Fiche d’investigation de fonctionnalité

Fonctionnalité : Recherche Principale	**Fonctionnalité #2**
Problématique : Afin de pouvoir retenir un maximum d’utilisateurs, nous cherchons à avoir un système de recherche Le plus efficace possible, qui donne l’illusion d’une recherche instantanée.	

Option 1 : Approche « Native » (Loops) (cf Figure)
Dans cette option, l’ensemble des éléments de recherche ont été codés en utilisant les boucles natives JS (for, while). Cela assure de très bonnes performances dans les itérations sur des tableaux qui peuvent se révéler complexes car contenant eux mêmes des objets et des tableaux imbriqués.

Avantages	**Inconvénients**
⊕ Meilleurs performances sur opération sur tableau ⊕ Syntaxe d’algorithme simple	⊖ Parcourt souvent l’ensemble des éléments d’un tableau ⊖ Code verbeux, n’utilisant pas les options JS modernes

Nombre d’opérations par seconde dans benchmark : 188 234

Option 2 : Approche « Fonctionnelle » (Filter) (cf Figure)
Dans cette option, on utilise les méthodes modernes JS telle que «.filter() » ce qui permet d’utiliser des algorithmes de tri propres aux versions modernes de Javascript. Leurs performances sont souvent meilleures que celles des boucles natives dans leurs domaines respectifs. Le code est aussi plus concis et simple à lire.

Avantages	**Inconvénients**
⊕ Excellentes performances de recherche et tri ⊕ Code lisible et concis ⊕Chaînage possible avec map, reduce, sort etc.. pour effectuer des opérations plus complexes ⊕ Nombre de variables à déclarer inférieur	⊖ Ces méthodes renvoient souvent un nouveau tableau Ce qui peut entraîner de grosses consommations mémoires selon le tableau d’origine. Idem lorsqu’elles font appel à des fonctions de callback - Pour des opérations extrêmement simples, les boucles for natives peuvent parfois se révéler plus performantes

Nombre d’opérations par seconde dans benchmark : 158 923

Solution retenue :
Étant donné que nous donnons la priorité à la rapidité d’exécution, c’est la méthode utilisant les boucles natives (for, while) que nous retenons. On constate en effet que cette méthode est en moyenne 20 % plus rapide que les méthodes fonctionnelles.

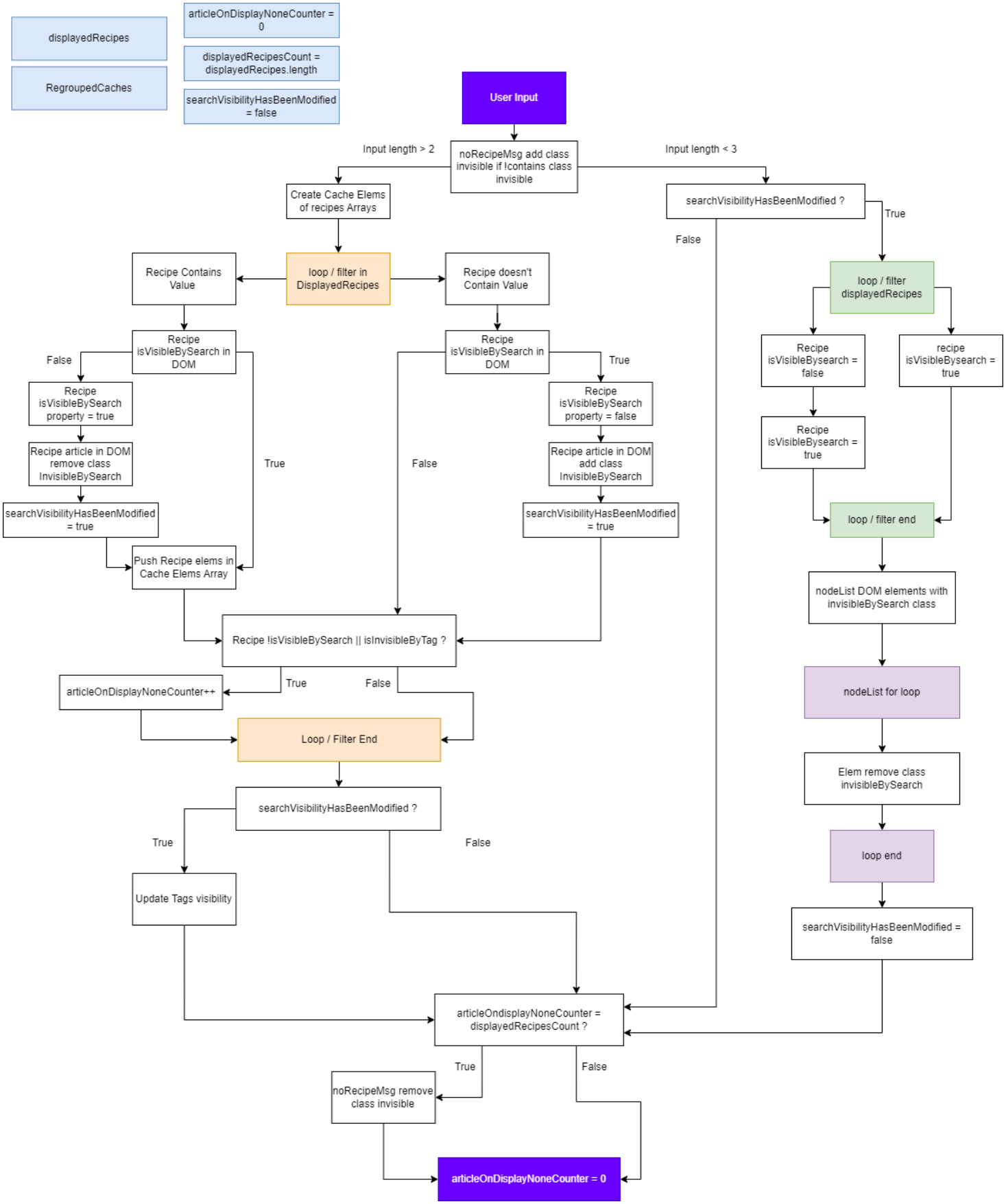

Figure 1 - Diagramme de la recherche Principale