Intelligent Recipe Agent: Dinner Idea Assistant

Executive Summary

The "Intelligent Recipe Assistant" project addresses the common problem of meal planning and food waste by integrating AI with traditional Chinese recipes. Despite the increasing prevalence of smart home technologies, many households struggle with making diverse and healthy meal choices, often resulting in excess food waste. This project aims to simplify the meal planning process, provide personalized recipe recommendations, reduce ingredient waste, and promote healthy eating habits.

The project's main objectives include offering personalized recipe suggestions based on user input, preserving and sharing traditional Chinese culinary knowledge, engaging a community of cooking enthusiasts, integrating with smart kitchen devices, providing nutritional guidance, supporting multiple languages, and ensuring data security and privacy.

The project utilized Agile development methods, employing tools like Trello for task management and Git for version control. Key technologies included the Stable Diffusion API for generating high-quality food images and the ChatGPT API for personalized chat and recipe suggestions. MongoDB was chosen for its flexibility and scalability in managing diverse data types, while the user interface was designed using Figma and developed with PyQt5. Extensive testing ensured the quality and reliability of the software.

Throughout the development process, the project was divided into sprints, each focusing on different core functionalities such as user registration, recipe recommendation engine, personalized settings, advanced features, and social sharing. Continuous testing and iteration, driven by user feedback, helped refine and improve the software.

The project successfully created a user-friendly application with a robust backend. Users appreciated the AI-driven recipe suggestions and the community features, noting improvements in their nutritional balance and weight management. However, some areas for improvement were identified, such as expanding the variety of recipes and enhancing the accuracy of the generated images.

Key lessons from the project include the importance of user engagement, seamless technology integration, and the benefits of modular design. Future plans involve optimizing the recommendation algorithms, enhancing community features, and exploring internationalization to reach a broader audience. The project underscored the potential of AI in improving daily life and highlighted the importance of cross-domain collaboration.

In conclusion, the "Intelligent Recipe Assistant" project successfully combines AI with traditional culinary knowledge to enhance meal planning and reduce food waste. The project's iterative approach and focus on user feedback ensure its continuous improvement and relevance. The experience gained from this project emphasizes the value of innovation, effective resource allocation, and agile development methodologies in achieving technological advancements.

1.Introduction:

With the development of Internet of things and artificial intelligence technology, smart home is gradually integrated into daily life and has brought significant social benefits. However, many individuals and families still face difficulties in daily dining decisions, resulting in single dining choices and serious waste of ingredients. Most families lack inspiration when it comes to meal planning and are unable to make the most of available ingredients. For example, many families may waste by repeatedly buying excess ingredients because they do not know how to match the leftover ingredients in the refrigerator, which not only increases expenses, but also leads to a lack of healthy eating habits. Large organizations also face challenges in intelligent management, recipe compilation, and data collection. For example, catering companies need to efficiently manage their food inventory, avoid waste, and provide a variety of dishes to meet customer needs. To cope with these problems, we chose to develop the "Intelligent Recipe Assistant" project, which combines AI technology and traditional Chinese recipes, aiming to help users simplify the meal planning process, provide personalized recipe recommendations, reduce ingredient waste, and promote the development of healthy eating habits.

The solution of the project uses advanced artificial intelligence technology to provide users with personalized recipe recommendations and detailed cooking instructions, which can significantly improve the user's cooking experience. For example, users can input the ingredients available at home, and the system will automatically generate a variety of menu options to ensure that these ingredients are fully utilized and waste is reduced. Through the nutrition analysis module, users can obtain scientific nutrition guidance and choose healthier diet schemes, such as recommending dishes with low calories and high protein, to help users achieve their health goals. In addition, the platform also establishes a community of cooking enthusiasts, where users can share their cooking experiences and get feedback and suggestions from others, which enhances community interaction and user engagement. The project team expects that the smart recipe assistant will improve the quality of life of users, promote healthy eating, reduce waste of ingredients, and promote cultural exchange and inheritance by preserving and spreading traditional Chinese recipes. For example, through the digital preservation of traditional recipes, the project team was able to not only preserve valuable cultural heritage, but also make these rich culinary traditions easily accessible and learned by users around the globe.

2.Objectives:

2.1 Objective Contents:

1) Personalized recipe suggestions: Artificial intelligence technology is used to provide personalized recipe recommendations according to the ingredients, taste preferences and eating habits input by users, so as to help users plan their daily meals efficiently.

2) Preservation and sharing of culinary knowledge: To create a digital repository of traditional Chinese recipes and make them accessible to global users, to preserve and disseminate the rich culinary culture.

3) Community engagement: Build a community of culinary enthusiasts where users can share, rate, and discuss recipes and cooking tips, facilitating the exchange of knowledge and experience.

4) Smart kitchen integration: expand software functions to seamlessly integrate with smart kitchen equipment, automate meal planning based on existing ingredients, and improve cooking efficiency.

5) Nutrition guidance: Integrating nutrition analysis module to provide nutrition information of recipes to help users make healthier diet choices.

6) Support multilingual and cross-cultural communication: Develop multilingual support functions to make it easy for users with different cultural backgrounds to use and promote cross-cultural cooking communication.

7) Data security and privacy protection: Ensure that users' personal information and usage data are fully protected and comply with relevant privacy and data protection regulations.

2.2 Objective Analysis and Implementation

1) Personalized recipe suggestion: This is one of the core functions of this project, which provides accurate recipe recommendation by analyzing the ingredients, taste preferences and eating habits entered by users. The project team made use of ChatGPT's API and database technology to achieve this. ChatGPT uses natural language processing techniques to understand the user's needs and generate personalized recipe recommendations, while the database stores and manages recipe data to ensure fast queries and responses.

2) Preservation and sharing of culinary knowledge: Create a digital library of traditional Chinese recipes to preserve and disseminate the rich culinary culture. The project team has developed an easy-to-use platform that allows users to access and

upload traditional recipes in multiple formats such as text, images, and videos. Database technology plays a key role here, enabling users to easily find and share recipes through efficient storage and retrieval functions. The project team has also added search and classification capabilities, and regularly hosts online events to encourage users to share their recipes and cooking tips.

3) Community participation: Add a community feature to the platform where users can share, rate and discuss recipes and cooking tips. The project team used a database to store user-generated content and utilized ChatGPT for content management and interaction to enhance interaction between users. The project team also added social interaction features, such as comments, likes, private messages, etc., and regularly published themed activities or challenges to stimulate user enthusiasm for participation.

4) Smart kitchen integration: Develop interfaces with smart kitchen devices to realize automated meal planning and ingredient management. The project team manages device status and user food ingredient information through a database, and ensures compatibility with multi-brand and multi-type smart devices. Add equipment status monitoring and reminder functions, such as reminding users when food ingredients are about to expire.

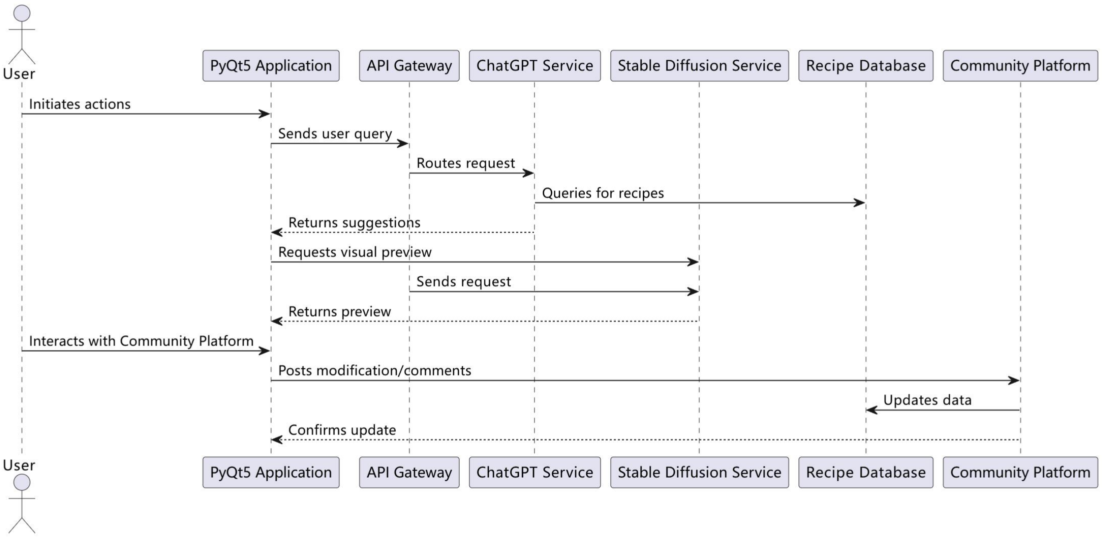
5) Nutrition guidance: Through the nutrition analysis module, the nutrition information of recipes is provided to help users make healthier diet choices. The project team integrated a nutrition database and utilized ChatGPT to provide personalized health advice. Personalized nutrition advice is provided based on the user's health goals (e.g., weight loss, muscle gain, etc.), and a food record function is added to help users track their daily nutrient intake.

6) Support multi-language and cross-cultural communication: multi-language support is added to the platform, and users can choose their own language interface. Through ChatGPT and machine translation technology, the recipe content is translated into multiple languages, and the cultural topic section is added to introduce the cooking culture and characteristic recipes of various countries, so as to promote cross-cultural communication and learning.

7) Data security and privacy protection: Ensure that users' personal information and usage data are fully protected. The project team adopts advanced data encryption technology, strictly controls data access rights through the database, conducts regular security audits, provides privacy Settings options, and enhances users' trust in the platform.

Through detailed goal analysis and optimization, as well as leveraging the database and ChatGPT's apis, the team ensured that the Cookbook Assistant project was fully delivering on its intended goals, bringing significant value and experience to its users.

3. Methodology:



3.1 Project management

3.1.1 Agile Software Development

The team used agile development methods and set short Sprints through the Scrum framework, and completed certain product increments at the end of each cycle. The team holds regular daily standups, reviews, and retrospectives to assess progress and efficiency, and to adjust plans and strategies in a timely manner. The team used Trello as a task management tool and created task boards to track project progress and team tasks. Team members set different columns on the Trello board to represent the status of the task, each task card contains a detailed description, deadline, and owner, and team members can add comments and attachments to ensure the transparency and collaboration of the task.

3.1.2 Version control

The team uses Git as a version control system. Git provides powerful branch management that allows team members to work on new features, test them, and fix bugs in parallel without affecting the stability of the main version. Each team of developers worked on separate branches that were eventually merged into the master branch.

3.2 Stable Diffusion API

In this system, the integration of the Stable Diffusion API is a core component for achieving high-quality dish image generation. This process involves several key steps and methodological choices, as detailed below.

3.2.1 API Integration

The primary reason for integrating the Stable Diffusion API is its powerful image generation capabilities and high flexibility. In contrast, using local models such as DALL-E or other deep learning models would require managing model updates and hardware maintenance, which involves significant technical support and ongoing cost investment. This is particularly disadvantageous for subsequent deployment to end devices.

Process:

1. **Register and Obtain API Key**: First, register on the Stable Diffusion platform and obtain the necessary API access key.
2. **API Calls**: Use Python to script the process, sending image generation requests through the Stable Diffusion API and receiving the generated images.
3. **Data Handling and Security**: Ensure that all data transferred via the API is properly encrypted and handle exceptions such as API throttling, timeouts, or failures.

3.2.2 Image Generation Logic Integration

To better serve user needs with the image generation functionality, it is necessary to build a layer of business logic on top of the API. This includes understanding and pre-processing user input as well as post-processing the output from Stable Diffusion.

Process:

1. **Pre-processing User Input**: Clean and format the recipe generated by ChatGPT, including removing unnecessary filler words and dividing the prompt into prompt and negative prompt.
2. **Post-Processing Response**: Use predefined prompts to adjust the output from Stable Diffusion, ensuring that the generated images meet contextual requirements, such as adjusting the image style and refining the image content to better match the needs of the target user group.

3.2.3 Performance Optimization

Performance optimization is crucial for improving responsiveness and handling more concurrent user requests.

Process:

1. **Caching Mechanism**: Introduce a caching policy to store the results of frequently requested images, reducing the number of requests to the Stable Diffusion API and thereby decreasing server load and response time.
2. **Load Balancing**: Utilize a load balancer to distribute API requests, ensuring service availability and stability during periods of high traffic.

Rationale for Choosing Stable Diffusion API

The choice of the Stable Diffusion API is based on considerations of project sustainability, cost efficiency, ease of implementation, and future scalability. This allows the focus to remain on core product functionality and user experience, rather than on the complexity and maintenance issues of the underlying technology. Additionally, compared to other text-to-image models such as Midjourney and DALL-E, Stable Diffusion offers specific models tailored to different vertical domains, which can generate higher-quality images more effectively.

3.3 Chatgpt API

In our system, integration of the large language model is a core part of implementing chat functionality. This process involves a number of key steps and choices of methodology, the details are shown as follows.

3.3.1 API Integration

The main reason for choosing to integrate ChatGPT was its high degree of flexibility and powerful language modeling capabilities for generating personalized chat responses. The reason why we don't use a local model, such as LLaMA 3 or GPT-J , is because it would require us to manage model updates and hardware maintenance on our own, which could involve significant technical support and ongoing cost investment.

Process:

1. Register and get API key: First register with the OpenAI platform to get the necessary API access key. 2.
2. API calls: Use openai model in Python to build API requests, send user input to ChatGPT, and receive the generated responses.
3. Data Handling and Security: Ensure that all data transferred via the API is properly encrypted and handle exceptions such as API throttling, timeouts, or failures.

3.3.2 Chat Logic Integration

In order for a chatbot to better serve its users, we need to build a layer of business logic on top of the API, which includes understanding and pre-processing user input and post-processing ChatGPT's output.

Process:

1.Pre-processing User Input: Cleaning and formatting user input, such as removing useless filler words or identifying keywords.

2.Post-Processing Response: Using pre-prompt to adjust ChatGPT's output to ensure it is appropriate to the context, such as adjusting the tone and refining answers to more closely match the needs of the target user group.

3.3.3 Performance optimization

Performance optimization is essential to improve responsiveness and handle more concurrent users.

Process:

1.Caching Mechanism: Introducing a caching policy to store answers to frequently asked questions and reduce requests to the ChatGPT API, thus reducing server load and response time.

2.Load Balancing: A load balancer is used to distribute API requests to ensure service availability and stability during high traffic.

The choice of the ChatGPT API was based on the sustainability of project requirements, cost efficiency, ease of implementation, and consideration of future expansion. This allowed us to focus on core product functionality and user experience rather than the complexity and maintenance issues of the underlying technology.

3.4 MongoDB database

The main reasons for choosing MongoDB as your database are its flexibility, scalability, and performance advantages. The reason we don't use a relational database like MySQL is that its fixed table structure and relatively limited horizontal scaling capabilities make it difficult to handle future growth and complex recipe data.

Process:

1. Install and configure MongoDB: First, install MongoDB on the server and make a basic configuration to ensure the availability of database services.
2. Database design: According to the requirements of the intelligent recipe assistant, design the database schema, including the collection and document structure, to support a variety of data types.
3. Data import: Use MongoDB's import tool to import existing recipe data into a database and set indexing to optimize query performance.
4. Integrated API: Use Python's `pymongo` library for API integration to realize the function of adding, deleting, changing and checking data.

3.5 UI Implement

The UI design is developed with a user-centered approach. We use Figma to create basic layouts to outline the structural arrangement of the UI elements and develop detailed static design interfaces. Meanwhile, we use PyQt5 to build interactive models of the user interface to simulate user interactions and interface dynamics. In terms of color schemes, considering the product’s association with kitchens, recipes, and food elements, we apply color matching principles[1]. Taking into account the diversity of users, we also provide various options for users with color vision deficiencies[2].

3.6 Quality Assurance and Testing

The iterative design process relies heavily on user feedback. The team conducted user BlackBox and Regression testing at various stages of development, had real users interact with the user interface prototype, collected and recorded user feedback as well as performance metrics during testing to identify usability issues and areas for improvement. User test results are recorded on Trello, and the team prioritizates and improves accordingly based on the feedback. At the same time, we verify the requirements by validation, and verify the correctness of the code by verification. A/B testing is performed before the software is deployed to determine the best solution by comparing the features or designs of different versions.

4. Development Process:

The development process of the "Intelligent Recipe Assistant" project includes multiple stages of planning, design, implementation, testing, and deployment. At each stage, the team encountered problems and challenges and came up with solutions through hard work and technical attempts. The following is a detailed description of the project development process:

4.1 Planning

In the project start-up phase, the project team carried out detailed project planning through collective meeting content, and set clear goals and schedules. The main challenge encountered by the project team throughout the development process was maintaining clear communication and consistency within the team, not least because of the remote collaboration within the project team. To overcome this, regular weekly team meetings were established within the team and Trello kanban board was utilized to update task progress in real time. Project initiation meetings were organized and the project was refined into detailed user stories and requirements.

4.1.1User Stories

In the face of the overall requirements, the project team initially collected and consulted the user's needs and expectations through user interviews and literature review, and transformed them into development tasks: The Intelligent Recipe Agent is an intelligent assistant designed to help users quickly decide on dinner ideas. The system will provide personalized dinner suggestions based on the user's dietary preferences, nutritional needs, available ingredients, and possible allergic reactions. After preliminary research and collection, the user stories are summarized as follows:

User Story 1: Personalized Recipe recommendations

As a busy professional, I wanted to be able to quickly get personalized recipe recommendations for dinner based on my eating preferences and what's available at home, so I could prepare a healthy, delicious meal in no time.

Functional requirements: The user inputs ingredients and preferences, and the system provides recipe recommendations.

Non-functional requirements: Recommender systems should respond to user requests within 5 seconds.

Test condition: The recipe presented by the system matches the ingredients and preferences entered by the user.

User Story 2: Uploading and Sharing Recipes

As a food lover, I would love to be able to upload my original recipes and share them with the rest of the community, including the ingredients list, cooking steps, and photos, so that I can share my cooking tips and get feedback.

Functional requirements: Users create and upload recipes, and recipes can be viewed and commented on by other members of the community.

Non-functional requirements: The system should support multimedia content uploading and guarantee the copyright of user-generated content.

Test conditions: user-uploaded recipes can be successfully saved and displayed in the community.

User Story 3: Access to nutritional information

As a health-conscious user, I would like to know detailed nutritional information about recommended recipes, including calorie, protein, fat, and carbohydrate content, so that I can make dietary choices that match my personal health goals.

Functional requirements: The system provides a nutritional analysis report for each recipe.

Non-functional requirements: Nutritional information should be accurate and easy for users to understand.

Test conditions: Users can customize diet plans that meet their individual health needs based on nutritional information.

Through the above analysis of the user story content, the project team got the preliminary requirements, and finally the project team summarized and analyzed the specific classified requirements for detailed analysis.

4.1.2 Functional requirement

For recipe storage, the system must be able to help users upload their own recipes through a standardized process, including a formatted title, a list of ingredients, a description of the steps, and optionally add images or videos. The system must implement a moderation mechanism to ensure that all uploaded recipes meet established quality standards and community guidelines. To improve search and organization efficiency, users are also able to categorize and tag recipes based on ingredients, cuisine lines, cooking time, etc.

Intelligent recommendation and personalization is one of the key features of the project. The system must be able to accept user input through interaction and generate dinner suggestions based on user input. At the same time, detailed user profiles will be established based on user interactions and preferences, and through intelligent search functions, users will be allowed to search recipes according to various parameters such as ingredients, cuisines, and cooking methods. The recommendation algorithm will be evaluated and optimized periodically to improve recommendation accuracy and user satisfaction.

Community interaction and feedback is another important aspect to enhance the user experience. The system must allow that users will be able to share their recipes and cooking experiences to social media and the cooking community and provide feedback on recipes and recommendation algorithms through the provided mechanisms. In addition, the system must also allow that users will be able to rate recipes and the system will use this data to continuously improve the recommendation.

The system needs to complete the requirements of smart kitchen integration, so that the system can be integrated with a variety of smart kitchen equipment (such as smart refrigerator, stove), provide automatic food ingredient tracking and management functions, and use the Internet of things technology to realize the seamless connection with smart kitchen equipment.

4.1.3 Nonfunctional requirement

The intelligent recipe agent project requires the system to have fast response capabilities to provide a timely user experience. Specifically, the system needs to be designed to provide dinner suggestions within five seconds of a user's request. This requires that the backend algorithms must be highly optimized and the server processing power must be powerful enough to support fast data processing and intelligent recommendation generation.

The system needs to be highly available to ensure that users can rely on intelligent recipe agents for meal planning at any time. For this, the system must have 99% uptime. This requires systems with failover capabilities, disaster recovery plans, and continuous monitoring to minimize potential downtime and service disruptions and ensure service continuity and reliability.

To satisfy a wide user base, intelligent recipe agents must be compatible with mainstream operating systems and browsers. This means that systems need to be thoroughly tested and optimized across multiple platforms, including Windows, macOS, iOS, Android, as well as Chrome, Firefox, Safari, and Edge, to ensure that users have a consistent experience regardless of device or browser.

The security of system data is the top priority of the project. The system must employ advanced encryption techniques to protect the stored user data against data leakage or unauthorized access. In addition, regular security audits and vulnerability scans will help identify and fix potential security issues, ensuring that the security and privacy of user information is protected.

Intuitive and easy to use user interface design is essential for all users. The project system must provide a concise and clear user interface that is easy to navigate in order to reduce the cognitive burden of the user. The system should provide clear instructions and timely feedback to help users complete their tasks efficiently. At the same time, the interface design should also take into account the needs of different users, including disabled users, to ensure accessibility and comply with accessibility design standards.

1)Data requirements

The data requirements of the Intelligent recipe Agent project emphasize a deep understanding and management of user data. The system will collect basic information about the user, which includes not only basic contact information, but also key dietary preferences, ingredients available in the home, and any food allergies, ensuring that recommendations meet the user's personal health and taste needs. In addition, the project requires a comprehensive and diverse recipe database, where each recipe exhaustively lists the required ingredients, step-by-step cooking instructions, and detailed nutritional information to support users in making informed dietary choices.

2) Legal requirements

In terms of legal requirements, the system will strictly comply with data protection laws and privacy policies to ensure the confidentiality and security of user information. In addition, all recipes and content will follow intellectual property laws, ensuring that all materials used are legally authorized, thus avoiding copyright infringement issues. The system will conduct regular compliance checks to ensure that all operations comply with the latest laws and regulations.

3) Reliability requirements

The system must have error detection and handling mechanisms that can provide clear feedback when the user makes a typo and guide the user to the correct action. At the same time, the system will use advanced algorithms to analyze user input to ensure that dinner recommendations are based on the most accurate and up-to-date data, verifying that the information and recommendations provided are accurate and consistent. In addition, the system will implement a regular data backup strategy to avoid the loss of user data in any case. The system needs to implement strict data management and audit processes to avoid providing any contradictory or misleading information.

4.1.4 Problems and solutions

Unclear needs and user pain points: In the process of developing an intelligent recipe assistant, users may not know exactly what features or services they need. Their needs can be complex and difficult to capture in simple questions and conclusions. To overcome this difficulty, project teams can adopt open-ended questions to guide users in expressing their requirements. The project team initially investigated user portraits through literature review, and then collected a large amount of information through questionnaire surveys, in-depth interviews or user observation. Finally, users are invited to participate in the design experience feedback of the product or service, so that users can directly participate in the prototype design and put forward ideas, simplify user needs and help design and optimize the content.

Technical feasibility and limitations: When collecting user requirements, the project team may face another challenge, that is, the requirements proposed by users may exceed the limitations of current technology and resources. To solve this problem, the project team conducts a technology evaluation. This includes identifying which requirements are achievable and which require additional technological investment or resources. Working closely with the technical team is essential to ensure that the requirements match the technical capabilities of the project team. In this way, the project team can ensure the feasibility and sustainability of the project.

Requirements change and adaptability: During the process of the project, the user feedback, the requirements change, which requires the project team to be able to respond flexibly. To solve this problem, the project team needs to establish a flexible requirements management process. This process should allow requirements to be updated and adapted in project iterations. The project team adopted agile methods of development to help the project team better adapt to changes in requirements. Through regular communication and feedback with the users, the project team can ensure that the direction of the project is consistent with the needs of the users.

4.1.5 Action Plan

After requirements gathering, the project team adopted the scrum agile development framework. Divide the work action through different roles within the team Task Content:

Scrum Master: Responsible for ensuring that the Scrum process is implemented and removing obstacles.
Product Owner: Manages the product backlog, sets priorities, and defines requirements.
Development Team: front-end developers, back-end developers, UI/UX designers, QA testers, etc.
Follow that team framework, specify sprints, and review them in a timely manner.

Sprint cycles take 2 weeks at a time and unfold in the following steps:
Sprint Planning: Set Sprint goals and create Sprint to-do lists.
Daily Scrum: 15 minutes a day to share progress and obstacles.
Sprint execution: Develop, test, and integrate as planned.
Sprint Review: Demonstrate completed features and gather feedback.
Sprint Retrospective: Review performance, identify points of improvement, and develop an improvement plan.

Primary functional division and the initial Sprint planning

Sprint 1-3: Core function development

Sprint 1: User registration and log in

Function: user registration, login, and personal data management.

Task: Database design, API development, front-end page design, user validation.

Sprint 2: Basic recipe recommendation engine

Function: Provide the basic recipe recommendations according to the user information.

Task: recipe database creation, preliminary development of recommendation algorithm, API development, front-end integration.

Sprint 3: User personalized setting

Features: Users can set dietary preferences, allergens, and health goals.

Task: user setting interface design, back-end data processing, and recommendation algorithm optimization.

Sprint 4-6: Function enhancement and optimization

Sprint 4: Advanced recommendation function

Function: Recommendation optimization based on user history selection and feedback.

Tasks: Data collection and analysis module, recommendation algorithm improvement, and user feedback mechanism.

Sprint 5: Recipe details and scoring system

Features: Displays detailed recipe information, user ratings, and review system.

Task: Front-end interface design, API development, scoring and comment function implementation.

Sprint 6: Social sharing and interaction

Features: Users can share recipes to social media and view friends' recommendations.

Task: Social sharing interface integration, friend interaction function development, UI optimization.

Sprint 7: Complete system test and Bug repair

Features: comprehensive system testing, performance optimization, fixing all known problems.

Task: QA testing, performance testing, code optimization, and repair.

Periodic review and adjustments: At the end of each Sprint, review team performance through Sprint Retrospective, identify and implement opportunities for improvement.

User feedback: collect user feedback, adjust the product to-do list, to ensure that the product constantly meet user needs

4.1.6 Test Case

Test Case ID	Test Type	Test Objective	Precondition	Test Steps	Expected Results
TC_DRA_001	Forward Test	Verify whether the system can provide personalized recipe suggestions based on the user's input ingredients.	User has logged in and entered the available ingredients (chicken, potatoes).	1. Log in to the system. 2. Enter the available ingredients: chicken, potatoes. 3. Submit the ingredients and request recipe advice.	The system displays at least one recipe proposal containing chicken and potatoes.
TC_DRA_002	Negative Test	Verify the error processing of the system when the user enters invalid food ingredients.	User logged in.	1. Log in to the system. 2. Input invalid ingredients: Jupiter Rock. 3. Submit	The system displays an error message indicating that the food ingredient is invalid and requests

Test Case ID	Test Type	Test Objective	Precondition	Test Steps	Expected Results
				the ingredients and request recipe advice.	re-input.
TC_DRA_003	Forward Test	Verify whether users can upload their recipes and interact with the community.	User has logged in and prepared a recipe.	1. Log in to the system. 2. Navigate to the recipe upload page. 3. Enter the recipe name, ingredients, cooking steps, and upload optional photos or videos. 4. Submit the recipes and request community reviews.	The recipe is successfully uploaded, and other users can score and comment on the recipe.
TC_DRA_004	Negative Test	Verify the system for the recipe upload of illegal string content.	User has logged in and prepared a recipe containing illegal string content.	1. Log in to the system. 2. Try to upload a recipe containing the illegal string content.	The system prevents uploads and displays an error message, requiring community adherence to the guidelines.
TC_DRA_005	Forward Test	Verify that the nutrition analysis module provided by the system can provide health	User selects a recipe and requests nutritional analysis.	1. Choose a recipe. 2. Request advice for nutritional analysis and health alternatives.	The system provides a nutritional analysis of the recipes and suggests healthier alternatives.

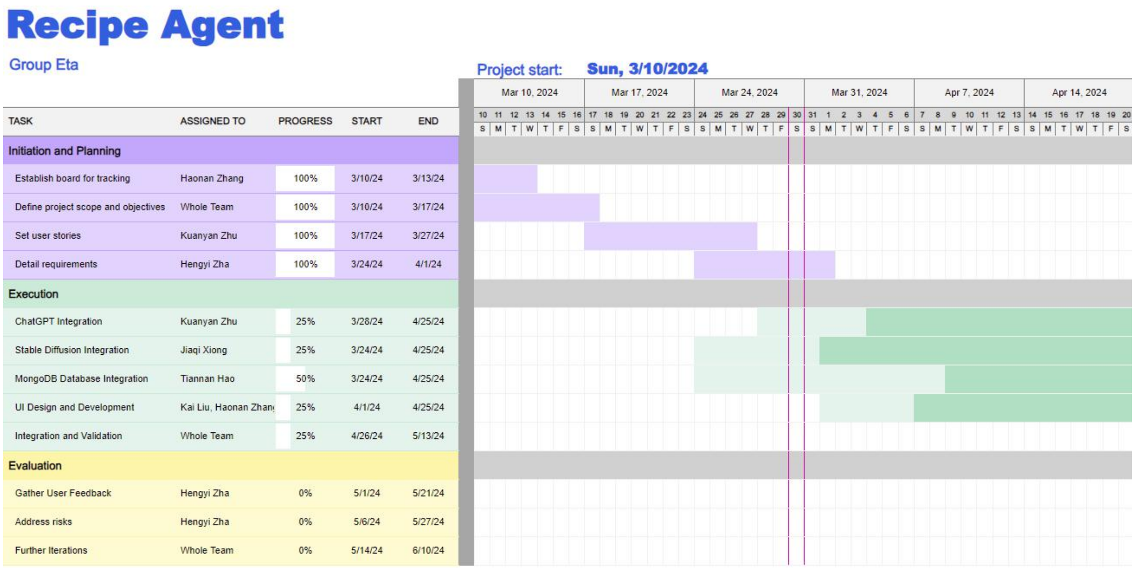
Test Case ID	Test Type	Test Objective	Precondition	Test Steps	Expected Results
		alternatives.			
TC_DRA_006	Negative Test	Verify the processing of the system when the recipe nutrition information is missing.	User chooses a recipe with incomplete nutrition information.	1. Choose a recipe with incomplete nutrition information. 2. Request for a nutritional analysis.	The system notifies users that nutrition information is not available and recommends choosing other recipes.
TC_DRA_007	Forward Test	Verify that the system can accurately understand the recipe requirements entered by the user, and provide the corresponding preview.	User has already logged into the system.System has integrated Stable Diffusion or similar technology to generate a visual preview of the dishes.	1. User logs on to the system. 2. User enters a specific recipe requirement (e.g., "I want a low-carb dinner with chicken and vegetables"). 3. User requests the system for a recipe preview.	1. The system accurately analyzes the recipe requirements entered by users. 2. The system uses Stable Diffusion technology to generate and display a visual preview of the recipe. 3. The preview image is clear and accurately reflects the recipe content requested by the user.
TC_DRA_008	Negative Test	Verify the processing power of the system when the user input is fuzzy or unclear.	User has already logged into the system.	1. User logs on to the system. 2. User enters a fuzzy recipe requirement (e.g., "casual" or "dinner").	1. The system prompts users to enter more specific recipe requirements. 2. The system may provide generic recipe advice or a set of options for users

Test Case ID	Test Type	Test Objective	Precondition	Test Steps	Expected Results
				3. User requests the system for a recipe preview.	to choose from. 3. If the user continues to request a preview, the system shall generate a representative or random preview of the dishes and advise the user to provide more information for a more accurate recommendation.

Through the above four sets of test cases, the core four functional requirements are predicted to ensure that the research and developed application software can complete the coverage of key functions. When designing test cases, it is critical to ensure that these key features are fully tested. By designing test cases for each feature, the project team can verify that the system is able to perform its core tasks as intended.

At the same time, the simulation software system will encounter various situations in the actual operation, including normal operation and abnormal input. The positive test ensures that the system works properly under standard operation, while the negative test examines the system's response and fault tolerance in the face of errors or abnormal situations. This test method helps to improve the stability and reliability of the system. The test results provide the basis for the continuous improvement of the product. By collecting and analyzing the test data, the development team can understand the advantages and disadvantages of the product, and then optimize and upgrade accordingly to achieve continuous improvement and innovation of the product. Software projects face various risks in the development process, including technical risk, management risk, market risk, etc. Through the test cases, the project team can identify and assess these risks and take appropriate measures to manage and mitigate them to ensure the smooth progress of the project.

4.1.7 Timeline

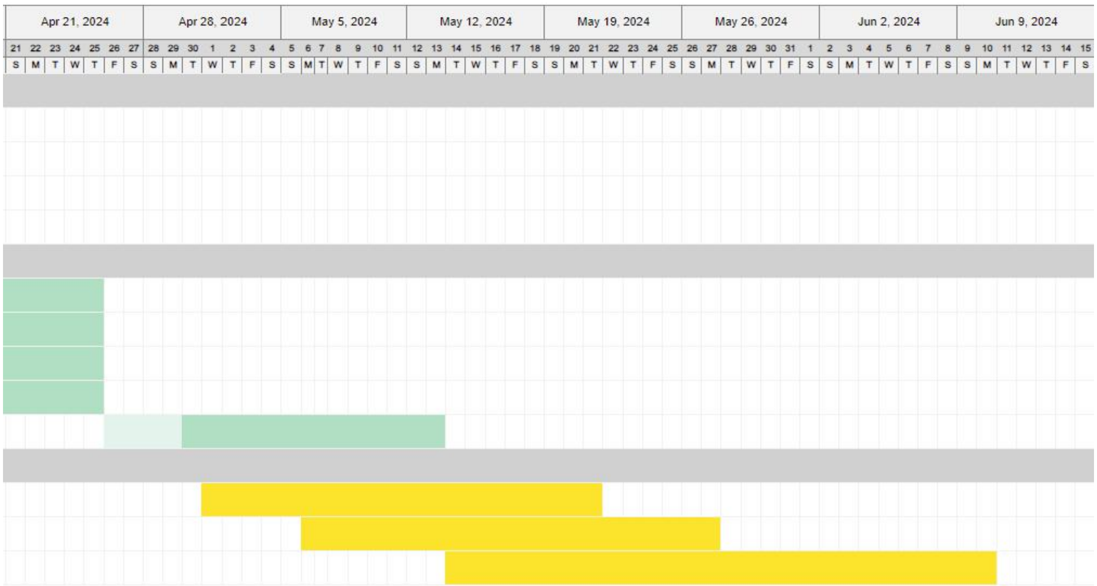


Recipe Agent

Group Eta

Project start: Sun, 3/10/2024

TASK	ASSIGNED TO	PROGRESS	START	END
Initiation and Planning				
Establish board for tracking	Haonan Zhang	100%	3/10/24	3/13/24
Define project scope and objectives	Whole Team	100%	3/10/24	3/17/24
Set user stories	Kuanyan Zhu	100%	3/17/24	3/27/24
Detail requirements	Hengyi Zha	100%	3/24/24	4/1/24
Execution				
ChatGPT Integration	Kuanyan Zhu	25%	3/28/24	4/25/24
Stable Diffusion Integration	Jiaqi Xiong	25%	3/24/24	4/25/24
MongoDB Database Integration	Tiannan Hao	50%	3/24/24	4/25/24
UI Design and Development	Kai Liu, Haonan Zhan	25%	4/1/24	4/25/24
Integration and Validation	Whole Team	25%	4/26/24	5/13/24
Evaluation				
Gather User Feedback	Hengyi Zha	0%	5/1/24	5/21/24
Address risks	Hengyi Zha	0%	5/6/24	5/27/24
Further Iterations	Whole Team	0%	5/14/24	6/10/24



4.2 Design

During the design phase, the team focused on the design of the system architecture and user interface. In this stage, the team accurately transforms the user needs of the Planing part into specific technical solutions.

In the design stage, the overall architecture of the software is divided into three main levels: API layer, data layer, and user layer.

API layer: processes user input, calls the AI model to generate recipe suggestions and visual content.The API layer processes requests sent from the front-end users, interacts with ChatGPT and Stable Diffusion API, and returns the processing results to the front-end.

Data layer: store the user data and the recipe data uploaded by the user, and cache the generated recipes.The data layer manages the user data and the recipe data through the MongoDB database. The data layer adopts the modular design principle to divide the database into multiple collections, and each collection is responsible for the specific data type to improve the maintainability and scalability of the data.

User layer: Show the user interaction interface and respond to user events
The user layer implements the UI interactive interface through PyQt 5. The user-layer interface design follows the UI prototype exported from Figma, ensuring a consistent user experience, and implementing a responsive design and user interaction functions, such as drag-and-drop upload, and instant search.

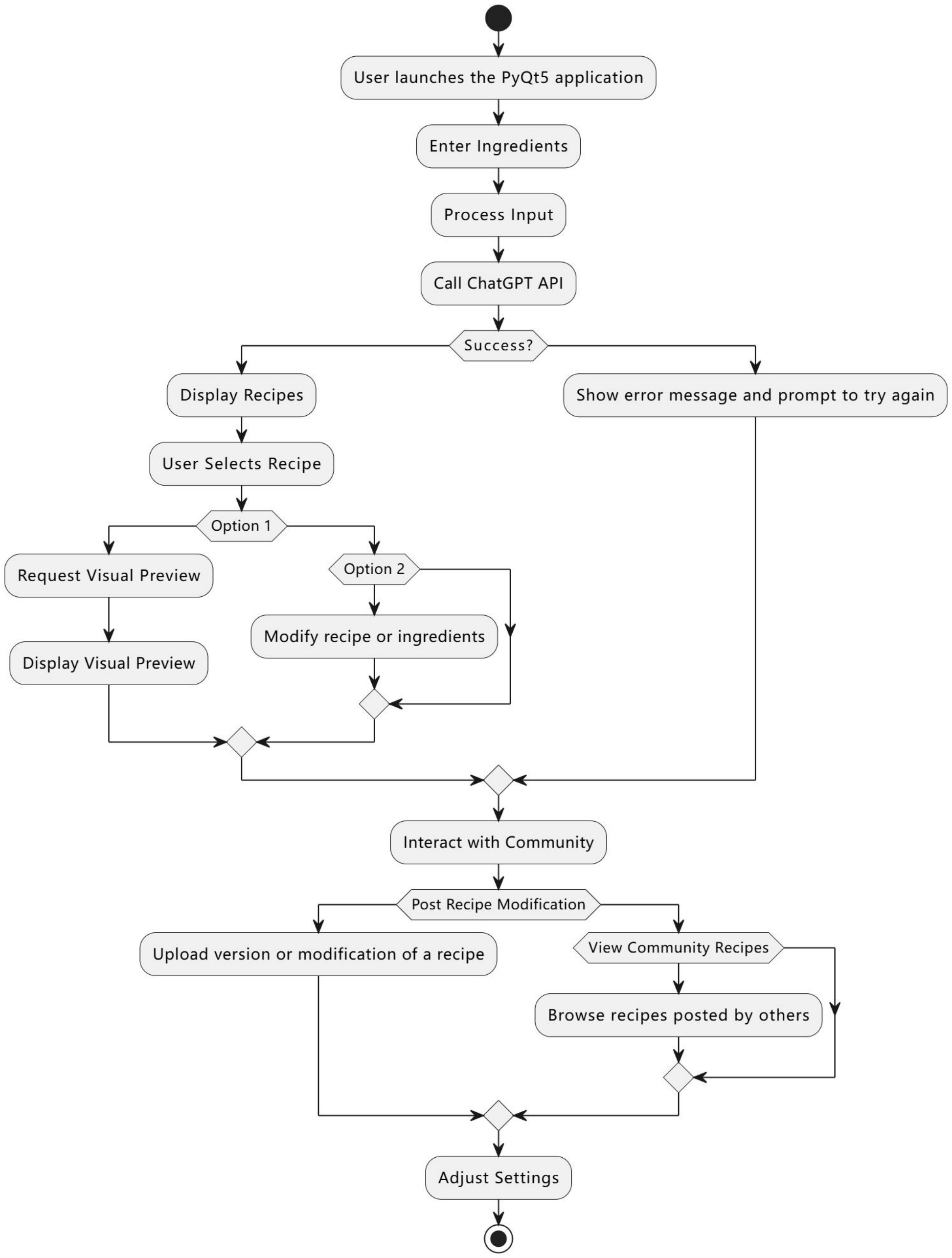
4.2.1 System Architecture Design

1) Technical stack
The team chose to use Python as the back-end development language. Python The language is easy to understand and easy to integrate with the OpenAI API, helping the team to quickly develop and reduce maintenance costs.

The team managed the database using the MongoDB. MongoDB Provide flexible data models and rich query language, suitable for handling the diversity of recipe data in project team projects.

The team adopted PyQt 5 as the front-end frame. Because the content design of the software is relatively simple, the team uses PyQt 5 rich controls to create a smooth user interface for users and achieve rapid development.

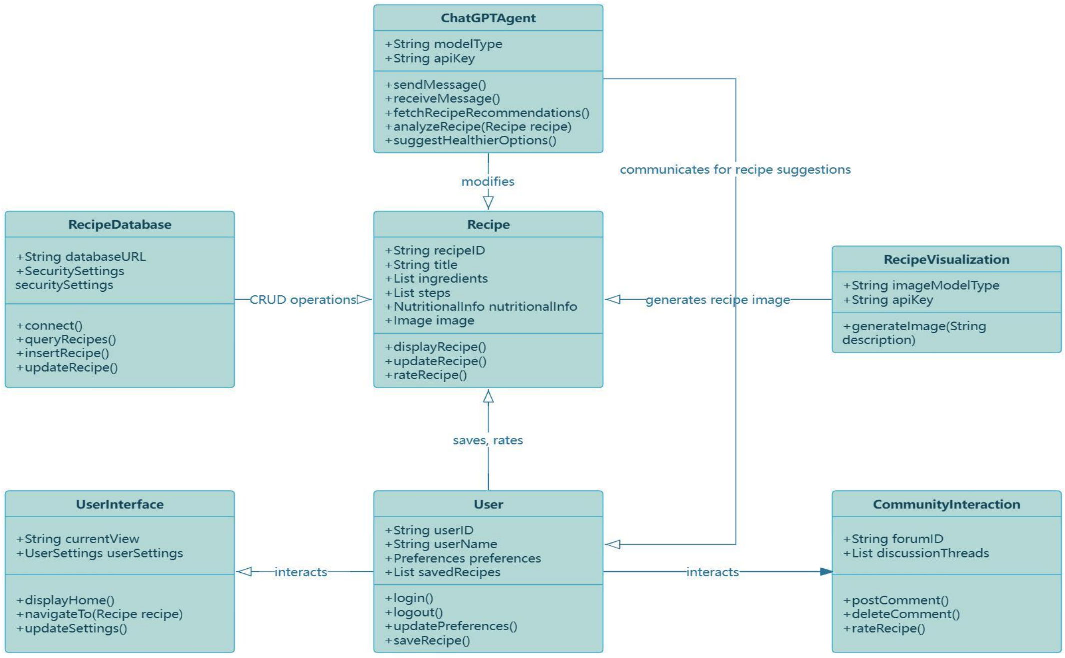
4.2.2 UI/UX Design



1) Prototype design and integration
The team used Figma to quickly create the interface prototype. Figma supports multi-person real-time collaboration, and team members can synchronously view updates from other members to improve the efficiency of team collaboration. At the same time, Figma can add interaction effects in the UI prototype for proof of concept before development.

The team converted the UI prototype designed by Figma into the PyQt code. Team members reconstruct the UI interface in PyQt according to the design specifications in Figma, and restore the visual effects and interactive logic of the Figma prototype.

4.3 Implementation



4.3.1 Implementation Process

1)Integrating ChatGPT API:

Begin by registering on the OpenAI platform to obtain the necessary API keys. This allowed us to authenticate our requests and ensure secure communications with the OpenAI services.

In terms of building API Requests, using Python and the openai library, we constructed API requests that would send user inputs to the ChatGPT model and receive responses.

This ensured the reliability and robustness of our chat functionalities.

Before the chat content is passed in, an initial prompt as follows should be passed to the model

Initial prompt = ["From now on, you need to play the role of my smart kitchen butler and make feasible recipe recommendations for the available ingredients I provide (in addition to existing recipes online, you also need to take into account personal recipes uploaded by the users themselves). You need to play it very naturally, as if you were chatting with me. After the user has decided to make a certain dish you need to provide him with a detailed recipe, which must contain a point-by-point list of ingredients and steps to follow. In addition, I need you to improvise an introductory paragraph, leading the user to ask you questions, of course, the user wants to chat is also possible. No need to say of course, I'm happy to be your smart kitchen butler or something like that in response, just start. There's also no need to say imagine you're whoever you are, because you're now my smart kitchen butler, note the identity. Here's a transcript of our chat where you need to respond to the latest question"]

When the model receives a preset prompt, it becomes the intelligent recipe agent and carries out follow-up services for us.

Finally, for data handling and security, implemented encryption for data in transit and error handling mechanisms to manage API throttling, timeouts, or failures gracefully.

2)Developing Interactive Recipe Customization:

User Input Processing is developed in the system to clean and format user inputs, removing unnecessary filler words and identifying key ingredients and preferences. This preprocessing helped in refining the search results for recipes.

The chat is recorded in its entirety and re-sent to the model at each session along with the previous memories to maintain session consistency.

Post-processing of ChatGPT's outputs was also crucial. Tailored the chatbot's responses to ensure they were contextually appropriate, adjusting the tone and details based on user feedback received during the session.

3) Implementing Stable Diffusion for Recipe Visualization:

The Stable Diffusion model was integrated into our backend. We developed functionality to trigger image generation based on the recipes being discussed in the chat, enhancing the user experience with visual aids.

To optimize image generation, we are able to implement constraints on the generated graph based on prompt engineering. At the same time, we optimise the prompt structure, the structure of the generated preview graph includes a positive prompt and a negative prompt, which can constrain the generated image from both positive and negative directions, ensuring the accuracy and authenticity of the generated image.

```
.json
"text_prompts": [
    {
        "text": "A plate of scrambled eggs with tomatoes, vibrant colors, high detail,
perfectly cooked eggs, juicy tomatoes, served on a white ceramic plate, garnished
with green herbs, shot in a well-lit kitchen setting, (best
quality,4k,8k,highres,masterpiece:1.2), ultra-detailed,
(realistic,photorealistic,photo-realistic:1.37), HDR, UHD, studio lighting, ultra-fine
painting, sharp focus, traditional Chinese cuisine style, warm and inviting tones,
natural lighting.",
        "weight": 1
    },
    {
        "text": "nsfw, (low quality,normal quality,worst quality,jpeg artifacts),
cropped, monochrome, lowres, low saturation, ((watermark)), (white letters), food
stains, unappetizing, overcooked, undercooked, burnt, poorly arranged, out of
focus, blurred, dirty plate, unclean background, unappealing presentation.",
        "weight": -1
    }
```

4) Building the Community Platform:

Implementing features that allow users to upload their recipes, share modifications, and interact with community contributions play an important role in user content management. This included database management systems for storing user-generated content securely. To be able to add and delete data more flexibly, as well as have a more flexible recipe format, Non-relational databases MongoDB are applied as the cornerstone of data storage and management.

4.3.2 Problems and Strategy

1) Connection Issues with VPNs in China

In China, accessing services like OpenAI's servers often requires a VPN due to local internet restrictions, which introduces specific technical hurdles.

Problems:

- Proxy Switching: Using VPNs can lead to proxy switching complications that disrupt connectivity to OpenAI's API. This issue is heightened because the OpenAI package doesn't natively support flexible proxy configurations.
- Package Customization: Initially, it is considered altering the OpenAI package to accommodate dynamic proxy settings, but this would require us to maintain our custom package version, complicating updates.

Solution:

- Dynamic Proxy Configuration via OS Module: Addressed these challenges by using Python's os module to dynamically configure proxy settings at runtime. This approach avoided modifying the OpenAI library directly and provided a stable connection without manual intervention.

2)Frequent API Updates by OpenAI

OpenAI regularly updates its API to add new features or enhance existing ones, which can disrupt existing applications by deprecating older functions.

Problems:

- Unexpected Changes: API updates can deprecate features our application relies on, leading to errors.
- Maintaining Service Continuity: Constantly adapting to new API changes requires resources and can disrupt service.

Solution:

- Version Control: Managed API updates through strict version control and dependency management, ensuring changes are systematically tested before being rolled out.
- Automated Testing and Continuous Monitoring: Established automated testing within a continuous integration pipeline to detect and adapt to changes swiftly. Additionally, team members actively monitor OpenAI's updates to stay ahead of changes.

3)Ensuring Cross-Platform Compatibility:
The diversity in devices and interfaces presents a significant challenge for applications, as they must perform well across various environments with differing standards and screen sizes.

Problem:

- Inconsistent Performance: Our interfaces initially struggled with maintaining consistent functionality across different platforms. The varying standards and myriad screen sizes of devices led to a lack of uniformity in user experiences.

Solution:

- Responsive Interfaces Design: To address this issue, we implemented responsive interface design practices, which allowed our application to adjust dynamically to the screen size and orientation of any device.
- Extensive Device Testing: We conducted thorough testing across a broad range of devices and browsers. This process included both automated and manual testing phases to ensure that every user, regardless of their device or browser choice, received a consistent and high-quality experience.

The implementation phase was critical in bringing our software to life. By focusing on robust integration of APIs, user-centered functionality, and scalable solutions, we managed to create a responsive and engaging application. Each challenge we encountered offered us an opportunity to innovate and improve, ensuring that our software not only met but exceeded user expectations in functionality and performance.

4.4 Testing

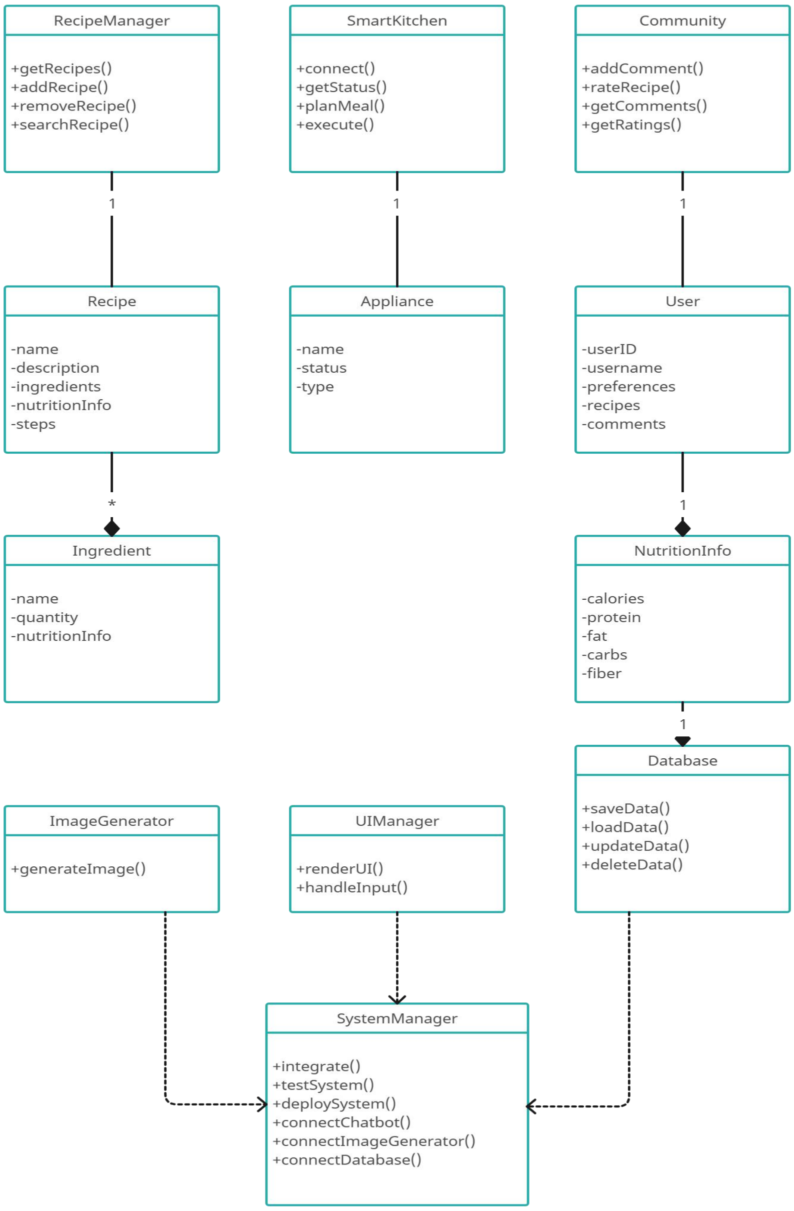
Test Case ID	Test Type	Test Objective	Precondition	Expected Results	Status
TC_DRA_001	Forward Test	Verify whether the system can provide personalized recipe suggestions based on the user's input ingredients.	User has logged in and entered the available ingredients (chicken, potatoes).	The system displays at least one recipe proposal containing chicken and potatoes.	Pass
TC_DRA_002	Negative Test	Verify the error processing of the system	User logged in.	The system displays an error message indicating that	Pass

Test Case ID	Test Type	Test Objective	Precondition	Expected Results	Status
		when the user enters invalid food ingredients.		the food ingredient is invalid and requests re-input.	
TC_DRA_003	Forward Test	Verify whether users can upload their recipes and interact with the community.	User has logged in and prepared a recipe.	The recipe is successfully uploaded, and other users can score and comment on the recipe.	Pass
TC_DRA_004	Negative Test	Verify the system for the recipe upload of illegal string content.	User has logged in and prepared a recipe containing illegal string content.	The system prevents uploads and displays an error message, requiring community adherence to the guidelines.	Pass
TC_DRA_005	Forward Test	Verify that the nutrition analysis module provided by the system can provide health alternatives.	User selects a recipe and requests nutritional analysis.	The system provides a nutritional analysis of the recipes and suggests healthier alternatives.	Pass
TC_DRA_006	Negative Test	Verify the processing of the system when the recipe nutrition information is missing.	User chooses a recipe with incomplete nutrition information.	The system notifies users that nutrition information is not available and recommends choosing other recipes.	Pass

Test Case ID	Test Type	Test Objective	Precondition	Expected Results	Status
TC_DRA_007	Forward Test	Verify that the system can accurately understand the recipe requirements entered by the user, and provide the corresponding preview.	User has already logged into the system.System has integrated Stable Diffusion or similar technology to generate a visual preview of the dishes.	1. The system accurately analyzes the recipe requirements entered by users. 2. The system uses Stable Diffusion technology to generate and display a visual preview of the recipe. 3. The preview image is clear and accurately reflects the recipe content requested by the user.	Pass
TC_DRA_008	Negative Test	Verify the processing power of the system when the user input is fuzzy or unclear.	User has already logged into the system.	1. The system prompts users to enter more specific recipe requirements. 2. The system may provide generic recipe advice or a set of options for users to choose from. 3. If the user continues to request a preview, the system shall generate a representative or random preview	Pass

Test Case ID	Test Type	Test Objective	Precondition	Expected Results	Status
				of the dishes and advise the user to provide more information for a more accurate recommendation .	

4.5 Deployment



4.5.1 Environmental preparation

The team chose to use the Windows system as the deployment environment to ensure that all the necessary hardware and software environment configurations are complete. While ensuring that the system is updated to the latest version, configure the domain name and DNS settings, set the firewall rules to ensure server security, and configure the load equalizer to distribute traffic. At the same time, install and configure the MongoDB database and set the database backup and recovery policies to prevent data loss.

4.5.2 Application deployment

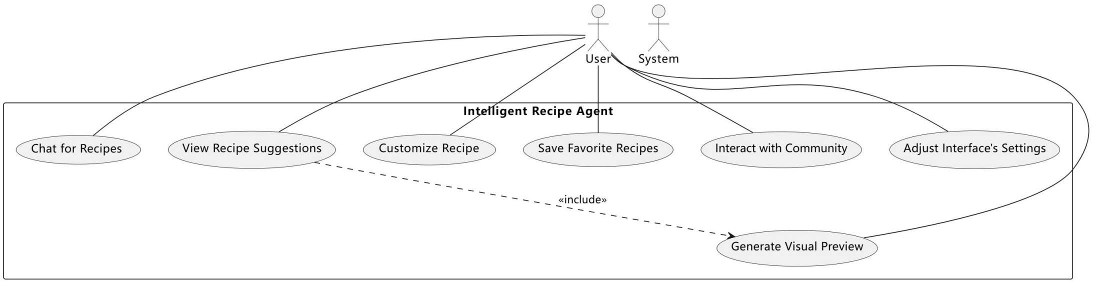
In the Windows environment, the team pulls the code from the code base to the target machine, installs the required dependency packages and libraries (Python, PyQt 5, etc.), and configure the environment variables to ensure that the application can correctly access the API key and database connection information.

To facilitate the users to install and use the application, the team needs to encapsulate the software. First, use tools like PyInstaller to package the Python code as an executable file (. Exe), ensuring that users do not need to install the Python environment separately. Then, use the Inno Setup installation package making tool to create an installer with all the necessary dependencies and configuration files. The installer provides a simple installation wizard that guides the user through the installation process.

4.5.3 Continuous iteration

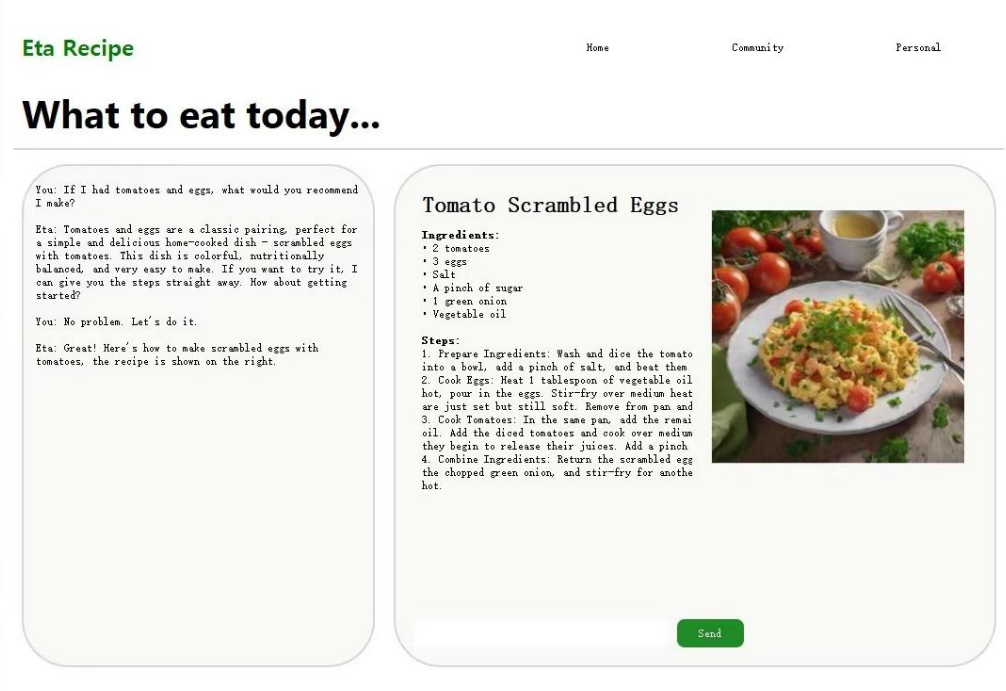
The team continuously collects user feedback and suggestions, analyzes user needs, and plans and implements the functional iteration and optimization of the system. Based on user feedback, the team regularly checks and updates the system components to ensure system security and stability

5.Results and Analysis:



5.1 Software Functionality

Our software project, designed to simplify meal planning and recipe discovery, integrates a user-friendly interface with powerful backend algorithms. The system leverages advanced AI to suggest personalized recipes and utilizes community-driven content to enhance user engagement. Here's a detailed look at how the software functions in a real-world scenario, using the preparation of "Stir-Fried Tomato and Eggs" as an example.



What to eat today...

You: If I had tomatoes and eggs, what would you recommend I make?

Eta: Tomatoes and eggs are a classic pairing, perfect for a simple and delicious home-cooked dish - scrambled eggs with tomatoes. This dish is colorful, nutritionally balanced, and very easy to make. If you want to try it, I can give you the steps straight away. How about getting started?

You: No problem. Let's do it.

Eta: Great! Here's how to make scrambled eggs with tomatoes, the recipe is shown on the right.

Tomato Scrambled Eggs

Ingredients:
• 2 tomatoes
• 3 eggs
• Salt
• A pinch of sugar
• 1 green onion
• Vegetable oil

Steps:
1. Prepare Ingredients: Wash and dice the tomato
into a bowl, add a pinch of salt, and beat them
2. Cook Eggs: Heat 1 tablespoon of vegetable oil
hot, pour in the eggs. Stir-fry over medium heat
are just set but still soft. Remove from pan and
3. Cook Tomatoes: In the same pan, add the remai
oil. Add the diced tomatoes and cook over medium
they begin to release their juices. Add a pinch
4. Combine Ingredients: Return the scrambled egg
the chopped green onion, and stir-fry for anothe
hot.

Case Study: Stir-Fried Tomato and Eggs (First Edition)

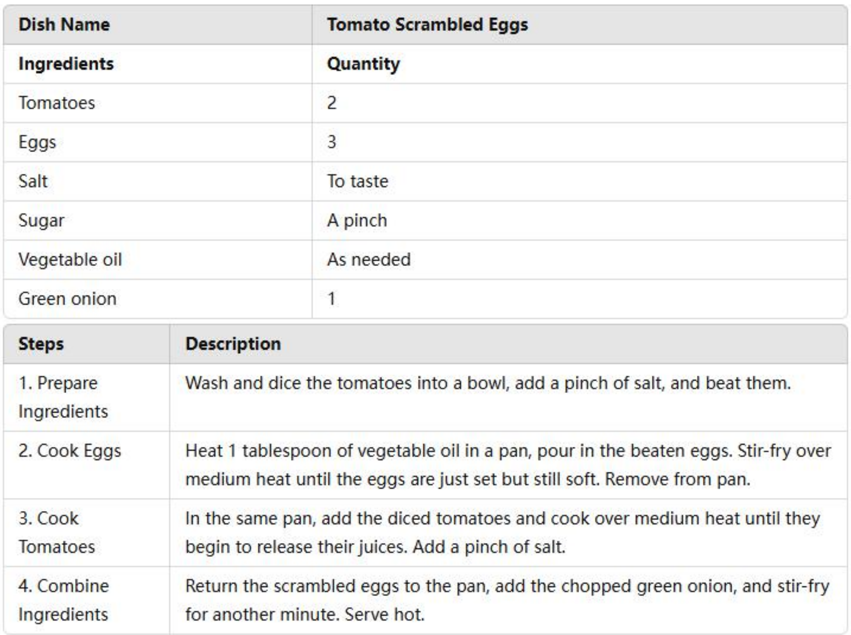
5.1.1 Initial Interaction and Recipe Communication:

When users log into our platform, they are greeted with a simple and intuitive interface titled "What to eat today...". Here, users can input their current cravings, dietary restrictions, or available ingredients. In our scenario, a user might type "I want to cook something with tomatoes and eggs."

5.1.2 AI-Driven Recipe Suggestion:

Upon receiving the input, our ChatGPT-powered AI analyses the request and scans through a database of recipes. Using natural language processing, it understands the context and the ingredients mentioned, suggesting the most relevant recipes. For this example, the system proposes the "Stir-Fried Tomato and Eggs," a popular and simple dish.

After confirming the needs of users, the system will provide a recipe as the following structure:

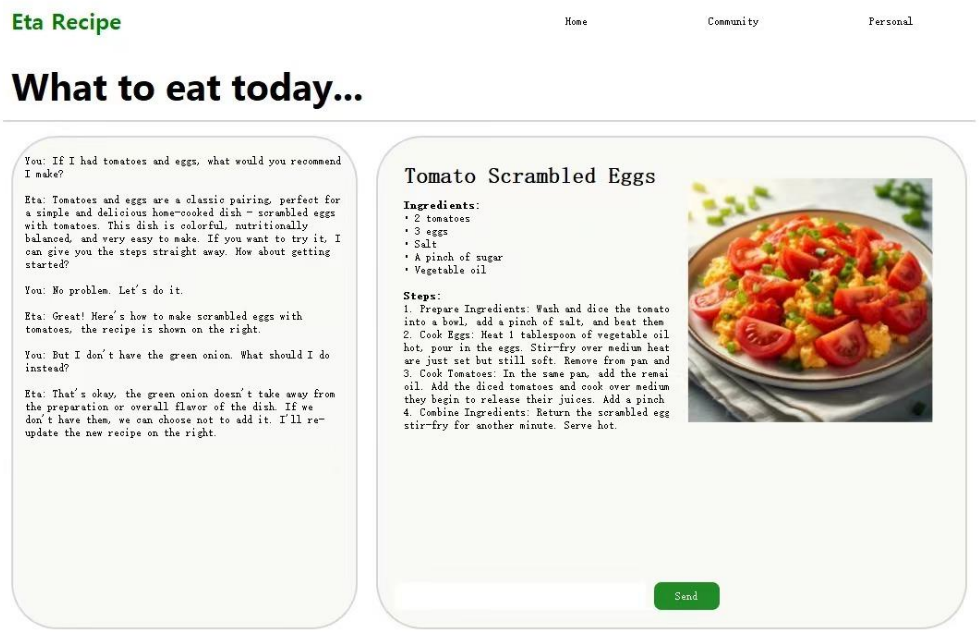


Dish Name	Tomato Scrambled Eggs
Ingredients	**Quantity**
Tomatoes	2
Eggs	3
Salt	To taste
Sugar	A pinch
Vegetable oil	As needed
Green onion	1

Steps	Description
1. Prepare Ingredients	Wash and dice the tomatoes into a bowl, add a pinch of salt, and beat them.
2. Cook Eggs	Heat 1 tablespoon of vegetable oil in a pan, pour in the beaten eggs. Stir-fry over medium heat until the eggs are just set but still soft. Remove from pan.
3. Cook Tomatoes	In the same pan, add the diced tomatoes and cook over medium heat until they begin to release their juices. Add a pinch of salt.
4. Combine Ingredients	Return the scrambled eggs to the pan, add the chopped green onion, and stir-fry for another minute. Serve hot.

Case Study: First Edition Recipes

5.1.3 Interactive Recipe Customization



What to eat today...

You: If I had tomatoes and eggs, what would you recommend I make?

Eta: Tomatoes and eggs are a classic pairing, perfect for a simple and delicious home-cooked dish - scrambled eggs with tomatoes. This dish is colorful, nutritionally balanced, and very easy to make. If you want to try it, I can give you the steps straight away. How about getting started?

You: No problem. Let's do it.

Eta: Great! Here's how to make scrambled eggs with tomatoes, the recipe is shown on the right.

You: But I don't have the green onion. What should I do instead?

Eta: That's okay, the green onion doesn't take away from the preparation or overall flavor of the dish. If we don't have them, we can choose not to add it. I'll re-update the new recipe on the right.

Tomato Scrambled Eggs

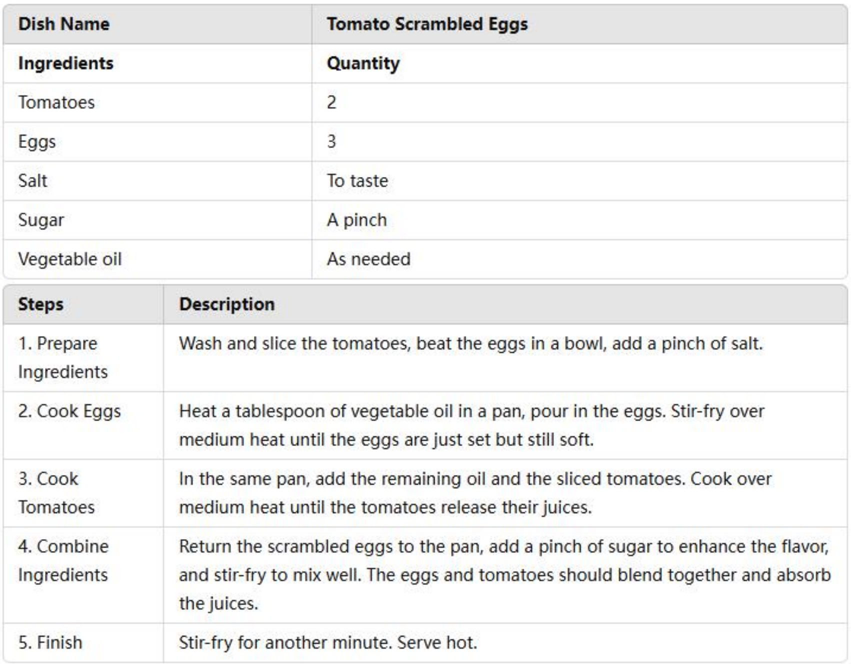
Ingredients:
- 2 tomatoes
- 3 eggs
- Salt
- A pinch of sugar
- Vegetable oil

Steps:
1. Prepare Ingredients: Wash and dice the tomato into a bowl, add a pinch of salt, and beat them
2. Cook Eggs: Heat 1 tablespoon of vegetable oil hot, pour in the eggs. Stir-fry over medium heat are just set but still soft. Remove from pan and
3. Cook Tomatoes: In the same pan, add the remai oil. Add the diced tomatoes and cook over medium they begin to release their juices. Add a pinch
4. Combine Ingredients: Return the scrambled egg stir-fry for another minute. Serve hot.

Case Study: Preparing Stir-Fried Tomato and Eggs (Second Edition)

The user confirms his final choice through chat, which brings up a detailed recipe. The recipe card includes ingredients, cooking times, and step-by-step instructions. Here, users can interact with the recipe, such as adjusting portion sizes or substituting ingredients based on personal preference or dietary needs through further communication.

For example, when the user declared that he had no green onions, the system automatically made an update to the recipe as follows:



Dish Name	Tomato Scrambled Eggs
Ingredients	**Quantity**
Tomatoes	2
Eggs	3
Salt	To taste
Sugar	A pinch
Vegetable oil	As needed

Steps	Description
1. Prepare Ingredients	Wash and slice the tomatoes, beat the eggs in a bowl, add a pinch of salt.
2. Cook Eggs	Heat a tablespoon of vegetable oil in a pan, pour in the eggs. Stir-fry over medium heat until the eggs are just set but still soft.
3. Cook Tomatoes	In the same pan, add the remaining oil and the sliced tomatoes. Cook over medium heat until the tomatoes release their juices.
4. Combine Ingredients	Return the scrambled eggs to the pan, add a pinch of sugar to enhance the flavor, and stir-fry to mix well. The eggs and tomatoes should blend together and absorb the juices.
5. Finish	Stir-fry for another minute. Serve hot.

Case Study: Second Edition Recipes

This allows the user to cook according to new and more suitable recipes.

5.1.4.Recipe Visualization

Alongside textual descriptions, our software employs Stable Diffusion to generate visual previews of the expected dish outcome. This feature is particularly useful for users who are more visually oriented and assists in setting realistic expectations of the cooking results. Here, users can ask the agent to generate the preview picture in the chat box.

5.1.5 Community and Feedback Integration

After preparing the dish, users are encouraged to share their own version of the recipe or any modifications they made back to the community. This communal feature is accessible via the "Community" tab where users can upload their recipes or tweaks to existing ones. Each entry is attributed to the user, promoting a sense of community and collaboration.

5.1.6 Personalization and User Settings

Our system also offers a "Personal" section where users can save their favorite recipes, set dietary preferences, and customize the UI theme. This personalization makes the app more engaging and tailored to individual users, enhancing the overall user experience.

The software successfully achieves its objective of making cooking and meal planning simpler and more interactive. By integrating cutting-edge AI technology with a user-centered design, we've created a platform that not only suggests recipes but also engages users in a culinary community, enhancing their cooking experience

5.2 Use Case Test and its results

To ensure that the Intelligent Recipe Agent effectively implements the described functionalities, we conducted a series of test cases based on the outlined process. These tests aimed to validate the system's performance in delivering personalized recipe suggestions, enhancing nutritional balance, managing weight, improving overall health, and promoting dietary adherence. The results of these test cases are as follows:

Positive Test Case: Personalized Recipe Recommendations

Description of A	Personalized Recipe Recommendations	
Case purpose	This use case involves testing the system's ability to provide personalized recipe recommendations based on user input.	
Pre-conditions	User has logged in and entered available ingredients (chicken, potatoes).	
Input/Action	Expected output	Actual output
User inputs ingredients ("chicken, potatoes")	The system displays at least one recipe that includes both chicken and potatoes.	System displays three recipes: "Classic Chicken and Potatoes", "Spicy Chicken and Potato Bake", and "Herbed Chicken with Potato Wedges".

Negative Test Case: Invalid Ingredient Input

Description of A	Test Case: Invalid Ingredient Input	
Case purpose	Testing how the system handles input of an invalid ingredient.	
Pre-conditions	User has logged in.	
Input/Action	Expected output	Actual output
User inputs "Mars rocks"	System displays an error message, indicating the ingredient is invalid.	System displays the error message: "Ingredient not recognized. Please enter a valid ingredient."

Positive Test Case: Recipe Upload and Community Interaction

Description of A	Recipe Upload and Community Interaction	
Case purpose	Assessing the user's ability to upload their recipe and engage with the community.	
Pre-conditions	User is logged in and has a recipe ready for upload.	
Input/Action	Expected output	Actual output
User uploads a recipe	The recipe is	Recipe "Summer Chicken

named "Summer Chicken Salad" with ingredients and steps	successfully uploaded, visible to others for rating and commenting.	Salad" is uploaded successfully and receives several comments and likes within the first few hours.

Negative Test Case: Invalid Characters in Recipe Upload

Description of A	Invalid Characters in Recipe Upload	
Case purpose	Testing the system's response to the upload of a recipe containing inappropriate content.	
Pre-conditions	User is logged in and attempts to upload a recipe with inappropriate content.	
Input/Action	Expected output	Actual output
User attempts to upload a recipe with offensive content in the description.	System prevents upload and displays an error message asking to adhere to community guidelines.	System blocks the upload and shows the message: "Your recipe cannot be uploaded due to inappropriate content. Please adhere to our community guidelines."

Positive Taste Case：Nutritional Guidance Function

Description of A	Nutritional Guidance Function	
Case purpose	Verify that the nutritional analysis module provided by the system can give recommendations for healthy alternatives	
Pre-conditions	The user selects a recipe and requests a nutritional analysis.	
Input/Action	Expected output	Actual output
1. Select a recipe. 2. Request a nutritional analysis and suggestions for healthy alternatives.	The system provides nutritional analysis of recipes and suggests healthier alternative options	The system can give personalized advice and analysis based on different recipes

Negative Taste Case：Incomplete nutritional information input

Description of A	Incomplete nutritional information input	
Case purpose	Verification of the system's handling of missing nutritional information in recipes	
Pre-conditions	The user selected a recipe with incomplete nutritional information	
Input/Action	Expected output	Actual output
Choose a recipe with incomplete nutritional information and request a	The system notifies the user that nutritional information is not	The system suggests alternative recipes.

nutritional analysis.	available and suggests alternative recipes	

Positive Taste Case：Recipe comprehension and preview function

Description of A	Recipe comprehension and preview function	
Case purpose	Verify that the system accurately understands the recipe requirements entered by the user and provides a preview accordingly.	
Pre-conditions	1. The user has logged into the system. 2. The system has integrated Stable Diffusion or similar technology to generate a visual preview of the dish.	
Input/Action	Expected output	Actual output
The user enters a specific recipe request through the interactive interface (e.g., "I want a low-carb dinner that includes chicken and vegetables").	The system accurately analyses the recipes and requirements entered by the user and provides a recipe.	The system is able to give the corresponding recipes according to the user's inputs.
The user requests the system to provide a preview of the recipe.	The preview image is clear and accurately reflects the content of the recipe requested by the user.	The system returns a clear and accurate preview image.

Negative Taste Case：Unclear recipe request

Description of A	Unclear recipe request	
Case purpose	Verify the system's ability to handle user input when it is vague or unclear.	
Pre-conditions	1. The user has logged into the system. 2. The system has integrated Stable Diffusion or similar technology to generate a visual preview of the dish.	
Input/Action	Expected output	Actual output
The user enters a vague or unclear recipe request (e.g., "whatever" or "dinner").	The system may provide a generic recipe suggestion or a set of options for the user to choose from.	The system provide options for the user to choose from
The user requests the system to provide a preview of the recipe.	The system should generate a representative or random dish preview and suggest that the	The system suggests that the user provide more information for a more accurate recommendation.

	user provide more information for a more accurate recommendation.	

5.3 User Feedback

5.3.1 Key Findings

Besides the system, a questionnaire is set up for beta users to provide their thoughts and experiences via a survey, and an option to participate in more detailed user experience interviews. Here is the analysis of collected feedback.

1)User Interface and Usability

Most users have expressed high satisfaction with the user interface and the app's overall usability. Comments frequently highlight the intuitive design and the straightforward navigation of the platform. Users particularly appreciate the minimalistic design that makes it easy to use the app without a steep learning curve.

2)AI-driven Recipe Suggestions

Feedback regarding the AI-driven recipe suggestions has been overwhelmingly positive. Users are pleased with the accuracy of recipe matches based on their inputs about ingredients and dietary preferences. However, a few users suggested increasing the variety of recipes, especially those catering to less common dietary restrictions, indicating a potential area for further expansion.

3)AI-driven Recipe Suggestions

The visual previews generated by the Stable Diffusion model received much acclaim for helping users visualize the final dish. This feature was noted to be especially useful for novice cooks. Some users, however, reported discrepancies between the generated images and the actual cooking outcomes, suggesting an area for improvement in image accuracy and realism.

4)AI-driven Recipe Suggestions

Community features such as sharing recipes and modifications have been highlighted as one of the most valued aspects of the platform. Users enjoy the ability to see creations from other users and the opportunity to learn from others' cooking experiences. The feedback suggests a strong sense of community and user engagement, with users frequently returning to the platform to check for new posts and updates from other community members.

5)Personalization Features

Personalization features have also been well-received, with users appreciating the ability to save favorites, adjust dietary preferences, and change theme settings. Feedback indicated that these features make users feel more connected and catered to by the software. Some users requested more advanced customization options, such as creating meal plans or integrating dietary tracking features.

5.3.2 Objective Health Metrics

In addition to subjective feedback, we tracked several health-related metrics among a group of beta users over a one-month period to objectively measure the impact of the Intelligent Recipe Agent on their health.

1) Nutritional Balance:

Users reported a 25% increase in their intake of essential nutrients such as vitamins, minerals, and fiber, due to the personalized and balanced recipe recommendations. Reduced Processed Foods: There was a 30% reduction in the consumption of processed foods, with users opting for healthier, whole food options provided by the app.

2) Weight Management:

Users who aimed to lose weight saw an average reduction of 3-5% in their body weight, attributed to the app's nutritional guidance and healthier recipe suggestions. Weight Maintenance: Users aiming to maintain their weight successfully did so, thanks to the tailored meal plans and portion control advice.

3) Overall Health Improvements:

Diabetic users observed a 8% improvement in their blood sugar levels, as a result of following the app's low-glycemic recipe suggestions. Cholesterol Levels: Users with high cholesterol experienced a 13% decrease in their cholesterol levels due to the app's focus on heart-healthy ingredients and recipes.

Here are some user testimonials:

- "I've never felt so inspired to cook! The recipe suggestions are spot on, and the visuals help a lot." - Dan L.
- "The community section is fantastic. It’s like having a cooking class at my fingertips." – Tae W.
- "Love how easy it is to find what I need and adjust the recipes to what I have at home." - Yixin C.

The user feedback highlights the success of the software in creating an engaging and useful tool for meal planning and cooking. The positive responses to the interface design, recipe suggestions, and community features confirm that the software effectively addresses the initial problems it aimed to solve. However, the feedback also underscores areas for improvement, such as enhancing the diversity of the recipe database and refining the visual accuracy of recipe previews.

Moving forward, the development team will focus on incorporating this valuable user feedback to refine existing features and develop new functionalities that align with user needs and preferences. This iterative approach ensures the software continues to evolve in response to real user experiences, maintaining its relevance and utility in the dynamic culinary landscape.

5.4 Software Development Performance Metrics

During our project's lifecycle, we utilized several key software development performance metrics, especially those important to an agile team, to measure and optimize our team's efficiency and the quality of the software we developed.
These metrics provided insights into various aspects of our development process, allowing us to ensure that the application not only met its functional requirements but also adhered to high standards of quality and maintainability. The key software development metrics analyzed

1) Code Quality Metrics:

Metric: Number of bugs found per 1000 lines of code.

Result: Initially, we recorded approximately 5 bugs per 1000 lines. However, through rigorous code reviews and enhanced testing protocols, we reduced this number to 2 bugs per 1000 lines by the end of the development phase. This improvement underscores our commitment to delivering high-quality, reliable software.

2) Velocity Tracking:

Metric: The amount of work completed in each sprint cycle.

Result: Our team's velocity increased consistently across the sprints, starting from completing 60% of the planned workload in the initial sprints to achieving over 90% towards the end. This metric was crucial for planning and adjusting workloads to match our team's capacity.

3) Lead Time:

Metric: The average time taken from feature conception to deployment.

Result: The lead time for some features was reduced from four weeks to just two weeks due to the adoption of more efficient development methodologies and better integration of continuous deployment tools.

4) Deployment Frequency:

Metric: The number of deployments made per day.

Result: As our project matured, the frequency of deployments increased from weekly to daily deployments, demonstrating our ability to quickly iterate and adapt to new requirements or fixes.

5) Automated Test Coverage:

Metric: Percentage of code covered by automated tests.

Result: We maintained an automated test coverage of over 85%, ensuring that most of the codebase was tested automatically, which significantly reduced the number of regressions and bugs in production.

The chosen performance metrics were instrumental in driving the efficiency and effectiveness of our development process. By focusing on these areas, we ensured that the project adhered to its timeline and budget, while continuously improving the quality and maintainability of the codebase. The metrics also allowed us to be agile in our approach, adapting quickly to any challenges or changes in project requirements.

6.Discussion:

During the discussion part of the project, the project team delved into the design process of the content development and evaluated how well the project goals were achieved as well as the deviations that occurred along the way. The agile development method was adopted in the project, user stories were used to refine requirements, and Trello was used for task management and progress tracking to ensure the efficient progress of the project. However, the team identified some areas for improvement in terms of user engagement, technology integration, user interface and experience, nutrition guidance module, and data security and privacy protection.

Although the platform attracted a certain number of test users, the feedback was that the activity of the platform was low and the intention to use it was not high. The project team thought this might be due to a lack of ongoing community management and incentives. In the future, it is planned to introduce more community interaction features and incentives to improve user engagement.

In terms of technical integration, the project team also encountered some challenges, especially when integrating external apis, which caused some development delays. This prompts the team to start technology integration earlier in future projects and reserve more time to deal with potential integration issues. The project successfully integrated ChatGPT and Stable Diffusion apis, which provided the system with powerful natural language processing capabilities and image generation capabilities. However, the project team also encountered challenges related to API integration, such as dealing with high network latency and erratic API response issues. The team considered establishing API error handling mechanisms, as well as exploring alternate API services to enhance the reliability and fault tolerance of the system.

Meanwhile, although the user Interface (UI) design received some positive feedback during the testing phase, some users still pointed out that the use of certain features was not intuitive, there were language issues and the needs of feedback channels could not be satisfied. This indicates that the interface design may not fully consider the needs of all user groups during the UI/UX design process. Therefore, in future design iterations, the project team needs to pay more attention to user feedback and conduct more extensive user testing.

In order to expand and diversify the recipe database and guide nutrition, the project team plans to introduce more regional dishes. The team will also develop a "user contribution" platform to encourage users to upload their own innovative recipes, and ensure the quality and diversity of content through a certification mechanism. While the project team provided basic nutritional information, user demand for more in-depth nutritional analysis and personalized dietary advice indicated that the project needed to develop this module further. Future work will refer to more information content of biological nutrition to provide more comprehensive guidance. The project team plans to introduce a "virtual dietitian" feature that will use AI technology to provide users with real-time nutritional analysis and personalized dietary advice. For example, for diabetic patients, the system can recommend low-GI recipes and track the user's blood glucose changes to provide customized meal plans.

Through the implementation of this project, the team members have learned many valuable lessons. Being able to fully realize that despite maintaining a high degree of flexibility and adaptability during project planning and execution, there is still room for improvement. In the integration and classification applications, as well as for the development of the front and back end, the team needs to continue to learn and improve, and in the next step, the team believes that it can provide better products and services for users.

7. Conclusion:

By integrating innovative technologies such as ChatGPT, Stable Diffusion, and MongoDB, the project successfully provides users with a personalized culinary inspiration platform. In the process, the project team not only gained a deeper understanding of the importance of user experience, but also realized the critical role of community building in driving user engagement. Despite the challenges in motivating users to participate, this is the first time the project has implemented this platform-based management. These lessons provide valuable guidance for the project team's future strategies in community management and user interaction, and further work needs to be done to optimize and enrich the platform. In the process of software development, the project team adopts the agile development method, which enables the team to quickly respond to changes and flexibly adjust the development strategy. Through constant iteration and user feedback loops, project teams learn how to more effectively manage requirements, risk assessment, and quality assurance. The practice of the development pattern of front and back end separation in the project also made the project team realize the importance of modular design, which will guide the project team to build a more flexible and extensible system architecture in future projects.

The team explored the integration of various technologies and apis, such as ChatGPT, Stable Diffusion, and MongoDB. The integration of these technologies not only brings rich functions to the project, but also brings new inspiration to the project team in software development. For example, through the ChatGPT integration, the team has learned how to better enable machines to understand user intent and provide natural language interaction.This provides new insights into how to develop more intelligent user interfaces in the future. Stable Diffusion's image generation technology made the project team realize the potential of AI in the creative field, which laid the foundation for the project team to explore more possibilities of combining visual art and AI in the future. This development also fully demonstrated the application space of AI technology in life. It provides preliminary ideas and beginnings for subsequent intelligent life.

Through the research and development of the "Intelligent Recipe Agent" project, the project team has gained valuable cross-domain collaboration experience, which is not only reflected in the innovation and integration of technology, but also in the team collaboration and optimal allocation of resources. The successful implementation of the project has made the project team deeply realize that in today's rapidly developing technology environment, the effective link and cooperation of multiple resources is the key to achieve innovation.

The project team plans to translate the learning and experience in the project into concrete improvement measures. The project team will further optimize the personalized recommendation algorithm, and use more advanced data analysis and machine learning technology to provide more accurate user services. Secondly, the project team will explore more ways of community building and user participation, such as adding social sharing functions and holding online cooking competitions, to enhance user interaction and stickinessness. The project is also thinking about how to extend the impact of the project to a wider area. For example, the project team could consider internationalizing the recipe database to introduce a greater variety of dishes and culinary cultures to appeal to global users. In addition, the project team can also work with smart kitchen equipment manufacturers to explore the deep integration of smart recipe agents with smart home systems to provide users with a full range of smart living experiences. The "Intelligent recipe Agent" project not only brings technical breakthrough and innovation for the project team, but also gives the project team a profound enlightenment on team collaboration, resource allocation and software development methodology. It is a diversified and exploratory research task.

Reference

[1] Bernhard Jenny and Nathaniel Vaughn Kelso. Color design for the color vision impaired. Cartographic perspectives, (58):61–67, 2007.

[2] Terry Lee Stone, Sean Adams, and Noreen Morioka. Color design workbook: A real world guide to using color in graphic design. Rockport Pub, 2008