Fiche d’investigation de fonctionnalité

|  |  |
| --- | --- |
| **Fonctionnalité :** ​Recherche | **Fonctionnalité #1** |
| **Problématique :** Créer un algorithme capable de trier les recettes récupérées dans le backend en utilisant des mots clés et/ou des filtres permettant de sélectionner les ingrédients, appareils et ustensiles. Cet algorithme doit être le plus efficace et rapide possible. | |

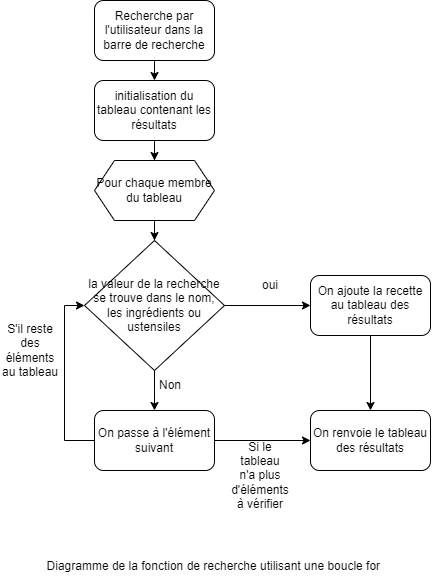
|  |  |
| --- | --- |
| **Option 1 : Création et utilisation d’une boucle *for* pour trier les éléments du tableau des recettes**  Dans cette option, nous utilisons une boucle for afin de trouver les éléments du tableau correspondant aux termes de la recherche. Le tableau des recettes contient toutes les recettes du site et lorsqu’un mot clé est entré dans la barre de recherche, l’algorithme va parcourir tous les éléments du tableau et retourner ceux contenant le mot clé dans un nouveau tableau résultat. Si aucun résultat n’est retourné, on peut afficher un message d’erreur. | |
| **Avantages**  ​⊕ Création d’un tableau regroupant les résultats et pouvant être facilement exploité  ​ ⊕ Utilisation d’une boucle *for* simple d’utilisation et pouvant être modifiée facilement selon les besoins | **Inconvénients**  ⊖ L’utilisation d’une boucle *for* peut être plus longue que l’utilisation de fonctions spécialisées pour le tri. |
| **Champs à utiliser :** Barre de recherche | |

|  |  |
| --- | --- |
| **Option 2 : Utilisation de la fonction d’objet Array.filter()**  Dans cette option, on utilise la fonction filter() permettant de retourner seulement les résultats correspondant à la recherche (qui remplissent la condition validée par la fonction *callback*. | |
| **Avantages**  ⊕​ Fonction spécialisée dans le tri.  ​⊕​ Performances supérieures à une boucle for.  ⊕ Retourne directement les éléments correspondant à la recherche | **Inconvénients**  ⊖ Moins de transparence sur son fonctionnement |
| **Champs à utiliser :** Barre de recherche | |

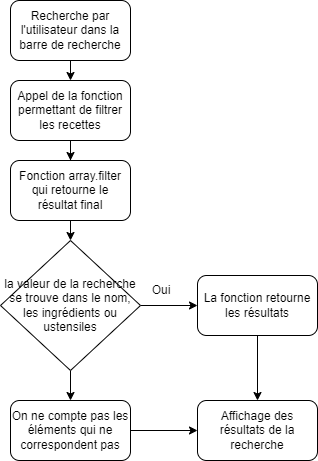
**Solution retenue :**

Nous avons donc retenu l'approche utilisant la fonction array.filter(), celle-ci étant bien plus rapide que la méthode utilisant la boucle *for* (en moyenne, le temps d’exécution est environ 2 à 3 fois moins long)

**Annexes**



**Figure 1 - Diagramme de la fonction de recherche utilisant une boucle for**

****

Non

**Figure 2 : Approche utilisant la fonction array.filter() de JavaScript**