

Fiche d'investigation de fonctionnalité

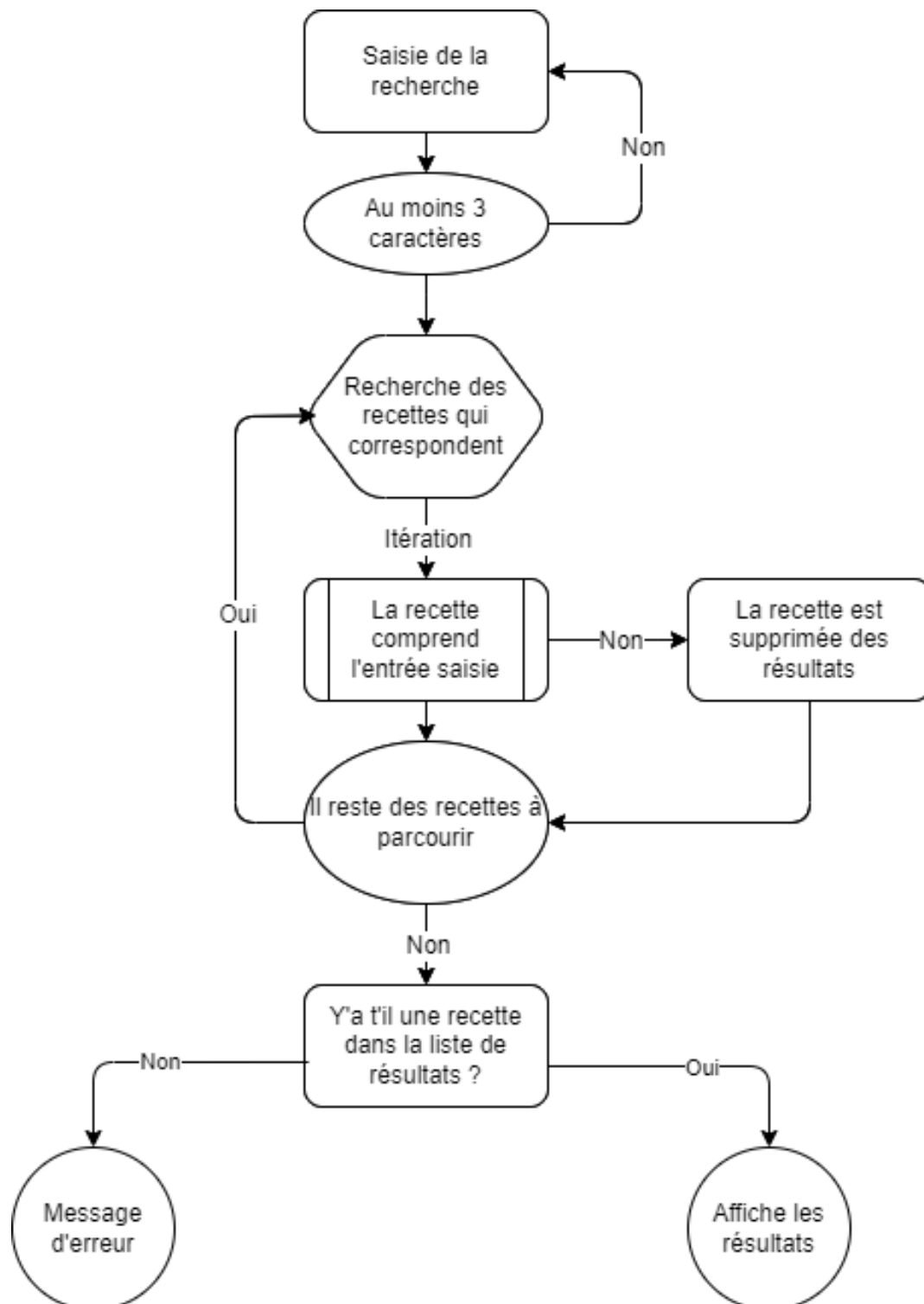
Fonctionnalité : Recherche principale	Fonctionnalité #1
Problématique : Afin de répondre aux besoins des utilisateurs, la recherche doit être la plus rapide possible.	

Option 1 : Programmation fonctionnelle (<i>Annexe 1</i>) Cette option utilise les méthodes de l'objet array (foreach, filter, map, reduce)	
Avantages : <ul style="list-style-type: none">• Code plus compréhensible• Peu de changements en cas de nouveaux filtres	Inconvénients : <ul style="list-style-type: none">• Incompatibilité avec certains navigateurs

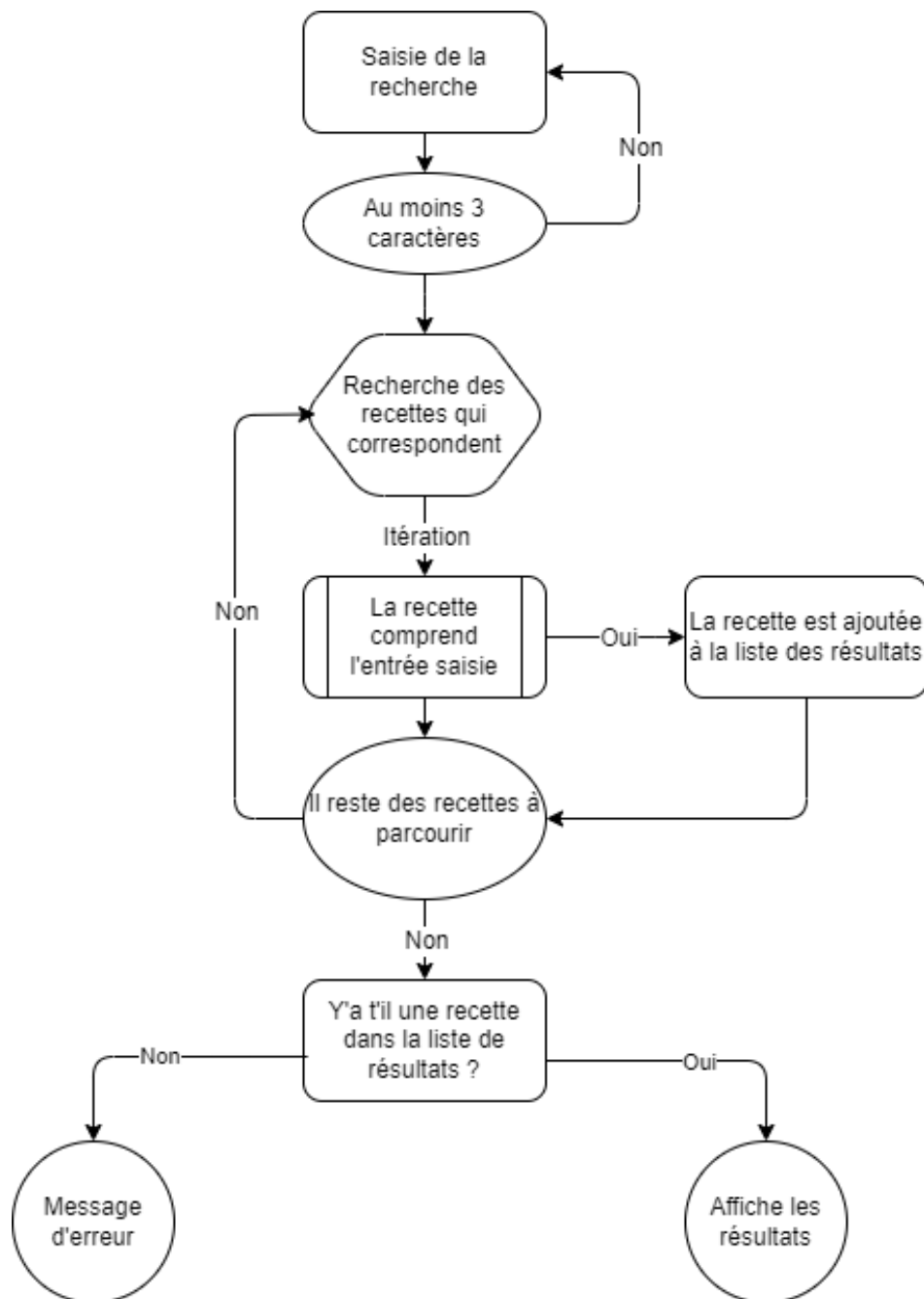
Option 2 : Boucles natives (<i>Annexe 2</i>) Cette option utilise les boucles while et for	
Avantages : <ul style="list-style-type: none">• Compatibilité avec tous les navigateurs	Inconvénients : <ul style="list-style-type: none">• Code plus complexe, moins logique• Beaucoup de changements en cas de nouveaux filtres

Solution retenue : Nous avons donc retenu l'approche fonctionnelle. Suite aux tests réalisés (<i>Annexe 3</i>), nous avons pu constater que la programmation fonctionnelle est plus rapide que les boucles natives. De plus, la mise en place de nouveaux filtres étant simplifiée, maintenir le site sera facilité en cas de mises à jour.

Annexe 1 : Programmation fonctionnelle



Annexe 2 : Boucles natives



Annexe 3 : Rapport de performances



Please login/register to save & publish tests

+ Setup HTML - click to add setup HTML

+ Setup JS - click to add setup JavaScript

Main

finished

6280270.17 ops/s \pm 2.71%

Fastest

```
//Affichage des recettes sur la page
function getRecipes(recipes) {
  recipeArray = recipes.map((recipe) => {
    // Remplit recipeArray avec toutes les données nécessaires
    return {
      recipe: recipe,
      ingredients: recipe.ingredients.map(ingredientConstructor),
      appliance: recipe.appliance,
      ustensils: recipe.ustensils,
      html: `
        <div class="recipe-card" data-id="${recipe.id}">
```

Algo 2

finished

5033146.64 ops/s \pm 0.2%

19.86 % slower

```
//Affichage des recettes sur la page
function getRecipes(recipes) {
  recipeArray = [];
  for (let i = 0; i < recipes.length; i++) {
    const recipe = recipes[i];
    recipeArray.push({
      recipe: recipe,
      ingredients: ingredientConstructor(recipe.ingredients),
      appliance: recipe.appliance,
      ustensils: recipe.ustensils,
      html: `
```

+ Test Case - click to add another test case

+ Teardown JS - click to add teardown JavaScript

+ Output (DOM) - click to monitor output (DOM) while test is running

▶ RUN again