SOFTWARE REQUIREMENTS SPECIFICATION

RECIPE ROULETTE

Software Engineering

Bachelor's Degree in Bioinformatics

Course 2023-24

(Version 0.3)
(30-04-24)

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
María/Alessandra/Carmen	26/04/2024	Implementation of revision	0.2
María/Alessandra/Carmen	30/04/2024	Improvement of structure and workflow	0.3

II. TABLE OF CONTENTS

1. PROJECT DOMAIN

1.1. PURPOSE

The purpose of the "Recipe Roulette" app is to provide a user-friendly platform that provides a 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. TARGET MARKET

This Software Requirements Specification (SRS) document outlines the functional and nonfunctional 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 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

Nearly a third of all the food (around 1.3 billion tonnes of food) produced each year is squandered or lost before it can be consumed. All this food produced but never eaten would be sufficient to feed two billion people. That's more than twice the number of undernourished people across the globe!
"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. It provides a 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

- https://www.wfp.org/stories/5-facts-about-food-waste-and-hunger
- Links to video recipes informing tips or visualizing the way to cook.
- IEEE Recommended practice for SRS

2. SYSTEM REQUIREMENTS

Functional Requirements	**Non-functional requirements**
UFR-1: User registration	NFR-001-PE-: Performance
UFR-2: User Login/Authentication	NFR-002-US: Usability
UFR-3: Delete Profile	NFR-003-REL: Reliability
UFR-4: Search Recipes by Ingredients	NFR-004-MT: Maintainability
UFR-5: Search Recipes by Dish Name	NFR-005-SE: Security
UFR-6: Filter Recipes	NFR 006-SC: Scalability
UFR-7: Upload, Modify, Delete Recipes	NFR-007-SU: Software Quality Attributes
UFR-8: Gain Points	
UFR-9: Check Reward Status	
MFR-10: Management of Software	
CFR-11: Approve recipe	
CFR-12: Send Notification	
CFR-13: Delete recipe	
CFR-14: Revise Content	

2.1. FUNCTIONAL REQUIREMENTS

Functional Req. ID #	**Actor**	**Functional Name**	**Functional Requirement Description**
UFR-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.
UFR-2	User	User Login/ Authentication	The user should be able to login the
UFR-3	User	Delete Profile	Users should have the option to delete their profile and all associated data from the app.
UFR-4	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.

UFR-5	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.
UFR-6	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.
UFR-7	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.
UFR-8	User	Gain Points	Users should earn points for sharing recipes on the platform. These points can be used for rewards.
UFR-9	User	Check Reward Status	Users should be able to check their current reward status, including the number of points earned and available rewards.
MFR-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 statististics ● Promote the software (marketing) ● Check the security ● Approve new delivery staff accounts (check documents id (photo) and a pdf CV)
CFR-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.
CFR-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.
CFR-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.
CFR-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.

2.2. NON-FUNCTIONAL REQUIREMENTS

2.2.1. 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

2.2.2. 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.

2.2.3. SECURITY REQUIREMENTS

Data Encryption: All sensitive data (such as user information) should be encrypted during transmission and storage. Ensure user data (personal information) is protected against loss, corruption, or unauthorized access.

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.

2.2.4. SCALABILITY REQUIREMENT

Horizontal Scalability: The system should support horizontal scaling to accommodate increased traffic and usage.

Vertical Scalability: The system should support vertical scaling to increase server capacity as needed.

Load Balancing: Implement load balancing to distribute traffic evenly and ensure scalability as the user base grows.

2.2.5. SOFTWARE QUALITY ATTRIBUTES

Usability: The app should have an intuitive user interface with easy navigation and clear instructions. Ensure usability through user testing and feedback.

Reliability: The app should be reliable, with minimal downtime and consistent operation.

Testability: Implement a robust testing framework to ensure the app is testable and maintainable.

Interoperability: The app should be interoperable with other systems, such as external delivery tracking and payment gateways.

2.3. EXTERNAL INTERFACE REQUIREMENTS

Description of the interfaces between the software and the world including user, ~~hardware,~~ software, and communication interfaces.

2.3.1. USER INTERFACES

App that should be compatible on both Android and IOS as well as all major web browsers for the web interface, so as to make the app available for all kinds of mobile devices.
The features should be able easy to read and use and to accommodate for all kinds of needs (different font or languages)

2.3.2. SOFTWARE INTERFACES

Our database would have all the input recipes users put as well as external recipes if possible.
We could integrate social medial so as to share recipes and attract more users

2.3.3. COMMUNICATIONS INTERFACES

App notifications to follow up on progress for each user and propose new recipes
Email for communication between app services and confirming accounts.

2.4. 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.

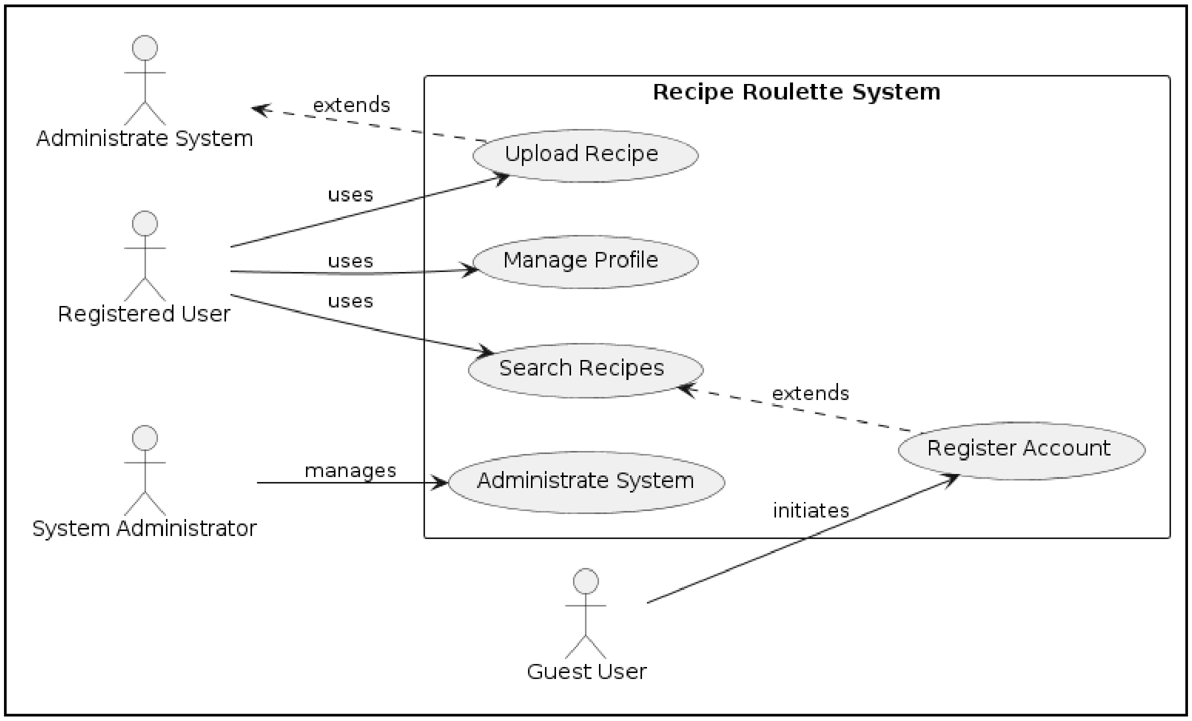
3. USE CASE DIAGRAM

3.1. OVERVIEW OF USE CASES

The use case model serves as a key tool in understanding and documenting how users interact with the "Recipe Roulette" system. This model helps to identify and define the roles of various actors and the fundamental processes they perform within the application. It provides a framework for analyzing the functional requirements by showcasing the sequences of actions between the user and the system to achieve specific goals. Through use cases, we capture user interactions that are essential for the system to fulfill its functionalities, ensuring that all user expectations are met systematically.

3.2. DETAIL USE CASE DIAGRAM

Below is the detailed use case diagram for the "Recipe Roulette" application. This visual representation includes all user interactions, system responses, and the relationships between different use cases. The diagram shows how various features of the system interconnect and interact with the users:



The diagram includes several actors (e.g., Registered User, Guest User, System Administrator) and use cases (e.g., Register Account, Search Recipes, Upload Recipe, Manage Profile).

3.3. USE CASE DESCRIPTIONS

Each use case identified in the diagram above is described in detail below, outlining the system's response to user actions:

UC1: Register Account
Actor: Guest User
Description: Allows a guest user to register and create a personal account within the application.
Preconditions: The user must have valid personal information and access to the internet.
Postconditions: The user account is created and the user can log in to access personalized features.

Basic Flow:

1. The user selects the "Register" option.
2. The user fills in the required information (e.g., name, email, password).
3. The system validates the information and creates a new user account.
4. The system confirms account creation and prompts the user to log in.

Alternative Flows:

- If the user enters an email that is already in use, the system prompts to try a different email.
- If the user provides incomplete information, the system displays an error and requests complete details.

UC2: Search Recipes
Actor: User (Registered or Guest)
Description: Allows users to search for recipes based on various criteria (ingredients, cuisine, dietary restrictions).
Preconditions: None.
Postconditions: Recipes matching the criteria are displayed.

Basic Flow:

1. The user accesses the search feature.
2. The user inputs search criteria and submits the search.
3. The system retrieves and displays recipes matching the criteria.

Alternative Flows:

- If no recipes match the criteria, the system displays a message indicating no results found and suggests altering search parameters.

UC3: Upload Recipe
Actor: Registered User
Description: Allows registered users to upload their own recipes into the database.
Preconditions: User must be logged in.
Postconditions: The recipe is added to the database and available for others to view.

Basic Flow:

1. The user navigates to the "Upload Recipe" section.

2. The user enters the details of the recipe (name, ingredients, steps) and submits.
3. The system validates the entered data and adds the recipe to the database.
4. The system confirms the successful addition of the recipe.

Alternative Flows:

- If mandatory fields are missing, the system prompts the user to complete all required fields before submission.

4. SYSTEM ANALYSIS

4.1. 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.

4.2. SYSTEM FEATURES

4.2.1. 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.

4.2.2. 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.

4.2.3. 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.

4.2.4. 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.

4.2.5. 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.

4.2.6. 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.

4.3. OPERATING ENVIRONMENT

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

4.4. 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.

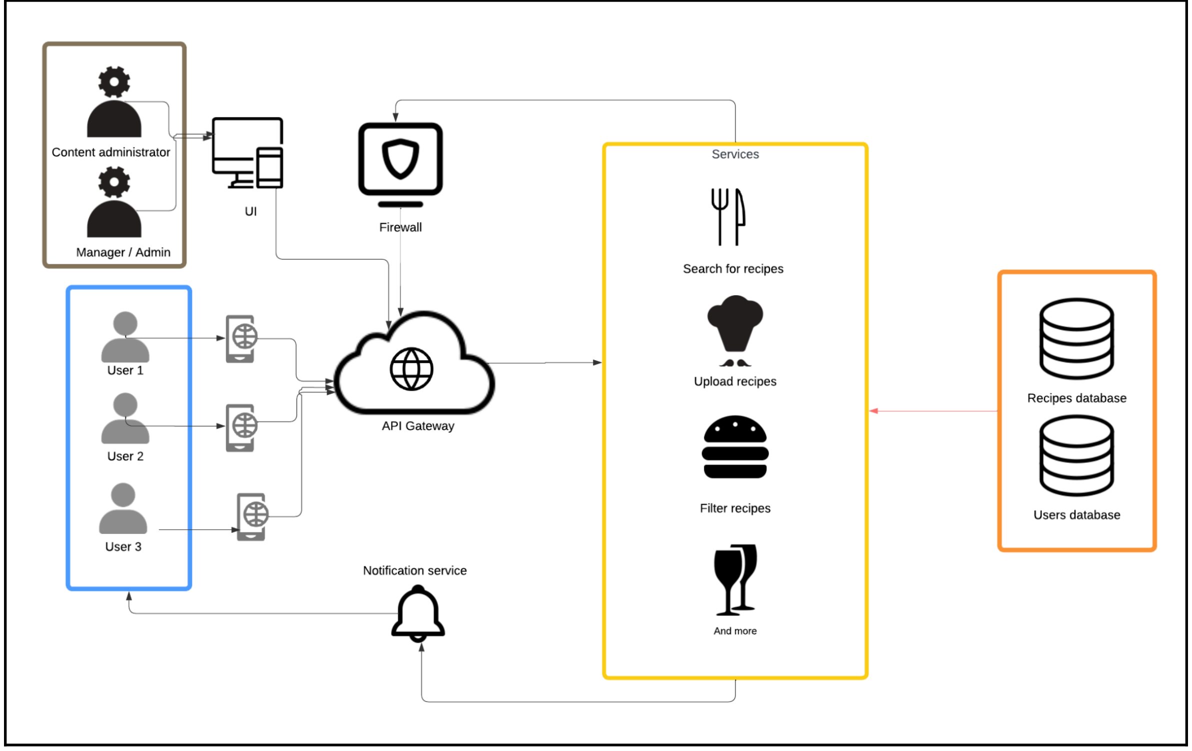
5. DESIGN

5.1. ARCHITECTURAL OVERVIEW

High-level architecture showing major system components and their interactions.

5.2. INTERFACE DESIGN

Architecture Diagram



Detailed designs for user interfaces, including mockups and navigation flows.

5.3. DATABASE AND DATA FLOW DESIGN

Description of database schemas and data flow diagrams.

5.4. SECURITY PROTOCOLS

Security measures and protocols to protect data and ensure privacy.