Fiche d'investigation de fonctionnalité

Fonctionnalité : Filtrage de recettes	**Fonctionnalité #1**
Problématique : Aaméliorer l'expérience utilisateur et de permettre aux utilisateurs de trouver plus facilement des recettes correspondant à leurs critères, nous souhaitons mettre en place une fonctionnalité de filtrage.	

Option 1 : Filtrage des Recettes avec methode Filter

Description de la Fonction : La fonction filterRecipesByText est responsable de filtrer les recettes en fonction du texte saisi par l'utilisateur en utilisant la méthode filter. Elle prend en entrée une liste de recettes ainsi que la valeur du texte saisie par l'utilisateur, puis filtre les recettes qui correspondent au texte saisi.

Avantages	**Inconvénients**
Utilise la méthode filter, une fonctionnalité intégrée de JavaScript, pour filtrer efficacement les éléments d'un tableau en fonction d'un critère. Offre une manière efficace de rechercher des recettes en fonction de critères textuels spécifiques. La normalisation du texte d'entrée assure une comparaison plus précise et insensible à la casse.La mise à jour des recettes actuelles et de l'affichage permet une expérience utilisateur fluide et réactive.	⊖ Nécessite une maintenance régulière pour s'assurer que la fonction de filtrage reste efficace avec l'évolution des données. La normalisation du texte peut entraîner une perte d'information dans certains cas, comme la distinction entre majuscules et minuscules.

Exigences: La méthode filter est utilisée de manière à assurer des temps de réponse rapides, même avec de grandes quantités de données.
Les fonctions de filtrage utilisent des algorithmes efficaces pour éviter les opérations coûteuses en temps.
Les itérations et les opérations de filtrage sont réalisées de manière à minimiser le temps de traitement et à maximiser l'efficacité de la méthode filter.

Option 2 : Utilisation des boucles 'for'

Description de la Fonction :
La fonction filterRecipesByText est responsable de filtrer les recettes en fonction du texte saisi par l'utilisateur en utilisant une boucle for. Elle prend en entrée une liste de recettes ainsi que la valeur du texte saisie par l'utilisateur, puis filtre les recettes qui correspondent au texte saisi.

Avantages	**Inconvénients**
-Utilisation de la boucle for, une structure de contrôle de base de JavaScript, pour parcourir efficacement les éléments d'un tableau et filtrer les recettes en fonction du texte saisi. Offre une manière efficace de rechercher des recettes en fonction de critères textuels spécifiques. La normalisation du texte d'entrée assure une comparaison plus précise et insensible à la casse. La mise à jour des recettes actuelles et de l'affichage permet une expérience utilisateur fluide et réactive.	Nécessite une maintenance régulière pour s'assurer que la fonction de filtrage reste efficace avec l'évolution des données. La normalisation du texte peut entraîner une perte d'information dans certains cas, comme la distinction entre majuscules et minuscules.

Exigences:
La boucle for est configurée de manière à garantir des temps de réponse rapides, même avec de grandes quantités de données.
Les algorithmes utilisés dans la boucle for sont optimisés pour minimiser la complexité temporelle et les opérations coûteuses en temps.
Les itérations sont réalisées de manière à minimiser le temps de traitement et à maximiser l'efficacité de la boucle for pour le filtrage des données.

Solution retenue :

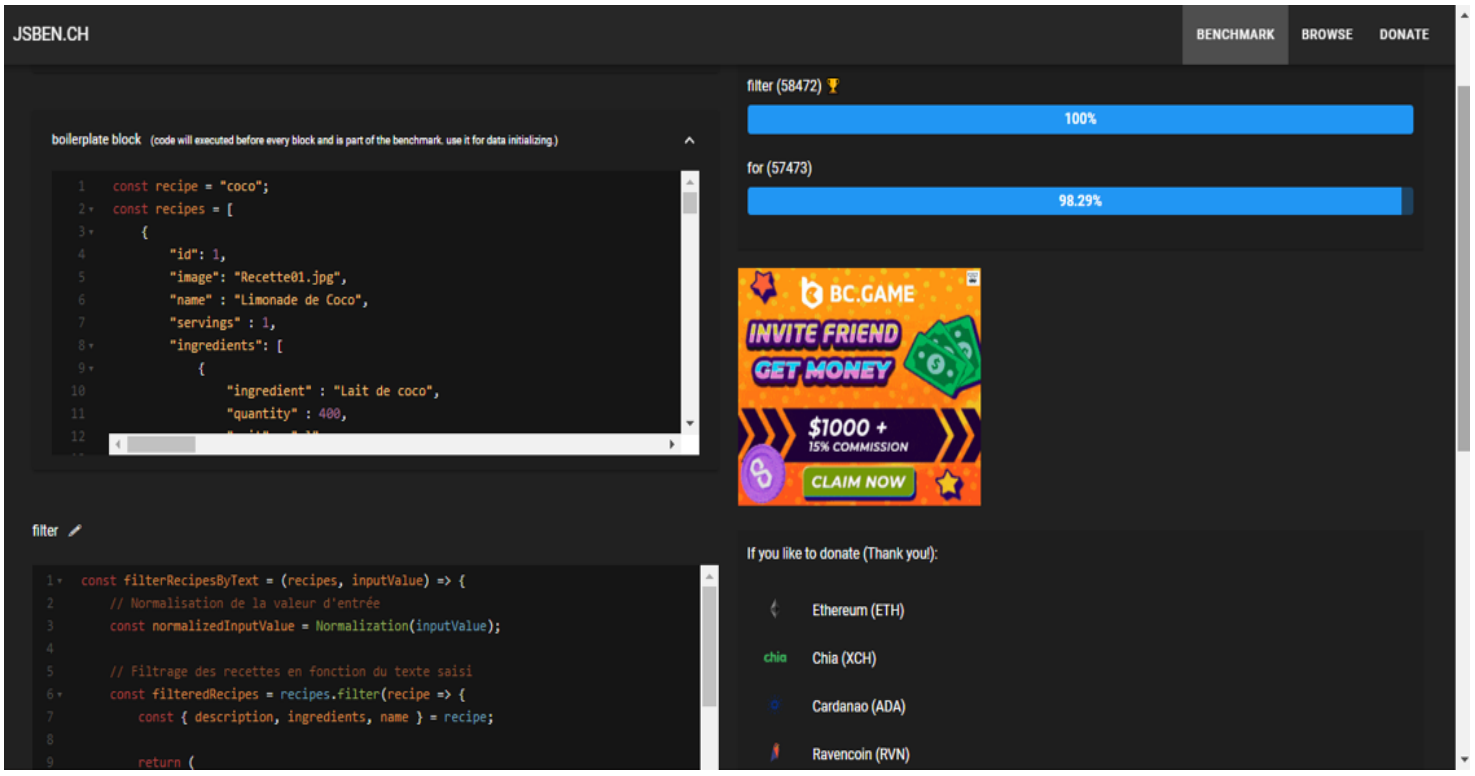
Filtrage par Texte avec la méthode filter

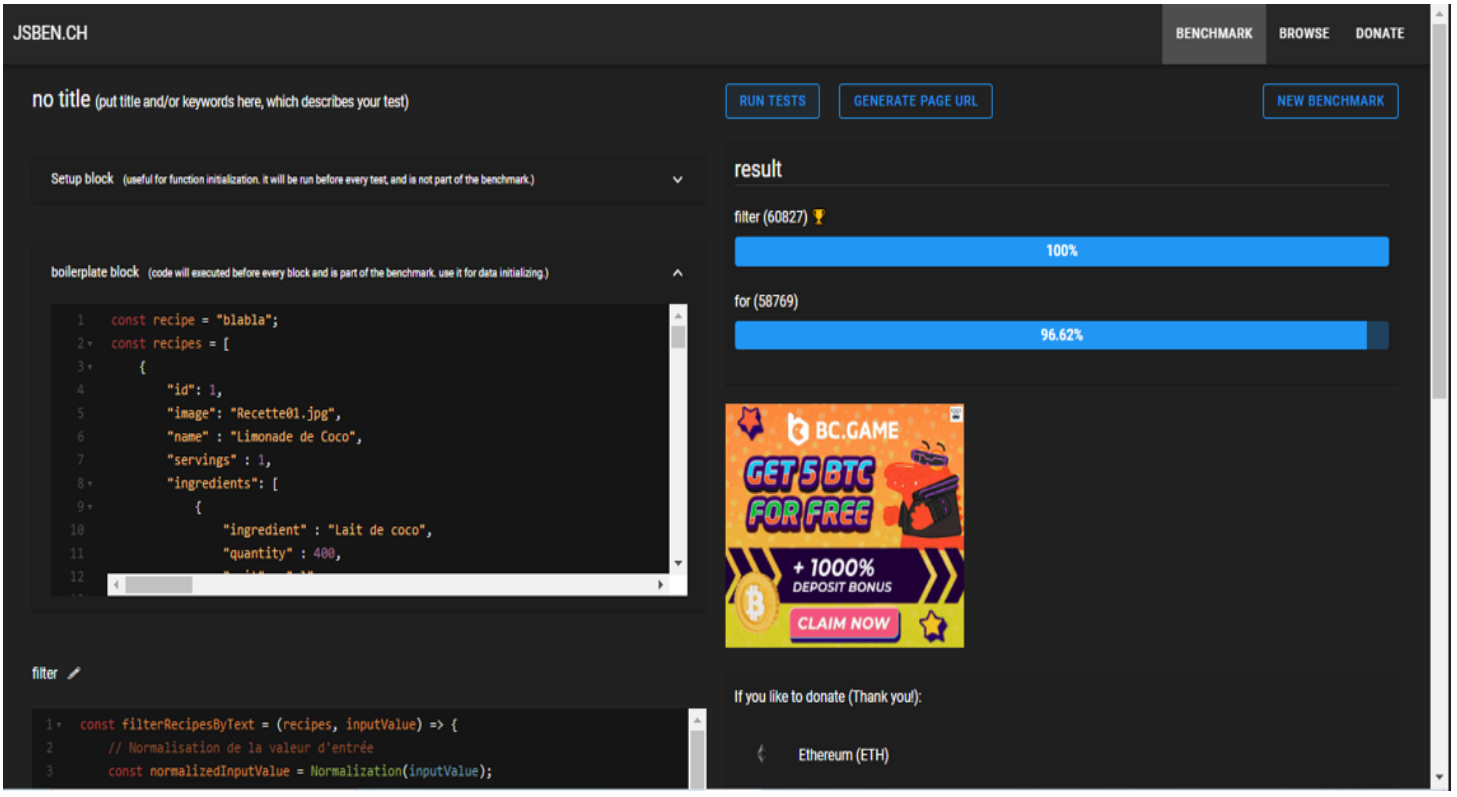
Après évaluation des différentes options, la solution retenue pour filtrer les recettes en fonction du texte saisi par l'utilisateur est l'utilisation de la méthode filter. Cette approche offre une manière efficace et concise de filtrer les éléments d'un tableau en fonction d'un critère spécifique, en l'occurrence, le texte saisi par l'utilisateur.

En utilisant la méthode filter, nous pouvons directement filtrer la liste des recettes en fonction du texte saisi, sans avoir besoin d'une boucle for explicite. Cette méthode offre une syntaxe plus claire et concise, ce qui rend le code plus lisible et plus facile à comprendre.

De plus, la méthode filter est une fonctionnalité intégrée de JavaScript, largement utilisée et bien prise en charge par les navigateurs, ce qui garantit une compatibilité élevée et des performances optimales.

En résumé, l'utilisation de la méthode filter pour filtrer les recettes par texte offre une solution élégante, efficace et réactive qui répond aux besoins des utilisateurs tout en maintenant des performances optimales et une expérience utilisateur fluide.


JSBEN.CH
BENCHMARK
BROWSE
DONATE
boilerplate block (code will executed before every block and is part of the benchmark. use it for data initializing.)
const recipe = "coco";
const recipes = [
{
"id": 1,
"image": "Recette01.jpg",
"name" : "Limonade de Coco",
"servings" : 1,
"ingredients": [
{
"ingredient" : "Lait de coco",
"quantity" : 400,
filter
const filterRecipesByText = (recipes, inputValue) => {
// Normalisation de la valeur d'entrée
const normalizedInputValue = Normalization(inputValue);
// Filtrage des recettes en fonction du texte saisi
const filteredRecipes = recipes.filter(recipe => {
const { description, ingredients, name } = recipe;
return (
filter (58472)
100%
for (57473)
98.29%
BC.GAME
INVITE FRIEND
GET MONEY
$1000 +
15% COMMISSION
CLAIM NOW
If you like to donate (Thank you!):
Ethereum (ETH)
Chia (XCH)
Cardanao (ADA)
Ravencoin (RVN)


JSBEN.CH
BENCHMARK
BROWSE
DONATE
no title (put title and/or keywords here, which describes your test)
RUN TESTS
GENERATE PAGE URL
NEW BENCHMARK
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.)
const recipe = "blabla";
const recipes = [
{
"id": 1,
"image": "Recette01.jpg",
"name" : "Limonade de Coco",
"servings" : 1,
"ingredients": [
{
"ingredient" : "Lait de coco",
"quantity" : 400,
filter
const filterRecipesByText = (recipes, inputValue) => {
// Normalisation de la valeur d'entrée
const normalizedInputValue = Normalization(inputValue);
result
filter (60827)
100%
for (58769)
96.62%
BC.GAME
GET 5 BTC
FOR FREE
+ 1000%
DEPOSIT BONUS
CLAIM NOW
If you like to donate (Thank you!):
Ethereum (ETH)

Annexes

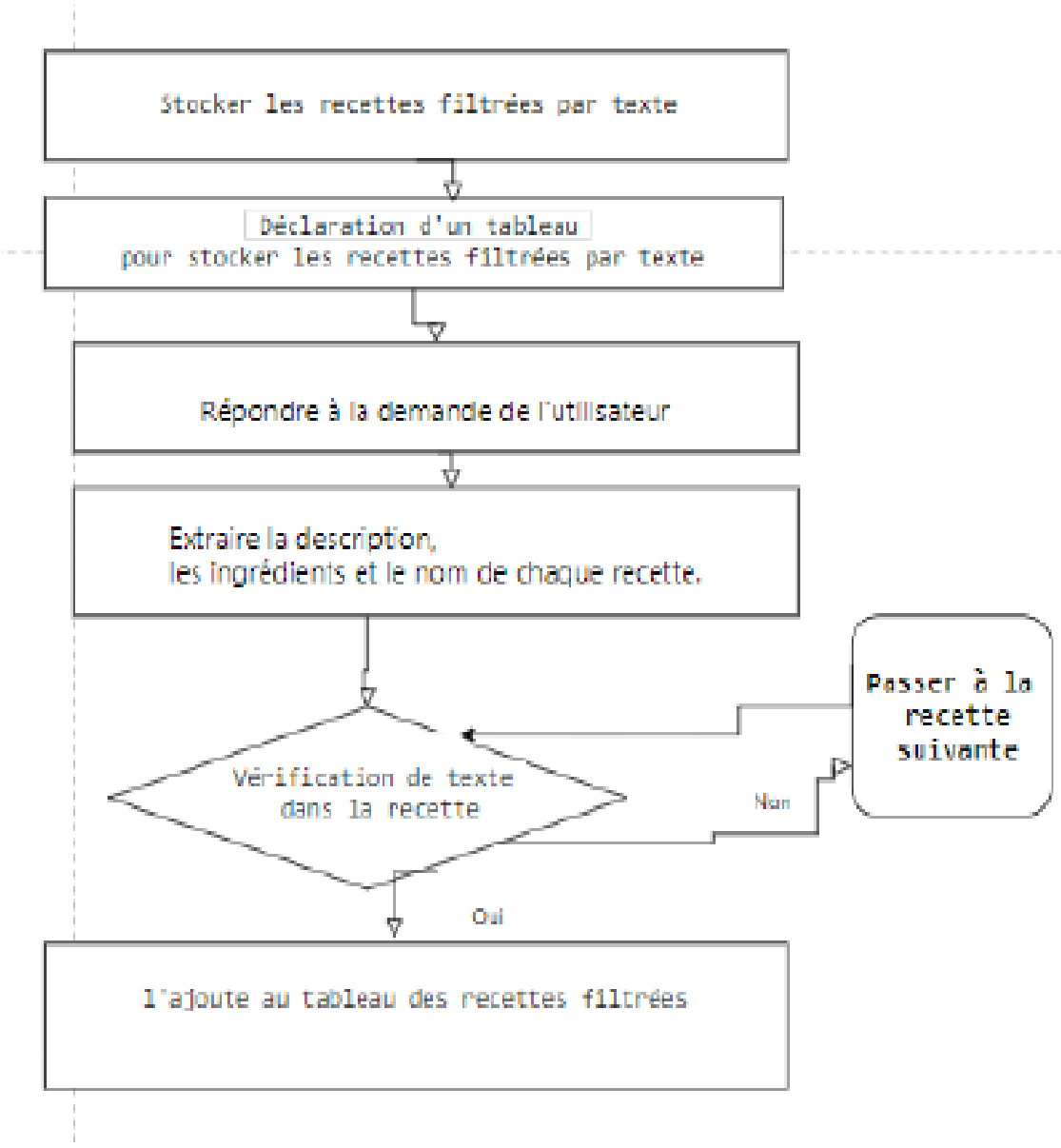


Figure 1: Filtrage des Recettes avec methode Filter

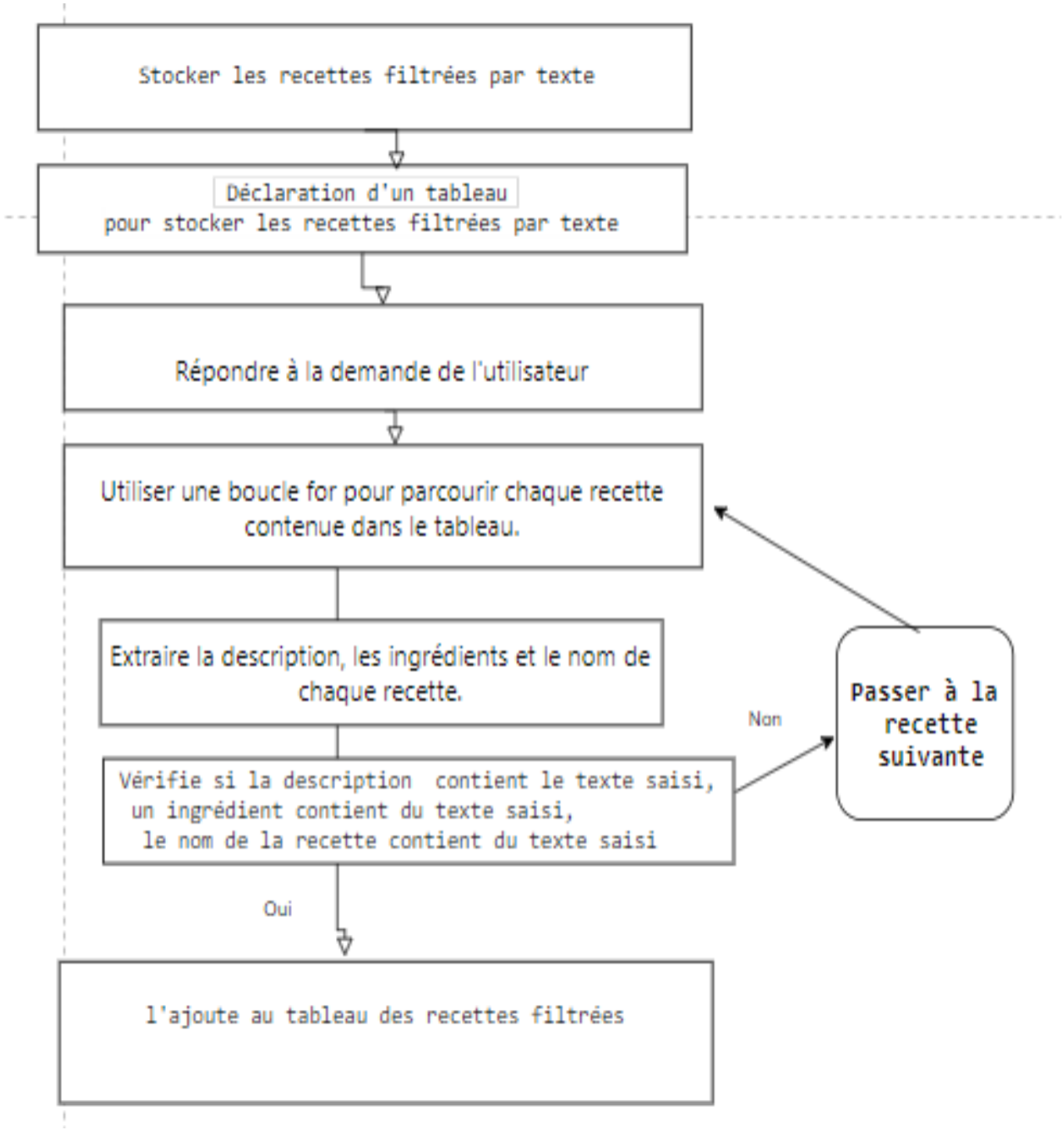


Figure 2 : Utilisation des boucles 'for'