

Ministère de l'Enseignement Supérieur,  
de la Recherche Scientifique  
Université de Monastir

\*\*\*\*\*

Institut Supérieur d'Informatique et de Mathématiques de Monastir



## Projet de Fin d'Etudes

En vue de l'obtention du

Diplôme de licence Appliquée en Informatique  
Systèmes Informatiques et Logiciels

Sujet

Conception et Développement d'une solution Web et Mobile pour le  
compte de la société Medor

Réalisé par

MANSOUR Hosni  
TAOUS Alaa

Sous la direction de

Dr.MAHMOUDI Ramzi (ISIMM)  
Mlle.BRAHEM Wafa (MediaPlus)

Année universitaire : 2015/2016

# Dédicaces

Du profond de mon cœur, je dédie ce travail à tous ceux qui me sont chers,

## À Mes Très Chers Parents

Que ce travail soit l'expression de ma reconnaissance pour vos sacrifices consentis, votre soutien moral et matériel que vous n'avez cesse de prodiguer. Vous avez tout fait pour mon bonheur et ma réussite.

Que Dieu vous préserve en bonne santé et vous accorde une longue vie.

## À Mon Frere et Mes sœurs

Vous étiez toujours présents pour m'aider et m'encourager. Je vous souhaite une vie pleine de bonheur et de succès et que Dieu vous protège et vous garde.

## À Tous mes amis...

*Hosni Mansour*

---

J'aimerai dédier ce rapport,

## À mes Parents,

qui m'ont comblé de leur soutien et m'ont voué un amour inconditionnel. Vous êtes pour moi un exemple de courage et de sacrifice continu.

Que cet humble travail témoigne mon affection, mon éternel attachement et qu'il appelle sur moi vos continues bénédictions.

## À mes frères et soeurs, à mes copains et à toutes la promo à l'ISIMM,

pour leurs compréhensions, leurs soutiens, leurs tendresses... Qu'ils trouvent ici l'expression de ma reconnaissance et le témoignage de ma gratitude ressentie.

## À ceux qui m'aiment,

Que ce modeste travail vous honore et vous témoigne mes reconnaissances.

Puisse cette étude apporter la pleine satisfaction à tous ceux qui la lisent...

Mille Merci.

*Alaa Taous*

# **Remerciement**

En préambule à ce rapport, nous souhaitons adresser ici tous nos remerciements aux personnes qui nous ont apporté leur aide et qui ont ainsi contribué à l'élaboration de ce rapport.

Nous tenons à remercier dans un premier temps, toute l'équipe pédagogique de l'ISIMM et les intervenants professionnels responsables de la formation sciences de l'informatique, pour avoir assuré la partie théorique de celle-ci.

Nous tenons à exprimer également notre profonde reconnaissance à Dr Mahmoudi Ramzi qui nous a encadrés durant ce projet de fin d'études, pour l'aide et les conseils concernant les missions évoquées dans ce projet, qu'elle nous a apporté lors des différents suivis.

Nous tenons à remercier tout particulièrement et à témoigner toute notre reconnaissance à Mlle Brahem Wafa pour l'aide et la confiance qu'elle nous a accordés durant la période du stage.

Nous exprimons notre sincère gratitude aux membres du jury, qui ont accepté d'évaluer ce projet de fin d'études.

Enfin, nous adressons nos plus sincères remerciements à tous nos proches et nos amis qui nous ont toujours soutenu et encouragé au cours de la réalisation de ce projet de fin d'études.

# Résumé

---

L'objectif principal de ce travail de PFE consiste à concevoir et développer une application web et mobile pour le compte de la société Medor.

Cette application permet aux internautes de consulter les produits et les services de la société cliente afin de trouver le produit adéquat pour l'acheter. L'application inclut aussi un service « Back Office » pour la partie web qui permet au personnel administratif de la société de gérer les clients, les produits, les informations et paramètres de site, aussi de consulter les commandes, les demandes devis et les messages des clients.

La partie web est développé sous le Framework Symfony2. Tandis que la partie mobile est développée avec Android SDK en utilisant les services web.

À travers ce document, nous allons décrire en détail les différentes étapes de réalisation de ce projet.

**Mots clés :** PHP, Symfony2, JQuery ,BackOffice ,Service Web, Android, Java.

## Abstract

---

This final of studies project aims to design and develop a web and a mobile application for Medor company.

This application allows users to view the company products and services in order to find the right product to buy. The application also includes a "Back End" service for the web part which allows company's administrative staff to manage customers, products, informations and website parameters, also check client's orders and messages.

The web part is developed with the Framework Symfony2. While the mobile part is developed with Android SDK using web services.

Through this document, we describe in detail the various stages of this project.

**Keywords :** PHP, Symfony2, JQuery ,Backend ,Web Service, Android, Java.

# Table des matières

<b>Dédicace</b>	i
<b>Remerciment</b>	ii
<b>Resumé</b>	iii
<b>Table des matières</b>	vi
<b>Liste des figures</b>	vii
<b>Liste des tableaux</b>	ix
<b>Introduction générale</b>	1
<b>Chapitre I Contexte et objectifs</b>	2
Introduction . . . . .	3
1 Positionnement et problématique . . . . .	3
1.1 Contexte général . . . . .	3
1.2 Problématique . . . . .	3
2 Etude de l'existant . . . . .	4
2.1 Ebay . . . . .	4
2.2 Kaymu . . . . .	5
2.3 Tayara . . . . .	5
2.4 Tableau comparatif . . . . .	6
3 Solution proposée . . . . .	7
4 Méthodologie de travail et modélisation . . . . .	9
4.1 Le cycle de vie d'un logiciel . . . . .	9
4.2 Processus simplifiés et modernes pour les applications web . . . . .	11
4.2.1 Méthodes d'analyse et de conception . . . . .	11
4.2.2 Méthodes Agile ou Processus Unifié ? . . . . .	11
4.2.2.1 La méthode Agiles . . . . .	11
4.2.2.2 Le Processus Unifié (UP) . . . . .	13
4.3 Mise en pratique du processus 2TUP . . . . .	14
4.4 Diagramme de gantt . . . . .	15
Conclusion . . . . .	17
<b>Chapitre II Analyse et capture des besoins</b>	18
Introduction . . . . .	19
1 Étude préliminaire . . . . .	19
1.1 Medico-Chirurgical . . . . .	19
1.2 El Faraj . . . . .	20
1.3 Locamed . . . . .	20
1.4 Geant Medical . . . . .	21

1.5	Sofamed . . . . .	22
1.6	Domital . . . . .	22
1.7	Tableau comparatif . . . . .	23
2	Besoins fonctionnels . . . . .	25
3	Besoins non fonctionnels . . . . .	26
4	Besoins techniques . . . . .	27
4.1	Étude des outils de développement . . . . .	27
4.2	Étude des différents Framework . . . . .	29
	Conclusion . . . . .	30
<b>Chapitre III Conception</b>		<b>31</b>
	Introduction . . . . .	32
1	Conception générique . . . . .	32
1.1	Le langage UML . . . . .	32
1.2	Les diagrammes d'UML . . . . .	32
1.3	Modèles d'UML utilisés . . . . .	33
2	Conception préliminaire . . . . .	33
2.1	Différences entre un site web dynamique et statique . . . . .	33
2.2	Architecture de site web dynamique - le modèle MVC . . . . .	34
2.3	Symfony2 et le modèle MVC . . . . .	36
2.4	Conception préliminaire des interfaces - maquettes . . . . .	36
3	Conception détaillée . . . . .	38
3.1	Identification des acteurs . . . . .	38
3.2	Identification des cas d'utilisations . . . . .	38
3.2.1	Modèle global de cas d'utilisation . . . . .	38
3.2.2	Description détaillée des cas d'utilisation . . . . .	41
3.2.2.1	Cas d'utilisation « Créer compte » - « S'authentifier » . . . . .	41
3.2.2.2	Les cas d'utilisation consulter produits et gérer panier . . . . .	42
3.2.2.3	Le cas d'utilisation « Chercher un produit » . . . . .	43
3.2.2.4	Le cas d'utilisation « Gérer ses commandes » . . . . .	44
3.2.2.5	Le cas d'utilisation « Gérer son compte » . . . . .	45
3.2.2.6	Les cas d'utilisation gérer catégories et produits . . . . .	46
3.2.2.7	Le cas d'utilisation « Gérer utilisateurs » . . . . .	49
3.2.2.8	Les cas d'utilisation gérer messages, commandes et site . . . . .	50
3.3	Diagrammes de séquence système . . . . .	52
3.3.1	Diagrammes de séquences « Authentification » . . . . .	53
3.3.2	Diagrammes de séquences « Gérer panier » . . . . .	53
3.3.3	Diagrammes de séquences de « Gérer les produits » . . . . .	54
3.4	Diagramme de composant . . . . .	55
3.5	Diagramme de classe . . . . .	56
	Conclusion . . . . .	60
<b>Chapitre IV Réalisation</b>		<b>61</b>
	Introduction . . . . .	62
1	Environnement de développement . . . . .	62
1.1	Environnement matériel . . . . .	62
1.2	Environnement logiciel . . . . .	62
2	Présentation de la solution Web . . . . .	64
2.1	Interface d'accueil . . . . .	64
2.2	Interface catégorie . . . . .	65

2.3	Interface contact . . . . .	65
2.4	Interface produit détaillé . . . . .	66
2.5	Interface commande . . . . .	67
2.6	Interface backend . . . . .	67
2.7	Interfaces gestion des produits . . . . .	68
2.8	Interfaces gestion des utilisateurs . . . . .	70
2.9	Interfaces de la liste des commandes . . . . .	71
2.10	Interfaces de la liste des messages . . . . .	72
	Conclusion . . . . .	72
<b>Chapitre V Extension</b>		<b>74</b>
	Introduction . . . . .	75
1	Environnement de développement . . . . .	75
1.1	Environnement matériel . . . . .	75
1.2	Environnement logiciel . . . . .	75
2	Présentation de la solution mobile . . . . .	76
2.1	Interface d'accueil . . . . .	76
2.2	Interface menu . . . . .	76
2.3	Interface catégorie . . . . .	77
2.4	Interface produit . . . . .	77
2.5	Interface demande devis . . . . .	77
2.6	Interface contact . . . . .	78
	Conclusion . . . . .	78
<b>Conclusion générale</b>		<b>79</b>
<b>Annexes</b>		<b>80</b>
<b>Annexe A</b>	<b>Hébergement site web</b>	<b>81</b>
<b>Annexe B</b>	<b>Service web</b>	<b>84</b>
<b>Annexe C</b>	<b>Soumettre l'application sur Google Play</b>	<b>86</b>

# Liste des figures

I.1	Capture de site eBay . . . . .	4
I.2	Capture de site Kaymu . . . . .	5
I.3	Capture de site Tayara . . . . .	6
I.4	Illustration de la solution . . . . .	8
I.5	Les étapes du processus 2TUP . . . . .	14
I.6	Diagramme de Gantt . . . . .	16
II.1	Capture de site Medicochirurgical . . . . .	19
II.2	Capture de site Elfaraj . . . . .	20
II.3	Capture de site Locamed . . . . .	21
II.4	Capture de site Geantmedical . . . . .	21
II.5	Capture de site Sofamed . . . . .	22
II.6	Capture de site domital . . . . .	23
II.7	Les langages de programmation côté serveur les plus utilisés . . . . .	27
III.1	Architecture d'un site statique . . . . .	34
III.2	Architecture d'un site dynamique . . . . .	34
III.3	Schéma de l'architecture MVC . . . . .	35
III.4	Parcours complet d'une requête dans Symfony2 . . . . .	36
III.5	Maquette - Page Accueil . . . . .	36
III.6	Maquette - Page categorie . . . . .	37
III.7	Maquette - Page Contact . . . . .	37
III.8	Diagramme de cas d'utilisation globale . . . . .	39
III.9	Diagramme des cas d'utilisation « Créeer compte » - « S'authentifier » . . . . .	41
III.10	Diagramme des cas d'utilisation « Consulter les produits » . . . . .	42
III.11	Diagramme de cas d'utilisation « Chercher un produit » . . . . .	43
III.12	Diagramme de cas d'utilisation « Gérer ses Commandes » . . . . .	44
III.13	Diagramme de cas d'utilisation « Gérer son compte » . . . . .	45
III.14	Diagramme des cas d'utilisation « Gérer les catégories » - « Gérer les produits » . . . . .	46
III.15	Diagramme de cas d'utilisation « Gérer les Utilisateurs » . . . . .	49
III.16	Diagramme des cas d'utilisation gérer messages, commandes et site . . . . .	50
III.17	Diagrammes de séquences de « Authentification » . . . . .	53
III.18	Diagrammes de séquences de « Gérer panier » . . . . .	54
III.19	Diagrammes de séquences de « Gérer les produits » . . . . .	55
III.20	Diagramme de composant . . . . .	56
III.21	Organisation de code selon MVC . . . . .	57
III.22	Diagramme de classe « Model » . . . . .	58
III.23	Diagramme de classe « Controller » . . . . .	59
III.24	Diagramme de classe « View » . . . . .	60
IV.1	IntelliJ IDEA . . . . .	63
IV.2	Xampp . . . . .	63
IV.3	dhtmlxGantt . . . . .	63

IV.4	Texmaker . . . . .	63
IV.5	Balsamiq mockup . . . . .	63
IV.6	Gliffy . . . . .	63
IV.7	Interface d'accueil . . . . .	64
IV.8	Interface catégorie . . . . .	65
IV.9	Interface contact . . . . .	66
IV.10	Interface produit détaillé . . . . .	66
IV.11	Interface commande . . . . .	67
IV.12	Interface Backend . . . . .	68
IV.13	Interface liste des produits . . . . .	69
IV.14	Interface ajouter produit . . . . .	69
IV.15	Interface liste des utilisateurs . . . . .	70
IV.16	Interface modifier utilisateur . . . . .	70
IV.17	Interface liste des commandes . . . . .	71
IV.18	Interface afficher commande . . . . .	71
IV.19	Interface liste des messages . . . . .	72
IV.20	Interface afficher message . . . . .	72
V.1	Android Studio . . . . .	75
V.2	GenyMotion . . . . .	75
V.3	Advanced REST client . . . . .	75
V.4	Interface d'accueil application mobile . . . . .	76
V.5	Interface menu application mobile . . . . .	76
V.6	Interface catégorie application mobile . . . . .	77
V.7	Interface produit application mobile . . . . .	77
V.8	Interface demande devis application mobile . . . . .	78
V.9	Interface contact application mobile . . . . .	78
A.1	Logo arvixe . . . . .	81
A.2	Arvixe Cpanel . . . . .	81
A.3	Gestion DNS du domaine medor.tn . . . . .	82
A.4	FileZilla . . . . .	82
A.5	Configuration base de données . . . . .	83
A.6	Phpmyadmin . . . . .	83
B.1	Service web . . . . .	84
B.2	JSON . . . . .	84
B.3	Advanced REST client . . . . .	85
B.4	Implémentation Service web . . . . .	85
C.1	Logo Google play . . . . .	86
C.2	Dashboard Google play . . . . .	86
C.3	Soumettre l'application sur Google play . . . . .	87
C.4	Medor sur Google play . . . . .	87

## Liste des tableaux

I.1	Tableau comparatif des applications web existantes . . . . .	7
I.2	Tableau comparatif des cycles de vie de logiciel . . . . .	10
I.3	Tableau représentant les méthodes Agile: XP et Scrum . . . . .	12
I.4	Tableau représentant les implémentations du Processus Unifié: RUP et 2TUP . . . . .	13
II.1	Tableau comparatif des applications web similaires en Tunisie . . . . .	24
II.2	Tableau comparatif des applications web similaires à l'étranger . . . . .	24
II.3	Tableau comparatif des langages de programmation . . . . .	28
II.4	Tableau comparatif des différents Framework . . . . .	30
III.1	Tablau des cas d'utilisation . . . . .	40

# Introduction générale

Le monde d'aujourd'hui est témoin d'un progrès énorme dans les différents domaines et plus particulièrement dans le domaine de la technologie de l'information, cette progression remarquable pousse les sociétés et les boutiques à considérer les sites et les applications mobiles de commerce électronique comme un dispositif global fournissant aux clients un pont de passage à l'ensemble des informations, des produits, et des services d'une façon plus rapide et plus facile.

Le présent travail s'inscrit dans le cadre du projet de fin d'études en vue de l'obtention du diplôme de Licence Appliquée en Informatique de l'Institut Supérieur d'Informatique et Mathématiques de Monastir (ISIMM) pour l'année universitaire 2015/2016, dans lequel nous allons concevoir et implémenter une solution web et mobile pour le compte de la société Medor spécialisée dans la vente du matériel médical et orthopédique.

Notre projet est réalisé au sein de la société Media Plus, une société de développement web et mobile située à Monastir qui nous a suivis et encadrés tout au long de la durée de notre stage.

Le présent document est organisé comme suit :

Dans le premier chapitre, nous présentons notre projet en étudiant son cadre général et la problématique avant de critiquer les solutions existantes pour dégager leurs insuffisances et proposer les orientations de notre solution future. La présentation de la solution souhaitée nous mène à bien analyser les différentes méthodologies de travail existantes afin de dégager celle la plus adéquate à notre projet.

Le deuxième chapitre consiste à formaliser les étapes préliminaires du développement d'un système afin de rendre ce développement plus fidèle aux besoins du client qui sont divisés en besoins fonctionnels, besoins non fonctionnels et besoins techniques.

Le troisième chapitre détaille la partie analyse et conception et explique de manière approfondie les fonctionnalités de l'application.

Le quatrième chapitre présente l'environnement de développement matériel et logiciel que nous allons adopter pour la partie web et nous allons présenter quelques captures d'écrans du site.

Le cinquième chapitre présente l'environnement de développement matériel et logiciel que nous allons adopter pour la partie Mobile et nous allons présenter quelques captures d'écrans de l'application.

# Chapitre I

## Contexte et objectifs

# Introduction

Au cours de ce chapitre, nous allons introduire notre projet en étudiant son cadre général et la problématique qui a poussé l'organisme d'accueil à réaliser cette application. Nous allons critiquer les solutions existantes pour dégager leurs insuffisances et proposer les orientations de notre solution future.

La présentation de la solution souhaitée nous mène à bien analyser les différentes méthodologies de travail existantes afin de dégager celle la plus adéquate à notre projet.

## 1 Positionnement et problématique

### 1.1 Contexte général

Le monde d'aujourd'hui est caractérisé par une transformation digitale représentant l'un de ses plus importants divers. En perpétuelle évolution, nous sommes témoins d'une véritable révolution. En 2010, 5 milliards d'équipements connectés à Internet ont été répertoriés par une étude d'IMS Research et rapportée par Le Monde Informatique. Et plus de 30 milliards sont envisagés en 2020.

Cette révolution est désormais bien ancrée dans notre quotidien et son impact se fait de plus en plus fort sur la façon dont nous, consommateurs, nous approprions l'information, la publicité et donc les marques.

Au vu de ce constat et de cette croissance, on ne peut pas négliger le passage des priorités des opérations de vente de main en main vers des ventes virtuelles ce que nous oblige à donner plus d'importance à la vente électronique. Ce sont les boutiques électroniques qui accélèrent et facilitent les traitements traditionnels tels que les processus de commande et de facturation en offrant un accès facile et rapide aux produits. Le défi est alors de faire des internautes d'aujourd'hui nos futurs clients à travers une interface agréable et une plate-forme performante qui répondent aux besoins de client et lui facilitent la procédure d'achat des biens et des services.

Dans cette perspective, on trouve que les équipements médicaux occupent une place importante sur le marché de la santé et de la qualité de vie. Ces produits sont très sensibles et concernent un nombre maximum des gens partout, d'où vient l'importance du domaine de vente du matériel Médio-Chirurgical, Orthopédique et matériel kiné ce que nous poussent à bien réfléchir aux difficultés qu'un client trouve en se déplaçant entre plusieurs boutiques pour trouver une solution satisfaisante. Ces déplacements peuvent être inutiles et même provoquent un gaspillage de temps car les produits sont délicats, multiples et chers.

### 1.2 Problématique

Dans le cadre de cette vision, la société «**Medor**», spécialisée dans la vente et la distribution des dispositifs Médicaux à usage unique, de matériels Orthopédiques, équipements Cliniques et Hôpitaux a pensé à automatiser son système de vente d'une manière qui permet d'informer un grand nombre de clients des produits disponibles et des services offerts en abandonnant les supports traditionnels tels que les annonces dans les journaux et les petites affiches.

La mission qui nous a été confiée au sein de «**Media Plus**», une société de Développement Web et Mobile située à Monastir, est la réalisation d'une application Web et Mobile pour la

société «**Medor**». Ce travail a pour objectif de présenter les produits de la société cliente à travers une interface web et mobile qui non seulement présente les services et les produits selon leurs catégories et types, mais elle offre également la possibilité de demande de devis et de contact par formulaire ou par Mail. L'application inclut ainsi un service Back Office : C'est un espace administrateur permettant la gestion de pages, catégories et de produits et leurs disponibilités, ainsi que la gestion de clients.

## 2 Etude de l'existant

La réalisation de tout projet doit être précédée par une étude de l'existant qui détermine les points faibles et les points forts des systèmes actuels et les besoins du client en vue de les prendre en considération lors de la conception et de la réalisation. Dans cette section nous mène alors à une étude basée sur les observations de différentes applications web spécialisées dans le domaine de vente électronique. Cette étude nous permet de dégager et donner leurs atouts et leurs faiblesses afin de déterminer les besoins et les traiter.

### 2.1 Ebay

eBay est une entreprise américaine de courtage en ligne. Toute personne inscrite sur le site eBay de son pays peut participer aux enchères de tous les pays. Une fois que l'inscription est validée, l'internaute peut :

**Acheter** : après vérification de l'adresse électronique indiquée à l'inscription.

**Vendre** : après vérification de l'identité du membre par courrier postal ou identification.

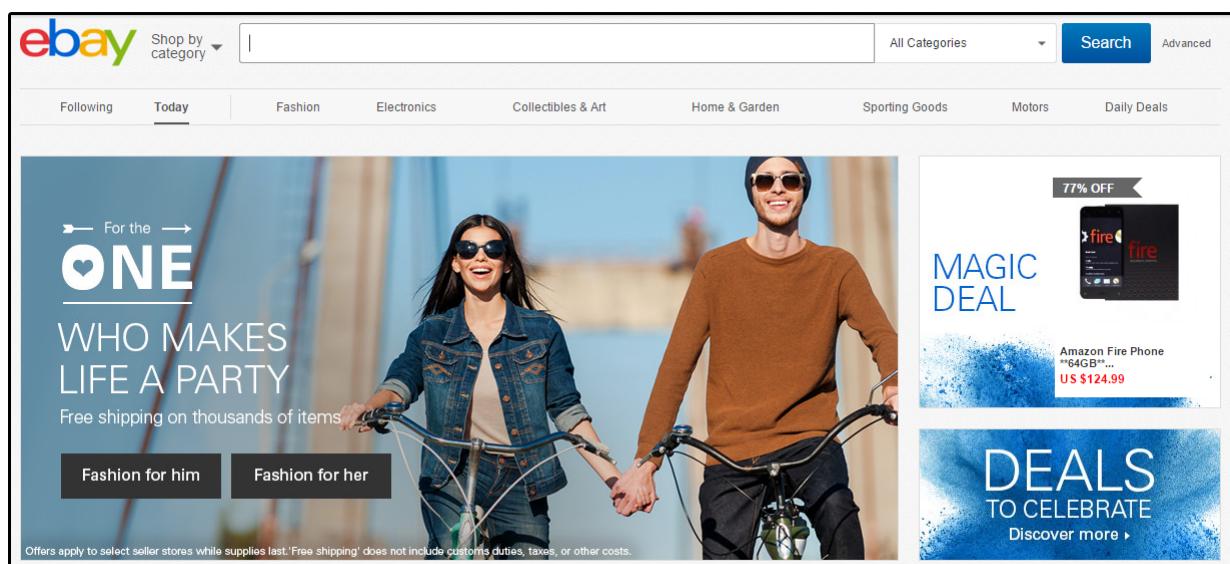


Figure I.1: Capture de site eBay

**Les points forts :**

- Le vendeur peut construire une boutique en ligne au sein de site eBay pour vendre ses produits.
- Paiement facile et sécurisé avec Paypal.

- eBay est une grande entreprise mondiale spécialisée en commerce électronique donc on évite les problèmes qui concernent l'hébergement et les performances du site...

#### Les points faibles :

- Il faut payer une commission sur le prix final et des frais d'insertion et de mise en vente.
- En Tunisie on a des problèmes avec les cartes bancaires internationales donc l'achat et la vente des produits sur des sites internationaux comme eBay n'est pas encore facile.

## 2.2 Kaymu

Kaymu est le premier site de vente en ligne en Tunisie, la plate-forme permet de mettre en relation les clients et les vendeurs dans le but de réaliser de très bonnes affaires tout en bénéficiant de l'expertise Kaymu.

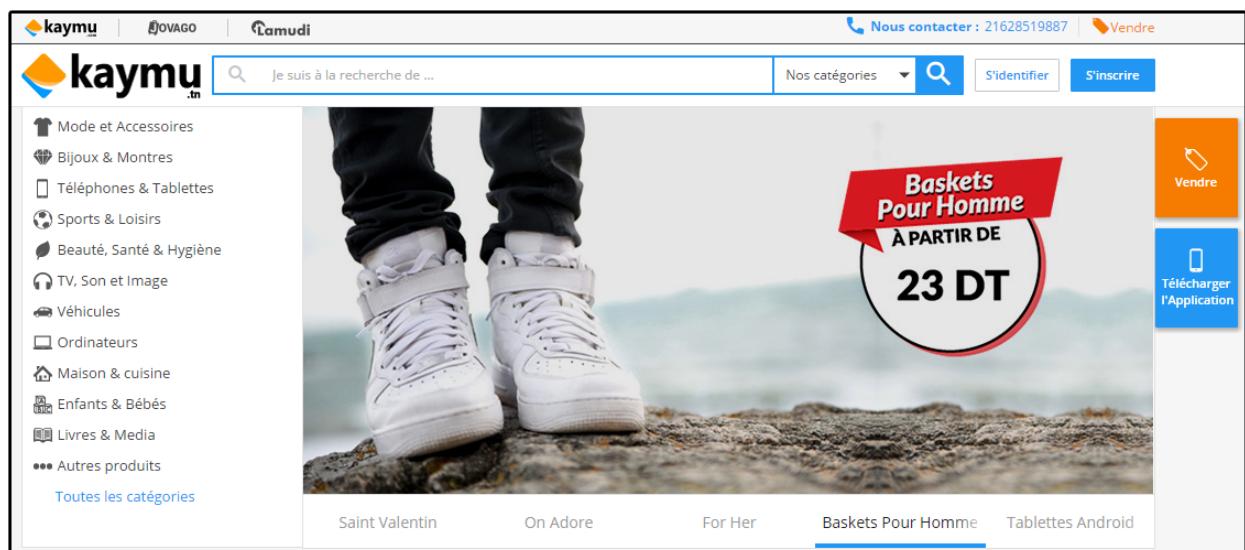


Figure I.2: Capture de site Kaymu

#### Les points forts :

- Le vendeur peut construire une boutique en ligne au sein de site Kaymu pour vendre ses produits.
- Kaymu organise les procédures du paiement, livraison...

#### Les points faibles :

- Il faut payer les frais de gestion des services de Kaymu.tn

## 2.3 Tayara

Tayara est un site des annonces en Tunisie qui offre aux internautes un espace pour poster des annonces gratuitement dans le but de vendre et d'acheter des produits...

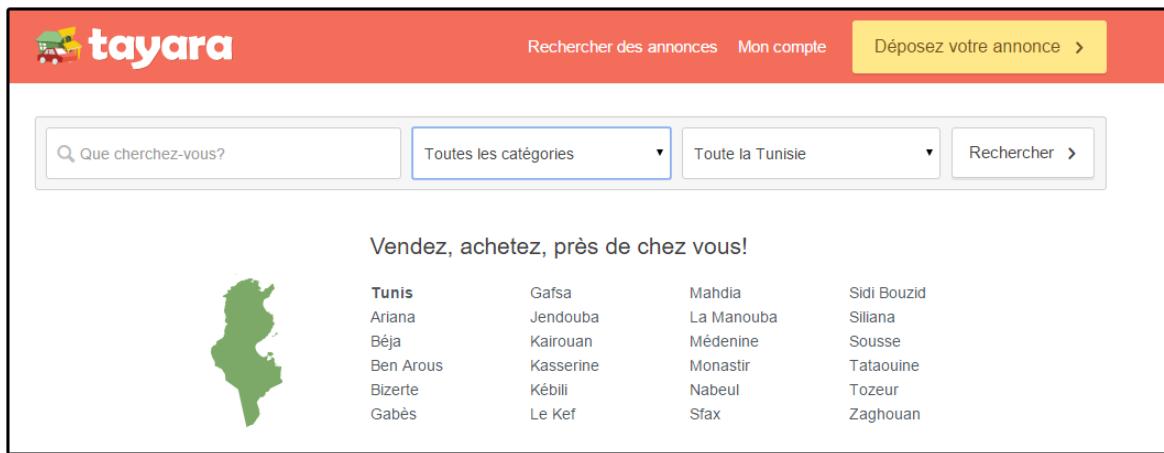


Figure I.3: Capture de site Tayara

#### Les points forts :

- Tayara et le site le plus connu en tunisie grace aux publicités dans les radios, TV...

#### Les points faibles :

- Il n'y a pas d'espace client.
- Pas de gestion des produits, il faut faire une annonce pour chaque produit.

## 2.4 Tableau comparatif

Nous pouvons classer le résultat de l'analyse des applications web existantes mentionnées précédemment selon huit critères (Cx) pris en considération dans le processus d'évaluation de ses applications :

- **C1: Attractivité** : Le visiteur doit se sentir ciblé, il faut que'il soit attiré par le site et que sa navigation soit orientée et balisée. C'est ici que l'apparence de site sera évaluée.
- **C2: Application mobile associée** : Avoir un site n'est pas de tout repos, avec une société et des internautes toujours plus exigeants. Il faut être en mesure de leur fournir une meilleure expérience et améliore la disponibilité du contenu, pour cela on a pris en considération dans l'évaluation si l'application admet-elle une version mobile ou non ?
- **C3: Espace client** : Un espace client permet de faciliter les actions des utilisateurs tel que gérer ses coordonnées, ses préférences, son adresse, etc.
- **C4: Gestion des produits** : Un site web est connu par ses fonctionnalités. Dans notre cas on est besoin d'une partie de gestion des produits donc on a comparé les sites par la disponibilité de cette fonctionnalité.
- **C5: Paiement en ligne** : Le paiement en ligne est aujourd'hui la méthode la plus simple et le plus rapide, en plus il est plus sécurisé que les méthodes classiques, donc pour cela on a choisi le paiement en ligne comme un critère dans l'évaluation des sites.
- **C6: Frais de service** : Les trois applications web qu'on a mentionnées sont gratuites mais cela n'empêche pas qu'ils ont des frais de commission.

- **C7: Rang global de site :** Un bon classement signifie que le site a une meilleure optimisation pour les moteurs de recherche, un nombre important des visites et plusieurs d'autres propriétés positives de sorte que si le rang est plus élevé l'application est plus fiable. On a utilisé Alexa.com<sup>1</sup> pour savoir le rang global des sites.
- **C8: Rang de site en Tunisie :** Dans le même cadre on a utilisé Alexa.com<sup>1</sup> pour savoir le rang des sites en Tunisie.

	<b>ebay.com</b>	<b>kaymu.tn</b>	<b>tayara.tn</b>
<b>Attractivité (c1)</b>	Bien ★★★★★	Moyenne ★★★★★	Faible ★★★★★
<b>Application Mobile (c1)</b>	Oui	Oui	Non
<b>Espace Client (C2)</b>	Oui	Oui	Non
<b>Gestion des produits (C3)</b>	Oui	Oui	Oui
<b>Paiement (C6)</b>	Oui	Non	Non
<b>Frais de service (C7)</b>	Oui (7,5% du prix total)	Oui (taux variable en fonction du niveau de service)	Non
<b>Rang Global (c4)</b>	22	63 413	8 481
<b>Rang TN (c5)</b>	-	92	11

Table I.1: Tableau comparatif des applications web existantes

### 3 Solution proposée

Après avoir détaillé notre problématique et bien analysé les différentes solutions existantes en Tunisie et à l'étranger, la question suivante doit se poser naturellement:

Est ce que les applications web existantes peuvent représenter une solution qui répond à notre problème ?

Les applications disponibles sur le marché sont globales et parfois payantes. Cependant, nous avons besoin d'une solution spécifique, moins couteuse et qui répond parfaitement aux besoins spécifiés par la société cliente en respectant son indépendance et son autonomie. Par conséquent, nous avons pensé à analyser et développer une application web et mobile qui répond exactement aux besoins de client.

Notre solution proposée est illustrée dans la figure suivante :

---

<sup>1</sup>Alexa.com est un site web principalement connu pour fournir des statistiques sur le trafic du Web.

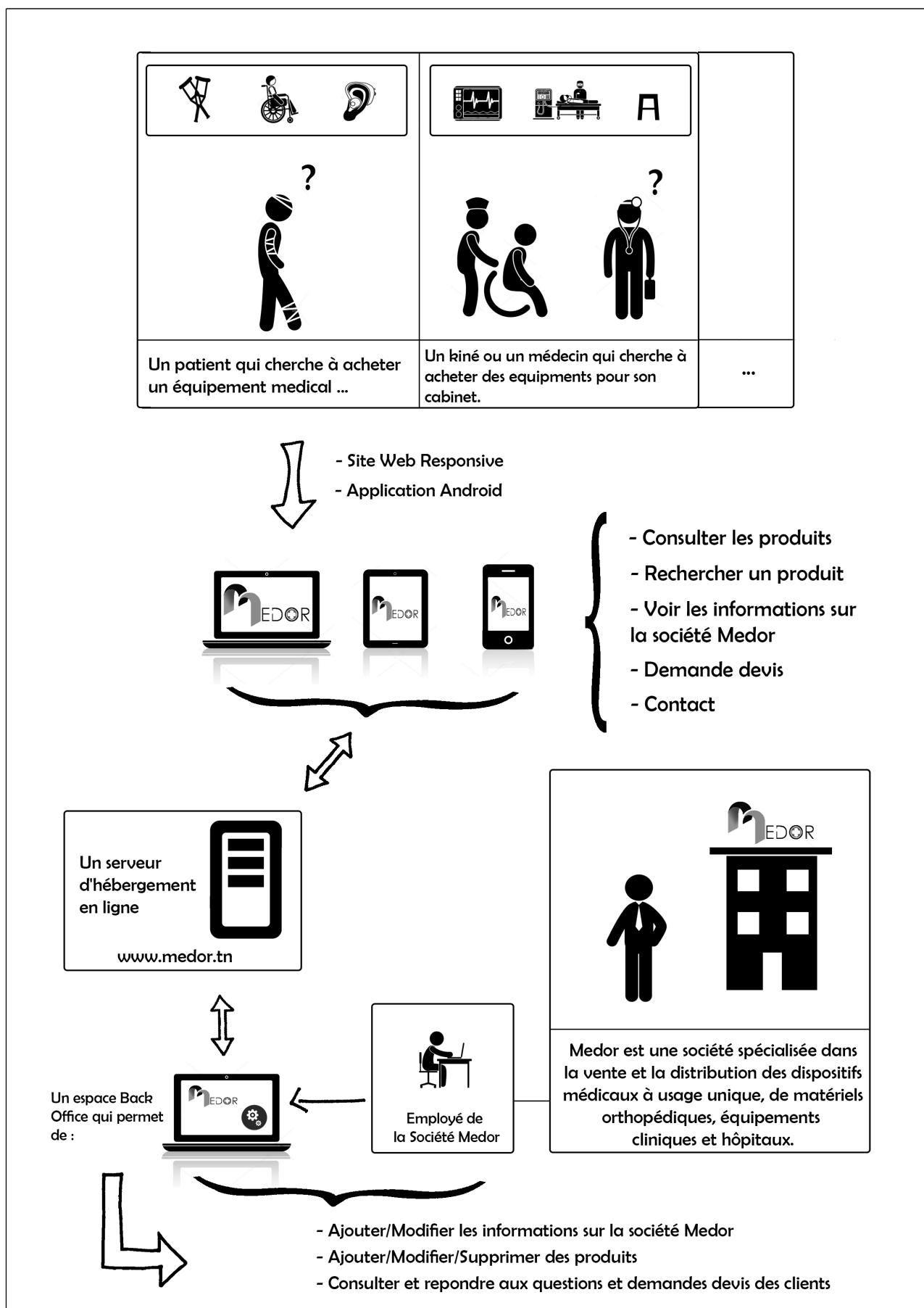


Figure I.4: Illustration de la solution

## 4 Méthodologie de travail et modélisation

Une méthodologie de développement est un cadre utilisé pour structurer, planifier et contrôler le développement d'une application. C'est le fait de modéliser un système avant sa réalisation pour bien comprendre son fonctionnement et assurer sa cohérence. Un modèle est ainsi un facteur de réduction des coûts et des délais. Il est donc indispensable pour assurer un bon niveau de qualité de produit dont la maintenance est efficace.

Nous présentons les cycles de vie d'un logiciel ainsi qu'aux principales méthodologies de développement dédiées aux démarches de conception orientées objet pour déterminer le processus à suivre dans ce travail.

### 4.1 Le cycle de vie d'un logiciel

Le cycle de vie d'un logiciel (lifecycle), désigne toutes les étapes du développement d'un logiciel, de sa conception à sa disparition. Il comprend généralement au minimum les étapes suivantes:

- **Analyse des besoins et faisabilité :** c'est-à-dire l'expression, le recueil et la formalisation des besoins du demandeur (le client) et de l'ensemble des contraintes, puis l'estimation de la faisabilité de ces besoins.
- **Spécifications ou conception générale :** il s'agit de l'élaboration des spécifications de l'architecture générale du logiciel.
- **Conception détaillée :** cette étape consiste à définir précisément chaque sous-ensemble du logiciel.
- **Codage (Implémentation ou programmation) :** c'est la traduction dans un langage de programmation des fonctionnalités définies lors de phases de conception.
- **Tests unitaires :** Ils permettent de vérifier individuellement que chaque sous-ensemble du logiciel est implémenté conformément aux spécifications.
- **Intégration :** L'objectif est de s'assurer de l'interfaçage des différents éléments (modules) du logiciel. Elle fait l'objet de tests d'intégration.
- **Qualification (ou recette) :** C'est-à-dire la vérification de la conformité du logiciel aux spécifications initiales.
- **Documentation :** Elle vise à produire les informations nécessaires pour l'utilisation du logiciel et pour des développements ultérieurs.
- **Mise en production :** C'est le déploiement sur site du logiciel.
- **Maintenance :** Elle comprend toutes les actions correctives et évolutives sur le logiciel.

Le tableau suivant compare les cycles de vie les plus utilisés:

<i>Cycle de vie de Logiciel</i>	<i>Principes</i>	<i>Avantages</i>	<i>Inconvénients</i>
<b>Modèle en Cascade</b>	<ul style="list-style-type: none"> <li>- Les étapes se terminent à des dates précises par la production de documents ou logiciels.</li> <li>- Processus linéaire.</li> <li>- Les résultats de l'étape sont examinés attentivement avant de passer à l'étape suivante.</li> </ul>	<ul style="list-style-type: none"> <li>- Facile à comprendre et à utiliser.</li> <li>- Les limites de chaque étape sont visibles.</li> <li>- vision globale du projet au cours du développement.</li> </ul>	<ul style="list-style-type: none"> <li>- Tous les besoins doivent être bien spécifiés au départ.</li> <li>- Difficulté de retour en arrière aux cas des anomalies.</li> <li>- L'intégration n'a lieu qu'à la fin du cycle.</li> </ul>
<b>Modèle en V</b>	<ul style="list-style-type: none"> <li>- Le développement des tests et du logiciel sont effectués de manière synchrone.</li> <li>- Avec toute décomposition doit être décrite la recomposition.</li> <li>- Toute description d'un composant est accompagnée de tests qui assurent leurs correspondances</li> </ul>	<ul style="list-style-type: none"> <li>- Prévention des erreurs : validations des produits à chaque sortie d'étape descendante.</li> <li>- Limitation des risques en cascade par validation de chaque étape.</li> <li>- Existence d'outils support (modèle très utilisé).</li> </ul>	<ul style="list-style-type: none"> <li>- Le processus n'est pas itératif.</li> <li>- Une mauvaise prise en compte des évènements concurrents.</li> <li>- Faiblissement contre les changements de la spécification des besoins</li> </ul>
<b>Modèle en Spirale</b>	<ul style="list-style-type: none"> <li>- Détermination des objectifs du cycle, analyse des risques et évaluation des alternatives</li> <li>- Développement et vérification de la solution retenue.</li> <li>- Revue des résultats et vérification du cycle suivant.</li> </ul>	<ul style="list-style-type: none"> <li>- Donne des indications sur les risques majeurs sans coût élevé.</li> <li>- La conception ne doit pas forcément être terminée.</li> <li>- Le développement se fait en interaction avec les clients</li> </ul>	<ul style="list-style-type: none"> <li>- Ce modèle est complexe.</li> <li>- La spirale peut être infinie.</li> <li>- Il est difficile de définir les objectifs et les points de validation intermédiaires entre les différentes étapes.</li> </ul>
<b>Modèle par Incrémentation</b>	<ul style="list-style-type: none"> <li>- Développement d'un noyau fonctionnel commun à un ensemble d'applications.</li> <li>- Développement de ces applications autour du noyau commun.</li> <li>- Le recouvrement (parallélisation) des différentes phases.</li> </ul>	<ul style="list-style-type: none"> <li>- Chaque développement est moins complexe.</li> <li>- Les intégrations sont progressives.</li> <li>- Possibilité de livrer et de mettre en service chaque incrément.</li> </ul>	<ul style="list-style-type: none"> <li>- Nécessité d'une bonne planification et une bonne conception.</li> <li>- Le coût total du développement du système n'est pas négligeable.</li> <li>- Les différentes interfaces doivent être bien définies</li> </ul>

Table I.2: Tableau comparatif des cycles de vie de logiciel

### Interprétation et Conclusion :

Méthodes classiques qui ne permettent pas une vision réelle sur le résultat.

- Manque de flexibilité aux modifications des spécifications ou des changements brusques.
- Démarche complexe et risque de ne pas respecter le cycle.

- Résultats souvent insatisfaisants aux besoins de client.

→ Méthodes à écarter... Recours aux processus modernes et simplifiés: les méthodes Agiles ou les processus Unifiés.

## 4.2 Processus simplifiés et modernes pour les applications web

### 4.2.1 Méthodes d'analyse et de conception

Les méthodes d'analyse et de conception fournissent une méthodologie et des notations standards qui aident à concevoir des logiciels. Ils peuvent être classés selon le suivant:

- **La distinction entre composition et décomposition :**

Elle met en opposition d'une part les méthodes ascendantes qui consistent à construire un logiciel par composition à partir de modules existants et, d'autre part, les méthodes descendantes qui décomposent récursivement le système jusqu'à arriver à des modules programmables simplement.

- **La distinction entre fonctionnelle et orientée objet :**

Dans la stratégie fonctionnelle (structuré) un système est vu comme un ensemble hiérarchique d'unités en interaction, ayant chacune une fonction clairement définie. Les stratégies orientées objet considèrent qu'un système est un ensemble d'objets interagissants qui décrivent son état.

### 4.2.2 Méthodes Agile ou Processus Unifié ?

Un processus définit une séquence d'étapes qui mènent à obtenir un système logiciel ou à évoluer un système existant en répondant à la question fondamentale: Qui fait quoi et quand? Deux types de processus de développement d'applications web sont possibles : on trouve d'une part les **méthodes agiles** qui sont actuellement en vogue et d'autre part on peut utiliser un **Processus Unifié UP** qui représente un cadre général et très complet de développement.

#### 4.2.2.1 La méthode Agiles

Cette méthode se base sur quatre valeurs fondamentales :

- "**Personnes et interactions plutôt que processus et outils**" : L'équipe est bien plus importante que les moyens matériels ou les procédures.
- "**Logiciel fonctionnel plutôt que documentation complète**" : Il est vital que l'application fonctionne. Le reste est secondaire. Il est préférable de commenter le code et de transférer les compétences au sein de l'équipe.
- "**Collaboration avec le client plutôt que négociation de contrat**" : Le client doit être impliqué dans le développement, il doit collaborer avec l'équipe et fournir un feedback continu sur l'adaptation du logiciel à ses attentes.
- "**Réagir au changement plutôt que suivre un plan**" : La planification initiale et la structure du logiciel doivent être flexibles aux demandes du client tout au long du projet.

Le tableau suivant compare les méthodes Agiles **XP** et **Scrum** :

<i>Méthodes Agiles</i>	<i>eXtreme Programming (XP)</i>	<i>Scrum</i>
<i>Définitions</i>	L'eXtreme Programming (XP) est un ensemble de pratiques qui couvre une grande partie des activités de la réalisation d'un logiciel, en passant par l'organisation de l'équipe de développement et les échanges avec le client.	Scrum est un processus agile qui s'articule autour d'une équipe soudée, qui cherche à atteindre un but en répondant à des problèmes complexes et changeants, tout en livrant de manière productive et créative des produits de la plus grande valeur possible.
<i>Principes</i>	<p>Un utilisateur à plein-temps dans la salle du projet pour assurer une communication directe entre les clients et les développeurs.</p> <ul style="list-style-type: none"> <li>- Écrire le test unitaire avant le code qu'il doit tester, afin d'être certain que le test sera systématiquement écrit ,et non pas négligé.</li> <li>- Programmer en binôme, afin de partager les connaissances et permettre aux débutants d'apprendre auprès des experts.</li> <li>- Intégrer de façon continue pour éviter le risque d'intégration des modules incompatibles à la fin du projet.</li> </ul>	<ul style="list-style-type: none"> <li>- L'équipe se focalise de façon itérative sur un ensemble de fonctionnalités à réaliser, dans des itérations appelées Sprints.</li> <li>- Chaque Sprint possède un but à atteindre à travers des fonctionnalités choisies à implémenter dans ce Sprint.</li> <li>- Le scrummaster se charge de réduire les perturbations extérieures et de résoudre les problèmes non techniques de l'équipe.</li> <li>- La participation active du client pour définir les priorités dans les fonctionnalités du logiciel et choisir lesquelles seront réalisées dans chaque Sprint.</li> </ul>
<i>Interprétations</i>	Méthodologie légère qui met l'accent sur l'activité de programmation et qui s'appuie sur la communication, la simplicité et le feedback. Elle est bien adaptée pour des projets de taille moyenne où le contexte évolue en permanence.	La méthode Scrum définit un cadre de travail permettant la réalisation de projets complexes qui sont divisés en plusieurs cycles de travail et qui nécessitent une longue durée de développement.

Table I.3: Tableau représentant les méthodes Agile: XP et Scrum

## Conclusion :

Méthodes adaptées pour les projets complexes ou de taille moyenne qui durent longtemps et qui nécessitent l'intervention d'une grande équipe de développement qui reste tout le long du projet en interaction directe avec le client pour résoudre les tâches complexes et réagir contre l'évolution permanente du contexte.

→ Vu que ce n'est pas le cas de notre projet, les méthodes Agiles sont à écarter... recours aux Processus Unifiés.

### 4.2.2.2 Le Processus Unifié (UP)

Le Processus Unifié est un processus de développement logiciel organisée suivant les quatre phases suivantes:

1. La phase d'**initialisation** permet de décider la poursuite du projet ou son arrêt.
2. La phase d'**élaboration** poursuit trois objectifs principaux en parallèle : (identifier et décrire, construire, lever les risques)
3. La phase de **construction** consiste à concevoir et implémenter l'ensemble des éléments opérationnels.
4. Enfin, la phase de **transition** permet de faire passer le système informatique des mains des développeurs à celles des utilisateurs finaux.

Les implémentations du Processus Unifié sont représentés dans le tableaux suivant:

	<b>RUP: Rational Unified Process</b>	<b>2TUP: Two Tracks Unified P</b>
<b>Les principes fondamentaux du Processus Unifié (UP)</b>	<p>Le Rational Unified Process (RUP) est l'une des plus célèbres implémentations de la méthode PU. Elle permet de donner un cadre au développement logiciel, répondant aux exigences fondamentales préconisées par les créateurs d'UML:</p> <ul style="list-style-type: none"> <li>- Une méthode de développement doit être guidée par les besoins des utilisateurs.</li> <li>- Elle doit être centrée sur l'architecture logicielle.</li> <li>- Elle doit être itérative et incrémentale.</li> </ul>	<p>Le 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, les messages, produire le cahier des charges et modéliser le contexte (le système est une boîte noire, les acteurs l'entourent et sont reliés à lui). Le processus s'articule ensuite autour de trois phases essentielles: une branche technique, une branche fonctionnelle et une phase de réalisation.</p>

Table I.4: Tableau représentant les implémentations du Processus Unifié: RUP et 2TUP

### Conclusion et choix de la Méthodologie à suivre :

Notre projet est basé sur un processus de développement bien défini qui va de la détermination des besoins fonctionnels attendus du système jusqu'à la conception et le codage final. C'est pour cela qu'on est besoin d'un cycle de développement qui dissocie les aspects techniques des aspects fonctionnels tout en commençant par une étude préliminaire. Notre choix s'est alors porté vers la méthode 2TUP vu qu'elle est caractérisée par une approche nouvelle et originale et qu'elle respecte le cadre de notre projet.

→ Nous adaptons alors le processus 2TUP pour la suite de notre travail.

### 4.3 Mise en pratique du processus 2TUP

2 Tracks Unified Process est un processus uniifié (c'est-à-dire construit sur UML, itératif, centré sur l'architecture et conduit par les cas d'utilisation). Ce processus de développement logiciel est caractérisé par le suivant :

- **Itératif et incrémental** : Le projet est découpé en itérations de courte durée (environ 1 mois) qui aident à mieux suivre l'avancement global.
- **Centré sur l'architecture** : Tout système complexe doit être décomposé en parties modulaires afin de garantir une maintenance et une évolution facilitées.
- **Piloté par les risques** : Les risques majeurs du projet doivent être identifiés au plus tôt, mais surtout levés le plus rapidement possible.
- **Conduit par les cas d'utilisation** : Les cas d'utilisation du futur système sont identifiés et décrits avec précision.

La démarche 2TUP est proposée pour le développement des logiciels fournissant les services sur Internet.

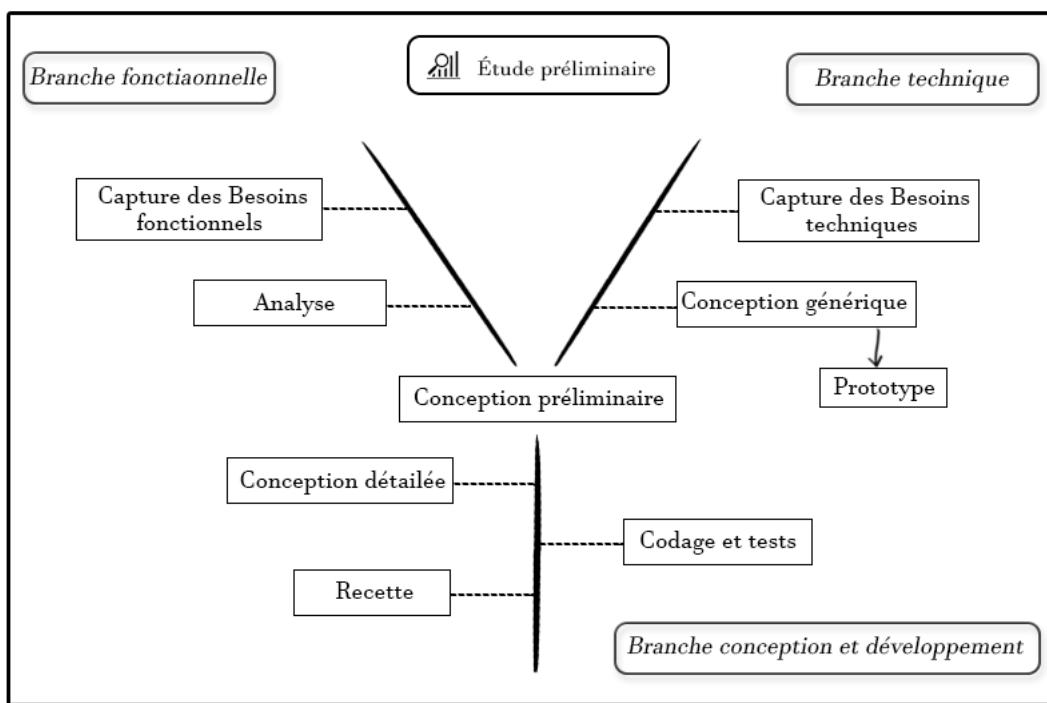


Figure I.5: Les étapes du processus 2TUP

Le principe fondateur du 2TUP est que toute évolution imposée à un logiciel peut se décomposer et se traiter parallèlement, suivant un axe fonctionnel et un axe technique. La réalisation du logiciel consiste à fusionner les résultats de ces deux branches du processus.

Selon le processus 2TUP que nous suivons, nous passons obligatoirement par les phases suivantes à la suite de notre travail:

1. **L'étude préliminaire** qui contient une description du service et les cas d'utilisation principaux, c'est une première version de la spécification générale.
2. **La capture des besoins fonctionnels**, qui définit le quoi faire à travers une *spécification générale* qui décrit le service à développer d'un point de vue fonctionnel et une *spécification détaillée* qui précise les traitements qui concernent chaque scénario des cas d'utilisation présent en spécification générale tout en respectant les contraintes *fonctionnelles* et *non fonctionnelles*.
3. **L'Analyse** où on effectue simultanément l'étude des données et l'étude des traitements à effectuer.
4. **La capture des besoins techniques** qui permettent de satisfaire les contraintes techniques présentes dans le cahier des charges et donc répondre aux attentes de client.
5. **La conception générique** qui définit le comment faire.
6. **La conception préliminaire** qui liste les exigences techniques à partir de l'étude préliminaire et définit l'architecture technique et les choix d'implémentation technique.
7. **La conception détaillée** qui précise l'implémentation technique de l'application. Elle consiste en la fusion de la spécification détaillée et de la conception générique, pour déterminer *comment faire le quoi faire dans le détail*. On y trouve le schéma de base de données, les diagrammes de classes et les diagrammes de séquence supplémentaires qui détaillent les interactions entre les composants du logiciel.
8. **Le Codage et les Tests** qui décrivent comment l'application est réalisée et installée sur un environnement d'exécution et comment la stratégie de validation est effectuée.

#### **4.4 Diagramme de gantt**

La planification est parmi les phases d'avant projet les plus importantes. Elle consiste à déterminer et à ordonner les tâches du projet et à estimer leurs charges respectives.

Pour planifier notre projet et suivre l'avancement, nous avons utilisé le diagramme de GANTT. Ce diagramme permet de planifier le projet et de rendre le suivi de son avancement plus simple. Il permet aussi de visualiser l'enchaînement et la durée des différentes tâches durant le stage.

La figure du diagramme de GANTT suivante représente les tâches principales à réaliser dans notre projet :

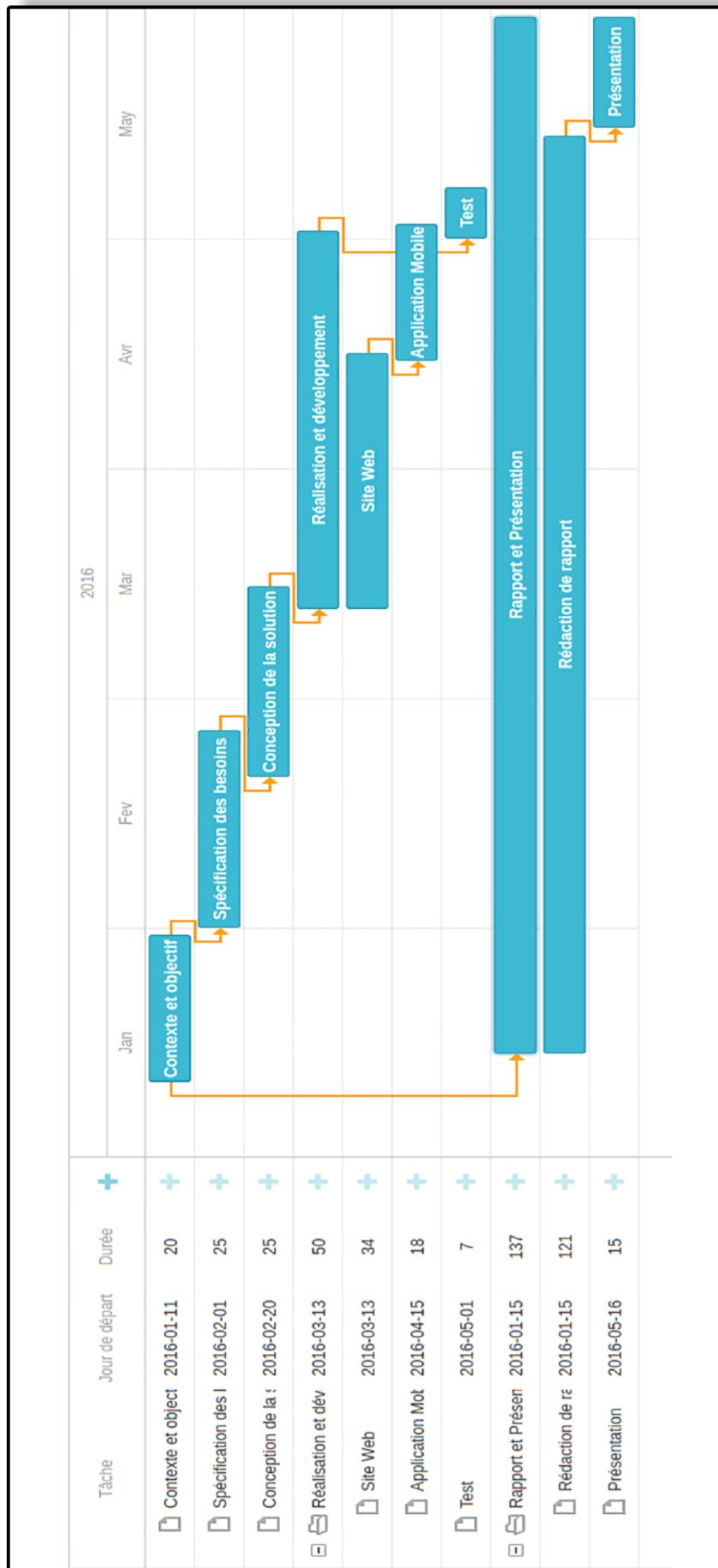


Figure I.6: Diagramme de Gantt

## Conclusion

Dans ce premier chapitre, nous avons fait une étude de l'existant nous permettant de bien entamer notre projet. Nous avons comparé les différentes méthodes de développement afin de clôturer ce chapitre par une présentation de la méthodologie 2TUP que nous allons adopter pour le développement de notre projet. Le chapitre suivant sera consacré à l'analyse de notre projet et à la capture des besoins.

## Chapitre II

### Analyse et capture des besoins

# Introduction

Le présent chapitre présente une analyse qui joue un rôle important dans la démarche ergonomique du développement d'une application.

Il a donc comme objectif de permettre de formaliser les étapes préliminaires du développement d'un système afin de rendre ce développement plus fidèle aux besoins du client qui sont divisés en besoins fonctionnels, besoins non fonctionnels et besoins techniques.

## 1 Étude préliminaire

D'après une étude du marché, on a trouvé une diversité des sites web dédiés au vente du matériel Médico-Chirurgical, Oethopédique et matériel kiné. On a pris à titre d'exemple les sites web suivants: medicochirurgical.com, elfaraj.com, locamed.tn, geantmedical.com, sofamed.com et domital.fr.

### 1.1 Medico-Chirurgical

Medico-Chirurgical est une société de distribution de matériel médical auprès des professionnels de la santé en Tunisie depuis 1965.



Figure II.1: Capture de site Medicochirurgical

**Les points forts :**

- Premier résultat de recherche sur Google avec le mot-clé "Matériel Médico et orthopédique tunisie".

**Les points faibles :**

- Construit avec flash, une technologie abandonnée dont les inconvénients sont multiples :
  - Le langage Flash ne prend pas en compte le référencement naturel
  - Il est difficile d'adapter le site sur l'ensemble des supports informatiques comme des tablettes, Smartphones...

- Pour lire ce langage l'internaute a besoin d'installer Flash Player sur son ordinateur.
- La consultation des articles n'est pas direct il faut télécharger le catalogue des produits en pdf.
- La fonctionnalité de recherche et du newsletter ne fonctionnent pas.

## 1.2 El Faraj

El Faraj est une société de vente de matériel médical aux particuliers et professionnels en Tunisie.



Figure II.2: Capture de site Elfaraaj

**Les points forts :**

- Plusieurs produits disponibles avec différents types.
- Les produits et leurs informations sont clairs et faciles à trouver.
- La possibilité de demander de renseignements pour chaque produit.

**Les points faibles :**

- L'utilisation du CMS Joomla version 1.5 une version obsolète qui peut menacer la sécurité de site.
- L'absence d'un espace client et la difficulté de trouver les informations de contact.

## 1.3 Locamed

Locamed est une société de vente et location de matériel médical (lit médicalisé, matelas anti escarres, fauteuil roulant, coussin anti escarres, soulève-personne, aide à la marche, ...) en Tunisie.

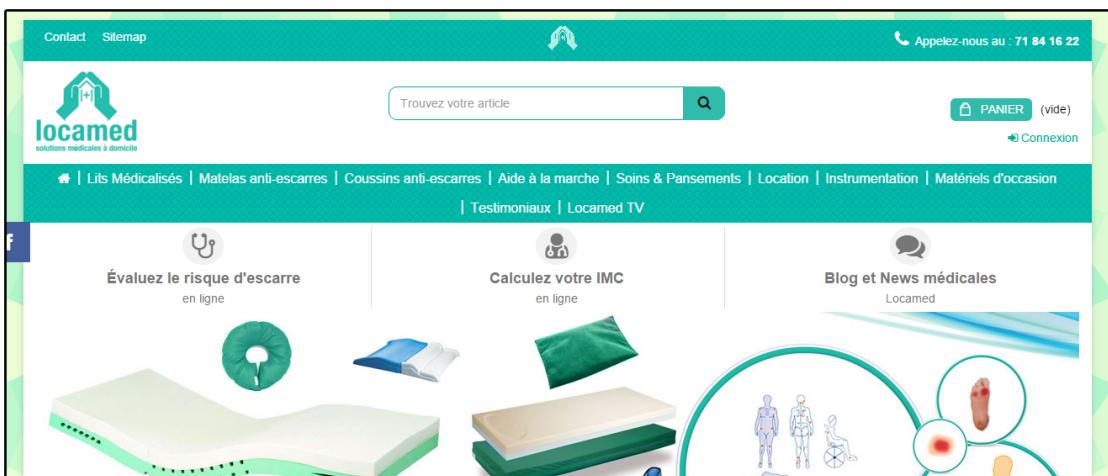


Figure II.3: Capture de site Locamed

#### Les points forts :

- Les produits, leurs prix et leurs informations sont clairs et faciles à trouver.
- Un espace client et un panier pour le commerce électronique.
- Un newsletter.
- D'après alexa.com le rang global de site est 1.1 million et 4 mille en Tunisie. la page facebook a 19 mille de fans ce qui montre que le site a plusieurs visiteurs.

#### Les points faibles :

- L'utilisation du CMS Prestashop, l'ajout des fonctionnalités additionnelles va coûter très chères car il n'y a pas de nombreux modules gratuits dans Prestashop.
- Le site n'est pas totalement responsive.

### 1.4 Geant Medical

Geant Medical est un siteweb de vente des matériel Médical au Casablanca, Marrakech.

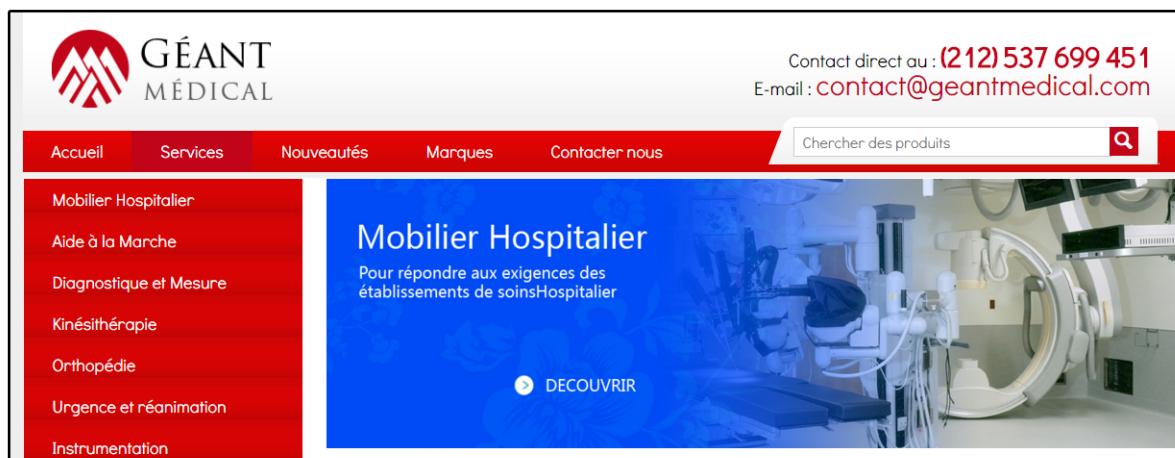


Figure II.4: Capture de site Geantmedical

### Les points forts :

- Le template de site est clair, simple et beau.
- Le site est rapide, les pages se chargent rapidement.
- Le site contient un espace client et un panier pour le commerce électronique.

### Les points faibles :

- Quelques catégories n'ont pas des produits.

## 1.5 Sofamed

Sofamed un site de vente et location de matériel médical, handicap et incontinence à l'étranger.



Figure II.5: Capture de site Sofamed

### Les points forts :

- Plusieurs produits disponibles avec différents types.
- Page d'accueil dynamique avec les promos, les nouveautés...
- un espace client et une panier pour le commerce électronique.

### Les points faibles :

- Le site n'est pas responsive.

## 1.6 Domital

Domital est un site de vente de matériel médical et paramedical en Gironde. Maintien à domicile, Hygiène, aide à la vie, Handicap, nutrition, Orthopédie, Maternité....



Figure II.6: Capture de site domital

#### Les points forts :

- Site simple et clair.

#### Les points faibles :

- Le site n'est pas responsive.
- Quelques parties de site sont statiques.

## 1.7 Tableau comparatif

Dans les tableaux suivant (l'un pour les applications web similaires en Tunisie et l'autre pour les applications web similaires à l'étranger), nous pouvons trouver le résultat de l'analyse et de critique des applications existantes mentionnées précédemment selon cinq critères (Cx) pris en considération dans le processus d'évaluation de ses applications:

- **C1: Espace Client :** Un espace client permet de faciliter les actions des utilisateurs tel que gérer ses coordonnées, ses préférences, son adresse, etc.
- **C2: Application mobile associée :** Avoir un site n'est pas de tout repos, avec une société et des internautes toujours plus exigeants. Il faut être en mesure de leur fournir une meilleure expérience et améliorer la disponibilité du contenu, pour cela on a pris en considération dans l'évaluation si l'application admet-elle une version mobile ou non ?
- **C3: Technologie utilisée / CMS utilisé(e):** Quel que soit la profondeur d'analyse du dispositif technique derrière le site web, il est très utile de connaître les technologies utilisées par le site web. Cela permet d'évaluer la maturité technique d'un projet web, mais surtout, les technologies utilisées peuvent expliquer certains problèmes, par exemple un temps de chargement très long ou des problèmes de sécurité.
- **C4: Nombre des pages indexés par le moteur de recherche Google :** Il est important d'avoir une idée du nombre de pages indexées par Google pour un site web donné. Le chiffre important parce qu'une page qui n'est pas dans l'index, il y a fort peu de chances qu'elle se retrouve dans les résultats du moteur de recherche...

- **C5: Rapidité :** Les internautes sont bien connus pour être des personnes impatientes, si le chargement de site est trop lent, l'intéressé n'aura pas la patience d'attendre et ira voir ailleurs sans même avoir eu le temps de jeter un œil sur le site. Pour cela on a utilisé gtmetrix.com<sup>2</sup> pour savoir le temps nécessaire (en seconde) pour charger la page d'accueil.

Applications web similaires en Tunisie :

	<b>medicochirurgical.com</b>	<b>elfaraj.com</b>	<b>locamed.tn</b>
<b>Client (C1)</b>	Non	Non	Oui
<b>Mobile (C2)</b>	Non	Non	Non
<b>Technologie / CMS (C3)</b>	Flash	Joomla	PrestaShop
<b>Pages indexés (C4)</b>	7	1130	166
<b>Rapidité (C5)</b>	4.4 s	11.5 s	8.3 s

Table II.1: Tableau comparatif des applications web similaires en Tunisie

Applications web similaires à l'étranger :

	<b>geantmedical.com</b>	<b>sofamed.com</b>	<b>domital.fr</b>
<b>Client (C1)</b>	Non	Oui	Non
<b>Mobile (C2)</b>	Non	Non	Non
<b>Technologie / CMS (C3)</b>	-	-	PHP
<b>Pages indexés (C4)</b>	786	7220	86
<b>Rapidité (C5)</b>	3.2 s	6.8 s	3.7 s

Table II.2: Tableau comparatif des applications web similaires à l'étranger

<sup>2</sup>GTmetrix est un outil gratuit qui permet d'analyser la performance d'une page internet. Il génère des scores et propose des recommandations pour améliorer le résultat.

## 2 Besoins fonctionnels

Le système doit permettre :

- **Au visiteur de :**
  - **Créer un compte** : Un visiteur peut créer un compte.
- **A l'internaute de :**
  - **Consulter les produits** : Chaque produit présent sur le site sera présenté en détail sur sa propre page représentant sa fiche technique. On y trouvera en particulier une image que l'internaute pourra agrandir, une description du produit, son prix et sa disponibilité.
  - **Chercher un produit** : l'internaute a la possibilité de chercher un produit par son code ou son nom. La résultat du recherche sera disponible sur une page particulière.
  - **Demande devis et contact** : Chaque utilisateur de l'application doit accéder facilement à un formulaire permettant la demande d'un devis ou contacte la société Medor.
  - **Gérer panier** : Lorsque l'internaute est intéressé par un produit, il peut l'enregistrer dans un panier virtuel. Il doit pouvoir ensuite à tout moment ajouter, supprimer ou encore modifier les quantités avant de passer la commande. Le panier ne sera pas sauvegardé dans la base de données, sa durée de vie n'excèdera pas celle de la visite de l'utilisateur.
- **Au client de :**
  - **Gérer ses commandes** : À tout moment, le client doit pouvoir accéder au formulaire du bon de commande, dans lequel il saisit ses coordonnées et les informations nécessaires pour obtenir sa facture. Le client devra pouvoir ensuite suivre ses commandes récentes.
  - **Gérer son compte** : D'une manière générale, le client devra pouvoir gérer son compte, c'est-à-dire modifier ses coordonnées, ses préférences, son adresse, etc.
- **Au responsable commercial de :**
  - **Gérer les catégories** : Le responsable commercial a le droit de gérer les catégories.
  - **Gérer les produits** : Le responsable commercial a le droit de gérer les produits.
- **A l'administrateur de :**
  - **Gérer le site** : L'administrateur peut gérer le site dans la partie « Backend ».
  - **Consulter contact et demande devis** : L'administrateur peut consulter et repérer au messages et demande devis et contact reçue.
  - **Gérer les utilisateurs** : L'administrateur peut gérer les comptes des membres.
  - **Gérer les partenaires** : L'administrateur peut gérer les partenaires de la société.
  - **Consulter les commandes** : L'administrateur peut consulter et manipuler les commandes effectuées.

## 3 Besoins non fonctionnels

### 1. Exigences de qualité

- **Ergonomie sobre et efficace :** Acheter un produit sur le Web ne doit pas prendre beaucoup de temps ou demander une maîtrise de mathématiques! La mise en page du site facilitera au maximum la démarche à l'aide d'une présentation claire et intuitive. Les sites trop riches et trop complexes n'incitent pas à l'achat, car ils demandent un effort intellectuel important et non souhaité.
- **Formulaire de commande simple :** Très souvent, l'internaute cale au moment d'acheter, car l'effort le plus important à fournir est le renseignement du bon de commande ! La conception et la présentation de celui-ci seront donc particulièrement soignées pour ne pas rebuter le client.

### 2. Exigences de performance

- **Besoin de sécurité :** L'application doit garantir à l'utilisateur connecté l'intégrité et la confidentialité de ses données. La sécurité du système est assurée par l'authentification des clients par un login et un mot de passe crypté.
- **Performance :** L'application doit fournir tous les statuts et informations en temps réel et d'une manière optimale.
- **La flexibilité :** Une ouverture sur d'autres domaines serait un atout majeur pour notre application. Ce point ne doit pas être mis à côté lors de la conception de l'architecture du projet.
- **L'intégrité :** Le système doit savoir comment traiter les échecs des interfaces, faire la capture des différentes erreurs d'entrées-sorties, effectuer le traitement des mauvaises données sans oublier l'intégrité des informations dans les tables de la base de données et des interfaces
- **La portabilité :** L'application doit s'exécuter sans problème sur tous les navigateurs. Il est ainsi primordial qu'elle soit compatible avec les différents appareils Android.

### 3. Contraintes de conception

- **Mise à jour des données :** Les informations relatives aux produits présentés sur le site proviendront essentiellement de deux sources complémentaires. La première servira à alimenter la base avec tous les nouveaux produits, la seconde à mettre à jour les données qui concernent le prix, l'état et la disponibilité des produits du catalogue.

Toutes ces informations seront saisies manuellement à travers un espace administrateur (back office) permettant de gérer les produits, catégories et les utilisateurs de l'application.

- **Les formulaires du site :** Les données saisies depuis le site web et enregistrées dans la base décrivant les coordonnées des clients, ainsi que les caractéristiques de leurs commandes seront mémorisées.

Toutes les données personnelles seront protégées et leur confidentialité sera garantie. Les commandes seront enregistrées, puis traitées ultérieurement par le service clientèle. Le client pourra consulter l'historique de toutes ses commandes.

## 4 Besoins techniques

L'étude technique est une phase d'adaptation de conception à l'architecture technique. Afin de bien expliquer nos choix technologiques, nous avons recours à une étude comparative entre les technologies qui peuvent être utilisées durant notre projet.

### 4.1 Étude des outils de développement

Nous commençons d'abord par présenter les différents langages de développement destiné à l'application web. Ci-dessous est une image qui présente les langages de programmation les plus utilisés côté serveur selon W3tech.com<sup>3</sup>.

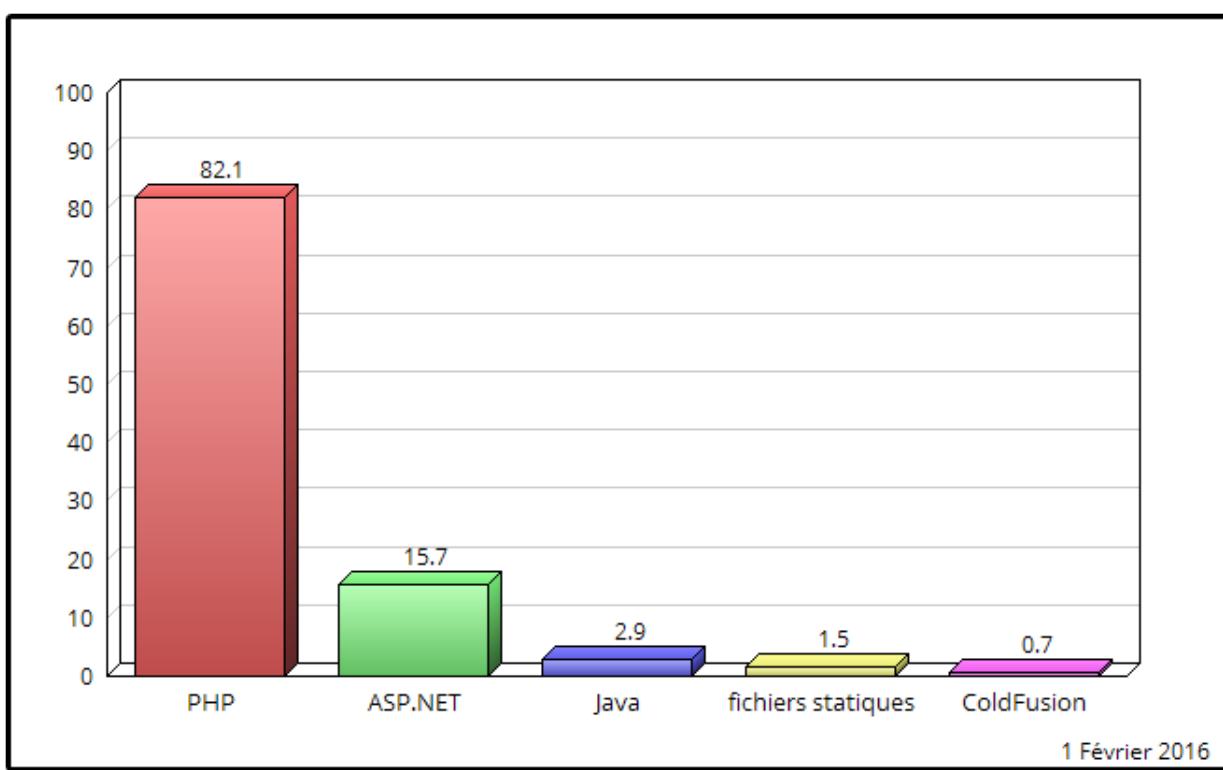


Figure II.7: Les langages de programmation côté serveur les plus utilisés

Dans cette partie, nous allons comparer point par point, les caractéristiques des Trois langages les plus utilisés afin de savoir le bon choix.

- **PHP** est un langage largement utilisé pour le développement de sites Web. Ce langage bénéficie d'une grande communauté de développeurs ainsi que d'un grand nombre de fonctionnalités disponibles.
- **JAVA** est un des langages de programmation les plus utilisés, principalement sur les réseaux ou pour les services web.
- **.Net** est un ensemble de technologies de programmation créé par Microsoft. Les programmeurs peuvent utiliser ASP.NET pour créer des sites webs dynamiques, des applications web ou des web services XML.

<sup>3</sup>W3tech.com est un site qui présente de nombreuses statistiques et études chiffrées sur les technologies du web. Pour créer son étude il se base sur les 10 premiers millions de sites web classés par Alexa<sup>1</sup>.

Critère	PHP	JAVA	.Net
Version	7	8	4.6.1
Type d'application cible	Web	Toutes	Toutes
Lourdeur de mise en place	Faible	Moyen	Moyen
Pré-requis	Un interpréteur PHP et un serveur web (Souvent Apache)	Une JVM	Un environnement de développement windows et un serveur web windows server
Multiplateforme	Oui	Oui	Non
Popularité (selon l'index TIOBE <sup>4</sup> )	2.7%	20.5%	4.2% (C#)
Environnement de développement	Simple éditeur de texte, ou IDE intégré	Eclipse, IntelliJ, NetBeans, JDeveloper...	visual studio

Table II.3: Tableau comparatif des langages de programmation

Nous allons comparer les différents points forts et points faibles de chaque langage tel que le meilleur choix n'est pas nécessairement le meilleur pour tous les projets.

## PHP

- Points forts :

- Il est gratuit.
- Il existe une grande variété de projets open source disponible pour celui-ci.
- Il ne nécessite pas beaucoup de code pour obtenir un résultat.
- L'hébergement du PHP est supporté presque partout.
- Beaucoup de documentation.

- Points faibles :

- Les failles de sécurité se sont révélées très nombreuses au fil des années.
- Certains professionnels qualifient ce langage d'amateur : il y a eu des erreurs de conceptions au départ, beaucoup de failles de sécurité, manque de vision pour le développement de gros projet.

## Java (JSP)

- Points forts :

- Très grande portabilité.
- Plus grand stabilité du code à travers le temps.

<sup>4</sup>L'index TIOBE mesure la popularité des langages de programmation sur base du nombre de pages web retournées par un moteur de recherche lorsqu'on lui soumet le nom du langage de programmation.

- Le langage est soutenu par Oracle.
- Sécurité excellente avec Java/CORBA.

- Points faibles :

- L'hébergement du JSP n'est pas disponible chez tous les hébergeurs.

## **ASP.NET**

- Points forts :

- Il est compilé, rapidité d'exécution une fois qu'il est exécuté.
  - Le langage a déjà une bonne structure et architecture pour la programmation.
  - Le langage est soutenu par Microsoft.
  - La possibilité de développer en VB .NET ou en C# (C Sharp).

- Points faibles :

- Le coût est nécessairement plus élevé que les autres puisqu'il n'est pas gratuit.
  - L'hébergement se fait exclusivement sous Windows Server .

## **Choix :**

Pour plusieurs raisons, nous allons adopter le langage **PHP** pour développer notre application web. Tout d'abord, PHP est gratuit et ne nécessite aucune licence d'utilisation. PHP est le langage de programmation Web le plus utilisé. Il existe une communauté de développeurs très active qui rend disponibles des dizaines de milliers de librairies PHP de grande qualité ainsi qu'une vaste quantité de documentation et tutoriels pour le bénéfice de chacun. Ces ressources facilitent notre travail et réduisent notre temps d'exécution.

## **4.2 Étude des différents Framework**

Généralement pour structurer et accélérer le développement, l'utilisation de framework est requise. Pour cela nous effectuerons des études tout en sélectionnant et comparant les framework PHP les plus connues suivantes :

- **Laravel** est un framework web open-source écrit en PHP respectant le principe modèle-vue-contrôleur et entièrement développé en programmation orientée objet. Laravel est distribué sous licence MIT, avec ses sources hébergées sur GitHub.
- **zend** Framework est un cadre de travail open source orienté objets pour des applications web basées sur PHP 5. Zend Framework est souvent appelé une "bibliothèque de composants" car ses composants sont faiblement couplés et vous pouvez les utiliser de manière plus ou moins indépendante. Mais Zend Framework propose aussi une solution avancée Model-View-Controller (MVC) qui peut être utilisée pour monter des structures de base pour vos applications.
- **Symfony2** est un framework MVC libre écrit en PHP 5. Il fournit des fonctionnalités modulables et adaptables qui permettent de faciliter et d'accélérer le développement d'un site web.

	<b>Laravel</b>	<b>Zend2</b>	<b>Symfony2</b>
<b>Dépendances</b>	$\geq$ PHP 5.3.7	$\geq$ PHP 5.3	$\geq$ PHP 5.3.2
<b>Licence</b>	MIT	BSD	MIT
<b>Systèmes de templates</b>	PHP, Twig	phtml	Blade, PHP, Custom
<b>ORM (mapping objet-relationnel)</b>	ELOQUENT ORM	Zend Db, intégration de doctrine 2 via composer	Doctrine 2 ou Propel
<b>MVC</b>	Oui	Oui	Oui
<b>Simplicité</b>	★★★★★	★★★★★	★★★★★
<b>Supporte ajax</b>	Non	Non	Oui
<b>Extension</b>	Oui	Non	Oui
<b>Générateur de code</b>	Non	Non	Oui

Table II.4: Tableau comparatif des différents Framework

### Choix :

Dans le monde des Framework, il n'y a pas de framework idéal, chacun possède ses avantages et ses inconvénients. Mais généralement l'utilisation d'un framework présente plusieurs avantages tels que l'accélération du temps de développement, la conservation d'un code homogène, les applications sont plus facilement sécurisées...

Dans notre application web nous allons utiliser le Framework **Symfony**, car tout d'abord il possède plus d'avantage que les autres Framework comme le support d'ajax et des extension... Aussi Symfony est le Framework le plus utilisé en Tunisie, donc à part que c'est un avantage d'expérience pour nous dans le marché de travaille, si le client veut des améliorations ou une maintenance pour son site, la société de développement ou on a effectué notre stage a des experts symfony. Même si le client décide de consulter une autre société, il va aussi trouver des experts dans symfony car comme on l'a dit Symfony est le Framework le plus utilisé en Tunisie.

## Conclusion

Durant ce chapitre, Nous avons non seulement exposé la mise en pratique du processus 2TUP, mais nous avons également capturé les différents besoins de notre projet. Nous avons classé ces besoins en besoins fonctionnelles, non fonctionnelles et techniques. Après l'analyse des exigences et le choix des techniques a utilisé, Il est temps de passer à la conception de notre application dans le chapitre suivant.

# Chapitre III

## Conception

# Introduction

Dans cette partie nous traitons l'aspect conceptuel de notre application. Pour la conception et la réalisation de cette dernière, nous nous appuyons sur le formalisme UML basé sur les diagrammes et offrant une flexibilité marquante.

## 1 Conception générique

La phase de conception permet de décrire de manière non ambiguë, le plus souvent en utilisant un langage de modélisation, le fonctionnement futur du système, afin d'en faciliter la réalisation.

### 1.1 Le langage UML

Pour faciliter notre tâche nous avons recours au langage de modélisation unifié (UML : Unified Modelling Language) c'est une notation qui permet de modéliser un problème de façon standard. Ce langage qui est né de la fusion de plusieurs méthodes existantes auparavant est devenu une référence en terme de modélisation objet, UML est caractérisé par:

- C'est un langage formel et normalisé.
- Il permet le gain de précision, encourage l'utilisation d'outils et constitue à cet effet un gage de stabilité.
- UML est un support de communication performant.
- Il cadre l'analyse et facilite la compréhension de représentations abstraites complexes. Son caractère polyvalent et sa souplesse en font un langage universel.

### 1.2 Les diagrammes d'UML

UML 2 s'articule autour de treize types de diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel. Ces types de diagrammes sont répartis en deux grands groupes :

#### 1. Six diagrammes structurels :

- **Diagramme de classes** : Il montre les briques de base statiques : classes, associations, interfaces, attributs, opérations, généralisations, etc.
- **Diagramme d'objets** : Il montre les instances des éléments structurels et leurs liens à l'exécution.
- **Diagramme de packages** : Il montre l'organisation logique du modèle et les relations entre packages.
- **Diagramme de structure composite** : Il montre l'organisation interne d'un élément statique complexe.
- **Diagramme de composants** : Il montre des structures complexes, avec leurs interfaces fournies et requises.
- **Diagramme de déploiement** : Il montre le déploiement physique des 'artefacts' sur les ressources matérielles.

## 2. Sept diagrammes comportementaux :

- **Diagramme de cas d'utilisation** : Il montre les interactions fonctionnelles entre les acteurs et le système à l'étude.
- **Diagramme de vue d'ensemble des interactions** : Il fusionne les diagrammes d'activité et de séquence pour combiner des fragments d'interaction avec des décisions et des flots.
- **Diagramme de séquence** : Il montre la séquence verticale des messages passés entre objets au sein d'une interaction.
- **Diagramme de communication** : Il montre la communication entre objets dans le plan au sein d'une interaction.
- **Diagramme de temps** : Il fusionne les diagrammes d'états et de séquence pour montrer l'évolution de l'état d'un objet au cours du temps.
- **Diagramme d'activité** : Il montre l'enchaînement des actions et décisions au sein d'une activité.
- **Diagramme d'états** : Il montre les différents états et transitions possibles des objets d'une classe.

### 1.3 Modèles d'UML utilisés

Les diagrammes qu'on a montré dans la partie précédente sont des moyens de description des objets ainsi que des liens qui les relient. Nous n'utilisons pas les treize types de diagrammes proposés par UML 2, mais seulement un tiers parmi eux en insistant particulièrement sur les diagrammes suivants :

- **Diagramme de cas d'utilisation**
- **Diagramme de séquence**
- **Diagramme de composants**
- **Diagramme de classes**

Cette limitation est largement suffisante pour la plupart des projets et particulièrement pour notre travail.

## 2 Conception préliminaire

### 2.1 Différences entre un site web dynamique et statique

Les pages de site web statique peuvent présenter toute forme de contenu: animations flash, des images, de la musique, vidéos, etc. Mais elles sont toujours présentées de la même façon. Elles ne changent pas au cours de temps ou selon l'utilisateur.

Si on met l'accent sur l'architecture utilisée pour les sites statiques, un ordinateur qui se connecte au serveur, demande une page. Celle-ci lui est directement servie par une réponse statique préparée auparavant.

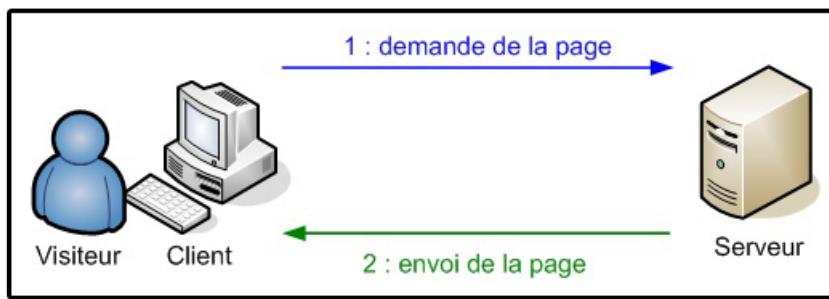


Figure III.1: Architecture d'un site statique

Contrairement, un site web dynamique pourra être accédé par plusieurs utilisateurs distincts. On prévoit un utilisateur de type administrateur dont le rôle est de gérer le contenu. Un site dynamique possède également souvent une plate-forme d'administration que l'internaute ne voit pas où le propriétaire du site peut changer le contenu de son site s'il le souhaite. De plus, un site web dynamique est un site dont les pages sont générées dynamiquement à la demande. Le contenu est obtenu en combinant l'utilisation d'un langage de scripts ou de programmation et une base de données. Si on met l'accent sur l'architecture utilisée pour les sites dynamiques, un ordinateur qui se connecte au serveur, demande une page. Celle-ci lui est servie par une réponse dynamiquement créée selon la requête client et après l'accès à la base de données.

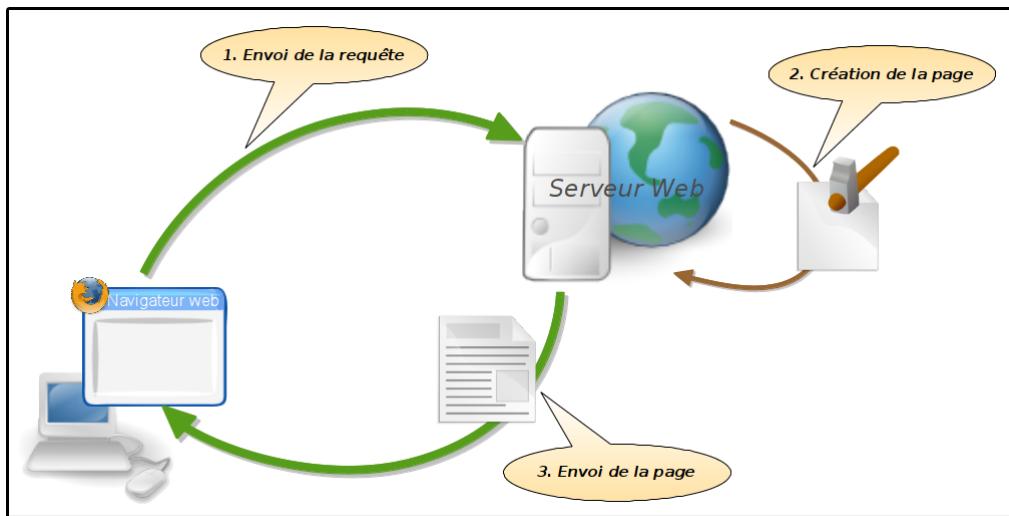


Figure III.2: Architecture d'un site dynamique

## 2.2 Architecture de site web dynamique - le modèle MVC

Avant de se lancer dans le développement d'un site web dynamique, il est indispensable de choisir un motif de conception (pattern design) présenté souvent par le Modèle-Vue-Contrôleur (MVC). Ce modèle est maintenant très répandu et accepté sans contestation, notamment pour la réalisation de sites Web dynamiques, à une organisation satisfaisant le but recherché d'une organisation rigoureuse et logique du code.

Cette architecture regroupe les fonctions nécessaires en trois catégories : un modèle (modèle de données), une vue (présentation, interface utilisateur) et un contrôleur (logique de contrôle, gestion des événements, synchronisation) :

1. **Le modèle (modèle de données):** Le modèle représente le cœur (algorithmique) de l'application: traitements des données, interactions avec la base de données, etc. Il décrit les données manipulées par l'application. Il regroupe la gestion de ces données et est responsable de leur intégrité. La base de données sera l'un de ses composants. Le modèle comporte des méthodes standards pour mettre à jour ces données (insertion, suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ces données.
2. **La vue (présentation, interface utilisateur):** Ce avec quoi l'utilisateur interagit se nomme précisément la vue. Sa première tâche est de présenter les résultats renvoyés par le modèle. Sa seconde tâche est de recevoir toute action de l'utilisateur. Ces différents événements sont envoyés au contrôleur. La vue n'effectue pas de traitement, elle se contente d'afficher les résultats des traitements effectués par le modèle et d'interagir avec l'utilisateur.
3. **Le contrôleur (logique de contrôle, gestion des événements, synchronisation):** Le contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de la vue et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle afin que les données affichées se mettent à jour.

Cette architecture a pas mal d'avantages pour qu'elle reste toujours la plus utilisée dans le monde de développement Web étant donnée qu'elle se caractérise par :

- L'allégement du poste de travail.
- La prise en compte de l'hétérogénéité des plates-formes (serveurs, clients, langages, etc.).
- L'introduction de clients dits " légers " (plus liée aux technologies Intranet/HTML qu'au 3-tiers proprement dit).
- Une meilleure répartition de la charge entre les différents entités clients et serveurs.

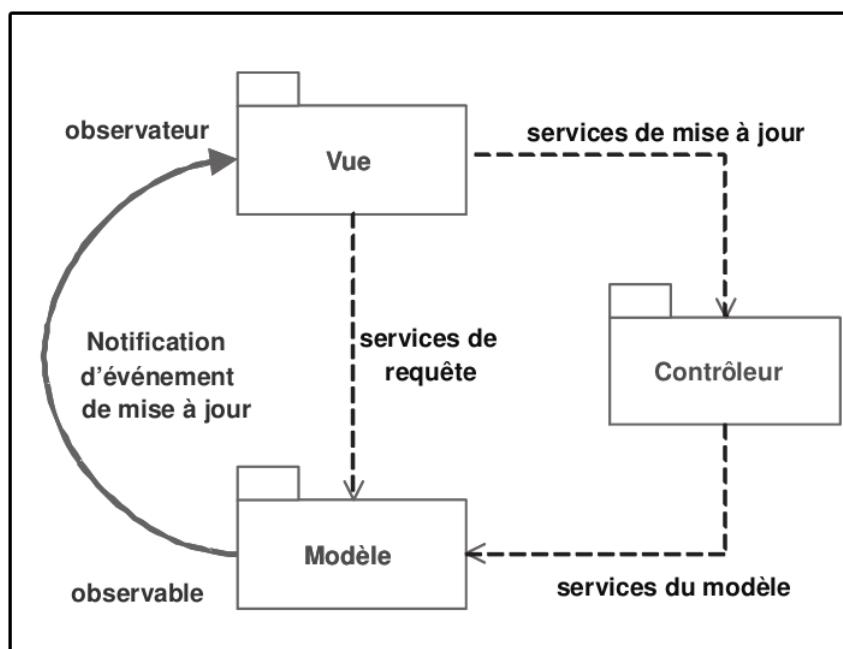


Figure III.3: Schéma de l'architecture MVC

## 2.3 Symfony2 et le modèle MVC

Symfony2 est un puissant framework qui nous permet de réaliser un site web dynamique et complexe rapidement, d'une façon structurée, avec un code clair et maintenable et en suivant l'architecture proposée par le modèle MVC, la figure suivante illustre le parcours complet d'une requête dans Symfony2:

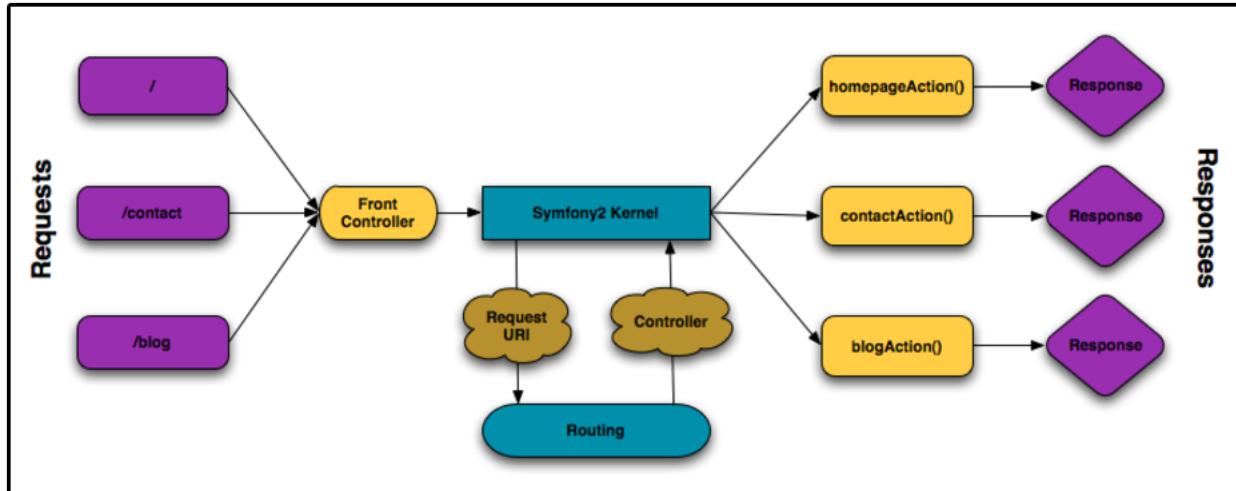


Figure III.4: Parcours complet d'une requête dans Symfony2

## 2.4 Conception préliminaire des interfaces - maquettes

Une maquette est un produit jetable donnant aux utilisateurs une vue concrète mais non définitive de la future interface de l'application. Elle est développée rapidement afin d'améliorer la relation développeur-client. Les maquettes de notre application sont les suivantes:

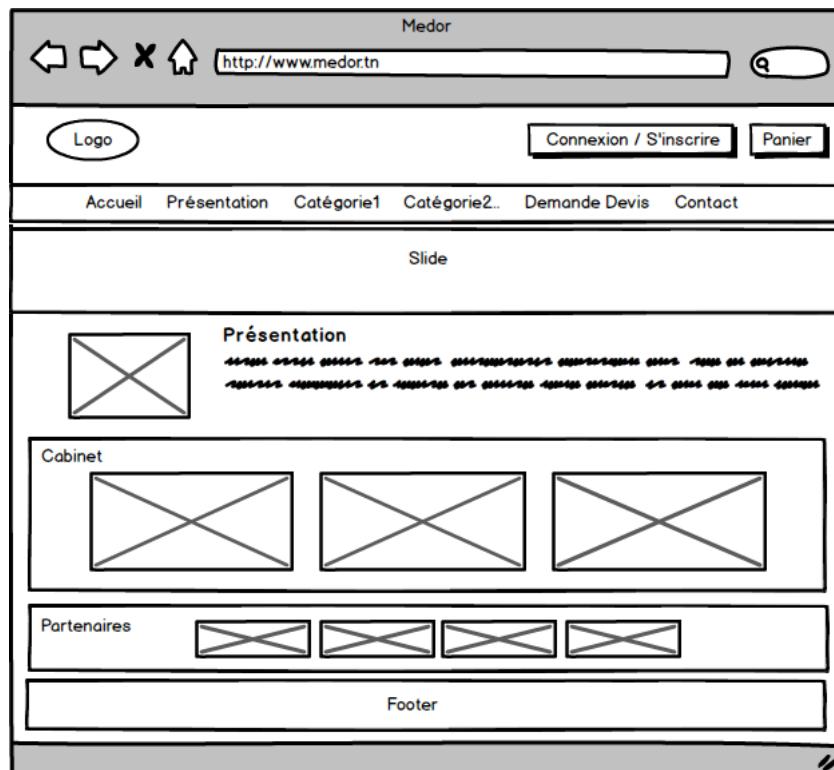


Figure III.5: Maquette - Page Accueil

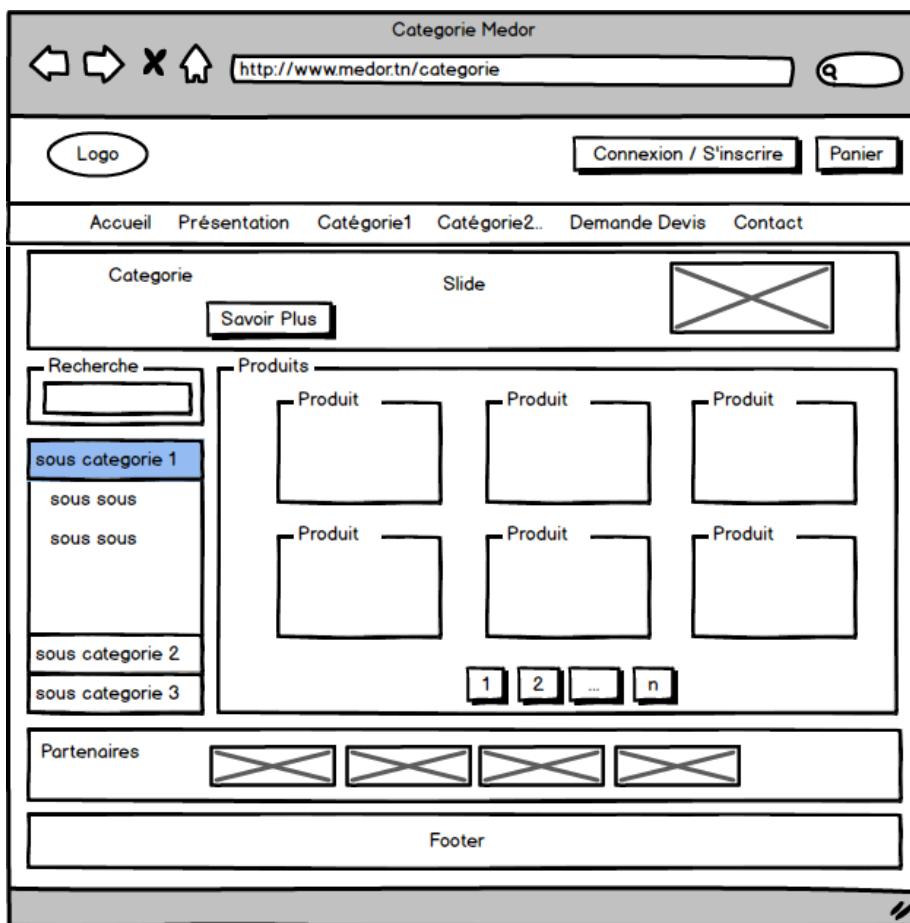


Figure III.6: Maquette - Page categorie

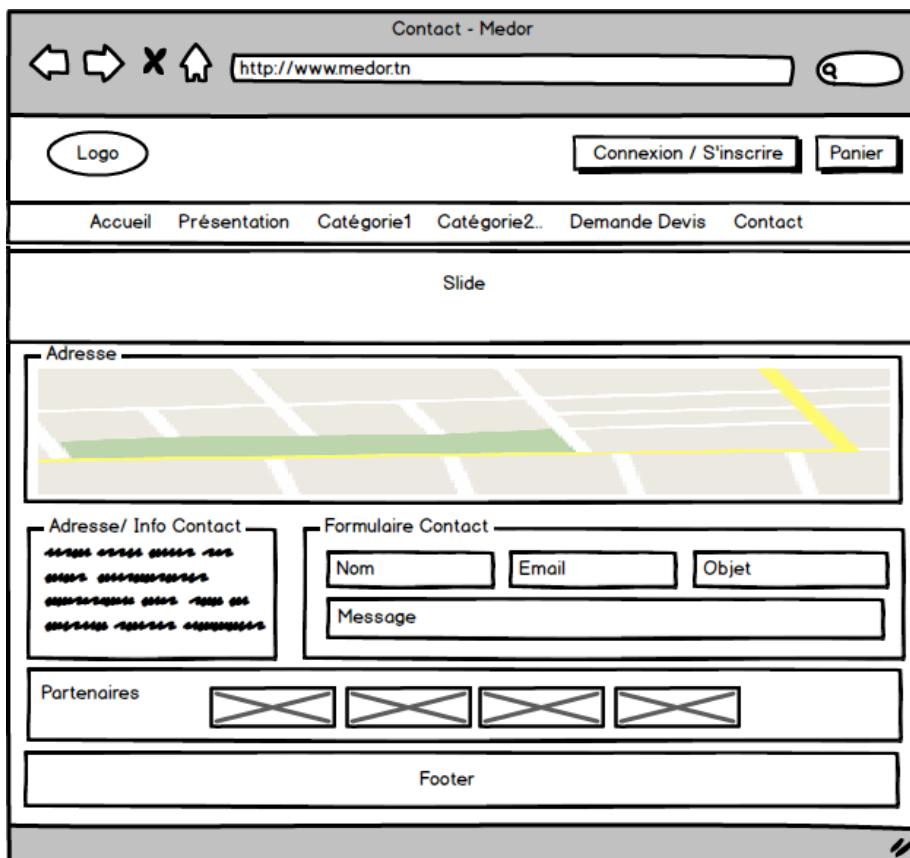


Figure III.7: Maquette - Page Contact

## 3 Conception détaillée

### 3.1 Identification des acteurs

Les acteurs qui interagissent avec le système sont :

- **le visiteur** : Il possède toutes les fonctions de l'internaute mais il peut créer un compte client.
- **L'internaute** : La personne qui visite le site peut chercher un produit ou contacter la société...
- **Le client** : L'internaute connecté qui peut passer une commande. Il s'agit bien sûr de l'acteur le plus important, celui pour lequel le site existe !
- **Le responsable commercial** : L'employé qui est en charge d'ajouter les produits et catégories.
- **L'administrateur** : le responsable du bon fonctionnement du site.

### 3.2 Identification des cas d'utilisations

Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Il est une unité significative de travail. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs (actors), ils interagissent avec les cas d'utilisation (use cases). Ce diagramme permet d'identifier les possibilités d'interaction entre le système et les acteurs. Il représente toutes les fonctionnalités que le système doit fournir.

#### 3.2.1 Modèle global de cas d'utilisation

Pour affiner le diagramme de cas d'utilisation, UML définit trois types de relations standardisées entre cas d'utilisation :

- Une relation d'inclusion, formalisée par le mot-clé «include» : le cas d'utilisation de base en incorpore explicitement un autre, de façon obligatoire.
- Une relation d'extension, formalisée par le mot-clé «extend» : le cas d'utilisation de base en incorpore implicitement un autre, de façon optionnelle.
- Une relation de généralisation/specialisation : les cas d'utilisation descendants héritent de la description de leur parent commun. Chacun d'entre eux peut néanmoins comprendre des interactions spécifiques supplémentaires.

Types de relations :

- L'association (trait plein avec ou sans flèche) entre acteurs et cas d'utilisation.
- La dépendance (flèche pointillée) entre cas d'utilisation, avec les mots-clés «extend» ou «include».
- La relation de généralisation (flèche fermée vide) entre cas d'utilisation.

La figure ci-dessous illustre le diagramme de cas d'utilisation globale, il nous permet d'obtenir une vision globale du comportement fonctionnel de l'application web.

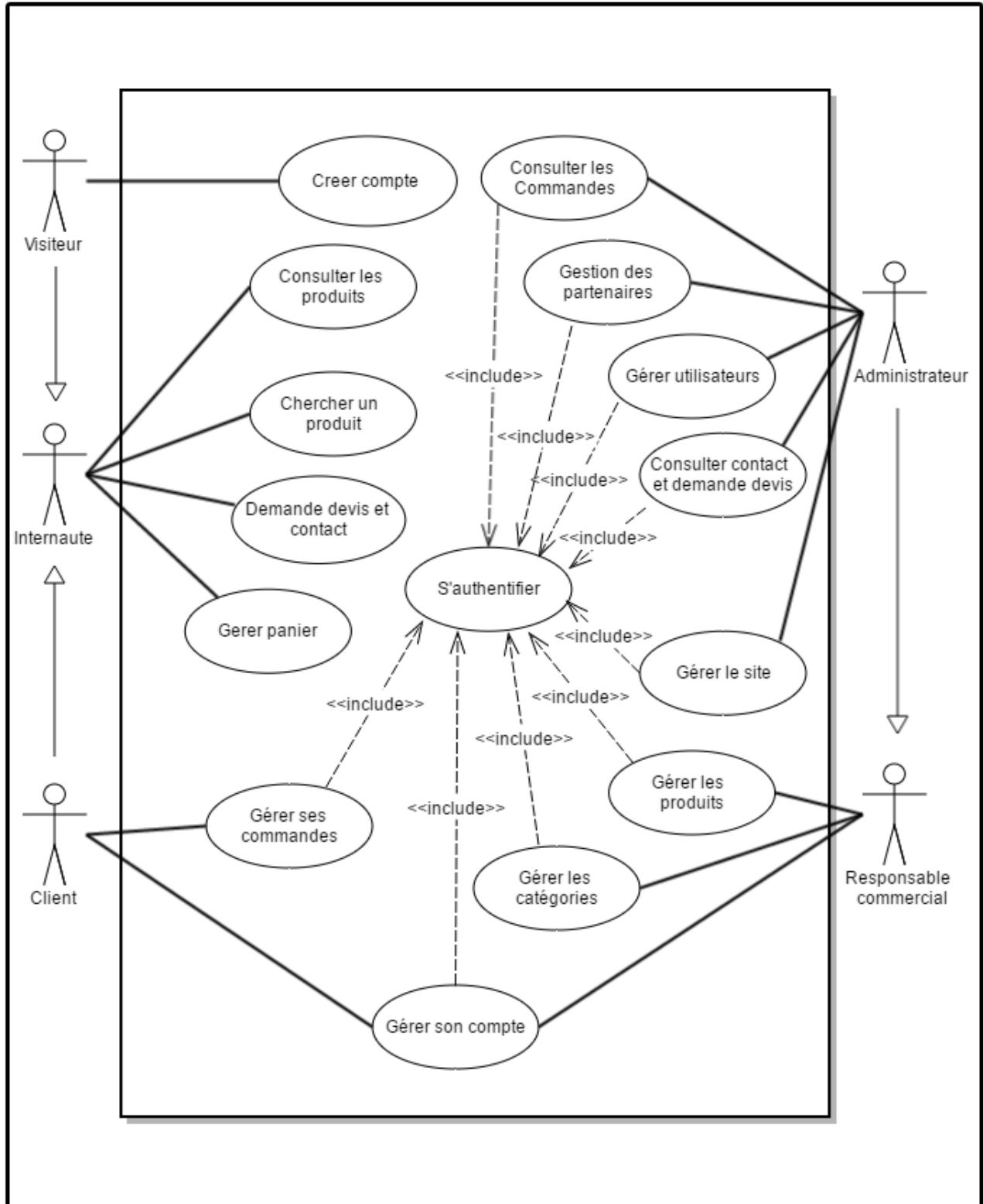


Figure III.8: Diagramme de cas d'utilisation globale

Le tableau suivant représente les différents cas d'utilisation, les acteurs impliqués et une description pour chaque cas :

Cas d'utilisation	Acteur(s)	Description
Créer Compte	Visiteur	Un visiteur peut créer un compte.
S'authentifier	Internaute	L'internaute doit s'authentifier pour effectuer des tâches que seul un utilisateur peut les faire.
Consulter les produits	Internaute - Client	L'internaute ou le client peut voir la liste des produits dans le site.
Chercher un produit	Internaute - Client	L'internaute ou le client a la possibilité de chercher un produit par son code ou son nom.
Contact et demande devis	Internaute - Client	L'internaute ou le client peut contacter la société ou demande un devis.
Gérer panier	Internaute - Client	L'internaute ou le client peut enregistrer les produits qu'il intéressent dans un panier virtuel.
Gérer ses commandes	Client	Le client termine l'achat en ligne avec la validation du commande.
Gérer son compte	Client - Administrateur - Responsable commercial	Les acteur peuvent gérer leur comptes et modifier des information tel que l'email le mot de passe...
Gérer les catégories	Administrateur - Responsable commercial	Un responsable ou un administrateur a le droit de gérer les catégories.
Gérer les produits	Administrateur - Responsable commercial	Un responsable ou un administrateur a le droit de gérer les produits.
Gérer le site	Administrateur	L'administrateur peut gérer le site dans la partie « Backend ».
Consulter contact et demande devis	Administrateur	L'administrateur peut consulter et rependre au messages et demande devis reçue.
Gérer utilisateurs	Administrateur	L'administrateur peut gérer les comptes des membres.
Gestion des partenaires	Administrateur	L'administrateur peut gérer les partenaires de la société.
Consulter les Commandes	Administrateur	L'administrateur peut consulter et manipuler les commandes effectuées.

Table III.1: Tablau des cas d'utilisation

### 3.2.2 Description détaillée des cas d'utilisation

Dans cette section nous allons présenter une description détaillée sur quelques cas d'utilisation.

#### 3.2.2.1 Cas d'utilisation « Créer compte » - « S'authentifier »

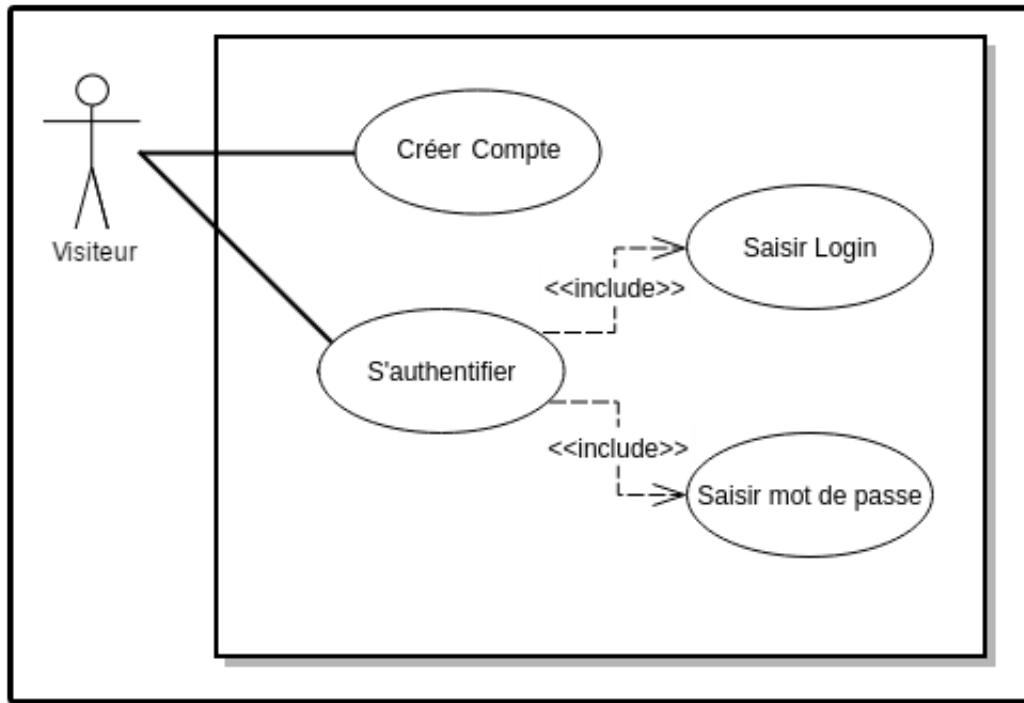


Figure III.9: Diagramme des cas d'utilisation « Créer compte » - « S'authentifier »

**Description textuelle de cas d'utilisation « Créer compte » :**

Sommaire d'identifiant	
Acteur(s)	Visiteur
Objectif	Le visiteur doit créer un compte pour effectuer des tâches que seul un utilisateur peut les faire.
Description d'enchainements	
Pré conditions	Le visiteur n'est pas inscrit.
Post-conditions	Le visiteur est soit inscrit ou non.
Scénario nominal	1- Le visiteur accéder au site et demande de créer un compte. 2- Le Système affiche un formulaire d'inscription. 3- Le visiteur remplir le formulaire par les champs nécessaire. 4- Le Système créer un compte client.
Alternatives	3.1- Le visiteur ne peut être inscrit : un champ vide. 3.2- Le Système revient sur l'affichage de la page de création de compte.

Description textuelle de cas d'utilisation « S'authentifier »:

Sommaire d'identifiant	
Acteur(s)	L'internaute
Objectif	L'internaute doit s'authentifier pour effectuer des tâches seul l'utilisateur peut les faire.
Description d'enchaînements	
Pré conditions	Néant.
Post-conditions	L'internaute est soit authentifié ou non.
Scénario nominal	1- L'internaute accède à la page d'identification. 2- Le Système valide l'authenticité de l'internaute. 3- L'internaute est authentifié.
Alternatives	2.1- L'internaute ne peut être authentifié : un champ invalide. 2.2- Le Système revient sur l'affichage de la page d'identification.

### 3.2.2.2 Les cas d'utilisation consulter produits et gérer panier

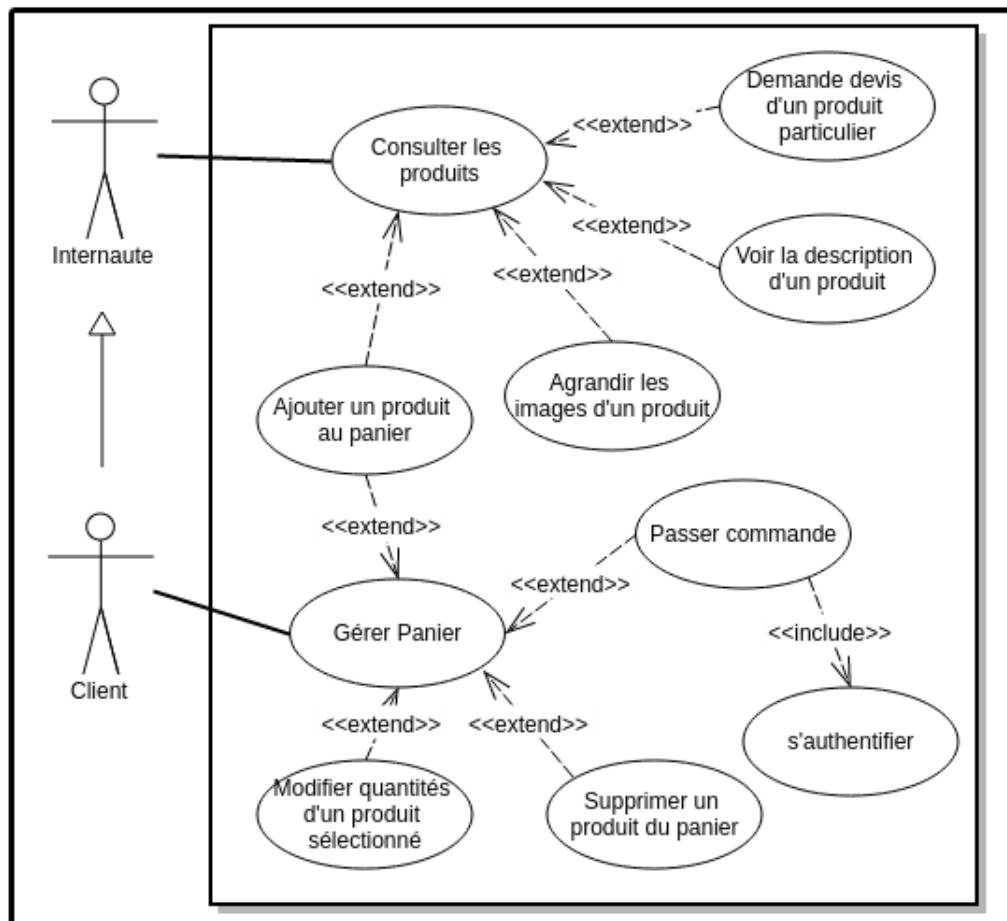


Figure III.10: Diagramme des cas d'utilisation « Consulter les produits »

Description textuelle des cas d'utilisation « Consulter les produits » - « Gérer panier » :

<b>Sommaire d'identifiant</b>	
<b>Acteur(s)</b>	L'internaute - Le client
<b>Objectif</b>	L'internaute peut consulter les catégories, les produits...et si il est intéressé par un produit, il peut l'enregistrer dans un panier virtuel qui peut le gérer à tout moment.
<b>Description d'enchaînements</b>	
<b>Pré conditions</b>	Les produits existent.
<b>Post-conditions</b>	un panier contient des produits sélectionnés pour passer à l'étape suivante ou un panier vide.
<b>Scénario nominal</b>	<p>1- L'internaute ou le client accède à la page des catégories et produits.</p> <p>2- L'internaute ou le client enregistre les produits qui l'intéressent dans un panier virtuel.</p> <p>3- Le Système lui affiche l'état de son panier. Chaque produit qui a été sélectionné est présenté sur une ligne, avec son nom, son image, son prix unitaire est affiché, la quantité, et le prix total de la ligne est calculé. Le total général est calculé par le Système et affiché en bas du panier avec le nombre des produits.</p> <p>5- L'Internaute ou le client continue ses achats.</p>
<b>Alternatives</b>	<p>3.1- le panier est vide. + Le Système affiche un message « Votre panier est vide ».</p> <p>3.2- L'internaute ou le client modifie les quantités des lignes du panier, ou en supprime. + Le cas d'utilisation reprend à l'étape 3 du scénario nominal.</p> <p>3.3- L'internaute fait passer une commande, le total à payer est affiché. + L'internaute s'identifie en tant que Client + L'internaute doit créer un compte client.</p>

### 3.2.2.3 Le cas d'utilisation « Chercher un produit »

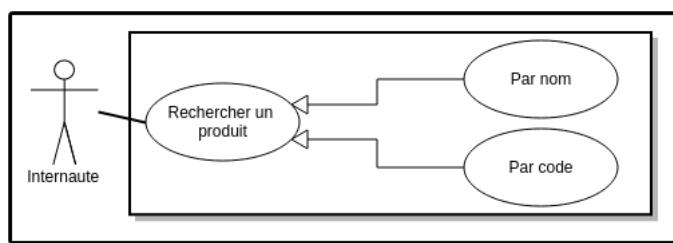


Figure III.11: Diagramme de cas d'utilisation « Chercher un produit »

Description textuelle de cas d'utilisation « Chercher un produit » :

Sommaire d'identifiant	
Acteur(s)	L'internaute - Le client
Objectif	L'internaute ou le client cherche un produit.
Description d'enchaînements	
Pré conditions	Produit doit exister.
Post-conditions	Produit trouvée.
Scénario nominal	1- L'internaute lance une recherche rapide à partir des mots-clés : un nom, un code. 2- Le système affiche une page de résultat. 3- L'internaute sélectionne un produit. 4- Le système lui présente une fiche détaillée pour le produit sélectionnée.
Alternatives	1.1- Le Système n'a pas trouvé le produit correspondant à la recherche. + Le Système affiche un message « Pas de produits trouvés ». 3.1- L'Internaute n'est pas intéressé par le résultat de la recherche.

### 3.2.2.4 Le cas d'utilisation « Gérer ses commandes »

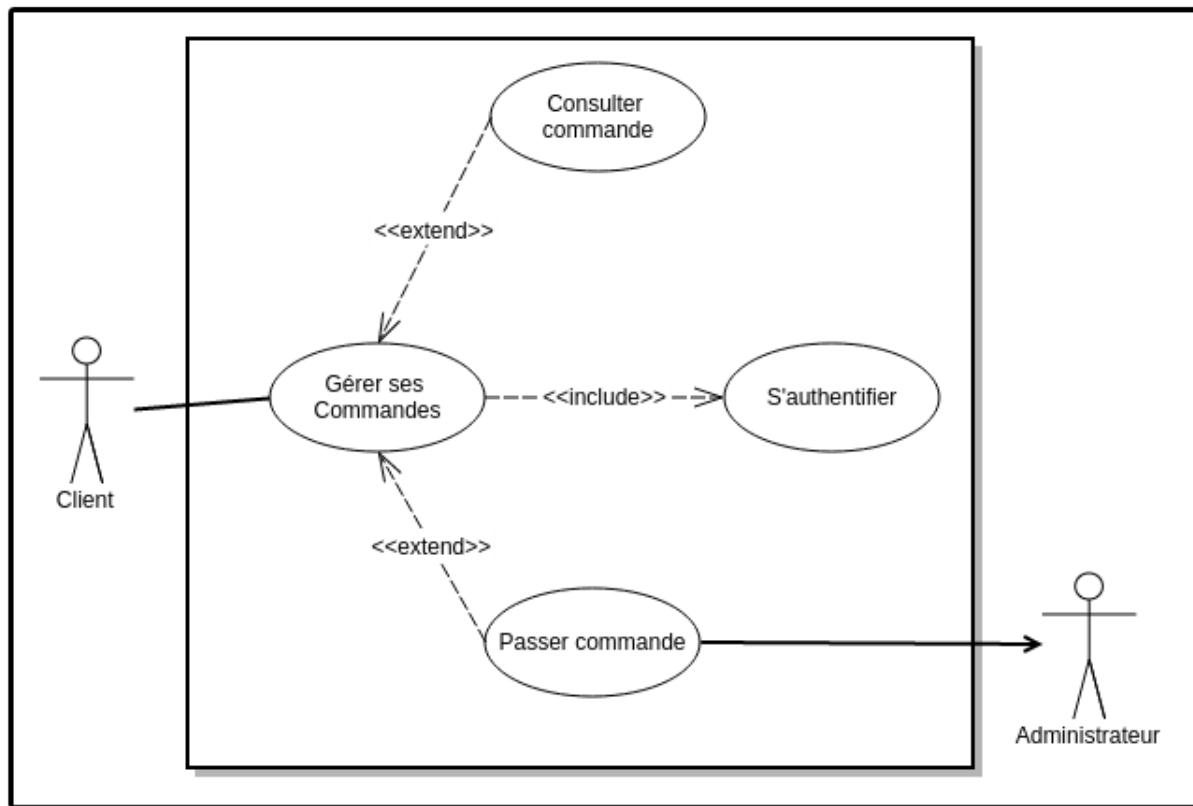


Figure III.12: Diagramme de cas d'utilisation « Gérer ses Commandes »

Description textuelle de cas d'utilisation « Gérer ses commandes » :

Sommaire d'identifiant	
Acteur(s)	Le client
Objectif	Le client peut à tout moment effectuer une commande.
Description d'enchaînements	
Pré conditions	L'acteur doit s'authentifier. Panier non vide.
Post-conditions	Commande effectué et enregistrée.
Scénario nominal	1- Le client choisit de faire une commande. 2- Le système affiche une page contient les produits de sa panier. 3- Le client confirme la commande. 4- Le système confirme et enregistrer la commande, offre la possibilité d'imprimer le reçu d'achat et envoie un email à l'administrateur.
Alternatives	3.1- le client annule sa commande, le système revient sur l'affichage du panier et le cas d'utilisation se termine avec échec.

### 3.2.2.5 Le cas d'utilisation « Gérer son compte »

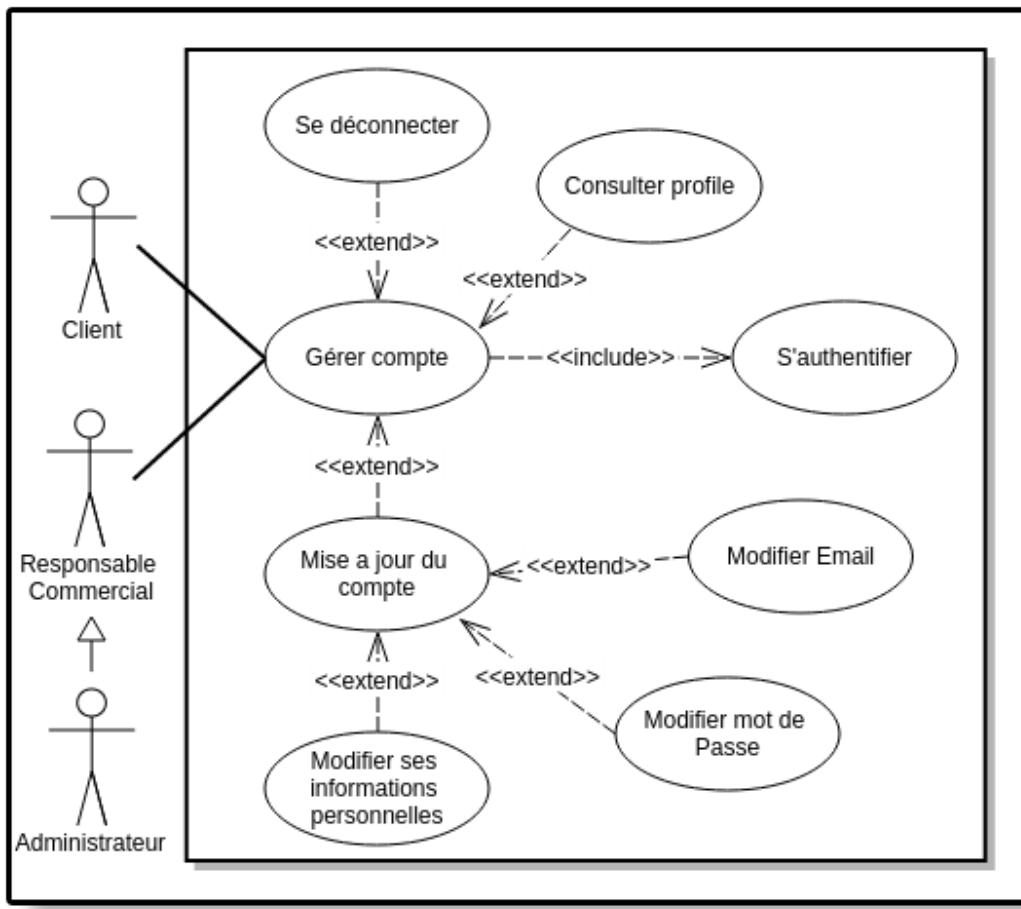


Figure III.13: Diagramme de cas d'utilisation « Gérer son compte »

Description textuelle de cas d'utilisation « Gérer son compte » :

Sommaire d'identifiant	
Acteur(s)	Le client - le responsable commercial - L'administrateur
Objectif	L'acteur peut gérer son compte et modifier certain information (Email, Mot de passe...).
Description d'enchaînements	
Pré conditions	L'acteur doit s'authentifier.
Post-conditions	Compte mise à jour.
Scénario nominal	1- Le client choisit de modifier certain information de son compte (nom, adresse,téléphone ....) 2- Le client confirme cette modification. 3. Le système le valide et mis a jour la BD ainsi que le compte.
Alternatives	2.1- l'acteur saisie des informations existe ou erroné alors le système revient sur l'affichage de formulaire et modification échoué.

### 3.2.2.6 Les cas d'utilisation gérer catégories et produits

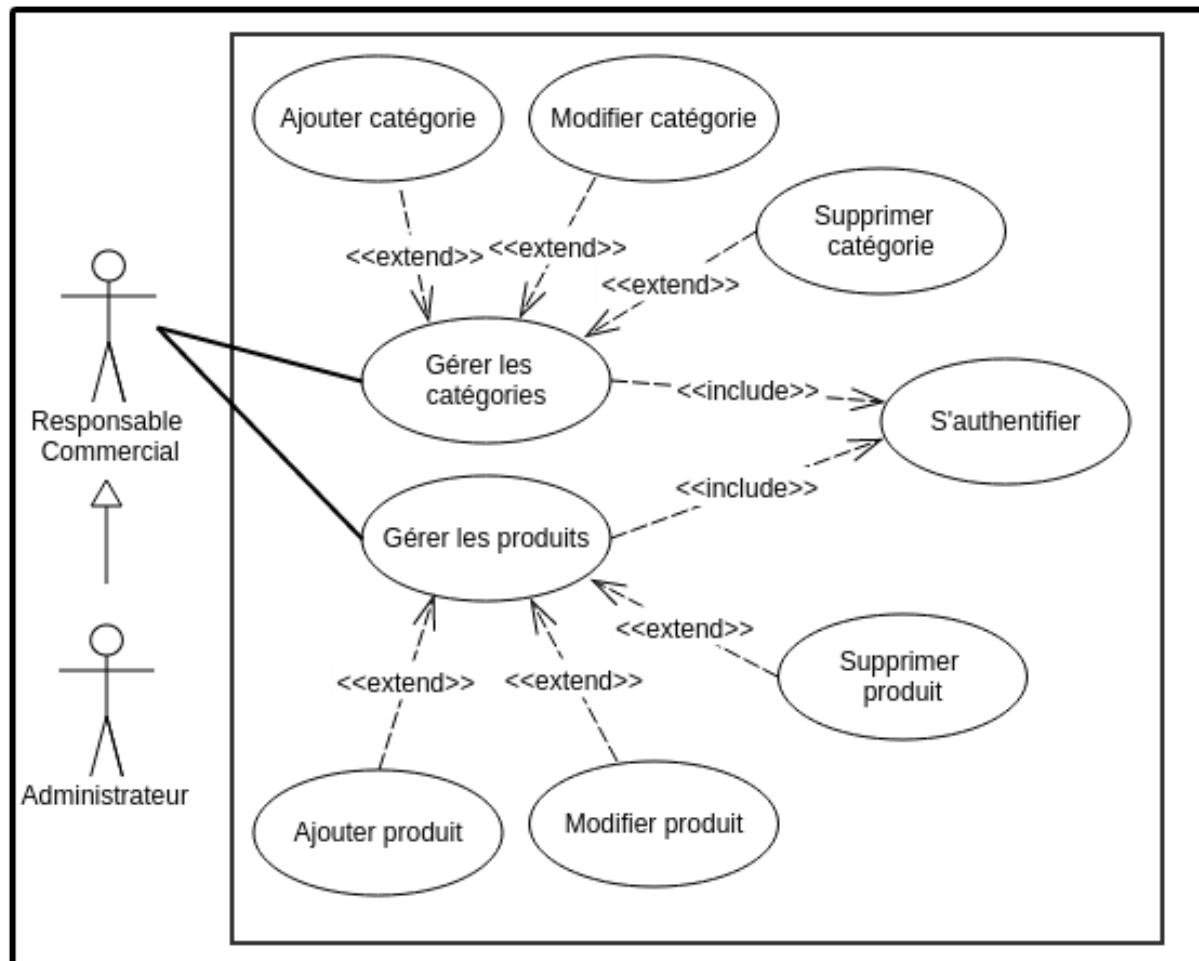


Figure III.14: Diagramme des cas d'utilisation « Gérer les catégories » - « Gérer les produits »

### Description textuelle de cas d'utilisation « Gérer les catégories » :

<b>Sommaire d'identifiant</b>	
<b>Acteur(s)</b>	Le responsable commercial - L'administrateur
<b>Objectif</b>	L'acteur a le droit de gérer les catégories tel que ajouter une catégories, supprimer une catégories ...
<b>Description d'enchainements</b>	
<b>Pré conditions</b>	L'acteur doit s'authentifier.
<b>Post-conditions</b>	Base de données mise à jour.
<b>Scénario nominal</b>	<p>1- Le système affiche les catégories qui existe dans le site.</p> <p>2- L'acteur choisit de modifier quelque catégories.</p> <p>3- Le système affiche le formulaire pour modifier une catégories.</p> <p>4- L'acteur remplit le formulair selon son besoin et confirme la modification.</p> <p>5- Le système valide la modification.</p> <p>6- L'acteur choisit de supprimer quelque catégorie.</p> <p>7. Le système valide la suppression.</p> <p>8- L'acteur choisit d'ajouter des nouvelles catégories, il saisie le nom de catégorie, la catégorie parente (si il y a une) et valide l'ajout.</p> <p>9-Le système valide l'ajout.</p>
<b>Alternatives</b>	<p>2.1 - l'acteur modifie la catégorie, il va remplir un formulaire (nom catégorie, catégorie parente...) :</p> <ul style="list-style-type: none"> <li>+ si il laisse un champ vide alors le système affiche un message « Champ obligatoire est vide ! » et réafficher la page.</li> <li>+ S'il saisie des informations invalide le système affiche un message « Champ invalide ! » et réafficher la page.</li> </ul> <p>8.1 - l'acteur ajout une catégorie, il va remplit un formulaire (nom catégorie, catégorie parente...) :</p> <ul style="list-style-type: none"> <li>+ si il laisse un champ vide alors le système affiche un message « Champ obligatoire est vide ! » et réafficher la page.</li> <li>+ S'il saisie des informations invalide le système affiche un message « Champ invalide ! » et réafficher la page.</li> </ul>

**Description textuelle de cas d'utilisation « Gérer les produits » :**

<b>Sommaire d'identifiant</b>	
<b>Acteur(s)</b>	Le responsable commercial - L'administrateur
<b>Objectif</b>	L'acteur a le droit de gérer les produits tel que ajouter un produit, supprimer un produit ...
<b>Description d'enchaînements</b>	
<b>Pré conditions</b>	L'acteur doit s'authentifier.
<b>Post-conditions</b>	Base de données mise à jour.
<b>Scénario nominal</b>	<p>1- Le système affiche les produits qui existe dans le site.</p> <p>2- L'acteur choisit de modifier quelque produits.</p> <p>3- Le système affiche le formulaire pour modifier un produit.</p> <p>4- L'acteur remplit le formulair selon son besoin et confirme la modification.</p> <p>5- Le système valide la modification.</p> <p>6- L'acteur choisit de supprimer quelque produit.</p> <p>7. Le système valide la suppression.</p> <p>8- L'acteur choisit d'ajouter des nouveaux produits, il saisie le nom de produit, leur catégorie et valide l'ajout.</p> <p>9-Le système valide l'ajout.</p>
<b>Alternatives</b>	<p>2.1 - l'acteur modifie le produit, il va remplir un formulaire (nom produit, catégorie de produit...) :</p> <ul style="list-style-type: none"> <li>+ si il laisse un champ vide alors le système affiche un message « Champ obligatoire est vide ! » et réafficher la page.</li> <li>+ S'il saisie des informations invalide le système affiche un message « Champ invalide ! » et réafficher la page.</li> </ul> <p>8.1 - l'acteur ajout un produit, il va remplit un formulaire (nom produit, catégorie de produit...) :</p> <ul style="list-style-type: none"> <li>+ si il laisse un champ vide alors le système affiche un message « Champ obligatoire est vide ! » et réafficher la page.</li> <li>+ S'il saisie des informations invalide le système affiche un message « Champ invalide ! » et réafficher la page.</li> </ul>

### 3.2.2.7 Le cas d'utilisation « Gérer utilisateurs »

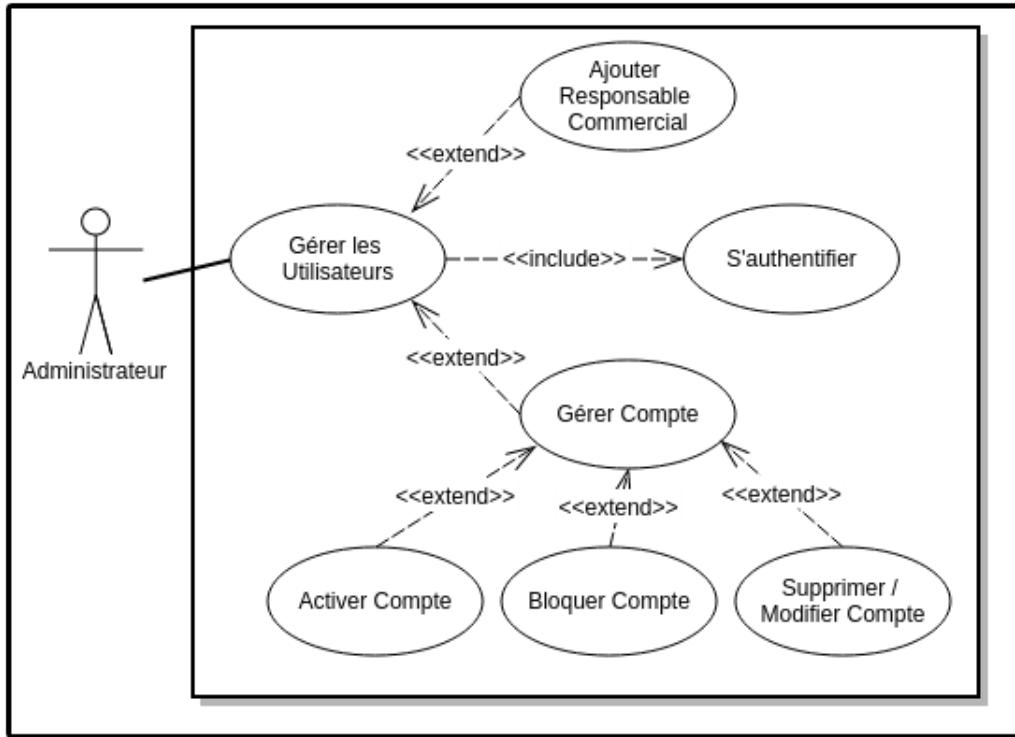


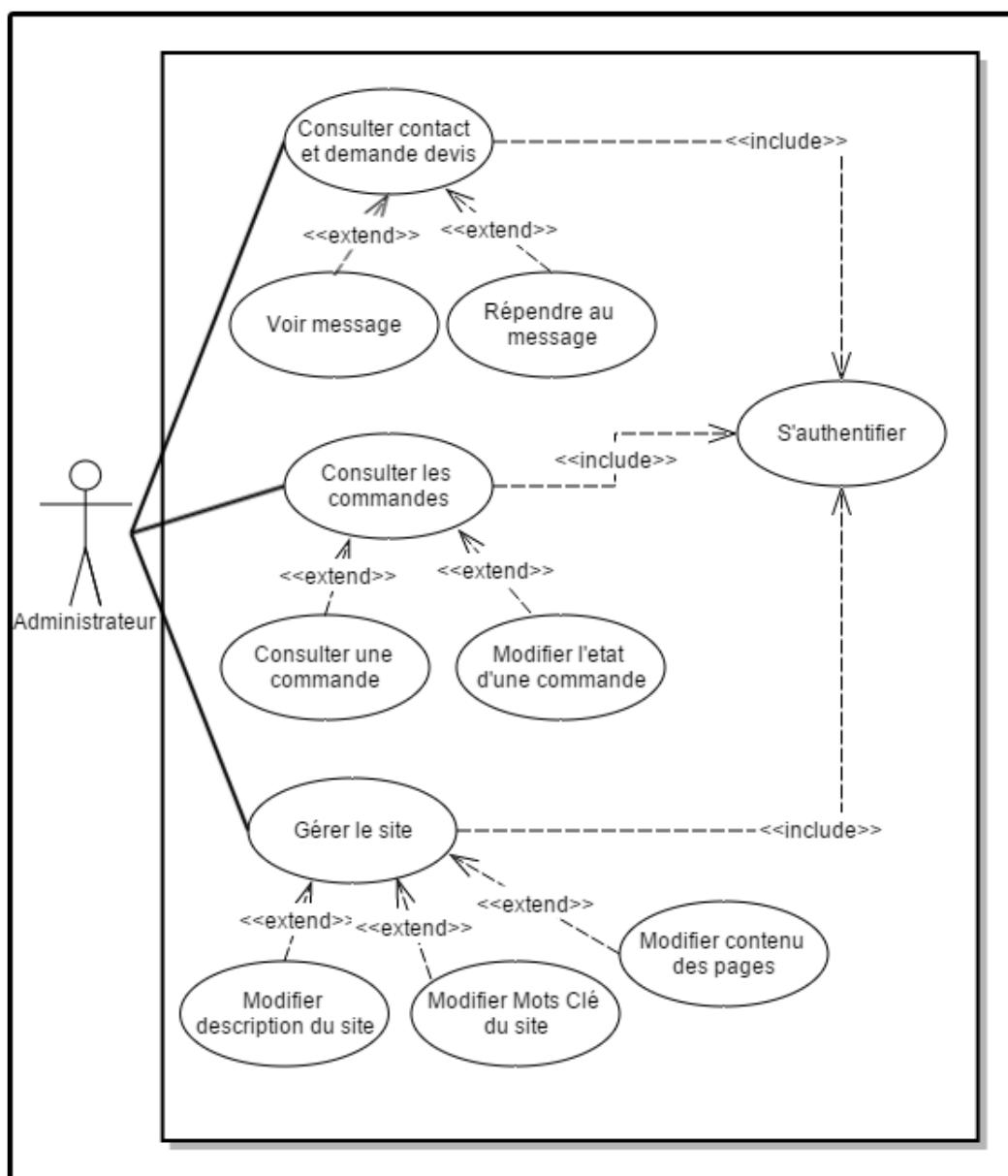
Figure III.15: Diagramme de cas d'utilisation « Gérer les Utilisateurs »

**Description textuelle de cas d'utilisation « Gérer les Utilisateurs » :**

Sommaire d'identifiant	
Acteur(s)	L'administrateur
Objectif	L'administrateur peut gérer les utilisateurs.
Description d'enchaînements	
Pré conditions	L'administrateur doit s'authentifier.
Post-conditions	Base de données mise à jour.
Scénario nominal	<p>1- L'administrateur demande la liste des utilisateurs.</p> <p>2- Le système affiche la liste de tous les utilisateurs.</p> <p>3 - L'administrateur peut choisir une action entre :</p> <ul style="list-style-type: none"> <li>+ Ajouter un responsable commercial                     <ul style="list-style-type: none"> <li>3.1.1 - Le système affiche un formulaire.</li> <li>3.1.2 - L'administrateur remplit le formulaire par les champs nécessaires et valide l'ajout.</li> <li>3.1.3 - Le système valide l'ajout.</li> </ul> </li> <li>+ Gérer un Compte                     <ul style="list-style-type: none"> <li>3.2.1 - Le système affiche un formulaire.</li> <li>3.2.2 - L'administrateur choisit de modifier quelques</li> </ul> </li> </ul>

<b>Scénario nominal</b>	informations sur l'utilisateur ou supprimer le compte ou désactiver le compte ou le reactiver. 3.2.3 - Le système valide la modification.
<b>Alternatives</b>	3.1 - l'administrateur ajoute un commercial : + champ invalide alors le système affiche un message « Champ Invalide ! » 3.2 - l'administrateur modifie un commercial : + si champ invalide alors le système affiche un message « Champ Invalide ! »

### 3.2.2.8 Les cas d'utilisation gérer messages, commandes et site



(a)- « Consulter contact et demande devis » (b)- « Consulter les commandes » (c)- « Gérer le site »

Figure III.16: Diagramme des cas d'utilisation gérer messages, commandes et site

**Description textuelle de cas d'utilisation « Consulter contact et demande devis » :**

<b>Sommaire d'identifiant</b>	
<b>Acteur(s)</b>	L'administrateur
<b>Objectif</b>	L'administrateur peut consulter et rependre au messages et demande devis reçue.
<b>Description d'enchaînements</b>	
<b>Pré conditions</b>	L'administrateur doit s'authentifier.
<b>Post-conditions</b>	Messages consultés.
<b>Scénario nominal</b>	<p>1- Le système affiche la liste des messages et demande devis des clients.</p> <p>2- L'administrateur consulte les messages et demande devis.</p> <p>3- L'administrateur choisit de rependre à un message ou une demande donc le système lui affiche un formulaire.</p> <p>4- L'administrateur remplit les champs et valide le formulaire.</p> <p>5- Le système envoie un email qui contient la réponse de l'administrateur au client ou internaute.</p>
<b>Alternatives</b>	<p>4.1- Si un champ obligatoire est vide le système affiche un message « Champ obligatoire vide ! » et réafficher la page.</p> <p>4.2- Si un champ est invalide le système affiche un message « Champ invalide ! » et réafficher la page.</p>

**Description textuelle de cas d'utilisation « Consulter les commandes » :**

<b>Sommaire d'identifiant</b>	
<b>Acteur(s)</b>	L'administrateur
<b>Objectif</b>	L'administrateur peut consulter les commandes.
<b>Description d'enchaînements</b>	
<b>Pré conditions</b>	L'administrateur doit s'authentifier.
<b>Post-conditions</b>	Commandes consultés.
<b>Scénario nominal</b>	<p>1- Le système affiche la liste des commandes.</p> <p>2- L'administrateur peut voir la liste des produit commandé par chaque client, la date du commande...</p> <p>3- L'administrateur peut changer l'état du commande, il peut le valider, annuler, supprimer...</p>
<b>Alternatives</b>	1.1- Il n'y a pas des commandes à afficher.

### Description textuelle de cas d'utilisation « Gérer le site » :

Sommaire d'identifiant	
<b>Acteur(s)</b>	L'administrateur
<b>Objectif</b>	L'administrateur peut gérer le site dans la partie « Backend ».
Description d'enchaînements	
<b>Pré conditions</b>	L'administrateur doit s'authentifier.
<b>Post-conditions</b>	Base de données mise à jour.
<b>Scénario nominal</b>	<p>1- L'administrateur peut choisir une action entre :</p> <ul style="list-style-type: none"> <li>+ Modifier les paramètres de site           <ul style="list-style-type: none"> <li>1.1- Le système affiche un formulaire du paramètre de site.</li> <li>1.2- L'administrateur choisit de modifier un paramètre, il modifie le champs et valide le formulaire.</li> <li>1.3- Le système valide la modification.</li> </ul> </li> <li>+ Modifier le contenu de site           <ul style="list-style-type: none"> <li>2.1- Le système affiche les pages de site.</li> <li>2.2- L'administrateur choisit de modifier le contenu d'une page donc il choisit la page, faire les modifications et enregistrer.</li> <li>2.3- Le système valide la modification.</li> </ul> </li> </ul>
<b>Alternatives</b>	<ul style="list-style-type: none"> <li>1.2.1- Si l'administrateur laisse un champ obligatoire vide le système affiche un message « Champ obligatoire Vide ! ».</li> <li>1.2.2- Si l'administrateur saisie un champ invalide le système affiche un message « Champ invalide ! ».</li> <li>2.2.1- Si l'administrateur laisse un champ obligatoire vide le système affiche un message « Champ obligatoire Vide ! ».</li> <li>2.2.2- Si l'administrateur saisie un champ invalide le système affiche un message « Champ invalide ! ».</li> </ul>

### 3.3 Diagrammes de séquence système

Le diagramme de séquences est la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML.

Ce diagramme est utilisé pour représenter certains aspects dynamiques d'un système : dans le contexte d'une opération, d'un système, d'un sous-système, d'un cas d'utilisation (un scénario d'un cas d'utilisation) selon un point de vue temporel.

Dans cette phase, et après identification des cas d'utilisation, nous représentons quelques scénarios à l'aide des diagrammes de séquences système.

### 3.3.1 Diagrammes de séquences « Authentification »

Un internaute demande son authentification et le système lui renvoie le formulaire à remplir, cet acteur saisie ses identifiants .Le système vérifie les identifiants s'ils sont valide le système affiche la page d'accueil sinon il envoie un message d'erreur.

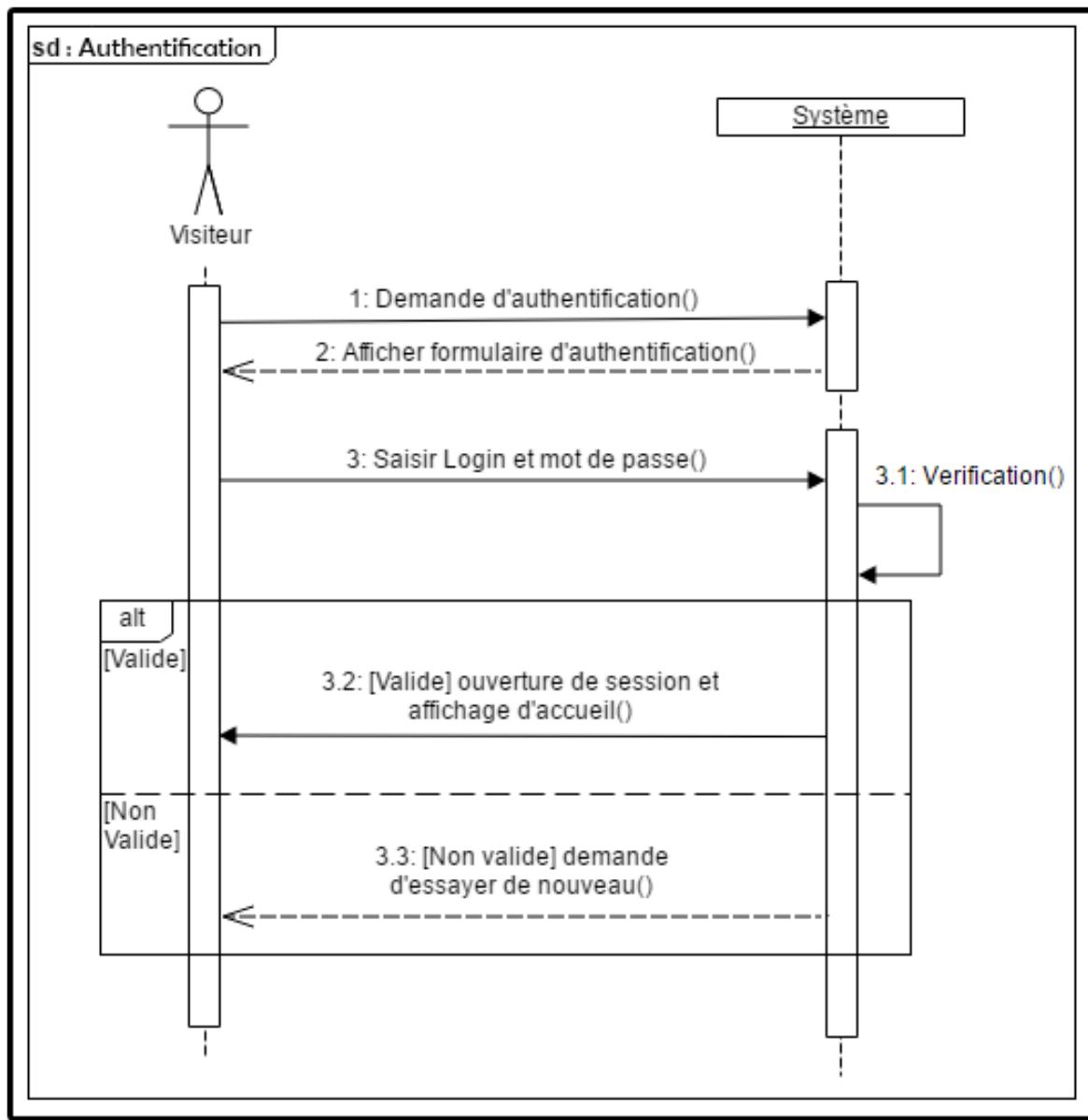


Figure III.17: Diagrammes de séquences de « Authentification »

### 3.3.2 Diagrammes de séquences « Gérer panier »

L'internaute visite la page panier dans laquelle les produits choisis seront affichés, si aucun produit n'a été trouvé un message « votre panier est vide » sera affiché si non l'internaute visualise les produits de son panier et modifie la quantité d'un produit ou supprime un produit

ou même vider son panier, chaque changement ou mis à jours sur un produit du panier sera affiché pour l'internaute.

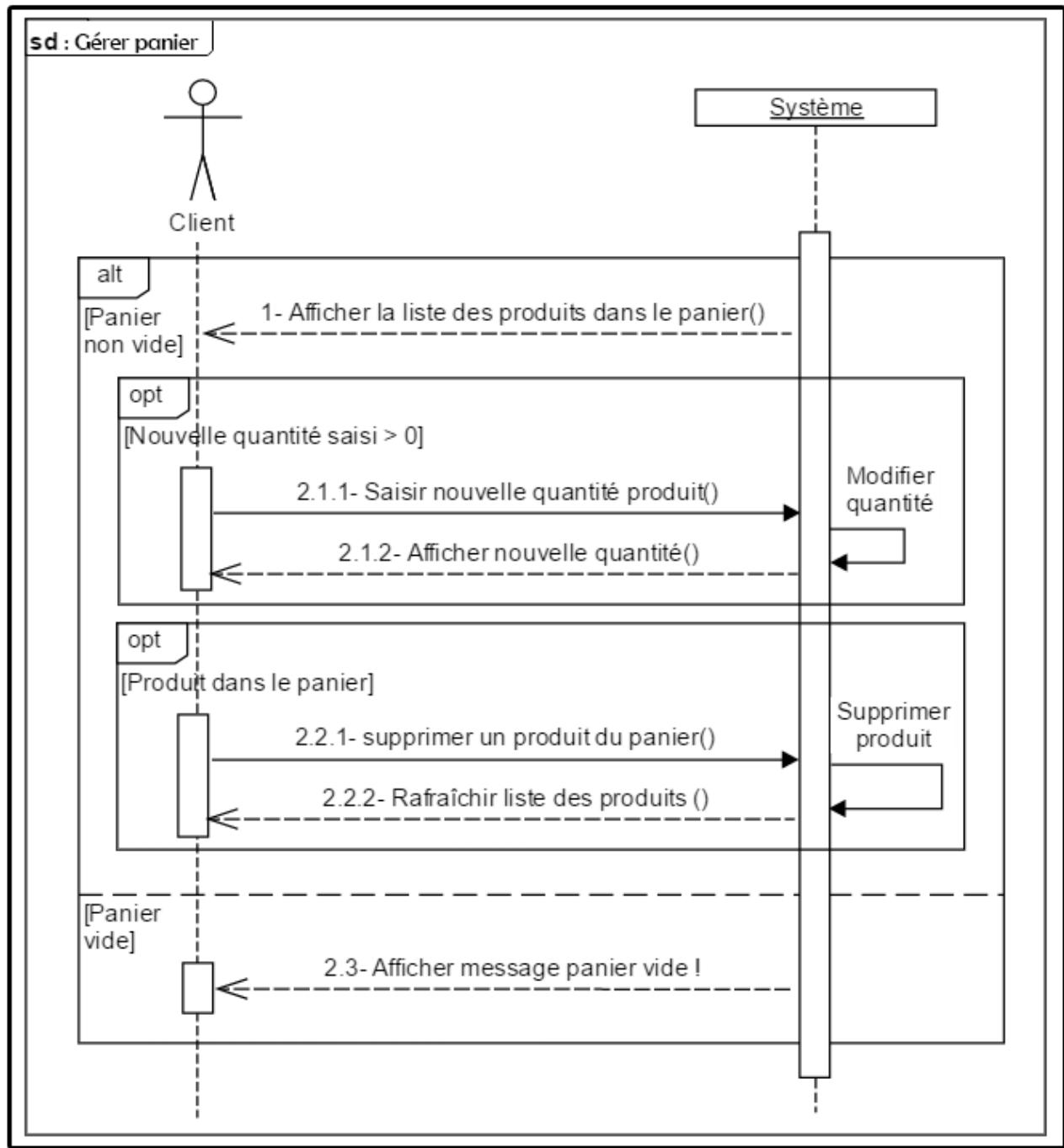


Figure III.18: Diagrammes de séquences de « Gérer panier »

### 3.3.3 Diagrammes de séquences de « Gérer les produits »

Après l'authentification, le responsable commercial consulte la page des produits du partie backend. Il peut ajouter un produit et dans ce cas le système lui affiche un formulaire, il doit le rempli avec les informations du nouveau produit et après la validation le système ajoute le produit à la base de donnée. Le responsable commercial peut aussi choisir de modifier un produits, dans ce cas il sélectionne le produit et il saisi les informations qu'il veut changer ou

il choisit de supprimer le produit et valide, dès que la validation est terminé les changements apportés au produits seront visible.

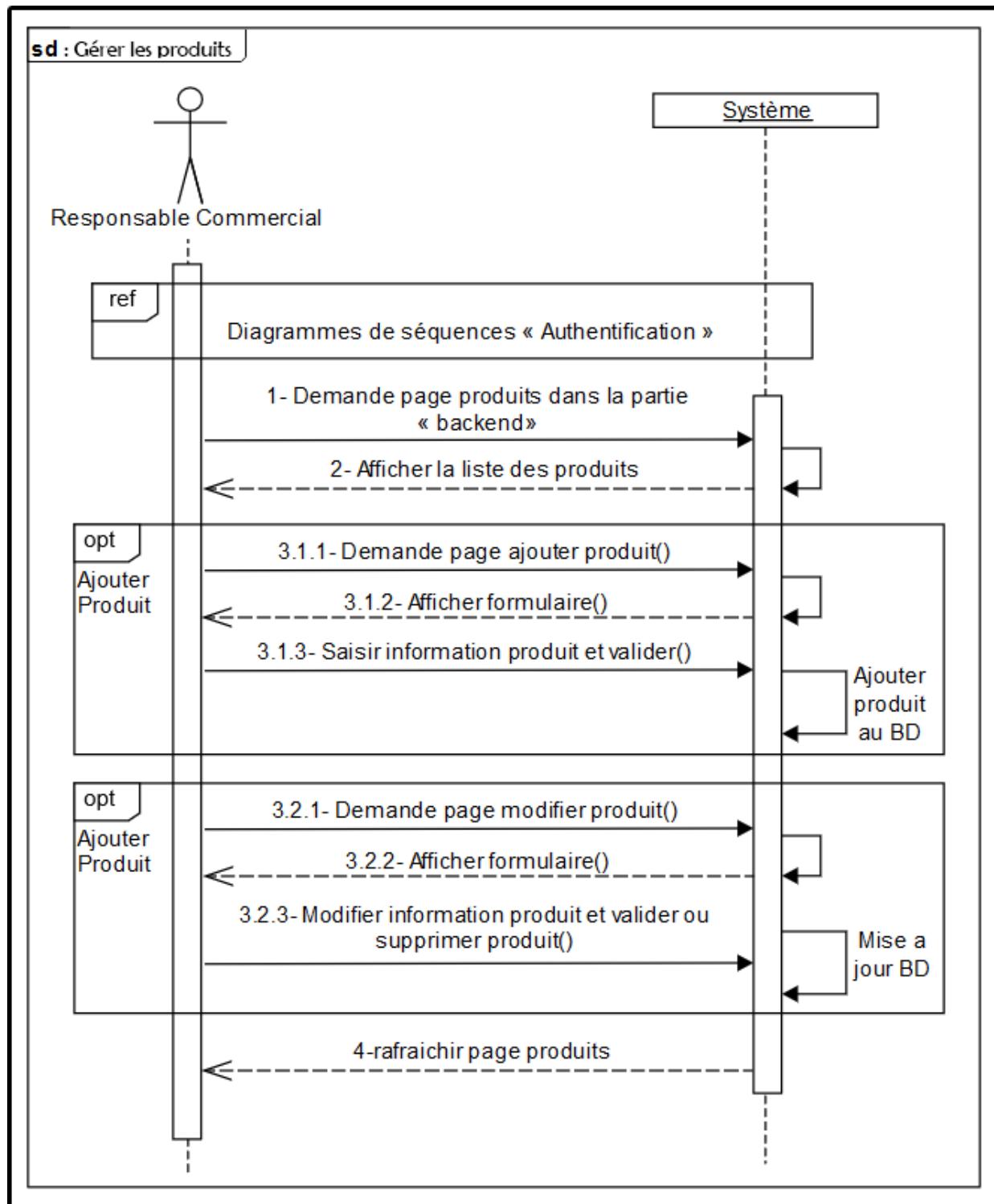


Figure III.19: Diagrammes de séquences de « Gérer les produits »

### 3.4 Diagramme de composant

Un diagramme de composants décrit l'architecture physique et statique d'une application informatique.

le diagramme suivant présente le fonctionnement de notre Solution :

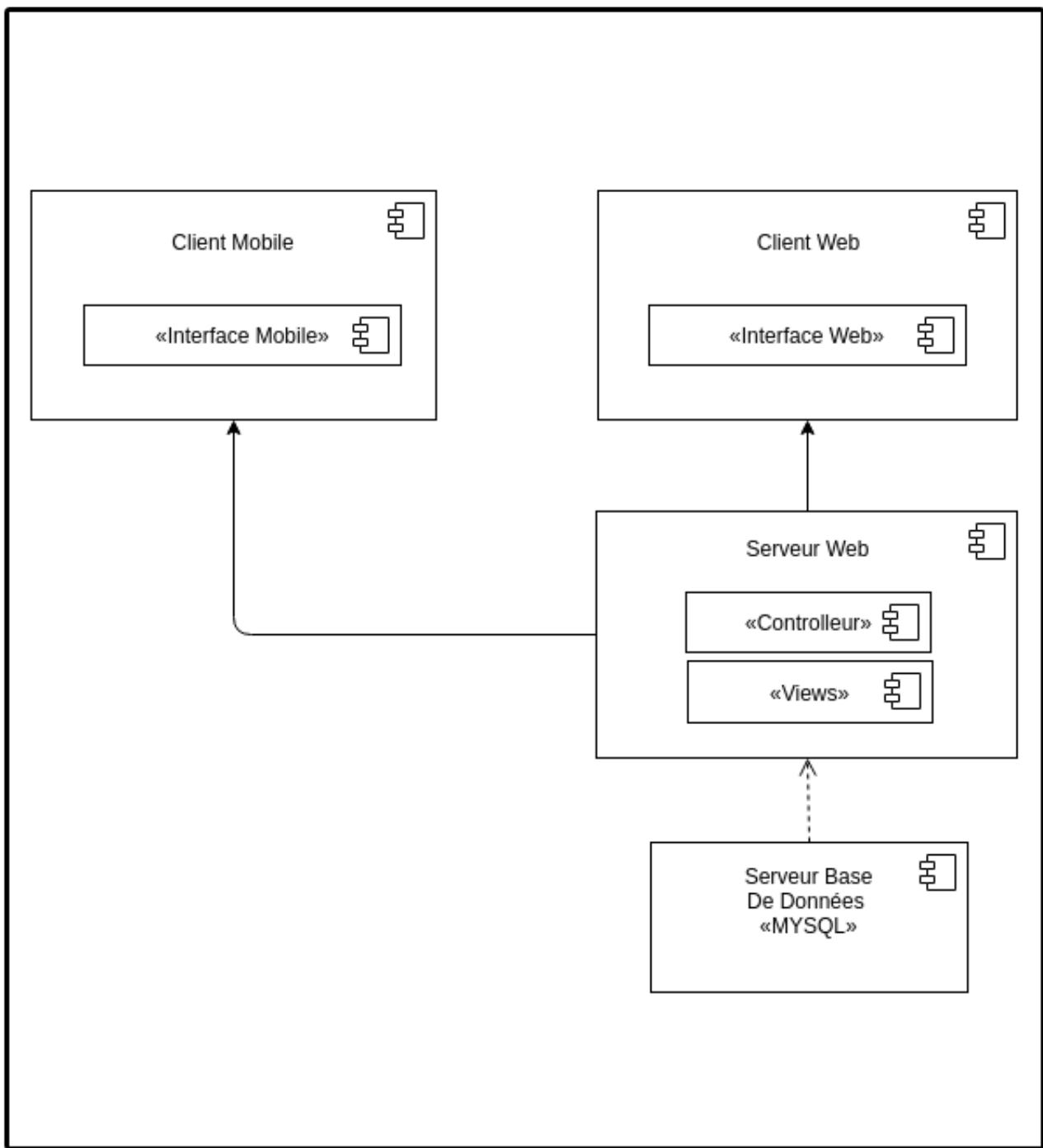


Figure III.20: Diagramme de composant

### 3.5 Diagramme de classe

Un diagramme de classes dans le langage de modélisation unifié (UML) est un type de diagramme de structure statique qui décrit la structure d'un système en montrant le système de classes, leurs attributs, les opérations (ou) les méthodes et les relations entre les classes.

Les figures ci-dessous illustrent les diagrammes de class de notre système selon le modèle MVC :

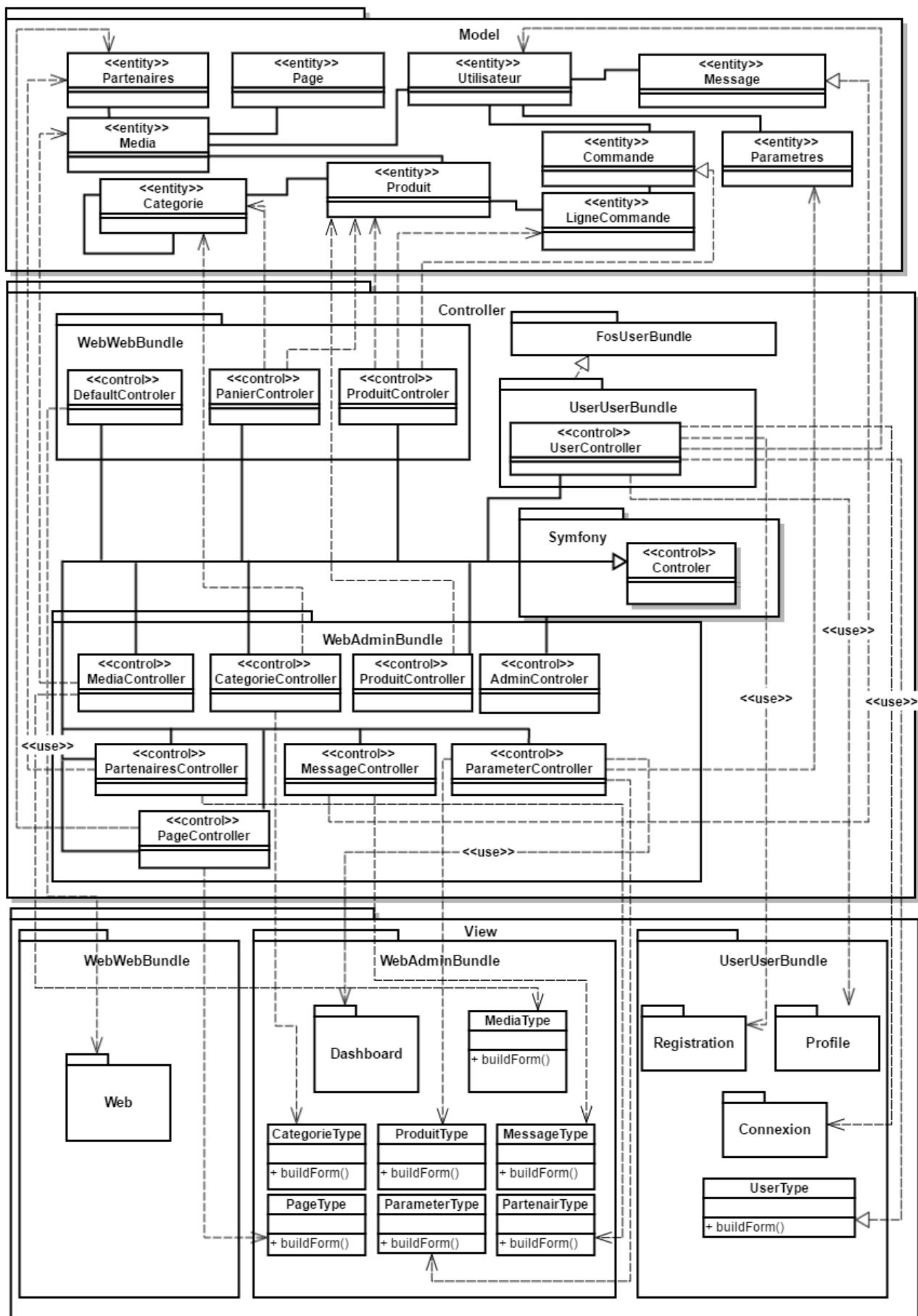


Figure III.21: Organisation de code selon MVC

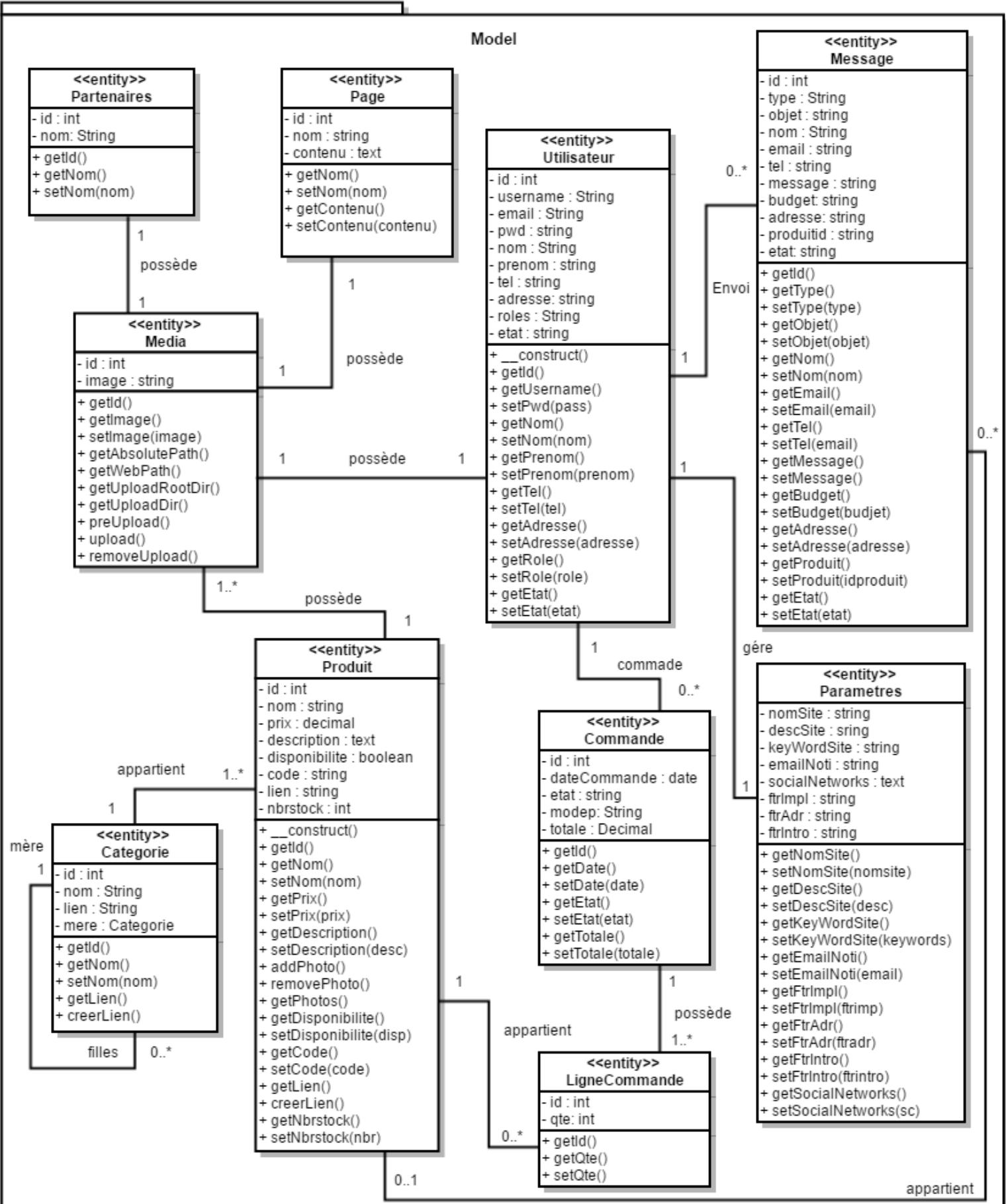


Figure III.22: Diagramme de classe « Model »

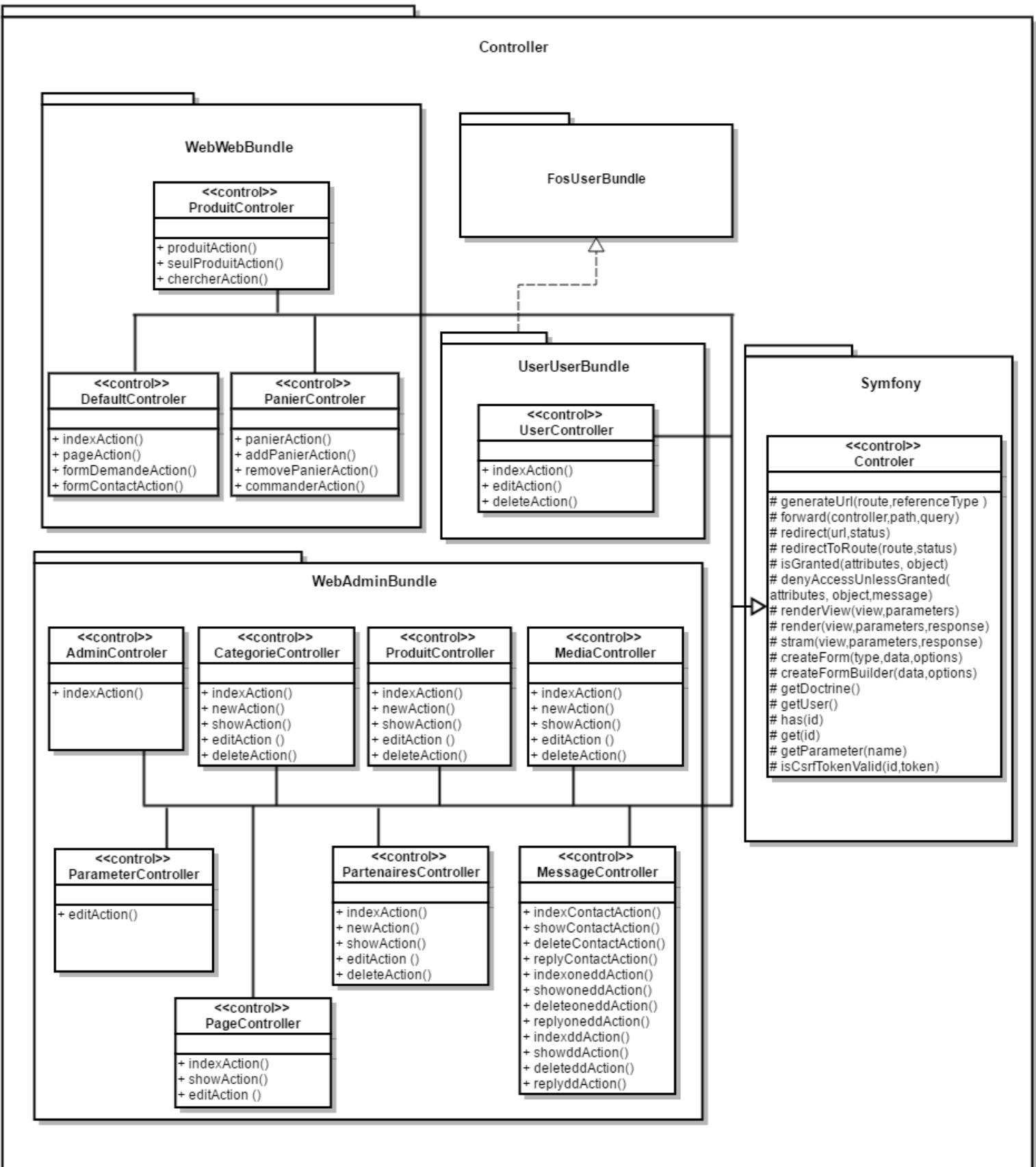


Figure III.23: Diagramme de classe « Controller »

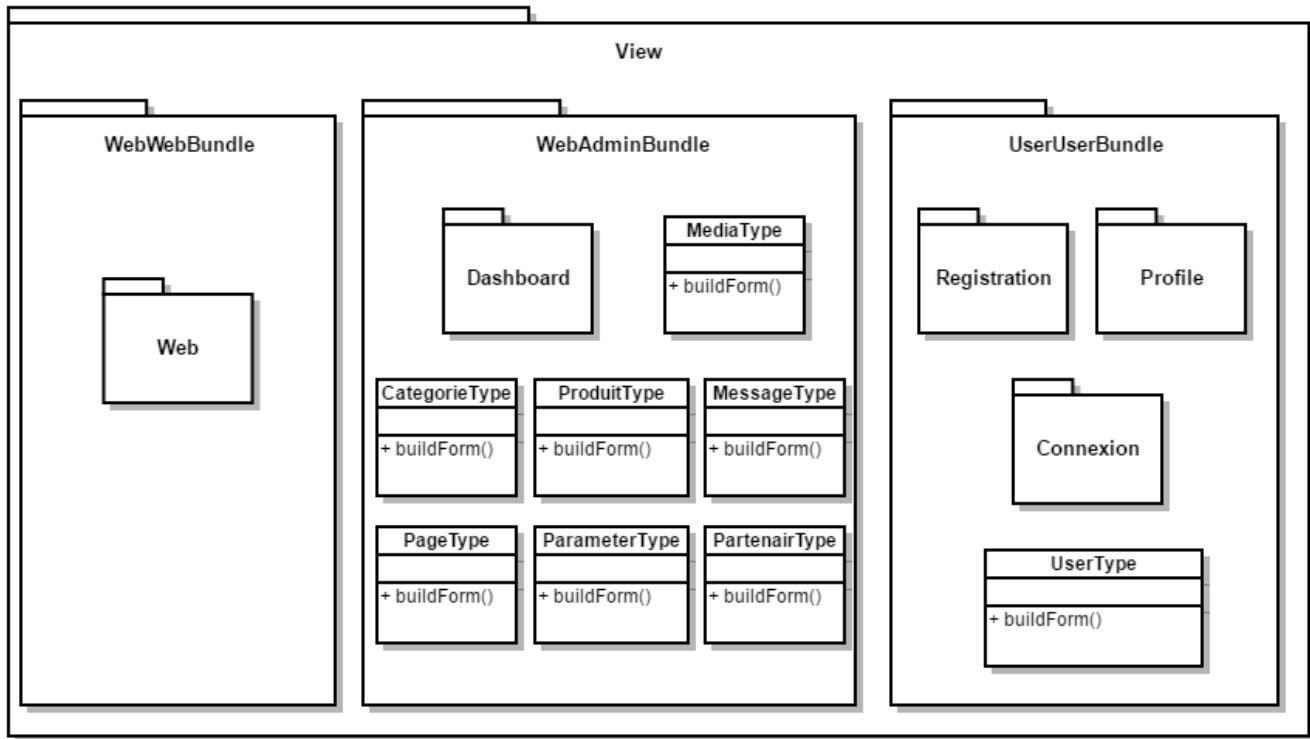


Figure III.24: Diagramme de classe « View »

## Conclusion

Durant ce chapitre, nous avons présenté la conception détaillée, à la fois, de notre base de données et de notre application Web. A ce stade, nous avons terminé une phase importante qui est la conception générale. C'est ainsi qu'on peut aborder une étape essentielle qui est la réalisation de l'application. Dans le chapitre suivant, nous allons parler en détails de cette phase.

# Chapitre IV

## Réalisation

# Introduction

La phase qui suit une conception est éventuellement l'implémentation. Après les trois chapitres précédents, nous arrivons à rassembler nos informations et réaliser notre système, nous commençons ce chapitre par décrire l'environnement de développement matériel et logiciel que nous avons adoptés. Nous présentons ensuite quelques captures d'écrans du site web.

## 1 Environnement de développement

Dans cette partie, nous présentons les différents outils matériels et logiciels nécessaires pour le développement de notre application.

### 1.1 Environnement matériel

Pour la réalisation de ce projet on a disposé d' :

- Un ordinateur :
  - Marque : Packard Bell
  - Processor : Intel(R) Core(TM) i5-3210M CPU 2.50GHz x 2
  - RAM : 4Go
  - Disque dur : 500Go
  - Système d'exploitation : Linux - Windows 10
- Un ordinateur :
  - Marque : Asus
  - Processor : Intel(R) Core(TM) i7-4500U CPU 1.80GHz x 4
  - RAM : 8Go
  - Disque dur : 1To
  - Système d'exploitation : Linux - Windows 10
- Un serveur d'hébergement chez arvixe.com<sup>5</sup>

### 1.2 Environnement logiciel

Cette section décrit l'environnement logiciel avec lequel nous avons réalisé ce projet avec le rapport.

- **IntelliJ IDEA** est un environnement de développement intégré (IDE) développé par JetBrains. IntelliJ est un IDE complet misant sur la productivité avec des systèmes d'auto-complétion intelligente, d'analyse de code en temps réel, de refactoring avancé l'intégration d'outils de tests et de debugging et une pléthore de raccourcis clavier permettant de réaliser presque n'importe quelle tâche rapidement sans jamais lever les mains du clavier pour utiliser la souris.

---

<sup>5</sup> Arvixe est une société d'hébergement Web qui offre une gamme de services allant de l'hébergement mutualisé jusqu'à l'hébergement dédié.



Figure IV.1: IntelliJ IDEA

- **Xampp** est une distribution Apache entièrement gratuite, facile à installer qui contient MariaDB, PHP et Perl. Le package open source XAMPP a été mis en place pour être incroyablement facile à installer et à utiliser.



Figure IV.2: Xampp

- **dhtmlxGantt** est un outil de la suite DHXTML (un framework JavaScript) qui permet d'intégrer des diagrammes de GANTT dans des pages web et de réaliser ainsi la gestion d'un projet.



Figure IV.3: dhtmlxGantt

- **Texmaker** est un éditeur LaTeX libre et gratuit avec support de l'unicode, correction orthographique, auto-complétion et repliage de code. Il intègre un afficheur pdf intégré avec support pour synctex et affichage en mode continu.



Figure IV.4: Texmaker

- **Balsamiq mockup** est un outil permettant de créer facilement des prototypes d'IHM électronique. Avec Balsamiq Mockups il est ainsi possible de prototyper tout type d'applications (desktop, web, smartphone, ...).



Figure IV.5: Balsamiq mockup

- **Gliffy** est une extension pour Google Chrome permet de créer des diagrammes et des organigrammes de qualité professionnelle d'une façon rapide et simple.



Figure IV.6: Gliffy

## 2 Présentation de la solution Web

Dans cette partie, nous allons exposer quelques interfaces de notre application web en essayant de présenter les fonctionnalités les plus importantes dans notre travail.

### 2.1 Interface d'accueil

L'interface d'accueil du site web donne une présentation générale sur la société Medor, elle est accessible par tous les visiteurs et elle aide les internautes à connaître les services offerts.

The screenshot shows the homepage of the Medor website. At the top, there is a navigation bar with links for Connexion / S'inscrire, Panier (0), Accueil, Présentation, Équipement Cabinet, Cabinet Réalisé, Accessoires orthopédiques, Aide A la marche, Sabot orthopédique, Demande Devis, and Contact. The main content area features a large image of a smiling female doctor in a white coat and blue shirt, with a stethoscope around her neck. Below this is a section titled "Présentation" featuring a blue medical device (possibly a defibrillator) and a text description of the company's mission. There is a "En savoir plus" button. Below this is a section titled "Cabinet" with three images of medical treatment rooms: "CABINET KINÉ", "CABINET MEDECIN", and "CABINET REALISE". At the bottom, there is a "Nos Partenaires" section with logos for AIRCASE, Case, deap, and i-TECH. The footer contains Site Map, Implantation (with a world map image), Adresse (with address details), Introduction (with company history), and a Newsletter sign-up form. Copyright information and social media links are also present.

Figure IV.7: Interface d'accueil

## 2.2 Interface catégorie

L'interface catégorie affiche les différents produits qui se trouvent au sein de la société Medor avec leurs prix unitaires et un bouton pour ajouter le produit au panier. On trouve encore les différentes catégories et sous-catégorie et un champ "chercher" pour faire une recherche.

**Equipement Cabinet**

Augmenter la valeur dans nos vies  
Design ergonomique  
Haute qualité.

Savoir Plus

**Cabinet Kiné**

- Appareil Portatif
  - Lampe Infra Rouge
  - Électrothérapie
  - Ultrason
- Appareil Professionnel
  - Pressothérapie
  - Ultrason pro
  - Electrothérapie pro
- Equipement
  - Table
  - Cage et Accessoires
  - Gueridon
  - Tabouret
  - Cauquin

**Equipement Cabinet**

Produit	Prix	Action
Infrared WHF-312 Lampe Infrar...	55.000 DT	Ajouter Au Panier
SMART TENS	40.000 DT	Ajouter Au Panier
MIO-CARE TENS	42.000 DT	Ajouter Au Panier
MIO-CARE BEAUTY	40.000 DT	Ajouter Au Panier
MIO-CARE FITNESS	35.000 DT	Ajouter Au Panier
MIO VEIN	35.000 DT	Ajouter Au Panier

Figure IV.8: Interface catégorie

## 2.3 Interface contact

L'interface contact présente les informations de contact de la société Medor, on trouve l'adresse pour chaque locale, les emails et les numéros de téléphone...

On peut aussi envoyer un message direct à trouve un formulaire.

Address : **Route De Kairouan Monastir Immeuble Ruspina**  
**Monastir 5000-TUNISIE,**  
**Tél.Fax : (+216) 73 449 231 GSM : 99 000 631 - 98 742 201**  
**medor-compta@hotmail.com**

Formulaire

<input type="text" value="Nom"/>	<input type="text" value="Email"/>	<input type="text" value="Téléphone"/>
<input type="text" value="Objet"/>		

Figure IV.9: Interface contact

## 2.4 Interface produit détaillé

Lorsque le client veut afficher le détails d'un produit il clique sur l'image ou le nom du produit pour transmettre à cet page qui lui affiche tous les informations (description, prix...).

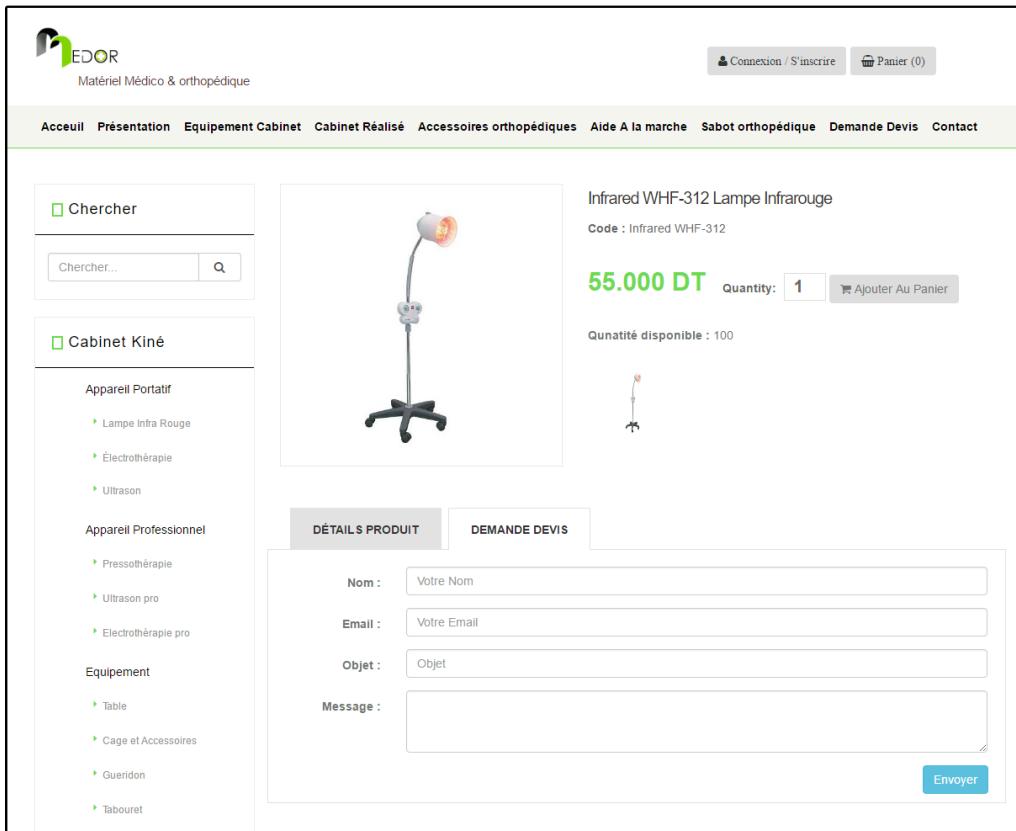


Figure IV.10: Interface produit détaillé

## 2.5 Interface commande

À travers cette interface et après l'authentification le client peut valider la commande, il peut vérifier ses informations personnelles en remplaçant les champs indiqués, et choisir la méthode de paiement.

The screenshot shows the EDOR website's command interface. At the top, there is a header with the logo 'EDOR' (Matériel Médico & orthopédique), user profile links ('Profile', 'Déconnexion'), and a 'Panier (2)' button. Below the header, a navigation menu includes 'Accueil', 'Présentation', 'Equipement Cabinet', 'Cabinet Réalisé', 'Accessoires orthopédiques', 'Aide A la marche', 'Sabot orthopédique', 'Demande Devis', and 'Contact'. The main content area starts with a section titled 'Produits :'. It displays a table of items in the shopping cart:

Nom	Prix	Quantity	Total
Canne Canadienne Code: C1-001	20.000 DT	1	20.000 DT
Canne 6 Code: C1-006	30.000 DT	1	30.000 DT
			<b>PRIX HT</b> 50.000 DT
			<b>TVA</b> 6.000
			<b>TOTAL TTC</b> 56.000 DT

Below the products table, there are two sections: 'Information Personel' and 'Information Commande'. The 'Information Personel' section contains fields for 'Nom', 'Prenom', 'Email', 'Téléphone', and 'Adresse'. The 'Information Commande' section contains fields for 'Date' (2016-05-16), 'Nombre des produit' (2), and 'TOTAL TTC' (56.000 DT). Further down, there is a section titled 'Modes De Paiement Disponibles :' with three radio buttons: 'Paiement par virement', 'Paiement par Mandat', and 'Paiement main à la main'. A blue button labeled 'Confirmer commande' is located next to the payment method section. A light blue box at the bottom left is titled 'Paiement main à la main' and contains the text: 'Merci d'imprimer le reçu d'achat dans la page suivante et nous visiter dans notre local (Adresse) avec le montant suivante : 56 DT.'

Figure IV.11: Interface commande

## 2.6 Interface backend

L'interface ci-dessous présenter la partie back office du site Web, dans l'accueil on trouve quelques statistiques et un menu qui permet la gestion administrative du site.

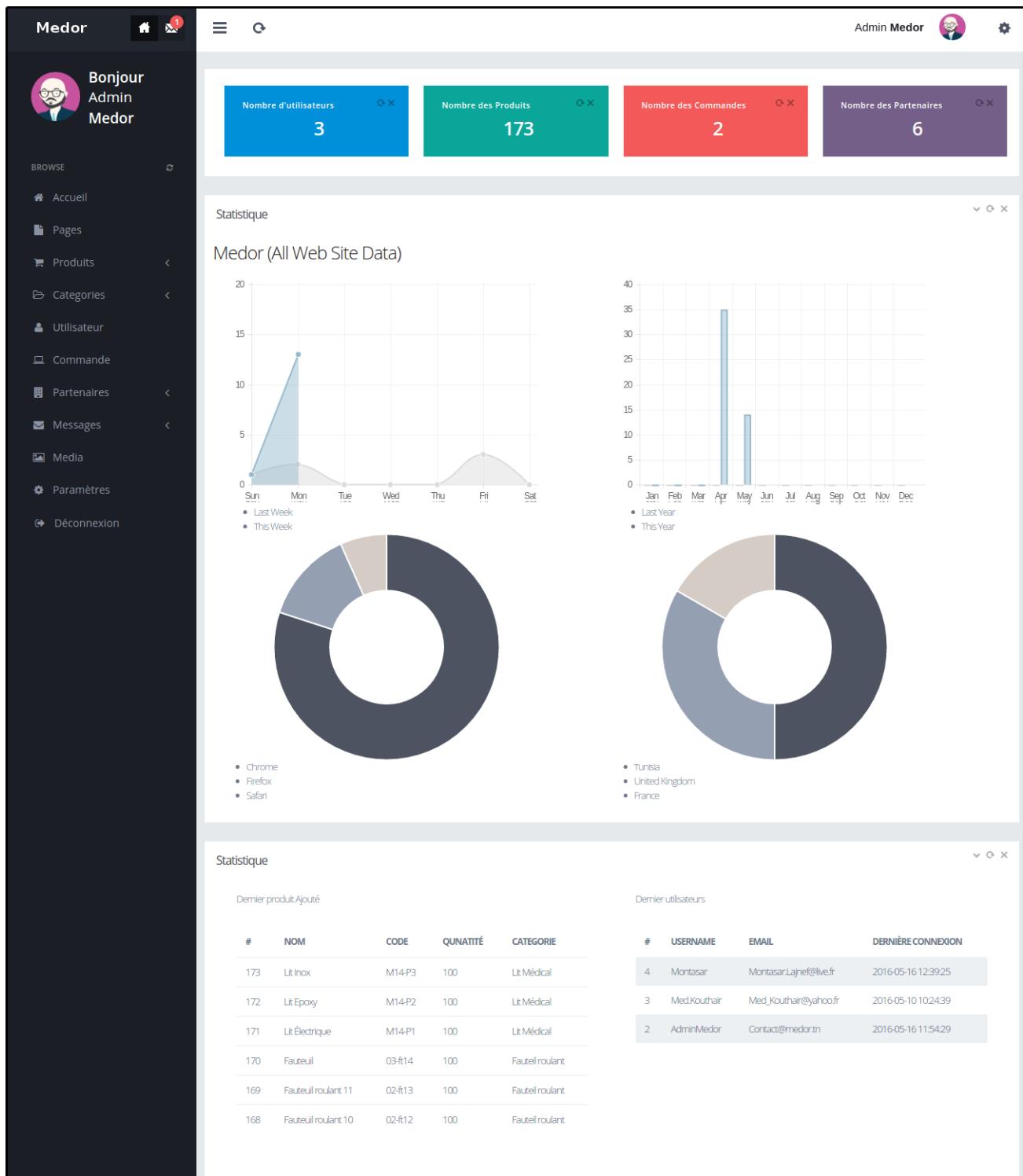


Figure IV.12: Interface Backend

## 2.7 Interfaces gestion des produits

L'interface gestion du produit permet de gérer les produits stockés dans le site. L'administrateur ou le responsable commercial peut ajouter un nouveau produit, il peut aussi le modifier ou le supprimer...

ID	NOM	CODE	PRIX	QUANTITÉ	CATÉGORIE	ACTIONS
1	Infrared WHF-312 Lampe Infrarouge	Infrared WHF-312	55.000 DT	100	Lampe Infra Rouge	<button>Afficher</button> <button>Modifier</button>
2	SMART TENS	SK124-00	40.000 DT	100	Électrothérapie	<button>Afficher</button> <button>Modifier</button>
3	MIO-CARE TENS	SK211-00	42.000 DT	100	Électrothérapie	<button>Afficher</button> <button>Modifier</button>
4	MIO-CARE BEAUTY	SK229-00	40.000 DT	100	Électrothérapie	<button>Afficher</button> <button>Modifier</button>

Figure IV.13: Interface liste des produits

Ajouter produit

Nom de Produit:

Code Produit:

Description:

Rich Text Editor toolbar

Categorie:

Medor

Quantité disponible

1

Prix:

0

Images:

[Ajouter une image](#)

[Retour à la liste](#) [Ajouter Produit](#)

Figure IV.14: Interface ajouter produit

## 2.8 Interfaces gestion des utilisateurs

L'interface gestion des utilisateurs permet de gérer les droits d'accès des différents utilisateurs : ajouter un responsable commercial, activer un compte utilisateur, désactiver un compte...

ID	USERNAME	EMAIL	DERNIÈRE CONNEXION	ROLE	ACTIONS
2	AdminMedor	Contact@medor.tn	2016-05-16 11:54:29	Admin	<button>Modifier</button>
3	MedKouthair	Med_Kouthair@yahoo.fr	2016-05-10 10:24:39	Utilisateur	<button>Modifier</button>
4	Montasar	Montasar.Lajnef@live.fr	2016-05-16 12:39:25	Utilisateur	<button>Modifier</button>
5	alaa	taous.alaa@gmail.com	2016-05-17 05:47:19	Utilisateur	<button>Modifier</button>

Figure IV.15: Interface liste des utilisateurs

Figure IV.16: Interface modifier utilisateur

## 2.9 Interfaces de la liste des commandes

L'interface liste des commandes permet aux administrateurs de gérer les commandes il peut confirmer une commande ou l'annuler, il peut voir la liste des produits commandés, la date de la commande...

Figure IV.17: Interface liste des commandes

		PRIX	TOTALE
QTE	NOM PRODUIT		
3	Canne Canadienne	20.000	60.000
1	Canne Tripode	45.000	45.000
		PRIX HT Totale	105.000
		TVA	12.600
		<b>TOTAL TTC</b>	<b>117.600</b>

Figure IV.18: Interface afficher commande

## 2.10 Interfaces de la liste des messages

À travers l'interface liste des messages l'administrateur peut examiner tous les messages et demande devis envoyées par les internautes et il peut aussi répandre aux messages à tout moment.

ID	ETAT	TYPE	OBJET	ACTIONS
3	Répondu.	Demande Devis d'un produit	Quantité disponible	<button>Afficher</button>
4	Vu.	DemandeDevis	Aide à la marche	<button>Afficher</button>
5	N'est pas vu.	Contact	Adresse Medor	<button>Afficher</button>

Figure IV.19: Interface liste des messages

Figure IV.20: Interface afficher message

## Conclusion

Dans ce chapitre, nous avons présenté l'environnement matériel et logiciel utilisé lors du développement du site web. Ensuite, on a présenté quelques captures d'écran montrant le bon fonctionnement de notre application.

# Chapitre V

## Extension

# Introduction

Le présent chapitre porte sur l'extension de notre application web en une application mobile.

## 1 Environnement de développement

Nous présentons ici les différents outils matériels et logiciels nécessaires pour le développement de notre application partie mobile.

### 1.1 Environnement matériel

Pour réaliser l'application mobile on a utilisé :

- Les deux ordinateurs mentionnés précédemment dans la partie web.
- Le serveur d'hébergement chez arvixe.com<sup>5</sup> mentionné précédemment.
- Un smartphone Samsung Galaxy i8262 et Evertek EverMiracle.

### 1.2 Environnement logiciel

Pour le côté logiciels, on a utilisé :

- **Android Studio** est un environnement de développement pour développer des applications Android. Il est basé sur IntelliJ IDEA.



Figure V.1: Android Studio

- **GenyMotion** est un émulateur android plus rapide et complet pour ordinateur. Il fonctionne avec VirtualBox en y créant une machine virtuelle.



Figure V.2: GenyMotion

- **Advanced REST client** est une extension pour google chrome permettre de tester une API Web.



Figure V.3: Advanced REST client

## 2 Présentation de la solution mobile

Dans cette partie de notre rapport, nous allons présenter quelques interfaces de l' application mobile résumant les différentes fonctionnalités offertes.

### 2.1 Interface d'accueil

L'interface d'accueil donne une idée générale sur société Medor. On trouve les derniers produits ajouter, les partenaires...



Figure V.4: Interface d'accueil application mobile

### 2.2 Interface menu

Cette interface expose les différents volets aux quels l'utilisateur peut accéder.

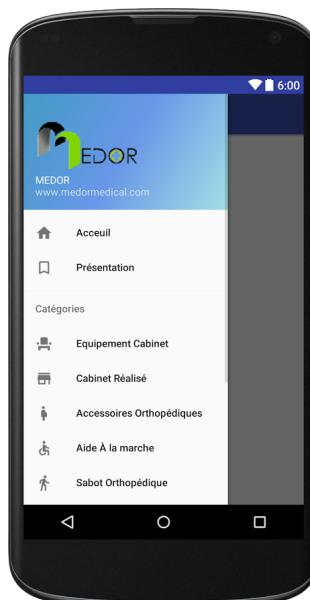


Figure V.5: Interface menu application mobile

## 2.3 Interface catégorie

L'interface catégorie affiche les différents produits avec leurs prix unitaires, on trouve aussi les différentes catégories et sous-catégorie.

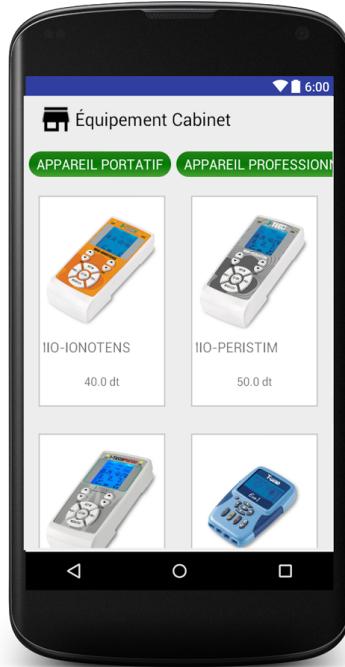


Figure V.6: Interface catégorie application mobile

## 2.4 Interface produit

L'interface Produit présente les détails d'un seul produit (description, prix, code...).



Figure V.7: Interface produit application mobile

## 2.5 Interface demande devis

L'interface demande devis permet aux utilisateurs de contacter la société Medor.

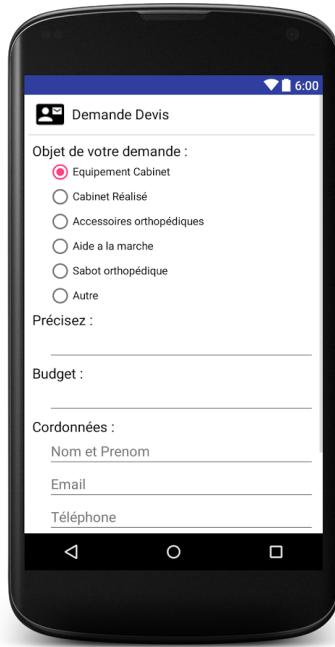


Figure V.8: Interface demande devis application mobile

## 2.6 Interface contact

Cette interface fournit aux utilisateurs les différentes coordonnées qui lui permettent de contacter la société.

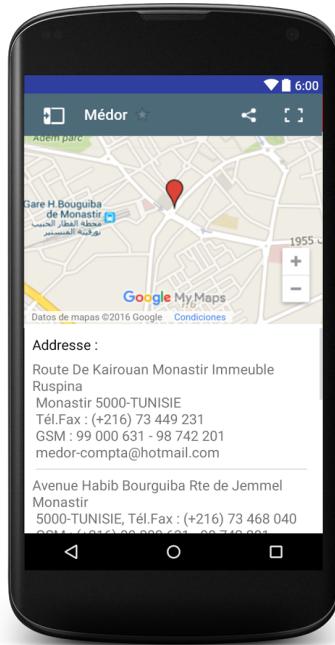


Figure V.9: Interface contact application mobile

## Conclusion

Afin de bien conclure ce dernier chapitre, nous avons présenté les différents éléments composant la réalisation de l'application mobile. En effet on a présenté les différentes interfaces ainsi que l'environnement matériel et logiciel favorable au bon fonctionnement de l'application mobile.

# Conclusion générale

Le présent rapport est réalisé dans le cadre de notre projet de fin d'études en vue de l'obtention du diplôme de la Licence Appliquée Sciences Informatiques.

Ce travail a pour objectif de concevoir et développer une application Web et mobile qui permet de consulter les produits et les services offerts par la société Medor.

Pour pouvoir compléter notre mission, nous avons détaillé les différentes étapes d'analyse, de conception et de réalisation de ce système.

Le premier chapitre a été consacré au cadre général du projet. Il a commencé par une phase de recherche et de documentation, traçant ainsi les repères de travail à effectuer.

Le deuxième chapitre a été consacré à l'étude de l'existant et l'étude conceptuelle où nous avons dégagé les insuffisances des applications préexistantes et nous avons mis en évidence la mise en place de l'architecture générale de notre système.

Une fois nos objectifs ont été fixés nous avons enchainé dans le troisième chapitre une conception afin de mener à bien notre projet. nous avons utilisé UML comme un langage de modélisation et 2TUP comme un processus de développement.

Le quatrième chapitre a été dédié à l'aspect implémentation qui contient le choix de l'environnement de travail ainsi qu'une copie de quelques interfaces réalisés de la partie web.

On termine par le cinquième chapitre qu'il a été consacré pour la réalisation de la partie Mobile.

Ce projet était bénéfique pour nous dans plusieurs sens. Il nous ouvrir sur le monde professionnel qui implique de la responsabilité et de la discipline ainsi que du travail collaboratif d'équipe, il a été une occasion pour améliorer nos connaissances dans les environnements de travail et de maîtriser les langages de programmation pour qui seront certes utiles au niveau professionnel.

Enfin, nous espérons que nous avons réussi à réaliser un site Web et une application mobile fonctionnels qui satisfont les besoins des futurs utilisateurs et qui répondent à leurs attentes.

## Annexes

# Annexe A

## Hébergement site web

Après avoir créer notre site, il faudra le rendre accessible. Pour cela, au sein de la société Media Plus nous avons réservé un nom de domaine **Medor.tn** chez OVH et un hébergement web mutualisé chez arvixe.



Figure A.1: Logo arvixe

la figure ci-dessous présente le panneau de contrôle (cPanel) de arvixe dans lequel nous pouvons créer une base de données, gérer les comptes FTP, ...

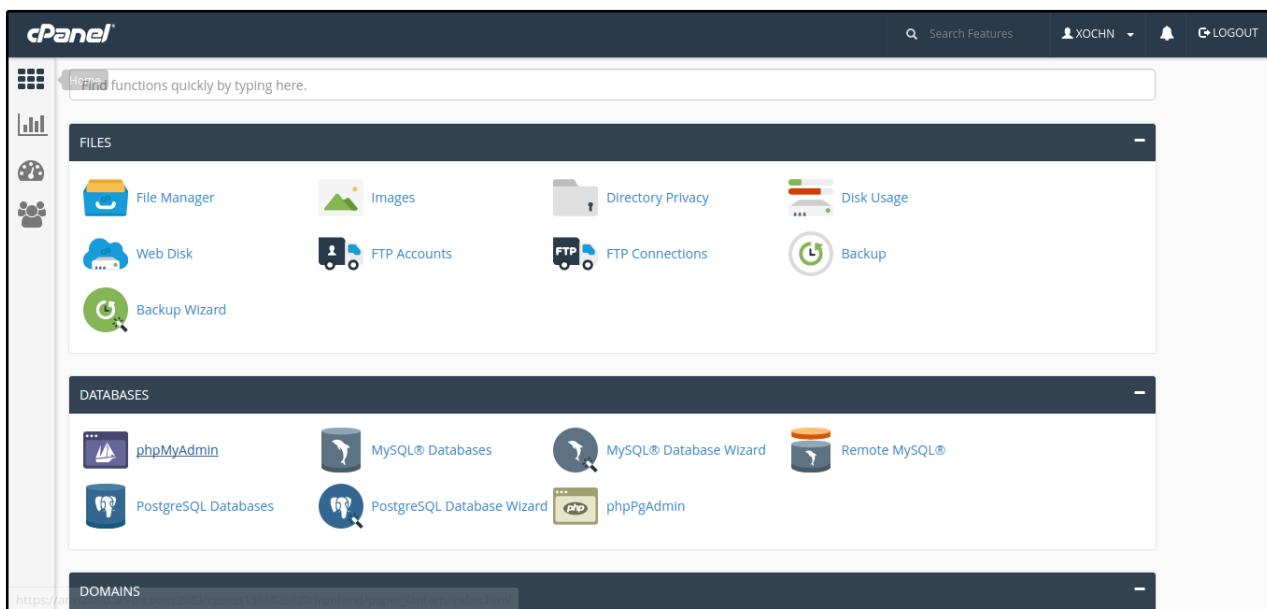


Figure A.2: Arvixe Cpanel

Nous commençons d'abord par pointer le domaine Medor.tn vers l'hébergement arvix, pour cela il faut entrer dans le panneau de contrôle d'OVH et change le DNS du domaine comme indiqué dans la figure ci-dessus :

The screenshot shows the DNS management section of a web hosting control panel. At the top, it displays the domain name "medor.tn" and its expiration date "Expires on Mar 18, 2017". There are buttons for "Renew", "Order an allDom pack", "Create my website", and "Add a sub-domain". Below these are tabs for "General information", "DNS zone", "DNS management" (which is selected), "Redirection", "DynHost", "Glue", "DS Records", "Recent tasks", and "More +". A warning message in orange says: "Pour vous assurer du bon fonctionnement de votre configuration, vous pouvez utiliser cet [outil de vérification DNS](#). ➔". The main area contains a table titled "DNS server" with columns for "DNS server", "IP associated", "Status", and actions. Two entries are listed:

DNS server	IP associated	Status	
ns2.armadillo.arvixe.com	-	Active	<a href="#">Edit</a> <a href="#">Delete</a>
ns1.armadillo.arvixe.com	-	Active	<a href="#">Edit</a> <a href="#">Delete</a>

Figure A.3: Gestion DNS du domaine medor.tn

On utilise après FileZilla, un client FTP qui permet d'envoyer des fichiers vers un serveur distant pour mettre en ligne notre site web.

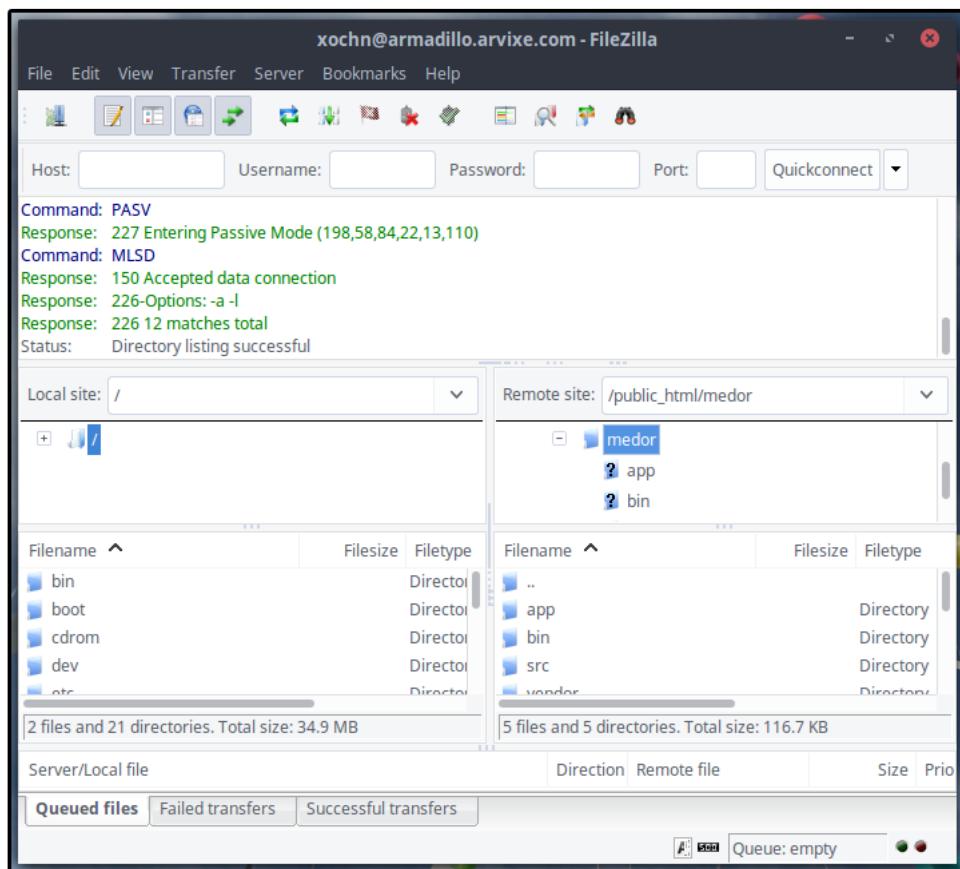


Figure A.4: FileZilla

Une fois le transfert des fichiers se termine, On créer une base de données et un utilisateur base de donnée puis on lui donne les priviléges comme indiqué sur la figure ci-dessus :

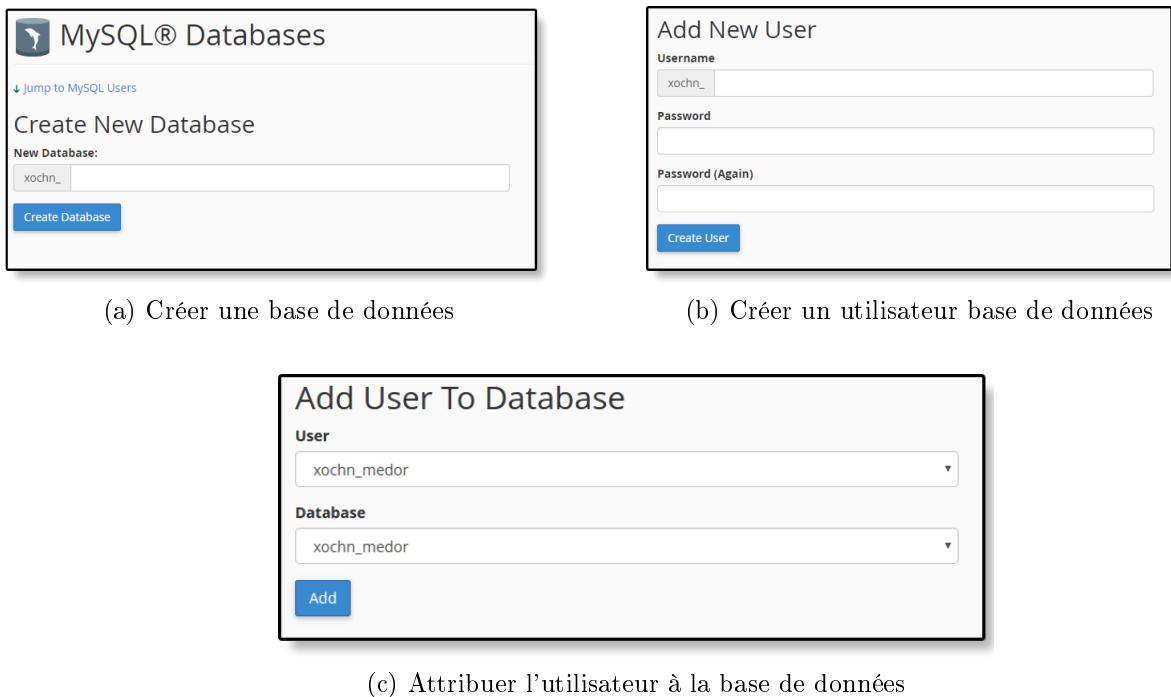


Figure A.5: Configuration base de données

Après la création de la base de données et de l'utilisateur, on se connecte sur phpmyadmin dans le serveur et on importe la base de données que nous avons l'exporter depuis phpmyadmin dans le serveur local.

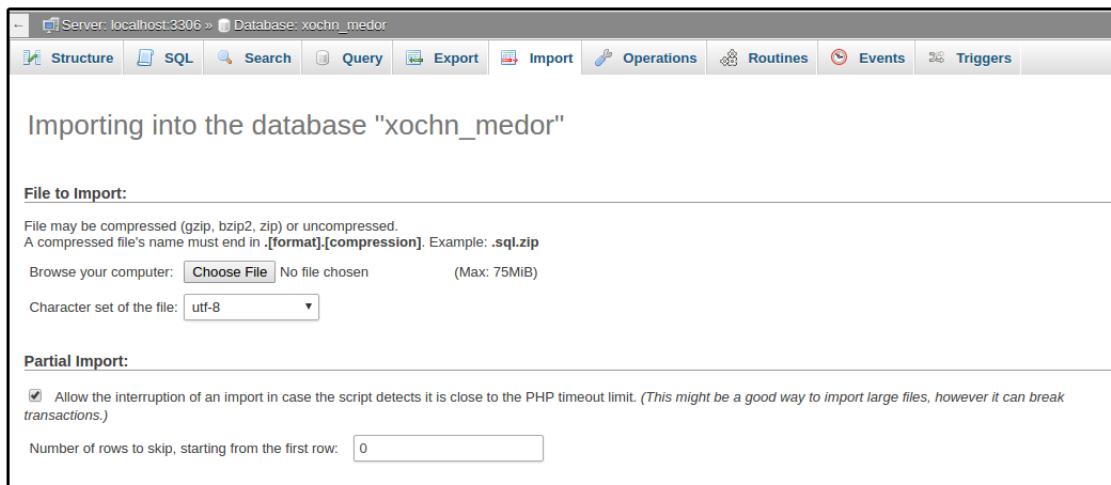


Figure A.6: Phpmyadmin

La dernière étape consiste à modifier les paramètres de connexion à la base de données dans Symfony et faire quelques modifications sur le fichier .htaccess afin que notre site devient accessible en ligne.

# Annexe B

## Service web

Afin de réaliser l'application mobile, nous avons utilisé les services web pour récupérer les données depuis le serveur.

- **FOSRestBundle** pour la restitution de données.
  - **JMSSerializerBundle** pour la sérialisation.

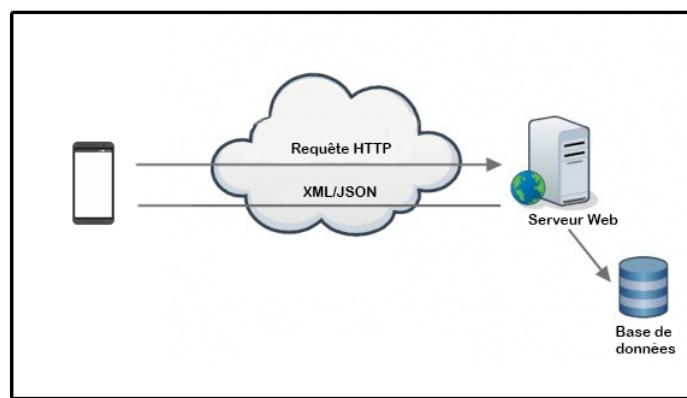


Figure B.1: Service web

Après avoir installé et configuré les bundles, nous avons créé des contrôleur qui permet de retourner les données en format JSON.

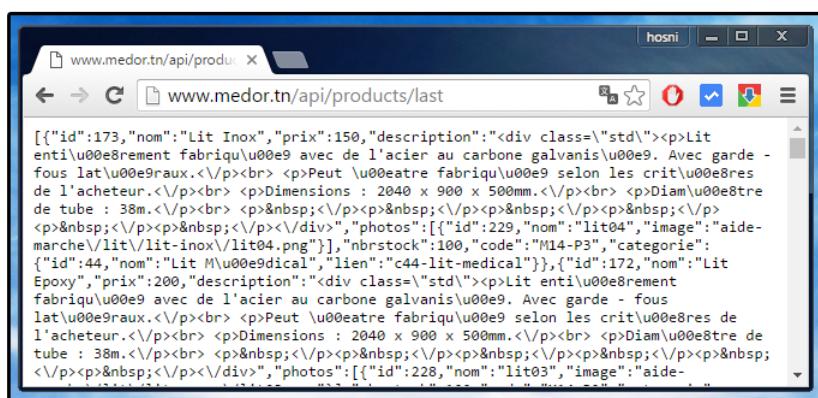


Figure B.2: JSON

Pour tester ces services web, nous avons utilisé "Advanced REST client" une extension pour google chrome.

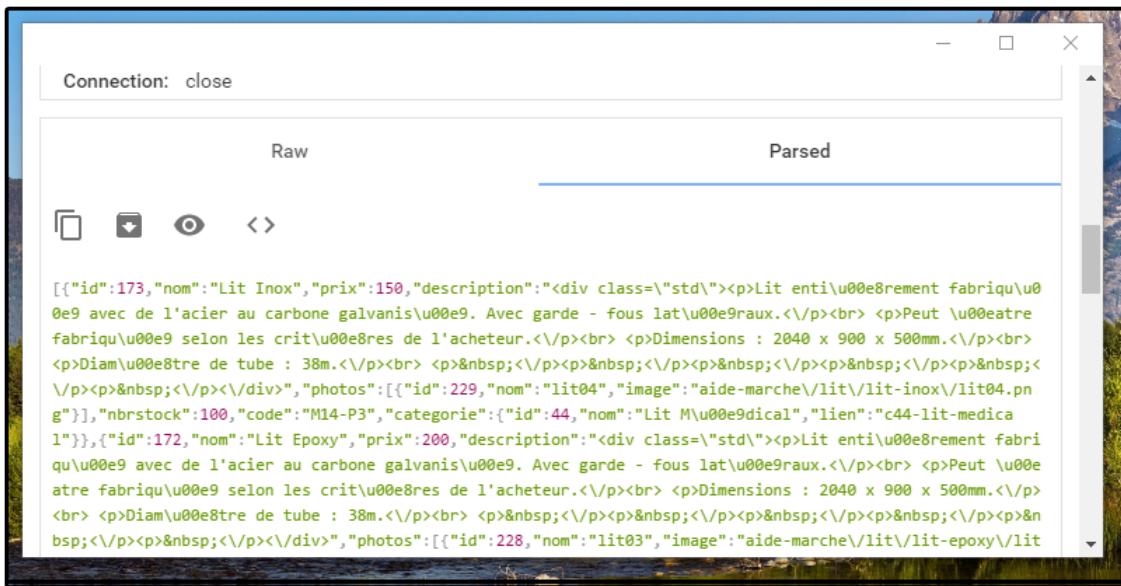


Figure B.3: Advanced REST client

Puis on a fait l'implémentation des services web dans l'application mobile.

```
try {
    JSONObject jsonObject = (JSONObject) jsa.get(i);
    Produit p = new Produit();
    p.setId(jsonObject.optInt("id"));
    p.setNom(jsonObject.optString("nom").toString());
    p.setDescription(jsonObject.optString("description").toString());
    p.setPrix(jsonObject.optDouble("prix"));
    p.setCode(jsonObject.optString("code").toString());
    Categorie c = new Categorie();
    c.setId(jsonObject.optJSONObject("categorie").optInt("id"));
    c.setNom(jsonObject.optJSONObject("categorie").optString("nom").toString());
    c.setLien(jsonObject.optJSONObject("categorie").optString("lien").toString());
    p.setCategorie(c);
    try {
        JSONArray photos = jsonObject.optJSONArray("photos");
        for (int j = 0; j < photos.length(); j++) {
            String urlImage = photos.getJSONObject(j).optString("image");
            p.getImagesURL().add(urlImage);
        }
    } catch (Exception e) {
    }
    produits.add(p);
} catch (Exception e) {
}
```

Figure B.4: Implémentation Service web

## Annexe C

# Soumettre l'application sur Google Play

Google Play est la boutique d'application la plus populaire dans le monde, pour cela il est l'endroit convenable pour soumettre notre application.



Figure C.1: Logo Google play

Il faut d'abord ouvrir un compte développeur, Une fois le compte est crée et activé on peut publier notre application.

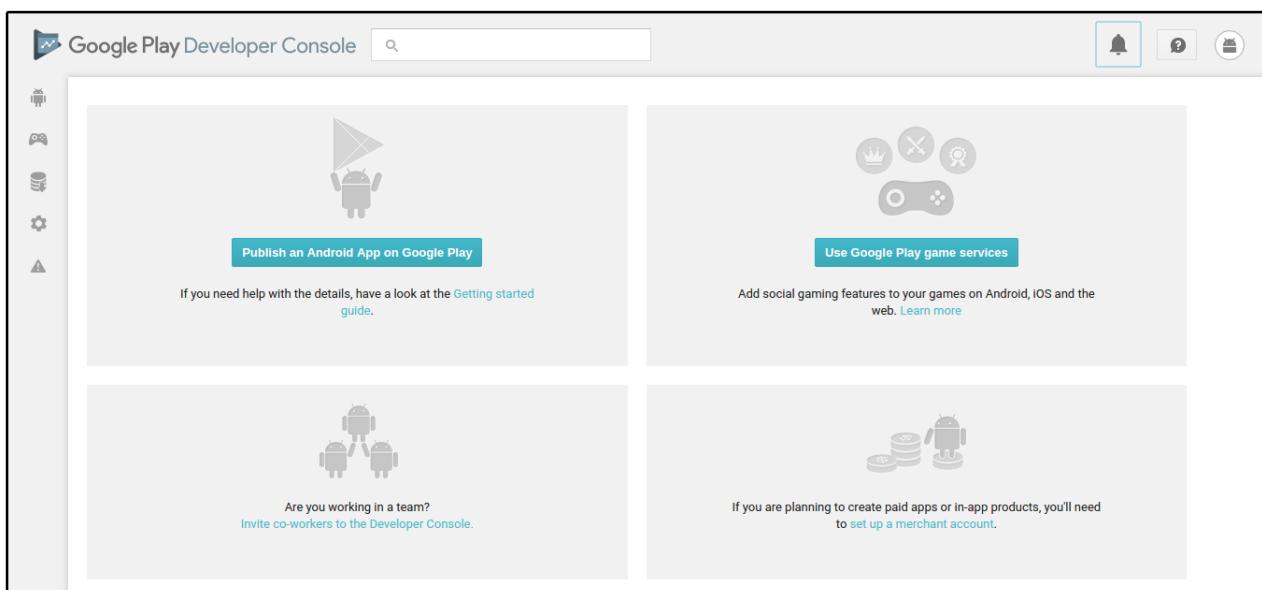


Figure C.2: Dashboard Google play

Afin de partager notre application Android sur Google Play, on importe le fichier APK et on remplit les informations ( le titre, l'icone, quelques images des interfaces, ...) et enfin on soumettre.

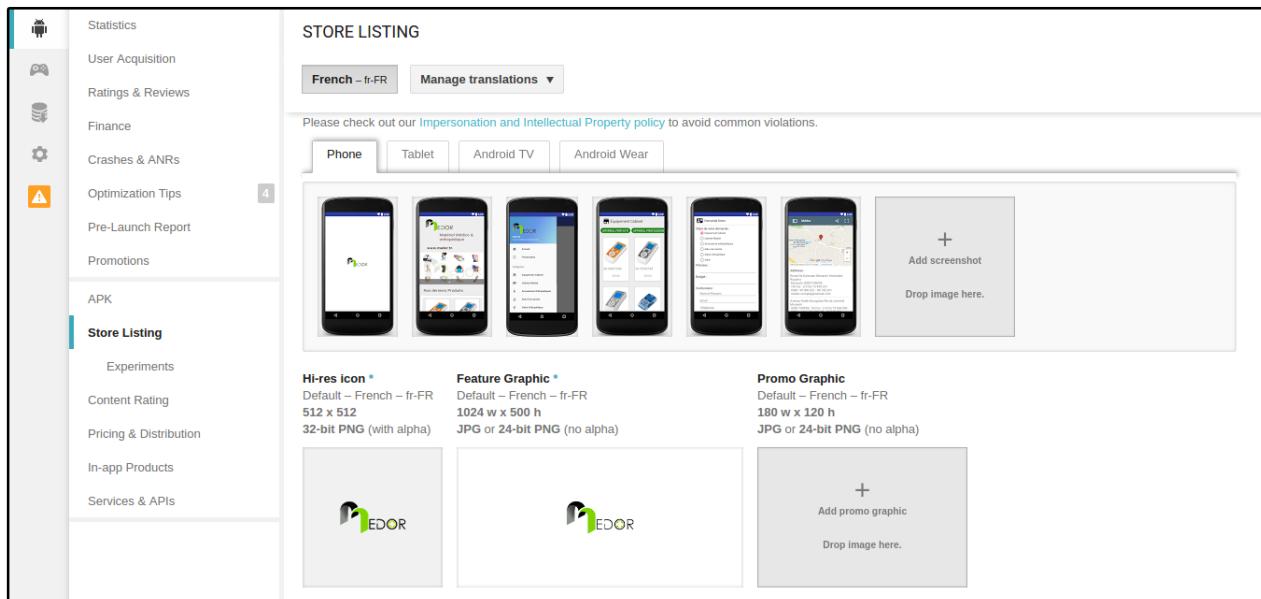


Figure C.3: Soumettre l'application sur Google play

Après quelques heures si google accepte notre application, elle sera disponible en ligne et tout le monde peut le télécharger.

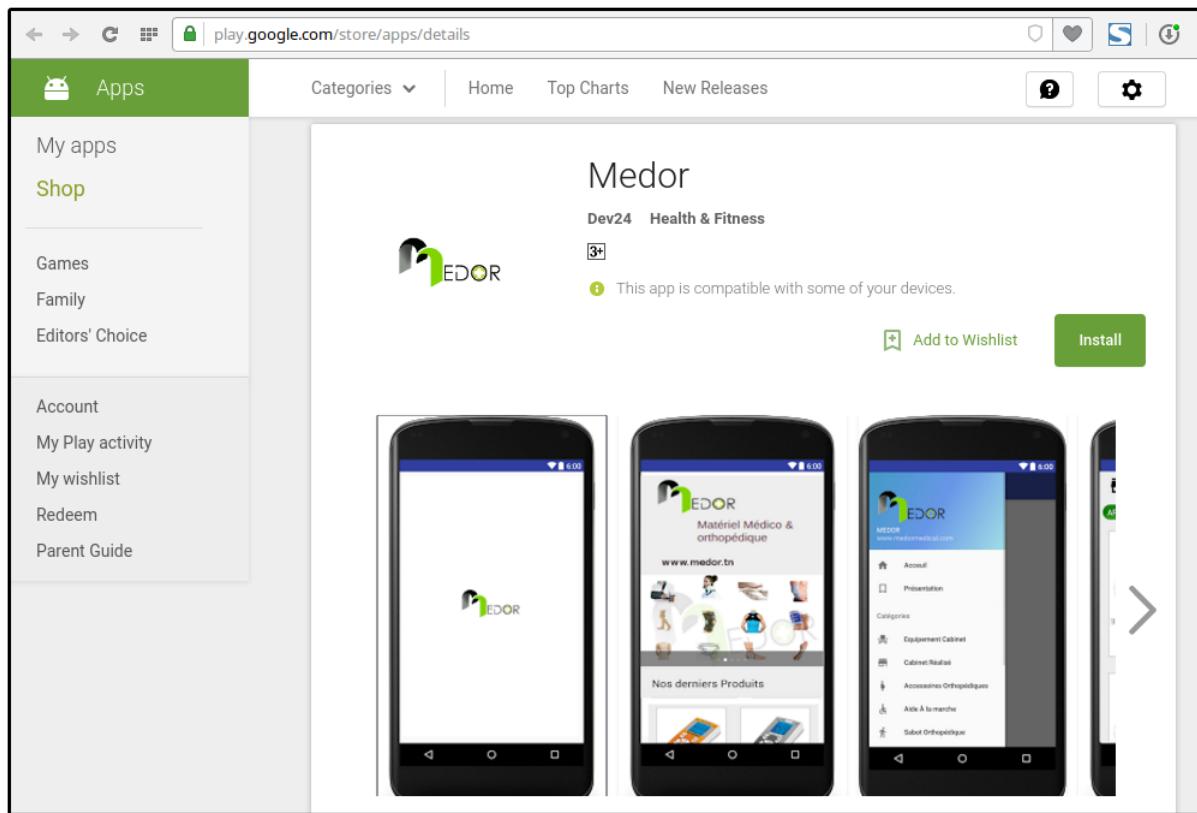


Figure C.4: Medor sur Google play