Inquirist

Projet de qualité logiciel

INF3 dlm-b

Réalisé par

Luca Campana

Lenny Boegli

Léon Muller

Matthieu Barbot

Table des matières

1. Introduction

Pour les cours de « qualité du logiciel (QDL) » et de « Java Entreprise Edition (JEE) », nous avons pour but de réaliser un projet en commun reliant les deux disciplines. Ce rapport reflète la partie réalisée lors du cours de qualité du logiciel et plus précisément celle concernant le déploiement et l'intégration continue de l'application développée en JEE. Pour être encore plus précis, ce rapport reflète principalement les tests effectués lors de l'intégration et du déploiement de l'application sur un des serveurs de l'école mis à disposition pour ce projet. Par « tests », nous entendons réaliser et présenter ceux vus en cours de QDL.

Notre application « Inquirist » est un application web développée en Java Entreprise Edition permettant de soumettre des sondages et de les partager à d'autres utilisateurs (au style de Doodle[1]). L'application permet de gérer deux genres d'utilisateurs : les utilisateurs « inscrits » (ayant un compte sur notre application) et les utilisateurs « guests » (n'ayant pas de compte sur l'application).

2. Concept de test

2.1. Attentes de la qualité du produit

- Interface utilisateur la plus intuitive et simple possible.
- Les erreurs à afficher doivent être le plus clair possible lorsqu'un formulaire (de création ou de login) n'est pas valide.
- Mettre en évidence le statut de l'utilisateur lorsqu'il se trouve sur l'application (guest ou connecté à son compte).
- Seuls les utilisateurs inscrits sur l'application peuvent créer des sondages.
- Les mots de passe des utilisateurs sont chiffrés dans la BDD.
- Il ne sera pas possible à un utilisateur de voter plusieurs fois à un même sondage.
- Les sondages peuvent être uniquement supprimés par l'utilisateur l'ayant créé.

2.2. Objectifs de tests

- Avoir au moins 30% de couverture des tests sur le projet.
- Avoir un temps de réponse raisonnable lorsqu'un utilisateur essaie d'atteindre une page (max 4 secondes).

[1] https://doodle.com/fr/

- Permettre une charge de 30 personnes minimum ? sur l'application web
- Avoir un code permettant une maintenabilité raisonnable (si nécessaire, il doit être possible de facilement comprendre et modifier notre code).

2.3. Périmètre de test

- Tests unitaires avec JUnit
- Tests d'intégration avec Katalon
- Tests de charge avec OctoPerf
- Tests d'acceptation et d'ergonomie avec des utilisateurs tests
- Qualité du code avec SonarCloud

2.4. Procédure de test

2.4.1. Equipe de développement

L'équipe de développement d'Inquirist est constituée de Luca Campana, Lenny Boegli, Léon Muller et Matthieu Barbot. Les deux premiers se sont principalement occupés de la partie de déploiement, d'intégration et des tests de l'application tandis que les deux autres se sont concentrés sur la partie de développement.

2.4.2. Étapes principales

Les étapes de tests seront exécutées dans l'ordre suivant :

1. Tests unitaires avec JUnit5
2. Tests d'intégration avec Katalon
3. Tests de charges avec OctoPerf
4. Analyse de la qualité de code avec SonarCloud
5. Tests d'ergonomie
6. Analyse des résultats sous la forme de rapport de tests

Si l'une des étapes rencontre un problème, celui-ci sera corrigé avant de passer à l'étape suivante.

3. Plan de test

3.1. Périmètre de test

3.1.1. Périmètre fonctionnel

- Création d'un nouveau compte et authentification
- Modification et suppression d'un compte (autorisé que si on est connecté)
- Création, modification et suppression d'un poll (autorisé que si on est connecté et qu'on est le créateur du poll)
- Vérification du chiffrement des MDP dans la BDD
- Ergonomie du site
- Tentative de voter plusieurs fois sur le même poll (impossibilité si déjà voté)

3.1.2. Types de tests

- Fonctionnalité
    - Gestion des rôles respectée (fonction refusée si l'utilisateur n'a pas le bon rôle)
    - Affichage correct des questionnaires
- Fiabilité
    - Validation de tous les tests mis en place
- Ergonomie
    - Les utilisateurs ne doivent pas être perdu en naviguant sur notre application
- Efficience
    - Temps de réponse acceptable même sous charge
- Maintenabilité
    - Pas de problèmes majeurs détectés par SonarCloud
- Portabilité
    - Affichage correct sur ordinateur, tablette et mobile (responsive)

3.1.3. Niveaux de test

- Unitaires : JUnit
- Intégration : Katalon
- Système : OctoPerf (tests de charge)
- Validation : utilisateurs externes

3.2. Stratégie détaillée de test

- Création et suppression d'un compte
    - Accéder à la page d'inscription
    - Tester l'inscription avec des données erronées
        - Mot de passe différent de la confirmation de mot de passe
    - Tester l'inscription avec des données valides
    - Accéder à la page d'authentification
    - Tester l'authentification avec un compte inexistant
    - Tester l'authentification avec un compte existant
    - Accéder à la page de son profil
    - Tester la suppression de compte
    - Accéder à la page d'un utilisateur public
- Création et suppression d'un poll
    - Accéder au formulaire de création d'un poll sans être authentifié
    - Accéder au formulaire de création d'un poll en étant authentifié
    - Tester la création avec le titre manquant
    - Tester la création avec toutes les données
    - Accéder à la page du poll créé
    - Tester la suppression du poll
    - Vérifier qu'un utilisateur ne peut pas voter deux fois pour un même poll
- Chargement du site, temps de réponse acceptable si une dizaine d'utilisateurs sont connectés
    - Dans OctoPerf, lancer un test de charge avec 10 utilisateurs. Le rapport généré par OctoPerf nous donne le temps de réponse moyen des requêtes.
- Ergonomie du site
    - On demande à notre famille et amis de tester le site et on observe leurs facilité d'utilisation et de navigation. On écoute au long du test leurs avis et remarques.
    - On teste l'affichage du site sur différents périphériques
        - Ordinateur
        - Tablette
        - Mobile

3.3. Infrastructure de test

Les environnements sont les suivant :

- Windows 10 pour les tests d'acceptation
- Linux (container docker) pour les tests unitaires et d'intégrations
- Windows 10, MacOs, iOS et Android (navigateurs Internet) pour les tests d'ergonomie

3.4. Gestion des risques

Dans le tableau ci-dessous, les valeurs pour la probabilité, l'impact et la criticité varient entre 0 et 1.

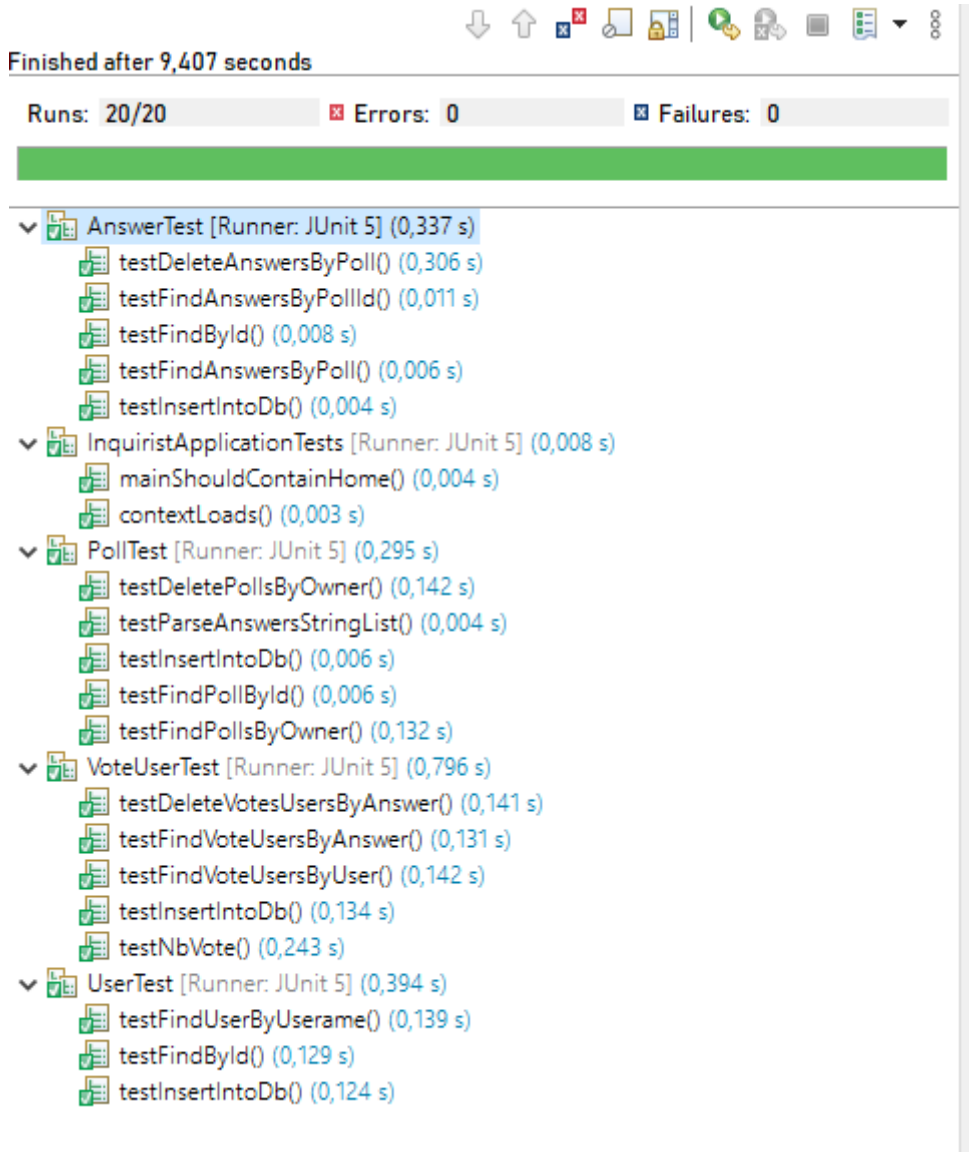
Description	Source	Catégorie	Proba.	Impact	Criticité	Remédiation
Manque de motivation	Interne	Psychologique	0.3	1.0	0.5	1) Penser aux conséquences en cas de non-travail rendu. 2) Avoir un cadre et une cohésion de groupe motivante
Défaut d'expérience dans la mise en place de tests	Interne	Compétence	0.8	0.5	0.4	Travail de recherche et de lecture avec mise en pratique en parallèle.
Serveur de déploiement HS	Externe	Technique	0.2	1.0	0.02	Détecter au plus vite la panne pour agit et relancer le serveur. Être à l'écoute constante du serveur.
Incompréhension entre les développeurs et les administrateurs serveurs	Interne	Organisation	0.4	0.6	0.24	Déterminer dès le début quels sont les tâches de chacun et comment organiser la liaison entre les deux groupes.
Inexpérience en Docker	Interne	Compétence	0.8	0.8	0.64	Pouvoir suivre une formation Docker au préalable. Rechercher de manière intense des solutions aux problèmes rencontrés.

4. Rapport de test

4.1. Tests unitaires

Notre objectif était de tester les principales routes de notre application ainsi que les opérations d'insertion, de recherche et de suppression.

Voici les résultats des tests unitaires réalisés (avec JUnit) :



4.2. Test d'intégration

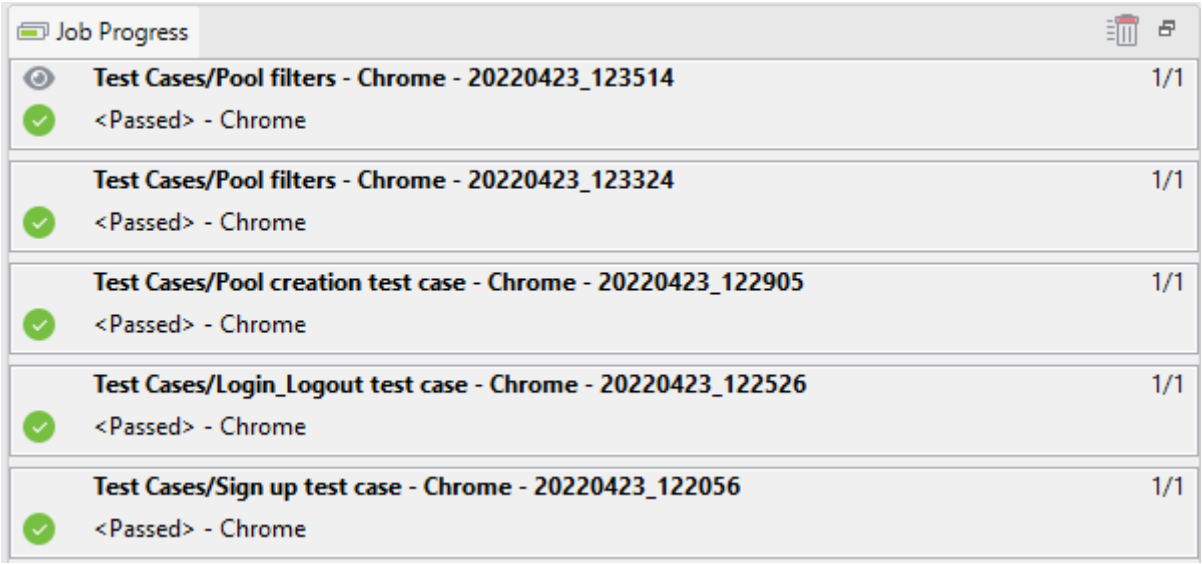
L'objectif des tests d'intégrations était de vérifier le fonctionnement des différentes fonctionnalités de base de notre application web à l'aide du logiciel Katalon. Le logiciel nous a permis de tester les éléments suivants :

- Visite de la page d'accueil

- Authentification et logout
- Création d'un utilisateur
- Création d'un pool
- Recherche d'un pool

Les tests ont d'abord été lancés en local (http://localhost:8080/)

Voici les résultats des tests obtenus :



Les scripts de tests Katalon sont disponibles dans le répertoire « **délivrable** ».

4.3. Tests de charge

Les tests de charge ont été réalisés avec OctoPerf en local et en déploiement sur le serveur.

Le scénario mis en place est le suivant :

- Un nombre de 30 utilisateurs sont en concurrence sur le site (limite maximale avec un compte OctoPerf gratuit).
- La durée du scénario est de 5 minutes.

Le grand souci que nous avons rencontré est que notre webapp sur le serveur n'est pas disponible sans avoir accès au VPN. Nous avons essayés plusieurs alternatives :

- D'ouvrir un port sur un de nos routeurs en local afin que Octoperf puisse avoir accès à notre site, mais les fonctionnalités d'ouverture des ports ne sont pas disponibles sur nos routeurs de maisons (à cause d'UPC…)

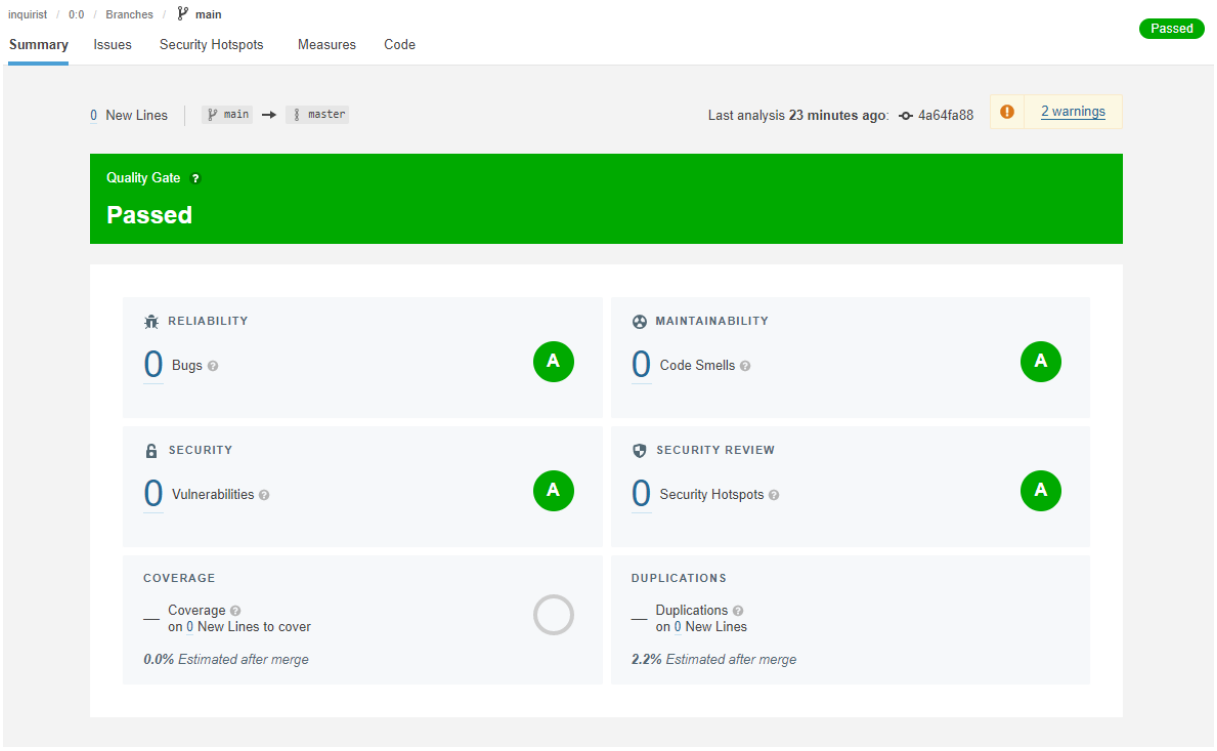
- Nous avons aussi tenté les tests en créant un agent local sur un poste connecté au VPN pour servir de proxy à OctoPerf via les "On-Premise" (https://doc.octoperf.com/private-hosts/on-premise/) sans succès (idée inspirée d'un projet des années précédentes)[2].

Vous pourrez tout de même trouver le rapport exporté depuis octoperf dans le dossier « délivrables » montrant qu'il n'arrive pas à se connecter.

4.4. Analyse SonarCloud

Le but d'effectuer l'analyse avec SonarCloud était de ne pas avoir de bugs, le moins de vulnérabilités possibles, le moins de Code Smells et au moins 30% de coverage.

Voici les résultats retournés par SonarCloud :



Les résultats peuvent aussi être visualisés sur le lien suivant : https://sonarcloud.io/summary/new_code?id=0%3A0&branch=main .

[2] https://gitlab-etu.ing.he-arc.ch/maxime.welcklen/skycast/-/blob/4ebe4e9241ec1f18642a15b25558f622e667e81e/docs/rapport.md

4.5. Test d'ergonomie

L'objectif de ce test est de vérifier si l'expérience utilisateur ainsi que l'ergonomie de l'application web est proche de nos attentes. Pour ce faire, nous avons demandé à une personne hors du groupe de développement de tester l'application.

Nous avons donné à cette personne une liste d'action, non détaillée, à réaliser et la possibilité de noter la facilité de l'action sur une échelle de 0 à 10.

Actions	Facilité
Création d'un compte	9
Connexion au compte	10
Déconnexion du compte	10
Création d'un poll	10
Recherche d'un poll	7

Moyenne	46/50 = 92%

Note :

- L'utilisateur testeur était perdu quant à la différence entre les rôles (READER, WRITER et ADMIN) → Aucune explication n'est donnée sur le site.
- Pour la recherche des polls, le site est sensible à la casse et ceci a surpris l'utilisateur. Pour l'utilisateur test, il ne s'attendait pas à avoir une différence entre les majuscules et les minuscules.