

Fiche d'investigation de fonctionnalité

Fonctionnalité : Affichage des recettes	Fonctionnalité #1 et #2
Problématique : Afin d'offrir une expérience utilisateur optimale, nous cherchons à optimiser la performance de l'affichage des recettes avec une recherche en temps réel	

Option 1 : Utilisation de la méthode "filter" pour la recherche

Dans cette option, nous utilisons la méthode "filter" pour filtrer les recettes selon les mots-clés de recherche. Cette méthode est asynchrone et utilise une approche de programmation fonctionnelle.

Avantages

- ⊕ Fonction asynchrone qui améliore la performance de l'affichage
- ⊕ Programmation fonctionnelle qui simplifie la lecture et la maintenance du code
- ⊕ Permet une recherche en temps réel grâce à l'ajout de caractères dans la barre

Inconvénients

- ⊖ Peut ralentir la performance si la liste de recettes est très longue
- ⊖ Nécessite l'utilisation de "split" et "some" pour filtrer les mots-clés

Option 2 : Utilisation d'une boucle "for" pour la recherche

Dans cette option, nous utilisons une boucle "for" pour parcourir les recettes et vérifier si elles contiennent les mots-clés de recherche. Cette méthode est également asynchrone mais utilise une approche plus impérative.

Avantages

- ⊕ Fonction asynchrone qui améliore la performance de l'affichage
- ⊕ Programmation impérative qui est plus facilement compréhensible pour les développeurs débutants
- ⊕ Permet une recherche en temps réel grâce à l'ajout de caractères dans la barre de recherche

Inconvénients

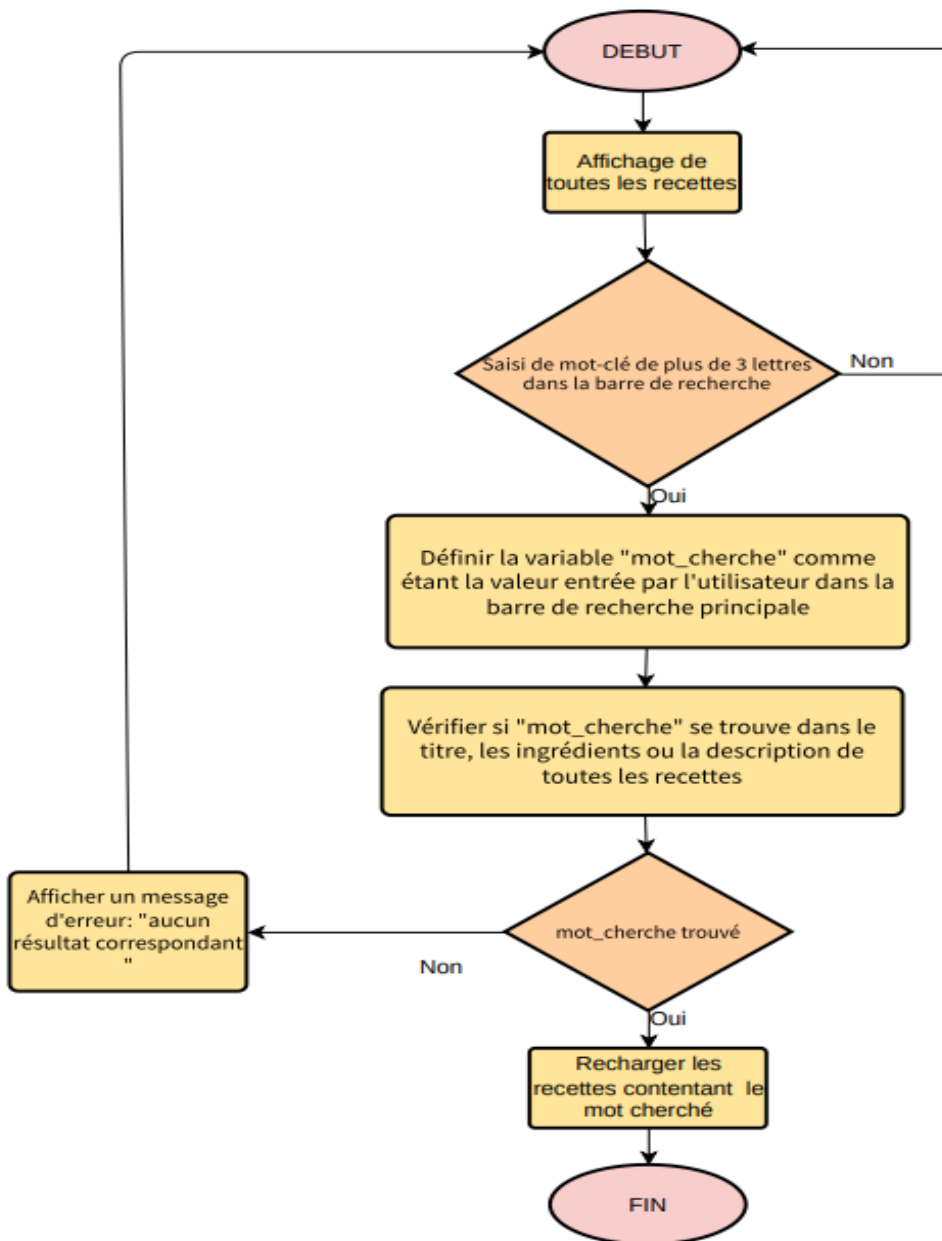
- ⊖ Peut ralentir la performance si la liste de recettes est très longue
- ⊖ Nécessite une boucle "for" pour parcourir les recettes

Solution retenue :

Malgré le fait que l'option 2 obtient un score plus performant avec l'outil **jsbench.ch** après comparaison de plusieurs blocs de codes avec l'usage des boucles **For** versus **la programmation fonctionnelle** avec les méthodes modernes des tableaux en JS telles que **filter, map, foreach**, etc ..., j'ai retenu l'option 1 car elle utilise une approche plus moderne avec la programmation fonctionnelle, qui est plus facilement **lisible** et **maintenable** pour **la recherche principale**. De plus, cette méthode permet une recherche en temps réel, ce qui améliore l'expérience utilisateur. Toutefois, il est important de noter que cette option peut ralentir la performance si la liste de recettes devienne volumineuse avec le temps.

Ce choix est conforté lors de la gestion de **la recherche avancée** avec les différents filtres : ingrédients, appareils et ustensiles. En effet avec **l'approche fonctionnelle**, on décompose le problème en un ensemble de fonctions qui doivent être exécutées et on définit les entrées et sorties de chaque fonction. A contrario, avec **une approche impérative**, on spécifie les étapes pour atteindre un objectif, donc de façon procédurale, ce qui peut entraîner moins de flexibilité (modularité) dans le code et l'ordre d'exécution est très important.

RECHERCHE SANS FILTRE OPTIONNEL



RECHERCHE AVANCEE AVEC FILTRES

Ingrédients et/ou Appareils et/ou Ustensiles

