SOFTWARE REQUIREMENTS SPECIFICATION

RECIPE ROULETTE

Software Engineering

Bachelor's Degree in Bioinformatics

Curse 2023-24

Group member:

Alessandra Bonilla Salon

Maria Lopez Moriana

Carmen Samedi

Supervisor:

Daniel Soto Alvarez

ESCI upf.
School of International Studies

I. REVISION HISTORY

Name	Date	Reason For Changes	Version
María/Alessandra/Carmen	16/04/2024	Creation of document	0.1

II. TABLE OF CONTENTS

1. INTRODUCTION

1.1. PURPOSE

The purpose of the "Recipe Roulette" app is to provide a user friendly platform that provides personalized cooking assistant that helps users utilize their available ingredients to discover and prepare new recipes. By doing so, the app aims to reduce food waste and enhance the cooking experience for individuals of varying culinary skill levels, from beginner cook to culinary expert.

1.2. INTENDED AUDIENCE

This Software Requirements Specification (SRS) document outlines the functional and non-functional requirements for "Recipe Roulette", it provides a detailed overview of the application's features, intended use, and constraints. It serves as a guideline for the development team to design and implement the software and as a contract between the stakeholder and the development team, ensuring that the final product meets the users' and stakeholders' expectations.
The intended audience would include:
Project Managers: Who organize and oversee of the project progression.
Developers: Use this document as reference for feature implementation of the app.
Quality control team: Who ensure the app meets the outlined specifications.
Users

1.3. PRODUCT SCOPE

"Recipe Roulette" is a mobile and web application that aims to minimize food wastage and promote cooking at home by recommending recipes based on ingredients that users already have. Provides an holistic approach to cooking by facilitating recipe sharing, community building, and culinary education. It allows users to enter ingredients, apply filters for dietary preferences, and receive recipes that they can prepare. Additionally, users can contribute recipes, earn rewards for engagement, and partake in a community of cooking enthusiasts. It includes a comprehensive notification system for engaging users with new recipes and app updates.

1.4. REFERENCES

- Food waste statistics and research
- Documentation of third parties if any
- IEEE Recommended practice for SRS

2. OVERALL DESCRIPTION

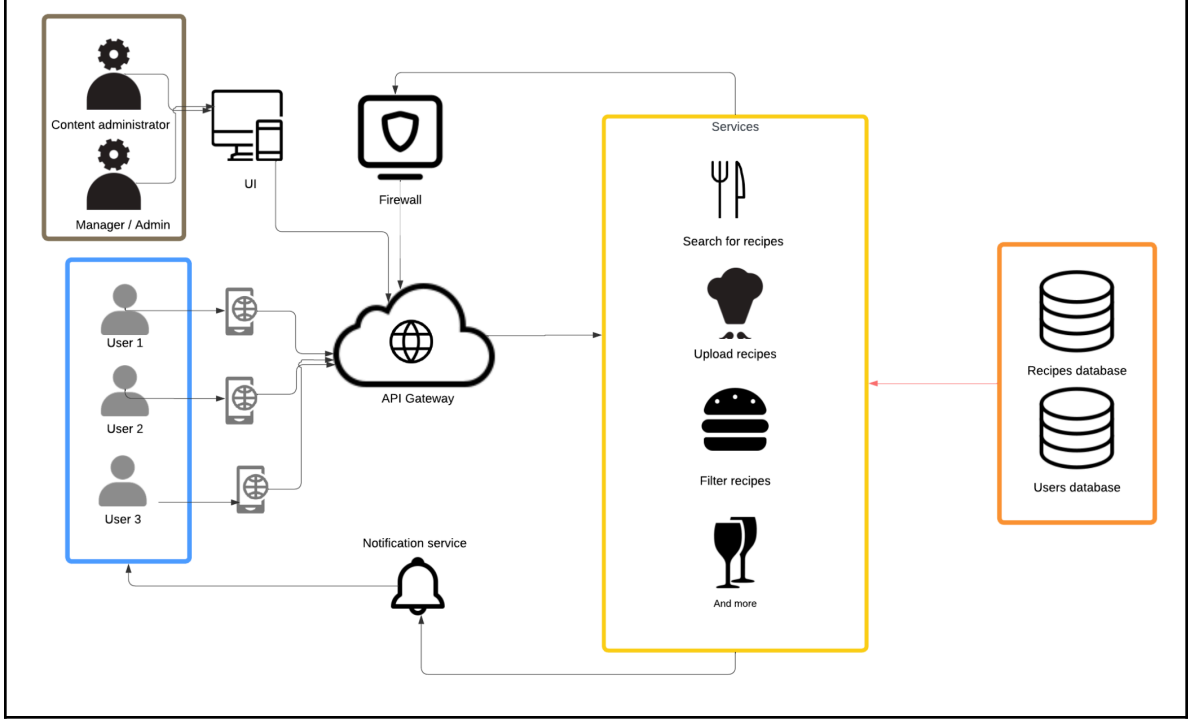
2.1. PRODUCT PERSPECTIVE

"Recipe Roulette" is introduced as a new application in the culinary space, distinct from existing products. Its main feature is to recommend recipes based on user-inputted ingredients, facilitating meal planning and cooking for individuals. The app is standalone and does not serve as a continuation of or replacement for any other product.

At its inception, "Recipe Roulette" will operate independently. Future updates may, however, extend its functionality to include integration with online grocery services and smart kitchen devices to further aid the cooking process. Additionally, features to share content on social media may be added to foster a sense of community among users.

The application will source recipe information from an assortment of databases and APIs. It may also offer insights into user preferences and behaviors to those conducting market research or analytics, pending the activation of such features.

Architecture Diagram



2.2. PRODUCT FEATURES

Functional Req. ID #	Actor	Functional Name	Functional Requirement Description
UF-1	User	User registration	The user should be able to register on the platform by providing necessary information such as username, email, password name, address, and phone number.
UF-2	User	Delete Profile	Users should have the option to delete their profile and all associated data from the app.
UF-3	User	Search Recipes by Ingredients	Users should be able to search for recipes by inputting the ingredients they have available. The app will then display recipes that can be made using those ingredients.
UF-4	User	Search Recipes by Dish Name	Users should have the option to directly search for recipes by the name of the dish they want to prepare.
UF-5	User	Filter Recipes	Users should be able to filter recipes based on various criteria such as time required for preparation, type of food (e.g., Mediterranean, Japanese), dietary restrictions (e.g., vegetarian, lactose intolerance), difficulty level, and allergies.
UF-6	User	Upload, Modify, Delete Recipes	Users should be able to upload their own recipes to the platform, as well as modify or delete recipes they have previously uploaded.
UF-7	User	Gain Points	Users should earn points for sharing recipes on the platform. These points can be used for rewards.
UF-8	User	Check Reward Status	Users should be able to check their current reward status, including the number of points earned and available rewards.
MF-1	Manager	Management of Software	Managers should have access to administrative tools to manage user accounts, app content, and overall system maintenance. Some actions that have to be able to access are ● Modify the interface ● Do updates if needed ● Check statistics ● Promote the software (marketing) ● Check the security ● Approve new delivery staff accounts (check documents id (photo) and a pdf CV)
CF-1	Content administrator	Approve recipe	The content administrator must have the capability to review submitted recipes, ensuring they meet the app's quality standards and guidelines. Upon review, the administrator can approve recipes for publication on the platform.

CF-2	Content administrator	Send Notifications	If a submitted recipe does not meet the app's criteria for publication (e.g., incomplete information, low quality or inadequate content), the content administrator should be able to notify the user who submitted the recipe, requesting modifications or additional information to meet the app's standards.
CF-3	Content administrator	Delete recipe	The content administrator should have the authority to remove recipes from the platform if they violate community guidelines, contain inappropriate content, or for any other valid reason deemed necessary by the administration team.
CF-4	Content administrator	Revise Content	The content administrator should be able to review and revise uploaded videos associated with recipes. This may involve checking for quality, ensuring they adhere to content guidelines, and making necessary edits or annotations to improve the user experience.

2.3. USER CLASSES AND CHARACTERISTICS

Novice Cooks: Users new to cooking who require step-by-step guidance.
Busy Professionals: Individuals seeking quick and easy meal solutions.
Dietary Restricted Users: People with specific dietary needs looking for suitable recipes.
Culinary Experts: Experienced chefs exploring new recipe ideas and sharing their knowledge.
Community Participants: Users who engage with the app through forums and recipe sharing.

2.4. OPERATING ENVIRONMENT

"Recipe Roulette" will be developed for iOS and Android platforms, as well as web browsers, ensuring wide accessibility.

2.5. DESIGN AND IMPLEMENTATION CONSTRAINTS

Device Compatibility: The app must be optimized for a wide range of device sizes and capabilities.
Internet Dependency: Real-time features and recipe updates require a stable internet connection.
Content Moderation: A system for moderating user-submitted recipes to maintain quality.
Scalability: The cloud infrastructure must handle an increasing number of users and data volume.
Regulatory Compliance: The app must comply with data protection and privacy laws.

2.6. USER DOCUMENTATION

Comprehensive documentation will be provided, including:

User Guide: Instructions on app functionalities and navigation.
FAQ Section: Answers to common questions for troubleshooting.
Video Tutorials: Visual guides for app usage and recipe preparation.

2.7. ASSUMPTIONS AND DEPENDENCIES

User Base Growth: The app's success is dependent on active user growth and engagement.
Ingredient Database: Access to a comprehensive and updatable ingredient and recipe database.
Technology Adoption: Assumption that users have access to mobile devices or computers with internet.

3. SYSTEM FEATURES

Functional Requirements	Non-functional requirements
F-01: User Registration and Profile Management	NF-PE: Performance
F-02: Recipe Discovery and Management	NF-US: Usability
F-03: Search and Filtering	NF-REL: Reliability
F-04: Community Engagement and Content Sharing	NF-SC: Scalability
F-05: Rewards System	NF-SE: Security
F-06: Notifications	NF-CO: Compliance
	NF-SU: Supportability

3.1. SF 1: RECIPE DISCOVERY ENGINE

This discovery engine is the core feature of the app, since it allows users to input the ingredients and obtain recipe suggestions that would be feasible for the ingredients in their fridges or pantries.
Functional Requirements:

- The engine must allow users to enter multiple ingredients as search criteria.
- The engine must provide recipe suggestions that utilize the inputted ingredients.
- The engine should offer the option to save favorite recipes for later reference.

3.2. SF 2: USER ACCOUNT MANAGEMENT

This management handles user registration, modifications and profile customization, and is in charge of data manipulation within the app.
Functional Requirements:

- Users must be able to create a new account using an email address.
- Users must be able to edit their profile information, preferences and allergies.
- The system must provide a secure method for users to delete their accounts and related data.

3.3. SF 3: DIETARY AND CUISINE FILTERING

This filter feature lets users apply specific filters to the recipe searches, catering to various dietary needs, like allergies or low-fat recipes, and cuisine preferences.
Functional Requirements:

- Users should be able to filter recipes by dietary restrictions such as vegetarian, vegan, gluten-free, etc.
- Users should be able to filter recipes based on dietary needs related to medical conditions such as allergies to a certain type of food or other diagnosis.
- Users should be able to filter recipes based on cuisine types; for example: Mediterranean, Asian, American, etc.
- The system must update the recipe suggestions in real-time as filters are applied.

3.4. SF 4: COMMUNITY ENGAGEMENT PLATFORM

The Community Engagement Platform encourages interaction among users through forums, recipe sharing, and social features.

Functional Requirements:

- Users must be able to post and share their own recipes with the community.
- The platform should provide a forum where users can discuss cooking techniques and ingredients.
- Integration with social media should be facilitated for sharing content outside the app, or sharing content from other media into the forum.

3.5. SF 5: REWARDS AND INCENTIVES

The Rewards and Incentives feature aims to enhance user engagement by offering rewards for various in-app activities.

Functional Requirements:

- The system must track user activity points earned by sharing recipes or participating in community discussions.
- Users should be able to redeem points for in-app features or content.
- The system should display the user's rewards status and history

3.6. SF 6: REAL-TIME NOTIFICATIONS

Real-Time Notifications keep users informed about updates, new recipes, and community activity.

Functional Requirements:

- Users must receive notifications for new recipes based on their preferences.
- The system must alert users about updates to the app or their community interactions.
- Users should have control over the types of notifications they receive.

4. EXTERNAL INTERFACE REQUIREMENTS

4.1. USER INTERFACES
4.2. HARDWARE INTERFACES
4.3. SOFTWARE INTERFACES
4.4. COMMUNICATIONS INTERFACES

5. OTHER NONFUNCTIONAL REQUIREMENTS

5.1. PERFORMANCE REQUIREMENTS

Response Time: The app should respond to user requests (e.g., searching for recipes, uploading recipes) within 2 seconds.

Load Times: The app's main screens (e.g., recipe search, user profile) should load in under 5 seconds.
Concurrent Users: The app should support a large number of users concurrently without significant performance degradation. Aim for at least 5,000 to 10,000 concurrent users.

Data Processing: The app should handle large datasets (e.g., thousands of recipes) efficiently and without performance issues

5.2. SAFETY REQUIREMENTS

Data Safety: Ensure user data (personal information) is protected against loss, corruption, or unauthorized access.

5.3. SECURITY REQUIREMENTS

Data Encryption: All sensitive data (such as user information) should be encrypted during transmission and storage.

User Authentication: Implement secure user authentication mechanisms, such as two-factor authentication (2FA) or OAuth.

Compliance with Privacy Regulations: The app must comply with privacy laws such as GDPR or CCPA, ensuring user data is handled in accordance with these regulations.

5.4. MAINTAINABILITY REQUIREMENTS

Modular Design: The app should be designed with modularity to facilitate maintenance and feature updates.

Code Quality: Follow coding standards and best practices to ensure maintainability.

Documentation: Provide comprehensive documentation for developers, including architecture diagrams, API documentation, and code comments.

Automated Testing: Implement automated tests to ensure the system is testable and maintainable over time.

5.5. SCALABILITY REQUIREMENT

5.6. SOFTWARE QUALITY ATTRIBUTES

6. OTHER REQUIREMENTS