

Fiche d'investigation de fonctionnalité

Fonctionnalité : Filtrer les recettes	Fonctionnalité #3
Problématique : Afin d'accéder rapidement à une recette, nous cherchons à trouver le meilleur algorithme de tri pour filtrer les recettes.	

Option 1 : Algorithme avec des boucles natives. Dans cette option, nous choisissons d'utiliser des boucles natives. Nous effectuons la recherche en itérant chaque recette à l'aide d'une boucle For et nous cherchons les correspondances avec le nom ou la description et sinon dans les ingrédients. Si une recette correspond nous l'insérons dans un nouveau tableau.
--

Avantages : <ul style="list-style-type: none">- Les boucles natives sont plus rapides pour parcourir un tableau.- Elles permettent de répéter des actions simplement.- Elles sont compatibles avec tous les navigateurs.- Elles sont très modulables et facilement utilisables.	Inconvénients : <ul style="list-style-type: none">- Elles deviennent plus lente lorsqu'elles sont imbriquées.- Leur utilisation est moins lisible/ Moins lisible.
---	---

Option 2 : Algorithme avec de la programmation fonctionnelle Dans cette option, nous choisissons d'utiliser de la programmation fonctionnelle. Nous filtrons le tableau des recettes avec la méthode filter() et avec une fonction callback nous cherchons les correspondances dans le nom, la description et les ingrédients que nous avons préalablement mis dans un tableau avec la méthode map(). La méthode filter() permet de retourner un tableau avec tous les éléments correspondants.

Avantages : <ul style="list-style-type: none">- Elles permettent d'avoir un code plus compréhensible et plus court/ Meilleur compréhension du code- Code plus court- Les méthodes filter() et map() sont compatibles sur tous les navigateurs.	Inconvénients : <ul style="list-style-type: none">- Leur application est plus difficile- Elles ne sont pas modulables
---	---

Solution retenue : La solution retenue est celle de l'option 2. Nous pouvons remarquer que la méthode filter() est très adaptés à notre situation et à notre utilisation. De plus, lors de la mesure de performance, nous remarquons que cette solution est aussi la plus performante.
--

Annexes

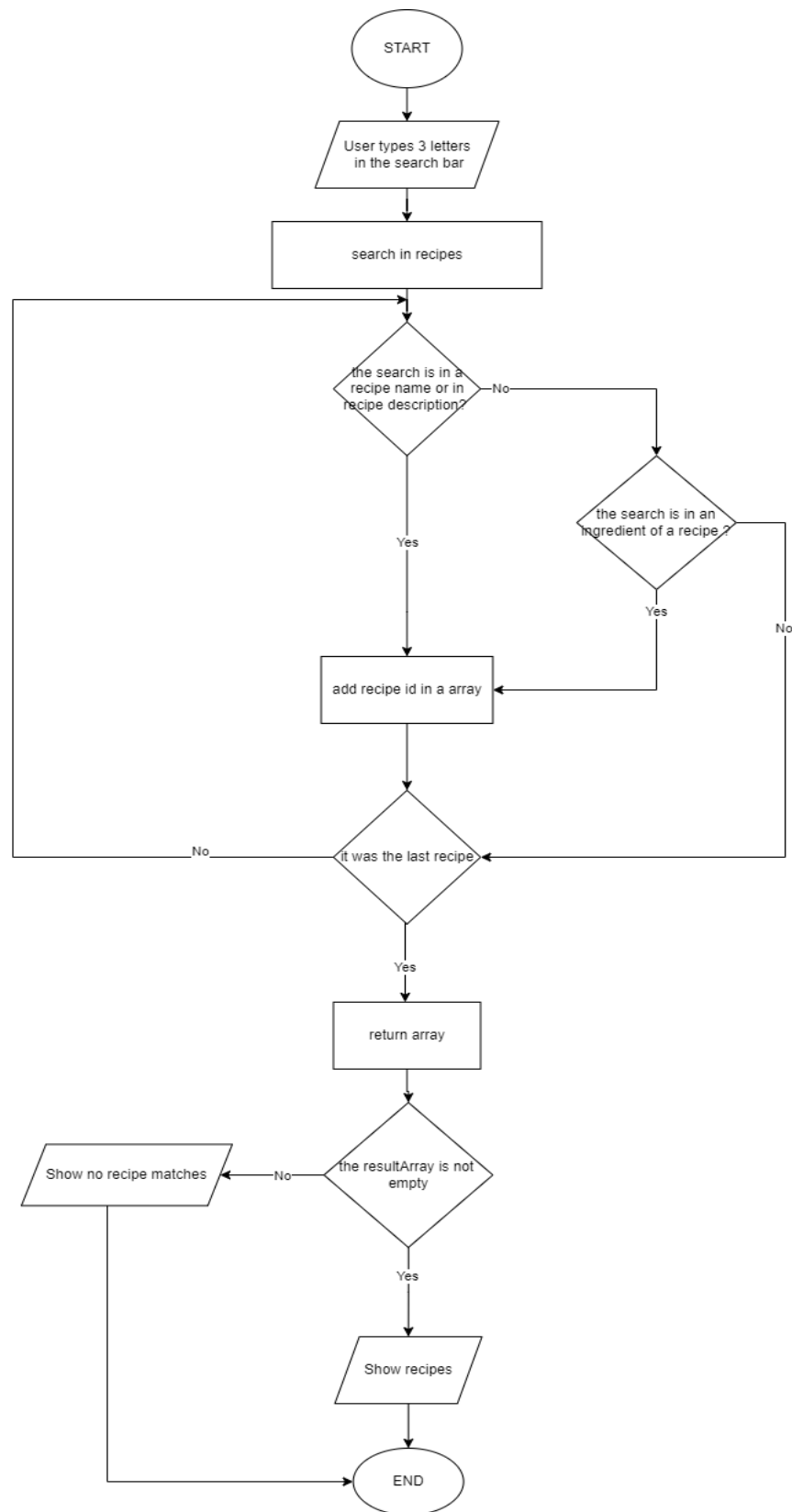


Figure 1 : Diagramme avec boucles natives.

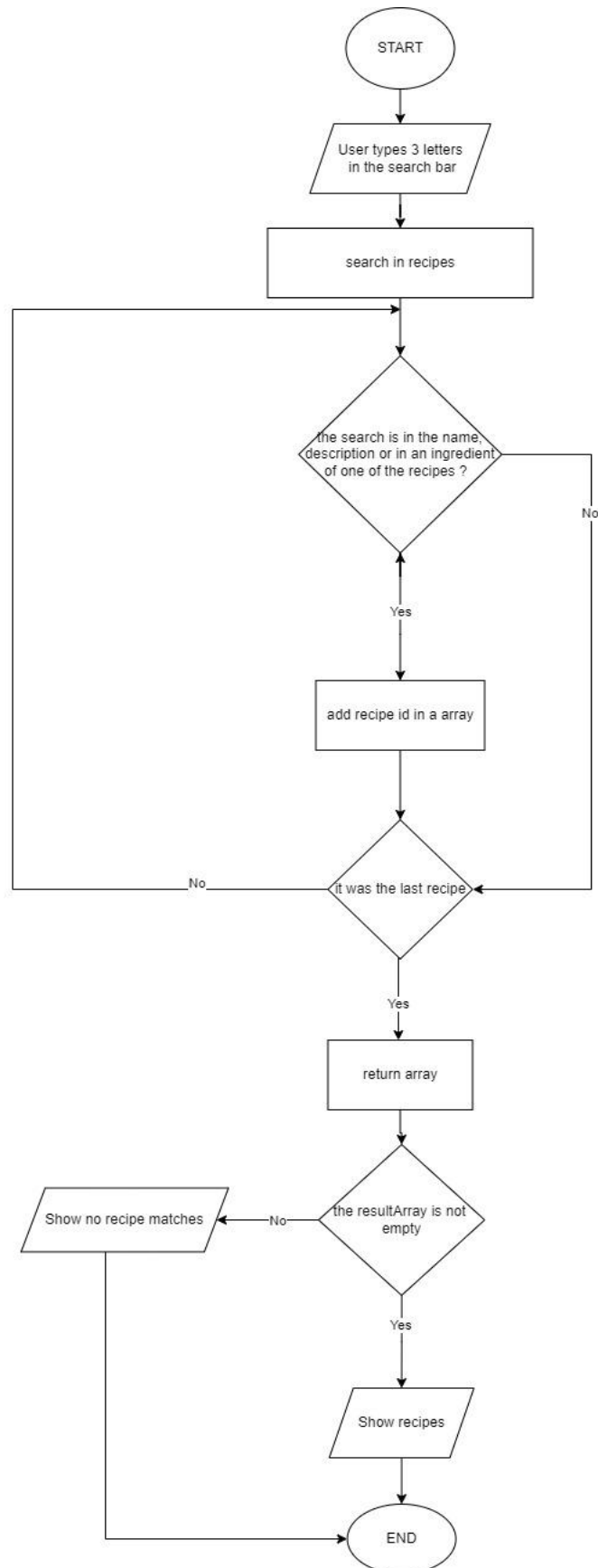


Figure 2 : Diagramme avec programmation fonctionnelle.