

Fiche d'investigation de fonctionnalité

Fonctionnalité : Recherche	Fonctionnalité #2
<p>Problématique : L'utilisateur doit pouvoir trier les recettes via un champ de recherche ainsi que 3 menus de sélections : ingrédients, appareils et ustensiles. La recherche doit renvoyer les recettes dont le nom, la description ou les ingrédients contiennent le texte entré et dont les ingrédients, appareils et ustensiles correspondent aux menus de sélections.</p> <p>Nous souhaitons donc un algorithme de recherche le plus fluide, rapide et facilement maintenable.</p>	
<p>Nombre de champs minimum : Aucun (1 champ recherche, 3 menus sélecteurs optionnels).</p> <p>Renvoie toutes les recettes lorsqu'aucun champ n'est rempli.</p>	

<p>Option 1 : Utilisation des boucles natives</p> <p>Dans cette option, on effectue une boucle qui va vérifier chaque recette une par une et comparer leur contenu à la saisie de la recherche. De la même manière, on vérifie chaque élément des menus un par un en comparant leur type et leur nom à ceux présents dans les recettes. L'algorithme renvoie alors toutes les recettes correspondantes.</p>	
<p>Avantages :</p> <ul style="list-style-type: none">• Assez simple à comprendre• Les données sont traitées directement	<p>Inconvénients :</p> <ul style="list-style-type: none">• Moins performant (traitement plus long)• Maintainabilité plus difficile (code plus long et verbeux, peut vite devenir compliqué en cas de modifications ou ajouts à la DB)

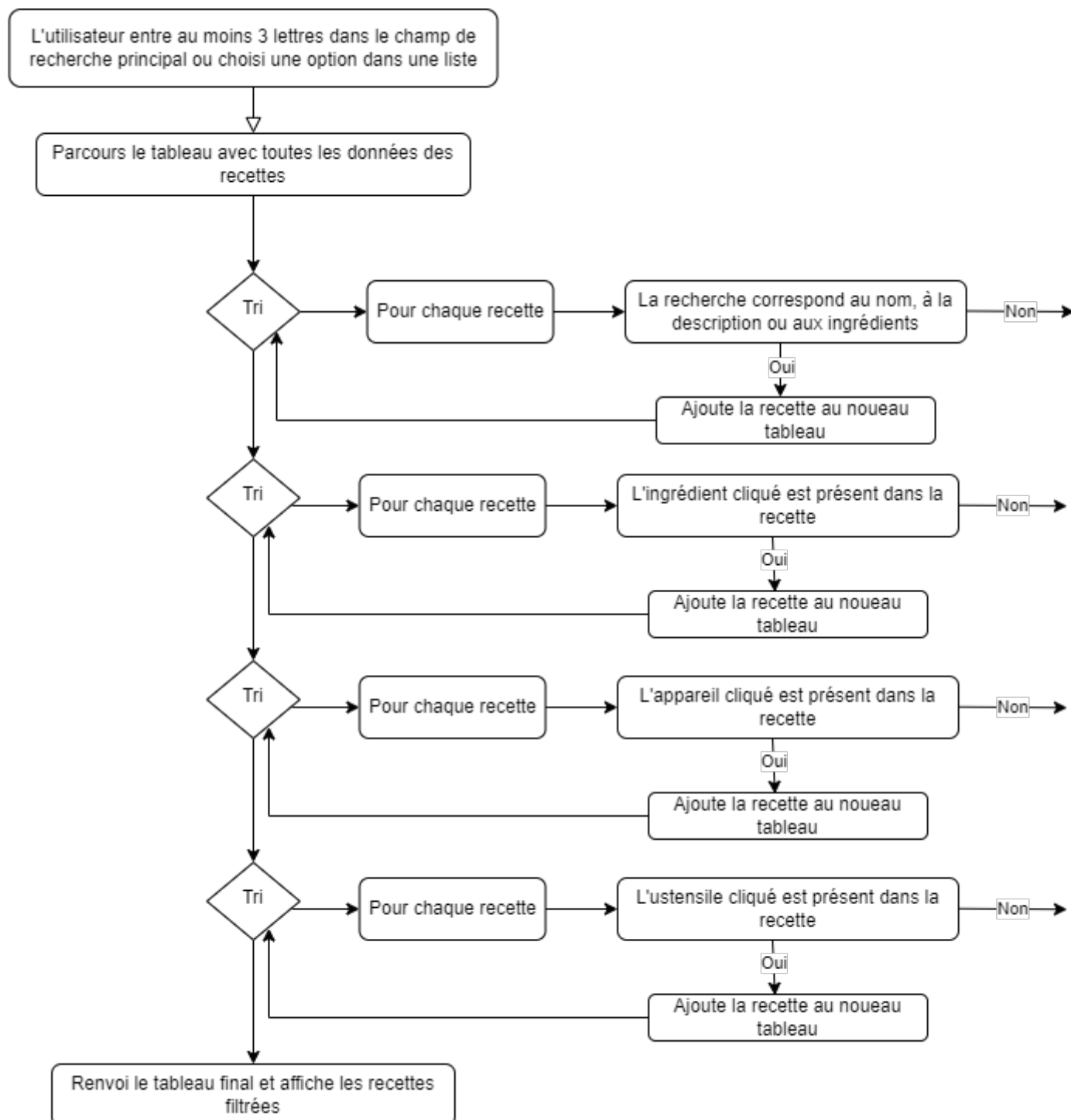
<p>Option 2 : Utilisation des méthodes d'objet array</p> <p>Dans cette option, on utilise les méthodes de l'objet array tels que map() et filter() pour pré-traiter les données des recettes et éléments de sélection. Ainsi on map d'abord les recettes, pour ensuite filtrer les tags et ne retourner que les éléments correspondants.</p>	
<p>Avantages :</p> <ul style="list-style-type: none">• Performant• Maintainable• Code plus compact	<p>Inconvénients :</p> <ul style="list-style-type: none">• Méthodes de filtrage un peu plus compliquées à comprendre, plus verbeuses

<p>Benchmarking tests :</p>	
<p>Option 1 : 106912215.58 ops/s \pm 2.09% 2.56 % slower</p>	<p>Option 2 : 109721787.03 ops/s \pm 3.27% Fastest</p>

<p>Solution retenue : Option 2, par soucis de maintenabilité du code et de performances.</p>

Annexes

Schéma option 1



Annexes

Schéma option 2

