

Fiche d'investigation de fonctionnalité

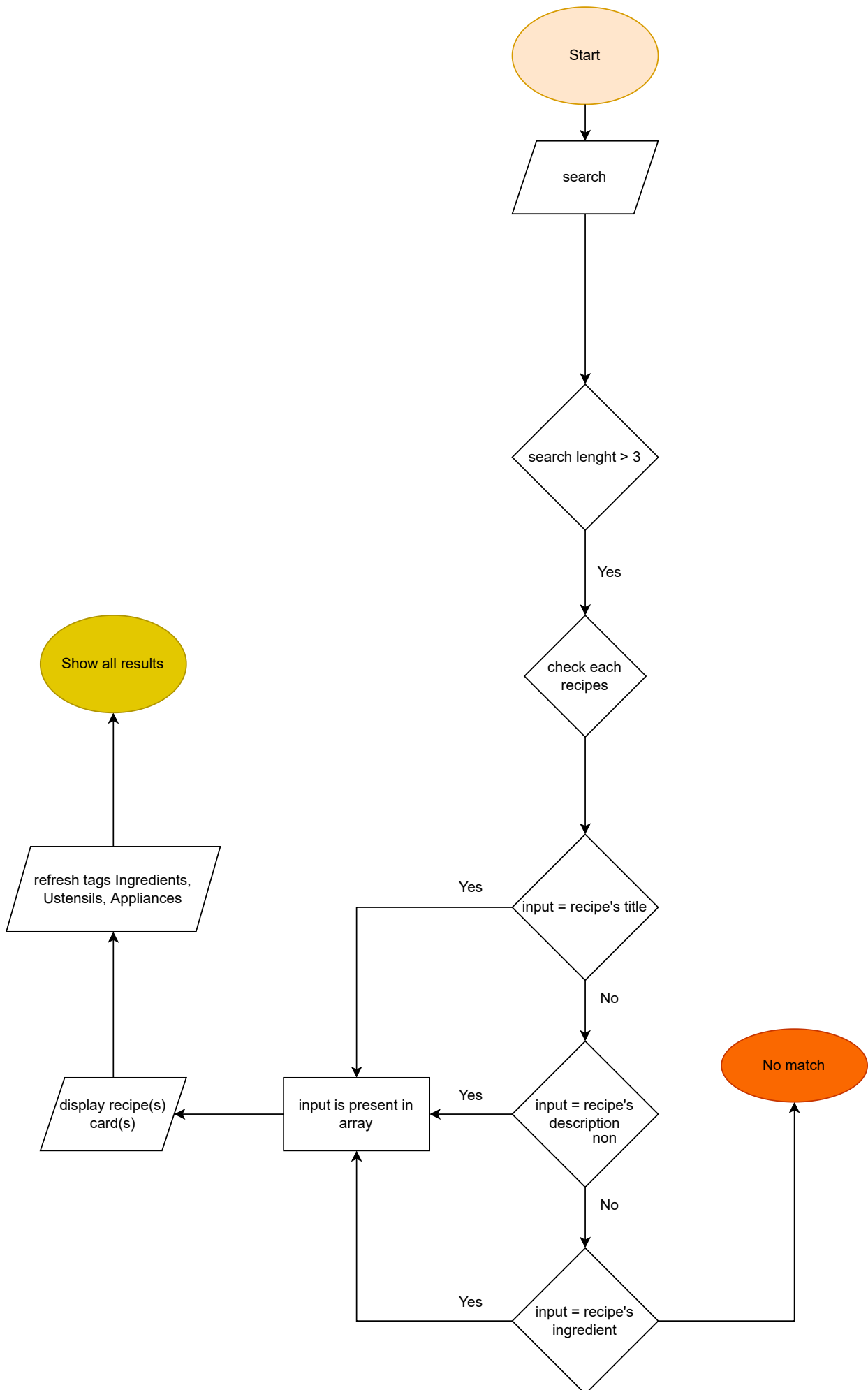
Fonctionnalité : Recherche de recettes	Fonctionnalité #1
Problématique : Filtre pour accéder rapidement à une recette correspondant à un besoin de l'utilisateur dans les recettes déjà reçues	

<u>Option 1 : Programmation fonctionnelle (annexe 1)</u> Utilisation des méthodes de l'objet Array (forEach, Filter...) Emploi ici de la méthode « filter » qui filtre les recettes suivant la saisie effectuée et les correspondances trouvées dans le nom ou la description ou les ingrédients de la recette. La recette trouvée est ajoutée à un tableau qui servira à l'affichage des recettes. De ce tableau, les différentes listes sont mises à jour.	
Avantages + code plus robuste et plus stable + code plus court + version plus rapide	Inconvénients - code moins lisible
Saisie de 3 caractères minimum dans le champ de recherche principal	

<u>Option 2 : Programmation native (annexe 1)</u> Utilisation des boucles (while, for ...). Ici utilisation de « for » qui itère sur le tableau des recettes et cherche s'il existe une correspondance entre la saisie, et le nom ou la description ou un des ingrédients de la recette. Si oui, la recette en question est ajoutée à un nouveau tableau qui servira à l'affichage des recettes trouvées. De ce tableau, les différentes listes sont mises à jour également	
Avantages - code plus lisible, plus facile à comprendre	Inconvénients - code moins stable, plus long - version plus lente
Saisie de 3 caractères minimum dans le champ de recherche principal	

Solution retenue :

La solution retenue est l'option 2 car 1.63% plus rapide que l'option 1 donc plus efficace. (test fait sur jsbench.me annexe 2)



Annexe 2

<div>Algo for</div> <div>finished</div> <div>1187765.13 ops/s \pm 0.88% 1.63 % slower</div>	<pre>let currentTabRecipes =recipes; filteredRecipes =recipes; searchedRecipes = recipes; const filtersDatas = Array.from(document.querySelectorAll(".tag button")); function searchBarAlgo () { const inputData = document.querySelector("#searchinput").value.toLowerCase(); if (inputData.length > 2) { let result = []; for (const element of filteredRecipes) { const match = inputMatch (inputData, element); if (match == true) { result.push(element); } } } }</pre>
<div>Algo filter</div> <div>finished</div> <div>1207430.39 ops/s \pm 0.47% Fastest</div>	<pre>let currentTabRecipes =recipes; filteredRecipes =recipes; searchedRecipes = recipes; const filtersDatas = Array.from(document.querySelectorAll(".tag button")); function searchBarAlgo() { const inputData = document.querySelector("#searchinput").value.toLowerCase(); if (inputData.length > 2) { searchedRecipes = filteredRecipes.filter(element => { const match = inputMatch (inputData, element); if (match == true) { return element; } }); } }</pre>
<div>Algo for</div> <div>finished</div> <div>1187765.13 ops/s \pm 0.88% 1.63 % slower</div>	
<div>Algo filter</div> <div>finished</div> <div>1207430.39 ops/s \pm 0.47% Fastest</div>	