Domaine Sciences et Technologies

Licence informatique

2021 - 2022

# Rapport de Projet GL

# (FoodApp)

**Préparé par : DABOUZ Mohammed**

**Licence 2 – Gr 2**

FoodApp c'est un projet d'une application desktop de recettes de cuisine qui permet à l'utilisateur de chercher par ingrédients et sauvegarder ses recettes préférées dans une rebrique **favoris** où il peut les consulter plus tard même s'il ne procède pas une connexion internet et tout ça dans une interface graphique pratique, fiable et facile à utiliser.

## **Partie 1 :** l'état de notre projet

### Lots livrés :

Lot 1 :

Dans ce premier lot, on a travaillé sur la lecture des fichier JSON à partir des requêtes envoyées à l'API, en créant la classe **JSONFileReader**. On a travaillé sur la liste des favoris, l'ajout et la suppression des recettes de la liste des favoris, pour cela, on a créé une classe **Receipt**, pour l'utiliser dans les méthodes de la classe **Favourites**.

On a aussi travaillé sur la structure du **CLI**, notamment l'affichage des recettes de la liste des favoris, et la recherche par ingrédients qu'on va implémenter par la suite.

Lot 2 :

Dans ce deuxième lot, on a implémenté les méthodes de la classe **Receipt**. Chaque méthode récupère des données concernant une recette en envoyant des requêtes à l'API, ou en allant chercher ces informations dans la liste des favoris dans le cas où il n'y a pas de connexion internet.

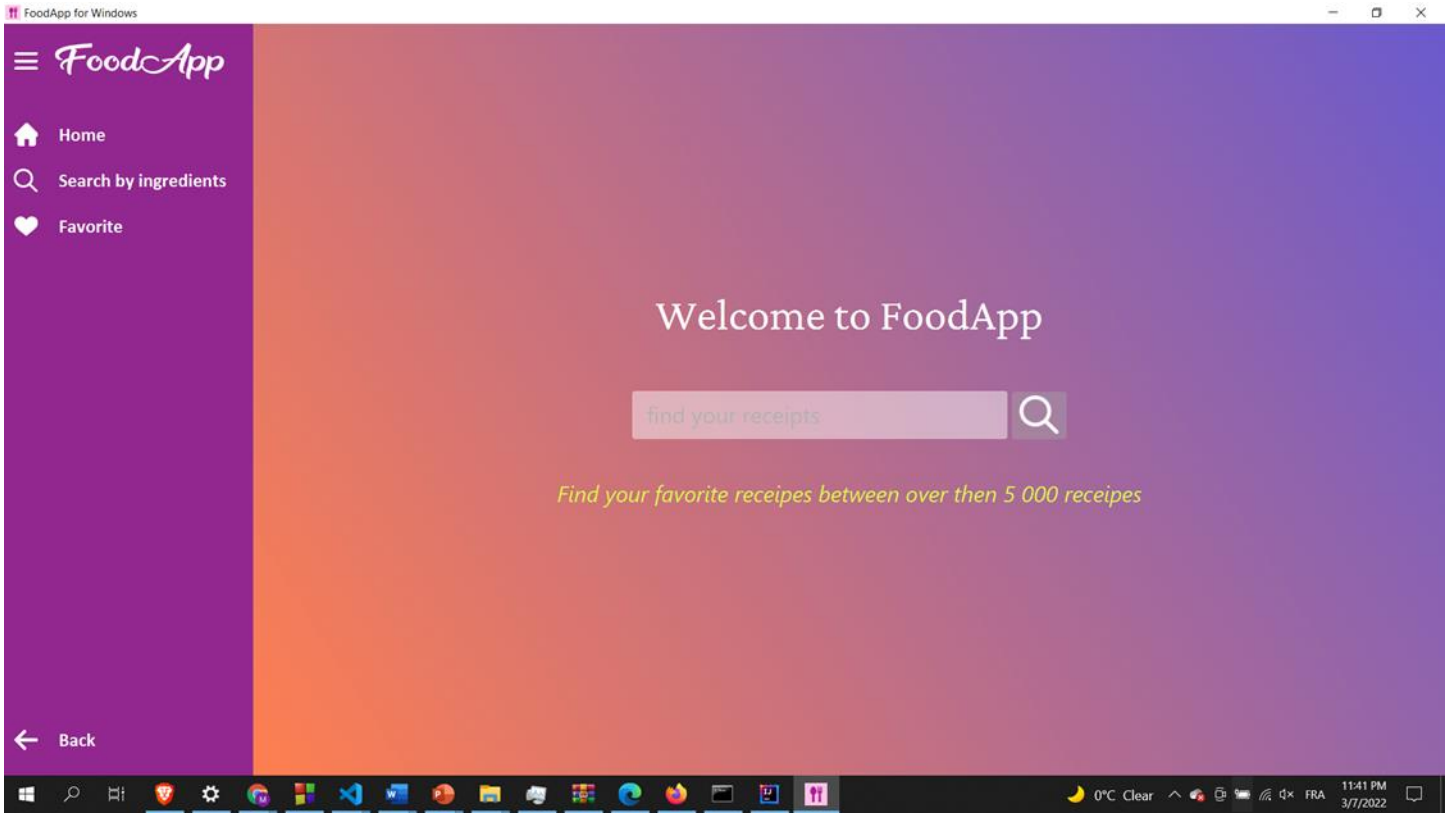
On a réalisé l'interface graphique de la page d'accueil, et de la barre de menu qui s'affiche à gauche, où on peut naviguer sur l'application. On a réalisé l'interface graphique de l'affichage des recettes sous forme des cartes. On a réalisé l'interface graphique de l'affichage des favoris.

Lot 3 :

On a réalisé l'interface graphique de l'affichage des ingrédients avec l'image, le nutriscore et les équipements et une petite description.

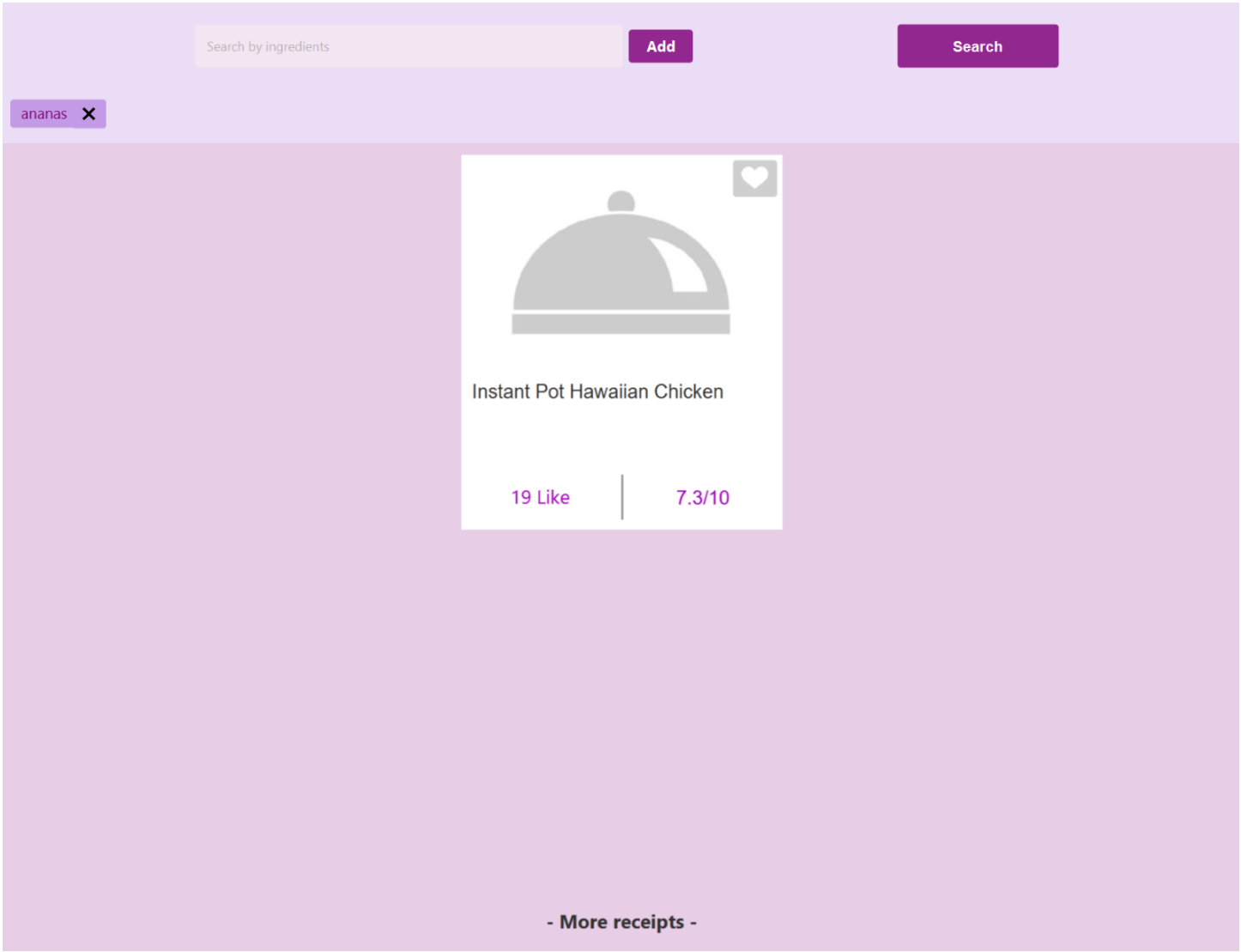
## Description des fonctionnalités :

Lorsque notre application **foodApp** est lancée, le client se retrouve au début dans la page d'accueil, et avec le bouton en haut à gauche il peut masquer ou afficher la barre de menu qui affiche les options de navigation. Dans la barre de menu il y a 5 boutons, le premier sert à afficher et masquer la barre de menu, le deuxième affiche l'accueil, le troisième affiche la page de recherche des recettes par ingrédients, le quatrième affiche les recettes de la liste des favoris et le dernier bouton sert à faire des retours aux pages consultés précédemment.

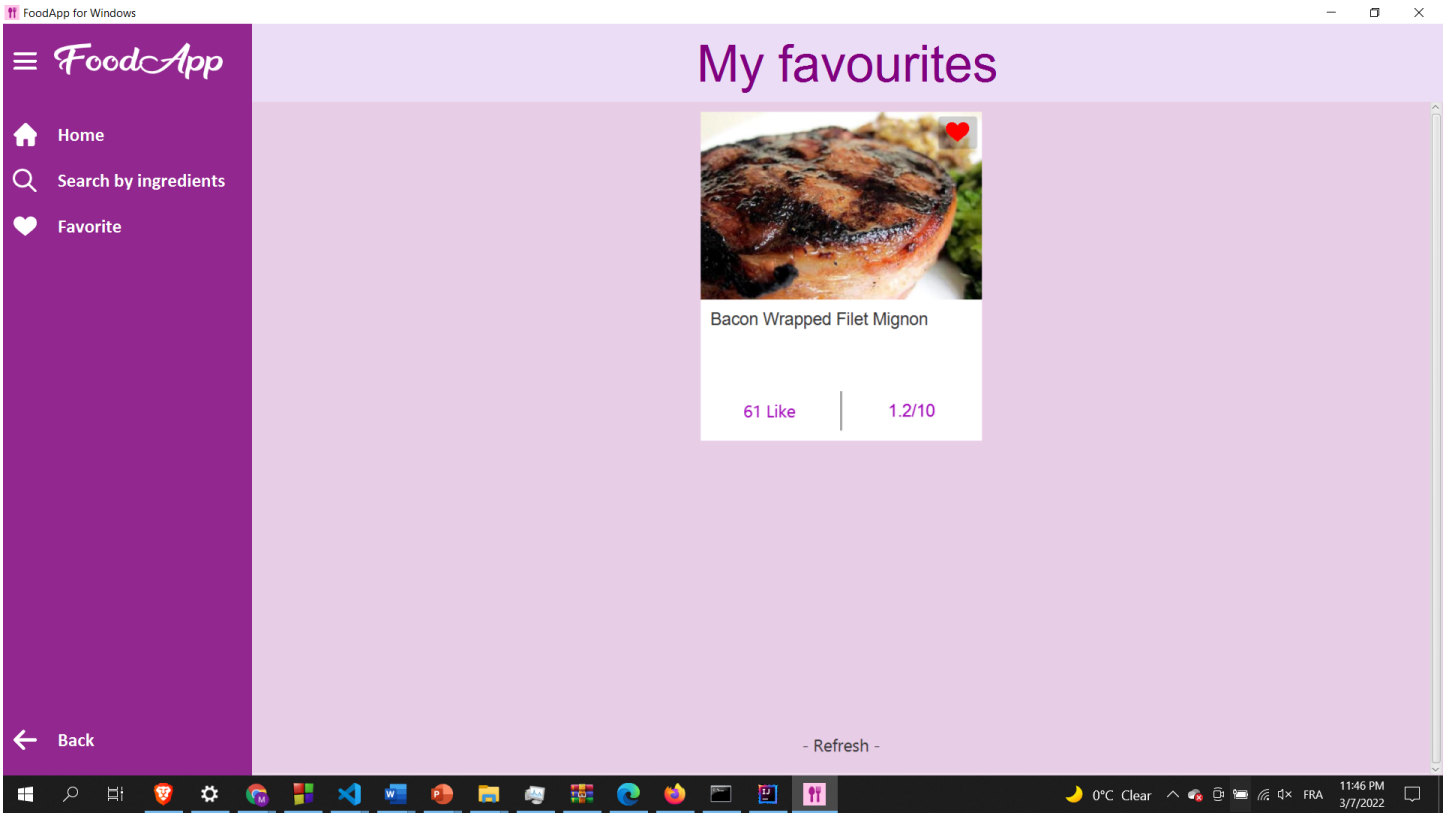



Lorsqu'on clique sur le bouton de recherche, on va sur la page de recherche de recettes par ingrédients, où on trouve un champ d'écriture, lorsqu'on tape un ingrédient dans ce champ et on clique sur le bouton **add**, on remarque l'ajout d'un bouton au-dessous du champ

d’écriture avec le nom de l’ingrédient, puis on continue comme ça si on veut ajouter plus d’ingrédients. Bien évidement on peut supprimer un ingrédient en cliquant sur la croix qui s’affiche dans le bouton à côté du nom de l’ingrédient. Pour afficher le résultat de notre recherche, on clique sur le bouton **Search** à droite du bouton **add**. Tout en bas de la page il y a un bouton “**More receipts**” qui affiche plus de recettes pour la même recherche.




En cliquant sur le bouton de favoris dans la barre de menu, une page s’affiche avec un titre en haut au milieux “**My favourites**”, puis toutes les recettes de la liste de favoris s’affichent sous forme de carte. Tout en bas il y a un bouton “**refresh**” qui actualise la liste des favoris si jamais il y a une modification, puis affiche son contenu.




Lorsqu'on clique sur les recettes affichées après une recherche ainsi qu'après l'affichage des favoris, on va dans une page où on peut voir une image de la recette, le nutiscore, les ingrédients de la recette, les équipements et une description de la recette.

FoodApp for Windows

FoodApp

Home

Search by ingredients

Favorite

Back

Bacon Wrapped Filet Mignon

By Foodista

Ready in 45 min

1.2/10

61 Like

ABCDE

## Summry

The recipe Bacon Wrapped Filet Mignon can be made in around 45 minutes. One serving contains 98 calories, 3g of protein, and 9g of fat. This recipe serves 4 and costs 39 cents per serving. 61 person were glad they tried this recipe. Head to the store and pick up Salt & Pepper, bacon, garlic, and a few other things to make it today. It works well as a cheap side dish. It is brought to you by Foodista. It is a good option if you're following a gluten free, dairy free, paleolithic, and primal diet. With a spoonacular score of 12%, this dish is not so excellent. Similar recipes are Bacon Wrapped Filet Mignon, Bacon Wrapped Filet Mignon, and Bacon Wrapped Filet Mignon.

0°C Clear FRA 11:45 PM 3/7/2022

FoodApp for Windows

FoodApp

Home

Search by ingredients

Favorite

Back

Pepper, bacon, garlic, and a few other things to make it today. It works well as a cheap side dish. It is brought to you by Foodista. It is a good option if you're following a gluten free, dairy free, paleolithic, and primal diet. With a spoonacular score of 12%, this dish is not so excellent. Similar recipes are Bacon Wrapped Filet Mignon, Bacon Wrapped Filet Mignon, and Bacon Wrapped Filet Mignon.

## Ingredients

- 4 inches beef tenderloin steaks
- 4 slice smoked bacon
- 2 large cloves garlic
- 1 serving Salt & Pepper

## Equipements

- toothpicks
- grill
- tongs

## Steps

step 1: Place the tenderloins on a large dish and wrap a slice of room temperature bacon around each filet, gently stretching the bacon if needed. Secure the ends of the bacon with a toothpick.

step 2: Cut the garlic cloves in half and rub both sides of each filet with the cut ends of the garlic. Season well with salt and pepper on both sides.

step 3: Pre-heat grill to very hot. With tongs, place each steak on the grill and for medium-rare cook for 3-4 minutes, rotating the steaks halfway through 90 degrees for nice grill marks.

step 4: Turn the steaks over and cook another 3-4 minutes.

step 5: Remove from the grill and let rest loosely covered for 5 minutes before removing toothpicks and serving.

step 6: Serve as is or with sauted garlic mushrooms

0°C Clear FRA 11:45 PM 3/7/2022

Pour ajouter ou supprimer une recette de la liste des favoris, il existe deux méthodes. Soit en cliquant sur le cœur affiché en haut à droite de chaque recette, soit en cliquant dans le cœur qui s'affiche dans la page après qu'on clique sur une recette. Si le cœur est rouge, ça veut dire que la recette est contenue dans la liste des favoris, sinon elle n'y appartient pas.

La consultation des recettes de la liste des favoris, peut se réaliser même sans connexion internet.

# Partie 3 :

## Ma contribution personnelle au projet :

## Implémentation de la classe Recette :

| Recette | |
|---|---|
| **Les attributs :**<br>Id | <br>Identifiant de la recette |
| **Les méthodes** | Toutes les méthodes commencent par consulter l'API si elles n'arrivent pas à recevoir des données, elles consultent le ficher json de favoris, dans ce cas si elles ne trouvent pas la recette elle return null. |
| ---<br>Public getId() :Entier | ---<br>Retourne l'identifient de la recette. |
| ---<br>Public getInformation() :Objet | ---<br>Retourne les informations de la recette. |
| ---<br>Public getSummary() : Objet | ---<br>Retourne le résumé de la recette. |
| ---<br>Public getEquipments() : Objet | ---<br>Retourne les équipements nécessaires pour faire la recette. |
| ---<br>Public getInstructions() : Objet | ---<br>Retourne les instruction a suivre pour faire la recette. |
| ---<br>Statique searchByIngredients(): Objet | ---<br>Cette fonction prend en paramètres : la liste des ingrédients, le nombre de recette et l'offset (le nombre de recettes à passer)<br>Et retourne une liste contient des recettes qui respecte la liste des ingrédients. |

## Implémentation de la classe JSONReader :

| JSONReader | |
|---|---|
| **Les attributs :** | |
| **Les méthodes :** | Toutes les méthodes commencent par consulter l'API si elles n'arrivent pas à recevoir des données, elles consultent le ficher json de favoris, dans ce cas si elles ne trouvent pas la recette elle return null. |
| Private getData() : Chaine de caractères | Prend une url en paramètre, créer une connexion internet, récupère les données et les retourne en forme chaine de caractère. |
| Statique getDataByUrl() :Objet | Récupère les données de la méthode getData() reforme ces données en format json et les retourne. |
| Statique getDataByPath() :Objet | Prend en paramètre un chemin vers un fichier json local, Récupère ses données et les retourne en format json. |
| Statique formatUrl() : Chaine de caractères | Cette méthode prend deux paramètres : une chaine de caractères contient la partie nom de domaine et la partie chemin d'une url, une HashMap des paramètres de url, et retourne une chaine de caractères contient une url complète. |

## Conception d'une partie de l'interface graphique et implémentation de ses classes de contrôles :

### La scène primaire : (foodapp.fxml + FoodappController.java)

Elle s'agit d'une scène fixée dans la fenêtre de l'application contient deux éléments principaux :

- Un panneau, pour la scène secondaire, dédié à afficher les défirent rubriques de l'application.

- Une barre de menu pliable (pour de gagner plus d'espace) permet à déplacer enter les défirent rubriques de l'application, elle contient :

    - Le logo l'application
    - Un bouton pour plier la barre de menu
    - Un bouton Back pour retourner aux scènes précédentes
    - Autres boutons associés aux rubriques de l'application.

| FoodappController | |
|---|---|
| **Les attributs :**<br>Les éléments de la rubrique.<br>FIFO : Liste chainée ----------> | Sauvegarde les scène précédentes |
| **Les méthodes :** | |
| ----------<br>Initialize() | ----------<br>Initialise le panneau par la rubrique d'accueil Home |
| ----------<br>menuOnOff()<br>goBack()<br>goHome()<br>favourite()<br>goSearchByIngredients() | ----------<br>Des Auditeurs associés aux boutons de la barre menu. |
| ----------<br>Public load() : Contrôleur | ----------<br>Prend en paramètre un chemin vers un panneau de type fxml, elle le lève sur la scène et elle l'ajoute sur FIFO |

## La rubrique d'accueil : (home.fxml + HomeController.java)

Elle contient un message d'accueil, une barre de recherche et quelles autres informations.

| HomeController | |
|---|---|
| **Les attributs :**<br>Les éléments de la rubrique. | |
| **Les méthodes :** | |

## La rubrique de recherche : (search.fxml + SearchController.java)

Dans cette rubrique, l'utilisateur peut faire des recherches de recettes par ces ingrédients.

| SearchController | |
|---|---|
| **Les attributs :**<br>Les éléments de la rubrique. | |
| **Les méthodes :**<br>addIngredientButton()<br>---<br>search()<br>---<br>Load()<br>---<br>loadMore()<br>---<br>showNoReceiptFound()<br>showNoInternet() | Ajoute un ingrédient saisi par l'utilisateur dans liste et elle l'affiche avec un bouton de suppression.<br>---<br>Auditeur associé au bouton Search qui lance une recherche dans l'API.<br>---<br>Elle télécharge les certes des recettes sur le panneau des cartes.<br>---<br>Auditeur associé au bouton More, qui Télécharge plus de recette sur le panneau des cartes.<br>---<br>Ces méthodes servent à afficher des messages en cas l'échoue de la recherche. |

## La carte de recette: (carte.fxml + CarteController.java)

Après la récupération de data qui contient les recettes recherchées, la classe contrôleur affiche ces recettes dans des cartes contienne l'image, le titre et quelques information générale de chaque recette.

| CarteController | |
|---|---|
| **Les attributs :**<br>Les éléments de la carte. | |
| **Les méthodes :** | |
| set() | Prend en paramètre une recette et initialise ses informations dans la classe. |
| update() | Mise en place les informations et l'image de la recette dans la carte |
| getReceipt() | Auditeur associé a la carte qui prend vers la rubrique Recette qui affiche le détail de la recette montrée sur la carte. |

## La rubrique Recette: (receipt.fxml + ReceiptController.java)

Cette rubrique affiche le détail de la recette tel que les ingrédients, les instructions, les équipements …

| ReceiptController | |
|---|---|
| **Les attributs :** | |
| **Les méthodes :** | |
| setReceiptInformation() | Initialise l'objet qui contient les informations de la recette. |
| updateReceiptInformation() | Mise en place l'image les informations générales de la recette dans le panneau d'information. |
| updateReceipt() | Mise en place les informations détaillés de la recette dans la rubrique. |

## Les difficultés envisagées :

### Difficulté liée à l'organisation de travail :

Le manque d'un cahier de charge de développeur qui modalise l'architecture logicielle et qui me permet de comprendre les parties de mes coéquipières ainsi qu'unifier la structure de données utilisée.

### Difficulté liée au côté développement :

J'ai envisagé quelques erreurs de développent dont je ne suis pas arrivé à comprendre et à résoudre, par exemple le centenaire de texte **Text** génère des erreurs chaque j'essaie de l'utilise alors j'étais oblige je le changer par **Label** qui n'est pas dédié à la base de supporter les long texte.