

Table des matières

Introduction générale	1
Chapitre I Présentation générale	2
Introduction	3
I. Présentation de l'organisme d'accueil	3
II. Présentation du sujet	3
III. Méthodologie et formalisme adoptée.....	4
Conclusion	7
Chapitre II Etude Technique	8
Introduction	9
I. Le business électronique	9
II. Environnement technique de développement	16
III. Sécurisation des applications web	21
IV. Application mobile.....	24
Conclusion	29
Chapitre III Etude préliminaire	30
Introduction	31
I. Présentation du projet à réaliser	31
II. Identification des acteurs.....	31
III. Recueil des besoins fonctionnels.....	31
IV. Recueil des besoins non fonctionnels	33
V. Modélisation du contexte.....	33
Conclusion	36
Chapitre IV Capture des besoins fonctionnels et techniques	37
Introduction	38
I. Capture des besoins fonctionnels	38
II. Description des cas d'utilisation	40
III. Structuration des cas d'utilisations dans des packages.....	56
IV. Identification des classes candidates	56
Conclusion	72
Chapitre V Analyse et conception	73

Introduction	74
Conclusion	94
Chapitre VI Réalisation.....	95
Introduction	96
Test et validation	119

Liste des figures

Figure 1 Le processus de développement en Y	6
Figure 2 Compte Client du site promouv.com	11
Figure 3 Paiement en ligne du site promouv.com	12
Figure 4 Erreur compte client du site promouv.com	12
Figure 5 Discussion instantanée du site promouv.com	13
Figure 6 Compte client du site affariyet.....	13
Figure 7 Paiement en ligne du site affariyet	14
Figure 8 Recherche du site tinydeal.....	15
Figure 9 Chat du site tinydeal.....	15
Figure 10 Structure du Framework Codeigniter	19
Figure 11 Structure du Framework Codeigniter 2	20
Figure 12 Fonctionnement du Framework Codeigniter.....	21
Figure 13 Les API de PhoneGap.....	27
Figure 14 Plateformes de développement.....	28
Figure 15 PhoneGap Build	28
Figure 16 Diagramme de contexte.....	36
Figure 17 Diagramme de Cas d'utilisation général pour l'application web	40
Figure 18 Diagramme de cas d'utilisation « Gérer profil »	42
Figure 19 Diagramme de cas d'utilisation « Gérer annonces »	43
Figure 20 Diagramme de cas d'utilisation « Gérer produit ».....	44
Figure 21 Diagramme de cas d'utilisation « Gérer messages »	45
Figure 22 Diagramme de cas d'utilisation « Gérer espace forum».....	46
Figure 23 Diagramme de cas d'utilisation « Consulter l'état du commerce».....	47
Figure 24 Diagramme de cas d'utilisation « Consulter les notifications».....	48
Figure 25 Diagramme de cas d'utilisation « Gérer les achats».....	49
Figure 26 Diagramme de cas d'utilisation « Gérer utilisateurs».....	50
Figure 27 Diagramme de cas d'utilisation « Superviser plateforme».....	51
Figure 28 Diagramme de cas d'utilisation « Gérer paiement»	52
Figure 29 Diagramme de cas d'utilisation « Effectuer paiement».....	53
Figure 30 Diagramme de cas d'utilisation « Gérer notifications»	54

Figure 31 Diagramme cas d'utilisation pour l'application mobile	55
Figure 32 Classe candidate « gestion inscription »	57
Figure 33 Classe candidate « gestion administration»	57
Figure 34 Classe candidate « gestion du commerce ».....	57
Figure 35 Classe candidate « gestion de l'interaction »	58
Figure 36 Classe candidate « gestion des achats »	58
Figure 37 Diagramme des classes participantes « S'inscrire, S'authentifier et Gérer profil» .	59
Figure 38 Diagramme des classes participantes « Gérer annonces»	59
Figure 39 Diagramme des classes participantes « Gérer produit»	60
Figure 40 Diagramme des classes participantes « Consulter l'état du commerce»	60
Figure 41 Diagramme des classes participantes « Consulter messages»	60
Figure 42 Diagramme des classes participantes « Gérer utilisateur et superviser plateforme»	61
Figure 43 Diagramme des classes participantes « Gérer espace forum».....	61
Figure 44 Diagramme des classes participantes « Gérer notification et consulter notification»	61
Figure 45 Diagramme des classes participantes « gérer paiement et effectuer paiement»...	62
Figure 46 Exemple d'héritage de Twig	63
Figure 47 Comparaison entre code PHP et twig	64
Figure 48 Grille de colonne de Twitter bootstrap.....	64
Figure 49 Responsive web design de Twitter bootstrap.....	65
Figure 50 Syntaxe de JQuery mobile.....	68
Figure 51 Configuration du path PHP de Codeigniter.....	69
Figure 52 Path PHP dans les contrôleurs Codeigniter.....	69
Figure 53 Routage de Codeigniter 1/2	69
Figure 54 Routage de Codeigniter 2/2	69
Figure 55 Fichier .htaccess de Codeigniter.....	70
Figure 56 Découpage en catégories.....	74
Figure 57 Diagramme de classe « Interaction».....	75
Figure 58 Diagramme de classe « Vente»	75
Figure 59 Diagramme de classe « Activité Membre»	76
Figure 60 Diagramme de classe	77

Figure 61 Schématisation du diagramme de séquence	80
Figure 62 Diagramme de séquence du cas d'utilisation « Gérer profil»	81
Figure 63 Diagramme de séquence du scénario « Ajs_N1 : Ajouter sujet».....	82
Figure 64 Diagramme de séquence du scénario « Ajs_N2 : Ecrire commentaire».....	83
Figure 65 Diagramme de séquence du scénario « Ach_N1 : Acheter produit».....	84
Figure 66 Diagramme d'état-transition « Produit».....	85
Figure 67 Diagramme d'état-transition « Utilisateur».....	86
Figure 68 Architecture 3tiers	87
Figure 69 Architecture MVC.....	87
Figure 70 Schéma de la base des données.....	88
Figure 71 Navigation Client « gérer achat 1/3»	90
Figure 72 Navigation Client « gérer achat 2/3»	91
Figure 73 Navigation Client « gérer achat 3/3»	91
Figure 74 Navigation Commerçant 1/2	93
Figure 75 Navigation Commerçant 2/2	93
Figure 76 Navigation Application mobile	94
Figure 77 Architecture de la plateforme e-commerce	96
Figure 78 Configuration environnement.....	98
Figure 79 Configuration emplacement application	98
Figure 80 Virtual host	98
Figure 81 Fichier .htaccess de Codeigniter.....	99
Figure 82 Fichier « database.php » de Codeigniter	99
Figure 83 Fichier « autoload.php » de Codeigniter.....	100
Figure 84 Fichier de configuration twig.php	100
Figure 85 Extrait de la librairie twig	101
Figure 86 Autoload de la librairie twig	101
Figure 87 Appellation des vue depuis le contrôleur avec twig	101
Figure 88 Block Login twig.....	102
Figure 89 Héritage de Template layout.twig.....	102
Figure 90 Crédit d'un nouveau projet PhoneGap	103
Figure 91 Arborescence du projet mobile.....	104
Figure 92 Emulateur Android	104

Figure 93 Page d'accueil	105
Figure 94 Interface « S'inscrire».....	106
Figure 96 Interface « formulaire d'inscription»	106
Figure 96 Interface « succès d'inscription».....	107
Figure 97 Interface « Mail de confirmation»	107
Figure 98 Interface «connexion FB»	108
Figure 99 Interface «gestion profil».....	108
Figure 100 Interface commerçant.....	109
Figure 101 Interface « gestion produit ».....	110
Figure 102 Interface « ajouter produit ».....	111
Figure 103 Interface « modifier produit ».....	112
Figure 104 Interface « Détail produit Commerçant »	112
Figure 105 Interface « Recherche»	113
Figure 106 Interface « Détails produit»	114
Figure 107 Interface « note produit».....	114
Figure 108 Interface « erreur fecebook»	115
Figure 109 Interface « partage facebook »	115
Figure 110 Interface « Ajouter au panier».....	115
Figure 111 Interface « Détails panier»	116
Figure 112 Interface « Espace forum»	117
Figure 113 Interface « Ajouter sujet»	117
Figure 114 Interface « Ajouter commentaire»	118
Figure 115 Interface « Notification»	118

Liste des Tableaux

Tableau 1 Tableau comparatif des sites.....	14
Tableau 2 Tableau comparatif des Framework.....	18
Tableau 3 Identification des messages de l'acteur commerçant.....	33
Tableau 4 Identification des messages de l'acteur Client	34
Tableau 5 Identification des messages de l'acteur Administrateur.....	34
Tableau 6 Rôle de chaque utilisateur finale	35
Tableau 7 Identification des cas d'utilisation.....	39
Tableau 8 Description du cas d'utilisation « S'inscrire ».....	41
Tableau 9 Description du cas d'utilisation « Gérer profil ».....	41
Tableau 10 Description du cas d'utilisation « Gérer les annonces »	42
Tableau 11 Description du cas d'utilisation « Gérer produit »	43
Tableau 12 Description du cas d'utilisation « Gérer messages ».....	44
Tableau 13 Description du cas d'utilisation « Gérer espace forum »	46
Tableau 14 Description du cas d'utilisation « Consulter l'état du commerce »	47
Tableau 15 Description du cas d'utilisation « Consulter les notifications»	48
Tableau 16 Description du cas d'utilisation « Gérer les achats»	49
Tableau 17 Description du cas d'utilisation « Gérer utilisateurs»	50
Tableau 18 Description du cas d'utilisation « Superviser plateforme»	51
Tableau 19 Description du cas d'utilisation « Gérer paiement».....	52
Tableau 20 Description du cas d'utilisation « Effectuer paiement»	53
Tableau 21 Description du cas d'utilisation « Gérer notifications»	54
Tableau 22 Organisation des cas d'utilisation en package	56
Tableau 23 Dictionnaire de données	79
Tableau 25 Test Gestion de la base de données.....	119
Tableau 26 Test Inscription et authentification	121
Tableau 27 Test Gestion produit	123
Tableau 28 Test Gestion Profil	124
Tableau 29 Test Noter produit	125
Tableau 25 Test Envoi message	125
Tableau 25 Test Forum.....	126

Introduction générale



Chapitre I Présentation générale



Introduction

Dans ce chapitre, nous mettrons notre travail dans son contexte et nous fixerons l'objectif de notre travail. Nous commencerons par présenter l'organisme d'accueil et terminons par énoncer la méthodologie et formalise adoptée.

I. Présentation de l'organisme d'accueil

MTS est une société de services en ingénierie informatique (SSII), elle assure le développement offshore des sites internet. Elle est spécialisée dans la conception et la mise en œuvre des sites et des applications web reposant sur les nouvelles technologies Web2.0 et Web3.0. Elle offre le conseil, le développement & l'intégration ainsi que la formation.

→ Cette société offre différents services aux bien de ses utilisateurs tel que la Création sites web, sites e-commerce, le référencement, le design, le web marketing ainsi que les applications mobiles.

II. Présentation du sujet

Le présent projet appartient au domaine du business électronique précisément au commerce électronique. L'idée consiste à la conception et au développement d'une plateforme e-commerce. Cette dernière permettra de regrouper les commerçants en leurs fournissant un espace virtuel d'exposition et de vente dynamique et instantané de leurs produits. C'est semblable aux grandes surfaces dans le monde réel. De plus la plateforme cherche à offrir un environnement centralisé semblable aux réseaux sociaux, mais orienté commerce, dont le but d'avoir un environnement d'interaction entre différents clients et commerçants.

C'est vrai que sur le marché tunisien et international existe déjà plusieurs sites marchands, cependant, dans notre projet nous ne pensons pas réinventer la roue, mais au contraire, nous chercherons s'inspirer de leurs solution, en vue de développer une solution meilleure en terme de satisfaction des utilisateurs et innovante en terme de fonctionnalités.

Parmi les principaux objectifs de la plateforme sera d'éviter une certaine ambiguïté autour des propriétaires des différents produits à vendre. Dans notre cas on présentera pour chaque produit exposé les informations sur le commerçant propriétaire.

Un autre objectif sera d'appliquer le principe du conseil avant vente. Ce concept s'étend vers la description d'un produit, la présentation des différents avis d'autres internautes concernant le produit et l'exposition de nouveaux produits ajoutés et ceux appréciées par les internautes.

Notre plateforme proposera un module de paiement propriétaire en ligne pour tout commerçant, une pratique qui commence peu à peu à faire partie des habitudes de l'internaute tunisien. Ce dernier aura le droit d'exposer ses produits pour vente ou seulement pour consultation pour faire de la publicité. Il a aura aussi la possibilité d'activer le paiement en ligne qui se fera à travers la plateforme.

Un espace forum sera présent dans notre application dans le but de fournir un environnement d'interaction pour les utilisateurs de la plateforme, de cette manière les clients pourront interagir avec les vendeurs des produits, avoir plus d'informations et poser des questions sur les fonctionnalités de certains produits directement à leurs vendeurs.

Le commerçant aura rôle central au niveau de cette plateforme, il disposera d'un ensemble d'interfaces pour gérer ses produits ainsi que les statistiques faites sur ces derniers et pour pouvoir interagir avec sa clientèle à travers la plateforme sans le recours à d'autres moyens externes comme la messagerie par exemple.

III. Méthodologie et formalisme adoptée

Pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet nous utiliserons UML, qui offre un standard de modélisation, pour représenter l'architecture logicielle. UML (Unified Modeling Language) est un langage de modélisation graphique.

UML définit principalement neuf types de diagramme pour représenter les différents points de vue de modélisation, UML 2.0 comporte ainsi treize types de diagrammes représentant autant de vues distinctes pour représenter des concepts particuliers du

système d'information. Une identification des différents diagrammes sera mise au niveau de l'annexe.

Devant le nombre de méthodologies disponibles, le choix parmi elles, devient difficile, beaucoup de questions peuvent se poser lors d'un démarrage d'un projet :

- Comment on va organiser le travail entre nous
- Quelles tâches attribuer à qui ;
- Quel temps faudrait-il pour terminer le projet ;
- Comment éviter des dérives et de mauvaises estimations qui vont allonger le temps de développement.
- Comment on va procéder pour que le produit soit évolutif et facilement maintenable.

Durant ce projet nous nous sommes inspirés de la méthodologie 2TUP. 2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs qui vont interagir avec le système à construire, les messages qu'échangent les acteurs et le système, à produire le cahier des charges et à modéliser le contexte (le système est une boîte noire, les acteurs l'entourent et lui sont reliés , sur l'axe qui lie un acteur au système on met les messages que les deux s'échangent avec le sens). Le processus s'articule ensuite autour de trois phases essentielles :

- Une branche technique
- Une branche fonctionnelle
- Une phase de réalisation

La branche fonctionnelle capitalise la connaissance du métier de l'entreprise. Cette branche capture des besoins fonctionnels, ce qui produit un modèle focalisé sur le métier des utilisateurs finaux.

La branche technique capitalise un savoir-faire technique et/ou des contraintes techniques. Les techniques développées pour le système le sont indépendamment des fonctions à réaliser.

La phase de réalisation consiste à réunir les deux branches, permettant de mener une conception applicative et enfin la livraison d'une solution adaptée aux besoins.

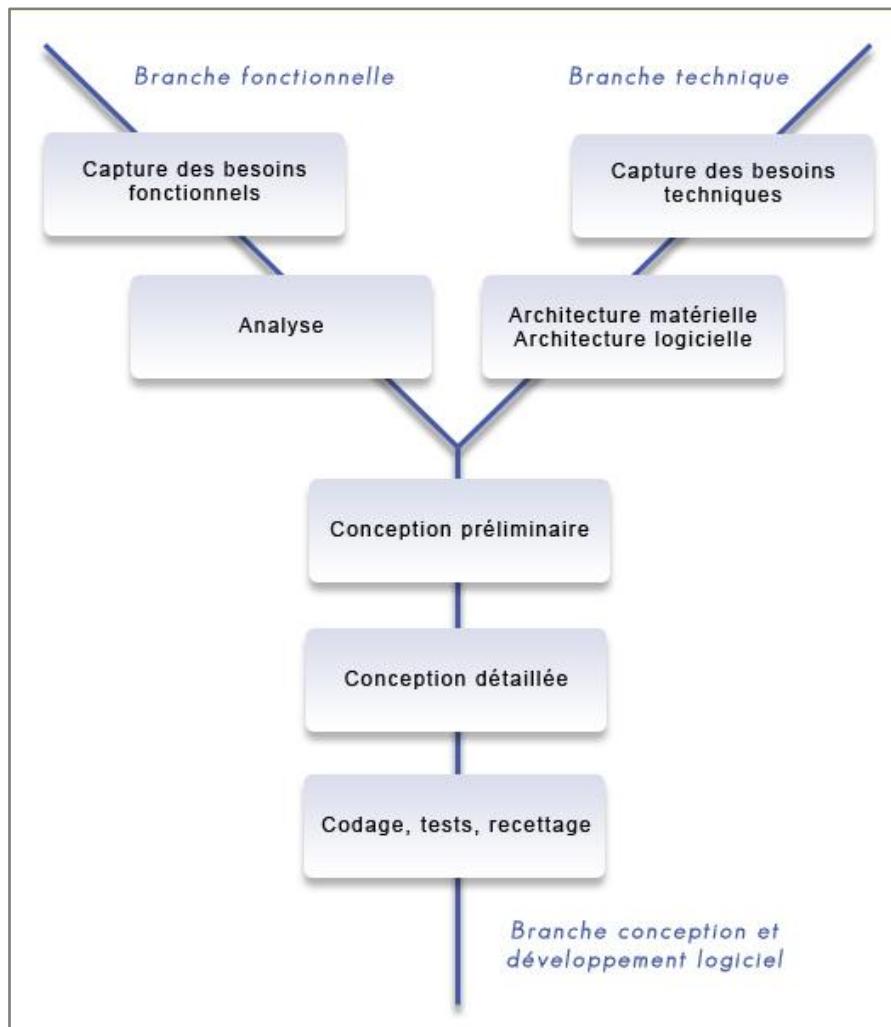


Figure 1 Le processus de développement en Y

Notre choix s'est porté sur la méthodologie 2TUP pour des raisons multiples : D'une part, 2TUP donne une grande importance à la technologie ce qui est important pour notre projet. D'autre part, 2TUP est un processus en Y qui contient une branche technique et autre fonctionnelle. Ces deux branches peuvent être exploitées en parallèle. De ce fait, si la

technologie évolue ou lors de déroulement du projet il y a eu apparence d'un besoin technique, la branche technique peut être traitée puis réintégrée dans le projet facilement.

Du fait que notre projet comporte un nombre important de fonctionnalité et détails qui n'étaient pas défini dès le début, 2TUP nous offre la possibilité de changer seulement la branche fonctionnelle sans toucher à l'autre branche.

Conclusion

A ce niveau nous avons pu élaborer un schéma général du contexte du projet ainsi que les objectifs majeurs à prendre en compte. Une présentation détaillée des technologies utilisées est dès lors nécessaire pour avoir une meilleure compréhension des exigences et contraintes du projet.



Chapitre II Etude Technique



Introduction

Ce chapitre a pour but de placer notre projet dans son contexte qui n'est autre que le « business électronique » ou encore « E-business », dans le but d'expliquer la terminologie requise ainsi que les différentes technologies que nous utiliserons pour développer notre solution. Nous ferons aussi un tour d'horizon de plusieurs solutions Web qui offrent des fonctionnalités similaires à notre travail. Nous présenterons en détail l'aspect sécurisation des applications Web vu l'importance de cet aspect dans les sites marchands et nous terminerons le rôle crucial du « mobile computing » dans le développement des applications Web.

I. Le business électronique

Traduit en « affaires électroniques » en français, c'est un mode de fonctionnement qui consiste à transformer les principaux processus commerciaux d'une entreprise, en y intégrant les technologies de communication et de l'information basée sur internet.

De nos jours, les entreprises utilisent Internet pour faire des affaires, c'est-à-dire pour communiquer avec leurs partenaires, accéder à leurs systèmes informatiques centrales et effectuer des transactions commerciales. Le domaine d'application des affaires électronique est plus large que le commerce électronique, il couvre également les achats électroniques (e-procurement en anglais), la gestion de la chaîne d'approvisionnement avec le traitement électronique des ordres, le service à la clientèle et les relations avec les partenaires. Il concerne aussi bien l'organisation du travail dans l'entreprise que sa façon de communiquer et d'échanger des données avec ses clients, ses sous-traitants, ses fournisseurs et ses partenaires.

Notre projet se concentre sur le commerce électronique qui désigne aujourd'hui une nouvelle façon d'acheter ou de vendre, tout en restant confortablement installé derrière son écran. Pour les entreprises, il s'agit d'un nouveau moyen de commercialisation de biens et services extrêmement rentable à condition d'employer les bonnes stratégies au bon moment...

1. Le commerce électronique

Selon la définition de l'OCDE¹, le commerce électronique ou e-commerce en anglais, est « la vente ou l'achat de biens ou de services, effectués par une entreprise, un particulier, une administration ou toute autre entité publique ou privée et réalisé au moyen d'un réseau électronique ». On distingue :

- L'échange électronique entre entreprises, souvent appelé B2B (acronyme anglais de Business to Business).
- Le commerce électronique à destination des particuliers, ou B2C (Business to Consumer). Il s'agit de sites web marchands, type télé-achat.
- Le commerce électronique entre particuliers, ou C2C (Consumer to Consumer). Il s'agit de sites web permettant la vente entre particuliers (immobilier, bourses, annonces, échanges...).
- L'échange électronique entre les entreprises privées et le gouvernement, souvent appelé B2G (Business to Government) ou B2A (Business to Administration).

Parmi les principaux biens et services vendus par internet aux particuliers (B2C), on peut citer :

- Le matériel informatique, hi-fi, vidéo, photo...
- Le secteur du tourisme et des loisirs : séjours, voyages, locations, billets de train, d'avion...
- Les biens de consommation courante (vêtements, meubles, électroménager, jouets).
- Les biens culturels, comme les: livres, CD et DVD

On trouve des systèmes de vente spécialement adaptés à internet : développement de photographies numériques, téléchargement de musique, vente aux enchères entre particuliers, location de DVD par internet.

¹ Organisation de coopération et de développement économiques est un est une organisation internationale d'études économiques.

2. Les solutions existantes sur le marché tunisien

Pour arriver à notre objectif, il nous a fallu effectuer une étude des les solutions similaires existantes sur le marché tunisien en vue de recueillir leurs points faibles pour pouvoir les éviter et les améliorer dans notre travail. Pour cela nous présenterons les fonctionnalités fournies par un échantillon représentatif de sites. Nous nous limiterons à étudier le site proumouv.com et affariyet.com.

a. Promouv.com : www.promouv.com

Le site promouv.com propose un choix complet de plus de 3 000 références de produits parmi les plus grandes marques déposées pour vente en ligne. Il offre :

- La recherche des produits par catégorie
- La description du produit
- Suivi des achats pour un client
- Possibilité d'avoir un compte client
- L'achat en ligne

Les deux figures suivantes présentent deux différentes fonctionnalités proposés par Proumouv.com La Figure 2 concerne le compte client, où se trouve les différentes informations en relation du client inscrit. La Figure 3 montre l'interface de l'achat en ligne proposée par ce dernier.

The screenshot shows the 'Mon Compte' (My Account) section of the Promouv.com website. On the left, there's a sidebar with a navigation menu including 'Tableau de bord du compte', 'Informations du compte', 'Carnet d'adresses', 'Mes commandes', 'Ententes de facturation', 'Profils récurrents', 'Mes commentaires', 'My Applications', 'Abonnements à la newsletter', and 'Mes produits téléchargeables'. Below this is another 'NAVIGATION' section with 'Petit Electro'. The main content area is titled 'Mon tableau de bord' (My Dashboard) and greets the user 'Bonjour, Mlle marwa briki !'. It explains that the dashboard provides a summary of recent activities and allows updating account information. It includes sections for 'INFORMATIONS DU COMPTE' (Contact info, Newsletter subscription), 'Carnet d'adresses' (Address book), 'ADRESSE DE FACTURATION PAR DÉFAUT' (Default billing address), and 'ADRESSE DE LIVRAISON PAR DÉFAUT' (Default delivery address). Each section has edit links.

Figure 2 Compte Client du site promouv.com

The screenshot shows the 'Commander' section of the website. It includes fields for 'Prénom' (First Name), 'Nom' (Last Name), 'Société' (Company), 'Abonnez-vous à la newsletter' (Subscribe to newsletter), 'Adresse' (Address), 'Ville' (City) set to 'Ariana', 'État/Région' (Region), 'Code postal' (Postal code), 'Pays' (Country) set to 'Tunisie', 'Téléphone' (Phone), 'Fax', and two radio button options for delivery address. Below these are tabs for 'Informations de livraison', 'Mode de livraison', 'Informations de paiement', 'Procédure de paiement', and 'Vérification de commande'. A sidebar on the right lists command status steps: 'AVANCEMENT DE VOTRE COMMANDE' (Facturation address, Shipment address, Delivery method, Payment method), 'RETROUVEZ-NOUS SUR FB' (Facebook link), and 'Promouv' logo.

Figure 3 Paiement en ligne du site promouv.com

En navigant sur ce site et exploitant ses services, nous avons rencontré une erreur se produisant lors du clique sur un lien dans l'interface du compte client de la Figure 2. La Figure 4 suivante montre l'erreur rencontrée.

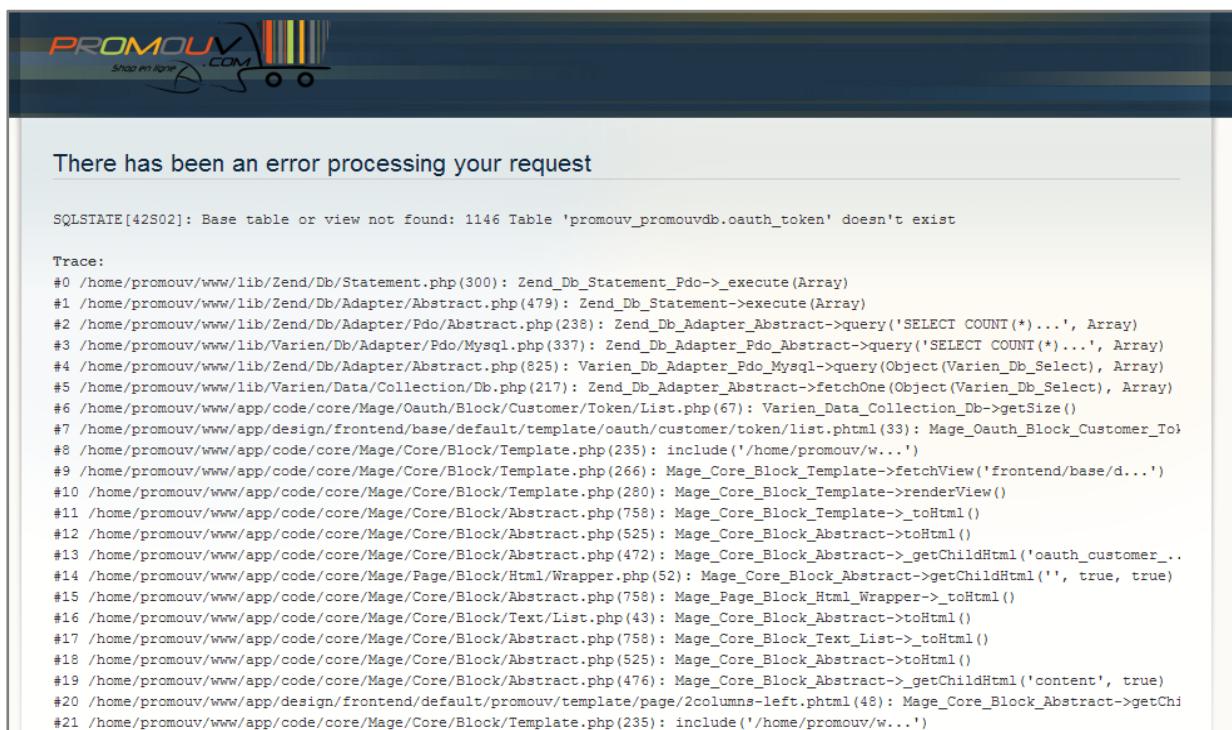


Figure 4 Erreur compte client du site promouv.com

Une autre remarque lors de l'étude de cette solution, c'est qu'ils offrent la possibilité de démarrer une discussion instantanée mais en réalité c'est juste un envoi de message tel que présente la Figure 5 .

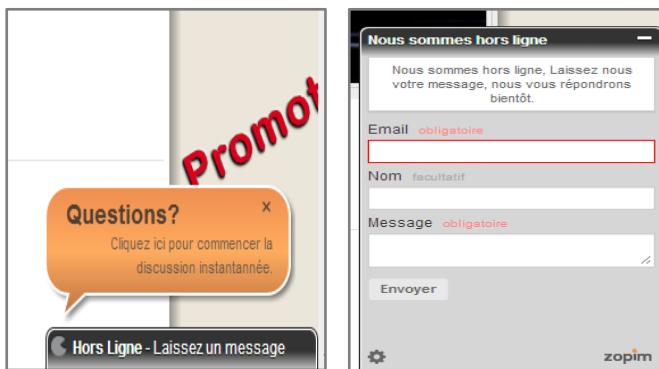


Figure 5 Discussion instantanée du site promouv.com

a. affariyet.com : www.affariyet.com

Affariyet c'est une boutique en ligne qui propose un ensemble de produits réservé exclusivement aux acheteurs sur le territoire tunisien, il assure les fonctionnalités suivantes :

- La recherche des produits par catégorie.
- Possibilité d'avoir un compte client « voir Figure 6 ».
- L'achat en ligne « voir Figure 7 ».

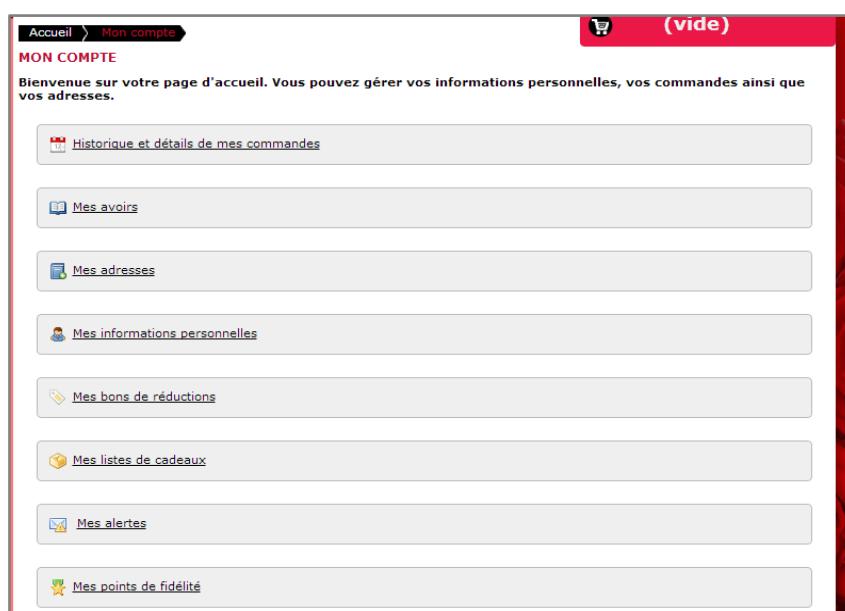


Figure 6 Compte client du site affariyet

Le paiement en ligne au niveau d'affariyet.com s'effectue sur deux étapes. D'abord, c'est l'enregistrement de la commande tel que montre la Figure 7, une fois cette étape s'est bien accomplie le client doit attendre la validation de la commande par un commercial qui lui enverra la notice de paiement pour validation.

Cette méthode peut être jugé très lente qui peut engendrer la perte des clients.



Figure 7 Paiement en ligne du site affariyet

b. Tableau comparatives

Nous avons dressé un tableau comparatif récapitulant les différentes fonctionnalités offertes par ces deux solutions, et des objectifs que nous nous sommes fixés pour notre solution futur. C'est clair qu'actuellement, sur le marché tunisien, il n'y a pas de solution qui remplit tous les objectifs visés.

	Proumouv.com	affariyet.com
Compte Client	✓	✓
Compte Commerçant	✗	✗
Recherche	✓	✓
Achat En ligne	✓	✓
Forum	✗	✗
Les nouveaux produits	✗	✓
Produits les plus appréciés	✗	✗

Tableau 1 Tableau comparatif des sites

3. Solution sur le marché international

Pour réaliser ce travail nous nous ne sommes pas contentés juste d'étudier des solutions tunisiennes, nous nous sommes inspirés aussi d'autres solutions internationales qui sont plus développées. Nous prendrons l'exemple du site tinydeal. Certes, il y a des sites beaucoup plus développés comme Amazon ou encore Expedia, mais on s'est limité aux sites dont les fonctionnalités sont utiles et envisageables pour la communauté visée et s'apparente plus aux habitudes du citoyens Tunisiens.

a. tinydeal : www.tinydeal.com

Tinydeal est une boutique chinoise en ligne. Elle offre différentes fonctionnalités comme l'inscription d'un client, la recherche par catégorie « voir Figure 8 » ou encore la vente en ligne et la messagerie instantanée « voir Figure 9 ».



Figure 8 Recherche du site tinydeal

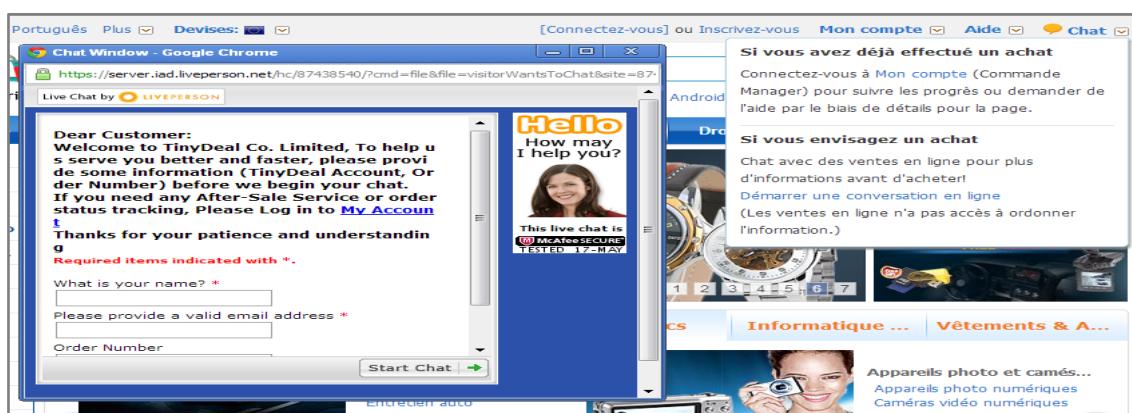


Figure 9 Chat du site tinydeal

La messagerie instantanée fournit par tinyDeal permet à un utilisateur de contacter un représentant du site dans le but d'assister les différents visiteurs de ce dernier. Un assistant peut être en communication instantanée avec 20 ou 30 personnes ce qui peut engendrer un retard pendant les réponses, aussi il n'y a pas d'interaction entre utilisateurs c'est juste une interaction avec un représentant du site.

II. Environnement technique de développement

Lors du développement de larges applications Web avec une panoplie de fonctionnalités à offrir, l'utilisation d'un Framework est devenue indiscutable.

1. Pourquoi un Framework

Lors de la création d'applications, utiliser un Framework améliore le développement en automatisant certaines tâches récurrentes. En apportant une structure, un Framework guide le développeur lors de l'écriture. Il l'aide à produire un code plus propre, plus efficace et plus facile à maintenir. En fait c'est un ensemble de bibliothèques et de conventions permettant le développement rapide d'applications. Il fournit suffisamment de briques logicielles et impose suffisamment de rigueur pour pouvoir produire une application aboutie et facile à maintenir. Ses composants sont organisés pour être utilisés en interaction les uns avec les autres.

2. Les Framework les plus utilisés

Pour effectuer un choix d'un Framework il faut avoir une idée sur les différents concurrents sur le marché, pour cela nous avons choisi d'étudier les plus utilisés pour développer des applications web e-commerce en PHP. Les Framework qui seront étudiés sont Cake PHP, Zend, Symfony et Codeigniter.

a. Cake PHP

Il fournit une base robuste pour votre application. Il peut se charger de chaque aspect, depuis la requête initiale de l'utilisateur jusqu'au rendu final d'une page web. Et puisque le

Framework suit les principes du patron de modélisation MVC (Modèle-Vue-Contrôleur), il permet de personnaliser et d'étendre facilement la plupart des aspects de l'application.

b. ZEND

Zend est un Framework pour développer en PHP5 créé en mars 2006 par Zend Technologies. Il est distribué sous la Licence BSD Modifiée. Le Zend Framework a été développé dans le but de simplifier le développement Web tout en recommandant les bonnes pratiques et la conception orientée objets en offrant des outils aux développeurs. Il permet aussi d'utiliser nativement le principe de MVC (Modèle-Vue-Contrôleur) mais ne l'oblige pas.

c. Symfony

C'est un Framework MVC open source, très puissant. Il est écrit en PHP 5. Il a pour vocation à simplifier la conception d'applications web de qualité. De plus SYMFONY est adaptable aussi bien, sur les plateformes Windows que sur les plateformes Unix dotées d'un serveur web.

d. CodeIgniter

CodeIgniter est un Framework PHP puissant, bien que léger, basé sur le fonctionnement de Ruby on Rails² et adoptant l'architecture MVC (Modèle-Vue-Contrôleur). Il a été développé à l'origine pour les propres besoins de l'entreprise EllisLab³ en 2006, il a depuis adapté le statut open source. Il est en constante évolution et son importante collection de librairies contribue largement aux gains de productivité que l'on attend d'un Framework, en matière de développement d'applications de toutes natures.

² Egalement appelé RoR ou Rails est un Framework web libre écrit en Ruby. Il suit le patron de conception MVC.

³ Société de développement logiciel situé à Bend, Oregon, elle développe que les applications PHP.

Voici un tableau comparatif de différents Framework :

	CodeIgniter	Symfony	Zend	CakePhp
Comptabilité PHP	Php4, Php5	Php5	Php5	Php4, Php5
Génération rapide des maquettes	✓	✓	✓	✓
Architecture MVC	✓	✓	✓	✓
Simplicité de prise en main	✓	✗	✗	✓
Rapidité	✓	✗	✗	✗

Tableau 2 Tableau comparatif des Framework

Notre choix s'est orienté vers le Framework Codeigniter. L'explication de ce choix ainsi qu'une étude technique de ce Framework seront fourni dans la section suivante.

3. Etude du Framework Choisi : le Framework Codeigniter :

Le choix de ce Framework ne s'est pas basé seulement sur les points forts de ce dernier en général, mais selon nos besoins en termes de fonctionnalité pour accomplir le présent projet. En effet, Codeigniter présente un ensemble intéressant de librairies riches en fonctionnalité. La plus intéressante dans notre cas c'est la librairie « cart » qui fournit plusieurs services pour la réalisation du panier dans un site e-commerce, une fonctionnalité centrale dans la solution visée. D'autres librairies semblent très importantes pour la réalisation de notre application comme la librairie « form_validation » qui sert à valider les règles déjà définies par le développeur pour un formulaire, la librairie « Email » est aussi très utile pour notre application puisque nous utiliserons l'envoi d'emails à différents niveaux. Nous aurons besoin d'afficher un nombre important de produit soit pour consultation chez un client ou bien au niveau de la gestion des produits pour commerçant, pour cela nous aurons besoin d'un mécanisme de pagination qui est offert par la librairie « pagination » de Codeigniter.

L'utilisation des sessions utilisateurs est primordial pour le présent projet, pour cela nous aurons besoin de différents services qui seront offerts par la librairie « session » de Codeigniter.

Codeigniter présente un ensemble important de « helpers », c'est un ensemble de fonctions réunies par thème. Il nous donne aussi la possibilité, soit de créer nos propres « helpers » ou bien de modifier ceux fournis par le Framework.

Le système de cache de Codeigniter lui procure d'excellentes performances et minimise l'impact sur le chargement des pages. Ce qui est très utile dans notre cas pour garder les utilisateurs de la plateforme et ne se dirige pas vers d'autres concurrents.

Enfin, Codeigniter est bien connu pour être le Framework idéal pour le développement des petites et moyennes applications. En effet il nécessite peu de ressources tout en offrant un maximum de fonctionnalités, ce qui est bien le cas de notre solution.

→ **Structure de CodeIgniter**

Dans ce qui suit une explication du fonctionnement du Framework ainsi que de ses principaux composants que nous utiliserons : CodeIgniter est composé de deux dossiers application et system ainsi qu'une page index.php. Le dossier system contient tous les codes natifs de CodeIgniter et le dossier application contient tous les codes de l'application à réaliser.

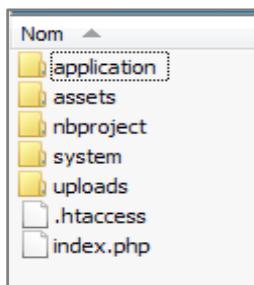


Figure 10 Structure du Framework Codeigniter

Nous pouvons ajouter des dossiers pour les besoins de l'application, le dossier assets qui contient les dossiers « css », « images » et « JavaScript », ainsi que le dossier uploads qui contiendra les fichiers qui seront téléchargés vers l'application.

Structure du dossier application dans lequel se trouve le code de l'application.

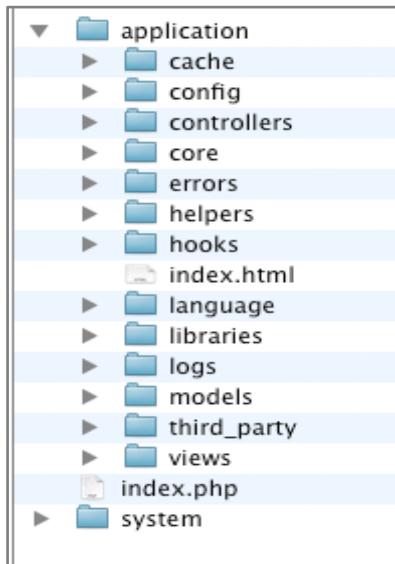


Figure 11 Structure du Framework Codeigniter 2

- **Config** : ce sont des fichiers permettant de configurer Codeigniter ou une bibliothèque. Certains sont inclus automatiquement, d'autres seulement lorsque vous les demandez.
- **Controllers** : ce dossier contiendra tous nos contrôleurs.
- **Errors** : ce sont les pages d'erreurs. Qui seront personnalisé selon le design de l'application.
- **Helpers** : c'est le répertoire pour les helpers. Les helpers fournis avec Codeigniter sont situés dans le dossier system.
- **Hooks** : c'est un dossier qui contient des fichiers assez complexes. Ils permettent d'exécuter des scripts à différents moments du processus d'exécution de Codeigniter.
- **Language** : ce répertoire contiendra tous les fichiers de langue dans le cas où on souhaite un site internationalisé.
- **Libraries** : nous placerons nos bibliothèques dans ce dossier. Comme les helpers, les bibliothèques fournies par Codeigniter sont situées dans un autre dossier.
- **Models** : le répertoire des modèles.
- **Views** : le répertoire des vues.

→ **Principe De fonctionnement**

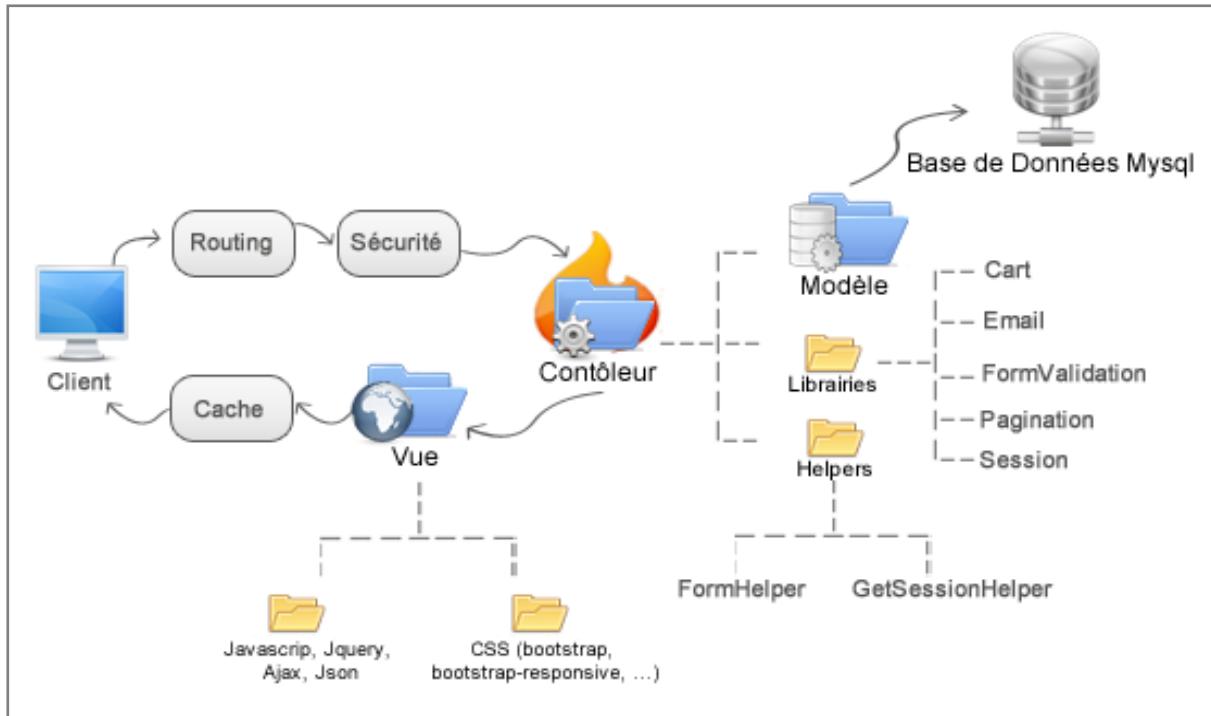


Figure 12 Fonctionnement du Framework Codeigniter

Une requête cliente sera routée vers le contrôleur en question, elle passe par les règles de sécurité mise en place pour arriver enfin au contrôleur. Ce dernier communique avec les différents modèles, librairies, helpers, etc. qui accèdent par suite à la base de données en cas de besoin. Une fois la requête est traitée, le contrôleur passe le résultat à la vue qui sera affiché au client.

III. Sécurisation des applications web

Une application Web est une application qui n'a besoin que du protocole HTTP pour être pilotée par un utilisateur. Celui-ci utilise un simple navigateur Web ou bien une application propriétaire utilisant le protocole HTTP.

Pour permettre l'utilisation d'une application Web, il suffit que le firewall protégeant celle-ci ne laissant entrer que le protocole HTTP sur le port 80 (et éventuellement HTTPS sur le port 443). Dans ces conditions, il semble difficile d'attaquer une application Web. Pourtant, à travers ce seul port 80, considéré comme « amical » mais devenu un port

« fourre-tout » par lequel passent de plus en plus de flux et de protocoles, il est possible de lancer des attaques extrêmement dangereuses. Les attaques que nous citerons sont l'interprétation des urls, Mauvais contrôle des données entrées par l'utilisateur, l'injection de code sql et Attaques sur les identifiants de session.

Nous allons développer notre application avec les mécanismes requis pour faire face à tous ces attaques.

1. Interprétation des URL:

Les URL utilisées dans le cadre d'une application Web sont du type :

`http:// www.monserveur.com/chemin/fichier.ext?param1=toto¶m2=alaplage`

Une URL de ce type est composée de plusieurs parties, nous citons le protocole utilisé, l'adresse du serveur, le chemin, le fichier concerné et l'ensemble de paramètres. Chacune de ces parties est susceptible d'être attaquée :

- **Protocole**: on peut essayer de remplacer le protocole `https://` par `http://`, par exemple, afin de désactiver une authentification par certificat client.
- **Serveur** : on peut le remplacer par son adresse IP ou par les noms de domaines d'autres sites hébergés sur le même serveur, afin d'avoir accès à d'autres parties du site.
- **Chemin** : on peut tenter de naviguer dans l'arborescence pour accéder à des parties du site non autorisées ou pour remonter dans l'arborescence par l'utilisation des « `../..` », ou en utilisant des vulnérabilités particulières (le bug Unicode sur IIS, par exemple).
- **Fichier** : son extension va déterminer de quel type d'exécutable il s'agit : CGI, scripts ASP, HTR ou autre code exécutable, etc. Plusieurs types de fichiers ont connu des vulnérabilités attachées à leur mode d'exécution.

2. Mauvais contrôle des données entrées par l'utilisateur :

Nous rencontrons encore trop souvent des applications Web où le contrôle des entrées par l'utilisateur se fait uniquement côté client. Dans ce cas, il faut toujours considérer qu'un attaquant pourra outrepasser le contrôle côté client et donc envoyer les données

qu'il désire au serveur. Nous pouvons citer l'exemple de certains caractères spéciaux dont la signification, au sein d'une chaîne alphanumérique, peut respecter une syntaxe particulière au niveau d'un langage de programmation ou d'un système d'exploitation, ce qui peut conduire à l'exécution de commandes hostiles.

3. Injection de code SQL :

L'injection SQL peut être une conséquence directe d'un mauvais contrôle des données entrées par l'utilisateur. En effet, les caractères « ' » et « ; » peuvent être utilisés pour enchaîner plusieurs requêtes SQL à la suite.

Supposant que nous avons la requête suivante tournant sur le serveur :

```
SELECT * FROM table_Clients WHERE champ_Nom=Name
```

Un attaquant peut saisir la requête suivante :

```
1; INSERT INTO table_Users VALUES ('Mon_login', 'Mon_password')
```

Et par suite la requête exécutée sera la suivante :

```
SELECT * FROM table_Clients WHERE champ_Nom=1; INSERT INTO table_Users VALUES  
('Mon_login', 'Mon_password')
```

Ce qui donne que le résultat est l'affichage des données concernant le client 1, suivi par l'ajout dans la base de données d'un utilisateur dont le couple login/mot de passe permettra à l'attaquant de se connecter à l'application comme un utilisateur authentifié.

4. Attaques sur les identifiants de session :

La plupart des systèmes d'authentification et de maintien du contexte des sessions Web des utilisateurs repose sur un identifiant de session, échangé à chaque page entre le client et le serveur, que ce soit au niveau du cookie, de l'URL ou d'un champ caché de formulaire. Une attaque classique consiste à voler la session d'un utilisateur qui vient de s'authentifier sur le système en essayant de deviner la valeur de son identifiant de session. Si la valeur de celui-ci est découverte, un attaquant peut alors utiliser l'application Web en lieu et place de l'utilisateur légitime.

IV. Application mobile

Récemment, les habitudes d'utilisation du téléphone sont en train de changer et nous nous connecterons plus de notre mobile que depuis notre ordinateur, les recherches mobiles ont été multipliés par 4 depuis 2010 et il y aura autant de mobile que de personnes sur terre d'ici à 2015 selon Gartner⁴(2010). Pour suivre cette évolution importante des mobiles nous allons intégrer un volet mobile pour notre application.

On appelle commerce mobile ou m-commerce toute activité commerciale passant par les technologies sans fil telles que les tablettes et les Smartphones. A l'âge d'or de l'internet mobile, le m-commerce est naturellement voué à se développer de plus en plus.

Avoir un site de m-commerce favorise le **drive to store**, c'est-à-dire le fait de voir un produit ou une enseigne sur son appareil mobile et de se rendre directement ensuite en magasin, notamment grâce à la géolocalisation.

C'est aussi vrai avec la redirection vers le **e-commerce** : 30% des utilisateurs surfent sur leur Smartphone tout en étant connectés à internet sur un autre équipement, ce qui favorise ce phénomène.

Le m-commerce possède aussi des inconvénients, l'étude du cabinet « Research Now » datant d'avril 2012 a sondé des individus sur ce qu'ils percevaient comme des freins à l'achat sur mobile. L'inaccessibilité à internet depuis les terminaux mobiles figure ainsi comme la première difficulté (59%). L'enquête fait également état d'un manque de maîtrise des fonctionnalités des mobiles par leurs utilisateurs (24%) ainsi qu'une méconnaissance de l'univers internet mobile (27%). L'ergonomie est également reconnue comme un frein au m-commerce. En effet, 11% des équipés mentionnent comme obstacle principal l'aspect inconfortable de la navigation sur mobile.

⁴ Entreprise américaine de conseil et de recherche dans le domaine des techniques avancées dont le siège social est situé à Stamford

Enfin, les craintes liées à la sécurité du paiement en ligne valent autant, voire plus pour le mobile que pour les ordinateurs. Ainsi, 59% des personnes interrogés déclarent ne jamais avoir acheté avec leur mobile pour des raisons de sécurité.

Enfin une question se pose, Ce mode de vente à distance va-t-il continuer à se développer ? Le m-commerce pourrait-il, à terme, concurrencer le e-commerce sur ordinateur ? La réponse est oui en raison du confort d'utilisation qui ne cesse d'augmenter sur les Smartphones.

Malgré toutes ces craintes, le développement d'un volet mobile pour toute application Web est indispensable. C'est pour cette raison que nous passerons en revues les différentes technologies et solutions mobiles pour développer ce volet.

1. Solutions mobile

La révolution des Smartphones s'évolue de plus en plus, conduisant les entreprises à se tourner très largement vers une solution mobile. La contrainte courante que rencontrent ces sociétés lors de leur entrée dans le monde du mobile est le choix de la stratégie à adopter face à la variété de plateformes existantes (Android, Blackberry, Windows Phone, Palm WebOS, Bada et Symbian). L'un des premiers choix à faire est de savoir vers quel type d'application il faut s'orienter. Il existe trois types d'applications mobiles, solution native, web Apps et solution hybride.

a. Solution native :

Les applications natives sont des logiciels conçus spécifiquement pour une plate-forme mobile basée sur un langage natif par exemple une application native iOS sera écrite en objective-C tandis qu'une application native Android sera écrite en Java. Les applications ainsi créées sont ensuite téléchargeables depuis une plateforme dédiée au système, généralement un magasin d'application type App Store d'Apple ou Android Market. Ces applications permettent de tirer parti de toute la puissance et toutes les possibilités du device mobile : accélération matérielle, capteurs, caméra, accès aux contacts, aux fonctions de téléphonie, etc.

b. Solution Web Apps :

Il est possible de développer, avec les technologies web récentes, des applications dont l'ergonomie est adaptée aux Smartphones. On accède à ces applications web via le navigateur internet du device mobile. On peut parler ici de la notion du Responsive Web Design (conception de sites web adaptatifs) qui est une technique de réalisation de site web, permettant de faire des sites qui s'adaptent automatiquement à l'espace disponible sur l'écran en se basant sur les Media Queries. (Fonctionnalité offerte par le langage CSS).

Au niveau de notre application nous avons appliqué cette notion en utilisant le Framework css « Twitter Bootstrap » qui sera expliqué plus tard dans le rapport.

c. Solution hybride

Les applications hybrides sont nées il y a quelques années avec des Framework tels que Phonegap ou Titanium. Ces Framework permettent de coder une application une fois en langage web (HTML/CSS/JavaScript) pour être déployée vers plusieurs plateformes différentes. L'application sera également disponible dans l'app store de la plateforme. Il existe différents types de solutions hybrides, chacun adoptant une stratégie différente. C'est ce mode de développement du volet mobile que nous avons choisi.

Nous avons choisi ce type de solution pour plusieurs raisons.

- Développement avec des outils et des langages plus accessibles (HTML, CSS, Javascript).
- Développement iPhone et Android en même temps en se basant sur les technologies standards, connues et exécutées par tous les smartphones du marché.
- Temps de développement réduit.

PhoneGap est le Framework qui sera utilisé pour le développement de notre application mobile.

→ **Framework PhoneGap**

PhoneGap, est un Framework open-source développé par Nitobi Software, qui permet de créer des applications hybrides avec les langages de développement web : HTML, CSS,

JavaScript. Cette appellation a évolué et on devrait plutôt parler de (Apache) **Cordova**. En effet, suite au rachat de Nitobi par Adobe, le nom du Framework a changé intégralement passant de PhoneGap à Apache Callback puis en s'appelant Apache Cordova.

Le tableau comparatif ci-dessous permet de savoir plateforme par plateforme Les APIs des OS pouvant être utilisés par PhoneGap:

	iPhone / iPhone 3G	iPhone 3GS and newer	Android	Blackberry OS 5.x	Blackberry OS 6.0+	WebOS	Windows Phone 7	Symbian	Bada
Accelerometer	✓	✓	✓	✓	✓	✓	✓	✓	✓
Camera	✓	✓	✓	✓	✓	✓	✓	✓	✓
Compass	✗	✓	✓	✗	✗	✓	✓	✗	✓
Contacts	✓	✓	✓	✓	✓	✗	✓	✓	✓
File	✓	✓	✓	✓	✓	✗	✓	✗	✗
Geolocation	✓	✓	✓	✓	✓	✓	✓	✓	✓
Media	✓	✓	✓	✗	✗	✗	✓	✗	✗
Network	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Alert)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Sound)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Vibration)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Storage	✓	✓	✓	✓	✓	✓	✓	✓	✗

Figure 13 Les API de PhoneGap

Parmi ces API nous utiliserons l'api **Storage** afin de pouvoir stocker les données du panier dans une base local pour ensuite l'envoyer à notre plateforme e-Commerce. Aussi, nous utiliserons l'api Event qui a pour but de déclencher un événement lorsque PhoneGap est entièrement chargé.

```
document.addEventListener("deviceready", yourCallbackFunction, false);
```

Généralement, nous devons ajouter un écouteur d'évènement via `document.addEventListener` dès le chargement du DOM de la page HTML chargé.

Pour pouvoir créer une application phonegap il faudra tout d'abord choisir la plateforme sur laquelle nous souhaitons développer. Le tableau ci-dessous récapitule la configuration par appareil.

IOS	Android	WindowsPhone
<ul style="list-style-type: none"> → MAC OS X Snow Leopard → La certification iOS développeur (pour publier votre app) → Xcode → Le SDK (kit de développement) de PhoneGap 	<ul style="list-style-type: none"> → Eclipse 3.4+ → Le SDK Android → Le plugin ADT pour Eclipse (Outils de développement Android) → Le SDK de PhoneGap 	<ul style="list-style-type: none"> → Windows 7 ou Vista SP2 → Le SDK des Windows Phone → Le SDK de PhoneGap

Figure 14 Plateformes de développement

Pour notre application nous utiliserons Android vu qu'il est open source.

Une partie **PhoneGap Build** a été mise en place pour faciliter l'utilisation aux développeurs. En créant un compte sur la partie Build du site PhoneGap, il est possible d'uploader les fichiers sources de notre application et ainsi télécharger les packages des plateformes correspondantes.

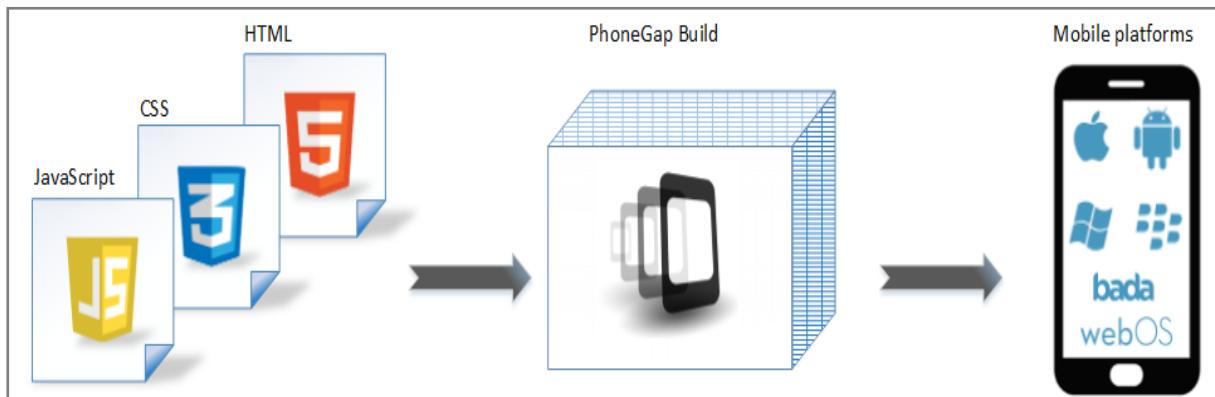


Figure 15 PhoneGap Build

Il y a **plusieurs avantages** à préférer cette méthode pour un usage quotidien :

- La vitesse d'utilisation
- La qualité et la sûreté du contenu : avant d'être disponible sur les stores, l'application est testée (plus ou moins bien en fonction du store) et validée par un groupe de testeurs.
- Bonne intégration avec jQueryMobile (jQM)
- Développement & Déploiement rapide des applications mobiles

Conclusion

Durant ce chapitre nous avons présenté les différents aspects en relation avec le projet allant du contexte du projet, passant par les Framework utilisés, une petite comparaison avec l'existant sur le marché arrivant à la sécurisation des applications web et nous avons terminé par la mise en œuvre du volet mobile de l'application.



Chapitre III Etude préliminaire



Introduction

L'étude préliminaire (ou Pré-étude) est la toute première étape du processus 2TUP. Un tel projet est très riche en fonctionnalités et détails, pour cela il faut un repérage des besoins fonctionnels et opérationnels, en utilisant principalement le texte, ou diagrammes très simples. Cette étape prépare les activités plus formelles de capture des besoins fonctionnels et de capture techniques.

I. Présentation du projet à réaliser

C'est une plateforme qui permettra de regrouper les différents commerçants en leur donnant la possibilité de présenter le commerce et d'effectuer les ventes en ligne. Cette plateforme fournira aux clients un environnement centralisé dans lequel ils trouveront les produits souhaités avec une simple recherche, ils pourront aussi effectuer les achats en ligne en toute sécurité. En plus des fonctionnalités commerciales, cette plateforme offrira un espace de communication entre les clients et les commerçants.

II. Identification des acteurs

Nous allons maintenant énumérer les acteurs susceptibles d'interagir avec le système :

- **Le commerçant** : C'est l'acteur qui va proposer ses services et ses produits aux différents clients.
- **Le client** : Il accède à la plateforme pour effectuer l'achat en ligne ou bien visualiser les différentes offres.
- **L'administrateur** : Il s'occupe de la gestion de la plateforme.
- **Module paiement** : C'est l'acteur qui va s'occuper de la gestion du paiement en ligne
- **Module paiement** : C'est l'acteur qui va s'occuper des notifications.

III. Recueil des besoins fonctionnels

- **Pour le commerçant** :

- Inscription (activation par mail et par l'Administrateur).
- Gestion du profil commerçant (modification).
- Gestion des Annonces (ajout, modification et suppression).
- Gestion des produits (ajout, modification et suppression).
- Gestion de la procédure de paiement.

- Gestion des messages (consulter, répondre).
- Gestion de l'espace forum (répondre, modification et suppression des réponses).
- Consulter les statistiques sur les produits.
- Consulter les Notifications.

→ **Pour le client**

- Inscription (Activation par mail).
- Gestion du profil client (modification).
- Consultation des produits et des annonces.
- Achat des produits.
- Suivie de ses achats.
- Envoi des messages aux commerçants.
- Recherche des Produits.
- Consultation du profil des commerçants.
- Attribuer une note aux produits.
- Gestion de l'espace forum (créer un sujet, répondre, modification et suppression des réponses).
- Consultation des notifications.

→ **Pour l'administrateur**

- Gestion des commerçants (Activer, Désactiver, supprimer).
- Supervision de plateforme et des statistiques (sur la plateforme : nombre des visiteurs, nombres des inscriptions...).
- Gestion des autorisations du forum (supprimer des réponses).

→ **Pour le module paiement**

- Gestion de procédure de paiement.

→ **Pour le module notification**

- Gestion des notifications et l'envoi des mails (Client, Commerçant et Administrateur).

IV. Recueil des besoins non fonctionnels

- L'application doit assurer la facilité de navigation et de recherche pour un client.
- L'application doit être robuste et performante en ce qui concerne le temps de réponse des requêtes et la notification qui doit se faire en temps réel.
- L'application doit assurer la sécurité du paiement en ligne.

V. Modélisation du contexte

Une fois le recueil des besoins est effectué, nous commençons la description du contexte du système. Elle consiste aux activités successives :

- L'identification des acteurs et messages.
- La réalisation du diagramme de contexte.

1. Identification des messages

Nous avons déjà identifié quatre acteurs se sont le système, l'administrateur, le commerçant et le client. Nous détaillerons maintenant les différents messages échangés entre le système et l'extérieur.

a. Acteur 1: Le commerçant

Messages émis	Messages reçus
<ul style="list-style-type: none"> - Envoi des informations concernant le profil lors de la création ou modification - Envoi des informations concernant l'annonce - Demande Modification ou suppression d'une annonce - Envoi des informations concernant le produit - Demande Modification ou suppression d'un produit - Demande des statistiques 	<ul style="list-style-type: none"> - Notification de la création du profil (envoi d'un mail) - Notification d'une annonce ajoutée (boite de dialogue). - Notification d'une annonce modifiée ou supprimée. - Notification d'un produit ajouté (boite de dialogue). - Notification d'un produit modifier ou supprimer - Affichage des statistiques

Tableau 3 Identification des messages de l'acteur commerçant

b. Acteur 2: Le client

Messages émis	Messages reçus
<ul style="list-style-type: none"> - Envoi des informations concernant le profil lors de la création ou modification - Envoi des données de modification du profil - Demande d'afficher les produits, les annonces - Demande d'achat - Demande d'afficher l'historique d'achat - Envoi de messages - Demande d'afficher un produit selon des critères de recherche - Demande de consultation du profil commerçant - Envoyer la note attribuée à un produit 	<ul style="list-style-type: none"> - Notification de la création du profil (envoi d'un mail) - Notification d'une mise à jour du profil (boîte de dialogue). - Affichage des annonces et produits clients - Affichage de l'interface du paiement en ligne - Affichage de l'historique des achats - Notification de succès de l'envoi - Affichage du profil commerçant

Tableau 4 Identification des messages de l'acteur Client

c. Acteur 3: L'administrateur

Messages émis	Messages reçus
<ul style="list-style-type: none"> - Demande d'afficher les différents commerçants (pour activation, désactivation ou bien suppression) - Demande de visualiser les statistiques concernant la plateforme 	<ul style="list-style-type: none"> - Affichage de la liste des commerçants, clients - Notification d'une mise à jour du profil (boîte de dialogue). - Affichage des statistiques

Tableau 5 Identification des messages de l'acteur Administrateur

2. Diagramme de contexte

A partir des informations obtenues lors des précédentes étapes, nous modéliserons le contexte de notre application. Ceci nous permettra de définir le rôle de chaque acteur dans le système :

Utilisateurs finaux	Descriptions des besoins fonctionnels
<u>Client</u>	<ul style="list-style-type: none"> - L'application doit permettre au client de : - S'inscrire - S'authentifier - Gérer son profil - Gérer ses messages envoyés et reçus - Gérer son espace forum - Consulter les différentes notifications reçus - Suivre ses achats - Effectuer la recherche - Noter les produits - Effectuer le paiement en ligne
<u>Commerçant</u>	<ul style="list-style-type: none"> - L'application doit permettre au commerçant de : - S'inscrire - S'authentifier - Gérer son profil - Gérer ses annonces et produits - Gérer ses messages - Gérer son espace forum - Consulter l'état de son commerce - Consulter les différentes notifications

Tableau 6 Rôle de chaque utilisateur finale

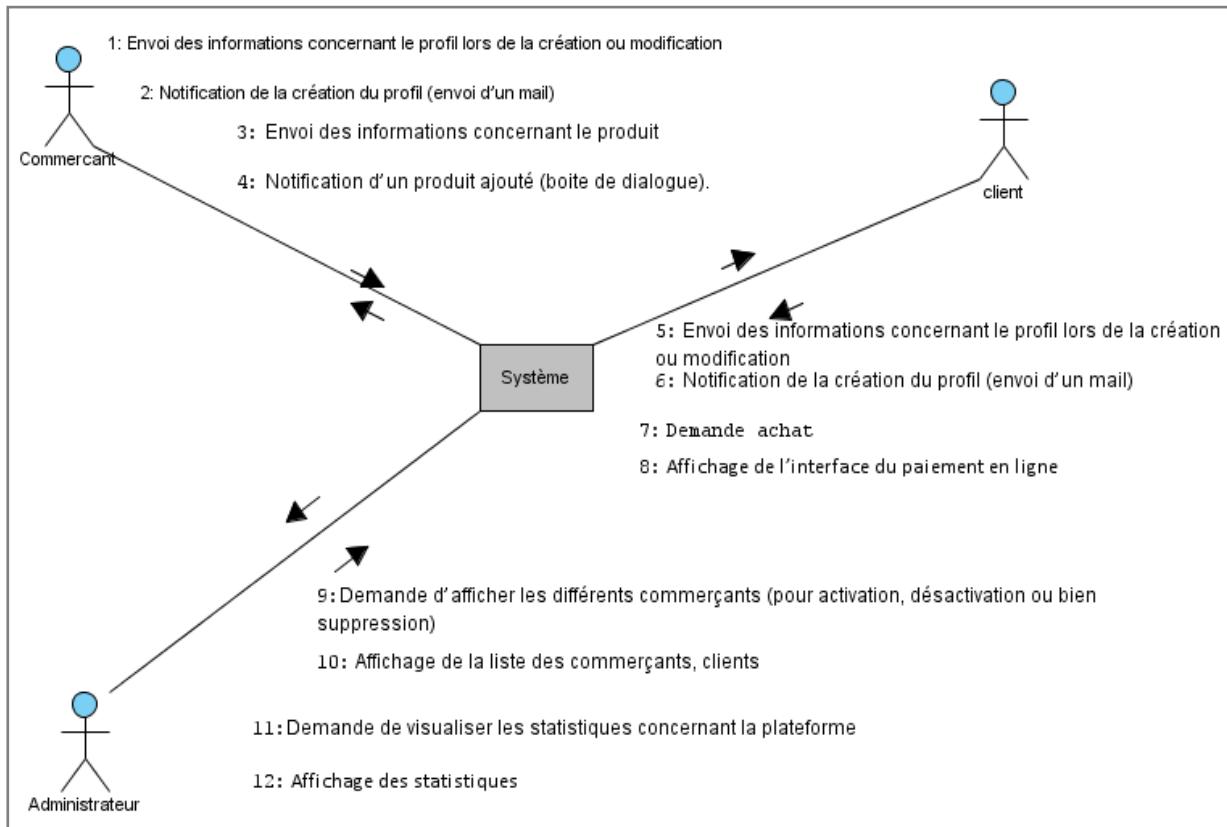


Figure 16 Diagramme de contexte

Conclusion

Cette phase nous a permis de mettre le système dans son contexte et de se préparer pour les activités formelles de capture des besoins fonctionnels et techniques qui seront l'objectif du prochain chapitre.



Chapitre IV Capture des besoins fonctionnels et techniques



Introduction

Cette phase représente un point de vue fonctionnel et technique de l'architecture système. Par le biais des cas d'utilisation, nous serons en contact permanent avec les acteurs du système en vue de définir les limites de celui-ci, et ainsi éviter de trop s'éloigner des besoins réels de l'utilisateur final. En deuxième partie nous définirons les besoins technique de notre application.

I. Capture des besoins fonctionnels

1. Liste préliminaires des cas d'utilisation

L'identification des cas d'utilisation, nous donne un aperçu des fonctionnalités futures que doit implémenter le système.

a. Identification des cas d'utilisation

Cas d'utilisation	Les acteurs du système	Message(s) émis / reçus
S'inscrire	- Le commerçant - Le client	- <u>Emis</u> : Émettre les données du profil - <u>Reçus</u> : Mail de confirmation
S'authentifier	- Le commerçant - Le client	- <u>Emis</u> : login et mot de passe. - <u>Reçus</u> : droit d'accès
Gérer profil	- Le commerçant - Le client	- <u>Emis</u> : Informations à modifier
Gérer les annonces	- Le commerçant - Administrateur	- <u>Emis</u> : ajout, modification, suppression - <u>Reçus</u> : notification de validation
Gérer les produits	- Le commerçant - Administrateur	- <u>Emis</u> : ajout, modification, suppression - <u>Reçus</u> : notification de validation
Gérer messages	- Le commerçant - Le client	- <u>Emis</u> : consulter, répondre et envoyer - <u>Reçus</u> : Message d'envoi (boîte de dialogue)
Gérer espace forum	- Le commerçant - Le client - L'administrateur	- <u>Emis</u> : répondre à une question, ajouter, supprimer un commentaire, ajouter des sujets - <u>Reçus</u> : notification des différents commentaires

Consulter l'état du commerce	- Le commerçant	- <u>Reçus</u> : Les différents états des transactions, ventes et statistiques
Consulter les notifications	- Le commerçant - Le client - L'administrateur	- <u>Reçus</u> : Affichage des différentes notifications
Gérer les achats	- Client - Administrateur	- <u>Emis</u> : Action réalisé (recherche, demande de visualisation...) - <u>Reçus</u> : Affichage du résultat demandé
Gérer utilisateurs	- L'administrateur	- <u>Emis</u> : activer, désactiver ou bien supprimer un utilisateur
Superviser la plateforme	- L'administrateur	- <u>Reçus</u> : Afficher l'état de la plateforme
Gestion du paiement	- Module paiement - Le commerçant	- <u>Emis</u> : Demande d'activation de la procédure de paiement - <u>Reçus</u> : notifie le commerçant de l'activation de la procédure
Effectuer le paiement	- Module paiement - Le client	- <u>Emis</u> : les données confidentiel du client (login, mot de passe, code de carte crédit) - <u>Reçus</u> : facture de succès de paiement ou notification d'échec
Gestion des notifications	- Module notification - L'administrateur - Le client - Le commerçant	- <u>Emis</u> : effectuer une tâche de la part de l'administrateur, client ou commerçant - <u>Reçus</u> : envoyer les notifications d'ajout, de validation, de refus, de suppression et de modification du commerçant, client, ou produit

Tableau 7 Identification des cas d'utilisation

b. Diagramme de cas d'utilisation général pour l'application web

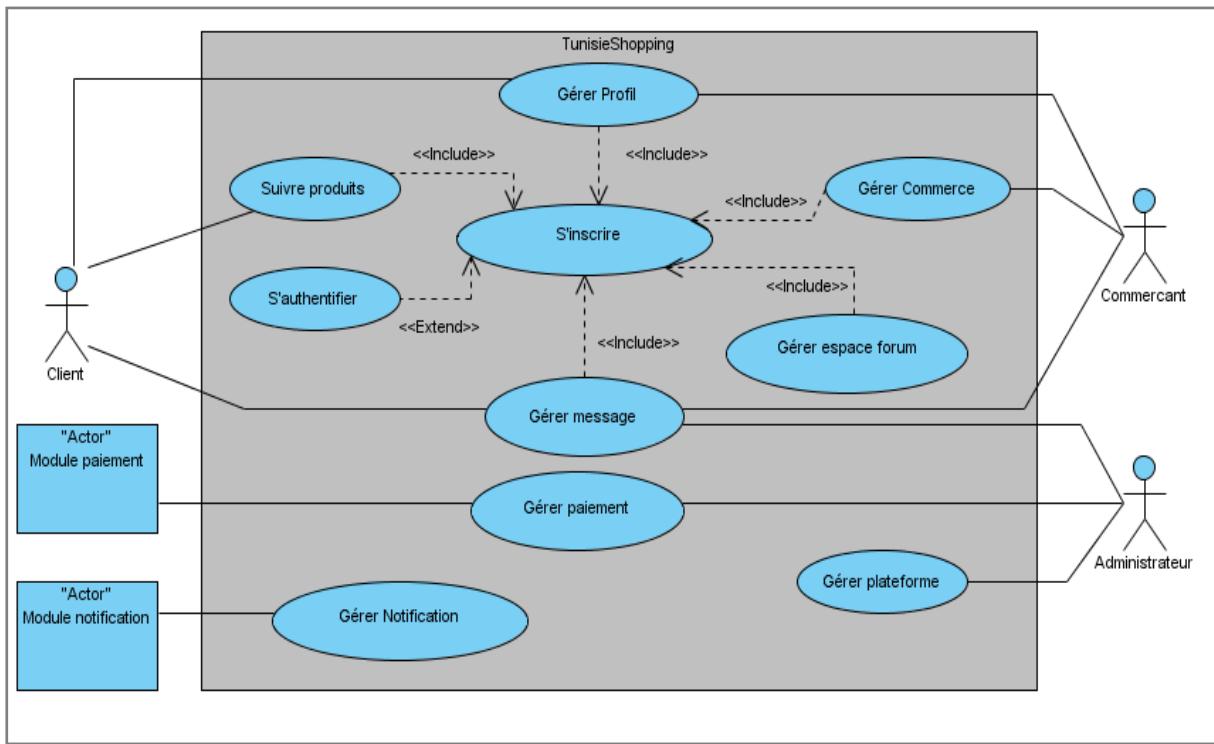


Figure 17 Diagramme de Cas d'utilisation général pour l'application web

Pour des raisons de lisibilité, nous avons regroupé certains cas d'utilisations au niveau de la figure qui seront détaillés au cours de ce chapitre.

Le cas d'utilisation « gérer commerce » regroupe les cas « gérer annonces, gérer produit et consulter l'état du commerce ». Pour le cas « suivre produits » il regroupe les cas « suivre achat, rechercher produits, effectuer paiement et noter produit ». « Gérer plateforme » c'est l'ensemble des cas « superviser plateforme, gérer utilisateurs ».

Le cas d'utilisation « gérer espace forum » est un cas composé de différents sous cas qui sont en relations avec différents acteurs d'où il sera détaillé plus tard.

II. Description des cas d'utilisation

Un cas d'utilisation fait l'objet d'une définition a priori qui décrit l'intention de l'acteur lorsqu'il utilise le système. Il décrit aussi les séquences d'actions principales que l'acteur est susceptible d'effectuer. Ces définitions servent à fixer les idées et n'ont pas pour but de spécifier un fonctionnement complet et irréversible.

1. Cas d'utilisation « S'inscrire »

a. Description textuelle de «S'inscrire»

Cas d'utilisation	S'inscrire
acteurs	<ul style="list-style-type: none"> - Le commerçant - Le client
résumé	<ul style="list-style-type: none"> - Chaque utilisateur doit s'inscrire en remplissant un formulaire et reçoit un mail d'activation de la part du système.
Enchaînement	<ul style="list-style-type: none"> - Pré conditions : _____ - Scenarios nominaux: <ul style="list-style-type: none"> → <u>Enchainement 1</u> : Choisir type du profil utilisateur (client /commerçant). → <u>Enchainement 2</u> : Remplir le formulaire Si login existe : [exception 1 : Login existant] → <u>Enchainement 3</u> : Activer profil L'utilisateur active son profil lorsqu'il reçoit un mail d'activation. - Post-conditions : Profil activé
Exception	<ul style="list-style-type: none"> - [Exception 1 : login existant] : le système affiche un message d'erreur et demande au utilisateur de changer le login.

Tableau 8 Description du cas d'utilisation « S'inscrire »

2. Cas d'utilisation « Gérer profil »

b. Description textuelle de «Gérer profil»

Cas d'utilisation	Gérer profil
acteurs	<ul style="list-style-type: none"> - Le commerçant - Le client
résumé	<ul style="list-style-type: none"> - Chaque utilisateur a le droit de faire des modifications dans son profil, il peut aussi modifier son mot de passe et bien sur visualiser son profil.
enchaînement	<ul style="list-style-type: none"> - Pré conditions : Utilisateur authentifié - Scenarios nominaux: <ul style="list-style-type: none"> → Enchainement 1 : Modifier profil Si login existe : [exception 1 : Login existant] → Enchainement 2 : Modifier mot de passe → Enchainement 3 : Voir profil - Post-conditions : Profil modifié, visualisé, mot de passe modifié.
Exception	<ul style="list-style-type: none"> - [Exception 1 : login existant] : le système affiche un message d'erreur et demande au utilisateur de changer le login.

Tableau 9 Description du cas d'utilisation « Gérer profil »

c. Diagramme de cas d'utilisation «Gérer profil»

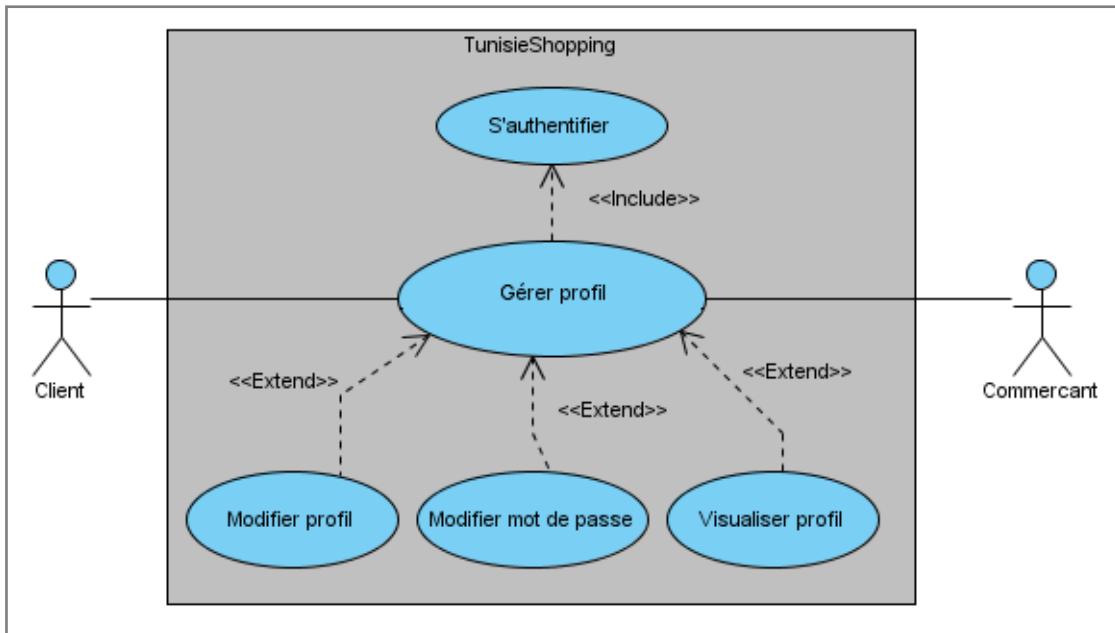


Figure 18 Diagramme de cas d'utilisation « Gérer profil »

3. Cas d'utilisation « Gérer les annonces»

a. Description textuelle de «Gérer les annonces»

Cas d'utilisation	Gérer les annonces
acteurs	- Le commerçant
résumé	- Chaque commerçant peut publier une annonce concernant un produit ou un service, il peut aussi la modifier ou bien la supprimer.
enchaînement	<ul style="list-style-type: none"> - Pré conditions : Commerçant authentifié - Scenarios nominaux : <ul style="list-style-type: none"> → Enchainement 1 : Publier annonce → Enchainement 2 : Supprimer annonce → Enchainement 3 : Modifier annonce - Post-conditions : Annonce publiée, supprimée, modifiée
Exceptions	_____

Tableau 10 Description du cas d'utilisation « Gérer les annonces »

b. Diagramme de cas d'utilisation «Gérer les annonces»

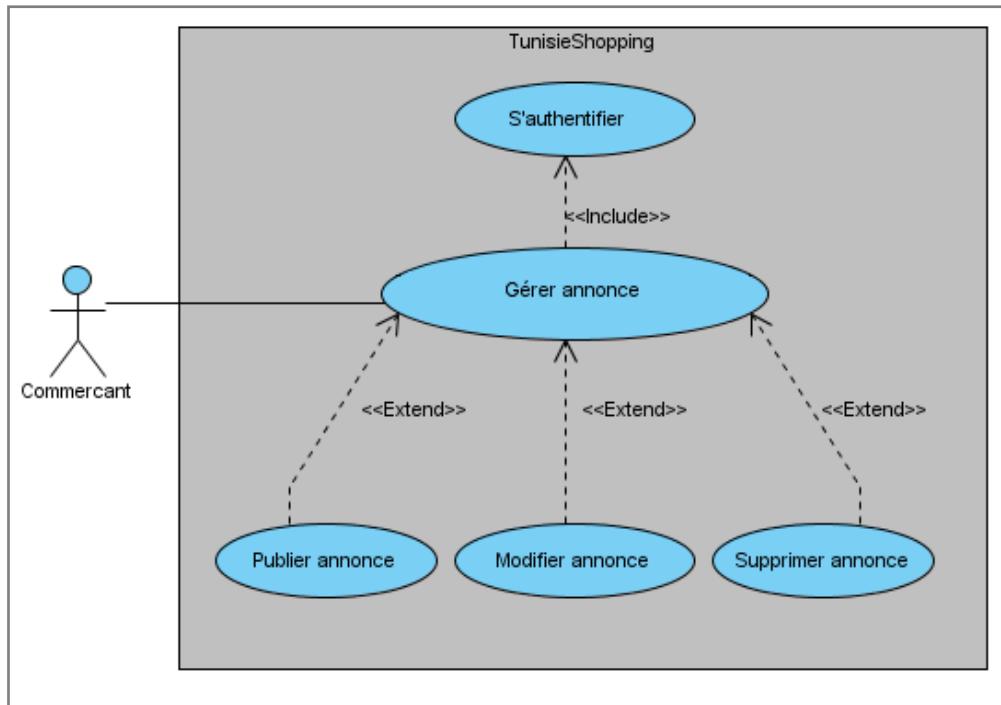


Figure 19 Diagramme de cas d'utilisation « Gérer annonces »

4. Cas d'utilisation « Gérer les produits»

a. Description textuelle de «Gérer les produits»

Cas d'utilisation	Gérer les produits
acteurs	- Le commerçant
résumé	- Chaque commerçant peut publier ses produits ainsi que toutes les informations nécessaires pour un produit.
enchaînement	<ul style="list-style-type: none"> - Pré conditions : Commerçant authentifié - Scenarios nominaux : <ul style="list-style-type: none"> → Enchainement 1 : Publier produit → Enchainement 2 : Supprimer produit → Enchainement 3 : Modifier produit - Post-conditions : Produit publié, supprimé, modifié
Exceptions	_____

Tableau 11 Description du cas d'utilisation « Gérer produit »

b. Diagramme de cas d'utilisation «Gérer produit»

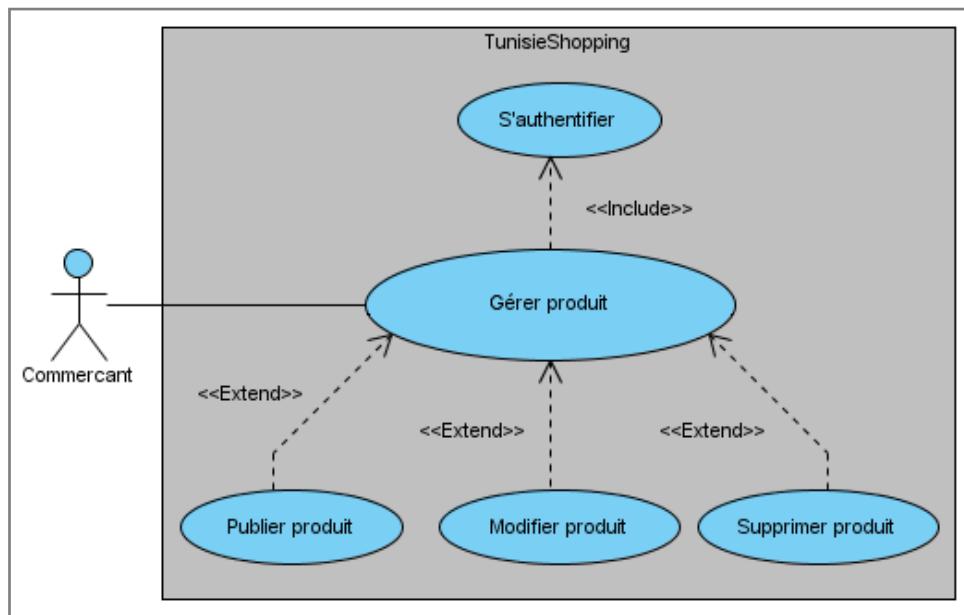


Figure 20 Diagramme de cas d'utilisation « Gérer produit »

5. Cas d'utilisation « Gérer messages»

a. Description textuelle de «Gérer messages»

Cas d'utilisation	Gérer messages
acteurs	<ul style="list-style-type: none"> - Le commerçant - Le client - L'administrateur
résumé	<ul style="list-style-type: none"> - Un client peut contacter un commerçant en lui envoyant un message privé, il peut consulter l'historique des ses messages ainsi que les réponses reçues de la part du commerçant - Un commerçant peut consulter sa boîte de réception messages et il peut répondre aux clients. - Un commerçant peut envoyer un message privé à l'administrateur.
enchaînement	<ul style="list-style-type: none"> - Pré conditions : Commerçant et/ou client authentifié - Scenarios nominaux : <ul style="list-style-type: none"> → Enchainement 1 : Créer message → Enchainement 2 : Consulter message → Enchainement 3 : Répondre à un message - Post-conditions : Message envoyé, reçu ou bien consulté
Exception	_____

Tableau 12 Description du cas d'utilisation « Gérer messages »

b. Diagramme de cas d'utilisation «Gérer messages»

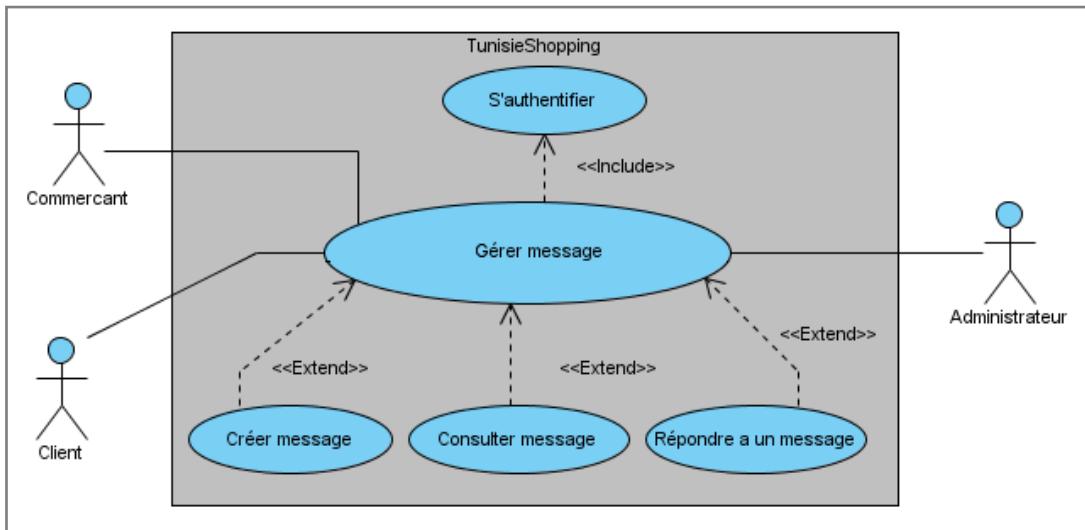


Figure 21 Diagramme de cas d'utilisation « Gérer messages »

6. Cas d'utilisation « Gérer espace forum »

a. Description textuelle de «Gérer espace forum»

Cas d'utilisation	Gérer espace forum
acteurs	<ul style="list-style-type: none"> - Le commerçant - Le client - L'administrateur
résumé	<ul style="list-style-type: none"> - Un commerçant peut ajouter ou bien modifier un sujet à discuter en forum, il peut aussi répondre aux questions posées par les clients - Un client peut créer un sujet, poser des questions, répondre. - Un commerçant, ou client peut signaler un commentaire - L'administrateur supprime les commentaires et/ou l'utilisateur signalé
enchaînement	<ul style="list-style-type: none"> - Pré conditions : Commerçant et/ou client authentifié - Scenarios nominaux : <ul style="list-style-type: none"> → Enchainement 1 : Ajouter sujet. → Enchainement 2 : Modifier sujet En cas de modification : [Exception 1 : notification] → Enchainement 3 : Marquer Sujet comme résolu. → Enchainement 4 : Créer commentaire En cas de publication d'un commentaire : [Exception 1 : notification] Si l'utilisateur essaie de publier un commentaire vide : [Exception 2 : Commentaire vide] → Enchainement 5 : Signaler commentaire Si l'utilisateur signale un commentaire : [Exception 3 : intervention de l'administrateur] → Enchainement 6 : Supprimer commentaire - Post-conditions : Notification envoyée, ou bien action exécutée

Exceptions	<ul style="list-style-type: none"> - [Exception 1 : Notification] : Le système envoie une notification au propriétaire du sujet dont on a ajouté un commentaire. - [Exception 2 : Commentaire vide] : Le système affiche un message d'erreur et demande à l'utilisateur de publier un commentaire non vide. - [Exception 3 : intervention de l'administrateur] : L'administrateur examine les commentaires signalés et prend les décisions nécessaires (suppression commentaire).
-------------------	---

Tableau 13 Description du cas d'utilisation « Gérer espace forum »

b. Diagramme de cas d'utilisation «Gérer espace forum»

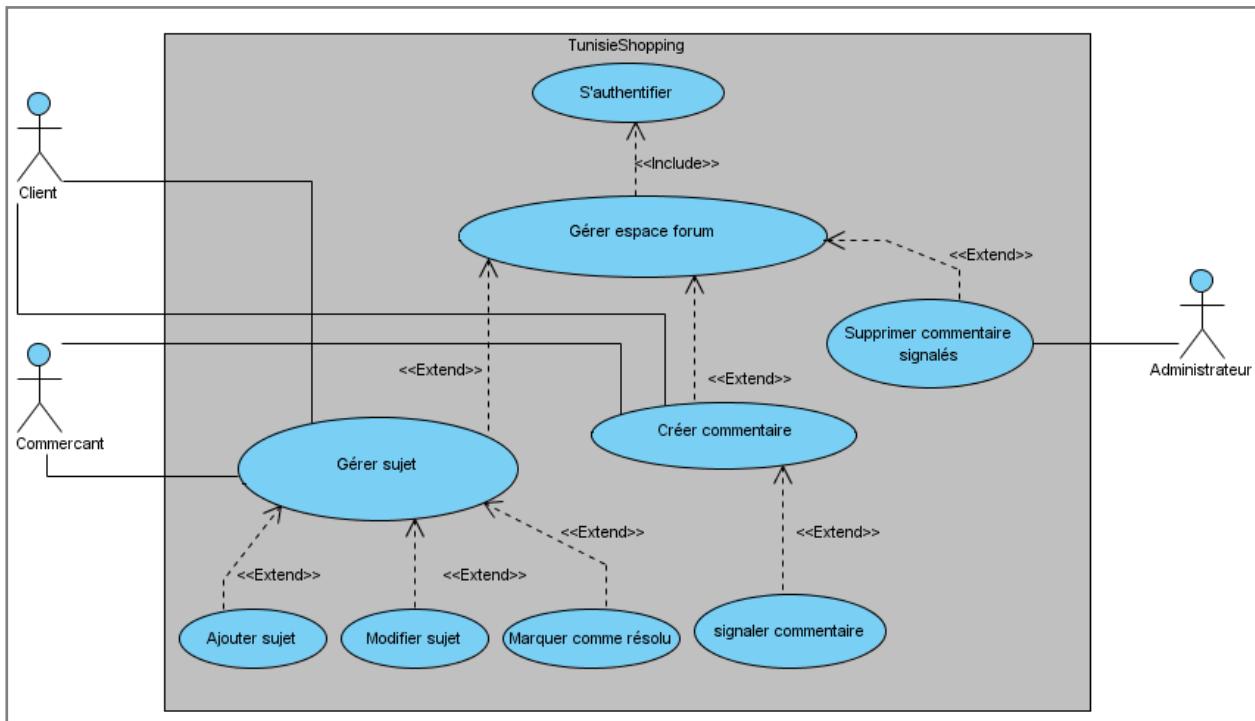


Figure 22 Diagramme de cas d'utilisation « Gérer espace forum»

7. Cas d'utilisation « Consulter l'état du commerce»

a. Description textuelle de «Consulter l'état du commerce»

Cas d'utilisation	Consulter l'état du commerce
acteurs	<ul style="list-style-type: none"> - Le commerçant
résumé	<ul style="list-style-type: none"> - Un commerçant peut consulter les ventes et les statistiques faites sur ses produits

enchaînement	<ul style="list-style-type: none"> - Pré conditions : Commerçant authentifié - Scenarios nominaux <ul style="list-style-type: none"> → Enchainement 1 : Consulter les statistiques sur les ventes → Enchainement 2 : Consulter les statistiques sur les produits - Post-conditions : Affichage des statistiques.
Exceptions	_____

Tableau 14 Description du cas d'utilisation « Consulter l'état du commerce »

b. Diagramme de cas d'utilisation «Consulter l'état du commerce»

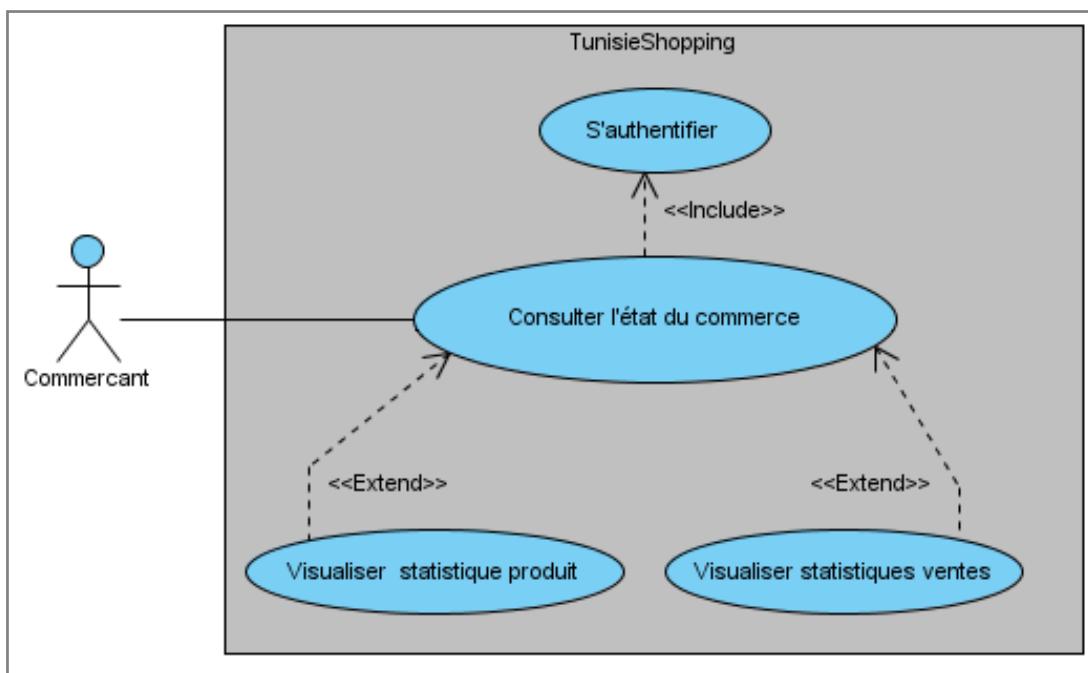


Figure 23 Diagramme de cas d'utilisation « Consulter l'état du commerce»

8. Cas d'utilisation « Consulter les notifications»

a. Description textuelle de «Consulter les notifications»

Cas d'utilisation	Consulter les notifications
acteurs	<ul style="list-style-type: none"> - Le commerçant - Le client
résumé	<ul style="list-style-type: none"> - Chaque utilisateur visualise ses propres notifications

enchaînement	<ul style="list-style-type: none"> - Pré conditions : Commerçant ou client authentifié - Scenarios nominaux <ul style="list-style-type: none"> → Enchainement 1 : Consulter notifications → Enchainement 2 : Supprimer les notifications - Post-conditions : Affichage des notifications.
Exceptions	_____

Tableau 15 Description du cas d'utilisation « Consulter les notifications»

b. Diagramme de cas d'utilisation «Consulter les notifications»

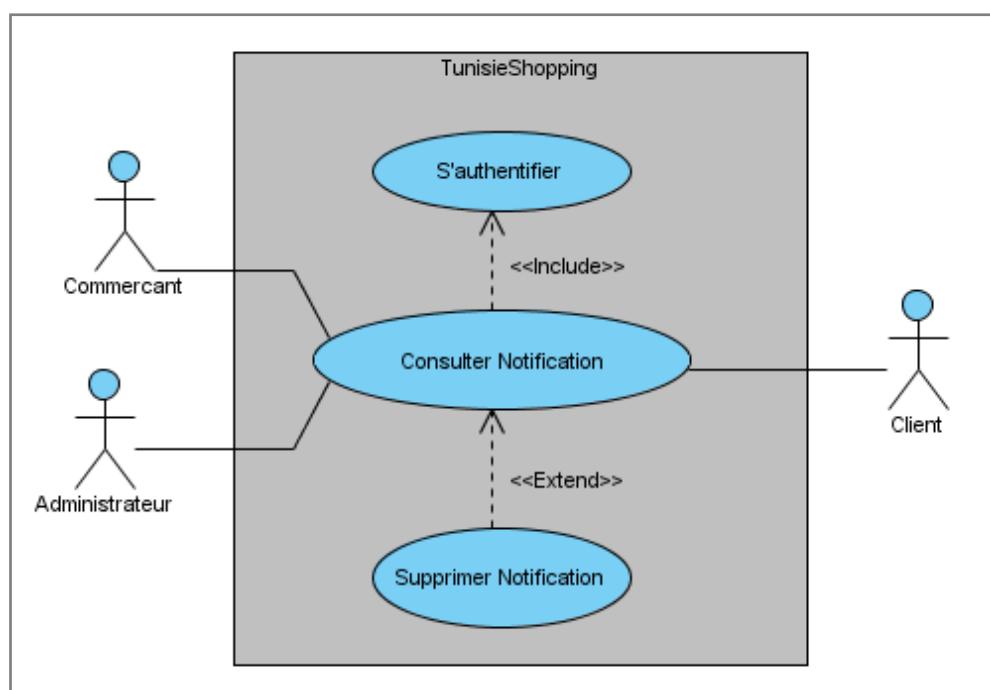


Figure 24 Diagramme de cas d'utilisation « Consulter les notifications»

9. Cas d'utilisation « Gérer les achats»

a. Description textuelle de «Gérer les achats»

Cas d'utilisation	Gérer les achats
acteurs	<ul style="list-style-type: none"> - Le client - Administrateur
résumé	<ul style="list-style-type: none"> - Chaque Client peut consulter les produits et les annonces. Il peut effectuer plusieurs actions.

enchaînement	<ul style="list-style-type: none"> - Pré conditions : Client authentifié. - Scenarios nominaux : <ul style="list-style-type: none"> → Enchainement 1 : Rechercher produit ou commerçant → Enchainement 2 : Sélectionner les produits à acheter. → Enchainement 3 : Réaliser le paiement en ligne <ul style="list-style-type: none"> Si le code de la carte est invalide : [Exception 1 : code erroné] Si le montant est insuffisant : [Exception 2 : Erreur paiement] → Enchainement 4 : Noter les produits → Enchainement 5 : Signaler produit → Enchainement 6 : Le client peut consulter l'historique de ses achats. Le client peut signaler un produit spécifiant la raison, par suite l'administrateur intervient. - Post-conditions : Produit acheté, ou bien juste consulté.
Exceptions	<ul style="list-style-type: none"> - [Exception 1 : code erroné] : le système affiche un message d'erreur exprimant que le code de la carte de paiement est erroné. - [Exception 2 : Erreur paiement] : Le système affiche un message d'erreur indiquant un problème de paiement.

Tableau 16 Description du cas d'utilisation « Gérer les achats»

b. Diagramme de cas d'utilisation «Gérer les achats»

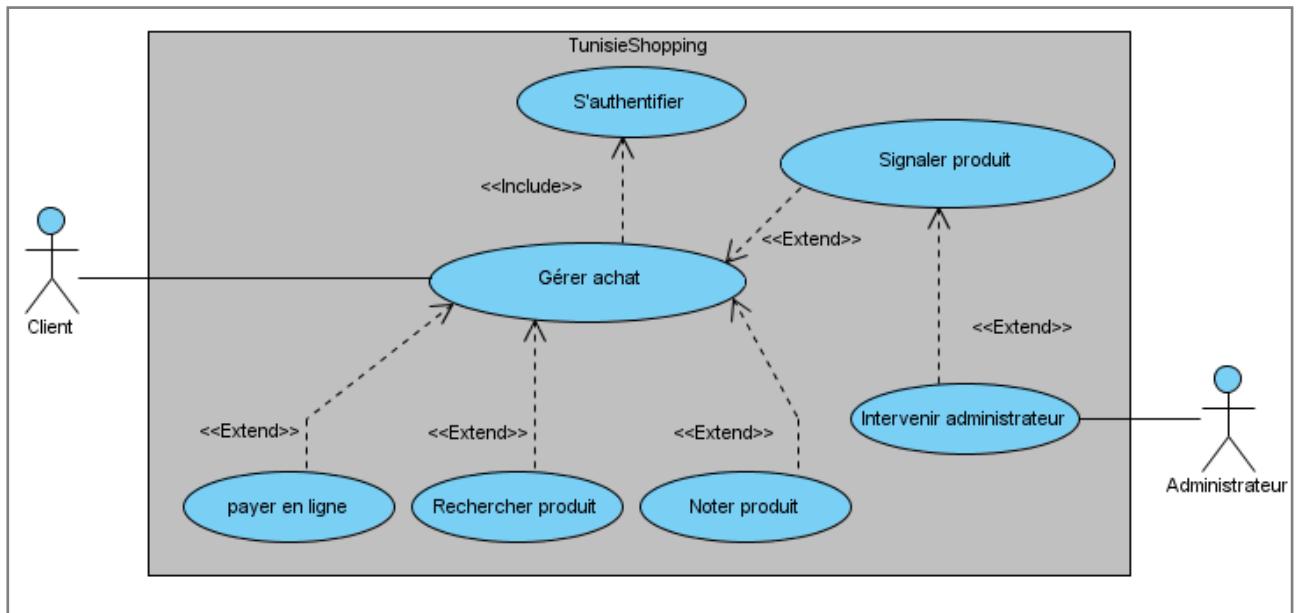


Figure 25 Diagramme de cas d'utilisation « Gérer les achats»

10. Cas d'utilisation « Gérer utilisateurs»

a. Description textuelle de «Gérer utilisateurs»

Cas d'utilisation	Gérer utilisateurs
acteurs	- L'administrateur
résumé	- L'administrateur consulte les profils des utilisateurs, il peut aussi supprimer ou bien désactiver un commerçant.
enchaînement	<ul style="list-style-type: none"> - Pré conditions : _____ - Scenarios nominaux : <ul style="list-style-type: none"> → Enchainement 1 : Consulter les profils → Enchainement 2 : Activer/Désactiver profil → Enchainement 3 : Supprimer profil - Post-conditions : Suppression ou de désactivation du profil
Exceptions	_____

Tableau 17 Description du cas d'utilisation « Gérer utilisateurs»

b. Diagramme de cas d'utilisation «Gérer utilisateurs»

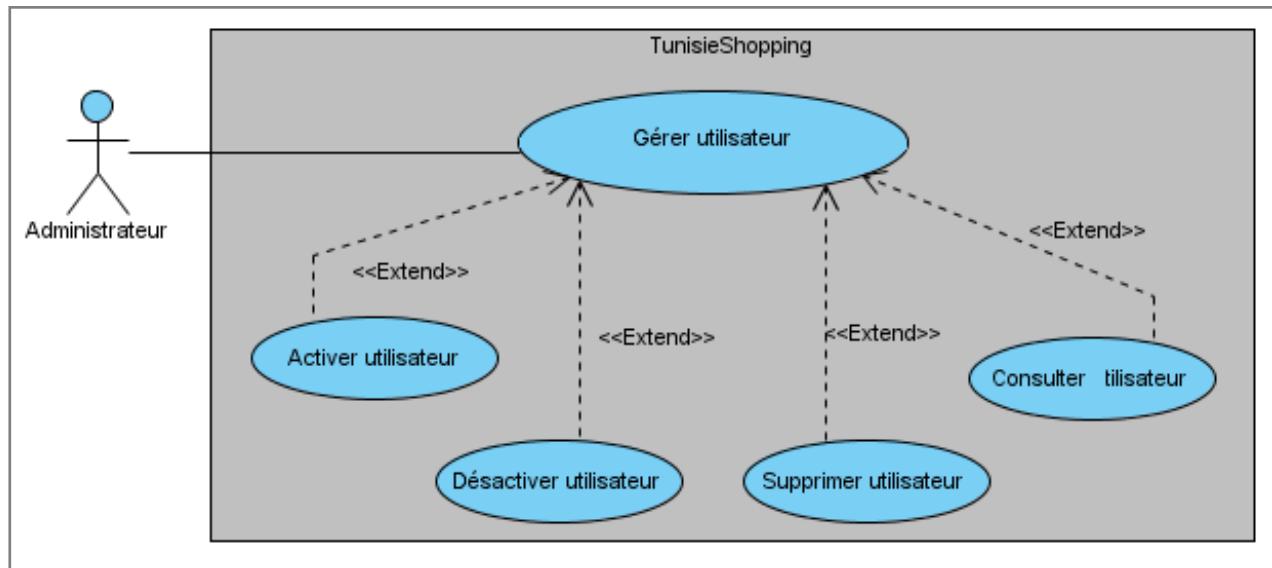


Figure 26 Diagramme de cas d'utilisation « Gérer utilisateurs»

11. Cas d'utilisation « Superviser plateforme»

a. Description textuelle de «Superviser plateforme»

Cas d'utilisation	Superviser la plateforme
acteurs	- L'administrateur
résumé	- Cette tache concerne les Analyses et les rapports concernant la plateforme (nombre des inscriptions, nombre de visiteurs, taux des clients fidèles...)
enchaînement	<ul style="list-style-type: none"> - Pré conditions : _____ - Scenarios nominaux : <ul style="list-style-type: none"> → Enchainement 1 : Consulter les statistiques - Post-conditions : Figure de statistique affichée
Exceptions	_____

Tableau 18 Description du cas d'utilisation « Superviser plateforme»

b. Diagramme de cas d'utilisation «Superviser plateforme»

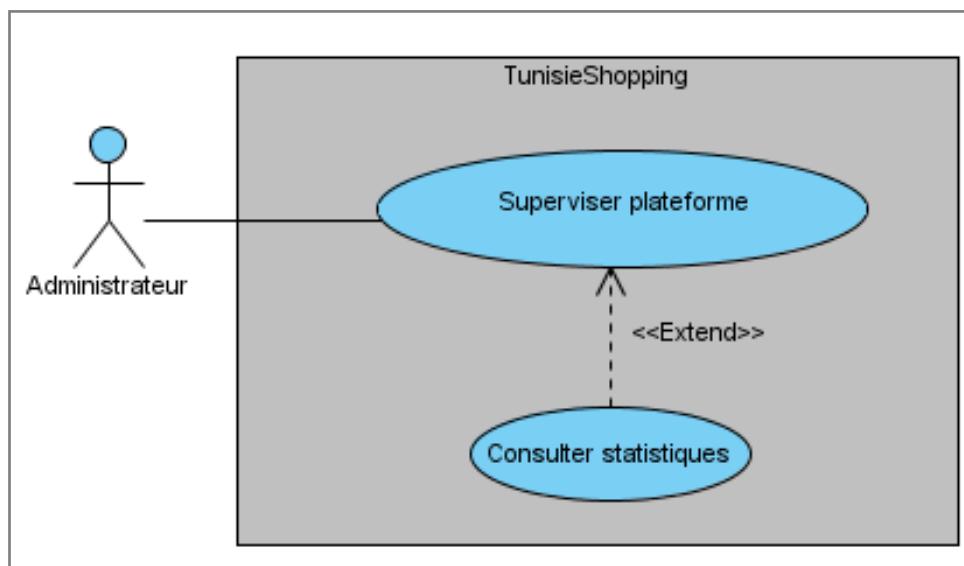


Figure 27 Diagramme de cas d'utilisation « Superviser plateforme»

12. Cas d'utilisation « gérer paiement»

a. Description textuelle de «gérer paiement»

Cas d'utilisation	Gestion de paiement
acteurs	<ul style="list-style-type: none"> - Le commerçant - Administrateur
résumé	<ul style="list-style-type: none"> - Cette tache constitue les étapes de la mise en œuvre de la procédure du paiement en ligne pour un commerçant.
enchaînement	<ul style="list-style-type: none"> - Pré conditions : Commerçant authentifié. - Scenarios nominaux : <ul style="list-style-type: none"> → Enchainement 1 : Demande d'activation du paiement → Enchainement 2 : Activation du paiement L'administrateur active le paiement. - Post-conditions : Procédure de paiement activé
Exceptions	_____

Tableau 19 Description du cas d'utilisation « Gérer paiement»

b. Diagramme de cas d'utilisation «Gérer paiement»

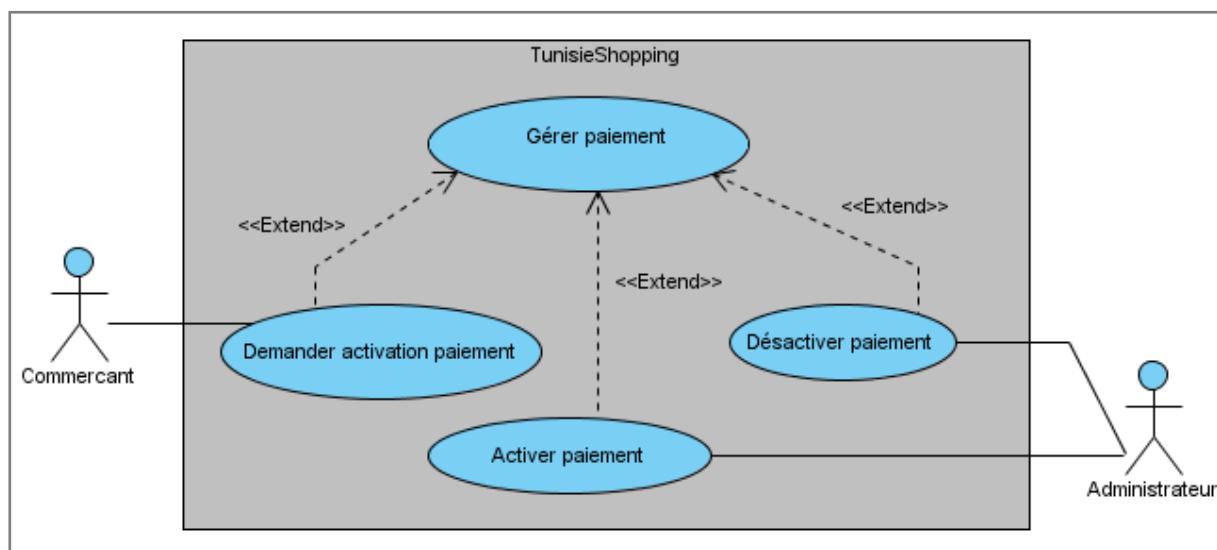


Figure 28 Diagramme de cas d'utilisation « Gérer paiement»

13. Cas d'utilisation « Effectuer paiement»

a. Description textuelle de «Effectuer paiement»

Cas d'utilisation	Effectuer le paiement
acteurs	<ul style="list-style-type: none"> - Module paiement - Le client
résumé	<ul style="list-style-type: none"> - Cette tache constitue les étapes du paiement en ligne par le client.
enchaînement	<ul style="list-style-type: none"> - Pré conditions : Client authentifié. - Scenarios nominaux : <ul style="list-style-type: none"> → Enchainement 1 : Choisir les articles à acheter → Enchainement 2 : Remplir le formulaire d'achat → Enchainement 3 : Recevoir un mail contenant la commande - Post-conditions : Commande envoyée par mail
Exceptions	_____

Tableau 20 Description du cas d'utilisation « Effectuer paiement»

b. Diagramme de cas d'utilisation «Effectuer paiement»

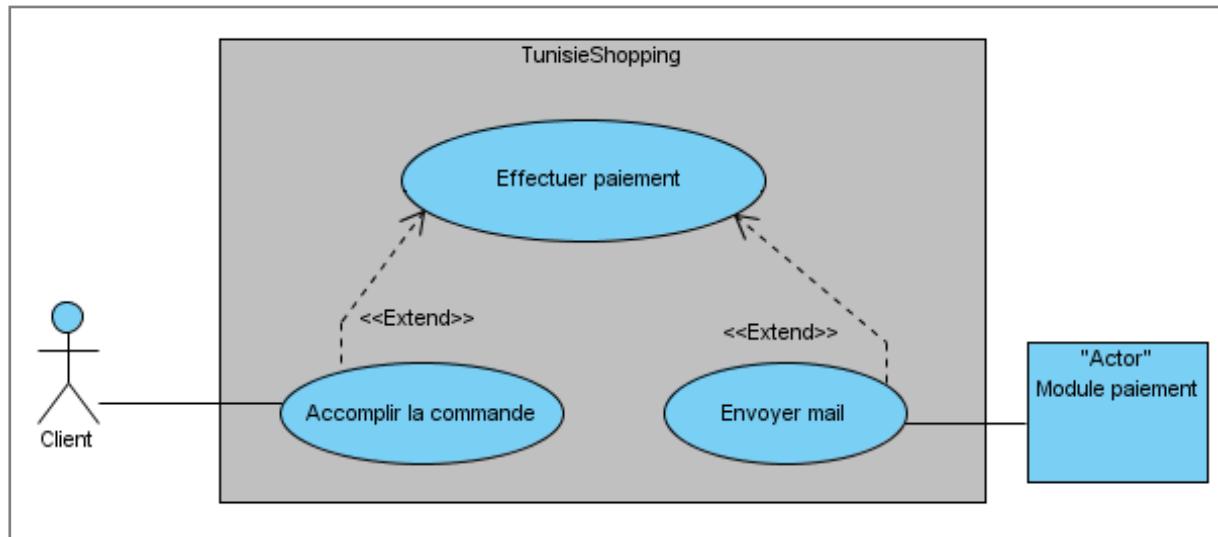


Figure 29 Diagramme de cas d'utilisation « Effectuer paiement»

14. Cas d'utilisation « Gérer notifications»

a. Description textuelle de «Gérer notifications»

Cas d'utilisation	Gérer notifications
acteurs	<ul style="list-style-type: none"> - Module notification - Le client - Le commerçant - L'administrateur
résumé	<ul style="list-style-type: none"> - La gestion des notifications pour la plateforme.
enchaînement	<ul style="list-style-type: none"> - Pré conditions : Client ou commerçant authentifié. - Scenarios nominaux <ul style="list-style-type: none"> → Enchainement 1 : Une action se produit (paiement en ligne, les commentaires et sujet dans le forum, envoi d'un message). → Enchainement 2 : Notification envoyée par le module de notification. - Post-conditions : Notification
Exceptions	_____

Tableau 21 Description du cas d'utilisation « Gérer notifications»

b. Diagramme de «Gérer notifications»

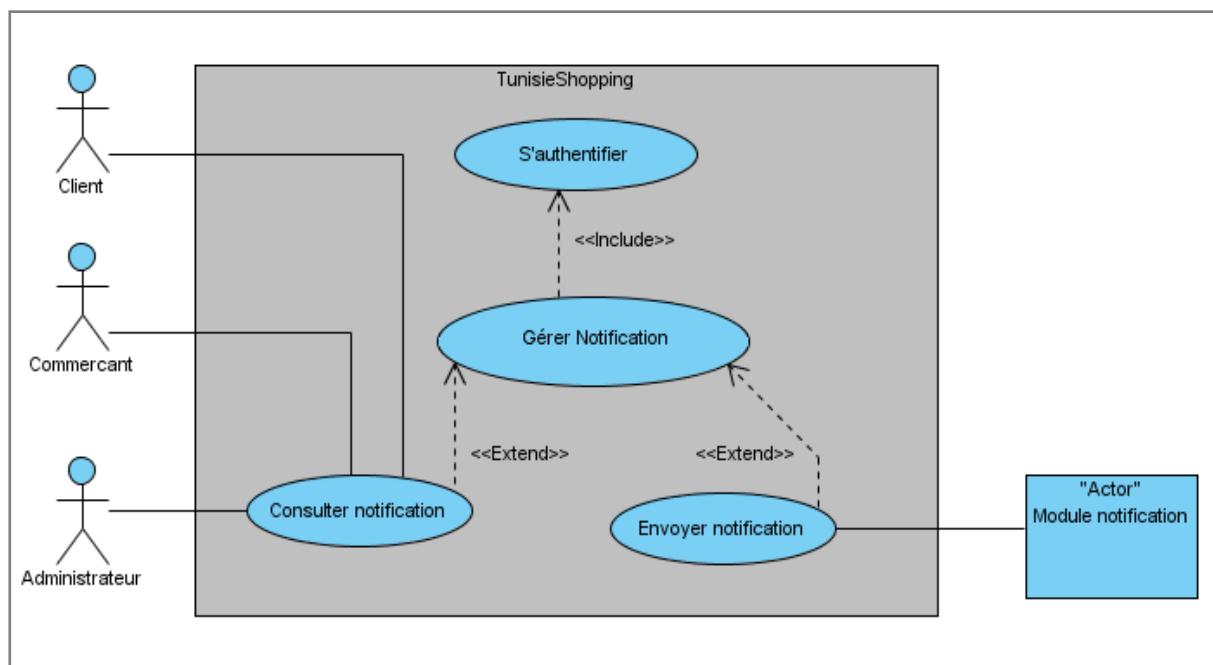


Figure 30 Diagramme de cas d'utilisation « Gérer notifications»

15. Diagramme de cas d'utilisation général pour l'application mobile

Au niveau de cette partie nous présenterons les différentes fonctionnalités que nous réaliserons pour le volet mobile de notre projet.

Les acteurs ayant accès aux fonctionnalités offertes par l'application mobile sont le client et le commerçant.

a. Fonctionnalités offertes par l'application mobile

- Recherche produit : Chaque utilisateur a la possibilité de faire des recherches soit par mot clés ou bien en accédant directement par les catégories disponibles.
- Se connecter : Un utilisateur déjà inscrit, a la possibilité de se connecter.
- Accès au profil : Un utilisateur déjà connecté, pourra accéder à son profil.
- Accès Forum : En accédant à l'application mobile, l'utilisateur aura accès au forum.
- Ajouter produit au panier : Nous offrant la possibilité d'ajouter produit au panier à partir de l'application mobile.
- Suivre achats : Un client peut à, partir de son téléphone portable, consulter la liste des achats qu'il a fait.

b. Diagramme de cas d'utilisation pour l'application mobile

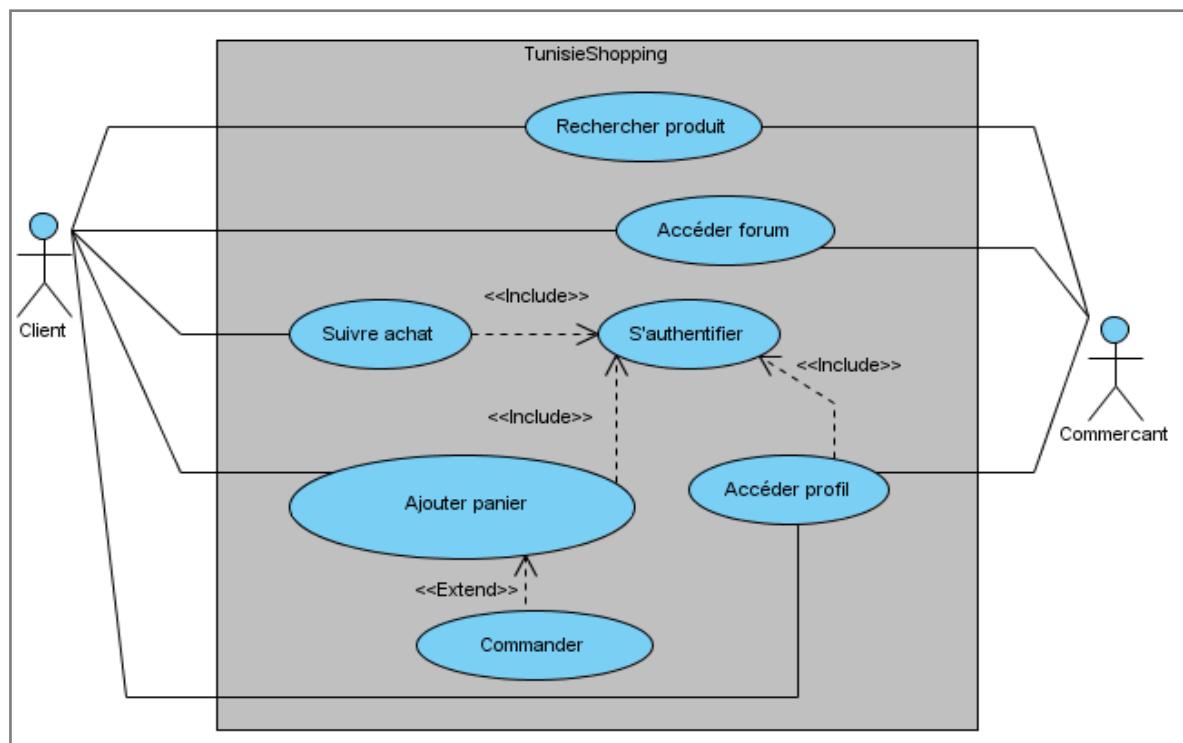


Figure 31 Diagramme cas d'utilisation pour l'application mobile

III. Structuration des cas d'utilisations dans des packages

Cette phase est utile lorsqu'il y a de nombreux cas d'utilisation, donc nous les regrouperons en package selon leur thématique. C'est une préparation à la prochaine phase qui sera le découpage en catégorie des classes.

Cas d'utilisation	Package
- S'inscrire - S'authentifier - Gérer profil	→ Gestion de l'inscription
- Gérer utilisateurs - Superviser la plateforme	→ Gestion de l'administration
- Gérer annonces - Gérer produits - Consulter l'état du commerce	→ Gestion du commerce
- Consulter messages - Gérer espace forum - Consulter les notifications - Gestion des notifications	→ Gestion de l'interaction
- Gérer les achats - Gérer paiement - Effectuer paiement	→ Gestion des achats

Tableau 22 Organisation des cas d'utilisation en package

IV. Identification des classes candidates

Cette phase sert à dégager les différentes classes pour se préparer à trouver celles du futur modèle statique d'analyse. L'identification de ces classes est basée sur les noms communs importants dans les descriptions textuelles des cas d'utilisation, une fois spécifiés nous définirons les responsabilités de chacune et ensuite nous formaliserons les diagrammes statiques.

1. Package gestion de l'inscription :

→ (S'inscrire, S'authentifier, Gérer profil)

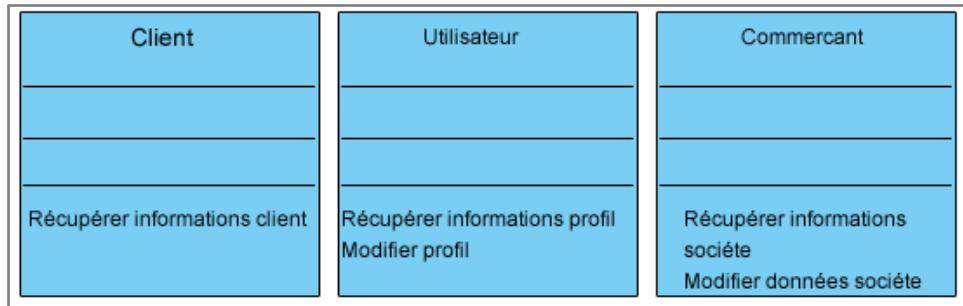


Figure 32 Classe candidate « gestion inscription »

2. Package gestion de l'administration

→ Gérer utilisateurs, superviser plateforme

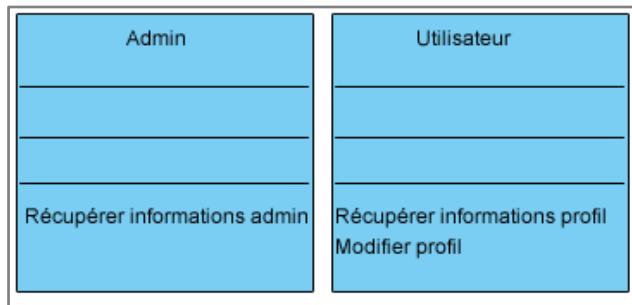


Figure 33 Classe candidate « gestion administration»

3. Package Gestion du commerce

→ Gérer annonces, Gérer produits, Consulter l'état du commerce

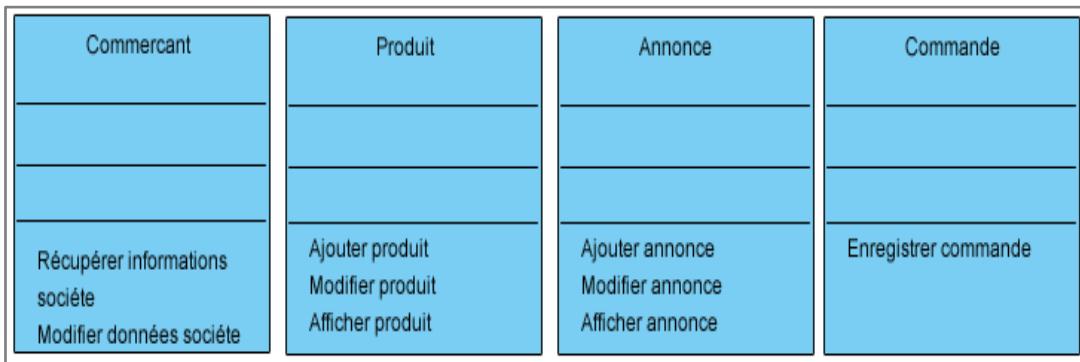


Figure 34 Classe candidate « gestion du commerce »

4. Package Gestion de l'interaction:

→ Gérer espace forum, Consulter les notifications, Gestion des notifications

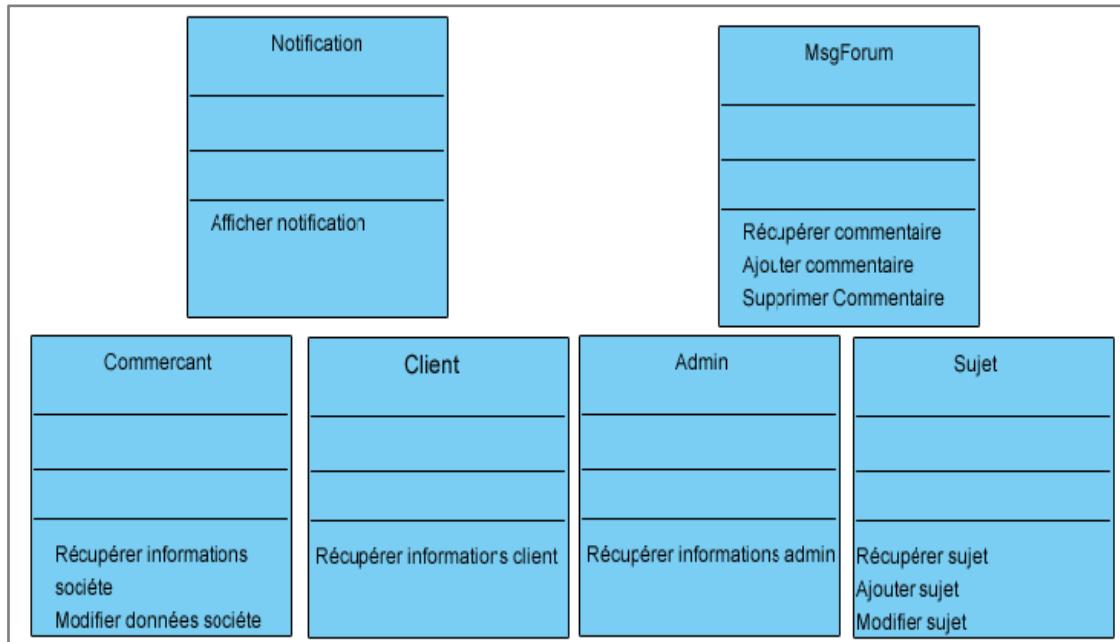


Figure 35 Classe candidate « gestion de l'interaction »

5. Package Gestion des achats

→ Gérer les achats, Gérer paiement, Effectuer paiement

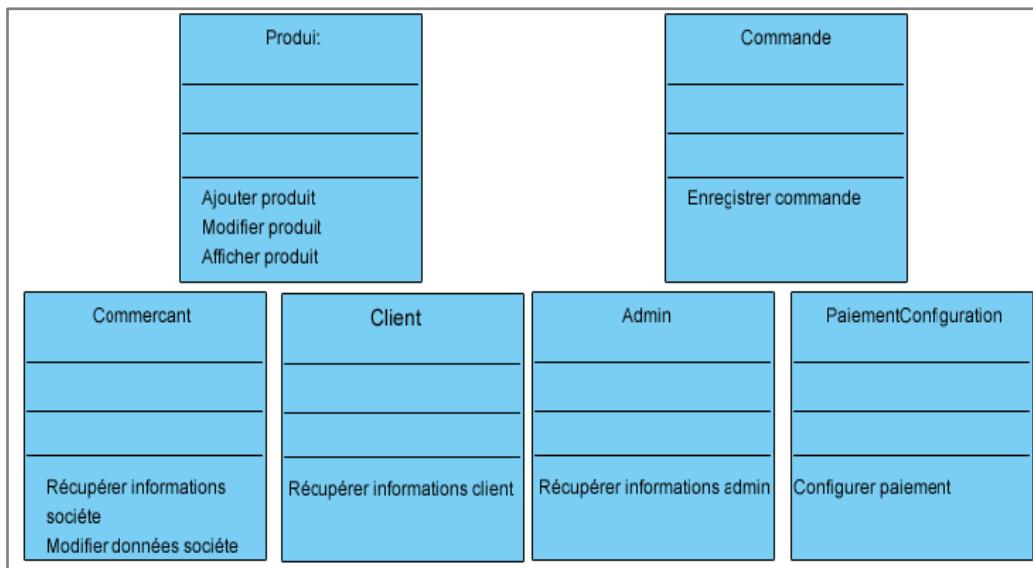


Figure 36 Classe candidate « gestion des achats »

Dans ce qui suit nous formaliserons ces concepts sous forme de classes et d'associations rassemblées dans des diagrammes statiques préliminaires que nous appelons diagrammes de classes participantes.

6. Diagramme des classes participantes

a. Classes participantes « S'inscrire », « S'authentifier » et « Gérer profil »

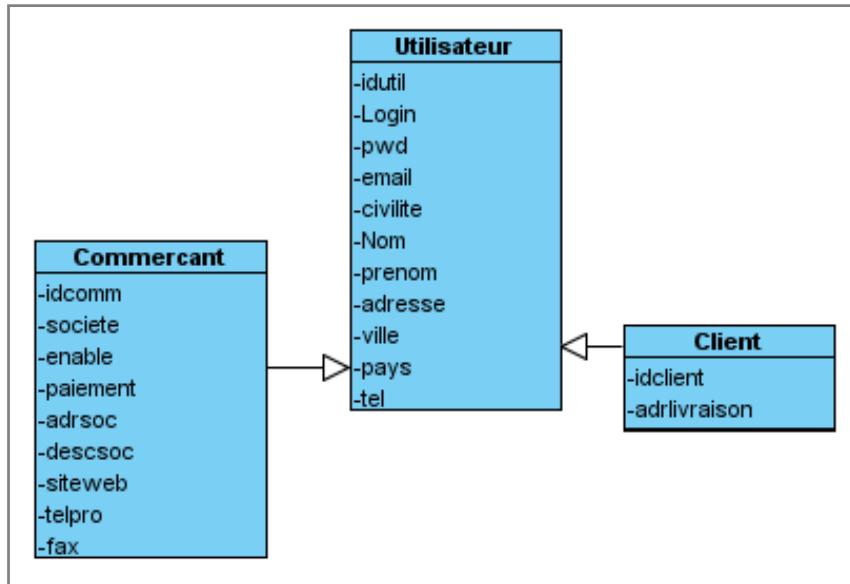


Figure 37 Diagramme des classes participantes « S'inscrire, S'authentifier et Gérer profil»

b. Classes participantes « Gérer annonces »

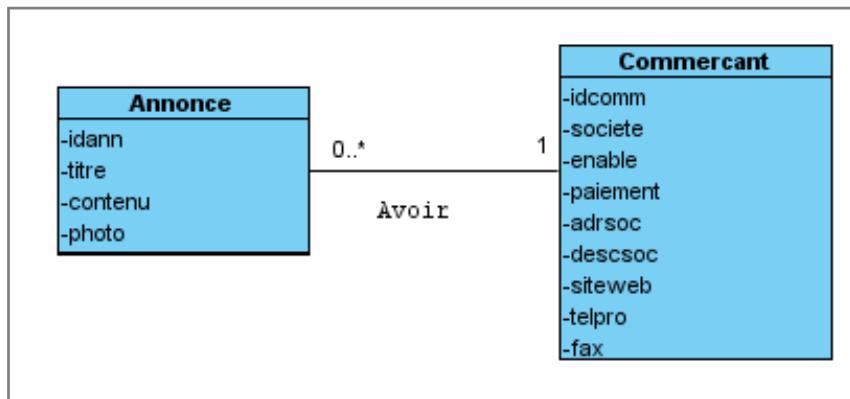


Figure 38 Diagramme des classes participantes « Gérer annonces»

c. Classes participantes « Gérer produit »

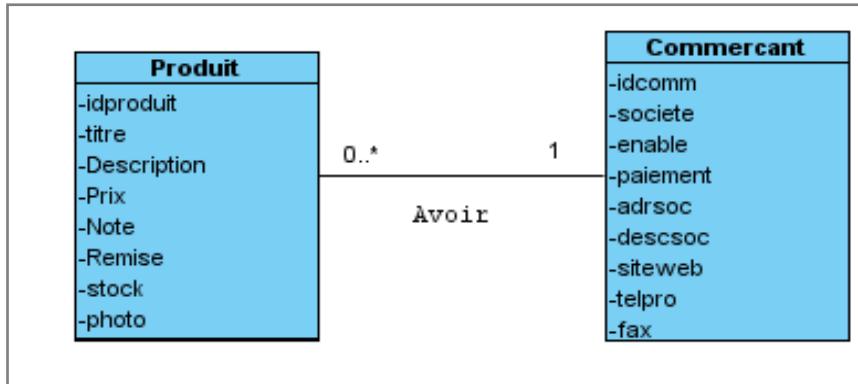


Figure 39 Diagramme des classes participantes « Gérer produit»

d. Classes participantes « Consulter l'état du commerce »

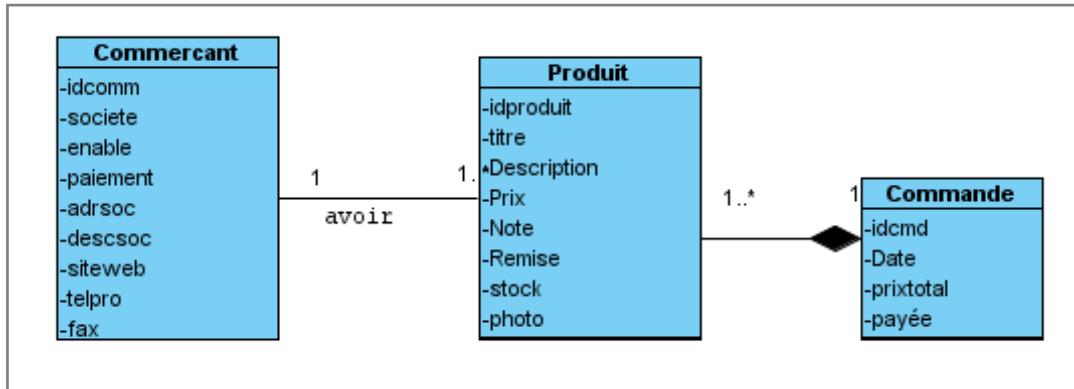


Figure 40 Diagramme des classes participantes « Consulter l'état du commerce»

e. Classes participantes « Consulter messages »

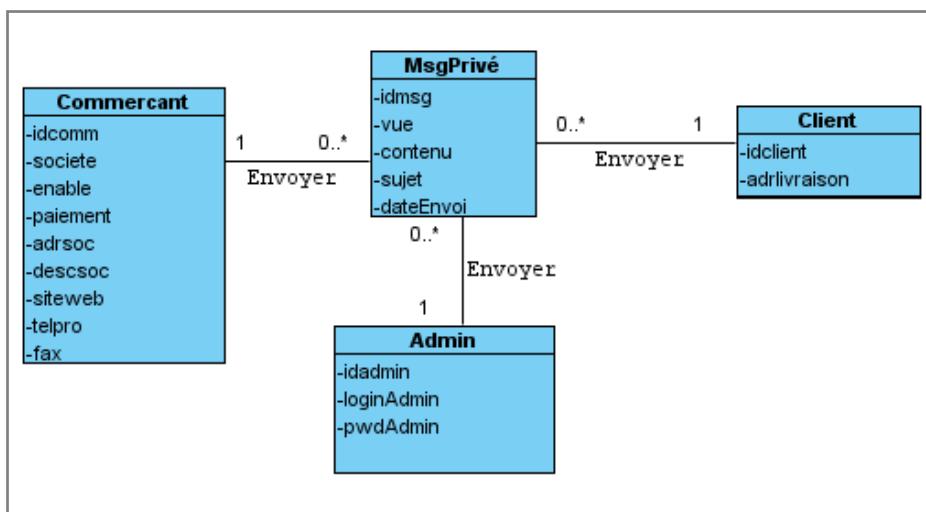


Figure 41 Diagramme des classes participantes « Consulter messages»

f. Classes participantes « Gérer utilisateur » et « superviser plateforme »

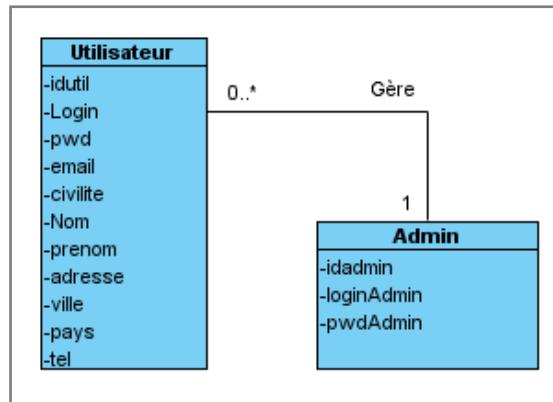


Figure 42 Diagramme des classes participantes « Gérer utilisateur et superviser plateforme»

g. Classes participantes « Gérer espace forum »

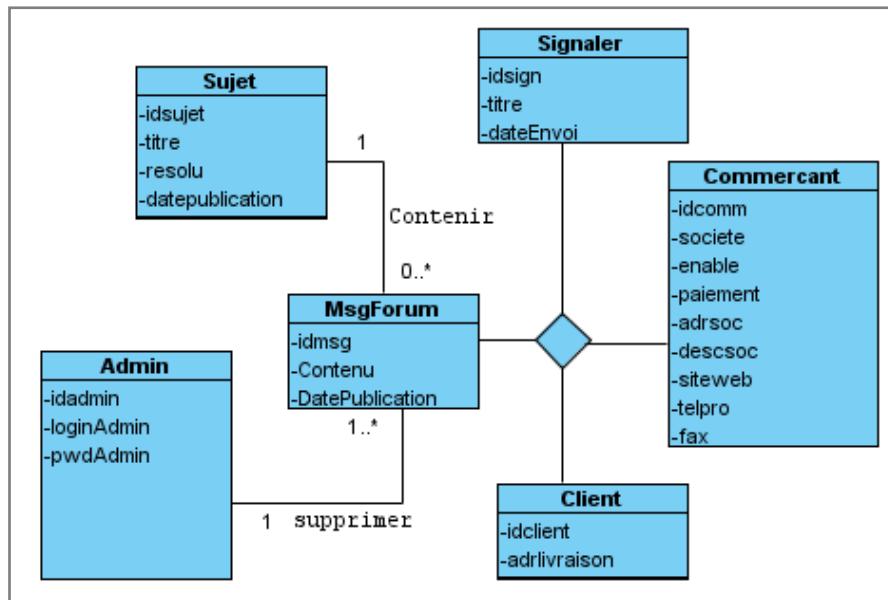


Figure 43 Diagramme des classes participantes « Gérer espace forum»

h. Classes participantes « Gérer notification » et « Consulter notification »

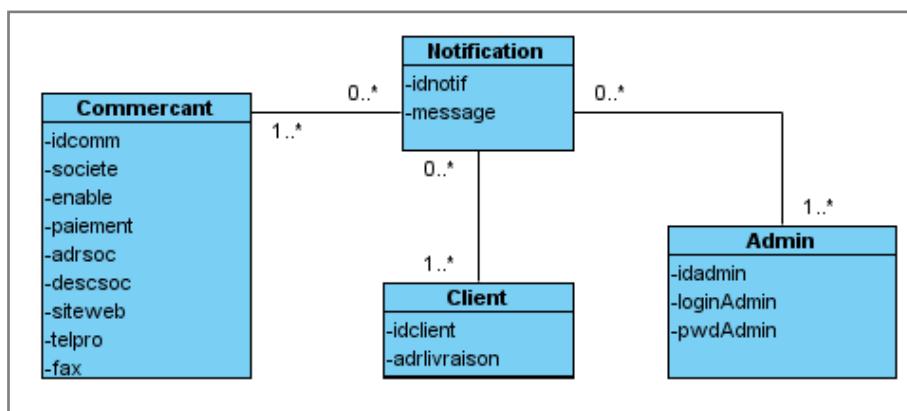


Figure 44 Diagramme des classes participantes « Gérer notification et consulter notification»

i. Classes participantes « Gérer paiement » et « Effectuer paiement »

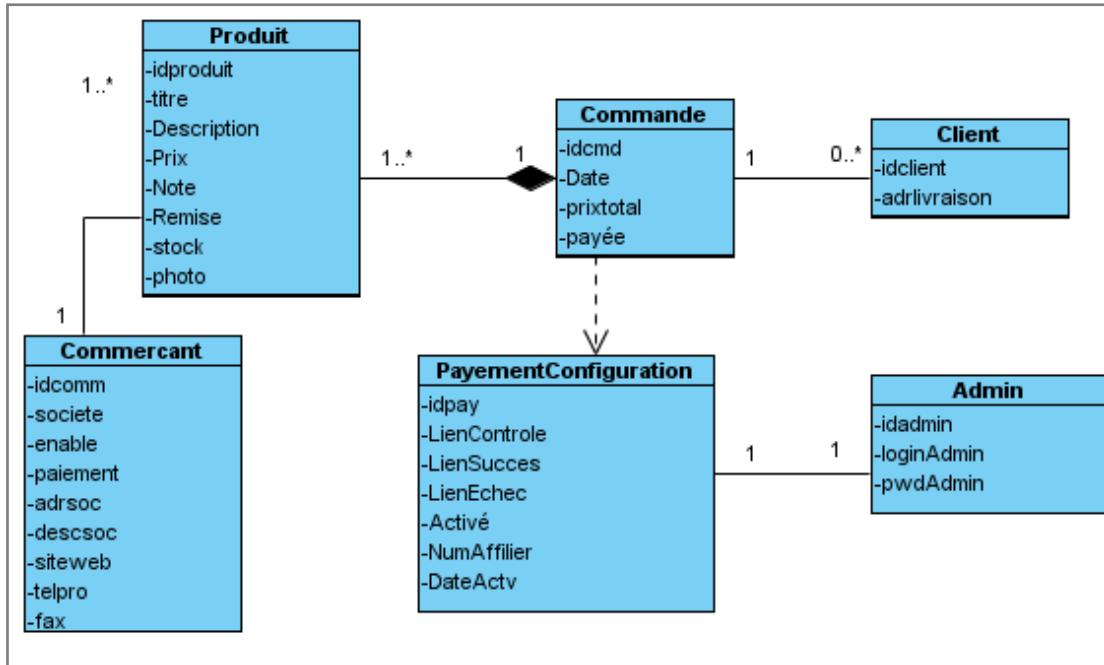


Figure 45 Diagramme des classes participantes « gérer paiement et effectuer paiement»

V. Capture des besoins techniques

Nous allons aborder les différentes étapes pour la mise en place de ce présent travail. Pour cela, nous allons commencer par présenter notre environnement de travail en termes de technologies et outils utilisés, passant par les langages de programmation adoptés et une dernière partie concernant la sécurisation de notre application.

Nous avons choisi pour le développement de l'application web le Framework CodeIgniter qui est basé sur le langage de programmation php5 coté serveur, coté client nous allons utiliser le langage HTML5.

Le langage que nous avons adopté pour le développement de la couche traitement c'est le php5. Il est utilisé principalement en tant que langage de script côté serveur, ce qui veut dire que c'est le serveur (la machine qui héberge la page Web en question) qui va interpréter le code PHP et générer du code (constitué généralement d'XHTML ou d'HTML, de CSS, et parfois de JavaScript) qui pourra être interprété par un navigateur. La version 5 permettra d'utiliser les espaces de noms, un élément fondamental de l'élaboration d'extensions, de bibliothèques et de Framework structurés.

1. Environnement de développement

Dans le développement de notre application, nous veillerons à utiliser les outils suivants et à respecter les principes suivants pour le développement de l'interface, vu l'importance de cette aspect dans les applications Web.

a. Interface web

- Les langages HTML5/CSS3 :

Nous avons choisi d'utiliser le Langage HTML5 qui est une évolution du HTML 4 qui a pour objectif de faciliter et d'uniformiser le travail des développeurs et intégrateurs grâce à une série de nouveautés. Un des changements majeurs réside dans la simplification de la structure du document HTML. Pour le Design, nous avons choisi CSS3.

- Utilisation d'un Moteur de Template :

Un moteur de Template est un système permettant la séparation de la structure d'une page web (le HTML/XHTML) et le traitement (PHP). Au lieu d'inclure le PHP dans le code HTML, le moteur développé en PHP, va permettre de mettre simplement des tags au sein du code HTML.

Nous avons choisi Twig comme moteur de Template, ses avantages sont :

- Sécuriser automatiquement les variables : Le moteur de Template s'occupe de l'échappement des variables sans avoir besoin de `htmlspecialchars`, `htmlentities()` ...
- L'héritage de Template

L'héritage de Template nous évite de réécrire un code plusieurs fois. Son principe c'est d'avoir un Template père qui contient le design du site contenant un ensemble de block et des Template fils qui seront appelés à la demande pour remplir ces blocks. Voici un exemple dans la figure suivante :

```
{% extends "SdzBlogBundle::layout.html.twig" %}  
{% block title %}{{ parent() }} - Index{% endblock %}  
  
{% block body %}  
    OK, même si il est pour l'instant un peu vide, mon blog sera trop bi  
    {% endblock %}
```

Figure 46 Exemple d'héritage de Twig

- Une syntaxe propre et condensée dédiée au design (réutilisabilité) : Les intégrateurs peuvent se concentrer sur un langage et une syntaxe qui à priori ne changeront pas, contrairement à PHP qui évolue (dépréciation de fonctions par exemple). Que dire de plus sur ce point, c'est aussi une histoire de goût.
- Un code HTML beaucoup plus lisible : Ceci est indéniable, le code HTML devient beaucoup plus lisible, si le moteur de Template est maitrisé il y aura moins de chance de se perdre dans le code.

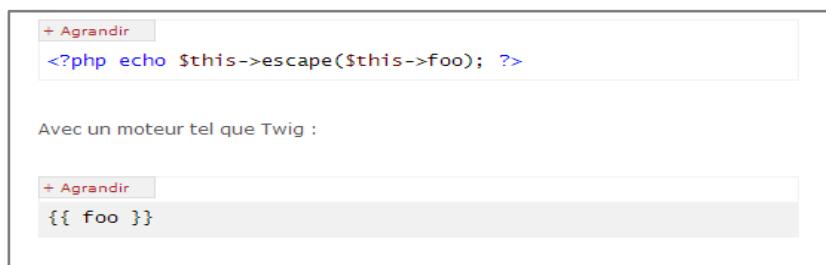


Figure 47 Comparaison entre code PHP et twig

- **Mise en application du principe « Responsive Web Design » :**

Le « responsive web design » est une façon de concevoir un site web pour que son contenu s'adapte automatiquement à la largeur et/ou à la hauteur de l'écran qui le visionne.

Pour appliquer cette notion nous avons choisi Twitter Bootstrap. C'est une collection d'outils utile à la création de sites web et 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.

Twitter Bootstrap propose plusieurs fonctionnalités :

- Une mise en page basée sur une grille de 12 colonnes.

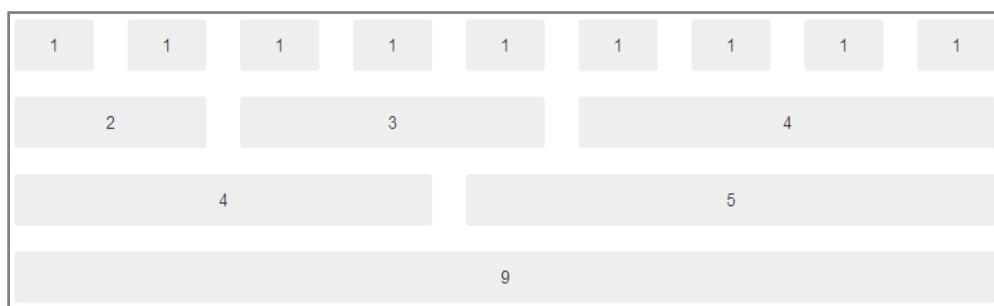


Figure 48 Grille de colonne de Twitter bootstrap

- Un **reset CSS** basé sur **Normalize.css** (il permet de remettre à zéro tous les CSS pour partir sur une base saine et compatible avec tous les navigateurs.).
- Du code fondé sur **HTML 5 et CSS 3.**
- Une bibliothèque totalement open source sous licence Apache.
- Du code qui tient compte du format d'affichage des principaux outils de navigation (responsive design) : Smartphones, tablettes...

The screenshot shows a web page titled "Responsive design". Under the heading "Enabling responsive features", there is a note: "Turn on responsive CSS in your project by including the proper meta tag and additional stylesheet within the <head> of your document. If you've compiled Bootstrap from the Customize page, you need only include the meta tag." Below this is a code block:

```
1. <meta name="viewport" content="width=device-width, initial-scale=1.0">
2. <link href="assets/css/bootstrap-responsive.css" rel="stylesheet">
```

A "Heads up!" note states: "Bootstrap doesn't include responsive features by default at this time as not everything needs to be responsive. Instead of encouraging developers to remove this feature, we figure it best to enable it as needed." To the right of this text are icons representing a desktop computer, a smartphone, and a tablet.

Under the heading "About responsive Bootstrap", it says: "Media queries allow for custom CSS based on a number of conditions—ratios, widths, display type, etc—but usually focuses around min-width and max-width." A bulleted list follows:

- Modify the width of column in our grid
- Stack elements instead of float wherever necessary
- Resize headings and text to be more appropriate for devices

At the bottom, a note says: "Use media queries responsibly and only as a start to your mobile audiences. For larger projects, do consider dedicated code bases and not layers of media queries."

Figure 49 Responsive web design de Twitter bootstrap

- Des **plugins jquery** de qualité.
- Un résultat **cross-browser**(c'est la possibilité pour toute application web, sous format HTML ou programmée avec un langage de script s'exécutant côté client de supporter plusieurs navigateurs web), en prenant même en compte IE7, donc une garantie de compatibilité maximale.

- **Utilisation de l'API Facebook :**

Dans le but de rendre la plateforme plus sociale nous utiliserons des API fournies par facebook appelés aussi Graph API. Facebook offre plusieurs fonctionnalités pour les applications web, il part du principe qu'il est possible de tout représenter sous formes d'objets et de connexions.

- Objets : personnes, photos, événements, pages, groupes, messages, vidéos...
- Connexion : amitié, contenu partagé, "like"

L'API nous permettra d'accéder à ces différents objets pour ensuite utiliser les connexions et avoir accès à d'autres objets. Pour cela, Facebook propose un Web Service.

Les requêtes se feront à l'aide d'une url et les réponses seront retournées, au choix, au format JSON ou XML. Au niveau de la présente application, nous nous servirons de la Login Facebook et du bouton partager qui seront utile pour notre cas.

- **Utilisation de bibliothèques Javascript :**

JQuery, Modernizer, JSON, et AJAX sont les technologies que nous utiliserons dans la couche interface pour un meilleur contrôle, pour plus de rapidité et une meilleure compatibilité de notre application.

- **Jquery** : C'est une bibliothèque JavaScript libre qui porte sur l'interaction entre JavaScript (comprenant Ajax) et HTML, et a pour but de simplifier des commandes communes de JavaScript. Vue que nous avons des différents cas dont certaine résultats dépendent des actions utilisateurs d'où la nécessité d'une interaction avec HTML ce qui explique le besoin d'utiliser jquery.
- **Modernizer** : C'est une bibliothèque JavaScript conçue pour détecter des fonctionnalités spécifiques de HTML et CSS dans les navigateurs. Puisqu'il est inutile d'embarquer un fichier complet de détection dans tous les sites web, un script sur mesure peut être construit en piochant parmi les fonctionnalités détectables. Il sera par exemple possible de savoir si le navigateur supporte les transformations CSS 3D, la vidéo HTML5, Canvas ou SVG, et de prévoir une alternative le cas échéant. Nous avons eu recours à ce principe à fin de garantir que l'application sera accessible de n'importe quel navigateur même s'il ne supporte pas le CSS3. Nous utiliserons la récente technologie CSS3 pour certaines décorations d'éléments de page, ces dernières pourraient également être traitées sans CSS3 à l'aide de méthodes plus longues et coûteuses en performances. La nous pourrons toucher l'utilité de cette bibliothèque JavaScript.

- **Json** : JavaScript Object Notation, est une forme d'écriture de données en JavaScript. Son avantage est de fournir un support pour une écriture simple et légère au format texte, relativement compréhensible par les développeurs JavaScript, mais aussi d'être nativement interprété contrairement au XML qui fait appel à de l'analyse syntaxique et parfois à DOM/XSLT pour accéder à sa structure et à son contenu. Il s'agit donc d'une arborescence de données, inspirée de XML mais dont l'emploi en JavaScript est plus aisé et plus performant, à partir du moment où on en connaît la structure.
- **Ajax** : c'est un concept de programmation Web reposant sur plusieurs technologies comme le JavaScript et le XML – d'où le nom AJAX. Il permet de faire communiquer une page Web avec un serveur Web sans occasionner le rechargeement de la page. C'est la raison pour laquelle JavaScript est utilisé, car c'est lui qui va se charger d'établir la connexion entre la page Web et le serveur.

- **Utilisation d'un mécanisme de gestion de version : Github**

C'est un service web d'hébergement et de gestion de développement de logiciels, utilisant le programme Git, Git est un logiciel de gestion de versions décentralisé. Le présent travail est réalisé en binôme d'où la meilleure façon de faciliter l'intégration des tâches, et par suite gagner en terme de temps, c'est d'utiliser un outil de gestion de versions.

b. Interface Mobile

Nous avons choisi d'utiliser JQuery Mobile et l'émulateur Android pour développer le volet mobile de notre application, partiellement généré via PhoneGap.

- **Jquery mobile**

C'est un framework qui présente un système d'interface utilisateur unifié qui fonctionne parfaitement sur tous les appareils mobiles les plus populaires et qui est construit sur les fondations solides de « jQuery » et « jQuery UI ». Axé sur des fonctionnalités riches, mais basé sur un code léger qui construit une amélioration progressive avec un système de thème et l'outil **ThemeRoller**⁵. Il inclut un système de navigation Ajax qui apporte

⁵ Permet de créer un thème, puis de télécharger un fichier CSS personnalisé qui est prêt à être utilisé

les transitions animées des pages et un ensemble de widgets UI. La Figure 50 présente le Template initial utilisé pour Jquery.

```
<div data-role="page">  Enveloppe de la page  
  
    <div data-role="header">  Entête de la page  
        <h1>Mon titre</h1>  
    </div><!-- /header -->  
  
    <div data-role="content">  Contenu de la page  
        <p>Hello world</p>  
    </div><!-- /content -->  
  
</div><!-- /page -->
```

Figure 50 Syntaxe de JQuery mobile

- **Emulateur Android**

Un émulateur est un outil qui permet de faire fonctionner, dans la plupart des cas, une console de jeux fixe ou portable à partir d'une autre plateforme. Une possibilité intéressante qui est grandement répandu sur Android, notamment grâce à son ouverture auprès de la communauté, qui offre le moyen de faire à peu près ce que l'on souhaite, car c'est un système d'exploitation mobile ultra-modulable.

2. Mécanisme de Sécurisation

L'aspect sécurité est primordial dans notre application. Dans le but de rassurer les cyberconsommateurs et garantir la crédibilité de la plateforme nous avons appliqué des différentes règles de sécurité que nous avons identifiée dans le chapitre étude technique afin de sécuriser les données de nos utilisateurs.

a. Sécurité de l'application

CodeIgniter, le Framework de notre application, propose un ensemble de règles de sécurité. Une des failles dans un site accessible par internet c'est la possibilité d'afficher le contenu et l'arborescence des dossiers de l'application et par suite un pirate peut deviner la technologie utilisée. Pour éviter ce problème on trouve sous chaque répertoire de notre projet un fichier « index.html » qui empêche la possibilité de lister le contenu des répertoires.

Il faut aussi empêcher l'accès direct aux fichiers php de l'application pour cela nous avons défini au niveau de notre fichier d'entrée de l'application c'est le fichier « index.php » le chemin d'accès au dossier système avec le code de la Figure 51 suivant :

```
// Path to the system folder
define('BASEPATH', str_replace("\\", "/", $system_path));
```

Figure 51 Configuration du path PHP de Codeigniter

Au niveau de tous les fichiers d'extension php de l'application on ajoute le code suivant au début :

```
<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');
```

Figure 52 Path PHP dans les contrôleurs Codeigniter

Le système des Url utilisé par Codeigniter peut éviter une attaque de type interprétation des url puisque les url utilisés sont de la forme Contrôleur/méthode. Nous avons désactivé aussi l'affichage de l'extension des fichiers. Or le format des url Codeigniter par défaut ne cache pas bien la technologie utilisée comme la montre la Figure 53 suivante :

```
http://www.monsite.com/index.com/controleur/methode
```

Figure 53 Routage de Codeigniter 1/2

Pour éviter ce problème nous avons mis en place le mécanisme de réécriture d'url d'Apache permettant de faire disparaître le nom du script index.php :

```
http://www.monsite.com/controleur/methode
```

Figure 54 Routage de Codeigniter 2/2

Le paramétrage apache nécessaire pour activer la réécriture des url se fait au niveau du fichier « .htaccess » se trouvant à la racine de notre application, son contenu est le suivant :

```
RewriteEngine On
RewriteCond $1 !^(index\.php|static|uploads|assets|js|img|css)
RewriteRule ^(.*)$ index.php/$1 [L]
```

Figure 55 Fichier .htaccess de Codeigniter

Pour contourner au mauvais contrôle de données entrées par les utilisateurs, nous avons contrôlé les données saisis par ces derniers au niveau des formulaires et champ de saisies, en limitant la longueur des données, validation des types de données (date, chaîne et nombre)...

Une autre menace c'est l'injection SQL, nous avons appliqué plusieurs méthodes pour filtrer et échapper les données saisies par l'utilisateur. Nous avons utilisé la classe « input » qui fournit deux buts essentiels, le premier c'est le prétraitement des données entrées et le deuxième c'est qu'elle fournit des différentes classes pour récupérer les données d'entrée et effectue le prétraitement. Les données passées par le Get/Post /Cookie seront filtrés en permettant seulement les caractères alphanumériques et quelques autres caractères, nous avons aussi le filtre XSS que nous pouvons activer de façon globale au niveau de fichier « config.php ».

Au cours de la présente application nous avons utilisé la classe active record offerte par CodeIgniter pour toutes les requêtes, en utilisant l'active record nous n'avons pas à nous occuper du SGBD utilisé et par suite l'application aura une meilleure portabilité.

En plus de la portabilité, les requêtes « **active record** » sont automatiquement protégées et les valeurs sont automatiquement échappées ce qui contourne au problème d'injection SQL.

Les mots de passe seront hachés avant l'enregistrement au niveau base de données, l'authentification, qui est un mécanisme de vérification de l'identité d'une personne, est obligatoire pour effectuer les services importants de notre plateforme et nous allons spécifier des différents droits d'accès selon utilisateurs.

Comme la gestion des risques est l'un des points forts de la méthodologie 2tup, nous devons alors déterminer les principaux risques qui peuvent retarder la réalisation du projet et essayer de les éviter. Nous pouvons extraire ces techniques et d'autres non techniques. Pour les risques non techniques ils s'entourent généralement sur la perte qui peut être provoquée soit par une erreur humaine ou bien par une défaillance du matériel utilisé,

comme solution pour éviter ce problème c'est l'utilisation d'un outil de gestion de version qui conserve les différentes versions d'un projet dans un espace de stockage public. Le plus important risque technique que nous pourront faire face avec, c'est les différent surface d'affichage susceptible d'être utilisé pour afficher notre application, pour remédier à ceci nous devons utiliser un mécanisme d'adaptation de notre interface avec les différent tailles d'écran.

b. Sécurité du paiement en ligne

Le présent projet offre la possibilité du paiement en ligne aux clients, d'où l'obligation que cette transaction s'effectue en toute sécurité pour garantir la crédibilité de notre projet. La phase du paiement est très sensible et critique. Pour qu'elle s'effectue en toute sécurité, dans notre cas, elle sera assurée par la solution de l'intermédiaire de paiement. Un intermédiaire de paiement est un professionnel qui met en rapport deux parties intéressées à la conclusion d'une opération bancaire. En Tunisie c'est Société Monétique de Tunisie (SMT) qui joue ce rôle. Le système se base sur le Serveur de Paiement Sécurisé (SPS) qui permet d'offrir les autorisations sur les cartes bancaires, dans un environnement sécurisé. Toutes les cartes de paiement locales ou étrangères sont acceptées. Le SPS permet à un site d'accepter des paiements en ligne dans des conditions de sécurité maximales. Les transactions initiées sur le site du commerçant sont prises en charge par le SPS pour le cryptage et le traitement sécurisé des données bancaires. Le serveur de la plateforme reçoit en ligne une confirmation du paiement et peut ainsi exécuter les étapes finales du processus de livraison des produits. Une fois une commande est faite, le système envoi les paramètres de la commande, tel que la référence de cette dernière, à notre intermédiaire de paiement. Ce dernier effectue les transactions nécessaires et renvoi le résultat de la commande (succès ou bien échec).

3. Choix de la plateforme d'hébergement :

Une plateforme e-commerce c'est une application web accessible par internet dont la base de données est hébergée sur un serveur distant d'où la nécessité de la protection des données.

Parmi les premières étapes à effectuer lors de création d'une application web, c'est de trouver un hébergeur qui répond à nos besoins. Dans notre cas les critères à prendre en compte sont :

- **Autoriser la réécriture des url** : L'hébergeur doit autoriser la réécriture des URL puisque le Framework utilisé lors de création de l'application est basé sur cet aspect.
- **Indexation par le serveur APACHE des dossiers interdite** : Au niveau de l'application cette caractéristique est garantie par le biais du fichier .htaccess
- **Stockage** : Chaque application est différente en termes de capacité de stockage qu'elle en a besoin. Pour notre cas l'espace de stockage doit être grand puisqu'on a une interaction assez importante, ainsi que l'ensemble des données concernant les produits peut être très grande.
- **Bandé passante** : c'est la quantité des données (pour le téléchargement et le téléversement) qui est fournie par l' hébergeur. La présente application se base essentiellement sur le téléversement.
- **Forum** : Puisque la plateforme offre un forum d'interaction pour les utilisateurs alors on doit choisir un service d'hébergement web qui supporte les forums.

Conclusion

Durant cette phase nous avons mis le point sur les besoins fonctionnels et techniques du système pour se préparer à la prochaine phase d'analyse et conception.



Chapitre V Analyse et conception



Introduction

Après avoir identifié les exigences fonctionnelles et techniques du système, on s'occupera durant cette phase d'analyse et de conception en clôturerons par présenter la navigation.

I. Analyse

La phase d'analyse consiste aux découpages en catégories, au développement du modèle statique et au développement du modèle dynamique.

1. Découpage en catégorie

Le découpage en catégories constitue la première activité de l'étape d'analyse. Ce découpage consiste en un regroupement logique de classes à forte cohérence interne et faible couplage externe. Nous avons regroupé les classes candidates en 5 catégories :

La catégorie notification a été isolée des autres puisqu'elle présente une fonctionnalité classique qui sera déclenché selon l'état de certaines autres classes qui font partie de la catégorie interaction, de même pour la catégorie paiement dont la classe dépend de la classe commande qui fait partie de la catégorie vente. La catégorie activité membre contient un ensemble de classe qui rend des services de mêmes natures à l'utilisateur, de même pour la catégorie interaction.

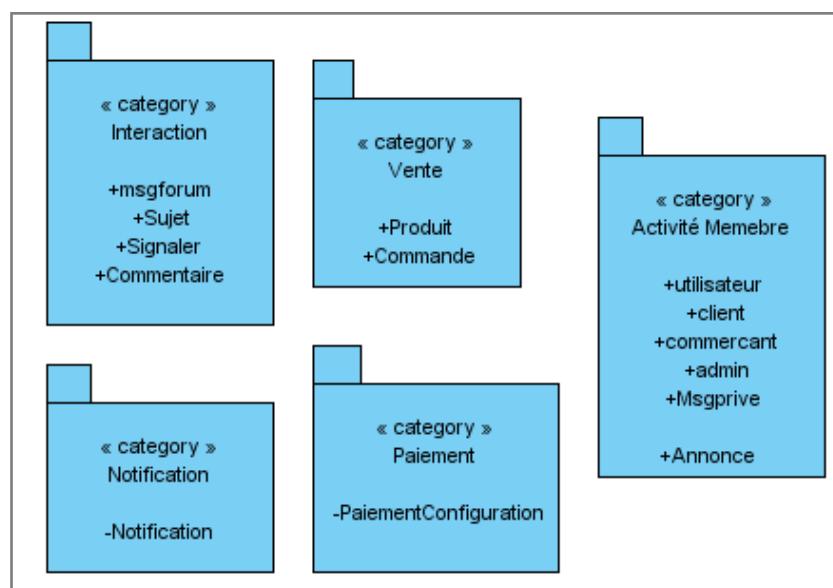


Figure 56 Découpage en catégories

2. Développement du modèle statique

Cette étape consiste à définir les différentes relations entre les classes d'une catégorie.

a. Diagramme de classe de la catégorie Interaction

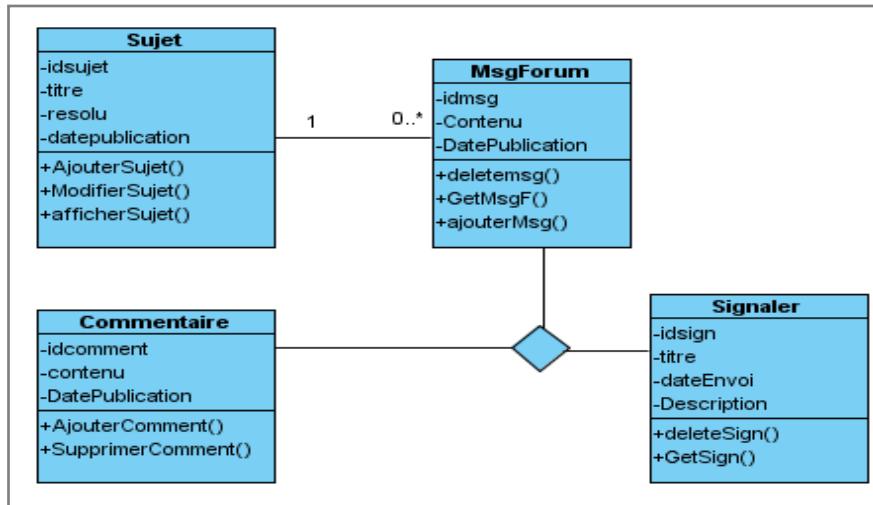


Figure 57 Diagramme de classe « Interaction »

b. Diagramme de classe de la catégorie Vente

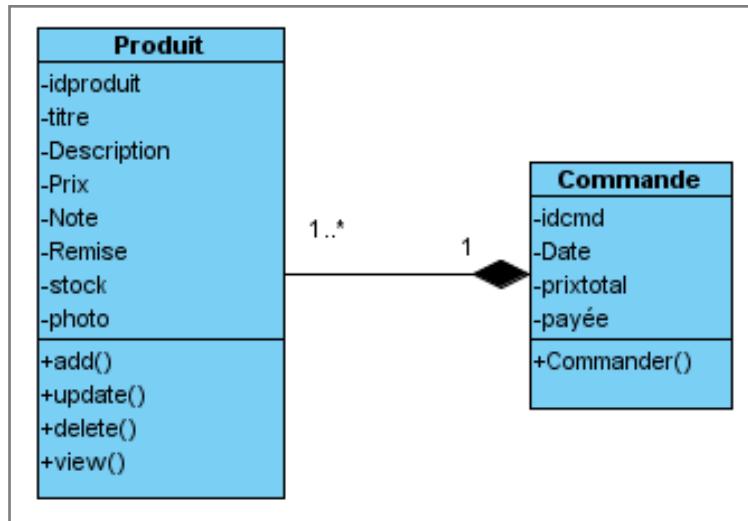


Figure 58 Diagramme de classe « Vente »

a. Diagramme de classe de la catégorie activité Membre

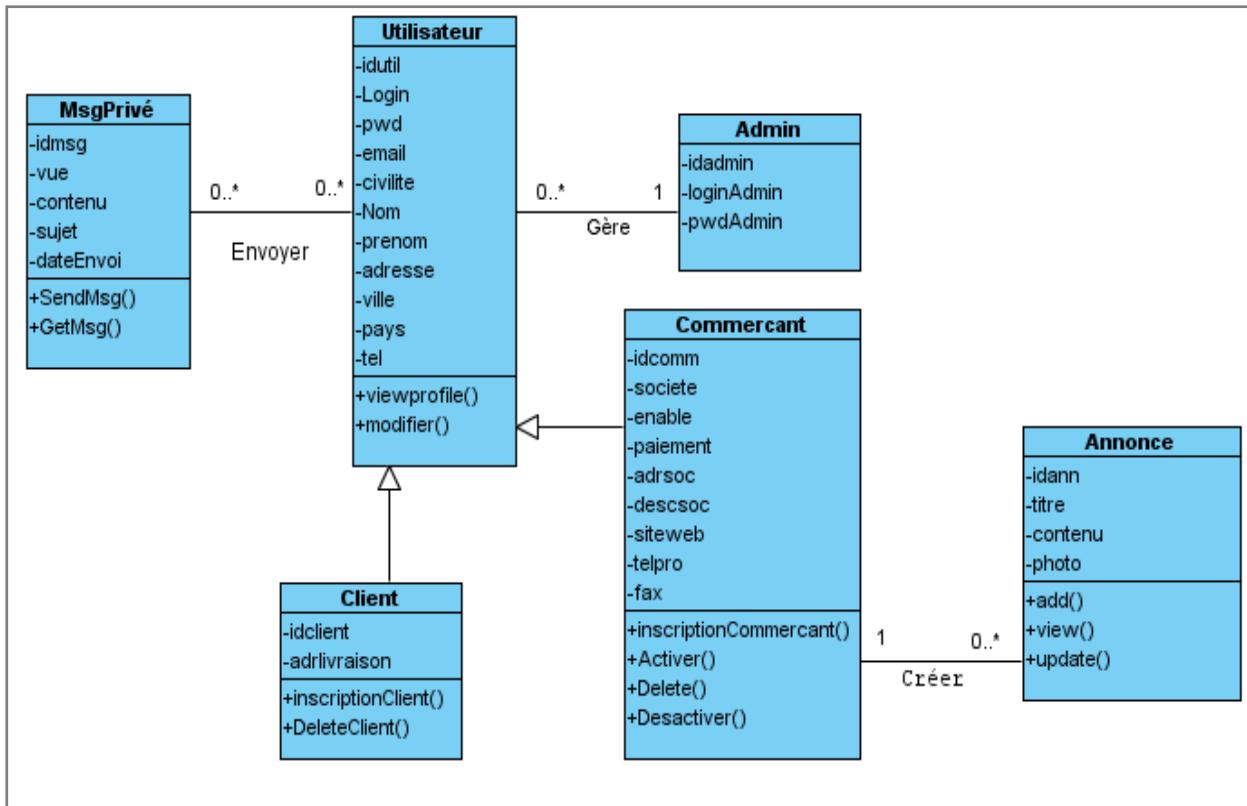


Figure 59 Diagramme de classe « Activité Membre»

b. Diagramme de classe de l'application

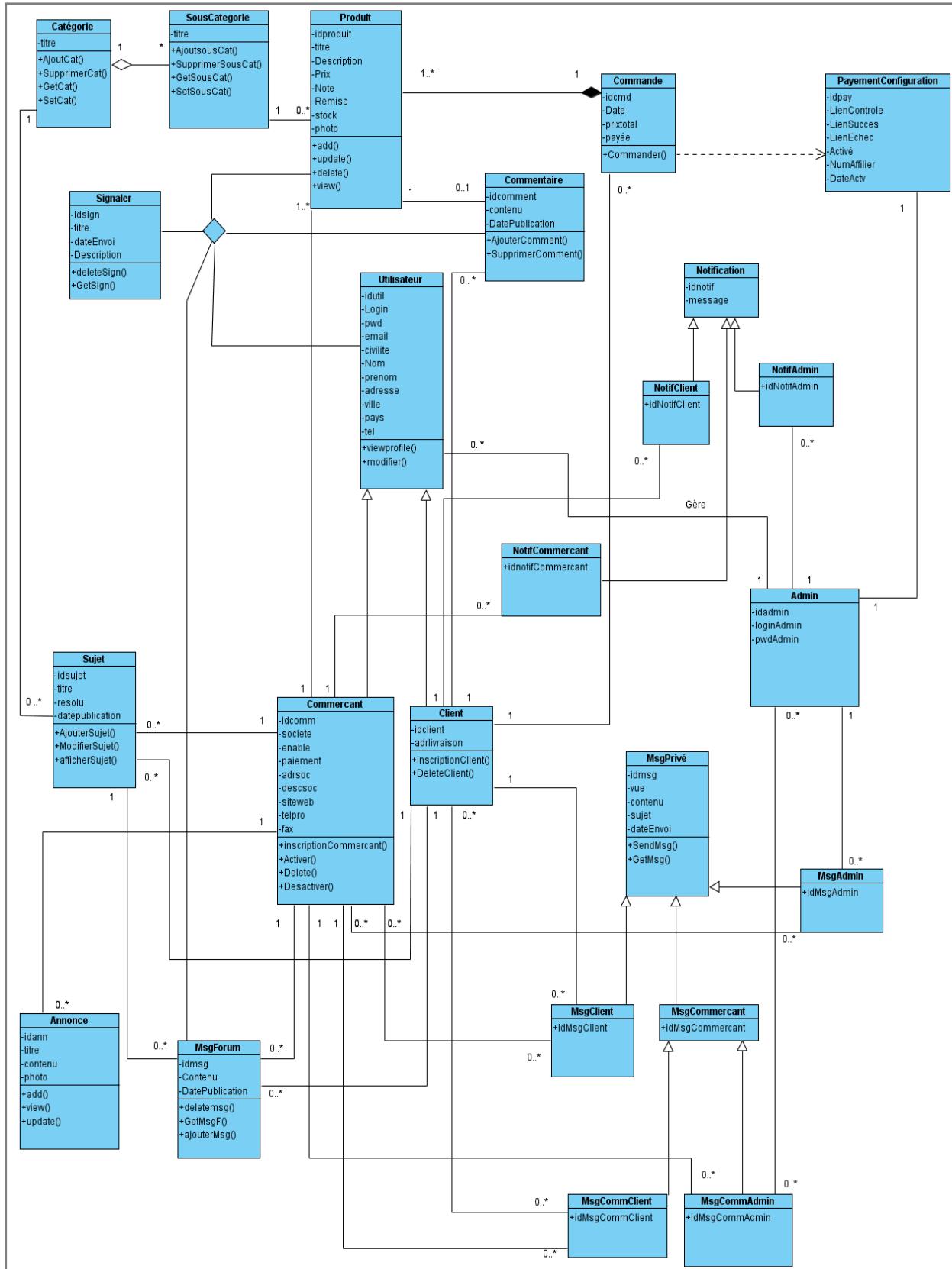


Figure 60 Diagramme de classe

c. Dictionnaire de données

Attribut	Explication
Titre	- Désigne le libelle selon la classe ou il existe : définit le nom de la catégorie, la sous catégorie, le titre d'une annonce et le libelle d'un produit
Description	- C'est une petite description du produit en question
prix	- Le prix de vente d'un produit
Note	- La note attribuée à un produit de la part des clients
Remise	- Le pourcentage de remise pour un produit
Stock	- La quantité du produit en stock
photo	- Une photo pour le produit, annonce
Date	- La date de réalisation de la commande d'achat
prixtotal	- Le prix total de la commande
payé	- Un champ booléen qui montre si le paiement est effectué ou non
Contenu	- Le message d'un commentaire, d'un message forum, d'une annonce ou d'un message privé
Datepublication	- La date de publication d'un commentaire, d'un sujet ou d'un message forum
Login	- Le nom d'utilisateur d'un internaute inscrit sur la plateforme
pwd	- Le mot de passe d'internaute inscrit sur la plateforme
email	- L'email d'internaute inscrit sur la plateforme
civilite	- La civilité d'internaute inscrit sur la plateforme (monsieur, madame ou mademoiselle)
Nom	- Le nom d'internaute inscrit sur la plateforme
Prenom	- Le prénom d'internaute inscrit sur la plateforme
adresse	- L'adresse d'internaute inscrit sur la plateforme
Ville	- La ville d'internaute inscrit sur la plateforme
Pays	- Le pays d'internaute inscrit sur la plateforme
tel	- Le numéro de téléphone d'internaute inscrit sur la plateforme
Lien contrôle	- Un des trois liens fournis pour accomplir la démarche de paiement en ligne

Lien success	- Un des trois liens fournis pour accomplir la démarche de paiement en ligne
Lien echec	- Un des trois liens fournis pour accomplir la démarche de paiement en ligne
NumAffilier	- C'est un numéro unique fournit de la part de l'intermédiaire de paiement en ligne
DateActv	- La Date d'activation de la procédure de paiement pour le commerçant
Message	- Le message a affiché lors d'envoi d'une notification
Resolu	- Cet attribut exprime si le sujet forum est résolu ou non
adrLivraison	- C'est l'adresse de livraison en cas d'achat en ligne
Societe	- Le nom de la société du commerçant
enable	- Exprime si le commerçant a été activé par l'administrateur ou pas.
paiement	- Signifie si le commerçant a activé le paiement en ligne ou non
adrsoc	- L'adresse de la société du commerçant
descsoc	- Description du domaine d'activité de la société
siteweb	- Le site web de la société
telpro	- Numéro de téléphone de la société
fax	- Numéro Fax de la société
Vue	- Désigne si le message a été consulté ou non
Sujet	- Définit l'objet du message
Dateenvoi	- La date d'envoi d'un message
Loginadmin	- Le nom d'utilisateur pour l'administrateur
pwdadmin	- Le mot de passe pour l'administrateur

Tableau 23 Dictionnaire de données

3. Développement du modèle dynamique

Cette étape présente la troisième activité de la phase d'analyse, durant laquelle on va détailler quelques scénarios en relations avec les cas d'utilisation par le biais des diagrammes de séquences. Dans une deuxième partie nous présenterons les diagrammes d'état-transition de certaines instances de classes.

a. Diagrammes de séquences

Nous présenterons les interactions dynamiques entre objets, nous schématiserons alors les dialogues, les contrôles et les entités.

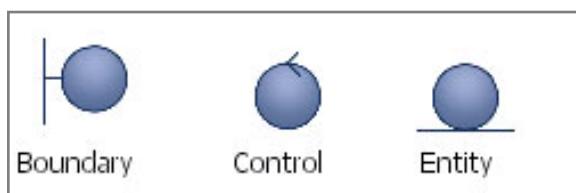


Figure 61 Schématisation du diagramme de séquence

- Scénario de « gérer profil »

Les différents scénarios qui sont en relation avec ce cas d'utilisation :

❖ Scénarios nominaux :

→ Prfl_N1 : Modifier profil

❖ Scénarios alternatifs :

→ Prfl_A1 : Modifier mot de passe

→ Prfl_A1 : Visualiser profil

❖ Scénarios d'exceptions :

→ Prf_E1 : Le nouveau login modifié existe déjà dans la base de donné.

❖ Description détaillée du scénario Prfl_N1

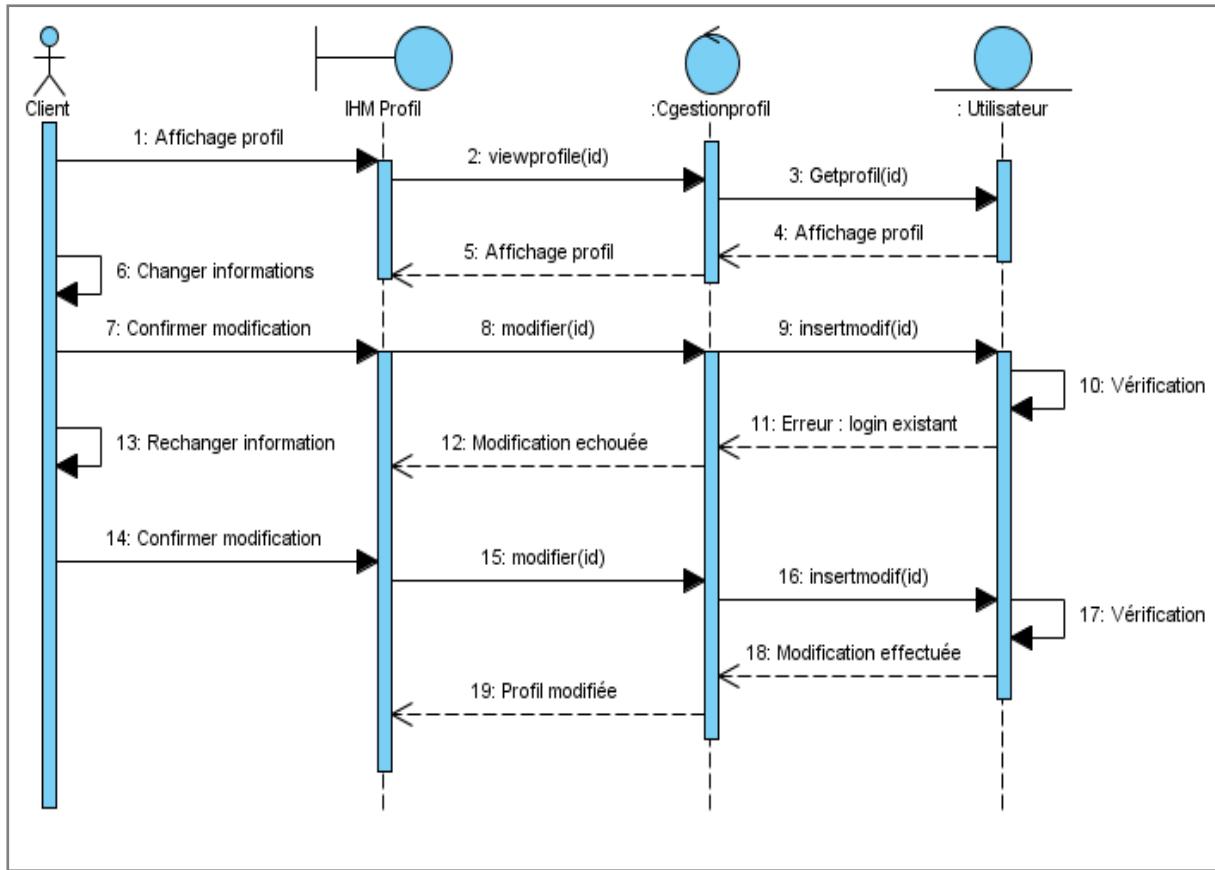


Figure 62 Diagramme de séquence du cas d'utilisation « Gérer profil »

- Scénario de « gérer espace forum »

Les différents scénarios qui sont en relation avec ce cas d'utilisation :

❖ Scénarios nominaux :

- Ajs_N1 : Ajouter sujet
- Ajs_N2 : Ecrire commentaire

❖ Scénarios alternatifs :

- Ajs_A1:Modifier sujet
- Ajs_A2 : Marquer sujet comme résolu
- Ajs_A3 : Supprimer commentaire
- Ajs_A4 : Signaler commentaire

❖ Scénarios d'exceptions :

- Ajs_E1: Le commentaire publié est vide
- Ajs_E2: Le sujet publié est vide

❖ Description détaillée du scénario Ajs_N1

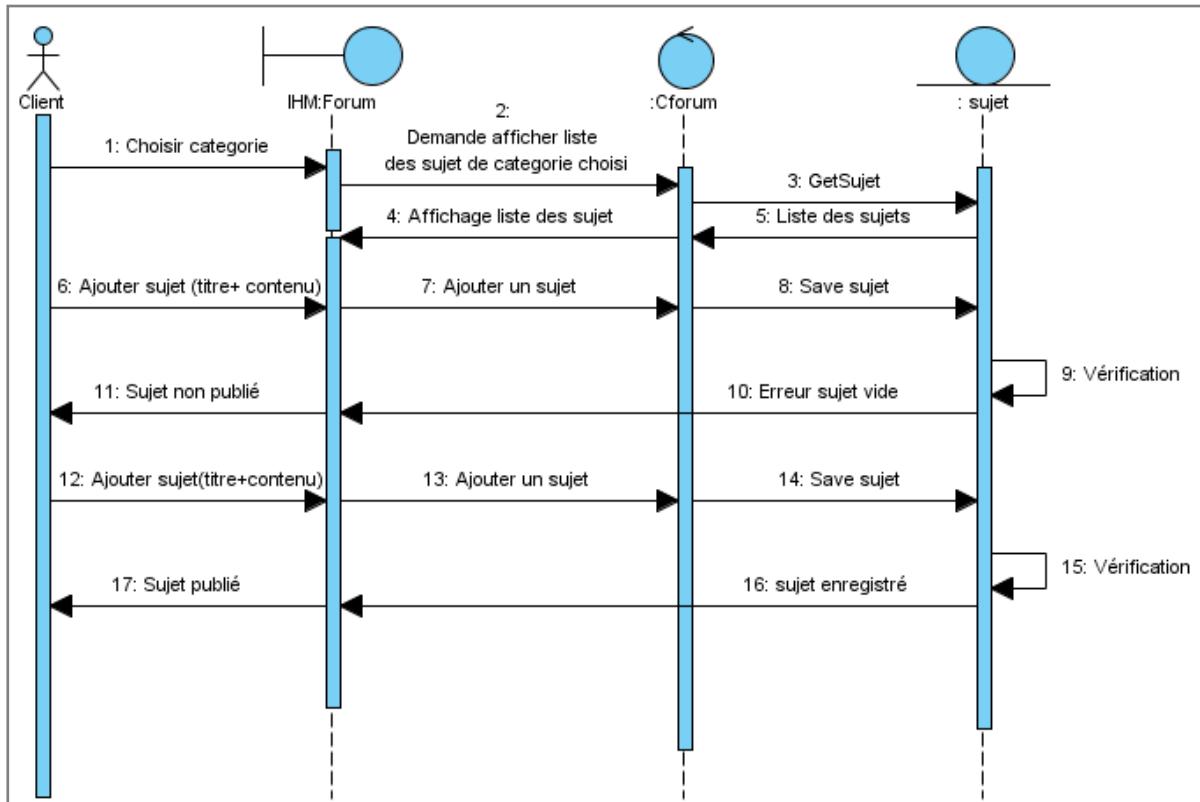


Figure 63 Diagramme de séquence du scénario « Ajs_N1 : Ajouter sujet»

❖ Description détaillée du scénario Ajs_N2

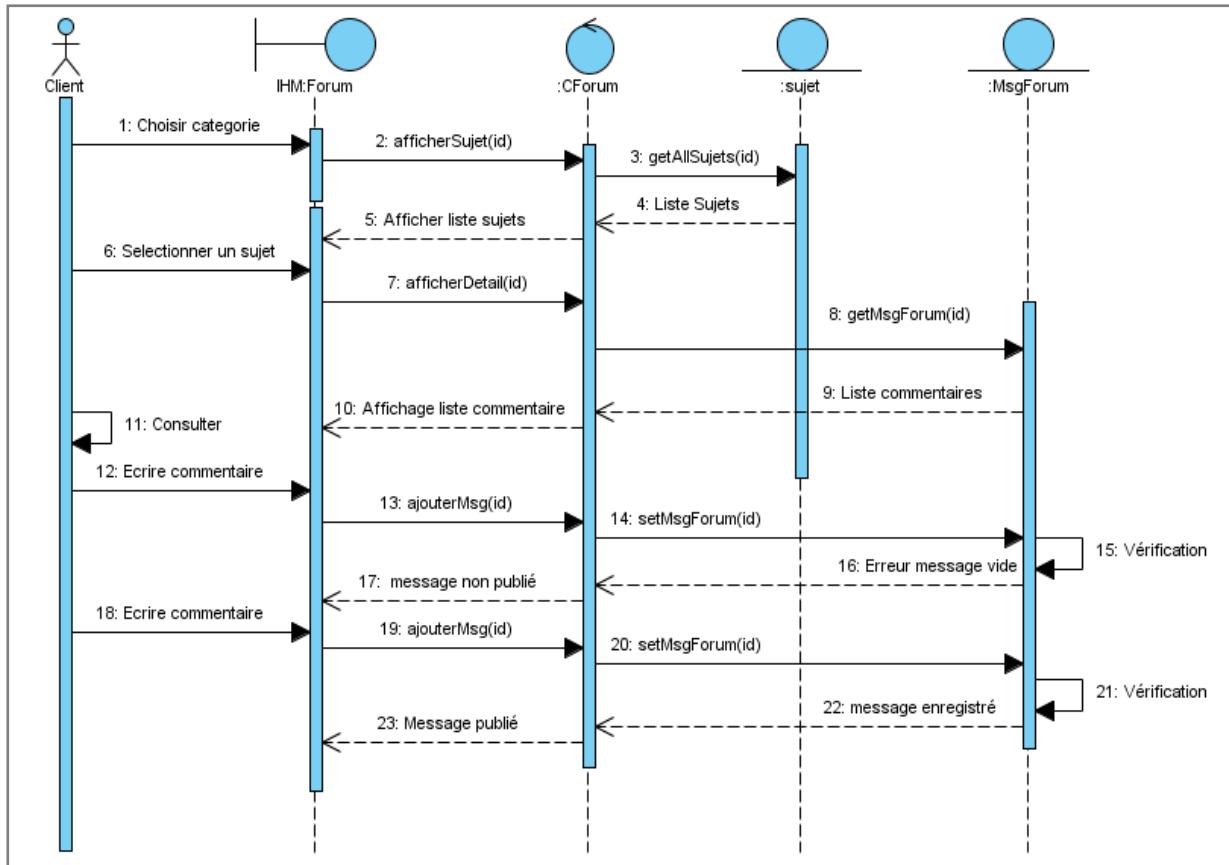


Figure 64 Diagramme de séquence du scénario « Ajs_N2 : Ecrire commentaire»

- Scénario de « gérer achats »

Les différents scénarios qui sont en relation avec ce cas d'utilisation :

❖ Scénarios nominaux :

→ Ach_N1 : Acheter produit

→ Ach_N2 : Suivre achats

❖ Scénarios alternatifs :

→ Ach_A1: Signaler produit, annonce

→ Ach_A2 : Noter commerçant

❖ Scénarios d'exceptions :

→ Ach_E1 : Informations de paiement non valide

❖ Description détaillée du scénario Ach_N1

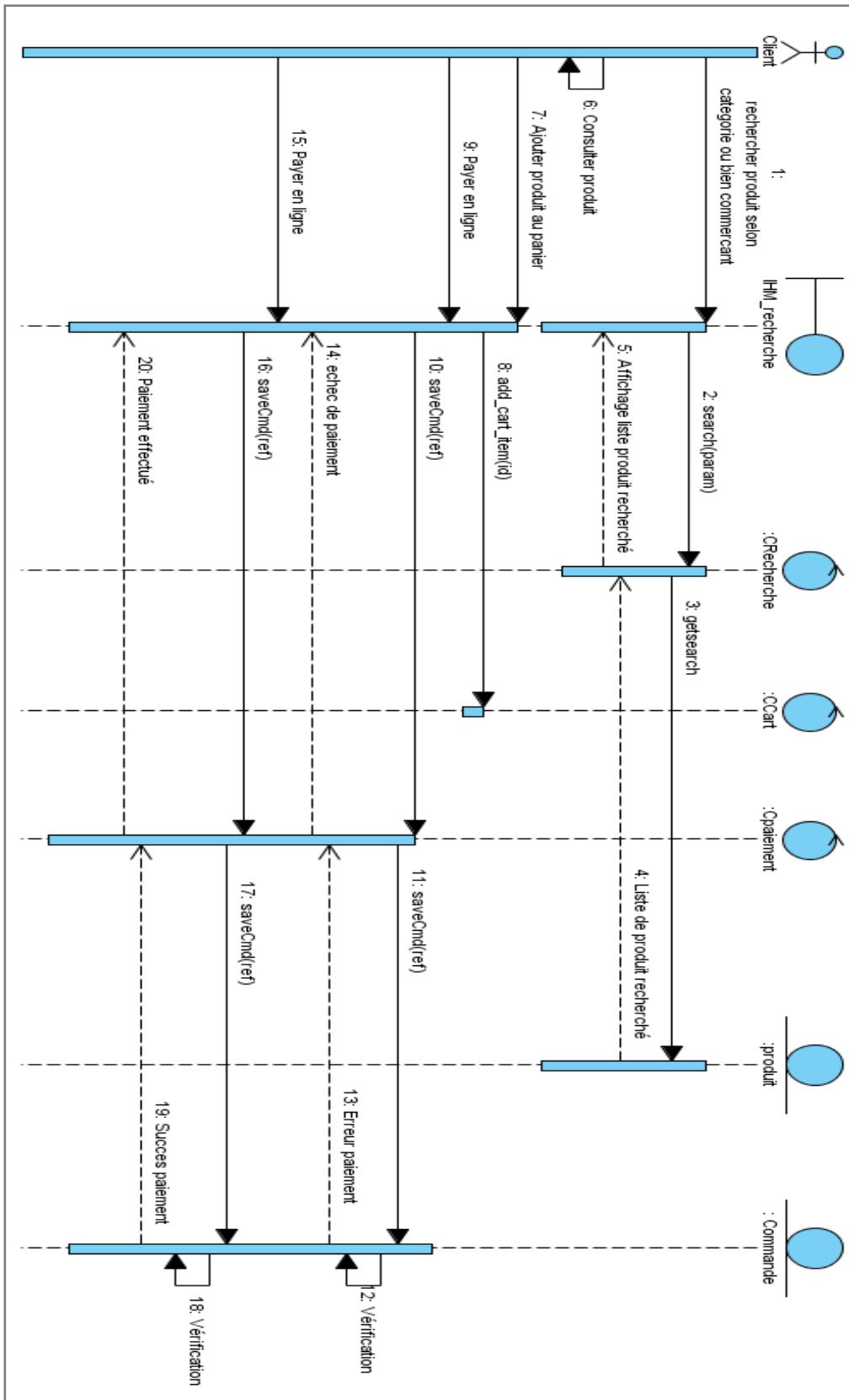


Figure 65 Diagramme de séquence du scénario « Ach_N1 : Acheter produit»

b. Diagrammes d'états-transitions

Le diagramme états-transitions est un automate à états finis décrivant les différents états par lesquels passe une instance quelconque d'une classe en réponse à des événements. On a choisi de présenter les diagrammes concernant les classes produit et Utilisateur.

- Diagramme d'état-transition pour la classe « Produit »

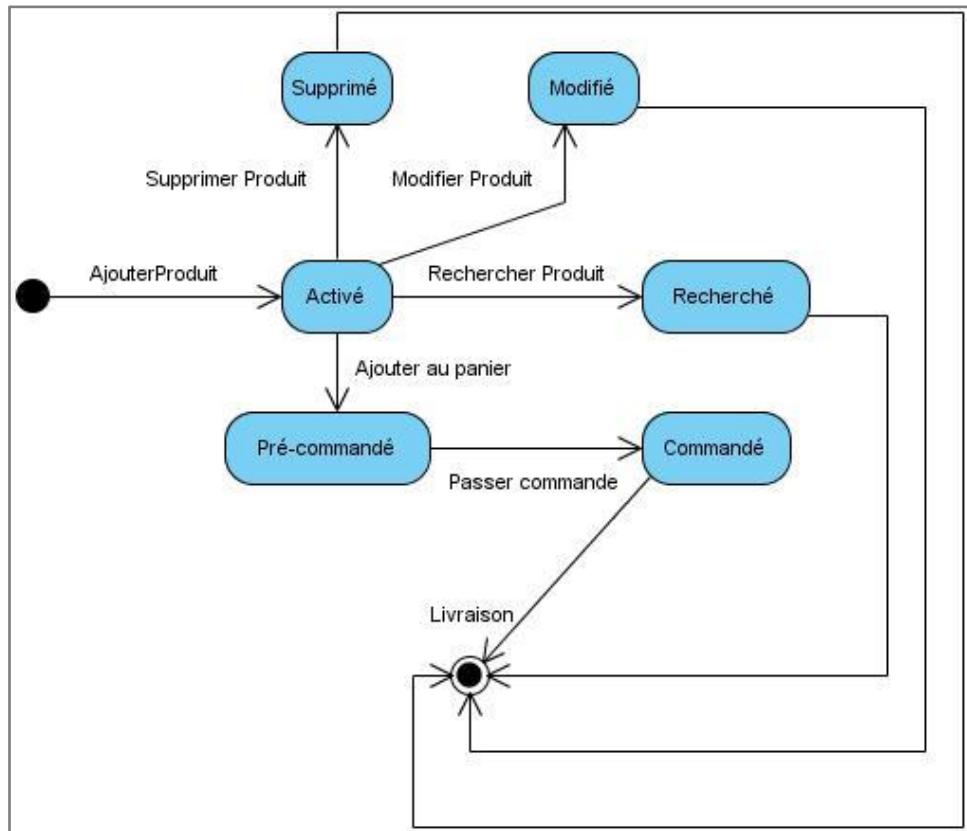


Figure 66 Diagramme d'état-transition « Produit»

- **Diagramme d'état-transition pour la classe «Utilisateur »**

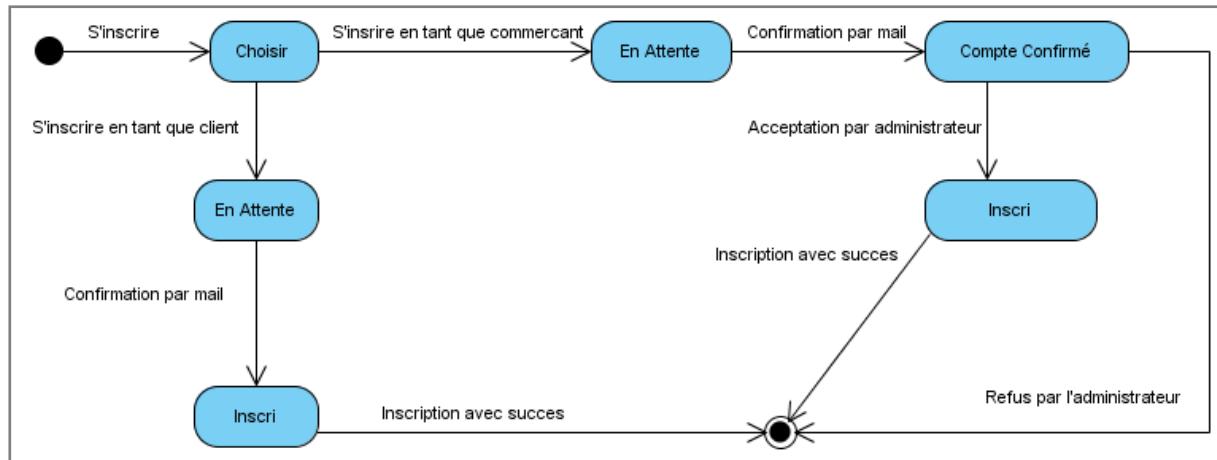


Figure 67 Diagramme d'état-transition « Utilisateur»

II. Conception

Au cours de cette partie nous commencerons par présenter l'architecture logicielle adoptée ainsi que le modèle MVC utilisé. Nous clôturerons cette partie par présenter le modèle de base de données.

1. Architecture logicielle

Pour avoir une architecture robuste, modulable et évolutive, il nous faut utiliser le principe de « couche ». Nous séparerons donc au maximum les différents types de traitement de l'application (Accès aux données, Métier, Présentation). Ceci correspond à une architecture 3-Tiers.

L'architecture 3-tiers permet de distinguer rigoureusement la couche «présentation», la couche «métier» et la couche «données» d'une application.

Le tiers présentation, réalise l'interface graphique de l'application, gère les interactions avec l'utilisateur et la logique de navigation. Le tiers métier contient des composants qui collaborent pour régler des problèmes métiers ou de logique applicative relatifs au domaine d'application du système d'information. Le tiers persistance est utilisé par la couche métier pour conserver des informations sur un support de persistance (base de données, fichiers, ...).

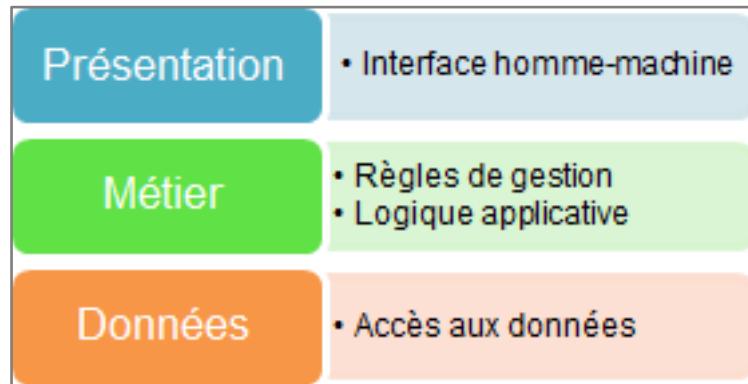


Figure 68 Architecture 3tiers

De nombreux Framework utilisent aujourd'hui MVC, qui est une méthode de conception pour le développement d'applications logicielles, car le but principal de ce motif est de séparer les couches logiques d'une application. Ce motif sera plutôt réservé aux sites complexes, dans lesquels beaucoup de redondances de code apparaissent en général, des sites souvent mis à jour, souvent remodelés. Dans le cadre d'une petite application web la complexité en termes de code, apportée par MVC, ne sera pas justifiée. En revanche pour tout autre projet web, la séparation en plusieurs couches permet à différentes équipes de travailler chacune sur une couche, indépendamment des autres. Et c'est aussi un gros avantage lorsqu'une couche doit évoluer, sans que les autres n'en aient besoin.

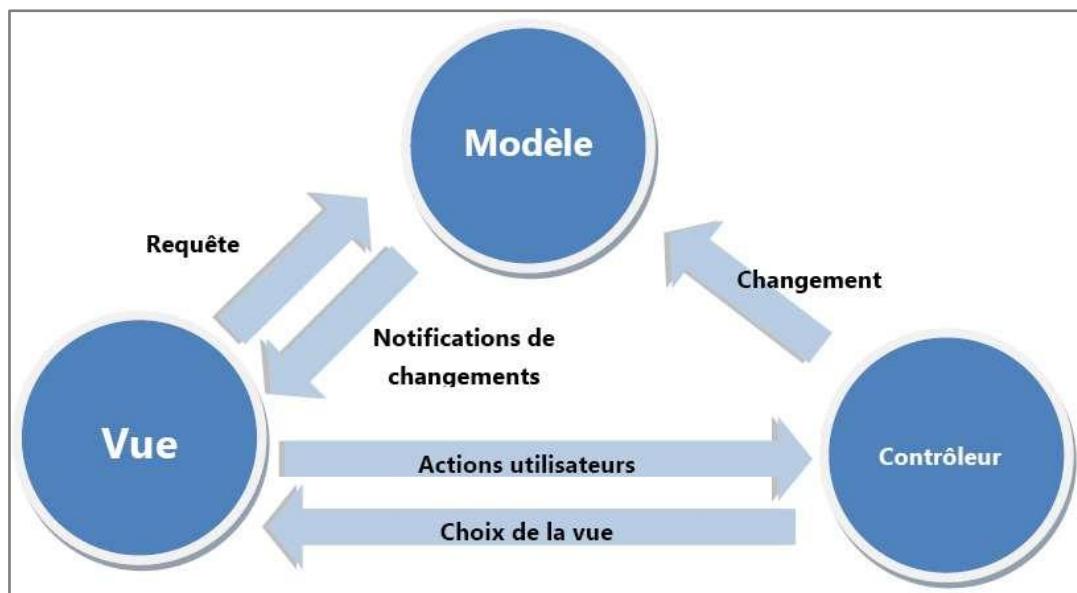


Figure 69 Architecture MVC

2. Schéma de la base de données

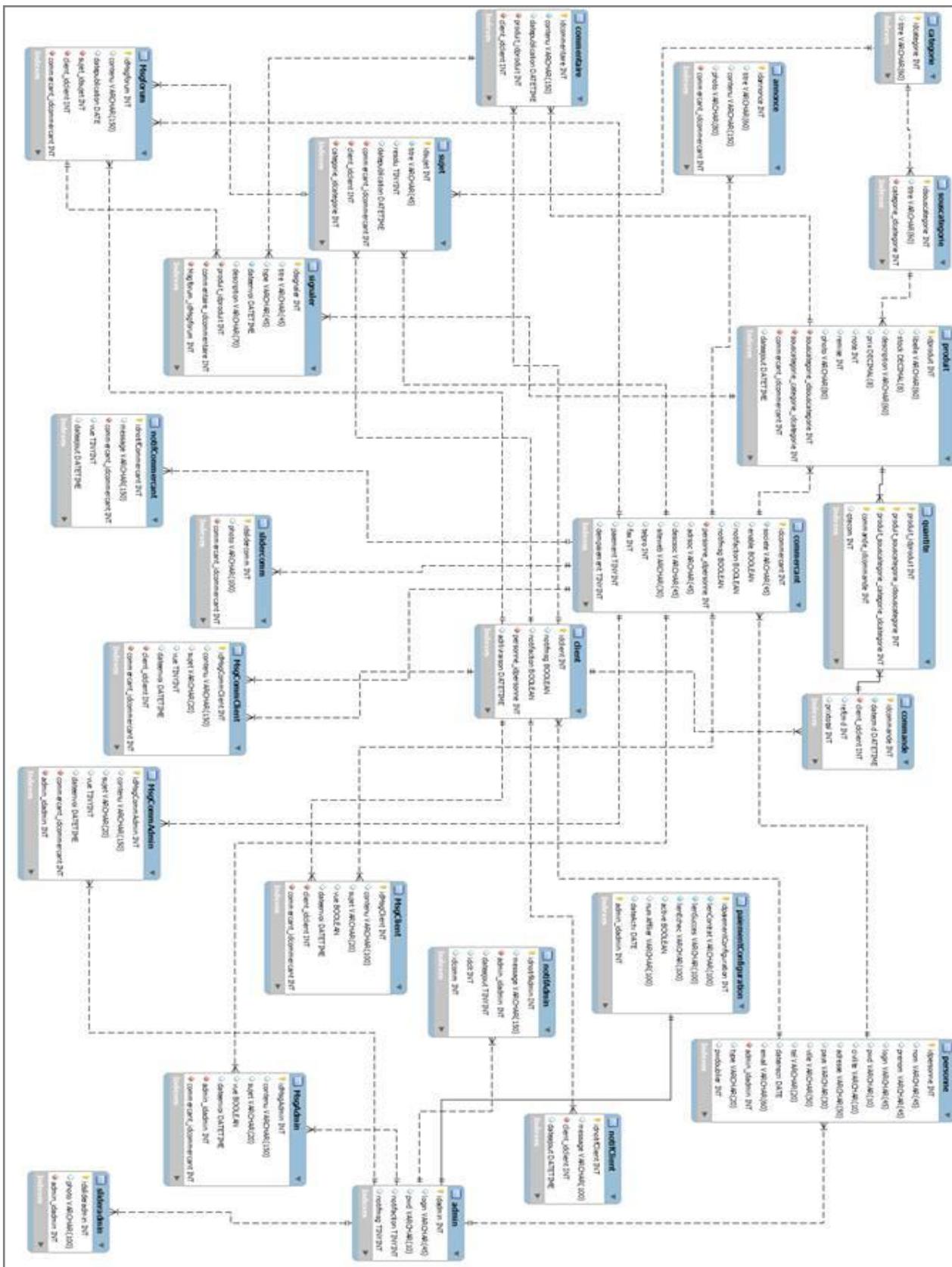


Figure 70 Schéma de la base des données

III. Modélisation de la navigation

UML 2 offre la possibilité de représenter formellement la navigation des acteurs au moyen d'un diagramme de navigation. Le diagramme de navigation représente un ajout important dans notre modélisation et fournit la possibilité de décrire précisément et exhaustivement les aspects dynamiques des interfaces. Nous nous intéressons dans cette partie à présenter :

Pour l'application web :

- La navigation du Client.
- La navigation du Commerçant.

Pour l'application mobile :

- La navigation du Client.

- **Notations graphiques des diagrammes de navigation**

- une page complète est représentée par un cadre «page» :
- un frame, « désignant un bloc », particulier à l'intérieur d'une page est représenté par un cadre «frame» :
- une erreur ou un comportement inattendu du système est représenté par un cadre «exception» avec un niveau de gris clair.
- une liaison vers un autre diagramme d'activité, pour des raisons de structuration et de lisibilité est représenté par un cadre «connector» avec un niveau de gris foncé.

1. Diagramme de navigation pour l'application web :

a. Navigation du client :

Nous nous intéressons aux deux cas d'utilisations:

- Gérer les achats
- Effectuer le paiement

- **Gérer les achats :**

Le client démarre par la page d'accueil de notre plateforme, comme indiqué dans la Figure 71 , à partir de laquelle il commence en général par sélectionner des produits pour pouvoir ensuite les commander, après avoir éventuellement géré son panier.

À partir de la page d'accueil, le client doit avoir la possibilité de lancer une recherche rapide, une recherche avancée, ou de flâner dans un ensemble de frame telles que : nouveautés, Les plus appréciés, etc. La recherche est un frame toujours présent dans les pages. La recherche elle-même est une action dont le résultat aboutit à une page contenant la liste des produits ou bien à une page d'erreur indiquant qu'aucun produit n'a été trouvé.

Selon le résultat de recherche, le client pourra soit sélectionner un produit pour obtenir ses détails, soit le mettre directement dans son panier, soit le noter, ou bien consulter le fournisseur du produit.

Si le nombre de produit trouvés est grand, le client pourra naviguer entre les différentes pages de résultats.

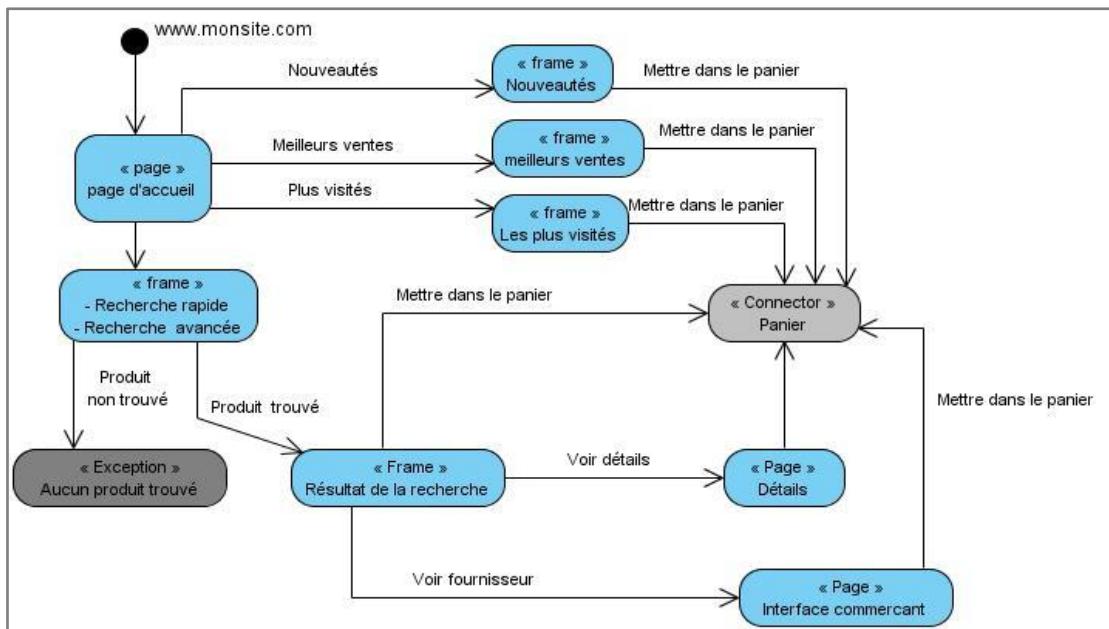


Figure 71 Navigation Client « gérer achat 1/3»

En effet, l'accès au panier est disponible de partout, y compris de la page d'accueil. Il faut donc traiter une «exception» Panier vide, qui empêche de passer à la page de commande ainsi que dans le cas où le panier est vide.

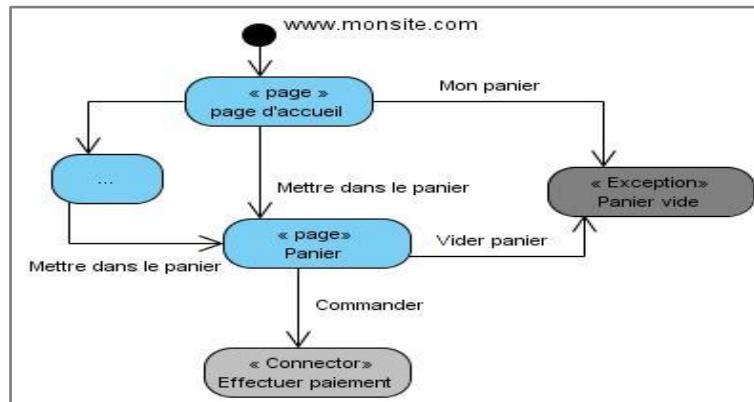


Figure 72 Navigation Client « gérer achat 2/3»

Par le cadre contenant des points, au niveau de la Figure 72, nous entendons dire que ce connecteur est relié avec d'autres pages qui sont déjà mentionnées dans la Figure 71.

La figure suivante détaille le connecteur « effectuer paiement ».

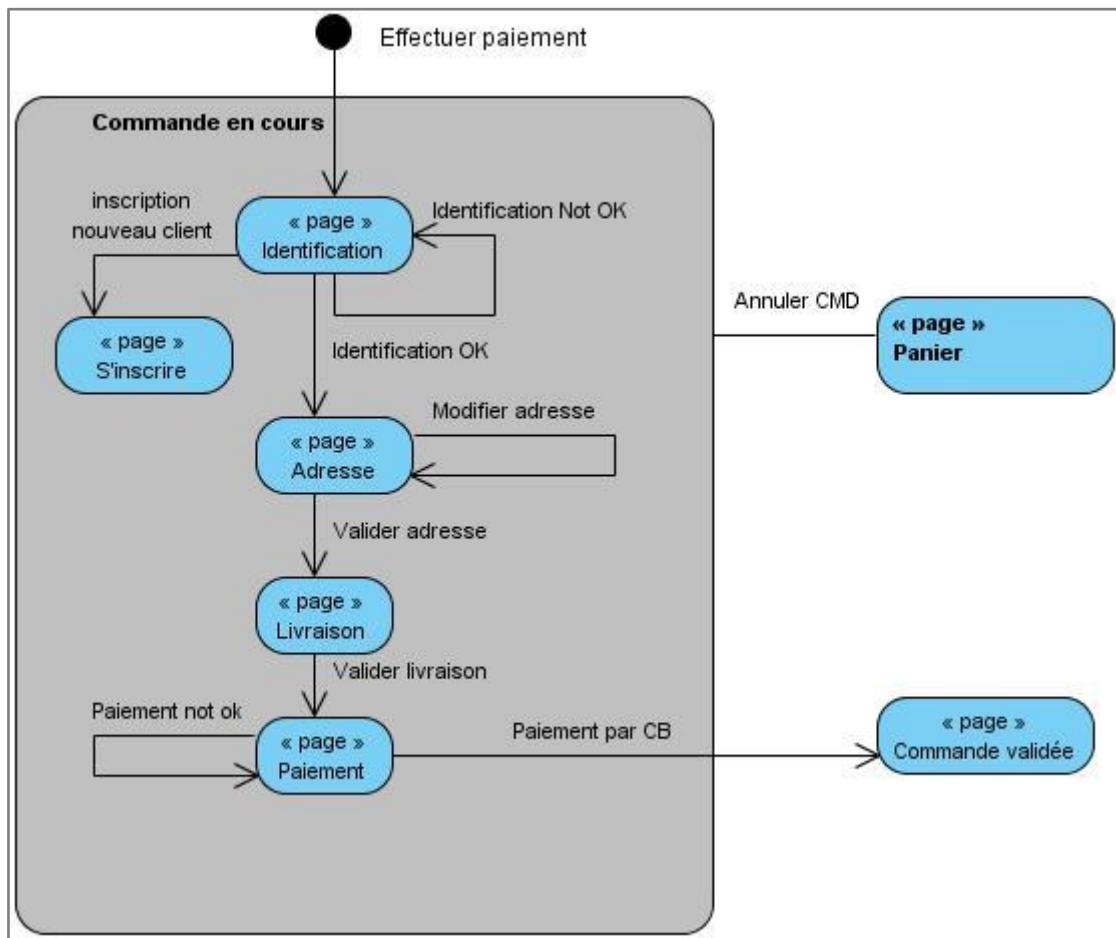


Figure 73 Navigation Client « gérer achat 3/3»

Vu que l'étape de la réalisation de la commande est très délicate, cette dernière s'effectue sans interruption, en cas d'annulation on doit refaire tous le travail du connector effectuer paiement présent dans la Figure 73.

b. Navigation du Commerçant :

Nous nous intéressons aux cas d'utilisations :

- S'inscrire
- S'authentifier
- Gérer produit
- Gérer annonce
- Visualiser l'état du commerce

À partir de la page d'accueil, un commerçant doit s'inscrire pour pouvoir gérer le contenu de son profil y compris les produits et les annonces.

- **S'inscrire :**

Une page d'inscription s'affiche, en tant que client ou commerçant, le commerçant remplis ses informations ainsi que les informations concernant sa société.

- **S'authentifier**

Il s'agit d'une simple page d'authentification où le commerçant doit saisir son nom d'utilisateur et son mot de passe afin d'accéder aux différentes pages de gestion produits et annonces.

- **Gérer produit, gérer annonces :**

Il s'agit d'une page dont s'affiche l'ensemble de produit du commerçant, qui a la possibilité soit d'ajouter, supprimer, modifier, consulter, ou bien rendre le produit publiable.

De même pour les annonces.

- **consulter l'état du commerce :**

A partir de son profil, un commerçant a la possibilité d'accéder à la page consulter l'état du commerce qui contient les statistiques des ventes et des notes de produit.

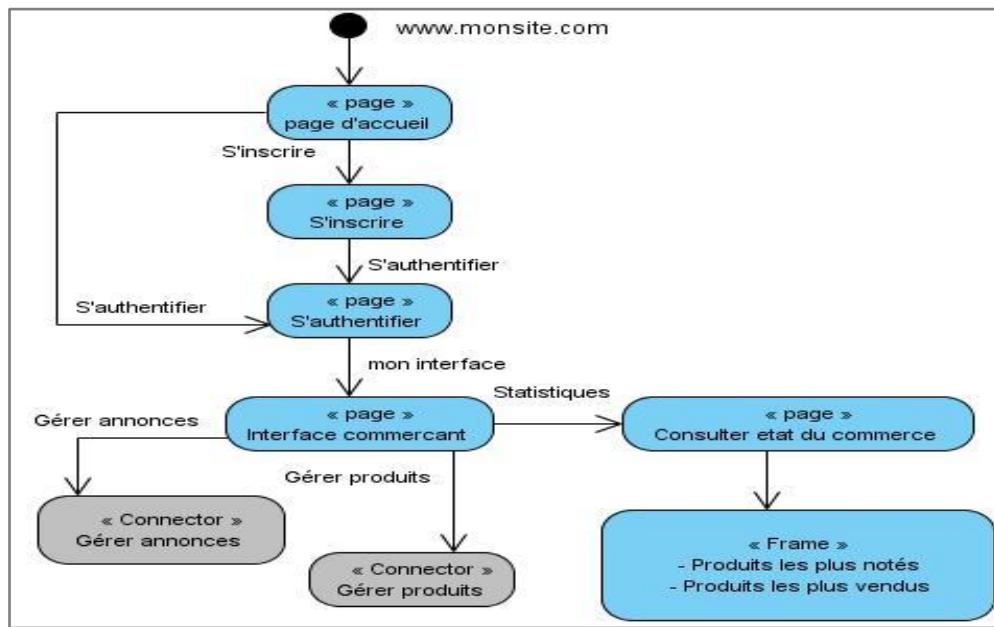


Figure 74 Navigation Commerçant 1/2

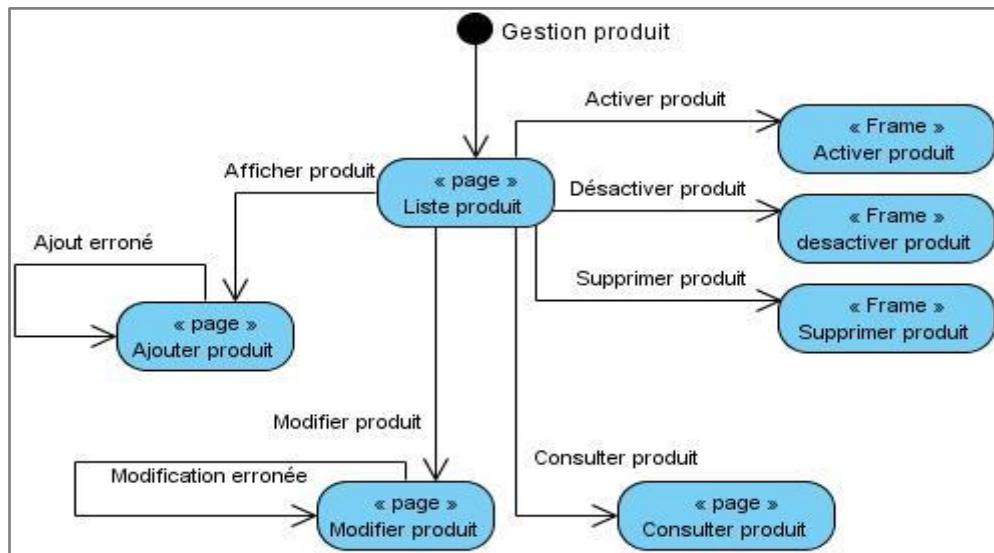


Figure 75 Navigation Commerçant 2/2

2. Diagramme de navigation pour l'application mobile:

La page d'accueil de l'application mobile contient un frame recherche qui permet aux clients de chercher par mot clé les produits, et un frame contenant les différentes catégories qui permet de trouver une liste de produit trié selon la catégorie souhaitée.

Une fois le produit est choisi une page de détails s'affiche qui contient toute les informations nécessaires. Le client peut consulter les informations sur le fournisseur du produit en cliquant sur le bouton fournisseur. Aussi, le client a le droit d'ajouter ce produit

sur son panier pour qu'ensuite il soit redirigé vers la page qui contient la liste de produit à acheter. Enfin deux choix existent, soit vider le panier ou bien effectuer une commande. La page d'accueil contient aussi un frame forum qui contient la liste des sujets par catégorie. La Figure 76 présente le diagramme de navigation complet coté client.

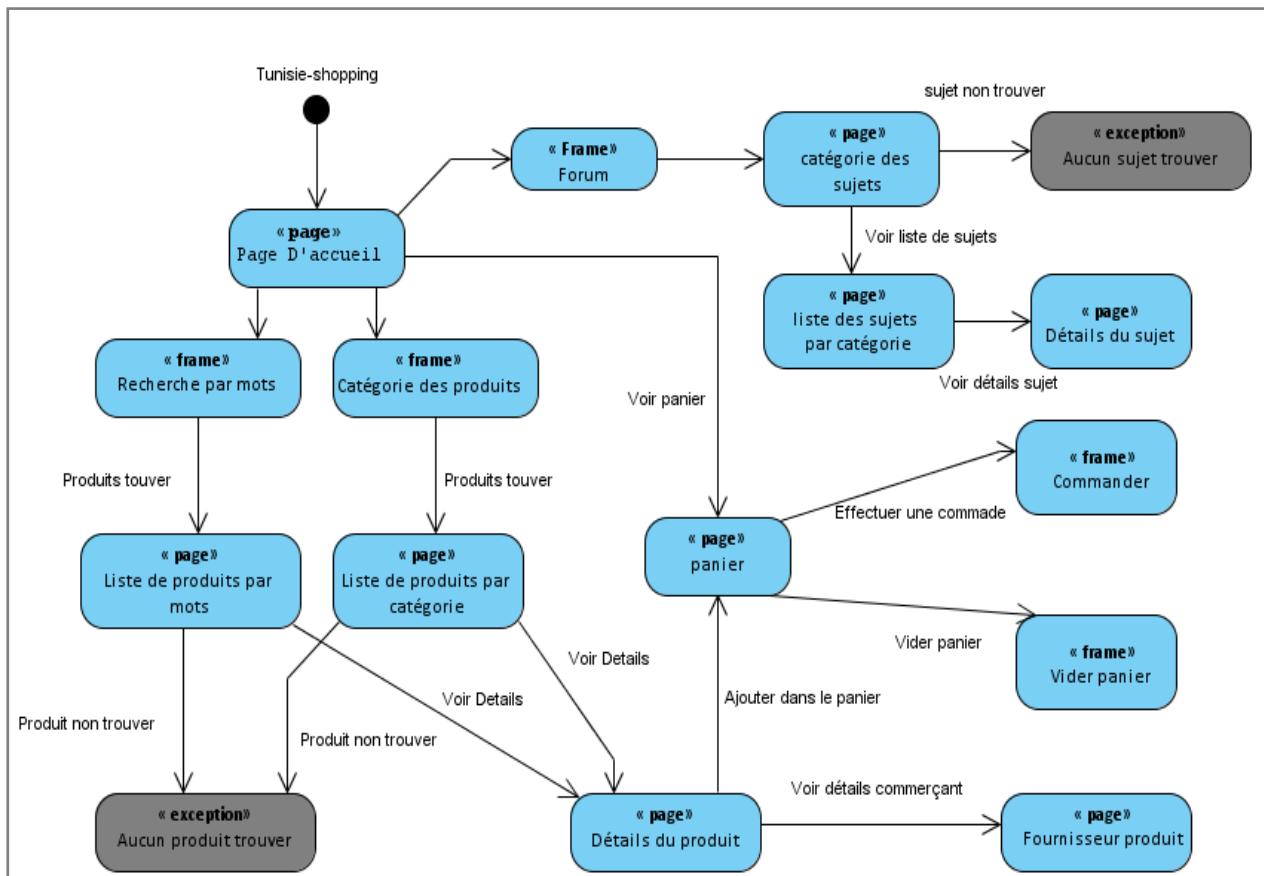


Figure 76 Navigation Application mobile

Conclusion

Tout au long de ce chapitre nous avons montré les étapes qui ont conduit à la construction du modèle d'analyse qui a pour but de définir la structure et le comportement des objets connus dans le métier des utilisateurs du système. Maintenant nous avons un modèle prêt à coder, l'implémentation de ce modèle sera l'objectif de l'étape suivante et le sujet du prochain chapitre où nous présenterons la réalisation qui doit être conforme à la conception déjà établie.



Chapitre VI Réalisation



Introduction

Ce chapitre sera consacré pour la phase d'implémentation qui constitue l'étape d'achèvement et d'aboutissement du projet. Nous détaillerons les étapes d'installation et configuration du Framework ainsi que nous présenterons les principaux scénarios de l'application.

I. Architecture de l'application

Notre application est composée d'une application web développée avec le Framework Codeigniter et une application mobile codé avec le Framework PhoneGap, la Figure 77 représente l'architecture de notre solution.

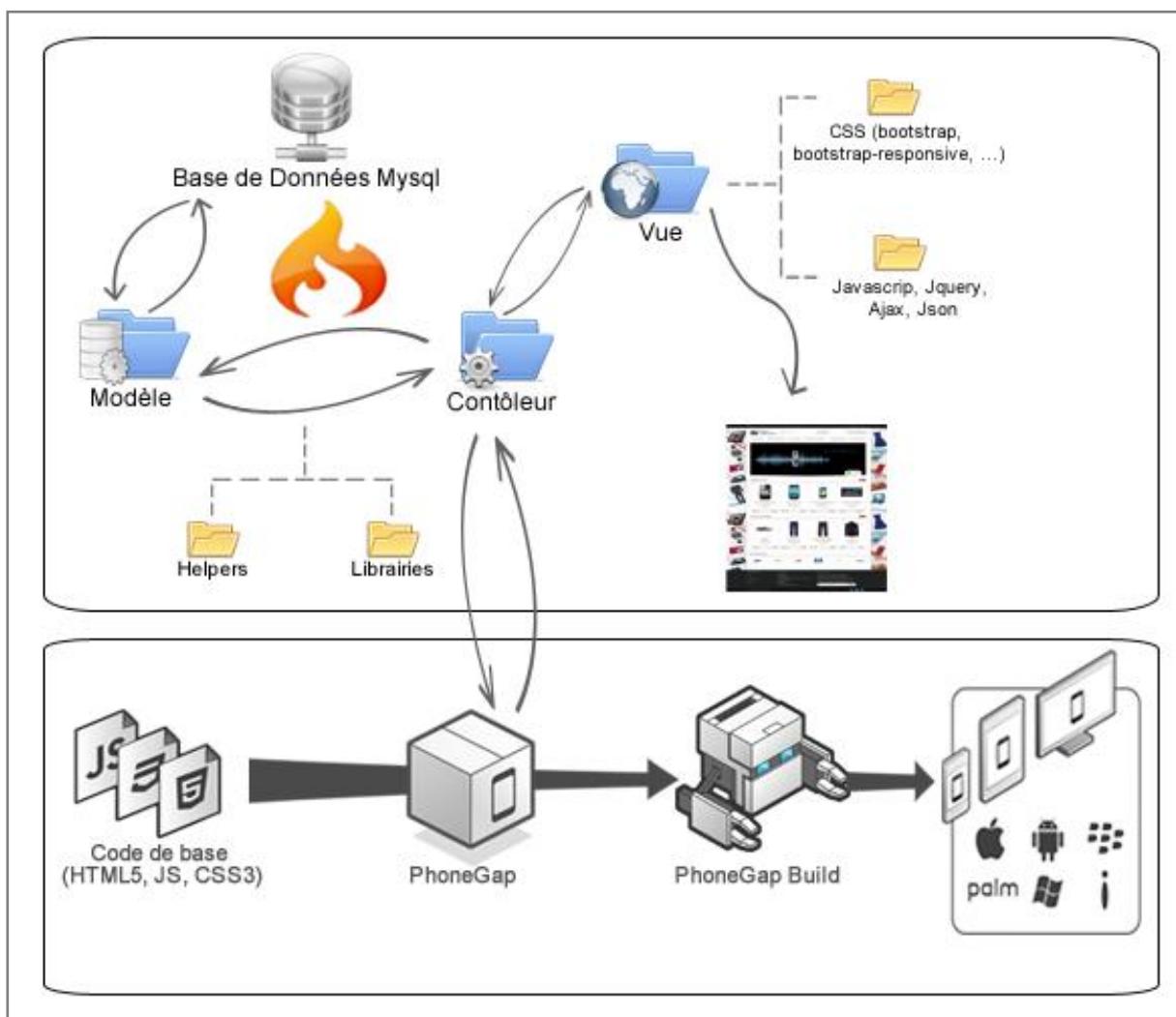


Figure 77 Architecture de la plateforme e-commerce

II. Installation et configuration de l'application

L'environnement logiciel choisi pour la réalisation de l'application c'est le Framework PHP « CodeIgniter » dans sa version 2.1.3

1. Environnement matériel

Tout au long de ce projet, le travail a été réalisé sur un environnement matériel comportant deux machines ayant les configurations suivantes :

- **Fabricant** : Toshiba, Dell
- Intel® Pentium® Dual CPU T4400 @2.20GHz, Intel Core2 Duo
- **RAM** : 4GO, 1GO
- **Système d'exploitation** : 32 bits

2. Environnement logiciel

Les outils logiciels utilisés lors de la phase de conception et de développement de l'application sont :

- Système d'exploitation : Windows Seven
- Outils de développement :
 - **Framework** : CodeIgniter
 - **Editeur de diagramme UML** : Visual Paradigm for UML 6.4 Enterprise Edition.
 - **Netbeans** : C'est un environnement de développement intégré (EDI) Version 6.8.
 - **Wamp server** : WampServer 2
 - **Logiciel de base de données visuelle** : MySQL Workbench 5.2 CE
 - **Outil de gestion de version** : GitHub
 - **Moteur de template** : Twig
 - **Framework CSS** : Twitter Bootstrap

3. Installation du Framework Codeigniter

L'installation du Framework Codeigniter consiste à télécharger le Framework, et le placer sous le dossier www de notre serveur web avec la réalisation de quelques configurations.

- **Définir l'environnement de travail**

Au niveau du fichier index.php la première chose à configurer c'est de définir si le Framework est en période de développement, de test ou de production.

```
define('ENVIRONMENT', 'development');
```

Figure 78 Configuration environnement

- **Définir le chemin vers notre application et le dossier system**

```
$system_path = 'system';
$application_folder = 'application';
```

Figure 79 Configuration emplacement application

4. Configuration de l'application

La configuration sera faite dans le fichier « config.php » se trouvant sous le dossier application. Nous commençons par la définition de l'adresse de la machine virtuelle de notre application comme définit dans la Figure 80

- **Définir le virtual host**

```
Listen 8081
<VirtualHost 127.0.0.1:8081>
ServerName localhost
DocumentRoot "C:\wamp\www\platform"
DirectoryIndex
<Directory "C:\wamp\www\platform">
AllowOverride All
Allow from All
</Directory>
</VirtualHost>
```

Figure 80 Virtual host

- **Activer l'url rewriting :**

Elle est utilisée pour éviter l'affichage de la page « index.php » au niveau de l'url. Pour cela on crée un fichier « .htaccess » contenant le code suivant :

```
1 RewriteEngine On
2 RewriteCond $1 !^(index\.php|static|uploads|assets|js|img|css)
3 RewriteRule ^(.*)$ index.php/$1 [L]
```

Figure 81 Fichier .htaccess de Codeigniter

Un autre fichier est important lors de la configuration c'est le fichier « database.php » qui contient les coordonnées de connexion à la base de données.

```
$db['default']['hostname'] = 'localhost';
$db['default']['username'] = 'root';
$db['default']['password'] = '';
$db['default']['database'] = 'bdcomm';
$db['default']['dbdriver'] = 'mysql';
$db['default']['dbprefix'] = '';
$db['default']['pconnect'] = TRUE;
$db['default']['db_debug'] = TRUE;
$db['default']['cache_on'] = FALSE;
$db['default']['cachedir'] = '';
$db['default']['char_set'] = 'utf8';
$db['default']['dbcollat'] = 'utf8_general_ci';
$db['default']['swap_pre'] = '';
$db['default']['autoinit'] = TRUE;
$db['default']['stricton'] = FALSE;

/*
 * End of file database.php
 */
/* Location: ./application/config/database.php */
```

Figure 82 Fichier « database.php » de Codeigniter

Le fichier « autoload » lui aussi facilite la tache de codage, il vous permet d'inclure dès le lancement de Codeigniter des bibliothèques, des helpers, ainsi que des modèles et fichiers de langue.

```
|   $autoload['libraries'] = array('database', 'session', 'xmlrpc');
*/
$autoload['libraries'] = array('database', 'session');

/*
-----
| Auto-load Helper Files
-----
| Prototype:
|
|   $autoload['helper'] = array('url', 'file');
*/
$autoload['helper'] = array('url', 'form', 'cookie');

/*
*/
```

Figure 83 Fichier « autoload.php » de Codeigniter

5. Intégration du moteur de Template

Afin de pouvoir utiliser twig il a fallu effectuer quelque configurations. Au début nous avons ajouté un fichier nommé twig.php dans le dossier config qui contient les configurations indiqué dans la Figure 84.

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
$config['template_dir']= APPPATH . 'views';
$config['template_ext']= 'twig';
$config['debug']= TRUE;
$config['cache']= FALSE;
$config['auto_reload']= TRUE;
$config['charset']= 'utf-8';
$config['base_template_class']= 'Twig_Template';
$config['strict_variables']= FALSE;
$config['autoescape']= FALSE;
$config['optimizations']= -1;
```

Figure 84 Fichier de configuration twig.php

Ensuite nous avons intégré la librairie twig dans le dossier « lib ». La Figure 85 présente un extrait de la librairie :

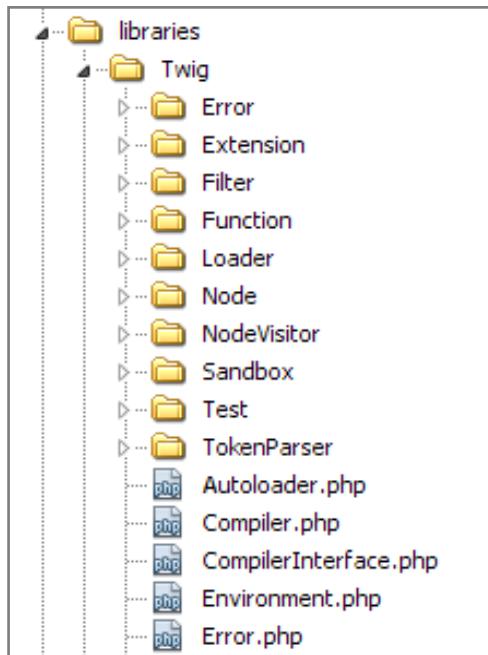


Figure 85 Extrait de la librairie twig

Pour ne pas répéter à chaque fois l'appel de la librairie twig dans tous les contrôleur nous l'avons ajouté dans le fichier autoload.php comme indiquer la Figure 86.

```
$autoload['libraries'] = array('database', 'session', 'cart', 'twig', 'form_validation', 'table',
```

Figure 86 Autoload de la librairie twig

Enfin, pour utiliser ce moteur de Template pour l'appellation des vue depuis le contrôleur nous avons utilisé la syntaxe indiquée dans la Figure 87 :

```
// load view
$this->twig->render('annonce/annonceList_view', $data);
```

Figure 87 Appellation des vue depuis le contrôleur avec twig

Coté vue, pour pouvoir appliquer le principe d'héritage des Template nous avons créé un fichier « template.twig » dont on a mis les blocks qui se répète tout au long du développement du site. Prenons l'exemple du block login dans la Figure 88 :

```
{% block login %}
{#---Navigator right---#}
<ul class="nav pull-right">
    {% if getsessionhelper() %}
        <li class="dropdown">
            <a href="#" class="dropdown-toggle" data-toggle="dropdown">
                <i class="icon-user icon-white"></i> {{ getsessionhelper() ['login'] }}
            <b class="caret"></b>
            </a>
            <.../>
        </li>
    {% else %}
        <li><a href="/inscription/inscription">Inscription</a></li>
        <li><a href="/inscription/login">login</a></li>
    {% endif %}
</ul>
{% endblock login %}
```

Figure 88 Block Login twig

Pour appeler ce Template dans les différents vus on utilise les syntaxes de la Figure 89 :

```
{% extends "/layout/layout.twig" %} héritage de template layout.twig

{% block title %} {{parent()}} - Login{% endblock title %} Modification d'un block existant
```

Figure 89 Héritage de Template layout.twig

6. Installation de PhoneGap pour Android

Pour pouvoir installer PhoneGap nous avons besoin de :

- Eclipse Classic (version 4.2.2)
- Android SDK (Version 4.2.2)
- ADT Plugin
- PhoneGap (Version 2.7.0)

Après avoir installé tous ces logiciels nous avons créé un répertoire dont nous avons décompressé l'archive phonegap-2.7.0 ensuite configuré la variable d'environnement en ajoutant ces liens

- Android SDK Tools ;C:\Android\SDK\tools
- Android platform-tools ;C:\Android\SDK\platform-tools
- Phonegap ;C:\PhoneGap\eclipse\plugins\org.apache.ant_1.8.3.v201301120609\bin

Pour Crée un nouveau projet PhoneGap nous avons besoin tous d'abord de l'invite de commande initialement situé au chemin suivant :

C:\PhoneGap\phonegap-2.7.0\lib\android\bin

Ensuite la commande suivante sera tapée :

./create C:\PhoneGap\PhoneGapWork\Plateforme com.ecommerce.platforme plateforme

Maintenant avec l'ide eclipse nous allons démarrer notre projet a partir de la source déjà existante comme le montre la Figure 90 suivante :

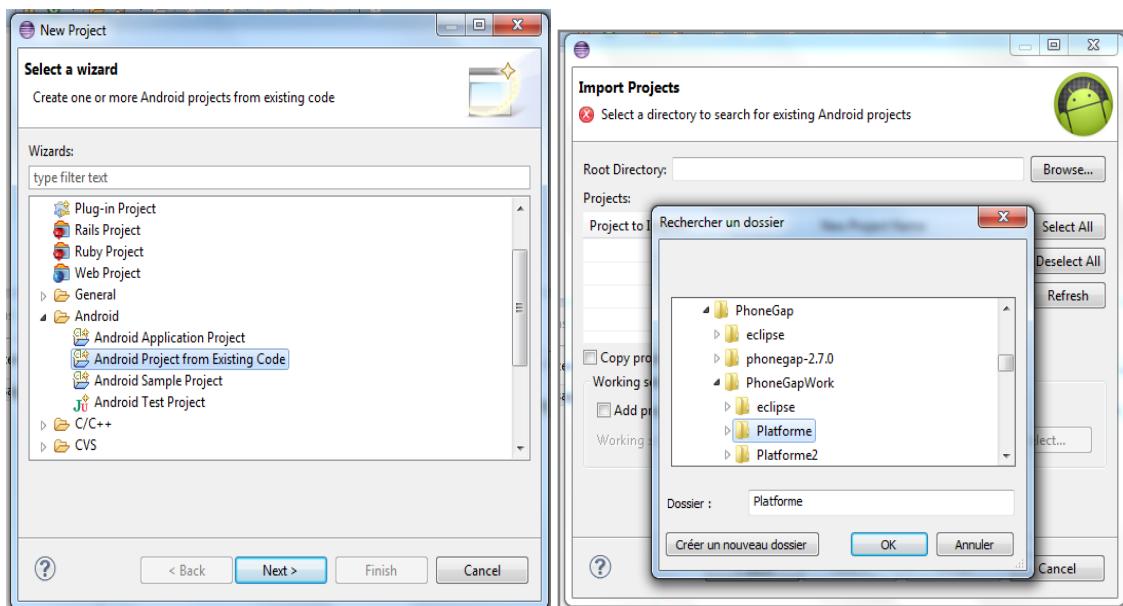


Figure 90 Création d'un nouveau projet PhoneGap

Enfin nous pourrons commencer ledéveloppement en ajoutant nos fichier html dans le dossier assets/www/ qui contiendra le fichier « cordova-2.7.0.js ». La Figure 91 présente l'arborescence de notre projet mobile :

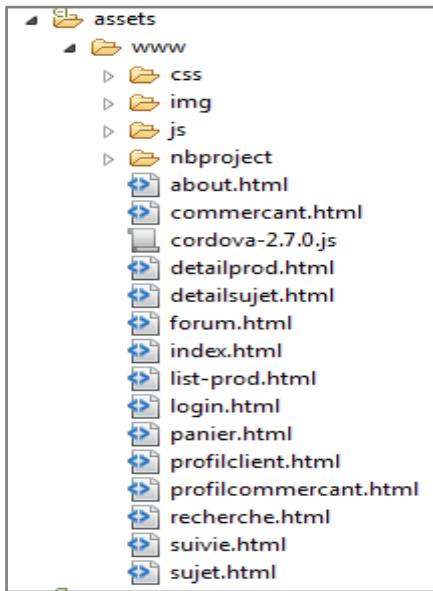


Figure 91 Arborescence du projet mobile

Pour pouvoir tester notre projet nous avons crée un émulateur Android en cliquant sur l'icône « Android Virtual device manager » et configurer l'émulateur comme indique la Figure 92.

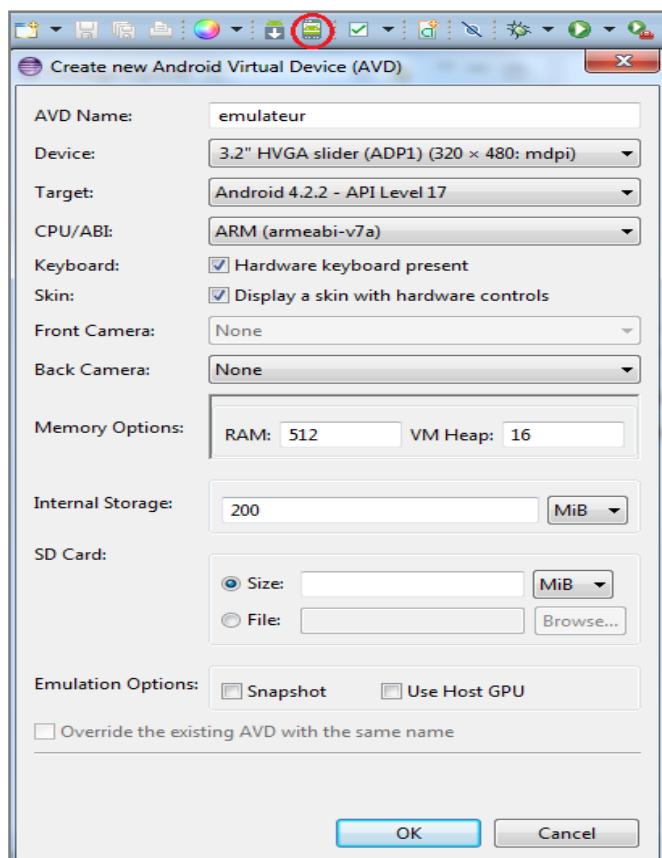


Figure 92 Emulateur Android

III. Les principaux scénarios

Durant cette partie nous mettrons en œuvre les différents scénarios susceptible de se produire en utilisant l'application, avec illustration de la démarche de développement sous le Framework Codeigniter au premier lieu et en suite celui au niveau du Framework PhoneGap.

La Figure 93 suivante montre l'interface d'accueil de notre application.

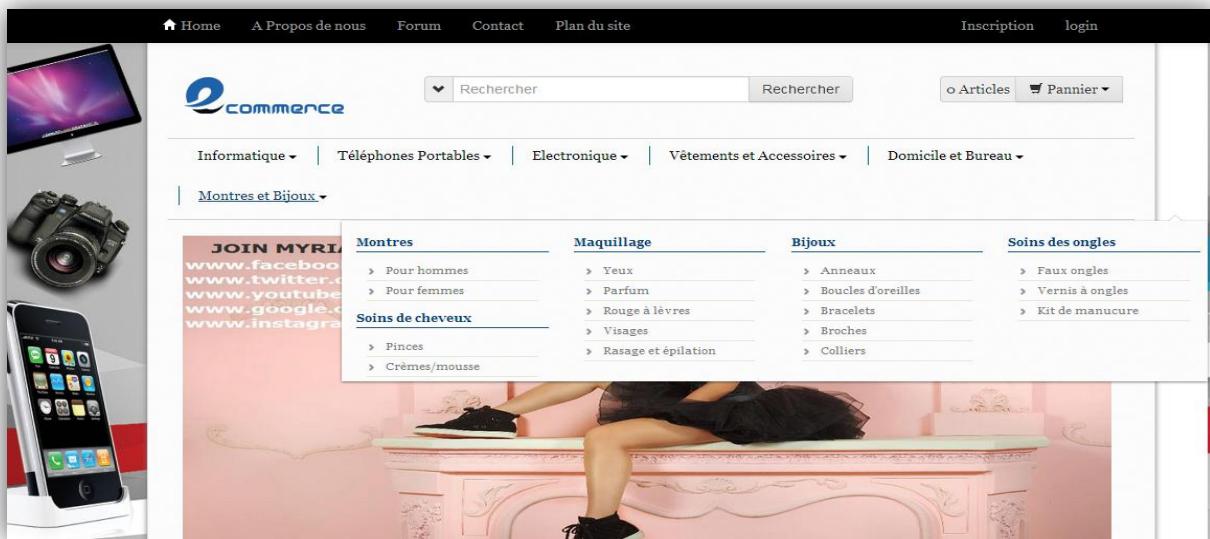


Figure 93 Page d'accueil

A partir de l'interface d'accueil chaque utilisateur pourra s'inscrire s'il est nouveau ou bien se connecter. Nous commencerons par détailler le scénario d'inscription et gestion profil.

Inscription et gestion profil

Un internaute accédant à la plateforme, pour profiter des différents services de cette dernière il doit avoir un compte.

Le scénario de l'inscription est simple, Il choisit s'il s'inscrira en tant que client ou commerçant tel que montre la Figure 94 suivante :

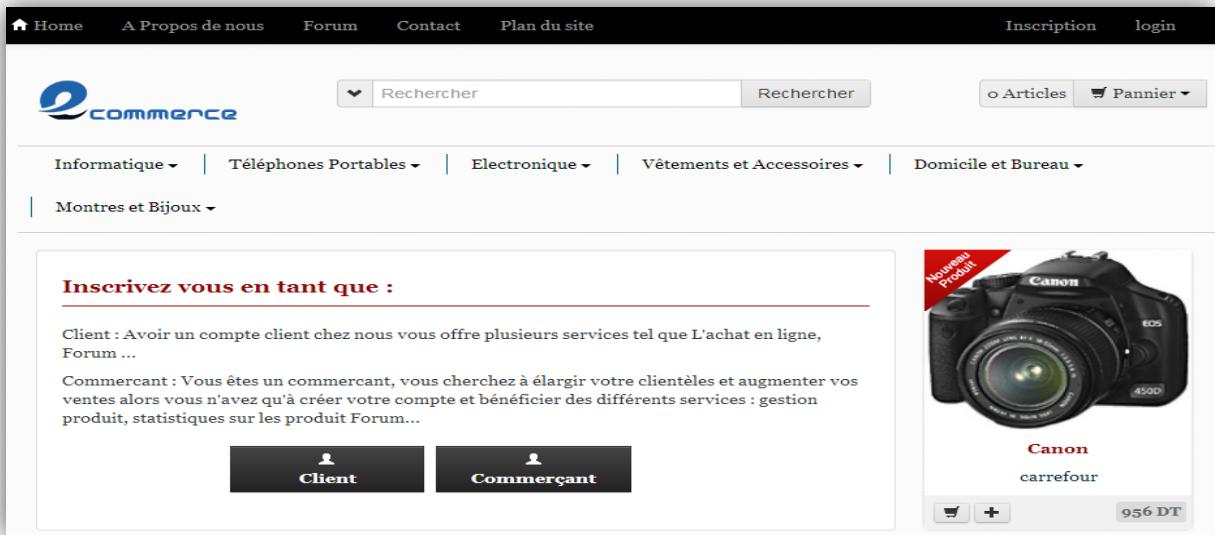


Figure 94 Interface « S'inscrire »

Prenant l'exemple d'un commerçant. En cliquant sur le bouton « commerçant » de la figure précédente, Un formulaire s'affiche contenant les données à insérer, comme montre la Figure 95 suivante :

Figure 95 Interface « formulaire d'inscription »

En validant l'inscription, les données saisies seront envoyés vers la méthode «SaveComm » du contrôleur «Inscription » pour enregistrer les données dans la base des données. Une fois enregistré un mail de confirmation sera envoyé au commerçant :

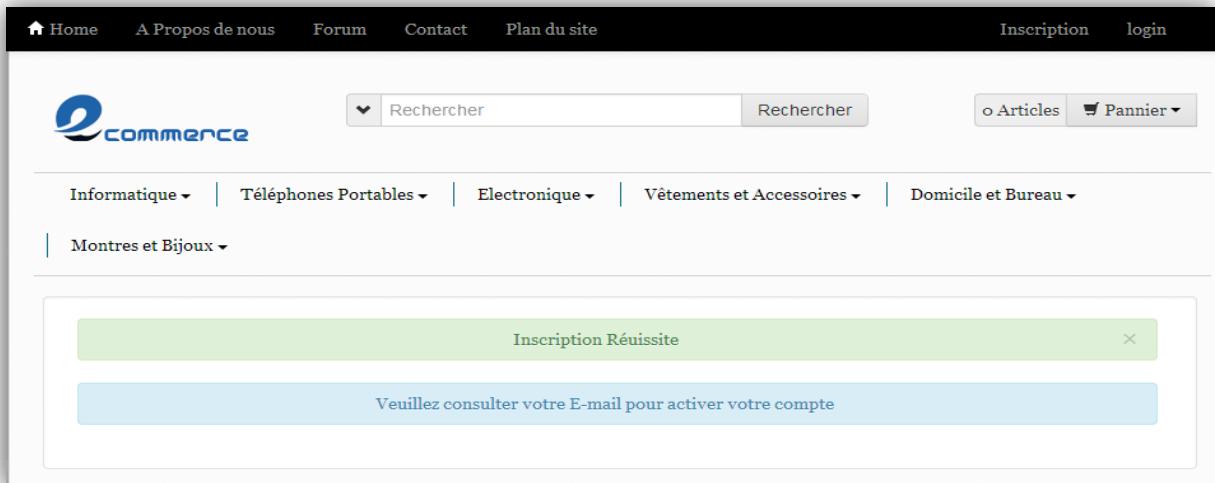


Figure 96 Interface « succès d'inscription»

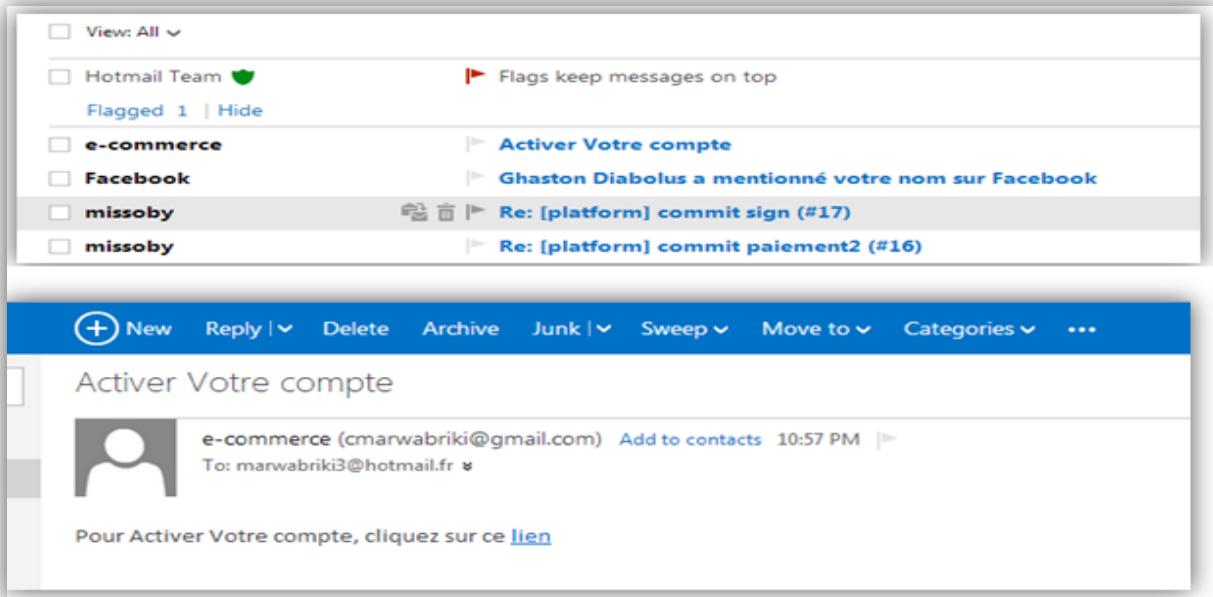


Figure 97 Interface « Mail de confirmation»

En cliquant sur le lien de la Figure 97, ce lien contient déjà en paramètre l'email du commerçant, haché et concaténé avec une chaîne de caractère. C'est pour vérifier que c'est la même personne qui va confirmer l'inscription. On récupère ce paramètre au niveau du contrôleur et on vérifie si c'est le même mail enregistré dans la base, si c'est le cas donc un champ booléen au niveau de la base sera mis à « true » et envoi d'une notification à l'administrateur pour l'informer qu'un nouveau commerçant vient d'être inscrit.

Un utilisateur a la possibilité de se connecter en utilisant les identifiants facebook, il doit être déjà connecté et l'inscription s'est faite avec le même email. Comme le montre la Figure 98 suivant :

Connectez Vous

Nom d'utilisateur
Votre Nom d'utilisateur

Mot de passe
Votre Mot de Passe

[Mot de passe oublier ?](#)

Se connecter

Ou Bien

Login avec facebook

Figure 98 Interface «connexion FB»

Une fois inscrit, chaque utilisateur a accès à son profil et a la possibilité de modifier ses propres informations, les consulter ou bien modifier son mot de passe. La Figure 99 suivante montre l'interface de gestion profil :

Home A Propos de nous Forum Contact Plan du site Messages Notifications hedivi

Informatique ▾ Téléphones Portables ▾ Electronique ▾ Vêtements et Accessoires ▾ Do... Voir Profil Modifier profil Modifier mot de passe Se Déconnecter

Informations Personnels

Nom *	ben ali	Adresse *	arianaupdate
Prénom *	hedi	Pays *	tunisia
E-mail *	pink9attoussa@live.fr	Ville *	nabeul
Login *	hedi	Numéro téléphone *	123456776
Société *	carrefour		

Figure 99 Interface «gestion profil»

Nous avons deux différents profils, le client et le commerçant, ces derniers ont des services en commun mais aussi chacun a des services spécifiques. Dans ce cadre nous présenterons deux différents scénarios. Le premier c'est le scénario commerçant ensuite le scénario client.

Scénario commerçant

En se connectant en tant que commerçant, nous serons redirigés vers l'interface du profil commerçant comme montre la Figure 100 suivante :

The screenshot shows the homepage of an e-commerce website. At the top, there is a navigation bar with links: Home, A Propos de nous, Forum, Contact, Plan du site, Messages, Notifications, and a user profile for 'hedi'. Below the navigation bar is a search bar with a dropdown menu and two buttons: 'Rechercher' and 'Rechercher'. Underneath the search bar is a horizontal menu with categories: Informatique, Téléphones Portables, Electronique, Vêtements et Accessoires, Domicile et Bureau, and Montres et Bijoux. The main content area has a header 'Votre Profil' with tabs for Suivi Vente, Statistique, and Gestion Commerce (which is currently selected). On the left side of the profile section, there is a table with personal information: Nom: ben ali, Ville: nabeul; Prénom: hedi, Adresse: arianaupdate; Login: hedi, Téléphone: 123456776; Pays: tunisia, Email: pink9attoussa@live.fr. On the right side, there is another table with company information: Societe: carrefour, Fax: 73245321; Adresse Société: hammamet, Site web: www.carrefour.tn. A sidebar on the right contains links: Gestion des produits, Gestion des annonces, and Gestion Slider.

Figure 100 Interface commerçant

A partir de la figure précédente, plusieurs services sont offerts tel que les statistiques, le suivi des ventes, la gestion des produits et annonces ainsi que les notifications et les messages. Nous allons traiter le plus important service offert à un commerçant c'est la gestion des produits.

Pour manipuler ses produits, un commerçant a accès à l'interface de la Figure 101 :

The screenshot shows a web application for managing products. At the top, there is a navigation bar with links for Home, A Propos de nous, Forum, Contact, and a user account section. Below the navigation is a search bar with a dropdown menu and a 'Rechercher' button. To the right of the search bar are links for Articles and Pannier. The main content area has a header 'Gestion des produits' and two tabs: 'Gestion des produits' and 'Gestion des annonces'. Below this is a button labeled 'Ajouter un produit'. The central part of the screen is a table with columns: Libelle, stock, prix, remise, active, and Actions. The table contains three rows of data: 'rouge à lèvres' (stock 54, price 123, discount 5, active), 'ordinateur' (stock 45, price 798, discount 12, active), and 'samsung' (stock 123, price 234, discount 0, active). Each row has a set of icons for edit, delete, activate, and deactivate. At the bottom of the table are navigation buttons for page 1, 2, and >.

Libelle	stock	prix	remise	active	Actions
rouge à lèvres	54	123	5	1	
ordinateur	45	798	12	1	
samsung	123	234	0	1	

Figure 101 Interface « gestion produit »

Comme le Framework utilisé est basé sur l'architecture MVC, pour cela chaque module nécessite ses propres Template (vue), contrôleurs ainsi que les modèles.

Chaque Template doit avoir une fonction (au niveau contrôleur) qui exécute un traitement et lui envoi le résultat, où il sera affiché. Toutefois, on peut créer des fonctions qui n'ont pas de Template associés. Généralement, ces méthodes exécutent des traitements en arrière-plan qui font appel à la couche modèle.

A partir de l'interface de la Figure 101, un commerçant a accès à trois autres interfaces qui sont : l'ajout d'un produit Figure 102, la modification d'un produit Figure 103, et l'affichage des détails d'un produit Figure 104.

Aussi un commerçant peut supprimer un produit ou bien le rendre visible par les clients ou pas, et ceci par l'action « activer » ou « désactiver ».

The screenshot shows a web-based form titled "Ajouter un produit". The form consists of several input fields and buttons. At the top, there is a dropdown menu labeled "Catégorie *". Below it are text input fields for "Libelle *", "Stock *", "Description *", "Prix *", and "Remise *". A file input field for "Photo *" contains the placeholder text "Choisissez un fichier" and "Aucun fichier choisi". At the bottom of the form are two buttons: "Valider" (in blue) and "Annuler" (in grey).

Figure 102 Interface « ajouter produit »

❖ Ce qui se passe en arrière plan

En cliquant sur « Ajouter produit », à partir de l'interface de Figure 101, la fonction « add » du contrôleur « produit » sera appelée. Cette dernière communique avec le modèle « produit_model » pour récupérer l'ensemble des catégories et les sous catégories. Ensuite elle passe le résultat à la vue appropriée.

Au niveau de la vue « ajoutProduit_view » un formulaire s'affiche, celui de la Figure 102. Puisque les sous catégories et les sous catégories dépendent de la sélection de la catégorie alors on a traité ce cas on utilisant JavaScript et Ajax.

On fait appel au fichier « categorie.js » qui contient une fonction anonyme Ajax qui récupère l'identifiant de l'option sélectionnée da la liste déroulante et selon ce dernier il affiche les sous catégorie correspondantes.

Une fois le remplissage du formulaire est terminé, la soumission envoi les données à la fonction « addproduct » du contrôleur « produit ». Cette dernière valide les données saisies, les envois au modèle qui les enregistre dans la base de données.

Modifier un produit

Photo *



Catégorie *

Libelle *

ordinateur

Stock *

45

Description *

dell inspiron

Figure 103 Interface « modifier produit »

❖ Ce qui se passe en arrière plan

La même démarche que celle d'ajout produit se passe au niveau de la modification. La seule différence c'est lorsqu'on appelle la vue pour afficher le formulaire, on lui passe les données concernant le produit indiqué. La récupération des données se fait par appel d'une méthode au niveau du contrôleur, qui passe ces données à la méthode appropriée du modèle qui s'occupera de la mise à jour au niveau base de données.

Détails d'un produit



Catégorie : Informatique **Stock :** 45
Sous Catégorie : Ordinateurs **Prix :** 798
Sous Sous Catégorie : Ordinateurs de bureau **Remise :** 12
Nom Produit : ordinateur

Description : dell inspiron

Figure 104 Interface « Détail produit Commerçant »

❖ Ce qui se passe en arrière plan

Tout se passe au niveau contrôleur et modèle. On récupère les données concernant le produit et on les passe au vue.

Scénario client

Un client lorsqu'il se connecte sera redirigé vers l'interface d'accueil de la plateforme ou il peut effectuer différentes actions tel que la recherche des produits, l'ajout produits au panier...

- La recherche

La rubrique de recherche est accessible tout au long de la navigation de la plateforme, pour cette raison l'interface de la recherche a été mise au niveau de l'interface principal de l'application appelé « layout ». La recherche peut se faire selon trois critères, soit par mot clés, par catégorie ou bien les deux ensembles comme illustré dans la Figure 105 suivante :

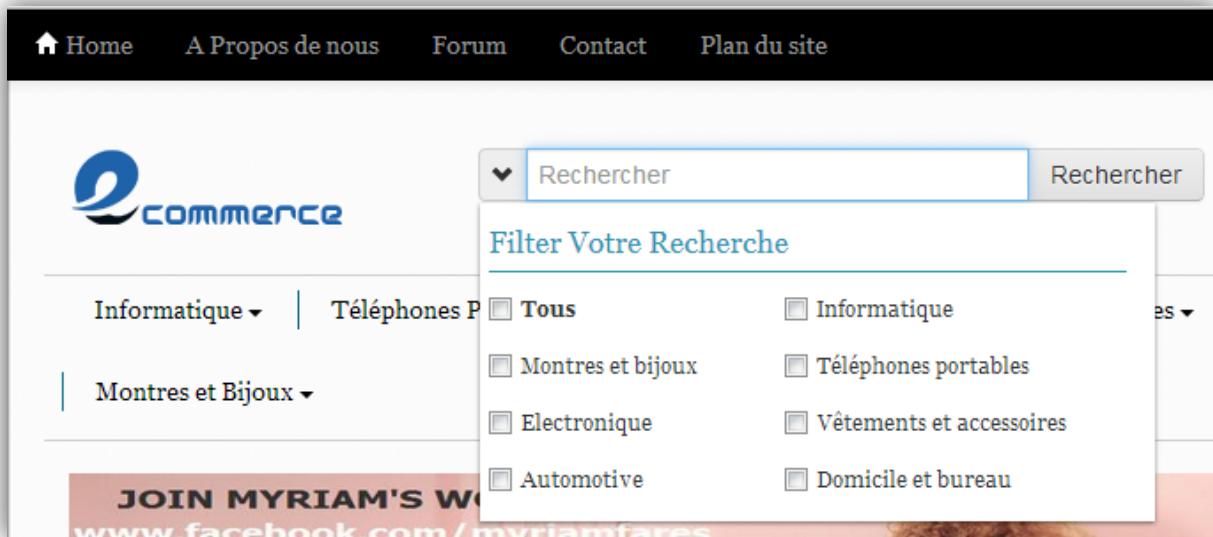


Figure 105 Interface « Recherche »

❖ Ce qui se passe en arrière plan

Comme les autres modules, nous avons préparé un contrôleur « recherche » dans lequel on récupère le critère saisi par l'utilisateur, ce dernier fait appelle aux vues spécifiques à la recherche pour l'affichage des résultats, ainsi que le contrôleur communique avec le modèle « recherche_model » pour récupérer les données nécessaires à partir de la base de données.

Le résultat de la recherche c'est une liste de produit, au niveau de laquelle un utilisateur peut ajouter ce dernier au panier, lui attribuer une note, le signaler, envoyer un message au propriétaire ou bien partager le produit en question sur Facebook. La Figure 106 suivante montre bien les services déjà cité :



Figure 106 Interface « Détails produit »

❖ Ce qui se passe en arrière plan

L'interface de la Figure 106 communique avec différents contrôleur selon l'action faite. Pour l'attribution d'une note à un produit, dès le clique de l'utilisateur sur les étoiles, un script javascript sera appelé qui récupère l'identifiant du produit et la note attribuée, ces deux paramètre seront passé à la fonction ajax qui est en relation avec le contrôleur « note », au niveau de ce dernier la mise à jour de la note du produit sera faite par le biais du modèle « note_model ». La nouvelle note sera renvoyé à la fonction ajax qui s'occupe de l'affichage de cette dernière au niveau de la même vue comme montre la figure suivante Figure 107.



Figure 107 Interface « note produit »

Une autre fonctionnalité intéressante c'est le partage Facebook par le biais du bouton « partager », cette dernière mais en action la méthode « publierFB » du contrôleur « afficheproduit ».

Si l'utilisateur n'est pas connecté à facebook l'interface de la Figure 108 sera affichée, sinon les informations nécessaires seront afficher sur la page facebook à l'aide de l'API utilisé lors de développement comme montre la Figure 109.



Figure 108 Interface « erreur facebook »



Figure 109 Interface « partage facebook »

- La gestion panier

L'achat en ligne c'est l'un des services offerts par la plateforme, pour cela nous pourrons ajouter un produit à notre panier comme illustré au niveau de la Figure 110 .



Figure 110 Interface « Ajouter au panier»

La gestion du panier est prise en charge par le contrôleur « cart », qui fait appelle aux différents vues et modèles en relation. Deux interfaces sont accessibles pour gérer son panier tel que montre la Figure 111 suivante :

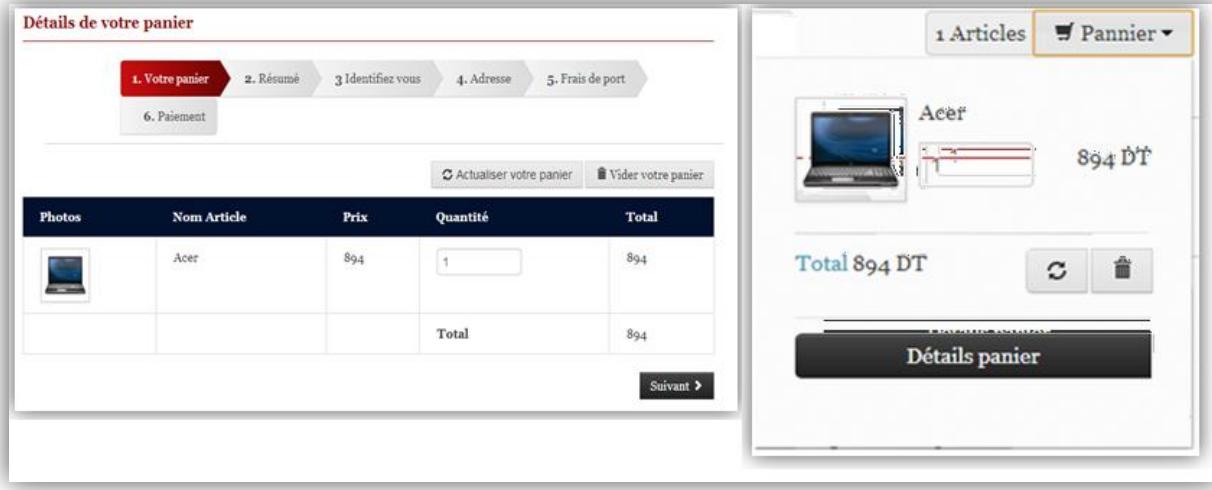


Figure 111 Interface « Détails panier»

❖ Ce qui se passe en arrière plan

Un client peut voir les détails de ses ajouts au panier en faisant appel à la méthode du contrôleur « index ».

Pour ajouter un produit au panier, la méthode « add_cart_item » du contrôleur « cart » sera appelée. Elle appelle par suite la méthode « validate_add_cart_item » du modèle. Cette dernière récupère l'identifiant et la quantité du produit ajouté et les insère dans la cart en utilisant la méthode prédéfinie « insert » de la librairie « cart ».

La mise à jour du panier nécessite l'appel de la méthode contrôleur « update_cart ». Elle communique avec la fonction du modèle « validate_update_cart », qui utilise la fonction prédéfinie « update » de la librairie « cart ».

Une fois connecté, un utilisateur constate qu'il existe différents moyen d'interaction entre internautes au niveau de la plateforme. Citons le forum ainsi que les notifications qui seront détaillé tout de suite.

Scénario Forum

Le module forum, de la Figure 112, est un module en commun pour tous les utilisateurs de la plateforme. Il suffit juste d'être connecté pour pouvoir accéder à cette interface d'interaction.

Les actions possibles par ce module :

- ✓ Ajout d'un sujet par catégorie
- ✓ Voir l'ensemble des sujets et commentaires
- ✓ Commenter un sujet



Figure 112 Interface « Espace forum »

Pour ajouter un sujet, il suffit de choisir une catégorie, par suite une interface s'affiche qui contient l'ensemble des sujets disponibles et la possibilité d'ajouter un sujet.

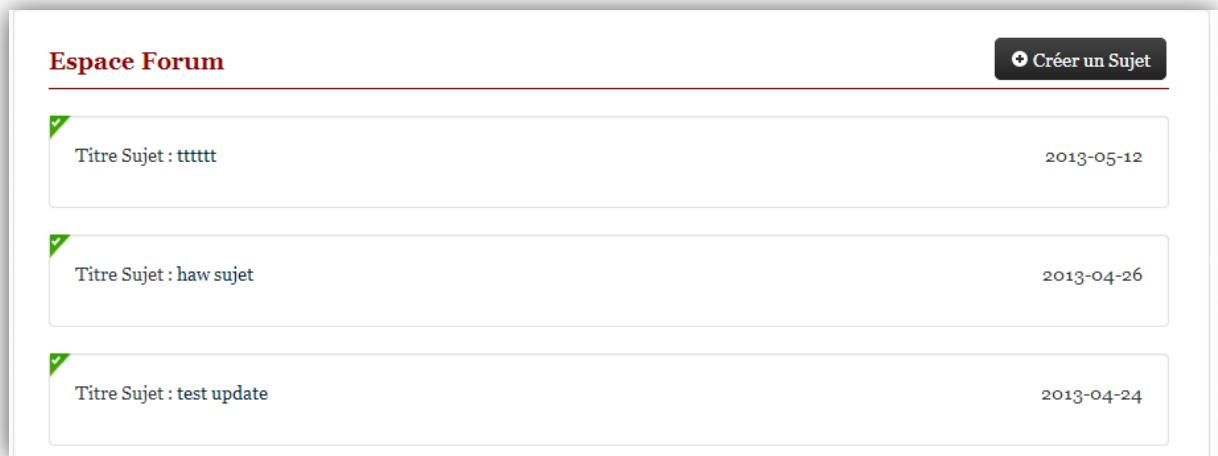


Figure 113 Interface « Ajouter sujet»

Pour ajouter un commentaire, il faut choisir un sujet, une interface contenant l'ensemble des commentaires disponible s'affiche avec un champ pour enregistrer un commentaire.

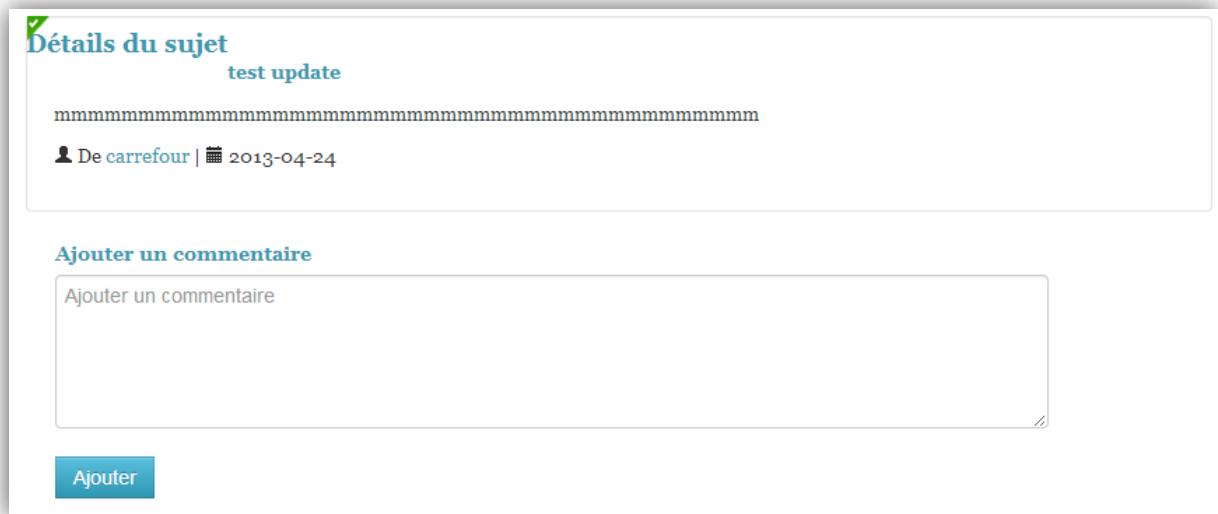


Figure 114 Interface « Ajouter commentaire »

Scénario Notification

L'aspect dynamique de la plateforme se renforce par le biais des notifications. Ces dernières se sont utilisées pour différents besoins, tel que notifier un utilisateur que son sujet sur le forum a été commenté, informer un utilisateur de la réception d'un message...



Figure 115 Interface « Notification »

❖ Ce qui se passe en arrière plan

Ces notifications dépendent de l'état de certains champs dans la base de données et non pas d'une action produite par l'utilisateur. Pour cette raison on a eu recours à l'utilisation de la technologie AJAX pour arriver à notre objectif.

On a intégré un fichier « notif.js » contenant une fonction AJAX qui s'exécute toute les six secondes et vérifie l'état de certains champ de la base de données par le biais d'une méthode du modèle « notif_model », si le contenu de ces champ est supérieur a zéro cela signifie qu'il existe une notification donc on change le style affiché.

IV. Test et validation

Les tests de vérification ou de validation visent à s'assurer que ce système réagit de la façon prévue par ses concepteurs (spécifications) ou est conforme aux attentes du client l'ayant commandé (besoins), respectivement.

Dans ce cadre on a effectué différents tests pour notre application, commençant par ceux concernant la compatibilité sur différents navigateurs :

- Firefox
- Safari
- Google Chrome version 11.

Gestion de la base de données

Action	Scénario de test	Résultat attendu	Résultat obtenu
Création de la base de données	Après avoir conçu la base de données en utilisant workbench, générer le code SQL et l'exécuter sous MySQL	Génération de toutes les tables avec succès, en vérifie par accès directe à la base de données sous l'interface phpMyadmin	Ok
Connexion à la Base de données	Accéder a la page d'accueil de l'application, et affichage des données à partir de l'application	si la connexion a échoué un message d'erreur s'affiche sinon la page d'accueil s'affiche correctement. Pouvoir interroger la base	Ok

Tableau 24 Test Gestion de la base de données

Module d'inscription et authentification

Action	Scénario de test	Résultat attendu	Résultat obtenu
Inscription client	Scénario Nominal : 1) Choisir de s'inscrire en tant que client	Scénario nominal : <ul style="list-style-type: none"> • Le profil du nouveau client est enregistré dans la base 	OK

	<p>2) Remplir le formulaire par les informations nécessaires et valider</p> <p>3) Recevoir le mail de confirmation</p> <p>4) Confirmation d'inscription en cliquant sur le mail reçu</p> <p>Scénario alternatif :</p> <ul style="list-style-type: none"> • Tester les contrôles de saisies sur les champs • Adresse mail doit être valide • Login et adresse mail doivent être unique • Tous les champs sont obligatoires 	<p>de données</p> <p>Scénario alternatif</p> <ul style="list-style-type: none"> • Aucun envoie des données vers la base est fait en cas d'un champ non valide • Tous les champs doivent afficher un message d'erreur 	
Inscription commerçant	Les mêmes scénarios nominaux et alternatifs	Les mêmes résultats attendus pour un client sauf qu'un commerçant ne peut pas se connecter que lorsque l'administrateur lui active le profil	OK
Authentification	<p>Scénario Nominal :</p> <ul style="list-style-type: none"> • Choisir de se connecter • Saisir le nom utilisateur et le mot de passe • Un utilisateur peut activer 	<p>Scénario Nominal :</p> <ul style="list-style-type: none"> • Connexion réussie et redirection selon le type utilisateurs (client, commerçant) 	OK

	<p>la fonctionnalité « se souvenir de moi »</p> <ul style="list-style-type: none"> • Valider <p>Scénario Nominal 2 :</p> <ul style="list-style-type: none"> • En cas de mot de passer oublier, cliquer sur le lien disponible « mot de passe oublier » • Entrer l'Email et récupérer le nouveau mot de passe <p>Scénario alternatif :</p> <ul style="list-style-type: none"> • Valider avec des champs vides • Tester avec un nom utilisateur et/ou mot de passe inexistant 	<ul style="list-style-type: none"> • Le nouveau client ne peut pas se connecter tant qu'il n'a pas confirmé son compte par mail • Une fois le compte confirmé, le client peut se connecter <p>Scénario Nominal 2 :</p> <ul style="list-style-type: none"> • Recevoir un mail pour modification de mot de passe. • Une fois mail reçu, redirection vers une page pour saisir le nouveau mot de passe. <p>Scénario alternatif :</p> <ul style="list-style-type: none"> • Des messages d'erreur doivent s'afficher en cas de champs vide. • Un message d'erreur doit s'afficher en cas d'entrées nom existant dans la base. 	
--	--	--	--

Tableau 25 Test Inscription et authentification

Module Gestion produit

Action	Scénario de test	Résultat attendu	Résultat obtenu
Ajouter un produit	<p>Scénario Nominal :</p> <ul style="list-style-type: none"> • Choisir d'ajouter un produit • Insérer les 	<p>Scénario Nominal :</p> <ul style="list-style-type: none"> • Le produit doit être bien enregistré dans la base de données. 	Ok

	<p>informations concernant un produit</p> <ul style="list-style-type: none"> • Valider <p>Scénario alternatif :</p> <ul style="list-style-type: none"> • Valider avec des champs vides. • Les champs stock, prix et remise ne doivent pas accepter des chaînes de caractères. 	<ul style="list-style-type: none"> • Le produit doit s'afficher dans la page de gestion produit. 	
Modifier produit	<p>Scénario Nominal :</p> <ul style="list-style-type: none"> • Choisir un produit et sélectionner l'option « modifier ». • Spécifier les nouvelles valeurs • Sauvegarder les mises à jour <p>Scénario alternatif :</p> <ul style="list-style-type: none"> • Valider avec des champs vides. 	<p>Scénario Nominal :</p> <ul style="list-style-type: none"> • Tous les attributs spécifiés au moment d'ajout du produit sont récupérés correctement et figurant dans l'interface graphique • Les informations du produit seront mises à jour dans la base de données. <p>Scénario alternatif :</p> <ul style="list-style-type: none"> • Réinitialiser les champs avec les anciennes valeurs 	OK
Supprimer produit	<ul style="list-style-type: none"> • Choisir le produit • Sélectionner l'option « supprimer » • Confirmer la demande de suppression 	<ul style="list-style-type: none"> • Affichage d'un message pour confirmer la suppression • Une fois confirmé, il y aura un message 	OK

		d'erreur ou bien le produit sera supprimer de la base de données	
Activer/désactiver produit	<ul style="list-style-type: none"> • Choisir le produit à activer/désactiver • Cliquer sur l'option souhaitée 	<ul style="list-style-type: none"> • Changement du champ « active » du produit selon l'option sélectionnée. 	Ok
Voir détails produit	<ul style="list-style-type: none"> • Choisir le produit • Cliquer sur l'option « voir détails » 	<ul style="list-style-type: none"> • Les informations concernant le produit s'affichent 	Ok

Tableau 26 Test Gestion produit

Module Gestion profil

Action	Scénario de test	Résultat attendu	Résultat obtenu
Voir profil	<ul style="list-style-type: none"> • Se connecter • Cliquer sur « voir profil » 	<ul style="list-style-type: none"> • Les informations concernant l'utilisateur connecté s'affichent 	OK
Modifier profil	<p>Scénario Nominal :</p> <ul style="list-style-type: none"> • Cliquer sur « modifier profil ». • Spécifier les nouvelles valeurs • Sauvegarder les mises à jour <p>Scénario alternatif :</p> <ul style="list-style-type: none"> • Les informations de 	<p>Scénario Nominal :</p> <ul style="list-style-type: none"> • Tous les attributs spécifiés au moment de l'inscription sont récupérés et ils s'affichent correctement. 	OK

	<ul style="list-style-type: none"> • Valider avec des champs vides. • Spécifier un nom utilisateur ou mail déjà existant 	<p>l'utilisateur seront mises à jour dans la base de données.</p> <p>Scénario alternatif :</p> <ul style="list-style-type: none"> • Réinitialiser les champs avec les anciennes valeurs et affichage des messages d'erreur • Afficher un message d'erreur spécifiant l'existence du mail ou nom utilisateur 	
Modifier mot de passe	<p>Scénario Nominal :</p> <ul style="list-style-type: none"> • Entrer le mot de passe courant • Entrer le nouveau mot de passe et sa confirmation <p>Scénario alternatif :</p> <ul style="list-style-type: none"> • Entrer un mot de passe courant incorrect • Le nouveau mot de passe et sa confirmation ne sont pas identiques. 	<p>Scénario Nominal :</p> <ul style="list-style-type: none"> • Mot de passe modifié avec succès. <p>Scénario alternatif :</p> <ul style="list-style-type: none"> • Message d'erreur du mot de passe courant incorrecte • Message d'erreur du nouveau mot de passe et sa confirmation non identique 	OK

Tableau 27 Test Gestion Profil

Module Noter produit

Action	Scénario de test	Résultat attendu	Résultat

			obtenu
Attribuer une note	<ul style="list-style-type: none"> • Choisir un produit • Afficher son détail • Attribuer une note 	<ul style="list-style-type: none"> • L'affichage de la note attribuée par l'utilisateur. • Une fois noté, la nouvelle note du produit s'affiche. 	OK

Tableau 28 Test Noter produit

Module Envoi message

Action	Scénario de test	Résultat attendu	Résultat obtenu
Envoyer message	Scénario Nominal : <ul style="list-style-type: none"> → <u>Pour le client</u> <ul style="list-style-type: none"> • Voir détail produit • Envoyer Message → <u>Pour le commerçant</u> <ul style="list-style-type: none"> • Dans son interface : • Il accède à l'ensemble des messages reçus de la part du client ou de l'administrateur. • Il peut aussi répondre à un message reçu • Il accède à l'ensemble des messages qu'il a envoyé 	Scénario Nominal : <ul style="list-style-type: none"> → <u>Pour le client</u> <ul style="list-style-type: none"> Message envoyé et redirection vers la page d'affichage des différents produits → <u>Pour le commerçant</u> <ul style="list-style-type: none"> • Affichage de l'ensemble des messages reçus avec une interface pour répondre à chaque message • Affichage de l'ensemble des messages envoyés avec une interface pour répondre à chaque message. 	OK

Tableau 29 Test Envoi message

Module Forum

Action	Scénario de test	Résultat attendu	Résultat obtenu
Création sujet	<ul style="list-style-type: none"> • Un client/commerçant doit se connecter pour pouvoir accéder à l'espace forum. <p>Scénario Nominal 1:</p> <ul style="list-style-type: none"> • Choisir une catégorie • Il sélectionne l'option « ajouter sujet ». • Une interface s'affiche pour saisir les données concernant le sujet à ajouter. <p>Scénario Nominal 2:</p> <ul style="list-style-type: none"> • Accéder à un sujet. • L'ensemble des commentaires concernant ce dernier s'affichent • L'utilisateur peut ajouter des commentaires. 	<p>Scénario Nominal1 :</p> <ul style="list-style-type: none"> • Le sujet ajouté s'enregistre dans la base de données et s'affiche au niveau de la catégorie en question. <p>Scénario Nominal 2:</p> <ul style="list-style-type: none"> • Le commentaire ajouté s'enregistre dans la base • Il s'affiche au niveau de l'interface d'affichage du sujet concerné 	Ok OK

Tableau 30 Test Forum

