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.

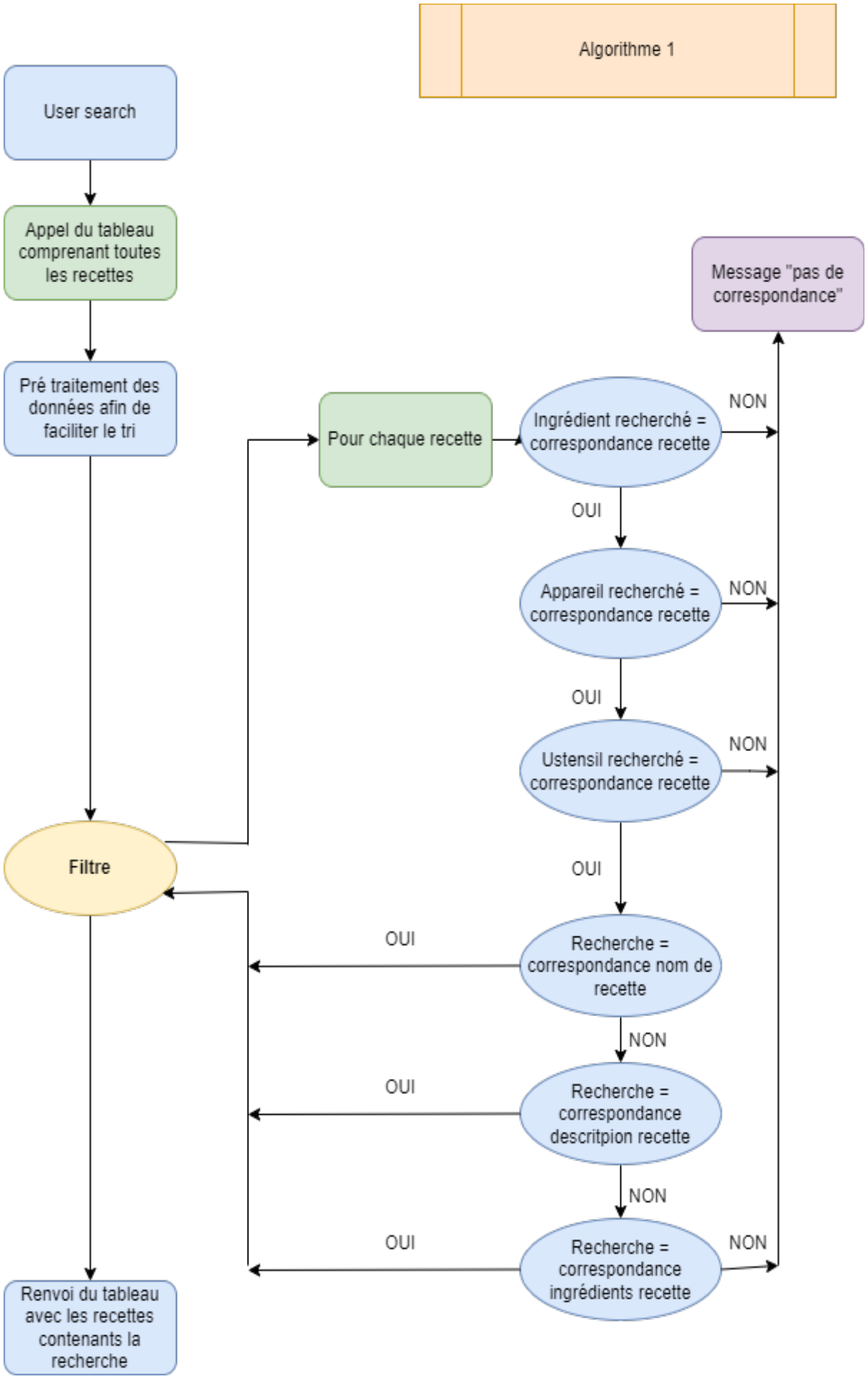
Nombre de champ : 1 (optionnel)
Nombre de sélecteur : 3 (optionnels)
Nombre minimum : 0 (liste complète des recettes)

<u>OPTION 1</u> : **Tableau pré-traité et filtré**

Dans cette solution, les données sont traités 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

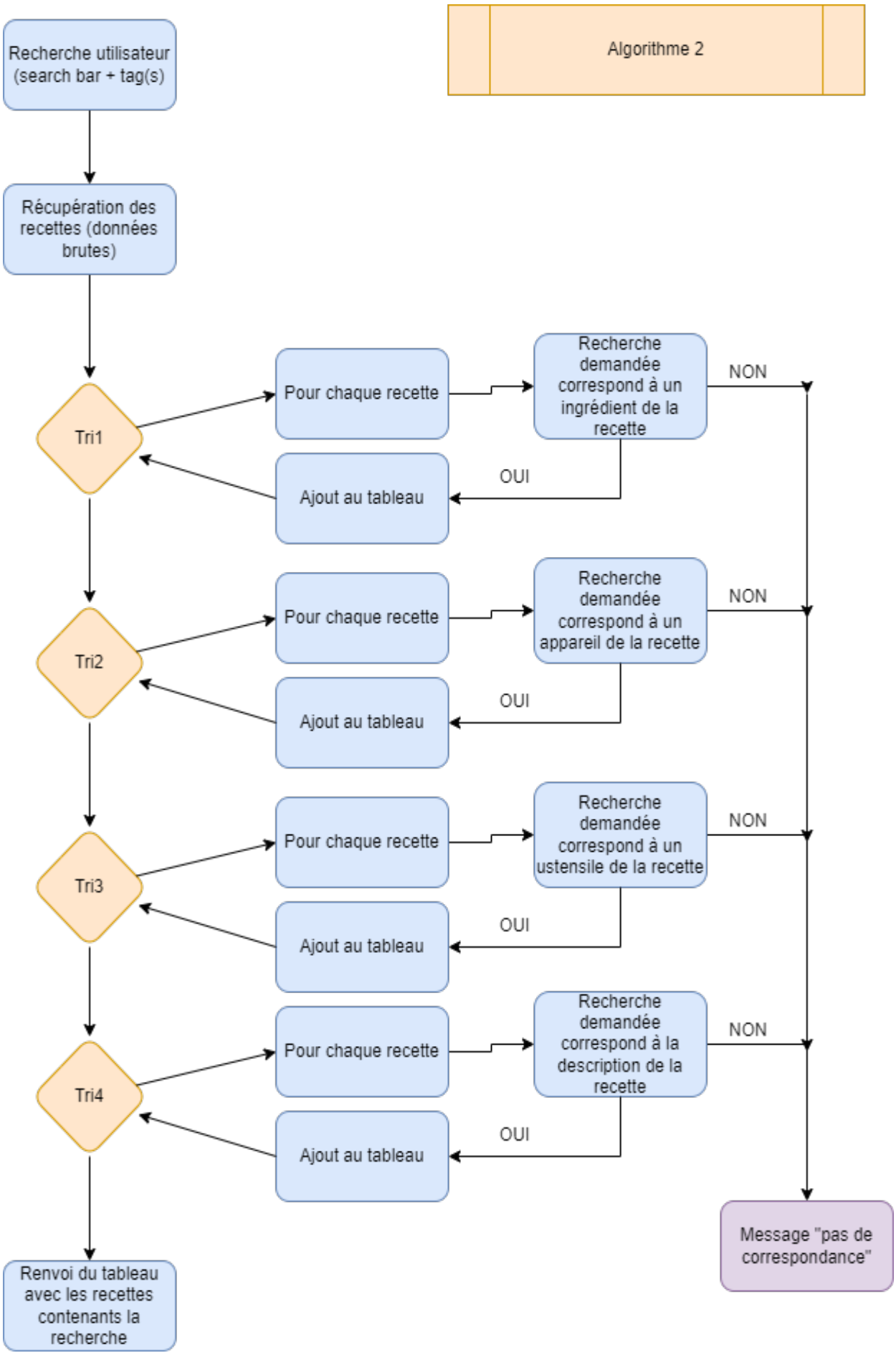

Algorithme 1
User search
Appel du tableau comprenant toutes les recettes
Pré traitement des données afin de faciliter le tri
Filtre
Renvoi du tableau avec les recettes contenants la recherche
Pour chaque recette
Ingrédient recherché = correspondance recette
Appareil recherché = correspondance recette
Ustensil recherché = correspondance recette
Recherche = correspondance nom de recette
Recherche = correspondance descritpion recette
Recherche = correspondance ingrédients recette
Message "pas de correspondance"
NON
OUI
NON
OUI
NON
OUI
OUI
NON
OUI
NON
NON

<u>OPTION 2</u> : 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


Algorithme 2
Recherche utilisateur (search bar + tag(s)
Récupération des recettes (données brutes)
Tri1
Pour chaque recette
Recherche demandée correspond à un ingrédient de la recette
NON
OUI
Ajout au tableau
Tri2
Pour chaque recette
Recherche demandée correspond à un appareil de la recette
NON
OUI
Ajout au tableau
Tri3
Pour chaque recette
Recherche demandée correspond à un ustensile de la recette
NON
OUI
Ajout au tableau
Tri4
Pour chaque recette
Recherche demandée correspond à la description de la recette
NON
OUI
Ajout au tableau
Message "pas de correspondance"
Renvoi du tableau avec les recettes contenants la recherche

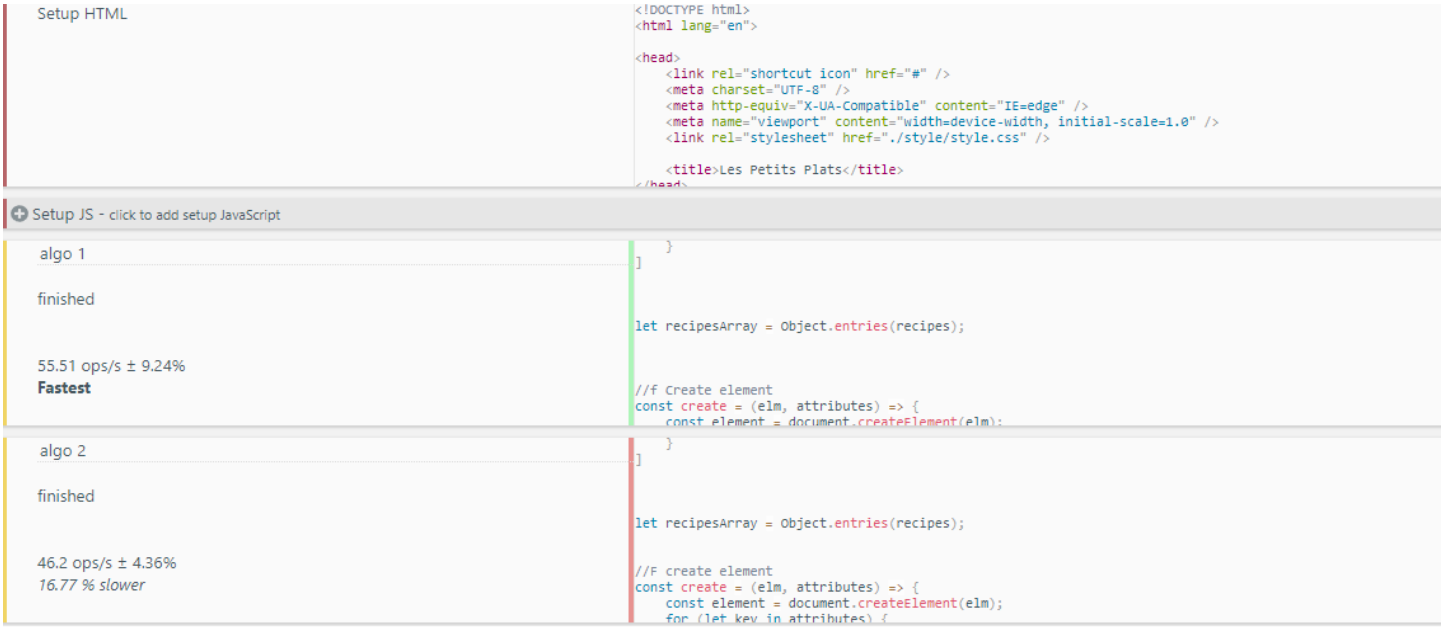
Solution retenue : Option 1 (performance, cf test)

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

Le pré-filtrage des données permet une recherche plus rapide, selon les critères demandés par l’utilisateur

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)



Différents tests effectués dans le javascript (présents sur les branches algo 1 et algo 2 du projet), avec l'utilisation de variantes startTime et endTime ainsi que de performance,now() ; renvoyant le temps nécessaire à la recherche lancée,