Fiche d'investigation de fonctionnalité

Fonctionnalité : Barre de recherche des recettes	**Fonctionnalité**
Problématique : Permettre d'avoir une barre de recherche des recettes performant utilisant différents filtres, en fonction des : ingrédients, des ustensiles et des appareils.	

Option 1 : Fonction de filtrage (filter()) - Figure 1

La fonction de filtrage « filter() » est utilisée pour créer un nouveau tableau qui contient un sous-ensemble des éléments d'un tableau d'origine en fonction de condition(s) donnée(s).

Cette fonction prend en entrée un tableau (nos recettes) et une fonction de filtre (nos filtres ingrédients, ustensiles et appareils), qui teste chaque élément du tableau **une seul fois** (ce qui en fait une méthode efficace). Si la fonction de filtre retourne « true » pour un élément donné (une recette), cet élément est inclus dans le nouveau tableau (nos recettes filtrées). Sinon, il est ignoré.

Avantages	**Inconvénients**
• Efficace en performance, car elle parcourt les éléments du tableau donné qu'une seule fois. • Facile à utiliser pour les cas de filtrage simple. • Retourne un nouveau tableau (conserve l'original) • Code plus lisible et maintenable.	• Peut être utilisé que pour des filtrages simples. • Il n'est pas possible de modifier les éléments du tableau d'origine.

Conclusion

Approche pouvant être efficace, facile à utiliser, et maintenable. Mais n'est viable que pour des filtres simples.

Option 2 : Recherche linéaire

La recherche linéaire est une méthode permettant de trouver un élément spécifique dans un tableau en parcourant chaque élément du tableau (plusieurs fois si il le faut), jusqu'à ce que l'élément recherché soit trouvé. Cette méthode se base sur des boucles native « for » qui pourra renvoyer un tableau avec les recettes correspondant à la recherche.

Avantages	**Inconvénients**
• Simple à mettre en place. • Flexible et peut être utilisé pour des cas de recherche plus complexes	• Généralement moins efficace en termes de performance car il parcourt les éléments plusieurs fois. • Il peut carrément être inefficace pour des tableaux de grande taille (avec énormément d'éléments) • Lisibilité du code complexe (plusieurs boucles imbriqués)

Conclusion :

Approche pouvant être efficace en cas de recherche avec beaucoup de complexités, mais rapidement limité si il y a beaucoup d'éléments à analyser.

Solution retenue :

J'opterai plutôt pour l'option 1 « Fonction de filtrage » pour ces avantages de simplicité et de clarté du code. De plus, elle sera plus facilement maintenable en cas de modification de ce système de recherche. (Une recette comportant plusieurs informations différentes, il sera plus facile de « moduler » l'algorithme en fonction des besoins futurs).

Annexes

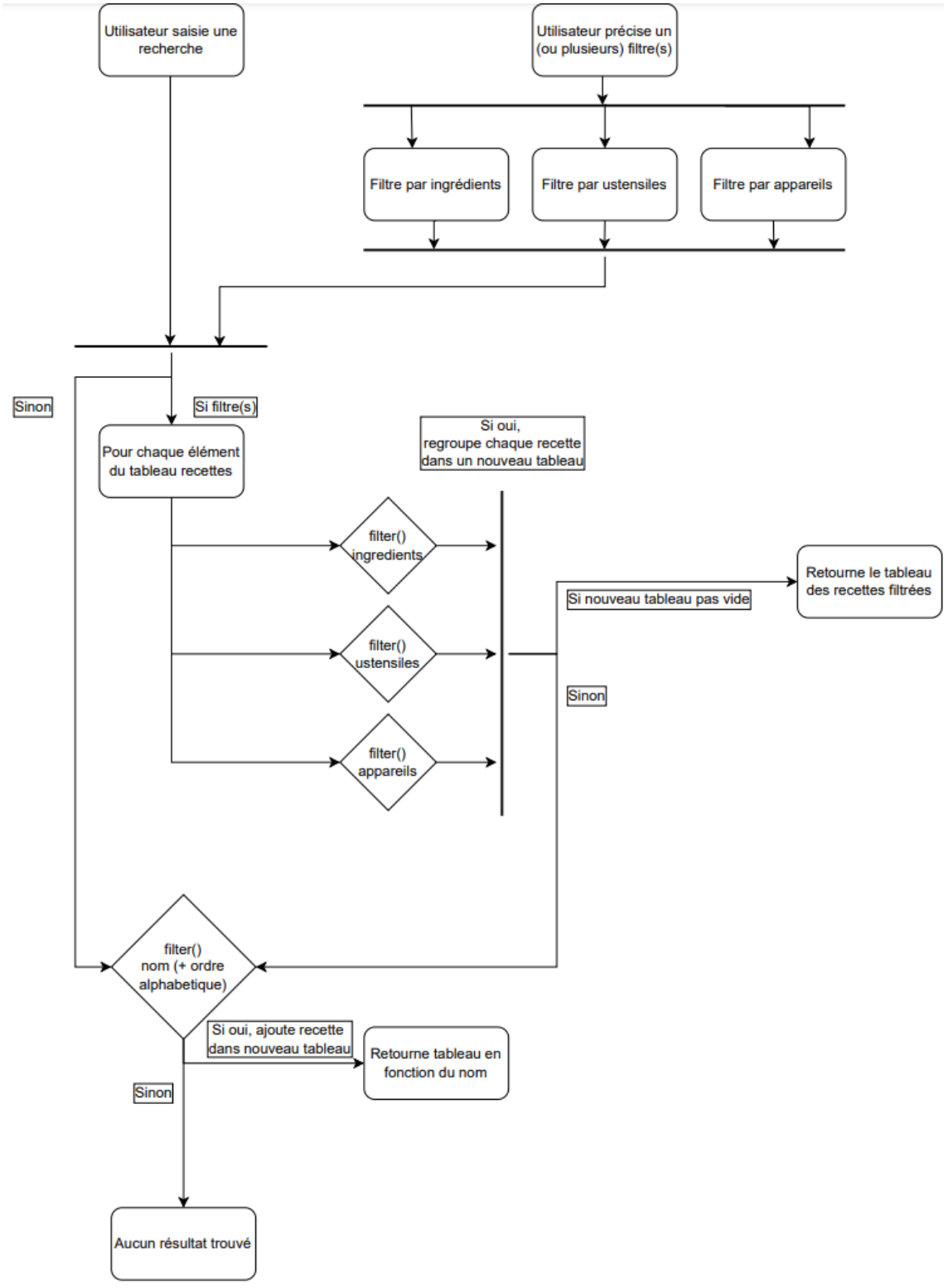

Figure 1 - Diagramme d'activité avec filter()

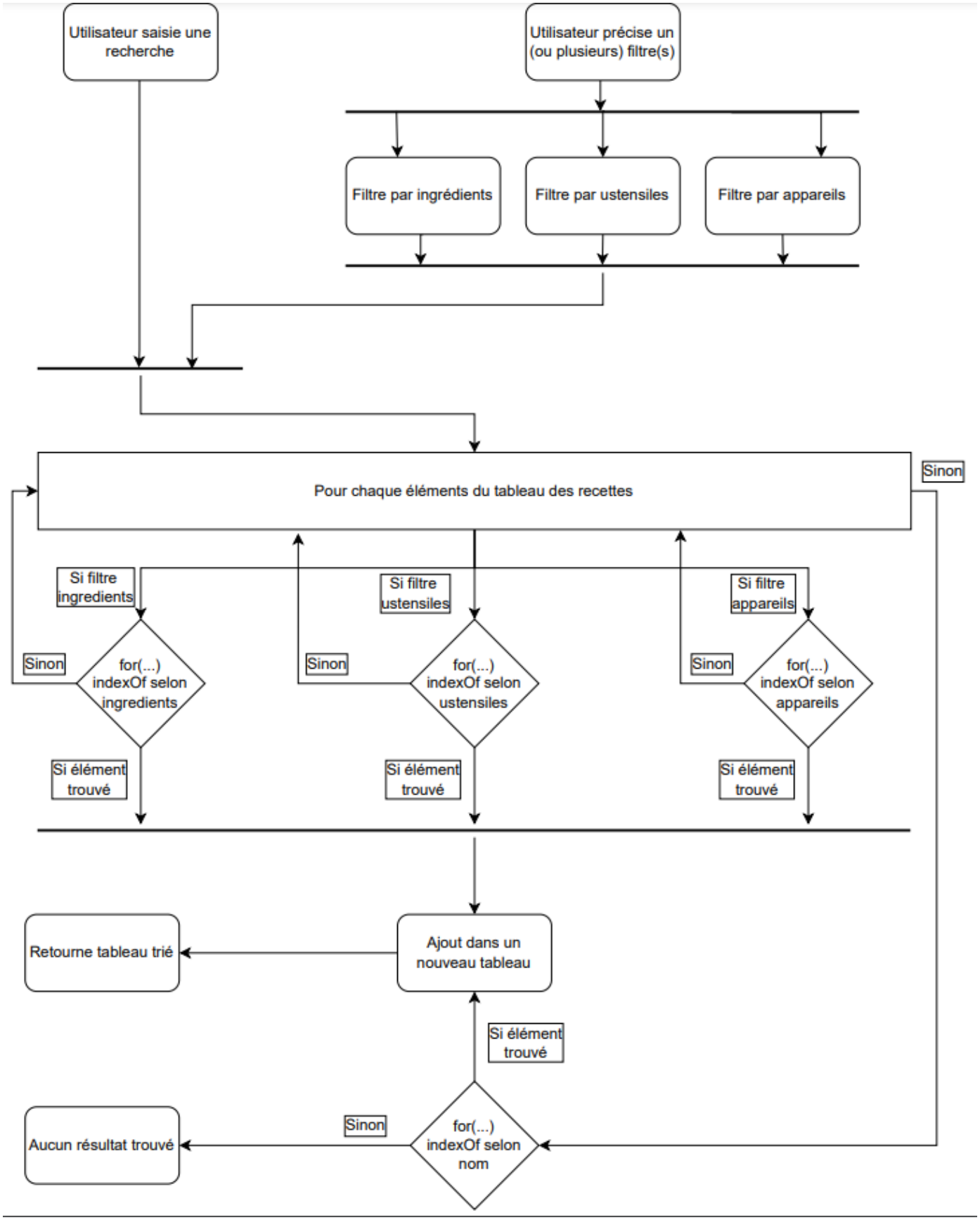

Figure 2 : Diagramme d'activité avec boucles « for »