

Fiche d'investigation de fonctionnalité

Fonctionnalité: Recherche à partir de la barre principale	Fonctionnalité#2
<p>Problématique: Développer deux algorithmes de recherche et sélectionner le plus performant. Les fonctions 'annexes' (écoute de la frappe clavier, changement de la visibilité des recettes, rafraîchissement de la page etc.) restent les mêmes, la différenciation de faisant sur la phase de recherche parmi les recettes.</p> <p>La recherche porte sur:</p> <ul style="list-style-type: none">- Les ingrédients des recettes- Le nom des recettes- La description (instructions) des recettes <p>La recherche est lancée dès la première frappe, mais les résultats ne sont affichés qu'une fois que la frappe clavier contient trois caractères. Lorsque cette condition a été remplie une fois dans la session, l'affichage devient alors dynamique et la recherche n'est plus soumise à un nombre de caractères.</p>	

Option 1: Boucles natives (cf. figure 1)

L'algorithme boucle à travers les recettes de la page en suivant des boucles natives (type for (let a=0...)) Pour les ingrédients, un booléen ingredientFound est implémenté à false, et passe à true lorsqu'un ingrédient est trouvé.

Avantages:

- + Plus rapide (2217366.79 ops/s)
- + Fait appel à des notions de code basiques

Inconvénients:

- Plus de lignes de codes: Plus de risques d'erreur, moins lisible et maintenable

Option 2: Boucles forEach et indexOf() (cf. figure 2)

L'algorithme boucle à travers les recettes de la page en suivant des boucles type forEach Pour les ingrédients, on utilise la fonction indexOf() au lieu de boucler parmi tous les ingrédients.

Avantages:

- + Moins de lignes de codes: Moins de risques d'erreur, plus lisible et maintenable

Inconvénients:

- Plus lent (33663.90 ops/s soit 98.48% plus lent)
- Fait appel à des notions de code un peu plus avancées (indexOf() étant un peu moins courant)

Solution retenue:

Nous avons retenu l'option1 car plus performante. (cf. figure3)

Figure 1: Option 1

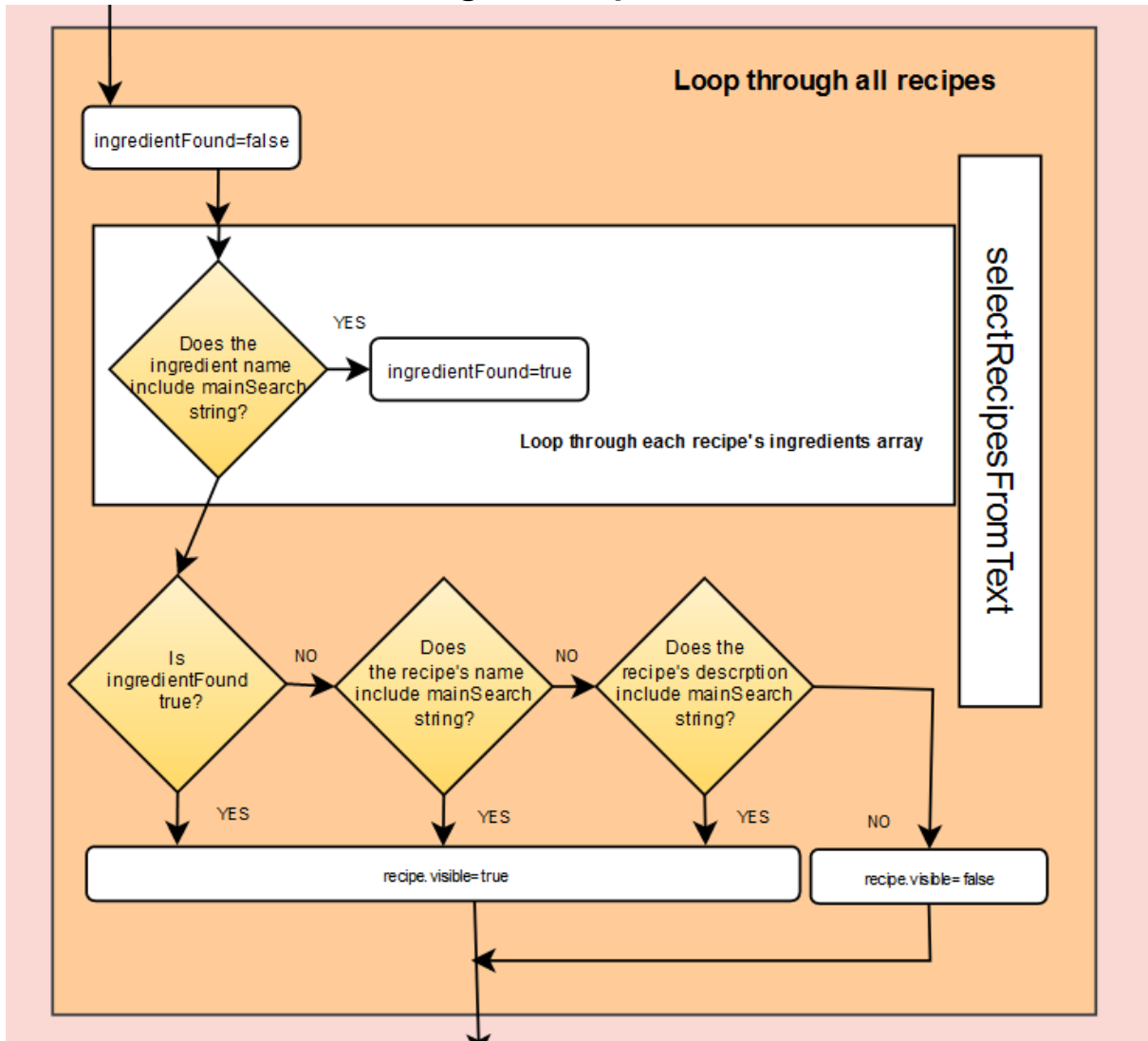


Figure 2: Option 2

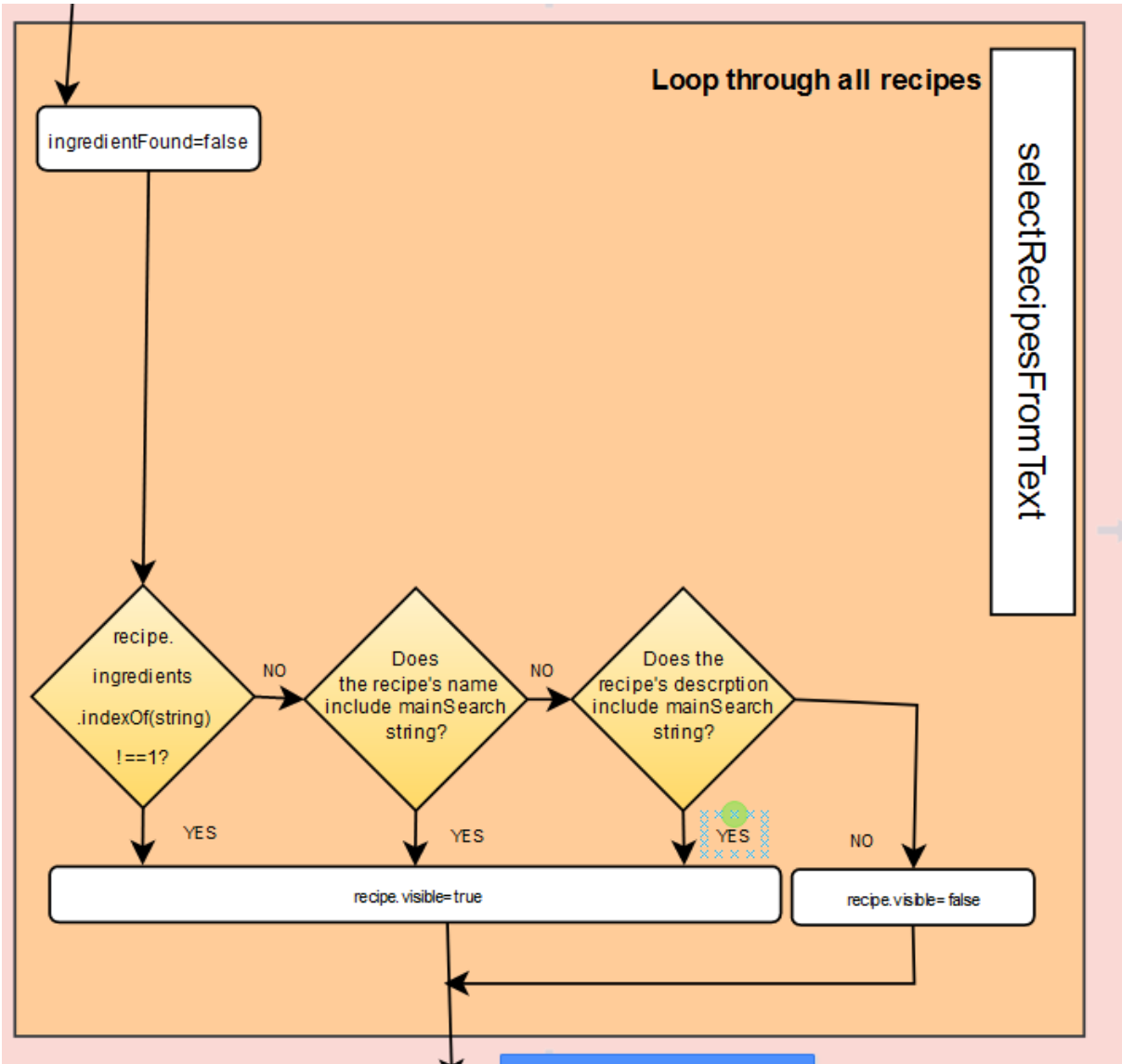


Figure 3: Comparaison des deux options via JSbench

<p>Search1</p> <p>finished</p> <p>2217366.79 ops/s ± 0.93%</p> <p>Fastest</p>	<pre> if (recipes[a].name.toLowerCase().includes(string) === true recipes[a].description.toLowerCase().includes(string) === true ingredientFound === true) { return true; } else { return false; } } </pre>
<p>Search2</p> <p>finished</p> <p>33663.09 ops/s ± 1.3%</p> <p>98.48 % slower</p>	<pre> let string = 'coco'; recipes.forEach((recipe) => { if (recipe.name.toLowerCase().includes(string) === true recipe.description.toLowerCase().includes(string) === true recipe.ingredients.indexOf(string) !== -1) { return true; } else { return false; } }); </pre>