

Fonctionnalité : Recherche	Fonctionnalité #1
Problématique : Afin d'assurer la fluidité et la performance du moteur de recherche, nous cherchons à avoir un algorithme de recherche rapide et presque instantanée.	

Option 1 : Usage des boucles natives Dans cette option, nous utiliserons les boucles natives comme "for" ou "while" pour traiter les tableaux. L'avantage: il donne généralement de meilleures performances que l'utilisation des méthodes de l'objet array. Le principal inconvénient : Le code est beaucoup plus verbeux dans la plupart des cas.	
Avantages <ul style="list-style-type: none">⊕ Performance est généralement meilleure, surtout pour boucler sur un tableau de plus de mille éléments⊕ La plupart des développeurs sont plus familiarisés avec les boucles qu'avec les méthodes de l'objet array.	Inconvénients <ul style="list-style-type: none">⊖ Code est plus verbeux
Nombre d'opérations par seconde: <ul style="list-style-type: none">- Recherche par keyword: 63 436- Recherche par tag: 108 156- Recherche par keyword et tag: 98 641	Nombre de lignes de code: 57 lignes

Option 2 : Usage des méthodes de l'objet array Dans cette option, nous utiliserons les méthodes de l'objet array comme "map", "filter", "reduce", "forEach" pour traiter les tableaux. L'avantage: le code est plus court, assurer la lisibilité Le principal inconvénient: la performance est inférieure à la boucle native, recommandée pour boucler sur un tableau de moins de mille éléments	
Avantages <ul style="list-style-type: none">⊕ Code est plus lisible lorsqu'elles gèrent une certaine logique en interne, comme la création d'un nouveau tableau ou le filtrage de valeurs spécifiques	Inconvénients <ul style="list-style-type: none">⊖ Plus lent que les boucles natives à cause de la fonction de rappel⊖ Certaines méthodes sont complexes à utiliser, par exemple: reduce
Nombre d'opérations par seconde: <ul style="list-style-type: none">- Recherche par keyword: 40 559- Recherche par tag: 52 767- Recherche par keyword et tag: 50 533	Nombre de lignes de code: 45 lignes

Solution retenue : Nous avons donc retenu l'option des boucles natives pour assurer la meilleure performance de recherche et garantir l'instantanéité.
Réalisons que le nombre de recettes augmentera et dépassera certainement les mille recettes, nous utiliserons les boucles natives pour répondre à l'évolutivité du système dans le futur.
Cependant, pour d'autres fonctionnalités nécessitant un traitement de tableau en boucle avec un nombre d'éléments inférieur à 1000, nous vous recommandons d'utiliser les méthodes de l'objet array pour faciliter la lecture du code source.

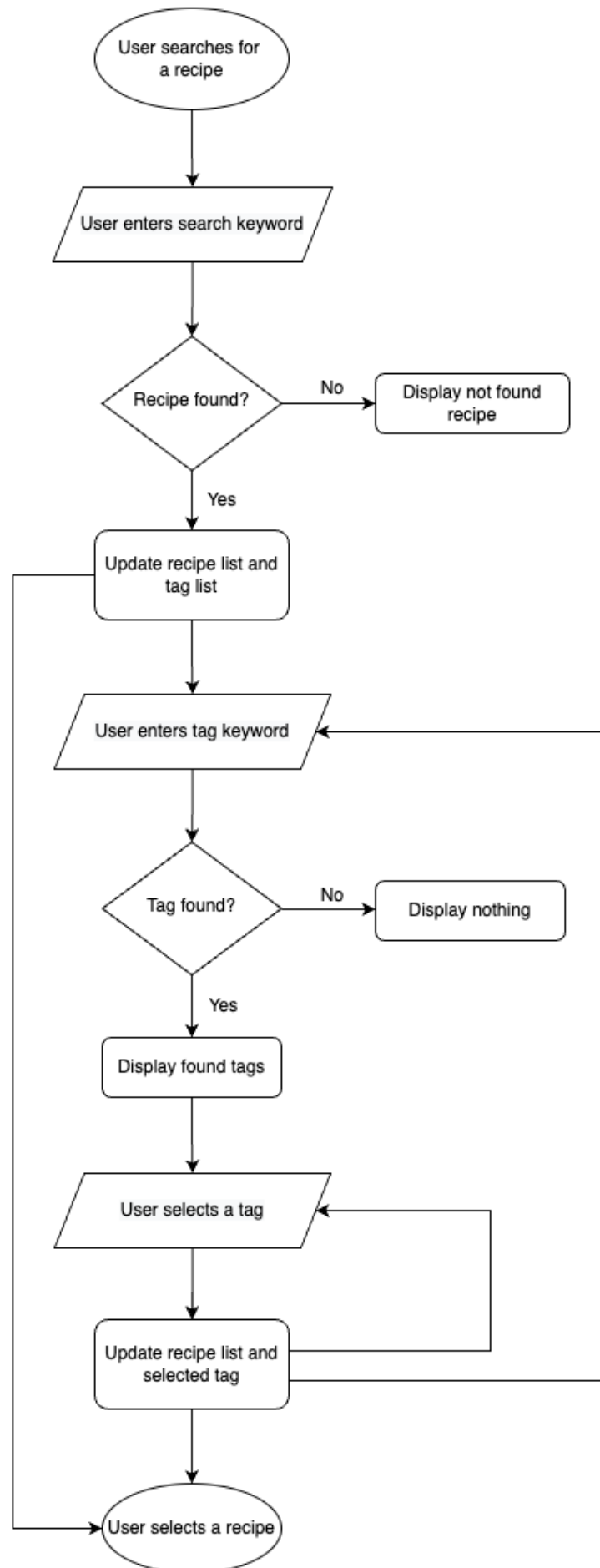


Figure 1 - Diagramme d'activité