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: Priya Pradeep

GitHub lead: Nathan Le Howland

Database lead: Samuel Elias

Document lead: Yasson Haddish

History Table

M2	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 shall contain letters, numbers, and special characters such as underscores or hyphens. 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 eight characters long. 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 on the RecipeReel website. Providing a registration function that allows general users to access additional features. For example, once a user registered for an account, it enables users to leave comments and ratings on recipes, follow other users, and post their own recipes.

- A general user shall be able to use specific terms, ingredients or keywords to help narrow down recipes that might fit their tastes using the search feature. This feature gives users search options and filters the results of their searches. This feature enhances our website's usability and functionality. Users would have to wait through numerous pages of recipes without a search tool, which can be tedious and time-consuming.

- A general user shall be able to scroll through recipes that are posted on RecpieReel. This function allows general users to view the content and explore the recipes on the website.

- A general user shall be able to search different categories of food, such as Italian food, Chinese food, Indian food, Mexican food etc.. This feature aids in users' exploration of new recipes, as well as helps users discover or re-experience their own cultural dishes.

- A general user shall be able to search different recipes based on ingredients, such as chicken, beef, pork, seafood, vegetables, etc. Users benefit from a more flexible and customized experience thanks to this feature. For example, if a user wanted to make a meal using beef they had recently purchased, they could use this feature to quickly and easily find recipes that contain beef.

- A general user shall be able to search for recipes based on cooking time. This is useful for users who are cooking on a time limit, who are looking for quick and easy recipes that have cook times that fit in their schedules.

- A general user shall be able to view the recipe details and information. This provides users with the necessary information pertaining to the

dish itself, such as ingredients, cooking time, difficulty level, as well as special instructions such as cooking techniques used for specific particular dishes.

2.1.2 Registered Users

- A registered user shall be able to log in. A logged in user should be able to use all registered user functions such as save their favorite recipes, like and leave a comment on a recipe, post their own recipes, and follow other authors/chefs. A logged in user should also be able to log out.

- A registered user shall be able create a post of a recipe and share it. This post will be shared to people that follow the registered user and to a general feed. At a minimum, a post will contain the recipe title, the author of the recipe, the ingredients list needed, the instructions to follow, and a picture of the dish.

- A registered user shall be able to view a feed of recipes based on users/chefs that they choose to follow. This feed will consolidate all of the users' followed accounts into one feed and also allow the user to interact with the posts. This can be displayed chronologically or by popularity.

- A registered user shall be able to follow other users/chefs. This builds a network of users/chefs that are personal to the user. Everyone’s palette is different and through building this network, users can build a collection/feed of recipes tailored to their taste.

- A registered user shall be able to leave comments/feedback on a recipe. These comments can serve as reviews or general help and guidance for the next person to use the recipe. Comments should be INSTAGRAM style (endless scroll of posts). Posts should also allow replies that can be made to comments so that users can discuss.

- A registered user shall be able to like/dislike comments. This is important so that other users can see if a comment, such as a suggestion to change the recipe, is valuable or not. This system could also bring a question to the attention of the author.

- A registered user shall be able to like/dislike a recipe. This is important so other users can get a sense if a recipe is good or not. This also can be used to judge a recipe's popularity.

- A registered user shall be able to delete a post . This allows the user to take down a recipe that they no longer would like to be posted. This feature gives 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.
-

2.1.3 Admin

- An admin shall be able to monitor and update all the content on our website. Admins are responsible for adding, editing, or removing the web content ,links, comments, or images on our website. This function is to make sure all the current content on the website is relevant to the user's needs.

- An admin shall be able to manage website functionalities. Admins are responsible for ensuring that the website functions properly, including troubleshooting technical issues and updating software as necessary.

- An admin user shall be able to leave reviews on recipes. This is beneficial for users as it helps showcase certain recipes and therefore increase traffic and search ranking for the website. It can also help build trust with the users, and encourage word-of-mouth marketing for the website.

- An admin shall be able to analyze website performance. Admins can know if our website is user-friendly or causes issues for users, and can make improvements based on the website performance, such as what content should be added to the website.

- An admin shall be able to keep user information and data stored safely, as well as have everything confidential stored encrypted. These are important for both admins and users. If user data is not stored securely, it can be vulnerable to hacking and causes data breaches. And this can also build a good trust with users. If user data is hacked or breached, they will lose their interests in our website and will hurt our website reputation as well.

- 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.

2.2 Desired functionalities

2.2.1 General Users

- A general user shall be able to register using a phone number. This is beneficial in terms of security as the phone number is unique to the individual and the password is only known by the user, the website is able to protect the user's information from being accessed by malicious users. This helps to ensure that the user's data is safe and secured.

- A general user shall be able to filter by author. The user will be able to filter posts through authors. This will allow the user to find content written by their favorite accounts, or to discover new accounts that align with their interests.

- A general user shall be able to search different recipes by the newest posted. A "newest recipe" page can keep the content on the website fresh and up-to-data. Being able to quickly view the newest recipes can help users who post to be discovered, as well as provides users searching for recipes and an easy way of finding new recipes they have not tried before.

- A general user shall be able to search recipes by the best reviewed. This feature helps users find dishes that are more likely to be worth their time and effort, instead of taking the chance of being the first to try a new dish that ends up being lackluster.

- A general user can sort search based on lowest or highest calorie count. This is beneficial as it allows users the option to browse the site for meals that fit their caloric range.

- A general user shall be able to search for recipes based on different dietary requirements, including gluten-free, vegan, vegetarian and low-carb. This feature is for users with specific dietary restrictions, allergies, and health goals. This makes it very useful for users adapting to new dietary plans or seeking self-improvement through their food choices.

- A general user shall be able to see promotions. Promotions can provide users with information about products or services that may be of interest to them. This function can potentially benefit our website, and we can make a profit from these advertisements.

2.2.2 Registered Users

- A registered user shall be able to verify their phone number. The user will be able to verify their phone number by entering their number into a text box, after which they will receive a text message containing a code. The user will then enter this code into the website to verify their phone number.

- A registered user shall be able to verify using 2FA. The user will be able to verify using two factor authentication with their email. This is achieved through the user receiving an email with a code in which they will input on the website.

- A registered user shall be able to edit their profiles, providing users with the ability to edit basic profile information such as their name, profile picture, and bio. This function would allow users to customize their profile by adding additional information, such as their favorite cruise.

- A registered user shall be able to not see promotions. This is beneficial as it helps to reduce distractions for users who are searching for recipes and other content. It also helps to ensure that users are not overwhelmed with promotional messages or advertisements.

- A registered user shall be able to download the recipes from the website. The user should be able to select the recipes they want to download and then have the option to download them as a PDF, Word document, or any other file format they desire.

- A registered user shall be able to search for recipe details by difficulty level. This is beneficial as it helps to ensure that users are able to quickly identify recipes that are within their skill level.

- A registered user shall be able to archive their posts. The user will be able to archive their post by clicking the “Archive” button located on the post. This will move the post to a separate section of their profile and will no longer be visible to other users.

- A registered user shall be able to edit security settings such as their email and password. To ensure that the user is authorized to change in these settings, the user must enter in the account password again. This guarantees that the owner of the registered account is making these changes.

- A registered user shall be able to leave a rating on a post. The user will be able to leave a rating on a post by using a scale of 1 to 5 stars.

- A registered user shall be able to save a post. This is beneficial for the user as they can avoid looking up favorite recipes later and avoid time consumption.

- A registered user shall be able to edit their post, this includes changing any contents of the post such as title, description, ingredients, instructions, photos, and so on. This also gives the user the capability to delete the post completely.

2.2.3 Admin

- An admin user shall be verified using two-factor authentication. It is beneficial for the users as it can reduce account takeover danger and increase account security. It also allows users to access data from any device without compromising sensitive information.

- An admin user shall be able to verify using their phone number. It is beneficial in validating the availability and validity of the client-provided number through phone verification. In order to combat fraud, it can also be used as an extra method of client identity verification.

2.3 Opportunistic functionalities

2.3.1 General Users

- A user shall have a meal planner tool. This tool can provide meal suggestions to the users based on their dietary preferences, restrictions, and available ingredients. By analyzing the ingredients that the user has on hand, the tool can suggest recipes and plan meals for the entire week. This feature will not only save time but also help users make healthier and informed choices about their food intake.

2.3.2 Registered Users

- Registered users shall be able to contact other registered users through private direct messages. This feature allows users to communicate with one another outside of reviews, giving them the opportunity to ask for

advice, recipe suggestions, and building connections with other registered users.

- Registered users shall be able to choose whether or not they want notifications or not. Notifications can help users keep up to date with the latest posts from users they have followed, as well as being informed on when someone posts a review on one of their posts. Though some users might opt out of having notifications on, those who use them will be more inclined to check for new recipes, as well as post their own creations to the website.

2.3.3 Admin

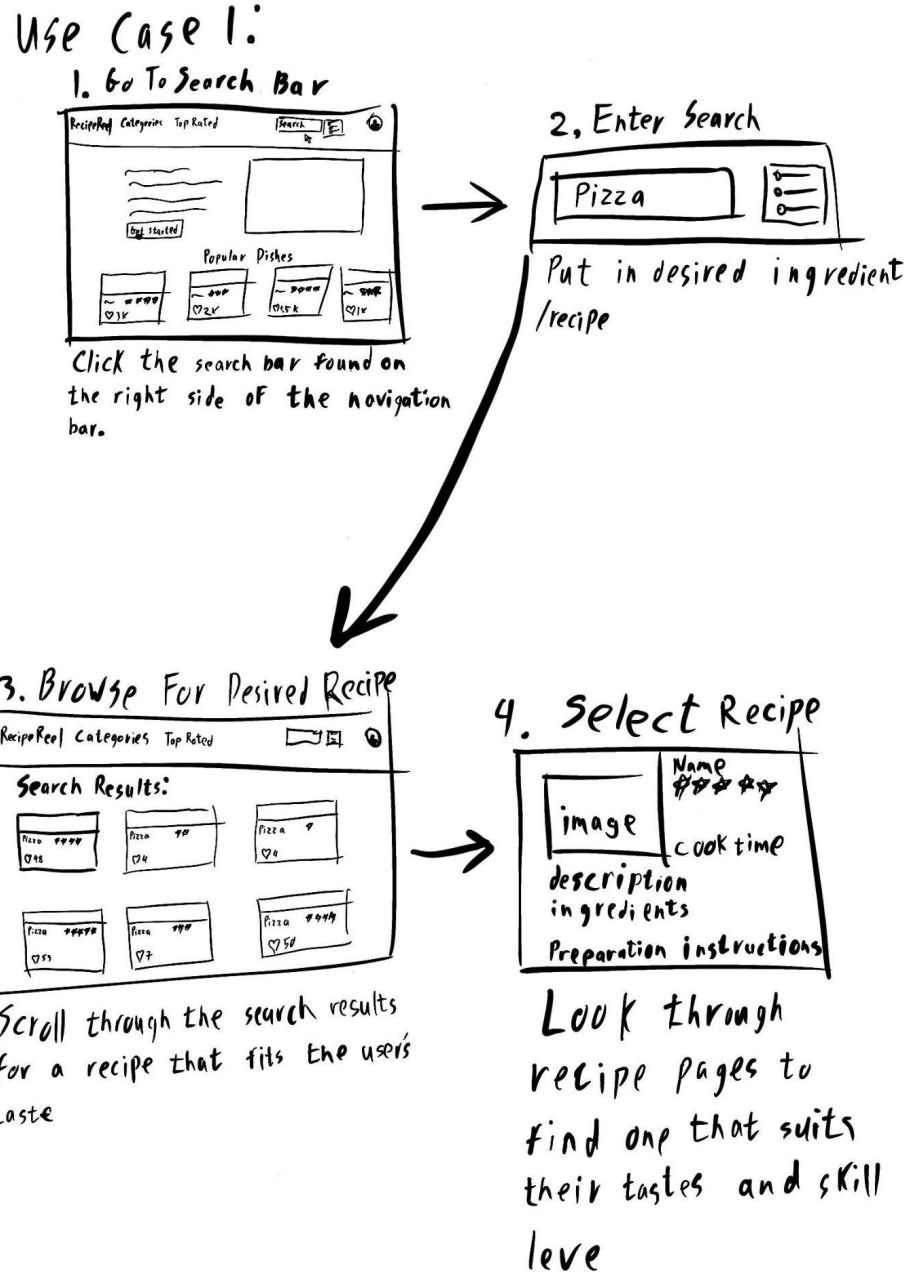
- An admin shall have monitoring tools that can track system performance metrics such as servers, applications, memory usage, and network traffic and so on. That enables a real-time view with detailed information on system performance and usage. This can help the admin to identify and address issues as they arise, ensuring that the website is running smoothly and providing a positive user experience. Examples of such tools are Nagios and New Relic.

- An admin shall have a revenue management tool. Revenue management tools can provide insights into customer behavior and preferences. This enables the admin to effectively manage and track the RecipeReel revenue streams. By having access to detailed revenue data, the admin can make informed decisions regarding the RecipeReel financial strategy and identify potential revenue opportunities. For example, if a large number of users are searching for a specific type of food, ingredient, recipe, the admin can use this information and use it for better production. Examples of revenue management are Salesforce Revenue Cloud and Google Ad Manager.

- The admin shall be able to integrate external APIs such as recipe databases or nutrition calculators, to analyze user needs and preferences in order to deliver a better satisfaction for users. For example, an access for recipe database can provide the admin with a large number of recipes, each with a list of ingredients and nutritional information. By having the insight of user preferences and dietary restrictions, the admin can recommend specific recipes that are more likely to be enjoyed by each user which is also the same with a nutrition calculator that can be used to analyze the nutrient content of each recipe and provide users with personalized nutritional recommendations. There are several nutrition calculator APIs available that can provide detailed information about the nutritional content of

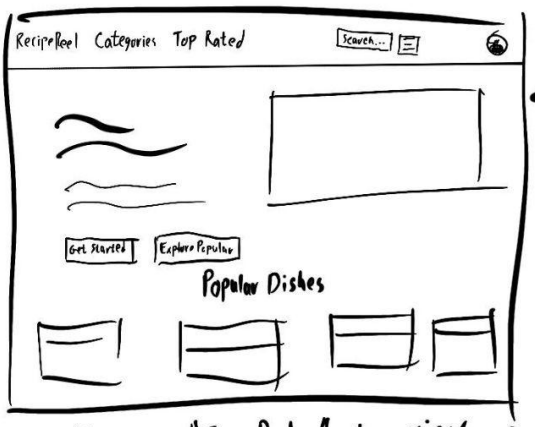
foods and ingredients. (Nutritionix API, FatSecret API, and USDA FoodData Central API.)

3. UI Mockups and Storyboards



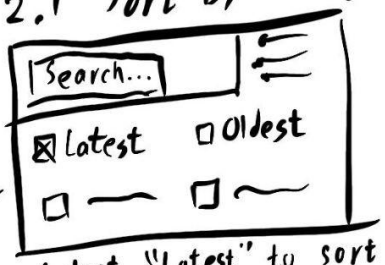
Use Case 2:

1. Go to "Top Rated" or filter



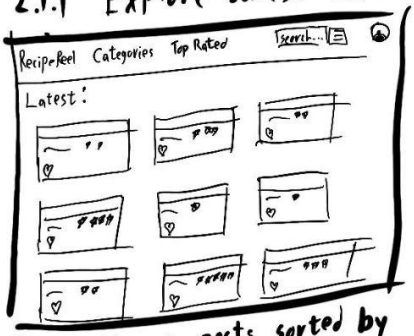
Click on "Top Rate" to view recipes with the best ratings. Click the Filter button to view other sorting options

2.1 Sort by "Latest"



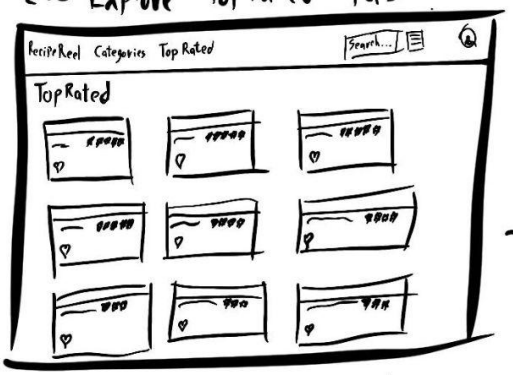
Select "Latest" to sort recipes by the most recent posts.

2.1.1 Explore Latest Posts



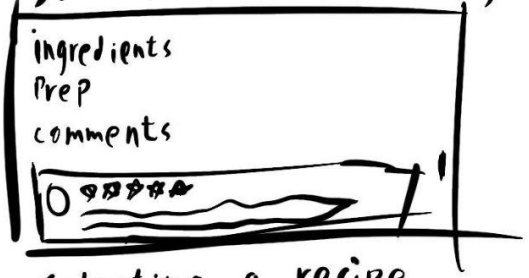
Browse through posts sorted by latest

2.2 Explore "Top Rated" Tab



Browse through top rated recipes on RecipeReel

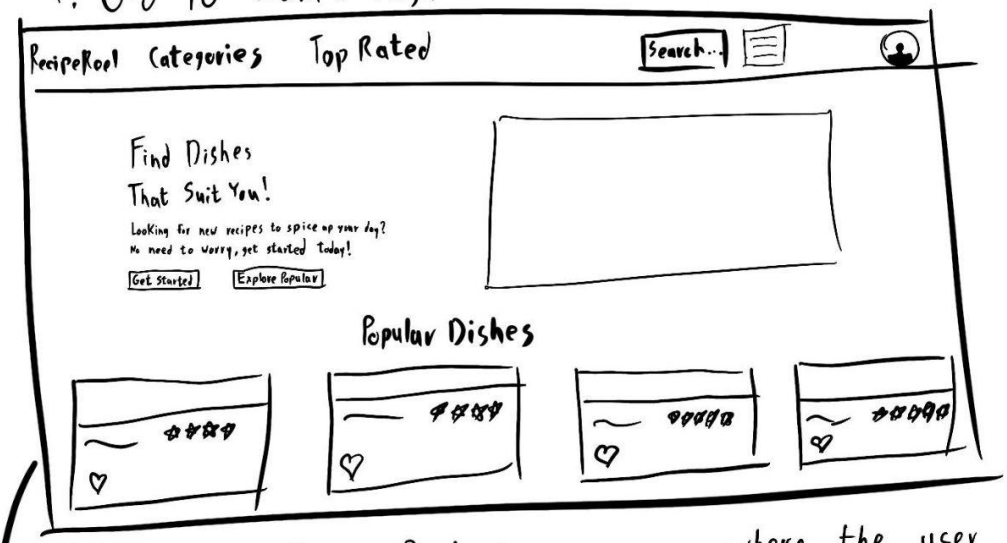
3. select and comments



Selecting a recipe and seeing what others think of it

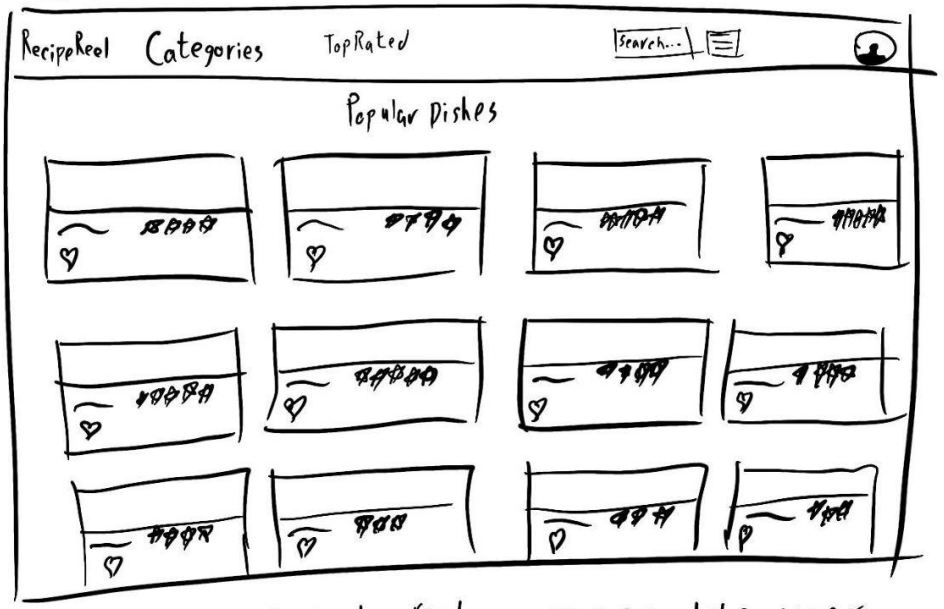
USE CASE 3:

1. Go To Home Page



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

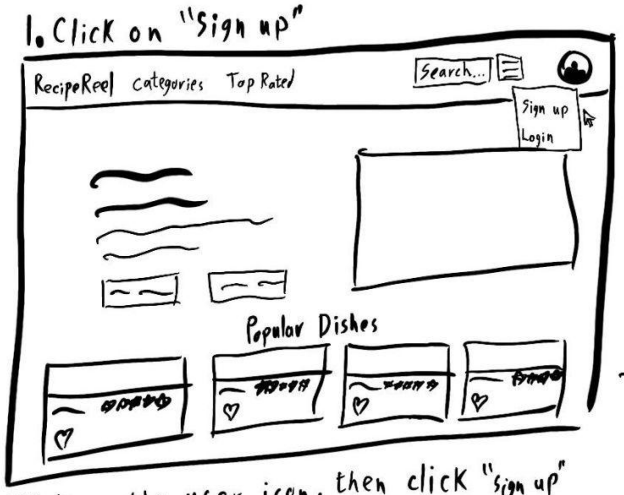
2. Explore



Browse Recipe Reel to find a recipe the user could enjoy

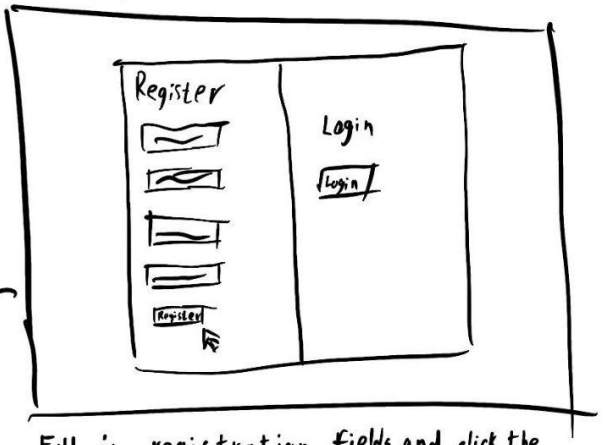
Use Case 4:

1. Click on "Sign up"



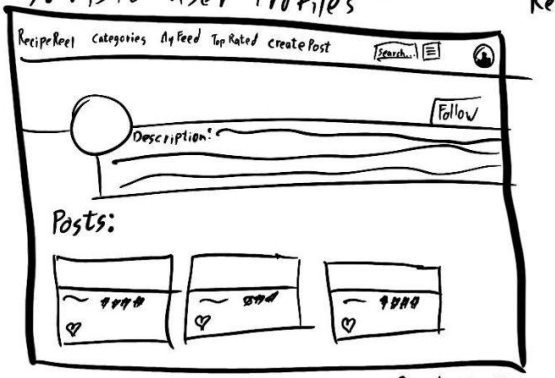
Click on the user icon, then click "Sign up" to access the registration page

2. Register



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

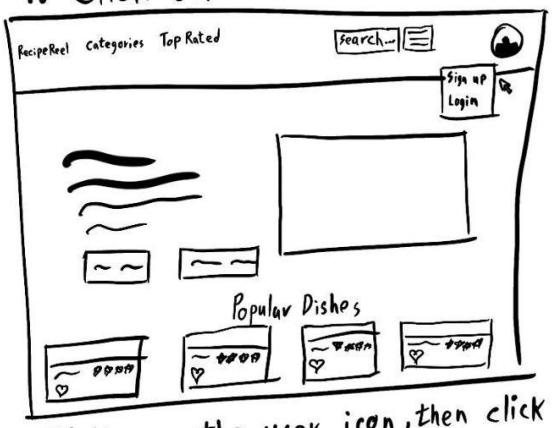
3. Visit User Profiles



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

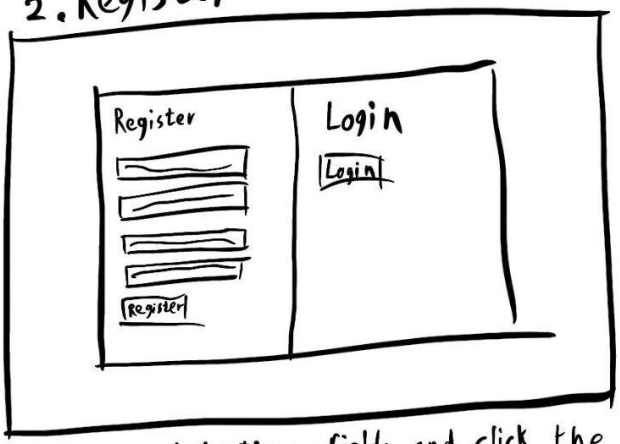
USE CASE 5:

1. Click on "Sign up"



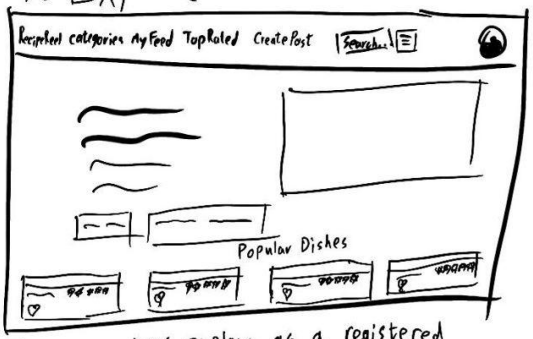
Click on the user icon, then click "Sign up" to access the registration page

2. Register



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

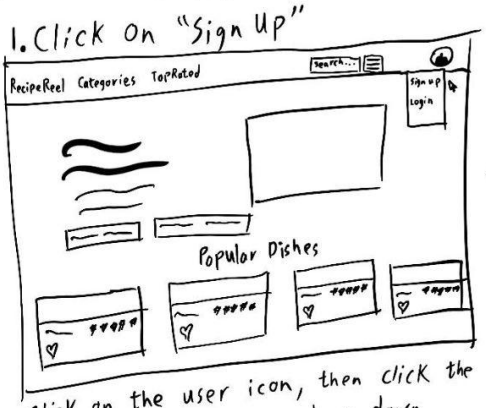
3. Explore



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

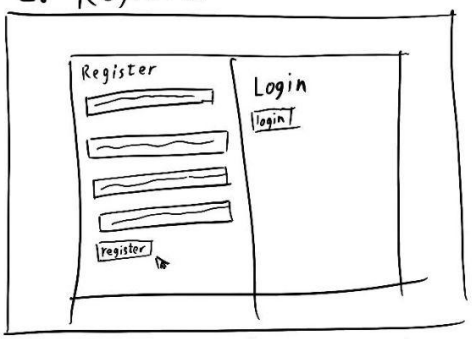
USE CASE 6:

1. Click on "Sign Up"



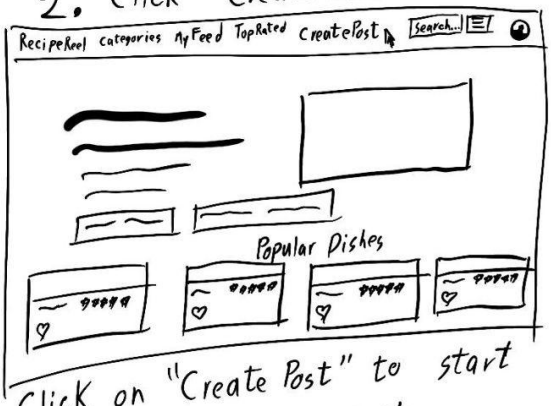
Click on the user icon, then click the "Sign Up" button on the drop down menu to access the registration page

2. Register



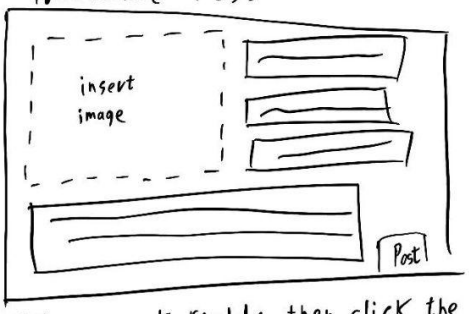
Fill in the registration fields and click the "Register" to finalize registering for a RecipeReel account

2. Click "Create Post"



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

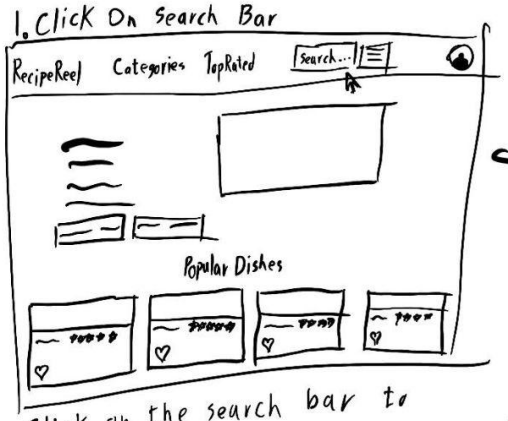
4. Create Post



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:

1. Click On Search Bar

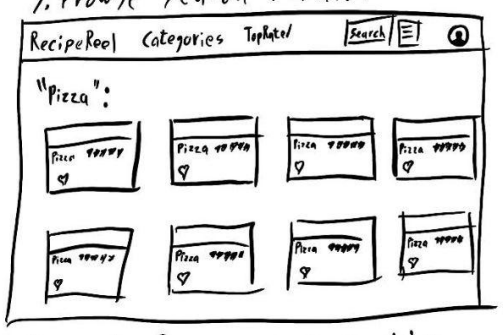


Click on the search bar to start insertin

2. Enter Recipe

Enter desired recipe into the search bar

3. Browse Search Results



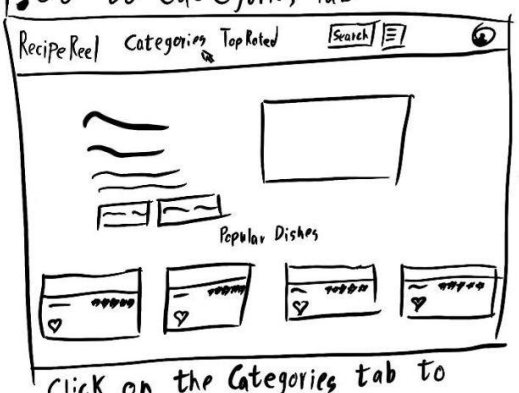
Search for a recipe the user could enjoy

4. Check Recipes

Look through recipes that fit the user's tastes and that the cook time fits in their schedule

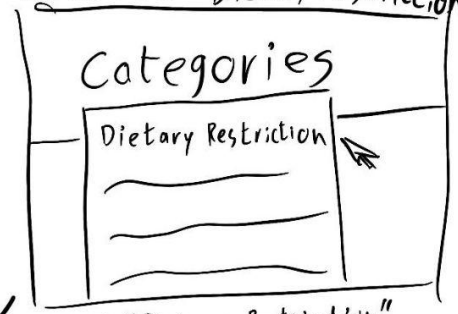
USE CASE 8:

1. Go to Categories Tab



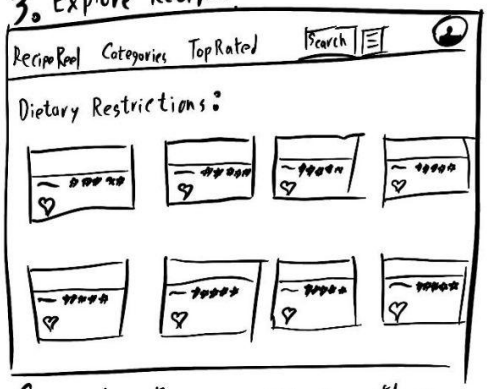
Click on the Categories tab to access the dropdown

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

1. The database shall store all information about Admins, registered users, recipes, ingredients, cuisine, occasion, and dietary restrictions.
2. The database shall allow users to create, read, update, and delete their account information, recipes, and dietary restrictions.
3. The database shall store user ratings and reviews for each recipe, as well as any associated comments or feedback.
4. The database shall allow users to search and filter recipes based on the various entities such as ingredients, cuisine, occasion, meal type, and dietary restrictions.
5. The database shall be able to handle large amounts of data effectively and efficiently.
6. The database shall ensure data reliability and consistency.
7. The database shall support multiple users accessing and modifying data simultaneously.
8. The database shall have a backup and recovery system in case of data loss or corruption.
9. The database shall secure and protect user data from unauthorized access or breach.
10. The database shall be scalable and manageable to adapt future growth and expansion.

4.2 Database Entities

- General User Entity: None

- Registered User Entity:
    - Attributes:
        - UserID: PK
        - Username
        - Email
        - Password: PK
        - Recipe
        - Comment_like
        - Comment_dislike
        - Comment
        - Rating
        - Following
        - Follower

    - Relationship:
        - Registered users shall make zero or many posts.
    - Domain:
        - UserID: integer
        - Username: alphanumeric
        - Email: alphanumeric
        - Password - alphanumeric

.

- Following Entity
    - Attribute:
        - followingID: PK
        - Title
        - Description
    - Relationship:
        - Registered users shall be able to be followed by zero or many users.
    - Domain
        - FollowingID: integer
        - Title: string
        - Description: string

- Follower Entity:
    - Attribute:
        - FollowerID: PK
        - Name: string
        - Description
    - Relationship:
        - Registered users shall be able to follow zero or many users.
    - Domain
        - FollowingID - integer
        - Name - string
        - Description - string

- Recipe Entity:
    - Attributes:
        - RecipeID:PK
        - UserID: PK
        - Title: string
        - Description: text
        - CreatedDate: timestamp
        - Instructions: text
        - Images: JPG/PNG
        - Ratings: FK
        - Categories(cuisine, occasion, DietaryRestriction,...)

        - Ingredients: string
        - Comments: text
        - CookingTime: int
        - DifficultyLevel: string
    - Relationships: This has many-to-one relationships with Users.
    - Domain:
        - RecipeID: integer
        - Title: string
        - Description: string
        - Instructions: string
        - Image: JPG/PNG
        - Ratings: Integer

- Ingredient Entity:
    - Attributes:
        - IngredientID: PK
        - Title: string
        - Description: string
    - Relationship: An ingredient shall be used by zero or many users.
    - Domain:
        - IngredientID: PK
        - Title: string
        - Description: string

- CookingTime Entity:
    - Attributes:
        - CookingTimeID
        - Time: timestamp
        - Description: Text
    - Relationship: One cooking time can be applied to many recipes.
    - Domain:
        - CookingTimeID: integer
        - Time: timestamp
        - Description: string

- Comment Entity:
    - Attribute:
        - CommentID: PK
        - Content: string
        - UserID: FK
        - PostID:FK
    - Relationship: Registered users shall be able to post zero or many comments.

    - Domain:
        - CommentID: integer
        - Date and Time: timestamp
        - Content - string

- Ratings Entity:
    - Attributes:
        - RatingID: PK
        - UserID: PD
        - RecipeID: FK
        - Rating: integer
    - Relationships: A rating shall be given by one user to one recipe.
    - Domain:
        - RatingID: PK integer
        - UserID: FK integer
        - RecipeID: FK integer

- Instruction Entity:
    - Attributes:
        - InstructionID: PK
        - Description: string
        - Title: string
    - Relationships: A recipe instruction can be used by one or many recipes.
    - Domain:
        - InstructionID: PK
        - RecipeID: FK string
        - Description: string

- Category Entity:
    - Attributes:
        - CategoryID: PK
        - Title: string
        - Description: string
    - Relationships:
        - A category can have many sub-categories.
        - A category can be associated with many recipes.
    - Domain:
        - CategoryID: integer
        - Title: string
        - Description: string

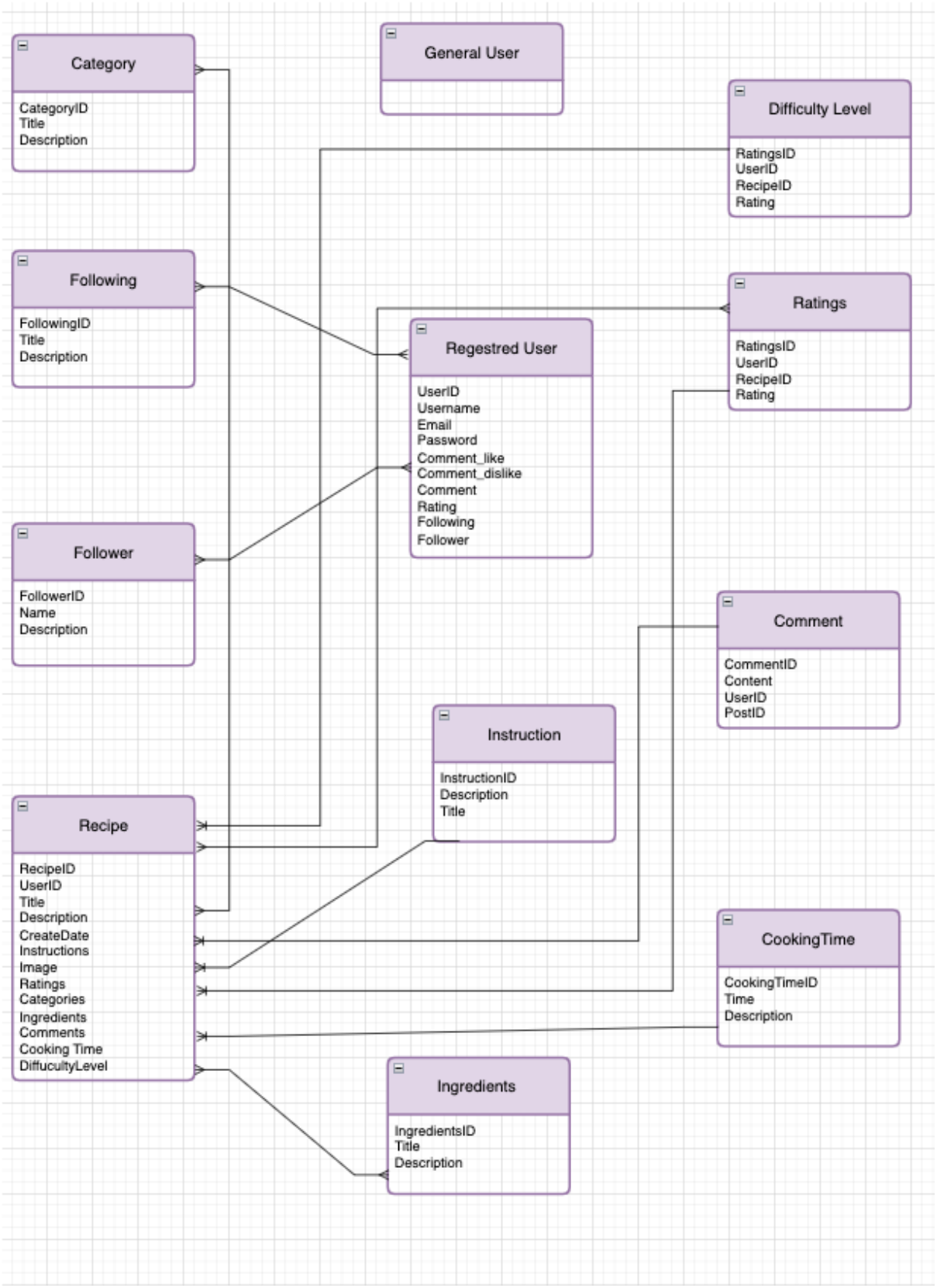
- Difficulty level

- Attributes:
  - DifficultyLevel: integer
  - Title: string
  - Description: string
- Relationship:
  - This has one to many relationship with recipe entities where a recipe can have only one difficulty level, but a difficulty level can be associated with multiple recipes.
- Domain:
  - DifficultyLevel: integer
  - Title: string
  - Description: string

Strong Entities:

1. General User
2. Registered User
3. Ratings
4. Follower Entity
5. Following Entity
6. Recipe
7. Cooking Time
8. Difficulty Level
9. Recipe Instruction
10. Comment

4.3 ERD



(Diagram 4.2.1, ERD)

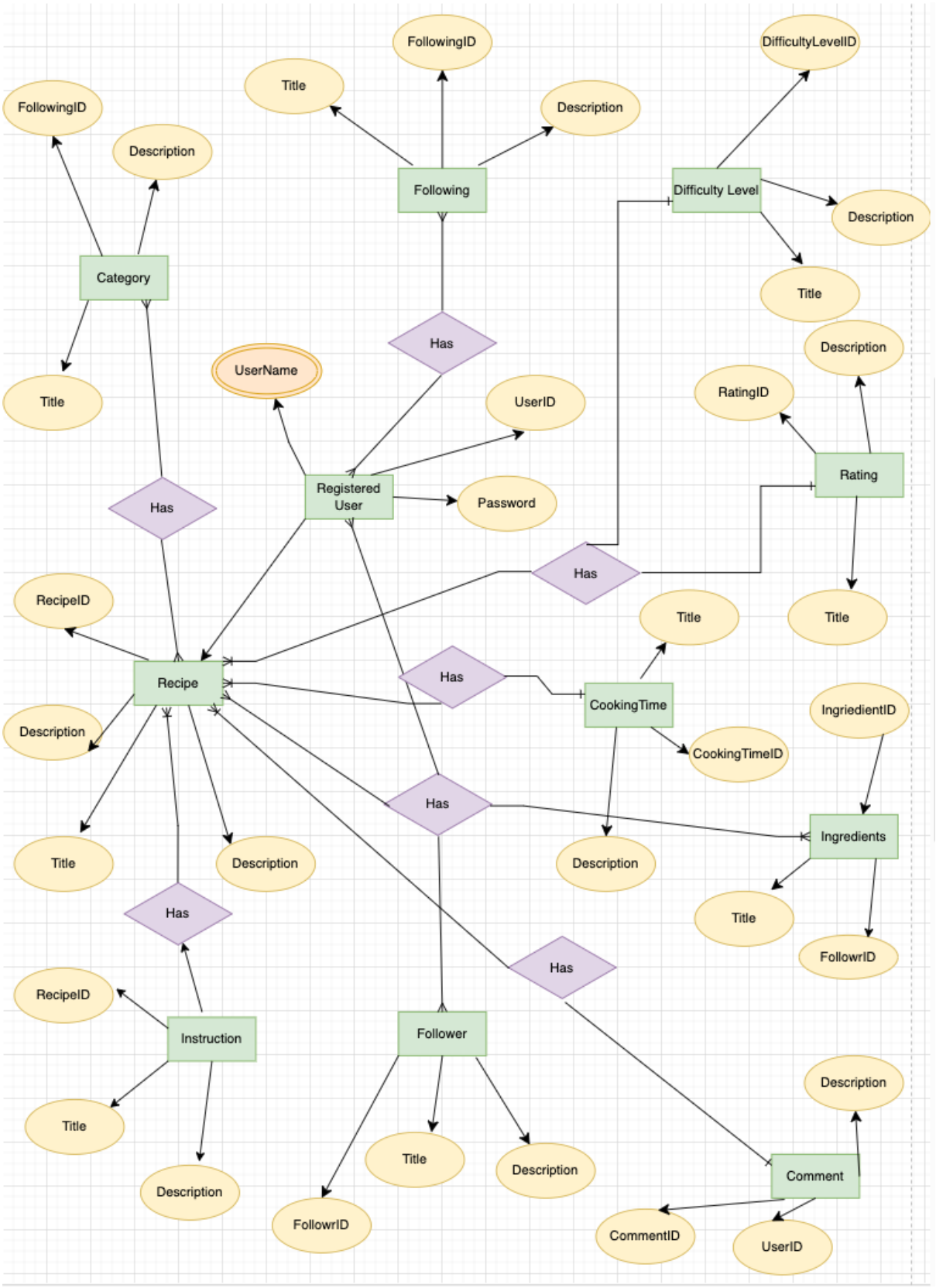


Diagram (4.2.2, ERD)

4.4 DBMS

We will use PostgreSQL for our database implementation management because it is a powerful open-source relational database system with many features that provide excellent data reliability and scalability. It also has the capability to handle large amounts of data and multiple users who are accessing it at the same time.

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.

4.6 Search/filter architecture and implementation

The search/filter architecture and implementation for the RecipeReel web app 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

High level APIs:

- Register:
    - The register body should contain the user's first name, last name, email, username, and hashed password. This is important for the system because it ensures that every user has a unique identity and secure password.

- Login:
    - Login body should contain the user's username or email, and the hashed password. This is important for the system because it allows for the user to have two forms of logging in, as well as a secure way of checking the password.

- Search:
    - The search body should contain filters, as well as post bodies that meet the search requirements and filters. This is important for the system because it displays a list of acceptable results based on the search query and filters.

- Post:
    - Post body should contain recipe instructions, username of the user who posted the recipe, ratings, and images. This is important for the system because it provides a readable format for users to read and comprehend the recipes they select.

- Rate:
    - Rate body should contain the number of likes a post/review has. This is important to the system because it helps users make informed decisions about whether or not they want to give recipes a try, as well as providing a way parameter for users to use in order to search for well liked recipes.

- Review:
    - Review body should contain text and the username of the user who posted the review. In addition, the time of the review was created. This is important to the system because it provides a readable format for the users of the comments posted

- Profile:
    - Profile body should contain the user's username and their posts. In addition to their follower and following count. This is important tot the

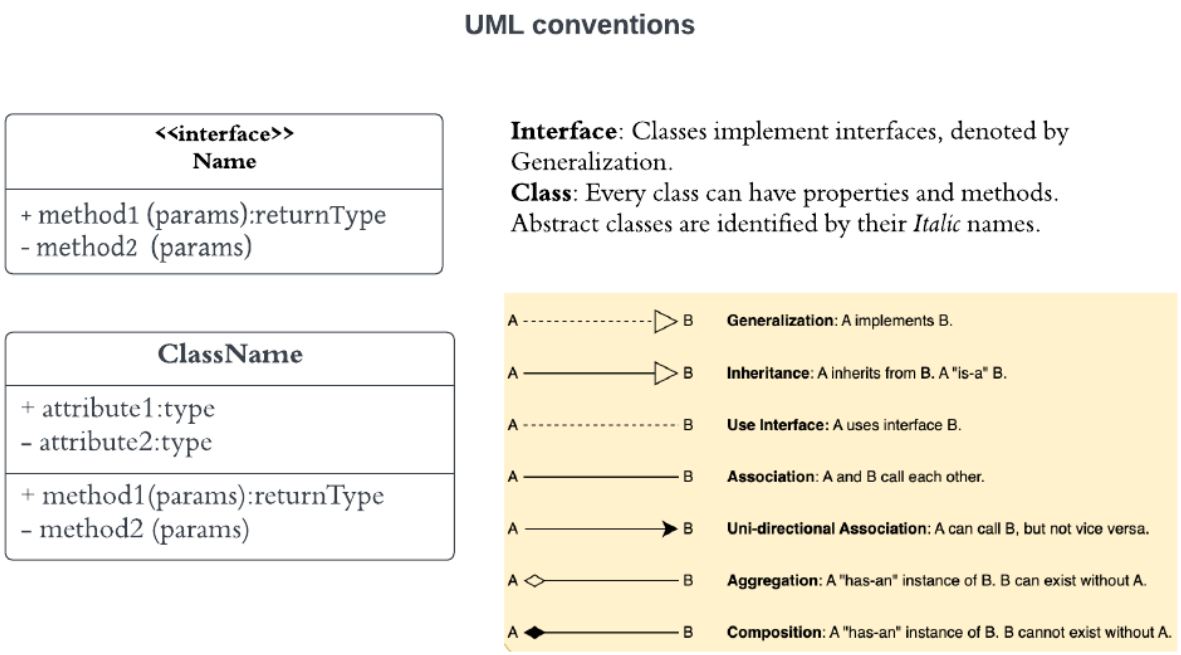
system because it allows for users to access and see everything they have done on the site.

- Feed:
    - Feed body for a registered user should include the users they follow and for users that are not registered, their feed should consist of post by random users organized from newest to oldest.

Non-trivial algorithm:

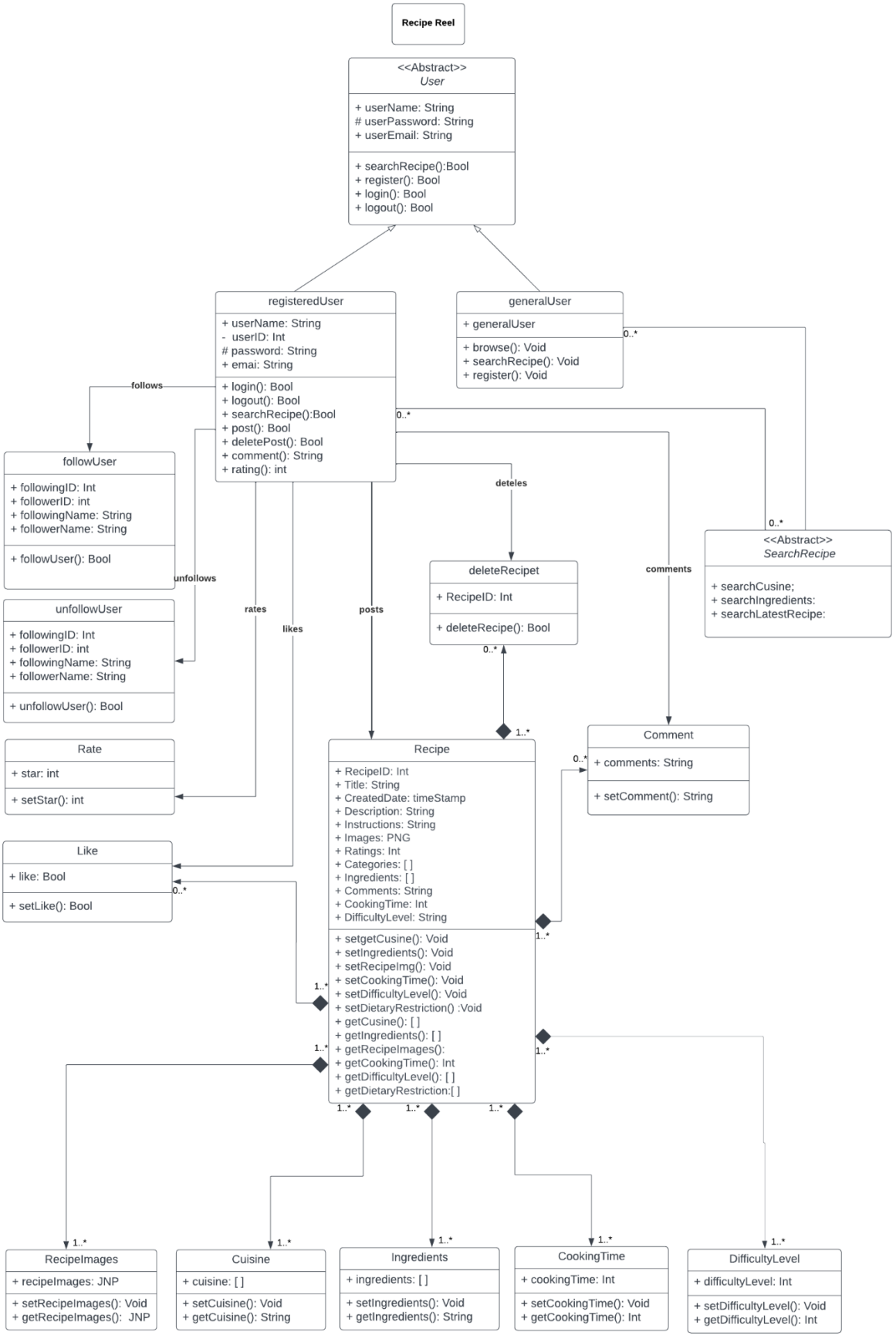
- Rating:
    - The algorithm will work by the user rating the recipe with dynamically changing UI stars. From there we will take the sum of all ratings and divide it by the total users who have rated, this will get the average rating.

6. High Level UML Diagrams

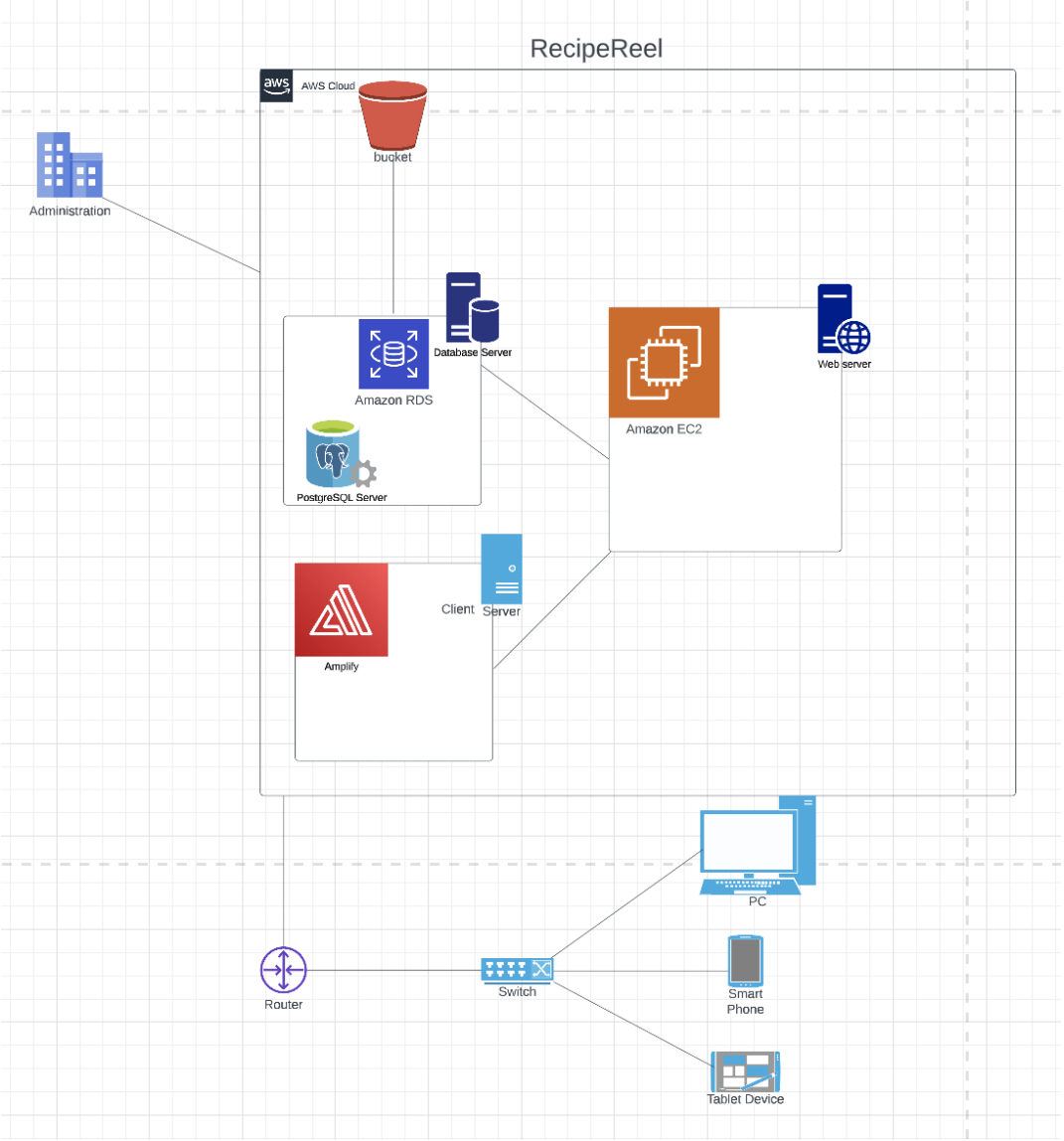


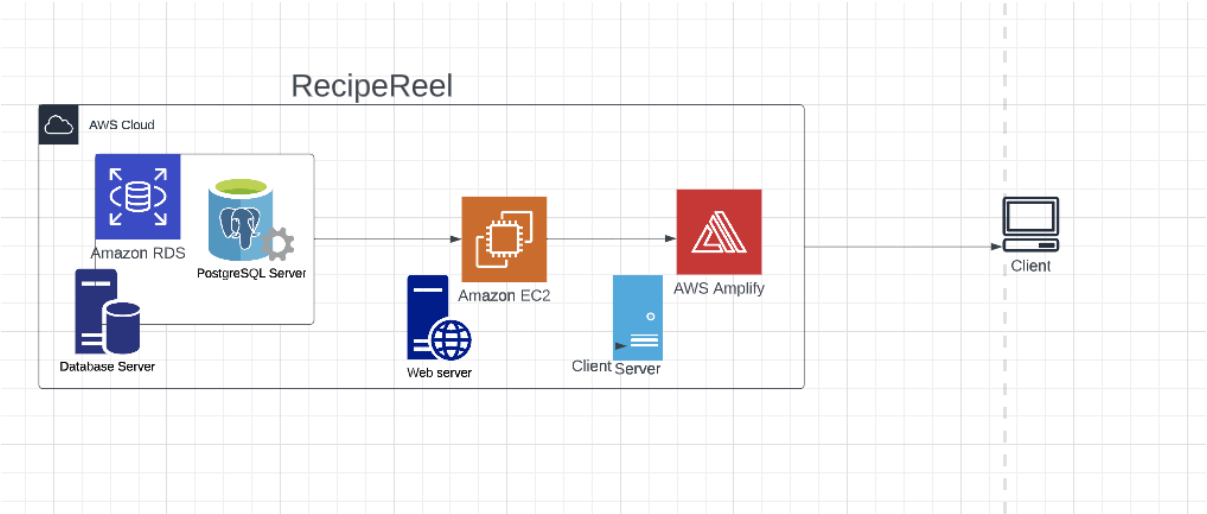
10 Main entities in RecipeReel Application UML diagram

1. General user
2. Register user
3. Post - delete Post depends on Post
4. FollowUser - unfollow User depends on Follow
5. Recipe – contains ingredients, cuisine, cooking time and difficulty level
6. Ingredient Entity - depends on Recipe, cannot exist without recipe
7. Cuisine Entity - depends on Recipe, cannot exist without recipe
8. Cooking Time - depends on Recipe, cannot exist without recipe
9. DifficultyLevel - depends on Recipe, cannot exist without recipe
10. Comment- depends on a post, cannot exist without a post and register user


Recipe Reel
<<Abstract>>
User
+ userName: String
# userPassword: String
+ userEmail: String
+ searchRecipe():Bool
+ register(): Bool
+ login(): Bool
+ logout(): Bool
registeredUser
+ userName: String
- userID: Int
password: String
+ emai: String
+ login(): Bool
+ logout(): Bool
+ searchRecipe():Bool
+ post(): Bool
+ deletePost(): Bool
+ comment(): String
+ rating(): int
generalUser
+ generalUser
+ browse(): Void
+ searchRecipe(): Void
+ register(): Void
follows
0..*
followUser
+ followingID: Int
+ followerID: int
+ followingName: String
+ followerName: String
+ followUser(): Bool
deteles
0..*
<<Abstract>>
SearchRecipe
+ searchCusine;
+ searchIngredients:
+ searchLatestRecipe:
unfollows
unfollowUser
+ followingID: Int
+ followerID: int
+ followingName: String
+ followerName: String
+ unfollowUser(): Bool
rates
likes
posts
deleteRecipet
+ RecipeID: Int
+ deleteRecipe(): Bool
comments
0..*
1..*
Rate
+ star: int
+ setStar(): int
Recipe
+ RecipeID: Int
+ Title: String
+ CreatedDate: timeStamp
+ Description: String
+ Instructions: String
+ Images: PNG
+ Ratings: Int
+ Categories: [ ]
+ Ingredients: [ ]
+ Comments: String
+ CookingTime: Int
+ DifficultyLevel: String
+ setgetCusine(): Void
+ setIngredients(): Void
+ setRecipeImg(): Void
+ setCookingTime(): Void
+ setDifficultyLevel(): Void
+ setDietaryRestriction() :Void
+ getCusine(): [ ]
+ getIngredients(): [ ]
+ getRecipeImages():
+ getCookingTime(): Int
+ getDifficultyLevel(): [ ]
+ getDietaryRestriction:[ ]
Comment
+ comments: String
+ setComment(): String
Like
+ like: Bool
+ setLike(): Bool
RecipeImages
+ recipeImages: JNP
+ setRecipeImages(): Void
+ getRecipeImages(): JNP
Cuisine
+ cuisine: [ ]
+ setCuisine(): Void
+ getCuisine(): String
Ingredients
+ ingredients: [ ]
+ setIngredients(): Void
+ getIngredients(): String
CookingTime
+ cookingTime: Int
+ setCookingTime(): Void
+ getCookingTime(): Int
DifficultyLevel
+ difficultyLevel: Int
+ setDifficultyLevel(): Void
+ getDifficultyLevel(): Int

7. High Level Application Network and Deployment Diagrams





8. Identify Actual Key Risks for Your Project at this Time

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 useable 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 ● Worked on	
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	8/10
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 ● Assisted with answering questions for team members	8/10

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	5/10
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 with answering questions for team members	8/10
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 for our Project at this Time	6/10
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	4/10