SOFTWARE REQUIREMENTS SPECIFICATION

For

Recipe Management System

Presented By:

Rajesh Kumar E

Rahul V

Naveen P R

Mohammed Shameem S

1. Introduction

1.1 Purpose:

The purpose of a Recipe Management System is to provide individuals with a user-friendly and centralized platform for efficiently organizing, creating, and sharing culinary recipes. By offering comprehensive details on ingredients and cooking instructions, the system enhances the cooking experience, empowering users to plan meals effectively and streamline grocery shopping through integrated shopping lists. The platform facilitates collaboration and community engagement by allowing users to share their favorite recipes, receive feedback through ratings, and contribute to a collective culinary knowledge base. With user authentication ensuring secure access, the Recipe Management System not only organizes recipes but also nurtures a dynamic and interactive cooking community, encouraging a culture of exploration, creativity, and shared gastronomic experiences.

1.2 Document Conventions:

Font Style/Size: Calisto MT / Text size 12 and 14

Language: English(India), English(United States)

Text Formatting: Use bold or italics to emphasize important points, headings, or key terms. Consistency in formatting enhances visual appeal.

Document Structure: Clearly define sections, including an introduction, main content, and conclusion.

Page Numbering: Include page numbers, typically in the header or footer, to assist readers in tracking their progress through the document.

1.3 Scope of the Project:

The scope of a development project defines the boundaries, objectives, deliverables, and constraints that shape the entire undertaking. It encompasses the features, functionalities, and outcomes the project aims to achieve. In the context of software development, the scope delineates the specific requirements, user interactions, and system behaviors that will be addressed during the project's lifecycle. It sets clear expectations for the project team and stakeholders, outlining what will be included and, equally importantly, what will not. A well-defined scope helps prevent scope creep, ensuring that the project stays on track, within budget, and aligned with the intended goals. It serves as a roadmap for decision-making, resource allocation, and risk management, providing a comprehensive understanding of the project's scale and complexity. Additionally, a detailed scope facilitates effective communication among team members, stakeholders, and any external entities involved in the project.

1.4 Definitions, Acronyms and Abbreviations:

JAVA - platform independence

SQL - Structured query Language

ER - Entity Relationship

UML - Unified Modeling Language

IDE - Integrated Development Environment

SRS - Software Requirement Specification

1.5 References:

Books:

Websites:

2. Overall Descriptions

2.1 Product Perspective:

The product perspective of a development project provides a holistic view of how the product fits into its environment. It outlines the relationships and interactions between the product and external entities, systems, or users. This perspective ensures a comprehensive understanding of the product's functionality, interfaces, and dependencies, contributing to effective integration and seamless user experiences. It is crucial for identifying potential challenges, aligning with existing systems, and fostering interoperability, ultimately guiding the project toward successful implementation within its broader context.

2.2 Product Function:

The product function refers to the core purpose and capabilities of a particular product. It defines what the product is designed to do and the problems it aims to solve. The functions of a product outline the features, operations, and interactions that users can expect. These functions are typically aligned with the product's overall goals and specifications. Understanding product functions is essential for both development teams and end-users, as it sets expectations and guides the design and implementation process to ensure that the product effectively fulfills its intended purpose.

2.3 User Classes and Characteristics:

User classes in a system represent distinct groups of individuals or entities with common characteristics and shared needs. These classes help in tailoring the user experience to specific requirements. Characteristics may include factors like technical expertise, preferences, or usage patterns. Understanding these user classes and their unique traits is crucial for designing interfaces and features that cater to diverse user needs. This user-centric approach enhances usability and ensures that the system is accessible and functional for a broad range of users.

Admin Features:

1. User Management:

- Create, edit, and delete user accounts.
- Manage user roles and permissions.

2. Recipe Management:

- Add, edit, and delete recipes.
- View and moderate user-submitted recipes.
- Monitor recipe ratings and comments.

3. Ingredient Management:

- Add, edit, and delete ingredients.
- Manage ingredient categories.

4. User Analytics:

- View user activity logs.
- Analyze popular recipes and user engagement.

5. System Configuration:

- Configure system settings and preferences.
- Manage email notifications and alerts.

6. Content Moderation:

- Moderate user-generated content (recipes, comments) for compliance and appropriateness.

7. Backup and Recovery:

- Perform data backups regularly.
- Implement recovery procedures in case of data loss.

User Features:

1. Recipe Creation:

- Create new recipes with details such as title, ingredients, instructions, etc.
- Add images to recipes.

2. Recipe Modification:

- Edit and update existing recipes.
- Modify ingredient details and instructions

3. Recipe Sharing:

- Share recipes with other users.
- Set privacy settings for shared recipes (public, private, or with specific users).

4. Rating and Feedback:

- Rate recipes on a scale.
- Leave comments and feedback on recipes.

5. Search and Browsing:

- Search for recipes based on title, category, or ingredients.
- Browse recipes by categories.

6. Shopping List:

- Create a shopping list based on selected recipes.
- Manage and update the shopping list.

7. Meal Planning:

- Plan meals by associating recipes with specific days.
- View and edit meal plans.

8. User Profile:

- View and edit user profiles.
- Track personal recipe collections and activity history.

9. Notifications:

- Receive notifications for recipe updates, comments, and shared recipes.

10. Accessibility:

- Access the system through various devices with a responsive design.

This division of features caters to the specific needs and responsibilities of administrators and regular users within the Recipe Management System.

2.4 Operating Environment:

The operating environment of a system encompasses the hardware, software, network configurations, and external factors that influence its performance. In the case of a Recipe Management System, it typically operates in diverse environments, including desktop computers, laptops, tablets, and smartphones, utilizing web browsers as the primary interface. The system relies on stable internet connectivity for seamless access and data retrieval. Compatibility with major web browsers such as Chrome, Firefox, and Safari is essential. The system should also adhere to security standards, ensuring data protection and privacy compliance. Considerations for different operating systems and devices enhance accessibility and usability.

2.5 Assumptions and Dependencies:

Assumptions:

- The system assumes users have a stable internet connection for seamless access and data retrieval.
- The system assumes compatibility with major web browsers (Chrome, Firefox, Safari) for optimal user experience.
- Assumption that users provide accurate authentication details during the registration process.
- Assumes users enter accurate information while creating and editing recipes and ingredients.
- Assumes users will comply with the terms of service, privacy policies, and community guidelines.

Dependencies:

- The system depends on the chosen DBMS (e.g., MySQL, PostgreSQL) for efficient data storage and retrieval.
- Dependencies on the selected web development framework (e.g., Django, Flask) for the system's architecture and functionality.
- Dependencies on external APIs (e.g., for authentication, notifications) and their availability and reliability.
- Dependency on a reliable hosting service (e.g., AWS, Heroku) for system deployment and uptime.

- Dependencies on ISPs for providing consistent and reliable internet connectivity to end-users.
- Dependencies on security protocols and practices to ensure data protection and user privacy.
- Dependencies on users' devices and browsers adhering to web standards and supporting required functionalities.
- Dependencies on the availability and expertise of the development team for ongoing maintenance, updates, and bug fixes.

2.6 Requirement

Software Configuration:

This software package is developed using java as front end which is supported by sun micro system. Microsoft SQL Server as the back end to store the database. Operating System: Windows NT, windows 98, Windows XP

Language: Java Runtime Environment, Net beans 7.0.1 (front end)

Database: MS SQL Server (back end)

Hardware Configuration:

Processor: Pentium(R)Dual-core CPU

Hard Disk: 40GB

RAM: 256 MB or more

2.7 Data Requirement:

The data requirements for a Recipe Management System encompass the need for a relational database to store user details, recipe information, and ingredient data. The database should include tables for users, recipes, ingredients, and their relationships. User-specific data would include login credentials, preferences, and activity logs. Recipe data should cover title, instructions, ratings, and sharing settings. Ingredient data would involve names, categories, and quantities. These data elements are critical for the system's functionality, providing a structured and organized framework for efficient data management and retrieval.

3. External Interface Requirement

External interface requirements for a Recipe Management System involve interactions between the system and external entities, including users and other applications. The user interface should be intuitive and accessible across various devices and web browsers, ensuring a seamless user experience. Additionally, the system may integrate with external APIs for functionalities like user authentication, notifications, or ingredient information retrieval. The interface should comply with web standards, accessibility guidelines, and security protocols to facilitate smooth interactions, data exchange, and interoperability with external components.

3.1 GUI

The Graphical User Interface (GUI) for a Recipe Management System plays a pivotal role in providing users with a visually appealing and user-friendly interaction platform. The GUI should feature an intuitive design with easy navigation, allowing users to efficiently create, edit, and manage recipes. Key components include:

1. Navigation Menu:

- Clearly defined menus for recipe management, user profiles, and settings.

2. Recipe Display:

- A visually appealing layout showcasing recipe details, including title, ingredients, instructions, and images.

3. User Profile:

- A dedicated section for users to view and edit their profiles, track activity, and manage shared recipes.

4. Search and Browsing:

- User-friendly search functionality and category-based browsing for discovering recipes.

5. Interactive Forms:

- Intuitive forms for creating and editing recipes, with input fields for title, instructions, ingredients, etc.

6. Rating and Comments:

- User-friendly interfaces for rating recipes and leaving comments, encouraging user engagement.

7. Shopping List and Meal Planning:

- Dedicated sections for creating and managing shopping lists and meal plans, enhancing user organization.

8. Notifications:

- An interface for users to receive and manage notifications related to recipe updates, comments, and shared recipes.

9. Responsive Design:

- Ensuring the GUI is responsive to different screen sizes and devices for a consistent user experience.

10. Color Scheme and Themes:

- A visually pleasing color scheme and theme that aligns with the overall aesthetic and user preferences.

11. Accessibility Features:

- Inclusion of accessibility features to ensure usability for users with diverse needs, such as readable text, proper contrast, and keyboard navigation.

12. Consistency:

- Consistent design elements, fonts, and button placements for a cohesive and familiar user experience.

13. Error Handling:

- Clear and user-friendly error messages to guide users in case of input mistakes or system issues.

14. Security Indicators:

- Visual indicators of secure connections and data encryption, fostering user trust in the system's security.

15. Feedback Mechanisms:

- Providing feedback to users after actions, such as successful recipe creation or update.

The GUI should strike a balance between aesthetics and functionality, ensuring that users can easily navigate, interact, and derive value from the Recipe Management System. Regular usability testing and user feedback should inform ongoing refinements to enhance the overall user experience.

4. System Features

Core System Features:

The Recipe Management System includes fundamental features like secure user authentication, recipe and ingredient management, sharing, and rating functionalities. Users can create, edit, and delete recipes, associate them with ingredients, and efficiently plan meals. The system facilitates seamless sharing, rating, and commenting on recipes, along with robust search and browsing capabilities. Users can create and manage shopping lists, receive notifications, and personalize their profiles, ensuring a comprehensive yet user-friendly experience.

Advanced System Features:

In addition to the core functionalities, the system offers advanced features such as analytics dashboards for user insights, community engagement options, external integrations, and support for multiple languages. Users can leave comments and feedback, participate in forums, and share recipes on social media. The system ensures security measures, backup options, and user education tools, enhancing overall usability and catering to diverse user preferences and needs.

5. Other Non-functional Requirements

5.1 Performance Requirement:

- The Recipe Management System must deliver a responsive experience with a response time of under 2 seconds for common user interactions.
- It should scale efficiently to accommodate a minimum of 1000 concurrent users, handle large datasets with optimized search and database query speeds, and support fast upload/download of media files.
- The system must maintain robust performance during peak loads, employ caching mechanisms, and implement fault tolerance to ensure continuous availability.
- Regular performance monitoring and load testing are essential to identify and address potential issues proactively.

5.2 Safety Requirement:

The Recipe Management System must prioritize user safety and data security. It should adhere to robust encryption standards to protect sensitive user information. Additionally, the system must implement measures to prevent unauthorized access, ensuring the confidentiality and integrity of user data. Regular security audits and updates are imperative to identify and address potential vulnerabilities, maintaining a secure environment for users to interact with the platform.

5.3 Security Requirement:

- The Recipe Management System must enforce stringent security measures, including secure user authentication, data encryption, and protection against common security threats.
- Access controls and authorization mechanisms must be in place to safeguard user data.
- Regular security audits, updates, and adherence to best practices are essential to mitigate potential vulnerabilities and ensure the overall integrity, confidentiality, and availability of the system.

5.4 Requirement Attributes:

- Each system requirement should possess specific attributes to ensure clarity and effectiveness.
- These attributes include uniqueness, ensuring each requirement is distinct; traceability, allowing for easy tracking and management; completeness, ensuring all necessary aspects are covered; feasibility, verifying practicality; and testability, allowing for effective validation.
- Additionally, requirements should be prioritized to guide development efforts and be accompanied by acceptance criteria, providing clear benchmarks for successful implementation.
- These attributes collectively contribute to the precision, manageability, and successful realization of system requirements.

5.5 Business Rules:

Business rules in the context of a Recipe Management System serve as guidelines and constraints that govern how the system operates within the broader organizational context. These rules define the relationships, behaviors, and processes that users and the system must adhere to. Examples include privacy regulations for user data, guidelines for recipe sharing, and criteria for user authentication. Business rules ensure consistency, compliance, and alignment with organizational policies, contributing to the effective and ethical functioning of the Recipe Management System.

5.6 User Requirement:

User requirements for a Recipe Management System outline the specific needs and expectations of the system's end-users. These include features such as user registration, intuitive recipe creation and editing interfaces, efficient search and browsing functionalities, and the ability to share and rate recipes. Users also expect secure authentication, personalized profiles, and notifications for recipe updates. The system should prioritize a user-friendly experience, responsiveness across devices, and accessibility features. These requirements collectively address the preferences and interactions of users, ensuring a satisfying and effective user experience with the Recipe Management System.

The admin provide certain facilities to the users in the form of :

- Backup and Recovery.
- Forgot Password.
- Data migration (i.e.) whenever user registers for the first time then the data is stored in the server.
- Data replication (i.e.) if the data is lost in one branch, it is still stored with the server.
- Auto Recovery (i.e.) frequently auto saving the information.
- Maintaining files (i.e.) File Organization.
- The server must be maintained regularly and it has to be updated from time to time.

6. Other Requirements

6.1 Data and Category Requirement:

For a Recipe Management System, data requirements include the need for a structured database to store user information, recipe details, and ingredient data. User data encompasses account information, preferences, and activity logs. Recipe data should cover elements such as title, instructions, ratings, and sharing settings. Ingredient data involves names, categories, quantities, and units. Additionally, category requirements involve organizing recipes and ingredients into logical and user-friendly categories, enhancing the system's navigability and helping users efficiently explore and manage their culinary preferences. These requirements collectively form the foundation for organized data management and user interaction within the system.

6.2 Appendix

A: Admin, Abbreviation, Acronym, Assumptions;

B: Books, Business rules;

C: Class, Client, Conventions;

D: Data requirement, Dependencies;

G: GUI;

K: Key;

L: Library, Librarian;

M: Member;

N: Non-functionalRequirement;

O: Operatingenvironment;

P: Performance,Perspective,Purpose;

R: Requirement, Requirement attributes;

S: Safety, Scope, Security, System features;

U: User, User class and characteristics, User requirement;

6.3 Glossary:

Recipe: A culinary document detailing the ingredients, quantities, and instructions required to prepare a specific dish.

Ingredient: A component used in the preparation of a recipe, identified by its name, quantity, and unit of measurement.

User: An individual interacting with the Recipe Management System, with roles including administrator and regular user.

Administrator: A user role with elevated privileges, responsible for system management, user permissions, and content moderation.

Regular User: A standard user role with access to common functionalities, such as recipe creation, browsing, and interaction.

Category: A classification system for recipes and ingredients, facilitating organization and user-friendly navigation within the system.

User Authentication: The process of verifying and confirming the identity of a user through secure login credentials.

Rating: A numerical or qualitative assessment assigned by users to recipes, reflecting their satisfaction or preference.

Sharing Settings: Configurable options that determine the visibility and accessibility of a recipe to other users within the system.

Wireframe: A visual representation or blueprint illustrating the layout and components of a user interface.

Encryption: The process of converting sensitive data into a secure format to protect it from unauthorized access.

Access Controls: Security measures that regulate user access to system features and functionalities based on predefined permissions.

API (Application Programming Interface): A set of protocols and tools that allows different software applications to communicate and share data.

Acceptance Criteria: Clearly defined conditions that must be met for a particular feature or functionality to be considered successfully implemented.

UAT (User Acceptance Testing): The final phase of testing where end-users evaluate the system to ensure it meets their requirements and expectations.

Unit Testing: The process of testing individual components or modules of the system to ensure they function correctly in isolation.

Database Schema: The blueprint or structure that defines how data is organized, stored, and accessed within a database.

Third-Party Integrations: External services or tools that are integrated into the Recipe Management System to enhance functionality, such as external APIs.

This glossary provides definitions for key terms related to the Recipe Management System, offering clarity and understanding of the system's components and functionalities.

6.4 Class Diagram:

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling, translating the models into programming code. Class diagrams can also be used for data modeling.

In software engineering, a class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects.