Fiche d’investigation de fonctionnalité

Fonctionnalité : Recherche de recettes	Fonctionnalité#1
Problématique : La recherche de recette étant l’intérêt principal du site, cette dernière doit être la plus rapide possible. Les données sont stockées sous différents tableaux qu ‘ils faut parcourir afin de trouver les informations correspondantes à la saisie utilisateur.	

Option 1 : Utilisation de boucles natives. Dans cette option nous récupérons la saisie utilisateur. Nous parcourons toutes les recettes une à une pour vérifier la suite de conditions désirées. Le principal inconvénient est que plus le jeu de données est important plus l’algorithme sera lent.	
Avantages : • Comptabilité du code avec des navigateurs ancien. • Lecture du code intuitive car composé d’éléments basiques (boucle & conditions). • Performance similaire sur tous les navigateurs (ne dépendant pas d’implémentation des différentes méthodes).	**Inconvénients :** • Code ne bénéficiant pas des optimisations offerte par les évolutions du JS. • Nombres d’opérations sur un temps T toujours inférieurs à l’implémentation #2

Option 2 : Utilisation des méthodes de l’objet Array. Nous utilisons les méthodes proposées par les dernières versions de JS (notamment .map()) afin de faire un tri des données, l’algorithme est similaire « boucle → condition → résultat » mais la partie « boucle » est différente de l’option #1 car optimisée (selon le moteur JS utilisés l’implémentation peut être différentes) .	
Avantages : • Performances accrues. • Peu de changement pour un gain de performance. • Code pouvant être plus compact (au détriment de sa clarté)	**Inconvénients :** • Code non compatible avec de vieux navigateurs.

<u>Conclusion :</u>

L’utilisation des méthodes proposées par l’objet Array grâce aux dernières versions de JS est toujours plus efficaces d’après les différents tests exécutés sur https://jsben.ch/Mm6bj.
Un gain moyen de 5 % d’opérations est obtenu.
Les deux implémentations utilisent le même algorigramme la différence de performance est donc obtenue exclusivement avec les avantages de la méthode .map()
Il aurait été intéressant de tester d’autres méthodes de cette objet permettant d’obtenir un résultat similaire.

Algorigramme