

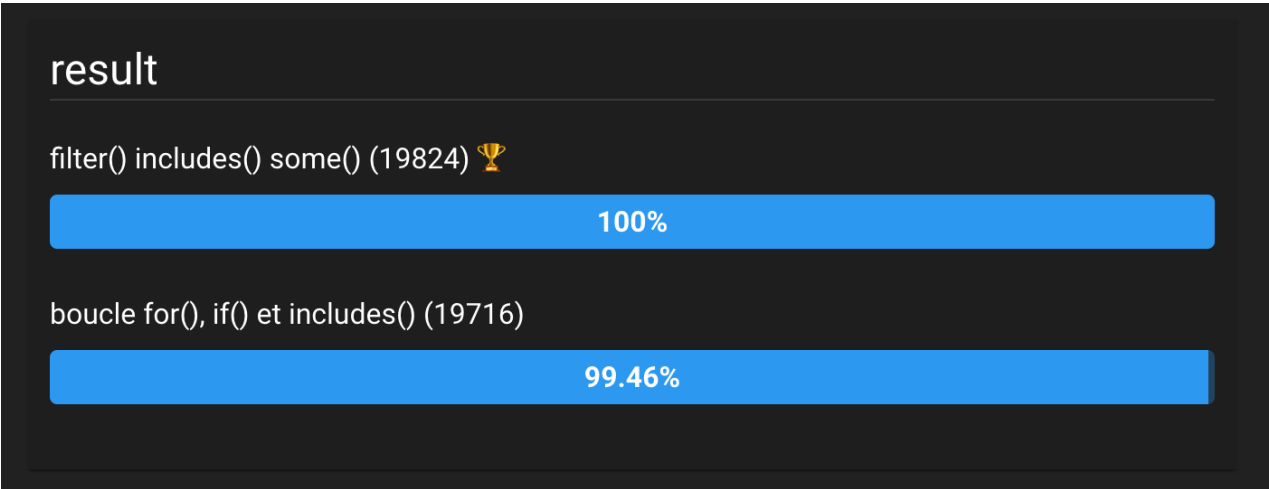
Fiche d'investigation de fonctionnalité

Fonctionnalité : Algorithme de recherche des recettes	Fonctionnalité #1
Problématique : Obtenir une solution performante et rapide lors de la recherche de recettes de cuisine parmi notre base de données.	

Option 1 : Array.filter() et Array.includes() Dans cette option, nous utilisons les fonctions associées au prototype Array afin de retourner directement un tableau avec les éléments correspondant à nos conditions .L'avantage du système c'est qu'il est plus rapide et concis à implémenter/maintenir.	
Avantages <ul style="list-style-type: none"> ⊕ Code concis ⊕ Performance rapide 	Inconvénients <ul style="list-style-type: none"> ⊖ Avec plusieurs fonctions imbriquées, il peut être difficile de s'y retrouver
Nombre de caractères minimum à renseigner : 3	

Option 2 : Boucle For et Array.includes() Dans cette option, nous utilisons une boucle for afin de parcourir chaque recette et vérifier pour chacune d'elles si le terme recherché est présent dans les données à disposition. Si c'est le cas, nous ajoutons la recette à un nouveau tableau qui sera renvoyé à la fin du parcours de la boucle.	
Avantages <ul style="list-style-type: none"> ⊕ Plus facile à comprendre pour qui ne connaît pas la fonction Array.filter() ⊕ Performance rapide 	Inconvénients <ul style="list-style-type: none"> ⊖ Code plus long ⊖ Utilisation d'un tableau supplémentaire, donc plus de ressources utilisée.
Nombre de caractères minimum à renseigner: 3	

Solution retenue: Suite aux tests des deux algorithmes effectués sur la plateforme JSBEN.ch, les 2 solutions semblent similaires. Très peu de différences sont remarquées et selon les sessions de tests, la solution 1 peut être plus performante tout comme cela peut être la solution 2 à la session suivante. Nous choisissons donc la solution 1 avec les fonctions du prototype Array afin de faciliter la lisibilité et la maintenance du code. https://jsben.ch/EmEzI
--



The screenshot shows a benchmark titled 'result' comparing two JavaScript solutions. The first solution, 'filter() includes() some() (19824)', is marked with a trophy icon and shows a 100% completion bar. The second solution, 'boucle for(), if() et includes() (19716)', shows a 99.46% completion bar.

Solution	Score	Performance
filter() includes() some() (19824) 🏆	19824	100%
boucle for(), if() et includes() (19716)	19716	99.46%



Annexes

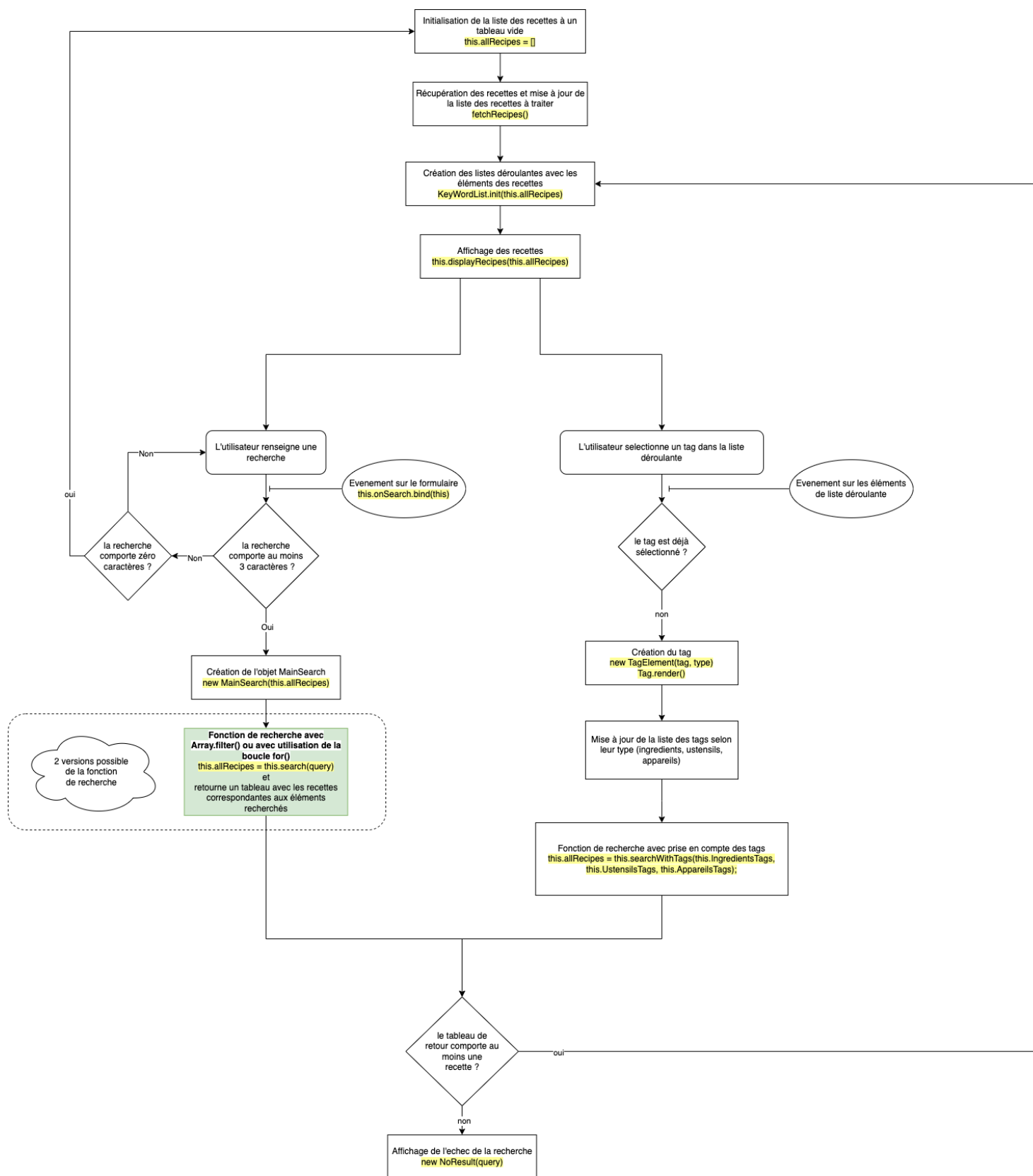


Figure 1 - Diagramme d'activité