

Fiche d'investigation de fonctionnalité

Fonctionnalité: Search	
Problématique: Dans le but d'une recherche fluide et dans un soucis d'optimisation, vous voulez un algorithme qui soit le plus performant possible.	

Option 1: Algorithme de recherche utilisant la programmation fonctionnelle Dans ce cas, l'algorithme de recherche va se baser sur les éléments de la méthode array(filter, reduce, map etc...) Le principal avantage à ces méthodes sont leurs maintenabilité car facile à lire. De l'autre côté l'inconvénient principal reste leur performance car plus d'étapes sont faites pour renvoyer une réponse.	
Avantages : <ul style="list-style-type: none">• Facile à mettre en œuvre• Maintenance	Inconvénients: <ul style="list-style-type: none">• Performances
Performances lors du jsBench : 1972.77 ops	

Option 2 : Algorithme de recherche utilisant les boucles natives Dans ce cas, l'algorithme de recherche va utiliser des boucles natives(for, while). L'avantage principal sera les performances. En revanche, moins simple à mettre en œuvre car plus de paramètres à entrer.	
Avantages: <ul style="list-style-type: none">• Performances• Meilleurs contrôles des itérations	Inconvénients: <ul style="list-style-type: none">• Mise en œuvre• Maintenance (lisibilité)
Performances lors du jsBench : 2657.78 ops	

Solution retenue : Malgré les avantages en termes de maintenance de la première option, la solution retenue sera: "l'algorithme de recherche utilisant les boucles natives" car l'objectif principal restant les performances, cette solution permet une meilleure optimisation.

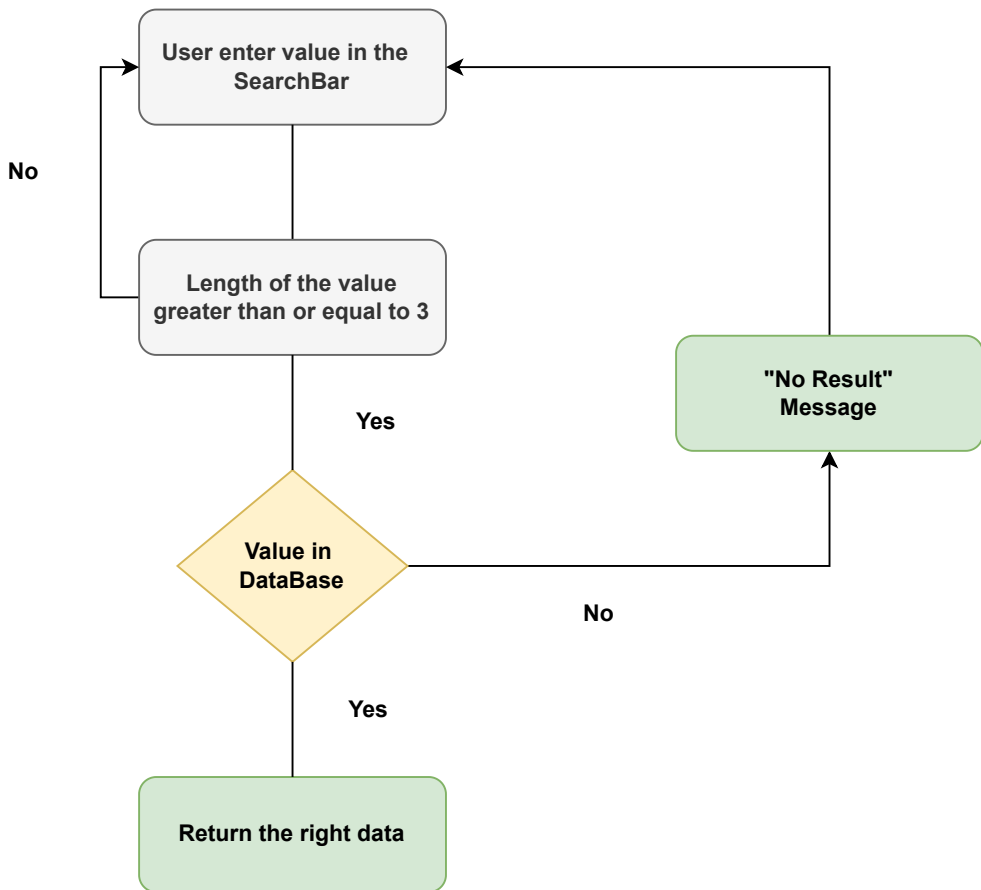


Figure 1 - Diagramme des deux algorithmes