Fiche d’investigation de fonctionnalité

Fonctionnalité : Recherche	**Fonctionnalité #1**
Problématique : L’objectif est de déterminer quelle méthode entre **fonctionnelle** et **native** est la plus efficace pour notre algorithme de recherche de recettes.	

Option 1 : Version native

La boucle **for** offre un contrôle total et une grande flexibilité, idéale pour les scénarios complexes et pour les performances optimales, surtout avec de grands ensembles de données. Compatible avec toutes les versions de JavaScript.

La boucle **for** peut être plus complexe à écrire et comprendre, et plus sujette à des erreurs comme les boucles infinies ou les erreurs d'indexation.

Avantages :	**Inconvénients :**
● Contrôle et Flexibilité ● Performance ● Compatibilité universelle	● Complexité ● Risque d'erreur

Option 2 : Version fonctionnelle

L’itération **forEach** facilite la lisibilité et la simplicité du code, convient bien à la programmation fonctionnelle, et automatise l'itération, réduisant ainsi le risque d'erreurs.

Cependant elle est moins flexible que la boucle for, notamment pour l'itération à rebours ou le saut d'éléments, et peut être légèrement moins performante pour de très grands ensembles de données.

Avantages :	**Inconvénients :**
● Lisibilité ● Moins d’erreurs	● Performance ● Moins flexible

Solution retenue :
Nous avons choisi d'utiliser l’**option 2** pour le projet, privilégiant ainsi la lisibilité et la simplicité du code. Cette approche facilite la maintenance, réduit les erreurs courantes et se concentre sur l'action à réaliser pour chaque élément du tableau.

Annexes

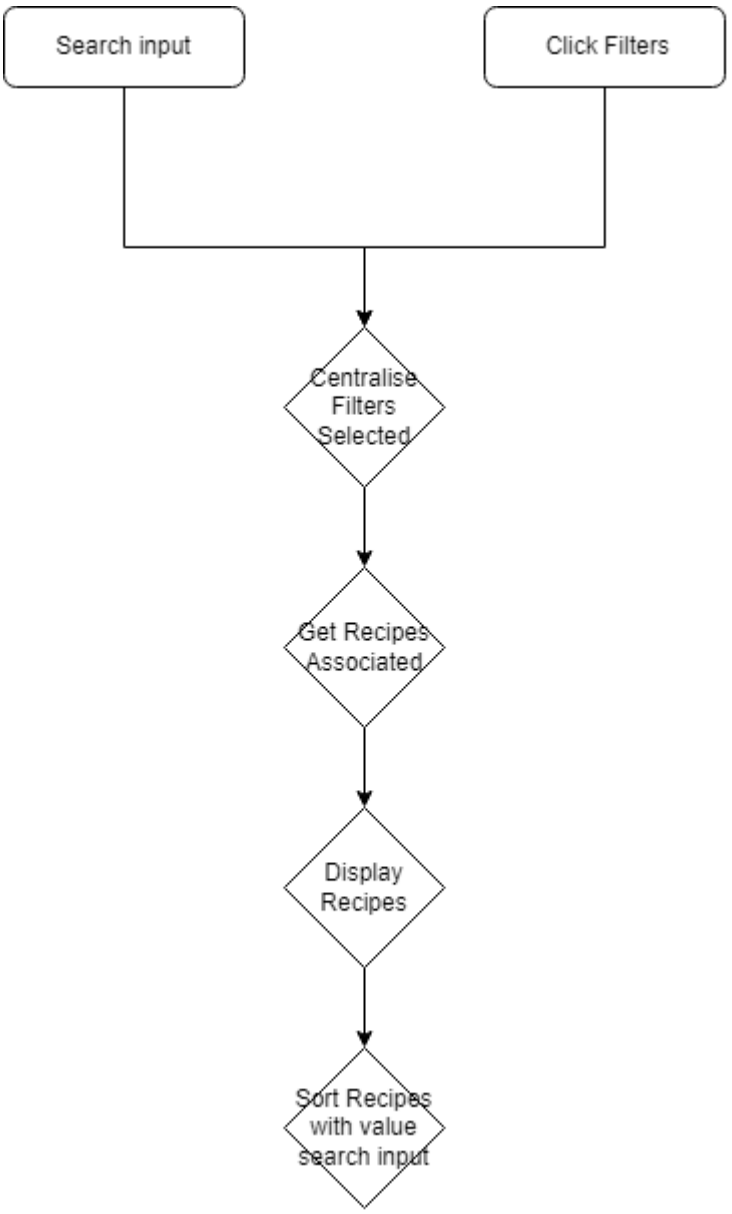

Diagramme algorithme de recherche