Document explicatif

1. Étapes des Github Actions

Workflows des tests : Il y a deux fichiers, celui des tests front et des tests back. Pour commencer avec les tests front, on a d'abord le nom global du workflow « name », ensuite la section « on » qui va désigner les branches sur lesquelles nous allons faire des « push » et des « pull_request ». Une autre section du fichier qui est « jobs », cette dernière réunit les workflows et décrivent leurs comportements, dans le fichier « frontend-tests-coverage.yml », il n'y a qu'un seul workflow nommé « Frontend-test ». Ce workflow tourne sous ubuntu (« runs-on ») dans le dossier front (« defaults ») du projet avec la version 16 de Node.js (« strategy »). La sous-section suivante est « steps », elle rassemble toutes les commandes que l'on souhaite automatiser, chaque commande est nommée par « name » et utilise soit « uses », soit « run ». Dans ce fichier, cela donne donc « Checkout » pour effectuer des vérifications avant de configurer Node.js, puis installer les dépendances avec « npm install », ensuite lancer les tests et générer les rapports de couverture. Pour les tests back, la procédure est exactement la même, il suffit de s'adapter avec la version du JDK et d'automatiser les commandes avec Maven.

Workflows CI/CD : Il y a un fichier « docker.yml », celui-ci se charge de déployer des images Docker sur Docker Hub. Les étapes sont les suivantes : des vérifications puis la connexion sur la plateforme Docker Hub, il faudra pour cela renseigner au préalable son nom d'utilisateur et un jeton en guise de mot de passe dans les paramètres du dépôt Github, ce sont les variables secrètes (Settings → Secrets and variables → Actions puis cliquer sur New repository secret et renseigner ses variables secrètes). Une fois la connexion réussie, configurer Docker puis construire et envoyer les images Docker vers Docker Hub pour le front et le back. Ces étapes indiquent dans quel dossier opérer, sélectionnent les fichiers Dockerfile et génèrent les commandes qui serviront de téléchargement de conteneurs depuis le Docker Hub (« tags »).

Workflows des analyses : il y a deux fichiers « sonarcloud_front.yml » et « sonarcloud_back.yml », ceux-ci automatisent les analyses avec l'outil SonarCloud. Il faudra renseigner au préalable les jetons front et back dans les variables secrètes pour avoir deux projets distincts. On retrouve bien sur les commandes Node en front et Maven en back, la dernière est celle des analyses avec SonarCloud.

À chaque commit, les workflows s’exécutent automatiquement, permettant aux développeurs de gagner du temps en évitant de répéter les mêmes commandes.

À noter : lorsqu'un développeur veut faire un commit de fichiers, si un workflow échoue, le commit de fichiers échoue également.

2. KPI proposés

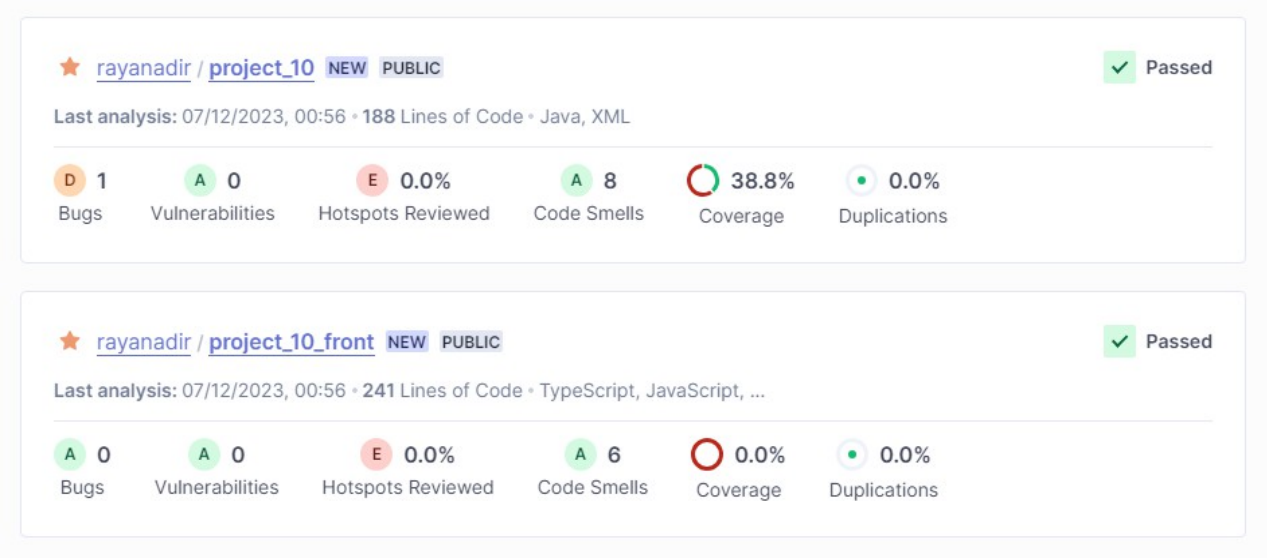
Deux KPI sont proposés dans cette section :

Le premier concerne le taux de couverture du code, les projets front et back doivent atteindre un minimum de 80%.

Le deuxième concerne les métriques optimales de SonarCloud, les deux projets doivent avoir pour notes « A » en bugs (donc 0), « A » en vulnérabilités, « A » en revue de sécurité (« Hotspots Reviewed ») et aussi un « A » en maintenabilité (« Code smells »). Un autre point sur lequel se pencher est la sévérité des « Issues », chaque projet recense les issues en trois niveaux : d'abord « High », « Medium » et « Low ». La règle proposée est la suivante : 0 issues en « High », jusqu'à trois issues tolérées en « Medium » et pas de limites pour les « Low » même s'il est préférable d'en avoir le moins possible.

En ce qui concerne les « New blocker issues », ce sont des dysfonctionnements qui peuvent fortement impacter le comportement attendu de l'application en production. Par exemple, une fuite de mémoire est un « blocker » et doit être immédiatement corrigé comme tous les « blockers » connus. SonarCloud désigne les « blockers » repérés pour pouvoir les corriger.

3. Analyse des métriques et des retours utilisateurs



Métriques	Projet back (project_10)	Projet front (project_10_front)
Bugs	1 (note D)	0 (note A)
Vulnérabilités	0 (note A)	0 (note A)
Revue de sécurité (Hotspot reviews)	0% (note E)	0% (note E)
Maintenabilité (Code smells)	8 (note A)	6 (note A)
Couverture (Coverage)	38,80%	0,00%
Duplications	0,00%	0,00%
Issues (High, Medium & Low)	3 High, 1 Medium, 5 Low	0 High, 5 Medium, 1 Low

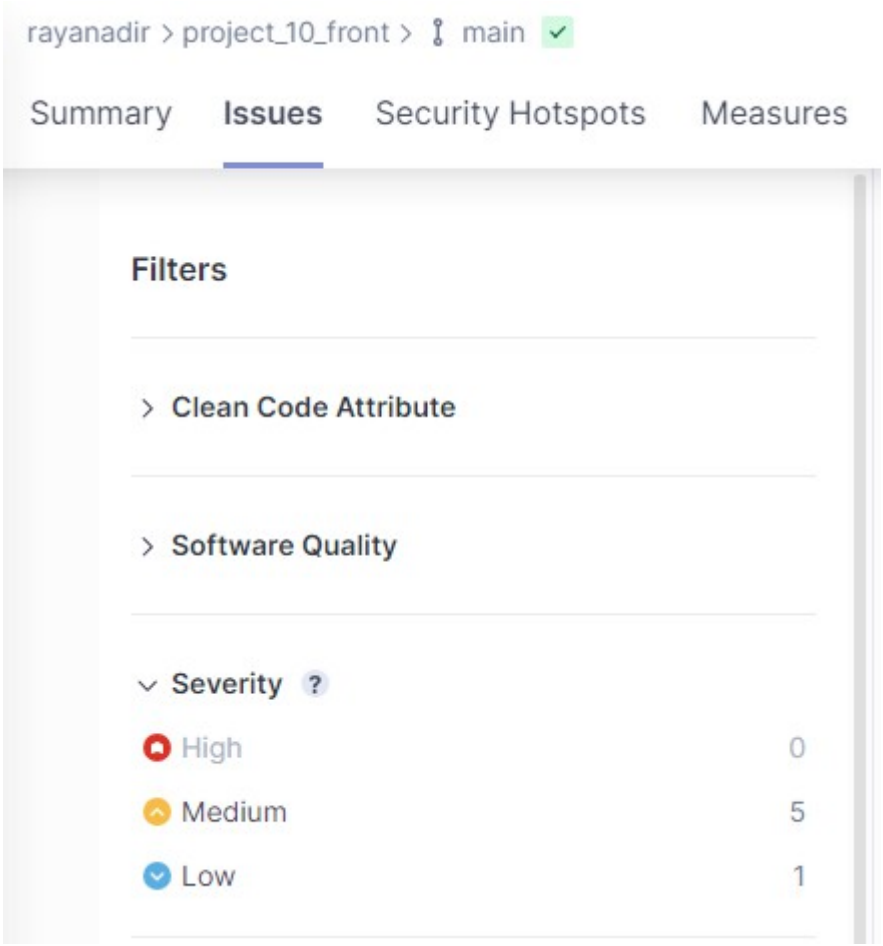


Illustration 1: Sévérité issues front

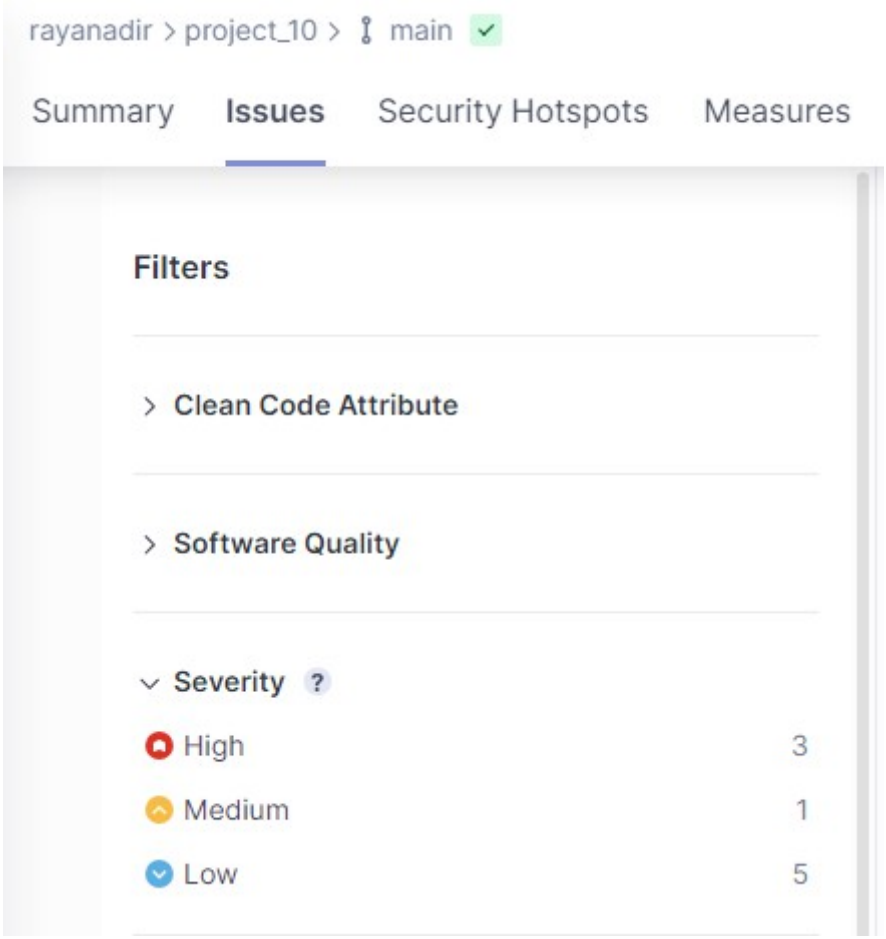
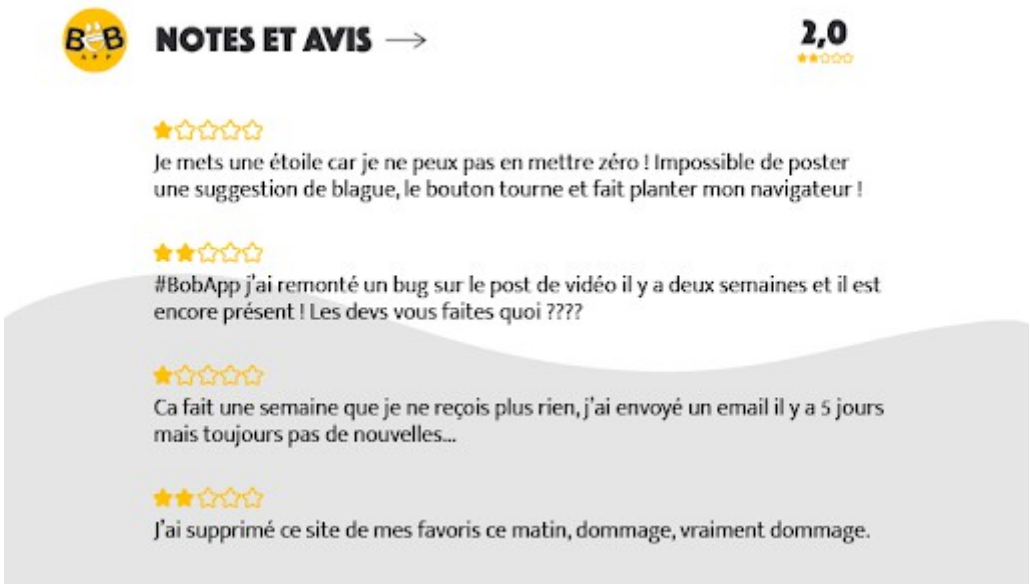


Illustration 2: Sévérité issues back

Notes et avis rapportés :



- Premier avis : post de suggestion de blague non fonctionnel, chargement et crash du navigateur.
- Deuxième avis : bug sur post de vidéo.

- Troisième avis : plus aucune récupération de données.