

My Chief Cook

Informatique et Gestion

Polytech Montpellier



USE CASES ANALYSIS

MyChieCook

Présenté par

[Corentin Clément]

[Alexis Fondard Martin]

[Anais Velcker]

[Richard Martin]

[Lucas Nouguier]

[Lead, Analytics]

[Development, Tests]

[Documentation, Use cases]

[Graphical User Interface]

[Database, Reports]



POLYTECH
MONTPELLIER



UNIVERSITÉ
DE MONTPELLIER

Contents

| | | |
|----------|---------------------------------------|----------|
| 1 | Problem Statement | 3 |
| 1.1 | Context | 3 |
| 1.2 | Problem | 3 |
| 1.3 | Constaints | 3 |
| 1.4 | Core Features | 4 |
| 2 | Use Cases Model | 4 |
| 3 | Use Cases Estimated Complexity | 5 |
| 4 | Use Cases Description | 5 |
| 4.1 | Ingredient management | 5 |
| 4.2 | Recipe management | 6 |
| 5 | Annexes | 7 |

1 Problem Statement

1.1 Context

You have been tasked by a start-up with building a proof of concept of their new cooking application. They need a prototype to show to investors and potential customers. The application will be used by chefs to manage their recipes (and their inventory). The application will also be used by customers to search for recipes and eventually wines to come with their meal.

1.2 Problem

The final system will be an interactive web app, but for the proof of concept, you will be building a desktop application capable of running on any devices. It will allow chefs to expose their recipes and partners while users will be able to order meals and wines. The application will also be able to suggest recipes based on the ingredients available in the user's fridge. Once they've made their choices, they must be able to pass command to either the restaurants, either to partners if they want to buy ingredients. They should as well receive wine recommendations if they chose to. Users will launch this prototype locally on their computer. For security and consistency reasons, some features will require some permissions (such as moderation).

For a more pleasant user experience, we'll need users to be able to create "favorite lists" which must be name and where they can add recipes to easily find later. They must have full ownership of these lists. There will also be a feature to allow users to manage their meal plan up until 3 weeks ahead. They will be able to choose recipes and place them as a midday or evening meal. From the calendar users must be able to see the ingredient shopping list and the calendar should be persistent and auto-update according to the current date (for instance by strike-through past meals).

Users can send feedbacks too, by rating and commenting recipes. To promote helpful comments, there will be a voting system for comments (upvote/downvote) and comments with a high score (lots of upvotes) will be among the first comment in the list, whereas comments with low score (lots of downvotes) will be among the last. In addition, They should be able to suggest new recipes to chefs by giving a description of what they think about (and eventually a name). Chefs will then be able to accept or reject the suggestion.

Chefs should be able to add to announcements to their page and manage them (update, hide, delete...). An announcement may have some labels to facilitate the search at a latter point

1.3 Constraints

Users should not be forced to expose their private life, so the system will retain as few information as possible :

- Mandatory fields : their username, email address
- Optional fields : name, address, social network accounts, gender, birthday

Yet, chefs must provide each fields for transparency reasons. Also, some features will required optional fields to be fulfilled, so user must be warned when some fields are missing.

1.4 Core Features

- Chose a recipe from ingredients and eventually auto-ajust proportions to match
 - the number of people
 - the number of meals
 - the current quantity of ingredients in the fridge
- Optionally add wine to the meal
- Make a meal plan for at most the next 3 weeks
- Redirect to partners' website to buy ingredients and/or wines

2 Use Cases Model

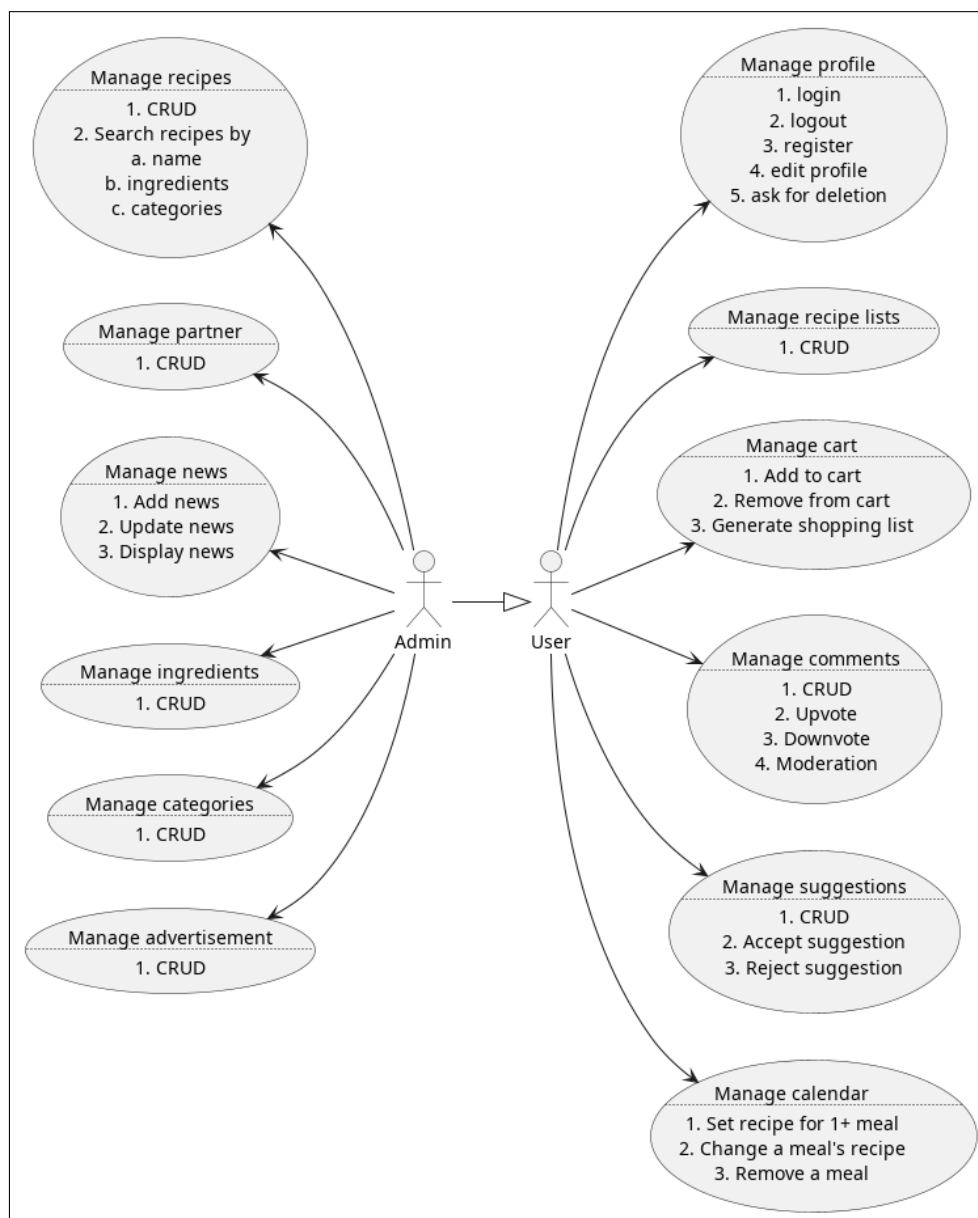


Figure 1: Use Cases Model

3 Use Cases Estimated Complexity

| | | |
|--|---|---|
| Manage profile <i>Assignee : Anaïs V.</i> <i>Complexity : 5/5</i> | Manage recipe lists <i>Assignee : Richard M.</i> <i>Complexity : 2/5</i> | Manage cart <i>Assignee : Richard M.</i> <i>Complexity : 3/5</i> |
| Manage recipes <i>Assignee : Lucas N.</i> <i>Complexity : 4/5</i> | Manage ingredients <i>Assignee : Lucas N.</i> <i>Complexity : 2/5</i> | Manage categories <i>Assignee : Corentin C.</i> <i>Complexity : 1/5</i> |
| Manage comments <i>Assignee : Alexis F.</i> <i>Complexity : 3/5</i> | Manage suggestions <i>Assignee : Corentin C.</i> <i>Complexity : 3/5</i> | Manage calendar <i>Assignee : Alexis F.</i> <i>Complexity : 3/5</i> |
| Manage news <i>Assignee : Anaïs V.</i> <i>Complexity : 2/5</i> | Manage partners <i>Assignee : Richard M.</i> <i>Complexity : 1/5</i> | Manage advertisement <i>Assignee : Corentin C.</i> <i>Complexity : 2/5</i> |

4 Use Cases Description

4.1 Ingredient management

4.1.1 Brief description

Ingredients are the base of recipes. As for recipes, they can be modified (created, updated, deleted) by chefs only. They will be able to specify a name, a list of tags (categories), a description and whether it's allergen or not. If it's allergenic, it must be emphasized.

Chefs should be able to search ingredients by name and the result should be sorted by name.

Still, even though it is possible, changing an ingredient it is not recommended as it may break some recipes. Deleting an ingredient is also possible, but it may break some recipes, so it should be possible only if no recipe depends upon this ingredient. Otherwise, the chef will receive a warning telling him that some recipes will be broken so he will have to update/delete the recipes first.

4.1.2 Flow of events

See annex [Figure 2](#)

4.1.3 Special requirements

Must be thread-safe in case multiple chefs are modifying the same ingredient.

4.1.4 Preconditions

The chef must be logged onto the system before this use case begins.

4.1.5 Postconditions

If it's successful, the ingredient will be added/updated/deleted from the database. Otherwise, the state remains unchanged.

4.1.6 Extension points

None

4.2 Recipe management

4.2.1 Brief description

Recipes can be modified (created, updated, deleted) by chefs only. They will be able to specify a name, a list of ingredients (and the quantities), some tags (categories), a description, and maybe a picture of it.

Recipes can be search by name, tags, ingredients, whether it contains allergens or not, and may be sorted by name or popularity.

If a recipe contains allergens, it must be specified and allergenic ingredients must appear in emphasized text.

A recipe can receive many comments and ratings from the users, which can also suggest recipes. Chefs will then accept or reject suggestions.

4.2.2 Flow of events

See annex [Figure 3](#)

4.2.3 Special requirements

Must be thread-safe in case multiple chefs are modifying the same recipe.

4.2.4 Preconditions

The chef must be logged onto the system before this use case begins.

4.2.5 Postconditions

If it's successful, the recipe will be added/updated/deleted from the database. Otherwise, the state remains unchanged.

4.2.6 Extension points

None

5 Annexes

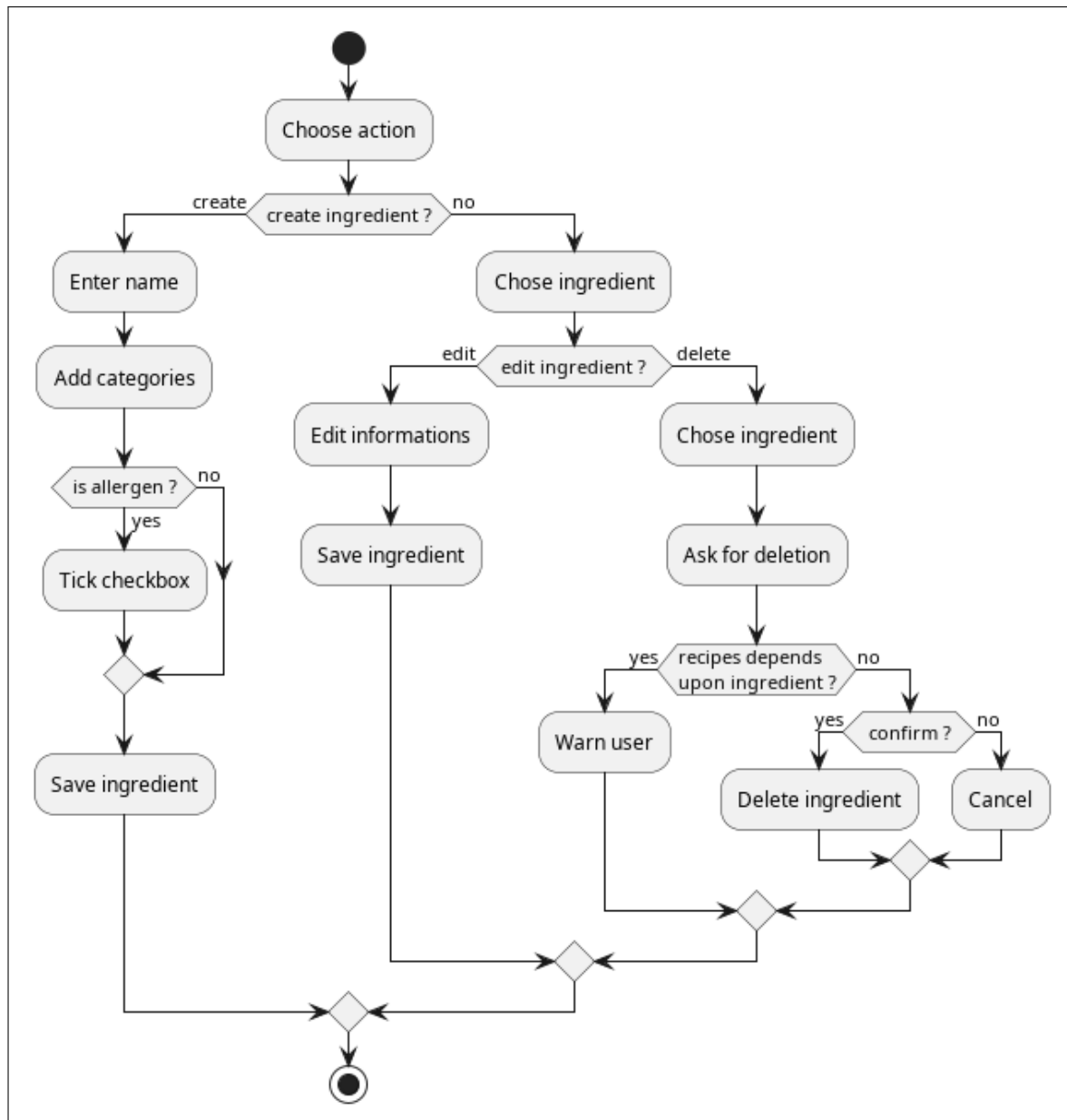


Figure 2: Ingredient management flow diagram (return to [subsection 4.1](#))

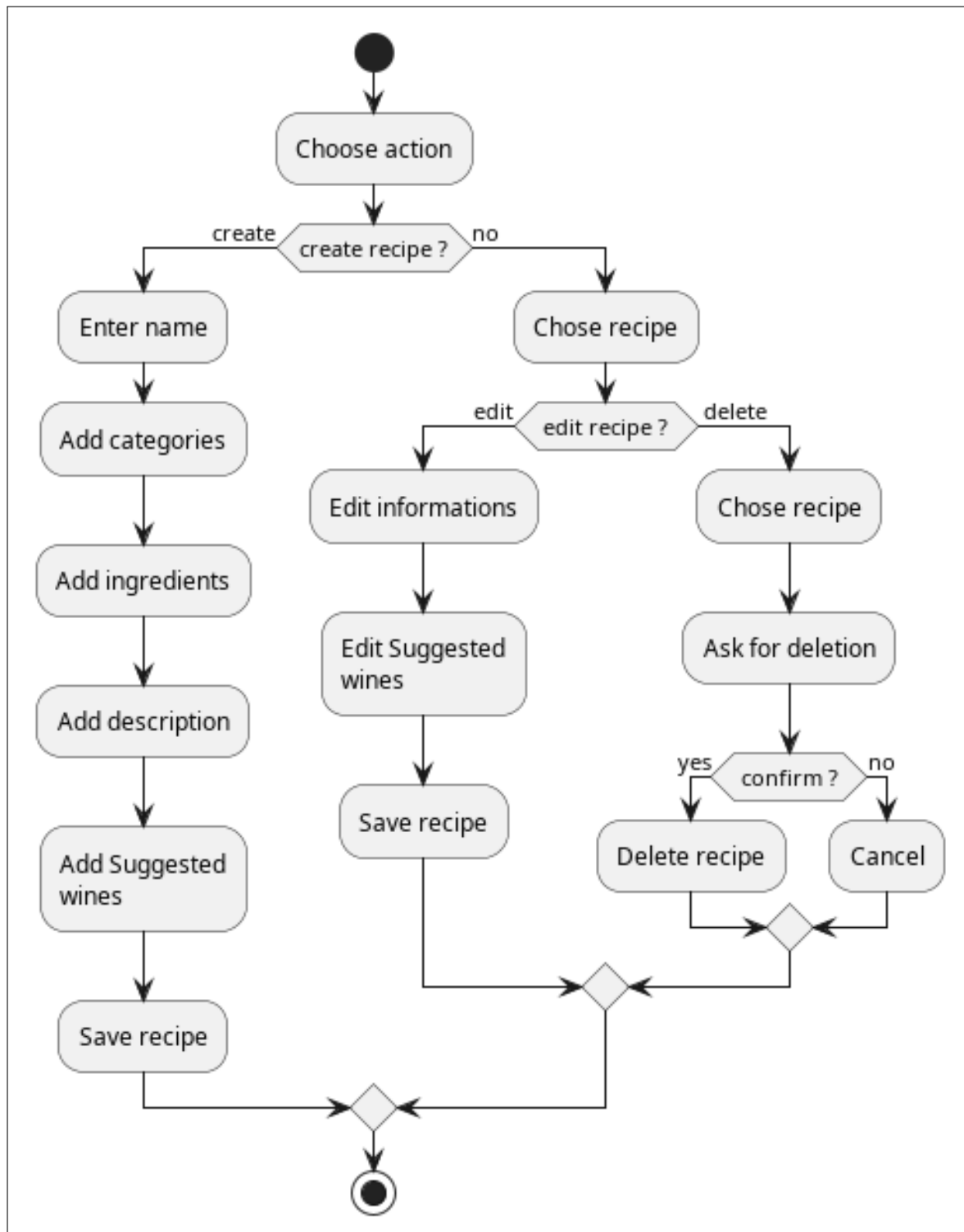


Figure 3: Recipe management flow diagram (return to [subsection 4.2](#))