

Mémoire du projet de fin d'études

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

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

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

Réalisé par :

Mme. TAOUFIQ Fatima

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

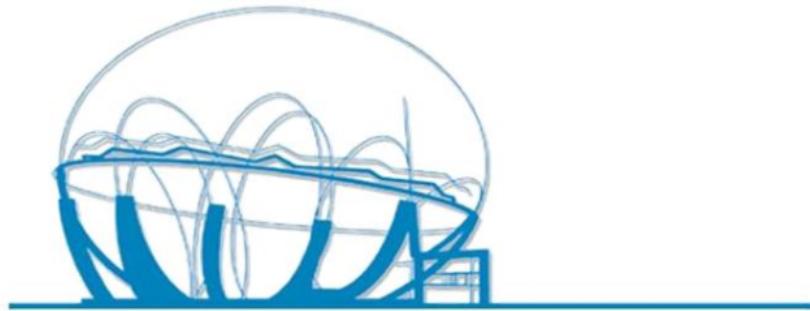
Pr. MARGHOUBI Rabia INPT - Encadrante interne

Pr. HAFIDDI Hatim INPT - Examinateur

Pr. RADGUI Amina INPT - Examinatrice

Mme. ELJABARI Sara SQLI - Encadrante externe

M. JIRARI Adil SQLI - Encadrant externe



AGENCE NATIONALE DE RÉGLEMENTATION DES TÉLÉCOMMUNICATIONS

INSTITUT NATIONAL DES POSTES ET TÉLÉCOMMUNICATIONS

Année universitaire : 2023/2024

Dédicace

٦٦

إلى الفالية على قلبي، إلى التي اختارها المولى فغابت عنا وبات رضاها يلازمنا،
إلى التي حلتنا في قلبها قبل حضنها،
إلى التي تكبدت عناء تربيتنا وسررت على راحتنا،
إلى التي احبتنا هبلاً شرططاً وأمنت بنا دائماً حتى في الاحظات التي لم نؤمن فيها بأنفسنا،
إلى التي تعبر اللغة بكل مفرداتها على أن توفيقها حقها،
إلى أعظم امرأة في حياتي،
إلى روحك الطاهرة أمي أهدي ثمرة نجاهي اليوم .

إلى قدوتي في الحياة، رمز الأصل والصودoric أبي،
إلى من شق الطريق أماننا لسلكه 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 "سند عضدك بأخيك"،
شكراً لأنتم نعم السندي،
شكراً لأنتم جسدتم كلمة الأخوة بكل ما تحمله من معنى،
شكراً لأنتم انترتم دربي وعززتم ثقتي بنفسي،
شكراً على نصائحكم وارشاداتكم،
شكراً لأنتم جزءاً لا يتجزأ من نجاهي.

إلى الأصدقاء الذين تقاسينا معهم لحظات كثيرة،
إلى من جعلوا الحياة أسهل بوجودهم،
إلى الذين دعمنا في الفرج قبل الفرج، شكراللوجودكم.

إلى كل الأساتذة الفاضلين،
إلى كل من عانني حرف لكم مني فائق التقدير والاحترام،
كل الشكر لجهوداتكم، دعيمكم وعطاءكم.

وفي إطار من لم يشكر الناس لم يشكر الله، شكراللكل من ساهم في هذا العمل المتواضع من قريب أو بعيد سواء بعلومة، نصيحة،
توجيه، دعم، نصيحة أو كلمة طيبة، جزاكم الله عن هذا ألف خير.

٩٩

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 MARGHOUBI**, 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, **Mme ELJABARI Sara** et **M. JIRARI Adil**, 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 **Pr. 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.

Résumé

Dans un contexte où le commerce électronique est en constante évolution, les entreprises doivent régulièrement mettre à jour et optimiser leurs plateformes pour rester compétitives. L'accélération des innovations technologiques et les attentes croissantes des consommateurs exigent une adaptation continue pour offrir des expériences utilisateur de haute qualité et répondre aux nouvelles exigences du marché.

Ce rapport présente un aperçu concis de mes contributions lors de mon stage de fin d'études effectué au sein de SQLI Maroc, dans le but d'obtenir le titre d'Ingénieur d'État en Télécommunications et Technologies de l'Information, spécialisation en Ingénierie de développement des services numériques à l'Institut National des Postes et Télécommunications. Ce projet avait pour objectif d'améliorer une plateforme e-commerce existante pour un client de luxe, en intégrant la méthode de paiement Payconiq spécifiquement pour le marché belge, tout en corrigeant divers bugs afin d'optimiser la performance du système.

En prenant en compte la complexité de SAP Hybris, la réalisation de ce projet a nécessité l'utilisation de technologies telles que JEE et Spring pour le développement, ainsi que JUnit pour le testing. L'intégration de Payconiq a été un défi majeur, nécessitant une attention particulière à la compatibilité et à la performance pour garantir une solution adaptée aux besoins spécifiques du marché belge.

Mots clés : E-commerce, Payconiq, SAP Hybris, JUnit, JEE, Spring, Déboggage

Abstract

In a context where e-commerce is constantly evolving, companies must regularly update and optimize their platforms to remain competitive. The acceleration of technological innovations and the growing expectations of consumers demand continuous adaptation to deliver high-quality user experience and meet new market requirements.

This report provides a concise overview of my contributions during my internship at SQLI Maroc, aiming to obtain the title of State Engineer in Telecommunications and Information Technologies, with a specialization in Advanced Software Engineering for Digital Services from the National Institute of Posts and Telecommunications. This project aimed to enhance an existing e-commerce platform for a prestigious client by integrating the Payconiq payment method specifically for the Belgian market, while also fixing various bugs to optimize system performance.

Considering the complexity of SAP Hybris, the realization of this project required the use of technologies such as JEE and Spring for development, as well as JUnit for testing. The integration of Payconiq was a major challenge, requiring particular attention to compatibility and performance to ensure a solution tailored to the specific needs of the Belgian market.

Key Words : E-commerce, Payconiq, SAP Hybris, JUnit, JEE, Spring, Debugging.

ملخص

في ظل التطور المستمر الذي يشهده مجال التجارة الإلكترونية، تجد الشركات نفسها مضطرة لتحديث وتحسين منصاتها بشكل منتظم للحفاظ على قدرتها التنافسية. إذ أن تسارع الابتكارات التكنولوجية وتزايد توقعات المستهلكين يفرضان تكيفاً مستمراً من أجل تقديم تجارب استخدام عالية الجودة وتلبية المطلبات الجديدة للسوق.

لذا فهذا التقرير يقدم نظرة موجزة عن مساهماتي التي قدمتها في شركة SQLI Maroc خلال فترة التدريب النهائي لنيل شهادة مهندس دولة في الاتصالات وتكنولوجيا المعلومات، بتخصص في هندسة البرمجيات المتقدمة للخدمات الرقمية من المعهد الوطني للبريد والمواصلات. كان الهدف من هذا المشروع تحسين منصة تجارة إلكترونية قائمة لأحد العملاء، من خلال دمج نظام الدفع Payconiq خصيصاً للسوق البلجيكي، مع تصحيح مجموعة من الأخطاء البرمجية لتحسين أداء النظام.

ونظراً لتعقيد منصة SAP Hybris ، تطلب تنفيذ هذا المشروع استخدام تقنيات مثل JEE و Spring للتطوير، بالإضافة إلى JUnit للاختبار. شكلت عملية دمج Payconiq تحدياً رئيسياً، حيث استلزمت اهتماماً خاصاً بالتوافق والأداء لضمان تقديم حل يلبي الاحتياجات المحددة للسوق البلجيكي.

. الكلمات المفتاحية : التجارة الإلكترونية، SAP Hybris ، Payconiq ، JEE ، Spring ، JUnit ، debugging ،

Liste des sigles et acronymes

AMER	America
API	Application Programming Interface
APAC	Asia-Pacific, Australia, and Canada
ANT	Another Neat Tool
DAO	Data Access Object
DEV	Development
EMEA	Europe, the Middle East, and Africa
HTML	Hypertext Markup Language
HAC	Hybris Administration Console
HMC	Hybris Management Console
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
INT	Integration
JEE	Java Enterprise Edition
JSE	Java Standard Edition
MEP	Mise En Production
OAT	Operational Acceptance Testing
OMS	Order Management System
PIM	Product Information Management
PRD	Production
PSD2	Payment Services Directive 2
QA	Quality Assurance
SCA	Strong Customer Authentication
SAP	Systems, Applications, and Products
SGBD	Système de Gestion de Bases de Données

TFS	Team Foundation Server
UAT	User Acceptance Testing
UML	Unified Modeling Language

Table des matières

Dédicaces	i
Remerciements	ii
Résumé	iii
Abstract	iv
Résumé arabe	v
Liste des sigles et acronymes	vi
Table des matières	x
Table des figures	xii
Liste des tableaux	xiii
Introduction	1
1 Contexte général du projet	2
1.1 Présentation de l'entreprise d'accueil SQLI	3
1.1.1 Groupe SQLI	3
1.1.2 SQLI Maroc	5
1.2 Présentation du projet	7
1.2.1 Cadre du projet et problématique	7
1.2.2 Objectifs du projet	7
1.3 Conduite de projet	8
1.3.1 Présentation des Équipes du Projet	8

1.3.2	Méthodologie de travail : Scrum	10
1.3.3	Capitalisation et suivi	11
1.3.4	Planification du projet	13
2	Analyse et spécification des besoins	17
2.1	Étude de l'existant	18
2.1.1	Méthodes de paiement actuelles	18
2.1.2	Particularités du marché belge	18
2.1.3	Justification de l'intégration de Payconiq by Bancontact	19
2.1.4	Processus de paiement actuel	19
2.2	Identification et analyse des exigences	21
2.2.1	Exigences fonctionnelles	21
2.2.2	Exigences non-fonctionnelles	24
3	Conception de la solution	26
3.1	Architecture de l'application	27
3.1.1	Architecture Globale	27
3.1.2	Architecture backend simplifiée	28
3.1.3	Environnement de livraison et de test	30
3.1.4	Architecture applicative	32
3.2	Conception détaillée	34
3.2.1	Diagramme de classe	34
3.2.2	Diagramme de séquence	35
4	Implémentation et Validation	37
4.1	Technologies utilisées	38
4.2	Illustration de l'Intégration de la Méthode de Paiement	40
4.3	Analyse et Correction des Anomalies	54
4.4	Tests et Validation	57
4.4.1	Tests	57
4.4.2	Validation des besoins fonctionnels	58
4.4.3	Validation des besoins non fonctionnels	58
Conclusion		60

Références

60

Table des figures

1.1	Logo de SQLI [1]	3
1.2	Chiffre Clés de SQLI	4
1.3	Clients du SQLI [1]	5
1.4	Départements de SQLI	6
1.5	Structure de l'équipe Seasonal Event	9
1.6	Structure de l'équipe Cart Checkout & Payment	9
1.7	Page d'accueil de Confluence	12
1.8	Backlog du projet sur Azure DevOps	12
1.9	Calendrier des formations	13
1.10	Structure Chanel One	15
2.1	Processus du paiement	20
2.2	Diagramme de cas d'utilisation	22
3.1	Architecture globale de l'application E-commerce	27
3.2	Architecture backend simplifiée de l'application E-commerce	29
3.3	Architecture de livraison et de test simplifiée	30
3.4	Environnements de test et production	31
3.5	Architecture applicative de l'application E-commerce	32
3.6	Diagramme de classe	34
3.7	Diagramme de séquence	35
4.1	Logo de SAP [5]	38
4.2	Logo de JEE [6]	38
4.3	Logo de Spring [7]	39
4.4	Logo de SonarQube [8]	39
4.5	Logo d'Ant [9]	40

4.6	Logo de JUnit [10]	40
4.7	Activation de Payconiq	41
4.8	Disponibilité de Payconiq dans la boutique en ligne	41
4.9	Ajout d'un produit au panier	42
4.10	Saisir les informations personnelles	42
4.11	Mode de livraison	43
4.12	Saisir un code promo	44
4.13	Facturation	45
4.14	Choix de mode de paiement	46
4.15	Finalisation du paiement via l'API Adyen	47
4.16	Écran de traitement en cours	47
4.17	Montant à payer	48
4.18	Paiement réussi	49
4.19	Confirmation de la commande	50
4.20	Informations de la commande	51
4.21	Message d'erreur en cas d'annulation du paiement	52
4.22	Confirmation de la commande dans le backoffice	53
4.23	Message de confirmation par email	54
4.24	Bugs détectés par SonarQube	55
4.25	Analyse avec SonarLint après correction	56
4.26	Exemple ticket de debugging	56
4.27	Couverture de code à 100%	57
4.28	Succès de tous les cas de test	57

Liste des tableaux

1.1	Fiche technique de SQLI Maroc	5
2.1	Description textuelle du cas d'utilisation "Passer Commande"	23
2.2	Description textuelle du cas d'utilisation "Finaliser l'achat d'une commande"	24

Introduction générale

Dans un environnement numérique en perpétuelle évolution, où les attentes des consommateurs et les technologies avancent rapidement, il est crucial pour les entreprises d'améliorer continuellement leurs plateformes e-commerce afin de rester compétitives. Ce projet de fin d'études s'inscrit dans ce contexte dynamique, en visant à améliorer une plateforme e-commerce pour un client spécifique et à intégrer la méthode de paiement Payconiq, particulièrement pertinente pour le marché belge.

L'objectif principal de ce projet est de moderniser la plateforme en ajoutant Payconiq comme nouvelle méthode de paiement, afin d'optimiser l'expérience utilisateur et de répondre aux exigences locales. Ce rapport se compose de quatre chapitres distincts :

Le premier chapitre présente le contexte général du projet, en détaillant l'organisme d'accueil, l'équipe impliquée, la problématique identifiée, les objectifs visés, ainsi que la méthodologie de travail adoptée.

Le deuxième chapitre est dédié à l'analyse de l'existant et à l'étude des besoins fonctionnels et non fonctionnels, en mettant en lumière les spécificités actuelles de la plateforme et les exigences liées à l'intégration de Payconiq.

Le troisième chapitre se concentre sur la conception du projet, illustrée par les diagrammes UML qui décrivent les structures et les interactions au sein de la plateforme.

Enfin, le quatrième chapitre traite la réalisation du projet, en détaillant le processus d'intégration de la nouvelle méthode de paiement et en évaluant son impact sur la plateforme.

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.

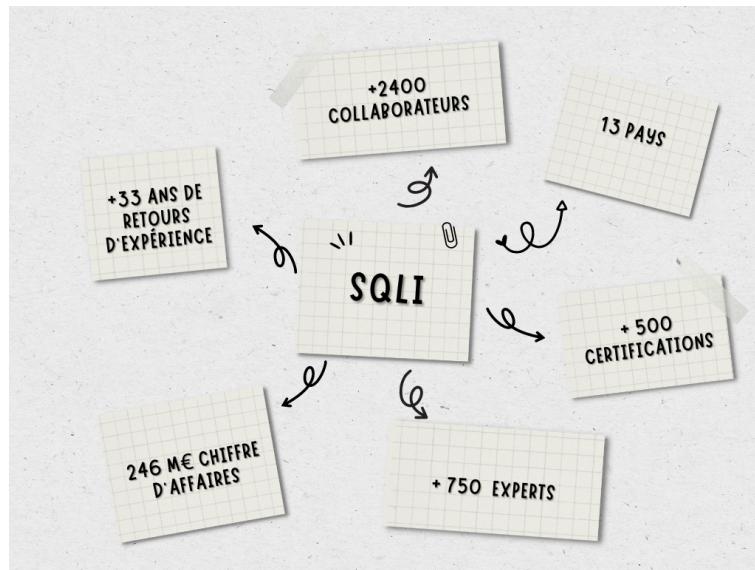


FIGURE 1.2 – Chiffre Clés de SQLI

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 (*Figure 1.3*)

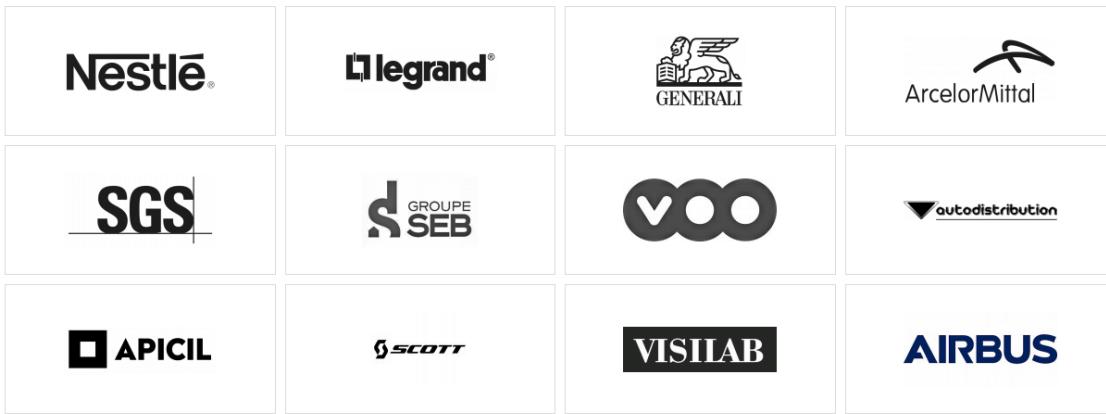


FIGURE 1.3 – Clients du SQLI [1]

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 mon stage PFE, Oujda et Casablanca. Le 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 INTERRACTIVE** : 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** : 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.

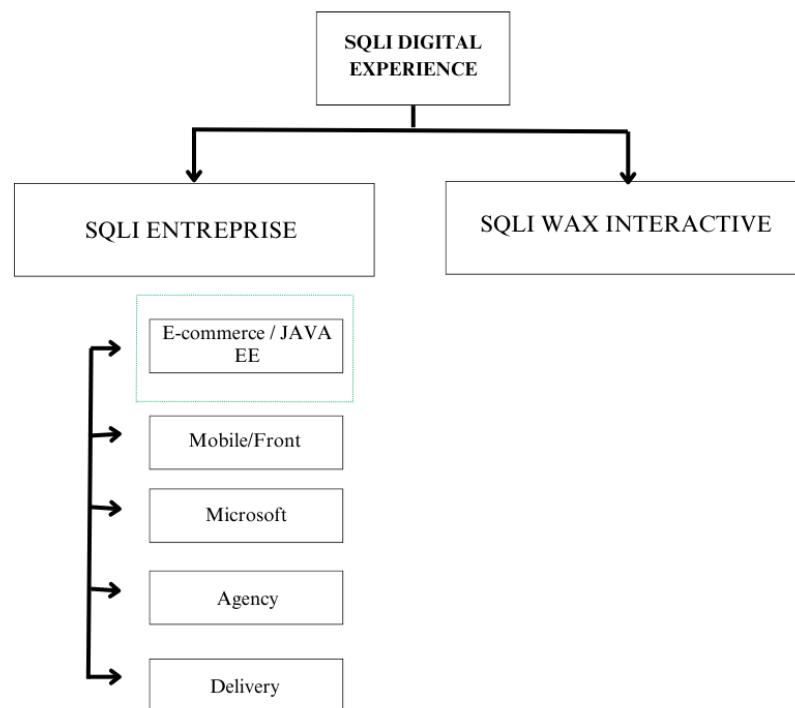


FIGURE 1.4 – Départements de SQLI

Mon stage de fin d'études s'est déroulé dans le département Java JEE, qui regroupe plusieurs projets destinés à des 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 de luxe, 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 du projet 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 de la gestion des fonctionnalités de recherche, un élément crucial permettant aux utilisateurs de trouver efficacement des produits au sein de la plateforme, ainsi que des aspects spécifiques aux lunettes, incluant des filtres de sélection basés sur les formes et types de montures, et de la mode. À mon arrivée, l'équipe se concentrat 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, nommée Seasonal Event, 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, nommée Cart Checkout & Payment, 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.

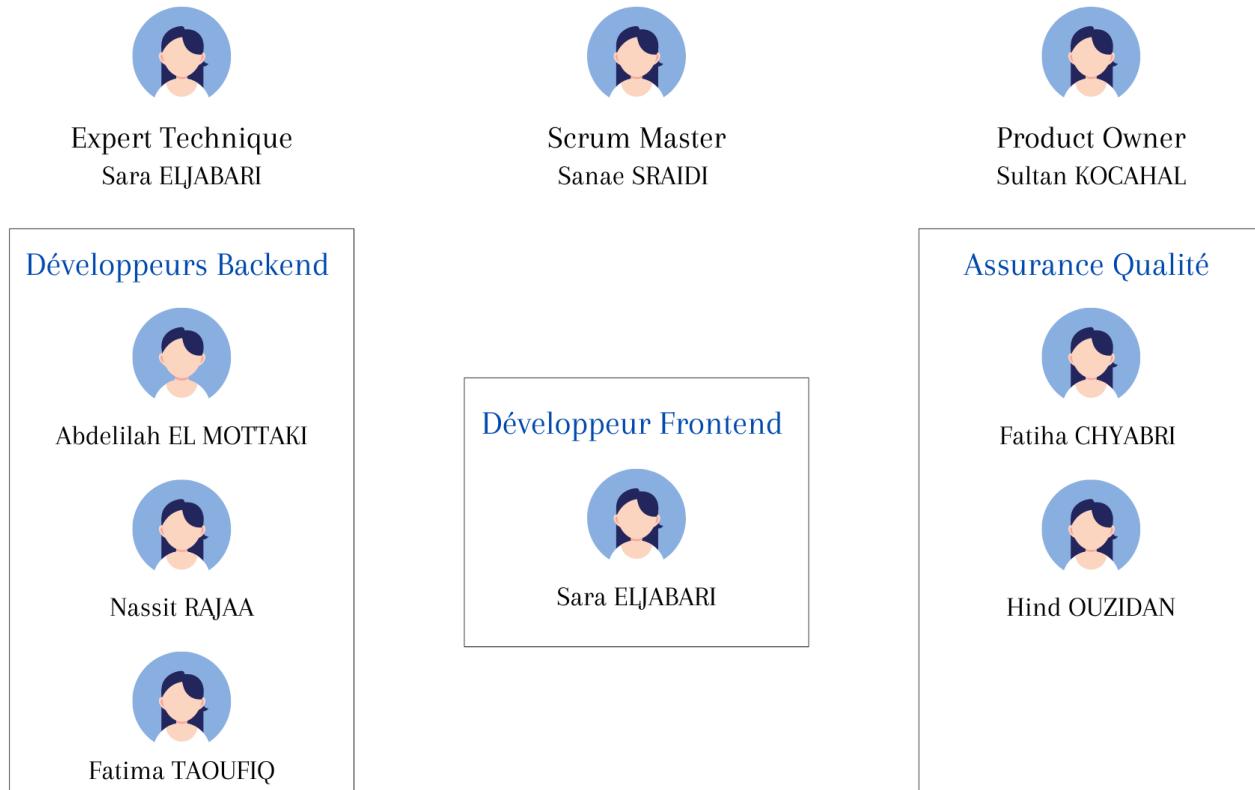


FIGURE 1.5 – Structure de l'équipe Seasonal Event

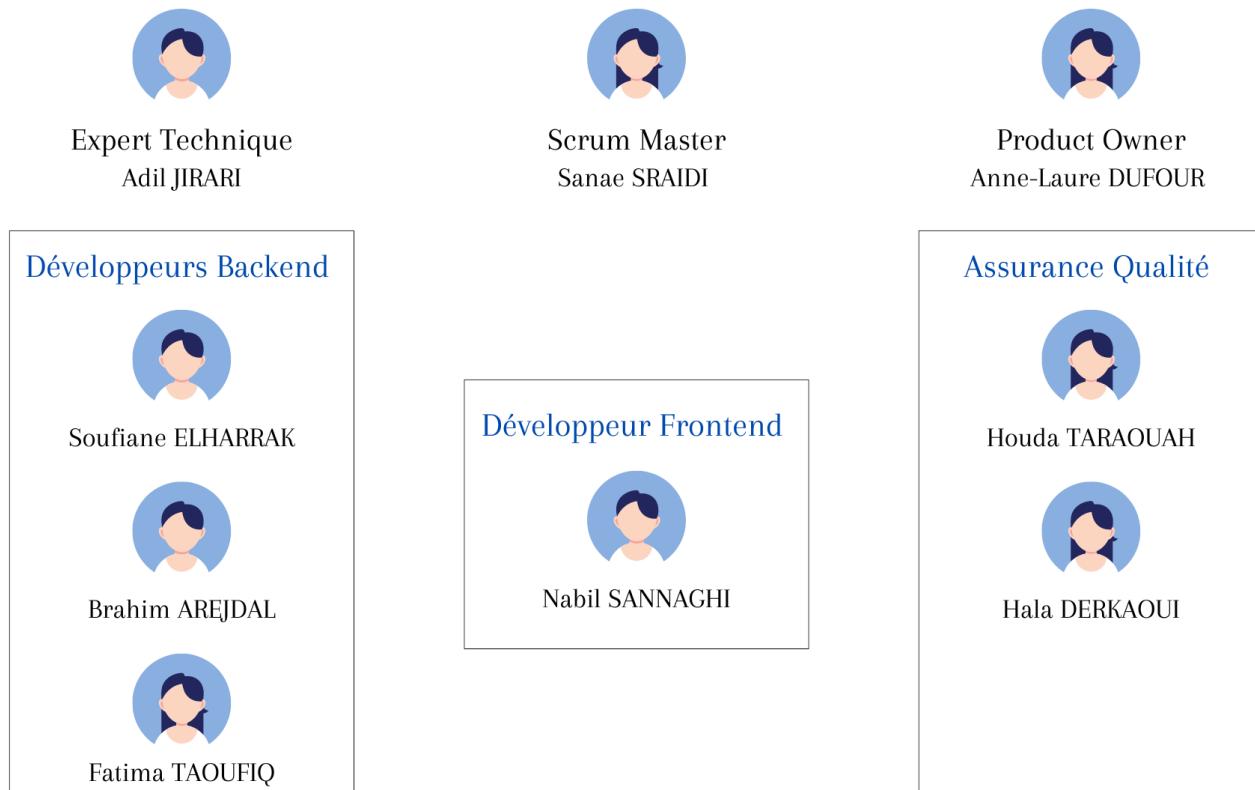


FIGURE 1.6 – Structure de l'équipe Cart Checkout & Payment

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 débuter 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.

This section is dedicated to the Global Digital Services team organization. In this section, you'll be able to find information such as:

- Methodologies
- Tools information
- Backlog Management guidelines
- Organization Chart
- Design toolkit
- Onboarding Kit
- GDS Must Read
- GDS Calendars...

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.

New	Need info	Spec to do	Dev to do	Dev in progress	Blocked
New item	<p>1267416 INT1>AMER> 3D view Page > X icon is not visible correctly on 3D view Page when High Contrast Mode is enabled.</p> <p>Sara ELJABARI</p> <p>State: Need info... Severity: Minor</p>	<p>1269389 QAT1>KR>Search Suggestions in search is not working as expected</p> <p>Sara ELJABARI</p> <p>State: Dev to do Severity: Major</p>	<p>1057436 [Seasonal event] [Sonar] [Critical bugs][Part 3] Cleaning</p> <p>Fatima TAOUFIQ</p> <p>State: Dev in progress Severity: Minor</p>	<p>1264581 POC- Restrictions requests optimization initiative</p> <p>Abdelilah EL MOTT...</p> <p>State: Dev in progress</p>	
	<p>1253922 Attackers can forge malicious links in order to compromise user's browser</p> <p>Nassit RAJAA</p> <p>State: Need info... Severity: Critical</p>				<p>1058243 [99.5][Cleaning] - Null Poi exception on YAPI pr availability</p> <p>Abdelilah EL M...</p> <p>State: Blocked Severity: Major</p>

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 Formation

Durant les deux premiers mois du stage, des formations spécifiques ont été dispensées afin d'assurer une montée en compétences dans les technologies clés utilisées sur le projet. Ces formations ont couvert plusieurs domaines techniques et méthodologiques essentiels, tels que :

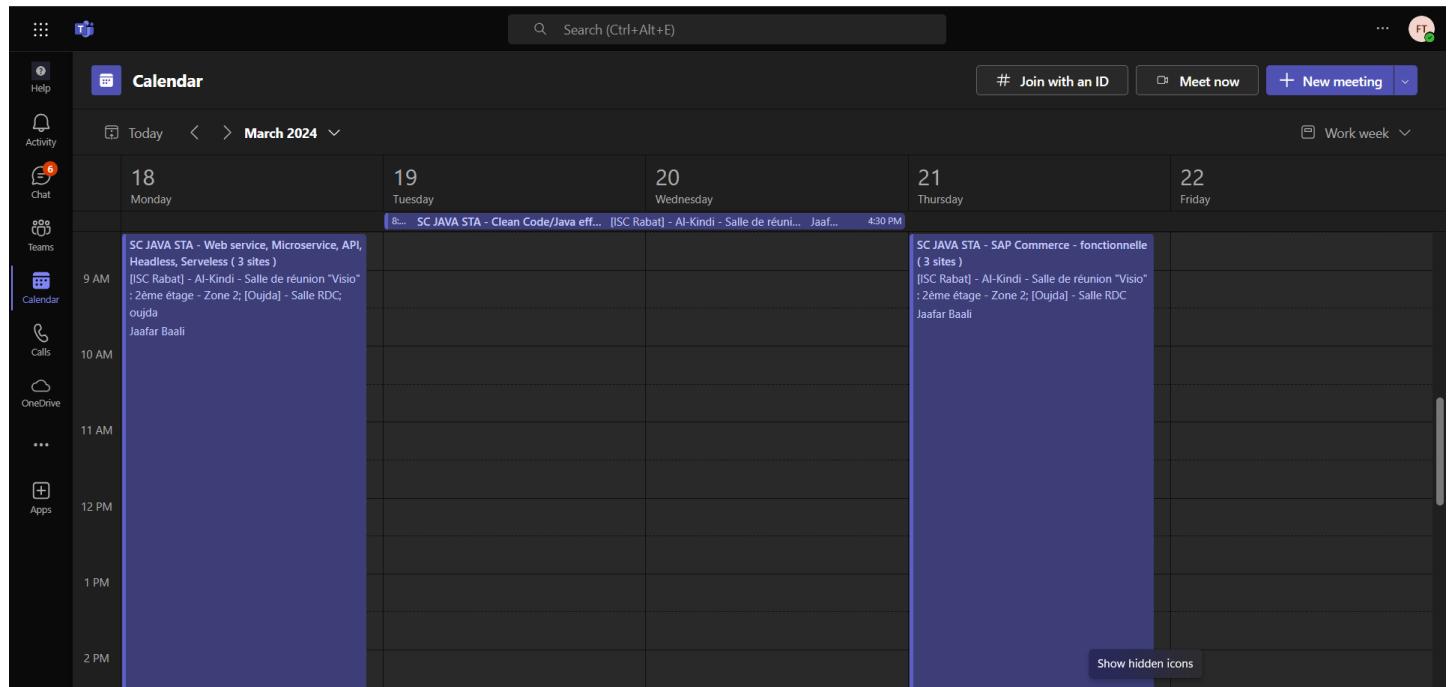


FIGURE 1.9 – Calendrier des formations

- ***Formations Spring*** : La formation sur Spring Framework a abordé les concepts fondamentaux, tels que l'inversion de contrôle (IoC) et Spring MVC. Ces notions ont été approfondies à travers des

exercices pratiques, permettant de comprendre l'architecture et le fonctionnement des applications Java développées avec ce framework.

- **SAP Commerce (Hybris)** : Ce volet a débuté par l'installation de la plateforme SAP Commerce. Les outils d'administration, tels que HAC, HMC et les Cockpits, ont ensuite été explorés, ainsi que les fonctionnalités techniques, notamment la configuration de CronJobs et la gestion des workflows.
- **Clean Code** : Cette formation, inspirée des principes de Robert C. Martin, s'est concentrée sur les bonnes pratiques de codage pour produire un code propre, lisible et maintenable. Des exercices pratiques ont été réalisés pour illustrer l'application concrète des concepts de structuration du code et de refactoring.
- **Software Craftsmanship** : La formation en Software Craftsmanship a mis l'accent sur l'excellence technique et la qualité du code, avec des pratiques telles que le pair programming, la revue de code et l'écriture de tests automatisés. L'objectif était de renforcer les compétences en production de code robuste et maintenable.
- **ICD Skills & Agilité** : Cette formation a couvert les compétences interpersonnelles et les méthodologies agiles. Les méthodologies Scrum et Kanban ont été présentées et appliquées à travers des cas pratiques, renforçant la gestion de projet et la collaboration en équipe.

1.3.4.2 Intégration du projet

J'ai été affecté au projet 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. La figure 1.10 représente l'organigramme global de notre client de Lux, qui se divise en trois domaines : E-commerce, Expérience, et E-services. Chacun de ces domaines est responsable d'un périmètre spécifique du projet, offrant une approche globale et complémentaire.

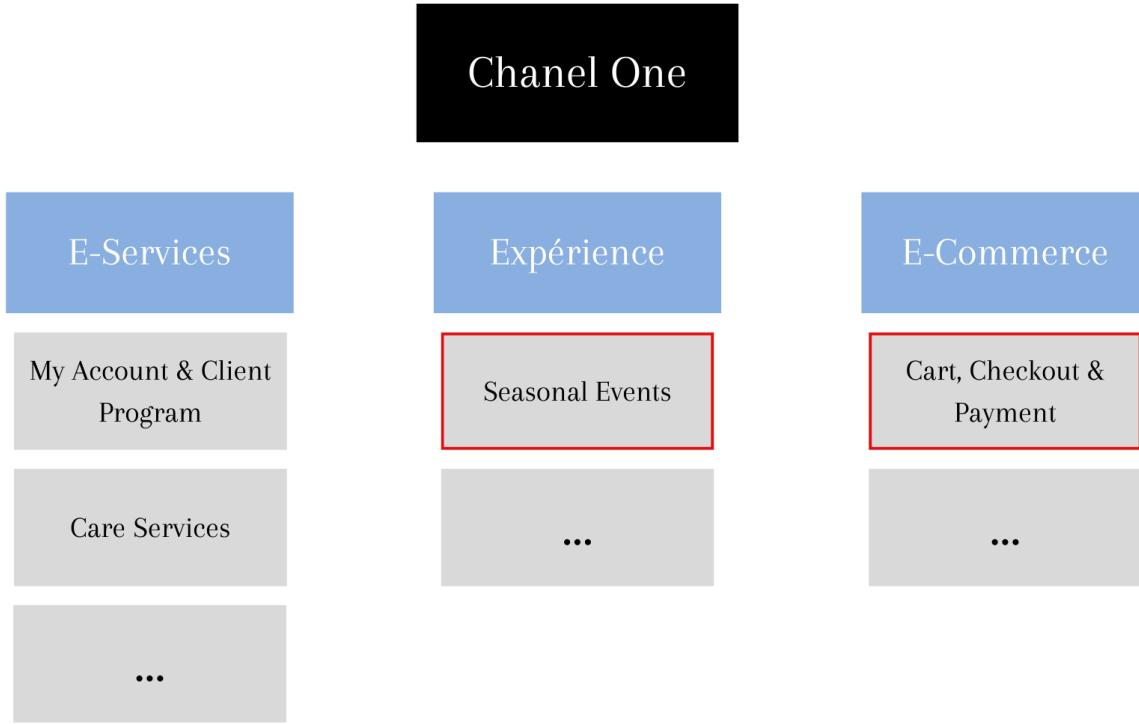


FIGURE 1.10 – Structure Chanel One

- **E-commerce** : Ce domaine englobe l'intégralité des processus liés à l'achat et à la gestion des commandes en ligne. Il couvre notamment la gestion des transactions, depuis l'ajout des produits au panier jusqu'à la finalisation du paiement. L'accent est mis sur l'intégration des méthodes de paiement sécurisées, comme Payconiq, tout en garantissant une fluidité optimale pour les clients. Il inclut également des services liés à la livraison, ainsi que la gestion des retours et des échanges, afin de fournir une expérience complète et sans friction, de l'achat à la réception des produits.
- **E-services** : Ce domaine se concentre sur les services numériques offerts pour enrichir l'expérience client. Cela comprend la gestion des comptes utilisateurs, permettant aux clients de suivre leurs commandes, de gérer leurs préférences et d'accéder aux programmes de fidélité. De plus, E-services inclut des fonctionnalités d'assistance après-vente, comme le suivi des réparations et des demandes spécifiques, et propose des outils permettant d'interagir avec les boutiques physiques, par exemple, en prenant rendez-vous ou en consultant la disponibilité des produits en magasin.
- **Expérience** : Ce domaine vise à améliorer en continu le parcours utilisateur. L'objectif est d'optimiser la présentation des produits et d'intégrer des fonctionnalités innovantes pour rendre la navigation sur le site plus intuitive et attractive. En combinant une esthétique raffinée et des outils techniques performants, le domaine Expérience garantit une interface utilisateur qui soit à la fois élégante et fonctionnelle.

Conclusion

Au cours de ce chapitre, le périmètre du projet a été défini, ainsi que la méthodologie et le planning suivis pour sa réalisation. Le chapitre suivant abordera la phase d'analyse et de spécification du système à développer, visant à comprendre en profondeur les besoins des utilisateurs et à construire un système adapté à ces exigences.

Chapitre 2

Analyse et spécification des besoins

Ce chapitre présente l'analyse de l'existant et la spécification des besoins pour l'intégration de Payconiq by Bancontact. Il examine les méthodes de paiement actuelles, les spécificités du marché belge, le processus de paiement en place, 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 nationale, Bancontact est devenu un élément incontournable du paysage financier belge. Cette solution offre aux consommateurs 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 Payconiq by Bancontact , 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 [3] 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 Payconiq by Bancontact

L'intégration de Payconiq by Bancontact sur notre plateforme e-commerce présente plusieurs avantages stratégiques, particulièrement pour conquérir et fidéliser la clientèle belge [4] :

- **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 Processus de paiement actuel

Le processus de paiement en place, présenté dans la figure 4.6, 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.

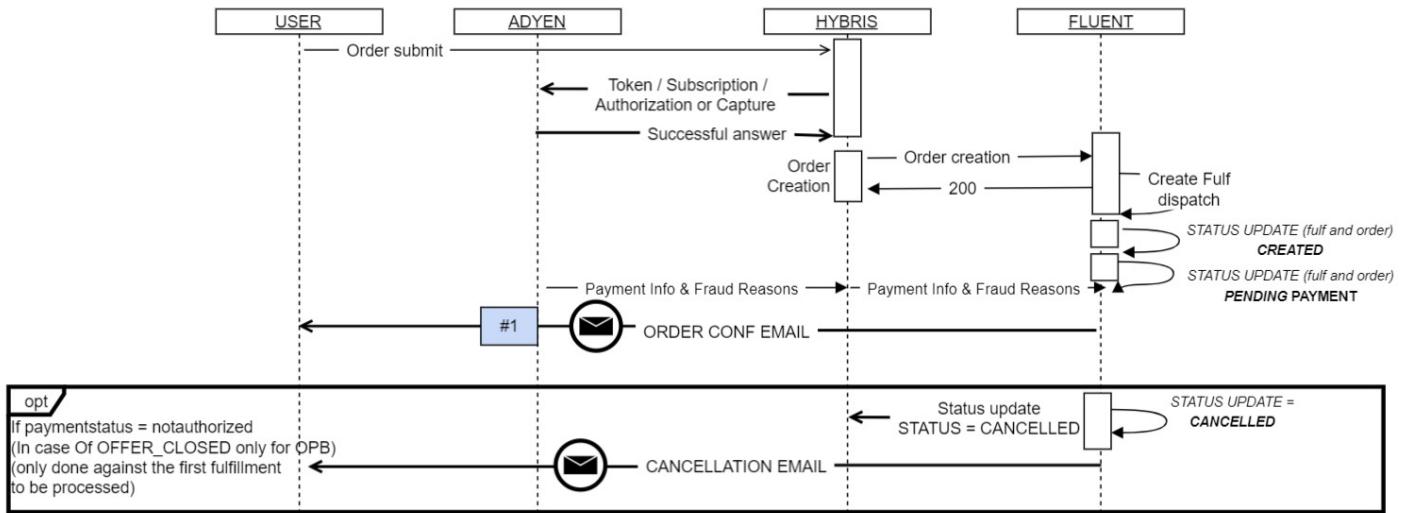


FIGURE 2.1 – Processus du paiement

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 envoie une requête à Adyen pour déclencher le processus de paiement.
3. **Traitements** : 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 Identification et analyse des exigences

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 devra 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 devra être 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 devront être 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 devra être 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 devra informer 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

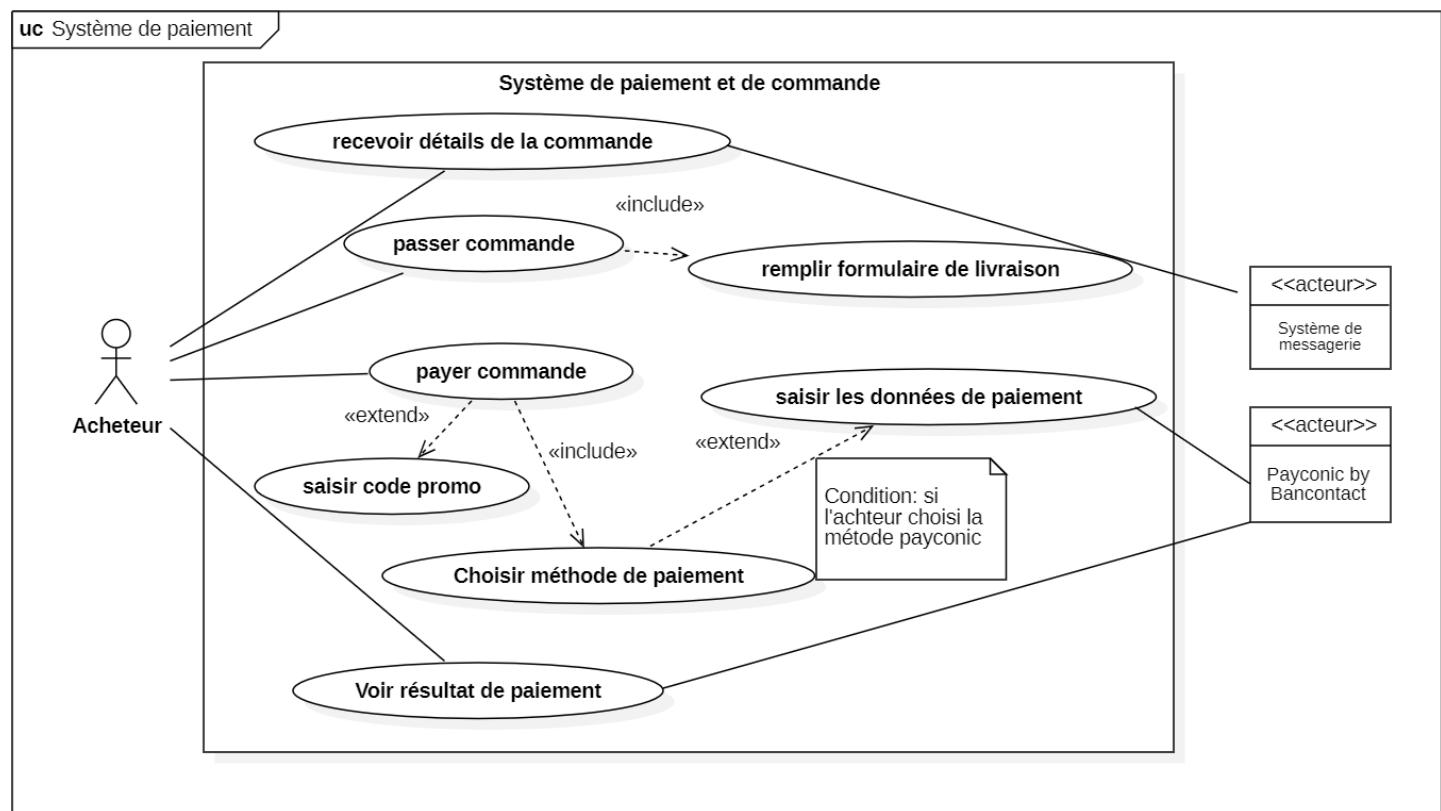


FIGURE 2.2 – Diagramme de cas d'utilisation

2.2.1.3 Analyse des cas d'utilisation

Passer Commande	
But	Confirmer l'achat des produits ajoutés au panier
Acteur	Acheteur
Pré-conditions	Avoir des produits dans le panier
Post-conditions	Saisir les informations de paiement

Scénario normal	<ul style="list-style-type: none"> — Consulter le panier ; — Cliquer sur "Passer commande" ; — Choisir le mode de livraison (à domicile ou click and collect) ; — Saisir les informations de livraison ; — Saisir l'adresse de facturation (peut être différente de l'adresse de livraison) ; — Vérification les informations requises ; — Redirection vers la page de paiement.
Scénario alternatif	<ol style="list-style-type: none"> 1. Un produit sélectionné n'est plus disponible en stock <ul style="list-style-type: none"> — Affichage du message "Produit indisponible" 2. Manque d'informations requises <ul style="list-style-type: none"> — Affichage du message indiquant l'information manquante

TABLE 2.1 – Description textuelle du cas d'utilisation "Passer Commande"

Finaliser l'achat d'une commande	
But	Finaliser l'achat d'une commande
Acteur	Acheteur
Pré-conditions	<ul style="list-style-type: none"> — Passer la commande — Disposer d'un compte payconiq actif
Post-conditions	Saisir les informations de paiement

Scénario normal	<ul style="list-style-type: none"> — Sélectionner Payconiq comme méthode de paiement ; — Génération d'un QR code Payconiq ; — Ouvrir l'application Payconiq by Bancontact et scanner le QR code ; — Confirmer le montant et autoriser le paiement dans l'application ; — Redirection vers une page de confirmation de commande ; — Recevoir une confirmation de paiement sur le site et par e-mail.
Scénario alternatif	<ol style="list-style-type: none"> 1. Fonds insuffisants <ul style="list-style-type: none"> — Affichage du message "Fonds insuffisants" 2. Manque d'informations requises <ul style="list-style-type: none"> — Affichage du message indiquant l'information manquante

TABLE 2.2 – Description textuelle du cas d'utilisation "Finaliser l'achat d'une commande"

2.2.2 Exigences non-fonctionnelles

Les exigences non-fonctionnelles sont essentielles pour assurer 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 le processus actuel et en identifiant les cas d'utilisation spécifiques au marché belge. Nous avons ensuite entamé l'analyse des besoins fonctionnels et non fonctionnels. Dans le chapitre suivant, nous aborderons la conception détaillée du projet.

Chapitre 3

Conception de la solution

Ce chapitre se concentre sur l'architecture de l'application. Il inclut également une exploration approfondie de la conception, illustrée par des diagrammes techniques de classe et de séquence, en se basant sur les spécifications établies dans le deuxième chapitre.

3.1 Architecture de l'application

3.1.1 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.

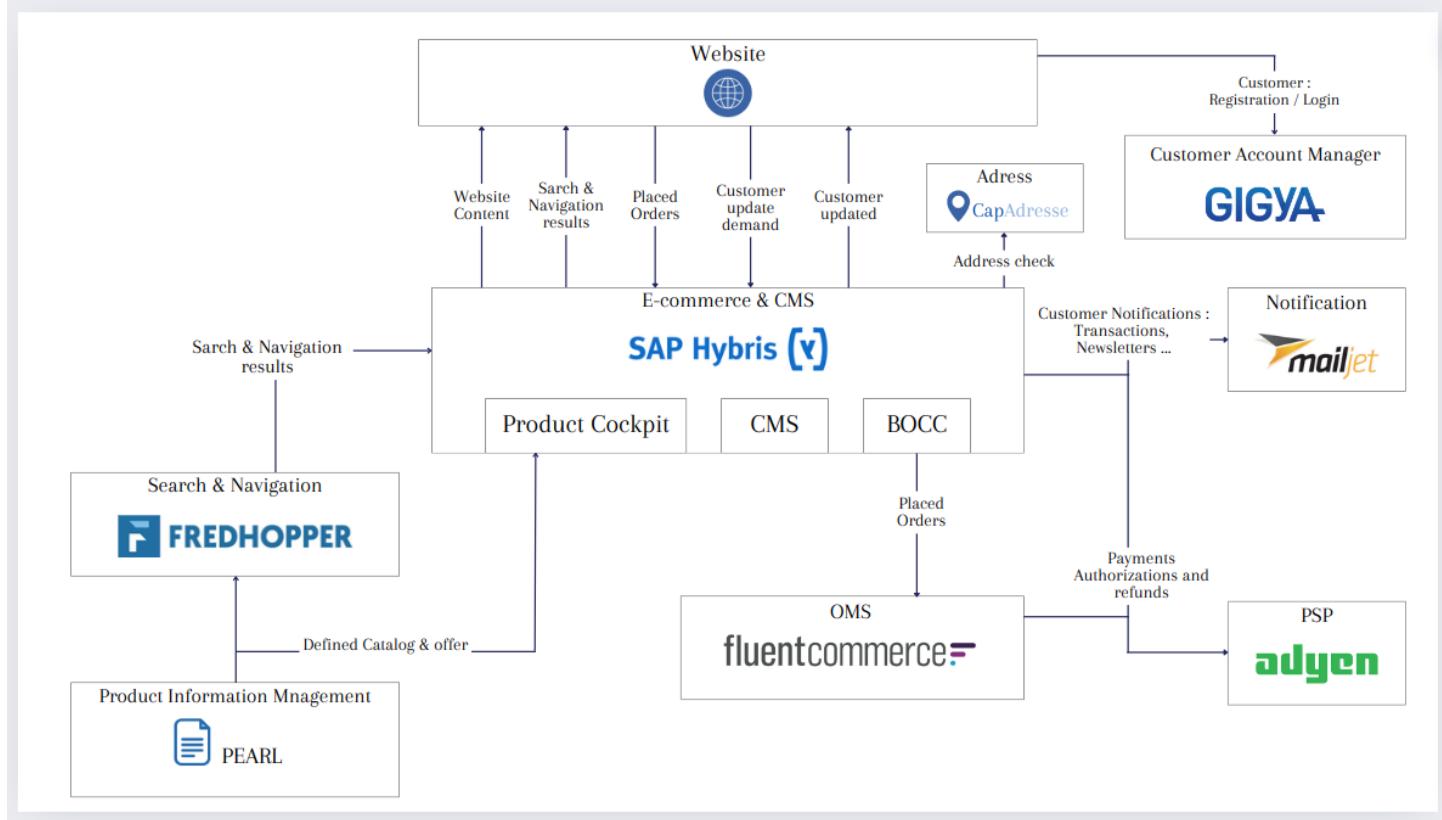


FIGURE 3.1 – Architecture globale de l'application E-commerce

- 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.).

3.1.2 Architecture backend simplifiée

Notre solution est basée sur la plate-forme SAP Hybris Et suit donc l'architecture présentée ci-dessous :

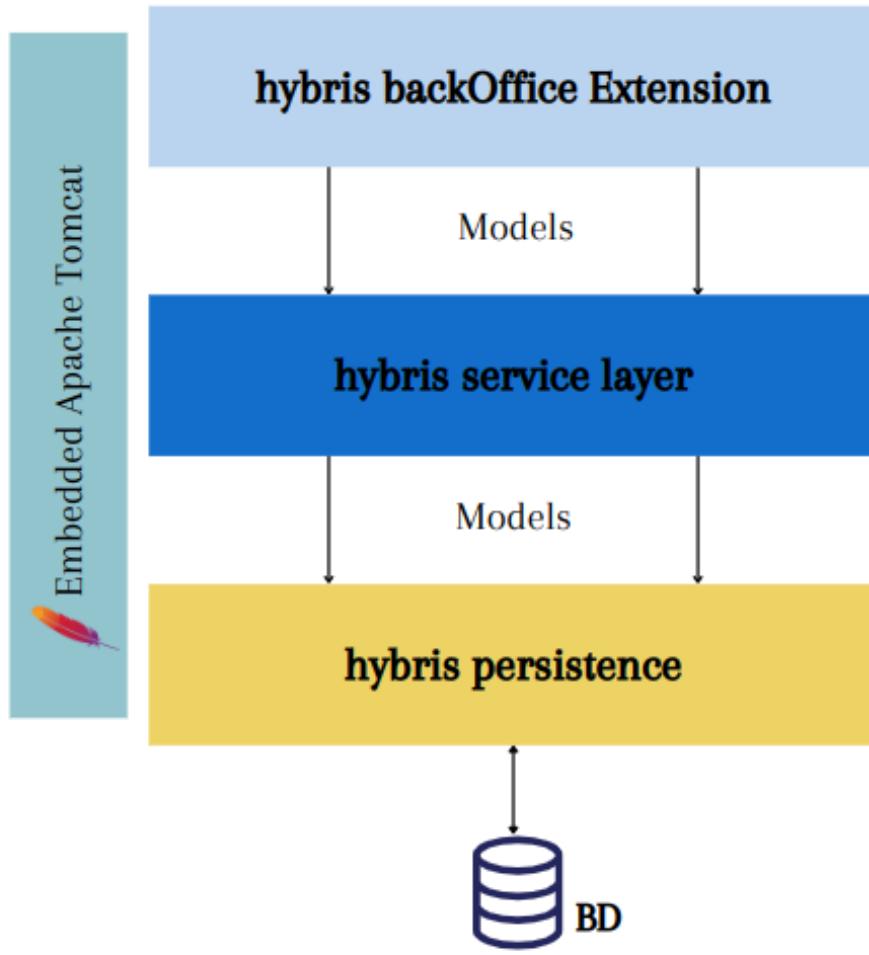


FIGURE 3.2 – Architecture backend simplifiée de l'application E-commerce

- **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.

3.1.3 Environnement de livraison et de test

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

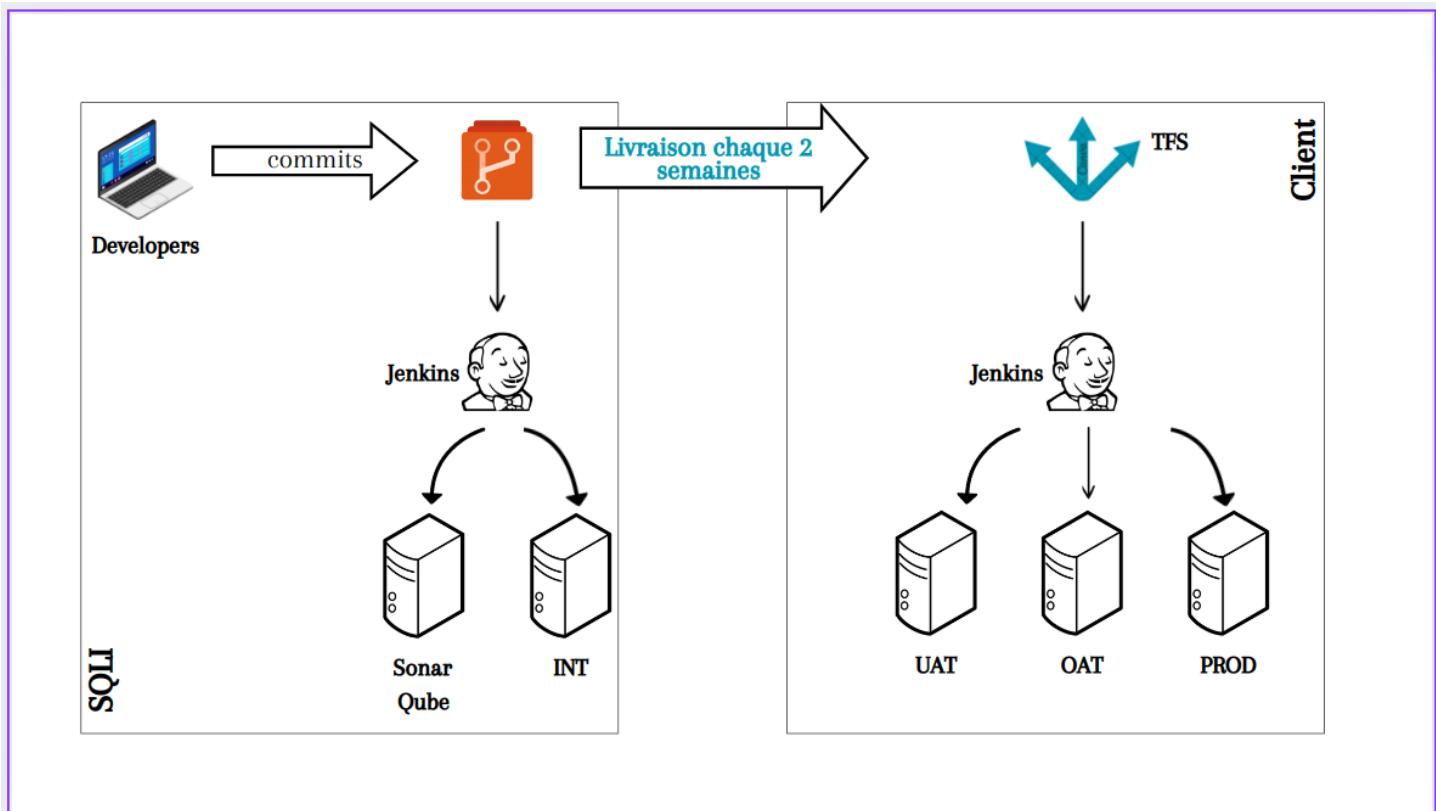


FIGURE 3.3 – Architecture de livraison et de test simplifiée



FIGURE 3.4 – Environnements de test et production

3.1.3.1 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.

3.1.3.2 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.

3.1.4 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 :

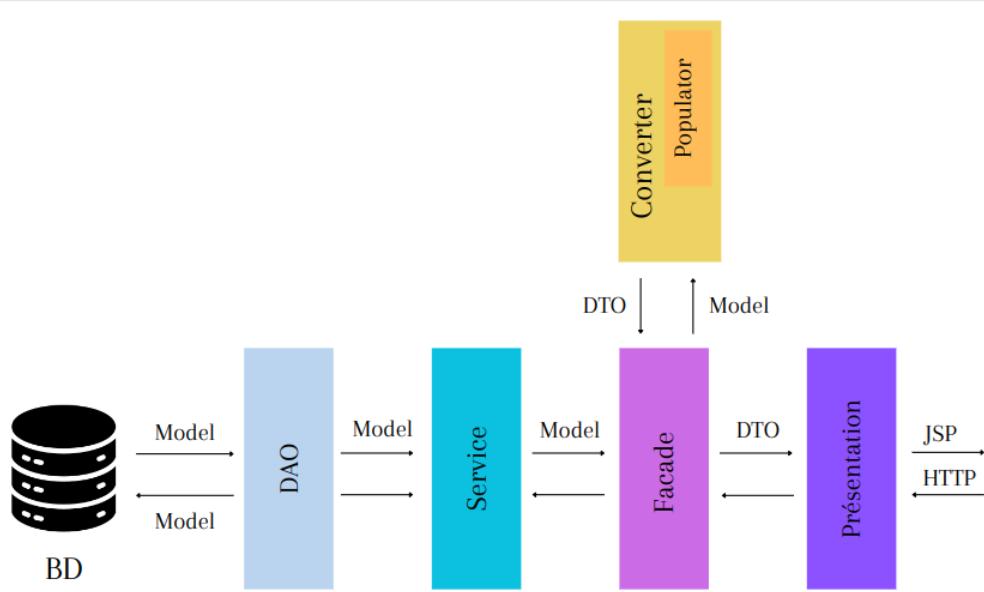


FIGURE 3.5 – Architecture applicative de l'application E-commerce

- **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 simplifie l'accès à un système complexe en fournissant une interface unique qui masque la complexité des sous-systèmes sous-jacents. Dans notre projet, la façade centralise les interactions entre la couche de services et la couche de présentation. Elle reçoit des objets modèles depuis les services, les transforme en DTO grâce à des Converters et Populators, et les transmet à la couche de présentation. Cette approche améliore la modularité et facilite la gestion des données au sein du système.

- **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 communications 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.

3.2 Conception detaillée

3.2.1 Diagramme de classe

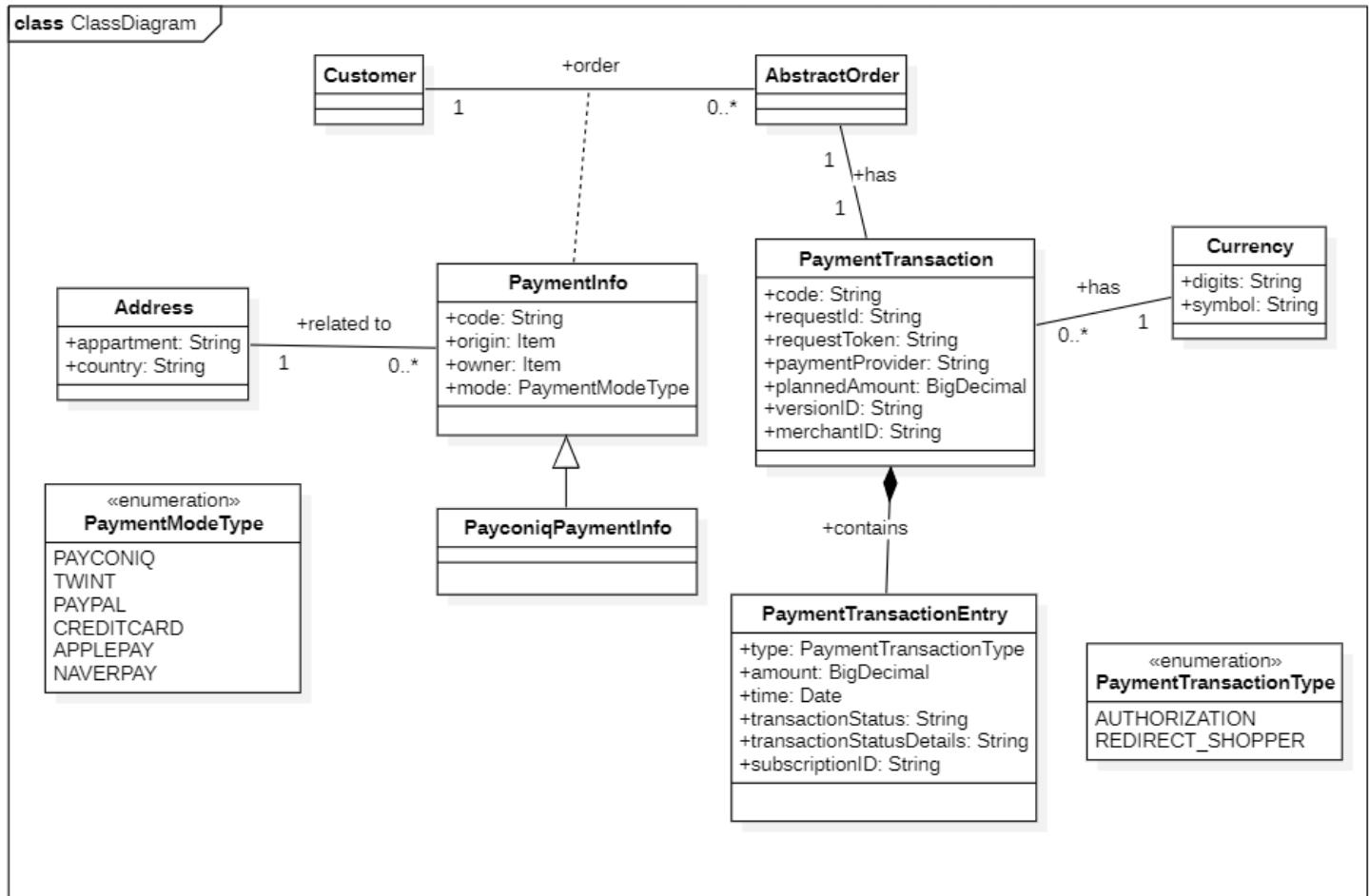


FIGURE 3.6 – Diagramme de classe

La figure ci-dessus représente le diagramme de classe de notre système. Un Client peut passer plusieurs Commandes (représentées par la classe AbstractOrder), et chaque commande est associée à une Transaction de paiement (PaymentTransaction). Cette transaction contient des détails tels que le fournisseur de paiement, le montant prévu, et les identifiants du marchand. Elle est liée à une ou plusieurs Entrées de transaction (PaymentTransactionEntry), qui décrivent des aspects spécifiques comme le type de transaction, le montant, l'heure, et l'état de la transaction. Les informations de paiement sont représentées par la classe PaymentInfo, qui inclut le mode de paiement (défini par l'énumération PaymentModeType), et peut être associée à une Adresse. Une spécialisation de PaymentInfo, nommée PayconiqPaymentInfo, est utilisée pour les transactions via Payconiq. Le système gère également les devises via la classe Currency, qui est associée aux transactions de paiement. En somme, ce diagramme

modélise la partie structurelle d'un volet de la gestion de paiement dans notre site e-commerce où les clients effectuent des paiements par divers moyens.

3.2.2 Diagramme de séquence

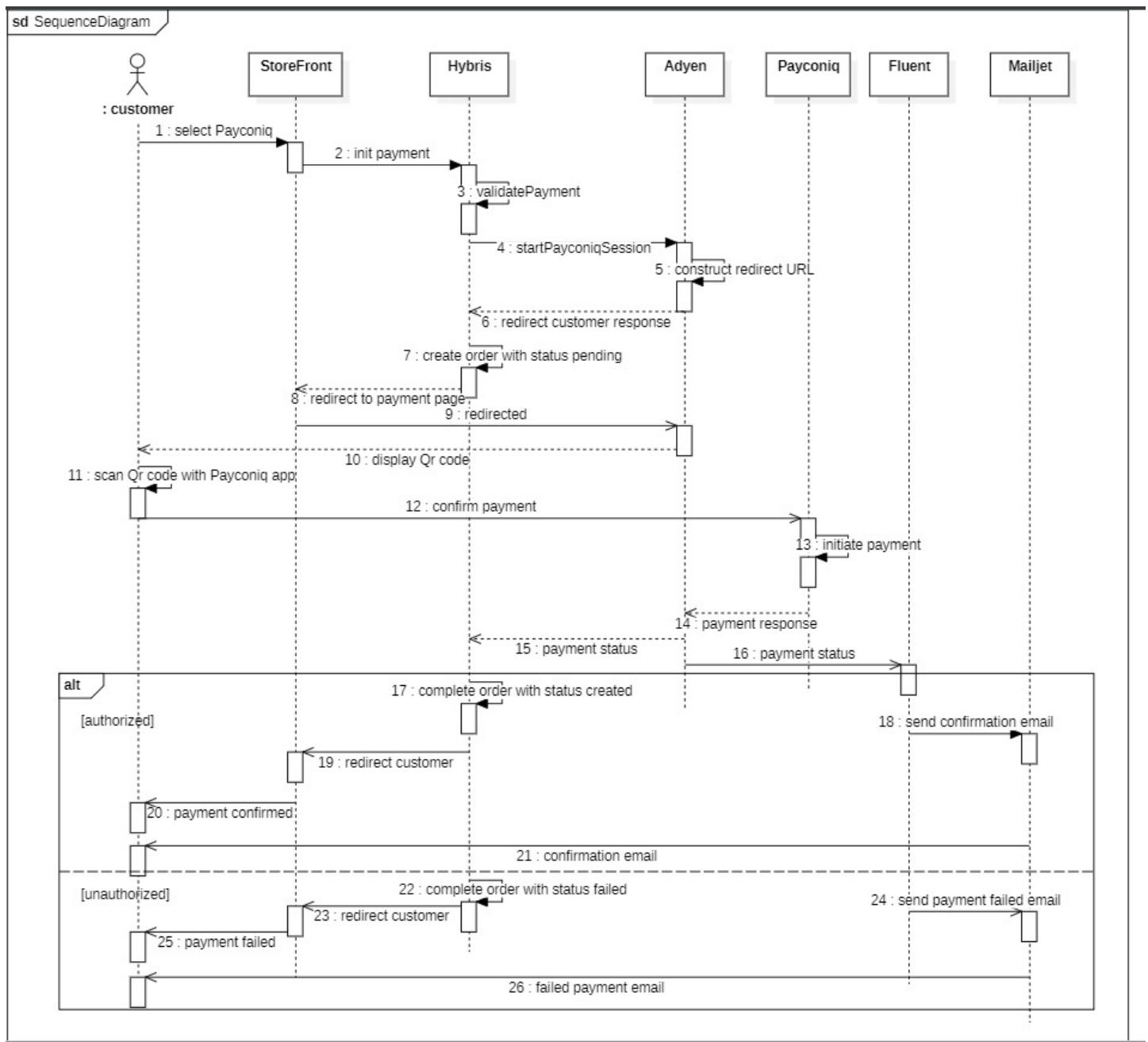


FIGURE 3.7 – Diagramme de séquence

Le processus de paiement via Payconiq commence lorsque le client sélectionne ce moyen de paiement sur la plateforme. La transaction est alors initialisée et validée pour s'assurer que toutes les conditions sont remplies. Ensuite, une session Payconiq est démarrée via Adyen, qui génère une URL de redirection

pour amener le client à la page de paiement Payconiq. Une fois redirigé, une commande est créée avec le statut "en attente". La page de paiement affiche un QR code que le client doit scanner avec son application Payconiq. Après avoir scanné le QR code, la confirmation du paiement est envoyée de Payconiq à Adyen, qui informe ensuite la plateforme e-commerce du statut du paiement.

Si le paiement est autorisé, la commande est complétée avec le statut "créée", le client est redirigé vers une page de confirmation, et un e-mail de confirmation de commande est envoyé. En revanche, si le paiement échoue, la commande est marquée avec le statut "échec", le client est redirigé vers une page d'échec de paiement, et un e-mail d'échec de paiement est envoyé.

Conclusion

Ce chapitre a été dédié à l'étude conceptuelle du projet. Après une présentation des architectures adoptées et des diagrammes techniques de classe et de séquence, 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.

4.1 Technologies utilisées

- SAP Commerce (Hybris)



FIGURE 4.1 – Logo de SAP [5]

SAP Commerce (Hybris) est une plateforme de commerce électronique robuste et évolutive, conçue pour gérer efficacement les catalogues de produits, les commandes, les paiements, ainsi que les interactions avec les clients à travers divers canaux tels que le web, les appareils mobiles, et les points de vente physiques. Dans le cadre de ce projet, SAP Commerce a été choisi pour sa capacité à intégrer des outils de suivi des performances et de marketing, offrant ainsi une gestion complète du commerce en ligne et une expérience utilisateur optimale.

- JEE (JSP)



FIGURE 4.2 – Logo de JEE [6]

JEE (JSP) a été utilisé pour développer la partie frontend de la méthode de paiement. En combinant du code Java et HTML, JSP facilite la création de pages web dynamiques permettant l'affichage interactif des informations de paiement et la gestion des transactions utilisateur. Cette technologie a permis de concevoir une interface utilisateur réactive tout en tirant parti de la flexibilité et de la puissance du backend Java pour assurer une gestion efficace des flux de paiement.

- **Spring**



FIGURE 4.3 – Logo de Spring [7]

Spring a été employé pour développer la logique métier associée à la nouvelle méthode de paiement. Il a permis la création de services back-end pour la validation des paiements, l'interaction avec les systèmes de paiement externes, et le traitement des retours d'information. Grâce à son intégration transparente avec SAP Commerce (Hybris), Spring a assuré une séparation claire des responsabilités et une maintenabilité accrue du code, tout en exploitant ses fonctionnalités avancées pour la gestion des transactions et la sécurité.

- **SonarQube**



FIGURE 4.4 – Logo de SonarQube [8]

SonarQube est un outil d'analyse statique de code permettant de détecter les bugs, les vulnérabilités de sécurité, et d'évaluer la qualité du code source. Dans ce projet, SonarQube a été utilisé pour analyser le code existant et les nouvelles fonctionnalités ajoutées. L'outil a permis de corriger les bugs et les failles de sécurité identifiés, tout en fournissant des recommandations pour améliorer la structure et la lisibilité du code, garantissant ainsi le respect des standards de qualité requis par le client.

- **Ant**



FIGURE 4.5 – Logo d'Ant [9]

Ant est un outil de build open-source qui permet d'automatiser diverses tâches liées au processus de développement, telles que la compilation, le déploiement, et l'exécution des tests. Dans le cadre de ce projet, Ant a été utilisé pour automatiser la compilation du code, faciliter le débogage, et exécuter les tests unitaires. Grâce à sa configuration native avec Hybris, qui utilise des scripts Ant prédéfinis, Ant a contribué à une organisation efficace des étapes de développement, assurant ainsi un flux de travail structuré et fluide.

- **JUnit**



FIGURE 4.6 – Logo de JUnit [10]

JUnit est un framework de tests unitaires en Java utilisé pour tester les composants individuels d'une application. Pour l'intégration de la nouvelle méthode de paiement, JUnit a été employé pour écrire des tests unitaires afin de vérifier la fiabilité des fonctionnalités back-end. Les tests couvraient divers scénarios, y compris la validation des transactions, la gestion des erreurs, et les interactions avec les systèmes externes.

4.2 Illustration de l'Intégration de la Méthode de Paiement

Cette partie fournit une vue détaillée des fonctionnalités développées à travers une série de captures d'écran. Avant de finaliser l'intégration de Payconiq via Adyen, il est crucial de vérifier que cette méthode de paiement est correctement activée dans le système. Comme le montre la figure 4.7, Payconiq est marqué comme actif dans le Cockpit d'administration de SAP Commerce. Cela confirme qu'il est prêt à

être utilisé pour le traitement des paiements, garantissant que les transactions effectuées avec Payconiq seront traitées correctement.



FIGURE 4.7 – Activation de Payconiq

Une fois activée, Payconiq a été ajoutée à la boutique en ligne pour le marché belge. La figure 4.8 montre que Payconiq apparaît désormais parmi les méthodes de paiement disponibles dans l'onglet eCommerce, aux côtés de giftcard, applepay-eu, et paypal-eu. Cette configuration permet aux utilisateurs de choisir Payconiq lors de la validation de leur commande.

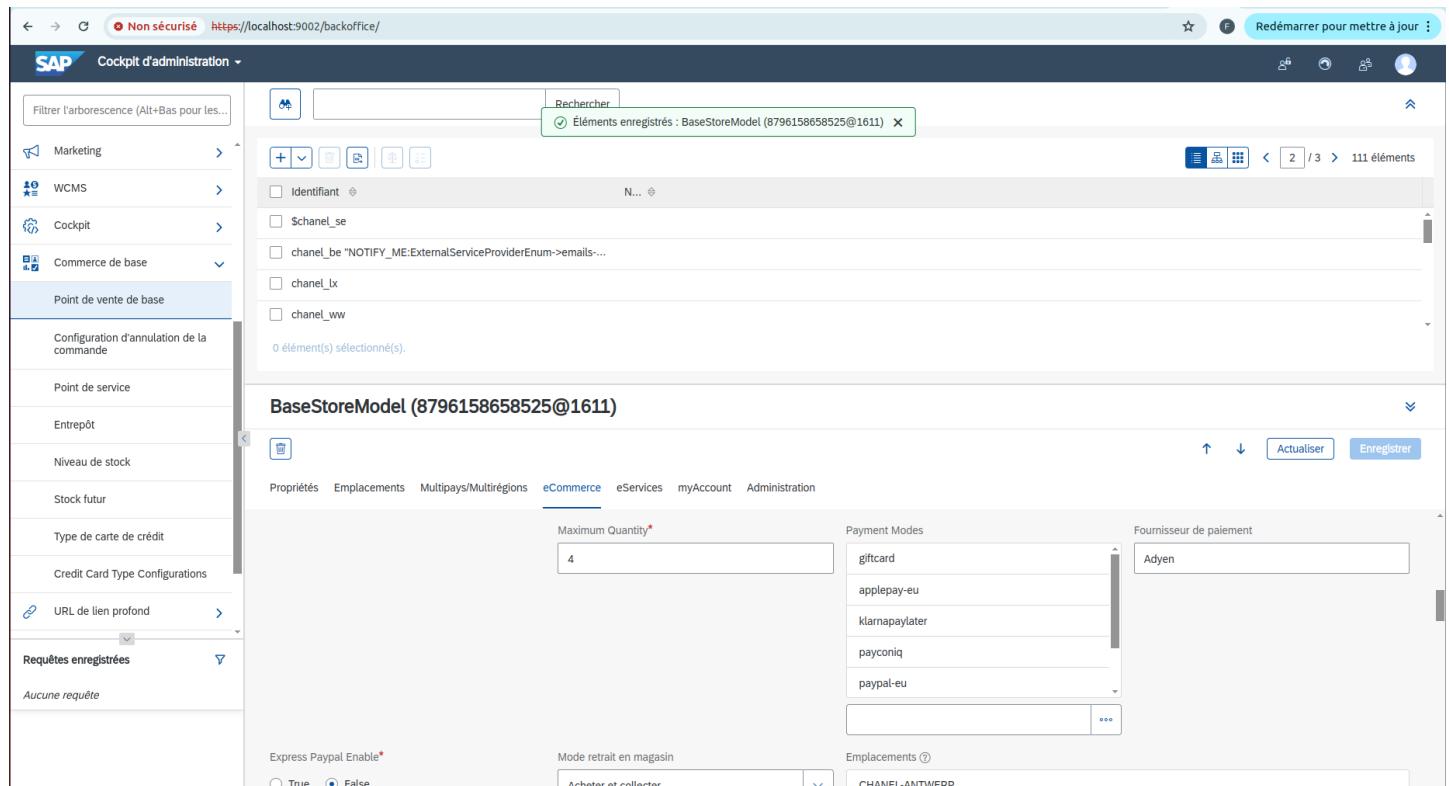


FIGURE 4.8 – Disponibilité de Payconiq dans la boutique en ligne

Lorsque le client sélectionne les articles désirés, il passe à la phase de paiement. La figure 4.9 illustre l'ajout d'un produit au panier, une étape préalable au paiement.

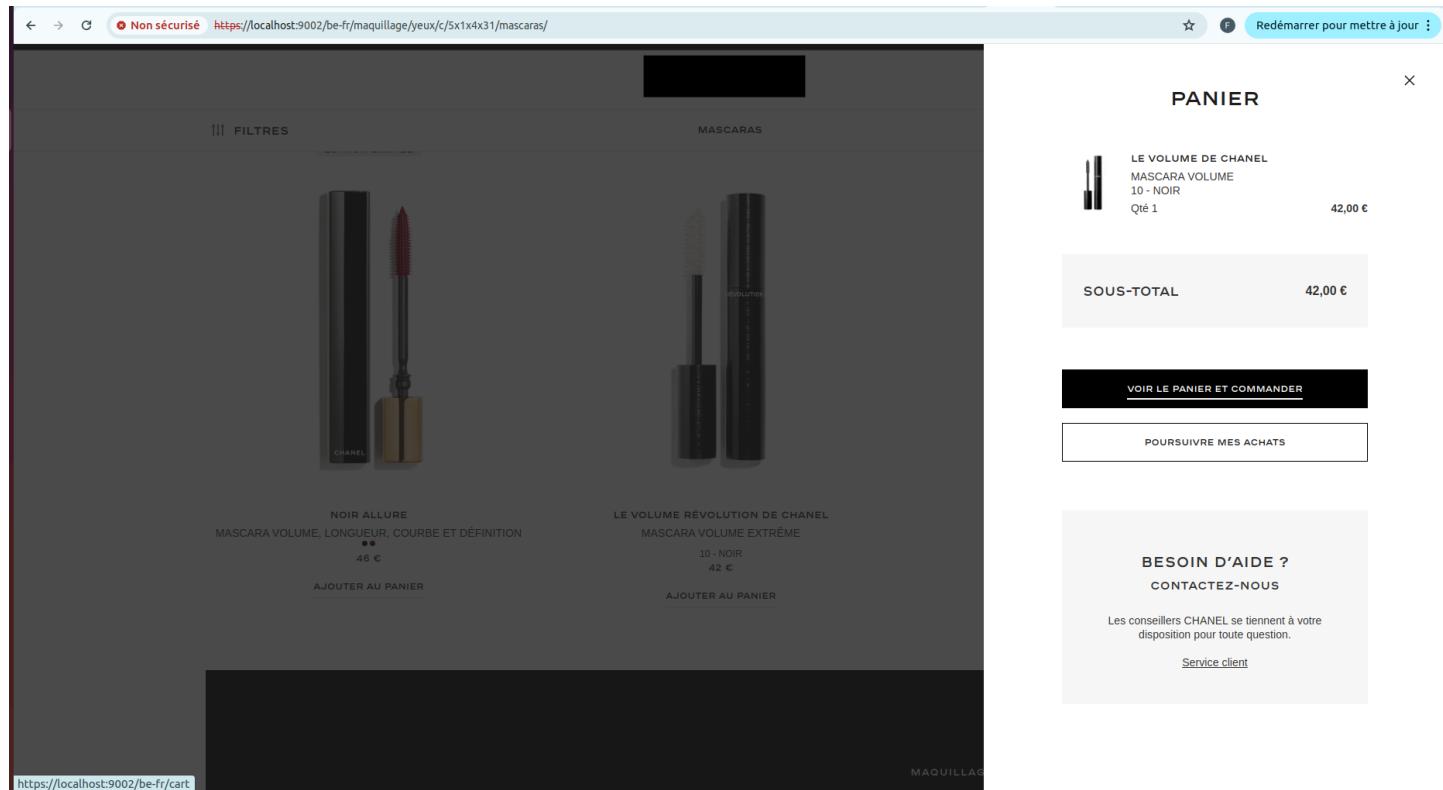


FIGURE 4.9 – Ajout d'un produit au panier

C'est ici que commence la première étape du processus de paiement, où l'utilisateur est invité à saisir ses informations personnelles, y compris l'adresse e-mail, comme le montre la figure 4.10.

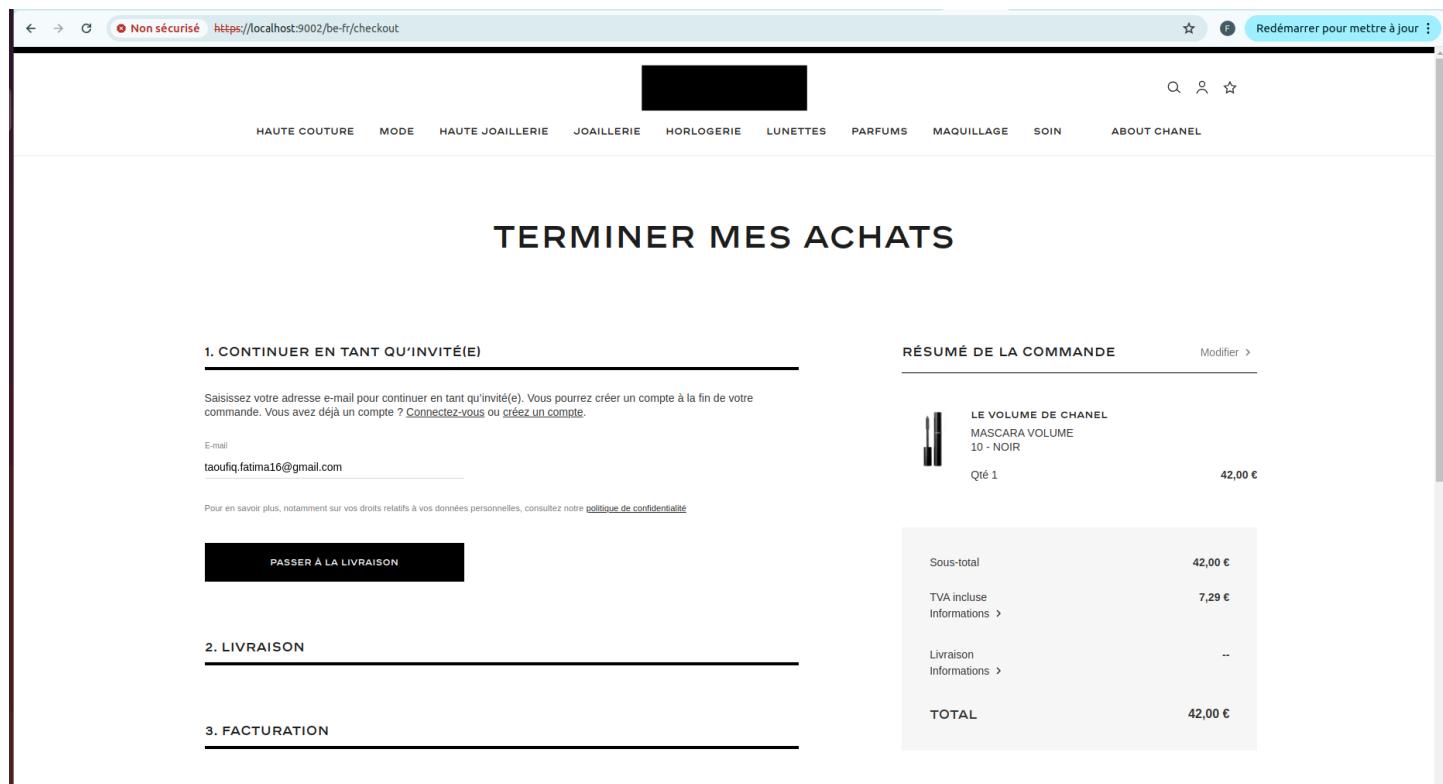


FIGURE 4.10 – Saisir les informations personnelles

Après avoir rempli ses informations personnelles, l'utilisateur passe à la deuxième étape du processus de paiement, dédiée à la sélection du mode de livraison. Comme illustré dans la figure 4.11, cette étape permet à l'utilisateur de choisir entre la livraison à domicile ou le retrait en magasin (Click & Collect). En fonction de l'option sélectionnée, l'interface offre les champs nécessaires : pour la livraison à domicile, l'utilisateur doit saisir une adresse complète, tandis que pour le retrait en magasin, il choisit le point de retrait souhaité. Cette personnalisation assure une expérience de commande adaptée aux préférences de livraison de chaque utilisateur.

1. CONTINUER EN TANT QU'INVITÉ(E)

La confirmation de votre commande sera envoyée à l'adresse taoufiq.fatima16@gmail.com

2. LIVRAISON

Vous explorez actuellement la boutique en ligne CHANEL Belgique. [Livraison vers un autre pays](#).

Veuillez sélectionner un mode de livraison

DOMICILE
livré chez vous

CLICK & COLLECT
en boutique CHANEL

Civilité Mme	Prénom Fatima	Nom TAOUFIQ
Pays Belgique	Code postal 1000	Ville BRUXELLES
Adresse (numéro + rue) ALLEE DES AMAZONES		Apt, suite, étage, etc. (facultatif)
Indicatif pays +32 (BE)	Numéro de téléphone 478 54 78 84	(?)

RÉSUMÉ DE LA COMMANDE

LE VOLUME DE CHANEL
MASCARA VOLUME
10 - NOIR

Qté 1 42,00 €

Sous-total	42,00 €
TVA incluse Informations >	7,29 €
Livraison Informations >	--
TOTAL	42,00 €

ÉCRIN

LE CLASSIQUE
Un coffret ou un sac recyclable, noir et blanc.

FIGURE 4.11 – Mode de livraison

Si un code promotionnel est disponible, l'utilisateur peut l'entrer dans le champ prévu à cet effet (*Figure 4.12*) pour bénéficier d'une réduction.

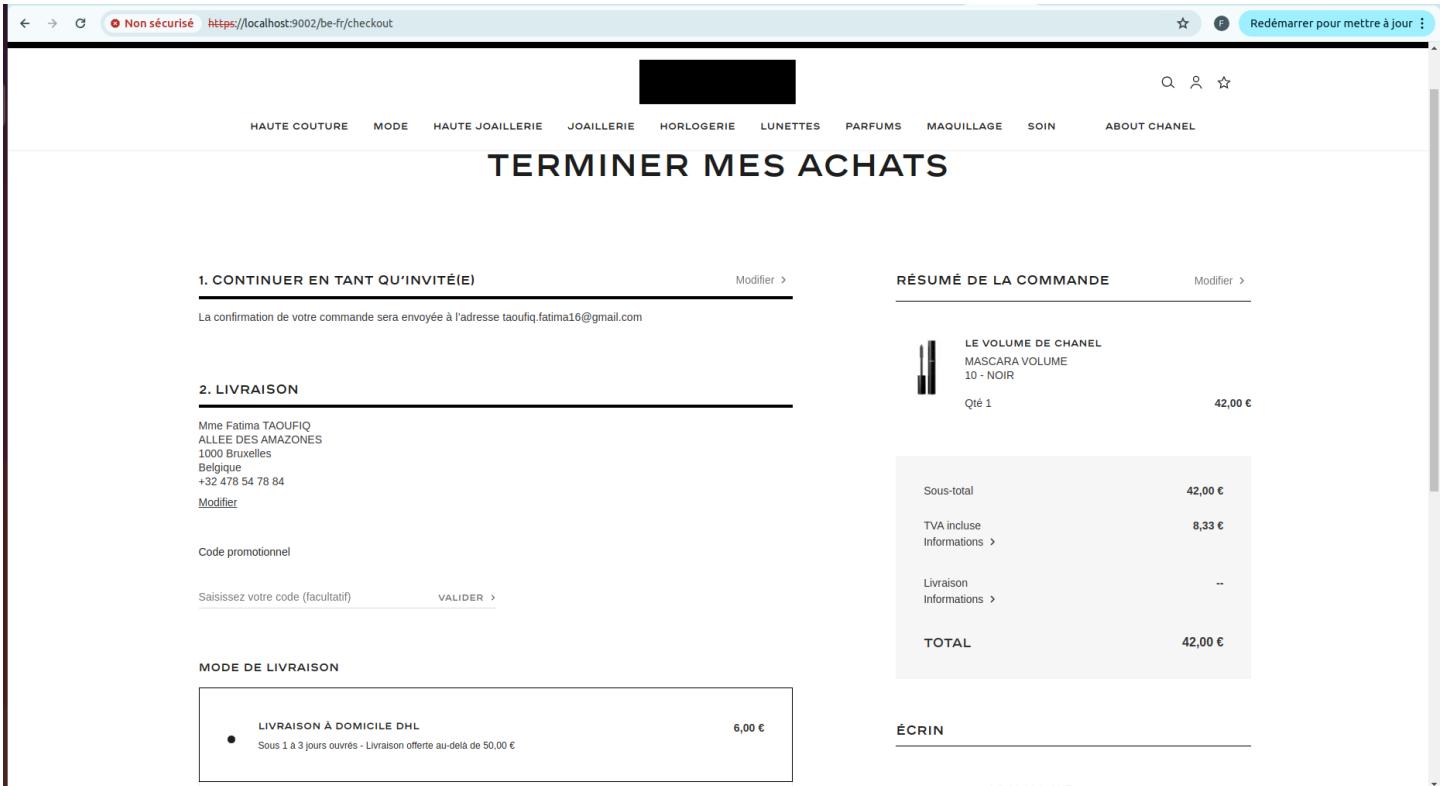


FIGURE 4.12 – Saisir un code promo

En l'absence de code, l'utilisateur poursuit directement vers l'étape suivante : la facturation. À ce stade, il est invité à saisir ses informations de facturation. L'interface (*Figure 4.13*) propose de reprendre automatiquement l'adresse de livraison pour simplifier le processus, mais permet également de renseigner une adresse différente si nécessaire. Cette flexibilité assure que les informations de facturation peuvent être adaptées selon les besoins de l'utilisateur, tout en garantissant que les détails de facturation restent distincts de ceux de la livraison, si tel est le souhait.

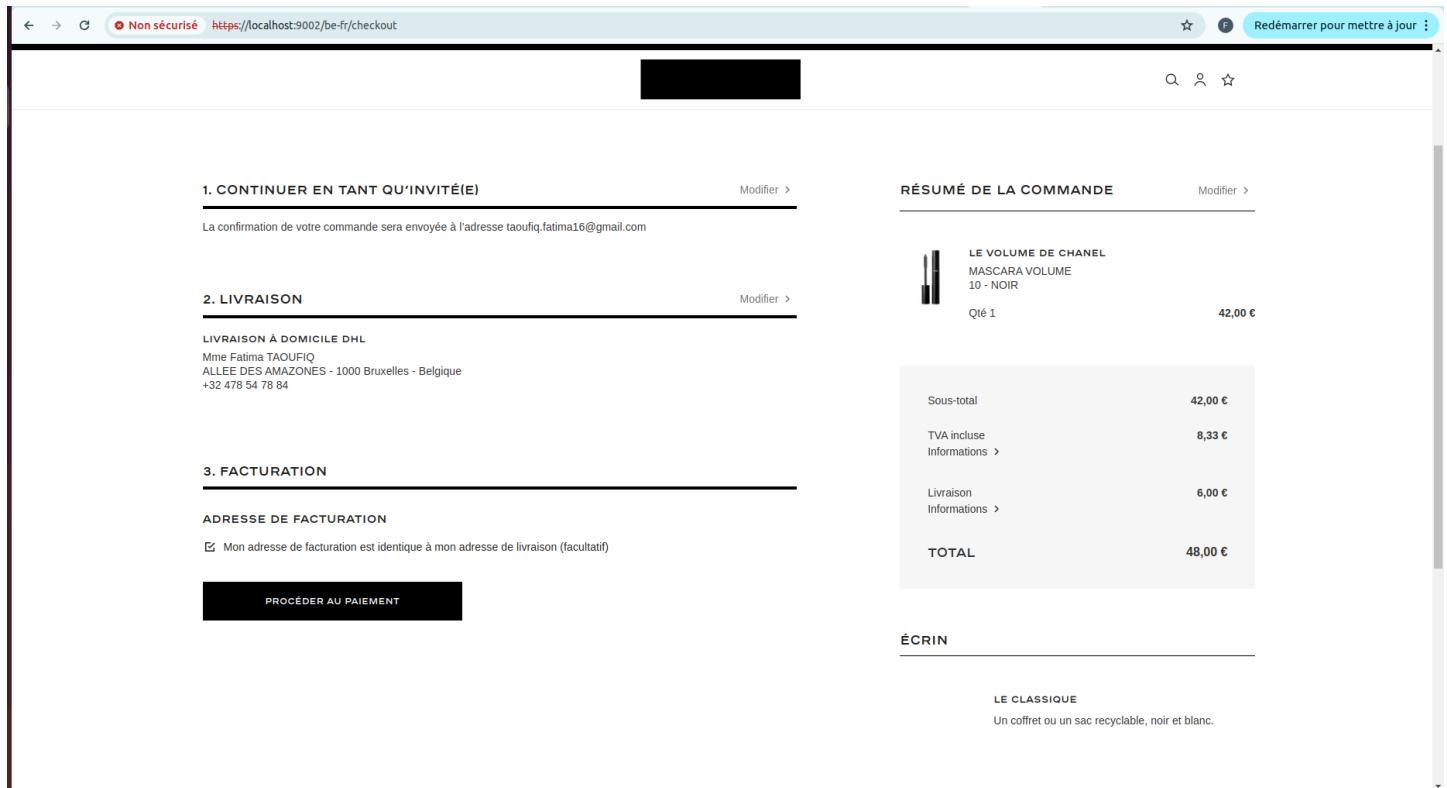


FIGURE 4.13 – Facturation

Après que le client a complété toutes ses informations de facturation, il accède à l'interface de sélection des modes de paiement (*Figure 4.14*). À cette étape, il peut choisir parmi plusieurs options disponibles, y compris Payconiq, qui est mise en avant comme méthode de paiement. L'interface est conçue pour faciliter le choix de la méthode préférée par l'utilisateur avant de procéder à l'étape suivante du processus de paiement.

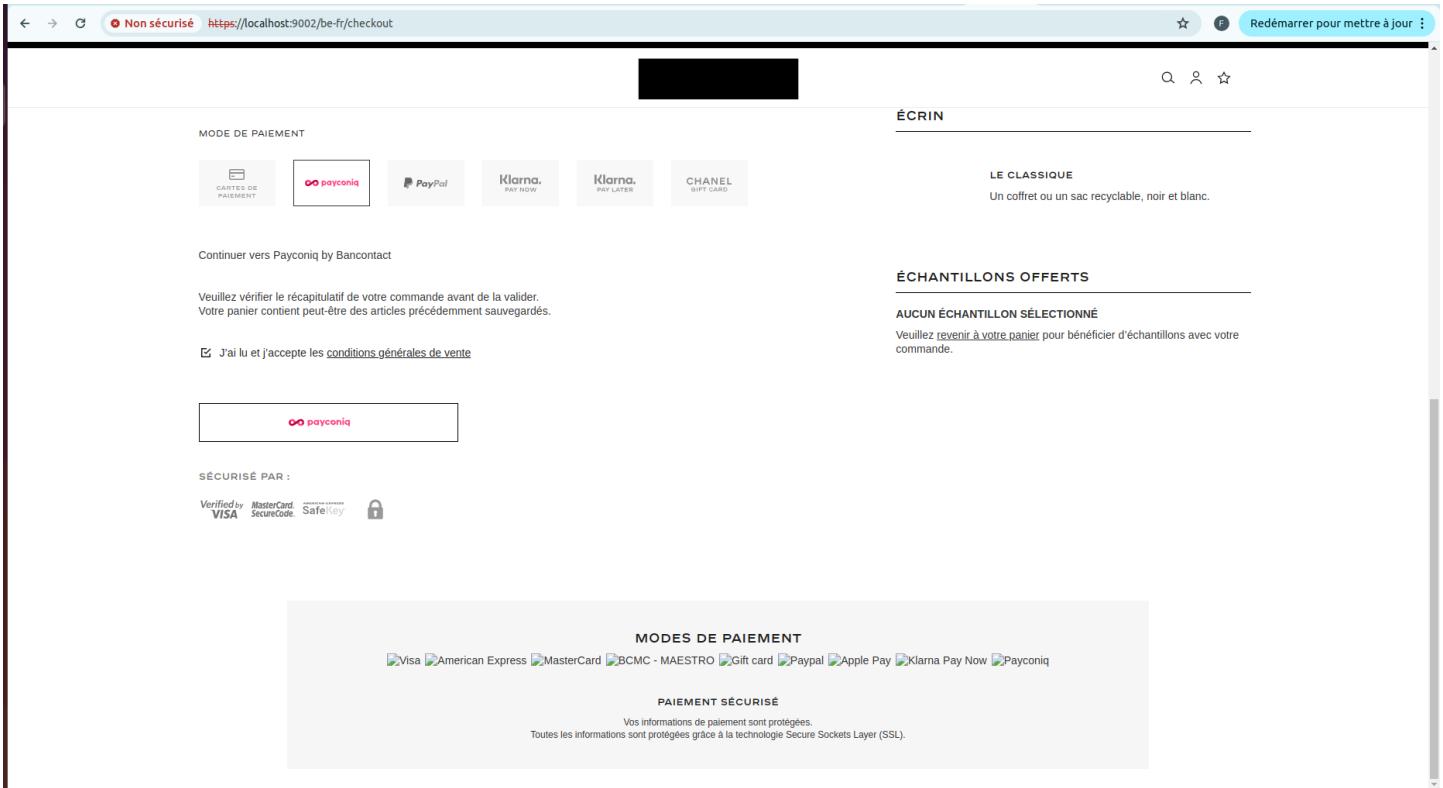


FIGURE 4.14 – Choix de mode de paiement

Après avoir sélectionné le mode de paiement, l'utilisateur est dirigé vers l'API Adyen. Sur cette interface, un QR code est généré pour finaliser la transaction. Comme l'illustre la figure 4.15, ce QR code doit être scanné dans les 15 minutes pour éviter l'annulation automatique de la commande. L'interface fournit également un compte à rebours indiquant le temps restant pour effectuer le paiement, garantissant ainsi une expérience utilisateur fluide et sécurisée.

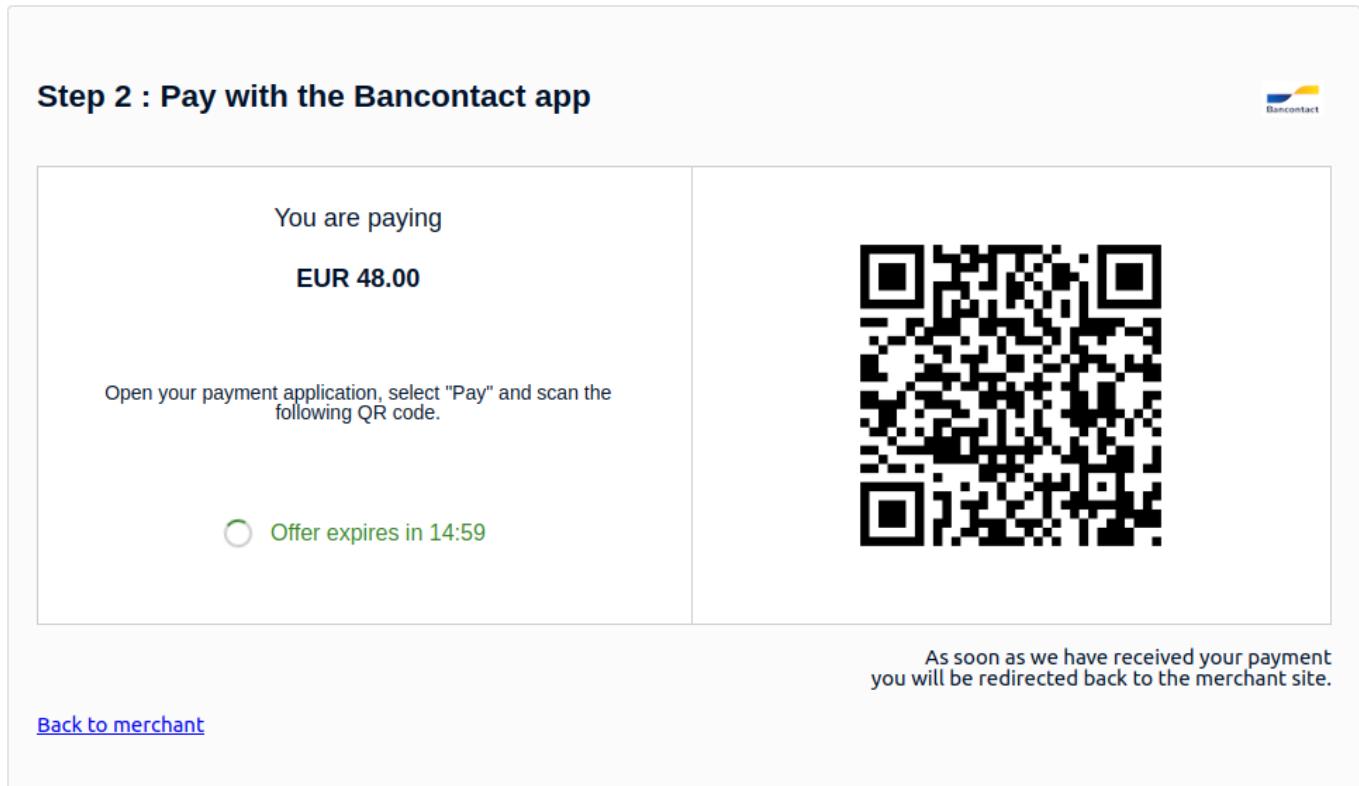


FIGURE 4.15 – Finalisation du paiement via l'API Adyen

Une fois redirigé vers la page de paiement, l'utilisateur traverse plusieurs étapes : Tout d'abord, il est accueilli par un écran indiquant qu'il doit patienter pendant que la transaction est traitée.

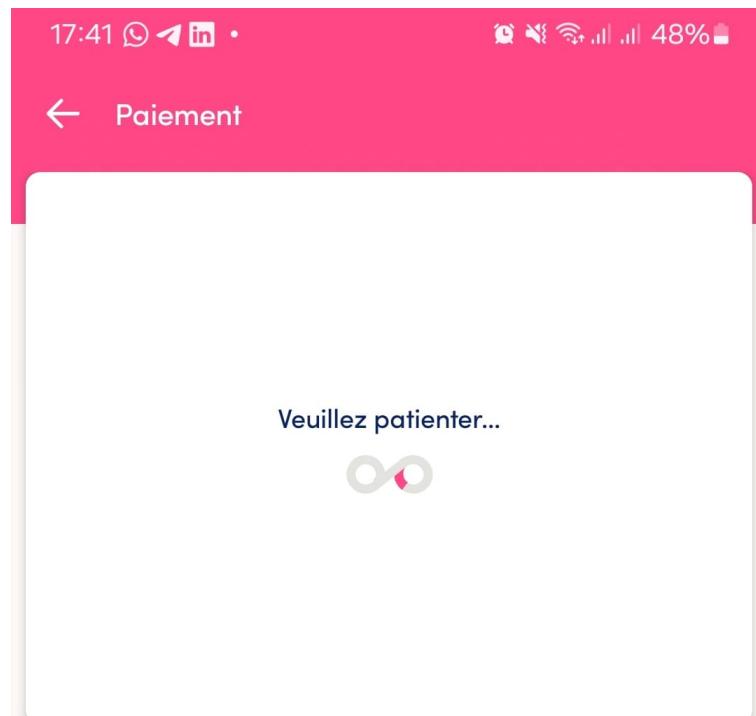


FIGURE 4.16 – Écran de traitement en cours

Ensuite, l'écran suivant affiche le montant total à payer et demande à l'utilisateur d'autoriser la transaction, comme le montre la figure 4.17

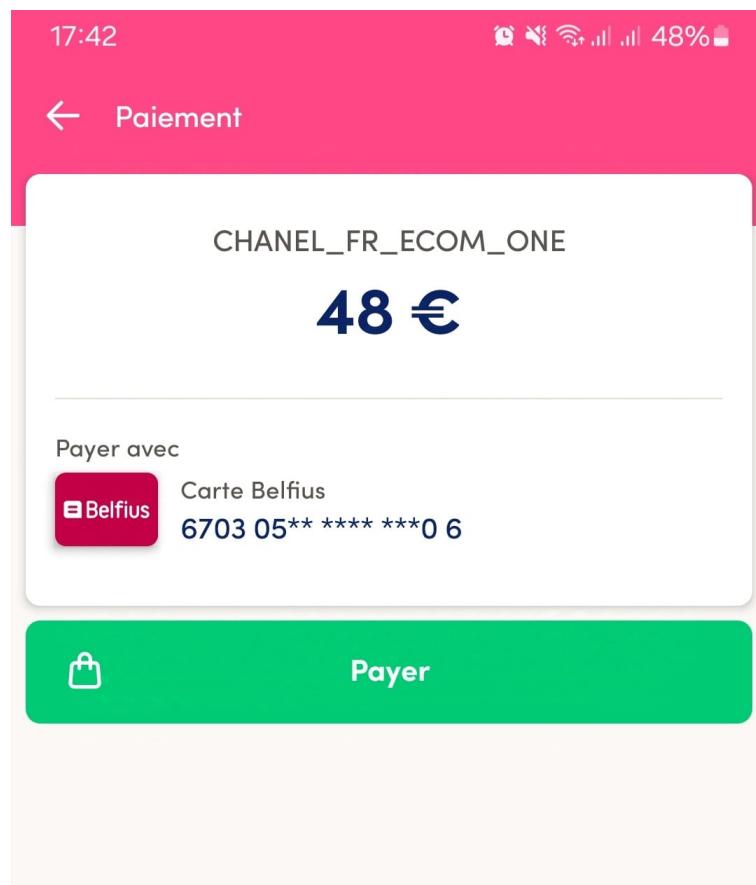


FIGURE 4.17 – Montant à payer

Enfin, lorsque la transaction est réussie, l'utilisateur est redirigé vers un écran de confirmation de paiement, illustré dans la figure 4.18

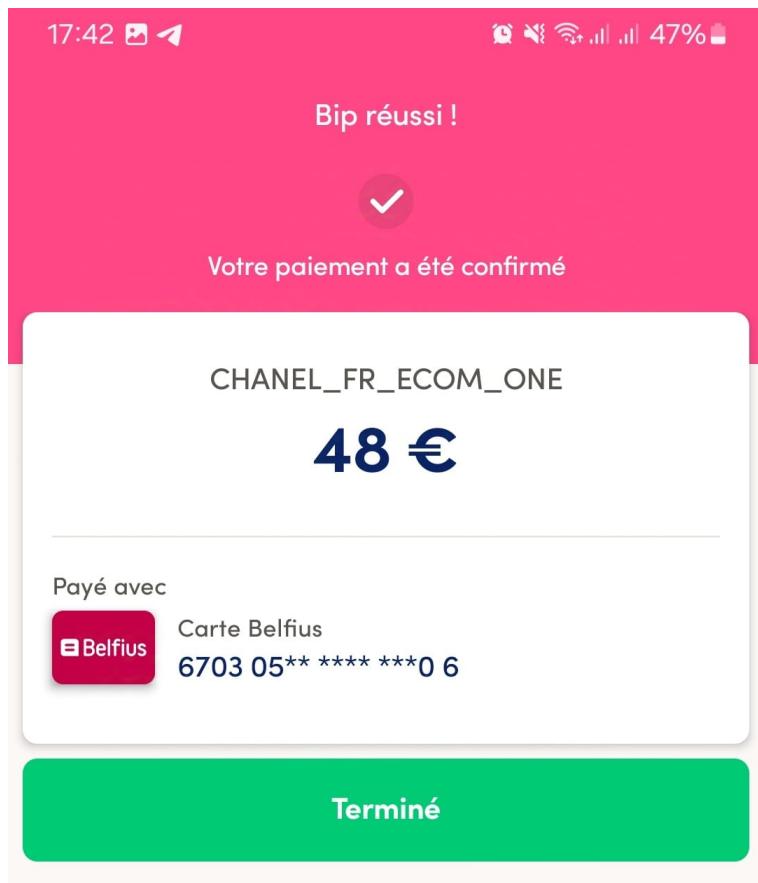


FIGURE 4.18 – Paiement réussi

Après la finalisation réussie du paiement, l'utilisateur est dirigé vers une page de confirmation de commande (*Figure 4.19*). Cette page présente un identifiant unique de suivi, facilitant le suivi de l'état de la commande.

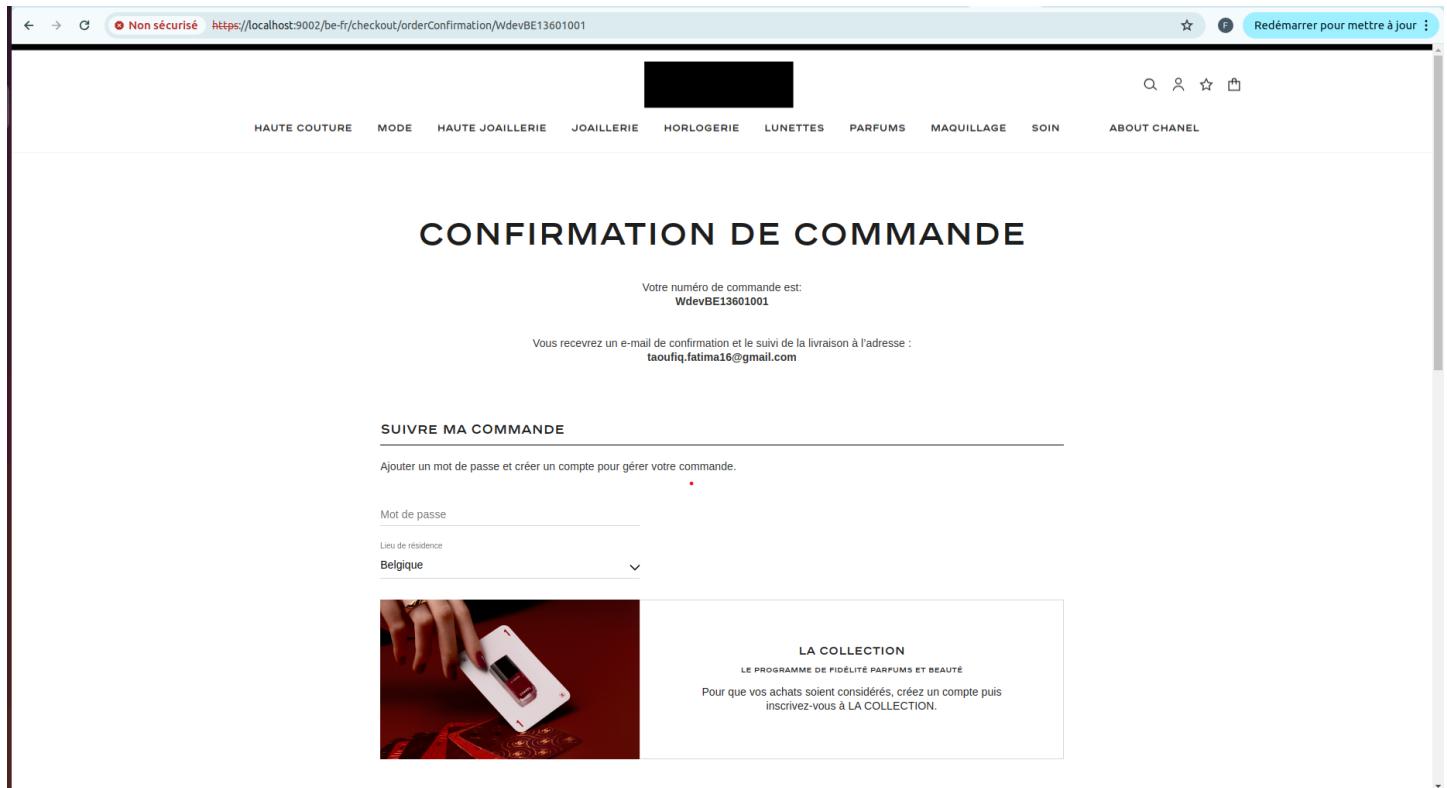


FIGURE 4.19 – Confirmation de la commande

En plus de cet identifiant, toutes les informations nécessaires sont fournies (*Figure 4.20*), telles que l'email pour le suivi de la livraison. L'interface propose également des options pour gérer la commande, notamment la possibilité de créer un compte utilisateur pour un accès facilité aux informations de commande et aux fonctionnalités associées.

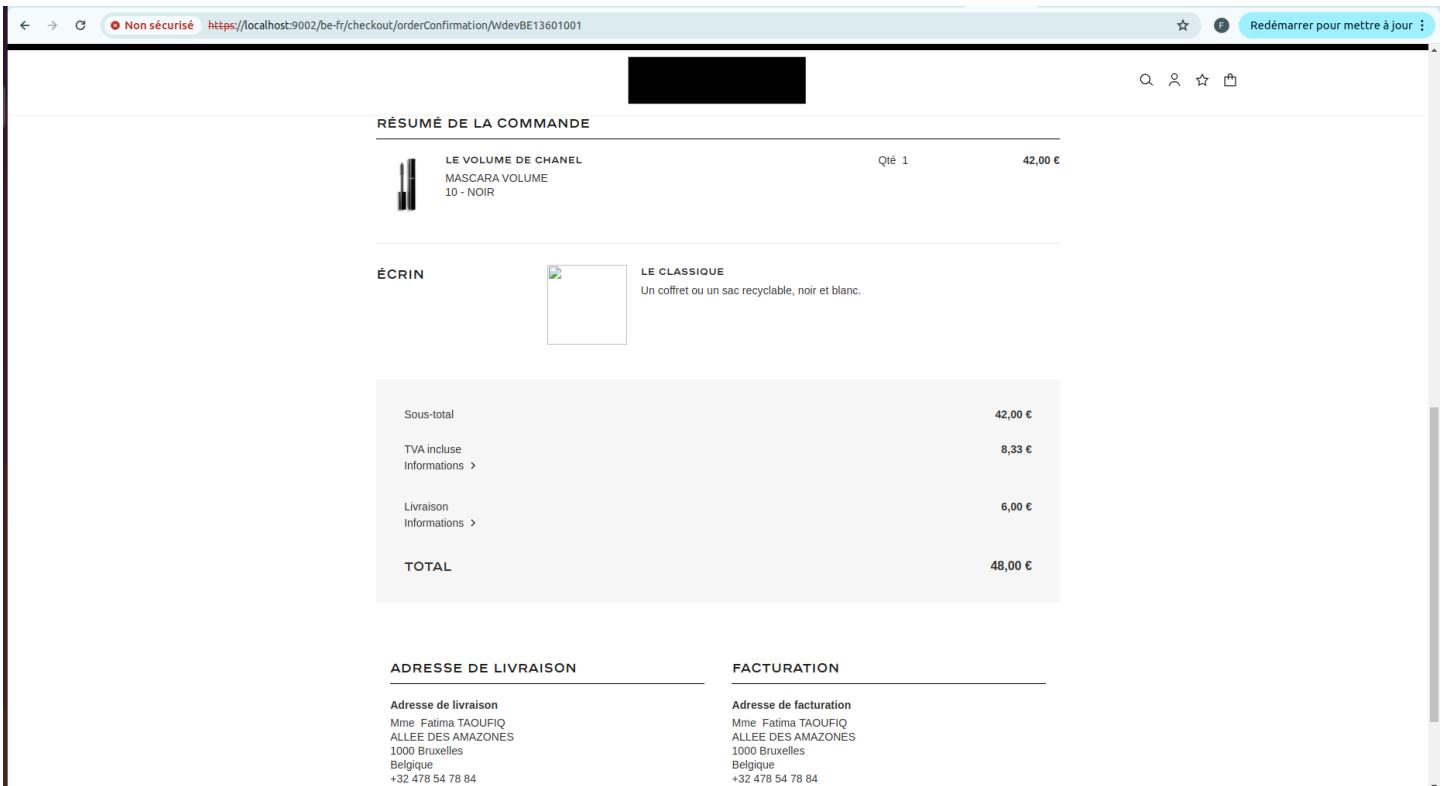


FIGURE 4.20 – Informations de la commande

En cas d'annulation ou de non-approbation du paiement, la page de confirmation de commande affichera un message d'erreur détaillé. Comme le montre la figure 4.21, un avertissement est clairement indiqué : "Remarque : les informations de paiement sont erronées. Veuillez vérifier ces informations afin de confirmer votre commande." Ce message signifie que le processus de validation du paiement n'a pas été complété correctement, nécessitant une révision des informations saisies pour procéder à la confirmation de la commande.

The screenshot shows a web browser window for <https://localhost:9002/be-fr/checkout/paymentError>. The page is titled "3. FACTURATION". It includes a summary table:

	Sous-total	55,00 €
TVA incluse Informations >		10,59 €
Livraison Informations >		6,00 €
TOTAL		61,00 €

Below the summary, there is a note: "Remarque : les informations de paiement sont erronées. Veuillez vérifier ces informations afin de confirmer votre commande."

The "MODE DE PAIEMENT" section shows payment method icons for "CARTE DE PAIEMENT", "payconiq", "PayPal", "Klarna PAY NOW", "Klarna PAY LATER", and "CHANEL GIFT CARD".

A note below the payment methods says: "Nous acceptons les modes de paiement suivants".

The "ÉCRIN" section describes "LE CLASSIQUE" as a "Un coffret ou un sac recyclable, noir et blanc".

At the bottom, there is a dropdown menu "Choisir une carte de paiement", fields for "Numéro de carte", "Date d'expiration" (Month: Mars, Year: Année), and a URL in the address bar: <https://localhost:9002/be-fr/checkout/paymentError>.

FIGURE 4.21 – Message d'erreur en cas d'annulation du paiement

Si la commande est effectuée avec succès, la figure 4.22 illustre l'interface backoffice qui confirme la création de la commande et la validation réussie du paiement. Dans cet écran, l'administrateur peut consulter les détails de la commande, s'assurer que le paiement a été correctement traité et apporter toute modification nécessaire. Cette étape est essentielle pour le suivi du processus de commande et permet de gérer efficacement les commandes après l'autorisation du paiement.

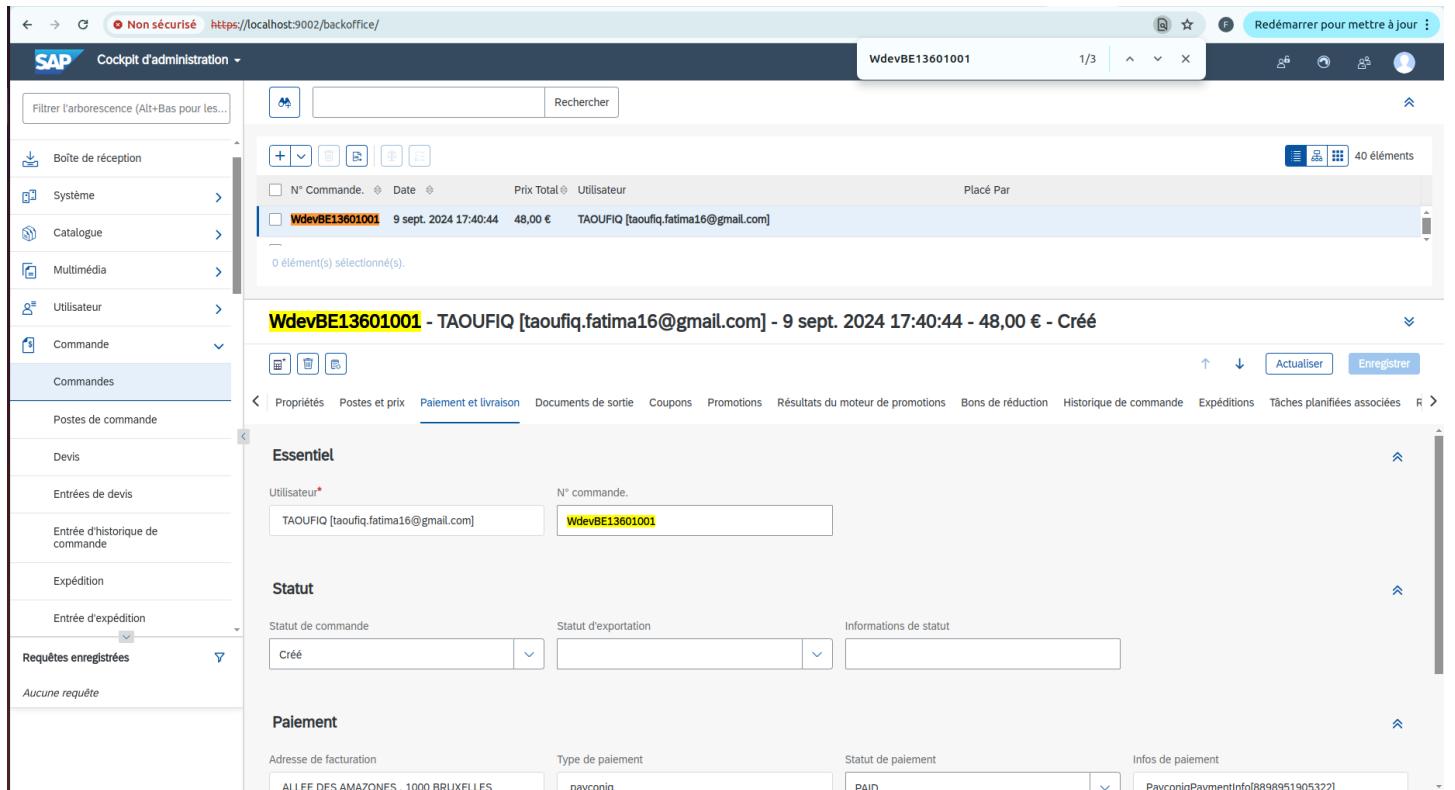


FIGURE 4.22 – Confirmation de la commande dans le backoffice

Une fois la commande validée, un e-mail est automatiquement envoyé à l'utilisateur, confirmant les détails de l'achat et assurant une traçabilité complète. Ce message inclut des informations essentielles telles que le numéro de commande, les articles achetés, le montant total, ainsi que les coordonnées de livraison. Cette étape permet à l'utilisateur de suivre sa commande de manière transparente et de vérifier les détails importants relatifs à son achat.

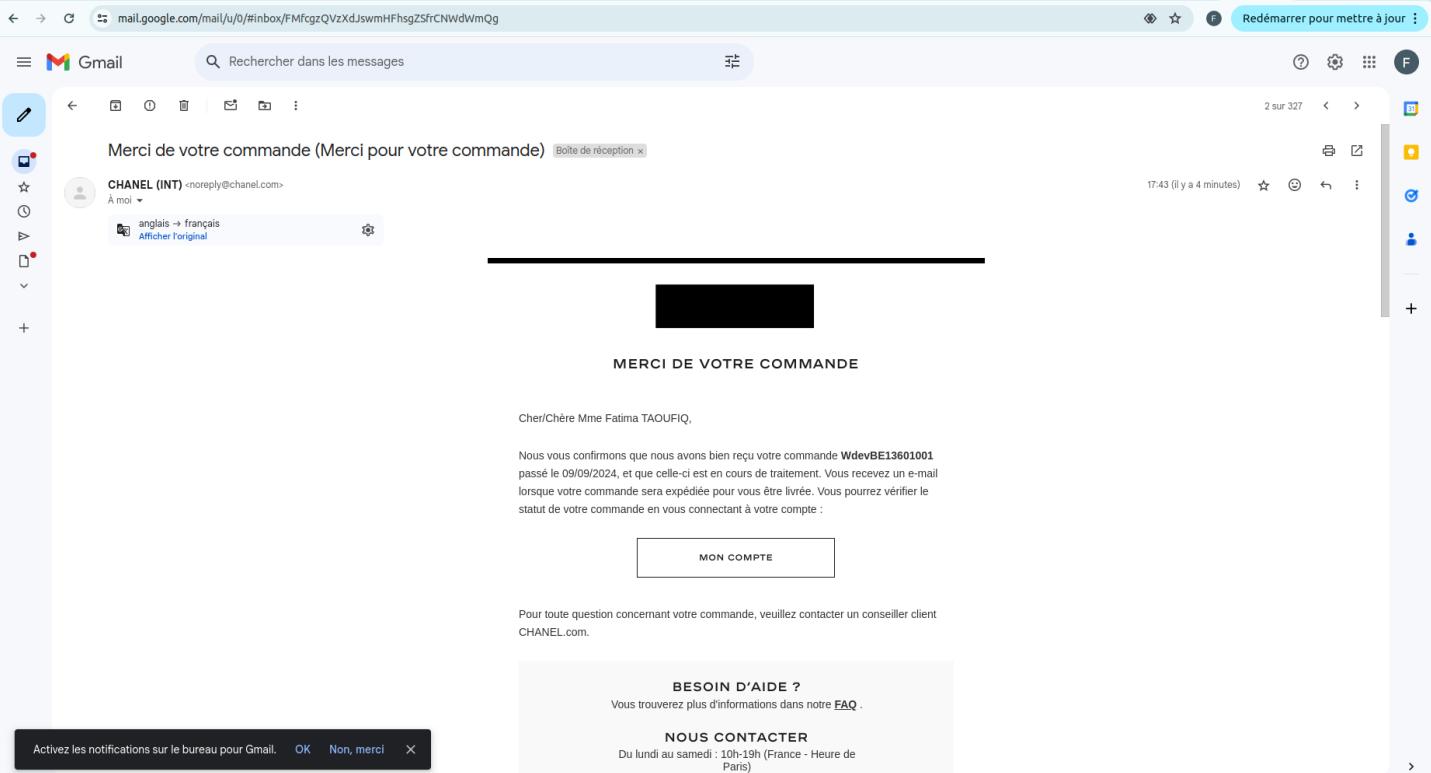


FIGURE 4.23 – Message de confirmation par email

4.3 Analyse et Correction des Anomalies

Au cours de cette période, une partie essentielle de mon travail a consisté à résoudre les problèmes détectés par SonarQube, un outil d'analyse statique qui permet d'améliorer la qualité du code. Comme illustré par les captures ci-dessous, SonarQube a mis en évidence plusieurs anomalies critiques, notamment des duplications de littéraux et des complexités cognitives élevées dans certaines méthodes du projet. Ces problèmes, jugés critiques, nécessitaient une attention immédiate pour garantir une bonne maintenabilité du projet.

En corrigeant ces anomalies, j'ai non seulement amélioré la maintenabilité du projet, mais aussi réduit les risques d'erreurs futures. Ce travail d'optimisation a permis de renforcer la robustesse du code et d'assurer que le projet respecte les bonnes pratiques de développement logiciel, en ligne avec les normes de qualité du secteur.

custom/.../chanel/hybris/yapi/controllers/SearchOverlayController.java		
<input type="checkbox"/> Define a constant instead of duplicating this literal "axisMap" 10 times.	8 months ago ▾ L264 ⚡ ⚡	design ▾
<input type="checkbox"/> Code Smell ⚡ Critical ⚡ Open ▾ nassit.raja@chanel.com ▾ 22min effort Comment		
<input type="checkbox"/> Define a constant instead of duplicating this literal "redirectUrl" 19 times.	8 months ago ▾ L265 ⚡ ⚡	design ▾
<input type="checkbox"/> Code Smell ⚡ Critical ⚡ Open ▾ nassit.raja@chanel.com ▾ 40min effort Comment		
<input type="checkbox"/> Refactor this method to reduce its Cognitive Complexity from 140 to the 15 allowed.	8 months ago ▾ L364 ⚡ ⚡	brain-overload ▾
<input type="checkbox"/> Code Smell ⚡ Critical ⚡ Open ▾ nassit.raja@chanel.com ▾ 2h10min effort Comment		
<input type="checkbox"/> Define a constant instead of duplicating this literal "redirectModel" 3 times.	8 months ago ▾ L387 ⚡ ⚡	design ▾
<input type="checkbox"/> Code Smell ⚡ Critical ⚡ Open ▾ nassit.raja@chanel.com ▾ 8min effort Comment		
<input type="checkbox"/> Define a constant instead of duplicating this literal "product" 4 times.	8 months ago ▾ L434 ⚡ ⚡	design ▾
<input type="checkbox"/> Code Smell ⚡ Critical ⚡ Open ▾ nassit.raja@chanel.com ▾ 10min effort Comment		
<input type="checkbox"/> Refactor this method to reduce its Cognitive Complexity from 140 to the 15 allowed.	8 months ago ▾ L933 ⚡ ⚡	brain-overload ▾
<input type="checkbox"/> Code Smell ⚡ Critical ⚡ Open ▾ nassit.raja@chanel.com ▾ 2h10min effort Comment		

FIGURE 4.24 – Bugs détectés par SonarQube

Pour corriger ces bugs critiques, plusieurs bonnes pratiques ont été mises en œuvre, notamment :

- **Extraction des chaînes de caractères répétitives** : Les chaînes de caractères fréquemment répétées dans le code ont été isolées et déclarées sous forme de variables constantes, réduisant ainsi la redondance et facilitant la maintenance.
- **Réduction de la complexité des méthodes** : Les méthodes présentant une complexité cognitive élevée ont été découpées en sous-méthodes plus simples et cohérentes, améliorant ainsi la lisibilité du code et facilitant sa compréhension.
- **Refactorisation du code en suivant le principe DRY (Don't Repeat Yourself)** : Les portions de code dupliquées ont été consolidées et regroupées dans des fonctions réutilisables, ce qui a permis de limiter les répétitions et d'optimiser la structure globale du projet.

Suite à une analyse avec SonarLint, nous avons vérifié que toutes les anomalies critiques ont bien été corrigées.

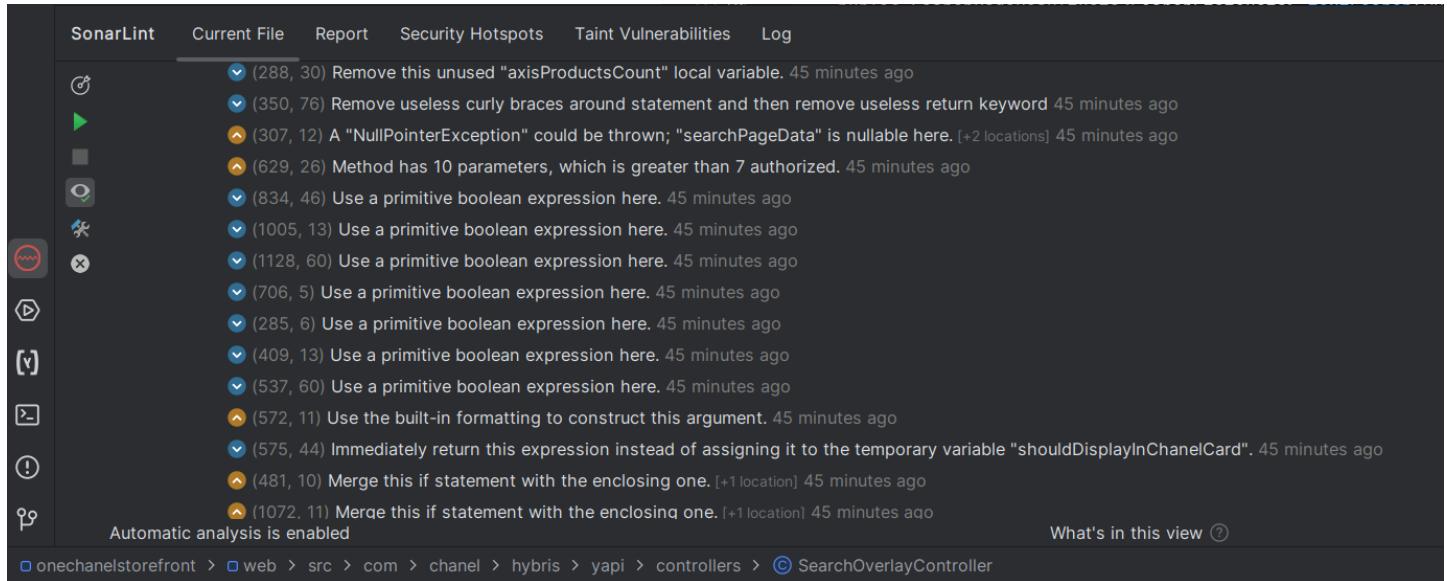


FIGURE 4.25 – Analyse avec SonarLint après correction

Dans le cadre de notre analyse, nous identifions si le problème provient de notre côté, au niveau du backend, ou s'il est lié à une autre équipe. Si le problème est de notre responsabilité, nous procédons à la correction. Dans le cas où il est lié à une autre équipe, nous demandons l'association du ticket à cette dernière et veillons à leur transmettre toutes les informations nécessaires pour faciliter leur analyse.

La figure 4.26 montre un exemple de ticket de bug utilisé pour suivre les anomalies détectées. La gestion des bugs est cruciale pour identifier et résoudre les problèmes techniques, garantissant ainsi la stabilité et la qualité du produit. Ce processus favorise également la collaboration entre les équipes, contribuant à améliorer l'expérience utilisateur.

BUG_1280872

1280872 INT1>Wishlist>CN>Error displayed when clicking on the glasses PDP and images are not displayed for all products on the PDP and wishlist pag

Sultan KOCAHAL 3 comments Add tag

State	<input checked="" type="radio"/> Need information	Area	One\2022\Experience\Seasonal Events	
Reason	MOVED	Moved out of state De...	Itegration	One\2024\IC3\Sprint 131

Discussion

Add a comment. Use # to link a work item, ! to link a pull request, or @ to mention a person.

Fatima TAOUIQ commented jeudi (edited)

After analysis, it appears that the issue related to the display of the images is due to a data problem.
@Sara ELJABARI @Sanae SRAIDI @Sultan KOCAHAL

FIGURE 4.26 – Exemple ticket de debugging

4.4 Tests et Validation

4.4.1 Tests

Les tests constituent une étape essentielle pour s'assurer du bon fonctionnement du système et de sa conformité aux attentes. Ils aident à repérer et corriger d'éventuelles anomalies avant la mise en production. Nous avons ainsi mené différents types de tests, allant des tests unitaires aux tests fonctionnels, réalisés sur plusieurs environnements :

- Tests unitaires :** Nous avons utilisé JUnit pour mener les tests unitaires, visant à valider chaque composant du système de manière isolée. Ces tests, réalisés en local, permettent de vérifier que les méthodes fonctionnent comme prévu, de gérer les exceptions, et d'assurer la robustesse du code avant toute intégration. Grâce à une approche rigoureuse, nous avons atteint une couverture de 100% du code, comme le montrent les captures d'écran ci-dessous. Cette couverture complète assure que chaque ligne de code a été testée et contribue à la fiabilité globale du système.

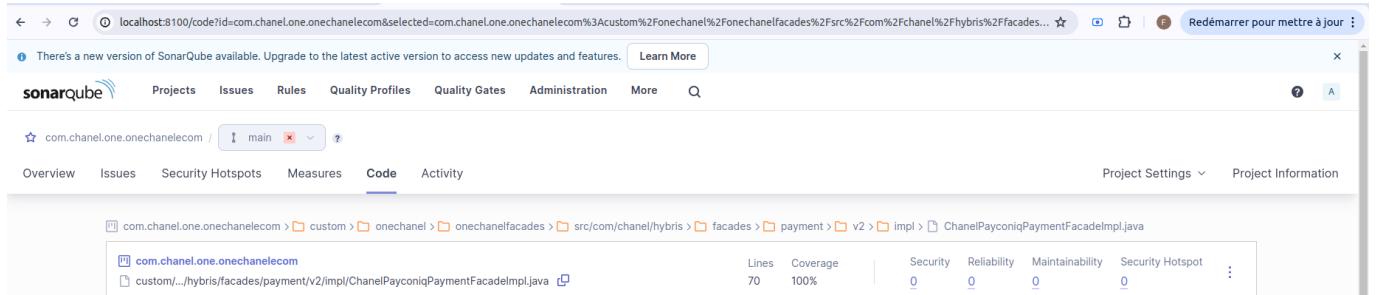


FIGURE 4.27 – Couverture de code à 100%

Name	Tests	Errors	Failures	Skipped	Time(s)	Time Stamp	Host
ChanelPayconiqPaymentFacadeImplTest	8	0	0		0.429	2024-09-12T21:38:19	RAB7F4HK13

Tests

Name	Status	Type	Time(s)
authorize_empty_RequestParams	Success		0.406
startSession_empty_RequestParams	Success		0.002
authorize_response_success_false	Success		0.003
startSession_response_success_false	Success		0.002
startSession_validateCartBeforeAuth_RedirectErreur	Success		0.002
startSession_command_exception	Success		0.003
startSession_response_success	Success		0.002
authorize_response_success	Success		0.002

[Properties »](#)

FIGURE 4.28 – Succès de tous les cas de test

- **Tests fonctionnels :** Les tests fonctionnels ont été pris en charge par l'équipe QA sur plusieurs environnements, tels que INT2, INT1, UAT et OAT. Ces tests permettent de valider que les fonctionnalités développées respectent les exigences métiers et opèrent correctement dans des conditions réalistes. INT1 est utilisé comme environnement partagé par toutes les équipes, tandis que INT2 est dédié à l'équipe Cart & Checkout Payment. Ces validations sont cruciales pour garantir la stabilité et la conformité du système avant sa mise en production.

4.4.2 Validation des besoins fonctionnels

Les besoins fonctionnels ont été validés avec succès sur l'environnement INT1. À ce stade, nous attendons la finalisation des tests dans tous les environnements pour pouvoir procéder au déploiement de la fonctionnalité en production. En parallèle, une validation finale est effectuée par le Delivery Manager afin de garantir que les objectifs fonctionnels et les exigences métiers sont entièrement satisfaits avant la mise en production.

Il est important de noter que la validation des besoins par le client est une étape cruciale qui n'a pas encore été réalisée. Nous planifions d'organiser une session de validation avec le client pour obtenir leur approbation formelle des besoins fonctionnels. Cette validation permettra de confirmer que les exigences sont bien comprises et acceptées avant le passage en production, garantissant ainsi que les attentes du client sont pleinement respectées.

4.4.3 Validation des besoins non fonctionnels

Pour garantir que le système répond aux critères essentiels de performance, il est crucial de valider les exigences non fonctionnelles définies préalablement.

- **Sécurité :** La sécurité des paiements est garantie par l'utilisation de l'API d'Adyen, qui intègre des mécanismes de sécurité avancés. Adyen utilise des protocoles de cryptage de bout en bout pour protéger les données sensibles des transactions. En outre, l'API d'Adyen respecte les exigences de la directive européenne PSD2 en offrant une authentification forte du client (SCA). Cette authentification peut inclure des facteurs biométriques, des mots de passe ou des codes à usage unique, assurant ainsi une protection renforcée contre les fraudes et garantissant un niveau de sécurité optimal pour chaque transaction traitée.

- **Maintenabilité :** La maintenabilité du système est assurée grâce à une architecture modulaire et une documentation complète, facilitant la gestion des mises à jour et des correctifs. Les processus de déploiement automatisés minimisent les risques d'erreurs et permettent une intégration fluide des nouvelles fonctionnalités. Cette approche garantit une gestion efficace des évolutions et des corrections nécessaires au bon fonctionnement du système.
- **Disponibilité :** La disponibilité du système est optimisée par la configuration de déploiements répartis sur plusieurs clusters géographiques. Le projet One utilise cinq clusters situés en Amérique (AMER), en Europe, au Moyen-Orient et en Afrique (EMEA), en Asie-Pacifique, en Australie et au Canada (APAC), en Russie et en Chine. Cette répartition permet d'assurer une haute disponibilité et une résilience accrue du système, offrant une réponse rapide aux demandes des utilisateurs, indépendamment de leur emplacement. Grâce à cette architecture distribuée, le système bénéficie d'une latence réduite et d'un temps de réponse optimisé. En cas de panne d'un cluster, le trafic est automatiquement redirigé vers les autres clusters, assurant ainsi la continuité du service et minimisant les interruptions. Cette approche contribue à maintenir une expérience utilisateur fluide et efficace.

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

Ce rapport a détaillé les résultats de mon projet de fin d'études, centré sur l'amélioration d'une plateforme e-commerce par l'intégration de la méthode de paiement Payconiq, spécialement adaptée au marché belge. L'objectif principal était de moderniser la plateforme pour optimiser l'expérience utilisateur et répondre aux exigences locales.

Au cours du projet, j'ai exploré divers aspects du développement de systèmes, tels que la planification technique, la conception, le déploiement et les tests. Cette expérience m'a permis d'appliquer les connaissances acquises à l'INPT tout en découvrant de nouvelles technologies dans un environnement professionnel. Mon rôle au sein de l'équipe de développement a été particulièrement enrichissant, m'offrant une perspective concrète sur la gestion de projets réels.

Bien que mon implication dans ce projet se termine ici, plusieurs pistes d'amélioration pour la plateforme e-commerce restent ouvertes. Il serait pertinent de poursuivre l'optimisation des processus de paiement, d'intégrer de nouvelles méthodes de paiement et de renforcer la sécurité. Ces développements permettront à la plateforme de rester compétitive et de s'adapter aux évolutions du marché et des technologies.

Les compétences acquises lors de ce projet seront précieuses pour mes futurs projets, notamment dans le cadre de mes prochaines missions avec SQLi. Ce projet de fin d'études a non seulement permis d'améliorer la plateforme e-commerce, mais a également constitué une étape significative dans ma formation, me préparant pour de nouveaux défis professionnels.

Références

- [1] *SQLI*,
<https://www.sqlil.com/ma-fr>
- [2] *les valeurs du groupe SQLI* ,
<https://www.sqlil.com/ma-fr/cariere/les-valeurs-sqli>
- [3] *Utilisateur de Payconiq*,
<chrome-extension://dbkidnlfklnjanneifjojofckpcogcl/pdf-viewer.html?file=https%3A%2F%2Feu-01.kc-usercontent.com%2F331a0185-e26c-01d0-41e9-c42bc79a1d2c%2F6bf42db6-6363-4041-b9d3-54c6548f8e14%2FPress%2520Release-22Feb2024.pdf>
- [4] *Justification de l'intégration de Bancontact by Payconiq*,
<https://www.payconiq.be/fr/professionnel/pourquoi-choisir-payconiq>
- [5] *SAP*,
<https://help.sap.com/docs/>
- [6] *JEE*,
<https://www.neosoft.fr/nos-publications/blog-tech/les-bases-de-la-securite-en-developpement-java-ee/>
- [7] *Spring Framework Documentation*,
<https://docs.spring.io/spring-framework/reference/index.html>
- [8] *SonarQube Documentation*,
<https://docs.sonarsource.com/sonarqube/latest/>
- [9] *Apache Ant Manual*,
<https://ant.apache.org/manual/index.html>
- [10] *JUnit 4 Documentation*,
<https://junit.org/junit4/javadoc/latest/>