

Fiche d'Investigation de Fonctionnalité - Recherche des recettes Fonctionnalité #03

Problématique : L'objectif de cette fonctionnalité est de permettre à l'utilisateur d'accéder rapidement à une recette correspondant à ses besoins en filtrant les recettes déjà reçues. Nous cherchons à comparer deux approches d'implémentation pour optimiser la recherche et offrir une expérience fluide à l'utilisateur.

Option 1 : Implémentation avec des boucles natives (Version A) Dans cette option, nous utiliserons des **boucles natives** telles que while, for, etc. pour effectuer la recherche des recettes. Cette approche est classique et peut être directe pour implémenter la fonctionnalité. Cependant, nous devons évaluer si elle offre des performances optimales lors de l'interrogation d'une grande quantité de données.

Avantages

- ⊕ Implémentation classique et directe
- ⊕ Utilisation des **boucles natives** pour la recherche

Inconvénients

- ⊖ Possibilité de performance sous-optimale avec de grandes quantités de données
- ⊖ Peut nécessiter une optimisation supplémentaire pour une expérience fluide

Option 2 : Implémentation en programmation fonctionnelle (Version B) Dans cette option, nous utiliserons les **méthodes de l'objet array** telles que foreach, filter, map, reduce, etc. pour effectuer la recherche des recettes. Cette approche peut être plus élégante et expressive, mais nous devons évaluer ses performances et sa capacité à gérer efficacement une recherche complexe.

Avantages

- ⊕ Implémentation élégante et expressive en utilisant des **méthodes array**
- ⊕ Possibilité d'optimisation des performances avec des méthodes adaptées

Inconvénients

- ⊖ Nécessite une bonne connaissance des méthodes array pour une implémentation optimale
- ⊖ Possibilité d'effet de bord sur les performances en cas d'usage incorrect des méthodes

Schéma (Algorithme) : (Décrire ici le schéma pour chaque version avec draw.io)

Version A (Boucles Natives) :

- L'utilisateur entre au moins 3 caractères dans la barre de recherche principale.
- Le système utilise des **boucles natives** pour rechercher les recettes correspondant à l'entrée utilisateur dans le titre, les ingrédients et la description des recettes.
- Les résultats de la recherche sont affichés dans l'interface.
- Les champs de recherche avancée sont actualisés avec les informations sur les ingrédients, les ustensiles et les appareils des recettes restantes.

Version B (Programmation Fonctionnelle) :

- L'utilisateur entre au moins 3 caractères dans la barre de recherche principale.
- Le système utilise les **méthodes de l'objet array** pour rechercher les recettes correspondant à l'entrée utilisateur dans le titre, les ingrédients et la description des recettes.
- Les résultats de la recherche sont affichés dans l'interface.
- Les champs de recherche avancée sont actualisés avec les informations sur les ingrédients, les ustensiles et les appareils des recettes restantes.

Ces deux versions se focalisent uniquement sur le champ de recherche principal pour la comparaison.

Solution retenue : Nous devons maintenant implémenter ces deux versions dans des branches distinctes sur Git et les tester pour évaluer leurs performances. La solution retenue sera basée sur l'analyse comparative des performances des deux versions. Une fois les tests effectués, nous choisirons l'approche offrant la meilleure expérience utilisateur et les performances les plus optimales pour la fonctionnalité de recherche des recettes.

JSben ne fonctionne pas chez moi je ne peux faire le comparatif des 2 code car leur site a planté à ce jour.

capture d'écran :

