Fiche d'investigation et de fonctionnalité

Fonctionnalité : algorithme de recherche	Fonctionnalité **#1**
Problématique : Développer un algorithme de recherche qui soit à la fois performant et rapide pour l'ergonomie d'usage des utilisateurs. Cet algorithme doit être décliné en 2 versions différentes qui proposent une solution algorithmique. Les 2 méthodes doivent ensuite être comparées dans un workbench afin d'en évaluer la performance. La solution la plus rapide est sélectionnée.	

Option 1 : Recherche dans les instances de recettes (cf Figure 1)	
La solution employée dans cette option est l'utilisation de la programmation orientée objet. Les caractères renseignés dans l'input de recherche principale sont comparés avec les propriétés de chaque instance relative au tableau de recettes. Entre autres : id, nom, ingrédients, description, appareils, ustensiles etc.	
Avantages : Utilisation tels quels des objets du fichier json Code + succinct, plus maintenable car mieux organisé	**Inconvénients :** La POO est moins intuitive que la programmation fonctionnelle La manipulation des objets nécessite d'être davantage rigoureux qu'avec la manipulation d'une chaîne de caractères (*cf option 2)*

Option 2 : Recherche dans une chaîne de caractères (cf Figure 2)	
Dans cette option la liste d'objets du format json est transformé en une chaîne de caractères. Les instances recettes sont alors transformées en des éléments d'un seul tableau comprenant les 50 recettes. Les propriétés sont supprimées pour garder uniquement le contenu textuel qui leur est associé. Les saisies renseignées dans l'input sont comparées à cette chaîne de caractère.	
Avantages : La recherche s'effectue de manière linaire en parcourant les chaînes de caractère d'un seul et même tableau. On peut ainsi utiliser les tableaux résultant de ces chaînes de caractère plus facilement grâce à des variables, sans faire appel à d'autres fonctions intermédiaires.	**Inconvénients :** Le code nécessite une étape supplémentaire qui est la transformation des objets du json en chaîne de caractères. Il est donc plus long.

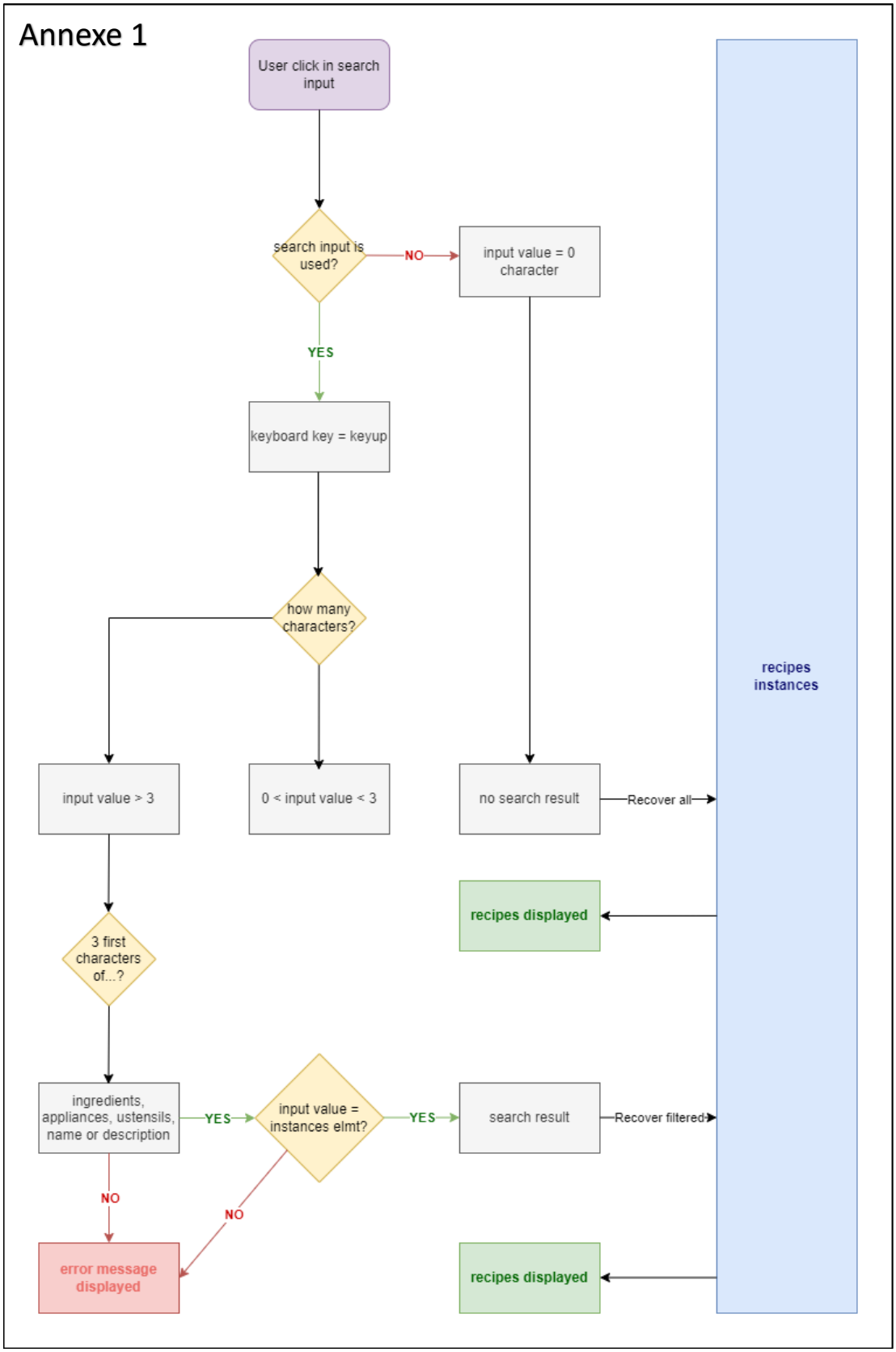


Figure 1: fonction de recherche dans les instances de recettes

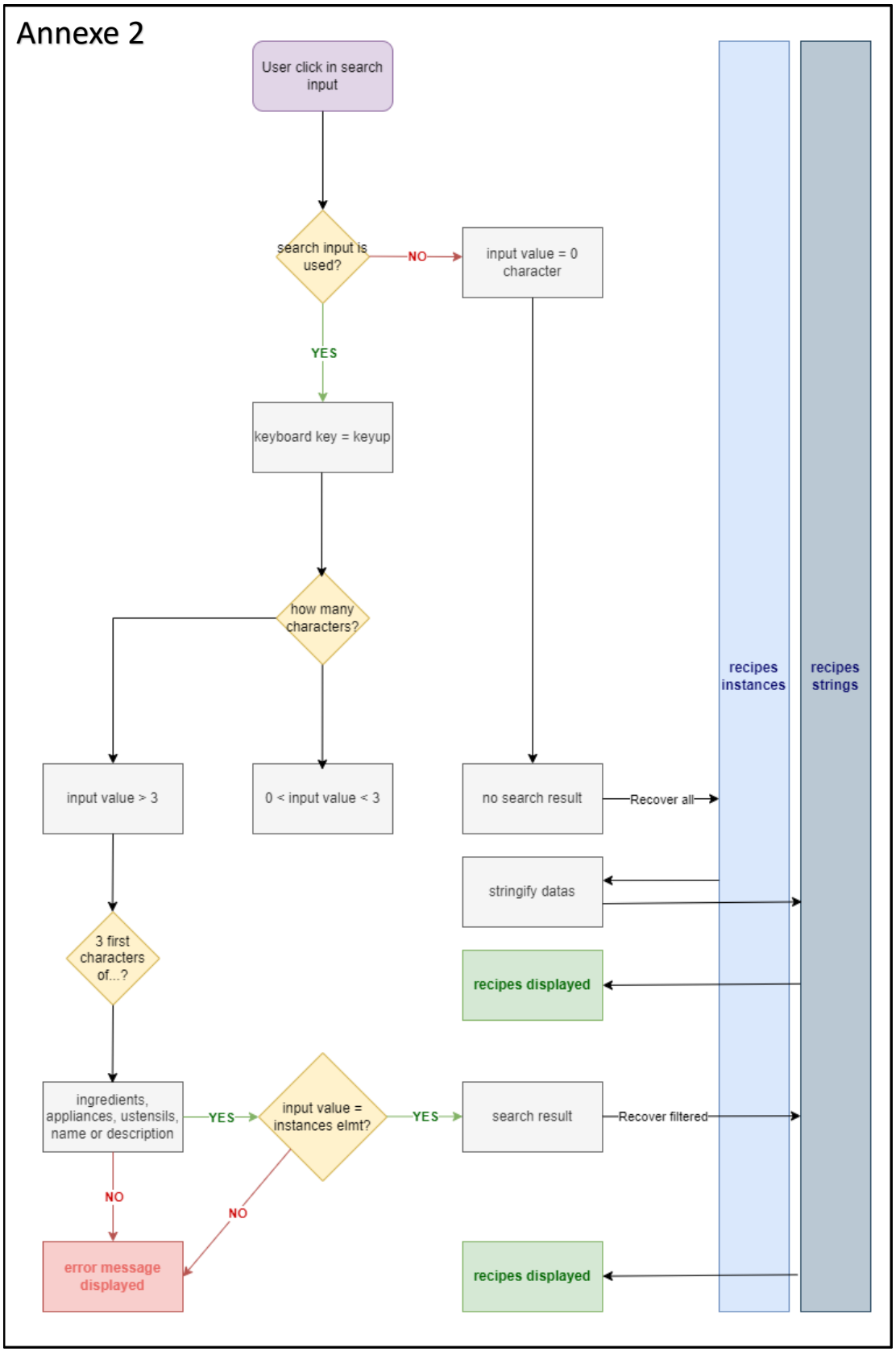
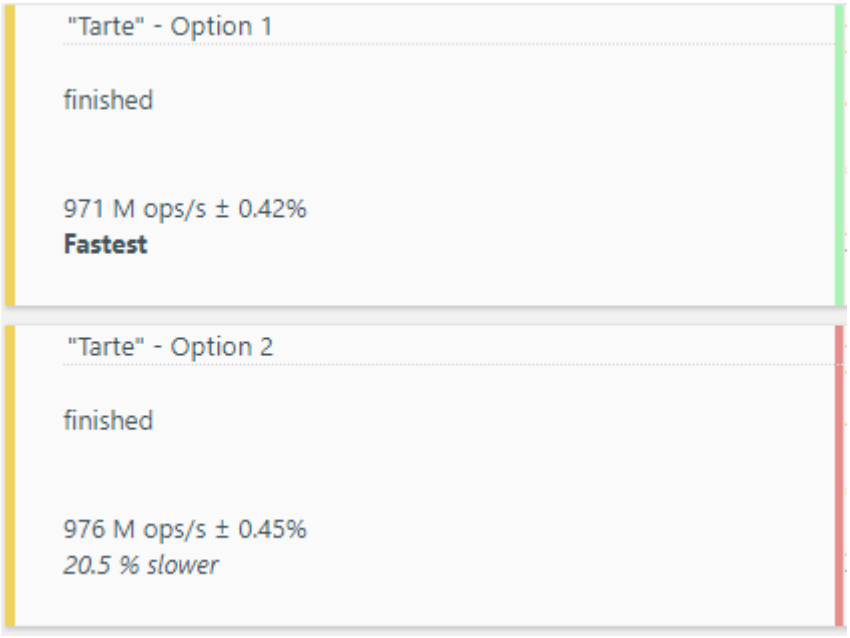


Figure 2 : fonction de recherche dans les recettes sous forme de chaîne de caractères

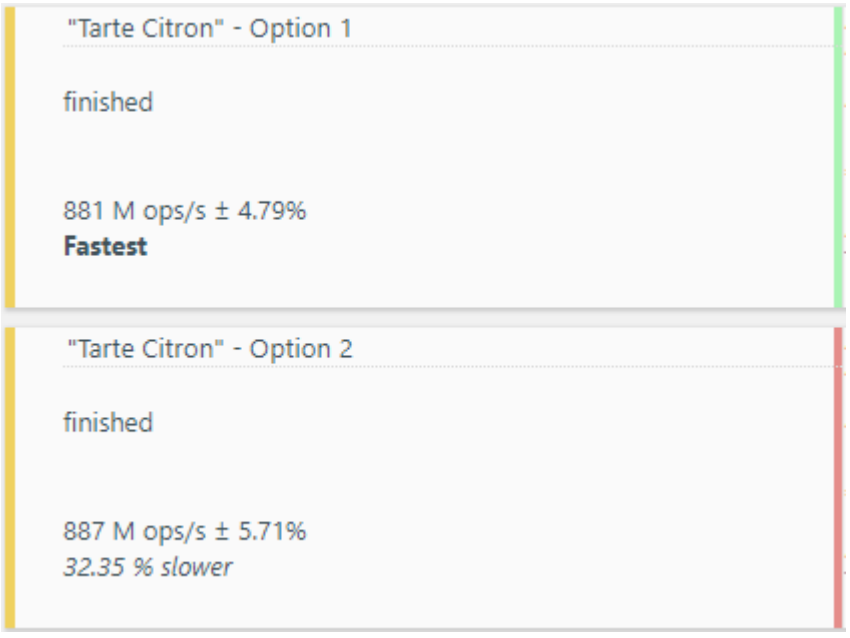
Annexe 3

BENCHMARCK : TABLEAU COMPARATIF N°1



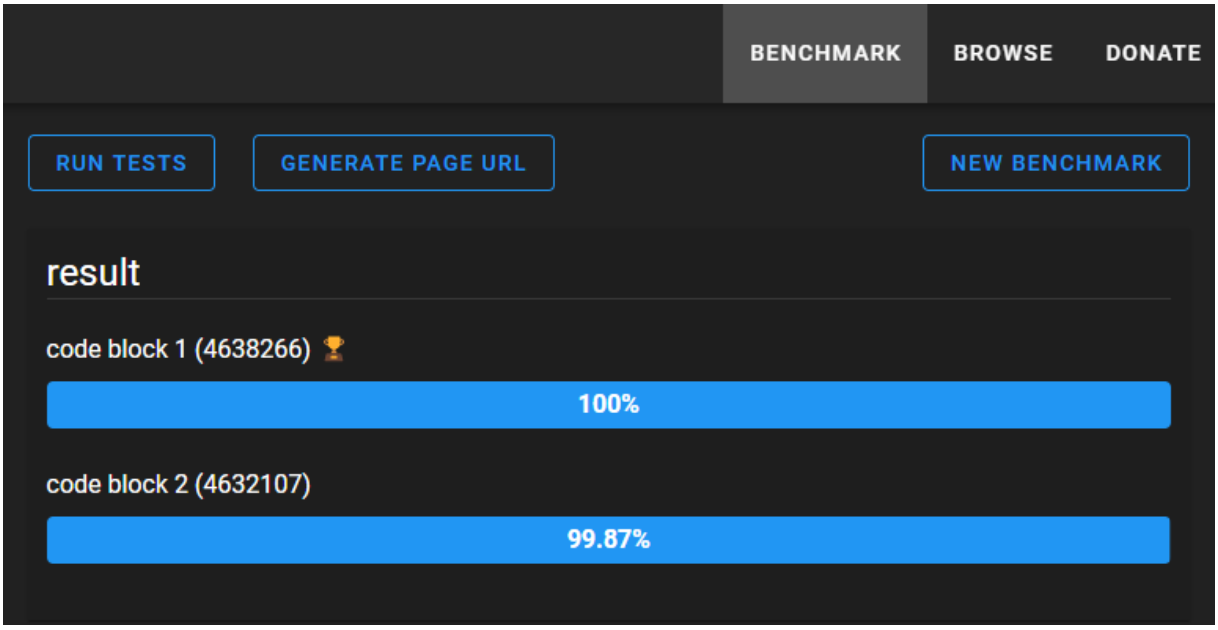
Comparaison des 2 méthodes réalisée sur le site jsbench.me. Temps de recherche pour « tarte ».
La méthode 1 qui utilise la POO est plus performante de 20.5 %

BENCHMARCK : TABLEAU COMPARATIF N°1-bis



Comparaison des 2 méthodes réalisée sur le site jsbench.me. Temps de recherche pour « tarte citron ».
La méthode 1 qui utilise la POO est plus performante de 32.35 %

BENCHMARCK : TABLEAU COMPARATIF N°2



Comparaison des 2 méthodes réalisée sur le site Jsben.ch
La méthode 1 qui utilise la POO est globalement plus performante de 0.13%

Solution retenue :

Les tests démontrent une grande différence entre les deux méthodes. Bien que la performance semble être quasi similaire sur le code en lui-même au niveau de sa structure (0.13% de différence), la mise en application illustre un écart de **20.5%** à **32.35 %.**

La plus optimale en termes de rapidité est donc la méthode 1. Elle est plus performante dans chacun des tests réalisés, avec 2 sites et 2 méthodes différentes.