

Fiche d'investigation de fonctionnalités

Fonctionnalité: Recherche	Fonctionnalité #1
Problématique: Afin d'améliorer le confort d'utilisation, nous recherchons une fonction de recherche la plus rapide possible	

Option 1: Fusion des catégories dans lesquelles faire la recherche. Dans cette option, nous créons une variable pour chaque recette qui va inclure le contenu des champs Titre, Ingrédients et Recette, avant de lancer une recherche sur cette variable pour chaque tag.	
Avantages <ul style="list-style-type: none">- Le code est simple à comprendre et maintenir.- Le code devrait être plus performant en utilisant moins de variables	Inconvénients <ul style="list-style-type: none">- Le comportement de la recherche est identique pour Titres, Ingrédients et Recette, et ne peut pas être dissocié.
Nombre de caractères minimum à renseigner pour lancer la recherche: 3 Nombre maximum de critères de recherche: illimité.	


Option 2: Utilisation de boucles pour rechercher séparément chaque catégorie. Dans cette option, nous traitons séparément les champs Titre, Ingrédients et Recette, et lançons pour chacun d'eux une recherche pour chaque tag.	
Avantages <ul style="list-style-type: none">- La boucle va s'interrompre dès qu'une correspondance est trouvée, potentiellement assurant de meilleures performances.- Les recherches dans Titre, Ingrédients et Recette étant faites séparément, il est possible de modifier le code pour avoir différents comportements	Inconvénients <ul style="list-style-type: none">- La logique du code est un peu plus complexe avec une boucle dans une boucle.
Nombre de caractères minimum à renseigner pour lancer la recherche: 3 Nombre maximum de critères de recherche: illimité.	


Solution retenue: Le test de performances ci-dessous a démontré que le code de l'option "Fusions de catégories" est plus performant, même si le gain est de seulement 1.1%. Compte tenu de la nature du site et de l'improbabilité qu'il soit un jour nécessaire de modifier le comportement de la recherche, il s'agit de l'option que nous avons retenue.


Annexes


JSBEN.CH


BENCHMARKBROWSE

no title  (put title and/or keywords here, which describes your test)


Description 

Setup block (useful for function initialization. it will be run before every test, and is not part of the benchmark.) 

Boilerplate block (code will executed before every block and is part of the benchmark. use it for data initializing.) 


code block 1 

```
5 let count = 0;
6 cards.forEach(card => {
7   const cardName = card.querySelector('h2').text;
8   const cardDescription = card.querySelector('p').text;
9   const cardIngredients = Array.from(card.querySelectorAll('p')).map(p => p.textContent);
10  const cardContent = [cardName, cardDescription, cardIngredients];
11  const matchesAnyTag = tags.some(tag => cardContent.includes(tag));
12  card.style.display = matchesAnyTag ? 'block' : 'none';
13  if (card.style.display !== 'none') {
14    count++;
15  }
16 });
17 updateCountDisplay(count);
18 }
```

code block 2 

```
18 ingredient.includes(tag) ?
19   isVisible = true;
20 }
```

result

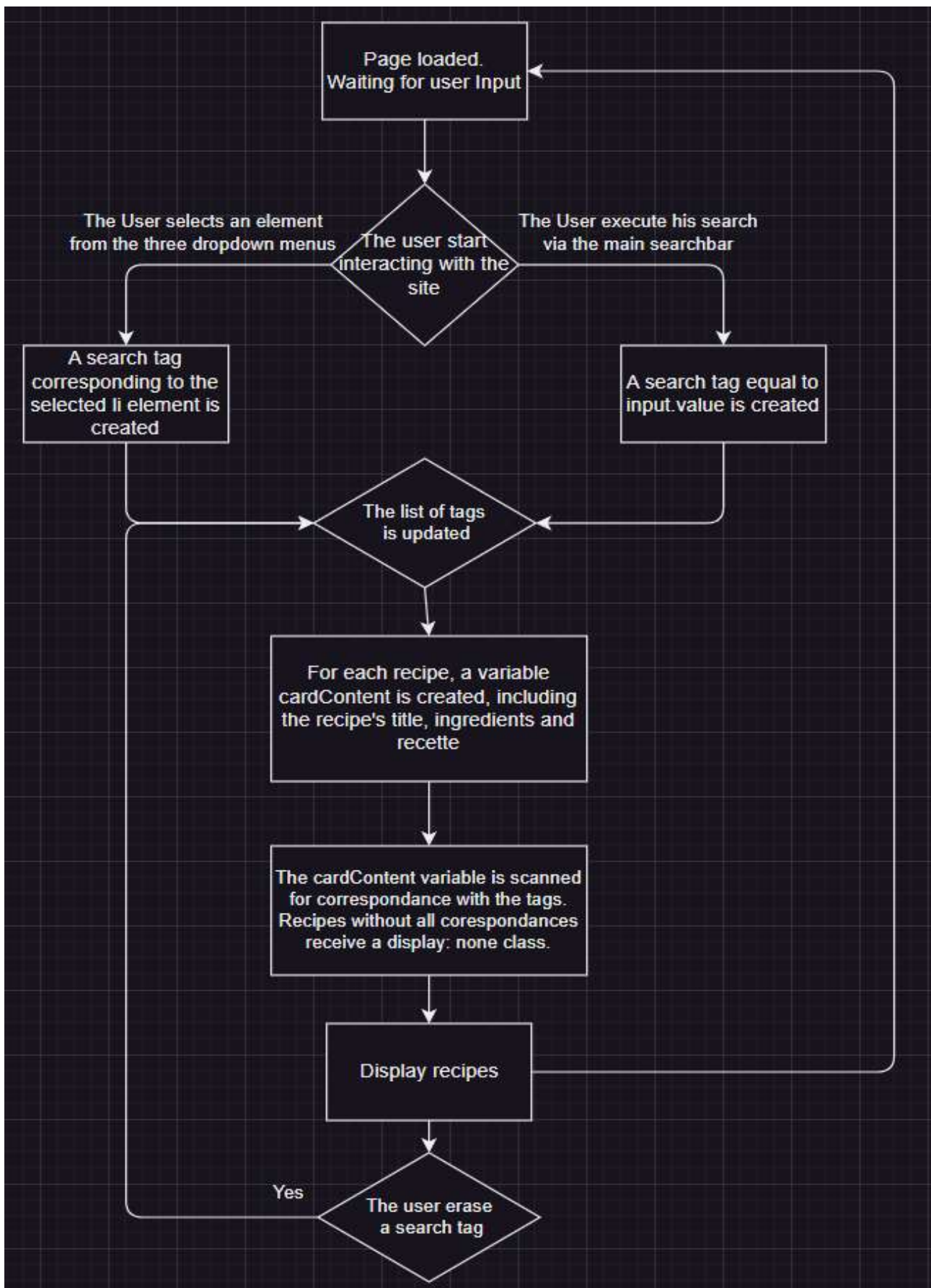
code block 1 (16090292) 

100%

code block 2 (15912927)

98.9%

Option 1:



Option 2:

