Buchmuller Nassim
El Kihal Ewan
Loeb Dorian
Rizzante–Madonna Alexandre
Turmo Baptiste

3ème année Groupe A1

R6.A.06 Maintenance: Analyse Final Compte rendu

Introduction

Cette phase concerne l'analyse d'un autre projet afin d'analyser les critères liés à la maintenance: https://github.com/LACOMBE-Dorian-2326065aa/maintenance.

Critères d'évaluation de la maintenabilité

Choix des techno, pertinence: 4
PHP 8.x et Symfony 7.4, encore maintenu => Choix pertinent pour le Backend car Symfony framework full-stack complet et PHP langage Web orientée Serveur. On peut donc facilement réaliser une API.
ORM Doctrine: permet de lier les types de données de la BDD et de l'API. Choix pertinent car bien supporté par Symfony.
Angular 18.0.0 et TypeScript 5.4.2 pour le frontend => Choix pertinent car Angular permet de gérer par composant les élément de l'interface, mais Angular 18 n'est plus supporté
Docker pour l'intégration continue => Pertinent car un conteneur Docker se lance de manière identique, peu importe l'environnement.
SQLite pour la base de données => Choix peu pertinent car peut être limité pour le stockage de beaucoup de données, malgré sa légèreté.

Principes SOLID respectés : Oui

- Respect de la responsabilité unique
- Respect de l'ouverture à l'extension et fermeture à la modification
- Respect du principe de Liskov
- Respect de la ségrégation des interfaces
- Respect de l'inversion des dépendances

Loi de Demeter : Oui: pas d'appel de méthodes sans relations hors certaines méthodes du langages ou framework.

Principe KISS : 5: Le code et l'architecture du logiciel sont simples, bien organisés, faciles à comprendre.

Documentation : Il manque l'explication des classes et méthodes dans le backend (pas de PHPDoc).
Qualité de la documentation: Plusieurs README expliquent bien comment utiliser le logiciel, même si certaines commandes manquent.
Conclusion: 3

Linter : Oui: PHP-CS-Fixer dans le backend, ESLint dans le frontend.
5 car tous les tests passent.

Usage de métriques : non: 1: Pas de métriques comme l'Index de Maintenabilité.

Structure des branches , pertinence : Seulement la branche de dev et main. Pas une fonctionnalité par branche et par personne.
Présence d'outils de contrôle (Linter…): Présence d'un fichier docker-compose et un Dockerfile, mais ne sont pas exécutées en tant qu'Actions sur GitHub (pas un workflow automatique).
Conclusion: 3

Tests :

- tests unitaires : oui, PHPUnit dans le backend
  pas d'erreur : 5: Les assertions ont fonctionné.
- behavior : oui, PHPUnit avec WebTestCase dans le backend
  pas d'erreur : 2: Les assertions n'ont pas fonctionné.
- finaux : oui, Cypress dans le frontend et Symfony Panther dans le backend
  pas d'erreur : 3: Les assertions ont fonctionné dans le backend, mais pas dans le frontend