

RecipeFusion: Recipe Recommendation & Generation System



**Tayyaba Ejaz
Laiba Hassan**

Supervised By
Dr. Sadia Ashraf

*Submitted for the partial fulfillment of BS Computer Science to the Faculty of
Engineering & Computer Science*

DEPARTMENT OF COMPUTER SCIENCE

NATIONAL UNIVERSITY OF MODERN LANGUAGES

ISLAMABAD

December, 2025

ABSTRACT

The growing demand for intelligent digital products in our everyday lives has sped up the aparting of AI-powered recommendation systems in different fields. RecipeFusion is a Generative AI-enabled recipe suggestion and meal preparation tool that helps users make better and smarter food choices. Unlike traditional recipe platforms based on simple filters or keyword-based search functions, RecipeFusion helps to overcome important limitations such as limited customization of the diet, lack of awareness about nutritional requirements, and inability to utilize the available ingredients.

The system has several different basic options such as AI-created recipes given names or ingredients, search for recipes based on ingredients, dietary preferences, and a smart meal planner with automatic shopping list creation. Moreover other functions like adding reviews, and recipes generating by GenAI, boosted by user's engagement. The frontend is developed with React and Tailwind CSS for a new and responsive user interface, while the backend and database layer is developed using Node.js and MongoDB. Generative AI models & external API to provide intelligent recipes generation & nutritional insights Scalability, reliability and fast performance are secured by deployment into Vercel.

Black-box techniques were employed to test the usability, functionality and stability of the systems. Evaluation results showed that RecipeFusion achieves its target goals including accurate recommendations, relevancy of recipes, and smooth user experience over recipes and recipes modules.

Other than effectiveness our system relies on availability of active internet connections, as well. In future work, some of these enhancements to the Pathfinder AI powered chat include expanding the collection of recipes, the integration of more advanced AI models to provide better individualization, the potential of building a mobile application and providing multilingual support for AI technology to increase access and usability across all countries.

CHAPTER 1

INTRODUCTION

1.1. Introduction:

IT technologies force people to alter their attitudes to their every-day life, and meal planning is no exception. However, the majority of traditional type of recipe websites lack much customization since they fail to take into account one-on-one diets and also fail to offer the chance to align suggestions with those ingredients available or the target markets health objectives. This creates a generic user experience, and this will not serve the requirements of those with a distinct dieting style - i.e. whole-food or plant-based users or sustainable and health-conscious users. These limitations create work and efficiency inefficiencies, and extra effort to those users who desire to receive customized recipe recommendations that would be resource, as well as preferences-based.

This gap is addressed through the intelligence Recipe fusion: intelligent recipe generation and recommends system which offers an intelligent easy to use web site that narrows meal recommendations based on user input, includes dietary restrictions, access to ingredients, and health goals, among others. It is a system based on artificial intelligence and data-driven analysis and provides both personalized and healthy recipes and gives detailed nutritional information to enable making informed choices. By making planning meals more lifestyle-based and more intuitive, RecipeFusion will support healthier lifestyles, reduce wastage of food, as well as, promote a sustainable approach to nutrition.

1.2. Motivation:

RecipeFusion has been established due to the growing market need of the smart, personalized and health conscious online meal planning services. The traditional recipe services are unlikely to work in the world where the awareness of health and sustainability is becoming a focal point of daily life, especially to its people that may follow some eating pattern or a lifestyle. These websites tend to offer the amount of content that is relatively close in scale, so that it is difficult to locate meals that would suit the nutritional goals of patients, the ingredients that they possess and their own nutritional constraints. This is so because planning healthy meals that fit in their schedule has become so frustrating, and time consuming to the users.

According to a report by Statista, more than 70 percent of consumers believe they desire to eat better, and a significant part of them cannot locate the recipes, which can assist them in satisfying their desires in a viable and personalized way. According to this dichotomy there is

an apparent need of complex mechanisms in comparison to naked search features and provides dynamic and customer-centric meal planning services.

There is also the recent global trend of health awareness and the shift toward sustainable living which has led to the wish to utilize systems characterized by a high level of Intelligence in order to encourage not only a healthy eating habit, but the ability to act in a manner that is sustainable to the environment. The long term potential of delivering smart and AI-driven digital solutions to this market segment is vast by all means since by 2027 the world market of health and wellness food will have reached more than one trillion dollars.

RecipeFusion will seal this gap because it will offer a new service that gives a search based on ingredients with a specific diet based on nutritional values, and a multitude of diets. It gives meal recommendations via AI, which allows a person to make informed, nutritious decisions and also reduce food waste and promote safer consumption practices through the application of generative AI. In that regard, one can say that RecipeFusion can serve the purpose of not only benefit the user in the context of cooking as such, but also serve to achieve other broader goals, including the betterment of the health of the population and environmental protection.

1.3. Problem Statement:

The traditional recipe finding apps are not endorsing the growing necessity of a customized, health conscious and effective food planning. The users also have a tendency to use generalized search filters or a system that works according to the specific search by keywords without taking into account the particular dietary restrictions, ingredients on hand, and achievement of particular nutrition goals. Disconnection like this leads to frustration, since users are not able to locate meals, which actually fit and match their preferences and lifestyles.

According to a survey made by the International Food Information Council, over 63 percent of consumers are trying their best to obtain the information related to healthy eating, but they are not even aware of where the food meeting their requirements could be found. Furthermore, the persons, who subscribe to special diets, such as vegetarian or allergen-free, are even more disadvantaged with using the regular platforms. Absence of inherent nutritional literacy and customization can be translated to wasteful research that is translated to unplanned planning, improper nutrition and food wastage. These problems show that there

is a need to find a solution to offer smart recipe recommendations that could be based on the input of the user in addition to encouraging informed, healthy, and sustainable diet practices.

1.3.1 Proposed Solution:

The RecipeFusion is an intelligent and customized means to locate a recipe due to the availability of the artificial intelligence platform and user-specific dietary practices.

The user can use the system to input on the items available in his or her pantry or fridge. Can choose filters for cuisine type like vegan, vegetarian, and so on. Moreover user can apply the filter for difficulty level and maximum cooking time too. Have suggested pre-made recipes to them. Look at the details of the nutritional value of food in each recipe. Integrate an inbuilt meal planner to plan the meals of the week.

By making meal planning easier and facilitating informed, healthy, and sustainable food decisions of individuals due to the inclusion of convenience, individuality, and dietary awareness, RecipeFusion will make the development of a meal for everyone more convenient.

1.3.2 Objectives:

The main goals of project Recipe Fusion are; To create a recipe recommendation system that will provide users with personalized meal recommendations using their preferences and the ingredients they have. To adopt AI-based filtering of dietary requirements including vegan, whole-food plant-based diets, etc. To show the information about calories to the user so that he can make informed decisions. To adopt a meal planner that will help to plan a weekly meal in an easy way. To promote and encourage new ideas for eating and minimizing the wastage of food by generating sustainable plans.

1.3.3 Scope of the Project:

The project aims at the creation of an online site that would improve the meal planning process by offering custom-made recipes. Through the assistance of gen AI, users will get customized meal recommendations according to their diet and accessible food supplies. The system will provide an interactive and smooth

experience, which is coupled with the integration of tools simplifying meal preparation and advanced healthy eating habits.

1.4. Goals and Objectives:

RecipeFusion aims to create a smart recipe generator that is web-based and can help to plan meals based on the unique preferences and dietary needs of users and help them with the selection of recipes, which can be made using the ingredients they have available. The platform will focus on the objectives of developing a convenient and health-conscience and interesting experience that will stimulate the users to make informed and sustainable food choices.

The specific objectives are; To allow users to get suggestions of recipes using already available ingredients. To offer dietary filtration services that can accommodate different types of preferences like vegan, gluten-free, whole-food plant-based diets, and so on. To add the nutritional details and assist with meal planning and an inbuilt grocery list maker. To add an gen-AI to provide interactive guidance and recipes discovery. To provide user account functions like registration, profile management and personalized history. To enhance and promote healthy food practices by minimizing food waste by matching recipes efficiently. To provide a clean, intuitive and user friendly interface to increase user satisfaction.

1.5. Scope of the study:

RecipeFusion is the scope of development of the web based solution providing a smart and tailored recipe suggestions and effective meal planning. The platform will give the users option to; Find recipes with the help of an intuitive interface basing on existing ingredients. Filters on searching: Type of Cuisine, max cook time, difficulty level and diet Create personal recipes using GenAI, depending on the ingredients/query typed by the user. Get calorie count on each recipe for the nutritional point of view. Create and maintain user accounts to have a personal experience in terms of saved preferences and history. Add reviews to the recipes which users like or dislike

1.6. Process model:

Agile process model is adopted in the development of RecipeFusion because of its user-friendly, flexible and iterative nature. RecipeFusion will be a smart and customized

recipe proposal and creation engine that will adjust to personal dieting requirements, accessible ingredients, and dietary objectives. These changing and varied user needs require development approach that promotes constant development and feedback responsiveness - and Agile is the best option.

Agile provides the opportunity to develop and implement the most important functions fast and in stages, including ingredient-based search, personalized recommendations, and nutritional analysis. This iterative quality enables the development team to develop, test and perfect functionalities in small short sprints. The flexibility of Agile makes sure that the improvements or amendments can be implemented in due time without any slowdown of overall workflow as more information is gathered through the use of user testing and feedback as new information is presented.

Furthermore, Agile encourages effective team work between cross-functional teams comprising of UI/UX designer, front and back-end developer, and quality assurance tester. The teamwork is essential to get a seamless user experience and technical stability of the project “RecipeFusion”. Regular sprint reviews and retrospectives can be used to ensure that the system is developed according to the expectations of the users and the technological best practices.

Essentially, Agile would enable the dynamic and changing environment of RecipeFusion to enable it to become a powerful user-oriented tool that will encourage healthy and sustainable eating as well as keep on evolving according to the interactions with real world users.

1.7. Nature of the project:

It is a web-based nature of the project and is interested in personalized recipe recommendations and meal planning in the health and wellness area. The project will combine the GenAI, user preference filters, and convenient UI elements to provide an interactive and smart platform to the modern, health-conscious users.

The most important aspects of the platform are Smart recipe discovery, where the user inputs the ingredients, dietary restrictions, and health goals, interactive recipe pages, which

give the cooking instructions, nutritional breakdown, and healthy and sustainable meals preparation tips. The features of user engagement, like favorites recipes, the possibility of leaving a review, and following the personal search to improve further recommendations.

The backend is done using Node.js with Express.js and for database MongoDB. It gave us a best and efficient framework to manage the user's data, their reviews added, search histories and all. For security purpose JWT-based authentication is done with role based access control.

1.8. Overview/Organization of the Project:

The report is organized in the following way:

Chapter 1: This is an introduction to the project which explains the motivation of the project, the problem statement, goals, and objectives, scope, and nature of the project.

Chapter 2: Talks about the background knowledge that is relevant, the previously existing systems and technologies and how they are limited in the context of the proposed solution.

Chapter 3: Stipulates the specifications of the system in terms of both functional and non-functional requirements, that is, what the system should accomplish and what technical criteria it should fulfill.

Chapter 4: The chapter pays attention to the system design, such as architectural views, database modeling and user interface designs.

Chapter 5: Outlines the implementation process, algorithms, modules and technologies that were utilized in the creation of the platform.

Chapter 6: Presents results and testing processes and validation of the system, as well as performance analysis.

Chapter 7: This is the final chapter of the report that summarizes the success of the project, the limitations and recommendations on how to improve the project.

CHAPTER 2

EXISTING SYSTEM

2.1 Introduction:

The chapter covers the history and current system regarding recipe recommendation and meal planning solutions. It will create an insight regarding significant concepts, technologies, and constraints of existing solutions, making the applicability and innovation of RecipeFusion relevant to these shortcomings.

2.2 Explanation of important constructs of Application Domain

2.2.1 Personalized Recipe Recommendations:

Personalized recipe sites entail providing meal recommendations personalized to a specific user depending on his or her dietary inclination and health goals and the latest search history. The systems increase user experience through promoting relevant time saving choices as well as decreasing the necessity of a manual decision of meal planning.

2.2.2 Ingredient-Based Search:

The application has a centralized system, which enables users to enter available ingredients and get customized recipes. This construct justifies a concept of food waste minimization and simplification of the cooking process using the food already available to users.

2.2.3 Feedback and UI:

Users are able to write down favorites, leave reviews and see saved meal plans. Not only does this construct enhance the user engagement and retention, but also, the system can use such feedback loops and patterns of usage to refine future suggestions.

2.2.4 Dietary Preference Filters:

The system also features filtering features enabling one to reduce the number of recipe results depending on the requirements of a particular diet. This is inclusive and assists in making sure that the user gets pertinent results that are specific to their needs.

2.2.5 Proposed System Key Features:

The following are the key features that will be provided in the proposed platform:

- Recipe recommendations and searching on the available ingredients
- Vegan, vegetarian, gluten free and other similar filters
- Custom recipes which are based on user's search history
- Managing user accounts and monitor their search history as well as the reviews added by these users

2.3 Existing Studies/Systems:

There are a number of systems, which provide recipe discovery and meal planning services, each having its advantages and disadvantages. Although they show the feasibility of digital tools to help make better dietary choices, the majority of them intensely lacks the provision of an intelligent and personalized experience. Some of the most popular systems, their functionality, technologies, and gaps that RecipeFusion tries to fill are highlighted in the table below.

Table 2.1: Existing Systems

Year	System	Contribution	Technologies Used	Limitations
2014	Yummly [1]	Recipe recommendations based on preferences	Web, Mobile App, Basic AI	Limited personalization, lacks ingredient-based search
2012	AllRecipes [2]	Ingredient search and recipe ratings	Web Technologies	No AI-generated recipes, no meal planner
2013	Eat This Much [3]	Calorie-based meal planning	Web, Nutrition API	Focuses on calories, not full dietary needs
2021	ChefGPT [4] [5]	Generated recipes using AI on user's input	AI/ML, NLP	Lacks in meal planning, there is no such system for that

These systems outline the growing interest in smart meal planning service. Nevertheless, they mostly do not have such holistic aspects as ingredient-conscious recommendations, built-in nutrition analysis, dynamic AI-powered recipe creation, and advanced chat-based

communication. Recipe Fusion will work on these deficiencies by giving a user friendly system.

2.4 Comparison of Existing Systems:

Table 2.2: Comparison of existing systems

Feature	Yummly	All Recipes	Eat this much	Chef GPT	Proposed System
AI Personalized	Basic	None	Low	Yes	Yes
Ingredient Based Search	No	Yes	No	Partial	Yes
Nutritional Information	Basic	Limited	Yes	No	Yes
Meal Planner	No	No	Yes	No	Yes
Custom recipe generation	No	No	No	Yes	Yes
User Experience	Good	Medium	Medium	Good	High

2.5 Summary:

This chapter revised the knowledge available in the domain and the current systems that were applicant to the project. Although there are a number of platforms that give advice on recipes and provide meal planning services, not many platforms provide a fully customized, ingredient conscious and nutrition built experience. Getting rid of these gaps, the proposed system RecipeFusion incorporates the latest technologies GenAI, real-time search by ingredients, and support of the interactive UI. RecipeFusion offers an advanced, scalable, and user-friendly solution to the current needs that today are more intelligent, personalized and health based by offering usability, scalability and capabilities to meet individual dietary requirements.

CHAPTER 3

REQUIREMENTS SPECIFICATION

3.1 Introduction:

This chapter explicates the specifications that would be required to develop the RecipeFusion - a recipe generation and recommendation system. These requirements can be defined as interface requirements and functional requirements which enable smooth interaction among the various parts of the system and its basic features that the system should provide respectively. Moreover, the chapter differentiates software and hardware interface requirements to present a detailed picture of the work of the system in terms of operation and technical requirements. Combined, these specifications can guarantee that RecipeFusion will be efficient, scalable, and meet the expectations of the users relating to intelligent meal timing.

3.2 Interface Requirements:

The interface requirements are those requirements that are necessary to enable effective communication between the different subsystems of the system. These requirements outline the nature of interaction and integration between various modules including the recommendation engine, user interface and the APIs. Also, this part provides the system interaction with external environments such as the web browsers, the cloud services, and other third-party data sources.

3.2.1 Hardware Interface Requirements:

To have a comfortable running and availability of the system, the following resources are required:

User-End Devices:

Users Devices used by users of the recipe application, particularly in rich media format (ex: viewing charts, filtering recipe) should have the following minimum requirements:

RAM: Any RAM has not less than 16 GB to make the rendering efforts appealing

Processor: minimum quad-core, 2.5 Ghz or more

Storage: SSD of at least 128GB of free space to rapidly load

Cloud or Local Server:

The backend which is based on Node.js will require a constant and scalable server architecture to process user authentication, real-time data requests and

administration reports.

Processor: Multi-Core (Quad-Cores or more) to stump upon parallel requests of API

RAM: 16GB or above to support many users simultaneously, queries in databases and file processing

Storage: 1TB of minimum storage space in form of SSD storage to store reviews, user information and usage analytics.

Connectivity: Connection to a high speed internet to provide speedy and responsive API connections.

3.2.2 Software Interface Requirements:

Frontend Technologies:

React.js: For building a responsive and dynamic UI.

Backend Technologies:

Node.js: Handles API logic, authentication and other tasks.

Express.js: Provides routing and middleware support for API development..

Database:

MongoDB: A NoSQL database used for storing user and roles, reviews and ratings, and search history details.

External APIs:

Email/OTP: Using Nodemailer for secure OTP delivery and password reset flow.

Operating System Capabilities:

Designed for cross-platform support, allowing access from:

Desktops and Laptops: Windows 10/11, macOS, Ubuntu/Linux

Mobile Devices: Android and iOS (via responsive web UI)

3.3 Functional Requirements:

Interfaces requirements show the feature that would be required in the system that would give smooth communication between various sections of a system.

3.3.2 Authentication and Authorization

Secure access to the system is granted so that only the user can access respective pages.

3.3.3 Recipe Discovery

Allow dietary preferences to be filtered (e.g. vegan, vegetarian, gluten-free, etc).

3.3.4 Custom Recipe Generation

Create recipes unique to get by searching using the available ingredients and preferences that match to health objectives.

3.3.5 Shopping List Generator

Auto generate the grocery list in accordance with the recipe that has been searched.

3.3.6 Recipe Saving and retrieval

Gives user an opportunity to save the favorite recipes as well as allowing easy access at any later time.

3.3.7 Meal Planning and saving

Gives the user option to have his or her own meal plan and save on number of calories and save.

3.3.8 Responsive Design

Brings Compatibility with laptops, tablets and mobile devices. Provide the smooth experience irrespective of the screen size and device.

3.3.9 Performance and Scalability

The ability to support several users at once with no performance. Objectives Support further extension with scalable architecture.

This set of requirements will allow RecipeFusion to provide a strong, intelligent platform that not only makes the meal planning process easier but also promotes healthy eating habits and overall user delight which will be achieved by making the platform to be personalized and interactive.

3.4 Use Case Model:



Figure 3.1: Use Case Diagram

3.5 Use Cases:

3.5.1 Use Case 1: Register

Register User Case helps new users to register an account with the platform. Registration creates an obligatory access to various features. Registration To gain access to the resources, user has to register using the basic credentials such as username, email, cell phone number, gender and password and allow user to log in to access the resources

Table 3.1: Use Case 1: Register

Aspect	Details
Actor	User
Description	Provides the user with the option of registering by filling in details like username, email, phone, gender and password.
Precondition	The user is not logged in and full features are not available to the user.
Trigger	When the user in question clicks on the Sign up button, he or she will be registered.
Postcondition	Account will already be created and user registered only need to log in rather than to register.
Primary Flow	<ul style="list-style-type: none">• User completed sign up form with all details.• The system checks the received information provided by user.• System records the information of the user in database.• User is given confirmation message
Alternate Flow	In case of existing user name or email message will be displayed to user
Includes/Extends	Input validation

3.5.2 Use Case 2: Login

The application of the use case allows registered users to authenticate and be granted access to application-specific features. An effective session of the information system is possible, and the user is able to search recipes and save them, among others.

Table 3.2: Use Case 2 - Login

Aspect	Details
Actor	User
Description	This gives the registered users an opportunity to log in with their email and password
Precondition	We have user registered in the account
Trigger	The user clicks the button of Log-in and enters credentials
Postcondition	User will be redirected to the corresponding dashboard having logged in.
Primary Flow	<ul style="list-style-type: none">• User key in email and password.• The credentials are checked by the system.• In case of a successful verification, the user shall be redirected to specific dashboard e.g; user or an administration dashboard depending on respective roles i.e. 1 for an administration and 2 for a user
Alternate Flow	The error message is displayed in case of erroneous credentials systems
Includes/Extends	Logic of authentication

3.5.3 Use Case 3: Search by Name

The use case enables users to search a particular recipe by typing a name of the recipe. It is also helpful most especially when the user is aware of what they are searching.

Table 3.3: Use Case 3 - Search by name

Aspect	Details
Actor	User
Description	The description allows the user to search through recipes based on a particular recipe name
Precondition	Should be logged in
Trigger	The user types in the name in the search bar (with the ability to choose filters as well) and presses the button "Search."
Postcondition	Recipes matching are presented on the screen

Primary Flow	<ul style="list-style-type: none"> • Name of recipe of user type. • The system finds the name of the corresponding recipe. • Displays the results
Alternate Flow	No recipe available - system displays "No recipes available"
Includes/Extends	Text-search

3.5.4 Use Case 4: Search by Ingredients

Gives the users the authority to locate recipes depending on their cooking ingredients hence reducing food waste and enhancing convenience.

Table 3.4: Use Case 4 - Search by ingredients

Aspect	Details
Actor	User
Description	They can be used to search recipes based on ingredients that the users have.
Precondition	It is required that a person is logged in.
Trigger	When the user enters the input of ingredients and clicks on the button of search, the entry is made and the search begins
Postcondition	The numbers of recipes containing that entered set of ingredients are presented
Primary Flow	<ul style="list-style-type: none"> • User is one who enters the ingredients. • The system finds recipes containing such ingredients. • Displays the results
Alternate Flow	No matches - system is offering comparable recipes or asking to broaden search.
Includes/Extends	Ingredient-matching logic

3.5.5 Use Case 5: Apply Filters

Such use case improves recipe search by allowing the user to use dietary, preparation time, or even cuisine to filter search results.

Table 3.5: Use Case 5 – Apply filters

Aspect	Details
Actor	User
Description	The dietary search and the cuisine search enable the user to narrow down the search of recipes

Precondition	The user is going to make a search
Trigger	User clicks on anywhere options
Postcondition	Filtered search results are updated
Primary Flow	<ul style="list-style-type: none"> • User chooses filters. • The match filter search in the system. • Filtered results shown.
Alternate Flow	No similarities located - displaying "No results"
Includes/Extends	Logic used to filter

3.5.6 Use Case 6: Production of Shopping List

This use case makes a grocery list based on chosen recipes which assist the users to plan and shop economically.

Table 3.6: Use Case 6 – Production of shopping List

Aspect	Details
Actor	User
Description	Establishes a shopping list in ingredients utilized in chosen recipe
Precondition	The user has chosen a recipes
Trigger	User presses on the button "Generate Shopping List"
Postcondition	List of ingredients is presented
Primary Flow	<ul style="list-style-type: none"> • User views a recipe • Offers ingredient list in the system
Alternate Flow	None of the recipes are chosen - make the user choose at least one recipe
Includes/Extends	-

3.5.7 Use Case 7: Recipe Recommendations

This use case provides users a recommendation of recipes depending on the chosen ingredients, filters, or past activity. It is meant to give consumers easy and customary recipe recommendations.

Table 3.7: Use Case 7 – Recipe Recommendations

Aspect	Details
Actor	User
Description	Recommends recipes depending on the preferences of users
Precondition	User logged in
Trigger	Automatically proposed by the system
Postcondition	Onscreen there is a display of recommended recipes
Primary Flow	System retrieves and shows individual suggestions
Alternate Flow	System proposes trending or popular recipes
Includes/Extends	-

3.5.8 Use Case 8: Save a Recipe

Gives the ability to book mark recipes to be deemed used later and that way it is easier to access favorite or even interesting recipes. The recipe will be stored in the favorites of the user and he can refer to it anytime he wishes

Table 3.8: Use Case 8 – Save a Recipe

Aspect	Details
Actor	User
Description	Allows one to store recipes in the personal account
Precondition	The user is already logged in and on a recipe
Trigger	When a user clicks on the save or bookmark icon of a recipe
Postcondition	Recipe appears in the list of recipes that are saved by the user
Primary Flow	<ul style="list-style-type: none">• User clicks on the save button.• System stores the recipe under the profile of the user• User gets success message

Alternate Flow	System display "Already saved"
Includes/Extends	Save/ Remove saved logic

3.5.9 Use Case 9: Add review

Gives users the chance to leave a review of a certain recipe to utilize later. User has the ability to share his thoughts in case he likes or does not like the recipe.

Table 3.9: Use Case 9 – Add review

Aspect	Details
Actor	User
Description	This allows one to provide review on a recipe
Precondition	The user is already logged in and on a recipe
Trigger	The user presses the button option of adding review on the recipe detail page
Postcondition	Review is stored and linked with the recipe
Primary Flow	<ul style="list-style-type: none"> • User gets to the review • System stores the review • User gets success message
Alternate Flow	No complete form
Includes/Extends	Input validation.

3.5.10 Use Case 10: Meal Planner

Is able to allow users to planner a meal utilizing GenAI that will produce the meal planner of the entire week.

Table 3.10: Use Case 10 – Meal Planner

Aspect	Details
Actor	User
Description	Helps users to receive generated meal plan
Precondition	User has been logged in and is at meal planner page
Trigger	The user triggers this by a mouse click on the Meal planner page of his dashboard button
Postcondition	Meal plan is assigned and in the page

Primary Flow	<ul style="list-style-type: none"> • User clicks create new plan. • System came up with 7 days meal plan • User can save the meal plan
Alternate Flow	Server is busy
Includes/Extends	Clicking the button.

3.5.11 Use Case 11: Admin Functions

i. View Users

Table 3.11(a): Use Case 11a – Admin Functions

Aspect	Details
Actor	Admin
Description	Shows the list of all registered users on the platform so that the administrator can view it
Precondition	Admin is logged in
Trigger	Goes to user management page
Postcondition	User list is displayed.
Primary Flow	<ul style="list-style-type: none"> • Admin logs in • Admin opens “All Users” view • Data are retrieved and displayed through system
Alternate Flow	System/database error is displayed with the correct message
Includes/Extends	Loading users.

ii. Edit/Update/Delete Users

Table 3.11(a): Use Case 11b – CRUD on Users

Aspect	Details
Actor	Admin
Description	Permits user records to be edited/ deleted by the admins
Precondition	Admin must be logged in to user list
Trigger	Admin clicks the edit or delete on the right of a user
Postcondition	User information has changed or been deleted

Primary Flow	<ul style="list-style-type: none"> • Admin selects a user • Admin clicks edit or delete • Update/removal is done on the system
Alternate Flow	User does not exist or system error
Includes/Extends	Validation of form, authorization check

iii. View/Delete Reviews

Table 3.11(c): Use Case 11c – View/Delete Reviews

Aspect	Details
Actor	Admin
Description	Enables one to view and delete reviews of the user
Precondition	Admin is logged in
Trigger	Review management section visited
Postcondition	Meeting is deleted by administration in case unprofessional
Primary Flow	<ul style="list-style-type: none"> • Clicks on reviews list • Establish any review to be deleted • System remove that review
Alternate Flow	Check Does not exist and deletion is not possible
Includes/Extends	Logic deleting and viewing

iv. View Search History

Table 3.11(d): Use Case 11d – View Search

Aspect	Details
Actor	Admin
Description	Shows a history of the search history of the users to analyze trend
Precondition	Admin is logged in
Trigger	Analytics or history will be accessed under the analytics or history option
Postcondition	System shows the history or search trends
Primary Flow	<ul style="list-style-type: none"> • Admin opens main tab • System loads and displays history data

Alternate Flow	Error Response Information not found
Includes/Extends	Statistics loading

3.6 Non-Functional Requirements:

Outlines the quality attributes that the system should comply with to operate correctly besides its functionality. When advising on meal planning and recipes, it centers its attention on responsiveness, user accessibility, reliability and security of the platform.

3.6.7 Performance:

This system should be such that when the user requests are critical functions like searching any recipe and accessing details of that recipe are displayed and can be viewed in 5 to 8 seconds once a request has been initiated.

The application is supposed to be optimized in order to keep load times to a minimum even when traffic is high or during peak usage times so that consistency in service delivery in the most efficient manner can be maintained.

3.6.8 Reliability:

The system should have usability of 99.9 which translates to a yearly downtime of less than 1 hour. This requirement is all-inclusive of all integration elements such as the database and the backend services.

It should have regular automatic backup and failover processes to correct against loss of data and recover fast just in case of server failures or system crashes.

3.6.9 Security:

The whole communications between client and server should be encrypted using HTTPS where sensitive user information should be encrypted and will not be intercepted or man in the middle attack.

Hashing of the user passwords should be done and the passwords stored using strong hash algorithms. Security features will consist of:

Password reset functionality

Session management

Role-based access control

3.6.10 Consistency:

RecipeFusion needs to ensure that there is consistency of the data in all user interactions. Any changes in user preferences, stored recipes or meal plans should be updated in real time throughout the system.

3.7 Resource Requirements:

The resource requirements establish the critical technical and infrastructural elements required in supporting ingredients based recipe discovery on the platform. It contains the specifications of the user-end devices, server environments, development tools as well as external services.

3.7.7 Equipment:

Hardware: It will require an energy consuming server that can serve several users on the server.

Software: The development of the software will be grounded on the latest web technologies. The main integrated development environment (IDE) on which the coding work will be performed will be VSC. Server side environment will be composed of Node.js, backend logic and routing is to be done by Express.js. Mongo database will be employed as a versatile and scalable NoSQL database to store the profiles of the users, information about the products and all information concerning it.

3.7.8 Human Effort:

Frontend Development (40% effort): It is the phase of designing and creating the user interface to generate a smooth and user-friendly user interface. Developers will use React.js, tailwind, and APIs to design recipe browsing, filtering, and dashboards to a large extent.

Backend Development (30% effort): Senior Backend engineers will code and build the server application: API, user authentication and data schema. The task involves performing the optimization of the data flow, the security and control

of the points of APIs integration. The backend will also offer an efficient and scalable completion of the requests and data storage with the use of Node.js.

Testing and Quality Assurance (20% effort): Testing will be done relating to functional correctness, security vulnerabilities, responsiveness in relation to various devices and conditions of the networks.

Deployment (10% effort): It is the phase of implementation of the application and its introduction to a real production environment. Deployment of cloud-based infrastructure (such as Vercel), environment variables. Investors will ensure that the process of deployment will be safe and consistent and will control the system performance.

3.8 Database Requirements:

MongoDB:

The database will contain following collections:

Users: stores details about registered users such as first name, last name, username, email, phone number, gender, role, and password, height weight, age, cuisine preference, any allergy and health condition etc.

Reviews: The information about the reviews made by users is stored in stores including user id, recipe id, title, review, rating etc.

SavedRecipes: Used to store data on the recipes saved by the users such as user id, recipe id, title, image etc.

SearchHistory: Saves information on search history conducted by users such as user id, query, type etc.

3.9 Project Feasibility:

The analysis determines the viability, the feasibility and practicability of the project. It takes into account the resources, technologies and potential constraints to be used to know whether the system will be easily implemented and maintained in the long run. This is because the feasibility analysis will enable the system to satisfy the needs of users as well as meet the project objectives and resources available.

3.9.1 Technical Feasibility:

RecipeFusion system could be developed because the current web development technologies are mature and stable enough. The application of the MERN stack (MongoDB, Express.js, React.js, Node.js) offers a solid and scaled platform to develop the backend and the frontend of the application. React allows creating an interactive and responsive user interface, whereas Node.js and Express.js promote an efficient management API.

3.9.2 Operational Feasibility:

RecipeFusion is also user-friendly and has a user-centered design, which makes it focus on performance and usability. The major workflows, that is, ingredient based search, meal planning and recipe generation are automated to provide speed in terms of response. The user experience is also good, with real-time response and their intuitive layout and the duration it takes to load the page.

Operationally, the site will need moderately priced server maintenance, regular database backup and the infrequent provision of updates to features. The system will be highly reliable with its well-engineered and responsive front and back end layer that is likely to allow it to handle growing user demands.

3.9.3 Legal and Ethical Feasibility:

RecipeFusion will also comply with the law and other ethical considerations of user privacy and responsibility in the digital world. Sensitive data of users, like email addresses and passwords will be encrypted and stored safely with the help of hashing algorithms, including bcrypt.js. To ensure that the access is not unauthorized, the system will use access control mechanisms and secure sessions.

3.10 Summary:

In this chapter, the key non-functional requirements of the RecipeFusion recipe recommendation and generation system reflected were on performance, scale, security, and reliability. These factors play a vital role in facilitating the response, safety, and reliability of the system by the users as well as multiple simultaneous sessions and preserving and maintaining accuracy of the data.

The chapter also involved the discussion of the essential resources that are necessary in terms of technological as well as human resources to facilitate successful development and maintenance of the platform. It paid special attention to the usage of modern and supported technologies, i.e. React.js, Tailwind, Node.js, and MongoDB, which altogether create a stable and scalable core to the RecipeFusion.

Lastly, the feasibility study validated the fact that it is a technically, operationally, and ethically sound project. RecipeFusion combines a user-friendly format and a technical infrastructure that will be highly successful in achieving its goals and delivering intelligent, personalized service of performing meal planning and healthy eating in the present day. All these factors create a powerful foundation of the successful implementation of the system.

CHAPTER 4

SYSTEM MODELING

4.1 Introduction:

System modeling is a crucial aspect of software development essentially a trade-off between the high level conceptualization and the actual on-the-ground implementation. It allows the system architecture, workflows, data sharing, and user interactions to be drawn in a structured and visual way that is easy to understand and which is informative to both the developers and designers as well as the stakeholders.

This chapter provides a system design of RecipeFusion a web-based recipe recommendation and generation system that enables the user to find meal choices that is based on his or her dietary requirements and available ingredients. The purpose of the platform is to make cooking easier, promote healthy eating, and increase the level of transparency of the ingredients by incorporating intelligent recommendations and traceable data streams.

The main aim of RecipeFusion is to provide the client with an intelligent, personalized, and user-friendly tool to search and store the recipes, taking into consideration their nutritional objectives, preferences (e.g., vegan, gluten-free), and the availability of ingredients. Basic traceability capabilities are also provided by the system, where the details of the source of critical ingredients can also be viewed by the users (where available).

The model techniques that have been embraced in order to achieve a successful, up scaled and user-friendly architecture are numerous:

Class Diagram - It depicts the fundamental objects in the system like Users, Ingredients, Reviews and search history and connection.

Activity Diagram - Shows the sequence of activities performed by the user like searching, saving, reviewing and filtering recipes by the available ingredients.

System Sequence Diagram - Diagram that illustrates realization of the flow of user interactions with the system during the main processes such as the process of gaining entry into the system, generating models of recommendations, and submitting a review.

Component Diagram - The component diagram visually depicts the modular structure of the system, i.e. the frontend interface, backend APIs, database table, and third-party services (e.g., authentication).

Deployment Diagram - Model of the physical architecture, i.e., client devices, cloud-based frontend-hosting (e.g. Vercel), backend deployment (e.g. Vercel) and the database (MongoDB).

Data Flow Diagram (DFD) - It is a map illustrating flow of data in the system as it comes in through the user and out of the system as a result.

Combined, these models will offer a bigger picture of the internal processes and external relations of the RecipeFusion system that will offer a well-documented base of the implementation, scaling, and future upgrades.

4.2 System Design:

System architecture design is a elementary process towards the creation of the RecipeFusion platform. It includes the process of setting the framework of the system and how various modules and services are to communicate with each other to accomplish the overall objective of intelligent and personalized recipe recommendations.

The input point to the design is the user with a responsive web interface. The user is able to take many actions, such as registration, dietary preferences, searching recipes, meal planning, as well as feedback. This interacts with an application backend that communicates with GenAI and a cloud-based database system that is in MongoDB atlas.

Real time user inputs in the form of ingredients availability, cuisine preferences, maximum time and dietary constraints are received and calculated. The GenAI then uses this data to provide recommendation of recipes using collaborative filtering and content-based techniques. Those recipes proposed are found in an organized recipe database and are re-introduced to the customer via web interface.

The general architecture of RecipeFusion is based on personalization that is user-fulfilled and modular, as well as scalable. The system is designed in a way that it encourages constant learning with the help of user feedback hence improving the accuracy of such future recommendations. Also, the communication is encrypted and the data is locked in safe depository areas which enhance the security and privacy of the data.

4.3 Design Approach:

In designing RecipeFusion, bottom-up design style was employed with emphasis laid

on the design and testing of the basic functional modules before connecting them together to create a complete application. The first step was to design and implement the backend infrastructure in the form of Node.js, Express.js, and MongoDB that acted as the basis of user management, and API endpoints. Similarly, the frontend interface was created with React.js, which puts a particular emphasis on the creation of an interactive and user-friendly environment in which recipes could be browsed, filtered, and saved.

All modules, frontend, and backends, were created separately and tested to make sure that they worked and were stable. Integration was also done progressively and it provided smooth connectivity between the user interface, API services and the underlying database. This disciplined model did not only increase defects of reliability and maintainability, but also made the task of debugging and quality assurance across the development spectrum simpler. By ensuring that every single element works independently and then connecting it to another, the team formed an easy and steady experience to end users and allowed problems to be scaled up later as well as adding features.

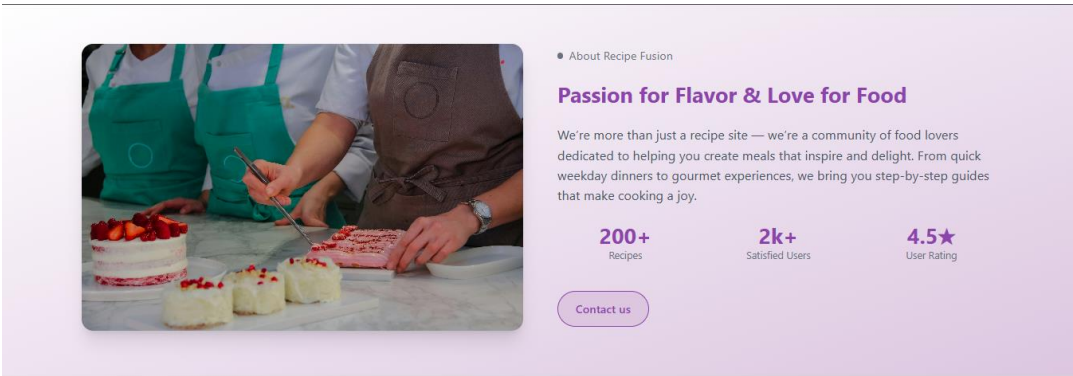
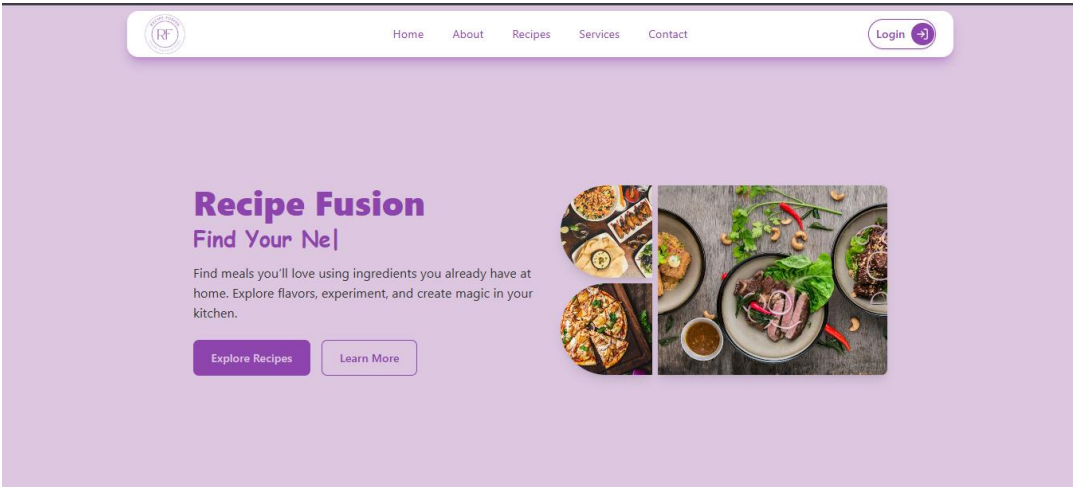
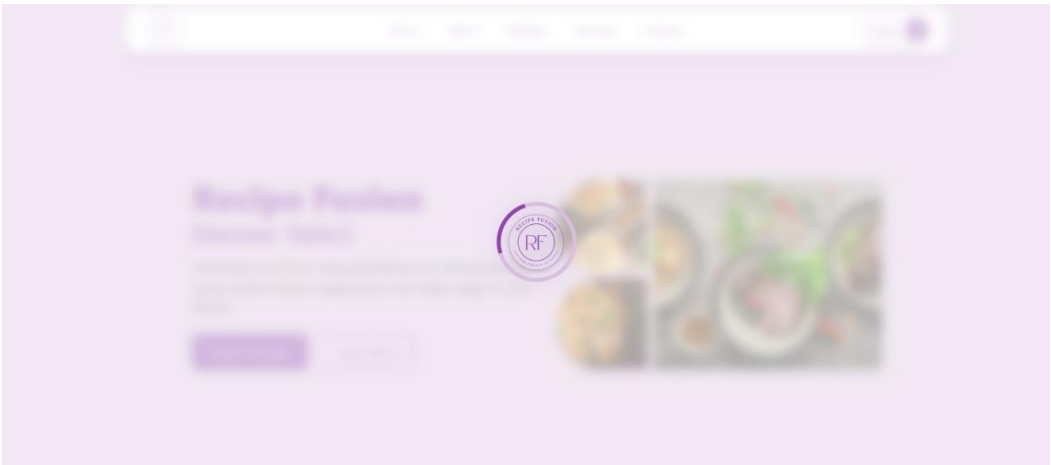
4.4 Interface Design:

The user interface system of RecipeFusion is centered on being simple, easy to use, and conveniently accessible to all types of users which could include casual users, people who are more nutrition-oriented and the administrators. An efficient layout and easy navigation pattern were embraced to be able to communicate with the system easily. The interface will comprise the elements of registration, logging in, defining the food preferences, food browsing, meal planning, and feedback. The last interface is developed according to the recommendations of the best practices of the user interface/user experience, which guarantees its functionality and appearance.

4.4.1. High-Fidelity Prototype:

The following are the screen shots of the UI of the platform:

Landing Screen:



• Explore different recipes

Popular Recipes

• Explore different recipes

Popular Recipes

Discover mouth-watering dishes from around the world — easy to make and full of flavor.



Classic Spaghetti Aglio e Olio



Creamy Tomato and Spinach Pasta



Classic Hyderabadi Chicken Dum Biryani



Aromatic Vegetable Dum Biryani



Classic Vanilla Sponge Cake



Decadent Chocolate Fudge Cake



Classic Spaghetti Carbonara



Lemon Garlic Shrimp Scampi with Linguine



Classic Vanilla Milkshake



Indulgent Chocolate Banana Peanut Butter



Classic Hyderabadi Chicken Dum Biryani

Description

A fragrant and flavorful Indian rice dish made with marinated chicken, basmati rice, and a blend of aromatic spices, cooked using the traditional dum method to lock in moisture and flavor.

Ingredients (19)

- 750g Chicken, cut into medium pieces
- 2 cups Basmati Rice
- 1 cup Plain Yogurt
- 2 tbsp Ginger-Garlic Paste
- 1 tsp Turmeric Powder
- 2 tsp Red Chili Powder (adjust to taste)
- 2 tsp Garam Masala
- 1 tsp Coriander Powder
- 1/2 tsp Cumin Powder
- Salt to taste
- 1/4 cup Lemon Juice

• What we offer

Our Services

Discover how we make your cooking journey easier and healthier with fresh ingredients, a wide variety of recipes, nutritional guidance, and simple step-by-step instructions.



Ingredient-Based Search

Find recipes instantly by entering the ingredients you already have in your kitchen.



Quick & Easy Recipes



Dietary Filters



Save Your Favorites

• Get In Touch

Contact Us

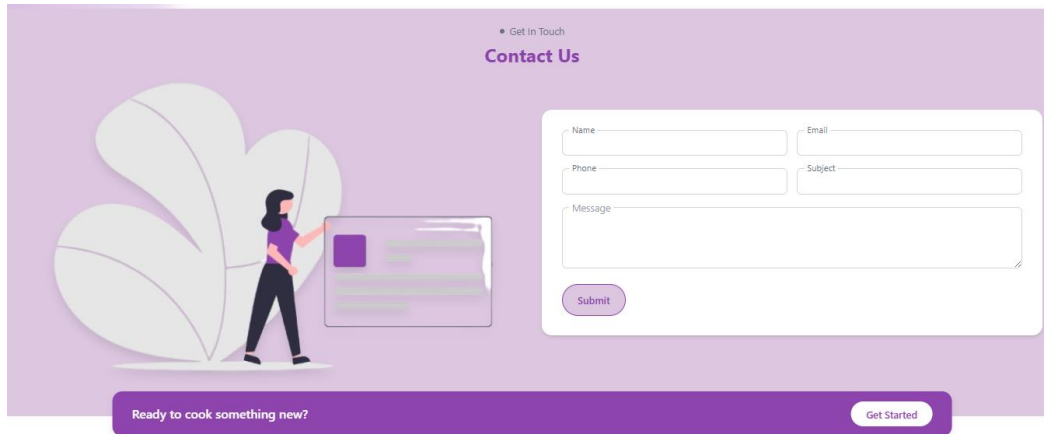


Fig 4.1 – Landing Screen

Login Screen: Users login from here to access their respective dashboards

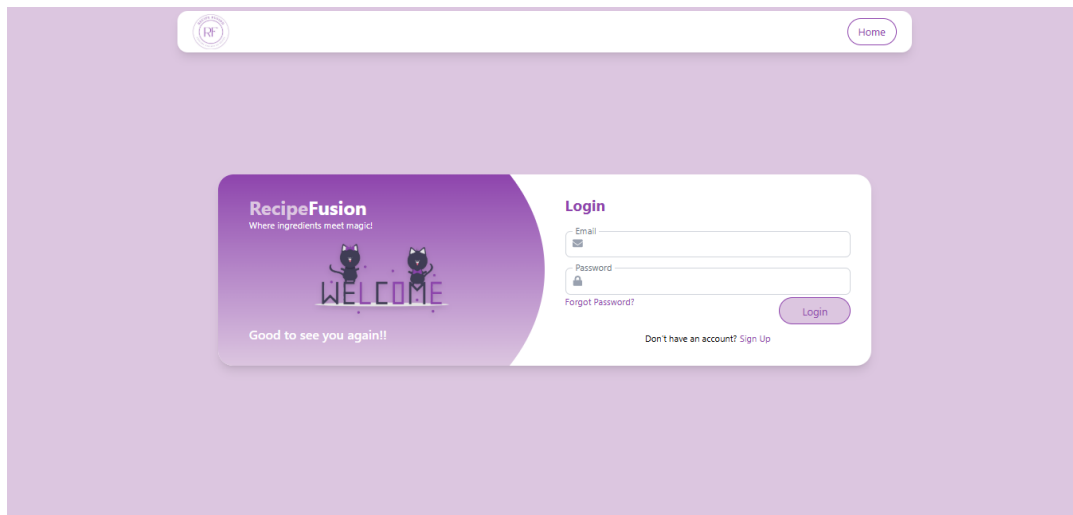


Fig 4.2 – Login Screen

Signup Screen: Users can register themselves to access particular features of the system

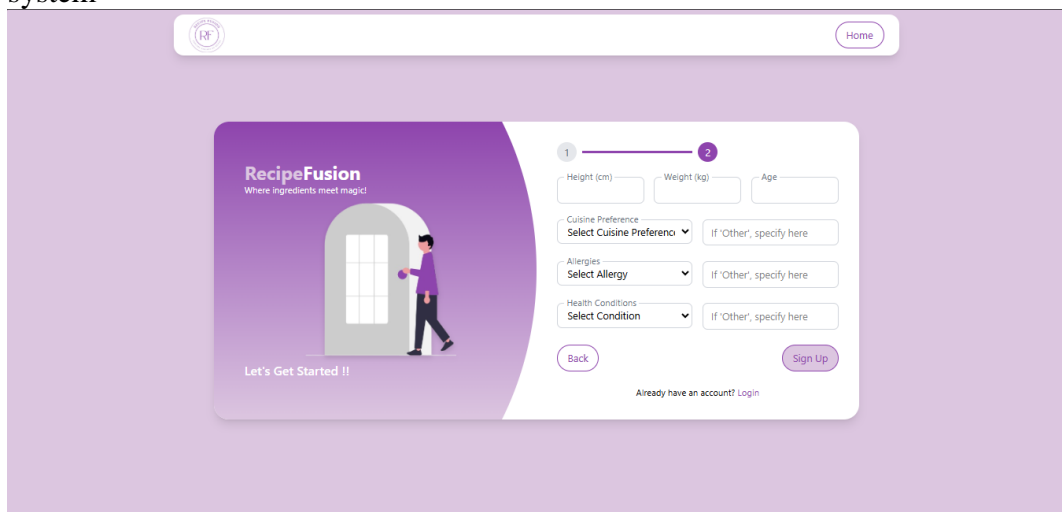


Fig 4.3 – Signup Screen

Admin Dashboard Screens:

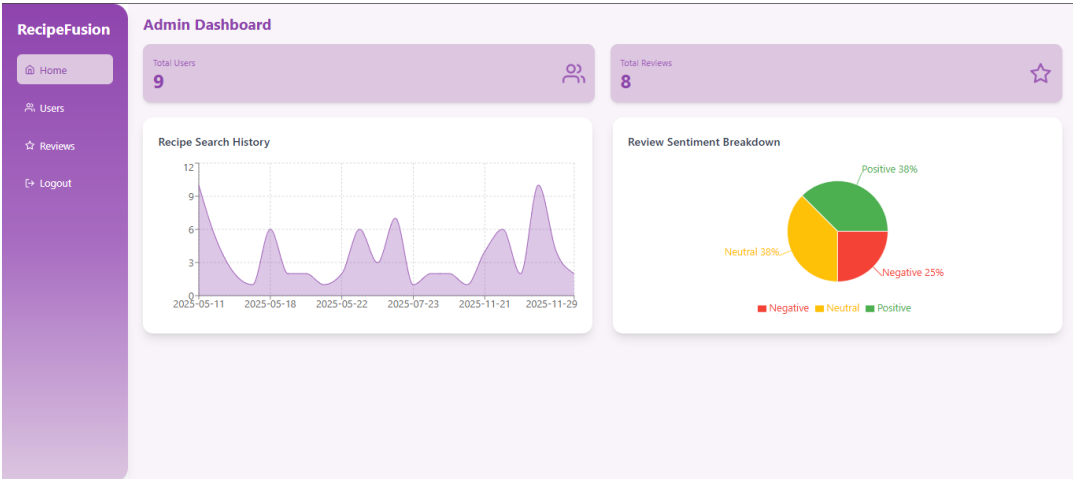


Fig 4.4 – Home Screen

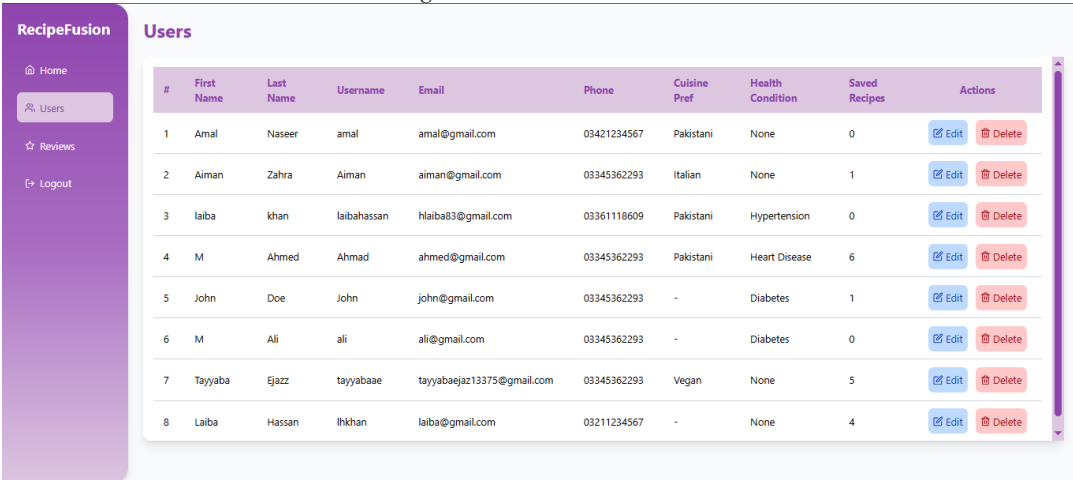


Fig 4.5 – Users Management Section

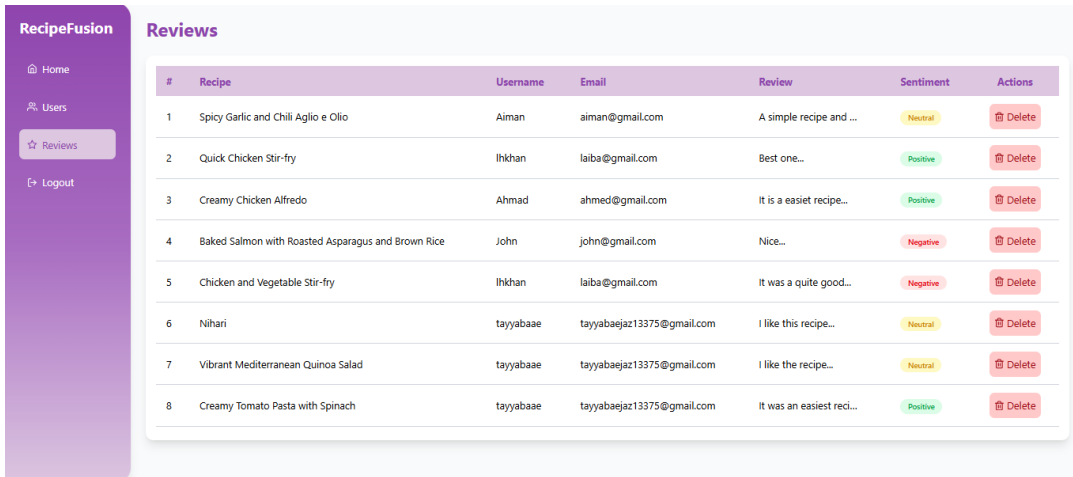


Fig 4.6 – Reviews Management Section

User Dashboard Screens:

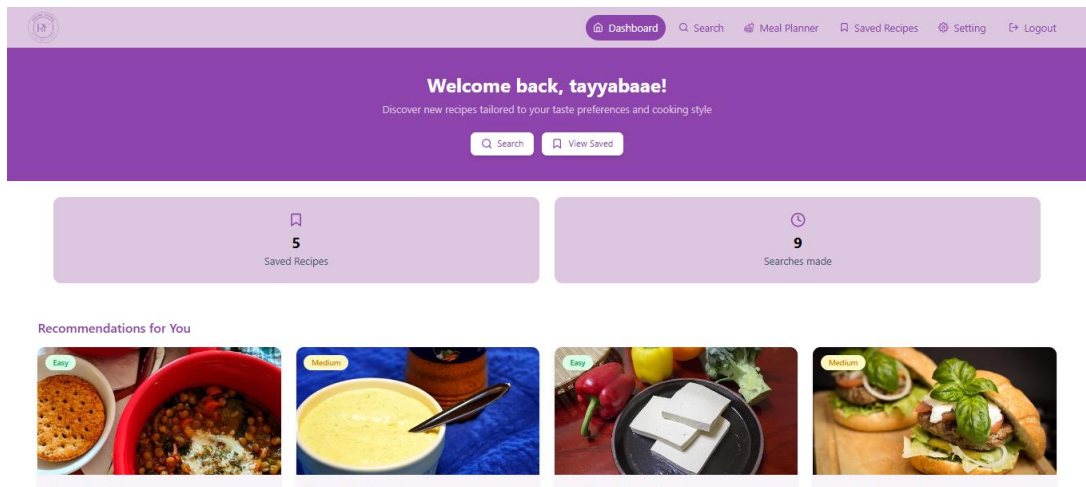


Fig 4.7 – Home Screen

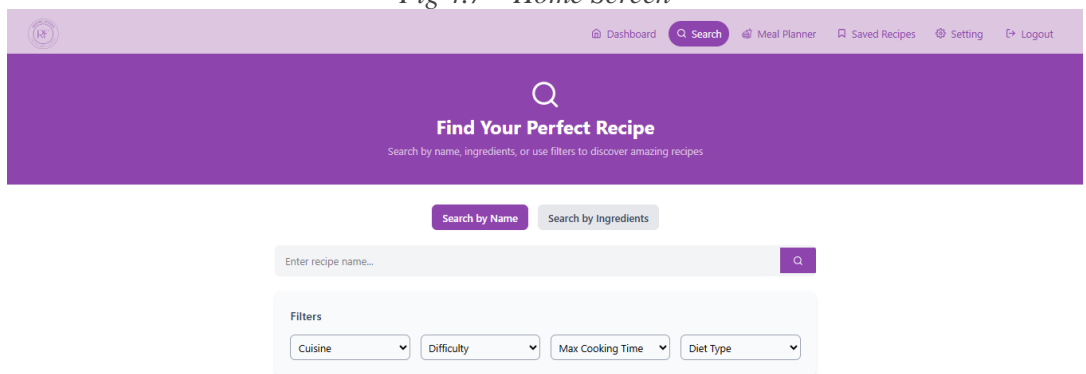


Fig 4.8 – Search Recipe Screen

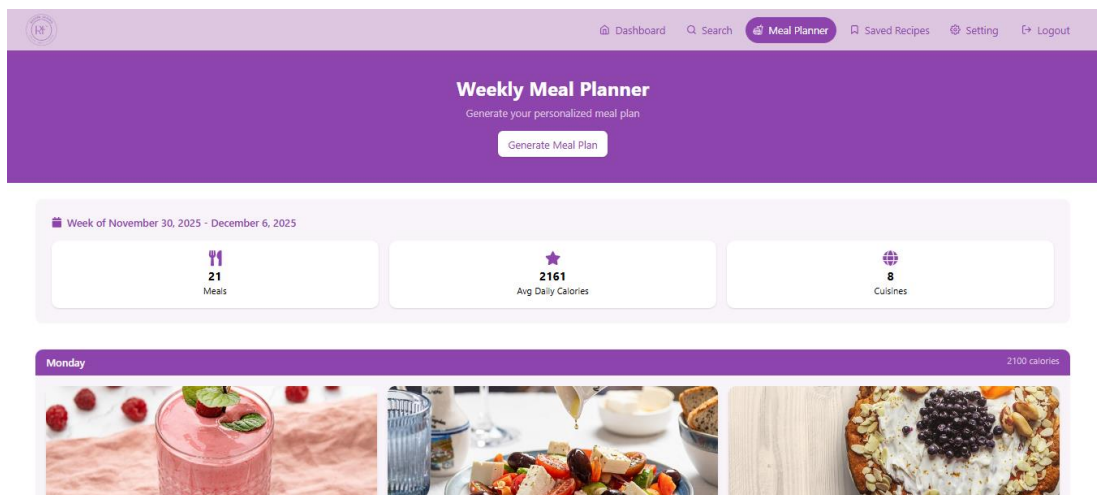


Fig 4.9 – Meal Planner Screen

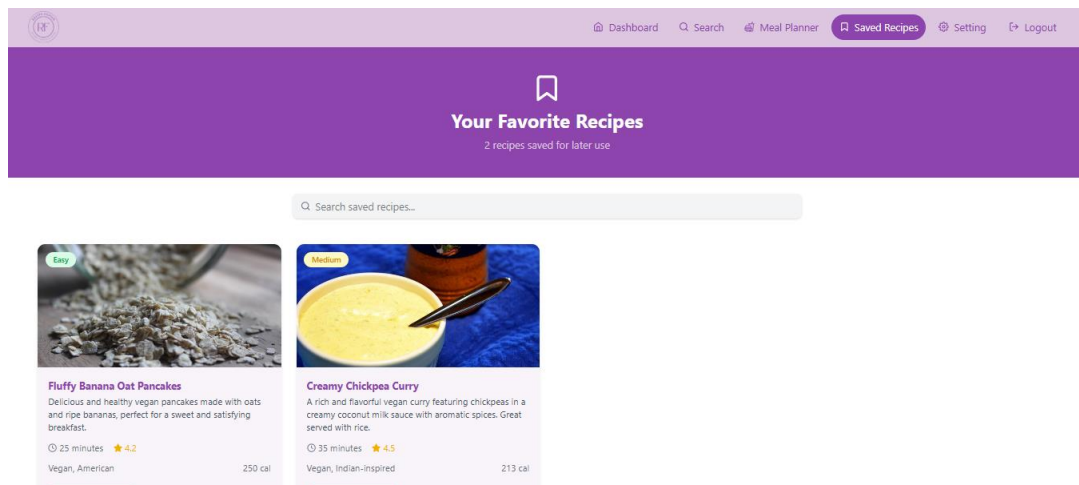


Fig 4.10 – Saved Recipes

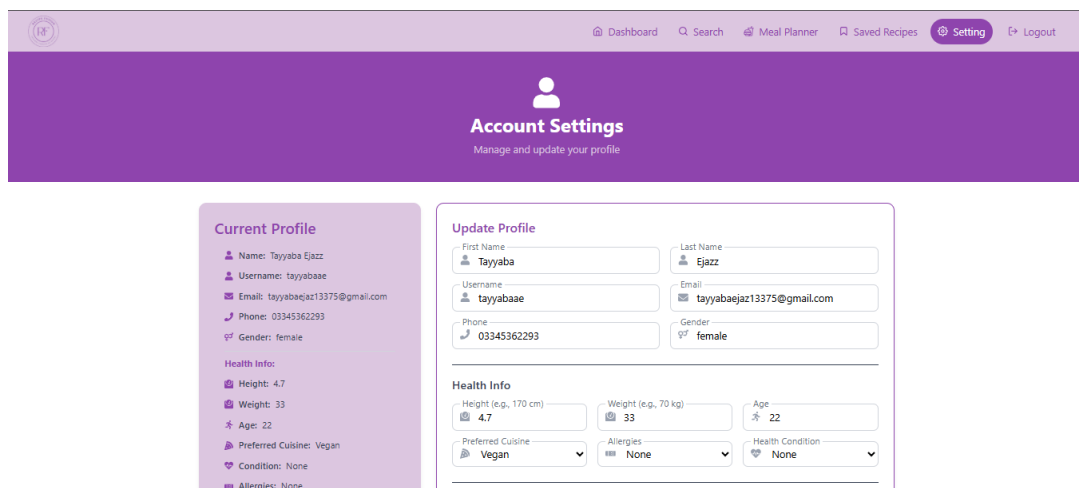


Fig 4.12 – Profile Settings

4.5 View Model Architecture:

The View Model of Architecture defines the structure of the RecipeFusion system offering a clear understanding to all interested parties. This model subdivides the system into several views by which the behavior, flow of data and the interactions of modules are clarified. It aids the developers, users, and administrators to be made aware of how the system will be structured, and it will be in performance and usability standard. The feedback used in the modeling process led to the possibility of successive improvement and designing is user-friendly and conforms to the project specifications.

4.5.1 Logical View:

A logical perspective (reflected in the relevant class diagram) views the main functional design of the RecipeFusion platform. It points out the key elements and interactions that yield the support of recipe recommendation, user personalization, meal

planning and ingredient-based search capability.

Key entities include:

User (General/Admin): Manages authentication, profile data and role based permissions. Recipes can be searched by users, added to favorites, reviewed, they get generated meal plans and can manage food preferences.

Recipe: This is a full description of the instructions, such as ingredients, steps to follow, cooking time, some quick facts (such as the cuisine, the number of people to serve, difficulties in preparation etc) and metadata added by the user: ratings and reviews. Shopping list can also be obtained in the form of the pdf which the user can get.

Ingredient: Stores separate food items, so that users can enter available ingredients in order to find recipes and traceability features (e.g. source or origin, where relevant).

Review: Beware of user rating and review of recipes.

Saved Recipe: This will enable the user to store away recipes so that it may be used in the future.

Search History: Keeps the history of the user of the search engine to deliver quicker suggestions and personalization in future.

The structure and relationship of these core entities are defined in the logical view which is represented in the form of class diagram. It guarantees logical connections of data and structure functioning throughout the system elements. This concept endorses the operational objectives of Recipe Fusion as it allows customizable user interfaces, effective recipe prescription, and scaling user and recipe information.

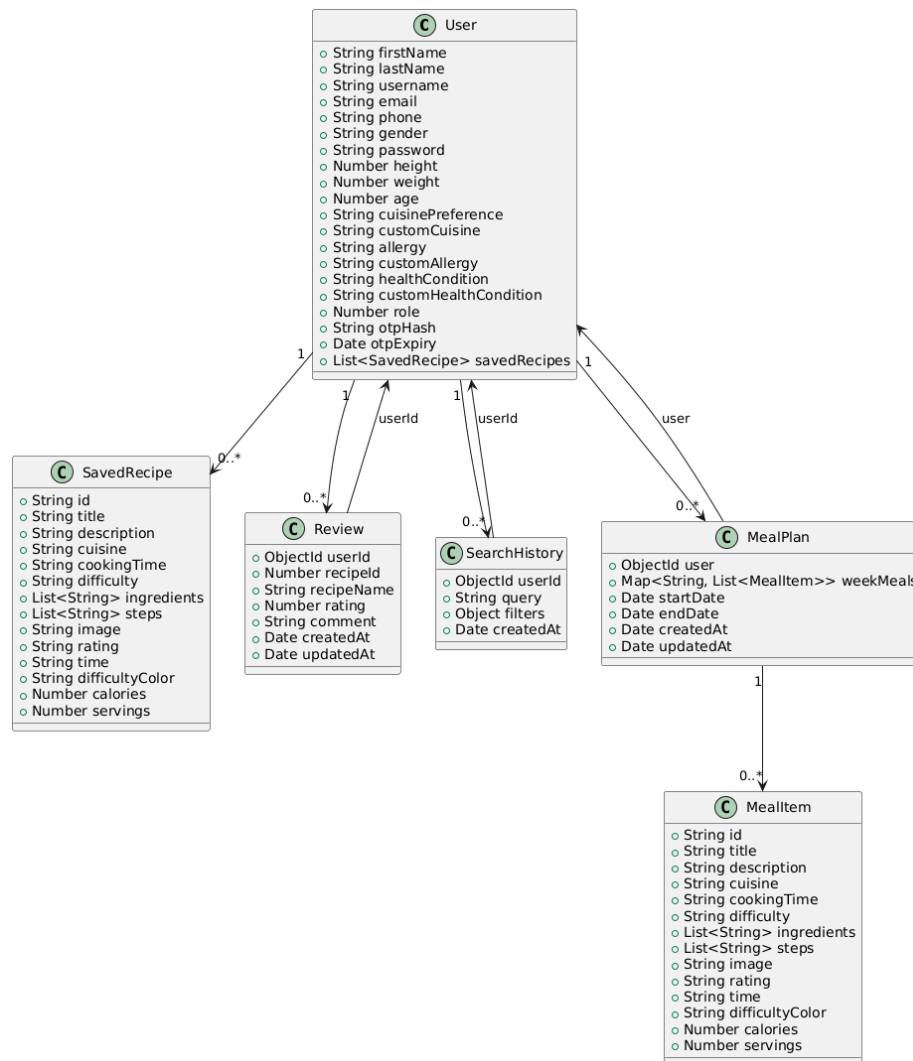


Fig 4.13 – Logical View

4.5.2 Process View:

The process view of the RecipeFusion platform describes the major runtime behaviors, simultaneous processes, and system interactions enabling to discover the personalized recipes and interact with users. This perception describes the real-time exposure of the system components to make it responsive, scaled and efficient.

The frontend consists of a registration or logging-in interface where the user will start his or her journey. On the terms of submitting the credentials of the login, the backend (Node.js/Express) authenticates the codes, provides a JWT token to manage a session, and loads a role-based dashboard. Several individuals can log in to the system and communicate with it at the same time without interfering with each other.

After authentication, users will have the opportunity to enter the available

ingredients, choose the dietary preference or search recipes. At the same time, the system asks GenAI to search and return similar data on recipes, which is filtered in the same way according to the products used, the type of cuisine or even the diet restrictions. These filters and search are done simultaneously giving the user a smooth view.

Reviewing can be made by users and it is saved and compiles up to affect the rating of recipes as well as its recommendations. At the same time, this has logged the search history to tailor other subsequent suggestions using collaborative filtering.

The admins are able to handle users, search history and reviews in an administrative dashboard. They are able to carry out tasks such as deleting objectionable content, evaluating the search history of the user, and other tasks of the entire content store. These processes are initiated without user-side interactions, and system fluidity is ensured.

Search suggestions, caching, handling of API requests are asynchronous to allow better performance in high-traffic situations. Other applications of concurrency include the ability to save recipes, update profiles or browse at another module by RecipeFusion.

The system is designed to have a frontend (written in React.js) that responds to users and is also dynamic and a backend that processes API logic, authentication, and database communication. MongoDB allows each recipe, user and preference to be elastically stored so that it can handle many read/writes at a time.

This operation process depicts how RecipeFusion is responsive and event driven allowing asynchronous operations and parallel processing so that there is a flow of culinary experience that is seamless and unique to the user in various roles and on various types of devices.

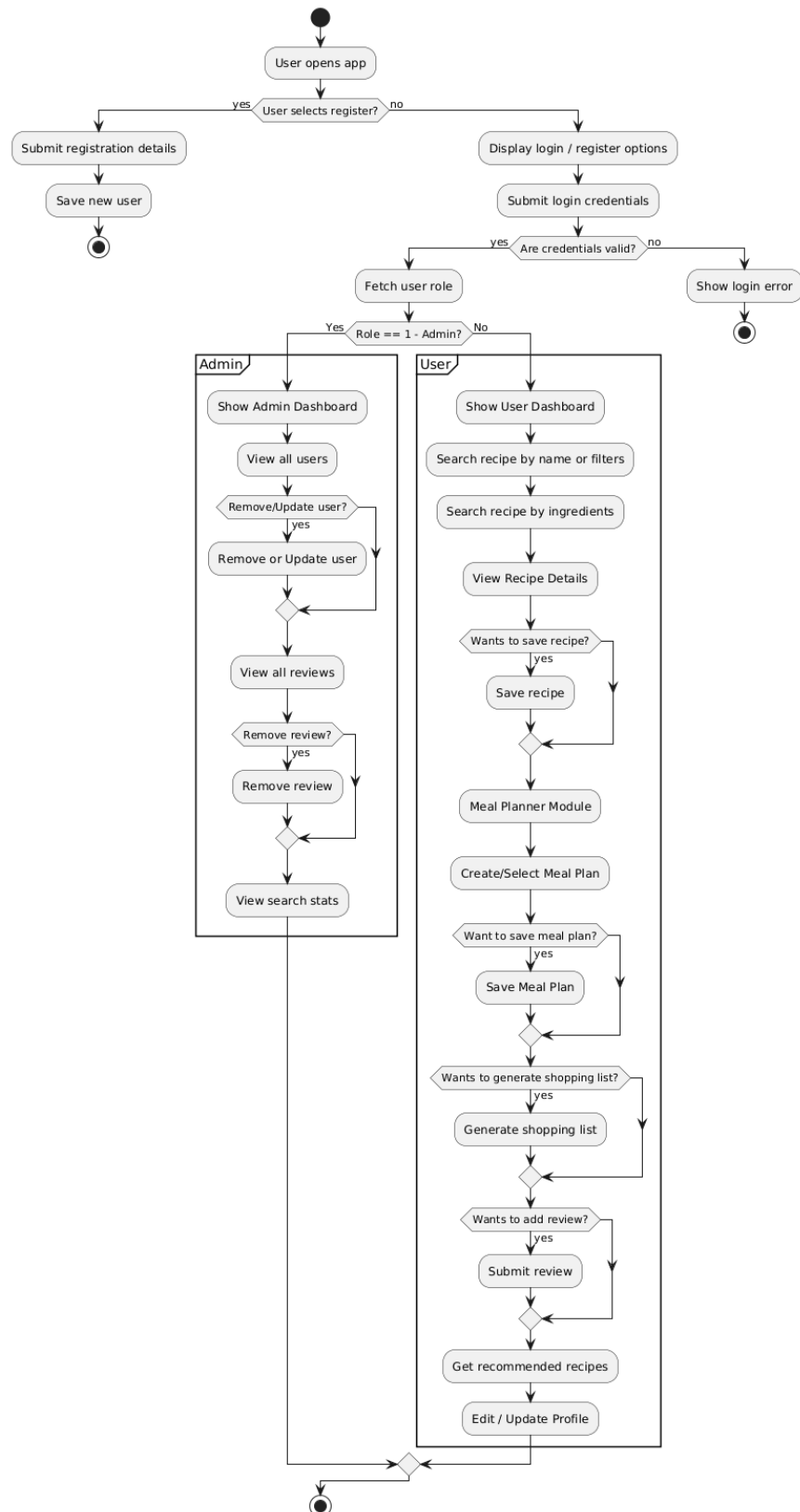


Fig 4.14 – Activity Diagram

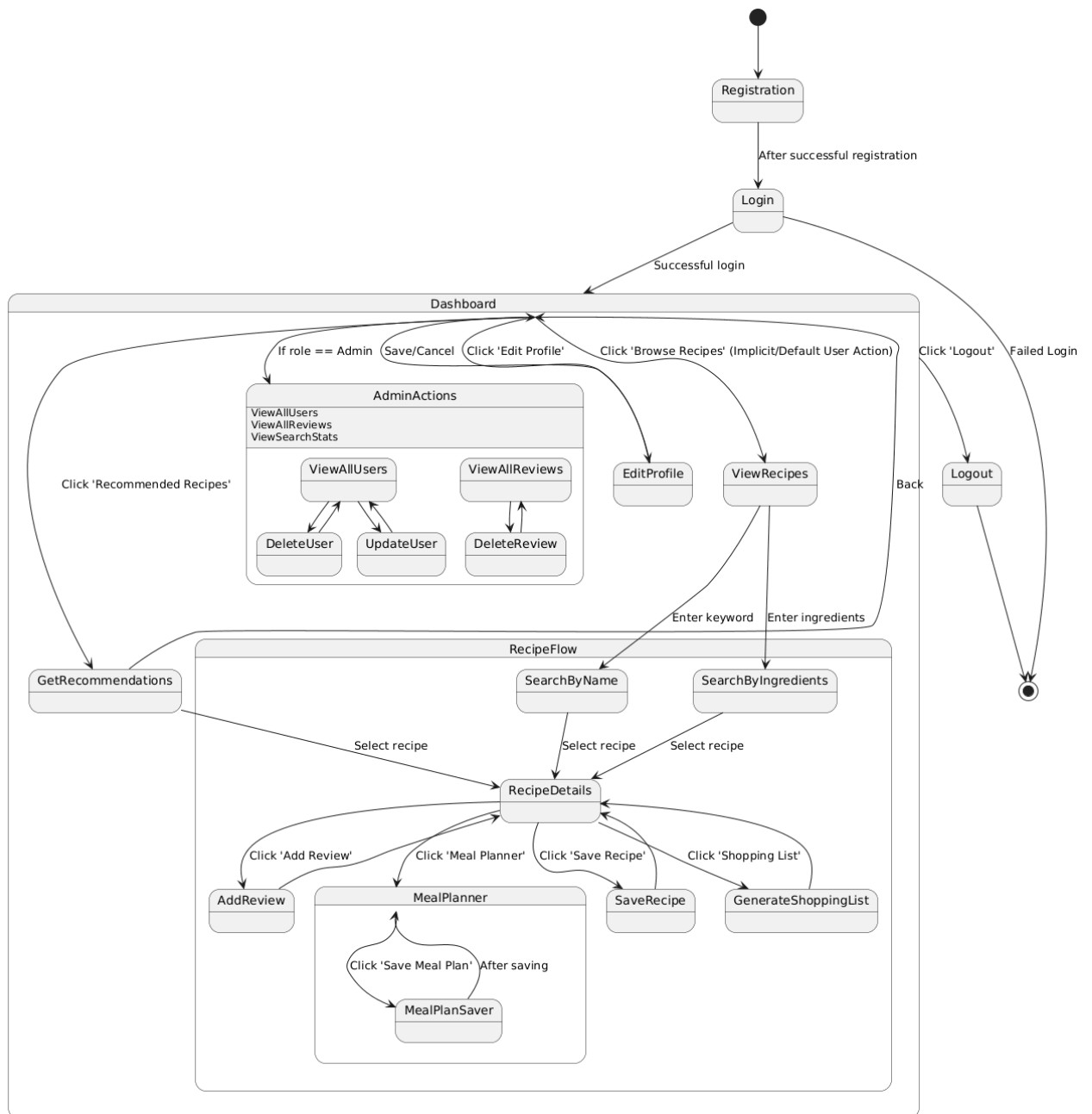


Fig 4.15 – State Diagram

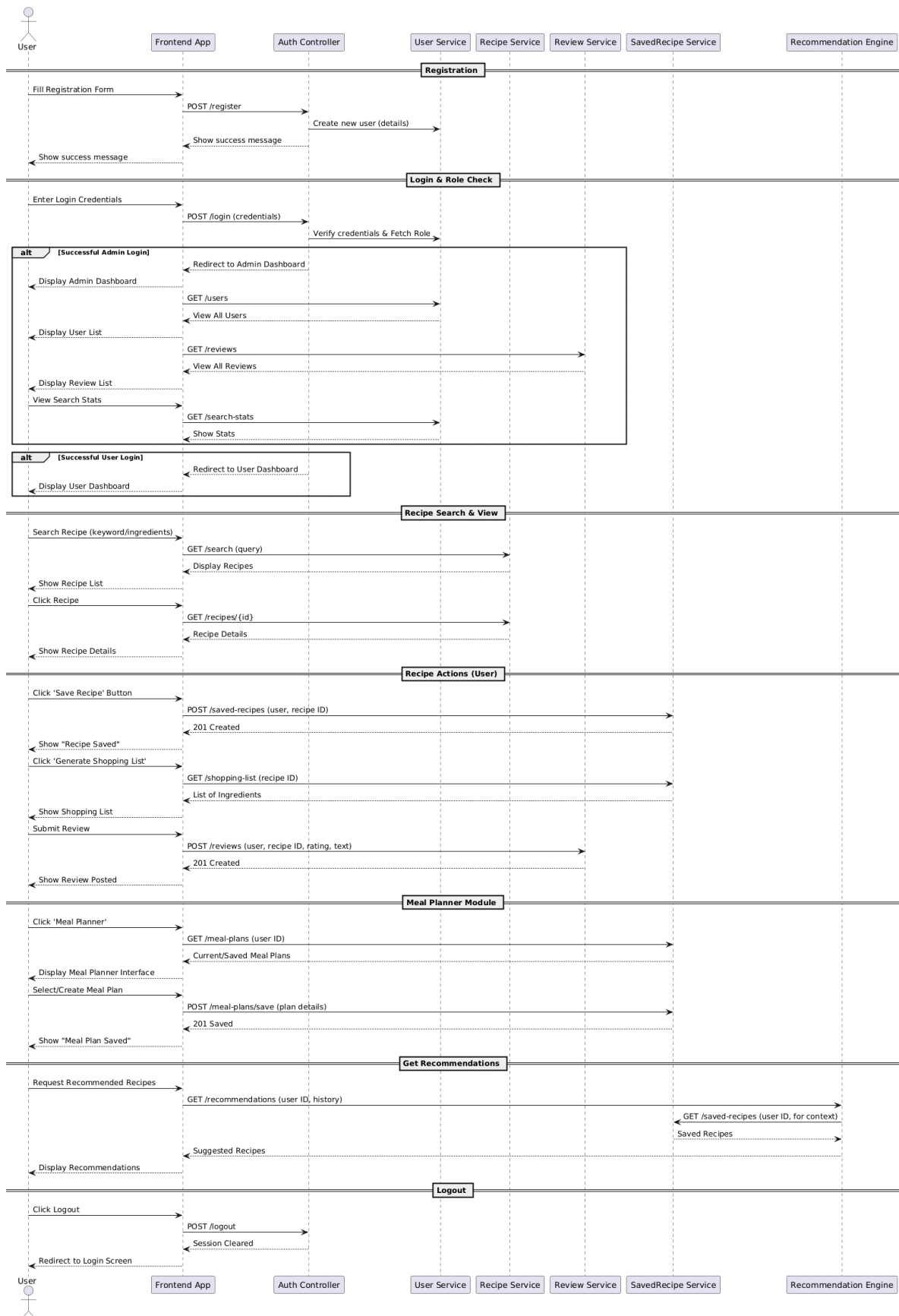


Fig 4.16 – Sequence Diagram

4.5.3 Deployment View:

The Deployment View of the RecipeFusion platform shows the distribution and hosting of the core software components in the various physical and cloud based infrastructure layers to provide high availability, performance as well as scalability.

Project of the system was created on the basis of React.js, Node.js and express.js and is deployed on the cloud hosting service like Vercel which provides responsive implementation to the user device. The layer allows users to engage with such features as ingredient-based search of recipes, personalized recipes, dietary filters, and dashboards to manage recipes.

A mongo database is hosted on mongoDB Atlas cloud database service that contains user data, recipe data, reviews, search history and saved recipes. It guarantees high availability data access that is reliable, secure with high replication capacity.

The deployment infrastructure contains environmental configuration management, secure communication based on HTTPS and role-based access control, the data integrity and safe communication between the user and the system are guaranteed. The use of load balancing and auto-scaling where appropriate are used to keep the performance within the range of traffic loads.

This deployment diagram shows the operating footprint of RecipeFusion in the cloud ecosystem with the focus on its modularity, scaling to the user-friendly bonds formed in the browsers and devices.

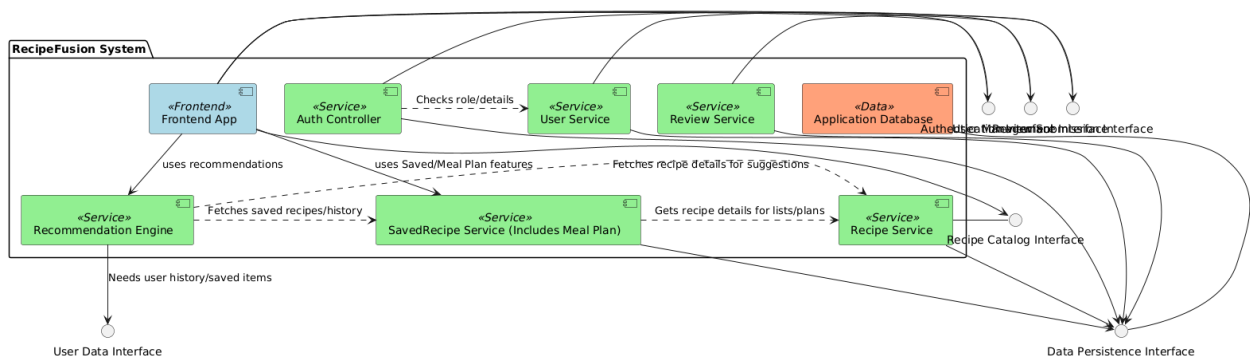


Fig 4.17 – Component Diagram

4.5.4 Physical View:

The Physical View of the RecipeFusion platform provides an overview of the underlying hardware, cloud infrastructure and network components that support the

operation of the system. RecipeFusion is implemented using a fully cloud-based approach for optimum performance, scalability and access.

The frontend interface, which is developed with the help of React.js, is hosted on platforms which are cloud-based, like Vercel that distribute static assets across the global content delivery networks (CDNs) to provide fast and responsive user experiences across both desktop and mobile devices.

The backend services using Node.js and Express.js are deployed on the cloud infrastructure Vercel. These services handle API requests, business logic, authentication, integration with external services, such as cloud media storage, and integration with recommendation logic.

The Mongo DB database is hosted on the basis of Mongo DB Atlas, a fully-managed cloud-based database cuckoo service that provides high availability, backup, encryption and automatic scaling. It contains important data such as user profile, recipes, ingredients, reviews, and search history.

All the communication between clients and servers is secured by using the protocol called the https protocol. Environment Variables and Secrets: Environment variables/secretes are securely handled using the configuration in our cloud platform in order to safeguard sensitive credentials and API keys.

The physical infrastructure is designed with an emphasis on reliability, redundancy and security that ensures that the RecipeFusion platform remains available, scalable and responsive to all users - from recipe seekers to administrators.

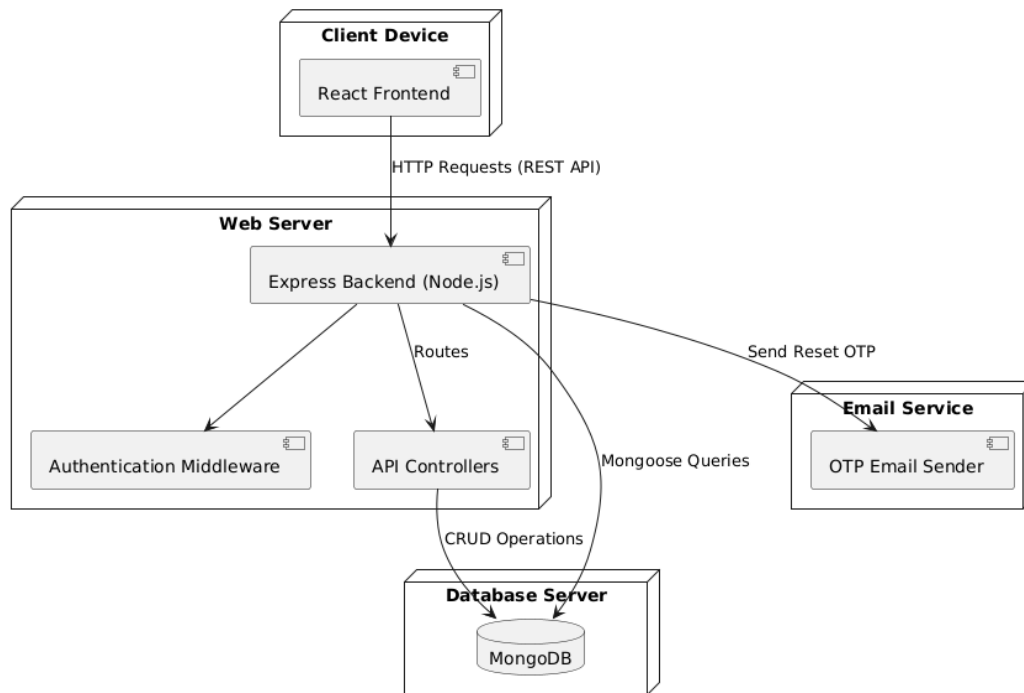


Fig 4.17 – Deployment Diagram

4.6 Entity Relationship Diagram (ERD):

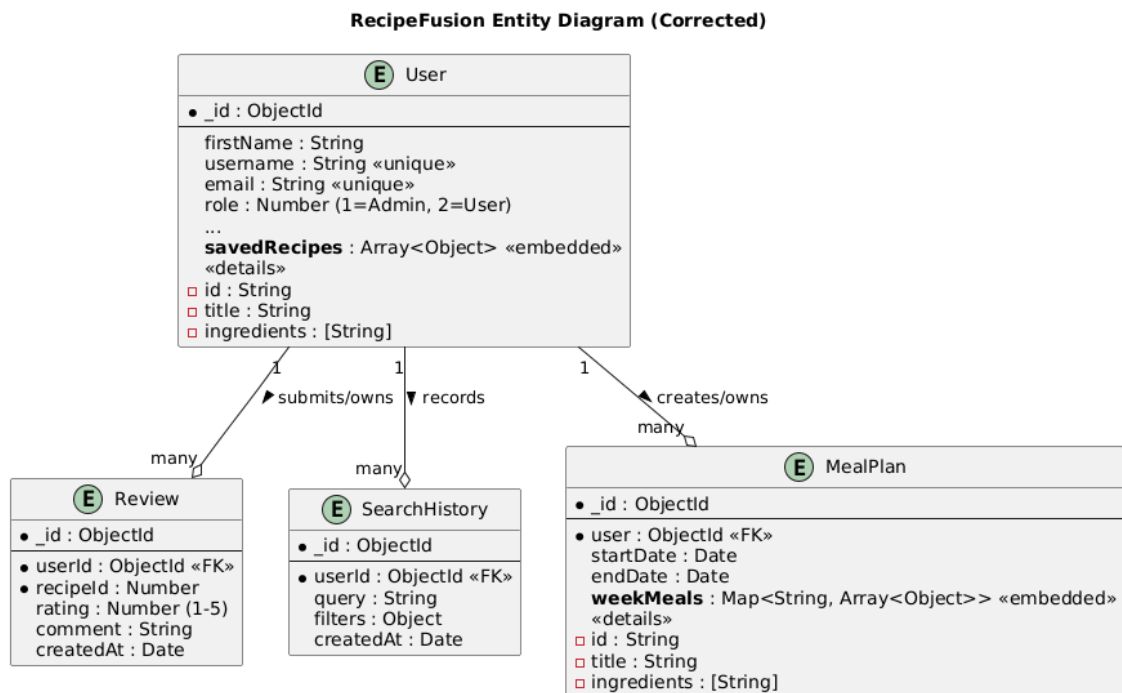


Fig 4.18 – ERD Diagram

4.7 Summary

In this chapter, we introduced the system modeling and design of the RecipeFusion platform, with a focus on the platform's modular system architecture and user-centric strategy with modern web technologies. The chapter started with an introduction to system modeling and a description of the bottom up design approach, which guided the increment of the key modules such as user management, recipe handling and ingredients based search, and personalized recommendations.

The design of the system is based on the 4+1 View Model of Architecture, which includes tremendous representation of the platform as it is logical, process, development and physical. The view is supplemented by using UML Class Diagram, UML Diagram for Activity, UML Diagram to Sequence, a UML Diagram for Components and deployment diagrams.

These models show how the flow and interaction of one component is related to another such as the user registration login, ingredient-based recipe search, dietary preference-based filtering, review submission, and content in the platform management by admins. The architecture also supports complex features such as nutritional data, review analysis and user behavior tracking.

The chapter ends with the deployment and physical infrastructure of the system, and how cloud-based services, like Vercel, MongoDB Atlas, provide scalability, performance, and secure access for end users and administrators. This is the architecture behind the platform's vision to encourage healthy eating habits and personalized culinary experiences and achieve it through a seamless and smart digital solution.

CHAPTER 5

IMPLEMENTATION

This is a chapter that discusses the entire implementation of the RecipeFusion system, how each of the modules was implemented and combined in order to build a seamless AI-based recipe discovery and meal planning system. The implementation phase is the translation of the conceptual design and requirements into working components by utilizing modern web development technologies. RecipeFusion is written with React and Tailwind CSS on the frontend side, Node.js and Express services on backend, utilizing MongoDB as the database management system and it has been deployed on Vercel for quick global access. Generative AI models are ingrained throughout the system to deal with things such as recipe creation and ingredient-based suggestions, and the intelligent planning of meals.

Each module was implemented separately to ensure clarity, maintainability and ease of testing. Once individual functions were confirmed they were linked together using rest APIs and react state management to achieve a consistent process flow both to users and administrators.

5.1 Modules of the Project:

This section explains some of the core modules which together form the RecipeFusion application. Each sub-module performs a specific set of tasks and contributes to the entire process recipe generation, recommendation, user management, meal planning.

5.1.1 Recipe Recommendation Module:

This module is responsible for the task of recommending recipes based on user preference and available ingredients.

It integrates:

A custom-made Node.js API for analyzing user requests

MongoDB collections having trending recipes

GenAI which gives recommendations for recipes

The system fetches user inputs and feeds those parameters into certain set of filtering rules (e.g. diet type, cooking time, ingredient availability) and produces a curated list of recipes which is shared at the frontend using the responsive React components.

5.1.2 Recipe Generation Module:

This module lets the user create recipes from either the recipe name or a list of available ingredients.

The backend uses GenAI to:

- i. Give all the details with step by step procedure
- ii. Suggest the quantity of ingredients
- iii. Give the calories information of a generated recipe
- iv. Suggest appropriate categories of meals

The recipe goes to the frontend, where the React renders the recipe nicely formatted and styled (Tailwind CSS). Users could save, copy and add the recipe to their meal plan.

5.1.3 Meal Planning Module:

This module offers AI- care weekly meal plans.

It performs:

- i. Dietary preferences analysis (Vegan, vegetarian, low carb, etc.)
- ii. Meals balancing (Calorie based)
- iii. Automatic distribution of breakfast, lunch and dinner

This module keeps a close communication with AI Recipe Generation module to get customized meals for each day of the week. The resulting plan is stored in MongoDB, and can be viewed, edited, and/or regenerated at a later time by the user.

5.1.4 Search Tracking Module:

The results users have looked at and interacted with are stored to personalize the recommendations.

This module logs:

- i. Top searched queries
- ii. AI recipe generation history

This data is stored to a specific collection of MongoDB and is expended through APIs for good user experience over a period of time.

5.1.5 User Authentication Module:

The authentication module provides security access to all users.

It includes:

- i. Encryption of credentials
- ii. JWT token generation
- iii. Session handling using cookies
- iv. Role- based redirection

React Hook Form is used to control input validation and Node.js is responsible for signup, login, and logout initialization and session verifications by structured APIs.

5.1.6 User Management Module (Admin):

This module allows the admin's panel to manage accounts of users.

Key actions include:

- i. Viewing registered users
- ii. Editing or remove any user(if necessary)

It ensures that the platform is secure, organized and as per the guidelines of the community.

5.1.7 Review and Rating Module

This module is the modules which handle user response on recipes. It enables them to leave their reviews, to rate meals and to add to the community-level quality upgrades.

Functions include:

- i. Enabling users to post recipes reviews
- ii. Saving ratings and comments

- iii. Showing the average ratings at recipe page.

Allowing the administrator to access, delete or control the uncivilized reviews

All the data of reviews is stored on MongoDB and obtained by means of secure Node.js APIs. This module assists in keeping the quality in recipes and also increases recommendations accuracy because user preferences are analyzed over time.

5.1.8 Saved Recipes and Meal Plans Module:

The personalization of the users is managed in this module as the user can save recipes and save AI-generated meal plans on their profile.

It supports:

- i. Storing favorite recipes and easy access to those recipes.
- ii. Saving meal plans obtained on a weekly basis, developed by AI.
- iii. Giving users an opportunity to remove the stored recipes and meal plans.
- iv. Saving information that is synchronized on devices by using the same account

All the items are saved in MongoDB collections and are dynamically displayed in React with protection routes. This module makes the process of cooking and planning more convenient due to its personalized approach.

5.1.9 Analytics and Insights Module (Admin):

The module is beneficial by equipping administrators with information regarding the use of platforms. It applies both MongoDB data and Node.js APIs to create visual reports which are shown on the panel of the administrator.

Key actions include:

- i. Showing a graph of the history of recipe searches.
- ii. Showing the overall reviews statistics

React charts and track user behavior give Admin these insights on how to improve the platform.

5.2 Summary:

In this chapter, the author broke down the implementation process of the RecipeFusion in detail. Every single module was designed in terms of modularity, performance, and the user-friendly interaction. Through combining the applications of cloud deployment, MongoDB to handle scalable storage, Node.js to provide API services, and React with Tailwind CSS to make the interface dynamic, RecipeFusion provides a cooking assistant which is highly responsive and smart.

Combination of AI-based recipe generation, ingredient-guided suggestions, meal planning, and administrative level of management means a wholesome and contemporary solution to the users who would love to have convenience, customization, and healthy eating assistance. The modules cooperate to build some unified platform that effectively echoes the aims, which it was planned at the very beginning of the work.

CHAPTER 6

TESTING, ANALYSIS AND VALIDATION

This chapter gives the test processes, processes and outcomes applied to test the functional performance and general reliability of RecipeFusion system. Being an AI-powered recipe recommendation and meal-planning system, RecipeFusion needs to be thoroughly validated to provide accurate recipe creation, properly generated recommendations, stability of back-end functions, smooth front-end interactions, and valuable user authentication.

The testing stage was aimed at evaluating the functionality of the individual modules, the relationship between the different parts of the system as well as how the end user experience was as envisioned in the project objectives. Structured testing methods were used to test both the functional and non-functional factors.

6.1 Introduction to Testing:

The testing process is a critical phase in the software development process where it is checked whether or not the implemented components can meet the specifications of the systems and the requirements stated by the users. In the case of RecipeFusion, testing was done to confirm the accuracy of recipes created by the AI, the accuracy of the meal-planning module as well as responsiveness of the user interface and the security of the authentication capabilities. The aim was to ensure that the system is effective, easy to use and it can manage real world use without breaking down.

6.2 Testing Methods Used:

Several methods of testing were conducted to be able to rate the application fully. Every approach was used to legitimize some functionality, integration, or usability of the system.

6.2.1 Black Box Testing:

Black box testing Black box testing was employed to test the application of the system without looking at its internal code. The testers were able to engage with the application and make bets with inputs and note outputs; this was to verify that recipe generation, search filters, and creation of meal plans all yielded results that were correct.

6.2.2 Functional Testing:

Functional testing was aimed at ensuring that every module was delivering its

assigned duties as it should. Additional functions, including recipe recommendation, search and submit review, log in, and shopping list were evaluated with predetermined cases of inputs.

6.2.3 Integration Testing:

As RecipeFusion is based on various interdependent elements, such as frontend interfaces, back-end APIs, database storage, and generative AI, the process of integration testing tested the network between these elements. This was in place to guarantee the flow of information between React frontend and Node.js server and MongoDB was done correctly.

6.2.4 Usability Testing:

Usability testing entailed how the users interacted with the interface. The participants rated the design by its clarity, usability, the ability to read the generated recipes, and design responsiveness. This assisted in supporting the fact that it is still an intuitive system that is user friendly.

6.3 Test Environment:

The following hardware and software set-up were used to perform testing:

Processor: Core i5 (minimum required) by Intel

RAM: 4 GB or above

Storage: 128 GB SSD

Operating System: Windows 10 / Windows 11

Browser: Google Chrome (newest version)

Backend Environment: Node.js and Express

Frontend & Deployment: React and Tailwind CSS hosted on Vercel

Database: MongoDB Atlas

A steady internet connection was needed, specifically with generative AI responses and API requests.

6.4 Test Cases and Results:

The testing process is a critical phase in the software development process where it is checked whether or not the implemented components can meet the specifications of the systems and the requirements stated by the users. In the case of RecipeFusion, testing was done to

confirm the accuracy of recipes created by the AI, the accuracy of the meal-planning module as well as responsiveness of the user interface and the security of the authentication capabilities. The aim was to ensure that the system is effective, easy to use and it can manage real world use without breaking down.

6.4.1 Test Case 1 - User Registration Module

This module provides an effective way of ensuring that once new users want to create an account, they will be able to do so by defining the necessary information about personal preferences and the necessary information. It confirms the accuracy of the input fields like name, username, email, password, gender, age, height, weight, preferences in cuisine and allergies. The process of registration can be segmented into two steps and this test case confirms both the form level and backend level validation. It makes sure that all the verified user data is safely stored in MongoDB whereas invalid data elicit relevant error messages. This aims at ensuring system integrity and a smooth onboarding process of new users.

Table 6.1: Test Case 1 – User Registration Module

INFORMATION IN GENERAL			
Test Stage	Unit User Registration		
Date of Test	20-11-2025	System Date, if applicable	20-11-2025
Testers	Tayyaba Ejaz	Test Case Number	TC-01
Description	Checks that user can be registered with reasonable credentials and invalid input attracts appropriate response		
Test Result	Pass	Incident number (if Applicable)	N/A
INTRODUCTION			

Requirement(s) to be tested	The user must have an option of registering with their name, username, email, password and a little more. The system should be able to make input validation and record of user to MongoDB
Roles and Responsibilities	Tester verifies by means of validation and backend response. Fix, set DB or validation error in case it has been found
Stop Procedures	Failure to pass the test is caused by failures on repeated validation, severe failures, and write errors in the DB
ENVIRONMENTAL NEEDS	
Hardware	Laptops, Desktops or Cellular Phones
Software	React Frontend, Node database, MongoDB
Procedural Requirements	The back-end server should be active, the internet should also be on, and the frontend should be available
TEST	
Test Items and Features	UI-form registration, backend registration validation, password regulations, (detect possible) duplicate e-mail
Input Specifications	Name, Username, Email, Gender, Phone number, Password, Height, Weight, Age, Cuisine preference, Allergies and Health condition
Procedural Steps	<ol style="list-style-type: none"> 1. Navigate to /register route 2. Type in valid user details in form 1 3. Re-tests on invalid inputs(missing field, weak password, etc) 4. The next option is to move on to form 2 by clicking next 5. Click on all credentials to create 2 6. Submit the form 7. Re-test with the means of validations
Expected Results of Case	<p>Registering successful valid input</p> <p>Invalid error message on input fields</p>
ACTUAL RESULTS	

Output Specifications	All tests were effectively passed, as well as Valid registrations saved. The invalid inputs were accompanied by how they should have.
------------------------------	---

6.4.2 Test Case 2 - Login & Role based Access Module

In this module, one is able to retrieve recipes by using the available ingredients at home. The system receives the input of ingredients provided by the user and processes them and retrieves the appropriate recipes through an AI-based recommendation logic. The test case allows verifying that the ingredient parser is functioning properly, the back end is properly filtering recipes, and only the information that is of interest is displayed. It also tests the behavior of the system in circumstances where none of the recipes correspond to the provided ingredients to make sure that there is appropriate error or fallback management.

Table 6.2: Test Case 2 – Login and Role based Access Module

INFORMATION IN GENERAL			
Test Stage	Unit User Login and redirect to respective panel		
Date of Test	20-11-2025	System Date, if applicable	20-11-2025
Testers	Tayyaba Ejaz	Test Case Number	TC-02
Description	Authenticates that the user is able to log in with valid credentials and will be redirected depending on his or her role(user/ admin)		
Test Result	Pass	Incident number (if Applicable)	N/A
INTRODUCTION			
Requirement(s) to be tested	System is required to check the validity of a user and access to authorized roles only.		

Roles and Responsibilities	Tester confirms the flow of logins and access, developer solves the problems with incorrect routing or tokens.
Stop Procedures	Stop in case of frequent failures in making the login because of problems with the server or database.
ENVIRONMENTAL NEEDS	
Hardware	Laptop, Desktop, or Mobile device
Software	React Frontend, Node.js backend, MongoDB Database
Procedural Requirements	Backend server should be operational, have access to the internet.
TEST	
Test Items and Features	UI form of the login form, token generation and role based routing
Input Specifications	Email and Password
Procedural Steps	<ol style="list-style-type: none"> 1. Navigate to /login route 2. Enter valid credentials 3. Try invalid credentials too 4. Try login to admin dashboard
Expected Results of Case	Accepted Results of Case Valid input into the system via successful login Invalid error message on input fields
ACTUAL RESULTS	
Output Specifications	Output Specifications All flows were as expected

6.4.3 Test Case 3 - Ingredient Based Recipe Search

In this module, one is able to retrieve recipes by using the available ingredients at

home. The system receives the input of ingredients provided by the user and processes them and retrieves the appropriate recipes through an AI-based recommendation logic. The test case allows verifying that the ingredient parser is functioning properly, the back end is properly filtering recipes, and only the information that is of interest is displayed. It also tests the behavior of the system in circumstances where none of the recipes correspond to the provided ingredients to make sure that there is appropriate error or fallback management.

Table 6.3: Test Case 3 – Ingredient based Recipe Search

INFORMATION IN GENERAL			
Test Stage	Unit		
	Ingredients search Recipe search		
Date of Test	20-11-2025	System Date, if applicable	20-11-2025
Testers	Tayyaba Ejaz	Test Case Number	TC-03
Description	Verifies that customer is capable of searching recipe with available ingredients as input (using commas to separate the ingredients)		
Test Result	Pass	Incident number (if Applicable)	N/A
INTRODUCTION			
Requirement(s) to be tested	System should be able to filter recipes on the basis of ingredients that are entered by the user		
Roles and Responsibilities	Tester decides how search results are accurate. Mismatches are solved by the developer		
Stop Procedures	Stop Procedures Stop in case results are irrelevant or the server is dead		
ENVIRONMENTAL NEEDS			
Hardware	Laptop, Desktop, or Mobile device		
Software	React Frontend, Node.js backend, MongoDB Database		

Procedural Requirements	Backend server should be operational, internet connection should exist to receive generated recipes by GenAI
TEST	
Test Items and Features	Add ingredients and click the Search button
Input Specifications	Chicken, tomato, etc.
Procedural Steps	<ol style="list-style-type: none"> 1. Navigate to /search route 2. Enter ingredients separated by commas 3. Click the search button 4. Get the generated recipes
Expected Results of Case	<p>Anticipated Findings of Case Relevant recipes displayed upon successful response.</p> <p>No results presented in the case of some error.</p>
ACTUAL RESULTS	
Output Specifications	All the outputs were within the expected behavior

6.4.4 Test Case 4 - Search recipe by Name Module

The module will enable users to retrieve recipes by typing in a particular name or a keyword of the recipe. It communicates with the AI powered recipe generator to give relevant and complete recipe suggestions. The test case confirms the accuracy of search query in bringing out the right recipe, the system is able to respond to various forms of queries and the results presented before user are useful. It also ascertains there is error handling where a query is either unclear, invalid or where there are network/API problems during recipe generation.

Table 6.4: Test Case 4 – Search recipe by Name

INFORMATION IN GENERAL			
Test Stage	Unit		
	Recipe Search by name		
Date of Test	20-11-2025	System Date, if applicable	20-11-2025
Testers	Tayyaba Ejaz	Test Case Number	TC-04
Description	Confirms that GenAI works out recipes according to the query of a user.		
Test Result	Pass	Incident number (if Applicable)	N/A
INTRODUCTION			
Requirement(s) to be tested	AI must produce full recipes.		
Roles and Responsibilities	Tester approved the quality of recipes produced, API issues are dealt with by the developer		
Stop Procedures	Stop if internet connection has gone dead		
ENVIRONMENTAL NEEDS			
Hardware	Laptop, Desktop, or Mobile device		
Software	React Frontend, Node.js backend, MongoDB Database		
Procedural Requirements	Backend server must be running, internet connection must be active to get generated recipes by GenAI		
TEST			
Test Items and Features	Search box with search button		

Input Specifications	Input such as query like Pasta.
Procedural Steps	<ol style="list-style-type: none"> 1. Navigate to /search route 2. Enter query for the recipe 3. Press search button 4. Get the generated recipes
Expected Results of Case	Relevant recipes displayed upon successful response. No results presented in the case of some error
ACTUAL RESULTS	
Output Specifications	All the outputs were within the expected behavior

6.4.5 Test Case 5 - Filtering on Recipes Module

The current module also will improve the search experience of the users, as they can narrow down the list of recipes according to the cuisine type, cooking duration, and level of difficulty. It makes sure that the opposing actions the filtering specifications are appropriately introduced to the created or fetched recipes and that only the matching results are retrieved. The test case will also make sure that filter logic is implanted properly, and users get clean, sorted and relevant results as per their choices. It also analyzes the reaction of the system in cases where the filters used do not find any matching recipes.

Table 6.5: Test Case 5 – Filtering on Recipes Module

INFORMATION IN GENERAL			
Test Stage	Unit Recipe Filters		
Date of Test	20-11-2025	System Date, if applicable	20-11-2025
Testers	Tayyaba Ejaz	Test Case Number	TC-05

Description	Ensures that it has created recipes based on the query made by the user and the filters have been applied		
Test Result	Pass	Incident number (if Applicable)	N/A
INTRODUCTION			
Requirement(s) to be tested	Bug(s) to test AI should form recipes in which filtering items added by a user are used.		
Roles and Responsibilities	Tester checked whether the filters are present or not on the generated recipes		
Stop Procedures	Stop when the internet connection is not available or there is no such recipes with the filters used.		
ENVIRONMENTAL NEEDS			
Hardware	Laptop, Desktop, or Mobile device		
Software	React Frontend, Node.js backend, MongoDB Database		
Procedural Requirements	Backend server should be available, the internet should be active to receive recipes generated by GenAI.		
TEST			
Test Items and Features	Searching option with filters Cuisine type, highest time of cooking, difficulty level		
Input Specifications	Search as Biryani		
Procedural Steps	<div>1. Navigate to /search route</div> <div>2. Enter query for the recipe</div> <div>3. Apply filters for cuisine, max cooking time, difficulty level</div> <div>4. Press search button</div> <div>5. Get the generated recipes</div>		

Expected Results of Case	Anticipated Findings of Case Relevant recipes displayed upon successful response. No results presented in the case of some error
ACTUAL RESULTS	
Output Specifications	Output Specifications All the outputs were within the expected behavior

6.4.6 Test Case 6 - View Complete Recipe Details

The module interface shows all the information about a chosen recipe and offers its user to have all the details, including the list of ingredients, preparation time, difficulty level, nutrition table, and photographs. The test case will also help in making sure that the recipe detail page loads successfully and that all the necessary fields have been included and are well designed accordingly. It also checks the system to ensure all the information is available in the backend and that the absent or partial data is managed respectively.

Table 6.6: Test Case 6 – View Component Recipe Details

INFORMATION IN GENERAL			
Test Stage	Unit Recipe detail page		
Date of Test	20-11-2025	System Date, if applicable	20-11-2025
Testers	Tayyaba Ejaz	Test Case Number	TC-06
Description	Checks that recipes generated contain all the needed details or not.		
Test Result	Pass	Incident number (if Applicable)	N/A
INTRODUCTION			
Requirement(s) to be tested	On the generated recipe, there should be recipe title, picture, description, etc.		

Roles and Responsibilities	Tester errors by taking care as to whether the recipe is progressing with everything or not.
Stop Procedures	Stop in case the internet connection is dead was not able to locate the detail page of the chosen recipe.
ENVIRONMENTAL NEEDS	
Hardware	Laptop, Desktop, or Mobile device
Software	React Frontend, Node.js backend, MongoDB Database
Procedural Requirements	Backend server should be in operation, internet connection should be stable
TEST	
Test Items and Features	Recipe detail page must include title of the recipe, image, description, ingredients, procedure, calories, level of difficulty, cuisine type and cooking time
Input Specifications	Click Card of a created recipe
Procedural Steps	<ol style="list-style-type: none"> 1. Navigate to /search route 2. Enter query for the recipe 3. Get the generated recipe cards 4. Click on the card to navigate to the recipe detail page 5. Get all the details of selected recipe here
Expected Results of Case	The ingredients of the chosen recipe appear on recipe detail page
ACTUAL RESULTS	
Output Specifications	All the outputs were in line with the expectations

6.4.7 Test Case 7 - Meal Plan Generation

The current module creates a customized meal plan each week, and it is designed on a weekly basis to serve its users according to their health history, dietary requests, and food allergies. With AI-powered reasoning, it comes up with a structured 7-day plan which has breakfast, lunch, and dinner program per day. The case study confirms that the created plan is comprehensive, balanced, and user-preferred. It also looks at the presence of the correct details in all recipes in the plan and makes sure that the system is able to process any error, including the API failure, or missing data about the user in a graceful manner.

Table 6.7: Test Case 7 – Meal Plan Generation

INFORMATION IN GENERAL			
Test Stage	Unit		
	Meal Planner		
Date of Test	20-11-2025	System Date, if applicable	20-11-2025
Testers	Tayyaba Ejaz	Test Case Number	TC-07
Description	Checks the creation of meal plan by the user data		
Test Result	Pass	Incident number (if Applicable)	N/A
INTRODUCTION			
Requirement(s) to be tested	Meal plan has to contain planned breakfast, lunch and dinner		
Roles and Responsibilities	Tester checks completeness		
Stop Procedures	Stop In case plan is not created or generated.		
ENVIRONMENTAL NEEDS			
Hardware	Laptop, Desktop, or Mobile device		

Software	React Frontend, Node.js backend, MongoDB Database
Procedural Requirements	Backend server must be running, internet connection must be stable.
TEST	
Test Items and Features	Meal plan is of all 7 days which has 21 total meals
Input Specifications	Click the button of generate meal plan
Procedural Steps	<ol style="list-style-type: none"> 1. Navigate to /mealPlan route 2. Click the button of generate new meal plan 3. Get the generated meal plan of 7 days 4. Click on the card to navigate to the recipe detail page included in the plan 5. Get all the details of selected recipe here 6. Can save the recipe or meal plan too
Expected Results of Case	he results of Case to be expected 7-day meal plan with three meals in a day breakfast, lunch, and dinner
ACTUAL RESULTS	
Output Specifications	Output Specifications All the outputs were in line with the expectations

6.4.8 Test Case 8 - Save a Recipe

This module allows the user to save the recipes they like or generate as their favorites to be accessed with a lot of ease in the future. It verifies that the recipe that has been chosen is properly saved on the user- saved recipes list on the database. In the test case, the frontend and backend are tested to ensure interaction to confirm that the save action is reliable, demonstrates the right success messages, and does not save the same item twice. It also ensures that recipes that are saved are shown in the proper way when revisited using the saved recipes page.

Table 6.8: Test Case 8 – Save a Recipe

INFORMATION IN GENERAL			
Test Stage	Unit Saving recipe to profile		
Date of Test	20-11-2025	System Date, if applicable	20-11-2025
Testers	Tayyaba Ejaz	Test Case Number	TC-08
Description	Recipe is saved by a given user		
Test Result	Pass	Incident number (if Applicable)	N/A
INTRODUCTION			
Requirement(s) to be tested	Recipe has to be stored in the profile of the user		
Roles and Responsibilities	Tester checks on the DB writes or write not		
Stop Procedures	Stop if save fails		
ENVIRONMENTAL NEEDS			
Hardware	Laptop, Desktop, or Mobile device		
Software	React Frontend, Node.js backend, MongoDB Database		
Procedural Requirements	Backend server must be running, internet connection must be stable.		
TEST			
Test Items and Features	Recipe saved or not		

Input Specifications	Click on save recipe button
Procedural Steps	<ol style="list-style-type: none"> 1. Navigate to /recipeDetail route 2. Click the button of save recipe 3. Get the success message of Recipe Saved Successfully 4. Recipe will be added to the saved recipe page from where it can also be removed
Expected Results of Case	Recipe saved successfully popup shows
ACTUAL RESULTS	
Output Specifications	Output Specifications All the outputs were in line with the expectations

6.4.9 Test Case 9 - Save a Meal Plan

The module allows the user to save a full weekly meal plan, which he/she would need to refer to later. It is used so that when the users create a food plan, they would be able to save it under their profile. Test case will confirm that meal plans have been correctly written to the database, connected with the right user ID, and can be accessed through the plans that have been saved section. It also verifies successful save responses and correct error responses in the event of save failure because of network problem or backend problem.

Table 6.4.9: Test Case 9 – Save a Meal plan

INFORMATION IN GENERAL			
Test Stage	Unit Saving Meal plan		
Date of Test	20-11-2025	System Date, if applicable	20-11-2025
Testers	Tayyaba Ejaz	Test Case Number	TC-09
Description	Meal plan is a saved one of the particular user		

Test Result	Pass	Incident number (if Applicable)	N/A
INTRODUCTION			
Requirement(s) to be tested	Tester checks Meal Plan should be saved in the particular user		
Roles and Responsibilities	Tester verifies the DB writing of saved meal or not.		
Stop Procedures	Stop in case of save meal plan failure		
ENVIRONMENTAL NEEDS			
Hardware	Laptop, Desktop, or Mobile device		
Software	React Frontend, Node.js backend, MongoDB Database		
Procedural Requirements	Backend server must be running, internet connection must be stable.		
TEST			
Test Items and Features	Meal Plan is saved or not		
Input Specifications	Click the button of Save Meal plan		
Procedural Steps	1. Navigate to /mealPlan route 2. Click the button of save meal plan 3. Get the success message of Meal Plan Saved Successfully		
Expected Results of Case	Meal Plan Saved Successfully Meal Plan Success Message appears		
ACTUAL RESULTS			

Output Specifications	Output Specifications All the outputs were in line with the expectations
------------------------------	--

6.4.10 Test Case 10 - Admin Manage Users and Reviews

This module enables the administrator to control users of the platform and reviews left on recipes. It also has the features to see all the registered users, update user details, delete the inappropriate users, view user reviews and delete abusive or irrelevant reviews. Test case helps in assuring that the privileges of an admin are enforced properly, data loads, all the CRUDs are carried out as desired. It also certifies access control, consistency and error tolerance in case of failover of the backend operations by the administrator.

Table 6.4.10: Test Case 10 – Admin Management

INFORMATION IN GENERAL			
Test Stage	Unit Admin Management		
Date of Test	20-11-2025	System Date, if applicable	20-11-2025
Testers	Tayyaba Ejaz	Test Case Number	TC-10
Description	Verifies the permission of the admin to see edit or delete user. Admin is authorized to read reviews and delete them		
Test Result	Pass	Incident number (if Applicable)	N/A
INTRODUCTION			
Requirement(s) to be tested	Users and reviews must exist in admin page		
Roles and Responsibilities	Tester has to check the functionality of the operations at the admin page or not		

Stop Procedures	Stop when the data load of the administrator fails
ENVIRONMENTAL NEEDS	
Hardware	Laptop, Desktop, or Mobile device
Software	React Frontend, Node.js backend, MongoDB Database
Procedural Requirements	Backend server must be running, internet connection must be stable.
TEST	
Test Items and Features	Admin is reading users and reviews, must be able to delete
Input Specifications	Login with the credentials of admin
Procedural Steps	<ol style="list-style-type: none"> 1. Navigate to /adminDashboard route 2. See the total users, option to edit and delete on Mange Users page 3. See the total reviews and option to delete on Mange Reviews page
Expected Results of Case	All the details appear
ACTUAL RESULTS	
Output Specifications	All the outputs were in line with the expectations.

6.5 Summary:

The chapter was the introduction of testing procedures and results of RecipeFusion system. With structured software testing techniques, such as black-box testing, functional testing, integration testing and usability review, every one of the modules was tested on high accuracy, stability and performance. The findings underscore the idea that the system is achieving its target goals as it has effective recipe suggestions, informative AI-generated meals plans, adequate authentication, and an easy-to-use interface based on all key functions. Even though there are some small drawbacks that may be noted, they do not add to the overall system functionality and usage. According to the results of the testing, RecipeFusion has a high level of performance, preserves data integrity, and is completely ready to be used in end users.

CHAPTER 7

CONCLUSION AND FUTURE WORK

This chapter gives the general conclusion of the RecipeFusion project and gives the possible future improvements which can bring more improvements to the system. The RecipeFusion development was to build an intelligent, user-friendly service that will be able to write a recipe and suggest a dish based on the available ingredients and give meal planning with the help of AI to help consumers to eat healthier and more organized. The system is effective in achieving its goals, through the structured design, the use of modern technologies, and a comprehensive testing.

7.1 Project Review and Project Achievements:

RecipeFusion was created as an intelligent, Generative AI-driven recipe suggestion and a meal-planning platform meant to overcome some of the most prevalent issues that one faces in their everyday cooking habits. Most of the users have challenges in determining what to cook and get recipes that best suit the available ingredients and the planning of balanced food. RecipeFusion is a good answer to these demands because it combines the AI-powered generation of recipes, suggestions, food planning, user reviews, and customized dashboards into one integrated system.

The React and Tailwind CSS were very useful in creating a responsive and modern user interface that the system is easy to navigate even though the user is not technical. The back-end in the form of Node.js and Express provided a scalable and efficient platform to handle the AI requests, user authentication, user stored data, and administration functions. In the meantime, MongoDB Atlas provided a secure and organized storage of the reviews and user accounts, meal plans, and search history. The ability to host the platform on Vercel also increased its performance due to the rapid access of the world and easy deployment processes.

The implication of the integration of the system with Generative AI is that the capabilities of the former are greatly boosted because they allow automatic recipe generation and ingredient-based search as well as AI-based plans of meals. As a result of comprehensive testing, such as functional testing, integration testing and usability testing, RecipeFusion was found to be reliable in its operation in all the modules. Basic ones like registering users, tracking searches, recipes, reviewing, and/or running analytics were all handled well and delivered the rightful results.

On the whole, the project has been able to fulfill its expectations of providing a

convenient, smart, interactive cooking helper which helps promote healthier eating habits, decreases food waste, as well as makes it easier to plan meals. RecipeFusion demonstrates that AI-based applications could be used to support the daily habits of users intelligibly provided they are designed and deployed adequately.

7.2 Limitations:

Although RecipeFusion is a working service some weaknesses were observed:

Reliance on Internet Connection: API requests and the generation of AI require a consistent connection to the Internet, affecting the speed of response in case of a weak network.

Variability of Nutritional Accuracy: Nutritional values delivered by AI do not always coincide with the actual values.

Limited Interaction Features: Services such as voice search, being multilingual or social sharing are not available.

Rate Limit of External API: When external API utilization is involved the request limitation can impact the reliability of the system.

These constraints do not interfere with usability of the systems but indicate the aspects that can still be enhanced in the subsequent versions.

7.3 Future Works:

In order to improve RecipeFusion more and expand its functionality, it is possible to add to it several improvements in the future version:

7.3.1 Advanced Personalization Using Machine Learning:

This suggests that message personalization can be automated using machine learning, enabling the system to recognize individual users' preferences once they have viewed multiple pages and responded to specific messages or anticipated outcomes.

7.3.2 Mobile Application Development:

The platform should be made more available by having a specific Android and iOS mobile application that can be used daily at the kitchen.

7.3.3 Multilingual Support:

The implementation of two or more languages would aid in targeting a worldwide customer base and assist the consumers in various locations.

7.3.4 Enhanced Nutritional Analysis:

Interference with specialization nutrition APIs might enhance accuracy in the number of calories and macronutrient/micronutrient contents.

7.3.5 Voice Enabled Assistant:

A voice activated interface would enable a user to cook freely as he or she follows instructions step by step.

7.4 Summary:

To sum up, RecipeFusion adeptly shows how the collaboration of AI and up-to-date web technologies and user-friendly design can help create a valuable digital cooking assistant. It also makes planning meals easier, enhances recipe identification and helps make healthier eating choices. Even though there are some shortcomings, the system is ready to be upgraded as needed, and further into a more sophisticated intelligent meal management platform. RecipeFusion could be developed to be an integrated solution to daily food preparation needs with constant advances.