

SAOUD David

Concepteur, Développeur
d'Applications

Lycée général et technologique Alain René Lesage
20 rue Winston Churchill
56000 Vannes

Année 2020

DOSSIER DE SOUTENANCE POUR LE TITRE :

**CONCEPTEUR, DÉVELOPPEUR
D'APPLICATIONS**

Table des matières

Remerciements.....	5
Introduction.....	6
Résumé en Français.....	7
Résumé en Anglais.....	7
Liste des compétences du référentiel couvertes par les projets.....	8
Première partie : Rapport de stage.....	9
I. Présentation de l'entreprise.....	10
1.1 Iole Solutions en quelques mots.....	10
1.2 Le produit développé par Iole Solutions.....	11
1.3 Analyse du besoin.....	12
II. Cahier des charges.....	12
III. Gestion du projet.....	14
3.1 Planning et suivi.....	14
3.1.1 Réunion initiale : attribution de l'objectif.....	14
3.1.2 Intégration au sein de l'équipe et suivi.....	14
3.2 Environnement humain.....	15
3.2.1 Équipe de développement et organisation.....	15
3.3 Objectifs de qualité.....	16
IV. Spécifications fonctionnelles du projet.....	16
V. Spécifications techniques du projet.....	17
5.1 Outils et technologies utilisées.....	17
5.2 Contraintes d'accès au code.....	17
5.3 Contraintes d'accessibilité.....	18
VI. Réalisations : conception et codage.....	18
6.1 Revue de code.....	18
6.2 Identification des problèmes : pré-tests.....	19
6.3 Veille technologique et recherches.....	21
6.4 Comparatif (benchmark).....	22
6.5 Solution proposée.....	23
6.6 Présentation de la nouvelle méthode aux dirigeants.....	25
6.7 Solution mise en œuvre et évolution.....	26
VII. Synthèse : Difficultés et satisfactions rencontrées.....	27
Seconde partie : Projet personnel.....	29
I. Présentation du projet.....	30
1.1 Analyse du besoin.....	30
1.2 Objectifs.....	30
II. Cahier des charges.....	30
III. Gestion du projet.....	32
3.1 Planning et suivi.....	32
3.2 Environnement technique : éco-conception du serveur.....	32
3.3 Objectifs de qualité.....	34
IV. Spécifications fonctionnelles du projet.....	34
4.1 Maquettage et interactions utilisateur.....	34
4.2 Diagramme de séquence.....	38
V. Spécifications techniques du projet.....	40
5.1 Choix technologiques.....	40
5.1.1 Framework Symfony.....	40
5.1.2 Versionnage.....	41

5.1.3 Analyse, conception et modélisation.....	41
5.1.4 Base de données.....	42
5.2 Domaine et hébergement.....	42
5.3 Accessibilité.....	43
5.4 Évolution.....	44
VI. Réalisations : conception et codage.....	44
6.1 Développement de composants d'accès aux données.....	44
6.2 Fonctionnalités les plus importantes.....	51
6.2.1 Module d'inscription.....	52
6.2.2 Module de connexion.....	54
6.2.3 Gestion profile utilisateur.....	54
6.2.4 Module d'administration.....	54
VII. Présentation du jeu d'essai.....	57
7.1 Données en entrée.....	57
7.2 Données attendues.....	58
7.3 Résultat obtenu.....	59
VIII. Veille sur les vulnérabilités de sécurité.....	60
8.1 Mise en place d'audits de détection de failles de sécurité.....	60
IX. Situation de travail ayant nécessité une recherche.....	62
Conclusion.....	63
Annexes.....	64

Index des figures

Figure 1 - Situation géographique de Iole Solutions.....	10
Figure 2 - Document spécifiant la mise en page attendue pour l'édition d'un document PDF.....	12
Figure 3 - Document de type facture à reproduire.....	13
Figure 4 - Organigramme Iole Solutions.....	15
Figure 5 - Rendu PDF obtenu après génération avec la librairie WkHtmlToPdf et TWIG.....	16
Figure 6 - Exemple de code présent à mon arrivée dans l'entreprise : méthode de classe dédiée à la construction de l'en-tête de documents.....	20
Figure 7 - Tableau comparatif de deux librairies.....	22
Figure 8 - Récupération, décompactage et installation.....	23
Figure 9 - Simple test pour vérifier la bonne installation de l'outil.....	23
Figure 10 - Données entrantes envoyées dans une vue Twig.....	27
Figure 11 - Carte mentale représentant la plupart des technologies utilisées et mises en place.....	31
Figure 12 - Diagramme de Gantt, représentant les étapes cruciales pour la création de ce projet.....	33
Figure 13 - Page d'accueil du site.....	35
Figure 13 - Page d'inscription.....	36
Figure 14 - Page de connexion.....	36
Figure 15 - Page de profile utilisateur.....	37
Figure 16 - Page du tableau de bord d'administration.....	37
Figure 17 - Page d'édition ou de création d'un article ou d'un projet.....	37
Figure 18 - Diagramme de séquence d'inscription.....	39
Figure 19 - Diagramme d'Entités Relationnelles (ERD).....	42
Figure 20 - Fichier « _mixins.scss » permettant l'adaptation des blocs HTML en fonction des tailles d'écrans.....	43
Figure 21 - Diagramme de classe des entités (tables).....	45
Figure 22 - Attributs et méthodes de classe d'entités.....	46
Figure 23 - Exemple de création d'entité à l'aide de l'assistant en ligne de commande maker.....	47

Figure 24 - Méthode de contrôleur récupérant tout les utilisateurs en base de données sous forme d'objets.....	48
Figure 25 - Logique mise en place pour la création d'un compte.....	53
Figure 26 - Logique mise en place pour la connexion.....	56
Figure 27 - Test E2E (end to end) de la fonctionnalité : inscription.....	58
Figure 28 - Ligne de commande pour lancer un test E2E via l'utilisation de la bibliothèque PANTHER.....	58
Figure 29 - Résultat du test E2E obtenu.....	59
Figure 30 - Résultat des tests mis en place.....	59
Figure 31 - Tableau de bord du logiciel ZED après test de pénétration.....	61
Figure 32 - Demande de tirage (pull request) effectuée par DependaBot sur une dépendance contenant une faille de sécurité.....	62

Remerciements

Je tiens à remercier l'équipe pédagogique du Greta de Vannes et tout particulièrement M. Nicolas Le Chenic, mon enseignant tuteur, pour son implication et son suivi lors de ma formation.

Je tiens également à remercier M. Philippe Guy, gérant de l'entreprise : Iole Solutions, pour la liberté qu'il m'a accordée pour réaliser la tâche m'ayant été confiée.

Je remercie M. Aurélien Heliou, coordinateur de l'équipe et assistant chef de projet, pour les conseils prodigues et pour sa disponibilité afin de répondre à mes questions.

Enfin, je remercie l'ensemble des collaborateurs de Iole Solution, salariés, stagiaires et apprentis pour leur accueil et leur disponibilité au sein de l'entreprise.

Introduction

J'ai réalisé un rapport en deux parties, couvrant les points requis pour la certification du titre de Concepteur Développeur d'Applications.

Lors de mon stage, la mission m'ayant été confiée par l'entreprise ne me permettant pas de faire valoir l'ensemble des compétences nécessaires pour la certification, j'ai donc décidé sur mon temps libre de réaliser un projet personnel reprenant les points non abordés en stage.

Ainsi, la première partie porte sur le stage en entreprise effectué au sein d'Iole Solutions, à Vannes et sur un module permettant la génération de documents au format PDF.

La seconde partie, quant à elle, détaille la réalisation d'un portfolio sous la forme d'une application web, ayant pour objectif d'étayer mon C.V. lors d'entretiens d'embauche afin de faire valoir mes compétences. Cette réalisation couvre l'ensemble du référentiel de compétences requis pour le titre.

Résumé en français

Réalisation d'un projet en deux parties pour répondre aux objectifs de la certification de Concepteur Développeur d'Applications.

La première partie porte sur un stage en entreprise réalisé au sein de Iole Solutions, une start-up localisée à Vannes qui développe un ERP afin de faciliter la gestion comptable et financière de PME. L'objectif de ce stage était de mettre en place un module permettant la génération de document de type facture au format PDF. Après une revue du code existant, considéré comme trop complexe et associé à une librairie inadaptée, j'ai effectué une veille technologique et ai réalisé un comparatif des librairies. Pour la réalisation de ce module, j'ai utilisé la librairie WkHtmlToPdf et le moteur de template TWIG rendant le tout fonctionnel. Ma proposition d'utiliser TWIG et WkHtmlToPdf a été validée et encouragée par les dirigeants. Le module a ensuite été intégré à leur application.

La deuxième partie porte sur la réalisation d'une application web sous la forme d'un portfolio qui a pour objectif de faciliter ma recherche d'emploi. J'ai utilisé le framework symfony 5, une base de données PostgreSQL, la librairie wkhtmltopdf et ai installé et configuré un serveur web sur un raspberry pi. J'ai également mis en place des mesures de sécurité préconisées par l'OWASP et j'ai eu comme ligne directrice l'éco-conception et l'approche mobile-first.

Ces deux projets réunis m'ont permis de façon complémentaire d'appréhender les différents points requis pour la certification.

Résumé en anglais

I chose to produce a report in two parts in order to meet the objectives of the title of Designer and Developer of Applications.

The first part concerns an internship i did in a company called Iole Solutions which is located in Vannes and is developing an Enterprise Resource Planning in order to facilitate the accounting and financial management of Small & Medium-Sized Companies. The objective of this internship was to set up a module allowing the generation of invoice type document in PDF format.

The second part concerns the creation of a web application in the form of a portfolio which aims to ease my job search. I used the symfony 5 framework, a PostgreSQL database, the wkhtmltopdf library and installed and configured a web server on a Raspberry PI. I also implemented the security measures recommended by the OWASP and had eco-design and a mobile-first approach as guidelines.

Together, these two projects allowed me to master the various points required for certification.

Liste des compétences du référentiel couvertes par les projets



RÉPUBLIQUE FRANÇAISE

Ministère chargé de l'emploi

ANNEXE : FICHE DE SUIVI DU PROJET EN ENTREPRISE

TITRE PROFESSIONNEL CONCEPTEUR(TRICE) DEVELOPPEUR(SE) D'APPLICATIONS

Document complété d'un commun accord entre le stagiaire, le responsable du stage en entreprise et le responsable du centre organisateur, à joindre aux résultats des évaluations passées en cours de formation.

Compétences Voir le détail dans le référentiel d'emploi, d'activités et de compétences du titre CDI	Cocher les compétences mises en œuvre lors du projet en entreprise (en totalité ou partiellement)
1. Maquetter une application	<input type="checkbox"/>
2. Développer une interface utilisateur de type desktop	<input type="checkbox"/>
3. Développer des composants d'accès aux données	<input type="checkbox"/>
4. Développer la partie front-end d'une interface utilisateur web	<input type="checkbox"/>
5. Développer la partie back-end d'une interface utilisateur web	<input type="checkbox"/>
6. Concevoir une base de données	<input type="checkbox"/>
7. Mettre en place une base de données	<input type="checkbox"/>
8. Développer des composants dans le langage d'une base de données	<input type="checkbox"/>
9. Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement	<input checked="" type="checkbox"/>
10. Concevoir une application	<input checked="" type="checkbox"/>
11. Développer des composants métier	<input checked="" type="checkbox"/>
12. Construire une application organisée en couches	<input type="checkbox"/>
13. Développer une application mobile	<input type="checkbox"/>
14. Préparer et exécuter les plans de tests d'une application	<input checked="" type="checkbox"/>
15. Préparer et exécuter le déploiement d'une application	<input type="checkbox"/>

Observations éventuelles :

Du stagiaire :

Du centre organisateur :

De l'entreprise :

Nom et prénom du stagiaire :

SAOUD DAVID

Signature

Entreprise :

iole Solutions

Nom du responsable du stage :

Aurélien Heliou

Signature et cachet :

Centre organisateur :

GRETAT BRETAGNE SUD

Nom du responsable du centre

organisateur

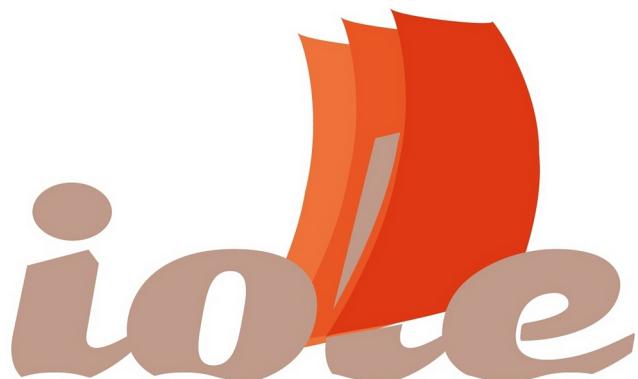
REGNIEZ Evelyne

Signature et cachet :

Première partie : Rapport de stage

« Module de génération de documents PDF »

Stage effectué au sein de Iole Solutions



Du 7 juillet 2020 au 28 août 2020

Maître de stage : M. HELIOU Aurélien

Entreprise : Iole Solutions, 12 rue Henri Becquerel, 56000 Vannes

Téléphone : 06.49.35.94.63

Enseignant tuteur Greta de Vannes : M. LE CHENIC Nicolas

I. Présentation de l'entreprise

1.1 Iole Solutions en quelques mots

Iole Solutions est une jeune start-up¹ vannetaise, éditrice de logiciels, créée en juillet 2018. Sa vocation est de développer une application spécifique à destination des PME (Petites et Moyennes Entreprises) et des ETI (Entreprise de Taille Intermédiaire).

L'idée, originelle de Iole Solutions est partie du constat de son fondateur M. Philippe Guy, à savoir : « *Il n'existe pas, à l'heure actuelle sur le marché, d'outil de gestion d'entreprise suffisamment performant pour être capable de répondre aux besoins réels des entreprises françaises ou étrangères* ». Il a donc créé la start-up Iole Solutions pour concevoir un outil informatique pertinent et évolutif visant à aider les dirigeants à tenir la barre de leur entreprise. L'objectif étant qu'il soit adapté quelque soit la taille et le secteur d'activité de la société, facilitant ainsi la gestion administrative et comptable mais aussi le service rendu au client.

Iole Solutions se situe au 12 rue Henri Becquerel à Vannes, dans le bâtiment « PIREN » (voir figure 1) du PIBS 2 (Parc Innovation Bretagne Sud), pépinière d'entreprises. Les locaux ont récemment été rénovés, offrant un espace de travail agréable. Celui-ci est constitué d'un plateau de travail de 12 postes, d'une salle de réunion et d'une pièce de détente. En ce qui concerne la forme juridique de Iole Solutions, il s'agit d'une Société par Actions Simplifiés à associé Unique (SASU). Elle a été créée et est gérée par M. Philippe Guy.

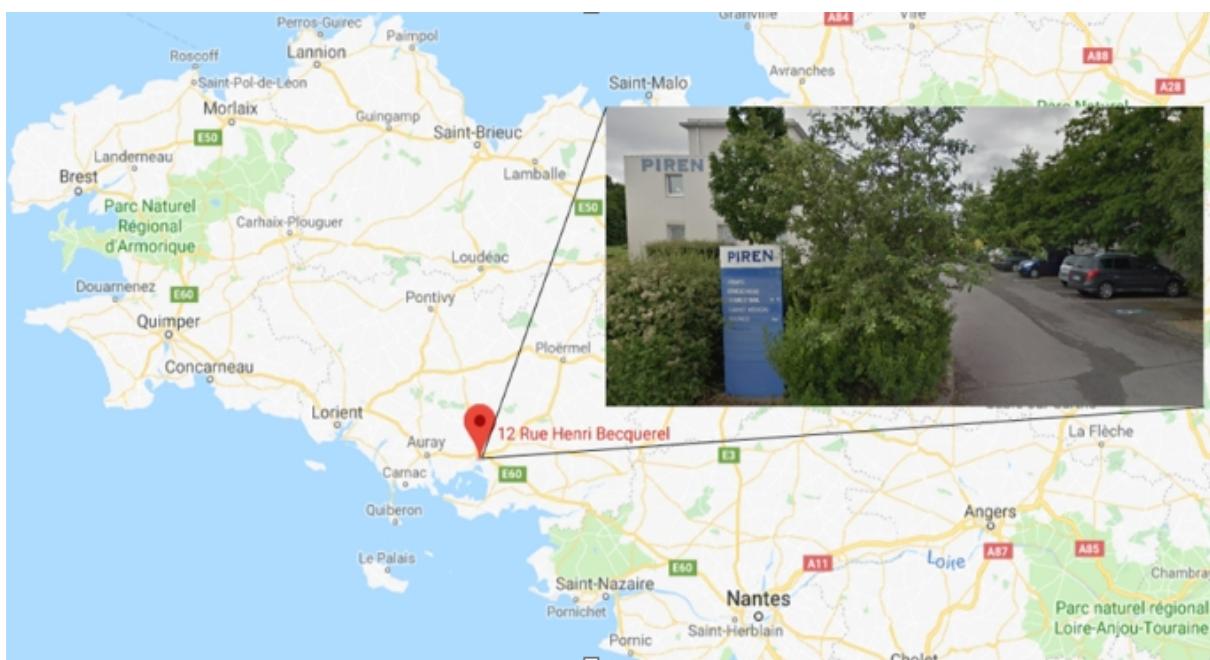


Figure 1 - Situation géographique de Iole Solutions.

1 - On désigne sous ce terme une entreprise nouvelle innovante à fort potentiel de croissance et de spéulation sur sa valeur future.

1.2 Le produit développé par Iole Solutions

Pour des raisons de confidentialité, l'infrastructure de l'application ne peut être que décrite dans ses grandes lignes. Nous avons d'ailleurs, au début du stage, signé un accord de non divulgation.

L'application de l'entreprise comprend différentes couches. Les niveaux les plus profonds constituent le moteur de l'application. Les niveaux les plus élevés sont destinés à l'affichage des menus, à l'affichage des requêtes faites par l'utilisateur et à l'utilisation générale de cet outil spécifique suivant le métier du client. Par ailleurs, deux bases de données lui sont associées, une base « technique » et une base « client ».

Par conséquent, le produit en cours de développement est une solution innovante, tant sur le plan de l'architecture logicielle que sur celui de l'expérience utilisateur. L'objectif de Iole Solutions est d'éditer cette application de gestion d'entreprise sous la forme d'un SAAS². Iole Solutions est en plein développement de leur propre Progiciel de Gestion Intégré (PGI en français, ERP : Enterprise Ressource Planning en anglais).

L'ERP d'Iole Solutions permet de construire des modules indépendants mais parfaitement compatibles sur une base de données unique et commune. Ce qui lui permet d'intégrer à leur application différentes fonctionnalités telles que :

- La gestion comptable et financière (ERP de gestion comptable, ERP de gestion de facturation),
- La gestion des stocks (logiciel ERP gestion des stocks),
- La gestion des ressources humaines,
- La gestion des fournisseurs (ERP fournisseurs grande distribution) ainsi que,
- La gestion de la vente,
- La gestion de la distribution ou encore,
- La gestion de l'e-commerce (ERP commerce, ERP de commerce détails spécialisé).

Autrement dit, l'ERP représente la « colonne vertébrale » d'une entreprise.

2 - Software As A Service ou logiciel en tant que service : modèle d'exploitation commerciale de logiciels dans lequel ceux-ci sont installés sur des serveurs distants plutôt que sur la machine d'un utilisateur, par exemple la plateforme Netflix, Dropbox ou encore les suites Office 365 et Adobe.

1.3 Analyse du besoin

Le site en cours de développement a pour objectif de faciliter la gestion administrative d'une entreprise. Une des fonctionnalités attendue et importante du site est la gestion comptable et financière, grâce à l'édition de documents PDF. Un module permettant cette fonctionnalité avait été mis en place par l'équipe précédente. Or, ce module n'était pas opérationnel. Il fallait donc le rendre fonctionnel.

II. Cahier des charges

Le cahier des charges m'ayant été fourni comportait deux documents : une expression de besoin pour générer un document PDF avec une structure spécifique (voir figure 2) et un modèle de facture à reproduire m'ayant été confiée par le designer de l'équipe (voir figure 3).

Expression du besoin :
PDF / Génération de document PDF

BESOINS FONCTIONNELS :

- En fonction du concept
 - o Courrier
 - o Facture (bon de livraison, devis, facture etc)
 - o Liste
 - o Etc
- En fonction d'un écran (recherche de champs, filtrage de champs, valeur etc)

ARCHITECTURE DU DOCUMENT PDF :

Un document c'est :

- Taille / orientation du doc / nombre de page (n pages)
- En tête de page
- En tête de doc |
- Bloc (n bloc possibles)
 - ⇒ En tête bloc / en tête répétés
 - ⇒ Lignes
 - Groupe_1
 - En tête Groupe_1 / en tête répétés
 - Groupe_1.2 (jusqu'à n groupe)
 - o En tête groupe_1.2 / en tête répétés
 - Ligne de détails groupe_1.2
 - o Pied de groupe_1.2 / pied répétés
 - o Ligne de détails groupe_1
 - Pied de groupe_1/ pied répétés
 - Bloc =>pied de bloc / pied répétés
 - Pied de doc
 - Pied de page

L'en-tête de Document n'est pas répété sur les pages suivantes, de même que le pied de document qui ne sera que sur la dernière page du document.

L'en-tête de page et le pied de page seront répétés sur chaque pages du document (avec option pour la 1^{ère} page).

Un bloc peut s'étaler sur plusieurs pages, son titre, sous totaux etc. pourront (option par total) être répétés sur chaque page ou le bloc sera présent.

Figure 2 - Document spécifiant la mise en page attendue pour l'édition d'un document PDF.



Bâtiment Piren
12, rue Henri Becquerel
56000 Vannes
Tél : +33 6 23 24 55 66

Détails facture :
N° de facture : FAC2020-01-025
Date d'émission : 06/01/20
Date d'échéance : 06/01/20
Référence client : GLA97432

Contact interlocuteur 1 :
Jean Vassal
Tél : +33 6 24 65 89 01
Mail : jvassal@iole.fr



Contact interlocuteur 2 :
Valentine Pontel
Tél : +33 6 24 65 89 01
Mail : v.pontel@iole.fr

LAUREEN NAUD
BÂTIMENT A, LOGEMENT 2
RÉSIDENCE LES COSTES
34630 SAINT THIBÉRY

Facture

Bon de commande : 459688M

Référence	Description - Du 23/04/19 au 03/05/19	Quantité	Prix unitaire	Code TVA	Remise	Montant
ZJG87	Objet avec ces caractéristiques offic temp HV2C..	2	10,00€	1	10%	20,00€
ZJ321	Objet avec ces caractéristiques offic temp HV2C..	2	5,00€	1		20,00€
ZJ23MQ	Objet avec ces caractéristiques offic temp HV2C..	2	300,00€	1	10%	20,00€
Sous-total						44,00€

Bon de commande : 456788F

Référence	Description - Du 23/04/19 au 03/05/19	Quantité	Prix unitaire	Code TVA	Remise	Montant
ZJG87	Objet avec ces caractéristiques offic temp HV2C..	2	10,00€	1	10%	20,00€
Sous-total						20,00€

Bon de commande : 459688M

Référence	Description - Du 23/04/19 au 03/05/19	Quantité	Prix unitaire	Code TVA	Remise	Montant
ZJG87	Objet avec ces caractéristiques offic temp HV2C..	2	10,00€	1	10%	20,00€
ZJ321	Objet avec ces caractéristiques offic temp HV2C..	2	5,00€	1		20,00€
ZJ23MQ	Objet avec ces caractéristiques offic temp HV2C..	2	300,00€	1	10%	20,00€
ZJ321	Objet avec ces caractéristiques offic temp HV2C..	2	5,00€	1		20,00€
ZJ23MQ	Objet avec ces caractéristiques offic temp HV2C..	2	300,00€	1	10%	20,00€
Sous-total						590,00€

Conditions et modalités de règlement :

Règlement par virement :

IBAN FR76 4590 4857 0301 9867 045 (Ajouter Libellé: FAC2020-IOLE)

Règlement par chèque à l'ordre de :

iole Solutions

Paiement sous 30 jours à réception de la facture. Pas d'escompte pour paiement anticipé. En cas de retard de paiement, application d'une indemnité forfaitaire pour frais de recouvrement de 40€ selon l'article D.441-5 du code du commerce.

Total HT	1309,54€	
Taux TVA	Base	Total TVA
TVA 20%	1004,00€	200,80€
TVA 10%	83,30€	8,33€
TVA 5,5%	127,46€	7,01€
Total TVA		216,54€
Total TTC		1430,90€

iole Solutions - SASU Société par actions simplifiée à associé unique - Capital social : 100 000,00 € - Siège social : 12, rue Henri Becquerel 56000 Vannes - RCS : Vannes B 842 110 835 - SIRET 842 110 835 00027 - Numéro de TVA intracommunautaire : FR3903050 - Code APE : 6201Z

Page 1/1

Figure 3 - Document de type facture à reproduire.

III. Gestion du projet

3.1 Planning et suivi

Au sein de l'entreprise, nous avions régulièrement des réunions avec l'équipe afin d'échanger sur nos avancements, demander des revues de code avant de fusionner nos changements sur la branche Git principale. La première semaine, j'ai essentiellement effectué de la lecture de code sur leur base de code existante, afin de mieux apprivoiser la structure MVC utilisée. Les semaines suivantes, j'ai identifié le problème responsable de l'inefficience du module de génération de PDF. J'ai ensuite réalisé une veille technologique avec comparatif. J'ai élaborer une solution, que j'ai présenté aux dirigeants lors d'une réunion, une fois celle-ci validée, j'ai développé la fonctionnalité.

3.1.1 Réunion initiale : attribution de l'objectif

Au cours de la réunion initiale, je me suis vu attribuer une mission spécifique qui permettrait aux utilisateurs de l'application de pouvoir générer des documents au format PDF à partir des vues de l'application. Ces vues étaient représentées sous formes de simples tableaux HTML sans aucun style appliqué. Ces tableaux pouvaient contenir des bons de livraison, des bons de commandes, des courriers type ou encore des factures et contenaient des données issues d'une base de données existante à laquelle je n'ai pas eu accès pour cause de confidentialité. Les documents une fois générés pouvaient alors être téléchargeable et imprimable directement.

Du code source existait, développé par plusieurs stagiaires et repris en main par les développeurs de l'entreprise, sans succès. Ce code, qui était destiné à la génération de documents PDF, ne fonctionnait que très partiellement.

3.1.2 Intégration au sein de l'équipe et suivi

Mon tuteur et responsable de stage était M. Heliou Aurélien, coordinateur projet et développeur PHP au sein de Iole Solutions.

De part le contexte sanitaire particulier, les semaines de travail dans l'entreprise étaient découpées comme suit dans le but d'éviter le contact avec les employés et autres stagiaires :

- 3 jours en télétravail,
- 2 jours en présentiel.

Nous n'utilisons que des outils et logiciels open source pour la communication, le partage de documents et le développement. Je suis intervenu dans le service Recherches et Développements niveau back-end en tant que développeur PHP pour développer les fonctionnalités nécessaires à l'avancement du projet. J'ai travaillé en collaboration avec M. Aurélien Heliou et M. Philippe Guy (Fondateur de la start-up et Architecte projet) en faisant régulièrement des réunions à distance et/ou en présentiel, afin de partager mon avancement et prendre en compte leurs commentaires, remarques et encouragements.

3.2 Environnement humain

3.2.1 Équipe de développement et organisation

Le chef de projet, M. Philippe Guy, assigne les différents chantiers à un ou plusieurs développeurs. Lors de la mise en place d'un nouveau chantier, le ou les développeurs concernés soumettent une structure de développement, constituée d'un cahier des charges et d'un descriptif de l'architecture. Une fois cette structure validée, le développement effectif peut commencer. Le chantier se termine par la présentation au chef de projet pour validation et par la rédaction de documentations prenant la forme d'un wiki, puis par la présentation au reste de l'équipe.



Figure 4 - Organigramme Iole Solutions.

Concernant l'organisation des tâches, un bilan sur ce qui est fait et ce qu'il reste à faire est réalisé chaque semaine (apparenté à une mêlée hebdomadaire de la méthode **Scrum**³). Cela permet de favoriser les échanges et l'entraide entre les membres de l'équipe. Enfin, les horaires de travail sont identiques pour tous : du lundi au vendredi de 8h30 à 12h15 et de 13h15 à 17h00.

3 - Scrum s'appuie sur le découpage d'un projet en « boîtes de temps », nommées *sprints* (« pointes de vitesse »). Les *sprints* peuvent durer entre quelques heures et un mois (avec un *sprint* médian à deux semaines). Chaque *sprint* commence par une estimation suivie d'une planification opérationnelle. Le *sprint* se termine par une démonstration de ce qui a été achevé.

3.3 Objectifs de qualité

Afin de s'assurer de la qualité du module que je devais développer, j'ai eu plusieurs réunions (une par semaine en général) avec M. Aurélien Heliou et M. Philippe Guy afin de faire valider mon travail et de soumettre des suggestions d'améliorations.

Le module terminé, c'est M. Aurélien Heliou qui est chargé d'effectuer une revue de code avant de fusionner la branche du module concerné sur la branche principale. La dernière semaine de mon stage, l'entreprise a décidé de mettre en place l'implémentation de tests unitaires et fonctionnels, ainsi qu'un logiciel d'audit de code (SonarQube), afin de mesurer la qualité du code, la dette technique générée, la duplication de code, la complexité du code et enfin la présence de codes smells⁴.

IV. Spécifications fonctionnelles du projet

The screenshot displays a PDF document with two pages of a bill. The first page contains the header 'Facture', the company logo 'idle', and contact information for Jean Vassal and Valentine Portet. It also includes a QR code and a reference number ZJ23MQ. The second page shows a table of items with quantities, unit prices, discounts, and totals, along with payment terms and a summary table at the bottom. Both pages have a footer with the company's address and a page number.

Référence	Description	Quantité	Prix Unitaire	Code TVA	Remise	Montant
ZJG87	Item 1	1	90,00 €	1	10 %	81,00 €
ZJG88	Item 2	5	40,00 €	1	70 %	60,00 €
ZJ788	Item 3	2	800,00 €	1	80 %	320,00 €

Sous-total: 461,00 €

Total HT	Taux TVA	Base	Total TVA
	TVA 20 %	1 004,00 €	200,80 €
	TVA 10 %	83,30 €	8,33 €
	TVA 5,5 %	127,46 €	7,01 €
	Total TVA		216,14 €
	Total TTC		1 599,14 €

Figure 5 - Rendu PDF obtenu après génération avec la librairie WkHtmlToPdf et TWIG.

Sur la figure 5, on peut observer le rendu d'une facture PDF, avec la méthode que j'ai mis au point. Comme spécifié dans l'expression des besoins (voir figure 2), l'on doit retrouver un en-tête ainsi qu'un pied de page sur chaque page, le pied de page prend en charge la pagination à l'aide de JavaScript. L'en-tête quand à lui possède un **QR code** unique à chaque type de document.

4 - En génie logiciel, les *code smells* ou *mauvaises odeurs* peuvent être de mauvaises pratiques de conception qui conduisent à l'apparition de défauts. Ces défauts sont souvent issus de mauvais choix d'implantation ou de conception <https://refactoring.guru/refactoring/smells>

V. Spécifications techniques du projet

5.1 Outils et technologies utilisés

L'environnement de développement (IDE) Visual Studio Code est utilisé par l'ensemble de l'équipe (back-end et front-end). Cet outil est configuré via un ensemble de plugins à paramétrier.

Aussi, un certain nombre d'extensions permettant d'améliorer les performances et la non-tracabilité des données, doivent être installées sur chaque poste. De plus, l'accès aux bases de données sous *PostgreSQL* se fait par l'intermédiaire de l'outil pgAdmin4 (plateforme open source de gestion de base de données relationnelles).



PHP

La partie Back-end de Iole Solutions est développée en PHP natif. J'ai donc utilisé ce langage de programmation pour le module qui m'a été confié.

Visual Studio Code

Nous avions un IDE (environnement de développement intégré) à notre disposition pour développer : Visual Studio Code.



GIT

Git était utilisé pour gérer le code source, afin de versionner le code et pouvoir conserver un historique. J'ai utilisé Gitea pour manipuler les dépôts Git.

PostgreSQL

Enfin, le SGBD (Système de Gestion de Base de Données) utilisé était PostgreSQL, pour gérer les différentes bases de données du projet. PgAdmin4 quant à lui proposait une interface graphique permettant de manipuler les tables plus facilement.



Pour finir, l'utilisation des applications *Google* est proscrite par l'entreprise pour garantir la confidentialité du projet. L'utilisation de logiciels Open Source est quant à elle encouragée.

5.2 Contraintes d'accès au code

Les postes de travail fonctionnent sous le système d'exploitation (ou OS) Windows 10, tout comme les sessions utilisateurs. Celles-ci sont hébergées sur un serveur distant ce qui permet de pouvoir se connecter à partir de n'importe quel poste via le protocole **RDP** (Remote Desktop Protocol). Ce système laisse la possibilité de moduler son voisinage en fonction des missions et des

besoins de collaboration. Les différents serveurs de tests virtualisés (*OS GNU/Linux Debian*) sont gérés à l'aide de Proxmox⁵.

5.3 Contraintes d'accessibilité

L'application d'Iole Solutions doit être compatible avec les navigateurs modernes actuels (notamment **Mozilla Firefox**, **Google Chrome**, **Safari** et **Opera**), mais supportera également une approche **responsive** afin de pouvoir accéder à l'application depuis divers périphériques tels que les smartphones, tablettes et ordinateurs portables.

VI. Réalisations : conception et codage

6.1 Revue de code

Ma première tâche a consisté à m'approprier la structure en place afin d'avoir une vision d'ensemble du projet et de sa structure. Iole Solutions travaille sur un **framework** maison développé en PHP natif et sans l'aide de gestionnaire de dépendances (type Composer ou NPM).

Cette approche amène son lot d'avantages et d'inconvénients. Il me semble que l'avantage d'un framework maison sans dépendance est la simplicité de prise en main à court terme avec peu de code inutilisé au début. Cependant cet avantage peut se transformer en inconvénient à terme avec le double travail qu'exige la maintenance du framework, son impact sur la documentation de l'application et la nécessité d'implémenter des fonctionnalités existantes par ailleurs.

En inspectant le code, j'ai pu remarquer qu'une librairie permettant de convertir une simple page HTML en PDF était installée à la racine du projet. Cette librairie était **TCPDF**, un outil libre et open source permettant la génération de documents PDF. Celle-ci rend cela possible grâce à un jeu de classes PHP et prend en compte le support complet de documents utilisant UTF-8 comme encodage de caractères.

J'ai pu constater que le code ne respectait pas certaines bonnes pratiques comme le principe de responsabilité unique (Exemple avec la méthode « **invoiceHeader()** » sur la figure 6). À cet instant, il m'a semblé nécessaire de retravailler le code avant de me lancer dans l'ajout de fonctionnalités.

5 - Plateforme open source de gestion de serveurs virtualisés.

6.2 Identification des problèmes : pré-tests

Après m’être documenté sur la librairie en place (TCPDF), j’ai effectué des tests de génération de documents au format PDF à l’aide de la base de code existante.

```

/*
 * Wiki :
 * EN : Generates HTML header
 * FR : Génère le header en HTML
 * @param boolean $isGeneric
 * @return string
 */
public function invoiceHeader($isGeneric = false)
{
    $htmlHeader = '<!DOCTYPE html>
<head>
<meta charset="utf-8">
<link type="text/css" rel="stylesheet" href="css/pdf.css"/>
</head>
<body>
<div class="headFac"><div class="logoAndInfo">;';

$output = [];

$image = new barcode_generator();
$image->output_image('png', 'qr', 'www.google.fr', '');
$this->setBarcodeName($image->getName());

if ($isGeneric) {
    echo $isGeneric;
    $headerGeneric1 = [
        0 => 'Expédition non facturée',
        1 => 'Anjou Maine Céréales',
    ];
    $headerGeneric2 = [
        'Utilisateur' => 'Jean Vassal',
        'DateTime' => '23/02/20 à 13h56',
    ];
    array_push($output, $headerGeneric1);
    array_push($output, $headerGeneric2);
    return $output;
    $htmlHeader .= '<div class="firstLine"></div>
<div class="bloc"><div class="semibold">Recherche :</div>
    . $headerGeneric1[0] . '<br/>' .
    $headerGeneric1[1] . '</div>
<div class="secondLine"></div>
<div class="bloc"><div class="semibold">Utilisateur :</div>' .
    $headerGeneric2['Utilisateur'] . '<br/><br/><div class="semibold">Date et heure :</div>' .
    $headerGeneric2['DateTime'] . '</div>
<div class="thirdLine"></div>
<div class="barCodeHeader">
<img src=' . $this->barcodeName . ' class=barCode style=width:150px; height:150px/>
</div></div>';
} else {
    // echo 'good';
    $adresseIole = [
        'adresse1' => 'Bâtiment Piren',
        'adresse2' => '12, rue Henri Becquerel',
        'codePostal' => '56000',
        'ville' => 'Vannes',
        'tel' => '+33 6 23 24 55 66'
    ];
    $contactInterlocuteur1 = [
        'nom' => 'Jean Vassal',
        'tel' => '+33 6 24 65 89 01',
        'mail' => 'jvassal@iole.fr'
    ];
    $contactInterlocuteur2 = [
        'nom' => 'Valentine Pontel',
        'tel' => '+33 6 24 65 89 01',
        'mail' => 'v.pontel@iole.fr'
    ];
    $invoiceDetails = [
        'num' => 'FAC2020-01-025',
        'dateEmission' => '06/01/20',
        'dateEcheance' => '06/01/20',
        'referenceClient' => 'GLA97432'
    ];
    array_push($output, $adresseIole);
    array_push($output, $contactInterlocuteur1);
    array_push($output, $contactInterlocuteur2);
    array_push($output, $invoiceDetails);

    return $output;

    $htmlHeader .= '<br/> <br/>' .
        $adresseIole['adresse1'] . '<br/>' .
        $adresseIole['adresse2'] . '<br/>' .
        $adresseIole['codePostal'] .
        $adresseIole['ville'] . '<br/>' .
        $adresseIole['tel'] . '</div>
<div class="firstLine"></div>
<div class="facDetail">
<div class="headFacTitle bold"> Détails facture : </div>
N° de facture : ' . $facture['num'] . '<br/>
Date d'émission : ' . $facture['dateEmission'] . ' <br/>
Date d'échéance : ' . $facture['dateEcheance'] . '<br/>
Référence client : ' . $facture['referenceClient'] . '
</div>
<div class="secondLine"></div>

<div class="ioleDetail">
<div class="headFacTitle bold"> Contact interlocuteur 1 : </div>
    . $contactInterlocuteur1['nom'] . '<br/>
Tél : ' . $contactInterlocuteur1['tel'] . '<br/>
Mail : ' . $contactInterlocuteur1['mail'] . '
<div class="headFacTitle mt-2 bold"> Contact interlocuteur 2 : </div>
    . $contactInterlocuteur2['nom'] . '<br/>
Tél : ' . $contactInterlocuteur2['tel'] . '<br/>
Mail : ' . $contactInterlocuteur2['mail'] . '
</div>

<div class="thirdLine"></div>
<div class="barCodeHeader">
<img src=' . $this->barcodeName . ' class=barCode style=width:150px; height:150px/>
</div>
</div>';
}
return $htmlHeader;
}

```

Figure 6 - Exemple de code présent à mon arrivée dans l'entreprise : méthode de classe dédiée à la construction de l'en-tête de documents.

N'ayant pas pu travailler sur les bases de données de l'entreprise et ne pouvant pas décrire en profondeur la structure de l'application pour cause de confidentialité, j'ai dû adapter le code en utilisant des tableaux multi-dimensionnels reflétant les champs de bases de données qu'ils utilisaient. J'ai donc mis au point un programme (disponible à l'adresse suivante : https://github.com/Slashflex/tcpdf_psql_php) permettant la génération d'un document au format PDF à partir de données issues d'une base de données PostgreSQL. Le résultat obtenu était fonctionnel mais générait une dette technique considérable, une maintenabilité du code complexe et des problèmes de décalage de marge(s). De plus, TCPDF étant un outil peu maintenu, peu fiable et ayant été mis en place par plusieurs stagiaires sur ce même projet sans succès. En partant des constats suivants : librairie inadaptée et code complexe, j'ai recherché une solution alternative plus adaptée à leurs besoins. J'ai donc pris l'initiative de commencer ma mission sur une base neuve et saine en prenant en considération les spécifications techniques et l'architecture imposées par l'entreprise.

6.3 Veille technologique et recherches

La situation au cours du stage ayant nécessité une recherche poussée concernait donc en premier lieu, les librairies. Celle utilisée étant inadaptée.

Ainsi, j'ai effectué une veille technologique listant les librairies existantes permettant la génération de PDF, afin d'effectuer un comparatif. Au cours des recherches, j'ai pu identifier une autre librairie permettant la génération de PDF : **WkHtmlToPdf**, outil en ligne de commande open source pour le rendu PDF de documents HTML ou la capture de contenu d'un site internet via son URL au format PDF à l'aide du moteur de rendu Qt WebKit. Celui-ci fonctionne entièrement « sans tête » (sans clavier ni souris) et ne nécessite pas de service d'affichage.

Démarches de recherche de solutions concernant un problème technique :

- Mots clés de recherche utilisés : *alternative to tcpdf*
- Site retourné et choisi en résultat : <https://php.libhunt.com/tcpdf-alternatives>

Grâce aux résultats retournés, j'ai rapidement dirigé mes recherches sur la librairie WkHtmlToPdf évoquée dans ce site comparant les alternatives open source disponibles. J'ai donc effectué une nouvelle recherche avec comme mots clés : *github wkhtmltopdf*, j'ai ensuite pu lire la documentation du dépôt officiel de cette librairie. <https://github.com/wkhtmltopdf/wkhtmltopdf>.

En inspectant rapidement l'historique d'activité du dépôt, j'ai constaté que la maintenance était régulière et qu'un nombre important de contributeurs actifs collaboraient sur ce projet. Je me

suis donc rendu sur le site internet de cette librairie dans la rubrique documentation (<https://wkhtmltopdf.org/usage/wkhtmltopdf.txt>), afin de comprendre comment l'utiliser.

6.4 Comparatif (benchmark)

Afin de justifier et vérifier mon choix de librairie, j'ai trouvé un tableau comparatif (voir figure 7) de ces deux librairies. On peut, sur cette figure, observer que la librairie **WkHtmlToPdf** est maintenue à jour plus régulièrement. C'est aussi un outil plus fiable, plus performant et avec la possibilité d'obtenir un résultat très correct en peu de temps. Linux Debian étant installé sur le serveur principal d'Iole Solutions, après discussion avec les dirigeants et leur ayant démontré le gain en maintenabilité apporté par cette librairie par rapport à celle utilisée jusqu'à présent par leurs collaborateurs, je leur ai demandé l'autorisation d'installer la librairie **WkHtmlToPdf** sur ce serveur afin de pouvoir commencer à m'en servir. Ils ont été très réceptifs et m'ont encouragé en ce sens.

TCPDF	WKHTMLTOPDF
Repository	
2,641	★ Stars 10,074
185	👁 Watchers 397
1,015	🍴 Forks 1,376
-	🕒 Release Cycle 231 days
-	🕒 Latest Version 5 months ago
7 months ago	🕒 Last Commit about 2 months ago
More	
-	Code Quality L3
325,119	⬇ Monthly -
PHP	↔ Language C++
GNU Lesser General Public License v3.0 only	© License -
PDF, Wkhtmltopdf, Php-reports, TCPDF, Qrcode, PDFD32000-2008, Datamatrix, Pdf417, Barcodes	Tags PDF

Figure 7 - Tableau comparatif de deux librairies.

Installation de **WkHtmlToPdf** (voir Figure 8)

Je mets tout d'abord à jour l'index de la liste des packages du système, puis j'installe l'outil permettant de récupérer du contenu depuis des serveurs web, à l'aide de cet outil « **wget** » je récupère le contenu du dépôt **Github** contenant l'archive officielle **WkHtmlToPdf**. J'exécute ensuite la commande « **dpkg -i** » qui permet le dépaquetage et l'installation de dépôt(s), enfin j'utilise la dernière commande pour forcer l'installation de tout paquet(s) décompacté(s) mais pas encore installé(s).

```

sudo apt update
sudo apt -y install wget
wget https://github.com/wkhtmltopdf/wkhtmltopdf/releases/download/0.12.5/wkhtmltox_0.12.5-
1.stretch_amd64.deb
sudo dpkg -i wkhtmltox_0.12.5-1.stretch_amd64.deb
sudo apt -f install

```

Figure 8 - Récupération, décompactage et installation

Figure 9 - Simple test pour vérifier la bonne installation de l'outil

Une fois cette librairie installée, je peux directement m'en servir.

Cette commande a pour but de générer l'intégralité du contenu de la page internet google.com en format pdf.

La commande fonctionne comme suit :

- J'invoque la librairie **WkHtmToPdf**,
- Je choisis la cible (soit un lien internet, soit un fichier au format HTML),
- Je choisis la destination ainsi que le nom du fichier en sortie.

Une liste non exhaustive d'options est disponible via ce [lien](https://wkhtmltopdf.org/usage/wkhtmltopdf.txt) (<https://wkhtmltopdf.org/usage/wkhtmltopdf.txt>) ou via la commande de terminal « **wkhtmltopdf -H** ».

6.5 Solution proposée

Afin de mettre en évidence l'efficacité de cette solution et de la soumettre aux dirigeants de l'entreprise, j'ai décidé de développer un module sur une base de code saine et indépendante de leur code. Le code en place, structuré en Programmation Orientée Objet, était complexe à comprendre de part :

- l'absence de commentaires et de documentations,
- une indentation et un anglais technique tous deux approximatifs,
- la forte présence de balises HTML.

Par ailleurs, tout document HTML était directement construit depuis les méthodes de classes PHP via une concaténation de variable de boucle en boucle et de condition en condition. J'ai donc cherché un moyen d'éviter à tout prix à l'entreprise de devoir générer du code HTML à l'intérieur de méthodes PHP. Ceci dans le but futur d'avoir une maintenabilité accrue, de pouvoir

personnaliser des documents à convertir en PDF en fonction des futurs clients de l'entreprise (adaptés à leur(s) besoin(s)), mais surtout d'éviter les répétitions de code et donc de réduire la dette technique.

Lors d'une réunion avec M. Aurélien Heliou et M. Phillippe Guy, j'ai pu émettre l'idée d'utiliser un moteur de template (modèles ou patrons en français) pour la génération de documents, ce qui réduirait considérablement en taille et en poids le code existant mais apporterait également une lecture du code plus aisée et limpide. Le moteur de template en question est Twig⁶, un projet créé par les fondateurs du cadiciel PHP Symfony. En effet, il fonctionne en parfaite symbiose avec PHP et possède des fonctionnalités puissantes telles que des filtres, des fonctions, des opérateurs de comparaison mais aussi la possibilité d'utiliser des conditions ou encore des boucles. Twig permet d'encoder la logique spécifique au domaine en plus des fonctionnalités de bas niveau (DSL⁷ : Domain-Specific Language). Cette solution d'utiliser un moteur de template m'est apparue évidente pour ce projet car sur mon temps libre et à la même période j'ai pu utiliser le framework PHP Symfony 5 et utiliser le moteur de template Twig (voir seconde partie du rapport), qui permet le rendu de code PHP et donc facilite la création de structure(s) de document(s) HTML à l'aide de boucles par exemple évitant ainsi la duplication de blocs de code HTML.

Exemple : si une table de base de données possède « x » champs et que l'on souhaite afficher ces champs dans une balise de type «**article**» en HTML, on ne fera qu'une boucle «**for**» dans la vue HTML afin d'obtenir au tant de balises «**article**» qu'il existe de champs en base de données. On aura ainsi un unique bloc de code dynamique, au lieu d'une multitude de blocs de codes HTML identiques de part leur structures.

J'ai donc effectué des tests de mise en place entre ma base de code découpée de leur projet et Twig, afin de leur démontrer la pertinence d'utiliser un moteur de template. Il s'agissait d'envoyer dans un fichier (.html, .twig ou .html.twig) un tableau à plusieurs dimensions via une instanciation des classes principales de Twig permettant le rendu de template, notamment grâce à la méthode «**render()**», ce dans le but de pouvoir utiliser ce tableau imbriqué dans un simple fichier html ayant pour but de reproduire un modèle de facture m'ayant été confiée par le designer de l'équipe (voir figure 3).

Tout le traitement pour les totaux, les calculs de TVA, des quantités etc. est ainsi effectué dans la vue html à l'aide de fonctions, boucles et conditions accessibles grâce à la syntaxe de Twig, plutôt que d'avoir à effectuer ces traitements dans une une classe PHP et de devoir construire le

6 - Twig est à la fois convivial pour les concepteurs et les développeurs en respectant les principes de PHP et en ajoutant des fonctionnalités utiles pour la création de modèles de pages.

7 - Un langage spécifique à un domaine (DSL) est un langage informatique spécialisé dans un domaine d'application particulier.

document html à l'intérieur des méthodes de cette classe (ce qui était le cas lorsque ce module m'a été attribué, avec un fichier de plus de 3 000 lignes de codes uniquement dédiées à la génération de documents PDF).

Ayant par contre décelé dans leur application l'inexistence de dépendances, j'en ai alors déduit qu'ils n'utilisaient pas de gestionnaire de dépendances, je leur ai donc expliqué que l'installation de Twig requérait, au premier abord, l'utilisation du gestionnaire de dépendances Composer, ils m'ont alors fait part de leurs réticences quand à l'utilisation de dépendances sur leur projet, ceci dans le but d'éviter toute maintenance de mise à jour ou de dépréciation. J'ai alors cherché un moyen de pouvoir installer Twig indépendamment de Composer, dans le but de prendre en compte leurs remarques et attentes. Au cours de recherches sur Internet, j'ai pu trouver une [solution](https://stackoverflow.com/questions/43506769/how-to-install-twig-without-using-composer) (disponible à l'adresse : <https://stackoverflow.com/questions/43506769/how-to-install-twig-without-using-composer>) à l'aide des mots clés suivants : *install twig without composer*.

J'ai donc mis en application cette solution :

Sur une branche Git (« **feature/invoice** »), à la racine du projet, je créer un dossier que je nomme Twig puis, je récupère l'archive zip via la commande « **wget** » depuis le dépôt officiel de la librairie Twig,

```
wget https://github.com/twigphp/Twig/archive/2.x.zip
```

j'extrais ensuite le dossier « **src** » contenu dans l'archive, dans le dossier Twig créé précédemment, ce qui donne Twig/src. Enfin, je renomme le dossier « **src** » en « **Twig** » :

```
mv 2.x/Twig-2.x/src/ Twig/Twig
```

puis je supprime le dossier 2.x original (2.x pour version majeur 2, version mineure x)

```
rm 2.x
```

Twig est maintenant prêt à être utilisé et n'a pas nécessité Composer pour son installation, de plus c'est une version stable et fonctionnelle qui ne nécessitera ni maintenance ni mise à jour.

6.6 Présentation de la nouvelle méthode aux dirigeants

Au cours d'une nouvelle réunion, je leur ai donc expliqué et démontré que Twig pouvait être utiliser indépendamment de Composer, ils m'ont alors félicité et encouragé à poursuivre le développement du module en ce sens. J'ai alors pu faire valoir la faisabilité de ma solution via une preuve de concept (POC : Proof Of Concept en anglais), après leur avoir expliqué le principe d'installation et d'utilisation de Twig et WkHtmlToPdf.

J'ai alors pu leur soumettre mes résultats et faire valider ma solution par le chef de projet. Le code source ainsi développé est présenté en annexes ci-jointes.

cf. Annexe 1 : Fichier « **loadTwig.php** » permettant le rendu PDF d'informations contenues dans des tableaux à plusieurs dimensions. (version en ligne disponible sur mon dépôt GitHub <https://github.com/Slashflex/CDA-rapport/blob/master/annexes/Extrait de code Stage - Iole/loadTwig.php>).

cf. Annexe 2 : Fichier « **test.html.twig** », recevant et traitant les données reçues par le fichier « **loadTwig.php** » (version en ligne disponible sur mon dépôt GitHub <https://github.com/Slashflex/CDA-rapport/blob/master/annexes/Extrait de code Stage - Iole/PDF/templates/test.html.twig>).

6.7 Solution mise en œuvre et évolution

J'ai réalisé un document technique reprenant les points requis pour la continuité de ce module, pour sa maintenance et sa maintenabilité. Ce document est basé sur l'implémentation que j'ai mis en place pour mener à bien cette mission. J'ai ainsi pu, la veille de mon départ, former un collaborateur de l'entreprise sur la méthode que j'ai mise en place pour parvenir au résultat escompté dans le but d'intégrer ma solution sur leur application. (document non reproduit pour cause de confidentialité car l'architecture de l'application y figurait).

L'installation d'outils de tests ayant été mise en place par un autre développeur la dernière semaine de mon stage, je n'ai malheureusement pas pu effectuer de jeux d'essai sur les fonctionnalités attendues par l'entreprise. Pour rappel, par soucis de confidentialité, je n'ai pas pu utiliser ou même avoir accès à la base de données de l'entreprise, donc les données que j'ai utilisées sont des tableaux clés => valeurs PHP à plusieurs dimensions (voir figure 10).

```

$invoice = [
    'BonDeCommande' => [
        0 => "459688M",
        1 => "45678BF",
        2 => "ZJ23MQ"
    ],
    "ColumnName" => [
        0 => "Référence",
        1 => "Description",
        2 => "Quantité",
        3 => "Prix Unitaire",
        4 => "Code TVA",
        5 => "Remise",
        6 => "Montant"
    ],
    "Value" => [
        [
            0 => "ZJG87",
            1 => "Item 1",
            2 => 1,
            3 => 90,
            4 => 1,
            5 => 10,
            6 => ""
        ],
        [
            0 => "ZJG88",
            1 => "Item 2",
            2 => 5,
            3 => 40,
            4 => 1,
            5 => 70,
            6 => ""
        ],
        [
            0 => "ZJ788",
            1 => "Item 3",
            2 => 2,
            3 => 800,
            4 => 1,
            5 => 80,
            6 => ""
        ]
    ]
];

$issuingCompany = [
    'adresse1' => 'Bâtiment Pirén',
    'adresse2' => '12, rue Henri Becquerel',
    'codePostal' => '56000',
    'ville' => 'Vannes',
    'tel' => '+33 6 23 24 55 66'
];

$customer = [
    'Nom' => 'LAUREEN',
    'Prénom' => 'NAUD',
    'Adresse1' => 'BÂTIMENT A, LOGEMENT 2',
    'Adresse2' => 'RÉSIDENCE LES COSTES',
    'CodePostal' => '34360',
    'Ville' => 'SAINT THIBERY',
];

```



```

$contact1 = [
    'nom' => 'Jean Vassal',
    'tel' => '+33 6 24 65 89 01',
    'mail' => 'jvassal@iole.fr'
];
$contact2 = [
    'nom' => 'Valentine Pontel',
    'tel' => '+33 6 24 65 89 01',
    'mail' => 'v.pontel@iole.fr'
];
$invoiceDetails = [
    'num' => 'FAC2020-01-025',
    'dateEmission' => '06/01/20',
    'dateEcheance' => '06/01/20',
    'referenceClient' => 'GLA97432'
];

```



```

$dataForTotal = [
    "ColumnName" => [
        0 => 'Taux TVA',
        1 => 'Base',
        2 => 'Total TVA'
    ],
    "Value" => [
        [
            0 => 20,
            1 => 1004,
            2 => ""
        ],
        [
            0 => 10,
            1 => 83.30,
            2 => ""
        ],
        [
            0 => 5.5,
            1 => 127.46,
            2 => ""
        ]
    ]
];

```

Figure 10 - Données entrantes envoyées dans une vue Twig.

VII. Synthèse : Difficultés et satisfactions rencontrées

La principale difficulté rencontrée a été le contexte sanitaire particulier dans lequel nous nous sommes trouvés. Pour autant, ce contexte, s'est finalement révélé intéressant sur un plan organisationnel car il a nécessité la mise en place de télétravail que j'ai trouvé bénéfique et source de satisfaction. En effet, je pense que dans le domaine du développement informatique, et ce stage me l'a démontré, l'utilisation de technologies et outils permettant le travail et la communication à distance est tout à fait possible sans pour autant freiner les capacités de chacun.

Ma seconde satisfaction est d'avoir réussi à faire prendre conscience à l'entreprise que le module en place avant mon arrivée générait une dette technique considérable et une lecture de code

complexe. J'ai donc pu, grâce à leur ouverture d'esprit et par ma force de proposition, mettre en avant mes connaissances et contribuer à fournir du code de qualité réglant une fonctionnalité très attendue par l'entreprise. J'ai éprouvé une grande satisfaction face à la liberté et à la confiance que les dirigeants m'ont accordées. Cette confiance s'est concrétisée par le fait d'avoir formé un de leur collaborateur pour intégrer ma solution de façon pérenne à leur application.

Seconde partie : Projet personnel

PORTFOLIO

/ FLX

≡



I. Présentation du projet

Le projet que j'ai choisi de réaliser est un portfolio sous la forme d'une application web. J'ai développé cette application à l'aide des technologies suivantes (voir figure 11).

1.1 Analyse du besoin

Le marché du travail étant de plus en plus exigeant et nécessitant de faire valoir ses compétences en quelques clics, j'ai songé que la réalisation d'un portfolio sous la forme d'une application web serait un atout lors d'entretiens d'embauche. Ce portfolio regroupe :

- une courte présentation personnelle,
- la possibilité de visionner ou télécharger mon CV,
- des projets réalisés au fur et à mesure de mon apprentissage,
- une partie blog où je pourrais publié mes documentations sur l'installation d'outils particuliers (en relation avec le développement web et informatique).

1.2 Objectifs

Réaliser un portfolio fonctionnel et aborder les points requis pour la certification que je n'ai pas pu voir au cours de mon stage.

II. Cahier des charges

Développer un site personnel regroupant les différents points énumérés dans l'analyse du besoin en prenant en compte :

- la sécurité de cette application,
- l'éco-conception,
- le mode **responsive** avec une approche **mobile first**,
- un coût de développement et de déploiement faible.

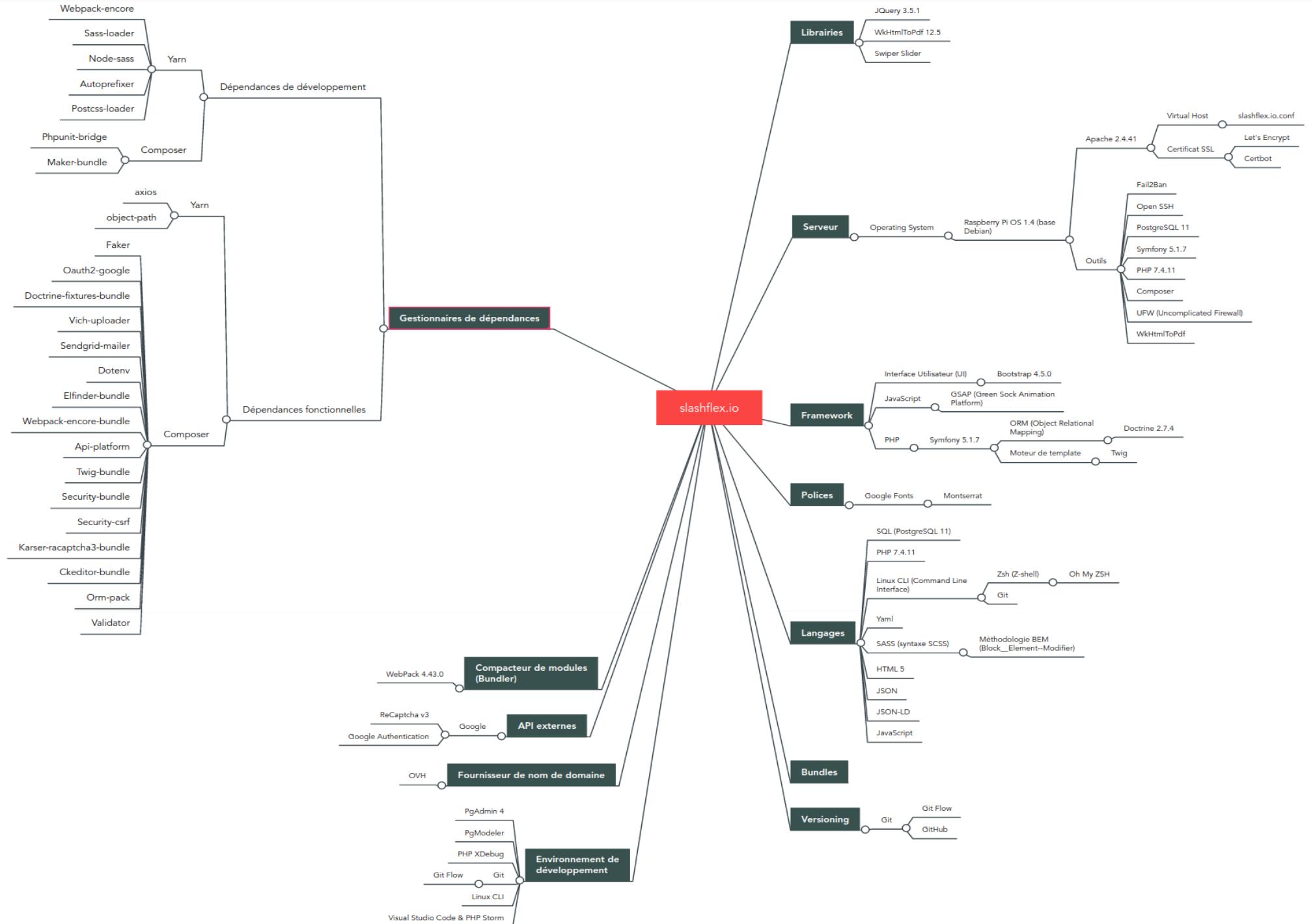


Figure 11 - Carte mentale représentant la plupart des technologies utilisées et mises en place.

III. Gestion du projet

3.1 Planning et suivi

Ce projet a été réalisé en autonomie complète sur mon temps libre à l'aide de recherches documentaires ciblées autour de points clés tels que la sécurité par exemple. Je m'étais fixé 3 mois et demi comme délai. J'ai débuté ce projet peu avant le début de mon stage (décrit en première partie de rapport) soit mi-juin. J'ai réussi à respecter le planning que j'avais initialement prévu, ceci grâce au diagramme de Gantt⁸ (voir figure 12) réalisé en amont du projet.

3.2 Environnement technique : éco-conception du serveur

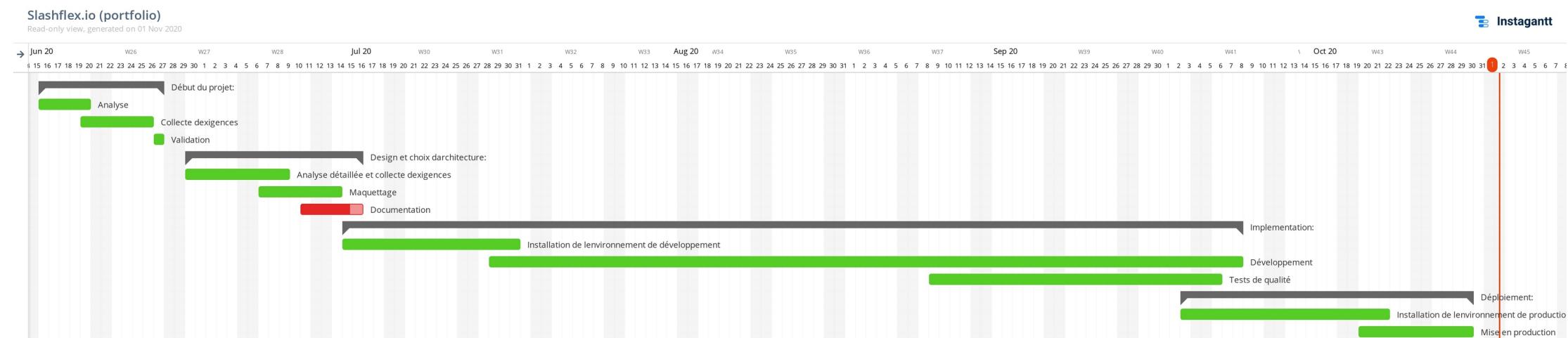
L'éco-conception étant, à mon sens, un point important à prendre en compte dans le cycle de développement d'une application, je souhaitais aussi mettre en production ce portfolio en ligne sur un Raspberry PI 4 que je possède à mon domicile et qui est configuré en tant que serveur web. En effet, ce type de périphérique est parfaitement adapté pour l'hébergement de sites internet (à faible trafic) ou pour être utilisé comme media center de part sa très faible consommation électrique. De plus, je ne voulais pas avoir à payer un hébergeur en ligne x€ par mois, quand je pouvais le faire gratuitement avec mes connaissances et sans devoir utiliser des serveurs déjà très énergivores. J'ai installé une distribution alternative de Debian (Raspberry PI OS), sur laquelle j'ai configuré Apache2 en environnement local, en environnement de production et de staging⁹.

J'ai également pris en charge un nouveau format pour les images sur internet : le format « **WEBP** », qui est un format d'image moderne offrant une compression supérieure sans perte pour les images sur le Web. En utilisant ce format, l'on peut créer des images plus petites et plus riches accélérant le Web. Pour ce faire, j'ai installé et utilisé un encoder développé par google (**cwebp** : <https://developers.google.com/speed/webp/docs/using>), qui est un outil en ligne de commande, permettant la conversion d'un fichier .png par exemple en fichier .webp, ce qui a pour effet de conserver la même qualité tout en réduisant en moyenne par 2 le poids de l'image.

8 - Un diagramme de Gantt, ou harmonogramme, est un type de diagramme à barres qui illustre un calendrier de projet. Ce graphique répertorie les tâches à effectuer sur l'axe vertical et les intervalles de temps sur l'axe horizontal. La largeur des barres horizontales dans le graphique indique la durée de chaque activité.

9 - Processus utilisé pour assembler, tester et examiner une nouvelle solution avant qu'elle ne soit mise en production.

Figure 12 - Diagramme de Gantt, représentant les étapes cruciales pour la création de ce projet.



3.3 Objectifs de qualité

Le but premier, qui m'a poussé à créer ce site, a été de pouvoir mettre en avant mes compétences en matière de développement auprès de recruteurs, donc pour trouver un emploi dans ce domaine. L'objectif, que je m'étais fixé, était de pouvoir déployer en production ce site et qu'il soit fonctionnel à 100 % avant la fin de ma formation soit le 27 novembre 2020.

Le second objectif fixé est d'obtenir un site efficient, rapide d'exécution et avec une approche « **mobile-first** » en ligne directrice. En effet, les smartphones étant de plus en plus utilisés pour la navigation sur internet, je trouve important de prendre en compte tout ces différents formats d'affichage dans le développement d'un site, mais surtout d'y penser dès le début de la phase de conception et maquettage.

Enfin, la sécurité sur Internet étant très importante, notamment pour protéger la vie privée des utilisateurs, j'ai mis l'accent sur la sécurité du projet tout au long du développement, dès la phase de conception avec une approche « **security by design** ». Les sites internet de l'Agence Nationale Française de Cybersécurité (ANSSI) et de la fondation OWASP (Open Web Application Security Project), décrivent très bien les failles de sécurité potentielles et comment s'en prémunir.

IV. Spécifications fonctionnelles du projet

4.1 Maquettage et interactions utilisateur

Ci-dessous, quelques maquettes représentant les parties les plus sollicitées par un utilisateur, avec une description des interactions possibles.

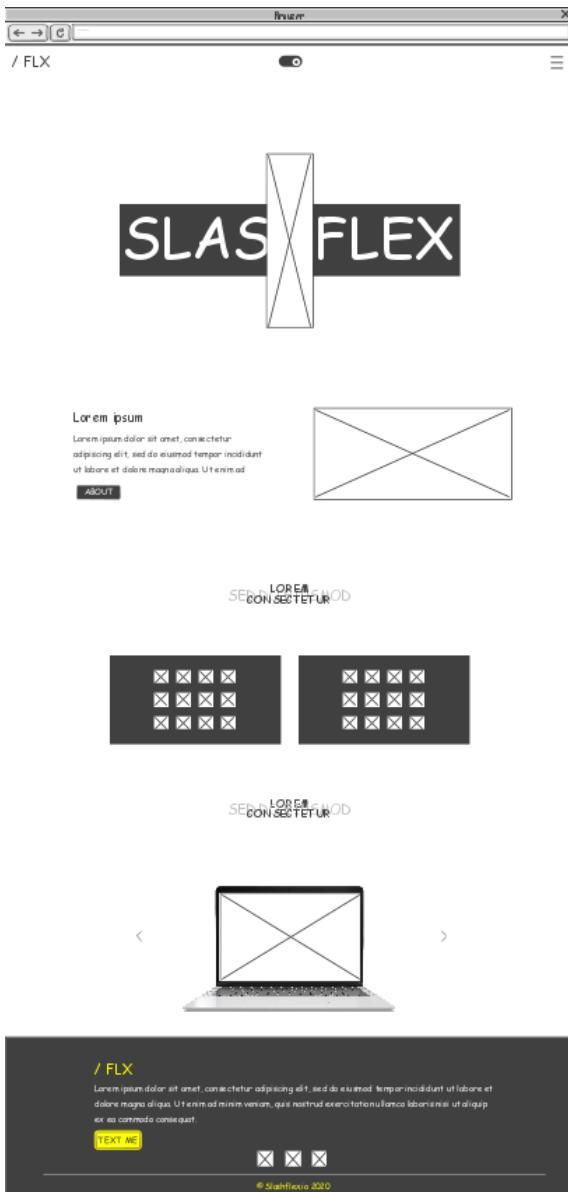


Figure 13 - Page d'accueil du site.

Dans la partie haute se trouvent 3 boutons :

- Le plus à gauche est utilisé pour revenir sur la page d'accueil depuis n'importe quelle autre page.
- Celui du milieu permet de basculer du thème actuel (blanc et noir), à un thème sombre.
- Enfin, le dernier sert à l'affichage du menu de navigation, redirigeant sur les autres pages du site.

Dans cette section, se trouve un court extrait me décrivant ainsi que mes compétences et un bouton redirigeant sur une page me présentant plus en détails.

Cette section dispose de deux blocs pivotant à 180° au clique. La partie visible affiche les outils et technologies que j'utilise au quotidien pour le développement côté front, la partie arrière visible au clique, elle, affiche les outils et technologies utilisées pour le développement côté back.

Cette section possède un carrousel d'images imbriqué dans une image d'ordinateur portable, l'on peut, à l'aide des flèches directionnelles ou au cliquer glisser faire défiler les images principales représentant divers projets développés en amont. Au clique sur un des projets, on est alors dirigé sur la page de celui-ci.

Enfin, le pied de page du site (présent sur chaque page), se dévoilant après avoir fait défiler la page jusqu'en bas. Il possède un bouton permettant de m'envoyer un e-mail directement, et 3 boutons de réseaux sociaux (mon profil GitHub, mon profil Linkedin et un envoi de mail). Il est présent sur chacune des pages

The screenshot shows a registration form titled "LOREM CONSECTETUR". The form consists of several input fields: "Email", "First name", "Last name", "Password", "Confirm password", and "Nickname". Below these fields is a text area containing placeholder text: "Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.". A radio button labeled "TEXT ME" is present. At the bottom of the form, there is a checkbox labeled "I accept the terms of service" followed by the text "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.". A "REGISTER" button is located at the bottom right.

Figure 13 - Page d'inscription.

La page en figure 13 est dédiée à l'inscription d'un utilisateur. Dessus, il renseignera ses informations personnelles, en veillant à respecter les validateurs de formulaires que j'ai mis en place pour gérer la sécurité de ces informations.

Le formulaire contient :

- Un champ e-mail,
- Un champ prénom,
- Un champ nom,
- Un champ mot de passe,
- Un champ confirmation de mot de passe,
- Un champ pseudonyme,
- Un bouton type radio invitant à accepter les termes d'utilisation du site.

Enfin, il contient un bouton permettant de soumettre ces informations au serveur et les persistées en base de données.

The screenshot shows a connection form titled "LOREM CONSECTETUR". The form includes fields for "Email" and "Password", and buttons for "LOGIN" and "GOOGLE". Below these fields is a text area containing placeholder text: "Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.". A "TEXT ME" button is present. At the bottom of the form, there is a text area containing placeholder text: "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.". A "TEXT ME" button is located at the bottom right.

Figure 14 - Page de connexion.

La page en figure 14 est dédiée à la connexion d'un utilisateur. Dans le formulaire, il renseignera son e-mail et son mot de passe utilisés lors de l'inscription. En cliquant sur le bouton de gauche, ses informations seront alors comparées avec les données en base de données et il sera alors connecté et redirigé sur la page de son profil. Il peut également cliquer sur le bouton de droite pour se connecter à l'aide de l'API¹⁰ de Google. Sur cette page, il peut également demander à réinitialiser son mot de passe ou demander que le système lui renvoie un e-mail de confirmation d'inscription.

10 - API : Interface de programmation : ensemble normalisé de classes, de méthodes, de fonctions et de constantes qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels.

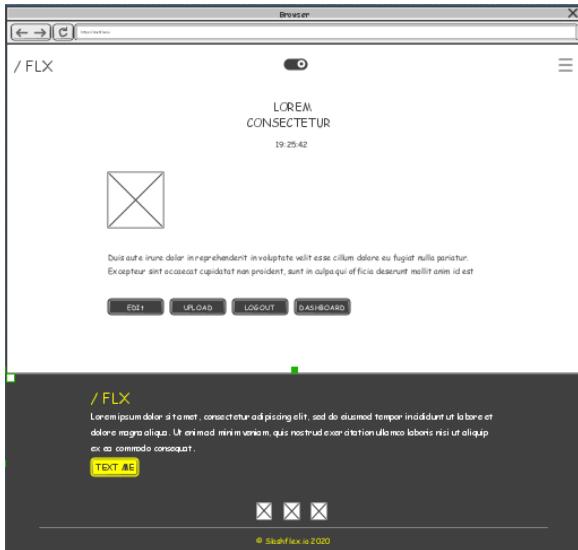


Figure 15 - Page de profil utilisateur.

En figure 15, la page de profil utilisateur, elle n'est accessible qu'une fois l'utilisateur connecté. Elle comprend l'heure actuelle, un avatar par défaut, un bouton lui permettant de mettre à jour ses informations (nom, prénom, mot de passe, e-mail, pseudonyme et description), un bouton lui permettant de téléverser depuis son ordinateur, une image faisant office d'avatar et un bouton pour se déconnecter. Le 4^e bouton est lui destiné à n'être afficher que sur la page de profil de l'administrateur du site, et redirige sur le panneau d'administration sécurisé de l'application.

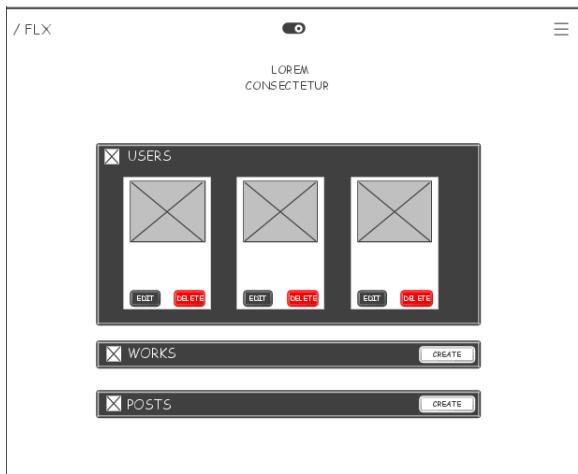


Figure 16 - Page du tableau de bord d'administration.

En figure 16, le tableau de bord d'administration de l'application, cette page n'est accessible qu'à l'administrateur du site, elle permet l'édition, la suppression et l'affichage des utilisateurs, des projets et des articles du blog.

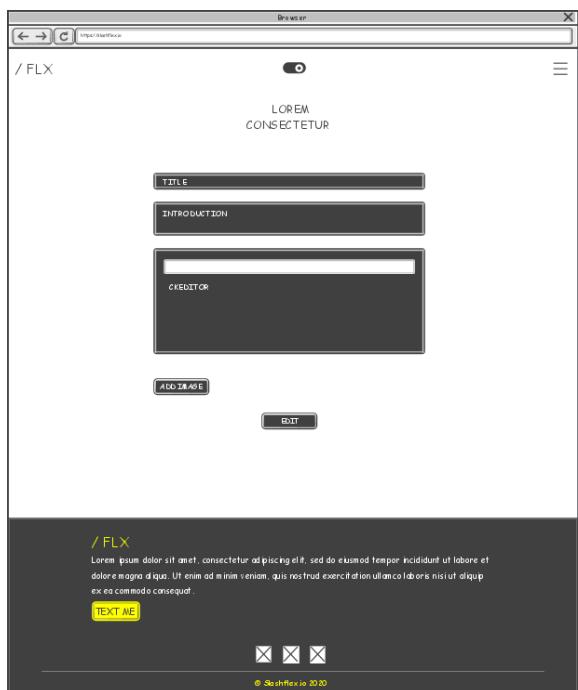


Figure 17 - Page d'édition ou de création d'un article ou d'un projet.

En figure 17, la page dédiée à la mise à jour ou la création d'un article de blog ou d'un projet, elle n'est accessible que pour l'administrateur du site, et possède un champ titre, un champ introduction, un champ contenu (où l'on peut insérer du code html ou markdown et le mettre en page directement à l'aide de CkEditor¹¹), mais aussi un champ de type fichier où l'administrateur peut téléverser une ou plusieurs images. La première image (de mise en avant) sera alors visible sur la page d'accueil du site dans le carrousel, les images téléchargées supplémentaires, seront affichées quant à elles sur leur page respective.

11 - Éditeur WYSIWYG permettant de modifier le contenu sous une forme qui ressemble à son apparence lors de l'impression ou de l'affichage en tant que produit fini tel qu'un document imprimé ou une page Web.

4.2 Diagramme de séquence

Le diagramme en figure 18 permet de montrer les interactions d'objets entre un utilisateur souhaitant créer un compte et le système. Dans un soucis de simplification, je n'ai pas traité les cas d'erreurs de validateurs de formulaire, mais ces erreurs sont bien gérées dans l'application et affichées au-dessus de chaque champ de formulaire impliqué. La dimension verticale du diagramme représente le temps, permettant de visualiser l'enchaînement des actions dans le temps, et de spécifier la durée de vie d'objets. Les périodes d'activité des objets sont symbolisées par des rectangles, et ces objets dialoguent à l'aide de messages.

Diagramme de séquence d'inscription

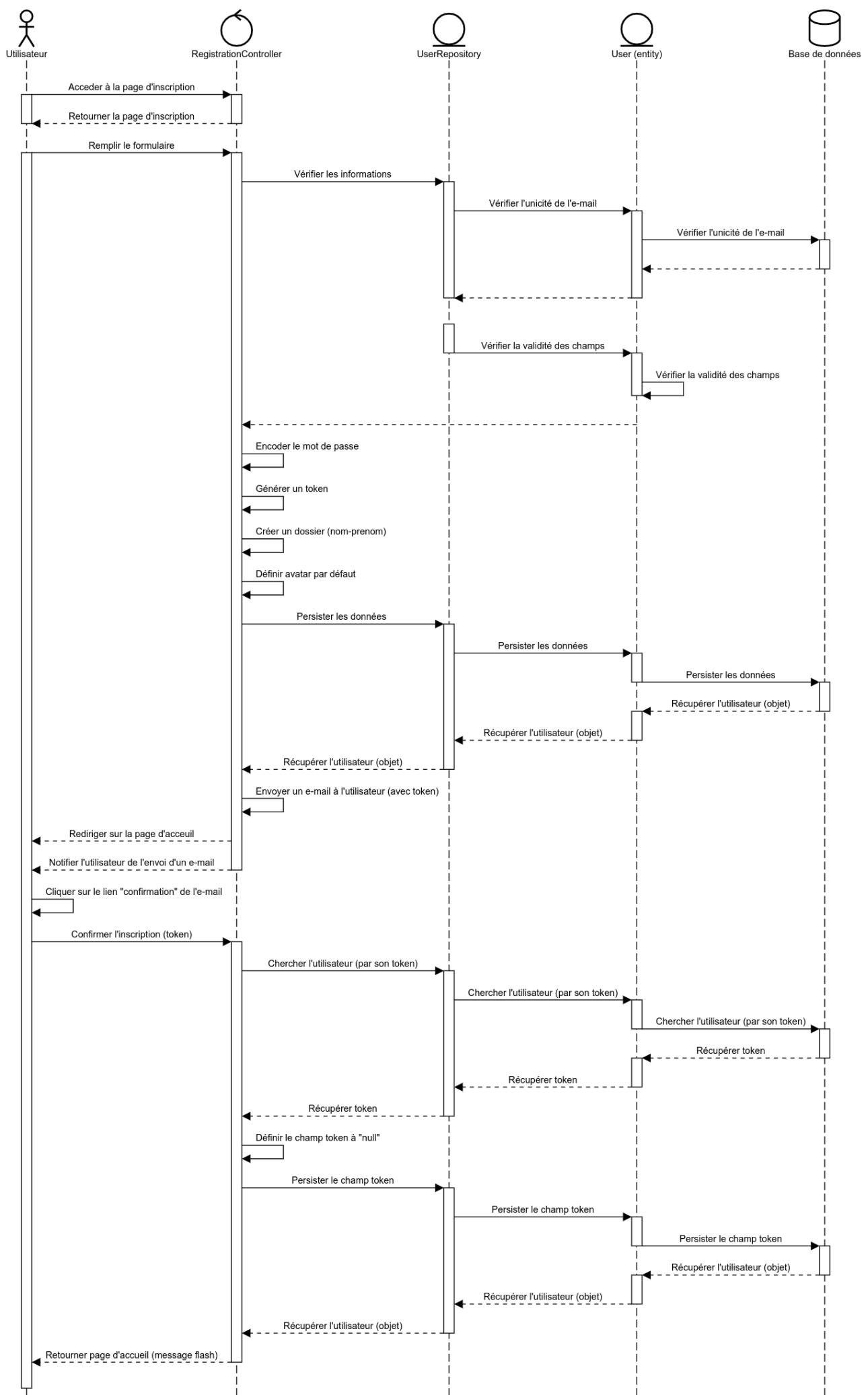


Figure 18 - Diagramme de séquence d'inscription.

V. Spécifications techniques du projet

5.1 Choix technologiques

5.1.1 Framework Symfony

Après avoir suivi six jours de cours sur le cadriel PHP Symfony au sein de la formation et ayant adopté l'architecture MVC proposée par celui-ci, j'ai choisi d'utiliser **Symfony** qui est un puissant framework (cadriel) open source permettant la conception de sites, d'applications web et intranets d'envergure importante. C'est un outil fiable, efficace et technologiquement avancé. Cela m'a permis d'approfondir mes connaissances en PHP, en Programmation Orientée Objet ainsi que de monter en compétences sur ce framework.

Structure d'une application Symfony (version ≥ 5)

assets/ Ce dossier gère les fichiers statiques de type CSS, JavaScript, images.

bin/ Contient l'ensemble des commandes de terminal que l'on peut lancer.

config/ Ce dossier est en quelque sorte le tableau de bord de votre application. Il contient l'ensemble des fichiers de configuration des services, bundle et autre module.

public/ Dossier de base de l'application. C'est en quelque sorte le fichier d'entrée, la racine publique du projet.

build/ Ce dossier contient notamment les fichiers de styles et JavaScript une fois compiler et minifier par WebPack¹².

src/ Dossier qui contient l'ensemble du code back end de l'application. Il contient une structure de plusieurs dossiers :

Controller/ Contient les fichiers contrôleurs. Fichiers ayant rapport avec les traitements de l'application.

Entity/ Contient des classes php qui servent à la représentation de la base de données.

DataFixtures/ Contient le jeu de données permettant de remplir la base de données.

Form/ Contient l'ensemble des classes spécifiques à la création de champs de formulaires.

Migrations/ Contient les fichiers de migrations en liaison avec la base de données.

Repository/ Contient les repository (permettant la sélection de données sous forme de méthode dans la base de données via l'ORM Doctrine).

12 - Webpack est un bundler de modules. Son objectif principal est de regrouper les fichiers JavaScript à utiliser dans un navigateur, mais il est également capable de transformer, regrouper ou empaqueter à peu près n'importe quelle ressource ou actif.

Security/ Contient les classes relatives à la sécurité de l'application, notamment pour l'authentification.

Service/ Permet de développer des services utilisés régulièrement, tel que l'envoi d'e-mail.

templates/ L'ensemble des vues de l'application.

tests/ Contient les fichiers pour la mise en place de tests.

var/ Contient les fichiers de cache et les logs (journaux) de l'application.

vendor/ Dossier inhérent à l'utilisation de composer. Il contient l'ensemble des librairies de votre application installé via composer (ce dossier n'est pas poussé sur GitHub).

.env Fichier de configuration global de votre application. Il contient par exemple des variables d'environnement ou bien l'url d'accès à la base de données (ce fichier n'est pas poussé sur GitHub).

webpack.config.js Fichier de configuration pour le compacteur de fichiers JavaScript, SASS. Et images.

5.1.2 Versionnage

J'ai utilisé Git pour le versionnage du code source et la plateforme Github pour déposer mon projet. Un système de branches a été utilisé avec comme ligne directrice de créer une branche par fonctionnalité (ex : **feature/authentication** ; branche dédiée à une fonctionnalité unique portant ici sur toute la partie authentication). Ce système de branche permet de découpler chaque fonctionnalité, dans le but d'avoir un code bien structuré, permettant la résolution d'éventuels problèmes à moyen / long terme plus aisée, mais aussi pour pouvoir rendre plus accessible l'évolutivité du code source. Enfin, GitHub propose entre autre, un système de wiki permettant la documentation, un système de ticket pour la mise en évidence et la résolution de problèmes et prend en charge le langage Markdown¹³ pour l'édition de documents veillant à spécifier les étapes d'installations ou de configuration d'un dépôt.

5.1.3 Analyse, conception et modélisation

J'ai utilisé le logiciel Visual Paradigm pour la création de diagrammes de séquence, de cas d'utilisation et d'activité, ainsi que les logiciels open source et gratuits Dbeaver Community pour la création de diagramme d'entités relationnelles et PgAdmin4 permettant la gestion des droits d'accès aux bases et entre autres la création de scripts SQL (fonctions, vues, procédures, déclencheurs etc.).

13 - Markdown est un langage de balisage léger avec une syntaxe de formatage de texte brut.

5.1.4 Base de données

J'ai choisi d'utiliser le Système de Gestion de Base de Données Relationnelle open source et gratuit (SGBDR) PostgreSQL car il met l'accent sur l'extensibilité et la conformité SQL. J'ai dans un premier temps réalisé une modélisation d'un diagramme d'entités relationnelles après avoir ciblé les fonctionnalités désirées. En effet, je souhaitais mettre en place un système de rôles, pour pouvoir développer toute une partie authentification et restreindre l'accès au cœur de l'application, à savoir l'administration du site. Ainsi, un utilisateur non authentifier peut naviguer librement sur le site. Seul l'administrateur peut créer des articles et des projets, un utilisateur classique peut, s'il est authentifié, publier un commentaire ou répondre à un commentaire sur un article choisi, éditer son profil et téléverser un avatar.

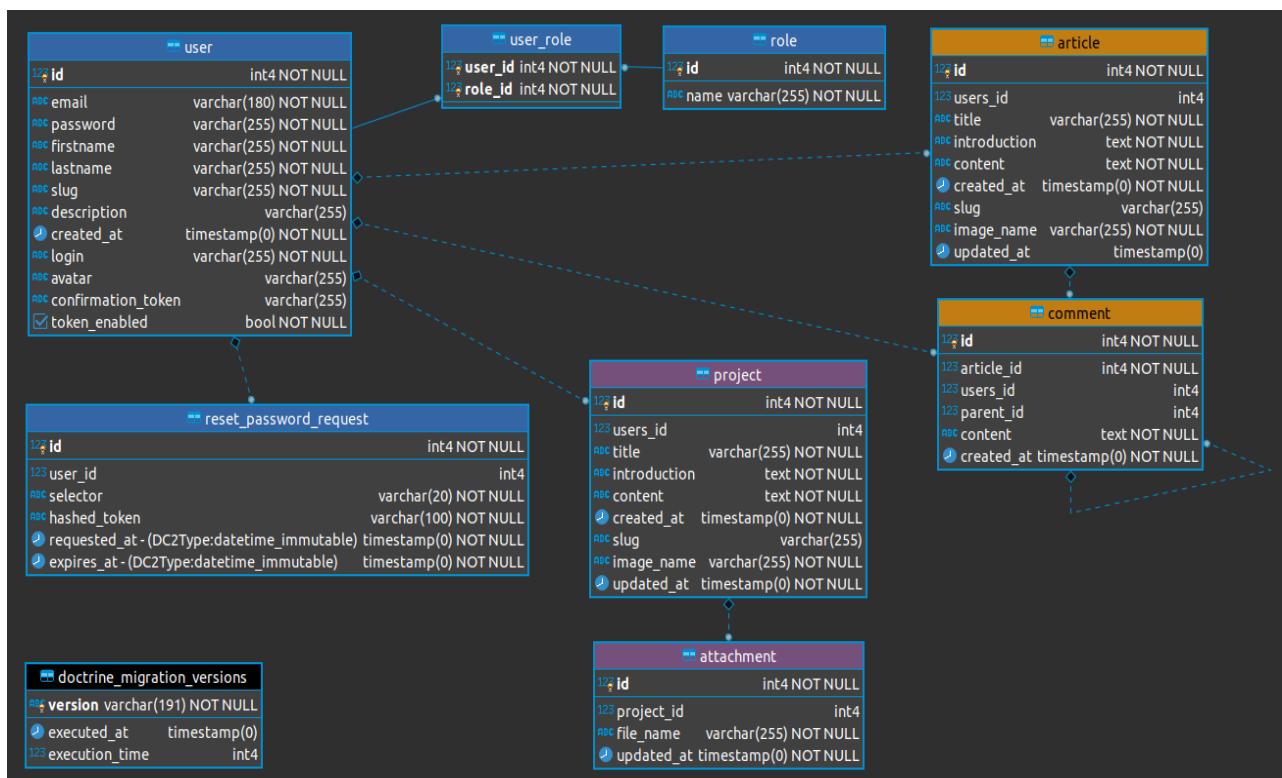


Figure 19 - Diagramme d'Entités Relationnelles (ERD)

5.2 Domaine et hébergement

Pour l'hébergement de mon site, j'ai choisi de le faire sur un Raspberry PI 4 que je possède à mon domicile, dans le but d'éviter des paiements récurrents et coûteux. Pour ce faire, j'ai dû installer les outils nécessaires au développement du site sur ce périphérique configuré en serveur Web et acheter un nom de domaine. J'ai choisi la société OVH¹⁴ (car française et ayant une bonne réputation ainsi qu'un très bon support en ligne) pour acheter le nom de domaine (slashflex.io). Le

14 - OVH est une société française de cloud computing qui propose des serveurs privés sécurisés (VPS), des serveurs dédiés et d'autres services web.

budget total de mon application revient à 12€ TTC par an (le prix du nom de domaine). Sinon aucun autre frais annexe n'est à prévoir.

5.3 Accessibilité

Le site est accessible sur l'ensemble des navigateurs actuels mise à part Internet Explorer car de moins en moins utilisé (moins de 2 % de part de marché mondial en février 2020). Ainsi, j'ai mis l'accent sur les navigateurs les plus utilisés, à savoir Chrome et Firefox.

J'ai utilisé SASS et la syntaxe SCSS afin de créer un système permettant la mise à l'échelle du site sur tout les support possibles, via une fonction « **respond()** » (voir figure 20) prenant en paramètre une taille d'écran (« **\$breakpoint** »).

```
@mixin respond($breakpoint) {
  @if $breakpoint == phone-small {
    @media only screen and (max-width: 25em) {
      @content;
    } // phone-small = 400px / 16px = 25em
  }
  @if $breakpoint == phone-mid {
    @media only screen and (min-width: 25.0625em) and (max-width: 28.125em) {
      @content;
    } // phone-mid = between 401px / 16px = 250625em and 450px / 16px = 28.125em
  }
  @if $breakpoint == phone {
    @media only screen and (max-width: 37.5em) {
      @content;
    } // phone = 600px / 16px = 37.5em
  }
  @if $breakpoint == tab-port {
    @media only screen and (max-width: 56.25em) {
      @content;
    } // tab-port = 900px / 16px = 56.25em
  }
  @if $breakpoint == tab-land {
    @media only screen and (max-width: 75em) {
      @content;
    } // tab-land = 1200px / 16px = 75em
  }
  @if $breakpoint == big-desktop {
    @media only screen and (min-width: 112.5em) {
      @content;
    } // big-desktop = 1800px / 16px = 112.5em
  }
}
```

Figure 20 - Fichier « **_mixins.scss** » permettant l'adaptation des blocs HTML en fonction des tailles d'écrans.

5.4 Évolution

J'ai choisi de rédiger l'ensemble du projet ainsi que tout les contenus en anglais, afin d'améliorer mon anglais. Je prévois néanmoins d'ajouter une fonctionnalité permettant de pouvoir traduire le site suivant la localisation où se trouve l'internaute au moment où il le consulte. Je prévois également de maintenir à jour les différentes dépendances utilisées, régler d'éventuelles failles de sécurité ainsi qu'ajouter du nouveau contenu dans la partie blog sous la forme d'articles. Enfin, j'envisage la refactorisation de certaines méthodes de classes générant une trop grande complexité et ne respectant pas le principe de responsabilité unique (évoquée en première partie de rapport).

VI. Réalisations : conception et codage

6.1 Développement de composants d'accès aux données

Pour la création de ce site, j'ai choisi d'utiliser le Système de Gestion de Base de Données Relationnelle PostgreSQL et l'ORM¹⁵ Doctrine couplés à l'interface graphique de gestion de base de données PgAdmin4. J'ai tout d'abord évaluer les besoins de mon application au travers de schémas papier que j'ai par la suite modélisé (voir Figure 19). Enfin, j'ai pu mettre en place ces relations de tables grâce au paquet « **symfony/maker-bundle** » au sein de mon projet. Ce paquet, installé via Composer, rend accessible une multitude de lignes de commandes permettant, via un assistant en ligne de commande, de créer des contrôleurs, des entités (via l'ORM Doctrine), des validateurs pour formulaires, etc.

Une fois satisfait des classes d'entités, je peux à l'aide du paquet Composer « **maker-bundle** », générer les entités (tables) du projet (voir Figure 23).

L'étape de création d'entités terminée, l'assistant de configuration (wizard¹⁶) « **maker** » a créé deux fichiers par entité :

- Le fichier de classe « **User.php** » (entité), comprenant tout les champs spécifiés, ainsi que leurs assesseurs (« **getters** » et « **setters** ») pour la création ou la récupération de champs.
- Le fichier de classe « **UserRepository.php** », permettant le dialogue entre une ou plusieurs entités et la base de données. Ce sont les classes Repository qui construisent les requêtes

15 - Object-Relational mapping : Mappage objet-relationnel en français, est un type de programme informatique qui se place en interface entre un programme applicatif et une base de données relationnelle pour simuler une base de données orientée objet.

16 - Assistant logiciel ou assistant de configuration est un type d'interface utilisateur qui présente à l'utilisateur une séquence de boîtes de dialogue qui guident l'utilisateur à travers une série d'étapes bien définies. Les tâches complexes, rarement exécutées ou inconnues peuvent être plus faciles à exécuter à l'aide d'un assistant.

SQL pour la création, mise à jour, suppression et la lecture de données via une syntaxe très simple et disponible avec l'ORM Doctrine. L'on peut également, dans ces fichiers Repository construire ses propres requêtes SQL sous forme de méthodes à l'aide de DQL (Doctrine Query Language).

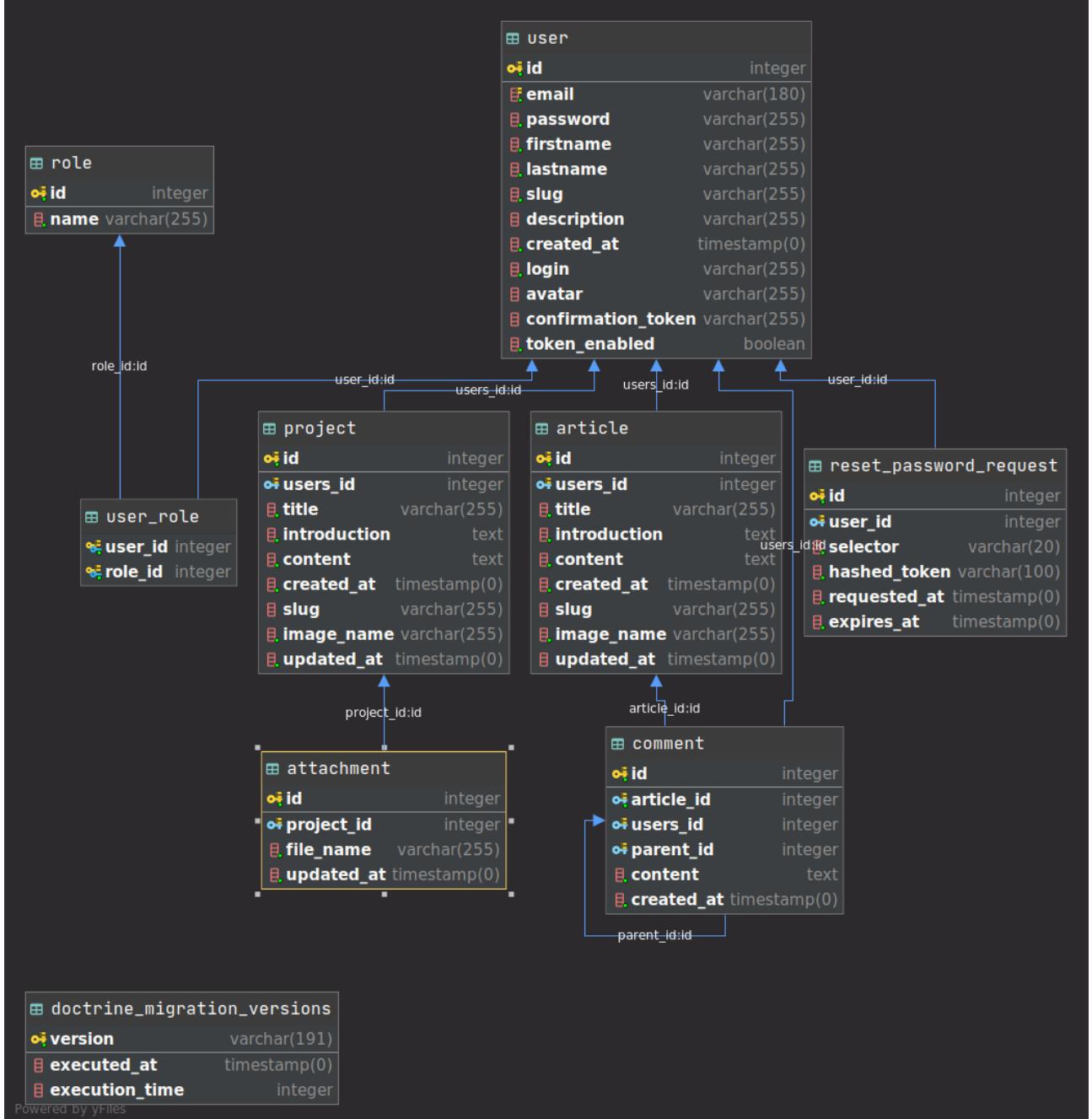


Figure 21 - Diagramme de classe des entités (tables).

Sur la figure 22, l'on peut voir les attributs « **private** » (attributs de classe avec un cadenas rouge) ainsi que leurs assesseurs « **public** » (méthodes de classe avec un cadenas vert) générés par le guide de configuration « **maker** ». Les relations d'entités sont également renseignées pendant la configuration des entités. J'ai donc désormais accès à tous ces « **getters** » et « **setters** » dans mon

application. Les getters sont des méthodes de classe pour récupérer des données en base de données et les setters sont disponibles pour affecter ou réaffecter des données dans les champs dédiés.

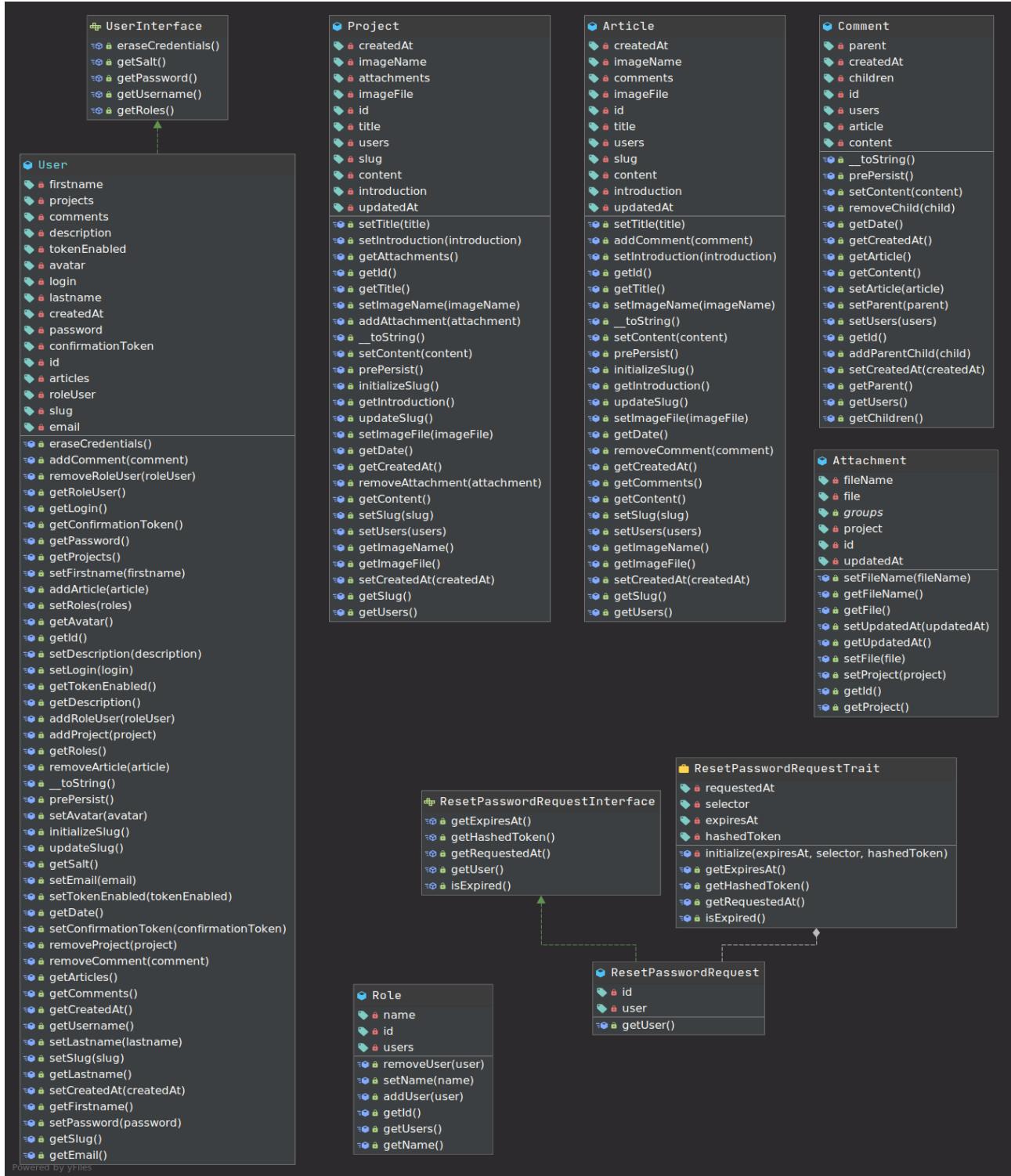


Figure 22 - Attributs et méthodes de classe d'entités.

Exemple : pour trouver et récupérer tout les objets utilisateurs enregistrés en base de données et les afficher sur la page du panneau d'administration du site, on peut, dans la méthode du contrôleur dédié, « demander » à la classe « **UserRepository** » d'effectuer cette action (voir figure 24).

```
x slashflex ~ /Applications/ TESTS /PHP/SYMFONY/rapport master bin/console make:entity User
created: src/Entity/User.php
created: src/Repository/UserRepository.php

Entity generated! Now let's add some fields!
You can always add more fields later manually or by re-running this command.

New property name (press <return> to stop adding fields):
> firstname

Field type (enter ? to see all types) [string]:
>

Field length [255]:
>

Can this field be null in the database (nullable) (yes/no) [no]:
>

updated: src/Entity/User.php

Add another property? Enter the property name (or press <return> to stop adding fields):
> lastname

Field type (enter ? to see all types) [string]:
>

Field length [255]:
>

Can this field be null in the database (nullable) (yes/no) [no]:
>

updated: src/Entity/User.php

Add another property? Enter the property name (or press <return> to stop adding fields):
> email

Field type (enter ? to see all types) [string]:
>

Field length [255]:
>

Can this field be null in the database (nullable) (yes/no) [no]:
>

updated: src/Entity/User.php
```

Figure 23 - Exemple de création d'entité à l'aide de l'assistant en ligne de commande **maker**.

```

<?php

namespace App\Controller;

use App\Entity\User;
use App\Repository\UserRepository;
use Symfony\Component\Routing\Annotation\Route;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\IsGranted;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;

class AdminController extends AbstractController
{
    private $userRepository;

    public function __construct(
        UserRepository $userRepository
    ) {
        $this->userRepository = $userRepository;
    }

    /**
     * Admin dashboard that displays all users
     *
     * @Route("/admin", name="admin")
     * @IsGranted("ROLE_ADMIN")
     */
    public function index()
    {
        $users = $this->userRepository->findAll();
        $admin = $this->getUser();

        return $this->render('admin/index.html.twig', [
            'title' => '/ FLX Admin dashboard',
            'users' => $users,
            'admin' => $admin
        ]);
    }
}

```

Figure 24 - Méthode de contrôleur récupérant tout les utilisateurs en base de données sous forme d'objets.

(voir figure 24). J'importe tout d'abord la classe **UserRepository**, puis j'injecte cette dépendance dans le constructeur de la classe du contrôleur, afin d'y avoir accès dans toutes les méthodes de la classe **AdminController** à l'aide du mot-clé « **\$this** ».

Ici, nous avons accès à toutes les méthodes disponibles via la classe **UserRepository**, soit dans ce cas, « **findAll()** » qui récupérera sous forme d'objets (via une requête SQL préparée et exécutée par l'ORM Doctrine de type : **SELECT * FROM user**), tout les utilisateurs existant en base de données. Enfin, je retourne la page souhaitée avec en paramètre, un tableau d'objets

utilisateurs, via la méthode « `render()` » de TWIG. Ainsi, l'on peut utiliser ces objets dans la vue à l'aide de la syntaxe de TWIG.

Une fois les entités créées sous forme de classes PHP et possédant des accesseurs (« `getters` » et « `setters` ») pour la création ou la récupération de champs, j'ai mis en place un système de jeu de données fictives appelées « `fixtures` » à l'aide du paquet « `faker` », afin de peupler la base de données de valeurs reflétant un usage réel avec l'installation du paquet « `doctrine/doctrine-fixtures-bundle` » afin de vérifier le bon fonctionnement des relations.

```
<?php

namespace App\DataFixtures;

use Faker\Factory;
use App\Entity\Role;
use App\Entity\User;
use App\Entity\Article;
use Doctrine\Persistence\ObjectManager;
use Doctrine\Bundle\FixturesBundle\Fixture;
use Symfony\Component\Security\Core\Encoder\UserPasswordEncoderInterface;

class AppFixtures extends Fixture
{
    private $passwordEncoder;

    /**
     * Cryptographic hash class
     * use : $passwordEncoder->encodePassword($user, 'plain_password')
     *
     * @param UserPasswordEncoderInterface $passwordEncoder
     */
    public function __construct(UserPasswordEncoderInterface $passwordEncoder)
    {
        $this->passwordEncoder = $passwordEncoder;
    }

    /**
     * Persist data to database
     * command : bin/console doctrine:fixtures:load
     *
     * @param ObjectManager $manager
     * @return void
     */
    public function load(ObjectManager $manager)
    {
        // Initialize Faker with french language
        $faker = Factory::create('fr_FR');
        // Create "ROLE_ADMIN" role
        $roleAdmin = new Role();
        $roleAdmin->setName('ROLE_ADMIN');
        // Create "ROLE_USER" role
        $roleUser = new Role();
        $roleUser->setName('ROLE_USER');

        $manager->persist($roleUser);

        // Create an admin user with default "ROLE_USER" + "ROLE_ADMIN"
        $userAdmin = new User();
        $userAdmin
            ->setFirstname($_ENV['DB_FIRSTNAME'])
            ->setLastname($_ENV['DB_LASTNAME'])
            ->setEmail($_ENV['DB_EMAIL'])
            ->setConfirmationToken(null)
            ->setTokenEnabled(true)
            ->setAvatar([
                'https://avatars0.githubusercontent.com/u/44773291?s=400&u=95e5053803b58ea34e747e4c00e42547705a2513&v=4'
            ])
            // Hash password
            ->setPassword($this->passwordEncoder->encodePassword($userAdmin, $_ENV['DB_PASSWORD']))
            ->setDescription($faker->paragraph(2))
            ->setLogin($faker->name())
            ->addRoleUser($roleAdmin)
    }
}
```

```

    →addRoleUser($roleUser)
    →initializeSlug();

$manager→persist($userAdmin);

$sexes = ['male', 'female'];
$users = [];

// Create 12 random users with random informations
for ($i = 1; $i ≤ 12; $i++) {
    $user = new User();
    $avatar = "https://randomuser.me/api/portraits/";
    $sexe = $faker→randomElement($sexes);
    $photoId = $faker→numberBetween(1, 99) . '.jpg';
    $avatar .= ($sexe == 'male' ? 'men/' : 'women/') . $photoId;

    $user
        →setFirstname($faker→firstName($sexe))
        →setLastname($faker→lastName)
        →setEmail($faker→email)
        →setAvatar($avatar)
        →setPassword($this→passwordEncoder→encodePassword($user, 'pwd'))
        →setDescription($faker→paragraph(2))
        →setLogin("user-$i")
        →addRoleUser($roleUser)
        →initializeSlug();

    $manager→persist($user);
    $users[] = $user;
}

// Create 20 random articles with random informations
for ($i = 1; $i ≤ 20; $i++) {
    $article = new Article();
    $contenu = '<p>' . join('</p><p>', $faker→paragraphs(mt_rand(20, 40))) . '</p>';
    $titre = $faker→sentence(3);
    $introduction = $faker→paragraph(2);

    $article
        →setTitle($titre)
        →setIntroduction($introduction)
        →setContent($contenu)
        →setUsers($userAdmin)
        →initializeSlug();
    $article→setImageName('random.jpg');

    $manager→persist($article);
}

$manager→flush();
}
}

```

Ainsi, je possède désormais en base de données ; 1 utilisateur ayant les rôles « **ROLE_ADMIN** et **ROLE_USER** », 12 utilisateurs possédant le rôle « **ROLE_USER** » ainsi que 20 articles. J'ai donc désormais accès à ces données dans mon application et je peux mettre en place par exemple un système de restriction d'accès aux pages requérant un rôle « **ROLE_ADMIN** ».

6.2 Fonctionnalités les plus importantes

Les fonctionnalités les plus importantes du site sont :

- Conception et mise en place d'une base de données relationnelle PostgreSQL,
- Un système complet d'authentification comprenant :
 - Un module de connexion sécurisé,
 - Un module d'inscription sécurisé avec système de confirmation,
 - Un module de réinitialisation de mot de passe.
- Le téléversement, téléchargement de fichiers et l'édition de documents au format PDF,
- Une API dédiée à la création de commentaire(s) grâce à une table de base de donnée récursive,
- La communication avec deux API de Google (Oauth et reCaptcha v3).

Je souhaitais également mettre en place un système de connexion à l'aide d'un réseau social (en l'occurrence Google). Cette méthode de connexion nécessite tout de même d'avoir au préalable effectué une inscription sur le site. Pour ce faire, j'ai dû effectuer une veille technologique sur les API de Google permettant l'authentification. J'ai pu utiliser ces mots clés pour mes recherches :

- *implement google auth* : lien retourné et choisi :
 - <https://developers.google.com/identity/sign-in/web/sign-in>, lecture et compréhension de la documentation.
- *google auth symfony* : lien retourné et choisi :
 - <https://github.com/knpuniversity/oauth2-client-bundle>, j'ai choisi ce site puisque qu'il correspondait complètement aux mots clé de ma recherche, il s'agit d'un package à installer en ligne de commande avec Composer¹⁷, de plus le lien retourné nous dirige sur le site GitHub, grâce à ce package et sa documentation pertinente, j'ai pu mettre en place un système de connexion via Google sur mon site en peu de temps.

17 - Outil de gestion de dépendances.

6.2.1 Module d'inscription

Ce module permet aux utilisateurs non enregistrés en base de données de pouvoir créer un compte sur le site après avoir entré leurs informations personnelles dans un formulaire dédié. Ces informations sont composées comme suit :

- Une adresse e-mail valide et unique en base de donnée respectant un format d'e-mail type,
- Un prénom et nom, devant contenir entre 3 et 20 caractères inclus,
- Un mot de passe comprenant au minimum 8 caractères, qui est crypté avant le stockage en base de donnée à l'aide de l'algorithme de hachage Bcrypt,
- Une confirmation du mot de passe, qui doit être strictement identique au champ de mot de passe décrit précédemment,
- Un pseudonyme, qui est utilisé sur le site afin de ne pas afficher publiquement l'adresse e-mail d'un utilisateur aux autres membres inscrits,
- Un bouton de type radio requérant d'accepter les termes d'utilisation du site.

Tout ces champs sont requis pour une inscription et possèdent leurs propres validateurs avec un système de notification du ou des champ(s) invalide(s), enfin, afin de détecter et réduire le spam de robots (attaque par force brute) et de trafic sur l'application, j'ai mis en place le système ReCaptcha¹⁸ version 3 de Google grâce à son API ainsi qu'un système de token CSRF¹⁹.

Une fois tous ces champs de formulaire validés et conformes aux standards de sécurité mis en place et préconisés par l'OWASP²⁰, le mot de passe est alors encrypté puis les données de l'utilisateur sont persistées en base de données, l'utilisateur est alors redirigé sur la page d'accueil et une notification lui indique qu'un e-mail lui demandant de confirmer son inscription vient de lui être envoyé, celui-ci se verra dans l'impossibilité de se connecter sur le site tant qu'il n'aura pas confirmé son inscription via l'e-mail reçu. En effet, à la validation du formulaire un token²¹ est généré et envoyé à l'utilisateur dans l'e-mail sous la forme d'un lien à cliquer, ce token expire au bout de deux heures. À l'expiration de ce token, l'utilisateur devra alors effectuer une demande d'envoi d'un nouvel e-mail confirmation sur la page de connexion de l'application où, un nouvel e-mail lui sera envoyé après que l'utilisateur ait renseigné son adresse e-mail. Au moment où

18 - Système de détection de trafic abusif ne demandant pas d'interaction à l'utilisateur.

19 - Cross Site Request Forgery : type de vulnérabilité des services d'authentification web. L'objet de cette attaque est de transmettre à un utilisateur authentifié une requête HTTP falsifiée qui pointe sur une action interne au site, afin qu'il l'exécute sans en avoir conscience et en utilisant ses propres droits.

20 - Open Web Application Security Project : est une fondation à but non lucratif qui travaille à améliorer la sécurité des logiciels et applications Web.

21 - Mot anglais pour désigner un code unique généré aléatoirement parmi une combinaison de caractères.

l'utilisateur aura cliqué sur le lien contenu dans l'e-mail, son compte sera alors confirmé en base de données via un champ spécifique de type *booléen* (token_enabled passant à *true*) et un champ de type *string* (confirmation_token) contenant le token qui sera quant à lui changé en *null*.

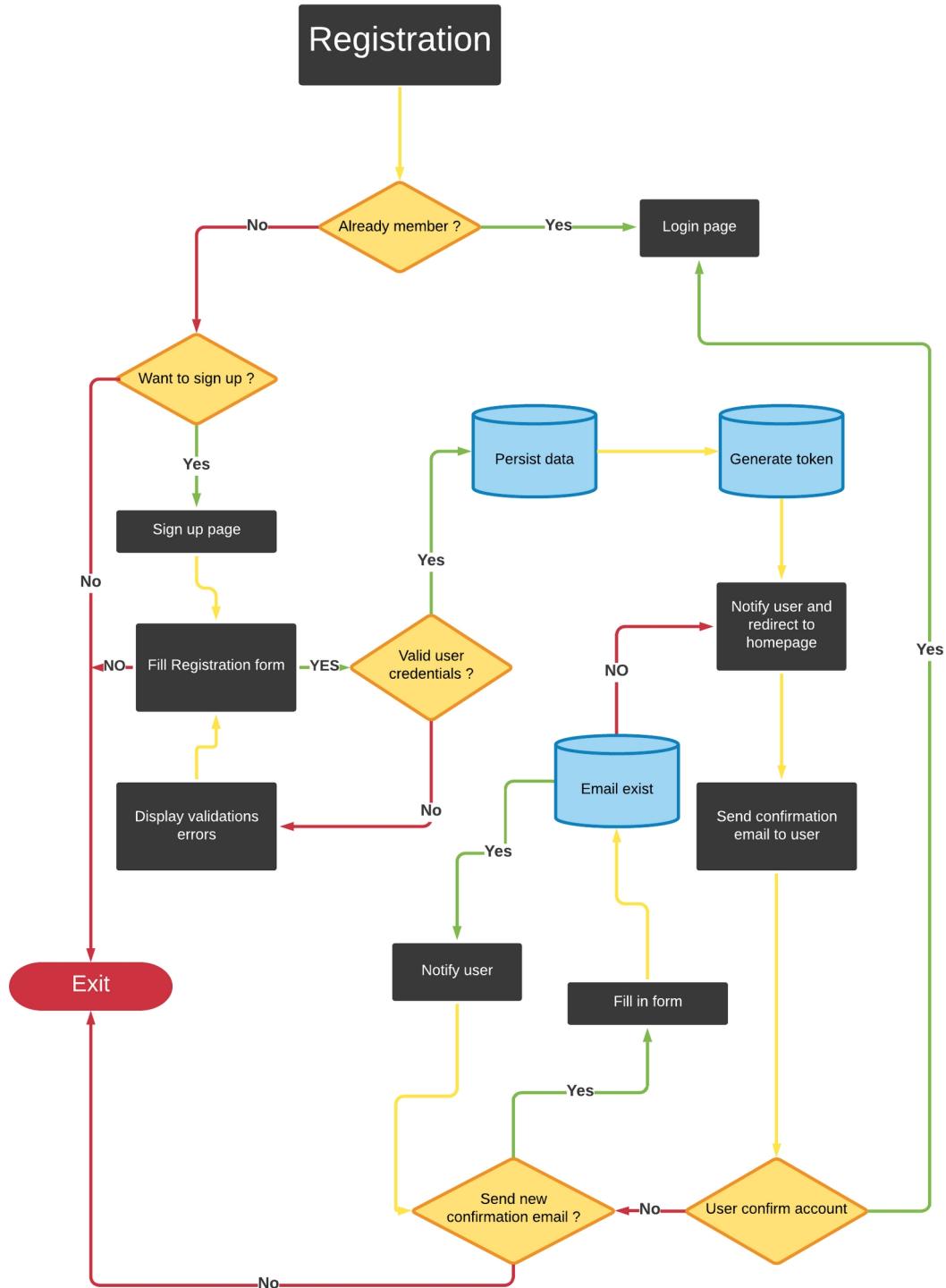


Figure 25 - Logique mise en place pour la création d'un compte.

La figure 25 illustre la logique mise en place dans l'application et le cheminement d'un utilisateur souhaitant s'enregistrer.

6.2.2 Module de connexion

Ce module permet à un utilisateur enregistré en base de données, de pouvoir se connecter de deux façons :

- en indiquant son e-mail et mot de passe renseignés au cours de son inscription,
- en se connectant à l'aide de l'API Google (cependant, lors de la connexion avec cette API, l'e-mail renseigné lors de l'inscription devra être identique à celui auquel l'utilisateur souhaite se connecter via Google, sinon une erreur sera levée et l'utilisateur en sera notifié).

Un système de validateurs définit directement dans les classes entités, précise à l'utilisateur si les informations sont incorrectes, mais n'affiche pas explicitement quel champ est invalide, afin de ne pas donner de piste à d'éventuelles attaques par force brute. Une fois connecté l'utilisateur est alors redirigé sur la page de son profil personnel.

La figure 26 illustre la logique mise en place dans l'application et le cheminement d'un utilisateur souhaitant se connecter.

6.2.3 Gestion profile utilisateur

Ce module permet à un utilisateur connecté de pouvoir téléverser un avatar à partir de son ordinateur. Ce fichier doit respecter un type MIME²² spécifique (image/gif, image/png, image/jpg, image/jpeg, image/webp ou image/gif uniquement), la taille du fichier téléchargé ne doit pas être supérieure à 2 mégaoctets. Il peut également mettre à jour ou modifier ses informations personnelles (e-mail, mot de passe, prénom, etc.) et se déconnecter. Je prévois d'ajouter une section aux profils utilisateurs leur permettant de voir l'ensemble des commentaires publiés et des réponses d'autres utilisateurs reçues sur leurs commentaires.

6.2.4 Module d'administration

Ce module permet à l'administrateur du site, d'avoir la main mise sur l'entièreté du site (création, mise à jour et suppression d'articles, de projets et d'utilisateurs). En effet, une page dédiée à ces fonctionnalités est accessible uniquement à l'administrateur. Un système de rôles étant en place, les utilisateurs créant un compte se verront automatiquement attribué le rôle : **ROLE_USER**, l'administrateur possède lui les rôles : **ROLE_USER** et **ROLE_ADMIN**. La page

22 - Identifiant de format de données sur internet en deux parties

dédiée aux tâches de création, mise à jour et suppression n'est accessible que si l'utilisateur possède le rôle ***ROLE_ADMIN***. Si l'utilisateur ne possède que le rôle ***ROLE_USER***, à la soumission du formulaire de connexion pour accéder à la partie réservée à l'administrateur, il sera redirigé sur une page d'erreur 404 : page non trouvée (l'erreur levée exacte est en fait une erreur HTTP possédant le code 403 pour : accès interdit (pour manque d'autorisation)).

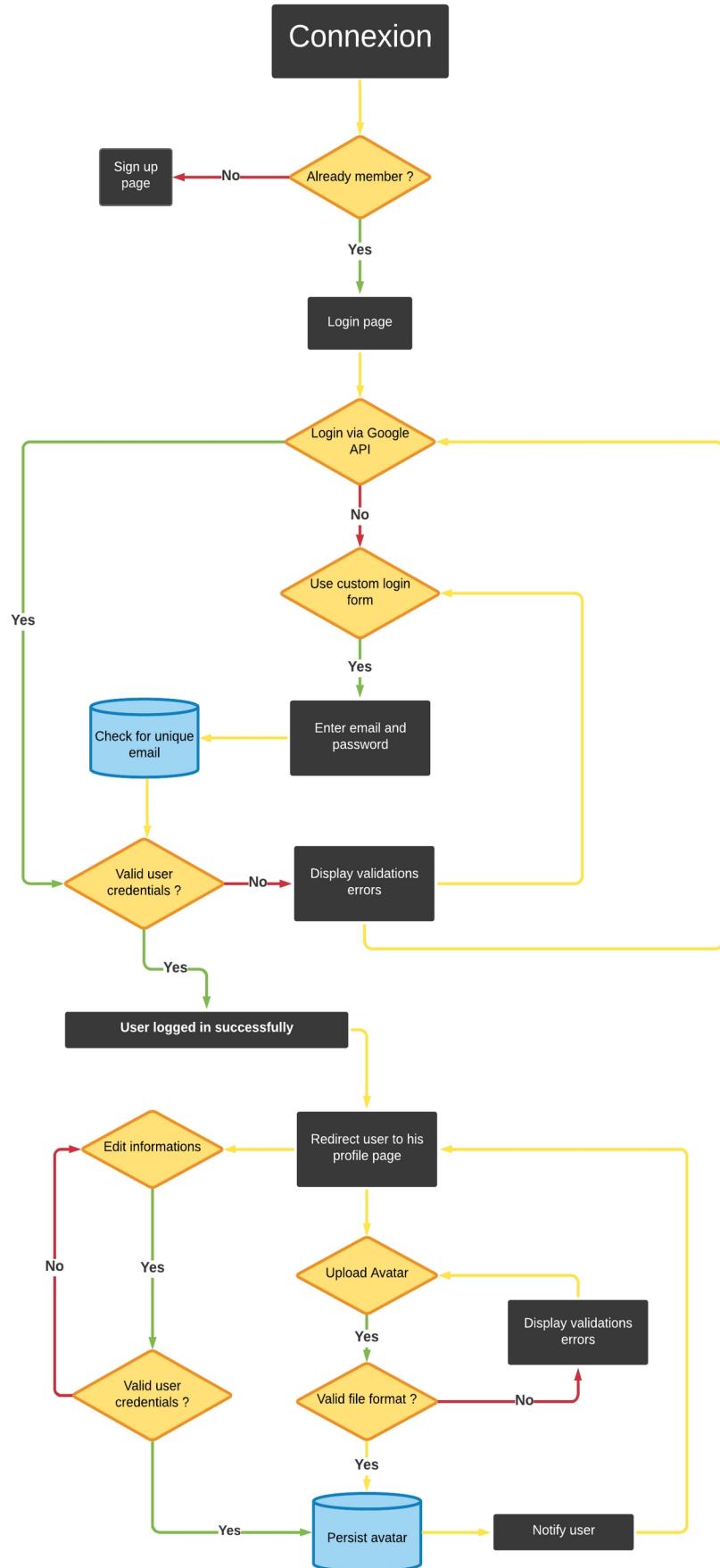


Figure 26 - Logique mise en place pour la connexion.

VII. Présentation du jeu d'essai

J'ai mis en place une série de tests sur mon projet visant à tester le bon fonctionnement de mon application et aidant au développement des différents fonctionnalités du projet. Pour ces tests, j'ai utilisé le framework « **PHPUnit** ».



J'ai également mis en place la bibliothèque « **Panther**²³ » afin d'effectuer un test visant à reproduire le comportement d'un utilisateur sur une partie ciblée du site (l'inscription). Cette bibliothèque permet la mise en place de tests E2E (end to end).

L'installation se fait via Composer à l'aide de la commande « **composer req --dev symfony/panther** » (--dev = dépendances de développement).

Ci-dessous, j'ai choisi de détailler le test d'une fonctionnalité importante du site, à savoir l'inscription d'un utilisateur.

7.1 Données en entrées

Les données en entrées pour ce test sont les informations renseignées par l'utilisateur dans le formulaire d'inscription du site. Le formulaire contient 6 champs de type « **input** », générés via la commande « **bin/console make:form Registration** ». Le test se déroule comme suit, je créé le navigateur primaire grâce à la méthode « **protected** » et « **static** » `createPantherClient()`, je demande ensuite au crawler d'effectuer une requête HTTP sur l'URL à tester puis de sélectionner le bouton de soumission du formulaire (*Register*) avec en paramètre un tableau de données issues des différents champs spécifiés. Je demande ensuite au navigateur de lancer deux scripts JavaScript ; un pour cliquer sur le bouton de type radio « accepter les termes d'utilisation » et l'autre pour soumettre le formulaire.

23 - Bibliothèque de tests de navigateur (comparable à Selenium) pour PHP et Symfony permettant notamment la mise en place de tests E2E (end to end). Cette méthode vise essentiellement à reproduire des scénarios d'utilisateurs réels afin que le système puisse être validé pour l'intégration et l'intégrité des données.

```

60     public function testSuccessFullRegistration()
61     {
62         // ACT
63         $userEmail = 'john@doe.com';
64         $client = static::createPantherClient();
65         $crawler = $client->request( method: 'GET', uri: '/register');
66         $crawler->selectButton( value: 'Register')->form([
67             'registration_form[email]' => $userEmail,
68             'registration_form[firstname]' => 'john',
69             'registration_form[lastname]' => 'doe',
70             'registration_form[password][first]' => '012345678',
71             'registration_form[password][second]' => '012345678',
72             'registration_form[login]' => 'johnny',
73         ]);
74         // Perform click actions
75         $client->executeScript( script: "document.getElementById('registration_form_agreeTerms').click()");
76         $client->executeScript( script: "document.querySelector('.button__login.mx-auto.d-block').click()");
77         // ARRANGE
78         // Search for newly created user by it's email
79         $user = self::$container->get( id: UserRepository::class)->findOneByEmail($userEmail);
80         // ASSERT
81         // that registered user exist in database
82         $this->assertEquals($userEmail, $user->getEmail());
83         // that newly created user only have the role of user
84         foreach ($user->getRoles() as $role) {
85             $this->assertEquals( expected: 'ROLE_USER', $role);
86             $this->assertNotEquals( expected: 'ROLE_ADMIN', $role);
87         }
88         // that a folder is created with user firstname and lastname as slug
89         $this->assertDirectoryExists( directory: $_SERVER['DOCUMENT_ROOT'].'public/uploads/avatars/'.$user->getSlug().'/');
90     }

```

Figure 27 - Test E2E (end to end) de la fonctionnalité : inscription.

7.2 Données attendues

Pour les données attendues, je souhaite vérifier la persistance en base de données des informations renseignées en demandant au conteneur de services de Symfony de chercher un utilisateur en base de données par l'e-mail utilisé lors de la soumission du formulaire, qu'un dossier se nommant « prénom-nom » à bien été créé et que l'utilisateur ne possède que le rôle « **ROLE_USER** ». Pour lancer ce test, j'exécute la commande suivante :

```

x slashflex ➤ ~/Applications/www/slashflex.io ➤ master ±
env PANTHER_NO_HEADLESS=1 PANTHER_CHROME_BINARY=/usr/bin/googl
e-chrome ./bin/phpunit --filter testSuccessFullRegistration --
debug

```

Figure 28 - Ligne de commande pour lancer un test E2E via l'utilisation de la bibliothèque PANTHER.

(en figure 28), J'assigne en variable d'environnement du projet deux variables, la première pour activer le mode sans tête du navigateur (affiche l'onglet console du navigateur, utile pour déboguer), la seconde pour spécifier le fichier binaire pour le lancement du navigateur Chrome. Ensuite, je spécifie le binaire d'exécution de **PHPUnit** qui se trouve dans le dossier « **bin** » à la

racine du projet puis je filtre le nom de la méthode à tester à l'aide de l'option « **--filter** », enfin, j'autorise le débogage à l'aide de l'option « **--debug** ». Cette commande lancera donc le test de ma méthode dans une fenêtre de navigateur Chrome contrôlée par le bibliothèque Panther, cette dernière remplira les champs du formulaires à l'aide des données spécifiées dans la méthode à tester puis validera le formulaire. À l'issue de ce test, je m'attends donc à trouver un utilisateur en base de données, possédant le rôle « **ROLE_USER** » ainsi qu'un dossier nommé « **john-doe** » créé dans le dossier « **public/uploads/avatars/** » de mon application.

7.3 Résultat obtenu

Sur la figure suivante, l'on distingue la commande pour lancer le test E2E d'inscription d'un utilisateur. L'on distingue également que le test est un succès ainsi que le temps d'exécution de celui-ci. Il est à noter que ce temps est particulièrement court étant donné que cette commande effectue un nombre important de traitements et d'actions.

```
slashflex ~/Applications/www/slashflex.io master ± env PANTHER_NO_HEADLESS=1 PANTHER_CHROME_BINARY=/usr/bin/google-chrome ./bin/phpunit --filter testSuccessFullRegistration --debug
PHPUnit 7.5.20 by Sebastian Bergmann and contributors.

Testing Project Test Suite
Test 'App\Tests\Controller\SecurityControllerTest::testSuccessFullRegistration' started
Test 'App\Tests\Controller\SecurityControllerTest::testSuccessFullRegistration' ended

Time: 5.78 seconds, Memory: 32.50 MB
OK (1 test, 4 assertions)
```

Figure 29 - Résultat du test E2E obtenu.

Pour lancer l'ensemble de mes tests, j'utilise uniquement le binaire de PHPUnit se trouvant dans le dossier « **bin/** » à la racine de mon application.

```
slashflex ~/Applications/www/slashflex.io master ± ./bin/phpunit
PHPUnit 7.5.20 by Sebastian Bergmann and contributors.

Testing Project Test Suite
..... 34 / 34 (100%)

Time: 9.49 seconds, Memory: 54.50 MB
OK (34 tests, 52 assertions)
```

Figure 30 - Résultat des tests mis en place.

L'on peut voir en vert le nombre de tests ainsi que les assertions. On distingue également des points, ceux-ci restent blanc quand un test est juste, sinon, si un test échoue, l'on trouvera un F entouré de rouge (pour failure, échec en français) en lieu et place d'un point.

VIII. Veille sur les vulnérabilités de sécurités

Mon application possédant un système de rôles, j'ai dû chercher un moyen de pouvoir refuser l'accès aux utilisateurs ne possédant pas le rôle **ROLE_ADMIN**, à certaines pages dédiées à l'administration du site comme par exemple la route permettant d'accéder à la page de connexion du panneau d'administration. J'ai également cherché un moyen de me prémunir contre les attaques CSRF (évoquées précédemment).

Mots clés utilisés :

- *symfony 5 security* : lien retourné et choisi :
 - <https://symfony.com/doc/current/security.html>
- *symfony csrf* : lien retourné et choisi :
 - <https://symfony.com/doc/current/security/csrf.html>

Documentations officielles du framework Symfony, donc ressources les plus pertinentes car mises à jour à régulièrement. J'ai donc pu suivre les recommandations de sécurité décrites dans ces documentations et implémenter ces deux systèmes de sécurité sur mon site.

8.1 Mise en place d'audits de détection de failles de sécurités

J'ai installé, mis en place et utilisé une machine virtuelle Debian 10, afin de pouvoir soumettre mon site à divers types d'attaques (par force brute, falsification de requêtes inter-sites (CSRF), spams de bots et crawlers²⁴ etc...). Le logiciel utilisé pour effectuer ces attaques est OWASP ZAP²⁵ (Zed Attack Proxy). Cet outil a pour but de neutraliser un site internet par tout les moyens possibles, par exemple en remplissant la base de données de millions de champs avec des valeurs totalement farfelues.

Suite à un test de pénétration, un rapport est alors généré avec une série de solutions possibles à mettre en œuvre afin de se prémunir des failles détectées. Je tiens à préciser que l'utilisation d'un tel outil sur un site internet en production est strictement interdit, c'est pourquoi j'ai utilisé une machine virtuelle afin de me couper de l'Internet et pouvoir effectuer des tests de pénétration en local. Pour ce faire j'ai dû réinstaller et reconfigurer tout les outils utilisés pour le développement du site : Apache, PostgreSQL, Symfony, PHP, Composer, NodeJs et Npm, Git et les

24 - Robots d'exploration et d'indexation.

25 - Scanner de sécurité d'applications Web open source. Il est destiné à être utilisé à la fois par les novices en sécurité des applications et par les testeurs de pénétration professionnels.

reconfigurer spécifiquement pour cette machine virtuelle. Ensuite, j'ai cloné mon dépôt depuis Github et lancé une série d'attaques via le logiciel **ZAP**. Ma base de données n'a été affectée en aucun cas (pas d'ajout ou suppression de champs, pas d'erreur détectées). Quelques erreurs mineures ont été levées suite à la non sécurisation du site par certificat SSL, ceci étant complètement normal étant donné que l'attaque s'est effectuée en local et complètement isolé de l'Internet.

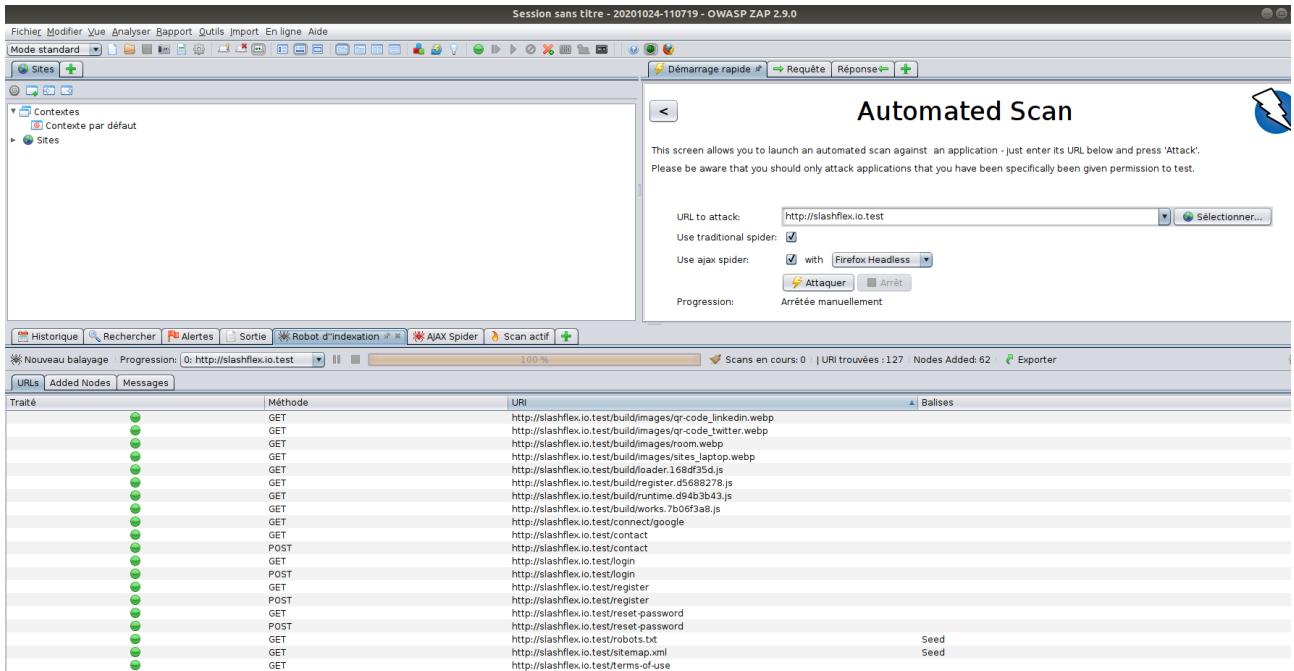


Figure 31 - Tableau de bord du logiciel ZED après test de pénétration.

J'ai également mis un place sur mon dépôt, un outil développé par Github (**Dependabot**). Cet outil a pour but de scanner les paquets de dépendances Composer et Npm ou Yarn d'un dépôt configuré pour utiliser cette fonctionnalité. Si l'outil détecte des failles de sécurité liées à des dépendances tierces, Github prévient par e-mail le(s) contributeur(s) du dépôt des marches à suivre pour s'en prémunir. Il s'agit pour la plupart du temps de mettre à jour une dépendance précise et spécifiée par l'outil Dependabot. C'est cet outil qui effectue une Pull Request (demande de tirage) sur le dépôt concerné.

Sur la figure 32, on aperçoit que l'outil Dependabot a effectué une demande de tirage pour la mise à jour du paquet « **symfony/http-kernel** » qui fixe une faille de sécurité. Je vérifie la pertinence de cette mise à jour en lisant les notes de sorties (« **Release notes** »), puis je choisis de fusionner cette demande de tirage sur la branche principale de mon dépôt (master) et je supprime la branche créée par Dependabot.

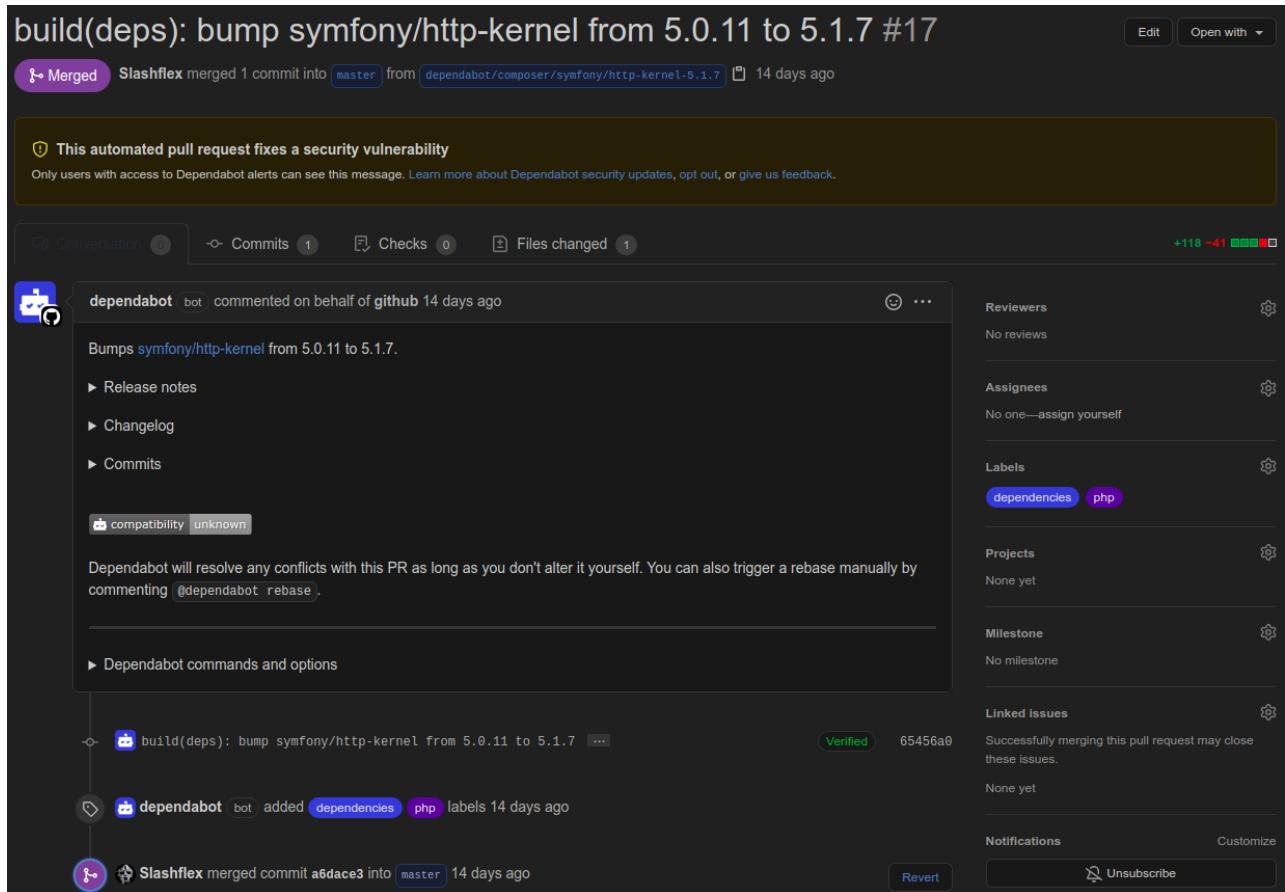


Figure 32 - Demande de tirage (pull request) effectuée par **DependaBot** sur une dépendance contenant une faille de sécurité.

IX. Situation de travail ayant nécessité une recherche

J'ai dû effectuer des recherches pour la mise en place des tests avec le framework Symfony et PHPUnit. Pour ce faire, j'ai utilisé Internet et le moteur de recherche de Google pour orienter mes recherches autour de mots clés précis et ayant du sens, comme : « **symfony unit testing** », « **end to end symfony** » ou encore « **phpunit symfony** ». Voici les liens retournés et que j'ai choisis pour leur pertinence, leur ancienneté et surtout leur provenance (documentations officielles de Symfony).

<https://symfony.com/doc/current/testing.html>

https://symfony.com/doc/current/components/phpunit_bridge.html

<https://symfony.com/blog/introducing-symfony-panther-a-browser-testing-and-web-scraping-library-for-php>

Ayant, par le passé, déjà utilisé des frameworks de tests pour PHP mais aussi JavaScript (Jest, Jasmine et Mocha), j'ai automatiquement ciblé les mots clés précédemment cités. L'usage de l'anglais technique est primordial, à mon sens, afin d'obtenir des résultats de recherches à jour et pertinents.

Conclusion

Le stage effectué chez Iole Solutions, m'a permis de découvrir le monde d'une entreprise de développement informatique, tout en expérimentant le télétravail. En parallèle, la réalisation de mon projet personnel sous forme de portfolio, m'a permis de mettre en pratique les compétences acquises au cours de ma formation et d'élargir ces compétences par des recherches personnelles par le biais de veille technologique et de lecture de documentations techniques.

Mon projet, à l'issue de la formation, est de trouver un emploi dans une entreprise en tant que développeur web. Le fait d'avoir pu effectuer du télétravail m'a permis de prendre conscience qu'il était possible de travailler pour une entreprise même si celle-ci se trouve éloignée de mon lieu de vie, sans pour autant affecter mon efficience, ma curiosité et ma passion pour ce domaine.

Tableau récapitulatif des points de la certification

Compétences		Stage	Projet personnel
Activité type 1	Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité	Maquetter une application	Non
		Développer des composants d'accès aux données	Non
		Développer la partie front-end d'une interface utilisateur web	Oui
		Développer la partie back-end d'une interface utilisateur web	Oui
Activité type 2	Concevoir et développer la persistance des données en intégrant les recommandations de sécurité	Mettre en place une base de données	Non
Activité type 3	Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité	Concevoir une application	Oui
		Développer des composants métier	Oui
		Construire une application organisée en couches	Non

Annexes

Annexe 1 : Fichier « **loadTwig.php** » permettant le rendu PDF d'informations contenues dans des tableaux à plusieurs dimensions. (version en ligne disponible sur mon dépôt GitHub <https://github.com/Slashflex/CDA-rapport/blob/master/annexes/Extrait de code Stage - Iole/loadTwig.php>).

```
<?php

use Twig\Environment;
use Twig\Extension\DebugExtension;
use Twig\Loader\FilesystemLoader;

require_once __DIR__ . "/PDF/L3_clsPrint.php";
require_once './arrayFinal.php';
/**
 * This file allow to use the Twig template engine.
 *
 * To be able to use Twig as a standalone (without composer)
 * - Download Twig from https://github.com/twigphp/Twig/archive/2.x.zip

```

```

* - Extract it to the relative path you want (in my case /Twig)
* - Rename Twig/src to Twig/Twig
* - include the spl_autoload_register() function below
* - Do not forget to use Fully Qualified Class Name (FQCN) to reference Twig's classes
*/
spl_autoload_register(function ($classname) {
    $dirs = [
        './Twig/' // ./path/to/dir_where_src_renamed_to_Twig_is_in
    ];
    foreach ($dirs as $dir) {
        $filename = $dir . str_replace('\\', '/', $classname) . '.php';
        if (file_exists($filename)) {
            require_once $filename;
            break;
        }
    }
});

$T3 = new L3_clsPrint();

/**
 * This creates a template environment with a default configuration and a loader
 * that looks up templates in the /PDF/templates directory. Different loaders are
 * available and you can also write your own if you want to load templates from
 * a database or other resources.
 */
$loader = new FilesystemLoader('./PDF/templates');
$twig = new Environment($loader, [
    'debug' => true
]);

// Adds Twig's ability to debug template(s) e.g use dump instead of var_dump inside templates
$twig->addExtension(new DebugExtension());
// QrCode in invoice's header
// $image = new barcode_generator();
// $image->output_image('png', 'qr', 'www.google.fr', '');

$priceTotalAllTab = 0;
$previousReduction = null;
$priceTotal = 0;
$reductionKey = null;
$quantityKey = null;
$priceKey = null;
$montantKey = null;
$totalHT = 0;
$idArray = [];
$newId = 0;
$creationInProgress = false;

for ($i = 0; $i < count($invoice); $i++) {
    $returnElement = " <br/>";
    $newId++;
    $idArray[$i] = $newId;
    $returnElement .= '<span id="head_current"' . $newId . '></span><br/><table class="tableau"><thead><tr>';
    $returnElement .= '<span class="bonDeCommande bold"> Bon de commande : ' . $invoice["BonDeCommande"][$i] . '</span>';
    foreach ($invoice["ColumnName"] as $keyHead => $valHead) {
        if ($valHead == "Remise") $reductionKey = $keyHead;
        if ($valHead == "Prix Unitaire") $priceKey = $keyHead;
        if ($valHead == "Montant") $montantKey = $keyHead;
        if ($valHead == "Quantité") $quantityKey = $keyHead;
        $returnElement .= " <th> " . $valHead . " </th>";
    }
    $returnElement .= '</tr></thead><tbody>';
    for ($a = 0; $a < count($dataPourTotaux[0]["Value"]); $a++) {
        $returnElement .= '<tr>';
        $creationInProgress = true;
        foreach ($dataPourTotaux[0]["Value"][$a] as $keyBody => $valBody) {
            if ($keyBody == $reductionKey) {
                if ($valBody == null) {
                    $previousReduction = null;
                    $remise = "";
                    $returnElement .= ' <td class="center"> ' . $remise . ' </td>';
                } else {

```

```

$previousReduction = $valBody;
$remise = strval($valBody) . " %";
$returnElement .= ' <td class="center"> ' . $remise . ' </td>';
}
} else if ($keyBody == $priceKey) {
$price = $valBody;
$priceFormat = number_format((($price), 2, ',', '') . $l3->devise;
$returnElement .= ' <td class="center"> ' . $priceFormat . ' </td>';
} else if ($keyBody == $quantityKey) {
$quantity = $valBody;
$returnElement .= ' <td class="center"> ' . $quantity . ' </td>';
} else if ($keyBody == $montantKey) {
$thePrice = $price * $quantity;
if ($previousReduction == null) {
$priceTotal += $thePrice;
$thePriceFormat = number_format((($thePrice), 2, ',', '') . $l3->devise;
$returnElement .= ' <td class="center"> ' . $thePriceFormat . ' </td>';
$thePrice = null;
} else {
$reduction = $thePrice * $previousReduction / 100;
$thePrice -= $reduction;
$priceTotal += $thePrice;
$reduction = null;
$thePriceFormat = number_format((($thePrice), 2, ',', '') . $l3->devise;
$returnElement .= ' <td class="center"> ' . $thePriceFormat . ' </td>';
$thePrice = null;
}
$previousReduction = null;
} else {
$returnElement .= ' <td class="center"> ' . $valBody . ' </td>';
}
}
$creationInProgress = false;
$returnElement .= '</tr>';
}
$priceTotalFormat = number_format((($priceTotal), 2, ',', ''));
$returnElement .= '</tbody></table>';
$returnElement .= '<div class="priceUnderTab"><span class="semibold bold">Sous-total</span> <span id=sousTotaux>' . $priceTotalFormat . '€ </span> </div>';
$totalHT += $priceTotal;
array_push($dataPourTotaux, $totalHT);
$priceTotal = null;
$priceTotalAllTab += $priceTotal;
}

// Render the template /PDF/templates/test.html with some data , call the render() method:
echo $twig->render('test.html.twig', [
'invoice' => $invoice, // array line 49
'footer' => $footer,
'adresse' => $adresselfole,
'contact1' => $contactInterlocuteur1,
'contact2' => $contactInterlocuteur2,
'facture' => $facture,
// 'qrCode' => $image->getName(),
'client' => $client,
'total' => $l3->subtotals($dataPourTotaux)
]);

```

Annexe 2 : Fichier « **test.html.twig** », recevant et traitant les données reçues par le fichier « **loadTwig.php** » (version en ligne disponible sur mon dépôt GitHub <https://github.com/Slashflex/CDA-rapport/blob/master/annexes/Extrait de code Stage - Iole/PDF/templates/test.html.twig>).

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<script src="js/pdf.js"></script>
<link type="text/css" rel="stylesheet" href="css/pdf.css" />

```

```

<title>Facture</title>
</head>

<body onload="getPdfInfo()">
<header>
  <div class="headFac">
    <div class="w-22-5">
      <div class="flex">
        
        <br />
        <br />
      </div>
      <div class="flex">
        {{adresse.adresse1}}<br />
        {{adresse.adresse2}}<br />
        {{adresse.codePostal}} {{adresse.ville}}<br />
        Tél : {{adresse.tel}}<br />
      </div>
    </div>
  </div>

  <div class="firstLine"></div>

  <div class="facDetail w-22-5">
    <div class="headFacTitle bold center">Détails facture :</div>
    <div class="infoFacture flex">
      N° de facture : {{facture.num}}<br />
      Date d'émission : {{facture.dateEmission}}<br />
      Date d'échéance : {{facture.dateEcheance}}<br />
      Référence client : {{facture.referenceClient}}
    </div>
  </div>

  <div class="secondLine"></div>

  <div class="ioleDetail w-22-5">
    <div class="headFacTitle bold flex">Contact interlocuteur 1 :</div>
    <div class="flex">
      {{contact1.nom}}<br />
      Tél : {{contact1.tel}}<br />
      Mail : {{contact1.mail}}
    </div>
    <div class="headFacTitle mt-2 bold flex">Contact interlocuteur 2 :</div>
    <div class="flex">
      {{contact2.nom}}<br />
      Tél : {{contact2.tel}}<br />
      Mail : {{contact2.mail}}
    </div>
  </div>

  <div class="thirdLine"></div>

  <div class="barCodeHeader">
    
  </div>
</div>

<div class="clientContact">
  {{client.Nom}} {{client.Prenom}} <br />
  {{client.Adresse1}} <br />
  {{client.Adresse2}}<br />{{client.CodePostal}} {{client.Ville}}
</div>

<div class="title">
  Facture
</div>
</header>

```

```

{%
    for facture in invoice.BonDeCommande %
}
<span id="head_current8"></span>
<table class="tableau">
    <span class="bonDeCommande bold">Bon de commande : {{facture}}</span>
    <thead>
        <tr>
            {% for column in invoice.ColumnName %}
                <th>{{column}}</th>
            {% endfor %}
        </tr>
    </thead>
    <tbody>
        {% for key, val in invoice.Value %}
            <tr>
                <td class="center" {% if key % 2 == 0 %}grey {% endif %}>{{val.0}}</td>
                <td class="center" description {% if key % 2 == 0 %}grey {% endif %}>{{val.1}}</td>
                <td class="center" {% if key % 2 == 0 %}grey {% endif %}>{{val.2}}</td>
                <td class="center" {% if key % 2 == 0 %}grey {% endif %}>{{val.3}}</td>
                <td class="center" {% if key % 2 == 0 %}grey {% endif %}>{{val.4}}</td>
                <td class="center" {% if key % 2 == 0 %}grey {% endif %}>{{val.5}}</td>
                {% set montant = val.3 * val.2 %} {% set discounted = montant * val.5 / 100 %}
                {% set result = montant - discounted %}
                <td class="center" {% if key % 2 == 0 %}grey {% endif %}>{{montant}}</td>
            </tr>
        {% endfor %}
    </tbody>
</table>
{% set addition = invoice.Value[0][5] %}
{% set discounted1 = invoice.Value[0][3] * invoice.Value[0][2] * invoice.Value[0][5] / 100 %}
{% set discounted2 = invoice.Value[1][3] * invoice.Value[1][2] * invoice.Value[1][5] / 100 %} {% set discounted3 = invoice.Value[2][3] * invoice.Value[2][2] *
    invoice.Value[2][5] / 100 %} {% set final = discounted1 + discounted2 + discounted3 %}
<div class="priceUnderTab">
    <span class="bold sousTotal">Sous-total</span>
    <span id="sousTotaux">{{final}} € </span>
</div>
{% endfor %}

<div class="flex mb-50">
    <!-- Informations RIB -->
    <div id="rib" class="rib">
        <p>
            <span class="bold">Conditions et modalités de règlement : </span>
        </p>
        <p>
            <span class="bold">Règlement par virement :</span><br />
            IBAN FR76 4590 4857 0301 9867 045 (Ajouter Libellé : FAC2020-IOLE)
        </p>
        <p>
            <span class="bold">Règlement par chèque à l'ordre de :</span><br />
            Iole Solutions
        </p>
    </div>
</div>

<p>Paiement sous 30 jours à réception de la facture. Pas d'escompte pour paiement anticipé. En cas de retard de paiement, application d'une indemnité forfaitaire pour frais de recouvrement de 40€, selon l'article D.441-5 du code du commerce.</p>
</div>

<!-- Tableau Totaux -->
<div class="totaux">
    {{total|raw}}
</div>
</div>
</div>

```

```
<footer>
  <div id="statuts">
    {{footer.nom}} - {{footer.forme}} - Capital social : {{footer.capital}} - Siège social :
    {{footer.siege}}
    -<br />
    RCS : {{footer.rcs}} - SIRET : {{footer.siret}} - Numéro de TVA intracommunautaire :
    {{footer.tvalntra}} - Code
    APE : {{footer.ape}}
    <br />
    <br />
    <div id="page_numbers">
      Page
      <span id="page_current"></span>
      <span>/</span>
      <span id="page_count"></span>
    </div>
  </div>
</body>
</html>
```