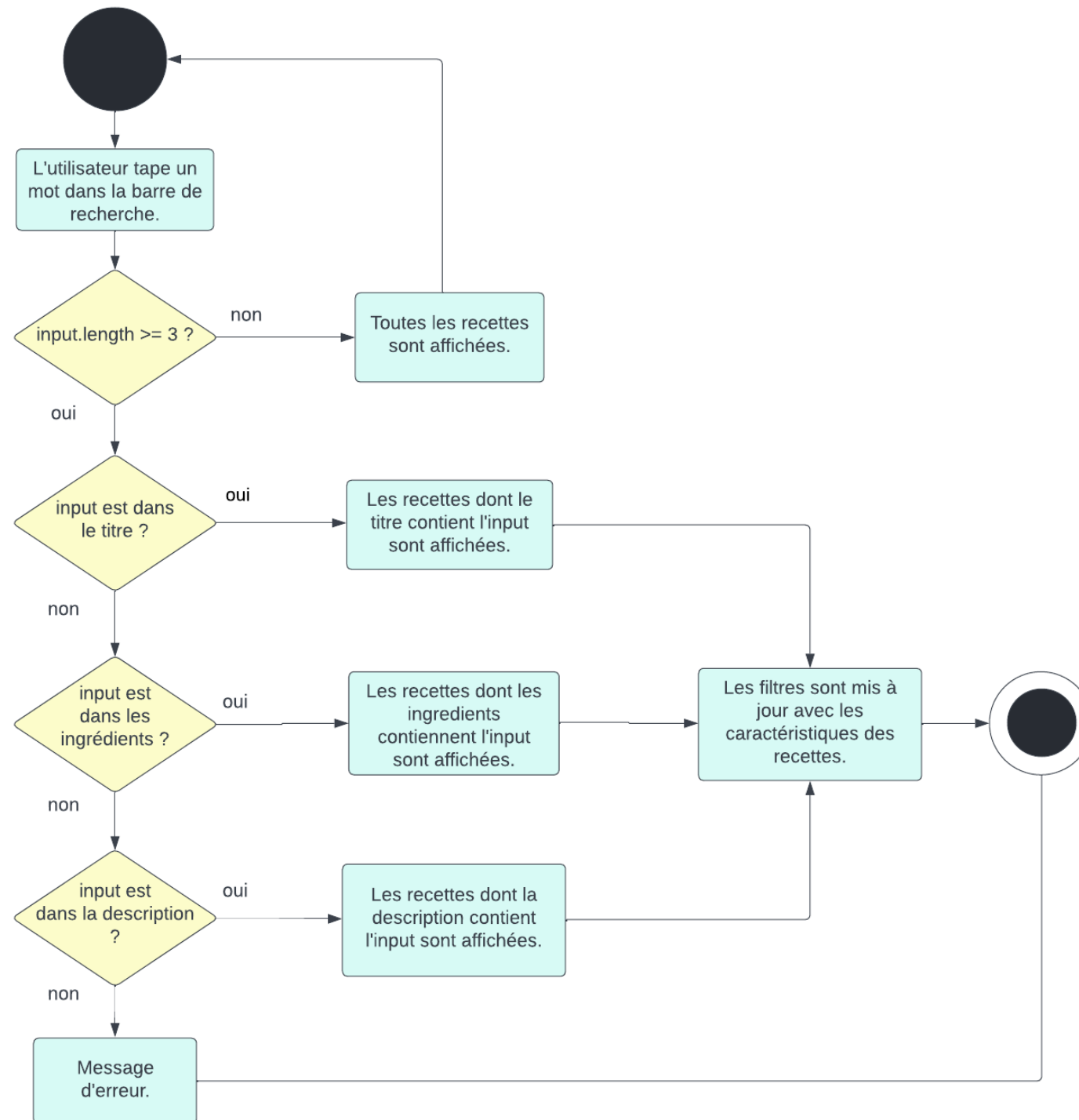


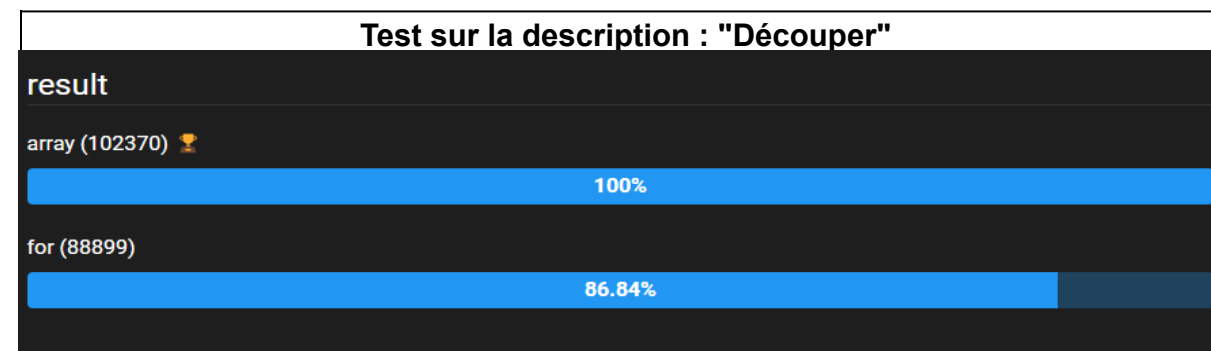
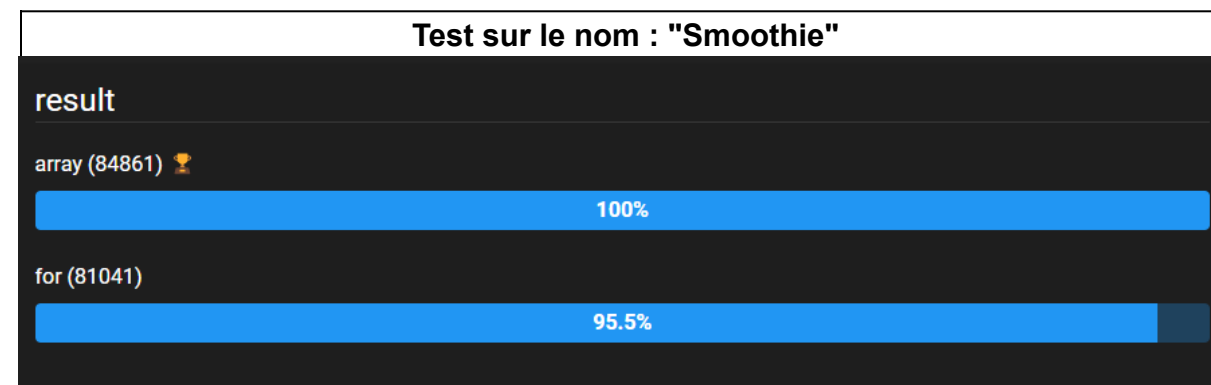
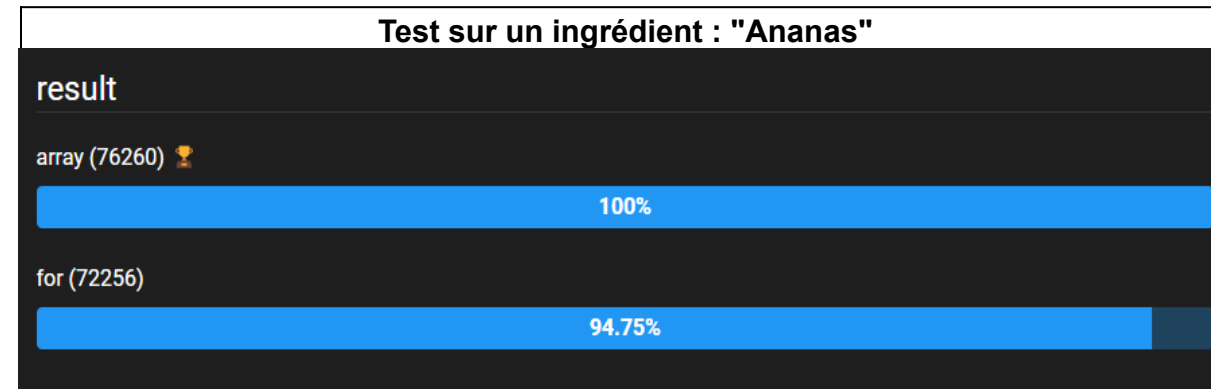
Fiche d'investigation de fonctionnalité

Fonctionnalité : Barre de recherche principale	Fonctionnalité
Problématique : Donner la possibilité à un utilisateur de filtrer les recettes via une barre de recherche; soit par titre, soit par ingrédients ou soit par description. Si aucune recette ne correspond à la recherche, un message d'erreur est retourné	
Option 1 : Méthode native avec des boucles 'for' et des conditions 'if'. Pour cela il faut créer un nouveau tableau, une variable pour les ingrédients (booléen avec une déclaration initiale à false), puis parcourir les ingrédients avec deux boucles 'for'. Si la recherche est concluante, il faut passer la variable à 'true'. Pour le titre et la description juste une condition 'if' est nécessaire. Les recettes trouvées sont ajoutées au nouveau tableau.	
Avantages: <ul style="list-style-type: none">- Code plus facile à lire	Inconvénients : <ul style="list-style-type: none">- Plus de lignes.- Plus de difficile à maintenir.- Plus lent.
Option 2 : Utilisation des méthodes array 'filter' et 'some' Le fichier data est parcouru grâce à la méthodes filter(), la méthode some() permet de retourner les ingrédients qui sont contenus dans le recherche.	
Avantages: <ul style="list-style-type: none">- Moins de ligne.- Plus facile à maintenir- Plus rapide	Inconvénients : <ul style="list-style-type: none">- Fonctions plutôt récentes

Algorithme



Test de performance



Test global

array method	<pre>//Searchbar function const mainSearch = () => { let filteredRecipes = recipes; // Get value of searchbar input and put it in lower case const result = document.getElementById("main__search").value.toLowerCase(); // Filter recipes by tags document.querySelectorAll("#resume__filter a").forEach((tag) => { const tagValue = tag.textContent.toLocaleLowerCase(); filteredRecipes = filteredRecipes.filter((recipe) => recipe.ingredients.some((item) => item.ingredient.toLowerCase().includes(tagValue)) recipe.appliance.toLowerCase().includes(tagValue) recipe.ustensils.some((ustensil) => ustensil.toLowerCase().includes(tagValue))); }); }</pre>
finished	
1990681132.65 ops/s ± 1.17%	
Fastest	
for loop	<pre>//Searchbar function const mainSearch = () => { let filteredRecipes = recipes; // Get value of searchbar input and put it in lower case const result = document.getElementById("main__search").value.toLowerCase(); // Filter recipes by tags document.querySelectorAll("#resume__filter a").forEach((tag) => { const tagValue = tag.textContent.toLocaleLowerCase(); filteredRecipes = filteredRecipes.filter((food) => food.ingredients.some((item) => item.ingredient.toLowerCase().includes(tagValue)) food.appliance.toLowerCase().includes(tagValue) food.ustensils.some((ustensil) => ustensil.toLowerCase().includes(tagValue))); }); }</pre>
finished	
1751137017.82 ops/s ± 3.84%	
12.03 % slower	

Conclusion

La méthode "array" utilisant les méthodes filter() et some() est plus performante d'environ 12%. Elle nécessite aussi moins de ligne de code et est plus facile à maintenir.

La méthode "array" sera donc utilisée pour la recherche principale du site "Les petits plats".