

# **RAPPORT DE PROJET**

## **Recette Culinaire**

**Génie Logiciel Groupe 8**

**ROCKA Haumana**

**TUAHIVAATETONOHITI Manihiki**

**MOLL Teanuanua**

**Licence Informatique 3ème Année**

## **1. Présentation du contexte**

## **2. Présentation du sujet**

### **a. Description générale**

### **b. Notre approche du sujet**

## **3. Développement de la démarche**

### **a. Conception et diagrammes**

### **b. Tests et développement**

## **4. Analyse des résultats**

## **5. Difficultés rencontrées et Axes d'améliorations**

## **6. Conclusion**

## **1. Présentation du contexte**

En cours de génie logiciel, dans le cadre de notre troisième année de licence informatique, nous avons été chargés de concevoir et développer un projet en groupe. Ce projet visait à mettre en œuvre des concepts clés de génie logiciel, notamment la modélisation, le développement itératif et les tests logiciels.

## **2. Présentation du sujet**

### **a. Description générale**

Le projet consiste à développer un programme permettant à des utilisateurs de gérer et partager des recettes culinaires.

Les fonctionnalités clés incluent :

- Ajouter une recette avec un titre, des ingrédients, des étapes de préparation, et un temps nécessaire.
- Rechercher des recettes par ingrédient ou type de plat.
- Noter et commenter une recette après l'avoir essayée.

- Consulter les recettes les mieux notées et les plus populaires.

Le programme doit également :

- Afficher une moyenne des notes attribuées à chaque recette.
- Permettre aux utilisateurs de consulter et d'imprimer une recette complète.

## **b. Présentation du sujet selon notre point de vue**

Notre équipe a abordé ce projet en se concentrant sur la création d'une solution qui permet non seulement la gestion des recettes, mais aussi une interaction active entre les utilisateurs. Notre équipe a choisi d'utiliser le cycle en V pour structurer les phases de conception, de développement, et de tests. Nous avons choisi de nous inspirer de plateformes comme **Marmiton** pour mieux appréhender les attentes des utilisateurs et comprendre les fonctionnalités essentielles d'un système de gestion de recettes.

Une de nos priorités était de rendre l'expérience utilisateur fluide et intuitive. Pour cela, nous avons intégré des éléments comme la possibilité pour chaque utilisateur de personnaliser son expérience en sauvegardant des recettes préférées ou en créant et partageant ses propres créations culinaires.

Nous avons également veillé à concevoir des outils permettant une interaction sociale enrichissante, comme la capacité de noter les recettes et de laisser des commentaires constructifs à condition de les avoir essayées. Ces fonctionnalités favorisent l'échange et la découverte au sein de la communauté.

Enfin, afin de répondre aux besoins pratiques, nous avons intégré la possibilité d'imprimer une recette dans un format complet et lisible, idéal pour un usage hors ligne. Parallèlement, un module administratif a été pensé pour superviser les activités sur la plateforme et garantir une expérience utilisateur de qualité.

## **3. Développement de la démarche**

### **a. Conception et diagrammes**

#### Diagramme de cas d'utilisation :

Ayant découvert le sujet, nous avons discuté ensemble du projet et des attentes demandées en commençant par un diagramme de cas qui a été rapide et unanime. **(Voir Annexe 1)**. Le diagramme de cas d'utilisation a été le point de départ de notre conception. Il a permis d'identifier les principaux acteurs et actions du système.

#### Diagramme de classes :

Nous avons continué la conception avec une première version de diagramme de classes. **(Voir Annexe 2, Figure 1)**. Ce diagramme n'ayant pas de classe pilotant l'ensemble, nous dirigea, après conseils de Monsieur Chabrier, vers une deuxième et dernière version plus compréhensible. **(Voir Annexe 2, Figure 2)**.

Le diagramme de classes modélisant toutes les différentes entités, leur composition, ainsi que leurs liens entre eux, pouvant servir de modèle pour le développement du programme.

### Diagramme d'activités :

Pendant ces rectifications, la création d'un diagramme d'activités montrant les suites d'actions possible faites par un utilisateur. **(Voir Annexe 3).**

### Diagramme de séquence :

Et la création d'un diagramme de séquence pour identifier les comportements attendus lors de l'ajout d'une recette par un utilisateur. **(Voir Annexe 4, Figure 1).** Ce diagramme montre les échanges et méthodes nécessaires entre les objets afin d'effectuer l'ajout d'une recette jusqu'à l'aboutissement, que cela soit accepté par le programme et enregistrer ou non. L'utilisateur peut ensuite choisir de faire une autre action, telle que rechercher une recette **(Voir Annexe 4, Figure 2)** ou bien même commenter et noter une recette. **(Voir Annexe 4, Figure 3).**

### Maquette de l'interface :

Des maquettes ont été conçues avec **Figma** pour représenter l'interface utilisateur, en mettant l'accent sur la simplicité et l'ergonomie. **(Voir Annexe 5, Figure 1 et Figure 2).** Grâce à cela, la conception du Design étant simple offre un gain de temps intéressant.

## **b. Tests et développement**

### Développement du programme :

Le développement s'est appuyé sur la méthode **Test-Driven Development (TDD)**. Chaque fonctionnalité a été développée et validée par des tests unitaires JUnit avant son intégration.

#### **Exemples des tests écrits :**

**Recherche par ingrédient** : un test a été écrit pour valider que les recettes contenant un ingrédient spécifique s'affichent correctement. **(Voir Annexe 6, Figure 1, 2 et 3)**

**Affichage par note** : un autre test vérifiait que les recettes les mieux notées apparaissent en tête de liste. **(Voir Annexe 6, Figure 4, 5 et 6)**

## **4. Analyse des résultats**

Bien que plusieurs composants du projet aient été achevés, nous n'avons pas réussi à finaliser le programme. La conception, incluant les diagrammes et maquettes, a été réalisée de manière satisfaisante, mais le développement du code a rencontré des obstacles.

## **5. Difficultés rencontrées et axes d'amélioration**

## Difficultés rencontrées

1. **Manque de communication** : Une coordination insuffisante a conduit à des retards dans certaines tâches.
2. **Planning chargé** : Les agendas personnels ont rendu difficiles les réunions de groupe régulières.
3. **Division du travail inefficace** : Les tâches n'ont pas été assignées de manière claire ou équilibrée.
4. **Absence de leadership** : L'absence d'un leader clair a freiné la prise de décision et la résolution rapide des problèmes.

## Axes d'amélioration

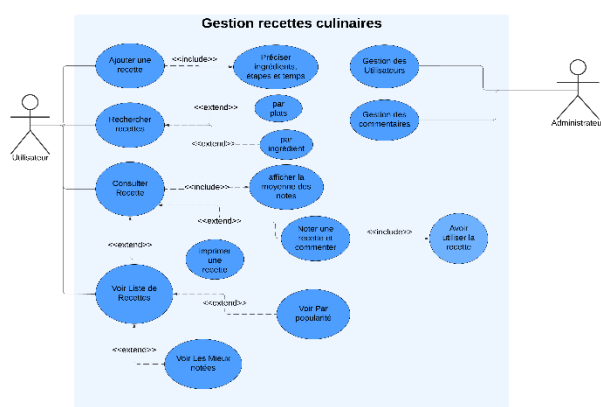
1. **Amélioration de la communication** : Utilisation d'outils comme Slack ou Trello pour rester connectés et mieux suivre les tâches.
2. **Planification régulière** : Organisation de réunions hebdomadaires pour assurer le suivi.
3. **Clarification des rôles** : Définir clairement les responsabilités de chaque membre dès le début.
4. **Désignation d'un leader** : Un membre doit être désigné comme responsable pour arbitrer les décisions.

## 6. Conclusion

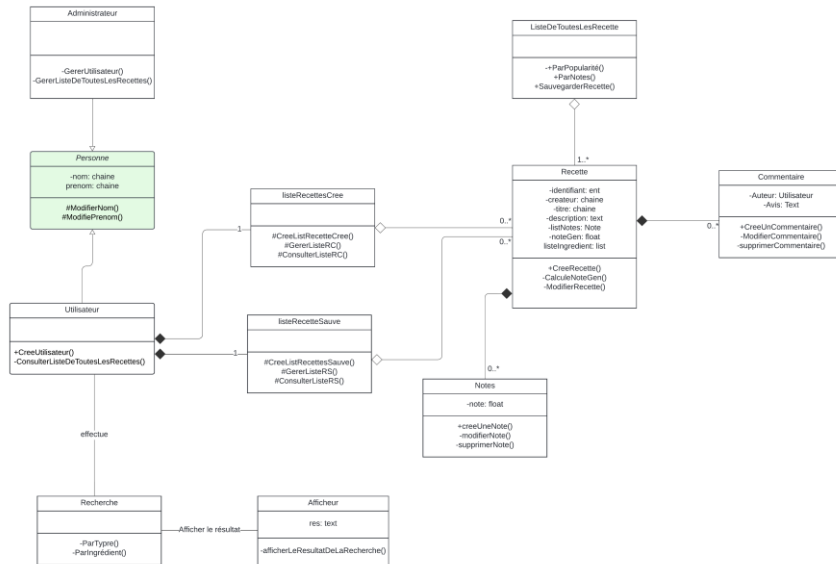
Ce projet a été une expérience enrichissante, malgré les défis rencontrés. Il nous a permis de mettre en pratique des concepts de génie logiciel et d'identifier les lacunes dans notre organisation de travail en groupe. Les solutions proposées pour surmonter ces obstacles seront essentielles pour la réussite de projets futurs.

# Annexe

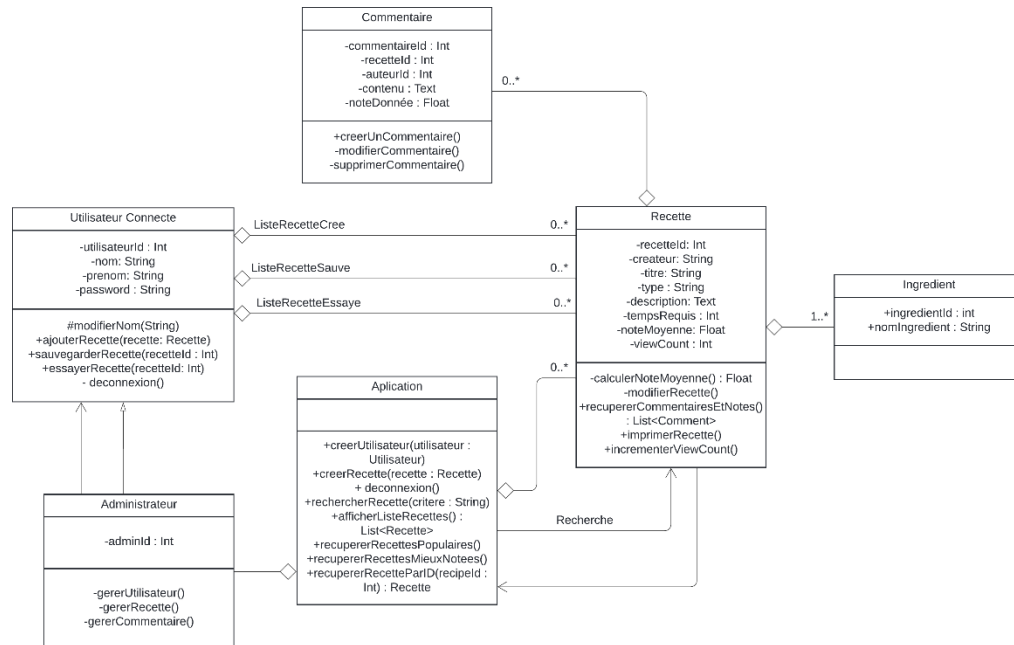
## Annexe 1



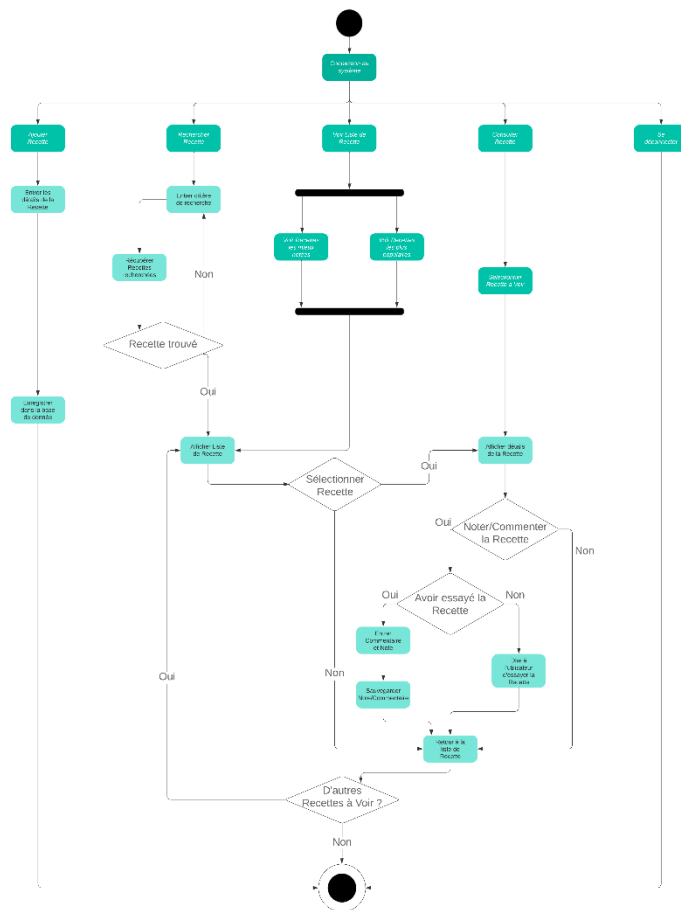
## Annexe 2, Figure 1



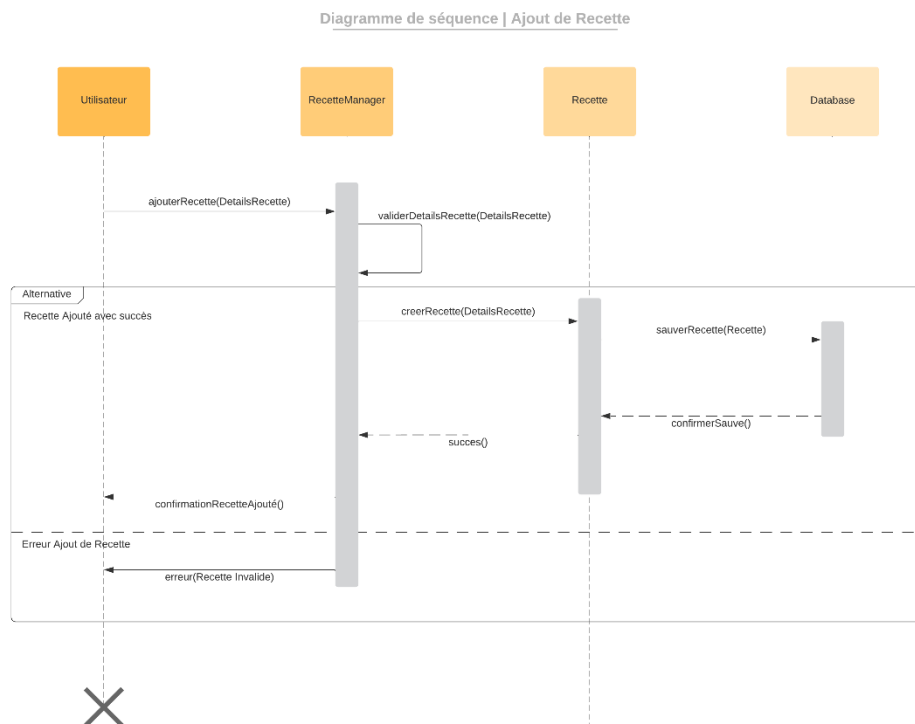
Annexe 2, Figure 2



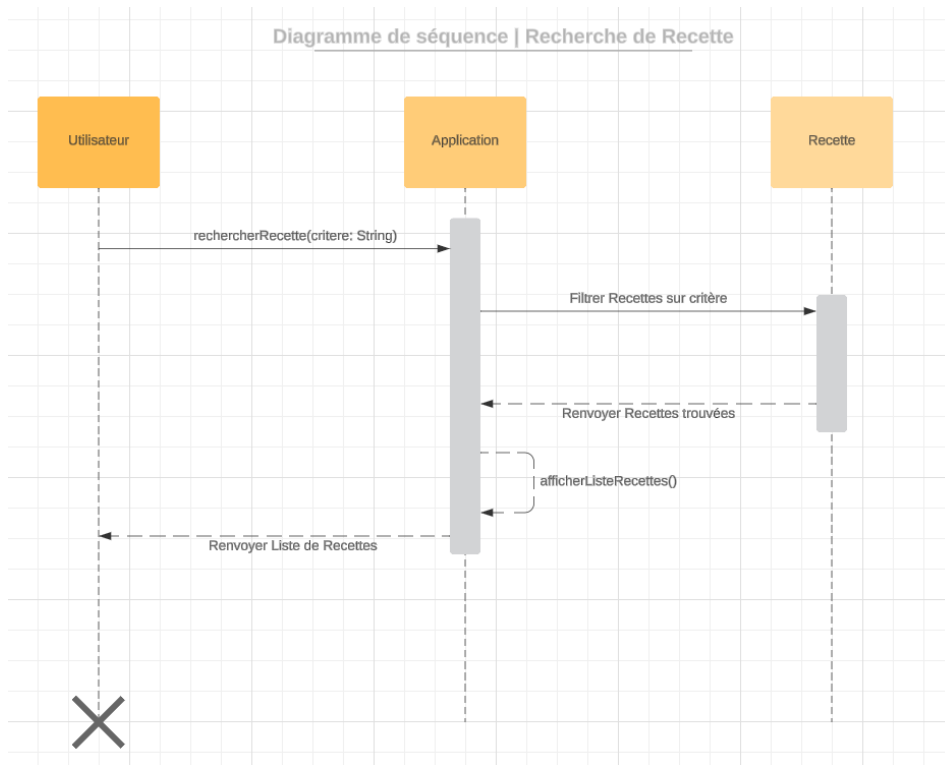
Annexe 3



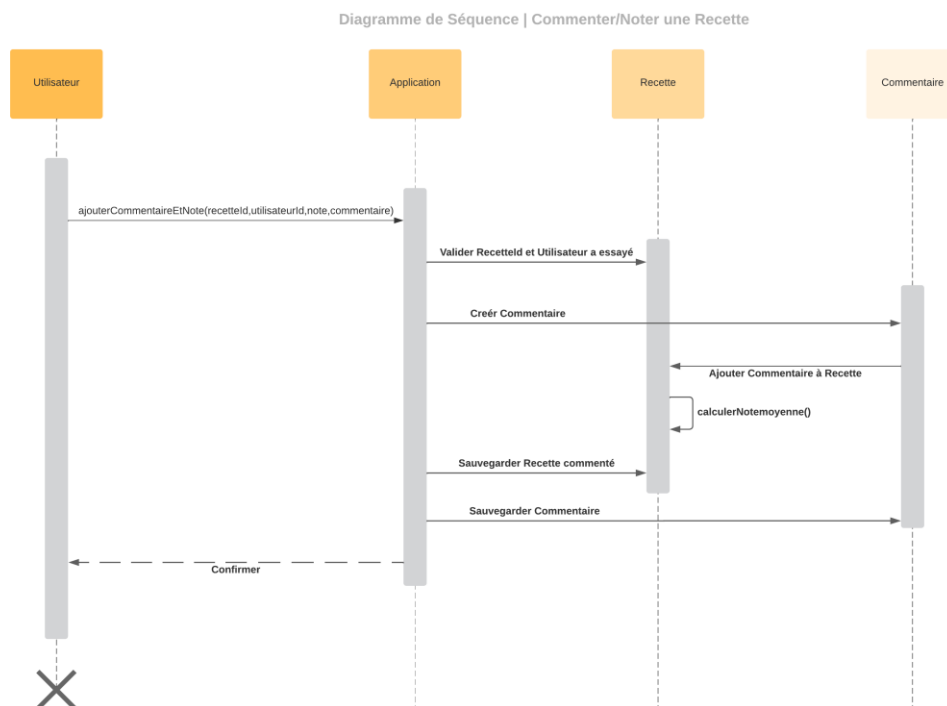
Annexe 4, Figure 1



Annexe 4, Figure 2

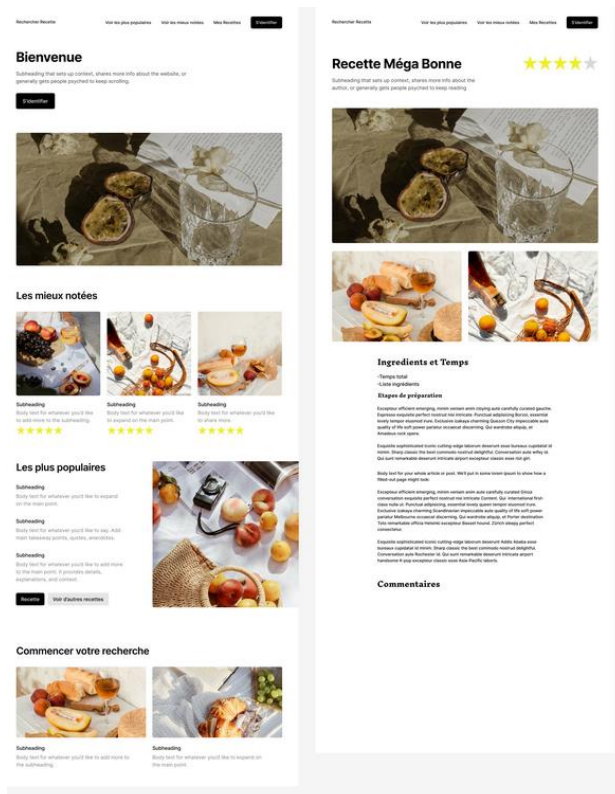


Annexe 4, Figure 3

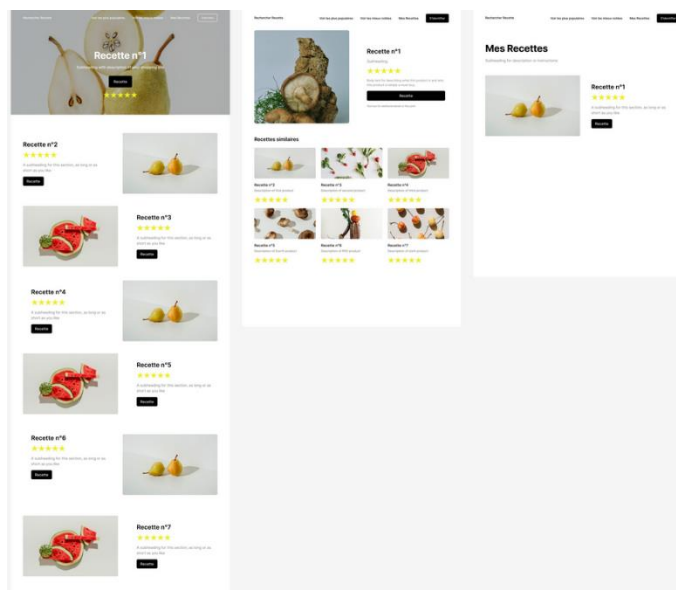


Annexe 5, Figure 1

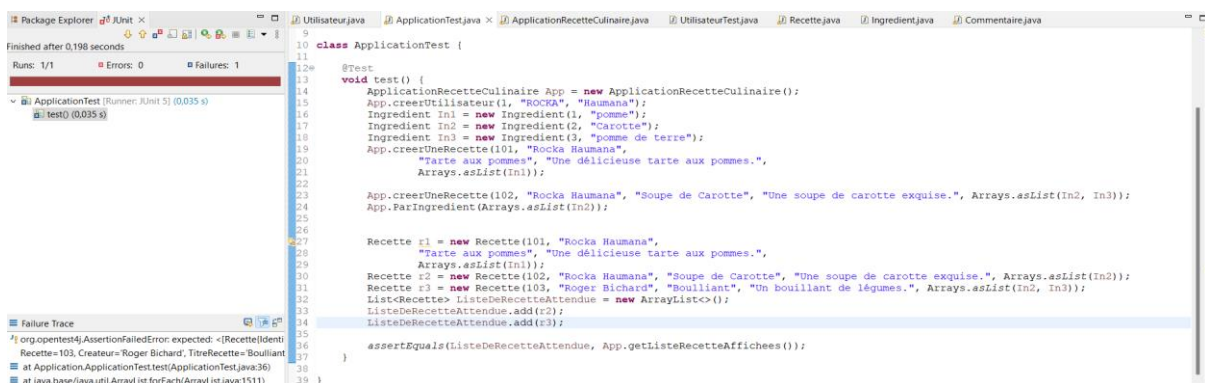




Annexe 5, Figure 2



Annexe 6, Figure 1



Annexe 6, Figure 2



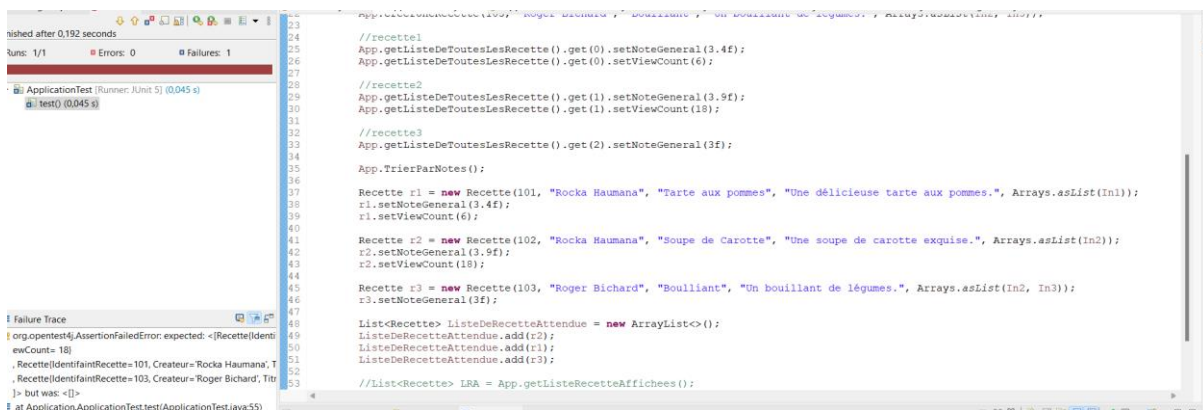
Annexe 6, Figure 3

```

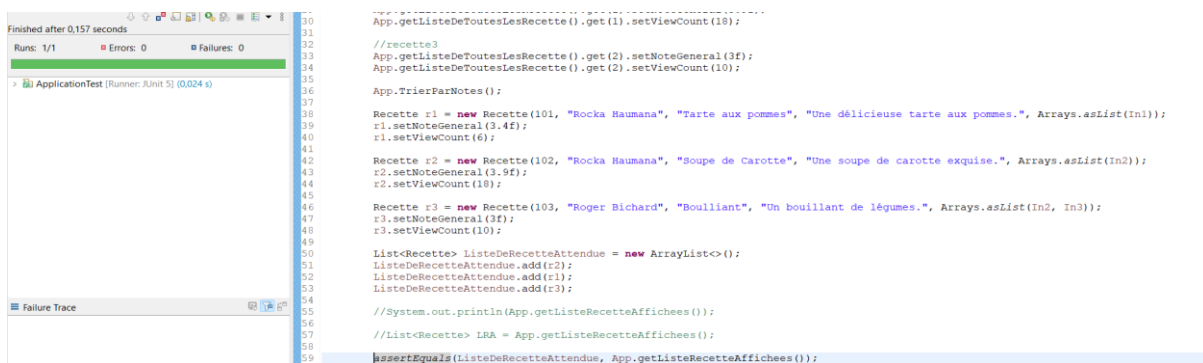
public void ParIngredient(List<Ingredient> ListeD_ingredient) {
    // Filtrer les recettes qui contiennent tous les ingrédients demandés
    ListeRecettesAffichees = ListeDeToutesLesRecettes.stream()
        .filter(recette -> recette.getListeIngredient().containsAll(ListeD_ingredient))
        .collect(Collectors.toList());
}

```

Annexe 6, Figure 4



Annexe 6, Figure 5



Annexe 6, Figure 6

```

protected void TrierParNotes() {
    List<Recette> copieTrie = ListeDeToutesLesRecettes.stream()
        .sorted(Comparator.comparing(Recette::getNoteGeneral).reversed())
        .collect(Collectors.toList());

    ListeRecettesAffichees = copieTrie.stream()
        .limit(10)
        .collect(Collectors.toList());
}

```