

Fiche d'investigation de fonctionnalité

Fonctionnalité : Recherche principale

Problématique: Afin d'offrir aux utilisateurs une recherche la plus rapide possible, nous allons en écrire deux versions. Dans un second temps, nous allons utiliser une application web Jsben.ch, qui va éprouver ces deux versions l'une envers l'autre et sélectionner la plus rapide.

Option 1 : Utiliser la fonction native filter()

Avantage:

Il y a une lecture simple du code.

Inconvénient:

Cette version par cette fonction native est censée être la plus rapide, or ce n'est pas le cas.

Option 2 : Utiliser une boucle for()

Avantage:

D'un premier regard, je pense que mettre les comparaisons dans un if(), doit faire gagner du temps.

N'ayant pas encore toute la compréhension de ce phénomène, j'écoute le résultat de Jsben.ch pour choisir cet algorithme, qui est plus rapide.

Inconvénient:

Lisibilité du code un peu moins claire.

Solution retenue :

Sur 50 recettes, la différence entre les deux options ne se voit pas en termes d'ergonomie pour l'utilisateur.

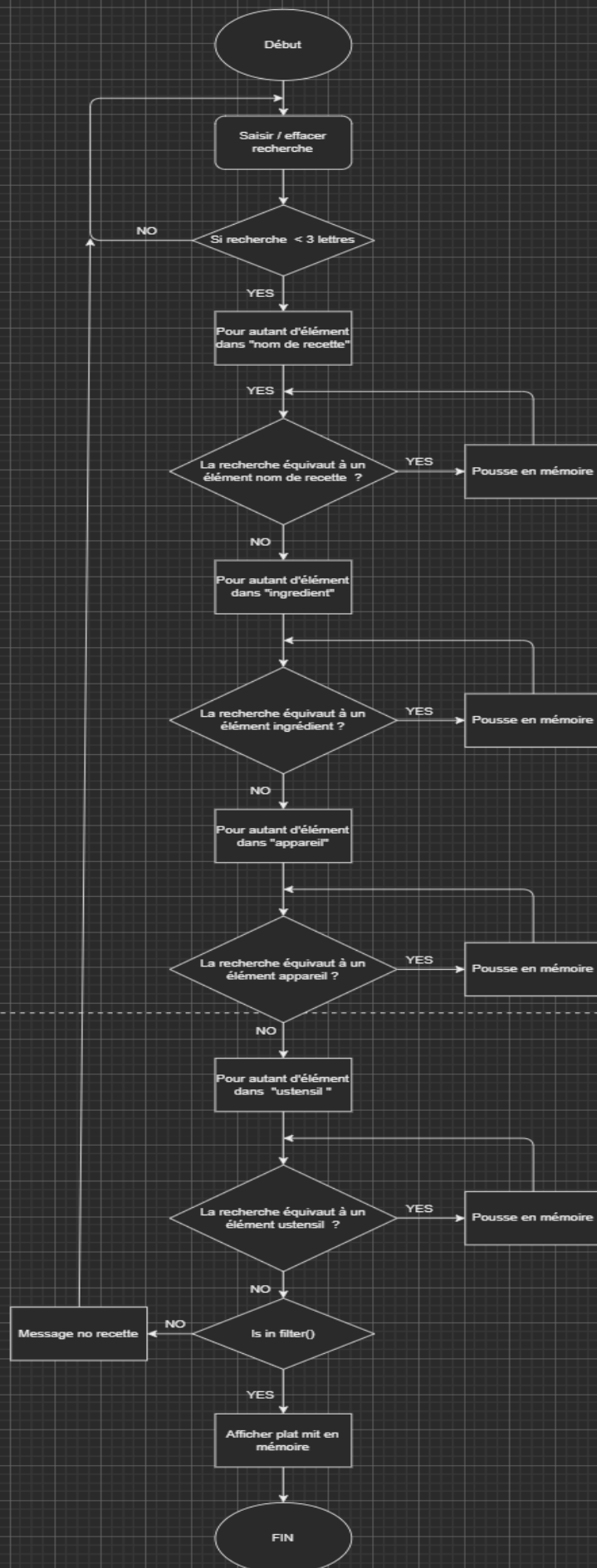
Par contre, pour savoir ce qu'il va se passer lorsque cette base de données va évoluer, il est bon de savoir qu'elle fonction est la plus rapide, grâce à des outils de comparaison.

Les concepteurs du langage Javascript cherchent à ce que leurs fonctions natives soient les plus performantes possibles, mais leur utilisation peut être plus ou moins bien utilisée.

C'est pour cela que des outils de comparaison de code tels que **jsben** existent.

Ci-joint le lien du benchmark : <https://jsben.ch/LBZTo>

Algorithme de recherche avec fonction native filter()



Algorithme de recherche avec boucle for

