

République Algérienne Démocratique et Populaire
Ministère de L'enseignement Supérieur et de la Recherche Scientifique

Université A/Mira de Béjaia

Faculté des Sciences Exactes

Département d'Informatique



Mémoire de Fin de cycle

En vue de l'obtention du diplôme Master professionnel en
Informatique

Spécialité : Administration et Sécurité des Réseaux

THÈME

Conception et réalisation d'une application
Web E-Commerce sous Java EE

Réalisé par :

M^r BENACHOUR Kouceila

M^r TOUATI Mourad

Devant le jury composé de :

Président : M^r SKLAB Youcef

Examineur : M^r FARAH Zoubeyr

Examinatrice : M^{lle} BENMERBI Samah

Promoteur : M^r ABBACHE Bournane

Co-Promoteur : M^r OUZZEGANE Redouane

PROMOTION 2014/2015

Résumé

Le commerce a depuis toujours été au cœur de l'économie, il est au centre des échanges entre individus, à ce titre il a été le facteur dénominateur de l'évolution de l'humanité, Aujourd'hui avec l'avènement du E-Commerce il continue son évolution. De nos jours, il existe plusieurs sites traitant sur le E-Commerce, malheureusement l'Algérie est très en retard dans ce domaine. Notre projet consiste à développer une application web E-Commerce afin de palier à ce problème, pour ce faire nous avons eu recours au langage de modélisation UML afin modéliser le projet, ce choix c'est porté sur le fait qu'il soit simple, détaillé, organisé et dans notre cas, le plus approprié. Pour la réalisation, nous avons eue recours à Java EE qui est une plateforme Java dont nous nous sommes servis afin d'implémenter les fonctionnalités de l'application.

Mots clés : Commerce, Économie, Application, Web, E-Commerce, Modéliser, UML, Java EE.

Abstract

Trade has always been at the heart of the economy, it is at the center of exchanges between individuals, it was the denominator factor of the evolution of humanity, today with the advent of E-business continues its evolution. Nowadays, there are several sites dealing on E-business, unfortunately Algeria is by far behind in this area. Our project is to develop an E-business web application, in order to solve this problem, to do this, we have used the UML modeling language to model the project, this choice is focused on the fact that it is simple, detailed, organized and in our case, the most appropriate. For the realization, we had recourse to Java EE which is a Java platform that we have used to implement the functionalities of the application.

Keys words : Trade, Economy, Application, Web, E-business, Modeling, UML, Java EE..

Remerciements

Nous tenons dans un premier temps à remercier le bon dieu le tout puissant qui nous a donné le courage et la volonté pour mener à bien ce modeste travail.

Nous tenons à remercier nos encadreurs M. ABBACHE Bournane et M. OUZZEGANE Redouane pour tout le temps qu'ils nous ont consacré, pour leur conseils précieux, pour toute leur aide et leur appui durant la réalisation de ce travail.

Nous tenons à remercier chacun des membres du jury pour nous avoir fait l'honneur d'examiner et d'évaluer notre travail. À tous nos enseignants et les membres du département informatique de l'université ABDERRAHMANE MIRA. Que tous ceux et celles qui ont contribué de près ou de loin à l'accomplissement de ce travail trouvent ici l'expression de nos remerciements les plus chaleureux.

Dédicaces

*À nos chers parents, pour leur persévérance,
et pour avoir suscité notre vocation
et permis d'achever nos études en tant que nous sommes actuellement,
À nos soeurs, et nos frères
pour leur soutien moral
et leur intérêt envers notre travail,
À tous nos amis et collègues,
À tous ceux qui nous ont aidés,
À tous ceux qui nous sont chers,
À tous ceux qui nous avons omis,*

nous dédions ce humble travail.

***Mourad,
Kouceila.***

Table des matières

Remerciements	2
Introduction Générale	2
1 Généralités	4
1.1 Commerce Electronique	5
1.1.1 Définition du E-Commerce	5
1.1.2 Commerce traditionnel vs E-Commerce	5
1.1.3 Les différents types de E-Commerce	7
1.1.4 Commerce électronique entre particuliers(C2C)	8
1.1.5 Les modes de paiement en ligne	9
1.1.6 Problématique	10
1.1.7 Objectifs du projet	11
1.2 Méthode de conception et démarche de développement	11
1.2.1 La méthode de conception	11
1.2.2 Définition UML (Unified Modeling Language)	11
1.2.3 Processus de développement	12
1.2.4 Démarche de développement	14
1.2.5 Le processus 2TUP (Two Track Unified Process)	14

2	Capture et analyse des besoins	17
2.1	Etude préliminaire	17
2.1.1	Description du projet	17
2.1.2	Identification des acteurs	18
2.2	Capture des besoins	22
2.2.1	Capture des besoins fonctionnels	22
2.2.2	Capture des besoins techniques	35
2.2.3	L'architecture d'application[11]	35
2.2.4	Le but d l'architecture en 3-tiere	36
3	Analyse et conception	37
3.1	Diagramme de séquence	37
3.1.1	Etude de cas d'utilisation	38
3.1.2	- Diagramme de séquence du cas d'utilisation « Attribuer une note »	50
3.2	Diagramme d'activité	50
3.2.1	Définition	50
3.2.2	Représentation des diagrammes d'activité	51
3.3	Diagramme de classe	53
3.4	Dictionnaire de données	54
3.5	Model relationnel	55
4	Réalisation	57
4.1	Description de l'environnement de développement	57
4.1.1	Java entreprise édition	57
4.1.2	Le serveur Apache-Tomcat	60
4.1.3	MySQL (My Structured Query Language)	60
4.1.4	HTML 5	61
4.1.5	CSS 3	61
4.1.6	JavaScript	61

4.1.7	Bootstrap	61
4.1.8	Sandbox	62
4.1.9	L'IDE Eclipse	62
4.2	Organigramme de l'application	63
4.3	Description de l'application	64
4.3.1	L'index (page d'accueil)	64
4.3.2	Produit	65
4.3.3	La boutique	65
4.3.4	Le panier	67
4.3.5	Le profil	67
4.3.6	PayPal	68
4.3.7	L'administrateur	69
	Conclusion Générale	71
	A Annexe A	75
A.1	Explication des diagrammes UML utilisés	75
A.1.1	Diagramme de cas d'utilisation	75
A.1.2	Diagramme de séquence	76
A.1.3	Diagramme d'activité	77
A.2	Modèle relationnel	78
A.2.1	Terminologie de l'approche relationnelle	78
A.2.2	Règles de dérivation du modèle relationnel à partir d'un modèle de classes	78

Table des figures

1.1	Commerce électronique entre particuliers(C2C)	8
1.2	Principe de fonctionnement d'un tiers de confiance	9
1.3	Principe de fonctionnement de PayPal	10
1.4	Le processus 2TUP	16
2.1	Diagramme de contexte	21
2.2	Le diagramme de cas d'utilisation associé à l'internaute	31
2.3	Le diagramme de cas d'utilisation associé au Membre	32
2.4	Le diagramme de cas d'utilisation associé à l'Administrateur	33
2.5	Le diagramme de cas d'utilisation global	34
2.6	L'architecture 3-tiers	35
3.1	Diagramme de séquence du cas d'utilisation « Effectuer une recherche »	38
3.2	Diagramme de séquence du cas d'utilisation « S'authentifier»	39
3.3	Diagramme de séquence du cas d'utilisation « Modifier un compte »	40
3.4	Diagramme de séquence du cas d'utilisation « Supprimer un compte »	41
3.5	Diagramme de séquence du cas d'utilisation « Gestion de produit »	42
3.6	Diagramme de séquence du cas d'utilisation «Supprimer un produit »	43
3.7	Diagramme de séquence du cas d'utilisation « Gestion de boutique »	44
3.8	Diagramme de séquence du cas d'utilisation « Supprimer une boutique »	45
3.9	Diagramme de séquence du cas d'utilisation « Gestion de panier »	46

3.10	Diagramme de séquence du cas d'utilisation « Gestion de commentaires »	47
3.11	Diagramme de séquence du cas d'utilisation «Supprimer un commentaire»	48
3.12	Diagramme de séquence du cas d'utilisation « Gestion de commande » . . .	49
3.13	Diagramme de séquence du cas d'utilisation « Attribuer une note »	50
3.14	Diagramme d'activité «Cas de la boutique»	51
3.15	Diagramme d'activité «Cas commande»	52
3.16	Diagramme de classe	53
4.1	Le design pattern MVC)	59
4.2	Le design pattern DAO)	60
4.3	Organigramme de l'application	63
4.4	Page d'accueil	64
4.5	Page Produit	65
4.6	Page boutique	66
4.7	Page panier	67
4.8	Page profil	68
4.9	Pages PayPal	69
4.10	Page administrateur	70

Liste des tableaux

1.1	<i>Commerce traditionnel vs E-Commerce</i>	5
2.1	Description des acteurs	18
2.2	<i>Identification des messages échangés</i>	20
2.3	Identification des cas d'utilisation	23
2.4	<i>Description textuelle du cas d'utilisation « Rechercher »</i>	24
2.5	<i>Description textuelle du cas d'utilisation « Modifier un compte »</i>	24
2.6	<i>Description textuelle du cas d'utilisation « S'authentifier »</i>	25
2.7	<i>Description textuelle du cas d'utilisation « Supprimer un compte »</i>	25
2.8	<i>Description textuelle du cas d'utilisation « Supprimer un produit »</i>	26
2.9	<i>Description textuelle du cas d'utilisation « Supprimer un commentaire »</i>	26
2.10	<i>Description textuelle du cas d'utilisation « Supprimer une boutique »</i>	27
2.11	<i>Description textuelle du cas d'utilisation « Gestion de produit »</i>	27
2.12	<i>Description textuelle du cas d'utilisation « Gestion de commentaires »</i>	28
2.13	<i>Description textuelle du cas d'utilisation « Gestion du panier »</i>	28
2.14	<i>Description textuelle du cas d'utilisation « Gestion de boutique »</i>	29
2.15	<i>Description textuelle du cas d'utilisation « Attribuer une note »</i>	29
2.16	<i>Description textuelle du cas d'utilisation « Gestion de commande »</i>	30

Liste des Acronymes

API : **A**pplication **P**rogramming **I**nterface.
B2B : **B**usiness to **B**usiness.
B2C : **B**usiness to **C**onsumer.
B2E : **B**usiness to **E**mployee.
B2G : **B**usiness to **G**overnment.
BDD : **B**ase **D**e **B**onnées.
CSS : **C**ascading **S**tyle **S**heets.
C2C : **C**onsumer to **C**onsumer.
DAO : **D**ata **A**ccess **O**bject.
HTML : **H**ypertext **M**arkup **L**anguage.
Java EE : **J**ava **E**nterprise **E**dition.
Java SE : **J**ava **S**tandard **E**dition.
JDBC : **J**ava **D**ata**B**ase **C**onnectivity
JSP : **J**ava **S**erver **P**ages.
MVC : **M**odèle **V**ue **C**ontrôleur.
MySQL : **M**y **S**tructured **Q**uery **L**anguage.
SGBD : **S**ystème de **G**estion de **B**ase de **D**onnées.
SQL : **S**tructured **Q**uery **L**anguage.
UML : **U**nified **M**odeling **L**anguage.
XML : **e**Xtensible **M**arkup **L**anguage.
XP : **e**Xtreme **P**rogramming.
2TUP : **T**wo **T**rack **U**nified **P**rocess.

Introduction Générale

Dans le monde des entreprises et des commerciales, la gestion de la commercialisation des produits reste un sujet sensible en vue de la difficulté qu'ils rencontrent dans la diffusion de l'information. Ces derniers nourrissent l'ambition d'avoir un plan marketing optimal avec un support de communication couvrant une région toujours plus vaste, c'est à cet effet que fut créé le commerce électronique qui contrairement à son homologue traditionnel ne fait pas les frais des contraintes géographiques, il permet alors un échange de biens et de service sans frontières atteignant des échelles nationale ou internationale pour les organismes importants.

Le concept de commerce électronique étant devenu plus important, il n'a cessé de se développer au fil du temps, des outils, des règles, des environnements de développement étant mis en place pour la bonne marche de son évolution.

Malheureusement cette notion de E-Commerce est quasi inexistante en Algérie dont l'informatisation étant très en retard, c'est pourquoi nous avons décidé de consacrer notre projet à la réalisation d'une application E-Commerce accessible pour tous afin de pallier à ce défaut. Cette application est destinée à tous genres d'utilisateur et se base sur le commerce électronique entre particulier.

Afin de mener à bien ce projet, nous avons décidé de le diviser en quatre différents chapitres :

- Le premier chapitre est consacré exclusivement à la présentation du commerce électronique avec la définition de la méthode de conception et la démarche de développement suivie.

- Le deuxième chapitre est consacré à la capture des besoins (fonctionnels et techniques) avec l'identification de nos acteurs et cas d'utilisations relatifs à ces acteurs.
- Le troisième chapitre intitulé analyse et conception représente la schématisation des cas d'utilisations sous forme de diagrammes de séquence et d'activité ainsi que la représentation objet du projet dans le diagramme de classe.
- Le quatrième et dernier chapitre est consacré à la présentation de l'application, avec tous les outils et environnement de développement qui ont servi à sa réalisation.
- Pour finir avec une conclusion et quelques perspectives.

1

Généralités

Introduction

Le commerce est défini comme l'échange de biens et de services entre individus, il représente le cœur de l'économie, au fil des âges il s'est développé au même coup que l'économie, revêtant des formes diverses et variées en offrant toujours plus de perspectives. De nos jours et avec l'apogée des nouvelles technologies, le commerce revête une forme quasi optimum, en offrant une opportunité d'échanges sans frontières, c'est l'avènement du commerce électronique.

Dans le chapitre suivant, nous définissons le commerce électronique avec ses différentes formes, ses avantages et ses inconvénients, nous spécifions également la démarche suivie pour la réalisation de notre projet.

1.1 Commerce Electronique

1.1.1 Définition du E-Commerce

Le e-commerce ou commerce électronique regroupe l'ensemble des transactions commerciales s'opérant à distance par le biais d'interfaces électroniques et digitales.

Le e-commerce englobe essentiellement les transactions commerciales s'effectuant sur Internet à partir des différents types de terminaux (ordinateurs, tablettes, smartphones, consoles, TV connectées) mais également celles réalisées à partir d'applications spécifiques (applications mobiles).

Le e-commerce est devenu le principal canal de la vente à distance ce qui explique le remplacement du terme de vente par correspondance par celui de vente à distance[1].

1.1.2 Commerce traditionnel vs E-Commerce

Le tableau suivant illustre les différences entre le commerce traditionnelle et le commerce électronique.[2]

Commerce traditionnel	E-Commerce
La forte dépendance à l'échange d'informations d'une personne à une autre.	Le partage de l'information est rendue facile par les canaux de communication électronique.
Les communications / transactions sont effectuées de manière synchrone. Une intervention manuelle est nécessaire pour chaque communication ou transaction.	La communication / transaction peut être effectuée de manière asynchrone. Le système électronique les gère automatiquement.
Il est difficile d'établir et de maintenir des pratiques standards dans le commerce traditionnel.	Une stratégie uniforme peut être facilement établie et maintenue dans le commerce électronique.
les compétences individuelles jouent un rôle très important dans les affaires.	En e-commerce ou marché électronique, il n'y a pas d'intervention humaine.
Aucune plate-forme uniforme pour le partage d'informations, car il dépend fortement de la communication entre personnes.	Le E-Commerce fournit une plate-forme universelle pour soutenir les activités commerciales à travers le monde.

TABLE 1.1 – Commerce traditionnel vs E-Commerce

A. Avantages du E-Commerce

Les avantages du E-Commerce peuvent être classés en trois grandes catégories :

- Avantages pour les entreprises
- Avantages pour les consommateurs
- Avantages pour la société

Avantages pour les entreprises

- Grace à l'utilisation du commerce électronique, les entreprises peuvent étendre leur commerce aux marchés nationaux et internationaux avec un investissement de capital minimum. Elles peuvent facilement localiser plus de clients, de meilleurs fournisseurs ainsi que les partenaires commerciaux appropriés dans le monde entier.
- Le E-Commerce aide les organisations à réduire les coûts, distribuer, récupérer et gérer les informations sur papier en numérisant ces dernières.
- Le E-commerce améliore l'image de marque de l'entreprise.
- Le E-commerce aide l'organisation à fournir de meilleurs services à la clientèle.
- Le E-Commerce permet de simplifier les processus d'affaires et les rendre plus rapides et plus efficaces.
- Le E-commerce réduit la paperasserie.[2]

Avantages pour les consommateurs

- Le client peut faire des transactions pour un produit ou une demande de renseignements sur des produits / services fournis par une société à tout moment, et à partir de n'importe quel endroit.
- Le E-Commerce fournit à l'utilisateur plus d'options pour comparer entre elles et sélectionner la moins chère et de meilleure qualité.
- Un client peut mettre des commentaires de révision sur un produit, voir ce que d'autres achètent ou voir les commentaires des autres clients sur un produit avant de faire un achat final.
- Le E-Commerce fournit l'option d'enchères virtuelles.
- Informations facilement accessibles. Un client peut voir les informations détaillées pertinentes en quelques secondes plutôt que d'attendre des jours ou des semaines.
- Le E-Commerce augmente la concurrence entre les organisations cela offre des rabais substantiels aux clients.[2]

Avantages pour la société

- Les clients n'ont pas besoin de se rendre à la boutique d'un produit, ainsi moins de trafic sur la route.
- Le E-Commerce aide à réduire le coût des produits, ainsi ils deviennent accessibles pour différents niveaux de classes social.
- Le E-commerce a permis d'élargir le champ de vente d'un produit et l'a rendu accessible à des régions où il ne l'était pas.[2]

B. Inconvénients

Les inconvénients du E-Commerce peuvent être classés en deux grandes catégories :

- Inconvénients techniques
- Inconvénients non techniques

Inconvénients techniques

- Des types particuliers de serveur web ou d'autres logiciels peuvent être exigés par le réglage de l'environnement e-commerce en dehors des serveurs de réseaux fournisseurs.
- Il pourrait y avoir des problèmes de compatibilité matérielle / logicielle
- Difficulté dans le maintien de la sécurité des transactions en ligne.[2]

Inconvénients non techniques

- **Coût initial** : Le coût de création / renforcement d'une application E-Commerce peut être très élevé. Il pourrait y avoir des retards dans le lancement en raison d'erreurs ou de manque d'expérience.
- **La résistance de l'utilisateur** : Beaucoup d'utilisateur se méfient d'acheter des produits en ligne surtout si le site est nouveau.
- Absence d'analyse physique du produit pendant l'achat.[2]

1.1.3 Les différents types de E-Commerce

Basés sur la nature de la relation vendeur-acheteur. On peut distinguer :

- L'échange électronique entre les entreprises privées et le gouvernement, souvent appelé B2G, acronyme anglais de business to government.

- Le commerce électronique entre entreprises, souvent appelé B2B, acronyme anglais de business to business.
- L'échange électronique entre une entreprise et ses employés, souvent appelé Intranet ou B2E, acronyme anglais de business to employee.
- Le commerce électronique à destination des particuliers, ou B2C, acronyme anglais de business to consumer. Il s'agit de sites web marchands.
- Le commerce électronique entre particuliers, ou C2C, acronyme anglais de consumer-to-consumer. Il s'agit de sites web permettant la vente entre particuliers.[2]

Dans le cadre de notre projet, nous nous sommes intéressés au commerce électronique entre particuliers (C2C).

1.1.4 Commerce électronique entre particuliers(C2C)

Le C2C renvoie aux systèmes destinés à supporter les échanges de particulier à particulier.



FIGURE 1.1 – Commerce électronique entre particuliers(C2C)

Dans ce type on trouve principalement 3 services :

- **Les ventes aux enchères** : L'interactivité de l'Internet a récemment conduit à mettre au point de véritables marchés électroniques reposant sur le principe des enchères. Ebay constitue l'archétype de ce type de service[3].
- **Les tiers de confiance** : Surveillent les annonces diffusées, servent d'intermédiaires pour le règlement des achats et s'assurent de la bonne exécution de la commande. Par exemple, le montant de l'achat est bloqué par la plateforme jusqu'à ce que l'acheteur ait confirmé en ligne la réception de sa commande et le bon

état de ses articles. À défaut, le site bloque le processus de paiement et intervient pour trouver une solution amiable entre l'acheteur et le vendeur[4].

- **Les petites annonces** : Ne sont que des supports de publication. À ce titre, ils n'offrent aucune protection spécifique au particulier[4].

La figure suivante représente le principe de fonctionnement d'un tiers de confiance :



FIGURE 1.2 – Principe de fonctionnement d'un tiers de confiance

1.1.5 Les modes de paiement en ligne

Il existe deux modes de paiement en ligne à savoir :

- Payer en ligne par carte bancaire
- Avec les services d'un intermédiaire (Google checkout, PayPal)[5]

Dans le cadre de notre projet nous avons eu recours au service de paypal comme intermédiaire aux transactions financières en ligne.

PayPal est sans aucun doute le site de paiements en ligne le plus populaire au monde. Les personnes disposant d'un compte Paypal peuvent l'approvisionner et se servir ensuite de l'argent pour faire leurs achats en ligne sans devoir indiquer à chaque fois leurs données de carte de crédit (bon pour des raisons de sécurité). Paypal permet donc non seulement d'envoyer des paiements (gratuitement), mais également d'en recevoir, ce qui est très utile pour ceux qui vendent des produits sur Internet (dans le cas d'un paiement reçu, Paypal prélève alors une petite commission).

En outre, Paypal est beaucoup utilisé par les entrepreneurs en ligne, car c'est une manière très simple de gérer sur son site les paiements[6].

Avantages de PayPal :

- **Sécurisé** : Effectuez vos achats sans communiquer vos coordonnées bancaires à une tierce personne.

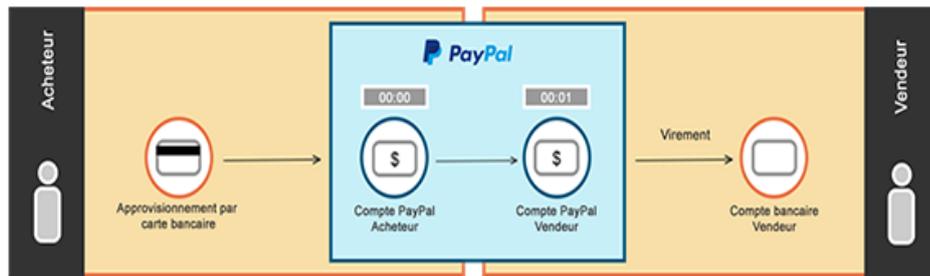


FIGURE 1.3 – Principe de fonctionnement de PayPal

- **Rapide** : Les paiements sont quasi-instantanés et bien plus rapides que l’envoi de chèques ou d’espèces
- **Avantageux** : L’envoi d’argent est gratuit et les tarifs professionnels sont simples et avantageux.
- **Pratique** : L’envoi de paiement s’effectue par une simple adresse email.
- **Mondial** : PayPal permet les paiements en seize devises et plus de 92 millions de comptes sont ouverts dans 55 pays et régions du monde.
- **Innovant** : PayPal développe sans cesse de nouveaux services et de nouveaux outils pour développer vos activités en ligne.

1.1.6 Problématique

Dans le commerce traditionnel la procédure d’achat est difficile et épuisante (se déplacer, voir tous les modèles de produits, passer d’un service à un autre pour remplir la commande, la soumettre puis effectuer le paiement en espèce ou par chèque, pour ensuite recevoir la facture et passer au magasin de stock pour la restitution de la marchandise).

De ce fait, nous avons souligné plusieurs problèmes auxquels nous tenterons de répondre dans notre projet :

- Difficulté à faire circuler l’information.
- Difficulté dans l’accès au produit.
- Offrir une interface conviviale et simple d’utilisation.
- Manque de contact entre individus.

1.1.7 Objectifs du projet

Notre travail consiste à créer une application web E-Commerce permettant l'échange de bien et de services en ligne, la conception et le développement de notre application vise à atteindre plusieurs objectifs :

- La facilité de la recherche et l'accès aux informations.
- Avoir un accès aux produits sans frontières.
- Offrir la possibilité aux membres d'acheter ou de vendre des produits en ligne.
- Offrir la possibilité aux membres d'avoir un espace de vente qui leur est propre, représenter sous forme de boutiques électroniques.
- Garantir des transactions financières sécurisées grâce au système PayPal.
- Offrir une évaluation pour les différents produits et boutiques existants.

1.2 Méthode de conception et démarche de développement

1.2.1 La méthode de conception

Modéliser, c'est décrire de manière visuelle et graphique les besoins et, les solutions fonctionnelles et techniques d'un projet.

Pour ce faire, on utilisera le langage UML2 (Unified Modeling Language) qui nous permettra de décrire l'application sous différents angles.[7]

1.2.2 Définition UML (Unified Modeling Language)

UML est un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes, qui donnent à chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour représenter le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, ... etc. Réaliser ces diagrammes revient donc à modéliser les besoins du logiciel à développer. Le langage UML fournit 13 diagrammes, qui sont classés selon deux types de vues :

- Diagrammes structurels ou diagrammes statiques (UML Structure) dans lesquels nous avons choisi d'utiliser le diagramme de classes (Class diagram).

- Diagrammes comportementaux ou diagrammes dynamiques (UML Behavior) dans lesquels nous avons choisi d'utiliser le diagramme de cas d'utilisation (Use case diagram), Diagramme de séquence (Sequence diagram) et le diagramme d'activités (Activity diagram).[7]

1.2.3 Processus de développement

Dans la gestion de projet, nous pouvons citer deux approches permettant de définir les besoins :

- La décomposition fonctionnelle (ou l'approche procédurale).
- L'approche objet (sur laquelle est basée UML).[7]

La décomposition fonctionnelle

L'approche par décomposition fonctionnelle considère que le logiciel est composé d'une hiérarchie de fonctions et de données. Les fonctions fournissent les services désirés et les données représentent les informations manipulées.

Ce découpage rend les fonctions interdépendantes, une simple mise à jour du logiciel à un point donné, peut impacter en cascade d'autres fonctions. On peut éviter cela en faisant attention à créer des fonctions très génériques. Mais respecter cette contrainte rend l'écriture du logiciel et sa maintenance plus complexe.[7]

L'approche objet

La programmation orientée objet consiste à modéliser informatiquement un ensemble d'éléments d'une partie du monde réel (que l'on appelle domaine) en un ensemble d'entités informatiques. Ces entités informatiques sont appelées objets. Il s'agit de données informatiques regroupant les principales caractéristiques des éléments du monde réel (taille, couleur, ...). Cette approche se constitue de plusieurs concepts de bases, à savoir :

- **L'objet** : L'objet est une unité atomique formée d'union d'état et d'un comportement, qui définit une représentation simplifiée d'entité du monde réel.
 - **L'état d'objet** : L'état d'objet est une situation stable dans la vie de l'objet dans laquelle l'objet attend une activation ou un événement.

- Le comportement d'objet : le comportement décrit les actions et les réactions de l'objet.
- L'identité : chaque objet a une identité unique.
- **La classe** : Une classe est une description abstraite d'un ensemble d'objets partagent les mêmes propriétés (attribut et association) et le même comportement.
- **L'association** : l'association est un concept qui permet de préciser les relations qui peuvent exister entre plusieurs objets. L'association est une abstraction de connexion symétrique entre les classes.
- **L'héritage** : L'héritage est un mécanisme de transmission des propriétés d'une classe (ses attributs et ses méthodes) vers une sous-classe. La classe dérivée est une version spécialisée de la classe base, La spécialisation et la généralisation permettent de construire des hiérarchies de classes.
 - La généralisation : Plusieurs classes peuvent être généralisées en une classe qui les factorise.
 - La spécialisation : Une classe peut être spécialisée en d'autres classes, afin d'y ajouter des caractéristiques spécifiques.
- **L'agrégation** : C'est une forme spéciale d'association de la forme « partie de » dans laquelle une classe (l'agrégat) est constituée d'un ensemble de l'autres classes.[7]

Nous avons choisi UML principalement pour trois avantages qui sont :

1. **Sa souplesse** : Une des forces d'UML est de permettre son utilisation à la carte. En effet, on peut l'utiliser aussi bien pour mettre en œuvre des développements basés sur des méthodes agiles (développement par prototypage, par exemple) que pour des processus de qualité logicielle assez complexes.
2. **Sa cohérence** : La forte utilisation d'UML dans le monde industriel vient de ce que les technologies modernes dérivent majoritairement des approches objet. La raison est double : d'une part, c'est à partir d'elle qu'UML fut construit, et, d'autre part, UML a su évoluer depuis sa création et suivre l'évolution des approches objet vers les approches composant avec l'avènement d'UML 2. UML est donc parfaitement en phase avec les technologies modernes.
3. **Ses performances** : L'utilisation d'UML rend le déroulement d'un projet plus rapide qu'avec des approches de modélisation plus lourdes grâce aux notations graphiques simples et efficaces, les outils qui le supportent ont fait de réels progrès en termes d'aide au développement (notamment à travers l'implémentation de

patrons de conception tels que les design patterns), mais aussi grâce à la génération de code ou aux modèles exécutables qui permettent de voir évoluer dynamiquement un modèle.

1.2.4 Démarche de développement

Un processus définit une séquence d'étapes, en partie ordonnées, qui concourent à l'obtention d'un système logiciel ou à l'évolution d'un système existant. L'objectif d'un processus de développement est de produire des logiciels de qualité qui répondent aux besoins de leurs utilisateurs dans des temps et des coûts prévisibles.

Actuellement, il existe plusieurs processus et approches chacun possède ses avantages et ses inconvénients. Dans le cadre de notre projet, nous avons inspiré pour le processus 2TUP qui ne nécessite pas de travailler dans une grande équipe contrairement au processus eXtreme Programming (XP) et les méthodes agiles dont le point fort réside dans le fait qu'ils valorisent l'interaction entre le client et l'équipe de développement, or dans notre projet nous n'avons pas de client à qui proposer des prototypes.[8]

1.2.5 Le processus 2TUP (Two Track Unified Process)

Le processus 2TUP est un processus unifié. Il gère la complexité technologique en donnant part à la technologie dans son processus de développement. 2TUP propose un cycle de développement qui dissocie les aspects techniques des aspects fonctionnels et propose une étude parallèle des deux branches : fonctionnelle (étude de l'application) et la technique (étude de l'implémentation).[8]

2TUP s'articule autour de trois phases :

La branche fonctionnelle

Les principales étapes de la branche fonctionnelle se présentent comme suit :

- L'étape de capture des besoins fonctionnels : Cette phase a pour objectif de définir La frontière fonctionnelle entre le système et son environnement. Les activités attendues des différents utilisateurs par rapport au système.
- L'étape d'analyse : consiste à étudier précisément les spécifications fonctionnelles de manière à obtenir une idée de ce que va réaliser le système en terme de métier.

La branche technique

La branche technique construit le squelette du système informatique et écarte la plupart des risques de niveau technique, cette branche est principalement divisée en deux étapes qui se présentent comme suit :

- La capture des besoins techniques : cette étape recense toutes les contraintes sur les choix de technologies pour la conception du système. Les outils et le matériel sélectionnés ainsi que la prise en compte des contraintes d'intégration avec l'existant conditionnent généralement des prés requis d'architecture technique.
- La conception générique : le rôle principal de cette étape est de définir les composants nécessaires à la construction de l'architecture technique. Cette conception est complètement indépendante des aspects fonctionnels.

La branche de conception réalisation

- La conception préliminaire : Cette étape permet de produire le modèle de conception par l'intégration du modèle d'analyse dans l'architecture technique.
- La conception détaillée : cette étape permet d'étudier comment réaliser chaque composant.
- L'étape de codage : permet de produire les composants du système et faire un teste au fur et à mesure sur les unités de code réalisées.
- L'étape de recette : consiste à valider les fonctionnalités du système développé.

La figure suivante illustre le processus 2TUP

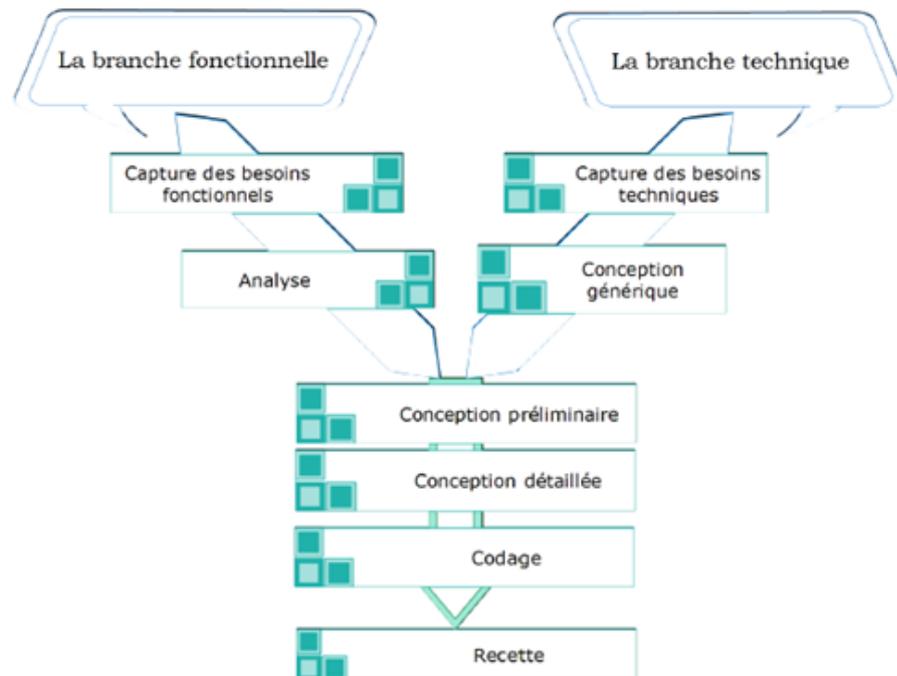


FIGURE 1.4 – Le processus 2TUP

Conclusion

Tout au long de ce travail nous avons défini le commerce électronique ainsi que ses différents concepts et approches, de ce qui a précédé nous pouvons conclure que le commerce électronique et l'une des innovations les plus importantes de ces dernières décennies, il offre une vision nouvelle du commerce, avec un accès au produit sans frontières.

Nous avons par la suite opté pour UML comme méthode de conception pour la réalisation de notre projet ainsi que la démarche de développement en 2TUP.

2

Capture et analyse des besoins

Introduction

Dans le chapitre suivant nous définissons les différents acteurs de notre projet dans la phase d'analyse de l'existant ainsi que les besoins fonctionnels où sont indiqués les différents cas d'utilisation et les besoins techniques où est définie l'architecture de l'application.

2.1 Etude préliminaire

2.1.1 Description du projet

Notre projet a pour ambition d'assurer les échanges entre particulier qui sont des membres inscrit dans l'application, ces membres ont la possibilité de vendre ou d'acheter des produits en ligne à travers un espace public, ils ont aussi la possibilité d'avoir des espaces qui leurs sont propre, ces membre peuvent aussi bien être de simple utilisateur lambda que des particuliers tel que des entreprise ou autre.

2.1.2 Identification des acteurs

Définition d'un acteur

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié, autrement dit un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données.[8]

Dans le cadre de notre étude, nous avons distingué trois acteurs qui interagissent directement avec le système ainsi qu'un quatrième acteur secondaire :

Acteur	Description
L'internaute.	L'internaute est toute personne qui visite le site et qui n'appartient pas aux membres inscrits.
Membre.	Personne qui est inscrite dans l'application et qui peut acheter ou vendre un produit.
Administrateur.	Personne qui gère et administre le site.
Paypal.	Paypal assure des transactions financières en ligne d'une manière sécurisée.

TABLE 2.1 – Description des acteurs

Description des rôles de chacun des acteurs

- **L'internaute** : C'est une personne qui peut surfer dans le site pour voir les produits qui existent et utiliser la barre de recherche.
- **Membre** : Le membre a le même privilège que l'internaute, ce à quoi s'ajoute la possibilité de poster une annonce et finaliser un achat. Il a aussi la possibilité d'avoir son propre espace où il lui est possible de regrouper l'ensemble de ses annonces sous forme d'une boutique électronique qu'il peut administrer à sa guise.
- **Administrateurs** : L'administrateur est la personne qui prend en charge la gestion et l'administration de l'application, il est en mesure d'apporter des modifications nécessaires à l'application. Pour gérer le système, l'administrateur doit suivre et maintenir plusieurs parties :
 - La partie gestion des comptes : L'administrateur est en mesure de résilier le compte d'un membre dont la fiabilité ou l'intégrité serait douteuse.

- La partie gestion des boutiques : L'administrateur est en mesure de supprimer une boutique dont le contenu serait indésirable.
 - La partie contrôle de publications : suppression des publications et des commentaires indésirables.
- **PayPal** : PayPal est un acteur secondaire sollicité par le système lorsqu'un client commande un produit. Il s'agit d'un service de paiement en ligne qui permet de payer des achats, de recevoir des paiements, ou d'envoyer et de recevoir de l'argent.

Identification des messages échangés

Un message représente la spécification d'une communication unidirectionnelle entre objets qui transporte de l'information avec l'intention de déclencher une activité chez le récepteur.[8]

Acteur	Message	Services offerts
Internaute	Demande de connexion vers une page	Assurer une connexion de l'utilisateur au serveur
	Demande d'informations (Produits, boutique)	Afficher une page dynamique qui contient des informations
	Demande d'authentification	Donner l'accès à un compte privé.
Membre	Demande d'ajouter, modifier, supprimer un produit	Sauvegarder ou supprimer le produit
	Demande d'ajouter, supprimer un produit du panier.	Sauvegarder ou supprimer le produit dans le panier
	Demande d'effectuer une commande	Redirection vers paypal
	Demande de commenter	Sauvegarder le commentaire
	Demande de créer, modifier, supprimer une boutique	Afficher un formulaire de création ou de modification ou confirmation de suppression .
Administrateur	Demande de supprimer un produit	Demande de confirmation, après supprimer le produit
	Demande de supprimer une boutique	Demande de confirmation, après supprimer la boutique
	Demande de supprimer un compte	Demande de confirmation, après supprimer le compte
	Demande de supprimer un commentaire	Demande de confirmation, après supprimer le commentaire.

TABLE 2.2 – Identification des messages échangés

Diagramme de contexte

Le diagramme de contexte d'un système est l'ensemble des objets échangés (produits, services et/ou informations) entre un système organisationnel (la boîte noire) et des acteurs de l'environnement externe du système à l'étude.

Ce diagramme nous permet de représenter et visualiser les composants de l'environnement de notre système ainsi que les différents acteurs qui interagissent avec.[9]

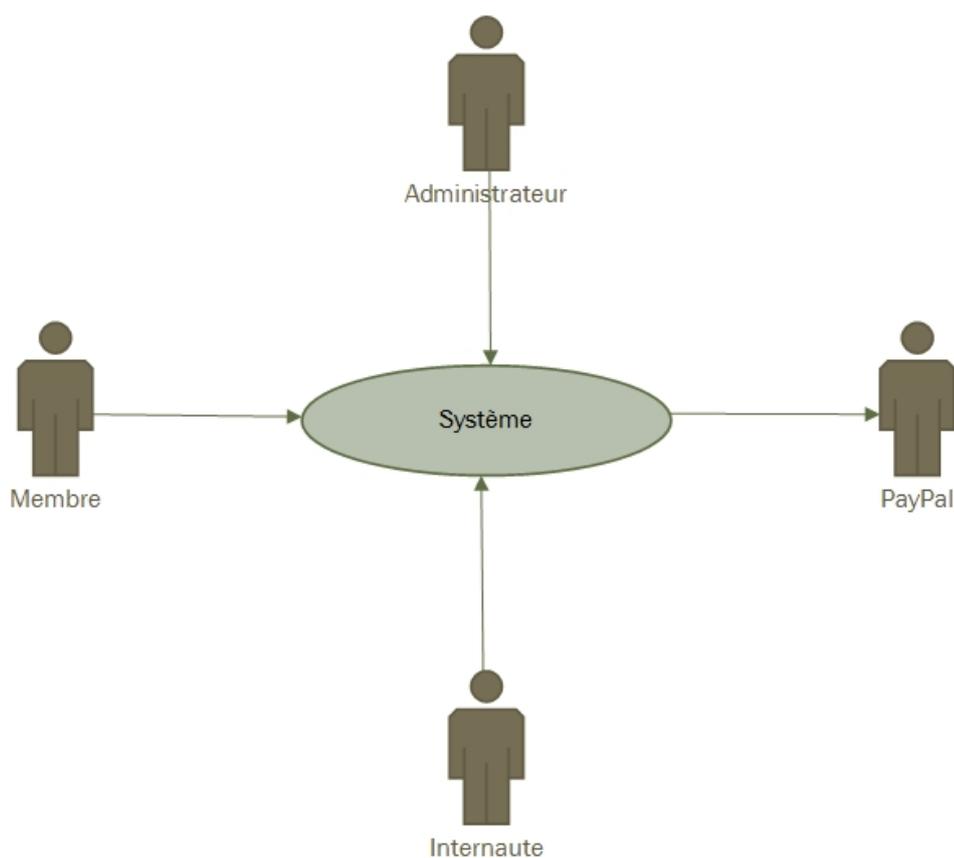


FIGURE 2.1 – Diagramme de contexte

2.2 Capture des besoins

2.2.1 Capture des besoins fonctionnels

Pour pouvoir spécifier les besoins fonctionnels de notre étude, le développement de l'application est décomposé en parties, chaque partie est constituée de plusieurs fonctionnalités, Cette répartition a pour but de simplifier la construction et la compréhension de l'application.

Dans la partie capture des besoins fonctionnels, nous avons défini une démarche à suivre qui se présente comme suit :

- La division du projet en cinq unités principales :
 - Gestion d'un produit
 - Gestion de commentaires
 - Gestion de panier
 - Gestion de commande.
 - Gestion de boutique.

Chaque unité est représentée par un ensemble de fonctionnalités.

- Définition des cas d'utilisation primaires suivants :
 - Recherche simple ou avancée sur le site.
 - Authentification.
 - Créer un compte.
 - Modifier un compte
 - Supprimer un compte
 - Attribuer une note.
 - Supprimer une boutique.
 - Supprimer un commentaire.

Identification des cas d'utilisations :

N	Cas d'utilisations		Acteur
1	- Rechercher	-Simple	-Internaute-Membre
		-Avancée	-Administrateur
2	- S'authentifier		-Membre-Administrateur
3	- Supprimer un compte		
4	- Supprimer un produit		
5	- Supprimer un commentaire		
6	- Supprimer une boutique		
11	-Gestion de commande	Ajouter	-Membre-PayPal
		Annuler	
7	- Modifier un compte		-Membre
8	-Gestion de produit	Ajouter	
		Modifier	
9	-Gestion de commentaires	Ajouter	
		Modifier	
10	-Gestion de panier	Ajouter	
		Supprimer	
12	-Gestion de boutique	Créer	
		Modifier	
13	-Attribuer une note	Noter un produit	
		Noter une boutique	

TABLE 2.3 – Identification des cas d'utilisation

Description textuelle des cas d'utilisations

Dans ce qui suit, nous allons décrire l'ensemble des cas d'utilisation.

● **Description textuelle de cas d'utilisation « Rechercher »**

Titre	Rechercher
Acteur	Internaute/Membre/Administrateur
Description	L'utilisateur a la possibilité de chercher un produit, une boutique
Précondition	L'internaute doit être connecté au site.
Scénario nominal	1- L'utilisateur se connecte au site. 2- L'utilisateur Saisi le(s) mot(s) clé dans la barre de recherche. 3- L'utilisateur Valide la recherche. 4- Le système affiche le résultat de la recherche avec un formulaire pour la recherche avancé.
Scénario alternatif	5. (A1) : -S'il n'existe pas de résultat, alors le système affiche un message de non existence de résultat pour la recherche.

TABLE 2.4 – Description textuelle du cas d'utilisation « Rechercher »

● **Description textuelle de cas d'utilisation « Modifier un compte »**

Titre	Modifier un compte
Acteur	Membre
Description	L'utilisateur a la possibilité de modifier les informations de son compte.
Précondition	L'utilisateur doit s'authentifier
Scénario nominal	1- L'utilisateur s'authentifie . 2- L'utilisateur demande à modifier son compte. 3- Le système affiche un formulaire. 4- L'utilisateur effectue les modifications désirée et valide. 5- Le système enregistre les modifications et affiche un message de confirmation.

TABLE 2.5 – Description textuelle du cas d'utilisation « Modifier un compte »

● **Description textuelle de cas d'utilisation « S'authentifier »**

Titre	S'authentifier
Acteur	Internaute
Description	L'utilisateur peut s'authentifier pour accéder à son compte.
Précondition	L'internaute doit être connecté au site.
Scénario nominal	<ol style="list-style-type: none"> 1- L'utilisateur saisit les informations nécessaires (Email et un mot de passe). 2- L'utilisateur valide l'authentification. 3- Le système vérifie le mot de passe et l'Email (pour confirmer l'existence du compte). 4- Le système ouvre la session. 5- L'utilisateur accède à son compte. .
Scénario alternatif	<ol style="list-style-type: none"> 3. (A1) : - Si un champ de saisis et vide, alors le système indique qu'un champ est incomplet et demande de ressaisir. 4. (A2) : Si le Email et/ou le mot de passe sont incorrect, le système affiche un message d'erreur.

TABLE 2.6 – Description textuelle du cas d'utilisation « S'authentifier »

● **Description textuelle de cas d'utilisation « Supprimer un compte »**

Titre	Supprimer un compte
Acteur	Membre/Administrateur
Description	L'utilisateur a la possibilité de supprimer un compte
Précondition	L'utilisateur doit s'authentifier
Scénario nominal	<ol style="list-style-type: none"> 1- L'utilisateur s'authentifie. 2- L'utilisateur demande à supprimer un compte. 3- Le système demande une confirmation. 4- L'utilisateur Valide la suppression. 5- Le système supprime le compte et affiche un message de confirmation.

TABLE 2.7 – Description textuelle du cas d'utilisation « Supprimer un compte »

● **Description textuelle de cas d'utilisation « Supprimer un produit »**

Titre	Supprimer un produit
Acteur	Membre/Administrateur
Description	L'utilisateur a la possibilité de supprimer un produit.
Précondition	L'utilisateur doit s'authentifier .
Scénario nominal	<ol style="list-style-type: none"> 1- L'utilisateur s'authentifie . 2- L'utilisateur demande à supprimer un produit . 3- Le système demande une confirmation. 4- L'utilisateur valide la suppression. 5- Le système supprime le produit et affiche un message de confirmation.

TABLE 2.8 – Description textuelle du cas d'utilisation « Supprimer un produit »

● **Description textuelle de cas d'utilisation « Supprimer un commentaire »**

Titre	Supprimer un commentaire
Acteur	Membre/Administrateur
Description	L'utilisateur a la possibilité supprimer un commentaire .
Précondition	L'utilisateur doit s'authentifier .
Scénario nominal	<ol style="list-style-type: none"> 1- L'utilisateur s'authentifie . 2- L'utilisateur demande de supprimer un commentaire. 3- Le système demande une confirmation. 4- L'utilisateur confirme la suppression. 5- Le système supprime le commentaire et affiche un message de confirmation.

TABLE 2.9 – Description textuelle du cas d'utilisation « Supprimer un commentaire »

● **Description textuelle de cas d'utilisation « Supprimer une boutique »**

Titre	Supprimer une boutique
Acteur	Membre/Administrateur
Description	L'utilisateur a la possibilité de supprimer une boutique .
Précondition	L'utilisateur doit s'authentifier.
Scénario nominal	<ol style="list-style-type: none"> 1- L'utilisateur s'authentifie. 2- L'utilisateur demande de supprimer la boutique. 3- Le système demande une confirmation. 4- L'utilisateur confirme la suppression. 5- Le système supprime la boutique et affiche un message de confirmation.

TABLE 2.10 – Description textuelle du cas d'utilisation « Supprimer une boutique »

● **Description textuelle de cas d'utilisation « Gestion de produit »**

Titre	Gestion de produit
Acteur	Membre
Description	L'utilisateur a la possibilité d'ajouter ou de modifier un produit.
Précondition	L'utilisateur doit s'authentifier .
Scénario nominal	<ol style="list-style-type: none"> 1- L'utilisateur s'authentifie. 2- L'utilisateur demande à (ajouter, modifier) un produit. 3- Le système affiche un formulaire. 4-L'utilisateur saisis les informations voulu puis confirme. 5-Le système enregistre puis affiche un message de confirmation.
Scénario alternatif	5-(A1) : Si les données saisies sont incorrectes (incompatibilité avec le type de champ, champ vide) alors le système affiche un message d'erreur.

TABLE 2.11 – Description textuelle du cas d'utilisation « Gestion de produit »

● **Description textuelle de cas d'utilisation « Gestion de commentaires »**

Titre	Gestion de commentaires
Acteur	Membre
Description	L'utilisateur a la possibilité d'ajouter ou de modifier un commentaire.
Précondition	L'utilisateur doit s'authentifier.
Scénario nominal	<ol style="list-style-type: none"> 1- L'utilisateur s'authentifie. 2- L'utilisateur demande d'ajouter/modifier un commentaire. 3- Le système affiche un formulaire. 4- L'utilisateur remplit le formulaire et valide. 5- Le système enregistre puis affiche un message de confirmation.

TABLE 2.12 – Description textuelle du cas d'utilisation « Gestion de commentaires »

● **Description textuelle de cas d'utilisation « Gestion du panier »**

Titre	Gestion du panier
Acteur	Membre
Description	L'utilisateur a la possibilité d'ajouter ou de supprimer un produit dans son panier .
Précondition	L'utilisateur doit s'authentifier .
Scénario nominal	<ol style="list-style-type: none"> 1- L'utilisateur s'authentifie. 2- L'utilisateur demande à (ajouter, supprimer) un produit dans son panier, le système demande une confirmation s'il s'agit d'une suppression, l'utilisateur confirme alors la suppression.

TABLE 2.13 – Description textuelle du cas d'utilisation « Gestion du panier »

● **Description textuelle de cas d'utilisation « Gestion de boutique »**

Titre	Gestion de boutique
Acteur	Membre
Description	L'utilisateur a la possibilité de créer ou de modifier une boutique.
Précondition	L'utilisateur doit s'authentifier.
Scénario nominal	<ol style="list-style-type: none"> 1- L'utilisateur s'authentifie . 2- L'utilisateur demande à créer/modifier une boutique . 3- Le système lui renvoi le formulaire de création/modification . 4- L'utilisateur saisi les informations et confirme . 5- Le système effectue l'action demander créer/modifier . 6- L'utilisateur accède à sa boutique.
Scénario alternatif	3. (A1) : -En cas de problème dans la saisi des informations (champ vide, le nom de boutique existe déjà) le système renvoi un message d'erreur.

TABLE 2.14 – Description textuelle du cas d'utilisation « Gestion de boutique »

● **Description textuelle de cas d'utilisation « Attribuer une note»**

Titre	Attribuer une note
Acteur	Membre
Description	L'utilisateur a la possibilité d'attribuer une note à un produit ou une boutique.
Précondition	L'utilisateur doit s'authentifier.
Scénario nominal	<ol style="list-style-type: none"> 1- Une fois que l'utilisateur finalise un achat (confirme la réception d'un produit) 2- L'utilisateur attribue une note à un produit ou à une boutique. 3- Le système affiche un message de confirmation.

TABLE 2.15 – Description textuelle du cas d'utilisation « Attribuer une note »

● Description textuelle de cas d'utilisation « Gestion du commande »

Titre	Gestion de commande
Acteur	Membre
Description	L'utilisateur a la possibilité d'effectuer ou d'annuler une commande
Précondition	L'utilisateur doit s'authentifier. Avant une annulation l'utilisateur (vendeur) ne doit pas valider la commande
Scénario nominal	<p>A- Effectuer une commande :</p> <ol style="list-style-type: none"> 1- L'utilisateur sélectionne le(s) produit(s) dans son panier pour effectuer la commande. 2- Le système renvoie l'utilisateur vers le site de paypal 3- Paypal lui affiche un formulaire 4- L'utilisateur saisi les informations et valide 5- Paypal demande une confirmation de la commande 6- L'utilisateur confirme la commande 7- Paypal procède au versement 8- Paypal envoie une alerte de succès au système et enregistre la commande 9- Le système envoie une requête d'enregistrement vers la base de donnée 10- La requête est enregistrée et une alerte de succès est envoyée au système 11- Le système envoie une notification d'enregistrement à l'utilisateur <p>B- Annuler une commande</p> <ol style="list-style-type: none"> 1- L'utilisateur sélectionne la commande et demande son annulation 2- Le système demande une confirmation 3- L'utilisateur confirme 4- Le système envoie la requête à la base de données 5- La commande est annulée et une confirmation est envoyée au système 6- Le système confirme l'annulation à l'utilisateur
Scénario alternatif	<p>A- 4(A1) : - en cas d'erreur paypal affiche un message d'erreur et réaffiche le formulaire.</p> <p>A- 6(A1) : - en cas d'échec de la transaction (solde insuffisant) paypal envoie une notification d'erreur à l'utilisateur.</p>

TABLE 2.16 – Description textuelle du cas d'utilisation « Gestion de commande »

Diagramme de cas d'utilisation

Définition : UML propose de représenter les cas d'utilisation sous une forme graphique nommée diagramme de cas d'utilisation. Les rôles sont définis pour chaque acteur. Une relation entre l'acteur et le cas d'utilisation représente une communication. Le cas d'utilisation est représenté par une ellipse qui porte son nom à l'intérieur. Des notes peuvent être portées sur le diagramme afin d'y ajouter des informations.

Un diagramme de cas d'utilisation montre les acteurs et les cas d'utilisation ensemble avec leurs relations. La relation entre un acteur et un cas d'utilisation est appelée association et correspond au fait que l'acteur participe à un cas d'utilisation. Les cas d'utilisation représentent les fonctionnalités d'un système, ou d'une entité d'un système, telles qu'elles sont sollicitées en interaction avec des événements extérieurs. Ils donnent une vision "haute" et dynamique du système.[10]

Le diagramme de cas d'utilisation associé à l'internaute

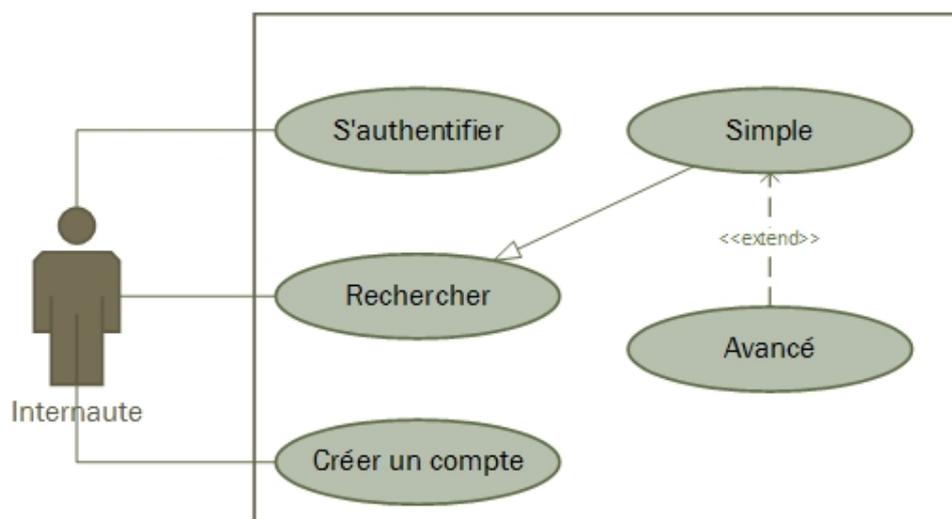


FIGURE 2.2 – Le diagramme de cas d'utilisation associé à l'internaute

Le diagramme de cas d'utilisation associé au Membre

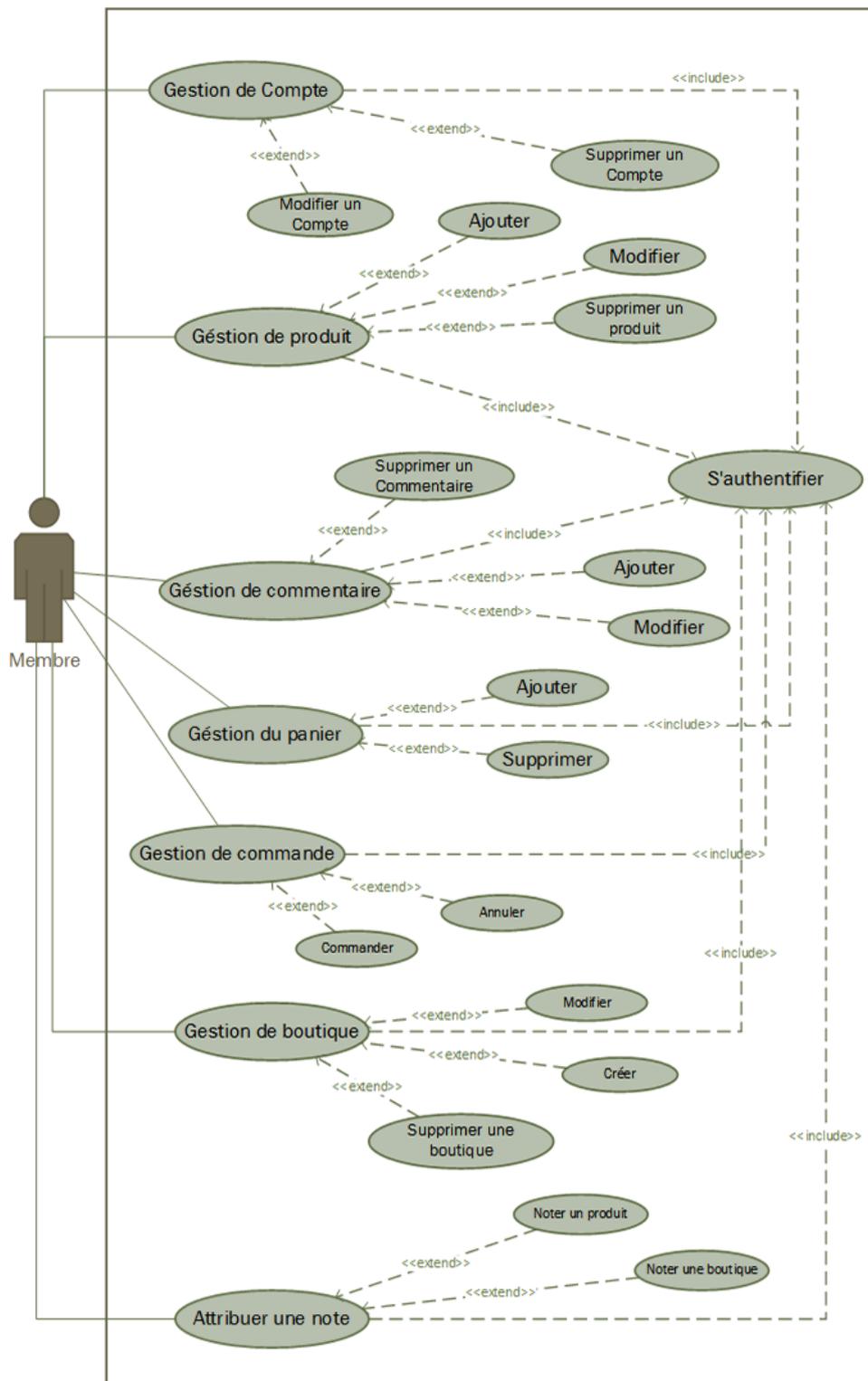


FIGURE 2.3 – Le diagramme de cas d'utilisation associé au Membre

Le diagramme de cas d'utilisation associé à l'Administrateur

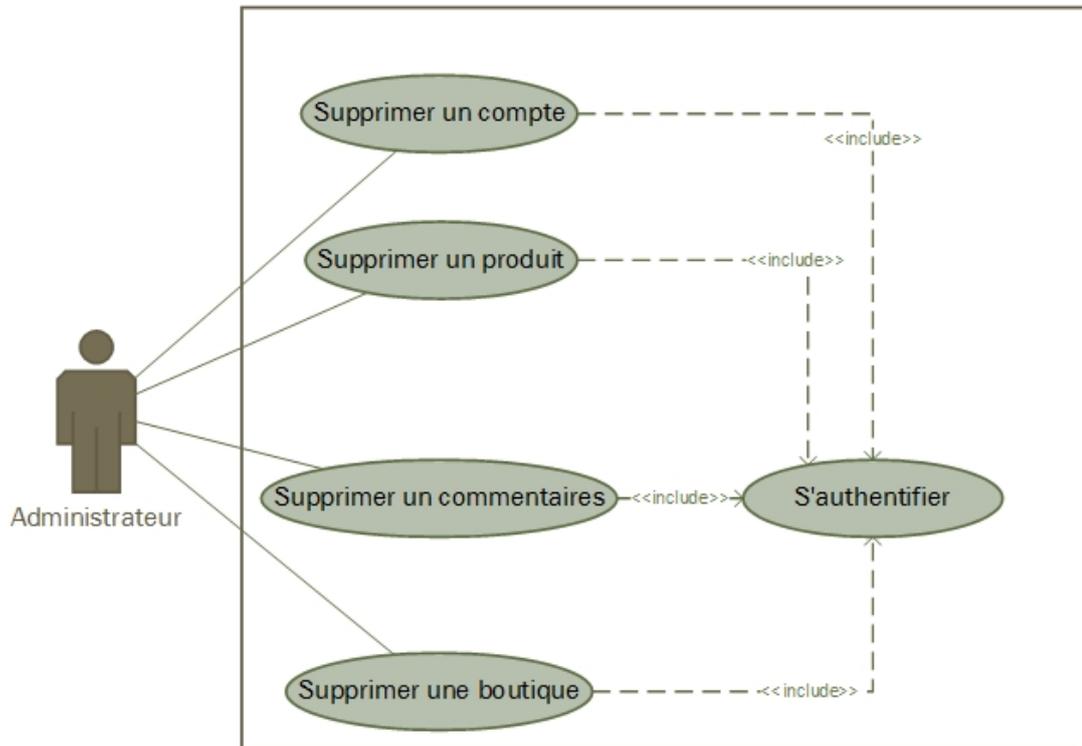


FIGURE 2.4 – Le diagramme de cas d'utilisation associé à l'Administrateur

Le diagramme de cas d'utilisation global

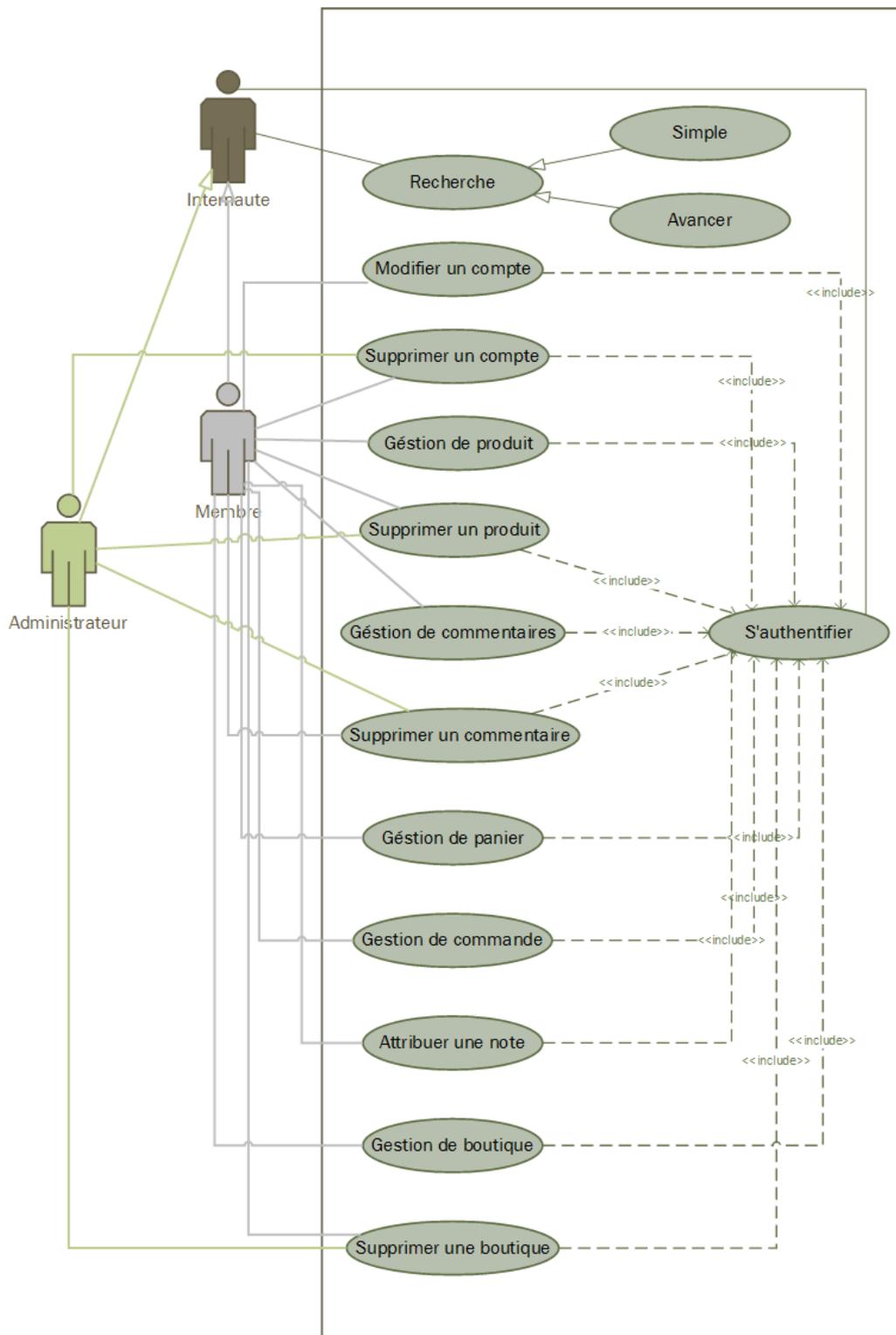


FIGURE 2.5 – Le diagramme de cas d'utilisation global

2.2.2 Capture des besoins techniques

La capture des besoins techniques couvre, par complémentarité avec celle des besoins fonctionnels, toutes les contraintes qui ne traitent ni de la description du métier des utilisateurs, ni de la description applicative. Le modèle de spécification logicielle concerne donc les contraintes techniques telles que les performances d'accès aux données, la sécurité du système, l'interopérabilité, l'intégration des applications, la volumétrie et le mode d'utilisation du système.

2.2.3 L'architecture d'application[11]

Dans le cadre de notre projet, nous avons choisi de mettre en place l'architecture 3-tiers qui sépare l'application en trois niveaux :

-Premier niveau : Il contient la couche présentation et les traitements locaux de la couche traitement pris en charge par le poste client.

-Deuxième niveau : Ce niveau est appelé « couche métier », il correspond à la partie fonctionnelle de l'application. Les opérations à réaliser, les fonctions d'accès aux données, tous ces traitements sont mis à disposition des utilisateurs et invoqués par leurs requêtes.

Troisième niveau : Il contient la couche données qui conserve les données dans le serveur sous forme d'une base de données.

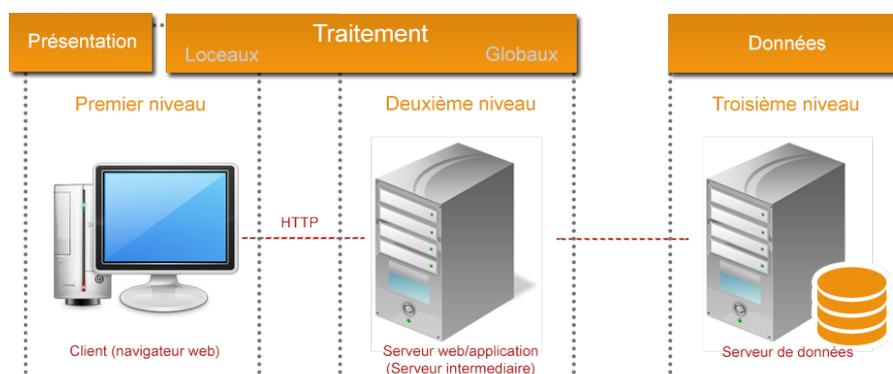


FIGURE 2.6 – L'architecture 3-tiers

2.2.4 Le but d l'architecture en 3-tiere

Cette architecture a pour but d'établir certaines exploitations du système parmi lesquelles :

- La partie client est composée uniquement d'affichage (pas de programmation, pas de requêtes SQL...). De ce fait, des modifications peuvent être réalisées au niveau de la logique des traitements locaux, globaux et au niveau des données sans que cela apporte un impact sur la couche cliente.
- Du point de vue développement, la séparation entre le client, le serveur et les données permet une spécialisation des développeurs et une meilleure répartition des tâches et fonctions (développeur de modèle/designer, programmeur, administrateur de bases de données...).

Conclusion

Dans ce chapitre nous avons fait une analyse de l'existant ainsi qu'une spécification des différents besoins fonctionnels et techniques, ces besoins regroupent l'ensemble des services qu'offre le système. Dans le chapitre qui suit nous allons aborder l'analyse et la conception.

3

Analyse et conception

Introduction

En se référant à la démarche 2TUP, et après avoir déterminé la partie de capture et analyse des besoins, on passe à la phase d'analyse et conception qui représente deux étapes essentielles à la démarche, ces étapes nous permettent de passer d'une structure fonctionnelle via les cas d'utilisation à une structure objet sous forme de modèle de classe. Au cours de ce qui suit, nous représenterons une vue dynamique du système par la modélisation des diagrammes de séquences et d'activités et une vue statique par la modélisation du diagramme de classe.

3.1 Diagramme de séquence

Le diagramme de séquence permet d'afficher les interactions qui représentent la séquence de messages entre les instances du système (les classes, les composants, les sous-

systèmes ou les acteurs), cette représentation peut se réaliser par un cas d'utilisation en considérant les différents scénarios associés.[12]

3.1.1 Etude de cas d'utilisation

- Diagramme de séquence du cas d'utilisation « Effectuer une recherche »

Pour effectuer une recherche, l'utilisateur doit saisir les mots clés spécifiques au résultat qu'il souhaite obtenir si des résultats existent alors le système l'affiche avec un formulaire de recherche avancé qui permettra à l'utilisateur d'effectuer une recherche plus poussée et ainsi mieux cibler le résultat souhaiter. Pour ce cas BDD fait référence à des produits ou boutiques.

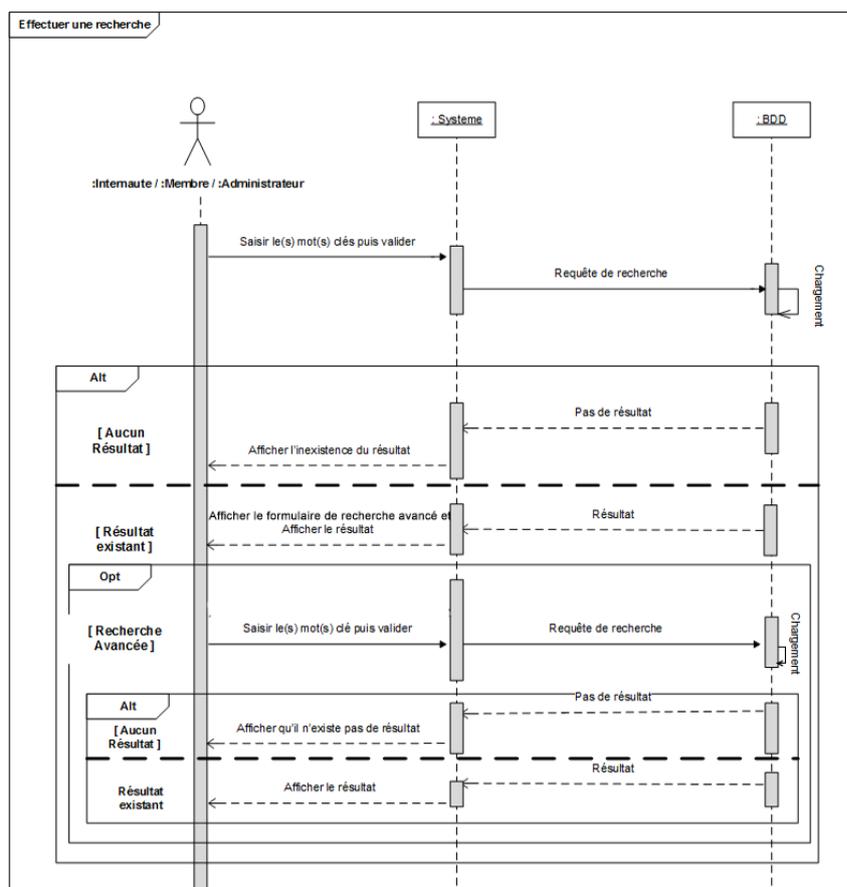


FIGURE 3.1 – Diagramme de séquence du cas d'utilisation « Effectuer une recherche »

- Diagramme de séquence du cas d'utilisation « S'authentifier »

Un utilisateur a la possibilité de s'authentifier en saisissant un Email et un mot de passe dans les champs qui leurs sont réservés, après validation, l'utilisateur accède à la session qui lui est propre si les données sont correctes et figurent dans la BDD, sinon le système renvoi un message d'erreur.

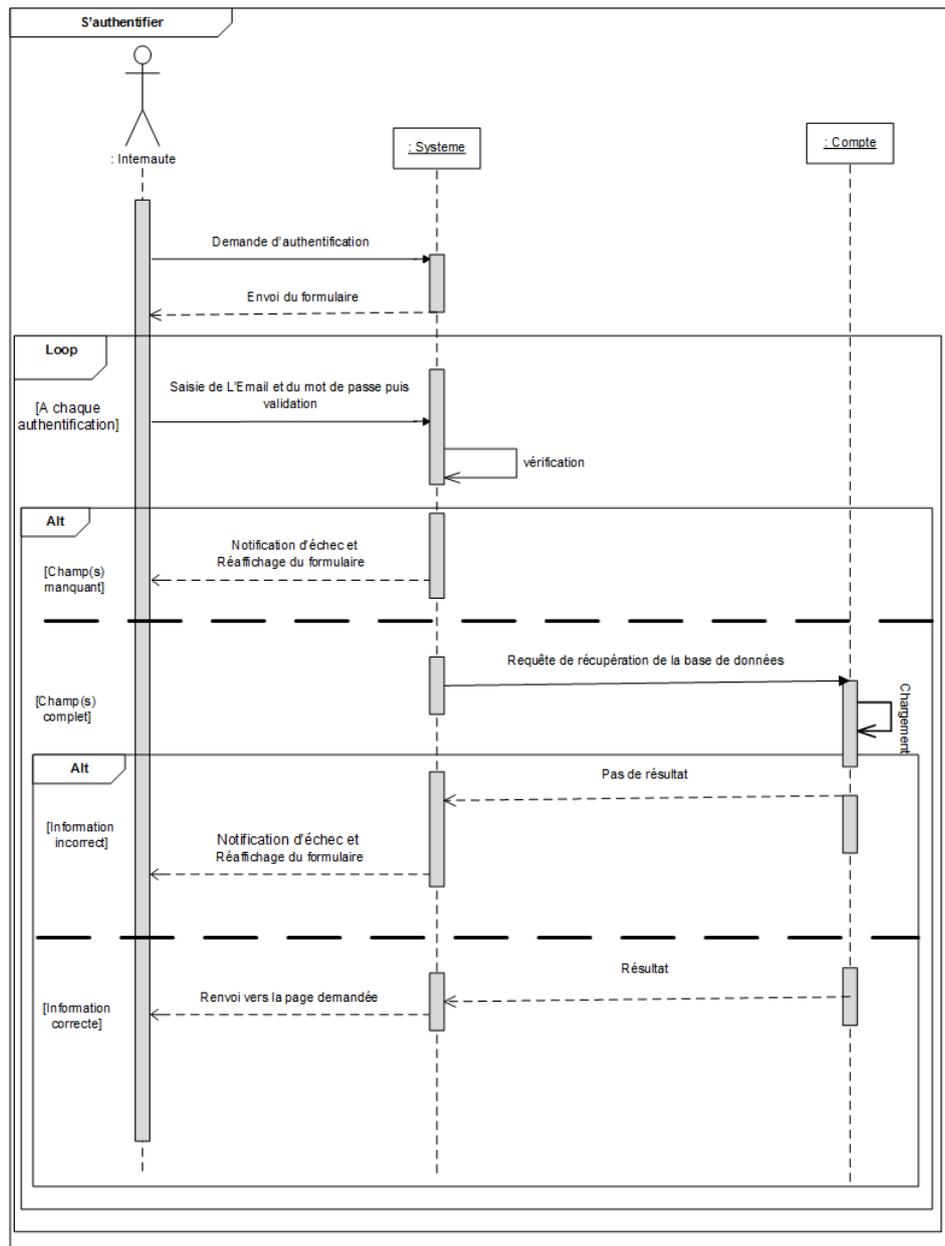


FIGURE 3.2 – Diagramme de séquence du cas d'utilisation « S'authentifier »

- Diagramme de séquence du cas d'utilisation « Modifier un compte »

Les membres ont la possibilité de modifier les données de leurs comptes personnels, pour ce faire, ils demandent une fois authentifiés à modifier leurs profils, le système leur renvoie alors un formulaire qu'ils remplissent, une fois les modifications voulues effectuées, ils valident, si aucune erreur n'est signalé, le système affiche un message de confirmation.

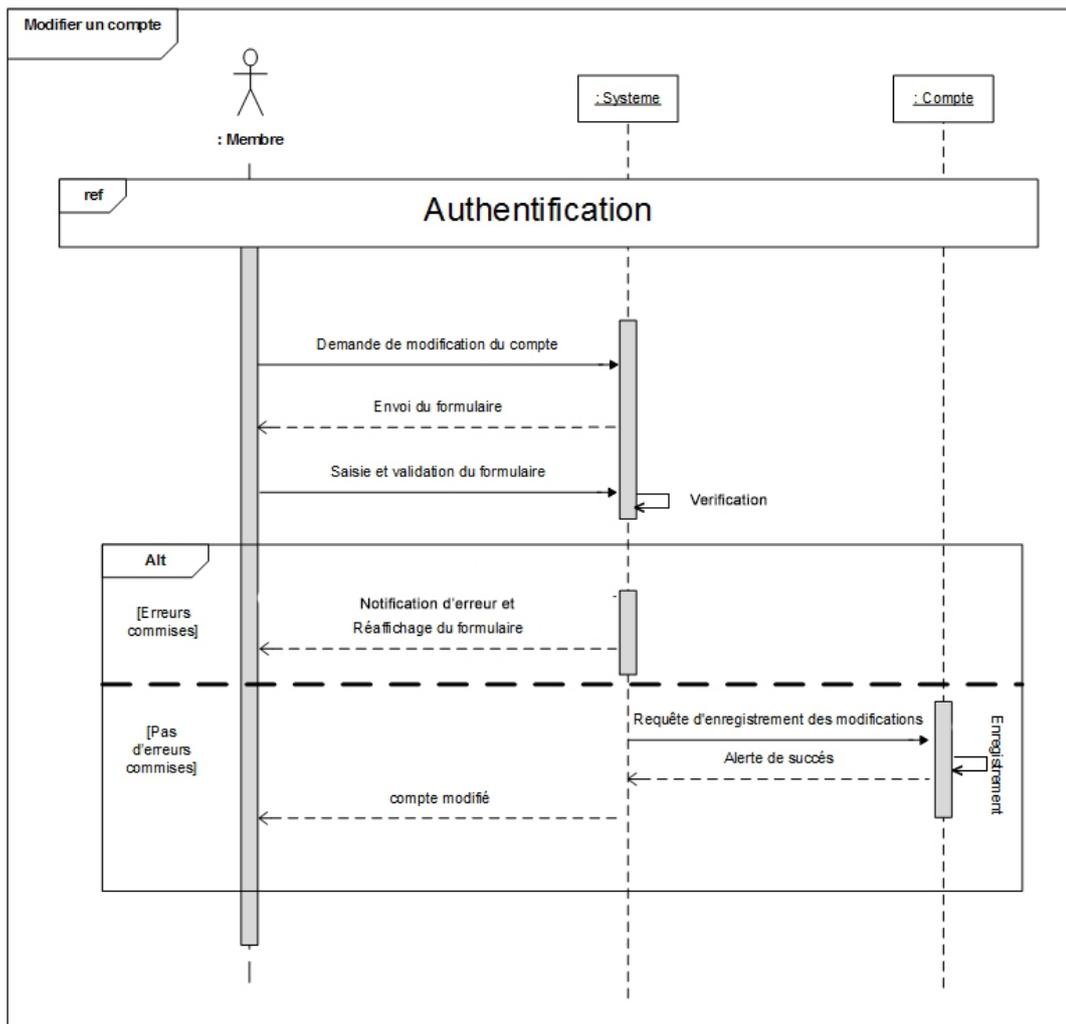


FIGURE 3.3 – Diagramme de séquence du cas d'utilisation « Modifier un compte »

- Diagramme de séquence du cas d'utilisation « Supprimer un compte »

L'utilisateur a la possibilité de résilier son compte s'il n'en a plus d'utilité ou dans le cas de l'administrateur s'il le juge indésirables, pour ce faire l'utilisateur demande de supprimer un compte, le système demande alors une confirmation, si l'utilisateur confirme alors le compte sera supprimé de la BDD, le système renvoi à l'utilisateur un message de confirmation.

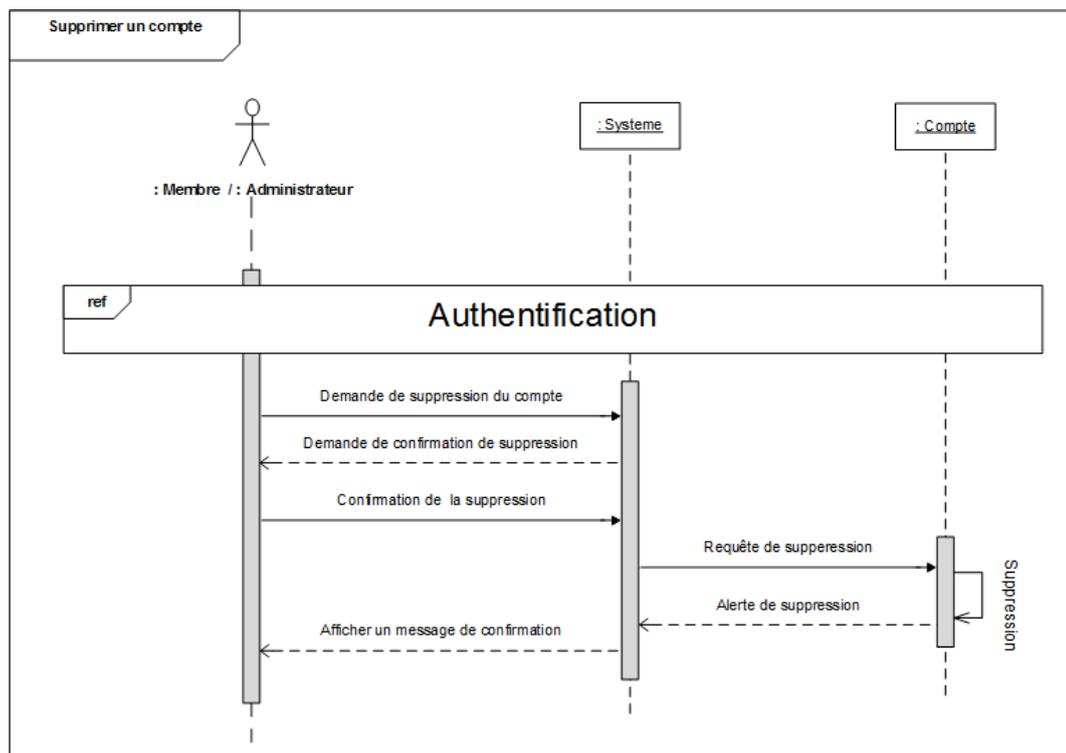


FIGURE 3.4 – Diagramme de séquence du cas d'utilisation « Supprimer un compte »

- Diagramme de séquence du cas d'utilisation « Gestion de produit »

L'utilisateur (membre) peut dans ce cas de figure avoir recours à deux cas d'utilisation, à savoir, ajouter et modifier un produit. Dans l'ajout d'un nouveau produit, l'utilisateur demande l'ajout, le système lui renvoie un formulaire, l'utilisateur remplit le formulaire et valide, si aucune erreur n'est commise alors le système envoie une requête d'ajout à la BDD et là confirme, sinon il affiche un message d'erreur, l'utilisateur a recours au même procédé pour la modification à l'exception faite, que dans ce cas il doit sélectionner un

produit au préalable.

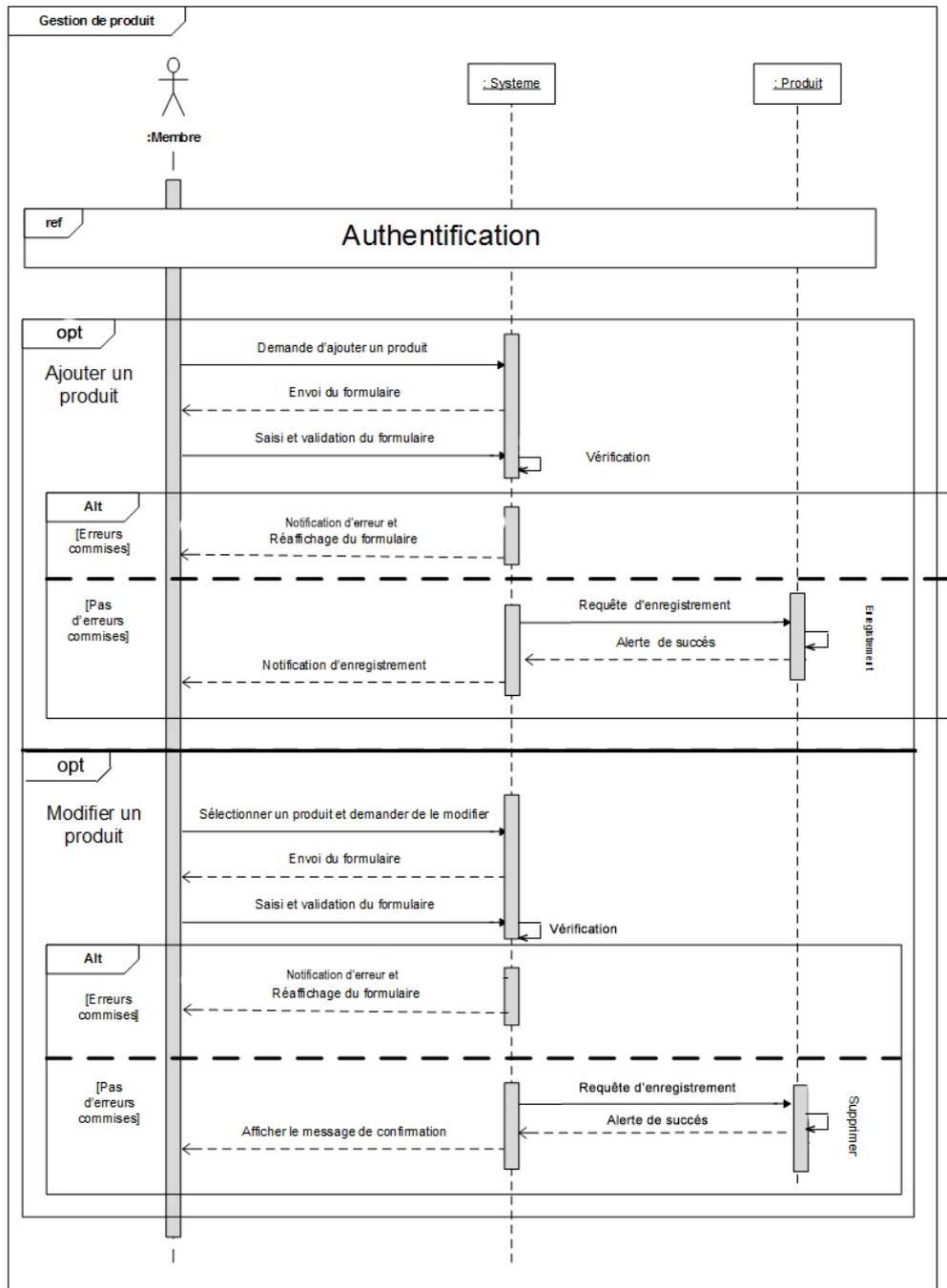


FIGURE 3.5 – Diagramme de séquence du cas d'utilisation « Gestion de produit »

- Diagramme de séquence du cas d'utilisation «Supprimer un produit »

L'utilisateur (membre, administrateur) a la possibilité de supprimer un produit, dans ce cas l'utilisateur demande de supprimer un produit, le système lui demande une confirmation, si l'utilisateur confirme alors le système envoie la requête de suppression à la BDD et confirme à l'utilisateur la suppression du produit.

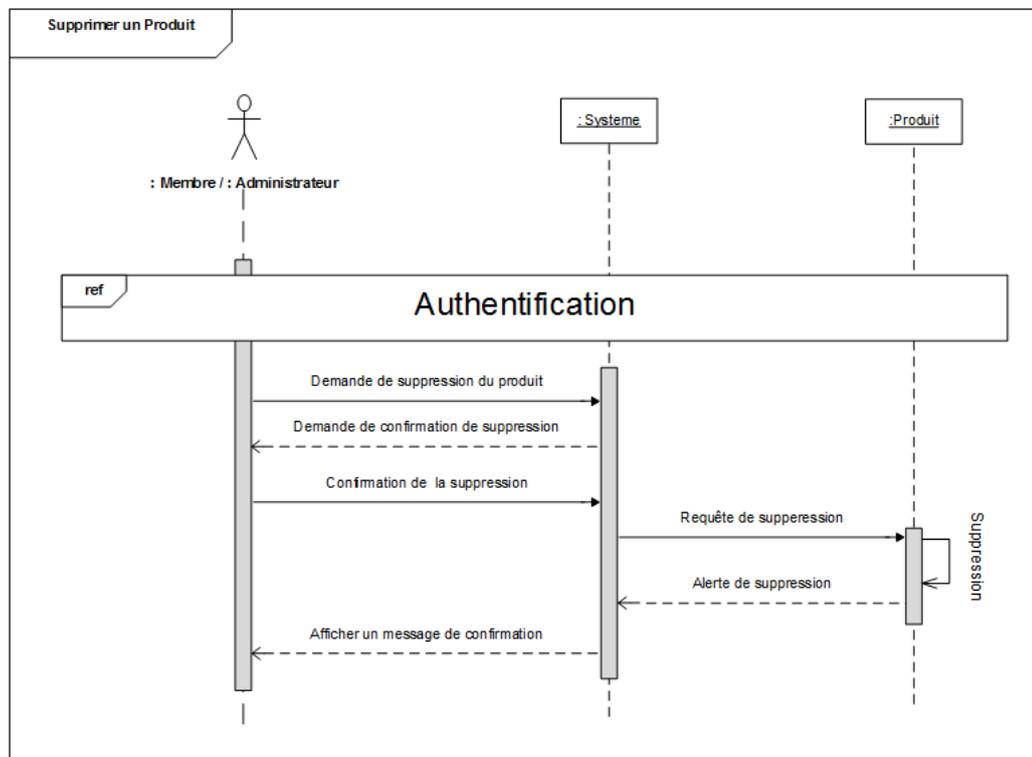


FIGURE 3.6 – Diagramme de séquence du cas d'utilisation «Supprimer un produit »

- Diagramme de séquence du cas d'utilisation «Gestion de boutique »

Le membre a la possibilité de créer une ou plusieurs boutiques, pour ce faire il demande une fois inscrit à créer une boutique, le système lui renvoie un formulaire qu'il remplit et valide une fois fini, si aucune erreur n'est commise alors la page est créée sinon le système renvoi un message d'erreur, il a aussi la possibilité de modifier ses boutiques, pour cela, il suit la même démarche que pour la création.

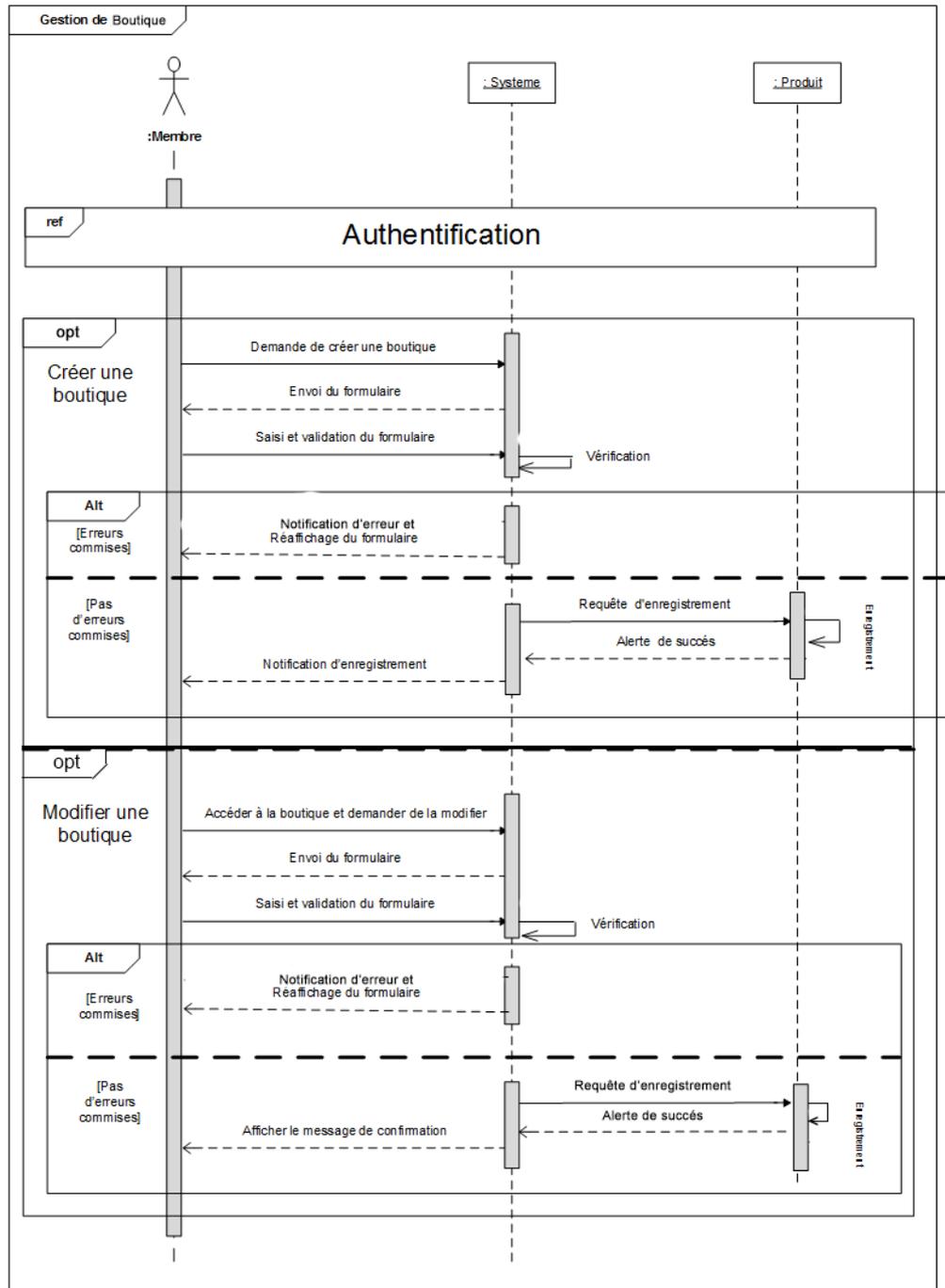


FIGURE 3.7 – Diagramme de séquence du cas d'utilisation « Gestion de boutique »

- Diagramme de séquence du cas d'utilisation « Supprimer une boutique »

L'utilisateur est en mesure de supprimer une boutique, il demande de supprimer la boutique, le système lui demande alors une validation, si l'utilisateur valide alors la requête est envoyée à la BDD et la boutique est supprimée, sinon la requête est abandonnée.

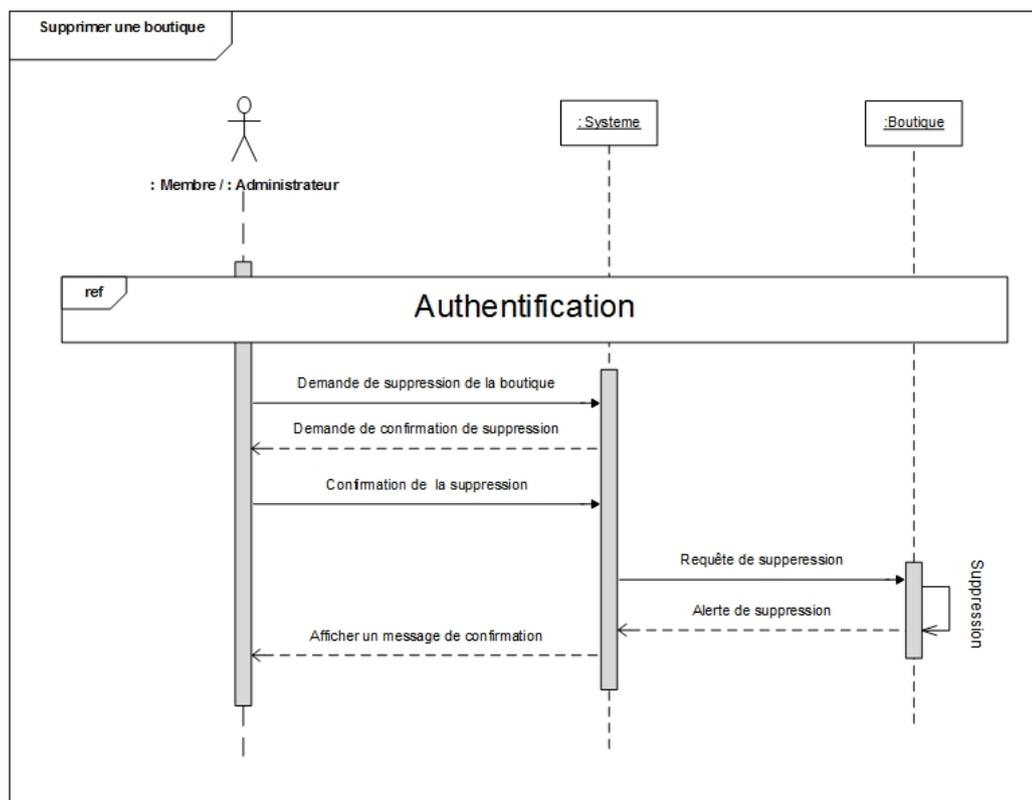


FIGURE 3.8 – Diagramme de séquence du cas d'utilisation « Supprimer une boutique »

- Diagramme de séquence du cas d'utilisation « Gestion de panier »

Un utilisateur est en mesure d'ajouter ou de supprimer des produits de son panier, dans le premier cas l'utilisateur demande d'ajouter le produit à son panier, le système demande une confirmation, s'il confirme alors le produit est ajouté, dans l'autre cas l'utilisateur demande d'accéder à son panier, il peut ensuite supprimer le produit, il valide, le système lui demande de confirmer, s'il confirme alors le produit est supprimé sinon la requête est abandonnée.

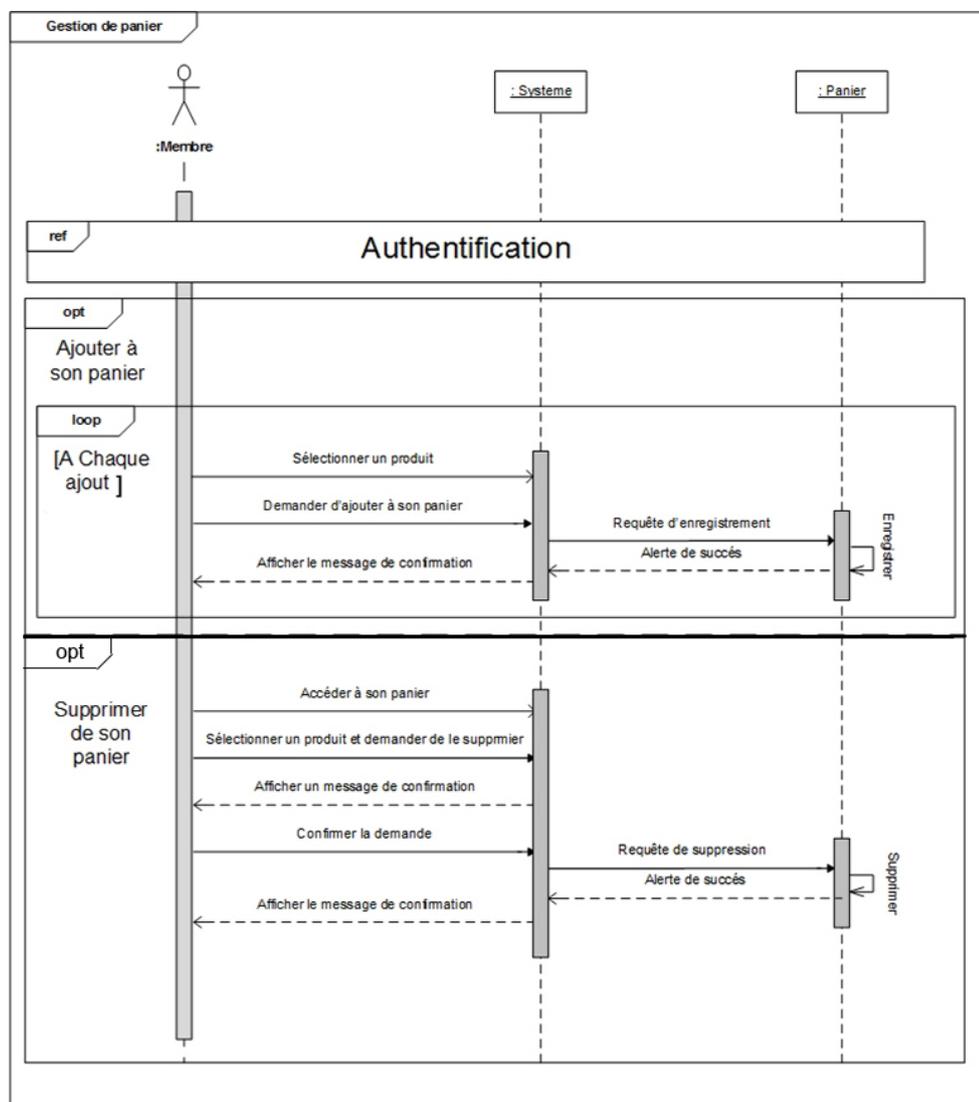


FIGURE 3.9 – Diagramme de séquence du cas d'utilisation « Gestion de panier »

- Diagramme de séquence du cas d'utilisation « Gestion de commentaires »

L'utilisateur est en mesure d'ajouter ou de modifier un commentaire, dans le premier cas, en sélectionnant un produit il saisit le commentaire et valide, le commentaire est alors ajouté, dans le deuxième cas, l'utilisateur demande à modifier son commentaire, le système lui renvoie un formulaire, l'utilisateur saisit le nouveau commentaire et valide, son commentaire est alors mis à jour.

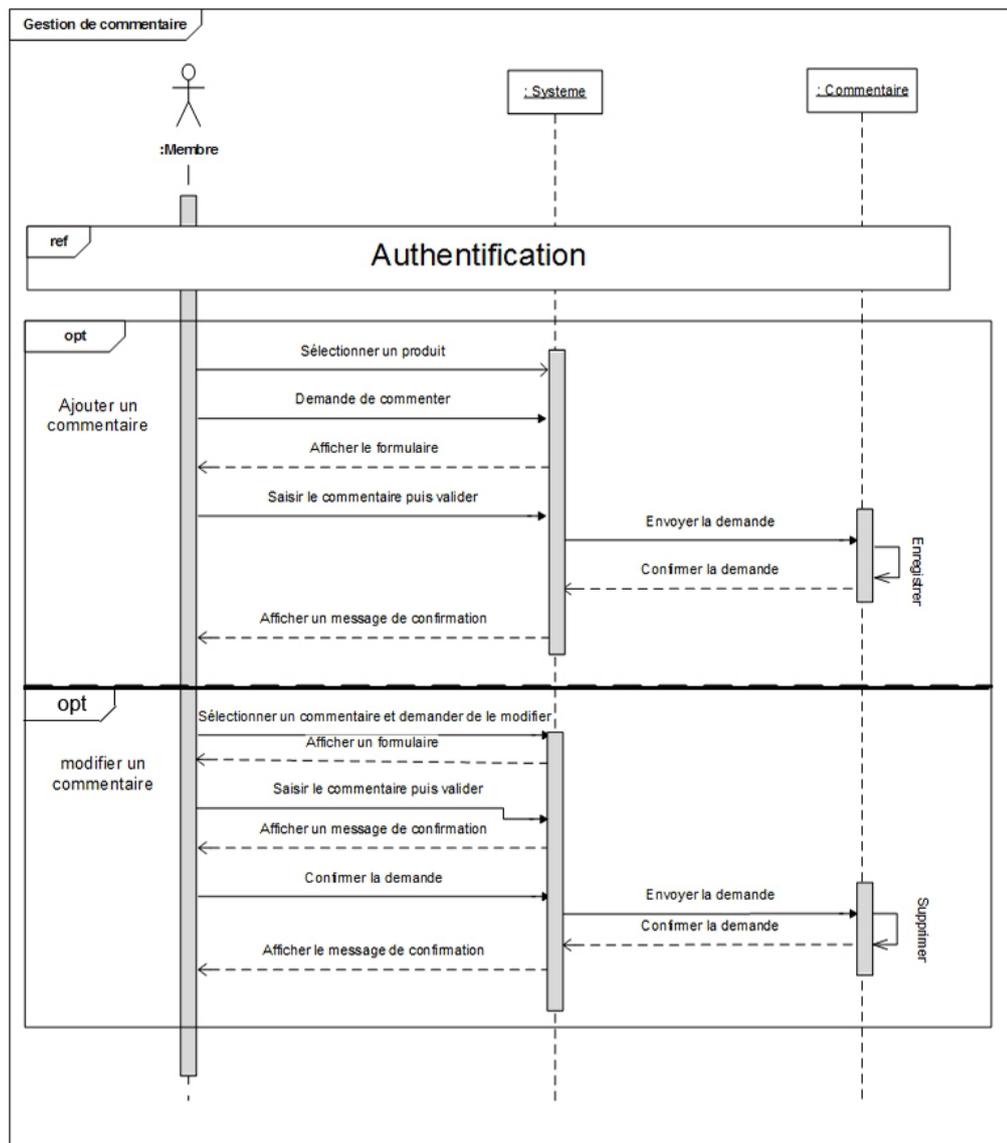


FIGURE 3.10 – Diagramme de séquence du cas d'utilisation « Gestion de commentaires »

- Diagramme de séquence du cas d'utilisation « Supprimer un commentaire »

L'utilisateur a la possibilité de supprimer son commentaire ou dans le cas où l'administrateur le juge indésirable, pour ce faire il demande la suppression du commentaire, le système demande une confirmation, s'il confirme alors le commentaire est supprimé dans le cas contraire la requête est abandonnée.

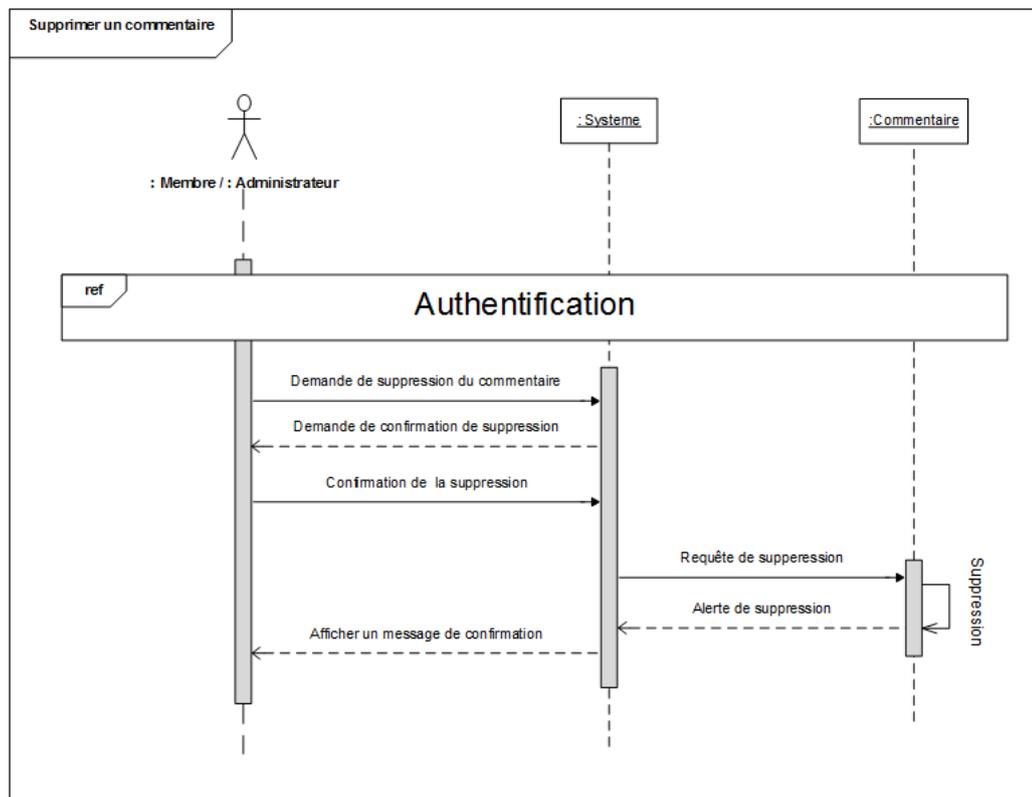


FIGURE 3.11 – Diagramme de séquence du cas d'utilisation «Supprimer un commentaire»

- Diagramme de séquence du cas d'utilisation « Gestion de commande »

L'utilisateur a la possibilité d'effectuer une commande, pour ce faire il sélectionne le produit dans son panier et le commande, il est alors redirigé vers PayPal qui lui affiche un formulaire d'authentification ; une fois authentifié, PayPal demande à l'utilisateur la confirmation de la commande, s'il valide et que son solde est bien suffisant alors PayPal procède au virement d'argent et alerte le système qui enregistre la commande et prévient

l'utilisateur.

L'utilisateur a aussi la possibilité d'annuler une commande si elle n'a pas été acceptée par le vendeur.

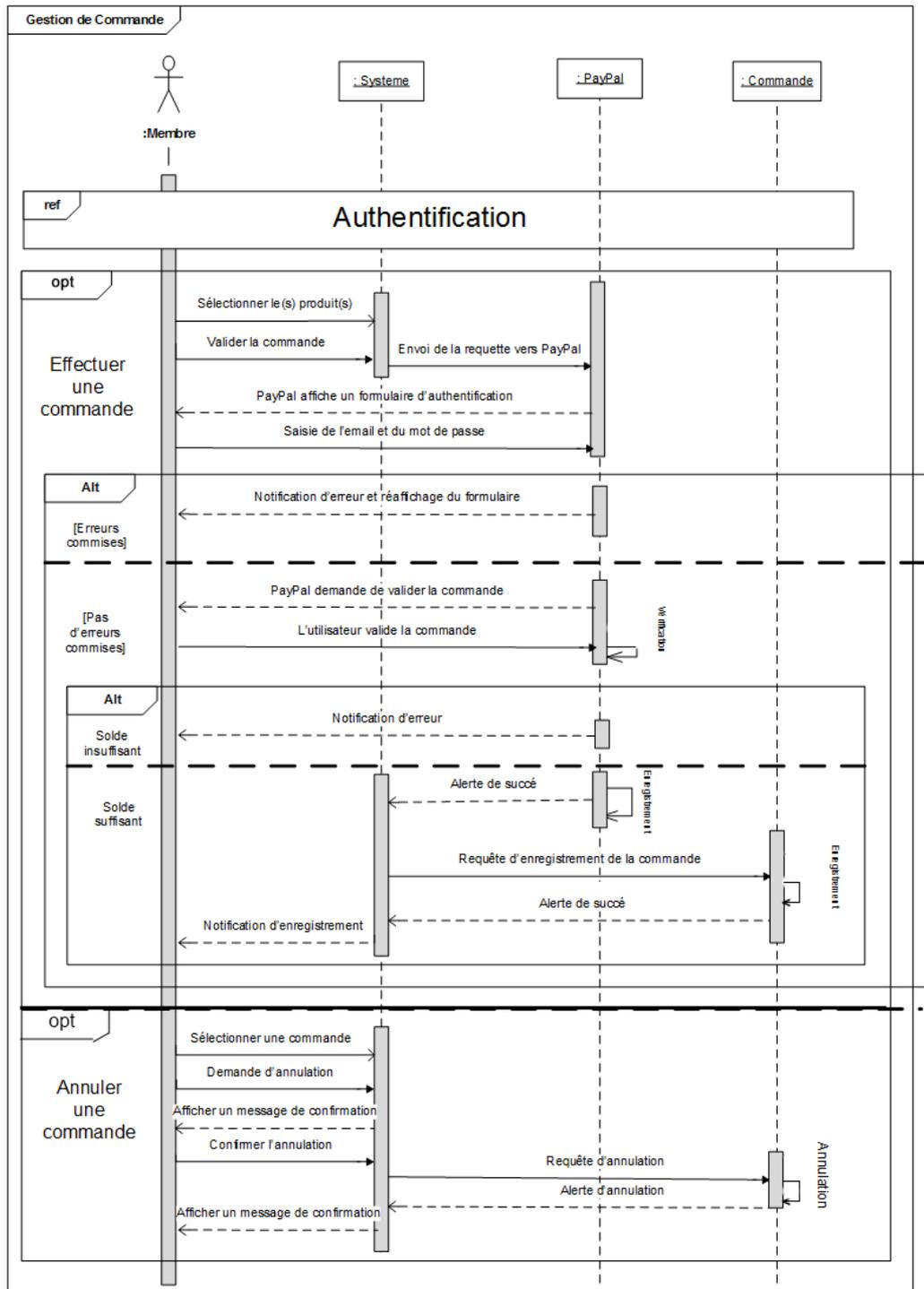


FIGURE 3.12 – Diagramme de séquence du cas d'utilisation « Gestion de commande »

3.1.2 - Diagramme de séquence du cas d'utilisation « Attribuer une note »

L'utilisateur a la possibilité d'attribuer des notes aux différents produits et boutiques quand ce dernier confirme la réception d'un produit, pour cela il demande à attribuer une note, le système renvoie un formulaire que l'utilisateur remplit, puis il valide. La note est alors enregistrée. Pour ce cas BDD fait référence à des produits ou boutiques.

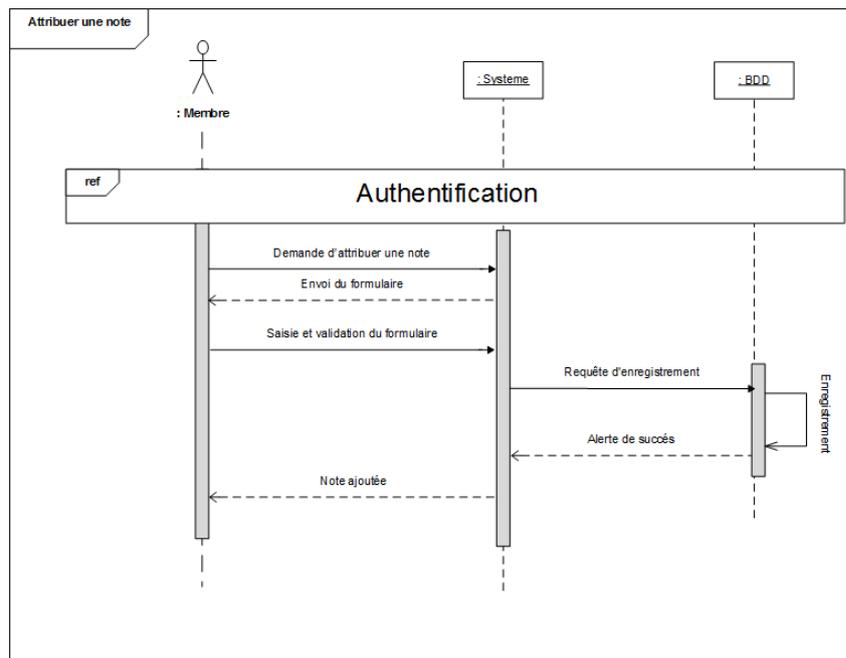


FIGURE 3.13 – Diagramme de séquence du cas d'utilisation « Attribuer une note »

3.2 Diagramme d'activité

3.2.1 Définition

Un diagramme d'activité est utilisé pour représenter le comportement d'une méthode ou le déroulement d'un cas d'utilisation, dans lequel les nœuds correspondent à l'exécution d'actions ou d'activités.[13]

3.2.2 Représentation des diagrammes d'activité

Cas de la boutique

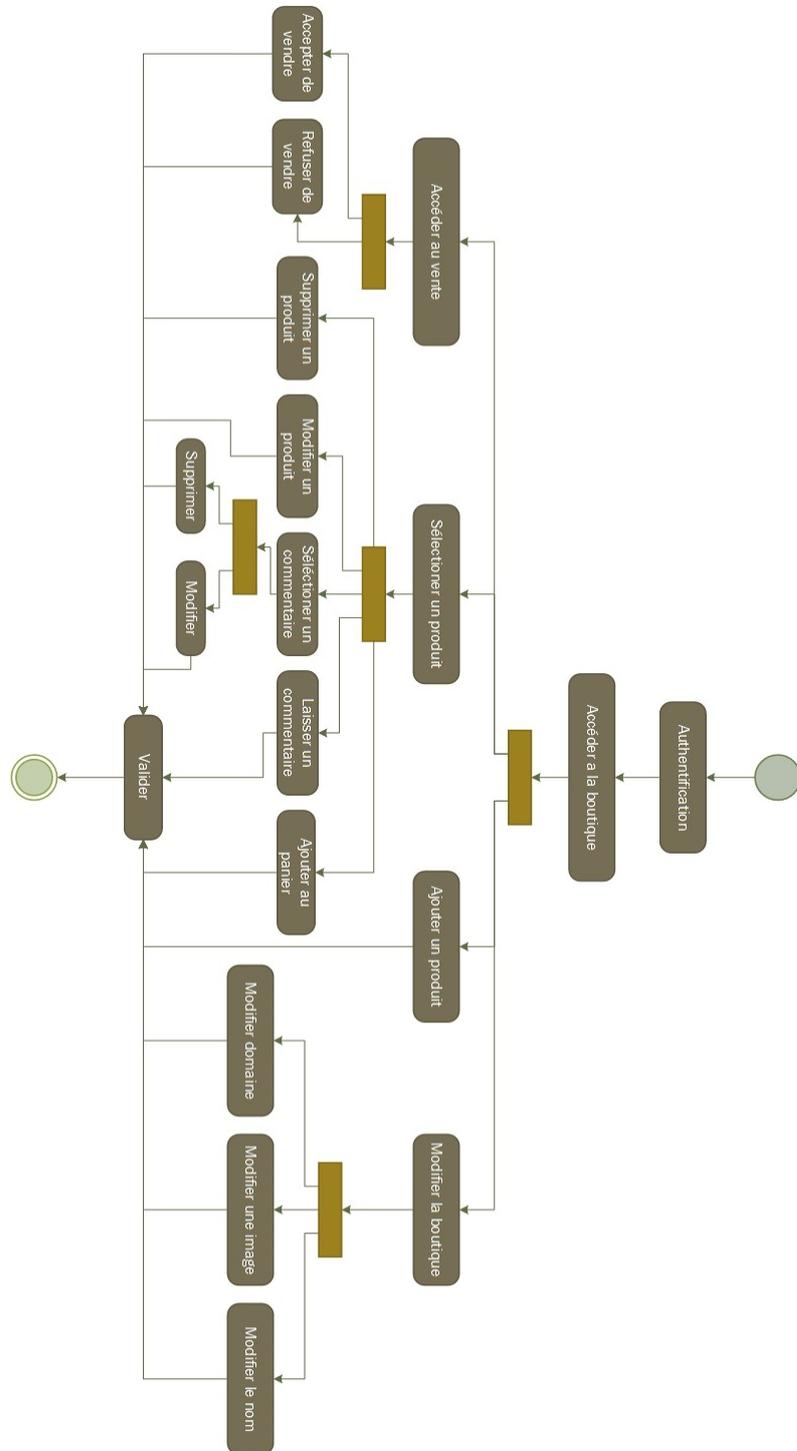


FIGURE 3.14 – Diagramme d'activité «Cas de la boutique»

Cas commande

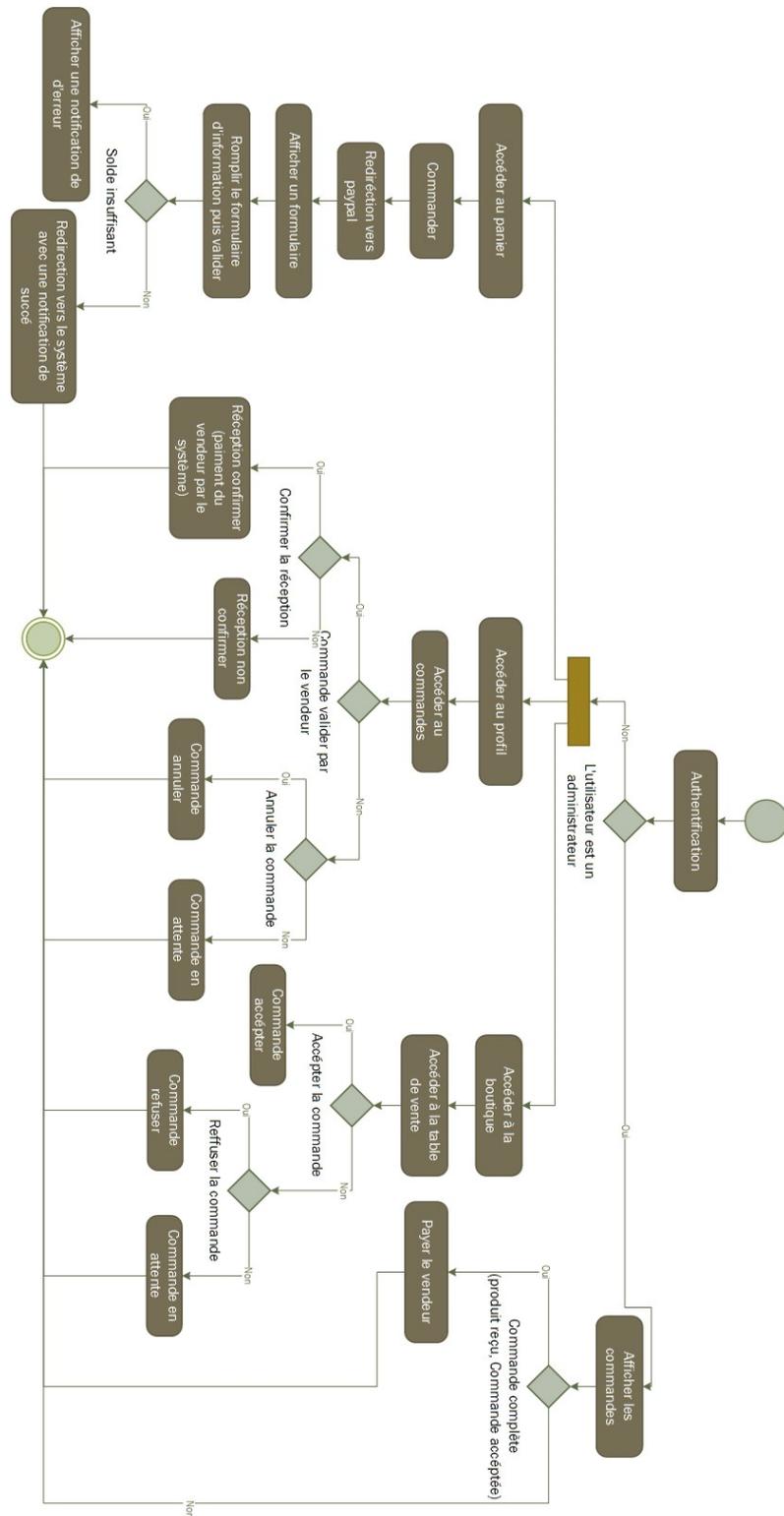


FIGURE 3.15 – Diagramme d'activité «Cas commande»

3.3 Diagramme de classe

Définition

Le diagramme de classes est généralement considéré comme le plus important dans la conception d'une application, il constitue l'un des pivots essentiels de la modélisation. Ce diagramme appartient au modèle statique et permet de définir la structure interne de notre système, car il fait abstraction des aspects temporels et dynamiques. Cette représentation est centrée sur les concepts de classe et l'association.[14]

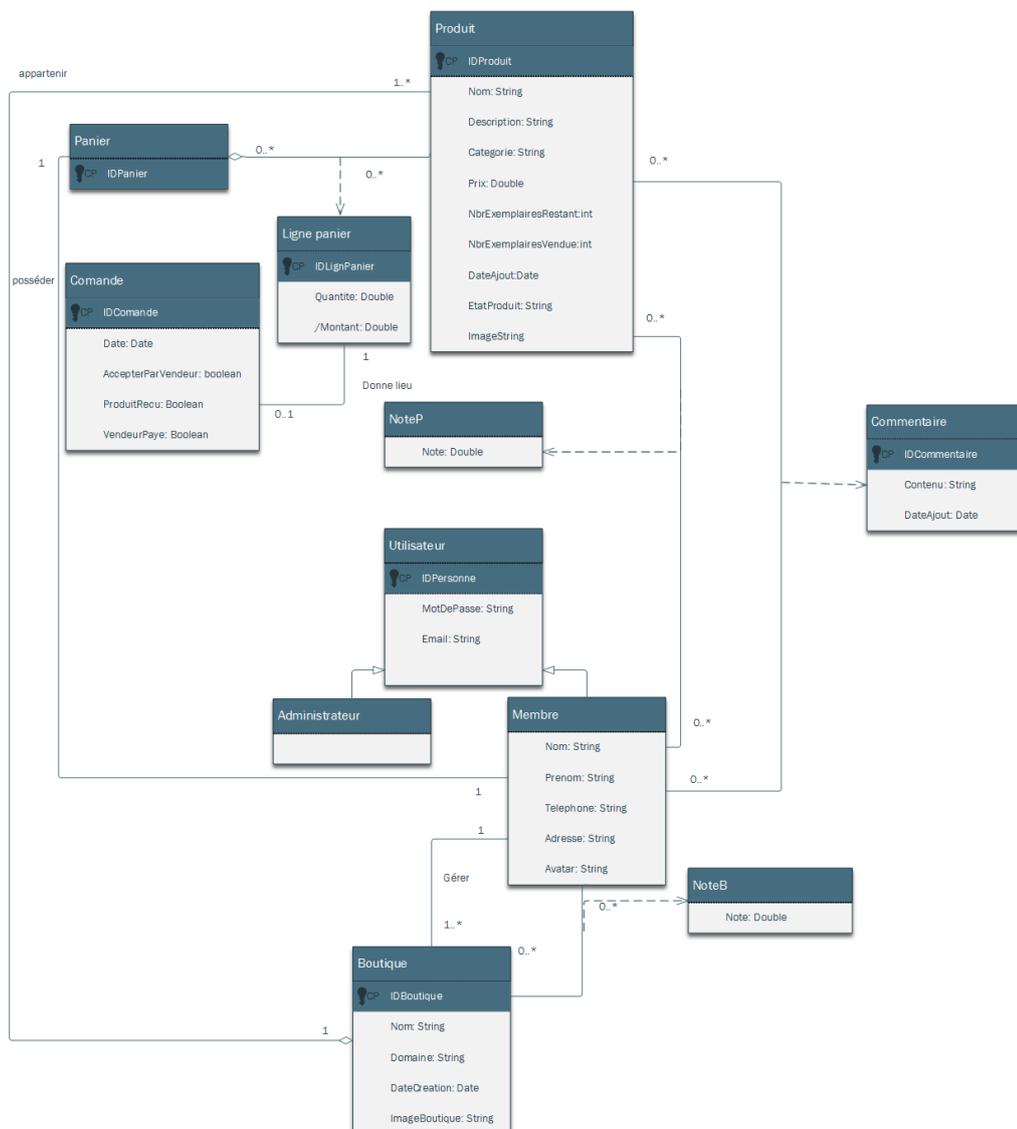


FIGURE 3.16 – Diagramme de classe

3.4 Dictionnaire de données

Attributs	Signification	Type	Taille
Utilisateur			
IDUtilisateur	Identifiant de l'utilisateur	Entier	3
MotDePasse	Mot de passe de l'utilisateur (membre, administrateur)	Alphanumérique	56
Email	Boite Email de l'utilisateur	Alphanumérique	45
NoteB			
Note	Valeur de la note attribuée	Réel	
Commentaire			
IDCommentaire	Identifiant du commentaire	Entier	6
Contenu	Contenu du commentaire	Alphanumérique	
DateAjout	Date d'ajout du commentaire	DATE	
Produit			
IDProduit	Identifiant du produit	Entier	6
Nom	Nom du produit	Alphanumérique	150
Description	Description du produit	Alphanumérique	
Catégorie	Catégorie du produit	Alphanumérique	20
Prix	Prix du produit	Réel	
NbrExemplaire Restant	Nombre d'exemplaire disponible du produit	Entier	3
NbrExemplaire Vendu	Nombre d'exemplaire vendu	Entier	3
DateAjout	Date d'ajout du produit	DATE	
Image	Image descriptive du produit	Alphanumérique	100
EtatProduit	Indique l'état de produit	Alphanumérique	19
Panier			
IDPanier	Identifiant du panier	Entier	6

Membre			
Nom	Nom du membre	Alphanumérique	45
Prenom	Prénom du membre	Alphanumérique	45
Telephone	Numéro de téléphone du membre	Alphanumérique	10
Adresse	L'adresse du membre	Alphanumérique	255
Avatar	Le nom de la photo de profil	Alphanumérique	100
Boutique			
IDBoutique	Identifiant de la boutique	Entier	4
Nom	Nom de la boutique	Alphanumérique	150
Domaine	Domaine d'activité de la boutique	Alphanumérique	150
DateCreation	Date de création de la boutique	DATE	
ImageBoutique	Photo de couverture de la boutique	Alphanumérique	100
Ligne Panier			
IdLignePanier	Identifiant de la ligne de panier	Entier	6
Quantité	Nombre d'exemplaire souhaité	Entier	3
/Mentant	Mentant total de la ligne	Réel	
Commande			
IDCommande	Identifiant de la commande	Entier	6
Date	Date de la commande	DATE	
AcceptePar Vendeur	Indique si le vendeur a accepté de vendre son produit ou pas.	Booléen	1
ProduitRecu	Indique si le produit est reçu par le client ou pas.	Booléen	1
VendeurPaye	Indique si le système a transféré l'argent au vendeur ou pas.	Booléen	1
NoteP			
Note	Valeur de la note attribuée.	Réel	

3.5 Model relationnel

- **Membre** (IDUtilisateur, MotDePasse, Email Nom, Prénom, Telephone, Adresse, Avatar)

- **Administrateur** (IDUtilisateur, MotDePasse, Email)
- **Boutique** (IDBoutique, Nom, Domaine, DateCreation, ImageBoutique, #IDUtilisateur)
- **Produit** (IDProduit, Nom, Description, Categorie, Prix, NbrExemplairesRestant, NbrExemplairesVendue DateAjout, EtatProduit, Image, #IDBoutique)
- **Commentaire** (IDCommentaire, #IDProduit, #IDUtilisateur, Contenu, DateAjout)
- **Panier** (IDPanier, #IDUtilisateur, /Total)
- **Lignepanier** (IDLignPanier, #IDProduit, #IDPanier, Quantite, /Mentant)
- **Commande** (IDCommande, Date, AccepterParVendeur, ProduitRecu, VendeurPaye, #IDLignPanier)
- **NoteP** (#IDProduit, #IDUtilisateur, Note)
- **NoteB** (#IDProduit, #IDUtilisateur, Note)

Conclusion

Au cours de ce chapitre, nous avons présenté l'étape d'analyse, qui nous a permis de réaliser des diagrammes de séquences associés à chaque cas d'utilisation, destinés à mieux les expliquer en tenant compte de toutes leurs alternatives. Ajouter à cela des scénarios de déroulements représentés sous forme de diagramme d'activité. Un diagramme de classe est aussi représenté afin de définir la structure objet de notre projet. Dans le prochain chapitre nous allons entamer la partie réalisation où est présentée l'application finale.

4

Réalisation

Introduction

Dans le chapitre suivant, nous présentons la partie réalisation qui représente la conclusion finale de notre projet, dans ce qui suit nous décrirons l'environnement ainsi que les outils qui nous ont servis pour la réalisation de l'application, nous terminerons enfin par la présentation de l'application.

4.1 Description de l'environnement de développement

4.1.1 Java entreprise édition

Le terme « Java EE » anciennement raccourci en « J2EE » signifie Java Enterprise Edition, il fait référence à une extension de la plate-forme standard. Autrement dit, la plate-forme Java EE est construite sur le langage Java et la plate-forme Java SE à quoi s'ajoute un grand nombre de bibliothèques remplissant différentes fonctionnalités que la

plate-forme standard ne remplit pas d'origine. L'objectif majeur de Java EE est de faciliter le développement d'applications web robustes et distribuées, déployées et exécutées sur un serveur d'applications.

Java EE regroupe un ensemble d'API chacun est destiné à faire un traitement spécifique. Dans le cadre de notre application, nous avons choisi d'utiliser les API suivants :

- **Servlet** : Une servlet est une classe Java, qui a la particularité de permettre le traitement de requêtes et la personnalisation de réponses.
- **JSP (Java Server Pages)** : Les pages JSP sont une des technologies de la plate-forme Java EE les plus puissantes, simples à utiliser et à mettre en place. Elles se présentent sous la forme d'un simple fichier au format texte, contenant des balises respectant une syntaxe à part entière. Le langage JSP combine à la fois les technologies HTML, XML, servlet et JavaBeans en une seule solution permettant aux développeurs de créer des vues dynamiques.
- **JDBC (Java DataBase Connectivity)** : C'est une API qui fait partie intégrante de la plate-forme Java, et qui est constituée de classes permettant l'accès depuis vos applications Java à des données rangées sous forme de tables. Dans la très grande majorité des cas, il s'agira bien entendu de bases de données stockées dans un SGBD.

JavaBean : Le JavaBean souvent raccourci en "bean", désigne tout simplement un composant réutilisable. En réalité, un bean est un simple objet Java qui suit certaines contraintes, et représente généralement des données du monde réel. Un bean n'a donc rien de spécifique au Java EE.

Java EE permet de coder son application comme on le souhaite, néanmoins, il est recommandé d'adopter une architecture plus ou moins standard, que tout développeur peut reconnaître, c'est-à-dire dans laquelle tout développeur sait se repérer.[15]

Dans le cadre de notre application, nous avons opté pour le design pattern MVC et DAO.

Le design pattern MVC (Model, Vue, Contrôle)

MVC est un patron de conception (design pattern en anglais) très répandu pour réaliser des sites web. Ce patron de conception est une solution éprouvée et reconnue permettant

de séparer l'affichage des informations, les actions de l'utilisateur et l'accès aux données. Comme son nom l'indique, le design pattern MVC est composé de trois parties :

- Tout ce qui concerne le traitement, le stockage et la mise à jour des données de l'application doit être contenu dans la couche nommée "Modèle" (le M de MVC) ;
- Tout ce qui concerne l'interaction avec l'utilisateur et la présentation des données (mise en forme, affichage) doit être contenu dans la couche nommée "Vue" (le V de MVC) ;
- Tout ce qui concerne le contrôle des actions de l'utilisateur et des données doit être contenu dans la couche nommée "Contrôle" (le C de MVC).

En pratique, le modèle correspond aux objets java, la vue correspond aux pages JSP, et le contrôleur correspond aux Servlet.[15]

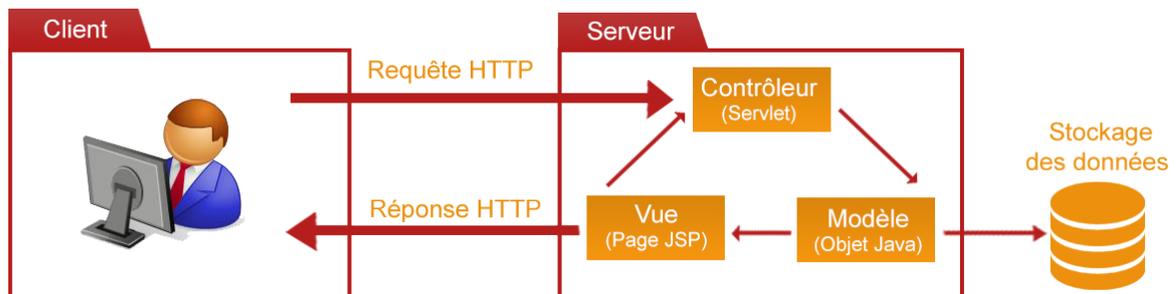


FIGURE 4.1 – Le design pattern MVC)

Le design pattern DAO

Le principe du pattern DAO est de séparer la couche modèle d'une application en deux sous-couches distinctes :

- Une couche gérant les traitements métier appliqués aux données, souvent nommée couche service ou métier.
- Une couche gérant le stockage des données, logiquement nommée couche de données. Il s'agit là des opérations classiques de stockage : la création, la lecture, la modification et la suppression. Ces quatre tâches basiques sont souvent raccourcies à l'anglaise en CRUD.[15]

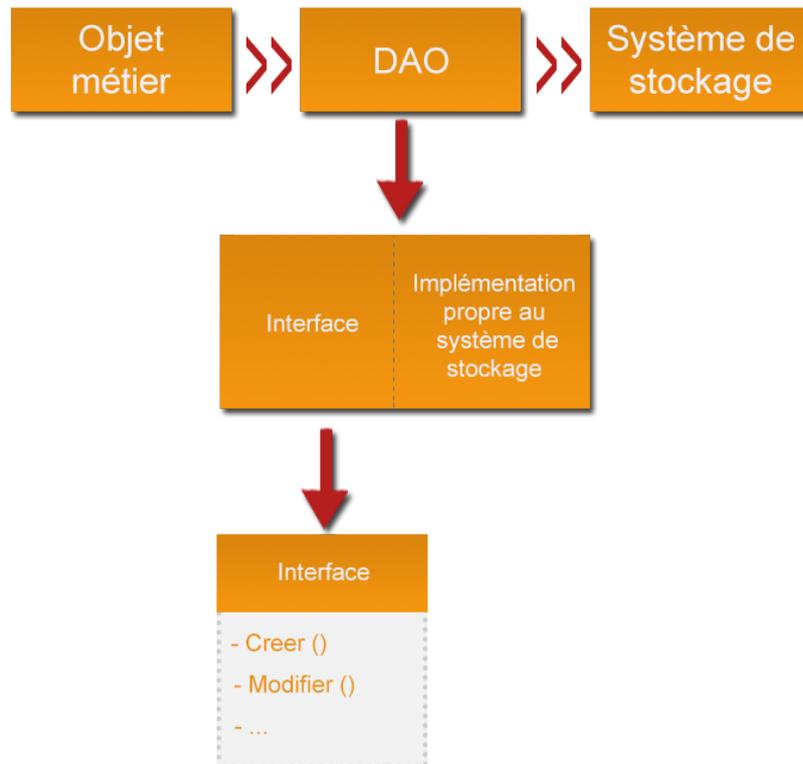


FIGURE 4.2 – Le design pattern DAO)

4.1.2 Le serveur Apache-Tomcat

Apache-Tomcat est le serveur d'application Java du projet Jakarta de la fondation Apache. Ce serveur libre, sous licence Apache permet d'exécuter des applications web développées avec les technologies java (Servlet, JSP ...).[16]

4.1.3 MySQL (My Structured Query Language)

MySQL est un serveur de bases de données relationnelles Open Source. Un serveur de bases de données stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de l'ensemble. Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête.[17]

4.1.4 HTML 5

C'est le langage universel utilisé sur les pages Web lisibles par tous les navigateurs Web (Internet Explorer, Google chrome, Mozilla, etc.). Ce langage fonctionne suivant l'assemblage et la combinaison de balises permettant de structurer et donner l'apparence voulue aux données textes, images et multimédias suivant la mise en page voulue.[23]

4.1.5 CSS 3

Le terme CSS est l'acronyme anglais de Cascading Style Sheets qui peut se traduire par "feuilles de style en cascade". Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML.[21]

4.1.6 JavaScript

Le JavaScript est un langage informatique utilisé sur les pages web. Le langage a été créé en 1995 par Brendan Eich pour le compte de Netscape Communications Corporation, Ce langage a la particularité de s'activer sur le poste client, en d'autres mots c'est votre ordinateur qui va recevoir le code et qui devra l'exécuter. C'est en opposition à d'autres langages qui sont activés côté serveur. L'exécution du code est effectuée par votre navigateur internet tel que Firefox ou Internet Explorer.

La particularité du JavaScript consiste à créer des petits scripts sur une page HTML dans le but d'ajouter une petite animation ou un effet particulier sur la page. Cela permet en général d'améliorer l'ergonomie ou l'interface utilisateur, mais certains scripts sont peu utiles et servent surtout à ajouter un effet esthétique à la page. L'intérêt du JavaScript est d'exécuter un code sans avoir à recharger une nouvelle fois la page.[22]

4.1.7 Bootstrap

kit CSS créé par les développeurs de Twitter, il permet de construire rapidement et facilement des sites web esthétiques et responsives. Bootstrap offre aussi des plugins jQuery de qualité pour enrichir les pages du site.[18]

4.1.8 Sandbox

Sandbox est un environnement utilisé par les développeurs dans la phase de test, il permet de simuler les différents services de PayPal grâce à des comptes virtuels.[19]

4.1.9 L'IDE Eclipse

Définition d'un IDE

Un IDE (environnement de développement intégré) est un outil qui a pour objectif de faciliter le développement sous un ensemble restreint de langages. Il contient un ensemble de fonctionnalités permettant de faciliter la tâche sous programmeur. Les principales fonctionnalités d'un IDE sont :

- Un éditeur de texte intelligent avec coloration des mots clés.
- Un compilateur permettant d'exécuter rapidement le programme.
- Un débogueur permettant de traquer les erreurs du programme.

Eclipse

Est un IDE libre, extensible et polyvalent développé en JAVA. Il permet de créer des projets de développement sous n'importe quel langage de programmation. Eclipse est principalement développé autour de la notion de plug-in. Ceci permet à ses utilisateurs de l'adapter selon leurs besoins.[20]

4.2 Organigramme de l'application

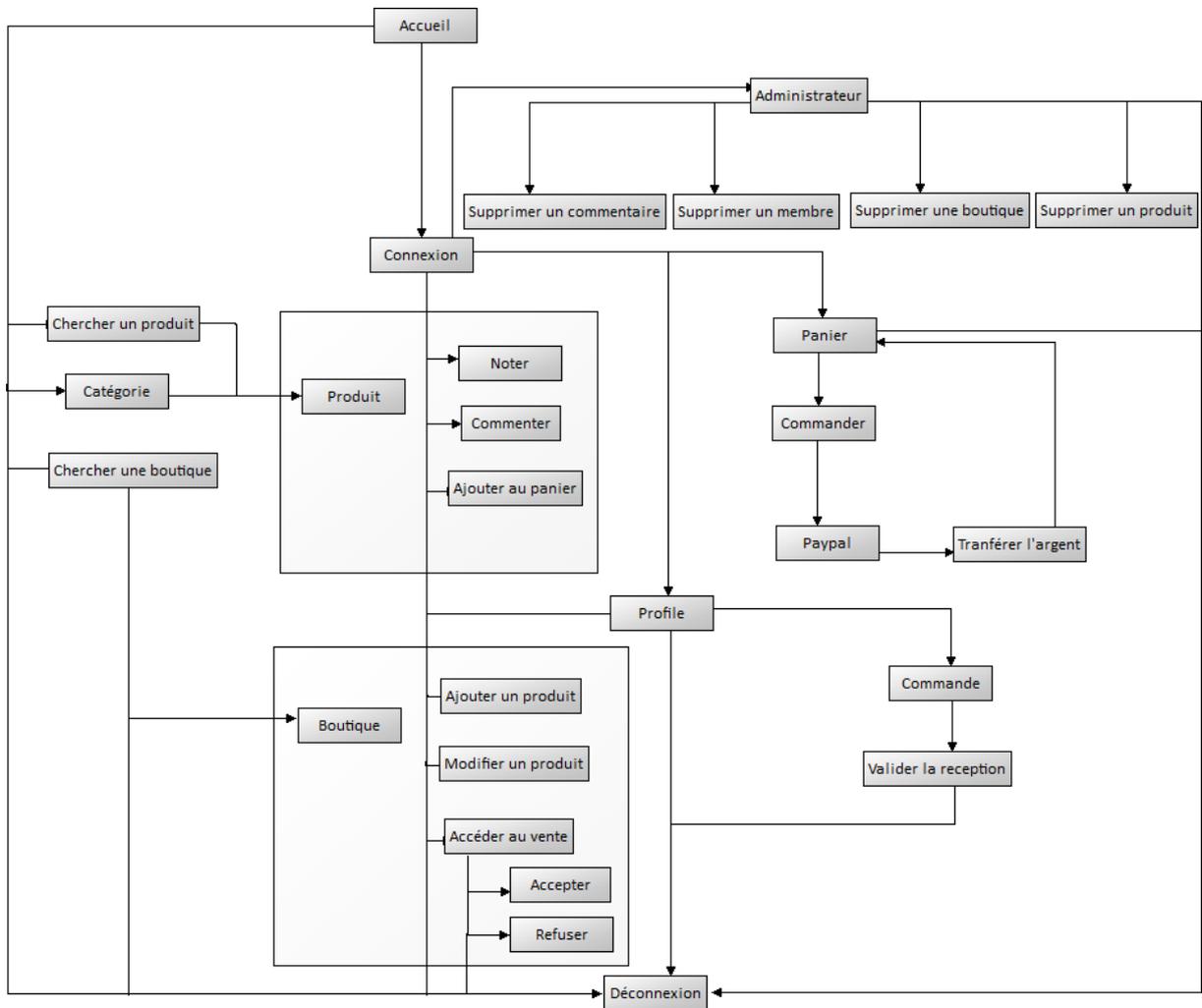


FIGURE 4.3 – Organigramme de l'application

4.3 Description de l'application

4.3.1 L'index (page d'accueil)

L'index représente la vitrine du site, il s'agit de la première page qui apparaît à l'utilisateur, de ce fait elle est censée résumer le principe de base de l'application elle joue ainsi un rôle de portail dans l'application.

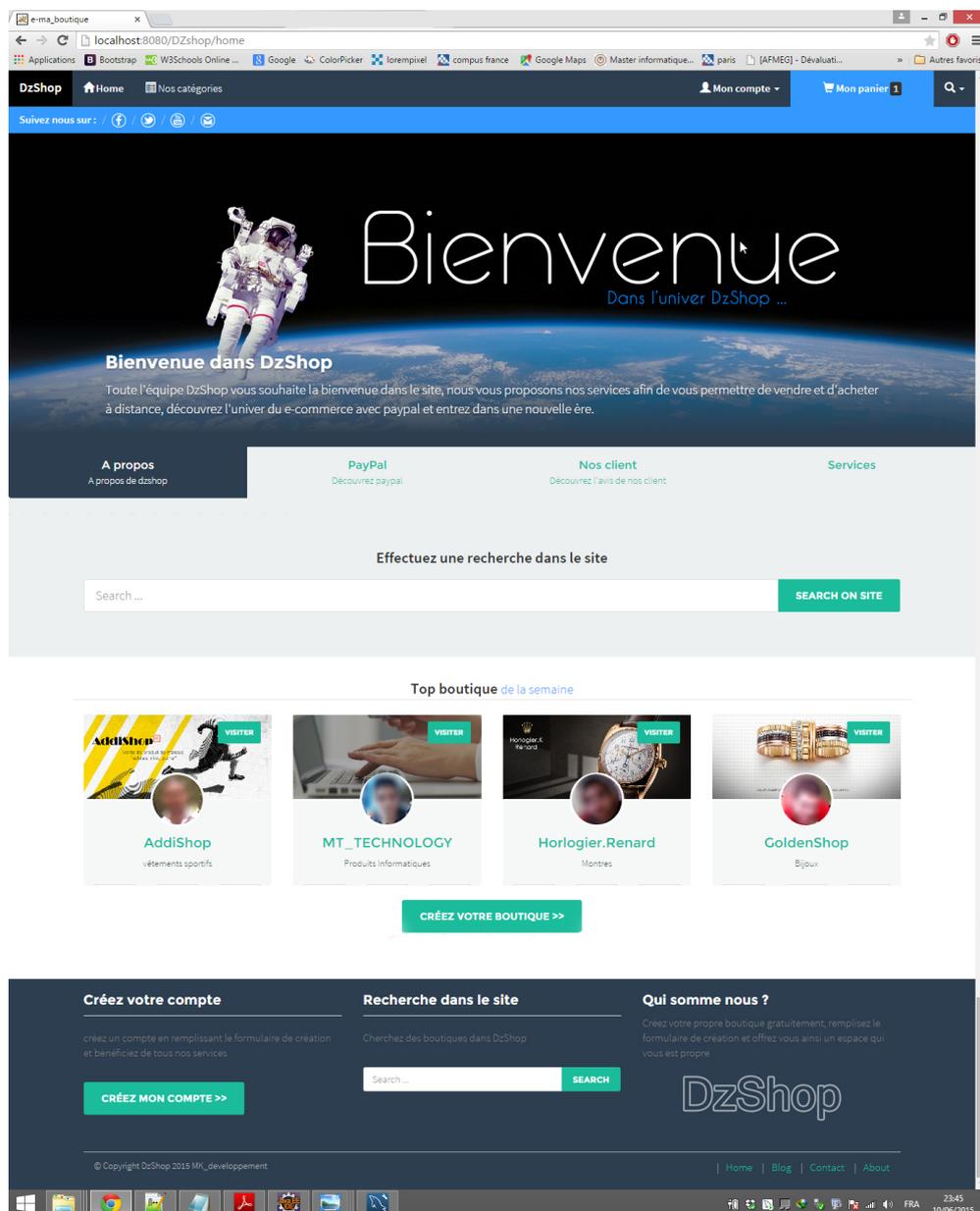


FIGURE 4.4 – Page d'accueil

4.3.2 Produit

L'utilisateur dans cette page retrouve le produit qu'il aura sélectionné, toutes les informations relatives à ce produit y sont relatées, il retrouve sa description, sa note ainsi que l'intégralité des commentaires des autres utilisateurs, dans cette page l'utilisateur est en mesure d'ajouter le produit à son panier d'achat ou de le modifier s'il s'agit du sien.

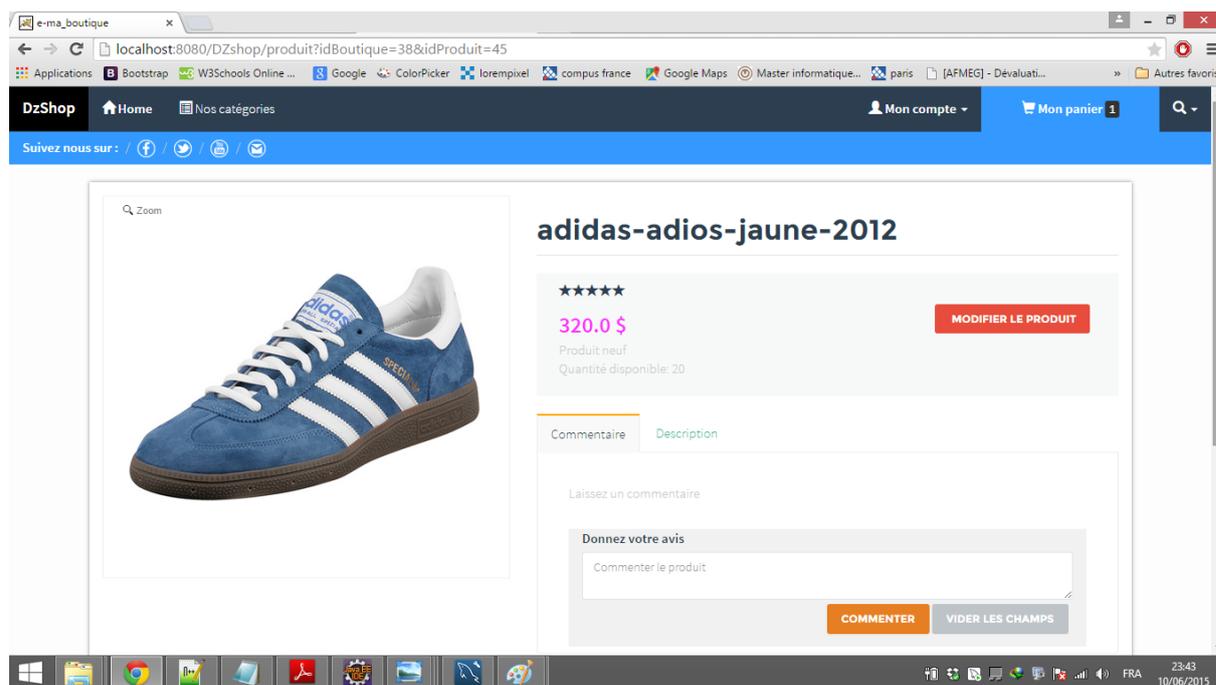


FIGURE 4.5 – Page Produit

4.3.3 La boutique

Une boutique représente un espace de vente spécifique pour un utilisateur elle ne regroupe ainsi que les produits propres à ce dernier, l'utilisateur est en mesure de gérer sa boutique comme il le souhaite, il lui est possible d'ajouter, modifier ou supprimer les produits qui y sont enregistrés. Il peut aussi accepter ou refuser une commande d'un client.

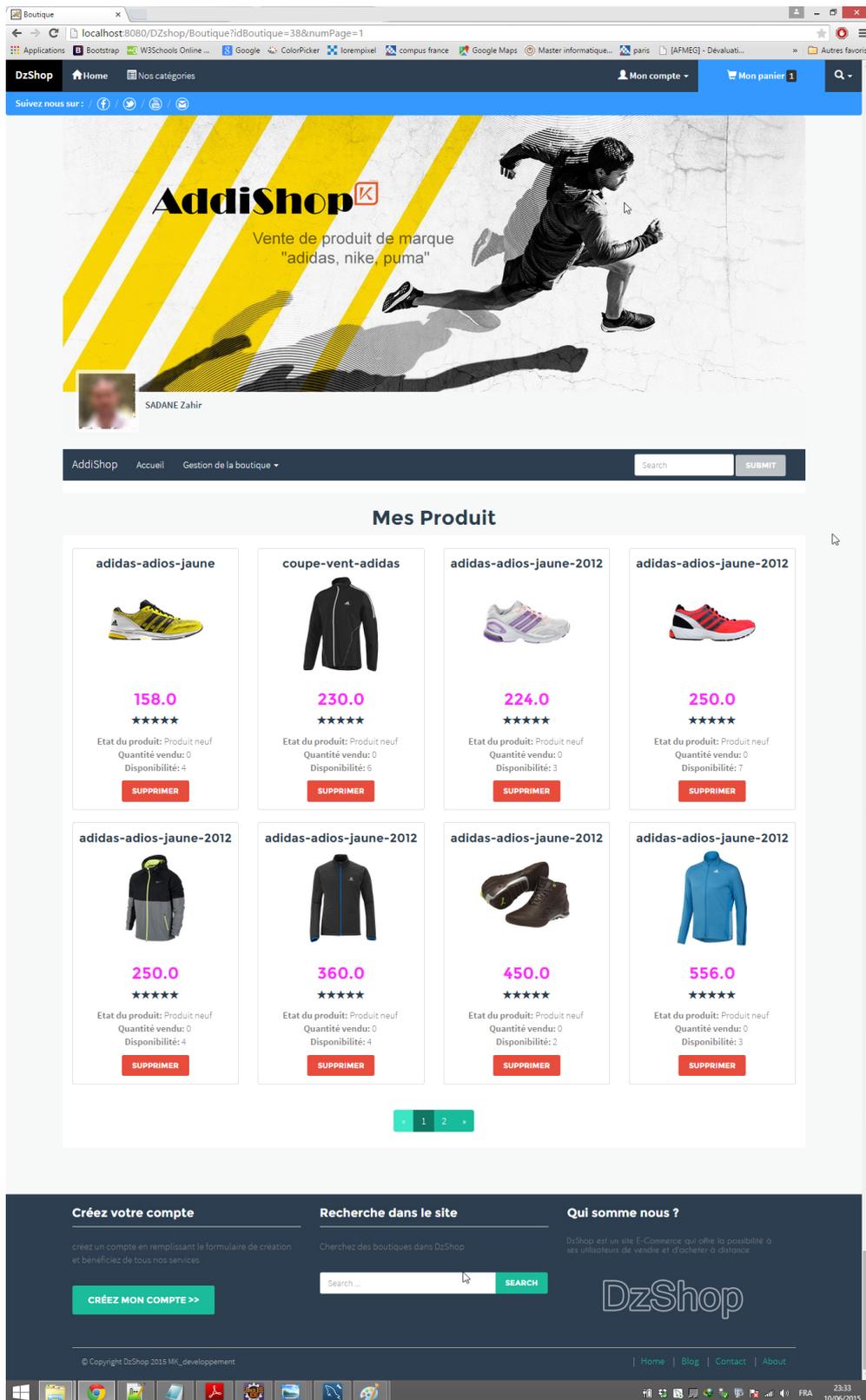


FIGURE 4.6 – Page boutique

4.3.4 Le panier

Le panier d'un utilisateur contient l'ensemble des produits qu'il aura sélectionnés préalablement, il est représenté sous la forme d'un tableau où chaque ligne représente un produit. Lors d'une commande, un utilisateur cible les lignes du panier et non le panier dans son intégralité.

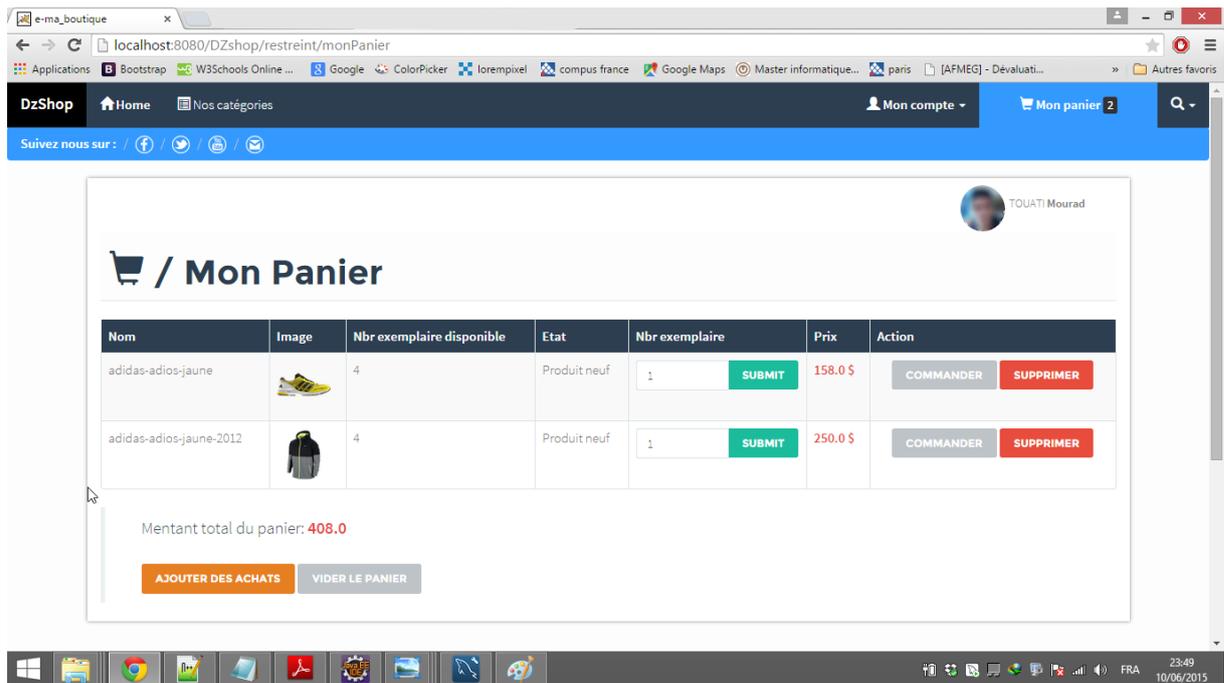


FIGURE 4.7 – Page panier

4.3.5 Le profil

La page profile regroupe l'ensemble des informations propres à un utilisateur et principalement la liste de ses boutiques ainsi que celle de ses commandes, c'est d'ailleurs dans cette dernière que sont finaliser les commandes (validation de la réception d'un produit, annulation de la commande, réclamation ... etc.).

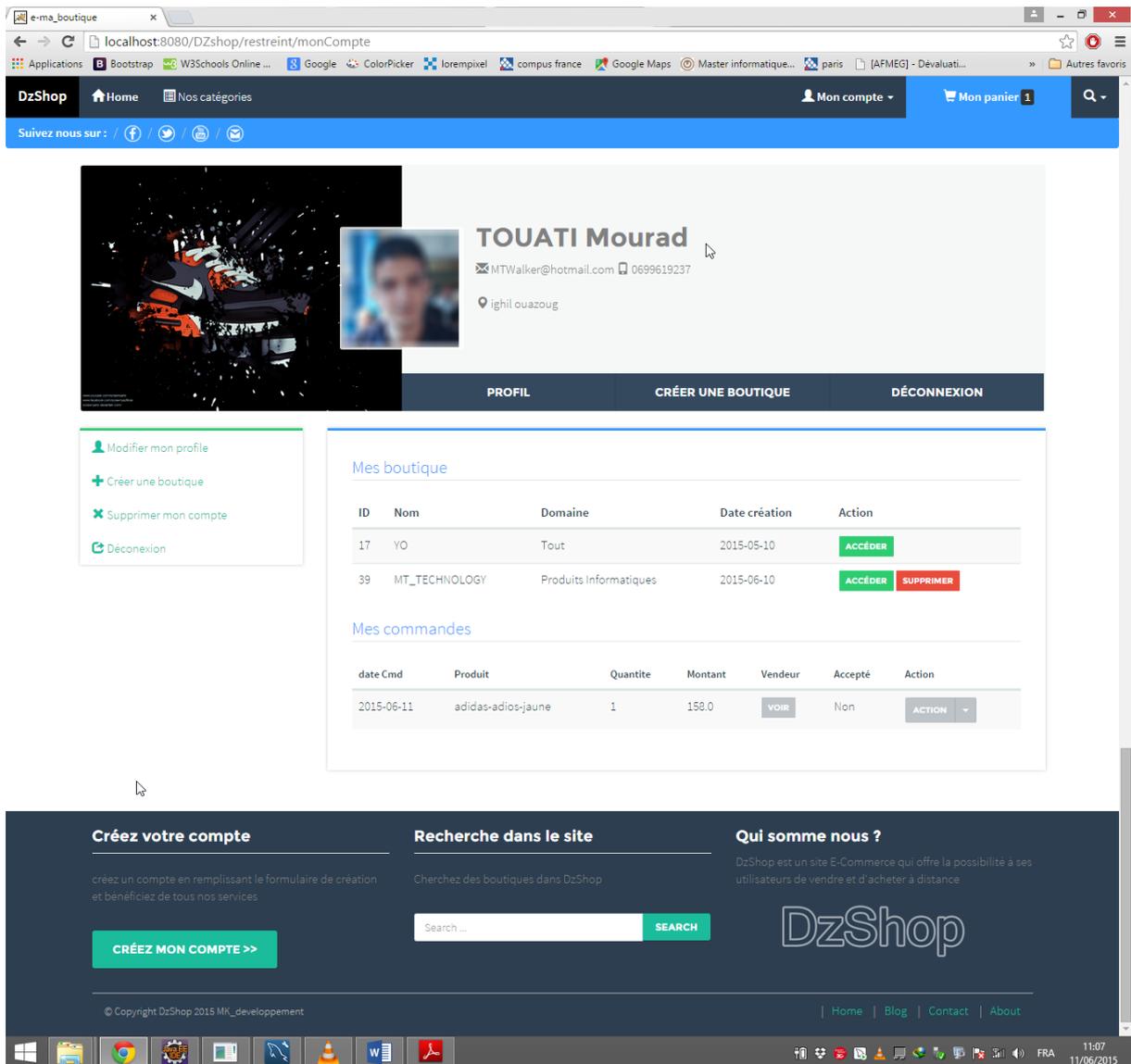
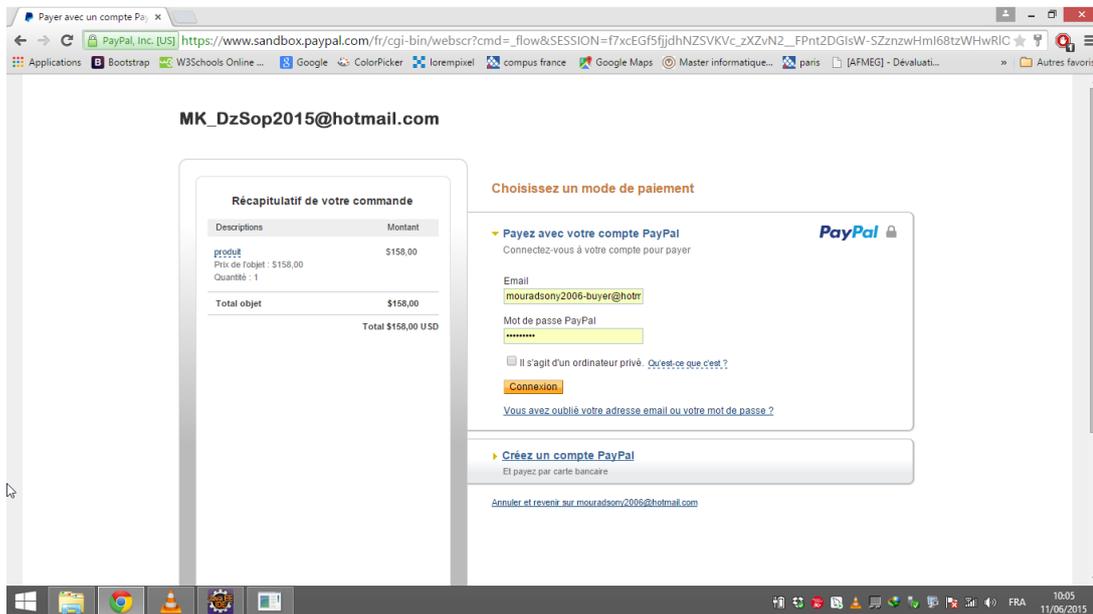


FIGURE 4.8 – Page profil

4.3.6 PayPal

Après la commande l'utilisateur est renvoyé vers une page paypal pour effectuer la transaction en entrant ses informations personnelles. À la fin il est redirigé vers dzshop où s'affiche un message de confirmation selon le déroulement de la transaction.

Page d'authentification paypal



transaction réussi, retour vers le système

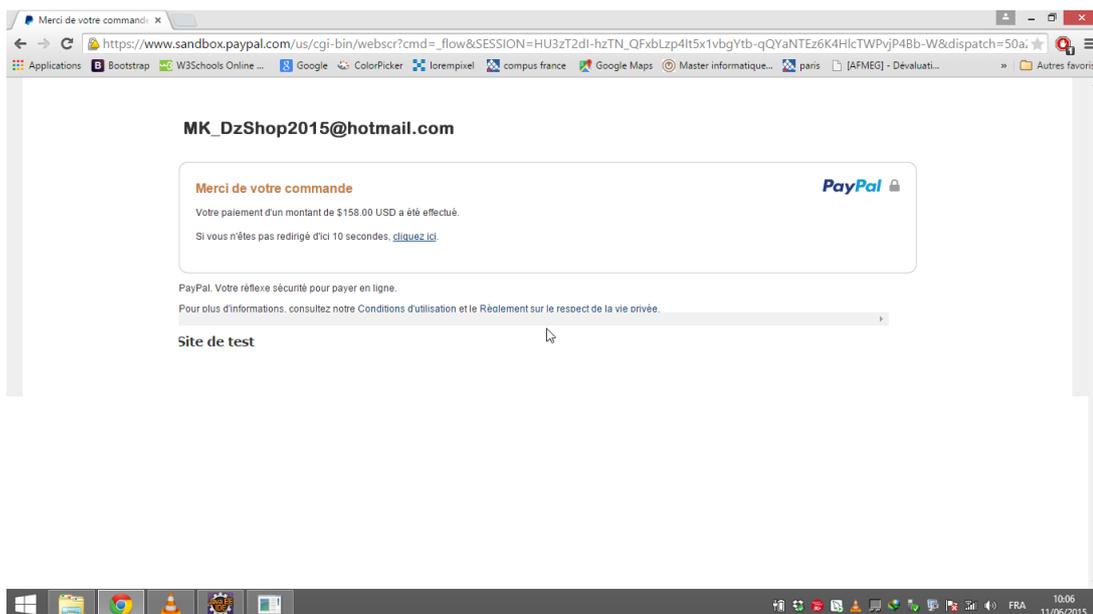


FIGURE 4.9 – Pages PayPal

4.3.7 L'administrateur

L'administrateur, une fois connecté est connecté vers cette page où sont affichés tous les comptes des membres inscrits, il peut ainsi supprimer le compte de n'importe quel utilisateur, il est aussi en mesure d'effectuer la même opération pour le reste des pages de la liste de gauche.

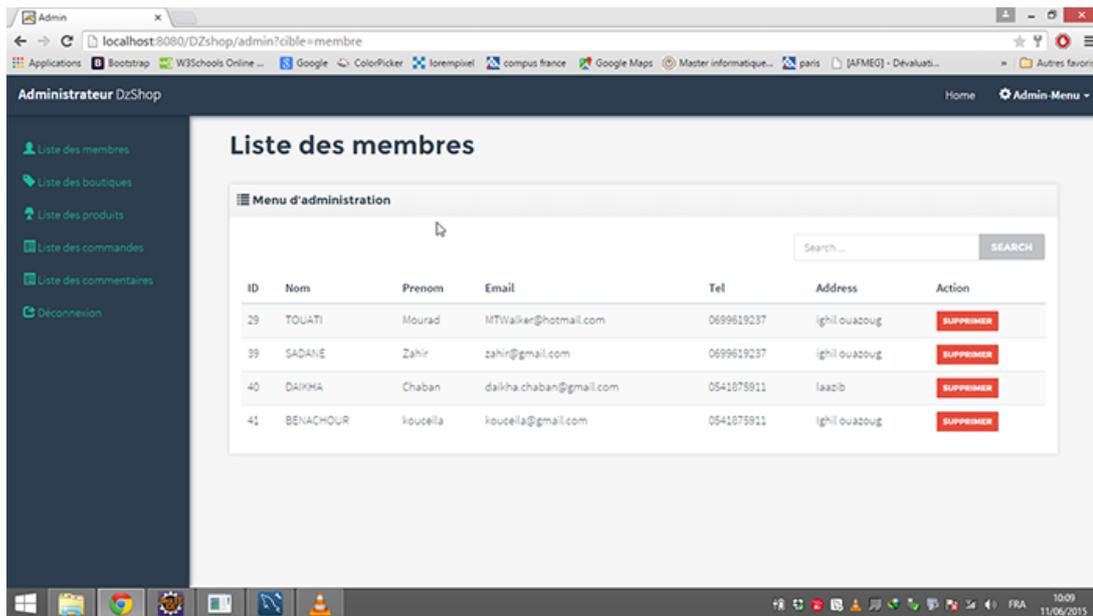


FIGURE 4.10 – Page administrateur

Conclusion

Ce dernier chapitre représente la conclusion de notre projet, dans ce chapitre nous avons décrit la partie réalisation où sont présentées les outils de développement ainsi que l'environnement de travail utilisé pour développer l'application. Nous sommes passés par la suite à la présentation de l'application avec des définitions et explications des différentes pages ainsi que des images représentatives de ces dernières.

Conclusion Générale

Tout au long de ce travail nous avons décrit ce qu'est le commerce électronique avec ces différents avantages et inconvénients, de ce qui a précédé, nous avons relevé une problématique qui a mis le point sur l'importance du E-Commerce, autant pour les commercial que pour les consommateurs.

L'application que nous avons réalisée a été conçue et modéliser de manière à ce qu'elle soit accessible pour tous, ergonomique, conviviale, simple d'utilisation sont autant de qualités consacré à faciliter la navigation des différents utilisateurs.

Pour la réalisation de l'application, nous avons eu recours à différents langages et outils de développement et de programmation telle que « Java EE » pour la mise en place des différents traitements et fonctionnalités de l'application ainsi qu'HTML, CSS, JavaScript et Bootstrap pour la réalisation de son aspect graphique, sans oublier MySql comme système de gestion de base de données.

Ce projet en plus de nous avoir permis d'assimiler certaines connaissances sur le e-commerce et la programmation orienté objet et nous a permis de suivre une méthode de conception pour la réalisation d'un projet en partant d'une idée pour arriver à un résultat final.

Perspectives

Le projet que nous avons réalisé au cours de ce mémoire a pour objectif d'instaurer en plus du paiement en ligne, une relation de confiance entre les consommateurs. De ce point

de vue, notre projet doit encore évoluer, pour cette raison, nous envisageons plusieurs perspectives dont :

- Offrir la possibilité aux consommateurs, qu'ils soient clients ou vendeurs de pouvoir effectuer des réclamations en cas de désagréments de toutes sortes.
- Elargir les possibilités de personnalisation des boutiques.
- Proposer des services de livraison afin de contrôler le transport des produits.
- Offrir une possibilité aux consommateurs d'effectuer leurs achats autrement qu'en ayant recours à paypal (carte bancaire, CCP ...etc.).

Bibliographie

- 1 : Définition du E-commerce : 2015, <http://www.definitions-marketing.com/Definition-E-commerce>.
- 2 : E-Commerce Overview : 2015, http://www.tutorialspoint.com/e_commerce/e_commerce_quick_guide.html
- 3 : Commerce électronique : ce que disent les chiffres et ce qu'il faudrait savoir : 2015, http://www.persee.fr/web/revues/home/prescript/article/estat_0336454_2000_num_339_1_7483
- 4 : Acheter entre particulier en toutes sécurités : 2015, <http://www.terrafemina.com/vie-pratique/finances-perso/outils/1666-acheter-entre-particuliers-en-toute-securite.html>
- 5 : PayPal : 2015, <http://www.definition-marketing.net/paypal/>.
- 6 : <https://www.paypal.com/fr/home>.
- 7 : <http://openclassrooms.com/courses/debutez-l-analyse-logicielle-avec-uml> .
- 8 : P.ROQUES, F.VALLÉE UML en action De l'analyse des besoins à la conception en Java. Eyrolles, Paris, France, 2003.
- 9 : Diagramme de contexte : 2015, <http://www.er.uqam.ca/nobel/r20014/ManTic/Diagrammedecontextedunsystmeorganisation.html>
- 10 : C. Soutou UML pour les bases de donn. Eyrolles Paris, France, 2012.
- 11 : La réalisation d'une application de contrôle total des processus d'un ordinateur distant : 2015, <http://www.memoireonline.com/05/12/5812/La-realisation-dune-application-de-contrle-total-des-processus-dun-ordinateur-distant.html>
- 12 <http://msdn.microsoft.com/fr-fr/library/vstudio/dd409377.aspx>.
- 13 <http://msdn.microsoft.com/fr-fr/library/vstudio/dd409377.aspx>.
- 14 P.ROQUES, F.VALL UML en action De l'analyse des besoins conception en Java. Eyrolles, Paris, France, 2003.
- 15 Créez votre application web avec java EE : 2015, <http://openclassrooms.com/>

- courses/creez-votre-application-web-avec-java-ee.
- 16 J.Lafosse Java EE : guide de développement d'applications web en Java. Edition ENI, 2009.
 - 17 Conception et développement d'une application mobile de vente flash sous android : 2015, http://www.memoireonline.com/11/13/7773/m_Conception-et-developpement-d-une-application-mobile-de-vente-flash-sous-android.html
 - 18 Prenez en main Bootstrap : 2015, <http://openclassrooms.com/courses/prenez-en-main-bootstrap>.
 - 19 https://developer.paypal.com/docs/classic/lifecycle/s_aboutaccounts/.
 - 20 apprenez à programmer en java : 2015, <http://openclassrooms.com/courses/apprenez-a-programmer-en-java>.
 - 21 CSS : 2015, <http://glossaire.infowebmaster.fr/css/>
 - 22 INTRODUCTION AU JAVASCRIPT : 2015, [HTTP://OPENCLASSROOMS.COM/COURSES/DYNAMISEZ-VOS-SITES-WEB-AVEC-JAVASCRIPT/INTRODUCTION-AU-JAVASCRIPT](http://OPENCLASSROOMS.COM/COURSES/DYNAMISEZ-VOS-SITES-WEB-AVEC-JAVASCRIPT/INTRODUCTION-AU-JAVASCRIPT)
 - 23 : HTML : 2015, <http://www.clashinfo.com/dico/definition-h/art45-html.html>



Annexe A

A.1 Explication des diagrammes UML utilisés

A.1.1 Diagramme de cas d'utilisation

Relation entre cas d'utilisation

Trois types de relations standards entre cas d'utilisations sont proposés par UML :

- **Include** : C'est la relation d'utilisation qui permet d'inclure la fonctionnalité d'un cas d'utilisation dans un autre. Ce qui permet d'effectuer une factorisation et donc de réduire la duplication de fonctionnalité. Elle se matérialise par une flèche en pointillés sur laquelle est écrit « include » et qui est orientée vers le use case factorisé.
- **Extend** : La relation d'extension indique qu'un cas d'utilisation est une variation d'un autre : l'un complète l'autre. On y a recours lors de circonstances exceptionnelles. Elle se matérialise par une flèche en pointillés sur laquelle est écrit « extend » et qui est orientée vers le cas d'utilisation de base.

- **Généralisation** : la généralisation indique qu'un cas d'utilisation est un cas particulier d'un autre cas.

A.1.2 Diagramme de séquence

Les concepts de base utilisés dans le diagramme de séquence sont les suivant :

- **Objet** : Un objet est une instance de la classe ou de l'interface responsable de la réalisation des opérations définies. Il est représenté graphiquement dans le diagramme de séquence par un carrée comportant son identifiant.

- **Ligne de vie** : Une ligne de vie représente l'ensemble des opérations exécutées par un objet. Un message reçu par un objet déclenche l'exécution d'une opération. Le retour d'information peut être implicite (cas général) ou explicite à l'aide d'un message de retour.

- **Message synchrone et asynchrone** : Dans un diagramme de séquence, deux types de message peuvent être distingués :

- **Message synchrone** : dans ce cas l'émetteur reste en attente de la réponse à son message avant de poursuivre ses actions. La flèche avec extrémité pleine symbolise ce type de message.

- **Message asynchrone** : dans ce cas, l'émetteur n'attend pas la réponse à son message, il poursuit l'exécution de ses opérations. C'est une flèche avec une extrémité non pleine qui symbolise ce type de message.

- **Fragment d'interaction** : Un fragment d'interaction correspond à un ensemble d'interactions auxquels on applique un opérateur, il se représente globalement comme un diagramme de séquence dans un rectangle avec indication dans le coin à gauche du nom de l'opérateur. Nous décrivons dans ce qui suit les opérateurs que nous utilisons pour l'analyse de notre application : alt, opt, loop et ref.

- **L'opérateur alt** : L'opérateur alt correspond à une instruction de test avec une ou plusieurs alternatives possibles. Il est aussi permis d'utiliser les clauses de type sinon. L'opérateur alt se représente dans un fragment possédant au moins deux parties séparées par des pointillés.

- **L'opérateur opt** : L'opérateur opt (optionnel) correspond à une instruction de test sans alternative (sinon). L'opérateur opt se représente dans un fragment possédant une seule partie.
- **L'opérateur loop** : L'opérateur loop correspond à une instruction de boucle qui permet d'exécuter une séquence d'interaction tant qu'une condition est satisfaite. L'opérateur loop se représente dans un fragment possédant une seule partie et englobant toutes les interactions faisant partie de la boucle.
- **L'opérateur ref** : L'opérateur ref permet d'appeler une séquence d'interactions décrite par ailleurs constituant ainsi une sorte de sous-diagramme de séquence.

A.1.3 Diagramme d'activité

Nœud d'action

Une action est un nœud d'activité exécutable. L'exécution d'une action peut être une transformation ou un calcul dans le système modélisé (affectation de valeur à des attributs, création d'un nouvel objet, calcul arithmétique, émission ou réception d'un signal, . . .)

Nœud de contrôle

Un nœud de contrôle est un nœud d'activités abstrait utilisé pour coordonner les flux entre les nœuds d'une activité. Nous citons ci-dessous types de nœud de contrôle que nous avons utilisé :

- Nœud initial (initial node) : Un nœud initial est un nœud de contrôle à partir duquel le flot débute.
- Nœud final (final node) : Un nœud final est un nœud de contrôle dans lequel le flux d'activité s'arrête.
- Nœud de décision (décision node) : Un nœud de décision est un nœud de contrôle, il permet de faire un choix entre plusieurs flux sortants. Les flux sortants sont sélectionnés en fonction de la condition de garde qui est associée à chaque arc sortant.
- Nœud de bifurcation (fork node) : Un nœud de bifurcation est un nœud de contrôle qui sépare un flux d'entrée en plusieurs flots concurrents en sortie.

A.2 Modèle relationnel

Le concepteur d'une BD relationnelle doit élaborer ce qu'il est convenu d'appeler le schéma relationnel de la base de données. Cette activité consiste à définir toutes les relations normalisées de la BD et les domaines de leurs attributs. Théoriquement cela consisterait à décrire par intention chaque relation, définir les domaines de chaque attribut de la relation. Les relations du schéma doivent toutes posséder les propriétés suivantes :

1. Une relation a un nom distinct de toutes les autres du même schéma.
2. Chaque attribut d'une relation ne peut recevoir qu'une seule valeur atomique (type de données simple).
3. Chaque attribut a un nom distinct.
4. Toutes les valeurs d'un attribut font parties du même domaine, même type de données et même contraintes d'intégrité.
5. Chaque tuple de la relation est distinct, pas de tuple en double.
6. L'ordre des tuples n'a pas d'importance.
7. L'ordre des attributs n'a pas d'importance.

A.2.1 Terminologie de l'approche relationnelle

- a) **Domaine** : Ensemble des valeurs admises pour un attribut, Il établit les valeurs acceptables dans une colonne.
- b) **Tuple** : Une ligne dans une table.
- c) **Clé primaire** : Ensemble minimal de colonnes permettant d'identifier de manière unique chaque tuple dans une table.
- d) **Clé étrangère** : Une ou plusieurs colonnes dans une table qui a pour but d'assurer une liaison entre deux tables.

A.2.2 Règles de dérivation du modèle relationnel à partir d'un modèle de classes

Les règles utilisées pour le passage du diagramme de classes de notre site au modèle relationnel, sont tirées de :

- a) **Règle 1 : Transformation des classes** Chaque entité devient une relation. L'identifiant de l'entité devient clé primaire pour la relation.
- b) **Règle 2 : Association un à plusieurs** Il faut ajouter un attribut de type clé étrangère dans la relation fils de l'association.
- c) **Règle 3 : Associations plusieurs à plusieurs ou classes-associations** L'association devient une relation dont la clé primaire est composée par la concaténation des identifiants des classes connectés à l'association. Les attributs de l'association doivent être ajoutés à la nouvelle relation.
- d) **Règle 4 : Associations un à un** Pour représenter une association 1 vers 1 entre deux relations, la clé primaire de l'une des relations doit figurer comme clé étrangère dans l'autre relation.
- e) **Règle 5 : Agrégations** Les agrégations s'implémentent comme les associations.
- f) **Règle 6 : Transformation de l'héritage** Décomposition descendante (push-down) : s'il existe une contrainte de totalité ou de partition sur l'association d'héritage, il est possible de ne pas traduire la relation issue de la sur-classe. Il faut alors faire migrer tous ses attributs dans la (les) relation(s) issue(s) de la (des) sous-classe(s).