

Fiche d'investigation de fonctionnalité

Fonctionnalité : Recherche de recettes + tags	Fonctionnalité #1
Problématique: Accéder rapidement à une recette correspondant à un besoin de l'utilisateur dans les recettes déjà reçues selon les données entrées	

Option 1: Utilisation des boucles avancées forEach, filter etc...	
Dans cette option, le code utilise des boucles et des méthodes de tableau avancées pour traiter rapidement nos recettes, garantissant ainsi à l'utilisateur des résultats en adéquation avec ses recherches.	
Avantages <ul style="list-style-type: none">- Code plus simple- Utilisation de méthodes pratiques et avancées- Facile à maintenir	Inconvénients <ul style="list-style-type: none">- Niveau avancé en javascript demandé
Nombre de champs minimum à remplir pour la recherche : 1 searchValue	

Option 2: Utilisation des boucles natives for, while etc...	
Dans cette option, le code utilise des boucles natives de JavaScript ainsi que des méthodes de tableau simples pour traiter rapidement nos recettes, garantissant ainsi à l'utilisateur des résultats en adéquation avec ses recherches.	
Avantages <ul style="list-style-type: none">- Connaissances de bas suffisantes- Utilisation de méthodes simples	Inconvénients <ul style="list-style-type: none">- Code plus complexe- Maintenance difficile
Nombre de champs minimum à remplir pour la recherche : 1 searchValue	

Solution retenue :
Après une évaluation approfondie des deux solutions algorithmiques pour résoudre la problématique, j'ai opté pour l'implémentation du code 1 dans mon application. Cette décision repose principalement sur des considérations de performance. Le code 1 a obtenu un score de 100% lors des tests effectués sur jsben.ch, ce qui indique une exécution optimale dans le contexte de référence. En revanche, le code 2 a obtenu un score de seulement 38%, ce qui suggère qu'il présente des performances inférieures par rapport à la solution de référence



Figure: Diagramme d'activité fonctionnalité de recherche de recettes