

ALGORITHME 01

Fiche d'investigation
concernant la
fonctionnalité de
recherche principale

Réception de la valeur entrée par l'utilisateur
dans l'input principal

Si la valeur
entrée possède plus de
3 caractères

Oui

La valeur est
normalisée pour
faciliter la recherche

Pour chaque recette dans le fichier
"recipes.json", l'algorithme vérifie si la valeur
entrée est présente dans :
- le nom de la recette
- la description de la recette
- les ingrédients de la recette

Si des recettes
incluent la valeur entrée
dans l'input

Non

Affichage du
message d'erreur

Oui

Les recettes en
question sont
ajoutées au tableau
"results"

Les résultats sont
affichés dans la
section "main" de la
structure html

Affichage de
toutes
les recettes

Non

Oui

L'utilisateur
efface la requête

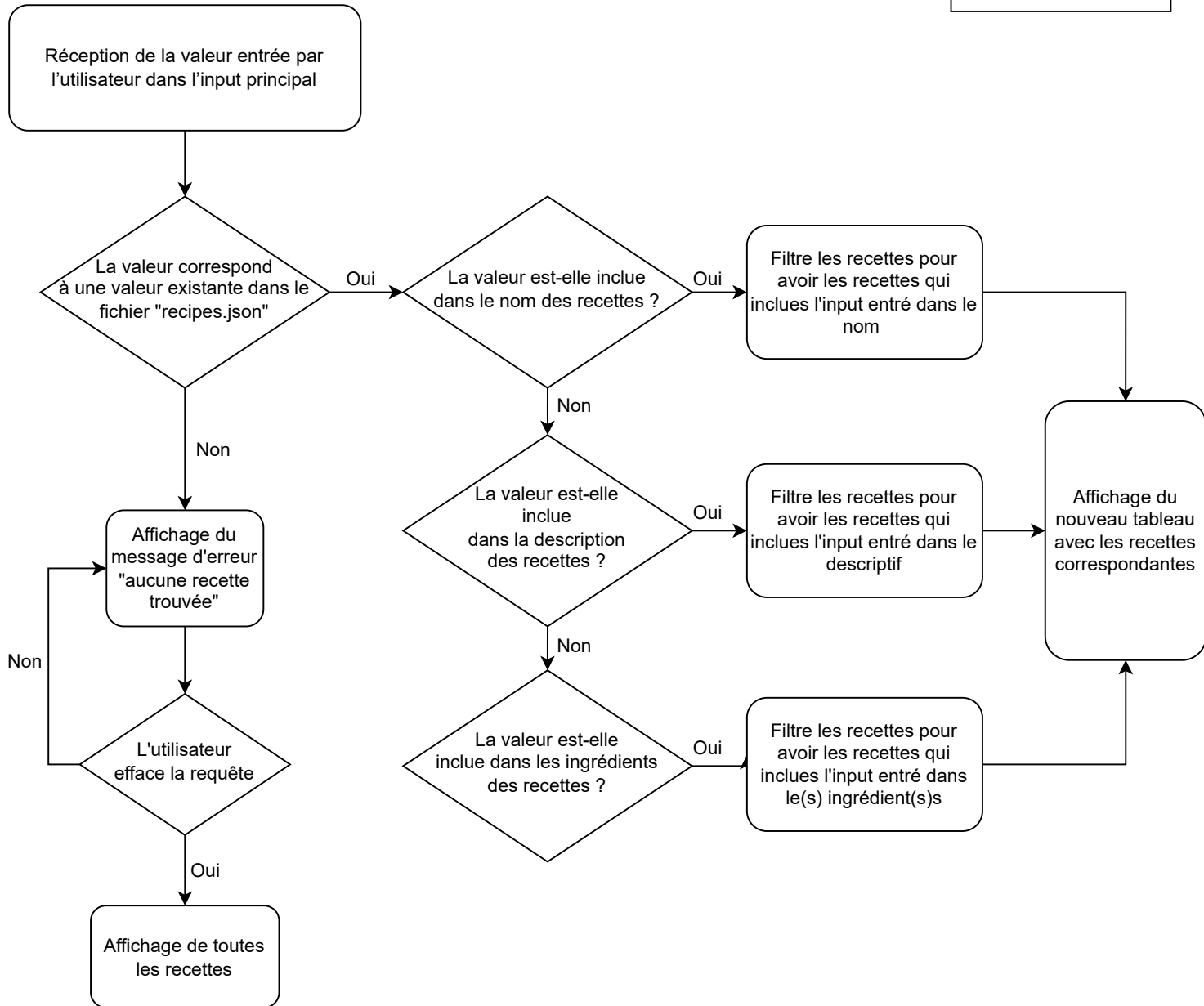
Non

Légende

Utilisation d'une boucle for sur l'ensemble des recettes.
Création d'un tableau vide et utilisation de la méthode push si la valeur entrée dans l'input est présente dans le nom de la recette, la description de la recette ou bien dans les ingrédients de la recette.

ALGORITHME 02

Fiche d'investigation
concernant la
fonctionnalité de
recherche principale



Légende

Utilisation de la méthode filter() pour créer un nouveau tableau si la valeur de la recherche correspond à une valeur liée aux ingrédients, appareils ou ustensiles d'une recette.

Fonctionnalité étudiée : barre de recherche principale.

Dans les deux cas, nombre de caractères minimum pour déclencher la recherche : 3.

Algorithme 01

Avantages :

L'utilisation de la boucle sur l'ensemble des recettes pour voir s'il y a une similarité parmi celles-ci avec la valeur entrée en input permet d'avoir un code plus lisible et logique.

Inconvénients :

Néanmoins, cette méthode est plus longue et demande plus de calculs.

Algorithme 02

Avantages :

L'utilisation de la méthode `filter()` permet d'utiliser une approche existante pour générer un nouveau tableau de recettes avec les résultats correspondants à l'input entré. S'agissant d'une méthode "officielle", il suffit alors de l'appeler et la paramétrer.

De plus, nous sommes davantage certain que cette méthode est compatible avec la plupart des navigateurs.

Inconvénients :

Moins compréhensible au premier abord.

JSBENCH.CH

D'après, le jsbench, l'algorithme le plus efficace est l'algorithme 1
Voir résultat ici : <https://jsben.ch/pgxQM>