

COMP9900 Project Proposal

Meal Recommendation System



Team: Brainstorm

Lab: M18Q

z5228748 Hanyue Jiang z5228748@ad.unsw.edu.au Back-end

z5319735 Shiyu Nie z5319735@ad.unsw.edu.au Front-end

z5295651 Zening Wang z5295651@ad.unsw.edu.au Front-end

z5298115 Han Yan z5298115@ad.unsw.edu.au Front-end

z5290496 Xueqing Ren z5290496@ad.unsw.edu.au Back-end

Submission date:

03 Oct, 2022

Contents

1 Background.....	1
1.1 Problem Statement.....	1
1.2 Existing systems analysis.....	2
2 User Stories and Sprints.....	4
2.1 Epics and user stories.....	4
2.1.1 Epic 1: User Registration & Validation & Profile	4
2.1.2 Epic 2: Recipes Maintaining.....	4
2.1.3 Epic 3: Like & Comment on Recipes	5
2.1.4 Epic 4: Searching Module.....	5
2.1.5 Epic 5: Subscribe Module.....	6
2.1.6 Epic 6: Recipes Ranking.....	6
2.1.7 Epic 7: Recommendation System	6
2.1.8 Epic 8: Favorites Module.....	6
2.2 Novelty.....	7
2.3 Product Backlog.....	8
2.4 Sprints	8
2.4.1 First Sprint	8
2.4.2 Screenshot for the First Sprint in Jira	9
2.4.3 Coming Sprints	9
3 Interface and Flow Diagram	11
3.1 Interface	11
3.1.1 Login and Register Page	11
3.1.2 Profile Page.....	11
3.1.3 Home Page.....	12
3.1.4 Filter Search Page	13
3.1.5 Create Recipes Page.....	13
3.1.6 Own Recipes Page	14
3.1.7 Subscriber Management and Post Page	14
3.1.8 Favorites Page.....	15
3.1.9 Ranking List Page	16
3.2 Flow Diagram	17
4 System Architecture	18
4.1 Architecture breakdown.....	18
4.1.1 Separate Back-end and Front-end System	18
4.1.2 Presentation layer.....	18
4.1.3 Business layer	18
4.1.4 Data layer	19
4.2 External actors	19
4.2.1 Users	19
4.2.2 Explorers.....	19
4.2.3 Contributors	19

4.3 Technologies descriptions	19
4.3.1 Presentation layer.....	19
4.3.2 Business layer	20
4.3.3 Data layer.....	20
5 Reference	21

1 Background

Profited from the rapid development and convenience of the Internet, people who like to cook around the world do not have to stick to the form of word of mouth or reading paper books to obtain new recipes. By sharing their recipes online and filtering the ones they are most interested in easily, people can improve and innovate food through mutual communication, which greatly promotes the integration and progress of food cultures in different regions. In addition, due to the difference in personal taste, establishing a way that can obtain recipes based on personal preferences in a short time can also bring more convenience to people's lives.

1.1 Problem Statement

This project is committed to building an online meal recommendation system to help users get their favorite recipes and communicate easily with other cooking enthusiasts. The user group of this platform can be divided into two main types: Explorers and contributors.

First of all, for all the platform users, they need a personal account to store and display their personal information including name, email, photo and gender. These are presented on a separate page and can be changed by the user. To meet this requirement, a reasonable registration process which allow users to set and confirm the password twice and the corresponding database is needed. We hope to reduce the mis operation of users and make the registration process more efficient.

As an explorer, what one user need is not only to obtain the recipes he/she is interested in through the search function, but also to establish a communication relationship with contributors. Including subscribing to the contributors to obtain the latest release, liking, commenting and collecting the recipes, and corresponding reverse operations. By improving the details of search and subscription functions, like the navigation of thumbnails, we aim to make the process of obtaining the desired information as efficient as possible. In addition, a personal recipe news feed is designed to display the latest and most popular recipes of the platform and help users get the information they want intuitively.

As a contributor, the user also needs to store the recipes they created. Simple and complete pages are designed to help them display recipes and communicate with other users. In order to make it easier for users to manage their published contents, the updating, deleting and locking functions are also included. Create and modify recipes have independent pages and fixed templates. Users can easily operate these functions without considering the format. The locking function can help users hide their recipes from other users, making them only visible to themselves.

Furthermore, a recommendation system based on the relevance of recipes' ingredients

is designed and used to provide users with corresponding similar ones when reading one recipe. To make the recommendations given by the system more suitable for users' preferences, a reasonable metric for the similarity of the ingredients of recipes, and intuitively push the thumbnail links of relevant recipes when each recipe is read by users.

1.2 Existing systems analysis

We choose two existing platform *kitchenstories* (<https://www.kitchenstories.com/en/recipes>) and *allrecipes* (<https://www.allrecipes.com>) as reference. The functions of each website is shown in detail in the tables below.

Functions/Website	kitchenstories	allrecipes
Personal Account	Do not need complex password when registering. (At least eight arbitrary characters.)	Need complex password when registering. (At least eight characters including lowercase letters, uppercase letters and numbers.)
	Display detailed basic personal information (including name, email, birthday and some customized contents).	
	Display personal recipes.	Display personal recipes, collections and reviews.
Search	Search by keywords.	
	Filter by dish type, diet, preparation time, ingredients(only include) and cuisine.	Filter by ingredients(both include or exclude).
	Results can be sorted.	Results can not be sorted.
	Thumbnail shows the name, uploaded time and number of likes of a recipe.	Thumbnail shows the name, brief introduction and rate of a recipe.
Detailed Recipe	Like, comment and collect a recipe.	
	Comments can be liked and replied.	Comments can be liked and sorted.
	Pictures and words for each cooking step.	Words for each cooking step.
	Feature: Links to How-to videos and tags.	Feature: Shopping list and local offers for ingredients.
Subscription	Not able to subscribe another user.	
Recommendation	Recommendation logic not explained.	Recommend by dish type and main ingredient.
	The recommendations can not be sorted.	

From the table, it can be seen that the two websites have some small innovations to improve the user experience, and each has its own emphasis on basic functions. In addition, we also find that neither of the two websites provide the function of subscription between users, which is an innovation for our project and can help users more easily obtain the recipes they may be interested in. We also hope to bring better user experience by improving our recommendation function.

2 User Stories and Sprints

2.1 Epics and user stories

2.1.1 Epic 1: User Registration & Validation & Profile

User Stories	<ul style="list-style-type: none">• As a web user, I want to have a private account so that I can save my own data.• As a web user, I want to have a strong password so that my account have security.• As a web user, I want to create a profile with personal information so that other web users can contact me.• As a web user, I want to upload a picture as my profile picture so that I can show visual information.• As a web user, I want to be able to reset password so that I can ensure my account's security.• As a web user, I want to validate my created account so that I can check if account name is legal.• As a web user, I want to see my followers count so that I can know how popular I am.
Acceptance Criteria	<ul style="list-style-type: none">• A user can create an account with personal information.• A warning will be displayed if: The password is not strong enough. The account name is illegal.• A user can login with the created account and change his/her account's information.

2.1.2 Epic 2: Recipes Maintaining

User Stories	<ul style="list-style-type: none">• As a contributor, I want to have a template for uploading a recipe so that upload operation can be simplified.• As a contributor, I want to create new recipes so that I can share them with other users.• As a contributor, I want to have a personal page listing all my recipes so that users can find them easily.• As a contributor, I want my recipes have their own page showing details so that users can see them clearly.• As a contributor, I want to update my recipes so that I can remove some mistakes.• As a contributor, I want to remove my recipes so that I can manage my recipes list.• As a contributor, I want my recipes can be classified by different features so that users can find them easily.
--------------	--

Acceptance Criteria	<ul style="list-style-type: none">• A user can manage his/her own recipes (including creating/deleting/modifying).• There should be a page for: All the recipes of one user. Each of these recipes.• There should be a fixed template for uploading recipes.• The recipes can be filtered by feature.
---------------------	--

2.1.3 Epic 3: Like & Comment on Recipes

User Stories	<ul style="list-style-type: none">• As an explorer, I want to comment on recipes, so that I can convey my opinion.• As an explorer, I want to delete my comments so that I can remove my improper thoughts.• As an explorer, I want to be able to click the like button, so that I can add it to my favorites.• As an explorer I want to click the like button again, so that I can remove it from my favorites.
Acceptance Criteria	<ul style="list-style-type: none">• A user can interact with recipes published by other users (including liking and commenting).• A user can cancel the former operations.

2.1.4 Epic 4: Searching Module

User Stories	<ul style="list-style-type: none">• As an explorer, I want to be able to search for recipes so that I can quickly find recipes I interested.• As an explorer, I want to search by selecting one or more of the search criteria such as ingredient, method, meal type, and recipe name, so that I can find relevant types of recipes quickly.• As an explorer, I want to get a recipe list with recipe name and photo thumbnail after using the search module, so that I can get the information more clearly.• As an explorer, I want to click on the thumbnail, so that I can get the details of the recipe.
Acceptance Criteria	<ul style="list-style-type: none">• A user can obtain recipes of interest through a variety of search criteria.• For a search result there should be recipes with: Name Photo Thumbnail (including details)

2.1.5 Epic 5: Subscribe Module

User Stories	<ul style="list-style-type: none">• As an explorer, I want to subscribe/unsubscribe users so that I can keep track on users I like.• As an explorer, I want to be informed when my subscriber published a new recipe so that I can keep track on them quickly.• As an explorer, I want to get my personal recipe news so that I can have a brief understanding of latest news.
Acceptance Criteria	<ul style="list-style-type: none">• A user can subscribe other users.• The updated information of one's followers will be displayed in the form of: Prompt icon Homepage push

2.1.6 Epic 6: Recipes Ranking

User Stories	<ul style="list-style-type: none">• As an explorer, I want to get a recipes' ranking list so that I can know the most popular/latest recipes.• As an explorer, I want to filter part of the ranking list so that I can get the recipes uploaded in a time period.
Acceptance Criteria	<ul style="list-style-type: none">• A user can get a reasonably designed recipes ranking list.• The recipe ranking list should be: Able to sort the recipes by the release time and popularity. Able to filter release time period.

2.1.7 Epic 7: Recommendation System

User Stories	<ul style="list-style-type: none">• As an explorer, I want to get a recommendation when I reading a recipe so that I can easily get similar recipes.• As an explorer, I want to get a top-5 recommendation so that I can have more selections.
Acceptance Criteria	<ul style="list-style-type: none">• There should be a reasonable metric to measure the similarity between recipes.• The recommended results should be sorted in descending order of similarity.

2.1.8 Epic 8: Favorites Module

User Stories	<ul style="list-style-type: none">• As an explorer, I want a favorite, so that I can review my likes many times.• As an explorer, I want to delete recipes in my favorites so that I can manage my favorites better.
--------------	---

Acceptance Criteria	<ul style="list-style-type: none">• A user can get a favorite folder to store his/her favorite recipes.• A user can manage recipes in their favorite folder (including adding and deleting).
---------------------	---

2.2 Novelty

In the two websites we selected for reference (*kitchenstories* and *allrecipes*), neither of them provides the subscription function between users, which limits users' attention to simple recipes. This means that if a user finds that he/she has similar tastes with another user, he/she can only follow that user through the browser's bookmark function, and can not get the information about the recipe update in time, which undoubtedly brings a lot of inconvenience. Our design helps to strengthen the connection between users by adding the subscription function which is commonly used in social platforms. Users are able to get the latest published content of followers in a more timely manner, and the content they pay attention to can more easily adapt to personal preferences.

We also noticed that users need to click to enter a separate page each time when viewing their personal information in the two websites we compared. To simplify this process, we try to display the number of a user's followers, fans and recipes in the form of hover box, so that users can more easily and intuitively obtain their social information.

Besides, the display of recipes on the two reference websites is not sequenced, which makes it difficult for users to find what they like most when facing a large number of recipes. That is difficult to satisfy users who are pursuing efficiency. Thus in our website a ranking list is designed to reflect the update time and popularity of recipes in the entire website. Users can choose the ranking standard of the list according to their own needs. We also plan to further optimize the ranking list so that users can accurately find the recipes updated on the website within a certain period of time. Our design hopes to provide better user experience for different users by more accurately fitting their personal preferences.

2.3 Product Backlog

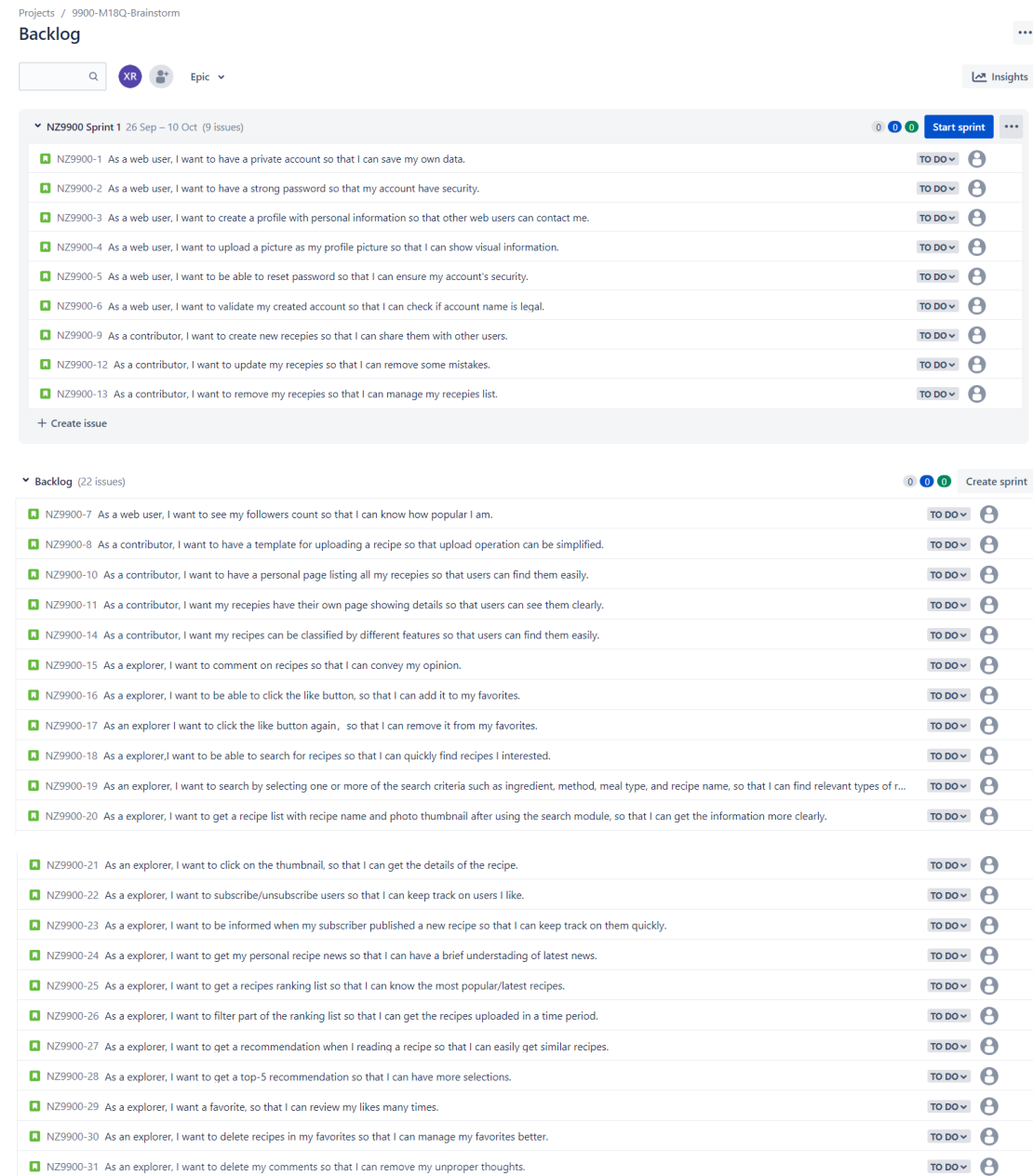


Figure 2.1 Product Backlog

2.4 Sprints

2.4.1 First Sprint

Sprint	Dates	Project Objectives	Jira Backlog ID
1	Start: Monday, Week 3, Sep 26 End:	The basic page layout design should be completed. The user registration function, the functions related to the user account and the login interface should be	NZ9900-1 NZ9900-2 NZ9900-3 NZ9900-4 NZ9900-5

	Monday, Week 5, Oct 10	completed. Implement some recipe based user management functions. Progress Demo A is scheduled on Monday in Week 5.	NZ9900-6 NZ9900-9 NZ9900-12 NZ9900-13
--	---------------------------	---	--

2.4.2 Screenshot for the First Sprint in Jira

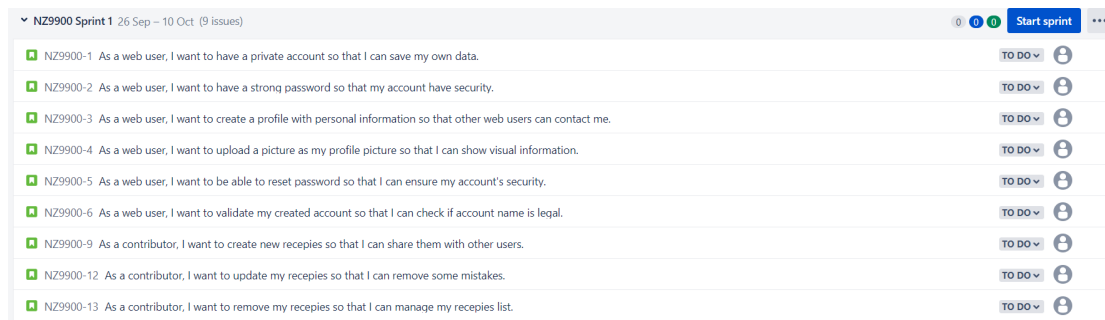


Figure 2.2 First Sprint

2.4.3 Coming Sprints

Sprint	Dates	Project Objectives	Jira Backlog ID
2	Start: Tuesday, Week 5, Oct 11 End: Monday, Week 8, Nov 1	The display and classification of recipes and all relevant pages should be completed. Users' interaction functions with other users' recipes should be completed. The search function of recipes in the platform and the related interface design should be completed. Retrospective A is on Monday in Week 7. Progress Demo B is scheduled on Monday in Week 8.	NZ9900-8 NZ9900-10 NZ9900-11 NZ9900-14 NZ9900-15 NZ9900-16 NZ9900-17 NZ9900-18 NZ9900-19 NZ9900-20 NZ9900-21 NZ9900-31
3	Start: Tuesday, Week 8, Nov 2 End: Monday, Week 10, Nov 14	The user's subscription function to other users should be implemented. The functions and interface design related to the recipe ranking list should be completed. The metric of the recommendation system based on the similarity of	NZ9900-7 NZ9900-22 NZ9900-23 NZ9900-24 NZ9900-25 NZ9900-26 NZ9900-27

		<p>recipes should be determined, and the recommendation function should be implemented.</p> <p>The user's favorite function and related interface design should be completed.</p> <p>Optimize the presentation of user information.</p> <p>After all the established functions are realized, the system should be tested.</p> <p>Retrospective B is on Monday in Week 9.</p> <p>Final Demo due on Monday in Week 10.</p>	<p>NZ9900-28</p> <p>NZ9900-29</p> <p>NZ9900-30</p>
--	--	--	--

3 Interface and Flow Diagram

3.1 Interface

3.1.1 Login and Register Page

On the login page, users can find the register and reset password entrance. When the users click the sign up button, it will jump to the registration page. And when the users clicks the reset password button, it will jump to the reset password page.



Figure 3.1 Login and Register Page

3.1.2 Profile Page

On the home page, after the user login, user avatar will be displayed on the top right of the page. When the user hovers over the avatar, a drop-down menu will pop up. In this menu, the user can see the number of fans, follow and published recipes. When the user click the Post button, it will jump to the post page. When the user clicks the Favorites button, it will jump to the favorites page. When the user clicks the Own recipes button, it will jump to the own recipes page. When the user clicks the Create Recipes button, it will jump to the own create Recipes page.

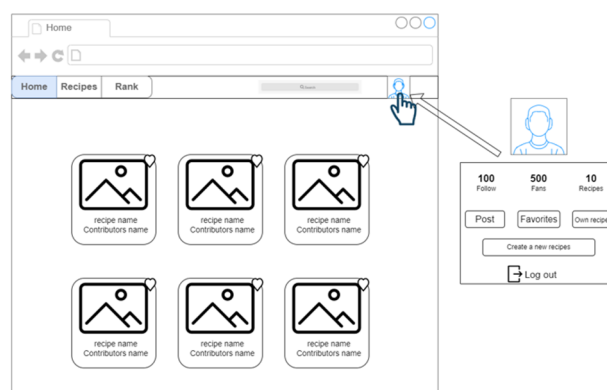
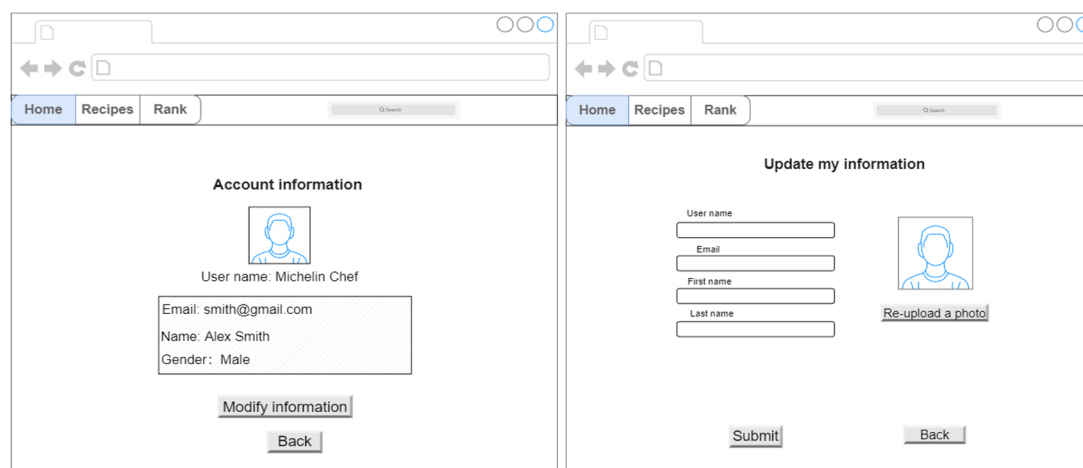


Figure 3.2 Profile Page

In addition, when the user click the avatar, it will jump to personal information page. In this page, user can see the detailed information you registered. By clicking the

Modify information button, user can modify the personal information.



(a)Account information

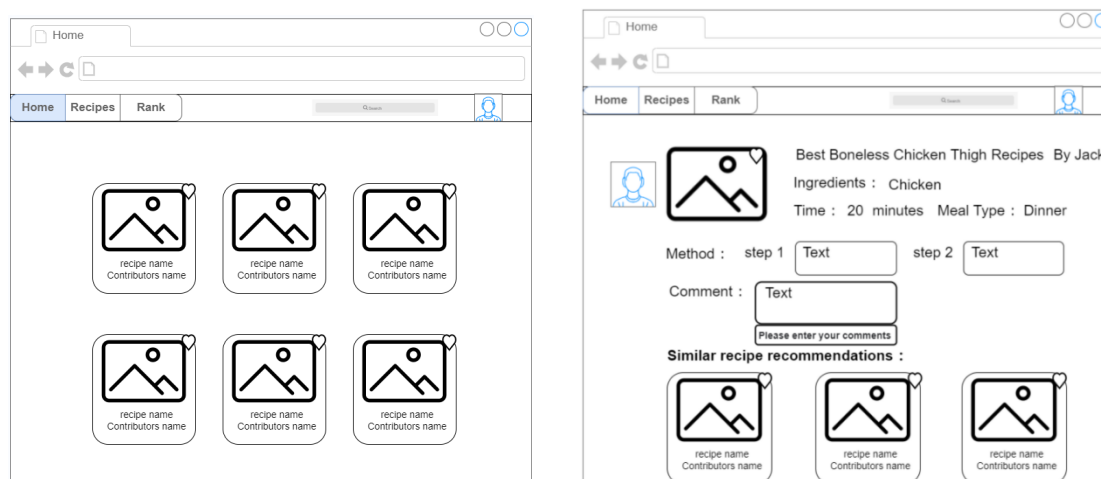
(b)Update information

Figure 3.3 Personal Information Page

3.1.3 Home Page

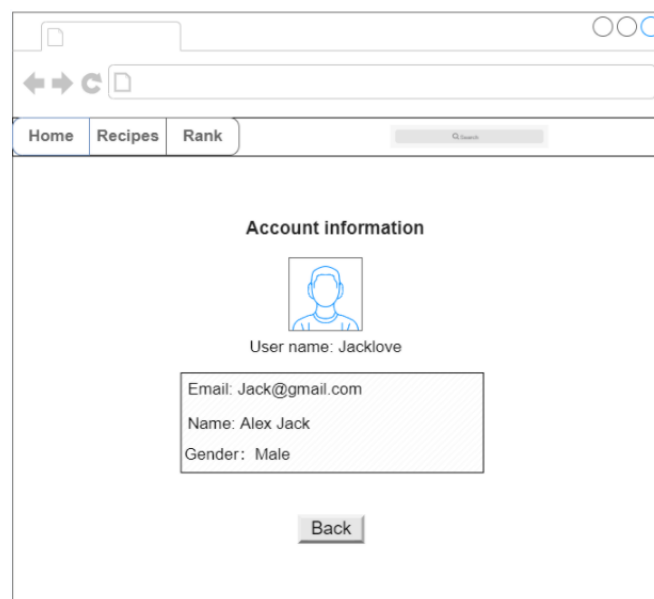
The homepage displays thumbnails of randomly recommended recipes, with the name of the recipe and the name of the contributor below the thumbnail. Users can like the recipe and add the recipe to their favorites by clicking the heart icon on the upper right of the thumbnail.

After the user clicks the thumbnail, the recipe interface is entered. In this interface the user can see the name of the recipe, the name of the contributor, the ingredients used in the recipe, time, meal type, method and comments about the recipe. At the bottom, users can see recommendations for recipes related to this recipe. Click the contributor avatar on the right to view the basic information of the contributor.



(a)Home Page

(b)Recipe Interface



(c) Personal Information Interface

Figure 3.4 Home page

3.1.4 Filter Search Page

Users can jump to this page by click Recipes button in home page or enter text search. On this page, users can filter recipes by selecting different filter conditions. In addition, recipes are also related to the input text.

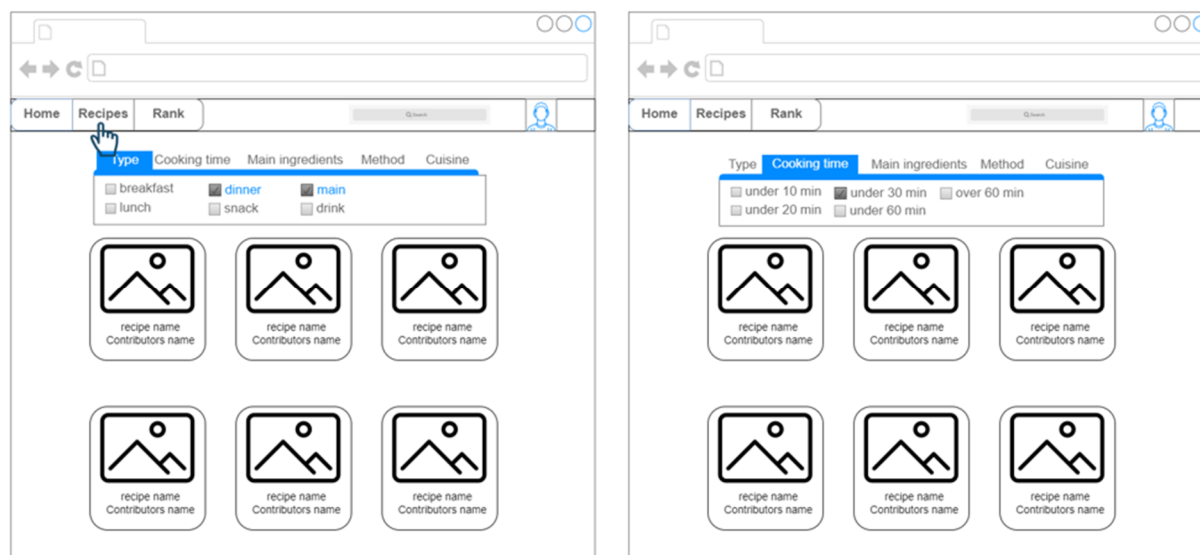


Figure 3.5 Filter Search

3.1.5 Create Recipes Page

On this page, the user can enter the name of the recipe, the type of meal, the ingredients, time, method and picture required for the recipe. Finally, click the submit button in the lower right corner to upload the recipe information.

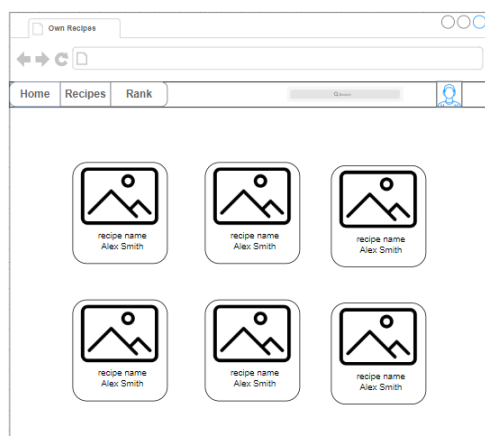
Figure 3.6 shows a web browser window titled 'Create Recipes'. The browser's address bar is empty. The page has a navigation bar with 'Home', 'Recipes', and 'Rank' tabs, and a 'Logout' button. The main content area is titled 'Please Create Your Recipe'. It contains the following form elements:

- 'Recipe Name:' followed by a text input field.
- 'Meal Type:' followed by a text input field.
- 'Ingredients:' followed by three text input fields and a '+' button to add more.
- 'Time:' followed by a text input field and the word 'minutes'.
- 'Method:' followed by 'step 1', a text input field, and a '+' button to add more steps.
- 'Photo:' followed by a '+' button to upload a photo.
- A 'Submit' button in the bottom right corner.

Figure 3.6 Create Recipes Page

3.1.6 Own Recipes Page

On this page users can see their own published recipes. Click on the recipe thumbnail to enter the modification interface, and you can modify the existing information of the recipe. Finally, click the submit button in the lower right corner to upload the revised recipe information.



(a)Own Recipes Page

Figure 3.7(b) shows a web browser window titled 'Own Recipes'. The browser's address bar is empty. The page has a navigation bar with 'Home', 'Recipes', and 'Rank' tabs, and a 'Logout' button. The main content area is titled 'Please Modify Your Recipe'. It contains the following form elements:

- 'Recipe Name:' followed by a 'Text' input field.
- 'Meal Type:' followed by a 'Text' input field.
- 'Ingredients:' followed by three 'Text' input fields and a '+' button to add more.
- 'Time:' followed by a 'Number' input field and the word 'minutes'.
- 'Method:' followed by 'step 1', a 'Text' input field, and a '+' button to add more steps.
- 'Photo:' followed by a '+' button to upload a photo.
- A 'Submit' button in the bottom right corner.

(b)Modification Interface

Figure 3.7 Own Recipes Page

3.1.7 Subscriber Management and Post Page

On the Subscriber Management page, users can manage their subscription targets. Each subscription target has two buttons. One is to view the target's personal information, and the other is to delete the target.

On the Post page, users can view the newly published recipes of subscription targets,

and click the button attached to each recipe to view the details of recipes.

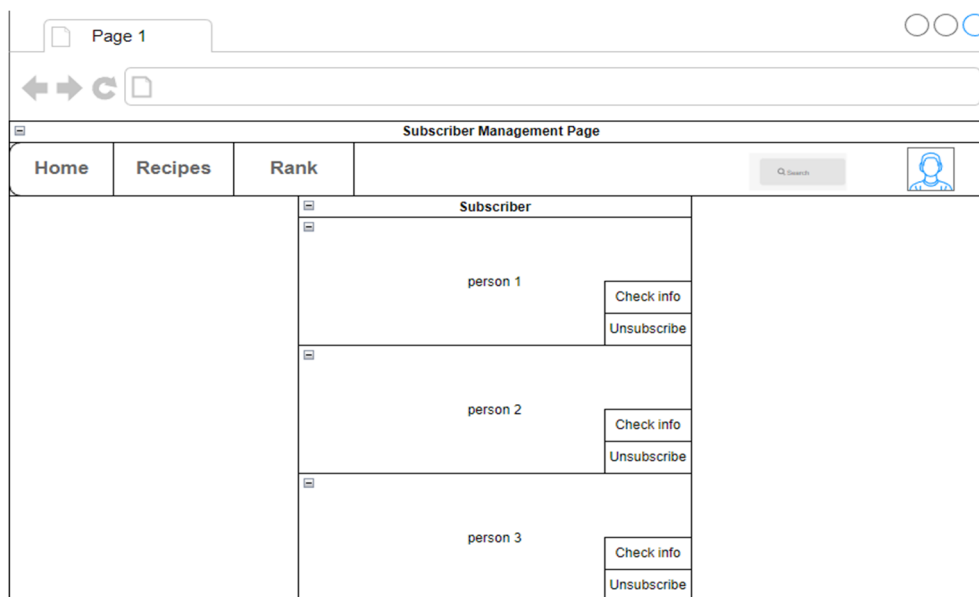


Figure 3.8 Subscriber management

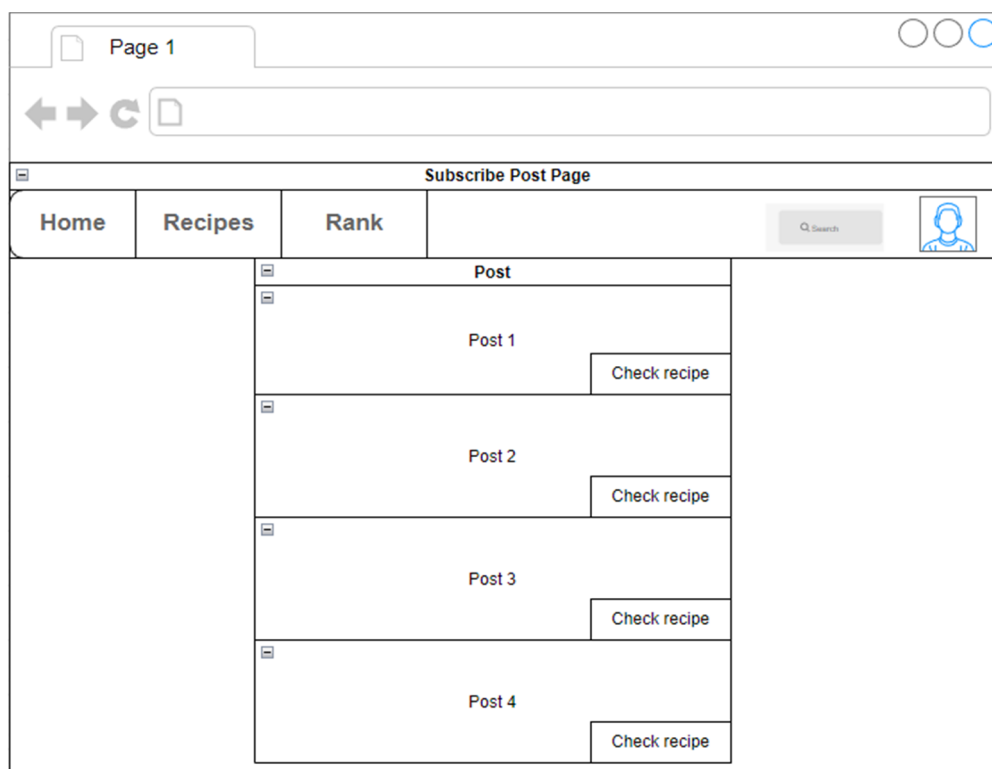


Figure 3.9 Subscribe Post

3.1.8 Favorites Page

On the Favorites page, users can manage their favorite recipes. Each recipe has two buttons. One is to view the recipe's information, and the other is to delete the recipe from their favorites.

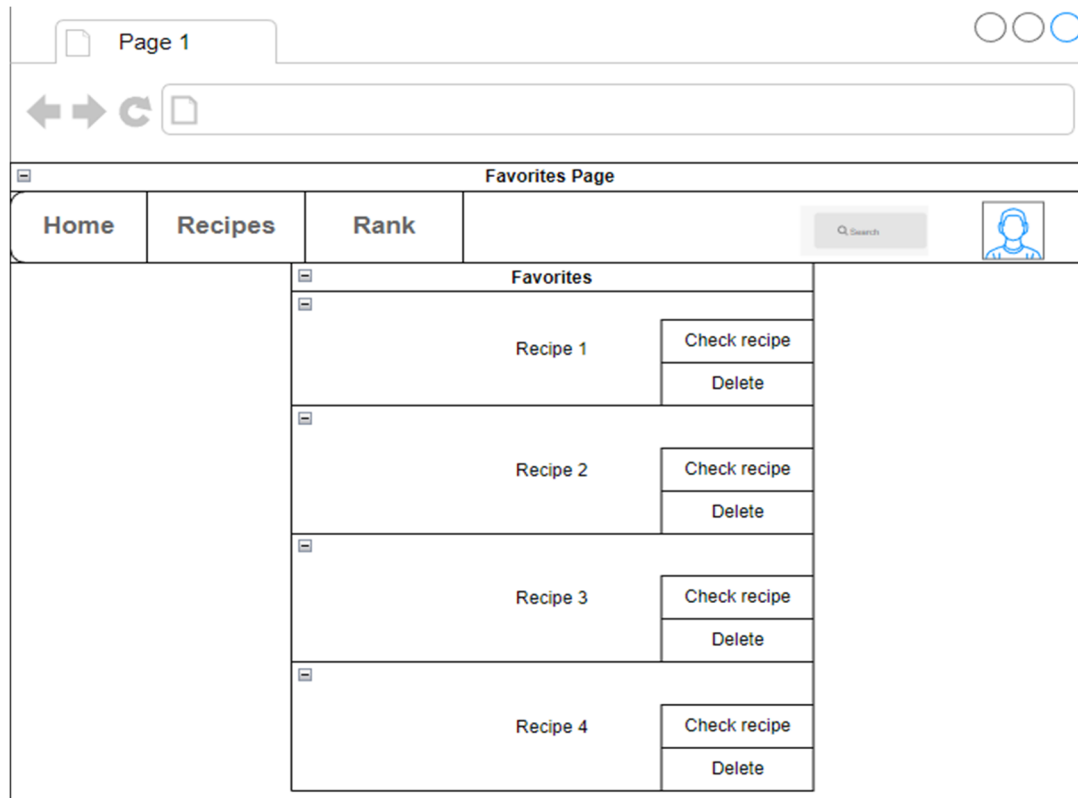


Figure 3.10 Favorites Page

3.1.9 Ranking List Page

On the Ranking List page, users can view the ranking list of popular recipes, and click the recipe name to view its details.

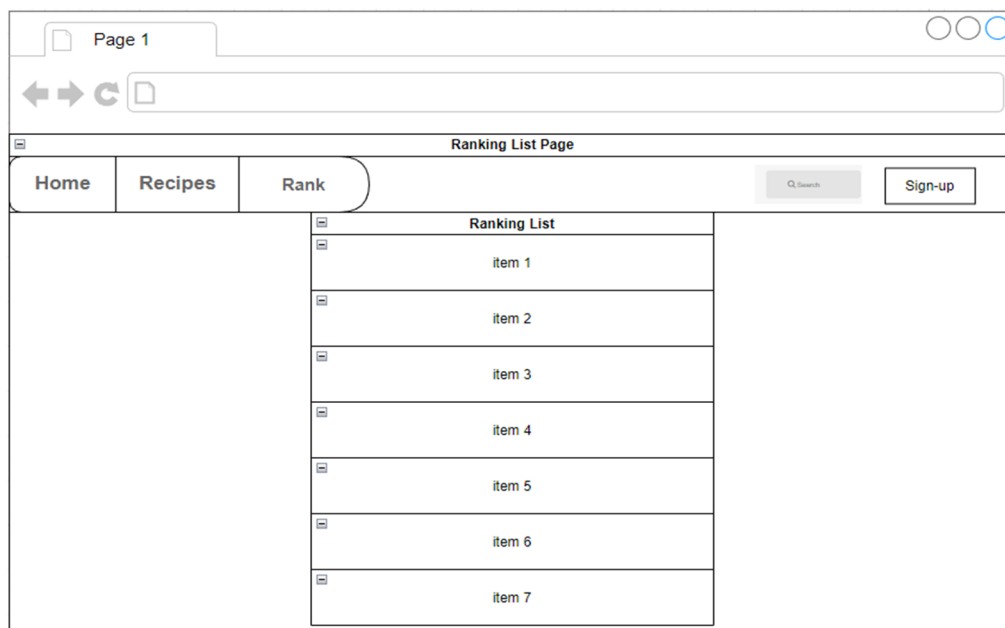


Figure 3.11 Ranking List Page

3.2 Flow Diagram

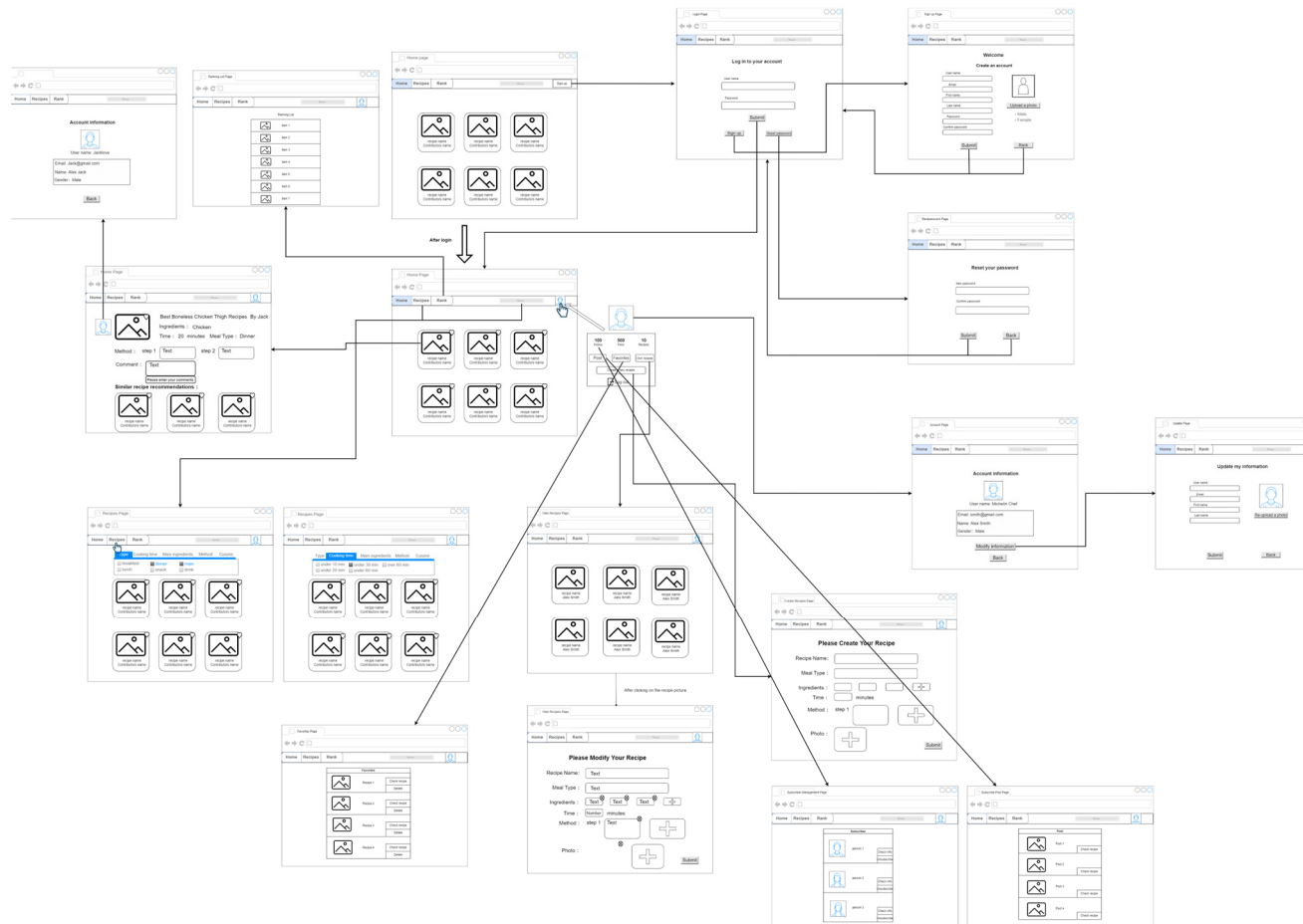


Figure 3.12 Page Relations

4 System Architecture

4.1 Architecture breakdown

4.1.1 Separate Back-end and Front-end System

We separate the whole system into two subsystem which is Back-end System and Front-end System. Two subsystems do not depend on each other and can interact with each other through Http Request/Response JSON.

The architecture of the system can be generally divided into 3 layers: presentation, business, data layer. Front-end System includes the Presentation Layer and Back-end System includes Business Layer and Data Layer.

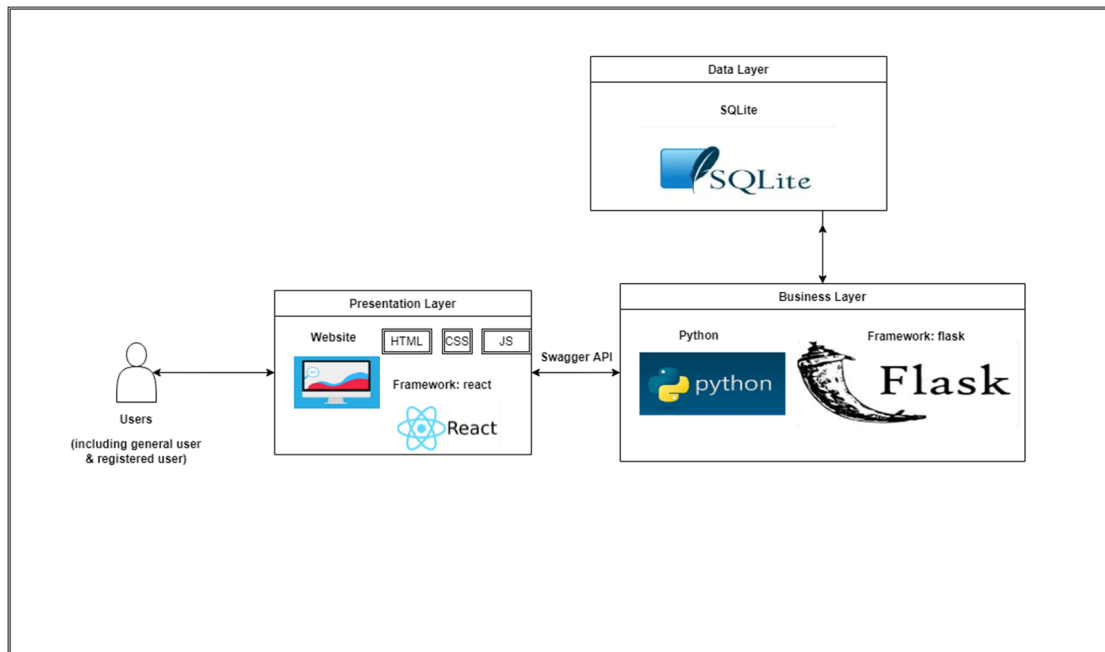


Figure 4.1 Architecture breakdown

4.1.2 Presentation layer

The presentation layer is what a system user sees or interacts with. When users express their needs, the presentation layer will provide users with application services. When receiving the instruction from the user, the presentation layer will send a request to the business layer, and change the content of the presentation layer according to the return result to feed back to the user.

4.1.3 Business layer

Business layer is the middle layer between presentation layer and data layer. This layer responsible for all business logic, we need to deal with any application logic that

is concerned with the retrieval, processing, transformation, and management of application data. And we also need to ensure data consistency and validity. In other words, when presentation layer have a request, business layer needs to get a solution by the logic we designed, then make operation to data layer and finally get the response to presentation layer.

4.1.4 Data layer

Data layer is where we store persistent and temporary data related to our system. It is usually be a database with structure designed by developer, we data layer gets a request from business layer, data layer should quickly do operations among select, insert, update and delete then response required data to business layer.

4.2 External actors

4.2.1 Users

User is the main external actor because user makes interaction with our system through web page. And users can be divided into two types: explorers and contributors.

4.2.2 Explorers

Explorers are those who search for recipes they interested and they could convey any opinion for any recipes.

4.2.3 Contributors

Contributors are those who maintain a set of recipes on our system, they can upload any recipes on our system depend on our data structure. They enrich the content of our website.

4.3 Technologies descriptions

4.3.1 Presentation layer

Front-end framework: React

We are using React framework in our front-end system. Compared to other frameworks, React introduces a concept called virtual DOM, which is interposed between JavaScript logic and the actual DOM. This concept improves web performance. React uses a declarative paradigm that makes it easy to describe applications. Building components through React makes the code easier to reuse and maintains high.

4.3.2 Business layer

Programming language: Python

Python is a widely used interpreted, high-level, general-purpose programming language, Python allows developers to express their ideas in less code.

Web framework: Flask

We use Flask as our back-end framework. Flask^[1] is a microframework which aims to keep code clean and easy to extend.

In our development, we mainly use Flask-Restful which is an extension for Flask. It makes support for building REST APIs quickly and it can automatically produce a swagger API doc with descriptions we want. This brings convenience for testing and front-end developer.

When interacting with database, we mainly use Flask-SQLAlchemy to do ORM operations, so we don't need to write any SQL language to do CRUD. With this method we can avoid SQL injection and ensure the security of the whole system.

As most of our system functionalities requires user authentication, we use JWT (JSON Web Token) as our authentication framework so that user can have access to some functionalities after login.

We use Flask Application Factory pattern to create out Flask app, so that we can separate logic between files and modules^[2]. It makes the code structure more readable and it brings convenience for co-operating development.

4.3.3 Data layer

Database: SQLite

SQLite is a relational database which implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine.

SQLite works well with Python because the Python standard library provides the sqlite3 module so there will be no configuration problems and don't need to do any installation.

And during development, database modification occurs frequently. SQLite is easy to do migrate with Flask-migrate module, which may improve development efficiency.

5 Reference

- [1] Wikipedia, “Flask is a micro web framework written in Python. ”[Online]. Available: [en.wikipedia.org/wiki/Flask_\(web_framework\)](https://en.wikipedia.org/wiki/Flask_(web_framework))
- [2] Todd Birchard, “Well-structured web apps separate logic between files and modules” [Online]. Available: hackersandslackers.com/flask-application-factory