

Mémoire du projet de fin d'études

Pour l'obtention du Diplôme d'Ingénieur d'État en Télécommunications et
Technologies de l'Information.

FILIERE : Advanced Software Engineering for Digital Services (A.S.E.D.S)

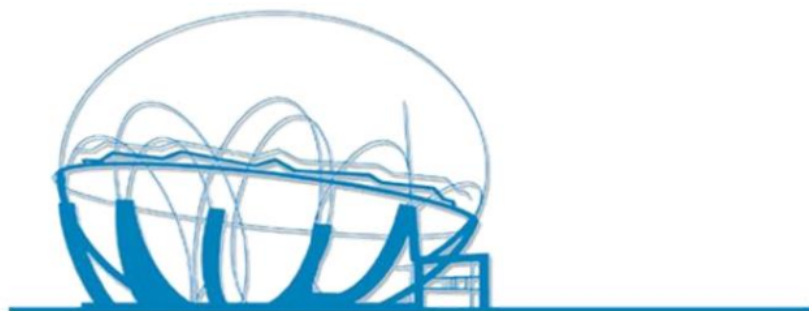
Optimisation d'une Plateforme E-commerce : Intégration de Payconiq

Réalisé par :

Mme. TAOUFIQ Fatima

Soutenu le 19 Septembre 2024, devant les membres de jury :

Pr. MARGHOUBI Rabia	INPT	- Encadrante interne
Pr. HAFIDDI Hatim	INPT	- Examineur
Pr. RADGUI Amina	INPT	- Examinatrice
Mme. ELJABARI Sara	SQLI	- Encadrante externe
M. JIRARI Adil	SQLI	- Encadrant externe



AGENCE NATIONAL DE RÉGLEMENTATION DES TÉLÉCOMMUNICATIONS

INSTITUT NATIONAL DES POSTES ET TÉLÉCOMMUNICATIONS

Année universitaire : 2023/2024

Dédicace

“

إلى الغالية على قلبي، إلى التي اختارها المولى فغابت عنا وبات رضاها يلازمنا،
إلى التي حملتنا في قلبها قبل حضنها،
إلى التي تكبدت عنا تربيته وسهرت على راحتنا،
إلى التي احبتنا حباً لا مشروطاً وآمنت بنا دائماً حتى في اللحظات التي لم نؤمن فيها بأنفسنا،
إلى التي تعجز اللغة بكل مفرداتها على أن توافيها حقها،
إلى أعظم امرأة في حياتي،
إلى روحك الطاهرة أُمِّي أهدى ثمرة نجاحي اليوم .

إلى قدوتي في الحياة، رمز الأمل و الصمود أُمِّي،
إلى من شق الطريق أمامنا لنسلكه بأمان،
إلى الذي ضحى بكل شيء من أجلنا ،
إلى من وضع سعادتنا أولوية حياته أسعدك الله وحفظك من كل شر.

إلى من قال فيهم المولى عز وجل "نشذ عضدك بأخيك"،
شكراً لأنكم نعم السند،
شكراً لأنكم جسدتكم كلمة الأخوة بكل ماتحملة من معنى،
شكراً لأنكم انتمم دربي وعززتم ثقتي بنفسي،
شكراً على نصائحكم وارشاداتكم،
شكراً لأنكم جزءاً لا يتجزأ من نجاحي.

إلى الأصدقاء الذين تقاسمنا معهم لحظات كثيرة،
إلى من جعلوا الحياة أسهل بوجودهم،
إلى الذين دعمونا في القرح قبل الفرع، شكراً لوجودكم.

إلى كل الأساتذة الفاضلين،
إلى كل من علمني حرفاً لكم مني فائق التقدير والاحترام،
كل الشكر لمجهوداتكم، دعمكم وعطاءكم.

وفي إطار من لم يشكر الناس لم يشكر الله، شكراً لكل من ساهم في هذا العمل المتواضع من قريب أو بعيد سواء بمعلومة، نصيحة، توجيه، دعم نفسي او كلمة طيبة، جزاكم الله عن هذا ألف خير.

”

Remerciements

Avant tout, je remercie Allah, le Tout-Puissant, de m'avoir accordé le courage et la patience nécessaires pour mener ce travail à terme dans des très bonnes conditions.

Je souhaite exprimer ma gratitude à toutes les personnes qui, par leur soutien ou par leur simple présence, ont contribué à rendre mon travail à la fois instructif, bénéfique et agréable.

Je souhaite exprimer ma profonde gratitude à mon encadrante, **Pr. Rabiaa MARGOUBI**, dont les connaissances, le savoir-faire et les précieuses orientations ont grandement facilité mon travail. Je lui suis reconnaissante pour ses conseils avisés, son suivi attentif, ainsi que pour la fierté et l'ambition que j'ai pu développer grâce à son soutien intensif et son aide précieuse.

Je souhaite également remercier chaleureusement mes encadrants externes, **M. JIRARI Adil** et **Mme ELJABARI Sara**, pour leur confiance, leur collaboration et leur soutien tout au long de mon stage. Leur supervision, leurs recommandations éclairées, leurs orientations précieuses et leur rigueur m'ont été d'une aide précieuse et ont facilité mon intégration. Je tiens aussi à adresser mes sincères remerciements aux membres de l'entreprise SQLI Maroc pour l'expérience enrichissante et captivante qu'ils m'ont offerte pendant ces mois de stage parmi eux.

Je tiens à exprimer ma sincère gratitude à mes examinateurs, **Pr. HAFIDDI Hatim** et **Mme RADGUI Amina**, pour leur évaluation et leurs précieux retours. Je remercie également l'ensemble du corps enseignant de l'Institut National des Postes et Télécommunications pour leur soutien, leur expertise et leur engagement, qui ont grandement contribué à l'enrichissement de mon parcours académique.

Table des figures

1.1	Logo de SQLI [1]	2
1.2	Chiffre Clés de SQLI	3
1.3	Clients du SQLI [1]	4
1.4	Départements de SQLI	5
1.5	Structure de l'équipe Seasonal Event	8
1.6	Structure de l'équipe Cart Checkout & Payment	8
1.7	Page d'accueil de Confluence	11
1.8	Backlog du projet sur Azure DevOps	11
1.9	Clendrier des formations	13
2.1	Processus du paiement	18
3.1	Architecture globale de l'application E-commerce	22
3.2	Architecture de backend de l'application E-commerce	24
3.3	Architecture de backend de l'application E-commerce	26
3.4	Architecture de backend de l'application E-commerce	26
3.5	Architecture Applicative de l'application E-commerce	28

Liste des tableaux

1.1	Fiche technique de SQLI Maroc	4
-----	---	---

Table des matières

Dédicaces	i
Remerciements	ii
Table des figures	iii
Liste des tableaux	iv
Table des matières	vi
1 Contexte général du projet	1
1.1 Présentation de l'entreprise d'accueil SQLI	2
1.1.1 Groupe SQLI	2
1.1.2 SQLI Maroc	4
1.2 Présentation du projet	6
1.2.1 Cadre du projet et problématique	6
1.2.2 Objectifs du projet	6
1.3 Conduite de projet	7
1.3.1 Présentation des Équipes du Projet	7
1.3.2 Méthodologie de travail : Scrum	9
1.3.3 Capitalisation et suivi	10
1.3.4 Planification du projet	12
2 Analyse et spécification des besoins	15
2.1 Étude de l'existant	16
2.1.1 Méthodes de paiement actuelles	16
2.1.2 Particularités du marché belge	16
2.1.3 Justification de l'intégration de Bancontact by Payconiq	17

2.1.4	Architecture du processus de paiement actuel	17
2.2	Etude fonctionnelle et non fonctionnelle	19
2.2.1	Exigences fonctionnelles	19
2.2.2	Exigences non-fonctionnelles	20
3	Conception de la solution	21
3.1	Architecture Globale	22
3.2	Architecture Backend	23
3.3	Environnement de livraison et test	25
3.3.1	Architecture interne	26
3.3.2	Architecture externe	27
3.4	Architecture applicative	27
4	Implémentation et Validation	30
	Conclusion	32
	Références	32

Chapitre 1

Contexte général du projet

Ce chapitre situe mon projet de fin d'études dans son environnement organisationnel et contextuel. Il commence par une présentation de l'organisme d'accueil, SQLI Maroc. Ensuite, il détaille la problématique ayant conduit à la réalisation de ce projet ainsi que les objectifs visés. Enfin, la méthodologie adoptée pour mener à bien le projet est abordée.

1.1 Présentation de l'entreprise d'accueil SQLI

Cette section initiale met en lumière le Groupe SQLI en mettant l'accent sur ses activités clés, son chiffre d'affaires ainsi que ses clients. Ensuite, l'accent sera mis sur SQLI Maroc, en mettant en avant ses valeurs fondamentales.

1.1.1 Groupe SQLI



FIGURE 1.1 – Logo de SQLI [1]

SQLI est une entreprise européenne de services numériques fondée en 1990 par Jean Rouveyrol et Alain Lefebvre. Elle se spécialise dans la conception, le développement et le déploiement de solutions digitales visant à créer des expériences unifiées [1]. Avec un effectif de 2400 collaborateurs répartis dans 13 pays, SQLI bénéficie d'une présence internationale solide.

Le succès de SQLI Digital Experience repose sur des valeurs fondamentales telles que la créativité, l'engagement et l'audace visionnaire. Ces valeurs imprègnent chaque aspect de l'entreprise, permettant de repousser les frontières de l'innovation et de concevoir des expériences digitales uniques et captivantes. [2]

1.1.1.1 Activités du groupe

Le groupe SQLI propose une gamme étendue de services pour accompagner les entreprises dans leur transformation numérique. Il inclut l'e-commerce, créant et optimisant des plateformes de vente en ligne performantes. Il offre également des plateformes d'expérience, conçues pour offrir des interactions utilisateur exceptionnelles. En matière de technologie et de transformation, il aide les entreprises à moderniser leurs infrastructures et leurs processus. Ses services de data et insights permettent d'exploiter les données de manière stratégique, tandis que son expertise en marketing digital et design améliore la visibilité et l'attrait des marques. Enfin, son conseil digital guide les entreprises dans l'élaboration et la mise en œuvre de leur stratégie numérique globale, assurant ainsi une transformation digitale réussie. [1]

1.1.1.2 Chiffres Clés du groupe

Les chiffres clés suivants présentent la situation actuelle de SQLI :

- Fort de 33 ans d'expérience et d'innovation, SQLI fonde son développement sur une expertise technologique de pointe et une politique de veille intensive.
- SQLI emploie plus de 2400 collaborateurs répartis dans 13 pays, notamment la France, l'Angleterre, la Suède, les Pays-Bas, l'Espagne, l'Allemagne, la Belgique, le Luxembourg, la Suisse et le Maroc.
- En 2022, le groupe SQLI a atteint un chiffre d'affaires de 251,2 millions de dollars. Ce succès est le résultat d'une offre bien alignée sur les attentes du marché et d'une reprise progressive de la demande de services informatiques.

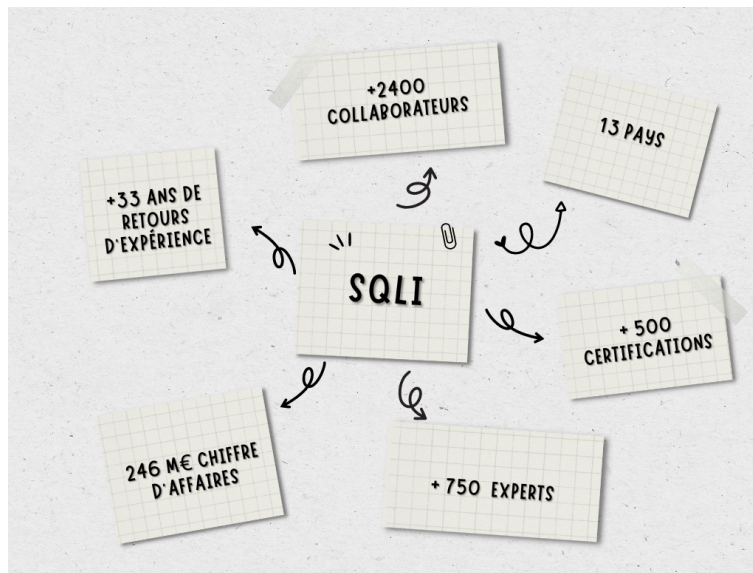


FIGURE 1.2 – Chiffre Clés de SQLI

1.1.1.3 Clients du groupe

SQLI collabore avec une vaste gamme de clients provenant de divers secteurs, y compris l'automobile, la distribution, la banque et l'assurance, le luxe et la mode, la santé, l'industrie et l'énergie, ainsi que les télécommunications. Les grandes entreprises internationales et les organisations locales font appel à SQLI pour ses solutions digitales innovantes, allant de l'optimisation des plateformes d'e-commerce à la transformation numérique des services financiers, en passant par la création d'expériences utilisateur uniques pour les marques de luxe et la digitalisation des processus industriels. Grâce à sa capacité à répondre aux besoins spécifiques de chaque secteur, SQLI bâtit des partenariats solides et durables avec ses clients (voir *Figure 1.3*)

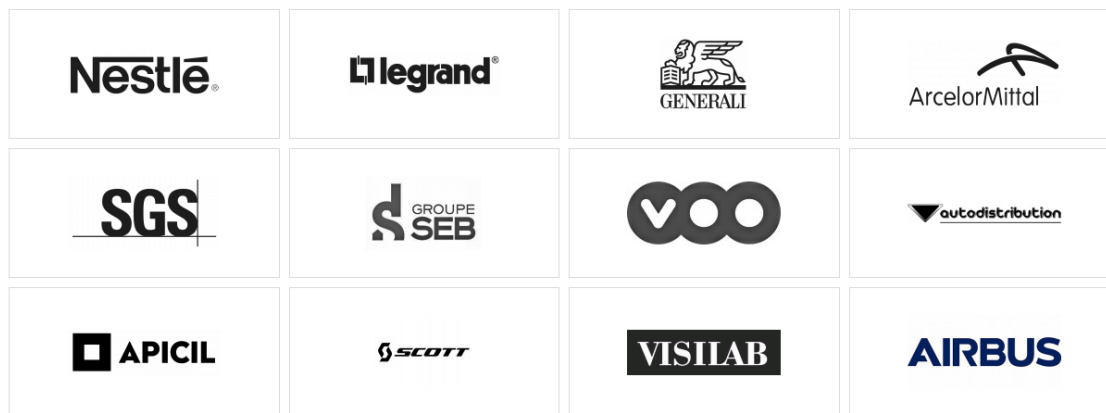


FIGURE 1.3 – Clients du SQLI [1]

1.1.2 SQLI Maroc

SQLI Maroc, créée en 2003 à Rabat par Eric Chanal, représente le centre de Delivery et d'Innovation du Groupe SQLI. Bénéficiant d'une solide expertise et d'une grande expérience, l'entreprise est présente sur trois sites stratégiques : Rabat, où j'ai eu l'opportunité d'effectuer notre stage PFE, Oujda et Casablanca. Le tableau suivant (*Tableau 1.1*) représente sa fiche technique :

Dénomination sociale	SQLI Digital Experience
Année de fondation	2003
Fondateur	Eric Chanal
Siège social	Rabat, Maroc
Activité	Conseil en systèmes et logiciels informatiques.
Effectif des employés	Plus de 900 collaborateurs.
Sites d'implantation	Rabat, Oujda et Casablanca.

TABLE 1.1 – Fiche technique de SQLI Maroc

SQLI Maroc comprend principalement deux structures essentielles, présentées dans la *Figure 1.4*, à savoir :

1. **SQLI WAX INTERACTIVE** : accompagne les clients dans leur transition vers la digitalisation afin de renforcer leur positionnement sur le marché. Cette entité intervient principalement sur le plan stratégique en collaborant étroitement avec les clients.
2. **SQLI ENTREPRISE** : Cette entité est chargée de la mise en œuvre des systèmes d'information pour les clients. Elle se compose de plusieurs Business Units spécialisées dans différents domaines :

- **E-commerce/ JAVA EE** : Se focalise sur la création et la mise en place de sites de e-commerce ainsi que sur le développement d'applications utilisant la technologie Java EE.
- **Mobile/Front** : Se spécialise dans le développement d'applications mobiles et l'interface utilisateur (front-end) pour les clients.
- **Microsoft** : S'occupe de la réalisation d'applications basées sur les technologies Microsoft.
- **Agency** : Joue un rôle transversal en assurant la conception de l'interface utilisateur (front-end) pour toutes les autres Business Units.
- **Delivery** : Se charge de la gestion des livraisons et des recettes auprès des clients.

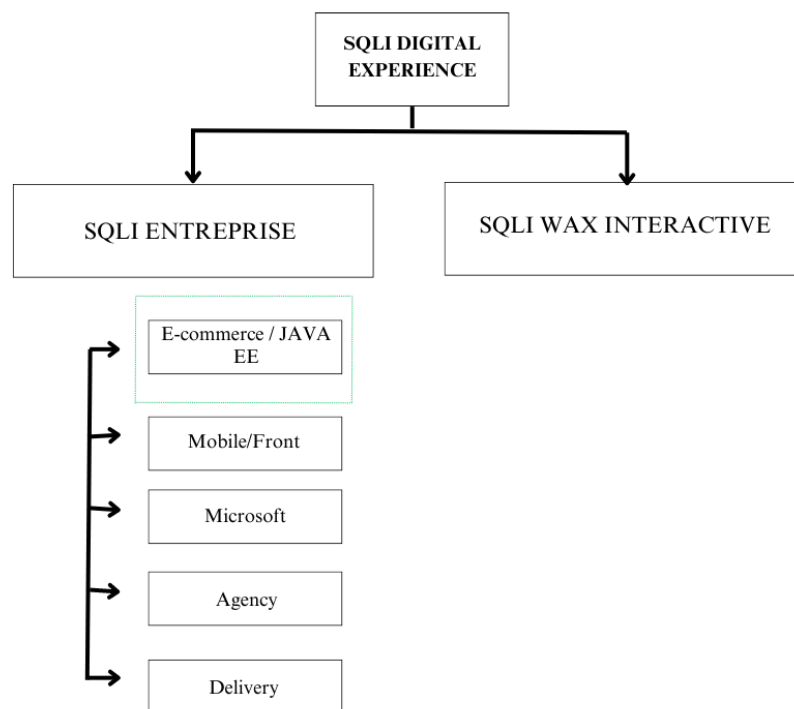


FIGURE 1.4 – Départements de SQLI

Mon stage de fin d'études s'est déroulé dans le département Java JEE, qui regroupe plusieurs projets destinés à de grandes entreprises clientes.

1.2 Présentation du projet

1.2.1 Cadre du projet et problématique

Dans le cadre d'un projet e-commerce pour un client, l'objectif est d'améliorer et d'optimiser sa plateforme actuelle. Il est indispensable de mettre à jour régulièrement cette plateforme, qui joue un rôle crucial dans les activités commerciales en ligne du client, afin de maintenir sa compétitivité et de répondre aux exigences du marché.

Pour cette amélioration, le travail inclut la correction de divers bugs qui affectent la performance et la fiabilité du système. La résolution de ces bugs est cruciale pour garantir une expérience utilisateur fluide et sans interruptions.

En parallèle, l'intégration de nouvelles fonctionnalités est nécessaire pour enrichir l'offre de la plateforme. L'un des changements majeurs est l'intégration de la méthode de paiement Payconiq, destinée spécifiquement au marché belge. Différents défis se posent lors de cette intégration, tels que la compatibilité avec l'architecture existante, la gestion des dépendances et l'assurance que cette nouvelle fonctionnalité ne provoque pas de régressions ou de nouveaux bugs.

L'enjeu majeur consiste donc à corriger les bugs existants tout en intégrant Payconiq de manière efficace, en maintenant la stabilité et la performance globale de la plateforme.

1.2.2 Objectifs du projet

Dans le cadre de ce projet, je participerai activement aux diverses activités de l'équipe, contribuant à la fois au développement des fonctionnalités demandées par le client et à l'amélioration continue du système. Les objectifs spécifiques de mon intervention sont les suivants :

- **Intégrer un nouveau mode de paiement, Payconiq, pour le marché belge**

Analyser et comprendre l'architecture existante pour intégrer Payconiq, tout en gérant les dépendances et en garantissant la compatibilité avec les autres modules de la plateforme, conformément aux spécificités techniques du marché belge. Cela inclut la configuration, le développement, et des tests rigoureux pour garantir une intégration fluide dans les différents flux de paiement existants.

- **Assurer les livraisons dans différents environnements (DEV, INTx, UAT, PRD)**

Garantir le bon fonctionnement du code dans chacun de ces environnements, conformément aux exigences de stabilité et de performance. Cela comprend des tests approfondis pour s'assurer que les nouvelles fonctionnalités et intégrations sont stables et opérationnelles avant la mise en production.

- **Respecter les meilleures pratiques, les normes, et l'architecture du projet**

Assurer la cohérence du code et la stabilité du projet en adhérant aux normes et pratiques de développement établies. Cela inclut le respect des principes d'architecture définis, l'application des bonnes pratiques de codage, et la proposition de solutions conformes aux standards en vigueur au sein de l'équipe.

- **Analyser et corriger les bugs détectés**

Assurer la maintenance corrective du système en identifiant, analysant, et résolvant les bugs remontés par les équipes ou découverts lors des tests, afin de minimiser leur impact sur le fonctionnement global de la plateforme.

1.3 Conduite de projet

1.3.1 Présentation des Équipes du Projet

Après ma période de formation, j'ai intégré une première équipe en tant que stagiaire backend. Cette équipe était responsable des aspects liés à la recherche, aux lunettes et à la mode pour le projet Chanel. À mon arrivée, l'équipe se concentrait sur le Plan 99.5, visant à analyser les bugs, nettoyer les logs, corriger les erreurs et refactoriser le code. La structure de cette équipe est illustrée dans la *Figure 1.5*.

Souhaitant approfondir mes compétences dans des domaines complémentaires, j'ai ensuite rejoint une autre équipe, chargée de l'intégration des nouvelles méthodes de paiement pour les différents marchés du client. La structure de cette équipe est illustrée dans la *Figure 1.9*.

Ces deux équipes font partie d'un projet plus vaste comprenant 15 équipes fonctionnelles différentes. Chaque équipe est composée d'un Scrum Master, d'un expert technique, d'un Product Owner, d'un développeur frontend et de deux responsables qualité, favorisant ainsi le développement d'une solution robuste et performante.

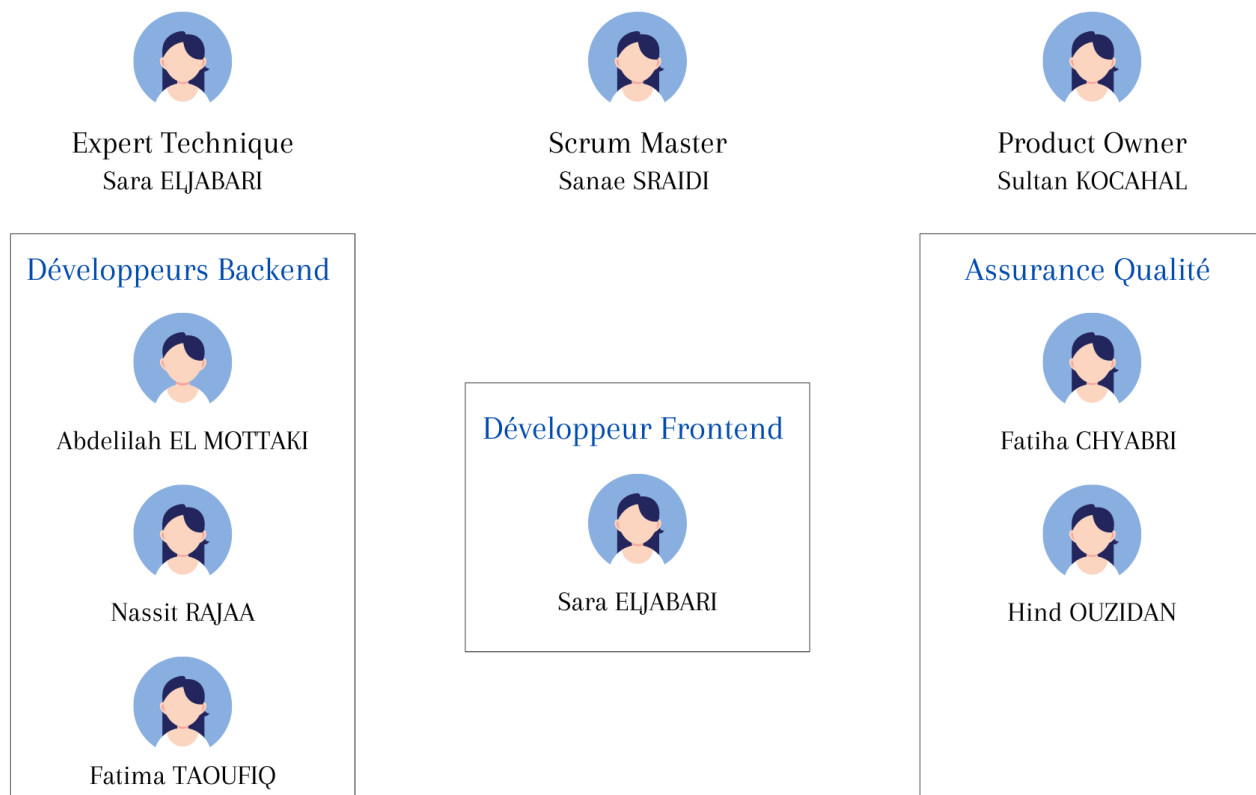


FIGURE 1.5 – Structure de l'équipe Seasonal Event

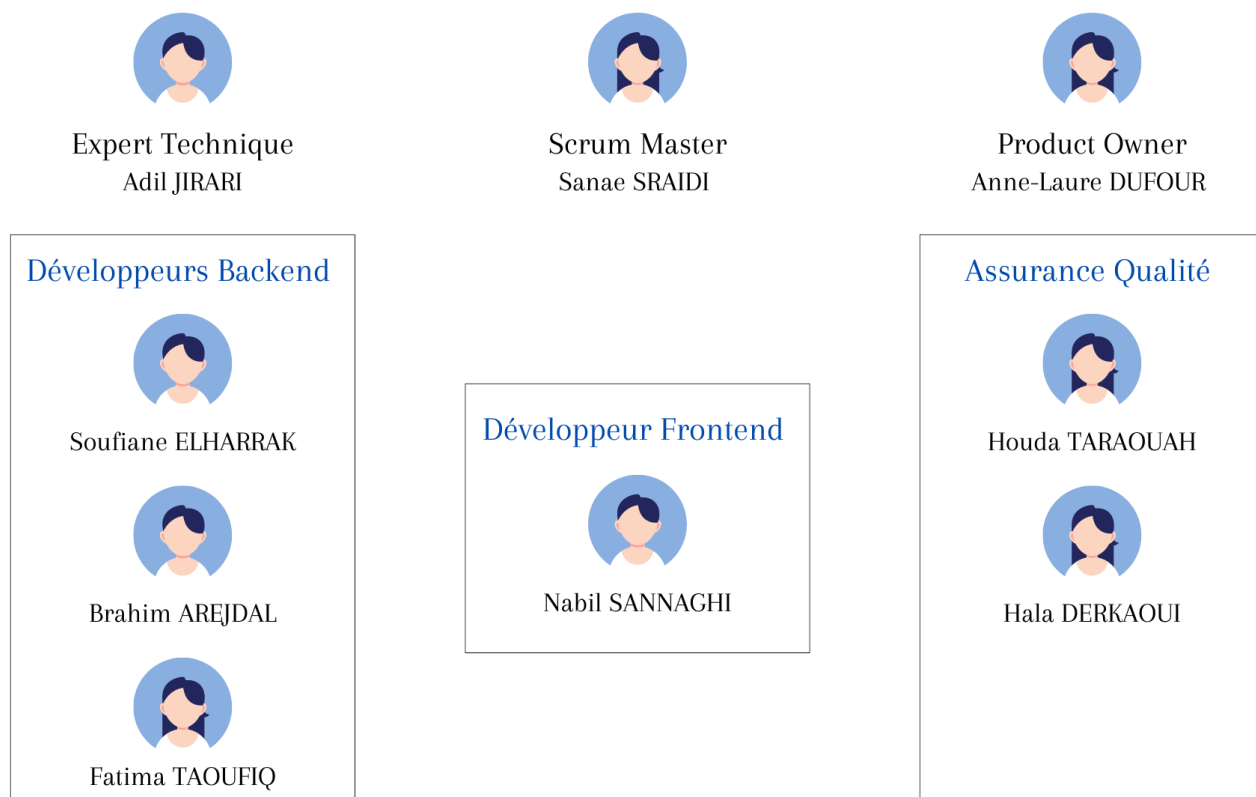


FIGURE 1.6 – Structure de l'équipe Cart Checkout & Payment

1.3.2 Méthodologie de travail : Scrum

Pour assurer une collaboration efficace au sein de l'équipe, nous avons opté pour la méthodologie Scrum, qui se caractérise par une approche itérative et incrémentale. Scrum nous permet de diviser le travail en sprints, des cycles de développement courts et cadencés, généralement de deux à quatre semaines. À la fin de chaque sprint, une version potentiellement livrable du produit est présentée, ce qui favorise la flexibilité et l'adaptation aux changements. Grâce à cette méthode, nous pouvons rester réactifs et ajuster rapidement notre travail en fonction des évolutions des besoins métiers. En intégrant les retours d'expérience du client à chaque itération, nous assurons une satisfaction optimale de ses attentes. Les rôles bien définis, tels que le Scrum Master, le Product Owner, et l'équipe de développement, garantissent une communication claire et une responsabilité partagée. Cette approche nous permet d'être efficaces tout en maintenant un rythme de travail soutenu et structuré.

- ***Planification du sprint (sprint planning)***

Avant de débiter chaque sprint, nous tenons une réunion de Sprint Planning. Cette réunion a pour objectif de définir les tâches prioritaires à accomplir au cours du sprint à venir. L'équipe, en collaboration avec le Product Owner, examine le backlog du produit pour identifier les éléments les plus critiques à traiter. Durant cette session, nous discutons des exigences, des objectifs du sprint, et nous évaluons la charge de travail nécessaire pour chaque tâche. Cette planification permet à l'équipe de se concentrer sur un ensemble de fonctionnalités claires et réalisables, tout en s'assurant que les ressources sont allouées de manière optimale. Ainsi, chacun sait précisément sur quoi se concentrer, ce qui contribue à une exécution efficace et coordonnée du sprint.

- ***Mêlée quotidienne (Daily Meeting)***

Chaque jour, nous tenons une réunion appelée Daily Meeting, d'une durée de 15 minutes chaque matin. Lors de cette réunion, chaque membre de l'équipe partage ce qu'il a accompli la veille, ce qu'il prévoit de faire aujourd'hui, et signale s'il rencontre des problèmes ou des blocages. Cette réunion permet à l'équipe de rester synchronisée et d'identifier rapidement les obstacles éventuels, favorisant ainsi une meilleure collaboration et une résolution rapide des problèmes.

- ***Revue de sprint (Sprint Review)***

La revue de sprint se tient à la fin de chaque sprint pour présenter et évaluer les fonctionnalités développées. L'équipe démontre le travail accompli aux parties prenantes, recueille leurs retours et discute des ajustements nécessaires. Ce moment est crucial pour valider les résultats, s'assurer qu'ils répondent aux attentes et planifier les prochaines étapes en fonction des feedbacks reçus.

- ***Rétrospective de sprint***

À la fin de chaque sprint, nous tenons une rétrospective de sprint, un moment privilégié pour discuter des succès et des aspects à améliorer dans notre travail. Pour détendre l'atmosphère et réduire le stress, nous intégrons également de petits jeux qui permettent de sortir de la routine, de mieux connaître les membres de l'équipe, et de renforcer notre cohésion. Ces activités, combinées à des discussions constructives, nous aident à identifier les points à optimiser et à faire évoluer nos pratiques de manière continue.

- ***Préparation du backlog (Backlog Refinement)***

Nous tenons régulièrement des réunions de préparation du backlog, également appelées sessions d'affinement du backlog. Lors de ces réunions, l'équipe Scrum se réunit pour examiner et ajuster les éléments du backlog du produit. Nous clarifions les exigences, estimons les efforts nécessaires pour chaque tâche, et réorganisons les priorités en fonction des retours des parties prenantes et des évolutions du projet. Cette réunion est essentielle pour maintenir le backlog bien structuré et aligné avec les objectifs du projet, ce qui facilite une planification plus efficace des sprints et assure une gestion optimale des priorités.

1.3.3 Capitalisation et suivi

Pour assurer un suivi efficace du projet, l'équipe utilise des outils de gestion de projet tels que Azure DevOps et Confluence.

- ***Confluence***

Confluence est utilisé pour la documentation et l'archivage des informations du projet, offrant une base de connaissances centralisée accessible à tous les membres de l'équipe. Cette plateforme permet de créer, organiser et maintenir des documents essentiels tels que les spécifications techniques, les guides de processus et les comptes rendus de réunions. En centralisant ces informations dans Confluence, nous facilitons la réutilisation des connaissances et assurons leur préservation au-delà de la durée du projet. Cette approche améliore la transparence, soutient la collaboration et garantit que les informations cruciales sont facilement accessibles pour toute l'équipe.



FIGURE 1.7 – Page d'accueil de Confluence

• Azure DevOps

Azure DevOps nous permet de gérer efficacement les tâches à réaliser et de suivre en temps réel l'avancement de chaque membre de l'équipe. En plus de ces fonctionnalités de gestion de projet, Azure DevOps sert également de dépôt centralisé pour le code et les artefacts du projet, facilitant ainsi la collaboration et l'intégration continue.

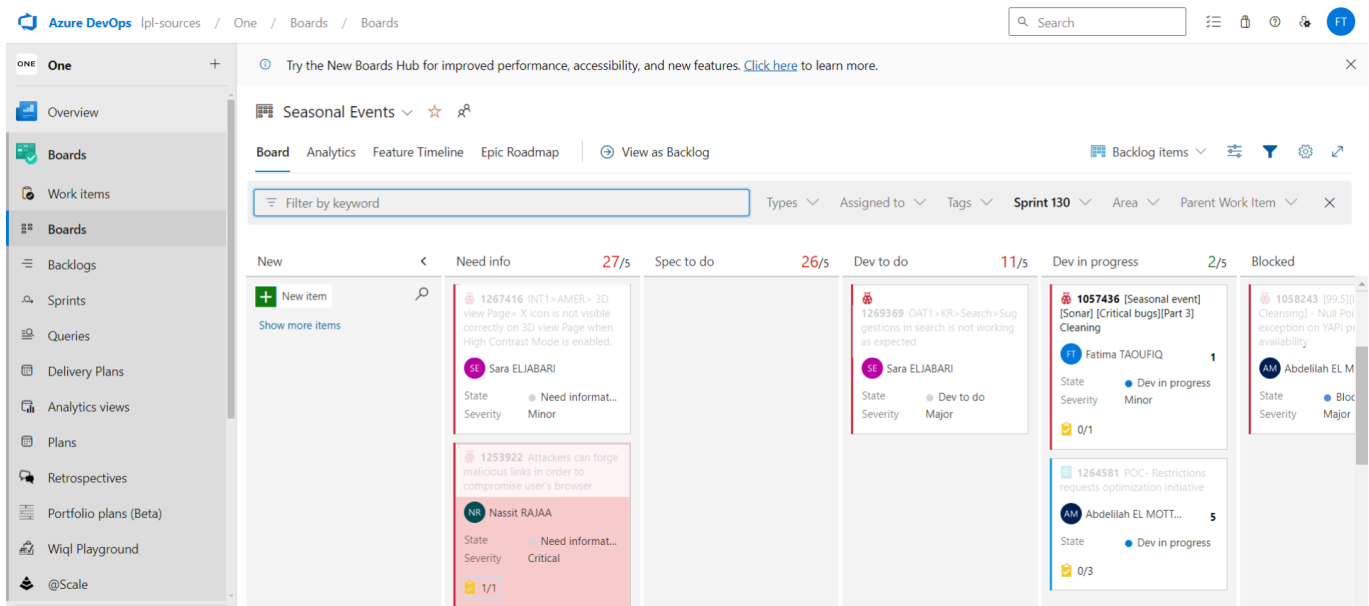


FIGURE 1.8 – Backlog du projet sur Azure DevOps

- **Microsoft Teams**

Microsoft Teams est un outil essentiel pour la communication et la collaboration au sein du projet. Il nous permet de maintenir une communication fluide entre les managers, les chefs de projet, et les membres de l'équipe. Lorsqu'il est nécessaire de contacter quelqu'un pour obtenir des informations, poser des questions ou résoudre des problèmes, j'utilise Teams pour envoyer des messages instantanés, organiser des réunions virtuelles ou partager des documents. Cet outil facilite également la coordination des tâches et le suivi des progrès en centralisant les échanges et les informations pertinentes, ce qui contribue à une gestion de projet efficace et une meilleure intégration des membres de l'équipe.

1.3.4 Planification du projet

1.3.4.1 Intégration

Notre expérience a débuté par une journée d'intégration organisée par l'équipe RH, en collaboration avec les managers de chaque département. Cette journée a été l'occasion pour nous de découvrir en profondeur la structure de l'organisation, de comprendre les rôles et responsabilités de chacun, et de faire connaissance avec les collègues avec qui nous allions collaborer. Cette intégration nous a permis de nous familiariser rapidement avec notre environnement de travail et de tisser des liens avec les membres de l'équipe, facilitant ainsi notre adaptation et notre engagement au sein de l'entreprise.

1.3.4.2 Formation

La phase de formation, cruciale dans le cadre de notre stage, nous a permis d'acquérir les bases nécessaires pour réussir dans notre domaine. Cette formation, d'une durée de deux mois, a été animée par des professionnels expérimentés qui ont partagé avec nous leur savoir-faire et leur expertise. Grâce à ces sessions, nous avons pu renforcer nos compétences techniques et notre compréhension des bonnes pratiques du métier. Voici les principales formations qui ont constitué notre parcours :

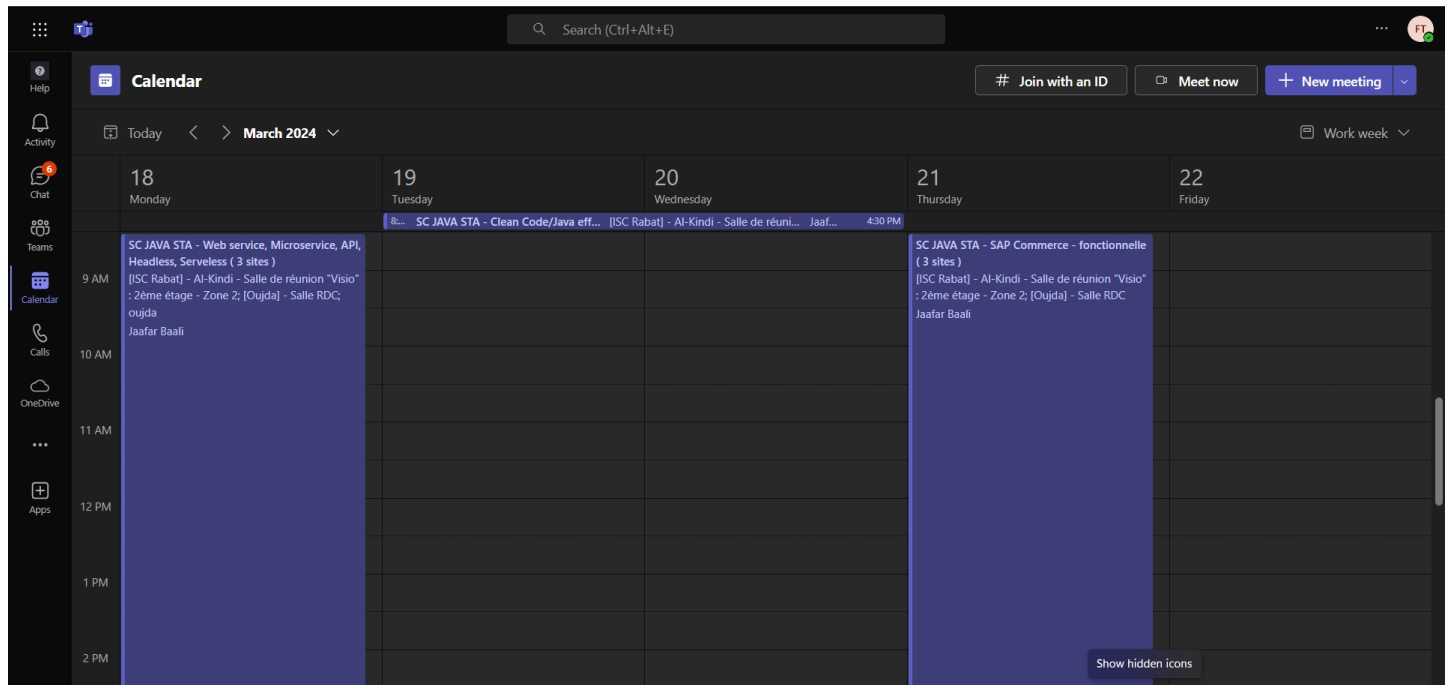


FIGURE 1.9 – Clendrier des formations

- **Formations Spring** : Durant cette formation, nous avons exploré les concepts fondamentaux du framework Spring, tels que l'inversion de contrôle (IoC) et Spring MVC. Ces notions clés ont été renforcées par des exercices pratiques pour mieux comprendre leur utilisation dans le développement d'applications Java.
- **Formations SAP Commerce (Hybris)** : Pendant cette formation, nous avons commencé par installer la plateforme SAP Commerce. Nous avons ensuite exploré les aspects fonctionnels, comme l'utilisation des outils d'administration HAC, HMC, et des différents Cockpits. En parallèle, nous avons approfondi des sujets techniques, tels que la configuration de CronJobs et la gestion des workflows.
- **Formations Clean Code** : Cette formation, inspirée des principes de Robert C. Martin, nous a appris les bonnes pratiques pour écrire du code propre, lisible et maintenable. Nous avons appliqué ces concepts à travers des exercices pratiques, axés sur la structuration du code et les techniques de refactoring.
- **Formations Software Craftsmanship** : La formation en Software Craftsmanship s'est concentrée sur l'excellence technique et la qualité du code. Nous avons étudié des pratiques comme le pair programming, la revue de code, et l'écriture de tests automatisés, renforçant ainsi notre capacité à produire du code robuste et maintenable.
- **Formations ICD Skills & Agilité** : Cette formation nous a initiés aux compétences interpersonnelles essentielles et aux méthodes agiles. Nous avons étudié des méthodologies comme Scrum

et Kanban, et nous avons mis en pratique des techniques de gestion de projet agile pour améliorer notre efficacité et notre adaptabilité en équipe.

1.3.4.3 Intégration du projet

J'ai été affecté au projet Chanel ONE en tant qu'ingénieur backend, avec pour mission principale l'intégration de la méthode de paiement Payconiq pour le marché belge. En plus de cette tâche, j'ai également travaillé sur la correction de bugs, en particulier ceux liés à Sonar, en mettant à profit ma solide connaissance de cet outil pour améliorer la qualité du code et réaliser les refactorings nécessaires. Mon intégration dans le projet a débuté par une phase de onboarding de 15 jours, au cours de laquelle j'ai installé l'environnement de développement et configuré le projet. Par la suite, j'ai participé à plusieurs réunions avec le manager et le Scrum Master pour clarifier le processus de travail et les spécificités du projet Chanel ONE, ce qui m'a permis de garantir une bonne compréhension du projet et une intégration fluide au sein de l'équipe.

Conclusion

Au cours de ce chapitre, nous avons mis l'accent sur le périmètre de notre projet. Nous avons éclairé la méthodologie et le planning suivis pour mener ce projet. Nous entamerons dans le chapitre suivant la phase d'analyse et spécification du système à développer au cours de laquelle nous comprenons en profondeur les besoins utilisateurs et construisons ainsi un système qui y répond.

Chapitre 2

Analyse et spécification des besoins

Ce chapitre présente l'analyse de l'existant et la spécification des besoins pour l'intégration de Banccontact by Payconiq. Nous examinerons les méthodes de paiement actuelles, les spécificités du marché belge, l'architecture existante, ainsi que les besoins fonctionnels et non fonctionnels.

2.1 Étude de l'existant

2.1.1 Méthodes de paiement actuelles

Le site e-commerce de la marque propose actuellement une gamme diversifiée de méthodes de paiement reconnues mondialement, comprenant :

- Visa
- MasterCard
- PayPal
- Klarna Pay Now
- Klarna Pay Later
- Chanel Gift Card

Bien que ces options répondent efficacement aux besoins d'une clientèle internationale, elles ne tiennent pas compte des spécificités locales de certains marchés clés, en particulier celui de la Belgique.

2.1.2 Particularités du marché belge

En Belgique, Bancontact s'est imposé comme l'une des méthodes de paiement privilégiées. Initialement conçu comme un système de paiement par carte de débit national, Bancontact est devenu un élément incontournable du paysage financier belge. Cette solution offre aux consommateurs belges la possibilité d'effectuer des paiements directs depuis leur compte bancaire, que ce soit en magasin, en ligne ou via une application mobile. Face à l'évolution rapide des technologies et des attentes des consommateurs, Bancontact a réalisé une fusion stratégique avec Payconiq, une solution de paiement mobile innovante. Cette alliance a donné naissance à Bancontact by Payconiq, offrant aux utilisateurs belges une solution de paiement intégrée couvrant à la fois les transactions par carte et les paiements mobiles via une application dédiée. L'adoption massive de cette solution en Belgique en fait un élément incontournable pour tout e-commerce aspirant à s'implanter solidement sur ce marché. Les chiffres parlent d'eux-mêmes : en 2023, près de 2 millions de Belges ont utilisé Payconiq pour leurs paiements mobiles, soulignant l'importance cruciale de cette méthode dans l'écosystème des paiements locaux.

2.1.3 Justification de l'intégration de Bancontact by Payconiq

L'intégration de Bancontact by Payconiq sur notre plateforme e-commerce présente plusieurs avantages stratégiques majeurs, particulièrement pour conquérir et fidéliser la clientèle belge :

- **Adoption généralisée** : Avec une base d'environ 2 millions d'utilisateurs Payconiq et une forte pénétration de Bancontact dans les habitudes de paiement quotidiennes, cette solution est profondément ancrée dans le comportement des consommateurs belges.
- **Simplicité et ergonomie** : L'application "Payconiq by Bancontact" offre une expérience de paiement fluide et intuitive, reposant sur un simple scan de QR code, ce qui optimise considérablement le parcours client.
- **Sécurité renforcée** : La synergie entre les systèmes Bancontact et Payconiq garantit un niveau de sécurité optimal pour les transactions, s'appuyant sur des protocoles de sécurité robustes et éprouvés.
- **Interopérabilité bancaire** : Le support étendu de Payconiq par les principales institutions bancaires belges facilite son adoption et renforce la commodité pour les clients, en centralisant leurs opérations financières.
- **Essor des paiements mobiles** : Face à la croissance exponentielle des paiements mobiles en Belgique, l'intégration de solutions comme Payconiq s'avère cruciale pour capter et fidéliser une clientèle, particulièrement auprès des jeunes générations habituées aux transactions via smartphone.

2.1.4 Architecture du processus de paiement actuel

Le processus de paiement en place, présenté dans la figure 3.5, repose sur l'interaction harmonieuse de plusieurs systèmes interdépendants, chacun jouant un rôle déterminant dans la sécurisation des transactions et l'optimisation du traitement des commandes.

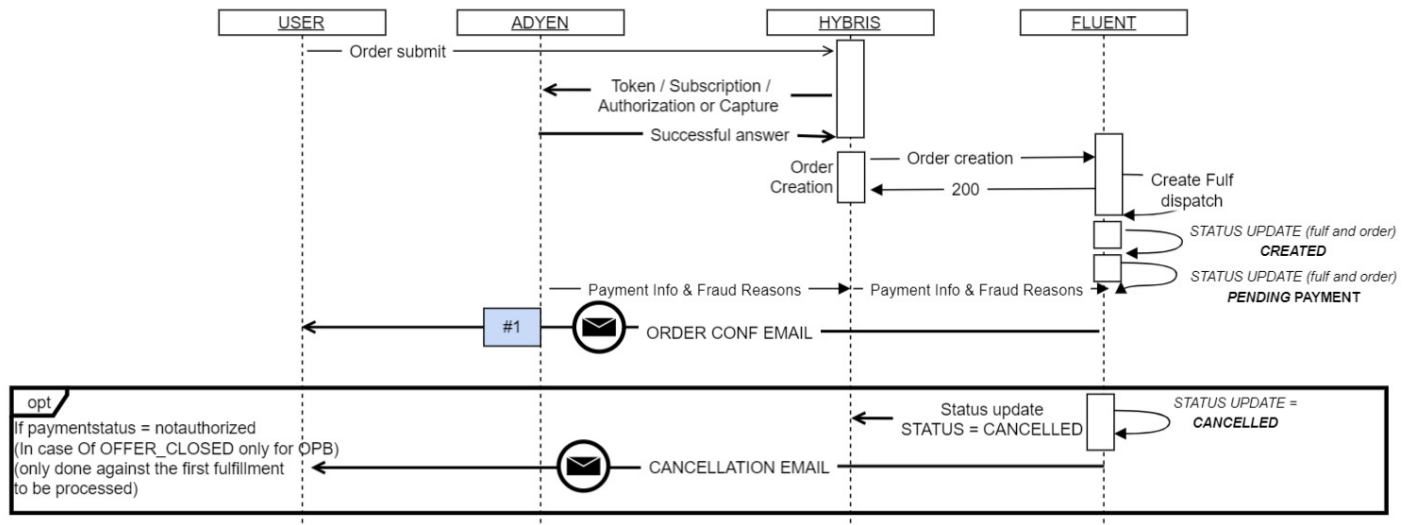


FIGURE 2.1 – Processus du paiement

Les composants clés du système sont les suivants :

- **Hybris** : Plateforme e-commerce centrale, Hybris est responsable de la création des commandes une fois le paiement validé. Après la soumission d'une commande par l'utilisateur, Hybris communique avec Adyen pour le traitement du paiement. Une fois la confirmation reçue, Hybris crée officiellement la commande et notifie Fluent pour la gestion de l'expédition et du suivi.
- **Adyen** : En tant que fournisseur de services de paiement, Adyen gère la transaction financière. Il traite les détails du paiement transmis par Hybris, réalisant des actions telles que l'autorisation, la capture des fonds, ou la gestion des abonnements. Adyen joue un rôle clé dans la sécurisation et la validation des paiements avant la finalisation de la commande.
- **Fluent** : Système de gestion logistique, Fluent est responsable du cycle de vie de la commande après sa création dans Hybris. Il assure le suivi du processus de traitement, y compris la préparation de l'expédition, et met à jour les statuts de la commande (par exemple, "CREATED", "PENDING PAYMENT", "CANCELLED"). Fluent communique également avec Hybris pour informer les utilisateurs de l'état de leur commande.

Pour mieux comprendre le fonctionnement de ces composants, examinons le flux du processus de paiement étape par étape :

1. **Initiation** : Le client valide son panier et soumet sa commande.
2. **Transmission** : Hybris communique les informations de paiement à Adyen.
3. **Traitement** : Adyen exécute la transaction et renvoie une confirmation à Hybris.
4. **Création** : Hybris enregistre la commande et notifie Fluent pour la gestion logistique.

5. **Suivi** : Fluent assure la gestion des statuts de commande et informe Hybris pour la mise à jour du client.

2.2 Etude fonctionnelle et non fonctionnelle

Dans le cadre de l'intégration de Payconiq à la plateforme e-commerce du client, il est essentiel de définir clairement les besoins fonctionnels et non fonctionnels du projet. Cette étude permettra d'identifier les exigences spécifiques liées à cette intégration, assurant ainsi une mise en œuvre réussie et une expérience utilisateur optimale.

2.2.1 Exigences fonctionnelles

2.2.1.1 Identification des fonctionnalités

- **Sélection de Payconiq comme méthode de paiement** : L'option Payconiq doit être disponible lors de la finalisation de la commande, présentée de manière intuitive et clairement distinguée parmi les autres choix de paiement.
- **Génération et affichage du QR Code pour le paiement** : Lors de la confirmation de la commande avec Payconiq, un QR code unique est généré automatiquement et affiché à l'utilisateur pour permettre un paiement sécurisé via l'application Payconiq.
- **Suivi des transactions dans l'historique des commandes** : Les transactions effectuées avec Payconiq sont accessibles dans l'historique des commandes de l'utilisateur, avec une identification claire comprenant la date, le montant et l'état du paiement.
- **Envoi d'un email de confirmation de commande** : Après la finalisation de la commande, un email de confirmation est envoyé à l'utilisateur, incluant les détails de la commande, la méthode de paiement utilisée (Payconiq), le montant total et les informations de suivi de la commande.
- **Message d'erreur en cas d'échec du paiement** : En cas d'échec du paiement avec Payconiq, un message d'erreur explicite informe l'utilisateur du problème, avec des options de paiement alternatives proposées pour finaliser la transaction sans interruption.

2.2.1.2 Diagramme de cas d'utilisation

2.2.1.3 Description textuelle de cas d'utilisation

2.2.1.4 Diagramme de séquence de système

2.2.2 Exigences non-fonctionnelles

Les exigences non-fonctionnelles sont essentielles pour améliorer la qualité des services de la plateforme, notamment en termes de sécurité, de maintenabilité et de disponibilité. Elles garantissent non seulement une expérience utilisateur optimale, mais aussi la pérennité et l'efficacité du système face aux évolutions technologiques et aux besoins des utilisateurs. Parmi ces exigences, on peut citer :

- **Sécurité** : La protection des informations sensibles des utilisateurs est cruciale dans tout système de paiement en ligne. L'intégration de Payconiq requiert une attention particulière à ces aspects pour éviter tout accès non autorisé et toute manipulation malveillante des données.
- **Maintenabilité** : La maintenabilité du système est essentielle pour garantir sa longévité et sa capacité à évoluer. Un code clair, bien structuré, et conforme aux meilleures pratiques de développement facilite l'ajout de nouvelles fonctionnalités et permet une correction rapide des bugs. Une bonne maintenabilité permet aux équipes de développement de diagnostiquer et de résoudre efficacement les problèmes, de déployer des mises à jour sans perturber le fonctionnement du système, et d'adapter rapidement le système aux évolutions technologiques et aux besoins des utilisateurs.
- **Disponibilité** : Le système doit être conçu pour faciliter le diagnostic, la résolution des problèmes, le déploiement des mises à jour, et l'adaptation aux évolutions technologiques et aux besoins des utilisateurs.

Conclusion

La phase d'analyse de l'existant et de spécification des besoins est cruciale pour le succès de notre projet. Nous avons abordé cette phase en examinant l'architecture actuelle, en identifiant les acteurs clés et les cas d'utilisation spécifiques au marché belge. Ensuite nous avons entamé l'analyse des besoins fonctionnels et non fonctionnels. Dans les chapitres suivants, nous aborderons la conception détaillée du projet d'intégration.

Chapitre 3

Conception de la solution

Ce chapitre se concentre sur l'architecture physique et applicative adoptée. Il inclut également une exploration approfondie de la conception, illustrée par des diagrammes techniques de classes et de séquences détaillées, en se basant sur les spécifications établies dans le deuxième chapitre.

3.1 Architecture Globale

L'architecture e-commerce, telle que présentée ici, est volontairement simplifiée pour se concentrer sur les composants essentiels sur lesquels nous avons travaillé. Elle s'appuie sur plusieurs composants clés pour gérer les interactions entre le client, le contenu du site, les commandes et les paiements.

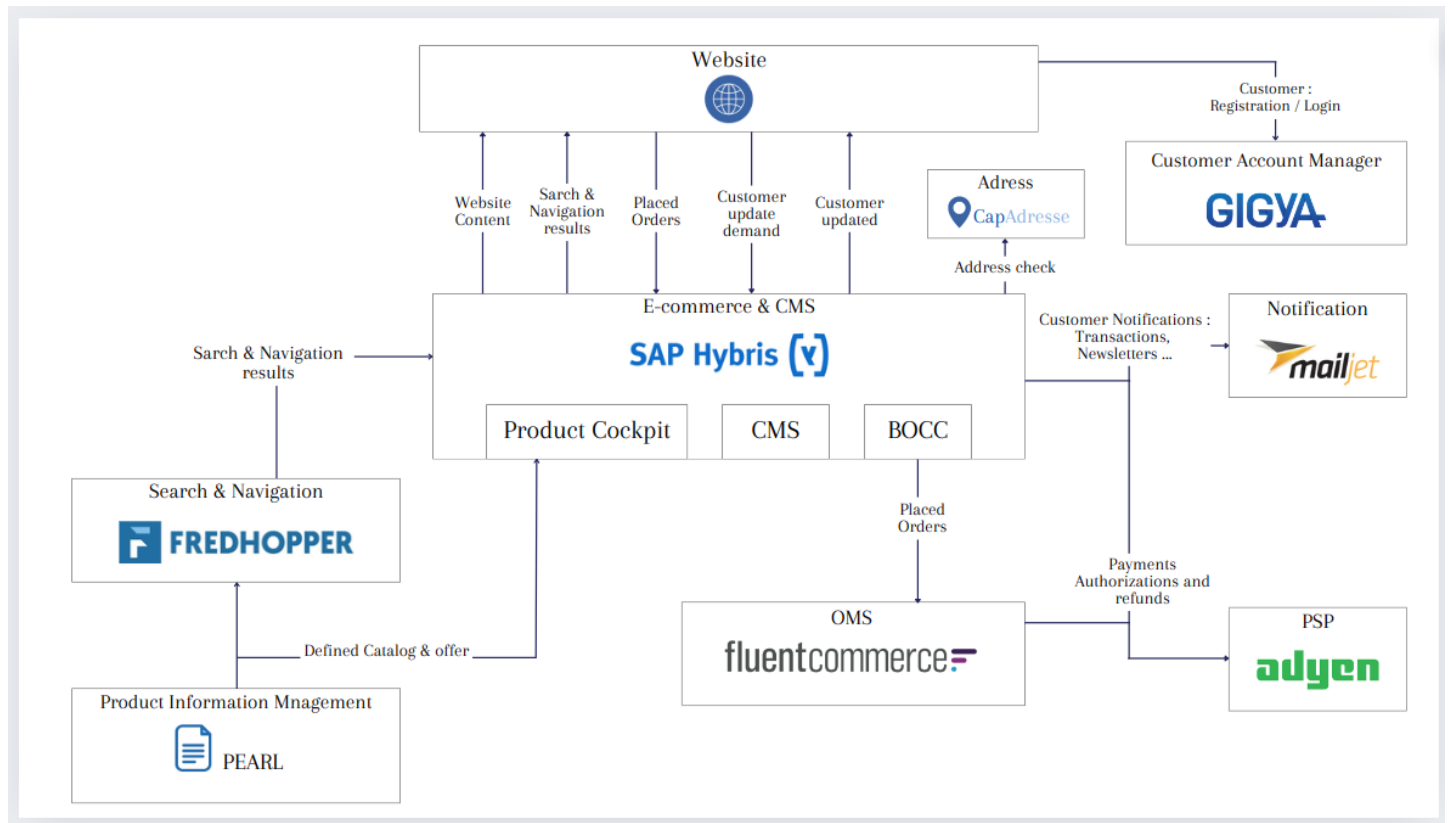


FIGURE 3.1 – Architecture globale de l'application E-commerce

- Le site web permet aux clients de naviguer et de consulter le contenu (produits, offres, etc.). Le contenu est géré par le module CMS de SAP Hybris, qui centralise les informations et les affiche sur le storefront.
- Le moteur de recherche Fredhopper se charge de fournir les résultats de recherche et de navigation aux utilisateurs, en se basant sur les catalogues et les offres définis via PEARL, un système de gestion de l'information produit (PIM).
- Une fois la commande placée par le client via le site web, elle est prise en charge par SAP Hybris. Elle est ensuite transmise à Fluent Commerce, un système de gestion des commandes (OMS), qui suit et gère les différents statuts de la commande, depuis sa validation jusqu'à son expédition.
- CapAdresse est utilisé pour valider l'adresse du client avant la confirmation de la commande, afin de garantir la précision de la livraison.

- Mailjet est utilisé pour envoyer des notifications aux clients concernant les transactions, les newsletters, et d'autres communications, comme la confirmation de commande, d'expédition, ou d'annulation en cas de problème de paiement.
- Gigya s'occupe de la gestion des comptes clients, notamment l'enregistrement et la connexion des utilisateurs sur le site.

Lorsqu'un client effectue un paiement sur le site, une requête est envoyée à Adyen pour obtenir une autorisation. Adyen traite la demande et renvoie une réponse indiquant si le paiement est "autorisé" ou "non autorisé". Si le paiement est refusé, le client est redirigé vers une page d'erreur. En revanche, si le paiement est approuvé, le client est dirigé vers la page de confirmation de commande, et simultanément, la commande est exportée vers Fluent Commerce, le système de gestion des commandes. Fluent Commerce est chargé de gérer le cycle de vie de la commande en attendant la notification d'Adyen confirmant que le paiement a bien été autorisé, car il peut être annulé en cas de fraude ou de problème de sécurité. Si la notification indique que le paiement a été annulé, Fluent communique avec Mailjet pour envoyer un e-mail d'annulation au client. Cependant, si le paiement est confirmé, un e-mail de confirmation de commande est envoyé via Mailjet. Par la suite, une fois la commande prête à être expédiée, Fluent envoie une requête de capture à Adyen. Dès qu'Adyen confirme que la capture a été effectuée avec succès, la commande est expédiée, et Mailjet envoie un e-mail de confirmation de livraison au client. En plus de ces tâches, Fluent Commerce prend également en charge plusieurs aspects logistiques de la commande, tels que la vérification de la disponibilité des produits, la gestion des entrepôts pour l'expédition, ainsi que le suivi du statut de la commande (en attente, en préparation, expédiée, etc.).

3.2 Architecture Backend

Notre solution est basée sur la plate-forme SAP Hybris et suit donc son architecture logicielle présentée dans la figure suivante :

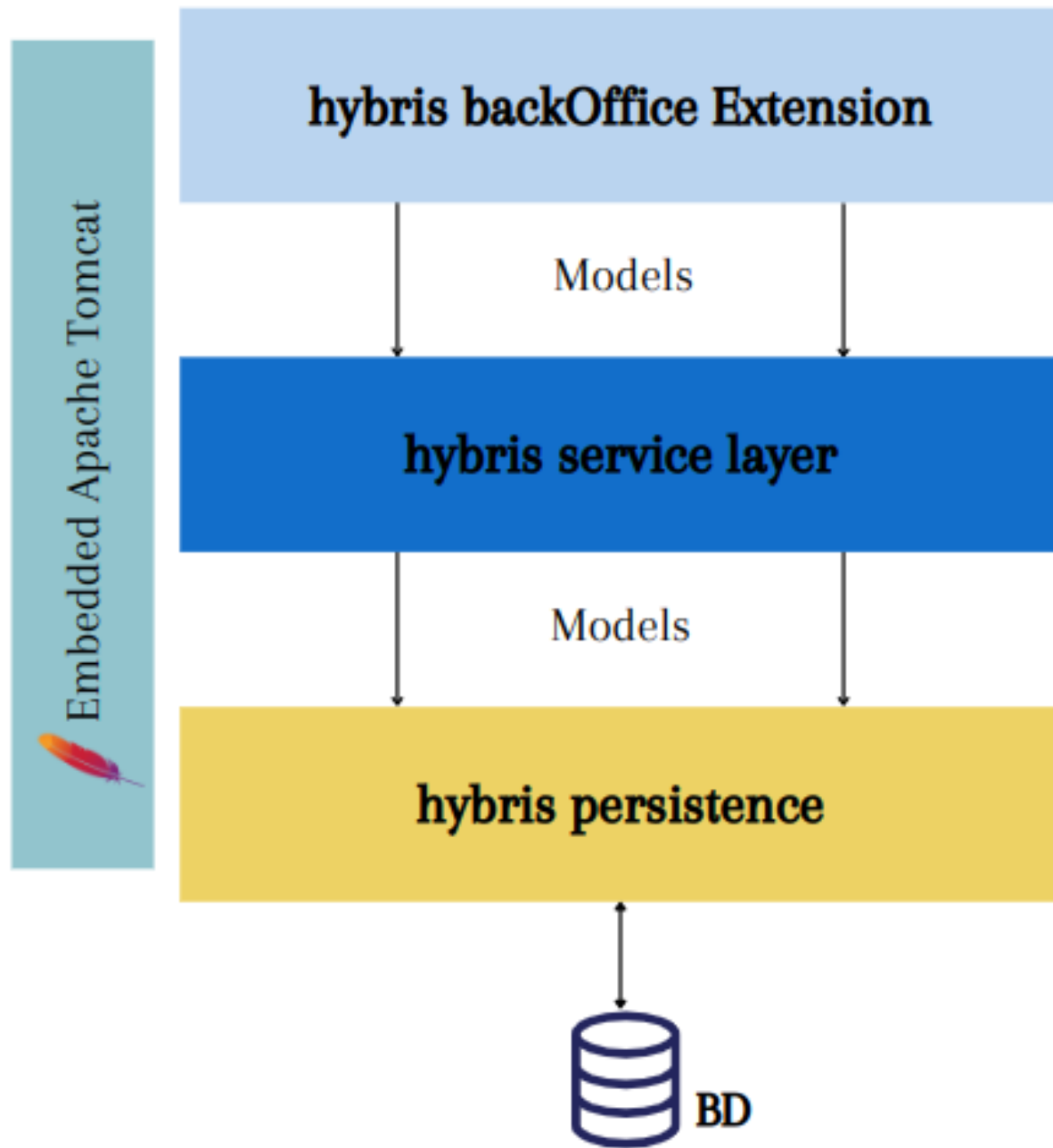


FIGURE 3.2 – Architecture de backend de l'application E-commerce

- **Serveur Apache Tomcat** : La plateforme Hybris utilise Apache Tomcat comme serveur HTTP intégré. Ce serveur d'application est responsable de l'hébergement de l'application Hybris et de la gestion des requêtes HTTP/HTTPS entrantes, assurant ainsi le bon fonctionnement de l'application web.
- **Extension Backoffice de Hybris** : L'extension Backoffice est une composante essentielle de Hybris qui permet aux utilisateurs métiers d'accéder aux fonctionnalités d'administration de contenu. Cela inclut la gestion du catalogue, des catégories, des produits, ainsi que des entités et types du

système. Cette extension fournit une interface graphique conviviale pour les utilisateurs finaux, facilitant la gestion et l'organisation des données. En outre, elle offre aux développeurs la possibilité de créer ou de personnaliser des composants Hybris en fonction des besoins spécifiques de l'entreprise.

- **Service Layer de Hybris** : La couche de service (Service Layer) représente la couche métier de l'application Hybris, où est implémentée la logique de gestion d'entreprise. Elle est constituée d'un ensemble de services qui encapsulent les règles métiers et les processus de l'entreprise. La couche de service communique à la fois avec l'extension Backoffice et la couche de persistance via des modèles, qui sont des représentations des entités de la logique métier. Ces modèles servent d'intermédiaires entre les différentes couches, facilitant ainsi la manipulation des données de manière cohérente et sécurisée.
- **Couche de Persistance de Hybris** : La couche de persistance est le composant qui assure l'interaction entre le Service Layer et la base de données. Elle est responsable de la gestion de toutes les opérations de lecture et d'écriture dans la base de données, garantissant que les données sont stockées de manière efficace et peuvent être récupérées de manière fiable.

3.3 Environnement de livraison et test

Cette section présente processus de déploiement du projet, en détaillant les différents serveurs utilisés. La structure se compose de deux parties principales : interne et externe.

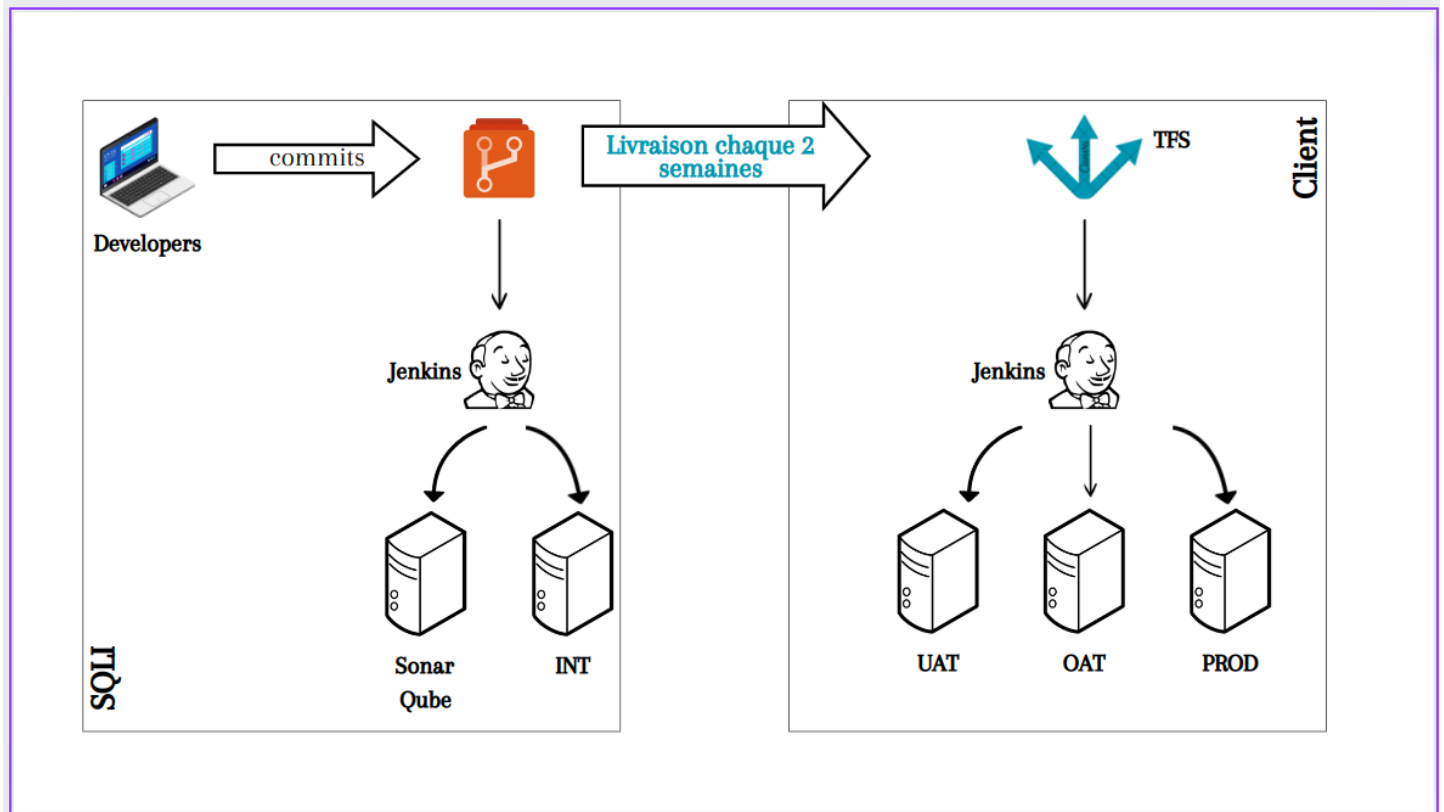


FIGURE 3.3 – Architecture de backend de l'application E-commerce



FIGURE 3.4 – Architecture de backend de l'application E-commerce

3.3.1 Architecture interne

L'architecture interne est implémentée au sein de SQLI et comprend un serveur d'intégration (INT) et un serveur SonarQube pour l'analyse de la qualité du code. Les développeurs effectuent des commits sur le dépôt Azure DevOps, puis un serveur d'intégration continue, Jenkins, récupère les dernières versions depuis ce dépôt pour effectuer une compilation automatique avec ANT. Jenkins lance ensuite une analyse SonarQube pour garantir la qualité du code. Après l'analyse SonarQube, le serveur vérifie que le nombre d'erreurs détectées ne dépasse pas les limites prédéfinies en fonction de la gravité des erreurs et des quotas définis sur le serveur SonarQube. Si ces contrôles sont satisfaits, Jenkins déploie le code sur le serveur d'intégration (INT). Ce processus d'intégration continue se répète tout au long du

sprint de trois semaines.

3.3.2 Architecture externe

L'architecture externe est implémentée dans l'environnement du client et se compose de trois serveurs : UAT (User Acceptance Testing), OAT (Operational Acceptance Testing) et PROD (Production). Toutes les deux semaines, une nouvelle version est livrée dans le système de gestion de versions du client, TFS (Team Foundation Server).

- **Serveur UAT** : Utilisé pour tester les livrables et s'assurer qu'ils répondent aux attentes de l'utilisateur final.
- **Serveur OAT** : Permet de déterminer si les livrables sont opérationnels et prêts à être intégrés dans l'environnement de production.
- **Serveur PROD** : Après les processus de test, une mise en production (MEP) est effectuée tous les trois mois, rendant la version finale disponible pour les utilisateurs finaux.

3.4 Architecture applicative

Dans le cadre de l'utilisation de la solution Hybris, il est conseillé de suivre l'architecture applicative qu'elle propose. Cette architecture est basée sur un modèle n-tiers largement utilisé dans les applications web, permettant une répartition claire des rôles et une meilleure structuration du code. Quatre principaux patrons de conception sont au cœur de cette architecture :

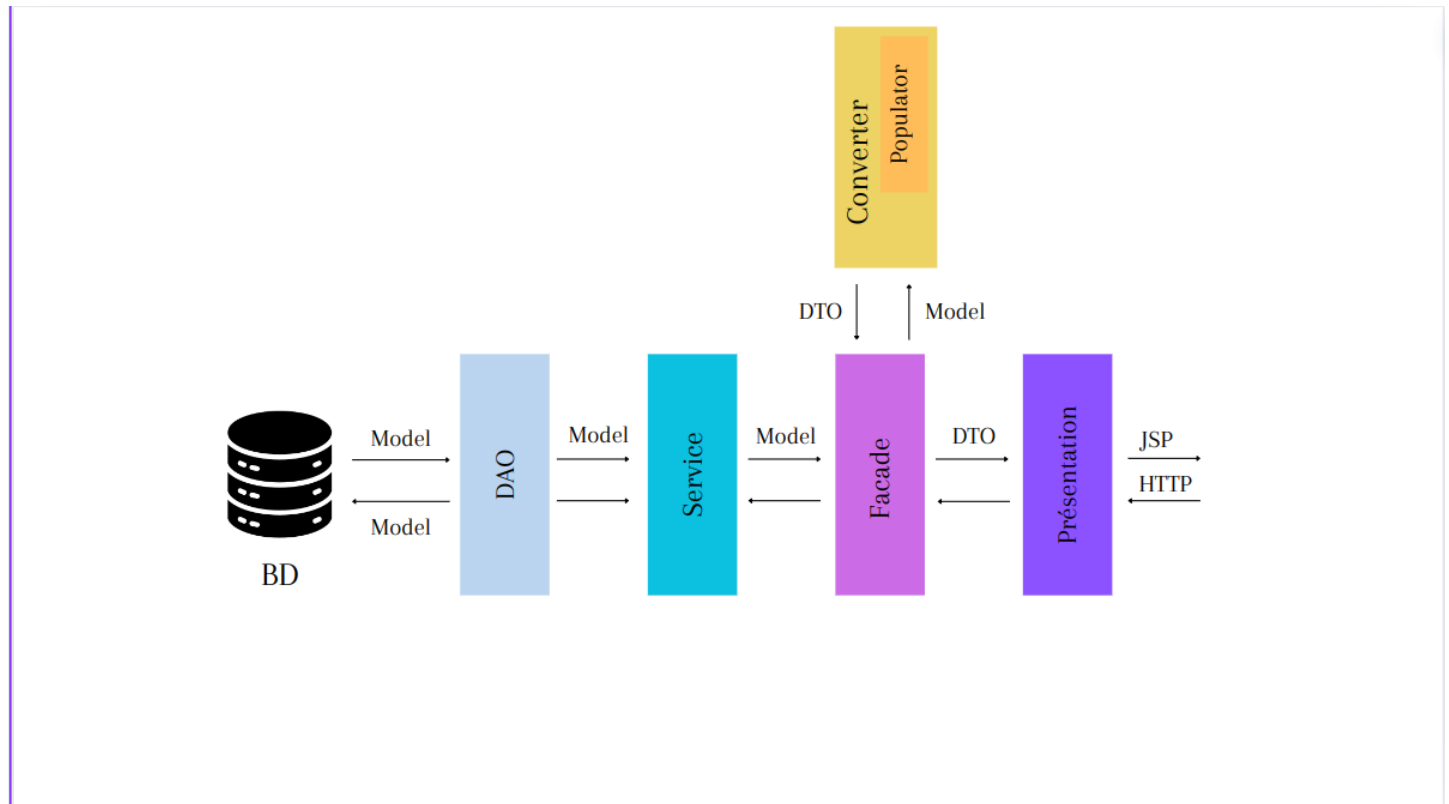


FIGURE 3.5 – Architecture Applicative de l'application E-commerce

- **Modèle MVC (Modèle-Vue-Contrôleur)** : Ce modèle permet de séparer distinctement la présentation, la logique métier et l'accès aux données. Cela garantit une organisation modulaire, rendant l'application plus facile à maintenir et à étendre. Par exemple, dans notre application, la vue pourrait être gérée par des pages JSP ou des composants front-end, tandis que les contrôleurs orchestrent les opérations de la logique métier encapsulée dans les services.
- **Patron de Façade** : Ce patron vise à simplifier l'accès à un système complexe en fournissant une interface unique et uniforme. Il permet d'interagir facilement avec des sous-systèmes tout en masquant leur complexité interne. Dans notre projet, une façade serait utile pour centraliser les interactions entre les services et les DAO, en facilitant ainsi l'appel des contrôleurs.
- **Patron DAO (Data Access Object)** : Ce patron permet d'accéder aux données sans être lié à un SGBD spécifique, en fournissant une abstraction qui rend l'application plus flexible. Il encapsule la logique d'accès aux données, permettant d'intégrer facilement différents SGBD sans modifier le code de l'application. Par exemple, le DAO pour "Produit" ou "Client" peut contenir toute la logique nécessaire pour interagir avec les données associées, indépendamment de la base utilisée.
- **Patron DTO (Data Transfer Object)** : Ce modèle optimise les échanges de données entre différentes couches de l'application en regroupant les informations dans des objets spécifiques. Il permet de réduire la surcharge de transfert de données, améliorant ainsi l'efficacité des commu-

nications entre les couches. Dans notre application, les DTO servent de pont entre la couche de service et la présentation, réduisant les dépendances et simplifiant les tests unitaires.

Adopter cette architecture permet non seulement une meilleure organisation du code, mais aussi d'assurer une évolutivité et une maintenance aisée à long terme.

Conclusion

Ce chapitre a été dédié à l'étude conceptuelle du projet. Après une présentation des architectures adoptées et des divers diagrammes techniques de classes et de séquences, une compréhension approfondie du projet a été acquise. La prochaine étape consistera à aborder l'implémentation et la validation de la solution, sujet du chapitre suivant.

Chapitre 4

Implémentation et Validation

Ce chapitre décrit l'implémentation du travail réalisé. Il commence par une présentation des technologies utilisées, suivie de captures d'écran illustrant les différentes fonctionnalités développées. Ensuite, les tests effectués sont exposés.

Conclusion

Ce chapitre a été consacré à la mise en place de la solution développée. Après une brève introduction des technologies employées, les captures d'écran ont mis en lumière les fonctionnalités réalisées. Les détails des tests entrepris ont ensuite permis de valider l'efficacité de la nouvelle méthode de paiement. Ces étapes ont assuré que la plateforme répond aux exigences fonctionnelles et maintient une performance optimale.

Conclusion générale

Références

[1] *SQLI*, <https://www.sqli.com/ma-fr>

[2] *les valeurs du groupe SQLI*, <https://www.sqli.com/ma-fr/carriere/les-valeurs-sqli>