VIETNAM - KOREA UNIVERSITY OF INFORMATION AND COMMUNICATION TECHNOLOGY
FACULTY OF COMPUTER SCIENCE

GRADUATION THESIS

TOPIC: SYSTEM FOR SHARING NUTRITIONAL INFORMATION AND RECIPES

Student Name: Le Nguyen Duy Nghia

Class: 20SE4

Major: Information Technology

Specialization: Software Engineering

Supervisor: Msc. Dang Thi Kim Ngan

Da Nang - 12/2024

VIETNAM - KOREA UNIVERSITY OF INFORMATION AND COMMUNICATION TECHNOLOGY
FACULTY OF COMPUTER SCIENCE

GRADUATION THESIS

TOPIC: SYSTEM FOR SHARING NUTRITIONAL INFORMATION AND RECIPES

Student Name:	**Le Nguyen Duy Nghia**
Class:	**20SE4**
Major:	**Information Technology**
Specialization:	**Software Engineering**
Supervisor:	**Msc. Dang Thi Kim Ngan**

Da Nang - 12/2024

ACKNOWLEDGEMENTS

Student life passes by quickly, but it remains a profoundly meaningful chapter of my life. During those years, I shared countless memories and moments of youthful passion and unyielding ambition with the Vietnam - Korea University of Information and Communication Technology, all aiming to build a brighter and better future. I want to express my deepest gratitude to the university, which has nurtured and supported me, shaping me into a useful member of society.

I also extend my heartfelt thanks to my supervisor, MSc. Đặng Thị Kim Ngân, who guided me through this thesis. Her support, feedback, and constructive criticism were invaluable in defining the direction and improving the quality of my work. My gratitude also goes to the other lecturers at the Vietnam - Korea University of Information and Communication Technology, who provided me with foundational knowledge and skills critical to my achievements today.

I would also like to thank my fellow students for accompanying me through the ups and downs, being my motivation to strive forward, and offering their support whenever I faced academic challenges.

Lastly, I am deeply grateful to my family and loved ones for their understanding, support, and steadfast presence. They have been my source of pride and motivation, enabling me to overcome all difficulties and seize my future.

The contributions, big or small, from all those mentioned above are priceless and have greatly shaped the quality of this thesis. I extend my sincerest gratitude to all of them.

DECLARATION

Name: Le Nguyen Duy Nghia

Student ID: 20IT323

Contact Number: 0762980159

Email: lndnghia.20it7@vku.udn.vn

Class: 20SE4

Program: Software Engineering

I am **Le Nguyen Duy Nghia**, hereby declare that this Graduation Thesis (GT) is my own work conducted under the guidance of Msc. Dang Thi Kim Ngan. The results presented in this GT are truthful and are solely my achievements. They are not plagiarized from any other works. All references in the GT, including images, tables, data, and quoted text, are clearly and fully cited in the bibliography. I take full responsibility for any plagiarism violations of university regulations.

Da Nang, day month year

Author of the Thesis

Le Nguyen Duy Nghia

ABSTRACT

In a rapidly evolving and diverse world, the demand for variety in cuisine is growing, emphasizing not only the creation of novel, appealing dishes but also their nutritional value. To address this, cooking and nutrition guidance systems have been developed. However, these systems often operate independently, creating unnecessary limitations and inconveniences for users interested in both culinary and nutritional aspects.

This thesis begins by analyzing the strengths and weaknesses of existing systems. Based on these analyses, a system for sharing nutritional information and recipes is proposed. This system serves as a flexible platform for users to share experiences and knowledge about cooking and nutrition. It offers detailed cooking guides paired with nutritional information, along with tools for monitoring health indicators and creating personalized diet and workout plans.

The system's interface is user-friendly and accessible, enabling users to conveniently access information without unnecessary complexity. Looking ahead, the system will continue to evolve and adapt to meet the increasing demands and expectations of users, incorporating the latest technologies and advanced features to enhance its value and utility compared to existing systems.

TABLE OF CONTENTS

LIST OF IMAGES

CHAPTER 1. INTRODUCTION TO THE TOPIC

1.1 Problem Statement

Current cooking websites often provide users with ways to create dishes without addressing the nutritional value of those dishes. There is no mention of whether the nutritional intake from these meals might affect users' health. Sometimes, the dishes on these platforms are not regulated for their preparation methods and nutritional components, leading to undesirable situations such as meals that are overly harmful, too high in calories, or even food poisoning.

Some systems offer articles on nutrition, but they are often simplistic and written in a generic style. Moreover, these systems do not integrate nutrition into the recipes themselves. This could reduce user satisfaction, especially for cooking enthusiasts who are also concerned about nutrition.

On the other hand, cooking or nutrition platforms are often just guides or articles. They generally lack user interaction or interaction between the users and the platform, creating a sense of boredom for the users.

Therefore, there is an urgent need for a system that can provide both cooking and nutrition information. Such a system should offer integrated articles on cooking that include the nutritional content of each dish. The system should also allow users to interact with each other, view their health metrics, and easily manage their diet and exercise routines. By meeting these requirements, a nutritional and recipe-sharing platform will improve user engagement, offering a comprehensive view of meals, nutrition, and health.

1.2 Objectives and Scope of the Topic

To address the limitations mentioned in Section 1.1, the goal of this project is to develop a system that integrates both cooking and nutrition information while enhancing user interaction and focusing more on users' health.

The scope of this project includes the development of a comprehensive platform where users can easily search for and access information about creative and innovative dishes while ensuring nutritional and health aspects are considered. The system will feature a special search tool that allows both traditional and image-based search functionalities.

Additionally, the platform will enhance user interaction, fostering the sharing of cooking experiences and nutritional knowledge, allowing the community to learn from

each other effectively. The system will also provide tools to calculate health metrics and manage diet and exercise routines, enabling users to conveniently and scientifically track and adjust their nutritional intake. Another key aspect of the project is designing a user-friendly interface, ensuring that users can quickly and easily navigate the system.

1.3 Solution Orientation

In terms of technology, the nutritional and recipe-sharing system is built on a web platform, utilizing cutting-edge technologies to ensure high performance and an optimal user experience. The user interface is developed with ReactJS to create smooth and flexible web applications. The server is built with Node.js, offering robust event handling capabilities, ensuring fast and efficient response times. Data is stored in MongoDB, providing scalability and flexible data management. Images and emails are stored and processed through AWS, ensuring stability and security. The system also uses Flask, combined with TensorFlow, Keras, and the VGG19 model, to provide an image-based search feature for dishes, offering users a fresh and convenient experience. Furthermore, the integration of Socket.io enhances real-time notifications, improving user interaction by enabling quick and effective connections between users.

In terms of algorithms, the system allows users to input their body metrics and then utilizes pre-integrated formulas to calculate BMI, TDEE, BMR, IBW, LBM, body fat percentage, daily calories burned, and the amount of water required daily, returning the results to the user.

By applying these technologies and algorithms, the project has contributed in the following ways:

- **Improved Searchability**: Users can easily search for dishes through the image search functionality. The system provides an attractive interface and a user-friendly experience.

Reduced Complexity: Older systems often failed to integrate cooking and health, forcing users who wanted both to jump back and forth between platforms, which created significant issues. Therefore, the integrated cooking and nutrition system eliminates this redundancy, allowing users to enjoy both features seamlessly without wasting time.**Improved Interaction**: The system offers a small forum where users can interact with each other, such as liking, sharing, or posting articles about nutrition. Additionally, the system has an account upgrade feature based on the number of followers to encourage interaction between users. This also enables users to unlock new features, satisfying their passion for becoming chefs or nutrition experts.

User-Centric Approach: The system focuses more on user health by allowing

users to plan their diet and exercise routines to improve their personal health.

1.4 Survey of the Current Situation

In the aftermath of the COVID-19 pandemic, the demand for home cooking and concern for nutrition among Vietnamese citizens has increased significantly. Social distancing and concerns about food safety have led many to spend more time cooking at home while focusing on maintaining a healthy diet.

According to a report from Market Research Vietnam Indonesia [1], 81% of Vietnamese consumers care about nutritional information on food packaging, and 61% admit that they lack confidence in nutritional advice due to often contradictory information. This demand has driven the development of websites related to cuisine and nutrition, such as Cooky.vn, Eva.vn, and Afamily.vn, which offer a wide variety of recipes and nutritional information, making it easier for consumers to plan balanced meals. The Top 10 Most Famous Cooking Websites in Vietnam [2] report also supports this trend.

Furthermore, according to a report from Vietnambiz [3], the usage rate of online food ordering services saw a significant increase between 2020 and 2021. Consumers not only sought delicious meals but also paid attention to the nutritional aspect to ensure their health during the pandemic.

To meet the growing demand, it is essential to develop a web-based system that integrates cooking and nutrition information. This system will allow users to easily search for and share recipes while accessing nutritional information efficiently, thereby improving overall health management. This approach will not only address current issues but also enhance the user experience in maintaining a healthy and balanced diet.

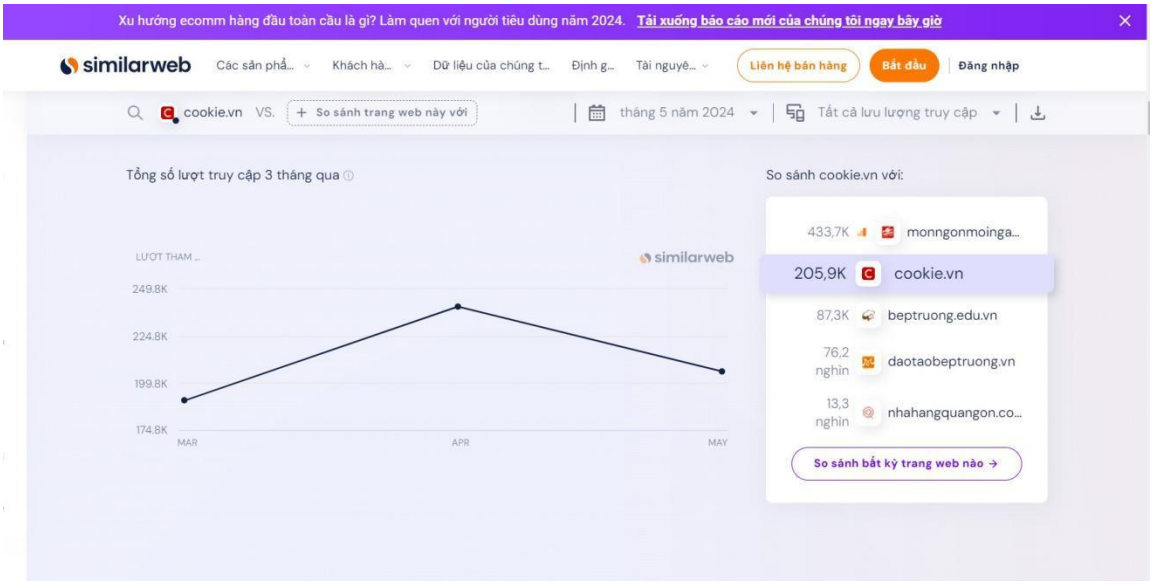


Figure 2.1: Website traffic image of Cooky

Additionally, nutrition apps like MyFitnessPal are becoming increasingly popular. According to a report by Semrush, MyFitnessPal received 14.5 million visits in April 2024, with users spending an average of nearly 8 minutes per session. Furthermore, the app boasts around 200 million registered users, highlighting its widespread popularity and the significant interest in tracking and improving dietary habits **MyFitnessPal traffic [5]**.

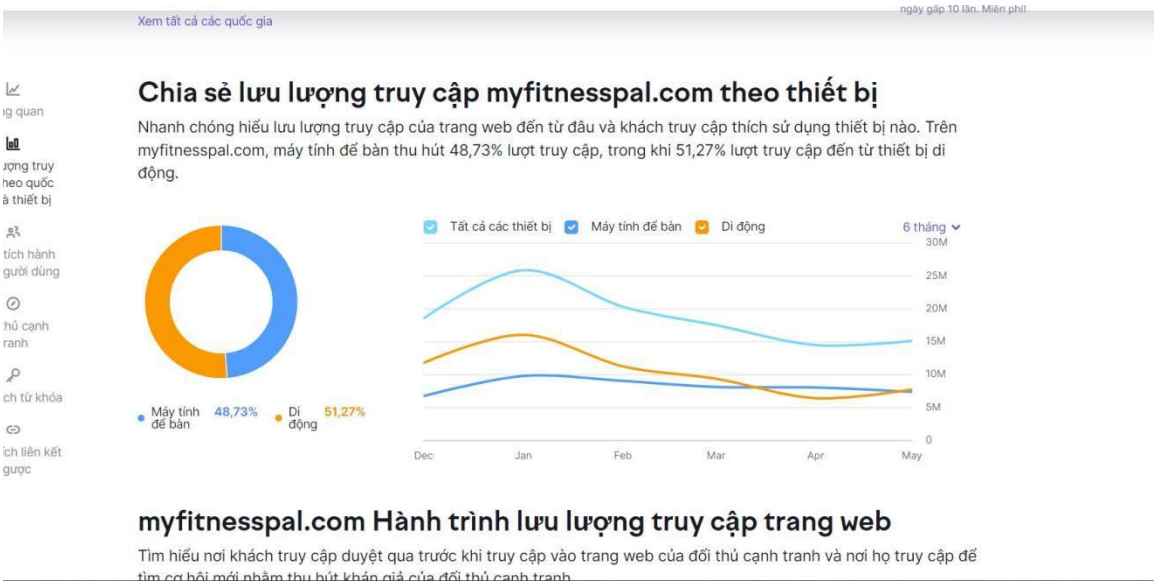


Figure 2.2: Image showing the traffic statistics of MyFitnessPal.

1.5 User Survey

In modern society, the demand for diverse and high-quality dishes is increasing. Most users want to explore and challenge themselves in creating new dishes while sharing culinary knowledge with the community. Cooking is not just seen as a pastime but also as a way to express creativity and community spirit. In addition, many users are concerned about their health, diet, and nutrition. They not only want to enjoy delicious dishes but are also particularly interested in choosing ingredients and cooking methods to ensure their meals are nutritionally balanced and healthy. This creates a diversity in consumer needs and desires regarding food.

A specific example of this change can be seen through the development of websites and apps that share cooking recipes and nutrition information. For instance, the website Cooky.vn attracted around 205.9 thousand visits in May 2024, from users eager to explore and share new recipes Cooky.vn traffic [4].

1.6 Project Structure

The rest of the thesis is organized as follows:

- **Chapter 1** Summary: The Need for an Integrated Cooking and Nutrition Platform. Integrating nutritional information directly into recipes. Providing user interaction features for sharing experiences and knowledge. Allowing users to track health metrics and manage diet/exercise routines.
- **Chapter 2** presents the survey of the current situation, business research process, and describes system functionalities through use case diagrams, indicating the flow of activities within the system. It also includes the non-functional requirements of the system.
- **Chapter 3** discusses the system architecture, including detailed designs for packages, user interfaces, and database design.
- **Chapter 4** covers the technologies used, such as ReactJS, Node.js, MongoDB, and Socket.io. It also addresses the implementation, testing, quality assurance, and performance of the system when delivered to users.
- **Chapter 5** discusses the challenges and issues encountered during the system's development and the solutions to address them. It also highlights the significant contributions of the thesis. Finally, it concludes with the results achieved and outlines potential future directions.

CHAPTER 2. SURVEY AND REQUIREMENT ANALYSIS

2.1 Survey of Existing Systems

Analyzing existing systems in both the international and Vietnamese markets will help me better understand the strengths and weaknesses of each system. This provides a solid foundation for building a new system that is innovative and meets user needs in the best possible way.

Through analysis, I can identify the strengths and weaknesses of existing systems. The strengths might include high interactivity, providing diverse and detailed information, or a user-friendly interface. For example, the website Cooky.vn has attracted millions of visits each month from users who want to explore and share new recipes.

However, the systems also have weaknesses, such as limited flexibility, a lack of nutritional information, or difficulty in user interaction. A typical example of a weakness might be the lack of interactivity between users on current recipe websites. Below is a table comparing and analyzing the existing systems:

Name	Data Information	Sharing	Search
Cooky.vn [6]	The system provides recipe-related information but does not integrate nutrition data.	The system allows users to share their cooking knowledge.	The system allows users to search for articles in a traditional (text) format.
Nutrition Association [7]	The system provides nutritional information.	The system only allows administrators to post articles.	The system allows users to search for articles in a traditional (text) format.
Health and Life [8]	The system provides articles sharing about food and nutrition.	The system only allows administrators to post articles.	The system allows users to search for articles in a traditional (text) format.

Table 2.1: Feature Analysis of Some Existing SystemBased on these analyses

I can apply the advantages and learn from the mistakes of existing systems to build a new system with improvements and unique strengths. This system will integrate all three types of information as analyzed in the table above, allowing users to post content while being managed by administrators to ensure a transparent and credible environment. In addition to the regular search function, the system will also incorporate an image-based search feature to make it easier for users to find information. This will

help the system stand out, attract users, and meet their expectations effectively.

The new system will focus on key modules including:

- **User Interaction Forum Module**:
    - Facilitates interaction between users by providing features such as posting questions, replying to discussions, and sharing experiences.
    - Enables users to engage with the system by reporting inappropriate content, suggesting improvements, or providing feedback.
- **Recipe and Blog Interaction Module**:
    - Allows users to interact with cooking posts and blogs through actions like liking, commenting, and bookmarking.
    - Supports the creation and management of nutrition albums, enabling users to share curated collections of recipes or related content.
- **Health Module**:
    - Offers health indicators to help users monitor their physical well-being, including BMI, caloric needs, and other metrics.
    - Provides tools for meal planning and workout scheduling to support users in achieving their health and fitness goals.
- **Admin Management Module**:
    - Equips system administrators with tools to oversee and manage user-generated content, such as forum posts and recipe submissions.
    - Includes features for monitoring user activities, resolving disputes, and ensuring compliance with community guidelines.

Additional Considerations:

- **Monetization Strategy:**
    - Explore potential revenue streams by introducing premium features, such as exclusive content or advanced tools for users.
    - Consider partnerships with food vendors or incorporating targeted advertising to generate additional income.
- **Scalability:**
    - Design the system architecture to handle a growing user base and increasing data volume efficiently.

- Plan for future expansions by ensuring compatibility with new features, technologies, and integrations.

- **Data Security:**
  - Implement advanced encryption and authentication mechanisms to safeguard user data against breaches.

Regularly update security protocols and conduct audits to maintain a secure and trustworthy platform.

2.2 Overview of Features

2.2.1 General Use Case Diagram

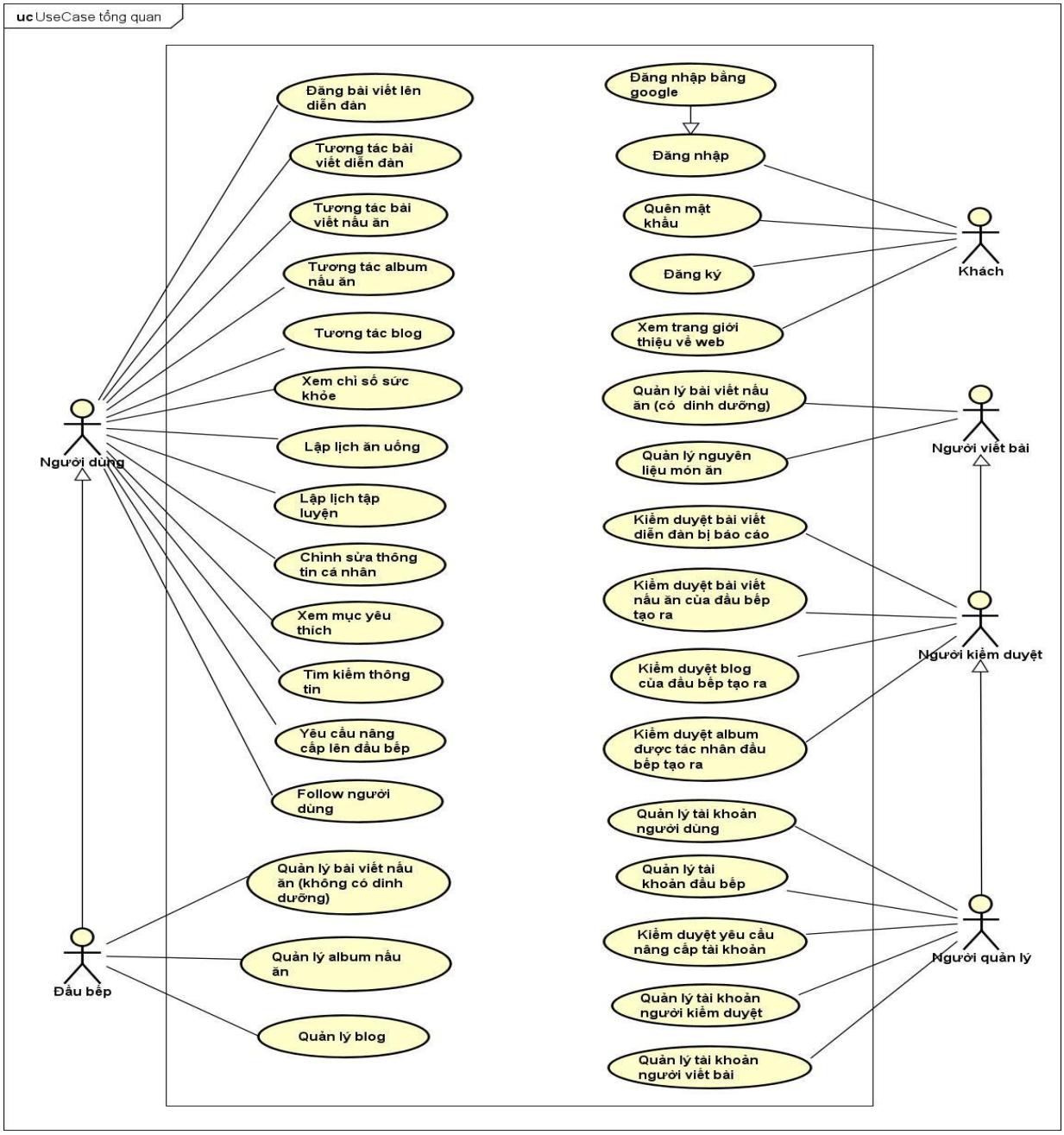


Figure 2.3: Overview Use Case Diagram

The system in the thesis is divided into two main parts: the website system for users and the management system for the administrative team. From the overview use case diagram (Figure 2.3), we can observe the following:

a. Actors in the System

The system includes 5 main actors, and there is also the "Guest" actor, who can view the website's introduction page, use the login function, reset the password, or register for a user account. Other functions require the user to log in before they can access them.

Actors in the User System

- **User**: This is the primary actor interacting with the system. The user can post questions and articles on the forum, view articles about cooking guides, nutritional albums, or blogs sharing cooking tips, etc. Additionally, the user can interact with and follow other users, as well as view personal health metrics based on which they can create meal plans or workout schedules to maintain a healthy lifestyle.
- **Chef**: Users can become a "Chef" actor once they reach 1,000 followers and submit a request to the system. If they do not reach the required number of followers, they can submit an upgrade request along with proof of experience for approval. Once upgraded, the Chef actor gains access to new functions, such as managing cooking posts without nutrition information, as well as the ability to manage their created recipe albums and blogs.

Actors in the Management System

- **Administrator**: At the highest level, the administrator has full access to all functions in the system. Their main tasks include upgrading user accounts to "Chef" status and managing accounts across the system.
- **Moderator**: Below the administrator, the moderator handles reported posts from users, manages posts created by "Chefs", and has the same rights as a content creator.
- **Content Creator**: At the lowest level, content creators have the right to create cooking posts with nutritional information and manage food ingredient details.

b. Types of Posts in the System

- **Forum Posts**: These are posts where users share experiences, achievements, or ask questions about cooking, nutrition, and health. This is a place where users can interact, learn from each other’s experiences, and seek advice on various food and health-related topics.

- **Cooking Posts**: These posts provide cooking instructions for users. This type of post is divided into two main categories:
    1. **Posts without nutritional information**: These posts only focus on providing detailed recipes and cooking instructions without nutritional details, typically created by "Chef" actors.
    2. **Posts with nutritional information**: These posts provide both cooking instructions and nutritional information about the dish. These posts are created by system "Admin" actors to help users understand the nutritional value of the dishes they prepare.
- **Blog**: Blogs contain posts about cooking tips, experiences, nutrition, and the latest news in the field. The content of blogs not only helps users improve their cooking skills but also provides valuable information about health, nutrition, and the latest food trends.

Cooking Albums: These are collections of cooking posts organized by specific themes, allowing users to easily find relevant information. These albums may include seasonal dishes, holiday-specific recipes, or special diets. The goal of albums is to create an organized library that helps users find the recipes and information they need quickly and conveniently.

2.2.2 Use case diagram breakdown: 'Forum post interaction

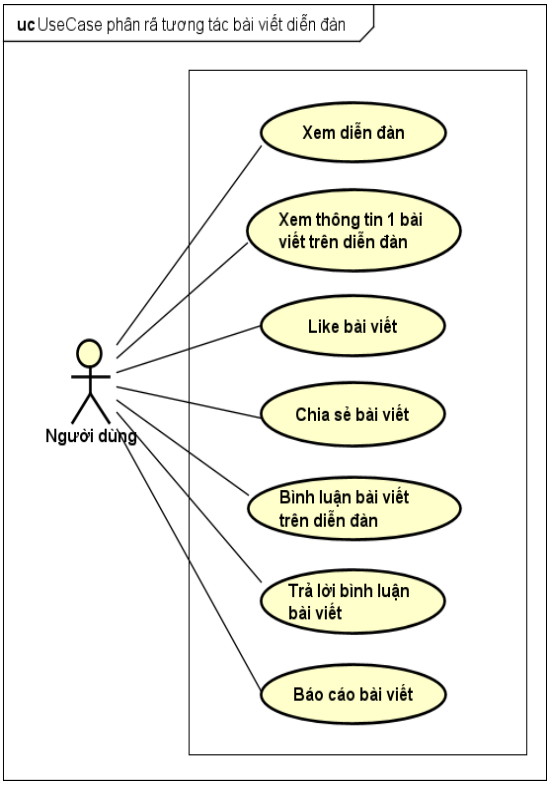


Figure 2.4: Breakdown use case diagram for "Forum post interaction"

From the analysis of the use case diagram for forum post interaction (Figure 2.4), we can identify the main functions that users can perform, such as viewing the list and details of posts on the forum. Users can also like, comment, reply to comments, and share posts. Additionally, if a user detects a post that violates community standards, they can report the post to the administration team to maintain the quality of content and ensure the system remains healthy.

2.2.3 Use case diagram breakdown for 'Cooking post interaction

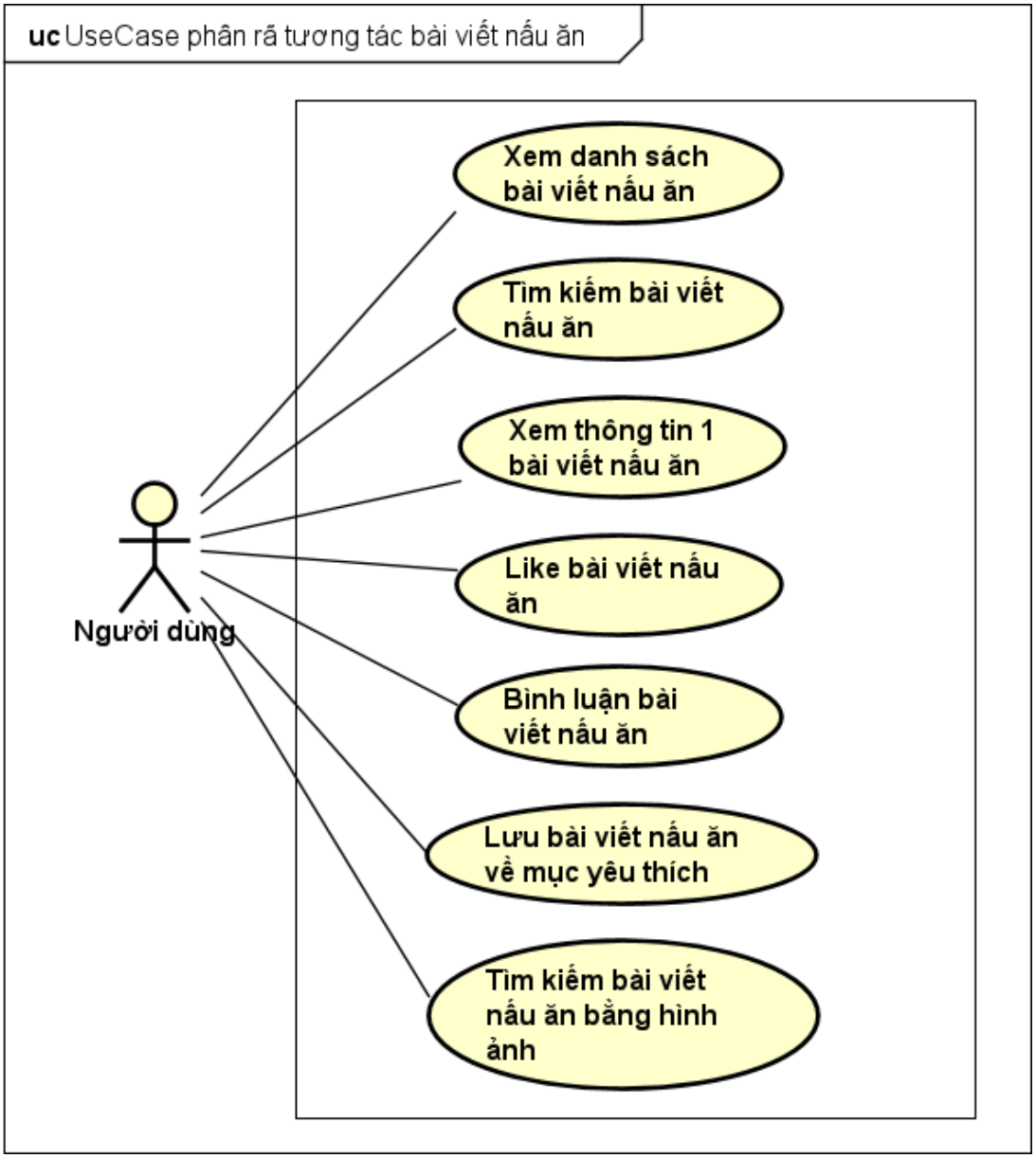


Figure 2.5: Use case diagram breakdown for "Cooking post interaction"

From the analysis of the use case diagram for cooking post interaction (Figure 2.5), we can see that users can view the list and details of cooking posts. In addition, users can like the posts, comment to exchange opinions, and share experiences with others.

They can also save their favorite posts into a personal section for easy access later. Notably, the system also integrates an image-based search function, making it easier and more convenient for users to find information related to the dishes they are interested in.

2.2.4 Use Case Diagram Breakdown: Cooking Album Interaction

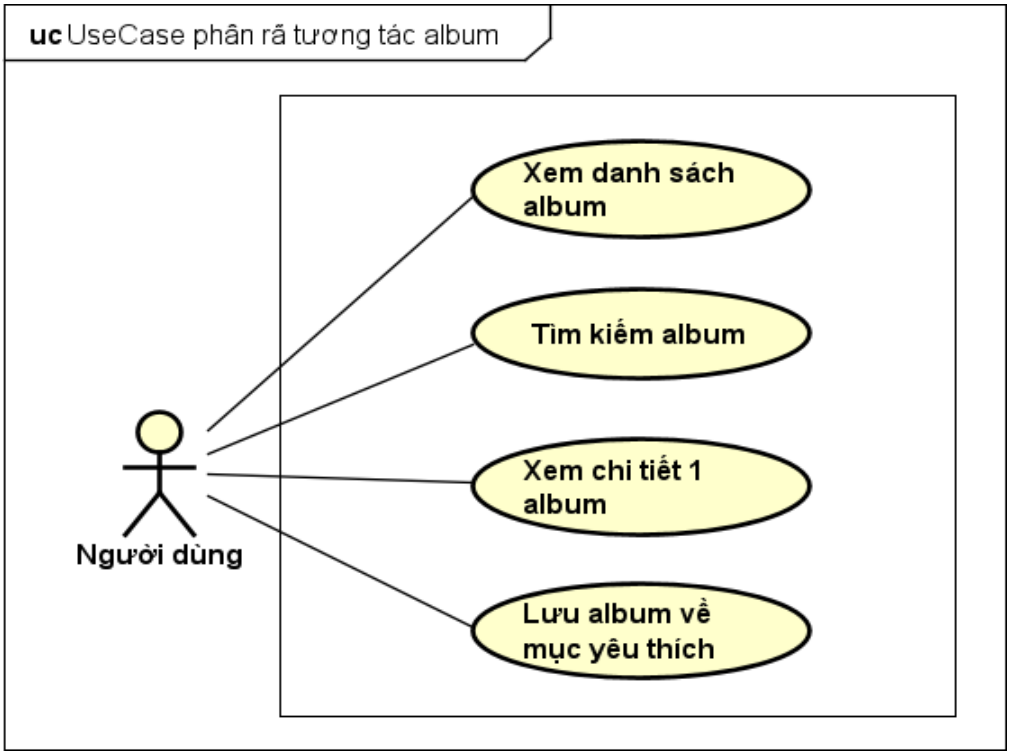


FIGURE 2.6: Use Case Diagram Breakdown - Cooking Album Interaction

From analyzing the use case diagram for album interaction in Figure 2.6, we can see that users can interact with albums by viewing the album list, viewing the details of a single album, liking, saving albums to their favorites, or searching for albums."

2.2.5 Use case diagram breakdown 'Blog interaction'

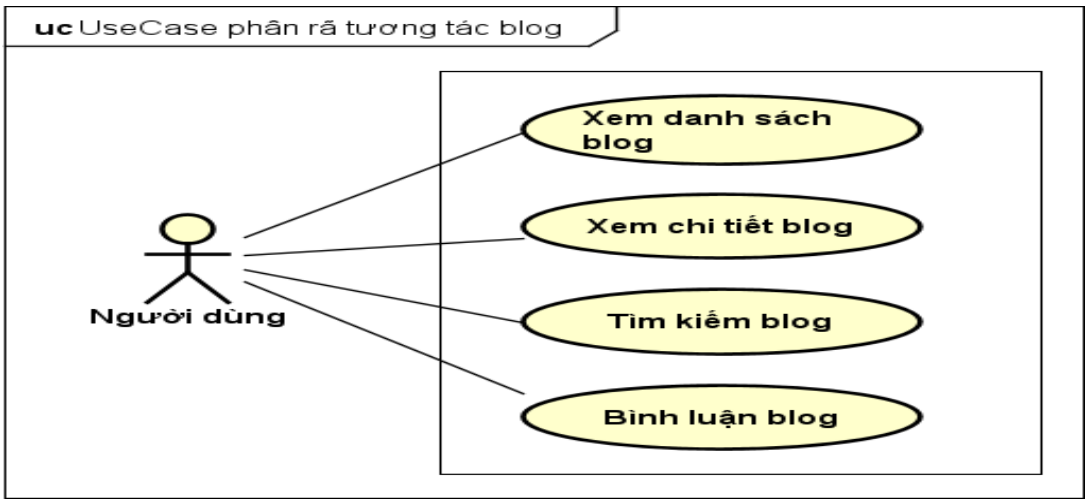


Figure 2.7: Use case diagram breakdown 'Blog interaction'

From the analysis of the blog interaction use case diagram 2.7, we can see that

users can interact with the blog by viewing the list, viewing the details of the blog, commenting, or searching for blogs.

2.2.6 Use case diagram breakdown "Managing cooking posts (without nutrition)"

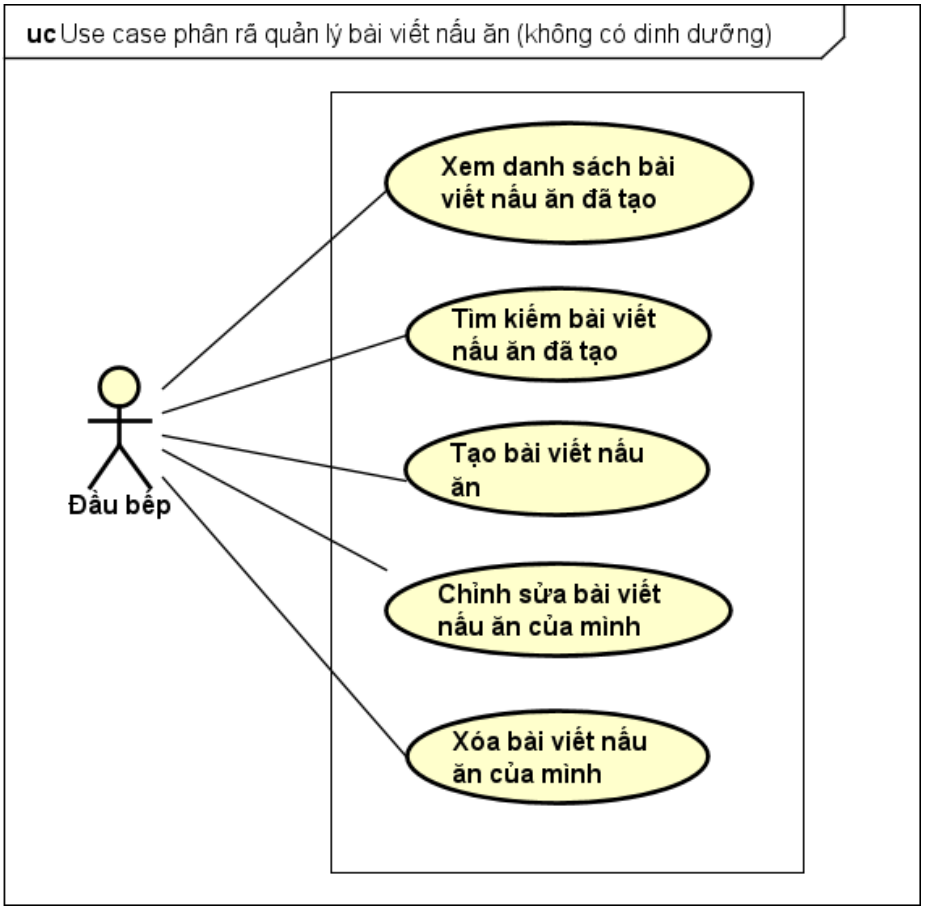


Figure 2.8: Use case diagram breakdown "Managing cooking posts (without nutrition)"

From the analysis of the use case diagram 2.8, we can see that chefs can manage and create cooking posts (these posts are created solely by the chef and do not include nutritional information of the dish and ingredients) and share them with the community on the system.

2.2.7 Use case diagram breakdown "Managing cooking albums"

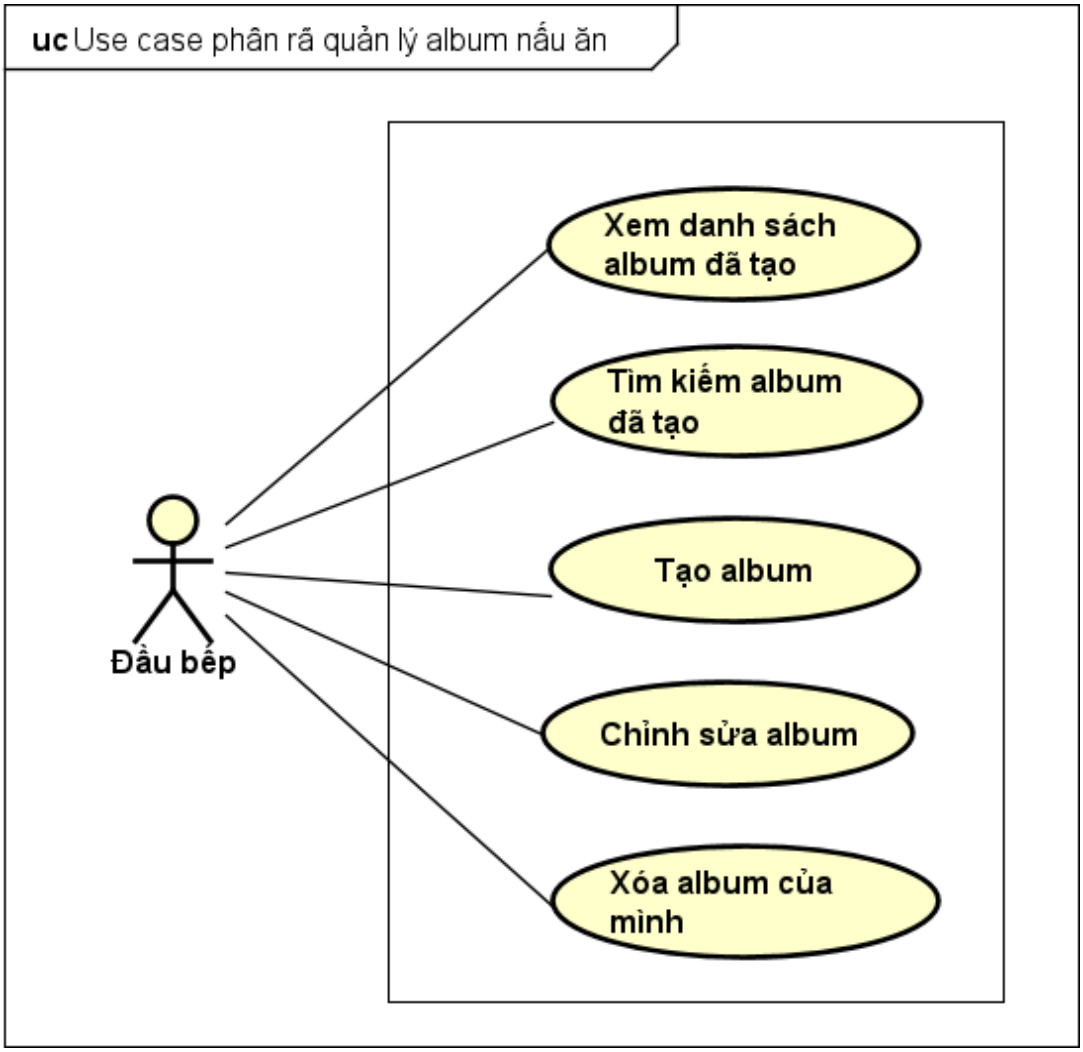


Figure 2.9: Use case diagram breakdown "Managing cooking albums"

From the analysis of the use case diagram 2.9, we can see that chefs can manage and create albums, which include a collection of cooking posts (with a common theme, such as health, nutrition, or holidays, etc.), and share them with the community on the system.

2.2.8 Use case diagram breakdown "Managing blogs"

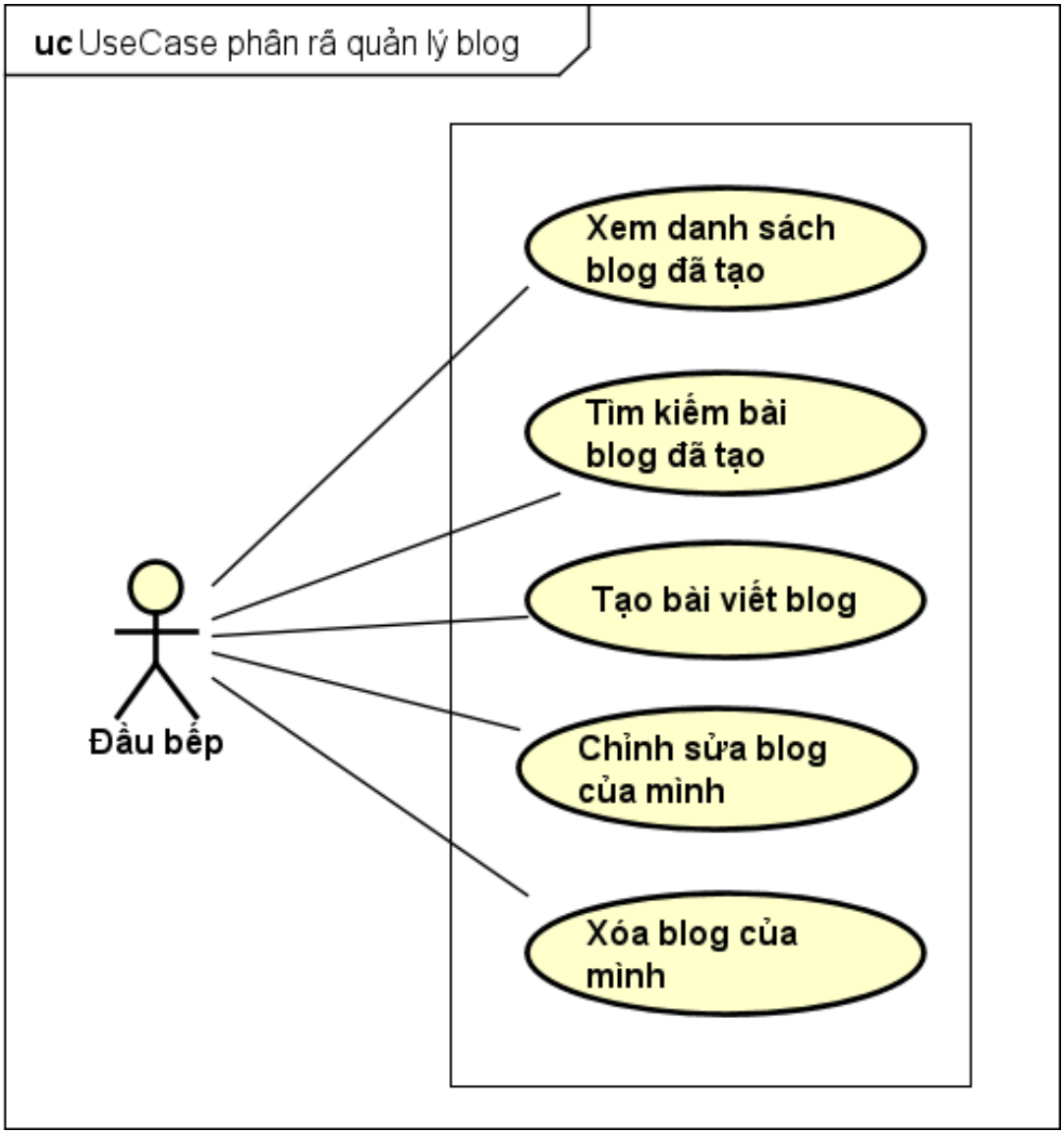


Figure 2.10: Use Case Diagram Breakdown "Managing Blogs"

From the analysis of the use case diagram 2.10, it can be seen that the chef can manage and create nutrition blogs, cooking tips, kitchen blogs, etc., and share them with the community on the system.

2.2.9 Use Case Diagram Breakdown "Managing Cooking Posts (with Nutrition)"

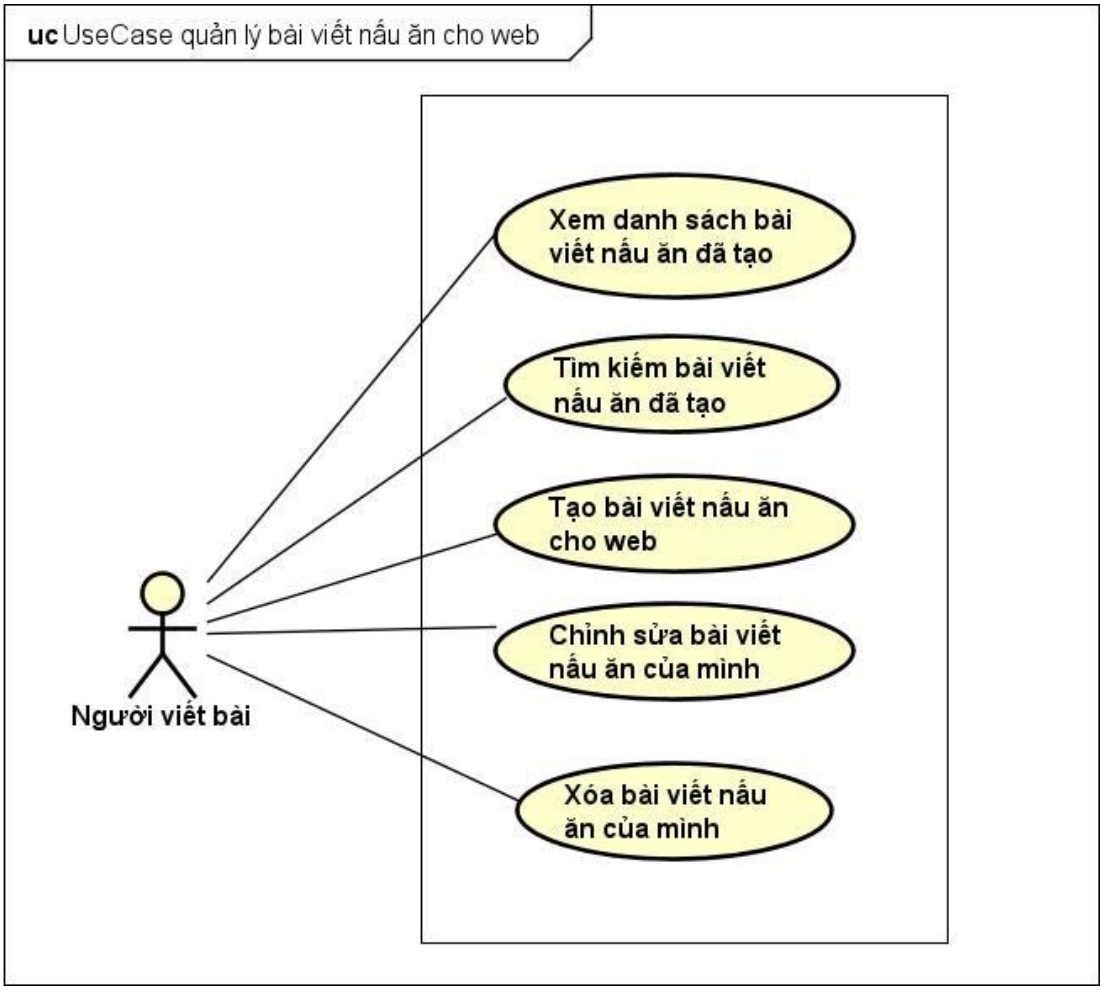


Figure 2.11: Use Case Diagram Breakdown "Managing Cooking Posts (with Nutrition)"

From the analysis of use case diagram 2.11, we can see that: The writer has the function of creating and managing cooking posts with nutritional information for the system. These cooking posts are different from those created by chefs, as they include additional nutritional details.

2.2.10 Business Process

In this section, I will dive into the business process, including the related use cases that work together to perform a specific task in the system.

a. Activity Diagram for Moderating Reported Posts and Questions on the Forum

This diagram combines two use cases: the use case "Report Post" (2.4) and the use case "Moderate Reported Forum Posts" (2.3).

By analyzing the diagram, we can see the flow of activities when a user reports a post for violation. Initially, the user selects a post and clicks the "Report" button. The system will display a form for the user to enter the reason for reporting. The user fills out the violation reason and submits it to the system. The system then forwards the report to the moderator for review. The moderator will assess whether the post violates the rules. If it does, the post will be deleted, and the user's violation count will be increased by one. If the post does not violate any rules, the moderator has the authority to reject the report and leave the post unchanged.

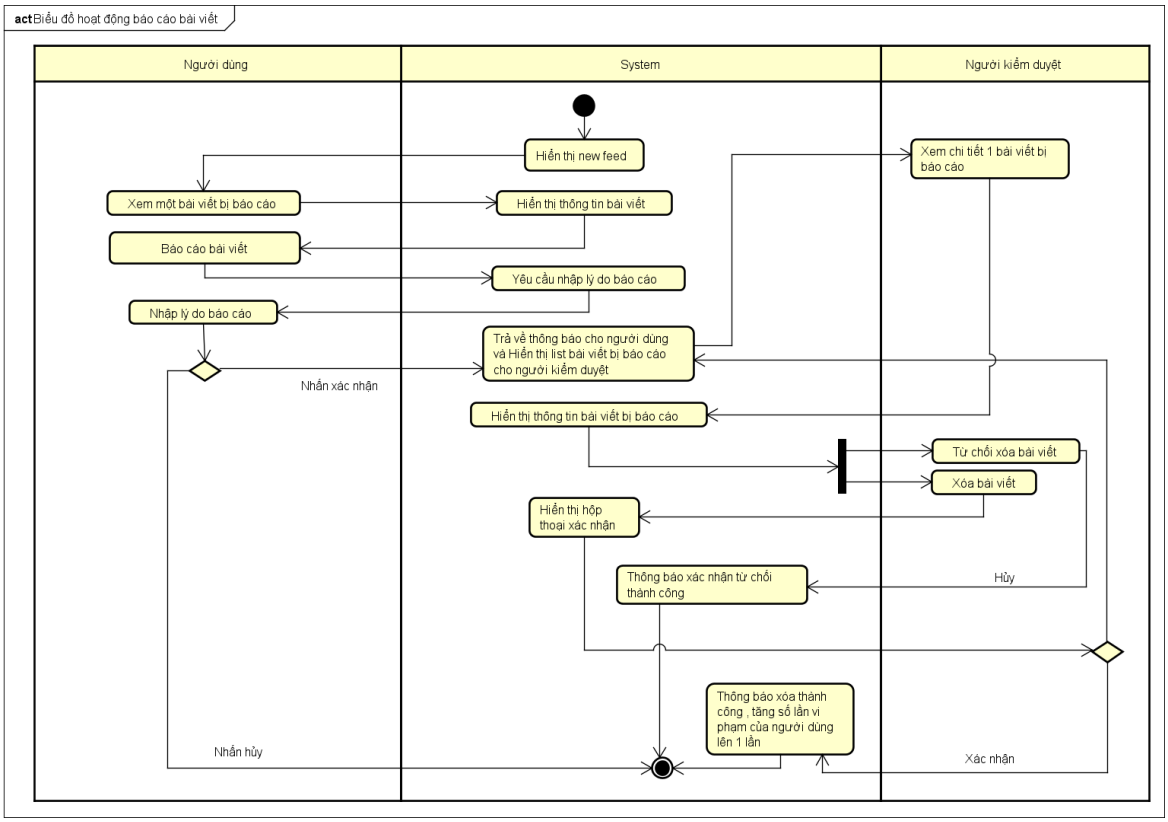


Figure 2.12: Activity Diagram for Moderating Reported Forum Posts and Questions

b. Activity Diagram for Upgrading a User to Chef

This diagram is a combination of two use cases: "Request to Upgrade to Chef" (2.3) and "Manage Upgrade Request" (2.3).

From the diagram analysis, we can see that there are two ways for a user to upgrade their account to Chef. The first is by meeting the condition of 1,000 followers, and the user will send a notification through the system to the admin for approval. The second is by submitting a form proving their experience in writing cooking and nutrition posts, which will be sent to the admin for review. The admin is responsible for deciding whether to upgrade the user’s account to Chef. If the conditions are met, after the admin approves the upgrade, the system will send a confirmation email to the user’s account.

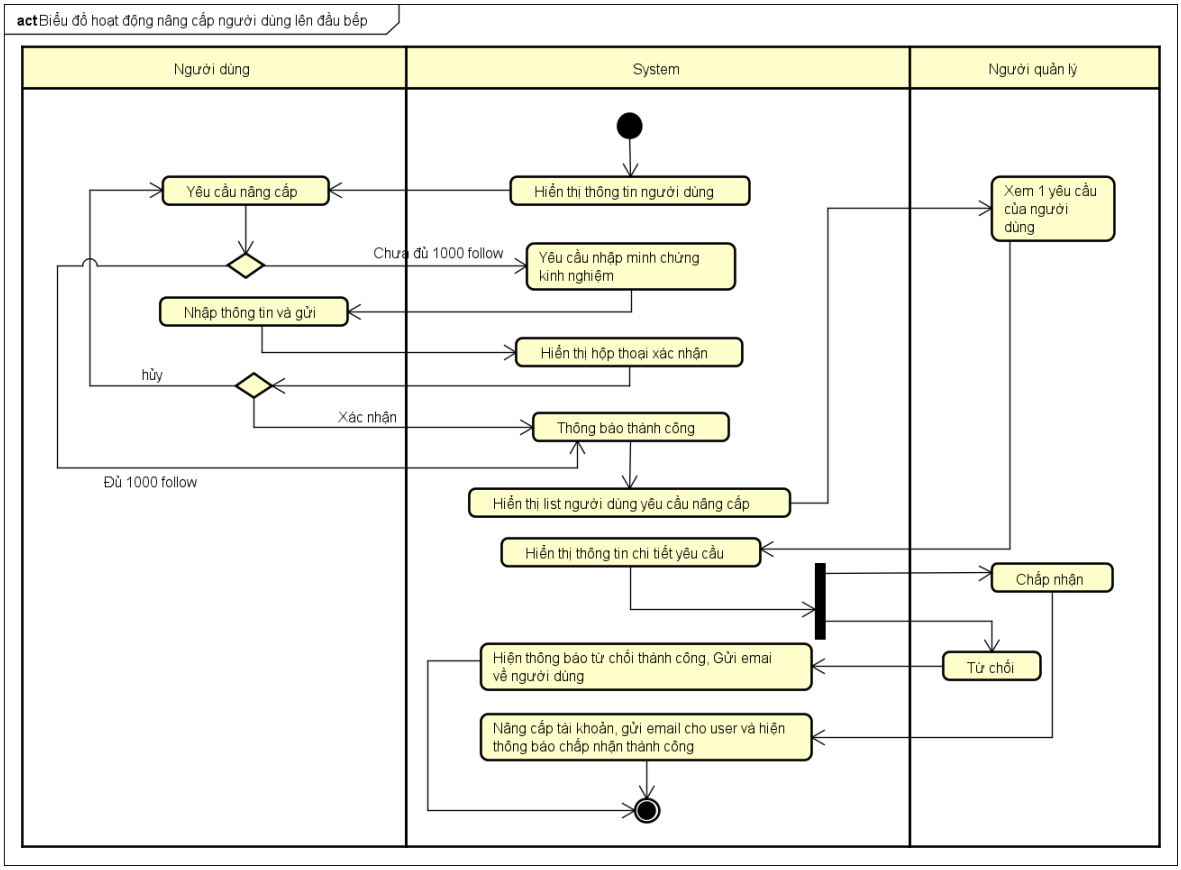


Figure 2.13: Activity Diagram for Upgrading a User to Chef

2.3 Functional Specification

2.3.1 Use Case Specification: "View Health Metrics"

The metrics that can be viewed include: BMI, TDEE, BMR, IBW, LBM, body fat percentage, calories burned during daily activities, and the required daily water intak

2.4 Non-Functional Requirements

The system not only provides functional requirements but also non-functional requirements such as user-friendliness, ease of use, scalability, high reliability, enhanced user interaction, and high performance:

- **User-Friendliness and Ease of Use**: The system offers an intuitive interface with an attractive design, making it user-friendly. Functions are clearly categorized and detailed for easy access.
- **Scalability**: The system offers an adaptable interface that works seamlessly across all devices such as tablets, phones, and laptops. The source code is modular and clearly separated, making it easy to upgrade and maintain the system.

- **High Reliability**: The verification of nutritional values from the Vietnamese food composition table, compiled by the Ministry of Health [9], ensures that the information provided on the system is reliable. Additionally, strict measures are applied to control content, ensuring only high-quality and valid information is shared.
- **Enhanced User Interaction**: The system provides a flexible and user-friendly forum, allowing users to interact with each other and with the content on the forum. This fosters a strong and vibrant community.

High Performance: Using high-performance frameworks such as ReactJs and ExpressJs, the system ensures smooth and fast operation, providing the best experience for users.

CHAPTER 3. DETAILED ANALYSIS AND DESIGN OF THE SYSTEM

3.1 System Architecture Design

3.1.1 Software Architecture Selection

In the system structure shown in Figure 3.1, the user interface is developed using the React.js framework to interact with the server. On the server side, the design principle of RESTful API is applied, using Node.js and Flask for the backend (Flask is used to write API endpoints for image-based search). Data is exchanged between the client and server in JSON format, facilitating flexible and efficient communication. The system also uses MongoDB as the database for storing data.

Additionally, the system integrates AWS (Amazon Web Services) services such as Amazon S3 for large data storage and Amazon SES (Simple Email Service) for automated email sending. Integrating these services provides flexibility and reliability in managing and processing data, as well as in communicating with users via email.

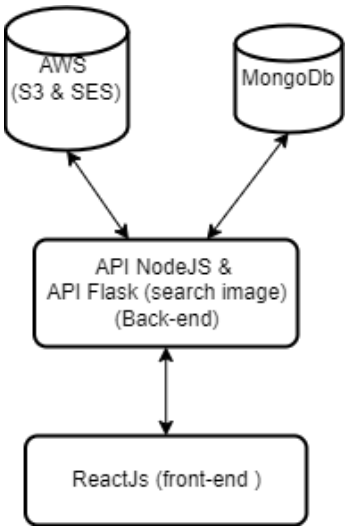


Figure 3.1: System Overview Design

The system's design follows the MVC (Model-View-Controller) architectural pattern. This pattern not only helps in creating applications with a clear organizational structure but also supports the development, maintenance, and expansion of the project in an easier manner. In the MVC model, the main components are clearly divided and have distinct responsibilities:

- **Model**: The Model represents the application's data and handles all database-

related operations.

- **View**: The View is the component that displays the user interface and presents information to the end user. It is responsible for rendering data and interacting with the user.
- **Controller**: The Controller acts as the application's control center, receiving user requests through the View and performing the corresponding actions. It contains the application's logic and coordinates interactions between the Model and View.

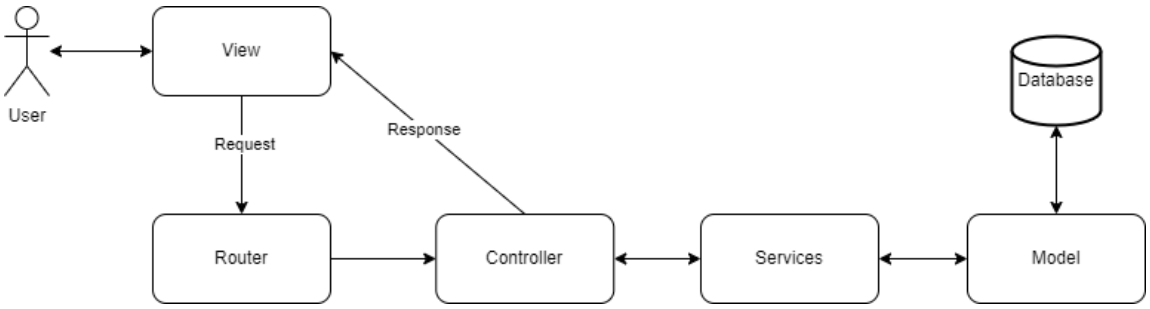


Figure 3.2: System Design Following the MVC Model

Figure 3.2 represents the overall system design based on the MVC model. The operation process begins with the View layer, where the user interface is displayed, and information is presented to the user. The user generates requests and sends them through the View interface to the server. The server receives the request via a Router, a component responsible for directing user requests to the corresponding Controller layer. The Controller layer serves as the heart of the system, where it calls the appropriate Services. The Service receives and processes the requests from the Controller, executing business logic and interacting with the database through the Model. Once the processing is complete, the Service returns the requested data to the Controller. The Controller then forwards this data to the View. Finally, the View displays the information to the user, completing the workflow of the system. This creates a continuous and efficient process, with each layer playing a crucial role in processing and transmitting data in a coherent manner.

3.1.2 Overall Design

a, Server-side Package Dependency Diagram for Node.js

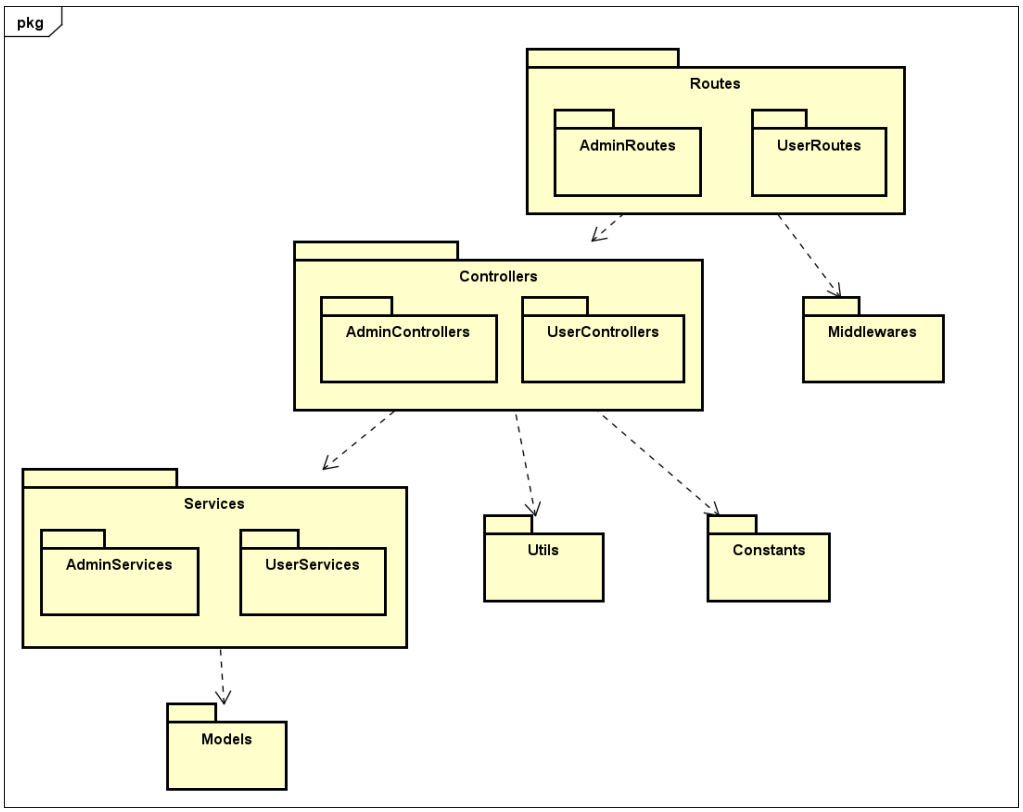


Figure 3.3: Server-side Package Dependency Diagram for Node.js

Server-side Package Dependency Diagram for Flask

The server side is designed with interdependent packages, including: Routes, Controllers, Middlewares, Services, Utils, Constants, and Models. These packages are organized in a structured and specialized manner to optimize the development and maintenance process of the system.

- **Routes:** This package is responsible for declaring and defining API endpoints, forwarding user requests to Controllers. It helps separate routing logic from data processing logic.
- **Controllers:** Controllers handle requests from Routes and forward them to Services for processing. They act as intermediaries between users and data, controlling the flow of data within the system.
- **Middlewares:** Middlewares play a crucial role in user authentication, permission checking, and removing unwanted data before it is forwarded to Controllers.
- **Services:** This package receives and processes requests from Controllers, executing the system logic and interacting with the database through Models.

Services act as a bridge between application logic and data.

- **Model:** Models contain classes corresponding to database tables, providing necessary functions for data manipulation, such as create, read, update, and delete (CRUD).
- **Utils:** This package contains utility functions used across the entire system, including functions like sending emails, handling system errors, and other supporting tasks.
- **Constants:** Constants store system constants and returned messages, making it easier to manage information and encoding.

b, Package Dependency Diagram for the Frontend

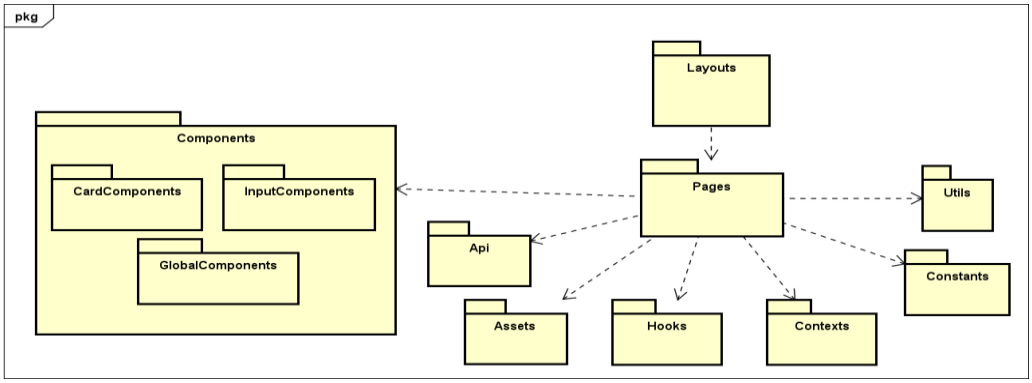


Figure 3.5: Package Dependency Diagram for the Frontend

The user interface is designed with a series of structured packages to optimize the development and maintenance of the system, including:

- **Layouts**: The Layouts package is the highest-level package in the system, responsible for holding and displaying the pages in the system, providing a common framework for the entire application.
- **Pages**: The Pages package contains the interface for the pages in the system, logically organized to allow for easy management and reuse.
- **Components**: The Components package contains reusable interface components, helping to create pages and user interfaces in a flexible and easy manner.
- **Api**: The Api package is responsible for connecting to the backend API, providing means to communicate with and query data.
- **Utils**: The Utils package contains rules, configurations, and utility functions to support the system, helping to manage and reuse resources efficiently.
- **Hooks**: The Hooks package contains custom functions based on ReactJS hooks, providing flexible and powerful functionalities for building the user interface.

- **Assets**: The Assets package stores the images and multimedia resources for the system.
- **Contexts**: The Contexts package manages global states for the entire system, offering a robust mechanism to share data and states across different components.

Constants: The Constants package holds constants and system response messages, helping to manage information and encoding more easily.

3.1.3 Detailed Package

a, Diagram of Cooking Post Interaction Functionality

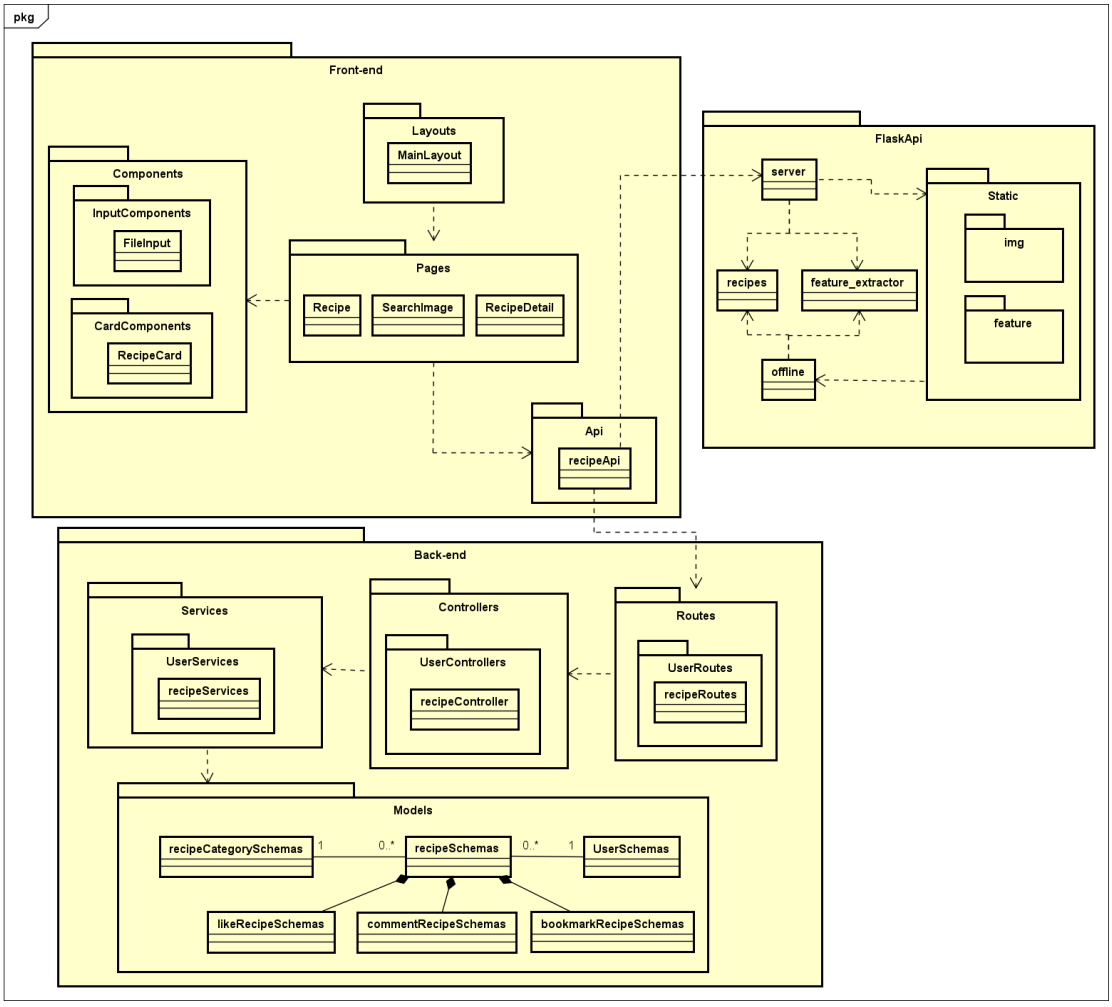


Figure 3.6: Detailed Package Diagram of Cooking Post Interaction Functionality

Based on Diagram 3.6, we can see that the MainLayout creates the layout framework containing the objects in the Pages package. The Pages package is designed to provide the interface for cooking posts, using input components and recipe tags from the Components package. The recipeApi object is responsible for communicating and retrieving data from the server. The server receives and processes the requests through recipeRoute, which contains the API endpoints, and then forwards them to recipeController. recipeController handles the requests before forwarding them to

recipeServices. recipeServices performs data processing tasks, including interacting with models to perform cooking post interactions. On the other hand, the interface is also connected to the Flask API server to perform image search functionality through the recipeApi object.

b, Detailed Package Diagram of Meal Scheduling Functionality

By building similarly, we have the detailed package diagram for meal scheduling functionality.

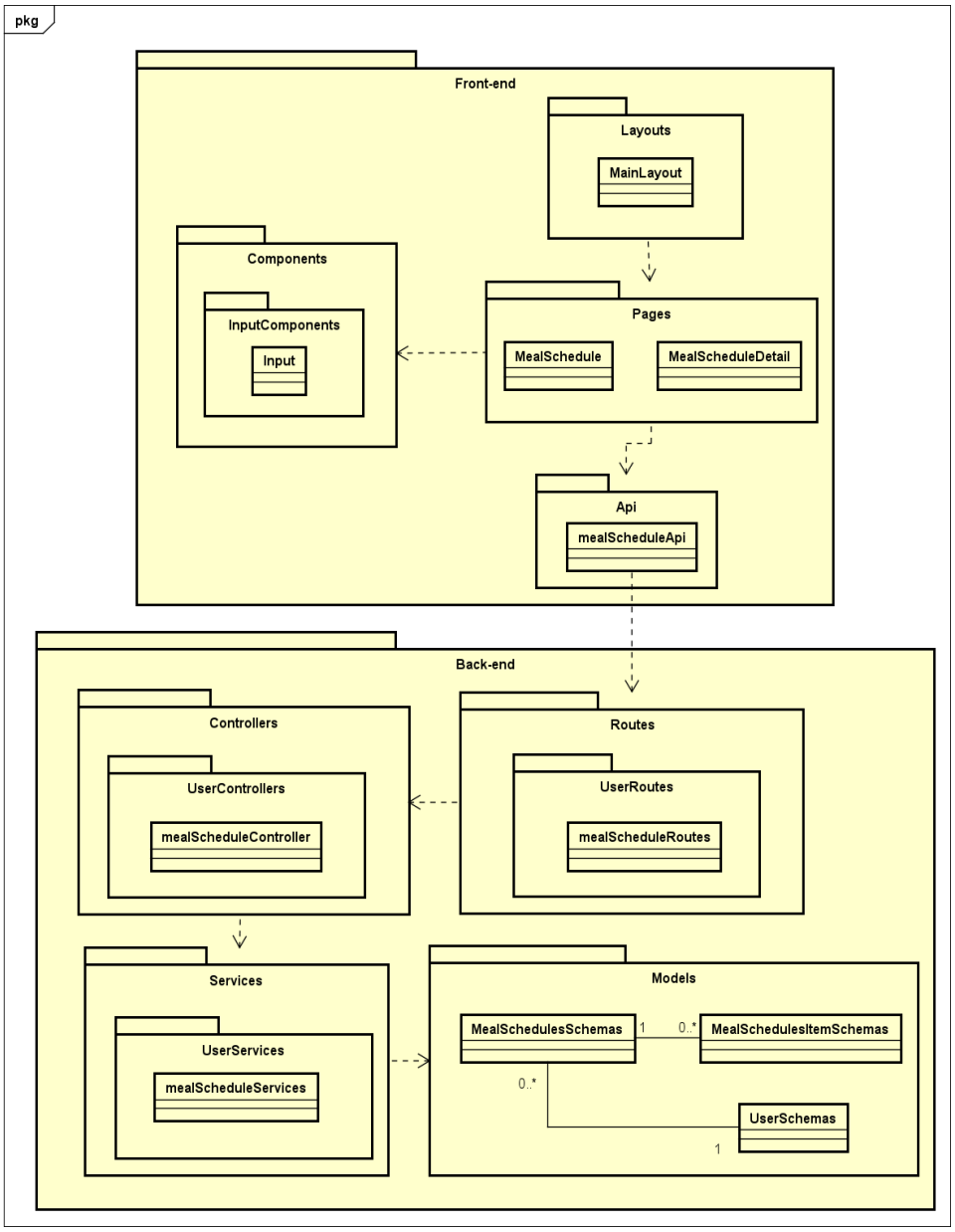


Figure 3.7: Detailed Package Diagram of Meal Scheduling Functionality

c, Detailed Package Diagram of Workout Scheduling Functionality

By following a similar structure, we have the detailed package diagram for the workout scheduling functionality.

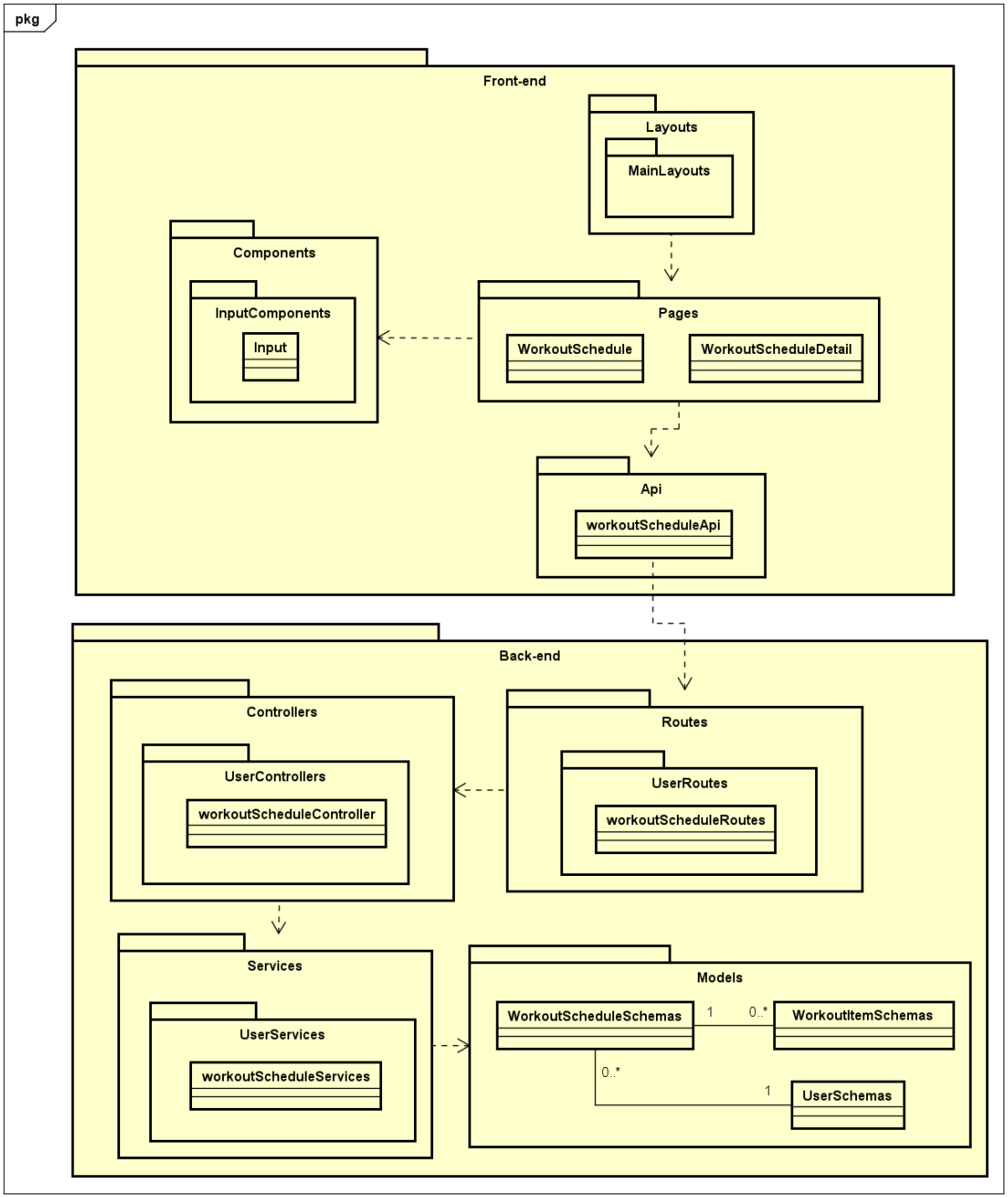


Figure 3.8: Detailed Package Diagram of Workout Scheduling Functionality

3.2 Detailed Design

3.2.1 Class Design

a, Class Diagram for Image Search Functionality

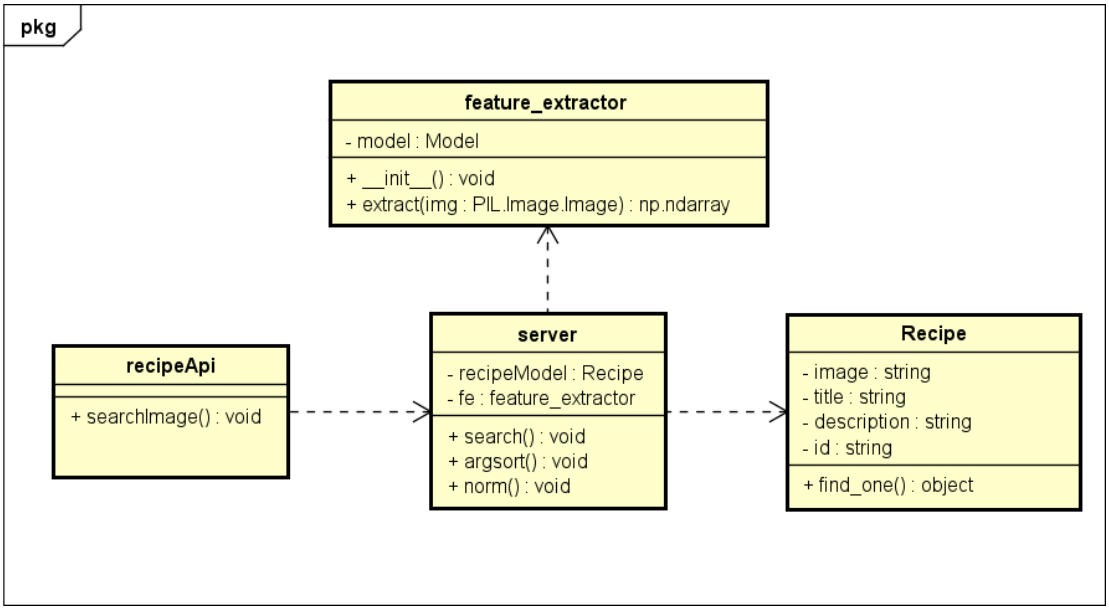


Figure 3.9: Class Diagram for Image Search Functionality

First, the user will upload the image they want to search for via the interface. The interface will call RecipeApi, which is responsible for connecting to the server to send and receive data from the uploaded image. When the server receives the request from RecipeApi, it will call the Feature Extractor class to process the image and extract features from the uploaded image. These features will then be compared and matched with the features of the food images already available in the system. Next, the server will call the Recipe class, which interacts with the database to retrieve a list of recipes that match the extracted features. Finally, the server will return the search results to RecipeApi, and the results will be sent back to the user through the interface.

b, Class Diagram of the Meal Planning Function

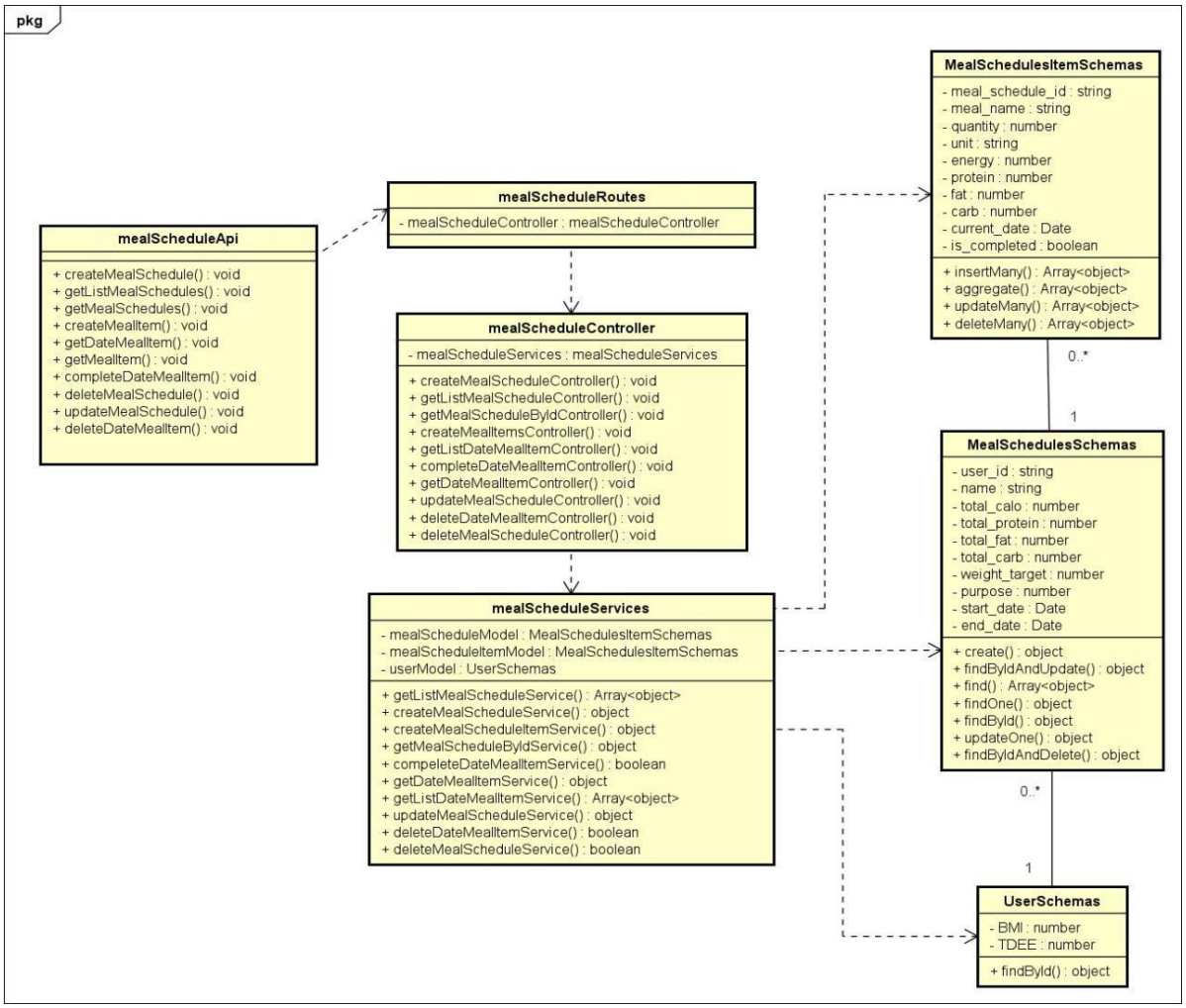


Figure 3.10: Class Diagram of the Meal Planning Function

First, the user interface will interact with the mealScheduleApi through mealScheduleRoutes to send and receive data. MealScheduleRoutes will forward these requests to the mealScheduleController, which acts as a bridge between the view and the model in the MVC pattern. The mealScheduleController will call the methods of mealScheduleServices to process the business logic. MealScheduleServices interacts with objects in the model to perform the user-requested operations. After processing, mealScheduleServices returns the result to the mealScheduleController. Finally, the mealScheduleController will send the result back to the user interface, completing the interaction loop.

c, Class diagram for workout schedule feature:

Similar to the meal scheduling feature, we have the class diagram for the workout scheduling feature.

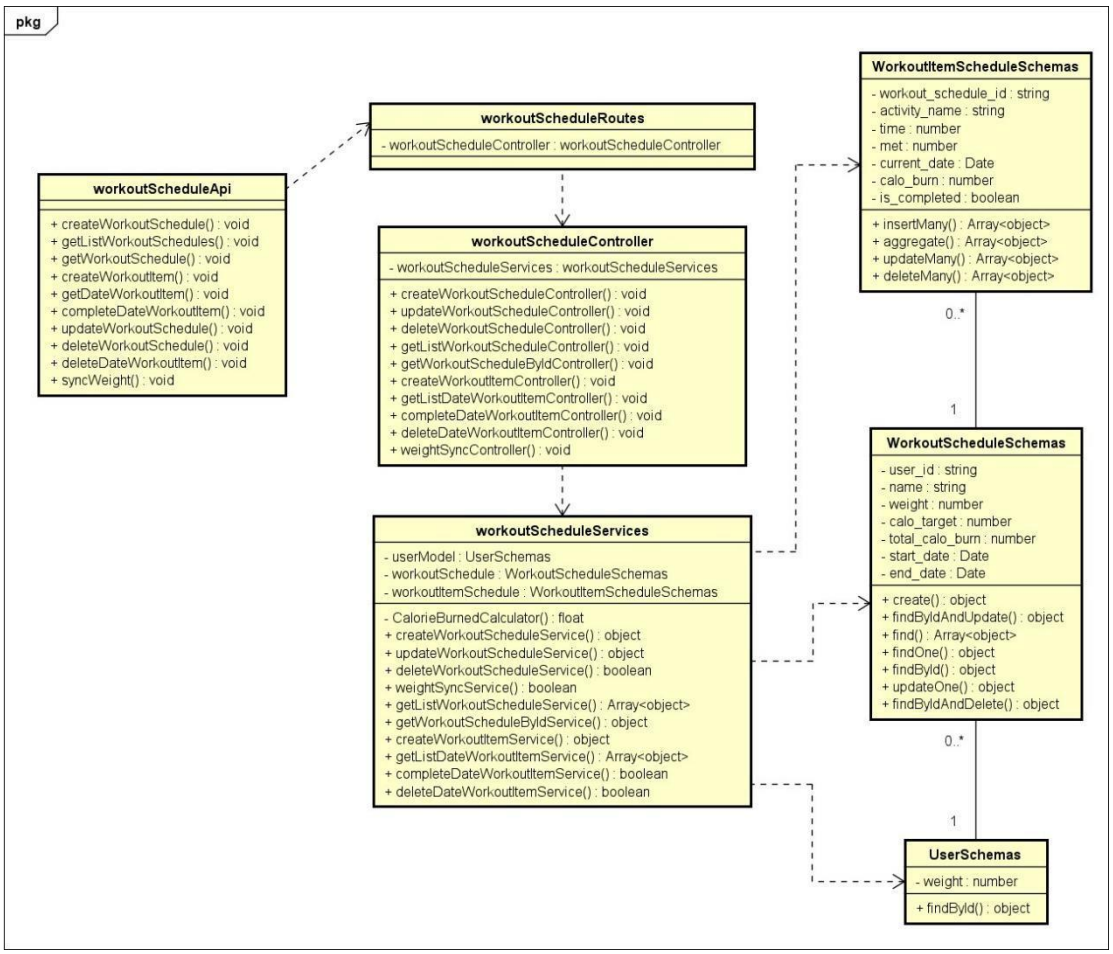


Figure 3.11: Class diagram for the workout schedule feature

Next, to describe the class diagram above, we have the following flow diagrams:

d, Flow diagram for the image search feature.

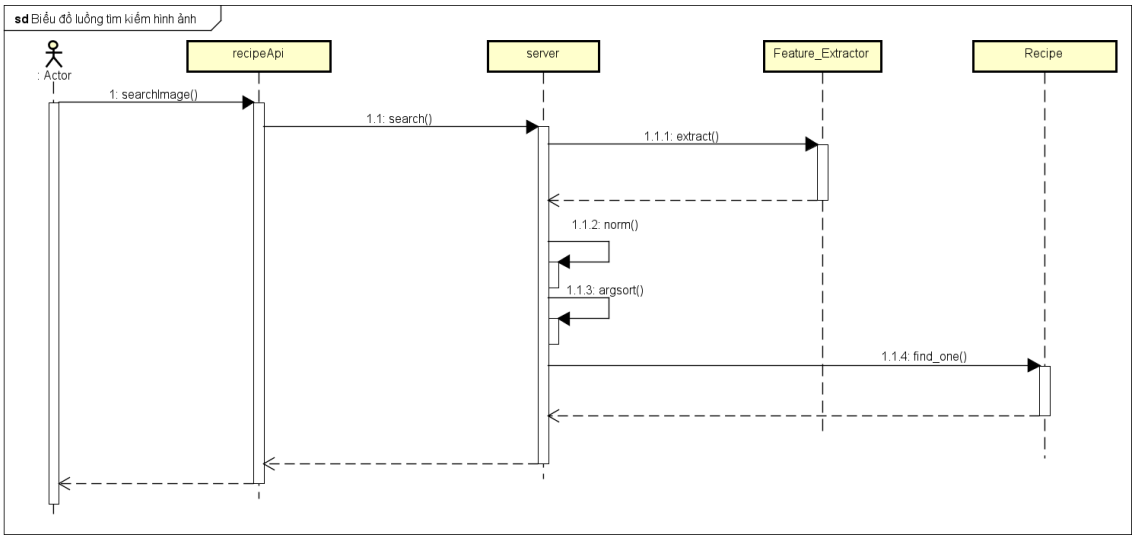


Figure 3.12: Flow diagram for the image search feature

e, Flow diagram for the meal schedule feature.

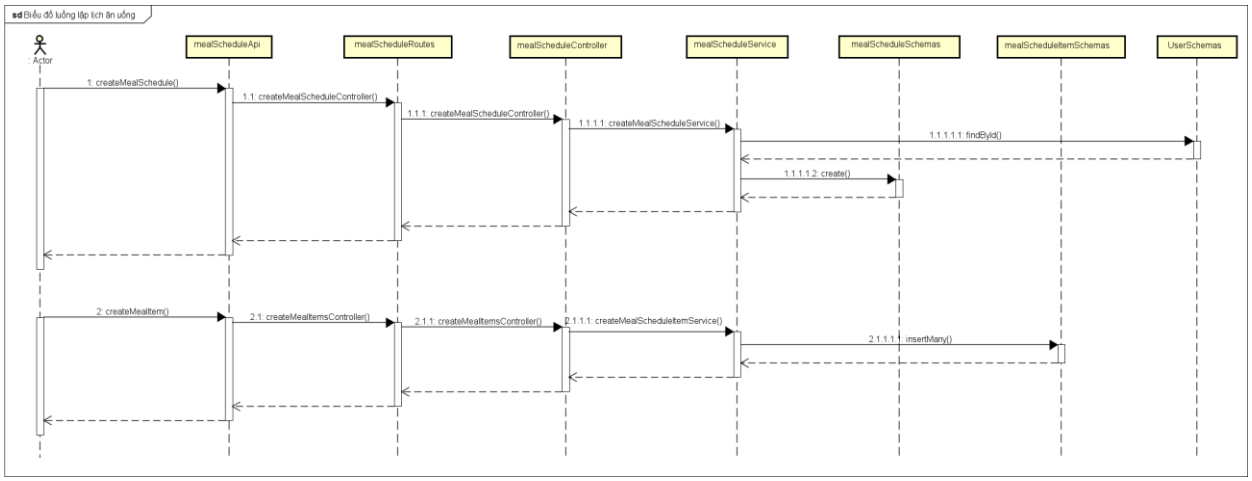


Figure 3.13: Flow diagram for the meal schedule feature.

f, Flow diagram for the workout schedule feature.

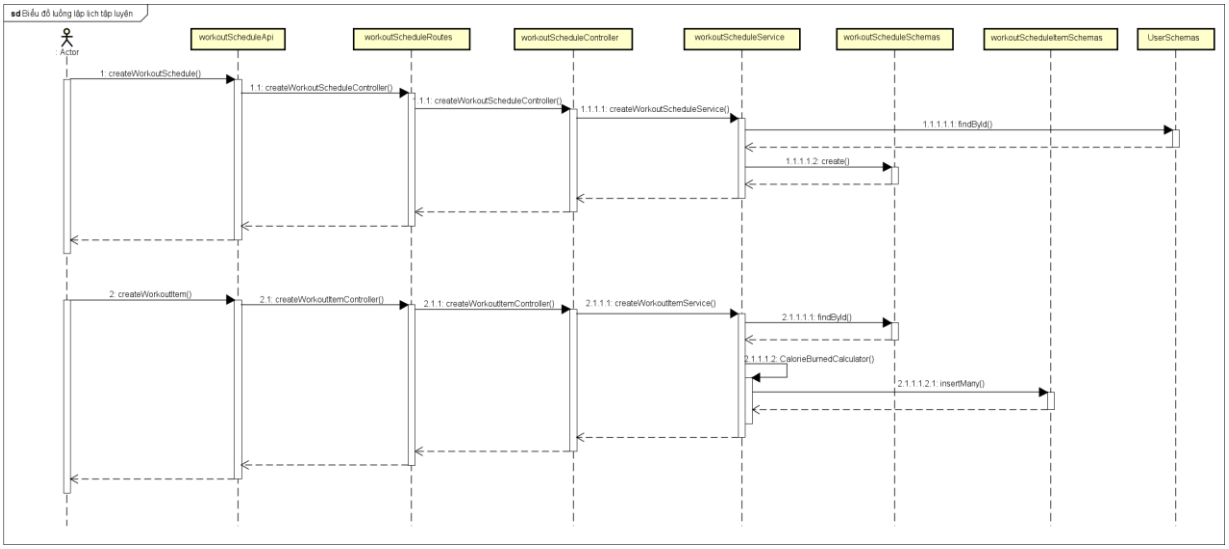


Figure 3.14: Flow diagram for the workout schedule feature.

3.2.2 Design Database

In the project, the system uses MongoDB as the database to store data. The design of MongoDB allows for more freedom and creativity, as it is not constrained by formal procedures, rigid algorithms, or specific rules. Instead of adhering to strict processes, we can freely create a flexible and dynamic data model. MongoDB does not impose structural barriers, allowing data to be adjusted according to the needs of the application naturally. On the other hand, in MongoDB, data is organized as flexible JSON documents. This object-oriented organization makes querying, scaling, and updating data much easier.

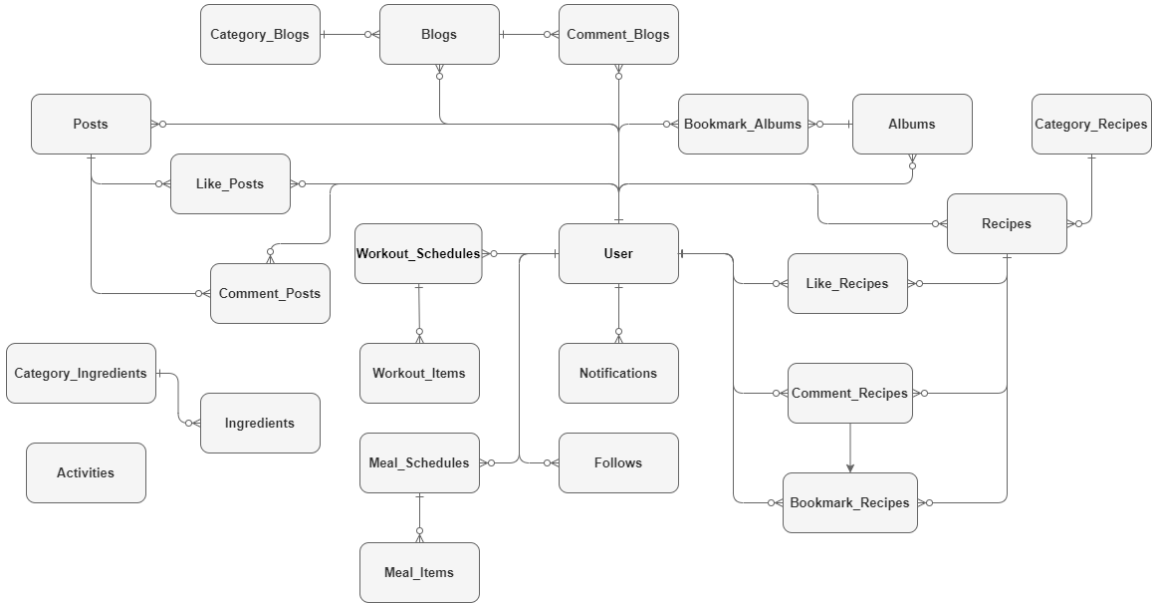
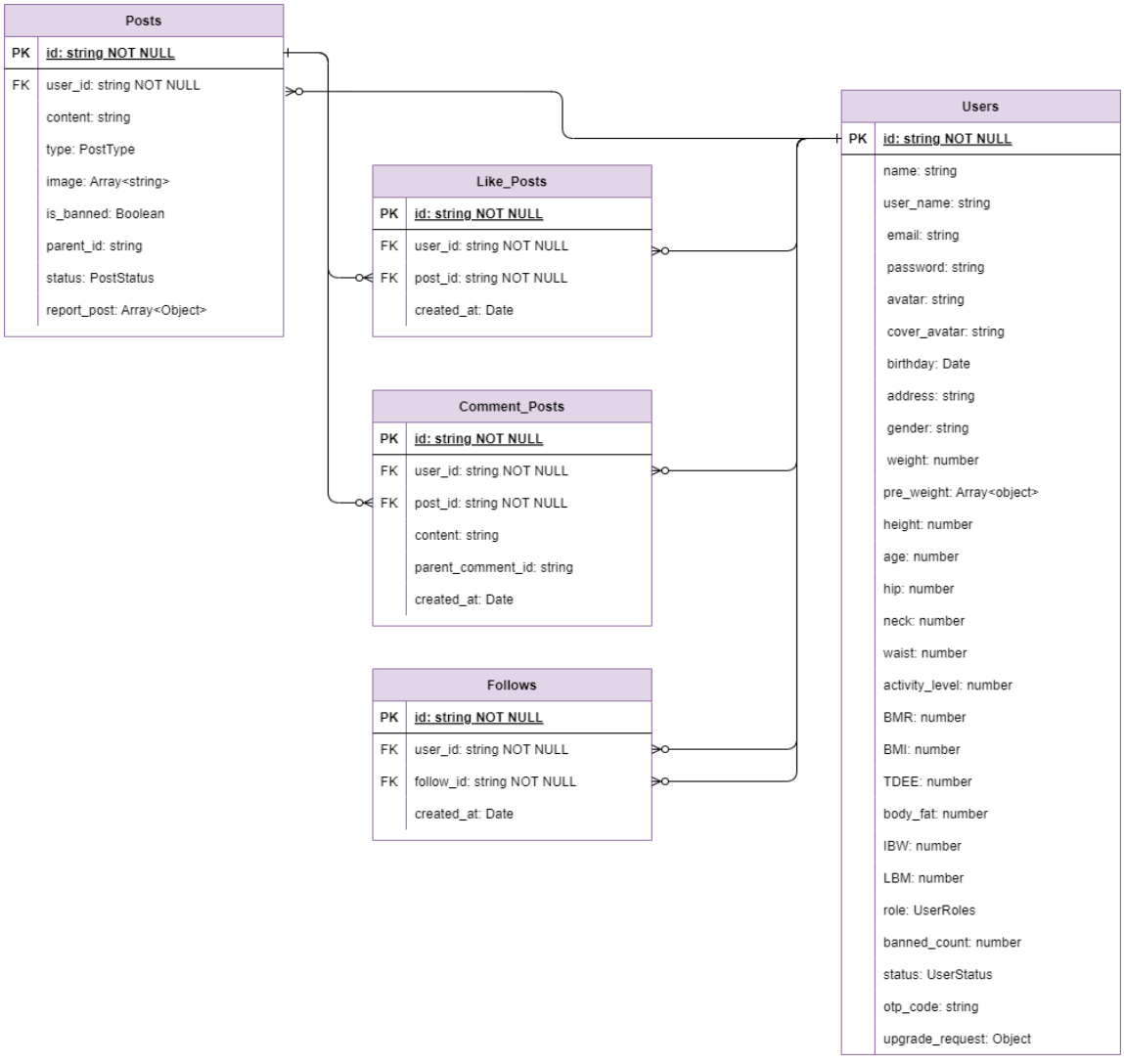
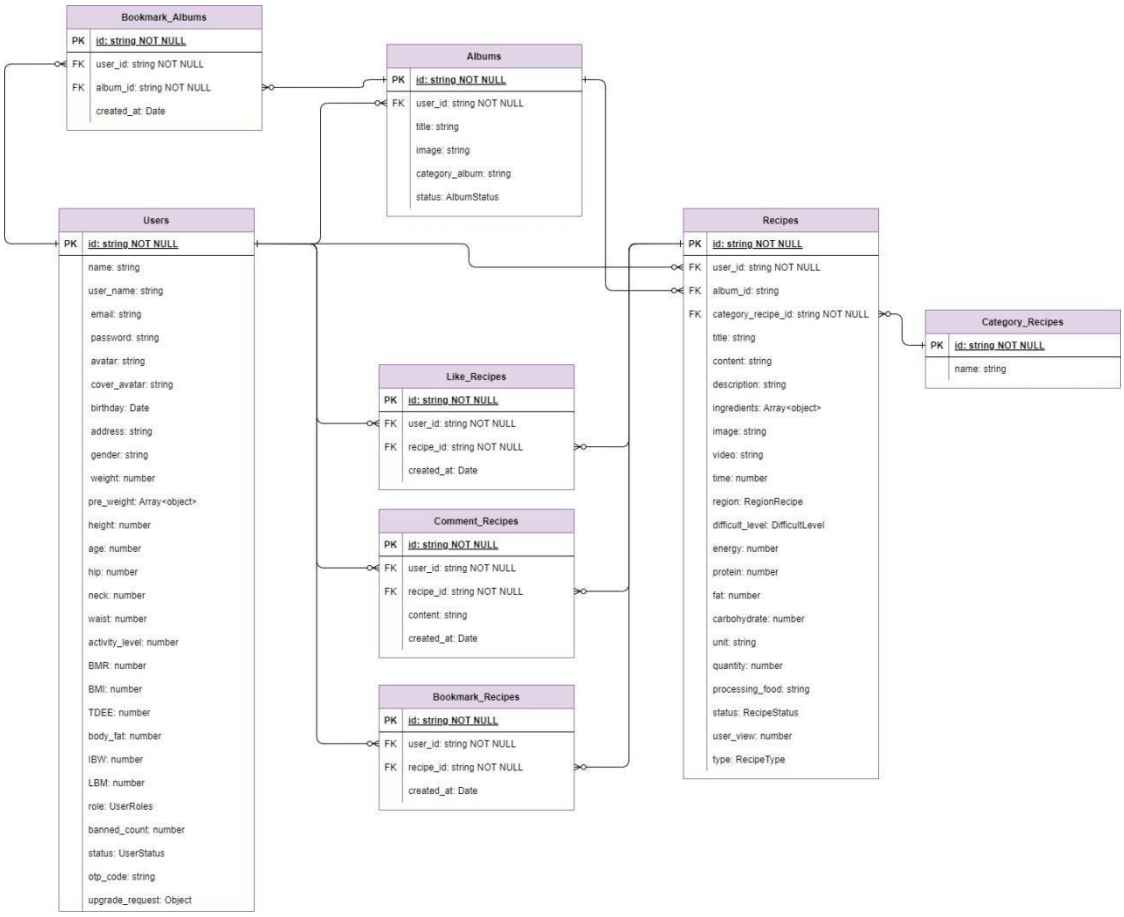


Figure 3.15: Overview of the Database Diagram

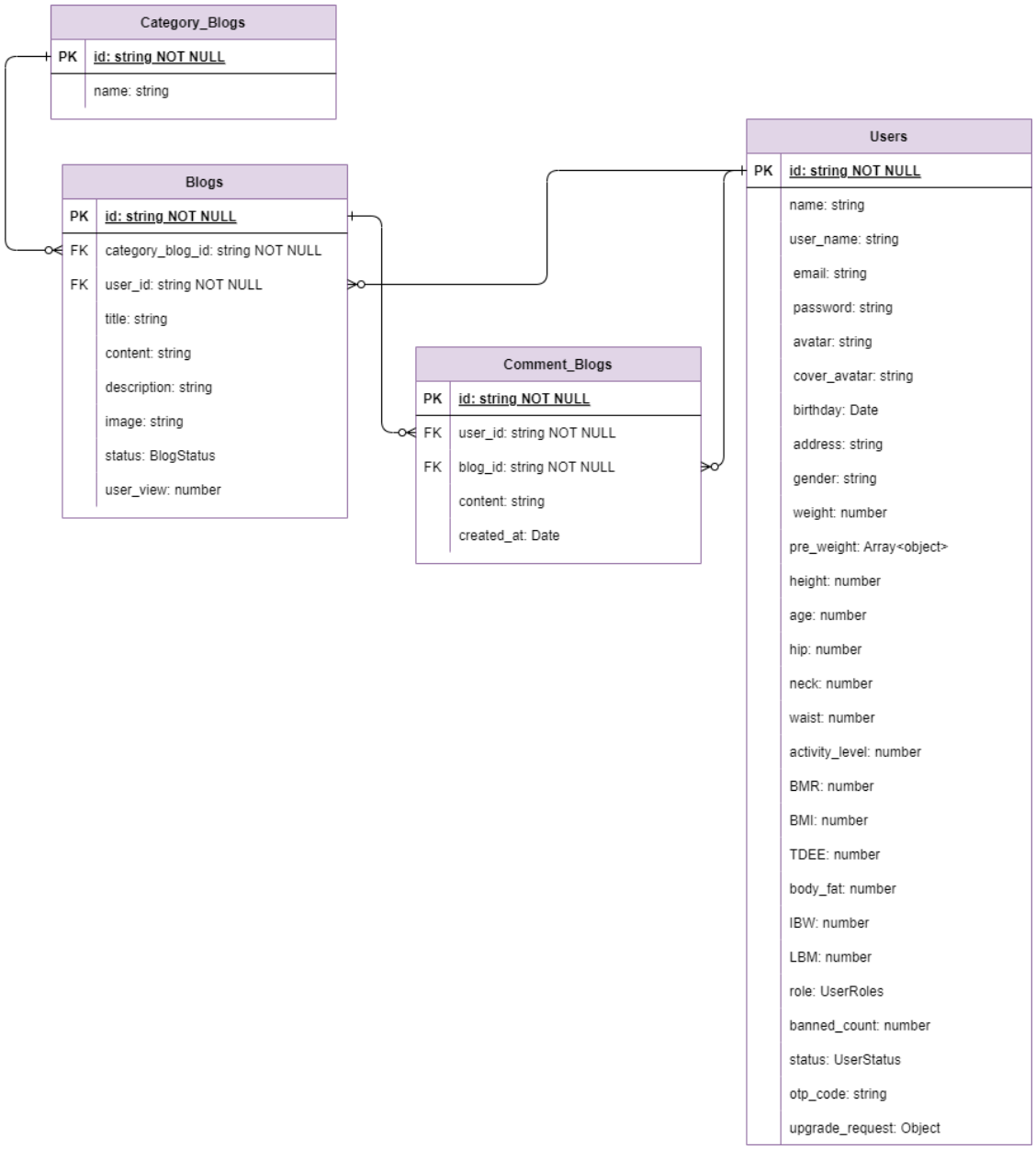
a, Database Cluster Diagram for Forum Posts, Questions, and User-to-User Interactions



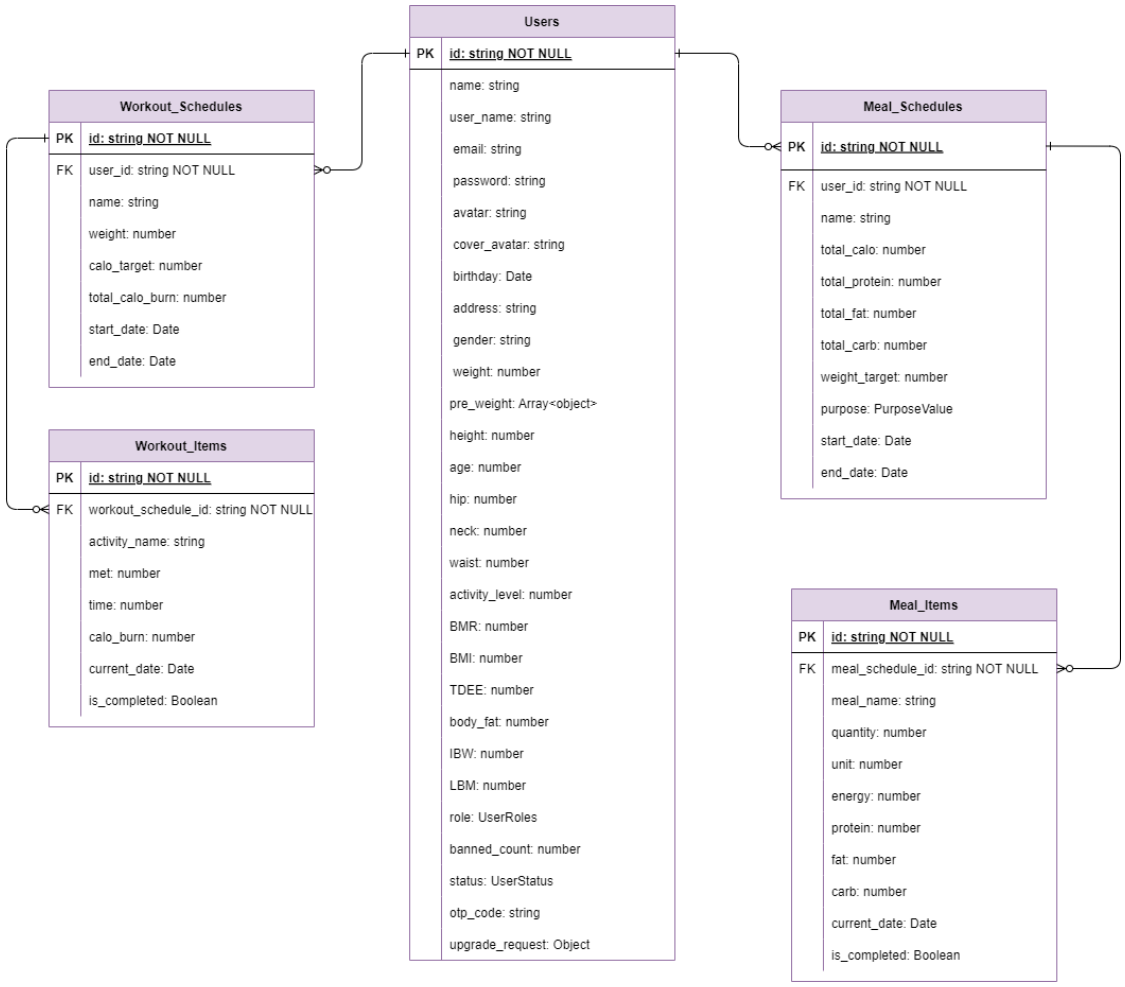
b, Database Cluster Diagram for Cooking Posts and Cooking Albums



c, Database Cluster Diagram for Blog

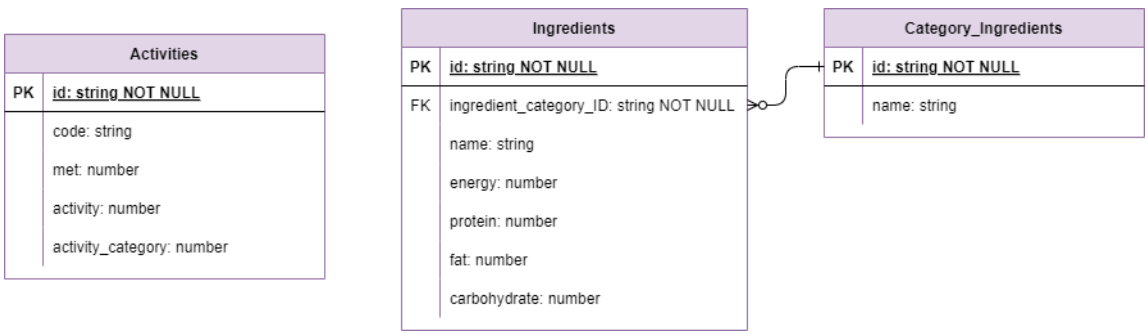


d, Database Cluster Diagram for Meal Schedule and Workout Schedule



e, Diagram of Supporting Tables for Storing Collected Data

These tables include nutritional data of ingredients and activities with MET values, which support functionalities such as managing cooking recipes with nutritional information, viewing nutritional indicators, meal scheduling, and workout scheduling.



a, Forum Post Interaction Interface Design

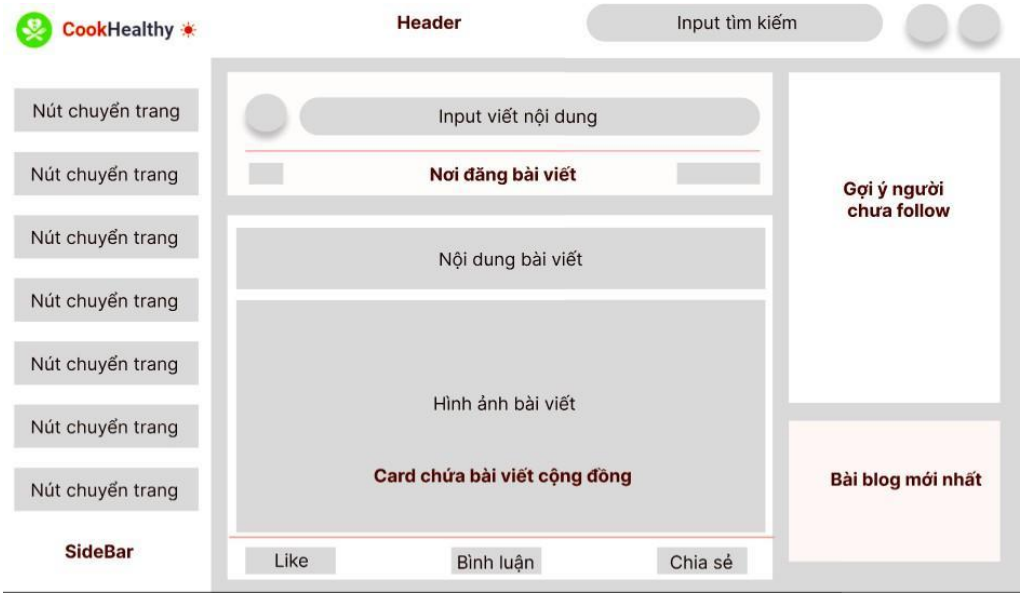


Figure 3.21: Forum Post Interaction Interface Design

According to the design in Figure 3.21, users can create posts, view content of posts, like, share, or comment on forum posts.

b, Cooking Recipe Post Interaction Interface Design

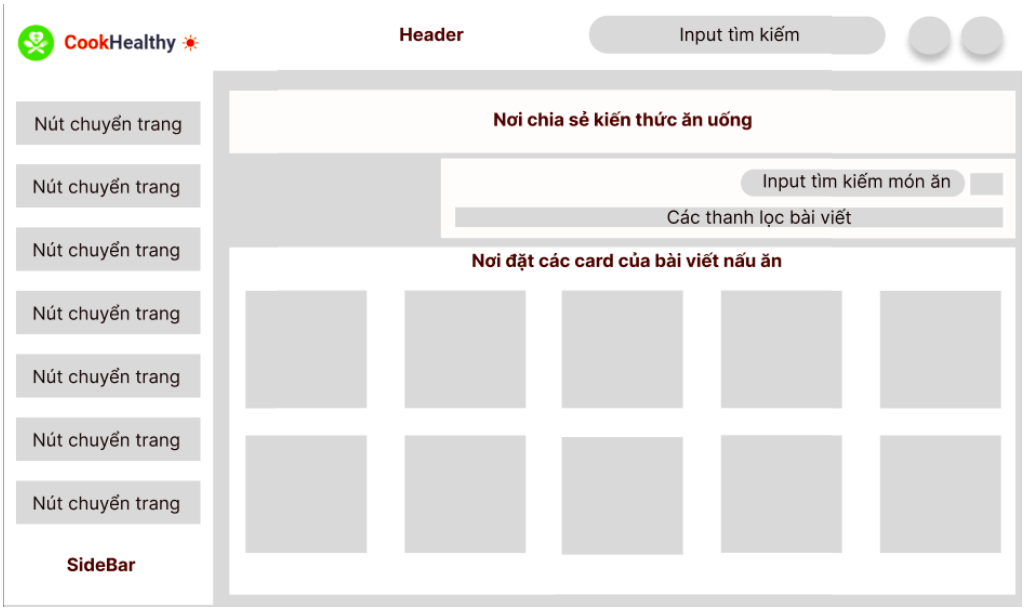


Figure 3.22: Cooking Recipe Post Interaction Interface Design

According to the design in Figure 3.22, users can explore, search, and view cooking recipe posts within the system.

CHAPTER 4: SYSTEM IMPLEMENTATION AND EVALUATION

4.1 Technologies Used

To create the web system for sharing nutrition information and recipes, the following technologies were used: ReactJS and Tailwind CSS for the user interface, NodeJS-ExpressJS and Flask for the system, Socket.io for notifications, MongoDB for the database, AWS for image storage and email sending

These technologies were integrated flexibly and efficiently to create a complete web system, providing the best user experience for sharing and exploring information about nutrition and cuisine.

4.1.1 NodeJS - ExpressJS

NodeJS is a platform developed in 2009 by American engineer Ryan Dahl. It is an execution environment written in C++ based on Chrome's V8 JavaScript engine. With an asynchronous model and non-blocking I/O, NodeJS can handle thousands of simultaneous connections without freezing the thread. Additionally, NodeJS is a unified runtime environment that allows developers to use JavaScript both on the server side and the browser, reducing fragmentation in application development.

ExpressJS is a framework for NodeJS, used to develop applications, websites, and APIs quickly and easily. ExpressJS provides a simple, easy-to-understand syntax but is a high-performance platform that allows rapid processing of web application requests while offering good scalability and flexibility.

Based on the above analysis, we can conclude that NodeJS - ExpressJS fully meets the requirements and is highly suitable for the web system of this thesis. Therefore, NodeJS - ExpressJS was chosen to develop the server-side for the system.

4.1.2 ReactJS

ReactJS is a JavaScript library launched in 2013 and developed and supported by the tech giant Facebook. ReactJS is used to build interactive components on a web system, primarily helping developers create user interfaces. The choice of ReactJS for developing the client side of this project emphasizes the following key advantages it offers:

ReactJS does not only limit rendering data on the server side but also extends this capability to the client side. This results in a smoother and faster user experience, with the ability to display data without the need to reload the entire page.

With its state management mechanism, ReactJS helps developers easily track and control the application's state efficiently. This optimizes the development and maintenance process of the application.

The ability to break down the interface into components is one of the strengths of ReactJS. This not only helps developers manage and scale the codebase more easily but also enables flexible and efficient code reuse.

Finally, ReactJS provides excellent performance, allowing users to experience seamless transitions between web pages quickly, without waiting for a full page reload. This brings convenience and comfort to users while using the application.

4.1.3 MongoDB

MongoDB was first introduced in 2009 and developed by MongoDB, Inc. It is a document-oriented database, a type of NoSQL database, designed to process and store data in a flexible and efficient manner, especially for complex or frequently changing data structures.

In MongoDB, data is organized and stored in documents, a flexible approach that more naturally reflects the traditional way relational database management systems (DBMS) handle data. This data structure uses BSON (Binary JSON), an extended version of JSON, to support a wide variety of data types such as dates, floating-point numbers, and binary data.

Unlike traditional relational DBMS, MongoDB does not require rigid relationships or a fixed schema upfront. Instead, it offers flexibility for applications by removing these constraints, while providing excellent query performance and horizontal scalability.

Therefore, MongoDB is an ideal choice for building a flexible, powerful, and scalable database to meet the needs of developing the web system for this project.

4.1.4 Socket.io

Socket.io is a powerful library for JavaScript, designed to create real-time web applications. Released in 2010 by Guillermo Rauch, Socket.io quickly became an essential tool for developing applications that require continuous interaction and low latency between the client and the server.

Socket.io operates based on the WebSocket protocol, which allows bidirectional communication between the client and server, enabling instant data transfer without the need to reload the webpage. Additionally, Socket.io supports fallback protocols like polling to ensure broad compatibility across different browsers. This helps maintain a stable connection even in unstable network environments.

One of Socket.io's key strengths is its ability to handle events in a flexible way. Developers can easily define and manage custom events to transmit information between the client and the server. Socket.io does not only focus on connecting and transferring data but also integrates robust security features and scalability capabilities. With user authentication mechanisms and load-balancing support, Socket.io ensures that applications can scale to serve a large number of users while maintaining high performance.

For this reason, Socket.io is a powerful tool for building notification features for users in the system. Thanks to its ability to transmit data instantly and maintain stable connections, Socket.io enhances real-time user interaction. This not only improves the user experience but also makes the system more flexible and efficient in providing timely information.

4.2 Building the Application

4.2.1 Achieved Results

Source Code Structure:

The system is divided into two main parts: the backend and the frontend, with each part responsible for distinct tasks. The backend is further divided into two parts: the **Node.js backend**, which handles user requests and business logic of the system, and the **Flask backend**, which is responsible for the image search functionality. The frontend is also divided into two parts: the **user interface** and the **admin interface**. Each part of the frontend is responsible for building the interface, allowing users to interact and send requests to the server. This division helps the system maintain clear responsibility and optimize work performance.

System Actors:

The system is divided into two types of users and three types of administrators:

User System:

- **Users** are the primary actors interacting with the system. They can post questions, articles on the forum, view cooking guides, nutrition albums, and cooking tip blogs. They can also interact, follow others, and view personal health metrics to plan their diet and exercise routines.
- **Chefs** are created when a user sends a request to the system after reaching 1,000 followers. If they don’t meet the requirement, they can submit a request to upgrade their account along with proof of their experience to become a chef. Chefs will have access to new functions, including managing cooking posts

without nutritional information, food albums, and blogs they have created.

Management System:

- **Administrators** have full rights within the system. Their main task is to upgrade user accounts to chef status and manage all system accounts.
- **Moderators** handle reported posts, manage chef posts, and have the same functions as post authors.
- **Post Authors** create cooking posts with nutritional information and manage dish ingredients.

By dividing the backend and frontend and implementing role-based access control, the system not only helps clarify responsibilities but also strengthens security by controlling access to different sections based on user roles. This ensures that each component of the system operates within its assigned scope and prevents unauthorized access to data. Additionally, managing permissions based on user groups makes it easier to adjust and scale the system without causing confusion or security risks.

4.2.2 Illustrating Main Functions:

a, Post Creation Interface for Articles and Questions on the Forum

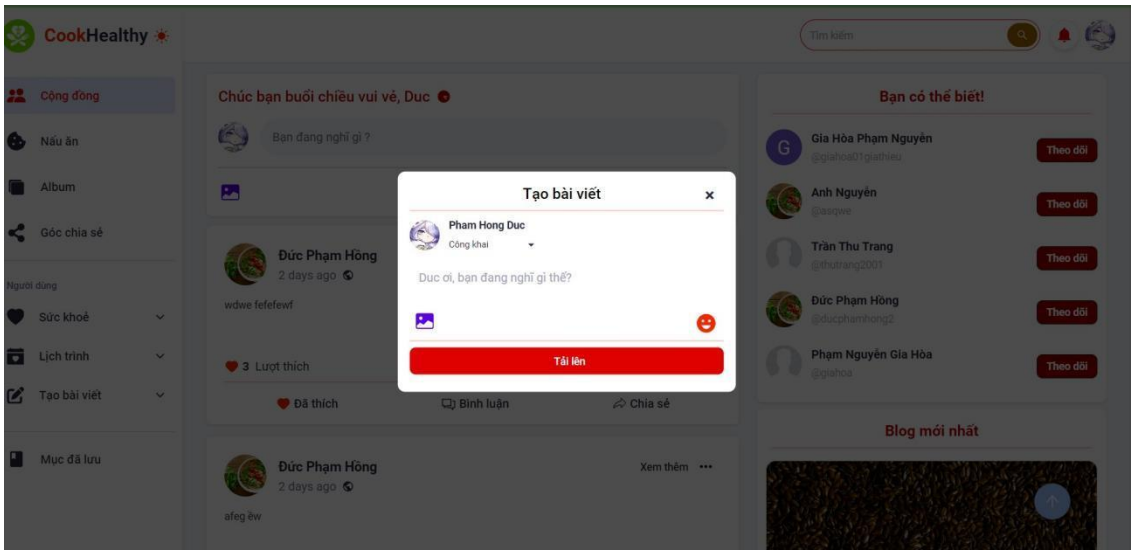


Figure 4.1: Post and Question Creation Interface on the Forum

When looking at the interface, users can post articles either as text or with images to share their experiences. They are also provided with flexible options such as posting publicly**,** sharing with followers only, or making the post visible only to themselves. This helps them control the privacy of the information they share on the platform, creating a flexible and safe environment for interaction and sharing experiences.

b, Forum Post Interaction Interface

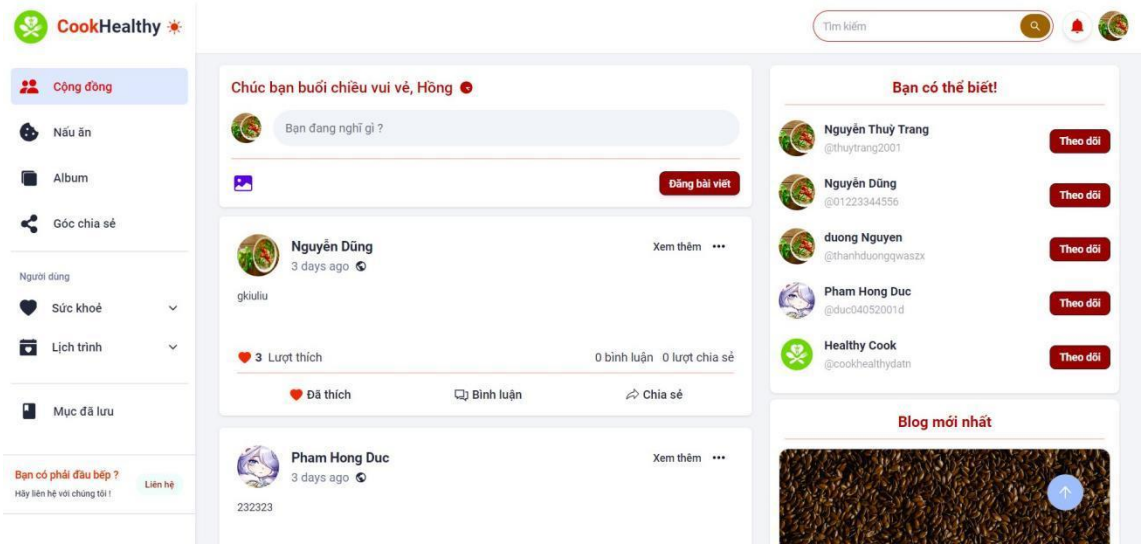


Figure 4.2: Forum Post Interaction Interface

Users can view the posts and questions submitted by forum members. They have the option to like, share, and comment on these posts. Additionally, two sections next to the posts suggest users that the viewer hasn't followed yet, as well as the latest news of the day, helping users easily expand their network and stay updated with the latest information.

c, Cooking Post Interaction Interface

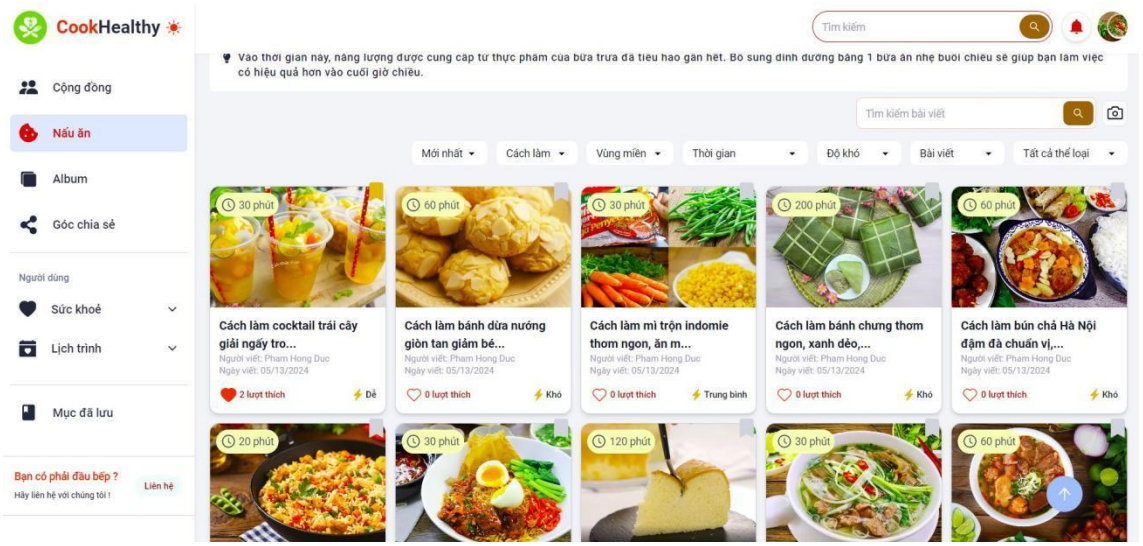


Figure 4.3: Cooking Post Interaction Interface

According to the design, users can explore cooking posts on the system. They can express their likes, save posts to their favorites, and search and filter content according to their needs. Notably, users can also utilize the image search feature through the icon next to the search bar, making the search process more intuitive and convenient.

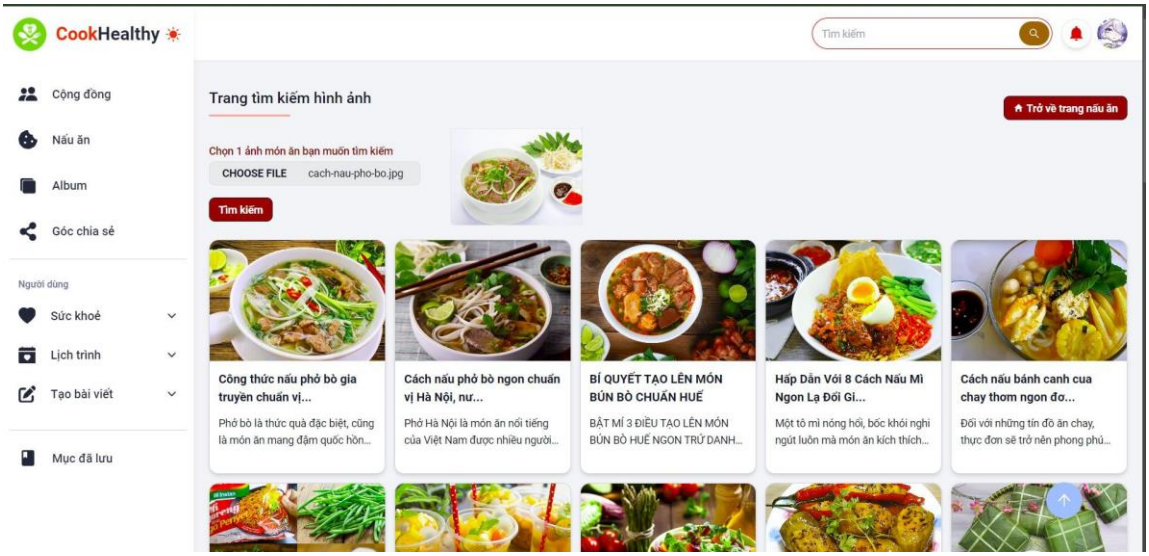


Figure 4.4: Image Search Interface

d, Health Metrics View Interface

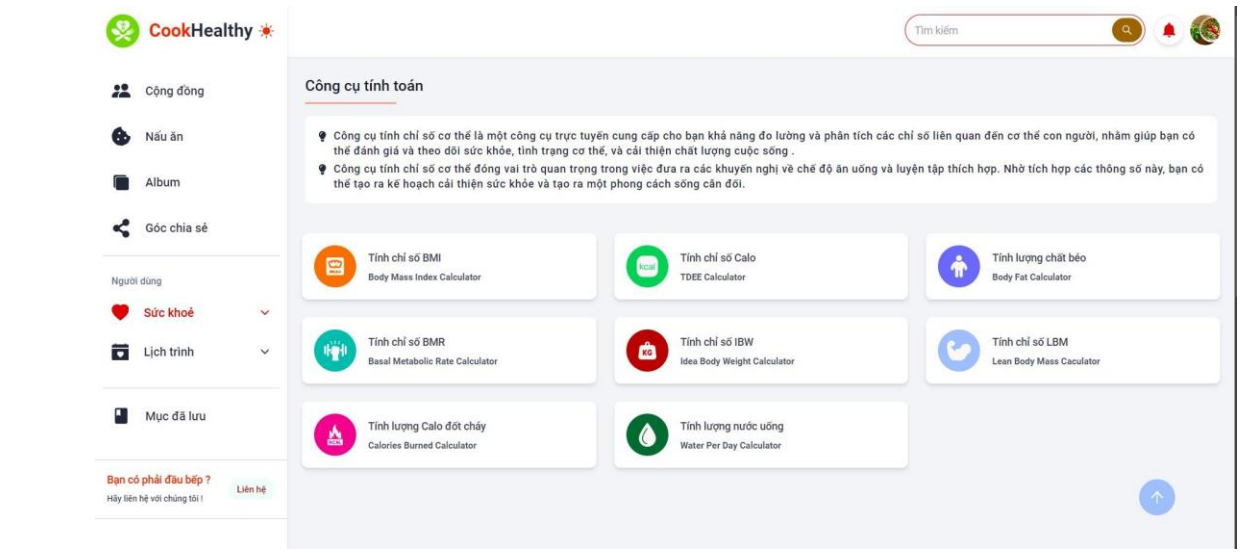


Figure 4.5: Health Metrics View Interface

Users can view and input their body metrics, allowing the system to calculate important health indicators such as BMI (Body Mass Index), TDEE (Total Daily Energy Expenditure), and other metrics. This information helps users better understand their health status and make adjustments to their diet and physical activity in a more scientific and effective way. Additionally, the system can provide suggestions and advice based on these metrics, helping users maintain a healthy and balanced lifestyle.

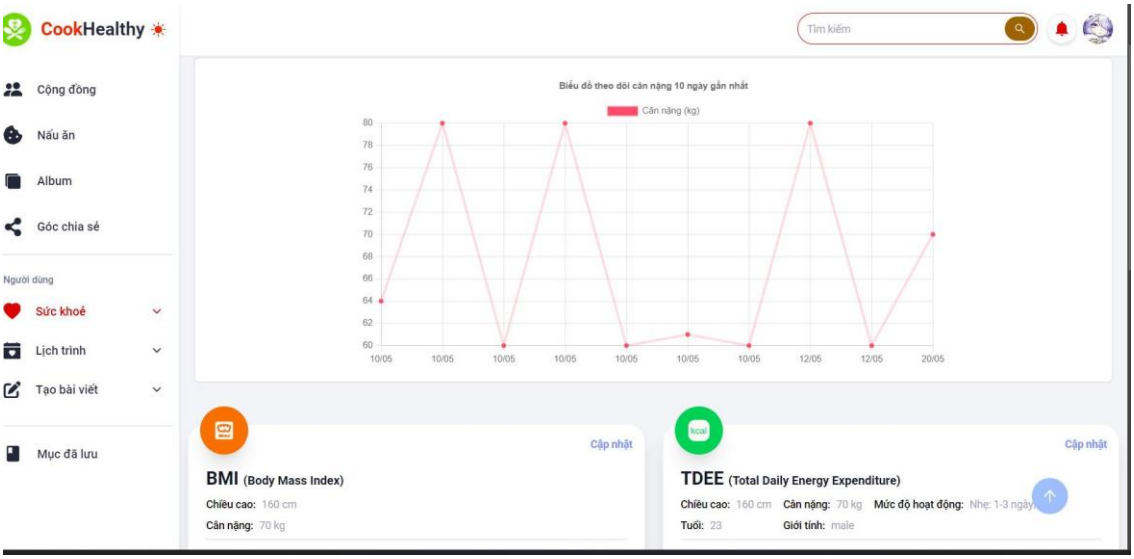


Figure 4.6: Interface for Viewing Calculated Metrics from the System

e, Meal Planning Interface

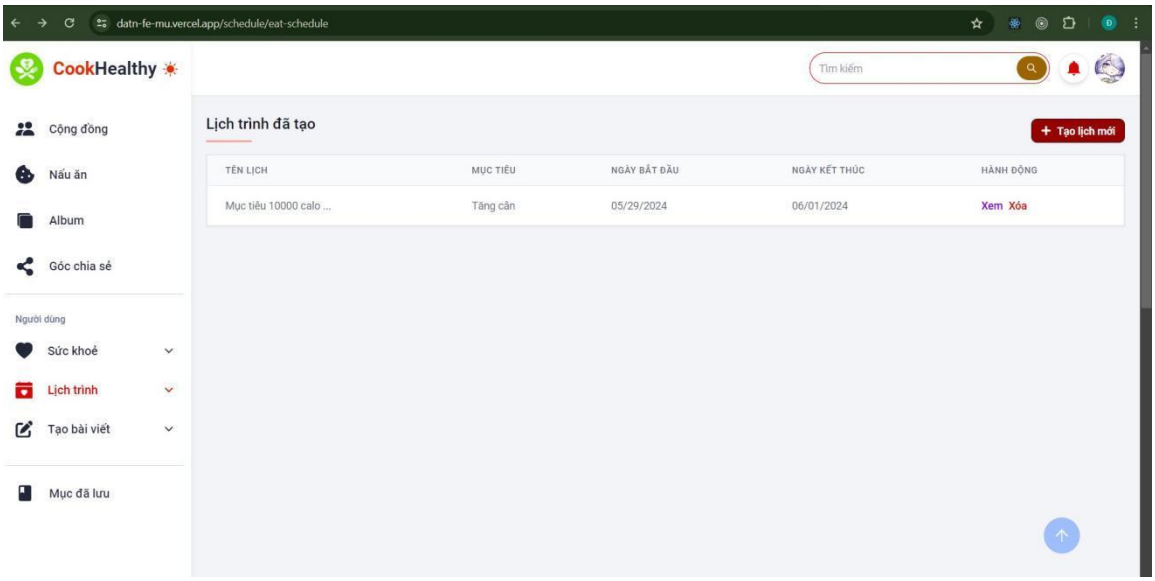


Figure 4.7: Meal Planning Interface

Looking at the diagram, we can see that users have the ability to create and manage their personal meal schedules. They can view the details of each item in the meal plan. Additionally, they can easily delete unnecessary items or modify the schedule according to their needs.

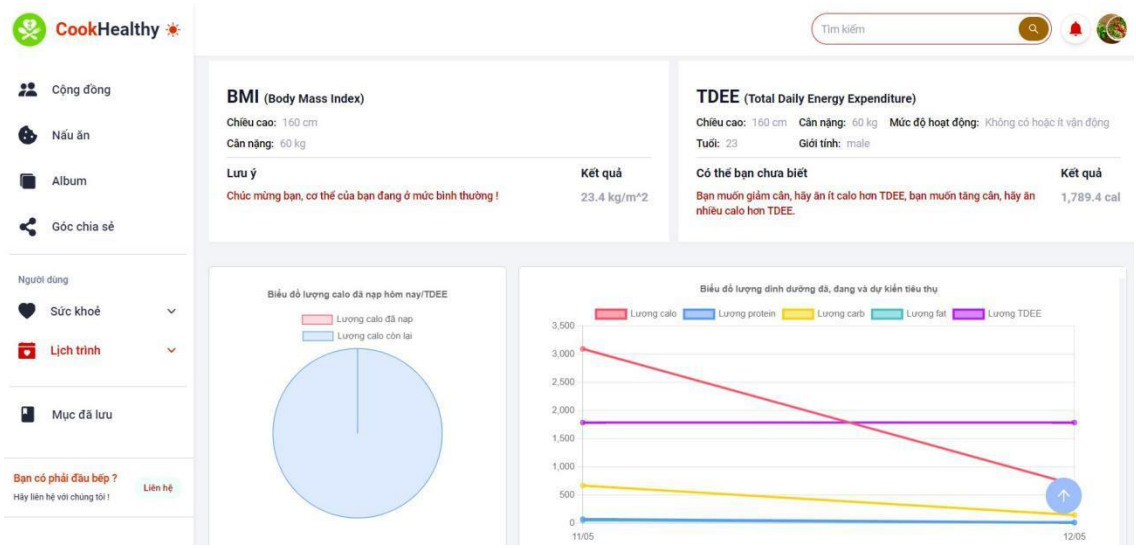


Figure 4.8: Detailed Meal Planning Interface

Looking at the detailed interface, users can see that the system calculates and displays a visual chart of the nutrients they have consumed, including calories, protein, carbohydrates, and other nutrients. Through this chart, users can easily track and assess their nutritional intake, helping them adjust their diet in a more scientific and balanced way. Additionally, having detailed nutritional information illustrated through the chart will help users better understand the factors influencing their health and raise awareness about maintaining personal health.

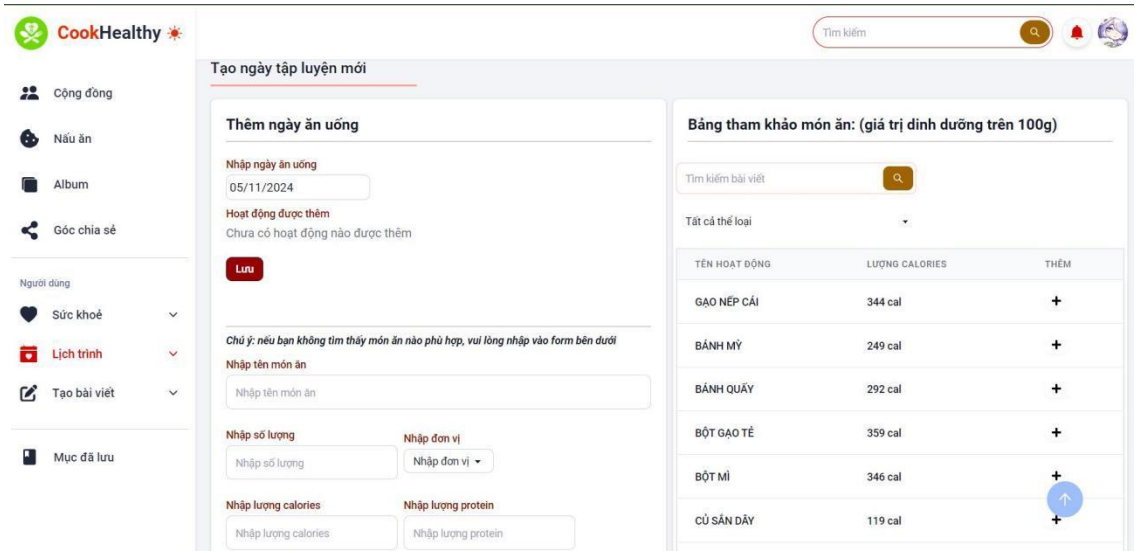


Figure 4.9: Meal Event Addition Interface

Users have the ability to add their meal events directly to the interface. This allows them to record and manage information about the food they consume. Adding these meal events helps users track and evaluate their diet in a more detailed and scientific way, enabling them to adjust and improve their lifestyle and health effectively.

f, Workout Schedule Interface

Similar to the meal schedule screen, users can also access and use the workout schedule screens on the system. These screens allow users to create and manage events related to their workouts and physical activities during specific times. They can record workout sessions, the type of activity, duration, and intensity level to track and assess their training progress.

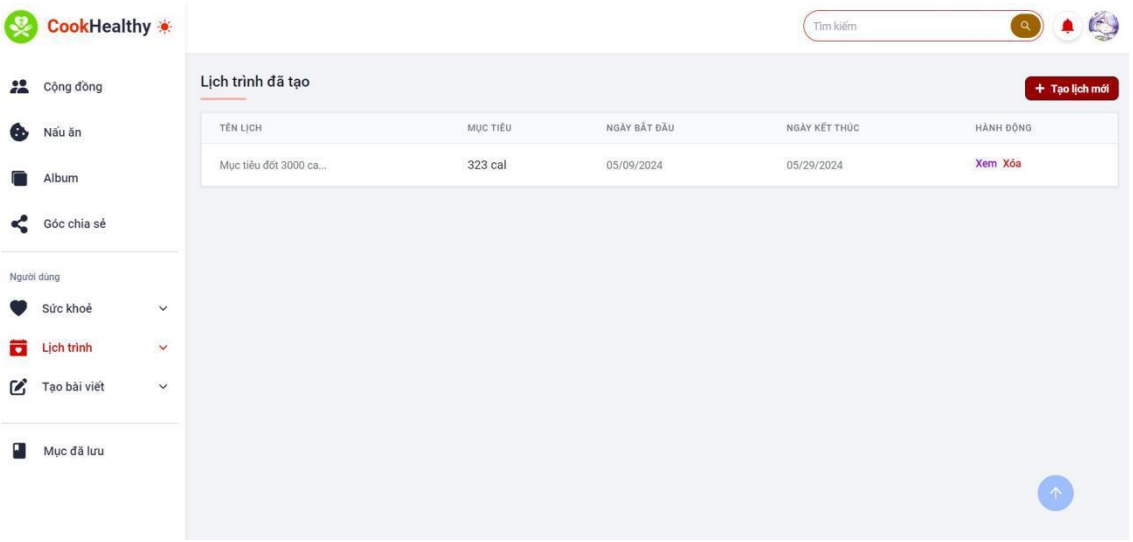


Figure 4.10: Workout Schedule Interface

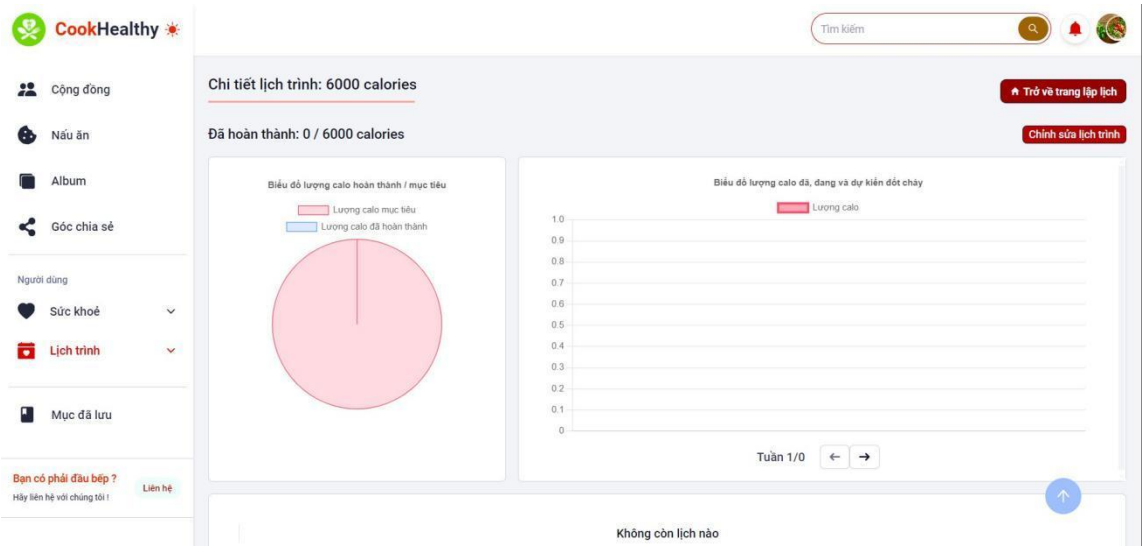


Figure 4.11: Detailed Workout Schedule Interface

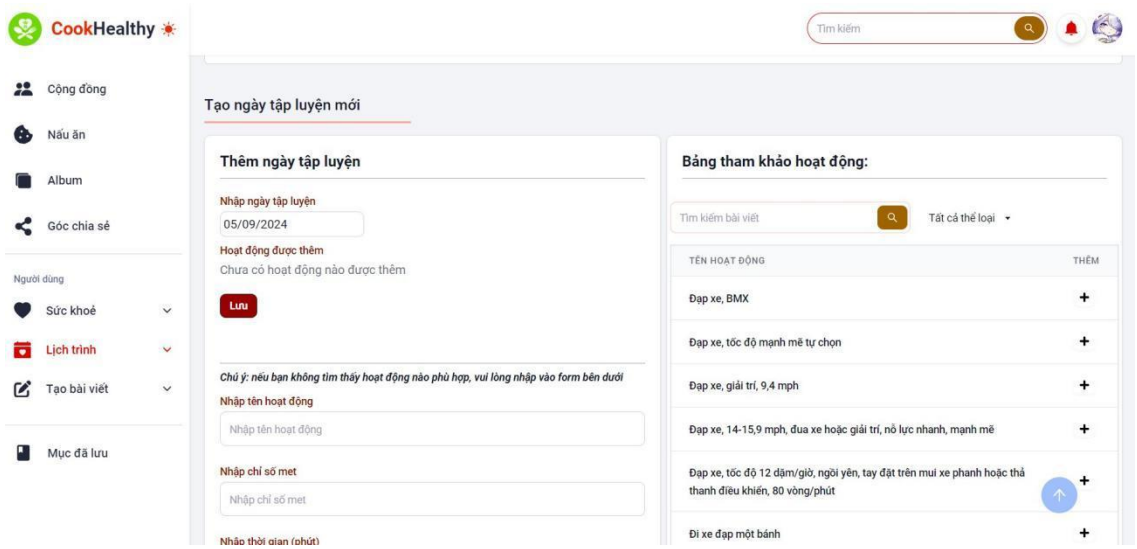


Figure 4.12: Workout Event Addition Interface

CHAPTER 5. CONCLUSION AND DEVELOPMENT DIRECTIONS

5.1 Notable Solutions and Contributions

5.1.1 Understanding the Requirements for Calculating Health Metrics for Users

a, Problem Statement

Traditional systems often fail to effectively integrate cooking and health management, making it difficult for users who are interested in both aspects. To address this issue, the web system for sharing nutritional information and cooking recipes has been developed. This system not only provides a rich and diverse collection of recipes but also ensures nutritional standards, allowing users to learn and apply them easily.

In particular, the system is designed to enable users to manage their personal health simply and accurately. Manual calculation of health metrics often leads to errors and inaccuracies, making health management challenging. This new system completely overcomes these limitations, providing accurate and convenient tools for users to monitor and maintain their health in a scientific and effective manner.

b, Solution

To address the above issue, I will present some formulas for calculating user health metrics that I have collected.

Calculating Body Mass Index (BMI)

The Body Mass Index (BMI) was developed by Belgian mathematician Adolphe Quetelet in the late 19th century. It is a measure used to assess an individual's weight relative to their height. The BMI is calculated by dividing a person's weight (in kilograms) by the square of their height (in meters). The formula for calculating BMI is as follows.

$$\text{BMI} = \frac{\text{weight (kg)}}{\text{height}^2\,(\text{m}^2)}$$

In which:

- **weight** is the person's weight (in kg).
- **height** is the person's height (in meters).

The resulting BMI is typically classified as follows:

- **Below 18.5**: Underweight
- **18.5 to 24.9**: Normal weight
- **25 to 29.9**: Overweight
- **30 or above**: Obesity

The **Basal Metabolic Rate (BMR)** is the rate at which the body requires energy to maintain basic functions like body temperature, heart, lung, and organ functions while at complete rest. It represents the number of calories needed to maintain the body's basic functions without considering physical activity.

There are many formulas for calculating BMR, but we will focus on the **Mifflin-St Jeor equation**, which is widely used. The formulas are as follows:

$$\text{BMR (Male)} = (9.99 \times \text{Weight (kg)}) + (6.25 \times \text{Height (cm)}) - (4.92 \times \text{Age}) + 5$$

$$\text{BMR (Female)} = (9.99 \times \text{Weight (kg)}) + (6.25 \times \text{Height (cm)}) - (4.92 \times \text{Age}) - 161$$

In this formula:

- **weight** is the person's weight (kg),
- **height** is the person's height (cm),
- **age** is the person's age.

Calculating Total Daily Energy Expenditure (TDEE)

The **TDEE (Total Daily Energy Expenditure)** is the total amount of energy a person expends in a day, including all activities, from basic functions like sleeping and energy required for daily activities such as walking, working, exercising, and energy used for food digestion.

TDEE is used to determine the number of calories a person needs to consume in a day to maintain their current weight. Based on the user's goals, TDEE can also be used to adjust daily calorie intake to achieve specific goals, such as losing weight, gaining weight, or maintaining weight.

This index, combined with the BMR index above, will be the two main formulas used throughout the system for calculations, helping users plan their nutrition schedule. The formula for calculating TDEE is :

$$\text{TDEE} = \text{BMR} \times \text{Activity Factor}$$

In this formula:

- **BMR** is the formula to calculate the Basal Metabolic Rate.
- **Activity** is the activity factor:
    - Sedentary or little exercise: TDEE = 1.2 x BMR
    - Lightly active: TDEE = 1.375 x BMR
    - Moderately active: TDEE = 1.55 x BMR
    - Very active: TDEE = 1.725 x BMR
    - Extremely active: TDEE = 1.9 x BMR

Calculating Body Fat

Body fat percentage is the proportion of fat in the body compared to the total body weight. It is an important indicator for assessing the health and development of the body. The formula for calculating body fat is (5.5), (5.6):

$$\text{Male} = \frac{495}{(1.0324 - 0.19077 \cdot \log(\text{waist} - \text{neck}) + 0.15456 \cdot \log(\text{height}))} - 450$$

$$\text{Female} = \frac{495}{(1.29579 - 0.35004 \cdot \log(\text{waist} + \text{hip} - \text{neck}) + 0.22100 \cdot \log(\text{height}))} - 450$$

Where:

- **waist** is the waist (cm).
- **hip** is the hip (cm).
- **neck** is the neck (cm).
- **height** is the height (cm).

Ideal Body Weight (IBW)

IBW (Ideal Body Weight) is an index used to estimate a person's ideal body weight based on their height. It is a widely used tool in the medical field, especially in evaluating nutritional status and treating obesity. There are many formulas used to calculate IBW,

but one of the most commonly used is the Devine formula, developed in 1974

$$\text{Male} = 50.0\,\text{kg} + 2.3\,\text{kg} \times \left(\frac{1\,\text{inch}}{5\,\text{feet}}\right)$$

$$\text{Female} = 45.5\,\text{kg} + 2.3\,\text{kg} \times \left(\frac{1\,\text{inch}}{5\,\text{feet}}\right)$$

Example: A guy who is 177 cm tall would be approximately 70 inches tall. Since 5 feet is equivalent to 60 inches, his IBW would be calculated as (2.3 × 10) + 50 = 73 kg.

Lean Body Mass (LBM)

LBM (Lean Body Mass) is the amount of body mass that is not fat, including all components of the body that are not fat, such as muscles, bones, water, and other non-fat tissues. The formula for calculating LBM is given as ,

$$\text{Male (kg)} = (0.32810 \times \text{weight}) + (0.33929 \times \text{height}) - 29.5336$$

$$\text{Female (kg)} = (0.29569 \times \text{weight}) + (0.41813 \times \text{height}) - 43.2933 \quad)$$

Calories Burned

The "calories burned" index refers to the amount of calories the body expends during activity and exercise over a specific period of time. This is an important indicator for measuring energy consumption and physical activity of the body. The formula for calculating calories burned is as follows (5.11):

In which:

- **weight** is the person's weight (kg).

height is the person's height (cm)

$$\text{Calories} = \frac{(\text{MET} \times 3.5 \times \text{weight(kg)})}{200} \times \text{time}$$

Amount of Water Needed per Day

The required daily water intake is the amount of water that the body needs to maintain proper hydration throughout the day. The formula for calculating the necessary water intake is as follows (5.12):

In which:

- **weight** is the person's weight (kg).
- **time** is the duration in minutes (cm).
- **calories** is the amount of calories burned per minute.

METs is the rate of energy expenditure relative to body weight during a specific physical task compared to a reference leve

$$\text{Water (liters)} = \left[\text{weight(kg)} + \left(\frac{\text{time}}{30\,\text{(minutes)}} \times 12\,\text{oz}\right)\right] \times 0.03\,\text{l}$$

In which:

- **water** is the amount of water needed to maintain hydration (liters).
- **weight** is the person's weight (kg).

c, Results

Below are images showcasing the implementation of health metric calculation functions in the backend source code.

```
private BMRCalculator(weight: number, height: number, age: number, gender: string) {
  if (gender === UserGender.male) {
    return parseFloat((9.99 * weight + 6.25 * height - 4.92 * age + 5).toFixed(1))
  }
  if (gender === UserGender.female) {
    return parseFloat((9.99 * weight + 6.25 * height - 4.92 * age - 161).toFixed(1))
  }
}
private TDEECalculator(weight: number, height: number, age: number, gender: string, activity: number) {
  const BMR = this.BMRCalculator(weight, height, age, gender) || 0
  return parseFloat((BMR * activity).toFixed(1))
}
private BodyFatCalculator(height: number, neck: number, waist: number, hip: number, gender: string) {
  if (gender === UserGender.male) {
    return parseFloat(
      (495 / (1.0324 - 0.19077 * Math.log10(waist - neck) + 0.15456 * Math.log10(height)) - 450).toFixed(1)
    )
  }
  if (gender === UserGender.female) {
    console.log(1.29579 - 0.35004 * Math.log10(waist + hip - neck) + 0.221 * Math.log10(height))
    return parseFloat(
      (495 / (1.29579 - 0.35004 * Math.log10(waist + hip - neck) + 0.221 * Math.log10(height)) - 450).toFixed(1)
    )
  }
}
private LBMCalculator(weight: number, height: number, gender: string) {
  // công thức hume
  if (gender === UserGender.male) {
    return parseFloat((0.3281 * weight + 0.33929 * height - 29.5336).toFixed(1))
  }
  if (gender === UserGender.female) {
    return parseFloat((0.29569 * weight + 0.41813 * height - 43.2933).toFixed(1))
  }
}
private CalorieBurnedCalculator(weight: number, time: number, met: number) {
  // (MET x 3.5 x cân nặng(kg)) / 200 x thời gian tập luyện (phút) = lượng calo tiêu hao
  const caloriePerMinutes = (met * 3.5 * weight) / 200
  return parseFloat((caloriePerMinutes * time).toFixed(1))
}
```

Figure 5.1: Implementation of Calculation Functions

Next are the images illustrating the development of the user health metrics module on the frontend.

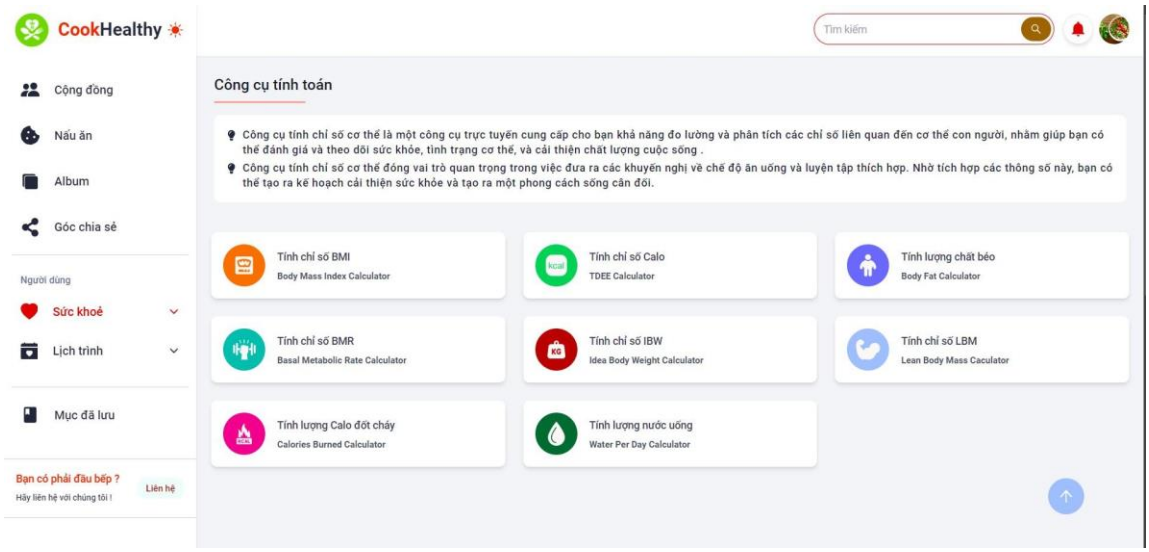


Figure 5.2: Health Metrics Viewing Interface

Users can select a calculation function, input the required body metrics, and press the calculate button as illustrated in the image below. This process enables users to quickly obtain results for key health metrics.

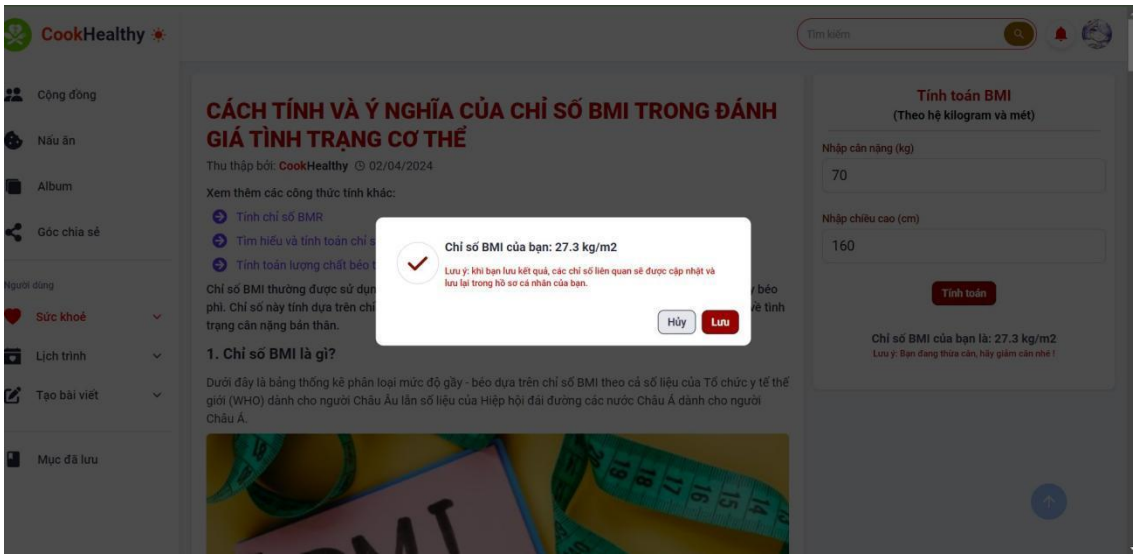


Figure 5.3: Health Metrics Calculation Interface

After users complete their health metric calculations, they can review their calculation history and track detailed weight charts. This feature helps users easily monitor their health progress and provides a more visual perspective on weight changes over time

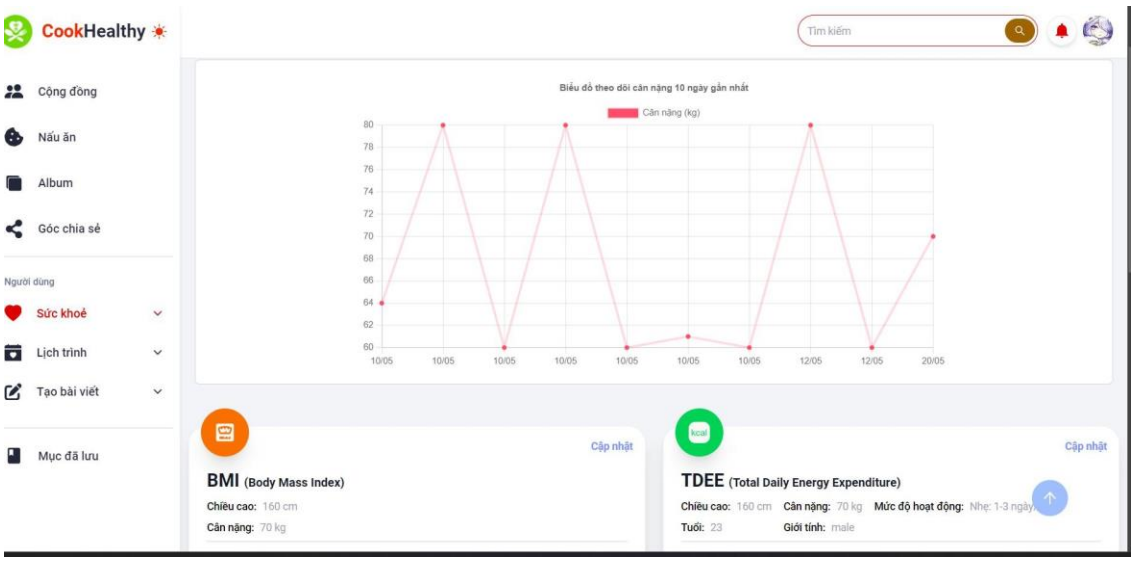


Figure 5.4: Interface for Viewing Calculated Metrics from the System

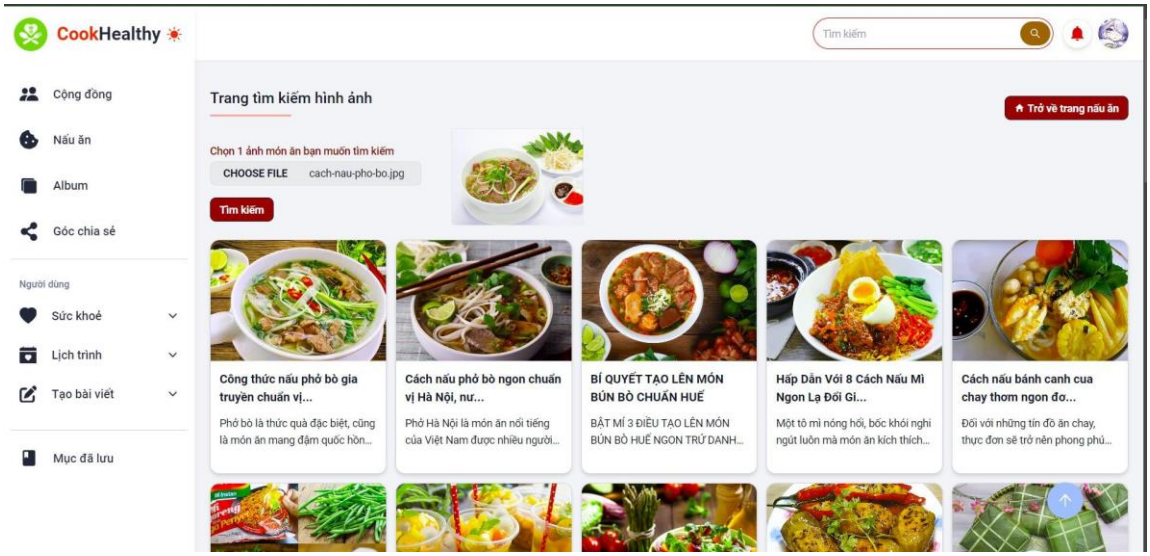


Figure 5.8: Image Search Interface

5.2 Conclusion

The project delivers a multifunctional system, providing users with the ability to share and access comprehensive information about cooking and nutrition. With its advanced image search functionality, users can conveniently and intuitively find information. The system also supports users in holistic health management by offering tools for meal planning, workout scheduling, and monitoring health indicators. Furthermore, the platform enhances social interaction through a forum where users can ask questions, post articles, and participate in discussions, fostering a vibrant and helpful community.

While the application is complete, there are still some limitations, such as suboptimal system performance, low accuracy of the image search feature, and a lack of diverse food data. Through this thesis, I have gained significant insights into the field

of health and nutrition, particularly in applying health metric calculations to practical scenarios.

Additionally, I have learned the process of developing a web application, from requirement gathering and surveys to designing workflows and using design tools to create a comprehensive interface. Notably, I gained experience using the VGG19 model and integrating it into the thesis. These skills and knowledge will undoubtedly serve as a strong foundation for my future career.

5.3 Future Development

To address the current limitations and enhance the application's effectiveness, I will focus on several key development directions in the future. First, improving system performance by optimizing the codebase and leveraging newer technologies and libraries is essential.

Additionally, expanding the database of dishes is a necessary step. This can be achieved by collecting and updating more data from various sources, providing users with more comprehensive and rich information. Finally, enhancing the application's interactivity by developing tools like a recipe recommendation system and sharing experiences in cooking and nutrition will improve the user experience and foster a more connected community.

Moreover, adding social features such as comments, ratings, and user-following options will increase user interaction and build stronger community connections.

==========

REFERENCES

[1] Cimigo, *Xu hướng ăn uống lành mạnh việt nam*. [Online]. Available: https://www.cimigo.com/vi/trends/nhu-cau-va-xu-huong-an-uong-lanh-manh-viet-nam/

[2] Cobe, *Top 10 website hướng dẫn nấu ăn nổi tiếng nhất việt nam*. [Online]. Available: https://www.cooky.vn/blog/top- 10- website- am-thuc-noi-tieng-nhat-viet-nam-2934

[3] vietnambiz, *Thị trường giao đồ ăn việt: Cuộc giằng co giữa các tên tuổi ngoại*. [Online]. Available: https://vietnambiz.vn/thi-truong- giao-do-an-viet-cuoc-giang-co-giua-cac-ten-tuoi-ngoai-20221215104438344.html

[4] Similarweb, *Cooky.vn traffic*. [Online]. Available: https://www.similarweb. com/website/cooky.vn/#overview

[5] Semrush, *Myfitnesspal traffic*. [Online]. Available: https://www.semrush. com/website/myfitnesspal.com/overview/

[6] *Website hướng dẫn nấu ăn cooky.vn*. [Online]. Available: https://www. cooky.vn

[7] *Hội dinh dưỡng việt nam*. [Online]. Available: https://hoidinhduong. vn/

[8] *Sức khỏe và đời sống*. [Online]. Available: https://suckhoedoisong. vn/

[9] B. Y. T.-V. D. DƯỠNG, *BẢNG THÀNH PHẦN THỰC PHẨM VIỆT NAM*. NHÀ XUẤT BẢN Y HỌC, 2007.

[10] *Journal-of-sport-and-health-science*. [Online]. Available: https://www. sciencedirect.com/journal/journal-of-sport-and-health-science/vol/13/issue/1