

Fiche d'investigation de fonctionnalité

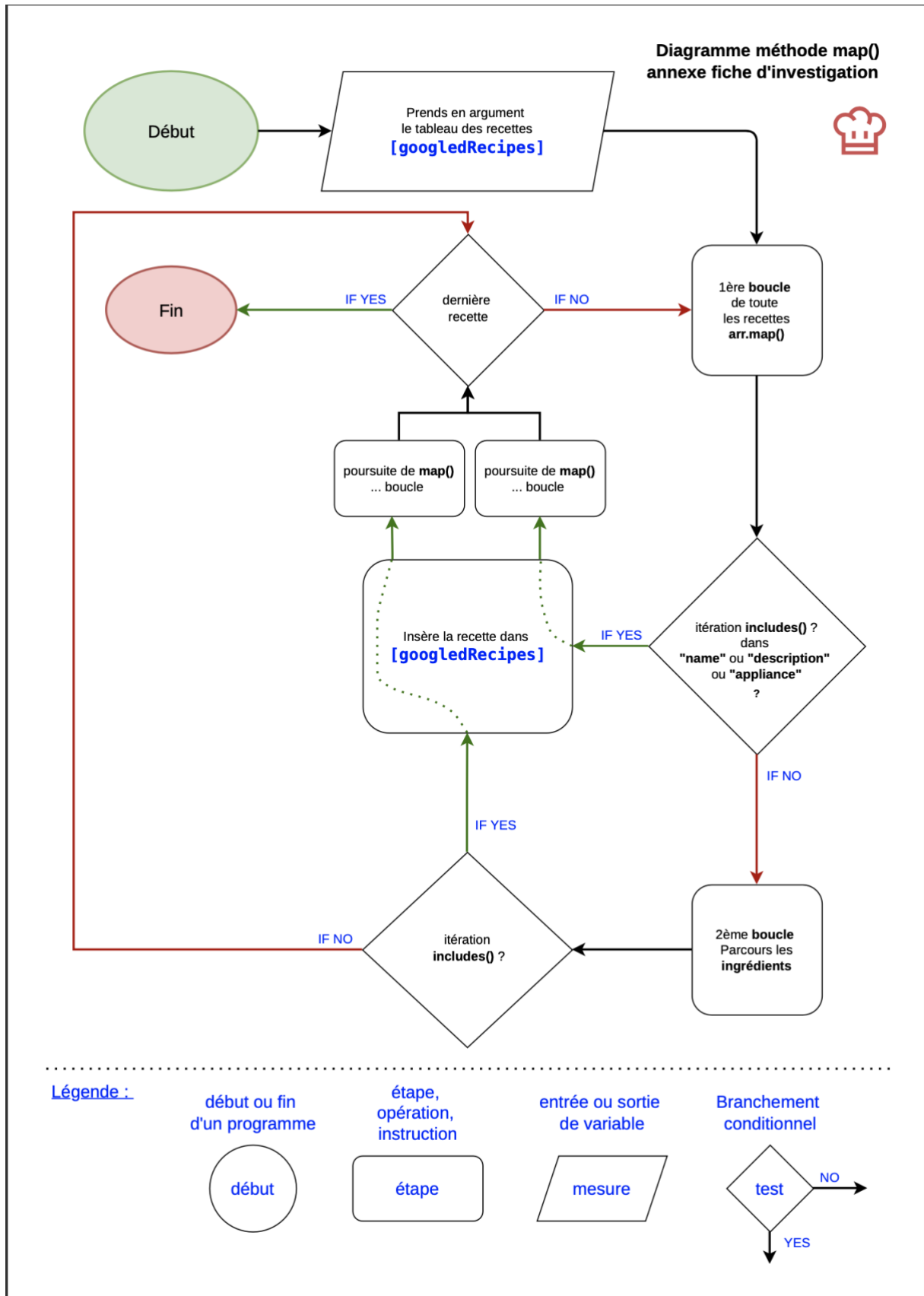
Fonctionnalité : Filtrer les recettes	Fonctionnalité #1
Problématique : Accéder rapidement à une recette correspondant à un besoin de l'utilisateur dans les recettes déjà reçues. Le cas d'utilisation commence lorsque l'utilisateur entre au moins 3 caractères dans la barre de recherche principale.	

Option 1 : Approche fonctionnelle <code>map()</code> includes()	
Avantages : <ul style="list-style-type: none"> ○ Compatible avec tous les navigateurs ○ Plus facile à lire ○ Nécessite moins de code ○ Permet de changer la nature des éléments lors du traitement : <code>map()</code> construit un nouveau tableau ○ Idéal pour utiliser une valeur de retour 	Inconvénient : <ul style="list-style-type: none"> ○ N'utilise pas les instructions « <code>break ;</code> » et « <code>continue ;</code> »
Nombre de boucle nécessaires : 3 Nombre de boucles optionnelles : 2	

Option 2 : Loop <code>for...of</code>	
Avantages : <ul style="list-style-type: none"> ○ Utilise les instructions « <code>break ;</code> » et « <code>continue ;</code> » 	Inconvénient : <ul style="list-style-type: none"> ○ N'est pas supporté par internet explorer ○ Moins facile à lire ○ Nécessite plus de code
Nombre de boucle nécessaires : 3 Nombre de boucles optionnelles : 2	

Solution retenue : Nous avons donc retenu l'algorithme « Loop <code>for...of</code> » pour ces différentes raisons : <ul style="list-style-type: none"> ▪ Parmi les boucles natives et fonctionnelles testées, la boucle « <code>for...of</code> » est apparue comme l'une des meilleures solutions selon le score obtenu au javascript Benchmark ; ▪ L'utilisation de la méthode « <code>indexOf()</code> » montre de meilleure performance que les méthodes « <code>includes()</code> » et « <code>match()</code> » ; ▪ Mettre la condition concernant le nom de la recette et la description dans le même « <code>if</code> » permet de gagner en performance ; ▪ L'utilisation des instructions « <code>break ;</code> » et « <code>continue ;</code> » permettent un gain de performance puisque dès qu'une condition est remplie, on passe à l'itération suivante. ▪ La condition « <code>if</code> » placée avant la seconde boucle permet de gagner en performance puisque la boucle parcourt un tableau.
--

Annexe 1



Annexe 2

