MINISTRY OF EDUCATION AND RESEARCH OF REPUBLIC OF MOLDOVA
TECHNICAL UNIVERSITY OF MOLDOVA
FACULTY OF COMPUTERS, INFORMATICS AND MICROELECTRONICS
DEPARTMENT OF SOFTWARE ENGINEERING AND AUTOMATICS

Kitchen Guru: Web application for food recipes

Project Report

Mentor: prof., Arama Radu-Vasile
Students: Bostan Victor, FAF-222
Dimbitchi Sergiu, FAF-222
Doschinescu Dan, FAF-222
Grosu Renat, FAF-222
Ostafi Eugen, FAF-222

Chișinău, 2023

Abstract

The project entiteled ”KitchenGuru” was developed by Bostan Victor, Dimbitchi Sergiu, Doschinescu Dan, Grosu Renat and Ostafi Eugen, enrolled in the Software Engineering program at the Technical University of Moldova. ”KitchenGuru” is an innovative web application designed to redefine home cooking with its user-centric recipe discovery platform and community engagement features. It provides personalized recipe suggestions based on users' available ingredients, as well as an option to generate AI recipes, promoting a sustainable, no-waste approach. The platform also serves as a social hub, allowing users to add feedback to recipes. This report explores Kitchen Guru's functionality, emphasizing its role in enhancing the home cooking experience through technology and community. The report is bifurcated into two parts: the first outlines the theoretical framework of the ”KitchenGuru” project, while the second details the practical modeling of the web application during the internship phase, seamlessly bridging conceptual understanding with real-world application.

Keywords: culinary innovation, recipe personalization, sustainable cooking, community engagement, AI integration.

Content

Introduction

In the dynamic world of culinary arts, technology and cooking have merged to create innovative and convenient home kitchen experiences. This report introduces ”KitchenGuru,” a web application designed to revolutionize home cooking. The choice of this topic is driven by the growing need for efficient, sustainable cooking practices and the rising interest in culinary arts across various skill levels. KitchenGuru stands out due to its relevance and novelty in today's era where food sustainability and personalized cooking experiences are increasingly important.

KitchenGuru's primary goal is to simplify recipe discovery and build a community of culinary enthusiasts. Its dual functionality helps users find recipes based on available ingredients and connects them with other cooks for recipe discussions. The app's uniqueness is enhanced by incorporating the OpenAI API, enabling AI-driven recipe suggestions and blending technology with culinary art.

The development of KitchenGuru involved thorough research in current culinary trends, user experience design, and advancements in artificial intelligence. This report will outline the sections, providing insight into the app's journey from concept to implementation, the challenges encountered, and the innovative solutions devised.

KitchenGuru represents a significant advancement in the culinary domain, offering innovative solutions to modern home cooking challenges and setting the stage for future developments in this field.

Part I

PBL Project Part

1 Domain Analysis

1.1 Overview of the domain

The culinary domain is a dynamic and ever-evolving landscape, increasingly influenced by the advent of technology and shifts in consumer lifestyle. In an age where convenience and personalization are paramount, the home cooking experience has become ripe for innovation. As individuals seek to balance busy schedules with the desire for healthy, home-cooked meals, the challenges of meal planning and food waste have come to the forefront [1]. It is within this context that Kitchen Guru positions itself as a trailblazer, offering a technologically advanced solution to the time-old question: ”What should I cook today?”

Kitchen Guru is more than just a recipe site; it is a culinary ecosystem designed to empower home cooks. By integrating an innovative ingredient-based recipe discovery tool, it addresses the all-too-common dilemma of utilizing what is already in the pantry, encouraging creativity and reducing food waste. This approach not only streamlines the meal preparation process but also promotes a more sustainable cooking practice by minimizing unnecessary purchases and utilizing leftovers, resonating with the growing consumer consciousness towards food sustainability [1].

In addition to solving the problem of meal indecision, Kitchen Guru leverages AI-driven recipe suggestions to elevate the user experience. This smart feature not only recommends recipes based on the ingredients at hand but also adapts to user preferences over time, learning from their choices and feedback [2]. This personalized touch reflects the broader trend towards AI in consumer services, where the technology is harnessed to cater to individual tastes and preferences, fostering a sense of connection and understanding between the user and the platform.

The interplay between technology and culinary arts has never been more significant, with a burgeoning demand for digital solutions that offer both functionality and engagement. Kitchen Guru’s ingredient-based search algorithm is a direct response to this demand, simplifying the search process while inspiring users with new and exciting culinary possibilities. It stands at the intersection of culinary innovation and technology, encapsulating the essence of modern home cooking—a blend of tradition and technology, of personal taste and community sharing.

As the domain continues to adapt to new technological capabilities and user expectations, Kitchen Guru’s role becomes increasingly central. It doesn’t merely adapt to the changes; it seeks to be a catalyst for transformation in the culinary space, redefining how people approach cooking at home and tackling issues that resonate with a global audience. The platform’s strategic direction aligns with the market’s trajectory, aiming to deliver a seamless, enjoyable, and sustainable cooking journey for its users.

1.2 Idea Validation

The inception of the Kitchen Guru project is rooted in the pressing problem of optimizing the home cooking experience. In today's fast-paced world, individuals struggle with meal planning, utilizing available ingredients efficiently, and minimizing food waste. This challenge transcends the personal domain, touching upon broader themes of sustainability and efficient resource utilization. A comprehensive understanding of this problem forms the bedrock of our project, guiding our approach and innovation.

The issue at hand impacts multiple domains: culinary arts, environmental sustainability, and consumer lifestyle. In the culinary domain, there's a clear need for simplifying meal preparation and enhancing the cooking experience. Environmentally, the imperative to reduce food waste is paramount, aligning with global sustainability goals. Consumer lifestyles are evolving, with a marked preference for convenience, health, and personalized experiences. Analyzing these domains enables us to gauge the problem's multifaceted impact and develop a tailored solution.

Kitchen Guru is designed for a diverse range of users, united by their common interest in cooking and the challenges that come with it. The primary target group includes busy professionals seeking quick and healthy meal options, sustainability advocates who prioritize eco-friendly cooking practices, and culinary enthusiasts who enjoy exploring new recipes and flavors. This diverse audience shares a common need for a solution that simplifies meal preparation, reduces food waste, and enhances the overall cooking experience. Within this broad demographic, specific sub-groups emerge, each with unique characteristics and requirements, which are exemplified through user personas.

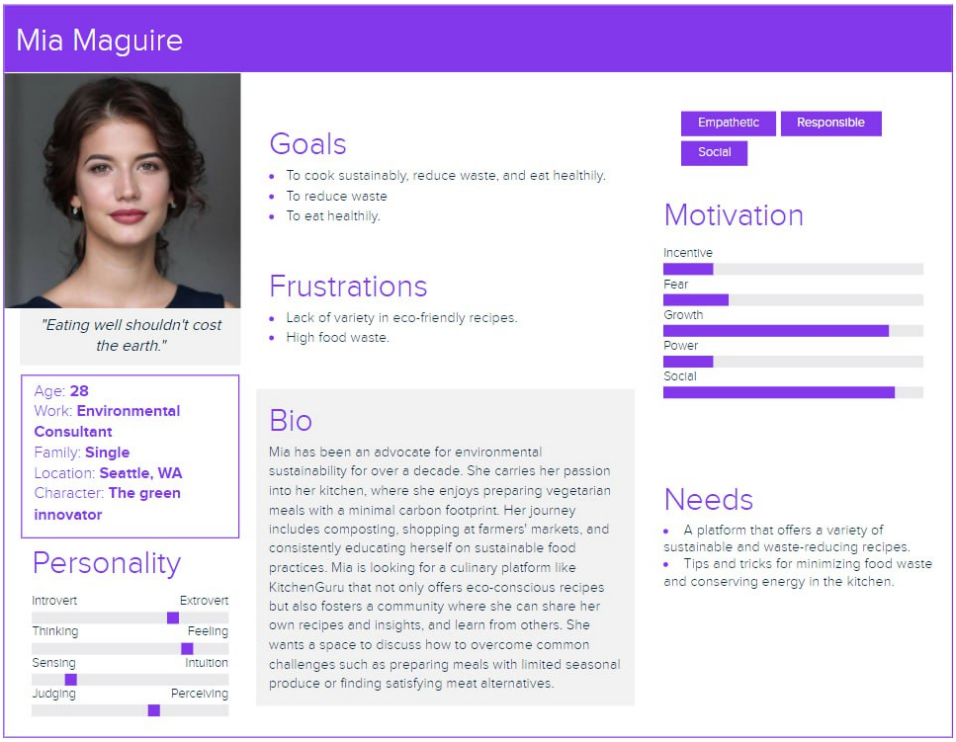

Figure 1.2.1 - Mia Maguire User Persona

Mia Maguire embodies the ”Green Innovator” persona, who is committed to sustainable living and minimizing waste. She is environmentally conscious and seeks to incorporate her passion for sustainability into her cooking habits. Mia finds value in cooking with a minimal carbon footprint, which includes composting and shopping at farmers' markets. She is on the lookout for a culinary platform like Kitchen Guru that offers eco-conscious recipes and fosters a community for sharing insights and learning from others. Her primary goals are to cook sustainably, reduce waste, and eat healthily.

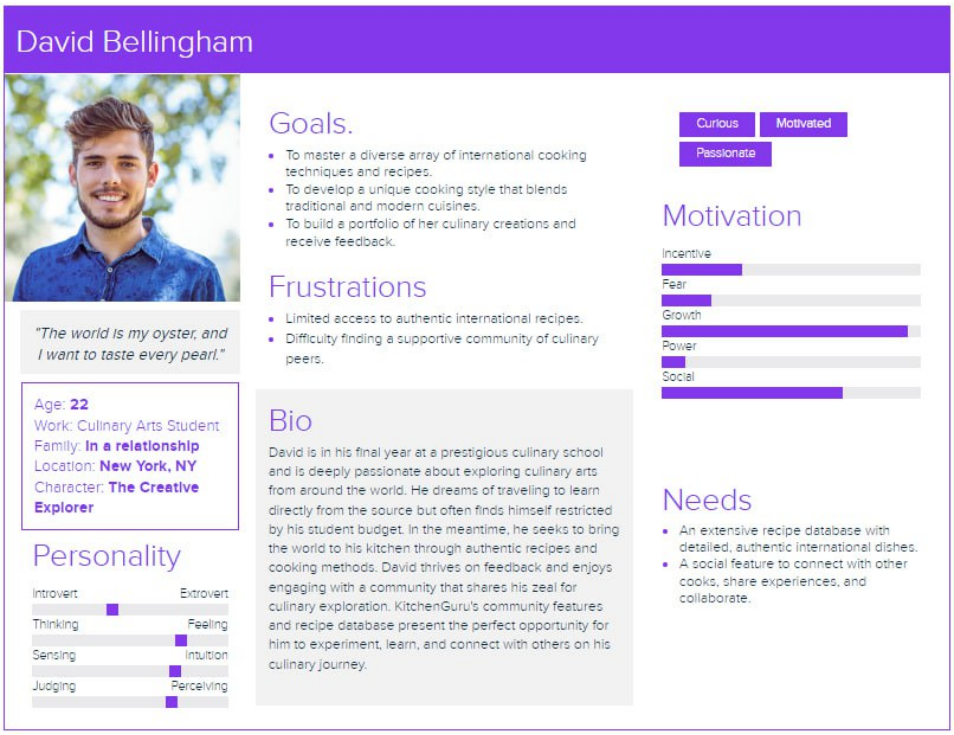

Figure 1.2.2 - David Bellingham User Persona

David Bellingham represents the ”Creative Explorer” persona. As a Culinary Arts student, David is passionate about exploring diverse cooking arts and bringing global flavors into his kitchen. He is eager to connect with a community that shares his zeal for culinary exploration. David seeks a comprehensive recipe database and social features that enable him to share experiences and collaborate with like-minded cooks. His main goals are to master diverse cooking techniques and develop a unique cooking style while building a portfolio of his culinary creations and receiving feedback.

1.3 Solution Proposal

Kitchen Guru introduces a unique dual-approach to address the challenges faced by modern home cooks. Our solution moves beyond traditional recipe platforms, innovatively addressing the paradox of abundant ingredients yet limited meal ideas.

We've crafted a user-friendly platform in Kitchen Guru, specifically designed to streamline recipe discovery. Here, users can effortlessly input their available ingredients, to which Kitchen Guru responds by generating a varied selection of recipes. This functionality not only encourages efficient use of pantry items but also supports a no-waste kitchen ethos, playing a crucial role in enhancing sustainability in daily cooking routines. The platform's effectiveness is further elevated by the introduction of AI-generated recipes, which adds an advanced layer of personalization and innovation to the recipe selection process.

Moreover, Kitchen Guru transcends being just a recipe platform; it's a community hub. It allows users to offer feedback, and participate in culinary discussions, fostering a shared culinary culture and community connection.

So, we propose the solution of KitchenGuru, a web application that seamlessly blends a smart recipe discovery tool with a vibrant cooking community. This platform not only simplifies the meal preparation process but also personalizes it through AI-driven suggestions, creating a unique and sustainable cooking experience for each user. KitchenGuru stands as a culinary companion and community platform, redefining home cooking in the digital age.

1.4 Comparative Analysis

The digital culinary space is characterized by a plethora of recipe websites, yet Kitchen Guru distinguishes itself by seamlessly blending advanced AI technology with a vibrant social fabric. This juxtaposition of technological prowess and community engagement sets Kitchen Guru apart in a crowded market.

While traditional recipe websites act as repositories of culinary data, Kitchen Guru transcends this static model by incorporating AI-driven functionalities that offer bespoke recipe recommendations. Beyond the AI capabilities, Kitchen Guru fosters a thriving social community where enthusiasts are not merely visitors but active participants. Unlike other platforms where interaction is limited, Kitchen Guru encourages its users to share their own culinary creations, exchange tips, and even modify existing recipes to suit their taste, creating a collaborative and dynamic culinary experience.

To illustrate the competitive edge of Kitchen Guru, consider the following comparative table, which juxtaposes Kitchen Guru with other leading recipe websites across various parameters.

Table 1.4.1 - Comparative Analysis of Recipe Websites

Feature / Website	**Kitchen Guru**	**Allrecipes**	**Food Network**	**Yummly**
Recipe Database	User-driven	Extensive	Extensive	Over 2 million+
User Profiles	Yes	Yes	-	Yes
AI-Driven Suggestions	Yes	No	No	No
Personalized Experience	High	No	No	No
Community Engagement	High	No	No	Account needed
Recipe Customization	User-driven	No	No	Account needed
Advanced Search Filters	Yes	Advanced	-	Many

In the comparative analysis of recipe websites, Kitchen Guru, Allrecipes, Food Network, and Yummly exhibit distinct features. Kitchen Guru uniquely offers a user-driven recipe database and AI-driven suggestions, highlighting its commitment to personalization. This contrasts with the extensive databases of Allrecipes and Food Network and the over 2 million recipes on Yummly. While user profiles are common across Kitchen Guru, Allrecipes, and Yummly, Food Network lacks this feature. Kitchen Guru excels in providing a highly personalized experience and high community engagement, a domain where Allrecipes and Food Network fall short. Yummly, although offering community engagement, requires account creation for full access. Both Kitchen Guru and Yummly enable recipe customization, a feature not present in Allrecipes and Food Network. Advanced search filters are a strength across all platforms, with Kitchen Guru and Allrecipes being particularly notable. Overall, Kitchen Guru's emphasis on AI-driven personalization and user engagement sets it apart, offering an innovative and tailored cooking experience compared to its competitors.

1.5 Domain Analysis Conclusions

In conclusion, the current chapter has provided a comprehensive domain analysis and customer validation for the Kitchen Guru project. Through our detailed domain analysis, we have identified key trends in culinary technology, consumer lifestyle shifts, and sustainability concerns, all of which Kitchen Guru addresses with its innovative features. The identified target groups, including busy professionals, sustainability advocates, and culinary enthusiasts, highlight the diverse range of users Kitchen Guru caters to. These groups are not only unified by their interest in cooking but also by their shared challenges in meal planning and desire for a sustainable, personalized cooking experience. Kitchen Guru's unique blend of AI-driven recipe suggestions, user-centric design, and community engagement positions it as a standout solution in the evolving culinary landscape, addressing the specific needs and preferences of these target groups effectively.

Part II

Internship

2 System Design

The system design phase is a meticulous process where we establish the software's architecture, components, and interfaces to align with our target users' requirements. This phase is segmented into various stages, each contributing to the overarching aim of conceptualizing and crafting the application's design. The outcomes of these individual stages are elaborated in the subsequent sections of this report.

2.1 Project Requirements

The next step for Kitchen Guru is to define and understand the system's requirements. The process of requirements analysis is pivotal for the success of any software product. These requirements detail the essential attributes and functionalities that the system must possess to fulfill the business objectives and stakeholders' expectations. By identifying and clearly specifying both functional and non-functional requirements, we can assure that Kitchen Guru not only meets but exceeds the expectations of its users, delivering a valuable and engaging experience. In this section, we will delve into both the functional and non-functional requirements, outlining their significance and how they contribute to the overall utility and performance of the application.

2.1.1 Functional Requirements

Functional requirements define the specific behaviors and functions of the Kitchen Guru application. These are the core features and operations that the software must perform to meet the needs of its users and achieve its intended purpose. Essential for guiding the development process, functional requirements ensure that the final product aligns with user expectations and project goals.

For Kitchen Guru, the functional requirements include:

Table 2.1.1 - Functional Requirements

Functional Requirement	Details
1. User Account Management	Secure registration, login, and profile management for users.
2. Ingredient Input Functionality	Allowing users to input ingredients they have on hand.
3. Recipe Suggestion Engine	System to suggest recipes based on user input and dietary preferences.
4. Community Interaction	Features for users to leave feedback and engage in discussions.
5. User Feedback Mechanism	Enabling users to rate recipes and provide reviews.
6. AI Recipe Generation System	System to generate recipes using AI, according to user input.
7. Data Synchronization	Ensuring user data (preferences, saved recipes) are consistent across different devices.

2.1.2 Non-Functional Requirements

Non-functional requirements are essential in defining the system's overall quality and operational characteristics. They do not describe specific behaviors of the application, like functional requirements, but instead focus on how the system should perform and be maintained. These requirements are crucial for ensuring user satisfaction, system reliability, and performance efficiency.

For Kitchen Guru, the non-functional requirements include:

Table 2.1.2 - Non-Functional Requirements

Non-Functional Requirement	Details
1. System Performance	Ensure fast response times and efficient processing of user requests.
2. Usability	User-friendly interface that is easy to navigate and intuitive for all user groups.
3. Security	Robust data protection measures, including secure user authentication and data encryption.
4. Scalability	Ability to accommodate an increasing number of users and data without performance degradation.
5. Compatibility	Cross-platform functionality for consistent user experience across various devices and operating systems.
6. Maintainability	Easy to update and maintain, with clear documentation and modular design for future enhancements.

2.2 System Modelling

In the design phase, it's crucial for the system's architecture to align with its requirements and functionalities. The architecture needs to be scalable, maintainable, and adhere to the system's quality attributes. UML (Unified Modeling Language) diagrams are pivotal in this process. They offer diverse perspectives and visual representations of the system's components and their interactions, aiding in identifying design flaws and ensuring consistency in the architecture. PlantUML, a versatile open-source tool, is instrumental in creating these UML diagrams. It supports various diagram types like use case, sequence, activity, and class diagrams. Utilizing PlantUML enhances the clarity and precision of our system documentation, simplifying the understanding of the system's design and implementation.

2.2.1 Use Case Diagrams

Use case diagrams are essential for understanding the high-level functionality of a system from a user's perspective. They depict the different interactions between actors (users or external systems) and the system itself. Use case diagrams help in defining and documenting the system's requirements, illustrating the various scenarios or use cases that the system can handle [3].

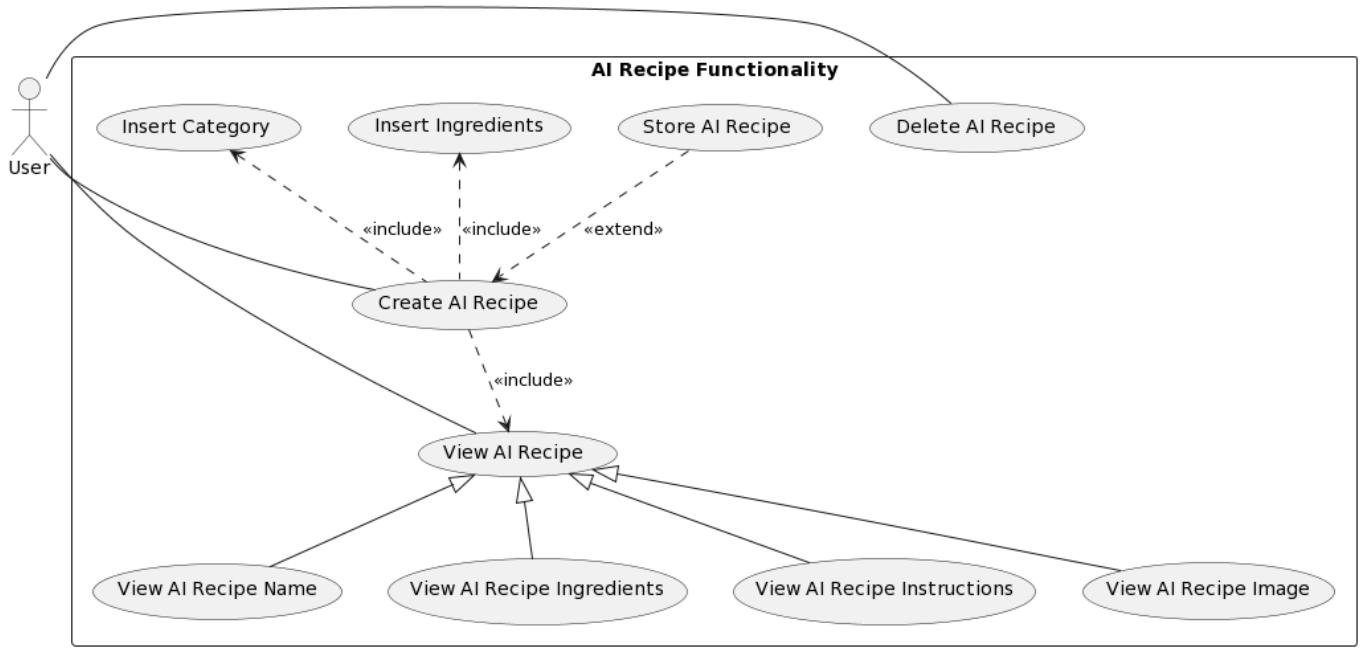

Figure 2.2.1 - Use Case Diagram for AI Recipe

Figure 2.2.1 illustrates the "AI Recipe Functionality" within the Kitchen Guru application. The user has the capability to perform several actions related to AI-generated recipes. The "Insert Category" and "Insert Ingredients" use cases are preliminary steps that feed into the central "Create AI Recipe" use case, indicating that these actions are essential components of recipe creation. Additionally, the user can "Store AI Recipe" for later use, or choose to "Delete AI Recipe" if it's no longer needed. Viewing a recipe is

detailed with options to see the recipe's name, ingredients, instructions, and image. Each of these use cases is directly accessible by the user, showcasing the user's interactive role in managing AI-generated recipes within the application.

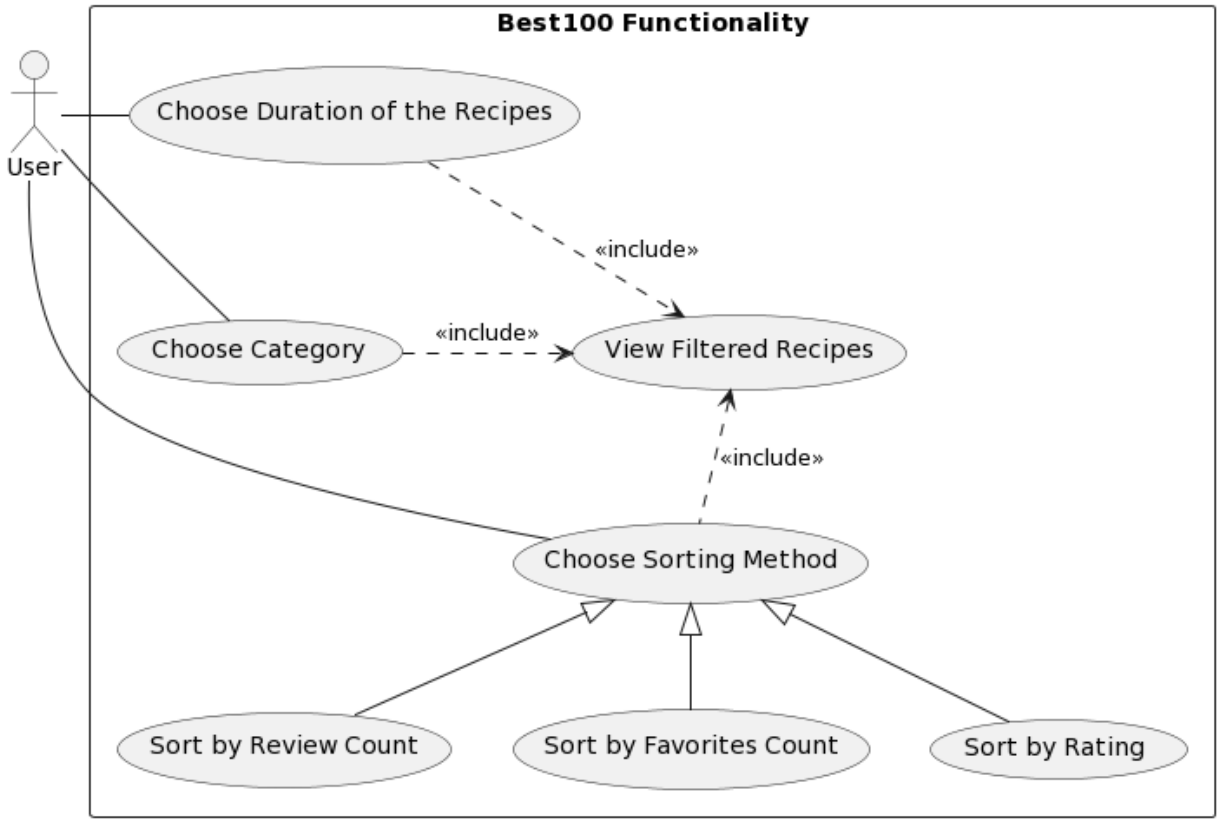

Figure 2.2.2 - Use Case Diagram for Best 100 page

Figure 2.2.2 depicts the "Best 100 Functionality" use case diagram for the Kitchen Guru application. The user is presented with multiple options to customize their view of the top recipes. Initially, the user can choose the duration of the recipes they are interested in, which is a filtering action that narrows down the list of recipes based on the time required to prepare them. In addition to duration, the user can also filter recipes by category. Both the duration and category choices are included in the "View Filtered Recipes" use case, indicating that these filters can be applied in combination to refine the recipe list further. After filtering, the user can choose how they would like the recipes to be sorted. The diagram offers three sorting methods: by review count, by favorites count, and by rating. These sorting options provide the user with the flexibility to view the recipes according to their preferences, whether they are looking for the most popular, the most liked, or the highest-rated recipes.

Figure 2.2.3 presents the "Recipe Interaction System" use case diagram within the Kitchen Guru application. This diagram illustrates the various interactions a user can have with a recipe. The central action is "View Recipe Details," which encompasses viewing the recipe's image, instructions, ingredients, and name. Adjacent to this, users can "View Recipe Reviews" to see what others have thought about the

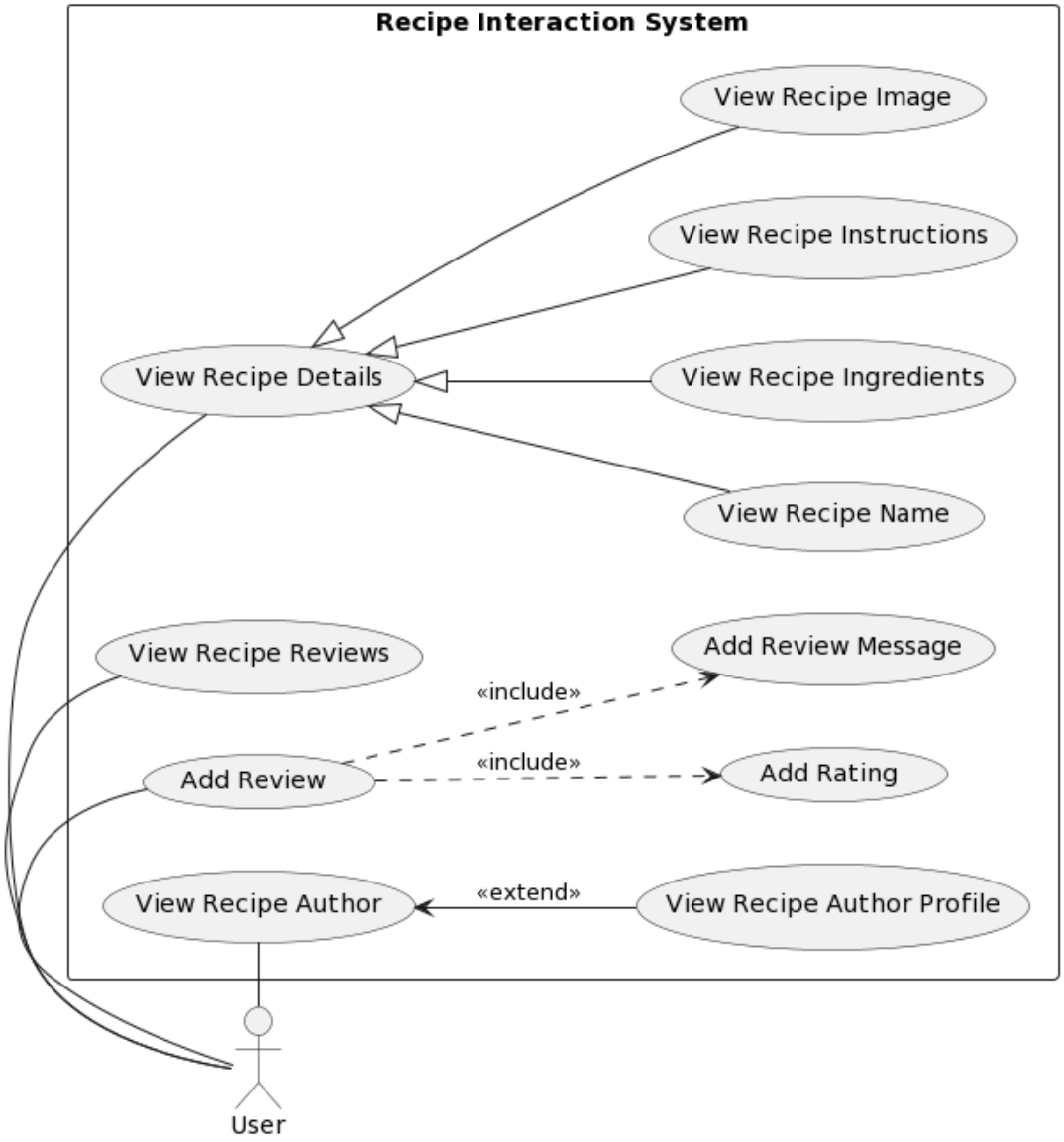

Figure 2.2.3 - Use Case Diagram for Recipe Interaction System

dish. If they wish to contribute, they can ”Add Review,” which includes writing a review message and giving a rating. Additionally, users can view the ”Recipe Author,” and if they want to learn more about the creator of the recipe, they can extend this action to ”View Recipe Author Profile.”

2.2.2 Activity Diagrams

Activity diagrams are used to model the workflow or business processes within a system. They are particularly valuable for capturing the sequential and parallel activities that occur in a system, making them vital for both analysis and design phases. Activity diagrams can help in identifying bottlenecks, optimizing processes, and improving the overall system efficiency [4].

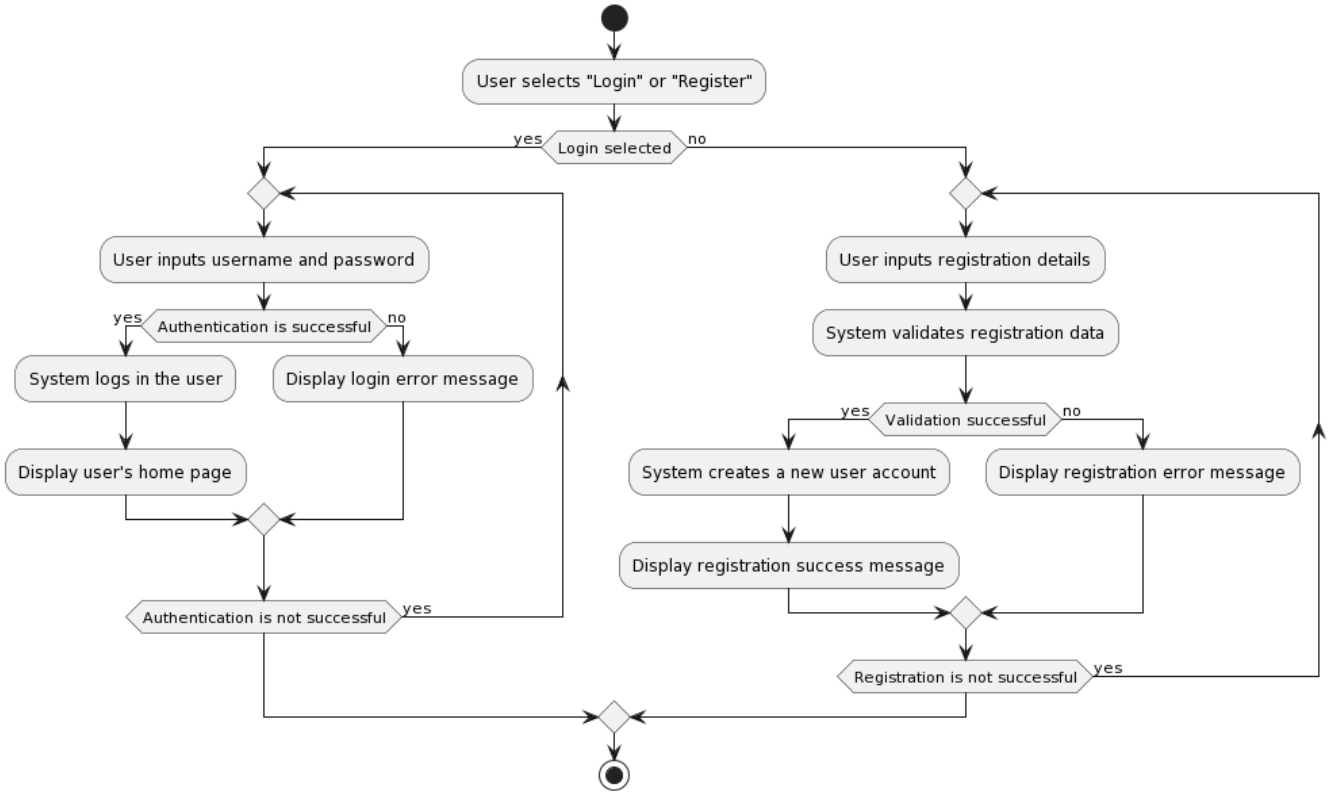

Figure 2.2.4 - Activity Diagram for Login/Registration

The activity diagram in Figure 2.2.4 illustrates the user flow for login and registration for Kitchen Guru. The process begins with the user choosing to either ”Login” or ”Register.” If ”Login” is selected, the user inputs their username and password. The system then attempts to authenticate the user. If authentication is successful, the user is logged in and directed to their home page. If not, a login error message is displayed. On the other hand, if the user selects ”Register,” they are prompted to input registration details. The system validates this data, and if the validation is successful, it creates a new user account and displays a success message. If the validation fails, a registration error message is displayed.

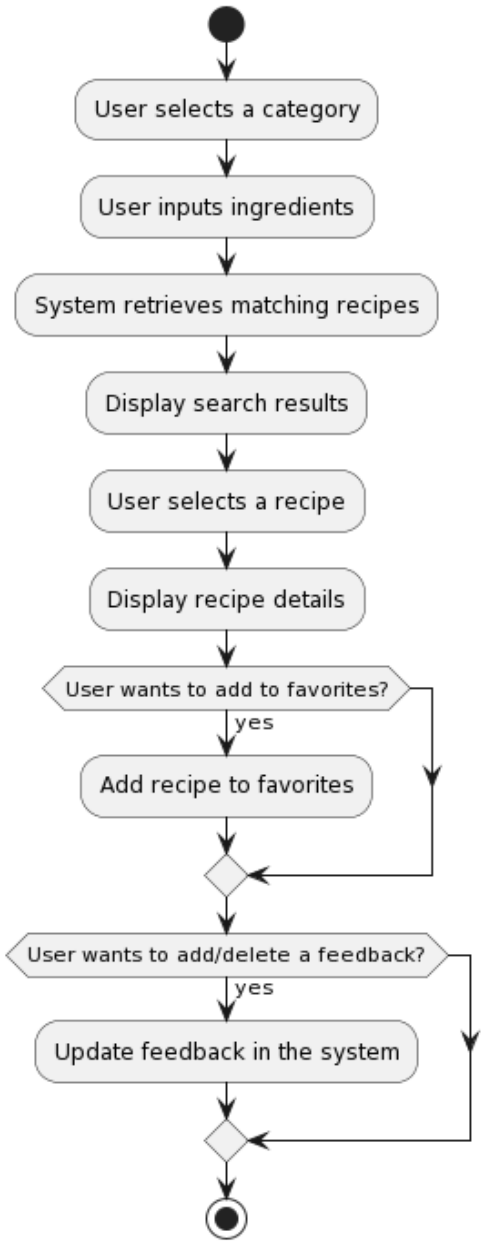

Figure 2.2.5 - Activity Diagram for Recipe Search

The activity diagram in Figure 2.2.5 details the process of searching for a recipe within Kitchen Guru. The user begins by selecting a recipe category, then inputs the ingredients they have available. With this information, the system retrieves matching recipes and displays the search results. Once the results are displayed, the user can select a recipe to view more details. At this point, the user has the option to add the recipe to their favorites—if they choose to do so, the recipe is added accordingly. Additionally, the user has the option to add or delete feedback on a recipe. If they opt to update feedback, the system processes this update.

The activity diagram in Figure 2.2.6 outlines the profile management workflow within the Kitchen Guru application. A user starts by viewing their profile page. From there, they have the option to edit their profile information. If they choose to do so, they are presented with a series of decisions: they can edit their username, password, and email address, with each option leading to the respective edit action if selected. Once the user has completed or bypassed the editing process, they can proceed to view their posted recipes. This activity is followed by viewing their favorite recipes and, finally, their recently viewed recipes. Each step in this process is user-initiated, offering a clear and structured path through the various facets of profile management. The diagram concludes with the user completing their interactions with the profile, indicated by the final end state symbol.

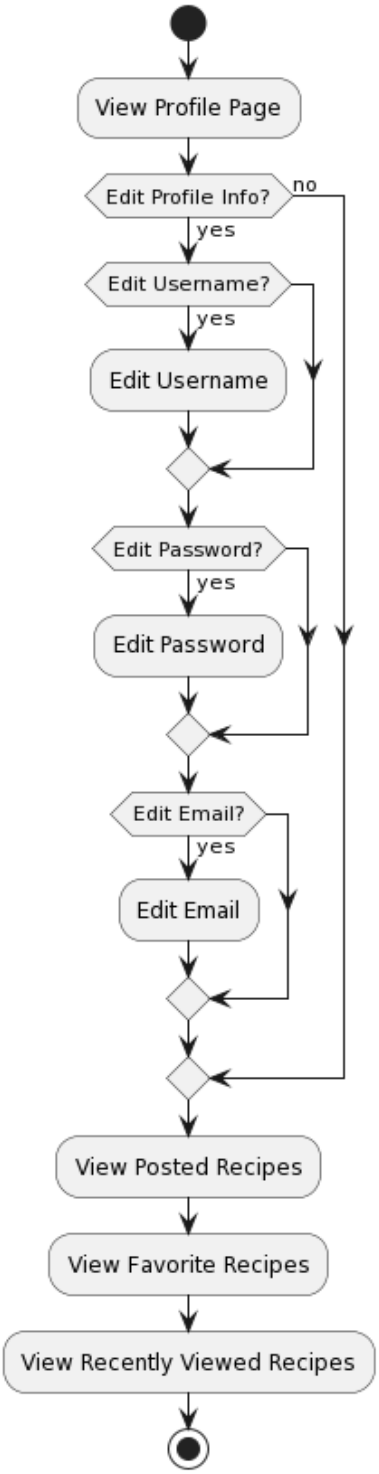

Figure 2.2.6 - Activity Diagram for Profile

2.2.3 Sequence Diagrams

Sequence diagrams focus on illustrating the interactions and relationships between different objects or components of a system over time. These diagrams are essential for modeling the dynamic behavior of a system, showing the sequence of messages exchanged among objects during the execution of a specific use case or scenario. Sequence diagrams aid in understanding how different system elements collaborate and communicate [5].

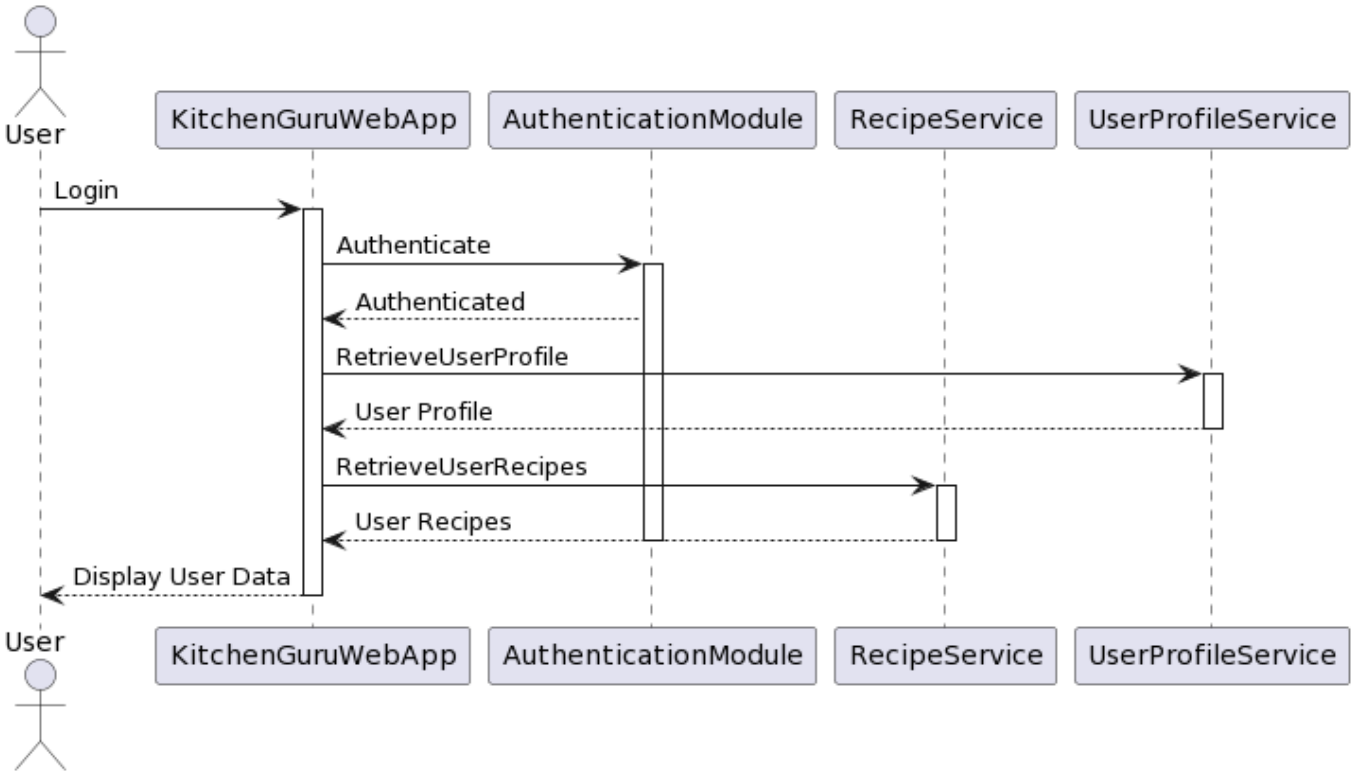

Figure 2.2.7 - Sequence Diagram for Login and Profile

The sequence diagram in Figure 2.2.7 outlines the login and profile retrieval process for a user on the Kitchen Guru application. The interaction begins with the user sending a login request to the KitchenGuruWebApp. The web app then communicates with the AuthenticationModule to authenticate the user. Upon successful authentication, indicated by the ”Authenticated” message, the web app proceeds to request the user's profile data from the UserProfileService. Once the UserProfileService returns the user's profile information to the web app, the web app makes another request to the RecipeService to retrieve the user's saved recipes. After the RecipeService provides the list of user recipes, the web app compiles the user's data, which includes both profile information and recipes. Finally, this compiled data is displayed to the user, completing the sequence.

Figure 2.2.8 showcases a sequence diagram that captures the recipe search process within the Kitchen Guru app. The interaction is initiated by the user, who inputs specific ingredients—such as

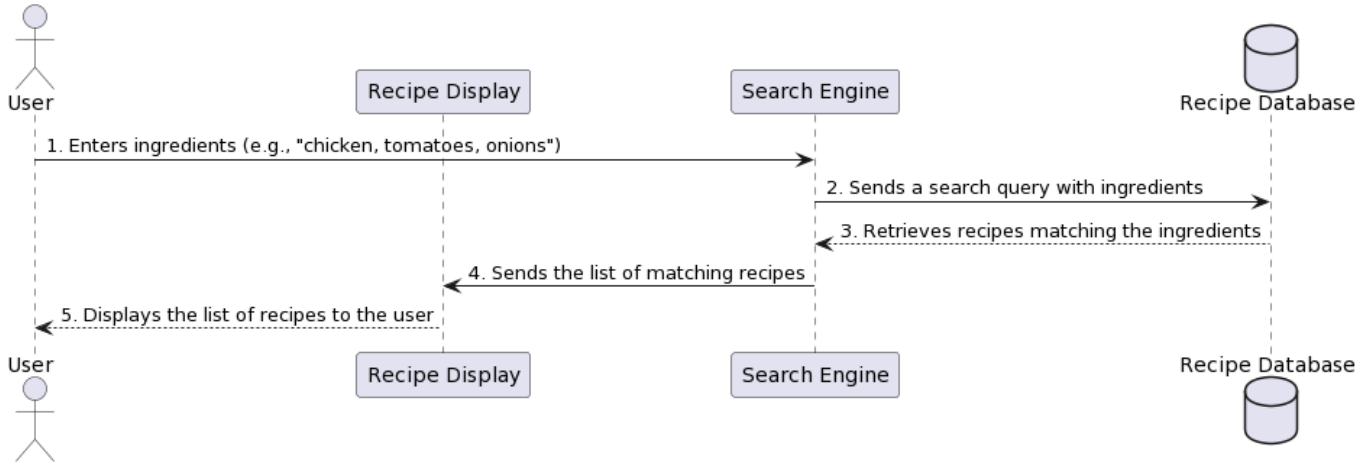

Figure 2.2.8 - Sequence Diagram for Recipe Search

”chicken, tomatoes, onions”—into the application’s recipe display interface. This input triggers a sequence of events starting with the recipe display component, which forwards a search query containing the user’s ingredients to the search engine component. Subsequently, the search engine queries the recipe database, retrieving a list of recipes that match the provided ingredients. Once the relevant recipes are fetched from the database, the search engine sends this list back to the recipe display. The user then sees the final output where the recipe display presents a curated list of recipes that utilize the entered ingredients.

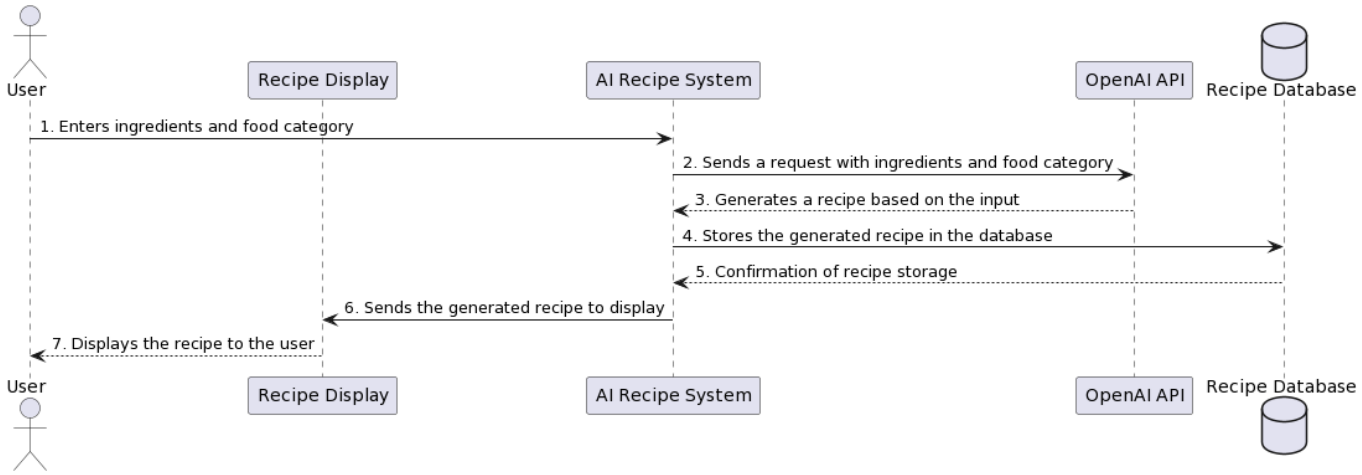

Figure 2.2.9 - Sequence Diagram for AI Recipe

Figure 2.2.9 is a sequence diagram that outlines the flow of operations for generating a recipe using an AI system within the Kitchen Guru app. The sequence initiates when the user enters ingredients and selects a food category via the Recipe Display. This action triggers the Recipe Display to send a request with the ingredients and food category to the AI Recipe System. Upon receiving the input, the AI Recipe System interacts with the OpenAI API to generate a recipe based on the user’s input. After the recipe is generated, the AI Recipe System instructs to store the new recipe in the Recipe Database. The database acknowledges the successful storage of the recipe and sends a confirmation back to the AI Recipe System.

Following this, the AI Recipe System sends the generated recipe back to the Recipe Display, which then presents the recipe to the user.

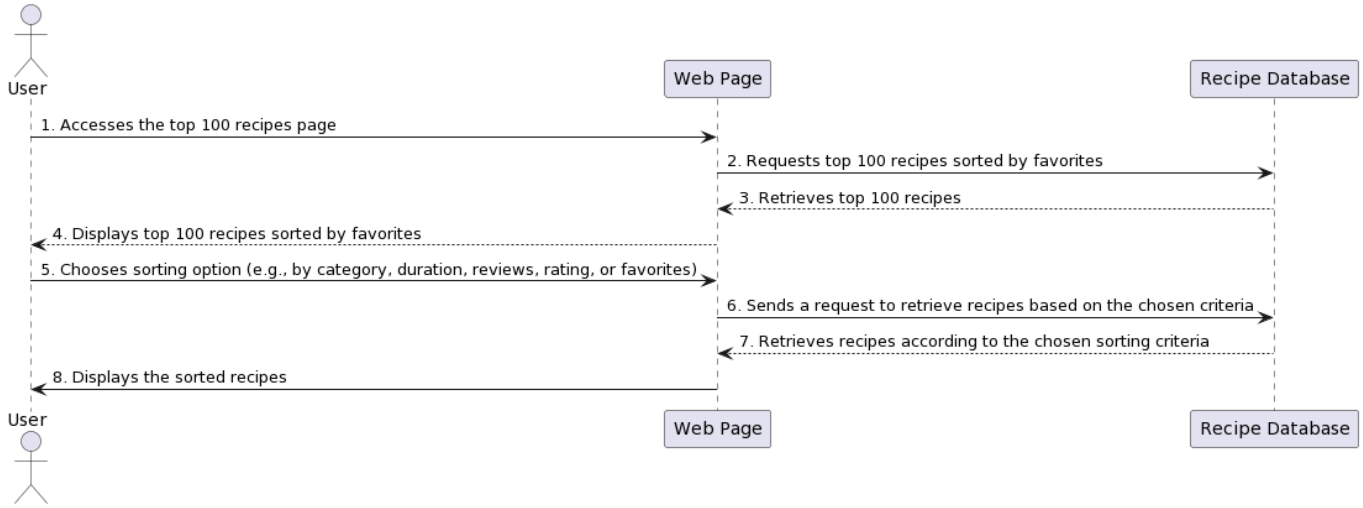

Figure 2.2.10 - Sequence Diagram for Top 100 Page

Figure 2.2.10 presents a sequence diagram that delineates the user interaction and system response for the Top 100 Recipes page in the Kitchen Guru app. The sequence commences with the user accessing the top 100 recipes page. This action prompts the Web Page component to send a request to the Recipe Database to retrieve the top 100 recipes sorted by favorites. Upon the retrieval of these recipes, the Web Page then displays the top 100 recipes to the user, sorted by the number of favorites. The user, given the choice to sort the recipes, selects a sorting option, which could be by category, duration, reviews, rating, or favorites. The Web Page responds to this selection by sending another request to the Recipe Database to retrieve recipes based on the chosen sorting criteria. The Recipe Database executes this request and retrieves the recipes according to the specified sorting criteria. These sorted recipes are then sent back to the Web Page, which subsequently displays the sorted recipes to the user.

2.2.4 Deployment Diagrams

Deployment diagrams provide a comprehensive view of the physical infrastructure of a system, including hardware components, software components, and their connections. They are crucial for planning the deployment of a system, showing the distribution of software components across different hardware nodes, networks, and servers. Deployment diagrams help in optimizing resource allocation, scalability, and fault tolerance [6].

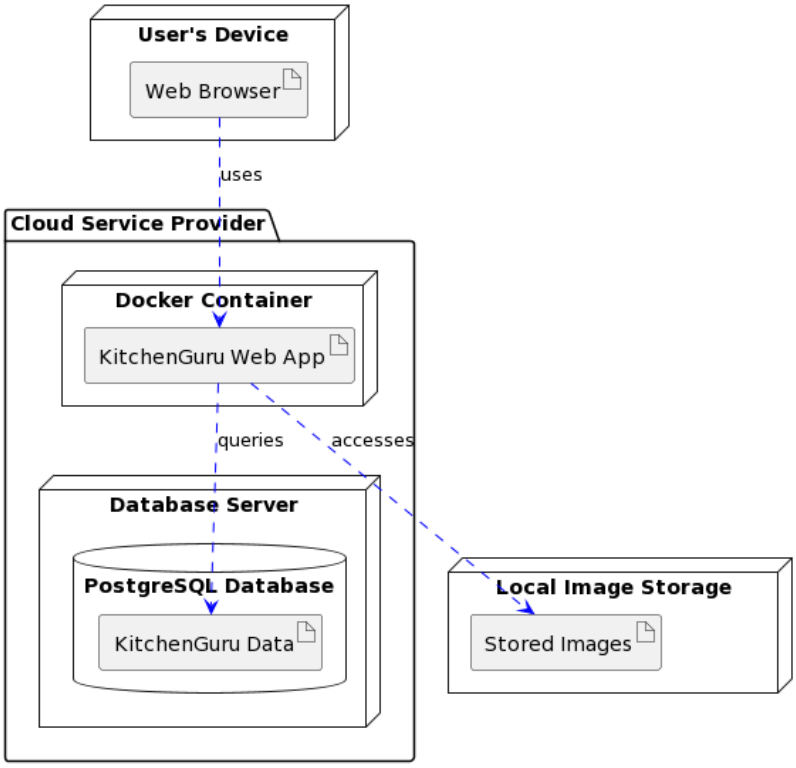

Figure 2.2.11 - Deployment diagram

Figure 2.2.11 depicts the deployment diagram for the Kitchen Guru application, illustrating the architecture and the environment in which the application operates. A user interacts with the application through a web browser on their device, which is the entry point for using Kitchen Guru's services. The web application itself is containerized using Docker, which encapsulates the Kitchen Guru Web App, ensuring that the environment is consistent and isolated from other applications. The Docker container resides within a cloud service provider's infrastructure, highlighting the app's cloud-based deployment model which allows for scalability and accessibility over the internet. The PostgreSQL Database is set up on a separate database server, indicating a clear separation between the application logic and the data storage. This database stores all the data related to Kitchen Guru, including user information, recipes, and more. Additionally, the diagram shows Local Image Storage, which is where all the images used by Kitchen Guru are stored. This could be on the same server as the database or on a different one, but the key point is that the storage is local, not cloud-based.

2.2.5 Component Diagrams

Component diagrams focus on the organization and dependencies among software components within a system. They illustrate how different modules or components interact and depend on each other. Component diagrams are vital for system architects and developers to understand the system's structure, promote modularity, and facilitate component reuse, ultimately leading to more maintainable and scalable software [7].

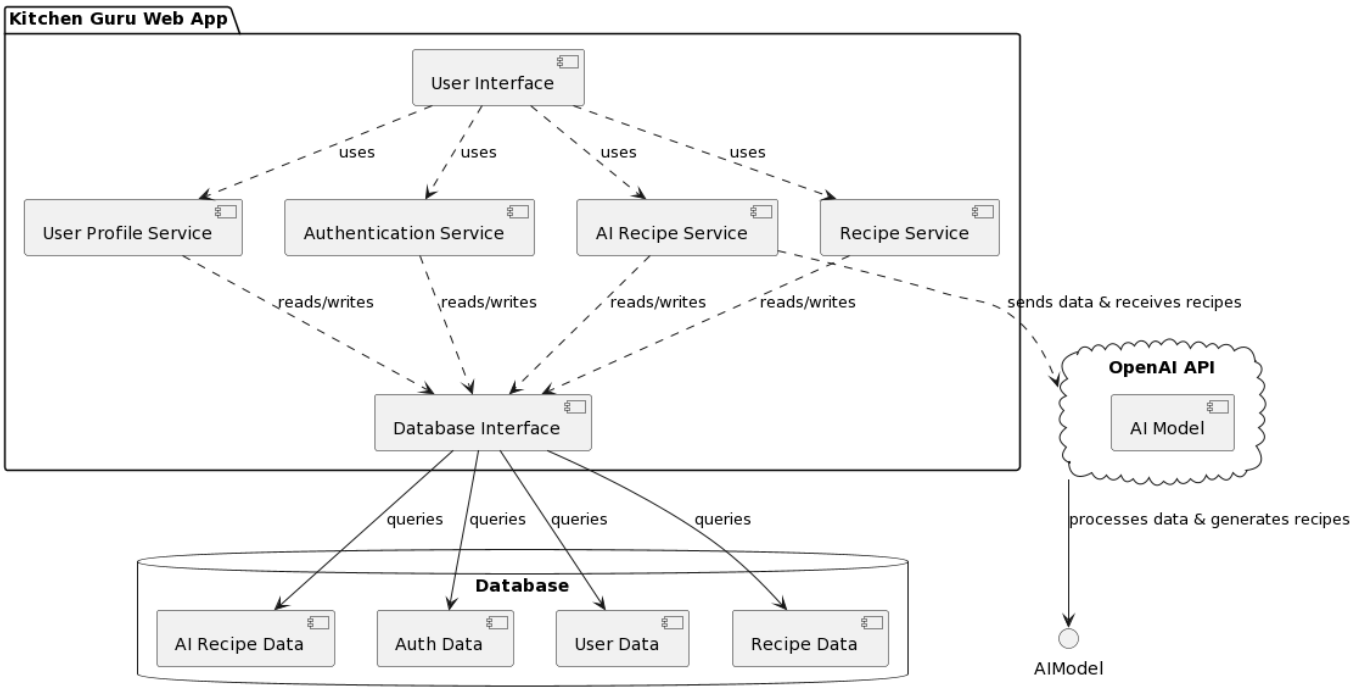

Figure 2.2.12 - Component diagram

Figure 2.2.12 is the component diagram for the Kitchen Guru Web App, showcasing the architecture and the modular design of the system. The central component is the ”User Interface” through which users interact with various services. It connects to the ”User Profile Service” for managing user profiles, the ”Authentication Service” for handling sign-in and security, the ”AI Recipe Service” for generating new recipes using AI, and the ”Recipe Service” for standard recipe management. The ”AI Recipe Service” is particularly crucial as it interfaces with the ”OpenAI API,” sending data to and receiving generated recipes from an external ”AI Model.” This demonstrates the app's innovative use of AI to enhance the user experience by providing personalized recipe suggestions. All these services interact with the ”Database Interface,” which abstracts the communication with the underlying ”Database.” This database stores ”AI Recipe Data,” ”Auth Data,” ”User Data,” and ”Recipe Data,” ensuring that all aspects of the application's data are organized and accessible.

2.3 Customer Validation & Enhancements

Prior to customer validation, we hypothesized that our website's main features—AI-generated recipes, ingredient-based search, and community engagement—would resonate strongly with our target audience. Validating our project is crucial, as it ensures our service aligns with customer needs and market demand. We selected a range of questions aimed at gauging user interest in these features, their daily cooking challenges, and their openness to community interaction.

Table 2.3.1 - List of Questions from Customer Validation Form

No.	Question
1	On a scale from 1 to 10, how serious do you consider the issue of daily indecision about what to cook and the problem of culinary waste?
2	On a scale from 1 to 10, how would you rate the effectiveness of our solutions in addressing the problem of daily cooking indecision and culinary waste?
3	How valuable do you think the ability to search for recipes by ingredients is?
4	Would you be interested in trying recipes created by other users on a cooking website?
5	How do you feel about the idea of AI-generated recipes?
6	How important is a comment section or a community aspect for you on a cooking website?
7	Based on what you know about Kitchen Guru, would you use our website regularly?
8	Compared to other recipe websites you know or use, how does Kitchen Guru stand out? What would make our website more appealing to you than others?

In total we gathered 52 responses on our form and the results were predominantly positive, confirming that our initial features hit the mark with our prospective users. Notably, the ability to search recipes by ingredients was highly valued, and there was a strong interest in trying recipes from other users, underscoring the appeal of our community-centric approach. The idea of AI-generated recipes was met with enthusiasm, suggesting that users are looking for innovative ways to enhance their cooking experience.

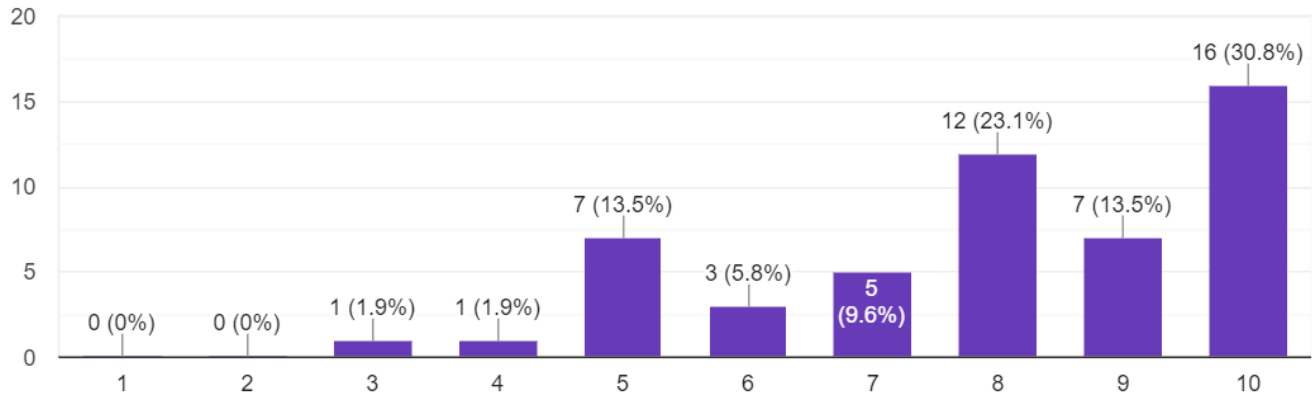

Figure 2.3.1 - Responses to "How serious is the issue of daily indecision about what to cook?"

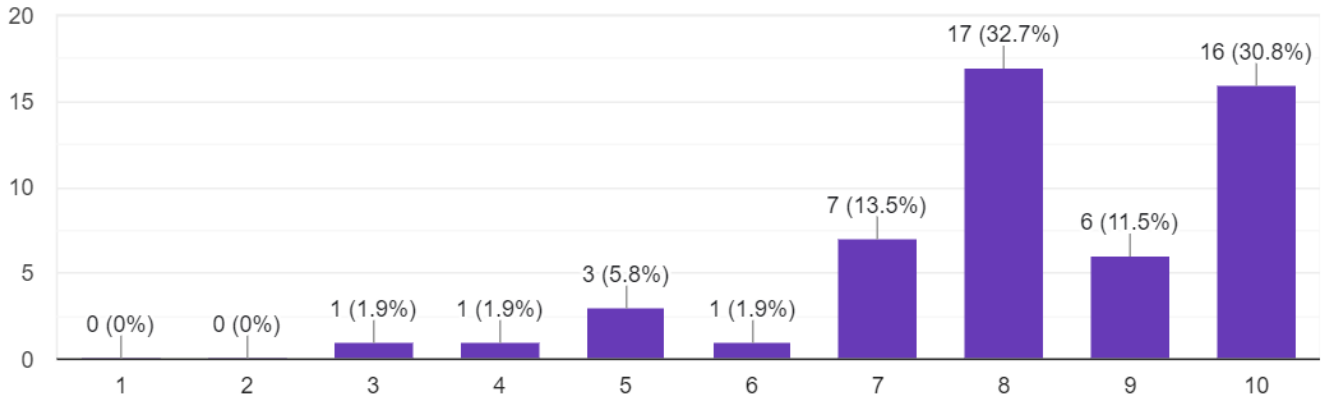

Figure 2.3.2 - Responses to ”How do you feel about AI-generated recipes?”

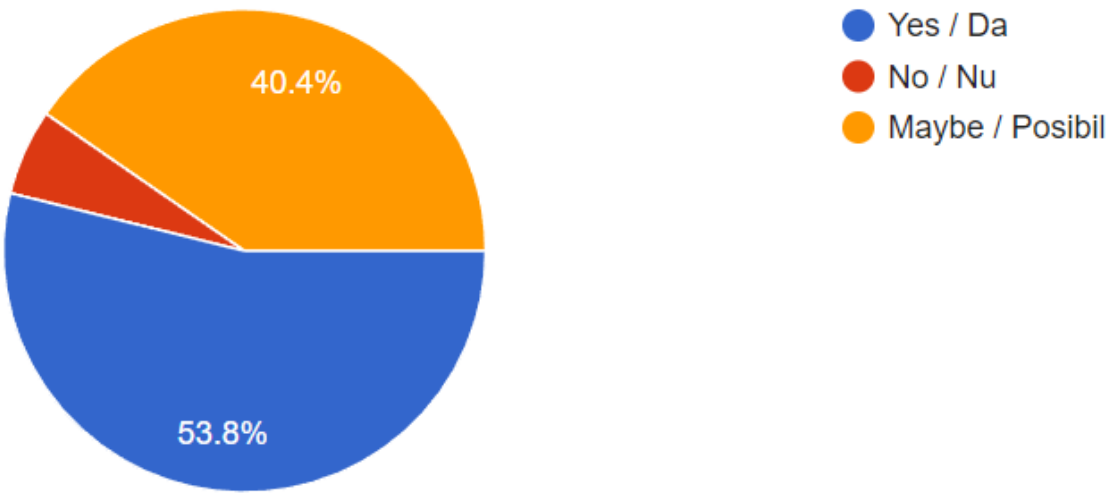

Figure 2.3.3 - Responses to ”Would you use our application regularly?”

However, some feedback pointed towards additional features that could enhance the user experience, such as a more robust social platform with friend lists and chat functions, as well as a desire for more detailed nutritional information.

In conclusion, while our core hypotheses were validated, new ideas emerged that could shape the future enhancements of Kitchen Guru, ensuring it remains a cutting-edge culinary platform that meets and exceeds user expectations.

3 Implementation Details

3.1 Version Control

Version control is a system that tracks the history of changes as a project evolves, allowing for efficient collaboration among team members. Version control systems (VCS) are fundamental to software development, providing a log of who made changes, what those changes were, when they were made, and why they were needed. This historical record is crucial for understanding the evolution of a project and for enabling developers to work independently while still remaining in sync with the team's progress [8].

Version control is especially important as it helps prevent conflicts between concurrent updates, tracks changes to prevent errors, and maintains the integrity of the codebase over time [9]. Git, a distributed version control system (DVCS), has become the most popular VCS today, as it allows every developer to have a full copy of the project and its history, enabling work even without a constant connection to a central repository.

We chose GitHub for its robust ecosystem and its ability to host Git repositories while providing an array of tools for code management and collaboration. GitHub facilitates better code through features like issues for threaded discussions, pull requests for change proposals, code reviews for quality assurance, and the GitHub Marketplace for apps that extend functionality. It supports the GitHub flow, an established workflow that streamlines the development process, helping developers work on updates, commit changes, open pull requests, and merge approved changes efficiently. With over 100 million developers and a wealth of integrations, GitHub is a platform that has significantly impacted how software is built [10].

For the Kitchen Guru project, the decision to create two separate repositories—one for the backend and one for the frontend—was made to maintain a clear separation of concerns. This approach allows different teams to work on each part of the project with minimal interference, making the development process more organized and manageable. It also simplifies tracking changes, managing dependencies, and testing, as each part of the application can be developed and deployed independently. The division into two repositories is reflective of best practices in modern software development, where modularity and clarity are key to successful project management.

3.2 Project Development

The design phase began with Figma, a versatile design tool that facilitates collaborative design efforts. Figma's web-based platform allows multiple team members to work concurrently across different operating systems, making it a perfect choice for remote and distributed teams. It's not just a tool for static designs; it supports prototyping, animation, and has an extensive array of plugins, which made it an ideal choice for creating the early design of Kitchen Guru's user interface.

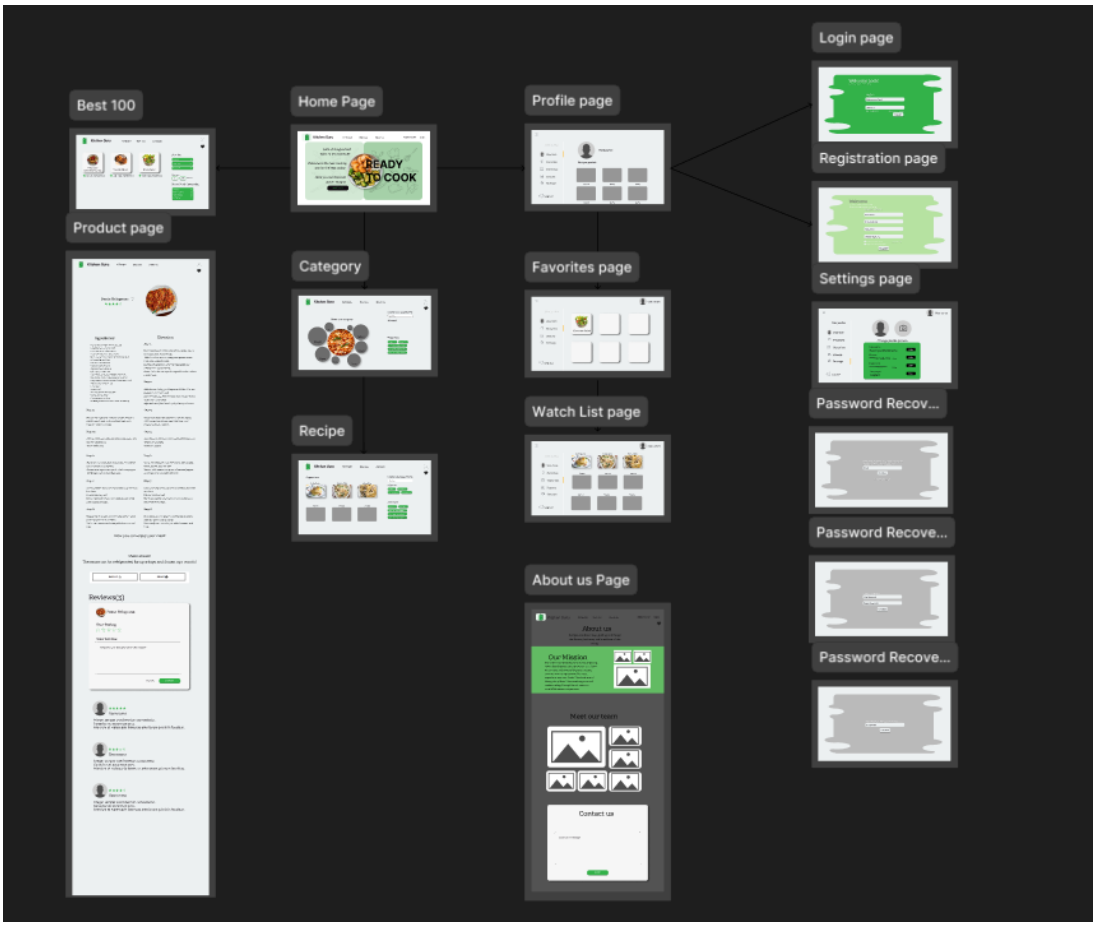

Figure 3.2.1 - Kitchen Guru mockup in Figma

In the initial stages of Kitchen Guru's development, it was imperative to establish a foundational database to support the core functionality of recipe discovery. To this end, we meticulously curated a starting collection of approximately 200 recipes, thoughtfully categorized to cater to a diverse range of culinary preferences. This initial compendium featured a variety of categories including fast-food, soups, salads, desserts, meat dishes, and seafood, ensuring a rich and versatile selection. Manually sourcing these recipes allowed us to populate our database with quality content, setting a high standard for the user experience from the very beginning. This foundational database not only served as a launchpad for the platform but also provided a valuable testing ground for refining search algorithms and user interface interactions.

For the backend technologies, we selected Python Django because of its "batteries-included" phi-

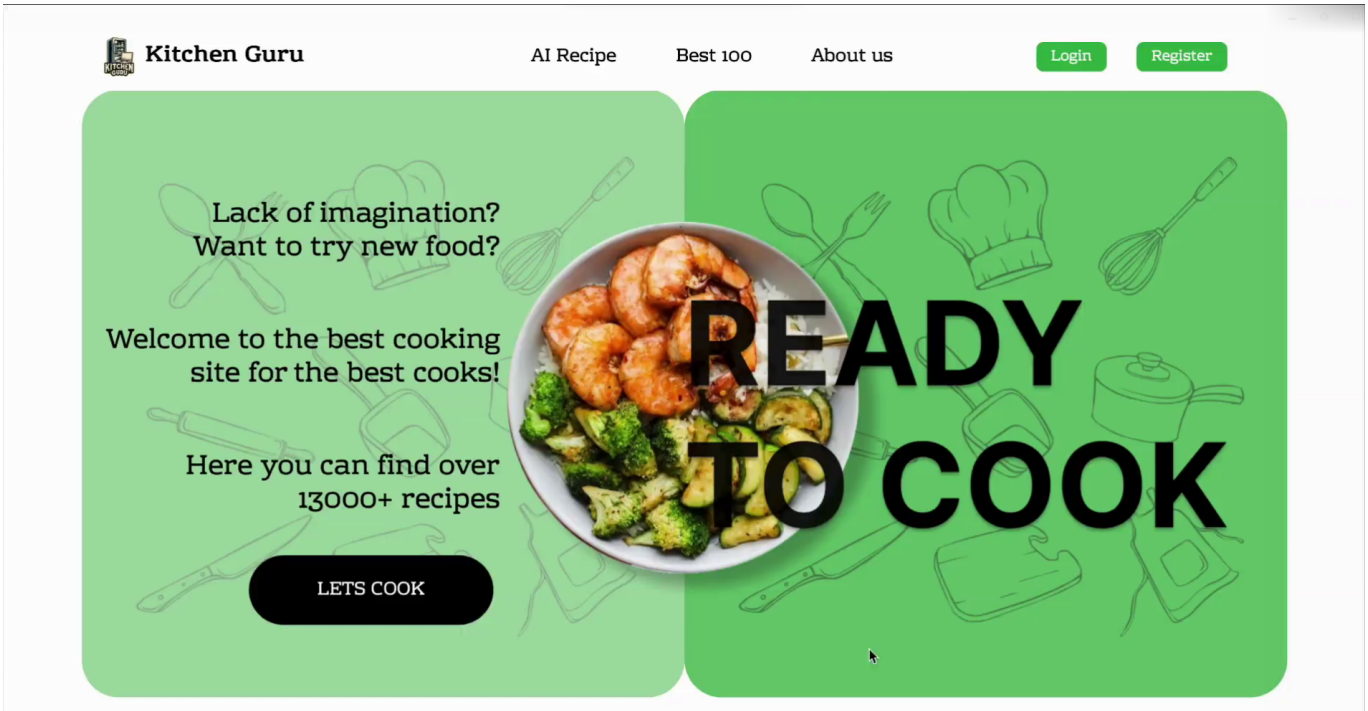

Figure 3.2.2 - Kitchen Guru Home Page

losophy. Django is renowned for its pre-built components, which accelerate the development process by handling common backend tasks. This high-level framework allows us to focus on writing unique business logic rather than reinventing the wheel for each new feature. Its scalability, robust performance, and flexibility align perfectly with the needs of a responsive web application like Kitchen Guru. For the frontend, we chose React with TypeScript, creating a robust and maintainable codebase. React's component-based architecture, coupled with TypeScript's static typing, ensures a more reliable and predictable UI development, which is particularly beneficial for complex applications.

Prior to diving into development, our team dedicated time to mastering the chosen technologies. This ensured that every team member could contribute effectively, understand the nuances of the tools, and apply best practices right from the start. It's noteworthy that our development team was thoughtfully segmented into two distinct groups to optimize our efforts: a trio of developers dedicated themselves to the backend intricacies, while a pair focused on the frontend experience. This strategic allocation of resources ensured that each aspect of Kitchen Guru was crafted with dedicated expertise and attention.

With a solid grasp of our tools, we moved on to the actual development of Kitchen Guru. Adopting an agile development methodology, we worked collaboratively, with GitHub serving as our central hub for version control and code collaboration. The decision to create separate repositories for the backend and frontend streamlined our workflow, allowing parallel development without conflicts. Throughout this phase, we've maintained a strong focus on quality, scalability, and user experience, ensuring that every feature we introduce brings us closer to realizing the vision of Kitchen Guru.

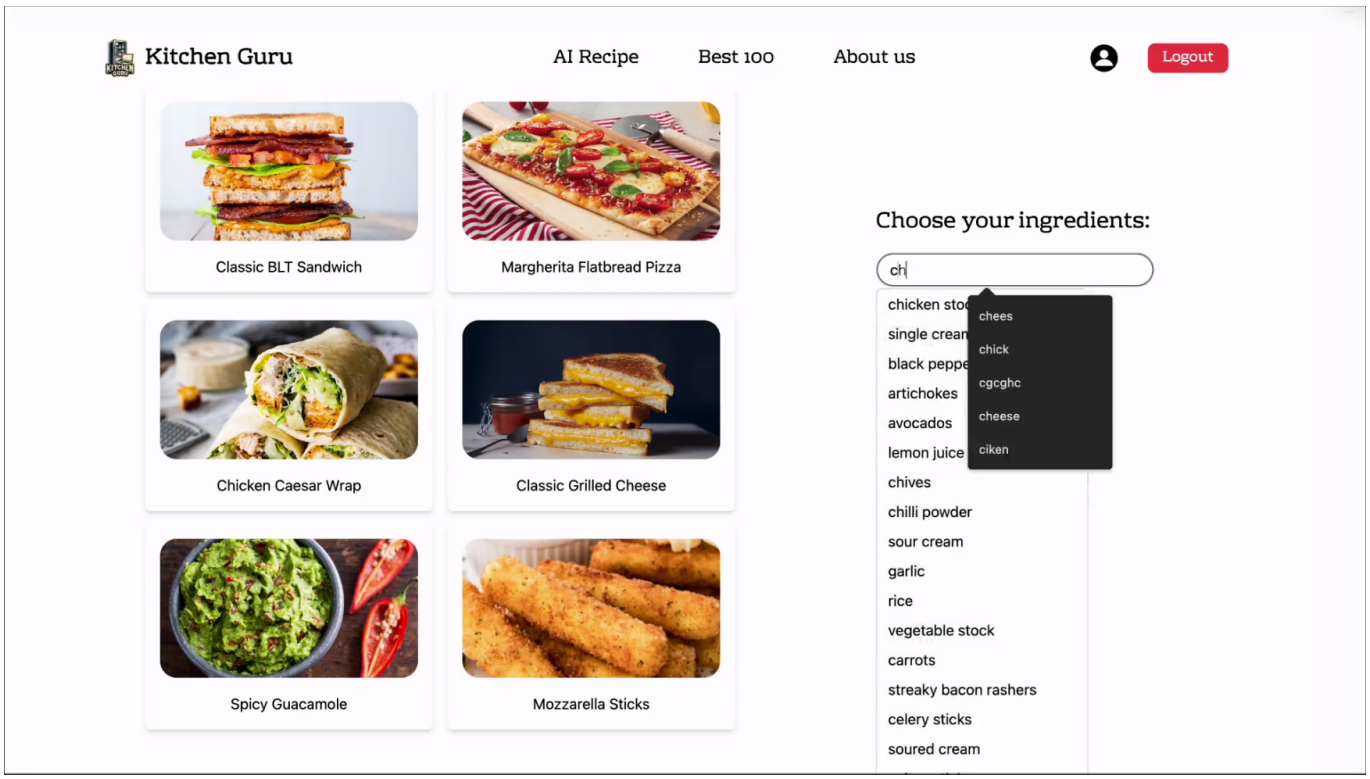

Figure 3.2.3 - Kitchen Guru Home Page

Docker emerged as an integral component during the development of Kitchen Guru due to its powerful containerization capabilities. As an open platform for developing, shipping, and running applications, Docker enables developers to package applications into containers—standardized executable components combining application source code with the operating system (OS) libraries and dependencies required to run that code in any environment. This ensures consistency across multiple development and release cycles, standardizing the environment in which the app runs, and minimizing discrepancies between development, staging, and production.

Our strategy involved creating three distinct Docker containers to encapsulate the different components of our application: the backend, the frontend, and the database. This trisection allowed us to modularize the application, streamline the development process, and enhance scalability.

The backend container hosts the Django application, encapsulating all the server-side logic, API endpoints, and interaction with the database. Isolating this layer in its own container ensures that we can scale the backend services independently based on the load, which is especially beneficial during traffic surges.

The frontend container serves the React application, delivering the user interface and static assets. This separation allows frontend developers to iterate on the user experience and interface without impacting the backend logic, facilitating a clear separation of concerns and enabling continuous integration and delivery.

The database container runs PostgreSQL, ensuring that our data storage is both robust and flexible. Containerizing the database simplifies local development and testing, as developers can work with a database environment identical to production, minimizing deployment-time surprises.

By containerizing these components, we not only simplified the development and deployment processes but also created a scalable and maintainable infrastructure. This approach allowed each part of Kitchen Guru to operate independently, providing the flexibility to update and scale each component according to specific needs without affecting the others.

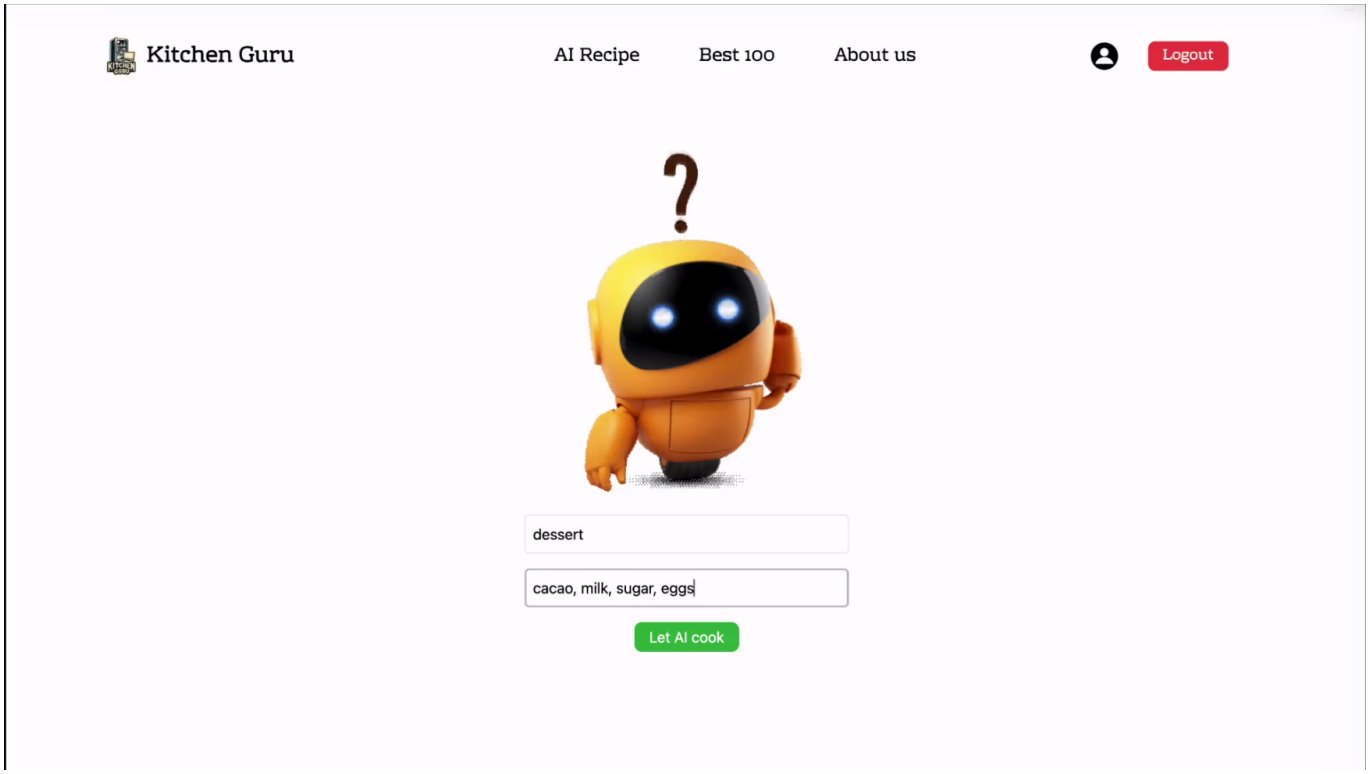

Figure 3.2.4 - Kitchen Guru Home Page

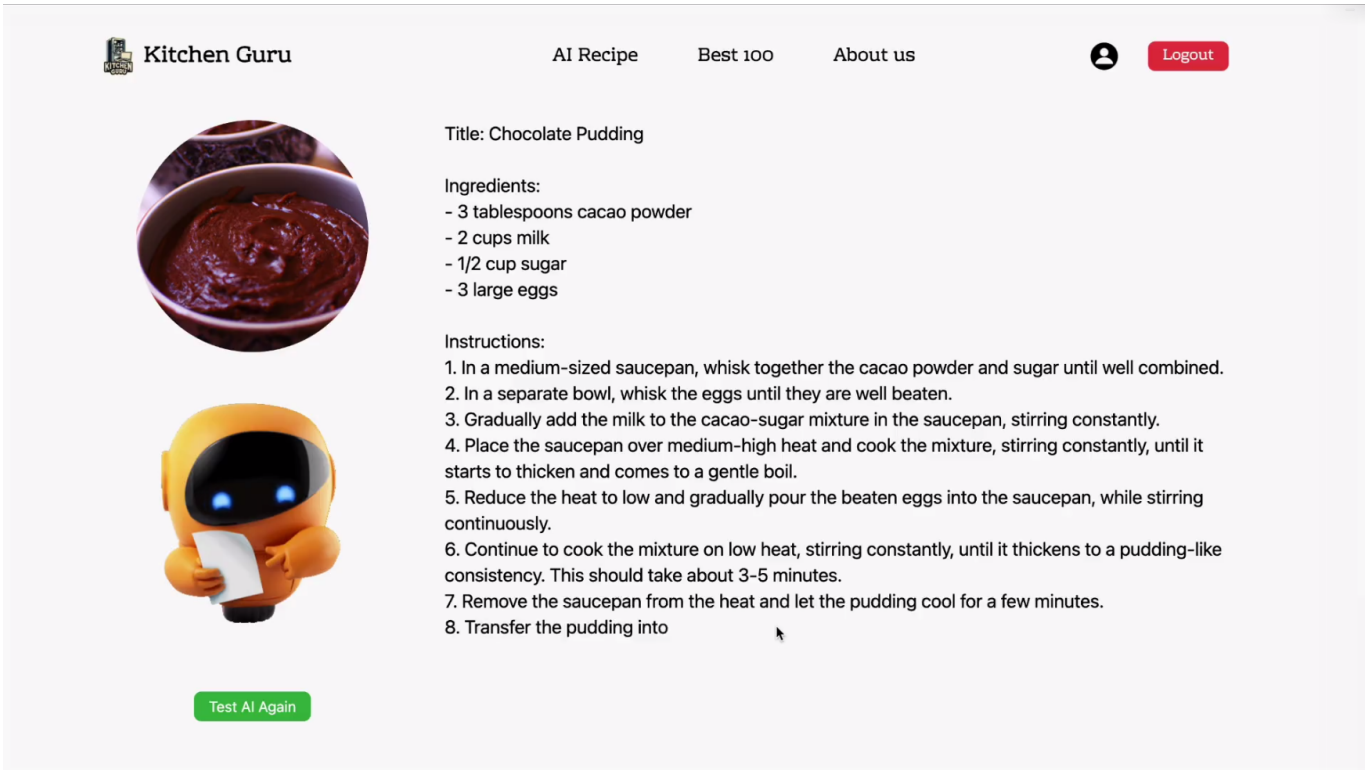

Figure 3.2.5 - Kitchen Guru Home Page

Figure 3.2.6 - Kitchen Guru Home Page

3.3 Application Deployment

A deployment service is an essential component of the software development lifecycle, serving as the mechanism through which software is delivered to users. It encompasses all the necessary activities and services to render a software system operational and accessible. This includes managing the hosting environment, directing traffic, ensuring security measures are in place, and dynamically scaling the application in response to user demand.

For the Kitchen Guru project, we pivoted to using Render for our deployment requirements. Render is an emerging cloud platform that provides a simpler, more user-friendly approach to deployment, offering a range of services that cater to modern application needs. Known for its ease of use and straightforward setup, Render is an excellent choice for teams looking to deploy applications without the complexity often associated with larger cloud platforms. Render also provides key features like instant deploys, automatic HTTPS, private networking, and a range of add-ons for databases and other services. This ensures our application can scale effortlessly and maintain high availability. Additionally, Render offers capabilities for continuous integration and deployment, which aligns perfectly with our agile development process. With Render, we anticipated a more intuitive deployment process, flexible scalability options, and a reliable platform that would support a smooth and efficient user experience for Kitchen Guru.

For our deployment strategy, we chose to deploy the backend and frontend of Kitchen Guru separately. This approach allowed us to optimize the performance and scalability of each component independently. We modified our code so that when the frontend makes a request, it uses the unique URL provided by Render for the backend service. This separation streamlined our deployment process and ensured efficient communication between the frontend and backend, enhancing the overall user experience. With Render, we anticipated a more intuitive deployment process, flexible scalability options, and a reliable platform that would support a smooth and efficient user experience for Kitchen Guru.

Conclusions

In conclusion, the Kitchen Guru project has traversed a remarkable journey from its conceptualization to its realization. Each phase—starting from a thorough domain analysis to customer validation, through to the meticulous system design and the agile development process—has contributed to the rich tapestry of this innovative culinary platform.

The domain analysis laid the groundwork, identifying the pressing needs within the culinary sphere and aligning our objectives with those insights. Kitchen Guru emerged as a solution not just to everyday cooking queries but as a beacon of sustainability and community in cooking. Idea validation, bolstered by positive user feedback, affirmed our direction and encouraged further refinement of the project's scope.

The system design phase translated our functional and non-functional requirements into tangible blueprints, utilizing an array of UML diagrams to ensure clarity and cohesion in our approach. Customer validation played a critical role in sharpening our features and revealing new avenues for growth, particularly emphasizing the community aspect and AI-integration within the platform.

Version control, spearheaded by GitHub, streamlined our collaborative efforts and allowed us to manage our codebase efficiently. Our team's division into backend and frontend specialists, each group working within dedicated repositories, optimized our development workflow and ensured focused expertise was applied to every aspect of Kitchen Guru.

As we deployed Kitchen Guru, we faced the challenges of integrating a rich user interface with robust backend services, ensuring scalability and a seamless user experience. The deployment strategy encompassed containerization with Docker and a PostgreSQL database, emphasizing security and performance.

Throughout the development and implementation, our team remained committed to quality, scalability, and a user-centric approach. The Kitchen Guru project served as a rich learning ground, reinforcing the importance of user feedback, agile methodologies, and the power of community in shaping a technology product.

Our experience with Kitchen Guru has underscored the transformative potential of technology in the culinary domain. We have learned the significance of user-centered design, the agility of development in a collaborative environment, and the importance of adapting to user feedback. As we conclude this report, we take pride in the strides made and look forward to the continued evolution of Kitchen Guru, confident that it will remain at the forefront of culinary innovation and community engagement.

Bibliography

[1] Hospitality Insights. "Top 2024 Food and Beverage trends revealed". https://hospitalityinsights.ehl.edu/top-food-beverage-trends-2024.

[2] The Ai Cuisine. "AI-Powered Cooking: The Next Generation of Home Kitchen Innovation". https://theaicuisine.com/ai-powered-cooking.

[3] Creately. "Use Case Diagram Tutorial (Guide with Examples)". https://creately.com/guides/use-case-diagram-tutorial/#:~:text=Use%20case%20diagrams%20provide%20a,external%20systems%20interact%20with%20it.

[4] Visual Paradigm. "Understanding Activity Diagrams in UML: A Comprehensive Guide". https://guides.visual-paradigm.com/understanding-activity-diagrams-in-uml-a-compreh#:~:text=An%20Activity%20Diagram%20is%20one%20of%20the%20behavioral,and%20decisions%20within%20a%20system%20or%20a%20process.#:~:text=An%20Activity%20Diagram%20is%20one,a%20system%20or%20a%20process.

[5] Visual Paradigm. "Understanding Sequence Diagrams: A Visual Blueprint of Software Interactions". https://guides.visual-paradigm.com/understanding-sequence-diagrams-a-visual-bluepri#:~:text=A%20sequence%20diagram%20is%20a,or%20perform%20a%20specific%20function.

[6] Visual Paradigm. "Modeling a Distributed System Using Deployment Diagram". https://guides.visual-paradigm.com/modeling-a-distributed-system-using-deployment-d#:~:text=Physical%20Representation%3A%20Deployment%20diagrams%20are,in%20visualizing%20the%20deployment%20architecture.

[7] Visual Paradigm. "Component Diagram vs deployment Diagram in UML". https://guides.visual-paradigm.com/component-diagram-vs-deployment-diagram-in-uml/#:~:text=,paradigm.com%E3%80%91.

[8] Atlassian. "What is version control?". https://www.atlassian.com/git/tutorials/what-is-version-control#:~:text=Version%20control%20also%20helps%20developers,some%20are%20better%20than%20others.

[9] Microsoft. "What is version control?". https://learn.microsoft.com/en-us/devops/develop/git/what-is-version-control.

[10] Github. "About Git". https://docs.github.com/en/get-started/using-git/about-git.