

## Fiche d'investigation de fonctionnalité

<b>Fonctionnalité:</b> algorithme de Recherche	Fonctionnalité #1
<b>Problématique :</b> Avoir un algorithme de recherche performant et surtout rapide pour le confort de l'utilisateur	

<b>Option 1 : Recherche dans les instance de recettes (figure 1)</b> Comme son nom l'indique, on utilise la programmation orientée objet. La valeur des caractères renseignés dans l'input est comparée avec les propriétés de la recette (nom, description, etc...)	
<b>Avantages</b> Peu de lignes de codes mise à profit du paradigme objet	<b>Inconvénients</b> Si on ajoute une propriété dans la classe Recette, nécessite de retravailler le code

<b>Option 2 : Recherche dans une liste de string (figure 2)</b> Convertit la liste d'objets en liste de string. La valeur des caractères renseignés dans l'input est ensuite comparée avec cette seule string.	
<b>Avantages</b> Si on ajoute une propriété dans la classe Recette, ne nécessite pas de retoucher au code. Donc plus facilement maintenable	<b>Inconvénients</b> code plus long avec d'avantages d'étapes pour parvenir au résultat

## ANNEXE 1

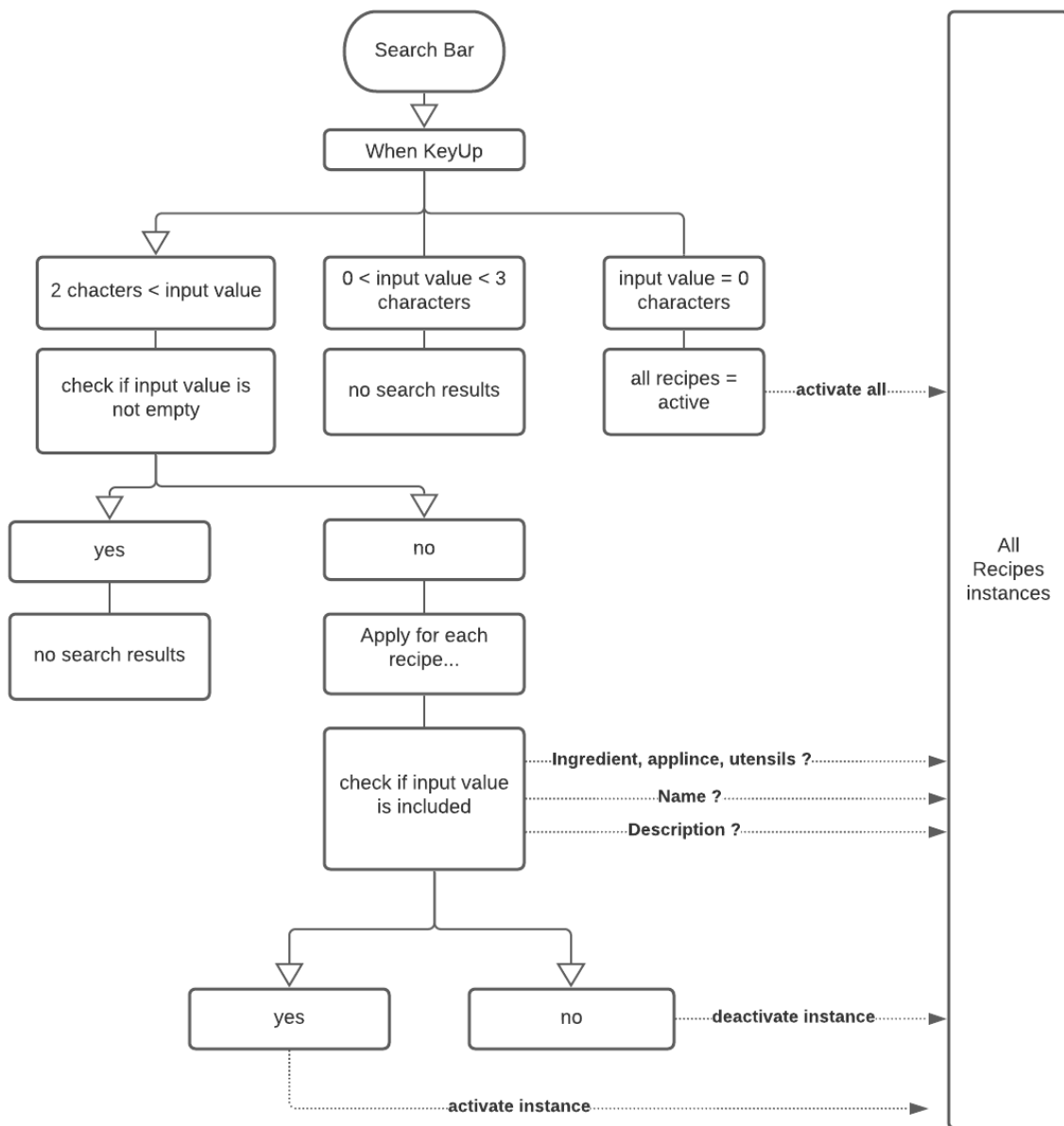


Figure 1 : fonction de recherche dans les instances de recettes

## ANNEXE 2

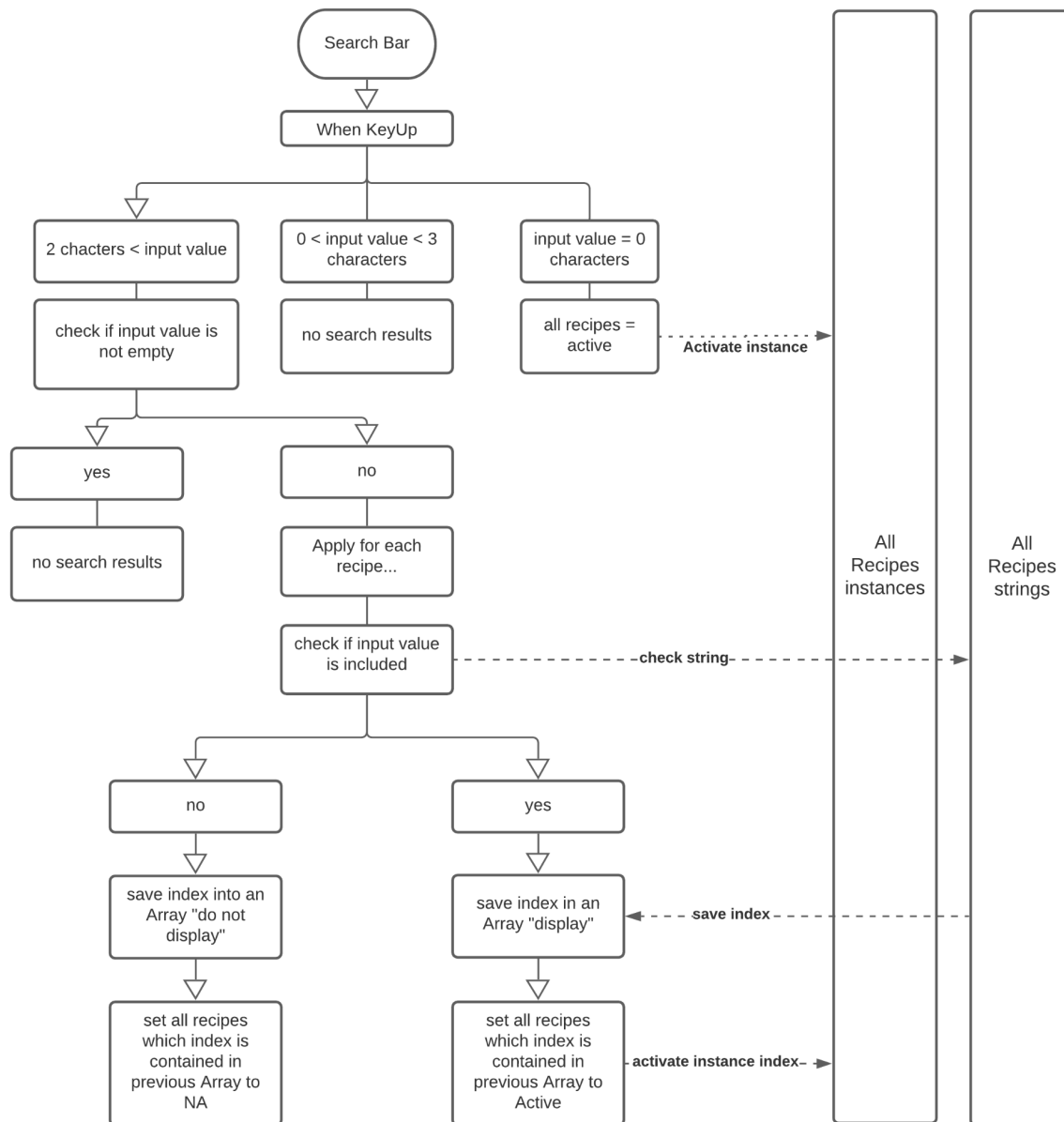


Figure 2 : fonction de recherche dans des listes de strings

### ANNEXE 3

"concombre Thon" - Méthode 1
finished
329479.59 ops/s $\pm$ 0.55%
<b>Fastest</b>

"concombre Thon" - Méthode 2
finished
272517.15 ops/s $\pm$ 0.63%
17.29 % slower

Tableau 1. Temps de recherche en fonction des méthodes pour "concombre thon"

"Thon" - Méthode 1
finished
658083.62 ops/s $\pm$ 0.61%
<b>Fastest</b>

"Thon" - Méthode 2
finished
546005.96 ops/s $\pm$ 0.97%
17.03 % slower

Tableau 2. Temps de recherche en fonction des méthodes pour "thon"

#### Solution retenue :

Au vu des tests il existe une grande différence entre les deux méthodes. La plus intéressante en termes de rapidité est donc la méthode 1, qui dans les deux exemples s'est avérée être la plus rapide.