# DEDICACES

Du plus profond de mon cœur, je dédie ce travail

A tous ceux, qui me sont chers….

A tous ceux, qui m’ont fait confiance….

À tous ceux, qui m’ont motivé….

À tous ceux, que j’aime…

À tous ceux, qui m’aiment….

À l’âme de mon père Mohamed….

À ma chère mère Hayet …….

A la joie de ma vie, mes frères et ma grande famille…

À mes amis, salah et loay…….

« La foi, la joie,

La tristesse, la détresse,

Marquer sa présence, sans une vraie apparence,

Être parmi ceux, qui vivent encore au creux,

Des âmes, des trames,

À travers ces quelques lignes,

Ma connaissance s’instigue,

À travers ces quelques mots,

Mon expérience ressort »

Kais Trabelsi

# Remerciements

Je remercie ALLAH de m’avoir accordé la connaissance de la science et de m’avoir aidé à réaliser ce travail. Je tiens à remercier chaleureusement et respectivement tous ceux qu’ont contribués de près ou de loin à la réalisation de ce projet de fin d’études, à savoir mes encadreurs MR. Hassan hachicha et à MR.Rachid souissi qui m’ont orienté et apporté leur aide précieuse à l’iset et au sein de NEJMA ou j’ai pu développer mon travail. Mes remerciements s’adressent aussi à MR. Ameur elleuch, le gérant de la société NEJMA. Je tiens à remercier les membres de Jury MR. Ahmed jmal et MR Nissen Masmoudi qui ont accepté d’évaluer mon travail. Je tiens à remercier également mes chers amis en souvenir de nos éclats de rire et des bons moments en souvenir de tout ce qu’on a vécu ensemble. J’espère de tout mon cœur que notre amitié durera éternellement. Ce travail fut difficile mais bénéfique de tous les points de vue. Merci

Sommaire

[DEDICACES 1](#_Toc483671933)

[Remerciements 2](file:///C:\Users\kais\Desktop\rapport2017.docx#_Toc483671934)

[Introduction Générale 5](#_Toc483671935)

[CHAPITRE 1 : ETAT DE L’ART 6](file:///C:\Users\kais\Desktop\rapport2017.docx#_Toc483671936)

[1. Introduction 7](#_Toc483671937)

[2. Recueil 7](#_Toc483671938)

[4. Définition du commerce électronique 8](#_Toc483671939)

[4. E**tude comparative des outils du marché** 10](#_Toc483671939)

[4. Conclusion 11](#_Toc483671939)

[CHAPITRE 2 : Etude Préalable 10](file:///C:\Users\kais\Desktop\rapport2017.docx#_Toc483671941)

[1. Introduction 11](#_Toc483671942)

[2 Définition de la mission 11](#_Toc483671943)

[2.1 Présentation de l’entreprise 11](#_Toc483671944)

[2.2 Définition du champ d’étude 12](#_Toc483671945)

[3. Etude de l’existant 13](#_Toc483671946)

[4. Analyse de l’existant 13](#_Toc483671946)

[5. Critique de l’existant 17](#_Toc483671946)

[6. Solution proposée 18](#_Toc483671949)

[7. Planning prévisionnel 18](#_Toc483671950)

[8. Conclusion 19](#_Toc483671951)

[CHAPITRE 3 : Analyse & Spécification des besoins 20](file:///C:\Users\kais\Desktop\rapport2017.docx#_Toc483671952)

[1. Introduction 21](#_Toc483671953)

[2. Spécification des besoins 21](#_Toc483671954)

[2.1 . Les besoins fonctionnels 21](#_Toc483671955)

[2.2 . Les besoins non fonctionnels 22](#_Toc483671956)

[3. Diagrammes des cas d’utilisation 23](#_Toc483671957)

[4. Conclusion 36](#_Toc483671966)

[CHAPITRE 4 : Conception 37](file:///C:\Users\kais\Desktop\rapport2017.docx#_Toc483671967)

[1. Introduction 38](#_Toc483671968)

[2. Environnement de modélisation UML 38](#_Toc483671969)

[3. Conception générale 38](#_Toc483671970)

[4. Conception détaillée 42](#_Toc483671976)

[5. Service web 38](#_Toc483671968)

[6. partie mobile 49](#_Toc483671981)

[7. Diagramme de séquences 50](#_Toc483671982)

[8. Diagramme d’activités. 61](#_Toc483671984)

[9. Diagramme d’états-transitions 66](#_Toc483671986)

[CHAPITRE 5 : Réalisation 67](file:///C:\Users\kais\Desktop\rapport2017.docx#_Toc483671988)

[1. Introduction 68](#_Toc483671989)

[2. Environnement de réalisation 68](#_Toc483671990)

[3. Schéma logique des données 71](#_Toc483671996)

[4. Les modèles navigationnels 73](#_Toc483671997)

[5. Les interfaces de l’application 74](#_Toc483671998)

[6. Apports 94](#_Toc483672012)

[a. Au niveau des connaissances techniques 94](#_Toc483672027)

[b. Au niveau de la conception et du développement 94](#_Toc483672028)

[7. Conclusion 94](#_Toc483672029)

[Conclusion générale et Perspectives 95](#_Toc483672030)

[NETOGRAPHIE 96](#_Toc483672031)

[Annexe 96](#_Toc483672032)

[1. Java Enterprise Edition(J2EE) 96](#_Toc483672033)

[2. JSP et Servlet 98](#_Toc483672034)

[3. Entreprise Java Beans(EJB) 99](#_Toc483672035)

[4. Android 103](#_Toc483672036)

[5. Services Web 104](#_Toc483672037)

[6. JSON 105](#_Toc483672038)

[4. Jersey Gson 105](#_Toc483672039)

**Liste des figures**

[**Figure 1: Logo de Nejma** 14](#_Toc483675379)

[**Figure2: Organigramme de la société** 15](#_Toc483675380)

[**Tableau 2 : tableau des solutions existantes** 17](#_Toc483675381)

[**Tableau 4 :Planning de réalisation du projet** 22](#_Toc483675382)

[**Figure 5 : diagramme de cas d’utilisation  général** 29](#_Toc483675383)

[**Figure 6 : diagramme de cas d'utilisation relatif à « l’administrateur »** 30](file:///C:\Users\kais\Desktop\rapport2017.docx#_Toc483675384)

[**Figure 7 : diagramme de cas d'utilisation relatif au « client »** 31](file:///C:\Users\kais\Desktop\rapport2017.docx#_Toc483675385)

[**Figure 8 : diagramme de cas d'utilisation relatif au « visiteur »** 32](file:///C:\Users\kais\Desktop\rapport2017.docx#_Toc483675386)

[**Figure 9 : diagramme de cas d'utilisation partie mobile relatif au « client »** 32](file:///C:\Users\kais\Desktop\rapport2017.docx#_Toc483675387)

[**Figure 10 : diagramme de cas d'utilisation relatif au « Secrétaire»** 33](file:///C:\Users\kais\Desktop\rapport2017.docx#_Toc483675388)

[**Tableau 4:Description textuelle du cas d’utilisation« authentification »** 34](#_Toc483675389)

[**Tableau 5 :Description textuelle du cas d’utilisation « gérer les Catégories»** 34](#_Toc483675390)

[**Tableau 6 :Description textuelle du cas d’utilisation « gérer les produits»** 35](#_Toc483675391)

[**Tableau 7 :Description textuelle du cas d’utilisation « gérer les messages»** 35](#_Toc483675392)

[**Tableau 8 :Description textuelle du cas d’utilisation « consulté la liste des produits »** 36](#_Toc483675393)

[**Tableau 9 :Description textuelle du cas d’utilisation « Ajouter produit au panier »** 37](#_Toc483675394)

[**Tableau 10 :Description textuelle du cas d’utilisation « Ajouter produit au liste favori »** 37](#_Toc483675395)

[**Tableau 11 :Description textuelle du cas d’utilisation « passer commande »** 38](#_Toc483675396)

[**Tableau 13: Description textuelle du cas d’utilisation « gérer les commandes »** 38](#_Toc483675397)

[**Figure 11 : architecture 4-tiers** 42](#_Toc483675399)

[**Figure 12: Diagramme de déploiement** 43](#_Toc483675400)

[**Figure 13 : Le modèle MVC** 44](#_Toc483675401)

[**Figure 14 : les couches de l'application** 44](file:///C:\Users\kais\Desktop\rapport2017.docx#_Toc483675402)

[**Tableau 14 :Dictionnaire des données** 47](#_Toc483675403)

[**Tableau 15 :Représentation de la relation classe d'association** 47](#_Toc483675404)

[Figure 15 : diagramme de classes 48](#_Toc483675405)

[**Figure 16 : diagramme de composants partie 1** 49](file:///C:\Users\kais\Desktop\rapport2017.docx#_Toc483675406)

[**Tableau 17 : description des classes** 53](#_Toc483675409)

[**Tableau 18 : stéréo types de Jacobson pour un diagramme de séquences** 53](#_Toc483675410)

[**Figure 18 : Diagramme de Séquences « Consulter la liste des produit»** 54](#_Toc483675411)

[**Figure 20 : Diagramme de séquences « Inscription   »** 54](#_Toc483675412)

[**Figure 19: Diagramme de Séquences « Authentification »** 55](#_Toc483675413)

[**Figure 21 : Diagramme de Séquences « Suppression produit »** 56](#_Toc483675414)

[**Figure 22 : Diagramme de Séquences « Ajouter produit au panier»** 57](#_Toc483675416)

[**Figure 23 : Diagramme de Séquences « Passer commande »** 58](#_Toc483675417)

[**Figure 24: Diagramme de Séquences « Gestion message»** 59](#_Toc483675418)

[**Figure 25 : Diagramme de Séquences « Authentification»** 60](#_Toc483675419)

[**Figure 27 : Diagramme de Séquences « Consulter liste produit»** 61](#_Toc483675420)

[**Figure 34 : Diagramme d'activité du processus de passage de la commande** 68](file:///C:\Users\kais\Desktop\rapport2017.docx#_Toc483675421)

[**Figure 35 : Diagramme d'états-transitions de la Commande** 69](#_Toc483675422)

[**Tableau 19 :Caractéristiques des outils utilisés** 71](#_Toc483675423)

[**Figure 36 : schéma logique de données** 75](file:///C:\Users\kais\Desktop\rapport2017.docx#_Toc483675424)

[**Figure 37 : les modèles navigationnels** 76](file:///C:\Users\kais\Desktop\rapport2017.docx#_Toc483675425)

[**Figure 38 : interface « accueil >>** 77](#_Toc483675426)

[**Figure 39 : interface « Liste produit »** 78](file:///C:\Users\kais\Desktop\rapport2017.docx#_Toc483675427)

[**Figure 40 : interface « menu client»** 79](#_Toc483675428)

**Introduction Générale**

Dans un monde actif et continuellement évolutif, la motivation d'avoir des moyens performants et efficaces de communication et d'échange d'informations devient de plus en plus fondamentale. Cette motivation donne naissance à une révolution favorisant le travail à distance et l'accès aux besoins en temps réduit à l’aide d’internet ce qui a bouleversé les habitudes de travail dans de nombreux métiers.

D’après des analyses diverses et statistiques effectuées, il s’avère que de plus en plus d’internautes se connectent désormais à internet via un ordinateur ou un Smartphone.

En effet, les systèmes d’informations ont répondu à un besoin vif pour n’importe quel type d’organisation, c’est la gestion d’information qui a parmi les enjeux les plus primordiaux pour les entreprises et touche pratiquement toutes les activités telles que la comptabilité, la production, la maintenance etc.

Dans le milieu de vente et d’achat en ligne, on assiste de plus en plus à la multiplicité, au commerce électronique, ceci a réagi les sociétés de vente qui proposent des promotions et des offres pour garantir la fidélité de ses clients.

La contribution principale de ce présent travail porte sur la création d’un site web dynamique et d’une application mobile pour la gestion du commerce électronique, afin de faciliter et de faire évoluer la communication avec la clientèle et de gérer les produits et les commandes.

Le présent rapport sera structuré en cinq chapitres :

* Le premier chapitre intitulé **« Etat de l’art »** qui mettra l’accent sur le champ d’étude de l’application et les nouvelles techniques utilisées dans le commerce électronique ainsi qu’une étude comparative des outils du marché qui supportent ces techniques.
* Le deuxième chapitre intitulé **« Etude Préalable »** sera consacré à la présentation du projet en précisant le cadre général du travail.
* Le troisième chapitre intitulé **« Analyse et Spécification des besoins»** présentera les acteurs du futur système, les besoins fonctionnels et non fonctionnels ainsi que le diagramme des cas d’utilisation générale, et les diagrammes des cas d’utilisation relatifs à chaque acteur.
* Le quatrième chapitre intitulé **« Conception»** auquel on élabore les diagrammes de séquence ainsi que le diagramme de classe complet.
* Le dernier chapitre intitulé **« Réalisation »** sert à présenter les environnements matériels et logiciels utilisés et à donner quelques interfaces de l’application.

# CHAPITRE 1 : Etat de l’art

## Introduction

L’état de l’art est une étape qui consiste à rechercher toutes les informations, techniques, nouveautés ou publications ayant un lien avec le domaine sur lequel on s’apprête à travailler.

Dans ce chapitre on va définir le champ d’étude ainsi que les techniques utilisées dans notre domaine d’étude.

## Recueil

Avec près de 60% de croissance en 2016 et alors que le commerce de détail recule de près de 12%, le commerce électronique rencontre un succès inédit, et constitue un véritable défi à la crise. Ce constat est toutefois à nuancer, tant le développement de cette nouvelle forme de vente est contrasté selon les biens et services commercialisés. Les achats "dématérialisés" (voyages, téléchargement de musique, de jeux ou d’applications pour téléphone portable), particulièrement adaptés au e-commerce, s’imposent logiquement en tête avec les produits technologiques (TV / Vidéo, matériel informatique) , alors que d’autres produits peinent à acquérir une position forte sur ce canal de vente. Il en va ainsi des denrées alimentaires dont le poids dans les ventes en ligne est plus discret, et la pénétration auprès des internautes est plus faible.

Le chiffre d’affaires du commerce électronique de détail devrait atteindre 70 milliards d’euros en 2016. Dans ce total, on estime que la vente de produits alimentaires représenterait au mieux 10 milliard d’euros. Les denrées alimentaires figurent parmi les catégories de produits les moins recherchées par les cyberacheteurs : à peine 11% d’entre eux déclarent avoir fait une commande de ce type au cours des 12 derniers mois, contre près de 50% dans certaines autres familles de produits.

1. **Définition du commerce électronique**

Le commerce électronique(ou commerce en ligne, vente en ligne ou à distance, parfois cybercommerce) est l'échange pécuniaire des biens, de services et d'informations par l'intermédiaire des réseaux informatiques comme Internet. On emploie également la dénomination anglaise e-commerce.

Dans le cadre du commerce intra-entreprise, on utilise depuis de nombreuses années des réseaux de type Échange de données informatisé (EDI). Des transactions électroniques se réalisent également sur les réseaux téléphoniques mobiles, on parle de m-commerce (mobile commerce).

Stricto sensu, le commerce électronique est défini comme l’emploi de moyens électroniques pour commander des biens et des services sur un réseau de télécoms. Il se confond alors avec la vente à distance, c’est-à-dire avec la possibilité d’acheter des biens sans avoir à se déplacer dans un magasin. Les observations et les débats portent le plus souvent sur cet aspect. On ne peut toutefois limiter l’impact de la numérisation sur le commerce à la seule électronisation de la fonction de commande. Les autres fonctions composant l’activité commerciale (la recherche d’informations, le paiement, le service après-vente, la relation client…) sont aussi impactées par l’emploi des réseaux électroniques. On passe ici du commerce électronique assimilé à la vente à distance à un phénomène plus général, porteur de transformations plus importantes à l’électronisation du commerce.

## Etude comparative des outils du marché

Avec 49,5 millions d’internautes ayant déjà effectué un achat en ligne, "*l’e-commerce séduit de nouveaux internautes chaque année*, annonce Bertrand Krug, le directeur adjoint de Médiamétrie, *ils sont près de 8 sur 10 à avoir déjà pratiqué l’achat en ligne, soit une augmentation de plus de 150% en 10 ans"*.

## Les 10 sites e-commerce les plus visités au premier trimestre 2017

| **Rang** | **Marques** | **Visiteurs uniques moyens par mois** | **Couverture moyenne (en % de la population internaute)** | **Visiteurs uniques moyens par jour** |
| --- | --- | --- | --- | --- |
| 1 | Amazon (B)\* | 18 134 000 | 37,9% | 2 010 000 |
| 2 | Cdiscount (B)\* | 11 175 000 | 23,4% | 947 000 |
| 3 | Fnac (B)\* | 9 328 000 | 19,5% | 668 000 |
| 4 | eBay (B)\* | 8 116 000 | 17,0% | 984 000 |
| 5 | Voyages-Sncf.com (B)\* | 6 440 000 | 13,5% | 466 000 |
| 6 | PriceMinister (B)\* | 5 874 000 | 12,3% | 453 000 |
| 7 | La Redoute (B)\* | 5 746 000 | 12,0% | 394 000 |
| 8 | vente-privee (B)\* | 5 360 000 | 11,2% | 826 000 |
| 9 | Darty (B)\* | 5 307 000 | 11,1% | 331 000 |
| 10 | Carrefour (B)\* | 5 286 000 | 11,1% | 393 000 |

**Tableau 1 : Comparaison des solutions existantes**

**CONCLUSION**

Comme le montre le tableau ci-dessus, et Après analyse des chiffres, des solutions existantes des sites web e-Commerce on voit donc bien que ce marché est très concurrentiel.

Il est donc très difficile pour les nouveaux sites de se démarquer et de gagner des parts de marché.

# CHAPITRE 2 : Etude Préalable

## 1. Introduction

Dans ce chapitre nous commençons à présenter l’organisme d’accueil au sein duquel nous avons effectué notre stage. Ensuite, nous donnerons un vue générale présentant une étude de l’existant. Enfin, nous proposons notre solution.

## Définition de la mission

Notre mission consiste à concevoir et développer une application web et une application mobile qui organise la gestion des ventes et d’achats en ligne et pour faciliter la communication avec les clients.

### Présentation de l’entreprise

Notre projet est réalisé au sein de la société NEJMA, une société commerciale dont sa mission est la vente et l’achat des poissons et des fruits de mer.

elle traite avec des hôtels, des restaurants, des pays étrangers (Maroc , Algérie, Portugals) et aux citoyens.

Cette entreprise situé au port de Sfax occupée 20 ouvriers, 3 techniciens, 2 ingénieurs, 5 transporteurs et une secrétaire.

Figure 1: Logo de Nejma

**Figure2: Organigramme de la société**

### Définition du champ d’étude

Des ventes de mains en mains vers des ventes virtuelles, passent les priorités des opérations de ventes des biens et des services, ce qui nous rend obligés de donner plus d’importance à la vente en ligne. Les boutiques en ligne sont depuis des années, largement conseillés pour les sociétés qui se basent sur la vente des produits et même des services. Ces types de sites web représentent un dispositif global fournissant aux clients un pont de passage à l’ensemble des informations, des produits et des services à partir d’un portail unique en rapport avec son activité. Les sites de vente en ligne permettent aux clients de profiter d’une foire virtuelle disponible et quotidiennement mise à jours sans la moindre contrainte, ce qui leur permettra de ne jamais rater les coups de cœur, ainsi une foire sans problèmes de distance géographique, ni d’horaire de travail ni de disponibilité de transport. D’ autre part ces sites offrent à la société un espace pour exposer ses produits à une plus large base de clientèle. Notre projet est réalisé dans le cadre du projet fin d’étude ayant comme objectif principal : la conception et la réalisation d’une boutique virtuelle pour le compte de NEJMA, une société commerciale de vente et d’achat.

Nous définissons ci-dessous les services de vente en ligne à notre clientèle:

## Etude de l’existant

L’étude de l'existant constitue le cœur de la phase d'analyse d'un projet. Cette étape est primordiale pour la mise en place de tout projet informatique ou autre, et qui permet de définir le contexte de fonctionnement ou bien le processus métier et dégager les différentes imperfections dans le système actuel afin de les corriger.

### Analyse de l’existant

La gestion traditionnelle d'une entreprise de vente de produit nécessite un grand effort de la part du personnel, il faudrait utiliser des formulaires en papier pour les clients, les contrats, des archives pour stocker toute ancienne opération ou transaction ce qui rend le travail pénible.

Les solutions informatiques courantes permettant de résoudre ces problèmes, ces solutions consistent à automatiser et informatiser la gestion à travers l'utilisation de différents logiciels.

L’achat d’un produit est un processus se fait par différentes méthodes :

* Un client passe une commande avec la méthode traditionnelle (en se présentent personnellement à l’entreprise).
* Un client passe une commande à travers le site web de l’entreprise.
* Un client appelle l’entreprise directement pour passer une commande.
* Un client envoie un email à l’entreprise pour passer une commande.

Ces méthodes peuvent dégager des problèmes.

Les clients cherchent les méthodes les plus simples, rapides et les plus modernes.

Pour plus d’information sur le cadre du projet, nous décidons d’analyser des solutions dans le marché de la gestion de vente en ligne afin de connaître les lacunes rencontrés.

Le tableau au-dessous représente des exemples des solutions dans le marché.

|  |  |
| --- | --- |
| **Site web** | **Environnement** |
| [Carfour\_Market](http://www.tunisiarentacar.com/Fr/anouar-rent-a-car_11_11) | HTML, Symfony,  PHP, Androïd et IOS |
| EBoyOOl | HTML, PHP |
| phoneNet | HTML, PHP |
| Royal Fish | Symfony, HTML, PHP |

**Tableau 2 : tableau des solutions existantes**

D’après ce tableau on remarque que :

* Les solutions mobiles sont limitées dans le marché.
* L’environnement JAVA n’est pas fortement utilisé dans le développement web.

Il est indispensable avant de se lancer dans la réalisation de tout projet, de bien étudier et analyser des projets similaires pour profiter des avantages et éviter les malveillances dans le présent projet. Pour cela on a visité des sites populaires de vente en ligne comme www.Mytek.tn et www.extrastores.com

1. **Analyse du Site Mytek**



**Figure 3: Partie publicitaire de Mytek**

* + 1. **Inconvénients :**

- Une très grande partie de la page réservée à la publicité de telle sorte que du premier coup l’utilisateur se sent pointé sur un site publicitaire et non pas sur un site de vente.

- Les catégories sont présentées sous forme d’une barre des menus ce qui limite l’ajout d’autres catégories

- Beaucoup de couleurs avec des tendances différentes causent une sorte d’incohérence entre les vues, le client se trouve perdu dans la grande variété des couleurs abandonnant le but principal du site.

- Les titres sont à 100% de niveau h4 et h5, une taille excessive

- Excès des images, des animations et des liens même en doublant les mêmes images dans la même page.

- Les liens contact, plan du site et favoris sont presque invisibles en haut de la page, ce qui diminue la chance de multiplicité des clients.

- Les meilleures ventes et les nouveaux articles sont affichés au milieu de la page avec les autres produits, ce qui ne leur attire point l’attention.

- Le panier est disponible pour tout le monde sans inscription, ce qui provoque les commandes non rigoureuses entrainant au remplissage de la base de données sans le moindre intérêt.

**2. Avantage :**

- Titres claires.

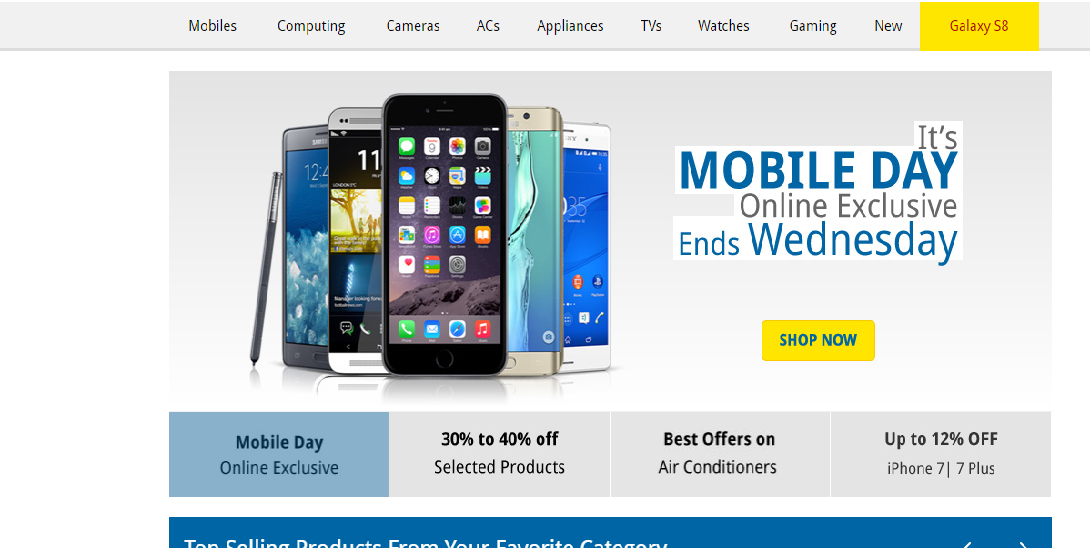
- Informations disponibles sur la même page ce qui donne une idée très claire sur les produits dès la première visite des pages. - Les animations se diffèrent par catégorie.

- Les liens des réseaux sociaux et des sites de même intérêt sont disponible en permanence.

- Les contacts des boutiques de Mytek sont clairs et nettes.

**II. Analyse du Site extrastores.**

**1-Inconvénients :**



**Figure 4: Partie publicitaire d’extrastores**

- Partie publicitaire énorme, qui prend une grande partie de la page

- Les catégories se présentent sous forme de sous titres en bas de la page ce qui ne donne pas l’intention des liens ou des sélections des catégories des produits.

- Les promotions et les nouveaux articles ne s’affichent pas en premier lieux pour attirer l’attention de l’utilisateur, ils sont affichés entre les autres articles.

- Le panier est aussi disponible pour tout le monde comme la majorité des sites de vente en ligne ce qui rend l’essayage des commandes sans achat plus simple et donc entraine la présence des milliers de commandes inutiles dans la base de données.

**2. Avantage :**

Pour le site web d’extrastores, on trouve plusieurs avantages surtout en ce qui concerne l’aspect ergonomique des pages.

- Une cohérence entre les tendances des couleurs choisies.

- Les liens sont clairs.

- La barre des menus représente une variété des liens logiques d'une importance primordiale.

1. **Synthèse**

|  |  |
| --- | --- |
| ***Les inconvénients rencontrés*** | ***Nos solutions proposées*** |
| L’incohérence des couleurs utilisées dans les pages des sites. | Utilisation des tendances d’une seule couleur, on peut choisir la couleur bleu et ses dérivés. |
| Les doublets et triplets des liens dans la même page. | Spécification des parties de la page c'est-à-dire chaque partie sera dédiée à son propre besoin. |
| La disparition de la barre des menues et de la bannière publicitaire en descendant vers le bas des pages. | Division des pages Template en sections fixes qui s’appellent en java les « playout » |
| Les problèmes d’ajout des catégories et leurs influences sur l’ergonomie des pages du site. | Présenter les catégories dans une liste déroulante qui reste toujours flexible à ajouter autant de catégories que possible. |
| La disponibilité de panier pour tout le monde sur la page accueil, menant à un bourrage de commandes non validées. | Le panier sera dans notre site disponible seulement pour les clients après inscription et authentification. |
| Un grand nombre d’images sur la même page. | Utilisation d’une «carousel» un panneau qui affiche seulement trois produits et donc max 10 images par page. |

**Tableau 3 : Les inconvénients et les solutions**

### Critique de l’existant

**A. Description de l’existant**

Comme toutes les sociétés commerciales, NEJMA possède sa méthode de présenter et de commercialiser ses produits. Cette méthode est divisée en deux principales étapes, la premières étape est l’exposition des produits par des affiches publicitaires, des dépliants, des spots publicitaire dans les radios et les télévisions et aussi par des vitrines qui se trouvent au sein de la société, la deuxième étape consiste à vendre les produits à guichet ou par l’intermédiaire des agents commerciaux.

**B. Critiques de l’existant**

Après avoir mené une étude sur l’étude de l’existant, on souligne les problèmes suivants :

* Site web non adaptatif (accessible pour toutes les appareils : pc, Smartphone, tablette …).
* manque de diffusion des informations plus rapide
* Absence d’ergonomie
* Temps de réponse non optimisé
* Manque de gestion des Exceptions (une exception ou une erreur peut dégager après une action humain).
* Absence d’application mobile qui suit en temps réel les commandes, les messages, et les produits.

## Solution proposée

Après avoir étudié l’existant et détecté les anomalies, on a vu que les solutions mobile sont limité dans le marché et que la plate-forme Java Entreprise Edition (voir l’annexe) n’est pas utilisé comme un environnement de développement.

Java Entreprise Edition a été conçu comme un environnement pour développer, déployer et exécuter des applications réparties pour le monde de l'entreprise. Ce contexte de l'entreprise se caractérise généralement par la nécessité d'assurer des niveaux de qualité de service tels que la sûreté de fonctionnement, la résistance à des charges d'exécution importantes ou encore la sécurité. Pour pouvoir répondre aux besoins des entreprises, nous nous sommes intéressés à la réalisation d’une application web et une application mobile dédiés à la gestion du vente en ligne afin de satisfaire les besoins de l’entreprise et des clients, on va implémenter un outil pour l’inscription des clients et un outil pour la gestion de vente en ligne (produits, clients, commandes etc.)

Notre futur système est composé de deux parties :

* Partie web : c’est l’application principale qui est utilisé par les gestionnaires et les clients.
* Partie mobile : c’est une partie destinée aux clients mobiles

## Planning prévisionnel

La clé principale de la réussite d’un projet est la présence d’un planning préétablie. Ce dernier nous permettra de décomposer le projet en un ensemble de tâches et d’activités afin d’estimer la période nécessaire pour la réalisation de chacune d’elle. Pour cela, et pour la réalisation de notre stage de fin d’études, on estime avoir le planning suivant :

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Semaine**  **Etape** | **Janvier** | | | | **février** | | | | | **mars** | | | | | **Avril** | | | | | **mai** | | | |
| **1** | **2** | **3** | **4** | | **1** | **2** | **3** | **4** | | **1** | **2** | **3** | **4** | | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| Etude préalable |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  |  |  |  |  |
| Conception |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  |  |  |  |  |
| Réalisation |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  |  |  |  |  |
| Test et Validation |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  |  |  |  |  |
| Rédaction du rapport |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  |  |  |  |  |

**Tableau 4 :Planning de réalisation du projet**

* Etude préalable : le résultat de cette phase est la détermination des objectifs à atteindre dans notre future application en partant de l’existant.
* Conception : il s’agit de détailler les spécifications des fonctions ainsi que la structure des données, des contrôles et des interfaces.
* Réalisation : il s’agit de réaliser l’implémentation des programmes et d’effectuer les tests unitaires.
* Test et Validation : il s‘agit de tester l’application
* Rédaction du rapport : description détaillée de notre travail.

## Conclusion

Dans ce chapitre on a effectué une étude de l’existant dans le marché, puis on a présenté ses lacunes. Ensuite nous avons présenté notre solution pour résoudre ces problèmes rencontrés.

Dans le chapitre suivant, on présentera les besoins fonctionnel et les besoins non fonctionnel ainsi que les diagrammes des cas d’utilisation.

# CHAPITRE 3 : Analyse & Spécification des besoins

## Introduction

La phase d'analyse et de spécification est la première étape du processus de développement qu’on a adopté. En effet, elle formalise et détaille ce qui a été ébauché au cours de l'étude préalable et permet de dégager l'étude fonctionnelle du système. Elle permet ainsi d'obtenir une idée sur ce que va réaliser le système en termes de métier (comportement du système). Tout au long de ce chapitre, on commencera par définir les besoins fonctionnels et non fonctionnels de la solution qu’on va proposer et présenter les diagrammes des cas d’utilisation.

## Spécification des besoins

La spéciation établit ce que le système doit faire (le QUOI) et les contraintes sous lesquelles il doit opérer.

L’ingénierie de la spéciation consiste donc à établir une communication entre les clients et les concepteurs du système

### . Les besoins fonctionnels

Les besoins fonctionnels sont les services offerts par notre système. Ils capturent les tâches ou activités qui doivent être formulées par l'application pour différents types d’acteurs.

Nous allons lister les différentes fonctionnalités de notre application :

* + L’inscription des clients : permet à un nouvel client (visiteur) de créer son propre compte
  + L’authentification des utilisateurs : permet à chaque utilisateur de s’authentifier et d’accéder à son espace de travail.
  + La consultation des produits et des tarifs : permet au client de consulter les tarifs et les produits de mer ainsi que leurs détails afin de voir les offres et les promotions.
  + La consultation des produits et des tarifs: permet au visiteur de consulter les tarifs et les produits de mer ainsi que leurs détails afin de voir les offres et les promotions.
* La gestion de panier : permet au client de gérer un panier (Ajouter, supprimer un produit, vider).
* La gestion des produit favori : permet au client de gérer une liste des produits (Ajouter, supprimer un produit, vider).
* Gestion des commandes (choix du type de paiement, choix de l’offre de paiement, choix du type de livraison)
* Le paiement de la commande en ligne.
* La confirmation de l’opération d’achat.
* La consultation des commandes : permet au client de consulter les commandes.
* La gestion des catégories : permet au gestionnaire de gérer les catégories (Ajouter, Modifier, Supprimer, Ajouter promotion).
* La gestion des produits : permet au gestionnaire de gérer les produits (Ajouter, Modifier, Supprimer, Ajouter promotion).
* La gestion des messages : permet au gestionnaire de gérer les messages (envoyer, répondre, supprimer).
* La gestion des utilisateurs : permet au gestionnaire de gérer les utilisateurs (Ajouter, Supprimer, Modifier, Accepter inscription).
* La notification au client, lorsqu’une commande est acceptée : permet de notifier le client en temps réel sur son Smartphone lorsqu’une réservation est confirmée.
* Les statistiques des ventes et des stocks.

### . Les besoins non fonctionnels

Les besoins non fonctionnels concernent les contraintes à prendre en considération pour mettre en place une solution adéquate aux attentes des concepteurs des architectures dynamiques.

Ces besoins sont important car ils agissent de façon indirecte sur le résultat et sur le rendement de l’utilisateur ce qui fait qu’ils ne doivent pas être négligés, pour cela il faut répondre aux exigences suivantes :

1. Fiabilité:

L’application doit fonctionner de façon cohérente sans erreurs et doit être satisfaisante.

1. Les erreurs :

Les ambigüités doivent être signalées par des messages d’erreurs bien organisés pour bien guider l’utilisateur et le familiariser avec notre site web.

1. Ergonomie et bonne Interface :

L’application doit être adaptée à l’utilisateur sans qu’il ne fournisse aucun effort (utilisation claire et facile) de point de vue navigation entre les différentes pages, couleurs et mise en textes utilisés.

1. Sécurité :

Notre solution doit respecter la confidentialité des données personnelles des clients qui reste l’une des contraintes les plus importantes dans les sites web.

1. Aptitude à la maintenance et la réutilisation :

Le système doit être conforme à une architecture standard et claire permettant sa maintenance et sa réutilisation.

1. Compatibilité et portabilité :

Un site web, quel que soit son domaine, son éditeur et son langage de programmation ne peut être fiable qu’avec une compatibilité avec tous les navigateurs web et tous les moyens que ce soit PC, IPAD ou Mobiles.

Notre application doit nécessairement assurer ces besoins :

* L’interface : avoir une application qui respecte les principes des Interfaces Homme/Machine (IHM) tels que l'ergonomie et la fiabilité.
* La performance : l’application devra être performante c'est-à-dire que le système doit régir dans un délai précis, quel que soit l’action de l’utilisateur.
* La convivialité : l’application doit être simple et facile à manipule.
* L’ergonomie: le thème adopte par l’application doit être inspiré des couleurs et du logo type de l’entreprise d’accueil.
* La sécurité : l’application devra être hautement sécurisée, les informations ne devront pas être accessibles à tout le monde, c'est-à-dire que le site web est accessible par un identifiant et un mot de passe attribué à une personne physique.
* L’extensibilité : dans le cadre de ce travail, l'application devra être extensible, c'est-à dire qu'il pourra y avoir une possibilité d'ajouter ou de modifier de nouvelles fonctionnalités.

## Diagrammes des cas d’utilisation

Le diagramme des cas d’utilisation décrit le comportement du système du point de vue utilisateur sous la forme d’action et de réaction. Un cas d’utilisation indique une fonctionnalité du système déclenché par un acteur externe au système. Dans ce diagramme interviennent trois éléments : les acteurs, le système et les cas d’utilisation. L’acteur représente un rôle joué par une personne ou un autre système en cours de la modélisation.

Dans ce qui suit, nous allons présenter les principaux diagrammes des cas d’utilisation relatifs aux différents acteurs, après les avoir identifiés.

### Identification des acteurs

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel, ou autre système) qui interagit directement avec le système étudié, en échangeant les informations (en entrée et en sortie). On identifie les acteurs en observant les utilisateurs directs du système, les responsables de la maintenance, ainsi que les autres systèmes qui interagissent avec lui. Dans notre contexte, nous avons dégagé 4 acteurs à savoir :

* ***Administrateur :*** c’est l’administrateur de l’application.
* ***Secrétaire :*** elle peut gérer les produits, les catégories, accepter les inscriptions et gérer aussi les commandes et les factures et consulter les statistiques.
* ***Client :*** il peut consulter les messages, les produits, les produit favoris, les commandes, le panier et les factures en plus passer une commande et contacter l’entreprise par message, ajouter un produit vers le panier ou dans la liste des favoris, il est considéré comme un client actif.
* **Le** **visiteur** : c’est un individu qui est entrain de fouiller sur le net, cherchant un produit pour l’acheter ou pour avoir une idée sur les modèles et les prix. Jusqu’à ce stade c’est un utilisateur inconnu donc il n’est pas encore un client.

### Identification des cas d'utilisation

Dans ce qui suit, nous décrivons la liste des cas d’utilisation par acteur.

**Administrateur**

* + S’authentifier
* Gérer les catégories
* Gérer les utilisateurs du système
* Gérer les produits
* Gérer les messages
* Ajouter des promotions(produit, catégorie)
* Consulter les statistiques

**Secrétaire**

* + S’authentifier
* Gérer les catégories
* Gérer les utilisateurs du systéme
* Gérer les produits
* Gérer les messages
* Ajouter des promotions (produit, catégorie)
  + Gérer les commandes
  + Gérer les factures
  + Consulter les statistiques

**Client**

* + S’inscrire
  + S’authentifier
  + Consulter les catégories
  + Consulter les produits
  + Consulter les offres
  + Ajouter un produit au panier
  + Ajouter un produit à la liste favori
  + Passer une commande
  + Modifier son profil
  + Consulter la liste ses commandes
  + Consulter la liste ses factures

**Visiteur**

* + S’inscrire
  + Consulter les catégories
  + Consulter les produits
  + Consulter les offres

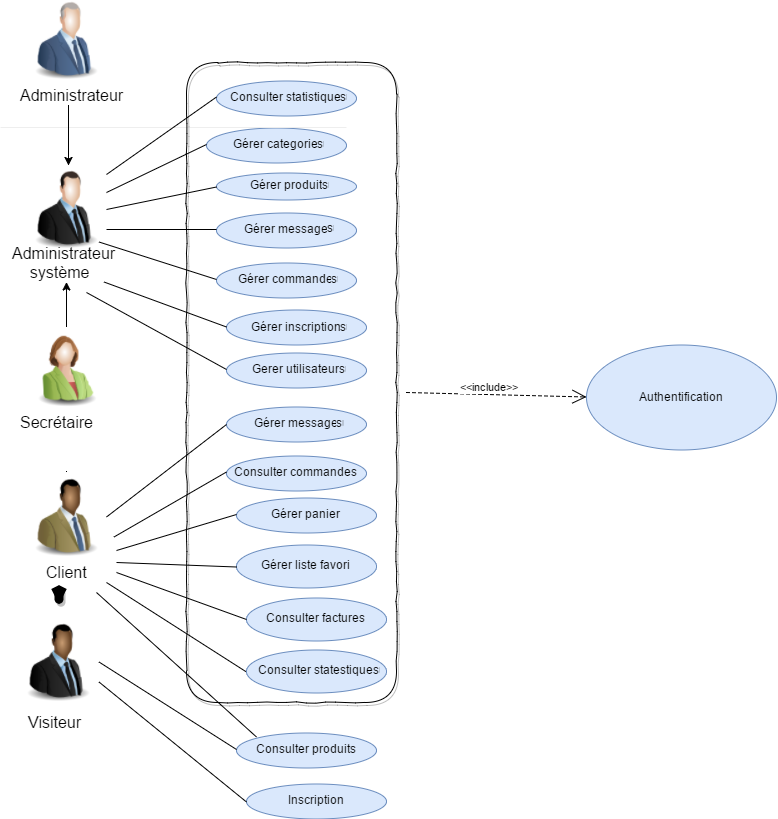
### Présentation des diagrammes des cas d’utilisation

Dans ce qui suit on va présenter le diagramme des cas d’utilisation générale ainsi que les diagrammes des cas d’utilisation relatifs à chaque acteur.



#### Diagramme de cas d’utilisation général

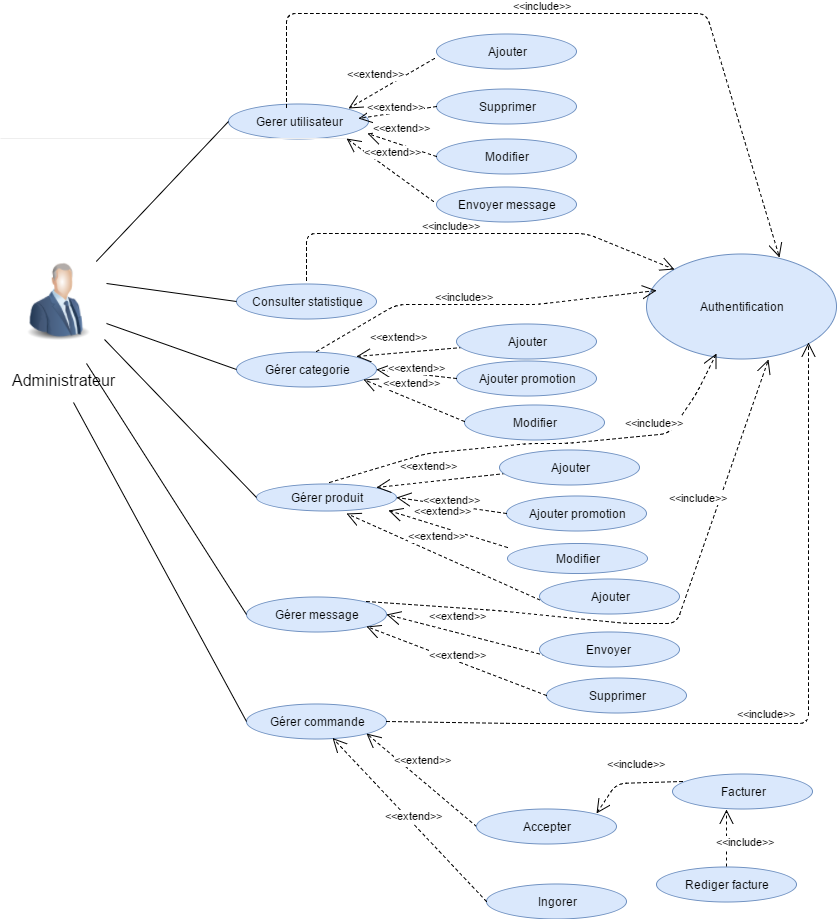
Ce diagramme représente les cas d’utilisation les plus importants et décrit le système d’une façon générale en précisant les acteurs qui vont utiliser l’application



**Figure 5 : diagramme des cas d’utilisation  général**

#### 3.3.2 Diagramme de cas d’utilisation relatif à « l’administrateur »

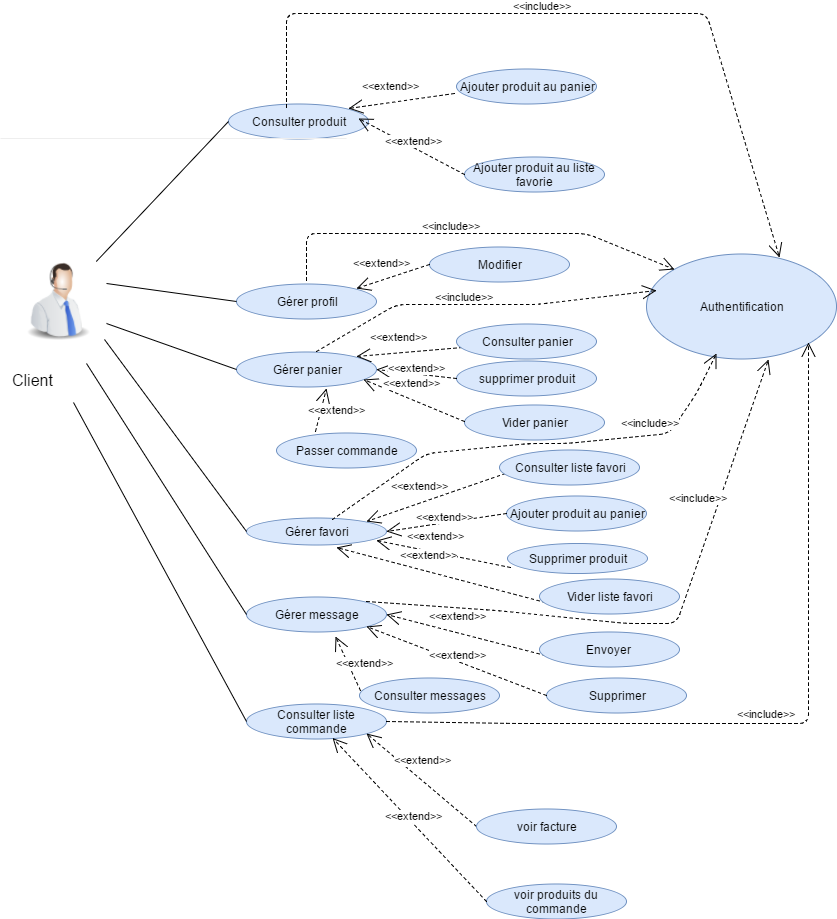
La figure ci-dessous représente le diagramme de cas d’utilisation relatif à l’administrateur.



**Figure 6 : diagramme de cas d'utilisation relatif à « l’administrateur »**

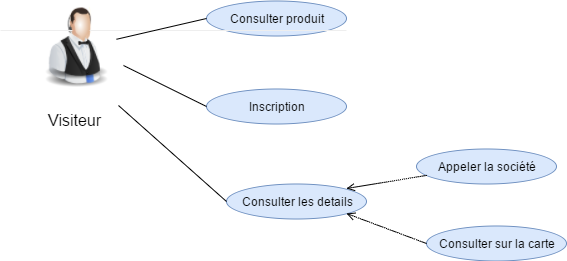
#### Diagramme de cas d’utilisation relatif au « client »

La figure ci-dessous représente le diagramme de cas d’utilisation relatif au client.



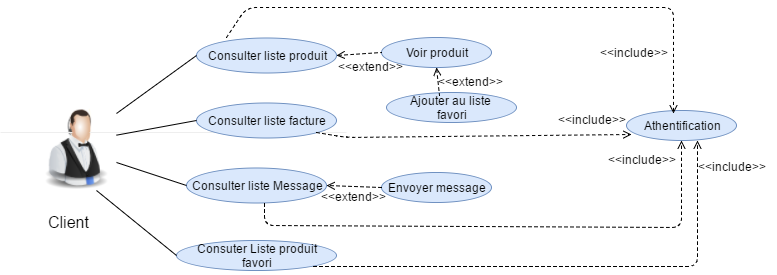
**Figure 7 : diagramme de cas d'utilisation relatif au « client »**

#### Diagramme de cas d’utilisation relatif au « visiteur »



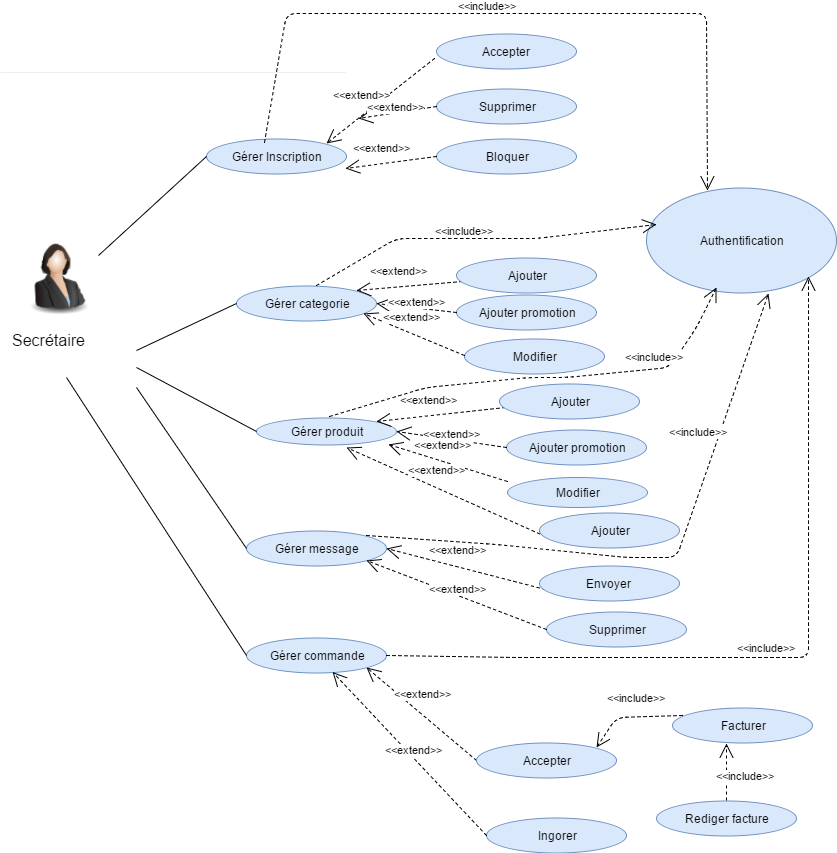
**Figure 8 : diagramme de cas d'utilisation relatif au « visiteur »**

* + 1. **Diagramme de cas d’utilisation partie mobile relatif au « Client »**



**Figure 9 : diagramme de cas d'utilisation partie mobile relatif au « client »**

#### Diagramme de cas d’utilisation relatif au « secrétaire »



**Figure 10 : diagramme des cas d'utilisation relatif au « Secrétaire»**

### Description textuelle des principaux cas d'utilisation

La description textuelle des cas d’utilisation donne une vue d’ensemble sur les différentes fonctionnalités du système et elle peut être affinée par des scenarios.

Dans le but de mieux comprendre notre système et les interactions avec les utilisateurs, nous allons détailler dans cette partie les scenarios des principaux cas d’utilisation.

|  |
| --- |
| **CU: Authentification** |
| **But:** Gérer l’authentification pour chaque utilisateur |
| **Acteurs :** Client, Administrateur, Secrétaire |
| **Pré-condition :** |
| **Post-condition :** Accès au module concerné |
| **DESCRIPTION DU SCENARIO NOMINAL**  « DEBUT»  01 : Le système invite l’acteur à entrer son login et son mot de passe.  02: L’acteur saisit le login et le mot de passe.  03: Le système vérifie les paramètres.  04: Le système ouvre l'espace de travail correspondant à l’utilisateur.  « FIN» |
| **DESCRIPTION DU SCENARIO ALTERNATI F**  Le login ou le mot de passe sont incorrects : ce scénario commence au point 03 du scénario  nominal.  01 : Le système informe l’acteur que les données saisies sont erronées et le scénario reprend  au point 01 du scénario nominal. |

**Tableau 4:Description textuelle du cas d’utilisation« authentification »**

|  |
| --- |
| **CU : Gérer les catégories** |
| **But :** Ce CU permet aux acteurs de gérer les catégories |
| **Acteurs :** secrétaire, administrateur |
| **Pré-condition:** l’acteur est authentifié |
| **DESCRIPTION DU SCENARIO NOMINAL**  « DEBUT»  01 : L’acteur choisie le menu des catégories.  02 : le système affiche la liste des catégories  03 : L’acteur clique sur Ajouter ou Supprimer ou Ajouter promotion  04 : le système valide la confirmation du processus.  « FIN» |

**Tableau 5 :Description textuelle du cas d’utilisation « gérer les Catégories»**

|  |
| --- |
| **CU : Gérer les produits** |
| **But :** Ce CU permet aux acteurs de gérer les produits |
| **Acteurs :** secrétaire, administrateur |
| **Pré-condition:** L’acteur est authentifié |
| **DESCRIPTION DU SCENARIO NOMINAL**  « DEBUT»  01 : la secrétaire choisie le menu des produits.  02 : le système affiche la liste des produits  03 : la secrétaire clique sur Ajouter ou Supprimer  04 : le système valide la confirmation de l’Ajouter ou la suppression.  « FIN» |

**Tableau 6 : Description textuelle du cas d’utilisation « gérer les produits»**

|  |
| --- |
| **CU : Gérer les messages** |
| **But :** Ce CU permet au secrétaire de gérer les messages |
| **Acteurs :** secrétaire, administrateur |
| **Pré-condition:** L’acteur est authentifié |
| **DESCRIPTION DU SCENARIO NOMINAL**  « DEBUT»  01 : L’acteur choisie le menu des messages.  02 : le système affiche la liste des messages  03 : L’acteur clique sur Ajouter ou répondre  0 4 *:* L’acteur saisir le message.  05 : L’acteur clique sur envoyer.  06 : le système valide la confirmation de l’envoi du message.  « FIN» |

**Tableau 7 :Description textuelle du cas d’utilisation « gérer les messages»**

|  |
| --- |
| **CU: Consulter la liste des produits** |
| **But :** Ce CU permet au client et le visiteur de consulter des produits ainsi que leurs détails |
| **Acteurs :** Visiteur, Client |
| **Pré-Condition :** |
| **Post-Condition :** le client accédé aux détails du produit |
| **DESCRIPTION DU SCENARIO NOMINAL**  « DEBUT»  01 : le client sélectionne le menu de la liste des produits  02 : Le système affiche la liste des produits  03 : le client sélectionne un produit  04 : Le système affiche les détails du produit dans un model  « FIN» |

**Tableau 8 : Description textuelle du cas d’utilisation « consulter la liste des produits »**

|  |
| --- |
| **CU : Ajout produit au panier** |
| **But :** Ce CU permet au client de remplir le panier avec un ou d’ensemble de produit |
| **Acteurs :** client |
| **Pré-condition:** le client est authentifié |
| **Post-Condition :** la quantité du produit égale à 0 ou supérieur de la quantité de stock |
| **DESCRIPTION DU SCENARIO NOMINAL**  « DEBUT»  01 : le client clique sur ajouter au panier.  02 : le client saisies la quantité du produit dans le model.  03: le système vérifié si le client est authentifié.  04 : le système vérifié si produit est disponible.  05 : une notification s’affiche montre l’ajout du produit et valider.  06 : le système informe le client le bon déroulement de l’ajout du produit  « FIN» |
| **DESCRIPTION DU SCENARIO ALTERNATI F**  Le client n’est pas authentifié : le système ouvre l’interface d’authentification puis il réoriente le client vers l’interface liste produit  La quantité n’est pas disponible : un message erroné s’affiche  Les données sont erronées : le système informe le client que les données sont erronées. |

**Tableau 9 :Description textuelle du cas d’utilisation « Ajouter produit au panier »**

|  |
| --- |
| **CU : Ajout produit à la liste favorite** |
| **But :** Ce CU permet au client de remplir la liste favori avec un ou d’ensemble de produits |
| **Pré-condition:** le client est authentifié |
| **DESCRIPTION DU SCENARIO NOMINAL**  « DEBUT»  01 : le client clique sur ajouter au favori.  02: le système vérifié si le client est authentifié.  03 : une notification s’affiche montre l’ajout du produit et valider.  04 : le système informe le client le bon déroulement de l’ajout du produit  « FIN» |
| **DESCRIPTION DU SCENARIO ALTERNATI F**  Le client n’est pas authentifié : le système ouvre l’interface d’authentification puis il réoriente le client vers l’interface liste produit  Les données sont erronées : le système informe le client que les données sont erronées. |

**Tableau 10 :Description textuelle du cas d’utilisation « Ajouter produit au liste favori »**

|  |
| --- |
| **CU : passer une commande** |
| **But :** Ce CU permet au client de passer une commande d’un ou d’ensemble de produit |
| **Acteurs :** client |
| **Pré-condition:** le client est authentifié |
| **DESCRIPTION DU SCENARIO NOMINAL**  01 : le client clique sur consulter le panier.  02 : le système vérifié si le client est authentifié.  03 : le système affiche l’interface panier.  04 : le client clique sur contenu traitement de la commande.  05 : le client choisi le mode de livraison.  06 : le client choisi l’offre de paiement.  07 : le client choisi le type du paiement.  05 : le système vérifie les informations saisies.  06 : le système informe le client sur le bon déroulement de la passage de la commande |
| **DESCRIPTION DU SCENARIO ALTERNATI F**  Le client n’est pas authentifié : le système ouvre l’interface d’authentification puis il réoriente le client vers l’interface réservation |

**Tableau 11 : Description textuelle du cas d’utilisation « passer commande »**

|  |
| --- |
| **CU : Gérer les commandes** |
| **But :** Ce CU permet au secrétaire de gérer les commandes |
| **Acteurs :** secrétaire, administrateur |
| **Pré-condition:** la secrétaire est authentifiée |
| **Post-Condition :** la secrétaire confirme ou annule les commandes |
| **DESCRIPTION DU SCENARIO NOMINAL**  « DEBUT»  01 : la secrétaire choisie le menu des commandes.  02 : le système affiche la liste des commandes  03 : le gérant clique sur confirmer ou annuler  04 :le système valide la confirmation de la commande ou l’annulation.  « FIN» |

**Tableau 13: Description textuelle du cas d’utilisation « gérer les commandes »**

|  |
| --- |
| **CU : Facturer une commande** |
| **But :** Ce CU permet au secrétaire de gérer les factures |
| **Acteurs :** secrétaire, administrateur |
| **Pré-condition:** la secrétaire est authentifiée |
| **Post-Condition :** la secrétaire n’a pas confirmé la commande |
| **DESCRIPTION DU SCENARIO NOMINAL**  « DEBUT»  01 : la secrétaire choisie le menu commandes.  02 : le système affiche la liste des commandes  03 : le gérant clique sur facturer la commande  04 : le système valide la confirmation de la facturation de la commande  05 : la secrétaire clique sur voir facture.  06 : le système affiche la facture de la commande  « FIN» |
|  |

**Tableau 12 : Description textuelle du cas d’utilisation « Facturer commande»**

## Conclusion

Ce chapitre présente une phase indispensable pour l’étude et l’analyse de notre application. Nous avons défini les différents besoins fonctionnels et non fonctionnels, nous avons présenté le diagramme des cas d’utilisation général ainsi que les digrammes des cas d’utilisation relatif à chaque acteur. Nous entamerons dans le chapitre suivant la conception de cette application qui comporte les diagrammes de séquence, les diagrammes d’activités et le diagramme des classes.

# CHAPITRE 4 : Conception

## Introduction

Dans le cycle de vie de notre projet, la conception représente une phase primordiale et déterminante pour produire une application de haute qualité.

Dans ce chapitre, on va détailler l’étude conceptuelle de la solution qu’on a proposée. En se basant sur le langage UML, nous allons élaborer le diagramme des classes du système permettant ainsi de présenter les diagrammes de séquence. Ensuite, on va montrer l’évolution de quelques objets clés en utilisant le diagramme d’états-transition et le diagramme d’activités.

## Environnement de modélisation UML

UML est un **langage** qui permet de représenter des modèles, mais il ne définit pas le processus d'élaboration des modèles : UML n’est donc pas une méthode de modélisation.

Cependant, dans le cadre de la modélisation d'une application informatique, les auteurs d'UML préconisent d'utiliser une démarche :

* **itérative et incrémentale :** Pour modéliser (comprendre et représenter) un système complexe, il vaut mieux s'y prendre en plusieurs fois, en affinant son analyse par étapes.
* **guidée par les besoins des utilisateurs du système :** Avec UML, ce sont les utilisateurs qui guident la définition des modèles.
* **centrée sur l'architecture logicielle :** Une architecture adaptée est la clé de succès d'un développement. Elle décrit des choix stratégiques qui déterminent en grande partie les qualités du logiciel (adaptabilité, performances, fiabilité etc.).

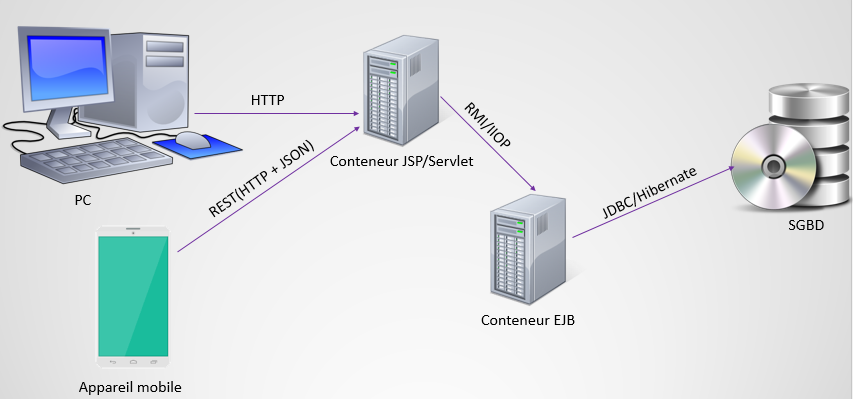
## Conception générale



#### Vue architecturale du système : Architecture 4-tiers

On va développer l’application en se basant sur l’architecture quatre-tiers. La figure suivante représente notre architecture adoptée.

La figure représente des différents composants de cette architecture :



**Figure 11 : architecture 4-tiers**

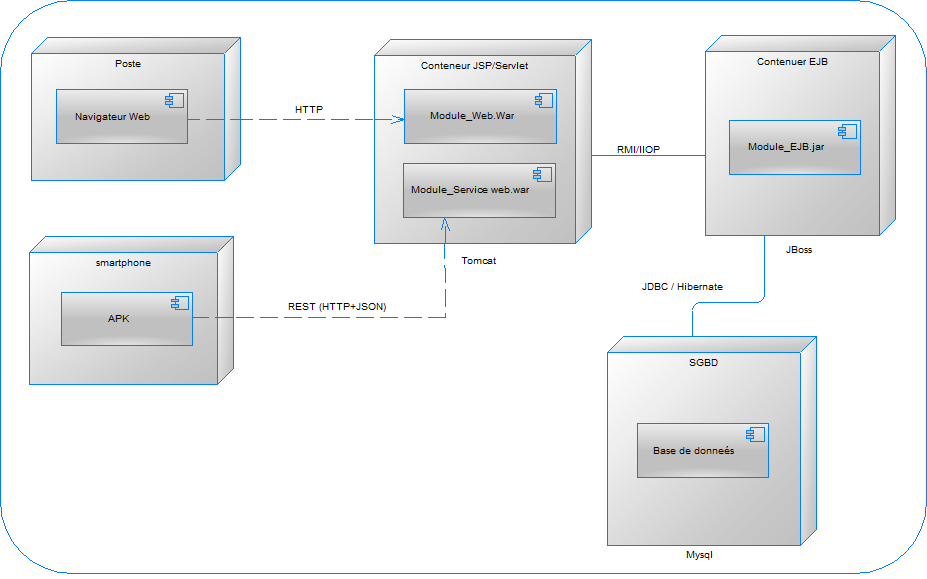
* Pc : il s’agit d’un client léger (navigateur web)
* Appareil mobile : il s’agit d’un client mobile (Android)
* Conteneur JSP/Servet : il s’agit d’un serveur d’application (JBOSS)
* Conteneur EJB : il s’agit d’un serveur d’application (JBOSS)
* SGBD : serveur de base de données (MySQL)

#### Diagramme de déploiement

Le diagramme de déploiement représente la vue physique d'une architecture en précisant les relations entre les composants et les équipements (les nœuds).

Le diagramme de déploiement spécifie un ensemble de constructions qui peuvent être utilisés pour définir l'architecture d'exécution des systèmes qui représentent l'affectation d'artefacts.

Le diagramme de déploiement de notre application est représenté par le diagramme ci-dessous:

**Figure 12: Diagramme de déploiement**

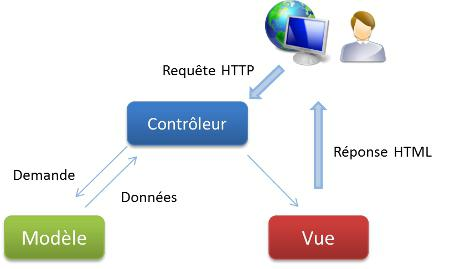
* + **Poste** : il s’agit d’un navigateur web ou un Smartphone
  + **Conteneur JSP/Servlet :** il s’agit d’un serveur web pour notre cas on utilise JBOSS.
* Module Web : un module web sera déployé sur apache Tomcat.
* Module Service Web : un module contient des services web d’une architecture REST à l’aide de GSON sera déployé sur apache Tomcat.
  + **Conteneur EJB :** il s’agit d’un serveur d’applications, pour notre cas on utilise JBOSS
* Module EJB : un module EJB qui géré la logique métier de l’application sera déployé sur JBOSS.
  + **SGBD** : c’est un serveur de base de données, pour notre cas on utilise MySQL.
* Base de donnés : une base de donnes relationnelles sera implémentée sur MySQL.

### Découpage en couches

On présente dans cette partie le découpage logique global des couches de notre application en se basant sur une architecture logicielle 4 tiers. Puis, on explique chaque couche en suivant une approche ascendante du niveau le plus bas vers le plus haut.

Notre conception est guidée par le choix des Framework utilisés pour réaliser le projet (détaillés dans le chapitre 5).

Ces Framework imposent le modèle MVC (Modèle-Vue-Contrôleur)



**Figure 13 : Le modèle MVC**

* **Contrôleurs :** permet la gestion des événements et la synchronisation du modèle et de la vue. Il traite tous les événements de l’utilisateur et éclanche par la suite les actions nécessaires à effectuer.
* **Vues :** Correspond à l’interface Homme/Machine. Elle permet d’afficher les données récupérées du modèle. Elle permet aussi de récupérer les actions de l’utilisateur et d’envoyer des événements au contrôleur. Aucun traitement n’est effectué dans cette entité.
* **Modèles :** Cette dernière couche d'abstraction permet de faire le lien entre le contrôleur et la base de données. Nous avons présentée par la suite le découpage logique adopté pour réaliser notre application nous avons appliqué un  découpage en couches.

**Figure 14 : les couches de l'application**



#### Couche « présentation »

C’est la couche la plus proche de l’utilisateur qui assure l’affichage et l’impression de l’interface graphique. Elle interprète les évènements de l’utilisateur et assure l’appel la couche Controller

#### Couche « Controller »

Elle permet de contrôler la cinématique des écrans, invoquer les appels aux services et permet de gérer les sessions utilisateurs.

#### Couche « Services »

C’est le cœur de fonctionnalité de l’application, elle présente le logique métier et expose la totalité des services aux utilisateurs.

#### Couche « domaine »

Cette couche représente tout objet nécessitant d’être stocké en base. Elle assure l’indépendance totale entre le client et le type de stockage utilisé (SGBDR, SGBDO, fichiers XML, …).

#### Couche « persistance »

Cette couche fournie les services de stockage des données vers la base de données, elle transforme les données en représentation objet.

## Conception détaillée



### Modèle de données

Maintenant nous allons présenter le modèle des données de notre application, nous utilisons le diagramme des classes pour expliquer notre modèle.

Le diagramme des classes exprime la structure statique du système en termes de classes et de relations.

Nous présentons dans ce qui suit :

* Le dictionnaire des données
* La représentation des associations entre les classes
* Le diagramme de classes

#### Dictionnaire des données

|  |  |  |
| --- | --- | --- |
| **Classe** | **Attribut** | **Description** |
| **Produits\_marins** | id  Nom  libelle  catégorie  prix  Quantité  Image  type | Identifiant du produit  Nom du produit  Libelle du produit  catégorie du produit  prix du produit  quantité en stock du produit  image du produit  le type du produit (frais/congeler) |
| **Promotion** | id  nbjour  prix | Identifiant de la promotion  Nombre de jour pour la promotion  Prix de l’offre |
| **Message** | Id  Objet  Message  Valide  Type  Adresse Emetteur | Identifiant du message  Objet du message  Le contenu du message  La validité du message (répondu, en attente)  Le type du message (envoyé, reçu)  L’adresse de l’émetteur |
| **Panier** | Id  Mantant | Identifiant du panier |
| **Ligne\_commande** | Id | Identifiant de la ligne commande |
| **Administrateur** | Cin  Nom  Prenom  Login  Password  Adresse  Telephone  Adresse Gmail | Cin de l’administrateur  Nom de l’administrateur  Prenom de l’administrateur  Login de l’administrateur  Password de l’administrateur  Adresse de l’administrateur  Telephone de l’administrateur  Adresse gmail de de l’administrateur |
| **Client** | Cin  Nom  Prenom  Login  Password  Adresse  Adresse livraison  Telephone  Adresse Gmail  Point | Cin du client  Nom du du client  Prenom du client  Login du client  Password du client  Adresse du client  Adresse livraison du client  Telephone du client  Adresse gmail du client  L’ensemble des points d’achat du client |
| **Facture** | Id\_facture  Date\_facture  Montant | Identifiant de la facture  Date de la facture  Montant de la facture |

**Tableau 14 :Dictionnaire des données**

#### Classe d’association

Une classe d'association est une classe qui fait partie d'une relation d'association entre deux autres classes.

Nous pouvant rattacher une classe d'association à une relation d'association pour fournir des informations supplémentaires sur la relation. Une classe d'association est identique à d'autres classes et peut contenir des opérations, des attributs ainsi que d'autres associations.

Maintenant nous allons représenter la relation classe d’association

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N°** | **Désignation** | **Classes participantes** | **Classe d’association** | **Cardinalités** |
| 1 | Posséde | Commande  Produit | Ligne\_Commande | 1..1  1..\* |

**Tableau 15 :Représentation de la relation classe d'association**

#### Présentation du diagramme des classes

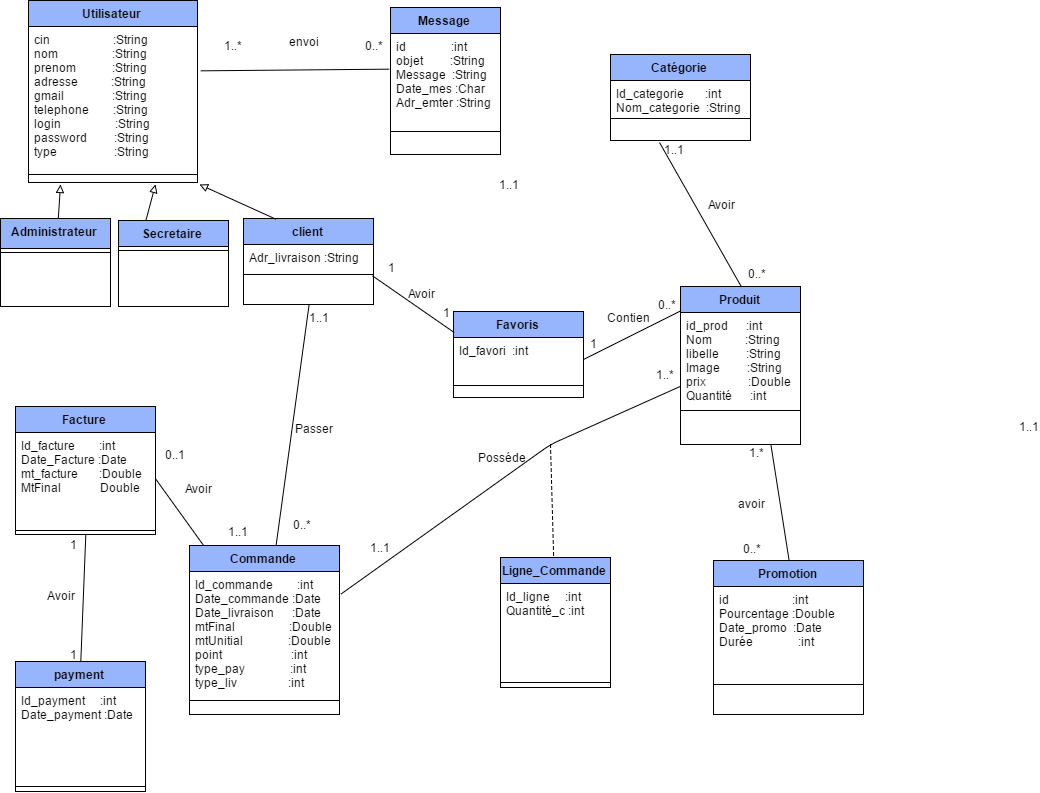


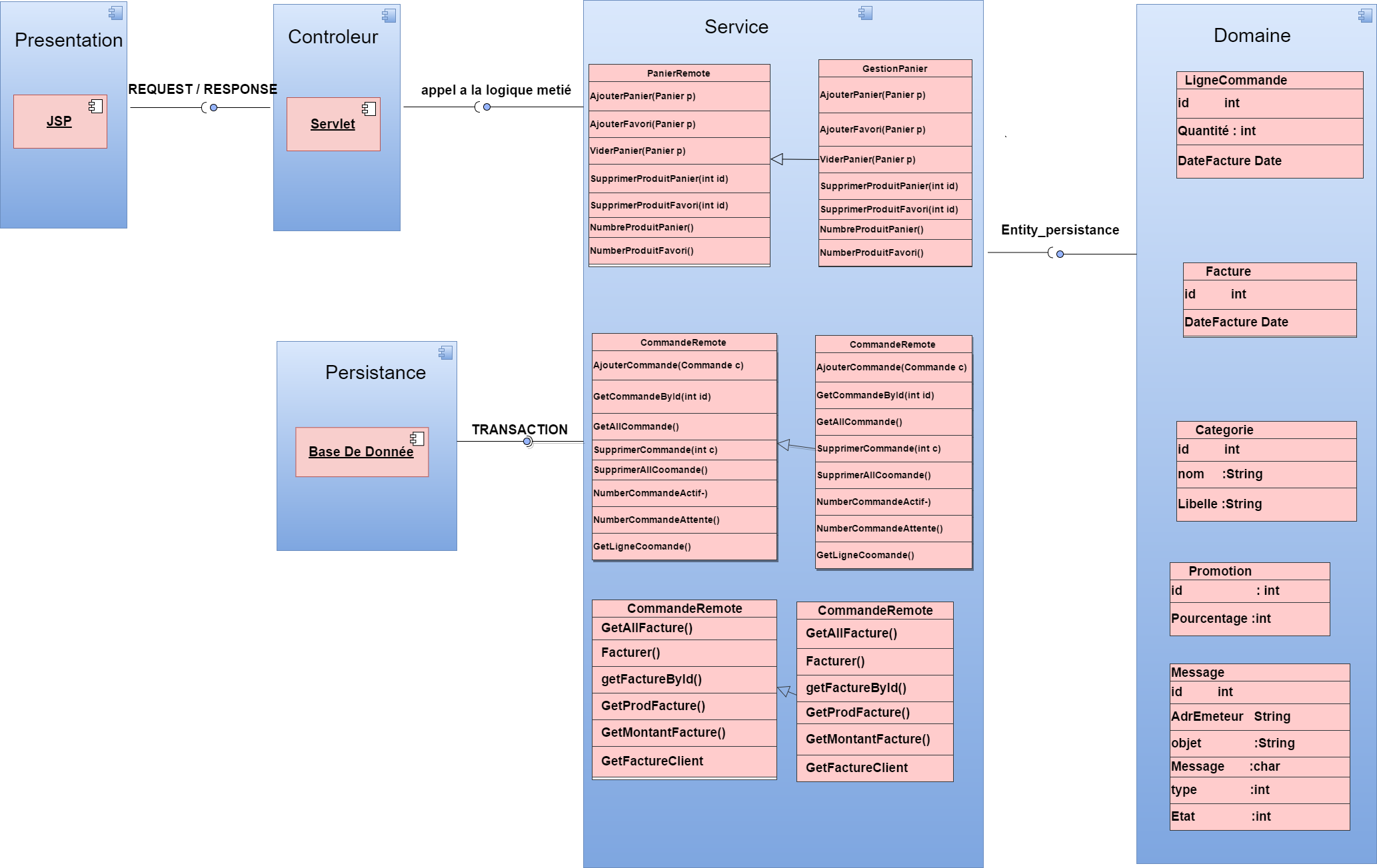
Figure 15 : diagramme des classes

### Diagramme de composants

Le diagramme des composants décrit l'organisation du système du point de vue des éléments logiciels comme les [modules](https://fr.wikipedia.org/wiki/Module_(programmation)) (paquetages, fichiers sources, [bibliothèques](https://fr.wikipedia.org/wiki/Biblioth%C3%A8que_logicielle), exécutables), les données (fichiers, bases de données) ou encore les éléments de configuration (paramètres, scripts, fichiers de commandes). Ce diagramme permet de mettre en évidence les dépendances entre les [composants](https://fr.wikipedia.org/wiki/Composant_logiciel) (qui utilisent quoi). Maintenant nous allons présenter le diagramme de composant qui décrit la relation entre les couches.



**Figure 16 : diagramme de composants partie 1**



**Figure 17 : diagramme de composants partie 2**

La couche services comporte les EJBSession, chaqu’un comporte une classe « Statless » qui implémente une interface « Remote ».

La couche domaine comporte les EJBEntity qui sont reliés avec des tables dans la base de données.

### Services Web

Le service web est une architecture orienté service permettant à des applications de communiquer entre elle directement sans se préoccuper des technologies sur lesquelles elles sont implémentées. Il se présente comme étant la solution pour répondre à l’interopérabilité des systèmes.

La spécificité des Web Services est l'utilisation de HTTP comme support des messages entre clients et serveur.

Il existe plusieurs technologies pour services web :

* Les services web de type [Représentationnel state Transfer](https://fr.wikipedia.org/wiki/Representational_state_transfer) (REST)
* Les services web SOAP

Le principal avantage de ce type est la facilité de mise en œuvre, l'agilité de la conception, et la nature légère. Chaque fois qu’on a besoin de mettre en place un composant rapidement opérationnel avec de bonnes performances et faible surcharge, il est souvent préférable d'utiliser REST.

Les services web sont utilisés pour sérialiser des javaBeans au format JSON, nous présentons dans le tableau suivant les services web qu’on a utilisé dans notre solution

|  |  |
| --- | --- |
| Classe | Description |
| Utilisateur Service | la servlet utilisateurservice est utilisée pour sérialiser les méthodes métier relatifs à l’entité utilisateur, récupérer utilisateur et, authentifié |
| Liste message | Sert à implémenter les méthodes : récupérer la liste des message par client . |
| Liste Produit | Sert à implémenter les méthodes : récupérer la liste des produits. |
| Liste produit Favori | Sert à implémenter les méthodes : récupérer la liste des produits favoris par client. |
| Liste Facture | Sert à implémenter les méthodes : récupérer la liste des facture par client. |
| Ajouter produit favori | Ce servlet permet de gérer l’entité panier (Ajouter produit au liste favori) |
| Envoyer message | Ce servlet permet de gérer l’entité message (envoyer message) |

**Tableau 16 :les services web**

### Partie mobile

#### 6.1. Description des Activités et des fragments

|  |  |
| --- | --- |
| Classe | Description |
| MainActivity | C’est l’Activité principale de l’application, elle nous permet d’invoquer l’appel des autres activités et autres fragments. |
| LoginActivity | cette activité permet au client de s’authentifier |
| NewMessage | cette activité permet au client d’envoyer un nouveau message vers l’administrateur |
| FactureActivity | Cette activité nous permet de consulter les détails d’ une facture |
| ProductActivity | Cette activité nous permet de consulter les détails d’ un produit |
| FactureFragment | Ce fragment comporte la liste des factures du client, elle permet d’invoquer l’appel à l’activité FactureActivity |
| MessageFragment | Ce fragment comporte la liste des messages du client, elle permet d’invoquer l’appel à l’activité New Message |
| ProduitFragment | Ce fragment comporte la liste des produits, il permet de placer un produit dans la liste des favories |
| FavorieFragment | Cette activité comporte la liste des produit favori qui permet de l’ajouter à partir du fragment produit |
| Alarm | Cette classe est un BroadcastReceiver qui sera implémentée pour démarrer le service |
| AlertService | Cette classe est un Service Background qui  Comporte un Thread. Elle invoque l’appel aux Services Web Liste message. Si un message est détecté, alors il s’affiche |
| Services | Cette activité sert à initialiser l’adresse et le port de notre serveur. |

**Tableau 17 : description des classes**

### Diagramme de séquences



#### Définition

Le diagramme de séquence présente des interactions entre les objets selon un point de vue temporel. La représentation se base essentiellement sur l’expression des interactions en insistant sur l’ordre chronologique des envois des messages. Dans ce diagramme, la représentation se concentre sur l’expression des interactions

Nous définissons les stéréotypes de Jacobson dans ce qui suit.

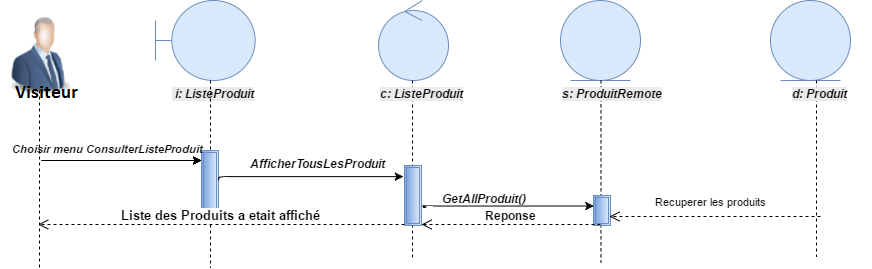
|  |  |
| --- | --- |
| **code** | **description** |
| **i** | **Interface** |
| **c** | **Controller** |
| **s** | **Services** |
| **d** | **domaine** |

**Tableau 18 : stéréotypes de Jacobson pour un diagramme de séquences**

#### Présentation des diagrammes de séquence

Dans ce qui suite nous présentent le diagramme de séquence relatif a quelque cas d’utilisation.

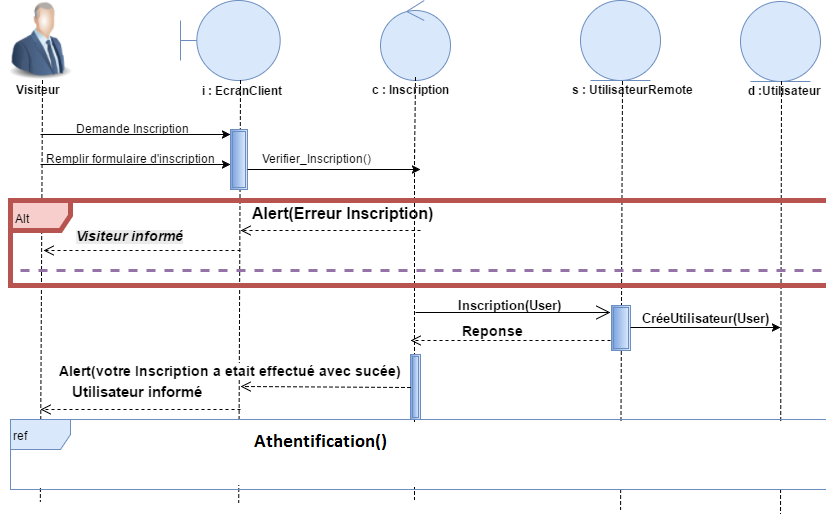
#### Diagramme de séquences « Consulter la liste des produit»



**Figure 18 : Diagramme de Séquences « Consulter la liste des produit»**

#### Diagramme de séquences « Inscription »

Pour bien profiter des privilèges Dédiés aux clients, un visiteur doit d’abords entamer la phase d’inscription avec succès et pour cela il faut qu’il passe par l’ensemble des séquences que nous allons simplifier par le schéma suivant:



**Figure 20 : Diagramme de séquences « Inscription   »**

- Le visiteur demande le formulaire d’inscription.

- Le formulaire s’affiche.

- Le visiteur rempli le formulaire.

- Une vérification de l’existence du client dans la base se lance.

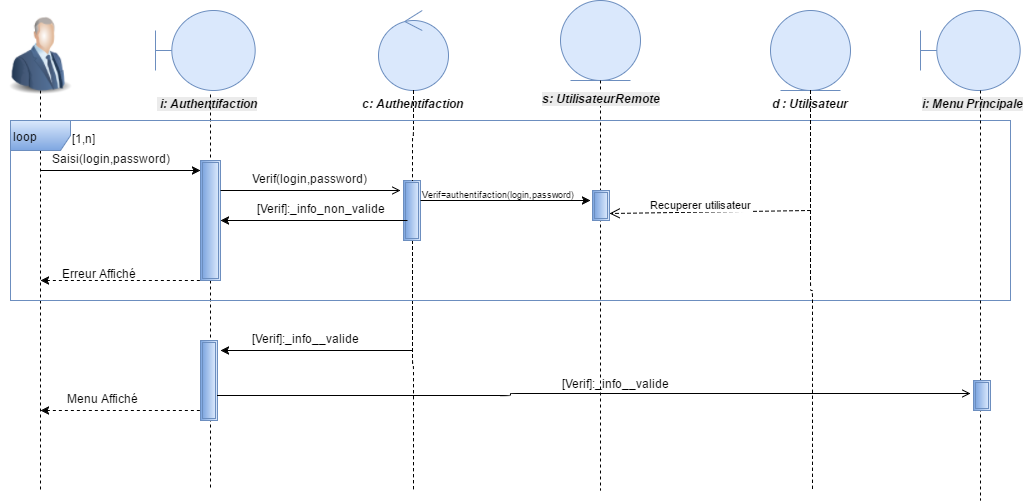
- Si le client existe déjà un message d’erreur s’affiche.

- Si c’est un nouveau client confirmation de l’inscription s’affiche.

#### Diagramme de séquences « Authentification »

Le diagramme de séquence «Authentification» présente le séquencement des interactions entre utilisateur, l’interface d’authentification, le service Utilisateur Remote et l’interface de profil. Dans ce diagramme Loop (1, n) indique qu’il y aura une répétition d’affichage de l’interface d’authentification jusqu’à la validation du pseudo et du mot de passe.

Le schéma suivant présente les séquences à effectuer pour entamer la phase d’authentification.



**Figure 19: Diagramme de Séquences « Authentification »**

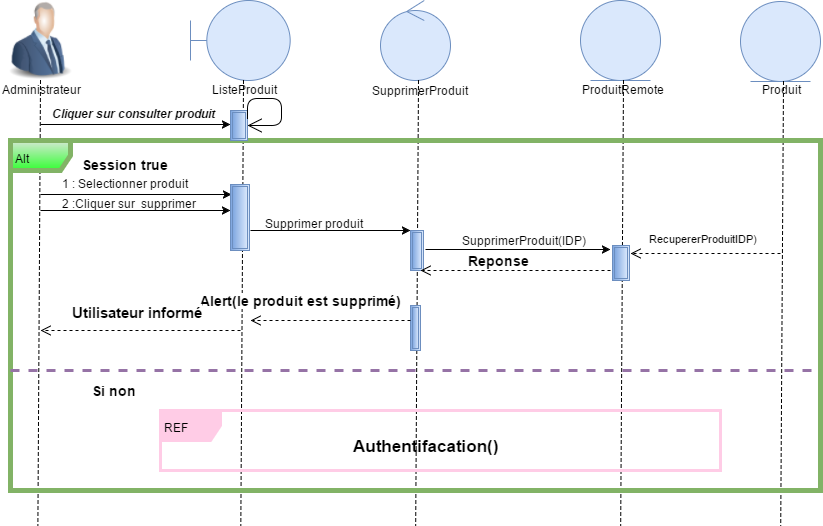
- Le client saisie son login et son mot de passe.

- Une vérification se lance dans la base de données.

- Après un temps de réponse l’authentification se valide ou ne message d’erreur s’affiche

#### Diagramme de séquences « Suppression produit »

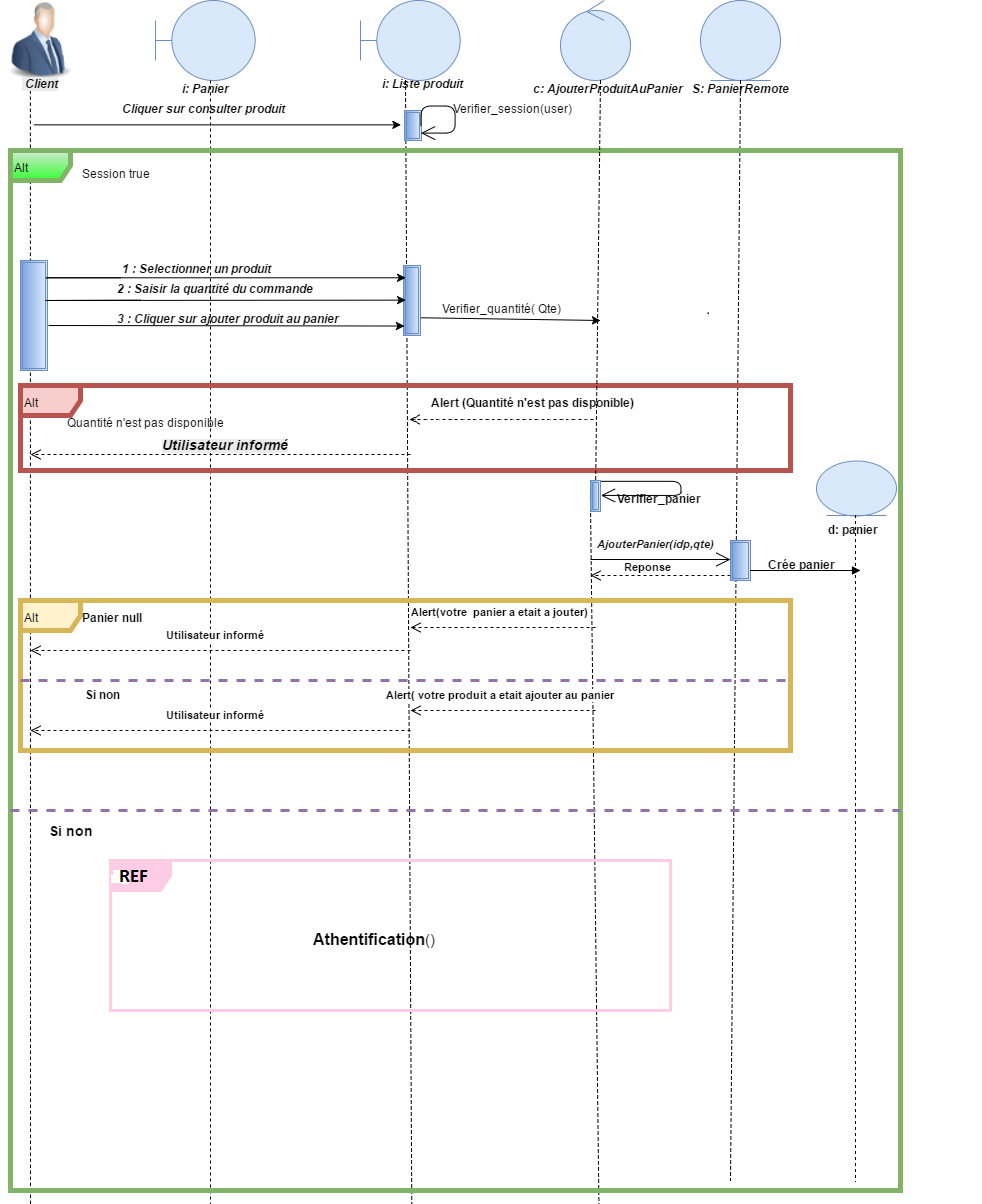
Parmi les scénarios dont l’administrateur prend en charge, nous pouvons mentionner la gestion des produits exposés sur notre site web telles que la consultation, l’ajout, la modification et la suppression que nous allons montrer dans le diagramme de séquence suivant.



**Figure 21 : Diagramme de Séquences « Suppression produit »**

#### Diagramme de séquences « Ajouter produit au panier»

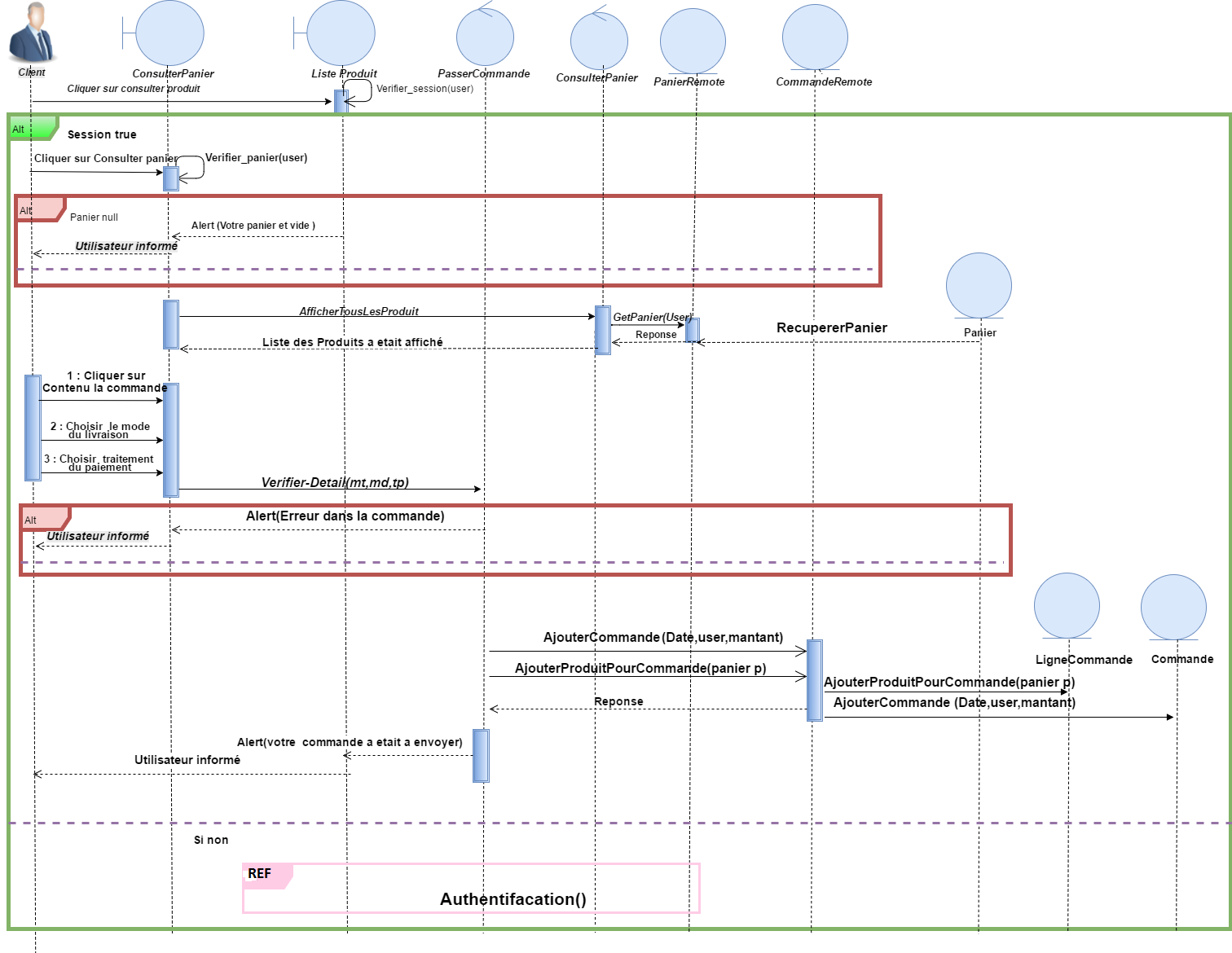
Le diagramme de séquence «Ajouter produit au panier» présente le séquencement des interactions entre le client, l’interface ListeProduit, l’interface Panier, le Controller AjouterProduitAuPanier et le service PanierRemote.

TRUE

**Figure 22 : Diagramme de Séquences « Ajouter produit au panier»**

#### Diagramme de séquences « Passer commande»

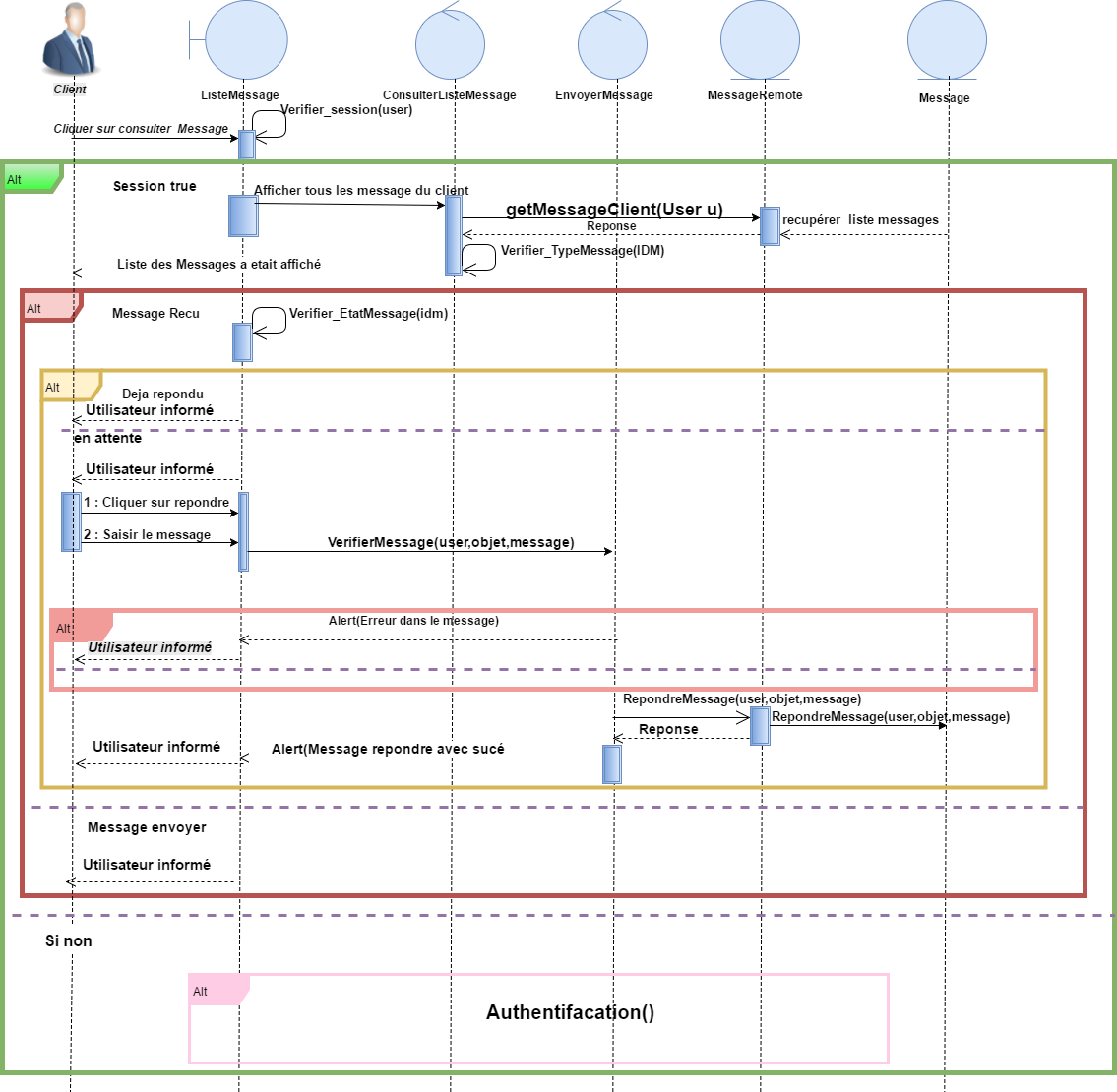
Le diagramme de séquence «Passer commande» présente le séquencement des interactions entre le client, l’interface ListeProduit, l’interface Panier, les Controller PasserCommande et ConsulterMonPanier et les deux services PanierRemote et CommandeRemote.



**Figure 23 : Diagramme de Séquences « Passer commande »**

#### Diagramme de séquences « Gestion message»

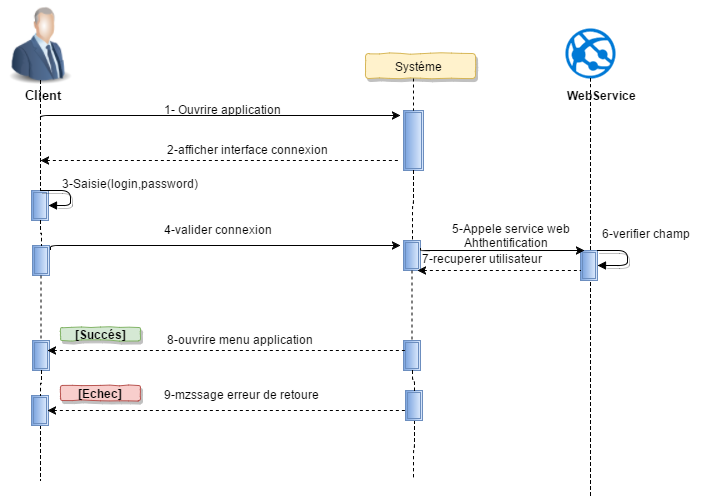
Le diagramme de séquence «Gestion message» présente le séquencement des interactions entre le client, l’interface ListeMessage, l’interface Message, les 3 Controller ConsulterListeMessage et EnvoyerMessage et RepondreMessage et le service MessageRemote.



**Figure 24: Diagramme de Séquences « Gestion message»**

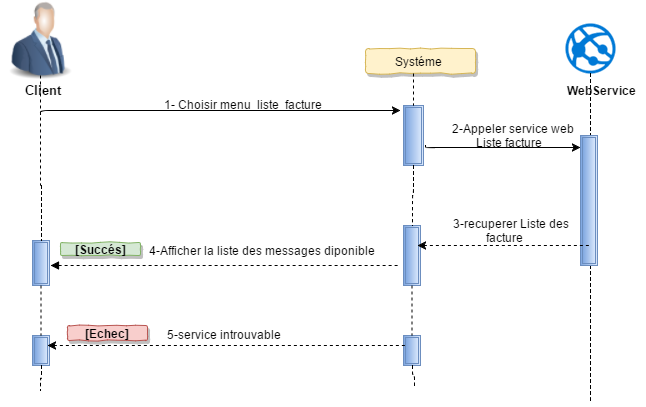
#### 7.3. Les Diagramme de séquence partie mobile

#### 7.3.1. Diagramme de séquences <<Authentification>>

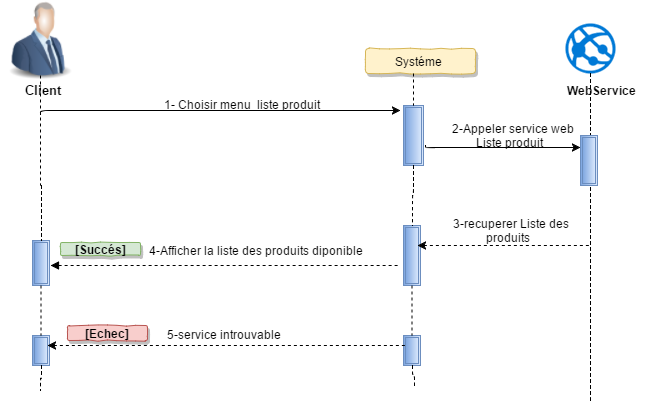


**Figure 25 : Diagramme de Séquences « Authentification»**

#### 7.3.2. Diagramme de séquences <<Consulter Liste facture>>

**Figure 26 : Diagramme de Séquences « Consulter Liste facture»**

#### 7.3.4 Diagramme de séquences <<Consulter liste produit>>



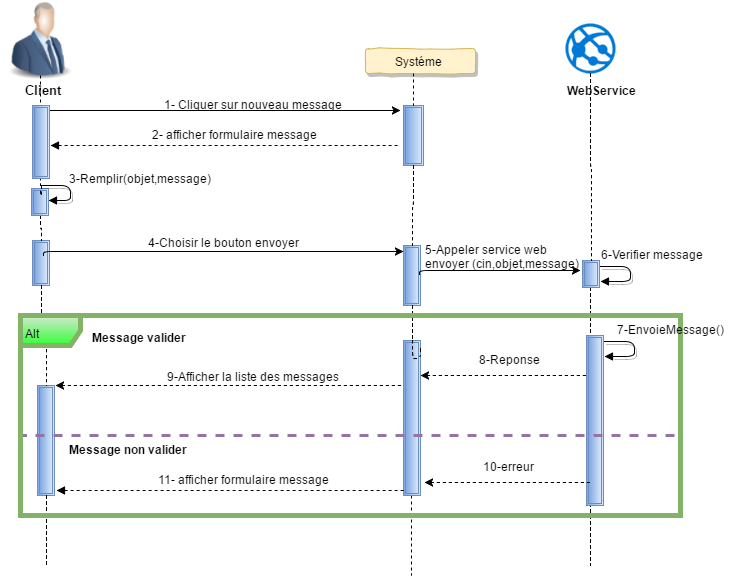
**Figure 27 : Diagramme de Séquences « Consulter liste produit»**

#### 7. 3.5 Diagramme de séquences <<Consulter liste message>>



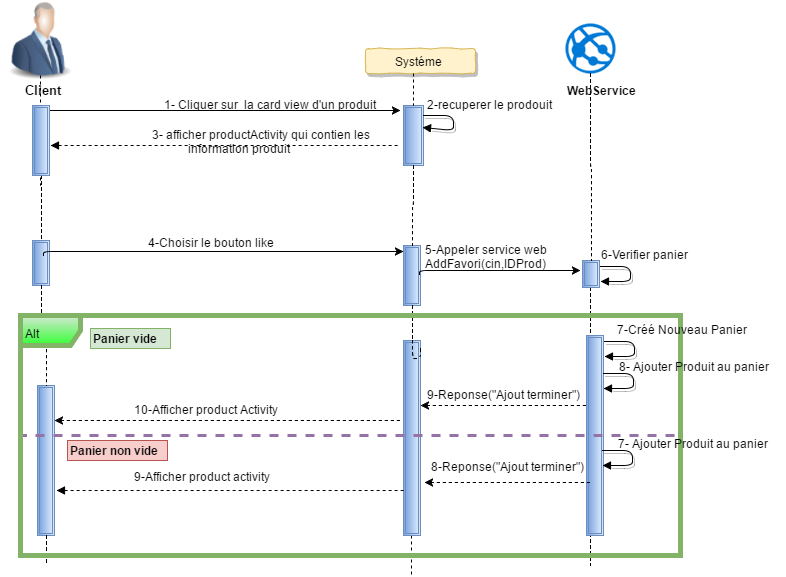
**Figure 28 : Diagramme de Séquences « Consulter liste message»**

#### 7. 3.6 Diagramme de séquences <<Envoyer message>>



**Figure 29: Diagramme de Séquences « Envoyer message»**

#### 7.3.7 Diagramme de séquences <<Ajouter produit au favori>>



**Figure 30: Diagramme de Séquences « Ajouter produit au favori»**

### Diagramme d’activités.



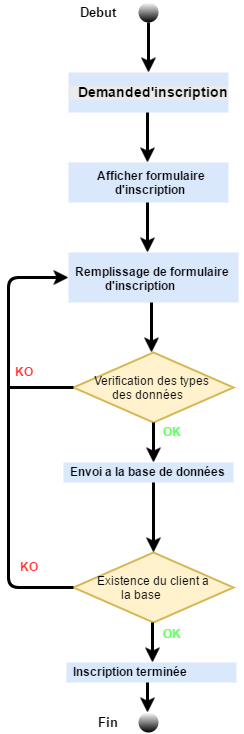
#### Définition

Les diagrammes d’activités permettent de mettre l’accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de représenter graphiquement le comportement d’une méthode ou le déroulement d’un cas d’utilisation.

**8.2. Les diagrammes d’activité de notre site web**

**a. Diagramme d’activité inscription client**

La phase d’inscription est indispensable pour passer d’un simple visiteur du site qui n’a le droit que de consulter les produits et leurs prix à un client qui peut acheter ses articles désirés et payer sa facture et donc attendre la livraison de sa commande à domicile.



**Figure 31 : *diagrammes d’activité<<Inscription>>***

- Le visiteur demande l’inscription.

- Le formulaire d’inscription s’affiche sur l’écran.

- Le visiteur remplit les champs demandé dans le formulaire.

- Le système vérifie les données entrées.

- Si les données sont acceptées, le système les envoie à la base si non, il revient à l’étape précédente.

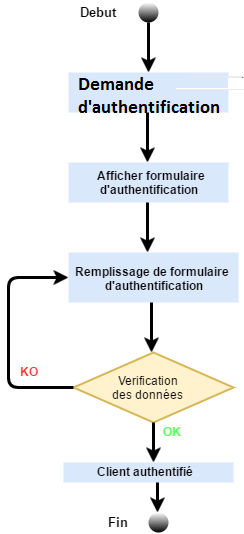
- Le serveur vérifie l’existence du client dans la base.

- Si le client existe déjà, un message d’erreur s’affiche.

- Si le client n’existe pas, l’inscription se termine avec succès.

**b. Diagramme d’activité authentification**

L'authentification est la procédure qui consiste, pour un système informatique, à vérifier l'identité d'une entité (personne, ordinateur etc.) afin d'autoriser son accès aux systèmes, réseaux, applications. Elle permet donc de valider l'authenticité de l'entité en question.



**Figure 32 : *diagrammes d’activité<<Inscription>>***

- Le client demande l’authentification en cliquant sur le bouton login.

- Le formulaire d’authentification s’affiche sur l’écran.

- Le client entre son nom d’utilisateur et son mot de passe.

- Le système vérifie les paramètres du client sur la base.

- La conformation du succès ou échec est envoyée au client.

**c. Diagramme d’activité gestion des articles**

Un webmaster est l'administrateur du site. Il a pour but de s'assurer de la fiabilité de ses services proposés, ainsi que leur audimat et rentabilité. Parmi ces services nous pouvons citer

- La gestion des produits.

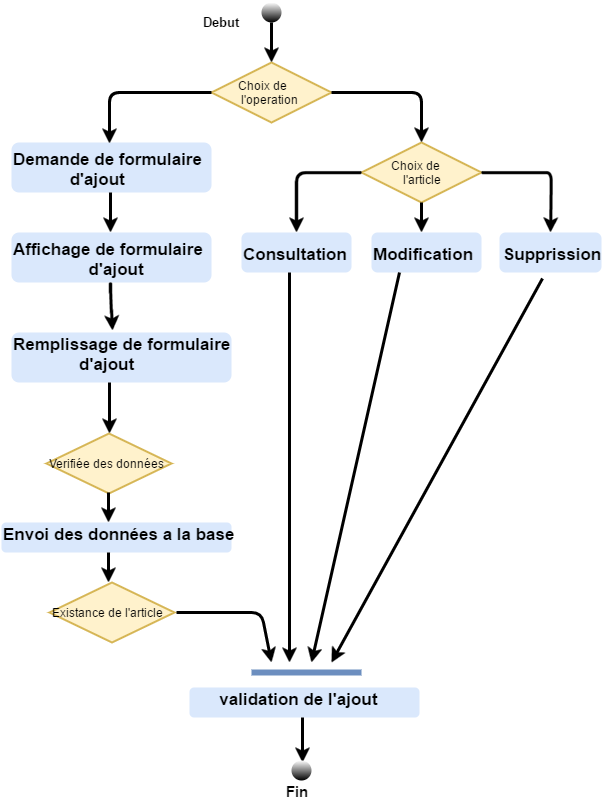
- La gestion des comptes utilisateurs.

- La gestion de la liste des catégories.

- La gestion de la liste des commandes.

- La gestion de la liste des messages.

- La gestion des listes des marques, des produits et beaucoup d’autres activités.



**Figure 33 : *diagrammes d’activité<<Gestion des articles>>***

- L’administrateur précise l’opération à appliquer sur l’article.

- S’il s’agit d’une opération d’ajout, l’administrateur demande le formulaire d’ajout des nouveaux articles.

- Ce formulaire s’affiche.

- L’administrateur saisit les données relatives à l’article concerné.

- Vérification de la validité des données saisies.

- En cas de validité, les données prennent chemin vers la base de données.

- Une deuxième vérification, en ce qui concerne l’existence de l’article dans notre base.

- Si non les données seront validées.

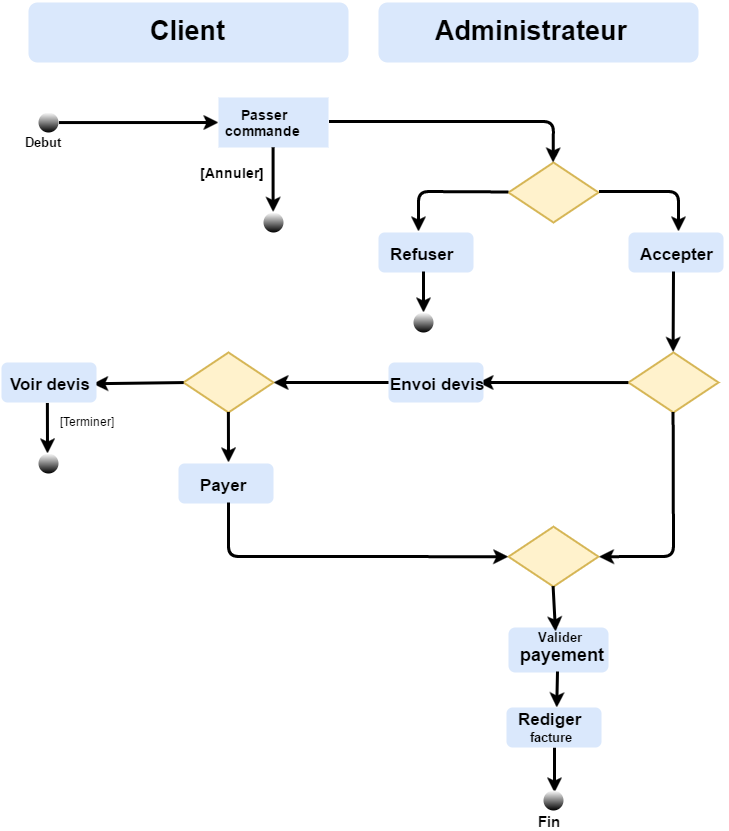
- Maintenant, si l’opération désirée et de gérer un article déjà existant dans la base, la sélection de cette article est la première étape.

- Choix du type de gestion qui peut être consultation, modification ou bien suppression.

- Et enfin la validation de l’opération.

**D. Diagramme d’activité gestion des commandes**

Dans ce qui suit, nous présentons le diagramme d’activité permettant de décrire le processus «Passage de la commande».



**Figure 34 : Diagramme d'activité du processus de passage de la commande**

- Le client précise l’opération « Passer commande ».

- L’administrateur à le choix d’accepter ou refuser la commande.

Si l’administrateur accepte la Commande

- L’administrateur précise l’opération «Rédige devis ».

- Le client à le choix de payer la commande en ligne.

- L’administrateur précise l’opération «Valider paiement ».

- L’administrateur précise l’opération «Rédiger facture ».

### 9. Diagramme d’états-transitions



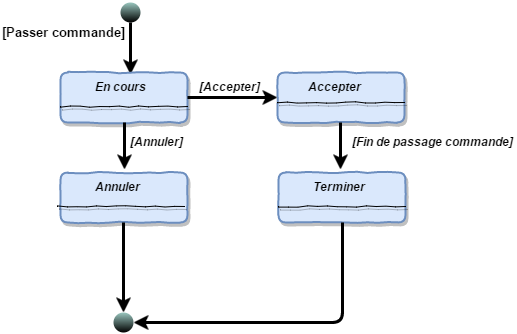
#### 9.1. Définition

Le diagramme d'états-transitions d'UML décrit le comportement interne d'un objet à l'aide d'un automate à états finis. Il présente les séquences possibles d'états et d'actions qu'une instance de classe peut traiter au cours de son cycle de vie en réaction à des événements discrets (de type signaux, invocations de méthode)...

Lorsque les scénarios ont été formalisés, la connaissance de l’ensemble des interactions entre les objets permet de représenter les règles de gestion dynamique du système. Il faut cependant se focaliser sur les classes aux comportements les plus riches afin de développer certaines de ces règles dynamiques. Dans ce cas on peut utiliser le concept de machine à état finis, le concept s’intéresse au cycle de vie d’un objet générique d’une classe particulière au fil de ses interactions avec l’extérieur, dans tous les cas possibles. Cette vue locale d’un objet, décrivant comment il réagit à des événements en fonction de son état courant et passe dans un nouvel état, est représentée graphiquement sous forme d’un diagramme d’états.

#### 9.2. Présentation du diagramme d’états-transitions

Le diagramme ci-dessous correspond aux différents états relatifs à l’objet « Commande »



**Figure 35 : Diagramme d'états-transitions de la Commande**

# CHAPITRE 5 : Réalisation

## Introduction

La réalisation du projet est la dernière étape à élaborer pour la mise en place d’une application informatique. C’est à partir de cette étape que nous obtiendrons un produit final qui fonctionne et qui peut évoluer. Notre application de vente et achat en ligne est destinée à la société NEJMA et aussi à leurs clients, donc elle doit être de qualité, performante et simple à utiliser.

Dans ce chapitre nous allons suivre étape par étape les phases nécessaires à la réalisation de notre application.

Notons que le succès de l’application n’est pas lié uniquement à sa bonne conception mais plutôt, et surtout au bon choix des outils logiciels qui est la clé de la réussite du produit final.

## Environnement de réalisation



### Environnement matériel

Pour concrétiser notre application, nous avons utilisé une P.C et un Smartphone ayant les caractéristiques suivantes:

|  |  |  |
| --- | --- | --- |
| **Marque** | **ASUS** | **Samsung** |
| **Processeur** | Intel ® Core ™ i7-5500U CPU @ 2.40 GHz | Dual Core 1.3 GHz |
| **RAM** | 8 GB | 1 GB |
| **Disque Dur** | 1 Tera | 4 GB |
| **Système d’exploitation** | Windows 10 64Bits | Android 5 lollipop |

**Tableau 19 :Caractéristiques des outils utilisés**

### Outils de développement

Les principaux outils qui ont contribué à la qualité du développement sont :

* **PowerAMC**

PowerAMC : est un logiciel de conception créé par la société SDP, qui permet de modéliser les traitements informatiques et leurs bases associées.

Nous avons utilisé le PowerAMC pour présenter les diagrammes de classe et les diagrammes de séquences

* **Eclipse JEE**

C’est un projet de la Fondation Eclipse visant à développer tout un environnement de développement libre, extensible, universel et polyvalent. Son objectif est de produire et fournir divers outils gravitant autour de la réalisation de logiciel, englobant les activités de codage logiciel proprement dites (avec notamment un environnement de développement intégré) mais aussi de modélisation, de conception, de test, de reporting, etc. Son environnement de développement notamment vise à la généricité pour lui permettre de supporter n'importe quel langage de programmation.

* **JBOSS**

Un serveur d'applications Java EE Libre écrit en Java, publié sous licence GNU LGPL. Étant écrit en Java, WildFly 7 peut être utilisé sur tout système d'exploitation fournissant une machine virtuelle Java (JVM). Le nom JBoss est aujourd'hui utilisé pour JBoss EAP, produit dérivé WildFly et faisant l'objet d'un support commercial.

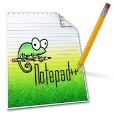
* **Android Studio**

Android Studio est un environnement de développement pour développer des applications Android. Il est basé sur IntelliJ IDEA. Permet principalement d'éditer les fichiers Java et les fichiers de configuration d'une application Android.

* **XAMMP**

Xammp est un ensemble de [logiciels](https://fr.wikipedia.org/wiki/Logiciel) permettant de mettre en place facilement un serveur web local, un [serveur FTP](https://fr.wikipedia.org/wiki/Serveur_FTP) et un [serveur de messagerie électronique](https://fr.wikipedia.org/wiki/Serveur_de_messagerie_%C3%A9lectronique)

C’est un outil visuel unifié pour les architectes de bases de données, les développeurs et les DBA, il est orienté Pour le développement de SQL et des outils d'administration complets pour la configuration du serveur, la gestion des utilisateurs.

* **Notepad++**

Notepad++ : est un éditeur de code source qui prend en charge plusieurs langages. Il nous a aidé dans le développement d’applications web.

* **Adobe Photoshop CS6**

Photoshop est un logiciel de retouche, de traitement et de dessin assisté par ordinateur édité par Adobe. Il est principalement utilisé pour le traitement de photographies numériques, mais sert également à la création d’images.

### Environnements de développement

* **JAVA**

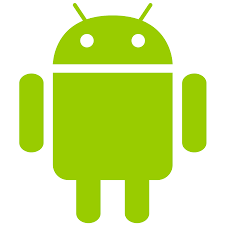
Java est un langage de programmation informatique orienté objet. Java permet de développer des applications client-serveur. C’est surtout côté serveur que Java s’est imposé dans le milieu de l’entreprise grâce aux servlets, et aux JSP (Java Server Pages).

* **JSP**

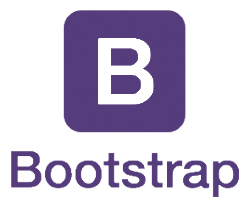
 Les JSP (Java Server Pages) sont une technologie Java qui permet la génération de pages web dynamiques. La technologie JSP permet de séparer la présentation sous forme de code HTML et les traitements écrits en Java sous la forme de JavaBeans ou de servlets (voir l’annexe).

* **EJB**

Enterprise JavaBeans (EJB) est une architecture de composants logiciels côté serveur pour la plateforme de développement Java EE (voir l’annexe).

* **Android**

Android, prononcé androïde, est un [système d'exploitation mobile](https://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27exploitation_mobile) basé sur le [noyau Linux](https://fr.wikipedia.org/wiki/Noyau_Linux) et développé actuellement par[Google](https://fr.wikipedia.org/wiki/Google). Le système a d'abord été conçu pour les [smartphones](https://fr.wikipedia.org/wiki/Smartphone) et [tablettes tactiles](https://fr.wikipedia.org/wiki/Tablettes_tactiles), puis s'est diversifié dans les objets connectés et ordinateurs comme les télévisions ([Android TV](https://fr.wikipedia.org/wiki/Android_TV)), les voitures ([Android Auto](https://fr.wikipedia.org/wiki/Android_Auto)), les ordinateurs ([Android-x86](https://fr.wikipedia.org/wiki/Android-x86)) et les [smart Watch](https://fr.wikipedia.org/wiki/Smartwatch)([Android Wear](https://fr.wikipedia.org/wiki/Android_Wear)). (Voir l’annexe)

* **Bootstrap**

Bootstrap est le Framework HTML, CSS, JS le plus populaire pour le développement des sites web adaptatifs et d’application web. Le Framework fournit également nombres d'éléments graphiques au format standardisé : boutons, libellés, icônes, miniatures, barres de progression, etc.

* **JSON** (JavaScript Object Notation)

JSON (JavaScript Object Notation) est un [format de données](https://fr.wikipedia.org/wiki/Format_de_donn%C3%A9es) textuelles dérivé

de la notation des [objets](https://fr.wikipedia.org/wiki/Objet_(informatique)) du langage [JavaScript](https://fr.wikipedia.org/wiki/JavaScript). Il permet de représenter de l’information structurée comme le permet [XML](https://fr.wikipedia.org/wiki/Extensible_Markup_Language) par exemple (voir l’annexe).

* **JavaScript**

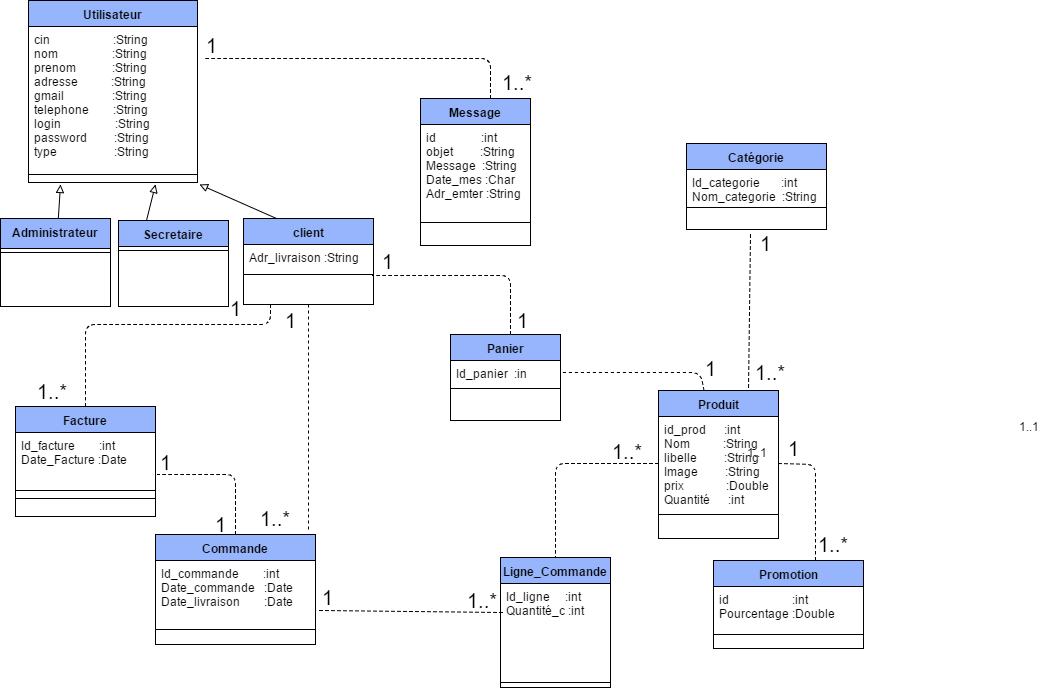
JavaScript est un langage de script orienté objet principalement utilisé dans les pages HTML. A l'opposé des langages serveurs (qui s'exécutent sur le site), JavaScript est exécuté sur l'ordinateur de l'internaute par le navigateur lui-même.

* **MySQL**

MySQL est un système degestion de bases de données relationnelles dédiées Open source. Il est très rapide, fiable etfacile à utiliser et gratuit.il a été développé à l’origine pour gérer des très grandes bases dedonnées beaucoup plus rapidement que des solutions déjà établies. Il offre un ensemble defonctionnalités large et riche. Sa rapidité et sa sécurisation en font un outil idéal pour lesapplications internet.

## Schéma logique des données

Au niveau de cette partie, on traite les passages des modèles objets au modèle relationnel de la base de donnés.

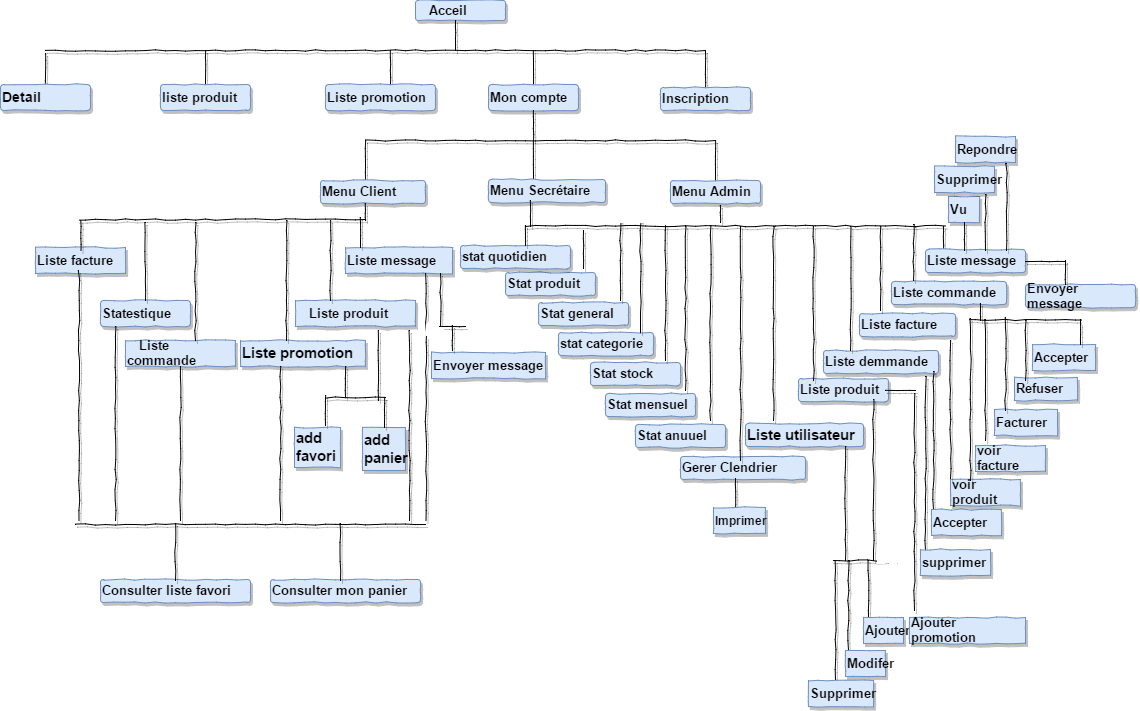


**Figure 36 : schéma logique de données**

## Les modèles navigationnels

Les modèles navigationnels (figure 41) servent à montrer la manière de passage entre les différentes pages de notre application web. Ils reflètent en fait sa structure, les menus principaux ainsi que les sous menus.

**Figure 37 : les modèles navigationnels**



## Les interfaces de l’application

En théorie, une interface doit être naturelle, efficace, fiable, de compréhension et d’utilisation facile. Pour cela nous essayons de respecter les contraintes IHM (interface homme machine).Vu le nombre élevé de ces interfaces, nous essayons de donner les plus importantes qui donnent un aperçu fidèle à l’esprit de l’application.

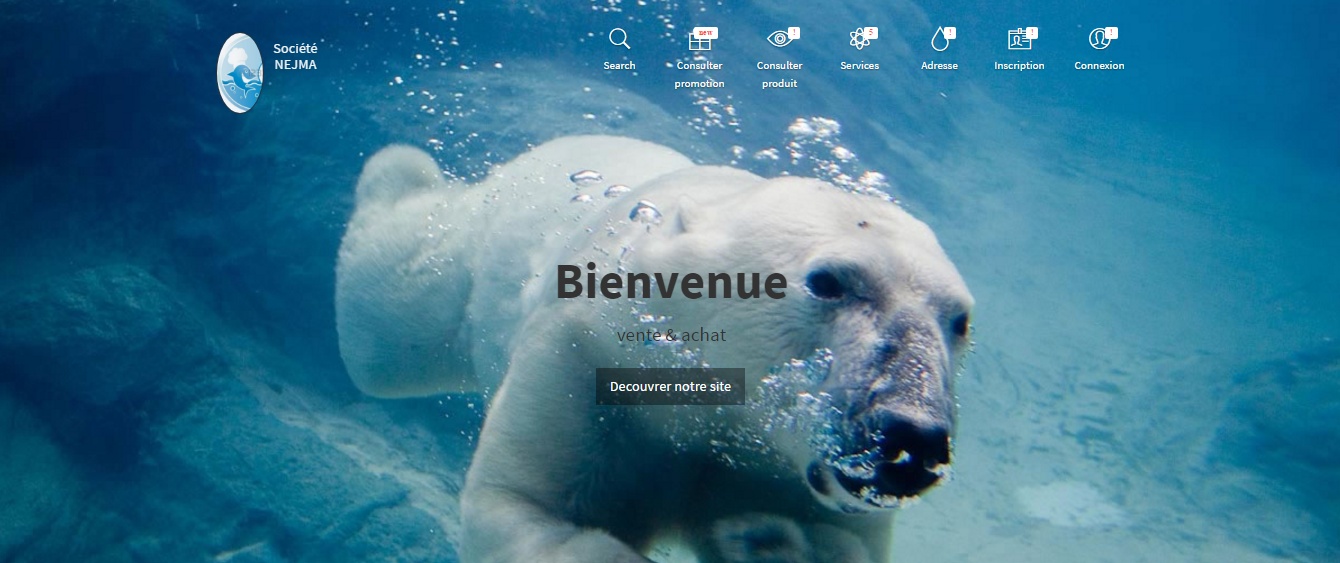


### Interfaces de l’application web



#### Interface « accueil »

Pour la partie web l’interface accueil ce devise en 4 partis :

* Interface générale.
* Zone publicitaire.
* Zone service.
* Zone map défini l’adresse de la société.

**Figure 38 : interface « accueil >>**

En plus l’interface « Accueil» est composée par ensemble de lien :

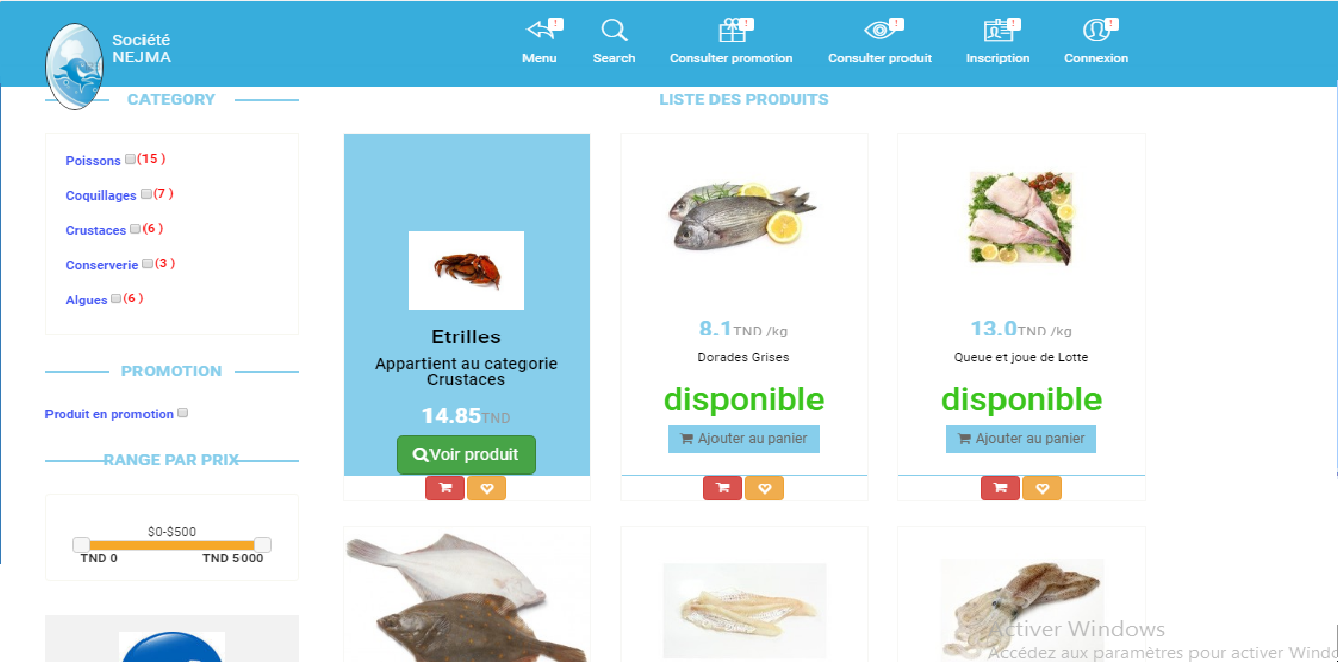
* + Liste produit : amène le client vers la liste des produits qui existent dans la base des données.
  + Liste produit promotion : amène le client vers la liste des produits en promotion qui existent dans la base des données.

L’interface aussi est composée par un ensemble de liens :

* + Consulter produits : qui amène le client pour consulter les produits

#### Interface « liste des produits  »

Avec cette interface on peut avoir la liste des produits qui existent dans la base des données. Le visiteur peut effectuer des recherches personnalisées à l’aide des filtres en fonction de plusieurs critères comme catégorie, prix, promotion etc., puis il sélectionner un produit ensuite il accéder aux détails du produit comme indiqué dans les figures ci-dessous



**Figure 39 : interface « Liste produit »**

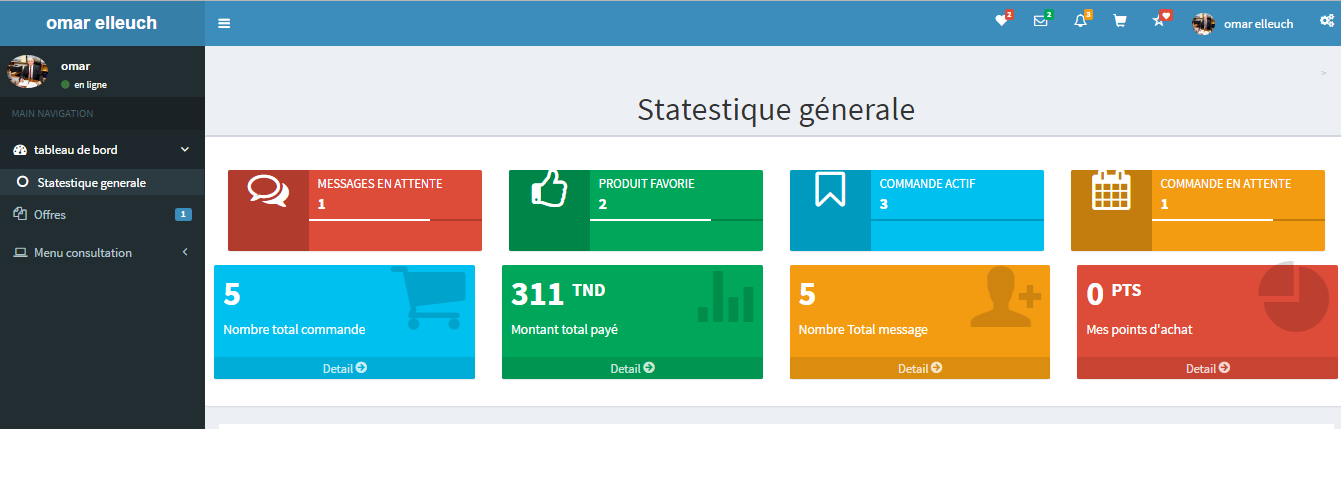
#### Interface « Menu client »

Avec cette interface, le client peut avoir des statistiques comme le nombre des commande réalisés, le nombre des messages, le nombre des produits favori, les points d’achat etc.

En plus que ça, cette interface possède une courbe qui est structurée au tour des commandes effectué au cours de l’année actuelle.

En plus l’interface « Menu client »est composée par ensemble de lien :

* + Liste produit : amène le client vers la liste des produits qui existent dans la base des données.
  + Liste des produits en promotion : amène le client vers la liste des produits en promotion qui existent dans la base des données.
  + Consulter panier : amène le client vers son panier enregistré dans la base des données.
  + Consulter liste favori : amène le client vers la liste favori enregistrée dans la base des données.
  + Liste message : amène le client vers la liste des messages qui existent dans la base des données.

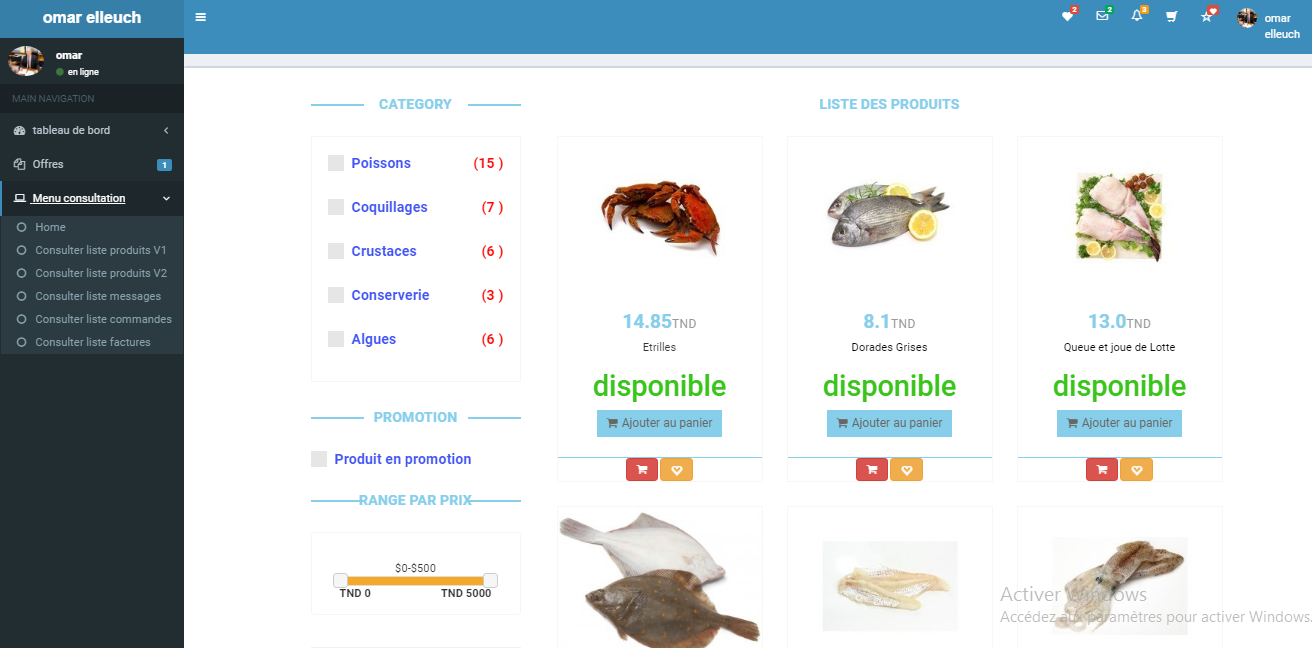


**Figure 40 : interface « menu client»**

#### Interface « Liste produit »

Avec cette interface on peut avoir la liste des produits qui existent dans la base des données. Le client peut effectuer des recherches personnalisés à l’aide des filtres en fonction de plusieurs critères telle que la catégorie, le prix, la promotion etc. Puis il sélectionne un produit et accède aux détails du produit comme indiqué dans les figures ci-dessous

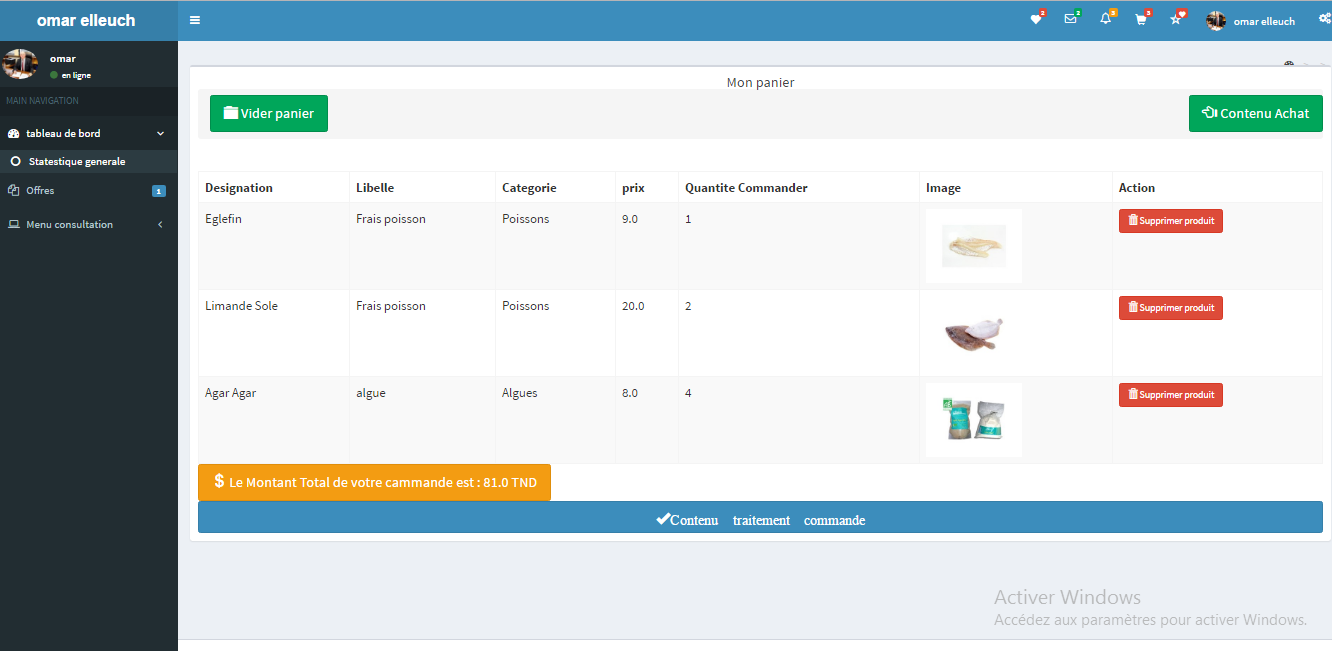
Le model « détail » permet au client de consulter les détails du produit et d’ajouter le produit soit à la liste des favoris soit à son panier.

 **Figure 41 : interface « Liste produit»**

#### Interface « Panier »

Avec cette interface on peut avoir le contenu du panier qui existe dans la session du client. Comme montre les figures ci-dessous, le client consulte son panier avec le prix du shoping et il a le choix de vider le panier ou supprimer un produit du panier.

Le model « Passer Commande » permet au client de remplir les informations de la commande puis de passer cette commande.



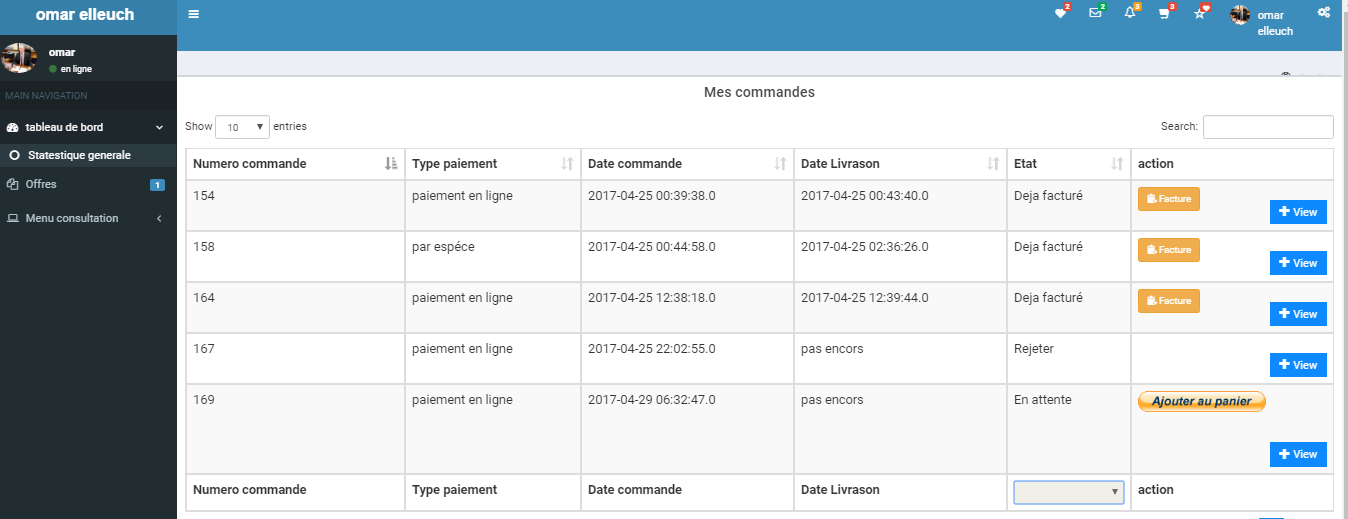
**Figure 42 : interface « panier»**

#### Interface « Liste commande »

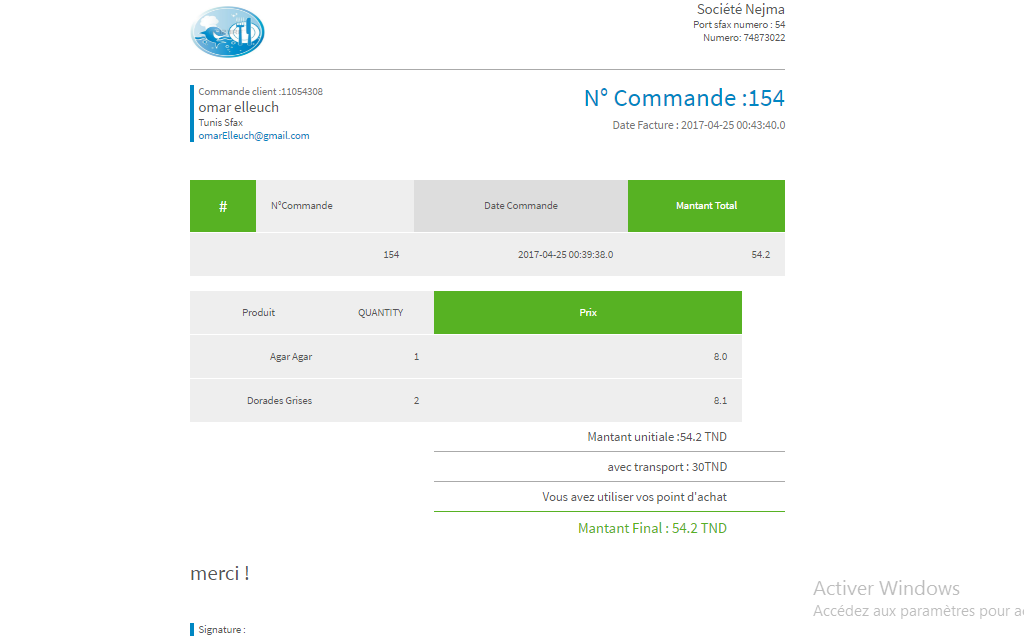
Avec cette interface on peut avoir la liste des commandes de tous les types qui existent dans la base des données. Le client peut effectuer des recherches personnalises à l’aide des filtres en fonction de plusieurs critères état, type etc. Comme montre les figures ci-dessous

L’interface est composé par un ensemble de liens :

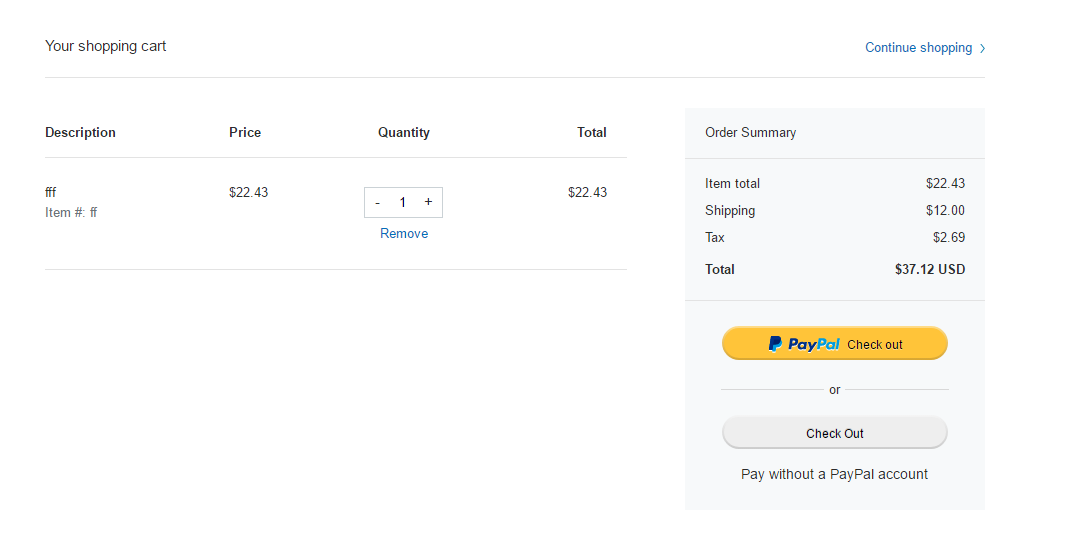
* + Facture : voir la facture d’une commande sélectionnée.(seulement si la commande et activé)
  + View : ce lien amène le client vers le détail de la commande
  + Ajouter panier : ce lien amène le client vers le site officiel du Paypal pour payer sa commande.(seulement si le type de paiement et en ligne)



**Figure 43 : interface « Liste commande»**



**Figure 44 : interface « Facture»**



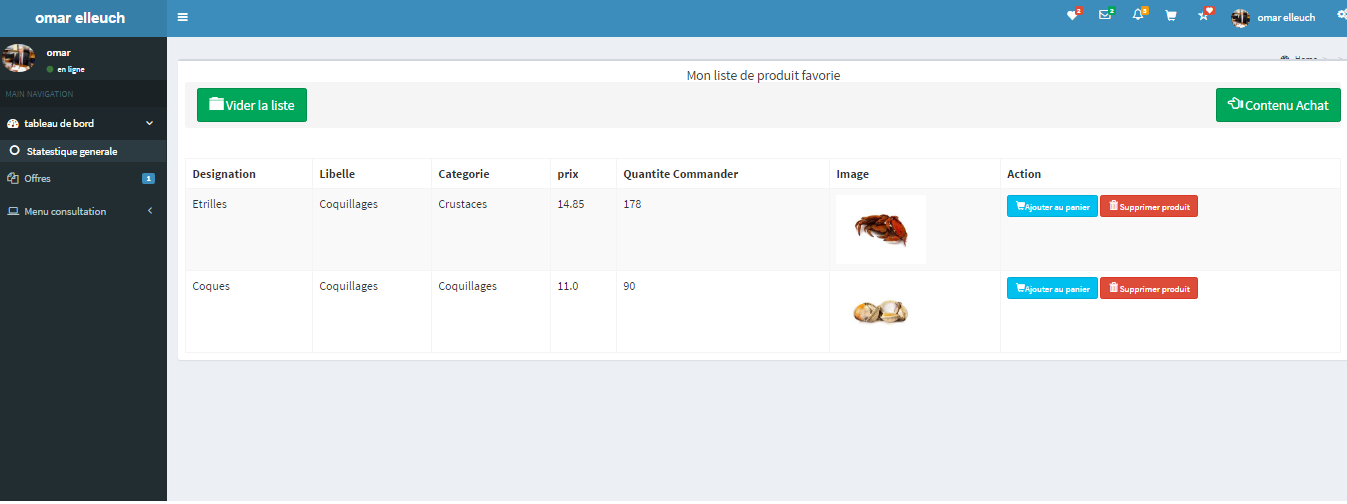
**Figure 45: interface « Panier paypal»**

#### Interface « Liste favori »

Avec cette interface on peut avoir la liste des favoris client enregistrés dans la base des données. Le client peut effectuer différentes actions sur la liste

* Vider la liste
* Contenu l’achat : ce lien amène le client vers la liste des produits.
* Supprimer produit de la liste.
* Ajouter panier : c’est model permet d’ajouter le produit au panier.

Comme montre dans les figures sous dessous



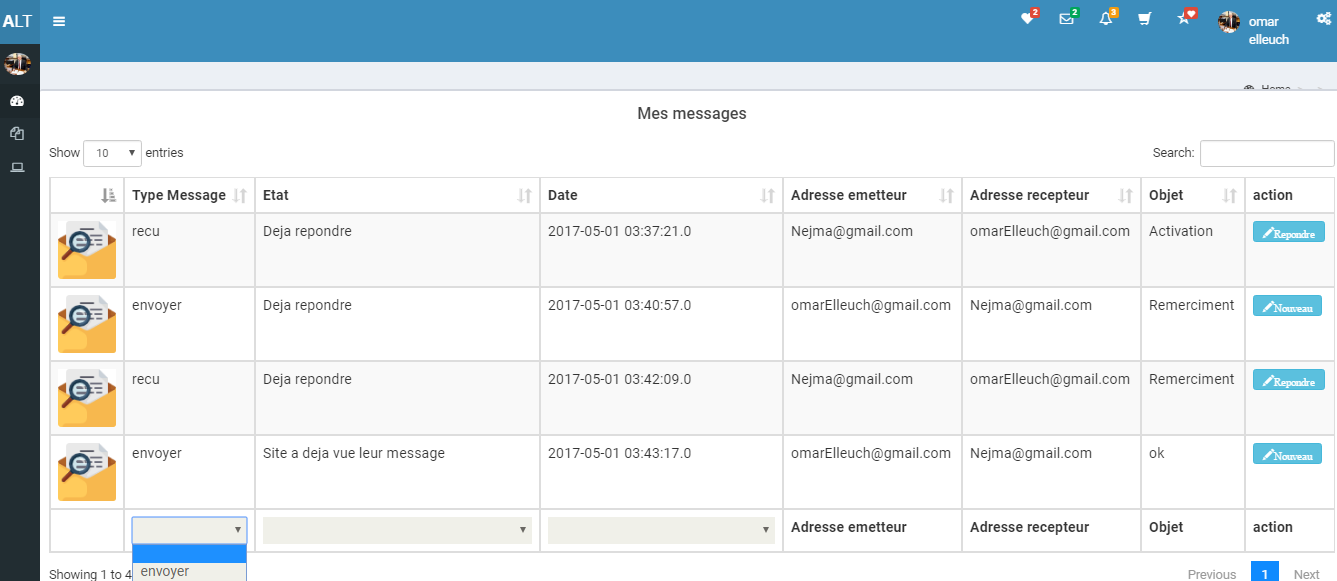
**Figure 46 : interface « Liste produit favori»**

#### Interface « Liste message »

Avec cette interface on peut avoir la liste des messages du client qui existent dans la base des données. Le client peut effectuer des recherches personnalisés à l’aide des filtres en fonction de plusieurs critères état, type, date etc. Comme montre les figures ci-dessous

En plus, le client peut effectuer les déférentes actions suivantes :

* + Nouveau : dans ce model le client va Traiter un message et l’envoyer vers l’administrateur
  + Répondre : dans ce model le client va répondre au message de l’administrateur il va Traiter un message et l’envoyer vers administrateur
  + Voir : ce model permet au client de voir le détail du message.



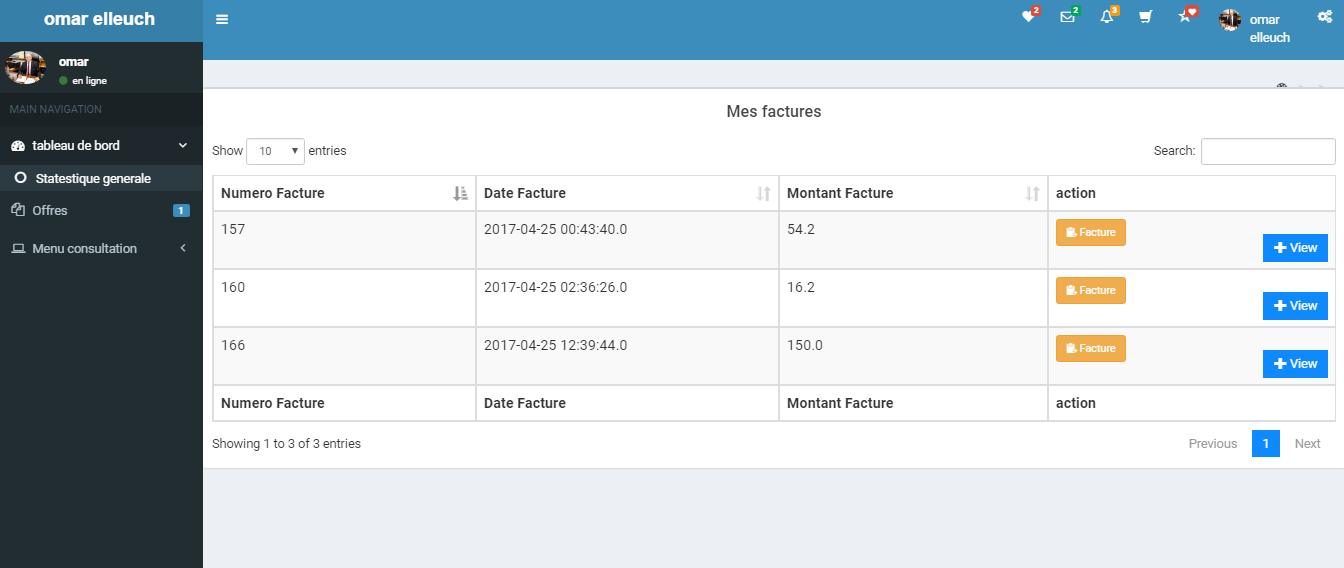
**Figure 47: interface « Liste message client»**

#### Interface « Liste facture »

Avec cette interface le client peut avoir la liste des Factures qui existent dans la base des données. Le client peut effectuer des recherches personnalisées à l’aide des filtres en fonction de plusieurs critères état, type, date etc. Comme montre les figures ci-dessous

Suivant, l’interface est composée par ensemble de liens :

* + Facture : voir la facture sélectionnée.
  + View : ce lien amène le client vers le détail de la commande



**Figure 48: interface « Liste commande»**

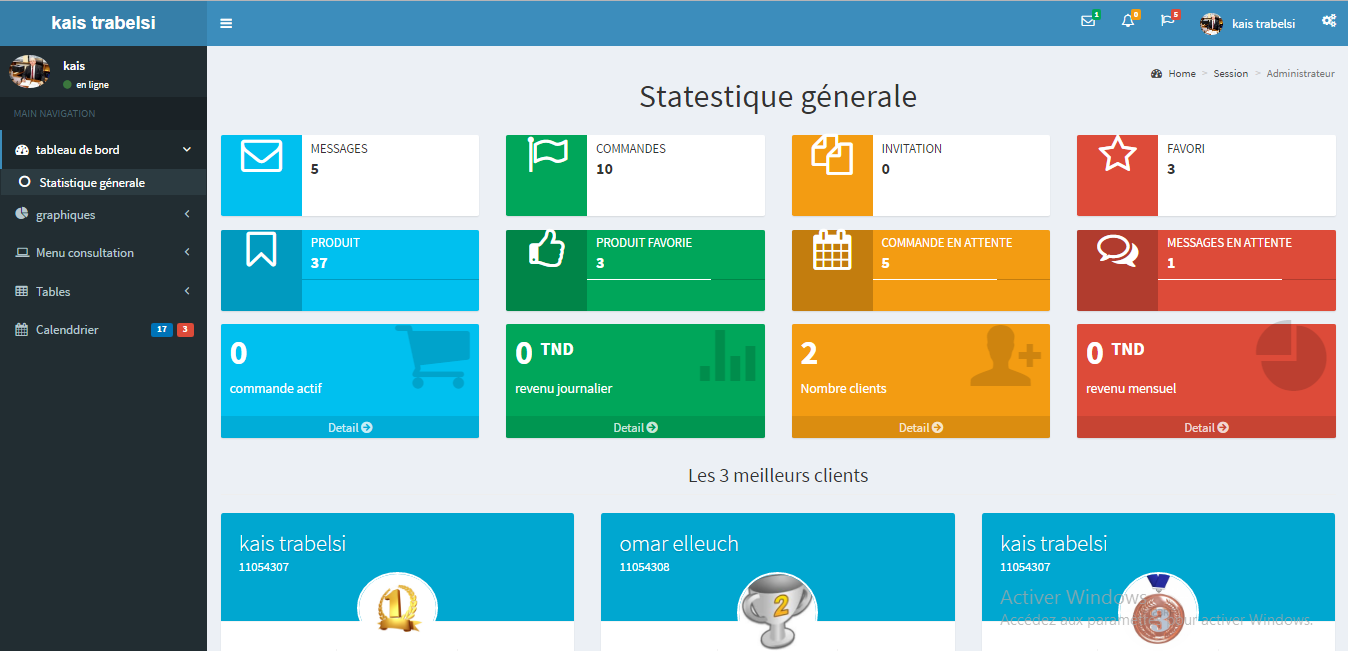
#### Interface « Menu administrateur »

Avec cette interface l’administrateur peut avoir une statistique générale qui existe dans la base des données comme le nombre des commandes a effectuer, le nombre des messages, le nombre des produits favori, les points d’achat, le nombre des clients etc.

En plus l’interface « Menu client »est composée par l’ensemble des liens suivants :

**Menu consultation :**

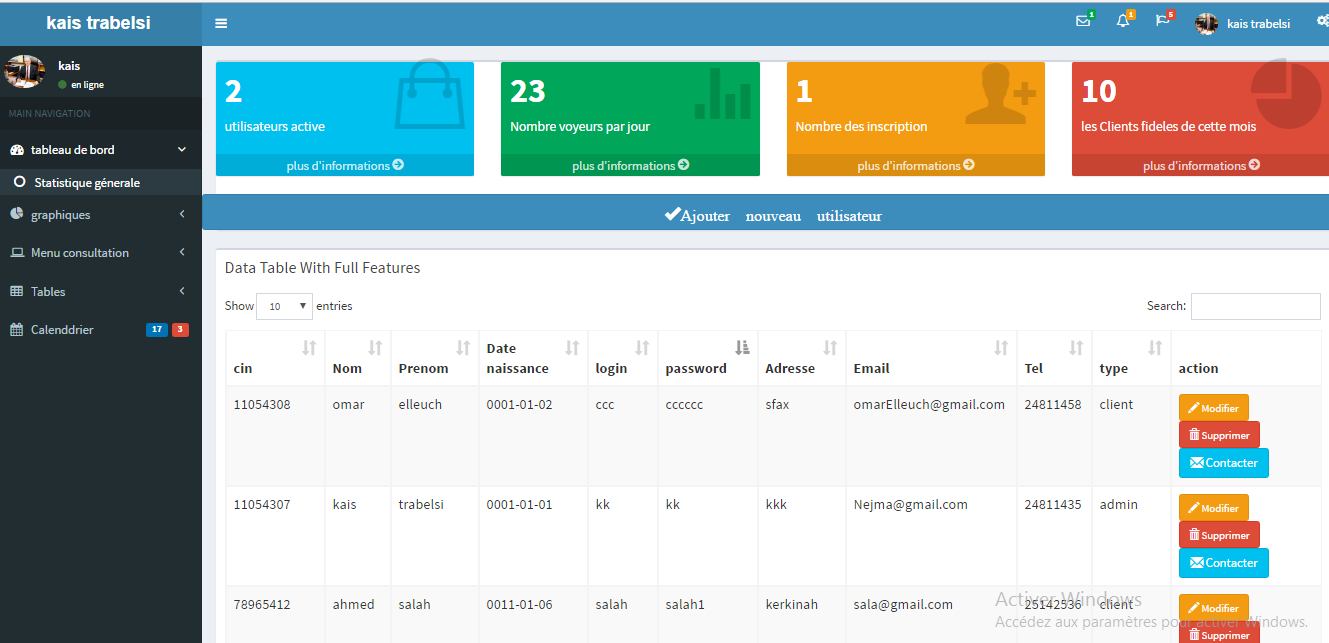
* + Liste produit : amène l’administrateur vers la liste des produits qui existent dans la base des données.
  + Liste Catégorie : amène le client vers la liste des produits en promotion qui existe dans la base des données.
  + Liste utilisateur : amène l’administrateur vers votre panier qui existe dans la base des données.
  + Consulter liste commande : amène l’administrateur vers la liste commande qui existe dans la base des données.
  + Liste message : amène l’administrateur vers la liste des messages qui existe dans la base des données.
  + Consulter liste demande : amène l’administrateur vers la liste demande qui existe dans la base des données.
  + Liste facture : amène l’administrateur vers la liste des factures qui existe dans la base des données.



**Figure 49 : interface «Menu administrateur»**

#### Interface liste utilisateur

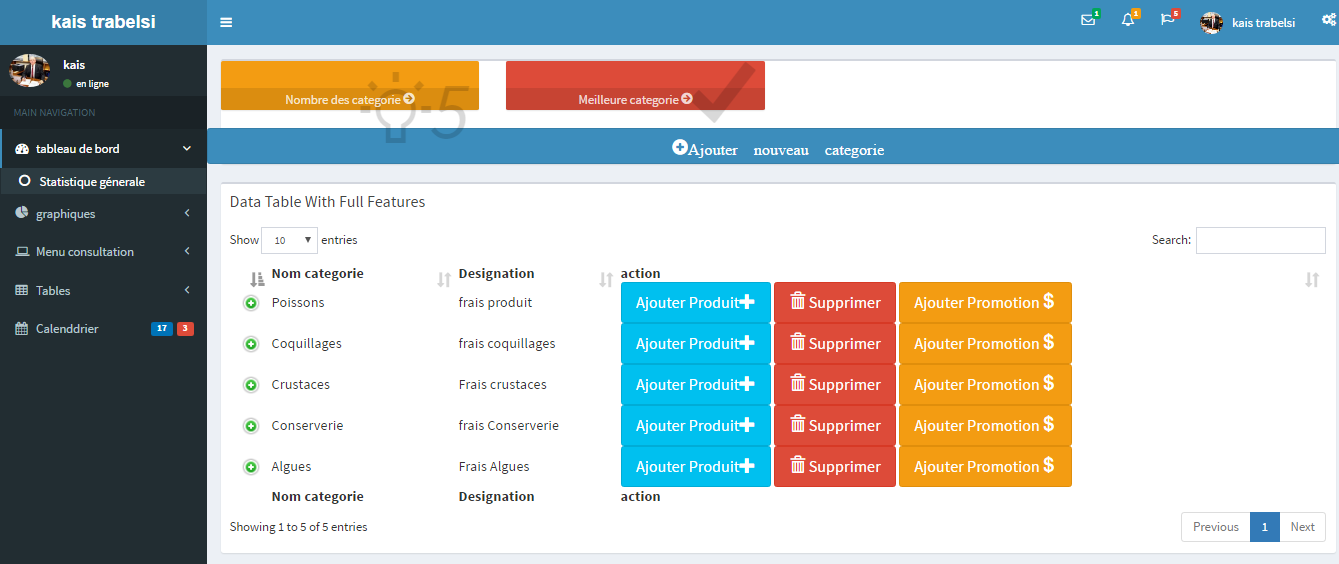
Cette interface permet à l’administrateur de consulter la liste des utilisateurs et il peut effectuer des recherches personnalises à l’aide des filtres en fonction de plusieurs critères état, type etc., de modifier les informations du client, supprimer un client, et aussi de contacter le client à travers un message

****

**Figure 50 : interface « Liste utilisateur»**

#### Interface liste catégorie

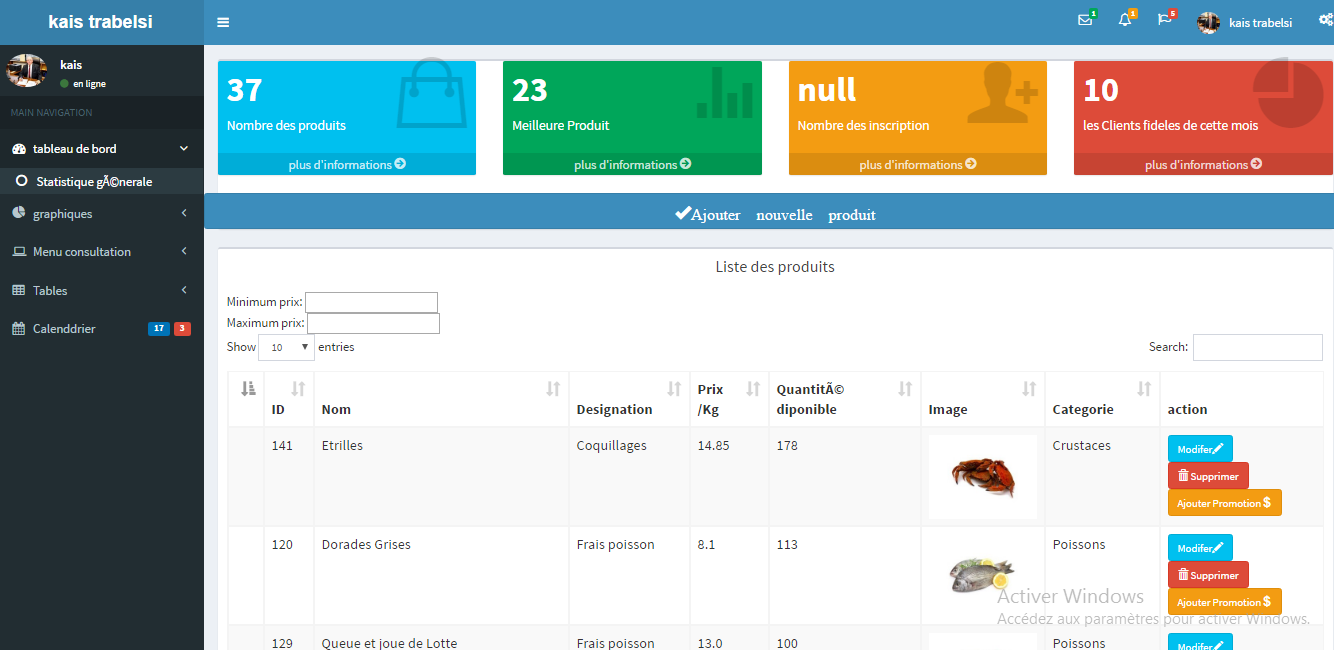
Cette interface permet à l’administrateur de consulter la liste des catégories et il peut effectuer des recherches personnalises à l’aide des filtres en fonction de plusieurs critères état, type etc., de modifier catégorie, supprimer catégorie, et aussi de faire une promotion d’une catégorie.



**Figure 51 : interface « Liste catégorie»**

#### Interface liste produit

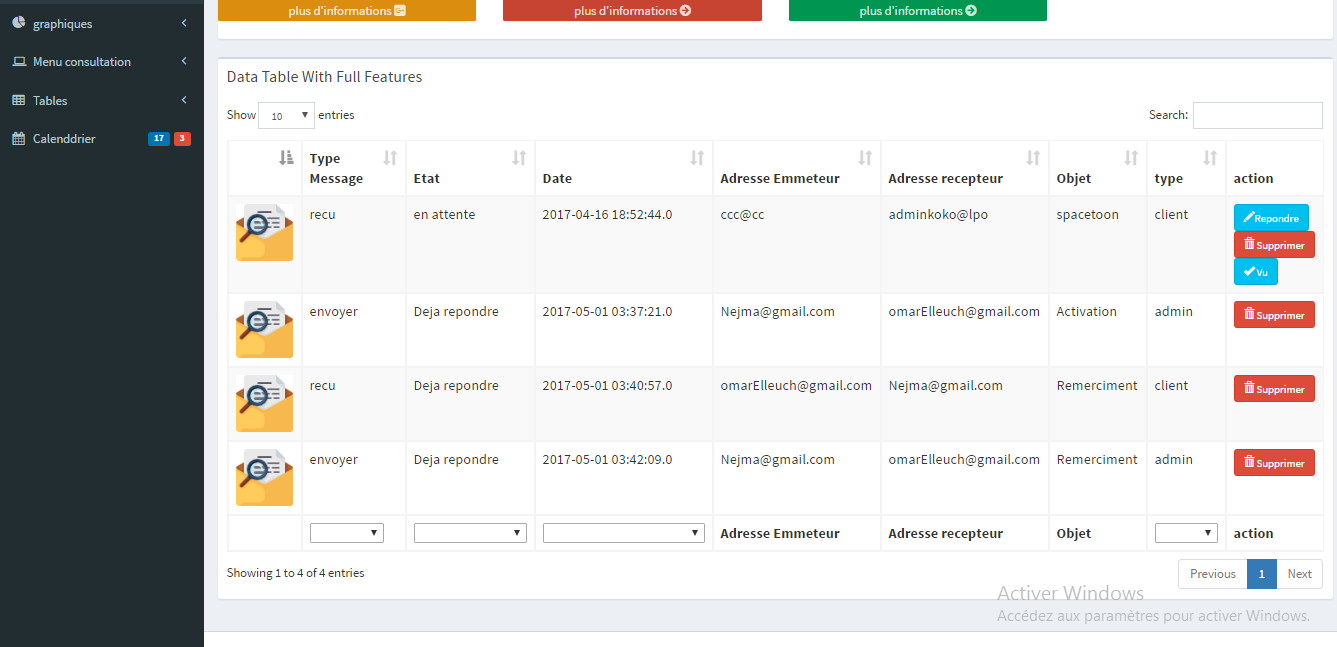
Cette interface permet l’administrateur de consulter la liste des produit et il peut effectuer des recherches personnalises à l’aide des filtres en fonction de plusieurs critères état, type etc., de modifier produit, supprimer produit, et aussi de faire une promotion a un produit.



**Figure 52 : interface « Liste produit»**

#### Interface liste message

Cette interface permet l’administrateur de consulter la liste des messages et il peut effectuer des recherches personnalises à l’aide des filtres en fonction de plusieurs critères état, type etc., de répondre a message client, faire une vue pour un message, et aussi de supprimer un message



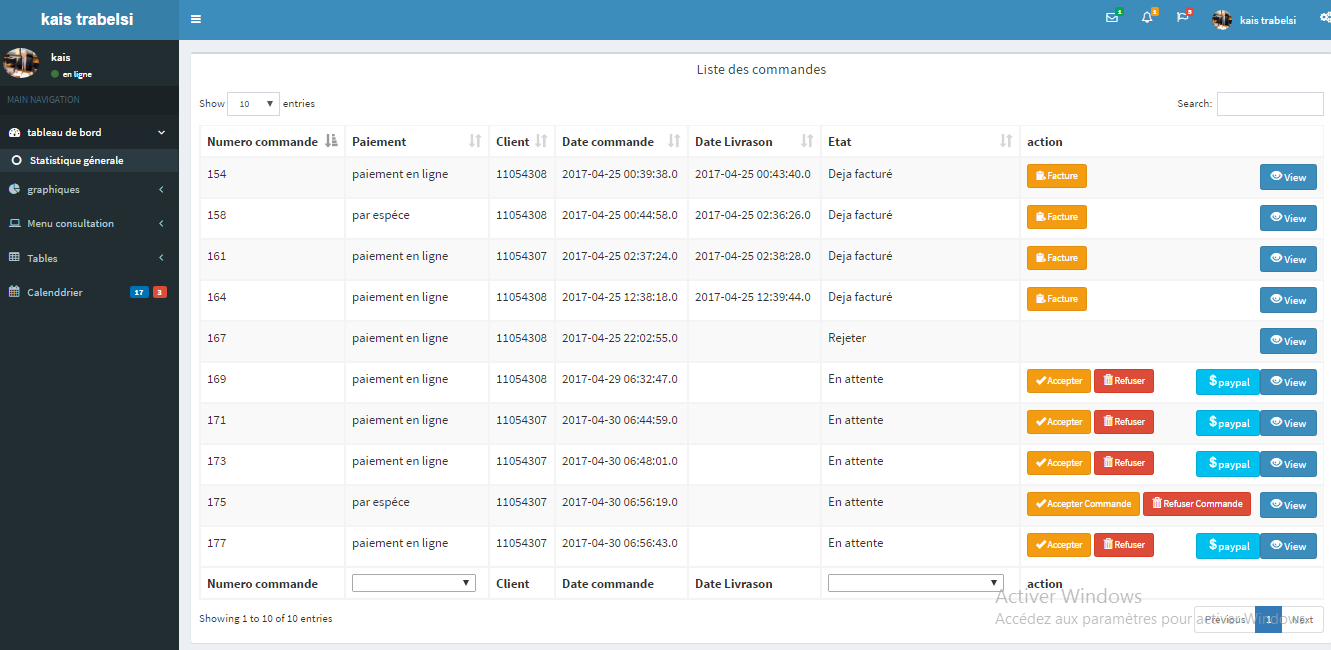
**Figure 53 : interface « Liste message»**

#### Interface liste commande

Cette interface permet l’administrateur de consulter la liste des commandes et il peut effectuer des recherches personnalises à l’aide des filtres en fonction de plusieurs critères état, type etc., d’accepter ou de refuser une commande, et aussi de facturer une commande

L’interface est composée par ensemble de liens :

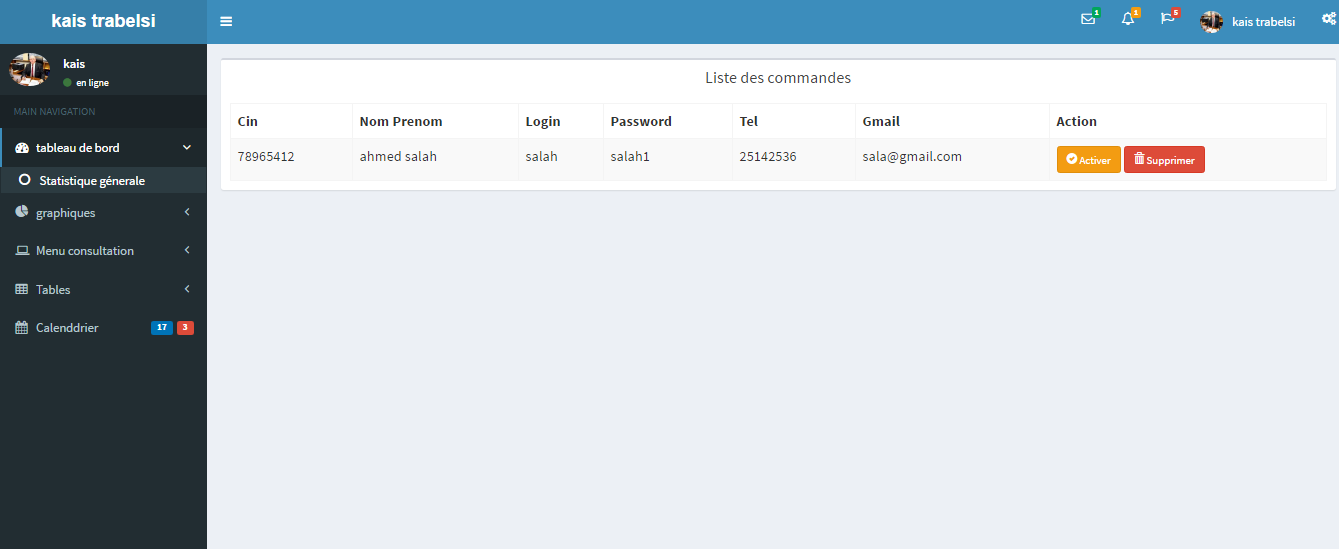
* Facture : voir la facture d’une commande sélectionné (seulement si la commande et activé)
* View : voir les détails de la commande
* Paypal : amener l’administrateur au site officiel «Paypal » pour vérifier le paiement de la commande



**Figure 54 : interface « Liste commande»**

#### Interface liste demande

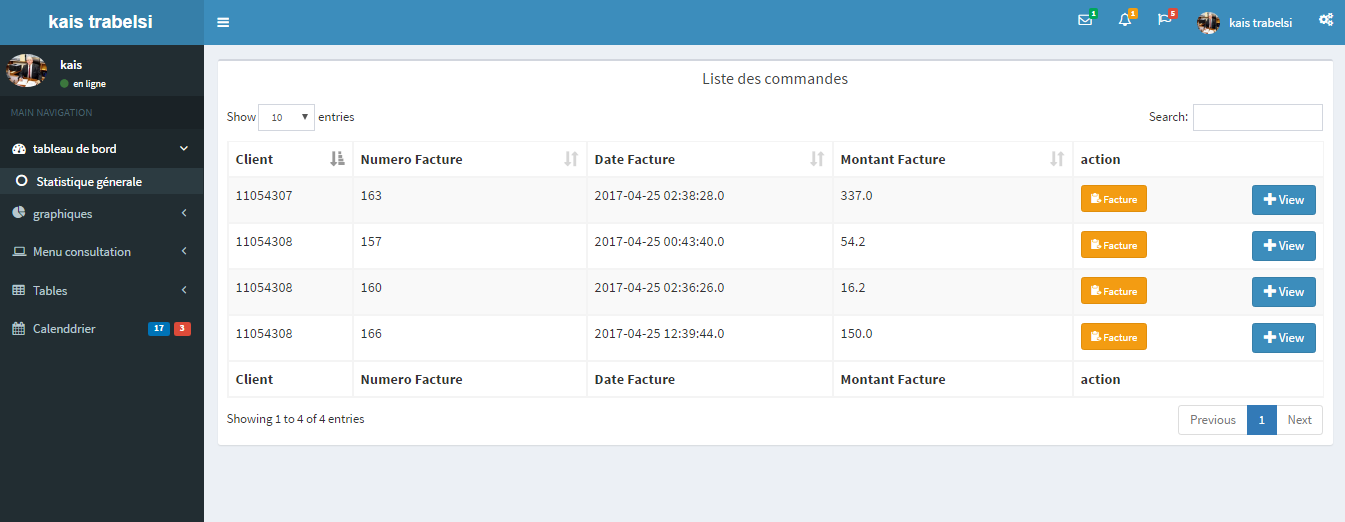
Cette interface permet à l’administrateur de consulter la liste des demandes et il peut effectuer des recherches personnalises à l’aide des filtres en fonction de plusieurs critères état, type etc., et aussi d’accepter ou de refuser une demande



**Figure 55 : interface « Liste demande»**

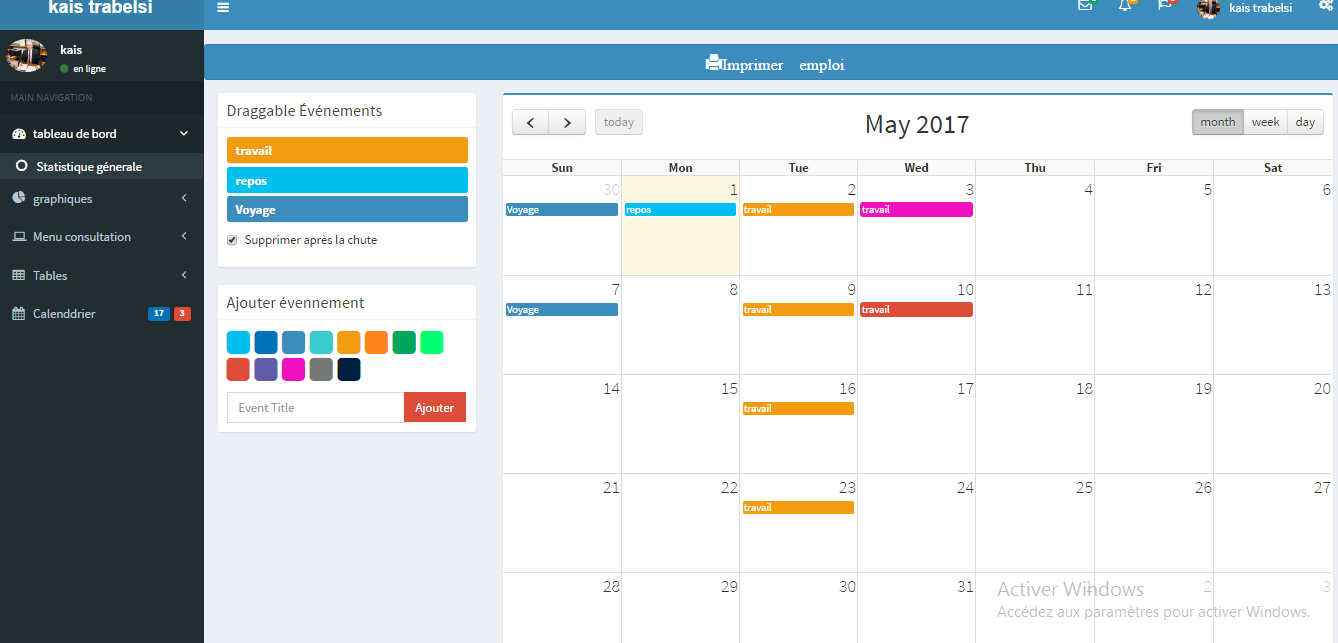
#### Interface liste facture

Cette interface permet à l’administrateur de consulter la liste des factures



**Figure 56 : interface « Liste facture»**

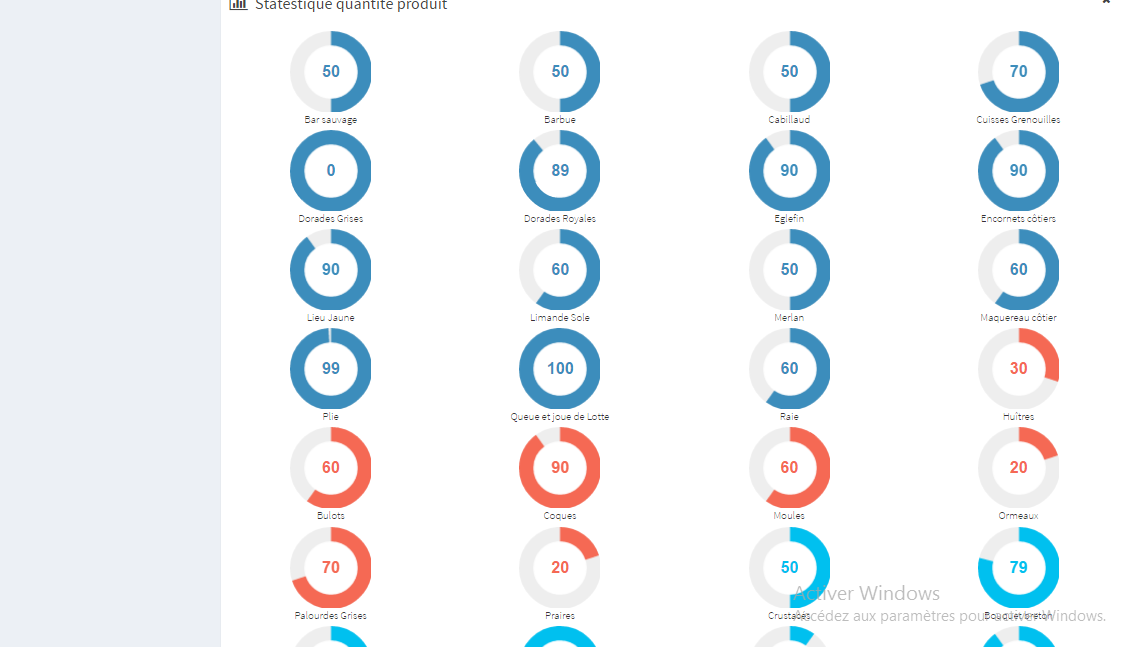
#### Interface emploi

Cette interface permet à l’administrateur de rédiger un emploi de travail et aussi d’imprimer cet emploi 

**Figure 57 : interface « Emploi»**

#### Interface Statistique des produits par quantité

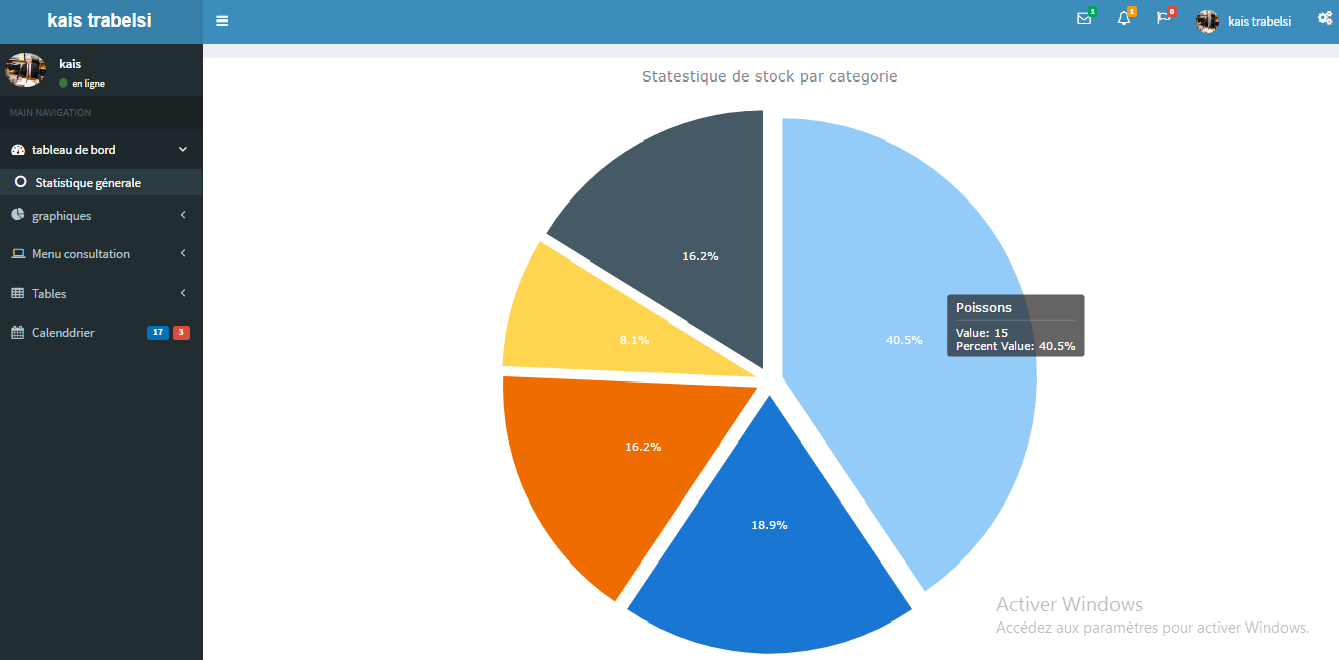
Cette interface permet à l’administrateur de visualiser les quantités de tous les produits par catégorie



**Figure 58 : interface « Statistique produit dans stock»**

#### Interface Statistique de stock

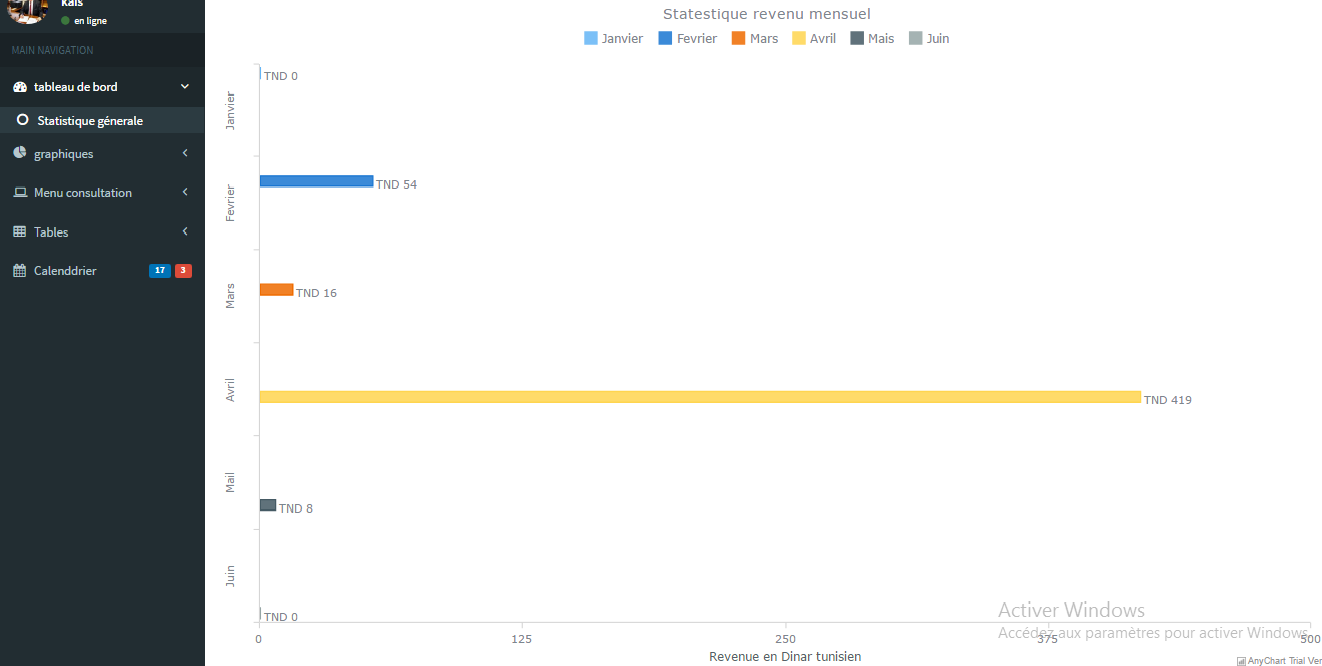
Cette interface permet à l’administrateur de visualiser le stock par catégorie



**Figure 59 : interface « Statistique de stock»**

#### Interface Statistique revenu mensuel

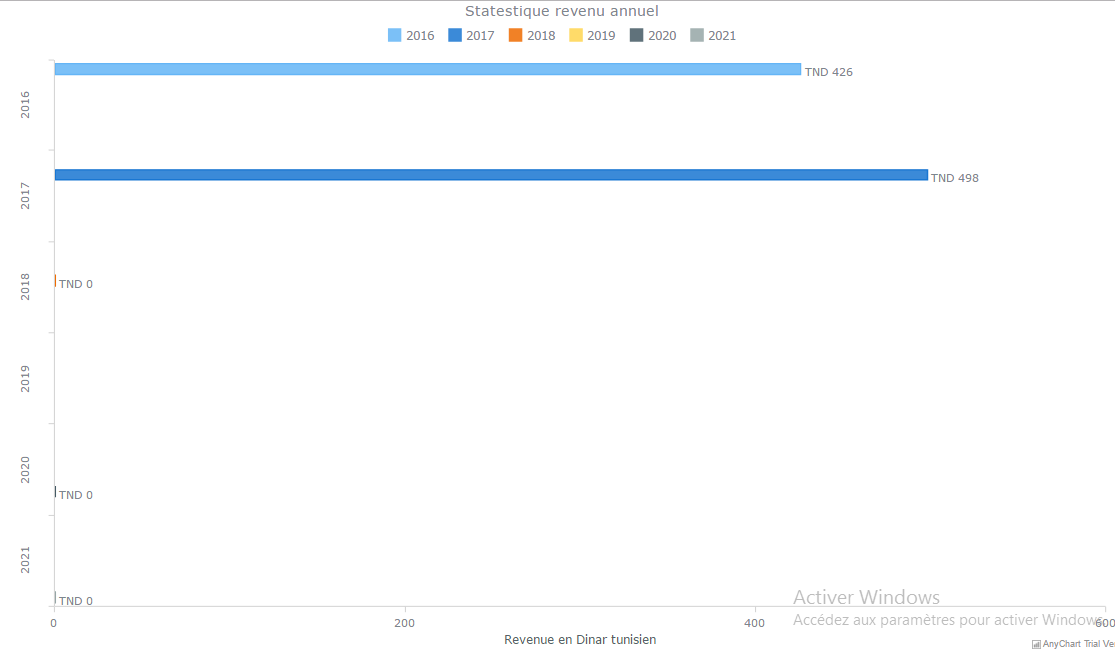
Cette interface permet à l’administrateur de visualiser les statistiques pour le revenu mensuel de chaque mois pendant l’année courante



**Figure 60: interface « Statistique revenu mensuel»**

#### Interface Statistique revenu annuel

Cette interface permet à l’administrateur de visualiser les statistiques du revenu de chaque année



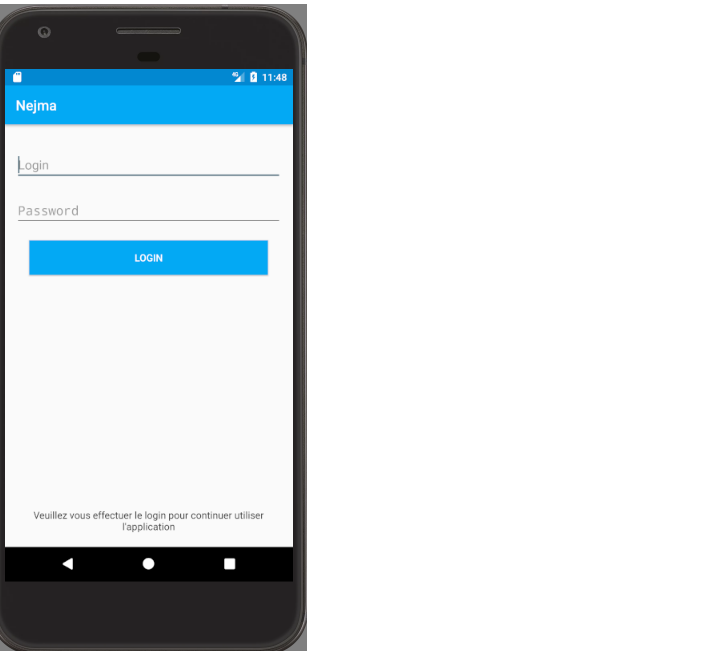
**Figure 61: interface « Statistique revenu annuel»**

### Interfaces de l’application mobile

### Interfaces menu principale

L’interface menue principale pour l’application mobile est présentée dans la figure ci-dessous.

Cette interface permet au client de saisir le login et le mot de passe pour pouvoir utiliser l’application.



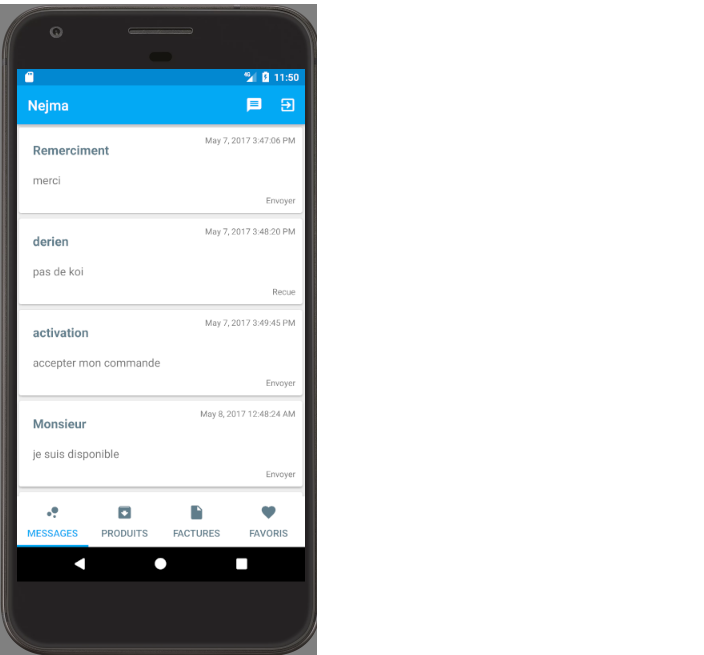
**Figure 62 : interface « login»**

### Interface liste des messages

L’interface « liste des messages principale interne » dans notre application mobile est présentée dans la figure ci-dessous.

Cette interface permet au client de consulter la liste des messages.

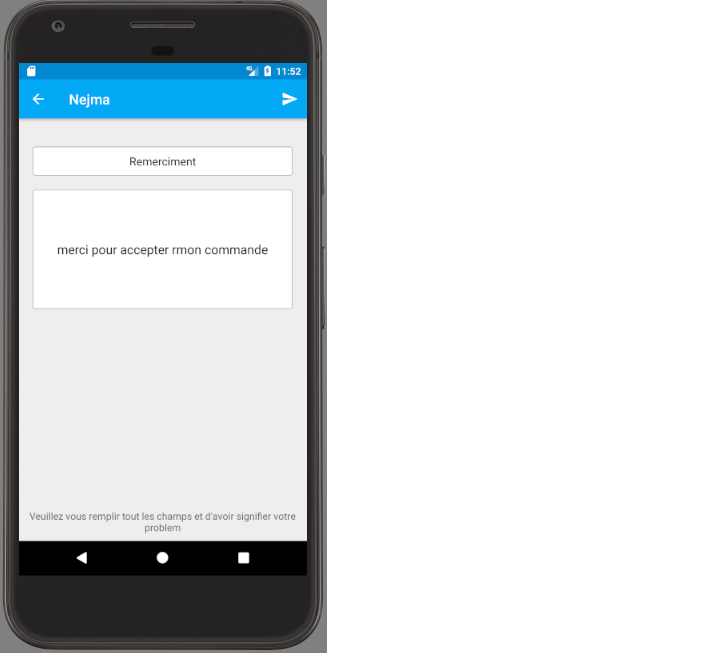
Cette interface possède un bouton qui amène l’utilisateur vers une autre activité intitulée «Envoyer Message ».



**Figure 63 : interface « Liste message»**

### Interface Envoyer message

Cette interface permet au client d’envoyer un message vers l’administrateur

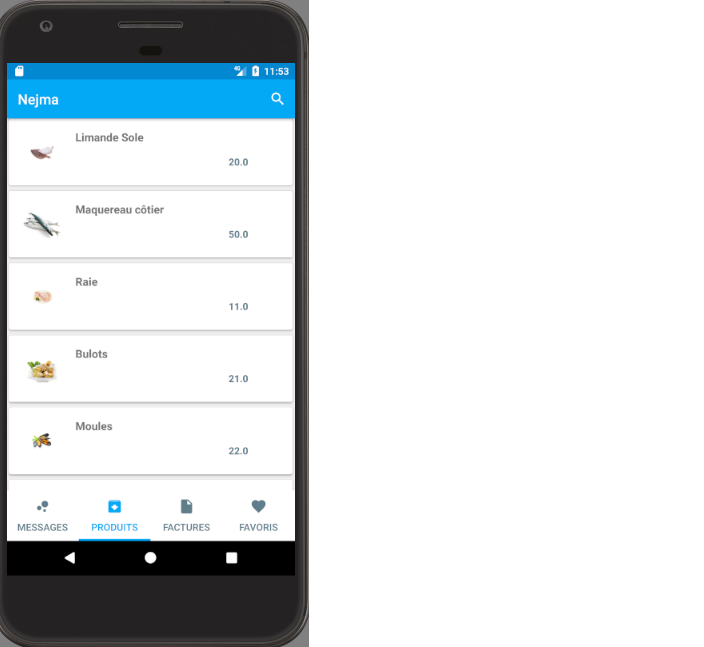


**Figure 64 : interface « message»**

### Interface liste des produits

Cette interface permet au client de consulter la liste des produits

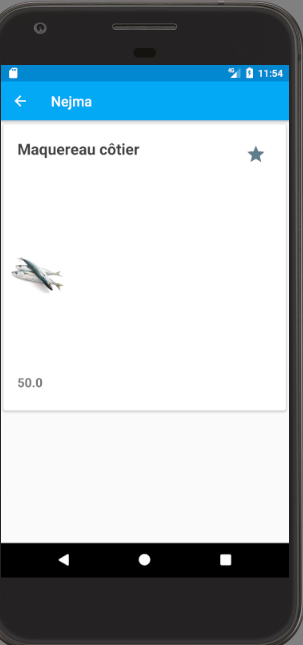
Cette interface possède un bouton qui amène l’utilisateur vers une autre activité pour consulter les détails d’un produit.



**Figure 65 : interface « Liste produit»**

### Interface détail produit

Cette interface permet au client de consulter les détails d’un produit sélectionné en plus que ca elle offre au client un bouton qui permet d’ajouter le produit dans la liste favori

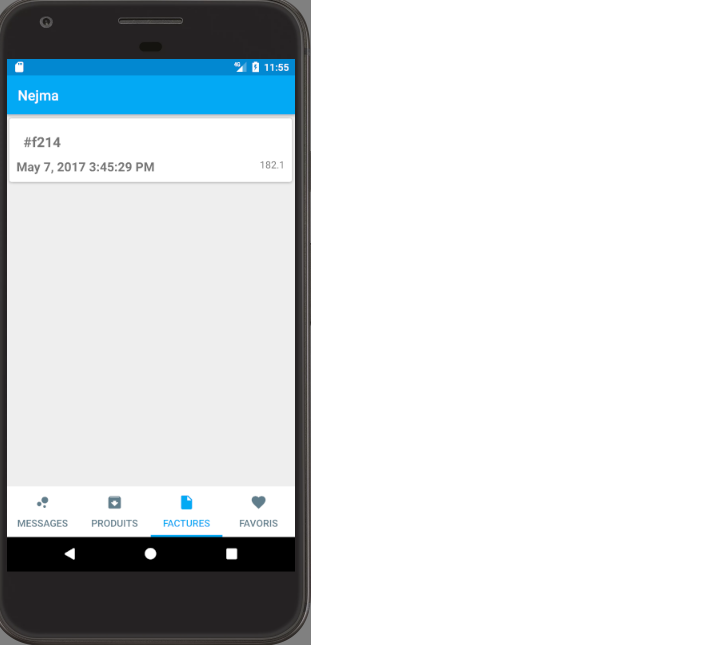


**Figure 66 : interface « détail produit»**

### Interface liste des factures

Cette interface permet au client de consulter la liste des factures

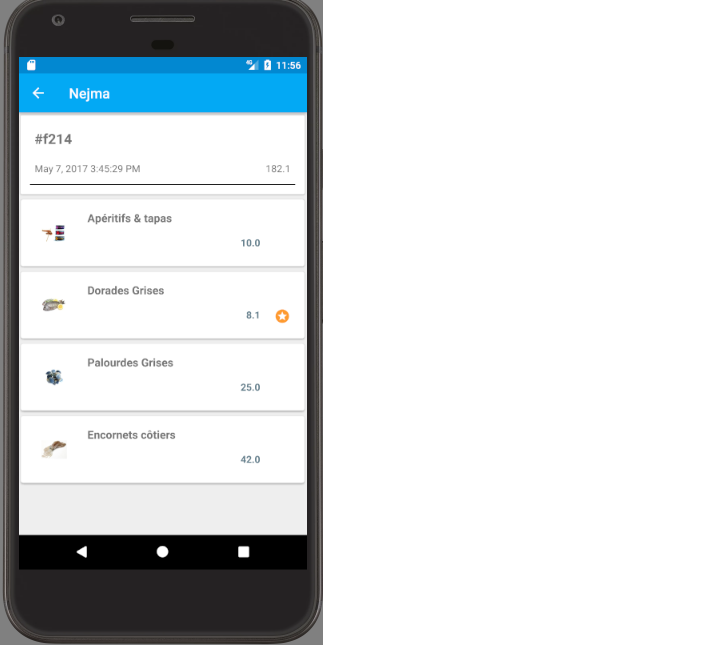
Cette interface possède un autre processus qui amène l’utilisateur vers les détails de la facture dans une autre activité lorsqu’il clique sur un item.



**Figure 67: interface « Liste facture»**

### Interface détail facture

Cette interface permet au client de consulter les détails d’une facture par exemple le montant de la facture, les produits achetés etc.

 **Figure 69 : interface « Facture détail»**

### Interface liste des produits favori

Cette interface permet au client de consulter la liste des produits favoris.

Cette interface possède un bouton qui amène l’utilisateur vers une autre activité qui s’appelle « détails produit » .



Figure 69 : interface « Liste favori»

## Apports

Il s’agit de décrire l’apport au niveau des connaissances techniques, de la conception et du développement.



### Au niveau des connaissances techniques

La réalisation d’une telle application nous a permis d’enrichir nos connaissances techniques telles que:

- La façon de collecter les informations et la manière de résoudre les problèmes rencontrés.

- L’application de nos connaissances théoriques en informatique.

- Mettre en œuvre les notions et les connaissances acquises au sein des modules « méthodologie de conception des bases de données ».

- L’organisation et la méthodologie du travail.

### Au niveau de la conception et du développement

L’apport acquis au niveau de la conception et du développement est appréhendé par les aspects suivants:

- L’application du langage de modélisation UML.

- La maitrise du langage HTML 5 et Java.

- La maitrise de la plateforme Java EE ainsi que les services web

- la maitrise de l’environnement de développement mobile « Android »

## Conclusion

Dans ce chapitre, nous avons pu présenter l’environnement et le processus de développement.

Nous avons exposé ainsi le résultat de développement à l’aide des aperçus écran. Nous avons clôturé par une évaluation du travail réalisé.

# Conclusion générale et Perspectives

Notre projet intitulé «Conception et développement d'une application web et mobile pour le commerce électronique» consiste à la conception et la réalisation d’un site web et une application mobile destinée pour les clients de la société NEJMA. Contrairement à la majorité des travaux existant sur le marché qui offrent des fonctionnalités limitées, nous avons réalisé un système qui permet à la fois de gérer des produits ainsi que leurs promotions, donner la possibilité aux gérants de guider les clients. En ce qui concerne la démarche, nous avons en premier lieu effectué une phase d’étude des différents outils existants. En deuxième lieu nous avons spécifié notre application pour discerner les fonctionnalités. En troisième lieu, nous avons procédé à sa conception ainsi qu’aux choix technologiques pour sa réalisation. Enfin, nous l’avons mise en œuvre. Toutes les fonctionnalités décrites dans le cahier des charges ont été développées et validées.

Il est important de noter que la réalisation de ce projet nous a été bénéfique sur le plan technique, ce projet nous a été une bonne occasion pour découvrir et maîtriser le langage de modélisation UML et les environnements de programmation «Java, Java EE et Android».

Comme toute expérience au niveau de l'application des concepts et langages, il n’est pas pouvoir éviter certains problèmes et difficultés au niveau de la modélisation conceptuelle et au niveau de l'implémentation et la programmation.

Cette solution ne prétend nullement être la meilleure, puisque en informatique, il n'y a pas de solution parfait absolue.

II est à noter que cette application peut être améliorée, pour répondre aux besoins d’autres spécialités plus appropriées.

BIBLIOGRAPHIE

* Titre : UML2 par la pratique

-Auteur :Pascal Roques

* Titre : Support de cours EJB3.

-Auteur : Mohamed Youssfi

* Titre : Programmation mobile avec Android.

-Auteur :Pierre Nerzic

NETOGRAPHIE

* <http://fr.openclassrooms.com>
* <http://fr.wikipedia.org>
* <http://www.developpez.net/>
* <http://www.tutorialspoint.com/>
* <https://jersey.java.net/>
* <https://github.com>
* <http://developer.android.com/>
* <http://getbootstrap.com/>

Annexe

## Java Enterprise Edition(J2EE)

Dans le monde actuel de l'informatique et des technologies de l’information et de la communication, les applications sont de plus en plus complexes, et doivent être réalisées en un minimum de temps et de coût et portable à travers les différents systèmes utilisés. Sun Microsystems a proposée pour cela la plate-forme Java Enterprise Edition ou Java EE, crée dans le but de faciliter la construction d'applications distribuées, et a doté le langage Java d’une plate-forme logicielle fiable pour répondre aux besoins des entreprises. Java EE (Java Enterprise Edition), anciennement appelée « J2EE » dans sa version 2, est une extension de la plate-forme standard Java SE qui contient les API (Application Programming Interface) de base de Java. Ajoutant un grand nombre de bibliothèques remplissant des fonctionnalités supplémentaires destinées aux applications d’entreprise, Cela inclut le stockage sécurisé des informations, leur manipulation et leur traitement.

Le but étant de faciliter le développement d'applications robustes, interopérables et capables de s'adapter aux changements et à une croissance rapide sans poser de problème de sécurité, en fournissant un ensemble de composants sous forme d'APIs. Java EE est une architecture performante qui exploite au maximum la technologie des objets distribués, ce qui permet d'avoir un système simple et léger à utiliser côté clients. Java EE se définit par les méthodes de programmation java destinées aux applications entreprises grâce à la spécification JCP (JSR 316)

La plate-forme Java EE est passée par plusieurs étapes nous citons :

* La première version standard J2EE 1.2 a été développée par Sun en 1999. Les Entreprises JavaBeans (EJB) permettaient de gérer la couche métier d'accès aux données alors que les Servlets et JavaServer Pages (JSP) permettaient de développer les applications serveur.
* La version J2EE 1.3 (2001) n'apportait pas de nouveautés essentielles mais des améliorations de la plate-forme à la suite des demandes utilisateur, comme la configuration par fichier XML.
* La version J2EE 1.4 (2003) Cette version apportait le support des services Web et des EJB version 2.1. Mais cette version conséquente était difficile à utiliser, tester et déployer. Les développements par couches, de type Modèle Vue Contrôleur (MVC), ne facilitent pas la compréhension de l'ensemble, et les technologies Servlet et JSP ne sont pas adaptées pour un développement rapide. En réponse à cela, quelques Framework ou outils spécialisés ont vu le jour, comme Struts et Spring pour la partie développement serveur, et Hibernate ou TopLink pour la persistance.
* La version Java EE 5 (2006) apportait une réponse à cela en proposant un outil de développement côté serveur adapté au design pattern MVC, nommé JavaServer Faces.

Cette nouvelle version était une étape majeure pour la plate-forme Java EE qui s'orientait vers la simplicité, en proposant les EJB 3.

* La version Java EE 6 (2009) se concentre sur la facilité d'utilisation de la plate-forme pour le développement, les tests et le déploiement. Cette version propose une simplification des EJB avec la version 3.1. Java EE est devenu un standard utilisé par de nombreuses sociétés à travers le monde professionnel et celui de l'Open Source. Plusieurs outils peuvent être associés à Java EE comme les serveurs d'applications (GlassFish, Websphere, Weblogic, JBoss...), des Framework de développement (Struts, Spring...), des outils de persistance. Ces mécanismes font en sorte qu'un programme puisse se terminer sans que ses données et son état d'exécution ne soient perdus. Ces informations de reprise peuvent être sauvegardées sur disque, éventuellement sur un serveur distant (un serveur de bases de données relationnelles, (EclipseLink, Hibernate...).

JEE est une plate-forme fortement orientée serveur, autrement dit, une application Java EE s'exécute sur un serveur d'applications. Un serveur d'applications est constitué d'un serveur HTTP et d'un conteneur web (pour exécuter les servlets et les JSP).

L’utilisation de JEE pour développer et exécuter une application propose plusieurs avantages Une architecture d'application basée sur les composants qui permet un découpage de l'application et donc une séparation des rôles lors du développement. La possibilité de s'interfacer avec le système d'information existant grâce à de nombreuses API : JDBC, JNDI, JMS, JCA ...

La possibilité de choisir les outils de développement et le ou les serveurs d'applications utilisés qu'ils soient commerciaux ou libres. Java EE permet une grande flexibilité dans le choix de l'architecture de l'application en combinant les différents composants. Ce choix dépend des besoins auxquels doit répondre l'application mais aussi des compétences dans les différentesAPI de Java EE. L'architecture d'une application se découpe idéalement en au moins trois tiers La partie cliente : c'est la partie qui permet le dialogue avec l'utilisateur. Elle peut être composée d'une application stand-alone, d'une application web ou d'applets. La partie métier : c'est la partie qui encapsule les traitements (dans des EJB ou des JavaBeans).

La partie donnée : c'est la partie qui stocke les données. Depuis une dizaine d’années, le monde des entreprises se confond de plus en plus avec le monde de l’Internet, et aujourd’hui la plus part des applications sont écrites pour être accessibles depuis un navigateur Web.

## JSP et Servlet

Les JSP (Java Server Pages), une technologie Java qui permettent la génération de pages web dynamiques. La technologie JSP permet de séparer la présentation sous forme de code HTML et les traitements sous formes de classes Java définissant un bean ou une servlet (une classe Java qui permet de créer dynamiquement des données au sein d'un serveur http).

Les servlets sont la base de la programmation Web JEE. Toute la conception d’un site Web en Java repose sur ces éléments que sont les servlets. Une servlet est un programme Java qui tourne sur la machine où est installé le serveur JEE, et qui est invoqué lorsqu’un navigateur client appelle l’URL liée à ce programme Pour exécuter une servlet, il suffit de saisir une URL qui désigne la servlet dans un navigateur.

Les JSP permettent d'introduire du code Java dans des tags prédéfinis à l'intérieur d'une page HTML. La technologie JSP mélange la puissance de Java côté serveur et la facilité de mise en page d'HTML côté client. Concrètement, les JSP sont basées sur les servlets. Au premier appel de la page JSP, le moteur de JSP génère et compile automatiquement une servlet qui permet la génération de la page web. Le code HTML est repris intégralement dans la servlet. Le code Java est inséré dans la servlet.

La servlet générée est compilée et sauvegardée puis elle est exécutée. Les appels suivants de la JSP sont beaucoup plus rapides car la servlet, conservée par le serveur, est directement exécutée. Il y a plusieurs manières de combiner les technologies JSP, les beans/EJB et les servlets en fonction des besoins pour développer des applications web. Comme le code de la servlet est généré dynamiquement, les JSP sont relativement difficiles à déboguer. Cette approche possède plusieurs avantages :

L’utilisation de Java par les JSP permet une indépendance de la plate-forme d'exécution mais aussi du serveur web utilisé. La séparation des traitements et de la présentation : la page web peut être écrite par un designer et les tags Java peuvent être ajoutés ensuite par le développeur. Les traitements peuvent être réalisés par des composants réutilisables (des Java beans). Les JSP sont basées sur les servlets : tout ce qui est fait par une servlet pour la génération de pages dynamiques peut être fait avec une JSP.

## Entreprise Java Beans(EJB)

Enterprise JavaBeans(EJB) est une architecture de composants logiciels côté serveur pour la plateforme de développementJava EE.

Cette architecture propose un cadre (Framework) pour créer des composants distribués (c’est-à-dire déployés sur des serveurs distants) écrit en langage de programmationJava hébergés au sein d'unserveur applicatif.

Les EJB permettant de :

* Représenter des données manipulées par l’application (EJB Entity),
* Proposer des services avec ou sans conservation d'état entre les appels (EJB Session),
* Accomplir des tâches de manière asynchrone (EJB Message DrivenBean).

Tous les EJB peuvent évoluer dans un contextetransactionnel.

Les Entreprise Java Bean ou EJB sont des composants qui permettent de d’implémenter le logique métier, ce sont des composant distribuésqui s’exécutent au sein d’un conteneur EJB qui fait partie du serveur d’application J2EE, le but des EJB est de faciliter la création d'applications distribuées pour les entreprises.

Une des principales caractéristiques des EJB est de permettre aux développeurs de se concentrer sur les traitements orientés métiers car les EJB et l'environnement dans lequel ils s'exécutent prennent en charge un certain nombre de traitements techniques en utilisant les services de l’infrastructure offert par le serveur d’application tel que :

◦La distribution

◦La gestion des transactions,

◦La persistance des données,

◦Le cycle de vie des objets

◦La montée en charge

◦La concurrence

◦La sécurité

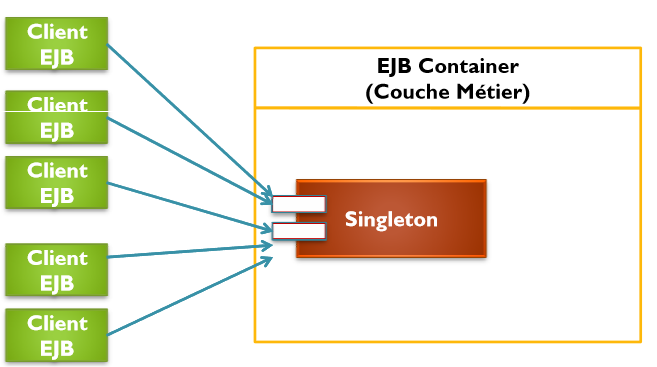
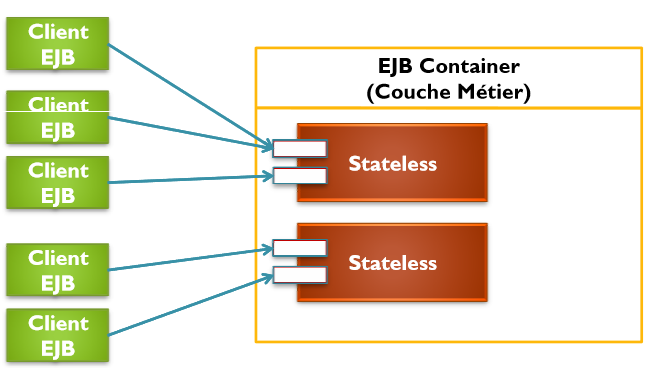
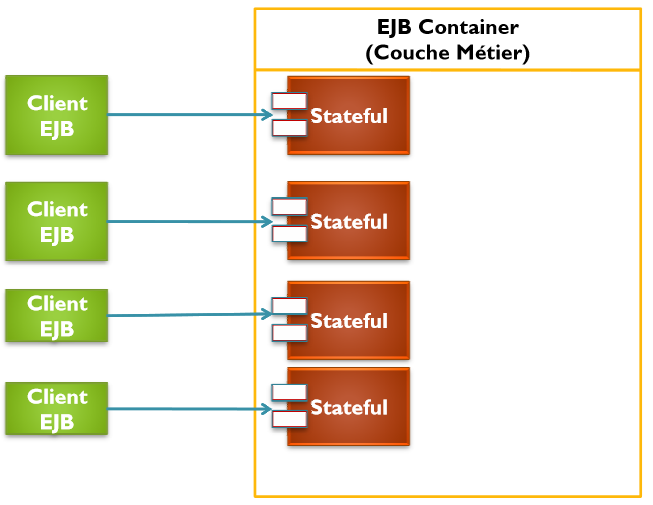
◦La sérialisation

◦Journalisation

◦Etc….

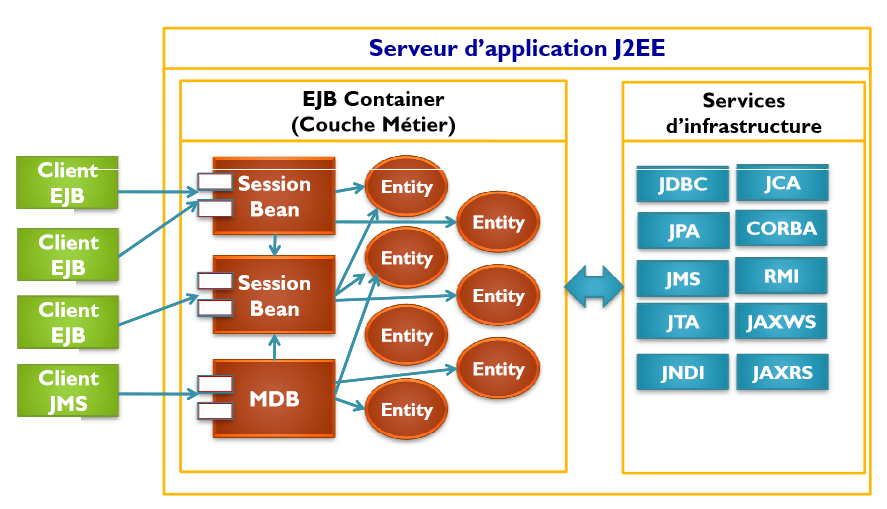
Autrement dit, EJB permet de séparerle code métier qui est propre aux spécifications fonctionnelles du code technique (spécification non fonctionnel).

Il existe trois types d'EJB:

* EntityBeans: Les EJB entités représentent les données manipulées par l’application (Composants persistants), chaque EJB Entityest associé à une table au niveau de la base de données
* Session Beans: Les EJB session sont des composants distribués qui implémentent les traitements du logique métier, ces composants sont accessibles à distance via les protocoles RMI et IIOP, il existe trois types d’EJB Session.
* Stateless (sans état): une instance est créée par le serveur pour plusieurs connexions clientes, ce type de beanne conserve aucune donnée dans son état.
* Statefull(avec état) : création d’une instance pour chaque connexion cliente, ce type de beanpeut conserver des données entre les échanges avec le client.
* Singleton (Instance Unique) : création d’une instance unique quel que soit le nombre de connexion.
* Message Driven Beans: Beans de messages, un listener qui permet de déclencher des traitements au niveau de l’application suite à la réception d’un message asynchrone JMS.
* **EJB Session Singleton**
* Création d’une seule instance
* **EJB Session Stateless**
* Création d’un pool d’instances
* **EJB Session Stateful**
* Création d’une instance pour chaque connexion

Le rôle du conteneur EJB et de :

* Gérer le cycle de vie des composants EJB de votre application : Instanciation, accès au bean, sérialisation, désérialisation et destruction du bean.
* Permettre aux EJB d’accéder aux services d’infrastructures offerts par le serveur d’application JEE.



## Android

Prononcé Android, est un [système d'exploitation mobile](http://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27exploitation_mobile) pour [Smartphones](http://fr.wikipedia.org/wiki/Smartphone), [tablettes tactiles](http://fr.wikipedia.org/wiki/Tablette_tactile), [PDA](http://fr.wikipedia.org/wiki/Assistant_personnel) et terminaux mobiles. C'est un système [open source](http://fr.wikipedia.org/wiki/Open_source)[2](http://fr.wikipedia.org/wiki/Android#cite_note-2), [3](http://fr.wikipedia.org/wiki/Android#cite_note-3) utilisant le [noyau Linux](http://fr.wikipedia.org/wiki/Noyau_Linux). Il a été lancé par une [startup](http://fr.wikipedia.org/wiki/Startup) du même nom rachetée par[Google](http://fr.wikipedia.org/wiki/Google)[4](http://fr.wikipedia.org/wiki/Android#cite_note-4) en [2005](http://fr.wikipedia.org/wiki/2005). D'autres types d'appareils possédant ce système d'exploitation existent, par exemple des [téléviseurs](http://fr.wikipedia.org/wiki/T%C3%A9l%C3%A9viseur), des[radio-réveils](http://fr.wikipedia.org/wiki/R%C3%A9veille-matin), des montres connectées, des [autoradios](http://fr.wikipedia.org/wiki/Autoradio) et même des [voitures](http://fr.wikipedia.org/wiki/Voiture).

**Caractéristiques :**

Dans le Guide du développeur, Android est défini comme étant une pile de logiciels, un ensemble de logiciels destinés à fournir une solution clé en main pour les appareils mobiles – [Smartphones](http://fr.wikipedia.org/wiki/Smartphone) et [tablettes tactiles](http://fr.wikipedia.org/wiki/Tablette_tactile). Cette pile comporte un système d'exploitation (comprenant un [noyau Linux](http://fr.wikipedia.org/wiki/Noyau_Linux)), les applications clés telles que le navigateur web, le téléphone et le carnet d'adresses ainsi que des logiciels intermédiaires entre le système d'exploitation et les applications[5](http://fr.wikipedia.org/wiki/Android#cite_note-smith-5). L'ensemble est organisé en cinq couches distinctes:

* le [noyau Linux](http://fr.wikipedia.org/wiki/Noyau_Linux) avec les [pilotes](http://fr.wikipedia.org/wiki/Pilote_informatique) ;
* des [bibliothèques logicielles](http://fr.wikipedia.org/wiki/Biblioth%C3%A8que_logicielle) telles que [WebKit](http://fr.wikipedia.org/wiki/WebKit), [OpenGL](http://fr.wikipedia.org/wiki/OpenGL), [SQLite](http://fr.wikipedia.org/wiki/SQLite) ou [FreeType](http://fr.wikipedia.org/wiki/FreeType) ;
* une [machine virtuelle](http://fr.wikipedia.org/wiki/Dalvik_(machine_virtuelle)) et des bibliothèques permettant d'exécuter des programmes prévus pour la [plate-forme Java](http://fr.wikipedia.org/wiki/Plate-forme_Java) ;
* un [Framework](http://fr.wikipedia.org/wiki/Framework) - kit de développement d'applications ;
* un lot d'applications standard parmi lesquelles il y a un [environnement de bureau](http://fr.wikipedia.org/wiki/Environnement_de_bureau), un carnet d'adresses, un navigateur web et un téléphone.

Les services offerts par Android facilitent notamment l'exploitation des réseaux de télécommunications [GSM](http://fr.wikipedia.org/wiki/Global_System_for_Mobile_Communications), [Bluetooth](http://fr.wikipedia.org/wiki/Bluetooth), [Wi-Fi](http://fr.wikipedia.org/wiki/Wi-Fi) et[UMTS](http://fr.wikipedia.org/wiki/Universal_Mobile_Telecommunications_System), la manipulation de médias, notamment de la vidéo [H.264](http://fr.wikipedia.org/wiki/H.264), de l'audio [MP3](http://fr.wikipedia.org/wiki/MPEG-1/2_Audio_Layer_3) et des images [JPEG](http://fr.wikipedia.org/wiki/JPEG) ainsi que d'autres formats, l'exploitation des senseurs tels que les capteurs de mouvements, la caméra, la boussole et le récepteur [GPS](http://fr.wikipedia.org/wiki/Global_Positioning_System), l'utilisation de l'écran tactile, le stockage en [base de données](http://fr.wikipedia.org/wiki/Base_de_donn%C3%A9es), le [rendu](http://fr.wikipedia.org/wiki/Rendu) d'images en 2D ou 3D en utilisant le [processeur graphique](http://fr.wikipedia.org/wiki/Processeur_graphique), l'affichage de [page web](http://fr.wikipedia.org/wiki/Page_web), l'exécution [multitâche](http://fr.wikipedia.org/wiki/Multit%C3%A2che) des applications et l'envoi de messages [SMS](http://fr.wikipedia.org/wiki/Short_Message_Service)[6](http://fr.wikipedia.org/wiki/Android#cite_note-lee-6),[5](http://fr.wikipedia.org/wiki/Android#cite_note-smith-5).

Bien que ne faisant pas partie de la pile de logiciels, l'[environnement de développement](http://fr.wikipedia.org/wiki/Environnement_de_d%C3%A9veloppement) qui comporte un émulateur de téléphone et un plugin pour [Eclipse](http://fr.wikipedia.org/wiki/Eclipse_(projet)) peut aussi être considéré comme une fonctionnalité d'Android[5](http://fr.wikipedia.org/wiki/Android#cite_note-smith-5). Et [Google Play](http://fr.wikipedia.org/wiki/Google_Play), une boutique en ligne permettant l'achat et le téléchargement d'applications pour Android, joue un rôle essentiel pour la popularité de ce système d'exploitation[6](http://fr.wikipedia.org/wiki/Android#cite_note-lee-6).

Android est distribué en [open source](http://fr.wikipedia.org/wiki/Open_source) sous [licence Apache](http://fr.wikipedia.org/wiki/Licence_Apache). La licence autorise les constructeurs qui intègrent Android dans leurs appareils à y apporter des modifications leur permettant de se distinguer de leurs concurrents[6](http://fr.wikipedia.org/wiki/Android#cite_note-lee-6) et il a été adopté par de nombreux constructeurs de produits concurrents de l'[iPhone](http://fr.wikipedia.org/wiki/IPhone)[6](http://fr.wikipedia.org/wiki/Android#cite_note-lee-6).

Le [noyau Linux](http://fr.wikipedia.org/wiki/Noyau_Linux) est utilisé pour les fondations d'Android, les services classiques des [systèmes d'exploitation](http://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27exploitation) : utilisation des périphériques, accès aux réseaux de télécommunication, manipulation de la mémoire et des processus et contrôle d'accès. Il s'agit d'une [branche](http://fr.wikipedia.org/wiki/Fork_(d%C3%A9veloppement_logiciel)) du noyau Linux, modifiée en vue de son utilisation sur des appareils mobiles. Le [X Window System](http://fr.wikipedia.org/wiki/X_Window_System), les outils de [GNU](http://fr.wikipedia.org/wiki/GNU), ainsi que certains fichiers de configuration qui se trouvent d'ordinaire dans les [distributions Linux](http://fr.wikipedia.org/wiki/Distribution_Linux) ne sont pas inclus dans Android. L'équipe de développement d'Android a apporté de nombreuses améliorations au noyau Linux, et la décision a été prise par la communauté de développement de Linux d'incorporer ces améliorations dans le noyau Linux

## Services Web

Un service web (ou service de la toile[1](http://fr.wikipedia.org/wiki/Service_web#cite_note-1)) est un [programme informatique](http://fr.wikipedia.org/wiki/Programme_informatique) de la famille des [technologies web](http://fr.wikipedia.org/wiki/Technologies_Web) permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. Il s'agit donc d'un ensemble de fonctionnalités exposées sur [internet](http://fr.wikipedia.org/wiki/Internet) ou sur un [intranet](http://fr.wikipedia.org/wiki/Intranet), par et pour des applications ou machines, sans intervention humaine, de manière synchrone ou asynchrone. Le protocole de communication est défini dans le cadre de la norme [SOAP](http://fr.wikipedia.org/wiki/SOAP) dans la [signature](http://fr.wikipedia.org/wiki/Signature_num%C3%A9rique) du service exposé ([WSDL](http://fr.wikipedia.org/wiki/Web_Services_Description_Language)). Actuellement, le protocole de transport est essentiellement [HTTP](http://fr.wikipedia.org/wiki/Hypertext_Transfer_Protocol)(S).

Le concept a été précisé et mis en œuvre dans le cadre de Web Services Activity[2](http://fr.wikipedia.org/wiki/Service_web#cite_note-2), au [W3C](http://fr.wikipedia.org/wiki/W3C), particulièrement avec le protocole SOAP. Associé avec les [Échanges de Données Informatisés (EDI)](http://fr.wikipedia.org/wiki/%C3%89change_de_donn%C3%A9es_informatis%C3%A9), le consortium [ebXML](http://fr.wikipedia.org/wiki/Electronic_Business_using_XML) l'a utilisé pour automatiser des échanges entre entreprises. Cependant le concept s'enrichit avec l'approfondissement des notions de ressource et d'état, dans le cadre du modèle [REST](http://fr.wikipedia.org/wiki/Representational_State_Transfer), et l'approfondissement de la notion de service, avec le modèle [SOA](http://fr.wikipedia.org/wiki/Architecture_orient%C3%A9e_services).

Dans sa présentation la plus générale, un service web se concrétise par un agent, réalisé selon une technologie informatique précise, par un fournisseur du service. Un demandeur, à l'aide d'un agent de requête, utilise ce service. Fournisseur et demandeur partagent une même sémantique du service web, tandis qu'agent et agent de requête partagent une même description du service pour coordonner les messages qu'ils échangent[3](http://fr.wikipedia.org/wiki/Service_web#cite_note-3).

Il existe plusieurs technologies derrière le terme services web :

Les services web de type [Representational state transfer](http://fr.wikipedia.org/wiki/Representational_state_transfer) (REST) exposent entièrement ces fonctionnalités comme un ensemble de ressources ([URI](http://fr.wikipedia.org/wiki/URI)) identifiables et accessibles par la syntaxe et la sémantique du protocole [HTTP](http://fr.wikipedia.org/wiki/HTTP). Les Services Web de type [REST](http://fr.wikipedia.org/wiki/Representational_state_transfer) sont donc basés sur l'architecture du [web](http://fr.wikipedia.org/wiki/Web) et ses standards de base : [HTTP](http://fr.wikipedia.org/wiki/HTTP) et [URI](http://fr.wikipedia.org/wiki/URI) ;

Les Services Web [WS-\*](http://fr.wikipedia.org/wiki/Liste_des_sp%C3%A9cifications_des_Services_Web_WS-*) exposent ces mêmes fonctionnalités sous la forme de services exécutables à distance. Leurs spécifications reposent sur les standards SOAP et [WSDL](http://fr.wikipedia.org/wiki/Web_Services_Description_Language) pour transformer les problématiques d'intégration héritées du monde [Middleware](http://fr.wikipedia.org/wiki/Middleware) en objectif d'interopérabilité.

## JSON

JSON (JavaScript Object Notation – Notation Objet issue de JavaScript) est un format léger d'échange de données. Il est facile à lire ou à écrire pour des humains. Il est aisément analysable ou générale par des machines. Il est basé sur un sous-ensemble du langage de programmation [JavaScript](http://www.crockford.com/javascript) ([JavaScript Programming Langage, Standard ECMA-262 3rd Edition - December 1999](http://www.ecma-international.org/publications/files/ecma-st/ECMA-262.pdf)). JSON est un format texte complètement indépendant de tout langage, mais les conventions qu'il utilise seront familières à tout programmeur habitué aux langages descendant du C, comme par exemple : C lui-même, C++, C#, Java, JavaScript, Perl, Python et bien d'autres. Ces propriétés font de JSON un langage d'échange de données idéal.

## Jersey Gson

Le développement de services Web RESTful qui prennent en charge de façon transparente d'exposer vos données dans une variété de types de supports de représentation et abstraite loin les détails de bas niveau de la communication client-serveur est pas une tâche facile sans une bonne boîte à outils. Afin de simplifier le développement des services Web RESTful et leurs clients en Java, une API JAX-RS standard et portable a été conçu.

Cadre Jersey RESTful Web Services est open source, la qualité de la production, cadre pour le développement RESTful Web Services en Java qui prend en charge JAX-RS API et sert de JAX-RS (JSR 311 et JSR 339) Référence mise en œuvre.

Cadre Jersey est plus que la mise en œuvre JAX-RS Référence. Jersey fournit son propre API qui étend la JAX-RS boîte à outils avec des fonctionnalités et utilitaires supplémentaires pour simplifier davantage le service RESTful et le développement de la clientèle. Jersey expose également de nombreux SPIs de vulgarisation afin que les développeurs puissent étendre Jersey pour mieux répondre à leurs besoins.

Objectifs du projet Jersey peuvent être résumés dans les points suivants:

* + Suivre l'API JAX-RS et de fournir des mises à jour régulières des implémentations de référence de la qualité de la production que les navires avec GlassFish.
  + Fournir des API pour étendre Jersey et construire une communauté d'utilisateurs et de développeurs.
  + et enfin il est facile de créer des services Web RESTful utilisant Java et la machine virtuelle Java