

## Directeur des Projets Informatiques

# Dossier de validation

Nom Prénom	Vennat Simon
Nom Prénom du tuteur	Chouchane Issam
Acronyme de la certification IPI visée	CPI
Niveau visé	7
Date de la soutenance	14 octobre 2022
Lieu de la soutenance	Lille



UNE ÉCOLE DU GROUPE IGS



## Table des matières

<b>Remerciements</b>	<b>4</b>
<b>Introduction</b>	<b>5</b>
<b>1. Présentation de l'entreprise</b>	<b>6</b>
1.1 Présentation générale	6
1.2 Environnement de travail	7
<b>2. Présentation du projet</b>	<b>10</b>
<b>3. Analyse métier du projet</b>	<b>12</b>
<b>3.1 Gérer les processus et la qualité</b>	<b>12</b>
3.1.1 Analyser des processus existants	12
3.1.2 Optimiser les processus existants	13
3.1.3 Technologies et outils utilisés	19
<b>3.2 Manager un projet</b>	<b>22</b>
3.2.1 Utiliser une méthode agile	22
3.2.2 Découper le projet en tâches et évaluer la charge de travail	25
3.2.3 Contrôler la réalisation des livrables	26
<b>3.3 Gérer les ressources du projet</b>	<b>29</b>
3.3.1 Gérer une équipe	29
3.3.2 Maîtriser le budget	31
<b>3.4 Gérer les risques du projet</b>	<b>36</b>
3.4.1 Identifier les risques du projet	36
3.4.2 Évaluer et hiérarchiser les risques	38
3.4.3 Définir les actions pour minimiser les risques	41
<b>3.5 Communiquer</b>	<b>45</b>
3.5.1 Répondre à un besoin	45
3.5.2 Mobiliser l'équipe pour favoriser l'avancement du projet	47
<b>Conclusion</b>	<b>53</b>
<b>Annexes</b>	<b>55</b>

## Remerciements

Je tiens tout d'abord à remercier l'ensemble des membres de l'entreprise Allstore pour leur participation à ma formation, pour l'accueil qu'ils m'ont réservé ainsi que pour l'ambiance de travail et leur bonne humeur au quotidien.

Plus particulièrement, je souhaite remercier mon tuteur, Monsieur Issam Chouchane, pour m'avoir fait confiance. Il a su m'accompagner de la meilleure des manières afin de me transmettre son expertise dans le développement informatique mais aussi dans la chefferie de projet et dans l'architecture logicielle.

Aussi, je tiens à remercier Thomas Bo, responsable digital, pour les précieux conseils et l'expertise qu'il a su m'apporter tout au long de ma formation.

Pour finir, je tiens à remercier l'école Aston pour m'avoir permis d'enrichir mes compétences et de m'avoir formé pour le métier de chef de projet spécialisé dans l'architecture logicielle.

Je pense également à mes proches qui m'ont sans cesse soutenu, tant personnellement que dans l'élaboration de mon projet professionnel.

Cette année d'apprentissage m'a permis d'enrichir mes compétences en chefferie de projet ainsi qu'en architecture logicielle et de m'épanouir au sein d'une entreprise, Allstore. En effet, le fait d'évoluer dans une telle entreprise m'a permis de maîtriser les notions de gestion de projet et d'architecture logicielle, notamment en mettant en pratique ce que j'ai appris lors de la formation.

## Introduction

Après avoir obtenu mon DUT Informatique, j'ai souhaité enrichir mes compétences avec l'alternance. J'ai ainsi intégré une licence professionnelle Développement Informatique et Outils Collaboratifs en alternance avec la société Chausport – Spodis. J'ai pu découvrir la vie en entreprise et travailler sur mon premier projet d'entreprise, me permettant d'obtenir ma première expérience professionnelle en tant que développeur. Cependant, en conséquence de la crise sanitaire, j'ai été contraint de réaliser la quasi-totalité de mon année en télétravail.

Par conséquent, j'ai décidé d'intégrer, au cours de l'année 2021-2022, la formation d'Aston Directeur de Projets Informatiques – Architecte Logicielle en alternance afin d'en apprendre plus sur la direction de projets informatiques, ainsi que sur l'architecture logicielle. En effet, toujours convaincu que l'alternance était le parcours qui me correspondait le mieux, j'ai souhaité rejoindre Aston afin d'acquérir une nouvelle expérience professionnelle, cette fois-ci en présentiel, tout en suivant une formation auprès de l'école.

J'ai fait le choix d'intégrer cette formation car je souhaitais continuer à enrichir mes compétences techniques avec l'architecture logicielle. Par ailleurs, il me semble également important d'acquérir des compétences en gestion de projet. En effet, bien que je ne souhaite, pour le moment, pas devenir chef de projet, je n'exclus pas cette possibilité sur le long terme. Ainsi, il était important pour moi de maîtriser les fondamentaux de ce métier.

En intégrant la société Allstore, j'ai pu développer mes compétences d'architecture logicielle, de gestion de projet et, d'un point de vue plus technique, de développeur web dans le cadre d'un projet intitulé Zoranne. J'ai également pu vivre une année supplémentaire en entreprise, sans les contraintes de télétravail imposées par la crise sanitaire l'année précédente.

# 1. Présentation de l'entreprise

## 1.1 Présentation générale

Happy Garden est une marque exploitée par la Société par Actions Simplifiée (SAS) Allstore. Le siège de l'entreprise se trouve à Wambrechies. L'entreprise se situe donc au sein de la Métropole Européenne de Lille.

Happy Garden est un site e-commerce spécialisé dans la vente de mobilier d'extérieur et d'intérieur. Le site est actuellement disponible en France et, depuis peu, en Espagne. Aussi, il est prévu d'ouvrir le site à d'autres pays dans le futur.

Par ailleurs, les produits Happy Garden sont aussi en vente sur la plupart des places de marché (18 en ce moment) telles que Amazon, But, Cdiscount, ManoMano, Maisons du monde, Conforama, Leroy Merlin, Carrefour etc. Ces places de marché représentent une grande part de notre chiffre d'affaires.

En outre, Happy Garden vise à se diversifier en proposant un plus large choix de produits.

Allstore est une entreprise à taille humaine qui a fêté ses 10 ans au mois de mars 2022. En effet, elle est, à l'heure actuelle, constituée de 21 collaborateurs.

Le confinement a conduit les gens à consommer beaucoup plus de mobilier, que ce soit d'intérieur ou d'extérieur. Ainsi, le chiffre d'affaires de l'entreprise a considérablement augmenté à la suite de cette période.

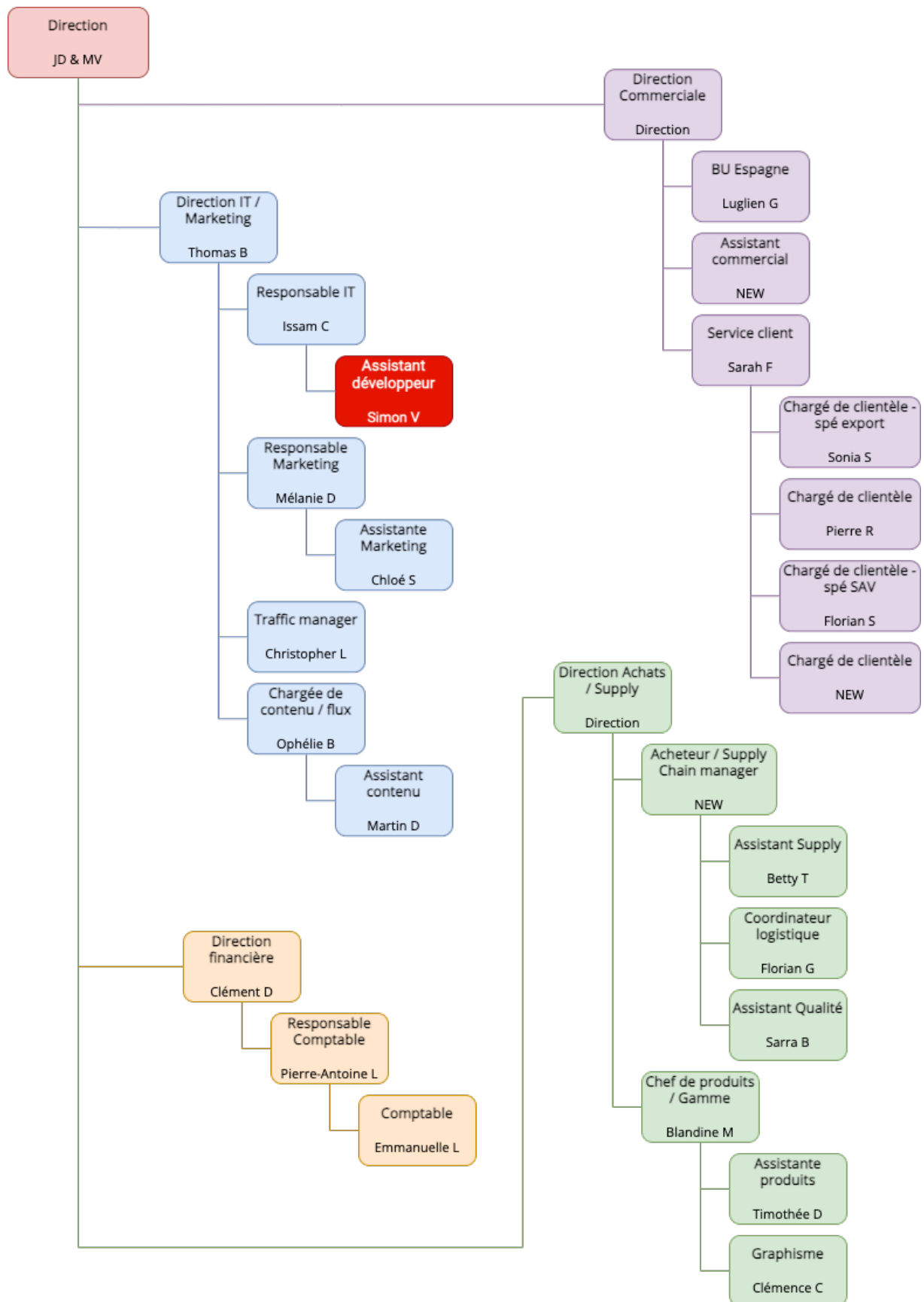
Par conséquent, cela a permis la naissance de nouveaux projets ainsi que le recrutement de nouveaux collaborateurs pour faire grandir et développer ces projets. En effet, c'est ainsi que le projet Zoranne est né et que le site a pu s'élargir à l'Espagne et bientôt à d'autres pays, comme notamment l'Allemagne.

## 1.2 Environnement de travail

L'ensemble des membres de l'entreprise est, depuis notre emménagement dans nos nouveaux locaux, présent dans une seule pièce : un open space.

L'entreprise est divisée en plusieurs pôles, équipes. En effet, l'open space est composé du service web, achat, comptabilité, logistique, service client et SAV.

Pour ma part, je fais partie du service IT (en bleu dans l'organigramme ci-dessous). Plus précisément, l'équipe dont je fais partie au sein du service web compte trois membres : Thomas : le responsable digital, Issam : mon tuteur et le responsable IT et moi-même, développeur.



Organigramme Allstore



Comme on peut le voir dans l'organigramme ci-dessus, j'ai pour ma part intégré l'entreprise en tant qu'assistant développeur (rectangle rouge dans l'organigramme ci-dessus). En ce sens, j'ai pu découvrir le métier de chef de projet informatique en gérant un projet complexe. Aussi, j'ai eu l'occasion de participer aux décisions du projet, notamment celles portant sur l'architecture de celui-ci. Par ailleurs, j'ai réalisé la quasi-totalité des développements concernant le projet sur lequel j'ai travaillé : Zoranne.

En effet, peu de temps après mon arrivée, mon équipe (Thomas et Issam) m'a présenté le projet Zoranne, sur lequel j'ai travaillé tout au long de l'année (et sur lequel je travaille encore en ce moment). À ce titre, je m'occupe de la quasi-totalité des développements (partie technique) de notre projet principal, Zoranne. De plus, étant donné que je suis arrivé avant que le projet Zoranne ait débuté, j'ai eu l'occasion de prendre des décisions et de participer à celles-ci concernant l'architecture du projet et son management. De même, je suis encore aujourd'hui amené à prendre les meilleures décisions possibles concernant le projet, avec l'aide et la validation de mon équipe.

Par ailleurs, concernant l'activité quotidienne des membres de mon service, mon tuteur Issam s'occupe principalement de la partie Ops du projet Zoranne. Aussi, il s'occupe des tickets concernant le site Happy Garden sous PrestaShop, que ce soit pour la France ou l'Espagne. Il est également en charge de la gestion du matériel informatique et de l'infrastructure de l'entreprise, avec notamment la connexion internet, etc.

Thomas, en tant que responsable digital a d'autres missions. Il apporte son expertise aux projets informatiques, notamment concernant les processus et les règles métier en place au sein de l'entreprise. Il dirige également les autres services web, tels que le service marketing, contenu ou encore projet. Il permet aussi d'ouvrir nos produits à de nouvelles places de marché et à de nouveaux pays. De plus, il met en place de nouvelles solutions, qu'elles soient internes (Zendesk) ou qu'elles concernent le site Happy Garden (nouvelles solutions de paiement).

## 2. Présentation du projet

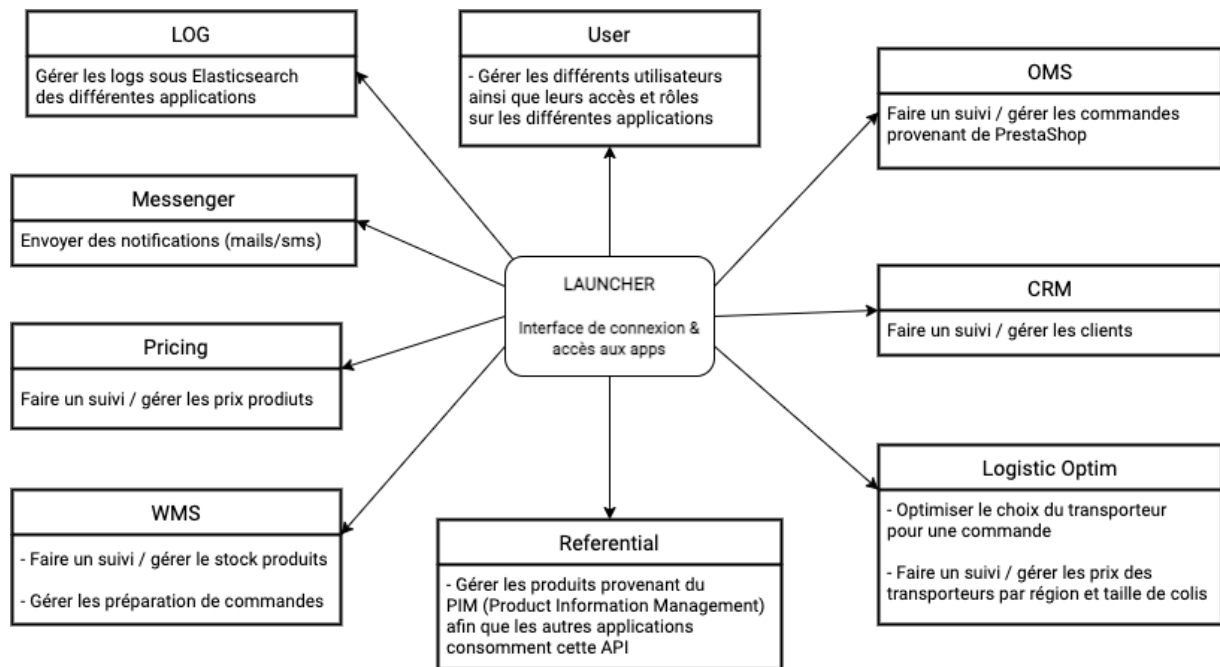
Au sein de mon année chez Allstore, j'ai eu l'occasion de travailler sur un projet intitulé Zoranne. Ce projet m'a été présenté dès mon arrivée dans l'entreprise, en juillet 2021. En effet, initialement, le site web Happy Garden était une application monolithique, c'est-à-dire d'un seul bloc, fonctionnant sur PrestaShop version 1.5.

Toutes les fonctionnalités, comme notamment le choix des transporteurs pour une commande ou encore la synchronisation du stock depuis notre ERP étaient donc présentes dans des modules PrestaShop.

En s'inspirant des architectures SOA et microservices, Zoranne vise à apporter plus de flexibilité et de maintenabilité sur les différents processus du site Happy Garden, l'entreprise évoluant de plus en plus suite au confinement.

Dans un premier temps, l'objectif du projet est donc de migrer les fonctionnalités déjà en place sur PrestaShop, mais au sein de nouvelles applications distinctes, tout en apportant une interface aux équipes métier.

Comme le montre le schéma ci-dessous, Zoranne est constitué d'une interface de connexion permettant d'accéder à un launcher. Celui-ci permet ensuite d'accéder aux différentes applications (présentes sur le schéma ci-dessous) auxquelles l'utilisateur a accès.



*Schéma des applications Zoranne, v0*

Ainsi, les équipes métier auront accès aux applications qui leur sont propres, et pourront effectuer leurs tâches dans une interface pensée spécifiquement pour leurs besoins, au lieu de, notamment, naviguer sur les différents onglets du back-office de PrestaShop.

Pour résumer, la première version de Zoranne propose les mêmes fonctionnalités déjà en place sur PrestaShop mais sur des applications distinctes, puis, les versions suivantes permettront de les étoffer et de les enrichir.

Par ailleurs, la mise en place de Zoranne va précéder la refonte graphique du site Happy Garden (réalisé par un tiers) avec la migration vers la version 1.7 de PrestaShop (1.5 vers 1.7).

## 3. Analyse métier du projet

### 3.1 Gérer les processus et la qualité

#### 3.1.1 Analyser des processus existants

L'objectif de la première version du projet Zoranne est d'effectuer la migration d'une application monolithique déjà fonctionnelle vers une architecture basée sur les microservices et adaptée à nos besoins. Afin de pouvoir effectuer cette migration de la meilleure des manières, j'ai dû réaliser une analyse de l'existant, c'est-à-dire de l'ensemble des fonctionnalités et processus initialement en place.

Pour cela, j'ai manipulé et parcouru le code PrestaShop ainsi que le back office. J'ai ainsi pu découvrir les diverses fonctionnalités utilisées et comprendre la plupart des processus mis en place.

Aussi, j'ai reçu un certain nombre d'explications de mes collègues, et plus particulièrement de Thomas. Ils ont pu m'apporter toutes les informations dont j'avais besoin afin de comprendre les processus existants.

Avant la mise en place du projet Zoranne, l'ensemble des processus concernant le site Happy Garden ou encore les différentes places de marché, étaient développés et exploités au sein de notre CMS : PrestaShop (version 1.5). Ainsi, les équipes métier travaillent toutes sur la même application, le back-office PrestaShop, afin de réaliser leurs tâches respectives.

Par ailleurs, l'ERP utilisé au sein de l'entreprise est Business Central (BC) de Microsoft. Les différents processus, comme par exemple le passage d'une commande, interagissent avec ce dernier.

Concernant le Product Information Management (PIM) exploité par Allstore, il s'agit de Quable. Effectivement, Quable nous permet de gérer l'ensemble de nos fiches produits. Ainsi, c'est sur le PIM que se base la diffusion de nos produits, que ce soit sur le site web Happy Garden ou encore sur les places de marché.

Auparavant, cette architecture monolithique était adaptée. Cependant, ces dernières années, Allstore connaît une croissance importante. Par conséquent, les projets deviennent, eux aussi, de plus en plus conséquents, le tout dans la même application. Il a donc été nécessaire d'optimiser l'architecture, et donc les processus existants.

Par ailleurs, une telle migration a un impact conséquent sur les processus existants. En effet, une fois que Zoranne sera en production, les équipes métier ne devrons plus travailler sur le backoffice de PrestaShop mais sur les différentes applications constituant Zoranne. Ainsi, la mise en place de Zoranne entrainera une modification importante des processus, notamment pour les équipes. Il sera donc nécessaire de les accompagner aux changements qu'implique la modification des processus.

Par conséquent, en apprenant à analyser l'existant, j'ai acquis de l'expérience. Ainsi, je pourrai mettre en œuvre cette expérience dans le futur. Effectivement, lorsque je serai de nouveau confronté à ce genre de projet, où la compréhension de l'existant est essentielle, je serai plus apte et efficace pour assimiler les différents processus.

### 3.1.2 Optimiser les processus existants

Le projet Zoranne a pour objectif d'optimiser les processus déjà mis en place, notamment afin d'accroître la flexibilité et la maintenabilité. En effectuant cette migration pour la réalisation de la première version du projet, j'ai aussi eu l'occasion d'améliorer les performances des différents processus.

En plus de gérer l'ensemble des processus (tous gérés par PrestaShop auparavant), Zoranne a également pour vocation de proposer une solution interne aux équipes métier (aussi au sein de PrestaShop auparavant).

En effet, Zoranne doit inclure différentes applications, découpées par thème, comme par exemple WMS, OMS, Pricing, etc. Ainsi, les fonctionnalités utilisées par les équipes métier et présentes dans le back-office de PrestaShop doivent être intégrées dans ces applications de Zoranne.

Par ailleurs, l'accès à Zoranne se fait de manière sécurisée. Tout d'abord, seules les adresses IP autorisées peuvent accéder aux différentes applications.

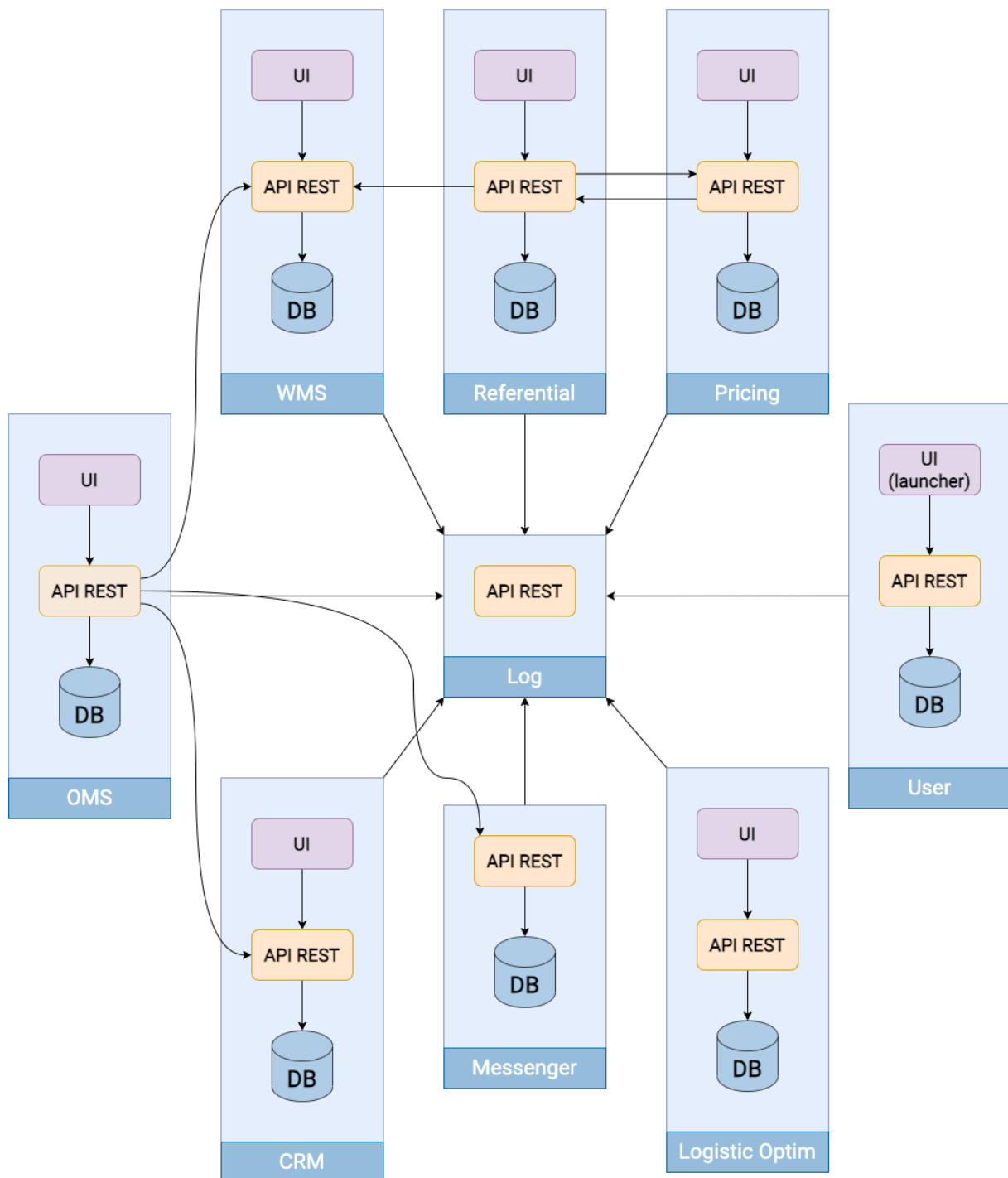
De plus, chaque membre de l'entreprise dispose d'accès et de droits différents (Viewer, Editor, Administrator). Par exemple, une personne du service client peut avoir accès à l'application OMS avec le rôle *Editor*, à l'application Pricing avec le rôle *Viewer*, et ne pas avoir accès à l'application WMS.

L'accès aux différentes applications se fait à partir d'un token, indispensable pour les requêtes back-end, généré lors de l'authentification sur le launcher.

En outre, la migration de Prestashop vers Zoranne a nécessité d'importer certaines données au sein des nouvelles bases de données Zoranne. Par exemple, pour l'application Pricing, je me suis occupé de créer un script permettant d'initialiser la nouvelle base de données Zoranne à partir de celle utilisée avec PrestaShop.

De même, j'ai effectué ce processus afin de récupérer l'ensemble des commandes PrestaShop dans Zoranne. Ainsi, les équipes métier telles que le service client pourront bénéficier de l'historique de l'ensemble des commandes.

L'architecture que j'ai mis en place s'inspire d'une architecture en microservices et d'une architecture SOA. En effet, une application (API REST) est développée par thème (WMS, OMS, Pricing, etc.) et chaque API possède sa propre base de données.



*Schéma montrant les interactions entre les différentes applications (appels HTTP) de Zoranne*

Le schéma ci-dessus met en évidence l'architecture du projet Zoranne. On peut ainsi remarquer que le projet est divisé en plusieurs applications distinctes, notamment composées d'une API REST développée avec Symfony.

De plus, on peut constater que chacune d'entre elles possède sa propre base de données (« DB » dans le schéma ci-dessus).

En outre, on peut voir les applications possédant une interface utilisateur développée avec Angular (« UI » dans le schéma ci-dessus) avec laquelle les équipes métier peuvent interagir. Ces interfaces utilisateurs ont été pensées afin de correspondre au mieux aux besoins des équipes métier. À partir des informations récoltées auprès d'elles, j'ai pu réaliser des maquettes me permettant d'effectuer, de la manière la plus ergonomique possible, les développements des différentes applications (cf. capture d'écran en annexe). De surcroît, afin que les utilisateurs bénéficient d'une certaine uniformité, j'ai pu choisir un admin template (thème d'administration) que nous avons acheté. Fuse, le thème que j'ai choisi, nous permet de proposer un design agréable et commun avec toutes les applications afin de proposer la meilleure expérience aux utilisateurs.

Les équipes métier utiliseront donc chacun principalement une application, et selon leur rôle, ils pourront accéder à d'autres applications.

Aussi, ce schéma met en évidence l'ensemble des interactions, via des appels HTTP, entre les applications qui composent Zoranne.

Je trouve que cette architecture est adaptée aux besoins car elle permet d'accroître la flexibilité, tout en assurant la maintenabilité sur le long terme.

Par ailleurs, cette architecture propose une scalabilité importante. Effectivement, dans un premier temps, Zoranne doit être « iso-fonctionnel » à ce qui est déjà en place sur PrestaShop. C'est-à-dire que, globalement, il doit proposer les mêmes fonctionnalités mais avec une architecture et des technologies différentes. Puis, une fois la première version du projet mise en place, l'objectif est d'enrichir chaque application, en fonction du besoin, afin de simplifier, d'améliorer et d'automatiser les processus côté back-end mais aussi sur les back-offices pour les équipes métier.



Aussi, un tel choix facilite le déploiement continu du code. En effet, chaque service (application) peut être déployé indépendamment des autres, et de manière continue et automatisée.

Cette architecture permet également d'être exploitée par d'autres entreprises qu'Allstore. Par exemple, Garden Max, la société sœur d'Allstore pourrait exploiter l'application WMS et, notamment, la synchronisation des stocks.

De surcroît, l'architecture choisie permet de limiter les risques liés aux défaillances serveurs. Effectivement, le projet étant découpé en plusieurs applications, si une défaillance serveur survient, elle sera isolée sur une application.

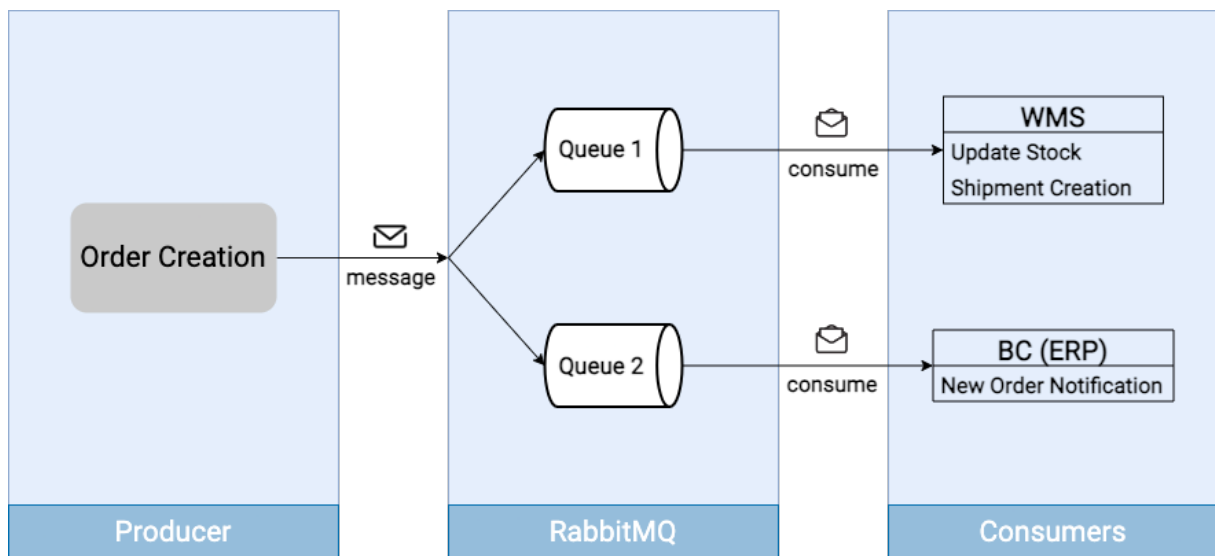
Une architecture en microservices aurait également pu convenir pour répondre au besoin en proposant une granularité plus fine dans le découpage. En effet, chaque application aurait pu elle-même être découpée en plusieurs microservices répondant chacun à un objectif spécifique.

Cependant, une architecture strictement en microservices demande plus de ressources, notamment en termes de temps, et n'était pas forcément pertinente dans ce contexte. En effet, une API REST et son application back-office sont parfaitement cohérentes et proposent un découpage suffisant. Ainsi, d'après moi cette architecture hybride est un bon compromis et constitue une solution adaptée au besoin.

Par ailleurs, la migration de notre application monolithique PrestaShop vers notre nouvelle architecture aurait pu entraîner des pertes de performances. En effet, avec notre architecture, les applications doivent communiquer entre elles. Par conséquent, avec l'aide de mon tuteur Issam, j'ai appris à mettre en place une communication asynchrone afin que les clients de notre site ne soient pas impactés par les éventuels ralentissements.

Pour cela, nous avons choisi RabbitMQ qui nous permet de gérer des messages en file d'attente. Ayant découvert les systèmes de gestion de message avec Zoranne, je considère que RabbitMQ est adapté à notre besoin car il nous permet de réaliser nos tâches de manière asynchrone, tout en étant plutôt simple à configurer.

Pour aller plus loin, j'ai installé le composant Messenger de Symfony permettant d'envoyer des messages à notre RabbitMQ.



*Schéma montrant l'envoi de message avec RabbitMQ lors de la création d'une commande*

Par exemple, comme le montre le schéma ci-dessus, lors de la création d'une commande sur notre OMS, des tâches volumineuses doivent être effectuées vers des applications externes, comme notamment la décrémentation de notre stock et la mise en préparation de la commande dans WMS, ou encore la notification de notre ERP Business Central. Pour que cela soit invisible pour le client, j'envoie simplement un message constitué d'une entité et d'un handler (action à réaliser) à notre serveur RabbitMQ pour chaque tâche à effectuer.

Par conséquent, la requête de création de commande se termine une fois la commande créée dans l'OMS et les messages ajoutés à la file d'attente (c'est-à-dire les « Queues » dans le schéma ci-dessus) de RabbitMQ. Ces messages sont ensuite consommés de manière asynchrone, sans surcharger nos serveurs. Ainsi, les tâches complexes sont réalisées rapidement et sont invisibles par les utilisateurs, qui bénéficient d'une expérience fluide.

### 3.1.3 Technologies et outils utilisés

J'ai eu la possibilité de participer aux choix des frameworks utilisés pour le développement du projet Zoranne.

J'ai fait le choix d'utiliser Symfony (langage PHP) pour les API REST (back-end) et Angular (langage TypeScript) pour le développement des applications du back-office. En effet, Symfony et Angular sont des frameworks que j'ai eu l'occasion de découvrir lors de ma formation précédente. J'avais beaucoup apprécié travailler avec ces derniers et je souhaitais appliquer mes connaissances dans le cadre d'un projet professionnel complet.

Aussi, Symfony et Angular sont des frameworks reconnus et possèdent une communauté importante. Par conséquent, ils sont stables et sont constamment enrichis par la communauté, qui participe également à la résolution des éventuels problèmes que l'on peut rencontrer lors des développements.

De surcroît, en choisissant le framework Symfony qui utilise le même langage de programmation que l'application existante (sous PrestaShop), on facilite le recrutement des éventuels futurs développeurs qui intégreront l'entreprise, car seul un langage est utilisé pour le back-end : le PHP. Ainsi, un développeur maîtrisant le langage PHP pourra à la fois assurer la maintenabilité du site sous PrestaShop et manipuler le back-end du projet Zoranne.

En ce qui concerne l'exécution de tâches planifiées sur nos serveurs, nous utilisons Jenkins. En effet, certaines tâches de nos applications nécessitent d'être exécutées régulièrement et automatiquement, à des horaires définis. Par exemple, notre script de synchronisation des stocks à partir de notre ERP (BC) est exécuté quatre fois par jour afin de mettre à jour la base de données exploitées par les différentes entités interrogeant le stock. Je trouve que Jenkins est adapté car il permet de répondre à notre besoin d'exécuter des tâches planifiées sur nos serveurs, tout en proposant une vue d'ensemble sur les processus planifiés grâce à son interface graphique.

Pour ce qui est des logs, j'ai découvert Elasticsearch grâce à mon équipe. Elasticsearch est une base de données NoSQL. Nous utilisons cette base de données afin de stocker les logs de nos applications, c'est-à-dire d'historiser les différents événements exécutés sur le serveur. Je considère qu'Elasticsearch est l'outil correspondant au mieux à la gestion des logs. En effet, celui-ci peut supporter un très gros volume de données ainsi que des montées en charges, ce qui est nécessaire dans le stockage de logs. De plus, une fois connecté au cluster, il est possible de visualiser l'ensemble des données stockées ainsi que de faire des recherches très performantes.

D'autre part, dans le cas où Elasticsearch ne serait plus opérationnel (serveur inaccessible par exemple), j'ai mis en place la possibilité d'enregistrer les logs au sein d'un fichier stocké sur le serveur.

Par ailleurs, j'ai eu l'occasion de travailler avec des outils collaboratifs. Effectivement, j'ai notamment utilisé Slack pour communiquer avec l'ensemble des membres de l'entreprise.

De même, j'ai eu l'occasion d'échanger par mail avec mes collègues par le biais de la messagerie professionnelle Outlook de Microsoft.

De plus, j'ai utilisé Jira afin de gérer, attribuer et planifier l'ensemble des tâches à réaliser pour nos différents projets.

Aussi, nous utilisons GIT pour la sauvegarde et la gestion des versions des projets. De plus, nous gérons le dépôt GIT à partir du logiciel SourceTree. Les projets sont hébergés sous le service web Bitbucket, et sont composés d'un dépôt par application. Effectivement, il existe un dépôt distant pour chaque application. Chaque application est ainsi indépendante.

Pour chacune des applications, et donc des dépôts distants, l'approche CI/CD est utilisée. Pour cela, nous utilisons la solution de Atlassian : Bamboo.

Ainsi, à chaque modification poussée sur la branche de développement d'un dépôt distant, les tâches planifiées d'intégration continue et de déploiement continu sur Bamboo permettent de récupérer la dernière version du projet et de la build automatiquement sur le serveur de préproduction.

En revanche, l'approche CI/CD n'est pas en place pour les serveurs de production. En effet, les déploiements sont réalisés manuellement depuis l'interface de Bamboo une fois que les modifications apportées ont été testées sur l'environnement de pré-production. Cela évite que des modifications non souhaitées se retrouvent sur l'environnement de production.

D'après moi, Bamboo correspond parfaitement à notre besoin pour l'intégration continue et le déploiement continu. Effectivement, le fait que nous utilisons la solution Git Bitbucket pour nos différents dépôts distants, appartenant également à Atlassian, nous garantit une compatibilité et un workflow de développement optimal.

Aussi, l'interface graphique de Bamboo permet de mettre en place les tâches de déploiement automatique ou encore de réaliser des déploiements manuels. Cette interface fournit également un aperçu des différents dépôts distants créés, sur les commits poussés sur ces dépôts et sur le statut des builds et des tâches de déploiement.

Par ailleurs, nous utilisons Jira, appartenant aussi à Atlassian, pour le suivi de nos tâches à réaliser. Ainsi, ces trois applications interagissent entre elles et nous fournissent un écosystème de développement et CI/CD optimal.

## 3.2 Manager un projet

### 3.2.1 Utiliser une méthode agile

Afin de rester flexible, il m'a paru évident de mettre en place une méthode agile. En effet, j'ai appris, d'une part lors de mes formations chez Aston et d'autre part en entreprise que les méthodes classiques consistant à planifier totalement un projet avant de le développer ne sont pas suffisamment efficaces. En effet, cette méthodologie manque de flexibilité et d'adaptabilité.

La méthode agile que nous avons choisie est la méthode Kanban. Ainsi, nous procédons en flux tirés, et non sous la forme de sprints comme l'impose la méthode Scrum. Par conséquent, nous listons l'ensemble des tâches à réaliser pour répondre au besoin, puis, en limitant leur nombre, nous les déplaçons dans la colonne « En cours » du tableau de Kanban et nous les réalisons.

Par ailleurs, nous avons un tableau de Kanban pour chaque composante de Zoranne (WMS, OMS, Pricing, etc.). D'après moi, Kanban est la méthode agile la plus adaptée à notre équipe ainsi qu'à notre projet Zoranne. Par ailleurs, la méthode Kanban permet de visualiser le workflow (flux de travail) grâce au tableau de Kanban.

A FAIRE	EN COURS	A TESTER	TERMINÉ
<p>[Module] - Sync de produits dans Prestashop. Zoranne Ref API ✓ = 2d ZOR-50 <span>4C</span></p> <p>[API] - Fonction de notification de BC pour la création de la commande côté ERP Zoranne OMS API ✓ = 1d ZOR-7 <span>5V</span></p> <p>[API] - WS Async de création d'une commande Zoranne OMS API ✓ = 3d ZOR-6 <span>5V</span></p> <p>[Job] - Création du Job d'alimentation de l'API Ref depuis Quable Zoranne Jobs ✓ = 1h ZOR-37 <span>4C</span></p> <p>[Job] - Sync des prix article dans canaux de vente Zoranne Jobs ✓ = 1h ZOR-45 <span>4C</span></p> <p>[API] - WS d'alimentation des produits depuis le channel Zoranne Ref API ✓ = - ZOR-39</p>	<p>[API] - WS Async de MAJ d'une commande Zoranne OMS API ✓ = 2d ZOR-9 <span>5V</span></p>	<p>[API] - WS WMS de MAJ de Stock (+n/-n) Zoranne WMS API ✓ = 3d ZOR-52 <span>5V</span></p> <p>[API] - Export des données produits depuis les canaux de vente Zoranne Ref API ✓ = 4h ZOR-40 <span>5V</span></p> <p>[API] - Appeler la méthode de MAJ de stock et de création de la ligne expédition dans WMS Zoranne OMS API ✓ = 2h ZOR-8 <span>5V</span></p> <p>[API] - WS qui permet de cloner une commande Zoranne OMS API ✓ = 4h ZOR-22 <span>5V</span></p> <p>[Job] - Import des fiches produits depuis BC vers Quable Zoranne Jobs ✓ = 2h ZOR-36 <span>5V</span></p> <p>[API] - WS qui retourne la liste des raisons de remboursement Zoranne OMS API ✓ = 2h ZOR-18 <span>5V</span></p>	<p>[Module] - WMS connector : fonctionnalité de sync des stock Zoranne Presta 1.5 modules ✓ = - ZOR-44 <span>5V</span></p>

Tableau de Kanban Jira – Projet Zoranne

Comme nous pouvons le voir ci-dessus, nous utilisons la fonctionnalité de création de projet Kanban de Jira. Ensuite, nous pouvons personnaliser les colonnes à afficher de nos tableaux de Kanban. Puis, nous pouvons créer l'ensemble des tâches à réaliser pour le projet en question et, éventuellement, fournir une estimation du temps de réalisation pour chacune d'entre elles. Enfin, le tableau de Kanban de Jira nous permet de visualiser l'ensemble des tâches avec leur état et une synthèse de leurs informations.

Je considère que la méthode Kanban est plus adaptée que la méthode Scrum, notamment pour ce qui concerne l'effectif des membres du projet. Effectivement, il est recommandé qu'une équipe Scrum soit composée d'au minimum trois membres actifs. Or, je suis seul à travailler pleinement sur Zoranne, accompagné de mon tuteur.

Aussi, le tableau de Kanban fournit plus de flexibilité, notamment en termes de tâches. En effet, ces dernières peuvent être mises à jour, réassignées et leur hiérarchisation peut également être modifiée en fonction du besoin. De plus, la méthode Kanban nous permet d'être agile en réalisant des tâches d'une application à l'autre. Plus précisément, nous pouvons, par exemple, réaliser une tâche sur le tableau de Kanban de WMS, puis une fois terminée, effectuer une tâche sur le tableau de Kanban d'OMS.

De même, Kanban est plus flexible en ce qui concerne les échéances. L'estimation de l'effort est d'ailleurs optionnelle. Zoranne étant un projet interne sans date limite fixe, cela correspond parfaitement à notre besoin. De plus, nous souhaitons conserver cette flexibilité, notamment pour continuer de traiter les tâches extérieures à Zoranne.

En outre, Kanban est parfait pour notre approche CI/CD. Effectivement, ce dernier propose une cadence en flux continu, sans interruption, où chaque tâche réalisée apporte une valeur ajoutée. Par conséquent, les délais de livraison sont accélérés et Kanban s'inscrit parfaitement dans une approche de livraison continue.

Par ailleurs, nous effectuons des daily meetings. Durant cette réunion, nous passons en revue les tickets des tableaux de Kanban sur lesquels nous avons travaillé. Chacun fait un bilan de ce qu'il a réalisé la veille, évoque les éventuelles difficultés rencontrées, puis exprime ce qu'il va faire au cours de la journée.

C'est aussi durant cette réunion que nous revoyons la priorisation des tâches lorsque cela est nécessaire. Ainsi, je sais exactement sur quoi je peux avancer au cours de la journée, et même sur le plus long terme.

Dans le futur, lorsque l'effectif de l'équipe de développement sera plus conséquent, je n'exclus pas de faire appel au framework Scrum qui requiert un certain nombre de membres ainsi que plus de rigueur.

Travailler dans le cadre d'une méthode agile telle que Kanban m'a permis de mettre en pratique la théorie vue lors de ma formation au sein d'Aston. Aussi, j'ai pu en apprendre plus sur les subtilités de la méthode agile en l'appliquant sur un cas concret et en faisant appel à mon esprit de synthèse.

Ainsi, grâce à cette expérience, je serai plus à l'aise pour la mise en place d'une méthode agile lors de mes futurs projets.

Aussi, j'ai, à certains moments, manqué de rigueur dans l'application de la méthode agile. Par exemple, certains jours, les daily meetings n'étaient pas réalisés. Par ailleurs, comme nous sommes un petit effectif, nous avons parfois tendance, lors d'échanges, à s'attribuer des tâches sans les intégrer au tableau de Kanban, ce qui va à l'encontre de la méthode Kanban. Par conséquent, lors de mes prochains projets, je ne ferai plus cette erreur.



### 3.2.2 Découper le projet en tâches et évaluer la charge de travail

Zoranne peut être défini comme un programme. En effet, on peut identifier plusieurs projets qui le composent, comme notamment le projet WMS, OMS, Pricing, etc. Chacun de ces projets possède sa propre raison d'être et ils existent de manière indépendante. Ces projets, par leurs liens, sont réunis au sein d'un programme (Zoranne) afin d'en faciliter leur gestion.

Puis, il a fallu découper chacun de ces projets en plusieurs tâches et les intégrer au sein de nos tableaux de Kanban dans Jira. La première version de Zoranne ayant pour but de proposer les mêmes fonctionnalités que celles présentes sur PrestaShop, les tâches créées dépendent directement des fonctionnalités de PrestaShop. Mes collègues Thomas et Issam, ayant intégré l'entreprise avant moi, connaissaient mieux les processus existants. Ainsi, ils se sont principalement occupés de découper les projets en différentes tâches, tout en m'expliquant leurs décisions et en m'apprenant les fonctionnalités et processus existants.

Ensuite, j'ai appris à découper un projet en plusieurs tâches méthodiquement. Ainsi, j'ai pu, plus tardivement, m'occuper de diviser le projet en tâches afin de créer les tickets au sein des tableaux de Kanban.

Effectivement, j'ai repris, lors d'échanges avec mon équipe, les objectifs du projet en question ainsi que l'ensemble des fonctionnalités présentes sur l'existant : PrestaShop. Aussi, j'ai pu collecter des informations, notamment concernant l'ergonomie et les fonctionnalités lors d'ateliers que j'ai organisé avec les équipes métier concernés par l'application en question.

À partir de cela, j'ai redécoupé ces tâches et fonctionnalités souhaitées en plusieurs sous-tâches distinctes et aussi courtes que possible, comme l'exige la méthode de Kanban. Chacune de ces tâches doit fournir un livrable clairement identifiable.

Puis, avec mon équipe, nous avons priorisé ces tâches et décidé lesquelles parmi celles-ci seraient intégrées dans la première version ou dans des versions ultérieures.

Découper nos différents projets a nécessité une certaine organisation. Dans le futur, notamment grâce à mon organisation, je serai plus à l'aise et efficace pour identifier les tâches composant un projet ainsi que pour créer leur ticket respectif.

Une fois le projet découpé en tâches, vient l'estimation de la charge de travail. En ce qui me concerne, j'ai, surtout au début, rencontré des difficultés pour estimer le temps de travail des différentes tâches. En effet, mon manque d'expérience a été une contrainte pour l'évaluation de la charge de travail. J'ai donc été assisté par Issam, mon tuteur. Puis, après un certain temps et ayant acquis de l'expérience, j'ai été capable, notamment grâce à ma prise de décision, de réaliser cette évaluation de la charge de travail.

Par ailleurs, avec la méthode Kanban, l'estimation de l'effort est optionnelle. Ainsi, sur certains projets, nous n'avons pas fait cette estimation, notamment lorsque nous devons travailler sur d'autres sujets entre la réalisation des tâches de ce projet.

En outre, j'ai compris que l'estimation de la charge de travail n'est pas définitive. Effectivement, celle-ci peut évoluer au fur et à mesure que l'équipe prend de l'expérience.

Avec Zoranne, j'ai appris à évaluer la charge de travail pour les tâches qui composent un projet et je pourrai donc mettre en œuvre cette compétence à l'avenir.

### 3.2.3 Contrôler la réalisation des livrables

Afin de contrôler la réalisation des livrables, je vérifiais régulièrement l'état des tickets au sein des tableaux de Kanban. Effectivement, je m'assurais que les tickets ne restaient pas trop longtemps dans la même colonne. Ainsi, je pouvais garantir que les tickets étaient bien réalisés en temps et en heure.

En outre, avec l'aide de mon tuteur Issam, en plus des déploiements automatiques en préproduction, nous effectuions des mises en production à la fin de chaque application. Par exemple, une fois que j'ai terminé l'application Pricing, nous avons pu la déployer en production afin de permettre aux équipes métier de contrôler la cohérence des données.

Par ailleurs, grâce à la méthode Kanban, nous pouvons effectuer une livraison continue. Ainsi, je vérifie que chaque tâche réalisée correspond bien à la demande formulée dans le ticket et donc, aux attentes. Ce contrôle qualité est essentiel afin de s'assurer que les développements soient conformes aux attentes.

Dans un premier temps, avant que je mette en place les tests unitaires, je faisais cette vérification manuellement, en testant que les différentes possibilités fournissaient bien le résultat attendu. D'abord, j'effectuais ce contrôle durant la phase de développement, avant de publier mon travail sur le répertoire Bitbucket. Ce premier contrôle permettait de publier uniquement les développements parfaitement fonctionnels en local. Ensuite, une fois ces développements validés en local puis publiés, ils étaient automatiquement déployés en préproduction. Enfin, avec mon équipe et les équipes métier concernées, j'effectuais de nouveaux des tests afin de garantir la qualité et la conformité des développements réalisés.

Puis, dans un second temps, après avoir mis en place les tests unitaires, le contrôle qualité se faisait automatiquement. En effet, pendant les développements, il me suffisait d'exécuter les tests unitaires précédemment écrits afin de contrôler la qualité du code. Puis, une fois les tests passés avec succès localement, je publiais le code sur git. Ensuite, les tests unitaires étant intégrés à notre approche CI/CD, le code était de nouveau testé et déployé en préproduction uniquement si les tests étaient passés avec succès.

Cependant, j'ai fait l'erreur de mettre en place les tests unitaires tardivement. Par conséquent, le fait de tester manuellement les développements prenait beaucoup de temps. Aussi, rien ne garantissait que le code publié et déployé ne comportait pas d'erreur. Le contrôle qualité était donc moins efficace.

Ainsi, à l'avenir, j'intégrerai les tests unitaires au processus de contrôle qualité dès le début du projet.

### 3.3 Gérer les ressources du projet

#### 3.3.1 Gérer une équipe

L'équipe de développement du projet Zoranne est constituée de trois personnes. Thomas, responsable digital, apporte son expertise sur les processus déjà en place et sur la pertinence des nouveaux processus à mettre en place. Issam est le responsable IT et mon tuteur au sein de l'entreprise. Il m'aide sur les choix techniques lorsque cela est nécessaire et sur la mise en place des serveurs (Ops).

Au sein de notre équipe, il y a de nombreuses interactions. En effet, chacun d'entre nous joue un rôle particulier et apporte son expertise afin de mener au mieux la réalisation du projet.

Pour cela, nous échangeons nos points de vue régulièrement. Effectivement, je demande notamment que l'on effectue des daily meetings. Ces petites réunions quotidiennes permettent de faire le point sur les tickets et le travail réalisé la veille et de discuter des éventuelles difficultés ou choses bloquantes rencontrées. Nous décidons également des prochaines tâches à réaliser au cours de la journée. Ainsi, je priorise les tâches en donnant mon point de vue et mes arguments les concernant. Nous en discutons ensuite tous les trois, en fonction des délais que nous souhaitons respecter, avant de prendre la décision finale. Aussi, travaillant de manière agile, il arrive que j'organise une petite réunion au cours de la journée lorsque je fais face à quelque chose de bloquant ou lorsque j'ai des doutes sur les règles métiers à mettre en place.

En mettant en œuvre ces réunions, j'ai su manifester et développer mon organisation. Par conséquent, à l'avenir, je serai capable d'employer cette compétence, que ce soit dans l'organisation de réunions ou encore, plus généralement, dans la planification d'un projet.

Pour ma part, étant arrivé dès le début du projet, j'ai pu participer aux toutes premières décisions sur le projet. J'ai notamment fait le choix des technologies à utiliser (Symfony et Angular).

Par ailleurs, je souhaitais que nous travaillions de manière agile. En effet, la méthode agile est pour moi la meilleure manière de mener à bien un projet. Cette méthode permet d'être extrêmement flexible et donc réactif.

Aussi, la communication est au centre des priorités, ce qui permet de rester focalisé sur les objectifs et d'éviter les quiproquos.

Le fait que chaque projet constituant le programme puisse être indépendant permet également de réduire le Time to Market des différentes applications, c'est-à-dire de réduire le temps entre le développement d'une application et son intégration dans les processus existants ainsi que sa livraison aux équipes métier. La méthode agile laisse également la possibilité de contrôler régulièrement les contraintes de délai et de ne pas s'en éloigner.

Au fur et à mesure, j'ai appris à plus participer lors des échanges et à défendre mon point de vue lorsque j'étais convaincu qu'il était correct. J'ai su, avec le temps, gagner en autonomie et proposer des solutions ainsi que débattre avec mon équipe afin de faire avancer les projets.

J'ai donc su développer ma prise de décision en défendant mes choix et mes points de vue. Je serai ainsi en mesure de mettre en œuvre cette compétence lors de mes futurs projets, notamment pour diriger une équipe et prendre rapidement les décisions qui me paraissent les plus adaptées.

En tant que développeur sur l'ensemble des projets, j'ai également dû apprendre à me gérer moi-même. En effet, lors des développements j'ai été amené à prendre des décisions.

Après avoir pris ces initiatives, j'expliquais mes choix auprès de l'équipe, qui validait ou non mes décisions. Avec le temps, j'ai appris à être plus autonome et à prendre plus d'initiatives.

Aussi, j'ai appris à gérer mon temps. Effectivement, au début, lorsque j'avais très peu d'expérience, il était pour moi difficile d'évaluer le temps de travail d'une tâche.

Il arrivait également que je perde du temps sur des petits détails non pertinents. Au fur et à mesure, j'ai su estimer le temps de travail nécessaire pour réaliser mes tâches. Grâce à la méthode agile, j'ai pu, avec l'équipe, avoir une idée régulière des délais à respecter.

Par ailleurs, lors de la réalisation de mes tâches, j'ai appris à prioriser celles-ci. En effet, au début, lorsque j'avais des doutes sur la priorité des tickets, j'en discutais avec l'équipe lors des daily meetings. Puis, avec le temps, j'ai su être autonome pour prioriser mes tâches, c'est-à-dire pour réaliser ce qui est vraiment pertinent en priorité.

L'acquisition et l'enrichissement de toutes ces compétences m'ont permis d'être plus efficace. En outre, ces compétences me serviront dans le futur, notamment dans la gestion d'un projet ou d'un ensemble de projets (programme) et d'une équipe. En effet, ayant en partie géré le programme et réalisé les développements des différents projets, lorsque je devrai gérer l'équipe d'un projet ou programme ou projet, je connaîtrai les problèmes auxquels les développeurs peuvent être confrontés. Je serai donc en mesure de les accompagner de la meilleure des manières.

### 3.3.2 Maîtriser le budget

Dans le cadre de Zoranne, je n'ai pas eu l'occasion d'établir, ni même de participer à l'établissement du budget. En effet, la gestion et le contrôle des coûts n'ont pas fait partie de mes missions.

Cependant, je suis conscient, notamment grâce à la formation que j'ai suivie chez Aston, qu'il est possible de maîtriser le budget en étant rigoureux et en mettant en place certaines actions. Lorsque j'ai appris les enjeux liés au budget, j'ai alors pu agir de sorte à ce qu'il soit respecté.

Tout d'abord, je suis conscient qu'un tel projet nécessite des ressources qui ont un impact sur le coût. Par exemple, le coût de l'hébergement OVH et des licences Atlassian permettant de gérer les tickets sur Jira ainsi que l'hébergement des développements sur Bitbucket est à prendre en compte.

Le coût d'un projet dépend directement de la charge, c'est-à-dire de la quantité de travail à allouer au projet, ainsi que du temps nécessaire à l'élaboration de ce dernier (la durée).

Par conséquent, il est essentiel de gérer les délais et de veiller à ne pas les dépasser. Effectivement, lorsque le temps de travail sur le projet est plus important que prévu lors de la budgétisation, le budget va être dépassé.

Ayant compris l'importance du temps de travail à allouer au projet ainsi que de la gestion des délais, j'ai pu faire en sorte, à mon échelle, d'éviter le dépassement du budget.

Pour cela, lors des échanges avec mon équipe, comme on peut le voir sur l'image ci-dessous, j'ai également appris à fournir sur Jira une estimation du temps de développement nécessaire pour les différentes fonctionnalités qui me sont attribuées. Effectivement, l'estimation de chacune des tâches permet ainsi d'obtenir une estimation globale de la charge de travail pour le projet.



[API] - WS WMS de MAJ de Stock (+n/-n)	Zoranne WMS API	<input checked="" type="checkbox"/> = 3d	ZOR-52 SV
[API] - Export des données produits depuis les canaux de vente	Zoranne Ref API	<input checked="" type="checkbox"/> = 4h	ZOR-40 SV
[API] - Appeler la méthode de MAJ de stock et de création de la ligne expédition dans WMS	Zoranne OMS API	<input checked="" type="checkbox"/> = 2h	ZOR-8 SV
[API] - WS qui permet de cloner une commande	Zoranne OMS API	<input checked="" type="checkbox"/> = 4h	ZOR-22 SV
[Job] - Import des fiches produits depuis BC vers Quable	Zoranne Jobs	<input checked="" type="checkbox"/> = 2h	ZOR-36 SV
[API] - WS qui retourne la liste des raisons de remboursement	Zoranne OMS API	<input checked="" type="checkbox"/> = 2h	ZOR-18 SV

*Estimation de la charge – Tableau de Kanban*

Puis, en effectuant un suivi régulier, je suis en mesure d'évaluer si nous sommes en avance ou si, au contraire, des retards commencent à apparaître, et ainsi corriger la tendance.

L'expérience que j'ai acquise durant ces échanges m'a donc permis d'apprendre à évaluer des tâches et m'a fait comprendre l'importance des estimations pour se situer concernant l'état d'avancement du projet.

Aussi, suite à nos échanges pendant les daily meetings, nous établissons sur notre système de suivi de projet Jira les tickets pour les tâches à réaliser. Plus précisément, nous utilisons des tableaux de Kanban pour visualiser l'ensemble des tâches de nos projets.

Ainsi, je peux contrôler l'état d'avancement des tickets et vérifier si les délais sont respectés. En effet, je vérifie que les tickets sont bien réalisés en temps et en heure et qu'ils sont bien placés au sein du tableau de Kanban. En outre, je m'occupe de contrôler s'il existe bien un ticket pour chaque tâche à réaliser. De même, je priorise les tâches que je vais réaliser lors des développements et je vérifie que les tickets ont le bon statut, c'est-à-dire qu'ils sont bien placés au sein du tableau de Kanban sur Jira.

Au début, j'ai fait l'erreur de laisser un nombre trop important de tâches dans les colonnes « en cours » des tableaux de Kanban. Cependant, cela a pour conséquence de supprimer le cadre que les tableaux de Kanban sont censés fournir. Aussi, en entraînant une surcharge de travail, notre organisation est altérée. Après avoir pris conscience de cette erreur, j'ai su la corriger en m'imposant des limites sur le nombre de tâches placées dans les colonnes « en cours ».

Grâce à mon organisation, j'ai donc appris à prioriser et à contrôler l'état d'avancement des tâches au sein des tableaux de Kanban. Dans le futur, je pourrai ainsi mettre en œuvre ces compétences dans le but de veiller au bon déroulement du projet.

Bien que Zoranne soit un projet interne et que nous sommes relativement flexibles concernant les délais, il est important que la charge de travail allouée au projet ne dépasse pas celle prévue initialement

J'ai ainsi appris à respecter des limites et à me responsabiliser, notamment grâce aux recettes avec les équipes métier et le président de l'entreprise, pendant lesquelles je présente le travail que j'ai réalisé.

Effectivement, lorsque la première version d'une application est terminée, j'organise une réunion (recette) avec le président de l'entreprise ainsi qu'avec l'équipe métier concernée par l'application.

La première version d'une application est constituée, au minimum, des mêmes fonctionnalités et processus (iso-fonctionnel) qu'auparavant, sur PrestaShop. En outre,

pour certaines applications, de nouvelles fonctionnalités sont intégrées afin de faciliter le travail des équipes métier.

Ainsi, lors des recettes avec les équipes métier, je présente l'application en remettant dans le contexte. Plus précisément, j'explique ce qu'elle vise à remplacer dans nos processus actuels et les éventuels changements que cela implique.

Ensuite, je réalise une démonstration de l'application front. Ainsi, je présente toutes les pages de l'application en détaillant les fonctionnalités et possibilités présentes sur cette dernière. J'explicite chaque élément et fais le parallèle avec ce qui était en place auparavant afin que les équipes comprennent ce que l'application remplace.

Puis, après avoir tout expliqué, je demande aux équipes métier de tester l'application. Plus particulièrement, je leur demande de vérifier si les données de l'application Zoranne sont les mêmes que sur PrestaShop. Par exemple, pour la synchronisation et la visualisation des stocks, il s'agit de vérifier que les quantités stock de PrestaShop sont égales à celles de Zoranne. Après la réalisation des tests et des éventuels ajustements, l'application est déployée en production et intégrée à nos processus.

Ces recettes m'ont permis d'apprendre à synthétiser le travail que j'ai réalisé et à être à l'aise pour communiquer lors des réunions.

Par ailleurs, j'ai appris à rendre des comptes sur le travail que j'ai effectué. En effet, le fait d'avoir des recettes à la fin de chaque première version des applications permet de respecter les délais fixés et impose une certaine qualité pour l'application. Effectivement, Il est nécessaire que les équipes soient convaincues par ce que je leur présente car c'est ce qui va venir remplacer leurs outils actuels.

D'autre part, ces réunions permettent d'avoir des retours des équipes afin d'améliorer l'application qui sera leur outil. Ainsi, les recettes avec les équipes métier et le président de l'entreprise m'ont permis de me responsabiliser.

Enfin, pour ce qui est de ROI, Zoranne permettra de faciliter l'ouverture de nouvelles marketplaces. En effet, l'application Referential fournit plus efficacement les éléments nécessaires à l'ouverture d'une place de marché tels que la traduction des fiches produits sur notre PIM (Quable), ou encore la génération d'un flux pour Shopping Feed.

De plus, Zoranne, de par son architecture, a la possibilité de s'adapter à d'autres ERP, et donc d'autres sociétés. Par conséquent, nous pourrions proposer Zoranne à d'autres entreprises et ainsi augmenter notre chiffre d'affaires.

Aussi, comme l'ensemble des applications constituant Zoranne sont développées en interne, les futures évolutions ne nécessiteront pas d'appel vers un tiers et nous ne serons pas limités par des contraintes extérieures. Zoranne a donc pour conséquence de réduire les coûts liés aux futures évolutions.

## 3.4 Gérer les risques du projet

### 3.4.1 Identifier les risques du projet

Lorsque j'ai commencé à travailler sur le projet Zoranne, j'ai appris à identifier les risques d'un tel projet. En effet, un projet de cette envergure nécessite d'avoir connaissance des risques afin d'éviter les imprévus. Ainsi, une fois les risques clairement identifiés, il est possible de mettre en place des mesures de prévention pour les éviter.

En ce qui me concerne, je me suis rendu compte de certains risques pendant la phase de développement. En effet, j'ai fait l'erreur de commencer le développement trop rapidement, en sous-estimant la phase d'analyse des risques.

Par conséquent, j'ai compris qu'il fallait, dès le début du projet, passer en revue l'ensemble des risques possibles. À l'avenir, je ne ferai donc plus l'erreur de sous-estimer la phase d'analyse et de me précipiter sur le développement.

Parmi les risques que j'ai identifiés, on retrouve notamment les délais. Effectivement, Zoranne est un projet très attendu par les membres de l'entreprise et le non-respect des délais peut entraîner un dépassement du budget. Ainsi, il est important de respecter les délais fixés et de livrer les applications rapidement.

Le risque de ne pas respecter les délais étant plutôt connu et commun à tous les projets, j'ai été capable de l'identifier dès le début du projet.

De plus, il est essentiel d'éviter les régressions par rapport aux processus déjà en place sur PrestaShop. En effet, l'objectif des premières versions de Zoranne est de proposer au moins les mêmes fonctionnalités qu'auparavant, sur PrestaShop. Par conséquent, il est primordial qu'il n'y ait pas de régressions.

Ainsi, j'ai su mettre en lumière ce risque avant la phase de développement.

Aussi, il est nécessaire de répondre correctement aux attentes des équipes métier et de ne pas s'en écarter. Effectivement, lorsqu'on ne communique pas suffisamment avec les équipes ou que l'on ne connaît pas les processus en place, le risque est de ne pas répondre aux besoins des collaborateurs et d'être hors sujet. De surcroît, les équipes ont beaucoup d'attentes car, depuis le début du projet, on ne répond plus à leurs nouveaux besoins dans PrestaShop, mais on les intègre au sein de Zoranne.

Lorsqu'on m'a présenté Zoranne, mon tuteur a mis l'accent sur le fait que le projet doit faciliter le travail des collaborateurs au quotidien. Ainsi, j'ai tout de suite compris que le fait de ne pas répondre aux besoins des équipes métier était un risque à ne pas négliger.

Par ailleurs, il est important de mettre en place une architecture permettant de s'adapter à d'autres ERP, et même à d'autres sociétés. En effet, d'après moi, le risque est de se focaliser sur le besoin et le contexte courant, sans se préoccuper de la réutilisation et de la flexibilité pour l'avenir. Ainsi, j'ai compris que, dès le début du

projet, il fallait penser à une architecture flexible afin que les différentes applications puissent être consommées par un autre ERP que celui que nous utilisons actuellement (BC), ou encore, une autre société que la nôtre, comme notamment la société sœur d'Allstore : Garden Max. Aussi, l'architecture doit permettre de rendre le projet pérenne et maintenable sur le long terme.

Cependant, j'ai pris conscience de ce risque tardivement, lors des développements. Par conséquent, j'ai dû modifier le modèle de la base de données, et donc mon code afin que l'architecture puisse être flexible et s'adapter à d'autres situations.

D'autre part, il existe un risque que les performances ne soient pas à la hauteur. Effectivement, lors de la migration d'une application monolithique vers une architecture en microservices, il est possible de perdre en performance.

Par conséquent, il faut rester vigilant et, dans la mesure du possible, optimiser le code existant afin d'avoir les meilleures performances possibles.

En revanche, je me suis aperçu qu'avoir de faibles performances était un risque pendant la phase de développement. Avant cela, je ne suspectais pas que la migration pourrait entraîner une perte de performances et que la rapidité était essentielle, tant pour les processus back que pour les applications front.

### 3.4.2 Évaluer et hiérarchiser les risques

Suite à l'identification des risques, j'ai appris, grâce à mon tuteur, à les évaluer et à les hiérarchiser. L'évaluation des risques permet ensuite d'effectuer leur hiérarchisation en fonction de l'incidence et de la probabilité de réalisation de ces risques. J'estime que la hiérarchisation des risques est primordiale car elle permet de se concentrer sur ceux qui représentent une réelle menace en priorité, tout en étant conscient des autres risques. De plus, l'évaluation des risques permet de mettre en place des mesures de préventions adaptées.

Pour cela, pour chaque risque que j'ai identifié, j'ai combiné leur gravité et leur probabilité d'apparition. En effet, un risque est défini par l'impact qu'il peut avoir s'il survient, ainsi que par la probabilité qu'il survienne.

Le fait d'avoir appris à évaluer et hiérarchiser des risques me permettra, lors de mes prochains projets, de connaître l'importance et l'impact que pourraient avoir chacun d'entre eux dès le début du projet, avant de commencer les développements.

		Gravité				
		1 Négligeable	2 Mineure	3 Modérée	4 Majeure	5 Catastrophique
Probabilité	5 Très probable	5	10	15	20	25
	4 Probable	4	8	12	16	20
	3 Possible	3	6	9	12	15
	2 Peu probable	2	4	6	8	10
	1 Très peu probable	1	2	3	4	5
		(1-6) : Risque faible		(7-12) : Risque modéré		(13-25) : Risque élevé

*Matrice des risques pour Zoranne*

La matrice des risques ci-dessus met en évidence l'importance des risques que j'ai identifiés, en fonction de leur probabilité et de leur gravité.

Ainsi, comme le montre la matrice des risques, d'après moi, le risque le plus important concerne les régressions. En effet, l'objectif principal du projet (pour la première version) étant de migrer les fonctionnalités sur PrestaShop vers plusieurs

applications, il est indispensable de ne pas avoir de régressions. S'il y a des régressions, les équipes métier ne seront pas convaincues par le projet et auront donc des difficultés à migrer de PrestaShop vers Zoranne.

Puis, j'estime que mettre en place une architecture qui ne permettrait pas d'évolution et qui serait propre à notre environnement uniquement représente le deuxième risque le plus important. Le projet Zoranne vise à remplacer une application monolithique tout en étant enrichi sur le long terme. Ainsi, il est nécessaire que l'architecture mise en place puisse être flexible et s'adapter à plusieurs environnements (ERP par exemple). Aussi, le fait de pouvoir proposer nos applications à d'autres sociétés n'est pas négligeable économiquement.

Ensuite, vient le risque de ne pas répondre correctement aux besoins des équipes métier. Effectivement, le projet Zoranne, et plus particulièrement les applications front, seront principalement utilisées par les équipes. Il est donc crucial de répondre à leurs attentes afin que la migration soit la plus simple possible.

En quatrième position, j'ai placé le risque d'avoir des performances faibles. En effet, la migration d'une application monolithique vers des microservices peut entraîner une perte de performance sur certains processus. Cependant, les tâches effectuées par Zoranne peuvent être complexes et gourmandes en ressources, comme notamment la synchronisation des stocks. Or, ces outils seront utilisés quotidiennement au sein de nos processus et par les équipes métier. Il faut donc avoir conscience de ce risque et le prendre en compte lors des développements. Lors de la hiérarchisation, j'ai placé ce risque à la quatrième position car la probabilité que ce dernier se manifeste est plus faible que pour les trois premiers.

Enfin, le point de vigilance qui est, d'après moi, le moins risqué concerne les délais. Effectivement, le fait que Zoranne soit un projet interne nous permet d'être plutôt flexible concernant les délais. Cependant, il s'agit tout de même d'un risque à



ne pas négliger car les équipes et le président de la société ont beaucoup d'attentes pour Zoranne. Par conséquent, il est important de présenter régulièrement le travail effectué, et donc de s'imposer des délais.

### 3.4.3 Définir les actions pour minimiser les risques

L'évaluation et la hiérarchisation des risques m'ont permis de mettre en place des actions de prévention afin de les minimiser. En effet, l'identification puis l'évaluation des risques n'ont de sens que si des actions sont mises en place pour les éviter.

Pour cela, j'ai repris la liste des risques que j'avais identifiés au début du projet. Puis, pour chacun d'entre eux et avec l'aide de mon tuteur Issam, j'ai réfléchi à une solution, à une manière de faire afin de les contourner. En ce qui concerne les risques que j'ai identifiés tardivement, lors de la phase de développement, je les ai immédiatement pris en compte pour la suite du projet.

Pour ce qui est d'éviter les régressions, j'ai estimé que la solution était composée de plusieurs éléments. Tout d'abord, il est nécessaire de connaître l'existant. En effet, afin d'éviter les régressions, il faut maîtriser les processus actuellement en place. Plus précisément, il est important de savoir quelles sont les fonctionnalités utilisées par les équipes ainsi que la finalité des tâches back-end (déclenchées manuellement ou automatiquement).

Ainsi, la communication est également une composante de la solution. Effectivement, afin d'obtenir toutes ces informations, j'ai principalement échangé avec Thomas, le responsable digital, qui maîtrise parfaitement les processus initialement en place. Au fur et à mesure des développements, il a pu m'apporter toutes les indications nécessaires afin que Zoranne ne comporte aucune régression par rapport aux fonctionnalités sur PrestaShop. Aussi, j'ai eu l'occasion d'échanger avec des membres des différentes équipes qui ont pu m'informer sur leur travail au quotidien ainsi que sur les fonctionnalités qu'ils utilisaient.

Enfin, d'après moi, le fait de tester que chaque fonctionnalité réalisée ait bien le comportement souhaité est essentiel pour éviter les régressions. Ainsi, pour chaque fonctionnalité, je teste que, pour tous les cas possibles, cette dernière se comporte correctement. Ensuite, grâce à notre approche CI/CD, chaque tâche poussée sur GIT se retrouve sur notre environnement de pré-production et peut être vérifiée par mes collègues Issam et Thomas. Plus tard, j'ai mis en place des tests unitaires permettant de vérifier que chaque développement a bien le comportement attendu. Ces tests unitaires sont exécutés lors du développement ainsi qu'à chaque déploiement, dans la continuité de notre approche CI/CD.

Ainsi, la mise en place de l'ensemble de ces mesures permet d'éviter les régressions par rapport aux fonctionnalités en place sur PrestaShop.

Puis, d'après moi, pour éviter le risque de ne pas réaliser une architecture flexible qui s'adapterait à d'autres environnements, il est important d'anticiper. En effet, il faut garder en tête dès la phase d'analyse du projet que le projet Zoranne a pour objectif de s'adapter à d'autres environnements, notamment d'autres ERP et même d'autres sociétés.

Le projet doit aussi être viable sur le long terme. Effectivement, il est essentiel que ce dernier puisse être amélioré et enrichi au fur et à mesure. Pour éviter ce risque, il faut donc penser à ce que pourrait être le besoin à l'avenir, en plus de ce qu'il est actuellement.

Pour cela, j'ai établi, avec l'aide de mon tuteur, une architecture basée sur les microservices. Ainsi, le projet est divisé en plusieurs applications distinctes et capables d'évoluer. J'ai également intégré dans mon modèle une dimension « *d'entreprise* » et de « *canal de ventes* » afin que les applications puissent être consommées par d'autres sociétés.

Ensuite, tout comme pour les régressions, j'ai considéré que la solution pour éviter de ne pas répondre correctement aux besoins des équipes dépendait de

plusieurs paramètres. En effet, j'ai compris, qu'une fois encore, la connaissance des processus et des tâches réalisées par les équipes métier était primordiale. Pour cela, j'ai échangé avec les équipes concernées par les applications à réaliser. Aussi, j'ai pu récolter beaucoup d'informations auprès de Thomas qui connaît très bien les tâches réalisées par les différentes équipes ainsi que les processus en place.

De plus, j'ai pris conscience que la meilleure façon de savoir si les applications correspondaient aux besoins des collaborateurs était de communiquer directement avec eux. Par conséquent, j'ai organisé, avec l'aide de mes collègues, des réunions avec les équipes concernées par l'application réalisée afin de leur présenter et d'échanger avec elles. Ainsi, j'ai pu récupérer leurs avis et apporter les modifications nécessaires pour que l'application réponde au mieux à leurs besoins.

Par ailleurs, j'ai compris que pour minimiser le risque d'avoir de faibles performances lors de la migration, il était important de mettre en place de l'asynchronisme. Effectivement, d'après moi, comme nos applications doivent communiquer entre elles pour fonctionner, il est nécessaire que cette communication soit asynchrone. Ainsi, les utilisateurs ne sont pas impactés par les tâches volumineuses car elles sont gérées sous la forme d'une file d'attente par RabbitMQ.

Par ailleurs, j'ai jugé qu'il ne fallait pas se contenter de reprendre le code existant au sein de PrestaShop. En effet, j'ai estimé qu'il fallait, dans la mesure du possible, optimiser le code existant. Ainsi, ces optimisations ont permis de compenser la perte de performance lors de la migration de l'application PrestaShop monolithique vers l'architecture basée sur les microservices.

En outre, les tests permettant de comparer les temps d'exécution des différentes requêtes sont évidemment nécessaires afin de s'assurer que les performances soient à la hauteur de nos attentes.

Ainsi, la mise en place de l'asynchronisme, l'optimisation du code existant et la réalisation de tests permettent de minimiser le risque de perdre en performance lors de la migration.

Enfin, j'ai appris que pour que le temps de réalisation du projet soit maîtrisé, et donc indirectement le budget, il fallait s'imposer des délais. En effet, pour ne pas dépasser le budget, il est nécessaire de se fixer des délais et de s'y tenir afin de rester productif et efficace. Pour cela, avec l'aide d'Issam et Thomas, j'ai revu la priorisation des tâches lorsque c'était nécessaire. Effectivement, j'ai donc priorisé les tâches que nous considérons comme essentielles et dépriorisé celles qui ne l'étaient pas.

Puis, pour respecter les délais fixés, j'estime qu'il est important de faire le point régulièrement, c'est-à-dire de faire l'analyse de l'état d'avancement des tâches lors d'échanges avec l'équipe. Ainsi, j'ai pu évaluer si les délais étaient ou menaient à être respectés ou si du retard risquait d'apparaître. Cette analyse permet ensuite de rectifier la tendance, c'est-à-dire d'augmenter la cadence si cela est nécessaire.

Le fait d'avoir appris à mettre en place des actions afin d'éviter les risques d'un projet me permettra, à l'avenir, d'être plus efficace sur leur gestion. Effectivement, lors de mes prochains projets je serai capable, notamment pour les risques que j'ai rencontrés sur Zoranne, de savoir quelles actions mettre en place pour les minimiser. Aussi, j'aurai conscience de l'importance de mettre en place ces actions le plus tôt possible afin d'éviter au mieux les risques identifiés.

Par ailleurs, j'ai commis l'erreur de mettre en place des tests unitaires très tardivement, lorsque plusieurs applications étaient déjà développées. Par conséquent, avant que ces derniers soient intégrés, du code comportant des erreurs était déployé en préproduction. J'ai donc compris que la mise en place des tests unitaires était primordiale, avant même de commencer à développer. Ainsi, à l'avenir, je ne commettrai plus cette erreur et je mettrai donc en place des tests unitaires dès le début du projet.

## 3.5 Communiquer

### 3.5.1 Répondre à un besoin

Peu de temps après mon arrivée dans l'entreprise, on m'a présenté le projet Zoranne. Dans le cadre d'une réunion kick off, mon équipe m'a alors exposé le besoin auquel ce projet doit répondre. En effet, suite au développement de l'entreprise, il devenait nécessaire de se séparer de l'architecture monolithique en place avec PrestaShop. Ainsi, le besoin était de mettre en place une nouvelle architecture en reprenant, dans un premier temps, les fonctionnalités déjà en place (ISO fonctionnel) pour, à terme, se séparer de PrestaShop pour tous nos processus métier. En outre, cette nouvelle architecture doit être scalable, c'est-à-dire qu'elle doit permettre de faire évoluer les différentes applications qui la composent, sans avoir à passer par des prestataires externes, diminuant ainsi les coûts. De surcroît, avant la réalisation de chacune des applications composant Zoranne, nous avons réalisé une réunion de kick off afin de clarifier le rôle et les fonctionnalités de celles-ci.

J'ai donc appris, à partir de la formulation de ce besoin, à mettre en place une solution pour y répondre.

Afin de répondre à ce besoin, j'ai commencé par identifier les acteurs concernés par le projet. Tout d'abord, les équipes métier concernées varient selon le projet.

En effet, l'application Logistic Optim sera principalement utilisée par Florian, le coordinateur logistique.

Aussi, l'application WMS sera majoritairement exploitée par Florian ainsi que par Christopher (traffic manager).

Ensuite, l'application Pricing sera quant à elle utilisée par Jimmy (président de l'entreprise) pour modifier les prix des produits, par le service marketing (notamment pour créer des opérations commerciales) ainsi que par le service client afin de consulter le prix des produits.

Pour ce qui est de l'application OMS, elle sera principalement utilisée par le service client afin de pouvoir retrouver la ou les commandes nécessitant une prise en charge, telle qu'un remboursement, la modification de la commande ou encore consulter la facture, le suivi de la commande, etc.

Je n'ai pas eu à répondre à un cahier des charges mais à un besoin que l'on m'a présenté et expliqué. L'élément principal qui m'a permis cela est la communication. En effet, c'est en communiquant que nous avons pu décider de l'architecture qui conviendrait au mieux pour répondre à ce besoin.

De surcroît, c'est également en échangeant, que ce soit avec mon équipe ou encore les équipes métier concernées, que j'ai pu mettre en place l'architecture choisie. Effectivement, nos échanges m'ont permis de maîtriser les fonctionnalités et processus initialement en place ainsi que de prendre, tout au long du projet, les meilleures décisions.

Par ailleurs, il a fallu accompagner les équipes métier aux changements qu'implique Zoranne. En effet, après la mise en production de Zoranne, les processus seront différents pour les équipes. Initialement, elles réalisaient toutes leurs tâches au sein de PrestaShop.

Cependant, avec Zoranne, l'objectif est que les équipes métier ne travaillent plus sur le backoffice de PrestaShop mais sur les différentes applications de Zoranne. Pour cela, il est tout d'abord nécessaire que ces applications correspondent parfaitement à leurs besoins afin qu'elles y voient une plus-value. Ainsi, avant de commencer les développements de l'application front-end, j'ai eu l'occasion d'échanger avec les collaborateurs afin d'identifier précisément leurs besoins, que ce soit en termes de projet qu'en termes de fonctionnalités, pour l'application sur laquelle ils allaient travailler.

Ensuite, une fois l'application réalisée, je la présentais dans le cadre d'une réunion aux équipes concernées par celle-ci. Je prenais soin de rappeler le contexte et d'expliquer clairement l'ensemble des fonctionnalités, en faisant le parallèle avec ce

qu'ils faisaient sur le backoffice de PrestaShop. Je restais également disponible et à l'écoute des éventuelles interrogations et idées d'amélioration.

D'après moi, l'accompagnement au changement est crucial. En effet, les différentes applications développées dans le cadre de Zoranne n'ont de sens que si elles utilisées par les équipes métier. Par conséquent, j'ai fait au mieux pour les accompagner durant cette migration.

Par conséquent, j'ai appris à communiquer avec mes collègues afin de répondre au mieux au besoin que l'on m'a présenté. J'ai ainsi compris l'importance de la communication pour mener à bien un tel projet.

Dans le futur, je pourrai donc mettre en œuvre la capacité à communiquer que j'ai acquise tout au long de l'élaboration du projet Zoranne.

### 3.5.2 Mobiliser l'équipe pour favoriser l'avancement du projet

D'après moi, la communication avec les membres de mon équipe, Issam et Thomas, est essentielle pour mener à bien le projet. En ce sens, j'ai appris à mobiliser mon équipe afin d'échanger et ainsi, prendre les meilleures décisions pour faire avancer le projet.

Je considère notamment qu'organiser et participer à des réunions est primordial dans un souci d'amélioration continue ainsi que pour échanger nos points de vue régulièrement.

Par conséquent, j'ai eu l'occasion d'organiser plusieurs types de réunions. Tout d'abord, suite à l'application de la méthode agile, j'ai commencé par mettre en place une réunion quotidienne d'une vingtaine de minutes avec les membres de mon équipe (Issam, Thomas et moi). Ces réunions quotidiennes, nommées « daily meetings », donnent lieu à des échanges essentiels pour l'avancée du projet. En effet, elles donnent

la possibilité de lever certaines interrogations et d'échanger sur les difficultés rencontrées. Aussi, en énonçant les tâches réalisées ainsi que les tâches à effectuer dans la journée, elles permettent de faire le point sur l'état d'avancement des tâches au sein du tableau de Kanban.

En revanche, j'ai pris conscience d'avoir commis une erreur dans la mise en place des daily meetings. Effectivement, j'ai manqué de rigueur dans leur réalisation. Plus précisément, certains jours, il est arrivé que nous ne réalisions pas cette réunion. Or, à mon sens, cette dernière est essentielle pour le bon déroulement du projet et ne doit pas être manquée. Ainsi, à l'avenir je serai plus rigoureux sur l'application de ces réunions.

La communication que j'ai développée, que ce soit lors des daily meetings ou encore au quotidien avec mes collègues, pourra ainsi me servir à l'avenir, notamment dans la gestion d'une équipe. Effectivement, je serai en mesure de gérer les daily meetings et, plus généralement, d'interagir avec mon équipe.

Ensuite, j'ai organisé des réunions visant à présenter aux équipes métier le travail réalisé. En effet, à l'issue de la réalisation de la première version de chaque application, je me suis occupé de les présenter aux équipes métier concernées. Ainsi, j'introduisais le sujet en mettant en parallèle l'existant avec l'objectif recherché par l'application Zoranne en question. Puis, je présentais l'application d'administration (que j'ai développée avec Angular) en expliquant le fonctionnement de chaque fonctionnalité et en la rapportant avec l'existant afin que ce soit le plus clair possible pour les équipes. Aussi, j'expliquais brièvement les processus backend, c'est-à-dire leur fonctionnement de manière générale et, pour les tâches automatisées, leurs horaires d'exécution.

Par exemple, le 9 septembre 2021, j'ai introduit la nouvelle application Logistic Optim auprès des personnes concernées telles que le Jimmy (président de l'entreprise), Florian (coordinateur logistique) et Christopher (traffic manager). Ensuite, début mars, j'ai présenté la partie stock de l'application WMS à ces mêmes personnes.



Étant plutôt timide, j'ai au début eu quelques difficultés à m'adresser de manière claire et concise aux équipes. Effectivement, il m'a fallu un peu de temps et d'expérience pour maîtriser et être à l'aise avec mes interlocuteurs.

Puis, j'ai eu l'occasion d'organiser et de participer à des ateliers pendant lesquels nous échangeons sur un sujet spécifique afin de l'éclaircir et développer la meilleure solution possible sur Zoranne. Plus précisément, je réalisais ces ateliers en petit comité, avec les personnes maîtrisant parfaitement le sujet à éclaircir. Ces ateliers m'ont permis d'avancer et de clarifier beaucoup de notions que je ne maîtrisais pas à cause de mon arrivée récente dans l'entreprise.

Par exemple, lorsque j'ai commencé à travailler sur l'application OMS au début du mois de juillet, j'ai organisé, avec l'aide Thomas, un atelier visant à collecter un maximum d'informations sur les besoins des principales personnes qui allaient utiliser l'application. Sarah, responsable du service client, nous a donc éclairer sur sa manière de travailler ainsi que sur les fonctionnalités qu'elle utilise sur PrestaShop. À partir de cela, nous avons donc pu réfléchir avec elle à la maquette finale de l'application OMS. L'objectif étant que celle-ci soit la plus ergonomique possible afin de faciliter au service client la migration de PrestaShop vers Zoranne, tout en optimisant leur efficacité.

Par ailleurs, j'ai récemment mis en place, avec l'aide de mon tuteur, des réunions hebdomadaires entre les membres de mon équipe. L'objectif de ces réunions est d'aborder des sujets plus généraux que ceux abordés lors des daily meetings. En effet, le but est d'échanger sur des sujets concernant l'entreprise. Aussi, dans une démarche d'amélioration continue, ces réunions nous permettent d'échanger sur ce que l'on peut améliorer dans notre façon de faire.

Je trouve ces réunions hebdomadaires importantes car elles nous permettent de nous remettre en question et, ainsi, de nous améliorer. Par conséquent, je considère que j'aurais dû les mettre en place plus tôt.

Enfin, en septembre 2021, nous sommes partis en séminaire d'entreprise au Portugal. Lors de ce séminaire, différents ateliers, constitués de petits groupes, étaient organisés. Ainsi, j'ai pu, avec mes collègues, échanger sur diverses interrogations concernant la suite et les possibilités d'évolution de l'entreprise. Aussi, j'ai participé à des activités de team-building qui m'ont permis d'en apprendre plus sur mes collègues et de créer des liens sociaux. Ces liens m'ont été utile pour mieux connaître chacun des membres des équipes métier, et ainsi avoir la meilleure approche possible pour les accompagner au changement depuis PrestaShop vers Zoranne.

À mon sens, ce séminaire a été très bénéfique, notamment pour faciliter la communication avec l'ensemble des membres de l'entreprise. En ce qui me concerne, cela faisait peu de temps que j'avais rejoint l'entreprise et j'ai pu créer des liens et des relations de confiance avec mes collègues, me permettant de leur proposer la meilleure réponse possible à leur besoin avec Zoranne.

En apprenant à mettre en place toutes ces réunions rigoureusement, j'ai pu développer mon organisation. À l'avenir, je saurai mettre en œuvre cette compétence, que ce soit dans la planification d'un projet ou encore dans la mise en place de réunions comme je l'ai fait avec Zoranne.

De plus, la présentation et l'explication des applications que j'ai réalisées aux équipes métier m'ont permis d'apprendre à synthétiser, à être clair et concis ainsi qu'à mieux gérer la pression lorsque je m'exprime en public. Par conséquent, dans le futur, je serai plus à l'aise pour prendre la parole et mener à bien des réunions.

En outre, grâce aux réunions hebdomadaires, j'ai appris à me remettre en question régulièrement afin de m'améliorer. J'ai également appris à prendre du recul sur une situation tant grâce aux réunions hebdomadaires que grâce aux activités de team building réalisées en séminaire. Ainsi, que ce soit dans mes futurs projets ou dans la vie de tous les jours, je pourrai employer ces compétences.

Par ailleurs, j'ai eu l'occasion de mobiliser les participants du projet Zoranne d'autres manières. Je pense notamment aux outils collaboratifs tels que notre messagerie professionnelle Outlook, ou encore Slack, notre plateforme de communication au sein de l'entreprise. En effet, tout au long du projet, j'ai pu échanger sur Slack, que ce soit avec mon équipe ou avec les équipes métier.

J'ai ainsi pu communiquer au quotidien avec mon équipe, tant à l'écrit qu'à l'oral, au sein d'un groupe Slack afin de lever certaines difficultés ou encore pour les informer.

Aussi, j'ai également pu échanger avec les équipes métier pour clarifier certains points qu'ils maîtrisaient et ainsi, leur proposer la meilleure solution possible avec Zoranne. En ce sens, nous avons également communiqué par Slack après les réunions (recettes) où je leur présentais une application Zoranne terminée. Effectivement, les équipes métier pouvaient alors me poser d'éventuelles questions et, de mon côté, je pouvais leur répondre et les informer.

Les échanges avec les membres de mon équipe ont été très enrichissants. En effet, mon tuteur Issam a pu me transmettre son expertise technique. J'ai ainsi appris les bonnes pratiques en termes de développement, d'architecture et de gestion de projet. J'ai également pu en apprendre plus en exploitation système (Ops), c'est-à-dire sur la mise en place de serveurs, en accord avec nos choix architecturaux.

En outre, Thomas a pu m'expliquer l'ensemble des processus existants. Effectivement, c'est vers lui que je me tournais lorsque j'avais besoin de comprendre le fonctionnement des processus initialement en place. Aussi, lors de nos échanges, il a su m'éclairer sur l'objectif et le besoin auquel Zoranne devait répondre.

Ainsi, nous formons une équipe parfaitement complémentaire, Thomas nous fournissant ses connaissances sur les règles métier en place et Issam et moi nous occupons de mettre en place la meilleure solution possible au besoin.

Par ailleurs, j'ai su pratiquer une écoute active lors de nos discussions. Je considère que l'écoute active est essentielle afin de s'assurer une compréhension mutuelle, c'est-à-dire de bien comprendre et de bien se faire comprendre lors d'un échange. Ainsi, j'ai appris à écouter attentivement et à reformuler, notamment en donnant des exemples, les propos de mon interlocuteur. Aussi, j'ai compris qu'il était important de poser des questions afin d'assurer une bonne compréhension.

Lors de mon arrivée dans l'entreprise, j'ai dû engranger un certain nombre de connaissances, notamment sur les processus et le fonctionnement de notre application monolithique PrestaShop. Ainsi, à chaque fois que l'on m'expliquait quelque chose, je pratiquais une écoute active afin d'être certain d'avoir bien intégré la notion que l'on me présentait.

À l'avenir, je serai donc capable de pratiquer l'écoute active, notamment dans le but de gérer d'éventuels conflits ou encore afin d'être à l'écoute et de bien comprendre une équipe.

## Conclusion

Zoranne a pour moi été un projet très enrichissant. Le fait d'effectuer la migration d'une application monolithique vers plusieurs applications distinctes a, pour moi, été très formateur. En effet, ce projet m'a permis de découvrir la gestion d'un projet complexe en mettant en pratique les connaissances que j'ai acquises au cours de ma formation chez Aston.

En outre, en tant que développeur, j'ai pu consolider mes acquis en développement web ainsi qu'en la maîtrise des outils collaboratifs tels que Slack et Outlook pour la communication, ou encore Jira et Bitbucket pour la gestion du projet.

Pour répondre au mieux au besoin, j'ai commencé par l'analyse du besoin. On m'a ainsi expliqué les enjeux du projet. J'ai rapidement compris, notamment grâce à ma formation, la pertinence du changement de l'architecture du projet, dans un souci de scalabilité et d'évolution.

Ensuite, j'ai pu participer et faire entendre mon point de vue sur les décisions du projet. Effectivement, j'ai pu échanger avec mon équipe sur l'architecture du projet afin de m'assurer qu'elle convienne parfaitement à la solution à mettre en place.

Aussi, j'ai fait le choix des technologies à utiliser lors du développement. J'ai ainsi démontré la pertinence d'utiliser les frameworks Symfony et Angular.

J'ai également soutenu l'idée d'utiliser une méthode agile. En effet, la méthode Kanban que nous avons choisi d'adopter est pour moi la plus adaptée à notre projet, notamment pour la flexibilité qu'elle fournit, ainsi que grâce à la visualisation que permettent les tableaux de Kanban.

Après avoir fait ces choix, j'ai pu procéder à l'analyse de l'existant en manipulant PrestaShop ainsi qu'en communiquant avec mes collègues, qui ont su m'éclairer sur le fonctionnement initial des processus. Cette analyse a pour moi été essentielle afin de réaliser la migration vers Zoranne.

Puis, j'ai appris à identifier les risques auxquels nous pouvions être confrontés au cours de l'élaboration du projet, afin de les évaluer et de les hiérarchiser. Après cela, avec l'aide de mon équipe, j'ai mis en place des actions pour éviter ces risques.

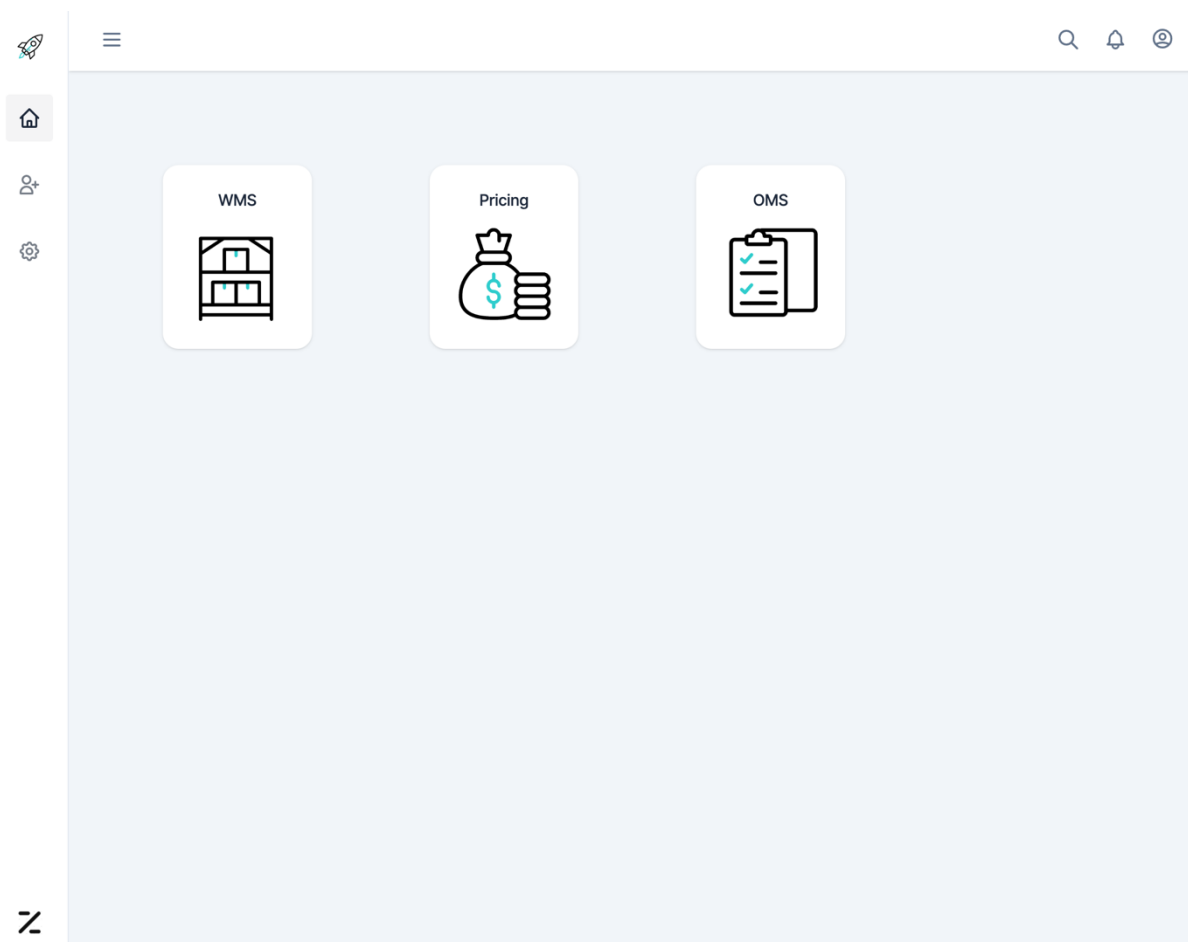
Une fois le travail d'analyse effectué, j'ai pu commencer le développement du projet. En parallèle du développement, j'ai pu participer au management du projet en appliquant la méthode Kanban avec rigueur. Aussi j'ai su communiquer régulièrement avec mon équipe afin de m'assurer que je ne m'écarterais pas du besoin. Aussi, j'ai appris à tester les développements réalisés afin d'assurer la qualité de ceux-ci.

Les interactions avec mon tuteur, Issam, ainsi qu'avec Thomas, m'ont permis d'évoluer et de mener à bien le projet. Effectivement, j'ai compris que la communication était essentielle pour répondre correctement à un besoin ainsi que pour favoriser l'avancement d'un tel projet. Ainsi, j'ai appris à échanger, notamment en organisant des réunions, avec mon équipe et avec les équipes métier afin de pouvoir mettre en place les solutions les plus adaptées au besoin.

Aussi, le fait d'avoir participé activement à un tel projet m'a permis d'acquérir de l'expérience en tant que chef de projet ainsi que dans l'architecture logicielle. Ainsi, si je décide d'emprunter cette voie dans le futur, je pourrai mettre en œuvre cette expérience et les compétences que j'ai acquises.

Ravi de cette expérience professionnelle chez Allstore, je poursuivrai, dès la fin de ma formation, l'aventure auprès d'eux en tant que développeur full stack. Je serai ainsi en mesure de mobiliser l'ensemble des compétences que j'ai acquises cette année dans le cadre du projet Zoranne. En effet, ce dernier doit encore être enrichi afin de proposer la meilleure expérience possible aux équipes métier ainsi que pour améliorer les processus existants.

## Annexes



*Launcher permettant d'accéder aux applications de Zoranne*

# Stock

Reset

Recherche

Réf. produit

Tous

Tous

Tous

Toutes

Référence Produit	Qté physique	Qté calculée	Qté à recevoir	Qté disponible ↓	Délai	Entrepôt	Emplacement	Compagnie	Date de mise à jour
B0303675	2432	2432	0	2432	3	HEPPNER	HEPPNER	BC France	20-07-2022 11:24
B0303674	2100	1680	0	1680	3	HEPPNER	HEPPNER	BC France	20-07-2022 11:24
1264GW	3215	1607	0	1607	3	GILLEMAN	GILLEMAN	BC France	20-07-2022 11:24
B0303449	1356	1356	0	1356	3	HEPPNER	HEPPNER	BC France	20-07-2022 11:24
528	1203	1203	0	1203	3	HEPPNER	HEPPNER	BC France	20-07-2022 11:24
B0303674-4	0	1050	0	1050	3	HEPPNER	HEPPNER	BC France	20-07-2022 11:24
1250GB-1	1953	976	0	976	3	GILLEMAN	GILLEMAN	BC France	20-07-2022 11:24
C-TE062	1168	934	0	934	3	HEPPNER	HEPPNER	BC France	20-07-2022 11:24
1264GA	1809	904	0	904	3	GILLEMAN	GILLEMAN	BC France	20-07-2022 11:24
TV01N	764	764	0	764	3	HEPPNER	HEPPNER	BC France	20-07-2022 11:26
1250NG-1	1525	762	0	762	3	GILLEMAN	GILLEMAN	BC France	20-07-2022 11:24
B0303499	893	714	0	714	3	HEPPNER	HEPPNER	BC France	20-07-2022 11:24
WT018T	683	683	0	683	3	HEPPNER	HEPPNER	BC France	20-07-2022 11:26
529	655	655	0	655	3	HEPPNER	HEPPNER	BC France	20-07-2022 11:24
WT017T	652	652	0	652	3	HEPPNER	HEPPNER	BC France	20-07-2022 11:26
1225NG	1074	644	0	644	3	GILLEMAN	GILLEMAN	BC France	20-07-2022 11:24
BQ554	764	611	0	611	3	HEPPNER	HEPPNER	BC France	20-07-2022 11:24

Lignes par page

20

1 - 20 sur 10960

<<

<

>

>>

Capture d'écran des stocks de l'application WMS de Zoranne



$\neq$ 

57

€

Prix

Réf. produit

Référence Produit	Presta ES	SF FR	SF ES
08164100B	26.9 € +	26.9 € + <small>29.9 €</small>	26.9 € + <small>29.9 €</small>
08164100P	29.9 € +	20 € - <small>29.9 €</small>	29.9 € + <small>29.9 €</small>
08164150B	29.9 € +	29.9 € + <small>29.9 €</small>	29.9 € + <small>29.9 €</small>
08164150P	29.9 € +	29.9 € + <small>29.9 €</small>	29.9 € + <small>29.9 €</small>
08166100	19.9 € +	19.9 € + <small>21.9 €</small>	19.9 € + <small>21.9 €</small>
08166150	19.9 € +	19.9 € + <small>21.9 €</small>	19.9 € + <small>21.9 €</small>
1_1272BW	0 € +	-	-
1_1272GA	0 € +	-	-
1_1272GW	0 € +	-	-

Lignes par page 20 1 - 20 sur 5074 < >

Prix

30 jours

18 € ▲ 20 € ▼ 16 €

Marge ▲  
9.52 € 47.60%

Stock : 0

Prix de base \*

29.90 €

Écotaxe \*

0.00 €

Mis à jour  
01-01-2022 00:00






OPÉRATIONS COMMERCIALES



Historique

Promo	Réduction	Date de début	Date de fin	Statut
Prix spécifique 01/01/2022 à 0:00 au 31/12/2500 à 0:00	20.00 € ▼	3/8/2022, 17:41:14 📅	1/1/2200, 00:00:00 📅	🟡

Mettre à jour

*Capture d'écran du détail d'un prix de l'application Pricing de Zoranne*





# Commandes

Client

Nouvelle commande

Reset

du

min

au


max

ID

Tous

Tous

Créée le	ID commande	Limite d'expédition	Canal	Statut	Détails	Qté	Montant
29/07/2022 à 09:17	2	1	SF FR	Païement accepté	Salon de jardin en teck LOMBOK - table ovale extensible - 6 places Ref : SETTEC02	1	300 €
29/07/2022 à 09:30	3	1	SF FR	Païement accepté	Salon de jardin en teck LOMBOK - table ovale extensible - 6 places Ref : SETTEC02	1	200 €
				Païement accepté	Salon de jardin en teck LOMBOK - table ovale extensible - 6 places Ref : SETTEC02	1	300 €
				Païement accepté	Chaise Salon de jardin alu extensible en textilène gris Ref : 1264GA	2	200 €
28/07/2022 à 12:12	1	1		Païement accepté	Chaise Salon de jardin alu extensible en textilène gris Ref : 1264GA	1	100 €



Lignes par page 20 1 - 3 sur 3 |< < > >|

Capture d'écran de la liste des commandes de l'application OMS de Zoranne