Software Requirements Specification

for

Mercedario Recetas

Version 1.0 approved

Prepared by Nicolas Federico Delgado Pérez

Universidad Cooperativa de Colombia

30/08/23

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Nicolas Delgado	07/09/23	Added contents within document.	1.0

1. Introduction

1.1 Purpose

The software requirements specified in this document pertain to the development of the "Mercedario Recetas" application for the Mercedario restaurant. This document outlines the functional and non-functional requirements for the application. The purpose of this software is to provide a comprehensive solution for managing recipes and dishes offered by the restaurant. It aims to streamline the recipe management process, enhance administrative control, and facilitate preparation planning for various quantities of servings.

1.2 Document Conventions

This Software Requirements Specification (SRS) adheres to standard documentation conventions. Priorities for higher-level requirements are assumed to be inherited by detailed requirements. Each requirement statement in this document is assigned its own priority level for clarity and reference.

1.3 Intended Audience and Reading Suggestions

This document is intended for various stakeholders involved in the development and usage of the "Mercedario Recetas" application. The primary audience includes:

- Developers: To understand the technical specifications and functional requirements of the application.
- Project Managers: To oversee the project and ensure it aligns with the specified requirements.
- Administrators: To comprehend their role and privileges within the application.
- Users: To gain insight into the capabilities and features available to them.
- Testers: To develop test cases based on the requirements.
- Documentation Writers: To assist in creating user manuals and documentation.

Readers are encouraged to start with the overview sections and then proceed to the relevant sections based on their roles and interests. The document is organized to provide a clear understanding of the software's functionality, user roles, and technical specifications.

1.4 Product Scope

The "Mercedario Recetas" application is designed to support the Mercedario restaurant in managing its recipes and dishes efficiently. The scope of the product includes:

- User Authentication: Users can log in with a username and password.
- Recipe Management: Administrators can view, add, edit, and delete recipes. They can specify details such as name, preparation time, serving size, ingredients, and preparation instructions.
- Ingredient Management: Administrators can maintain a list of common ingredients, including details such as name, unit, unit cost, source, and calorie content.

- Preparation Planning: Administrators can plan recipe preparation for a specific number of people, calculate nutritional information, ingredient quantities, and estimated costs.
- High Availability: The system should be available for use between 7 am and 8 pm from Monday to Friday.
- Security: The system ensures user identity and data confidentiality.
- Performance: The system should handle between 30 and 100 operations per minute.

The application's primary goal is to streamline recipe management, reduce manual efforts, and enhance the overall efficiency of the restaurant's kitchen operations.

1.5 References

This Software Requirements Specification (SRS) follows all the necessary information to define the requirements for the "Mercedario Recetas" application precised on the following document.

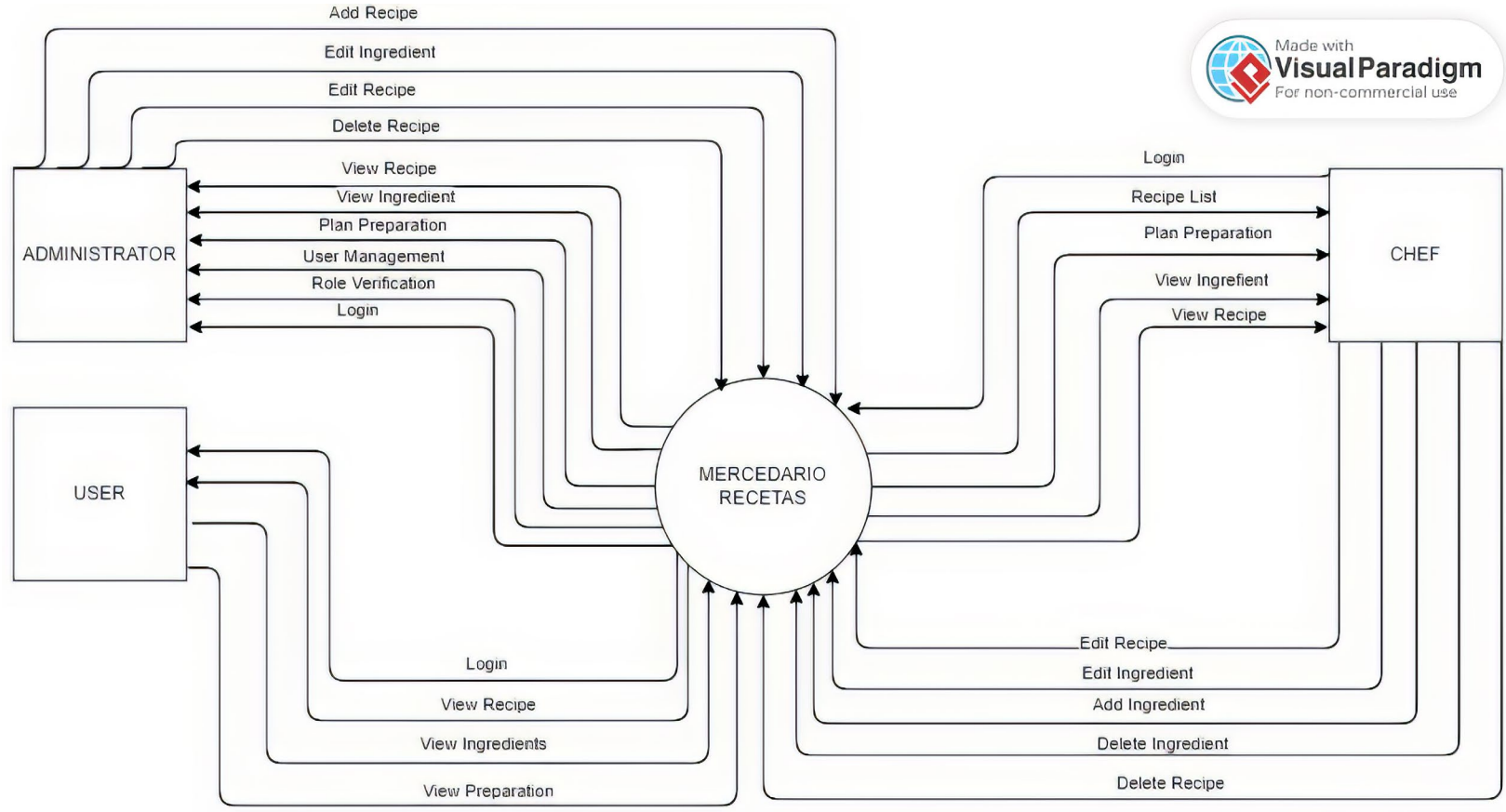
- Project Statement (Universidad Cooperativa de Colombia) https://campusvirtual.ucc.edu.co/d2l/le/enhancedSequenceViewer/579534?url=https%3A%2F%2Fa07daa8f-b6be-4652-b09b-4c1708a52dd7.sequences.api.brightspace.com%2F579534%2Factivity%2F6509922%3FfilterOnDatesAndDepth%3D1

2. Overall Description

2.1 Product Perspective

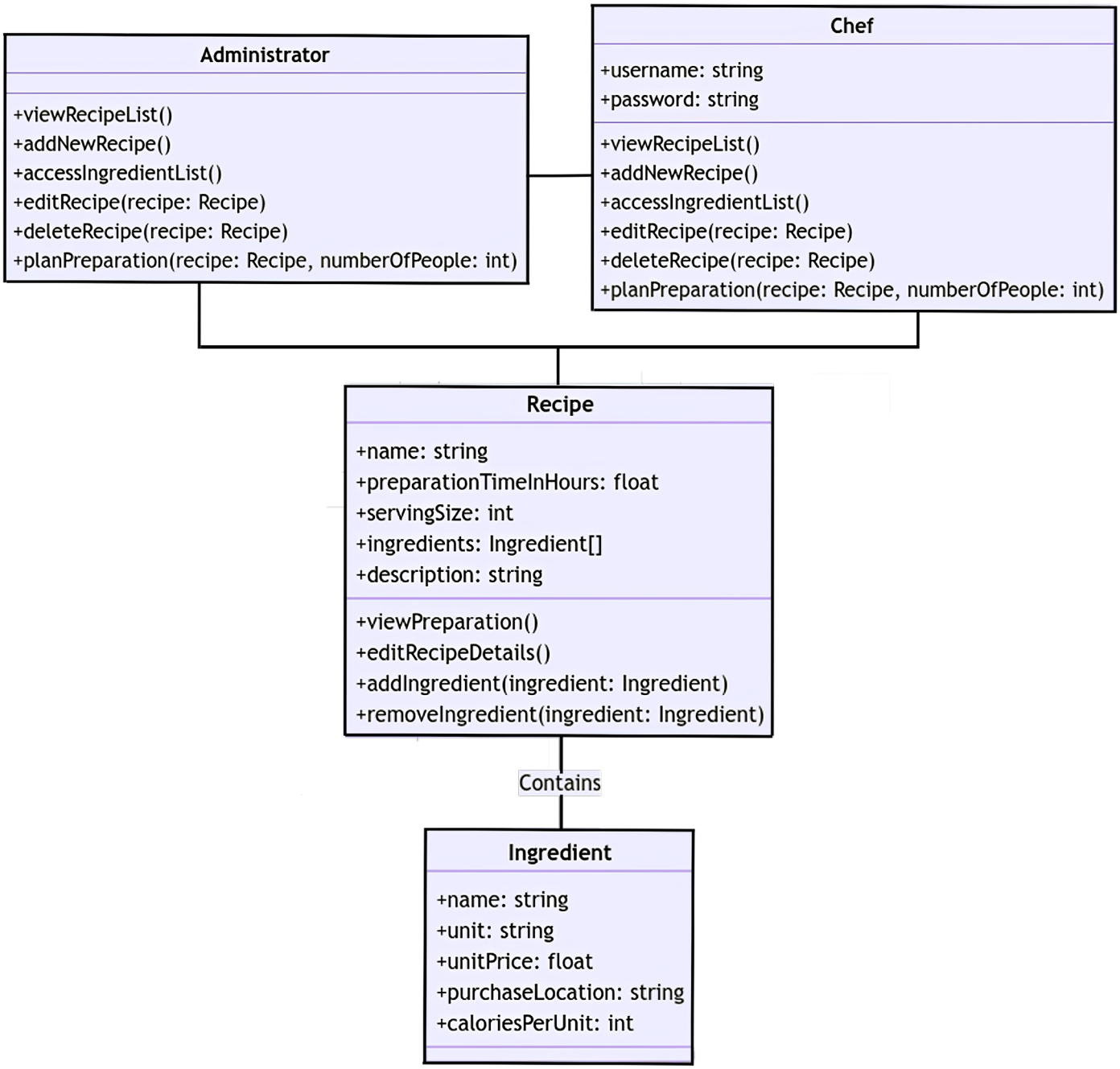
The "Mercedario Recetas" application stands as an independent and self-contained solution tailored specifically for the Mercedario restaurant. It does not belong to an existing product family, nor does it serve as a replacement for any existing systems. This application is entirely novel, conceived to empower the restaurant with efficient tools for the comprehensive management of its recipes and dishes.

This software operates within the context of the restaurant's daily operations, offering a suite of functions that encompass recipe management, ingredient handling, and preparation planning. It does not rely on or interface with any larger systems and does not carry significant external dependencies. The following context diagram showcases this:


Add Recipe
Edit Ingredient
Edit Recipe
Delete Recipe
View Recipe
View Ingredient
Plan Preparation
User Management
Role Verification
Login
ADMINISTRATOR
USER
MERCEDARIO RECETAS
Login
Recipe List
Plan Preparation
View Ingrefient
View Recipe
CHEF
Edit Recipe
Edit Ingredient
Add Ingredient
Delete Ingredient
Delete Recipe
Login
View Recipe
View Ingredients
View Preparation
Made with VisualParadigm
For non-commercial use

2.2 Product Functions

The product boasts a rich array of functions, catering to the diverse needs of the restaurant staff. These functions include:



2.3 User Classes and Characteristics

The "Mercedario Recetas" application caters to two primary user classes:

1. **Restaurant Administrator:** This user class possesses an in-depth understanding of the restaurant's operations and technical expertise. Administrators have full access to the system and can perform tasks such as managing recipes, ingredients, and planning preparations.

2. **Chef:** These users have access to view and manage recipes and preparation information and possess permissions to make edits or plan preparations. They require a high level of understanding about the restaurant operations and some technical proficiency.

2.4 Operating Environment

The software will operate within the following environment:

- **Hardware Platform:** The application will run on a device that meet the specified technical requirements.

- **Operating System:** It will be compatible with operating systems that are supported by the chosen technology stack.

- **Additional Software:** The software must coexist harmoniously with any other software essential for the restaurant's operation, such as point-of-sale systems or other management tools.

2.5 Design and Implementation Constraints

Design and implementation are subject to certain constraints, including:

- **Policy and Regulatory Compliance:** The software must adhere to corporate policies and relevant regulations.

- **Hardware Limitations:** The system must work within specified hardware constraints, including timing and memory requirements.

- **Interfaces:** If required, the application must interface effectively with other systems.

- **Technology and Tools:** Specific technologies, tools, and databases specified by the client must be utilized.

- **Security:** The application must adhere to data security and privacy regulations.

- **Design Conventions and Standards:** Design conventions and programming standards, as specified, must be followed during development.

This software is pretended to be executed in console and written in python. Main restriction to implement the above-mentioned design implementations.

2.6 User Documentation

User documentation to be delivered alongside the software will encompass user manuals, online help resources, and tutorials. The exact formats and delivery standards for this documentation will be determined in later project stages.

2.7 Assumptions and Dependencies

The following assumptions are considered during the development and operation of the "Mercedario Recetas" application:

1. **User Roles and Accessibility**:
    - The program assumes that it will be used by all restaurant staff responsible for food preparation, including chefs, ingredient distributors, kitchen assistants, and others. The program is designed to accommodate common employees as regular users without special permissions.

2. **Ingredient Management**:
    - The program depends on the ingredient distributor's role to accurately input the quantity and type of ingredients entering the restaurant. To ensure that the program accurately reflects available ingredients, it is recommended that multiple users have the ability to edit the ingredient section of the program.

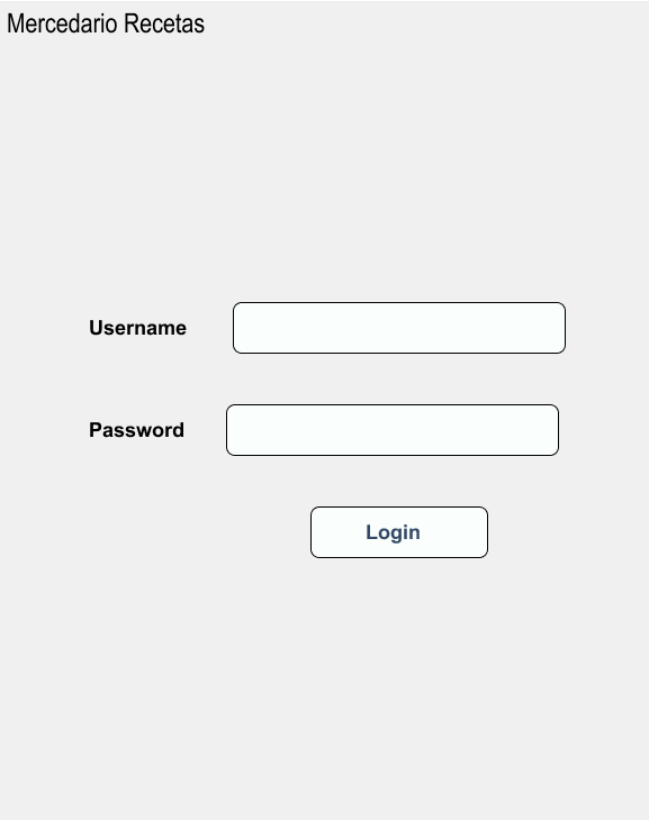
3. **Ingredient Usage in Recipes**:
    - It is assumed that each recipe will use whole numbers of ingredients, and there won't be records of portions or fractions of an ingredient already used. In other words, it is not expected that leftover or recycled portions of an ingredient will be accounted for in a recipe.

3. External Interface Requirements

3.1 User Interfaces

- All screens will have a consistent light gray color scheme to enhance readability and maintain a cohesive visual identity.
- All screens will include a back button or an option to return to the previous menu, allowing users to navigate easily.
- All screens will maintain a responsive design, adjusting to different terminal sizes for a seamless user experience.
- All screens will adhere to a standardized format for displaying lists, ensuring consistency when presenting recipes, ingredients, or other data.
- All screens will use easily understandable language in prompts and messages to facilitate user comprehension.
- All screens will incorporate features for easy data entry, editing, and deletion, following a user-friendly format.

- All screens will prioritize accessibility, allowing users with varying levels of technical expertise to interact with the application effectively.



3.2 Hardware Interfaces

Hardware interfaces are not relevant to this instance.

3.3 Software Interfaces

Hardware interfaces are not relevant to this instance.

3.4 Communications Interfaces

This software application does not communicate in between interfaces.

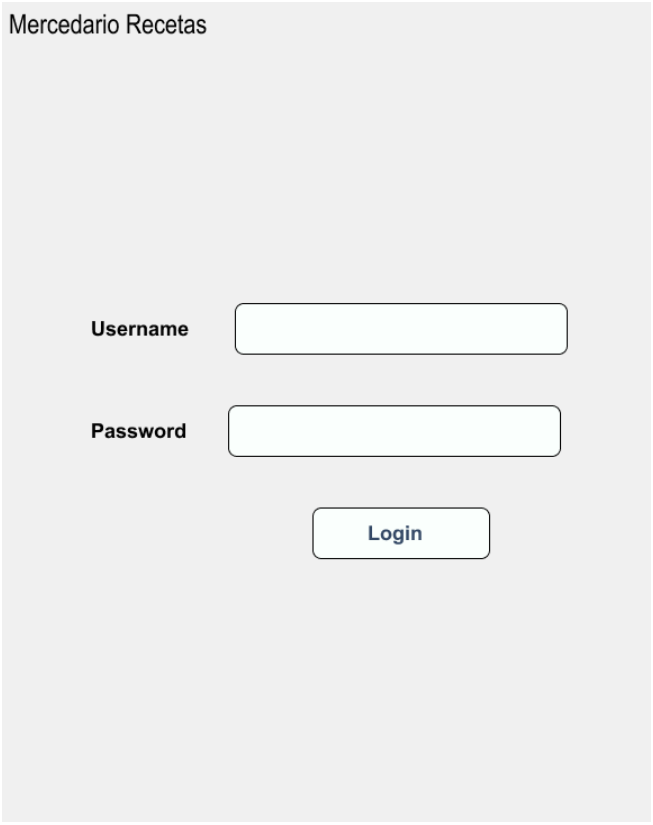
4. System Features

This section contains the functional requirements for the product by system features, the major services provided by the product. This section is organized by each requirements features, including user stories and mockups.

Functional Requirement 4.1: User Authentication

4.1.1: User Login

- **User Story:** As a user, I want to log in using my username and password.
- **Acceptance Criteria:**
    1. The login screen provides input fields for entering a username and password.
    2. On successful login, the user is redirected to the main menu.
    3. On unsuccessful login, an error message is displayed.
    4. Passwords are securely stored and transmitted.



Functional Requirement 4.2: Recipe Management

4.2.1: View Recipes

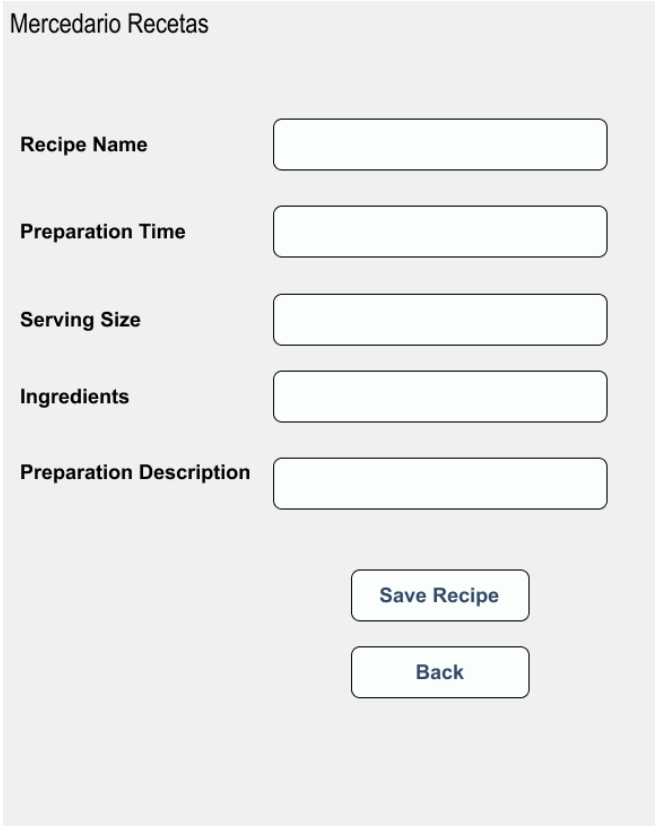
- **User Story:** As an administrator, I want to view a list of all registered recipes.
- **Acceptance Criteria:**

1. The system displays a list of all registered recipes with their details.

4.2.2: Add Recipe

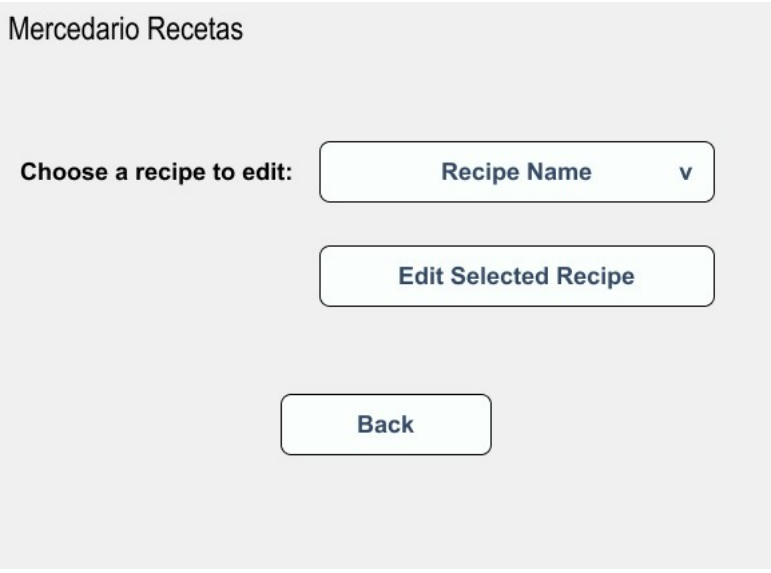
- **User Story:** As an administrator, I want to add a new recipe, specifying its name, preparation time, serving size, ingredients, and preparation description.
- **Acceptance Criteria:**
    1. The "Add Recipe" interface allows the administrator to input the name, preparation time, serving size, ingredients, and preparation description.

2. After adding, the new recipe is listed among the recipes.



4.2.3: Edit Recipe

- **User Story:** As an administrator, I want to edit an existing recipe, including changing any of its attributes and adding or removing ingredients.
- **Acceptance Criteria:**
    1. The "Edit Recipe" interface allows the administrator to change any attributes of an existing recipe.



2. Ingredients can be added or removed as needed.
3. After editing, the updated recipe is reflected in the list.

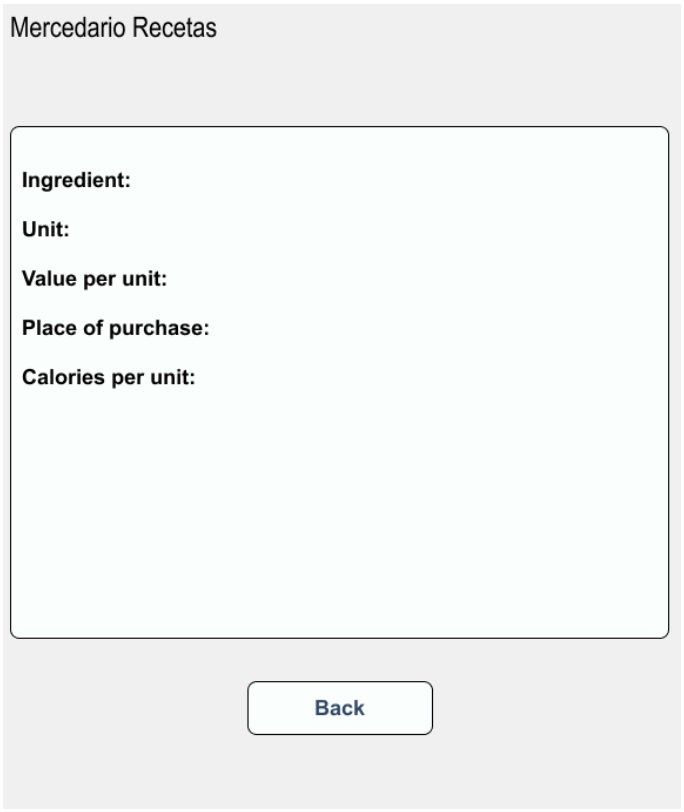
4.2.4: Delete Recipe

- **User Story:** As an administrator, I want to delete a recipe.
- **Acceptance Criteria:**
    1. The system allows the administrator to delete a recipe.
    2. Once deleted, the recipe is removed from the list.

Functional Requirement 4.3: Ingredient Management

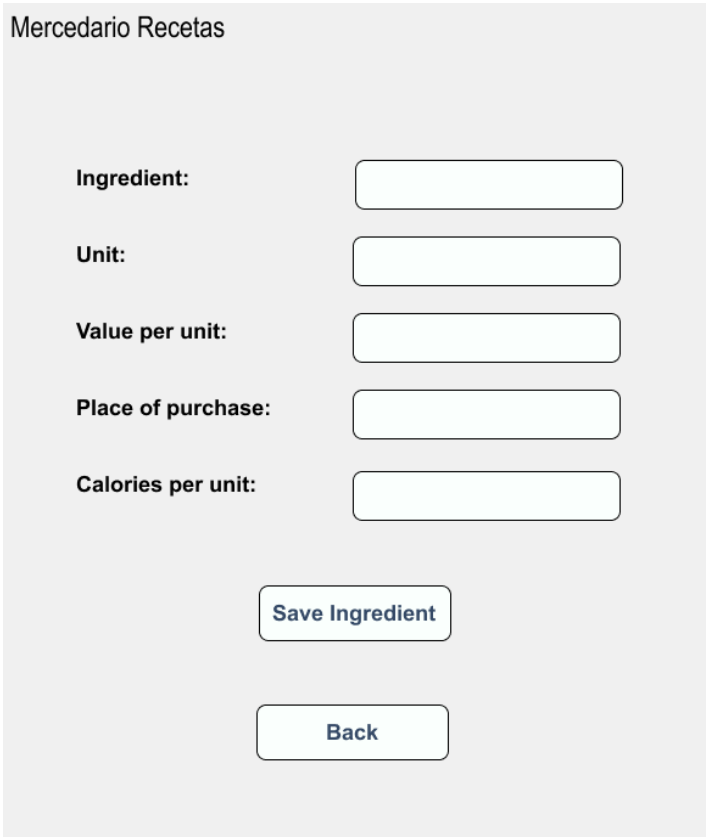
4.3.1: View Ingredients

- **User Story:** As an administrator, I want to view a list of all ingredients.
- **Acceptance Criteria:**
    1. The system displays a list of all registered ingredients with their details.



4.3.2: Add Ingredient

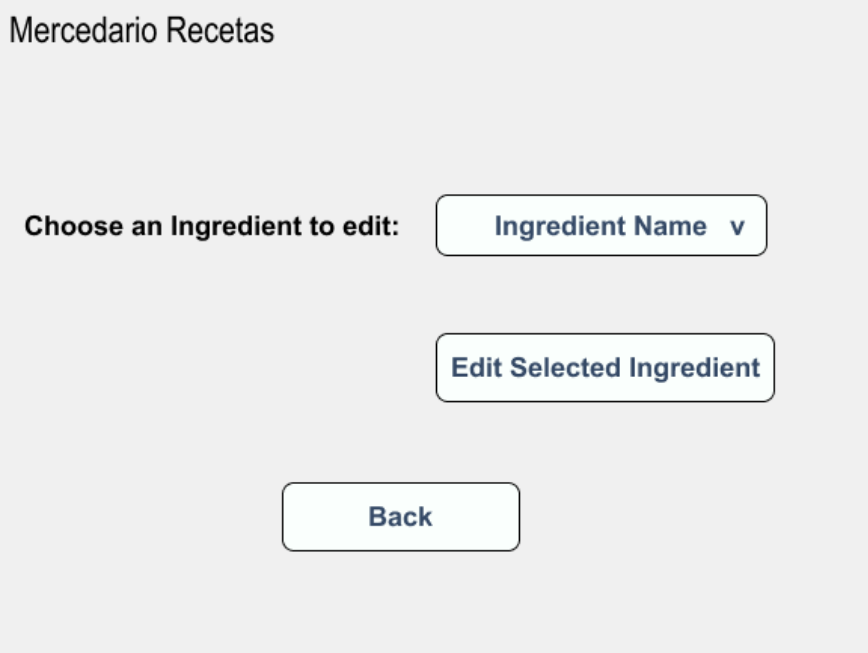
- **User Story:** As an administrator, I want to add a new ingredient, specifying its name, unit, value, source, and calorie content.



- **Acceptance Criteria:**
    1. The "Add Ingredient" interface allows the administrator to input the name, unit, value, source, and calorie content of a new ingredient.
    2. After adding, the new ingredient is listed among the ingredients.

4.3.3: Edit Ingredient

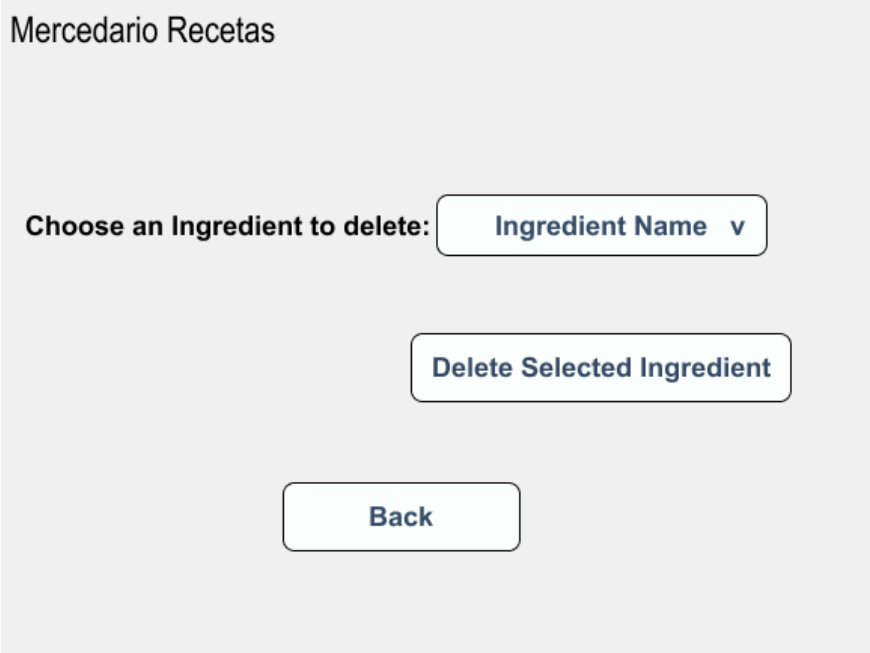
- **User Story:** As an administrator, I want to edit an existing ingredient.
- **Acceptance Criteria:**
    1. The "Edit Ingredient" interface allows the administrator to change any attributes of an existing ingredient.
    2. After editing, the updated ingredient is reflected in the list.



4.3.4: Delete Ingredient

- **User Story:** As an administrator, I want to delete an ingredient, but only if it's not used in any recipe.
- **Acceptance Criteria:**

1. The system allows the administrator to delete an ingredient, but only if it's not being used in any recipe.
2. If an ingredient is used in a recipe, it cannot be deleted, and an error message is displayed.



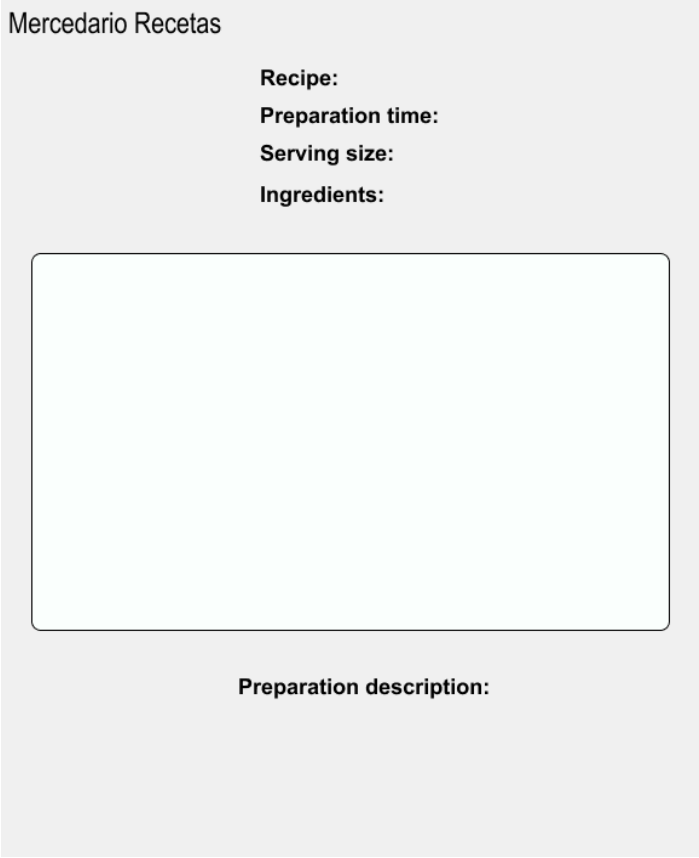
Functional Requirement 4.4: Recipe Preparation

4.4.1: Plan Recipe

- **User Story:** As an administrator, I want to plan the preparation of a recipe for a specific number of people.
- **Acceptance Criteria:**
    1. The "Plan Recipe" interface allows the administrator to specify the number of people for the preparation.
    2. The chosen recipe is displayed with the specified number of people.

4.4.2: Calculations

- **User Story:** The system should calculate the required amount of ingredients, preparation time, serving size, preparation details according to the data provided.
- **Acceptance Criteria:**
    1. The system accurately calculates the required amount of ingredients, preparation time, serving size, preparation details. Results are displayed to the administrator.



Functional Requirement 4.5: User Management

4.5.1: Add User

User Story: As an administrator, I want to add a new user.

- **Acceptance Criteria:**

1. The "Add User" interface allows the administrator to input user information.
2. After adding, the new user is registered in the system.

4.5.3: Delete User

User Story: As an administrator, I want to delete a user.

- **Acceptance Criteria:**

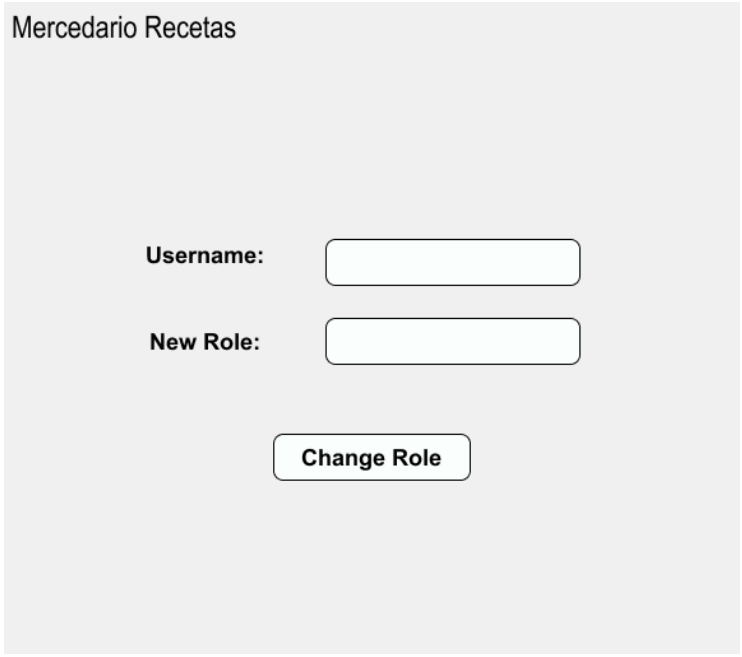
1. The system allows the administrator to delete a user from the system.
2. Users cannot delete themselves.
3. After deletion, the user is removed from the system.

4.5.4: Change User Roles

User Story: As an administrator, I want to change user roles.

- **Acceptance Criteria:**

1. The administrator can change user roles within the system.
2. Role changes are reflected accurately in the system.



5. Other Nonfunctional Requirements

5.1 Identified Non-Functional Requirements

1. **NFSR-01: Access Security**
    - Description: The system must guarantee the identity of the users that enter and only allow the administrator to make edits to the registered information.
2. **NFSR-02: Data Confidentiality**
    - Description: The system must guarantee the confidentiality of data transmitted between client machines and the device, ensuring that sensitive information is not intercepted or compromised.
3. **NFSR-03: Performance**
    - Description: The system must be able to process between 30 and 100 operations per minute, ensuring optimal performance even at peak load times.
4. **NFSR-04: Data Integrity**
    - Description: All operations performed on the system must be successful and ensure the integrity of the stored data. No data loss or corruption should occur.
5. **NFSR-05: Usability**
    - Description: The user interface should be intuitive and easy to use for the administrator. Actions such as adding, editing, and deleting recipes or ingredients should be clear and understandable.
6. **NFSR-06: Time Availability**
    - Description: The system must be highly available for operations, specifically between 7 am and 8 pm from Monday to Friday. This ensures that the system is accessible during the designated time frame for restaurant-related activities.

5.2 Significant architectural requirements

1. **NFSR-01: Access Security**

- The requirement for access security implies the need for robust authentication and authorization mechanisms. This has a significant impact on the system's architecture as it involves designing and implementing secure user authentication and access control features.

2. **NFSR-02: Data Confidentiality**

- Ensuring data confidentiality requires the implementation of encryption and secure data transmission mechanisms. This has a significant impact on the architecture as it involves incorporating encryption protocols and secure communication channels.

3. **NFSR-03: Performance**

- Meeting performance requirements, especially the ability to handle 30 to 100 operations per minute, is a significant architectural consideration. This may involve designing a scalable and efficient system architecture, including considerations for load balancing and optimization.

4. **NFSR-04: Data Integrity**

- Ensuring data integrity involves designing the system architecture to handle transactions reliably and securely. This includes mechanisms for data validation, error handling, and transaction management.

5. **NFSR-05: Usability**

- While usability is more of a user experience aspect, it can still impact the system architecture, particularly the design of the user interface and how the components interact to provide a seamless and intuitive experience for the administrator.

6. **NFSR-06: Time Availability**

- Time availability requirements necessitate a robust architectural design to ensure the system is highly available during the specified time frame. This may involve redundant systems, failover mechanisms, and load balancing to meet availability goals.

5.2.1 Quality Scenario

ID: NFSR-01	Access Security	Version	1.0
Source	**Stimulus**	**Artifact**	**Environment**
User	Attempted login with username/password	System (Authentication and Authorization components)	Production Environment
Response		**Response Measure**	
Grant/Deny (Successful login or access denial)		Successful login or access denial in less than 2 seconds.	

ID: NFSR-02	Data Confidentiality	Version	1.0
Source	**Stimulus**	**Artifact**	**Environment**
Data transmission processes	Data exchange or transmission events	System components for encryption and secure communication.	Production Environment.
Response		**Response Measure**	
Implementation of encryption and secure communication.		Assurance of confidential data transmission between user and system the whole time.	

ID: NFSR-03	Performance	Version	1.0
Source	**Stimulus**	**Artifact**	**Environment**
System load and user interaction.	User requests, transactions, and overall system activity.	System architecture, including components for load balancing and optimization	Production Environment, where the system must handle varying loads efficiently.
Response		**Response Measure**	
Designing a scalable and efficient system architecture to meet the performance requirement of handling 30 to 100 operations per minute.		System response time within a range of 1 to 5 seconds, ensuring that operations are processed efficiently	

ID: NFSR-04	Data Integrity	Version	1.0
Source	**Stimulus**	**Artifact**	**Environment**
System data transactions	Events triggering data transactions	System architecture with data integrity mechanisms	Production Environment
Response		**Response Measure**	
Robust architecture for data validation, error handling, and transaction management.		Successful data transactions within a response time range of 0.5 to 2 seconds	

ID: NFSR-05	Usability	Version	1.0
Source	**Stimulus**	**Artifact**	**Environment**
Administrator interactions.	Admin actions	User interface for admin tasks.	Production Environment.
Response		**Response Measure**	
System architecture prioritizing admin-friendly UI.		Admin task completion time within 5 to 15 seconds, ensuring efficient and user-friendly interactions.	

ID: NFSR-06	Time Availability	Version	1.0
Source	**Stimulus**	**Artifact**	**Environment**
User	User interactions	Architectural components for high availability.	Production Environment during the designated time frame.
Response		**Response Measure**	
Robust design for high availability.		System uptime of 99.9% during 7 am to 8 pm, Monday to Friday, ensuring continuous availability for user interactions.	

5.2.2 Utility Tree