Version overview

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author | Comment |
| 21.10.2024 | 1 | Gozman-Pop Maria-Eliza  Borzan Călina-Annemary | Start exploring the general functionality of the project, write the introduction part and establish the actors. |
| 4.11.2024 | 2 | Gozman-Pop Maria-Eliza  Borzan Călina-Annemary | Write in detail about each actor, draw the use case diagrams and describe all use cases. |
| 10.11.2024 | 3 | Gozman-Pop Maria-Eliza  Borzan Călina-Annemary | Review functionalities, small changes to the descriptions and update glossary terms. |

System requirements for < Project Title>

[Introduction 2](#_Toc181610559)

[1 General view of actors and use cases 2](#_Toc181610560)

[2 Detailed description of actors and use cases 2](#_Toc181610561)

[2.1 Actors 2](#_Toc181610562)

[2.1.1 Administrator 2](#_Toc181610563)

[2.1.2 User 3](#_Toc181610564)

[2.2 Use cases 4](#_Toc181610565)

[*2.2.1* Use Case Diagram for Administrator 4](#_Toc181610566)

[2.2.2 Use Case Diagram for User 4](#_Toc181610567)

[3 Glossary – used terms 5](#_Toc181610568)

# Introduction

The project presented in this document is *YumCycle*, a fridge contents management application. This mobile app is designed around helping people of all ages and places in managing their fridge leftovers. The idea has started in our own dorm room, where every week, we buy a lot of unnecessary groceries and always end up throwing some of them away. This is how this project was born, we realized we could optimize the way we handle our aliments and also find new recipes to cook and eat healthier. The main focus of this app is to help the user keep track of what is in their fridge and generate recipes based on this information, so that we will minimize waste.

In more detail, we will implement a bar-code scanning mechanism based on pictures provided by the user, and we will also have the option to add them manually. Then, our application will provide helpful instructions on how to cook delicious meals, with no need to buy other products. The user can add their own recipes, right from their grandma’s cookbook, or they can benefit from the wide database that we will connect to: <https://www.barcodelookup.com/4056489662860>.

# General view of actors and use cases

|  |  |  |
| --- | --- | --- |
| ***Actor*** | ***Type*** | ***Description*** |
| *Administrator* | *Human* | *The person administrating the system, the ones that aprove recipes that can be posted, and the ones that choose who is banned from using the app.* |
| *User* | *Human* | *Performs general operations over the system, such that updating the virtual fridge contents, adding recipes, searching for food in the connected database.* |

# Detailed description of actors and use cases

## Actors

### Administrator

**Type**: human

**Detailed description**:

This actor is the person administrating the system. The administrator is the one responsible for the maintenance of the application. They are the ones managing the user’s activity in case of misbehaviour, the ones who choose who get the opportunity to add their recipes to the common database, and enhancing the user experience based on public reviews.

**Where is used:**

The Administrator is involved in the following use cases:

* Administrator Log-In
* Administrator Log-Out
* Validating User Recipes
* Deleting Users
* Enhancing User Experience

### User

Type: human

**Detailed description:**

The User is an individual who interacts with the fridge contents management app to keep track of and manage their household inventory. They can scan barcodes to add items, view product details such as expiration dates, and organize their fridge and pantry items efficiently. Users can also contribute by submitting recipes that use the ingredients they have on hand, share cooking tips, and provide feedback on existing recipes in the database. Additionally, Users can receive notifications for items that are about to expire, set reminders to restock products, and customize their preferences based on dietary needs or personal favorites.

**Where is used:**

* User Registration
* User Log-In
* User Log-Out
* Scanning a Product
* Adding a Product Manually
* Viewing Fridge Contents
* Submitting a Recipe
* Searching for Recipes
* Managing Favorites
* Rating and Reviewing Recipe

## Use cases

### **Use Case Diagram for Administrator**

A diagram of a person's diagram

Description automatically generated

* + - 1. **Administrator Log-In**

**Description:** Allows the Administrator to securely access the system.

**Precondition:** The Administrator has valid credentials.

**Postcondition:** The Administrator is successfully logged into the system and has access to all administrative functions.

**Behaviour:** The Administrator enters their username and password. The system verifies the credentials and, if valid, grants access to the Administrator dashboard. If the credentials are incorrect, an error message is displayed, and access is denied.

#### **Administrator Log-Out**

**Description:** Logs the Administrator out of the system securely.

**Precondition**: The Administrator is logged into the system.

**Postcondition**: The Administrator is securely logged out, and the session is terminated.

**Behaviour:** The Administrator selects the log-out option. The system ends the current session and returns to the log-in screen, ensuring no administrative data remains accessible.

#### **Validating User Recipes**

**Description:** Administrator reviews and approves or rejects user-submitted recipes.

**Precondition:** The Administrator is logged in and there are pending recipe submissions to review.

**Postcondition:** Recipes are either approved and added to the database or rejected and returned to the user with feedback.

**Behaviour:** The Administrator reviews each submitted recipe, checking for completeness, quality, and adherence to guidelines. If a recipe is approved, it is published in the common database. If not, the Administrator provides feedback, and the recipe remains unpublished.

#### **Deleting User**

**Description:** Allows the Administrator to remove user accounts.

**Precondition**: The Administrator is logged in and has identified a user account that requires deletion.

**Postcondition**: The user’s account and associated data are permanently removed from the system.

**Behaviour**: The Administrator selects the user to be deleted and confirms the action. The system deletes the user’s data, including any saved items and preferences, and logs the deletion action.

#### **Validating User Recipes**

**Description:** Uses user feedback to improve the app.

**Precondition**: The Administrator has access to user feedback and system performance data.

**Postcondition**: A plan for enhancements is created or improvements are implemented.

**Behaviour**: The Administrator reviews public feedback and system performance metrics, identifies areas that need improvement, and either schedules updates or applies changes to improve user experience.

### Use Case Diagram for User

A diagram of a user

Description automatically generated

* + - 1. **User Registration**

**Description:** Creates a new user account in the app.

**Precondition:** The User has downloaded the app and wants to create an account.

**Postcondition:** The User’s account is created, and they are logged in.

**Behaviour:** The User fills out a registration form with personal details, such as name, email, and password. The system verifies the information, creates a new user profile, and logs the user into the app.

* + - 1. **User Log-In**

**Description:** Logs the user into their account.

**Precondition:** The User has an existing account.

**Postcondition:** The User is logged into their account and can access their inventory and preferences.

**Behaviour:** The User enters their username and password. The system authenticates the credentials and grants access if they are valid. If not, an error message prompts the user to try again.

* + - 1. **User Log-Out**

**Description**: Logs the User out of their account securely.

**Precondition**: The User is currently logged into the system.

**Postcondition**: The User is logged out, and the session is ended to prevent unauthorized access.

**Behavior**: The User selects the "Log Out" option in the app. The system terminates the active session, clears any cached data, and redirects the User to the login screen. The User must log in again to regain access to their inventory and other features.

* + - 1. **Scanning a Product**

**Description**: Adds a product to the inventory using a barcode scan.

**Precondition**: The User has a product with a barcode and the app is ready to scan.

**Postcondition**: The scanned product is successfully added to the User’s inventory with relevant details, such as name.

**Behaviour**: The User uses the app to scan the barcode of the product. The system retrieves the product details from the barcode database and adds the item to the User’s inventory. If the product is not found in the database, the system may prompt the User to manually enter missing information.

* + - 1. **Adding a Product Manually**

**Description:** Manually adds a product to the inventory.

**Precondition:** The User wants to add a product without a barcode.

**Postcondition:** The product is added to the User’s inventory.

**Behaviour:** The User manually enters details such as the product name, quantity. The system saves this information in the User’s inventory.

* + - 1. **Submitting a Recipe**

**Description:** User submits a recipe for Administrator approval.

**Precondition:** The User is logged in and wants to share a recipe using ingredients from their inventory.

**Postcondition:** The recipe is submitted for Administrator review.

**Behaviour:** The User fills out a recipe submission form, including ingredients, instructions, and a photo. The system saves the submission and sends it to the Administrator for validation.

* + - 1. **Searching for a Recipe**

**Description:** Allows the user to search for recipes using ingredients from their fridge inventory. **Precondition:** User is logged in and has items in their inventory.

**Postcondition:** User receives a list of matching recipes.

**Behaviour:** The User enters keywords or selects ingredients they want to use. The system searches the recipe database and returns a list of recipes that include those ingredients, which the user can view and choose from.

* + - 1. **Managing Favourites**

**Description:** Allows the user to leave feedback on recipes they’ve tried.

**Precondition**: User has viewed or tried a recipe.

**Postcondition**: The rating and review are saved in the system and visible to other users.

**Behaviour**: The User selects a recipe and assigns a rating (e.g., from 1 to 5 stars) and/or writes a review. The system stores the feedback and updates the recipe’s overall rating, displaying the new review for the community to see.

* + - 1. **Rating and Reviewing a Recipe**

**Description:** Allows the user to save and manage their favorite recipes and frequently used products.

**Precondition**: User is logged in.

**Postcondition**: The selected recipes or products are saved to the user's favourites list.

**Behaviour**: The User can mark a recipe or product as a favourite by tapping a save or heart icon. The system adds the item to the favourites list, where the user can easily access it later. Users can also remove items from their favourites list as needed.

# Glossary – used terms

|  |  |  |
| --- | --- | --- |
| **Term** | **Use cases** | **Description** |
| Administrator | |  | | --- | | 2.1.1, 2.2.1 |  |  | | --- | |  | | |  | | --- | | A system actor responsible for managing the application, validating recipes, handling user accounts, and ensuring app quality and user experience. |  |  | | --- | |  | |
| |  | | --- | | User |  |  | | --- | |  | | |  | | --- | | 2.1.2, 2.2.2 |  |  | | --- | |  | | |  | | --- | | A system actor who interacts with the app to manage fridge inventory, search for recipes, contribute feedback, and customize preferences. |  |  | | --- | |  | |
| |  | | --- | | Recipe |  |  | | --- | |  | | |  | | --- | | 2.2.1.3, 2.2.2.6, 2.2.2.7, 2.2.2.9 |  |  | | --- | |  | | |  | | --- | | A set of instructions for preparing food using specific ingredients; users can submit recipes for approval and share them within the app's community. |  |  | | --- | |  | |
| |  | | --- | | Inventory |  |  | | --- | |  | | |  | | --- | | 2.2.2.4, 2.2.2.5 |  |  | | --- | |  | | |  | | --- | | The collection of food items and ingredients managed by the user in the app, which can be added via barcode scanning or manual entry. |  |  | | --- | |  | |
| |  | | --- | | Barcode Scan |  |  | | --- | |  | | |  | | --- | | 2.2.2.4 |  |  | | --- | |  | | |  | | --- | | A feature that allows users to add items to their inventory by scanning product barcodes; uses an external database to retrieve product information. |  |  | | --- | |  | |
| |  | | --- | | Recipe Database |  |  | | --- | |  | | |  | | --- | | 2.2.2.7 |  |  | | --- | |  | | |  | | --- | | A repository within the app containing approved recipes, which users can browse based on available ingredients or other preferences. |  |  | | --- | |  | |
| |  | | --- | | Favorites |  |  | | --- | |  | | |  | | --- | | 2.2.2.8 |  |  | | --- | |  | | |  | | --- | | A feature that lets users save preferred recipes and frequently used products for easy access. |  |  | | --- | |  | |
| |  | | --- | | Rating & Review |  |  | | --- | |  | | 2.2.2.9 | |  | | --- | | A feedback system enabling users to rate and review recipes they’ve tried, contributing to community insights and recommendations. |  |  | | --- | |  | |
| |  | | --- | | Feedback |  |  | | --- | |  | | |  | | --- | | 2.2.1.5, 2.2.2.9 |  |  | | --- | |  | | |  | | --- | | User-provided input on recipes or the app itself, which the administrator uses to improve functionality and user satisfaction. |  |  | | --- | |  | |
| |  | | --- | | Fridge Contents Management |  |  | | --- | |  | | |  | | --- | | 1, 2 |  |  | | --- | |  | | |  | | --- | | The central purpose of the app, focused on helping users optimize and track their stored food items to reduce waste and improve meal planning. |  |  | | --- | |  | |
| |  | | --- | | Dietary Preferences |  |  | | --- | |  | | |  | | --- | | 2.2.2.9 |  |  | | --- | |  | | User-specific settings that personalize recipe suggestions and notifications based on dietary restrictions, allergies, or nutritional goals. |

|  |  |  |
| --- | --- | --- |
| **Term Groups** | **Included terms and term groups** | **Description** |
| |  | | --- | | Actors |  |  | | --- | |  | | |  | | --- | | Administrator, User |  |  | | --- | |  | | |  | | --- | | Individuals or entities that interact with the app, each with specific roles and permissions. |  |  | | --- | |  | |
| |  | | --- | | Inventory Management |  |  | | --- | |  | | |  | | --- | | Inventory, Barcode Scan |  |  | | --- | |  | | |  | | --- | | Features related to managing, organizing, and tracking food items within the app. |  |  | | --- | |  | |
| |  | | --- | | User Engagement |  |  | | --- | |  | | Recipe, Rating & Review, Favorites Feedback | Features designed to enhance the user experience through interaction, customization, and community contributions. |