Rapport de Validation et Tests d'une Application de Gestion de Taches

Réalisé par : Dali zaina

2024/2025

Table de matière

Table des figures

Tables des tableaux

1. Introduction

L'application testée est une **application web de gestion des tâches**. Son objectif est de permettre aux utilisateurs d'**ajouter, afficher et supprimer des tâches**. La deuxième version de l'application utilise **LocalStorage** pour conserver les tâches même après un rafraîchissement de la page.

Pour garantir la stabilité et la performance de l'application, plusieurs outils de tests ont été utilisés :

- **PHPUnit** : Assure que la classe TaskManager fonctionne correctement.
- **Selenium IDE** : Automatise les tests d'interface utilisateur pour vérifier les interactions avec les tâches (ajout, suppression).
- **JMeter** : Simule des charges utilisateur pour tester les performances et identifier les éventuels ralentissements sous forte sollicitation.

Ce rapport vise à **valider le bon fonctionnement de l'application** après l'ajout d'une nouvelle fonctionnalité (utilisation de **LocalStorage**). Il décrit les **tests réalisés** et les résultats obtenu.

Vous trouvez touts les codes dans le dépôt git suivant :
https://github.com/ZainaDali/Gestion-des-taches-.git

2. Résultats des Tests

2.1. Tests Fonctionnels (PHPUnit)

Les tests unitaires ont été réalisés avec **PHPUnit**, un framework de test pour PHP. Ils permettent de **vérifier individuellement les fonctionnalités** de la classe TaskManager.

Chaque test a été exécuté via la commande suivante :

```
vendor/bin/phpunit --testdox tests
```

Le tableau suivant illustre les tests fonctionnels couvrent les opérations principales

Tests	Description	Résultat
testAddTask()	Vérifie que l'ajout d'une tâche fonctionne correctement en s'assurant que la liste contient bien 1 élément après l'ajout.	Succès
testRemoveTask()	Vérifie que la suppression d'une tâche fonctionne correctement en s'assurant que la tâche supprimée n'est plus dans la liste et que les autres tâches sont bien réindexées.	Succès
testGetTasks()	Vérifie que getTasks() retourne bien toutes les tâches ajoutées.	Succès
testGetTask()	Vérifie que getTask() retourne bien une tâche spécifique en fonction de son index.	Succès
testRemoveInvalidIndexThrowsException()	Vérifie que la tentative de suppression d'une tâche avec un index invalide	Succès

	génère une exception OutOfBoundsException.	
testGetInvalidIndexThrowsException()	Vérifie que la tentative de récupération d'une tâche avec un index invalide génère une exception OutOfBoundsException.	Succès
testTaskOrderAfterRemoval()	Vérifie que l'ordre des tâches restantes est bien conservé après suppression d'une tâche intermédiaire.	Succès

Tableau 1 : Tests fonctionnels couvrant la classe TaskManager

La figure suivante montre l'exécution de notre test

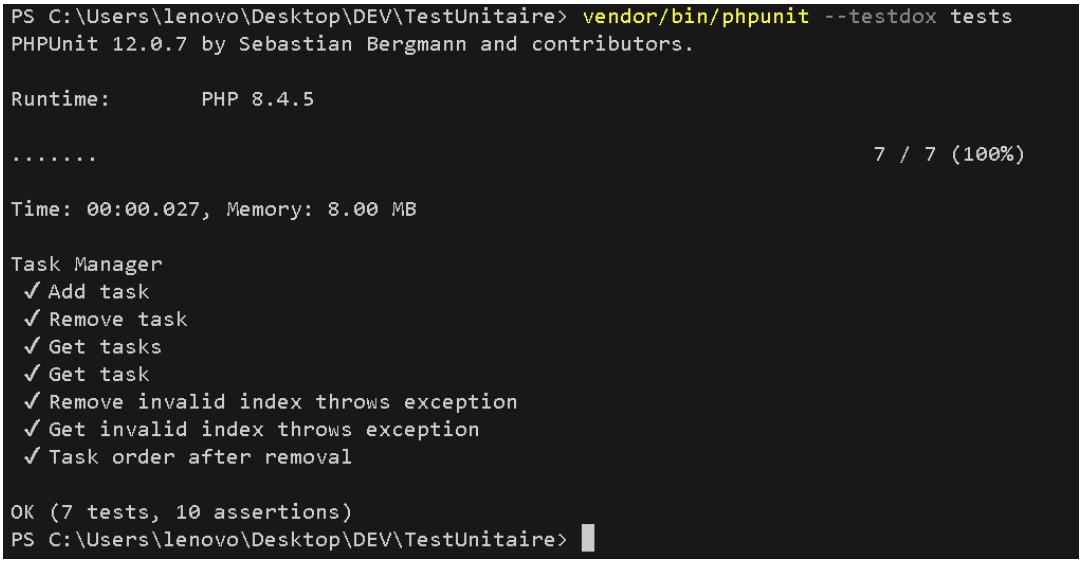

Figure 1 : exécution de test TaskManagerTest.php

2.2. Tests End-to-End (E2E) avec Selenium

Les tests **End-to-End (E2E)** ont été réalisés avec **Selenium IDE** pour vérifier que **l'ensemble du processus utilisateur fonctionne correctement**.
L'objectif est d'automatiser les scénarios réels afin de s'assurer que les fonctionnalités principales de l'application sont opérationnelles.

Le tableau suivant illustre les tests E2E

Scénario testé	Description	Resultat
Ajouter une tâche	Vérifier que l'ajout d'une tâche fonctionne correctement et qu'elle s'affiche dans la liste.	Succès
Supprimer une tâche	Vérifier qu'une tâche peut être supprimée et qu'elle ne s'affiche plus après l'action.	Succès
Persistance des tâches	Vérifier que les tâches et leurs durées restent disponibles après rechargement.	Echec

Tableau 2 : Les tests End-to-End (E2E

Les figures ci-dessous illustre l'exécution des tests

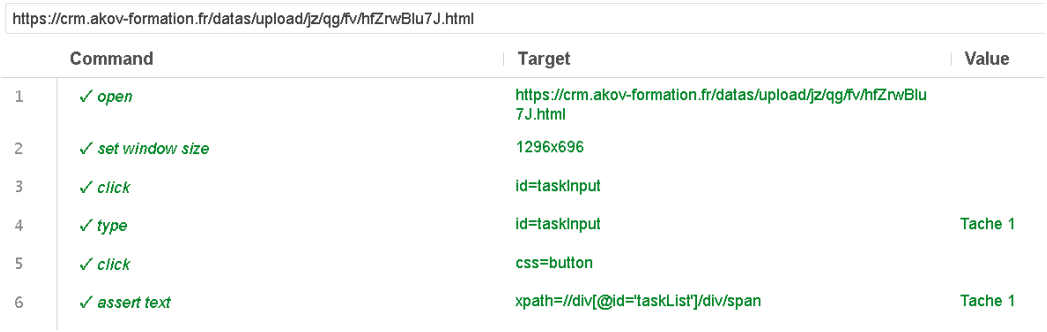

Figure 2 : exécution de test 'Ajout d'une tache'

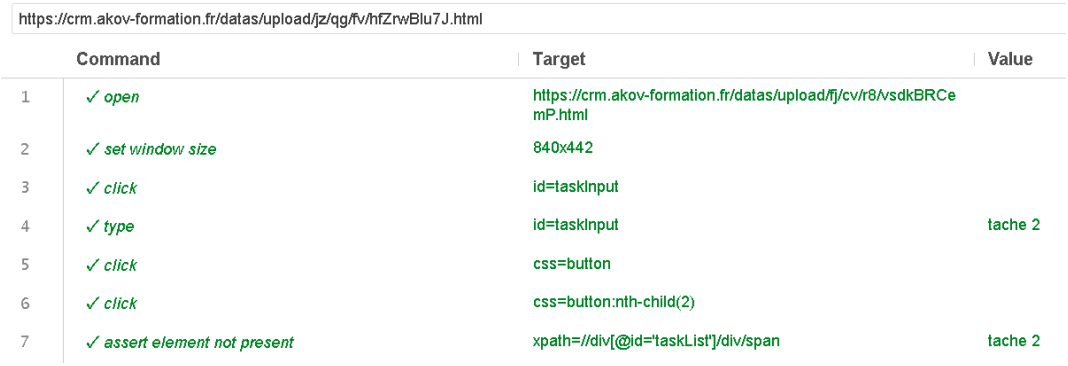

Figure 3 : exécution de test 'suppression d'une tache'

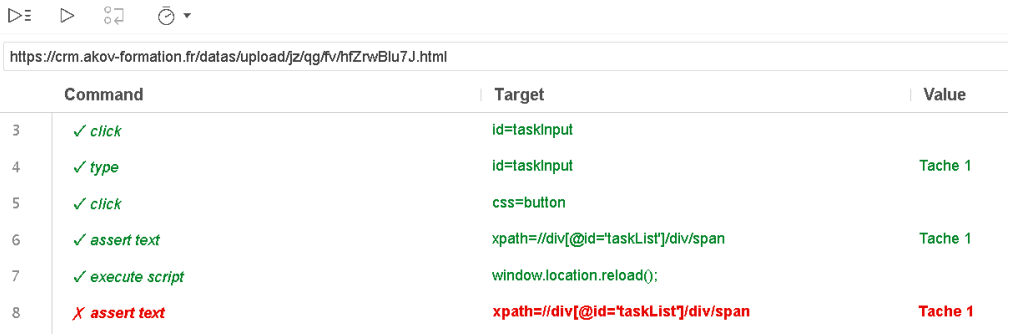

Figure 4 : exécution de test 'Ajout d'une tache et rechargement '

2.3. Tests de non-Régression

Les tests de **non-régression** ont pour but de **s'assurer que les fonctionnalités existantes** de l'application **ne sont pas impactées** après l'ajout d'une nouvelle fonctionnalité.
Dans ce cas, après l'utilisation de LocalStorage

Le tableau suivant illustre les tests E2E

Scénario testé	Description	Resultat
Ajouter une tâche	Vérifier que l'ajout d'une tâche fonctionne correctement et qu'elle s'affiche dans la liste.	Succès
Supprimer une tâche	Vérifier qu'une tâche peut être supprimée et qu'elle ne s'affiche plus après l'action.	Succès
Persistance des tâches	Vérifier que les tâches et leurs durées restent disponibles après rechargement.	Succès

Tableau 3 : Les tests End-to-End (E2E

Les figures ci-dessous illustre l'exécution des tests

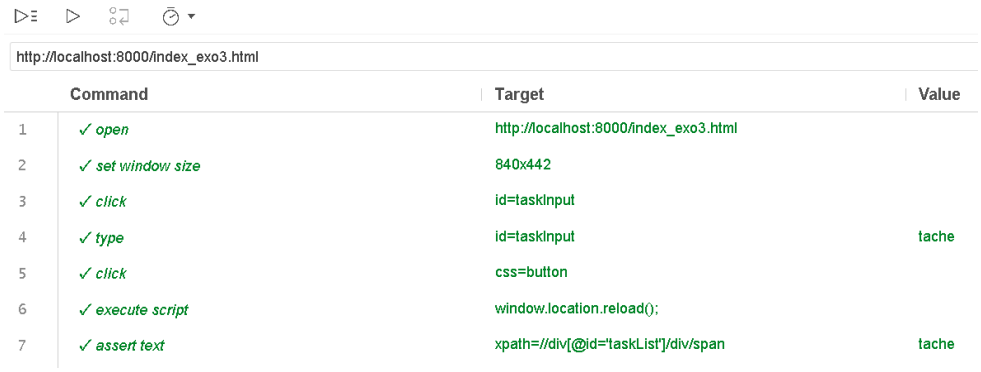

Figure 5 : exécution de test 'Ajout d'une tache et rechargement'

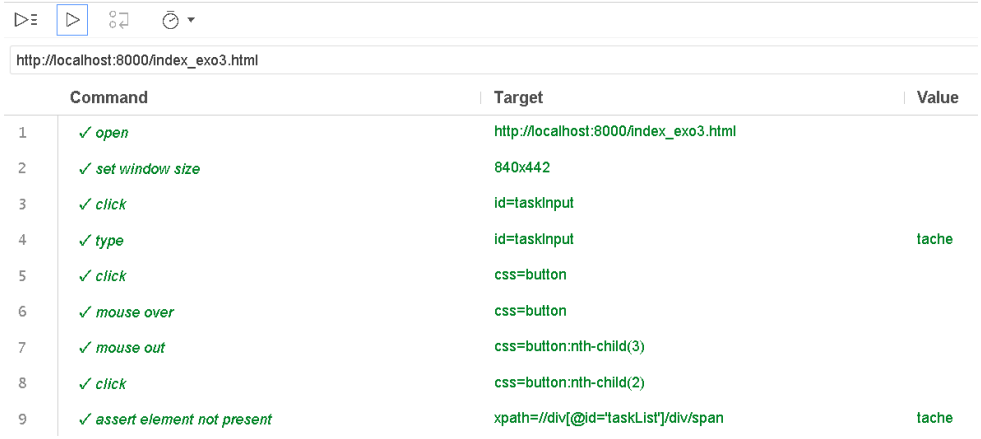

Figure 6 : exécution de test 'suppression d'une tache'

2.4. Tests de Performance avec JMeter

Les tests de performance ont été réalisés avec **JMeter** afin d'évaluer la **capacité de l'application** à gérer plusieurs utilisateurs simultanés.

Dans ce test on fait une Simulation de 100 utilisateurs qui charge notre application et le tableau suivant illustre le résultat

Nombre d'utilisateurs	100
Type de requêtes simulées	GET
Durée du test	2min13s

Tableau 4 : résultat de lancement des requêtes

Ci-dessous des figures montrent montre le **rapport de performance et les graphes de résultats généré par JMeter**

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB...	Sent KB/sec	Avg. Bytes
HTTP Request	100000	107	1	5079	256,28	79,14%	752,3/sec	1816,12	20,53	2472,1
TOTAL	100000	107	1	5079	256,28	79,14%	752,3/sec	1816,12	20,53	2472,1

Figure 7 : Summary Report

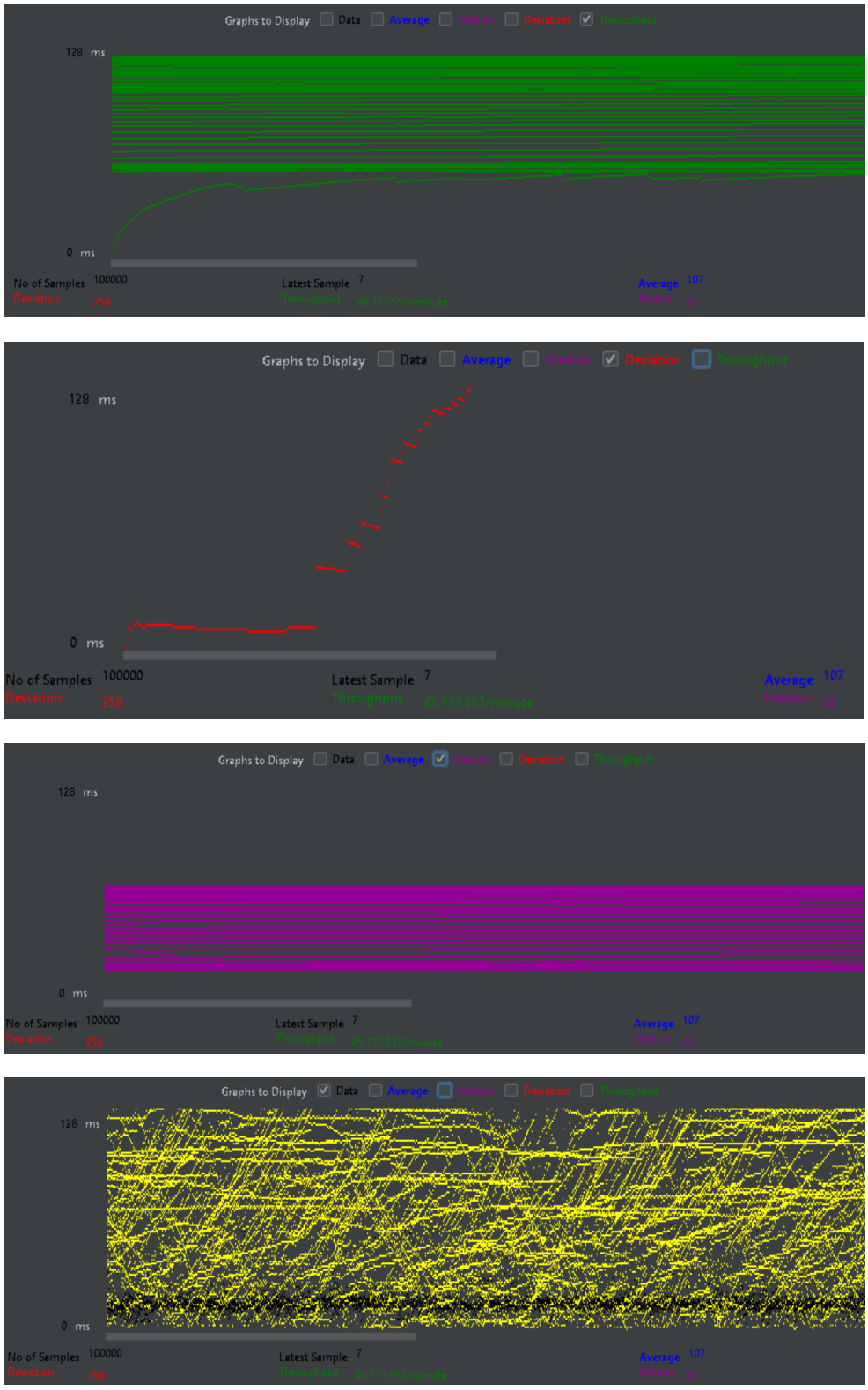

Figure 8 : Graph Results

3. Conclusion

Le projet de gestion des tâches a été soumis à une analyse à travers une série de tests visant à garantir son bon fonctionnement, sa stabilité et ses performances. L'objectif principal était d'assurer que toutes les fonctionnalités essentielles – **ajout et suppression des tâches** – opèrent correctement, même après l'introduction de nouvelles fonctions.

Les **tests fonctionnels réalisés avec PHPUnit** ont permis de valider le comportement interne du gestionnaire de tâches en s'assurant que chaque action produisait bien le résultat attendu. Ensuite, les **tests End-to-End (E2E) avec Selenium** ont simulé des interactions utilisateur afin de vérifier le bon fonctionnement de l'interface web. Les **tests de non-régression** ont confirmé que les modifications apportées n'ont pas impacté les fonctionnalités existantes. Enfin, les **tests de performance avec JMeter** avaient pour objectif d'évaluer **la capacité de l'application à gérer un grand nombre d'utilisateurs simultanés** et à mesurer **la rapidité de réponse.**