

# Rapport de validation et tests de l'application de gestion d'utilisateurs

## 1. Introduction

L'application évaluée est une plateforme de gestion des utilisateurs permettant d'ajouter, modifier et supprimer des comptes. L'objectif principal est de garantir la fiabilité et le bon fonctionnement du système à travers divers tests automatisés.

Les outils utilisés pour ces test sont :

- **PHPUnit** pour effectuer les tests unitaires et fonctionnels
- **Selenium** pour l'automatisation des tests d'interface utilisateur
- **k6** pour les tests de performance et de charge

Ce rapport présente l'ensemble des tests effectués, leurs résultats, ainsi que les éventuelles régressions et axes d'amélioration.

## 2. Résultats des tests

### 2.1. Tests fonctionnels (PHPUnit)

Les tests unitaires et fonctionnels ont été réalisés avec PHPUnit afin de vérifier les principales fonctionnalités de l'application.

Tests	Results
testAddUser()	Success
testAddUserEmailException()	Success
testUpdateUser()	Success
testRemoveUser()	Success
testGetUsers()	Success
testInvalidUpdateThrowsException()	Success
testInvalidRemoveThrowsException()	

Tableau 1 - Résultats des tests PHPUnit

```
C:\wamp64\bin\php\php8.2.18\php.exe "C:/Users/WilliamLG/Desktop/Perso/Cours/Efrei/Tests unitaires  
Testing started at 11:04 ...  
PHPUnit 11.5.14 by Sebastian Bergmann and contributors.  
  
Runtime:      PHP 8.2.18  
Time: 00:00.080, Memory: 8.00 MB  
  
OK (7 tests, 11 assertions)  
  
Process finished with exit code 0
```

Figure 1 - Validation des tests PHPUnit

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

Les tests E2E ont été réalisés pour simuler des interactions utilisateur complètes.

Voici les scénarios utilisateur qui ont été testés :

- Ajout d'un utilisateur et vérification de son apparition dans la liste des utilisateurs
- Modifications des informations d'un utilisateur
- Suppression d'un utilisateur et vérification de sa disparition

Pour effectuer ces tests avec Selenium, voici les étapes :

### Lancement du navigateur

- Le script Selenium ouvre une fenêtre du navigateur internet et se rend sur <http://localhost:8000/>

### Test d'ajout d'un utilisateur

- Cliquer sur le champ Nom (input avec id=name) et saisir le nom du nouvel utilisateur.
- Cliquer sur le champ Email (input avec id=email) et saisir l'adresse email du nouvel utilisateur.
- Cliquer sur le bouton Ajouter (input de type submit).
- Récupérer le texte de l'élément de la liste <ul id="userList"> et s'assurer qu'il contient le nom de l'utilisateur et l'email qui été saisis.

### Test de modification d'un utilisateur

- Ajouter un utilisateur comme pour le test précédent avec un email différent.
- Cliquer sur le bouton d'édition du nouvel utilisateur (✎).
- Remplacer le nom de l'utilisateur par un autre nom dans le champ Nom.
- Remplacer (ou non) l'email de l'utilisateur dans le champ Email.

## LE GAVRIAN

- Cliquer sur le bouton Ajouter (input de type submit)
- Récupérer le texte de l'élément de la liste <ul id="userList"> qui a été modifié et vérifier que les nouvelles informations sont correctes

## Test de suppression d'un utilisateur

- Ajouter un utilisateur comme pour les étapes précédentes avec un email différent.
- Cliquer sur le bouton de suppression du nouvel utilisateur (X).
- Vérifier que l'utilisateur ne se trouve plus dans la liste.

## Fermeture du navigateur

- Le script Selenium ferme la session à la fin du test.

Tests	Results
Ajout d'un utilisateur et vérification de sa présence dans la liste	Success
Modification d'un utilisateur	Success
Suppression d'un utilisateur	Success

Tableau 2 - Résultats des tests E2E avec Selenium

Command	Target
open	/
set window size	1552x832
click	id=name
type	id=name
type	id=email
click	css=button
assert text	css=p

Log:

```

Running 'Test Add User'
1. open on / OK
2. setWindowSize on 1552x832 OK
3. click on id=name OK
4. type on id=name with value William OK
5. type on id=email with value william@test.fr OK
6. click on css=button OK
7. assertText on css=p with value William (william@test.fr) OK
'Test Add User' completed successfully
  
```

Figure 2 - Validation des tests d'ajout de l'utilisateur et la vérification de sa présence dans la liste

## LE GAVRIAN

The screenshot shows a test runner interface with the following details:

- Tests:** A sidebar lists "Test Add User", "Test Remove User", and "Test Update User".
- URL:** http://localhost:8000
- Log:**
  - 4. type on id=name with value William OK
  - 5. click on id=email OK
  - 6. type on id=email with value testUpdate@test.fr OK
  - 7. click on css=button:nth-child(4) OK
  - 8. click on css=li:nth-child(1) button:nth-child(1) OK
  - 9. click on id=name OK
  - 10. type on id=name with value Jean OK
  - 11. click on id=email OK
  - 12. type on id=email with value testUpdate@test.com OK
  - 13. click on css=button:nth-child(4) OK
  - 14. assertText on css=li:nth-child(1) > p with value Jean (testUpdate@test.com) OK
- Message:** "Test Update User' completed successfully"

Figure 3 - Validation de la modification d'un utilisateur

The screenshot shows a test runner interface with the following details:

- Tests:** A sidebar lists "Test Add User", "Test Remove User", and "Test Update User".
- URL:** http://localhost:8000
- Log:**
  - Running "Test Remove User"
  - 1. open on / OK
  - 2. setWindowSize on 1536x816 OK
  - 3. click on id=name OK
  - 4. type on id=name with value William OK
  - 5. click on id=email OK
  - 6. type on id=email with value testRemove@test.fr OK
  - 7. click on css=button:nth-child(4) OK
  - 8. mouseOver on css=button:nth-child(4) OK
  - 9. mouseOut on css=button:nth-child(4) OK
  - 10. click on css=li:nth-child(1) button:nth-child(2) OK
  - 11. assertElementNotPresent on css=li:nth-child(3) OK
- Message:** "Test Remove User' completed successfully"

Figure 4 - Validation du test de suppression d'un utilisateur

## 2.3. Tests de Non-Régression

Une nouvelle fonctionnalité d'ajout de l'âge de l'utilisateur a été ajouté. Les tests de non-régression ont permis de vérifier que cette modification n'affectera pas les fonctionnalités existantes.

Tests	Results
Ajout d'un utilisateur et vérification de son affichage dans la liste	Success
Modification de l'utilisateur	Success
Suppression de l'utilisateur et vérification de son absence dans la liste	Success
Affichage de l'âge de l'utilisateur	Success

Tableau 3 - Résultats des tests de Non-Régression après ajout d'une nouvelle fonctionnalité

Step	Command	Target
3	click	id=name
4	type	id=name
5	type	id=email
6	click	css=button
7	assert text	css=p

Log:

```

Running 'Test Affichage Age'
1. open on / OK
2. setWindowSize on 1936x1048 OK
3. click on id=name OK
4. type on id=name with value Test OK
5. click on id=email OK
6. type on id=email with value test@test.fr OK
7. click on id=age OK
8. type on id=age with value 23 OK
9. click on css=button OK
10. assertText on css=p nth-child(2) with value 23 ans OK
'Test Affichage Age' completed successfully

```

Figure 5 - Résultats des tests de Non-Régression dans Selenium

## 2.4. Tests de Performance avec k6

Un test de charge a été effectué pour mesurer la performance du serveur sous une charge élevée de 500 utilisateurs simultanés ajoutant des comptes.

Métrique	Valeur
Nombre de requêtes envoyées	7275
Temps de réponse moyen	4.11s
Nombre d'erreurs	5.29% (385 requêtes échouées)

Tableau 4 - Résultat des tests de performance

```

× Statut 200
↳ 94% - ✓ 6890 / × 385
× Réponse rapide (<500ms)
↳ 30% - ✓ 2246 / × 5029

checks.....: 62.79% 9136 out of 14550
data_received.....: 15 MB 225 kB/s
data_sent.....: 1.8 MB 28 kB/s
http_req_blocked.....: avg=57.12ms min=0s med=0s max=1.15s p(90)=0s p(95)=614.64ms
http_req_connecting.....: avg=55.86ms min=0s med=0s max=1.14s p(90)=0s p(95)=607.49ms
http_req_duration.....: avg=4.11s min=0s med=613.08ms max=59.86s p(90)=878.55ms p(95)=58.97s
{ expected_response:true }....: avg=1.09s min=14ms med=605.28ms max=59.86s p(90)=791.64ms p(95)=880.59ms
http_req_failed.....: 5.29% 385 out of 7275
http_req_receiving.....: avg=327.16µs min=0s med=0s max=54.99ms p(90)=940.74µs p(95)=1ms
http_req_sending.....: avg=79µs min=0s med=0s max=54.99ms p(90)=0s p(95)=508.37µs
http_req_tls_handshaking.....: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=4.11s min=0s med=612.69ms max=59.86s p(90)=878.37ms p(95)=58.97s
http_reqs.....: 7275 111.134505/s
iteration_duration.....: avg=4.17s min=16ms med=617.17ms max=1m0s p(90)=921.14ms p(95)=1m0s
iterations.....: 7275 111.134505/s
vus.....: 59 min=59 max=500
vus_max.....: 500 min=500 max=500

```

Figure 6 - Graphique des performances généré par k6

## 3. Problèmes détectés et solutions proposées

### Problèmes rencontrés :

- L'utilisation de k6 a démontré qu'il y a un taux d'échec (5.29%) et un temps de réponse maximum (59.86s) très élevé. Cela indique donc un problème de performance sérieux.

### Solution(s) proposées :

- Limiter le nombre de connexions simultanées à la base de données

## 4. Conclusion

Les tests ont confirmé le bon fonctionnement de l'application tout en révélant quelques anomalies qui ont ensuite été corrigées. L'intégration de l'ajout de l'âge de l'utilisateur a été validée avec succès, sans impact sur les fonctionnalités existantes.

### Améliorations à apporter :

- Optimisation de la base de données pour que les tests de charge soient plus performant
- Système de gestion des échecs de requêtes pour les tests de charge avec un mécanisme de nouvelle tentative automatique pour les requêtes ayant échoué.