Ingegneria del software e progettazione web
Progetto A.A. 2022/2023

FOODO

0292831

Flavio Campobasso

0293712

Carlo Maria Fioramanti

Index of contents:

1 Introduction

1.1. Aim of the documentation

The aim of the documentation is to provide a full description of the software system developed following a well-defined approach, based on practices of software engineering, toward the satisfaction of the project goals.

1.2. Overview of the defined system

The "Foodo" project is a system that allows people to manage their pantry in the kitchen, including all the products they own, and allows chefs to publish their recipes.
The beating heart of "Foodo" is the possibility provided to users to be able to generate the recipes they can do with the ingredients they possess in the kitchen, as well as the ability to search for a recipe through the name.
The chef can publish a recipe specifying the name, ingredients, description, and photo.
The basic user can see the recipes, follow the chef and evaluate the recipe with a rating ranging from 1 to 5.
In addition, the basic user can see the recipes published only by the chefs below.
Both the user base and chef can manage their pantry (by entering products specifying name, quantity, expiration date, and type of product), generate/ search recipes and see their profile with their photo and their information.
In the system, the user can interact with the software through two interfaces: one graphic and the other by command line.

1.3. HW and SW requirements

The software and hardware requirements:

- RAM: 2GB of free RAM
- CPU: any modern CPU
- Disk Space: 4GB
- Monitor resolution: 1024x768
- Operating System: Microsoft 8 or later, macOS 10.14 or later, any Linux distributions that supports Gnome, KDE or Unity DE

1.4. Related System, Pros and Cons

Other apps/websites like Giallo Zafferano, whisk and My Recipes Box allow you to see recipes but not to generate them based on the ingredients you enter.
"Foodo" instead gives the possibility to generate recipes based on the ingredients that the user inserts.
A con is that our system is single-user and does not allow simultaneous use by different users.
You also have to add the ability to create an account.
A pro is that not all apps/ sites allow you to have a personalized profile and generate recipes based on the ingredients.

2 User Stories

1) As a user, I want to add products to the pantry*, so that I can see the products I own.
2) As a user, I want to enter a product’s expiration date, so that I can see when it's expired.
3) As a user, I want to follow my favorite chefs, so that I can see their recipes.
4) As a chef, I want to add a recipe, so that the users can see it.
5) As a chef, I want to see the reviews of my recipe, so that I can understand if the recipe is liked.
6) As a user, I want to do a review, so that the other user can read it.

*Pantry = list of products that the user owns.

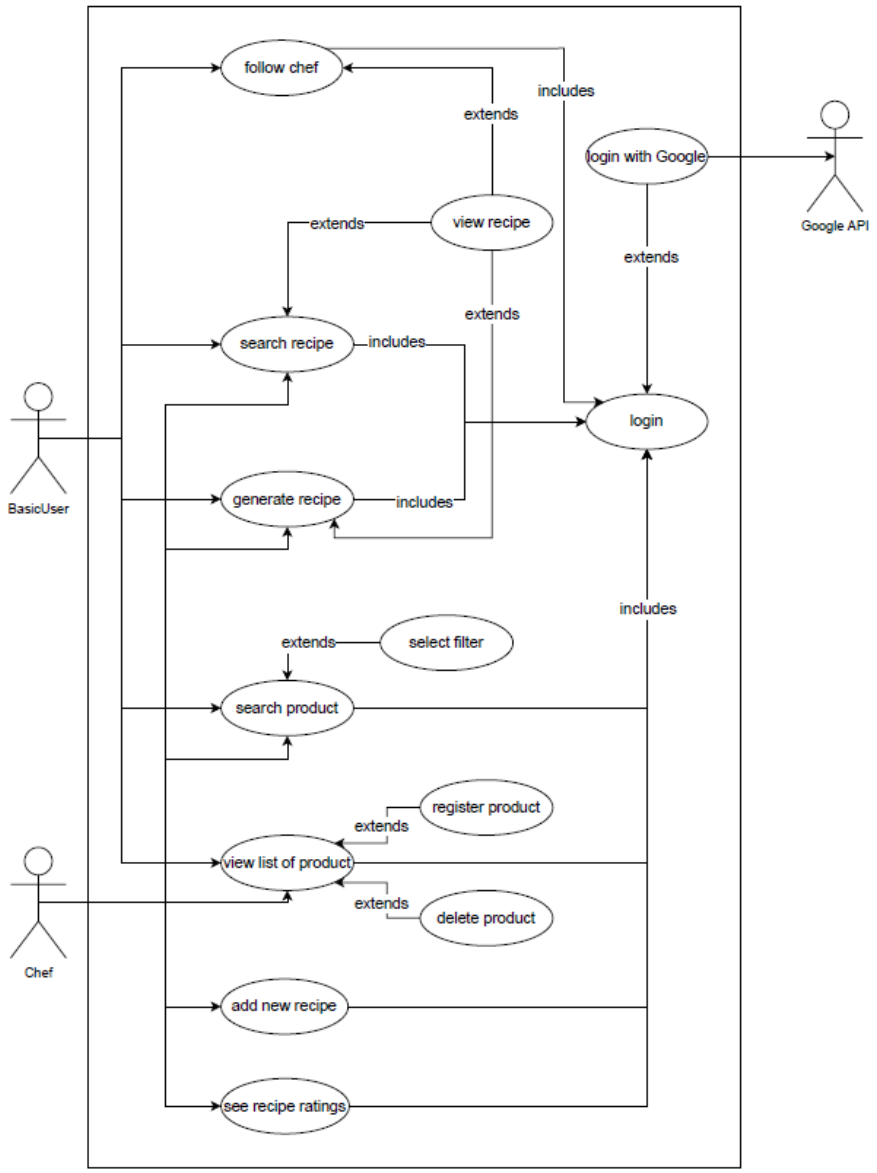
3 Functional Requirements

1. The system shall search for recipes that the user can make with the products he has entered.
2. The system shall give the possibility to the users to insert products in the pantry specifying name, quantity, expiration, and type of product.
3. The system shall give users the possibility to search for a product both by name and by type.
4. The system shall display the review to the chef.
5. The system shall display the chef's recipe to the basic user.
6. The system shall publish the user's review.

4 Use Cases

4.1 Diagram

Link: https://github.com/carlo-fioramanti/Deliverables_Foodo/blob/57b93020003763c6df59be3bbdbdb28c9261031f/Use%20Case%20Diagram/UC%20diagram.drawio.pdf



Login with Google are not implemented

4.2 Internal Steps

Use case: Use the pantry.

1. The user requests to use the pantry.
2. The system displays the pantry.
3. The user inserts the product info.
4. The user adds the product.
5. The system gets available the product in the pantry.
6. The user deletes a product.
7. The system updates the pantry without the product.

4a. *The name field is empty:* the system shows an error message.

4b. *The expiration field is empty*: the system shows an error message.

4c. *The quantity field is empty*: the system shows an error message.

4d. *The product type field is empty:* the system shows an error message.

9a. *The user has not selected any product:* the system shows an error message.

Use case: Insert a recipe.

1. The chef requests to insert a new recipe.
2. **The chef logs in**.
3. The system prepares a blank schedule form.
4. The chef inserts the recipe info.
5. The chef confirms the recipe.
6. The system memorizes the recipe.

5a. *The name is invalid*:* the system prompts the schedule form again.
5b. *The chef doesn't insert the ingredients:* the system shows an error message.
5c. *The chef doesn't insert the recipe image:* the system shows an error message.
5d. *The chef doesn't insert the recipe description:* the system shows an error message.

*
Invalid: the name is already inserted by the chef, or the field is empty.

5 Storyboards

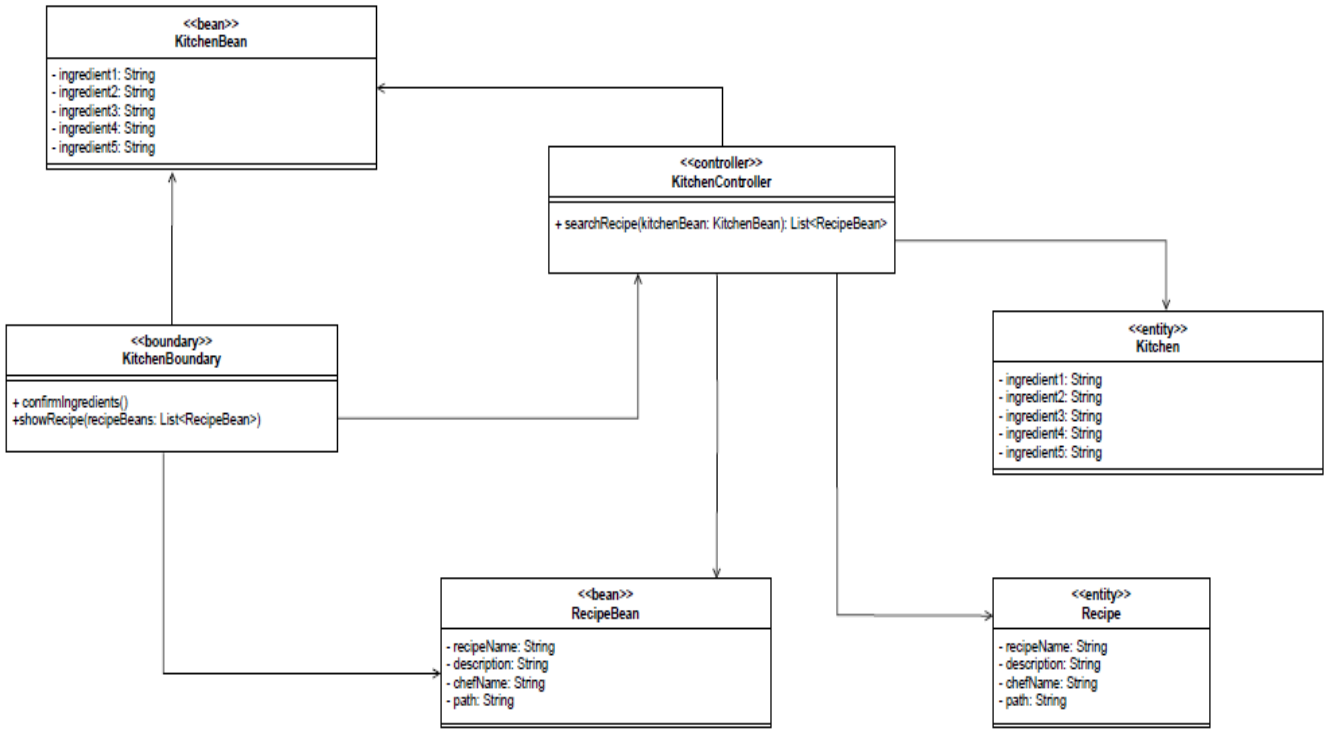
Link: https://github.com/carlo-fioramanti/Deliverables_Foodo/tree/35229800fe66ba93ec53213ba1b512f415359f69/UI%20Prototypes

6 Class Diagram

6.1 BCE

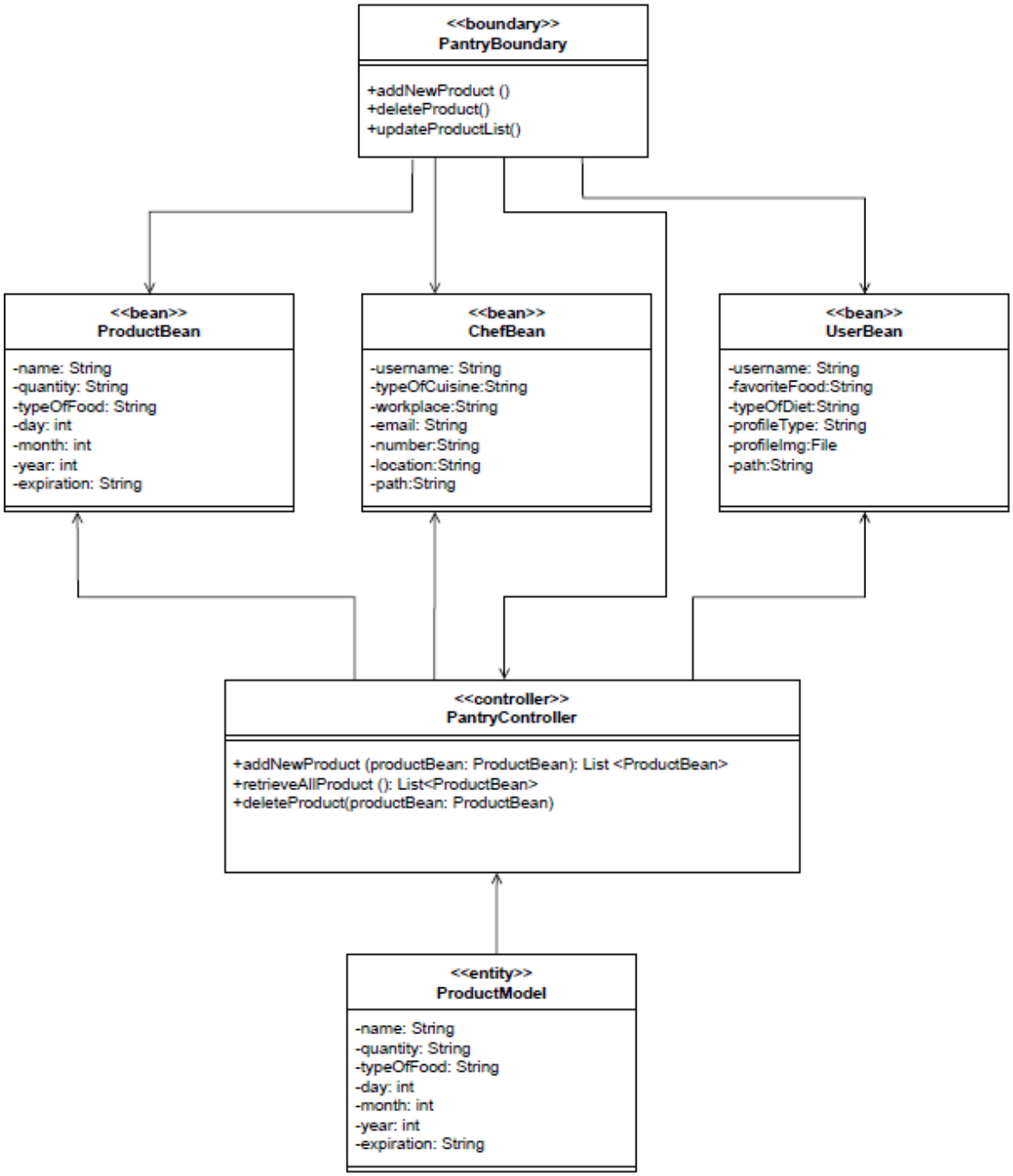
Link: https://github.com/carlo-fioramanti/Deliverables_Foodo/blob/c64849ac013a7428396e3b037f70f7162fd1fc51/BCE/BCEGenerateRecipe.pdf

Generate recipe:



Link: https://github.com/carlo-fioramanti/Deliverables_Foodo/blob/b37987587c712dc53f451468a9260dce3888f7e1/BCE/BCEAddNewProduct.pdf

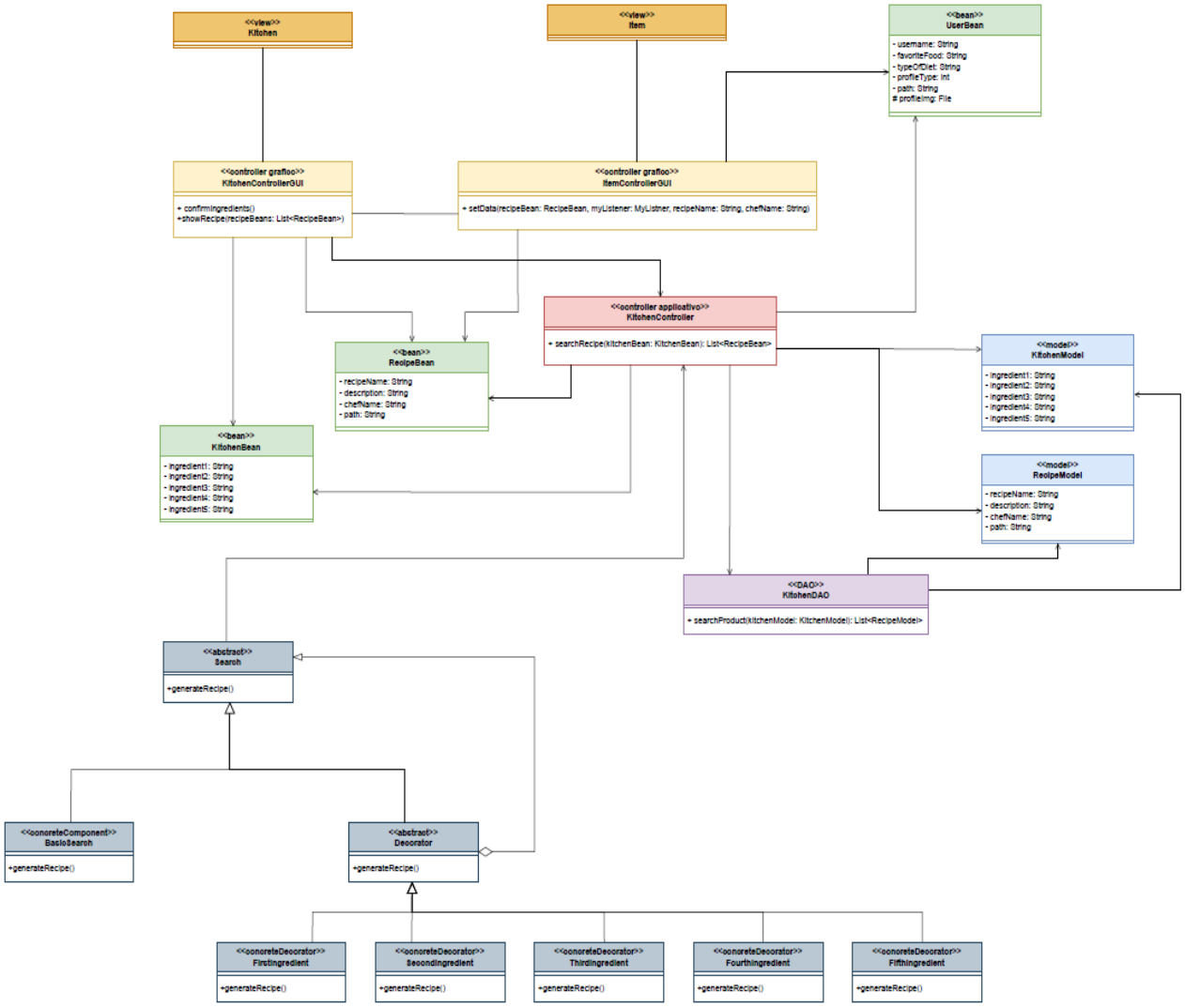
Add new product:



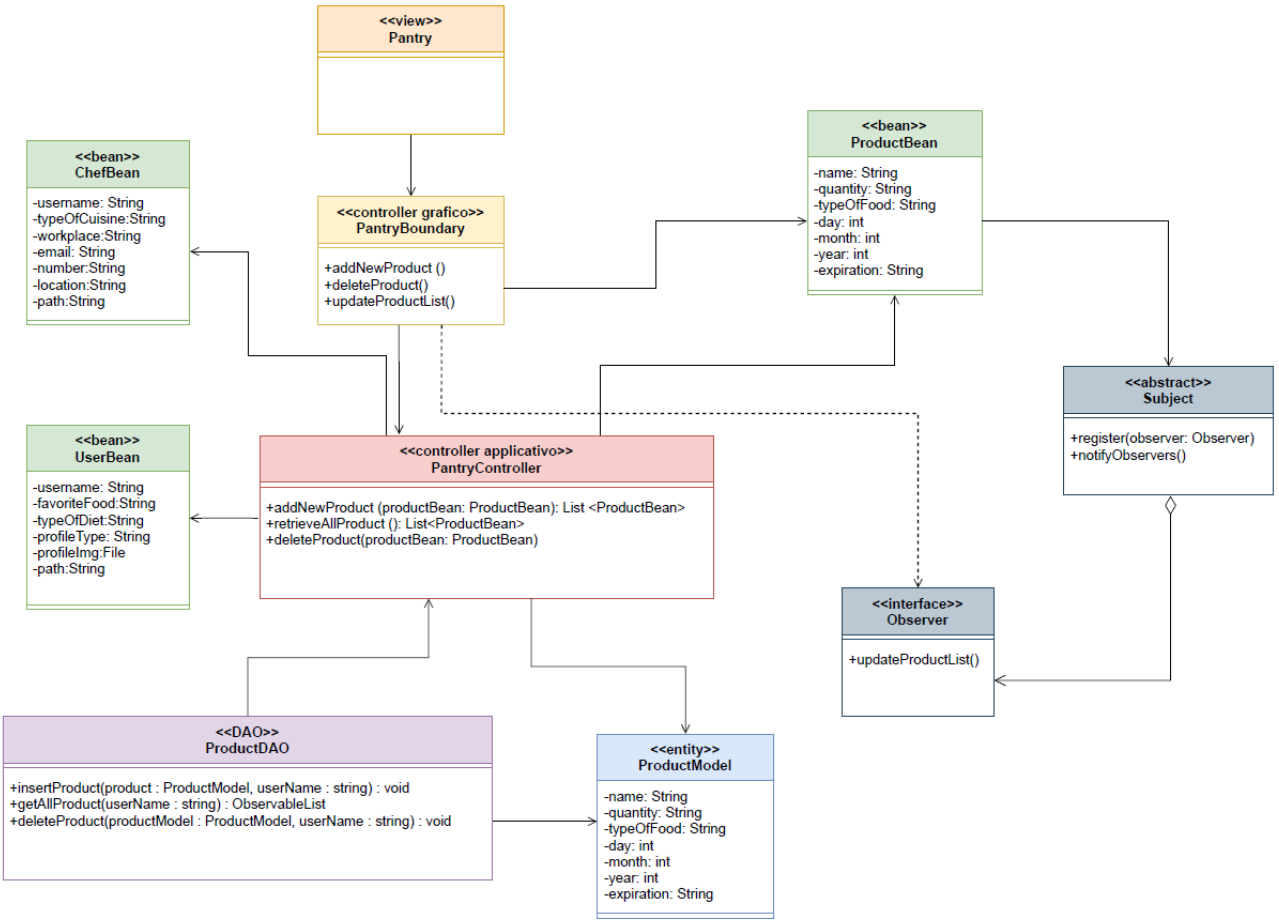
6.2 MVC

Link: https://github.com/carlo-fioramanti/Deliverables_Foodo/tree/b37987587c712dc53f451468a9260dce3888f7e1/MVC

Generate recipe:



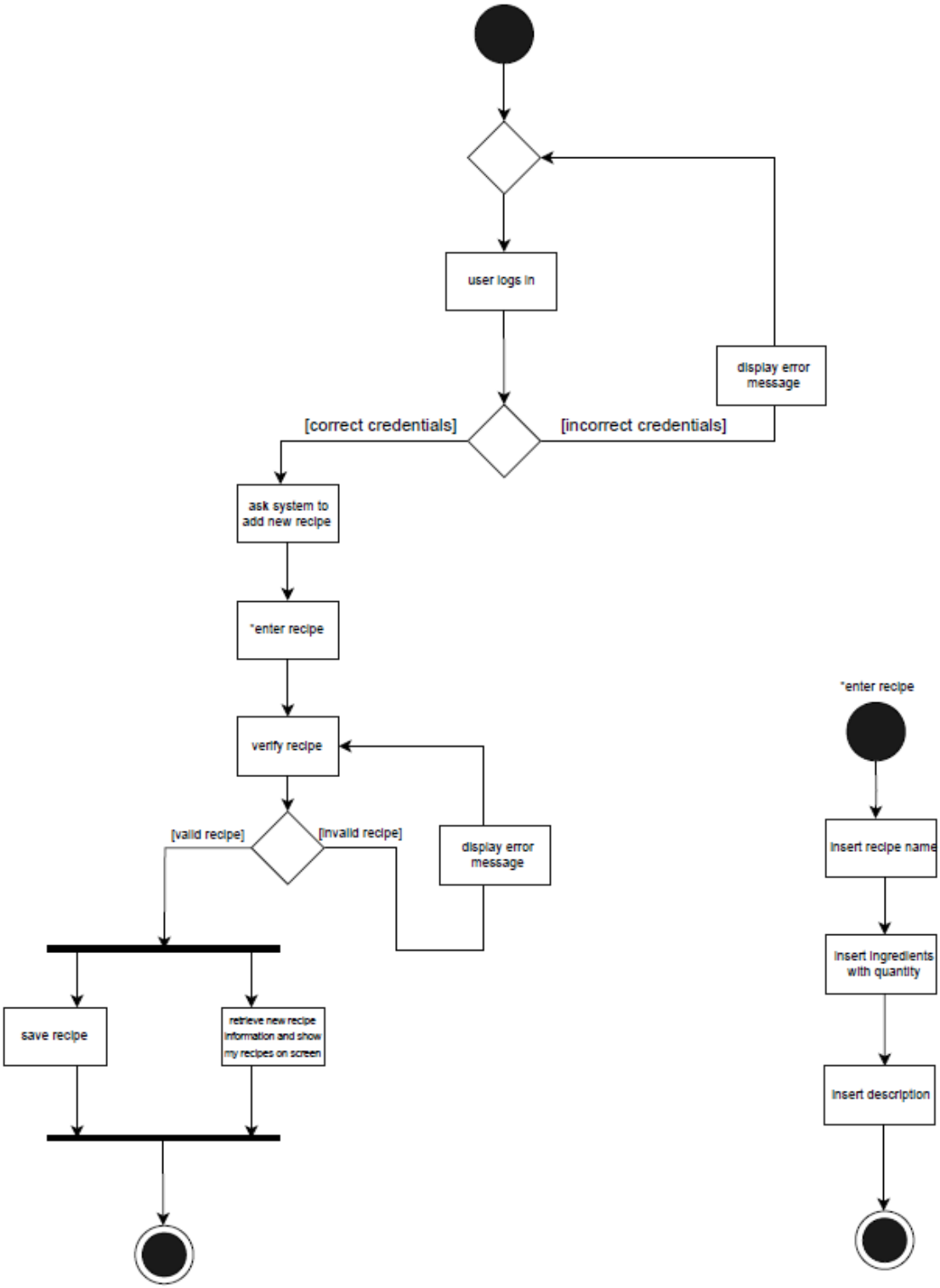
Add new product:



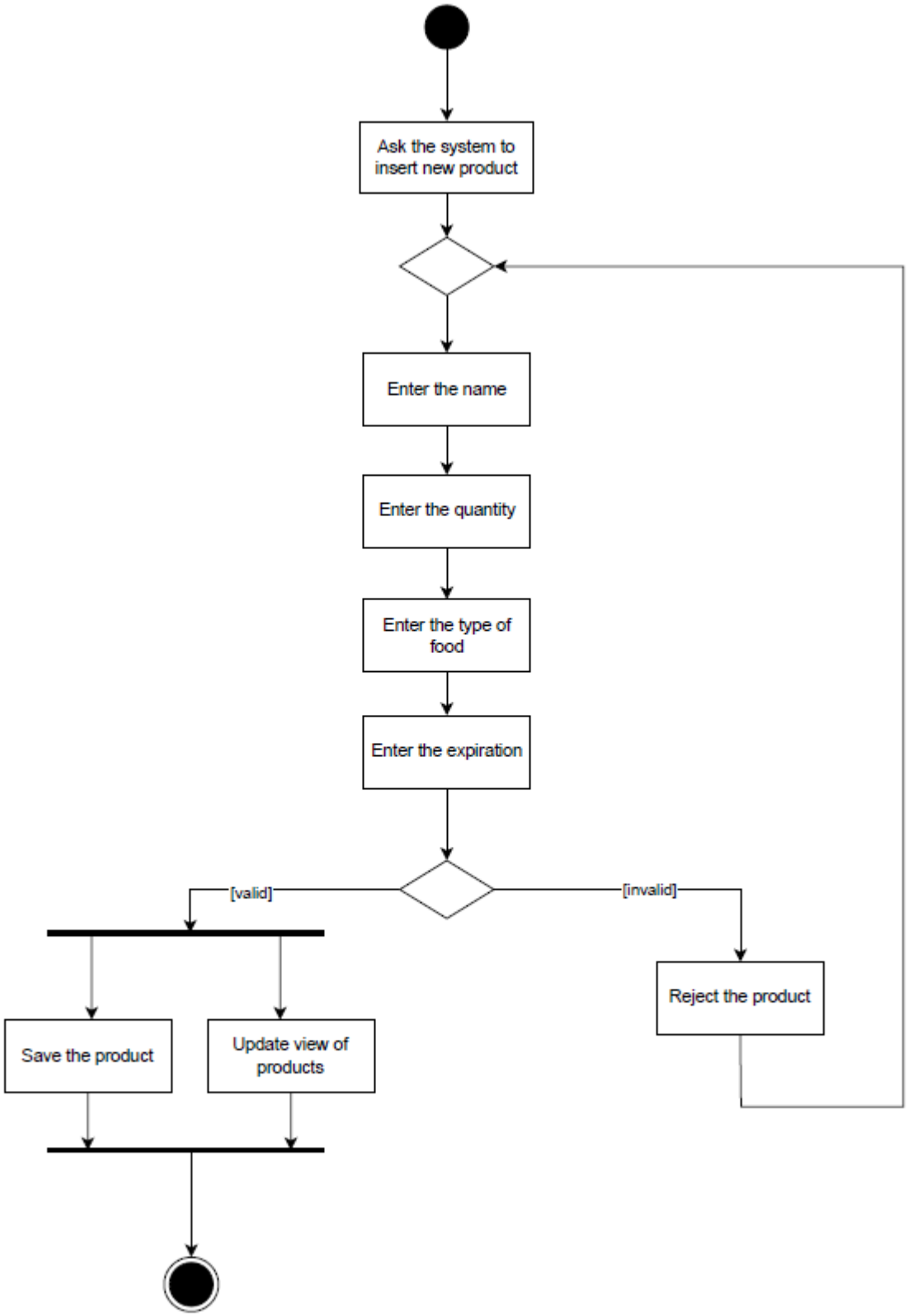
7 Activity Diagram

Link: https://github.com/carlo-fioramanti/Deliverables_Foodo/tree/0cf68ea45bfc2b5809eff20d24dfa518fe2b6f0b/Activity%20Diagram

Add new recipes:



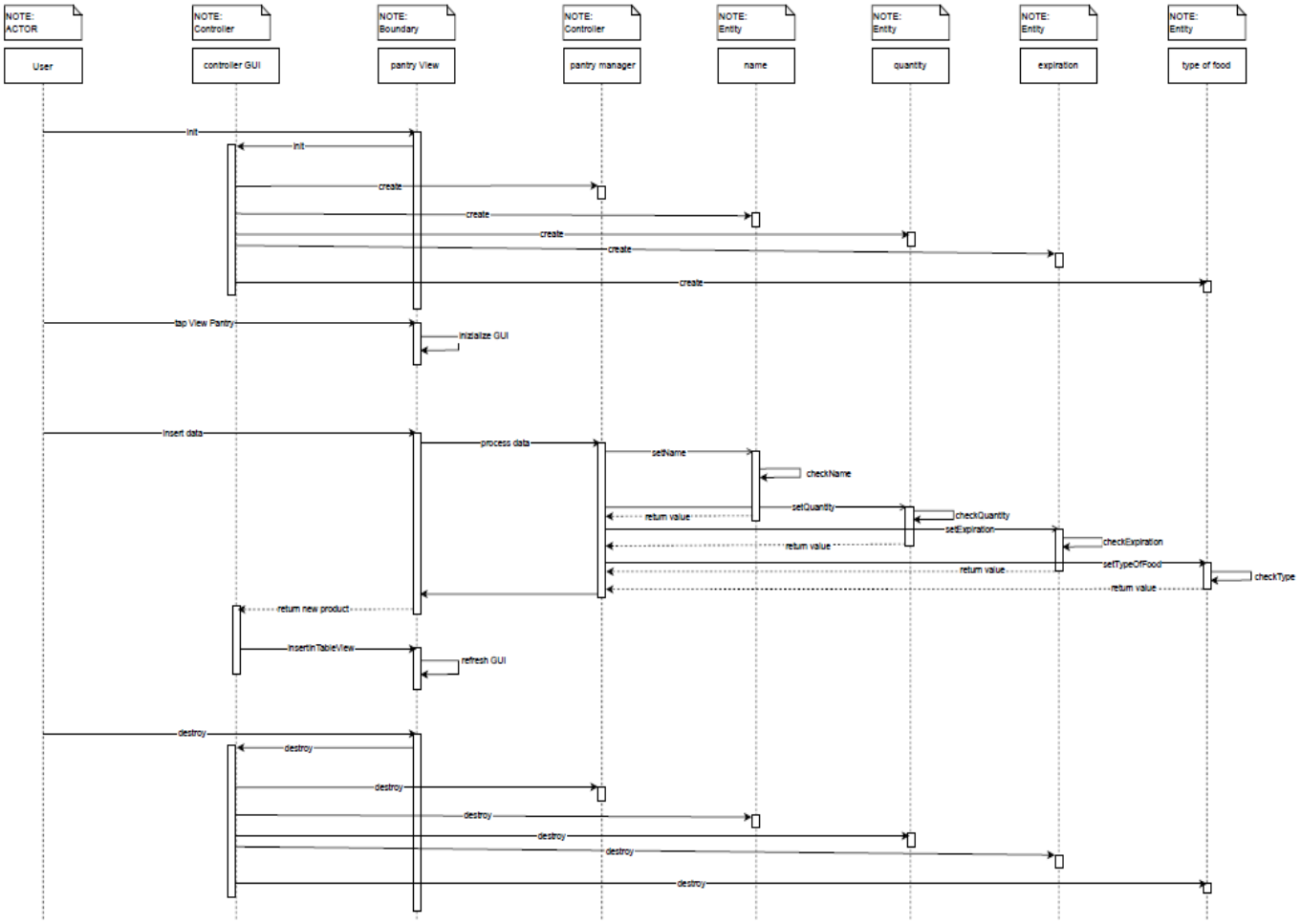
Add new product:



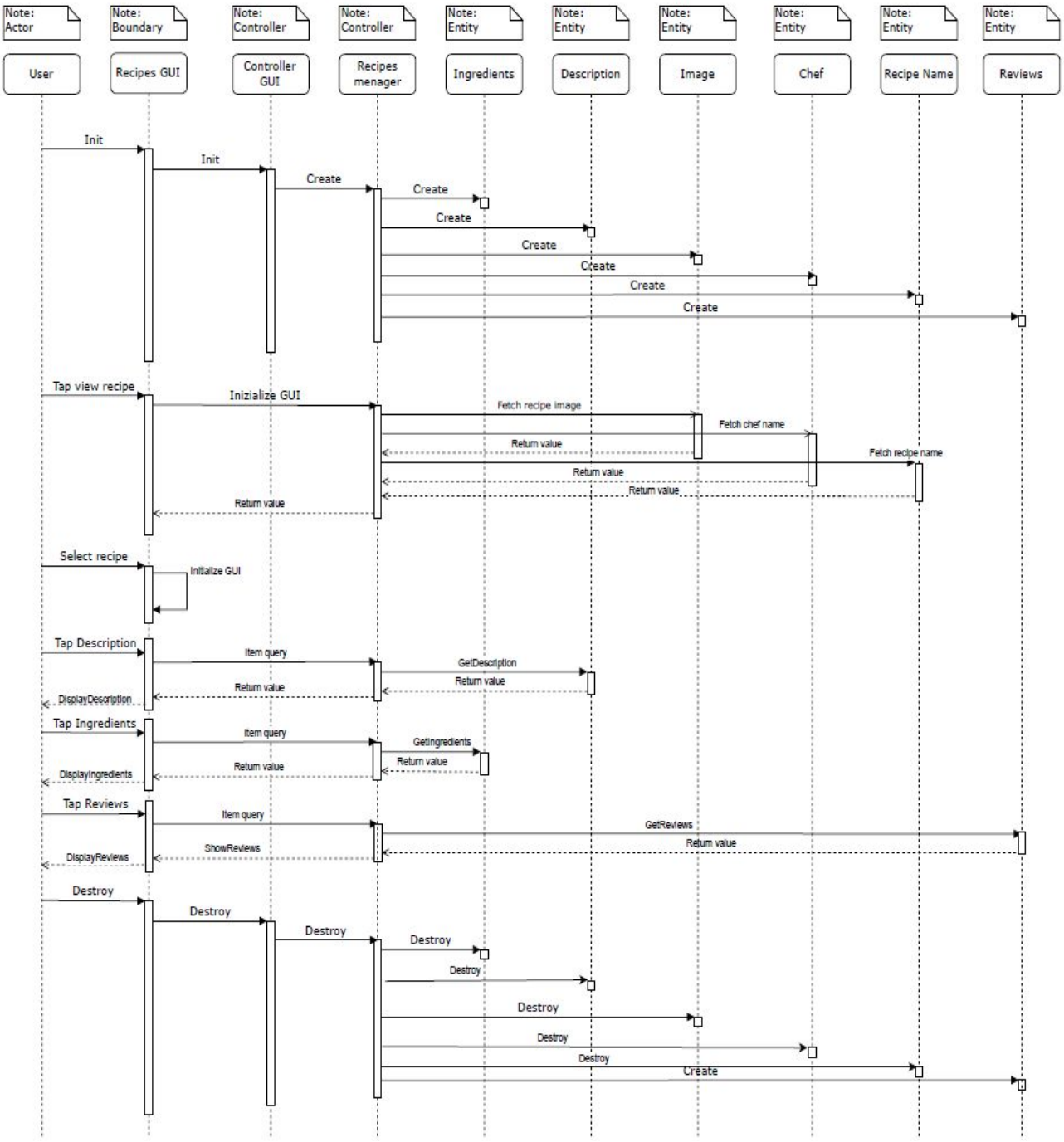
8 Sequence Diagram

Link: https://github.com/carlo-fioramanti/Deliverables_Foodo/tree/0cf68ea45bfc2b5809eff20d24dfa518fe2b6f0b/Sequence%20Diagram

Add new product:



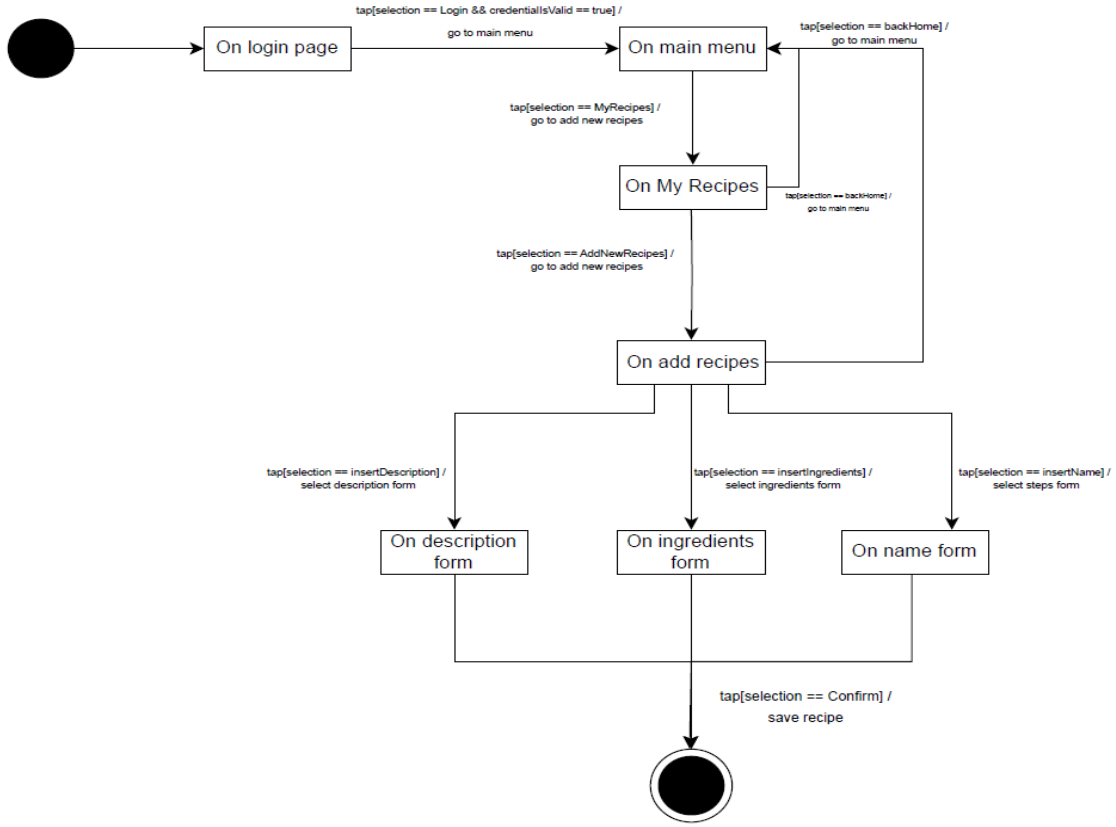
View recipe:


Note: Actor
Note: Boundary
Note: Controller
Note: Controller
Note: Entity
Note: Entity
Note: Entity
Note: Entity
Note: Entity
Note: Entity
User
Recipes GUI
Controller GUI
Recipes menager
Ingredients
Description
Image
Chef
Recipe Name
Reviews
Init
Init
Create
Create
Create
Create
Create
Create
Create
Tap view recipe
Inizialize GUI
Fetch recipe image
Fetch chef name
Return value
Fetch recipe name
Return value
Return value
Return value
Select recipe
Initialize GUI
Tap Description
Item query
GetDescription
Return value
Return value
DisplayDescription
Tap Ingredients
Item query
GetIngredients
Return value
Return value
DisplayIngredients
Tap Reviews
Item query
GetReviews
Return value
ShowReviews
DisplayReviews
Destroy
Destroy
Destroy
Destroy
Destroy
Destroy
Destroy
Destroy
Create

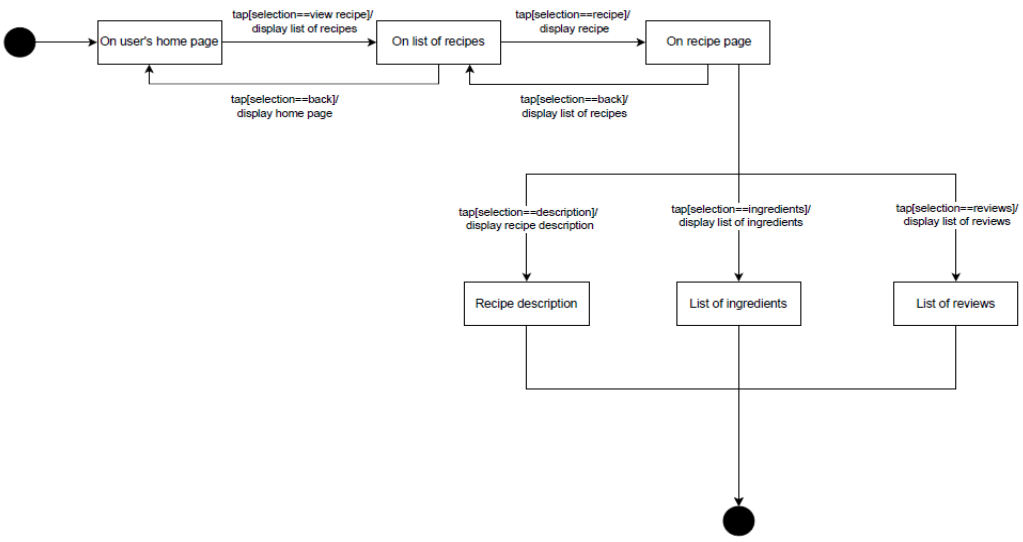
9 State Diagram

Link: https://github.com/carlo-fioramanti/Deliverables_Foodo/tree/0cf68ea45bfc2b5809eff20d24dfa518fe2b6f0b/State%20Diagram

Add new recipes:



View recipes:



10 Testing

10.1 Selenium Test Via GUI

Link: https://github.com/carlo-fioramanti/Deliverables_Foodo/tree/0cf68ea45bfc2b5809eff20d24dfa518fe2b6f0b/Test%20Selenium%20GUI

10.2 Selenium Test Via API

Link: https://github.com/carlo-fioramanti/Deliverables_Foodo/tree/0cf68ea45bfc2b5809eff20d24dfa518fe2b6f0b/Test%20Selenium%20API

11 Code

Link: https://github.com/carlo-fioramanti/foodo.git

12 Video

Link: https://github.com/carlo-fioramanti/Deliverables_Foodo/tree/0cf68ea45bfc2b5809eff20d24dfa518fe2b6f0b/Video

13 Sonar Cloud

Link: https://sonarcloud.io/project/overview?id=carlo-fioramanti_foodo