

Fiche d'investigation de fonctionnalité

Fonctionnalité : Recherche de recettes	Fonctionnalité #2
Problématique : L'objectif est de proposer un moteur de recherche rapide et fluide, capable de trier efficacement un large nombre de recettes en fonction d'un mot-clé. Cette recherche est un élément crucial de l'expérience utilisateur pour permettre un accès facile et rapide aux informations souhaitées.	

Option 1 : Programmation Fonctionnelle Dans cette option, la recherche est effectuée en utilisant des méthodes fonctionnelles disponibles dans JavaScript comme filter, map, reduce, et forEach. L'algorithme parcourt les recettes, puis vérifie si le mot-clé est présent dans les ingrédients, ustensiles, ou le nom de la recette.	
Avantages <ul style="list-style-type: none">⊕ Utilisation des fonctionnalités modernes de JavaScript⊕ Code plus concis et lisible	Inconvénients <ul style="list-style-type: none">⊖ Moins performant pour de très grands volumes de données⊖ Moins optimal en termes de consommation mémoire
Nombre de champs minimum analysés : 3 (nom de la recette, ingrédients, ustensiles)	

Option 2 : Boucles Natives Dans cette option, la recherche est effectuée via des boucles natives (for, while, etc.), permettant un contrôle plus direct du flux d'exécution. Cela implique que chaque recette est vérifiée de manière séquentielle et les résultats sont ajoutés manuellement à une collection.	
Avantages <ul style="list-style-type: none">⊕ Meilleure performance sur de grands ensembles de données	Inconvénients <ul style="list-style-type: none">⊖ Code plus long et plus difficile à lire⊖ Nécessite plus de gestion manuelle des états
Nombre de champs minimum analysés : 3 (nom de la recette, ingrédients, ustensiles)	

Solution retenue : Nous avons retenu l'option Programmation Fonctionnelle. En plus de ses meilleures performances, elle offre une meilleure lisibilité et maintenabilité, ce qui est essentiel pour la pérennité du projet.



Annexes

cf. Draw.io file

Figure 1 - Alorigramme de la barre recherche