



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

Université ABDERRAHMANE MIRA BEJAÏA

Faculté des Sciences Exactes

Département d'informatique

MÉMOIRE DE FIN DE CYCLE

En vue de l'obtention d'un master professionnel en

Informatique spécialité génie logiciel

Thème

***Conception et réalisation d'une application
web e-commerce***

Cas d'étude : LG SHOW ROOM

Présenté par :

M^r BAHRI Mustapha
M^r HAMDAROU Ahmed

Jury composé de :

Président : M^r LARBI Ali
Examineur : M^r AKILAL Karim
Promoteur : M^r ALOUI Abdelouhab

Promotion 2016/2017

Remerciements

Au terme de ce travail qui marque la fin du cycle de master de notre formation au sein de l'université ABDERRAHMANE MIRA, il nous est donné l'opportunité d'exprimer notre gratitude à tous ceux qui, de loin ou de près, ont matériellement ou moralement contribué à la réalisation de notre modeste travail. Qu'ils trouvent ici l'expression de notre considération.

Au Seigneur Dieu, qui nous a donné la vie, l'intelligence et le courage de réaliser ce travail.

A nos chers parents par leur affection et amour de nous avoir donné la vie et l'éducation. voilà aujourd'hui nous sommes comptés parmi les hommes intellectuels du monde. Qu'ils se réjouissent du fruit de leur progéniture.

Nous exprimons nos vifs remerciements, notre profonde gratitude et notre reconnaissance à notre encadreur Monsieur ALOUI Abdelouhab, qui a dirigé ce travail dans la continuité de notre stage. Nous remercions spécialement le personnel du distributeur officiel de LG, le propriétaire, d'avoir accepté de nous accueillir à LG ShowRoom et de nous avoir mis dans les meilleures conditions de travail.

Nous tenons également à remercier les membres de jury d'avoir accepté d'examiner notre travail.

Enfin, Nous remercions, de tout coeur, tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Dédicaces

A

NOS PARENTS

POUR LES SACRIFICES DEPLOYES A NOS
EGARDS. POUR LEUR PATIENCE, LEUR AMOUR ET LEUR
CONFIANCE EN NOUS. ILS ONT TOUT FAIT POUR NOTRE
BONHEUR ET NOTRE REUSSITE. QU'ILS TROUVENT DANS CE
TRAVAIL, LE TEMOIGNAGE DE NOTRE PROFONDE AFFECTION ET
DE NOTRE ATTACHEMENT INDEFECTIBLE. NULLE DEDICACE NE
PUISSE EXPRIMER CE QUE NOUS LEUR DEVONS QUE DIEU
LEUR RESERVE LA BONNE SANTE ET UNE
LONGUE VIE.

A

NOS CHERS FRERES ET SOEURS

A

TOUS NOS AMIS EN TEMOIGNAGE DE NOS SINCERES
RECONNAISSANCES POUR LES EFFORTS QU'ILS ONT CONSENTIS
POUR NOUS SOUTENIR AU COURS NOS ETUDES.
QUE DIEU NOUS GARDE TOUJOURS UNIS.

A

TOUTE PERSONNE QUI NOUS A APPORTE DE L'AIDE.

TABLE DES MATIÈRES

1	Cadre général et étude préalable	2
1.1	Introduction	2
1.2	Cadre général du sujet de stage	2
1.3	Présentation de l'organisme d'accueil	2
1.4	Présentation du sujet	3
1.5	Étude préalable	3
1.5.1	Analyse du site officiel de LG	3
1.5.2	Analyse du site officiel de ARDIS	5
1.6	Organisation du rapport	6
1.7	Conclusion	7
2	Étude de l'existant et analyse des besoins	8
2.1	Introduction	8
2.2	Études de l'existant	8
2.2.1	Description de l'existant	8
2.2.2	Critiques de l'existant	8
2.2.3	Solution proposée	9
2.3	Cycle de vie du projet	10
2.3.1	Définition d'un cycle de vie	10
2.3.2	Les activités d'un cycle de vie	10
2.3.3	Exemples de cycles de vie :	11
2.4	Analyse des besoins	14
2.4.1	Définition	14
2.4.2	L'intérêts de l'analyse des besoins	14
2.4.3	Identification des acteurs	14
2.4.4	Principales Fonctionnalités	15
2.5	Gestion de sécurité de l'application	15
2.5.1	Confidentialité	16

2.5.2	Authentification :	16
2.5.3	Droits d'accès	16
2.6	Conclusion	16
3	Conception	18
3.1	Introduction	18
3.2	Conception générale	18
3.2.1	Architecture MVC (Modèle-vue-contrôleur)	18
3.2.2	Langage de modélisation	20
3.3	Conception détaillée	22
3.3.1	Diagrammes de cas d'utilisation	22
3.3.2	Diagrammes d'activités :	26
3.3.3	Diagrammes des séquences	32
3.3.4	Diagramme de classes	38
3.3.5	Script de création de la base de données :	41
3.3.6	Les chartes graphiques de notre application web :	41
3.4	Conclusion	44
4	Réalisation	45
4.1	Introduction	45
4.2	Présentation des concepts informatiques utilisés	45
4.2.1	Développement Front-end	45
4.2.2	Modélisation des données	46
4.2.3	Mapping objet-relationnel	46
4.2.4	Gestion des sessions	46
4.2.5	Transaction Business Logic	46
4.2.6	La sécurité des applications WEB	47
4.3	les technologies utilisées au développement	47
4.3.1	HTML	47
4.3.2	CSS	47
4.3.3	JavaScript	47
4.3.4	Bootstrap	47
4.3.5	JAVA	48
4.3.6	JSP	48
4.3.7	JSTL	48
4.3.8	Java servlet	48
4.3.9	JPA	48
4.3.10	Java EE	48
4.3.11	EJB	49
4.3.12	JDBC	49
4.4	Outils de développement	49

4.4.1	NetBeans IDE	49
4.4.2	GlassFish	49
4.4.3	MySQL	49
4.4.4	MySQL Workbench	50
4.4.5	Adobe PhotoShop CC	50
4.5	Le MVC de notre application	50
4.6	Les interfaces développées dans notre application	51
4.6.1	Interface d'accueil	51
4.6.2	Interface d'inscription	52
4.6.3	Interface d'authentification	53
4.6.4	Interface du catalogue	54
4.6.5	Interface du fiche technique d'un produit	55
4.6.6	Interface du panier	56
4.6.7	Interface d'ajout d'un produit	57
4.7	Conclusion	57

TABLE DES FIGURES

1.1	Organigramme du Distributeur "LG"	3
1.2	Page d'accueil du site officiel de LG	4
1.3	Page d'accueil du site officiel de Ardis	6
2.1	Modèle de cycle de vie en cascade.	11
2.2	Modèle de cycle de vie en spirale.	12
2.3	Modèle de cycle de vie en V.	13
3.1	Le schéma de la structure générique de l'architecture MVC.	19
3.2	Le traitement d'une demande d'un client dans l'architecture MVC.	20
3.3	Diagramme de cas d'utilisation d'un visiteur.	24
3.4	Diagramme de cas d'un client.	25
3.5	Diagramme de cas de l'administrateur.	26
3.6	Diagramme d'activité d'inscription.	30
3.7	Diagramme d'activité d'authentification.	31
3.8	Diagramme d'activité d'ajout d'un article.	32
3.9	Diagramme de séquence d'inscription.	34
3.10	Diagramme de séquence d'authentification.	35
3.11	Diagramme de séquence de remplissage de panier.	36
3.12	Diagramme de séquence effectuer une commande.	37
3.13	Diagramme de séquence d'ajout d'un produit.	38
3.14	Diagramme de classes de notre application.	40
3.15	Le SQL généré de la base de données.	41
3.16	La charte graphique de l'accueil.	42
3.17	La charte graphique de consultation d'un produit.	43
3.18	La charte graphique de panier.	44
4.1	Le MVC de notre application. [15]	50
4.2	Interface d'accueil.	51

4.3	Interface d'inscription.	52
4.4	Interface d'authentification.	53
4.5	Interface du catalogue.	54
4.6	Interface du fiche technique d'un produit.	55
4.7	Interface du panier.	56
4.8	Interface d'ajout d'un produit.	57

INTRODUCTION GÉNÉRALE

En ces temps modernes où l'informatique occupe de plus en plus de place dans la vie quotidienne, ce vaste domaine est devenu l'outil indispensable de l'homme. En effet, le stockage et le traitement des données sont automatisés dans plusieurs domaines d'applications tels que les administrations, les entreprises, mais aussi dans le domaine du commerce.

Les boutiques en ligne sont depuis des années, largement conseillées pour les sociétés qui se basent sur la vente de produits et même de services. Ces types de sites web représentent un dispositif global fournissant aux clients un pont de passage à l'ensemble des informations, des produits, et des services à partir d'un portail unique en rapport avec son activité.

Les sites de vente en ligne permettent aux clients de profiter d'une foire virtuelle disponible et quotidiennement mise à jour sans la moindre contrainte, ce qui leur permet de ne jamais rater les nouvelles offres, de ce fait on obtient une foire sans problèmes de distance géographique, ni d'horaire de travail ni de disponibilité de transport. D'une autre part ces sites permettent à la société de profiter de cet espace pour exposer ses produits à une plus large base de clientèle.

Notre projet est réalisé dans le cadre du mémoire de master Génie Logiciel ayant comme objectif principal : la conception et la création d'une boutique virtuelle pour un distributeur exclusif de la marque « LG » qui est une société de production et de vente de produits électroniques comme les téléphones portables, télévisions, frigos, etc.

Le premier chapitre s'intitule «Cadre général et étude préalable» où nous présentons notre thème en général, et nous analysons deux projets similaires pour profiter des avantages et éviter les inconvénients dans le présent projet.

Le second chapitre qui s'intitule «Etude de l'existant et analyse des besoins» est consacré à l'analyse de l'état actuel de l'entreprise dont nous représenterons l'ensemble des besoins fonctionnels et non fonctionnels.

Le troisième chapitre qui a pour titre «conception» est consacré à la conception de l'application à mettre en œuvre.

Le dernier chapitre qui est intitulé «Réalisation» sera consacré à l'implémentation de notre application.

Enfin, nous terminerons par une conclusion générale et des perspectives.

CHAPITRE 1

CADRE GÉNÉRAL ET ÉTUDE PRÉALABLE

1.1 Introduction

Nous commençons dans ce premier chapitre par une mise en contexte de notre projet, en spécifiant le cadre de son élaboration, présentant l'organisme d'accueil au sein duquel nous avons effectué le stage relatif au présent projet, une présentation du sujet dont nous parlerons de l'objectif de ce mémoire ainsi qu'un survol sur le plan de travail suivi pour réaliser ce mémoire, et nous terminerons par une étude préalable de deux sites web e-commerces dont le principe est le même avec notre projet.

1.2 Cadre général du sujet de stage

Notre projet se portera sur la conception et la réalisation d'une application web commercial pour le distributeur Algérien de la marque « LG », ce projet a été proposé dans le cadre de l'élaboration d'un stage de deux mois au niveau de cette entreprise, inclus dans la formation master génie logiciel.

1.3 Présentation de l'organisme d'accueil

Sarl Algérienne Electro Distributeur Officiel LG comme l'indique son nom est une société algérienne qui représente les produits LG en Algérie. Les principales activités de cette entreprise est la vente et la réparation des produits électroniques comme les téléphones portables, les téléviseurs, les climatiseurs et aussi les articles électroménagers comme les réfrigérateurs, les cuisinières, les aspirateurs de la marque internationale LG de la basse à la haute gamme.[Figure 1.1]

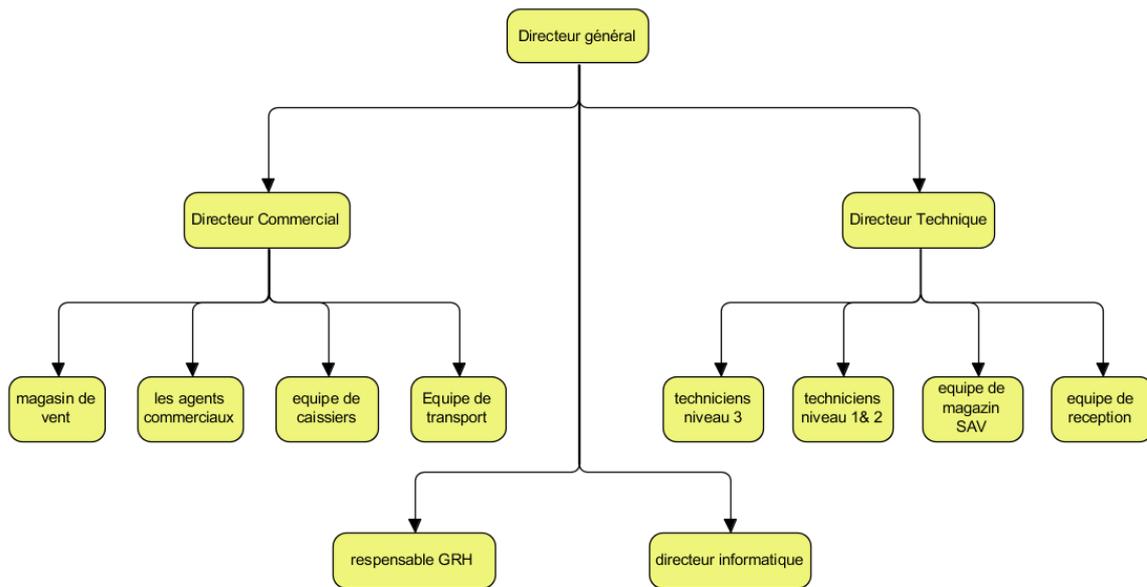


FIGURE 1.1 – Organigramme du Distributeur "LG"

1.4 Présentation du sujet

Le sujet de notre mémoire consiste à développer une boutique en ligne pour présenter, commercialiser et livrer les produits aux clients, que nous estimons devenir de plus en plus nombreux. L'objectif de ce mémoire est de concevoir et développer une application web commerciale qui doit permettre l'inscription des visiteurs pour devenir clients, le suivi des commandes effectuées, la gestion de paiement en ligne, le suivi des livraisons et la mise à jour des nouveaux produits.

1.5 Étude préalable

1.5.1 Analyse du site officiel de LG

1.5.1.1 Description

LG.com est le site officiel de la célèbre marque de production des produits électroniques de la haute gamme, propose un large choix d'articles avec leur différents prix, des réductions et des offres spéciales. Son objectif est de faire des vente en ligne des produits électroniques proposés sous forme d'un catalogue web.

1.5.1.2 Les fonctionnalités et les offres de site

Ce site web e-commerce offre les services suivants :

- Vente de téléphones, d'appareils photos et d'accessoires.
- Vente de produits électroménagers.
- Possibilité d'effectuer une recherche des produits par plusieurs catégories.

- Le déploiement d'un espace client pour l'internaute.
- Le déploiement d'un espace panier.
- Promotions.
- Inscription.
- Description de la marque.
- Contact.

Adresse électronique du site : www.LG.com/dz

1.5.1.3 Aperçu de la page principale

La page principale est l'interface d'accueil de ce site. Elle représente la première interaction entre le client et le site. Elle nous affiche une liste de catégories des produits TV/vidéo/audio, mobile, électroménagers et climatiseurs, avec un bouton de recherche qui nous permet de rechercher un produit spécifique. [Figure 1.2]

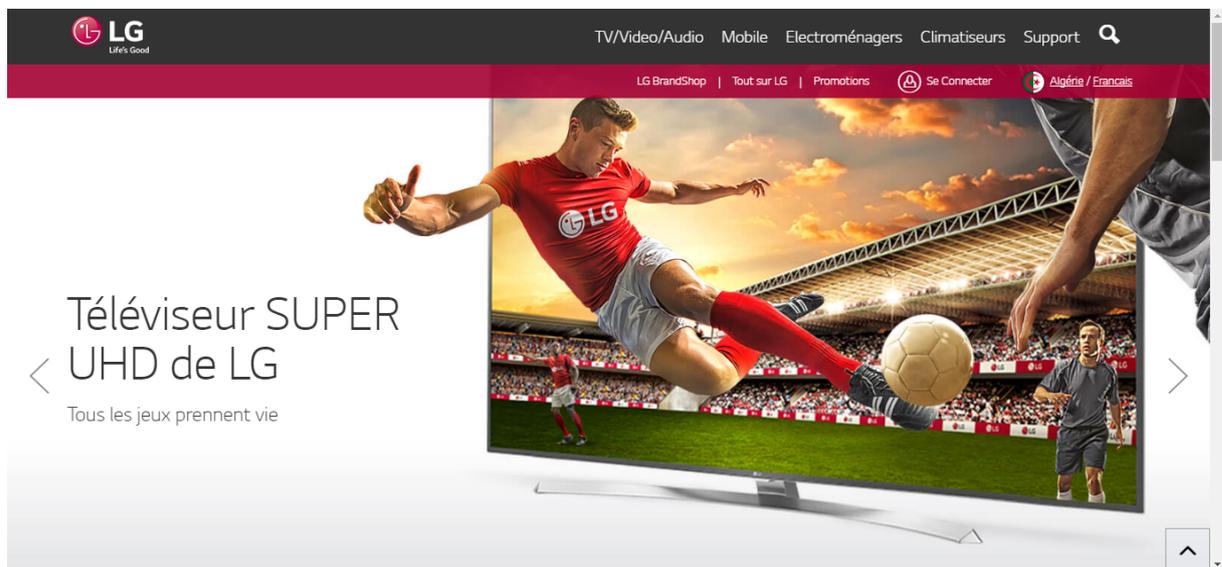


FIGURE 1.2 – Page d'accueil du site officiel de LG

1.5.1.4 Les avantages du site

- Assure une consultation d'une catégorie riche de plusieurs modèles de produits.
- Présence de notifications de nouveautés et de promotions.
- Une très haute qualité graphique de l'image.
- un texte lisible qui encourage l'utilisateur à poursuivre sa navigation.
- La consultation du site est très facile donc n'importe quel client peut facilement naviguer.

1.5.1.5 Les Inconvénients du site

- Les catégories sont présentées sous forme d'une barre de menu ce qui limite l'ajout d'autres catégories.

- Procédure de vente très lente.

1.5.2 Analyse du site officiel de ARDIS

1.5.2.1 description

ARDIS.dz est le site officiel de la chaîne algérienne d'hypermarchés ARDIS avec lequel ils commercialisent leurs produits de plusieurs marques et plusieurs catégories (alimentation, accessoires, habillement...) et leurs offres et promotions.

1.5.2.2 Les fonctionnalités et les offres du site

Le site web ardis.dz offre les services suivants :

- Vente de produits de plusieurs catégories (alimentation, électroménagers ...).
- promotions et remises.
- Déploiement d'un espace paniers.
- Inscription.
- Contact.
- Description de l'hypermarché.

Adresse électronique du site : www.ardis.dz

1.5.2.3 Aperçu de la page principale

La page principale est la première interaction entre le visiteur et le site. Elle nous affiche une liste de catégories des produits (Alimentation, produits frais, textile ..) avec un bouton de recherche qui nous permet de rechercher un produit spécifique.[Figure 1.3]



FIGURE 1.3 – Page d'accueil du site officiel de Ardis

1.5.2.4 Les avantages de site

- Une bonne organisation des produits par catégories ce qui facilite la recherche des produits.
- Les catégories sont présentées dans une liste ce qui facilite l'ajout d'une nouvelle catégorie.
- Les nouveaux produits et promotions sont affichés en premier lieu pour attirer les clients
- L'espace panier est accessible même pour les visiteurs qui rend les commandes plus simple.

1.5.2.5 Les inconvénients de site

- Une très grande partie de la page réservée à la publicité et donc l'utilisateur se croit sur un site publicitaire et non un site de vente.
- Beaucoup de couleurs avec des tendances différentes ce qui cause une sorte d'incohérence entre les vues.
- Excès d'images, d'animations et de liens même en doublant les mêmes images dans la même page.

1.6 Organisation du rapport

Pour un bon travail il nous faut un rapport bien structuré qui peut être exploité après la mise en place de cette application, pour cela nous allons organiser notre présent rapport de la manière suivante :

- Dans le premier chapitre intitulé « Cadre général et étude préalable », nous allons mettre notre projet dans son cadre général en définissant la société d'accueil, présentant le sujet et nous allons prendre deux applications web de même principe que le nôtre, comme des

exemples afin de les analyser et dégager les bénéfices et les inconvénients et donc obtenir une idée plus claire de ce que nous devrions faire dans notre application.

- Dans le deuxième chapitre intitulé «Etudes de l'existant et analyse des besoins», nous présentons l'étude préliminaire de la méthode utilisée avant la mise en place de notre système et nous allons spécifier l'ensemble des besoins fonctionnels et non fonctionnels liées à notre application.
- Pour le troisième chapitre intitulé «conception», nous abordons la phase de conception. Nous spécifions d'abord la méthode de conception adaptée, après nous présentons les différents diagrammes de notre application web.
- Enfin et au niveau du quatrième et dernier chapitre intitulé «Réalisation», nous allons présenter notre application web, en mentionnant les différents environnements de travail utilisés pour entamer le projet, ainsi qu'en citant les principales interfaces réalisées.

1.7 Conclusion

Dans ce premier chapitre nous avons mis le sujet dans son cadre général et nous avons analysé deux sites web e-commerce. nous commençons le deuxième chapitre intitulé «Etudes de l'existant et analyse des besoins», dans lequel nous expliquerons la méthode utilisée par le distributeur pour vendre ses produits avant la mise en place de notre application, suivie d'une analyse des besoins pour réaliser une nouvelle solution plus efficace.

CHAPITRE 2

ÉTUDE DE L'EXISTANT ET ANALYSE DES BESOINS

2.1 Introduction

Dans ce deuxième chapitre, nous étudions la manière de vente actuelle, suivie d'une critique pour pouvoir se concentrer sur les problèmes à résoudre après la réalisation de notre projet. Ainsi, ce chapitre présente un ensemble de fonctionnalités front office et back office.

2.2 Études de l'existant

2.2.1 Description de l'existant

Comme toutes les sociétés commerciales, le distributeur officiel de LG possède sa manière de présenter et de commercialiser ses produits. En effet, pour acheter un équipement électronique dans cette boutique tel qu'un téléphone portable ou un climatiseur, le client doit se déplacer au local de la société afin de chercher une offre de vente qui satisfait ses besoins. Ses déplacements peuvent être inutiles et peuvent même provoquer un gaspillage de temps. D'ailleurs, même le vendeur n'a aucun moyen pour mettre à disposition ses annonces de vente et services, à l'exception des supports traditionnels tels que les journaux, les petites affiches et la présentation des produits aux vitrines. Ainsi, un moyen fiable et automatisé permettant d'informer un grand nombre de clients des offres de vente et des services nécessaires.

Vu l'accroissement de la technologie Internet, la vente en ligne est devenu une nécessité incontournable pour les commerçants.

2.2.2 Critiques de l'existant

La solution actuelle est manuelle, posant ainsi des problèmes différents, à savoir :

- Risque de mélanger les documents ; ce qui peut être fatal.
- L'enregistrement des clients se fait manuellement sur papier.

-
- La perte de la clientèle est possible, surtout quand le traitement de leurs demandes traîne ou prend beaucoup de temps pour être livré.
 - Le règlement des factures se fait en espèce ou par chèque, qui nécessite le déplacement du client.
 - Les produits sont classés par catégorie (TV/Audio/Vidéo, mobile et matériels électroménagers) et par sous-catégorie non liées et non hiérarchisées, ce qui rend la recherche plus pénible.

2.2.3 Solution proposée

Grâce à l'Internet, de nouvelles perspectives de développement apparaissent dans l'élargissement du marché économique.

La création d'une application web a pour but de valoriser l'image de la société et faire des économies. L'utilisation d'Internet, comme moyen de communication de masse, permet également de baisser le coûts marketing et d'autres frais. Avec la transmission du haut débit et la robuste sécurité de moyens de paiement, la confiance des utilisateurs en ce qui concerne l'e-commerce est croissante. La plupart des personnes adultes utilisent aujourd'hui l'internet pour faire des achats. Les consommateurs et les entreprises s'orientent de plus en plus vers les boutiques en lignes qui permettent la comparaison, la disponibilité des produits et la vérification des prix d'où l'économie considérable du temps.

Ce projet consiste en la mise en place d'une application Web qui gère la commercialisation de matériels électroniques de la marque LG. Ceci est possible à travers des catalogues en ligne proposant des produits aux meilleurs prix par rapport aux concurrents. La société n'aura donc qu'à agencer ses produits et bien sûr de mettre sa base de données à jour. Les clients peuvent consulter l'application, faire une inscription et commander les produits, qui sont par la suite livrés à domicile. Cette boutique en ligne permettra d'offrir beaucoup de services à savoir :

- Recherche de produit.
- Consultation de catalogues de produits.
- Lancer une commande en ligne.

Cette application Web permettra de cibler une nouvelle catégorie de clientèles (locale et internationale), et d'offrir une meilleure qualité de service en communication et en commerce. Cette application devra contenir des sessions séparées :

- **Session visiteur** : c'est l'ensemble des interfaces où on peut consulter les catalogues des produits (prix, fiche technique...), la disponibilité des produits, les promotions et peut aussi faire une inscription afin d'être un membre pour qu'il puisse faire des commandes.
- **Session client** : c'est l'ensemble des interfaces accessible après une authentification où le client peut effectuer ses commandes et remplir son panier.
- **Session administrateur** : c'est l'interface de l'administrateur (superviseur). Après une authentification, l'administrateur a l'accès à la gestion des produits et à la gestion des clients (ajouter, modifier, supprimer).

2.3 Cycle de vie du projet

2.3.1 Définition d'un cycle de vie

Le cycle de vie d'une application comprend toutes les étapes depuis sa conception et sa réalisation jusqu'à sa mise en œuvre. L'objectif d'un tel découpage est de permettre de définir des jalons intermédiaires permettant la validation du développement du logiciel et la vérification de son processus de développement.

L'origine de ce découpage provient du constat que les erreurs ont un coût si élevé qu'elles sont détectées tardivement dans le processus de réalisation. Le cycle de vie permet de détecter les erreurs le plutôt possible et d'obtenir une qualité conforme aux exigences.

2.3.2 Les activités d'un cycle de vie

Le cycle de vie à suivre pour réaliser une application web, comprend généralement les activités suivantes :

- **Spécification des besoins** : elle consiste à définir les exigences d'une façon formelle et son intégration dans une stratégie globale.
- **Conception générale** : dans cette activité, il s'agit de la préparation de l'architecture générale du logiciel.
- **Conception détaillée** : elle consiste à définir précisément chaque sous-ensemble du logiciel.
- **Développement** : (ou programmation) il s'agit d'une traduction des fonctionnalités définies dans la phase de conception en langage de programmation (code source).
- **Tests unitaires** : ils permettent de vérifier individuellement que chaque sous-ensemble du logiciel est implémenté conformément aux normes définies dans la conception.
- **Intégration** : dite aussi tests systèmes, elle consiste à vérifier que le logiciel correspond exactement au cahier des charges du projet en obtenant enfin un manuel d'utilisation bien détaillé aux utilisateurs.
- **Validation** : c'est-à-dire la validation de conformité de l'application avec les buts spécifiés à la première étape de cycle de vie.

2.3.3 Exemples de cycles de vie :

2.3.3.1 Modèle de cycle de vie en cascade

La figure suivante [figure 2.1] représente le schéma de modèle de cycle de vie en cascade :

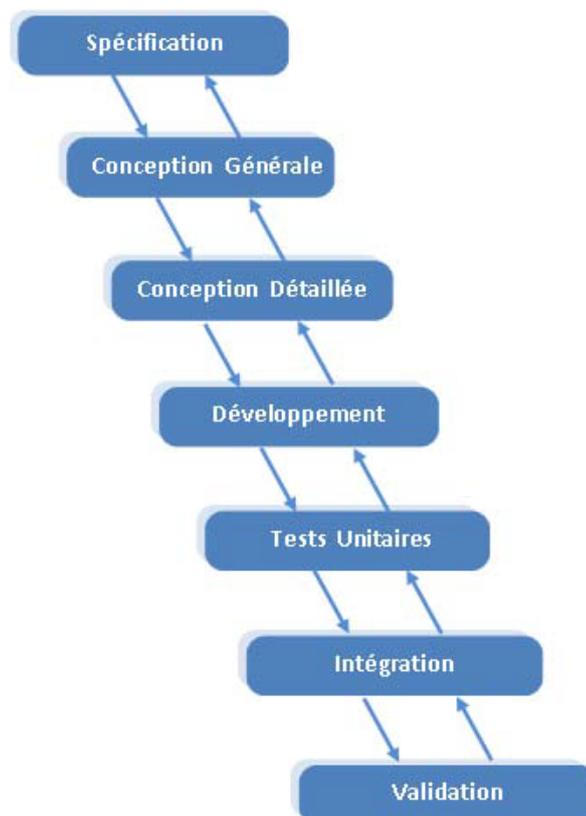


FIGURE 2.1 – Modèle de cycle de vie en cascade.

Le modèle de cycle de vie en cascade a été mis au point dès 1966, puis formalisé aux alentours de 1970.

Dans ce modèle le principe est très simple : chaque phase se termine à une date précise en produisant certains documents ou logiciels. Les résultats sont définis à la base des interactions entre étapes, ils sont soumis à une revue approfondie et on ne passe à la phase suivante que s'ils sont jugés équivalents aux normes. Le modèle original ne comportait pas la possibilité de retour en arrière. Celle-ci a été rajoutée ultérieurement sur la base qu'une étape ne remet en cause que l'étape précédente, ce qui s'avère insuffisant dans la pratique. [03]

2.3.3.2 Modèle de cycle de vie en spirale

La figure suivante [figure 2.2] représente le schéma de modèle de cycle de vie en spirale :

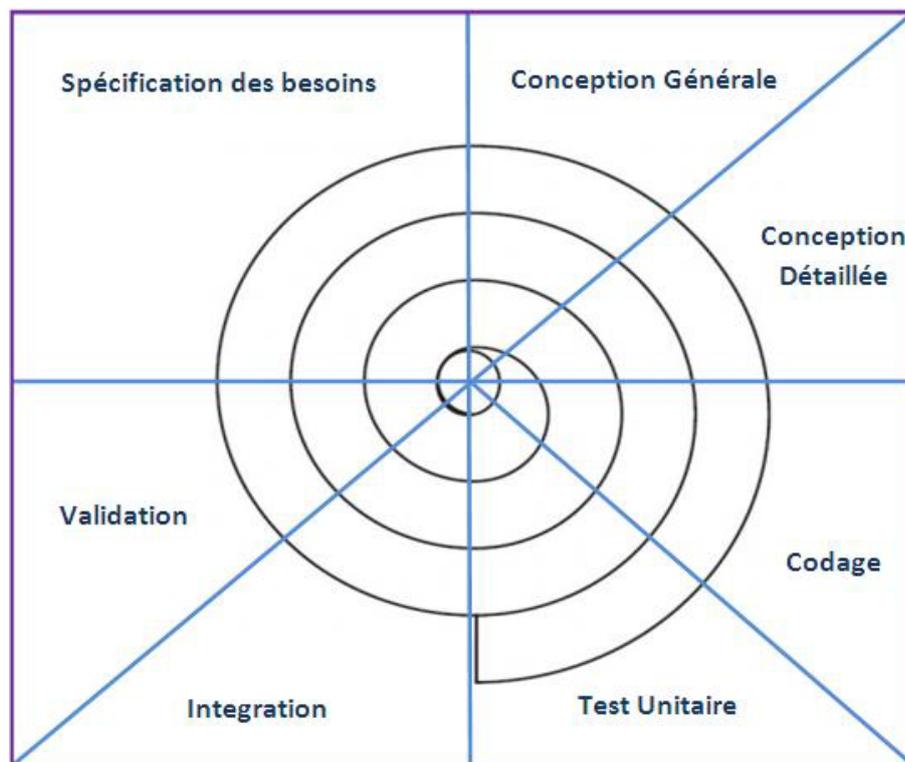


FIGURE 2.2 – Modèle de cycle de vie en spirale.

Le modèle en spirale a été défini par Barry Boehm en 1988 dans son article "A Spiral Model of Software Development and Enhancement".

Le modèle en spirale (spiral modèle) est un modèle de cycle de vie qui reprend les différentes étapes du cycle en V. Par l'implémentation de versions successives, le cycle recommence en proposant un produit de plus en plus complet. Il met cependant plus l'accent sur la gestion des risques que le cycle en V [13]. Chaque cycle de la spirale se déroule en quatre phases :

1. détermination des objectifs, des alternatives et des contraintes.
2. analyse des risques, évaluation des alternatives.
3. développement et vérification de la solution retenue.
4. revue des résultats et vérification du cycle suivant.

2.3.3.3 Modèle de cycle de vie en V

La figure suivante [figure 2.3] représente le schéma de modèle de cycle de vie en V :

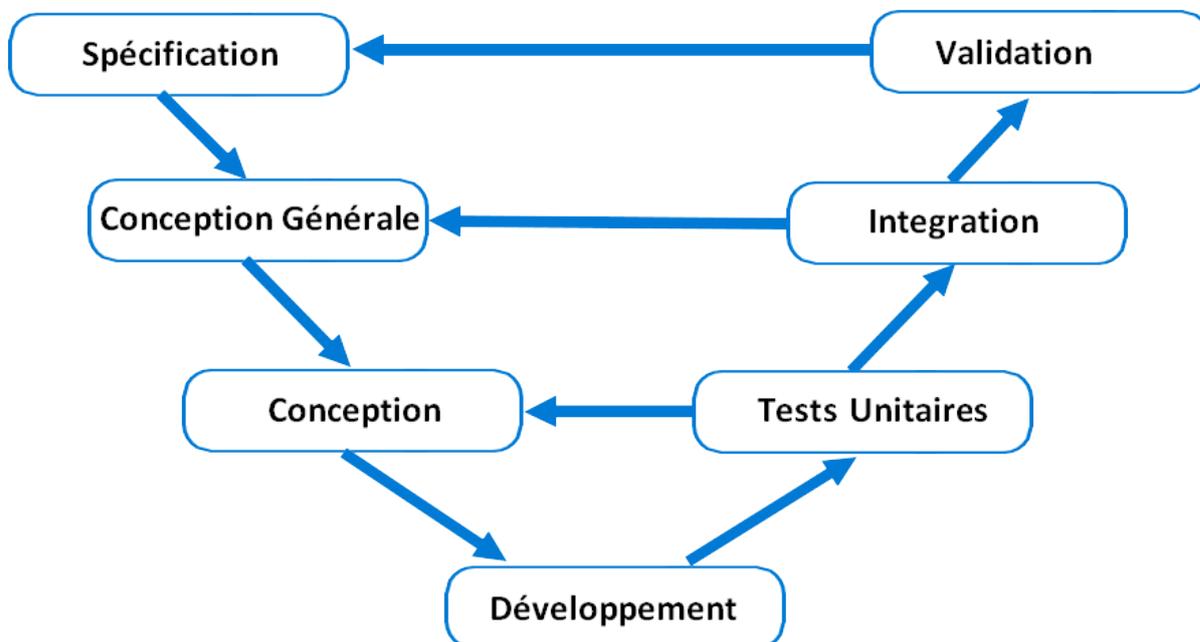


FIGURE 2.3 – Modèle de cycle de vie en V.

Le modèle du cycle de vie en V est un modèle conceptuel de gestion de projet, standardisé les années 1980 (Goldberg, 1986), il dérive du modèle en cascade suite au problème de réactivité de ce dernier. Il permet, en cas d'anomalie, d'éliminer le retour aux étapes précédentes tardivement. Dans le déroulement de notre projet, on a utilisé ce modèle de cycle de vie car, les exigences sont bien connues et non sujettes à la modification en plus, il offre beaucoup d'avantages, aussi il est adéquat au développement de notre travail.

Les avantages du modèle du cycle de vie en V sont les suivants :

- La qualité de la mise en œuvre des tests (Les tests sont définis à l'issue de chaque phase), donc augmente la qualité du logiciel.
- Modèle éprouvé dans l'industrie.
- Normalisé (ISO-12207, MILSTD-498...)
- Chaque livrable doit être testable.
- Deux types de tâches sont réalisées en parallèle : Verticalement on prépare l'étape suivante et Horizontalement : on prépare la vérification de la tâche en cours.
- Simple et facile à comprendre.
- Force la documentation : une phase ne peut se terminer avant qu'un document soit validé.

Comme tous les modèles de cycle de vie d'un logiciel admettent des avantages et des inconvénients, Les inconvénients du modèle du cycle de vie en V sont les suivants :

- La validation finale par le client très tardive augmente les risques de dépassement de délai et donc l'augmentation du coût.
- Modèle dirigé par les documents.
- Non compréhensibles par les clients.
- Ne contient pas d'activité de l'analyse du risque.
- Ne marche que si les exigences sont stables et le problème connu.
- Ne gère pas explicitement le changement des spécifications.
- Manque de flexibilité (ne traite pas les évolutions).
- Problèmes découverts en phase de validation
- Irréaliste dans de nombreux cas.

2.4 Analyse des besoins

2.4.1 Définition

Appelée aussi spécification des besoins. Elle représente la première phase du cycle de développement d'une application. Elle doit décrire les fonctionnalités du logiciel et les contraintes sous lesquelles celui-ci doit être réalisé. Les besoins sont exprimés par le client dans le cahier des charges rédigé en langage naturel.

2.4.2 L'intérêts de l'analyse des besoins

- Permet de décrire les fonctionnalités de l'application et les contraintes sous les quelles celle-ci doit être réalisé.
- Permet au développeur de proposer au client un cahier des charges réaliste et convenant à ses besoins.
- Avoir une description générale du projet
- Permet de connaître les futurs utilisateurs de l'application.
- Connaître les besoins et les attentes du client.

2.4.3 Identification des acteurs

Nous allons maintenant énumérer les acteurs susceptibles d'interagir avec le système. Tout d'abord, nous commençons par définir ce qu'est un acteur.

Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié.[11]

Les acteurs de notre application sont :

- **Le visiteur** : est un internaute qui visite une ou plusieurs pages de l'application durant une période de temps.
- **Le client** : toute personne ayant déjà consulté l'application et qui a fait l'inscription peut par la suite, réserver des produits.

-
- **L'administrateur** c'est le superviseur qui contrôle et rectifie l'application web pour assurer le bon fonctionnement du système.

2.4.4 Principales Fonctionnalités

2.4.4.1 Fonctionnalités en front office

Pour un bon fonctionnement du système, l'application doit permettre d'accéder aux fonctionnalités suivantes :

La gestion du catalogue des produits :

- Affichage de l'ensemble des caractéristiques de chaque produit dans une organisation homogène.
- Tri des produits pour chaque catégorie par prix (croissant - décroissant).
- Affichage des meilleures ventes.

La gestion de l'espace client :

- Inscription du client.
- Système d'authentification des clients par login et mot de passe.
- Paiements et transactions sécurisés.
- Gestion des commandes.
- impression de fichiers (PDF) : le client aura la possibilité de télécharger les bons de commandes, les devis, les factures, etc.

2.4.4.2 Fonctionnalités en back office

L'administrateur a accès à l'application via un login et un mot de passe. Il pourra ainsi assurer : **La gestion des utilisateurs :**

- Ajout / modification / suppression des profils (administrateurs / clients).

La gestion des commandes :

- Ajout / Modification / Suppression d'une commande.
- Gestion des factures et des bons de livraison.
- Édition des factures et des bons de livraison au format HTML et PDF.
- Gestion des bons de réduction.

La gestion du contenu informationnel :

- Ajout / modification / suppression des actualités (activation / désactivation).
- Ajout / modification / suppression des promotions et des remises.

La gestion du catalogue :

- Ajout / modification / suppression des fiches produits.

2.5 Gestion de sécurité de l'application

L'interface doit être accessible par un navigateur internet. Il est ainsi important de définir des profils et leurs droits.

L'utilisateur client possède un login et un mot de passe personnel. Il lui sera possible d'accéder à son espace client et d'effectuer des commandes.

Notre application doit répondre aux aspects de sécurité suivants :

2.5.1 Confidentialité

La confidentialité est le maintien du secret des informations. Transposée dans le contexte de l'informatique, la notion de confidentialité peut être vue comme la « protection des données contre une divulgation non autorisée ».

Il s'agit tout d'abord d'interdire l'accès en lecture et en écriture aux informations considérées comme privées à chaque utilisateur.

Un autre aspect concerne la confidentialité des transmissions, qui devront être cryptées pour tous les services où l'utilisateur est identifié.

Les fichiers de données personnelles (clients, commandes et abonnements) sont sécurisés afin d'assurer la tranquillité des visiteurs.

Le client est redirigé vers le serveur du service de paiement qui confirme le succès ou l'échec du règlement de la commande, à aucun moment les coordonnées des transactions des clients ne doivent être divulguées.

2.5.2 Authentification :

L'authentification consiste à assurer l'identité d'un utilisateur, c'est-à-dire de garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être. Un contrôle d'accès peut permettre (par exemple par le moyen d'un mot de passe qui devra être crypté) l'accès à des ressources uniquement aux personnes autorisées.

L'application devra procéder à une authentification sécurisée des utilisateurs afin d'assurer la traçabilité de leurs actions, ainsi que l'intégrité et la non-falsification des informations.

Toutes les pages intégrant un formulaire de saisie d'identifiant et de mot de passe pour les accès privés seront sécurisées. Les échanges de données sensibles avec le serveur seront cryptés.

2.5.3 Droits d'accès

L'accès sera contrôlé selon le profil d'utilisateur. Pour chaque type d'information, les droits d'accès (lecture, écriture) devront être déterminés par la catégorie et par l'identité de l'utilisateur.

2.6 Conclusion

Dans ce chapitre, nous avons élaboré une étude du système existant, posé la problématique et proposé une solution, ensuite nous avons présenter quelques exemples de cycle de vie dont

notre choix c'est porté sur le modèle en V.

Dans le chapitre suivant intitulé « conception », nous allons présenter la conception générale et détaillée où nous allons définir les différents diagrammes de séquence et la description des cas d'utilisations.

3.1 Introduction

Après avoir présenté l'étude de l'existant et l'analyse des besoins dans le chapitre précédent, nous allons à présent passer à la conception de notre application. Nous avons divisé ce chapitre en deux parties. La première partie est la conception générale où nous présenterons le modèle de conception MVC (Modèle-vue-contrôleur), que nous avons exploité dans la conception de notre projet, ensuite, nous allons donner une description de ce modèle et ces composants, aussi nous expliquerons ce qu'est un langage de modélisation UML, les points forts et les points faibles de ce dernier. La deuxième partie est la conception détaillée, où nous allons présenter les différents diagrammes des cas d'utilisation, d'activités, des séquences, de classes ainsi quelques chartes graphiques de l'application.

3.2 Conception générale

3.2.1 Architecture MVC (Modèle-vue-contrôleur)

3.2.1.1 Définition du Architecture MVC

Le Modèle-vue-contrôleur (MVC) est un motif d'architecture logicielle très répandu et fort utile, destiné aux interfaces graphiques. lancé en 1978 Trygve Reenskaug, il est aujourd'hui fortement recommandé dans l'univers J2EE. Néanmoins il faut retenir que c'est un modèle de conception, et il est donc indépendant du langage de programmation .

Le MVC est un modèle de conception qui repose sur la volonté de séparer les données, les traitements et la présentation.

3.2.1.2 Les composants de l'architecture MVC

L'architecture MVC est décomposée en trois étapes :

- Le **modèle** : Représente les données et les règles métiers. C'est dans ce composant que s'effectuent les traitements liés au cœur du métier. Les données peuvent être liées à une

base de données, des services Web,... Il est important de noter que les données sont indépendantes de la présentation. En d'autres termes, le modèle ne réalise aucune mise en forme. Ces données pourront être affichées par plusieurs vues. Du coup le code du modèle est factorisé : il est écrit une seule et unique fois puis réutilisé par chaque vue.[14]

- La **vue** : correspond à l'IHM (l'interface homme-machine). Elle présente les données et interagit avec l'utilisateur. Dans le cadre des applications Web, il s'agit d'une interface HTML, mais n'importe quel composant graphique peut jouer ce rôle.[14]
- Le **contrôleur** : quant à lui, se charge d'intercepter les requêtes de l'utilisateur, d'appeler le modèle puis de rediriger vers la vue adéquate. Il ne doit faire aucun traitement. Il ne fait que l'interception et la redirection.[14]

La figure suivante présente Le schéma de la structure générique de l'architecture MVC. [Figure 3.1]

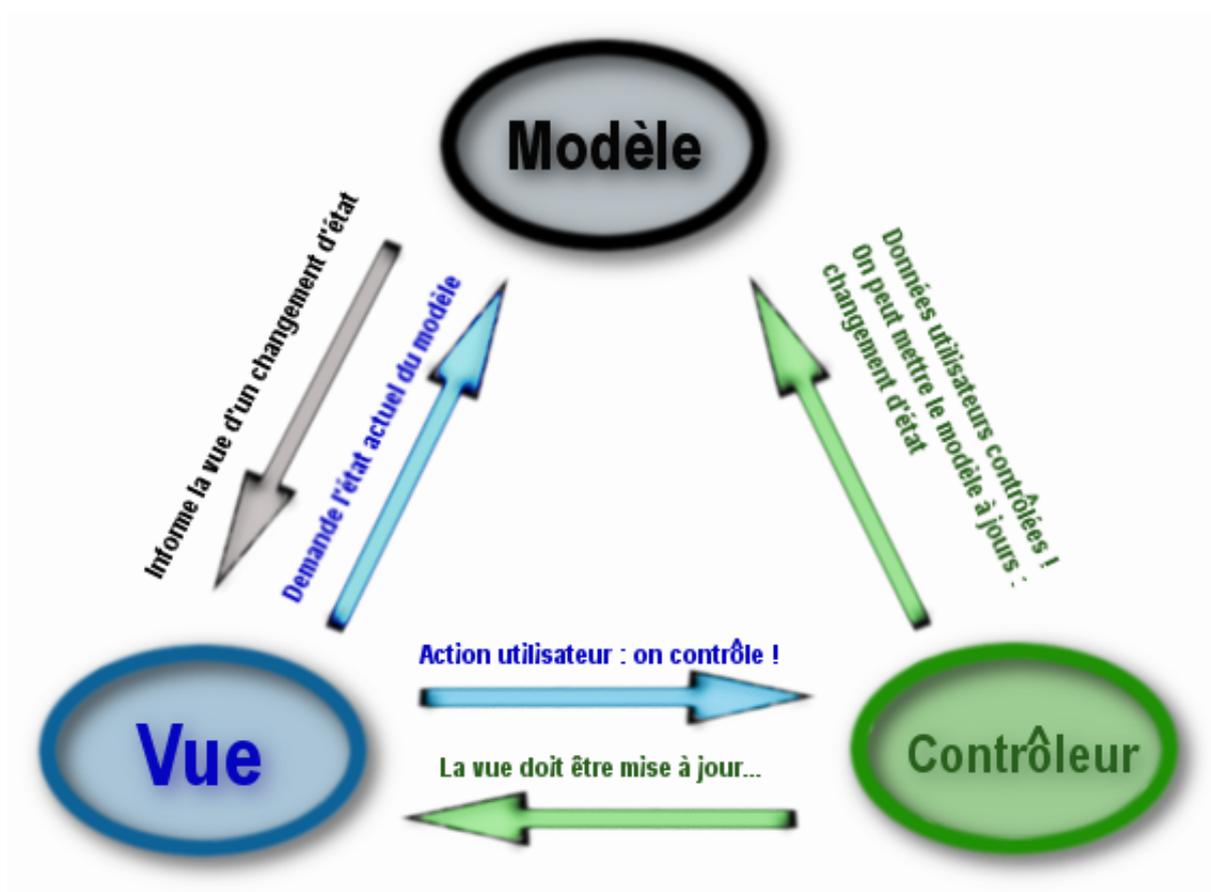


FIGURE 3.1 – Le schéma de la structure générique de l'architecture MVC.

3.2.1.3 Le traitement d'une demande d'un utilisateur dans l'architecture MVC

La figure suivante présente le traitement d'une demande d'un utilisateur dans l'architecture MVC.[Figure 3.2]

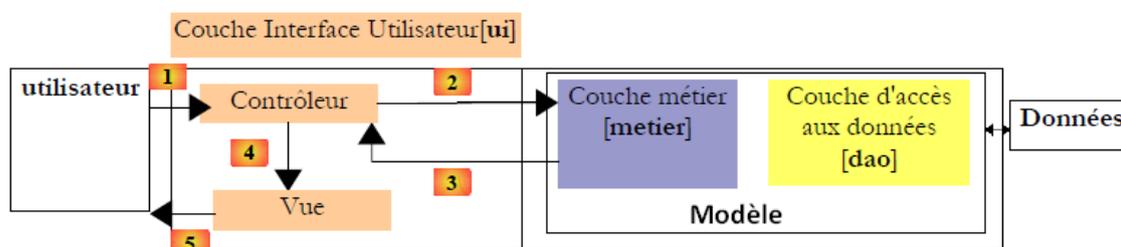


FIGURE 3.2 – Le traitement d'une demande d'un client dans l'architecture MVC.

Le traitement d'une demande d'un client se déroule selon les étapes suivantes :

1. L'utilisateur fait une demande au contrôleur. Ce contrôleur voit passer toutes les demandes des clients. C'est la porte d'entrée de l'application. C'est le C de MVC.
2. Le contrôleur traite cette demande. Pour ce faire, il peut avoir besoin de l'aide de la couche métier, ce qu'on appelle le modèle M dans la structure MVC.
3. Le contrôleur reçoit une réponse de la couche métier. La demande du client a été traitée. Celle-ci peut générer plusieurs réponses possibles, soit une page d'erreurs si la demande n'a pu être traitée correctement sinon une page de confirmation.
4. Le contrôleur choisit la réponse (une vue) à envoyer au client. cette dernière est le plus souvent une page contenant des éléments dynamiques. Le contrôleur fournit ceux-ci à la vue.
5. La vue est envoyée au client. C'est le V de MVC.

Une telle architecture est souvent appelée "architecture 3-tier" ou à 3 niveaux.

3.2.2 Langage de modélisation

3.2.2.1 Définition du langage de modélisation

Un langage de modélisation est un langage qui peut être utilisé pour exprimer des informations, des connaissances ou des systèmes dans une structure qui est définie par un ensemble cohérent de règles.

Les règles sont utilisées pour l'interprétation de la signification des composants dans la structure. Un langage de modélisation peut être graphique ou textuel.

Les langages de modélisation graphiques utilisent des techniques de diagrammes avec des symboles associés à des noms qui représentent les concepts et des lignes qui connectent les symboles

et qui représentent les relations et les diverses autres annotations graphiques pour représenter les contraintes.

Les langages de modélisation textuels utilisent typiquement des mots-clés standardisés accompagnés de paramètres pour rendre les expressions interprétables par les ordinateurs [08].

3.2.2.2 Langage de modélisation UML (Unified Modeling Language)

Définition de l'UML

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. UML unifie à la fois les notations et les concepts orientés objet. Il ne s'agit pas d'une simple notation graphique, car les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage. UML unifie aussi les notations nécessaires aux différentes activités d'un processus de développement et offre, par ce biais, le moyen d'établir le suivi des décisions prises, depuis l'expression des besoins jusqu'au codage [02].

Les caractéristiques du l'UML

UML est une norme :

Il est nécessaire qu'une méthode objet soit définie de manière rigoureuse et unique afin de lever les ambiguïtés. De nombreuses méthodes objet ont été définies, mais aucune n'a su s'imposer en raison du manque de standardisation. A partir de 1997, UML est devenue une norme de l'OMG (Object Management Group), ce qui lui a permis de s'imposer en tant que méthode de développement objet et être reconnue et utilisée par de nombreuses entreprises.

UML est un langage de modélisation objet :

Il permet d'exprimer, d'élaborer et de modéliser au sens de la théorie des langages, de ce fait il contient les éléments constitutifs de ce derniers : concepts, une syntaxe et une sémantique.

UML décrit un méta modèle :

La puissance et l'intérêt d'UML est qu'il normalise la sémantique des concepts qu'il véhicule, il repose sur un méta modèle pour permettre à n'importe qui de déchiffrer son intention de manière non équivoque, il est donc primordiale de s'accorder sur la sémantique des éléments de modélisation, bien avant de s'intéresser à la manière de les présenter.

3.2.2.3 Les points forts et les points faibles du l'UML :

— Les points forts du l'UML :

UML est un langage formel et normalisé : Il permet le gain de précision, encourage l'utilisation d'outils et constitue à cet effet un gage de stabilité [12].

UML est un support de communication performant : Il cadre l'analyse et facilite la compréhension de représentations abstraites complexes. Son caractère polyvalent et sa souplesse en font un langage universel [12].

— **Les points faibles du l’UML :**

La mise en pratique d’UML nécessite un apprentissage et passe par une période d’adaptation.

UML n’est pas à l’origine des concepts objets, mais en constitue une étape majeure, car il unifie les différentes approches et en donne une définition plus formelle.

L’intégration d’UML dans un processus n’est pas triviale et améliorer un processus est une tâche complexe et longue. Les auteurs d’UML sont tout à fait conscients de l’importance du processus, mais l’acceptabilité industrielle de la modélisation objet passe d’abord par la disponibilité d’un langage d’analyse [12].

3.3 Conception détaillée

3.3.1 Diagrammes de cas d’utilisation

3.3.1.1 Définition

Les rôles des diagrammes de cas d’utilisation sont de recueillir, d’analyser et d’organiser les besoins, ainsi que de recenser les grandes fonctionnalités d’un système. Il s’agit donc de la première étape UML pour la conception d’un système.

Un cas d’utilisation est une unité cohérente représentant une fonctionnalité visible de l’extérieur. Il réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l’acteur qui l’initie. Un cas d’utilisation modélise donc un service rendu par le système, sans imposer le mode de réalisation de ce service [03].

3.3.1.2 Composition du diagramme de cas

Le diagramme de cas se compose de trois éléments principaux :

- **Un Acteur** : c’est l’idéalisaiton d’un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système. Il se représente par un petit bonhomme avec son nom inscrit dessous.



- **Un cas d’utilisation** : Un cas d’utilisation représente une fonctionnalité du système. celle-ci est définie par une action déclenchante, un ou plusieurs déroulements possibles et éventuellement une fin. Le cas d’utilisation se représente par une ellipse contenant un nom décrivant la fonctionnalité et éventuellement un stéréotype.

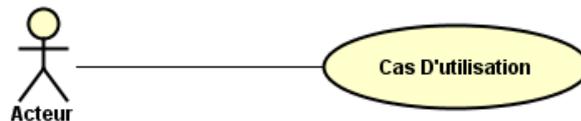


Les relations :

— Association entre cas et acteur :

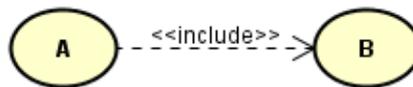
Les acteurs demandant des services aux systèmes, ils sont le plus souvent à l'initiative des échanges avec le système.

On représente une association entre un acteur et un cas d'utilisation par une ligne pleine.

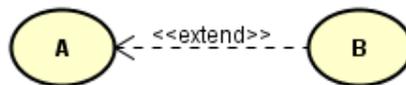


— Les Relations entre les cas d'utilisation :

1. **Inclusion** : B est une partie obligatoire de A et on lit A inclut B (dans le sens de la flèche).



2. **Extension** : B est une partie optionnelle de A et on lit B étend A (dans le sens de la flèche).



3. **Généralisation** : le cas A est une généralisation du cas B et on lit B est une sorte de A.



— Relations entre acteurs :

La seule relation possible entre deux acteurs est la généralisation : un acteur A est une généralisation d'un acteur B si l'acteur A peut-être substitué par l'acteur B. Dans ce cas, tous les cas d'utilisation accessibles à A le sont aussi à B, mais l'inverse n'est pas vrai. Le symbole utilisé pour la généralisation entre acteurs est une flèche avec un trait plein dont la pointe est un triangle fermé désignant l'acteur le plus général.



3.3.1.3 Diagrammes des cas d'utilisations de notre application web

Diagramme de cas d'utilisation d'un visiteur : le diagramme suivant [Figure 3.3] représente le cas d'utilisation d'un visiteur

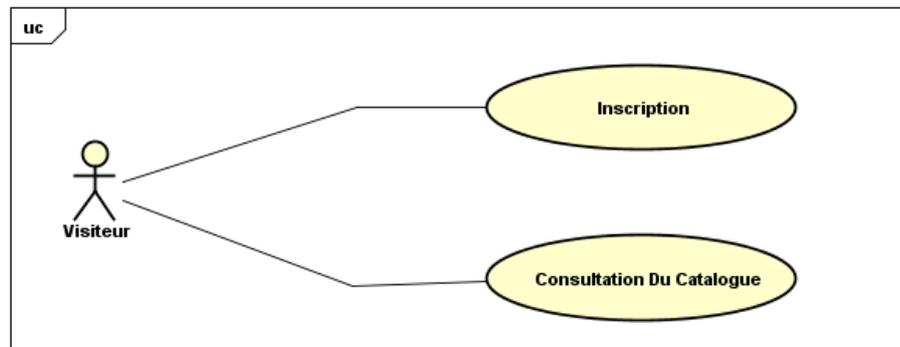


FIGURE 3.3 – Diagramme de cas d'utilisation d'un visiteur.

Le visiteur ou bien l'internaute ne possède que la possibilité de consulter les catalogues des produits et la possibilité de s'inscrire pour devenir client sur notre application web.

Diagramme de cas d'un client : le diagramme suivant [Figure 3.4] représente le cas d'utilisation d'un client

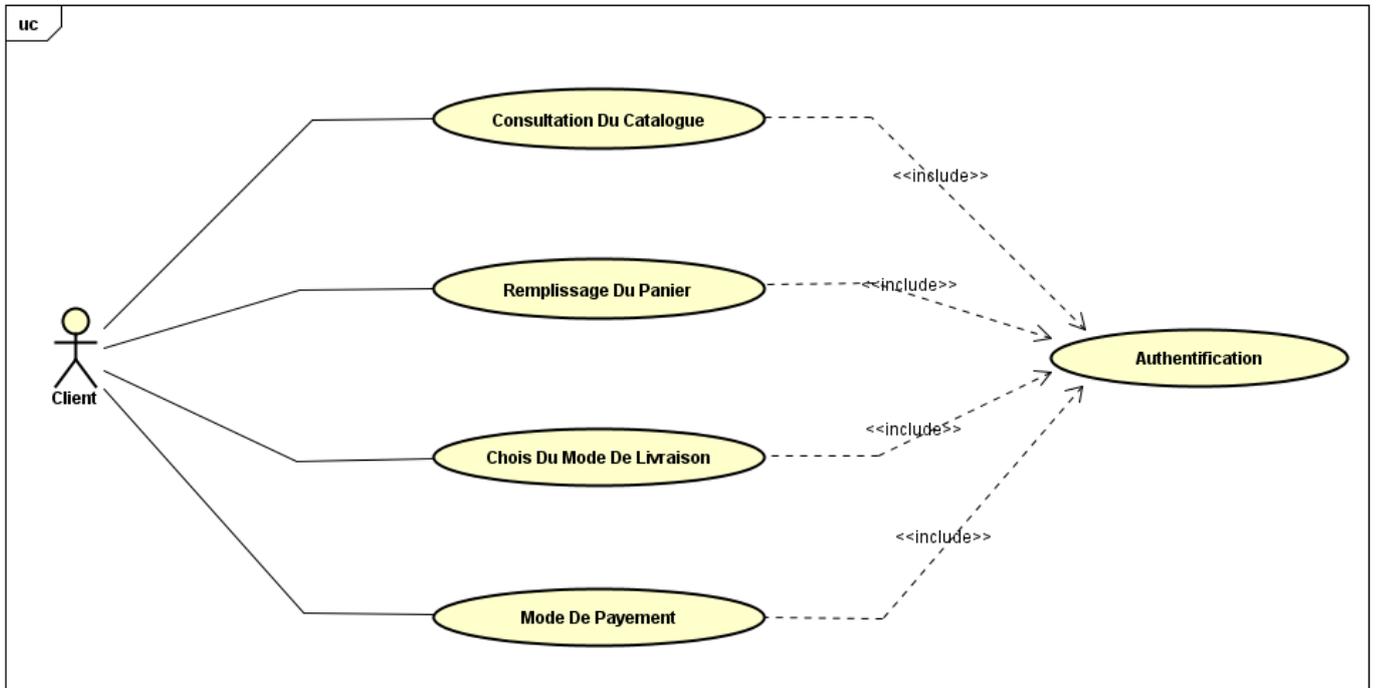


FIGURE 3.4 – Diagramme de cas d'un client.

Après l'inscription du visiteur, il deviendra client. Il aura la possibilité de continuer toute une procédure d'achat en ligne sur notre application web après une authentification.

Diagramme de cas de l'administrateur : le diagramme suivant [Figure 3.5] représente le cas d'utilisation de l'administrateur

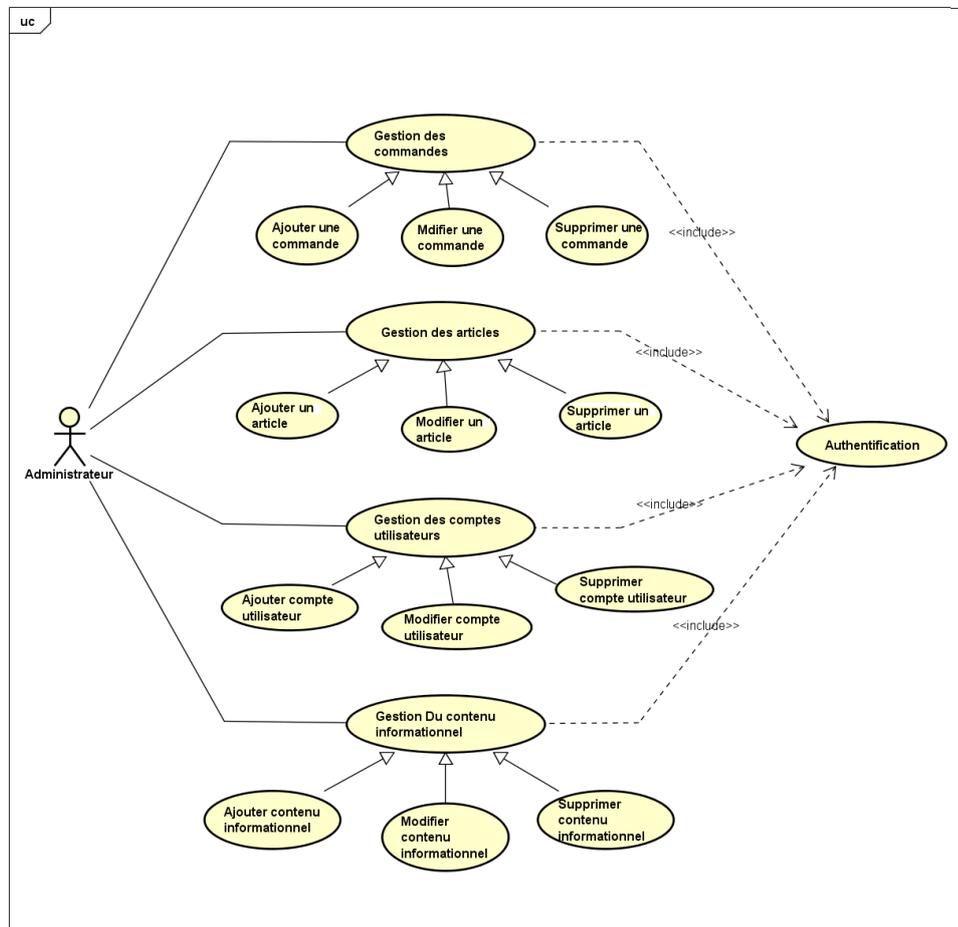


FIGURE 3.5 – Diagramme de cas de l'administrateur.

L'administrateur est la personne qui est en charge de la mise à jour de l'application. Après une authentification, il a la possibilité de gérer les commandes effectuées par les clients, il est responsable de la gestion des produits et de la gestion des clients.

3.3.2 Diagrammes d'activités :

3.3.2.1 Définition

Une activité définit un comportement décrit par une série organisée d'unités dont les éléments simples sont les actions. Le flot d'exécution est modélisé par des nœuds reliés par des arcs (transitions). Le flot de contrôle reste dans l'activité jusqu'à ce que le traitement soit terminé.

Une activité est un comportement et peut comporter des paramètres en entrée ou en sortie, ainsi que des variables locales au même titre qu'une opération.

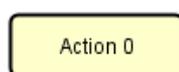
Les diagrammes d'activités permettent ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation [02].

3.3.2.2 Composition de diagramme d'activité :

Après avoir expliqué le diagramme d'activités et son fonctionnement, nous abordons les nœuds qui sont essentiels dans un diagramme d'activités.

Nœud d'activité : représente une exécution d'un mécanisme, autrement dit, un déroulement d'étapes séquentielles.

Un nœud d'activité est représenté par un rectangle aux coins arrondis qui contient sa description textuelle.

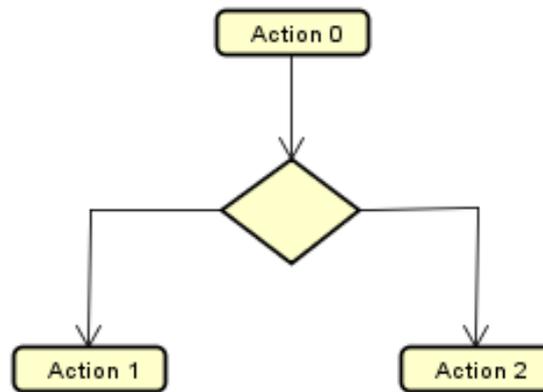


Une transition : Le passage d'une activité vers une autre est matérialisé par une transition. Graphiquement les transitions sont représentées par des flèches en traits pleins qui connectent les activités entre elles. Elles sont déclenchées dès que l'activité source est terminée et provoquent automatiquement et immédiatement le début de la prochaine activité à déclencher (l'activité cible).

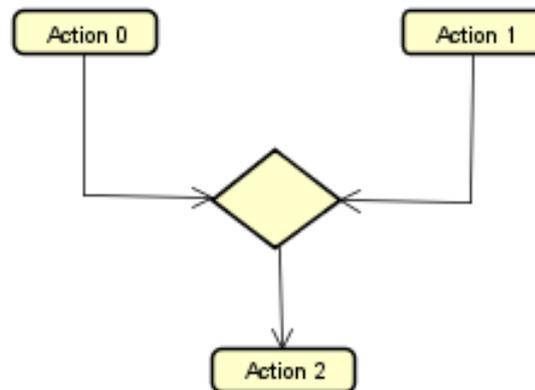


Nœud de décision : C'est un nœud de contrôle qui permet de faire un choix entre plusieurs flots sortants. Il possède un arc entrant et plusieurs arcs sortants, accompagnés de conditions de garde pour conditionner le choix. Quand le nœud de décision est atteint et qu'aucun arc en aval n'est franchissable (ce qui veut dire qu'aucune condition n'est vraie), cela signifie que le modèle est mal formé.

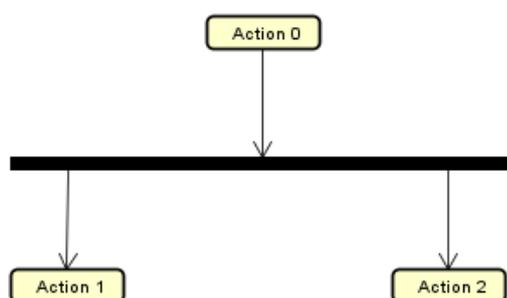
Un nœud de décision est représenté par un losange.



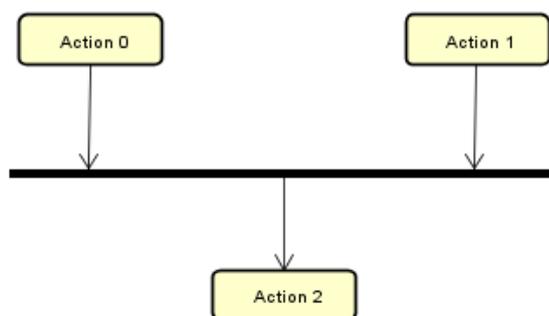
Nœud de fusion : Un nœud de fusion est un nœud de contrôle rassemblant plusieurs flots alternatifs entrants en un seul flot sortant. On ne l'utilise pas pour synchroniser des flots concurrents mais pour accepter un flot parmi plusieurs. Un nœud de fusion est représenté par un losange comme le nœud de décision.



Nœud de bifurcation : Un nœud de bifurcation est un nœud de contrôle qui sépare un flot ou plusieurs flots concurrents. Il possède donc un arc entrant et plusieurs arcs sortants. Il est représenté par un trait plein épais.



Nœud d'union : C'est un nœud de contrôle qui synchronise des flots multiples. Il possède plusieurs arcs entrants et un seul arc sortant. Lorsque tous les arcs entrants sont activés, l'arc sortant l'est aussi également. Un nœud d'union est aussi représenté par un trait plein épais.



3.3.2.3 Les diagrammes d'activités de notre application :

- **Diagramme d'inscription :** La phase d'inscription est la phase qui permet à un visiteur de passer d'un simple visiteur à un client pour qu'il puisse faire ces achats en ligne. Dans ce diagramme, l'utilisateur demande une inscription au système, ce dernier lui affiche le formulaire d'inscription, après le remplissage du formulaire par le visiteur, le système vérifie si les données saisies sont incorrectes ou bien existes déjà dans la base de données, alors le système lui demande de remplir à nouveau le formulaire sinon, l'inscription est validée.[Figure 3.6]

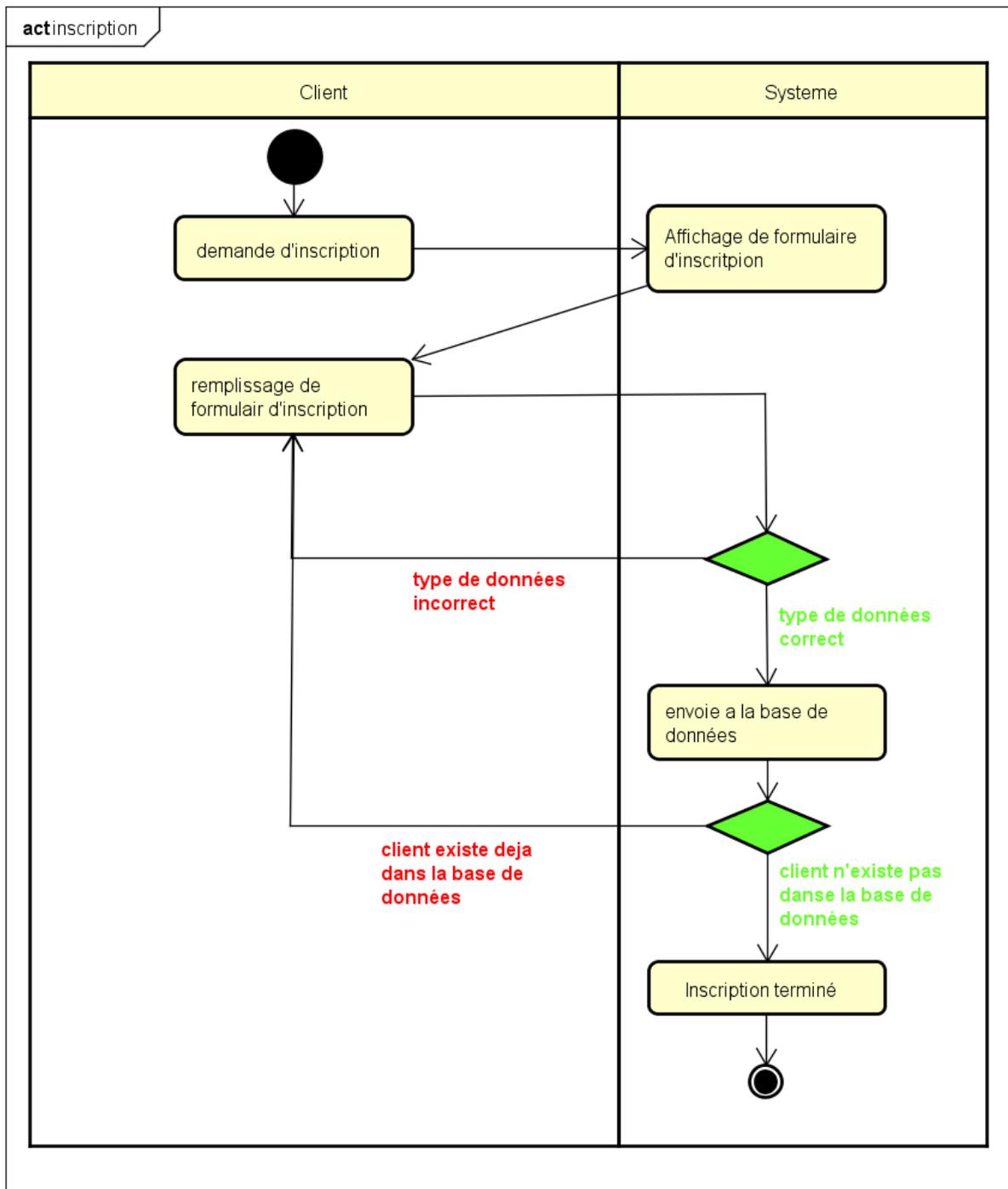


FIGURE 3.6 – Diagramme d’activité d’inscription.

- **Diagramme d’authentification** : L’authentification pour un système informatique est un processus permettant au système de s’assurer de la légitimité de la demande d’accès faite par une entité (être humain ou un autre système...) afin de lui autoriser l’accès à des ressources du système (systèmes, réseaux, applications...) conformément au paramétrage du contrôle d’accès.

Dans ce diagramme, l'utilisateur demande l'authentification au système, ce dernier lui affiche le formulaire d'authentification, après le remplissage du formulaire par l'utilisateur, le système vérifié si le login et le mot de passe saisis existent dans la base de données alors, l'authentification est réussite, sinon le système affiche un message d'erreur et lui demande de remplir à nouveau le formulaire. [Figure 3.7]

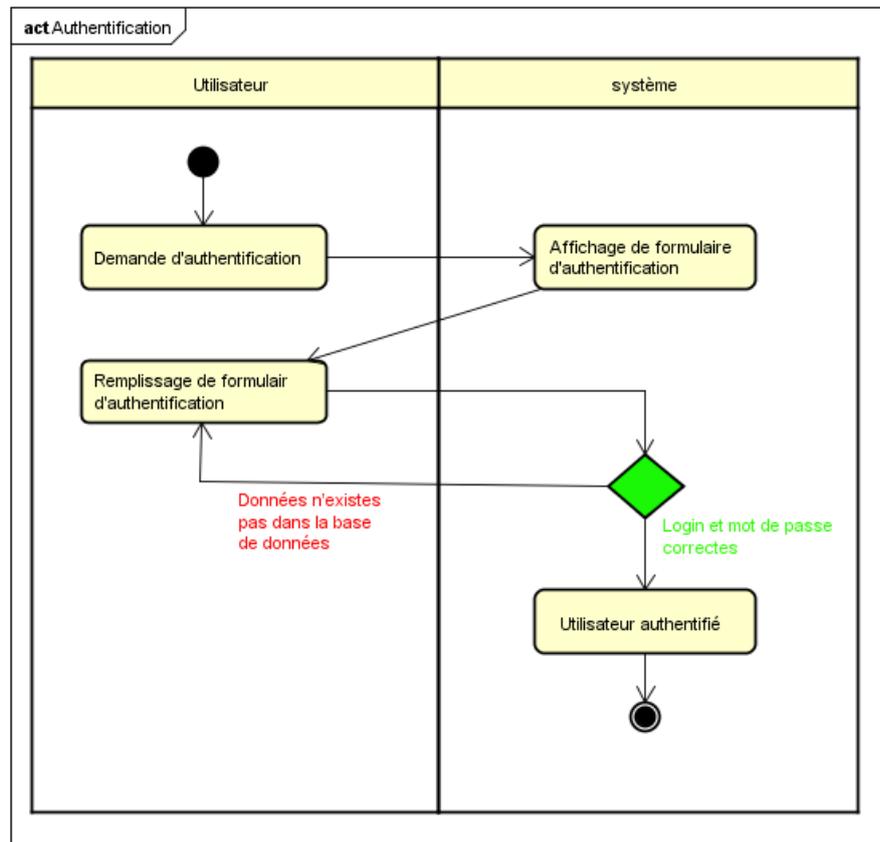


FIGURE 3.7 – Diagramme d'activité d'authentification.

- **Diagramme d'ajout d'un article** : Pour l'ajout d'un article à la base de données, l'administrateur de l'application, demande au système l'ajout d'un article, le système lui affiche le formulaire, après le remplissage du formulaire par l'administrateur, le système vérifie si les types de données saisies sont correctes, si c'est le cas alors l'ajout est validée et l'article est enregistré, sinon, le système demande à l'administrateur de remplir à nouveau le formulaire. [Figure 3.8]

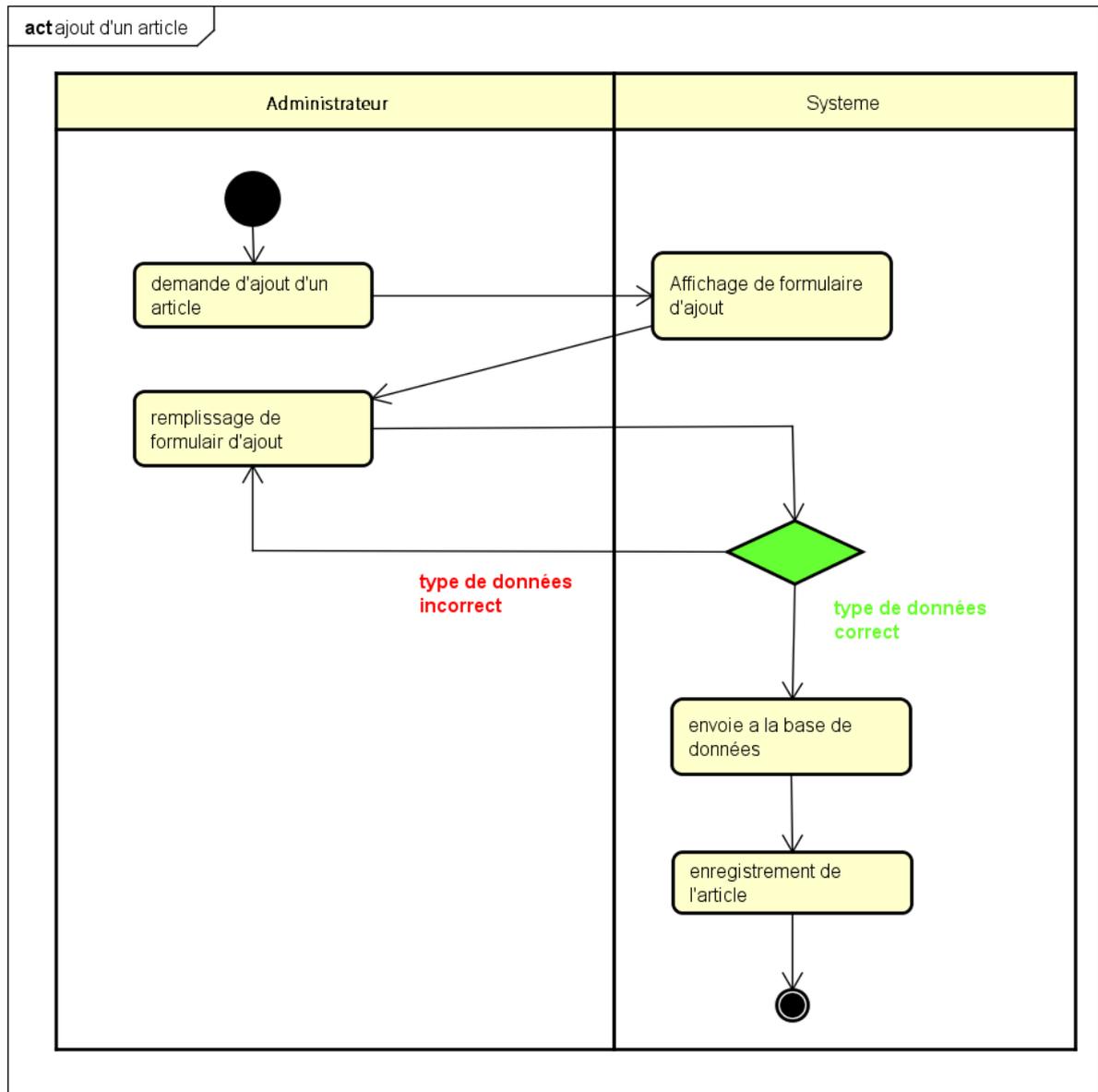


FIGURE 3.8 – Diagramme d'activité d'ajout d'un article.

3.3.3 Diagrammes des séquences

3.3.3.1 Définition

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique.

Le diagramme de séquence permet de montrer les interactions d'objets dans le cadre d'un scénario d'un diagramme des cas d'utilisation. Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système. Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets [02].

3.3.3.2 Composition de diagrammes des séquences :

- **Les Lignes de vie** : Une ligne verticale en trait pointillés, qui représente la séquence des événements, produite par un participant, pendant une interaction, alors que le temps progresse en bas de ligne. Ce participant peut être une instance d'une classe, un composant ou un acteur.
- **Les Messages** : Les principales informations contenues dans un diagramme de séquence sont les messages échangés entre les lignes de vie :
 - Ils sont représentés par des flèches.
 - Ils sont présentés du haut vers le bas le long des lignes de vie, dans un ordre chronologique.
 - Un message définit une communication particulière entre des lignes de vie (objets ou acteurs).

Plusieurs types de messages existent, les plus communs sont :

- L'invocation d'une opération : message synchrone (appel d'une méthode de l'objet cible).
 - L'envoi d'un signal : message asynchrone (typiquement utilisé pour la gestion événementielle).
 - La création ou la destruction d'une instance de classe au cours du cycle principal.
- Un message synchrone bloque l'expéditeur jusqu'à la réponse du destinataire. Le flot de contrôle passe de l'émetteur au récepteur. Si un objet A envoie un message synchrone à un objet B, A reste bloqué tant que B n'a pas terminé.
- Un message asynchrone n'est pas bloquant pour l'expéditeur. Le message envoyé peut être pris en compte par le récepteur à tout moment ou ignoré.
- **Commentaires** : on peut attacher un commentaire à n'importe quel point sur une ligne de vie.
 - **Occurrences d'exécutions** : représente la période d'exécution d'une opération.
 - **Fragments** : Un fragment (ou cadre) permet d'identifier une sous-partie d'une interaction afin que celle-ci soit référencées par d'autres interactions ou de lui spécifier des conditions particulières d'exécution (boucle, optionnel, ...).
- quelques exemples des fragments :
- **sd** : fragment du diagramme de séquence en entier.
 - **alt** : fragment alternatif (Si ... Alors ... Sinon ...).
 - **opt** : fragment optionnel.
 - **ref** : référence à une interaction dans un autre diagramme.

3.3.3.3 Les diagrammes des séquences de notre application web :

Diagramme de séquence d'inscription : Le diagramme représenté dans la figure ci-après [Figure 3.9] décrit les scénarios possibles lors d'une inscription d'un visiteur. En effet :
Le visiteur demande l'accès à l'interface inscription, la vue reçoit sa demande et la renvoie au

contrôleur comme étant une requête, la réponse du contrôleur est le lien de l'interface inscription, le visiteur reçoit le formulaire d'inscription et le remplit, la vue vérifie les types des données, si ils ne sont pas valides donc un message d'erreur lui sera affiché au niveau de la vue sinon, le contrôleur récupère les données saisies et crée l'objet client et il l'envoi au modèle, ce dernier le renvoi à la base de données. Une vérification de l'existence du client au niveau de la base de données, si le client n'existe pas alors il sera enregistré dans la base de données et la vue affiche un message de confirmation de l'inscription, sinon, la vue lui affiche un message d'erreur (client existe déjà).

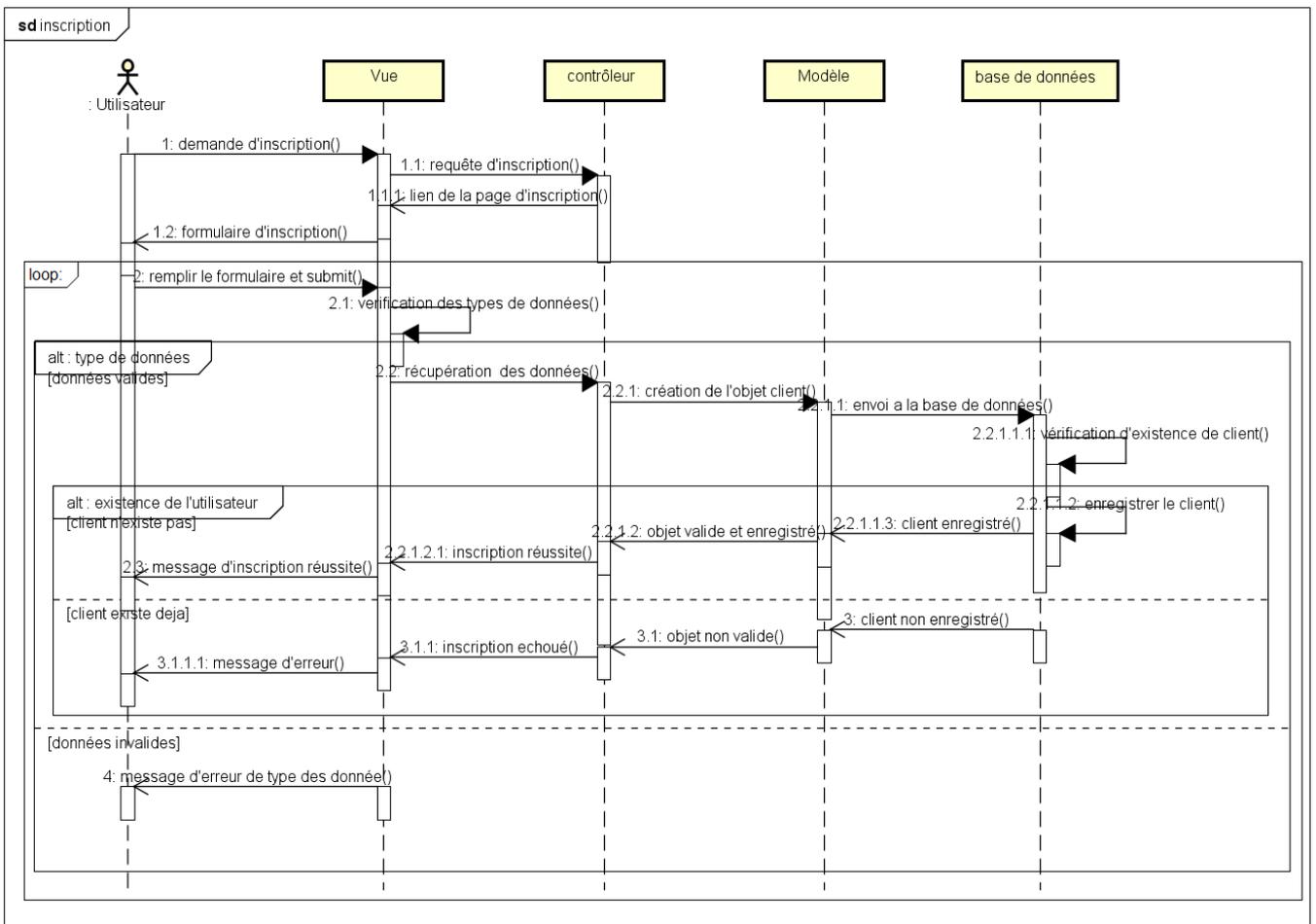


FIGURE 3.9 – Diagramme de séquence d'inscription.

Diagramme de séquence d'authentification : Le diagramme suivant [Figure 3.10] représente les scénarios possibles lors d'une authentification de l'utilisateur. En effet :

L'utilisateur demande l'accès à l'interface authentification, la vue reçoit sa demande et la renvoie au contrôleur comme étant une requête, la réponse du contrôleur est le lien de l'interface authentification, l'utilisateur reçoit le formulaire authentification et le remplit (login et mot de

Le client fait une demande d'ajout d'un produit dans son panier, la vue récupère sa demande et envoie la requête d'ajout de produit au contrôleur, ce dernier ajoute le produit dans la session panier, la vue reçoit la réponse de contrôleur et affiche au client le produit sélectionné dans le panier.

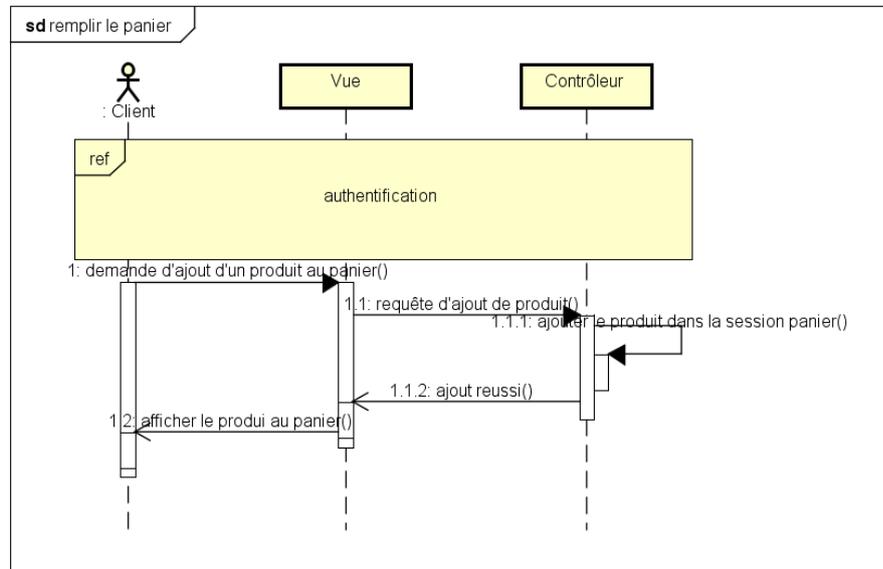


FIGURE 3.11 – Diagramme de séquence de remplissage de panier.

Diagramme de séquence effectuer une commande

Le diagramme suivant [Figure 3.12] représente les opérations possibles quand un client effectue sa commande en ligne. En effet :

Le diagramme ci-dessous fait la référence au diagramme de séquence remplissage de panier [Figure 3.11].

Pour qu'un client puisse effectuer une commande, il doit d'abord choisir des produits et remplir son panier.

La vue envoie la requête au contrôleur (demande commande), ce dernier enregistre les produits sélectionnés depuis le panier et le modèle crée la commande dans la base de données, cette dernière enregistre cette commande dans la table commande et envoie un message au modèle comme quoi elle est enregistrée, le contrôleur reçoit un message depuis le modèle étant l'objet est valide et enregistré et confirme pour la vue la réussite et la validité de la commande, la vue affiche pour le client un message de confirmation de la commande et lui donne un choix pour l'impression de la commande.

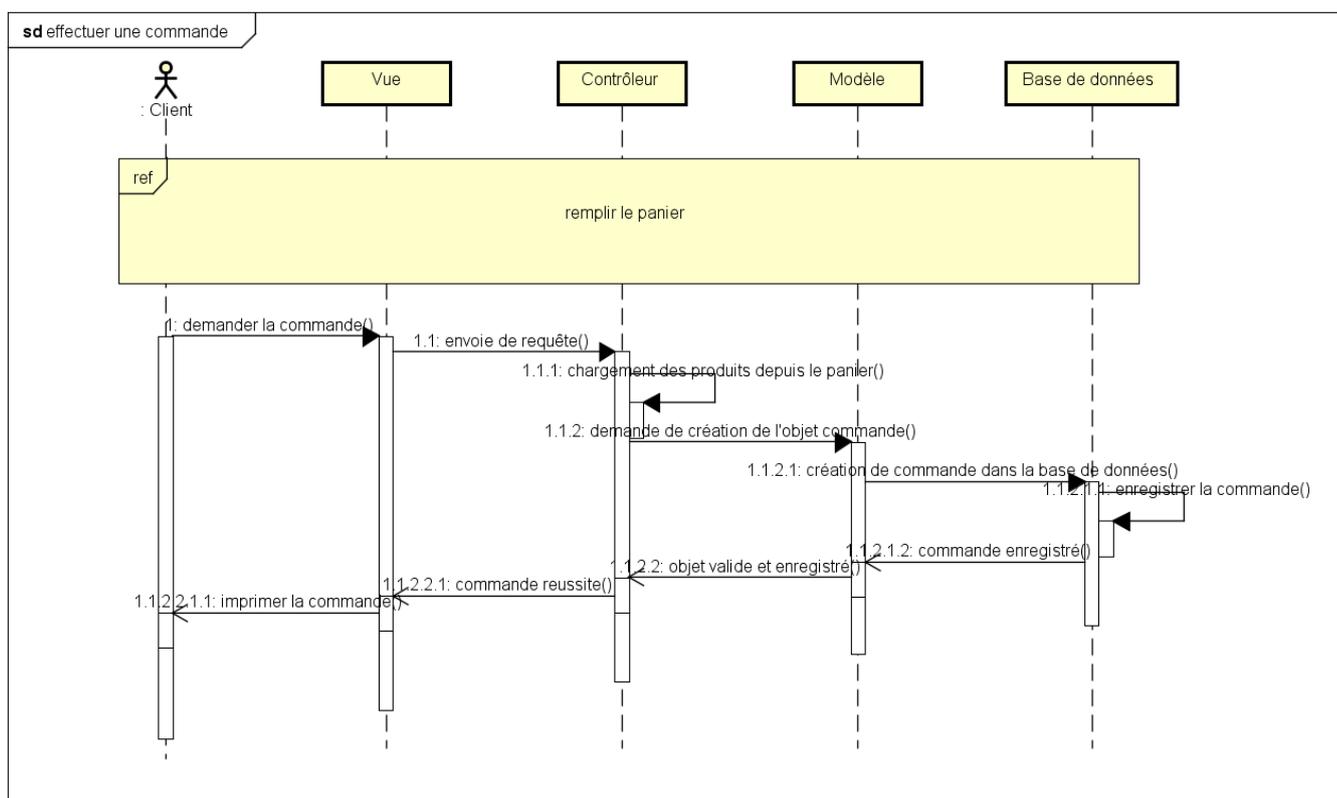


FIGURE 3.12 – Diagramme de séquence effectuer une commande.

Diagramme de séquence d'ajout d'un produit :

Le diagramme suivant [Figure 3.13] représente les différentes opérations quand l'administrateur ajoute un produit à la base de données. En effet :

L'administrateur demande le formulaire d'ajout d'un produit, la vue affiche le formulaire, ensuite il remplit et valide le formulaire, la vue vérifie les types de données saisies si elles sont correctes, si ce n'est pas le cas alors elle lui affiche un message d'erreur, sinon, le contrôleur récupère les données saisies et les envoie au modèle comme un objet, ce dernier renvoie l'objet à la base de données, celle-ci vérifie l'existence de l'objet, si l'objet n'existe pas dans la base de données alors l'enregistrement du produit sera effectué au niveau de la base de données, le contrôleur reçoit un message depuis le modèle comme quoi l'objet valide et enregistré et renvoie le message à la vue, ensuite elle informe l'administrateur de la réussite de l'ajout du produit par un message de confirmation. Sinon si le produit existe déjà dans la base de données alors le produit ne sera pas enregistré. La vue affiche un message d'erreur et informe l'administrateur de l'existence du produit dans la base de données.

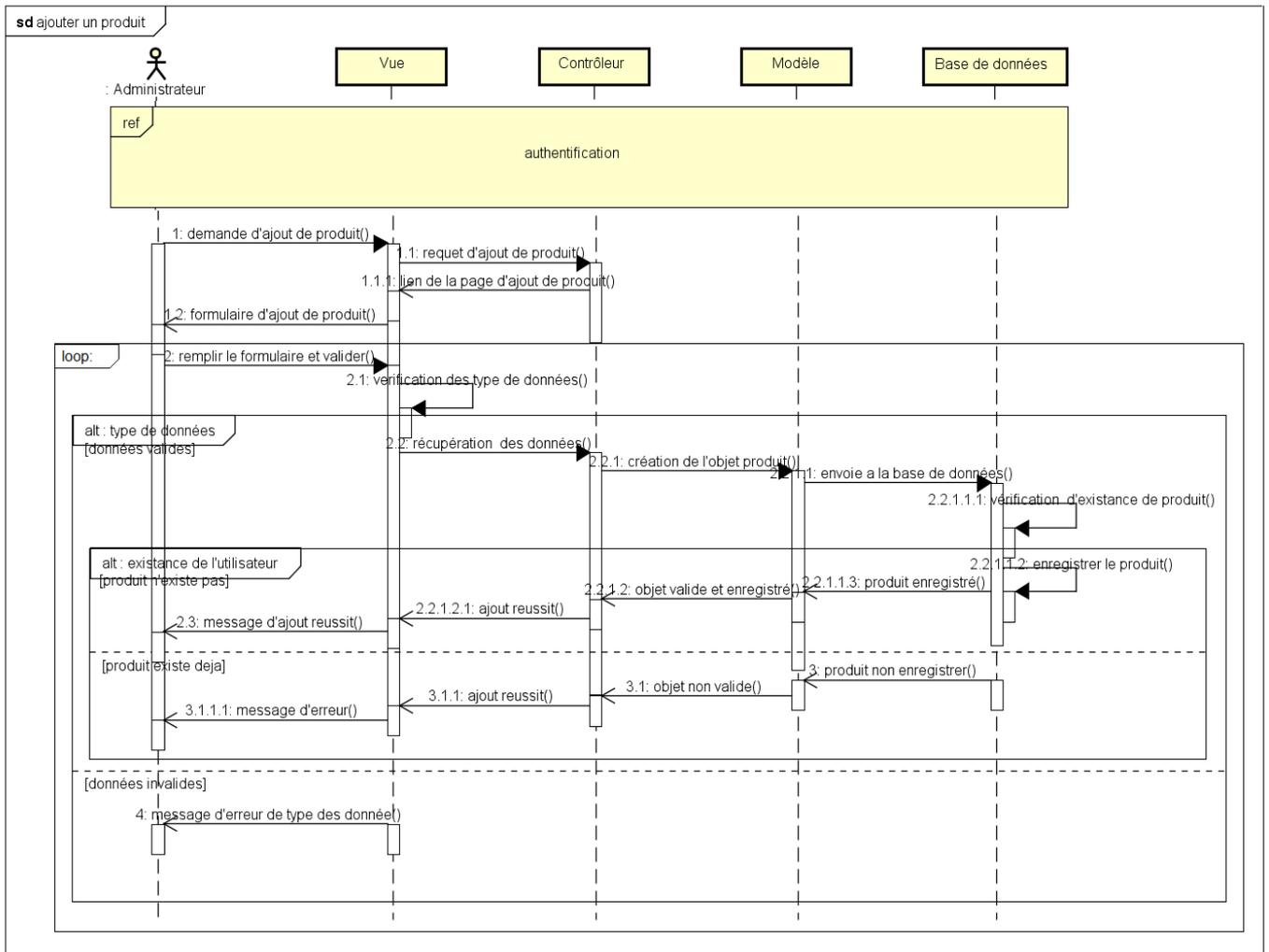


FIGURE 3.13 – Diagramme de séquence d'ajout d'un produit.

3.3.4 Diagramme de classes

3.3.4.1 Définition

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation.

Un diagramme de classes fournit une vue globale d'un système en présentant ses classes, interfaces et collaborations, et les relations entre elles. Les diagrammes de classes sont statiques : ils affichent ce qui interagit mais pas ce qui se passe pendant l'interaction.

Alors que le diagramme de cas d'utilisation montre un système du point de vue des acteurs, le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation [02].

3.3.4.2 Principales composants de diagramme des classes

En général un diagramme de classe peut contenir les éléments suivants :

Les classes : Une classe est la description formelle d'un ensemble d'objets ayant une sémantique et des caractéristiques communes.

Une classe est un classeur. Elle est représentée par un rectangle divisé en trois compartiments. Le premier indique le nom de la classe, le deuxième ses attributs et le troisième ses opérations [02].

Les attributs : les attributs représentent les données encapsulées dans les objets des classes. Chacune de ces informations est définie par un nom, un type de données, une visibilité et peut être initialisé. Le nom de l'attribut doit être unique dans la classe [02].

Les relations : il existe plusieurs relations reliant les différentes classes de diagramme, parmi eux on trouve :

- La relation d'héritage est un principe de division par généralisation et spécialisation, représenté par un trait reliant les deux classes et dont l'origine (classe mère) se distingue de l'autre extrémité (classe fille) par un triangle. La classe fille hérite de la classe mère de tous les attributs et méthodes, qu'ils soient publics, protégés ou privés.
- La relation d'association est une connexion sémantique entre deux classes (relation logique). Une association peut être nommée. L'invocation d'une méthode est une association. Elle peut être binaire, dans ce cas elle est représentée par un simple trait, ou n-aire, les classes sont reliées à un losange par des traits simples. Ces relations peuvent être nommées.
- L'agrégation est une association avec relation de subordination, représentée par un trait reliant les deux classes et dont l'origine se distingue de l'autre extrémité (la classe subordonnée) par un losange vide [02].

3.3.4.3 Diagramme de classes de notre application :

Le diagramme suivant [Figure 3.14] représente les différentes classes et leurs relations qui constituent notre base de données.



FIGURE 3.14 – Diagramme de classes de notre application.

3.3.5 Script de création de la base de données :

La génération de la base de données au format SQL standard donne le code SQL suivant [Figure 3.15] :

```

1  -- MySQL Workbench Forward Engineering
2
3  SET @OLD_UNIQUE_CHECKS=@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
4  SET @OLD_FOREIGN_KEY_CHECKS=@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
5  SET @OLD_SQL_MODE=@SQL_MODE, SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';
6
7  -----
8  -- Schema e_commerce
9  -----
10 -----
11 -- Schema e_commerce
12 -----
13 -----
14 CREATE SCHEMA IF NOT EXISTS `e_commerce` DEFAULT CHARACTER SET utf8 ;
15 USE `e_commerce` ;
16
17 -----
18 -- Table `e_commerce`.`magasin`
19 -----
20 CREATE TABLE IF NOT EXISTS `e_commerce`.`magasin` (
21   `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
22   `nom` VARCHAR(45) NOT NULL,
23   `adresse` VARCHAR(200) NOT NULL,
24   PRIMARY KEY (`id`))
25 ENGINE = InnoDB;
26
27 -----
28 -- Table `e_commerce`.`categorie`
29 -----
30
31 CREATE TABLE IF NOT EXISTS `e_commerce`.`categorie` (
32   `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
33   `nom` VARCHAR(45) NOT NULL,
34   PRIMARY KEY (`id`))
35 ENGINE = InnoDB;
36
37 -----
38 -- Table `e_commerce`.`article`
39 -----
40
41 CREATE TABLE IF NOT EXISTS `e_commerce`.`article` (
42   `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
43   `name` VARCHAR(45) NOT NULL,
44   `price` DECIMAL(8,2) NOT NULL,
45   `description` TINYTEXT NULL,
46   `mis_a_jour` TIMESTAMP NULL,
47   `magasin_id` INT UNSIGNED NOT NULL,
48   `categorie_id` INT UNSIGNED NOT NULL,
49   PRIMARY KEY (`id`),
50   INDEX `fk_product_shop_idx` (`magasin_id` ASC),
51   INDEX `fk_product_sub_category1_idx` (`categorie_id` ASC),
52   CONSTRAINT `fk_product_shop`
53     FOREIGN KEY (`magasin_id`)
54     REFERENCES `e_commerce`.`magasin` (`id`)
55     ON DELETE NO ACTION
56     ON UPDATE NO ACTION,
57   CONSTRAINT `fk_product_sub_category1`
58     FOREIGN KEY (`categorie_id`)
59     REFERENCES `e_commerce`.`categorie` (`id`)
60     ON DELETE NO ACTION
61     ON UPDATE NO ACTION)
62 ENGINE = InnoDB;
63
64 -----
65 -- Table `e_commerce`.`client`
66 -----
67
68 CREATE TABLE IF NOT EXISTS `e_commerce`.`client` (
69   `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
70   `nom` VARCHAR(45) NOT NULL,
71   `prénom` VARCHAR(45) NOT NULL,
72   `tel` VARCHAR(14) NOT NULL,
73   `adresse` VARCHAR(45) NOT NULL,
74   `ville_region` DECIMAL(2) NOT NULL,
75   `cn_numero` VARCHAR(19) NOT NULL,
76   `email` VARCHAR(45) NOT NULL,
77   `mot_de_passe` VARCHAR(256) NOT NULL,
78   PRIMARY KEY (`id`),
79   UNIQUE INDEX `cc_number_UNIQUE` (`cn_numero` ASC),
80   UNIQUE INDEX `email_UNIQUE` (`email` ASC))
81 ENGINE = InnoDB;
82
83 -----
84 -- Table `e_commerce`.`commande`
85 -----
86
87 CREATE TABLE IF NOT EXISTS `e_commerce`.`commande` (
88   `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
89   `montant` DECIMAL(8,2) NOT NULL,
90   `date_creation` TIMESTAMP NOT NULL,
91   `numero_confirmation` INT UNSIGNED NOT NULL,
92   `id_client` INT UNSIGNED NOT NULL,
93   PRIMARY KEY (`id`),
94   INDEX `fk_customer_order_customer1_idx` (`id_client` ASC),
95   CONSTRAINT `fk_customer_order_customer1`
96     FOREIGN KEY (`id_client`)
97     REFERENCES `e_commerce`.`client` (`id`)
98     ON DELETE NO ACTION
99     ON UPDATE NO ACTION)
100 ENGINE = InnoDB;
101
102 -----
103 -- Table `e_commerce`.`articles_dans_commande`
104 -----
105
106 CREATE TABLE IF NOT EXISTS `e_commerce`.`articles_dans_commande` (
107   `article_id` INT UNSIGNED NOT NULL,
108   `commande_id` INT UNSIGNED NOT NULL,
109   `quantite` SMALLINT NOT NULL,
110   PRIMARY KEY (`article_id`, `commande_id`),
111   INDEX `fk_product_has_customer_order_customer_order1_idx` (`commande_id` ASC),
112   INDEX `fk_product_has_customer_order_product1_idx` (`article_id` ASC),
113   CONSTRAINT `fk_product_has_customer_order_product1`
114     FOREIGN KEY (`article_id`)
115     REFERENCES `e_commerce`.`article` (`id`)
116     ON DELETE NO ACTION
117     ON UPDATE NO ACTION,
118   CONSTRAINT `fk_product_has_customer_order_customer_order1`
119     FOREIGN KEY (`commande_id`)
120     REFERENCES `e_commerce`.`commande` (`id`)
121     ON DELETE NO ACTION
122     ON UPDATE NO ACTION)
123 ENGINE = InnoDB;
124
125 -----
126 -- Table `e_commerce`.`user_group`
127 -----
128
129 CREATE TABLE IF NOT EXISTS `e_commerce`.`user_group` (
130   `iduser_group` INT UNSIGNED NOT NULL AUTO_INCREMENT,
131   `email` VARCHAR(45) NOT NULL,
132   `name` VARCHAR(45) NOT NULL,
133   PRIMARY KEY (`iduser_group`))
134 ENGINE = InnoDB;
135
136
137 SET SQL_MODE=@OLD_SQL_MODE;
138 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
139 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
140

```

FIGURE 3.15 – Le SQL généré de la base de données.

3.3.6 Les chartes graphiques de notre application web :

Une charte graphique aboutit généralement à la création de modèles de pages (en anglais Template) servant comme des gabarits pour la création des applications web ou des sites web.

Les Template sont des interfaces créés sous forme de calques ou bien des pages web représentant le squelette graphique des pages.[03]

Les figures suivantes représentent quelques exemples des chartes graphiques de notre application web.[Figure 3.16][Figure 3.17][Figure 3.18]

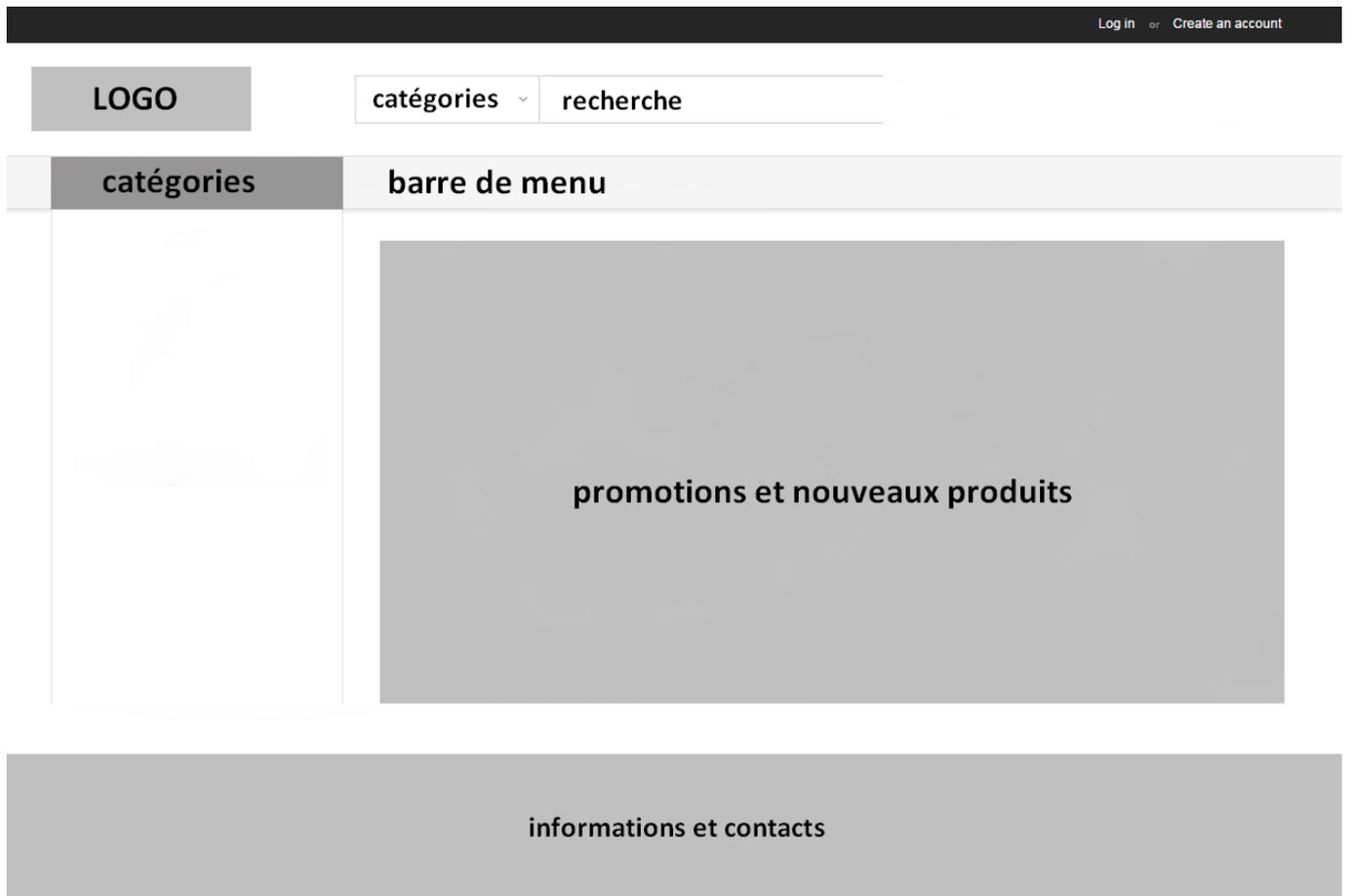


FIGURE 3.16 – La charte graphique de l'accueil.

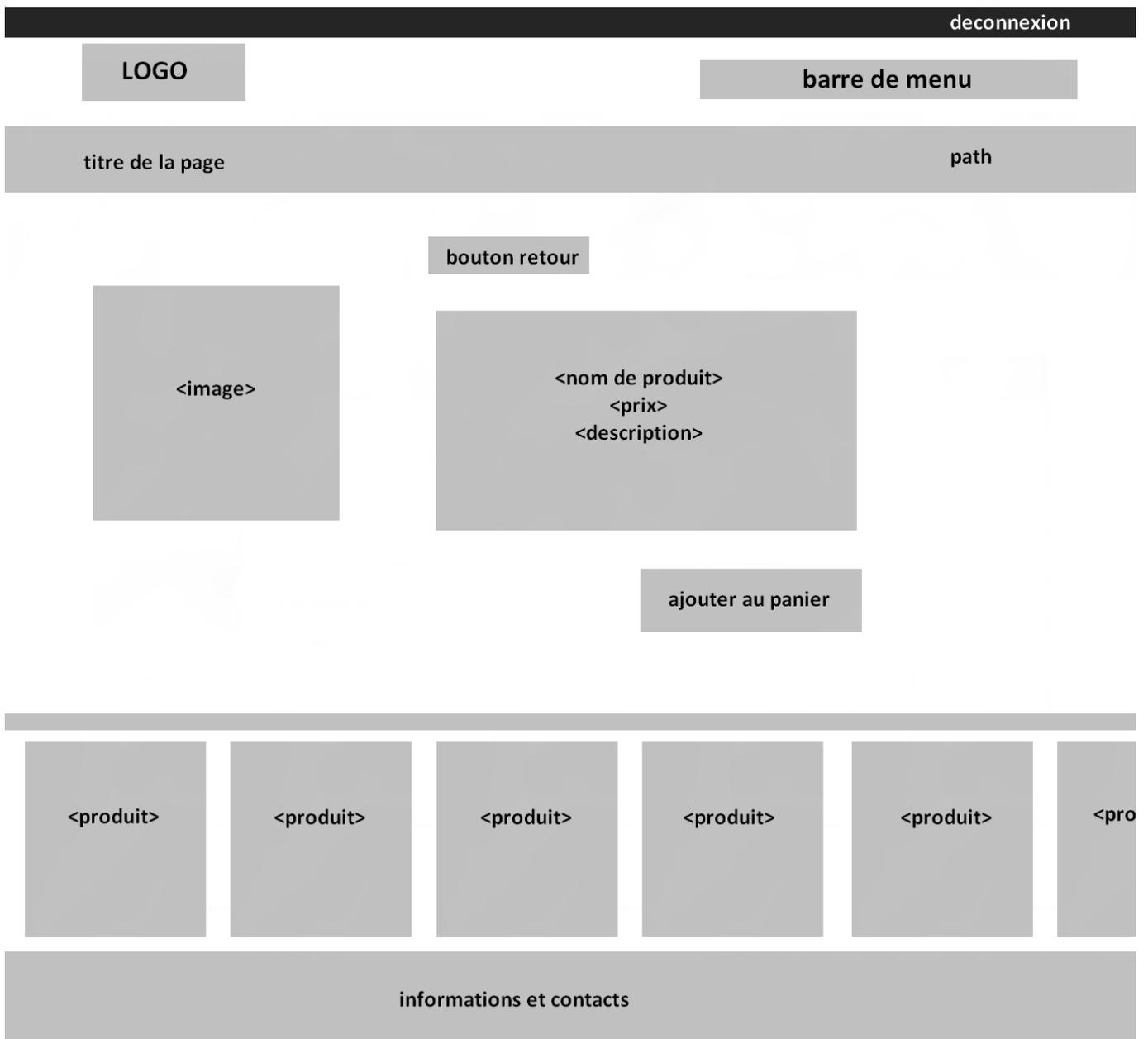


FIGURE 3.17 – La charte graphique de consultation d'un produit.

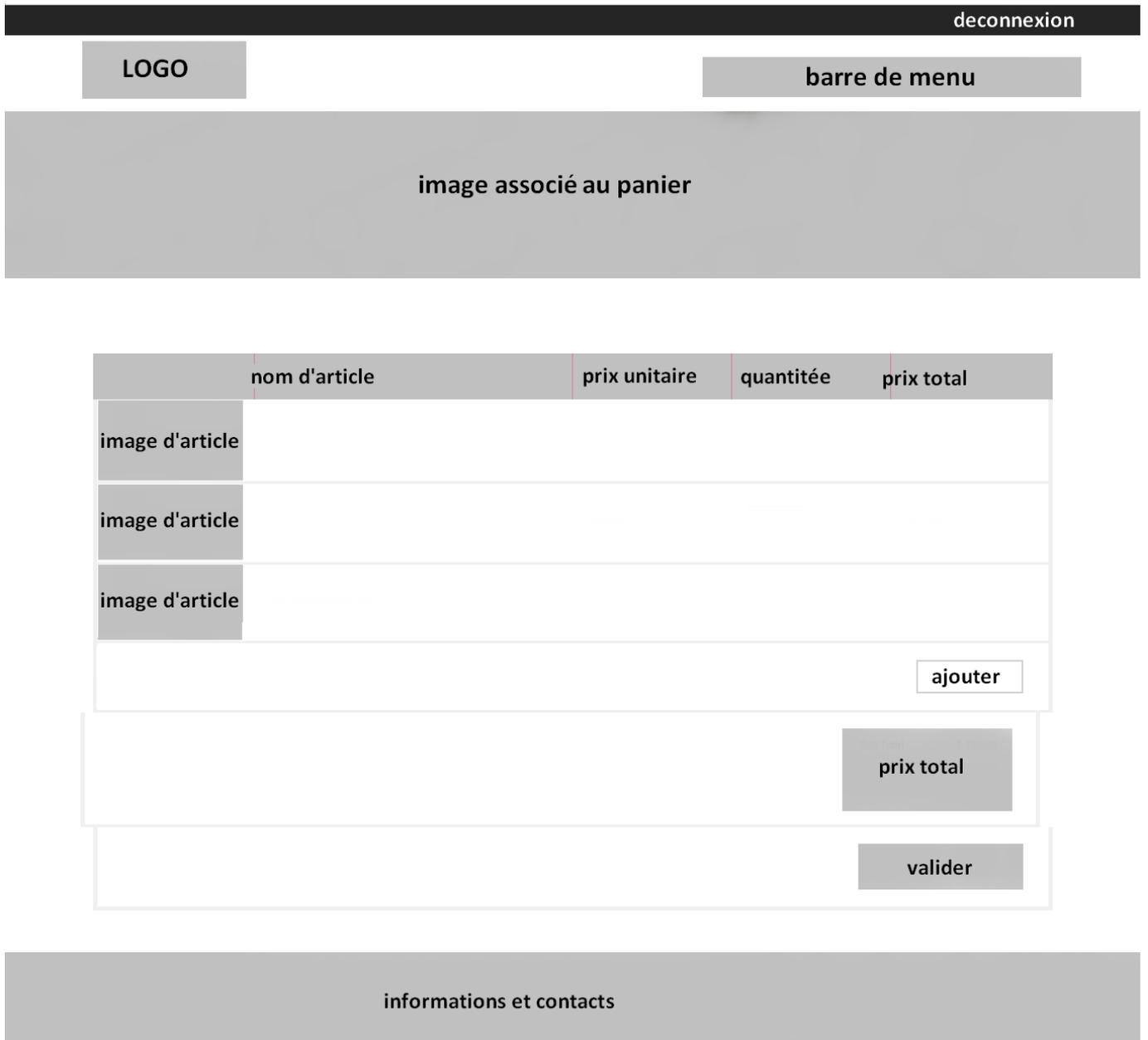


FIGURE 3.18 – La charte graphique de panier.

3.4 Conclusion

Dans ce chapitre, nous avons élaboré notre conception qui était très importante pour pouvoir visualiser le fonctionnement de notre application d'une façon abstraite en mettant en œuvre les différentes interactions entre les utilisateurs qui composent le système. Nous avons établi un diagramme de classe pour la création de la base de données. Le chapitre suivant sera consacré à la réalisation.

CHAPITRE 4

RÉALISATION

4.1 Introduction

Dans ce chapitre, nous allons passer à la dernière étape qui est la réalisation. Cette étape est cruciale pour la mise en place de ce que nous avons fait auparavant. Nous entamerons ce chapitre par la présentation des différents concepts, technologies et outils utilisés pour développer notre application ensuite nous mettrons en œuvre quelques interfaces réalisées.

Dans le chapitre précédent, nous avons proposé un modèle complet pour un système de vente en ligne mais, nous avons réalisé juste la partie la plus importante du projet, à savoir la conception générale et la conception détaillée.

4.2 Présentation des concepts informatiques utilisés

Le développement d'une application WEB demande l'inclusion de quelques concepts de développement web que nous allons citer dans ce qui suit :

4.2.1 Développement Front-end

il s'agit des éléments de l'application que l'on voit à l'écran et avec lesquels on peut interagir. Ces éléments sont composés de HTML, CSS et de Javascript contrôlés par le navigateur web de l'utilisateur.

Les champs de compétence du Front-End peuvent être séparés en deux :

1. Le **design**
2. Le **développement** HTML, CSS, Javascript

Le design est traditionnellement réalisé par un web designer qui produit des maquettes graphiques à l'aide de Photoshop ou Fireworks. Cependant de plus en plus de web designer ont franchi la barrière et savent coder en HTML et CSS. Dans certains cas ils sont aussi capables de

produire du Javascript.

Auparavant, lorsque l'on parlait de développeur, on sous-entendait développeur Back-End. Maintenant on s'aperçoit que certains web designers possèdent également des compétences en développement.

Le développeur Front-End est donc une personne qui peut être issue du milieu du web design en ayant renforcé ses connaissances en développement. Il peut aussi être un développeur qui a choisi de se spécialiser dans les technologies et langages du Front-End tels que :

- HTML
- CSS
- Javascript

4.2.2 Modélisation des données

Dans la conception d'un système d'information, la modélisation des données est l'analyse et la conception de l'information contenue dans le système.

Il s'agit essentiellement d'identifier les entités logiques et les dépendances logiques entre ces entités. La modélisation des données est une représentation abstraite, dans le sens où les valeurs des données individuelles observées sont ignorées au profit de la structure, des relations, des noms et des formats des données pertinentes, même si une liste de valeurs valides est souvent enregistrée. Le modèle de données ne doit pas seulement définir la structure de données, mais aussi ce que les données veulent vraiment signifier.

4.2.3 Mapping objet-relationnel

Un mapping objet-relationnel (en anglais object-relational mapping ou ORM) est une technique de programmation informatique qui crée l'illusion d'une base de données orientée objet à partir d'une base de données relationnelle en définissant des correspondances entre cette base de données et les objets du langage utilisé. On pourrait le désigner par « correspondance entre monde objet et monde relationnel ».

4.2.4 Gestion des sessions

L'implémentation de gestion de session définit le mécanisme d'échange qui sera utilisé entre l'utilisateur et l'application Web pour partager et échanger en continu l'ID de session. Il existe plusieurs mécanismes disponibles dans HTTP pour maintenir l'état de la session dans les applications Web, telles que les cookies (en-tête HTTP standard), les paramètres d'URL, les arguments d'URL sur les requêtes GET, les arguments corporels sur les requêtes POST, tels que le formulaire caché Les champs (formulaires HTML) ou les en-têtes HTTP propriétaires.

4.2.5 Transaction Business Logic

Transaction Business Logic est la logique requise dans le traitement des transactions de base de données pour appliquer les règles métier. Il est parfois caractérisé comme logique d'exécution,

car la transaction doit être rejetée si les règles ne sont pas adoptées.

4.2.6 La sécurité des applications WEB

La sécurité des applications Web, est une branche de la sécurité de l'information qui traite spécifiquement la sécurité des sites Web, des applications Web et des services Web. À un niveau élevé, la sécurité des applications Web s'appuie sur les principes de la sécurité des applications mais les applique spécifiquement aux systèmes Internet et Web.

4.3 les technologies utilisées au développement

Nous allons présenter les différentes technologies et langages utilisés dans le développement de notre application web

4.3.1 HTML

L'HTML est un langage informatique utilisé sur l'internet. Il est utilisé pour créer des pages web. L'acronyme signifie HyperText Markup Language, ce qui signifie en français "langage de balisage d'hypertexte". Cette signification porte bien son nom puisque effectivement ce langage permet de réaliser de l'hypertexte à base d'une structure de balisage [04].

4.3.2 CSS

Cascading Style Sheets c'est un langage informatique qui sert à décrire la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C). Introduit au milieu des années 1990, CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années 2000 [03].

4.3.3 JavaScript

JavaScript est un langage de programmation de scripts léger et interprété. Il est conçu pour créer des pages web interactives et des applications centrées sur le réseau. JavaScript est gratuit et très simple à mettre en œuvre car il est intégré au HTML. Il est ouvert et multiplate-forme.

4.3.4 Bootstrap

Bootstrap est une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub.

4.3.5 JAVA

Java est un langage de programmation orienté objet de haut niveau développé initialement par Sun Microsystems et publié en 1995. Java fonctionne sur une variété de plates-formes, telles que Windows, Mac OS et les différentes versions d'UNIX.

4.3.6 JSP

Le JavaServer Pages ou JSP est une technique basée sur Java qui permet aux développeurs de créer dynamiquement du code HTML, XML ou tout autre type de page web. Cette technique permet au code Java et à certaines actions prédéfinies d'être ajoutées dans un contenu statique. Depuis la version 2.0 des spécifications, la syntaxe JSP est complètement conforme au standard XML.

4.3.7 JSTL

La JavaServer Pages Standard Tag Library (JSTL) est une composante de la plate-forme JEE de développement. Elle étend la spécification JSP en ajoutant une bibliothèque de balises pour les tâches courantes, comme le travail sur des fichiers XML, l'exécution conditionnelle, les boucles et l'internationalisation. JSTL a été développée par le Java Community Process (JCP). Le 8 mai 2006, la JSTL 1.2 est sortie.

4.3.8 Java servlet

une servlet est une classe Java qui permet de créer dynamiquement des données au sein d'un serveur HTTP. Ces données sont le plus généralement présentées au format HTML, mais elles peuvent également l'être au format XML ou tout autre format destiné aux navigateurs web. Les servlets utilisent l'API Java Servlet.

4.3.9 JPA

La Java Persistence API (abrégée en JPA), est une interface de programmation Java permettant aux développeurs d'organiser des données relationnelles dans des applications utilisant la plateforme Java.

4.3.10 Java EE

Java Enterprise Edition, ou Java EE, est une spécification pour la programmation Java de Sun plus particulièrement destinée aux applications d'entreprise. celles-ci sont considérées dans une approche multi-niveaux. Dans ce but, toute implémentation de cette spécification contient un ensemble d'extensions au Framework Java standard (JSE, Java Standard Edition) afin de faciliter la création d'applications réparties.

4.3.11 EJB

Enterprise Java Beans (EJB) est une architecture de développement pour construire des applications de niveau d'entreprise hautement évolutives et robustes à déployer sur un serveur d'applications compatible J2EE tel que JBOSS, Web Logic, etc...

4.3.12 JDBC

JDBC (Java DataBase Connectivity) est une interface de programmation créée par Sun Microsystems depuis racheté par Oracle Corporation pour les programmes utilisant la plateforme Java. Elle permet aux applications Java d'accéder par le biais d'une interface commune à des sources de données pour lesquelles il existe des pilotes JDBC. Normalement, il s'agit d'une base de données relationnelle, et des pilotes JDBC sont disponibles pour tous les systèmes connus de bases de données relationnelles.

4.4 Outils de développement

La création et le développement d'une telle application nécessite l'utilisation de quelques outils que nous allons citer dans ce qui suit.

4.4.1 NetBeans IDE

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License) et GPLv2. En plus de Java, NetBeans permet la prise en charge native de divers langages tels le C, le C++, le JavaScript, le XML, le Groovy, le PHP et le HTML, ou d'autres (dont Python et Ruby) par l'ajout de greffons. Il offre toutes les facilités d'un IDE moderne (éditeur en couleurs, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

4.4.2 GlassFish

GlassFish est le nom du serveur d'applications Open Source Java EE 5 et désormais Java EE 7 avec la version 4.1 qui sert de socle au produit Oracle GlassFish Server (anciennement Sun Java System Application Server de Sun Microsystems). Sa partie Toplink persistence provient d'Oracle. C'est la réponse aux développeurs Java désireux d'accéder aux sources et de contribuer au développement des serveurs d'applications de nouvelle génération.

4.4.3 MySQL

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server.

4.4.4 MySQL Workbench

MySQL Workbench est un outil visuel unifié pour les architectes de base de données, les développeurs et les DBA. MySQL Workbench fournit la modélisation des données, le développement SQL et des outils d'administration complets pour la configuration du serveur, l'administration des utilisateurs, des sauvegardes et bien plus encore. MySQL Workbench est disponible sous Windows, Linux et Mac OS X.

4.4.5 Adobe PhotoShop CC

il s'agit d'un logiciel de retouche image de très grande base des filtres et des effets comme il serre dans ses dernières versions d'établir des animations, donc il nous a aidés dans la construction de la bannière publicitaire.

4.5 Le MVC de notre application

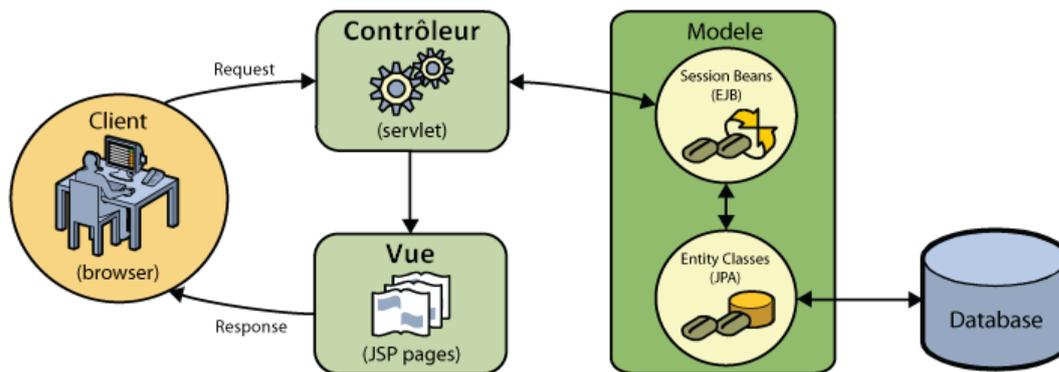


FIGURE 4.1 – Le MVC de notre application. [15]

Comme la figure [Figure 4.1] le montre, le MVC utilisé dans notre application est en base JAVA, le contrôleur c'est la java servlet, la vue c'est les pages JSP et pour le modèle nous avons le EJB comme session beans et le JPA comme les classes des entités .

4.6 Les interfaces développées dans notre application

4.6.1 Interface d'accueil

C'est la page d'accueil qui s'affiche dès l'accès à notre application web.

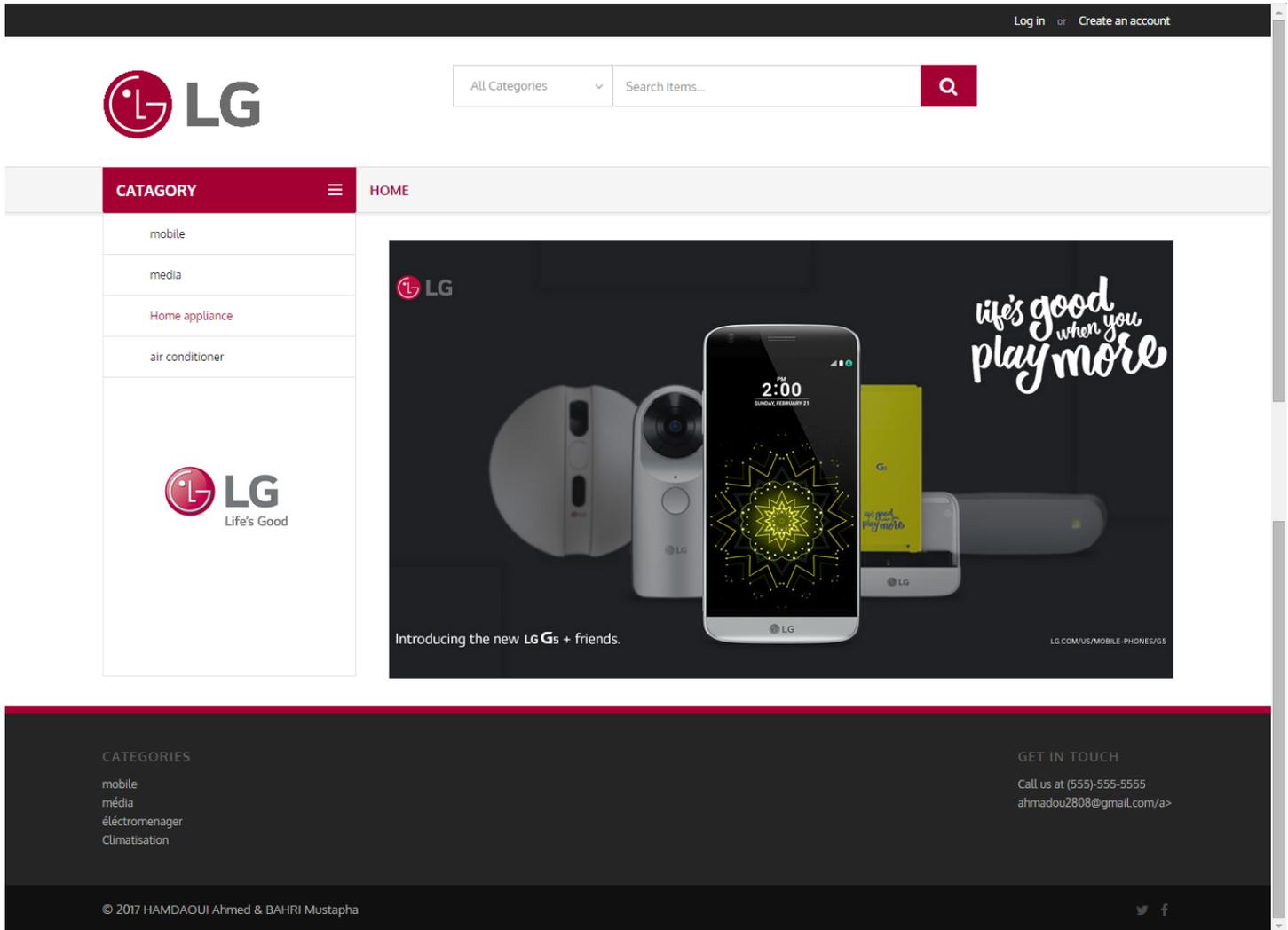
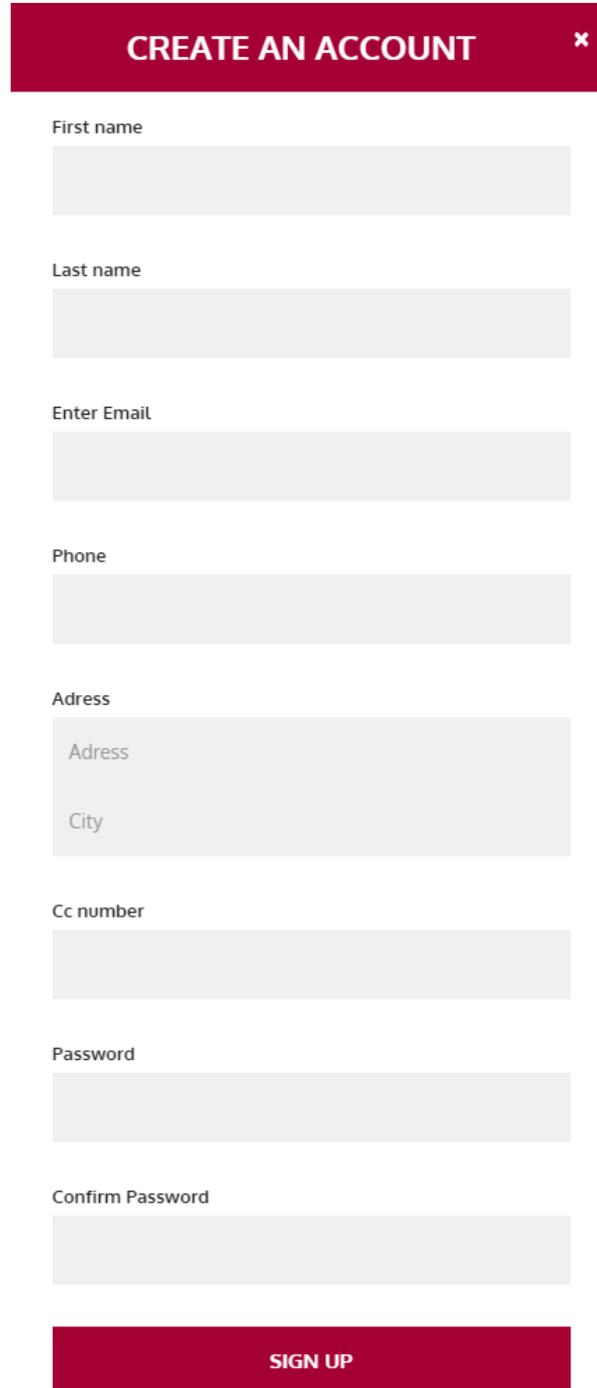


FIGURE 4.2 – Interface d'accueil.

4.6.2 Interface d'inscription

C'est l'interface où un visiteur va s'enregistrer afin d'être membre pour qu'il puisse faire des commandes.



CREATE AN ACCOUNT ✕

First name

Last name

Enter Email

Phone

Adress

Cc number

Password

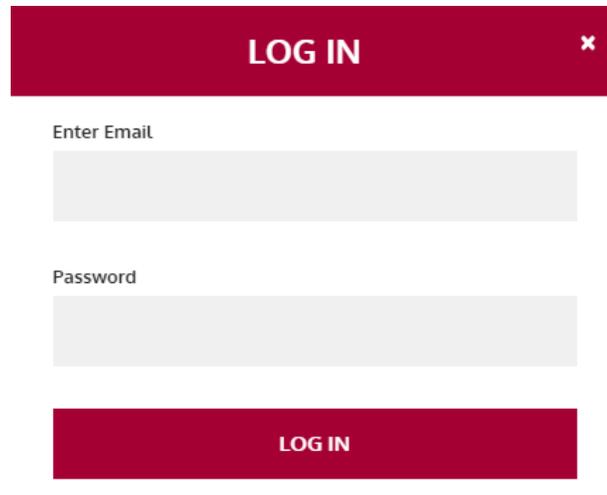
Confirm Password

SIGN UP

FIGURE 4.3 – Interface d'inscription.

4.6.3 Interface d'authentification

C'est l'interface où l'utilisateur peut accéder à son compte en remplissant son login et son mot de passe.



The image shows a login interface. At the top is a dark red header bar with the text "LOG IN" in white, uppercase letters, and a small white "x" icon in the top right corner. Below the header, there are two input fields. The first is labeled "Enter Email" and the second is labeled "Password". Both input fields are currently empty and have a light gray background. Below the input fields is a dark red button with the text "LOG IN" in white, uppercase letters. The entire interface is centered on a white background.

FIGURE 4.4 – Interface d'authentification.

4.6.4 Interface du catalogue

Cette interface affiche un ensemble de produits par la catégorie sélectionnée, chaque produit est affiché par une image correspondante, le nom de produit et son prix.

The screenshot displays the LG mobile product catalog interface. At the top, there is a navigation bar with "Log in" or "Create an account" and a search icon. Below this is the LG logo and navigation links for "HOME" and "CATEGORIES". The main content area is titled "PRODUCT GRID VIEW" and includes a breadcrumb "HOME / MOBILE". A left sidebar menu lists categories: "mobile" (selected), "media", "Home appliance", and "air conditioner". The main grid shows four smartphone products:

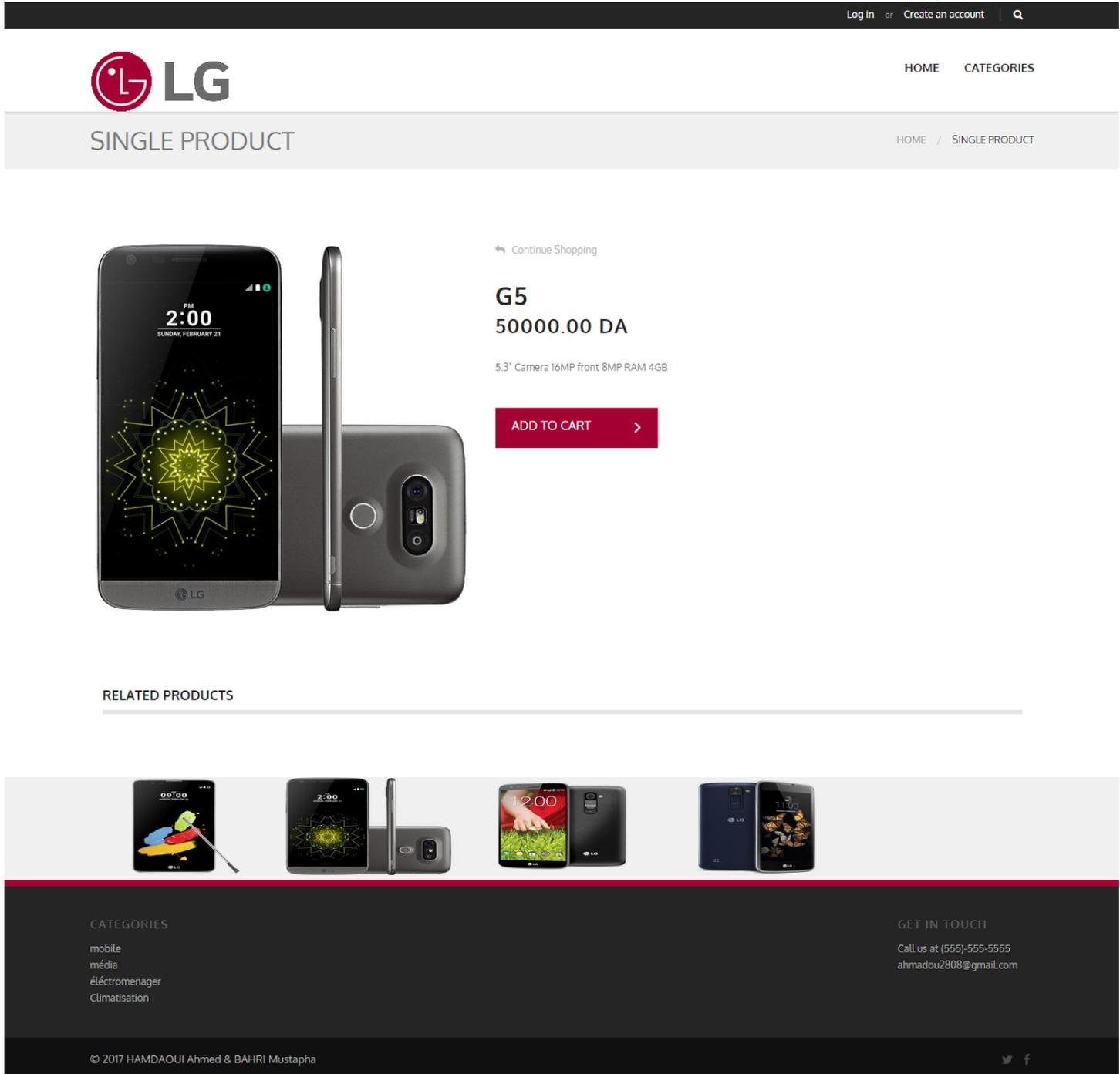
- STYLUS 2**: 24000.00 DA
- G5**: 50000.00 DA
- LG VS980W**: 30000.00 DA
- K8**: 26000.00 DA

The footer contains a "CATEGORIES" list (mobile, média, électroménager, Climatisation), a "GET IN TOUCH" section with contact information (Call us at (555)-555-5555, ahmadou2808@gmail.com/a>), and a copyright notice: © 2017 HAMDAOUI Ahmed & BAHRI Mustapha.

FIGURE 4.5 – Interface du catalogue.

4.6.5 Interface du fiche technique d'un produit

C'est l'interface qui s'affiche lors de sélection d'un produit. C'est là où on peut trouver des informations détaillées sur ce dernier.



Log in or Create an account

HOME CATEGORIES

SINGLE PRODUCT HOME / SINGLE PRODUCT

Continue Shopping

G5
50000.00 DA

5.3" Camera 16MP front 8MP RAM 4GB

ADD TO CART >

RELATED PRODUCTS

CATEGORIES

- mobile
- média
- électromenager
- Climatisation

GET IN TOUCH

Call us at (555)-555-5555
ahmadou2808@gmail.com

© 2017 HAMDAOUI Ahmed & BAHRI Mustapha

FIGURE 4.6 – Interface du fiche technique d'un produit.

4.6.6 Interface du panier

L'interface panier affiche l'ensemble de produits sélectionnés pour l'achat, dont la quantité de produits, le prix unitaire, le prix total et un bouton pour valider.

Log out | 76000.0 DA

HOME CATEGORIES MY ACCOUNT

CART

HOME / CART

	Product Name	Price	Quantity	Sub Total
X	 G5	50000.00	<input type="text" value="2"/>	100000.0
X	 K8	26000.00	<input type="text" value="1"/>	26000.0

BACK TO SHOPPING UPDATE CART

Sub Total 76000.0 DA
Grand Total 76000.0 DA

CHECKOUT

CATEGORIES

- mobile
- média
- électroménager
- Climatisation

GET IN TOUCH

Call us at (555)-555-5555
ahmadou2808@gmail.com/a>

© 2017 HAMDAOUI Ahmed & BAHRI Mustapha

FIGURE 4.7 – Interface du panier.

4.6.7 Interface d'ajout d'un produit

cette interface permet à l'administrateur d'ajouter un produit dans la base de données après le remplissage de formulaire d'ajout.

The screenshot displays the 'ADD PRODUCT' interface. At the top, there is a dark navigation bar with 'Log out' and a search icon. Below this is a white navigation bar with the LG logo and menu items: HOME, ADD_PRODUCT, CATEGORIES, CUSTOMERS, and ORDERS. A breadcrumb trail below the navigation bar reads 'HOME / ADD PRODUCT'. The main content area is titled 'ADD PRODUCT' and contains a form with the following elements:

- Product Name:** A text input field.
- Price:** A text input field.
- Category:** A dropdown menu with 'Mobile' selected.
- Shop:** A dropdown menu with 'LG Show Room' selected.
- Description:** A large text area for product details.
- images:** A section with a 'Choose File' button and the text 'No file chosen'.

At the bottom of the form, there are two buttons: 'BACK' and 'ADD PRODUCT'. The footer of the page is dark and contains the following information:

- CATEGORIES:** mobile, média, électroménager, Climatisation.
- GET IN TOUCH:** Call us at (555)-555-5555, ahmadou2808@gmail.com/a>
- Copyright:** © 2017 HAMDAOUI Ahmed & BAHRI Mustapha.

FIGURE 4.8 – Interface d'ajout d'un produit.

4.7 Conclusion

Dans ce chapitre, nous avons décrit brièvement le processus de réalisation de notre application en spécifiant l'environnement de développement, l'implémentation de la base des données et la démarche suivie pour la réalisation. En effet, nous avons achevé l'implémentation et les tests de

tous les cas d'utilisation, tout en respectant la conception élaborée. En d'autres termes, nous détenons la version finale du logiciel, installée dans notre environnement de développement.

CONCLUSION GÉNÉRALE

Le présent rapport est le résultat de notre stage que nous avons effectué dans le cadre de la réalisation de notre projet de fin d'études pour l'obtention de diplôme master professionnel en informatique option génie logiciel.

Lors de ce stage de deux mois, nous avons pu mettre en pratique nos connaissances théoriques acquises durant notre formation à l'université, de plus, nous sommes arrivés à réaliser les objectifs que nous avons mis au début de cette période bien que nous ayons vécus des difficultés, les méthodologies que nous avons utilisées pour les dépasser sont des signes de satisfaction.

Nous avons, tout d'abord, entamé notre étude par la capture des besoins qui est une étape cruciale et nécessaire pour mieux assimiler le système déjà existant, puis par la définition des principaux intervenants et l'identification des besoins. Ensuite, nous avons procédé à l'analyse qui a permis la conception d'une architecture de base stable, qui est le point de départ à la conception dans laquelle nous avons utilisé UML comme langage de modélisation. Enfin, la réalisation, qui nous a permis de développer notre application en tenant compte de l'architecture matérielle et de l'environnement logiciel.

Nous avons toutefois réalisé une application web composée de plusieurs modules contenant les fonctionnalités nécessaires pour le bon déroulement des différentes gestions côté client (panier, commande, profils, paiement, livraison...) et côté administrateur (clients, produits, commandes, ...).

Perspectives

De ce qui précède, il est difficile de prétendre avoir eu une solution idéale, toutefois nous espérons avoir répondu un tant soi peu à notre problématique et confirmé nos hypothèses et dans ce qui suit nous allons essayer de lister des perspectives de notre système. Autrement dit, nous allons présenter les améliorations qui peuvent y être apportées.

- Intégrer le paiement en ligne par les master card ou les cartes VISA.
- La cartographie des clients selon une zone géographique donnée c'est-à-dire du GeoMarketing avec l'intégration de Google maps ; Ce qui nous permettra de mieux cibler nos clients.
- L'amélioration de la sécurité du paiement en adoptant 3DSecure qui est un système permettant de s'assurer, lors de chaque paiement en ligne, que la carte est bien utilisée par son titulaire.
- L'amélioration de la gestion de stock de la boutique en ajoutant d'autres fonctionnalités pour offrir une meilleure gestion à l'administrateur de l'application.

BIBLIOGRAPHIE

- [01] : Cycle de vie d'un logiciel, Réalisé sous la direction de Jean-François PILLOU fondateur de CommentCaMarche.net, Septembre 2015
- [02] : UML 2 De l'apprentissage à la pratique, Laurent AUDIBERT, 2013
- [03] : Conception et développement d'un site web de e commerce pour le compte de LSAT Nokia, Maroua Chaabani Lamia Bouafif, 2013
- [04] : Définition de HTML, infowebmaster.fr, informations et ressources pour webmasters, 2007
- [05] : UML 2 par la pratique, Pascal Roque, 2009
- [06] : Apprenez à programmer en Java, cysboy, 2013
- [07] : UML 2 Analyse et conception - Mise en œuvre guidée avec études de cas Joseph Gabay, David Gabay, 2008.
- [08] : cours HTML,CSS,JAVAScript de w3schools, 2017
- [09] : site des tutoriels pour langage de programmation tutorialsponit.com, 2017
- [10] : Forum pour les fameux problems rencontrés en programmation
- [11] : Décrire sa base de données à l'aide d'un modèle conceptuel de données, Article (centre d'écologie fonctionnelle et Evolutive-France), Guillaume HARRY, Marie-Claude QUIDOZ, 2005.
- [12] : Méthodologie des systèmes d'info: UML, Frédéric di Gallo, 2000/2001
- [13] : A Spiral Model of Software Development and Enhancement. Barry Boehm. 1988
- [14] : Advanced ActionScript with Design Patterns, Joey Lott et Danny Patterson, Adobe Press, 2007
- [15] : Le tutorial de e commerce de netbeans, 2017

Résumé

L'intérêt de ce travail est de créer une boutique virtuelle pour le distributeur officiel de la marque LG « LG ShowRoom », ce projet se traduit par la mise en œuvre d'une application web qui gère le fonctionnement de vente en ligne de ce dernier. L'utilité de cette application web est de faciliter la commercialisation des produits et d'établir les commandes des achats. La modélisation de cette application web a été faite avec UML en s'appuyant sur le modèle en V comme cycle de vie. Notre application a été basée sur le langage de programmation java où nous avons utilisé l'architecture de pattern MVC pour organiser le code source.

Mots clés

SGBD, UML, BDD, MySQL, SQL, WEB, JEE, JSP, IDE, JAVA, JavaScript, HTML, MVC, Bootstrap, GlassFish.

Abstract

In the context of presenting a final project study, efforts have been devoted to the implementation of a web application that works as a virtual shop for the LG's official distributor, to achieve this goal for the modeling this web application was made by following the UML following the V life cycle. The implementation of the application was based on java, using the MVC pattern to organise the source code.

Keywords

SGBD, UML, BDD, MySQL, SQL, WEB, JEE, JSP, IDE, JAVA, JavaScript, HTML, MVC, Bootstrap, GlassFish.