



# Hungry

Making cooking simple

## Απαιτήσεις Χρηστών και Προδιαγραφές Συστήματος

**Del.1.1**

Version 0.1

12/04/2021



# Hungry

Making cooking simple

Έγγραφο Απαιτήσεων Χρηστών

εκδ. 0.1

12/04/2021

## Μέλη της Ομάδας Ανάπτυξης

| Όνομα             | ΟΑ     | Email  |
|-------------------|--------|--|
| Α. Συμεωνίδης     | *      | <a href="mailto:asymeon@issel.ee.auth.gr">asymeon@issel.ee.auth.gr</a> |
| Ε. Κάλλα          | Hungry | <a href="mailto:elenikalla@ece.auth.gr">elenikalla@ece.auth.gr</a>     |
| Ε. Κελέση         | Hungry | <a href="mailto:elpidakelesi@ece.auth.gr">elpidakelesi@ece.auth.gr</a> |
| Ι. Κωνσταντινίδης | Hungry | <a href="mailto:iordaniak@ece.auth.gr">iordaniak@ece.auth.gr</a>       |
| Α. Παναγιώτου     | Hungry | <a href="mailto:alexandp@ece.auth.gr">alexandp@ece.auth.gr</a>         |



## Πίνακας Περιεχομένων

|   |    |
|---|----|
| Πίνακας Περιεχομένων                              | 3  |
| Λίστα Σχημάτων                                    | 4  |
| 1 Απαιτήσεις Συστήματος                           | 6  |
| 1.1 Λειτουργικές απαιτήσεις (Σενάρια χρήσης)      | 6  |
| 1.2 Χρήστες και εξωτερικά συστήματα               | 8  |
| 1.2.1 Χρήστες                                     | 8  |
| 1.2.2 Εξωτερικό Σύστημα                           | 8  |
| 1.3 Significant Non-Functional Requirements       | 9  |
| 1.3.1 Performance Requirements                    | 9  |
| 1.3.2 Security Requirements                       | 9  |
| 1.3.3 Portability Requirements                    | 9  |
| 1.3.4 Interoperability Requirements               | 9  |
| 1.3.5 Scalability Requirements                    | 10 |
| 1.4 Lexicographical Definitions                   | 10 |
| 2 Σενάρια Χρήσης                                  | 11 |
| 2.1 Διάγραμμα σεναρίων χρήσης                     | 11 |
| 2.2 Feature 1: Create Account                     | 11 |
| 2.3 Feature 2: Edit Personal Information          | 13 |
| 2.4 Feature 3: Edit MyIngredients                 | 16 |
| 2.5 Feature 4: Search Recipes                     | 18 |
| 2.6 Feature 5: Edit Favorites from Recipe Details | 20 |
| 2.7 Feature 6: Comment Recipe                     | 22 |
| 2.8 Feature 7: Rate Recipe                        | 23 |
| 2.9 Feature 8: CookedIt Button                    | 24 |
| 2.10: Feature 9: Select from Favorites            | 26 |
| 2.11: Feature 10: Remove from Favorites           | 27 |
| 2.12: Feature 11: Add New Recipe                  | 29 |
| 3 Επιδεικτικά γραφικά παράθυρα διεπαφής           | 31 |
| 3.1 Home Page                                     | 31 |
| 3.2 Login   | 31 |
| 3.3 Register                                      | 32 |
| 3.4 Recipes                                       | 32 |
| 3.5 Filters                                       | 33 |
| 3.6 Recipe  | 33 |
| 3.7 My Ingredients                                | 34 |
| 3.8 Search Ingredients                            | 34 |
| 3.9 Add Recipe                                    | 35 |



# Hungry

Making cooking simple

Έγγραφο Απαιτήσεων Χρηστών

εκδ. 0.1

12/04/2021

|                                  |    |
|----------------------------------|----|
| 3.10: Account                    | 35 |
| 3.11 : Edit Account Details      | 36 |
| 3.12 Favorites                   | 36 |
| 3.13 Remove Favorite             | 37 |
| 4 Στατική Μοντελοποίηση          | 38 |
| 4.1 High Priority                | 38 |
| 4.1.1 Home Page Package          | 38 |
| 4.1.2 Display Account Package    | 41 |
| 4.1.3 Add Recipe Package         | 46 |
| 4.1.4 Recipe Interaction Package | 51 |
| 4.1.5 Search Recipes Package     | 56 |
| 4.2 Medium Priority              | 60 |
| 4.2.1 My Ingredients Package     | 60 |
| 4.3 Low Priority                 | 64 |
| 4.3.1 Favorites Package          | 64 |
| Παράρτημα Ι – Γλωσσάριο          | 67 |



## Λίστα Σχημάτων

Σχήμα 1. Λεζάντα Σχήματος

|                                 |    |
|---------------------------------|----|
| Εικόνα 1: Home Page             | 31 |
| Εικόνα 2: Log In                | 31 |
| Εικόνα 3: Register              | 32 |
| Εικόνα 4: Recipes               | 32 |
| Εικόνα 5: Filters               | 33 |
| Εικόνα 6: Recipe (1)            | 33 |
| Εικόνα 7: Recipe (2)            | 33 |
| Εικόνα 8: My Ingredients        | 34 |
| Εικόνα 9: Search Ingredients    | 34 |
| Εικόνα 10: Add Recipe           | 35 |
| Εικόνα 11: Account              | 35 |
| Εικόνα 12: Edit Account Details | 36 |
| Εικόνα 13: Favorites            | 36 |
| Εικόνα 14: Remove Favorite      | 37 |

## Περιγραφή Εφαρμογής

Η κεντρική ιδέα της εφαρμογής μας είναι ότι ο χρήστης θα καταχωρεί σε μία λίστα τα υλικά που διαθέτει και θα μπορεί να αναζητά συνταγές που μπορούν να εκτελεστούν με τα υλικά αυτά.

Link για Διαγράμματα Σεναρίων Χρήσης:

<https://drive.google.com/file/d/1paQNDLak9QxOpcl4oGKYU7oLLKmWShhX/view?usp=sharing>



# 1 Απαιτήσεις Συστήματος

## 1.1 Functional requirements(Usage scenarios)

### <FR- 1>

The user must be able to **create** and **edit** an Account.

**Description** : The user must be able to enter their username, email, password, profile picture in order to create their profile. They must also be able to edit this information.

**User Priority** : High. It is essential that the users create an Account so that they can access their profile at any time. It is also important for them to edit their information.

**Technical Priority** : High. The creation of an Account is necessary for the system so that the registered users can start browsing the application.

### <FR- 2>

The user must be able to **edit** the list of My Ingredients.

**Description**: The user must be able to edit the list of available ingredients as the quantity may change.

**User Priority** : High. It is essential that the users edit the list of available ingredients at any time.

**Technical Priority** : High. The list of available ingredients is necessary for the operation of the system.

### <FR- 3>

The user must be able to **search** My Ingredients.

**Description**: The user must be able to search the list of available ingredients using keywords.

**User Priority** : High. It is essential that the users can easily find the ingredients' quantity that they want to edit.

**Technical Priority** : Low. The search for ingredients is not necessary for the operation of the system.

### <FR- 4>

The user must be able to **search** for Recipes.

**Description** : The users must be able to search all the recipes or the ones that they have in their favorites.

**User Priority** : High. Recipe search is fundamental for the user.

**Technical Priority** : High. Recipe search is a fundamental functionality of the system.

### <FR - 5>

The user must be able to **select** a Recipe.

**Description** : The user must be able to select a recipe in order to see its details.



12/04/2021

---

**User Priority** : High. It is essential that the user select a recipe so that they see its details and the cooking instructions.

**Technical Priority** : High. The selection of the recipe is a fundamental functionality of the system.

#### <FR - 6>

The user must be able to **select** certain filters.

**Description** : The user must be able to select certain filters in order to search for recipes according to their food preferences and to the list of available ingredients.

**User Priority** : High. Searching for recipes using filters saves time and makes it much easier.

**Technical Priority** : Medium. Searching for recipes using filters is not a fundamental function of the system.

#### <FR - 7>

The user must be able to **edit** the list of Favourites.

**Description** : The user must be able to save a recipe to favorites so that it can be more easily accessed.

**User Priority** : High. Saving a recipe to favorites is very important for the user as they do not need to search for this recipe every time they use the web application.

**Technical Priority** : Medium. Saving a recipe to favorites is not necessary for the operation of the system.

#### <FR - 8>

The user must be able to **rate** a Recipe.

**Description** : The user must be able to rate a recipe from 1 to 5 stars

**User Priority** : High. Rating a recipe is important for the user as it makes it easier for them to decide which recipe to choose.

**Technical Priority** : Medium. Rating a recipe does not belong to the basic functionality of the system.

#### <FR - 9>

The user must be able to **comment** on a Recipe.

**Description** : The user must be able to comment on a Recipe.

**User Priority** : Medium. It is important for the user to share their opinion on a recipe.

**Technical Priority** : Low. Commenting on a recipe does not belong to the basic functionality of the system.

#### <FR - 10>

The user must be able to **add** a new Recipe.

**Description** : The user must be able to share a new Recipe after entering the details of their recipe (title, description, ingredients, photo, number of portions, category).

**User Priority** : High. It is important for the user to share their ideas by creating new recipes.



**Technical Priority** : High. The recipes stored in the database are essential for the functionality of the system, so adding more recipes is very important.

#### <FR - 11>

The user must be able to **remove** the Ingredients that were used in a Recipe.

**Description** : The user must be able to automatically remove the ingredients that were used in the cooking of a recipe from the list of available ingredients.

**User Priority** : Medium. Automatic removal of the ingredients is very useful to the user as it saves them time.

**Technical Priority** : Low. The automatic removal of the ingredients is not necessary for the operation of the system.

## 1.2 Users and external systems

### 1.2.1 Users

- **Unregistered user(Guest)**

The unregistered users have to create an account in order to use the application.

- **Registered user**

The registered users have to enter their email and password in order to log in and browse the application.

### 1.2.2 External System

- **Database**

The database will keep all the personal data of the users, their available ingredients, the list of the recipes they upload, their ratings and comments and the list of their favorite recipes.





## 1.3 Significant non-functional requirements

### 1.3.1 Performance requirements

#### <NFR- 1>

The system must have a response time of less than 100 ms.

**Description :** The system must have a specific time response, so that the user has a high quality application experience.

**User Priority :** High. It is quite important for the user to be able to quickly navigate between GUI's.

**Technical Priority :** Low. It is not important for the system to have a low time response.

### 1.3.2 Security Requirements

#### <NFR- 2>

User's personal data must be protected using encryption due to GDPR.

**Description :** The personal data of users must be stored encrypted, as violating the laws set by the GDPR imposes heavy fines on companies.

**User Priority :** High. It is important for the user to know that their personal information is handled by a reliable source under the rules of data interception.

**Technical Priority :** High. It is important for the system to be able to protect the personal information of the users and prevent their interception.

### 1.3.3 Portability Requirements

#### <NFR- 3>

The system must be accessible from desktops (Desktop mode) using Windows 8 and above, Linux or MacOS and mobile devices(Mobile Mode) using iOS or Android.

**Description :** The system must be functional in both Desktop and Mobile Mode even with the latest versions of operating systems.

**User Priority :** High. It is critical for the user to access the application from multiple types of devices.

**Technical Priority :** High. Accessibility from multiple types of devices is essential for the system in order to attract more users.

### 1.3.4 Interoperability Requirements

#### <NFR- 4>

The system must be able to store and retrieve data from a database.

**Description :** Data such as user information, recipes favorites etc are stored and organized in a database.

**User Priority :** Low. The method of storing and retrieving data is not an essential part of the user's experience.



**Technical Priority** : High. It is important for the application to use a powerful system for storing and retrieving data.

### 1.3.5 Scalability Requirements

#### <NFR- 5>

The system must allow up to 1000 users to be connected at the same time.

**Description** : The system must be properly designed to allow 1000 users to use the application at the same time.

**User Priority** : High. This requirement is necessary for the user as they must be able to log in to the application without waiting for other users to log out.

**Technical Priority** : High. The scalability of the system is a necessary part of its functionality.

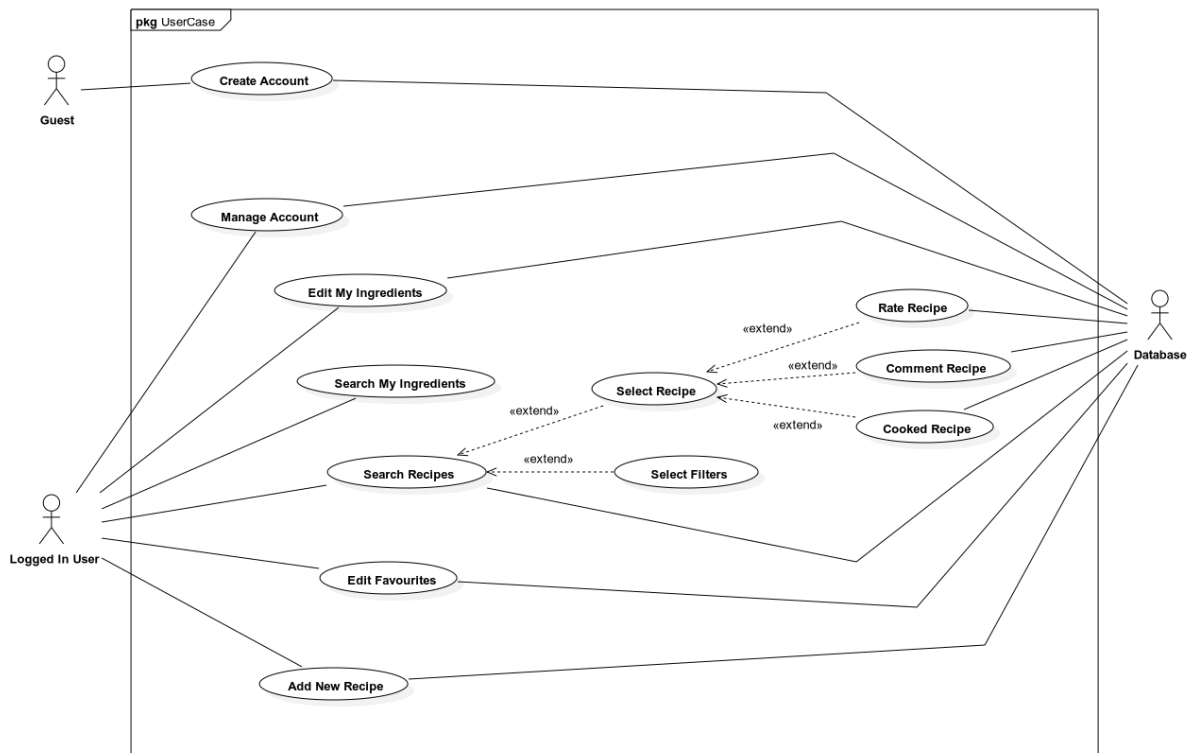
## 1.10 Lexicographical Definitions

- **User** : The person who uses the application.
- **Account**: The user's personal information (username, password, email, photo, ingredients, favorites, recipes)
- **My Ingredients**: The user's available ingredients and their quantity.
- **Recipe**: A recipe and its details (description, photo, title, ingredients, comments, rating, creator).
- **Favorites**: The list of user's favorite recipes.
- **System**: The essential parts that construct the application
- **Edit**: Add or remove items.
- **Filters**: Keywords that are selected in order to filter the recipes' search results.



## 2 Σενάρια Χρήσης

### 2.1 Διάγραμμα σεναρίων χρήσης



### 2.2 Feature 1: Create Account

#### Scenario: Successful Register

When I click Register

Then I am asked to enter my information

When I enter my information

|Username

|Password

|Email

Then I am asked to validate my email

When I validate my email

Then I am asked to edit My Ingredients

When I edit My Ingredients

And press the arrow button

Then I am directed to the Home Page



Hungry

Making cooking simple

### Scenario: Unsuccessful Register

When I click Register

Then I am asked to enter my information

|Username

|Password

|Email

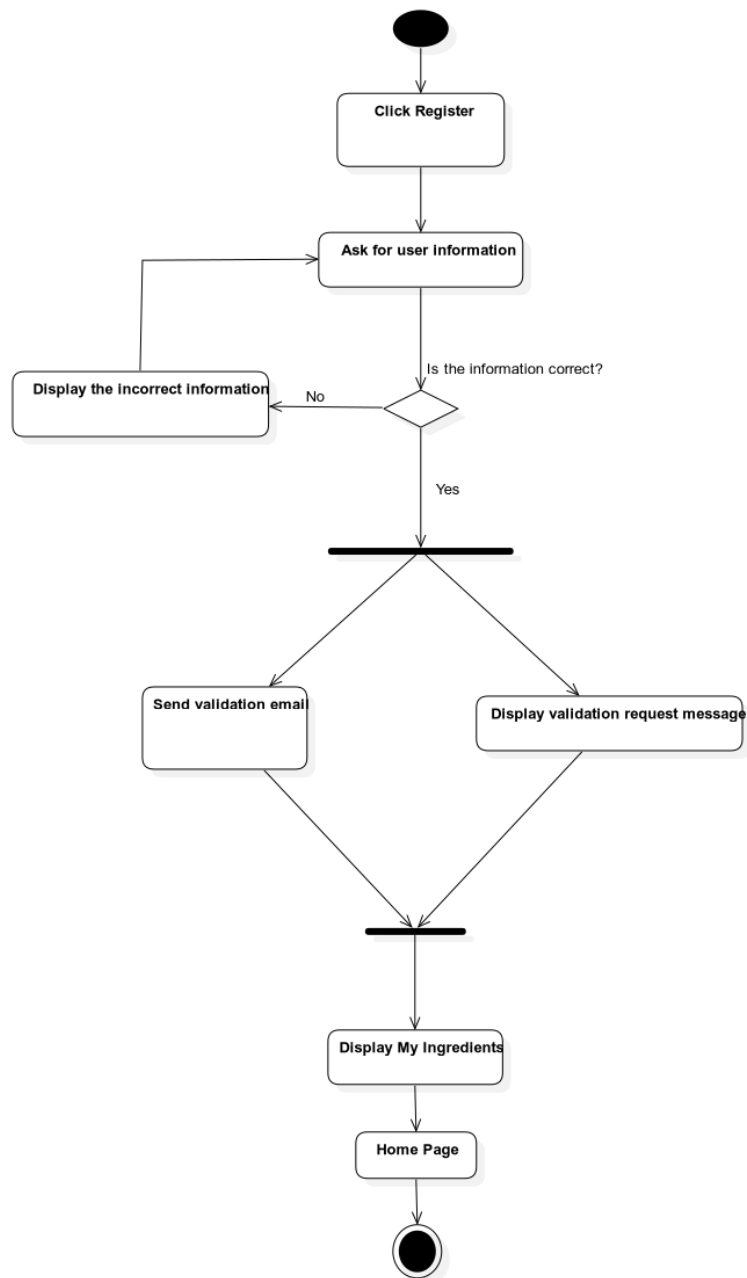
But my information is not correct

Then I am asked to reenter them



# Hungry

Making cooking simple



## 2.3 Feature 2: Edit Personal Information

### Background:

Given that I am logged in

### Scenario: Successful Edit

When I click Edit My Account



Then I am asked to choose what to edit

| Username

| Password

| Email

When I edit my information

And press Save

Then I am directed to the page of My Account

## Scenario: Unsuccessful Edit

When I click Edit My Account

Then I am asked to choose what to edit

| Username

| Password

| Email

When I edit my information

And press Save

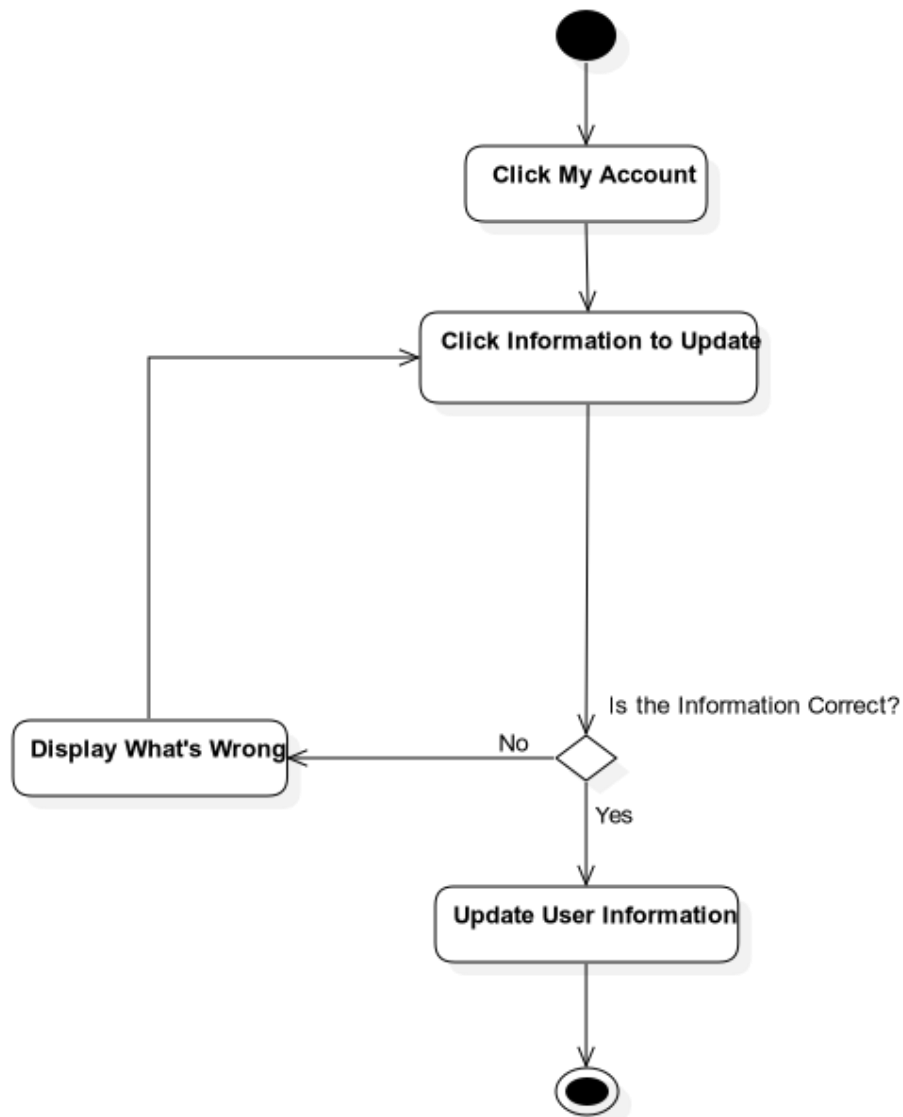
But my information is not correct

Then I am asked to reenter them



# Hungry

Making cooking simple





## 2.4 Feature 3: Edit My Ingredients

### Background:

Given that I am logged in

### Scenario: Edit with list

When I click My Ingredients

Then I am shown the list of all the ingredients

When I edit from the list of all the ingredients

Then my changes are saved

And I am shown the updated list of all the ingredients

### Scenario: Edit with search

When I click My Ingredients

Then I am shown the list of all the ingredients

When I click search

Then I am shown the search bar

When I fill in any word

Then I am shown all the ingredients that match the word

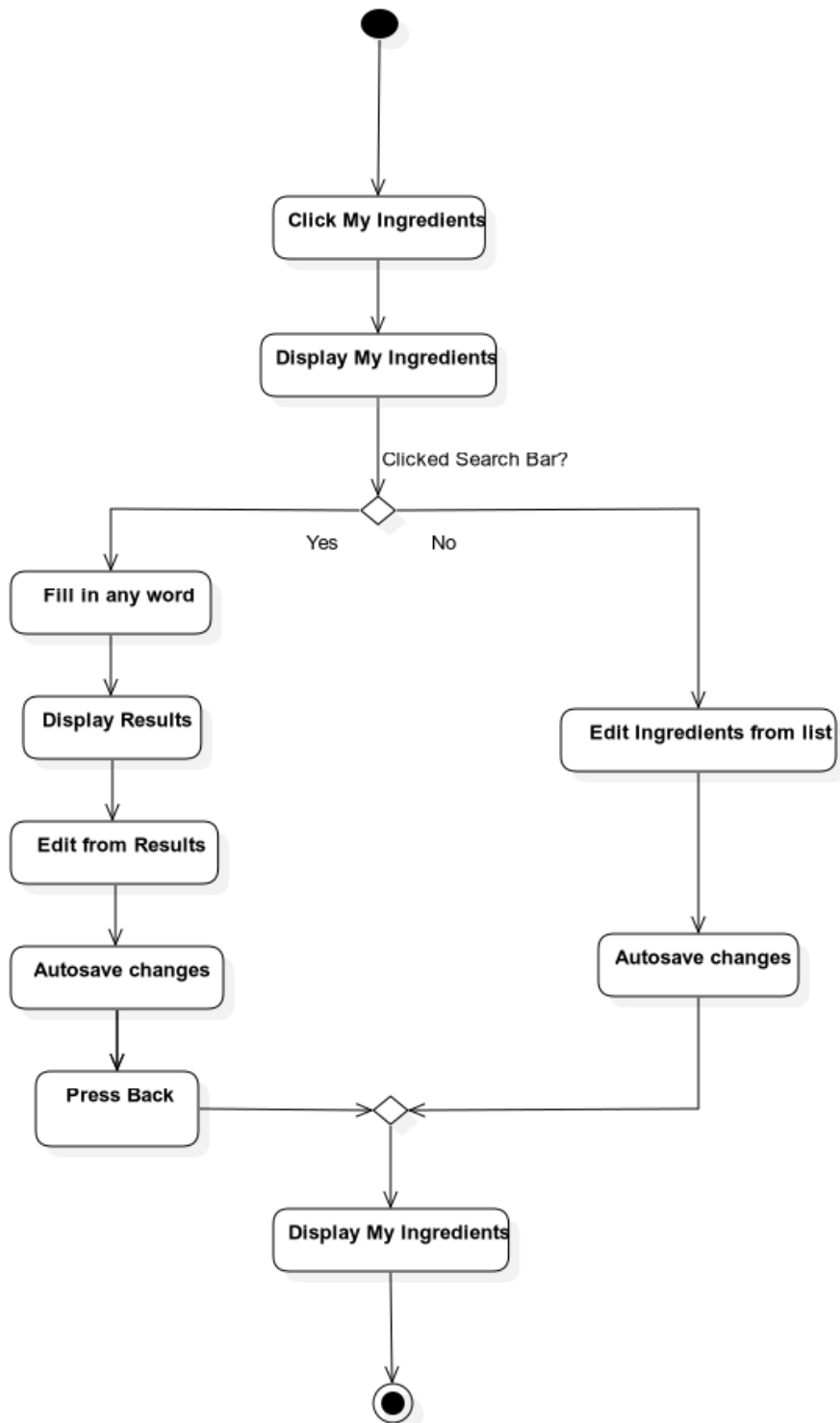
When I edit the ingredients from the results

Then my changes are saved

When I press back

Then I am shown the updated list of all the ingredients







## 2.5 Feature 4: Search Recipes

### Background:

Given that I am logged in

### Scenario: Find Recipe

When I click on Recipes

Then I see a list of Recipes

When I click on a Recipe

Then I am shown the Details of this Recipe

### Scenario: Find Recipe with Filters

When I click on Recipes

Then I see a list of all the Recipes

When I click Filters

Then I am shown all the filters

When I choose the filters that I want

And I close the Filters tab

Then I am shown a list of the Recipes that match the filters

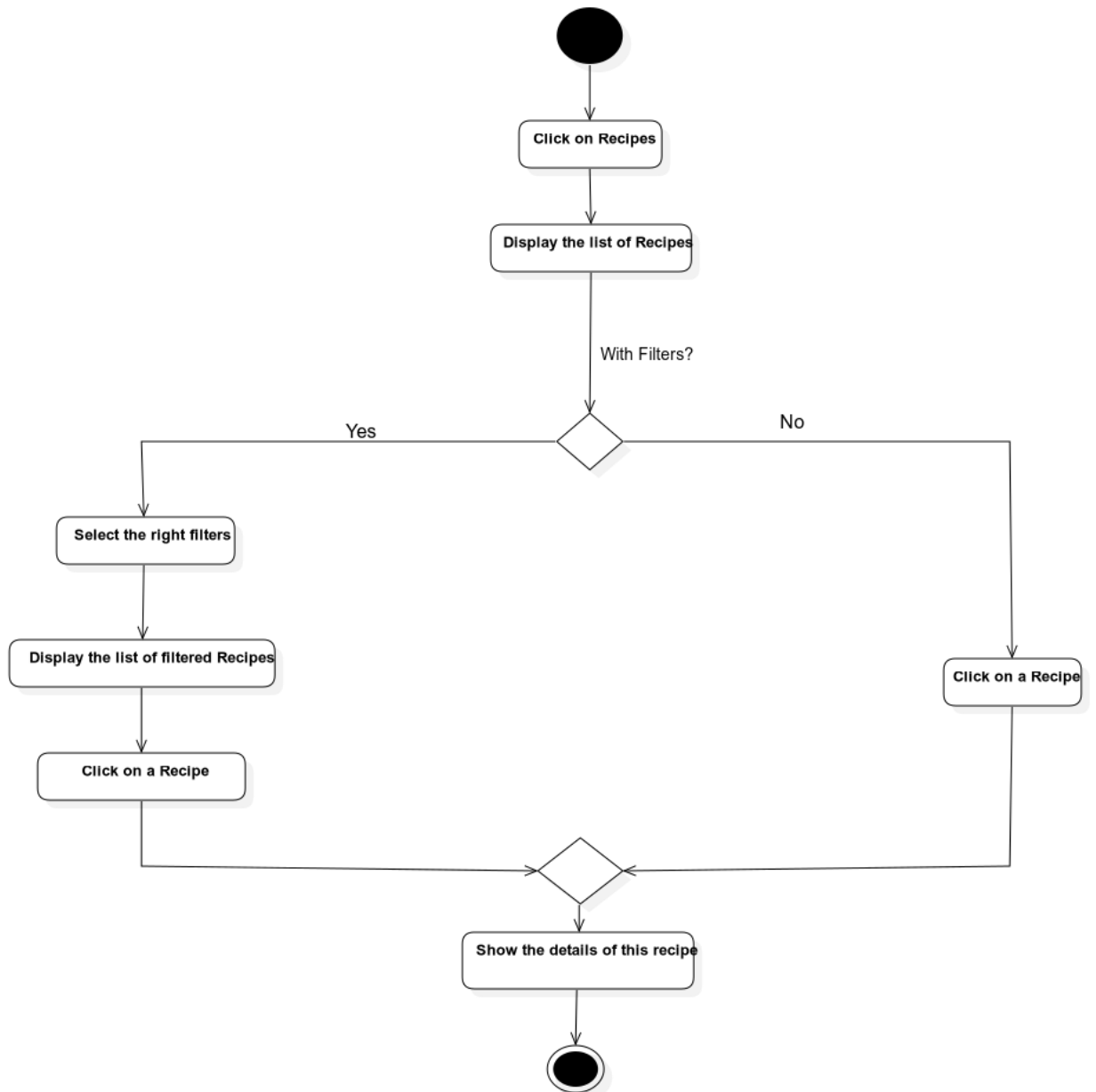
When I click on a Recipe

Then I am shown the Details of this Recipe



# Hungry

Making cooking simple





## 2.6 Feature 5: Edit Favorites from Recipe Details

### Background:

Given that I am logged in

And I have selected a Recipe

### Scenario: Add to Favorites

When I click the heart on top of the Recipe

And the heart is White

Then The Recipe is added to the Favorites List

And the heart on top of the Recipe becomes Red

And I am shown the Details of this Recipe

### Scenario: Remove from Favorites

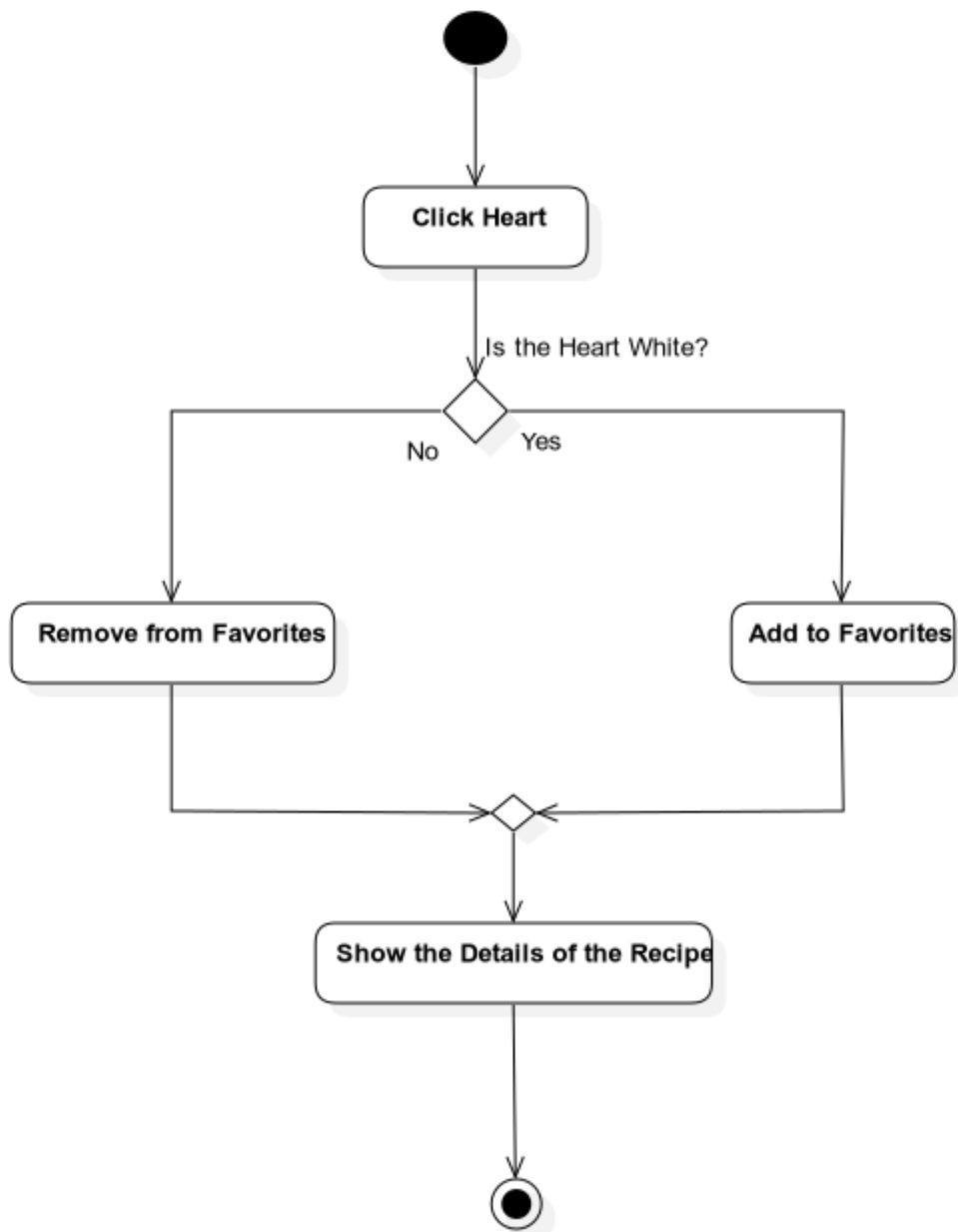
When I click the heart on top of the Recipe

And the heart is Red

Then The Recipe is removed from the Favorites List

And the heart on top of the Recipe becomes White

And I am shown the Details of this Recipe





## 2.7 Feature 6: Comment Recipe

### Background:

Given that I am logged in

And I have selected a Recipe

### Scenario: Comment Recipe

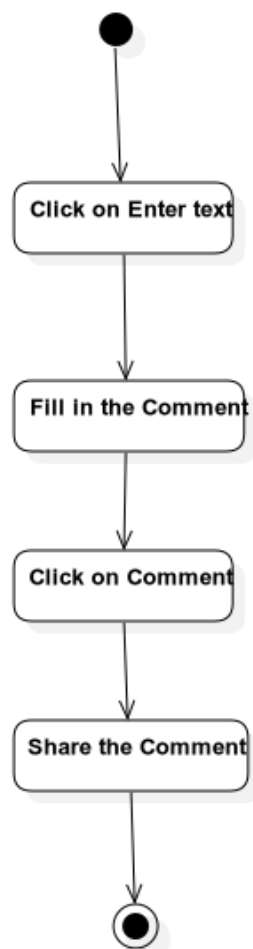
When I click on Enter text

Then I am allowed to write the comment

When I write the comment

And click on Comment

Then the comment is published





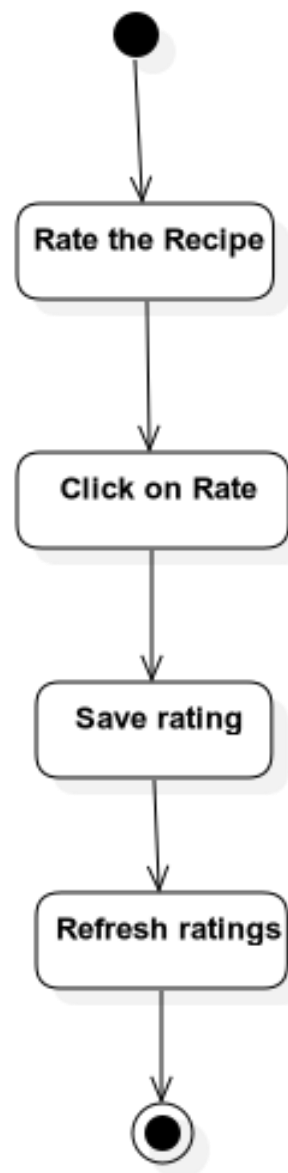
## 2.8 Feature 7: Rate Recipe

### Background:

Given that I am logged in  
And I have selected a Recipe

### Scenario: Rate Recipe

When I rate the Recipe  
And I click on Rate it  
Then my rating is saved





## 2.9 Feature 8: Cooked It Button

### Background:

Given that I am logged in

And I have selected a Recipe

### Scenario: Cooked it without missing ingredients

When I have selected a number of portions

And I click on Cooked it

And I have all the ingredients in my Ingredients

Then the ingredients of the Recipe are removed from My Ingredients

### Scenario: Cooked it with missing ingredients

When I have selected a number of portions

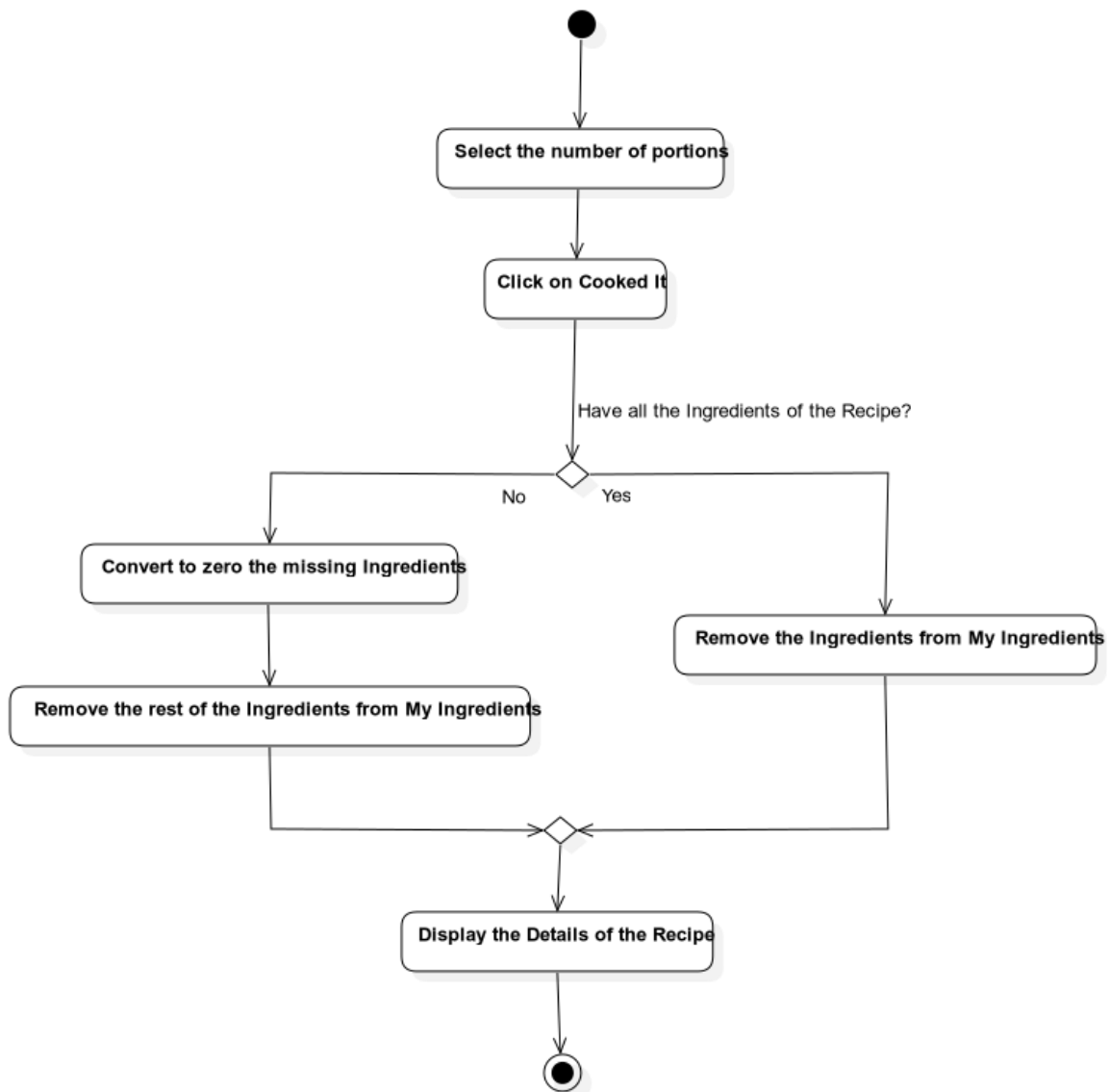
And I click on Cooked it

And I don't have all the ingredients in my Ingredients

Then the ingredients of the Recipe that are more than the ones from My Ingredients are converted to zero

And the rest of the ingredients are removed from My Ingredients







## 2.10 Feature 9: Select From Favorites

### Background:

Given that I am logged in

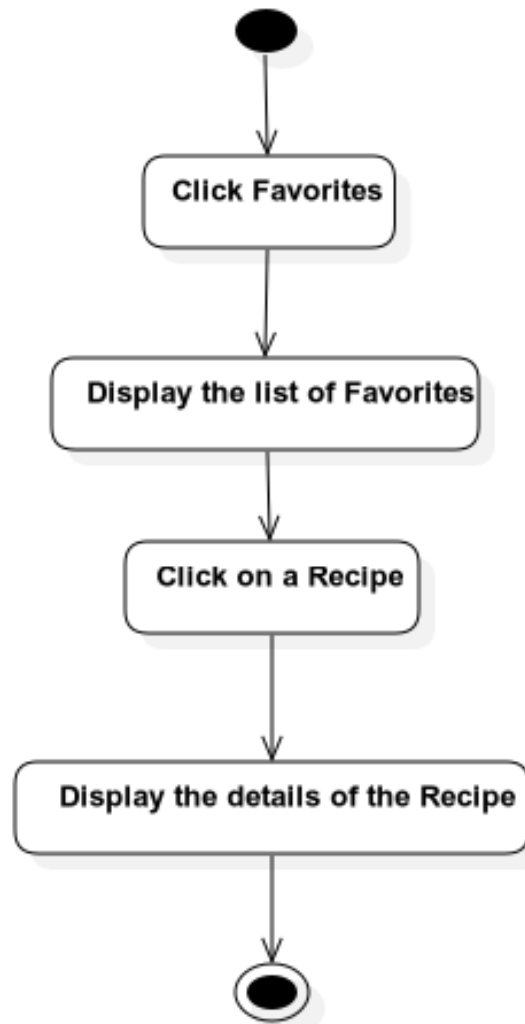
### Scenario: Selected from Favorites

When I have clicked on Favorites

Then I am shown a list of my Favorites

When I click on a Recipe

Then I am shown the Details of this Recipe





## 2.11 Feature 10: Remove From Favorites

### Background:

Given that I am logged in

### Scenario: Remove from Favorites

When I have clicked on Favorites

Then I am shown a list of my Favorites

When I click on the heart of the Recipe

Then I am shown the Are you Sure Dialog

When I click Yes

Then the Recipe is removed from my Favorites

### Scenario: Missclick from Favorites

When I have clicked on Favorites

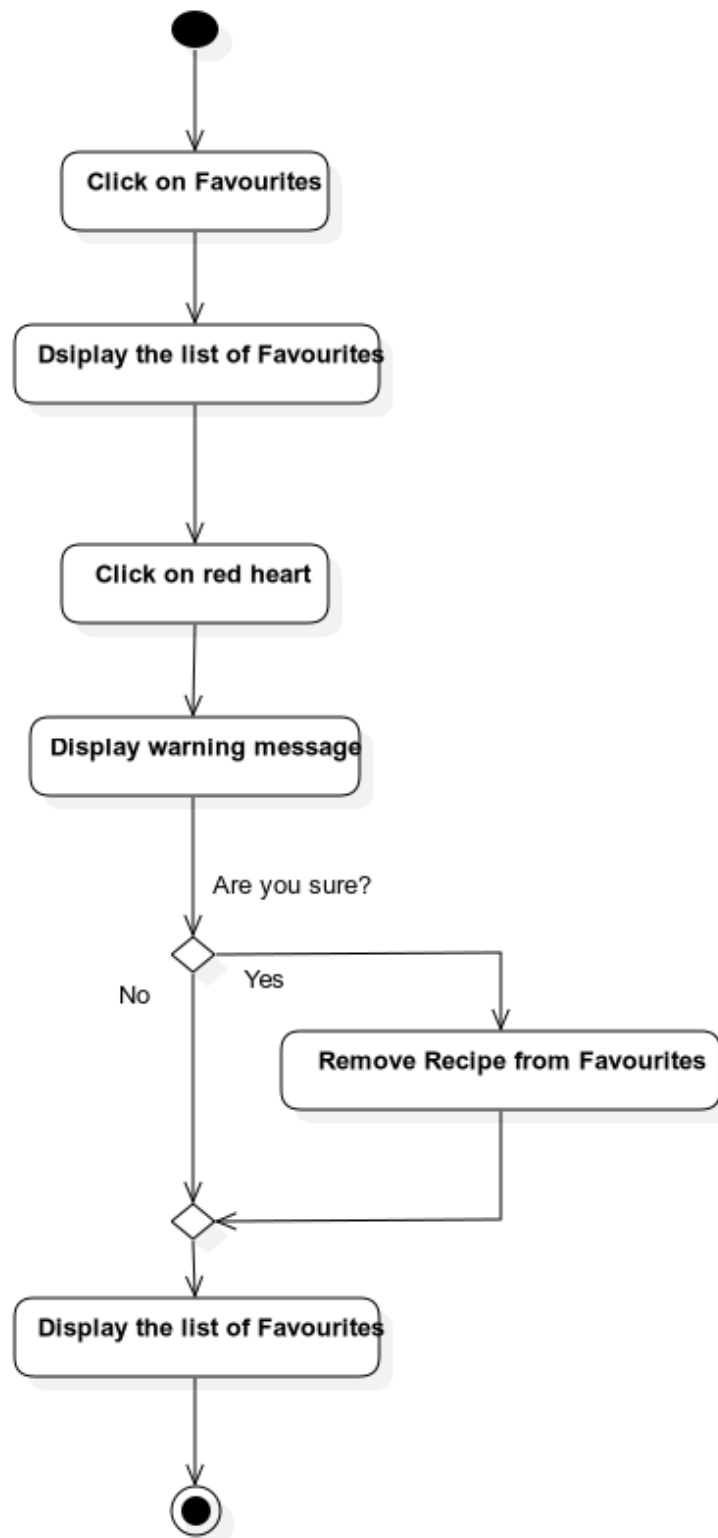
Then I am shown a list of my Favorites

When I click on the heart of the Recipe

Then I am shown the Are you Sure Dialog

When I click No

Then the Recipe is removed from my Favorites





## 2.12 Feature 11: Add New Recipe

### Scenario: Add Recipe without Picture

When I click add Recipe

Then I am shown a tab where I can add the details of a recipe

When I fill the details

- | Title
- | Ingredients
- | Description for the Recipe execution
- | Scalable button
- | Number of portions

Then I am asked to add the Recipe

And the Recipe is published

### Scenario: Add Recipe with Picture

When I click add Recipe

Then I am shown a tab where I can add the details of a recipe

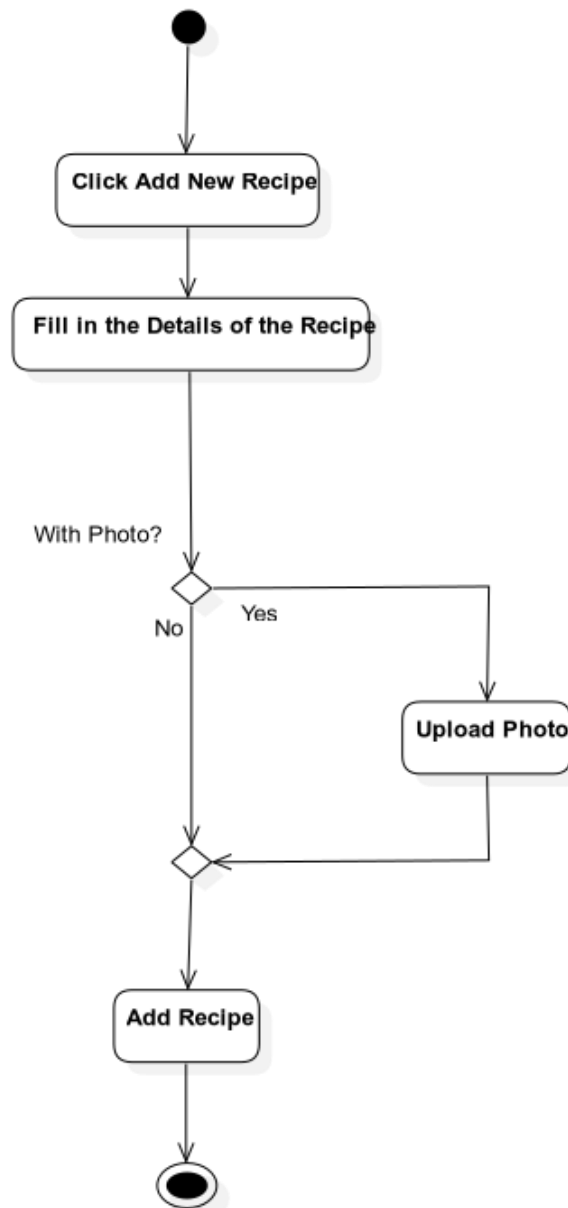
When I fill the details

- | Title
- | Ingredients
- | Description for the Recipe execution
- | Scalable button
- | Number of portions

And I upload the picture of the Recipe

Then I am asked to add the Recipe

And the Recipe is published





## 3 Επιδεικτικά γραφικά παράθυρα διεπαφής

### 3.1 Home Page



1: The Home Page where the main functions of the app are displayed

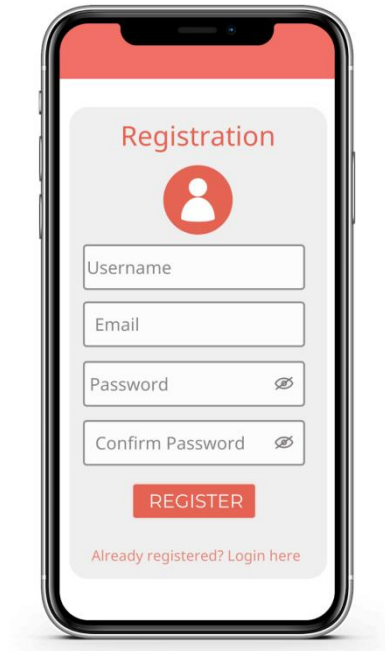
### 3.2 Login



2: The login page of the app where the user can Login

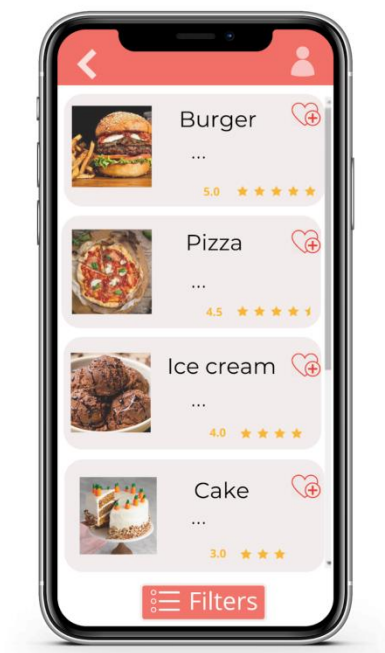


### 3.3 Register



3: The Registration Page of the app where the user can Register

### 3.4 Recipes



4: The Recipe Page of the app where all the available Recipes are displayed





# Hungry

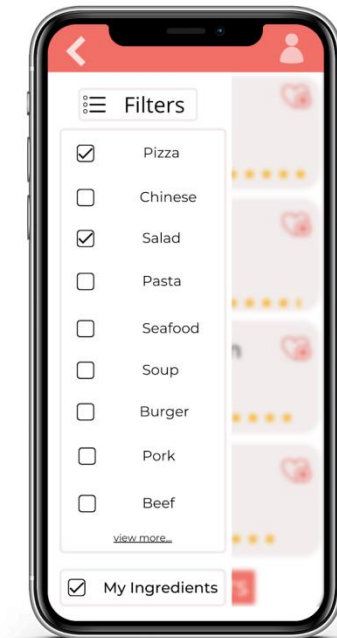
Making cooking simple

Έγγραφο Απαιτήσεων Χρηστών

εκδ. 0.1

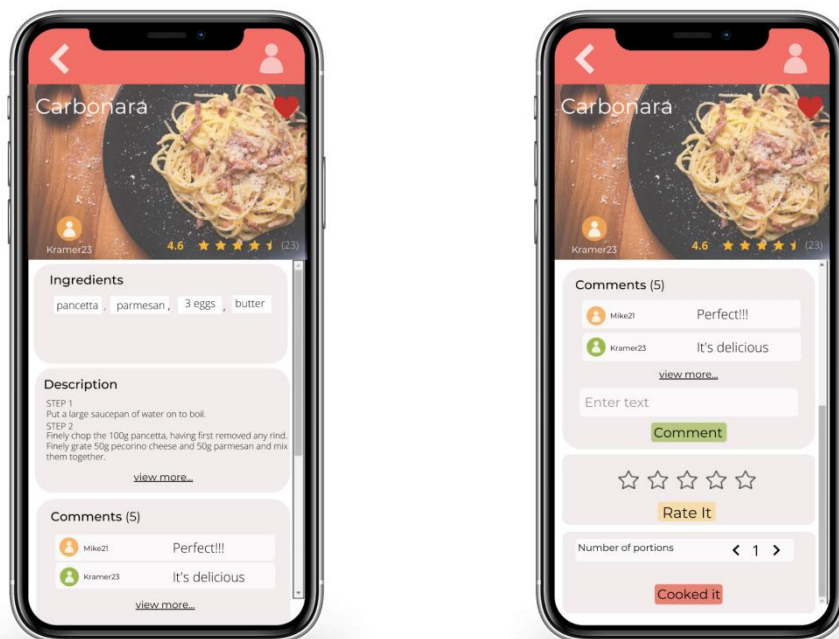
12/04/2021

## 3.5 Filters



5: The Filters page of the app where the user can select filters

## 3.6 Recipe



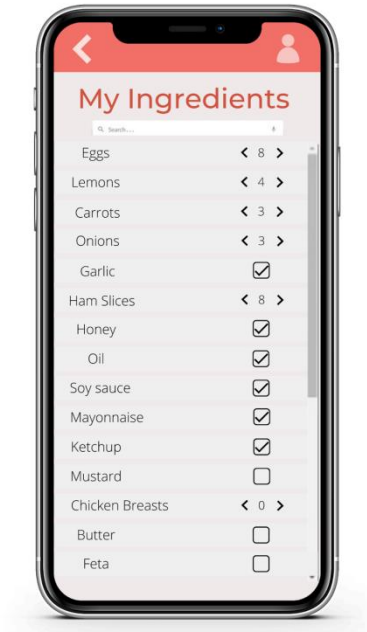
6,7: The Recipe page of the app where all the details of the chosen Recipe are displayed



# Hungry

Making cooking simple

## 3.7 My Ingredients



8: The My Ingredients page of the app where all the available Ingredients are displayed in order to be selected and modified by the user

## 3.8 Search Ingredients



9: The Search Ingredients page of the app where the user can search for a specific ingredient



# Hungry

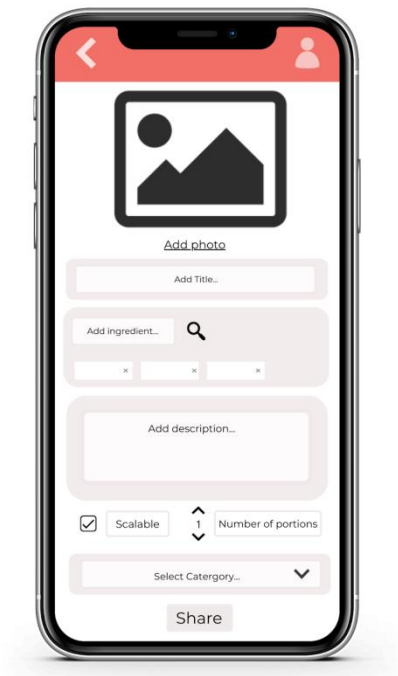
Making cooking simple

Έγγραφο Απαιτήσεων Χρηστών

εκδ. 0.1

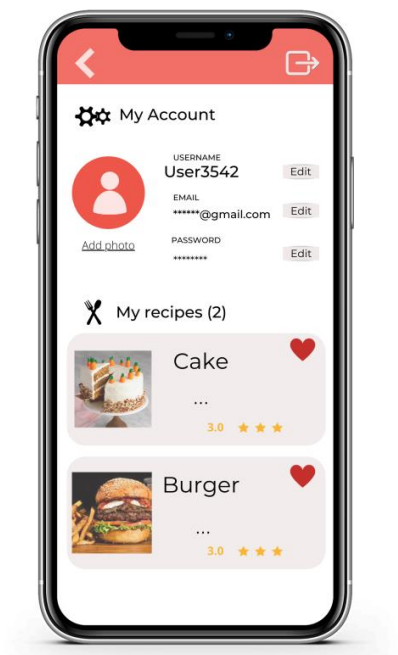
12/04/2021

## 3.9 Add Recipe



10: The Add Recipe page of the app where the user can add their own Recipe

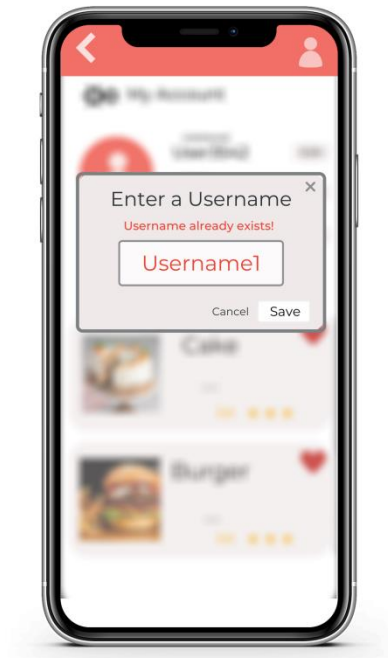
## 3.10 Account



11: The Account page of the app where the user can view their account details

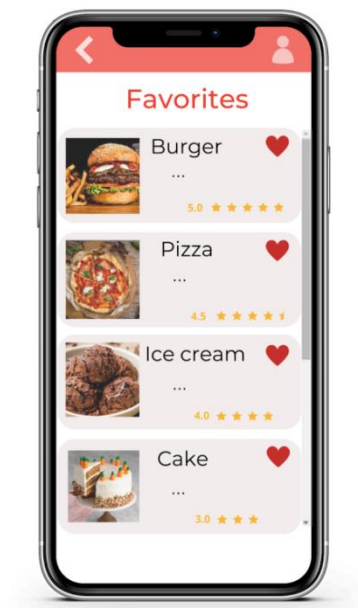


### 3.11 Edit Account Details



12: The EditAccount page of the app where the user can modify their own details

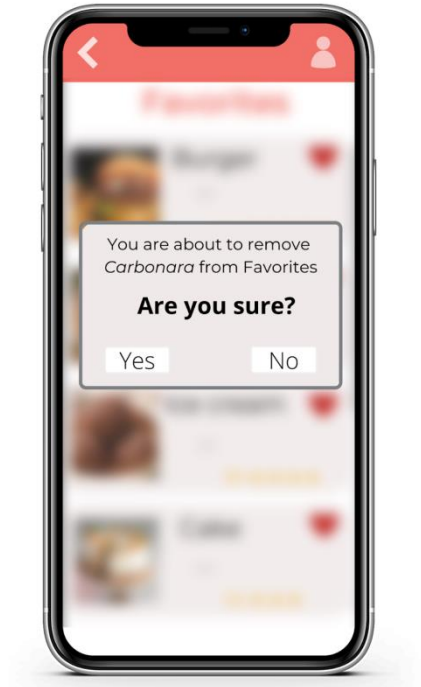
### 3.12 Favorites



13: The Favorites Page of the app where the user can view their favorite recipes



### 3.13 Remove Favorite



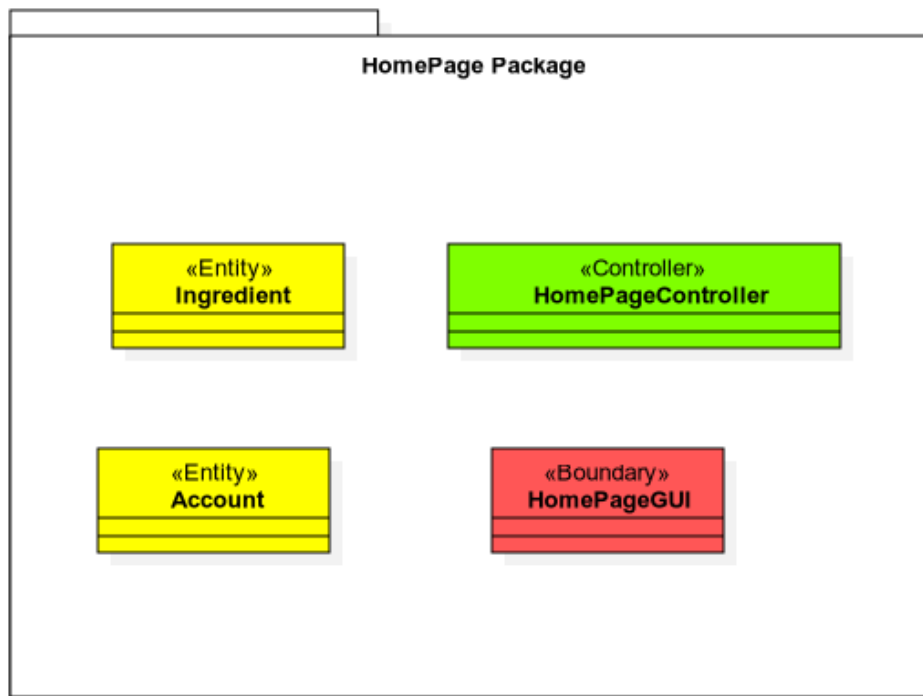
14: The Remove Favorite page of the app



## 4 Στατική Μοντελοποίηση

### 4.1 High Priority

#### 4.1.1 HomePage Package



4.1: HomePage Package

#### Entity Ingredient

Η οντότητα αυτή περιέχει τα χαρακτηριστικά ενός συστατικού.

Ιδιότητες της κλάσης Ingredient:

- name: Το όνομα του συστατικού.
- isScalable: Άμα είναι ένα μετρήσιμο συστατικό (πχ μήλα), ή μη μετρήσιμο (πχ αλάτι).
- quantity: Η ποσότητα σε περίπτωση που είναι μετρήσιμο και η ύπαρξη του ή όχι αν δεν είναι μετρήσιμο.

Μέθοδοι κλάσης Ingredient:

Ως κλάση entity, οι μέθοδοι της κλάσης απαρτίζονται από έναν constructor, καθώς και getters και setters για τις ιδιότητες της.

#### Entity Account



Η οντότητα αυτή περιγράφει τον συνδεδεμένο χρήστη. Αυτό το αντικείμενο θα το έχουν όλοι οι controller προκειμένου να μην χρειάζεται περιττή επικοινωνία με την βάση δεδομένων.

Ιδιότητες της κλάσης Account είναι:

- username: Το ψευδώνυμο του χρήστη
- password: Ο κωδικός του χρήστη που θα χρησιμοποιεί για να μπει στην εφαρμογή
- email: Το ηλεκτρονικό mail του χρήστη
- accountID: Το ID που έχει ο λογαριασμός στο database
- photo: Η εικόνα του avatar του χρήστη
- myRecipesID: Μια λίστα από τις συνταγές που έχει μοιραστεί ο χρήστης
- favoritesID: Μια λίστα από τις αγαπημένες συνταγές του χρήστη
- MyIngredients: Μια λίστα από τα συστατικά που διαθέτει ο χρήστης

Ως κλάση entity, οι μέθοδοι της κλάσης απαρτίζονται από έναν constructor, καθώς και getters και setters για τις ιδιότητες της.

### **Controller HomeController**

Η κλάση αυτή περιέχει τις μεθόδους που ελέγχουν το περιβάλλον της αρχικής σελίδας.

Η μόνη ιδιότητα της κλάσης HomeController είναι το αντικείμενο Account.

Μέθοδοι της κλάσης είναι:

- HomeController(account: Account): ο constructor, που δέχεται σαν όρισμα τον λογαριασμό του χρήστη
- displayAccount(): η συνάρτηση αυτή μεταβιβάζει τον χρήστη στο λογαριασμό του δημιουργώντας τον controller AccountController
- displayAddRecipe(): η συνάρτηση αυτή μεταβιβάζει τον χρήστη στην σελίδα που μπορεί να προσθέσει μια συνταγή δημιουργώντας τον controller AddRecipeController
- displayFavorites(): η συνάρτηση αυτή μεταβιβάζει τον χρήστη στην σελίδα που περιέχει τα αγαπημένα του δημιουργώντας τον controller FavoritesController
- displayMyIngredients(): η συνάρτηση αυτή μεταβιβάζει τον χρήστη στην σελίδα που περιέχει τα συστατικά του δημιουργώντας τον controller MyIngredientsController



- `displayRecipes()`: η συνάρτηση αυτή μεταβιβάζει τον χρήστη στην σελίδα που μπορεί να αναζητήσει συνταγές δημιουργώντας τον controller `DisplayRecipesController`

### **Boundary HomePageGUI**

Η κλάση αυτή αντιπροσωπεύει το γραφικό περιβάλλον της αρχικής σελίδας. Είναι η κύρια σελίδα του παραγράμματος από όπου ο χρήστης μπορεί να περιηγηθεί στην υπόλοιπη εφαρμογή.

Ιδιότητες της κλάσης `HomePageGUI`:

- `account`: Το κουμπί στο πάνω αριστερό μέρος της σελίδας που θα μεταβιβάζει τον χρήστη στον λογαριασμό του
- `logo`: Η φωτογραφία του λογότυπου με το σλόγκαν κεντρικό πάνω μέρος της σελίδας
- `myIngredients`: Το κουμπί που θα μεταβιβάζει τον χρήστη στα συστατικά του
- `favorites`: Το κουμπί που θα μεταβιβάζει τον χρήστη στις αγαπημένες συνταγές του
- `addRecipe`: Το κουμπί που θα μεταβιβάζει τον χρήστη στην σελίδα που μπορεί να προσθέσει συνταγή
- `recipes`: Το κουμπί που θα μεταβιβάζει τον χρήστη στην αναζήτηση συνταγών

Μέθοδοι της κλάσης `HomePageGUI`:

- `HomePageGUI()`: ο constructor του αντικειμένου
- `onAccount()`: Καλεί την συνάρτηση `displayAccount(account: Account)` από τον ελεγκτή `HomePageController` για την μεταβίβαση του χρήστη στον λογαριασμό του
- `onMyIngredients()`: Καλεί την συνάρτηση `displayMyIngredients(account: Account)` από τον ελεγκτή `HomePageController` για την μεταβίβαση του χρήστη στα συστατικά του
- `onRecipes()`: Καλεί την συνάρτηση `displayRecipes(account: Account)` από τον ελεγκτή `HomePageController` για την μεταβίβαση του χρήστη στην αναζήτηση συνταγών
- `onAddRecipe()`: Καλεί την συνάρτηση `displayAddRecipe(account: Account)` από τον ελεγκτή `HomePageController` για την μεταβίβαση του χρήστη στην σελίδα που μπορεί να προσθέσει συνταγή
- `onFavorites()`: Καλεί την συνάρτηση `displayFavorites(account: Account)` από τον ελεγκτή `HomePageController` για την μεταβίβαση του χρήστη στην σελίδα που περιέχει τα αγαπημένα του





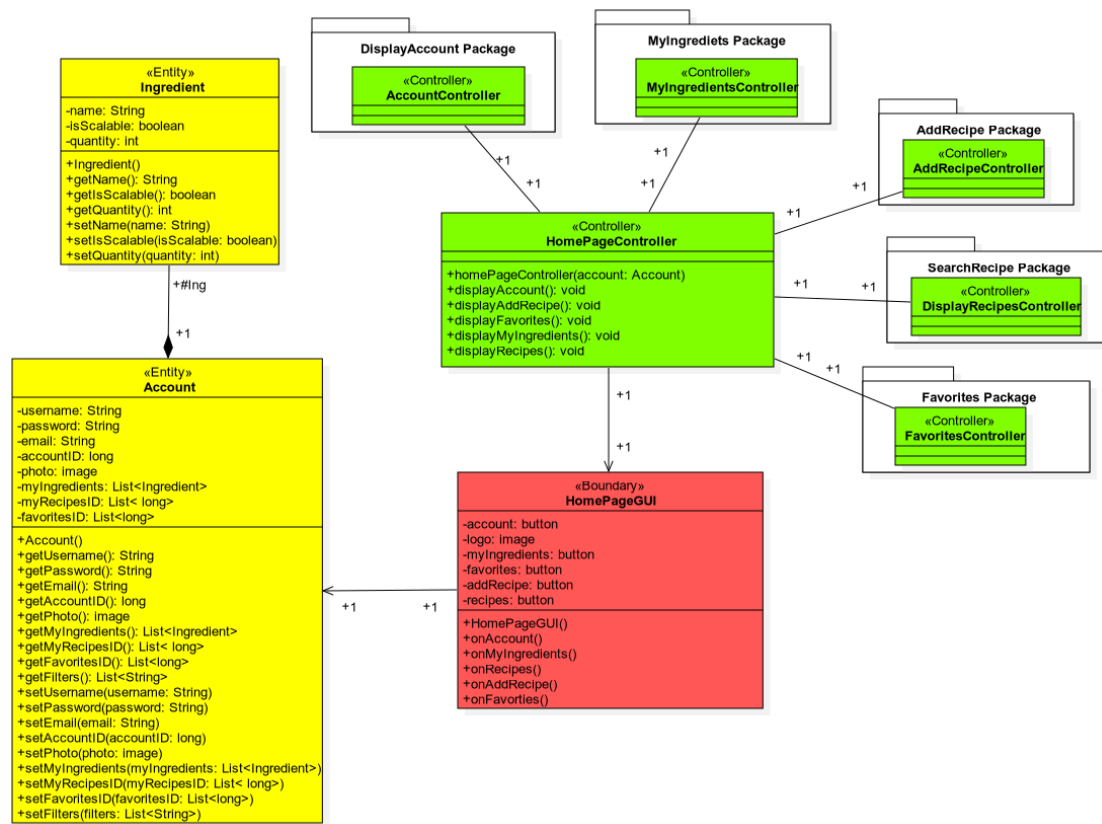
# Hungry

Making cooking simple

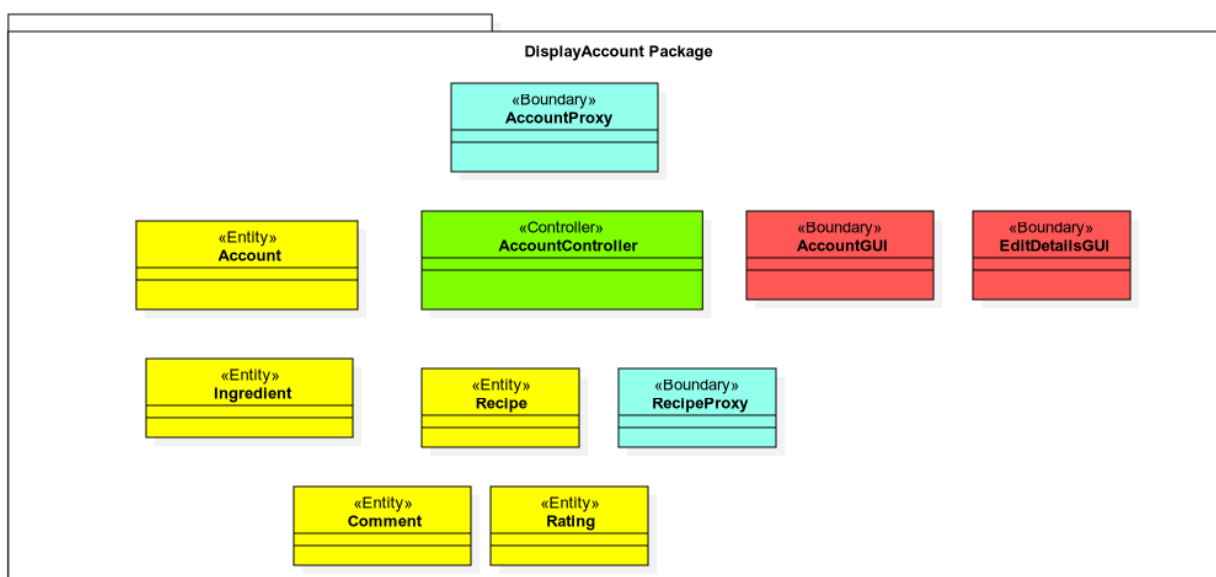
Έγγραφο Απαιτήσεων Χρηστών

εκδ. 0.1

12/04/2021



## 4.1.2 DisplayAccount Package





---

### **Boundary AccountProxy**

Η κλάση αυτή χρησιμοποιείται για την επικοινωνία της εφαρμογής με την βάση δεδομένων, ώστε να συνδέονται και να εγγράφονται χρήστες καθώς και να ανανεώνονται δεδομένα χρηστών.

Ως κλάση μεσολαβητή, η κλάση AccountProxy δεν έχει ιδιότητες.

Μέθοδοι της κλάσης AccountProxy:

- addUser(account: Account): η συνάρτηση αυτή εγγράφει τον χρήστη στη βάση δεδομένων. Επιστρέφει boolean, για το αν εκτελέστηκε επιτυχώς η συνάρτηση.
- loadUser(account: Account): η συνάρτηση αυτή συνδέει τον χρήστη στην εφαρμογή. Επιστρέφει boolean, για το αν εκτελέστηκε επιτυχώς η συνάρτηση.
- editUser(account: Account): η συνάρτηση αυτή επεξεργάζεται έναν ήδη υπάρχων χρήστη στη βάση δεδομένων. Επιστρέφει boolean, για τον αν εκτελέστηκε επιτυχώς η συνάρτηση.

### **Controller AccountController**

Η κλάση αυτή περιέχει τις μεθόδους που ελέγχουν το περιβάλλον με τα δεδομένα του λογαριασμού του χρήστη.

Ιδιότητες της κλάσης AccountController:

- account: Αντικείμενο τύπου Account.
- recipes: Λίστα αντικειμένων τύπου Recipe.

Μέθοδοι της κλάσης AccountController είναι:

- AccountController(account: Account): ο constructor, που δέχεται σαν όρισμα τον λογαριασμό του χρήστη
- onScroll(): Η αναγνώριση της περιήγησης του χρήστη σε όλη τη σελίδα
- logout(): Αποσύνδεση του χρήστη από την εφαρμογή
- addPhoto(photo: image): Προσθήκη ή ανανέωση της φωτογραφίας του χρήστη
- editUsername(username: String): Αλλαγή του ψευδώνυμου του χρήστη
- editEmail(email: String): Αλλαγή της ηλεκτρονικής διεύθυνσης του χρήστη
- editPassword(password: String): Αλλαγή του συνθηματικού του χρήστη
- goBack(): Μετάβαση του χρήστη στην προηγούμενη σελίδα.
- displayRecipes(): Η συνάρτηση η οποία φορτώνει αυτόματα την λίστα των συνταγών που έχει δημιουργήσει ο χρήστης από τη λίστα myRecipesID. Επιστρέφει boolean για το αν εκτελέστηκε επιτυχώς η συνάρτηση.



- `displayRecipe(recipeID: long)`: Η συνάρτηση η οποία μεταβιβάζει τον χρήστη στις λεπτομέρειες της συνταγής που επέλεξε, δημιουργώντας τον controller `recipeInteractionController`.
- `closePopUp()`: Η συνάρτηση η οποία κλείνει το παράθυρο `EditDetailsGUI`.
- `saveDetail(type: String, detail: String)`: Η συνάρτηση η οποία ελέγχει και αποθηκεύει το νέο email/password/username. Επιστρέφει boolean για το αν εκτελέστηκε επιτυχώς η συνάρτηση.
- `TextBox()`: Η συνάρτηση που επιτρέπει στον χρήστη να εισάγει κείμενο.

### **Boundary AccountGUI**

Η κλάση αυτή αντιπροσωπεύει το γραφικό περιβάλλον της σελίδας όπου ο χρήστης μπορεί να δει και να επεξεργαστεί τα προσωπικά του δεδομένα.

Ιδιότητες της κλάσης `AccountGUI`:

- `logout`: Το κουμπί που αποσυνδέει τον χρήστη από τον λογαριασμό του.
- `back`: Το κουμπί που μεταβιβάζει τον χρήστη στην προηγούμενη σελίδα.
- `accountLabel`: label
- `userPhoto`: Η φωτογραφία που έχει ανεβάσει ο χρήστης
- `addPhoto`: Το κουμπί που θα επιτρέπει στον χρήστη να αλλάζει την φωτογραφία του.
- `username`: label
- `password`: label
- `email`: label
- `editUsername`: Το κουμπί που θα επιτρέπει στον χρήστη να αλλάζει το ψευδώνυμο του.
- `editPassword`: Το κουμπί που θα επιτρέπει στον χρήστη να αλλάζει τον κωδικό του.
- `editEmail`: Το κουμπί που θα επιτρέπει στον χρήστη να αλλάζει το email του.
- `recipeLabel`: label
- `recipeFrames`: Μία λίστα από πλαίσια μέσα στα οποία αναγράφονται οι πληροφορίες των συνταγών του χρήστη.
- `recipePhotos`: Μία λίστα από τις φωτογραφίες των συνταγών του χρήστη.
- `recipeTitle`: Μία λίστα από labels με τα ονόματα των συνταγών.
- `recipeRating`: Μία λίστα από labels με τις βαθμολογίες των συνταγών.
- `recipeHeart`: Μία λίστα από labels.



Μέθοδοι της κλάσης AccountGUI:

- AccountGUI(): ο constructor του αντικειμένου
- onScroll(): Καλεί την συνάρτηση scrollPage() από το AccountController.
- onEditUsername(): Καλεί την συνάρτηση editUsername(username: String) από το AccountController.
- onEditPassword(): Καλεί την συνάρτηση editPassword(password: String) από το AccountController.
- onEditEmail(): Καλεί την συνάρτηση editEmail(email: String) από το AccountController.
- onAddPhoto(): Καλεί την συνάρτηση addPhoto(photo: image) από το AccountController.
- onRecipeFrame(): Καλεί την συνάρτηση displayRecipe(recipeID: long) από το AccountController.
- onBack(): Καλεί την συνάρτηση goBack() από τον ελεγκτή AccountController για την μεταβίβαση του χρήστη στην προηγούμενη σελίδα.

### **Boundary EditDetailsGUI**

Η κλάση αυτή αντιπροσωπεύει το γραφικό περιβάλλον της σελίδας όπου ο χρήστης έχει επιλέξει να επεξεργαστεί τα προσωπικά του δεδομένα.

Ιδιότητες της κλάσης EditDetailsGUI:

- detailLabel: Το χαρακτηριστικό που επέλεξε ο χρήστης να επεξεργαστεί
- detailTextBox: Το textbox μέσα στο οποίο ο χρήστης γράφει το νέο email/password/username.
- close: Το κουμπί που κλείνει το παράθυρο επεξεργασίας.
- save: Το κουμπί που ελέγχει την ορθότητα του νέου email/password/username και ενημερώνει το database.

Μέθοδοι της κλάσης EditDetailsGUI:

- EditDetailsGUI(type: String): ο constructor του αντικειμένου
- onClose(): Καλεί την συνάρτηση closePopUp() από το AccountController.
- onSave(): Καλεί την συνάρτηση saveDetail(type: String, detail: String) από το AccountController.
- onTextBox(): Καλεί την συνάρτηση textBox() από το AccountController.



12/04/2021

- 
- ```

classDiagram
    class Account {
        <<Entity>>
        -username: String
        -password: String
        -email: String
        -accountID: long
        -photo: Image
        -myRecipesID: List<long>
        -favoritesID: List<long>
        +Account()
        +getUsername(): String
        +getPassword(): String
        +getEmail(): String
        +getAccountID(): long
        +getPhoto(): Image
        +getMyRecipesID(): List<long>
        +getFavoritesID(): List<long>
        +setUsername(username: String)
        +setPassword(password: String)
        +setEmail(email: String)
        +setAccountID(accountID: long)
        +setPhoto(photo: Image)
        +setMyRecipesID(myRecipesID: List<long>)
        +setFavoritesID(favoritesID: List<long>)
    }

    class Ingredient {
        <<Entity>>
        -name: String
        -isScalable: boolean
        -quantity: int
        +Ingredient()
        +getName(): String
        +getIsScalable(): boolean
        +getQuantity(): int
        +setName(name: String)
        +setIsScalable(isScalable: boolean)
        +setQuantity(quantity: int)
    }

    class Recipe {
        <<Entity>>
        -title: String
        -photo: Image
        -recipeID: long
        -creator: String
        -filter: String
        -description: String
        -isScalable: boolean
        -portions: int
        +Recipe()
        +getTitle(): String
        +getPhoto(): Image
        +getRecipeID(): long
        +getCreator(): String
        +getFilter(): String
        +getDescription(): String
        +getIsScalable(): boolean
        +getPortions(): int
        +setTitle(title: String)
        +setPhoto(photo: Image)
        +setRecipeID(recipeID: long)
        +setCreator(creator: String)
        +setFilter(filter: String)
        +setDescription(description: String)
        +setIsScalable(isScalable: boolean)
        +setPortions(portions: int)
    }

    class Comment {
        <<Entity>>
        -text: String
        -username: String
        -photo: Image
        +getText(): String
        +getUsername(): String
        +getPhoto(): Image
        +setText(text: String)
        +setUsername(username: String)
        +setPhoto(photo: Image)
    }

    class Rating {
        <<Entity>>
        +rating: float
        +numRatings: int
        +getRating(): float
        +getNumRatings(): int
        +setRating(rating: float)
        +setNumRatings(numRatings: int)
    }

    class AccountController {
        <<Controller>>
        +AccountController(account: Account)
        +onScroll(): void
        +logout(): boolean
        +addPhoto(photo: Image): boolean
        +editUsername(username: String): boolean
        +editEmail(email: String): boolean
        +editPassword(password: String): boolean
        +goBack(): void
        +displayRecipes(): boolean
        +displayRecipe(recipeID: long): boolean
        +closePopUp(): void
        +saveDetail(type: String, detail: String): boolean
        +textBox(): void
    }

    class AccountGUI {
        <<Boundary>>
        -logout: button
        -back: button
        -accountLabel: label
        -userPhoto: image
        -addPhoto: button
        -username: label
        -password: label
        -email: label
        -editUsername: button
        -editPassword: button
        -editEmail: button
        -recipeLabel: label
        -recipeFrames: list<Frame>
        -recipePhotos: list<photo>
        -recipeTitle: list<label>
        -recipeRating: list<label>
        -recipeHeart: list<photo>
        +AccountGUI()
        +onScroll()
        +onEditUsername()
        +onEditPassword()
        +onEditEmail()
        +onAddPhoto()
        +onRecipeFrame()
        +onBack()
        +onLogout()
    }

    class EditDetailsGUI {
        <<Boundary>>
        -detailLabel: label
        -detail: textbox
        -close: button
        -save: button
        +EditDetailsGUI(type: String)
        +onBack()
        +onClose()
        +onSave()
        +onTextBox()
    }

    class AccountProxy {
        <<Boundary>>
        +addUser(account: Account): boolean
        +editUser(account: Account): boolean
        +loadUser(account: Account): boolean
    }

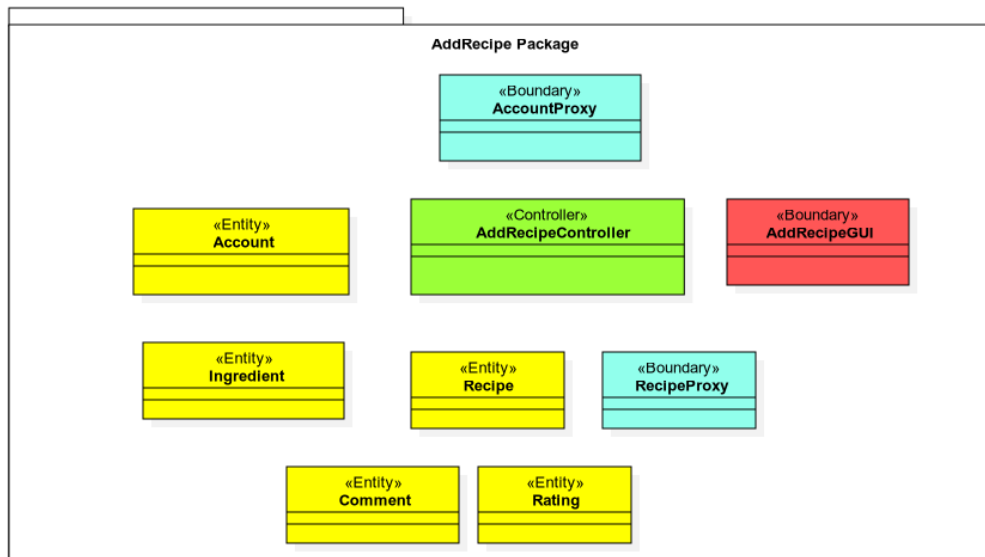
    class RecipeProxy {
        <<Boundary>>
        +addRecipe(recipe: Recipe): boolean
        +fetchRecipes(recipesID: List<long>): List<Recipe>
        +addRating(recipe: Recipe): boolean
        +addComment(recipe: Recipe): boolean
    }

    class RecipeInteractionController {
        <<Controller>>
    }

    Account "1" --> "1" AccountController
    AccountController "1" --> "1" AccountProxy
    AccountController "1" --> "1" RecipeProxy
    AccountController "1" --> "1" RecipeInteractionController
    AccountController "1" --> "1" AccountGUI
    AccountController "1" --> "1" EditDetailsGUI
    Account "1" --> "1" Ingredient
    Ingredient "1" --> "1" Recipe
    Recipe "1" --> "1" Comment
    Recipe "1" --> "1" Rating
    AccountGUI <|-- EditDetailsGUI
  
```



## 4.1.3 AddRecipe Package



### Entity Recipe

Η οντότητα αυτή περιγράφει μία συνταγή.

Ιδιότητες της κλάσης Recipe είναι:

- title: Το όνομα της συνταγής
- photo: Μία φωτογραφία της συνταγής
- recipeID: Το ID που έχει η συνταγή στο database.
- creator: Το username του χρήστη που κοινοποίησε την συνταγή.
- filter: Η κατηγορία στην οποία ανήκει η συνταγή.
- description: Η περιγραφή της εκτέλεσης της συνταγής.
- isScalable: Μία boolean μεταβλητή που δηλώνει αν η συνταγή είναι εκτελέσιμη για λιγότερες ή περισσότερες μερίδες.
- portions: Ο αριθμός των μερίδων που παράγονται με τα συγκεκριμένα υλικά.
- comments: Μία λίστα από αντικείμενα τύπου Comment με όλα τα σχόλια των χρηστών για την συγκεκριμένη συνταγή.
- rating: Ένα αντικείμενο τύπου Rating με την βαθμολογία της συνταγής και τον αριθμό των χρηστών που την βαθμολόγησαν.



Μέθοδοι της κλάσης Recipe:

Ως κλάση entity, οι μέθοδοι της κλάσης απαρτίζονται από έναν constructor, καθώς και getters και setters για τις ιδιότητες της.

#### **Entity Comment**

Η οντότητα αυτή περιγράφει ένα σχόλιο.

Ιδιότητες της κλάσης Comment είναι:

- text: Το περιεχόμενο του σχολίου.
- username: Το username του χρήστη που έκανε το σχόλιο.
- photo: Η φωτογραφία του χρήστη που έκανε το σχόλιο.

Μέθοδοι της κλάσης Comment:

Ως κλάση entity, οι μέθοδοι της κλάσης απαρτίζονται από έναν constructor, καθώς και getters και setters για τις ιδιότητες της.

#### **Entity Rating**

Η οντότητα αυτή περιγράφει την αξιολόγηση της συνταγής.

Ιδιότητες της κλάσης Rating είναι:

- rating: Η βαθμολογία της συνταγής.
- numRatings: Ο αριθμός των χρηστών που αξιολόγησαν την συνταγή.

Μέθοδοι της κλάσης Rating:

Ως κλάση entity, οι μέθοδοι της κλάσης απαρτίζονται από έναν constructor, καθώς και getters και setters για τις ιδιότητες της.

#### **Controller AddRecipeController**

Η κλάση αυτή περιέχει τις μεθόδους που επιτρέπουν στον χρήστη να προσθέσει μία νέα συνταγή.

Ιδιότητες της κλάσης AddRecipeController είναι:

- Account: Αντικείμενο τύπου Account.
- Recipe: Αντικείμενο τύπου Recipe.

Μέθοδοι της κλάσης AddRecipeController είναι:

- addRecipeController(account: Account): ο constructor, που δέχεται σαν όρισμα τον λογαριασμό του χρήστη



- goBack(): Μετάβαση του χρήστη στην προηγούμενη σελίδα.
- addPhoto(photo: image): Προσθήκη της φωτογραφίας της συνταγής.
- addTitle(title: String): Προσθήκη ονόματος στην συνταγή.
- addIngredient(ingredient: Ingredient): Προσθήκη ενός υλικού της συνταγής.
- removeIngredient(ingredient: Ingredient): Αφαίρεση ενός υλικού από τα υλικά της συνταγής.
- searchIngredient(keyword: String): Αναζήτηση ενός υλικού.
- addDescription(description: String): Προσθήκη κειμένου για την περιγραφή της συνταγής.
- toggleScalable(): Επιλογή για το αν η συνταγή μπορεί να πραγματοποιηθεί για μεταβλητό αριθμό ατόμων.
- increasePortions(): Η συνάρτηση αυτή αυξάνει τον αριθμό μερίδων της συνταγής κατά 1.
- decreasePortions(): Η συνάρτηση αυτή μειώνει τον αριθμό μερίδων της συνταγής κατά 1.
- showCategory(): Η συνάρτηση αυτή δείχνει στον χρήστη τις κατηγορίες από τις οποίες μπορεί να επιλέξει για να κατατάξει τη συνταγή του.
- selectCategory(): Η συνάρτηση δίνει στο χρήστη τη δυνατότητα επιλογής της κατηγορίας που ανήκει η συνταγή του.
- share(): Η συνάρτηση αυτή καταχωρεί τη νέα συνταγή στο database. Επιστρέφει boolean για το αν εκτελέστηκε επιτυχώς η συνάρτηση.
- displayAccount(): η συνάρτηση αυτή μεταβιβάζει τον χρήστη στο λογαριασμό του δημιουργώντας τον AccountController

### **Boundary AddRecipeGUI**

Η κλάση αυτή αντιπροσωπεύει το γραφικό περιβάλλον της σελίδας όπου ο χρήστης μπορεί να προσθέσει μία νέα συνταγή.

Ιδιότητες της κλάσης AddRecipeGUI:

- account: Το κουμπί στο πάνω αριστερό μέρος της σελίδας που θα μεταβιβάζει τον χρήστη στον λογαριασμό του
- back: Κουμπί που οδηγεί τον χρήστη στην σελίδα του λογαριασμού του.
- addPhoto: Το κουμπί με το οποίο ο χρήστης θα προσθέτει μία φωτογραφία στη συνταγή που πρόκειται να ανεβάσει.
- recipePhoto: Το δεύτερο κουμπί με το οποίο ο χρήστης θα προσθέτει μία φωτογραφία στη συνταγή που πρόκειται να ανεβάσει.
- addTitle: Το textbox μέσα στο οποίο ο χρήστης μπορεί να προσθέσει το τίτλο της συνταγής που πρόκειται να ανεβάσει.
- addIngredient: Το textbox μέσω του οποίου ο χρήστης μπορεί να αναζητήσει και να προσθέσει ένα υλικό στη συνταγή που πρόκειται να ανεβάσει.





- `removeIngredient`: Λίστα από κουμπιά μέσω των οποίων ο χρήστης μπορεί να αφαιρέσει υλικά από τη συνταγή που πρόκειται να ανεβάσει.
- `ingredientsList`: Λίστα από labels με τα ονόματα των υλικών.
- `addDescription`: Το textbox στο οποίο ο χρήστης μπορεί να προσθέσει την περιγραφή της συνταγής που πρόκειται να ανεβάσει.
- `scalable`: label
- `scalableBox`: Check box μέσω του οποίου ο χρήστης μπορεί να επιλέξει αν η συνταγή μπορεί να πραγματοποιηθεί για μεταβλητό αριθμό ατόμων.
- `numberOfPortions`: label
- `upArrow`: Το κουμπί μέσω του οποίου ο χρήστης μπορεί να αυξήσει τα portions κατά 1.
- `downArrow`: Το κουμπί μέσω του οποίου ο χρήστης μπορεί να μειώσει τα portions κατά 1.
- `category`: label
- `categoryFrame`: Το κουμπί με το οποίο ο χρήστης ανοίγει τη λίστα με τις κατηγορίες φαγητών.
- `share`: Το κουμπί μέσω του οποίου ο χρήστης μπορεί να καταχωρήσει τη συνταγή του στο database.

#### Μέθοδοι της κλάσης `AddRecipeGUI`:

- `AddRecipeGUI()`: ο constructor του αντικειμένου
- `onAccount()`: Καλεί την συνάρτηση `displayAccount(account: Account)` από τον ελεγκτή `AddRecipeController` για την μεταβίβαση του χρήστη στον λογαριασμό του
- `onBack()`: Καλεί την συνάρτηση `goBack()` από τον ελεγκτή `AddRecipeController` για την μεταβίβαση του χρήστη στην προηγούμενη σελίδα.
- `onAddPhoto()`: Καλεί την συνάρτηση `addPhoto(photo: image)` από τον ελεγκτή `AddRecipeController`.
- `onAddTitle()`: Καλεί την συνάρτηση `addTitle(title: String)` από τον ελεγκτή `AddRecipeController`.
- `onAddIngredient()`: Καλεί την συνάρτηση `addIngredient(ingredient: Ingredient)` από τον ελεγκτή `AddRecipeController`.
- `onRemoveIngredient()`: Καλεί την συνάρτηση `removeIngredient(ingredient: Ingredient)` από τον ελεγκτή `AddRecipeController`.



12/04/2021

- 
- ```

classDiagram
    class Account {
        -username: String
        -password: String
        -email: String
        -accountID: long
        -photo: Image
        -myRecipesID: List<long>
        -favoritesID: List<long>
        +Account()
        +getUsername(): String
        +getPassword(): String
        +getEmail(): String
        +getAccountID(): long
        +getPhoto(): Image
        +getMyRecipesID(): List<long>
        +getFavoritesID(): List<long>
        +setUsername(username: String)
        +setPassword(password: String)
        +setEmail(email: String)
        +setAccountID(accountID: long)
        +setPhoto(photo: Image)
        +setMyRecipesID(myRecipesID: List<long>)
        +setFavoritesID(favoritesID: List<long>)
    }

    class Ingredient {
        -name: String
        -isScalable: boolean
        -quantity: int
        +Ingredient()
        +getName(): String
        +getIsScalable(): boolean
        +getQuantity(): int
        +setName(name: String)
        +setIsScalable(isScalable: boolean)
        +setQuantity(quantity: int)
    }

    class Recipe {
        -title: String
        -photo: Image
        -recipeID: long
        -creator: String
        -filter: String
        -description: String
        -isScalable: boolean
        -portions: int
        +Recipe()
        +getTitle(): String
        +getPhoto(): Image
        +getRecipeID(): long
        +getCreator(): String
        +getFilter(): String
        +getDescription(): String
        +getIsScalable(): boolean
        +getPortions(): int
        +setTitle(title: String)
        +setPhoto(photo: Image)
        +setRecipeID(recipeID: long)
        +setCreator(creator: String)
        +setFilter(filter: String)
        +setDescription(description: String)
        +setIsScalable(isScalable: boolean)
        +setPortions(portions: int)
    }

    class Comment {
        -text: String
        -username: String
        -photo: Image
        +Comment()
        +getText(): String
        +getUsername(): String
        +getPhoto(): Image
        +setText(text: String)
        +setUsername(username: String)
    }

    class Rating {
        +rating: float
        +numRatings: int
        +Rating()
        +getRating(): float
        +getNumRatings(): int
        +setRating(rating: float)
        +setNumRatings(numRatings: int)
    }

    class AccountProxy {
        <<Boundary>>
        +addUser(account: Account): boolean
        +editUser(account: Account): boolean
        +loadUser(account: Account): boolean
    }

    class AddRecipeController {
        <<Controller>>
        +addRecipeController(account: Account)
        +goBack(): void
        +addPhoto(photo: Image): boolean
        +addTitle(title: String): void
        +addingIngredient(ingredient: Ingredient): void
        +removeIngredient(ingredient: Ingredient): void
        +searchIngredient(keyword: String): void
        +addDescription(description: String): void
        +toggleScalable(): boolean
        +increasePortions(): void
        +decreasePortions(): void
        +showCategory(): void
        +selectCategory(): void
        +share(): boolean
        +displayAccount(): void
    }

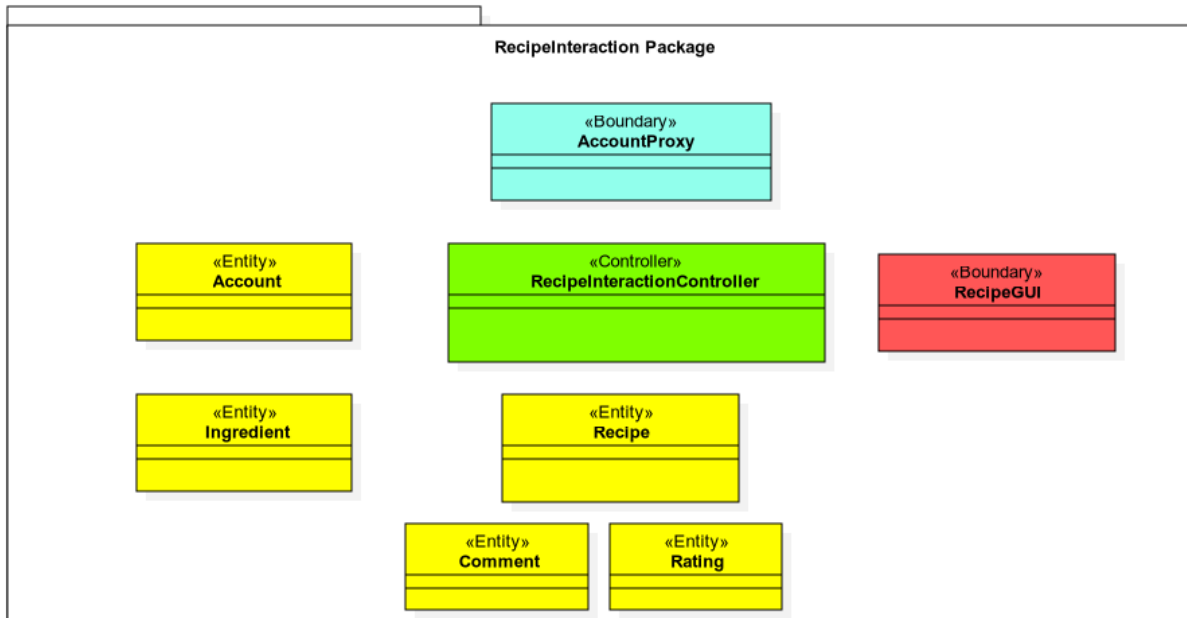
    class AddRecipeGUI {
        <<Boundary>>
        -account: button
        -back: button
        -addPhoto: button
        -recipePhoto button
        -addTitle: textbox
        -addingIngredient: textbox
        -removeIngredient: List<button>
        -ingredientsList: List<label>
        -addDescription: textbox
        -scalable: label
        -scalableBox: checkBox
        -numberPortions: label
        -upArrow: button
        -downArrow: button
        -category: label
        -categoryFrame: button
        -share: button
        +AddRecipeGUI()
        +onBack()
        +onAccount()
        +onAddPhoto()
        +onAddTitle()
        +onAddingIngredient()
        +onRemoveIngredient()
        +onAddDescription()
        +onCheckBox()
        +onUpArrowPortions()
        +onDownArrowPortions()
        +onCategory()
        +onShare()
    }

    class RecipeProxy {
        <<Boundary>>
        +addRecipe(recipe: Recipe): boolean
        +fetchRecipes(recipelds: List<long>): List<Recipe>
        +addRating(recipe: Recipe): boolean
        +addComment(recipe: Recipe): boolean
    }

    Account "1" -- "1" AccountProxy
    Account "1" -- "1" AddRecipeController
    AddRecipeController "1" -- "1" AddRecipeGUI
    AddRecipeController "1" -- "1" RecipeProxy
    AddRecipeController "1" -- "1" Ingredient
    AddRecipeController "1" -- "1" Recipe
    AddRecipeController "1" -- "1" Comment
    AddRecipeController "1" -- "1" Rating
    
```
- The diagram illustrates the architecture of a recipe application, organized into three main layers: Entities, Boundaries, and Controllers.
- Entities (Yellow):**
    - Account:** Manages user information (username, password, email, accountID, photo, myRecipesID, favoritesID) and provides methods for account management.
    - Ingredient:** Represents recipe components with attributes like name, scalability, and quantity.
    - Recipe:** Represents a recipe with attributes like title, photo, recipeID, creator, filter, description, scalability, and portions.
    - Comment:** Represents user feedback with attributes like text, username, and photo.
    - Rating:** Represents user ratings with attributes like rating and numRatings.
  - Boundaries (Cyan/Red):**
    - AccountProxy:** Acts as a bridge between the Account entity and the AccountController, providing methods like addUser, editUser, and loadUser.
    - AddRecipeGUI:** The user interface for adding recipes, containing various controls like buttons, textboxes, and labels.
    - RecipeProxy:** Acts as a bridge between the Recipe entity and the AddRecipeController, providing methods like addRecipe, fetchRecipes, addRating, and addComment.
  - Controllers (Green):**
    - AddRecipeController:** The core logic for adding and managing recipes, interacting with the Account, Ingredient, Recipe, Comment, and Rating entities and their respective proxies.
- Relationships are indicated by arrows with multiplicity (e.g., 1, +1, +.., +1).



## 4.1.4 RecipeInteraction Package



### Controller RecipeInteractionController

Η κλάση αυτή περιέχει τις μεθόδους που επιτρέπουν στον χρήστη να αλληλεπιδράσει με μία συνταγή.

Ιδιότητες της κλάσης AddRecipeController είναι:

- Account: Αντικείμενο τύπου Account.
- Recipe: Αντικείμενο τύπου Recipe.

Μέθοδοι της κλάσης AddRecipeController είναι:

- RecipeInteractionController(account: Account, recipeID: long): ο constructor, που δέχεται σαν όρισμα τον λογαριασμό του χρήστη και το id της συνταγής.
- RateIt(rating: int): Η συνάρτηση μέσω της οποίας ο χρήστης θα αξιολογεί την συνταγή. Επιστρέφει boolean για το αν εκτελέστηκε επιτυχώς η συνάρτηση.
- comment(comment: Comment): Η συνάρτηση μέσω της οποίας ο χρήστης θα προσθέτει ένα σχόλιο στη λίστα με αντικείμενα τύπου Comment της συνταγής.



- `ShareComment()`: Η συνάρτηση μέσω της οποίας το σύστημα θα αποθηκεύει στη βάση δεδομένων την λίστα από τα σχόλια της συνταγής. Επιστρέφει boolean για το αν εκτελέστηκε επιτυχώς η συνάρτηση.
- `CookedIt()`: Η συνάρτηση μέσω της οποίας το σύστημα θα αφαιρεί τα υλικά της συνταγής από τα διαθέσιμα υλικά του χρήστη. Επιστρέφει boolean για το αν εκτελέστηκε επιτυχώς η συνάρτηση.
- `ToggleFavorite()`: Η συνάρτηση μέσω της οποίας ο χρήστης θα προσθέτει και θα αφαιρεί τη συνταγή από τη λίστα με τα αγαπημένα του. Επιστρέφει boolean για το αν εκτελέστηκε επιτυχώς η συνάρτηση.
- `ViewMoreDesc()`: Η συνάρτηση η οποία θα εμφανίζει την υπόλοιπη περιγραφή της συνταγής.
- `ViewMoreComment()`: Η συνάρτηση η οποία θα εμφανίζει τα υπόλοιπα σχόλια της συνταγής.
- `ScrollPage()`: Η αναγνώριση της περιήγησης του χρήστη σε όλη τη σελίδα
- `addPortions()`: Η συνάρτηση με την οποία ο χρήστης θα μπορεί να αυξήσει τις μερίδες της συνταγής στην περίπτωση που η συνταγή το επιτρέπει.
- `removePortions()`: Η συνάρτηση με την οποία ο χρήστης θα μπορεί να μειώσει τις μερίδες της συνταγής στην περίπτωση που η συνταγή το επιτρέπει.
- `displayAccount(account:Account)`: η συνάρτηση αυτή μεταβιβάζει τον χρήστη στο λογαριασμό του δημιουργώντας τον `AccountController`
- `goBack()`: Μετάβαση του χρήστη στην προηγούμενη σελίδα.

### Boundary RecipeGUI

Η κλάση αυτή αντιπροσωπεύει το γραφικό περιβάλλον της σελίδας όπου ο χρήστης μπορεί να δει τις λεπτομέρειες μίας συνταγής.

Ιδιότητες της κλάσης `AddRecipeGUI`:

- `account`: Το κουμπί στο πάνω αριστερό μέρος της σελίδας που θα μεταβιβάζει τον χρήστη στον λογαριασμό του
- `back`: Κουμπί που οδηγεί τον χρήστη στην σελίδα του λογαριασμού του.
- `RecipeName`: label
- `recipePhoto`: image
- `creator`: label
- `ratings`: label
- `recipeHeart`: Το check box μέσω του οποίου ο χρήστης μπορεί να προσθέσει ή να αφαιρέσει μία συνταγή από τα αγαπημένα του.



- RecipeFrames: Τα frames μέσα στα οποία παρουσιάζονται οι λεπτομέρειες κάθε συνταγής.
- ingredientsLabel:label
- ingredients: Λίστα από labels
- description: label
- descriptionDetails:label
- descriptionViewMore: Κουμπί μέσω του οποίου ο χρήστης θα μπορεί να προβάλλει την υπόλοιπη περιγραφή της συνταγής.
- CommentsLabel:label
- enterText: Το textbox μέσα στο οποίο ο χρήστης θα μπορεί να πληκτρολογεί το σχόλιό του.
- CommentB:Το κουμπί μέσω του οποίου ο χρήστης θα δημοσιεύει το σχόλιό του.
- commentsFrames:Τα frames μέσα στα οποία περιέχονται τα σχόλια των χρηστών.
- Comments: Λίστα από labels με τα σχόλια των χρηστών
- commentsViewMore: Κουμπί μέσω του οποίου ο χρήστης θα μπορεί να προβάλλει τα υπόλοιπα σχόλια της συνταγής.
- StarsList: Λίστα από κουμπιά μέσω των οποίων ο χρήστης θα επιλέγει τη βαθμολογία της συνταγής.
- RateIt: Κουμπί μέσω του οποίου ο χρήστης θα δημοσιεύει την βαθμολογία που μόλις καταχώρησε.
- PortionsLabel:label
- portionsNumber: Label με τον αριθμό των μερίδων
- cookedIt: Κουμπί μέσω του οποίου ο χρήστης θα επιλέγει ότι μαγείρεψε τη συνταγή.
- leftArrow: Κουμπί μέσω του οποίου ο χρήστης θα μειώνει τις μερίδες της συνταγής εφόσον είναι επιτρεπτό.
- rightArrow: Κουμπί μέσω του οποίου ο χρήστης θα αυξάνει τις μερίδες της συνταγής εφόσον είναι επιτρεπτό.

#### Μέθοδοι της κλάσης RecipeGUI:

- RecipeGUI(): ο constructor του αντικειμένου
- onAccount(): Καλεί την συνάρτηση displayAccount(account: Account) από τον ελεγκτή RecipeInteractionController για την μεταβίβαση του χρήστη στον λογαριασμό του



# Hungry

Making cooking simple

- `onBack()`: Καλεί την συνάρτηση `onBack()` από τον ελεγκτή `RecipeInteractionController` για την μεταβίβαση του χρήστη στην προηγούμενη σελίδα.
- `OnViewMoreComments()`: Καλεί την συνάρτηση `viewMoreComments()` από τον ελεγκτή `RecipeInteractionController`
- `OnViewMoreDescription()`: Καλεί την συνάρτηση `viewMoreDesc()` από τον ελεγκτή `RecipeInteractionController`
- `OnHeart()`: Καλεί την συνάρτηση `toggleFavorite()` από τον ελεγκτή `RecipeInteractionController`
- `OnEnterText()`: Καλεί την συνάρτηση `comment(comment: Comment)` από τον ελεγκτή `RecipeInteractionController`
- `OnComment()`: Καλεί την συνάρτηση `shareComment()` από τον ελεγκτή `RecipeInteractionController`



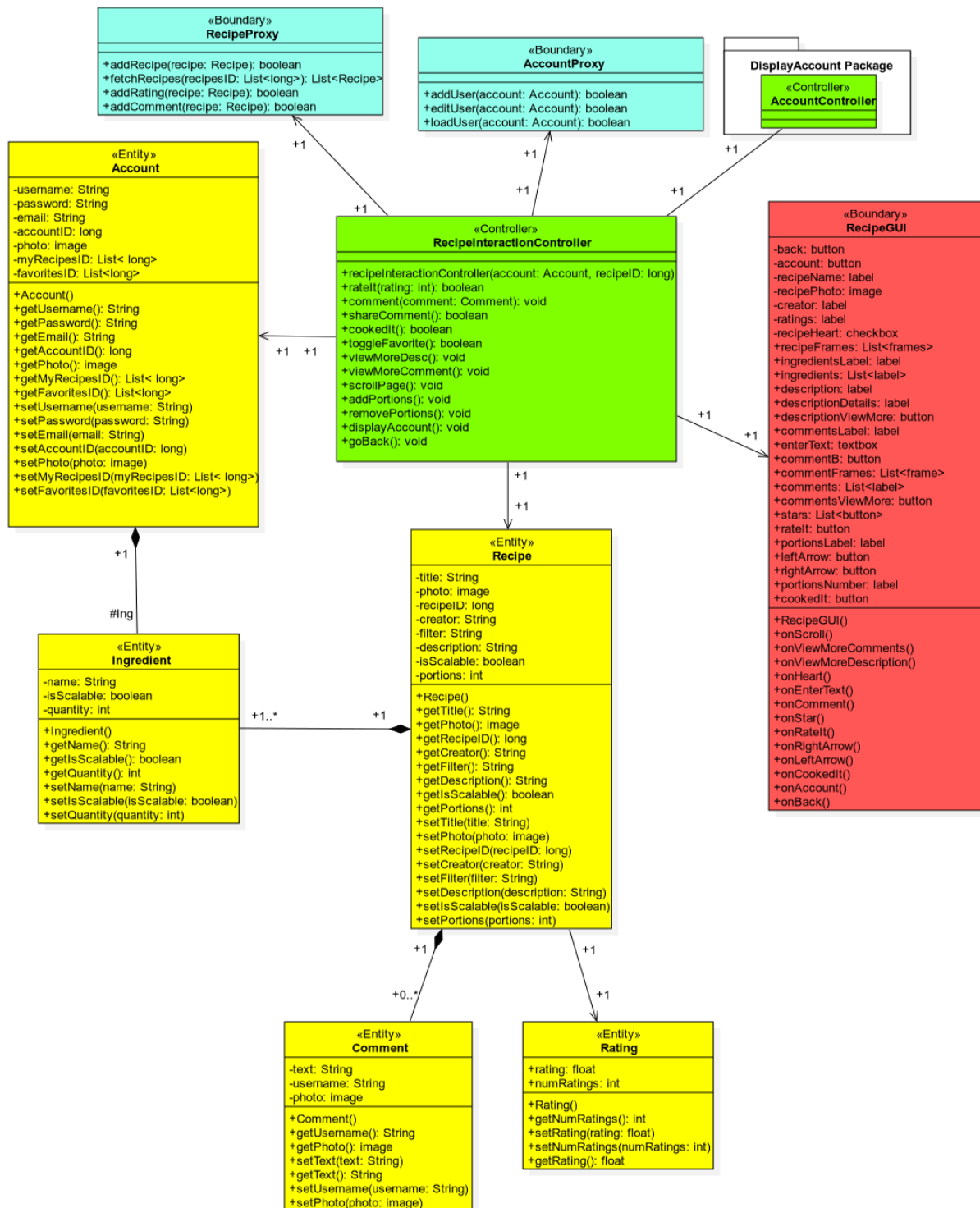
# Hungry

Making cooking simple

Έγγραφο Απαιτήσεων Χρηστών

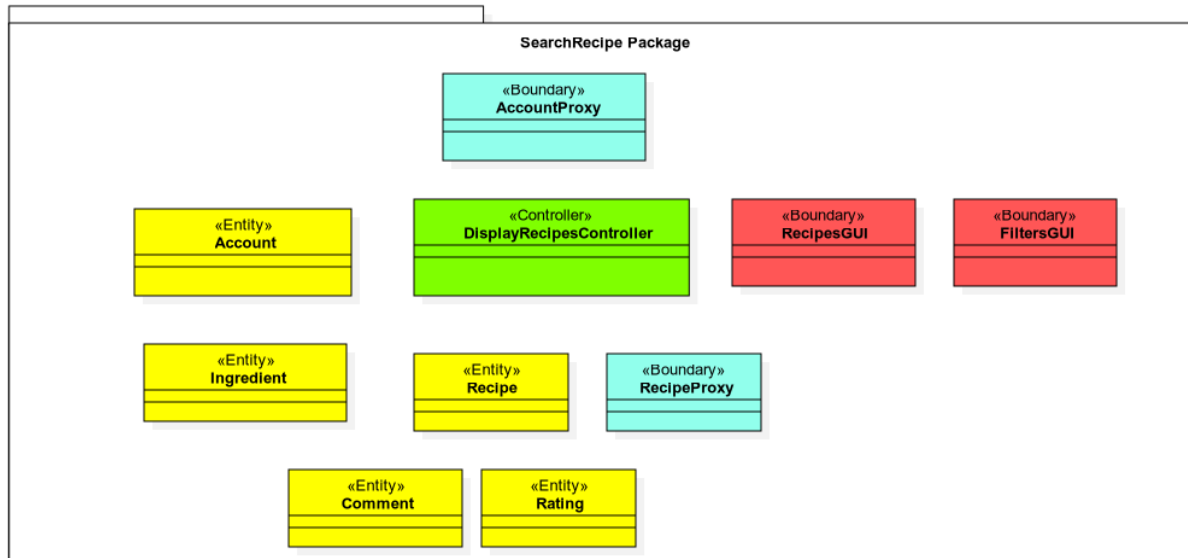
εκδ. 0.1

12/04/2021





## 4.1.5 SearchRecipes Package



### Controller DisplayRecipesController

Η κλάση αυτή περιέχει τις μεθόδους που επιτρέπουν στον χρήστη να αναζητήσει συνταγές.

Ιδιότητες της κλάσης AddRecipeController είναι:

- Account: Αντικείμενο τύπου Account.
- Recipe: Μία λίστα αντικειμένων τύπου Recipe.

Μέθοδοι της κλάσης DisplayRecipesController είναι:

- displayRecipesController(account:Account) : Ο constructor του αντικειμένου displayRecipesController
- displayRecipes(recipesIDs: List<long>) : Η συνάρτηση η οποία εμφανίζει τα αποτελέσματα αναζήτησης συνταγών.
- toggleFavorite() : Η συνάρτηση η οποία προσθέτει ή αφαιρεί μία συνταγή από τη λίστα των αγαπημένων
- displayFilters() : Η συνάρτηση η οποία εμφανίζει λίστα με τα φίλτρα αναζήτησης δημιουργώντας ένα αντικείμενο filtersGUI.
- filterResults(filters List<String>): Η συνάρτηση η οποία καλείται όταν έχει επιλεγθεί τουλάχιστον ένα φίλτρο από το filtersGUI. Επιστρέφει μία λίστα από τα IDs των συνταγών που αντιστοιχούν στα φίλτρα αναζήτησης.
- DisplayAccount() : Η συνάρτηση αυτή μεταβιβάζει το χρήστη στον λογαριασμό του





- `goBack()`: Η συνάρτηση η οποία μεταβιβάζει τον χρήστη στην προηγούμενη σελίδα
- `viewMoreFilters()` Η συνάρτηση αυτή εμφανίζει τα υπόλοιπα φίλτρα
- `filterCheckBox(row:int)`: Η συνάρτηση αυτή δέχεται ως όρισμα την γραμμή που βρίσκεται το φίλτρο αναζήτησης και επιλέγει/αποεπιλέγει το αντίστοιχο φίλτρο.
- `scrollPage()`: Η αναγνώριση της περιήγησης του χρήστη σε όλη τη σελίδα.

## Boundary RecipesGUI

Η κλάση αυτή περιγράφει το γραφικό περιβάλλον της αναζήτησης συνταγών

Οι ιδιότητες της κλάσης αυτής είναι:

- `back` : Κουμπί που επιστρέφει τον χρήστη στην προηγούμενη σελίδα
- `account`: Κουμπί που μεταβιβάζει το χρήστη στο λογαριασμό του.
- `Filters`: Κουμπί που εμφανίζει στον χρήστη τα φίλτρα αναζήτησης
- `recipeFrames`: Λίστα από frames μέσα στα οποία εμφανίζονται οι συνταγές
- `recipePhotos`: Λίστα από φωτογραφίες των συνταγών
- `recipeTitles`: Λίστα από labels με τους τίτλους των συνταγών
- `recipeRatings`: Λίστα από labels με τις βαθμολογίες των συνταγών
- `recipeHearts`: Λίστα από checkboxes με τα οποία ο χρήστης προσθαφαιρεί μία συνταγή από τη λίστα με τα αγαπημένα του.

Οι μέθοδοι της κλάσης είναι:

- `recipesGUI()`: Ο constructor της κλάσης
- `onScroll()` : Καλεί την συνάρτηση `scrollPage()` του ελεγκτή `DisplayRecipesController`
- `onBack()`: Καλεί την συνάρτηση `goBack()` του ελεγκτή `DisplayRecipesController`
- `onFilters()`: Καλεί την συνάρτηση `displayFilters()` του ελεγκτή `DisplayRecipesController`
- `onFrame()`: Καλεί την συνάρτηση `displayRecipe(recipeID:long)` του ελεγκτή `DisplayRecipesController`
- `onHeart()`: Καλεί την συνάρτηση `toggleFavorite()` του ελεγκτή `DisplayRecipesController`
- `onAccount()`: Καλεί την συνάρτηση `displayAccount()` του ελεγκτή `DisplayRecipesController`



## Boundary FiltersGUI

Η κλάση αυτή περιγράφει το γραφικό περιβάλλον των φίλτρων αναζήτησης.

Οι ιδιότητες της κλάσης αυτής είναι:

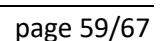
- back : Κουμπί που επιστρέφει τον χρήστη στην προηγούμενη σελίδα
- account: Κουμπί που μεταβιβάζει το χρήστη στο λογαριασμό του.
- filtersLabel: label
- filters:Λίστα από label με τις ονομασίες των φίλτρων
- viewMoreButton: Το κουμπί με το οποίο ο χρήστης μπορεί να δει τα υπόλοιπα φίλτρα
- filtersCheckBoxList: Λίστα από checkboxes με τα οποία ο χρήστης προσθαφαιρεί τα φίλτρα αναζήτησης.

Οι μέθοδοι της κλάσης είναι:

- filtersGUI(): Ο constructor της κλάσης
- onScroll() : Καλεί την συνάρτηση scrollPage() του ελεγκτή DisplayRecipesController
- onBack(): Καλεί την συνάρτηση goBack() του ελεγκτή DisplayRecipesController
- onViewMore(): Καλεί την συνάρτηση viewMoreFilters() του ελεγκτή DisplayRecipesController
- onFilterCheckBox(): Καλεί την συνάρτηση filterCheckBox(row:int) του ελεγκτή DisplayRecipesController



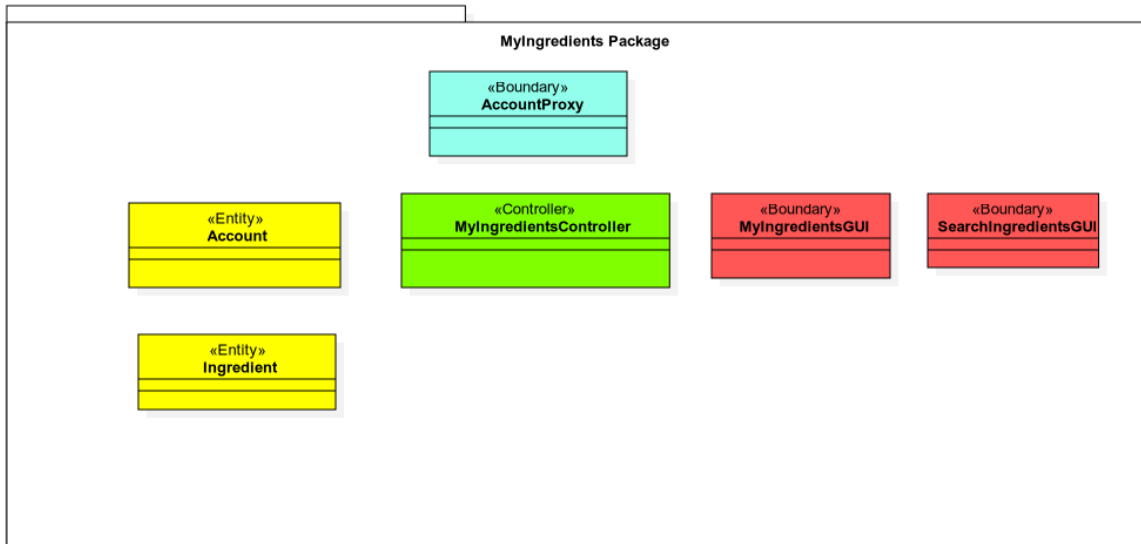
12/04/2021





## 4.2 Medium Priority

### 4.2.1 MyIngredients Package



#### Controller MyIngredientsController

Η μόνη ιδιότητα της κλάσης MyIngredientsController είναι ένα αντικείμενο Account

Μέθοδοι της κλάσης είναι:

- myIngredientsController(account: Account): ο constructor, που δέχεται σαν όρισμα τον λογαριασμό του χρήστη
- scrollPage(): Η αναγνώριση της περιήγησης του χρήστη σε όλη τη σελίδα.
- displayIngredients(): Η συνάρτηση η οποία εμφανίζει την λίστα των Ingredient και της ποσότητάς τους.
- goBack(): Μετάβαση του χρήστη στην προηγούμενη σελίδα
- searchIngredient(keyword:String): Η συνάρτηση μέσω της οποίας ο χρήστης μπορεί να κάνει search ένα νέο ingredient. Επιστρέφει αντικείμενο τύπου Ingredient
- increaseQuantity(ingredient:Ingredient): Η συνάρτηση αυτή ελέγχει άμα το Ingredient είναι μετρήσιμο. Άμα είναι αυξάνει την ποσότητα του κατά 1, αλλιώς κάνει την ποσότητα του 1.
- decreaseQuantity(ingredient:Ingredient): Η συνάρτηση αυτή ελέγχει άμα το Ingredient είναι μετρήσιμο. Άμα είναι μειώνει την ποσότητα του κατά 1, αλλιώς κάνει την ποσότητα του 0.
- displayAccount(): Η συνάρτηση που μεταβιβάζει τον χρήστη στο λογαριασμό του δημιουργώντας τον Account Controller.



### **Boundary MyIngredientsGUI**

Η οντότητα αυτή αναπαριστά τη διεπαφή της σελίδας προβολής της λίστας των υλικών που διαθέτει ο χρήστης.

Ιδιότητες της κλάσης FavoriteGUI:

- back: Κουμπί που οδηγεί τον χρήστη στην προηγούμενη σελίδα διεπαφής
- account: Κουμπί που οδηγεί τον χρήστη στην σελίδα του λογαριασμού του.
- Search:Textbox μέσω του οποίου ο χρήστης μπορεί να αναζητά ένα νέο υλικό
- ingredientsLabel: label
- ingredientFrames: Η λίστα με τα πλαίσια των υλικών
- ingredientTitles: Η λίστα με τους τίτλους των υλικών
- ingredientQuantity: Η λίστα με τις ποσότητες των υλικών
- ingredientCheckBox: Λίστα με checkboxes με τα οποία ο χρήστης θα επιλέγει αν διαθέτει ή όχι το κάθε υλικό.
- leftArrow: Λίστα με κουμπιά με τα οποία ο χρήστης θα μπορεί να μειώνει την ποσότητα των υλικών
- rightArrow: Λίστα με κουμπιά με τα οποία ο χρήστης θα μπορεί να αυξάνει την ποσότητα των υλικών

Μέθοδοι κλάσης MyIngredientsGUI:

- MyIngredientsGUI() : Ο constructor της κλάσης
- onScroll():Καλεί τη συνάρτηση scrollPage() του ελεγκτή MyIngredientsController.
- onBack():Καλεί τη συνάρτηση goBack() του ελεγκτή MyIngredientsController.
- displayIngredients() : Καλεί τη συνάρτηση displayIngredients() του ελεγκτή MyIngredientsController.
- onSearch(): Καλεί τη συνάρτηση searchIngredient () του ελεγκτή FavoritesController
- onAccount(): Καλεί τη συνάρτηση displayAccount του MyIngredientsController.
- onRightArrow():Καλεί τη συνάρτηση increaseQuantity (ingredient:Ingredient) του MyIngredientsController.
- onLeftArrow():Καλεί τη συνάρτηση increaseQuantity (ingredient:Ingredient) του MyIngredientsController.
- onCheckBox():Καλεί τη συνάρτηση increaseQuantity (ingredient:Ingredient) του MyIngredientsController αν το checkbox είναι αποεπιλεγμένο αλλιώς καλεί την decreaseQuantity(ingredient:Ingredient) του MyIngredientsController.



### **Boundary SearchIngredientsGUI**

Η οντότητα αυτή αναπαριστά τη διεπαφή της σελίδας προβολής της αναζήτησης ενός υλικού στη λίστα των υλικών που διαθέτει ο χρήστης.

Ιδιότητες της κλάσης FavoriteGUI:

- back: Κουμπί που οδηγεί τον χρήστη στην προηγούμενη σελίδα διεπαφής
- account: Κουμπί που οδηγεί τον χρήστη στην σελίδα του λογαριασμού του.
- ingredientFrames: Η λίστα με τα πλαίσια των υλικών
- ingredientTitles: Η λίστα με τους τίτλους των υλικών
- ingredientQuantity: Η λίστα με τις ποσότητες των υλικών
- ingredientCheckBox: Λίστα με checkboxes με τα οποία ο χρήστης θα επιλέγει αν διαθέτει ή όχι το κάθε υλικό.
- leftArrow: Λίστα με κουμπιά με τα οποία ο χρήστης θα μπορεί να μειώνει την ποσότητα των υλικών
- rightArrow: Λίστα με κουμπιά με τα οποία ο χρήστης θα μπορεί να αυξάνει την ποσότητα των υλικών

Μέθοδοι κλάσης SearchIngredientsGUI:

SearchIngredientsGUI() : Ο constructor της κλάσης

- onScroll(): Καλεί τη συνάρτηση scrollPage() του ελεγκτή MyIngredientsController.
- onBack(): Καλεί τη συνάρτηση goBack() του ελεγκτή MyIngredientsController.
- onSearch(): Καλεί τη συνάρτηση searchIngredient () του ελεγκτή FavoritesController
- onAccount(): Καλεί τη συνάρτηση displayAccount του MyIngredientsController.
- onRightArrow(): Καλεί τη συνάρτηση increaseQuantity (ingredient:Ingredient) του MyIngredientsController.
- onLeftArrow(): Καλεί τη συνάρτηση increaseQuantity (ingredient:Ingredient) του MyIngredientsController.
- onCheckBox(): Καλεί τη συνάρτηση increaseQuantity (ingredient:Ingredient) του MyIngredientsController αν το checkbox είναι αποεπιλεγμένο αλλιώς καλεί την decreaseQuantity(ingredient:Ingredient) του MyIngredientsController.



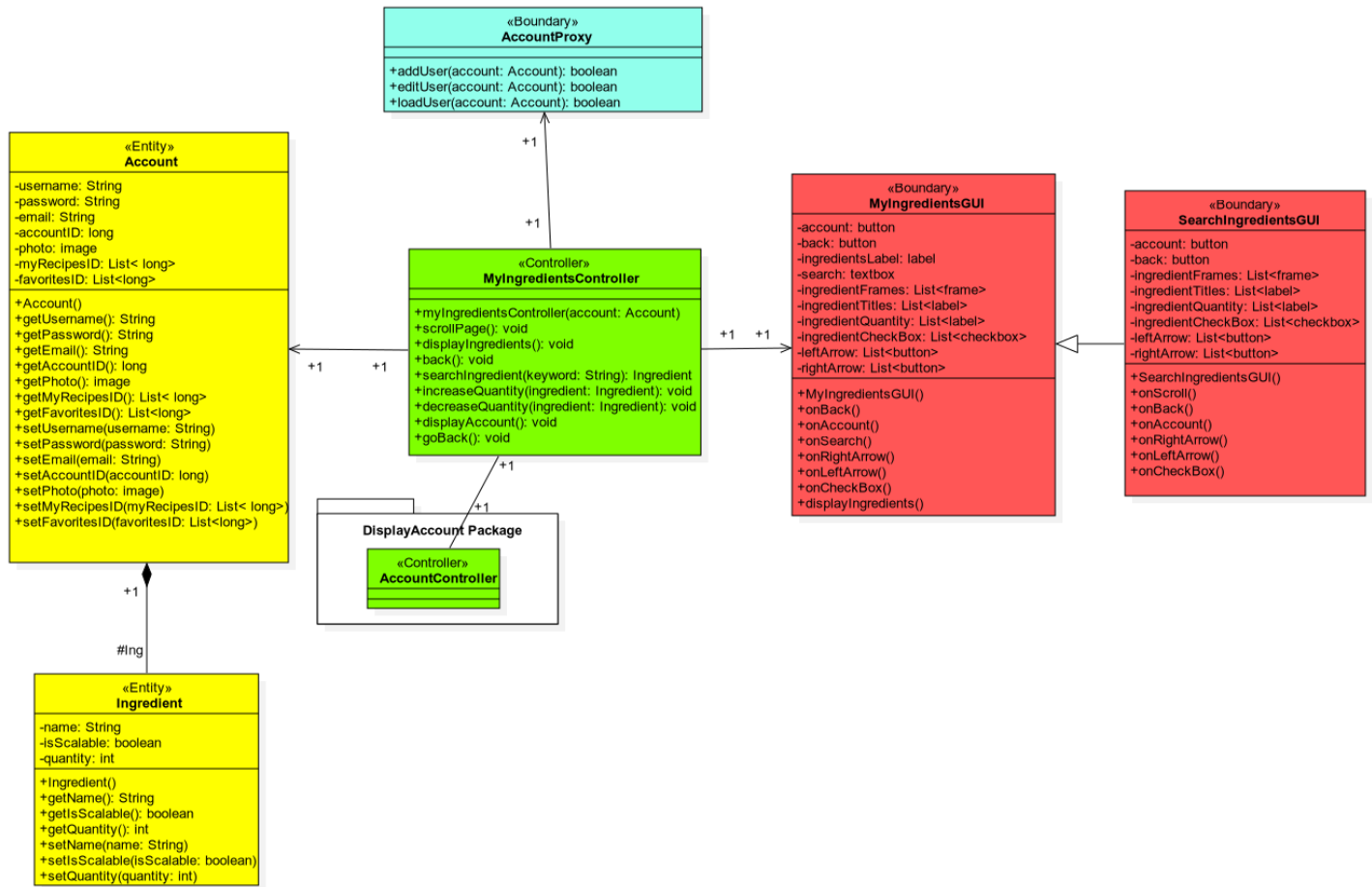
# Hungry

Making cooking simple

Έγγραφο Απαιτήσεων Χρηστών

εκδ. 0.1

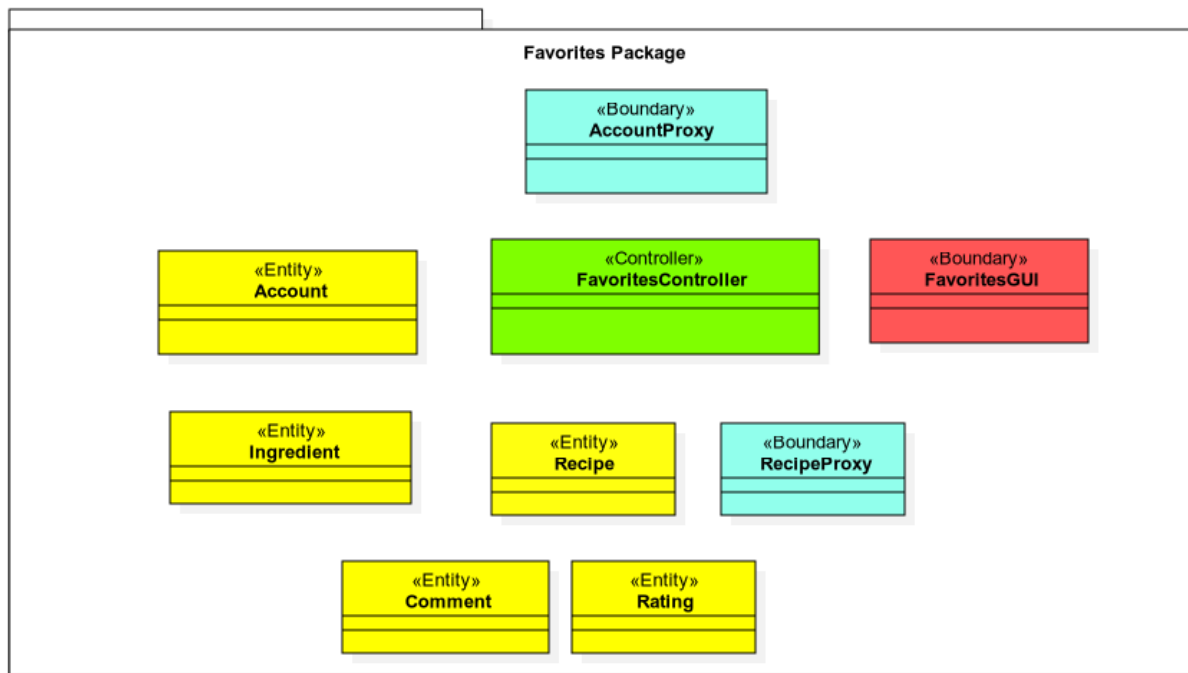
12/04/2021





## 4.3 Low Priority

### 4.3.1 Favorites Package



#### Boundary FavoritesGUI

Η οντότητα αυτή αναπαριστά τη διεπαφή της σελίδας προβολής των αγαπημένων συνταγών του χρήστη.

Ιδιότητες της κλάσης FavoriteGUI:

- back: Κουμπί που οδηγεί τον χρήστη στην προηγούμενη σελίδα διεπαφής
- account: Κουμπί που οδηγεί τον χρήστη στην σελίδα του λογαριασμού του.
- favouritesLabel: Ο τίτλος της σελίδας Favorites.
- recipeFrames: Η λίστα με τα πλαίσια των συνταγών
- recipePhotos: Η λίστα με τις φωτογραφίες των συνταγών
- recipeTitle: Η λίστα με τους τίτλους των συνταγών
- recipeRating: Η λίστα με τις αξιολογήσεις των συνταγών
- recipeHeart: Λίστα από check-boxbuttons σε σχήμα καρδιάς που δίνει την δυνατότητα στον χρήστη να προσθέσει ή να αφαιρέσει μία συνταγή από τα αγαπημένα του.

Μέθοδοι κλάσης FavoritesGUI:

- FavouritesGUI() : Ο constructor της κλάσης
- onScroll(): Η αναγνώριση της περιήγησης του χρήστη σε όλη τη σελίδα.
- onBack(): Μετάβαση του χρήστη στην προηγούμενη σελίδα.





- onFrame() : Καλεί τη συνάρτηση displayRecipe() του ελεγκτή FavoritesController
- onHeart(): Καλεί τη συνάρτηση removeFavorite() του ελεγκτή FavoritesController
- onAccount(): Καλεί τη συνάρτηση displayAccount του AccountController.

### Controller FavoritesController

Ιδιότητες της κλάσης FavoritesController:

- account: Αντικείμενο τύπου Account.
- recipes: Λίστα αντικειμένων τύπου Recipe.

Η κλάση αυτή περιέχει τις μεθόδους που

Μέθοδοι της κλάσης είναι:

- FavoritesController(account: Account): ο constructor, που δέχεται σαν όρισμα τον λογαριασμό του χρήστη.
- displayRecipes(): η συνάρτηση αυτή μεταβιβάζει τον χρήστη στην λίστα των αγαπημένων του συνταγών. Επιστρέφει boolean για το αν εκτελέστηκε επιτυχώς μία συνάρτηση.
- removeFavorite(recipeID): η συνάρτηση αυτή αφαιρεί μία συνταγή από τα αγαπημένα. Επιστρέφει boolean για το αν εκτελέστηκε επιτυχώς μία συνάρτηση.
- displayRecipe(recipeID: long): η συνάρτηση αυτή μεταβιβάζει τον χρήστη στην σελίδα της συνταγής που επέλεξε. Επιστρέφει boolean για το αν εκτελέστηκε επιτυχώς μία συνάρτηση.

### Boundary RecipeProxy

Η κλάση αυτή χρησιμοποιείται για την επικοινωνία της εφαρμογής με την βάση δεδομένων ώστε προβάλλονται να προστίθενται και να επεξεργάζονται οι συνταγές.

Ως κλάση μεσολαβητή η κλάση RecipeProxy δεν έχει ιδιότητες.

Μέθοδοι της κλάσης RecipeProxy:

- addRecipe(recipe: Recipe) : η συνάρτηση αυτή καταχωρεί μία νέα συνταγή στην βάση δεδομένων. Επιστρέφει boolean για το αν εκτελέστηκε επιτυχώς μία συνάρτηση.
- fetchRecipes(recipeIDs: List<long>): η συνάρτηση αυτή φορτώνει και εμφανίζει μία λίστα συνταγών από την βάση δεδομένων. Επιστρέφει μία λίστα με αντικείμενα της κλάσης Recipe.
- addComment(recipe: Recipe): η συνάρτηση αυτή, η οποία δέχεται ως όρισμα ένα αντικείμενο της κλάσης Recipe, προσθέτει ένα νέο comment στην συνταγή.
- addRating(recipe: Recipe): η συνάρτηση αυτή, η οποία δέχεται ως όρισμα ένα αντικείμενο της κλάσης Recipe, προσθέτει ένα νέο rating στην συνταγή.



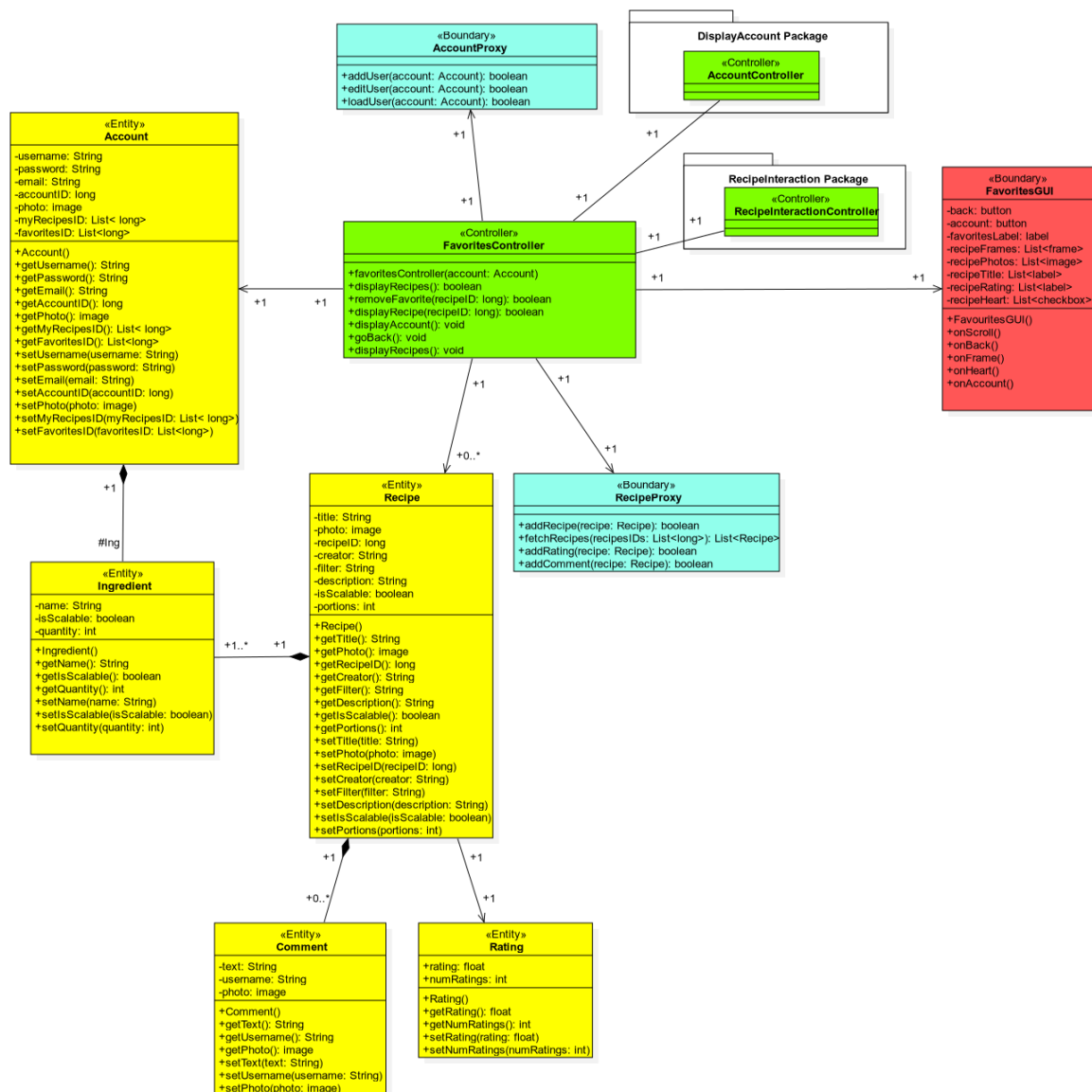
# Hungry

Making cooking simple

Έγγραφο Απαιτήσεων Χρηστών

εκδ. 0.1

12/04/2021





Hungry

Making cooking simple

## Παράρτημα Ι – Γλωσσάριο

Το σετ των ακρωνυμίων που χρησιμοποιείτε στο έγγραφο

|     |                         |
|-----|-------------------------|
| FR  | Λειτουργική Απαίτηση    |
| NFR | Μη Λειτουργική Απαίτηση |
| OA  | Ομάδα Εργασίας          |