

Fiche d'investigation de fonctionnalité

Fonctionnalité : Recherche principale des recettes

Problématique : Développer l'algorithme de recherche le plus performant

Option 1 : Recherche avec les méthodes de l'objet tableau

Dans cette option, nous avons la possibilité d'effectuer une recherche grâce à des méthodes dédiées à la manipulation des tableaux. Le tableau des recettes est filtré.

Avantages

Filtrages de tableaux facilités grâce aux méthodes

Inconvénients

Nombre d'opérations peu élevé.

Option 2 : Recherche avec les boucles natives de javascript

Dans cette option, la recherche s'effectue en faisant appel aux boucles natives proposées par javascript (for,while). Le tableau des recettes est parcouru et chaque recette correspondant à la recherche est ajouté à un autre tableau.

Avantages

Nombre d'opérations très élevé

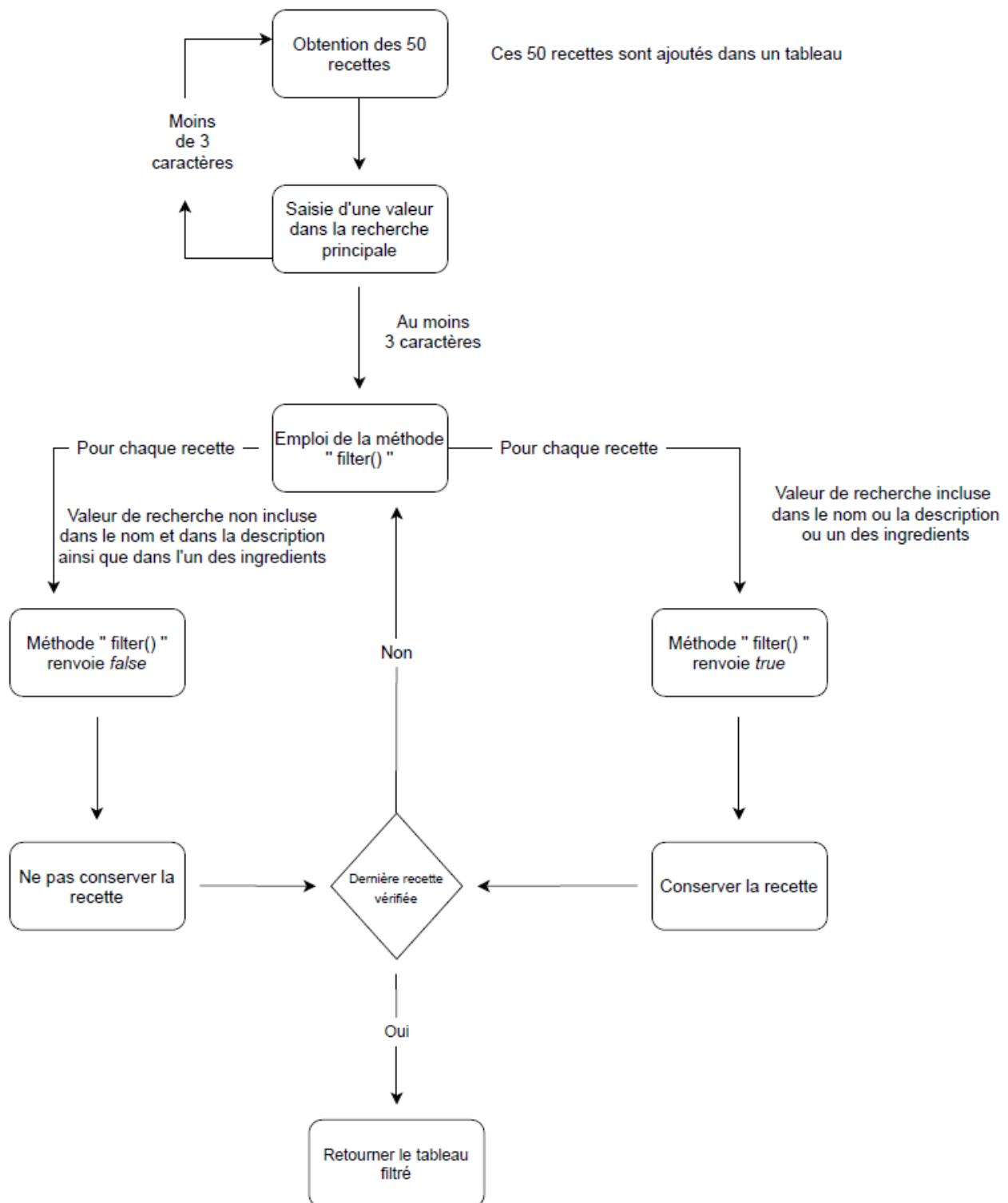
Inconvénients

Utilisation potentielle de plusieurs tableaux

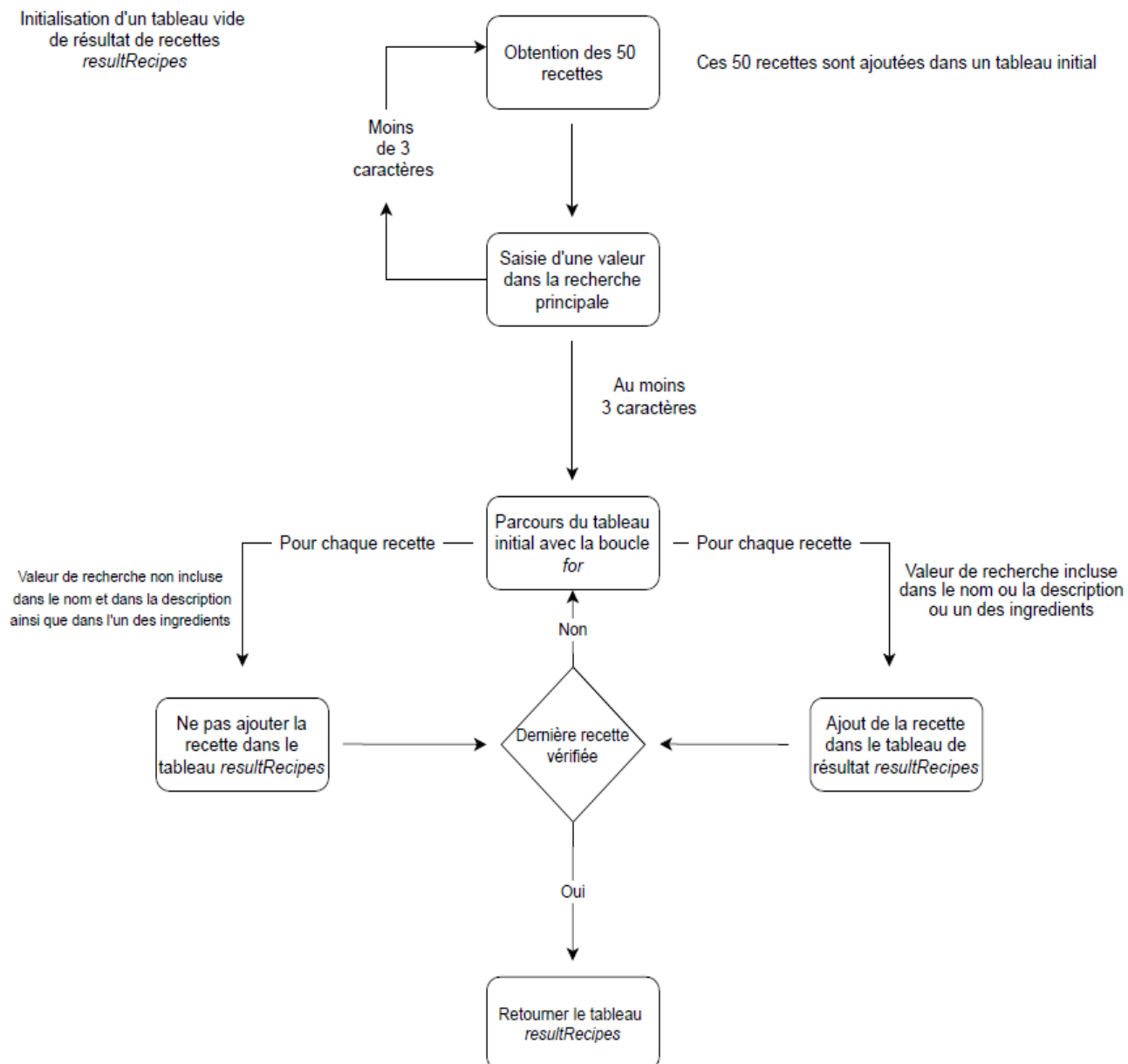
Solution retenue :

La solution retenue est celle des méthodes de l'objet tableau, bien que le nombre d'opérations est supposé être plus faible par rapport aux boucles natives, la performance, selon l'outil jsbench, donne la première option gagnante. (voir en annexe)

Annexes



Option 1 : Recherche avec les méthodes de l'objet tableau




Option 2 : Recherche avec les boucles natives de javascript

Algorithmes et résultats :

code block 1 

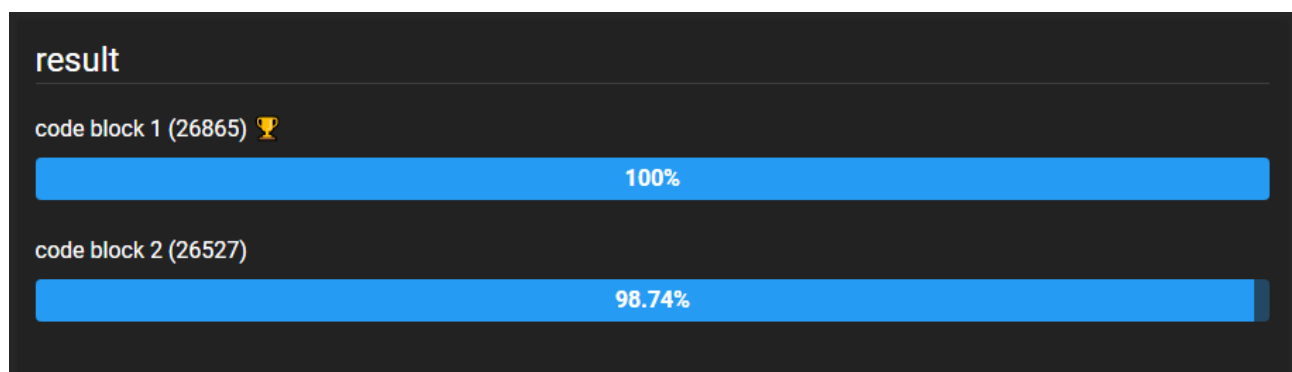
```
1 function mainInputSearch(recipesArray, value) {
2   let filterRecipes = recipesArray.filter((recipe) => {
3     return recipe.name.toLowerCase().includes(value.toLowerCase()) ||
4     recipe.description.toLowerCase().includes(value.toLowerCase()) ||
5     recipe.ingredients.some(ingredient =>
6       ingredient.ingredient.toLowerCase().includes(value.toLowerCase()));
7   });
8   return filterRecipes;
9 }
10
11 mainInputSearch(recipes, search);
12
```

Fonctionnalité de l'option 1 (code block 1)

code block 2 

```
1 function nativeMainSearch(recipesArray, value) {
2   //recherche avec les boucles natives
3   let resultRecipes = [];
4   for (let i = 0; i < recipesArray.length; i++) {
5     let ingredients = recipesArray[i].ingredients;
6     if (recipesArray[i].name.toLowerCase().includes(value.toLowerCase()) ||
7     recipesArray[i].description.toLowerCase().includes(value.toLowerCase())) {
8       resultRecipes.push(recipesArray[i]);
9     } else {
10      for (let j = 0; j < ingredients.length; j++) {
11        if (ingredients[j].ingredient.toLowerCase().includes(value.toLowerCase())) {
12          resultRecipes.push(recipesArray[i]);
13        }
14      }
15    }
16  }
17  return resultRecipes;
18}
```

Fonctionnalité de l'option 2 (code block 2)



Comparaison des algorithmes