Software Requirements Specification

for

QuickPick

Prepared by

Group ID: 20

Niyati Nagar	202412051
Pankaj Mainani	202412056

1 Introduction

QuickPick is a smart grocery shopping platform designed to simplify meal planning and grocery shopping for users. By integrating traditional grocery shopping with a recipe-based feature, it allows users to select recipes and automatically add required ingredients to their cart.

1.1 Document Purpose

The purpose of this Software Requirements Specification (SRS) document is to define the functional and non-functional requirements for the QuickPick platform. It serves as a blueprint for developers, testers, and stakeholders involved in the design, development, and deployment of the system. This document ensures that all parties have a clear understanding of the project's objectives, features, and constraints.

The SRS also acts as a reference point throughout the development lifecycle, enabling consistent communication and alignment among team members. It includes detailed descriptions of the user interfaces, functional requirements, use cases, and quality attributes, ensuring that the final product meets user expectations and industry standards.

1.2 Product Scope

QuickPick is an innovative online grocery shopping platform that enhances the user experience by integrating recipe-based shopping with traditional grocery purchasing. The platform enables users to browse products, manage shopping carts, and securely complete transactions. Its standout feature is the ability to select recipes and automatically add required ingredients to the cart, streamlining meal planning and grocery shopping.

Key benefits of QuickPick include:

- Simplified meal planning through recipe integration.
- Real-time inventory tracking to ensure product availability.
- Secure payment processing and robust user data protection.

The platform is designed to cater to busy individuals and families who value convenience, efficiency, and a personalized shopping experience.

1.3 Intended Audience and Document Overview

This document is intended for the following audiences:

- **Development Team**: To understand the system requirements and implement the software accordingly
- **Project Manager**: To plan project activities and allocate resources
- **Client/Stakeholders**: To validate that the system requirements align with business objectives
- **Quality Assurance Team**: To develop test plans and verify system compliance

- **Professor/Course Instructor**: To evaluate the project requirements and documentation

The document is organized into the following sections:

- **Introduction :** Provides an overview of the document's purpose, scope, and intended audience.
- **Overall Description :** Describes the product's context, functionality, design constraints, and assumptions.
- **Specific Requirements :** Details external interface requirements, functional requirements, and use case models.
- **Non-Functional Requirements :** Covers performance, safety, security, and software quality attributes.
- **Other Requirements :** Includes additional requirements such as database design and internationalization.

1.4 Definitions, Acronyms and Abbreviations

- **API :** Application Programming Interface
  A set of rules and protocols for building and interacting with software applications.
- **COMET :** Concurrent Object Modeling and Architectural Design Method
  A methodology used for software design that focuses on concurrent object modeling.
- **CSS :** Cascading Style Sheets
  A style sheet language used for describing the presentation of a document written in HTML or XML.
- **DB :** Database
  An organized collection of data, generally stored and accessed electronically.
- **GDPR** : General Data Protection Regulation
  A regulation in EU law on data protection and privacy for individuals within the European Union and the European Economic Area.
- **HTTPS** : Hypertext Transfer Protocol Secure
  An extension of HTTP used for secure communication over a computer network.
- **HTML :** Hypertext Markup Language
  The standard markup language for creating web pages and web applications.
- **IEEE :** Institute of Electrical and Electronics Engineers
  An international organization that develops global standards for various industries, including software engineering.
- **JWT :** JSON Web Token
  An open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object.
- **MongoDB :** MongoDB Database
  A source-available cross-platform document-oriented database program.
- **Node.js :** Node.js Runtime Environment
  An open-source, cross-platform JavaScript runtime environment that executes JavaScript code outside a web browser.

- **PCI DSS :** Payment Card Industry Data Security Standard
  A security standard for organizations that handle branded credit cards from major card schemes.
- **React.js :** React JavaScript Library
  A JavaScript library for building user interfaces, particularly single-page applications.
- **REST :** Representational State Transfer
  A software architectural style that defines a set of constraints to be used for creating web services.
- **SRS :** Software Requirements Specification
  A description of a software system to be developed, listing the system's features and functionalities.
- **UML :** Unified Modeling Language
  A general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.
- **WCAG :** Web Content Accessibility Guidelines
  A set of guidelines for improving web accessibility for people with disabilities.
- **U1-U6 :** Use Case Numbers
  Unique identifiers for use cases described in the SRS document.

1.5 Document Conventions

This document follows the IEEE Standard for Software Requirements Specifications (IEEE 830-1998). The formatting guidelines used are:

- Font Style: Arial
- Font Size: 12 for body text, 14 for section headings
- Spacing: Single-spaced with 1” margins
- Text Formatting: Italics for comments, bold for key terms
- Section Numbering: Follows IEEE format with hierarchical numbering (e.g., 1.1, 1.2, etc.)

1.6 References and Acknowledgments

- IEEE Software Requirements Specification Standard (IEEE 830-1998)
- React.js Documentation
- Node.js Documentation
- Express.js Documentation
- MongoDB Documentation

.

2 Overall Description

2.1 Product Overview

QuickPick is a comprehensive e-commerce solution designed to transform the online grocery shopping experience. It is a new, self-contained product that combines traditional grocery shopping with recipe-based shopping to create a unique user experience.

The system operates within the broader ecosystem of online food retail, payment processing, and delivery services. It interfaces with various external systems including payment gateways, and inventory management systems to provide a seamless experience to the end user.

2.2 Product Functionality

The QuickPick provides the following major functions:

- **User Account Management :** Users can securely register, login and also check their profiles.
- **Product Browsing and Search :** Users can browse different products with search functionality and filters.
- **Recipe-Based Shopping :** Users can browse different recipes with search functionality and filters.
- **Shopping Cart Management :** Users can check their cart, update and delete the items.
- **Payment Processing :** Process the payment and confirmation.
- **Inventory Management :** Stock details and management.

2.3 Design and Implementation Constraints

The development of the QuickPick is subject to the following constraints:

Technology Steck Constraints:

- Frontend must be developed using React.js with Redux for state management.
- Backend must be implemented using Node.js and Express.js
- MongoDB must be used as the primary database
- REST API architecture must be followed for client-server communication

Development Methodology:

- The COMET (Concurrent Object Modeling and Architectural Design Method) must be used for software design
- UML modeling language must be used for system design documentation
- Agile development methodology will be followed with two-week sprint cycles

Security Constraints:

- User authentication must implement JWT (JSON Web Token)
- Payment processing must comply with PCI DSS standards
- Personal data handling must comply with relevant data protection regulations

Third-party Integration Constraints:

- Payment gateway integration is limited to selected providers
- Image storage must utilize cloud storage solutions

2.4 Assumptions and Dependencies

Assumptions

- Users have basic knowledge of web/mobile applications and online shopping
- Users have access to internet connections with sufficient bandwidth to use the application
- Users have access to at least one supported payment method
- The delivery service will be available in the targeted geographical areas
- Recipe data can be mapped to actual grocery products in the inventory
- The system will initially support English language only, with plans for multilingual support in future versions
- Users will provide accurate delivery address information

Dependencies

- Availability and reliability of third-party payment gateway services
- Reliability of delivery partner services
- Availability of cloud hosting services
- Dependency on MongoDB Atlas for database management
- Reliance on third-party libraries and frameworks, including React, Node.js, and Express

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The QuickPick will provide a responsive web interface accessible through desktop and mobile browsers, with plans for dedicated mobile applications in future releases. The user interface will follow modern design principles with an emphasis on usability and accessibility.

Key UI components include:

- **Home Page**
  - Featured products carousel
  - Recipe highlights section
  - Promotional banners
  - Category navigation
  - Search bar

- **Product Listing Page**
  - Grid/list view toggle
  - Filter panel (price, brand)
  - Sort options
  - Pagination controls

- **Product Detail Page**
  - Product images gallery
  - Pricing information
  - Quantity selector
  - Add to cart button
  - Product description
  - Customer reviews section

- **Recipe Browse/Search Page**
  - Recipe cards with images
  - Quick view options
  - Filter by cuisine, preparation time, difficulty
  - Search functionality

- **Recipe Detail Page**
  - Recipe image
  - Ingredients list with "Add all to cart" button

- **Shopping Cart Page**
    - Product list with images
    - Quantity adjusters
    - Remove item option
    - Price summary
    - Checkout button

- **Checkout**
    - Address selection/entry
    - Payment method selection
    - Order summary
    - Confirmation page

The interface will support touch interactions for mobile users and will provide keyboard navigation accessibility features. Error messages and validation feedback will be displayed prominently near the relevant input fields.

3.1.2 Hardware Interfaces

The platform does not directly interact with hardware components but relies on standard web technologies. For user devices, the application will use responsive design techniques to adapt to different screen sizes and resolutions.

3.1.3 Software Interfaces

The platform interacts with external software components to enhance functionality:

- **Payment Gateways :**
    - Integration with third-party services like Stripe for secure payment processing.
- **Notification Services :**
    - Integration with SMS/email APIs for order confirmations and updates.
- **Authentication Services :**
    - Use of JWT (JSON Web Token) for secure user authentication.
- **Database :**
    - MongoDB for storing user profiles, product details, orders, and recipes.

3.2 Functional Requirements

3.2.1 User Management

F1: User Registration and Authentication
The system shall allow users to create accounts using email or social media authentication.

F2: User Profile Management
The system shall enable users to view and update their profile information, including name, contact details, and preferences.

F3: Address Management
The system shall allow users to add, edit, delete, and select multiple delivery addresses.

F4: Password Management
The system shall provide functionality for users to reset their password via email verification.

3.2.2 Product Catalog Management

F5: Product Browsing
The system shall allow users to browse products by categories, subcategories, and featured collections.

F6: Product Search
The system shall provide a search functionality with filters for price range, brand, dietary restrictions, and other relevant attributes.

F7: Product Details
The system shall display comprehensive product information including images, description, price, nutritional information, and customer reviews.

3.2.3 Recipe Management

F8: Recipe Browsing
The system shall allow users to browse recipes by categories, cuisine types, preparation time, and difficulty level.

F9: Recipe Details
The system shall display comprehensive recipe information including ingredients, preparation steps, cooking time, serving size, and nutritional information.

F10: Recipe-to-Cart Conversion

The system shall allow users to add all ingredients from a recipe to their shopping cart with a single click.

3.2.4 Shopping Cart Management

F11: Cart Operations
The system shall allow users to add, remove, and update quantities of products in their cart.

F12: Cart Persistence
The system shall maintain cart contents across sessions and devices for logged-in users.

F13: Price Calculation
The system shall automatically calculate subtotals, taxes, delivery charges, and total price.

3.2.5 Checkout Process

F14: Checkout Flow
The system shall guide users through a step-by-step checkout process including address selection and payment method selection.

F15: Order Review
The system shall allow users to review their order details before confirmation.

F16: Payment Processing
The system shall securely process payments through multiple payment methods including credit/debit cards, digital wallets, and cash on delivery.

F17: Order Confirmation
The system shall provide order confirmation with a unique order ID and estimated delivery time.

3.2.6 Order Management

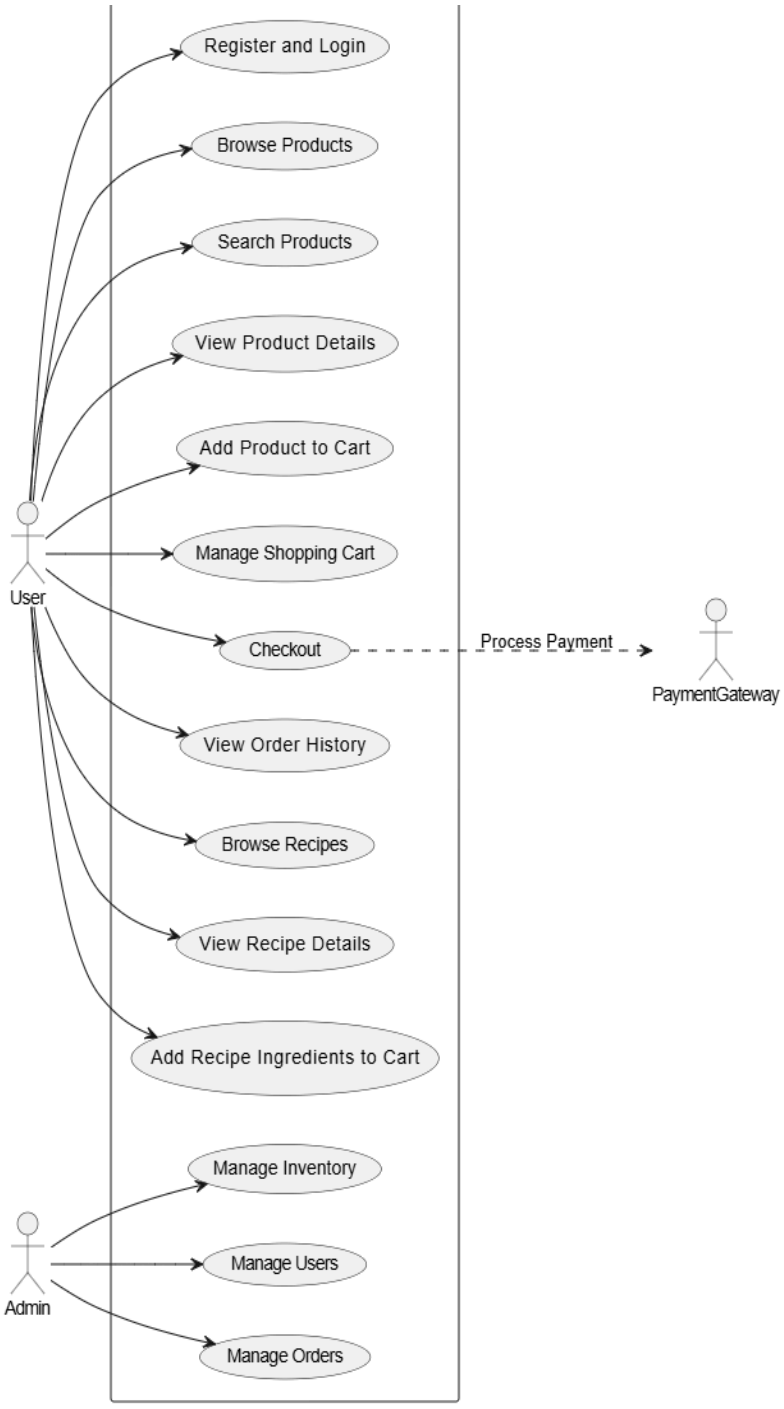
F18: Order History
The system shall maintain and display order history for users.

3.2.7 Admin Functions

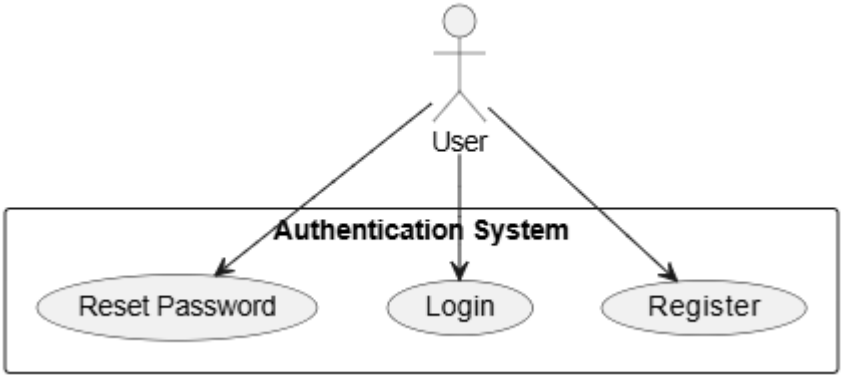
F19: Inventory Management
The system shall provide administrators with tools to manage product inventory, including adding new products, updating stock levels, and setting price information.
F20: Order Management
The system shall allow administrators to view, process, and update order statuses.
F21: User Management

The system shall enable administrators to manage user accounts, including viewing user information and handling support requests.
F22: Content Management
The system shall provide tools for administrators to manage recipes, product categories, and promotional content.

3.3 Use Case Model



3.3.1 Use Case #1: Register and Login (U1)



Author – Chinmai Kewlani

Purpose - To allow users to create an account and log in to access personalized shopping features.

Requirements Traceability – F1: User Registration and Authentication

Priority - High

Preconditions -

- The system must be online.
- The user must have a valid email or social media account.

Post conditions -

- The user account is successfully created.
- The user is logged into the system.

Actors – User

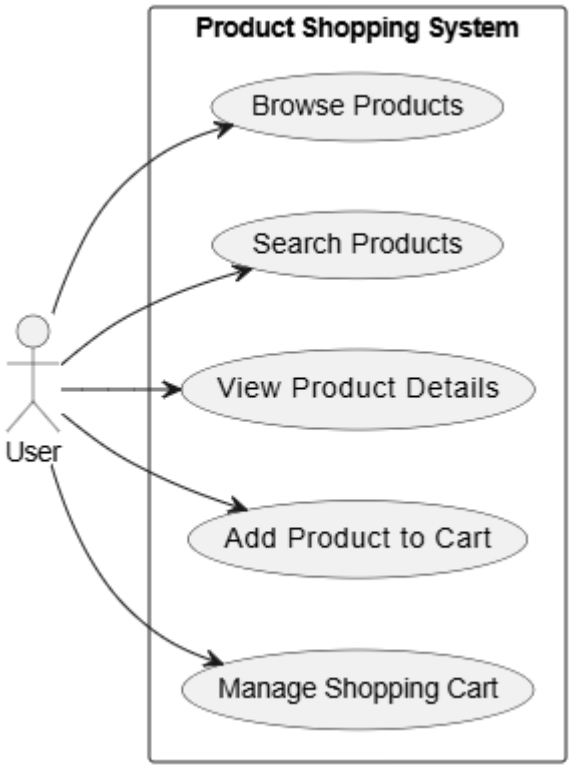
Flow of Events

1. Basic Flow -
    1. User selects "Register" or "Login" option.

2. User enters credentials (email, password, or social login).
3. System verifies credentials.
4. If valid, the user is logged in and redirected to the homepage.

2. Alternative Flow -
    - If a user forgets their password, they can reset it via email verification.
3. Exceptions -
    - Invalid credentials → System displays an error message.
    - Server error → System informs the user and asks them to retry later.

3.3.2 Use Case #2: Browse Products (U2)



Purpose - To allow users to browse available products in different categories.

Requirements Traceability – F5: Product Browsing

Priority - High

Preconditions -

- The system must have products listed.
- The user must be logged in (optional).

Post conditions -

- The user has viewed product details or added items to the cart.

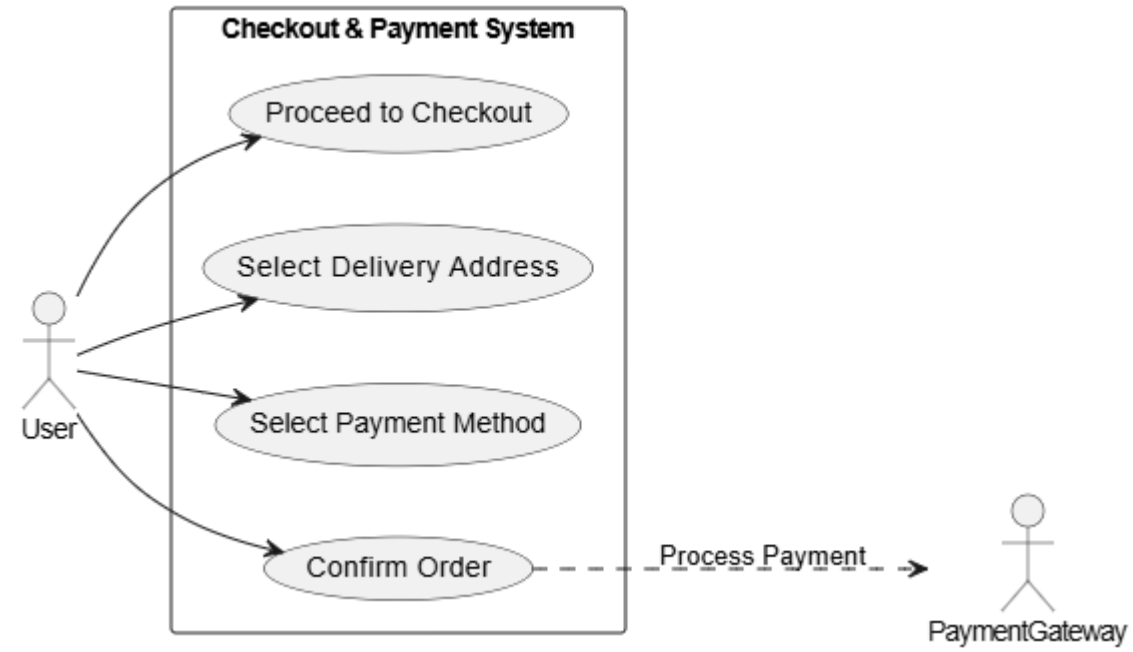
Actors – User

Flow of Events

1. Basic Flow
    a. User selects a product category.
    b. System displays a list of products.
    c. User clicks on a product to view details.
2. Alternative Flow
    - User applies filters or sorts the products.
3. Exceptions -
    - No products available → System displays a "No products found" message.

Notes/Issues - Need to optimize product loading time.

3.3.3 Use Case #3: Checkout & Payment Processing (U3)



Purpose - To allow users to finalize purchases and make payments.

Requirements Traceability –

- F14: Checkout Flow
- F16: Payment Processing

Priority - High

Preconditions -

- User has items in the cart.
- User is logged in.

Post conditions -

- Order is successfully placed.
- Payment is processed.

Actors –

- **User** (primary actor)
- **Payment Gateway** (secondary actor)

Flow of Events

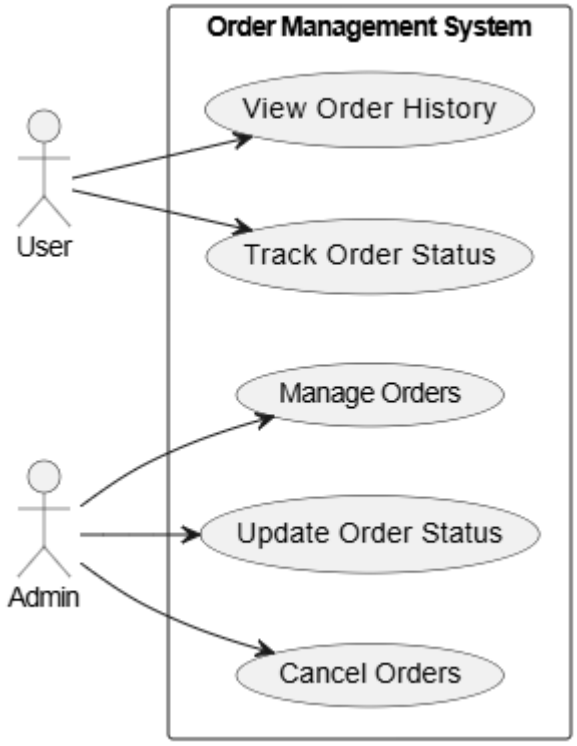
1. Basic Flow -
    a. User selects "Checkout".
    b. System prompts user to select a delivery address.
    c. User selects a payment method.
    d. Payment is processed.
    e. Order is confirmed, and an email notification is sent.
2. Exceptions -

- Payment failure → System notifies the user and asks to retry.
- Out-of-stock items → System prompts user to remove or replace the item.

Includes View Cart

Notes/Issues - Consider implementing multiple payment options.

3.3.4 Use Case #4: Order Management (U4)



Author – Vrushil Parikh

Purpose - To allow users to track orders and admins to manage orders.

Requirements Traceability –

- F18: Order History
- F20: Order Management

Priority - High

Preconditions -

- Orders must exist in the system.
- The user must be logged in.

Post conditions -

- Users can view past orders.
- Admins can manage and update orders.

Actors –

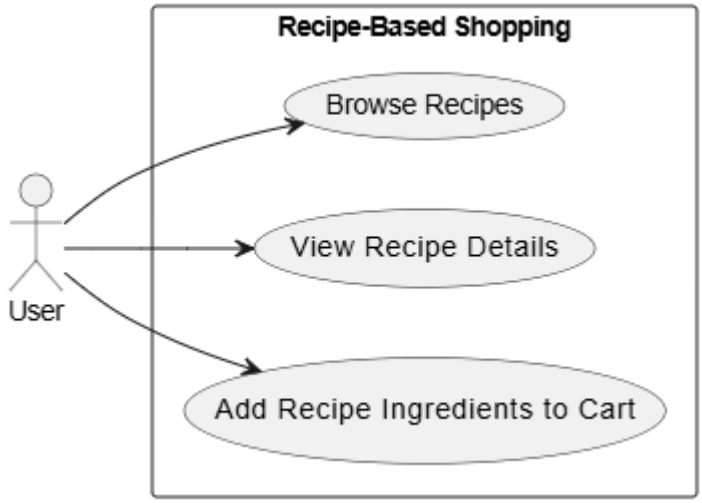
- User
- Admin

Flow of Events

1. Basic Flow -
    a. User selects "Order History".
    b. System displays past orders with status updates.
    c. Admin selects "Manage Orders".
    d. Admin updates order status (e.g., shipped, delivered, canceled).
2. Alternative Flow -
    - User cancels an order before it is shipped.
3. Exceptions -
    - Order details not available → System displays an error message.

Notes/Issues - Need to add notifications for order status updates.

3.3.5 Use Case #5: Recipe-Based Shopping (U5)



Purpose - To allow users to browse recipes and add ingredients directly to the cart.

Requirements Traceability –

- F8: Recipe Browsing
- F10: Recipe-to-Cart Conversion

Priority - Medium

Preconditions -

- The system must have recipes stored in the database.
- The user must be logged in.

Post conditions -

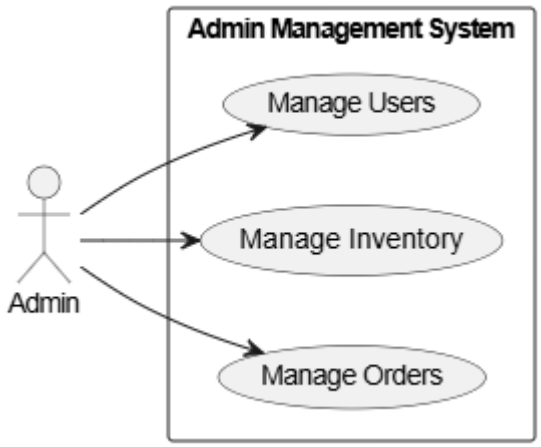
- The user successfully adds ingredients from a recipe to their cart.

Flow of Events

1. Basic Flow -
    a. User selects "Browse Recipes".
    b. System displays a list of recipes.
    c. User selects a recipe to view details.
    d. User clicks "Add Ingredients to Cart".
    e. System adds ingredients to the shopping cart.
2. Alternative Flow -
    - User removes unwanted ingredients before adding to the cart.
3. Exceptions -
    - Some ingredients are out of stock → System notifies the user.

Notes/Issues - Need to ensure ingredient mapping with inventory items.

3.3.6 Use Case #6: Admin Management (U6)



Author – Virag Koradiya

Purpose - To allow admins to manage users, inventory, and orders.

Requirements Traceability –

- F19: Inventory Management
- F21: User Management

Priority - High

Preconditions - The admin must be logged in.

Post conditions - Admin can successfully manage system resources.

Actors – Admin

Flow of Events

1. Basic Flow -
    a. Admin logs in.
    b. Admin selects the management option (Users, Inventory, Orders).
    c. Admin makes the necessary updates.
    d. System saves changes.
2. Exceptions - Data update fails → System prompts the admin to retry.

Notes/Issues - Need to add role-based access control for admin users.

4 Other Non-functional Requirements

4.1 Performance Requirements

Performance requirements specify how the system should behave under various conditions to ensure a smooth and responsive user experience.

- P1: Page Load Time
    - The platform must load pages within 2 seconds under normal network conditions.
    - This ensures users can quickly browse products, view recipes, and complete purchases without delays.
- P2: Concurrent User Handling
    - The system must support up to 100 concurrent users without performance degradation.
    - This ensures the platform remains responsive during peak traffic periods, such as during sales or promotional events.
- P3: Recipe-to-Cart Processing Time
    - The system must process recipe-to-cart requests (e.g., adding all ingredients from a recipe to the cart) within 2 seconds .
    - This ensures users experience minimal delay when interacting with the recipe-based shopping feature.
- P4: Search Response Time
    - The search functionality must return results within 1 second for queries involving up to 10,00 products.
    - This ensures users receive quick feedback when searching for products or recipes.

4.2 Safety and Security Requirements

Safety and security requirements ensure the platform protects user data, prevents unauthorized access, and complies with relevant regulations.

- S1: Data Encryption
    - All user data transmitted between the client and server must be encrypted using HTTPS .
    - This ensures sensitive information, such as login credentials and personal details, remains protected from interception.
- S2: Payment Security
    - Payment processing must comply with PCI DSS (Payment Card Industry Data Security Standard) principles.
    - This ensures payment information is handled securely and minimizes the risk of fraud.
- S3: User Authentication
    - The platform must implement secure user authentication mechanisms to prevent unauthorized access to user accounts.
    - This ensures only legitimate users can access their accounts and perform actions like placing orders.
- S4: Data Privacy Compliance
    - The platform must comply with GDPR (General Data Protection Regulation) principles for handling user data.
    - This ensures user data is collected, stored, and processed in a transparent and secure manner, protecting user privacy.

4.3 Software Quality Attributes

Software quality attributes define the characteristics of the platform that are important to both customers and developers. These attributes ensure the system is reliable, usable, maintainable, and scalable.

4.3.1 Reliability

- R1: System Uptime
    - The system must achieve 99.9% uptime to ensure consistent availability for users.
    - This ensures users can access the platform whenever they need to shop, minimizing disruptions.
- R2: Error Recovery
    - The system must recover gracefully from errors (e.g., database connection failures) within 5 seconds .
    - This ensures users are not locked out of the platform due to temporary issues.

4.3.2 Usability

- U1: Usability Score
    - The platform must achieve a usability score of 80% or higher based on feedback from user testing.

        - This ensures the platform is intuitive and easy to use for a wide range of users.
- U2: Accessibility Compliance
    - The platform must comply with WCAG 2.1 Level AA accessibility standards.
    - This ensures users with disabilities can navigate and interact with the platform effectively.

4.3.3 Maintainability

- M1: Codebase Maintainability
    - The platform's codebase must be modular and well-documented to facilitate future updates and bug fixes.
    - This ensures developers can easily make changes to the system as requirements evolve.
- M2: Version Control
    - The platform's source code must be managed using a version control system (e.g., Git).
    - This ensures changes are tracked, and collaboration among team members is seamless.

4.3.4 Scalability

- SC1: Horizontal Scaling
    - The platform must support horizontal scaling to handle increased traffic as the user base grows.
    - This ensures the system can accommodate growth without compromising performance.
- SC2: Database Optimization
    - The platform must use indexing and query optimization techniques to ensure efficient database operations.
    - This ensures fast retrieval of product, order, and recipe data even as the dataset grows.

Data Flow Diagram Level 0