



# Dédicace

Chers membres de nos familles et chers amis,

Nous souhaitons dédier ce modeste travail à chacun d'entre vous, qui avez été une source inestimable de soutien et de motivation tout au long de cette expérience. Vos encouragements constants, votre confiance indéfectible en nos capacités et vos conseils précieux ont joué un rôle déterminant dans notre réussite.

Nous tenons à exprimer notre profonde gratitude pour votre patience, votre compréhension et votre générosité tout au long de ce parcours. Vous avez été notre moteur lors des moments difficiles, nous encourageant à surmonter les obstacles et les défis qui se sont dressés devant nous. Grâce à votre présence bienveillante, nous avons pu avancer avec confiance et persévérance.

Nous espérons sincèrement que ce travail saura répondre à vos attentes et vous plaira. Nous avons investi un effort considérable pour produire un travail de qualité, et cela n'aurait pas été possible sans votre soutien inconditionnel. Votre confiance en nos capacités nous a inspirés et motivés à donner le meilleur de nous-mêmes.

Nous tenons à vous exprimer notre gratitude la plus profonde pour votre soutien indéfectible. Nous sommes extrêmement reconnaissants d'avoir pu compter sur votre présence à nos côtés tout au long de ce périple. Nous espérons sincèrement que ce travail sera une source de satisfaction pour vous, car votre bonheur et votre satisfaction sont pour nous une récompense inestimable.

Encore une fois, du fond du cœur, nous vous remercions pour votre amour, votre encouragement et votre soutien sans faille. Nous sommes bénis de vous avoir dans nos vies et espérons que ce travail témoignera de notre appréciation sincère.

# Remerciements

Nous sommes extrêmement reconnaissants de pouvoir exprimer nos sincères remerciements et notre profonde gratitude à toutes les personnes qui ont contribué de près ou de loin à la réalisation de notre projet de fin d'année.

Tout d'abord, nous aimerions adresser nos plus sincères remerciements à M. ERRAHA Brahim, notre professeur et encadrant au sein de l'école. Nous sommes extrêmement reconnaissants pour le temps qu'il nous a consacré, les précieux conseils qu'il nous a prodigués et sa volonté de partager son expertise avec nous. Son soutien inconditionnel tout au long du projet a été d'une valeur inestimable.

Nous souhaitons également exprimer notre profonde reconnaissance à M. ESSAFA Mohamed, notre encadrant au sein de l'entreprise et notre maître de stage. Nous sommes très reconnaissants de l'opportunité qui nous a été offerte de mener à bien ce projet au sein de leur société. La confiance qu'ils ont placée en nous et leur soutien constant ont été des éléments clés de notre réussite.

Nous tenons également à remercier chaleureusement M. MAMARCH Abdelkarim, le Président Directeur général de l'entreprise, pour nous avoir accueillis et avoir rendu possible la réalisation de ce projet. Sa vision et son leadership ont été une source d'inspiration pour nous tout au long du processus.

Enfin, nous souhaitons exprimer notre profonde gratitude envers tous les membres du jury pour l'honneur qu'ils nous font en acceptant de juger notre travail. Leur expertise et leurs commentaires constructifs vont toujours contribuer à notre apprentissage et à notre développement professionnel.

Nous sommes réellement reconnaissants envers chacune de ces personnes pour leur contribution inestimable à notre projet. Leur soutien, leur mentorat et leur confiance ont été essentiels à notre succès. Nous sommes profondément reconnaissants et honorés d'avoir eu l'opportunité de travailler avec eux et de bénéficier de leur expertise.

# Résumé

Le présent mémoire décrit le travail réalisé dans le cadre de notre projet de fin d'année, effectué au sein de la société STOREINO. Notre projet vise à développer une plateforme en ligne permettant d'améliorer les processus d'affiliation des produits par les marketeurs affiliés.

Avec la préférence croissante des consommateurs pour les achats en ligne, STOREINO souhaite capitaliser sur cette tendance en créant un site web pour les affiliés. Ce site permettra aux affiliés de promouvoir les produits de STOREINO et de recevoir une commission sur les ventes réalisées grâce à leurs efforts de marketing. L'objectif est donc d'accélérer le taux de ventes tout en offrant aux affiliés l'opportunité de tirer profit de cette activité.

Pour mener à bien ce projet, nous avons commencé par réaliser une analyse conceptuelle des besoins en utilisant le langage de modélisation UML. Cette étape nous a permis de comprendre les exigences fonctionnelles et non fonctionnelles de la plateforme d'affiliation.

Ensuite, nous avons entamé la phase de réalisation, où nous avons présenté l'environnement matériel, technologique, logiciel et les principales interfaces de notre travail. Nous avons sélectionné les outils et les technologies appropriés pour développer le site web d'affiliation, en veillant à ce qu'il réponde aux exigences identifiées précédemment.

Au cours de cette phase, nous avons travaillé sur la conception et le développement de la plateforme, en mettant l'accent sur l'expérience utilisateur, la sécurité des transactions et la convivialité du site. Nous avons également intégré un système de suivi des ventes et de génération de rapports pour permettre aux affiliés de suivre leurs performances et leurs gains.

Enfin, nous avons procédé à des tests approfondis pour vérifier la robustesse et la fiabilité de la plateforme. Des tests d'intégration, de performance et de sécurité ont été effectués afin de garantir un fonctionnement optimal du site web.

Mots-clés : UML, marketeurs affiliés.

# Abstract

This thesis describes the work carried out as part of our end-of-year project, carried out within the company STOREINO. Our project aims to develop an online platform to improve affiliate product affiliation processes.

With the growing consumer preference for online shopping, STOREINO wants to capitalize on this trend by creating a website for affiliates. This site will allow affiliates to promote STOREINO products and earn a commission on sales made through their marketing efforts. The objective is therefore to accelerate the rate of sales while offering affiliates the opportunity to profit from this activity.

To carry out this project, we started by carrying out a conceptual analysis of the needs using the UML modeling language. This step allowed us to understand the functional and non-functional requirements of the affiliate platform.

Then, we started the realization phase, where we presented the material, technological and software environment and the main interfaces of our work. We have selected the appropriate tools and technologies to develop the affiliate website, ensuring that it meets the requirements identified previously.

During this phase, we worked on the design and development of the platform, with a focus on user experience, transaction security, and site usability. We have also integrated a sales tracking and reporting system to allow affiliates to track their performance and earnings.

Finally, we conducted extensive testing to verify the robustness and reliability of the platform. Integration, performance, and security tests have been carried out in order to guarantee the optimal functioning of the website.

Keywords : UML, Affiliate.

# Liste des abréviations

Abréviation	Signification
API	Application programming interface
CSS	Cascading Style Sheets
DOM	Document object model
GUI	Graphical user interface
HTML	Hypertext Markup Language
HTTP	Hypertext transfer protocol
IDE	Integrated Development Environment
IoT	Internet of things
JS	Javascript
JWT	JSON Web Tokens
MVC	Model View Controller
npm	Node Package Manager
ODM	Object document mapper
ORM	Object-relational mapping
PFA	Projet de fin d'année
Sass	Syntactically Awesome Style Sheets
SGBD	Système de gestion de base de données
UML	Unified Method Language
URL	Uniforme resource locator
VS Code	Visual Studio Code

# Table des figures

FIGURE 1: METHODES DE PAIEMENT QU'OFFRE STOREINO .....	16
FIGURE 2 : PARTENAIRES DE LIVRAISON STROEINO .....	17
FIGURE 3 : PLANIFICATION PAR L'OUTIL TRELLO.....	26
FIGURE 4: METHODE EN CASCADE .....	27
FIGURE 5 : DIAGRAMME DE GANTT .....	28
FIGURE 6 : DIAGRAMME D'HERITAGE ENTRE LES ACTEURS .....	31
FIGURE 7 : DIAGRAMME DE CAS D'UTILISATION D'UN VISITEUR .....	31
FIGURE 8 : 1ERE PARTIE DU DIAGRAMME DE CAS D'UTILISATION D'UN AFFILIE(AFFILIATE) .....	32
FIGURE 9 : 2EME PARTIE DU DIAGRAMME DE CAS D'UTILISATION D'UN AFFILIE(AFFILIATE).....	32
FIGURE 10 : DIAGRAMME DE CAS D'UTILISATION D'UN RESPONSABLE DE PRODUITS(PRODUCT MANAGER).....	33
FIGURE 11 : DIAGRAMME DE CAS D'UTILISATION D'UN RESPONSABLE DE COMPTES(ACCOUNT MANAGER) .....	33
FIGURE 12 : DIAGRAMME DE CAS D'UTILISATION D'UN RESPONSABLE DE COMMANDES(ORDER MANAGER) .....	33
FIGURE 13 : DIAGRAMME DE CAS D'UTILISATION D'UN FOURNISSEUR(SUPPLIER).....	34
FIGURE 14 : DIAGRAMME DE CLASSE, PARTIE UTILISATEUR.....	35
FIGURE 15 : DIAGRAMME DE CLASSE, PARTIE AFFILIE.....	36
FIGURE 16 : DIAGRAMME DE CLASSE, PARTIE VISITEUR.....	36
FIGURE 17 : DIAGRAMME DE CLASSE, PARTIE CUSTOMER.....	37
FIGURE 18 : DIAGRAMME DE CLASSE DES INFORMATIONS DE PAIEMENT D'UN AFFILIE(AFFILIATE) .....	37
FIGURE 19 : DIAGRAMME DE CLASSE, PARTIE PRODUIT .....	38
FIGURE 20 : DIAGRAMME DE CLASSE, PARTIE FACTURATION.....	38
FIGURE 21 : DIAGRAMME DE CLASSE DANS SON ENSEMBLE .....	39
FIGURE 22 : DIAGRAMME DE SEQUENCE D'INSCRIPTION D'UN VISITEUR .....	40
FIGURE 23 : DIAGRAMME DE SEQUENCE D'AUTHENTIFICATION .....	41
FIGURE 24 : DIAGRAMME DE SEQUENCE DE VERIFICATION DES ROLES .....	42
FIGURE 25 : DIAGRAMME DE SEQUENCE, RECUPERATION DE LA LISTE DES UTILISATEURS.....	42
FIGURE 26 : DIAGRAMME DE SEQUENCE, RECUPERATION D'UN UTILISATEUR PAR SON CODE.....	43
FIGURE 27 : DIAGRAMME DE SEQUENCE, MISE A JOUR D'UN UTILISATEUR .....	43
FIGURE 28 : DIAGRAMME DE SEQUENCE, SUPPRESSION D'UN UTILISATEUR.....	44
FIGURE 29 : DIAGRAMME DE SEQUENCE, RECUPERATION DES INFORMATIONS DE PAIEMENT MAD D'UN AFFILIE.....	44
FIGURE 30 : DIAGRAMME DE SEQUENCE, RECUPERATION DES INFORMATIONS DE PAIEMENT USD D'UN AFFILIE .....	45
FIGURE 31 : DIAGRAMME DE SEQUENCE, SUPPRESSION DES INFORMATIONS DE PAIEMENT MAD D'UN AFFILIE .....	45
FIGURE 32 : DIAGRAMME DE SEQUENCE, SUPPRESSION DES INFORMATIONS DE PAIEMENT MAD D'UN AFFILIE.....	46
FIGURE 33 : DIAGRAMME DE SEQUENCE, RECUPERATION DES PROMOTIONS D'UN AFFILIE.....	47
FIGURE 34 : DIAGRAMME DE SEQUENCE, RECUPERATION DE LA LISTE DES PRODUITS .....	48
FIGURE 35 : DIAGRAMME DE SEQUENCE, RECUPERATION D'UN PRODUIT.....	48
FIGURE 36 : DIAGRAMME DE SEQUENCE, CREATION D'UN PRODUIT .....	49
FIGURE 37 : DIAGRAMME DE SEQUENCE, RECUPERATION DES STATISTIQUES DES PRODUITS .....	49
FIGURE 38 : DIAGRAMME DE SEQUENCE, RECUPERATION DES DEMANDES DE PROMOTIONS.....	50
FIGURE 39 : DIAGRAMME DE SEQUENCE, MISE A JOUR D'UNE DEMANDE DE PROMOTION .....	50
FIGURE 40 : DIAGRAMME DE SEQUENCE, RECUPERATION DES DONNEES PERSONNELS D'UN AFFILIE .....	51
FIGURE 41 : DIAGRAMME DE SEQUENCE, RECUPERATION DE LISTE DES OFFRES.....	51
FIGURE 42 : DIAGRAMME DE SEQUENCE, RECUPERATION DES DETAILS D'UN PRODUIT .....	52
FIGURE 43 : DIAGRAMME DE SEQUENCE, RECUPERATION DE LISTE DES COMMANDES .....	52
FIGURE 44 : DIAGRAMME DE SEQUENC, RECUPERATION DES STORES.....	53
FIGURE 45 : DIAGRAMME DE SEQUENCE, ASSOCIATION D'UN STORE AU COMPTE STROEINO NETWORK .....	53
FIGURE 46 : DIAGRAMME DE SEQUENCE, SUPPRESSION D'UN STORE.....	54
FIGURE 47 : DIAGRAMME DE SEQUENCE, RECUPERATION DE LISTE DES FACTURES .....	54
FIGURE 48 : DIAGRAMME DE SEQUENCE, RECUPERATION DES STATISTIQUES D'UN AFFILIE .....	55
FIGURE 49 : DIAGRAMME DE SEQUENCE, RECUPERATION DES INFORMATIONS DE PAIEMENT MAD D'UN AFFILIE.....	55
FIGURE 50 : DIAGRAMME DE SEQUENCE, AJOUT DES INFORMATIONS DE PAIEMENT MAD D'UN AFFILIE .....	56
FIGURE 51 : DIAGRAMME DE SEQUENCE, AJOUT DES INFORMATIONS DE PAIEMENT USD D'UN AFFILIE.....	57
FIGURE 52 : DIAGRAMME DE SEQUENCE, RECUPERATION DES INFORMATIONS DE PAIEMENT USD D'UN AFFILIE .....	57
FIGURE 53 : DIAGRAMME DE SEQUENCE, MISE A JOUR DES INFORMATIONS DE PAIEMENT USD D'UN AFFILIE .....	58
FIGURE 54 : DIAGRAMME DE SEQUENCE, RECUPERATION DE LA LISTE DES VISITEURS.....	58

FIGURE 55 : DIAGRAMME DE SEQUENCE, CREATION DE DEMANDE DE PROMOTION .....	59
FIGURE 56 : DIAGRAMME DE SEQUENCE, RECUPERATION DE LA LISTE DES PROMOTIONS D' UN AFFILIE .....	59
FIGURE 57 : DIAGRAMME DE SEQUENCE, RECUPERATION DE LA LISTE DES PRODUITS D'UN FOURNISSEUR.....	60
FIGURE 58 : DIAGRAMME DE SEQUENCE, RECUPERATION DES DETAILS D'UN PRODUIT .....	60
FIGURE 59 : DIAGRAMME DE SEQUENCE, RECUPERATION DES STATISTIQUES DE FOURNISSEUR .....	61
FIGURE 60 : DIAGRAMME DE SEQUENCE, RECUPERATION DE LA LISTE DES COMMANDES .....	61
FIGURE 61 : DIAGRAMME DE SEQUENCE, MISE A JOUR D'UNE COMMANDE .....	62
FIGURE 62 : DIAGRAMME DE SEQUENCE, RECUPERATION D'UNE LANDING-PAGE D'UN PRODUIT.....	63
FIGURE 63 : DIAGRAMME DE SEQUENCE, COMMANDER UN PRODUIT.....	63
FIGURE 64 : DIAGRAMME D'ETAT-TRANSITION D'UN UTILISATEUR .....	64
FIGURE 65 : DIAGRAMME D'ETAT-TRANSITION D'UNE PROMOTION.....	64
FIGURE 66 : DIAGRAMME D'ETAT-TRANSITION D'UNE COMMANDE.....	65
FIGURE 67 : ARCHITECTURE NODEJS.....	67
FIGURE 68 : STRUCTURE GLOBAL DU PROJET .....	75
FIGURE 69 : REPERTOIRE API.....	75
FIGURE 70 : REPERTOIRE CONFIG.....	76
FIGURE 71 : REPERTOIRE DATA.....	76
FIGURE 72 : REPERTOIRE LOGS.....	77
FIGURE 73 : REPERTOIRE MIDDLEWARE.....	77
FIGURE 74 : REPERTOIRE MODELS.....	78
FIGURE 75 : REPERTOIRE START .....	78
FIGURE 76 : REPERTOIRE ROUTES .....	79
FIGURE 77 : REPERTOIRE SERVICES .....	79
FIGURE 78 : TESTS SERVICE ET VALIDATEURS.....	80
FIGURE 79 : TESTS API ET MIDDLEWARE .....	80
FIGURE 80 : REPERTOIRE UTILITIES .....	80
FIGURE 81 : REPERTOIRE VALIDATORS .....	81
FIGURE 82 : STRUCTURE D'UN JWT .....	82
FIGURE 83 : PROCESSUS D'AUTHENTIFICATION .....	83
FIGURE 84 : PAGE DU LOGIN .....	84
FIGURE 85 : DASHBOARD D'AFFILIE .....	84
FIGURE 86 : PAGE DES PROMOTIONS D'AFFILIE.....	85
FIGURE 87 : PAGE DES OFFRES .....	85
FIGURE 88 : PAGE DE LA LISTE DES UTILISATEURS .....	86
FIGURE 89 : FORMULAIRE DE CREATION D'UN COMPTE.....	86
FIGURE 90 : PAGE DES DETAILS D'UN UTILISATEUR.....	87
FIGURE 91 : PAGE DE LA LISTE DES DEMANDES DE CREATION DE COMPTE.....	87
FIGURE 92 : FORMULAIRE DE GESTION D'UNE DEMANDE DE CREATION D'UN COMPTE .....	88
FIGURE 93 : PAGE D'UNE RESSOURCE NON AUTORISEE .....	88
FIGURE 94 : PAGE D'UNE RESSOURCE INTROUVABLE .....	89
FIGURE 95 : GESTION DES ERREURS DE VALIDATION.....	89



# Liste des tableaux

TABLEAU 1: FICHE TECHNIQUE DE LA SOCIETE STROEINO .....	15
---	----

# Table des matières

<b>Liste des abréviations.....</b>	<b>6</b>
<b>Table des figures.....</b>	<b>7</b>
<b>Liste des tableaux.....</b>	<b>9</b>
<b>Introduction générale .....</b>	<b>12</b>
<b>Chapitre 1 : Présentation de l'entreprise.....</b>	<b>14</b>
Introduction .....	15
Présentation de l'entreprise d'accueil .....	15
Histoire de l'entreprise .....	15
Fiche technique .....	15
Domaines d'expertise .....	16
Services .....	16
Méthodes de paiement .....	16
Sociétés de livraison .....	17
Conclusion du chapitre .....	17
<b>Chapitre 2 : Cahier des charges.....</b>	<b>18</b>
Spécification et analyse des besoins.....	19
Cadre générale .....	19
Démarche de travail .....	25
Conclusion du chapitre.....	28
<b>Chapitre 3 : Analyse et conception.....</b>	<b>30</b>
Analyse et conception du système .....	31
Diagramme des cas d'utilisation .....	31
Diagramme de classe .....	36
Diagramme de séquence.....	41
Diagramme d'état-transition.....	65
Conclusion du chapitre.....	65
<b>Chapitre 4 : Etude technique .....</b>	<b>67</b>
Introduction .....	68
Outils et technologies de développement.....	68
Frameworks .....	70
Outils et IDE.....	70
Bibliothèques.....	71

Bibliothèques de test .....	72
SGBDs .....	72
Conclusion du chapitre .....	72
<b>Chapitre 5 : Implémentation.....</b>	<b>74</b>
Backend .....	75
Sécurité.....	84
Front-end .....	86
Conclusion du chapitre.....	90
<b>Conclusion.....</b>	<b>90</b>
<b>Webographie et Bibliographie .....</b>	<b>91</b>

# Introduction générale

Grâce à l'avancement du développement informatique, les sites web de e-commerce ont révolutionné l'achat de produits en ligne, offrant aux consommateurs une expérience pratique, sécurisée et personnalisée. Les moteurs de recherche performants, les descriptions détaillées, les systèmes de paiement sécurisés et les options de livraison flexibles ont contribué à la croissance exponentielle de l'industrie du e-commerce.

Dans ce contexte, les entreprises de e-commerce ont saisi l'opportunité de travailler avec des sites web d'affiliation en mettant en place des programmes d'affiliation. Ces programmes permettent aux marketeurs de promouvoir les produits ou services sur leurs sites web ou leurs réseaux sociaux. En échange, les éditeurs de contenu reçoivent une commission pour chaque vente générée à partir de leur lien d'affiliation.

Dans le cadre de notre formation d'ingénieurs en deuxième année en génie informatique, nous avons effectué un stage de PFA au sein de l'entreprise STOREINO. Notre objectif principal était de créer une plateforme web d'affiliation innovante visant à faciliter les processus d'affiliation mais aussi la vente des produits.

Afin de mener à bien ce projet, nous avons adopté une approche méthodique en utilisant la méthode en cascade pour la gestion de projet et la méthode MVC comme base de notre architecture technique. Nous avons choisi d'utiliser Node JS et Vue JS pour le développement front-end, ainsi que UML pour la conception de notre référentiel.

Notre mémoire de projet est structuré en cinq chapitres. Le premier chapitre présente une vue d'ensemble de l'organisme d'accueil, ainsi qu'une brève introduction à la problématique et à la solution proposée.

Le deuxième chapitre se concentre sur notre cahier des charges, détaillant la solution que notre plateforme offre aux utilisateurs.

Dans le troisième chapitre, nous abordons l'analyse et la conception de la plateforme, en décrivant en détail les règles de gestion et les diagrammes correspondants.

Le quatrième chapitre se concentre sur l'architecture, les technologies et les outils que nous avons sélectionnés pour la réalisation de notre projet.

Enfin, le dernier chapitre décrit la mise en œuvre concrète du projet, avec des captures d'écran illustrant chaque module développé.

En conclusion, notre rapport de projet résume notre expérience de stage au sein de STOREINO et présente les résultats obtenus. Ce stage nous a permis d'acquérir une précieuse expérience professionnelle et de mettre en pratique nos connaissances théoriques pour résoudre une problématique réelle.

# Chapitre 1      Présentation de L'entreprise

---

# Introduction

Avant d'aller plus loin dans notre rapport, il est important de présenter notre projet en commençant par présenter brièvement l'entreprise d'accueil, citer ces quelques missions et s'arrêter sur ses services offerts, pour introduire par la suite la problématique, suivie par les objectifs du projet et une analyse détaillée des besoins, avant de présenter en dernier point la méthode de conduite du projet adoptée.

## 1. Présentation de l'entreprise d'accueil

"STOREINO" est une société de commerce électronique pour le paiement à la réception, fruit de plus de 10 ans d'expérience dans le domaine. Traction électronique arabe ou étrangère et contrôle de celle-ci en toute liberté a pour objectif de rendre l'e-commerce facile et simple à appliquer pour tous et toutes, afin d'avoir un impact réel sur l'avenir du commerce électronique.

### 1.1. Histoire de l'entreprise

Avec un cumule de 10 ans d'expérience en Technologie de l'Information, les fondateurs de STOREINO ont pris conscience des difficultés rencontrées par les e-commerçants au niveau du paiement. Le problème étant dû à l'absence de système de paiement à la livraison et l'inexistence d'une solution innovante pour y remédier.

STOREINO a vu le jour en 2015, comme étant la première plateforme e-commerce dotée d'un système de paiement à la livraison dans tout le continent.

Aujourd'hui, STOREINO a réussi à concrétiser le rêve de plus de 60.000 e-commerçants en leur proposant tous les services, les outils et l'assistance nécessaire ainsi que de nombreux avantages concurrentiels dont ils ont besoin pour démarrer et booster leur business en ligne.

### 1.2. Fiche technique

<b>Nom</b>	STOREINO
<b>Type</b>	SARL
<b>Adresse</b>	Technopark 1ème étage Bureau TA121 Bab Al Madina Qr, Agadir 80000
<b>Site web</b>	<a href="https://www.STOREINO.com">https://www.STOREINO.com</a>
<b>Taille</b>	Petite entreprise
<b>Année de création</b>	2015
<b>Numéro de téléphone</b>	+ 212 6 61 38 21 42

Tableau 1: Fiche technique de la société STOREINO

## 1.3. Domaines d'expertise

### Accompagnement :

STOREINO garantit un service de qualité qui apporte à ses clients une entière satisfaction : Conseil professionnel, Intervention rapide, Maintenance à distance et Formation ciblée.

### Conseil en informatique

STOREINO assure un conseil d'expertise dans le domaine informatique adapté aux besoins de ses clients et leurs projets de développement.

## 1.4. Services

### Vendre partout

Après avoir créé votre boutique une seule fois et l'avoir synchronisée avec les autres plateformes, vous pouvez commencer à vendre facilement en ligne.

### Vendre sur une boutique en ligne

Vous pouvez créer un site Web en moins de 5 minutes à partir de zéro en utilisant des outils de conception simples, conviviaux et personnalisables, sans avoir besoin d'écrire de code ou de programmer, ni d'installer de logiciel sur votre ordinateur.

### Vendre via les réseaux sociaux

Profitez de votre présence sur les réseaux sociaux, que ce soit pour partager des photos et des événements quotidiens ou pour vendre des produits. Vous pouvez facilement ajouter vos produits sur Facebook ou Instagram et convertir les visiteurs en clients potentiels vers votre magasin.

### Vendre sur les moteurs de recherche

En mettant à votre disposition une synchronisation entre le catalogue des produits de votre boutique et Google Shop, vous pouvez désormais vendre vos produits sur le moteur de recherche Google en toute simplicité.

## 1.5. Méthodes de paiement



Figure 1: Méthodes de paiement qu'offre STOREINO



## 1.6. Sociétés de livraison



Figure 2 : Partenaires de livraison STROEINO

## 1.7. Conclusion du chapitre

Ce chapitre a permis de présenter de manière concise et détaillée l'entreprise d'accueil, ses services, son domaine d'expertise et ses partenaires. En examinant ces informations, il devient évident que l'environnement dans lequel nous avons travaillé sur notre projet est extrêmement propice à la réussite et à l'innovation. L'entreprise d'accueil possède une solide réputation dans son domaine et est soutenue par un réseau de partenaires influents. Sa vaste gamme de services témoigne de son engagement à répondre aux besoins variés de ses clients. En collaborant avec une telle entreprise, nous avons eu l'opportunité de nous immerger dans un environnement dynamique et stimulant, favorisant ainsi notre propre développement professionnel et celui de notre projet.

# Chapitre 2

## Cahier Des Charges

---

## **2. Spécification et analyse des besoins**

Les discussions menées avec le service ont été essentielles pour définir les fonctionnalités clés du projet. Grâce à ces échanges, nous avons pu identifier les besoins et les attentes de l'utilisateur final, ainsi que les contraintes techniques à prendre en compte. Ces discussions ont été un processus collaboratif, permettant à l'équipe de travailler de manière étroite avec les parties prenantes afin de déterminer les éléments fondamentaux du produit final. En analysant attentivement les retours et les suggestions du service, nous avons pu affiner et affirmer notre vision, en nous assurant que les fonctionnalités intégrées répondent pleinement aux exigences du projet.

### **2.1. Cadre générale**

#### **2.1.1. Nom de projet**

Nous avons choisi de nommer notre projet "STOREINO Network", une appellation qui reflète notre vision de connecter les magasins STOREINO et de créer un réseau dynamique d'affiliés et de vendeurs.

#### **2.1.2. Description du projet**

STOREINO Network est une plate-forme numérique qui permet aux utilisateurs de démarrer, développer et faire évoluer une entreprise affiliée à partir de zéro.

#### **2.1.3. Date de début du projet**

Le projet a officiellement débuté le 22 février 2023.

#### **2.1.4. Objectifs**

Dans le cadre de ce projet, notre objectif est de développer une plateforme qui offrira aux utilisateurs la possibilité de démarrer une activité de marketing d'affiliation et de simplifier la gestion de leurs opérations. Notre objectif principal est de créer une plateforme conviviale et efficace, offrant des fonctionnalités avancées pour les affiliés et les propriétaires d'entreprise. Les objectifs spécifiques de ce projet comprennent l'autorisation des utilisateurs à démarrer une activité de marketing d'affiliation, la facilitation de la gestion des opérations des affiliés en les regroupant sur une seule plateforme et la possibilité pour les utilisateurs de lier leurs magasins STOREINO.

## Objectifs :

- Autoriser les utilisateurs à démarrer une activité de marketing d'affiliation : Notre objectif principal est de permettre aux utilisateurs de démarrer facilement et rapidement leur propre activité de marketing d'affiliation. Nous fournirons des outils intuitifs pour faciliter la création de compte, l'inscription en tant qu'affilié, et la promotion des produits ajoutés par les vendeurs aux magasins.
- Faciliter la gestion des opérations des affiliés en les regroupant sur une seule plateforme : Nous visons à simplifier la gestion des opérations des affiliés en regroupant toutes les fonctionnalités nécessaires sur une seule plateforme. Cela comprend la gestion des commissions, le suivi des performances, la gestion des paiements et les outils de rapport pour permettre aux affiliés de suivre et d'optimiser leurs activités plus facilement.
- Autoriser les utilisateurs à lier leurs magasins STOREINO : Un autre objectif clé est de permettre aux utilisateurs de lier leurs magasins STOREINO à notre plateforme. Cela offrira une intégration fluide entre la plateforme d'affiliation et les magasins des utilisateurs, facilitant ainsi la gestion des stocks, des commandes et des ventes pour les affiliés.

### 2.1.5. Cibles

Notre plateforme cible les **marketeurs affiliés** ainsi que les **fournisseurs**, leur offrant un espace centralisé pour maximiser leurs opportunités de collaboration et d'échanges dans le domaine du marketing d'affiliation.

### 2.1.6. Portée du projet

Notre plateforme, disponible dans plusieurs langues telles que l'anglais, l'arabe et le français, répondra aux besoins d'une audience diversifiée. De plus, elle prendra en charge les devises du dirham marocain et du dollar américain, facilitant ainsi les transactions et la gestion des revenus pour les utilisateurs basés au Maroc, en Côte d'Ivoire, au Sénégal, au Burkina Faso, au Mali et en Guinée Conakry. Pour une flexibilité optimale, une variété de méthodes de paiement, y compris PayPal, TransferWise, Payoneer et les virements bancaires, seront disponibles sur notre plateforme lors des transactions financières.

## 2.1.7. Détails du projet

### 2.1.7.1. Utilisateurs de la plateforme

Notre application accueille une variété d'utilisateurs, offrant une expérience adaptée à leurs besoins spécifiques. Parmi les utilisateurs, nous comptons :

- Les visiteurs (visitor)
- Les affiliés (Affiliate)
- Les fournisseurs (supplier)
- Les administrateurs

Les administrateurs peuvent occuper des rôles tels que responsable de compte(account manager), responsable de produits(product manager) ou responsable des commandes(order manager).

### 2.1.7.2. Les besoins fonctionnels

En tant que **visiteur**, vous pouvez :

- Créer un compte pour devenir affilié.
- Parcourir les pages d'accueil.
- Changer la langue de l'interface.

En tant qu'**affilié**, vous avez la possibilité de :

- Vous connecter à votre compte.
- Réinitialiser votre mot de passe.
- Visualiser votre tableau de bord.
- Parcourir toutes les offres disponibles.
- Afficher les détails de chaque offre.
- Promouvoir des produits.
- Demander des produits spécifiques.
- Accéder à votre lien de parrainage.
- Ajouter des produits à votre boutique STOREINO liée.
- Consulter la liste de vos offres approuvées.
- Voir les détails de chaque offre approuvée.
- Ajouter des noms de domaine personnalisés.
- Supprimer des noms de domaine personnalisés.

- Modifier vos noms de domaine.
- Consulter la liste de vos noms de domaine supprimés.
- Voir vos paramètres personnels.
- Modifier les informations de votre compte, telles que votre numéro de téléphone.
- Changer votre mot de passe.
- Ajouter ou modifier vos informations de facturation en USD et MAD.
- Connecter ou supprimer un magasin STOREINO.
- Consulter vos prospects.
- Afficher les statistiques des visiteurs pour chaque offre.
- Voir une liste complète de tous vos prospects par offre.
- Consulter l'historique de vos prospects.
- Visualiser vos factures en fonction de vos affiliations.
- Consulter vos statistiques personnelles.
- Exporter vos statistiques.
- Contacter notre support en cas de besoin.

En tant qu'**administrateur**, vous avez la possibilité de :

- Vous connecter à votre compte.
- Visualiser votre tableau de bord.

En tant que **responsable de produit**, vous avez la possibilité de :

- Créer de nouvelles offres.
- Supprimer ou mettre à jour des offres existantes.
- Consulter une liste de demandes.
- Approuver ou rejeter les demandes reçues.
- Visualiser toutes les offres disponibles.
- Afficher les détails de chaque offre.
- Consulter les statistiques globales.
- Afficher les statistiques spécifiques à chaque offre.

En tant que **responsable de comptes**, vous avez la possibilité de :

- Afficher tous les utilisateurs.
- Consulter les statistiques des utilisateurs.
- Visualiser toutes les demandes d'enregistrement d'utilisateurs.
- Approuver ou rejeter les demandes d'enregistrement des utilisateurs.
- Créer ou modifier d'autres comptes de gestionnaire.

En tant que **responsable des commandes**, vous avez la possibilité de :

- Afficher une liste de prospects(leads).
- Modifier le statut du prospect(lead).

En tant que **vendeur**, vous avez la possibilité de :

- Visualiser votre tableau de bord.
- Consulter vos commandes et leur statut.
- Accéder à vos produits.
- Consulter vos statistiques.
- Consulter vos prospects.
- Afficher vos actions et l'historique des actions effectuées.

### 2.1.7.3. Les besoins non fonctionnels

Les besoins non fonctionnels, également connus sous le nom de contraintes ou exigences de qualité, regroupent les aspects essentiels à prendre en compte lors du développement et de l'exploitation efficace du système. Ils englobent des éléments tels que la performance, la sécurité, la fiabilité, l'évolutivité,

l'interopérabilité et l'utilisabilité. Ces besoins jouent un rôle crucial dans la conception et l'implémentation du système, car ils définissent les attentes et les critères de réussite liés à son fonctionnement global. Ils fournissent un cadre pour garantir que le système répond aux normes de performance, de sécurité et de convivialité attendues, tout en tenant compte des contraintes spécifiques de l'environnement dans lequel il sera déployé. En accordant une attention adéquate aux besoins non fonctionnels, les développeurs peuvent s'assurer que le système répondra aux exigences globales de qualité et offrira une expérience utilisateur satisfaisante.

#### **2.1.7.3.a. Convivialité et flexibilité**

L'application doit offrir une interface conviviale, flexible, et ergonomique, afin de faciliter son utilisation par l'utilisateur. Cela comprend la mise en place de fonctionnalités intuitives, une disposition claire des éléments, et une navigation fluide pour une expérience utilisateur optimale. En privilégiant ces caractéristiques, l'application deviendra plus accessible et conviviale, permettant à l'utilisateur de l'exploiter de manière aisée et efficace.

#### **2.1.7.3.b. Rapidité**

L'application doit optimiser les processus afin de garantir des temps de réponse raisonnables, assurant ainsi une expérience utilisateur fluide et réactive. Cela implique une optimisation du code, une gestion efficace des requêtes et une utilisation judicieuse des ressources système. En veillant à ce que les tâches soient exécutées de manière efficiente, l'application pourra traiter les demandes de manière rapide et fournir des résultats dans des délais acceptables. Une attention particulière doit être accordée à l'identification et à la résolution des éventuels goulots d'étranglement ou des zones de latence afin de maintenir des temps de réponse optimaux, quel que soit le volume de données ou l'activité de l'utilisateur.

#### **2.1.7.3.c. Efficacité**

L'application doit fonctionner de manière optimale en termes d'utilisation des ressources, telles que le CPU et la mémoire, afin de garantir une performance fluide et efficace. Il est crucial d'optimiser le code et d'adopter des pratiques de programmation efficaces pour minimiser la charge sur les ressources. Cela peut inclure des techniques telles que la mise en cache des données fréquemment utilisées, la gestion appropriée des connexions réseau et la limitation des opérations gourmandes en ressources.



En utilisant les ressources de manière efficace, l'application pourra fonctionner de manière plus réactive, éviter les ralentissements et les goulots d'étranglement, et améliorer globalement l'expérience utilisateur.

#### **2.1.7.3.d. Maintenabilité**

Le code de l'application doit être soigneusement structuré, compréhensible et lisible, dans le but de faciliter les évolutions du système et de permettre une localisation et une correction aisées des erreurs. Pour cela, il est important d'adopter des conventions de nommage claires, d'utiliser des commentaires pertinents et d'organiser le code en modules logiques. En favorisant la lisibilité du code, les développeurs pourront comprendre plus facilement son fonctionnement, le maintenir et le mettre à jour efficacement au fil du temps. De plus, une structure bien organisée facilitera la collaboration entre les membres de l'équipe de développement, favorisant ainsi le partage des connaissances et l'amélioration collective du codebase.

## **2.2. Démarche de travail**

Afin d'assurer une représentation claire et compréhensible de la structure et du comportement de notre projet, nous avons fait le choix d'utiliser l'outil UML (Unified Modeling Language). En utilisant les diagrammes et les notations standardisées de l'UML, nous pouvons modéliser les différentes entités, leurs relations et les interactions entre elles, ce qui facilite la communication et la compréhension du projet. Cette approche nous permet de visualiser de manière systématique les différents aspects du projet, favorisant ainsi une meilleure collaboration et une gestion efficace des tâches.

### **2.2.1. Présentation du langage UML**

Le langage UML (langage de modélisation unifié) a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Il est destiné à l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement. L'UML a des applications qui vont au-delà du développement logiciel, notamment pour les flux de processus dans l'industrie. Il ressemble aux plans utilisés dans d'autres domaines et se compose de différents types de diagrammes. Dans l'ensemble, les diagrammes UML décrivent la limite, la structure et le comportement du système et des objets qui s'y trouvent.

### 2.2.2. Planification de travail

L'objectif fondamental de la planification des tâches est de fournir une vision holistique de notre projet ainsi que de sa progression. Cette pratique permet non seulement de superviser l'exécution de chaque tâche individuelle en tenant compte de ses ressources, de ses dates de début et de fin, ainsi que de la durée déjà consacrée, mais également de suivre l'avancement global du projet.

Chaque tâche a des répercussions sur la date de fin du projet dans son ensemble, et la planification des tâches permet d'évaluer ces conséquences. Pour atteindre ces objectifs, nous avons utilisé l'outil de gestion des tâches Trello, qui nous a permis de structurer et d'organiser efficacement nos activités tout au long du projet.

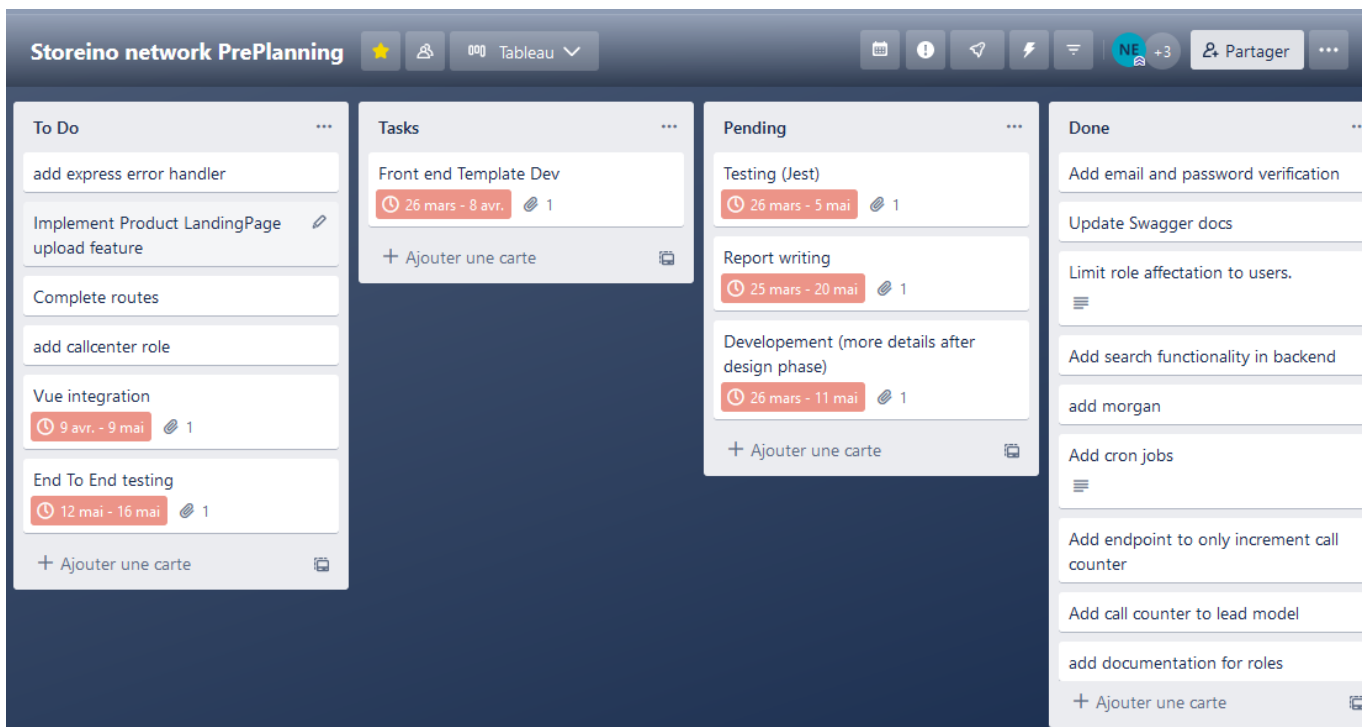


Figure 3 : Planification par l'outil Trello

### 2.2.3. Cycle de vie de développement

Nous avons choisi la méthode en cascade comme cycle de vie de développement, car nous avons réalisé une conception préalable approfondie et les besoins semblaient suffisamment clairs pour travailler selon cette approche.

La méthode en cascade est une approche linéaire où chaque phase du processus est réalisée séquentiellement, avec une phase se terminant avant que la suivante ne débute. Dans notre cas, après avoir effectué une conception détaillée du projet, nous avons une bonne compréhension des exigences et des spécifications nécessaires. Cela nous a permis de planifier les étapes de développement de manière précise, en suivant un ordre séquentiel.

L'avantage de la méthode en cascade dans cette situation était sa structure claire et son approche disciplinée. En travaillant de manière séquentielle, nous pouvions suivre un plan établi et aborder chaque phase du projet de manière systématique. Cela nous a aidés à mieux estimer les ressources nécessaires, à planifier les délais et à répartir les tâches de manière organisée.

Cependant, il est important de noter que la méthode en cascade présente certaines limites. Par exemple, une fois qu'une phase est terminée, il peut être difficile de revenir en arrière et de corriger des erreurs ou d'apporter des modifications majeures. De plus, si de nouveaux besoins ou des changements significatifs surviennent au cours du développement, il peut être difficile de les intégrer de manière fluide dans la structure en cascade.

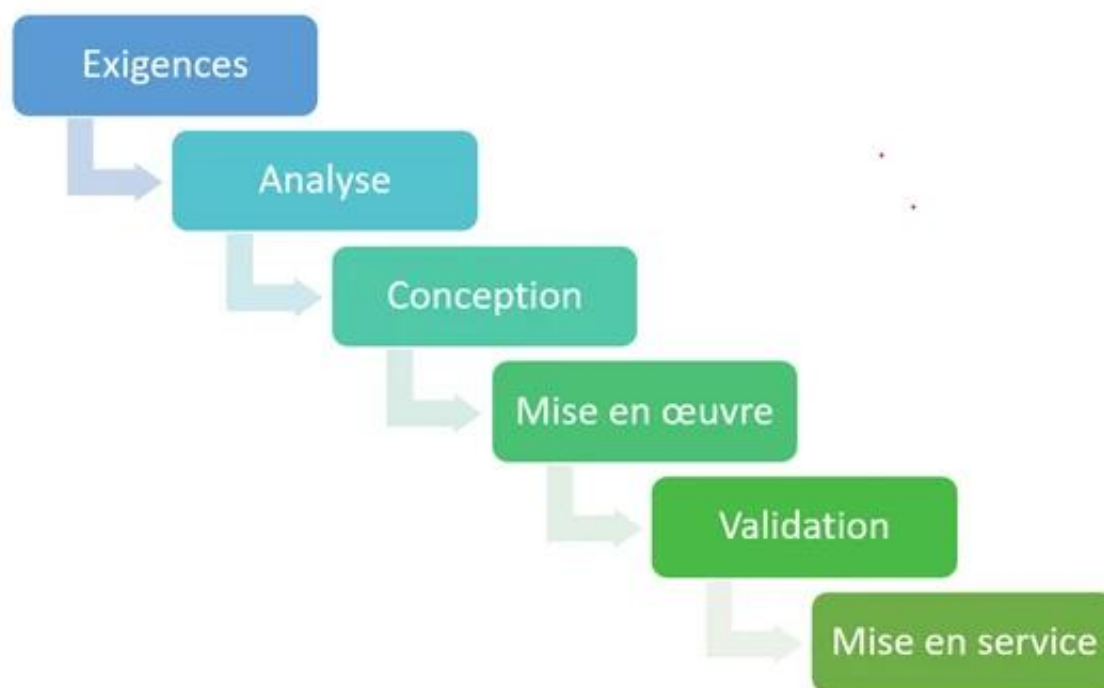


Figure 4: Méthode en cascade

#### 2.2.4. Diagramme de Gantt

Le diagramme de Gantt a été utilisé pour planifier et répartir les tâches de notre projet en fonction du temps. Toutefois, en raison de la complexité des technologies utilisées lors de la phase de développement et des contraintes temporelles, nous n'avons pas pu respecter strictement les délais initialement définis dans le diagramme.

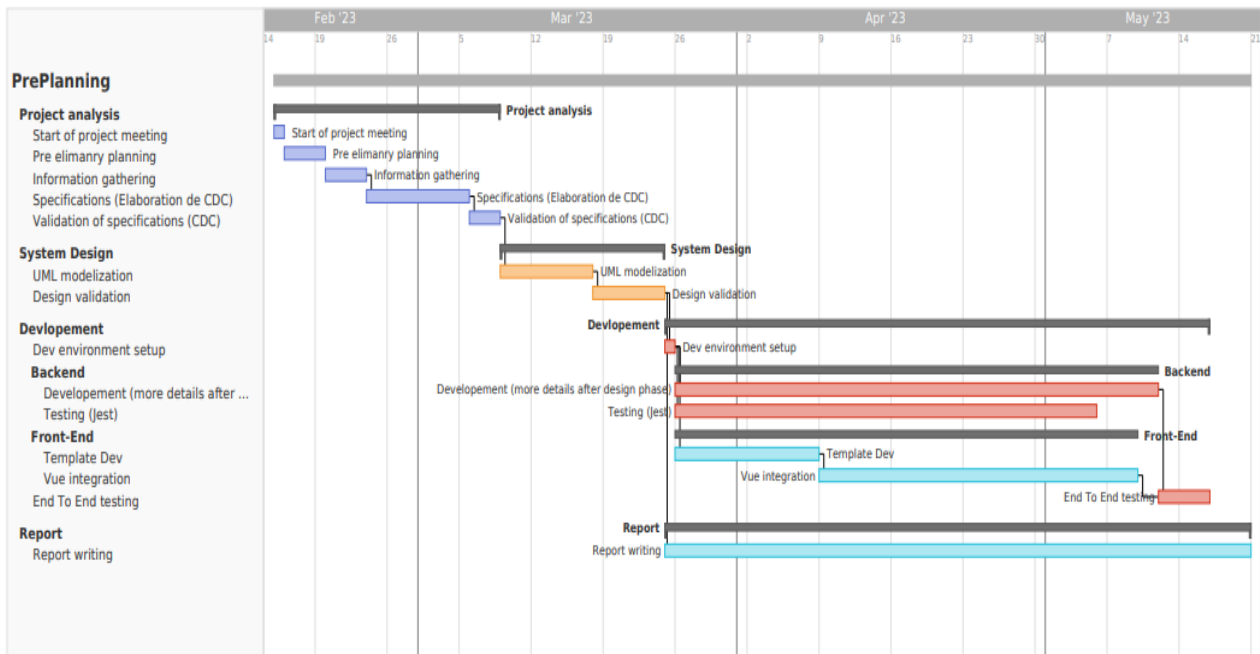


Figure 5 : Diagramme de gantt

## 2.3. Conclusion du chapitre

En conclusion, ce chapitre aborde les éléments essentiels de la spécification et de l'analyse des besoins dans le projet. Nous avons examiné en détail le cadre général du projet, la méthodologie de travail, le diagramme de Gantt et la méthode en cascade. Nous avons également exploré l'utilisation de Trello comme outil de gestion de projet, l'UML pour la modélisation, ainsi que les différents types de besoins, à savoir les besoins fonctionnels et non fonctionnels. Enfin, nous avons identifié et étudié les utilisateurs de l'application, afin de prendre en compte leurs besoins et attentes.

# Chapitre 3

## Analyse et Conception

---

## 3. Analyse et conception du système

Après avoir élaboré les aspects fondamentaux de notre projet, nous entamons maintenant la phase cruciale du cycle de vie du projet, à savoir la conception. Cette étape vise à décrire en détail le fonctionnement de l'ensemble de l'application. Dans ce chapitre, nous aborderons la conception et la modélisation de notre projet.

### 3.1. Diagramme des cas d'utilisation

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet, mais pour le développement, les cas d'utilisation sont plus appropriés. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. C'est une unité significative de travail. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs, ils interagissent avec les cas d'utilisation.

#### 3.1.1. Définitions

##### 3.1.1.1. Cas d'utilisation

Ils permettent de décrire l'interaction entre l'acteur et le système. L'idée forte est de dire que l'utilisateur d'un système logiciel a un objectif quand il utilise le système ! Le cas d'utilisation est une description des interactions qui vont permettre à l'acteur d'atteindre son objectif en utilisant le système

##### 3.1.1.2. Acteurs

Ce sont des entités externes qui interagissent avec le système, comme une personne humaine ou un robot. Une même personne (ou robot) peut être plusieurs acteurs pour un système, c'est pourquoi les acteurs doivent surtout être décrits par leur rôle, ce rôle décrit les besoins et les capacités de l'acteur.

##### 3.1.1.3. Relations

Trois types de relations sont pris en charge par la norme UML et sont graphiquement représentées par des types particuliers de ces relations. Les relations indiquent que le cas d'utilisation source présente les mêmes conditions d'exécution que le cas issu

##### 3.1.1.4. Inclusion

Le cas d'utilisation de base en incorpore explicitement un autre, de façon obligatoire, à un endroit spécifié dans ses enchaînements.

### 3.1.1.5. Extension

Le cas d'utilisation de base en incorpore implicitement un autre, de façon optionnelle, à un endroit spécifié indirectement dans celui qui procède à l'extension.

### 3.1.1.6. Généralisation

Les cas d'utilisation descendants héritent de la description de leur parent Commun. Chacun d'entre eux peut néanmoins comprendre des relations spécifiques supplémentaires avec d'autres acteurs ou cas d'utilisation.

## 3.1.2. Diagrammes

### 3.1.2.1. Les acteurs de système

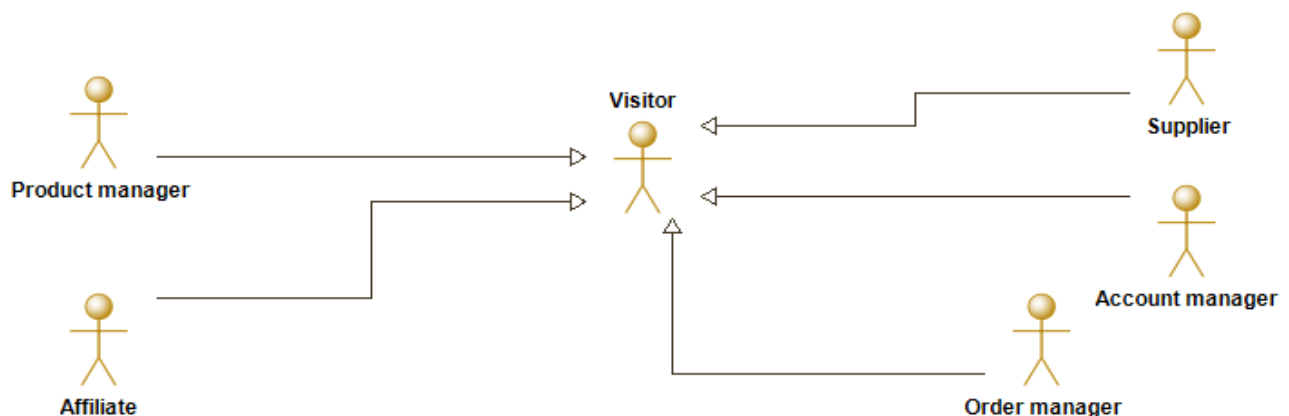


Figure 6 : Diagramme d'héritage entre les acteurs

### 3.1.2.2. Cas générale d'un visiteur

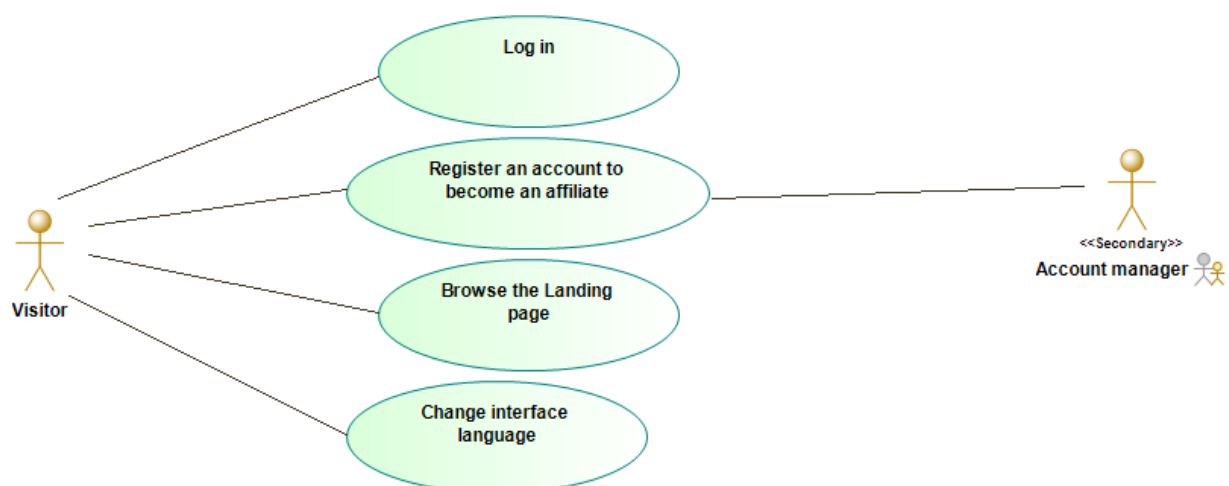


Figure 7 : Diagramme de cas d'utilisation d'un visiteur

### 3.1.2.3. Cas d'affilié

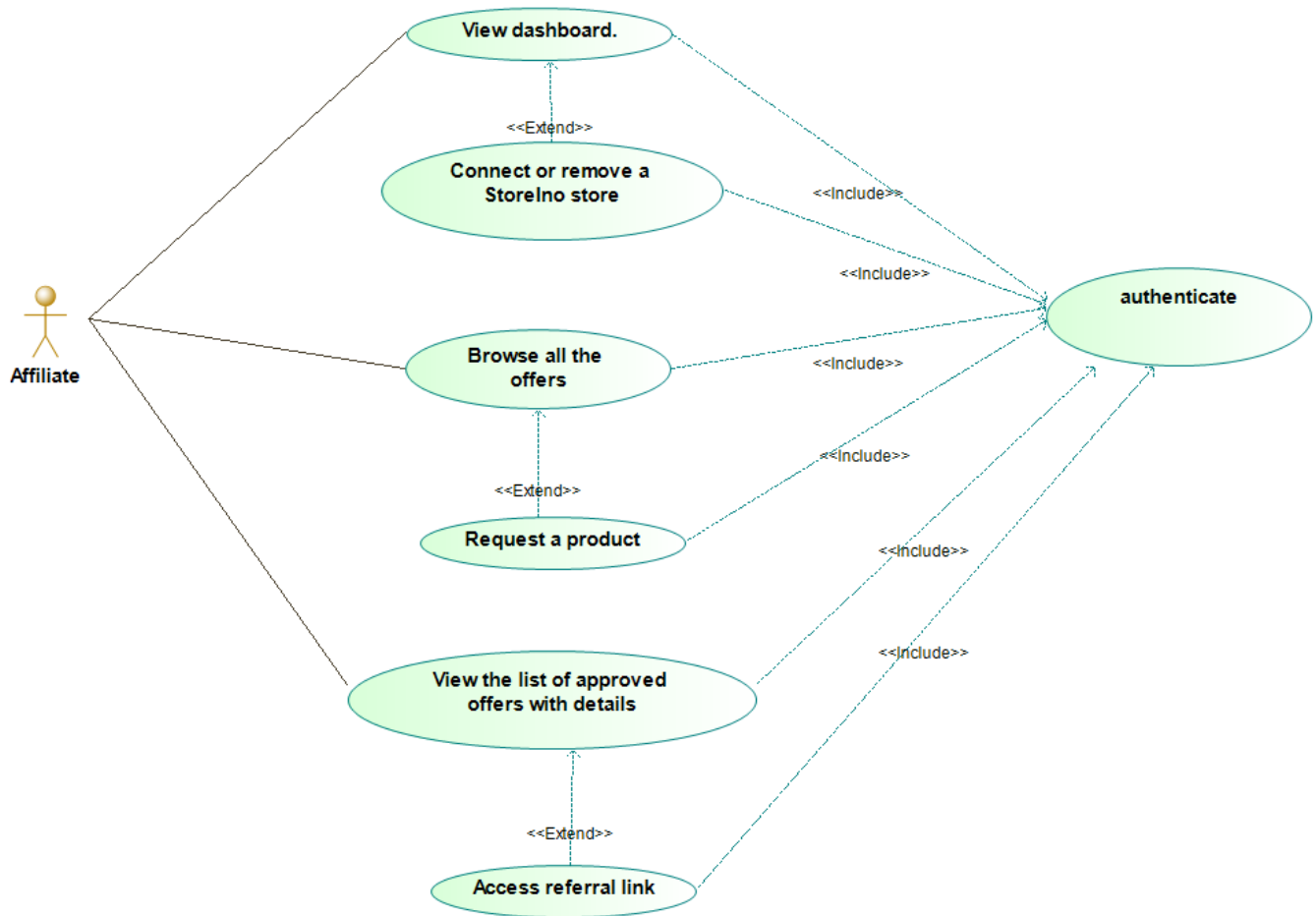


Figure 8 : 1ère partie du diagramme de cas d'utilisation d'un affilié(affiliate)

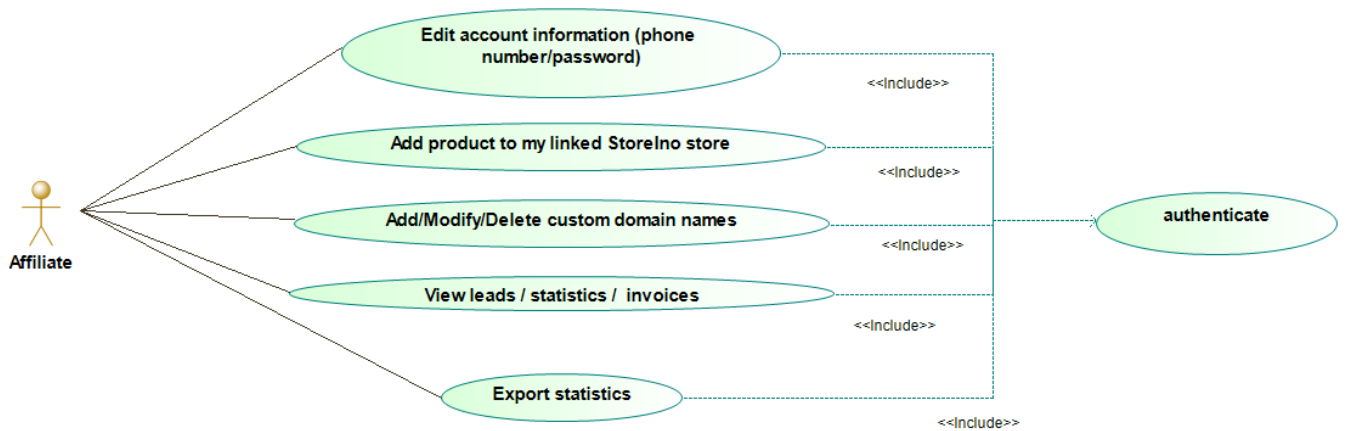


Figure 9 : 2éme partie du diagramme de cas d'utilisation d'un affilié(affiliate)



### 3.1.2.4. Cas de gestionnaire de produits

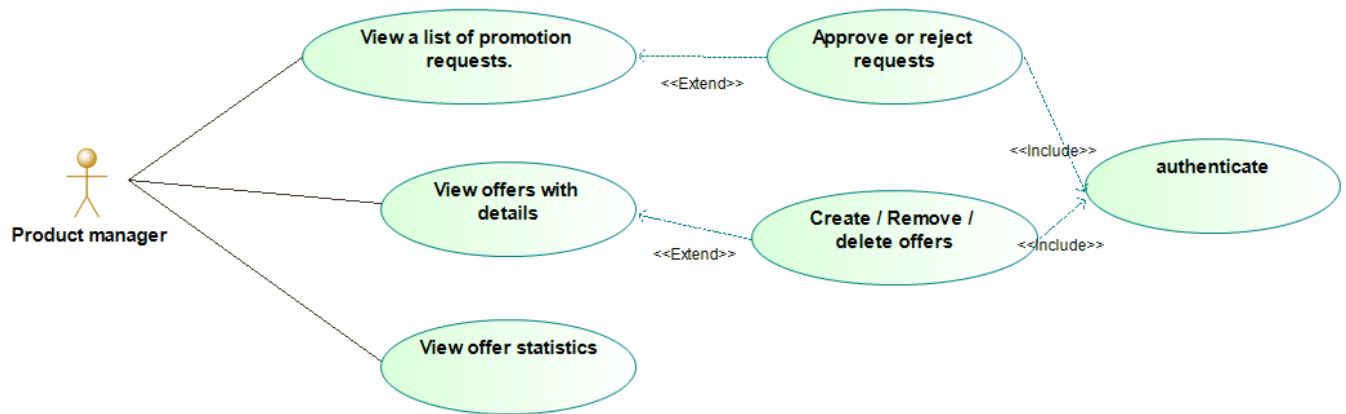


Figure 10 : Diagramme de cas d'utilisation d'un responsable de produits(product manager)

### 3.1.2.5. Cas de gestionnaire de comptes

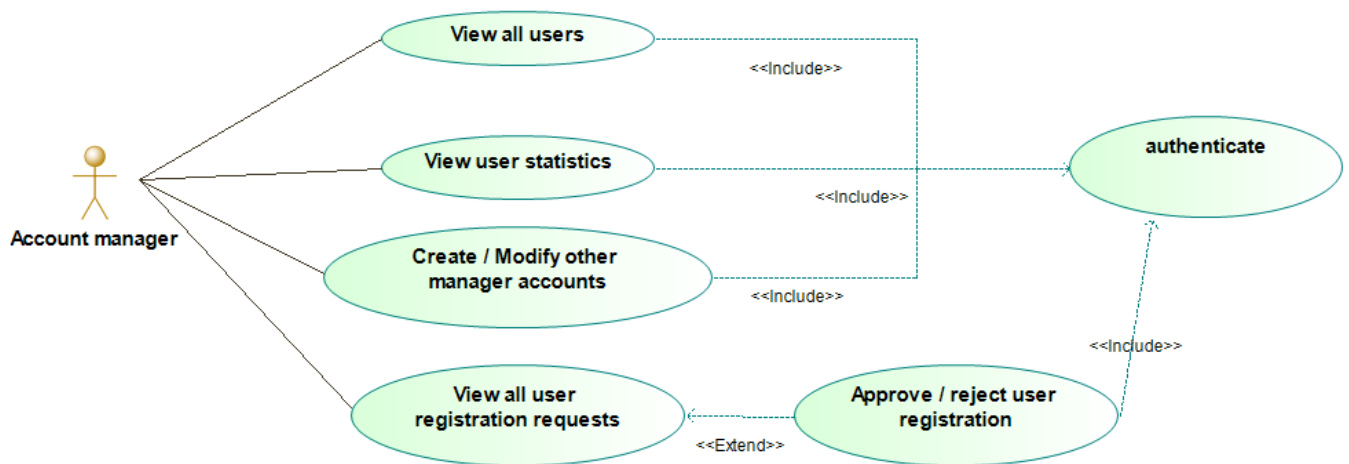


Figure 11 : Diagramme de cas d'utilisation d'un responsable de comptes(account manager)

### 3.1.2.6. Cas de gestionnaire de commandes

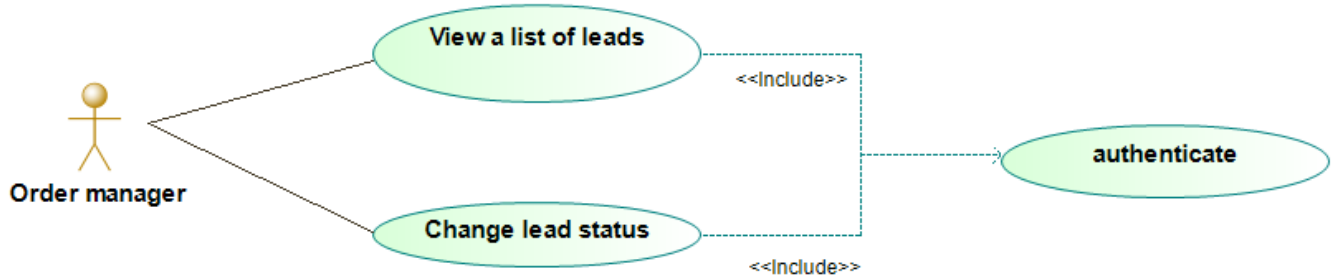


Figure 12 : Diagramme de cas d'utilisation d'un responsable de commandes(order manager)

### 3.1.2.7. Cas de fournisseurs

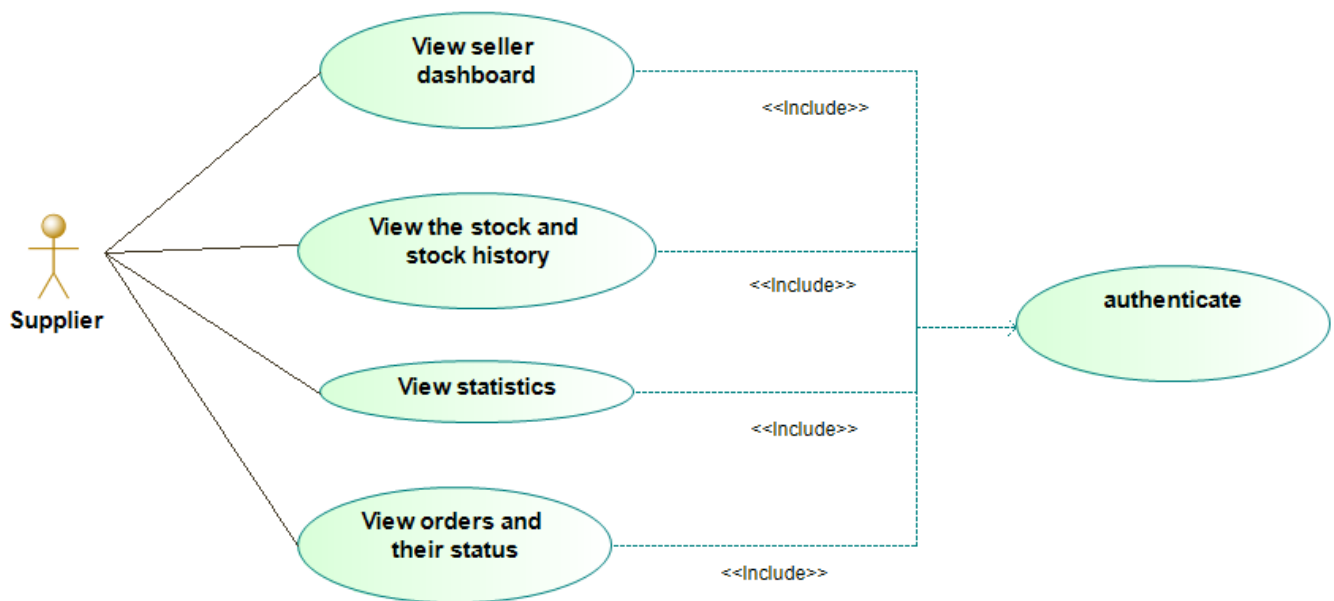


Figure 13 : Diagramme de cas d'utilisation d'un fournisseur(supplier)

## 3.2. Diagramme de classe

### 3.2.1. Définition

Le diagramme de classes est un outil essentiel en génie logiciel pour représenter visuellement les classes, les interfaces et les relations entre elles dans un système. Il offre une vue d'ensemble structurée des composants d'un programme, permettant aux développeurs de mieux comprendre l'organisation et les interactions entre les différentes parties du système.

Une classe est un élément fondamental de la programmation orientée objet. Elle regroupe des fonctions et des attributs liés qui décrivent le comportement et les caractéristiques d'un objet. Les fonctions définies dans une classe, appelées méthodes, encapsulent les actions que l'objet peut effectuer, tandis que les attributs représentent les données associées à l'objet. En utilisant des classes, les développeurs peuvent modéliser un programme de manière plus intuitive, en divisant une tâche complexe en plusieurs modules plus simples et réutilisables.

### 3.2.2. Diagrammes

#### 3.2.2.1. Diagramme de classe partie d'utilisateur

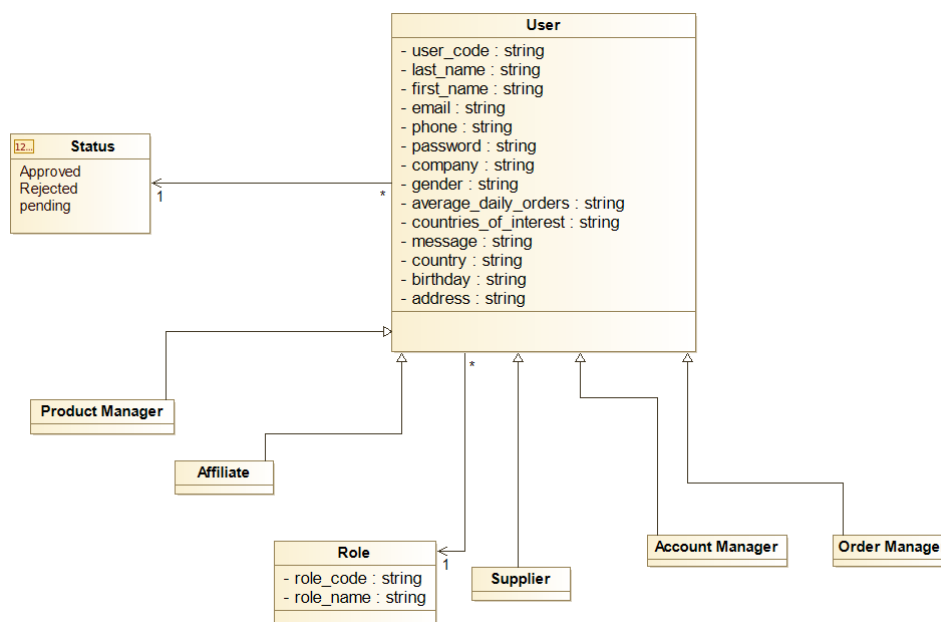


Figure 14 : Diagramme de classe, partie utilisateur

### 3.2.2.2. Diagramme de classe partie affilié

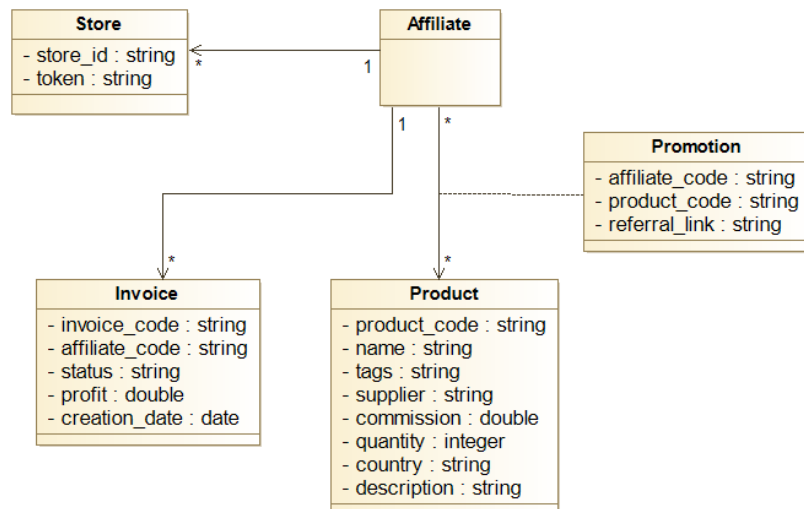


Figure 15 : Diagramme de classe, partie affilié

### 3.2.2.3. Diagramme de classe partie visiteur

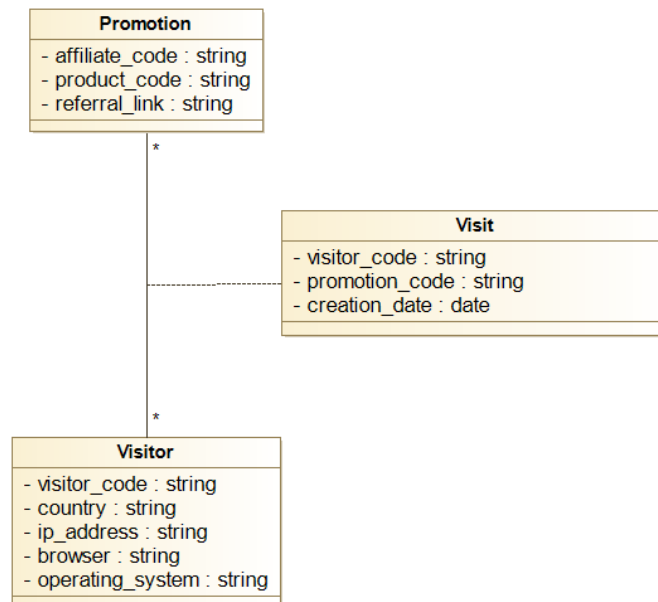


Figure 16 : Diagramme de classe, partie visiteur

### 3.2.2.4. Diagramme de classe partie customer

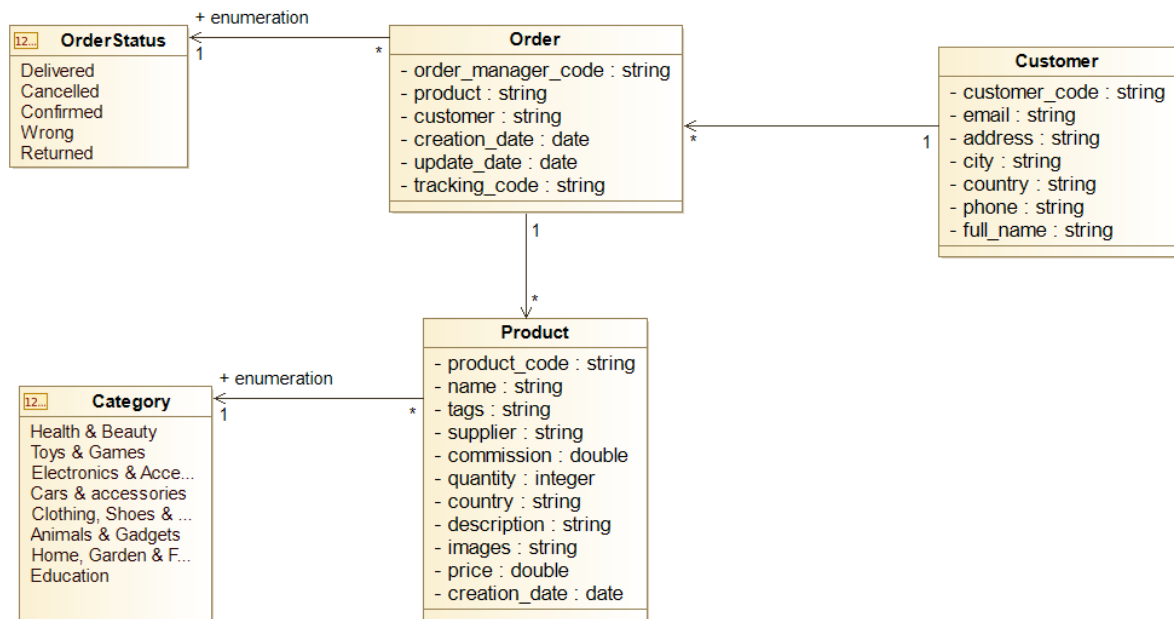


Figure 17 : Diagramme de classe, partie customer

### 3.2.2.5. Diagramme de classe Affiliate billing

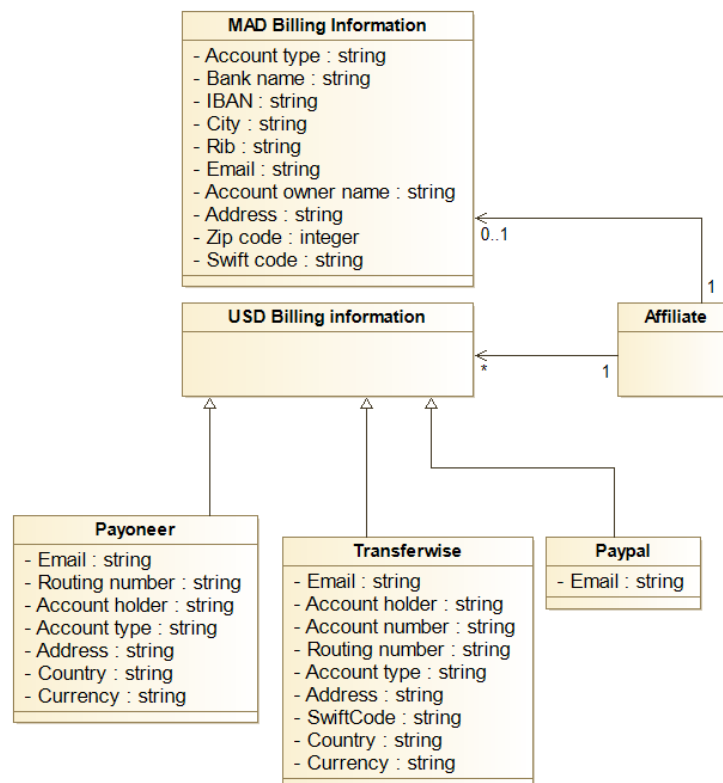


Figure 18 : Diagramme de classe des informations de paiement d'un affilié(affiliate)

### 3.2.2.6. Diagramme de classe partie produit

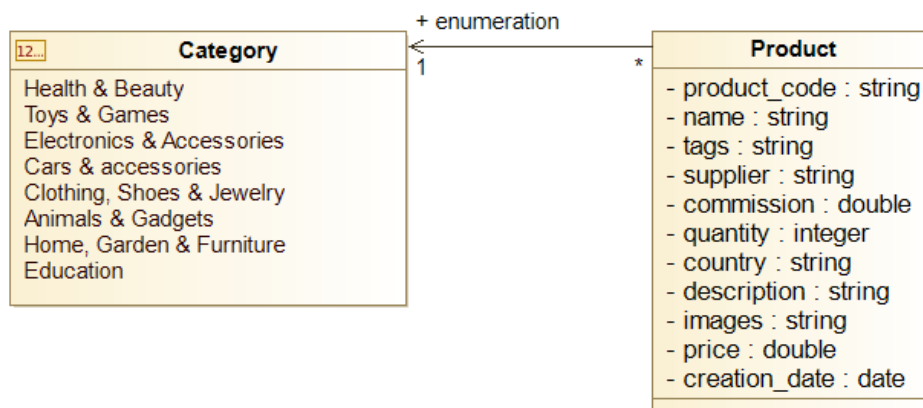


Figure 19 : Diagramme de classe, partie produit

### 3.2.2.7. Diagramme de classe partie invoice

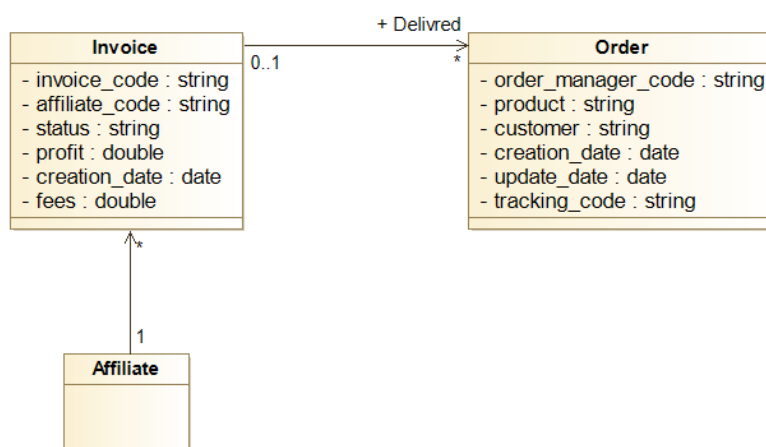


Figure 20 : Diagramme de classe, partie facturation

### 3.2.2.8. Vue d'ensemble

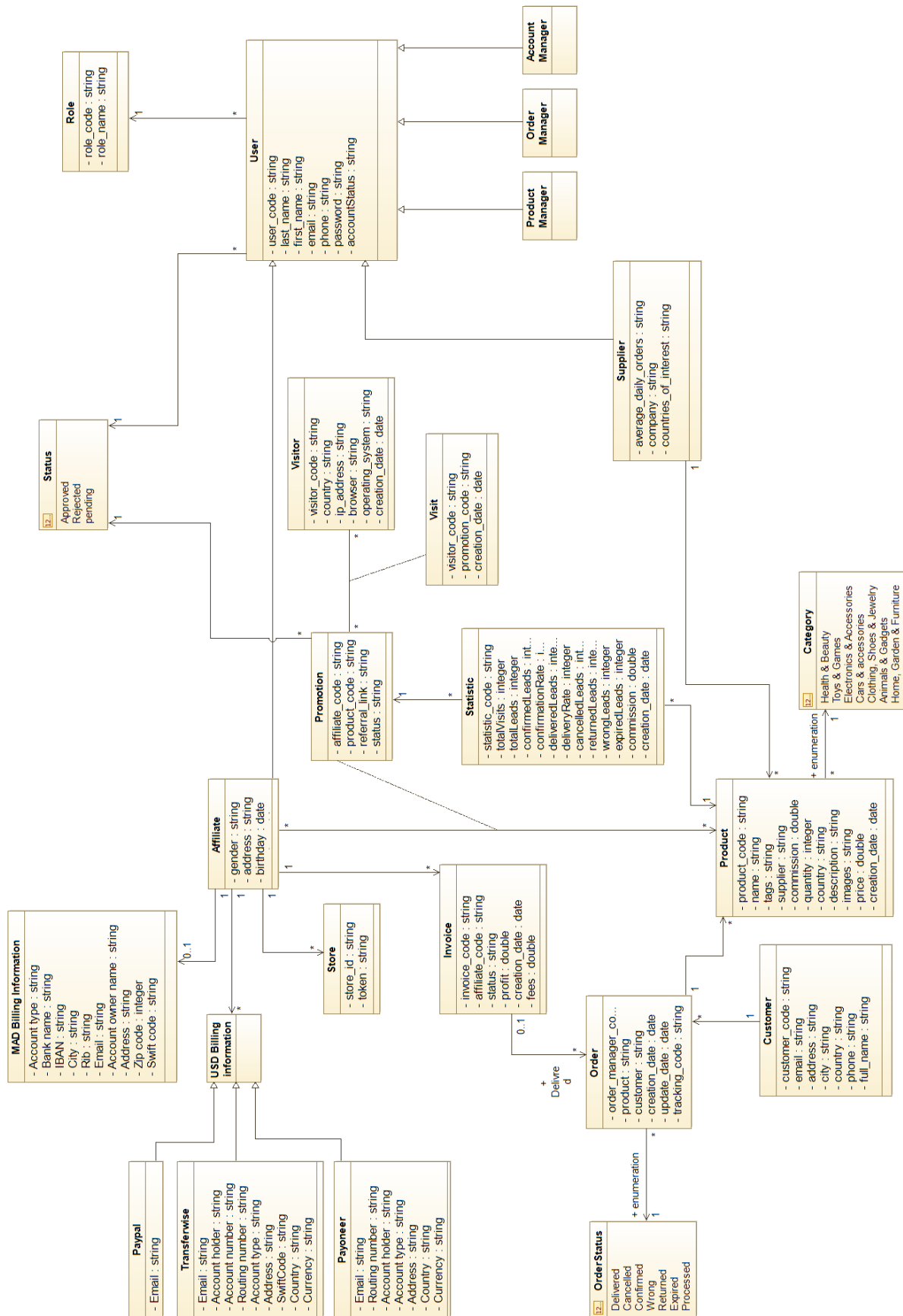


Figure 21 : Diagramme de classe dans son ensemble

### 3.3. Diagramme de séquences

#### 3.3.1. Définition

Un diagramme de séquence est un diagramme d'interaction qui expose en détail la façon dont les opérations sont effectuées : quels messages sont envoyés et quand ils le sont. Les diagrammes de séquence sont organisés en fonction du temps. Le temps s'écoule au fur et à mesure que vous parcourez la page. Les objets impliqués dans l'opération sont répertoriés de gauche à droite en fonction du moment où ils prennent part dans la séquence de messages.

#### 3.3.2. Diagrammes

##### 3.3.2.1. Inscription des visiteurs

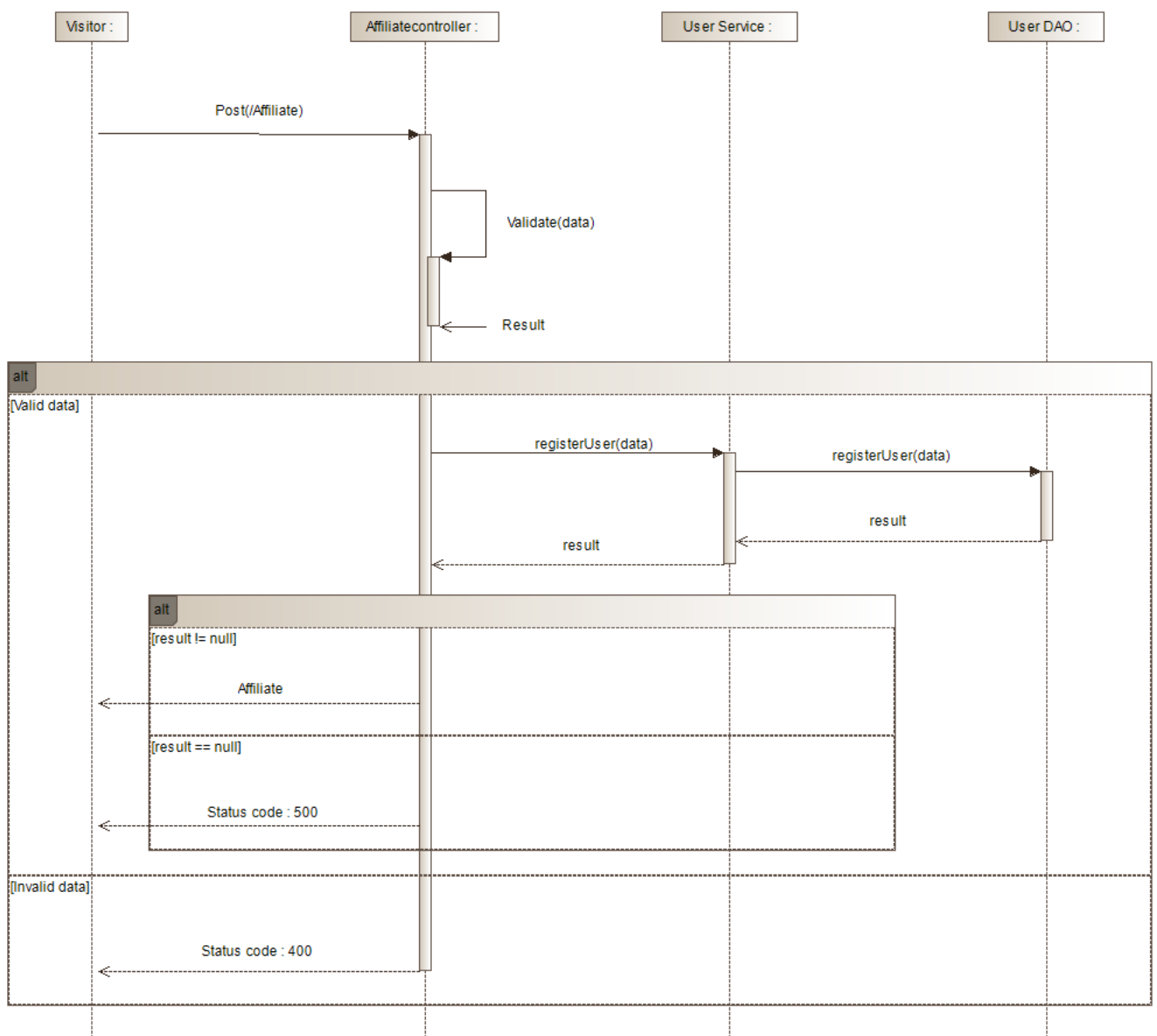


Figure 22 : Diagramme de séquence d'inscription d'un visiteur



### 3.3.2.2. Authentification d'utilisateur

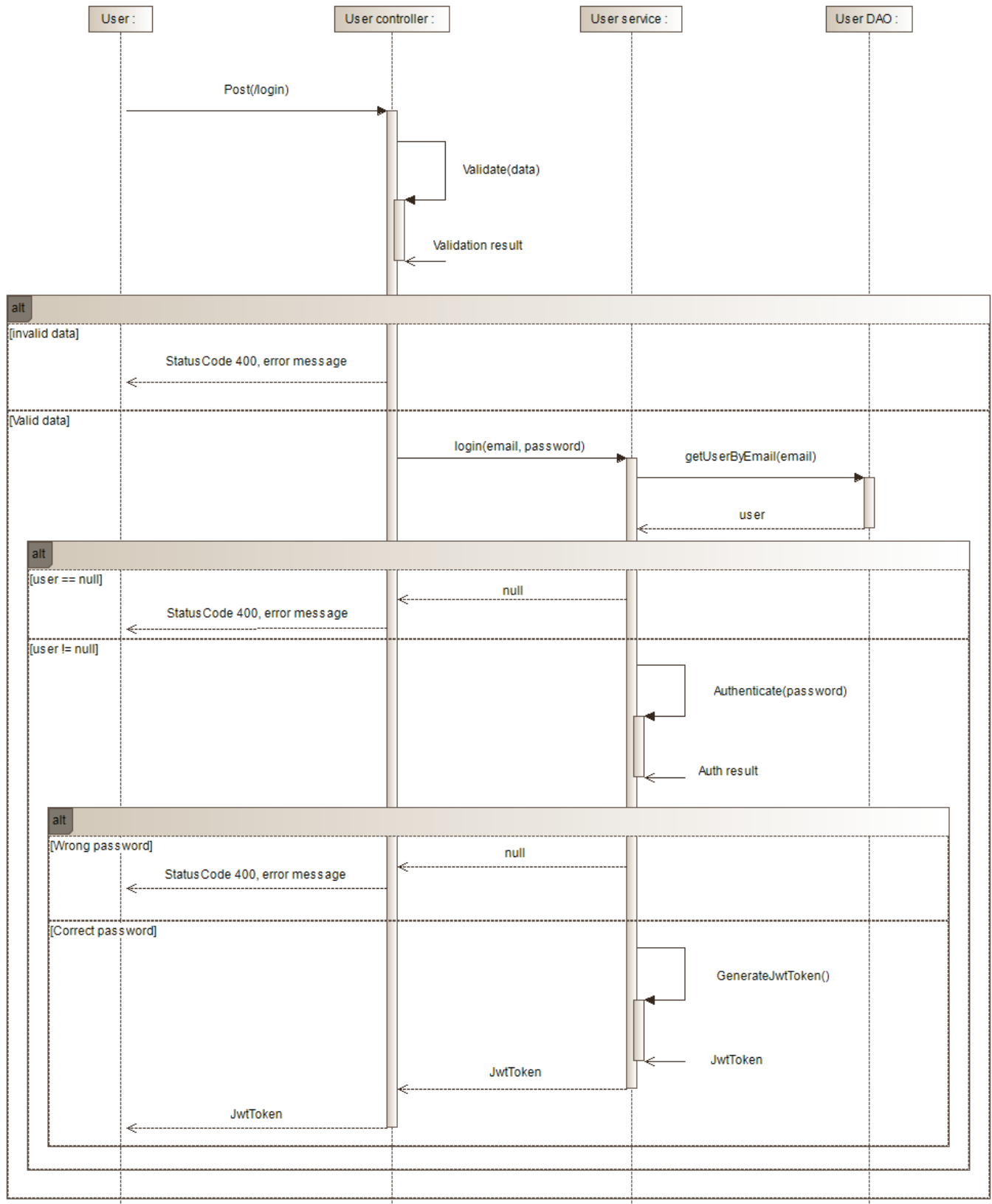


Figure 23 : Diagramme de séquence d'authentification

### 3.3.2.3. Vérification de rôle

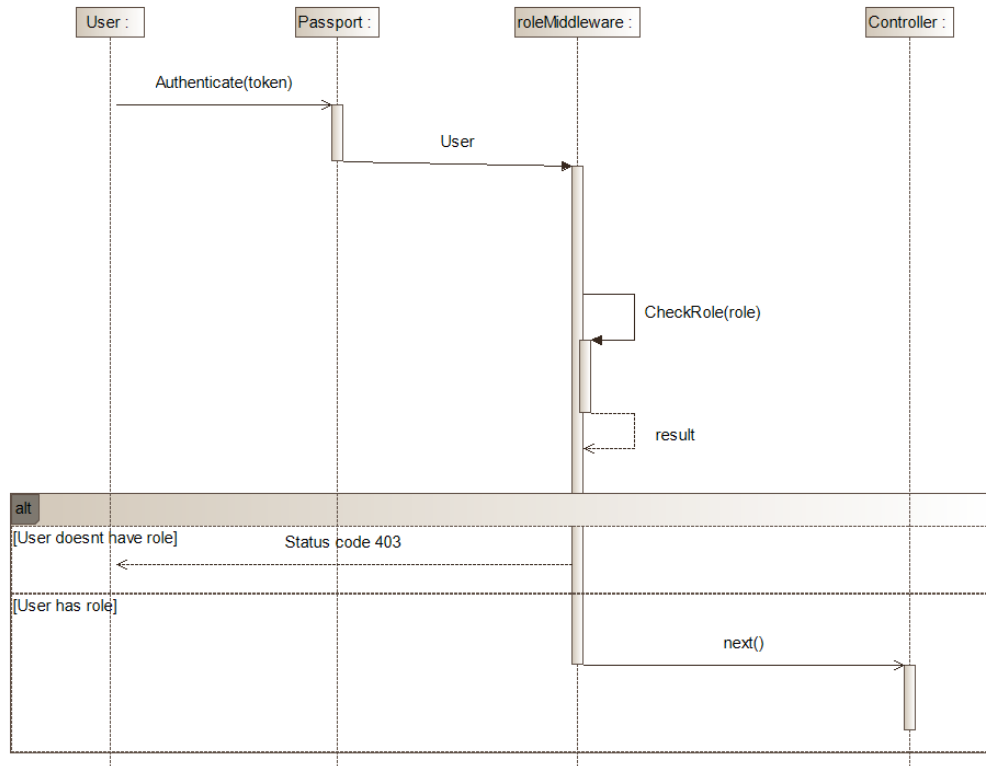


Figure 24 : Diagramme de séquence de vérification des rôles

### 3.3.2.4. Gestionnaire des comptes

#### 3.3.2.4.a. Afficher la liste des utilisateurs

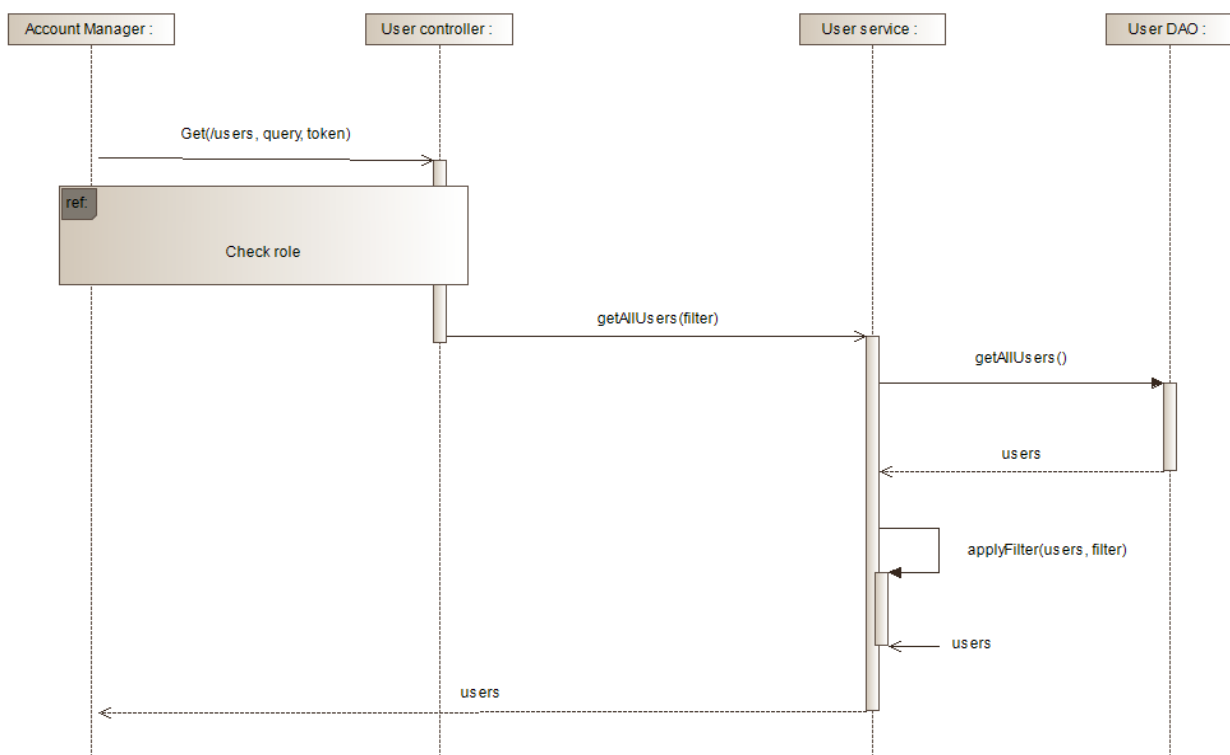


Figure 25 : Diagramme de séquence, récupération de la liste des utilisateurs

### 3.3.2.4.b. Récupérer un utilisateur par son identificateur

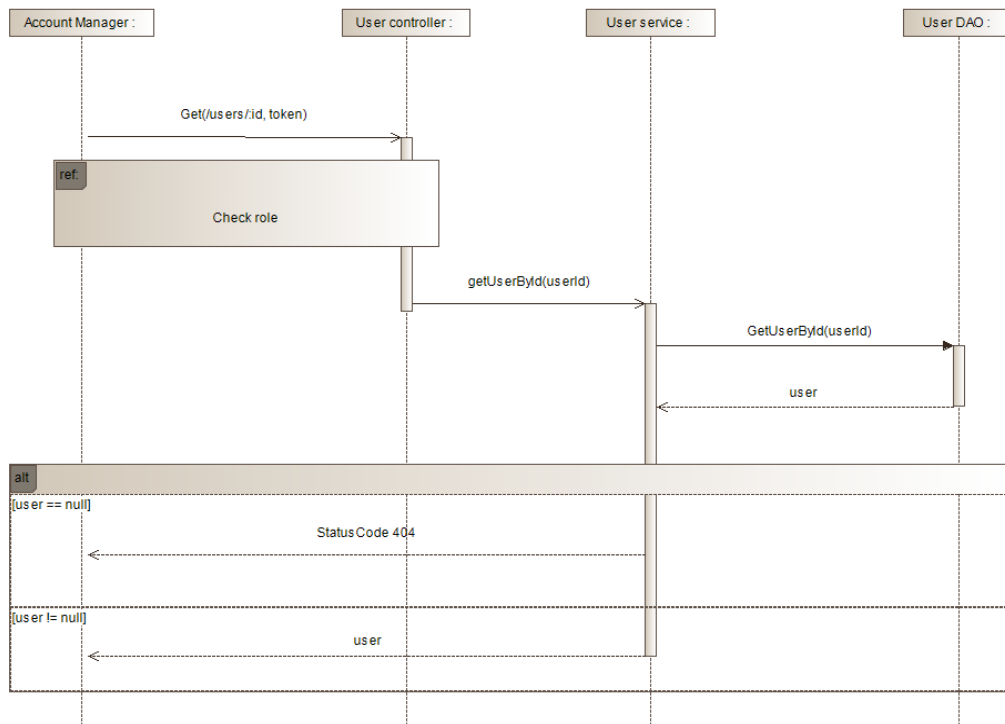


Figure 26 : Diagramme de séquence, récupération d'un utilisateur par son code

### 3.3.2.4.c. Mettre à jour les coordonnées d'un utilisateur

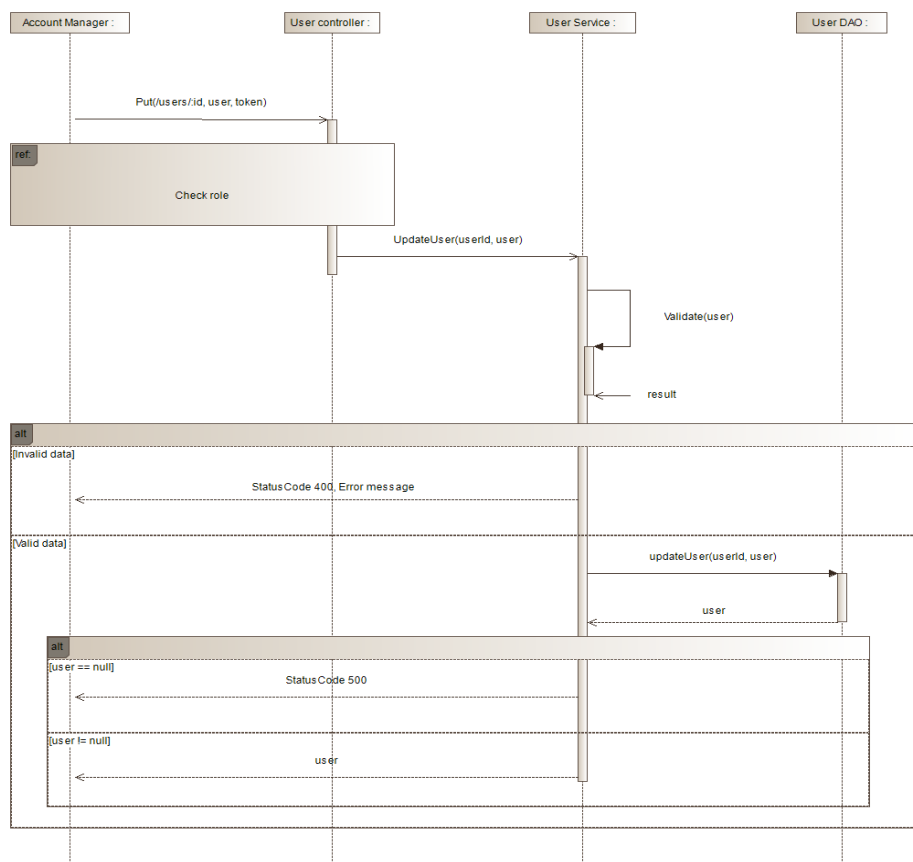


Figure 27 : Diagramme de séquence, mise à jour d'un utilisateur

### 3.3.2.4.d. Supprimer un utilisateur

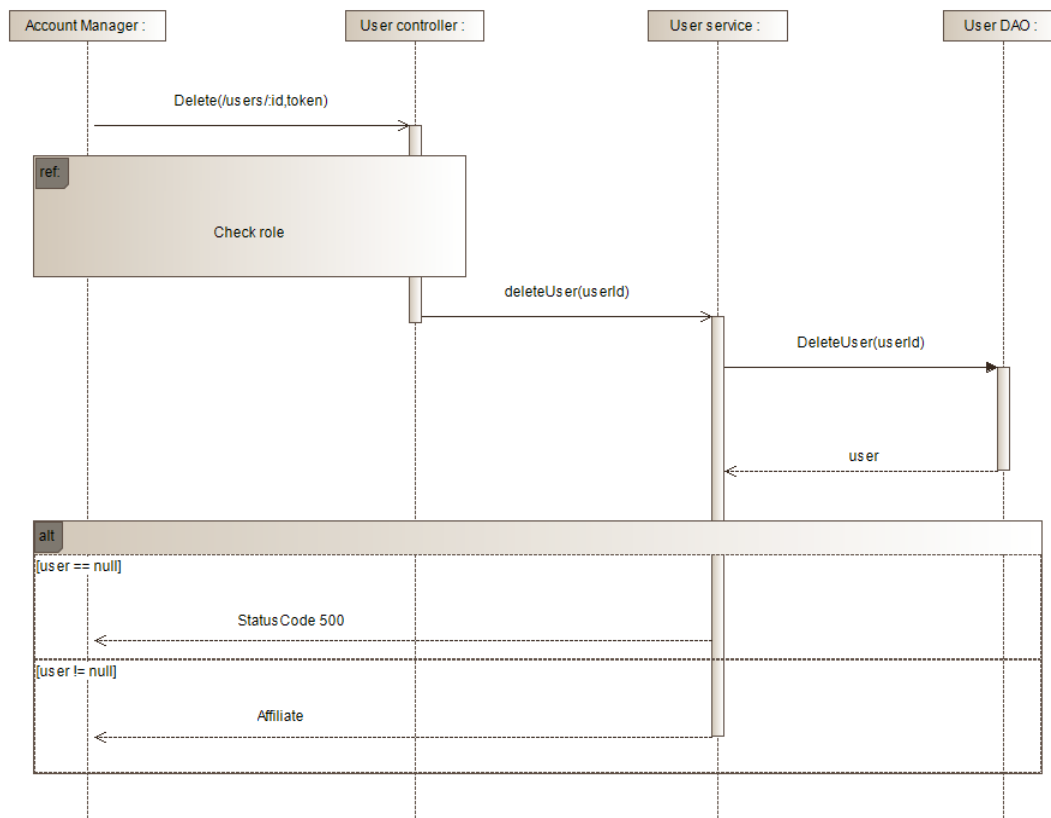


Figure 28 : Diagramme de séquence, suppression d'un utilisateur

### 3.3.2.4.e. Récupérer les informations de facturation en MAD d'un affilié

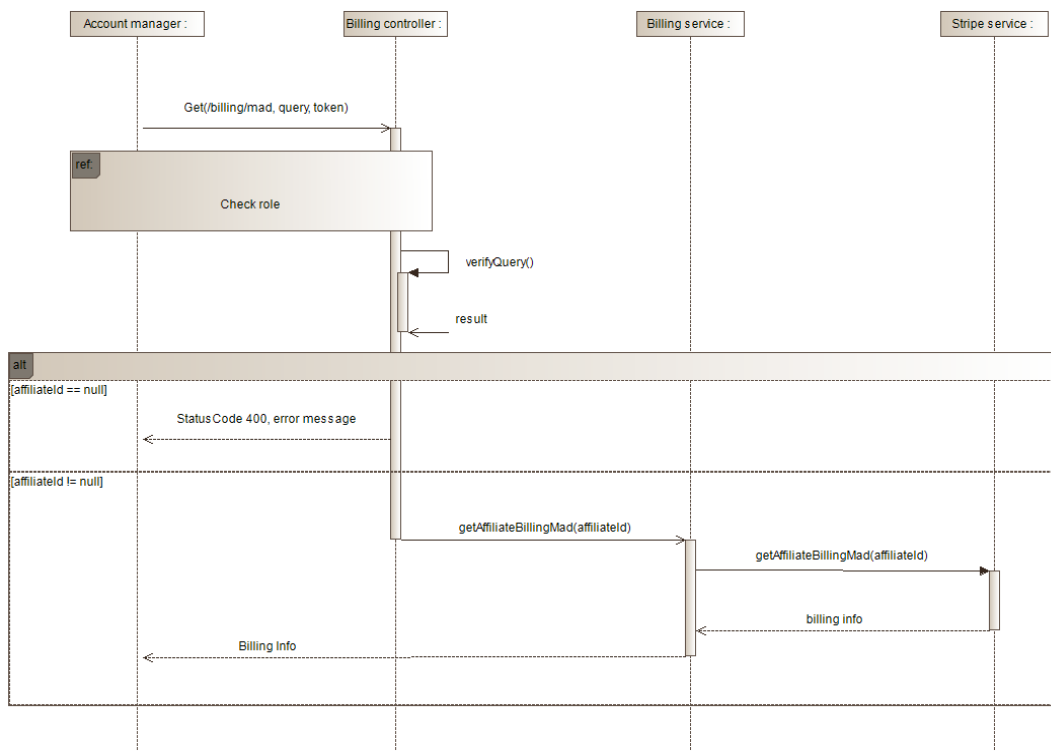


Figure 29 : Diagramme de séquence, récupération des informations de paiement MAD d'un affilié

### 3.3.2.4.f. Récupérer les informations de facturation en USD d'un affilié

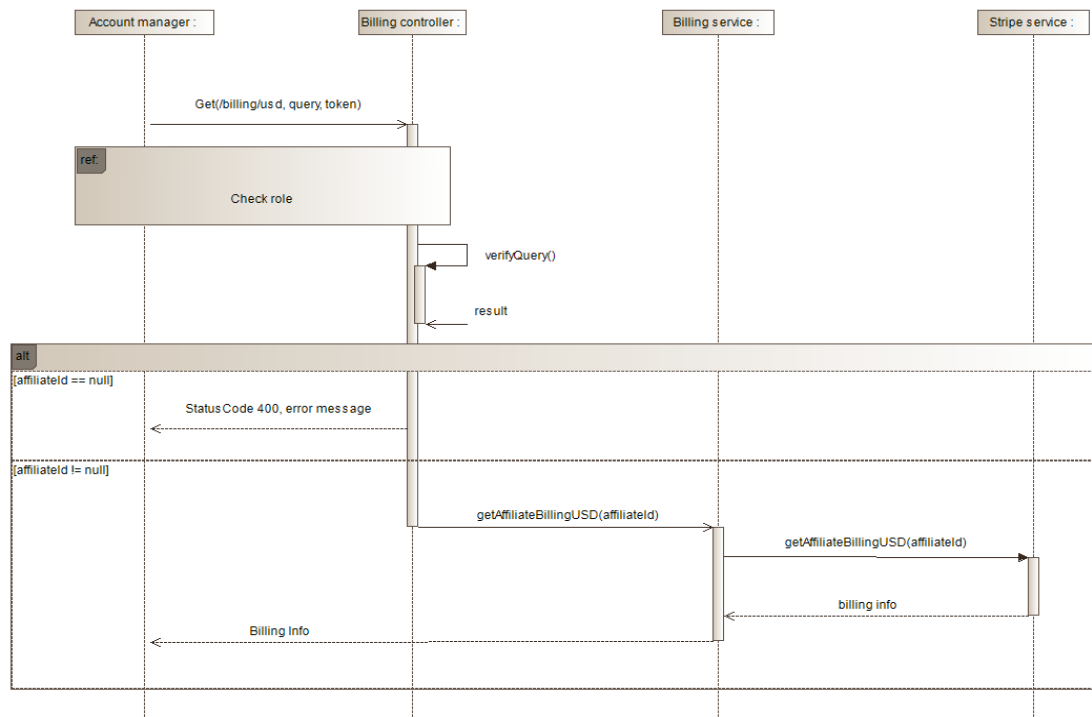


Figure 30 : Diagramme de séquence, récupération des informations de paiement USD d'un affilié

### 3.3.2.4.g. Supprimer les informations de facturation MAD d'un affilié

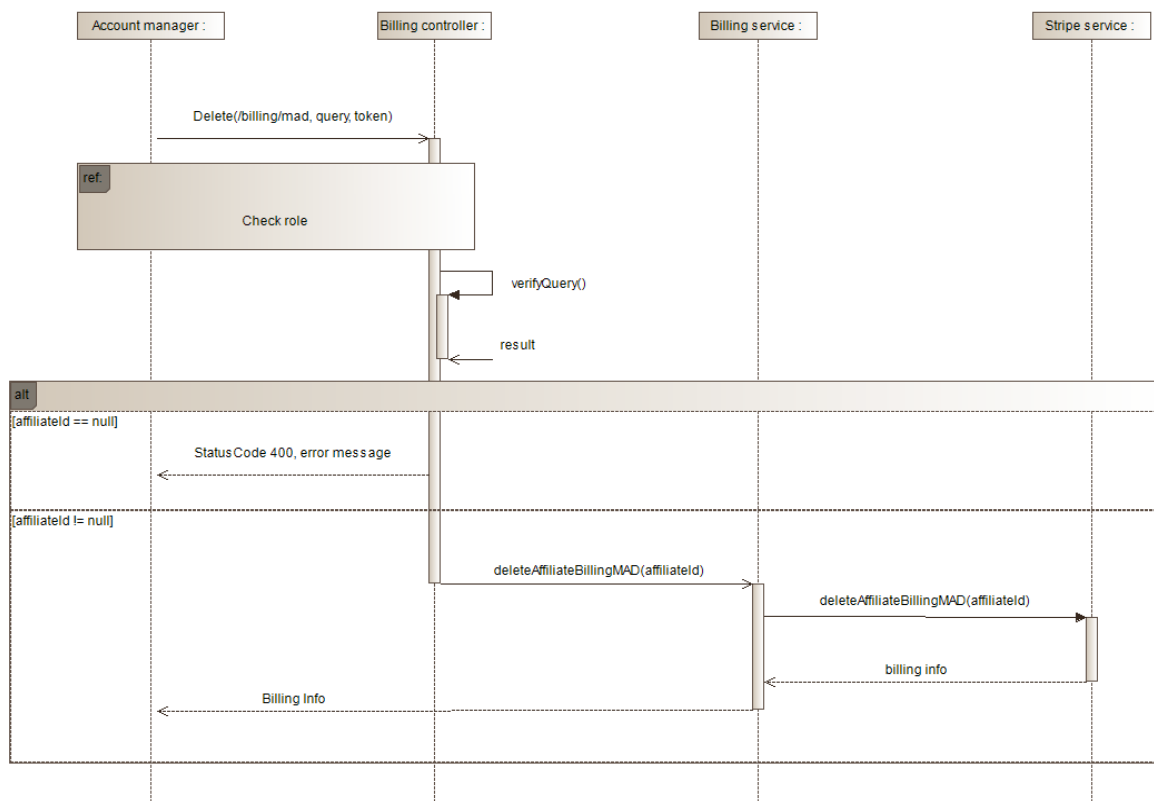


Figure 31 : Diagramme de séquence, suppression des informations de paiement MAD d'un affilié

### 3.3.2.4.h. Supprimer les informations de facturation USD d'un affilié

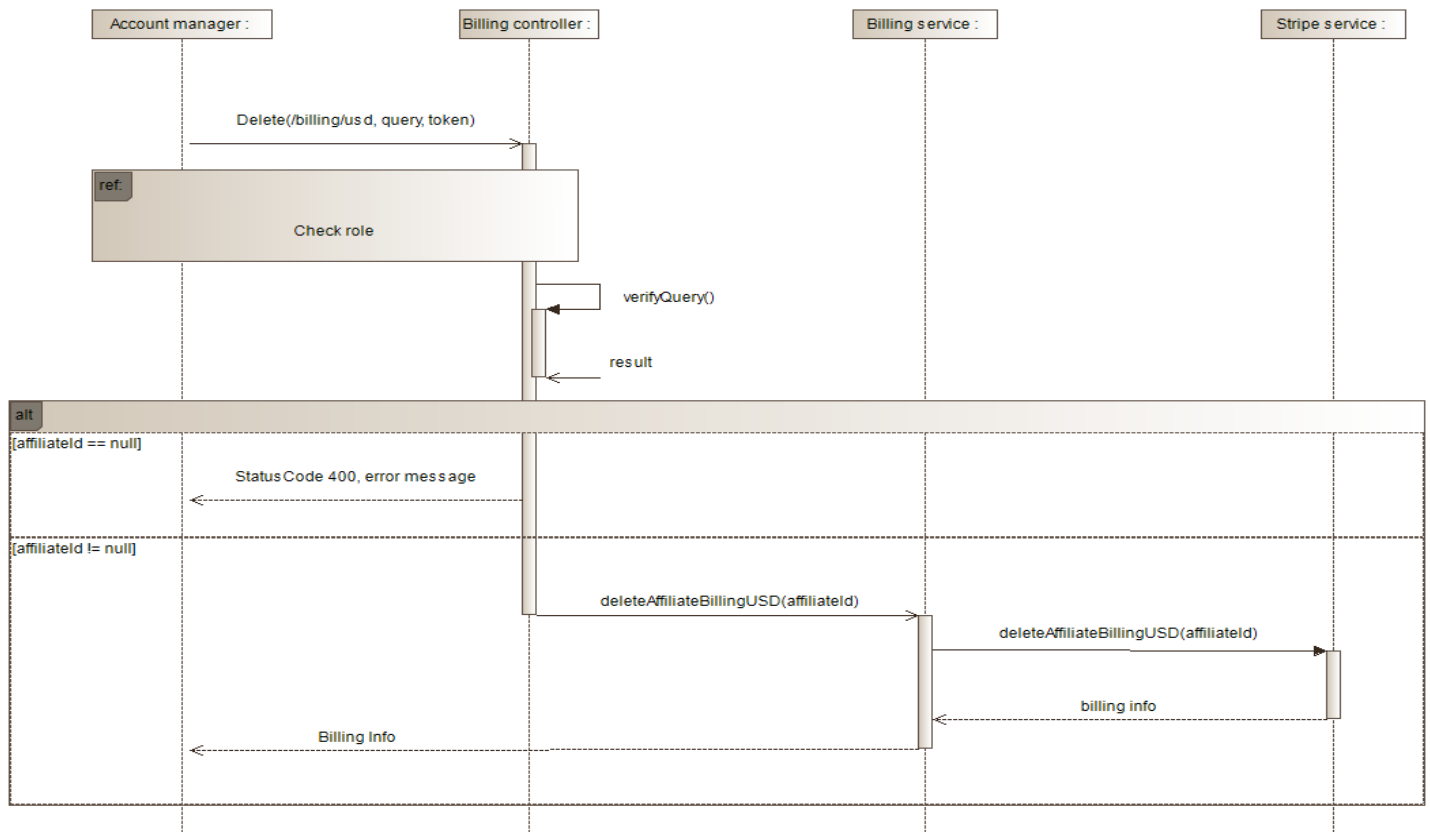


Figure 32 : Diagramme de séquence, suppression des informations de paiement MAD d'un affilié

### 3.3.2.4.i. Récupérer les promotions d'un affilié

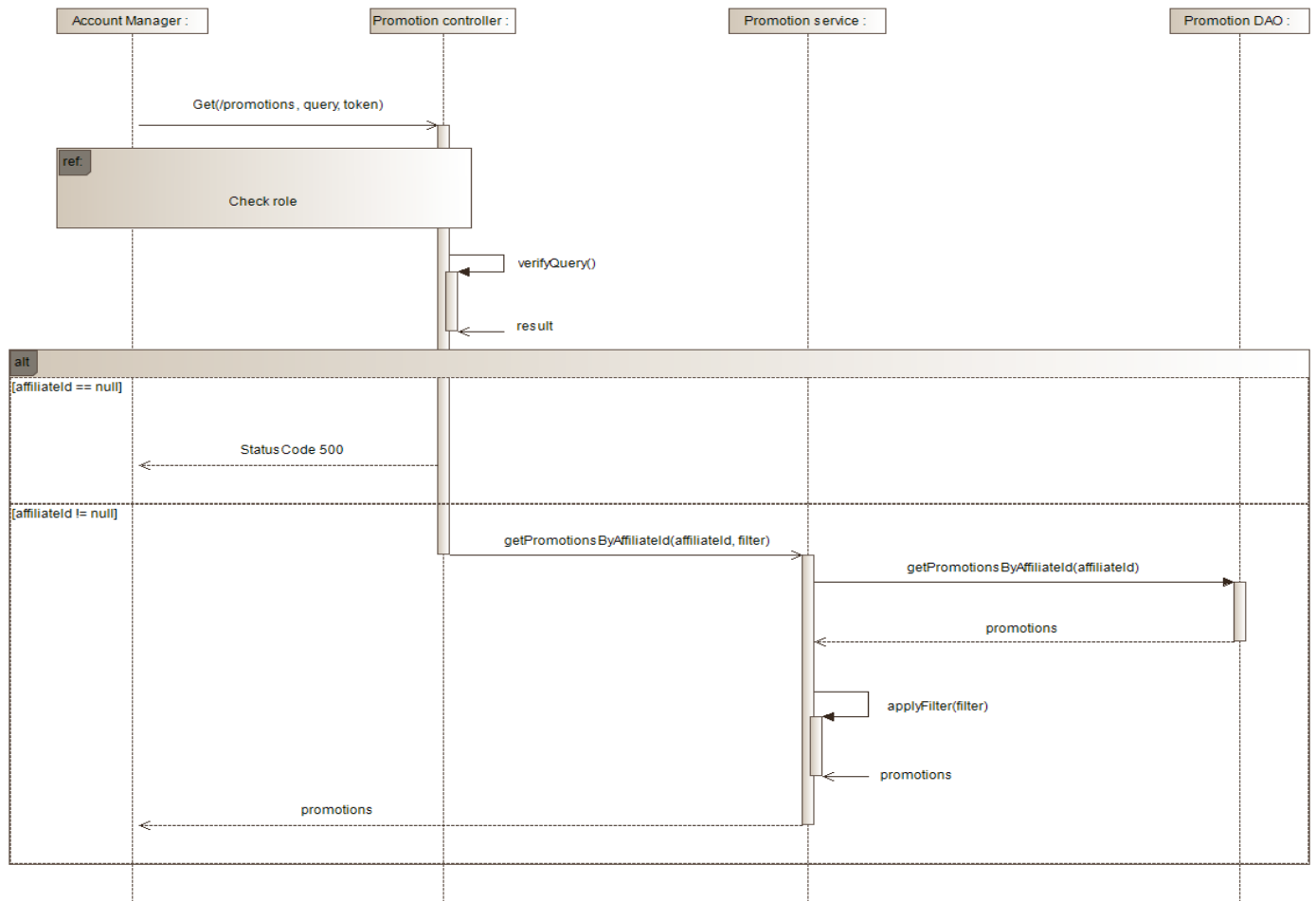


Figure 33 : Diagramme de séquence, récupération des promotions d'un affilié

### 3.3.2.5. Gestionnaire des produits

#### 3.3.2.5.a. Afficher la liste des produits

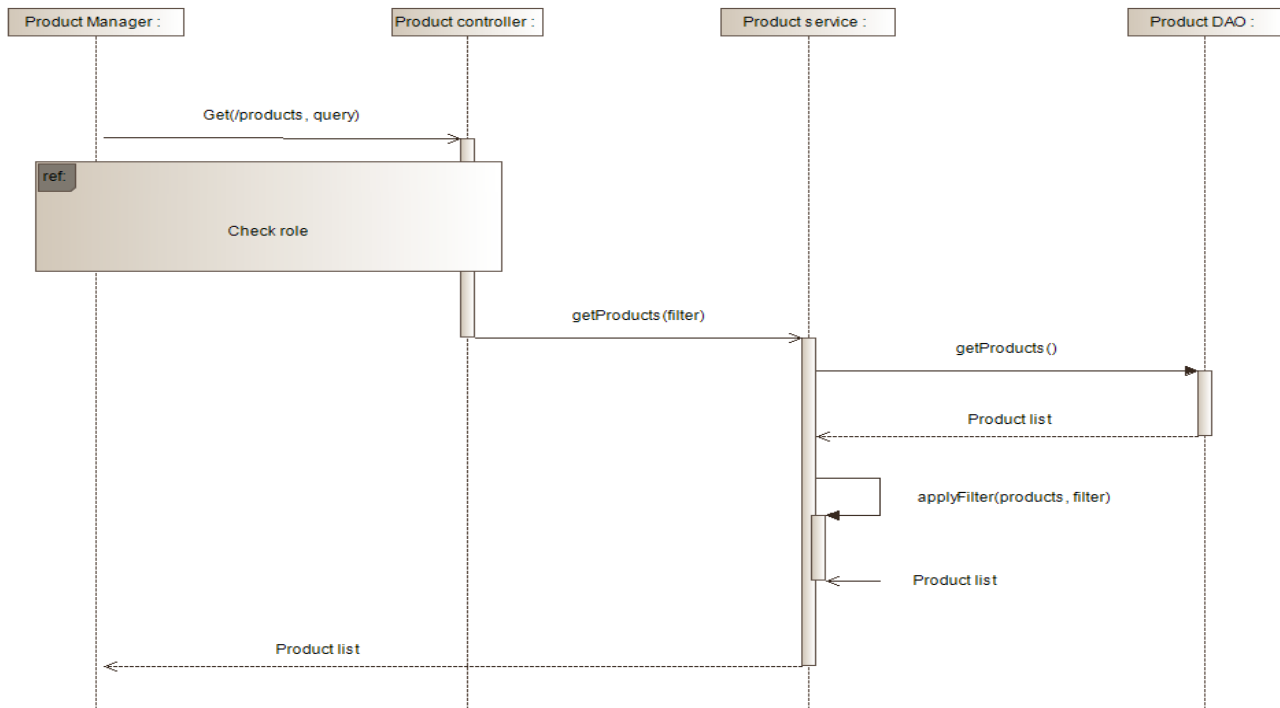


Figure 34 : Diagramme de séquence, récupération de la liste des produits

#### 3.3.2.5.b. Récupérer un produit avec son identificateur

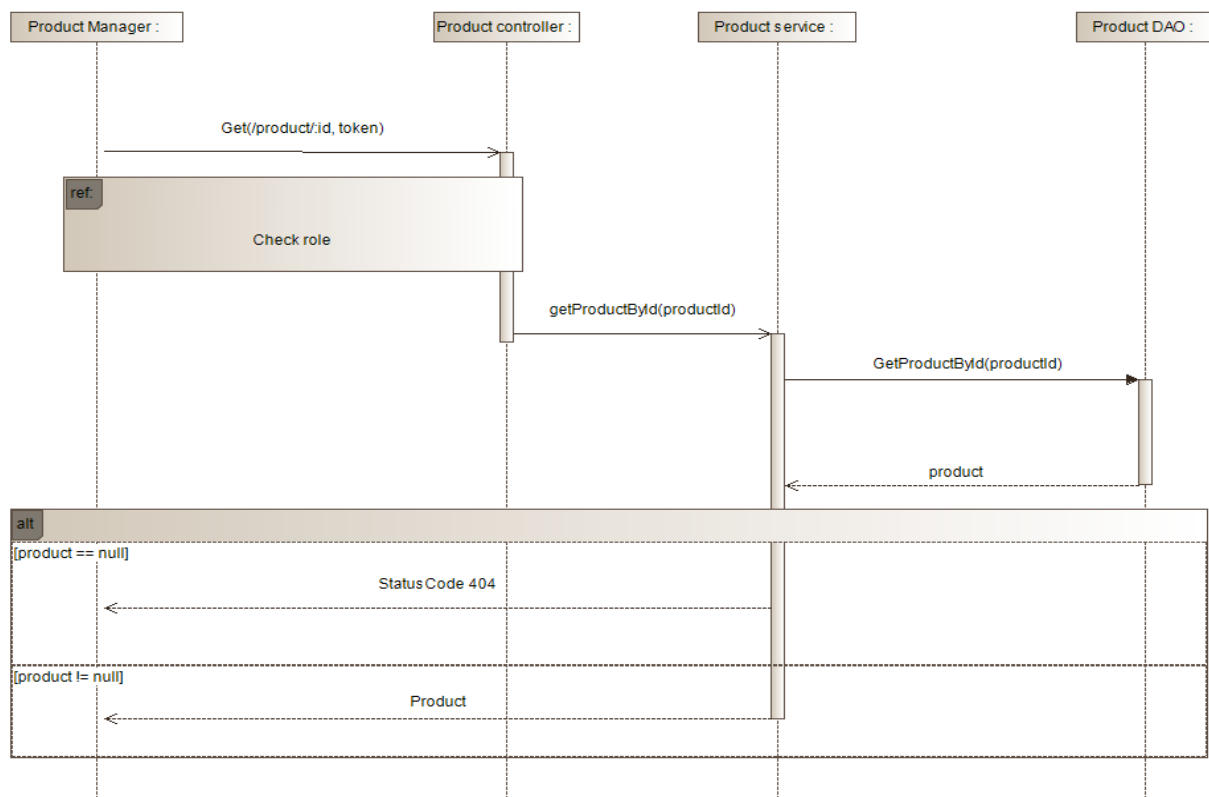


Figure 35 : Diagramme de séquence, récupération d'un produit



### 3.3.2.5.c. Ajouter un produit

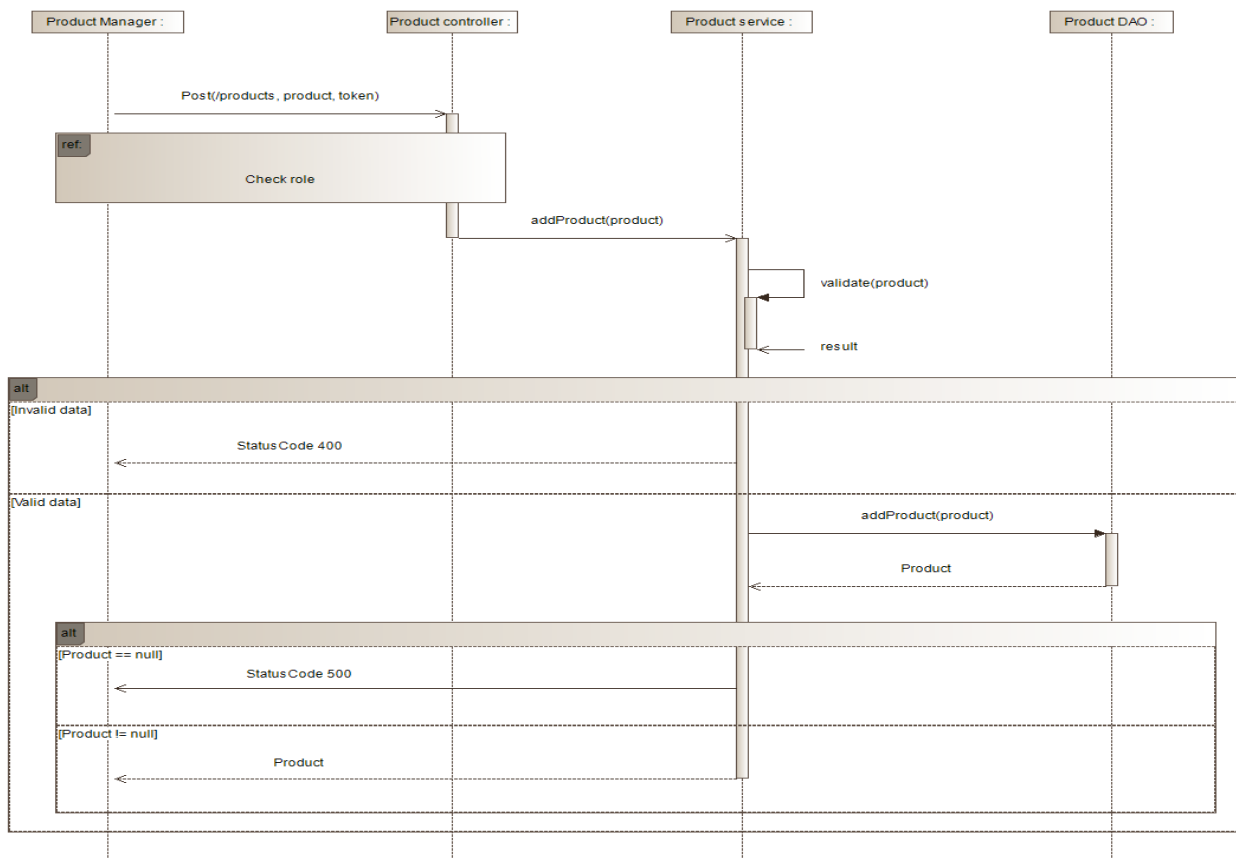


Figure 36 : Diagramme de séquence, création d'un produit

### 3.3.2.5.d. Récupérer les statistiques des produits

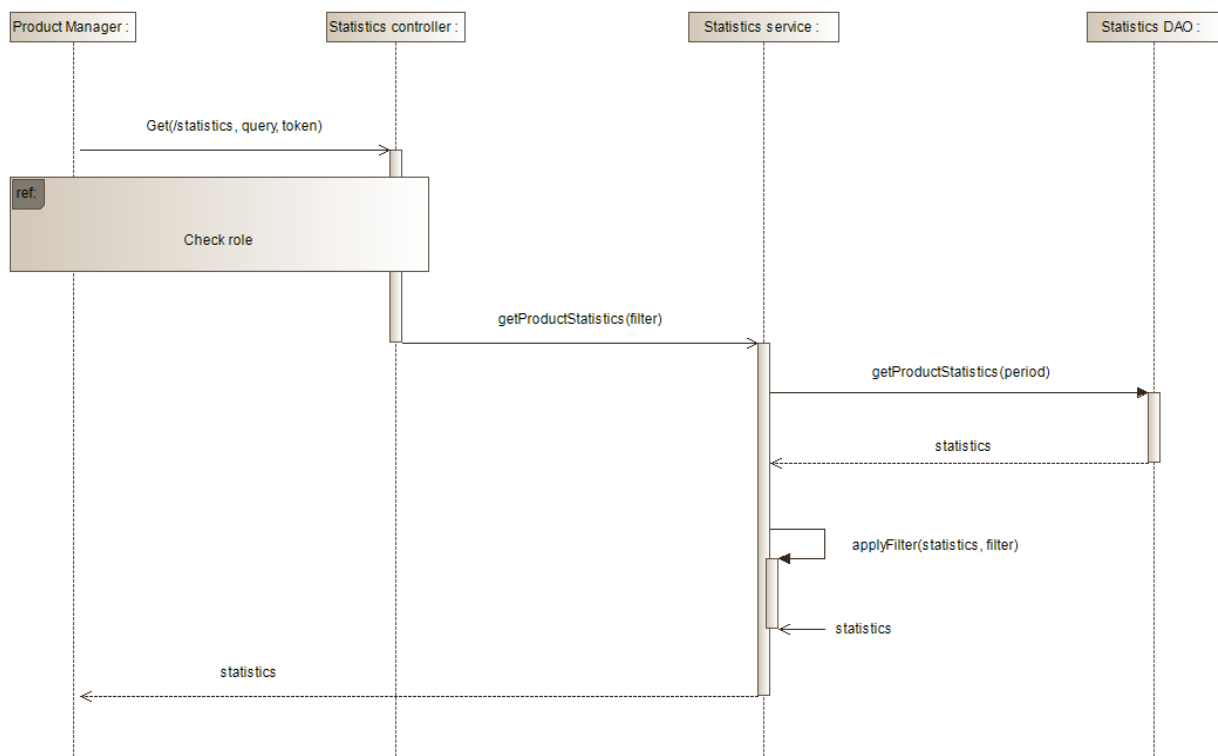


Figure 37 : Diagramme de séquence, récupération des statistiques des produits

### 3.3.2.5.e. Récupérer les demandes de promotions des produits

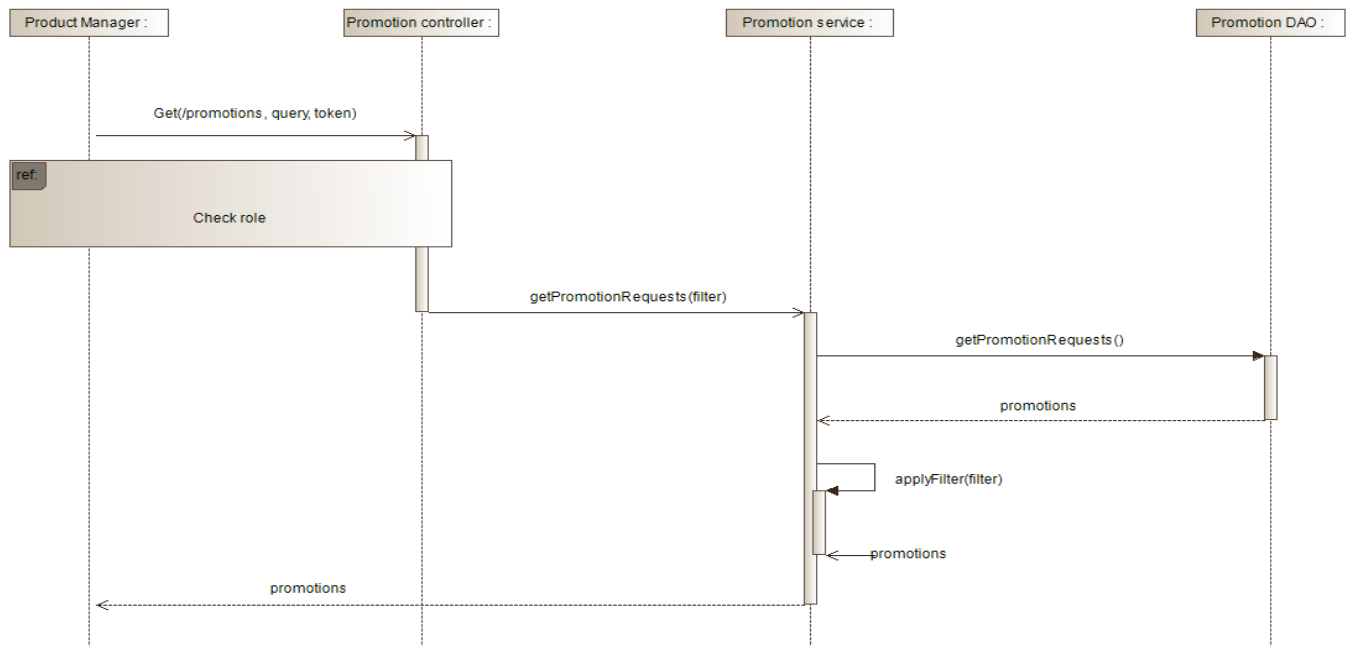


Figure 38 : Diagramme de séquence, récupération des demandes de promotions

### 3.3.2.5.f. Mettre à jour une demande de promotion

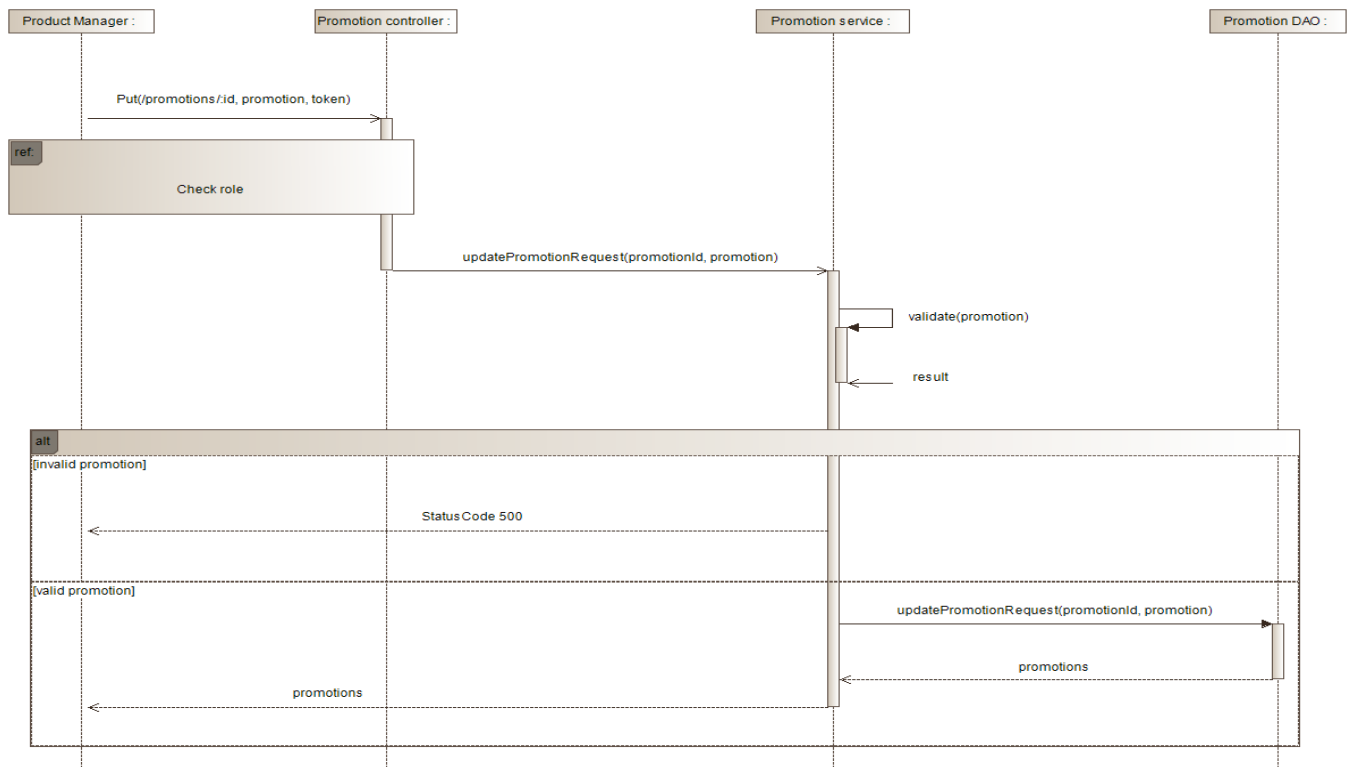


Figure 39 : Diagramme de séquence, mise à jour d'une demande de promotion

### 3.3.2.6. Affilié

#### 3.3.2.6.a. Récupérer mes informations

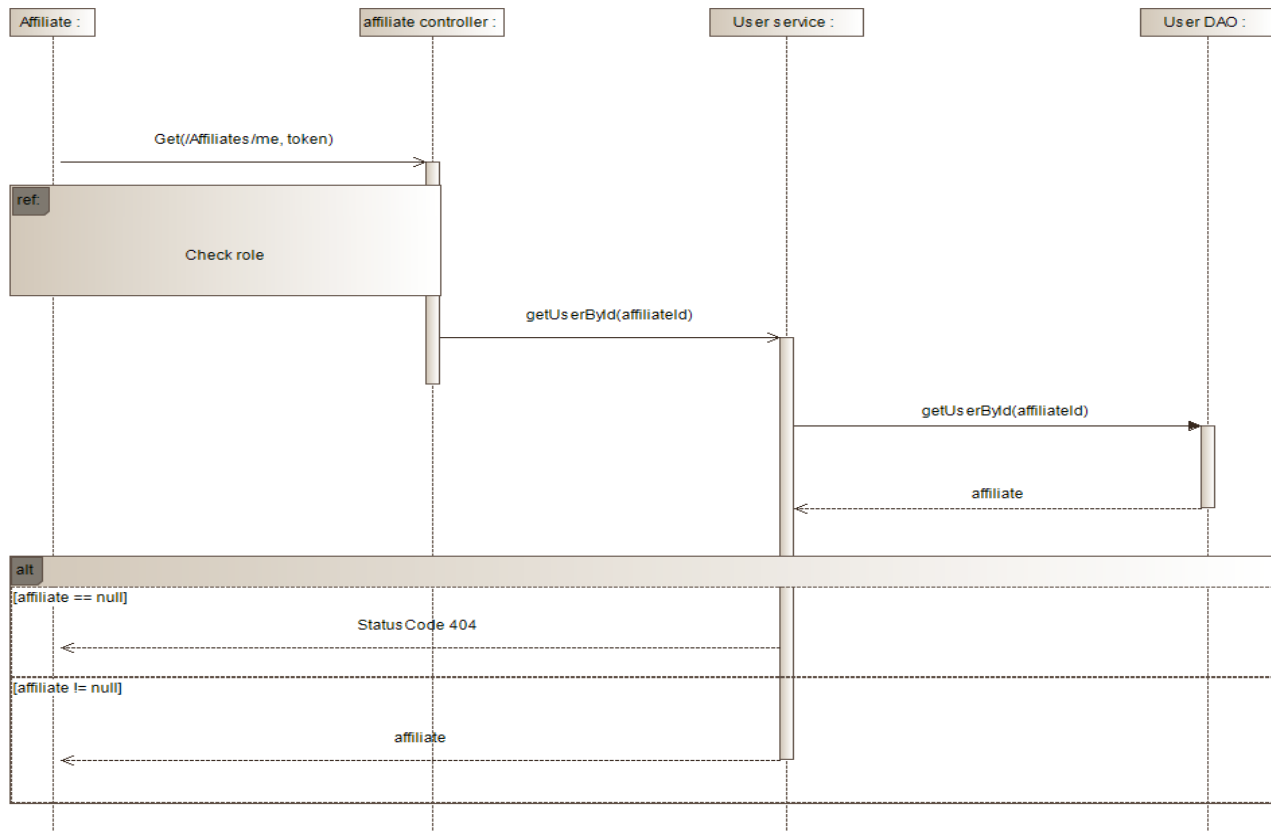


Figure 40 : Diagramme de séquence, récupération des données personnels d'un affilié

#### 3.3.2.6.b. Récupérer la liste des offres

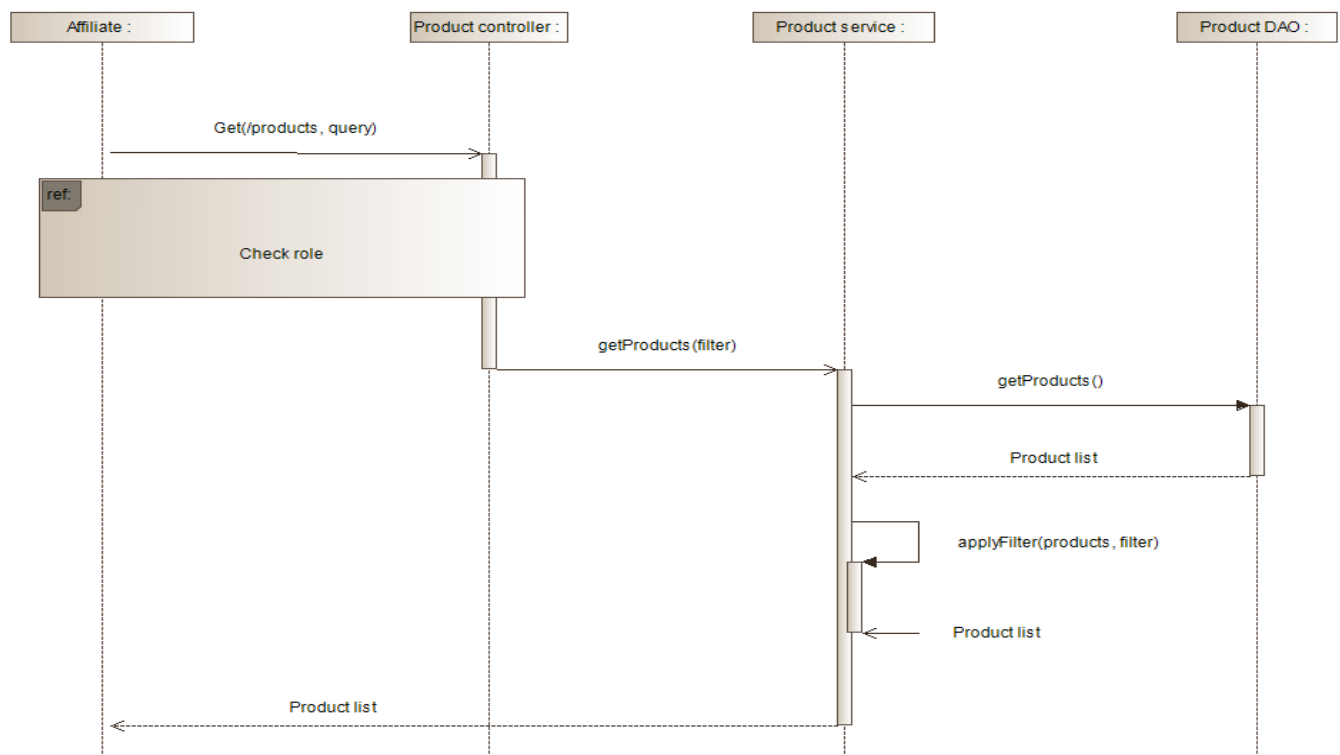


Figure 41 : Diagramme de séquence, récupération de liste des offres

### 3.3.2.6.c. Récupérer un produit par son identifiant

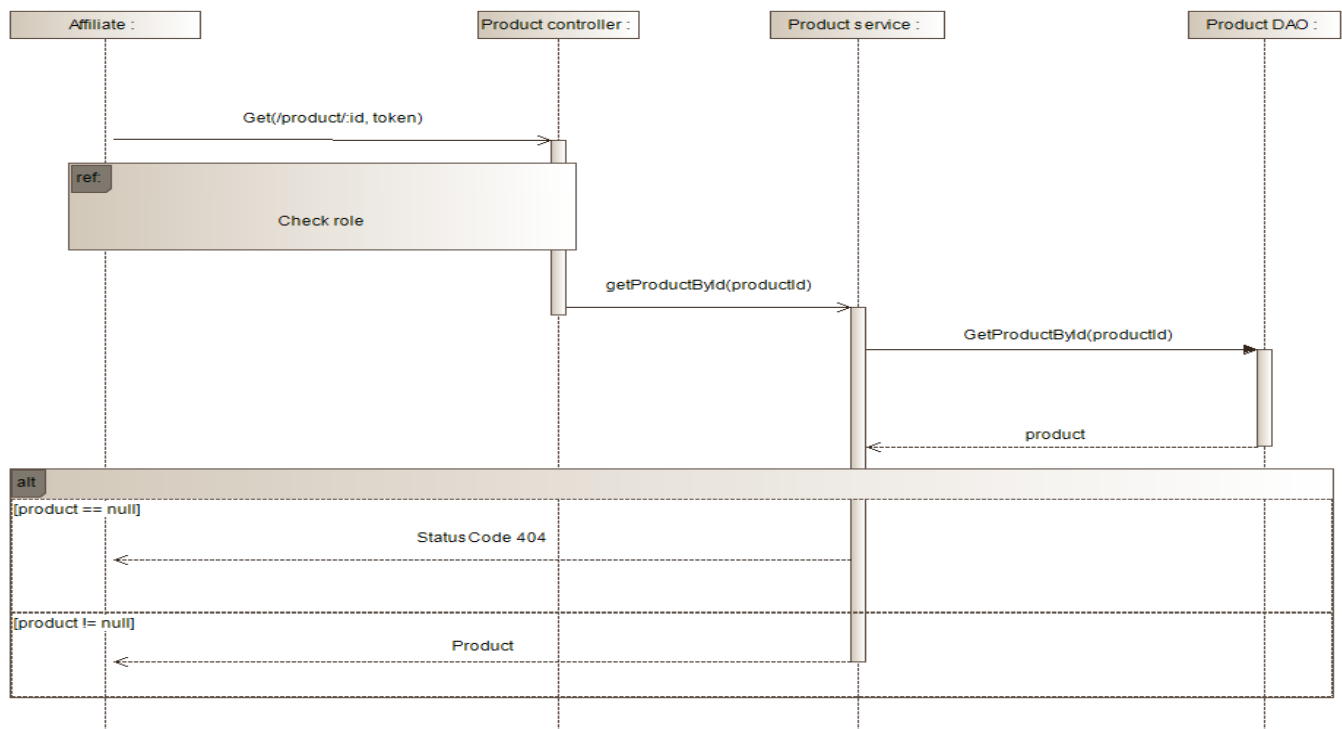


Figure 42 : Diagramme de séquence, récupération des détails d'un produit

### 3.3.2.6.d. Récupérer la liste des commandes

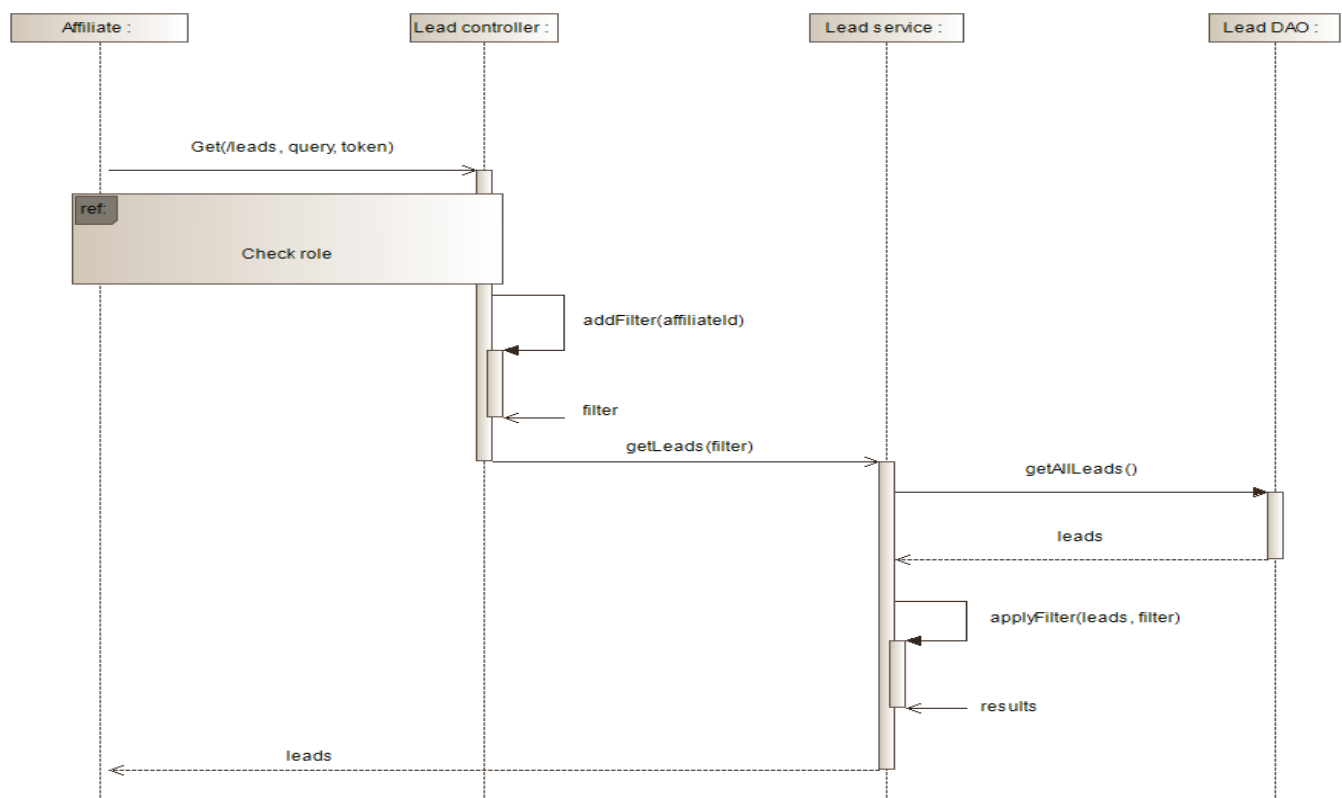


Figure 43 : Diagramme de séquence, récupération de liste des commandes

### 3.3.2.6.e. Récupérer ses propres stores

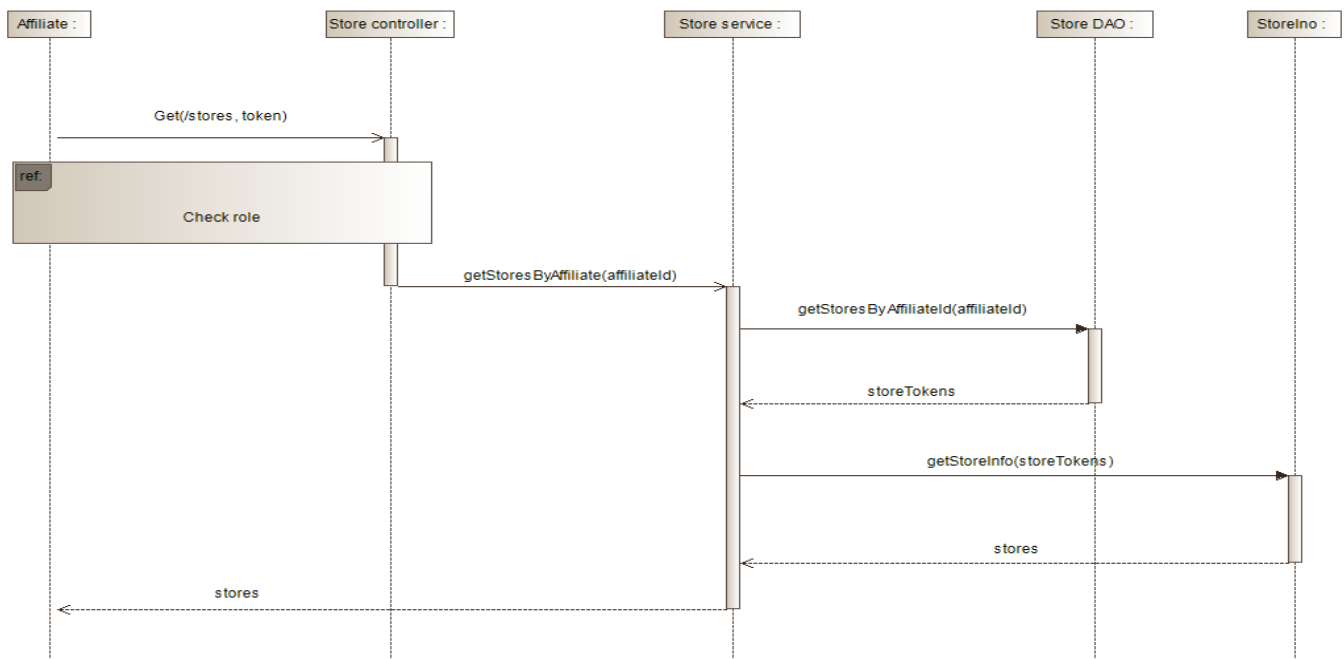


Figure 44 : Diagramme de séquence, récupération des stores

### 3.3.2.6.f. Ajouter un store

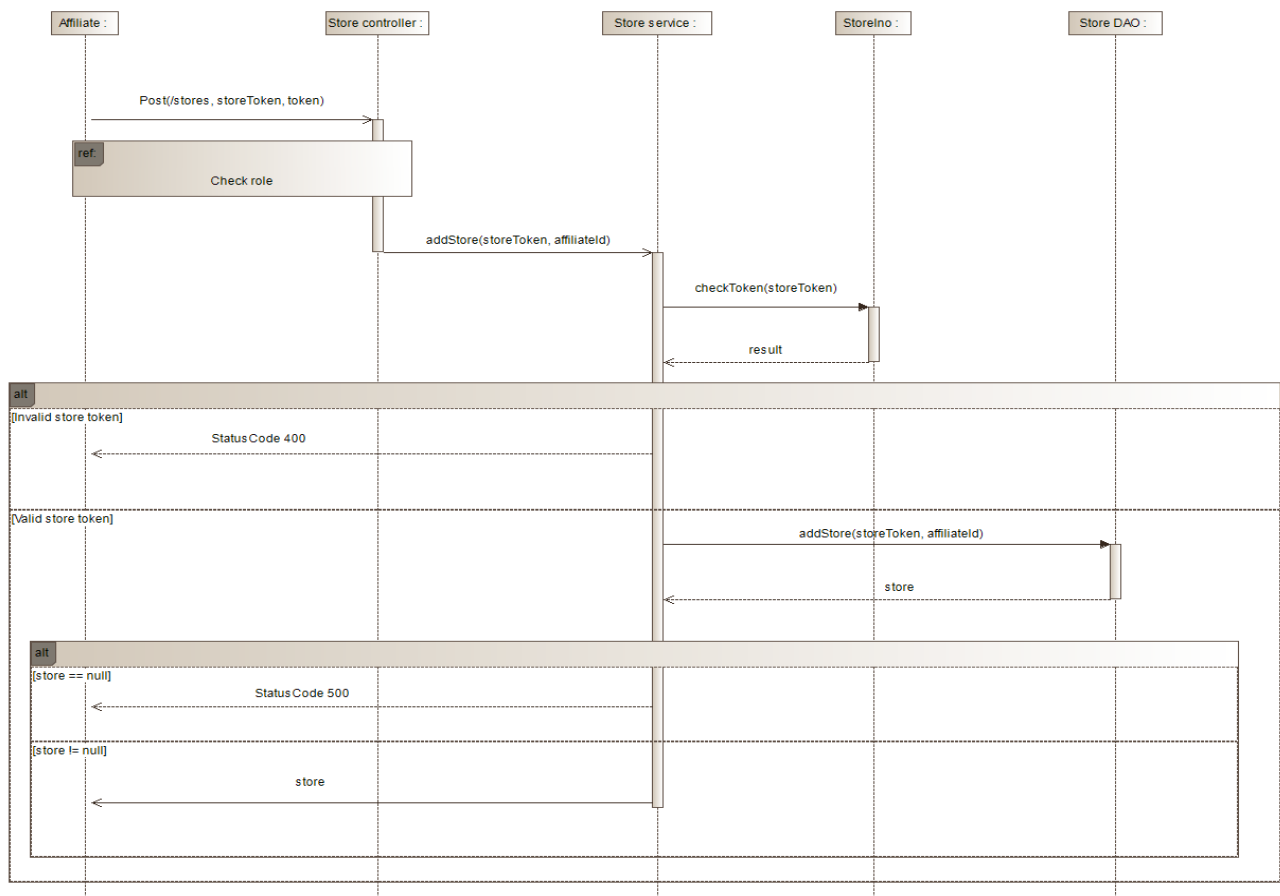


Figure 45 : Diagramme de séquence, association d'un store au compte STROEINO NETWORK

### 3.3.2.6.g. Supprimer un store

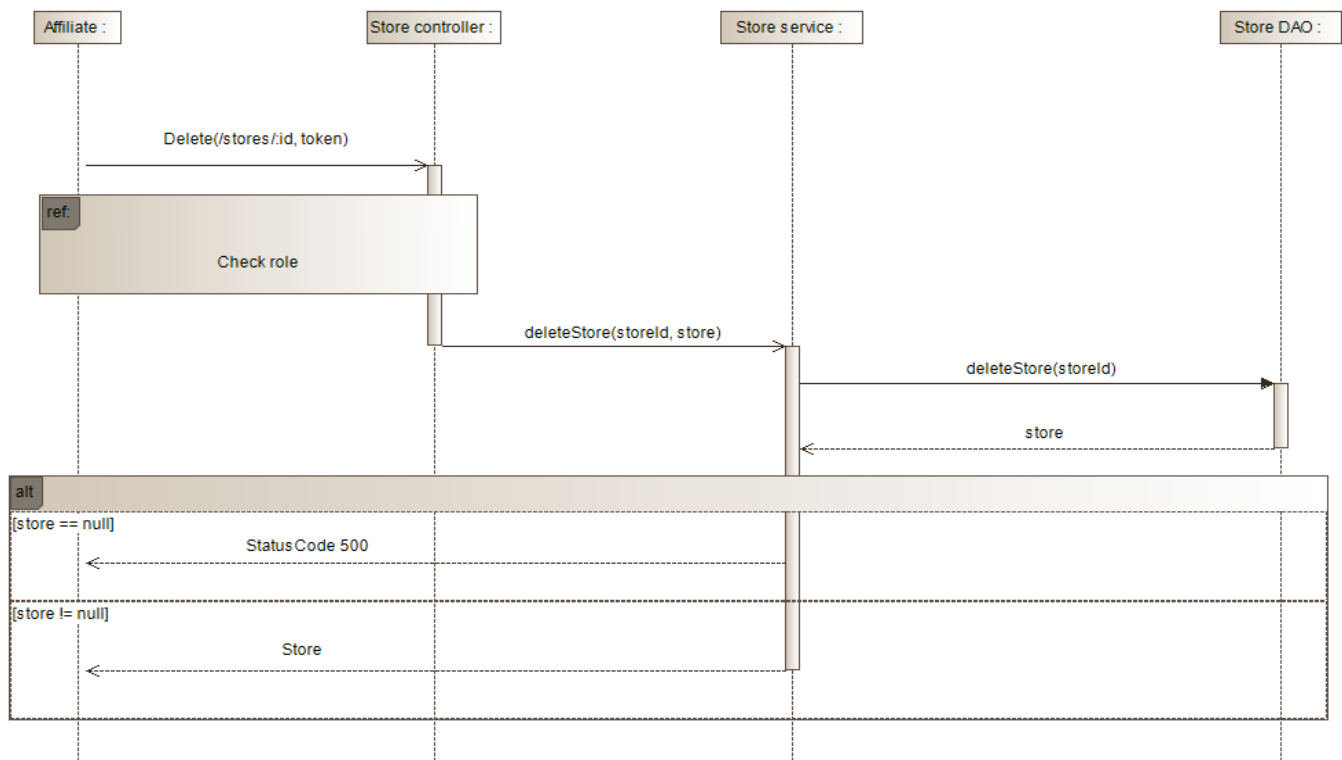


Figure 46 : Diagramme de séquence, suppression d'un store

### 3.3.2.6.h. Récupérer la liste des factures

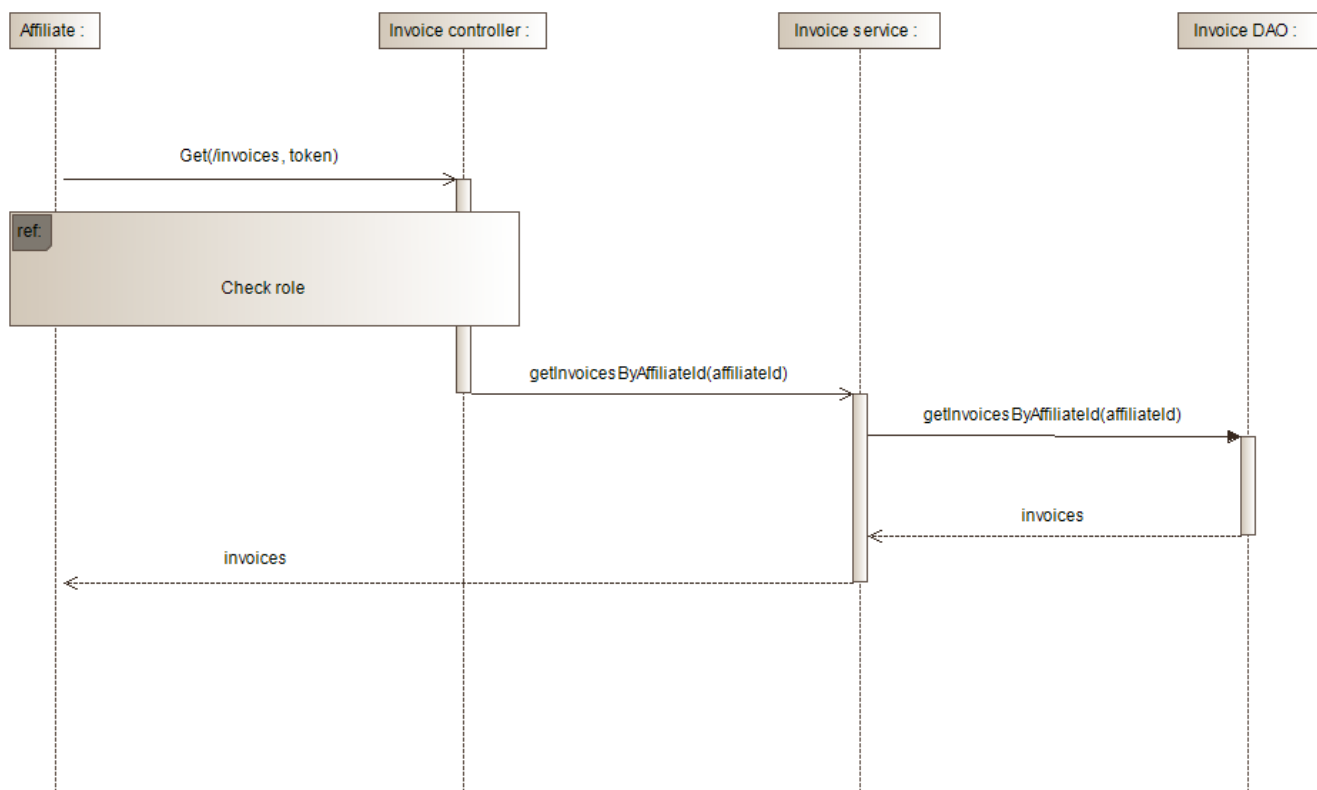


Figure 47 : Diagramme de séquence, récupération de liste des factures

### 3.3.2.6.i. Récupérer ses propres statistiques

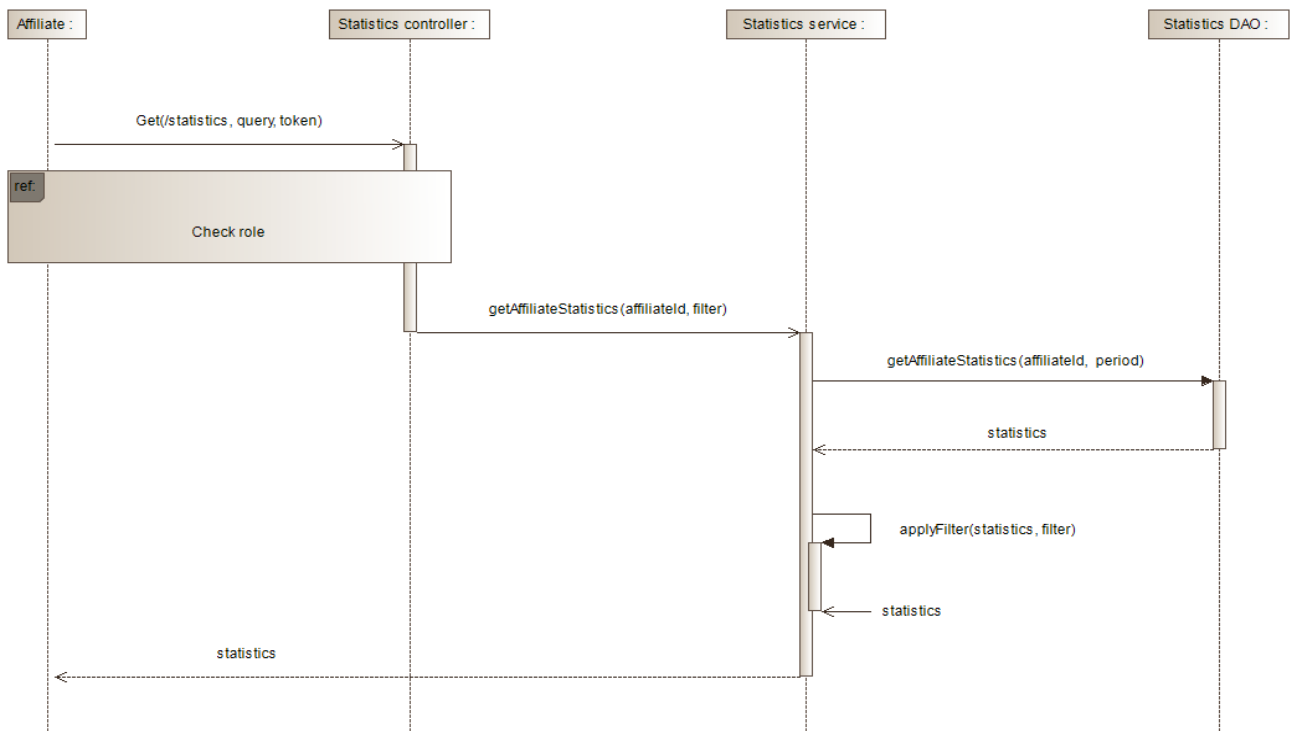


Figure 48 : Diagramme de séquence, récupération des statistiques d'un affilié

### 3.3.2.6.j. Récupérer ses informations de facturation en MAD

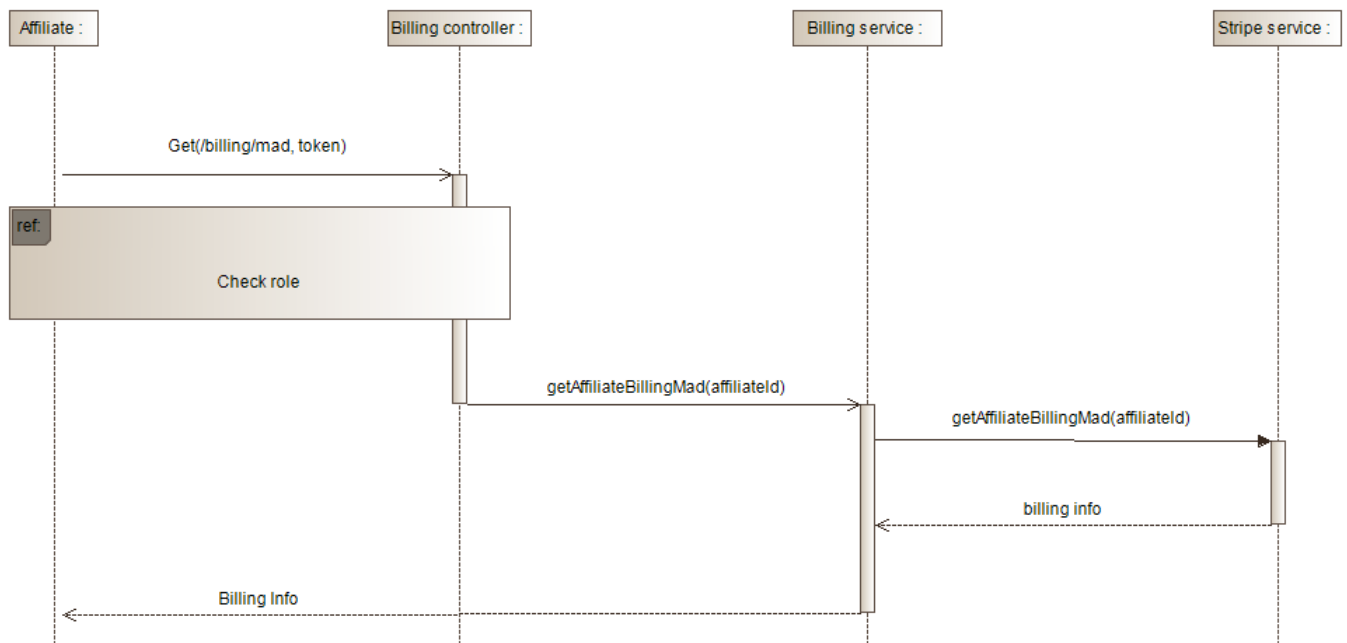


Figure 49 : Diagramme de séquence, récupération des informations de paiement MAD d'un affilié

### 3.3.2.6.k. Ajouter les informations de facturation en MAD

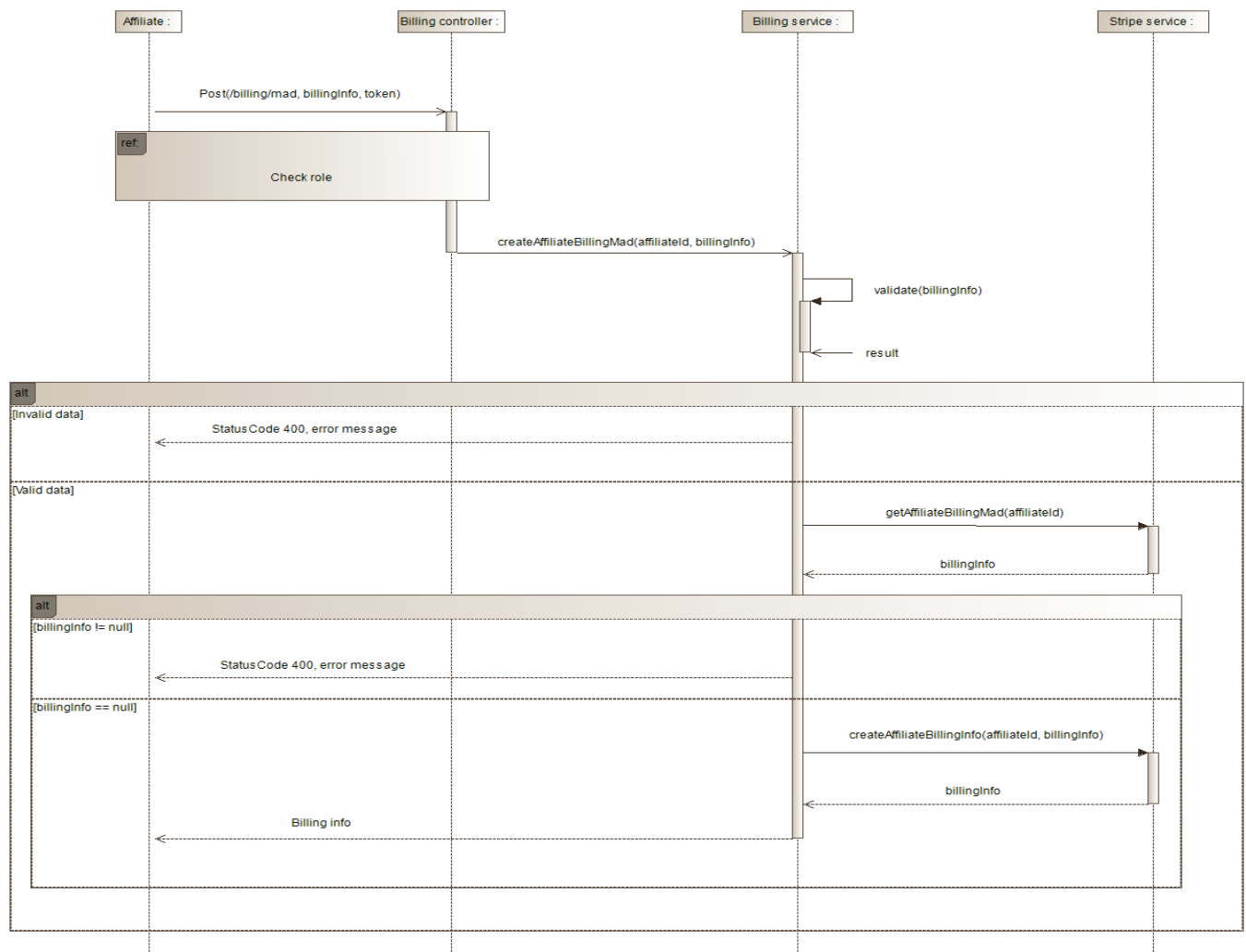


Figure 50 : Diagramme de séquence, ajout des informations de paiement MAD d'un affilié



### 3.3.2.6.l. Ajouter les informations de facturation en USD

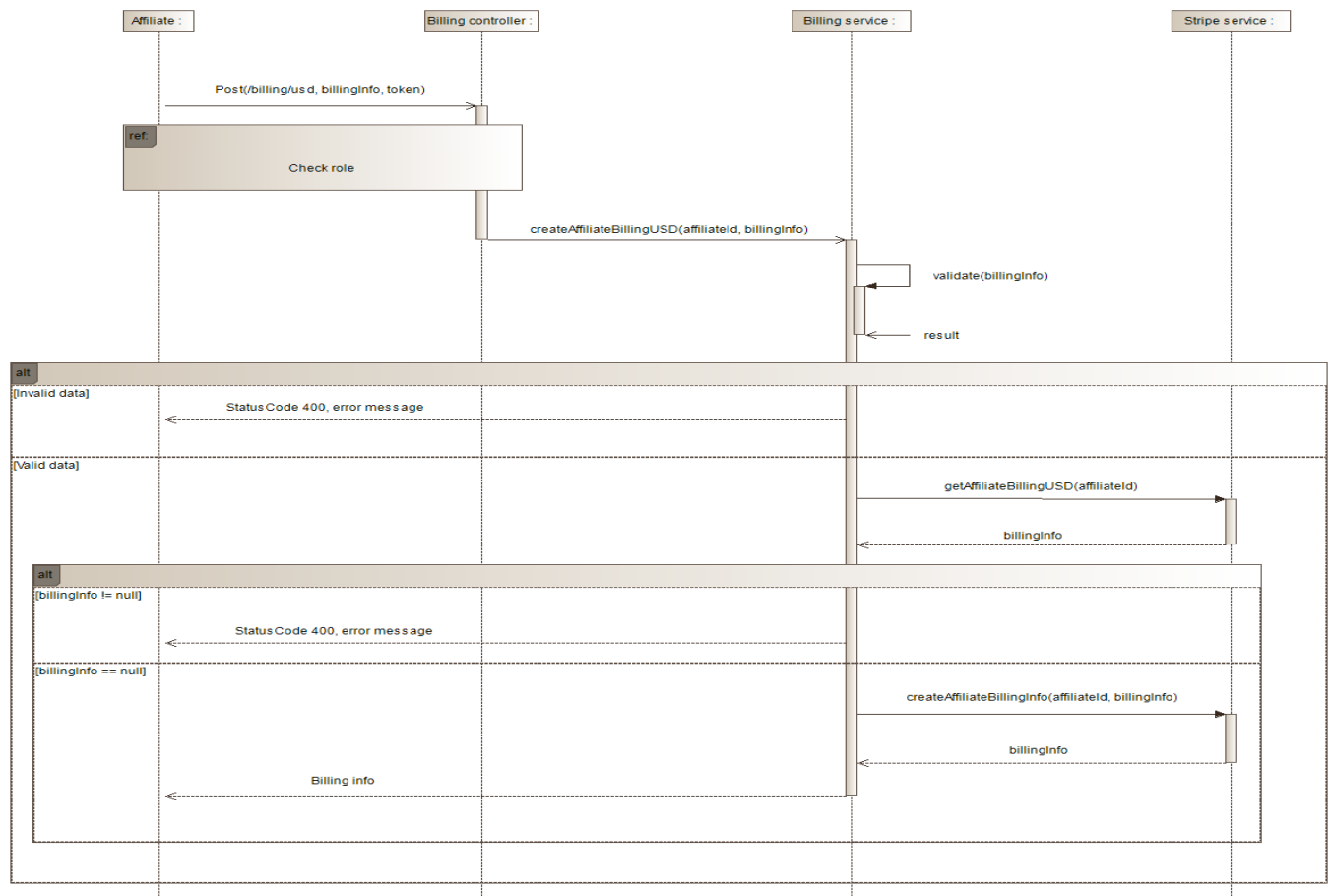


Figure 51 : Diagramme de séquence, ajout des informations de paiement USD d'un affilié

### 3.3.2.6.m. Récupérer mes informations de facturation en USD

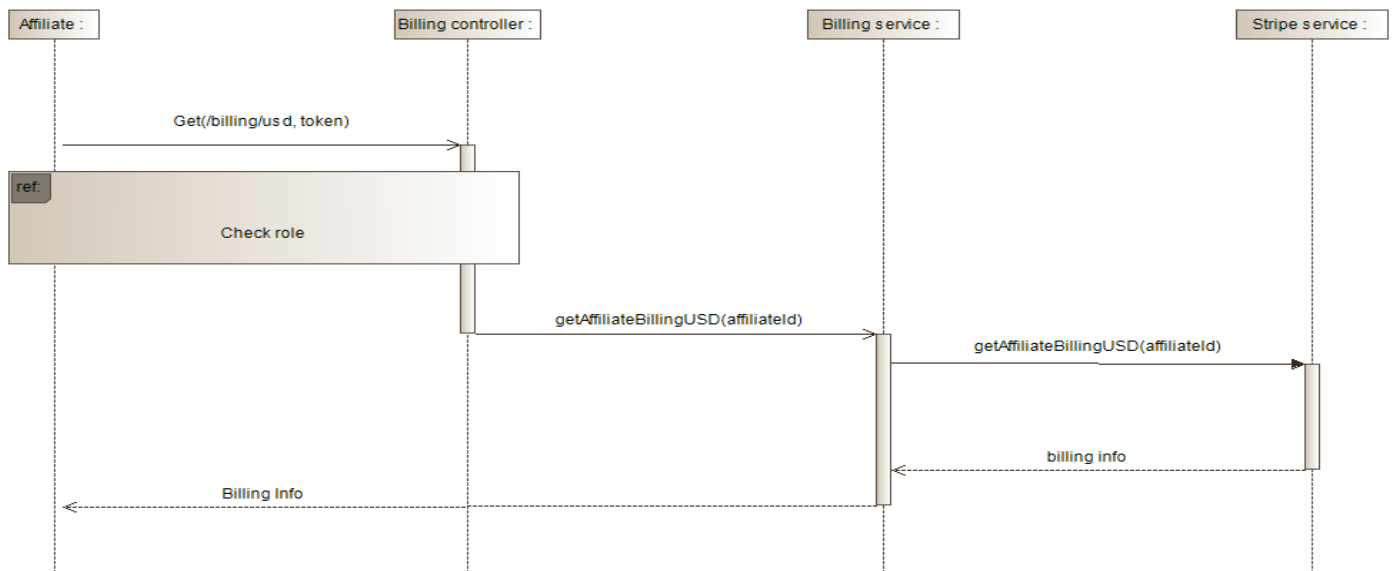


Figure 52 : Diagramme de séquence, récupération des informations de paiement USD d'un affilié

### 3.3.2.6.n. Mettre à jour les informations de facturation en USD

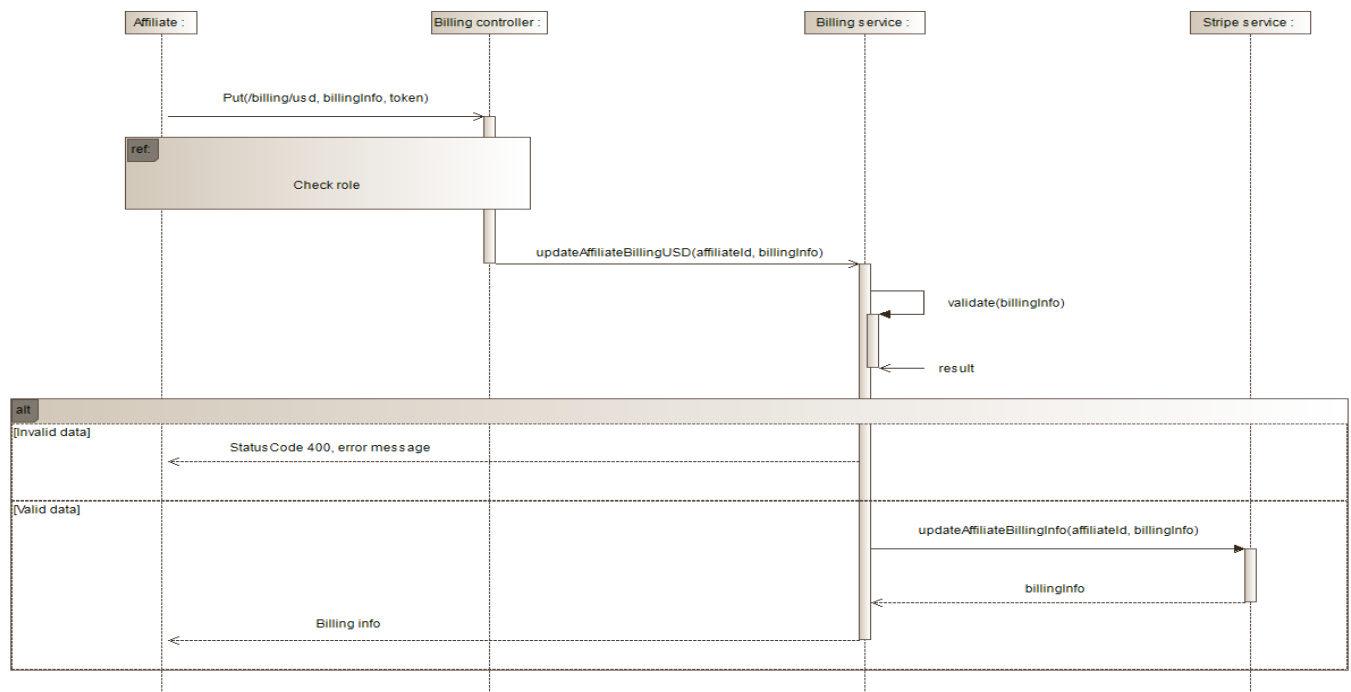


Figure 53 : Diagramme de séquence, mise à jour des informations de paiement USD d'un affilié

### 3.3.2.6.o. Récupérer la liste des visiteurs

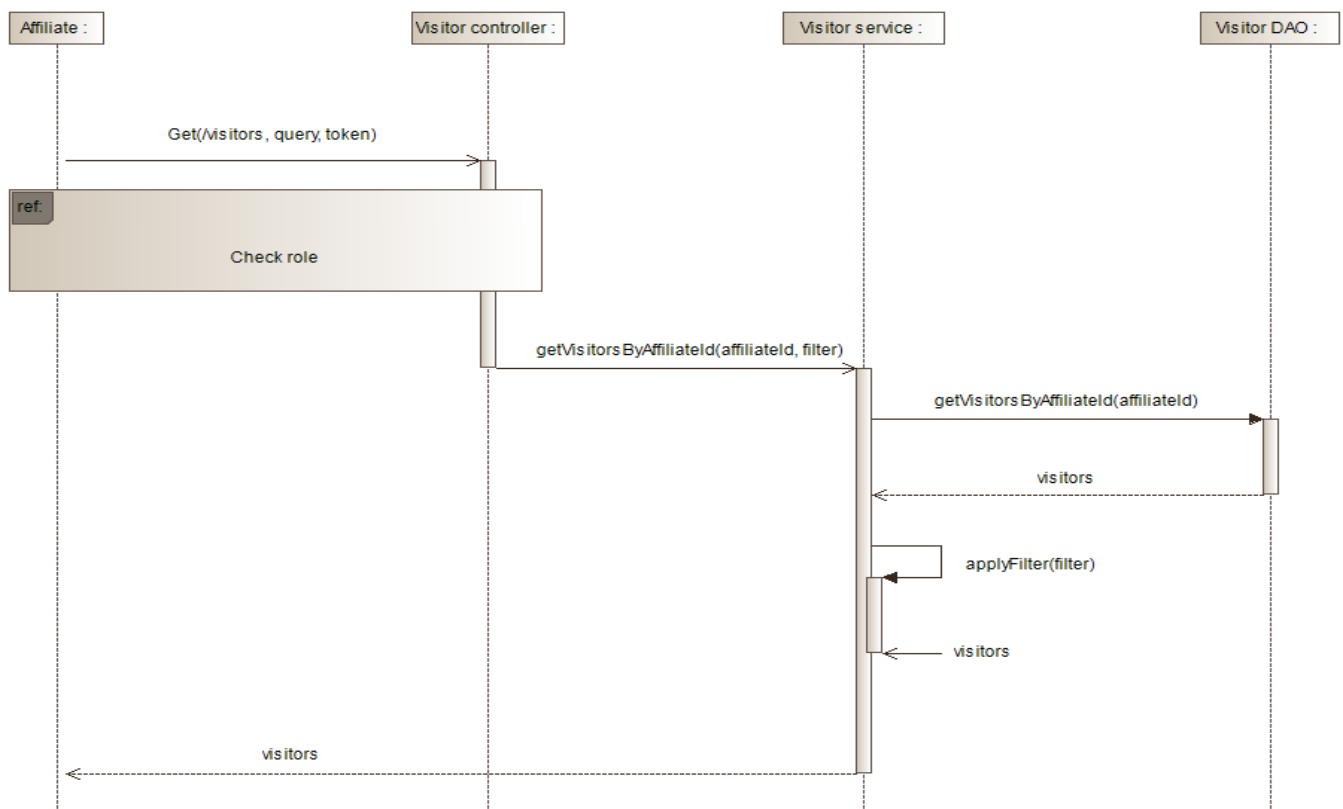


Figure 54 : Diagramme de séquence, récupération de la liste des visiteurs

### 3.3.2.6.p. Créer une demande de promotion

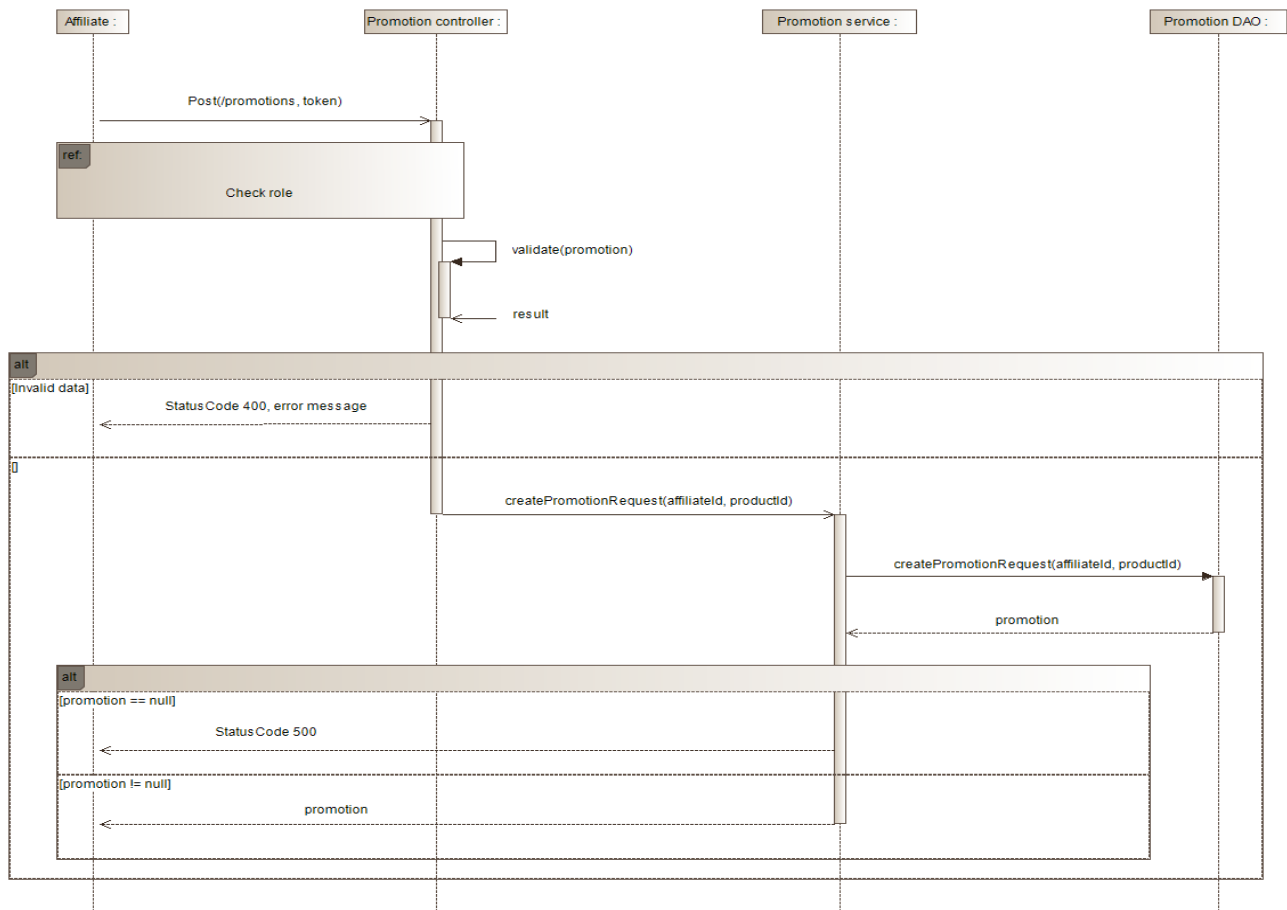


Figure 55 : Diagramme de séquence, création de demande de promotion

### 3.3.2.6.q. Récupérer la liste des promotions

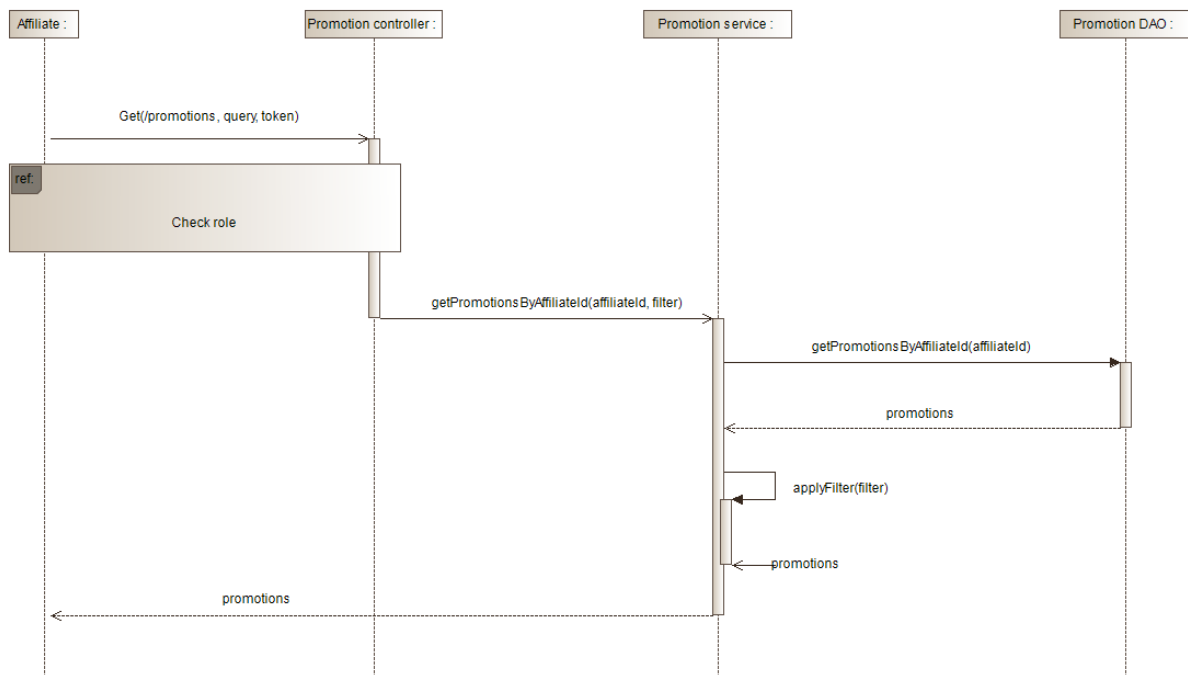


Figure 56 : Diagramme de séquence, récupération de la liste des promotions d'un affilié

### 3.3.2.7. Fournisseur

#### 3.3.2.7.a. Récupérer la liste de ses propres produits

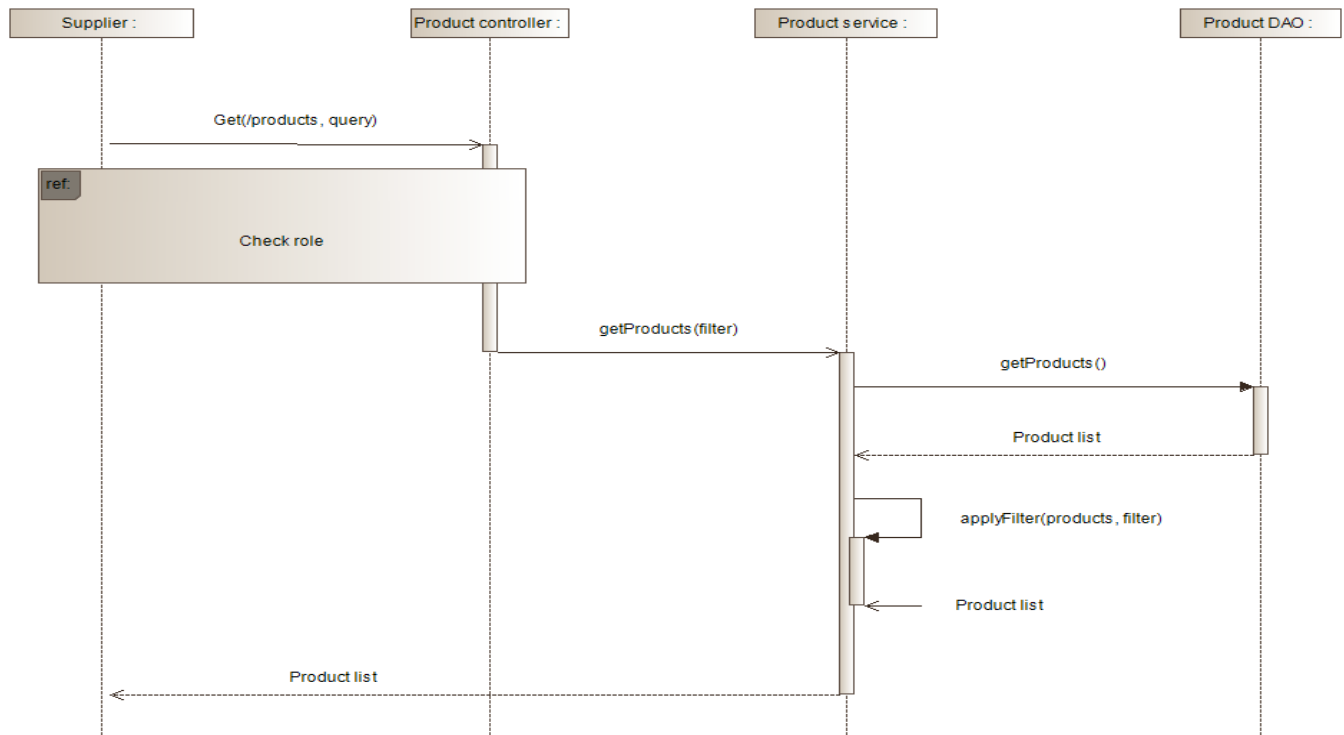


Figure 57 : Diagramme de séquence, récupération de la liste des produits d'un fournisseur

#### 3.3.2.7.b. Récupérer un produit par son identifiant

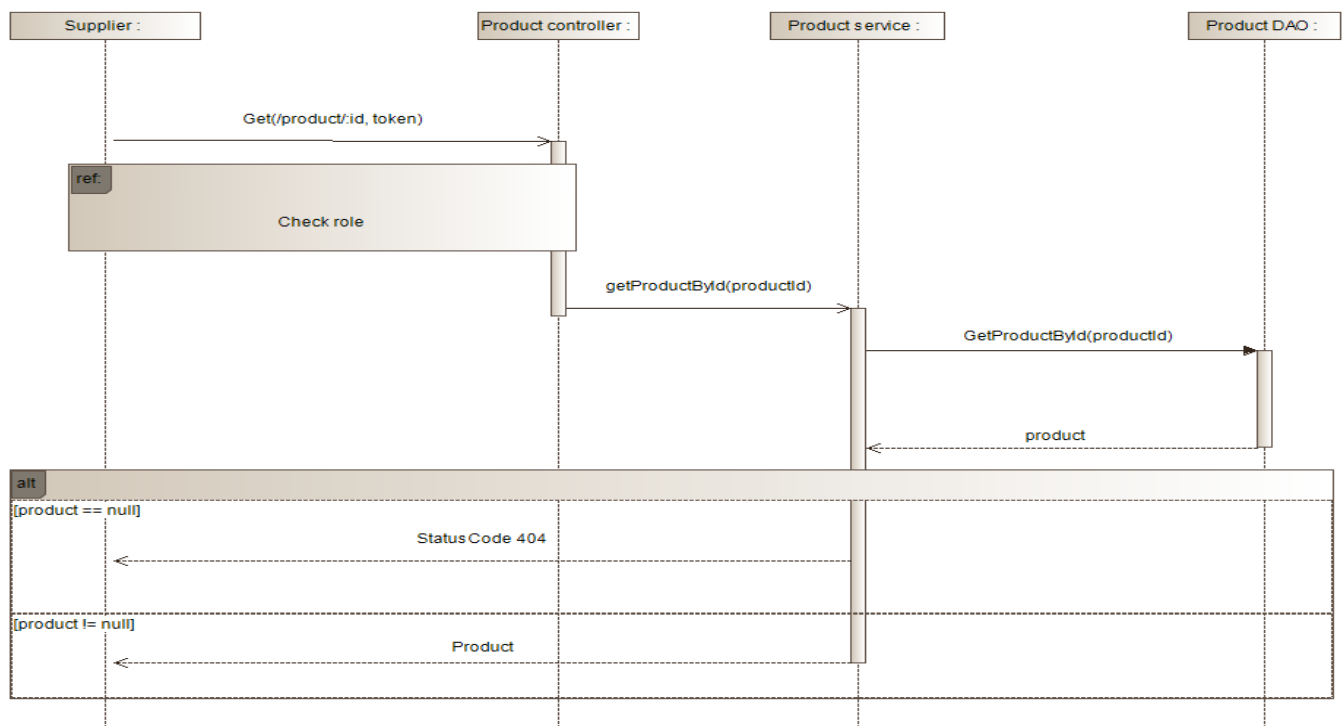


Figure 58 : Diagramme de séquence, récupération des détails d'un produit

### 3.3.2.7.c. Récupérer ses propres statistiques

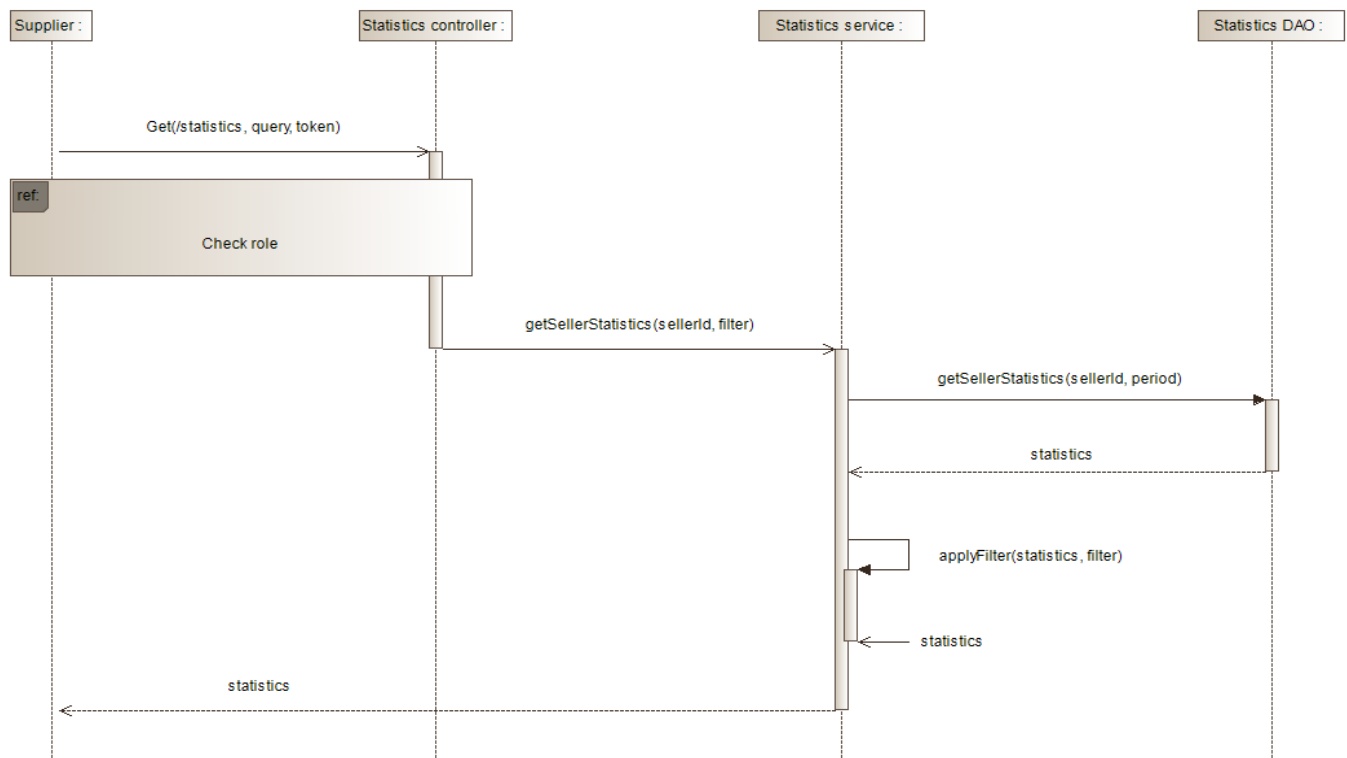


Figure 59 : Diagramme de séquence, récupération des statistiques de fournisseur

### 3.3.2.8. Gestionnaire de commande

#### 3.3.2.8.a. Récupérer la liste des commandes

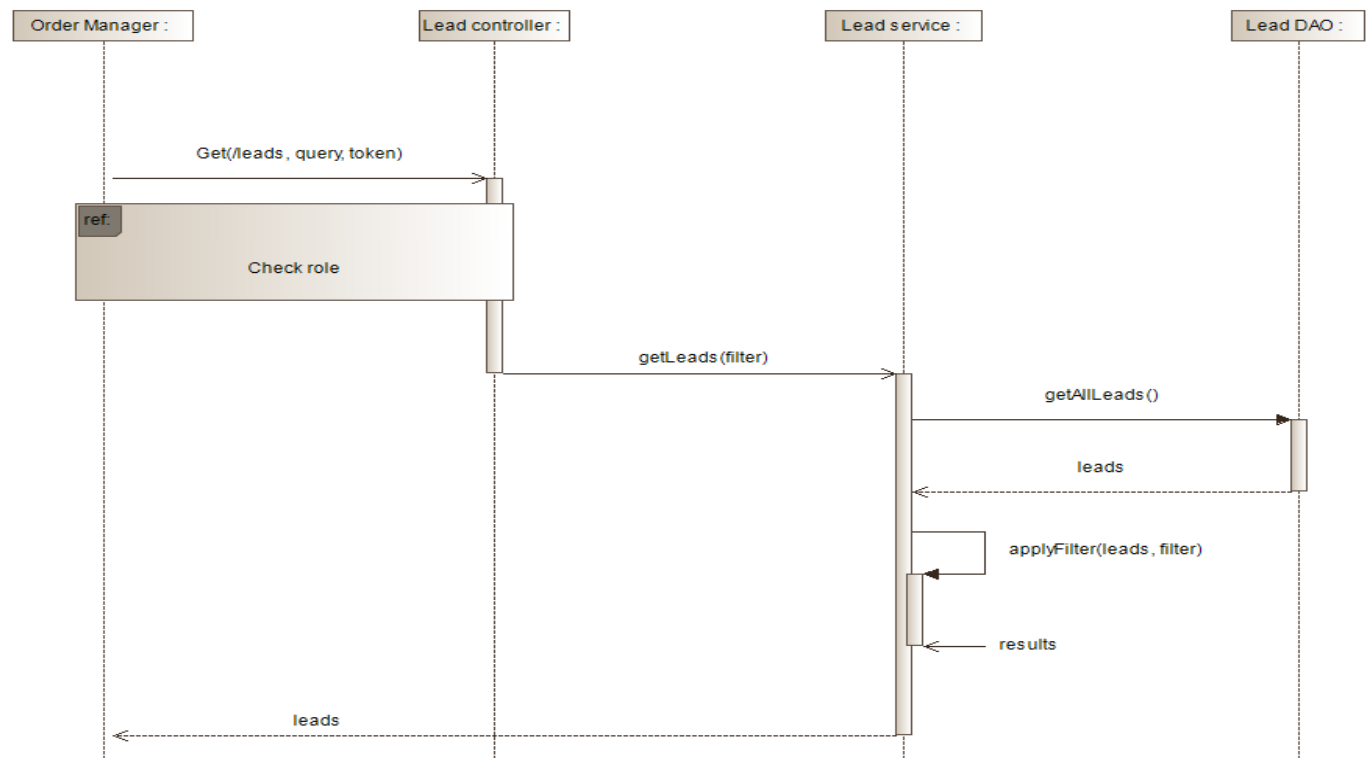


Figure 60 : Diagramme de séquence, récupération de la liste des commandes

### 3.3.2.8.b. Mettre à jours une commande

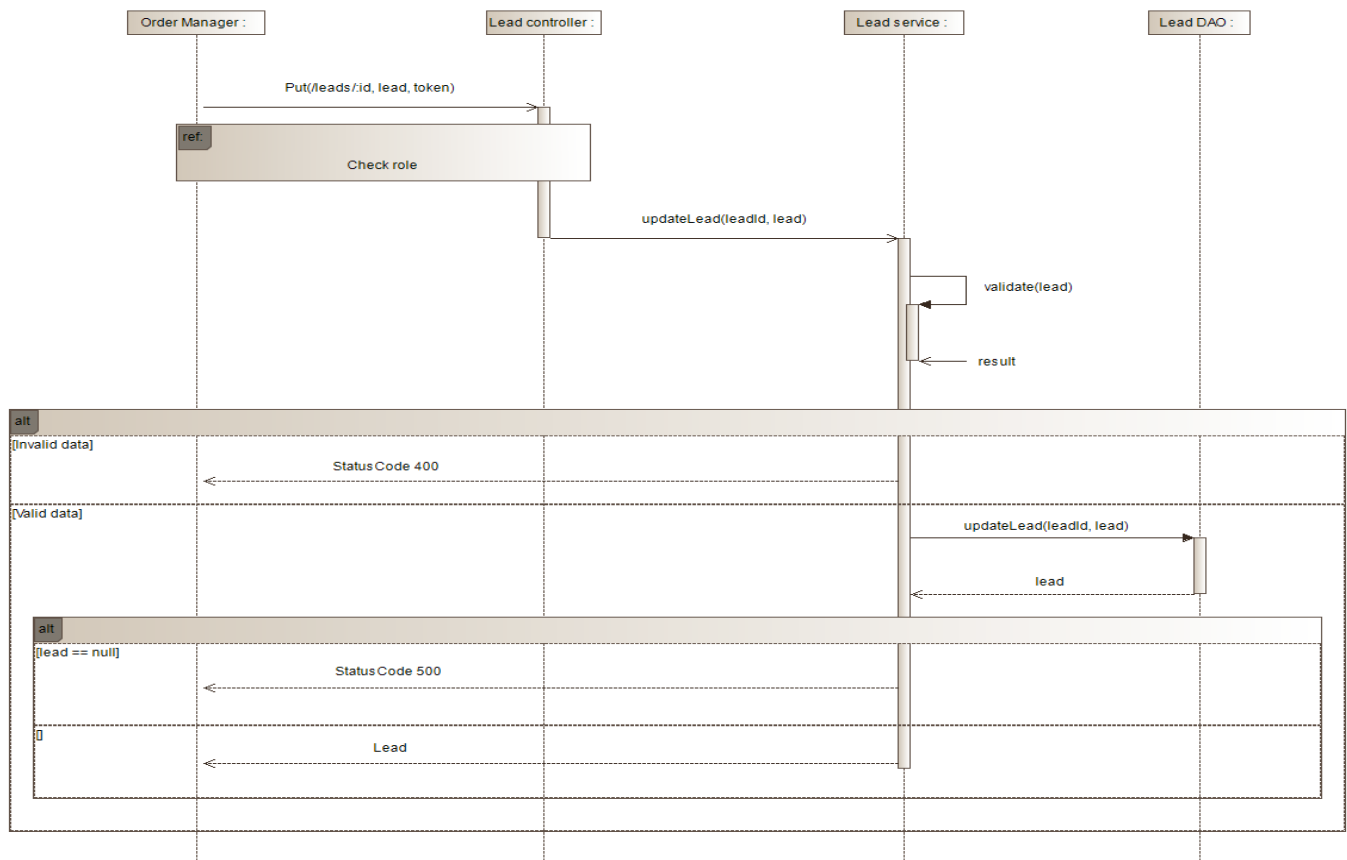


Figure 61 : Diagramme de séquence, mise à jour d'une commande

### 3.3.2.9. Visiteur

#### 3.3.2.9.a. Visualiser les pages d'accueil (landing pages)

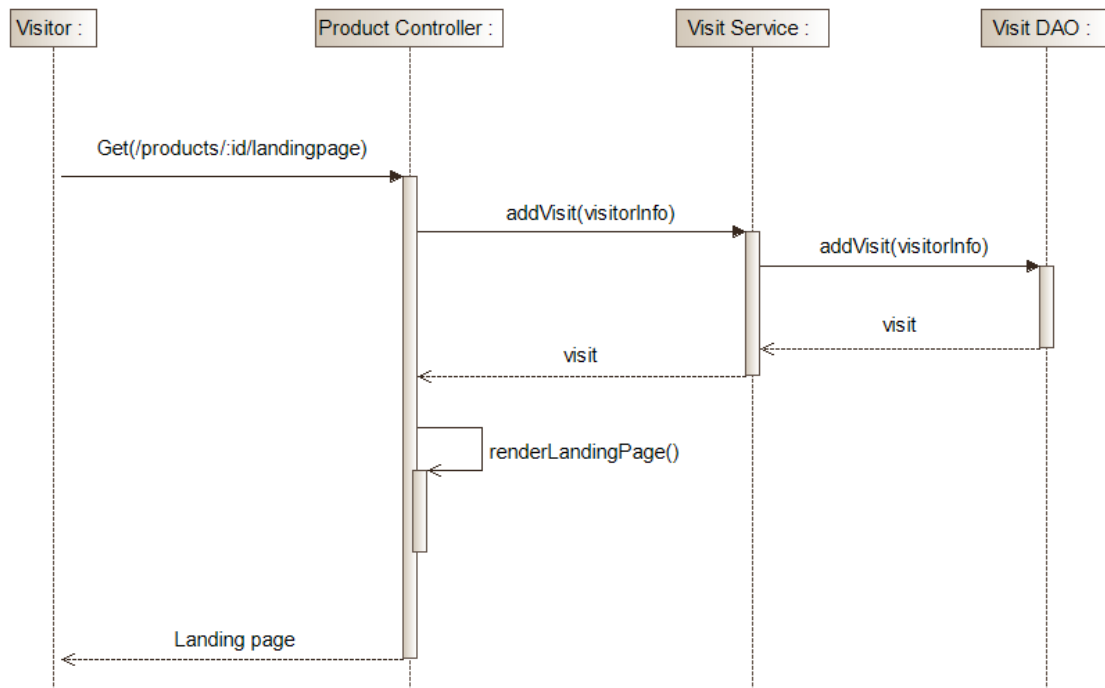


Figure 62 : Diagramme de séquence, récupération d'une landing-page d'un produit

#### 3.3.2.9.b. Commander un produit

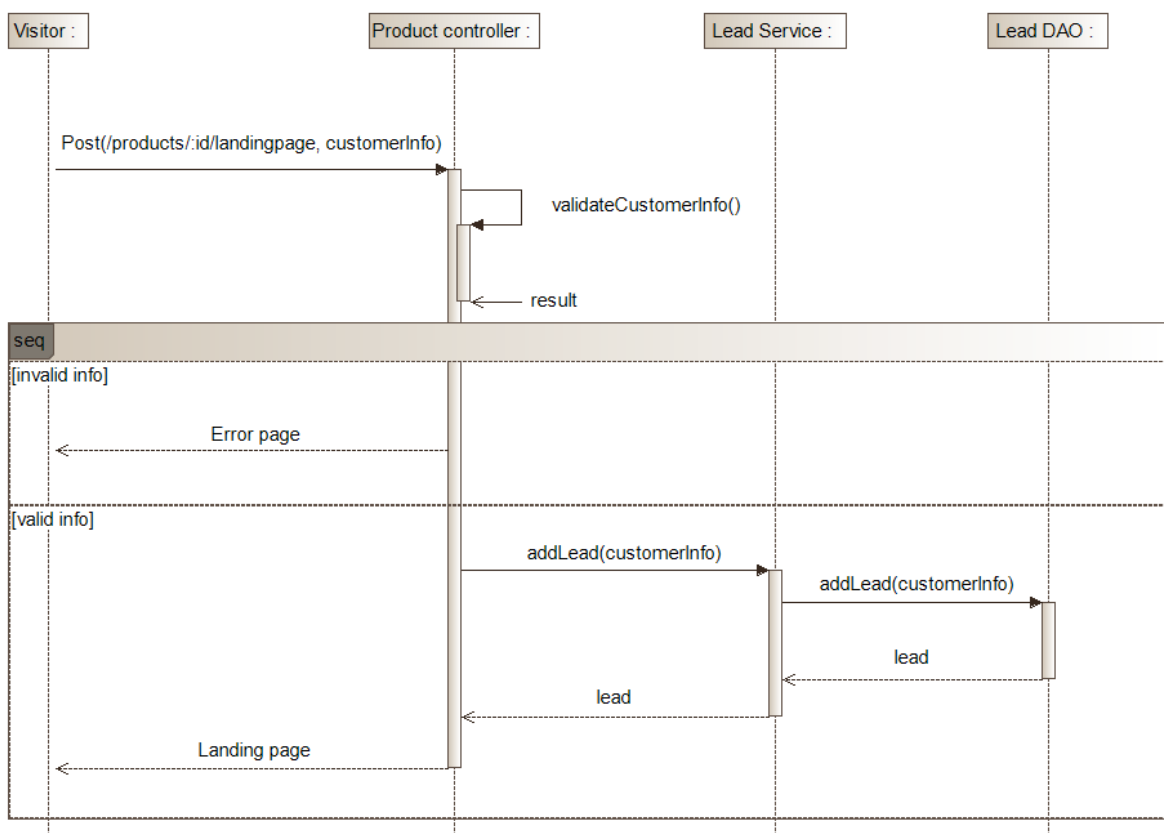


Figure 63 : Diagramme de séquence, commander un produit

## 3.4. Diagramme d'état-transition

### 3.4.1. Définition

Un diagramme d'état-transition est une représentation graphique utilisée en génie logiciel pour décrire le comportement d'un système. Il montre les différents états qu'un objet ou un système peut prendre et les transitions entre ces états déclenchées par des événements spécifiques. Il est utilisé pour modéliser et visualiser le comportement dynamique d'un système, en mettant l'accent sur les changements d'état et les transitions.

### 3.4.2. Diagrammes

#### 3.4.2.1. Diagramme d'état-transition d'un utilisateur

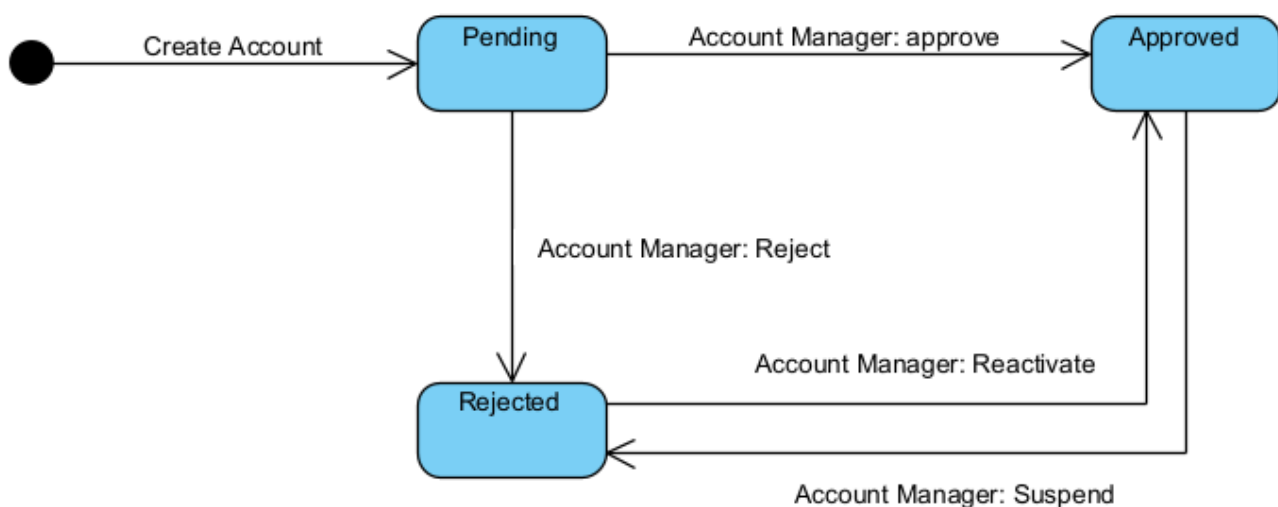


Figure 64 : Diagramme d'état-transition d'un utilisateur

#### 3.4.2.2. Diagramme d'état-transition d'une promotion

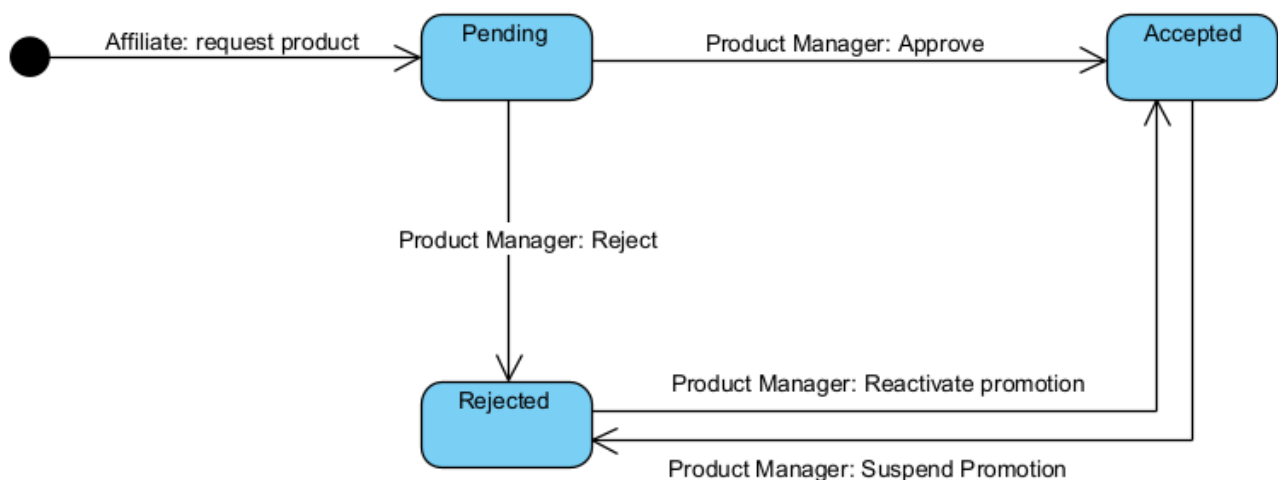


Figure 65 : Diagramme d'état-transition d'une promotion



### 3.4.2.3. Diagramme d'état-transition d'une commande

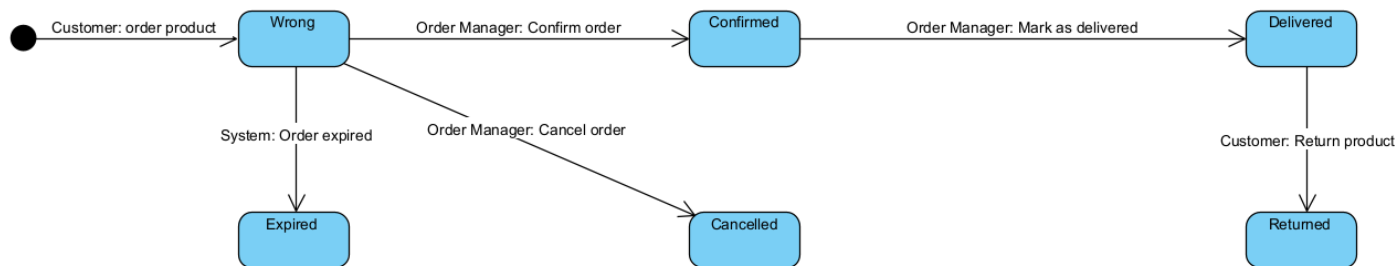


Figure 66 : Diagramme d'état-transition d'une commande

## 3.5. Conclusion du chapitre

En conclusion de ce chapitre consacré à la modélisation et à la conception de notre projet, nous avons présenté différents diagrammes qui ont permis de représenter la logique métier de manière claire et structurée. Ces diagrammes constituent des outils essentiels pour la compréhension et la communication de notre projet.

# Chapitre 4

## Etude Technique

---

## 4. Etude technique du projet

### 4.1. Introduction

Dans ce chapitre consacré à l'étude technique, nous allons présenter l'environnement de travail que nous avons utilisé pour la réalisation de ce projet. Celui-ci se divise en deux aspects fondamentaux : l'environnement matériel, comprenant les équipements physiques nécessaires, et l'environnement logiciel, qui englobe les outils et les programmes utilisés.

### 4.2. Outils et Technologies de Développement



**Node.js** est un environnement d'exécution JavaScript côté serveur, qui permet de créer des applications web dynamiques et évolutives. Contrairement à JavaScript côté client qui s'exécute dans un navigateur web, Node.js est utilisé pour exécuter du code JavaScript côté serveur. Il est basé sur le moteur JavaScript V8 de Google, et il utilise un modèle de programmation orienté événements pour permettre des opérations non bloquantes et une haute performance. Node.js est très populaire pour le développement de serveurs web, d'applications réseau, de bases de données en temps réel, de systèmes de messagerie et de beaucoup d'autres types d'applications back end. (NodeJS, n.d.)

#### Node.js Architecture

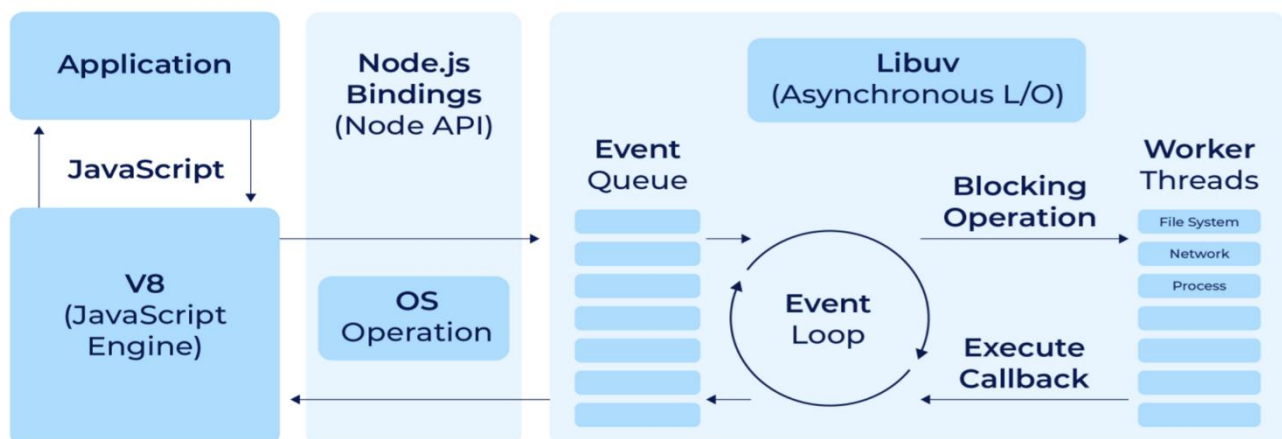


Figure 67 : Architecture NodeJS



**JavaScript** est un langage de programmation de haut niveau, dynamique et interprété, principalement utilisé pour créer des applications web interactives côté client. Il permet de manipuler le contenu **HTML**, de créer des animations, d'interagir avec des serveurs web, de valider des formulaires, d'effectuer des calculs complexes et bien plus encore. Js est également utilisé côté serveur grâce à l'environnement d'exécution **Node.js**, ce qui permet d'écrire des applications web côté serveur. Il est largement utilisé pour le développement web et mobile, mais il peut également être utilisé pour créer des applications de bureau, des jeux et des programmes d'automatisation. JS est un langage de programmation largement utilisé et il est constamment mis à jour avec de nouvelles fonctionnalités pour répondre aux besoins changeants des développeurs. (*JavaScript / MDN*, 2023)



**HTML** (Hypertext Markup Language) est le langage de balisage standard utilisé pour créer des pages web. Il est utilisé pour structurer et formater le contenu d'une page web en utilisant des balises, qui sont des éléments de code qui définissent la signification et l'apparence du contenu. Les balises HTML permettent de définir la structure de la page, d'ajouter des images, des vidéos, des liens, des formulaires et bien plus encore. HTML est le langage de base pour la création de pages web statiques, qui ne changent pas en fonction de l'interaction de l'utilisateur. (*HTML / MDN*, 2023)



**CSS** (Cascading Style Sheets) est un langage de style utilisé pour décrire la présentation et l'apparence d'une page web. Il permet de séparer la mise en forme du contenu d'une page HTML, en spécifiant les styles de mise en forme tels que la couleur, la taille, la police, la disposition, les effets de transition, etc. (*CSS / MDN*, 2023)



**Sass** (Syntactically Awesome Stylesheets) est un langage de préprocesseur qui est utilisé pour améliorer les capacités de CSS (Cascading Style Sheets). Sass permet aux développeurs web d'écrire un code CSS plus efficace, flexible et maintenable en introduisant des fonctionnalités telles que les variables, le nesting, les mixins, les fonctions et l'héritage. (*Documentation Sass*, n.d.)

## 4.3. Frameworks



**Vue.js** est un framework JavaScript progressif pour la création d'interfaces utilisateur interactives et dynamiques. Il se concentre sur la vue de l'application et offre une approche basée sur les composants pour la création d'interfaces utilisateur réutilisables et modulaires. Il est utilisé pour la construction de Single-Page Applications et permet la liaison de données bidirectionnelle, la gestion d'événements, la manipulation du DOM, la composition de composants, le routage et la gestion des états. (VueJs, n.d.)



**Express.js** est un framework web pour Node.js, un environnement d'exécution JavaScript côté serveur. Il fournit une infrastructure robuste pour la création de serveurs web basés sur Node.js, en offrant des fonctionnalités de routage, de gestion de middleware, de traitement des requêtes et de réponses, de gestion des erreurs, de gestion des sessions, de gestion des cookies, et bien plus encore.

(ExpressJs, n.d.)

## 4.4. Outils et IDE



**Visual Studio Code** (ou VS Code) est un éditeur de code source développé par Microsoft pour Windows, Linux et macOS. Il offre une expérience de développement moderne, intuitive et personnalisable pour une variété de langages de programmation, y compris JavaScript, TypeScript, Python, Java et bien d'autres. (VsCode, n.d.)



**POSTMAN**

**Postman** est un outil de développement d'API qui permet aux développeurs de tester, déboguer et documenter les API en utilisant une interface utilisateur conviviale. Il offre des fonctionnalités telles que la simulation de différentes méthodes HTTP, la capture et la gestion des réponses, ainsi que la création de collections et de scripts de tests automatisés pour améliorer la productivité des développeurs. (Postman, n.d.)



**MongoDB Compass** est une interface graphique utilisateur (GUI) pour MongoDB, une base de données NoSQL populaire. Il permet aux développeurs de visualiser, explorer et manipuler des données stockées dans des bases de données MongoDB, ainsi que de créer des requêtes et des agrégations. (MongoDB Compass, n.d.)



**Trello** est un outil de gestion de projet en ligne qui utilise une méthode appelée "tableau Kanban". Il permet de visualiser les projets en cours, les tâches à accomplir et les deadlines, en utilisant des cartes, des listes et des tableaux. (*Trello*,

n.d.)



**Git** est un outil open-source et gratuit, disponible pour plusieurs systèmes d'exploitation, notamment Linux, macOS et Windows. Il est largement utilisé dans l'industrie du développement de logiciels, ainsi que dans d'autres domaines où la gestion de code source et de fichiers est importante. (*About / Git*, n.d.)



**npm** (Node Package Manager) est un gestionnaire de paquets pour la plateforme de développement Node.js. Il permet aux développeurs de Node.js d'installer, de partager et de gérer des bibliothèques de code et des modules tiers, ainsi que de publier leurs propres packages. (*About / Npm*, n.d.)

## 4.5. Bibliothèques



**Passport** est une bibliothèque d'authentification pour Node.js. Il permet aux développeurs de gérer facilement l'authentification des utilisateurs dans leurs applications Node.js, en fournissant une interface simple et flexible pour l'authentification basée sur des stratégies. (*PassportJs*, n.d.)



**Winston** est une bibliothèque de gestion des journaux (logs) pour les applications Node.js. Il permet aux développeurs de créer des journaux d'application de manière flexible et personnalisée, en fournissant une variété de fonctionnalités pour la gestion des journaux. (*Winston*, 2023)



**Mongoose** est une bibliothèque d'ODM (Object Data Mapping) pour MongoDB et Node.js. Il fournit une interface simple et expressive pour travailler avec MongoDB en utilisant des schémas pour modéliser les données. (*Mongoose*,

n.d.)



**Joi** est une bibliothèque de validation de données pour Node.js. Il permet aux développeurs de valider les données entrant dans leur application, en s'assurant qu'elles respectent les règles et les contraintes spécifiées. (*Joi*, n.d.)



**bcrypt** est une implémentation logicielle de l'algorithme de hachage de mot de passe bcrypt. Elle fournit des fonctions permettant de générer des hachages de mots de passe sécurisés en utilisant l'algorithme bcrypt, ainsi que des fonctions pour comparer un mot de passe en clair avec un hachage de mot de passe stocké. (*Bcrypt*, 2022)



**JWT** (JSON Web Tokens) est un framework d'authentification et d'autorisation pour les applications web. Il permet de créer des jetons sécurisés sous forme de chaînes de caractères encodées en base64 qui contiennent des informations sur l'utilisateur authentifié et les autorisations accordées. (*JWT - Jwt.io*, n.d.)



**Axios** est un framework JavaScript qui permet d'effectuer des requêtes HTTP depuis un navigateur ou un serveur Node.js. Il fournit une interface simple et intuitive pour effectuer des requêtes AJAX et récupérer des données depuis des APIs externes. (*Axios*, n.d.)



**Vuetify** est une bibliothèque de composants d'interface utilisateur pour Vue.js, qui fournit des composants prêts à l'emploi pour la création d'applications web modernes et responsives. Cette bibliothèque de composants est conçue pour faciliter le processus de développement et améliorer la productivité en fournissant des composants d'interface utilisateur préconçus, réutilisables et hautement personnalisables. (*Vuetify*, n.d.)



**Iconify** est un framework d'icônes pour les développeurs web. Il permet d'inclure des icônes dans des applications web en utilisant une syntaxe simple et intuitive, et offre une bibliothèque d'icônes gratuite et open source. (*Iconify Documentation*, 2023)



**CASL** est un framework JavaScript open source qui permet de gérer les autorisations dans les applications web en définissant des règles d'autorisation déclaratives et en utilisant des stratégies d'autorisation pour les vérifier. (CASL, n.d.)



**Pinia** est un framework d'état de gestion de données pour Vue.js. Il permet de gérer l'état global de l'application de manière intuitive, en utilisant les fonctionnalités les plus récentes de Vue 3 et en évitant les inconvénients des solutions plus anciennes. (Pinia, n.d.)

## 4.6. Bibliothèques de test



**Jest**

**Jest** est un framework de test JavaScript open source développé par Facebook. Il est principalement utilisé pour tester des applications web construites avec des frameworks populaires tels que React, Vue.js, Angular et Node.js. (Jest, n.d.)

## 4.7. SGBD



**mongoDB**

**MongoDB** est un système de gestion de base de données NoSQL (Not Only SQL) open source, orienté document, créé en 2007. Il est conçu pour stocker et gérer de grands volumes de données semi-structurées et non structurées, ce qui en fait un choix populaire pour les applications web, mobiles et IoT (Internet of Things). (MongoDB, n.d.)

## 4.8. Conclusion du chapitre

Dans ce chapitre, nous avons présenté divers outils et technologies utilisés tout au long du projet. Les outils de conception, tels que les logiciels de modélisation, ont permis une planification visuelle des étapes. Les langages de programmation, frameworks et bibliothèques ont été utilisés pour le développement des fonctionnalités. Les outils de gestion de projet, comme les plateformes de suivi des tâches, ont assuré une gestion efficace. Ces outils et technologies ont été indispensables pour mener à bien chaque aspect du projet, en garantissant une conception solide, un développement robuste et une gestion optimale des ressources.



# Chapitre 5

## Implémentation

---

## 5. Implémentation et structure de projet

Dans cette section finale, nous sommes ravis de vous présenter de manière exhaustive les résultats tangibles de notre travail à travers la visualisation de captures d'écran représentant les interfaces soigneusement conçues. Ces captures d'écran, obtenues à partir d'une variété de scénarios d'utilisation, permettront aux lecteurs de saisir pleinement l'aspect visuel et l'ergonomie de notre projet, mettant en évidence l'effort considérable que nous avons déployé pour créer une expérience utilisateur immersive et intuitive. Chaque détail, du choix des couleurs à la disposition des éléments interactifs, a été minutieusement pensé pour garantir une navigation fluide et agréable.

En outre, pour souligner la rigueur et la profondeur de notre démarche, nous inclurons également des extraits de code pertinents illustrant les étapes cruciales de la mise en œuvre de notre projet. Ces extraits, soigneusement sélectionnés parmi notre base de code complète, mettront en évidence les choix techniques judicieux que nous avons faits pour résoudre des défis spécifiques.

### 5.1. Backend

#### 5.1.1. La structure du projet

La structure d'un projet Node.js joue un rôle crucial dans l'organisation efficace du code et facilite le processus de développement. En adoptant une approche couramment utilisée, on divise le projet en différents répertoires et fichiers en fonction de leur fonctionnalité respective. Le répertoire principal du projet contient généralement les fichiers de configuration essentiels tels que `package.json`, qui répertorie les dépendances et les scripts de démarrage. Cette organisation claire et bien définie permet de maintenir, tester et étendre le projet Node.js de manière collaborative plus facilement. En fait, en suivant une structure solide, les développeurs peuvent travailler de manière plus efficace, en se concentrant sur les parties spécifiques du projet sans se perdre dans un enchevêtrement de fichiers. La capture d'écran ci-jointe illustre la structure que nous avons mise en place pour notre backend, offrant une vision visuelle de cette organisation soignée.

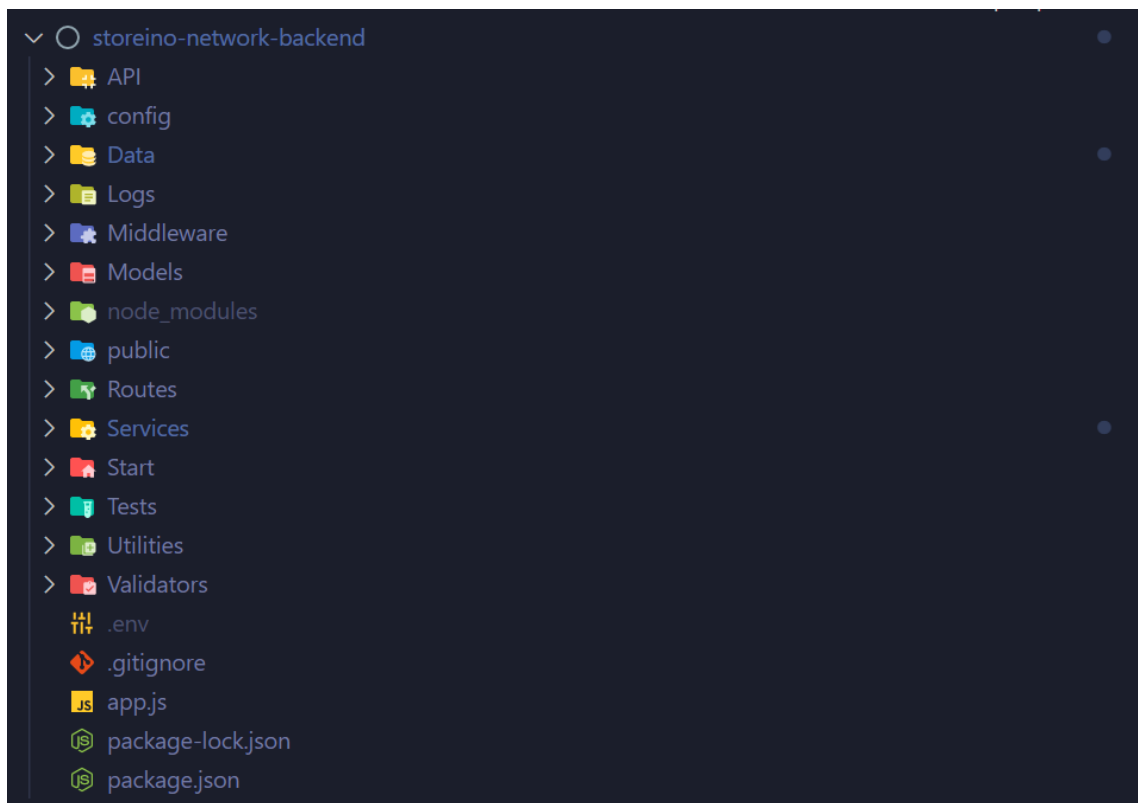


Figure 68 : Structure global du projet

### 5.1.2. Répertoire "API"

Ce répertoire regroupe les fichiers définissant les endpoints de l'API de notre application. Il est subdivisé en plusieurs modules pour organiser de manière structurée les différentes ressources de l'API, telles que les utilisateurs, les produits, etc.

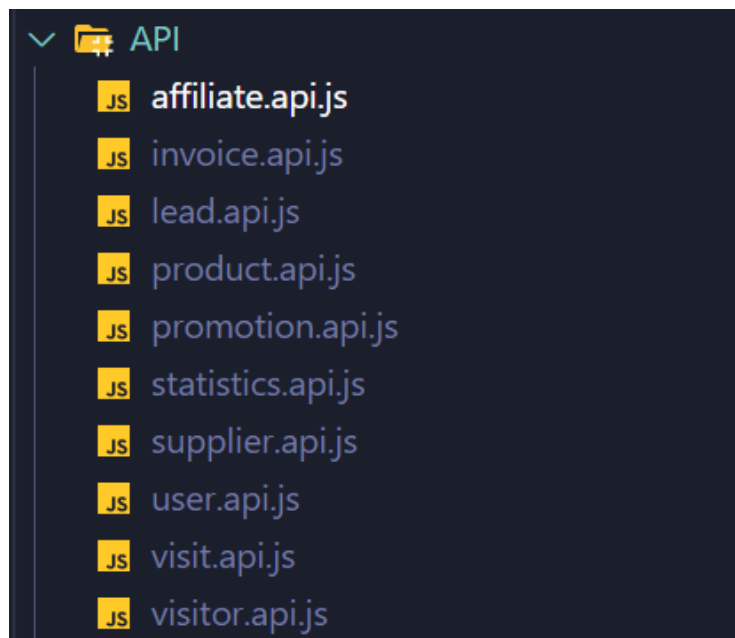


Figure 69 : Répertoire API

### 5.1.3. Répertoire "config"

Dans ce répertoire, nous plaçons les fichiers de configuration de notre site web, tels que les paramètres de base de données, les clés d'API, les variables d'environnement, et bien d'autres. Ces fichiers centralisent les paramètres de configuration, facilitant ainsi leur utilisation dans le code.



Figure 70 : Répertoire config

### 5.1.4. Répertoire "Data"

Le répertoire "data", également connu sous le nom de "dao" selon les préférences de certains, joue un rôle fondamental en tant que couche la plus basse de l'architecture de notre projet. Il abrite les fichiers essentiels qui définissent les opérations CRUD (Create, Read, Update, Delete). Ces opérations sont responsables de la manipulation des données au niveau le plus fondamental de notre application. Le répertoire "data" garantit une séparation claire entre la logique métier et la persistance des données, offrant ainsi une structure organisée et maintenable à notre projet.

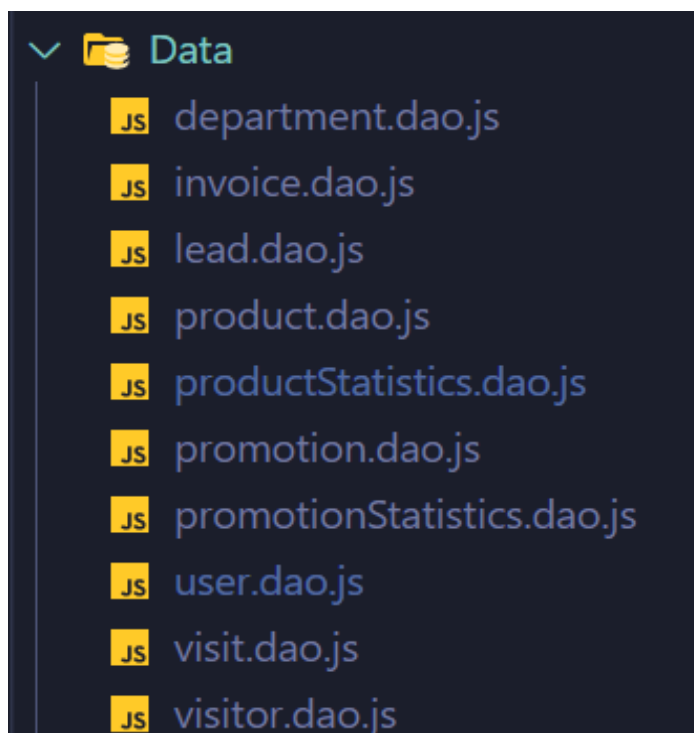


Figure 71 : Répertoire Data

### 5.1.5. Répertoire "Logs"

Ce répertoire est utilisé pour stocker les fichiers de logs générés par l'application, que ce soit le trafic de Morgan ou les logs levés explicitement par les fonctions offertes par Winston. Les logs sont essentiels pour le suivi des erreurs, le débogage et l'analyse des performances.

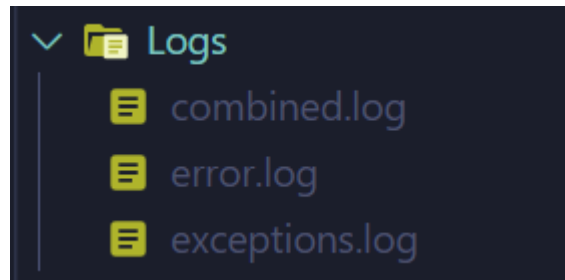


Figure 72 : Répertoire Logs

### 5.1.6. Répertoire "Middleware"

Ce répertoire contient les fichiers qui définissent des middlewares personnalisés pour notre site web. Les middlewares sont des fonctions intermédiaires qui peuvent être utilisées pour effectuer des opérations telles que la validation des données, l'authentification des utilisateurs, la gestion des erreurs, etc., avant que les requêtes n'atteignent les routes.

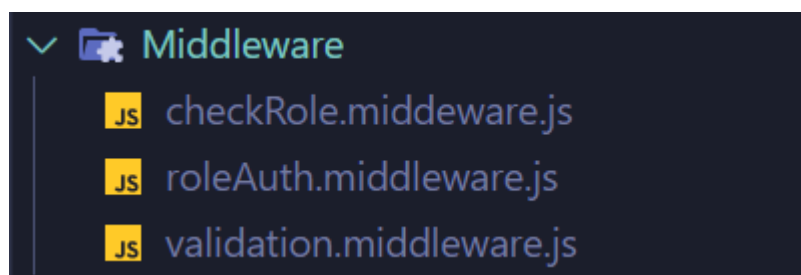


Figure 73 : Répertoire Middleware

### 5.1.7. Répertoire "Models"

Ce répertoire contient les fichiers qui définissent les modèles de données de notre application. Les modèles représentent la structure et les règles de manipulation des données au sein de l'application. Nous utilisons la bibliothèque d'ODM Mongoose pour faciliter la gestion de ces modèles.

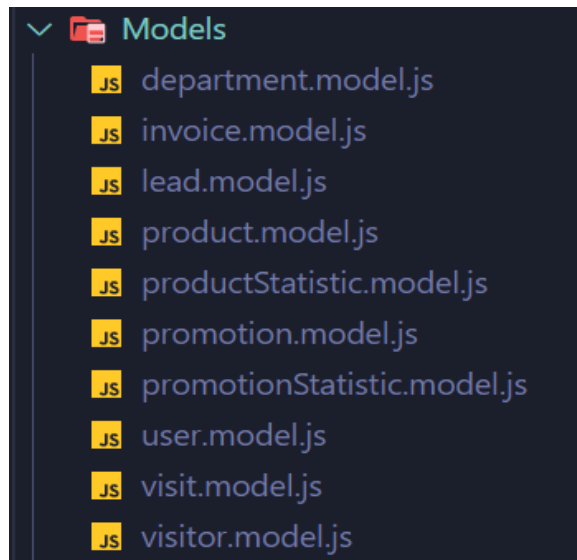


Figure 74 : Répertoire Models

### 5.1.8. Répertoire "Start"

Ce répertoire contient les fichiers de démarrage pour l'application, tels que le démarrage du serveur HTTP, la connexion à la base de données, le chargement des différents modules et l'initialisation de la base de données, etc.

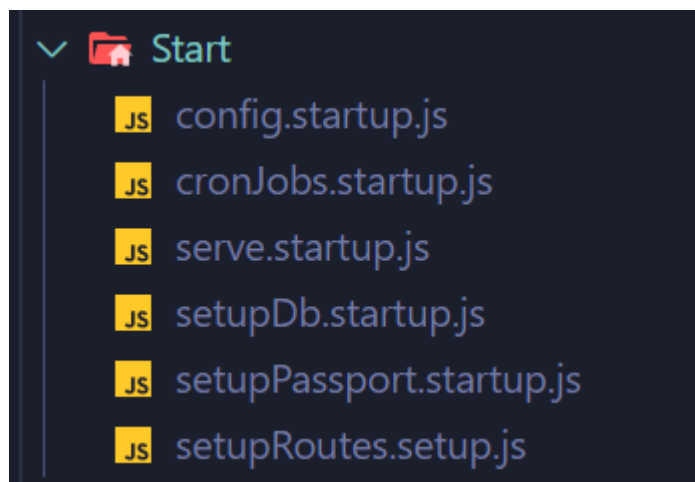


Figure 75 : Répertoire Start

### 5.1.9. Répertoire "node\_modules"

C'est le répertoire où toutes les dépendances installées via npm sont stockées. Il est généré automatiquement lorsque l'on exécute la commande d'installation des dépendances.

### 5.1.10. Répertoire "Public"

Ce répertoire contient les fichiers statiques accessibles au public, tels que les fichiers HTML, CSS, JavaScript, images, etc. Nous l'utilisons comme dossier de sauvegarde des images qui seront utilisées par le front-end.

### 5.1.11. Répertoire "Routes"

Ce répertoire contient les fichiers qui définissent les routes de l'application, c'est-à-dire les URL qui mènent à différentes fonctionnalités. Les routes définissent les actions à effectuer en réponse aux requêtes HTTP reçues par le serveur.

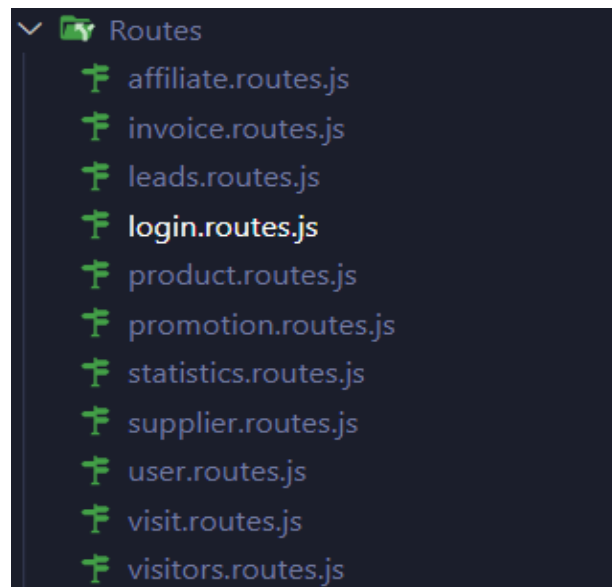


Figure 76 : Répertoire Routes

### 5.1.12. Répertoire "Services"

Dans ce répertoire, nous plaçons les fichiers qui encapsulent la logique métier de l'application. Les services fournissent des fonctionnalités spécifiques et peuvent être utilisés par les contrôleurs ou les routes pour traiter les requêtes et manipuler les données.

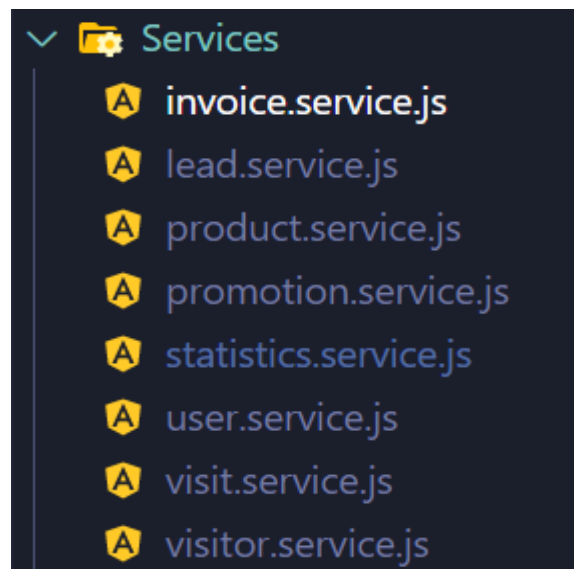


Figure 77 : Répertoire Services

### 5.1.13. Répertoire "Tests"

Ce répertoire est utilisé pour stocker les fichiers de tests de l'application. Les tests sont écrits à l'aide du framework Jest afin de vérifier le bon fonctionnement des différentes parties de l'application.

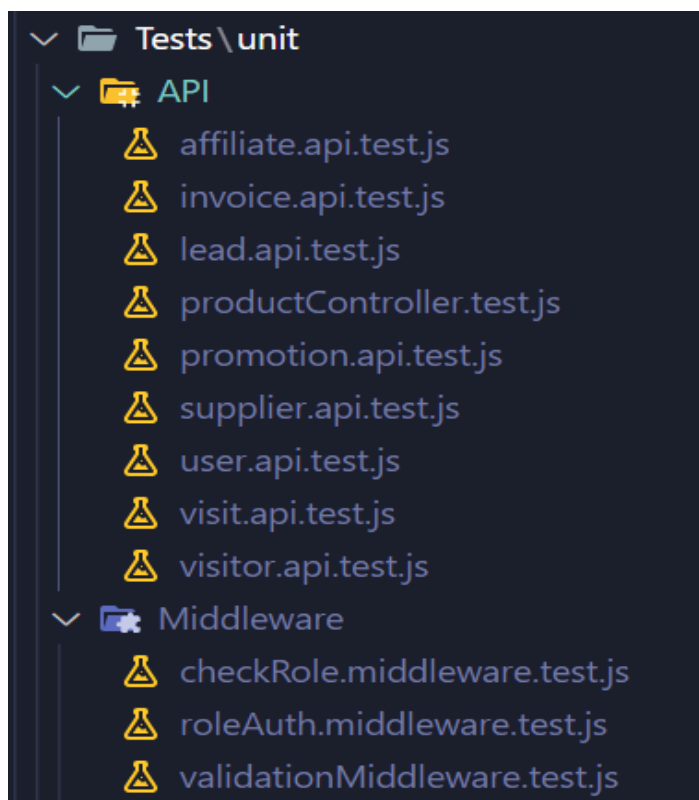


Figure 79 : Tests API et Middleware



Figure 78 : Tests Service et validateurs

### 5.1.14. Répertoire "Utilities"

Dans ce répertoire, nous plaçons les fichiers utilitaires, des constantes et les fonctions réutilisables utilisées dans plusieurs parties de l'application.



Figure 80 : Répertoire Utilities



### 5.1.15. Répertoire “Validators”

Ce répertoire contient des fichiers qui incluent des fonctions de validation personnalisées pour différents types de données ou champs, permettant par exemple de vérifier la validité d'une adresse e-mail, la complexité d'un mot de passe, la conformité d'un numéro de téléphone, etc.

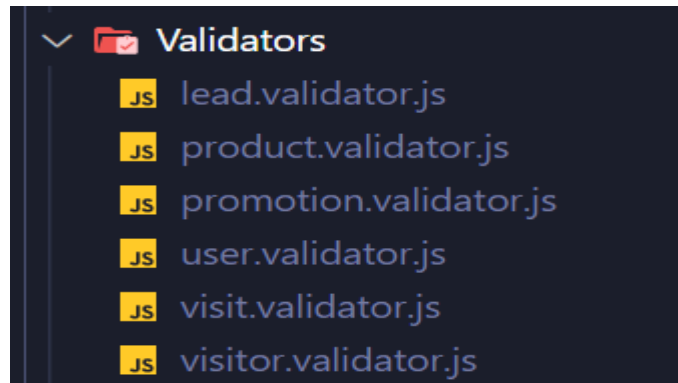


Figure 81 : Répertoire Validators

## 5.2. Sécurité

### 5.2.1. Structure d'un JWT

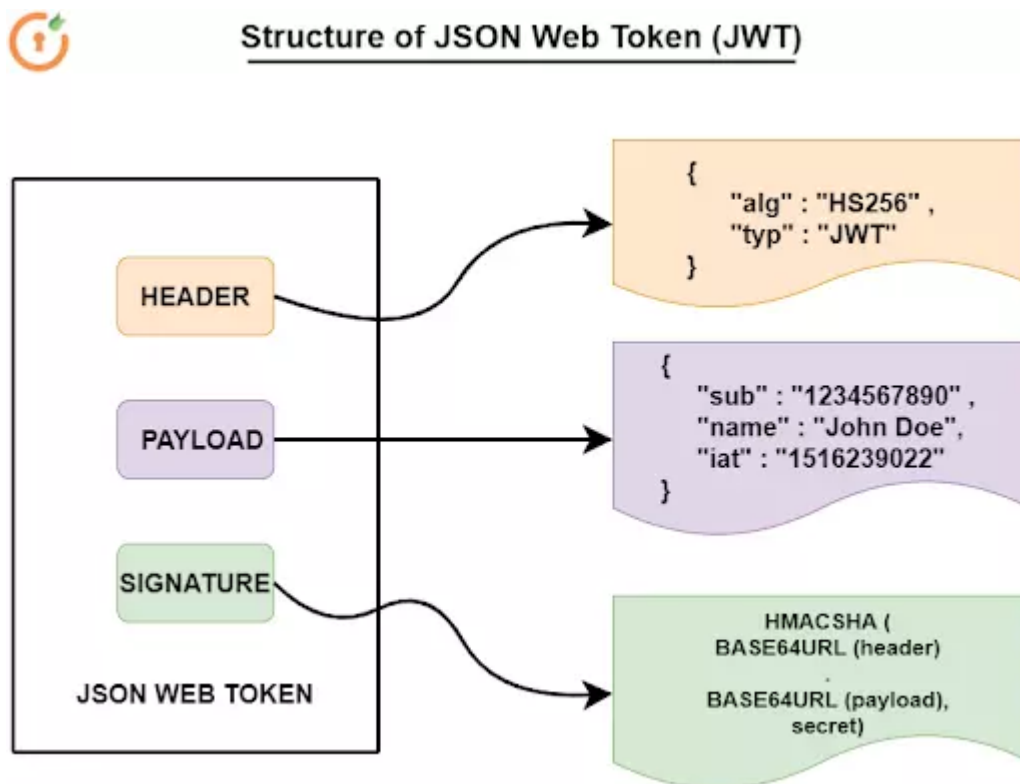


Figure 82 : Structure d'un JWT

Un JSON Web Token (JWT) est une chaîne de caractères qui se compose de trois parties distinctes : l'en-tête (header), la charge utile (payload) et la signature.

L'en-tête contient des métadonnées sur le type de token utilisé et l'algorithme de signature associé au JWT. Ces informations permettent au serveur de savoir comment décoder et vérifier le JWT.

La charge utile (payload) contient des données supplémentaires que l'on souhaite transporter dans le JWT, telles que l'identité de l'utilisateur, les autorisations ou toute autre information pertinente.

La signature est calculée en utilisant la clé secrète du serveur. Elle est générée en prenant l'en-tête encodé en Base64, la charge utile encodée en Base64, puis en les signant avec l'algorithme spécifié dans l'en-tête. La signature est ensuite ajoutée au JWT pour garantir son intégrité. (JWT - Jwt.io, n.d.).

## 5.2.2. Processus d'autorisation

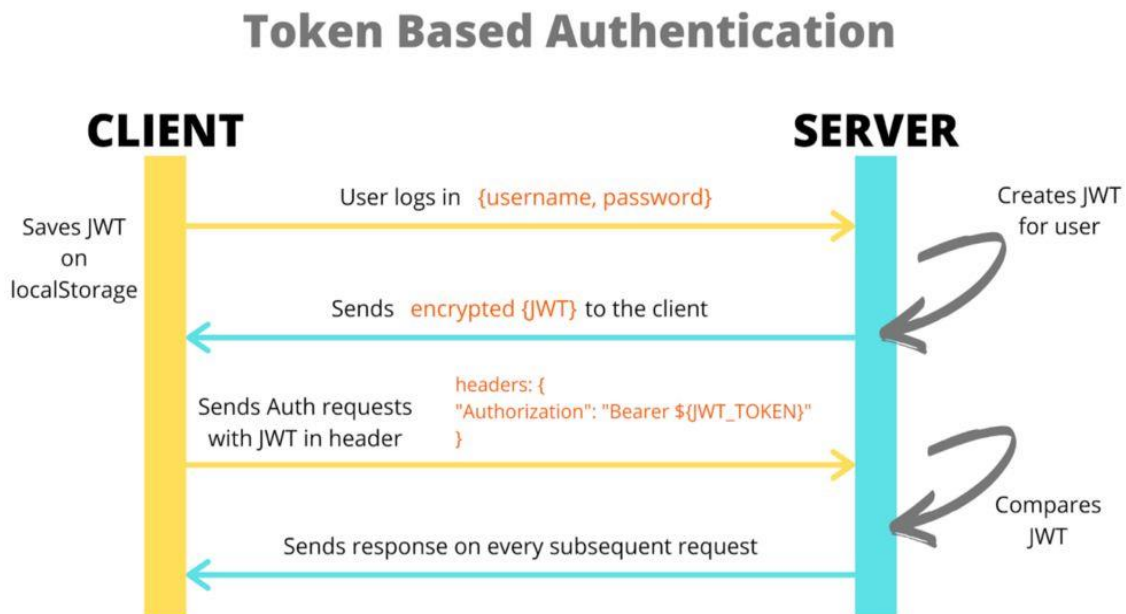


Figure 83 : Processus d'authentification

L'utilisateur envoie ses informations de connexion, telles que son nom d'utilisateur et son mot de passe, à un serveur d'authentification qui vérifie les informations de connexion et génère un JWT.

Le JWT contient des données spécifiques à l'utilisateur, telles que son identifiant et ses autorisations. Il est signé avec une clé secrète connue uniquement par le serveur, puis ce dernier renvoie le JWT à l'utilisateur.

L'utilisateur stocke le JWT localement, généralement dans un cookie ou dans le stockage local du navigateur.

Lorsque l'utilisateur souhaite accéder à une ressource protégée, il envoie une requête au serveur contenant le JWT. Cela peut être fait en incluant le JWT dans l'en-tête d'autorisation de la requête.

Le serveur vérifie la validité du JWT. S'il est valide, le serveur extrait les informations du JWT pour identifier l'utilisateur et vérifier les autorisations.

Si le JWT est valide et que l'utilisateur dispose des autorisations requises, le serveur autorise l'accès à la ressource demandée. Sinon, il renvoie une erreur d'autorisation.

## 5.3. Front-end

### 5.3.1. Login

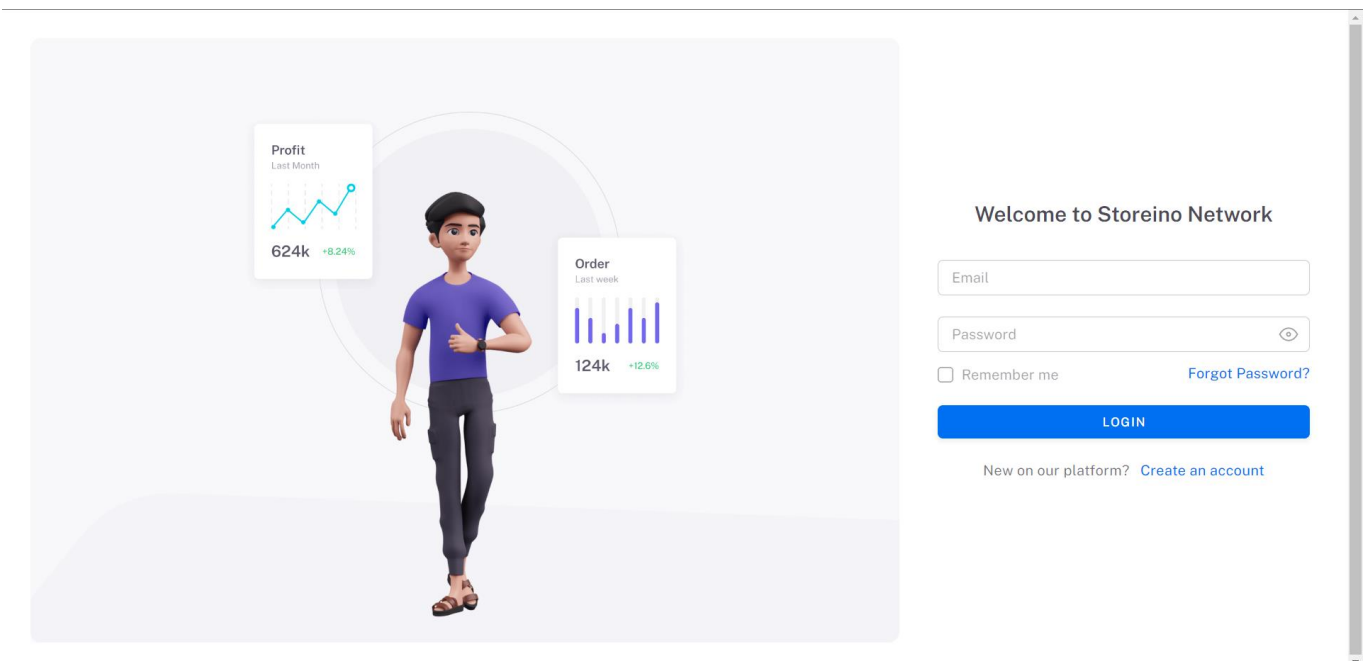


Figure 84 : Page du login

### 5.3.2. Coté affiliate

#### 5.3.2.1. Dashboard

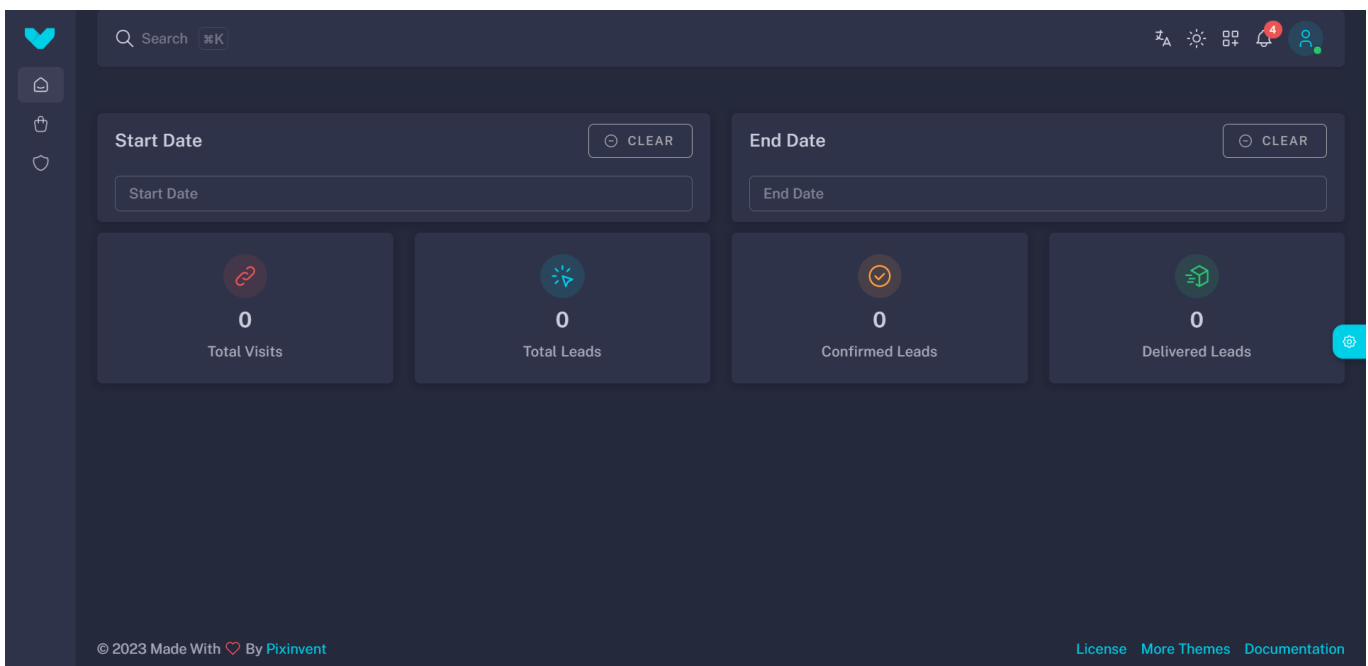


Figure 85 : Dashboard d'affilié

### 5.3.2.2. Mes promotions

Start Date  CLEAR End Date  CLEAR

Start Date  End Date

10  Search  Select Status

PRODUCT	COMMISSION	CATEGORY	COUNTRY	STATUS	ACTIONS
Phone	200	Electronics & Accessories	Morocco	Approved	
Headphones	50	Electronics & Accessories	Morocco	Pending	

Showing 1 to 2 of 2 entries

© 2023 Made With ❤ By Pixinvent License More Themes Documentation

Figure 86 : Page des promotions d'affilié

### 5.3.2.3. Liste des offres

Start Date  CLEAR End Date  CLEAR

Start Date  End Date

12  Select Category  Select Country

**Phone**  
2000 MAD  
Electronics & Accessories  
Electronics Phone Apple  
Commission: 200 MAD  
Country: Morocco  
REQUEST

**Headphones**  
200 MAD  
Electronics & Accessories  
Electronics Music Beats  
Commission: 50 MAD  
Country: Morocco  
REQUEST

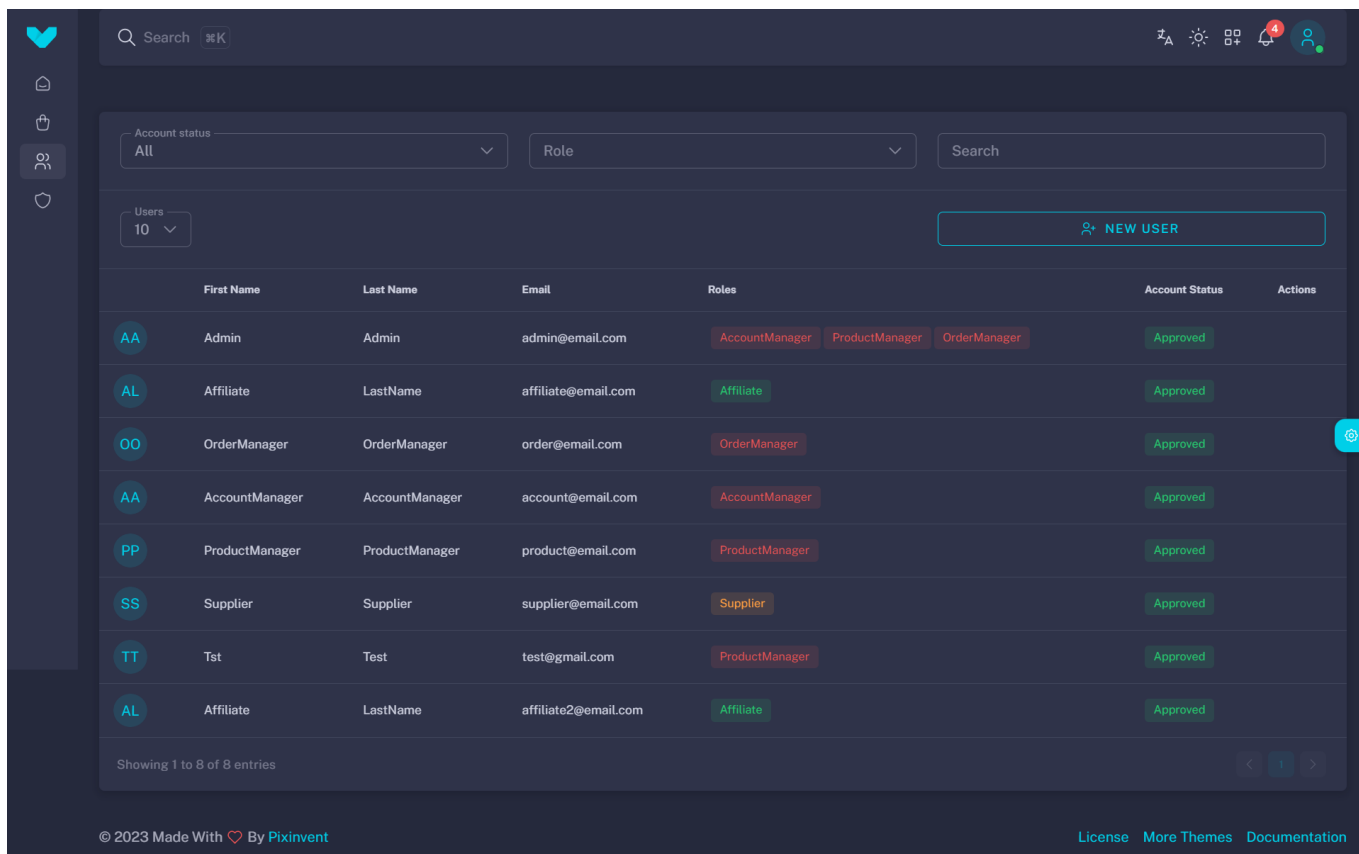
**Camera**  
200 MAD  
Electronics & Accessories  
Electronics Camera  
Commission: 50 MAD  
Country: Morocco  
REQUEST

Showing 1 to 3 of 3 entries

Figure 87 : Page des offres

## 5.3.3. Gestionnaire des comptes

### 5.3.3.1. Liste des utilisateurs

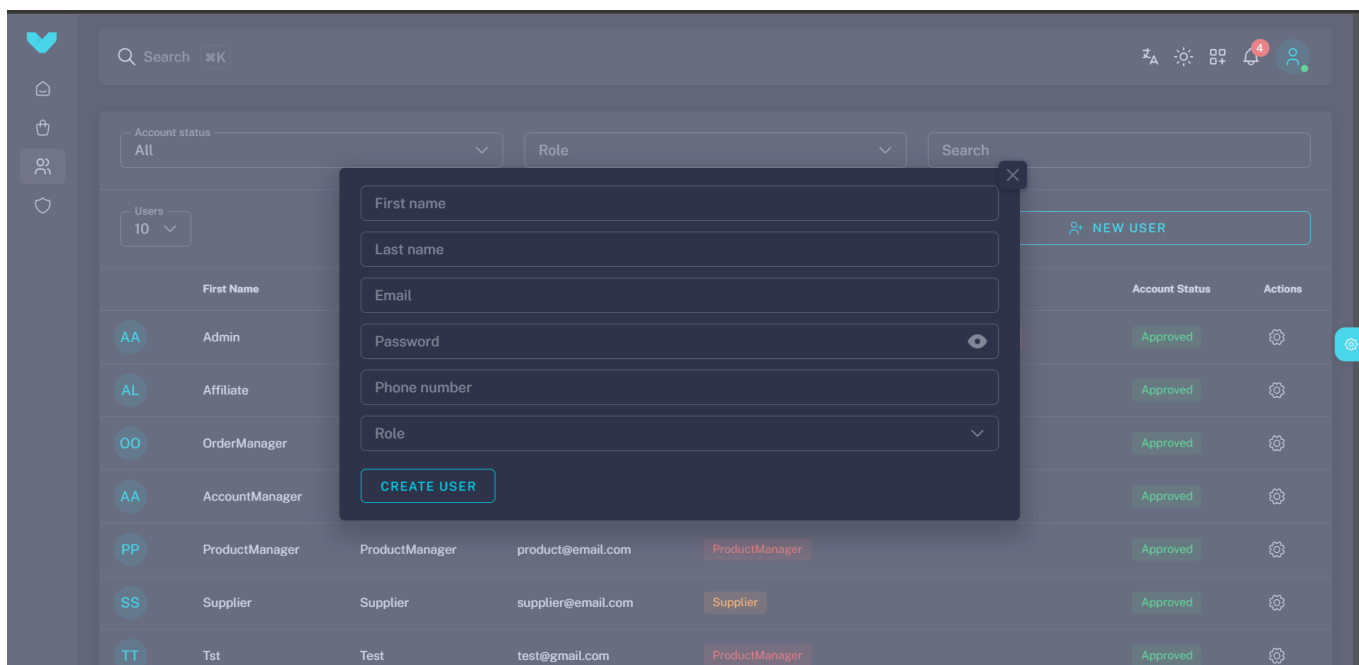


The screenshot shows a user management interface with a dark theme. At the top, there is a search bar and a sidebar with navigation icons. Below the search bar, there are filters for 'Account status' (set to 'All') and 'Role'. A 'Search' button is also present. A 'Users' dropdown shows '10' users. A 'NEW USER' button is located on the right. The main area contains a table with columns: First Name, Last Name, Email, Roles, Account Status, and Actions. The table lists 8 users with various roles like Admin, Affiliate, OrderManager, AccountManager, ProductManager, and Supplier. Each user has a corresponding role label and an 'Approved' status. A 'Showing 1 to 8 of 8 entries' message is at the bottom of the table.

	First Name	Last Name	Email	Roles	Account Status	Actions
AA	Admin	Admin	admin@email.com	AccountManager, ProductManager, OrderManager	Approved	
AL	Affiliate	LastName	affiliate@email.com	Affiliate	Approved	
OO	OrderManager	OrderManager	order@email.com	OrderManager	Approved	
AA	AccountManager	AccountManager	account@email.com	AccountManager	Approved	
PP	ProductManager	ProductManager	product@email.com	ProductManager	Approved	
SS	Supplier	Supplier	supplier@email.com	Supplier	Approved	
TT	Tst	Test	test@gmail.com	ProductManager	Approved	
AL	Affiliate	LastName	affiliate2@email.com	Affiliate	Approved	

Figure 88 : Page de la liste des utilisateurs

### 5.3.3.2. Créer un administrateur des comptes, produits ou des commandes



The screenshot shows the same user management interface as Figure 88, but with a 'CREATE USER' modal form overlay. The form contains fields for: First name, Last name, Email, Password (with a toggle for visibility), Phone number, and Role (a dropdown menu). A 'CREATE USER' button is at the bottom of the form. The background table is partially visible behind the modal.

Figure 89 : Formulaire de création d'un compte

### 5.3.3.3. Visualiser les détails d'un utilisateur

Search #K

AL

DETAILS

Name: Affiliate LastName

Email: affiliate@email.com

Status: Approved

Roles: Affiliate

Country: Morocco

EDIT SUSPEND

OVERVIEW SECURITY

User's promotions overview

PRODUCT	CATEGORY	COUNTRY	STATUS
	Electronics & Accessories	Morocco	<span>Approved</span>
	Electronics & Accessories	Morocco	<span>Pending</span>

© 2023 Made With By Pixinvent

[License](#) [More Themes](#) [Documentation](#)

Figure 90 : Page des détails d'un utilisateur

### 5.3.3.4. Liste des demandes de création de compte

Search #K

10

Search by email Role

First Name	Last Name	Email	Role	Account Status	Actions	
AL	Affiliate	LastName	affiliate2@email.com	<span>Affiliate</span>	<span>Pending</span>	

Showing 1 to 1 of 1 entries

© 2023 Made With By Pixinvent

[License](#) [More Themes](#) [Documentation](#)

Figure 91 : Page de la liste des demandes de création de compte

5.3.3.5. Gérer une demande de création de compte

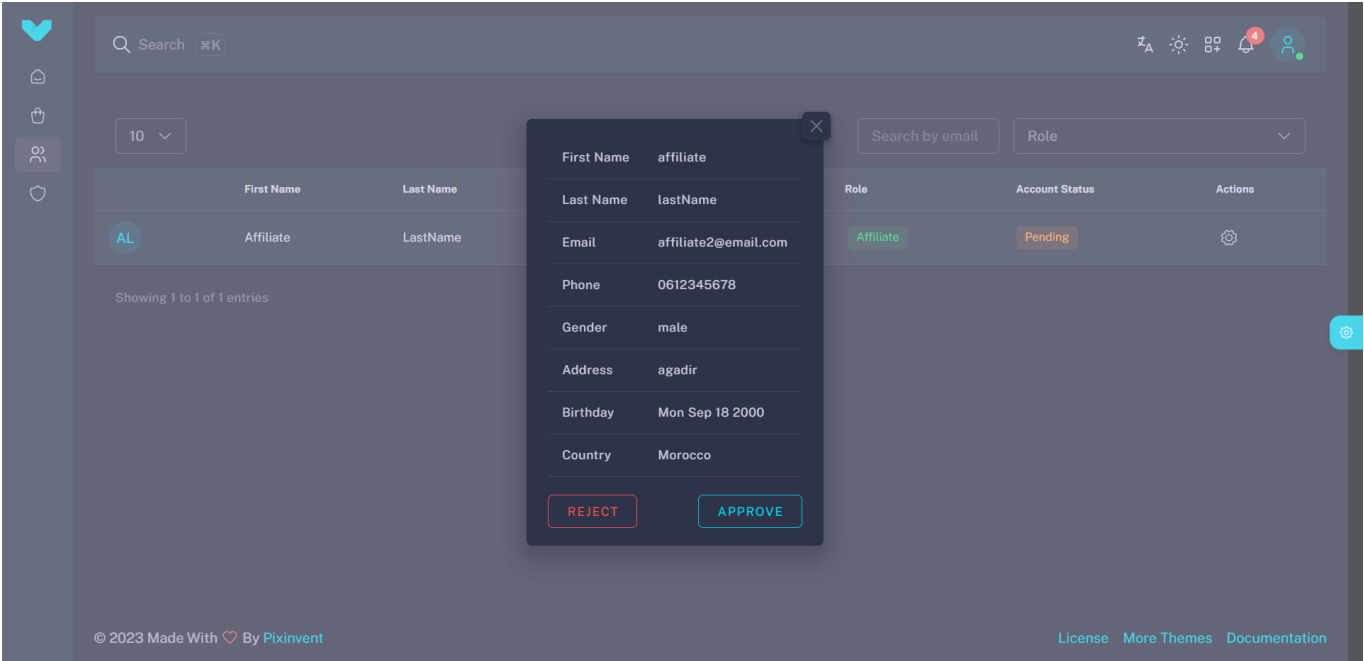


Figure 92 : Formulaire de gestion d'une demande de création d'un compte

5.3.4. Exceptions

5.3.4.1. Non autorisée

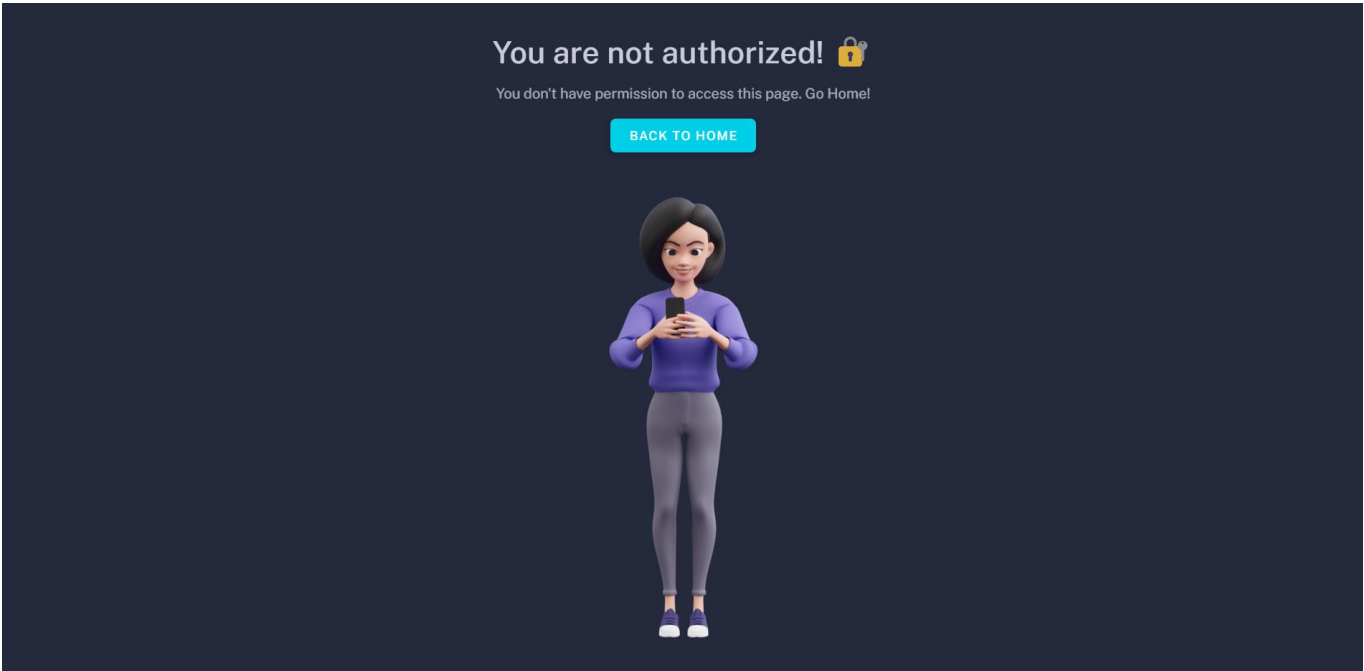


Figure 93 : Page d'une ressource non autorisée



### 5.3.4.2. Page introuvable

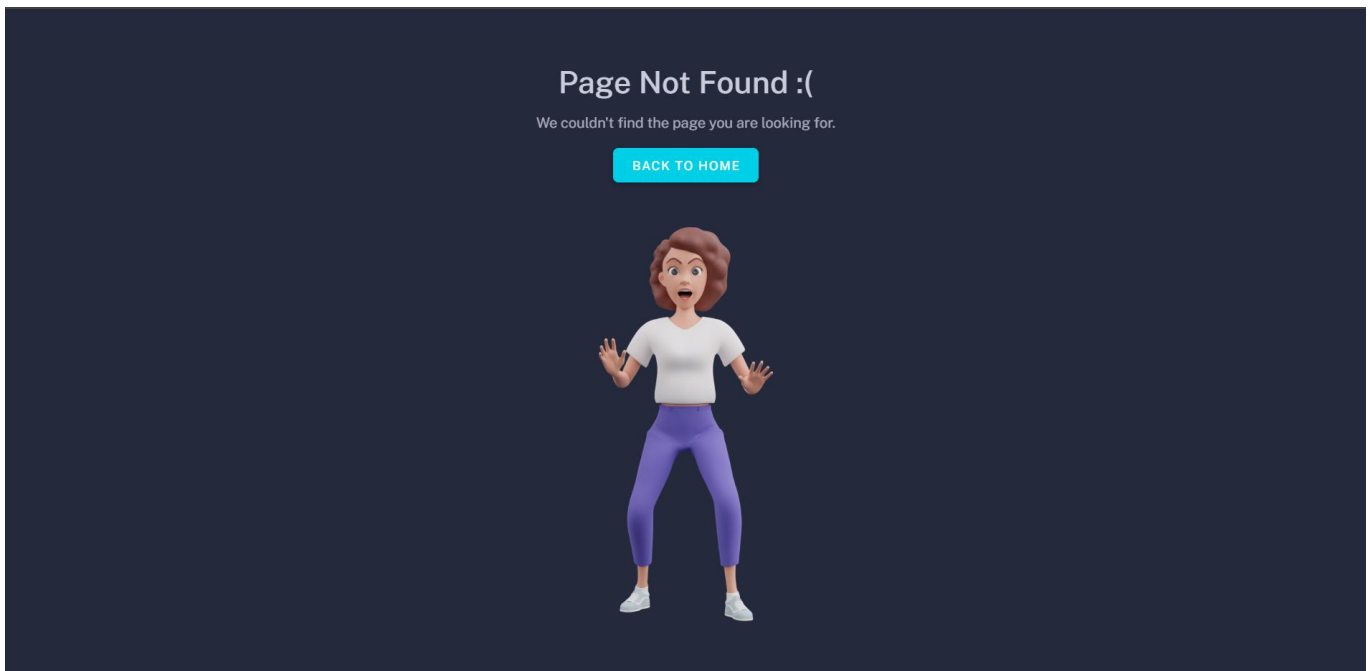


Figure 94 : Page d'une ressource introuvable

### 5.3.4.3. Erreurs de validation

"first\_name" is not allowed to be empty

First name

Last name

Ait Brik

Email

error

Password

.....

Phone number

06123456789

Role

Product Manager

CREATE USER

Figure 95 : Gestion des erreurs de validation

## 5.4. Conclusion du chapitre

En conclusion, ce chapitre a exposé la structure globale du projet, tant du point de vue du backend que du frontend. Nous avons également présenté les interfaces graphiques qui illustrent le fonctionnement de notre application. Ces informations clés nous permettent de visualiser l'architecture du projet dans son ensemble, en mettant en évidence la cohérence entre les différentes composantes. En combinant une solide structure backend avec des interfaces utilisateur conviviales, notre application offre une expérience utilisateur fluide et intuitive.

## 6. Conclusion

Pour la mise en œuvre de ce projet, nous avons tout d'abord procédé à une étude conceptuelle approfondie afin d'identifier les différents modules nécessaires à cette application, ainsi qu'à une évaluation des outils et technologies pouvant être adaptés à sa réalisation. Ensuite, nous avons réalisé une analyse et une conception du projet en utilisant le formalisme UML. Nous avons élaboré plusieurs diagrammes pour mieux structurer le projet, ce qui a grandement facilité sa mise en œuvre. Enfin, nous avons implémenté les différents modules de l'application. Les résultats obtenus lors de cette dernière phase répondent aux exigences et aux besoins du cahier des charges.

Durant ce stage, nous avons eu l'occasion de nous familiariser avec le monde professionnel et de mettre en pratique nos connaissances, ce qui nous a permis d'acquérir de nouvelles compétences et de maîtriser de nouveaux outils. Travailler sur ce projet nous a offert une vision détaillée de la gestion et de la réalisation de projets.

Grâce à ces différentes étapes, nous avons pu développer une application qu'on espère répondre aux attentes du cahier des charges. Cette expérience nous a également permis d'approfondir nos compétences en matière de gestion de projet et d'élargir notre expertise dans l'utilisation d'outils technologiques spécifiques. Nous sommes convaincus que les connaissances acquises lors de ce projet seront d'une grande valeur pour nos futures entreprises et projets. Nous sommes fiers du travail accompli et des résultats obtenus grâce à notre engagement et à notre dévouement tout au long de ce processus.

# Webographie et Bibliographie

- About / Git.* (n.d.). Git SCM. Retrieved 2023, from <https://git-scm.com/about/branching-and-merging>
- About / npm.* (n.d.). npm Docs. Retrieved 2023, from <https://docs.npmjs.com/about-npm>
- Axios.* (n.d.). Axios. Retrieved 2023, from <https://axios-http.com/fr/docs/intro>
- bcrypt.* (2022, October 6). npm. Retrieved 2023, from <https://www.npmjs.com/package/bcrypt>
- CASL.* (n.d.). CASL. Retrieved 2023, from <https://casl.js.org/v6/en/guide/intro>
- CSS / MDN.* (2023, April 15). MDN Web Docs. Retrieved 2023, from <https://developer.mozilla.org/en-US/docs/Web/CSS>
- Documentation Sass.* (n.d.). Sass. Retrieved 2023, from <https://sass-lang.com/documentation/>
- ExpressJs.* (n.d.). Express - Node.js web application framework. Retrieved 2023, from <https://expressjs.com/>
- HTML / MDN.* (2023, May 9). MDN Web Docs. Retrieved 2023, from <https://developer.mozilla.org/en-US/docs/Web/HTML>
- Iconify Documentation.* (2023, May 17). Iconify. Retrieved 2023, from <https://iconify.design/docs/>
- JavaScript / MDN.* (2023, May 1). MDN Web Docs. Retrieved 2023, from <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- Jest.* (n.d.). Jest · Delightful JavaScript Testing. Retrieved 2023, from <https://jestjs.io/>
- Joi.* (n.d.). joi.dev. Retrieved 2023, from <https://joi.dev/api/?v=17.9.1>
- JWT - jwt.io.* (n.d.). JWT.io. Retrieved 2023, from <https://jwt.io/introduction>
- MongoDB.* (n.d.). MongoDB. Retrieved 2023, from <https://www.mongodb.com/docs/manual/>
- MongoDB Compass.* (n.d.). MongoDB. Retrieved 2023, from <https://www.mongodb.com/products/compass>
- Mongoose.* (n.d.). Mongoose. Retrieved 2023, from <https://mongoosejs.com/docs/guide.html>
- NodeJS.* (n.d.). Node.js. Retrieved 2023, from <https://nodejs.org/en/about>

*PassportJs*. (n.d.). Passport.js. Retrieved 2023, from <https://www.passportjs.org/docs/>

*Pinia*. (n.d.). Pinia. Retrieved 2023, from <https://pinia.vuejs.org/introduction.html>

*Postman*. (n.d.). Postman API Platform. Retrieved 2023, from <https://www.postman.com/>

*Trello*. (n.d.). Trello | Atlassian Support. Retrieved 2023, from <https://support.atlassian.com/trello/resources/>

*VsCode*. (n.d.). Visual Studio Code. Retrieved 2023, from <https://code.visualstudio.com/docs>

*VueJs*. (n.d.). Vue.js. Retrieved 2023, from <https://vuejs.org/guide/introduction.html>

*Vuetify*. (n.d.). Vuetify. Retrieved 2023, from <https://vuetifyjs.com/en/introduction/why-vuetify/>

*winston*. (2023, May 26). npm. Retrieved 2023, from <https://www.npmjs.com/package/winston>