Fiche d'investigation de fonctionnalité

Fonctionnalité : Rechercher les recettes	**Fonctionnalité #2**
Problématique : Implémenter un algorithme de recherche des recettes rapide afin de répondre à l'attente des utilisateurs qui souhaitent une recherche quasi-instantanée.	

Option 1 : Boucle native (for)

Cette approche consiste à parcourir manuellement les données une à une et de sélectionner les données selon les critères de recherche.

Avantages	**Inconvénients**
⊕ Contrôle complet sur les conditions et la logique de recherche ⊕ Plus flexible	⊖ Code plus long et difficile à maintenir ⊖ Plus de risques d'erreur liés à la manipulation manuelle des boucles

Performance JSBEN.CH: Boucles natives for (3758581) - 98.64%

Option 2 : Programmation fonctionnelle

Cette approche utilise les méthodes de l'objet array et de manipulation de tableaux tels que filter, map, every et includes. Cette option utilise donc les méthodes natives de Javascript.

Avantages	**Inconvénients**
⊕ Code plus concis et lisible (Clean Code + Green IT) ⊕ Facilite l'ajout ou la modification des critères de recherche.	⊖ Moins bon contrôle, cadre rigide.

Performance JSBEN.CH: Objet Array (3810487) - 100%

Solution retenue :
La performance et la rapidité d'exécution sont les critères de sélection prioritaires.
Après analyse, la programmation fonctionnelle (option 2) sera retenue pour sa capacité à gérer rapidement et simplement les recherches tout en gardant un code clair et concis. Cette option est plus en accord avec les principes Green IT et Clean Code.
La première option des boucles for, quoique entièrement personnalisable et presque aussi performante, est plus longue et pourrait être plus difficilement adaptable aux futurs évolution du site.

Annexes

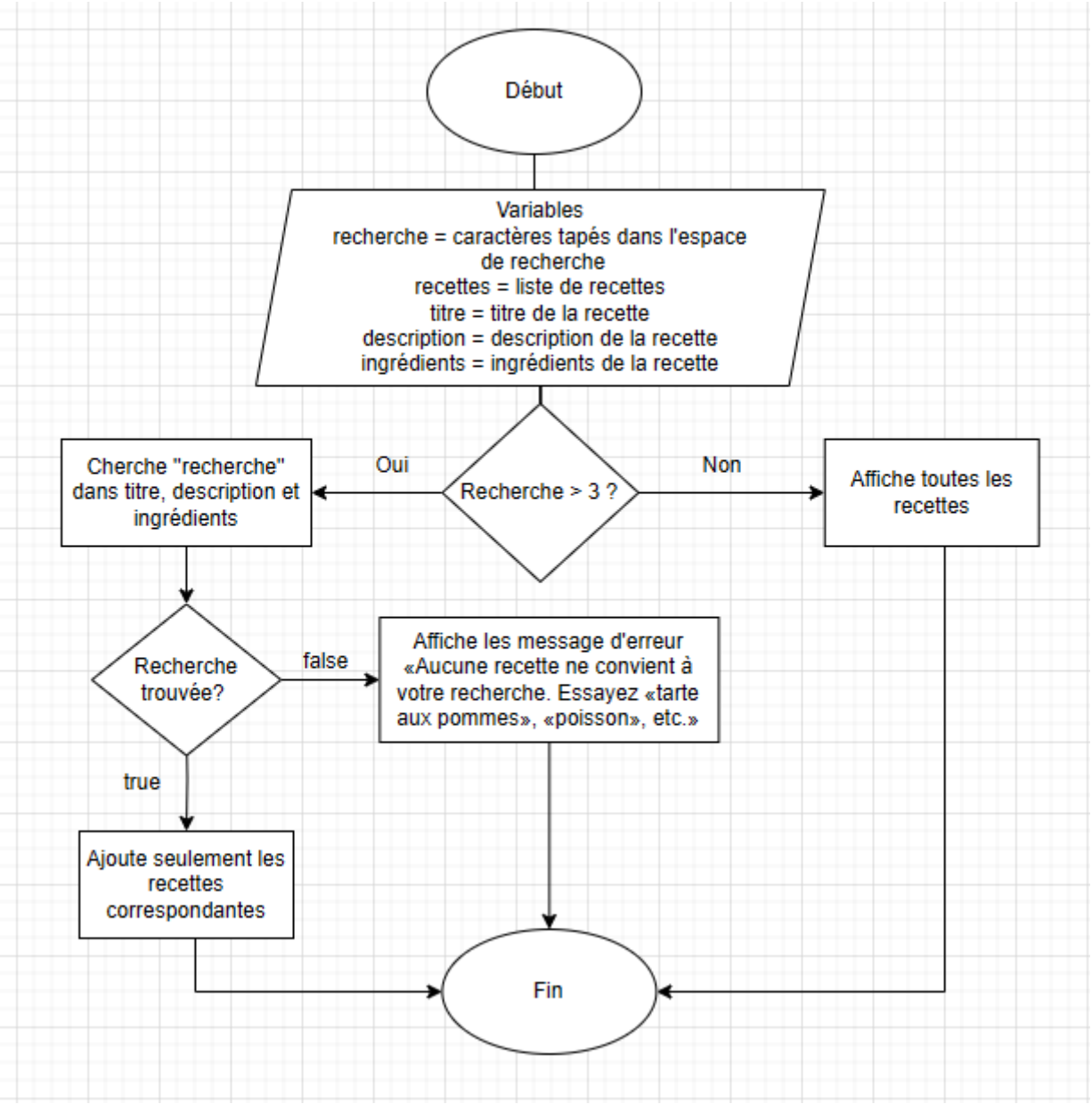

Figure 1 - Algorigramme de recherche option 1

Annexes

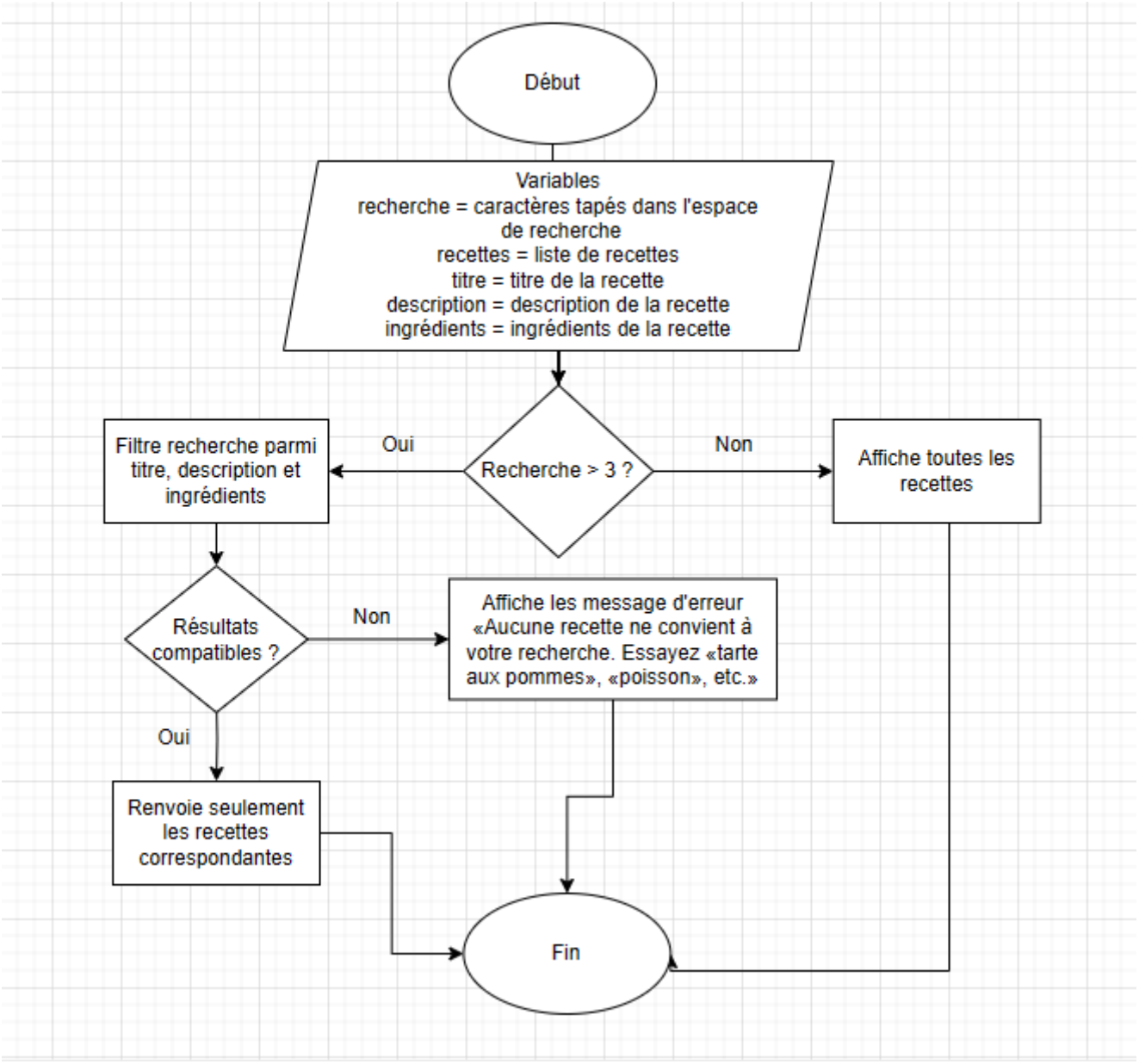

Figure 1 - Algorigramme de recherche option 2