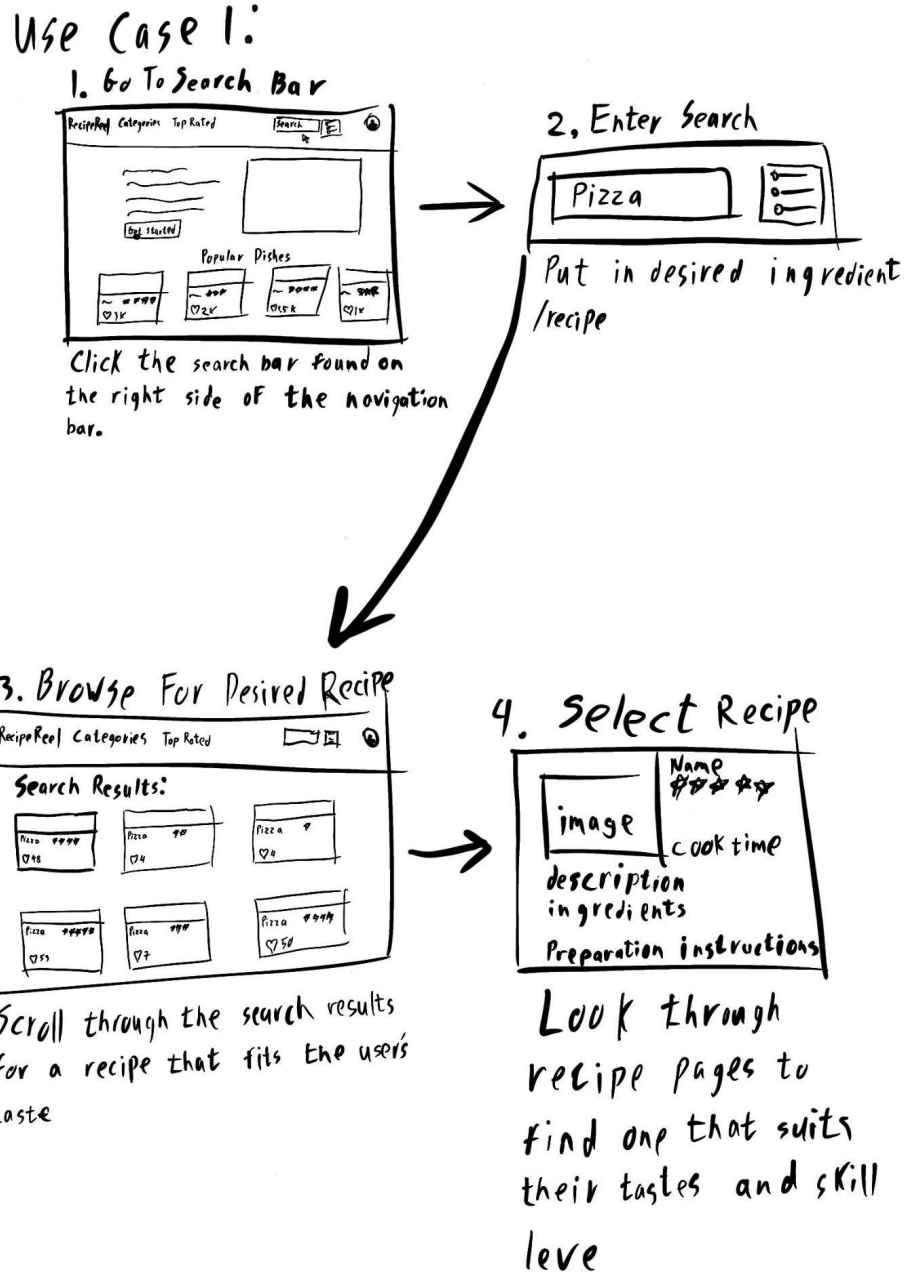
2.3.2 Registered Users

- Registered users shall be able to privately message each other.
- Registered users shall be able to opt for notifications.

2.3.3 Admin

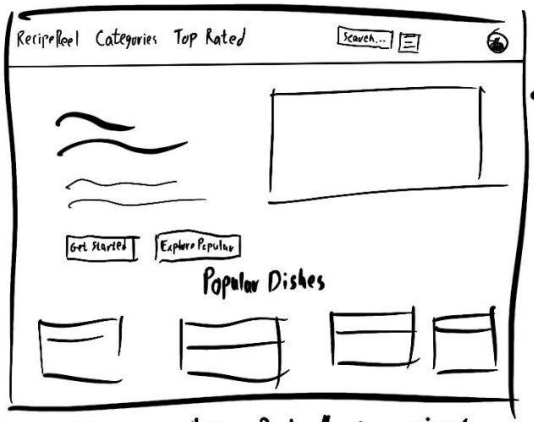
- An admin shall have monitoring tools for tracking system metrics.
- An admin shall have a revenue management tool.

3. UI Mockups and Storyboards



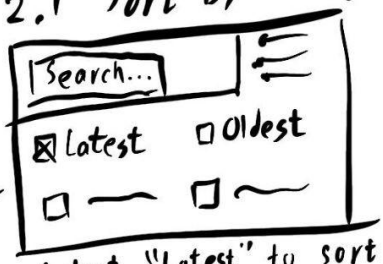
Use Case 2:

1. Go to "Top Rated" or filter



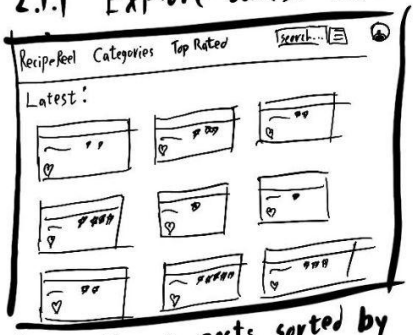
Click on "Top Rate" to view recipes with the best ratings. Click the filter button to view other sorting options

2.1 Sort by "Latest"



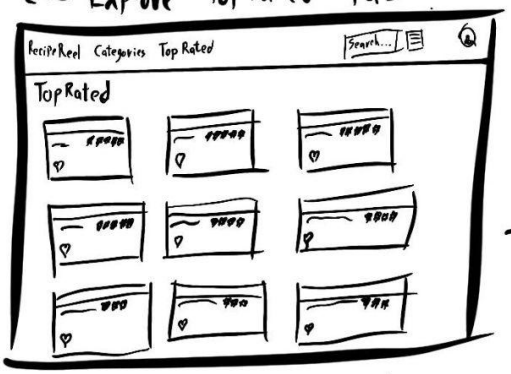
Select "Latest" to sort recipes by the most recent posts.

2.1.1 Explore Latest Posts



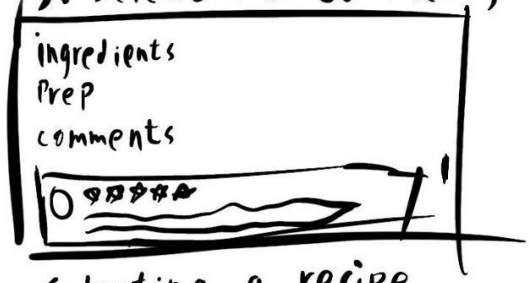
Browse through posts sorted by latest

2.2 Explore "Top Rated" Tab



Browse through top rated recipes on RecipeReel

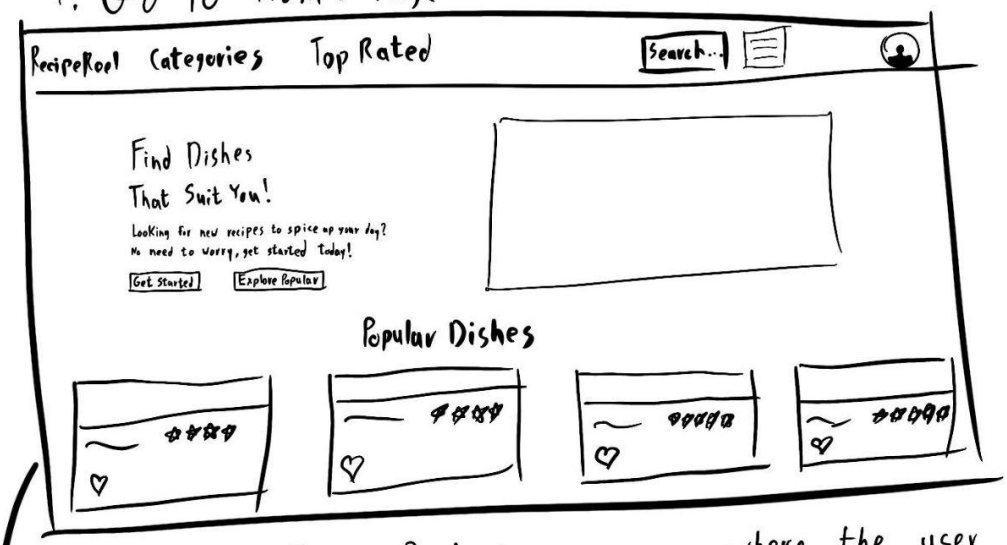
3. select and comments



Selecting a recipe and seeing what others think of it

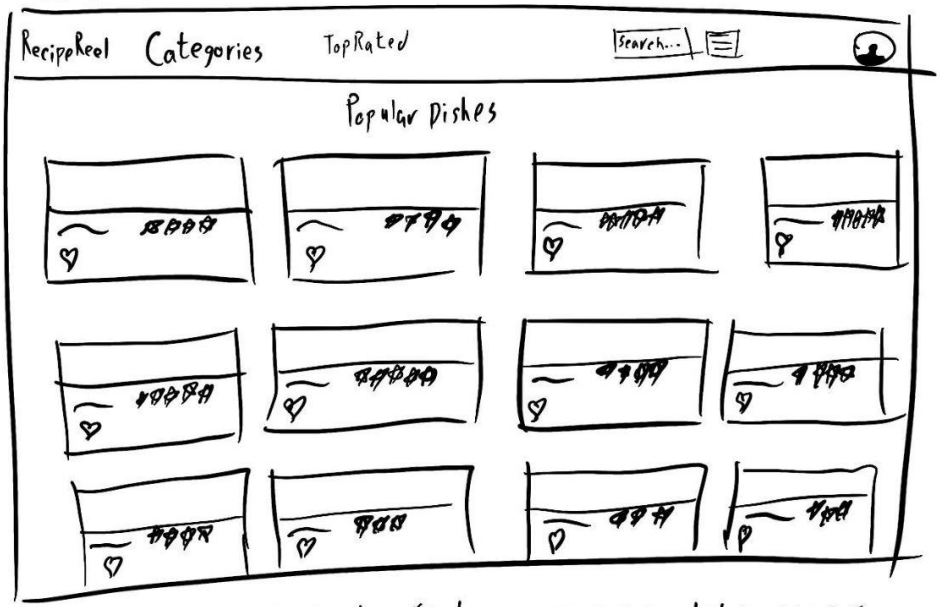
USE CASE 3:

1. Go To Home Page



Access the Recipe Reel home page where the user will be met with popular dishe

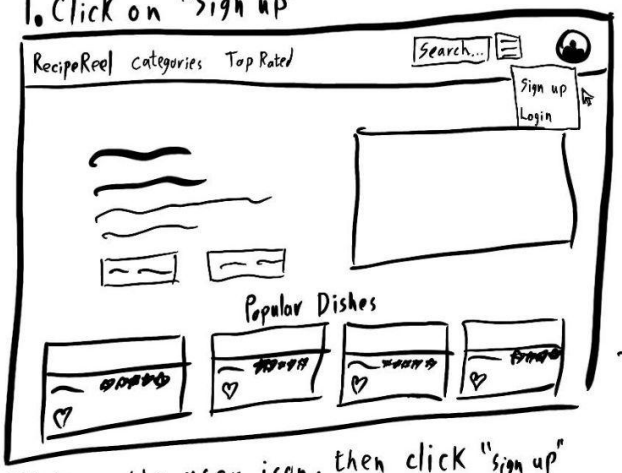
2. Explore



Browse Recipe Reel to find a recipe the user could enjoy

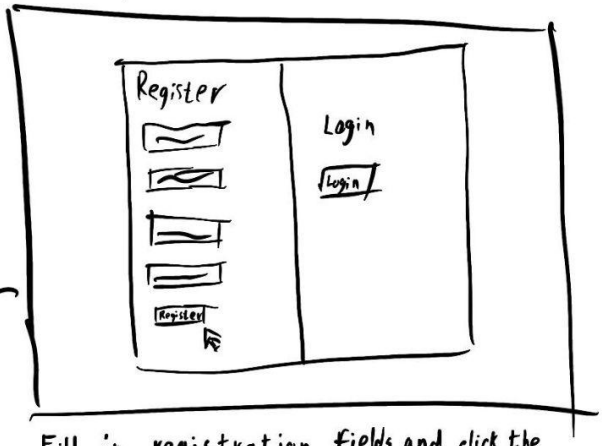
Use Case 4:

1. Click on "Sign up"



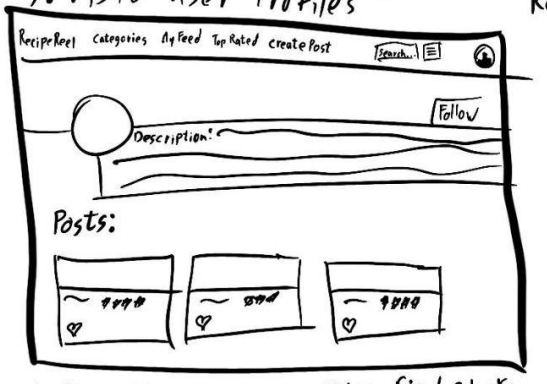
Click on the user icon, then click "sign up" to access the registration page

2. Register



Fill in registration fields and click the "Register" to finalize registering for a RecipeReel account

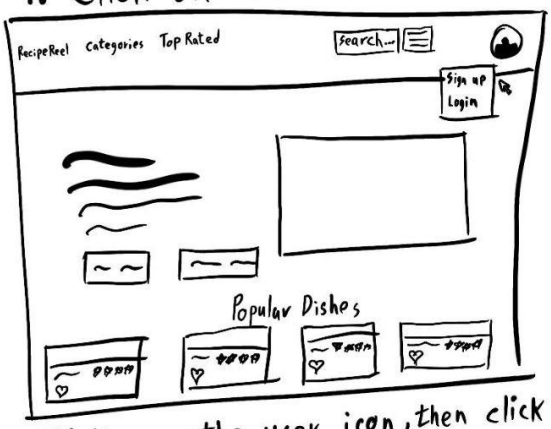
3. Visit User Profiles



Visit other user profiles, find chefs the user enjoys, and follow them to stay up to date on their recipes

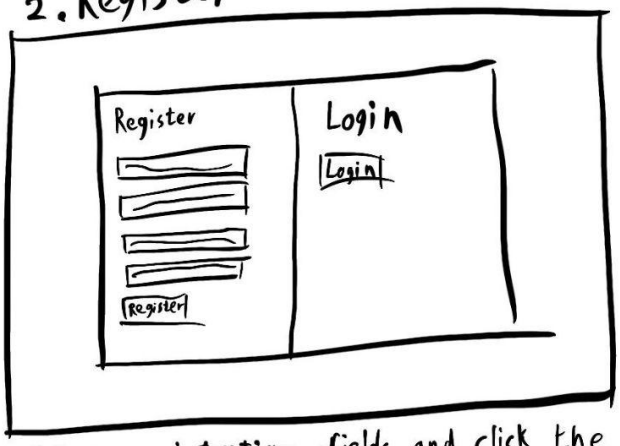
USE CASE 5:

1. Click on "Sign up"



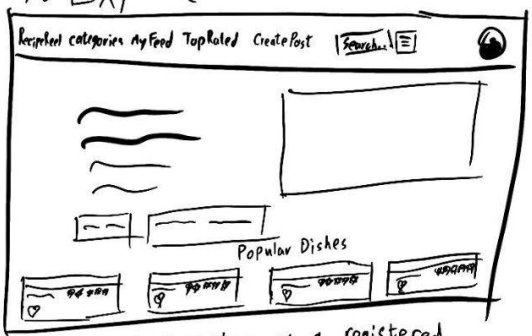
Click on the user icon, then click "Sign up" to access the registration page

2. Register



Fill in registration fields and click the "Register" to finalize registering for a RecipeReel account

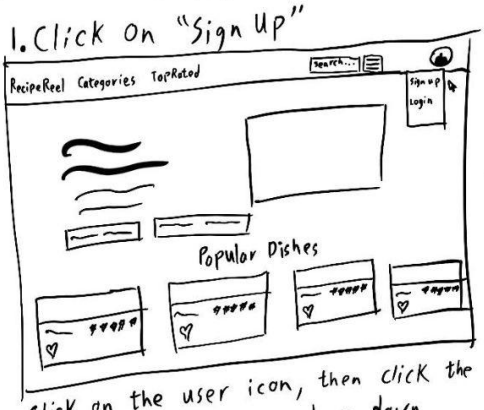
3. Explore



User can now explore as a registered user, and search for recipes with registered user functionality

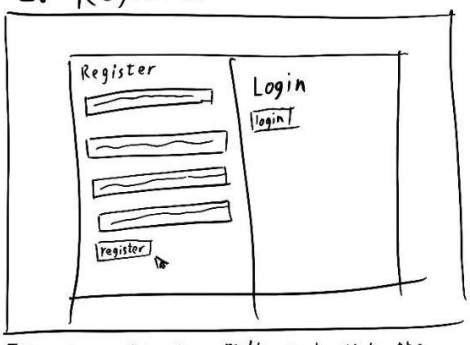
USE CASE 6:

1. Click on "Sign Up"



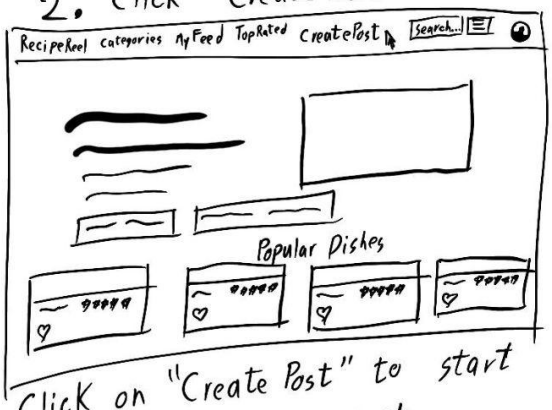
Click on the user icon, then click the "Sign Up" button on the drop down menu to access the registration page

2. Register



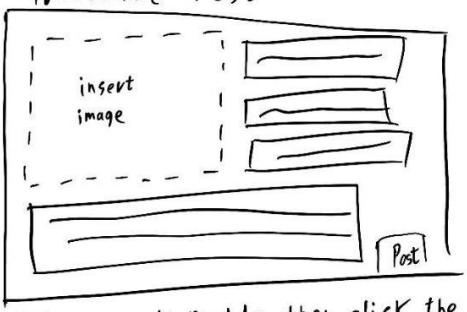
Fill in the registration fields and click the "Register" to finalize registering for a RecipeReel account

2. Click "Create Post"



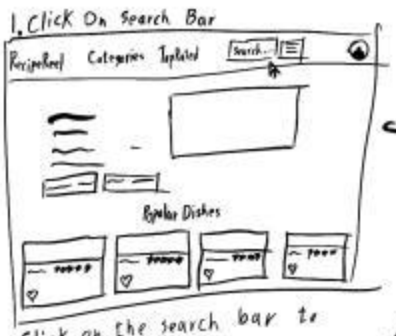
Click on "Create Post" to start creating a new recipe post

4. Create Post

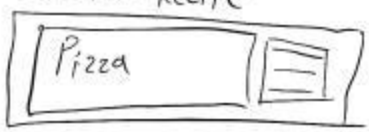


Fill in post fields, then click the post button in the bottom right corner to post it for other users to view

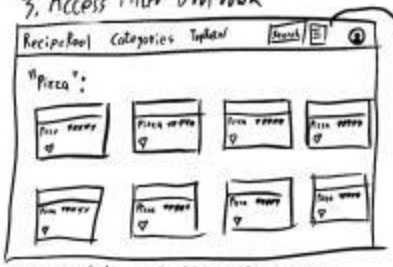
USE CASE 7:



Click on the search bar to start inserting



Enter desired recipe into the search bar



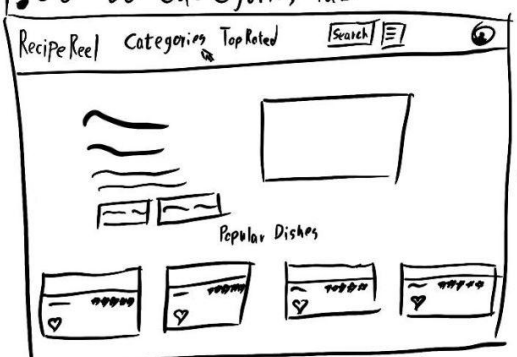
Click Filter button to access the filter dropdown

Select how to sort the search by either shortest or longest cook times. Then select a recipe

Select recipe that fits in user's schedule

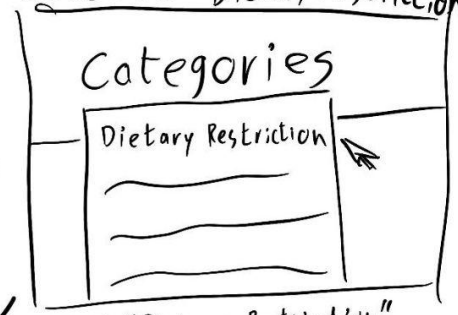
USE CASE 8:

1. Go to Categories Tab



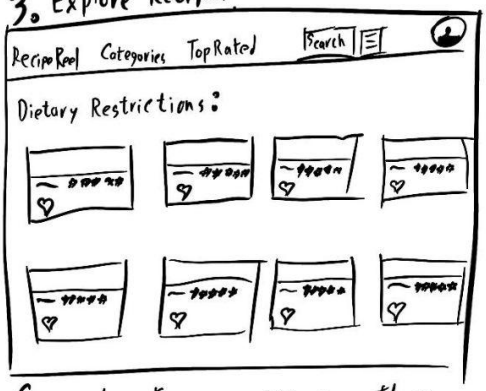
Click on the Categories tab to access the dropdown

2. Select "Dietary Restriction"



Click "Dietary Restriction" to view recipes that fit dietary restrictions

3. Explore Recipes



Search for recipes that fit the user's dietary restrictions

4. High level database architecture and organization

4.1 Database Requirements

- Registered users shall be able to follow zero or many users.
- Registered users shall be able to have zero or many followers.
- A registered user is zero to many with recipes.
- A recipe is one or many with ingredients.
- A recipe is zero or many with comments.
- A recipe is zero or many with ratings.
- A recipe is one or many with instructions.
- A recipe is zero or many with categories.
- A recipe is many to many with categories.

4.2 Database Entities

- General User Entity: None
- **Registered User Entity**: Weak Entity
    - Attributes:
        - id: K int
        - username : varchar 255
        - email : varchar 255
        - password: text (hashed)
        - profile_picture: text (URL)

- **Following Entity**: Weak Entity: depends on registered user
    - Attributes:
        - user_id: K int
        - following_user: K int

- **Follower Entity**: Weak Entity: depends on registered user
    - Attributes:
        - user_id: FK, references Registered User id int
        - follower_user: FK, references Registered User id int

- **Recipe Entity**: Weak Entity: depends on a registered user
    - Attributes:
        - id: PK int
        - user_id: FK, references Registered User id int

        - title: varchar 255
        - description: text
        - created_at: timestamp
        - cooking_time: int
        - difficulty: int
        - recipe_image: text (URL)

- **Ingredient Entity**: Weak Entity: depends on recipe
    - Attributes:
        - id: PK
        - recipe_id: FK, references Recipe id int
        - amount: varchar 255
        - ingredient: varchar 255

- **Comment Entity**: Weak entity: depends on registered user and recipe
    - Attributes:
        - id: PK int
        - user_id: FK, reference Registered User id int
        - recipe_id: FK, references Recipe id int
        - comment: text

- **Comment Likes Entity:** Weak entity: depends on registered user and comment
    - Attributes:
        - user_id: K int
        - comment_id: K int
        - like: boolean

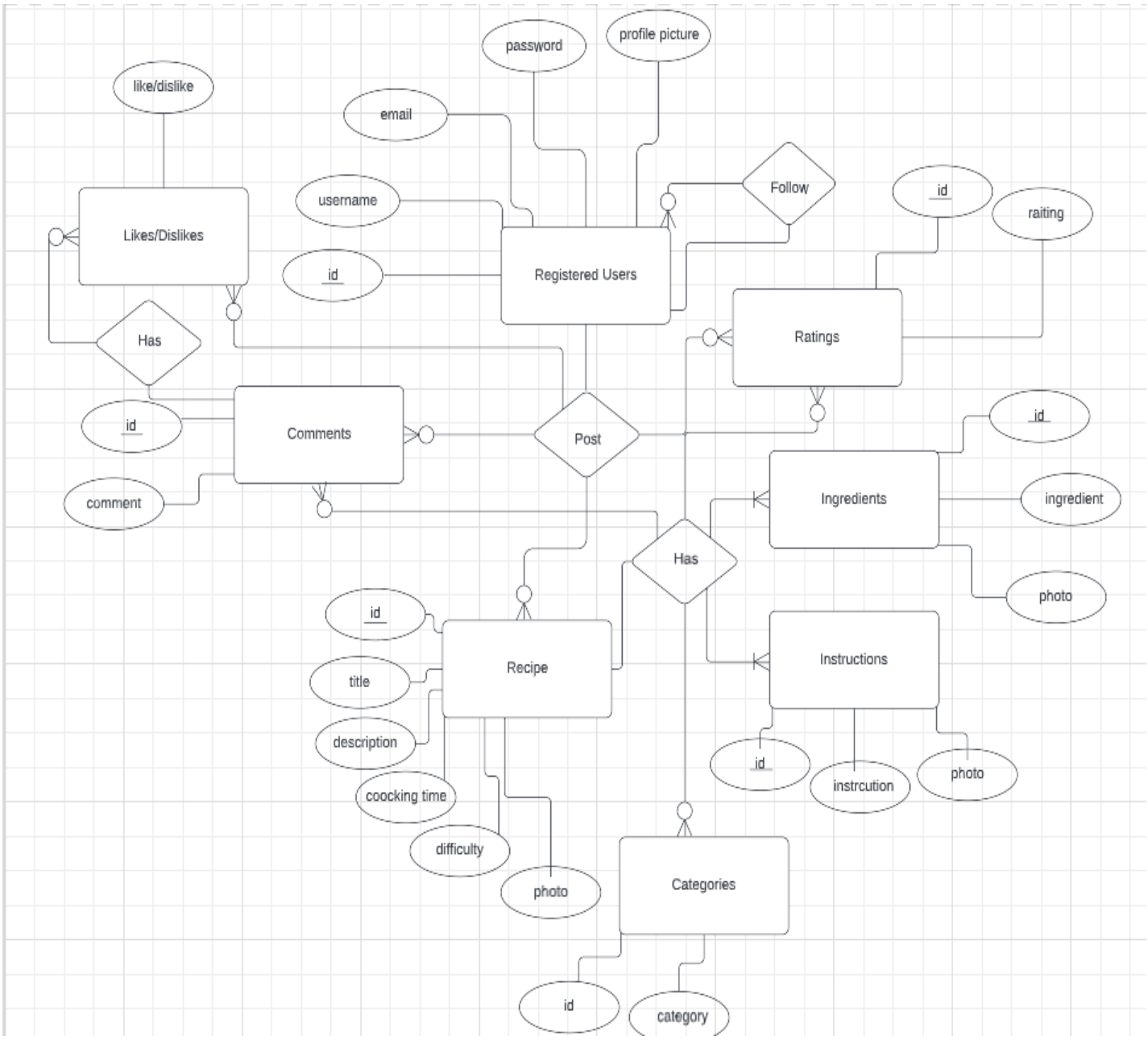
- **Ratings Entity**: Weak Entity: depends on registered user and recipe
    - Attributes:
        - user_id: FK, references Registered User id int
        - recipe_id: FK, references Recipe id int
        - rating: int

- **Instruction Entity**: Weak Entity: depends on recipe
    - Attributes:
        - id: PK int
        - recipe_id: FK, references Recipe id int
        - order: int
        - instruction: text
        - Photo_url: text (URL)

- **Category Entity**: Strong Entity
    - Attributes:
        - Id: PK int
        - Category: varchar 255

- **Category to Recipe**: (join table)
    - Attributes:
        - recipe_id: FK, references Recipe id int
        - Category_id: FK, references Category id int
        - Weak Entity: depends on recipe and category

4.3 ERD



(Diagram 4.3.1, ERD)

4.4 DBMS

We will use PostgresQL DBMS for our project because a few members have experience with this technology.

4.5 Media storage

RecipeReel is a web app where users share images of food only, so we shall keep the images in file systems rather than in DB BLOBs. Storing images in file systems means that the images are physically saved in a folder on the server's hard drive. The web app can then simply reference the file path to display the image on the webpage. While storing images in DB BLOBs (Binary Large Objects), the image is saved directly in the database as a binary object, and this can cause latency that may be slower than direct file access when retrieving and displaying images. Additionally, the database size can grow significantly as more images are added, which can cause issues with backups and maintenance. Therefore, keeping images in file systems will allow for easier management of the images, as they can be easily organized and accessed outside of the database. To achieve this we will use amazon.s3 server to store the media. WE upload to this server and store a URL in our DB to access the media.

4.6 Search/filter architecture and implementation

The search/filter architecture and implementation for the RecipeReel web application shall be designed to provide a fast and efficient search experience for users, enabling them to find the recipes they are looking for quickly and easily. The search functionality can be implemented using a combination of keyword search and filters, and the database terms to be searched would be stored safely and securely in the database and would be used for properly optimizing the search results.

RecipeReel shall use a ranking algorithm for searching and filtering. Since our website is going to gather information about post engagements on the account they follow, this algorithm would help to analyze and look for keywords and other relevant information that later can be used for recommendation. This algorithm also counts the engagement of each post, like the number of reactions, comments, and shares it has gotten. The ranking algorithm investigates the user's search query, looking for keywords, phrases, and other important information. then it matches the user's search query with important contacts in the platform. It shall have access to search our database for posts, users, cuisine, ingredients, and more, if necessary.

5. High level APIs and Main Algorithms

- Add New Registered User to Database
    - In order to add a new Registered User to the database, the required fields on the registration page must be filled out properly. The new registered user API validates the user data, such as username, email address, and password, and checks for any duplicate entries in the database. If the user data is valid the API generates a unique user ID and hashes the user's password, then stores this information in the database. The API sends a response to the client application indicating success or failure of the request. If there are any errors, such as validation errors the register user API sends an appropriate error message to the client application.

- Login User
    - In order for a registered user to login the required fields on the login page must be filled out properly. The login API verifies the email and password entered by the user with the data that is stored in the database. If the credential matches, the user is authenticated and the API generates a JWT token that allows the user to stay logged in for a set period of time. The API sends a response to the client application indicating success or failure of the request. If successful, the user is directed to the homepage.

- Rating Posts
    - When a registered user rates a post, the API will record their UserID, PostID, and the rating value they assigned to the post. The registered user will make a rating by clicking on the 1-5 stars. Once the rating is given, the rating post API will update the number of stars for the post by using the average rating and then update the new total number of ratings. Then the API sends a response to the client application indicating success or failure of the request.

- Modifying Rating of a Post
    - Before a user can modify their rating of a post, they must be logged in to their account. When the post rating is modified, the API takes their UserID, the PostID of that post, and the newly assigned rating, and proceeds to check if the user has previously rated the post. If the API finds a matching UserID, it will proceed to replace the previous rating value with the new rating value, and then update the posts average rating. If the user is removing their rating, the same process will occur, however this time the API will remove the UserID and rating value from

the posts, and update both the average rating value, as well as the number of ratings.

- Following Registered Users
    - Before a user can follow another registered user, they must be logged in to their account. When a registered user clicks the “Follow” button on another user's profile page, the system will create a new record in the “Followers” database. This record will include the User ID of the person pressed the “Follow” button and the User ID of the person they want to follow. After the API creates a new record, the person being followed has their total number of followers updated, and the person who followed will have their total number of followers updated. The user that follows will start to see the following user’s content in their feed.

- Unfollowing Registered User
    - To unfollow a registered user, the user must be logged in to their account and currently following the person they want to unfollow. The user will go to the profile page of the user they want to unfollow and click on the "Unfollow" button. TheAPI will search the "Followers" database to identify the relevant record containing the UserID of the person initiating the unfollow and the User ID of the person being unfollowed. Then the system will remove it from the "Followers" database. After the system finishes the unfollow request, the person that was being followed has their total number of followers updated, and the person who unfollowed will have their total number of followers updated. The user that followed will stop seeing the following user’s content on their feed.

- Liking Post
    - In order to like a post, the user must be logged into their account. The user will like a post by hitting the heart icon, which will turn red once clicked. Once clicked, the API will create a new record in the "Likes" database, including the User ID of the person liking the post and the Post ID of the liked post. After the API finishes the like request, the API will update the post's total number of likes.

- Unliking Posts
    - Before a user can unlike a post, a user must be logged in to their account and have previously liked the specific post. The user will go to the post they want to unlike and click on the red heart icon which will make it gray again. The system will search the "Likes" database to identify the relevant record containing the UserID of the person disliking the post and the PostID of the post being disliked. Then the API will remove it from the "Likes" database. After the API finishes the unlike

request, the post's total number of likes will be updated.(And the APi will send a response to client indicating whether the request was successful or not)

- Creating Posts
    - The user must provide input for all required fields on the post creation page. The create post API then receives the user's post data, such as UserID, title, description, and other requirements. The system ensures that all inputs are complete and valid. If the post data is acceptable, the API generates a unique PostID and associates it with the author's UserID. The API stores the post information, including the PostID, title, content, and metadata, in the database. Subsequently, the API sends a response to the client application indicating the success or failure of the post creation request. Upon a successful request, the user is redirected to the newly created post page.

- Deleting Posts
    - To delete a post, the user must be logged in as the author of the post. The user will initiate deleting the post by hitting the delete button on their post. The API will then compare the UserID with the PostID, if they are connected in the system, the API will remove the post. The API sends a response to the client application indicating success or failure of the request. If successful the user will see their comment deleted off the screen.

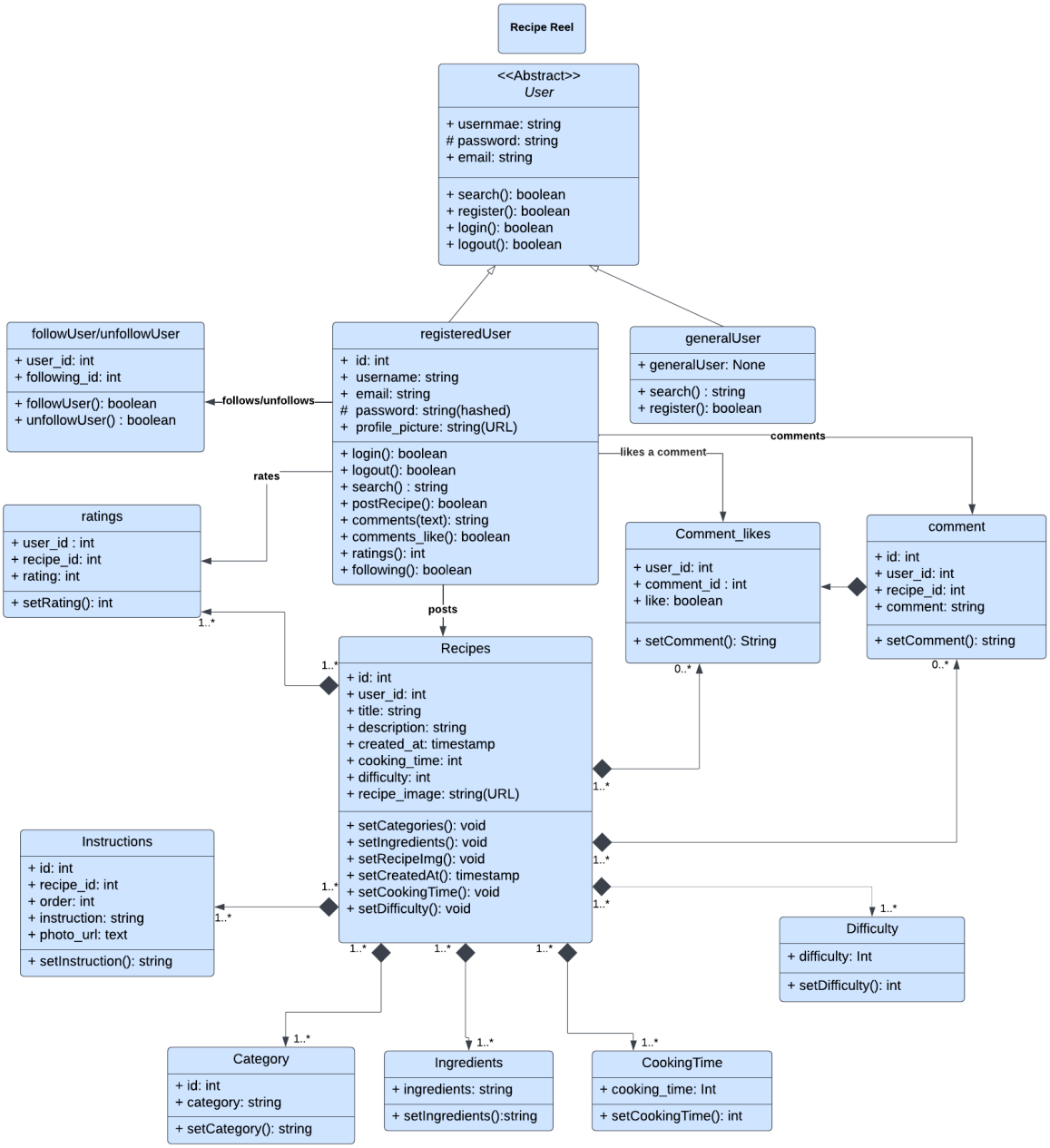
- Searching for Recipes
    - To search for recipes, the user enters a query into the search bar, which may include recipe names, ingredients or other queries. The search recipes API receives the imputed query and uses the backend Postgres query to search the database for recipes that contain the user’s query. The query uses multiple joins and ILIKE conditions to match the user’s query with the correct recipes. Once the API finds the matching recipes the API sends a response to the client which includes the search results and associated data wanted. Finally, the data is displayed on the client for the user to view.

- Commenting on Posts
    - To comment on a post, the registered user must be logged in and click on the post they want to comment on. The user must fill out the comment box and then click the submit button. The comment API will take in the user’s comment and other relevant data such as UserID and PostID to create a CommentID. The API then stores all of the data in the database. The API then sends a response to the client application indicating success or failure of the request. If successful the user will see their comment load on the screen.

- Deleting Comments
    - To delete a comment, the user must be logged in as the author of the post. The user will initiate deleting the comment by hitting the delete button on their comment. The system will then compare the UserID with the CommentID, if they are connected in the system, the API will remove the comment. The API sends a response to the client application indicating success or failure of the request. If successful the user will see their comment deleted off the screen.

6. High Level UML Diagrams

6.1 UML



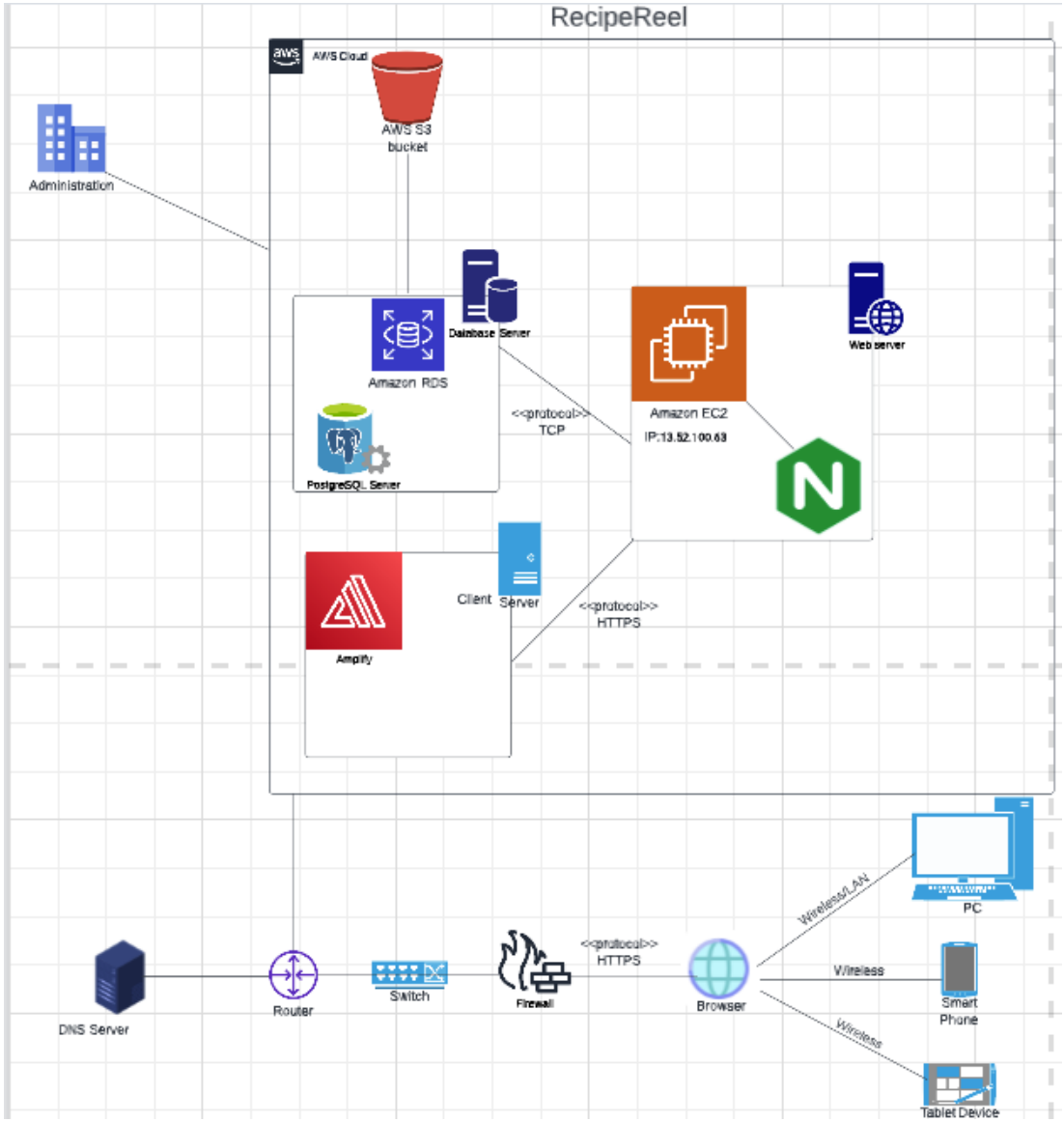
10 Main entities in RecipeReel Application UML diagram

1. Register user
2. Recipe: contains ingredients, cuisine, cooking time and difficulty level, registered user can post a recipe, but not vice versa

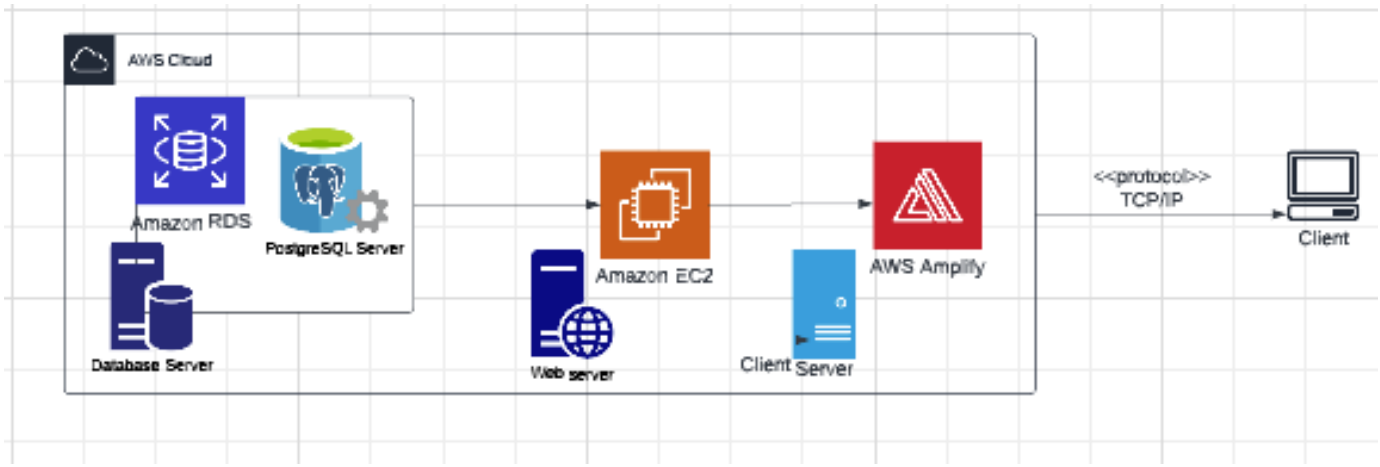
3. Category: depends on a recipe, cannot exist without a recipe
4. Ingredients: depends on a recipe, cannot exist without a recipe
5. Cooking time: depends on a recipe, cannot exist without a recipe
6. Difficulty: depends on a recipe, cannot exist without a recipe
7. Comment: depends on a recipe, cannot exist without a recipe and register user
8. Comment_like: depends on a comment, cannot exist without a comment
9. followUser/unfollowUser: depends on a registered user
10. Ratings : depends on a recipe, cannot exist without a recipe and it has a unidirectional association relationship with a registered user. A registered user can rate, but not vice versa.

7. High Level Application Network and Deployment Diagrams

7.1 Application Network



7.2 Deployment Diagram


AWS Cloud
Amazon RDS
PostgreSQL Server
Database Server
Amazon EC2
Web server
Client Server
AWS Amplify
<<protocol>>
TCP/IP
Client

8. Identify Actual Key Risks of The Project

8.1 Skills risks:

Our team has encountered some problems with skills. Regarding high level database design docs, some people on the team did not understand how relational databases work. This led to some entities that made no sense, some throws had FK and Pk that did not connect them to other tables. Some rows are nonsensical, some rows have been deleted, re added, then deleted again because of a lack of understanding of RDS principles.

Moving forward, skills in the group are still a risk. If the high level architecture cannot be agreed upon in a cohesive way because of a lack of understanding, we are scared that the group will lack the skills to actually create a usable RDS architecture.

To remedy the skills risk, we have been trying to get to the same understanding in our weekly meetings.

8.2 schedule risks:

We encountered some schedule risks for properly managing the time and resources needed to complete tasks on time. To solve this, we are using Discord, Trello and weekly meetings to carefully plan and estimate the time and resources needed for each task, and regularly monitor our progress to ensure we are on track.

Everyone has their own life and own schedule, this is unavoidable. Our scheduled meeting has not been an issue, everyone has been able to make it or they have been able to let people know about their issues if they cannot make it. Over spring break, schedule problems started to arise. A few members were able to work on M2 over the break while a few team members were totally unreachable. This caused the team to fall very behind on M2.

We need to fix this, if team members have schedule issues, they need to be vocal about it. We cannot have schedule issues and communication issues at the same time. There should be no communication issues. We have Discord and e-mail.

8.3 technical risks

We encountered technical risks and faced some technical challenges that we hadn’t an insight before, such as integrating certain technologies or addressing compatibility issues. To solve this, we are doing our due diligence to conduct thorough research and testing to identify potential technical challenges, address them and work on them proactively. We also collaborate with other team members who have experience with similar tasks.

Some technical risks we expect to encounter are problems with AWS since all of us are using AWS products for the first time. We are using AWS RDS, EC2, S3, and Amplify to host our application and media. Since we are new to these technologies, there will be problems that come up that we will have to research and learn how to solve. This has already happened with Amplify; we had to learn how to build and start the client on the Amplify server.

I think the next best way to resolve these problems is to do stuff early and allow time for problems. We know we are new to these technologies, so we need to allow time for mistakes, correct them, and learn how to do it right in the first place.

8.4 teamwork risk

We encountered teamwork risks as a result of communication or collaboration issues among team members. To solve this, we set up a team lead in order to arrange clear communication channels and guidelines for collaboration and regularly check in with team members to ensure everyone is on the same page. We can also use several collaboration tools and techniques, such as Trello, Discord, Zoom,and Google Docs, to support effective teamwork.

We have encountered major teamwork risks. There has been a huge lack of communication from a few team members. This lack of communication and general procrastination from some team members has led to problems in completing quality milestones on time. For example, a lack of communication leads to the completion of a section, but this section does not make sense with what was agreed upon before. Since it was turned in at the last minute, there is no time for the review team to review, give feedback, and for this feedback to be implemented. Lack of communication and the drive to complete things in a timely manner have led to a messy project and workflow. Several team members have taken initiative and taken on sections that have been assigned to other members but have not been completed.

This needs to be resolved immediately, and the team needs to communicate. Discord messages cannot be ignored. Sections need to be completed within days, not a week or so from assignment, so that reviewers can review and give feedback, and this feedback can be implemented. Questions need to be asked; if things are unclear, ask. This is so that we are not doing work that needs to be corrected later; this is a wasted effort.

8.5 legal/content risks:

One potential legal or content risk for our project could be a copyright violation if we use images or other content without proper licensing or permission. To address this risk, we plan to ensure that all images and other content used in the project are either owned by us or obtained from public domains. We shall keep track of all sources of

content and ensure that proper acknowledgement is given. When necessary, we shall seek legal advice to ensure compliance and licensing with copyright laws.

A big risk we have on this website is publishing copyrighted works on the website. Recipes that are published in paid publications such as books, newspapers, and magazines are likely to be copyrighted

9. Project Management

We use Trello to keep track of our tasks. We have created different lists for each stage of the task on Trello, such as the "TO DO" list, the "Doing" list, and the "Done" list. This system ensures that each team member is aware of what tasks are pending, in progress, and completed.

Our team has a weekly meeting on Monday from 5 to 6 pm and an optional meeting on Wednesday from 3:30 to 4 pm. During the Monday meeting, we discuss our project's milestone sections, and our team lead assigns tasks to each member based on their expertise. The team lead writes these tasks on the "TO DO" list in Trello. Each member is responsible for writing their task on the "Doing" list with their corresponding name. When the task is completed, they move it to the "Done" list. This allows team members to track their progress and helps us complete tasks on time.

To communicate with each other, we use Discord. We have set up various channels to communicate about different topics, such as a general channel for discussions, an off-topic channel for non-work-related conversations, and channels for frontend, backend, and database discussions. We also have an "Announcements" channel where the team lead can send updates about the project's progress or any important notifications.

The follow-up method When the team leader assigns a task with a one-week deadline, the team leader follows up with each individual team member two days after assigning the task and asks about the progress and challenges that the team member is facing. The team lead offers help and encourages the team member to discuss with other members in the Discord group.

10. Detailed List of Contributions

Team member	Contributions/Activities	Score
Team lead Yueling Liu	● Refined 7 data definition in Section 1 ● Refined 7 must have functional requirements for general users in Section 2 ● Refined 5 must have functional requirements for admin ● Worked on Section 6 UML diagram ● Implemented login and register page for front-end ● Reviewed documentation	
Backend lead Duncan Herington	● Refined 6 data definitions in Section 1 ● Refined 4 desired functional requirements for general users ● Refined 9 desired functional requirements for registered users ● Worked on High level APIs and main Algorithms in Section 5 ● Designed front-end styling ● Code front end ● Implemented front-end and back-end	10/10
Backend lead Marcel Azouri	● Proofreaded all functional requirements for Priority 1, Priority 2 and Priority 3, including general users, registered users, and admins in Section 2 ● Aided team members in fixing their work by providing feedback to each team member. ● Worked on UI Mockups and Storyboard for front-end in Section 1 ● Worked on High level APIs and main Algorithms in Section 5	10/10

Frontend lead Priya Pradeep	• Refined 6 entities in Section 1 • Refined 2 desired functional requirement for register user in Section 2 • Refined 1 must have functional requirement for admin in section 2 • Refined 2 desired functional requirements for admins in Section 2 • Reviewed documentation M2	8/10
Github lead Nathan Le Howland	• Refine 4 data definitions for general user and its sub data items in Section 1 • Refined 8 must have functional requirements for registered users in Section 2 • Worked on Application Network and Deployment Diagrams in Section 7 • Assisted creating database tables • Assisted helping database ERD	10/10
Database lead Samuel	• Refined 7 data definitions in Section 1 • Refined 1 opportunistic functional requirements for general users in Section 2 • Refined 3 opportunistic functional requirements for admins in Section 2 • Worked on Database architecture and organization in Section 4 • Worked on section 8 Identify Actual Key Risks	8/10
Document lead Yasson	• Refined 5 data definitions in Section 1 • Refined 3 opportunistic functional requirements registered users in Section 2 • Worked on Database architecture and organization in Section 4	8/10

SW Engineering CSC 648-05 Spring 2023

RecipeReel

T03 Milestone 3

Team lead: Yueling Liu yliu50@sfsu.edu

Backend lead: Duncan Herington, Marcel Azouri

GitHub lead: Nathan Le Howland

History table

M3V2	May 18, 2023
M3V1	April 27, 2023
M2V2	April 27, 2023
M2V1	April 3, 2023
M1V2	April 3, 2023
M1V1	March 2, 2023

Table of Contents

1. Data Definitions

1. General user: This entity is a user that accesses the site with no registered account. A general user will have access to a limited selection of features of the site. A general user is allowed to browse and search recipes but cannot interact with posts such as leaving a comment, saving a post, or giving a like to a post.

2. Register an account: It is an entity that allows a general user to register/sign up for an account so they can access more functions of the site. The purpose of this entity is to allow a general user to create an account and start using the social features of the site.

3. Search**:** It is an entity that allows users to search for recipes by specific keywords. These keywords such as ingredients. General users have access to search because we want them to be able to look up recipes that they would like to cook.

4. Registered user**:** It is an entity that created an account and can sign in to the website from different devices. This user will have a unique email and a password.

    4.1 Username

    It is an entity that is typically a string of characters that was chosen by the user to identify themselves. The length of the username shall be limited to a minimum of 6 characters and a maximum of 20 characters. The username is required when logging in on the website. It's paired with a password.

    4.2 Password

    It is an entity that refers to a combination of characters, strings, numbers, or letters that was chosen by users to secure their account and prevent unauthorized access. The password shall be a minimum of 6 characters and a maximum of 20 characters with a special character. This entity is required when logging into the website. When both the username and password are correctly entered, the user will be granted access to their account.

    4.3 Following

    It is an entity that refers to a registered user who follows another user's content on the website. When a user follows another user, they can view the updates about the new contents that the following user has posted. The purpose of this entity is to build connections between users, help users find similar interests in recipes, and discover new recipes and content with other users.

4.4 Unfollowing

It is an entity that allows a registered user to unfollow certain users. The purpose of this entity is to allow users to see only recipes from specific users that they are interested in following.

4.5 Posts

It is an entity that allows a registered user to post personal recipes, allowing them to help spread new varieties and food ideas. The post contains 1-6 images per post, with a maximum of 2000 characters.

4.6 Delete posts

It is an entity that allows a registered user to delete their own recipes. When a user deletes a recipe, the content will be removed from the website, and all the comments from other users are no longer visible on the website. The purpose of this entity is to give users the ability to manage their accounts. It also can be helpful if a user accidentally posts something; they can quickly delete it by being seen by other users.

4.7 Like

It is an entity that allows users to express their positive reaction or appreciation for a particular recipe post. The purpose of a “like” entity is to provide a simple and quick way for users to show their support or approval without having to write a comment or engage in a more extended conversation. This also allows users to see how popular or well-received a particular recipe is.

4.8 Dislike

It is an entity that allows users to express negative sentiment or dissatisfaction. The purpose of a “dislike” entity is to provide users with a way to give feedback on a recipe that they don’t enjoy or find useful without having to write a comment or engage in an extended conversation.

4.9 Comment

It is an entity that allows users to engage in a conversation or provide feedback on a particular recipe post. Users can write a text-based response, which they can then share publicly for other users to see and respond to. This can be used to communicate thoughts, views, or ideas in a comment entity in a more extensive and detailed manner, as well as to interact with other users.

4.10 Rate

It is an entity that allows users to provide a way to express their opinions and experiences with a particular recipe and help other users make informed decisions. The

rating would help provide feedback to the particular user who posted the recipe and help other users decide whether or not to try it out.

5. Top Rated: Top rated is an entity that will provide users with ratings based on several factors such as the quality and uniqueness of the recipes, the level of detail in the instructions, the variety of recipes offered, and difficulty of cooking.

6. Latest Post**:** Latest post entity for a recipe website would be the most recently published recipe on the website. This would be a new recipe that has just been added to the website's collection. The latest post entity is important as it keeps the website fresh and up-to-date, and provides users with new content to explore and engage with.

7. Recipe**:** Recipe is an entity that will provide users information about a specific recipe. It includes other entities such as, recipe title, which is the name of the dish; A brief description of this recipe; Ingredient list which informs the user all the ingredients needed to make the dish; Cooking time, which tells the user the amount of time required to cook the dish; Recipe images that user posted, and comments from other users etc.

   7.1 Recipe title

   It is an entity that will provide the user with a name for a particular recipe. It provides the user with information or a description that accurately reflects the content of the recipe.

   7.2 Description

   It is an entity that will provide the user with a short summary of the dish that highlights its main ingredients and background. The description may also provide additional details such as serving size, cooking time, and any special equipment or techniques that are needed.

   7.3 Cooking Time

   Cooking time is an entity that will provide users with an estimation of the total time required to cook a specific dish. The purpose of this entity is to allow users to plan their meal preparation and manage their time. The format of this entity will be displayed as hours and minutes. Cooking time also interacts with other entities, such as preparation time, which tells the user the amount of time required for preparing the dish, and resting time, which tells the user how long to let the food sit before eating.

   7.4 Difficulty

Difficulty is an entity that will provide the user with an idea of how hard the dish will be to make for themselves. By having this feature, it allows the user to find dishes that suit their level of cooking skills.

7.5 Ingredients

It is an entity that users shall search for recipes based on the ingredients they have on hand. The ingredients shall include various items that are used in cooking recipes, such as chicken, fish, pork, beef, vegetables, and others. These ingredients can be represented as an enumeration or a list, and the search shall return recipes that can be made with those ingredients.

7.6 Instruction

Instructions are an entity that will provide the user with step-by-step guidance on how to prepare and cook a dish. Instructions also interact with other entities, such as the ingredient list, which informs the user of all the ingredients needed to make the dish, and the cooking method, which informs the user if they need a grill or stove to cook certain meals.

7.8 Category

It is an entity that shall allow users to easily navigate and filter through the available recipes based on their preferences. It is an entity that serves as a container for organizing and grouping various sub-data items such as cuisine, ingredient, occasion, and dietary restrictions. The category entity will provide an efficient and useful way for users to find the recipes they are looking for.

7.9 RecipeImages

It is an entity that users post on the website; a registered user can post 1-6 images that are related to the recipe. Some good photographs can also make the recipe more appealing and attractive, which can increase engagement and interest among users.

2. Functional Requirements

2.1. Priority 1

2.1.1. General User

2.1.1.1. A general user shall be able to register for an account of RecipeReel using a username, password and email.

2.1.1.2. A general user shall be able to view recipes posted on the website using a feed.

2.1.1.3. A general user shall be able to search or filter recipes using predefined categories/cuisines.

2.1.1.4. A general user shall be able to search or filter recipes using predefined ingredients.

2.1.1.5. A general user shall be able to search using cooking time.

2.1.1.6. A general user shall be able to search using a keyword or term that matches a recipe's ingredients

2.1.1.7. A general user shall be able to view recipes, details, and comments.

2.1.2. Registered User

2.1.2.1. A registered user shall be able to log in with a username and password.

2.1.2.2. A registered user shall be able to save their favorite recipes.

2.1.2.3. A registered user shall be able to leave comments on recipes.

2.1.2.4. A registered user shall be able to like comments on recipes.

2.1.2.5. A registered user shall be able to dislike comments.

2.1.2.6. A registered user shall be able to post recipes to ReciperReel.

2.1.2.7. A registered user shall be able to delete a recipe they posted to RecipeReel.

2.1.2.8. A registered user shall be able to follow other registered users.

2.1.2.9. A registered user shall be able to unfollow other registered users.

2.1.2.10. A registered user shall be able to log out.

2.1.2.11. A registered user shall be able to view a feed of recipes(posts) based on other registered users they follow.

2.1.2.12. A registered user shall be able to rate a recipe using a 1-5 scale.

2.1.3. Admin

2.1.3.1. An admin shall be able to use tools such as PGAdmin and AWS tools to monitor and update all site contents.

2.1.3.2. An admin shall be able to manage website functionalities using source code.

2.1.3.3. An admin shall be able to analyze site performance using AWS amplify console.

2.1.3.4. Admin shall keep user information and data stored safely using a postgres DB.

2.1.3.5. An admin shall be able to manage the amount of storage the website has in order to make sure there's enough to store contents such as images, recipes, descriptions, and more.

2.1.3.6. An admin shall manage the amount of storage the website used using AWS RDS console.

2.2. Priority 2

2.2.1. General Users

2.2.1.1. A general user shall be able to register using a phone number.

2.2.1.2. A general user shall be able to filter by author.

2.2.1.3. A general user shall be able to filter recipes by date added.

2.2.1.4. A general user shall be able to filter by highest review.

2.2.1.5. A general user shall be able to filter based on calorie count.

2.2.1.6. A general user shall be able to filter by difficulty level.

2.2.1.7. A general user shall be able to search for recipes that do not include ingredients.

2.2.2. Registered Users

2.2.2.1. A registered user shall be able to verify with their phone number.

2.2.2.2. A registered user shall be able to enable 2FA using a 2FA app.

2.2.2.3. A registered user shall be able to edit their profiles.

2.2.2.4. A registered user shall be able to download a pdf of a recipe.

2.2.2.5. A registered user shall be able to archive their recipes.

2.2.2.6. A registered user shall be able to edit their security information.

2.2.2.7. A registered user shall be able to save posts.

2.2.2.8. A registered user shall be able to edit their recipes.

2.2.2.9. A registered user shall be able to delete their recipes.

2.2.3. Admin

2.2.3.1. An admin user shall be verified using two-factor authentication.

2.2.3.2. An admin user shall be able to verify using their phone number.

2.3. Priority 3

2.3.1. General Users

2.3.1.1. A user shall have access to a meal planner tool for their meal criteria.

2.3.2. Registered Users

2.3.2.1. Registered users shall be able to privately message each other.

2.3.2.2. Registered users shall be able to opt for notifications.

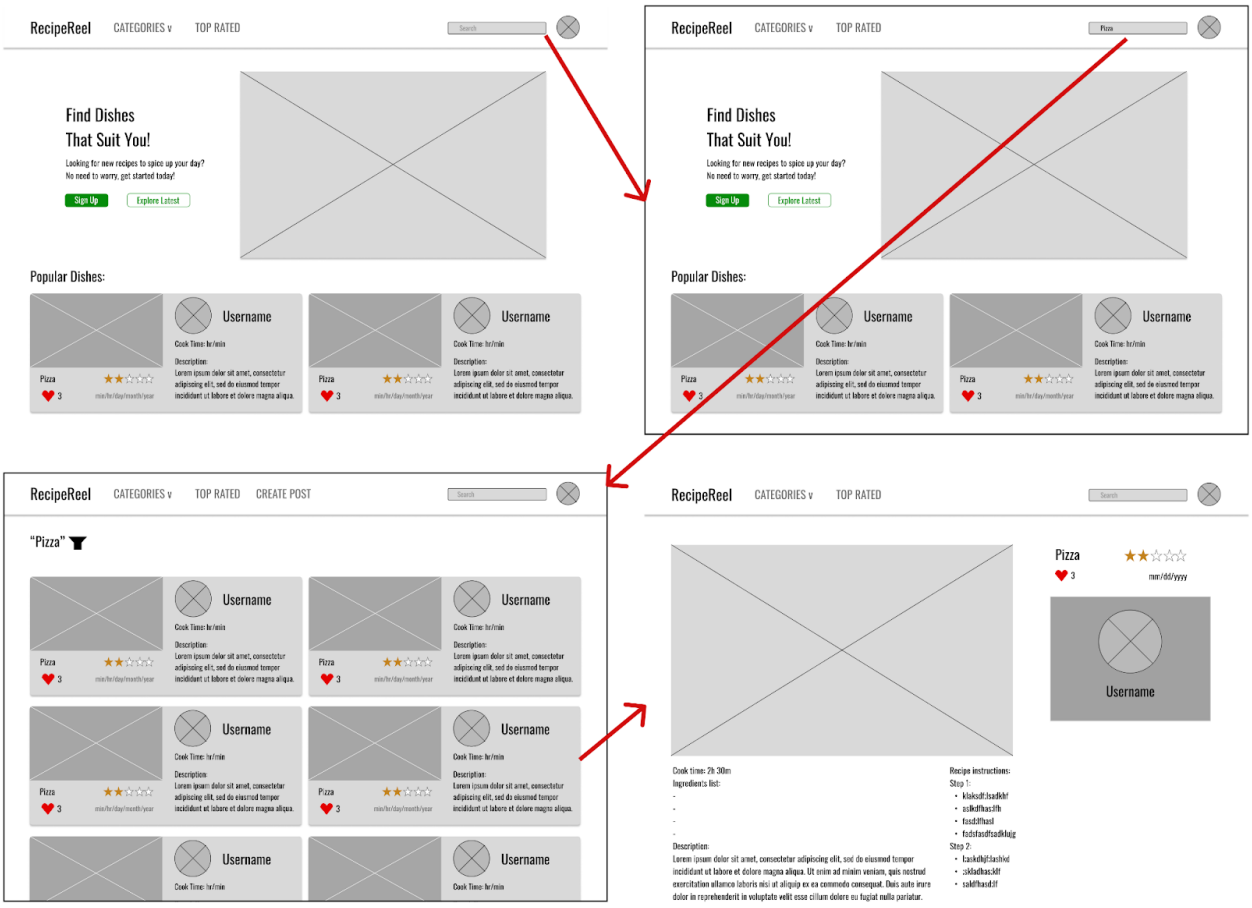
2.3.3. Admin

2.3.3.1. An admin shall have monitoring tools for tracking system metrics.

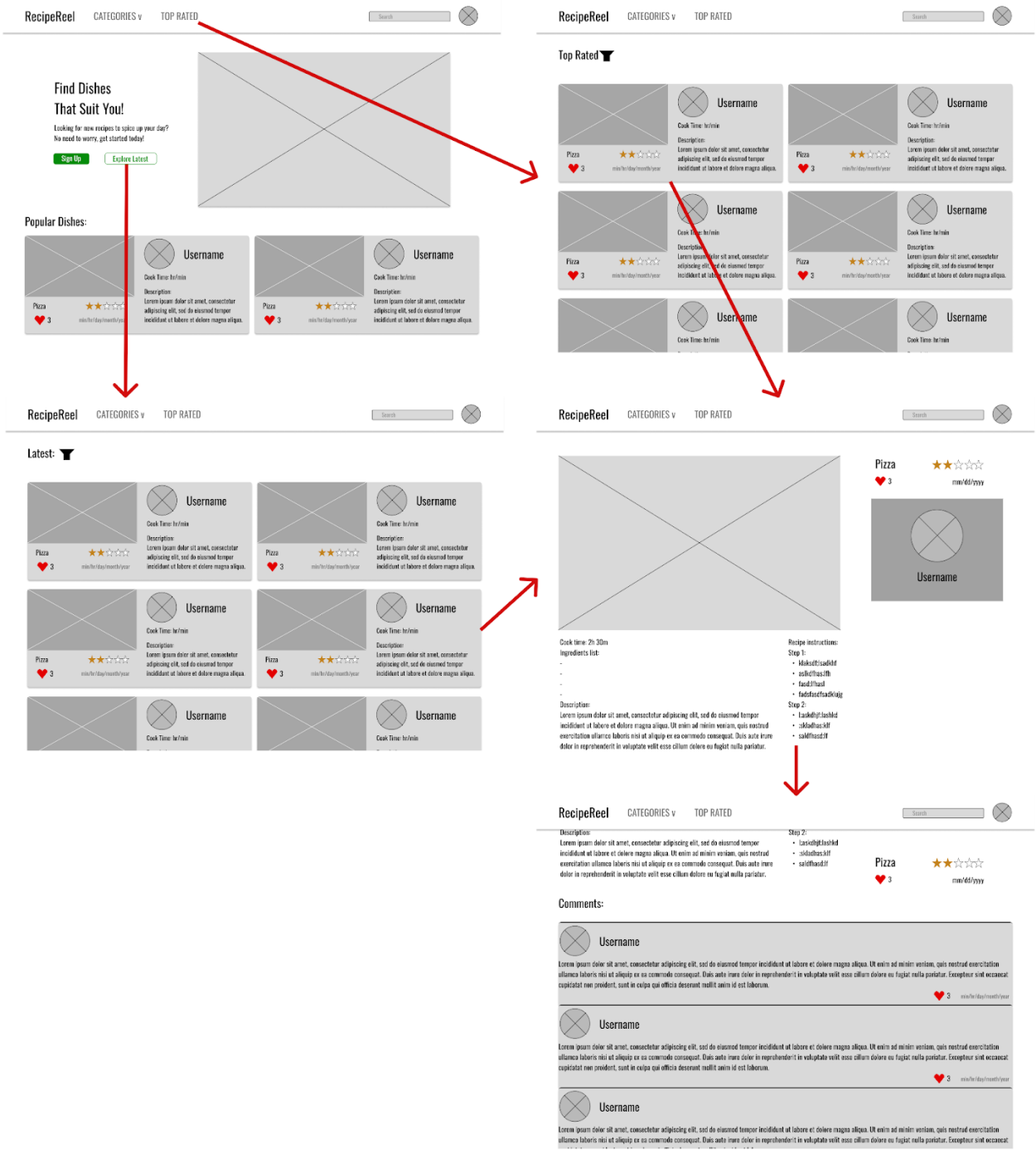
2.3.3.2. An admin shall have a revenue management tool.

3. Wireframes Based on Mockups/Storyboards V2

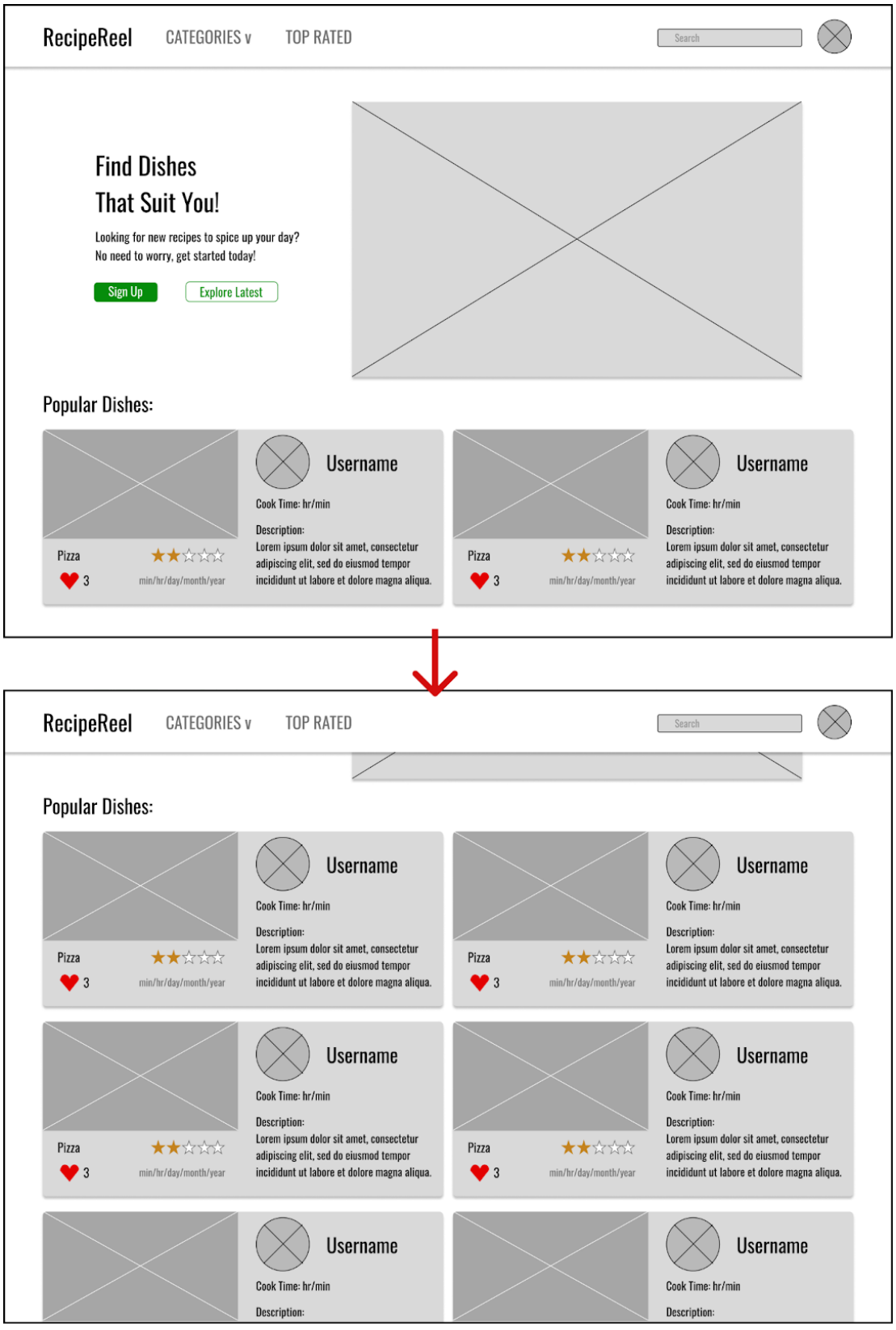
Use Case 1:



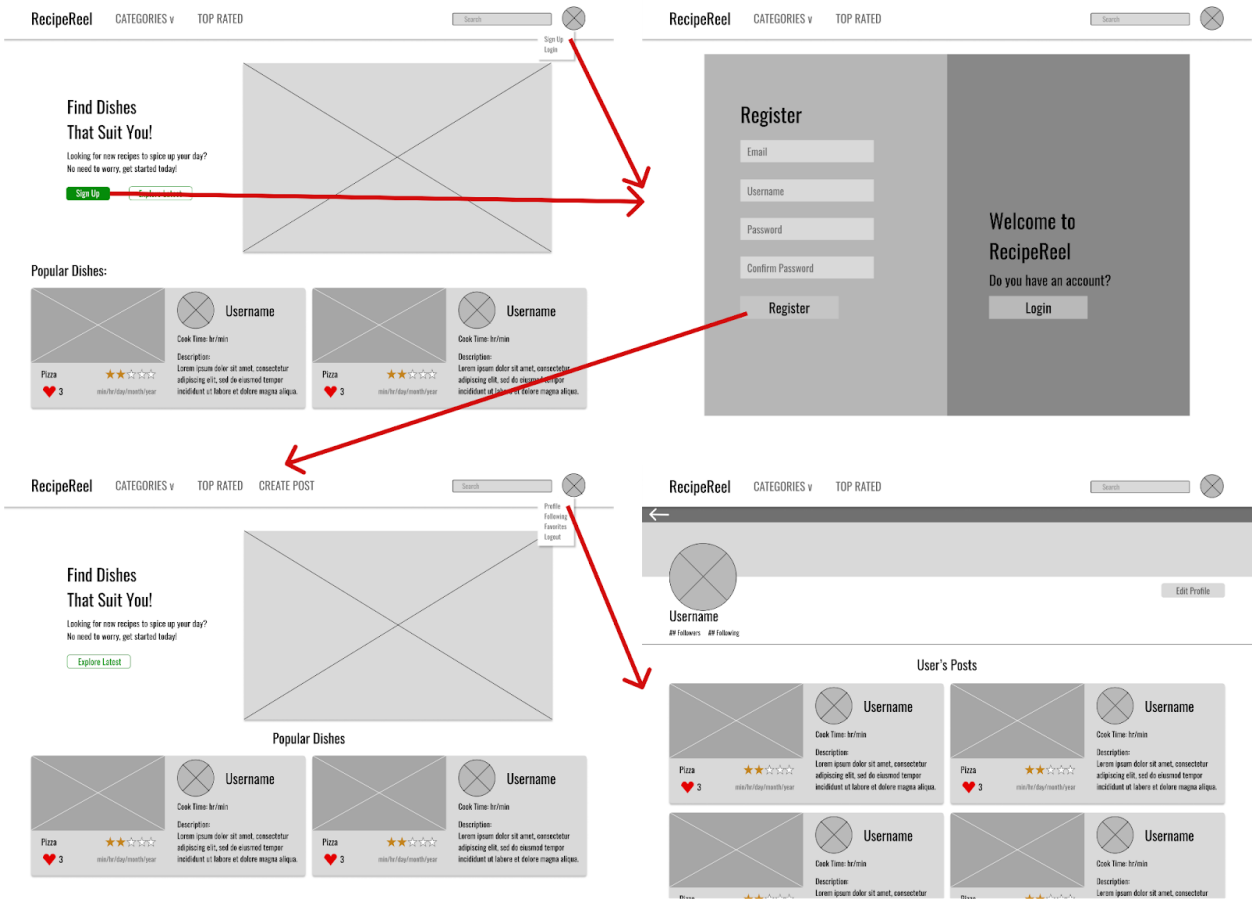
Use Case 2:


RecipeReel
CATEGORIES v
TOP RATED
Find Dishes
That Suit You!
Looking for new recipes to spice up your day?
No need to worry, get started today!
Sign Up
Explore Latest
Popular Dishes:
Username
Top Rated
Latest:
Pizza
mm/dd/yyyy
Cook time: 2h 30m
Ingredients list:
Recipe instructions:
Step 1:
Step 2:
Description:
Comments:

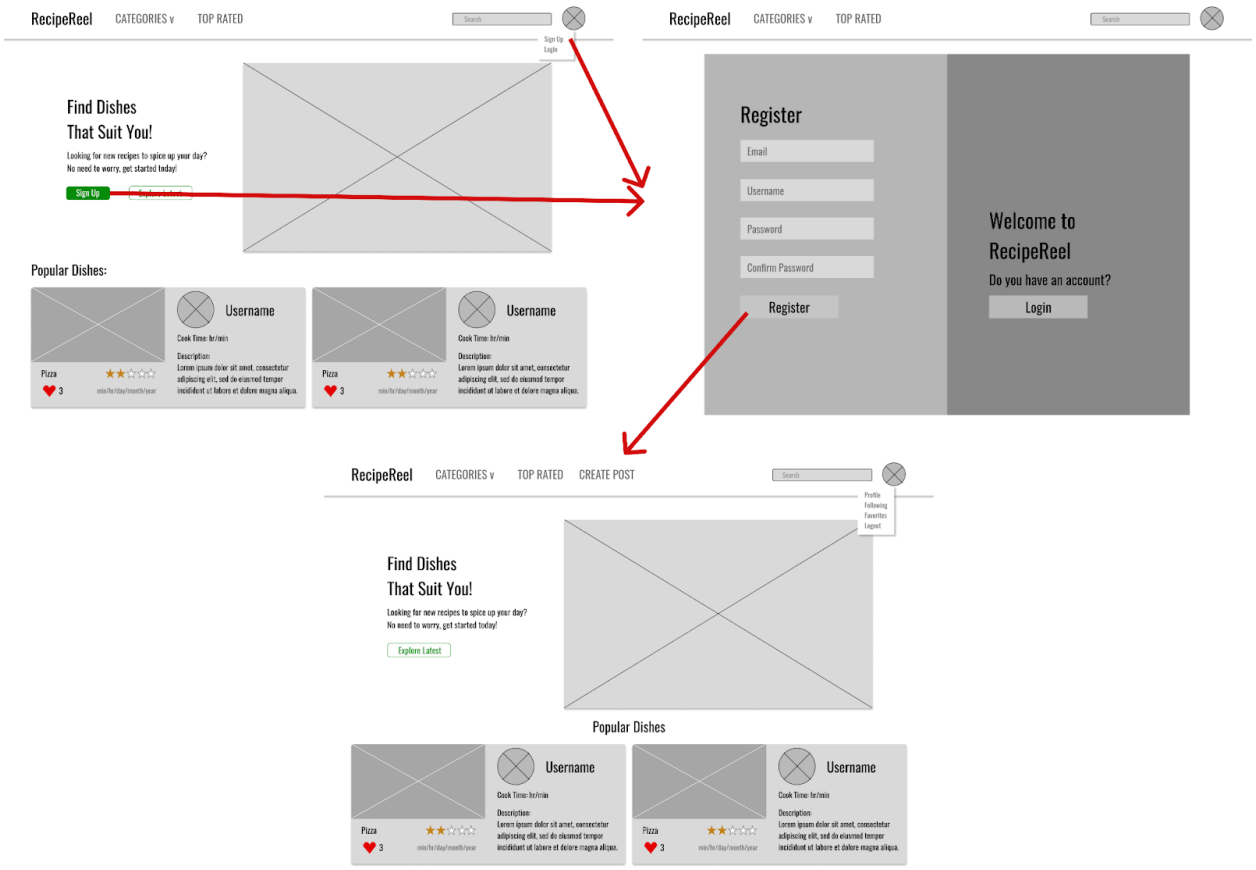
Use Case 3:


RecipeReel
CATEGORIES v
TOP RATED
Search
Find Dishes
That Suit You!
Looking for new recipes to spice up your day?
No need to worry, get started today!
Sign Up
Explore Latest
Popular Dishes:
Username
Cook Time: hr/min
Description:
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Pizza
3
min/hr/day/month/year
RecipeReel
CATEGORIES v
TOP RATED
Search
Popular Dishes:
Username
Cook Time: hr/min
Description:
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Pizza
3
min/hr/day/month/year

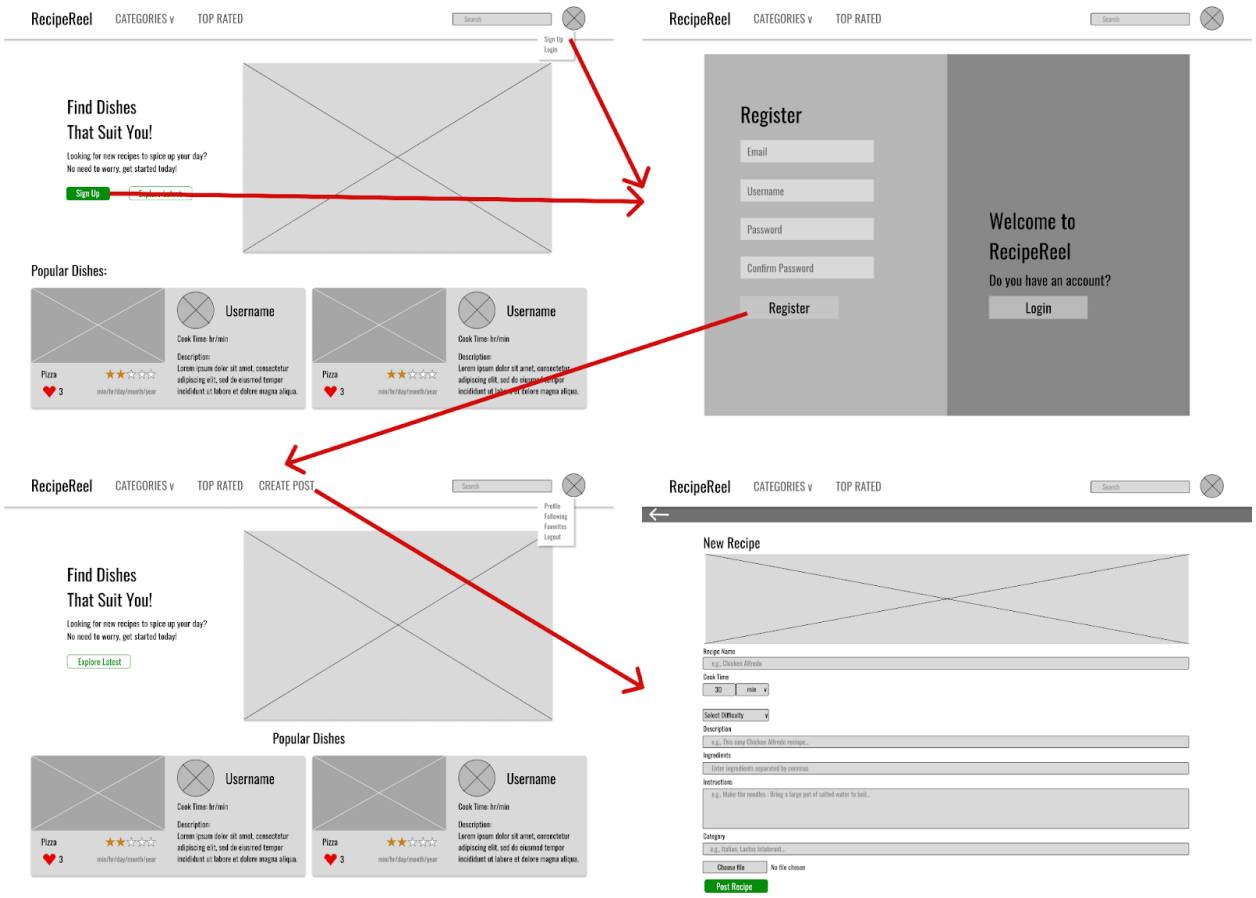
Use Case 4:


RecipeReel
CATEGORIES ∨
TOP RATED
Search
Sign Up
Login
Find Dishes
That Suit You!
Looking for new recipes to spice up your day?
No need to worry, get started today!
Sign Up
Explore Latest
Popular Dishes:
Username
Cook Time: hr/min
Description:
Pizza
3
Register
Email
Username
Password
Confirm Password
Register
Welcome to
RecipeReel
Do you have an account?
Login
CREATE POST
Profile
Following
Favorites
Logout
Popular Dishes
Edit Profile
Followers
Following
User's Posts

Use Case 5:


RecipeReel
CATEGORIES ∨
TOP RATED
Search
Sign Up
Login
Find Dishes
That Suit You!
Looking for new recipes to spice up your day?
No need to worry, get started today!
Sign Up
Explore Latest
Popular Dishes:
Username
Cook Time: hr/min
Description:
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Pizza
3
min/hr/day/month/year
Register
Email
Username
Password
Confirm Password
Register
Welcome to
RecipeReel
Do you have an account?
Login
CREATE POST
Profile
Following
Favorites
Logout
Popular Dishes

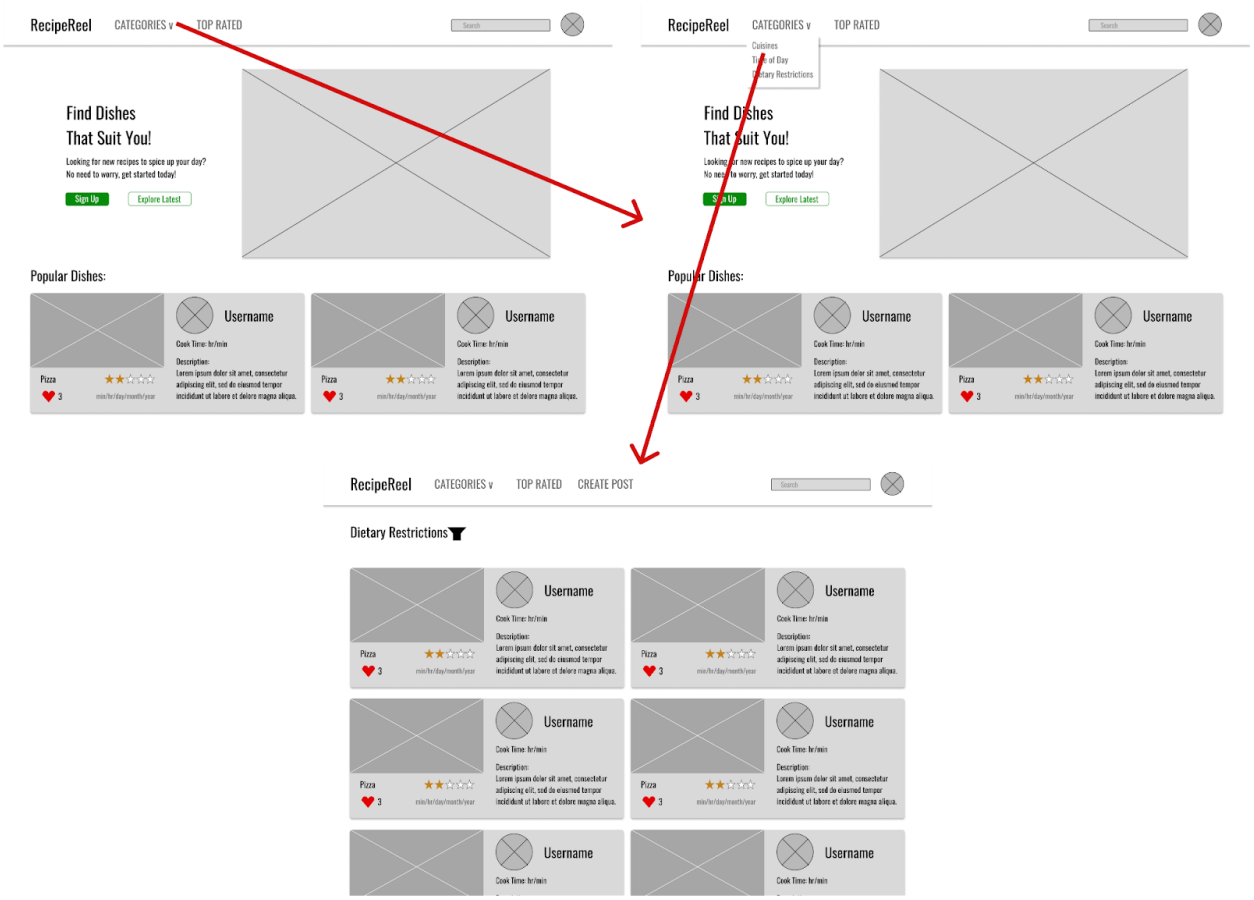
Use Case 6:


RecipeReel
CATEGORIES ∨
TOP RATED
Search
Sign Up
Login
Find Dishes
That Suit You!
Looking for new recipes to spice up your day?
No need to worry, get started today!
Sign Up
Explore Latest
Popular Dishes:
Username
Cook Time: hr/min
Description:
Pizza
3
RecipeReel
CATEGORIES ∨
TOP RATED
Search
Register
Email
Username
Password
Confirm Password
Register
Welcome to
RecipeReel
Do you have an account?
Login
RecipeReel
CATEGORIES ∨
TOP RATED
CREATE POST
Search
Profile
Following
Favorites
Logout
Find Dishes
That Suit You!
Looking for new recipes to spice up your day?
No need to worry, get started today!
Explore Latest
Popular Dishes
Username
Cook Time: hr/min
Description:
Pizza
3
RecipeReel
CATEGORIES ∨
TOP RATED
Search
New Recipe
Recipe Name
Cook Time
30
min
Select Difficulty
Description
Ingredients
Instructions
Category
Choose file
No file chosen
Post Recipe

Use Case 7:

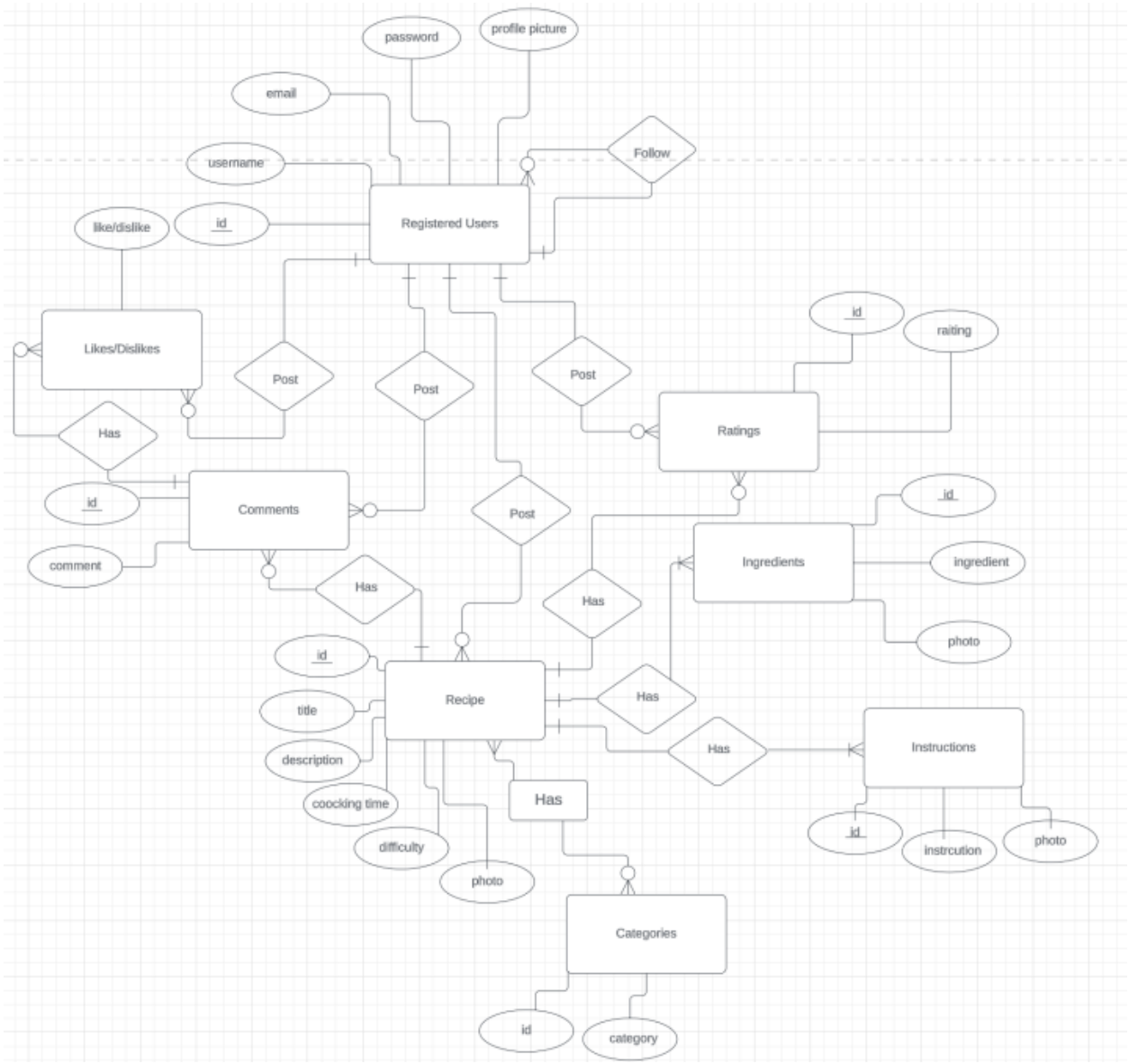

RecipeReel
CATEGORIES v
TOP RATED
CREATE POST
Find Dishes
That Suit You!
Looking for new recipes to spice up your day?
No need to worry, get started today!
Sign Up
Explore Latest
Popular Dishes:
Username
"Pizza"
Pizza
Cook time: 2h 30m
Ingredients list:
Description:
Recipe instructions:
Step 1:
Step 2:

Use Case 8:

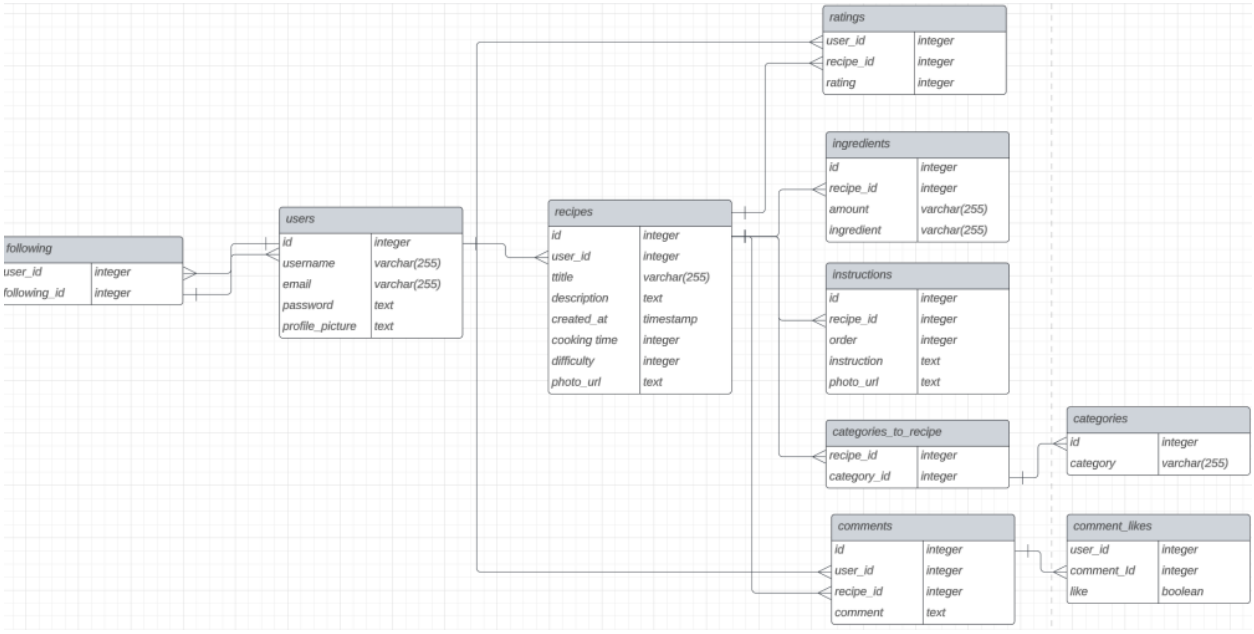

RecipeReel
CATEGORIES v
TOP RATED
Search
Find Dishes
That Suit You!
Looking for new recipes to spice up your day?
No need to worry, get started today!
Sign Up
Explore Latest
Popular Dishes:
Username
Cook Time: hr/min
Description:
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Pizza
3
RecipeReel
CATEGORIES v
TOP RATED
Cuisines
Time of Day
Dietary Restrictions
Find Dishes
That Suit You!
Popular Dishes:
RecipeReel
CATEGORIES v
TOP RATED
CREATE POST
Search
Dietary Restrictions
Username
Pizza
3

4. High level database architecture and organization V2

4.1 ERD Diagram

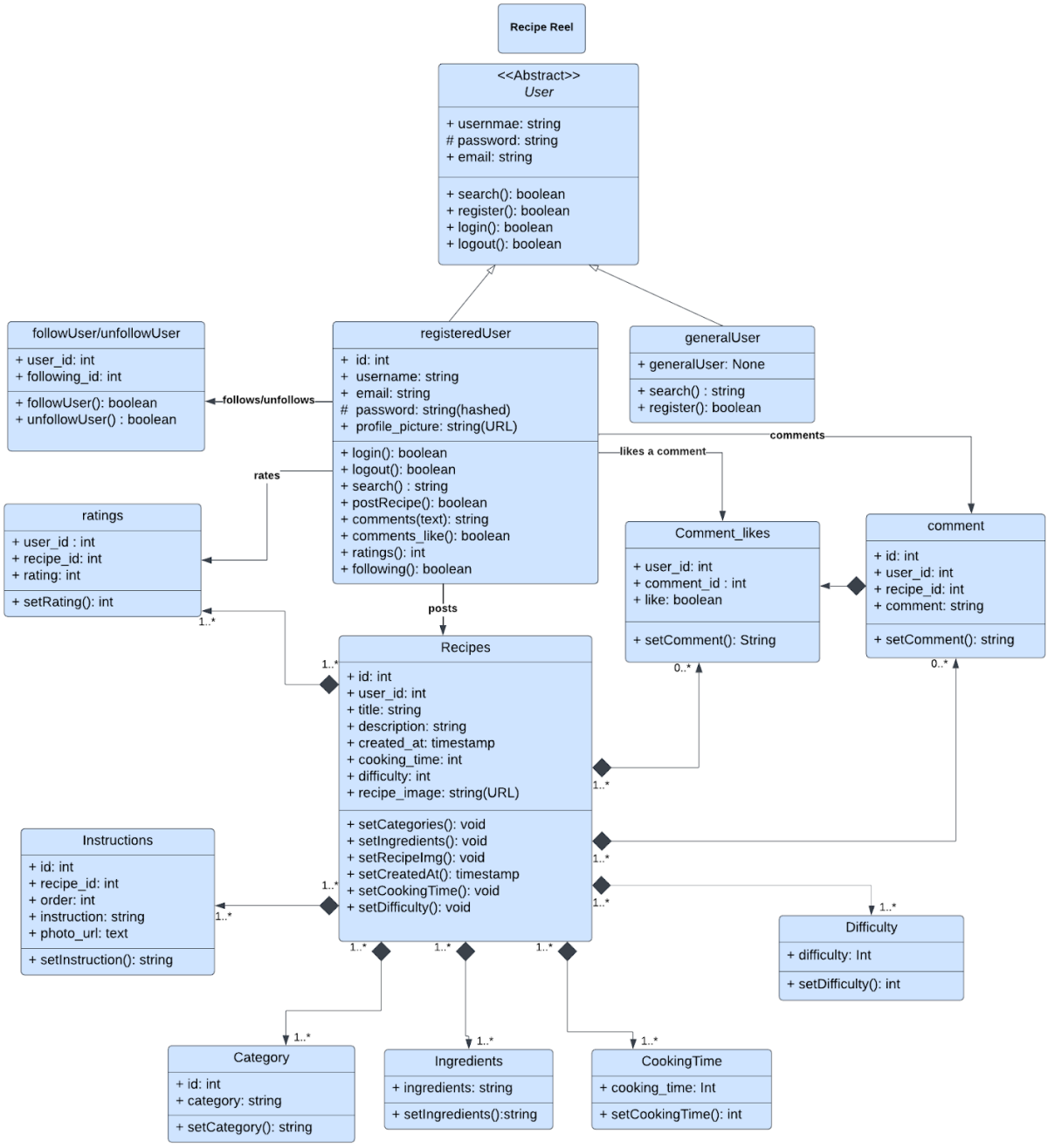


4.2 EER Diagram

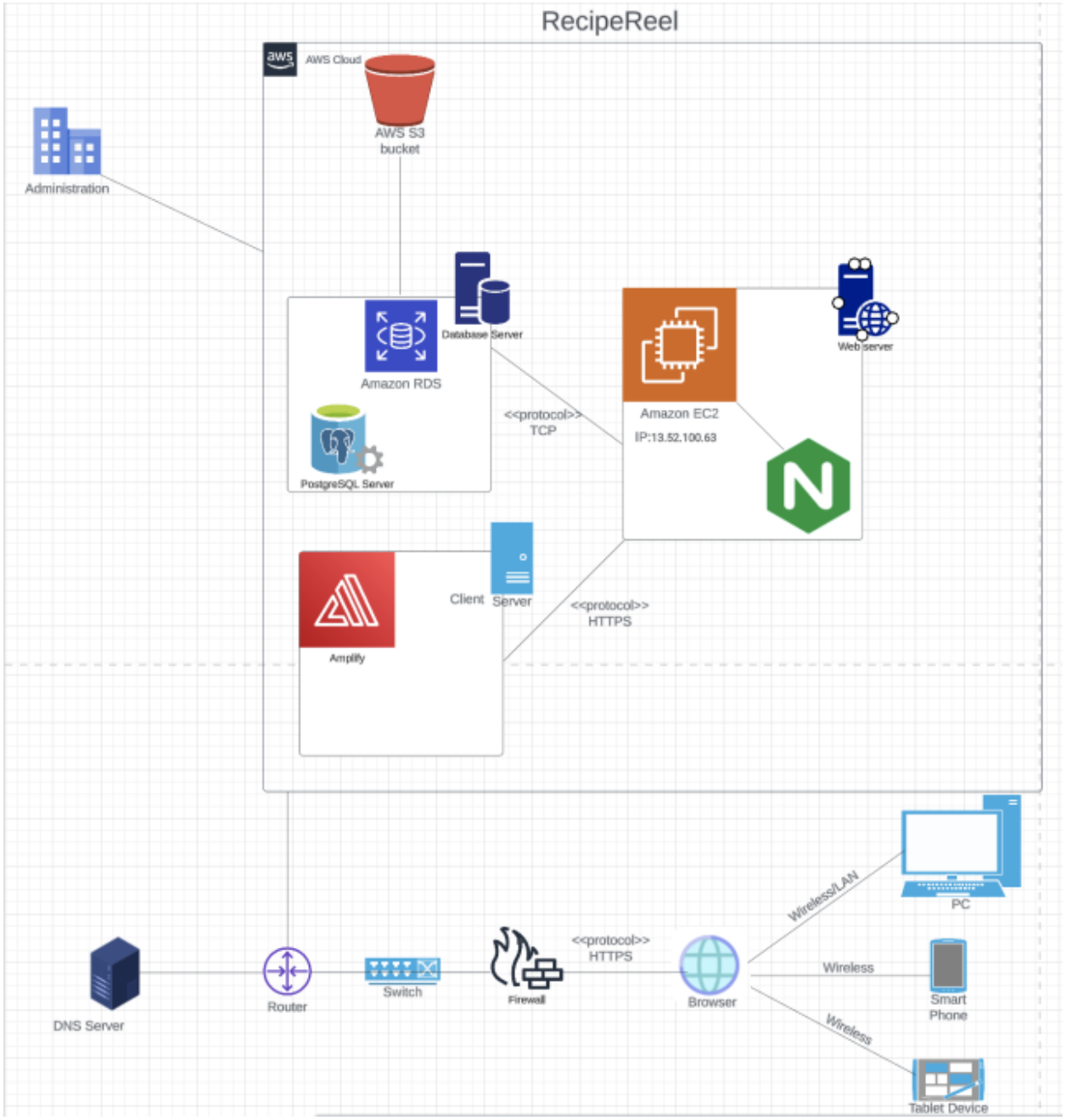


5. High Level Diagrams V2

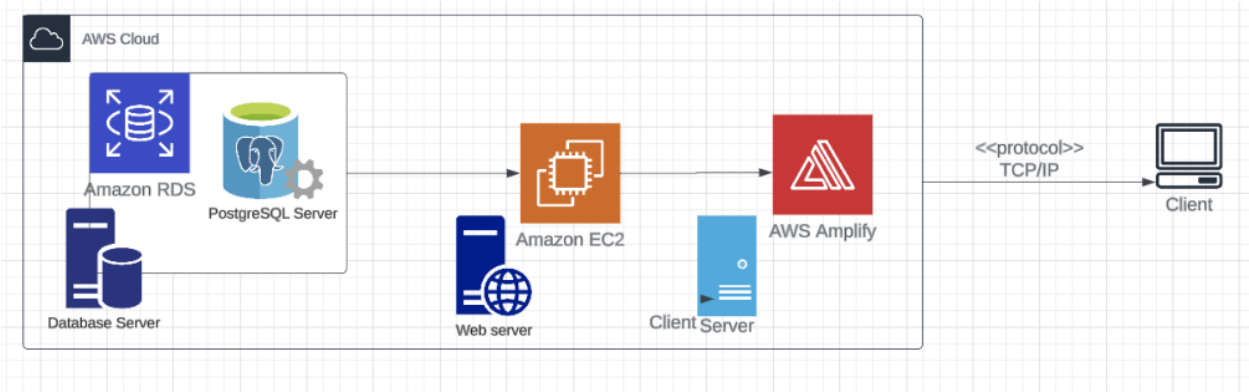
5.1 UML



5.2 Application Network Diagram



5.3 Deployment Diagram



6. Team Member Contributions

Team member	Contributions/Activities	Score
Team lead Yueling Liu	• Assigned tasks to each team member • Collaborated with team members • Organized meetings and followed up on progress • Designed login page in the frontend • Designed register page nn the frontend • Implemented validation for register page in the frontend • Reviewed documentation and provided feedback to team members • Reviewed front end code and provided feedback to team members	
Backend lead Duncan Herington	• Implemented route for login, logout, search in the backend • Connected login, logout, search routes to the front end • Implemented login validation in the back end • Implemented login validation in the front end • Added jwt tokens for login in the backend • Redesigned homepage postcards • Implemented the category drop-down menu to the navbar and made the navbar dynamic to both logged-in and logged-out users • Reviewed code in the back end • Helped fix High Level APIs and Main Algorithms of milestone 2 V2	9/10
Backend lead Marcel Azouri	• Created wireframe UI models • Created wireframe storyboards • Fixed storyboards for Milestone 2 V2 • Helped fix High Level APIs and Main Algorithms of milestone 2 V2	9/10

Github lead Nathan Le Howland	• Implemented user and recipe models in the back end • Seeded database tables • Implemented following/follower in the front end • Implemented following/follower in the back end • Implemented register endpoint in the back end • Completed AWS S3 integration • Responsible for managing and updating the project's GitHub repository • Managed regular updates and maintained the deployment of the application • Implemented user registration with profile picture upload functionality in the backend • Assisted Section 4 database ERD in M2 • Completed EER in M3	9/10

SW Engineering CSC 648-05 Spring 2023

RecipeReel

T03 Milestone 4

Team lead: Yueling Liu yliu50@sfsu.edu

Backend lead: Duncan Herington, Marcel Azouri

GitHub lead: Nathan Le Howland

History table

M4V1	May 18, 2023
M3V2	May 18, 2023
M3V1	April 27, 2023
M2V2	April 27, 2023
M2V1	April 3, 2023
M1V2	April 3, 2023
M1V1	March 2, 2023

Table of Contents

1. Product Summary

- Product Name: RecipeReel
- Selling our product:

  Welcome to the future of recipe sharing !RecipeReel is not just an average recipe website; it's a dynamic and engaging social platform designed to connect food enthusiasts from all walks of life. At RecipeReel, we believe that cooking is more than just following instructions; it's an art form that should be shared, and enjoyed together. Our platform goes beyond the traditional recipe website and it creates a vibrant and interactive space where food lovers can connect, inspire, and be inspired with one another. It provides a user-friendly interface that feels like a social media platform, users can browse recipes, they can rate a recipe,leaving a comment to engage with the author who posted the recipes when they are signed up and logged in. We are aiming to make cooking a social experience, building connections with like-minded individuals who share their love for cooking.

- Final Priority 1 Functions
  - General User:
    - A general user shall be able to register for an account on RecipeReel using a username, password, and email
    - A general user shall be able to search using a keyword or terms that match a recipe's ingredients or title
    - A general user shall be able to view recipes, details, and comments.
  - Registered User:
    - A registered user shall be able to log in with an email and password.
    - A registered user shall be able to leave comments on recipes.
    - A registered user shall be able to pst recipes to RecipeReel
    - A registered user shall be able to log out.
    - A registered user shall be able to follow other registered users.
    - A registered user shall be able to unfollow other registered users.

        - A registered user shall be able to delete a recipe they posted to RecipeReel.
- What makes RecipeReel unique?

  RecipeReel believes that cooking is not just about the food, but about community. In order to reflect this, RecipeReel behaves more like a social media platform rather than a traditional recipe website. This could help others discover the best recipe and create a fun and engaging and interactive space where food enthusiasts can come together to share their passion for cooking.

- URL to RecipeReel: https://master.d3u7tlcu99u8gi.amplifyapp.com/

2. Usability Test Plan

Usability Test Plan

The purpose of this usability test is to evaluate the usability of the five main functions of RecipeReel. The five functions to be tested are: Post a recipe, following a user, searching for a recipe based on keywords,rating on a recipe, and deleting a post of the recipe. The test measures the effectiveness, efficiency, and user satisfaction of each function.

- Post a Recipe
    - Test Objective:

      This test checks each post contains a recipe name, description, cooking time, difficulty level, ingredients, instructions and category. The reason it checks these inputs is that the recipe name is essential as it helps other users quickly find the recipe they are interested in by enforcing having a recipe name. And the description provides a brief overview of the recipe, which helps users understand if the recipes are the one they are looking for before diving into it. Cooking time allows the user to plan time accordingly. Difficulty level is crucial for users to gauge whether a recipe aligns with their cooking skills and experience. The list of ingredients is vital as it is an essential and necessary component for the recipe. This also helps users to search recipes by entering ingredients. We test the instructions as it provides clear directions on how to prepare the recipe, step by step instruction ensures users can follow along easily and achieve cooking their desired meal. Testing category helps users explore recipes based on their preferences, or specific cuisine. Images of recipes are highly beneficial and it makes the recipe appealing, but it is an optional. By enforcing these inputs in each post, the website can provide users with comprehensive and consistent information across all recipes.
    - Test Description:

      System Setup:

The system being tested is our website that allows registered users to post their own recipes. Our website has a user registration and login process in place to ensure that only authorized users can access the posting functionality. The system is accessible via a specific URL provided for testing purposes.

Starting Point:

The starting point for this test is the recipe posting page. This page serves as the entry point for users to enter information about their recipes and publish them on the website. Users are expected to navigate to this page after logging into the system.

Intended Users:

The target users for this test are registered users of the recipe website who wish to post their recipes. These users have already completed the registration and login process.

Measurements:

Time: The time it takes for a user to post a recipe will be measured. This includes the time spent on filling out the necessary fields, uploading images, and submitting the recipe. This measurement helps assess the efficiency of the posting process.

Clicks: The number of clicks made by a user during the recipe posting process will be recorded. This measurement helps evaluate the user interface's intuitiveness and the ease of navigating through the required steps. And the clicks are varies and depend on the complexity and difficulty of the recipe.

Ease of Process: The overall ease of the posting process will be evaluated, taking into account factors such as clarity of instructions, intuitiveness of the user interface. This measurement helps determine the user-friendliness of the recipe posting feature.

By conducting this test, we aim to ensure that the recipe posting process is user-friendly, efficient, and meets the expectations of registered users. These collected measurements will be used to identify any potential

issues, improve the user experience, and enhance the overall functionality of the recipe posting feature on the website.

URL: https://master.d3u7tlcu99u8gi.amplifyapp.com/post-recipe (It only for registered user)

- Comment on a posted recipe
    - Test Objective:

      The objective of this test is to ensure that registered users can leave comments on other users' posted recipes. We test the features below:

      1. Testing if a user enters a valid comment in the input field
      2. Testing the “Post Comment” button works properly
      3. Testing that the comment is successfully added to the recipe post
      4. Testing if the comment appears in the correct order

      This reason for testing commenting on other users' posted recipes focuses on fostering user engagement, as it plays a crucial role in creating interactions and building a sense of community among users. This aspect of the website aims to differentiate it from traditional recipe websites and provide a social media-like experience for users, and this is one of the unique features of our website. It also helps admins to evaluate if these comments are relevant to recipes.

    - Test Description:

      System Setup:

      The system being tested is our website that allows registered users to comment on other’s user’s posted recipes.

      Starting Point:

      The starting point of this test is a recipe post displayed on the home page or when a user searches for a recipe and that displays on the application's user interface.

Intended Users: The target users for this test are registered users of our recipe website who wish to comment on other users' posted recipes. These users have already completed the registration and login process.

Measurements:

Time: The time it takes for a user to comment on a recipe will be measured. This measurement helps assess the efficiency of the commenting process and evaluate their engagement with one another.

Comment Quality: Evaluating the quality of comments can be subjective but can provide insights into the level of user interaction and meaningful discussions.

User Engagement: Measuring user engagement who leave comments, the frequency of return visits by users to the recipe page, and the duration spent on the page. These help determine the overall user interest and involvement with the commenting feature.

- Searching for a recipe based on keywords
    - Test Objective: the objective of this test is to ensure that both registered and unregistered users can efficiently search for recipes using keywords, such as recipe names or ingredients.We test the features below:
        1. Testing the performance of the search
        2. Testing if the search query valid input returns correct results

      This reason for testing search is to make sure that all users have an efficient way of finding specific recipes that match their query. This will save the user time and enhance their experience.

    - Test Description:

      System Setup:

      The system being tested is the search feature, which is in the navigation bar. This search feature handles keywords to query recipes for the user.

      Starting Point:

The starting point of this test is any page on the website as the search bar is located in the nav bar which is on every page.

Target users:

The targeted users are all users on the website, that includes registered and unregistered users who want to search for specific recipes

Measurements:

Search Accurate:

Measure the accuracy of search results by comparing the relevance of the displayed recipes to the user's search query. This can be assessed through manual evaluation or automated methods, such as comparing search keywords with recipe titles, descriptions, and tags.

Search Completion Time:

Measure the time taken for the search feature to display relevant results after the user enters their search query. This helps assess the speed and efficiency of the search functionality.

URL: https://master.d3u7tlcu99u8gi.amplifyapp.com/ type ingredients in search bar field

- Rate a posted recipe
    - Test Objective: To test whether registered users can rate recipes and whether there are problems in the scoring process. We test:
        1. Regular users cannot rate recipes, since only registered users can rate them.
        2. Registered user can rate successfully

      The reason we test this is to gather feedback on the quality of the posted recipes. This function can provide valuable insights into what users find valuable, interesting. Rating systems can encourage user engagement and participation, and enhance user’s experience.
    - Test Description:

      System Setup:

The system being tested is our website that allows registered users to rate on other user's posted recipes.

Starting Point:

The starting point of this test is a recipe post displayed on the home page or when a user searches for a recipe and that displays on the application's user interface.

Target users:

The target users for this test are registered users of our recipe website who wish to rate on other users posted. These users have already completed the registration and login process.

Measurements:

Engagement: We measure if the posted recipe encourages other users' engagement

Satisfaction: By testing rating a posted recipe, we measure the satisfaction for users and quality of the recipe.

- Deleting a posted recipe
    - Test Objective: To test whether registered users can delete their own post or recipe content. This test verifies that the delete functionality is working properly and that users encounter any issues when attempting to delete a post.
        1. Test if the recipe actually exists, if the recipe does not exist we cannot delete it
        2. Test if the user that requests to delete the recipe is the owner of the recipe, if the user is not the owner of the recipe than they should not be allowed to delete it
        3. Test that the recipe has been deleted, we need to make sure that all data tied to this single recipe is deleted.

    - Test Description:

      System Setup:

The system being tested is our server. The front end sends data to the server in the form of URL params and request body. The server uses this data to delete data in the DB.

Starting Point:

The starting point for this test is the front-end client. The user must find a post that they own and delete it. The client then sends a DEL request to the server along with some data.

Target users:

The target users for this test are registered users of our recipe website who wish to delete their own posted recipes. These users have already completed the registration and login process.

Measurements:

Successful Deletion:

We need to make sure the post is deleted successfully. All the information such as recipe name, descriptions, comments, ratings, and images, are completely removed and no longer displayed on the user's interface after deleting a recipe.

Data Integrity:

We need to make sure the right recipe is deleted. We also need to delete all tables that have a FK of recipe_id. The measure of accuracy is if all this data is deleted from DB.

Usability Test Table - Effectiveness

Test Case	Completed %	Errors	Comments
Posting a Recipe	90%	Only 1 Image for each recipe is stored	We intended to allow user post a recipe with 1-6 maximum images, however we are able to store 1 image only for each recipe
Commenting on other user's posted recipes	80%	insert into "comments" for type integer is "undefined" for these posted recipes displayed on homepage	Commenting on recipes that have been searched from search bar worked and displayed on the comment filed
Searching for a recipe based on keywords	100%	None	It works when user search from home page and enter ingredient keywords
Rating a Post	0%	None	We haven't implemented this feature in this milestone
Deleting a Post	50%	It doesn't rendering in the front end	We have delete a post api set up and it worked, however we haven't completed this feature

Usability Test Table - Efficiency

Test Case	Time to Completion	Number of Clicks	Ease of Use	Comments
Post a Recipe	Easy recipe: 5-12 minutes; Intermediate recipe: 10-25 minutes: Difficult recipe: 15-25 minutes or even longer	Easy recipe: 30-45 clicks Intermediate recipe: 45-55 clicks Difficult recipe: up to 55 clicks	Satisfied	This test varies, and it depends on how difficult the recipe is. The more ingredients and instructions the longer it takes and the more clicks.
Commenting on other users posted recipes	Within 1 minute	2	Satisfied	It only worked for a searched recipe that displayed on the user's interface.
Searching for a recipe based on keywords	Within 1 minute	2	Very Satisfied	User can enter a ingredient keyword, such as rice, it will display all the recipes that contains rice
Rating a Post	Within 1 minute	2-5	Not Satisfied	Not working
Deleting a Post	Within 1 minute	2	Not Satisfied	Not working in the front end

User Satisfaction

User 1:

	Very Satisfied	Satisfied	Neutral	Unsatisfied	Very Unsatisfied
Posting a recipe was easy to understand		✓			
I felt confident that my recipe was posted successfully.		✓			
The restrictions on the number of images and characters were reasonable.			✓		
Finding and following other users was easy.					✓
The updates about the following users' content was clear and helpful.				✓	
Overall, I was satisfied with the process of following a user.					✓
The search results were relevant to the keywords entered.	✓				
I was able to filter and refine my search results as needed.				✓	
Overall, I was satisfied with the search functionality.	✓				
Rating a recipe was easy.					✓
The rating system was fair and accurate.					✓
I felt that my rating would be helpful to other users.				✓	
Deleting a post was easy to understand.		✓			
I felt confident that my post was successfully deleted.		✓			
Overall, I was satisfied with the ability to delete my own posts.			✓		
Commenting on a post was easy to understand	✓				
I felt that my comment was posted successfully		✓			

Overall, I was satisfied with the ability to comment on a posted recipe .			✓		

User 2:

	Very Satisfied	Satisfied	Neutral	Unsatisfied	Very Unsatisfied
Posting a recipe was easy to understand	✓				
I felt confident that my recipe was posted successfully.	✓				
The restrictions on the number of images and characters were reasonable.				✓	
Finding and following other users was easy.					✓
The updates about the following users' content was clear and helpful.				✓	
Overall, I was satisfied with the process of following a user.					✓
The search results were relevant to the keywords entered.	✓				
I was able to filter and refine my search results as needed.					✓
Overall, I was satisfied with the search functionality.	✓				
Rating a recipe was easy.					✓
The rating system was fair and accurate.					✓
I felt that my rating would be helpful to other users.			✓		
Deleting a post was easy to understand.		✓			
I felt confident that my post was successfully deleted.			✓		
Overall, I was satisfied with the ability to delete my own posts.			✓		
Commenting on a post was easy to understand	✓				

I felt that my comment was posted successfully	✓				
Overall, I was satisfied with the ability to comment on a posted recipe .		✓			

3. QA Test Plan

3.1. QA Test 1

3.1.1. Test Objectives: “Create Post” required fields.

3.1.2. HW & SW Setup: Windows computer on Google Chrome

3.1.3. Feature to be tested: Checking to see if all fields are filled out before a post can be made.

Test #	Description	Input	Expected Output	Results
1	Test to see if post can be created with only recipe name, ingredients and instructions	Filled fields: Recipe name Ingredients Instruction	A warning error: “Oops! It looks like you forgot to select a cooking time for your recipe”	Pass
2	Test to see if the post can be created with recipe name, ingredients, instructions, cooking time, and difficulty filled out.	Filled fields: Recipe name Cooking time Difficulty Ingredients Instruction	A warning error: “Oops! It looks like you forgot to enter a name for your recipe”	Pass
3	Test to see if posts can be created if all fields are filled out.	Filled fields: Recipe name Description Cooking time Difficulty Ingredients Instruction Category	A warning error: “~ops! It looks like you forgot to select a cooking time for your recipe”	Pass

3.2. QA Test 2

3.2.1. Test Objectives: Creating a password

3.2.2. HW & SW Setup: Windows computer on Google Chrome

3.2.3. Feature to be tested: Checking to see if a password is greater than or equal to 6, less than or equal to 20, and contains a special character.

Test #	Description	Input	Expected Output	Results
1	Test password less than 6 characters	“hi!”	“Password must be 6-20 characters long and contain a special character.”	Pass
2	Test password greater than 20 characters	“PrettyPleaseGiveUsGoodGradeProfessor!!!”	“Passwords must be 6-20 characters long and contains a special character.”	Pass
3	Test password greater than 6 characters, less than 20 characters, and containing a special character	“Pass123!”	“created account successfully”	Pass

3.3. QA Test 3

3.3.1. Test Objectives: Load times

3.3.2. HW & SW Setup: Windows computer on Google Chrome

3.3.3. Feature to be tested: Check to see how fast loading parts of the website.

Test #	Description	Input	Expected Output	Results
1	Test load time logging in	Login	Less than 10 seconds	Pass
2	Test load time	Top Rated	Less than 10	Pass

	going to “Top Rated”		seconds	
3	Test load time of search	Chicken	Less than 10 seconds	Pass

3.4. QA Test 4

3.4.1. Test Objectives: Works on other browsers

3.4.2. HW & SW Setup: Windows computer on Google Chrome

3.4.3. Feature to be tested: Make sure the website is functional on a variety of browsers.

Test #	Description	Input	Expected Output	Results
1	Test if RecipeReel works on Google Chrome	Google Chrome	Working	Pass
2	Test if RecipeReel works on Microsoft Edge	Microsoft Edge	Working	Pass
3	Test if RecipeReel works Firefox	Firefox	Working	Pass

3.5. QA Test 5

3.5.1. Test Objectives: Works on other operating systems

3.5.2. HW & SW Setup: Google Chrome

3.5.3. Feature to be tested: Make sure the website is functional on a variety of operating systems

Test #	Description	Input	Expected Output	Results
1	Test if RecipeReel work on Mac	None	Working	Pass
2	Test if RecipeReel work on Windows	None	Working	Pass

3	Test if RecipeReel work on Linux	None	Working	Pass

4. Code Review

4.1. Coding Style:

Our application is using the standard React framework, and we have followed:

- Indentation and formatting :
    - We used used “Prettier”extension on Visual Studio Code that make sure a consistent indentation style of two spaces
    - Used proper line breaks and indentation to improve code readability

- Naming conventions:
    - We followed the convention using camelcase for handler functions and variables(e.g, handlePostRecipe, recipeImage)
    - We used Reserve PascalCase for React component names (e.g., CategoryCard and PopularDIshesCards in our application)

- Component Lifecycle and Hooks:
    - Follow the recommended guidelines for using useEffect, useState, useContext, and other React Hooks

- Styling:
    - Used SCSS to styling pages that support nesting, allowing nest CSS selectors within one another. This helps improve readability and maintainability by grouping related styles together. We used used media query to ensure website is responsive with different size of devices
    - Used Bootstrap that helped us quickly build responsive pages,it simplified the process of creating a consistent user interface

4.2. Chose the code

```
import React, { useEffect, useState } from 'react';
import DashboardCard from '../components/Cards/DashboardCard';
import Filterbar from '../components/filterbar/Filterbar';
import { useHistory } from "react-router-dom";
```

```
// MUI
import MenuItem from '@mui/material/MenuItem';
import FormControl from '@mui/material/FormControl';
import Select from '@mui/material/Select';

// Bootstrap
import Col from 'react-bootstrap/esm/Col';
import Row from 'react-bootstrap/esm/Row';
import Container from 'react-bootstrap/esm/Container';

const Search = ({ location }) => {
 const [results, setResults] = useState([]);
 const [title, setTitle] = useState('');

 const [filter, setfilter] = React.useState('');

 const handleChange = (event) => {
   setfilter(event.target.value);
 };

 const history = useHistory();

 useEffect(() => {
   const fetchResults = async () => {
     const query = new
URLSearchParams(location.search).get('query');
     try {
       const response = await fetch(
         `${process.env.REACT_APP_REQ_URL}/search?query=${query}`
       );
       const data = await response.json();

       if (response.ok) {
         setResults(data.results);
         console.log("Results:", data.results[0].recipe_title);

         setTitle(data.results[0].recipe_title);
         console.log("Recipe title:", title);

       } else {
         throw new Error(data.error);
       }
```

```
    } catch (err) {
      console.error(err);
    }
  };

  fetchResults();
}, [location.search]);

 const handleDashboardCardClick = (recipe_id) => {
  history.push(`/post/${recipe_id}`);
};

return (
  <>
    <Container>
      <Filterbar title={title}/>

      {results.length > 0 ? (
        results.map((result) =>
         <DashboardCard
            key={result.recipe_id}
            result={result}
            onClick={() =>
handleDashboardCardClick(result.recipe_id)}
          />)
      ) : (
        <p style={{textAlign: 'center'}}>No results found.</p>
      )}
    </Container>
  </>
);
};

export default Search;
```

Why did we choose this code?

This is one of the main functions of our website. We used React Components that handles search functionality and renders a list of searches using a card.

Code review feedback from a peer in our team:
Grouped the MUI imports and Bootstrap imports separately to improve code readability which is a good practice.
It seems the MenuItem, FormControl, and Select imports from MUI are not used in the code. It should be removed to avoid unnecessary imports.
Error Handling seems to only catch any errors that occur during the fetch request, and only log the error to the console. Maybe adding error handling to displaying an error message to the user would be a better user experience.
Overall, the code looks well-structured and readable. The useEffect hook is used correctly to fetch data based on the search query.

Code review feedback from Team02 :
Code looks well formatted and looks fairly self-explanatory. However, the overall purpose of the code is a little unclear since there are no function headers or inline comments. And just a minor thing, but you use both “useState” and “React.useState”. Is there a reason for using one and then the other?

5. Self-check: Best practices for safety (½ page) -

5.1. Major protected assets

5.1.1. User password:

To encrypt the password in the DB we use bcrypt to encrypt the password. (see screenshot 1)

We store this hashed password to the database. When a user logged in we use bcrypt to compare our hashed password to the user supplied password. If there is a match, the user is validated.(see screenshot 2)

For example, when a user registers an account with a username : nate, password: pass123, it will be hashed and stored an encrypted string in the password column in our database. (see screenshot 3)

```
// Hash the password
const hashedPassword = await bcrypt.hash(password, 10);
```

Screenshot1 : using bcrypt

```
// take in password and compare with user password from email
const passwordMatch = await bcrypt.compare(password, dbUserData.password);
```

Screenshot 2: comparison with user password

	id [PK] integer	username character varying (255)	email character varying (255)	password text
1	1	nate	nate@email.com	$2b$10$q5yXtDVCra0b8qfJlC8CCuNph4jIFFvvQAqpoTRftoc6s9lY25DYi

Screenshot 3: encrypted password in database

5.2. Input data validation list

5.2.1. Username:

We have a validation for the username input, which occurs in both the frontend and backend to ensure data integrity.

In the front end, the username must be a minimum of 6 characters long and a maximum of 20 characters long. This is to enforce a specific length requirement for usernames.
We used a if statement to check the length of password:

```
  if (userName.length < 6 || userName.length > 20) {
    errors.username = 'Username must be between 6 and 20
characters';
    setValidationErrors(errors); // set validation errors state
    console.log(setValidationErrors(errors));
    toast.error('Username must be between 6 and 20 characters', {
      position: toast.POSITION.TOP_CENTER,
    });
    return;
  }
```

In the back end, our system uses a regular expression pattern to validate the username. It checks if the username consists of any alphanumeric characters or underscore and hyphen. And it has to be at least 6 characters long but not more than 20 characters long.

```
const usernameRegex = /^[a-zA-Z0-9_-]{6,20}$/;
if (!usernameRegex.test(username)) {
    console.log('Invalid username from Backend');
    return res.status(400).json({
      message:
        'Username must be 6-20 characters and may only contain
letters, numbers, underscores, and hyphens.',
    });
  }
```

5.2.2. Password:

We have a validation for the password input, which occurs in both the frontend and backend to ensure data integrity. The password must be a minimum of 6 characters long and a maximum of 20 characters long with a special character.

We used regular expressions to validate passwords in both frontend and backend.

In the front end we check if the password matches this regular expression pattern.

```
  if (!password.match(/^(?=.*[\W_])[a-zA-Z0-9\W_]{6,20}$/)) {
    errors.password =
      'Password must be 6-20 characters long and contains a
special character.';
    toast.error(errors.password, {
      position: toast.POSITION.TOP_CENTER,
    });
    return;
  }
```

In the backend, we have a const passwordRegex variable and we check if the input matches this pattern:

```
  const passwordRegex = /^(?=.*[\W_])[a-zA-Z0-9\W_]{6,20}$/;
  if (!passwordRegex.test(password)) {
    console.log('Invalid password from Backend');
    return res.status(400).json({
      message:
        'Password must be a combination of letters, numbers, and
special characters, with a maximum length of 20 characters.',
    });
  }
```

5.2.3. Email:

We have a validation for the email input, which occurs in both the frontend and backend to ensure email integrity.
In both front end and back end we used a regular expression used to validate email addresses.

In the front end we check if the email matches this regular expression pattern.

```
  if (!email.match(/^[^\s@]+@[^\s@]+\.[^\s@]+$/)) {
```

```
    errors.email = 'Please enter a valid email address';
    toast.error(errors.email, {
      position: toast.POSITION.TOP_CENTER,
    });
    return;
  }
```

In the back end, we have a const emailRegex variable and we check if the input matches this pattern:

```
  const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
      if (!emailRegex.test(email)) {
    console.log('Invalid email from Backend');
    return res.status(400).json({ message: 'Invalid email
address.' });
  }
```

5.2.4. Policy and agreement

The policy and agreement checkbox on a register page when a user registers an account on our website, this checkbox must be selected to indicate their acceptance and agreement to the website's terms of service, privacy policy, or any other legal agreements or policies that govern the use of our website.

For policy and agreement input, we only need to validate this input in the front end.

```
const [isChecked, setIsChecked] = useState(false);

 if (!isChecked) {
    toast.error(
      'Please agree to the Privacy Policy before submitting the
form.',
      {
        position: toast.POSITION.TOP_CENTER,
        className: 'toast-message',
      }
    );
```

```
    return;
  }
```

5.2.5. Search bar

Search bar validation involves functions that work correctly and meets specific criteria which are ingredients for our website. Our system checks if it is an empty query and displays a prompt No results found message.

We are validating the search accuracy which we first need to validate the search functionality to retrieve recipes that match the provided query correctly.

Code we used:

```
 static async search(query) {
   // return await knex('recipes')
   // .join('categoriesToRecipe','categoriesToRecipe.recipeID',
'recipes.ID')
   // .join('categories', 'categories.ID',
'categoriesToRecipe.categoryID')
   // .where('category', 'like', `%${query}%`)
   // .orWhere('title', 'like', `%${query}%`);
   const searchQuery = `
   WITH search_result AS (
     SELECT DISTINCT r."id" AS recipe_id, r."title" AS
recipe_title, r."description" AS recipe_description,
r."created_at" AS recipe_CreateAt, u."username" AS userName
     FROM public.recipes r
     JOIN public.users u ON r."user_id" = u."id"
     LEFT JOIN public.categories_to_recipe ctr ON r."id" =
ctr."recipe_id"
     LEFT JOIN public.categories c ON ctr."category_id" = c."id"
     LEFT JOIN public.comments cm ON r."id" = cm."recipe_id"
     LEFT JOIN public.ingredients i ON r."id" = i."recipe_id"
     LEFT JOIN public.instructions ins ON r."id" = ins."recipe_id"
     LEFT JOIN public.ratings rt ON r."id" = rt."recipe_id"
     WHERE (
         r.title ILIKE '%${query}%'
```

```
        OR r.description ILIKE '%${query}%'
        OR c.category ILIKE '%${query}%'
        OR u.username ILIKE '%${query}%'
        OR cm.comment ILIKE '%${query}%'
        OR i.ingredient ILIKE '%${query}%'
        OR ins.instruction ILIKE '%${query}%'
    )
  )
  SELECT * FROM search_result
  ORDER BY recipe_id ASC;
`;
  const { rows } = await client.query(searchQuery);
  return rows;
}
```

6. Self-check: Adherence to original non-functional specs

6.1. System requirements

6.1.1. The website shall be able to use client – server. DONE

6.1.2. The website shall be able to handle when visitor scale is high. DONE, AWS handlings load balancing

6.1.3. The website shall be able to support both cellular and Wi-Fi networks - DONE

6.1.4. The website shall be able to support any user using any web browser. DONE

6.1.5. The database host shall be able to have a web server to host. DONE RDS is hosting the DB

6.1.6. The database host network shall be able to have good security. DONE

6.2. Performance requirements

6.2.1. The website reload shall be able to provide fast service. DONE

6.2.2. The website shall be able to have minified code to reduce file size.

6.2.3. The website shall be able to optimize image quick. DONE

6.2.4. The website shall be able to use a performance monitoring tool to catch bottlenecks. DONE, AWS amplify and EC2 provide monitoring tools

6.2.5. The website database server shall be able to expect multiple users without any delay. DONE

6.2.6. The website database server shall be able to expect multiple requests. DONE, built into AWS RDS

6.3. Privacy

6.3.1. The website shall be able to provide clear and concise policies. DONE

6.3.2. Users shall be able to agree on how their openly information may be shared. DONE

6.3.3. The website shall be able to collect data like name, date of birth, email, and more personal information. DONE

6.3.4. The website shall be able to avoid collecting unnecessary data. DONE

6.3.5. The collected data shall be able to be used for improvement of the website and user experiences. DONE

6.3.6. All confidential user data shall be able store encrypted data. DONE

6.3.7. For more protection firewalls and protection systems shall be able be used. DONE, AWS handles this

6.3.8. Users shall be able to be trained not to share sensitive information. DONE

6.4. Storage

6.4.1. The website shall be able to have a good amount of storage that can handle contests like food images, food recipes, food description and more. DONE, AWS EDS provides sufficient storage for this app at FREE TIER

6.4.2. The website shall be able to support all file formats and sizes for better and effective performance. NONE.

6.4.3. The website shall be able to have a limit on the maximum file size that can be uploaded to prevent excessive resource usage. NONE

6.4.4. The website shall be able to have another storage for emergency backups. NONE

6.5. Security

6.5.1. The website shall be able to use HTTPS. (Secure Transfer protocol). DONE, SSL generated with certbot

6.5.2. The website shall be able to implement security measures to protect user data. DONE, passwords are encrypted

6.5.3. The website shall be able to have a strong password policy. DONE

6.5.4. The website shall be able to use a password transcription method. NONE

6.5.5. The website shall be able to offer multi-factor authentication to prevent unauthorized access. NONE.

6.6. Marketing and Legal requirements

6.6.1. Popular social media sites shall be able to be used to promote websites. NONE

6.6.2. The website shall be able to be user friendly so any user can browse comfortably. DONE

6.6.3. The information provided on the website shall be able to be clear for anyone to understand. DONE

6.6.4. The website shall be able to select a specific target audience for marketing. DONE.

6.6.5. Links of the website shall be able to be pushed through emails and other forms. DONE

6.6.6. The website shall be able to be branded. DONE

6.6.7. The website shall be able to have a name and logo. 80% DONE, this website has a name but not a logo

6.6.8. The website shall be able to accept feedback. NONE

6.6.9. The website shall be able to protect itself-from any legal claims. DONE

6.7. Content

6.7.1. The website will be using a clean and simple sans-serif font as it is easy to read and doesn't detract from the content. DONE

6.7.2. The website will be using a clear hierarchy for headings, subheadings, and other important information. Additionally, using bullet points or numbered lists for ingredients and instructions can make the recipe easier to follow. DONE

7. Team Member Contributions

Team member	Contributions/Activities	Score
Team lead Yueling Liu	● Completed section 1 Product Summary ● Completed section 2 Usability Test plan for Post a recipe and comment on a posted recipe ● Completed section 3 Usability effectiveness test ● Completed section 3 Usability efficiency test ● Completed section 3 User Satisfaction Survey with 2 users who are not in CS field ● Completed section 4 Code review ● Completed section 5.2 Input data validation list ● Completed section 6 Non-functional specs checkup ● Front end feature for posting a recipe ● Added delete post feature in the FE ● Delete a post API call in the FE	
Backend lead Duncan Herington	● completed section 2 Usability Test plan for Search recipe ● Created comment API ● Connected comment API to front end ● Assisted with connecting post recipe to API	9
Backend lead Marcel Azouri	● Completed section 1 ● Tested Non-functional requirements	7
Github lead Nathan Le Howland	● Completed section 2 Usability Test plan for delete a posted recipe ● Completed section 5.1 protected assets ● Completed section 6 non-functional specs checkup ● Created Post recipe API	9

	● Created Save a recipe API ● Created a like a comment API ● Created a dislike comments API ● Created a get all recipe API ● Maintained database	

4. Team Member Contributions

Team member	Contributions/Activities	Score	Signature from team member
Team lead Yueling Liu	● Completed P1 Function requirement in Front end ● Maintained code maintainability and organization ● Designed responsive web: ensured that websites and web applications worked on were optimized for different devices and screen sizes ● Project Planning and Coordination: Collaborates with team members to define project requirements, timelines ● Communication and Collaboration : Organized regular team meetings, stand-ups, and discussions to ensure everyone is aligned and aware of project progress, challenges, and updates. ● Document reviews and feedback: Conducted reviews of the documentation to ensure its quality, completeness		Yueling
Backend and front end lead Duncan Herington	● Completed P1 function requirement in Front end ● Maintained code maintainability and organization: Wrote clean, modular, and reusable code. ● Collaboration and Communication: Actively participated in team discussions, shared their ideas, kept other team members updated with great communication ● Assisted technical help: Assisted with technical challenges, review code, provide feedback, and help improve the team's overall technical skills and knowledge in the front end ● Coded front end for favoriting ● Coded front and back end for comments ● Coded front end for following and followers	10	Duncan Herington

Document lead: Marcel Azouri	• Collaboration and Communication: collaborated with frontend developers, UI/UX designers • Document reviews and feedback: Conducted reviews of the documentation to ensure its quality, completeness • Web UI design: Utilized Figma to explore and visualize various design options for the user interface • Collaborated with with front end lead, efficiently implement UI designs	10	Marcel Azouri
Github/ database lead Nathan Le Howland	• API Development: developed efficient APIs for P1 functional requirements • Database Management: designed and implemented database schemas/tables, optimized queries for performance, and ensure data integrity and in a good architecture • Deployment and Infrastructure: Set up deployment, managed the infrastructure for hosting backend applications, worked with AWS to deploy and scale applications efficient • Collaboration and Communication: Actively participated in team discussions, shared their ideas, kept other team members updated with great communication	10	Nathan Le Howland

5. Post analysis

Leading this team has taught me the invaluable lesson of building trust and the importance of communication.

One of the major challenges I encountered before the team splitting was the lack of effective communication among team members. I have faced difficulties in reaching out to one another and had limited visibility into the progress of individual tasks. This created a sense of disconnect and hindered our ability to work cohesively as a team.

Despite the efforts to address the communication challenges within the team and give everyone an opportunity to improve, the situation did not progress as I expected. We experienced team conflict and disagreements among the team . As a result, we had to make the difficult decision to split the team. Team splitting can be a challenging decision to make, and I felt as though it was a failure of my leadership, but sometimes it is necessary to ensure project success and maintain a positive working environment. By reorganizing the team, we aimed to address any issues, and create a more suitable framework for achieving our goals..

Moving forward, the split teams actually benefited from a fresh start, with improved communication structures and a better understanding of the importance of effective collaboration. The rest of us have been very cooperative with one another.

The second challenge was time management. After splitting the team, balancing priorities and deadlines became a significant challenge for us. Github master had to be responsible for github, database, as well as backend. Duncan and I had to work on all front end features and debugging, coding and code reviewing. It required effective time management skills to ensure that tasks are completed on schedule and that team members have the necessary resources to accomplish their goals. As a result, we left some incomplete features we intended to implement at the beginning.