

Rapport de stage

de deuxième année BTS SIO SLAM
effectué du 5 Janvier au 15 Février 2026

Table des matières

Introduction.....	3
Présentation de l'entreprise et contexte du stage.....	3
Présentation de l'entreprise.....	3
Présentation du contexte technique du stage.....	4
Besoins identifiés.....	5
Travail à réaliser durant le stage.....	6
Analyse de l'étude existante et l'étude d'opportunité.....	7
Analyse de l'existant.....	7
Étude d'opportunité.....	8
Planning prévisionnel.....	8
Mise en place et configuration de solutions logicielles.....	9
Développement et maintenance d'applications.....	10
Gestion de bases de données.....	11
Tests et validations.....	11
Support aux utilisateurs et accompagnement technique.....	12
Analyse critique des activités.....	12
Difficultés rencontrées.....	12
Solutions mises en œuvre.....	13
Compétences acquises et développées.....	14
Comparaison avec les attentes initiales.....	16
Apport du stage et perspectives.....	16
Critique du planning.....	17
Bilan des objectifs atteints.....	17
Apport du stage en termes de compétences et de connaissances.....	18
Propositions d'amélioration pour l'entreprise.....	19
Perspectives d'avenir (professionnel, poursuite d'études).....	19
Conclusion.....	20
Synthèse des principaux points.....	20
Réponse à la problématique.....	20
Remerciements.....	22
Annexes.....	23
Bibliographie.....	29

Introduction

Présentation de l'entreprise et contexte du stage

Dans le cadre de ma formation, j'ai effectué un stage au sein de l'entreprise *Swantrad*, une startup française spécialisée dans la digitalisation des traductions assermentées. Ce stage s'inscrit dans un contexte de transformation numérique d'un secteur encore largement basé sur des processus papier, malgré des enjeux juridiques et sécuritaires majeurs.

Swantrad a pour ambition de moderniser et sécuriser la traduction officielle grâce à une plateforme entièrement numérique, conforme aux normes européennes en vigueur. Mon stage s'est déroulé dans un environnement innovant, au cœur d'un projet en phase de développement actif, mêlant enjeux techniques, juridiques et métiers.

Présentation de l'entreprise

Secteur d'activité

Swantrad évolue dans le secteur des *services numériques appliqués à la traduction assermentée*. La plateforme s'adresse principalement aux traducteurs assermentés, mais également aux professionnels du droit (notaires, avocats), aux entreprises et aux administrations ayant besoin de traductions officielles juridiquement reconnues.

La startup se positionne comme un acteur pionnier en France en intégrant la *signature électronique qualifiée conforme au règlement européen eIDAS*, permettant de garantir l'authenticité, l'intégrité et la valeur juridique des documents traduits dans toute l'Union européenne.

Taille et organisation de l'entreprise

Swantrad est une startup à taille humaine, fondée à Valbonne (Sophia Antipolis), au sein d'un écosystème technologique dynamique. L'entreprise a été officiellement lancée le **18 septembre 2025 à Nice**.

L'équipe fondatrice est composée de profils complémentaires :

- des experts techniques,
- des professionnels du droit et de la traduction,
- des profils orientés marketing et stratégie.

Cette organisation favorise une collaboration étroite entre les différents pôles (technique, produit, marketing et métier), permettant des prises de décision rapides et une forte agilité dans le développement du produit.

Service / département d'accueil

Mon stage s'est déroulé principalement au sein du pôle technique, en lien direct avec le développement et la qualité de la plateforme *Swantrad*. Ce service est chargé de :

- concevoir et maintenir l'application web,
- garantir la fiabilité, la sécurité et la conformité juridique des fonctionnalités,
- améliorer la qualité du code et la stabilité du système,
- accompagner l'évolution du produit en fonction des retours utilisateurs.

Le travail s'effectue en collaboration régulière avec les équipes produit et métier, afin d'assurer l'adéquation entre les besoins fonctionnels des traducteurs et les choix techniques.

Environnement de travail et technologies utilisées

L'environnement de travail est résolument orienté *startup technologique*, avec des méthodes modernes de développement logiciel et une forte culture de la qualité.

La plateforme *Swantrad* repose sur un *environnement web moderne*, incluant notamment :

- un backend basé sur *PHP et le framework Laravel*,
- une base de données relationnelle pour la gestion des utilisateurs, missions et documents,
- des outils de versioning et de collaboration via *GitHub*,
- une chaîne d'intégration continue (*CI/CD avec GitHub Actions*) pour automatiser les tests et garantir la stabilité du code,
- des outils de qualité logicielle tels que *SonarQube* pour l'analyse du code,
- l'intégration de services tiers sécurisés (paiement via *Stripe*, signature électronique via *Yousign*).

Cet environnement m'a permis de travailler dans des conditions proches de celles d'un projet professionnel réel, avec des exigences élevées en matière de qualité, de sécurité et de fiabilité.

Présentation du contexte technique du stage

Au début du stage, le projet *Swantrad* était déjà opérationnel en production. Toutefois, l'analyse du code existant a rapidement mis en évidence plusieurs faiblesses techniques, principalement liées à des développements réalisés sur une période étendue par différentes équipes. Cette évolution progressive du projet ne respectait pas toujours une homogénéité globale ni l'ensemble des bonnes pratiques recommandées par le framework Laravel.

Lors de la prise en main du projet, plusieurs problématiques techniques ont été identifiées. Le projet ne pouvait pas être lancé correctement en environnement local sans adaptations spécifiques, ce qui compliquait le travail de développement et de maintenance. De plus, certaines migrations de base de données se sont révélées incomplètes ou incohérentes, rendant la gestion du schéma fragile. L'analyse du code a également mis en lumière la présence de nombreux éléments inutilisés, tels que des contrôleurs, des vues Blade et des fichiers CSS non exploités, ainsi qu'une duplication importante de code en HTML, CSS et PHP.

Par ailleurs, le projet disposait de quelques tests automatisés, mais ceux-ci restaient partiels, peu structurés et insuffisamment intégrés dans un processus de validation continue. Cette situation augmentait le risque de régression lors des modifications et rendait les évolutions plus délicates à sécuriser. L'utilisation de l'outil SonarQube a confirmé l'existence d'une **dette technique** estimée à environ 42 jours de développement, avec plus de 2100 Code Smells identifiés à l'état initial.

Ainsi, bien que le projet soit fonctionnel d'un point de vue utilisateur, il demeurait **fragile sur le plan technique**, difficile à faire évoluer et exposé à des risques importants lors de futures améliorations ou corrections.

Besoins identifiés

À la suite de l'analyse de l'état initial du projet, plusieurs besoins techniques prioritaires ont été identifiés afin d'assurer la fiabilité, la maintenabilité et l'évolutivité de la plateforme **Swantrad**.

Stabilisation de l'environnement de développement

Le premier besoin concernait la stabilisation de l'environnement de développement local. En effet, le projet ne pouvait pas être exploité correctement en local sans de nombreuses adaptations, ce qui constituait un frein majeur au travail de maintenance et d'amélioration du code.

Il a donc été nécessaire de résoudre divers problèmes de configuration liés aux versions de PHP, aux extensions manquantes et aux incompatibilités entre dépendances. Ce travail a également impliqué l'importation et la correction manuelle de la base de données à partir de **dumps** existants, ainsi que la réparation du **bootstrap Laravel** et des chemins de chargement. Enfin, la mise en place d'un workflow de développement fonctionnel, incluant le serveur local, la gestion des assets et l'utilisation de **Vite**¹, était indispensable pour travailler dans des conditions professionnelles.

Réduction de la dette technique

L'analyse initiale du projet à l'aide de l'outil **SonarQube** a mis en évidence une dette technique importante. Celle-ci se traduisait notamment par un grand nombre de problèmes de maintenabilité, des duplications de code à forte sévérité, des structures conditionnelles complexes et peu lisibles, ainsi que des constantes dupliquées et des fonctions présentant une complexité cognitive élevée.

Face à ces constats, un besoin central s'est imposé : *réduire la dette technique existante* tout en garantissant la stabilité fonctionnelle de l'application. Cela impliquait la mise en œuvre de refactorisations ciblées, progressives et mesurables, afin d'améliorer la qualité du code sans introduire de régressions.

Nettoyage du code et suppression du code mort

L'étude approfondie du projet a révélé la présence de nombreux éléments inutilisés, tels que des contrôleurs jamais appelés, des vues Blade non référencées, des fichiers CSS chargés sans être

¹ Vite est un outil de build et de serveur de développement front-end.

exploités, ainsi que des duplications d'identifiants HTML susceptibles de provoquer des comportements imprévisibles côté interface.

Un besoin essentiel consistait donc à *nettoyer le projet* en identifiant précisément ces éléments afin de les supprimer de manière sécurisée. Cette étape devait être réalisée avec précaution, afin d'éviter toute suppression de composants utilisés dynamiquement ou indirectement par l'application.

Sécurisation des évolutions futures par les tests

Enfin, la couverture de tests existante demeurait limitée et aucune stratégie complète de *tests unitaires et d'intégration* n'était formalisée. Les tests présents ne permettaient pas de sécuriser efficacement l'ensemble des fonctionnalités critiques, ce qui rendait chaque modification du code potentiellement risquée.

L'introduction d'une *culture de tests* s'est donc révélée indispensable. Elle avait pour objectifs de prévenir les régressions, de sécuriser les évolutions futures du projet et de répondre aux exigences professionnelles ainsi qu'aux attendus pédagogiques de l'épreuve E5 du BTS SIO, qui évalue la capacité de l'étudiant à mobiliser ses compétences informatiques dans un contexte professionnel réel.

Travail à réaliser durant le stage

Afin de répondre aux besoins, plusieurs axes de travail complémentaires m'ont été confiés dans le cadre du stage. Ces missions avaient pour objectif d'améliorer la stabilité, la qualité et la pérennité de la plateforme *Swantrad*.

Mise en place et stabilisation de l'environnement local

La première mission consistait à *mettre en place un environnement de développement local pleinement fonctionnel*. Cette étape devait débuter par le clonage du dépôt *GitHub* et la prise en main du projet existant. Il était ensuite nécessaire de configurer l'ensemble de l'environnement de développement, notamment en résolvant les incompatibilités liées aux versions de *PHP*, à *Composer*² et aux extensions requises.

Cette mission incluait également l'importation de la base de données à partir de *dumps existants*, ainsi que sa correction afin d'assurer sa cohérence avec le code applicatif. Enfin, la mise en place d'un workflow de développement stable était attendue, intégrant le serveur local, la gestion des ressources et l'utilisation de *Vite*, afin de travailler dans des conditions proches d'un environnement professionnel.

Analyse de la qualité du code existant

Une fois l'environnement stabilisé, il était prévu de réaliser une *analyse approfondie de la qualité du code existant*. Pour cela, l'outil **SonarQube** devait être mis en place afin d'évaluer objectivement

² *Composer* est un gestionnaire de dépendances de *PHP*, qui permet d'installer et de maintenir les bibliothèques nécessaires au projet.

l'état du projet. Cette analyse devait permettre d'identifier les duplications de code, les *code smells*³ ainsi que les principaux problèmes de maintenabilité.

En parallèle, un recensement du code mort devait être effectué, incluant les contrôleurs, les vues *Blade* et les fichiers CSS non utilisés. L'ensemble de ces constats devait aboutir à la rédaction d'un *plan d'amélioration priorisé*, reposant sur une analyse croisant l'impact des actions proposées et le risque de régression fonctionnelle.

Refactorisation et amélioration de la maintenabilité

Sur la base du plan d'amélioration défini, un travail de *refactorisation progressive* devait être engagé. Celui-ci consistait notamment à supprimer de manière sécurisée le code mort identifié, tout en veillant à préserver les fonctionnalités existantes.

Des actions ciblées devaient également être menées afin de réduire les duplications de code en HTML, CSS et PHP. L'introduction de constantes visait à éliminer les littéraux dupliqués et à améliorer la cohérence globale du code. L'ensemble de ces interventions avait pour objectif d'améliorer la lisibilité, la structure et la maintenabilité du projet, chaque modification devant être vérifiée fonctionnellement.

Amélioration et renforcement des tests automatisés

Enfin, une mission importante du stage portait sur l'amélioration et le renforcement des tests automatisés existants. Cette tâche devait comprendre la prise en main des outils de test fournis par Laravel et PHPUnit, ainsi que la création et la correction de tests unitaires ciblant les *helpers* et certaines fonctions critiques du projet.

Des tests d'intégration devaient également être développés ou ajustés afin de valider plusieurs parcours utilisateurs clés. Cette mission devait impliquer la résolution de problèmes techniques bloquants empêchant l'exécution correcte des tests, notamment en lien avec la configuration de l'environnement et la base de données.

Une fois ces obstacles levés, l'automatisation de l'exécution des tests et leur intégration au workflow CI/CD a permis d'assurer une validation régulière du code avant intégration et de réduire les risques de régression.

Analyse de l'étude existante et l'étude d'opportunité

Analyse de l'existant

Avant d'engager les travaux techniques, une analyse de l'existant a été réalisée afin de comprendre le fonctionnement global de la plateforme *Swantrad*, son niveau de maturité et ses limites techniques.

Sur *le plan fonctionnel*, la plateforme répondait à son objectif principal : proposer un environnement numérique permettant aux traducteurs assermentés de gérer leurs missions, leurs documents et la

³ Un *code smell* est un signal indiquant un problème potentiel de conception du code, affectant sa lisibilité et sa maintenabilité sans empêcher son fonctionnement.

certification des traductions via des services conformes au *règlement eIDAS*. Le projet était donc opérationnel en production et déjà utilisé par des professionnels, constituant une base solide.

En revanche, l'*analyse technique* a mis en évidence plusieurs fragilités structurelles. La présence d'une dette technique importante, l'existence de code mort, ainsi que des duplications de logique rendaient le projet plus complexe à maintenir et à faire évoluer. Par ailleurs, bien que quelques tests automatisés soient déjà en place, leur couverture restait limitée et ils n'étaient pas pleinement intégrés dans un processus de validation continue. Cette situation ne permettait pas de sécuriser efficacement l'ensemble des fonctionnalités critiques de l'application.

De plus, l'environnement de développement local nécessitait des adaptations spécifiques, compliquant le travail des développeurs et rendait certaines phases de validation plus longues que prévu.

Cette situation impliquait un risque important : toute évolution fonctionnelle ou correction pouvait entraîner des régressions, ralentir le développement et augmenter les coûts de maintenance à moyen et long terme.

Étude d'opportunité

L'étude d'opportunité a permis de démontrer l'intérêt de mener un travail de stabilisation et d'amélioration technique sur le projet *Swantrad*, aussi bien d'un point de vue technique que stratégique.

Sur *le plan technique*, la réduction de la dette technique, le nettoyage du code et la mise en place de tests automatisés étaient nécessaires pour fiabiliser la plateforme. Ces actions permettaient d'améliorer la qualité du code, de faciliter sa compréhension et de sécuriser les évolutions futures, notamment dans un contexte de croissance du produit.

Sur *le plan organisationnel*, l'introduction d'outils tels que *SonarQube* et d'un pipeline *CI/CD* offrait la possibilité d'industrialiser le développement, en automatisant les contrôles de qualité et en réduisant les risques liés aux mises en production.

Enfin, sur *le plan stratégique*, *Swantrad* évolue dans un secteur à forts enjeux juridiques et réglementaires, où la fiabilité, la traçabilité et la sécurité des documents sont primordiales. Disposer d'une base technique saine et maîtrisée constitue donc un levier essentiel pour accompagner la montée en charge de la plateforme et l'intégration de futures fonctionnalités.

Planning prévisionnel

Un planning prévisionnel a été établi en début de stage afin d'organiser les différentes missions à réaliser et de structurer l'avancement du travail sur la durée du stage. Ce planning, représenté sous la forme d'un *diagramme de Gantt*, permettait de visualiser la répartition des tâches dans le temps et d'identifier les grandes phases du projet.

Le stage était initialement découpé en plusieurs étapes successives. Une première phase était dédiée à la prise en main du projet et à la mise en place de l'environnement de développement local. Elle était suivie d'une phase d'analyse de l'existant, comprenant l'étude de la qualité du code et l'identification

de la dette technique. Une troisième phase concernait les travaux de refactorisation et de nettoyage du code, avant d'aborder la mise en place des tests automatisés et leur intégration au workflow CI/CD.

Ce planning prévisionnel a servi de cadre de référence tout au long du stage, tout en restant suffisamment flexible pour s'adapter aux contraintes techniques rencontrées et aux priorités définies au fil de l'avancement du projet.

Le diagramme de Gantt correspondant se trouve en *Figure 1 (cf Figure 1 : Diagramme de Gantt)* de ce rapport.

Mise en place et configuration de solutions logicielles

Dans le cadre du stage, plusieurs solutions logicielles ont été mises en place et configurées afin de permettre le développement, l'analyse et la sécurisation du projet *Swantrad*. Ces actions ont concerné aussi bien l'environnement de développement local que les outils de qualité et d'automatisation utilisés dans un contexte professionnel.

Mise en place de l'environnement de développement local

La première étape a consisté à stabiliser et configurer *l'environnement de développement local*, indispensable au travail sur le projet. Celui-ci reposait sur une architecture web classique basée sur *PHP* et le framework *Laravel*.

Cette phase a nécessité l'installation et la configuration des éléments suivants :

- le langage *PHP*, avec les extensions requises par *Laravel*,
- le gestionnaire de dépendances *Composer* pour l'installation des bibliothèques du projet,
- un serveur local permettant l'exécution de l'application,
- une base de données relationnelle pour la gestion des utilisateurs, missions et documents.

L'importation de la base de données a été réalisée à partir de *dumps* existants, puis ajustée afin d'assurer la cohérence entre le schéma de données et le code applicatif. Cette configuration a permis d'obtenir un environnement local fonctionnel, proche de celui utilisé en production.

Configuration de la gestion des ressources avec Vite

La gestion des fichiers front-end (CSS et JavaScript) reposait sur l'outil *Vite*, intégré nativement à *Laravel*. *Vite* a été configuré afin d'assurer le chargement correct des ressources en environnement de développement et leur compilation pour la production.

Cette configuration était essentielle pour garantir le bon fonctionnement de l'interface utilisateur et éviter des erreurs liées aux fichiers manquants ou mal référencés. Elle a également permis d'améliorer les performances et le confort de développement grâce au rechargement automatique des fichiers lors des modifications.

Utilisation des outils d'analyse et de qualité du code

Dans une démarche d'amélioration continue, l'outil *SonarQube*, déjà présent dans l'environnement du projet, a été utilisé afin d'analyser la qualité du code de *Swantrad*. Cet outil permet d'identifier objectivement les duplications, les code smells et les problèmes de maintenabilité.

Son exploitation a constitué une étape essentielle du stage. Les résultats fournis par *SonarQube* ont servi de base à la priorisation des actions correctives, en orientant les travaux de refactorisation vers les zones les plus critiques du projet.

Configuration et amélioration de la solution de tests automatisés

Le projet disposait déjà d'une base de tests automatisés. Toutefois, leur configuration et leur intégration nécessitaient des ajustements afin de garantir leur fiabilité.

Les outils fournis par *Laravel et PHPUnit* ont ainsi été configurés et adaptés afin de permettre l'exécution stable de tests unitaires et de tests d'intégration. Cette étape a impliqué l'adaptation de certains paramètres, notamment au niveau de la base de données de test et des variables d'environnement, afin d'assurer des tests reproductibles et isolés.

Cette amélioration du dispositif existant a permis de sécuriser davantage les évolutions du code et de réduire les risques de régression.

Exploitation et fiabilisation du pipeline CI/CD

Un pipeline *CI/CD* basé sur *GitHub Actions* était déjà en place au sein du projet. Dans le cadre du stage, ce pipeline a été analysé, adapté et fiabilisé afin de garantir une exécution correcte des tests dans un environnement Linux.

Des ajustements ont notamment été réalisés pour corriger des problèmes liés à la sensibilité à la casse des fichiers, aux migrations de base de données et à la configuration des variables d'environnement.

Grâce à ces améliorations, chaque évolution du projet est désormais validée par une série de contrôles automatisés avant intégration, contribuant ainsi à renforcer la fiabilité et la stabilité globale de la plateforme *Swantrad*.

Développement et maintenance d'applications

Dans le cadre de ce stage, mon travail ne portait pas sur le développement de nouvelles fonctionnalités, mais principalement sur la *maintenance évolutive et corrective* de l'application web *Swantrad*.

Cette maintenance avait pour objectif d'*améliorer la qualité du code existant, de réduire la dette technique et de fiabiliser l'application*, tout en conservant un comportement fonctionnel strictement identique pour les utilisateurs. Les interventions ont concerné plusieurs couches de l'application *Laravel*, notamment les vues *Blade*, les *helpers PHP*, certains *contrôleurs* ainsi que les *feuilles de styles CSS*.

Les actions réalisées ont notamment consisté à supprimer de manière sécurisée du code mort, tel que des *vues Blade et des fichiers CSS non utilisés*, à corriger des *duplications de code* signalées par l'outil

SonarQube, et à améliorer la lisibilité et la structure du code par l'ajout d'accolades, la factorisation de certaines logiques et l'introduction de constantes.

Des refactorisations ciblées ont également été menées sur des fichiers CSS volumineux, avec l'adoption d'une approche *mobile-first*, tout en veillant à préserver le rendu visuel existant. Chaque modification a fait l'objet de vérifications fonctionnelles systématiques afin de garantir l'absence de régression.

Cette approche de maintenance progressive, documentée et testée, a permis d'améliorer durablement la maintenabilité du projet sans introduire de dysfonctionnements fonctionnels.

Gestion de bases de données

La base de données du projet **Swantrad** est volumineuse et complexe, car elle couvre de nombreux aspects métier tels que la gestion des utilisateurs, des traducteurs, des services, des commandes, des documents, des paiements et des certifications. Dans ce contexte, mon stage ne portait pas sur la conception d'un nouveau modèle de données (MCD/MLD), mais sur la *compréhension, l'exploitation et la fiabilisation de l'existant*.

Le travail réalisé a principalement consisté à analyser la structure de la base de données à travers les *modèles Eloquent⁴ de Laravel*, à identifier les incohérences entre les migrations, le schéma réel et les tests automatisés, ainsi qu'à importer et corriger des dumps de base de données afin de permettre un fonctionnement stable en environnement local.

Des ajustements ont également été apportés à certaines migrations et factories afin d'assurer la cohérence entre les données attendues par l'application et celles utilisées lors de l'exécution des tests. Enfin, une base de données dédiée aux tests a été configurée afin de garantir une isolation stricte entre les différents environnements.

Cette démarche m'a permis de mieux comprendre les relations entre les entités du projet et de prendre conscience de l'importance d'une cohérence constante entre le code applicatif, les migrations et les données, en particulier dans un contexte de tests automatisés et d'intégration continue.

Tests et validations

Une part essentielle du stage a été consacrée à *l'amélioration et à la fiabilisation du dispositif de tests existant*. Avant le début du stage, le projet disposait déjà de quelques tests automatisés, mais leur couverture restait limitée et leur intégration dans un processus de validation continue n'était pas pleinement opérationnelle.

Les travaux réalisés ont permis d'enrichir significativement cette base en développant des *tests unitaires* ciblant des fonctions critiques, notamment les *helpers*, les *mécanismes d'extraction de texte et le formatage des données*. Des *tests d'intégration* ont également été conçus afin de couvrir des parcours utilisateurs essentiels, tels que *l'authentification, l'inscription et l'accès aux différentes interfaces de l'application*. En complément, des tests de non-régression ont été mis en place afin de sécuriser les corrections apportées lors des opérations de maintenance.

⁴ Les modèles Eloquent constituent la couche de liaison entre le code applicatif et la base de données.

Un environnement de test dédié a été configuré, reposant sur une base de données isolée, garantissant ainsi l'indépendance des tests par rapport aux environnements de développement et de production. Par ailleurs, des *scénarios de test détaillés* ainsi qu'une *documentation d'exécution* ont été rédigés afin de faciliter leur compréhension, leur utilisation et leur maintenance.

L'ensemble des tests a ensuite été intégré dans le *pipeline CI/CD via GitHub Actions*, permettant leur exécution automatique dans un environnement Linux standardisé. À l'issue de ces travaux, l'intégralité des 278 tests automatisés s'exécute avec succès, représentant 977 assertions validées, contribuant ainsi à renforcer la fiabilité du projet et à sécuriser ses évolutions futures.

Support aux utilisateurs et accompagnement technique

Bien que le stage ne comprenne pas de support utilisateur direct au sens classique (hotline ou assistance client), j'ai participé à un *support technique interne* auprès de l'équipe projet.

Ce support s'est notamment traduit par la documentation de l'*architecture technique du projet*, la *rédaction de guides techniques (tests, CI/CD, résolution d'erreurs)*, ainsi que par la traçabilité des corrections à l'aide d'un *historique Git clair et commenté*. Des outils et des procédures ont également été mis à disposition afin de faciliter la prise en main du projet par d'autres développeurs.

L'ensemble de ces actions contribue à améliorer la transmission des connaissances, la maintenabilité du projet et l'autonomie des futurs intervenants techniques sur la plateforme *Swantrad*.

Analyse critique des activités

Difficultés rencontrées

Au cours du stage, plusieurs difficultés ont été rencontrées, principalement liées à la reprise d'un projet existant complexe, à son historique technique et à la mise en place de bonnes pratiques de qualité logicielle sur une application déjà en production.

Prise en main globale du projet Swantrad

L'application avait été développée sur une période relativement longue par différentes équipes, ce qui entraînait une hétérogénéité du code et une architecture parfois difficile à appréhender. Certaines parties du projet ont nécessité une analyse approfondie afin de comprendre les interactions entre les contrôleurs, les *modèles Eloquent*, les *vues Blade* et la base de données.

Cette phase de compréhension, bien qu'indispensable, a demandé un investissement important en temps et en rigueur.

Configuration et stabilisation de l'environnement de développement local

Le projet ne pouvait pas être exécuté correctement sans ajustements spécifiques, notamment en raison d'incompatibilités entre les versions de *PHP*, les extensions requises, les dépendances *Composer* et certaines migrations de base de données. Des erreurs bloquantes apparaissaient parfois uniquement

lors de l'exécution de certaines fonctionnalités ou des tests, ce qui compliquait leur diagnostic et leur résolution.

Mise en place et fiabilisation des tests automatisés

De nombreux tests échouaient initialement en environnement d'intégration continue (*GitHub Actions*), alors qu'ils fonctionnaient en local. Ces écarts étaient principalement dus aux différences entre les environnements *Windows* et *Linux*, notamment la sensibilité à la casse des noms de fichiers et de classes, ainsi qu'à des configurations d'environnement non explicitement définies. L'identification et la correction de ces problèmes ont nécessité une compréhension approfondie du cycle de vie applicatif et du fonctionnement du *pipeline CI/CD*.

Gestion de la dette technique existante

Cette difficulté a été constante tout au long du stage. Ces opérations de nettoyage du code, de refactorisation et de correction des duplications devaient être menées avec prudence afin d'éviter toute régression fonctionnelle sur une application déjà utilisée en production. Cela impliquait de tester systématiquement chaque modification et de s'appuyer sur des tests de non-régression pour garantir la stabilité du projet.

Complexité et taille de la base de données du projet

Certaines incohérences entre les migrations, les modèles et les tests ont dû être identifiées et corrigées, notamment lors de l'exécution des tests automatisés en environnement *CI*. La nécessité de maintenir une cohérence stricte entre le schéma de base de données, le code applicatif et les données de test a exigé une attention particulière.

Ces différentes difficultés ont parfois ralenti l'avancement des travaux, mais elles ont surtout permis de développer une méthodologie de travail rigoureuse, basée sur l'analyse, la documentation, la validation systématique et l'amélioration continue de la qualité du projet.

Solutions mises en œuvre

Afin de répondre aux difficultés rencontrées durant le stage, une série de solutions techniques et organisationnelles a été progressivement mise en œuvre, avec pour objectif principal de stabiliser le projet, d'en améliorer la qualité et de sécuriser son évolution dans un contexte de production.

Démarche d'analyse structurée de l'existant

Face à la complexité du projet et à l'hétérogénéité du code issue de développements successifs, une démarche d'analyse structurée de l'existant a été mise en place afin de faciliter la prise en main globale de l'application.

Cette démarche s'est appuyée sur une étude approfondie des différentes couches du projet, notamment les contrôleurs, les modèles Eloquent, les vues Blade et leurs interactions avec la base de données. L'objectif était de comprendre précisément l'architecture applicative ainsi que les dépendances entre les composants, afin de pouvoir intervenir de manière sécurisée et cohérente.

Afin de formaliser cette compréhension et de capitaliser sur le travail d'analyse réalisé, des notes techniques et des documents internes ont été rédigés. Cette documentation a permis de structurer

progressivement la vision du projet et de réduire les risques d'erreur lors des phases ultérieures de maintenance et de refactorisation.

Stabilisation et normalisation de la configuration locale

Un travail spécifique a été entrepris pour répondre aux difficultés liées à l'environnement de développement. Les incompatibilités de versions de *PHP*, les dépendances manquantes et les problèmes liés aux migrations ont été identifiés puis corrigés, permettant d'obtenir un environnement cohérent et fonctionnel. La vérification systématique des dépendances *Composer* et l'utilisation de configurations explicites ont contribué à réduire les erreurs liées à l'exécution du projet et à fiabiliser les phases de développement.

Prise en compte de la cohérence entre les environnements

Face aux écarts constatés entre l'exécution locale et l'environnement d'intégration continue, une attention particulière a été portée à la cohérence entre les environnements. Les problèmes de sensibilité à la casse des fichiers et des classes ont été corrigés en respectant strictement les conventions de nommage du framework *Laravel*. Les variables d'environnement nécessaires à l'exécution des tests ont été explicitement définies, garantissant un comportement reproductible du projet quel que soit l'environnement d'exécution.

Fiabilisation des tests automatisés

La fiabilisation des tests automatisés a constitué un axe central des solutions mises en œuvre. Des tests unitaires et d'intégration ont été créés ou corrigés afin de couvrir les fonctionnalités critiques de l'application. Des tests de non-régression ont été ajoutés pour sécuriser les opérations de maintenance et s'assurer que les corrections apportées n'altéraient pas le comportement existant. Une base de données dédiée aux tests a également été configurée afin d'isoler complètement les environnements et d'éviter toute interaction involontaire avec les données de production.

Approche progressive et mesurée de la maintenance

Pour traiter la dette technique identifiée, une approche progressive et mesurée de la maintenance a été privilégiée. Le code inutile ou obsolète a été supprimé de manière contrôlée, les duplications ont été réduites et la lisibilité du code améliorée grâce à la factorisation et à l'introduction de constantes. Chaque intervention a été accompagnée de vérifications fonctionnelles et de tests automatisés afin de garantir la stabilité de l'application.

Enfin, l'ensemble de ces actions a été intégré dans un *pipeline CI/CD* basé sur *GitHub Actions*, permettant l'automatisation des contrôles qualité et de l'exécution des tests à chaque modification du code. Cette automatisation assure désormais une validation systématique avant toute intégration, réduisant significativement les risques de régression et contribuant à l'amélioration continue de la qualité du projet *Swantrad*.

Compétences acquises et développées

Ce stage m'a permis de développer et de consolider de nombreuses compétences techniques et professionnelles, en lien direct avec les exigences d'un projet informatique réel et les attendus de ma formation.

Compétences en maintenance applicative

Sur le plan technique, j'ai renforcé mes compétences en maintenance applicative au sein d'un projet existant basé sur le *framework Laravel*. J'ai appris à analyser un code complexe, à comprendre une architecture applicative déjà en place et à intervenir de manière ciblée afin d'améliorer la qualité du code sans modifier le comportement fonctionnel de l'application. Cette expérience m'a permis de mieux appréhender les notions de *dette technique*, de *refactorisation* et de *maintenabilité*.

Compétences en gestion de bases de données

Le stage m'a également permis de développer des compétences en gestion de bases de données, notamment à travers l'analyse des *modèles Eloquent*, des migrations et de la cohérence entre le schéma de données, le code applicatif et les tests. J'ai pris conscience de l'importance de la fiabilité des données et de la nécessité d'adapter les environnements (local, test, intégration continue) afin de garantir un fonctionnement stable et un comportement reproductible quel que soit l'environnement.

Compétences en tests automatisés et en validation logicielle

Une part importante des compétences acquises concerne les tests automatisés et la validation logicielle. J'ai appris à concevoir et à mettre en place des *tests unitaires*, des *tests d'intégration* et des *tests de non-régression*, ainsi qu'à configurer un environnement de test isolé. L'intégration de ces tests dans un *pipeline CI/CD* m'a permis de mieux comprendre les enjeux de l'automatisation, de la fiabilité et de la sécurisation des évolutions d'un projet.

Compétences en gestion de versions et collaboration avec Git

Ce stage m'a également permis d'approfondir ma maîtrise de l'outil *Git* dans un contexte professionnel. J'ai utilisé régulièrement les principales *commandes de gestion de versions* (clone, add, commit, pull, push, merge, résolution de conflits) afin de suivre l'évolution du code et d'assurer la traçabilité des modifications réalisées.

La gestion des branches et l'analyse de l'historique des commits m'ont permis de mieux comprendre l'organisation du travail collaboratif et l'importance d'une documentation claire des changements apportés. Cette pratique m'a sensibilisée à la rigueur nécessaire dans la rédaction des messages de commit et dans la gestion des mises à jour entre environnements.

Approche méthodologique et organisationnelle

Sur le plan méthodologique et organisationnel, ce stage m'a permis de développer une approche rigoureuse du travail, basée sur l'analyse préalable, la documentation, la vérification systématique des modifications et l'amélioration continue. J'ai également renforcé mes compétences dans l'utilisation d'outils professionnels tels que *GitHub*, *SonarQube* et *GitHub Actions*.

Enfin, ce stage a contribué au développement de compétences transversales importantes, telles que l'autonomie, la capacité d'adaptation face à des problématiques techniques complexes, ainsi que la communication technique à travers la rédaction de documentation et de guides destinés à d'autres développeurs.

Comparaison avec les attentes initiales

Au début du stage, les attentes principales portaient sur l'*amélioration de la qualité globale du code, la réduction de la dette technique et la mise en place de pratiques permettant de sécuriser durablement les évolutions du projet Swantrad*. L'objectif n'était pas de corriger l'ensemble des anomalies existantes, mais de cibler en priorité les problématiques les plus critiques en termes de maintenabilité et de fiabilité, tout en garantissant la stabilité d'une application déjà en production.

L'analyse initiale réalisée à l'aide de *SonarQube* mettait en évidence un nombre important de duplications de code, de *code smells* à forte sévérité et de problèmes de maintenabilité. Ces indicateurs ont servi de base de référence afin de définir les axes de travail et de prioriser les actions à mener durant le stage. (*cf Figure 2 : Analyse SonarQube – État initial du projet (avant le stage)*)

Au cours du stage, des actions correctives ont été menées sur certaines zones ciblées du projet, notamment en matière de nettoyage de code, de réduction de duplications localisées et d'amélioration de la lisibilité. Toutefois, les indicateurs globaux n'ont pas tous évolué de manière significative. Si certaines métriques générales (comme le nombre total de Code Smells ou le rating global) restent relativement stables, des améliorations ciblées et mesurables ont été obtenues, notamment la réduction de 135 Code Smells de sévérité High, la suppression des 66 IDs HTML dupliqués et la diminution du taux de duplication de code de 4,6 % à 4,1 %. (*cf Figure 3 : Analyse SonarQube – État du projet en fin du stage*)

Il convient également de souligner que l'évolution du projet durant cette période ne repose pas uniquement sur les actions menées dans le cadre du stage, mais sur le travail collectif de l'équipe technique. Dans ce contexte, les améliorations apportées ont contribué à renforcer certaines zones critiques du code, même si leur impact n'apparaît pas toujours de manière visible dans les métriques globales de SonarQube.

En revanche, des avancées concrètes et mesurables ont été réalisées en matière de fiabilisation des tests et d'automatisation des validations. Les captures d'écran présentées dans ce rapport illustrent notamment l'exécution réussie des tests en environnement local (*cf Figure 4 : Résultat de l'exécution des tests en local (« php artisan test »*)) ainsi que dans le pipeline GitHub Actions. (*cf Figure 5 : Exécution des tests dans le pipeline GitHub Actions*) Ces éléments démontrent la mise en place d'un cadre de validation plus structuré et plus sécurisé qu'en début de stage.

Ainsi, même si les indicateurs globaux n'ont pas connu une évolution spectaculaire, les objectifs principaux du stage — stabilisation de l'environnement, amélioration ciblée du code et sécurisation des évolutions — ont été atteints de manière cohérente et progressive.

Apport du stage et perspectives

Ce stage au sein de la startup *Swantrad* a constitué une expérience déterminante dans mon parcours de formation. Il m'a permis de découvrir le fonctionnement d'un projet informatique réel, déjà en production, et d'évoluer dans un environnement professionnel exigeant, où les choix techniques ont un impact direct sur la stabilité et l'évolution de la plateforme.

Sur le plan personnel et professionnel, cette immersion m'a apporté une meilleure compréhension des attentes du monde de l'entreprise, notamment en matière de rigueur, d'organisation et de

responsabilité. Le fait de travailler sur une application utilisée par des professionnels m'a sensibilisée à l'importance de livrer un travail fiable, documenté et durable, et m'a permis de développer une posture plus professionnelle face aux problématiques rencontrées.

En termes de perspectives, les travaux réalisés durant le stage ont contribué à renforcer les bases techniques du projet **Swantrad**. Les actions menées en matière de qualité du code, de maintenance et de sécurisation des évolutions offrent désormais un cadre plus sain pour la poursuite du développement de la plateforme. Ces améliorations pourront être prolongées par des actions futures telles que la poursuite du refactoring, l'enrichissement de la couverture de tests et l'optimisation continue des performances et de la sécurité.

Enfin, ce stage s'inscrit comme une étape importante dans la construction de mon projet professionnel. Il m'a permis de mieux appréhender les réalités du métier et de confirmer mon intérêt pour des environnements techniques exigeants, orientés qualité et fiabilité.

Critique du planning

Un planning prévisionnel avait été établi en début de stage afin d'organiser les différentes missions à réaliser et de structurer l'avancement du travail. Ce planning constituait un cadre de référence, tout en restant adaptable en fonction des contraintes techniques rencontrées.

Au cours du stage, *plusieurs difficultés imprévues*, notamment liées à la prise en main du projet, à la stabilisation de l'environnement de développement et à la mise en place des tests unitaires et des tests d'intégration, *ont eu un impact sur le rythme d'avancement des travaux*. Le temps nécessaire à l'analyse des problèmes et à la recherche de solutions adaptées a parfois été plus important que prévu, ce qui a limité la possibilité d'aborder certaines améliorations complémentaires initialement envisagées.

Cependant, ces ralentissements ont été compensés par une priorisation rigoureuse des tâches et une adaptation progressive du planning. *Les objectifs définis en accord avec le tuteur de stage ont été respectés*, en privilégiant les actions à plus forte valeur ajoutée pour le projet, notamment en matière de qualité du code, de fiabilisation des tests et de sécurisation des évolutions.

Ainsi, même si l'ensemble des actions souhaitées n'a pas pu être réalisé dans le temps imparti, le planning initial a globalement été tenu. Cette expérience m'a permis de mieux appréhender la gestion du temps dans un contexte professionnel réel et de comprendre l'importance de l'adaptation et de la priorisation face aux contraintes techniques.

Bilan des objectifs atteints

Les objectifs définis en début de stage visaient principalement à améliorer la qualité, la stabilité et la maintenabilité du projet **Swantrad**, tout en sécurisant ses évolutions dans un contexte de production. À l'issue du stage, le bilan peut être considéré comme globalement positif, les objectifs prioritaires ayant été atteints.

La stabilisation de l'environnement de développement local a permis de disposer d'un cadre de travail fonctionnel et cohérent, facilitant les opérations de maintenance et d'amélioration du code. Les

problèmes liés à la configuration, aux dépendances et aux migrations ont été identifiés puis corrigés, rendant le projet exploitable dans des conditions professionnelles.

L'analyse de la qualité du code à l'aide de *SonarQube* a permis d'identifier les zones à fort impact technique et de prioriser les actions à mener. Les travaux de maintenance et de refactorisation ont contribué à améliorer la lisibilité et la structure du code, avec une réduction mesurable de 135 Code Smells de sévérité High et la suppression complète des 66 IDs HTML dupliqués identifiés dans l'analyse initiale.

La fiabilisation des tests automatisés constitue également un objectif majeur atteint. Des tests unitaires, d'intégration et de non-régression ont été développés ou renforcés, puis intégrés dans un pipeline *CI/CD*, permettant désormais une validation systématique du code à chaque évolution. Cette automatisation renforce la fiabilité du projet et limite les risques de régression.

Enfin, même si l'ensemble des améliorations envisagées initialement n'a pas pu être réalisé dans le temps imparti, les actions menées ont apporté une valeur ajoutée au projet. Elles ont permis de consolider les bases techniques existantes et de fournir à l'équipe de développement des outils ainsi qu'une méthodologie favorisant une amélioration continue de la plateforme *Swantrad*.

Apport du stage en termes de compétences et de connaissances

Sur le plan des compétences et des connaissances, ce stage m'a permis d'approfondir de manière significative mes acquis théoriques en les confrontant à des situations concrètes. J'ai développé une meilleure compréhension du fonctionnement d'une application web complexe, notamment en matière de maintenance applicative et de gestion d'un code existant.

J'ai renforcé mes connaissances liées au *framework Laravel*, en particulier sur l'architecture d'une application, l'utilisation des *modèles Eloquent*, la *gestion des migrations* et *l'interaction avec la base de données*. Le travail réalisé autour de la *dette technique* et de la *refactorisation* m'a permis de mieux comprendre les enjeux de maintenabilité et de lisibilité du code dans un projet de grande taille.

Le stage m'a également permis de consolider mes compétences en gestion de versions avec *Git*. L'utilisation régulière des commandes de suivi et de gestion du code (*add, commit, pull, push, merge, résolution de conflits*) m'a sensibilisée à l'importance de la traçabilité des modifications, de la rigueur dans la rédaction des messages de commit et du travail collaboratif au sein d'une équipe technique

Par ailleurs, j'ai développé des compétences en matière de tests automatisés et de validation logicielle. La *conception de tests unitaires* et de *tests d'intégration*, ainsi que leur intégration dans un pipeline *CI/CD*, m'a offert une vision concrète des bonnes pratiques utilisées dans le développement logiciel moderne.

Enfin, ce stage m'a permis de développer des compétences transversales essentielles, telles que *l'autonomie*, *la capacité d'analyse*, *la résolution de problèmes techniques* complexes et *la communication technique* à travers la rédaction de documentation et de guides destinés à d'autres développeurs.

Propositions d'amélioration pour l'entreprise

À l'issue du stage, plusieurs pistes d'amélioration peuvent être envisagées afin de prolonger la démarche engagée et de renforcer durablement la qualité du projet *Swantrad*.

Sur le plan technique, une première amélioration consisterait à poursuivre la *réduction de la dette technique identifiée* par les outils d'analyse. Bien que des actions ciblées aient déjà été menées, la taille et la complexité du projet justifient la mise en place d'un plan d'amélioration à moyen terme, reposant sur une priorisation régulière des actions à fort impact.

Il serait également pertinent d'enrichir progressivement la couverture des tests automatisés, notamment sur les parcours utilisateurs critiques et les fonctionnalités métier sensibles. Cette démarche contribuerait à renforcer la fiabilité de l'application et à sécuriser durablement ses évolutions.

Enfin, l'exploitation régulière et formalisée des indicateurs fournis par *SonarQube* ainsi que des résultats du *pipeline CI/CD* pourrait être renforcée au sein du processus de développement, notamment par la mise en place de revues périodiques ou d'objectifs qualité mesurables. Cette démarche permettrait d'ancrer durablement la plateforme dans une logique d'amélioration continue de la qualité logicielle.

Perspectives d'avenir (professionnel, poursuite d'études)

Ce stage a constitué une *expérience très enrichissante*, tant sur le plan technique que personnel. Il m'a permis de mieux comprendre les exigences du monde professionnel et de mettre en pratique les connaissances acquises au cours de ma formation dans un contexte réel.

Les compétences développées durant ce stage représentent une base solide pour la poursuite de mon parcours de formation. Les notions abordées, telles que *la maintenance applicative, l'analyse de la qualité du code et les tests automatisés*, constituent des acquis importants qui me seront utiles dans mes futurs apprentissages et projets.

Sur le plan professionnel, cette expérience m'a permis de *gagner en maturité, en autonomie et en capacité d'analyse*, tout en me donnant une vision plus concrète des réalités du travail en entreprise. Elle m'a également permis de mieux identifier les domaines qui m'intéressent, sans pour autant m'enfermer dans une orientation définitive à ce stade de mon parcours.

Ainsi, ce stage s'inscrit comme une étape formatrice et positive, qui contribuera à nourrir mes réflexions et mes choix futurs, quels qu'ils soient.

Conclusion

Synthèse des principaux points

Ce stage réalisé au sein de la startup *Swantrad* s'est inscrit dans un contexte de *maintenance et d'amélioration de la qualité d'un projet informatique existant*, déjà déployé en production. Dès le début du stage, l'analyse de l'existant a mis en évidence plusieurs fragilités techniques, notamment une dette technique importante, une couverture de tests limitée, des duplications de code et des difficultés liées à la configuration de l'environnement de développement.

Afin de répondre à ces constats, les travaux menés se sont articulés autour de plusieurs axes majeurs : la stabilisation de l'environnement de développement local, l'exploitation des outils d'analyse de qualité tels que *SonarQube*, la réduction ciblée de certaines zones de dette technique, le nettoyage de code obsolète et l'amélioration du dispositif de tests automatisés. Ces actions ont été réalisées de manière progressive et maîtrisée, dans le respect de la stabilité fonctionnelle de l'application.

Le renforcement et l'intégration des tests unitaires et d'intégration dans un *pipeline CI/CD* ont constitué un apport structurant du stage. Cette démarche contribue désormais à sécuriser les évolutions du projet et à inscrire la plateforme dans une dynamique d'amélioration continue. Par ailleurs, le travail de documentation et de formalisation technique a participé à renforcer la maintenabilité du projet et à faciliter sa compréhension par d'autres développeurs.

Dans l'ensemble, ce stage a permis d'apporter une contribution concrète au projet *Swantrad*, tant sur le plan technique qu'organisationnel, tout en constituant une expérience professionnelle cohérente avec les objectifs de la formation BTS SIO option SLAM.

Réponse à la problématique

La problématique principale de ce stage portait sur la capacité à *améliorer la qualité, la maintenabilité et la fiabilité d'une application web existante*, tout en garantissant la stabilité d'un projet déjà déployé en production et sans introduire de régressions fonctionnelles.

Les travaux réalisés ont permis d'apporter une *réponse progressive et encadrée* à cette problématique. La stabilisation de l'environnement de développement a constitué un prérequis essentiel, en offrant un cadre de travail fiable et reproductible. L'analyse de la qualité du code à l'aide de *SonarQube* a permis d'identifier les zones les plus sensibles du projet et de cibler les actions à mener en priorité, notamment en matière de *maintenabilité et de dette technique*.

Les opérations de maintenance, de *refactorisation et de nettoyage du code* ont contribué à améliorer la lisibilité, la structure et la compréhension de certaines parties critiques de l'application, tout en respectant strictement son comportement fonctionnel initial. Néanmoins, en raison de la taille du projet, de son historique et du fonctionnement des outils d'analyse automatisée, ces améliorations ne se traduisent pas nécessairement par une amélioration spectaculaire du *Quality Gate global*, mais elles

se reflètent clairement dans certaines métriques ciblées, notamment la diminution des Code Smells à forte sévérité et la réduction des duplications.

En revanche, l'amélioration et la fiabilisation du dispositif de tests automatisés, ainsi que leur intégration dans un *pipeline CI/CD*, apportent une réponse concrète et durable à la question de la fiabilité. Ces outils permettent désormais de sécuriser les évolutions futures, de détecter plus rapidement les anomalies et de mieux maîtriser la qualité du code dans le temps.

Ainsi, même si les indicateurs de qualité n'ont pas connu une amélioration spectaculaire sur la période du stage, la problématique a été traitée de manière progressive, structurée et professionnelle. Le projet ***Swantrad*** dispose désormais d'un cadre technique plus structuré, mieux outillé et plus sécurisé, permettant d'envisager une amélioration continue de la qualité sans compromettre la stabilité de l'application en production.

Remerciements

Je tiens à remercier l'ensemble de l'équipe de **Swantrad** pour l'accueil, la confiance et le cadre de travail professionnel qui m'ont été offerts tout au long de ce stage.

Je remercie tout particulièrement mon tuteur de stage, **Monsieur TERENCE FERUT**, pour son accompagnement, sa disponibilité et les conseils apportés, qui m'ont permis de progresser tant sur le plan technique que méthodologique. Son encadrement a constitué un élément essentiel à la réussite des missions qui m'ont été confiées et à la bonne compréhension des enjeux du projet.

Je souhaite également remercier **Monsieur Alexandre KAABI**, président de la startup **Swantrad**, pour m'avoir accordé sa confiance et pour l'opportunité qui m'a été donnée d'effectuer ce stage au sein de l'entreprise.

Enfin, je remercie mon établissement de formation ainsi que les enseignants pour l'accompagnement pédagogique fourni, qui a permis de donner un cadre structurant à cette expérience professionnelle.

Annexes

Annexe 1 : Arborescence du dossier d'annexes

L'ensemble des livrables techniques réalisés durant le stage a été regroupé dans un dossier d'annexes intitulé **ANNEXES_STAGE_SWANTRAD**.

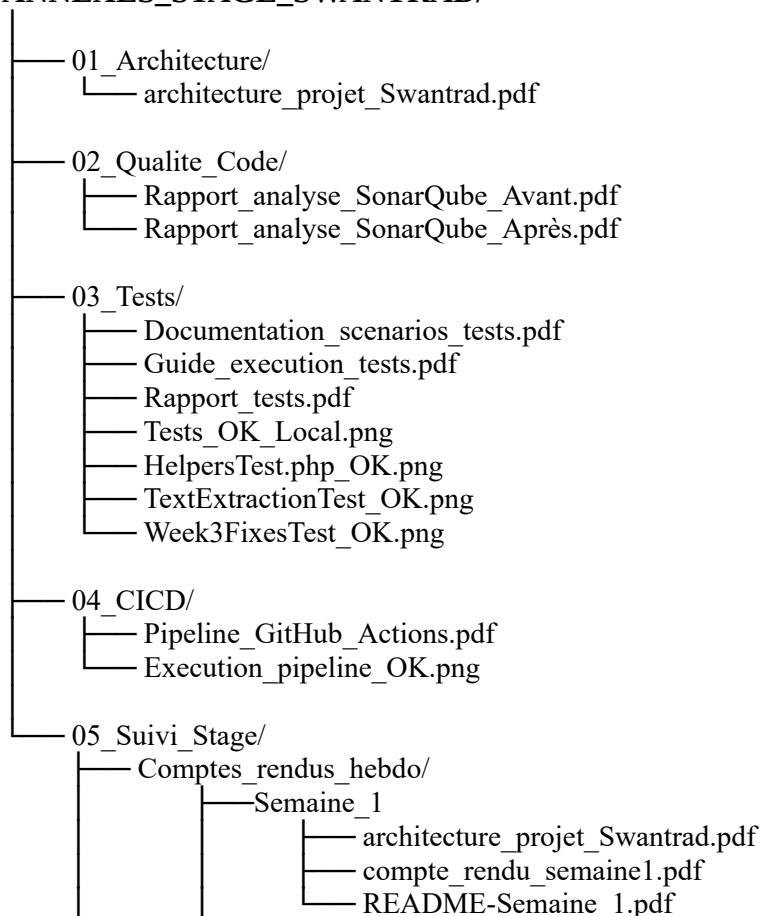
Afin de préserver la lisibilité du rapport principal, seuls les éléments les plus représentatifs ont été intégrés directement dans le corps du document.

Le dossier d'annexes contient notamment :

- les documents d'architecture du projet,
- les rapports d'analyse de la qualité du code (SonarQube),
- la documentation et les preuves liées aux tests automatisés,
- les éléments relatifs à l'intégration continue (CI/CD),
- l'ensemble des livrables et comptes rendus hebdomadaires du stage.

Ce dossier est fourni en complément du rapport sous la forme d'une archive, permettant au jury d'accéder à l'intégralité des travaux réalisés.

ANNEXES_STAGE_SWANTRAD/



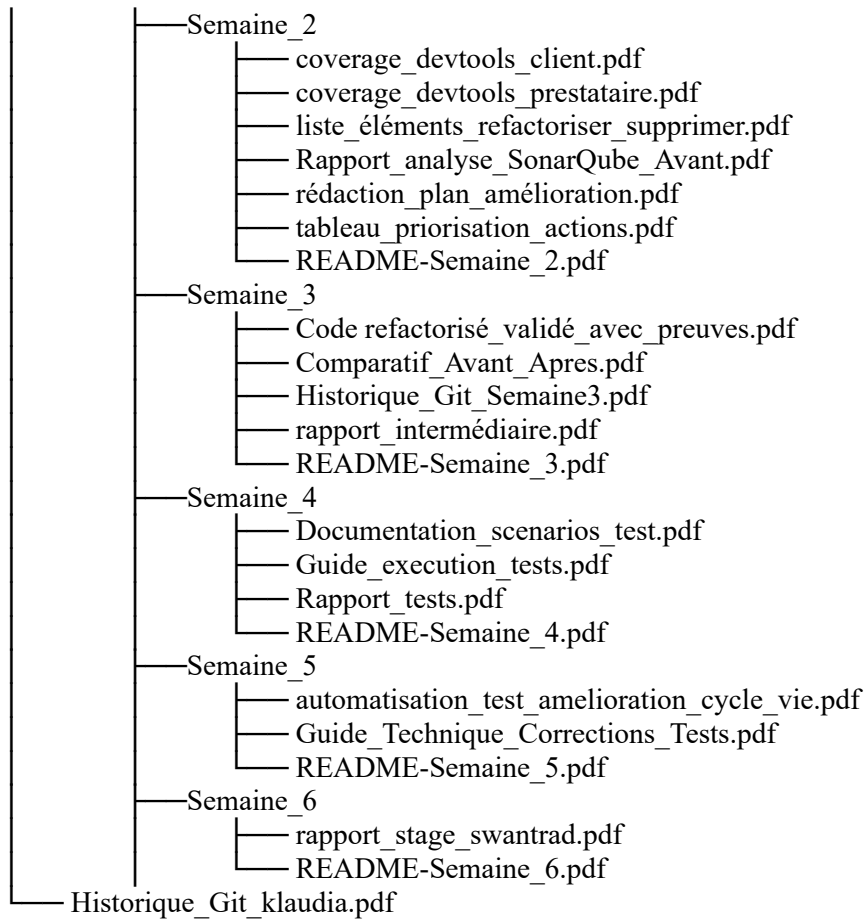


Figure 2: Analyse SonarQube – État initial du projet (avant le stage)

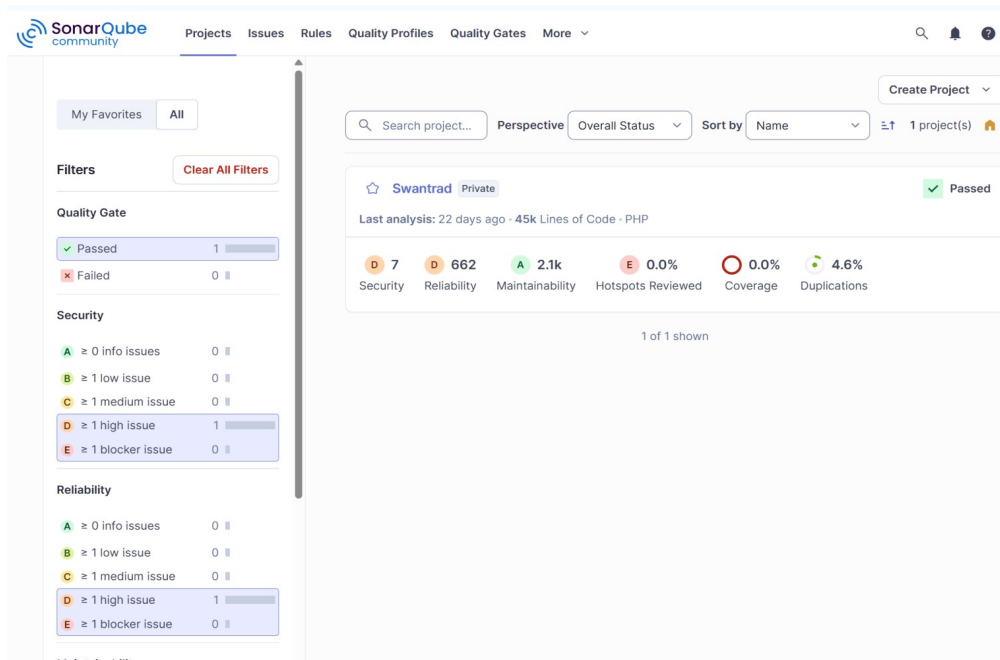


Figure 3: Analyse SonarQube – État du projet en fin de stage

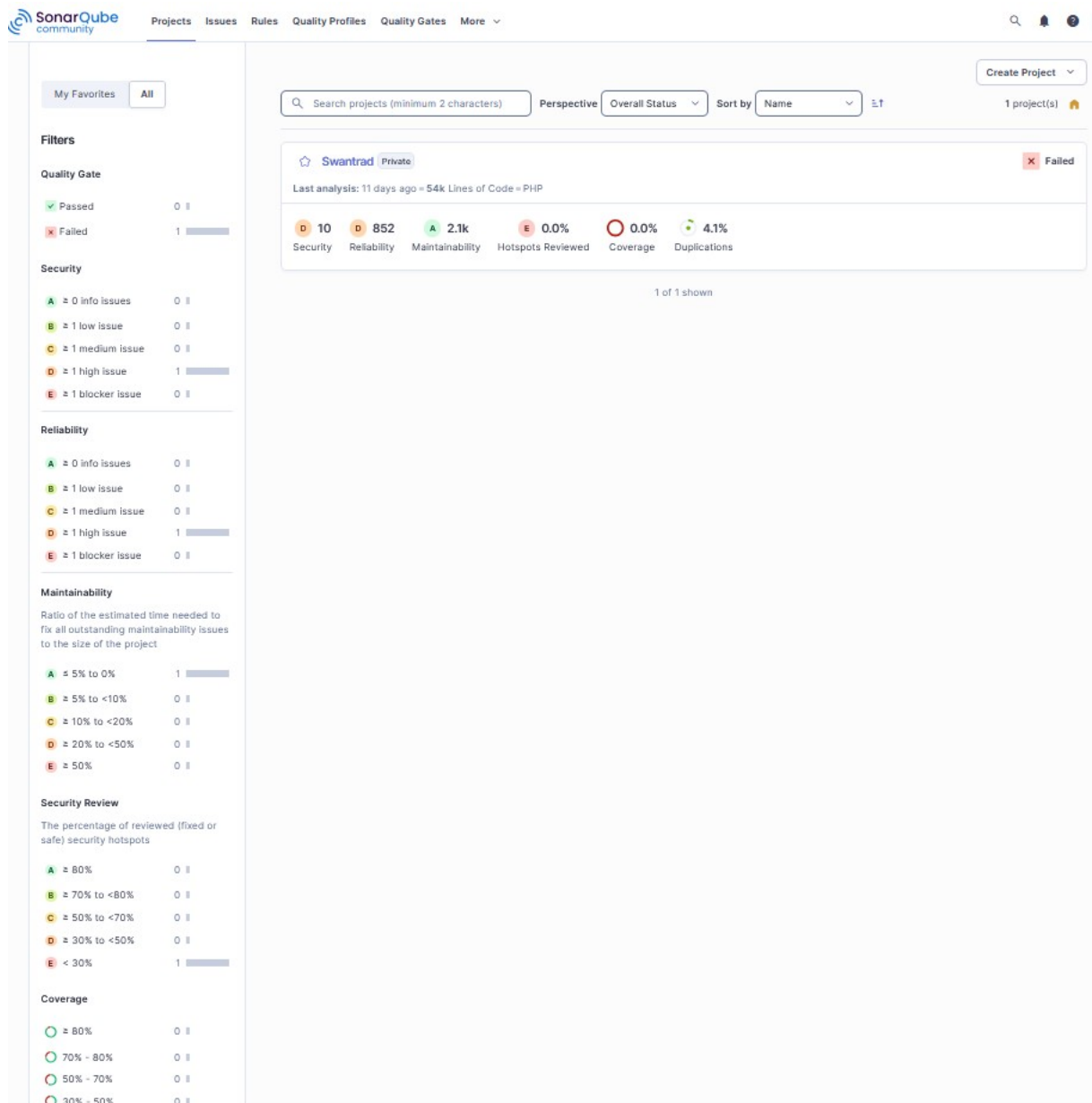


Figure 4: Résultat de l'exécution des tests en local (« php artisan test »)

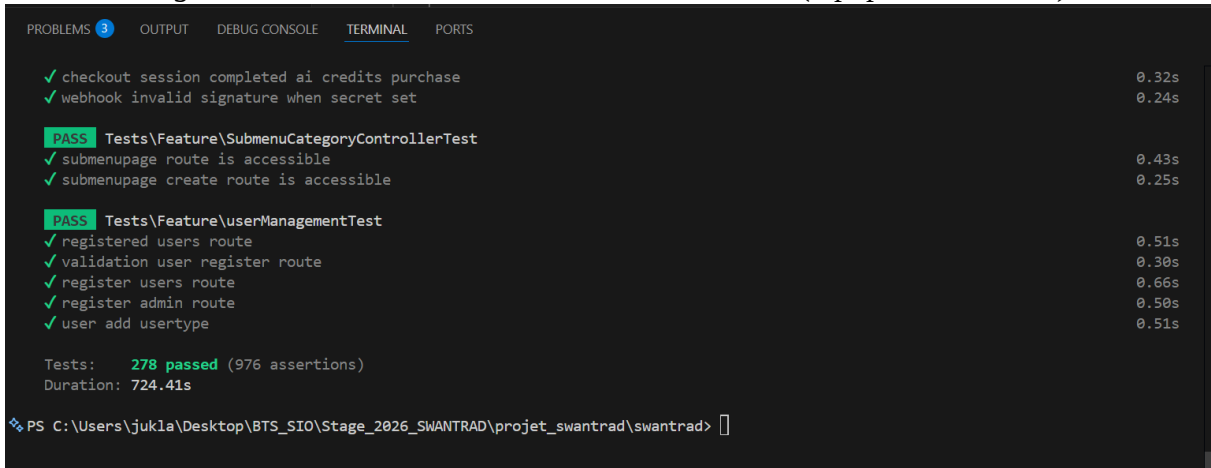
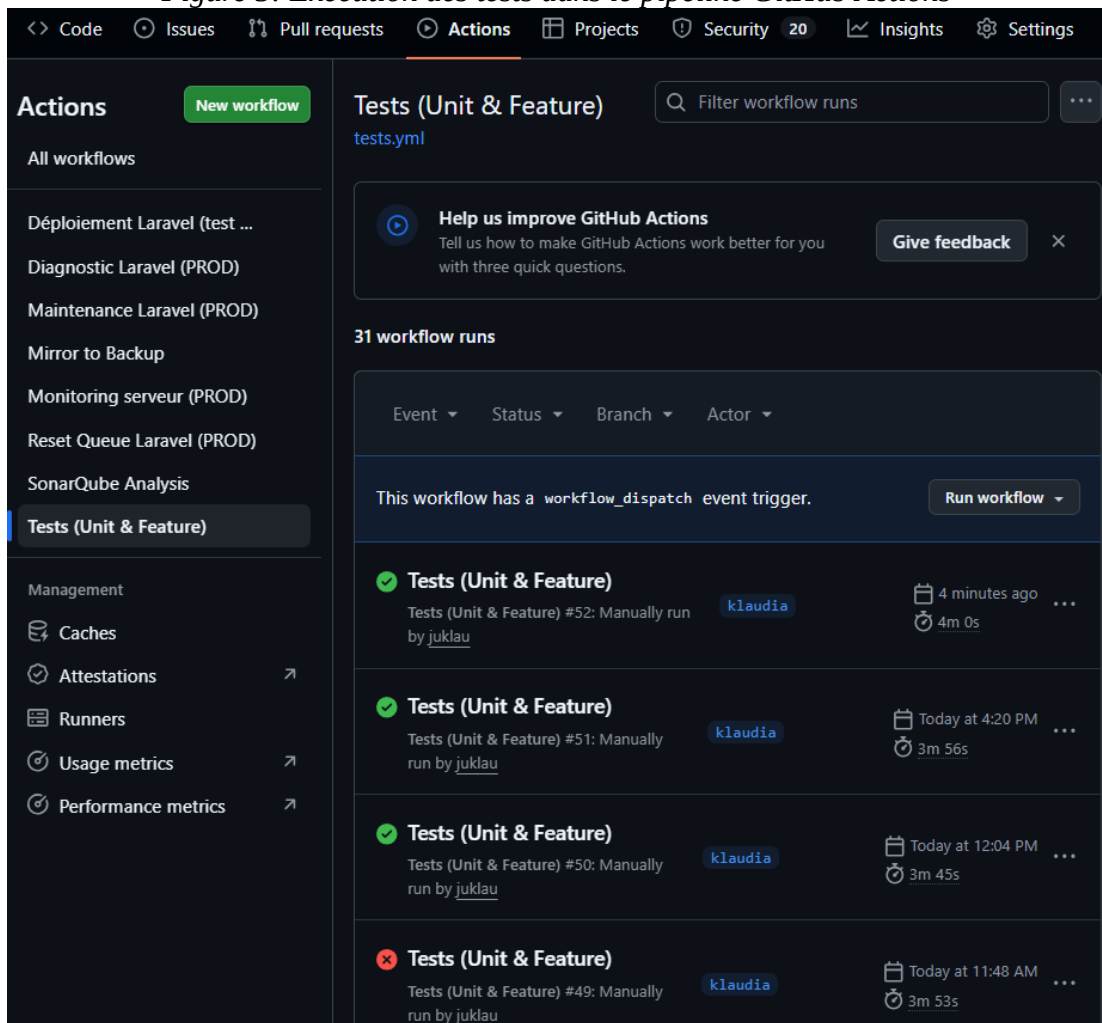


Figure 5: Exécution des tests dans le pipeline GitHub Actions



Bibliographie

Swantrad. (2026). *Plateforme de digitalisation des traductions assermentées.*

Disponible sur : <https://www.swantrad.com>

Laravel. (2026). *Laravel Documentation.*

Disponible sur : <https://laravel.com/docs>

PHPUnit. (2026). *PHPUnit Manual.*

Disponible sur : <https://phpunit.de/documentation.html>

GitHub. (2026). *GitHub Actions Documentation (anglais).*

Disponible sur : <https://docs.github.com/en/actions>

GitHub. (2026). *Documentation GitHub Actions (français).*

Disponible sur : <https://docs.github.com/en/actions>

SonarSource. (2026). *SonarQube Documentation.*

Disponible sur : <https://docs.sonarsource.com/sonarqube/>

PHP. (2026). *PHP Manual.*

Disponible sur : <https://www.php.net/manual/fr/>

MySQL. (2026). *MySQL 8.0 Reference Manual.*

Disponible sur : <https://dev.mysql.com/doc/>