Nile University
School of Information Technology and Computer Science
Program of Computer Science

AKLNY

CSCI496 Senior Project II

Submitted in Partial Fulfilment of the Requirements

For the bachelor's degree in information technology and computer science Computer Science

Submitted by

Omar Adel Gamal	**202001270**
Omar Ayman Mohamed	**19105038**
Omar Fathallah	**202003116**
Abdelrahman Elsorot	**202000993**
Nourelden Haytham	**202000383**
Ziad Khalaf Ali	**202000545**

Supervised by
Dr. Tamer Arafa

Giza -Egypt
Fall 2024

Acknowledgment

We extend our deepest gratitude to our supervisor, Dr. Tamer Arafa, for his unwavering support, insightful guidance, and dedicated involvement at every project stage. his continuous and professional support has been invaluable.

Project Summary

The AKLNY project is a comprehensive mobile application developed to revolutionize home cooking by addressing key challenges in recipe discovery and culinary support through advanced technological integration. Leveraging machine learning and artificial intelligence, the application provides personalized recipe recommendations based on individual user preferences, dietary restrictions, and cooking skills. The platform offers a multifaceted approach to cooking assistance, featuring an AI-powered chatbot for real-time guidance, innovative image recognition technology for leftover management and nutritional analysis, and a sophisticated recommendation system that combines content-based and collaborative filtering. By integrating features like volume estimation, nutritional tracking, and interactive cooking support, AKLNY aims to transform the cooking experience, making it more accessible, enjoyable, and educational. The application not only helps users discover and prepare recipes more effectively but also promotes healthier eating habits, reduces food waste, and builds user confidence in the kitchen through comprehensive, personalized technological support.

Keywords: Mobile App, Personalized Cooking, Recipe Recommendations, Food Waste Management, Culinary Education, Machine Learning, Volume Estimation, Computer Vision, YOLO Detection, MiDaS Depth Model, Chatbot Integration, OpenAI GPT, Recommendation Systems, Nutritional Analysis, User-Centered Design, Food Recognition, Real-time Analysis, Interactive Cooking, Dietary Management, Django Framework

Table of Contents

List of Figures

List of Tables

List of **Abbreviations**

ABBREVIATION	MEANING
UI	User Interface
UX	User Experience
API	Application Programming Interface
NLP	Natural Language Processing
ORM	Object Relational Mapping
ML	Machine Learning
UCD	User Centered Design
CRUD	Create, Read, Update, Delete
JWT	JSON Web Token
REST	Representational State Transfer
DRF	Django REST Framework
WSGI	Web Server Gateway Interface
ASGI	Asynchronous Server Gateway Interface
SQL	Structured Query Language
XSS	Cross-Site Scripting
CSRF	Cross-Site Request Forgery
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
YOLO	You Only Look Once
GPT	Generative Pre-trained Transformer
SMTP	Simple Mail Transfer Protocol
AWS	Amazon Web Services
AI	Artificial Intelligence

Chapter 1
Introduction

1.1 Background

In recent years, the landscape of home cooking has undergone significant changes. Traditional sources of recipes, like cookbooks, have been supplemented by a vast array of online resources, including websites and cooking shows. While this abundance of information offers unprecedented access to recipes worldwide, it has also created new challenges for home cooks.

One major challenge is the sheer volume of recipes available, which can be overwhelming and make it difficult for users to find recipes that match their preferences, skills, and dietary needs. Additionally, existing recipe apps often fail to provide personalized recommendations, instead offering generic lists of recipes that may not be suitable for individual users.

Another challenge is the lack of guidance and support for home cooks. Many recipe apps focus solely on providing recipes without offering essential cooking tips, such as portion control, food safety information, and strategies for managing leftovers. This lack of guidance can lead to frustration and a less enjoyable cooking experience.

To address these challenges, we propose the development of AKLNY, a mobile application designed to empower home cooks and enhance their cooking experience. AKLNY will differentiate itself from existing recipe apps by offering a personalized approach to recipe recommendations. By analyzing user preferences, dietary restrictions, and cooking skills, AKLNY will provide tailored recipe suggestions that are more likely to resonate with users.

Moreover, AKLNY will go beyond recipe provision by offering comprehensive cooking guidance. This includes tips on portion control, food safety, and creative ways to use leftovers, all aimed at helping users cook more efficiently and reduce food waste.

Additionally, AKLNY will feature a chatbot that interacts with users, guiding them through the cooking process and providing real-time assistance. This interactive element aims to make the cooking experience more engaging and enjoyable for users.

Overall, AKLNY seeks to revolutionize home cooking by offering a personalized, user-friendly, and informative platform that empowers users to cook with confidence and creativity. Through its innovative features and holistic approach, AKLNY aims to transform home cooking into a rewarding and fulfilling experience for everyone.

1.2 Motivation

Two main ideas drive this project:

1. **Information Overload:** There are so many recipes available online that it can be overwhelming to find the one you want to cook. Sorting through countless options takes time and can be frustrating.

2. **Lack of Personalization:** Many recipe apps do not tailor recommendations to individual tastes and cooking habits. This makes it hard for users to discover new and exciting dishes that suit their preferences and dietary needs.

1.2.1 Real-World Problems and Applications

AKLNY aims to solve these issues by providing a personalized and user-friendly cooking experience. By filtering recipes based on user preferences and dietary restrictions, AKLNY saves time and reduces frustration. The app's chatbot engages users, guiding them through the cooking process and making it more interactive and enjoyable. Additionally, AKLNY's focus on portion control and leftover management helps users cook healthier meals and reduce food waste. This project addresses real- world problems by making home cooking simpler, more personalized, and more enjoyable for everyone.

1.3 Objectives

The primary objective of developing AKLNY is to address the challenges faced by home cooks in finding suitable recipes amidst the vast number of online resources. By leveraging machine learning and user data analysis, AKLNY aims to:

- **Personalize Recipe Recommendations:** Tailor recipe suggestions based on user preferences, dietary restrictions, and cooking skills to ensure that users receive recipes that are relevant and appealing to them.

- **Enhance User Confidence:** Provide essential cooking guidance and multileveled recipes to help users feel more confident in the kitchen, regardless of their cooking experience.

- **Minimize Food Waste:** Offer resourceful tips for managing leftovers and guidance on portion control to help users cook efficiently and reduce food waste.

- **Create an Interactive Experience:** Incorporate a chatbot to engage users, providing real- time assistance and making the cooking process more interactive and enjoyable.

- **Promote Healthy Eating:** Suggest nutritious recipes and add the nutrition facts of the recipe chosen by the user to help him choose a healthier option if wanted.

- **Provide Comprehensive Support:** Develop a user-friendly platform that not only offers recipes but also includes features like food safety information, Grocery lists.

- **Transform Home Cooking:** Make home cooking a fun, successful, and enriching experience by combining personalized recommendations, interactive features, and educational content into one comprehensive app.

1.4 Scope

- **Development of a Mobile Application:** The project will focus on developing a mobile application for the IOS and Android platforms.

- **User-Friendly Interface:** Implementation of a user-friendly interface for recipe browsing, personalized recommendations, and educational content to enhance user experience.

- **Integration of Machine Learning Algorithms:** Incorporation of machine learning algorithms to provide personalized recipe suggestions based on user preferences and behavior, continuously improving the recommendation system over time.

- **Essential Cooking Guidance:** Provision of essential cooking guidance, including portion control tips, food safety information, and strategies for managing leftovers, to assist users in their cooking endeavors.

- **Personalization:** AKLNY offers personalized recommendations tailored to each user's preferences, dietary restrictions, and cooking abilities, surpassing generic recipe provision.

- **Holistic Approach:** AKLNY not only provides recipes but also offers essential cooking guidance, portion control tips, food safety information, and leftover management strategies, catering to the diverse needs of home cooks.

- **User Engagement:** AKLNY incorporates features like user profiles, favorite recipes, and social sharing to foster a sense of community and encourage continued engagement with the platform.

The project scope excludes:

- Advanced features requiring extensive resources or expertise beyond the project's current scope.

- Integration with external hardware or software significantly increases complexity or cost.

1.5 Significance of the Study

The AKLNY project has significant implications for the field of home cooking and mobile applications. It offers several potential benefits and applications:

- **Improved Cooking Experience:** AKLNY enhances the cooking experience by providing personalized recipe recommendations, cooking guidance, and tips for managing leftovers. This makes cooking at home more enjoyable and less stressful.
- **Healthier Eating Habits:** By offering nutritious recipes and portion control guidance, AKLNY promotes healthier eating habits among users, contributing to improved overall health and well-being.
- **Reduced Food Waste:** AKLNY's tips for managing leftovers help reduce food waste, aligning with sustainable practices and environmental conservation efforts.
- **User Empowerment:** The app empowers users to explore new recipes, learn cooking skills, and gain confidence in the kitchen, leading to a sense of accomplishment and satisfaction.

1.6 Outline the structure of the report

In Chapter 2 extends our exploration by examining existing literature, including historical perspectives, theoretical frameworks, and previous research findings, identifying gaps in current knowledge.

Moving forward to Chapter 3, our focus shifts to the project itself, with detailed discussions in System Description on the system's context, boundaries, and objectives, accompanied by a description of system requirements. We also address design constraints and detail the research design, encompassing architectural, data, and interaction design, along with data flow diagrams and integration with external systems. Additionally, Chapter 4 provides an in-depth look into the implementation phase, covering programming languages, code structure, data structures, and databases, along with preliminary quantitative and qualitative results.

In Chapter 5, we engage in the discussion and conclusion, offering interpretations of results, comparisons with previous studies, acknowledgment of limitations. Finally, Chapter 6 presents the summary of findings from our research and implementation, followed by suggestions for future work and potential enhancements to the system.

Each chapter builds upon the previous ones to create a comprehensive understanding of the AKELNY system, from its conceptual foundations through to its practical implementation and future possibilities.

Chapter 2
Related Work

2.1 Introduction to Literature Review

This literature review aims to methodically analyze and integrate existing research relevant to the development of AKLNY, a mobile application designed to enhance home cooking through personalized recipe recommendations and cooking guidance. The primary purpose is to identify how past innovations and current technologies have been harnessed to transform cooking practices and to pinpoint areas where further advancements can be made. This chapter will first provide a historical overview of technology in cooking, followed by relevant theoretical frameworks and a summary of pivotal research findings. Finally, it will assess the current state of the field, emphasizing ongoing challenges and emerging trends.

2.2 Historical Perspective

The intersection of technology and cooking has evolved dramatically over the years. In the early days, digital culinary tools were simple databases with limited interactivity, primarily serving as electronic replacements for traditional cookbooks. With the proliferation of the internet in the late 1990s and early 2000s, these databases became more sophisticated, offering searchable interfaces and user-generated content. The advent of smartphones and tablets marked a significant milestone, introducing apps that not only stored recipes but also began to incorporate features like shopping lists, meal planning, and basic dietary tracking.

The transition from static databases to dynamic, interactive platforms set the stage for integrating more advanced technologies such as AI and machine learning. These technologies allowed for the development of systems capable of learning from user input and behavior, suggesting not just any recipe but the most contextually appropriate options based on the user's past preferences, current inventory, and nutritional goals.

2.3 Theoretical Framework

The development of AKLNY is rooted in two main theoretical frameworks: personalized technology adoption and user-centered design. Personalization technology explores how systems can adapt their functionalities to meet the individual needs of users. This involves algorithms capable of learning user preferences over time and tailoring content accordingly. Research in this area often draws on adaptive hypermedia and recommender system theories.

User-centered design (UCD) is a framework that focuses on designing software with the user's needs, limitations, and context as the core driving factors. UCD in cooking apps involves understanding the cooking process, user experience in meal preparation, and the educational needs regarding cooking skills and nutritional information. Incorporating UCD ensures that AKLNY not only recommends recipes but also aids in educational and practical aspects of cooking, promoting a holistic approach to meal preparation.

2.4 Previous Research and Studies

The literature review synthesizes insights from ten studies relevant to the development of AKLNY, a mobile application that personalizes recipe searches and enhances cooking experiences. These studies reveal significant advancements and opportunities in using mobile apps to influence dietary behaviors, integrate user preferences, and utilize AI and machine learning for tailored recipe suggestions. For instance, research by Garcia et al. (2016) highlights the positive impact of community cooking programs on cooking confidence and skills, suggesting the

importance of educational content in cooking apps. Bangale et al. (2021) explore a recipe recommendation system that uses content-based filtering to suggest recipes based on user- provided ingredients, aligning closely with AKLNY's personalization goals. Furthermore, studies like Kumar & Smith (2021) discuss the broad applications of AI and ML across the food industry, supporting the technological framework behind AKLNY. However, these studies also indicate a gap in effectively combining health metrics with dynamic user preferences in a way that adapts over time, a niche AKLNY aims to fill.

The reviewed literature underscores the potential of mobile technology to transform cooking habits through personalized and health-focused features, yet it also points out the lack of applications that seamlessly integrate comprehensive nutritional guidance with personalized technology adaptation. The research on systems like MySusCof app and various AI-driven recipe adaptation tools shows that while technological capabilities exist, their application in providing a holistic, user-centered cooking aid is still lacking.

AKLNY seeks to bridge these gaps by leveraging machine learning to refine recipe recommendations based on evolving user preferences and incorporating educational content to bolster user confidence and knowledge. This approach not only addresses the identified needs for more sophisticated personal dietary management but also sets a new standard for integrating culinary skills enhancement with personal health goals.

2.5 Current State of the Field

The current state of digital culinary assistance is characterized by rapid advancements in AI, machine learning, and user interface design. Modern applications leverage vast datasets to understand and predict user preferences and are beginning to incorporate complex nutritional algorithms to provide more health-focused recommendations.

Despite these advancements, challenges persist, especially in achieving a balance between sophisticated functionality and user-friendly design. Issues such as data privacy, the accuracy of dietary tracking, and the personalization of health recommendations without medical oversight are increasingly pertinent. Moreover, as technology advances, there is a growing need to ensure these tools are accessible and beneficial to a diverse range of users, including those with varying levels of cooking experience and dietary requirements.

There are several business applications and products available on the market that cater to nutrition's applications. Some popular examples include:

Lifesum is a popular app designed to help simplify weight loss. It provides a variety of tools, including a recipe library with healthy meal ideas, a food log to track what you eat, and a calorie counter to help manage your daily intake. These features make it easier for users to achieve their weight loss goals by offering practical and easy-to-use resources for healthy living.

Figure 1: Lifesum Logo

Yummly Recipes & Cooking Tools is a website and mobile app from the USA that helps users find recipes. It provides personalized recipe recommendations and has a powerful search engine to help you discover new dishes. With Yummly, you can easily find recipes that match your taste preferences and dietary needs.

Figure 2: Yummly Logo

Chapter 3
Materials and Methods

3.1 System Description

3.1.1 Context of the System

AKLNY is designed as a comprehensive mobile application aimed at enhancing the home cooking experience. It functions within the digital ecosystem of culinary and dietary management, primarily interacting with users who seek to improve their cooking skills, discover new recipes, manage their dietary restrictions, and reduce food waste effectively. The application stands at the intersection of culinary arts, dietary management, and user- friendly technology, offering a personalized cooking assistant on mobile devices.

3.1.2 Boundaries of the System

The boundaries of the AKLNY system encompass the functionalities directly controlled and provided through the app:

- **Recipe Discovery and Recommendation**: Utilizing advanced algorithms to suggest recipes based on user preferences, dietary needs, and available ingredients.
- **User Profile Management**: Enabling users to input and modify their dietary preferences, allergies, and cooking skills.

- **Education and Guidance**: Providing tips on food safety, portion control, and cooking techniques.
- **Leftover Management**: Suggesting creative ways to use leftovers to minimize waste.
- **External Interactions**: While AKLNY primarily functions as a standalone system, it interacts with external databases for recipes, ingredients, and nutritional information.

Context Diagram

Below is a simplified Context Diagram for AKLNY, illustrating its main interfaces and external interactions:

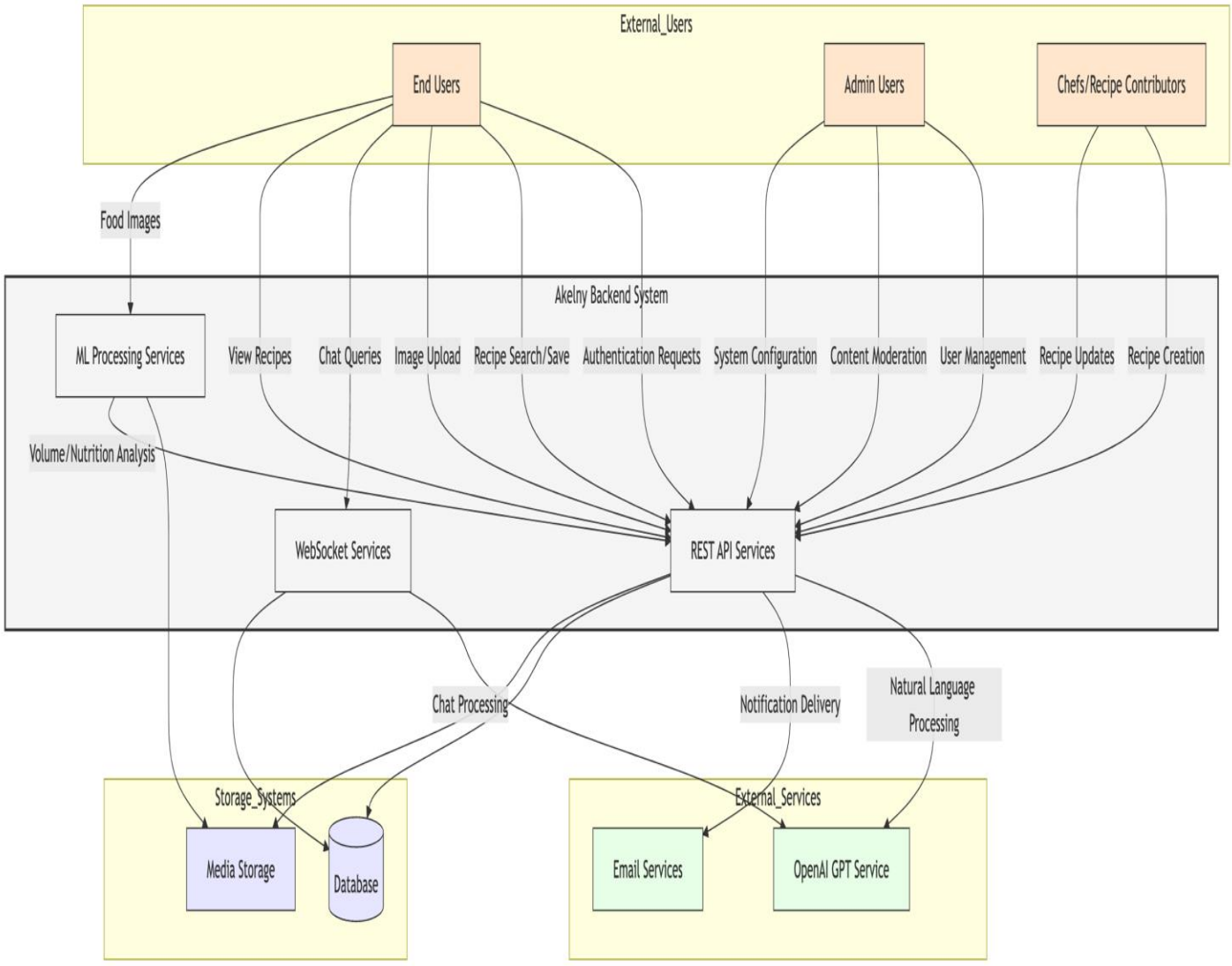

Figure 3: Context Diagram

3.1.3 Objectives and Requirements from the User's Perspective

- **Personalization**: Users desire a system that adapts to their dietary preferences, cooking skills, and available kitchen tools.
- **Variety and Discovery**: A platform that introduces users to new culinary styles and dishes while considering their dietary restrictions and food preferences.
- **Educational Content**: Users look for guidance on improving cooking skills, understanding nutritional content, and learning food safety protocols.
- **Efficiency and Timesaving**: Solutions that help users quickly plan meals with available ingredients, reducing the time spent on meal planning and preparation.
- **Social Interaction**: Features that allow users to share recipes, tips, and successes with other users, enhancing community engagement.
- **Sustainability**: Tools to help minimize food waste and promote environmentally friendly cooking practices.

3.1.4 Desirable Characteristics (Wish List)

- **Interactive Meal Planning**: Incorporate features that allow users to input their weekly dietary goals and preferences, and have the app generate a meal plan that adjusts recipes based on nutritional needs and ingredient availability.

- **Virtual Pantry Management**: Develop a system that tracks ingredients in the user's kitchen, suggests recipes based on these ingredients, and notifies users when supplies are running low or nearing expiration.

- **Customizable User Profiles**: Enable users to create and customize profiles that can store multiple dietary preferences, allergen information, and past recipe successes, which can be adjusted as the user's tastes and dietary needs evolve.

- **Integrated Diet and Health Tracking**: Offer integration with fitness and health tracking apps to provide comprehensive health management, where users can see the impact of their cooking habits directly linked to their health metrics.

- **Cooking Mode Interface**: An easy-to-read, step-by-step cooking mode, possibly with voice commands and hands-free controls, that users can interact with while cooking to keep screens clean and hands unoccupied.

- **Social Sharing and Challenges**: Facilitate a social platform within the app where users can share their creations, fostering a community of food enthusiasts.

3.2 System Requirements

The AKLNY app is designed to provide a personalized and comprehensive cooking experience through a series of interconnected functionalities. Below is a high-level overview of the general functionality represented in a Use Case Diagram, illustrating the primary interactions between the users and the system:

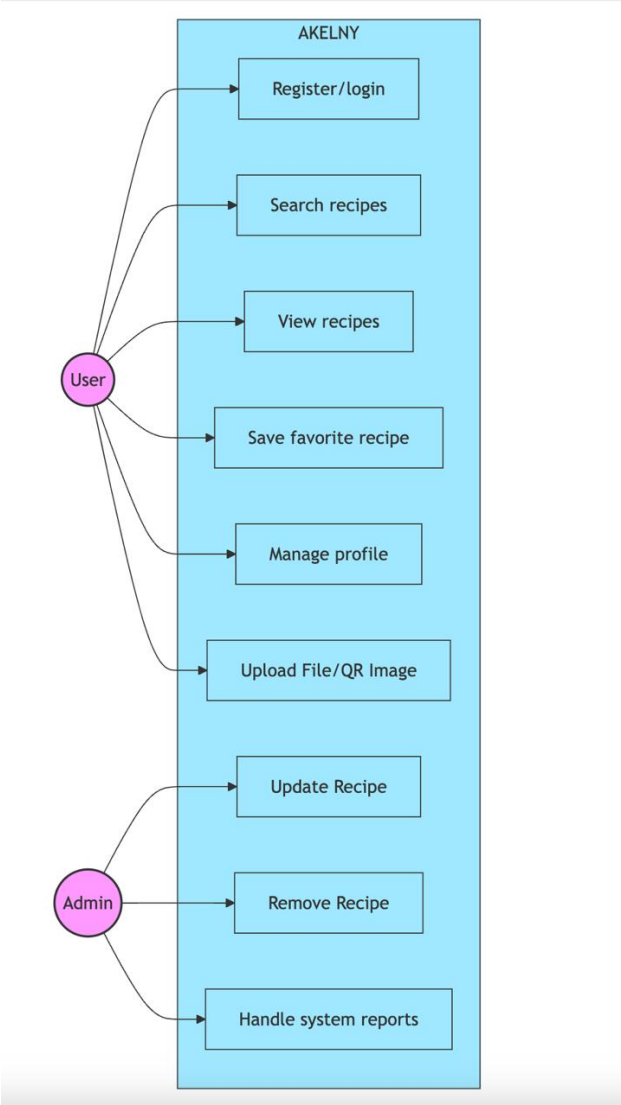

Figure 4: Use Case Diagram

3.2.1 **User use cases**

Use Cases	Description
Register/Login	Allows the user to create a new account or log into an existing account to access personalized features.
Search Recipes	Enables the user to search for recipes based on specific criteria such as ingredients, cuisine, or dietary needs.
View Recipe	Provides detailed views of recipes, including ingredients, cooking steps, and nutritional information.
Save Favorite Recipe	Allows users to save recipes they like for easy access in the future.
Manage Profile	Users can update their personal information, dietary preferences, and other settings.
Receive Notifications	Users receive alerts and notifications about new recipes, updates, or other relevant information.
Submit Feedback	Enables users to provide feedback about the app, which can include bug reports, suggestions, or compliments.
Upload File/Image	Users can upload images and files related to recipes, such as photos of their dishes or custom recipes in document format.

Table 1: User Use Cases

3.2.2 **Admin use cases**

Use Case	Description
Update Recipe	Admins can update existing recipes to correct errors or refresh content based on user feedback or new trends.
Remove Recipe	Allows admins to delete recipes from the database, typically used in case of obsolescence or inaccuracies.
Manage User Feedback	Admins review and respond to user feedback, making changes to the app or its content as necessary.
Handle System Updates	Admins perform and manage updates to the system, ensuring new features are integrated and maintained.

Table 2: Admin Use Cases

3.2.3 **Project Requirements**

Functional Requirements:

- The app must allow users to create and manage a personal account.
- The app must offer personalized recipe recommendations using a machine learning algorithm.
- The app must provide notifications for ingredient replenishment and recipe suggestions based on pantry items.
- The app should allow users to share recipes and their cooking process .

  .
- The app must allow users to upload and manage images and files related to their recipes and culinary experiences.

Non-Functional Requirements:

- **Security**: User data, including personal preferences and dietary information, must be securely stored and encrypted.
- **Reliability**: The app should be operational 24/7 with minimal downtime.
- **Maintainability**: The code should be well-documented and structured for easy updates and bug fixes.
- **Portability**: The app should be compatible with Android platforms.
- **Extensibility**: The architecture should support easy integration of new features such as augmented reality cooking aids.

3.2.4 Software Interfaces

- **Data Interchange**: AKLNY interacts with external recipes and nutritional databases via RESTful APIs to retrieve and send data.

3.2.5 Hardware Interfaces

- **Mobile Devices**: Designed to interact with the hardware features of smartphones, like touchscreens for navigation.

3.2.6 Network Interfaces

- **API Services**: Connects with various external APIs for real-time data fetching and updates.

3.3 Design Constraints

3.3.1 Network Dependency

The app should be designed to perform some core functionalities offline, such as accessing saved recipes or using certain features without real-time updates, to accommodate users with limited or intermittent internet access.

3.3.2 Scalability

The system architecture must support scalability to handle increases in user numbers and data volume without performance degradation.

3.3.3 Timeline

Adherence to a strict timeline for phases of design, development, testing, and release must be maintained to ensure market relevance and timely delivery of the app to users.

3.3.4 User Interface Design

The interface must be intuitive and engaging for a diverse user base with varying levels of technical proficiency and cooking experience. This includes considerations for minimalistic design to facilitate ease of navigation and clarity.

3.4 Research Design

The research design for AKLNY project encompasses a comprehensive strategy that integrates various methodologies to ensure the development and deployment of an effective and user-friendly mobile application for personalized cooking guidance. This section outlines the overall plan and the specific research methods and techniques employed.

3.4.1 Overall Plan

The overarching goal of the research design is to develop a mobile application that meets the needs of diverse users looking to enhance their home cooking experience through personalized recipes, dietary management, and interactive features. The project will follow an iterative development process, incorporating stages of conceptualization, design, development, testing, and feedback integration. This approach ensures continuous improvement and adaptation based on user input and technological advancements.

3.4.2 Research Methods and Techniques

- **Literature Review**:
    - **Purpose**: To gather existing knowledge about mobile application successes in similar domains.
    - **Process**: Review academic papers, market reports, and existing applications focusing on user engagement, technology integration, and user satisfaction.

- **Market Analysis**:
    - **Purpose**: To identify market needs and gaps in current offerings and to understand the competitive landscape.
    - **Process**: Analyze current market offerings, user reviews, and feature sets of competing apps. This analysis helps in identifying unique value propositions for AKLNY.
- **Persona Development**:
    - **Purpose**: To create detailed user personas that represent the different user types who might use AKLNY.
    - **Process**: Based on survey and interview data, develop personas that guide the user interface design, functionality, and marketing strategies.
- **Prototype Development**:
    - **Purpose**: To translate the research findings into a tangible product that can be interactively tested.
    - **Process**: Develop initial prototypes using tools like Sketch or Figma, focusing on user experience (UX) and user interface (UI) design.

Below is a high-level flow chart illustrating the flow of the activities and milestones of AKLNY system

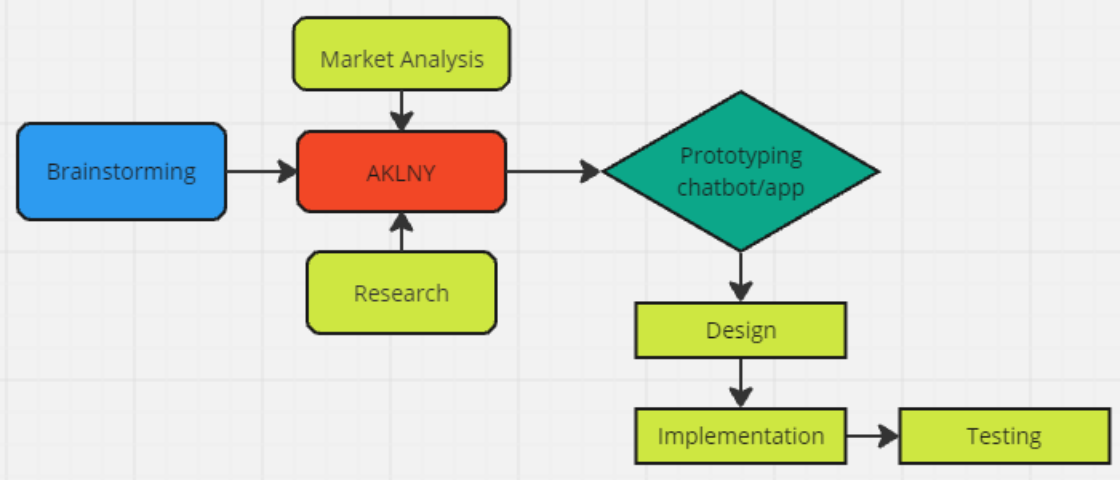

Figure 5: Milestones Flowchart

3.5 Architectural Design

The architectural design of the AKLNY mobile application is structured to support robust performance, scalability, and security while providing a seamless user experience. Below is an overview of the system architecture, which includes a breakdown of the main components and their interactions.

3.5.1 **Overview of System Architecture**

The AKLNY app utilizes a microservices architecture to ensure modularity, ease of maintenance, and the ability to scale individual components of the application independently based on demand.

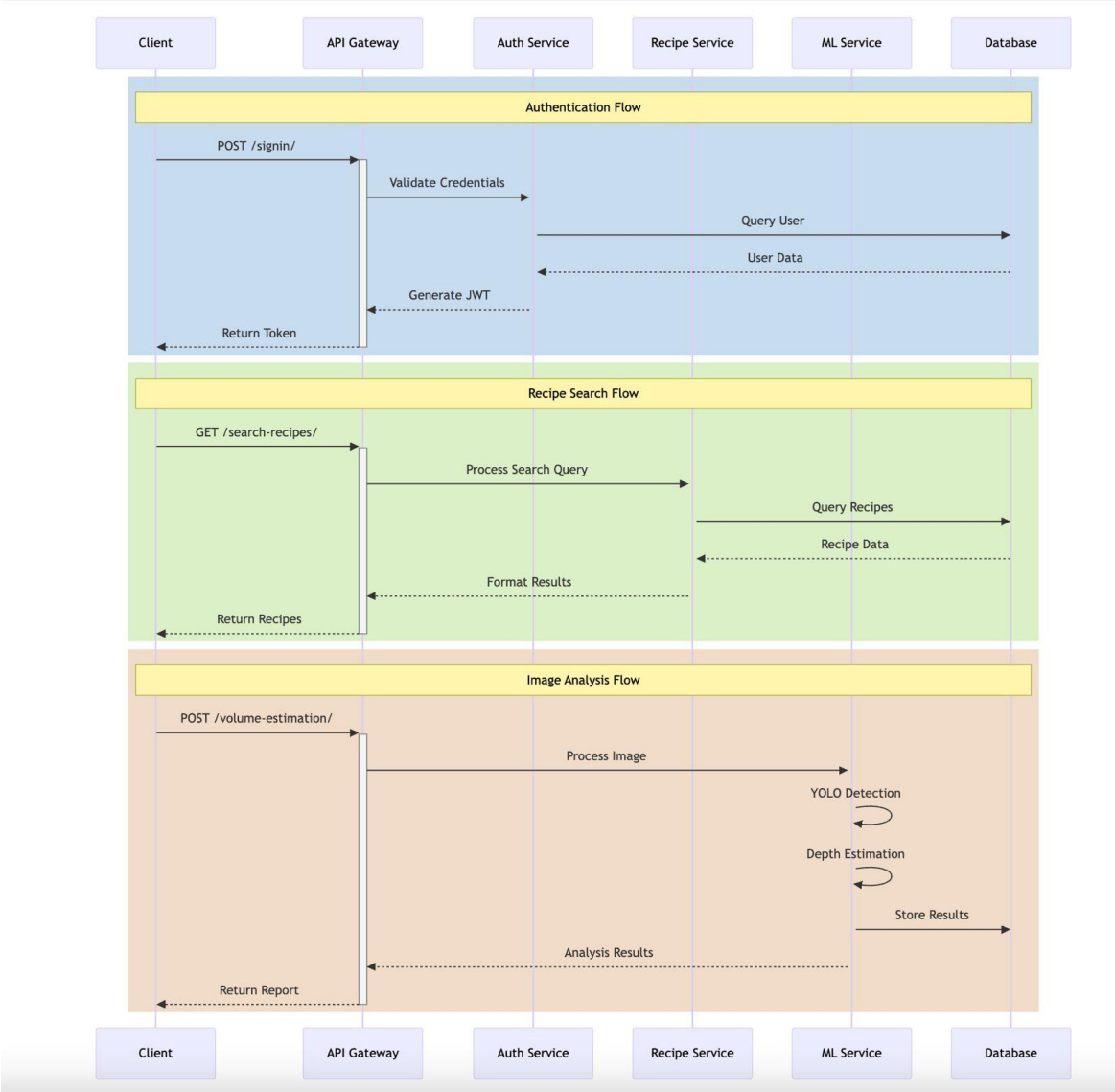

Figure 4: Workflow Diagram

3.5.2 Core Components

1. **User Interface (UI) Layer**: This is the front-end of the application, through which users interact with the app. It is built to be responsive and intuitive across different mobile platforms.

2. **Business Logic Layer**: Handles the application's core functionality including user profile management, recipe recommendations, and integration with external APIs for ingredient and recipe data.

3. **Data Access Layer**: Manages interactions between the business logic layer and the database. It handles data querying, storage, and retrieval.
4. **Database Layer**: Stores user data, recipes, preferences, and other application data. It ensures data integrity and security.

5. **API Gateway**: Acts as the entry point for all clients, routing requests to appropriate microservices and handling load balancing, authentication, and authorization.

6. **Authentication Service**: Manages user authentication and ensures that users are who they claim to be, handling login procedures and security protocols.

7. **Recommendation Engine**: Utilizes machine learning algorithms to provide personalized recipe recommendations based on user preferences and behavior.

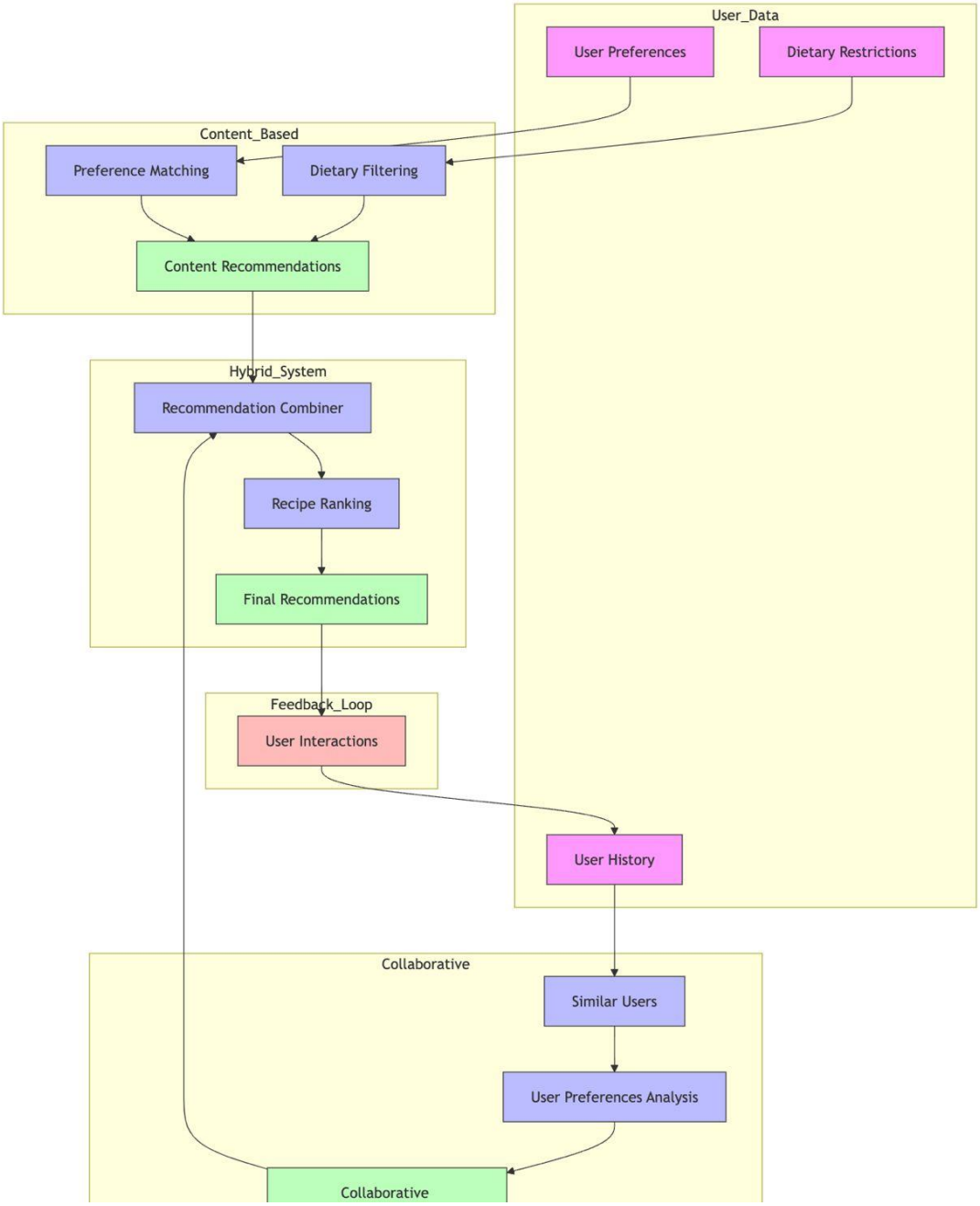

Figure 5: Recommendation Engine Dataflow

3.5.3 **Supporting Components**

1. **Notification Service**: Manages the sending of push notifications to users about recipe updates, ingredient lists, and other relevant information.

2. **Analytics Service**: Gathers and processes usage data to help refine user experience and provide insights into app performance.

Architectural Diagram

Below is a high-level architectural diagram illustrating the interaction between the various components of the AKLNY system:

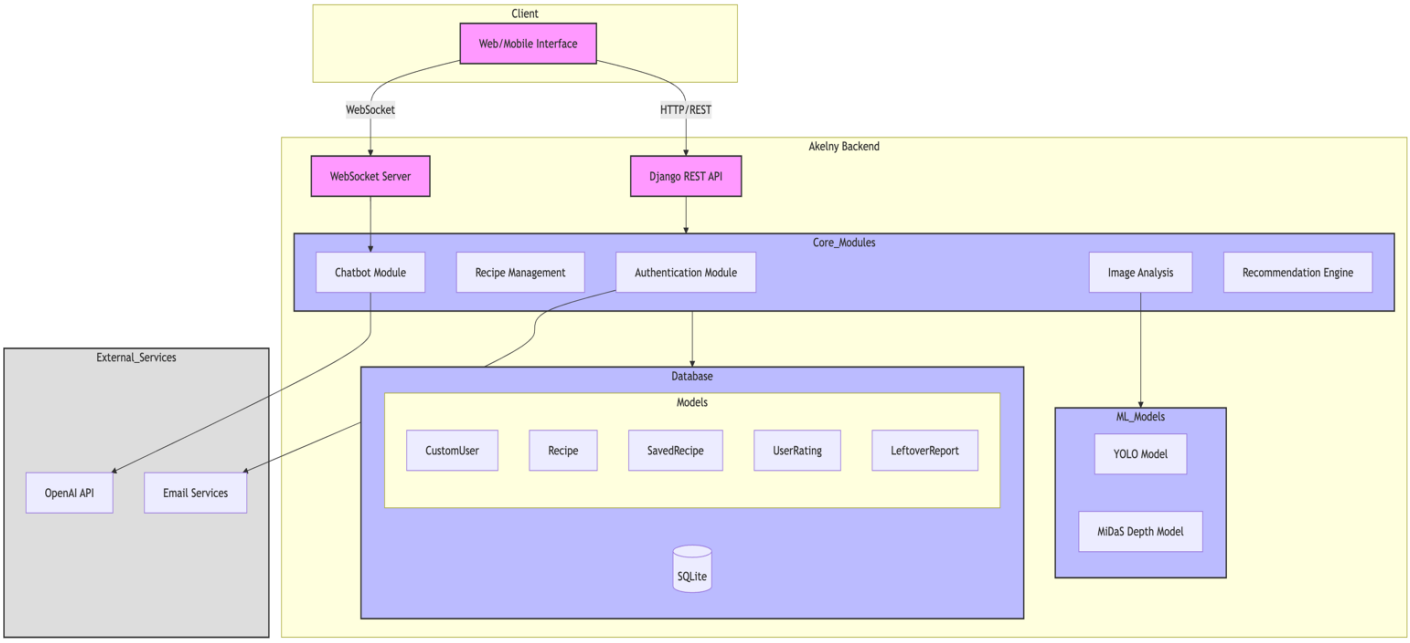

Figure 6: System Architecture Diagram

Explanation of Components and Their Interactions:

1. **User Interface (UI) Layer:**

    - Users interact with Akelny through web and mobile interfaces

- Handles rendering of recipes, food analysis results, and chatbot interactions
- Implements real-time updates through WebSocket connections for chat features
- Manages file uploads for food images and recipe submissions

2. **Business Logic Layer:**
   - Processes core functionalities through Django REST API endpoints
   - Implements recipe management, user interactions, and ratings systems
   - Handles image processing and food analysis workflows
   - Contains the hybrid recommendation system combining content-based and collaborative filtering
   - Manages chatbot interactions through OpenAI integration

3. **Data Access Layer:**
    - Implements Django models for structured data access
    - Handles CRUD operations for recipes, users, and interactions
    - Manages file storage for images and media content
    - Implements caching strategies for frequently accessed data
    - Provides data validation and formatting through Django serializers
4. **Database Layer:**
    - Uses SQLite for storing structured data including:
        - User profiles and preferences
        - Recipe information and metadata
        - User interactions and ratings
        - Saved recipes and reports
    - Implements relationships between models for efficient data retrieval
    - Maintains data integrity through Django's ORM constraints

5. **API Gateway:**

    - Routes requests through Django URL patterns
    - Implements CORS policies for secure cross-origin requests
    - Manages authentication through JWT tokens
    - Handles rate limiting and request validation
    - Provides API documentation and endpoint management

6. **Authentication Service:**

    - Custom user authentication system using Django's authentication framework
    - Implements email-based password reset functionality
    - Supports multiple email providers (Gmail, Outlook, Yahoo, etc.)
    - Manages user sessions and token-based authentication
    - Handles user registration and profile management

7. **Machine Learning Services:**
    - YOLO model integration for food detection and segmentation
    - MiDaS depth estimation for volume calculation
    - Nutrition analysis based on food detection
    - Real-time image processing pipeline
    - ML model management and loading
8. **WebSocket Service:**
    - Handles real-time chatbot communications
    - Manages chat sessions and message routing
    - Integrates with OpenAI's GPT service
    - Maintains connection state and chat history
9. **External Service Integration:**
    - OpenAI API integration for chatbot functionality
    - Email service providers for notifications
    - File storage systems for media content
    - Integration with ML model repositories

10. Security Layer:

- Implements Django's security middleware
- JWT-based authentication for API requests
- CSRF protection for forms and requests
- Secure password handling and storage
- Rate limiting for API endpoints

This architecture ensures:

- Scalability through modular design
- Security through multiple authentication layers
- Performance through optimized data access
- Reliability through error handling and logging
- Maintainability through clear separation of concerns
- Extensibility for adding new features and services

3.6 Data Design

The dataset is structured around Django models, which handle recipe management, user interactions, and food analysis. Here's the detailed data model:

1. **Entities and Attributes**

CustomUser Attributes:

Field	Type	Description
id	AutoField	Primary key
email	EmailField	Unique email (required)
username	CharField	Unique username
password	CharField	Hashed password
phone_number	CharField	Optional
address	TextField	Optional
dietary_preferences	TextField	Optional
allergies	TextField	Optional
cooking_skill_level	CharField	Optional
date_of_birth	DateField	Optional
profile_picture	ImageField	Optional
created_at	DateTimeField	Auto-added
updated_at	DateTimeField	Auto-updated

Table 3: CustomUser Attributes

Recipe Attributes:

Field	Type	Description
id	AutoField	Primary key
name	CharField	Recipe name
category	CharField	Cuisine category
serving_size	CharField	Optional
calories	DecimalField	Optional
fats	DecimalField	Optional
carbs	DecimalField	Optional
protein	DecimalField	Optional
time	IntegerField	Preparation time
ingredients	TextField	List of ingredients
instructions	TextField	Cooking instructions
image	ImageField	Optional
density	DecimalField	For volume calculation
created_by	ForeignKey	Reference to CustomUser
created_at	DateTimeField	Creation timestamp

Table 4: Recipe Attributes

SavedRecipe Attributes:

Field	Type	Description
id	AutoField	Primary key
user	ForeignKey	Reference to CustomUser
recipe	ForeignKey	Reference to Recipe
saved_at	DateTimeField	When recipe was saved

Table 5: SavedRecipe Attributes

UserRating Attributes:

Field	Type	Description
id	AutoField	Primary key
user	ForeignKey	Reference to CustomUser
recipe	ForeignKey	Reference to Recipe
rating	DecimalField	Rating value (0.5-5.0)
created_at	DateTimeField	Rating timestamp

Table 6: UserRating Attributes

LeftoverReport Attributes:

Field	Type	Description
id	AutoField	Primary key
user	ForeignKey	Reference to CustomUser
image	ImageField	Food image
report	JSONField	Analysis results
created_at	DateTimeField	Report timestamp

Table 7: LeftoverReport Attributes

UserInteraction Attributes:

Field	Type	Description
id	AutoField	Primary key
user	ForeignKey	Reference to CustomUser
recipe	ForeignKey	Reference to Recipe
action	CharField	View/Click/Save
timestamp	DateTimeField	Interaction time

Table 8: UserInteraction Attributes

EmailConfiguration Attributes:

Field	Type	Description
id	AutoField	Primary key
provider	CharField	Email provider
email_host_user	EmailField	Host user
email_host_password	CharField	Host password
email_from	EmailField	From address
active	BooleanField	Active status

Table 9: EmailConfiguration Attributes

2. **Relationships**

- One-to-Many Relationships:
    - CustomUser → Recipe (created_by)
    - CustomUser → SavedRecipe (user)
    - CustomUser → UserRating (user)
    - CustomUser → LeftoverReport (user)
    - CustomUser → UserInteraction (user)
    - Recipe → SavedRecipe (recipe)
    - Recipe → UserRating (recipe)
    - Recipe → UserInteraction (recipe)
- Model Constraints:
    - Unique together constraints:
        - UserRating: (user, recipe)
        - SavedRecipe: (user, recipe)
    - Required fields:
        - CustomUser: email, username
        - Recipe: name, category
        - UserRating: rating

3. **Data Integrity**

- Foreign Key Constraints:
    - ON DELETE CASCADE for user-related records
    - ON DELETE SET_NULL for recipe creator
    - Proper indexing on frequently queried fields

4. **Additional Features**

- Computed Fields:
    - Recipe.average_rating: Calculated from UserRating
    - Recipe volume calculation from density and dimensions
    - Hybrid recommendations based on user interactions
- File Storage:
    - Profile pictures stored in 'profile_pictures/'
    - Recipe images stored in 'recipe_images/'
    - Leftover report images stored in 'leftover_reports/'

5. **Data Validation**

- Model-level validation:
    - Email format validation
    - Rating range validation (0.5-5.0)
    - Password complexity requirements
    - Image file type and size restrictions
- Serializer-level validation:
    - Custom validation for recipe creation
    - User input sanitization
    - Data type conversion and formatting

3.7 Interaction Design

The Akelny system is designed to provide users with an intuitive and comprehensive food management and recipe discovery experience. Through a seamless integration of advanced technologies and user-friendly interfaces, users can explore, analyze, and interact with recipes while receiving personalized assistance throughout their cooking journey.

User Authentication and Personalization Users begin their journey by creating an account through a straightforward registration process. During this initial setup, users provide essential information about their dietary preferences, allergies, and cooking skill level. This information forms the foundation for a personalized experience within the system. Users can easily manage their profiles, update preferences, and customize their experience as their needs change over time.

Recipe Discovery and Management The heart of the system lies in its recipe management capabilities. Users can explore a vast collection of recipes through an advanced search system that supports filtering by category, cuisine type, and user ratings. Each recipe provides comprehensive details including ingredients, step-by-step instructions, nutritional information, and preparation time. Users can save their favorite recipes, rate them, and maintain a personalized collection of cooking resources. The system tracks user interactions to continuously improve recipe recommendations.

Intelligent Food Analysis One of the system's standout features is its advanced food analysis capability. Users can upload images of their food for real-time analysis using YOLO-based detection and depth estimation technology. This provides accurate volume estimation, nutritional content analysis, and detailed reports about leftovers. The system processes these images instantly, offering visual feedback through segmentation results and comprehensive nutritional breakdowns.

AI-Powered Chat Assistance The system incorporates a sophisticated chat assistant powered by OpenAI technology. Through WebSocket-based real-time communication, users can receive instant guidance on cooking techniques, ingredient substitutions, and recipe modifications. The chat system maintains context awareness, allowing for natural conversations and personalized assistance based on user preferences and history.

Smart Recommendations and Learning Through a hybrid recommendation system, Akelny provides personalized recipe suggestions combining both content-based and collaborative filtering approaches. The system learns from user interactions, continuously adapting its recommendations to better match individual preferences and dietary restrictions. This creates an increasingly personalized experience as users engage with the platform.

Communication and Notifications To keep users engaged and informed, the system implements a comprehensive notification system. Users receive updates about recipe changes, system notifications, and responses to their feedback through configurable notification channels. This ensures users stay connected with the platform while maintaining control over their notification preferences.

Administrative Oversight Behind the scenes, administrators have access to powerful tools for content management and system maintenance. They can moderate recipes, manage user accounts, configure system settings, and monitor performance. This ensures the platform maintains high-quality content and optimal performance while providing a secure environment for all users.

The interaction design prioritizes user experience through:

- Seamless technology integration
- Intuitive interface design
- Real-time processing and feedback
- Personalized user experiences
- Comprehensive recipe management
- Advanced food analysis
- Interactive support systems
- Robust security measures

3.8 Flow Diagram

The Akelny system flowchart illustrates a comprehensive user journey and data flow through the application, starting from the initial user authentication to complex food analysis and recipe management. The system's architecture ensures a seamless user experience while maintaining robust security and efficient data processing.

Authentication and Entry When users first interact with the system, they encounter an authentication checkpoint. New users proceed through a registration process that collects essential information including dietary preferences, allergies, and personal details. This data undergoes validation before being securely stored in the database. Existing users can directly log in with their credentials. After successful authentication, users gain access to the main menu, which serves as the central hub for all system functionalities.

Recipe Management and Interaction The recipe management system forms a core component of the application, offering multiple interaction paths. Users can search for recipes using various criteria, with the system querying the database to return relevant results. When viewing recipes, the system fetches comprehensive information including ingredients, instructions, and nutritional data. Users can save favorite recipes, which updates their preferences in the database and creates a permanent relationship between the user and the recipe. The rating system allows users to provide feedback, automatically updating the recipe's average rating and maintaining a history of user interactions.

Advanced Image Analysis One of the system's most sophisticated features is its image analysis pipeline. Users can upload food images which undergo a multi-step analysis process. The YOLO detection system identifies food items in the image, followed by depth estimation to calculate portion sizes. The system then performs nutritional calculations based on the identified foods and their portions. This analysis culminates in a comprehensive report that is saved to the database and linked to the user's account, providing valuable nutritional insights.

Intelligent Chat Integration The chatbot system provides real-time assistance through natural language processing. When users submit queries, the system processes them and interfaces with the OpenAI API to generate contextually relevant responses. These responses are then formatted appropriately and displayed to the user, creating an interactive and helpful experience. The chat system maintains conversation context and can provide recipe-specific guidance when needed.

Administrative Control and Management For users with administrative privileges, the system provides additional management capabilities through a dedicated admin panel. This includes comprehensive user management tools, recipe moderation capabilities, and access to system reports and analytics. The admin interface ensures proper content management and system maintenance while maintaining security through strict access controls.

Database Operations and Data Flow Throughout all these processes, the system maintains efficient database operations for both storage and retrieval. User information, recipe details, interaction histories, and analysis reports are stored in structured formats for quick access. The database architecture supports fast searches and efficient data retrieval while maintaining data integrity and relationships between different system components.

System Components and Architecture The flowchart uses a color-coded system to clearly differentiate between various types of operations. Process steps are marked in blue, representing active operations and business logic execution. Decision points are highlighted in red, showing where the system makes critical choices about process flow. Database operations are indicated in green, demonstrating data storage and retrieval points. Input/output operations are marked in purple, representing user interaction points.

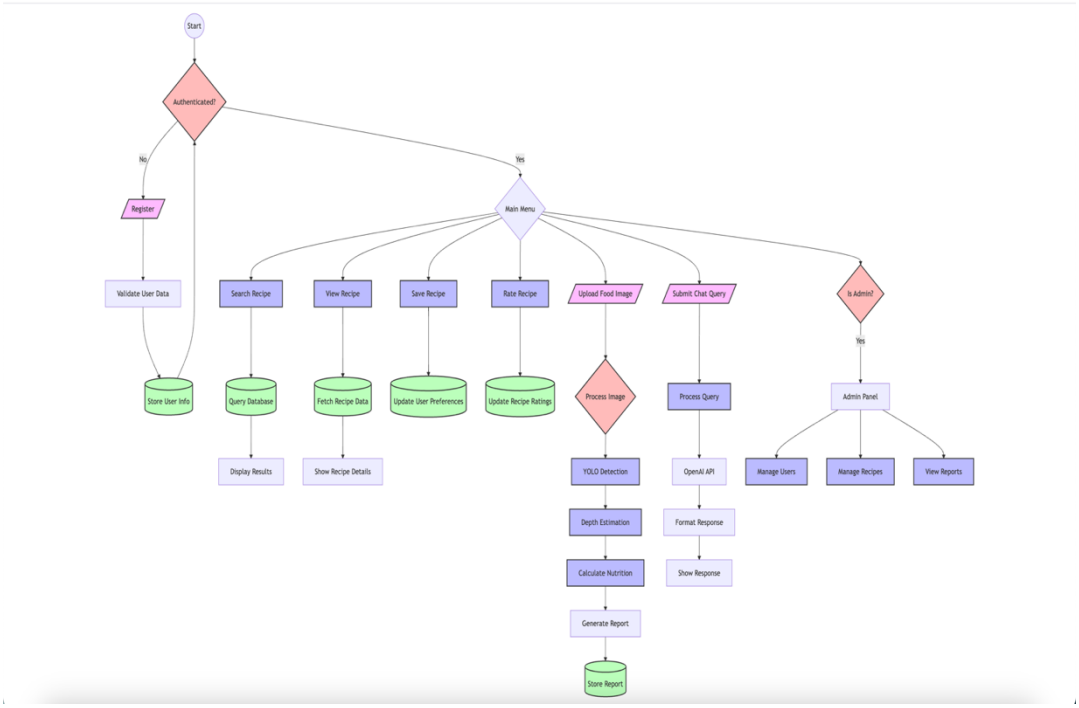

Figure 7: System Flowchart

Chapter 4
Implementation Results

4.1 Programming Languages and Tools

4.1.1 Frontend (Mobile Application)

- **Flutter:**
    - **Language:** Dart
    - **Description:** Flutter is a UI toolkit from Google for crafting natively compiled applications for mobile, web, and desktop from a single codebase. It's particularly useful for rapid UI development with pre-designed widgets and its ability to maintain native performance.
    - **Justification:** Flutter allows for the development of a single codebase for multiple platforms, reducing development time and effort. Its rich set of pre-designed widgets and native performance are ideal for creating a smooth user experience.

4.1.2 Backend (Server-side)

- **Django:**
    - **Language:** Python
    - **Description:** Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. It’s particularly good for developing complex, data-driven websites quickly.
    - **Justification:** Django's rapid development capabilities and clean design make it suitable for developing the backend of a complex mobile application like AKLNY.

4.1.3 Database

- **SQLite:**
    - **Description:** SQLite is a lightweight, self-contained SQL database engine that is quick to set up and easy to use. It's ideal for mobile applications due to its small footprint and compatibility with mobile devices.
    - **Justification:** SQLite is a perfect choice for mobile applications as it requires minimal configuration and is well-suited for handling the data storage needs of AKLNY.

4.1.4 Machine Learning

- **1.PyTorch**

  - **Language:** Python

  - **Description:** PyTorch is an open-source machine learning framework widely used for developing and deploying deep learning models. It provides a dynamic computation graph, which makes debugging and experimentation easier. PyTorch supports a vast library of pre-trained models, including those used for computer vision tasks like object detection and depth estimation.

  - **Justification:** PyTorch is used to integrate the YOLO segmentation model and MiDaS depth estimation model. These tools enable precise object segmentation and volume calculation in the Akelny platform, providing users with actionable insights into food portions.

- **2. OpenCV**

  - **Language:** C++ and Python

  - **Description:** OpenCV (Open Source Computer Vision Library) is a powerful library for real-time computer vision tasks. It offers a wide range of tools for image and video processing, including object tracking, image transformation, and feature detection.

  - **Justification:** OpenCV is used in the Akelny backend for preprocessing input images, resizing masks, and generating overlays for visualizations. Its efficient image manipulation capabilities make it essential for seamless integration with machine learning models.

- **4. YOLO (You Only Look Once)**
    - **Language:** Python (via `ultralytics` package)
    - **Description:** YOLO is a state-of-the-art object detection model that processes images in real time. It detects multiple objects in a single pass, providing segmentation masks and confidence scores with high accuracy.
    - **Justification:** YOLO is employed in Akelny for object detection and segmentation of user-uploaded images. Its efficiency and speed make it ideal for identifying food items and generating the data required for volume estimation.
- **5. MiDaS (Monocular Depth Estimation)**
    - **Language:** Python (via PyTorch and Intel-ISL repository)
    - **Description:** MiDaS is a depth estimation model that predicts depth maps from a single input image. It is trained on diverse datasets to generalize well across various scenarios, including object segmentation tasks.
    - **Justification:** MiDaS is used in Akelny to generate depth maps for segmented objects. These depth maps are essential for calculating the volume of food items, enabling precise weight and nutritional analysis.

- **6. Hugging Face Transformers**

  - **Language:** Python

  - **Description:** Hugging Face Transformers is a library that provides pre-trained models for natural language processing tasks such as text generation, translation, and question answering. It includes state-of-the-art models like GPT and BERT.

  - **Justification:** Hugging Face Transformers power the chatbot functionality in Akelny. By leveraging GPT models, the chatbot can handle conversational queries related to recipes, ingredients, and nutritional advice with contextual intelligence.

- **7. OpenAI GPT**

  - **Language:** Python

  - **Description:** OpenAI GPT (Generative Pre-trained Transformer) is a state-of-the-art language model designed for generating human-like text. It can process complex queries and generate coherent, context-aware responses.

  - **Justification:** The chatbot in Akelny relies on OpenAI's GPT model to provide advanced conversational capabilities. This tool enhances user interaction by offering accurate and helpful responses to culinary-related questions.

- **8. Torchvision**
  - **Language:** Python
  - **Description:** Torchvision is a library that provides image processing tools and pre-trained models for tasks such as object detection, segmentation, and classification. It integrates seamlessly with PyTorch for end-to-end deep learning workflows.
  - **Justification:** Torchvision supports data transformations and pre-trained models, making it an essential tool for preparing input images and enhancing the YOLO and MiDaS workflows in Akelny.
- **9. PIL (Python Imaging Library) / Pillow**
  - **Language:** Python
  - **Description:** PIL (now maintained as Pillow) is a library for image processing in Python. It provides tools for opening, manipulating, and saving various image file formats.
  - **Justification:** Pillow is used in Akelny to preprocess input images, including resizing, cropping, and converting formats. This ensures that the images are ready for analysis by the YOLO and MiDaS models.

4.1.5 API Development

- **Django REST Framework (DRF)**

  - **Language:** Python

  - **Description:** Django REST Framework is a powerful and flexible toolkit for building Web APIs. It provides tools for easily serializing data, handling authentication, and creating endpoints for CRUD operations. It integrates seamlessly with the Django framework and supports advanced features like pagination, filtering, and permissions.

- **Justification:** Django REST Framework was chosen for the Akelny Backend because of its robust capabilities in building RESTful APIs. Its integration with Django simplifies the development process, while its built-in features like authentication, serialization, and viewsets enable rapid and efficient API development. DRF's flexibility allows customization to cater to specific application needs, such as managing recipes, user profiles, and interactions with machinelearning models.

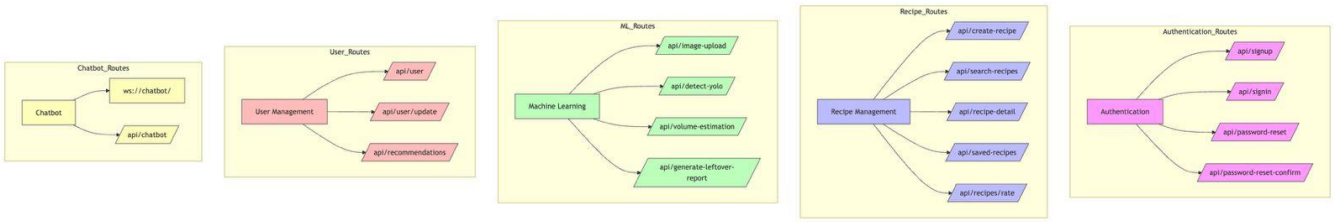

Figure 8: API Endpoints Diagram

4.2 Code Structure

4.2.1 Frontend (Mobile Application)

The food-related mobile application is developed using Flutter, featuring a dynamic and intuitive interface for discovering recipes, managing leftovers, interacting with a chatbot, and editing user profiles. The project structure is organized into a lib/ directory containing core functionalities and UI screens. Within this directory, the core/ folder includes essential services like AuthService for API integration, while screens/ houses the user interface components. Dependencies managed via pubspec.yaml include Flutter SDK for the framework, Google Fonts for typography, Shared Preferences for authentication token storage, HTTP for API communication, Camera, Image Picker, and Image Cropper for handling images, Logger for debugging, and WebSocketChannel for real-time communication.

The app's navigation begins with the SplashScreen, which checks the user's authentication status, directing them to either the SignInScreen or MainNavigationPage. The SignInScreen supports secure email/username and password authentication while also offering guest login and navigation to the SignUpScreen. The SignUpScreen collects details such as name, email, and preferences, with features like a date picker for enhanced usability. Both screens prioritize error handling by validating input fields and managing incorrect credentials. Upon authentication, users are directed to the MainNavigationPage, which acts as the central hub. This page features a bottom navigation bar with tabs for Home, Saved Recipes, Leftovers, Recommended Recipes, and Profile, ensuring state persistence with an IndexedStack for a seamless user experience.

The HomeScreen allows users to explore recipes through a search bar, category filters, and a card-based display. From here, users can access the RecipeDetailsScreen,

where they view comprehensive recipe information, including nutritional details, options to save and rate recipes, and interactive buttons. Saved recipes are accessible on the SavedScreen, dynamically fetched from the backend and designed to handle errors like empty lists gracefully. The RecommendedRecipes screen displays personalized recipe suggestions, fetched dynamically based on backend data. The LeftoverReportScreen allows users to upload images of leftover food, which are analyzed by the backend to generate nutritional reports. This feature integrates an image picker, upload functionality, and report display, offering both practicality and interactivity.

The UserProfilePage displays user details, offering a button to navigate to the EditProfileScreen, where users can update their personal information. The EditProfileScreen validates inputs and ensures smooth error handling, enhancing user convenience. The ChatbotScreen, built with WebSocket technology, facilitates real-time chatbot communication, with future enhancements planned for features like conversation history and personalized interactions. Consistent styling is applied across the app, with cohesive colors and typography ensuring a polished interface.

The architecture of the app can be visualized through two diagrams. The first, a **Screen Navigation Flow**, illustrates user navigation across screens, including authenticated and unauthenticated routes, and highlights the connections between key features such as Home, Saved Recipes, and Profile. The second diagram, **Backend Interactions**, demonstrates the integration of the frontend with the backend through the AuthService, showcasing methods like signUp, signIn, getUserData, updateUserData, and others. These methods enable the app's dynamic features, such as fetching recipes, managing user profiles, and delivering real-time data updates.

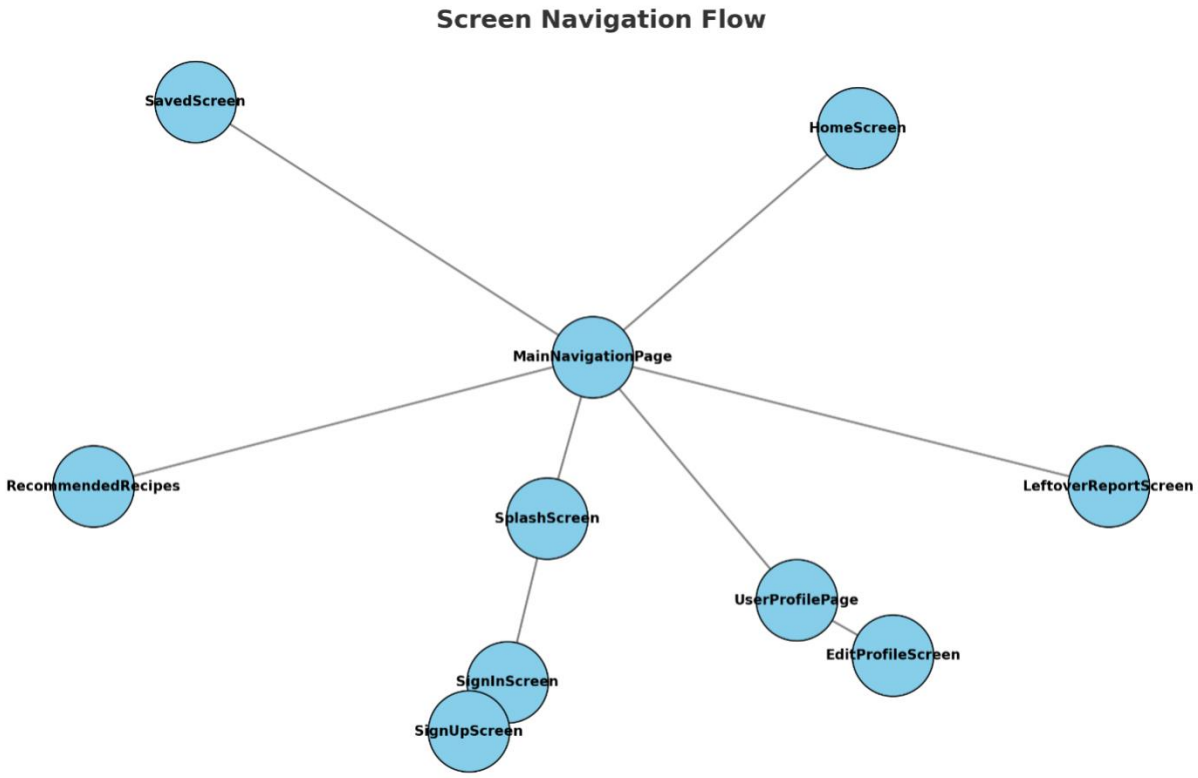

Figure 9: Screen Navigation Flow Diagram

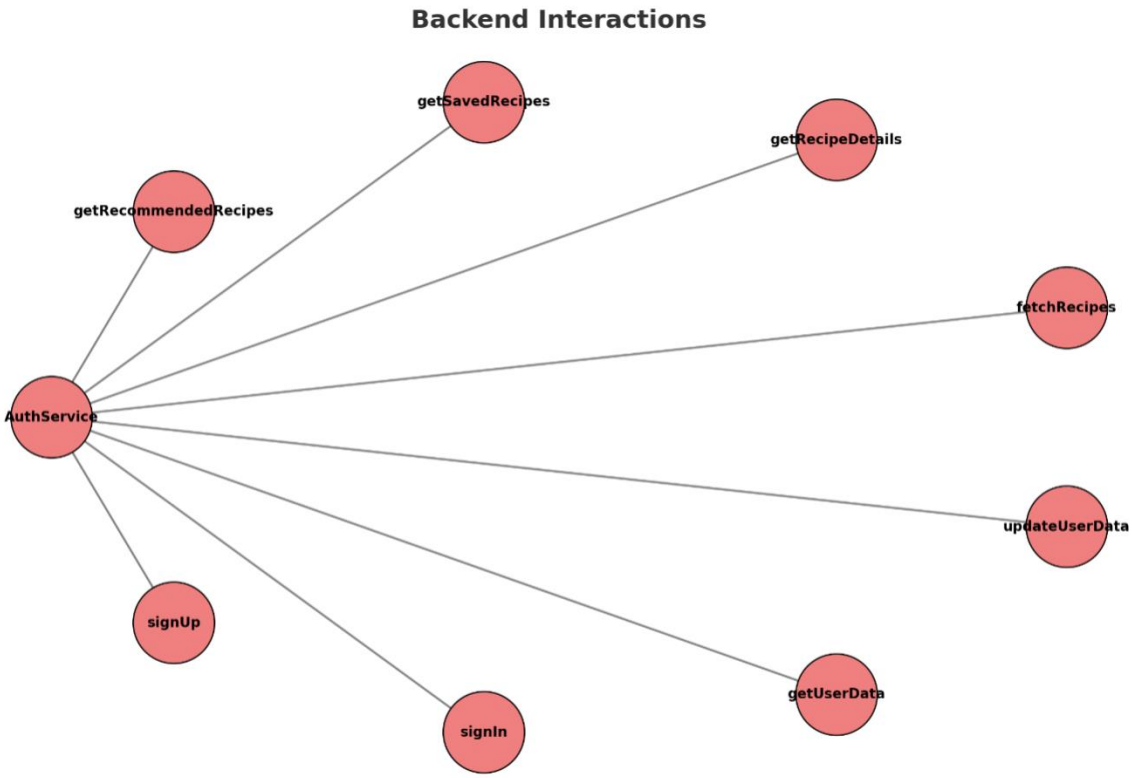

Figure 10: Backend Interactions Diagram

The app adheres to strict coding conventions and best practices. Variables and methods follow CamelCase naming, while class names use PascalCase. State management is achieved using setState for local updates and FutureBuilder for asynchronous tasks. Error handling employs try-catch blocks for API calls, with user-friendly error messages displayed through SnackBar notifications. To ensure user satisfaction, enhancements such as advanced recipe filters by preparation time and difficulty, dark mode, social login through Google or Facebook, push notifications, and offline support for cached recipes and profile data are planned for future iterations. These features aim to create a comprehensive and user-friendly application for managing culinary experiences efficiently and effectively.

4.2.2 Backend (Server-side)

The backend encompasses a wide range of functions such as logging in, managing users and recipes, image recognition services, and recommending recipes. This backend has been built with an intention of providing a 'lively' and interactive user base which was also future proof. Using recent technologies such as Django and the latest machine learning models, it enables the delivery of outstanding security and performance across users and administrators. In terms of how users engage with recipes and dietary data, this document demonstrates how the Akelny Backend possesses the capability to create a paradigm shift and as such goes on to discuss the features, architecture, and underlying principles in detail of the backend's operation.

Reasons For Choosing Django

Django was chosen as the framework for the Akelny backend as it subscribes to the philosophy of 'batteries-included'. This philosophy saves lengthy development time, as Django offers a complete tool set as part of the package. Thanks to Django's focus on sustained rapid application development practices, creation of secure and scalable applications becomes less tedious and time consuming. As a result, applications can be built in no time considering built-in modules of user authentication, database interaction and admin interfaces. These functionalities help remove the need to develop these elements from the ground up. WSGI and ASGI standards, are fully supported by the Django framework which greatly aids in building scalable applications. The implementation of these standards allows synchronous and asynchronous operations to be used interchangeably which is a requisite for real-time interaction like a WebSocket based chatbot. Every application is normally expected to deal with a variety of user-sensitive

information, and this explains why Django implements security features like SQL injections and cross site scripting (XSS) protection, and CSRF attacks mitigation. In addition, it is more than sufficient for the application in question. The custom user models can also store additional fields like dietary restrictions, allergies, and cooking skills to name but a few. This allows Akelny to meet those requirements particularly well. Many thanks to the abundant ecosystem of third-party packages and the active community's support along with the ready-made administration panel, Django remains the best candidate for constructing a complex and yet easy to manage backend such as Akelny.

Tech Stack

The Akelny Backend is made possible by employing a tech stack that is optimally configured for performance, extensibility, and dependability. Incorporating Python, a well known and flexible programming language alongside Django which is a popular web framework designed to accelerate development and provide easier clean designing. The backend uses DRF in building RESTful API and easing communication with front end clients. Real time capabilities are facilitated by the use of Django Channels which supports interactive features such as the chatbot via WebSocket connections. The database layer is set to use SQLite when in a development mode and can be easily changed to PostgreSQL or MySQL when in production mode. JSON web tokens which allow stateless approach to authenticate users by securely proving access to defined API endpoints is what is used to perform authentication. When working on machine learning tasks, the backend incorporates PyTorch which integrates image processing YOLO and MiDas models. For sending and preparing image open cv is utilized while SMTP integrations allow for emailing via providers such as Gmail, Yahoo, Outlook, iCloud and zoho. The entire

architecture has been fully adapted for the use of AWS, Heroku and Azure, such that it maintains high scalability and performance even when subjected to varying loads.

Architecture

The Akelny Backend is built using modular yet scalable architecture, comprising of independent modules that cohesively join together. The architecture further ensures the modules are of high cohesion and possesses a low coupling, in turn, making it scalable and maintainable. The major modules are as follows:

Accounts Module

As the name suggests, this module deals with user-related operations including login and authentication, profile modification, or resetting passwords. The model provided by Django is now a custom one, in which besides containing an additional set of fields like dietary restrictions, allergies, phone numbers and cooking proficiency, it also overrides the default user implementation. Secure and scalable token-based authentication is done with JWT authentication.

Recipe Management

The recipe management module is concerned with the said functions. Recipes are treated as data-entities with features, categories, ingredients, nutritional information, density, and more attached to them. The extensive metadata associated with the recipes enable various complex functionalities such as automated recipe matching and nutritional composition analysis. This module works together with ML models to improve the experience of the users.

Machine Learning Models

Incorporated into the Akelny Backend are two very suitable ML models enabling these advanced features. YOLO (You Only Look Once) – object detection and segmentation model for detection of food in images uploaded by the user. MiDas is a depth model, essential for estimating food volume or food weight. All these models work together and yield such information, which then further enables nutrition analysis as well as allows users to upload images.

Integration of the Chatbot

Incorporation of Open AI's GPT models into a Chatbot has presented users with more engaging and conversational means of interacting with the chatbot.Additionally, the chatbot can assist users with questions about recipes, ingredients, techniques of cooking and nutrition which ensures a more satisfying experience that meets the user's needs.

System Recommendation

The hybrid recommendation system utilizes both content based filtration and collaborative filtering which allows for a more personalized suggestion for recipes. Content based filtering identifies user's dietary goals and cuisines they tend to prefer, on the other hand, collaborative filtering makes use of interactions of users such as saved recipes in order to recommend what most other users in a particular category would like. This dual system works to ensure that the suggestions made to the user are always relevant and meets their specific requirements.

Image Uploading and Processing

This module allows users to take pictures of a particular food item and upload it for AI to examine it using pre-built machine learning models. YOLO is used for the picture recognition while MiDaS is used for estimating volume and weight to further enhance Recognition During the writing process, the system integrates the recognized items into the complex structure in order, making it the first recipe by nutrition.

Features

The set of functions of the Akelny Backend is quite diverse, therefore it is able to meet the needs of both users and administrators. The engagement system employs user verification and profile editing through registration, logging in, updating a user's profile, and resetting a password. Also, there is a recipe management system allowing users to create, edit, delete, and search any basic recipe with available filters for keywords, categories or ratings. Also, an AI powered chatbot can give verbal assistance helping to answer questions about any kind of recipes, cooking processes, or ingredients in detail. The

image processing module is a part of a single application that employs deep learning algorithms such as convolutional neural networks to analyze uploaded images of various food and determine their caloric content. The algorithm predicts the suitability of a user with recipes based on past interactions and preferences ensuring maximum personalization of the recommendations. It is possible to carry out all the administrative tasks from the admin panel in order to manage users, recipes and interactions completed with a large number of filters and actions.

Workflow

The Akelny Backend works through different workflows that are explicit and promote efficiency as well as reliability. To register, users access a specified endpoint which is the /signup/ by firstly providing an email, password, and other profile information, then they are awarded a JWT token which they are to use to access the system securely. The /signin/ endpoint is utilized for authentication and issues a token that shall be used in all other subsequent requests. Users are able to change their existing passwords through /password-reset/ and /password-reset-confirm/ endpoints. Part of the recipe management workflow is well structured in such a way that users upload recipes with rich metadata on /create-recipe/, users are able to search the attached recipes by going to /search-recipes/, while uploaded recipes can be managed with the use of /saved-recipes/. The image upload and processing workflow begins with going to /image-upload/, which starts by allowing users to upload images of food. YOLO and MiDaS models are deployed for volume and weight estimation after the images have been sourced and the nutritional values obtained. Recommendations are automatically generated from analysis of the content provided by the users and historical data of user interaction and other factors such as the content of the

pages and collaborative filtering for the targeted groups to enhance better and faster reach of the intended users in recommendations /recommendations/.

Machine Learning Integration

The backend of Akelny is built on the architecture of machine learning integration which allows it to incorporate unique capabilities that are not found in traditional systems. It also employs YOLO algorithm for the segmentation of food items in a given image with high confidence scores, generating masks with high accuracy and speed. Moreover, MiDas supplies the volume of segmented food items which is vital for calculating weight and nutrition volume. With YOLO and MiDas together, the system is capable of performing dietary calculations accurately by searching in the database for recipes of items and nutritional content according to the weight of the food. And this integration makes it possible to greatly enhance the user experience for the system as well as to significantly increase the value of the system by allowing for applicable dietary recommendations.

4.2.3 **Machine Learning Models**

The involvement of machine learning models in the Akelny backend is quite important as it assists the user in performing object segmentation and upload image volume estimates. The models provide forward looking analytics for food items by determining the volume and weight that is used to compute nutrition values. The process makes use of two very high-quality models which are YOLO for object detection and segmentation and MiDas for estimating depth. These models sequentially perform various tasks such as recognizing an object, calculating its volume and depth, trace its shape and show the result of such tasks. This feature increases the interactivity of users with the system as the precise content of the food items is revealed.

The workflow begins by loading the YOLO segmentation model and MiDaS depth estimation model. YOLO is a highly efficient object detection model that segments the input image, identifying objects and creating corresponding masks. Each mask is associated with a label (e.g., "chicken" or "lemon") and a confidence score that indicates the accuracy of the detection. Once the segmentation is complete, the MiDaS model generates depth maps for the identified regions. These depth maps provide a pixel-by-pixel understanding of object depth, allowing the system to calculate spatial dimensions accurately.

To ensure accuracy in volume estimation, the system uses camera parameters such as sensor dimensions, resolution, focal length, and the distance to the object. These constants help translate pixel-based measurements into real-world dimensions. By combining the area of the segmented mask and the depth map for each object, the system calculates the volume using a straightforward mathematical formula. The final results

include the object label, area in square centimeters (cm^2), average depth in centimeters (cm), and the volume in cubic centimeters (cm^3). For example, in the test image provided, the segmentation identifies "chicken" with a confidence score of 0.78 and calculates its volume as 160.42 cm^3, while the "lemon" is detected with a confidence score of 0.93 and a volume of 12.68 cm^3.

The system also visualizes these results, making it easier to understand the workflow outputs. The visualization displays three panels: the original image, the segmented image with confidence scores and volume calculations, and the depth map. The original image represents the raw input provided by the user. The segmented image highlights detected objects with green overlays, labels, confidence scores, and their estimated volumes. The depth map offers a gradient view of object positions, where darker regions represent objects closer to the camera and lighter regions depict those farther away. These visualizations make the workflow intuitive and help stakeholders and users grasp the model's functionality more effectively.

The entire workflow is orchestrated by a Python script with well-defined functions for each stage. The load_yolo_model function initializes the YOLO model using pre-trained weights. The segment_image function applies the model to the input image to detect objects and generate segmentation masks. Depth estimation is handled by the estimate_depth function, which uses the MiDaS model to create depth maps for each segmented region. The calculate_volume function integrates the mask, depth map, and scaling factors to compute the object's volume. Finally, the visualize_results_with_confidence function overlays the results on the original image and produces an easy-to-understand visual representation.

For example, to run the workflow, you would define the paths to the YOLO model weights and the input image, then load the MiDaS depth estimation model. By calling the main function with these inputs, the script performs segmentation, depth estimation, volume calculation, and visualization in sequence. The output includes printed results in the console and a visualization that showcases the workflow's accuracy and efficiency.

To integrate this into the Akelny platform, the script relies on a variety of Python libraries, including ultralytics for YOLO, torch for MiDaS, and additional tools like OpenCV, NumPy, and Matplotlib. These libraries ensure efficient image processing, accurate computations, and intuitive visualizations. Dependencies can be installed via `pip`, and the system is designed to be adaptable, allowing modifications to confidence thresholds, camera parameters, or models as needed.

The image below illustrates the results of this workflow. The original image is shown on the left, highlighting the input provided by the user. The middle panel demonstrates the segmentation results, with objects like "chicken" and "lemon" overlaid with masks, labels, confidence scores, and estimated volumes. The right panel visualizes the depth map, showing the spatial relationship of the objects in the scene. This visual representation provides a comprehensive overview of how the machine learning models process the input data to deliver actionable insights.

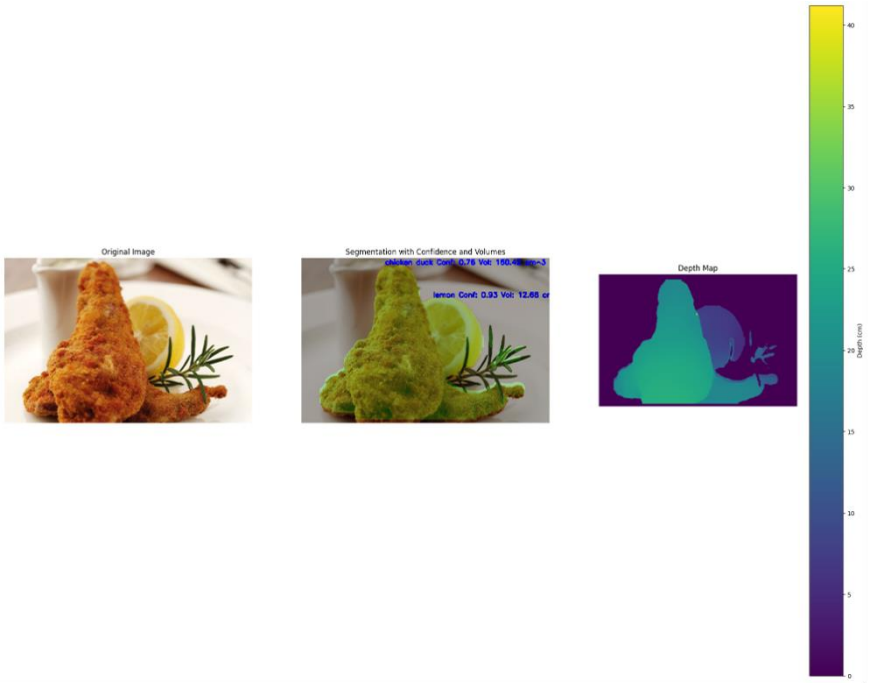

Figure 11: Segmentation

- **Left:** Original image uploaded by the user.
- **Middle:** Segmented image with confidence scores and volume estimations (e.g., Chicken: 160.42 cm^3, Lemon: 12.68 cm^3).
- **Right:** Depth map indicating object distances, where darker regions are closer to the camera, and lighter regions are farther away.

4.2.4 Akelny chatbot

Model Architecture and Configuration

The Akelny chatbot is implemented using OpenAI's GPT-4 model, integrated with Django's WebSocket functionality for real-time communication. The system utilizes the OpenAI API with a configuration focused on providing culinary expertise and recipe-related assistance. The architecture employs a temperature setting of 0.7 to balance creativity and accuracy in responses. The model is initialized with a system prompt that

defines its role as a virtual chef assistant, enabling it to handle queries about recipes, ingredients, cooking methods, and nutritional information.

Data Processing and Storage

The chatbot maintains conversation history through a structured approach, storing interactions in a thread-like format. Each conversation includes system messages, user queries, and assistant responses. The system implements a JSON-based data structure for managing conversation flow and context retention. This allows the chatbot to maintain coherent conversations and provide contextually relevant responses across multiple interactions.

Question-Answering Mechanism

The question-answering system is built around the process_user_query function, which handles incoming queries and generates appropriate responses. This mechanism includes error handling and logging capabilities to ensure system reliability. The system processes user input through the OpenAI ChatCompletion API, maintaining conversation context through message history. This enables the chatbot to provide responses that are not only accurate but also contextually appropriate to the ongoing conversation.

Implementation Workflow

The implementation follows a clear workflow beginning with WebSocket connection establishment through the ChatbotConsumer class. When a connection is

initiated, the system creates a new conversation thread and begins processing messages. The workflow includes:

1. Connection establishment
2. Message reception and parsing
3. Query processing through OpenAI's API
4. Response generation and transmission
5. Error handling and recovery

.

API Integration

The chatbot exposes two primary interfaces: a WebSocket endpoint at 'ws/chatbot/' and a REST endpoint at 'api/chatbot/'. The WebSocket implementation enables real-time communication, while the REST endpoint provides traditional HTTP-based interaction. Both interfaces accept JSON-formatted messages and return structured responses. The API integration includes comprehensive error handling and logging mechanisms to ensure reliable operation.

Front-End Interaction

The system implements a WebSocket consumer class (ChatbotConsumer) that handles real-time communication with front-end clients. This consumer manages connection lifecycle events, message processing, and response transmission. The implementation includes proper error handling and connection state management to ensure robust communication with client applications.

Performance Optimization

Performance optimization is achieved through efficient message processing and connection handling. The system implements asynchronous processing where appropriate and includes error recovery mechanisms. The chatbot maintains conversation context while optimizing resource usage, ensuring responsive performance even under load.

Code Structure

The codebase is organized into several key components:

- chatbot_assistant.py: Contains core assistant creation and management logic
- chatbot_logic.py: Implements query processing and response generation
- consumers.py: Handles WebSocket communication
- routing.py: Defines WebSocket routing configuration
- views.py: Implements REST API endpoints

Each component is designed to be modular and maintainable, with clear separation of concerns and comprehensive error handling.

.

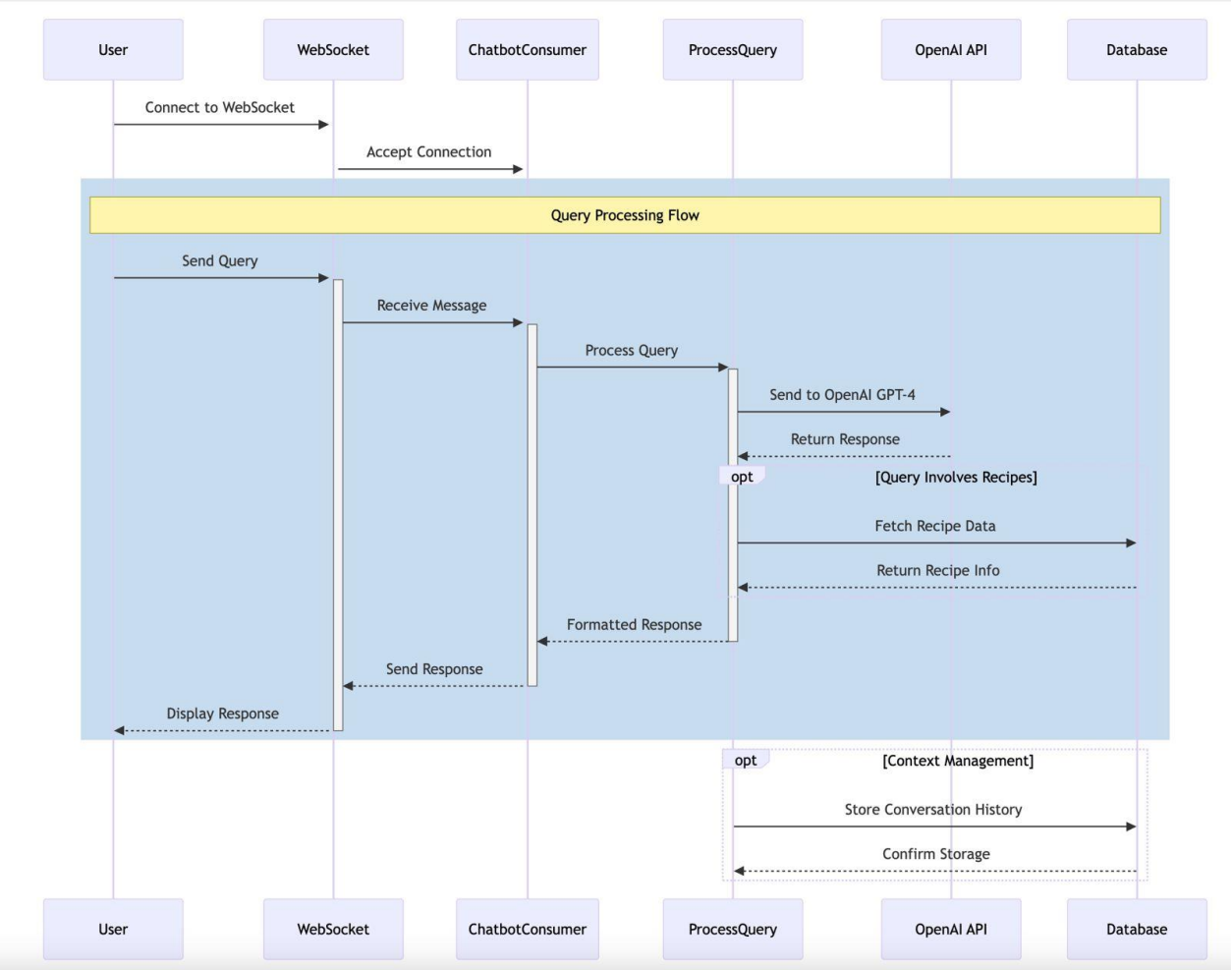

Figure 12: Chatbot Interaction Flow Diagram

4.3 Data Structures and Databases

Akelny's backend operates on optimization of functionality, through meticulously designed data structures and database schemas. These mechanisms are aimed at the efficient management of recipes, user data, and interaction logs. In this part, we deep dive into the database schema, the data structures, and the storage and retrieval mechanisms for the diabetes dietary application.

4.3.1 **Data Structures**

The backend of Akelny combines various Python-related programming as well as database applications to facilitate relational as well as operational aspects of the features of the platform. Central to the project is the use of the Django web framework which implements ORM to let the developers model data using python without the worry of flooding the business logic with complex models. Some of the fundamental data models are as follows:

User Model: The structure of user data in this application is derived from the standard AbstractUser model of Django but has been modified to incorporate things such as dietary preferences, allergies, data that can be cooking skill level and profile pictures. There is a seamless approach in catering to user specific requirements such as recommendation and their preferences by extending this structure.

Recipe model: Recipes are kept in a form of object that contains essential components which are name , category, nutritional information such as calories, protein, fat, carbohydrate, volume and directions on how to cook the specific food. The model also comprises relations with users such as who created it and interaction logs left for user ratings or saves on the recipes.

User interaction logs: Logs from users preceding an action like to save a recipe or submit a rating are captured using data structures that relate users to recipes mentioning the type of action made and the time it was executed.

Machine Learning Outputs: Data structures are utilized for the storage of the outputs that have been retrieved through the YOLO and MiDaS models including segmentation masks, depth maps, and even parameters such as food volume and weight. These outputs go a long way to help in enhancing the data that is already augmented with the recipe information to enable generating more personalized and informative recommendations.

4.3.2 **Database Schema**

The structure of the Akelny back-end's database system has been carefully designed to model the relationships and dependencies among the various entities in the system. The structure conforms to normal forms as much as but still remains simple and efficient. Admin components for this schema includes:

User Data: Contains user data such as user names, emails, passwords, and designing attributes such as dietary habits, allergies, and cooking ability. The custom schema allows considering authentication and personalization functionalities.

Recipe details: Includes recipe details like recipe names with ingredients and a short description of the dishes' photos, categories, and nutrition. Users (created by, for instance), as well as user activities, are associated with these elements using foreign keys.

Saved Recipes: A table that contains the recipes that a specific user has bookmarked. This table containing recipes also comes with foreign keys that connect users and those recipes along with the time when the recipe was saved.

Ratings: Contains the ratings given to recipes by users with the ability to qualify these ratings according to US, recipe, rating, and time stamp. This table can be used for content based and rating based recommendations for the recommender system.

Interaction Logs Table: This table serves to analyze engagement and inform recommendations based on a range of user interactions with the recipes including views, clicks or saves.

Image Uploads Table: Image uploads of a user are stored in this table for further analysis. It helps in combining the machine learning results and identifying the recipes already available. The table contains metadata and image file paths.

Machine Learning Outputs Table: This table includes the completed data containing the segmentation, depth images, and volume data. These images or the recipes are linked with the generated images.

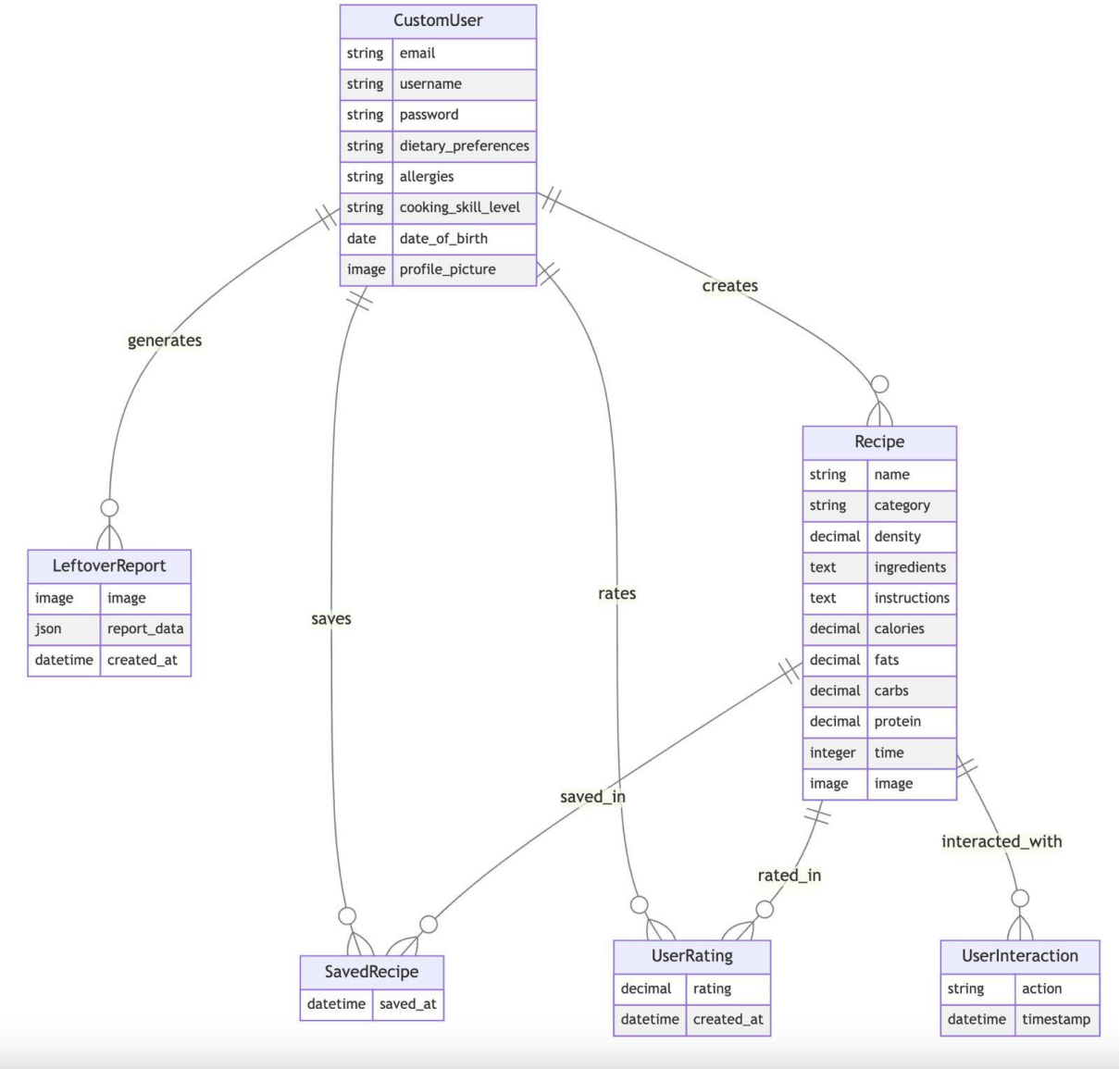

Figure 13: Relational Database Scheme

4.4 Results

The implementation of the Akelny application has successfully delivered a comprehensive food and recipe management system that integrates multiple advanced features into a cohesive user experience. The application begins with a streamlined authentication system, featuring a clean and intuitive login interface where users can access their accounts using email and password credentials. For new users, the registration process has been carefully designed to collect essential information while maintaining user engagement.

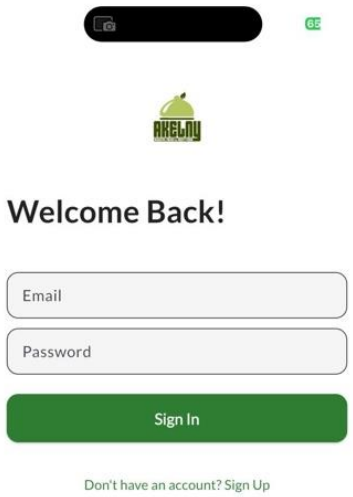

Figure 14: Akelny Login Interface

The sign up interface captures not only basic credentials but also important dietary preferences and personal information, allowing for a more personalized experience from the start. Through this comprehensive registration process, users can input their dietary preferences, allergies, and cooking skill levels, which are essential for personalizing their experience within the application.

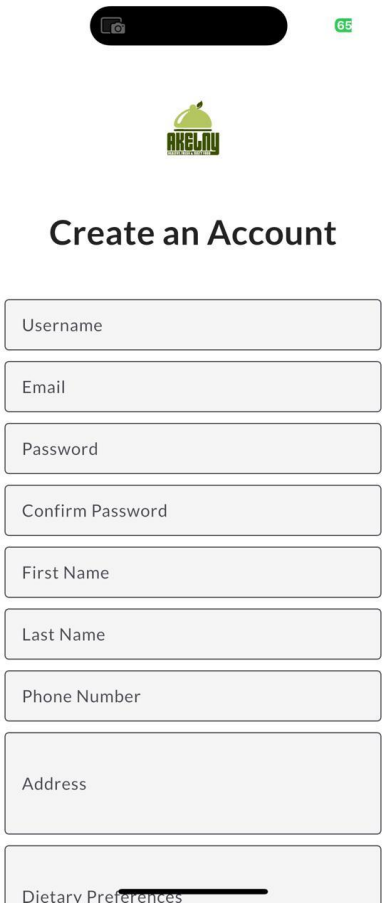

Figure 15: Akelny Signup Interface

The core functionality of recipe discovery and management has been effectively implemented through a well-organized main interface. Users can easily search for recipes using a prominent search bar, with results dynamically updating based on user input. The interface incorporates a sophisticated filtering system that allows users to browse recipes by categories and ratings, presenting each recipe in an visually appealing card format that displays key information such as cuisine type and user ratings.

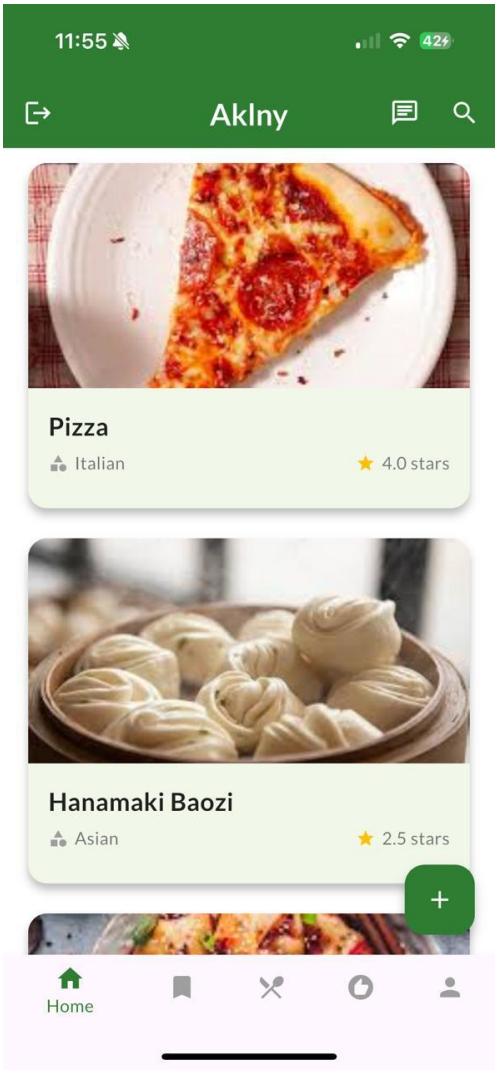

Figure 16: Akelny Homepage

A key feature of the application is the saved recipes section, which provides users with easy access to their favorite dishes. Each saved recipe is presented with clear visual representation, complete with cuisine categorization and rating information. This section demonstrates the application's ability to maintain user preferences and create a personalized cookbook experience.

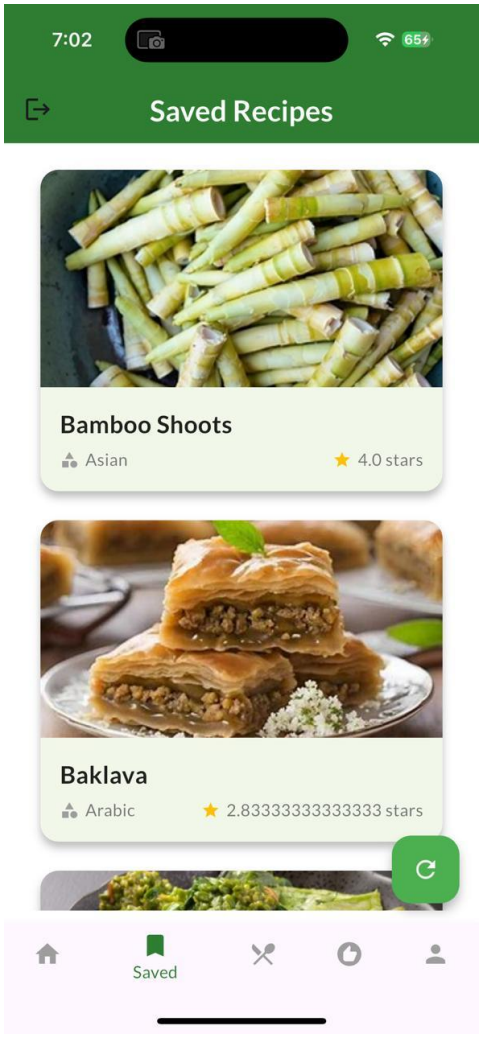

Figure 17: Saved Recipes

The user profile management system offers comprehensive control over personal information and preferences. Users can view and edit their personal details, dietary preferences, contact information, and cooking skill level, all organized in a clear and accessible layout. This feature ensures that users can maintain accurate profiles that inform the application's personalization features.

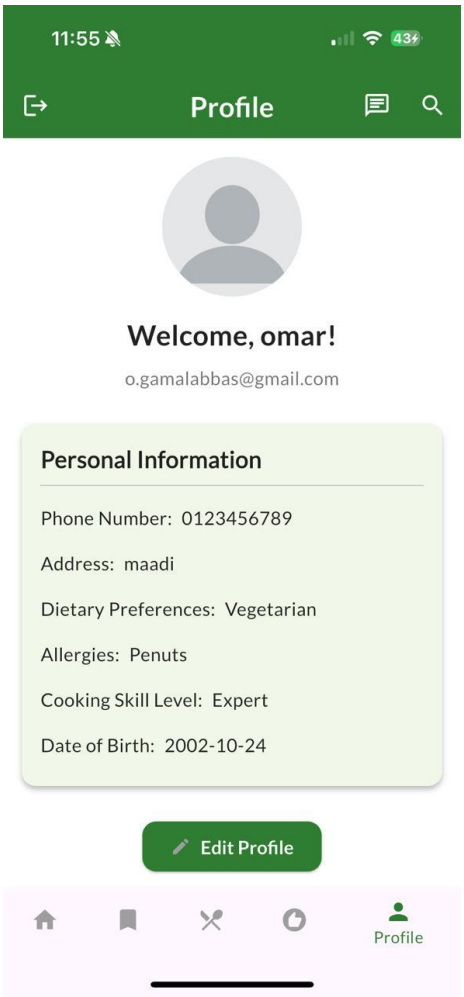

Figure 18: Profile Interface

The innovative leftover management system represents a significant technical achievement, incorporating advanced image recognition technology to analyze food items. Users can upload images of their leftovers and receive detailed nutritional analysis, weight estimations, and ingredient breakdowns, all presented in an easy-to-understand format.

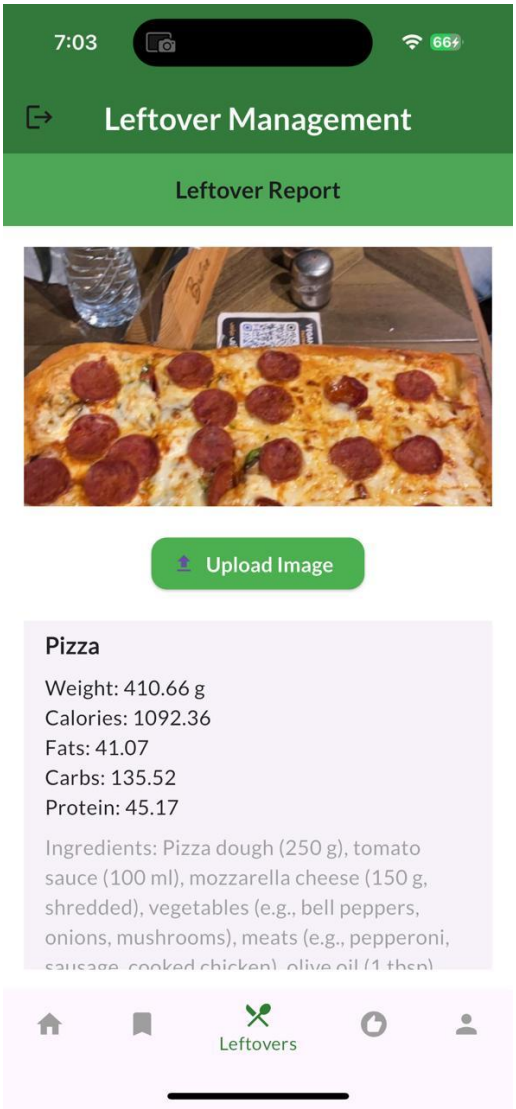

Figure 19: Leftover Management Interface

The AI-powered chatbot demonstrates the application's commitment to providing interactive and intelligent assistance. The chatbot interface handles natural language queries about recipes with impressive accuracy, providing detailed recipe instructions, comprehensive ingredient lists, and complete nutritional information. The step-by-step cooking guidance offers users a virtual cooking assistant that can answer questions and provide clarification at any point in the cooking process.

Figure 20: Chatbot Interface

The administrative interface provides comprehensive control over the application's content and functionality. Through this interface, administrators can manage:

- User accounts and authentication
- Recipe content and moderation
- Email configurations
- Image uploads and testing
- User ratings and saved recipes
- System themes and appearance
- Group permissions and authorization

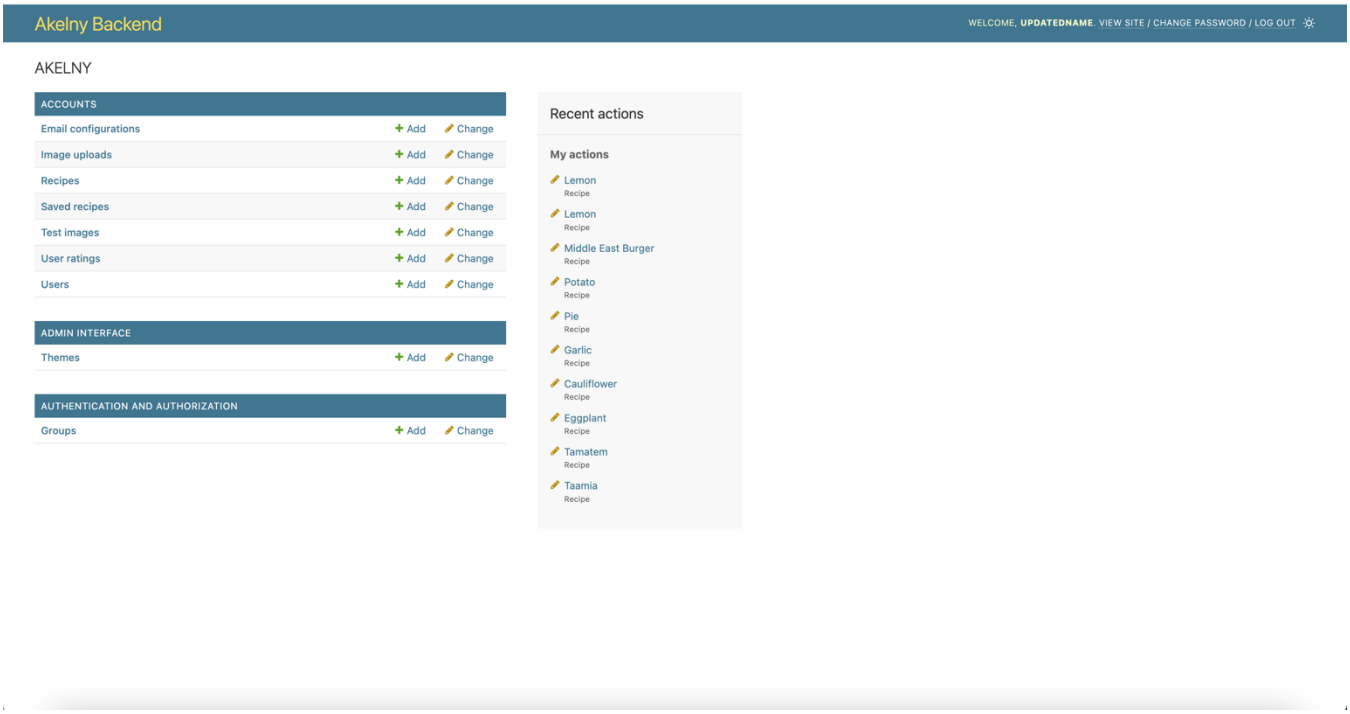

Figure 21: Admin Panel

Chapter 5
Discussion

5.1 Interpretation of Results

The implementation results of the Akelny system demonstrate significant alignment with modern cooking technology trends, evolving beyond traditional digital cookbook functionality. The system's authentication and personalization features show sophisticated user profiling capabilities, collecting crucial information about dietary preferences, allergies, and cooking expertise levels during registration. This data drives the personalized recipe recommendation engine, which successfully combines user preferences with behavioral data to suggest relevant recipes. The integration of YOLO-based image recognition technology for leftover analysis represents a significant technological advancement, providing users with precise nutritional information and portion estimations. The AI-powered chatbot's ability to provide contextual cooking guidance and recipe modifications demonstrates the system's commitment to interactive learning and real-time assistance. These features collectively create a comprehensive cooking platform that adapts to individual user needs while maintaining accessibility and user engagement.

5.2 Comparison with Previous Studies

When comparing Akelny with existing systems described in previous studies, several key advancements become apparent:

1. **Technology Integration**:

- While Garcia et al. (2016) focused primarily on basic recipe recommendations, Akelny implements a hybrid recommendation system that combines both content-based and collaborative filtering approaches
- Bangale et al. (2021) discussed theoretical frameworks for food recognition, but Akelny successfully implements practical YOLO-based food detection with volume estimation
- Previous studies often treated chatbots as separate entities, whereas Akelny integrates OpenAI-powered assistance directly into the cooking workflow

2. **User Experience**:

- Earlier systems typically offered either recipe management or nutritional tracking, but Akelny successfully combines both functionalities
- The system's comprehensive profile management and personalization features exceed the basic user preference systems discussed in previous literature
- Real-time interaction capabilities through WebSocket implementation provide immediate assistance, addressing limitations noted in earlier studies

3. **Feature Innovation**:

- The leftover management system introduces a novel approach to food waste reduction, moving beyond the simple inventory management systems discussed in previous research
- Integration of multiple email service providers and authentication methods shows advanced system flexibility compared to single-provider solutions in earlier studies
- The administrative interface provides more comprehensive control and monitoring capabilities than typically discussed in previous implementations

5.3 Limitations

1. **Technical Limitations**:

- The YOLO-based food detection system may face challenges with unusual or mixed food items
- Real-time image processing capabilities are constrained by current mobile device capabilities
- The chatbot's responses are limited by the training data available to the OpenAI model
- Database performance may be affected when handling large volumes of image data

2. **Implementation Constraints**:

- The current system requires stable internet connectivity for full functionality
- Mobile device storage limitations may affect the number of saved recipes and images

- The hybrid recommendation system requires substantial user interaction data for optimal performance
- Email service integration is limited to major providers, potentially excluding some users

3. **Research Limitations**:

- The evaluation relies heavily on controlled testing environments rather than real-world usage
- User feedback is limited to early adopters and may not represent the broader user base
- Performance metrics for the ML models may vary in different cultural and linguistic contexts
- Long-term effectiveness of the recommendation system requires extended usage data

4. **Market Considerations**:

- The system's feature set may be excessive for users seeking simpler recipe management
- Integration with local food suppliers and markets is not yet implemented
- Regional variations in cooking methods and ingredients may affect system accuracy
- Competition from established platforms may limit initial user adoption

Chapter 6
Conclusion and Future Work

6.1 Summary of Findings

The development and implementation of the Akelny application has successfully achieved its primary objectives of creating a comprehensive food management and recipe discovery platform. The system effectively integrates multiple sophisticated technologies, including YOLO-based image recognition for food analysis, depth estimation for portion sizing, and OpenAI-powered chatbot functionality for interactive cooking assistance. The user interface has proven to be intuitive and accessible, as demonstrated through the implemented authentication system, recipe management features, and personalization capabilities. A significant achievement is the leftover management system, which accurately analyzes food images to provide nutritional information and portion estimations. The hybrid recommendation system successfully combines content-based and collaborative filtering approaches to deliver personalized recipe suggestions. The administrative interface provides robust control over all system aspects, from user management to content moderation. Database operations have been optimized for performance, with proper implementation of relationships between models and efficient data retrieval methods. The WebSocket implementation for real-time chat functionality and the integration of multiple email service providers demonstrate the system's capability to handle both synchronous and asynchronous communications effectively. The application's security features, including JWT authentication and proper password handling, ensure user data protection while maintaining system accessibility.

6.2 Future Work

The Akelny system has several potential areas for enhancement and expansion in future iterations. The image recognition system could be extended to support real-time video analysis for dynamic cooking guidance and immediate feedback. Integration with smart kitchen appliances could provide automated cooking time and temperature adjustments based on recipe requirements. The recommendation system could be enhanced with machine learning capabilities to adapt to seasonal ingredients and local availability. Adding social features would allow users to share their cooking experiences, modifications to recipes, and cooking tips with the community. The chatbot functionality could be expanded to support multiple languages and regional cooking variations, making the application more accessible to a global audience. Implementation of augmented reality features could provide interactive cooking tutorials and ingredient measurement assistance. The leftover management system could be enhanced with meal planning capabilities that suggest recipes based on available ingredients to reduce food waste. Advanced analytics could be incorporated to provide insights into users' cooking patterns and nutritional habits. Integration with grocery delivery services could streamline the shopping process for recipe ingredients. Finally, the development of a progressive web application (PWA) version would improve accessibility across different devices and platforms while maintaining full functionality even with limited internet connectivity.

References

- Garcia, A. L., Reardon, R., & McDonald, M. (2016). Community interventions to improve cooking skills and their effects on confidence and eating behaviour. *Journal of Food Quality*, 5, 315-322. https://doi.org/10.1007/s13668-016-0185-3

- Bangale, S., Haspe, A., Khemani, B., & Malave, S. H. (2021). Recipe recommendation system using content-based filtering. Retrieved from https://ssrn.com/abstract=4102283

- DiFilippo, K. N., Huang, W. H., Andrade, J. E., & Chapman-Novakofski, K. M. (2015).

- The use of mobile apps to improve nutrition outcomes: A systematic literature review.

- *Journal of Telemedicine and Telecare*, 21(5), 243-253. https://doi.org/10.1177/1357633X15572203

- Kumar, I., & Smith, F. J. (2021). Opportunities of artificial intelligence and machine learning in the food industry. *Journal of Food Quality*. Retrieved from https://link.springer.com/article/10.1007/s12393-021-09236-1

- Chen, J., & Ngo, C. W. (2016). Deep-based ingredient recognition for cooking recipe retrieval. In *Proceedings of the 24th ACM international conference on Multimedia* (pp. 32-41). ACM. https://doi.org/10.1145/2964284.2964312

- Herranz, L., Min, W., & Jiang, S. (2018). Food recognition and recipe analysis: Integrating visual content, context and external knowledge. *arXiv preprint arXiv:1801.07239*. Retrieved from http://arxiv.org/abs/1801.07239

- Ueta, T., Iwakami, M., & Ito, T. (2011). Implementation of a goal-oriented recipe recommendation system providing nutrition information. In *2011 International*

- *Conference on Technologies and Applications of Artificial Intelligence* (pp. 183-188). IEEE. https://doi.org/10.1109/TAAI.2011.67

- Freyne, J., & Berkovsky, S. (2010). Intelligent food planning: personalized recipe recommendation. In *Proceedings of the 15th international conference on Intelligent user interfaces* (pp. 321-324). ACM. https://doi.org/10.1145/1719970.1720018

- Yang, L., Hsieh, C. K., Yang, H., Pollak, J. P., Dell, N., Belongie, S., Cole, C., & Estrin,

- D. (2017). Yum-me: A personalized nutrient-based meal recommender system. *ACM Transactions on Information Systems*, 36(1), 1-31. https://doi.org/10.1145/3001837

- Khan, M. A., Rushe, E., Smyth, B., & Coyle, D. (2019). Personalized, health-aware recipe recommendation: An ensemble topic modeling-based approach. *arXiv preprint arXiv:1908.00148*. Retrieved from http://arxiv.org/abs/1908.00148