



Les petits plats

Investigation de fonctionnalité

Règles de gestion

Ces points doivent absolument être respectés durant le développement :

- 1. La recherche doit pouvoir se faire via le champ principal ou via les tags (ingrédients, ustensiles ou appareil)
- 2. La recherche principale se lance à partir de 3 caractères entrés par l'utilisateur dans la barre de recherche
- 3. La recherche s'actualise pour chaque nouveau caractère entré
- 4. La recherche principale affiche les premiers résultats le plus rapidement possible
- 5. Les champs ingrédients, ustensiles et appareil de la recherche avancée proposent seulement les éléments restant dans les recettes présentes sur la page
- 6. Les retours de recherche doivent être une intersection des résultats. Si l'on ajoute les tags "coco" et "chocolat" dans les ingrédients, on doit récupérer les recettes qui ont à la fois de la coco et du chocolat.
- 7. Comme pour le reste du site, le code HTML et CSS pour l'interface (avec ou sans Bootstrap) devra passer avec succès le validateur W3C.
- 8. Aucune librairie ne sera utilisée pour le JavaScript du moteur de recherche

Fonctionnalité : recherche

Problématique : Trier parmi les recettes disponibles, via un champ de recherche et des menus sélectifs (ingrédients, appareils, ustensiles),

La recherche retournera les recettes dont le nom, la description ou les ingrédients contiennent la recherche effectuée. Et dont l'appareil, les ingrédients et les ustensiles correspondent aux tags sélectionnés.

Afin de pouvoir offrir aux utilisateurs une recherche pertinente et rapide.

Nombre de champ : 1 (optionnel)

Nombre de sélecteur : 3 (optionnels)

Nombre minimum : 0 (liste complète des recettes)

OPTION 1: Tableau pré-traité et filtré

Dans cette solution, les données sont traitées et filtrées en amont pour faciliter le travail de l'algorithme, par une recherche de mots. Le tableau est filtré ensuite à travers plusieurs fonctions, afin de trouver une correspondance entre la recherche utilisateur (mots + tags) et le tableau des données préalablement effectué.

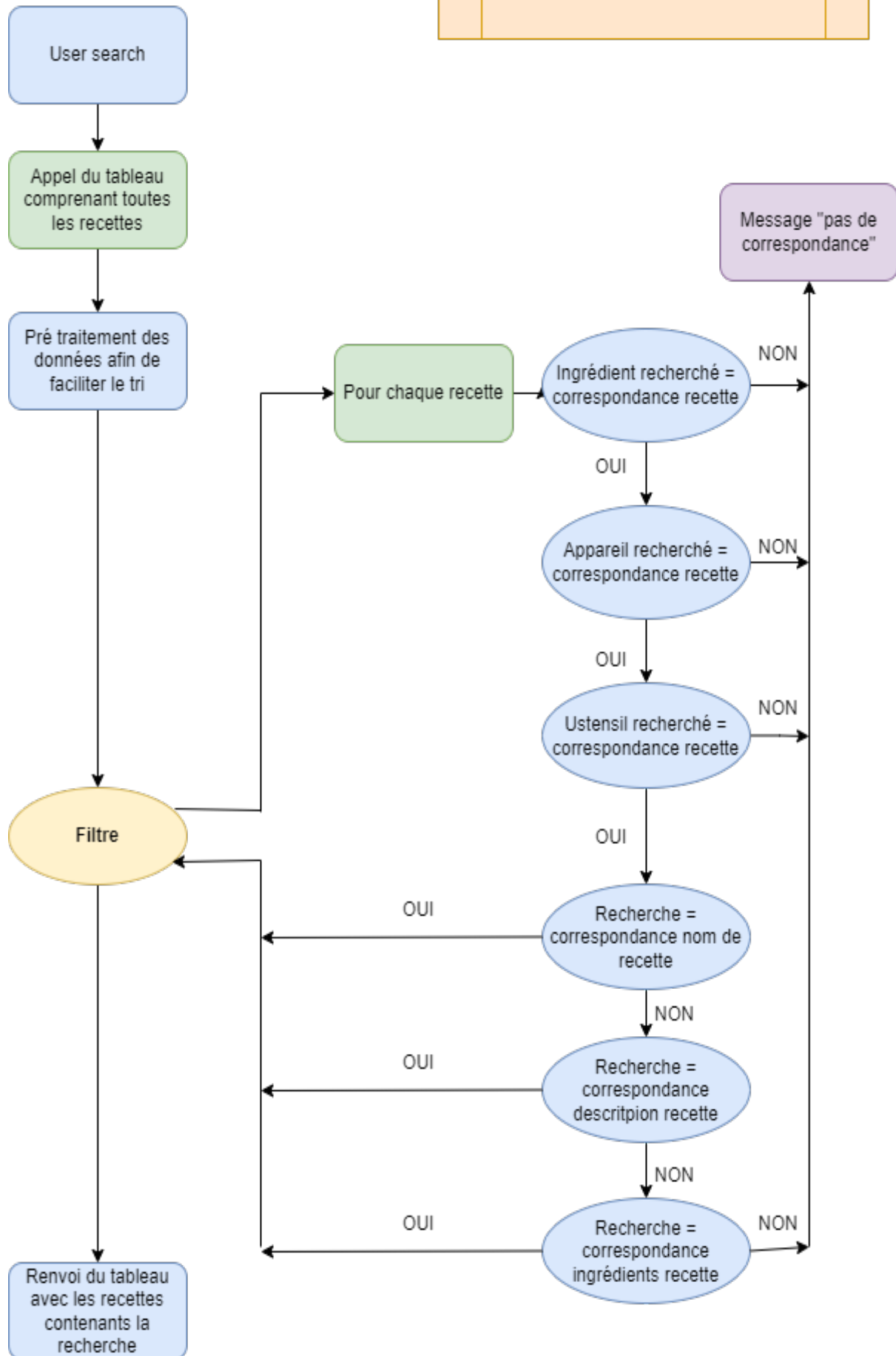
Avantages :

- Performance
- Fonctions simples
- Extraction

Inconvénients :

- Prétraitement un peu long
- fonction binaire

Algorithme 1



OPTION 2 : Tri « étape par étape »

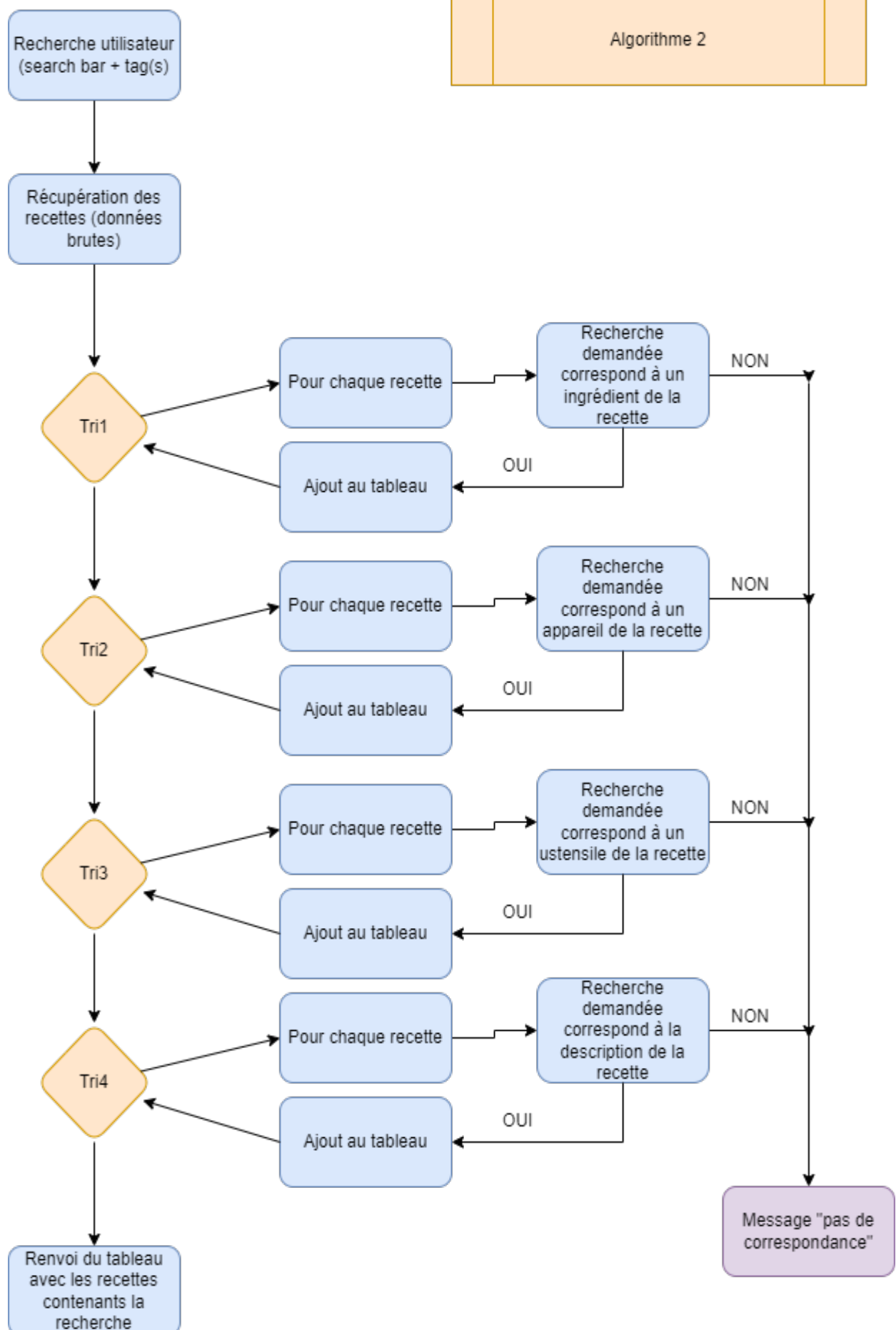
Dans cette approche, les données sont récupérées à chaque requête (ingrédient / appareil / ustensile / description), puis renvoyées à chaque fin de tri dans un nouveau tableau, afin de renvoyer un tableau final de recette correspondante(s) à la recherche.

Avantages :

- Simplicité
- Données brutes

Inconvénients :

- Peu performant (rébarbatif)
- Complication si changement de BDD
- analyse ligne par ligne



Solution retenue : Option 1 (performance, cf test)

Pour sa rapidité d'exécution, l'algorithme n°1 sera donc notre choix

TESTS

Test de rapidité effectué sur jsbench.me

Il en ressort que l'algorithme n°1 est plus rapide et performant que l'algorithme n°2 (entre 16 % et 25 % selon les tests effectués)

Setup HTML	<pre><!DOCTYPE html> <html lang="en"> <head> <link rel="shortcut icon" href="#" /> <meta charset="UTF-8" /> <meta http-equiv="X-UA-Compatible" content="IE=edge" /> <meta name="viewport" content="width=device-width, initial-scale=1.0" /> <link rel="stylesheet" href="./style/style.css" /> <title>Les Petits Plats</title> </head></pre>
Setup JS - click to add setup JavaScript	
algo 1	<pre>] }</pre>
finished	<pre>let recipesArray = Object.entries(recipes); //f Create element const create = (elm, attributes) => { const element = document.createElement(elm);</pre>
55.51 ops/s ± 9.24%	
Fastest	
algo 2	<pre>] }</pre>
finished	<pre>let recipesArray = Object.entries(recipes); //F create element const create = (elm, attributes) => { const element = document.createElement(elm); for (let key in attributes) {</pre>
46.2 ops/s ± 4.36%	
16.77 % slower	