

Document Explicatif

1. GitHub Actions

a. Workflows de tests

nodejs.yml (Front-End) : Installe les dépendances, construit le projet et exécute les tests pour l'application frontend Node.js.

maven.yml (Back-End) : Exécute des tests, génère des rapports de couverture de code et construit le projet backend Java avec Maven.

b. Workflows d'analyses

sonarqube.yml (SonarCloud) : Analyse le code source des applications backend et frontend avec SonarQube pour détecter des bugs et des mauvaises pratiques.

c. Workflows de CI/CD

docker-image.yml (DockerHub) : Automatise la construction et le push des images Docker pour les applications backend et frontend sur Docker Hub.

d. Définitions des étapes

nodejs.yml

actions/checkout@v3 : Récupère le code source du dépôt GitHub.
actions/setup-node@v3 : Installe la version spécifiée de Node.js.
npm ci : Installe les dépendances du projet en utilisant npm.
npm run build : Construit le projet Node.js.
npm run test : Exécute les tests de production pour le projet Node.js.
actions/upload-artifact@v3 : Télécharge le rapport de couverture de code généré comme un artifact.

maven.yml

actions/checkout@v4 : Récupère le code source du dépôt GitHub.
actions/setup-java@v3 : Installe JDK 17 pour l'exécution des commandes Maven.
mvn -B package --file pom.xml : Construit le projet Maven en mode batch pour réduire le bruit des logs.
mvn clean install : Effectue une installation propre du projet Maven.
mvn jacoco : Génère un rapport de couverture de code avec Jacoco.
actions/upload-artifact@v3 : Télécharge le rapport de couverture de code généré comme un artifact.

sonarqube.yml

actions/checkout@v4 : Récupère le code source du dépôt GitHub avec une profondeur de fetch complète pour une meilleure analyse.
actions/setup-java@v1 : Installe JDK 17 pour l'exécution des commandes Maven.
actions/cache@v3 (SonarCloud packages) : Met en cache les packages SonarCloud pour accélérer les builds futurs.

docker-image.yml

actions/checkout@v4 : Récupère le code source du dépôt GitHub.
docker/login-action@v3 : Se connecte à Docker Hub en utilisant les identifiants fournis.
docker/setup-buildx-action@v3 : Configure Docker Buildx pour permettre la construction d'images multi-plateformes.
docker/build-push-action@v6 : Construit et pousse les images Docker sur Docker Hub.

actions/cache@v3 (Maven packages) : Met en cache les dépendances Maven pour accélérer les builds futurs.

mvn -B verify org.sonarsource.scanner.maven:sonar-maven-plugin

-Dsonar.projectKey=p10-bobapp_back : Construit et analyse le code backend avec SonarQube.

actions/setup-node@v3 : Installe la version spécifiée de Node.js pour le frontend.

npm ci : Installe les dépendances du projet frontend en utilisant npm.

npm run build : Construit le projet frontend.

sonarsource/sonarcloud-github-action@master : Analyse le code frontend avec SonarCloud en utilisant les paramètres fournis.

2. KPI proposées

a. Taux de couverture

Je propose un taux de couverture minimum de 80% pour le front-end et le back-end.

Avantages :

1. **Amélioration de la qualité du code** : Un taux de couverture élevé assure une meilleure détection des bugs et des erreurs potentielles, améliorant ainsi la stabilité et la fiabilité du code.
2. **Meilleure maintenabilité** : Avec une couverture de code élevée, il est plus facile de comprendre et de maintenir le code, ce qui facilite les modifications futures et réduit les risques de régressions.

Inconvénients :

1. **Temps de développement accru** : Atteindre un taux de couverture de 80% peut nécessiter un effort supplémentaire en écriture de tests, ce qui peut rallonger le temps de développement initial.
2. **Risque de tests redondants** : Pour atteindre ce taux, il peut arriver que certains tests soient redondants ou peu utiles, ce qui pourrait compliquer la suite de tests sans apporter une réelle valeur ajoutée.

b. SonarCloud

KPI Proposées pour SonarCloud

Pour garantir la qualité optimale des deux projets (front-end et back-end), les métriques de SonarCloud doivent satisfaire les critères suivants :

1. **Security** : Note "A" (0 vulnérabilités détectées).
2. **Reliability** : Note "A" (0 bugs détectés).
3. **Maintainability** : Note "A" (0 "code smells" critiques).
4. **Hotspots Reviewed** : Note "A" (tous les hotspots de sécurité doivent être examinés).

Niveaux de Sévérité des Issues :

- **High** : 0 issues tolérées.
- **Medium** : Jusqu'à trois issues tolérées.
- **Low** : Pas de limites spécifiques, mais il est préférable d'en avoir le moins possible.

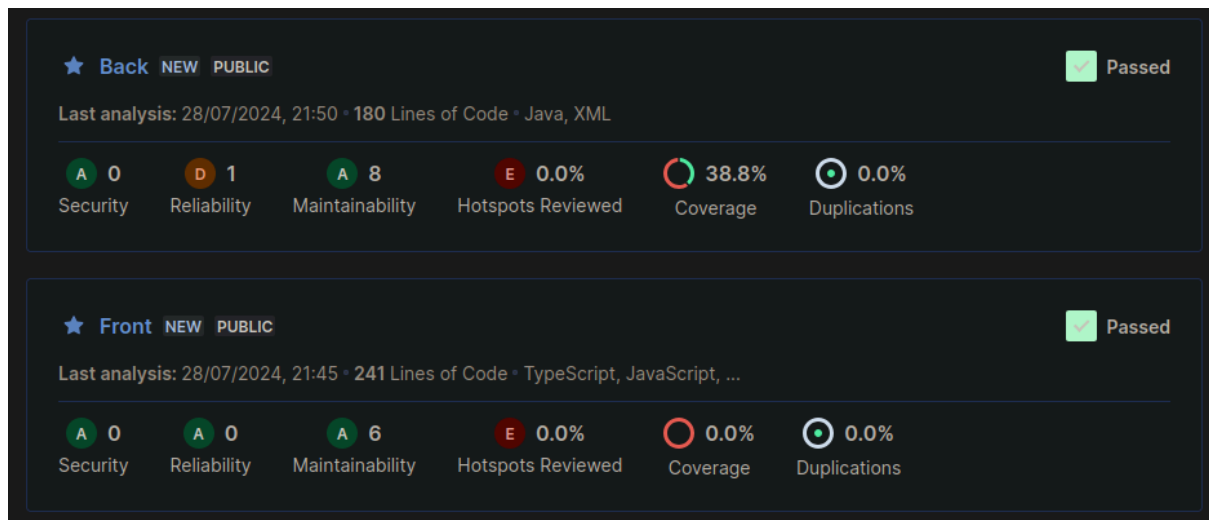
New Blocker Issues : Les "New blocker issues" sont des dysfonctionnements critiques qui peuvent grandement affecter le comportement attendu de l'application en production. Par exemple, une fuite de mémoire est considérée comme un "blocker" et doit être corrigée immédiatement, comme tous les autres "blockers" identifiés. SonarCloud désigne clairement les "blockers" repérés pour permettre leur résolution rapide.

Coverage : Un taux de couverture de code minimum de 80% est recommandé pour garantir une bonne qualité des tests.

Duplications : Il est préférable de minimiser le pourcentage de code dupliqué dans le projet, même s'il n'y a pas de seuil strict, car cela impacte la maintenabilité.

3. Métriques et retours utilisateurs

a. Métriques



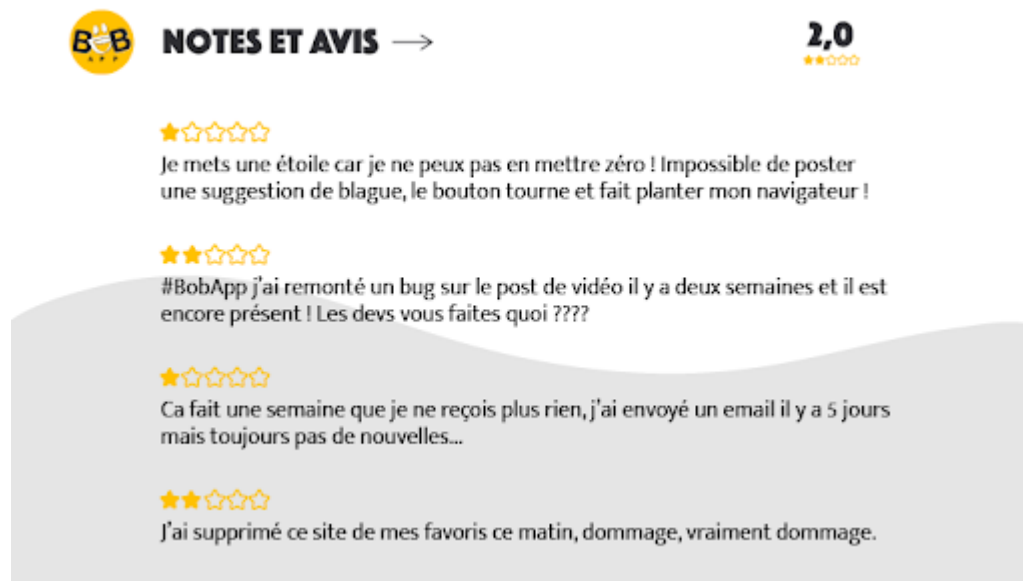
Back-End :

- 0 Security (A)
- 1 Reliability (D)
- 8 Maintainability (A)
- 0.0% Revue de sécurité / Hotspots Reviewed (E)
- 38.8% code coverage

Front-End :

- 0 Security (A)
- 0 Reliability (A)
- 6 Maintainability (A)
- 0.0% Revue de sécurité / Hotspots Reviewed (E)
- 0.0% code coverage

b. Notes et avis



Retours Pertinents :

1. **Bug d'affichage lors de la soumission de suggestions de blagues :**
 - **Problème :** Le bouton tourne indéfiniment et fait planter le navigateur.
 - **Gravité :** Critique. Ce problème affecte directement l'expérience utilisateur et empêche l'utilisation d'une fonctionnalité clé.
 -
2. **Bug non résolu concernant la lecture de vidéos :**
 - **Problème :** Un bug signalé il y a deux semaines concernant la lecture de vidéos n'est toujours pas corrigé.
 - **Gravité :** Critique. Ce type de problème persistant peut entraîner une perte de confiance des utilisateurs dans la capacité de l'équipe à maintenir l'application.
3. **Absence de réponse du support client :**
 - **Problème :** Un utilisateur se plaint de ne pas avoir reçu de réponse à un email envoyé il y a plus de cinq jours.
 - **Gravité :** Élevée. La réactivité du support est essentielle pour maintenir la satisfaction des utilisateurs, surtout lorsqu'ils rencontrent des problèmes.
4. **Perte d'intérêt et désinstallation de l'application :**
 - **Problème :** Un utilisateur a supprimé le site de ses favoris, exprimant sa déception.
 - **Gravité :** Moyenne à élevée. Ce feedback indique une frustration générale qui pourrait entraîner une diminution de la base d'utilisateurs.

