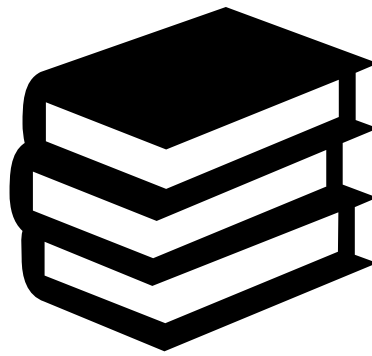


# Booker



Qualité de Logiciel

INF3 DLM B

Joris MONNET

Adrien PAYSANT



## Table des matières

Introduction.....	4
Concept de test .....	5
Attentes de la qualité du produit .....	5
Objectifs de test .....	5
Périmètre de test.....	5
Procédure de test .....	5
Equipe .....	5
Jalons .....	5
Environnement de test.....	5
Plan de communication.....	5
Plan de test.....	6
Périmètre de test.....	6
Périmètre fonctionnel .....	6
Type de tests .....	6
Niveaux de tests .....	6
Stratégie détaillée de test .....	7
Infrastructure de test .....	8
Gestion des risques .....	8
Rapport de test.....	9
Tests unitaires .....	9
Test d'intégration .....	9
Test de charge .....	10
Test d'Ergonomie .....	10
Rapport SonarCloud .....	11
Conclusion .....	11

## Introduction

Au cours du second semestre de troisième année de Bachelor informatique de la Haute-Ecole Arc de Neuchâtel, un projet combinant deux cours a été réalisé. Ce projet vise à mettre en pratique les connaissances acquises lors du cours de qualité de logiciel sur un projet de développement logiciel mené en Java Enterprise Edition (JEE), qui met alors en œuvre les compétences développées en cours de Java Enterprise Edition. La partie qualité de logiciel vise l'intégration continue du projet JEE, l'intégration d'automatisation, l'évaluation du code, le test de performance de l'application et enfin un déploiement semi-automatique. La partie JEE vise la réalisation d'une application WEB (JEE ne sera pas détaillé ici, rapport dédié à la qualité de logiciel).

Le projet JEE est nommé Booker, c'est une application qui permet de consulter des informations sur des livres, d'en ajouter, de les commenter et de les noter. En outre, trois rôles d'utilisateurs ont été mis en place. Il s'agit des lecteurs normaux, des auteurs qui peuvent ajouter des livres et des administrateurs qui ont accès à toutes les fonctionnalités.

## Concept de test

### Attentes de la qualité du produit

- L'interface est utilisable par des utilisateurs novices
- Un super utilisateur (admin) peut tout modifier
- La qualité de code est évaluée avec SonarCloud
- La résistance à la charge est évaluée avec OctoPerf
- Tests de fonctionnalité avec Katalon

### Objectifs de test

- Rendre le code apte à la maintenance
- Tester les fonctionnalités à 25%
- Tester la charge avec 10 utilisateurs simultanés

### Périmètre de test

- Test de fonctionnement avec Katalon/utilisateur
- Test de charge avec OctoPerf
- Evaluation du code avec SonarCloud
- Tests unitaires avec Spring et JUnit (optionnel)

### Procédure de test

#### Equipe

L'équipe de développement est composée de Joris Monnet et Adrien Paysant, les deux collaborateurs s'occupent du développement des tests et du développement de l'application.

#### Jalons

- Tests avec Katalon
- Tests avec OctoPerf (si mise en place possible)
- Analyse SonarCloud
- Mise en forme dans un rapport de tests

#### Environnement de test

Les tests Katalon, l'analyse SonarCloud et les tests unitaires seront automatisés via le CI/CD également mit en place. Les tests OctoPerf seront réalisés localement, à cause de la contrainte des serveurs qui nous sont fournis).

#### Plan de communication

Afin de pouvoir traiter les problèmes et bug au plus vite, chaque bug se verra associé à une issue du repository gitlab avec le label bug. Cela permet un suivi et une mise en évidence des bugs.

## Plan de test

### Périmètre de test

#### Périmètre fonctionnel

- Création d'un compte et connexion
- Modification d'informations de son compte
- Ajout d'un livre pour un auteur
- Modification d'un livre
- Recherche simple
- Recherche avancée

#### Type de tests

- Fiabilité
  - Passe les tests
- Fonctionnalité
  - Inscription
  - Connexion
  - Livre
  - Note/Commentaire
  - Edition de son profile
  - Recherche
- Efficacité et résistance à la charge
  - Temps de réponse inférieur à une seconde avec OctoPerf
- Maintenabilité
  - Analyse SonarCloud correcte
- Portabilité
  - Affichage mobile & desktop pertinent

#### Niveaux de tests

- Système : OctoPerf
- Unitaire : Spring et JUnit
- Intégration : Katalon

## Stratégie détaillée de test

- Respect des rôles
  - Accéder à la page ajout de livre en tant qu'auteur
  - Accéder à la page ajout de livre en tant que lecteur
  - Page « mes livres » pour auteur et non utilisateur et non admin
  - Modifier tous les livres quand on est admin
- Ajout de livre
  - Accès à la page
  - Ajout avec tous les champs
  - Ajout avec des champs manquants
- Note/Commentaire
  - Accès à la section
  - Mettre une note
  - Mettre à jour sa note
  - Ne pas pouvoir noter son propre livre
  - Mettre un commentaire
  - Editer ses commentaires
- Edition de son profile
  - Accès à la page
  - Modification courriel
  - Modification mot de passe
  - Modification nom & prénom
- Recherche
  - Accès à la page
  - Fonction de recherche simple
  - Fonction de recherche avancée
- Inscription
  - Accès à la page
  - Avec formulaire bien rempli
  - Avec formulaire mal rempli
- Connexion
  - Accès à la page
  - Avec compte existant
  - Avec un compte inexistant

### Infrastructure de test

- Windows 10 – Chrome : tests d'acceptation, intégrations et ergonomie
- Linux : tests unitaires et d'intégrations

### Gestion des risques

Voici notre analyse des risques pour le projet Booker.

Description	Source	Catégorie	Probabilité	Impact	Criticité	Remédiation
Retard Developpement JEE	Interne	Gestion Temps	0.2	0.3	0.06	Planification
Retard Developpement QDL	Interne	Gestion Temps	0.1	1	0.1	Planification
Démotivation	Interne	Humaine	0.2	0.5	0.1	Team Building
Manque de maitrise des outils en œuvre	Interne	Technique	0.3	0.7	0.21	Investir Temps pour formation
Serveur non disponible-problèmes divers	Externe	Technique	0.5	0.7	0.35	Notifier l'admin

Ainsi le risque principal du projet de ne pas pouvoir mettre en place le pipeline de test du fait de l'état alternatif des serveurs.



## Rapport de test

### Tests unitaires

Le but des tests unitaires est de vérifier les fonctionnalités principales. A cette fin, nous avons mis en place des tests unitaires autour du modèle utilisateur permettant de vérifier ces insert, accès et suppressions de la base de données.

### Test d'intégration

Les tests d'intégrations visent à vérifier le fonctionnement des fonctionnalités de l'application via le logiciel Katalon.

Ainsi nous testons l'accès à toutes les pages de l'application ainsi que le changement de mot de passe utilisateur à titre d'exemple d'édition.

Voici donc le rapport de test de Katalon :

booker\_login\_register

Execution Environment

Host name	adrien.paysant - TIC19-PAYSANTA.intra.eiaj.ch		
OS	Windows 10 64bit		
Katalon version	7.9.1.208		
Browser	Chrome 90.0.4430.93		

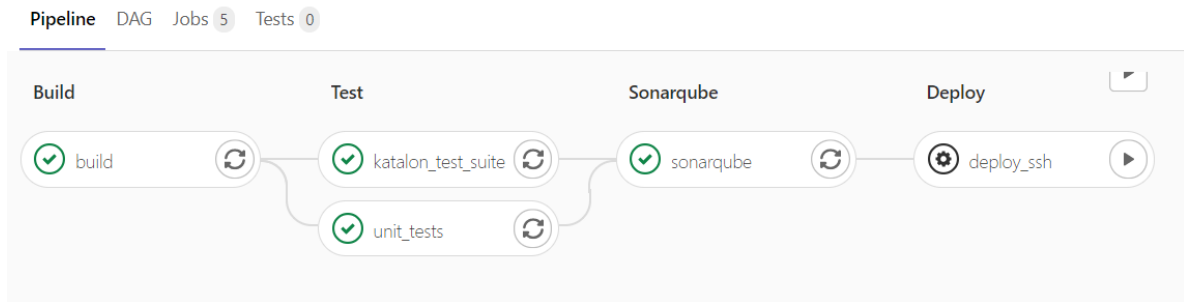
Summary

ID	Test Suites/booker_login_register		
Description			
Total	2		
Passed	2	Failed	0
Error	0	Incomplete	0
Start	2021-05-01 21:07:58	End	2021-05-01 21:08:12
Elapsed	14,506s		

#	ID	Description	Status
1	Test Cases/navigationPages		PASSED
2	Test Cases/changePassword		PASSED

Les différents autres tests d'intégrations ont été réalisés par les utilisateurs à la fois sur le serveur et en phase de test locale.

Concernant l'intégration, un pipeline CI/CD a été mis en place pour automatiser le build, les tests (SonarCloud, tests unitaires, ... ) et le déploiement. En voici le résultat :



### Test de charge

Les tests de charges visent à évaluer la résistance de l'application. Ces tests OctoPerf ne peuvent pas être réalisés via les serveurs (impossible à mettre en place à cause de la configuration des serveurs).

La solution de secours proposée impose une redirection de port vers le localhost.

Cependant, les deux membres de l'équipe ne possédant pas de connexion internet privée, il est alors impossible de mettre en place les tests de charges.

Cela n'est pas impactant, car il n'est pas pertinent de réaliser un test de charge sur une autre machine que celle de déploiement.

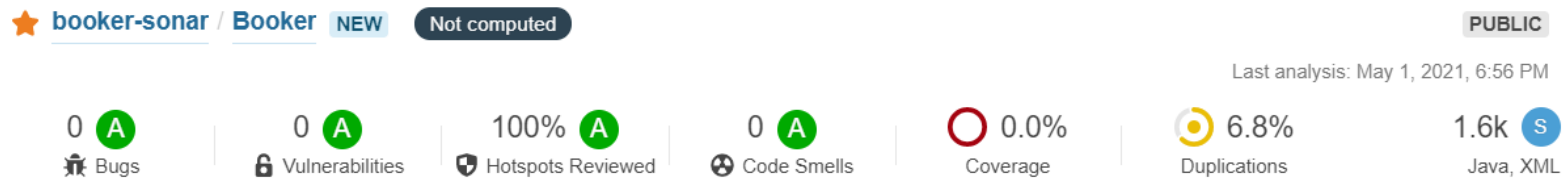
### Test d'Ergonomie

Afin de rendre compte de la pertinence des solutions d'intégrations proposée, nous avons soumis notre application web à des utilisateurs totalement externes aux cycles de développement

Le résultat est concluant, car l'utilisation de notre application ne présente pas de troubles majeurs à l'utilisation.

## Rapport SonarCloud

Voici le rapport d'analyse SonarCloud de notre projet :



Nous pouvons donc dire que notre projet répond aux exigences de qualité de code.

## Conclusion

Ainsi, au travers de projet nous avons pu acquérir et mettre en pratique des compétences dans le déploiement et l'intégration continue avec Gitlab. Ce projet nous a également permis d'utiliser des outils variés et nouveaux pour nous tels que Docker, SonarCloud, OctoPerf ou encore Katalon. Par ailleurs la réalisation de ce projet à permit la mise en pratique des méthodes et concepts étudiés en cours de Qualité de Logiciel. Les différentes approches de tests permettent une couverture de test présentant un grand recouvrement du projet. Nous avons réussi à mette en place la majorité de nos spécifications détaillées, nous pouvons donc considérer une certaine réussite dans la mise en place de l'intégration et du déploiement continu.

Après ce projet, il serait intéressant de réutiliser les bonnes pratiques misent en œuvres dans ce projet lors de futur déploiement.