UNIVERSITE CHEIKH ANTA DIOP DE DAKAR

FACULTE DES SCIENCES ET TECHNIQUE

DEPARTEMENT MATHS-INFORMATIQUE

Laboratoire d'Algèbre, de Cryptologie, de Géométrie Algébrique et Applications (L.A.C.G.A.A)

MASTER II EN TRANSMISSION DES DONNEES ET SECURITE DE L'INFORMATION (TDSI)
MEMOIRE DE FIN DE CYCLE

Thème : Conception et implémentation d'une application E-Commerce

Réalisé par DIOP Fatima Binetou Yarassoul, sous la direction de M. Ousmane NDIAYE, Enseignant à l'UCAD

Présenté et soutenu le 04 juin 2015 devant le jury :

Président du jury : Pr Oumar DIANKHA, Professeur à L'UCAD

Membres du jury :

Dr Demba SOW, Enseignant à L'UCAD

M. Ahmed KHALIFA, Enseignant à L'UCAD

M. Heikreo ADJEOUA, Enseignant à L'UCAD

M. Ousmane NDIAYE, Enseignant à L'UCAD, Encadreur

ANNEE UNIVERSITAIRE 2013-2014

DEDICACES

Je dédie ce travail à :

- Prophète Mouhamad (PSL), ma référence.
- Sa fille Fatima Binetou Rassoul, mon idole.
- Ma mère, pour sa bravoure, sa générosité, sa bienveillance et sa bonne croyance.
- Mon père, pour sa bienfaisance, sa foi inébranlable et sa compassion envers les autres. Il n'a jamais cessé de nous encourager dans nos études et dans la bonne pratique musulmane.
- Mon oncle, pour son soutien financier et moral durant tout mon cursus scolaire, pour sa sérénité, son savoir-faire et sa bonté.
- Ma jumelle, mes frères, sœurs, cousins, cousines, neveux et nièces que j'aime de tout cœur.
- Mon homonyme ; qu'Allah lui apporte longévité, quiétude et plénitude et lui accorde toutes les bonnes attitudes et celles de Binetou Rassoul !

REMERCIEMENTS

Tout d'abord, je rends grâce à Dieu, le tout miséricordieux.

J'adresse mes remerciements les plus sincères aux personnes qui ont apporté leur soutien financier ou moral pour la réussite de ce travail et de mon cursus scolaire.

Ainsi, je remercie en premier lieu mes parents (ma mère, mon père et mon oncle) qui n'ont jamais cessé de m'encourager, de me soutenir.

En second lieu, je remercie :

- mon encadreur, M. Ousmane NDIAYE, pour sa disponibilité et ses encouragements tout au long de la réalisation de ce mémoire ;
- M. Oumar DIANKHA, Professeur à L'UCAD, qui m'a vraiment soutenu, ainsi que M. MASSALY ; jamais j'oublierai la faveur, acte noble, qu'ils m'avaient donnée ;
- les membres du jury, ainsi que toute l'équipe de la TDSI.

Enfin, je remercie ma jumelle, tous mes frères, sœurs, cousins, cousines et amis.

RESUME

Nous avons constaté que les habillements féminins décents deviennent de plus en plus rares dans notre pays ; les femmes sénégalaises accordent beaucoup plus d'importances à la culture occidentale, ce qui est contradictoire à notre religion. A cet effet, nous avons pensé concevoir quelque chose qui pourra leur inciter à mélanger les deux, autrement dit, s'habiller avec élégance tout en respectant les chartes de notre religion.

Pour ce faire, l'internet serait un bon outil pour pouvoir s'ouvrir à eux. Dans ce sens, nous avons pensé concevoir et implémenter une application de commerce électronique (E-Commerce) permettant de vendre via internet des articles féminins tels que le voile, des bodys, robes, chemises, et tant d'autres articles, très en vogue et très décents.

Cependant, la majorité des applications web souffrent de leur vulnérabilité. La sécurité des applications web dépend de nombreux facteurs, tels que les connaissances du développeur mais aussi la conception et la complexité du site, le coût et les délais de fabrication et même parfois aussi étonnant que cela puisse paraître de la responsabilisation de ses utilisateurs.

Ainsi, nous avons utilisé la technologie Java EE pour la réalisation de notre application et un serveur appelé Glassfish pour gérer la sécurité tout en tenant compte des attaques qui existent.

TABLE DES MATIERES

LISTE DES FIGURES

INTRODUCTION

Dans le cadre de l'obtention du diplôme de Master en Transmission des Données et Sécurité de l'Information (TDSI), nous avons eu à faire un travail personnel afin de concrétiser une bonne partie de nos connaissances acquises durant la formation.

Vu les évolutions rapides des nouvelles technologies, l'informatique est devenue un facteur incontournable dans tous les domaines. L'informatique n'est pas simplement une science de bits, mais elle implique les réseaux, les langages de programmations, les bases de données, entre autres. En occurrence, l'internet est une interconnexion de réseaux permettant d'avoir une bonne visibilité à travers le monde et d'avoir une mobilité facile.

Par ailleurs, nous avons constaté que les habillements féminins décents deviennent de plus en plus rares dans notre pays ; les femmes sénégalaises accordent beaucoup plus d'importances à la culture occidentale, ce qui est contradictoire à notre religion. A cet effet, nous avons pensé concevoir quelque chose qui pourra leur inciter à mélanger les deux, autrement dit, s'habiller avec élégance tout en respectant les chartes de notre religion.

Pour ce faire, l'internet serait un bon outil pour pouvoir s'ouvrir à eux. Dans ce sens, nous avons pensé concevoir et implémenter une application de commerce électronique (E-Commerce) permettant de vendre via internet des articles féminins tels que le voile, des bodys, robes, chemises, et tant d'autres articles, très en vogue et très décents.

Après quelques mois de recherche, nous avons eu à réaliser un travail répondant à nos aspirations. Nous allons vous présenter dans la première partie le cadre de référence et l'approche méthodologique ; dans la deuxième partie l'approche technique et enfin dans la troisième partie l'aspect sécuritaire.

CHAPITRE 1 : REFERENCES ET APPROCHE METHODOLOGIQUE

1.1 REFERENCES

1.1.1 Présentation générale du laboratoire LACGAA

Le laboratoire LACGAA a pour objectifs :

- La formation à la recherche fondamentale et appliquée dans les domaines de la Cryptographie, de la Théorie des codes, de l'Algèbre, de la Géométrie et de leurs applications (en logique, en informatique, en sécurité de l'information, en biologie, en robotique etc.) par :
  - des enseignements pour les jeunes doctorants durant leur première année d'inscription en thèse ;
  - l'encadrement des jeunes doctorants durant toute la durée de leur thèse ;
  - la mise en place d'un cadre approprié pour l'épanouissement des jeunes doctorants.
- L'organisation de la recherche par la mise en place d'un cadre approprié pour l'épanouissement des chercheurs et le développement de la recherche.
- La création de licences et de masters professionnels et filières de recherches en algèbre, géométrie et leurs applications, notamment en sécurité informatique.

Ses principaux domaines de recherche sont l'algèbre et ses différentes applications :

- Algèbre commutative, Algèbre non commutative, Algèbre associative, Algèbre non associative.
- Géométrie algébrique commutative et non commutative, Homologie et Co-homologie, Théorie algébrique et Analytique des nombres.
- Cryptographie, Théorie des Codes correcteurs d'erreurs, Théorie du signal.
- Informatique théorique, Sécurité informatique etc.

Le Directeur du laboratoire se nomme Professeur Mamadou SANGHARE.

1.1.2 Présentation du TDSI

Depuis 2004, le laboratoire LACGAA est le seul de la sous-région spécialisé dans la formation et la recherche en cryptographie et dans les domaines de la sécurité de l'information.

Fort d'une expérience de 8 ans, le LACGAA a déjà ouvert une licence et un master en Transmission de Données et Sécurité de l'Information (TDSI), et une formation doctorale.

Le laboratoire LACGAA a déjà formé en master, plus de 100 titulaires du master2 (niveau ingénieur) qui travaillent dans les entreprises en France, aux Etats-Unis, au Sénégal et dans la sous-région; et en licence, plus de 40 techniciens.

En Thèse, le laboratoire est en train de former plus de 10 thèses en Codage et Cryptologie à Dakar et en France dont 5 sont déjà terminés.

Ainsi, il y a des conditions d'admission pour les nouveaux bacheliers, les étudiants ou les travailleurs :

- Licence en Transmission de Données et Sécurité de l'Information L1 : Titulaire d'un Bac scientifique.
- Licence en Transmission de Données et Sécurité de l'Information L3 : Titulaire d'un Bac + 2 ans.
- Master en Transmission de Données et Sécurité de l'Information : Licence (bac+3 ans) scientifique.

1.2 APPROCHE METHODOLOGIQUE

1.2.1 Problématique

1.2.1.1 Définition du Commerce Electronique

On appelle « Commerce électronique » (ou E-Commerce) l'utilisation d'un média électronique pour la réalisation de transactions commerciales. La plupart du temps, il s'agit de la vente de produits à travers le réseau Internet, mais le terme de E-Commerce englobe aussi les mécanismes d'achat par Internet.

1.2.1.2 Importance du Commerce Electronique

En bref le E-Commerce est une nouvelle pratique de vente de biens et de services en ligne sur Internet.

Ce moyen de distribution facilite les transactions sur Internet et permet la création et le développement de relations en ligne.

Il est utilisé par certains grands commerçants locaux exposant leurs produits afin d'échanger avec un plus large public. Ainsi, les ménages peuvent acheter sans se déplacer, à tout moment de la journée et en quelques secondes.

Avec l'achat en ligne, fini le stress des cabines d'essayage, la contrainte des fermetures des boutiques à 19H, les difficultés à se garer, les longues attentes devant les caisses, etc. Il suffit de s'installer confortablement devant son écran d'ordinateur pour dénicher de belles affaires.

Il favorise aussi les échanges dans le monde entier.

1.2.1.3 Commerce Electronique au Sénégal

Le commerce électronique mondial bat son plein. Le Sénégal, à l'instar des pays africains, n'a pas encore fait le saut. Il est entravé par de nombreuses lianes qu'il hésite à couper. Les initiatives individuelles ou de groupes, aussi pertinentes qu'elles soient, n'arrivent pas à prospérer du fait de la frilosité de nombreux autres acteurs qui s'arc-boutent sur des profits, les yeux fermés. Mais l'assaut du marché mondial par le commerce électronique est tel qu'il n'est plus possible de résister longtemps. L'alternative est de s'y lancer ou de disparaître derrière le concept, une foultitude d'activités.

Le E-Commerce n'existait pas il y a 15 ans alors qu'il représente aujourd'hui un chiffre d'affaire de 144 Milliards d'euros dans le monde, valeur qui est en constante évolution. Quelle est la part de l'Afrique dans ce nouveau moyen de commerce international, quelle est la part du Sénégal ? Des questions qui paraissent très intéressantes mais dont les réponses sont malheureusement sans ambiguïté. Il n'y a pas vraiment de chiffres officiels mais on peut quand

même dire avec certitude que la part de l'Afrique est minime. Pourquoi un tel état de faits compte tenu de la qualité des ressources et du potentiel dont dispose l'Afrique ?

Le E-commerce est simplement l'utilisation de la technologie internet dans le cadre d'achats et de ventes de biens, de marchandises ou de services. Quand on voit la part qu'occupe le commerce dans les activités économiques des Sénégalais, il est légitime de se demander pourquoi nos commerçants et hommes d'affaires ne s'approprient-ils pas les formidables outils que propose le web pour faire décoller leurs activités.

La première chose à se demander est la problématique de la vulgarisation du net dans le Sénégal. Il est certain que malgré les offres de moins en moins chères proposées par les fournisseurs d'accès du Sénégal, internet reste inconnu pour une certaine catégorie de la population. En effet pour papa « Goorgorlou » qui se tue tous les jours pour trouver de quoi nourrir sa famille dans un pays où la misère ne cesse d'augmenter, acheter un ordinateur est loin d'être une priorité : c'est un luxe qu'on laisse aux familles qui ont déjà pu résister à l'impérieux appel de la faim. S'il faut en plus de cela payer un abonnement mensuel qui s'ajoute aux dépenses liées à l'eau et à l'électricité, « Goorgorlou » n'est plus en mesure de s'en tirer d'affaire. Il est clair donc que malgré l'absence de chiffres explicites sur la pénétration d'internet dans les foyers sénégalais, la majorité de notre population n'y a pas accès ou du moins n'a pas un accès régulier et facile à cet outil.

Un deuxième point à relever est l'absence de services et produits proposés dans le web sénégalais. C'est en quelque sorte le serpent qui se mord la queue : je n'achète pas de produits sénégalais sur internet car il n'y en a tout simplement pas. De l'autre côté, je ne propose pas de produits sur internet car il n'y a pas d'acheteurs. Pourtant, le marché est bien là. Plusieurs aventuriers ont déjà tenté de se lancer à l'assaut même si peu ont réussi.

Le business numérique n'est pas le fort de « Gorgorlou », vendeur de tissus au marché HLM ou de « Diekk », grande importatrice de marchandises plus ou moins légales. L'informel emploie beaucoup plus de personnes que le formel dans les pays sous-développés. Le constat est donc simple, c'est le même que pour « Gorgorlou » qui a autre chose à faire que de prendre un abonnement internet pour sa famille. La plus grande partie des échanges de marchandises se

fait de manière informelle et pour ces types d'échanges, utiliser internet comme outil de marketing n'est pas forcément la première chose à laquelle on pense.

Un autre point important est le fait que les entreprises Leader du Sénégal ne prennent pas leur responsabilité pour tenter de susciter un nouveau type de comportement chez les clients sénégalais. Un leadership fort de la part d'entreprises reconnues est souvent un moteur et un exemple pour faire évoluer un marché. L'une des entreprises les plus modernes et plus riches du Sénégal a l'un des sites internet les moins ergonomiques et les moins fonctionnels. Ces concurrents ne sont pas mieux lotis. Le nouveau client ne peut presque pas souscrire à une offre, encore moins payer en ligne.

Les contraintes techniques constituent des blocages essentiels qui mettent du plomb dans les ailes du E-Commerce :

- Les compétences

Sur le plan des compétences techniques dans le domaine des nouvelles technologies, le Sénégal n'a absolument rien à envier à aucun pays au monde. Chaque année des centaines et des centaines d'ingénieurs sénégalais sortent des plus grandes écoles du Sénégal et du monde. Ces mêmes Sénégalais qui aujourd'hui occupent des postes de responsabilités dans des missions et projets d'envergures internationales. Sous ce rapport, les ressources humaines douées et préparées ne manquent pas au Sénégal, elles demandent juste à être employées dans l'intérêt du pays.

- Les infrastructures techniques

C'est sans doute l'une des clés principales du problème. C'est bien de parler de l'utilisation des nouvelles technologies, de la création de sites marchands à profusion mais s'il n'y a pas des infrastructures techniques accessibles à tous et de manière démocratique, tous les efforts fournis ne serviraient à rien.

- Le problème des plateformes de paiement électronique

Il est pratiquement impossible de parler de commerce sans parler de flux monétaires, d'échanges économiques qui en découlent, de moyens de paiements. Dans le cas de l'E-Commerce, il est important d'avoir des outils monétiques adaptés qui permettront aux clients d'acheter et de faire leurs paiements sans soucis de sécurité et aux fournisseurs de gérer leurs comptes et de fournir des services en ligne. Inutile de dire qu'au Sénégal on est encore très loin du compte dans ce domaine. Déjà les comptes bancaires ne sont pas aussi nombreux qu'on le voudrait. Dans de nombreux pays un jeune de dix-huit ans devrait pouvoir avoir un compte, ne serait-ce qu'avec 10.000 Francs au maximum. Cela permettrait aux jeunes de s'adapter très tôt au processus bancaire et à la gestion de l'argent à travers un compte bancaire et oublier un peu la tirelire.

C'est vrai qu'il y a de très grandes banques au Sénégal et qu'au moins dans ce secteur la concurrence est vive mais il faudrait aussi remarquer que ces mêmes banques ne font pas grand-chose pour doter la majorité de la population de moyens de paiements efficaces. On voit quelques GAB par ci ou par là mais, il est très rare de voir le mode de paiement par carte bancaire dans les entreprises et les commerces. Inutile donc de dire que le moyen privilégié de « Gorgorlou » est le paiement par liquidité ou parles chèques, à défaut. On se demande pourquoi les banques ne s'impliquent pas plus dans le changement, dans l'adoption des nouveaux moyens de paiement par « Gorgorlou » d'autant plus qu'ils seraient les premiers à en profiter.

Le développement du E-Commerce passe donc par le développement de plateformes bancaires plus modernes et accessibles à tous. Encore une fois l'expertise technique est là, il faudrait juste une volonté conjointe des banques et de l'état. Une volonté qui est nécessaire pour permettre d'abord aux Sénégalais d'accéder à des technologies et techniques simples utilisées partout dans le monde, volonté qui permettra aussi de propulser un nouveau type de comportement dans les échanges par le moyen des nouvelles technologies : le E-Commerce.

Tout au moins le peu de sites E-Commerce qui existent n'ont pas un développement assez fulgurant leur permettant de se faire remarquer et d'avoir la confiance des consommateurs.

L'E-Commerce nécessite la mise en place d'une politique de sécurité irréprochable, vu la vulnérabilité du web. De manière générale, au Sénégal, les blocages qu'on pourrait citer concernent plusieurs domaines tels que :

- La sécurité : Confier son numéro de code bancaire à un site web avec un back-end douteux est un grand risque. En plus, avec la technologie, personne n'est exempt de failles. Donc un système de paiement assez fiable devrait être employé.
- La disponibilité des produits : ce serait assez outrageux d'exposer des produits dont on n'a pas la possession.
- La clientèle : la plupart des gens n'ont pas tellement confiance à la technologie et donc mettre une solution E-Commerce avec une bonne clientèle relève beaucoup de miracle.

1.2.1.4 Objectif de notre application de Commerce Electronique

- Utiliser l'Internet pour vulgariser, à travers le monde, nos produits disponibles dans notre boutique.
- Réduire les dépenses de marketing global.
- Fournir de l'information instantanée à nos prospects et clients en permettant de voir les descriptions des produits et informations sur les prix à partir de notre site.
- Inciter les femmes musulmanes à s'habiller décemment.
- Utiliser les nouvelles technologies pour montrer que la femme musulmane en a une bonne part.
- Gérer l'aspect sécuritaire de notre application.
- Améliorer le service clientèle.
- Faire de sorte que notre boutique ait une bonne réputation mondiale.
- Etc.

1.2.2 Etude sur le marché

1.2.2.1 Décret sur le Commerce Electronique au Sénégal

Le présent projet de décret est en application des dispositions de la loi n° 2008-08 du 25 janvier 2008 sur les transactions électroniques.

Les précisions apportées ont trait notamment :

- ✓ aux activités liées au commerce électronique ;
- ✓ aux obligations d'information du fournisseur électronique de biens ou de services ;
- ✓ à la responsabilité contractuelle du fournisseur électronique de biens ou de services ;
- ✓ à la publicité par voie électronique ;
- ✓ aux contrats sous forme électronique.

Telle est l'économie du présent projet de décret.

En voici quelques articles du décret :

Article 8 : Toute personne, qui exerce l'activité définie aux articles 8 et 10 de la loi sur les transactions électroniques, a l'obligation de fournir au consommateur les informations suivantes :

1) le cas échéant, le nom du directeur de publication ;
2) une adresse électronique et postale pour des réclamations éventuelles ;
3) un numéro de téléphone ou de fax ;
4) les indications sur les dispositions relatives à la protection des données à caractère personnel ;
5) les caractéristiques essentielles du produit ou du service proposé ;
6) le prix du bien ou du service avec toutes les taxes comprises ;
7) le cas échéant, les frais de livraison ;
8) la durée de validité de l'offre ;
9) la monnaie de facturation, les modalités de paiement, de livraison ou d'exécution et le cas échéant, les conditions de crédit proposées ;
10) les conséquences d'une mauvaise exécution ou d'une inexécution des engagements du fournisseur ;
11) l'existence ou l'absence d'un droit de rétractation ;
12) les informations relatives aux services après-vente et aux garanties commerciales existantes ;

13) les conditions relatives à la conclusion (la date et l'heure), à la durée et la résiliation des contrats en ligne ;
14) le mode de remboursement des sommes versées par le consommateur en cas de rétractation de sa part ;
15) le coût de l'utilisation d'un service en ligne ;
16) les conséquences de l'absence d'une confirmation des informations relatives aux prestations en ligne.

Ces informations doivent être non équivoques, lisibles, d'un accès facile et permanent à partir de la page d'accueil du site web du fournisseur électronique de biens ou de services.

Article 9 : Pour s'assurer que le consommateur a bien pris connaissance des obligations auxquelles il a souscrit, le fournisseur électronique de biens ou de services doit mettre à sa disposition les informations mentionnées à l'article 24 de la loi sur les transactions électroniques.

Ces informations doivent être accessibles et reproduites, en cas de besoin, par le consommateur en vue de leur conservation.

Article 10 : Les informations mentionnées à l'article 8 du présent décret doivent être fournies par tout moyen adapté au service utilisé et accessibles à tout stade de la transaction, dans le respect des principes qui régissent la protection des personnes frappées d'incapacité juridique, notamment les mineurs et les incapables.

Lorsqu'il est en mesure de le faire, le fournisseur électronique de bien ou de services doit mettre en place un service permettant au consommateur de dialoguer directement avec lui.

Article 12 : Pour tout contrat conclu par voie électronique, le consommateur dispose d'un délai de sept jours ouvrables pour se rétracter, sans indication de motif et sans pénalités.

Toutefois, si le fournisseur électronique de biens ou de services n'a pas satisfait aux obligations d'information prévues à l'article 10 de la loi sur les transactions électroniques, le délai de rétractation est de trois mois.

Article 24 : Quiconque propose, à titre professionnel, par voie électronique, la fourniture de biens ou la prestation de services, met à la disposition de la clientèle les conditions contractuelles applicables d'une manière qui permette leur conservation et leur reproduction. Sans préjudice des conditions de validité mentionnées dans l'offre, son auteur reste engagé par elle tant qu'elle est accessible par voie électronique de son fait.

1.2.2.2 Avantages du Commerce Electronique

Une solution E-Commerce permet à un vendeur disposant d'une plateforme adéquate de mettre ses produits à la disposition du public via internet et d'en recevoir le paiement par ce même cheminement. Il ne se charge pas contrairement au e-business de l'aspect fidélisation du client par divers moyens qui seront mis en place, mais se contente juste de mettre en pratique les fonctionnalités liées au panier du client sur la plateforme.

Pour un marché désordonné comme le nôtre où les mêmes produits sont rarement aux mêmes places, ce genre de solution serait carrément un bonheur parce qu'alliant rapidité des recherches, économie du déplacement et peut être même d'autres avantages. Au Sénégal, un marché peut pousser tellement vite qu'on ne s'en rendrait même pas compte et avec des produits très diversifiés, une plateforme E-Commerce serait assez souhaitable par tout le monde.

Il serait intéressant de connaître les raisons qui peuvent inciter un acheteur à passer la commande sur l'internet. A l'heure actuelle, nombreux sont les individus qui cherchent à réaliser des économies. Le moindre centime est bon à prendre. En faisant leurs emplettes sur Internet, et c'est encore plus vrai par ces temps de crise économique, les consommateurs recherchent bien entendu les prix bas avec les nombreuses promotions proposées toute l'année, et cherchent à économiser les frais d'essence avec des frais d'envoi peu élevés, voire gratuits. Mais le prix n'est pas le seul argument. Les magasins en ligne présentent également l'avantage d'être accessibles à tout moment. Reste maintenant le côté immatériel du commerce électronique qui freine certains consommateurs qui aiment bien voir avant d'acheter. Bien souvent, ils se servent d'Internet pour comparer les articles et les prix, pour finalement se tourner vers les magasins.

De jour en jour, et à un rythme accéléré, le Commerce Electronique révolutionne les canaux de ventes habituels en offrant aux entreprises un accès à de nouveaux marchés, ainsi qu'un moyen inégalé pour fidéliser et satisfaire les attentes de leurs clients existants tout en leur permettant d'optimiser leur activité de vente et de recouvrement.

D'autres principaux avantages se présentent aussi :

- Gagner de nouvelles parts de marché :

Un site web est une vitrine commerciale accessible 24h/24 et 7j/7. Un site web est un investissement rentable car il vous assure une présence commerciale dans le monde entier.

Vous pouvez ainsi conquérir de nouvelles parts de marché, sans pour autant disposer d'une présence physique ou d'une agence commerciale. Votre site vous permet d'acquérir de nouveaux clients et les fidéliser, de réaliser de nouvelles ventes aux clients existants et de développer de nouveaux marchés.

Le web est, par conséquent le moyen de communication que vous devez impérativement intégrer dans votre stratégie marketing.

- Vendre vos produits en direct :

Le nombre de sites marchands connaît une croissance exponentielle. Même les analystes les plus pessimistes annoncent des chiffres incroyables. Cette explosion émane aussi bien des grandes entreprises que des plus petites entités.

- Offrir à votre client le meilleur choix et service :
    - Réduction des délais de vente et de livraison.
    - Facilité de traitement des commandes ou des paiements.
    - Visibilité sur les produits et les achats.

Votre client, choisit d'une manière simple et rapide son produit ou service, remplit un bon de commande et paie en toute sécurité. Quelques jours plus tard, il recevra son achat ou pourra accéder immédiatement au service ou à l'information acheté.

Ainsi, votre client gagne du temps et économise de l'argent : ses randonnées virtuelles lui permette de trouver facilement ce qu'il cherche et lui évite les embouteillages, et les interminables files d'attente aux caisses. En plus, il aura accès à des produits et services innovants et à des prix attractifs.

1.2.2.3 Types de Commerce Electronique

Quatre principaux types de Commerce Electronique sont identifiés par les professionnels. Il s'agit du « Business to Business (B2B) » qui renvoie aux échanges entre entreprises, le « Business to Consummers (B2C) » concernant les activités commerciales entre des entreprises et des particuliers, le « Consummers to Consummers (C2C) » pour les échanges entre des particuliers et le « Business to Government (B2G) » relatif aux appels d'offres publics électroniques.

Le B2B est l'une des formes les plus anciennes du Commerce Electronique puis qu'elle existe depuis 1970 avec les échanges de données informatisées (Edi). Le Commerce Electronique B2B est, de même, la forme la plus importante parmi toutes celles qui existent. Selon les services e-business du Gartner Group, le marché mondial du commerce électronique B2B va passer de 145 Milliards de dollars en 1999 à 7,29 Trillions de dollars en l'an 2004. Il représentera alors 7% des 105 Trillions de dollars américains de l'ensemble des transactions électroniques. L'Afrique en général et le Sénégal en particulier, sont en marge de cette activité. Selon les spécialistes, le marché du commerce électronique à la plus forte croissance en Amérique du nord. Les principales contraintes pour le B2B, comme pour tous les autres types de E-Commerce, demeurent la connectivité, les livraisons et la volonté d'acheter en ligne.

1.2.2.4 Choix de notre application de Commerce Electronique.

Le Sénégal est un pays où plus de 90% de la population sont des musulmans. Ainsi, avec la colonisation plusieurs aspects impactent sur notre éducation. En effet, avec la mondialisation, les habitudes, les comportements, la culture et l'éducation des occidentaux sont imités par la plupart des Sénégalais alors que certains d'entre eux sont interdits par notre religion musulmane. Parfois, certains personnes en sont conscientes mais ne peuvent pas s'en échapper à causes des influences. Ainsi des comportements vicieux sont en passe d'être considérés

comme de la mode. C'est pourquoi nous avons pensé les encourager à respecter la religion tout en restant dans les tendances de la mode. Certes, il existe dans le marché des produits vestimentaires décents pour les femmes mais ils sont rares, disponibles que durant une période déterminée. Ces dits produits sont souvent importés et sont vendus chers. Ainsi notre boutique leur aidera à avoir un accès facile aux produits.

CHAPITRE 2 : APPROCHE TECHNIQUE

2.1 LES LANGAGES DE PROGRAMMATION

2.1.1 Généralités

Un langage de programmation est un code de communication, permettant à un être humain de dialoguer avec une machine en lui soumettant des instructions et en analysant les données matérielles fournies par le système, généralement un ordinateur. Le langage permet à la personne qui rédige un programme, de faire abstraction de certains mécanismes internes, généralement des activations et désactivations de commutateurs électroniques, qui aboutissent au résultat désiré.

L'activité de rédaction du code source d'un programme est nommée programmation. Elle consiste en la mise en œuvre de techniques d'écriture et de résolution d'algorithmes informatiques, lesquelles sont fondées sur les mathématiques. À ce titre, un langage de programmation se distingue du langage mathématique par sa visée opérationnelle (une fonction et par extension, un programme, doit retourner une valeur), de sorte qu'un langage de programmation est toujours un compromis entre la puissance d'expression et la possibilité d'exécution.

Les langages de programmation permettent de définir l'ensemble des instructions effectuées par l'ordinateur lors de l'exécution d'un programme. Il existe des milliers de langages de programmation, la plupart d'entre eux étant réservés à des domaines spécialisés.

2.1.2 Etudes comparatives des langages les plus utilisés

Les langages de programmation peuvent être regroupés par typologie :

- **Langages de programmation compilés :**
  - **Java**

Le Java est un langage de programmation orienté objet qui a été développé par Sun Microsystems dans les années 90. C'est l'un des langages de programmation les plus

populaires, devenu un standard chez les programmes d'entreprise, dans les contenus et jeux web ou encore dans les applications mobiles.

C'est également le langage de programmation utilisé pour le système d'exploitation mobile Android. Le Java est fait pour travailler sur de nombreuses plateformes : un programme codé sur Mac OS X pourrait par exemple fonctionner sur Windows.

- **Langage C**

Le langage C est le langage de programmation le plus vieux et le plus utilisé au monde. Il est créé dans les années 70 et fournit les bases de nombreux autres langages, comme le C#, le Java… Il est donc conseillé de s'y intéresser de près. Le C est principalement utilisé pour intégrer des systèmes d'opérations et des applications incrustées dans des pages web.

- **C++**

Le C++ est un langage de niveau intermédiaire qui possède des fonctions orientées objet. Il était à l'origine fait pour améliorer le langage C. Le C++ est le moteur de nombreux programmes très connus comme Firefox, Winamp ou encore ceux de la suite Adobe. Il est utilisé pour développer des applications, des systèmes d'exploitation, des serveurs et clients internet très performants ou encore des jeux vidéo.

- **C#**

Le C#, prononcé « C sharp », est un langage de programmation à paradigmes multiples développé par Microsoft via la .NET initiative. Il combine des principes du C et du C++ et est un langage généraliste utilisé pour développer des logiciels pour Microsoft et les plateformes Windows.

- **Objective**-C

L'Objective-C est un langage de programmation généraliste. Orienté objet, il est utilisé par les systèmes d'exploitation et applications Apple. Il fait fonctionner OS X et iOS ainsi que leurs interfaces de programmation. Il peut ainsi être utilisé pour développer des applications

iPhone, ce qui a généré un engouement massif pour ce langage de programmation qui a longtemps été considéré comme démodé.

- **Langages interprétés :**
  - **Python**

Le Python est un langage de programmation de haut niveau utilisé pour les sites Internet et les applications mobiles. Il est considéré comme un langage facile à maîtriser pour les débutants grâce à sa syntaxe compacte et à sa très bonne lisibilité, ce qui permet aux développeurs d'utiliser moins de lignes de code pour exprimer une fonction qui en aurait nécessité plus dans d'autres langages. Il fait fonctionner les applications web d'Instagram, Pinterest et est utilisé par Google, Yahoo et même la NASA.

- **JavaScript**

Le JavaScript est un langage de programmation développé par Netscapec, sa syntaxe est un dérivé du C. Il peut être utilisé d'un navigateur Internet à un autre et est considéré comme essentiel dans le développement de fonctionnalités web interactives et/ou animées. Il peut également être utilisé dans le développement de jeux. De nombreux « interprètes » du JavaScript sont directement intégrés dans les extensions de Chrome, Safari, Adobe Acrobat et la suite créative d'Adobe.

- **PHP**

Le PHP (pour Hypertext Processor) est un langage de programmation libre utilisé pour le développement de sites Internet et d'applications dynamiques. Il peut être directement incrusté dans des documents HTML plutôt que par un fichier externe, ce qui l'a rendu très populaire chez les développeurs web. Le PHP fait fonctionner plusieurs Millions de sites Internet (environ 200 Millions), notamment WordPress et Facebook.

- **Ruby**

Le Ruby est un langage orienté objet utilisé pour développer des sites Internet et des applications mobiles. Il est créé pour être simple et facile à écrire. Il fait fonctionner le Ruby on

Rails. Comme le Python, le Ruby est considéré comme étant un langage de programmation idéal pour les néophytes.

- **HTML**

L'Hypertext Markup Language, généralement abrégé HTML, est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte, c'est ce qui explique son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web. Il est souvent utilisé conjointement avec des langages de programmation (JavaScript) et des formats de présentation (feuilles de style en cascade). HTML est initialement dérivé du Standard Generalized Markup Language (SGML).

- **Langage pour les données :**
  - **SQL**

Le Structured Query Language (SQL) est un langage spécialisé pour gérer les données en relation avec une base de données de systèmes de management. Il est plus communément utilisé pour ses fonctions de requêtes, qui cherchent des informations dans les bases de données.

- **MATLAB**

MATLAB (matrix laboratory) est un langage de programmation de quatrième génération stimulé par un environnement de développement du même nom ; il est utilisé à des fins de calcul numérique. Développé par la société The Math Works, MATLAB permet de manipuler des matrices, d'afficher des courbes et des données, de mettre en œuvre des algorithmes, de créer des interfaces utilisateurs, et peut s'interfacer avec d'autres langages comme le C, C++, Java, et Fortran.

Le sondage s'est donc porté sur les langages les plus utilisés parmi les 1500 existants répertoriés par Wikipedia. Parmi la communauté Développez.com, les premiers langages se sont majoritairement portés sur : C (35%), Python et C++ viennent en seconde position à égalité (16%), puis vient Java (13%) et Ruby (6%). C#, Ocaml, HTML/CSS, PHP, et Matlab représentent moins de 12% de la totalité des résultats.

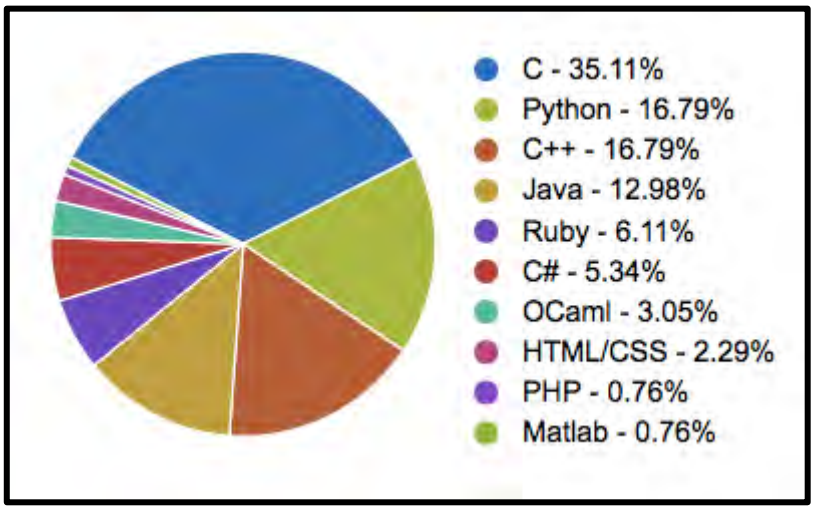


Figure 2-1 : Sondage sur les langages de programmation

2.2 JAVA ENTREPRISE EDITION (J2EE)

2.2.1 Généralités

Le terme « Java » fait bien évidemment référence à un langage, mais également à une plate-forme : son nom complet est « Java SE » qui signifie Java Standard Edition, et était anciennement raccourci en « J2SE ». Celle-ci est constituée de nombreuses bibliothèques, ou API.

Le terme « Java EE » signifie Java Enterprise Edition, et était anciennement raccourci en « J2EE ». Il fait quant à lui référence à une extension de la plate-forme standard. Autrement dit, la plate-forme Java EE est construite sur le langage Java et la plate-forme Java SE, et elle y ajoute un grand nombre de bibliothèques remplissant tout un tas de fonctionnalités que la plate-forme standard ne remplissait pas à l'origine.

J2EE est un ensemble de technologies, créées par SUN MicroSystems, permettant de réaliser des applications WEB mais aussi des applications standalones, en utilisant le langage JAVA.

Cette édition est dédiée à la réalisation d'applications pour les entreprises.

L'objectif majeur de Java EE est de faciliter le développement d'applications web robustes et distribuées, déployées et exécutées sur un serveur d'applications.

Elle est composée de deux parties essentielles :

- ✓ un ensemble de spécifications pour une infrastructure dans laquelle s'exécutent les composants écrits en java : un tel environnement se nomme serveur d'application.
- ✓ un ensemble d'API qui peut être obtenu et utilisé séparément. Pour être utilisées, certaines nécessitent une implémentation de la part d'un fournisseur tiers. Cet ensemble regroupe les servlets, JSP, services web, etc.

L'utilisation de J2EE pour développer et exécuter une application propose plusieurs avantages :

- ✓ une architecture d'application basée sur les composants qui permettent un découpage de l'application et donc une séparation des rôles lors du développement (modèle MVC) ;

- ✓ la possibilité de s'interfacer avec le système d'information existant grâce à de nombreuses API : JDBC, JNDI, JMS, etc.
- ✓ la possibilité de choisir les outils de développement et le ou les serveurs d'applications utilisés qu'ils soient commerciaux ou libres.

J2EE 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 différentes API de J2EE.

La maîtrise du langage Java constitue un prérequis dans le développement d'applications avec J2EE.

2.2.2 Etudes comparatives de quelques API fournis par J2EE

❖ JDBC

JDBC (Java Data Base Connectivity) est une API permettant de travailler avec des bases de données relationnelles. Elle permet d'envoyer des requêtes SQL à une base, de récupérer et d'exploiter le résultat. Elle permet aussi d'obtenir sur une même base des informations et les tables qu'elle comporte.

Le code Java utilisant l'API JDBC est indépendant de la base elle-même grâce à l'utilisation de drivers spécifiques fournis par les vendeurs. Bien sûr, les requêtes JDBC utilisées doivent être standards (et ne pas exploiter des fonctionnalités spécifiques à la base utilisée) pour que l'ensemble reste portable.

L'API JDBC est fournie en standard avec le JDK depuis la version 1.1 de Java. Les versions ultérieures fournissent de nouvelles fonctionnalités (comme la manipulation de résultats de requêtes comme des Java Beans, la gestion de pools de connexions, les traitements par batchs ou la sérialisation d'objets Java en base).

❖ RMI

RMI (Remote Method Invocation) est une API fournissant une approche de haut niveau de la programmation distribuée. On peut ainsi invoquer des méthodes d'un objet distant (résidant sur

un serveur) de la même manière que l'on appelle les méthodes d'un objet local. Cette API est présente dans le JDK standard depuis la version 1.1 et a été améliorée dans la version 1.2.

Si cette API est simple à mettre en œuvre pour le développeur, elle implique que le serveur (dans lequel résident les objets distribués) et les clients soient écrits en Java. Cette situation est maintenant assez commune pour que la mise en œuvre de RMI soit envisageable. Si ce n'est pas le cas, on préférera de se tourner vers une solution qui puisse être implémentée dans d'autres langages, comme CORBA.

- **Java IDL**

Comme nous l'avons vu ci-dessus, l'utilisation de RMI ne peut s'envisager que si le client et le serveur sont écrits en Java. Si ce n'est pas le cas, il est possible sur un objet distant en utilisant une solution basée sur CORBA (Common Object Request Broker Architecture), standard défini par l'OMG. La plateforme Java 2 inclue un ORB (Object Request Broker) permettant à un programme Java de communiquer avec d'autres ORBs et donc avec d'autres objets CORBA.

L'interface d'un objet CORBA est décrite dans un langage indépendant de la plateforme et du langage d'implémentation appelé IDL (Interface Description Language). Un compilateur IDL est fourni permettant de générer les classes nécessaires à un objet Java pour communiquer avec un ORB.

- **JNDI**

JNDI (Java Naming and Directory Interface) est une API pour communiquer avec les services de nommage et d'annuaire en réseau. On peut ainsi y chercher des objets Java par un chemin ou des valeurs d'attributs. Il existe des ponts avec les principaux services d'annuaires (comme LDAP, NIS ou NDS) et avec les registry de RMI ou CORBA.

Dans la pratique, JNDI est utilisé couramment dans la plateforme J2EE pour récupérer des objets par un nom symbolique (on peut ainsi récupérer une connexion à une base de données, une instance d'une interface distante, etc.).

- **EJB**

Les EJB (pour Enterprise Java Beans) sont des composants (au même titre que des Java Beans) destinés à tourner dans un serveur d'application EJB pour encapsuler des services de données ou logique métier. Il est, en effet, souvent intéressant de déporter le logique métier du client vers le tiers du milieu d'une application distribuée.

La valeur ajoutée des EJB réside dans les services fournis par le serveur. Le framework EJB est ainsi tenu d'assurer, de manière transparente, la sécurité, la persistance, le support réseau et la gestion des transactions aux composants. On enlève ainsi une épine du pied du développeur qui peut se consacrer pleinement à l'implémentation du logique métier.

L'API EJB et les services assurés par le serveur sont décrits dans des spécifications (dont la version 2.0 vient de paraître). Les EJB sont donc plus qu'une simple API, ils forment aussi un framework pour objets métiers. Ils forment une pièce maîtresse de la plateforme J2EE à tel point qu'on identifie parfois J2EE aux EJB.

- **Servlets**

Les Servlets peuvent être comparées à des Applets côté serveur : ce sont des objets tournant sur un serveur pour répondre aux requêtes du client de manière dynamique. Les Servlets sont appelées à remplacer les scripts CGI.

Les avantages des Servlets par rapport à d'autres technologies sont : la portabilité (entre les systèmes d'exploitation et les serveurs), la persistance entre les requêtes qui leur donne un avantage en terme de performance par rapport à d'autres technologies comme les CGI, et enfin l'accès à l'ensemble de la plateforme Java (qui leur permet ainsi d'accéder aisément à des bases de données avec JDBC).

- **JSP**

Les JSP (Java Server Pages) sont comparables aux ASP de Microsoft : ce sont des pages HTML comportant du code imbriqué. Elles rendent les mêmes services que des Servlets mais elles présentent l'avantage d'être beaucoup plus proche du document HTML que du code Java.

On peut ainsi en confier l'écriture à des « designers » web puis envoyer le résultat à des développeurs pour y insérer les appels au code Java.

Les JSP sont compilées automatiquement, lors du premier appel, en Servlets. De plus, les serveurs gérant les Servlets sont souvent capables de servir des JSP.

❖ JMS

JMS (Java Message Service) est une API d'échange asynchrone de messages ou d'évènements critiques entre applications. Comme JNDI et JDBC, JMS est une API construite pour reposer sur des services de messagerie existant fournis par divers vendeurs.

JMS permet maintenant de gérer les transactions et est utilisée pour la communication asynchrone entre EJB (des EJB pilotés par messages ou message driven beans, nouveauté de la version 2.0 des spécifications EJB).

❖ JTA

JTA (Java Transaction API) est une API permettant de gérer les transactions distribuées. Elle utilise un service de gestion des transactions distribuées avec lequel elle communique à travers l'API XA (standard défini par l'Open Group).

L'utilisation directe de l'API JTA reste cependant complexe, et les serveurs d'applications gèrent les transactions de manière transparente pour l'utilisateur. JTA peut donc être vue comme une API bas niveau utilisée par les développeurs de serveurs d'applications plutôt que par les développeurs d'applications d'entreprise.

❖ Autres API

Les implémentations J2EE doivent aussi fournir un certain nombre d'autres API, parmi lesquelles :

- **JavaMail** : permet d'envoyer des e-mails et doit inclure aussi Java Activation Framework (JAF).

- **JAXP** : Java API for XML Parsing est une API qui unifie les différentes implémentations de parsers XML (parsers SAX, DOM et processeurs XSLT).
- **JCA** : Java Connector Architecture permet l'interconnexion d'une application J2EE avec un système d'information d'entreprise par la gestion de pools, des transactions et de la sécurité.
- **JAAS**: Java Authentication and Authorization Service fournit une implémentation Java du standard PAM (Plugable Authentication Module).

2.3 LA CONCEPTION DES SYSTEMES D'INFORMATIONS

2.3.1 Merise

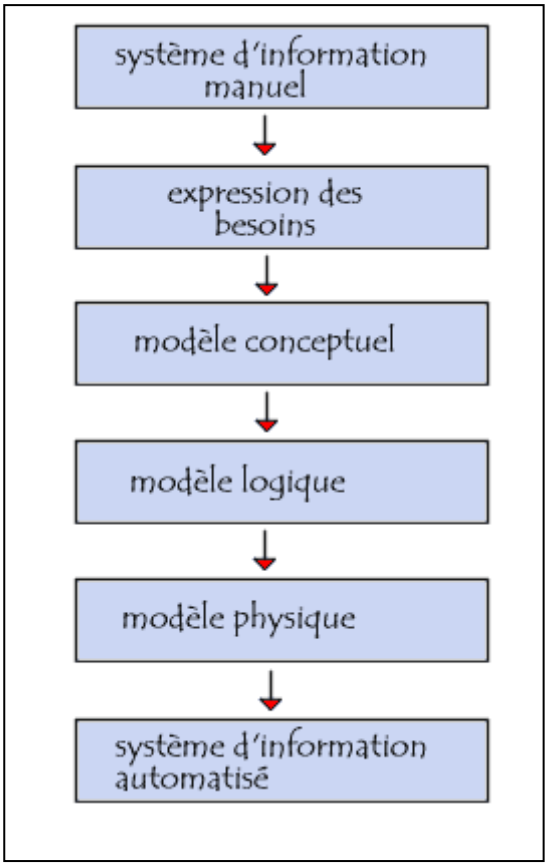
Merise (prononcée « Meurise ») est une méthode d'analyse, de conception et de gestion de projet informatique. Le but de cette méthode est d'arriver à concevoir un système d'information. La méthode Merise est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques.

La séparation des données et des traitements assure une longévité au modèle. En effet, l'agencement des données n'a pas à être souvent remanié, tandis que les traitements le sont plus fréquemment.

Merise a été très utilisée dans les années 1970 et 1980 pour l'informatisation massive des organisations. Cette méthode reste adaptée pour la gestion des projets internes aux organisations, se limitant à un domaine précis. Elle est en revanche moins adaptée aux projets transverses aux organisations, qui gèrent le plus souvent des informations à caractère sociétal (environnemental et social) avec des parties prenantes.

La conception du système d'information se fait par étapes, afin d'aboutir à un système d'information fonctionnel reflétant une réalité physique. Il s'agit donc de valider une à une chacune des étapes en prenant en compte les résultats de la phase précédente. D'autre part, les données étant séparées des traitements, il faut vérifier la concordance entre données et traitements afin de vérifier que toutes les données nécessaires aux traitements sont présentes et qu'il n'y a pas de données superflues.

Cette succession d'étapes est appelée cycle d'abstraction pour la conception des systèmes d'information :



2.3.2 UML

Le langage de modélisation unifié ou Unified Modeling Language (UML), est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Etant un langage graphique, il est couramment utilisé en développement logiciel et en conception orientée objet. Cette technique améliore la modélisation des Systèmes d'information.

UML est le résultat de la fusion de précédents langages de modélisation objet : Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard adopté par l'Object Management Group (OMG).

❖ Les points forts d'UML

UML est un langage formel et normalisé :

- ✓ il a un gain de précision ;
- ✓ il est un gage de stabilité ;
- ✓ il encourage l'utilisation d'outils ;

UML est un support de communication performant ;

- ✓ il cadre l'analyse ;
- ✓ il facilite la compréhension de représentations abstraites complexes ;
- ✓ son caractère polyvalent et sa souplesse en font un langage universel.

❖ Les points faibles d'UML

La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation. Même si l'espéranto est une utopie, la nécessité de s'accorder sur des modes d'expression communs est vitale en informatique. UML n'est pas à l'origine des concepts objets, mais en constitue une étape majeure, car il unifie les différentes approches et en donne une définition plus formelle.

Le processus (non couvert par UML) est une autre clé de la réussite d'un projet. Or, l'intégration d'UML dans un processus n'est pas triviale et l'amélioration d'un processus est une tâche complexe et longue. Les auteurs d'UML sont tout à fait conscients de l'importance du processus, mais l'acceptabilité industrielle de la modélisation objet passe d'abord par la disponibilité d'un langage d'analyse objet performant et standard.

2.3.3 XML

Le XML, acronyme d'eXtensible Markup Language (qui signifie langage de balisage extensible), est un langage informatique qui sert à enregistrer des données textuelles. Ce langage a été standardisé par le W3C en février 1998 et est maintenant très populaire. Ce langage, similaire à l'HTML de par son système de balisage, permet de faciliter l'échange d'information sur l'internet.

Contrairement à l'HTML qui présente un nombre finit de balises, le XML donne la possibilité de créer de nouvelles balises à volonté.

Les avantages du XML sont multiples :

- ✓ Lisibilité : il est facile pour un humain de lire un fichier XML car le code est structuré et facile à comprendre. En principe, il est même possible de dire qu'aucune connaissance spécifique n'est nécessaire pour comprendre les données comprises à l'intérieur d'un document XML.
- ✓ Disponibilité : ce langage est libre et un fichier XML peut être créé à partir d'un simple logiciel de traitement de texte (un simple bloc-notes suffit).
- ✓ Interopérabilité : quelques soit le système d'exploitation ou les autres technologies, il n'y a pas de problème particulier pour lire ce langage.
- ✓ Extensibilité : de nouvelles balises peuvent être ajoutée à souhait.
- ✓ Plusieurs parseurs XML différent doivent en principe (s'ils sont bien codés) produire le même résultat.
- ✓ Tous les navigateurs internet récents intègrent un parseur XML, pour lire les documents de ce langage informatique.

2.4 CHOIX DES API POUR L'APPLICATION

2.4.1 La persistance des données en Java

2.4.1.1 Définition

Le langage Java instancie des objets en mémoire et les manipule à travers des méthodes modifiant ainsi leur état. Cet état n'est cependant accessible que lorsque la JVM (Java Virtual

Machine) s'exécute : si celle-ci s'arrête, le contenu de la mémoire disparaît ainsi que les objets et leur état. L'un des fondements de la programmation consiste à réutiliser ces données. On appelle cela la persistance des données.

La persistance est ainsi le fait d'exister dans la durée. Un objet qui reste en l'état lorsqu'il est sauvegardé, puis rechargé plus tard, possède la propriété de persistance. Le langage Java, d'une part, et certains frameworks, d'autre part, nous permettent de rendre persistants les objets de différentes manières.

2.4.1.2 Hibernate

- **Définition**

Hibernate est une solution open source de type ORM (Object Relational Mapping) qui permet de faciliter le développement de la couche persistance d'une application. Hibernate permet donc de représenter une base de données en objets Java et vice versa.

Hibernate facilite la persistance et la recherche de données dans une base de données en réalisant lui-même la création des objets et les traitements de remplissage de ceux-ci en accédant à la base de données. La quantité de code ainsi épargnée est très importante d'autant que ce code est généralement fastidieux et redondant.

Hibernate est très populaire notamment à cause de ses bonnes performances et de son ouverture à de nombreuses bases de données.

Les bases de données supportées sont les principales du marché : DB2, Oracle, MySQL, PostgreSQL, Sybase, SQL Server, Sap DB, Interbase, ...

Hibernate est un logiciel écrit sous la responsabilité de Gavin King qui, fait partie entre autre, de l'équipe de développement de JBOSS.

L'ensemble des données nécessaires au fonctionnement de l'application sont sauvegardées dans une base de données. La manipulation des données peut se faire de différentes manières : par l'accès directement à la base en écrivant les requêtes SQL adéquates ou utiliser un outil

d'ORM (Object Relationnal Mapping) permettant de manipuler facilement les données et d'assurer leur persistance. Il en existe plusieurs.

- **Objectif**

L'objectif est de réduire le temps de développement de l'application en éliminant une grande partie du code SQL à écrire pour interagir avec la base de données et en encapsulant le code SQL résiduel. Les développeurs manipulent les classes dont les données doivent être persistantes comme des classes Java normales. Seule une initialisation correcte d'Hibernate doit être effectuée, et quelques règles respectées lors de l'écriture et de la manipulation des classes persistantes.

- **Les avantages**

Hibernate génère le code SQL nécessaire, ce qui rend l'application plus portable (s'adapte à la base de données).

La persistance est transparente. Vous pouvez faire de vos classes métiers des classes persistantes sans ajout de code.

La récupération de données est optimisée. On peut interroger la base de données de plusieurs façons (Requête SQL, langage HQL...).

Portabilité du code en cas de changement de la base de données.

- **Les inconvénients**

Il est difficile de faire des requêtes complexes avec HQL (ex: sous requêtes).

Etant une technologie récente, il reste des problèmes à résoudre (ex: les fichiers de mapping ne sont pas toujours bien formés).

Hibernate ne se base pas sur les standards.

2.4.1.3 Java Persistence API

❖ **Définition**

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

La Java Persistence API est à l'origine issue du travail du groupe d'experts JSR 220.

La persistance dans ce contexte recouvre 3 zones :

- ✓ l'API elle-même, est définie dans le paquetage javax.persistence ;
- ✓ le Langage Java Persistence Query (JPQL) ;
- ✓ l'objet/les métadonnées relationnelles.

L'API de persistance de Java, JPA, a deux aspects :

- ✓ Le premier est la possibilité d'associer des objets à une base de données relationnelle. La configuration par exception permet aux fournisseurs de persistance de faire l'essentiel du travail sans devoir ajouter beaucoup de code, mais la richesse de JPA tient également à la possibilité d'adapter ces associations à l'aide d'annotations ou de descriptions XML. Que ce soit une modification simple (changer le nom d'une colonne, par exemple) ou une adaptation plus complexe (pour traduire l'héritage), JPA offre un large spectre de possibilités. Vous pouvez donc associer quasiment n'importe quel modèle objet à une base de données existante.
- ✓ Le second aspect concerne l'interrogation de ces objets une fois qu'ils ont été associés à une base. Élément central de JPA, le gestionnaire d'entités permet de manipuler de façon standard les instances des entités. Il fournit une API pour créer, rechercher, supprimer et synchroniser les objets avec la base de données et permet d'exécuter différentes sortes de requêtes JPQL sur les entités, comme des requêtes dynamiques, statiques ou natives. Le gestionnaire d'entités autorise également la mise en place de mécanismes de verrouillage sur les données.

❖ Les spécifications de JPA

Toutes les classes et les annotations de cette API sont dans le javax.persistence paquet. Les principales composantes de l'API sont les suivantes :

- ✓ Object Relational Mapping (ORM), qui est le mécanisme d'objets cartographiques à des données stockées dans une base de données relationnelle.
- ✓ Une API gestionnaire d'entités de base de données pour effectuer des opérations liées, telles que Create, Read, Update, Delete (CRUD) des opérations. Cette API vous permet d'éviter d'utiliser l'API JDBC directement.
- ✓ Le langage Java Persistence Query (JPQL), ce qui vous permet de récupérer des données avec un langage de requête orienté objet.
- ✓ Les transactions et les mécanismes de verrouillage lors de l'accès aux données simultanément fournis par Java Transaction API (JTA). Ressources locales (non JTA) les transactions sont également pris en charge par l'API.
- ✓ Rappel et les auditeurs pour accrocher la logique métier dans le cycle de vie d'un objet persistant.

❖ Les avantages

- ✓ Recherche automatique des déployées métadonnées.
- ✓ Configuration standardisée (Unité de Persistance).
- ✓ Code normalisé de l'accès aux données, cycle de vie, et la capacité de pouvoir remplacer les annotations avec descripteur de fichier.

❖ Les inconvénients

Bien que les interfaces standards soient belles, il y a des lacunes lors de la commutation :

- ✓ Non pris en charge de toutes les stratégies de succession.
- ✓ Le fichier normalisé descripteur est essentiellement un wrapper autour du fournisseur.

Il y a aussi une absence de certains aspects bénéfiques de Hibernate. Par exemple, la requête par critère Entity Manager, propagation à travers les méthodes / objets.

2.4.1.4 Différence entre JPA et Hibernate

JPA est un cadre de gestion des données relationnelles dans les applications Java, tandis que Hibernate est une implémentation spécifique de l'JPA (si idéalement, JPA et Hibernate ne peut pas être directement comparés). En d'autres termes, Hibernate est l'un des cadres les plus populaires qui met en œuvre des JPA. Hibernate implémente JPA Hibernate Annotation par Entity Manager et les bibliothèques qui sont mises en œuvre sur le dessus des bibliothèques Hibernate Core. Les deux Entity Manager et annotations suivent le cycle de vie d'Hibernate. La nouvelle version de JPA (JPA 2.0) est entièrement prise en charge par Hibernate 3.5. JPA a l'avantage d'avoir une interface qui est normalisée de sorte que la communauté des développeurs soit plus familière avec elle, plutôt qu'avec Hibernate. D'autre part, natif Hibernate API peut être considéré comme plus puissant parce que ses caractéristiques sont un sur-ensemble de celui de l'JPA.

NB : Notre choix se porte sur JPA.

2.4.2 La couche métier

2.4.2.1 Spring

Spring est un socle pour le développement d'applications, principalement d'entreprises mais pas obligatoirement. Il fournit de nombreuses fonctionnalités parfois redondantes ou qui peuvent être configurées ou utilisées de plusieurs manières : ceci laisse le choix au développeur d'utiliser la solution qui lui convient le mieux et/ou qui répond à ses besoins.

- **Objectif**

Le but de Spring est de faciliter et de rendre productif le développement d'applications, particulièrement les applications d'entreprises.

Spring propose de nombreuses fonctionnalités de base pour le développement d'applications :

- ✓ Un conteneur léger implémentant le design pattern IoC pour la gestion des objets et de leurs dépendances en offrant des fonctionnalités avancées concernant la configuration et l'injection automatique. Un de ses points forts est d'être non intrusif dans le code de l'application tout en permettant l'assemblage d'objets faiblement couplés.
- ✓ Une gestion des transactions par déclaration offrant une abstraction du gestionnaire de transactions sous-jacent.
- ✓ Faciliter le développement des DAO de la couche de persistance en utilisant JDBC, JPA, JDO ou une solution open source comme Hibernate, iBatis, ... et une hiérarchie d'exceptions.
- ✓ Un support pour un usage interne à Spring (notamment dans les transactions) ou personnalisé de l'AOP qui peut être mis en œuvre avec Spring AOP pour les objets gérés par le conteneur et/ou avec AspectJ.
- ✓ faciliter la testabilité de l'application.

❖ **Avantages et inconvénients**

Spring est un framework open source majoritairement développé par SpringSource mais il n'est pas standardisé par le JCP.

Il est très largement utilisé dans le monde Java, ce qui en fait un standard de facto et constitue une certaine garantie sur la pérennité du framework.

Spring propose une très bonne intégration avec des frameworks open source (Struts, Hibernate, ...) ou des standards de Java (Servlets, JMS, JDO, ...)

Toutes les fonctionnalités de Spring peuvent être utilisées dans un serveur Java EE et pour la plupart dans un simple conteneur web ou une application standalone.

Les fonctionnalités offertes par Spring sont très nombreuses et les sujets couverts ne cessent d'augmenter au fur et à mesure que de nouvelles versions et de nouveaux projets sont ajoutés au portfolio.

La documentation de Spring est complète et régulièrement mise à jour lors de la diffusion de chaque nouvelle version.

La mise en œuvre de Spring n'est pas toujours aisée car il existe généralement plusieurs solutions pour mettre en œuvre une fonctionnalité : par exemple, généralement avec Spring 3.0, une fonctionnalité est utilisable par configuration XML, par annotations ou par API. Bien sûr cela permet de choisir mais cela impose un arbitrage selon ses besoins.

Il n'est pas rare que les livrables aient une taille importante du fait des nombreuses librairies requises par Spring et ses dépendances.

2.4.2.2 Entreprise Java Bean 3

Les Entreprises Java Bean ou EJB sont des composants serveurs donc non visuels qui respectent les spécifications d'un modèle éditées par Sun. Ces spécifications définissent une architecture, un environnement d'exécution et un ensemble d'API.

Le respect de ces spécifications permet d'utiliser les EJB de façon indépendante du serveur d'applications J2EE dans lequel ils s'exécutent, dès l'instant que le code de mise en œuvre n'utilise pas d'extensions proposées par un serveur d'applications particulier.

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 tel que la gestion des transactions, la persistance des données, la sécurité, ...

La plate-forme Java EE propose de mettre en œuvre les couches métiers et la persistance avec les EJB. Particulièrement intéressant dans des environnements fortement distribués, jusqu'à la version 3, leur mise en œuvre est assez lourde sans l'utilisation d'outils tels que certains IDE ou XDoclet.

Il existe deux types d'EJB : les beans de session (session beans) et les beans entité (les entity beans). Depuis la version 2.0 des EJB, il existe un troisième type de bean : les beans orientés message (message driven beans). Ces trois types de bean possèdent des points communs notamment celui de pouvoir être déployés dans un conteneur d'EJB.

Les sessions beans peuvent être de deux types : sans état (stateless) ou avec état (stateful).

Les beans de session sans état peuvent être utilisés pour traiter les requêtes de plusieurs clients. Les beans de session avec état ne sont accessibles que lors d'un ou de plusieurs échanges avec le même client. Ce type de bean peut conserver des données entre les échanges avec le client.

Les beans entité assurent la persistance des données. Il existe deux types d'entity bean :

- ✓ persistance gérée par le conteneur CMP (Container Managed Persistence) ;
- ✓ persistance gérée par le bean BMP (Bean Managed Persistence).

Avec un bean entité CMP (Container Managed Persistence), c'est le conteneur d'EJB qui assure la persistance des données. Un bean entité BMP (Bean Managed Persistence), assure lui-même la persistance des données grâce au code inclus dans le bean.

La version 3 des EJB vise donc à simplifier le développement et la mise en œuvre des EJB qui sont fréquemment jugés trop complexes et trop lourds à mettre en œuvre.

Cette nouvelle version majeure des EJB propose une simplification de leur développement tout en conservant une compatibilité avec sa précédente version. Elle apporte de très nombreuses fonctionnalités dans le but de simplifier la mise en œuvre des EJB.

Cette simplification est rendue possible notamment par :

- ✓ l'utilisation des annotations ;
- ✓ la mise en œuvre de valeurs par défaut qui répondent à la plupart des besoins (configuration par exception) ;
- ✓ le descripteur de déploiement qui est facultatif ;
- ✓ l'utilisation de POJO et de JPA pour les beans de type entity ;
- ✓ l'injection de dépendances côté serveur mais aussi côté client (l'interface Home qui gérait le cycle de vie est abandonnée) qui remplace l'utilisation directe de JNDI.

Tous ces éléments délèguent une partie du travail du développeur au conteneur d'EJB.

2.4.2.3 EJB3 vs Spring Framework

Plus important encore, le Spring est une mise en œuvre tout en EJB 3.0 est une spécification. Mais ils ont quelques zones de chevauchement, par exemple ils peuvent fournir un mécanisme pour des services de middleware avec les applications Java. Spring a été développé comme une réaction contre EJB, ce qui rend la comparaison entre les deux naturelle. Surtout maintenant que la nouvelle version d'EJB est disponible, il est un bon moment pour réévaluer la façon dont EJB 3.0 a abordé les lacunes des versions précédentes.

- **Caractéristiques des EJB 2.x, Spring et EJB 3.0**

EJB 2.x :

- ✓ Permet au développeur d'écrire un tas d'objets redondants.
- ✓ Les descripteurs de déploiement ont des difficultés.
- ✓ Les beans entité sont à moitié cuit et non-portable.
- ✓ Les beans entité sont lents. Remoting et la synchronisation mondiale sont overkill.

Spring Framework :

- ✓ Programmation POJO.
- ✓ Beaucoup plus facile à développer, tester et déployer.
- ✓ Donne 90 +% de ce qu'EJB fait.

EJB 3.0 :

- ✓ EJB 3 embrasse la programmation POJO à travers les annotations.
- ✓ Le verbose descripteur de déploiement XML a été rendue facultative.
- ✓ Le concept Entity Bean est plus géré par le conteneur.
- ✓ EJB 3 adopte JPA, un paradigme basé sur API similaire à Hibernate, TopLink et JDO.
- ✓ Object Relational Mapping et les requêtes d'objets ont été totalement définis au lieu d'être laissée à des fournisseurs de conteneurs pour le tri.

- ✓ EJB 3 fait un usage intensif de « défaillant intelligent » chaque fois que c'est possible. Ceci est une idée similaire à « convention plutôt que configuration » dans le monde Rails.

NB : Notre choix se porte sur EJB3.

2.5 PRESENTATION DE L'APPLICATION

2.5.1 Cas d'utilisation

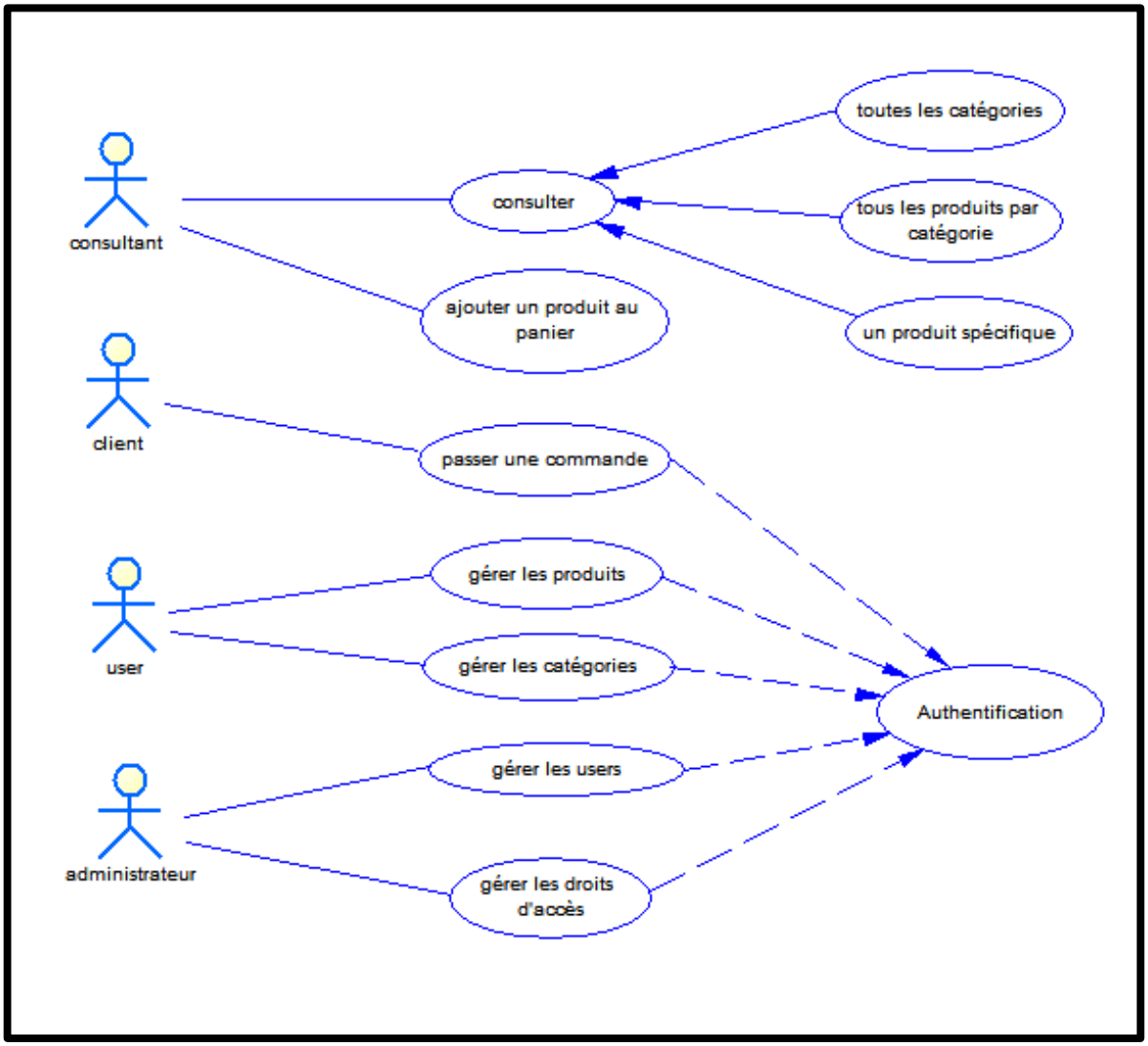


Figure 2-2 : Cas d'utilisation

2.5.2 Diagramme des classes

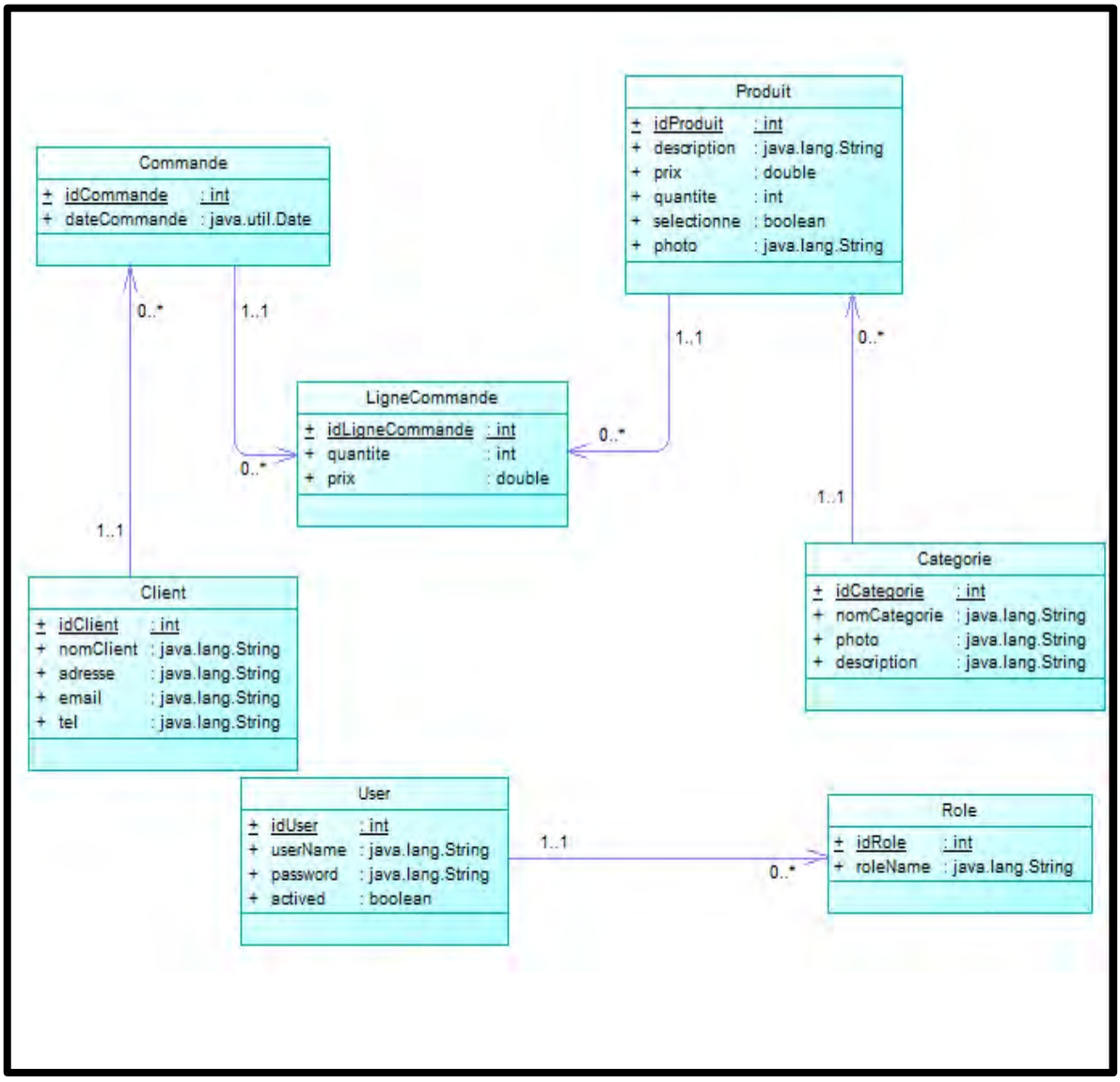


Figure 2-3 : Diagramme des classes

2.5.3 Application 3-tiers et modèle MVC

Une application web possède souvent une architecture 3-tiers, autrement dit elle est souvent divisée en trois niveaux ou couches :

- ✓ couche présentation ;
- ✓ couche métier ;
- ✓ couche accès aux données ou dao.

La couche présentation ou interface utilisateur est l'interface (graphique souvent) qui permet à l'utilisateur de piloter l'application et d'en recevoir des informations.

La couche métier implémente les algorithmes « métier » de l'application. Cette couche est indépendante de toute forme d'interface avec l'utilisateur. Ainsi, elle doit être utilisable aussi bien avec une interface console, une interface web, une interface de client riche. Elle doit ainsi pouvoir être testée en-dehors de l'interface web et notamment avec une interface console. C'est généralement la couche la plus stable de l'architecture. Elle ne change pas si on change l'interface utilisateur ou la façon d'accéder aux données nécessaires au fonctionnement de l'application.

La couche dao s'occupe de l'accès aux données, le plus souvent des données persistantes au sein d'un SGBD.

Les couches métier et dao sont normalement utilisées via des interfaces Java. Ainsi la couche métier ne connaît de la couche dao que son ou ses interfaces et ne connaît pas les classes les implémentant. C'est ce qui assure l'indépendance des couches entre-elles : changer l'implémentation de la couche dao n'a aucune incidence sur la couche métier tant qu'on ne touche pas à la définition de l'interface de la couche dao. Il en est de même entre les couches interface utilisateur et métier.

L'architecture MVC prend place dans la couche interface utilisateur lorsque celle-ci est une interface web.

Cette figure, illustre l'architecture 3-tiers et la mise en place du MVC.

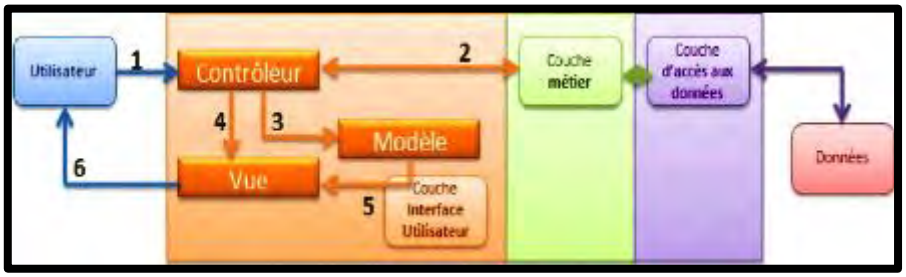


Figure 2-4 : Architecture 3-tiers et mise en place du MVC

Explication de la figure : le traitement d'une demande d'un client se déroule selon les étapes suivantes :

1) Le client fait une demande au contrôleur. Celui-ci voit passer toutes les demandes des clients. C'est la porte d'entrée de l'application. C'est le « C » de MVC.

2) Le contrôleur « C » traite cette demande. Pour ce faire, il peut avoir besoin de l'aide de la couche métier. Une fois la demande du client traitée, celle-ci peut faire appel à diverses réponses. Un exemple classique est :

- ✓ une page d'erreurs si la demande n'a pu être traitée correctement ;
- ✓ une page de confirmation.

3) Le contrôleur choisit la réponse (une vue) à envoyer au client. Choisir la réponse à envoyer au client nécessite plusieurs étapes:

- o choisir l'objet qui va générer la réponse. C'est ce qu'on appelle la vue « V », le « V » de MVC. Ce choix dépend en général du résultat de l'exécution de l'action demandée par l'utilisateur ;
- o lui fournir les données dont il a besoin pour générer cette réponse. En effet, celle-ci contient le plus souvent des informations calculées par le contrôleur. Ces informations forment ce qu'on appelle le modèle « M » de la vue, le « M » de MVC. L'étape 3 est lié donc au choix d'une vue « V » et à la construction du modèle « M » nécessaire à celle-ci.

4) Le contrôleur « C » demande la vue choisie de s'afficher. Il s'agit le plus souvent de faire exécuter une méthode particulière de vue « V » chargée de générer la réponse au client.

5) Le générateur de vue « V » utilise le modèle « M » préparé par le contrôleur « C » pour initialiser les parties dynamiques de la réponse qu'il doit envoyer au client.

6) La réponse est envoyée au client. La forme exacte de celle-ci dépend du générateur de vue. Ce peut être un flux HTML, PDF, Excel, etc. Dans notre application, et pour plus de simplicité, la couche métier est intégrée au générateur de vue.

2.5.4 Partie Administrateur

C'est la partie qui permet de gérer l'application. Elle nécessite une authentification. Voici quelques captures illustrant quelques pages de cette partie.

Figure 2-5: Page d'accueil côté Administrateur

C'est la page d'accueil. A partir de cette page, l'administrateur a le droit d'ajouter des catégories, des produits, faire des mises à jour, suppression etc.

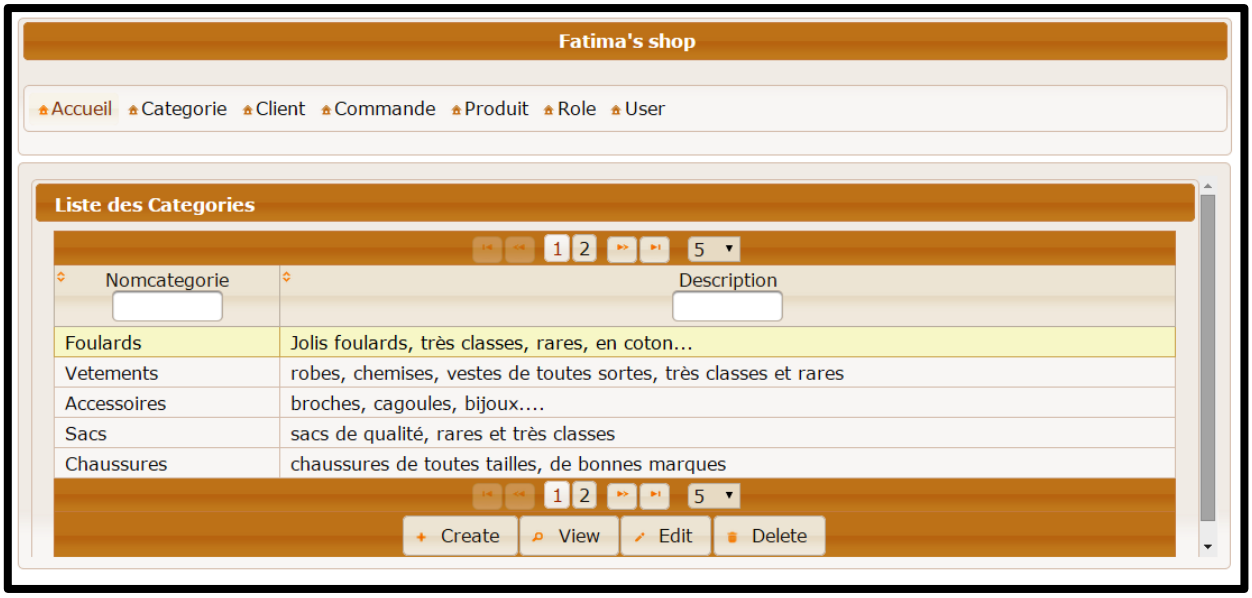


Figure 2-6 : Rubrique Catégorie

C'est la rubrique catégorie. De là, on peut créer, mettre à jour ou supprimer une catégorie.

Figure 2-7 : Ajout Catégorie

Cette fenêtre permet de créer une nouvelle catégorie.

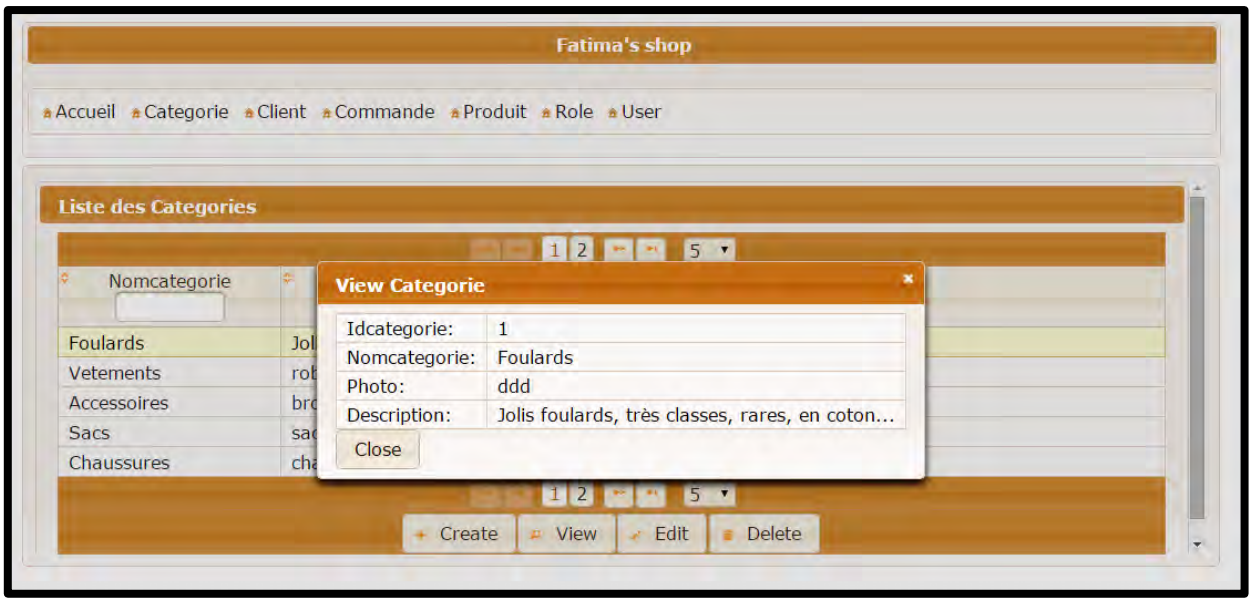


Figure 2-8 : Caractéristiques d'une catégorie

Cette fenêtre affiche les caractéristiques d'une catégorie.

Figure 2-9 : Modification d'une catégorie

Cette fenêtre permet de modifier une catégorie.

2.5.5 Partie Utilisateur

C'est la partie qui représente la boutique. Elle permet aux utilisateurs de consulter un produit, de faire une commande, etc. L'utilisateur doit avoir obligatoirement un compte pour pouvoir valider ses commandes.

Ci-dessous des captures pour visualiser quelques pages de l'interface utilisateur.

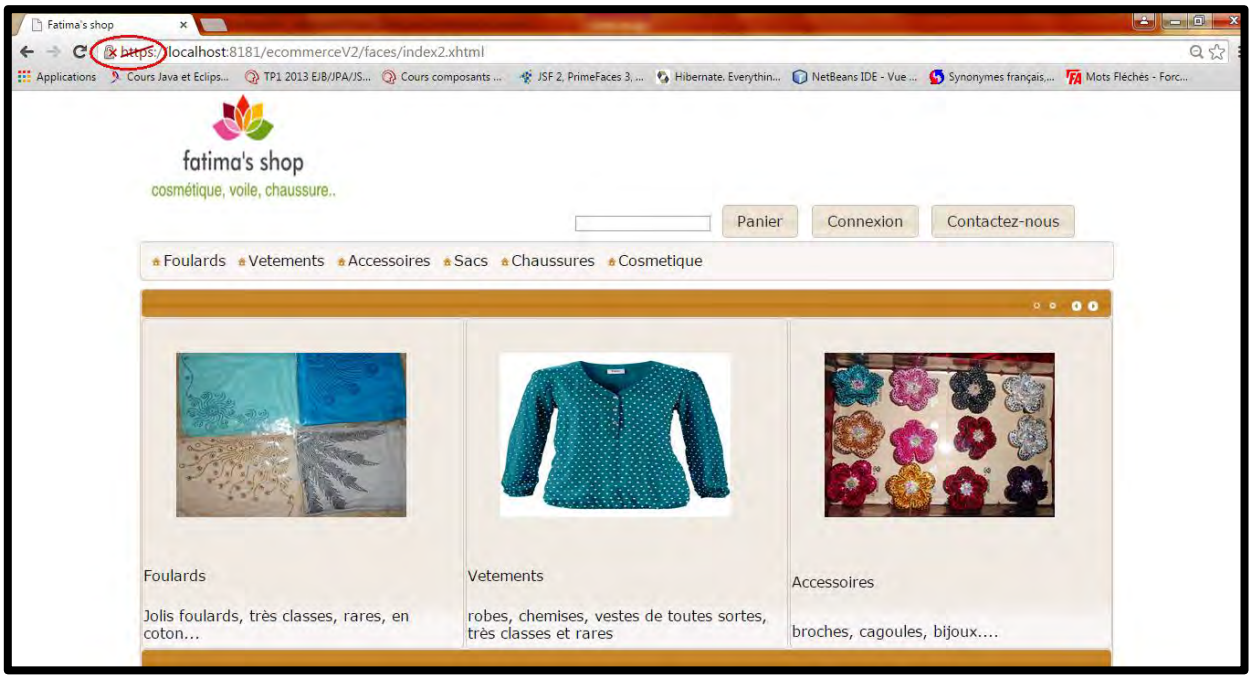


Figure 2-10 : Page d'accueil de l'application côté Utilisateur

C'est la page d'accueil de notre boutique en ligne. Elle est composée d'un entête, constitué du logo de la boutique, des boutons « panier », « connexion » et « contactez-nous », d'un espace pour rechercher un produit, d'un menu de la liste des catégories existant dans notre boutique, et le corps de la page.

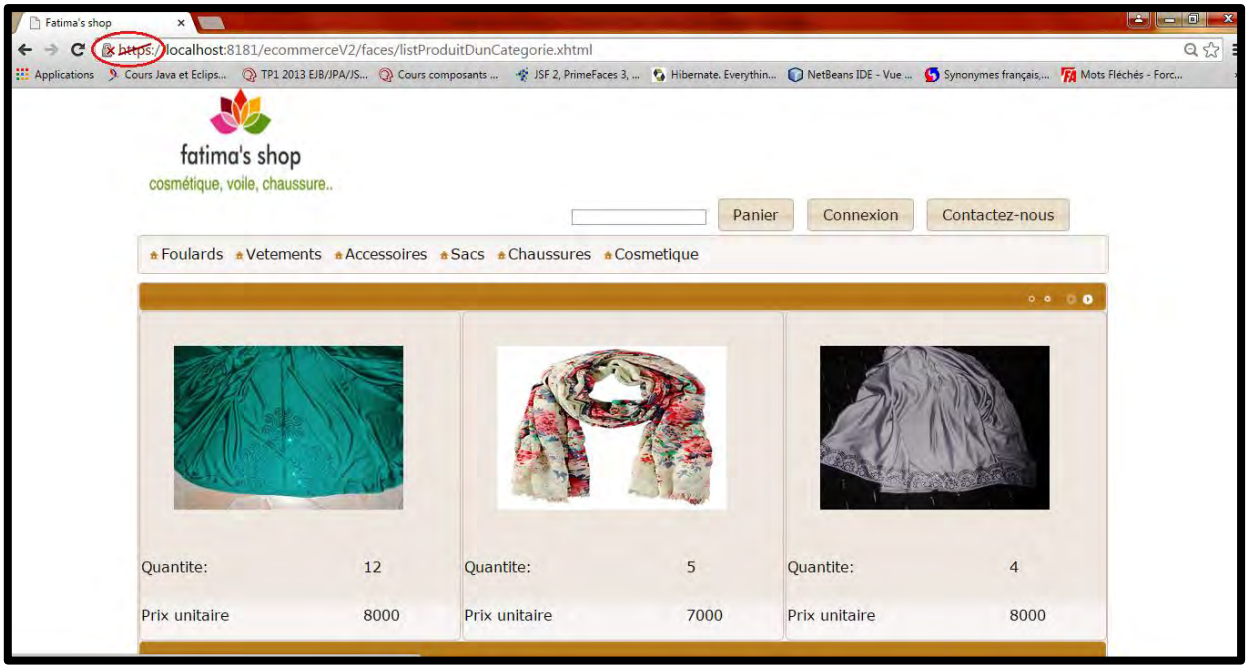


Figure 2-11 : Liste des produits d'une catégorie

Affiche une de ces pages ou d'autres selon la catégorie sur laquelle on a cliqué et affiche l'ensemble des produits existant dans la catégorie.

Figure 2-12 : Un produit choisi pour l'ajouter au panier

Affiche cette page quand on clique sur un produit spécifique d'une catégorie. Le bouton « ajouter au panier » permet d'ajouter le produit sélectionné avec la quantité voulue dans le panier.

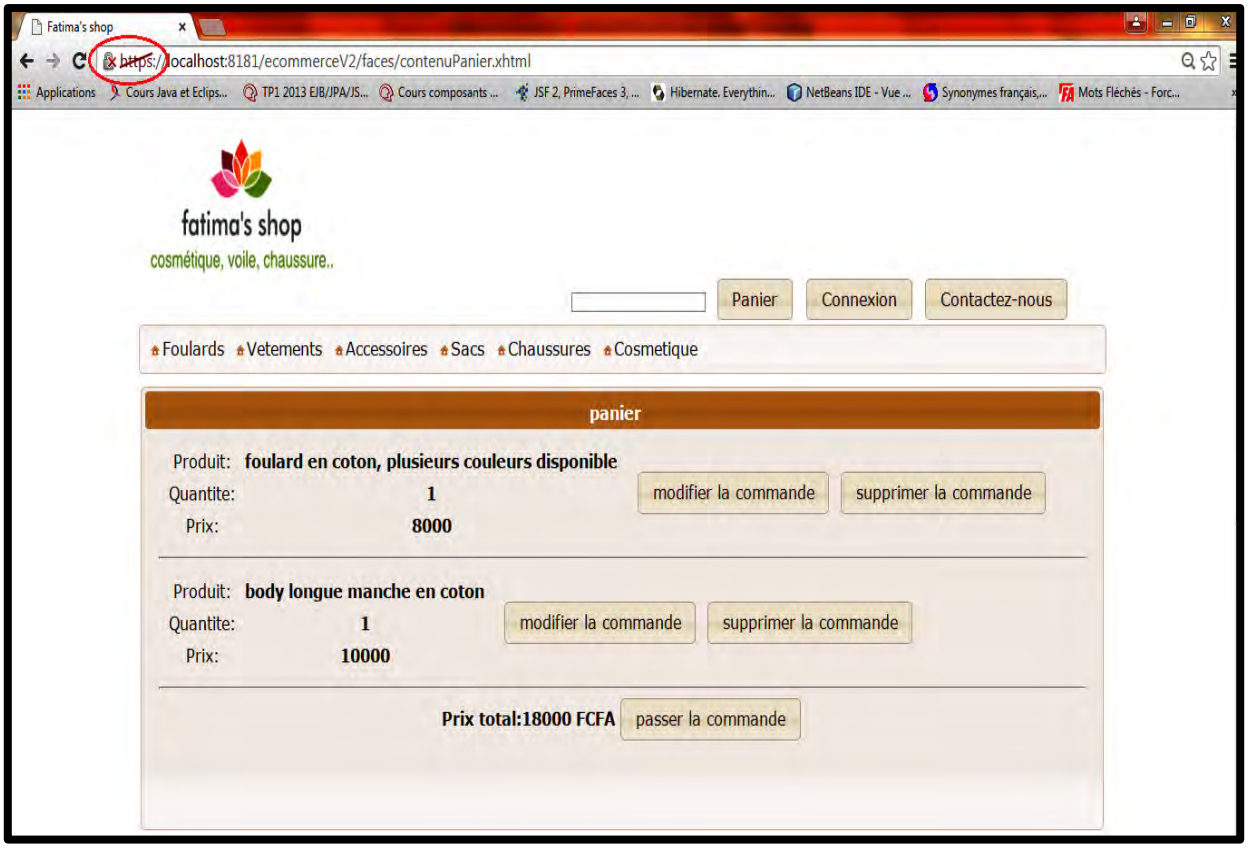


Figure 2-13 : Contenu du panier

Cette page nous affiche tous les produits qui se trouvent dans le panier du client. Le bouton « passer la commande » nous permet de valider notre commande. Pour cela, il nous redirige vers la page de connexion.

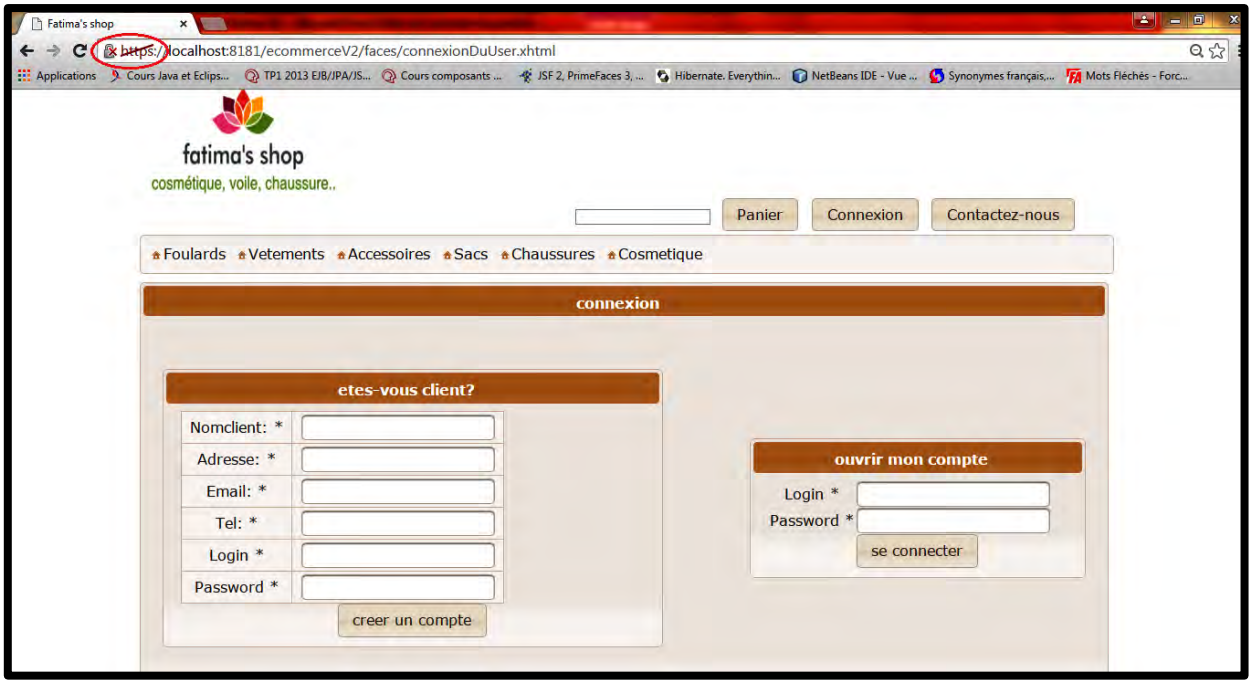


Figure 2-14 : Page de connexion et de création de compte

Cette page permet au client de se connecter pour pouvoir valider ses commandes ou de créer un compte s'il n'en a pas.

❖ **Remarque :**

Nous remarquons que toutes les pages de notre application commencent par « https ». Cela signifie que la transaction des données entre le client et notre serveur est sécurisée. Autrement dit, lorsqu'elle est légitime, une connexion HTTPS est dite « sécurisée » de par le fait qu'elle chiffre la connexion établie entre l'application client et l'application serveur, les données sont alors incompréhensibles pour une personne mal intentionnée qui souhaiterait observer la connexion et effectuer une analyse des trames en vue de récupérer des informations sensibles comme des données d'authentification, par exemple.

Les trames HTTPS n'offrent cependant pas plus de visibilité aux applications de sécurité souhaitant vérifier, par analyse, l'intégrité des données qu'elles transportent ; celles-ci pouvant s'avérer être des codes malicieux (virus, backdoor, vers) potentiellement dangereux pour la sécurité du poste client et donc pour l'infrastructure même de l'entreprise.

CHAPITRE 3 : ASPECT SECURITAIRE

3.1 LA CRYPTOGRAPHIE MODERNE

3.1.1 Généralités

La cryptographie est une discipline ancienne. Déjà dans l'antiquité, les Grecs avaient inventé des méthodes pour chiffrer les messages. L'une d'entre elles, datant du VIème siècle avant J.C., consistait à enrouler une bande de papier autour d'un cylindre, puis à écrire le message sur la bande. Une fois déroulé, le papier était envoyé au destinataire qui, dès qu'il possédait le diamètre du cylindre, pouvait déchirer le message.

Pendant de nombreuses années, la cryptographie était exclusivement réservée au domaine militaire et diplomatique. La littérature sur le sujet était donc très peu abondante. La première publication fondamentale dans ce domaine a été l'article de Claude Shannon de 1949 – « *The communication theory of secrecysystems* » [Sha]- dans lequel il jette les bases mathématiques d'un système de communication chiffrée, à partir de la définition d'un nouveau modèle : la théorie de l'information. Une contribution importante a ensuite été celle de Feistel, avec la publication, au début des années 1970, de ses travaux sur les schémas de chiffrement itératifs par blocs [Fei1, Fei2], qui ont conduit en 1977 à la proposition de l'algorithme DES comme standard de chiffrement à clef secrète pour des applications non classifiées. L'accroissement de la puissance des ordinateurs ayant remis en cause la sécurité du DES, il a été remplacé en octobre 2000 par un nouveau standard appelé AES. Cet algorithme est l'aboutissement de recherches récentes notamment dans le domaine de la cryptanalyse.

Mais l'avancée majeure en cryptographie a incontestablement été la publication, en 1976, de l'article « *New directions in cryptography* » [Dif], de WhitfieldDiffie et Martin Hellman. Cet article introduit le concept révolutionnaire de cryptographie à clef publique. Même si les auteurs ne donnent pas les réalisations pratiques d'un système à clef publique, les propriétés d'un tel système sont clairement énoncées. En outre, ils présentent un protocole par lequel deux entités peuvent convenir d'une clef secrète à partir de la connaissance préalable des seules données publiques. La première réalisation d'un système à clef publique est due à Ronald

Rivest, Adi Shamir et Leonard Adleman, en 1978 : c'est le RSA [Riv]. Depuis lors, la littérature sur ce sujet n'a cessé de se développer.

Plus récemment, pour faire face aux nouvelles menaces induites par le développement des réseaux et la numérisation massive des documents, la cryptographie a dû offrir de nouvelles fonctionnalités : la garantie de l'authenticité des messages (provenance et contenu) réalisée par des algorithmes de signature numérique et la certification de l'identité d'une personne techniques d'identification en sont les principaux exemples.

Nous allons vous présenter deux grandes catégories de procédés cryptographiques les plus utilisées : les algorithmes de chiffrement, qui servent à protéger la confidentialité des données, et les algorithmes de signature qui, comme les signatures manuscrites, garantissent la provenance et l'intégrité des messages. Ainsi, nous vous parlerons des fonctions de hachages.

3.1.2 Le chiffrement

Un algorithme de chiffrement transforme un message, appelé texte clair, en un texte chiffré qui ne sera lisible que par son destinataire légitime. Cette transformation est effectuée par une fonction de chiffrement paramétrée par une clef de chiffrement. Un interlocuteur privilégié peut alors déchirer le message en utilisant la fonction de déchiffrement s'il connait la clef de déchiffrement correspondant. Un tel système n'est sûr que s'il est impossible à un intrus de déduire le texte clair du message chiffré, et a fortiori de retrouver la clef de déchiffrement.

Cette formalisation a maintenant un peu plus d'un siècle. A cette époque, les cryptographes ont pris conscience qu'il n'était pas réaliste de faire reposer la sécurité d'un système de chiffrement sur l'hypothèse qu'un attaquant n'a pas connaissance de la méthode utilisée. La publication récente sur Internet des spécifications d'algorithmes propriétaires, comme celui utilisé dans le système GSM, nous a encore montré qu'il est impossible de conserver un algorithme secret dans le long terme. En conséquence, la sécurité d'un algorithme de chiffrement doit uniquement reposer sur le secret de la clef de déchiffrement. Par ailleurs, le fait de rendre publiques les méthodes de chiffrement et de déchiffrement offre une certaine garantie sur la sécurité d'un système, dans la mesure où tout nouvel algorithme cryptographique est immédiatement confronté à la sagacité de la communauté scientifique.

On distingue deux grands types d'algorithmes de chiffrement, les algorithmes à clef secrète et les algorithmes à clef publique. Chacune de ces deux classes possède ses propres avantages et inconvénients. Les systèmes à clef secrète nécessitent le partage d'un secret entre les interlocuteurs.

La découverte en 1976 des systèmes à clef publique a permis de s'affranchir de cette contrainte, mais elle n'a pas pour autant apporter des solutions parfaites, dans la mesure où tous les algorithmes de chiffrement à clef publique, de par leur lenteur, ne permettent pas le chiffrement en ligne. Dans la plupart des applications actuelles, la meilleure solution consiste à utiliser un système hybride, qui combine les deux types d'algorithmes.

- **Le chiffrement à clef secrète**

Les algorithmes de chiffrement à clef secrète (ou symétriques ou encore conventionnels) sont ceux pour lesquels l'émetteur et le destinataire partagent une même clef secrète. Autrement dit, les clefs de chiffrement et de déchiffrement sont identiques. L'emploi d'un algorithme à clef secrète lors d'une communication nécessite donc l'échange préalable d'un secret entre les deux protagonistes à travers un canal sécurisé ou par le biais d'autres techniques cryptographiques.

- **Le chiffrement à clef publique**

La cryptographie à clef publique (ou asymétrique) évite le partage d'un secret entre les deux interlocuteurs. Dans un système de chiffrement à clef publique, chaque utilisateur dispose d'un couple de clefs, une clef publique qu'il met en général à disposition de tous dans un annuaire, et une clef secrète connue de personne. Par exemple, pour envoyer un message confidentiel à Bob, Alice chiffre donc le message clair à l'aide de la clef publique de Bob. Ce dernier, à l'aide de la clef secrète correspondante, est le seul en mesure de déchiffrer le message reçu.

La notion essentielle sur laquelle repose le chiffrement à clef publique est celle de fonction à sens unique avec trappe. Une fonction est appelée à sens unique si elle est facile à calculer mais impossible à inverser. Impossible signifie ici infaisable en un temps réaliste avec une puissance de calcul raisonnable. On considère comme étant impossible, par exemple, un calcul qui, reparti sur un Milliard de processeurs en parallèle, nécessiterait un Milliard d'années. Une telle

fonction est dite à trappe si le calcul de l'inverse devient facile dès que l'on possède une information supplémentaire (la trappe). Il est très simple de construire un système de chiffrement à clef publique à partir d'une fonction à sens unique avec trappe. La procédure de chiffrement consiste simplement à appliquer la fonction au message clair. La fonction étant à sens unique, il est très difficile de l'inverser, c'est-à-dire de déterminer le message clair à partir du message chiffré, sauf si on connait la trappe, qui correspond à la clef secrète du destinataire. Toute la difficulté réside donc dans la recherche de ces fonctions très particulières. Leur construction s'appuie généralement sur des problèmes mathématiques réputés difficiles ; le plus célèbre est celui de la factorisation de grands nombres entiers, qui est à la base du système RSA.

3.1.3 La signature numérique

Dans de nombreuses communications, la confidentialité des données importe peu mais il est nécessaire de s'assurer de leur provenance et de leur intégrité, c'est-à-dire de vérifier qu'elles n'ont pas été modifiées lors de la transmission.

Un procédé de signature numérique consiste à adjoindre au texte clair un petit nombre de bits qui dépendent simultanément du message et de son auteur. Pour obtenir les mêmes fonctionnalités que la signature que l'on appose au bas d'un texte à support papier, il faut que chacun puisse vérifier une signature mais que personne ne puisse l'imiter.

Un schéma de signature est donc composé d'une fonction de signature et d'une fonction de vérification. La fonction de signature est paramétrée par une clef secrète propre au signataire ; elle associe à tout message clair une signature. La fonction de vérification, elle, ne nécessite la connaissance d'aucun secret. Elle permet à partir du message clair et de la signature de vérifier l'authenticité de cette dernière.

Un schéma de signature doit donc posséder un certain nombre de propriétés. En particulier, il doit être en pratique impossible de contrefaire une signature : seul le détenteur de la clef secrète peut signer en son nom. La signature ne doit plus être valide si le message clair a été modifié ; ce faisant il est impossible de réutiliser une signature. Enfin, le signataire ne doit pas pouvoir nier avoir signé un message.

Un schéma de signature garantit donc :

- ✓ L'identité de la personne émettant le message ;
- ✓ L'intégrité des données reçues, c'est-à-dire l'assurance que le message n'a pas été modifié lors de sa transmission ;
- ✓ la non-répudiation du message, ce qui signifie que l'émetteur du message ne pourra pas nier en être l'auteur.

C'est pourquoi les procédés de signature numérique constituent une preuve au même titre que la signature manuscrite. Leur valeur juridique est désormais reconnue par la loi (loi 2000- 230 du 13 mars 2000).

3.1.4 La fonction de hachage

Pour signer des messages, on a recours, avant d'appliquer des algorithmes de chiffrement, à des fonctions de hachage cryptographiques.

Une telle fonction, dont la description est entièrement publique, transforme une chaîne binaire de longueur quelconque en une chaîne binaire de longueur fixée (généralement 128 ou160 bits), appelée condensé ou haché. Deux raisons motivent l'utilisation de ces fonctions : la première, est la lenteur des systèmes à clef publique : les messages à signer étant relativement longs, il serait trop coûteux de les signer in-extenso ; on leur applique donc d'abord une fonction de hachage, et on signe le haché du message, et non le message lui-même.

La deuxième raison, liée à la sécurité, est d'empêcher certains types d'attaques sur les schémas de signature. Par exemple dans le cas de RSA, la signature du produit de deux messages est le produit des signatures de chacun d'eux. Cette particularité de RSA le rend vulnérable à la forge de messages (certes le plus souvent inintelligibles), mais pouvant néanmoins représenter une menace, surtout lorsque le mécanisme de signature est utilisé à des fins d'identification. Le hachage du message avant signature permet de pallier cette faiblesse.

Pour pouvoir être utilisée dans des applications cryptographiques, une fonction de hachage doit cependant satisfaire la contrainte suivante : il doit être impossible en pratique de

trouver une collision, c'est-à-dire deux messages qui aient le même haché. Pour que la recherche de collisions nécessite au moins 2^{64} essais, le haché doit avoir une longueur d'au moins 128 bits.

La plupart des fonctions de hachage utilisées actuellement sont des améliorations de la fonction MD4. Cette dernière était fréquemment utilisée avant sa cryptanalyse en 1996. Parmi les principales fonctions de hachage, on peut citer la fonction MD5 (Message Digest 5). Cette fonction produit un condensé de 128 bits. Certaines faiblesses dans sa construction ont été décelées récemment mais elles ne mettent pas directement en cause sa sécurité. Toutefois, certains préfèrent éviter son utilisation. On peut donc lui préférer le standard américain SHA-1 (Secure Hash Algorithm), utilisé dans le schéma de signature DSA ou la fonction RIPEMD-160, qui a été conçue dans le cadre d'un projet européen. Ces deux fonctions ont également l'avantage de produire des condensés de 160 bits.

3.2 NOTION DE SECURITE WEB

Selon les experts en la matière, la majorité des applications web souffrent de leur vulnérabilité. Il est pourtant nécessaire de comprendre que la sécurité des sites Internet ne souffre pas de manichéisme : un site peut être plus ou moins sûr, et la sécurité est souvent une affaire de compromis. Elle dépend de nombreux facteurs, tels que les connaissances du développeur mais aussi la conception et la complexité du site, le coût et les délais de fabrication et même parfois aussi étonnant que cela puisse paraître de la responsabilisation de ses utilisateurs.

3.3 QUELQUES TYPES D'ATTAQUES

Voyons quelques attaques recensées pour bien comprendre la complexité du problème et l'ampleur des failles de sécurité potentielles d'un site Internet.

3.3.1 Injection SQL (« SQL injection »)

L'Injection SQL est l'une des vulnérabilités les plus courantes des applications web.

SQL est le langage utilisé pour exécuter des actions sur les bases de données. L'injection SQL consiste à tenter de modifier la requête prévue par le développeur pour exécuter une action différente (illicite) sur la base de données. Par exemple, dans un formulaire de connexion contenant un nom d'utilisateur et un mot de passe, un pirate peut tenter d'écrire un code SQL dans l'un de des champs pour court-circuiter la nécessité de présenter le mot de passe. Si l'application ne prend pas la peine de vérifier le contenu des deux champs et de bloquer le texte suspect, elle est vulnérable à ce type d'attaque.

Autrement dit, la méthode d'attaque par injection SQL exploite l'application web par injection de requêtes malveillantes, entraînant la manipulation de données. Presque toutes les bases de données SQL et les langages de programmation sont potentiellement vulnérables et plus de 60% des sites web s'avèrent vulnérables à l'injection SQL.

- **Solutions :**

Ces menaces peuvent être atténuées, voire évitées, avec les outils et les connaissances appropriées. En effet, deux solutions peuvent être apportées. La première consiste à échapper les caractères spéciaux contenus dans les chaînes de caractères entrées par l'utilisateur en utilisant des fonctions. La seconde solution consiste à utiliser des requêtes préparées : dans ce cas, une compilation de la requête est réalisée avant d'y insérer les paramètres et de l'exécuter, ce qui empêche un éventuel code inséré dans les paramètres d'être interprété.

3.3.2 Cross-Site Scripting (XSS)

Le Cross-Site Scripting est une des attaques les plus connues.

En utilisant une vulnérabilité XSS, contrairement à une injection SQL, le pirate s'attaque à l'utilisateur plutôt qu'à l'application web.

Les attaques de type XSS consistent en l'insertion de bouts de code non prévu au sein d'une page web qui sera exécuté du côté du client par les navigateurs. La faille existe dès que l'application web affiche sur une page des données étrangères sans filtrer les balises HTML et en particulier la balise <script>.

On distingue deux types d'attaques XSS : le XSS permanent et le XSS transitoire.

Le XSS transitoire profite en général de failles sur les paramètres d'url afin d'y injecter un code fallacieux ; l'url sera alors envoyée à la victime par mail, site malveillant, MSN, etc. Ce type d'attaque fait souvent corps avec l'ingénierie sociale.

Le XSS permanent apparait lorsque l'application web enregistre dans une base de données ou un fichier le code fallacieux pour ensuite l'afficher sur une page publique (forum, blog, etc.). Ainsi chaque client qui affiche la page vérolée devient une victime.

- **Solutions :**

La meilleure façon de vous protéger en tant qu'utilisateur est de ne suivre que les liens du site web principal que vous souhaitez afficher. Si vous visitez un site Web et la relie à « CNN » par exemple, au lieu de cliquer sur elle visiter le site principal de « CNN » et d'utiliser son moteur de recherche pour trouver le contenu. Cela pourrait probablement éliminer les quatre-vingt-dix pour-cent du problème. Parfois XSS peut être exécuté automatiquement lorsque vous ouvrez un e-mail, pièce jointe, lire un livre, ou un babillard poste. Une des meilleures façons de vous protéger est d'éteindre Javascript dans les paramètres de votre navigateur. Dans Internet Explorer, il faut transformer les paramètres de sécurité élevée. Cela peut empêcher le vol de cookies, ceci est en général une chose sûre à faire.

3.3.3 Attaque d'URL sémantique (« Semantic URL attack »)

Cette technique consiste à modifier les variables passées en clair par l'URL. Ainsi, l'adresse http://www.exemple.com?utilisateur=aicha&mail=aicha@exemple.com transmet deux variables :

- ✓ « utilisateur », dont la valeur est « aicha »
- ✓ « mail », dont la valeur est « aicha@exemple.com ».

Si une application ne vérifie pas correctement que ces variables correspondent bien à l'utilisateur authentifié, un attaquant peut par exemple injecter le code ou se faire passer pour

un autre utilisateur et découvrir des informations le concernant, modifier son mot de passe, etc., selon les défauts de l'application web.

Bien souvent, il leur faut procéder par tentatives multiples pour trouver les failles d'un site, mais les pirates savent être patients et méthodiques, et ils créent des robots qui vont tenter de découvrir toute faille permettant des opérations illicites.

- **Solutions :**

Un moyen d'éviter les attaques d'URL sémantiques est d'utiliser la session des variables.

3.3.4 Attaque CSRF (« Cross-Site Request Forgeries »)

Là encore, le pirate s'attaque à l'utilisateur plutôt qu'à l'application web. En effet, les attaques CSRF consistent à faire exécuter des requêtes (GET, POST ou PUT sur un serveur IIS) involontairement aux utilisateurs accrédités d'un site vulnérable. L'envoi des requêtes se fait lorsque la victime visite un site malveillant, un site victime d'une attaque XSS ou en cliquant sur un lien corrompu.

Le vecteur le plus souvent utilisé pour ce type d'attaque est la balise html <img/>.

- **Solutions :**

La seule méthode vraiment efficace consiste à utiliser des « tokens » aléatoires (secret) sur toutes les pages sensibles. Ce jeton doit être envoyé au serveur (en champs caché) lors de la soumission d'un formulaire ou d'actions critiques. Si l'URL envoyée ne contient pas ce nombre aléatoire qui ne peut être prédit par l'attaquant, le serveur web ne traitera pas cette dernière.

Une autre possibilité réside dans le fait que le serveur web implémente une table de nombre aléatoires qui sert à définir le nom d'une variable en fonction d'une session donnée. Dans ce cas, l'information ne peut pas être prédite par le pirate.

Une dernière possibilité consiste à obliger l'utilisateur à valider chaque action critique par la soumission de son mot de passe. Une fois que la requête d'ordre de transfert d'argent a été

effectuée, l'utilisateur doit confirmer avec un paramètre (dont le pirate ne dispose pas). L'attaque est alors anéantie.

Du côté client, la problématique réside dans le fait d'empêcher le navigateur d'effectuer des requêtes sans l'autorisation préalable du client. Voici quelques conseils précieux pour aider le client à parer ce genre d'attaque:

- ✓ ne pas utiliser un « client mail » qui interprète les codes HTML ;
- ✓ ne pas sauvegarder les identifiants dans le navigateur ;
- ✓ ne pas utiliser la fonction « remember me » proposée par de nombreux sites ;
- ✓ ne pas suivre les liens suspects ;
- ✓ se déconnecter lorsque vous avez fini de visiter les sites sensibles.

3.3.5 Hameçonnage

L'hameçonnage, « phishing » ou filoutage est une technique utilisée par des fraudeurs pour obtenir des renseignements personnels dans le but de perpétrer une usurpation d'identité. La technique consiste à faire croire à la victime qu'elle s'adresse à une entité digne de confiance (banque, administration, etc.) afin de lui soutirer des renseignements personnels : mot de passe, numéro de carte de crédit, date de naissance, etc. C'est une forme d'attaque informatique reposant sur l'ingénierie sociale. Elle peut se faire par courrier électronique, par des sites web falsifiés ou autres moyens électroniques.

Les criminels informatiques utilisent généralement l'hameçonnage pour voler de l'argent. Les cibles les plus courantes sont les services bancaires en ligne, les fournisseurs d'accès Internet et les sites de ventes aux enchères tels qu'eBay et Paypal. Les adeptes de l'hameçonnage envoient habituellement des courriels à un grand nombre de victimes potentielles.

Typiquement, les messages ainsi envoyés semblent émaner d'une société digne de confiance et sont formulés de manière à alarmer le destinataire afin qu'il effectue une action en conséquence. Une approche souvent utilisée est d'indiquer à la victime que son compte a été désactivé à cause d'un problème et que la réactivation ne sera possible qu'en cas d'action de sa part.

Le message fournit alors un hyperlien qui dirige l'utilisateur vers une page web qui ressemble à s'y méprendre au vrai site de la société digne de confiance. Arrivé sur cette page falsifiée, l'utilisateur est invité à saisir des informations confidentielles qui sont alors enregistrées par le criminel.

En 2007, ces criminels informatiques ont changé de technique en utilisant un moyen de piratage appelé attaque de l'homme du milieu pour recueillir les informations confidentielles données par l'internaute sur le site visité.

Il existe différentes variantes :

- ✓ le « spearphishing », qui vise une personne précise, par exemple sur des réseaux sociaux;
- ✓ l' « in-session phishing », qui tente de récupérer la session utilisateur durant la navigation.

3.3.6 Attaque DDOS

C'est une attaque d'un pirate, sur un serveur informatique, de façon à l'empêcher d'offrir le service pour lequel il est destiné.

Les victimes du déni de service ne sont pas uniquement celles qui le subissent, les postes compromis (daemons et masters) et les postes clients qui n'arrivent pas à accéder aux services désirés sont également les victimes des pirates qui effectuent le DoS. De nos jours, le piratage peut être acquis aisément, l'attaquant peut donc être un utilisateur lambda tant que son poste est relié au réseau mondial.

Le but principal est que l'accès au serveur d'une entreprise devienne impossible aux clients, le but n'étant pas d'altérer les données contenues et échangées ni de voler des informations, mais plutôt de nuire à la réputation de l'entreprise en empêchant l'accès aux divers services fournis aux clients en provoquant un ralentissement significatif ou une saturation du système voire le crash du système. A l'origine, les pirates n'étaient intéressés que par la renommée d'avoir réussi à faire tomber un réseau. Aujourd'hui, la raison de ces attaques est le chantage, en effet ces criminels sont principalement motivés par l'argent.

On a deux types de DoS :

- ✓ Déni de service par saturation : submerger une machine d'un grand nombre de requête afin qu'elle ne soit plus apte à répondre aux demandes des clients.
- ✓ Déni de service par exploitation de vulnérabilités : exploiter une faille du système dans le but de le rendre inutilisable.

Le principe est d'envoyer une très grande quantité de paquets, dont la taille est relativement importante, en même temps, voire sur une longue période.

Le principe du Distributed Denial of Service (DDoS) consiste à utiliser une grande quantité de postes « Zombies », préalablement infectées par des « backdoors » ou « troyens », dans l'intention de paralyser la réponse du serveur attaqué. Les maîtres sont eux-mêmes reliés aux postes « daemons ».

Le pirate se sert des postes maîtres pour contrôler les postes daemons qui effectueront l'attaque, sans cela, le pirate devrait se connecter lui-même à chaque daemons ce qui serait plus long à mettre en place, et plus facilement repérable.

Pour utiliser les masters et daemons, il est nécessaire d'exploiter des failles connues (FTP…). Le pirate se connecte aux masters en TCP pour préparer l'attaque, ces derniers envoient les commandes aux daemons en UDP.

3.3.7 Attaque Heartbleed

Certains sites web nécessitent ou proposent un chiffrement de la navigation. Entrer un mot de passe, effectuer une transaction bancaire, avoir des échanges épistolaires requérant un certain niveau de discrétion : autant d'actions que le serveur va « chiffrer », c'est à dire rendre illisibles pour qui n'a pas la « clef ».

Concrètement, cette sécurisation des communications entre l'internaute et le serveur est généralement symbolisée par un verrou à côté de l'URL.

OpenSSL est un des services offrant cette protection aux internautes. Qui trouve une faille dans OpenSSL et parvient à obtenir les clefs de chiffrement, peut lire toutes les informations échangées entre l'internaute et le serveur. C'est ce qui vient de se passer : des experts ont découvert la faille et certains développeurs se disent en mesure de récupérer toutes sortes de données comme des mots de passe Yahoo ou des historiques de recherche sur le moteur de recherche crypté DuckDuckGo.

- **Pourquoi s'appelle-t-elle « Heartbleed » ?**

Heartbleed signifie en français « cœur qui saigne », ce qui n'aide pas vraiment à rendre les événements moins dramatiques. La faille a été baptisée ainsi par l'équipe d'OpenSSL car elle affecte une extension du logiciel qui se nomme « heartbeat », battement de cœur en français. La faille permettant à des données de fuiter, le nom était tout trouvé. Fait original, le bug a d'ores et déjà un visuel.

- **Comment la faille a-t-elle été découverte ?**

C'est un ingénieur de Google Security, NeelMehta, qui a découvert l'existence de la faille, probablement en menant une analyse de sécurité poussée. Une équipe d'ingénieurs de l'entreprise de sécurité informatique Codenomicon aurait également trouvé la faille.

- **Est-ce vraiment grave ?**

Plutôt, oui. Comme on l'a vu précédemment, des données sensibles peuvent être extraites par une personne qui exploiterait cette faille. Or, le logiciel en question est utilisé par deux sites sur trois. Une proportion à relativiser cependant : toutes les versions d'OpenSSL ne contiennent pas la faille. Certaines entreprises de sécurité parlent tout de même de millions de systèmes vulnérables. Plusieurs sites importants ont par ailleurs confirmé avoir été affectés, tels Imgur, OkCupid, Flickr, Redtube... Yahoo semble être le seul géant du Web à avoir été touché. Une liste non exhaustive est disponible sur Github. Une adresse propose également de tester si un

site est vulnérable ou non. Le plus grand dommage semble pour l'instant porter sur la confiance que les internautes plaçaient dans le petit cadenas à côté de l'URL...

- **Des pirates ont-ils pu récupérer vos données ?**

Oui. La faille serait apparue dans une version d'OpenSSL mise en ligne en 2012. En deux ans d'existence à l'insu de tous, il est tout à fait possible que des personnes mal intentionnées aient découvert le bug et l'aient exploité (ayant donc accès à certains de vos compte ou profils). Il n'y a cependant pour l'heure aucune preuve que ça ait été le cas.

- **Devez-vous changer tous vos mots de passe ?**

La faille maintenant identifiée, les codes et serveurs vont être mis à jour rapidement. Yahoo par exemple a déjà annoncé avoir sécurisé ses serveurs centraux, et devrait avoir rapidement terminé. Il faut également que tous les sites renouvellent leurs clefs, ce qui coûte de l'argent et du temps.

3.4 SOLUTION POUR L'APPLICATION : SERVER GLASSFISH

La sécurité est à propos de la protection des données, c'est la façon de prévenir l'accès non autorisé ou d'endommager les données qui est en stockage ou en transit. Le GlassFish Server est construit sur le modèle de sécurité Java, qui utilise un bac à sable où les applications peuvent fonctionner en toute sécurité, sans risque potentiel pour les systèmes ou les utilisateurs. La sécurité du système affecte toutes les applications dans l'environnement GlassFish Server.

Les fonctions de sécurité du système sont les suivantes:

- ✓ Authentification
- ✓ Autorisation
- ✓ Vérification des comptes
- ✓ Firewalls
- ✓ Certificats et SSL
- ✓ Outils de gestion de la sécurité du système

3.4.1 Authentification

L'authentification est la façon par laquelle une entité (un utilisateur, une application ou un composant) s'assure qu'une autre entité est bien celui qu'il prétend être. Une entité utilise la sécurité des informations d'identification pour s'authentifier. Les pouvoirs peuvent être un nom d'utilisateur, un mot de passe, un certificat numérique, ou autre chose. Habituellement, les serveurs ou les applications exigent que les clients s'authentifient. En outre, les clients peuvent exiger des serveurs une authentification. Lorsque l'authentification est bidirectionnelle, il est appelé l'authentification mutuelle.

Lorsqu'une entité tente d'accéder à une ressource protégée, GlassFish Server utilise le mécanisme d'authentification configurée pour cette ressource, pour déterminer si l'accès est autorisé. Par exemple, un utilisateur peut entrer un nom d'utilisateur et un mot de passe dans un navigateur web, et si l'application vérifie ces informations d'identification, l'utilisateur est authentifié. L'utilisateur est associé à cette identité de sécurité authentifiée pour le reste de la session.

3.4.2 Autorisation

L'autorisation, également connu sous le contrôle d'accès, est le moyen par lequel les utilisateurs ont obtenu l'autorisation d'accéder aux données ou effectuer des opérations. Une fois qu'un utilisateur est authentifié, le niveau d'autorisation de l'utilisateur détermine quelles opérations le propriétaire peut effectuer. L'autorisation d'un utilisateur est basée sur le rôle de l'utilisateur.

Java contrat Autorisation de conteneurs (JACC) est la partie de la spécification Java EE qui définit une interface pour les fournisseurs d'autorisation enfichables. Cela vous permet de mettre en place des modules de plug-in tiers pour effectuer l'autorisation. Par défaut, le GlassFish Server fournit un moteur simple, basé sur des fichiers d'autorisation conforme à la spécification de JACC. Vous pouvez également spécifier d'autres fournisseurs tiers JACC.

JACC fournisseurs utilisent les API Java Authentication and Authorization Service de (JAAS). JAAS permet aux services d'authentifier et d'appliquer des contrôles d'accès sur les

utilisateurs. JAAS implémente une version du cadre standard Pluggable Authentication Module (PAM) de la technologie Java.

3.4.3 Vérification des comptes

L'audit est le moyen utilisé pour capturer les événements liés à la sécurité dans le but d'évaluer l'efficacité des mesures de sécurité. GlassFish Server utilise des modules d'audit pour capturer des pistes d'audit de toutes les décisions d'authentification et d'autorisation. GlassFish Server fournit un module de vérification par défaut, ainsi que la possibilité de personnaliser les modules d'audit.

3.4.4 Firewalls

Un pare-feu contrôle le flux de données entre deux ou plusieurs réseaux, et gère les liens entre les réseaux. Un pare-feu peut être constituée de deux éléments matériels et logiciels. Les lignes directrices suivantes ont trait principalement au serveur GlassFish :

- En général, les pare-feu doivent être configurés de sorte que les clients peuvent accéder aux ports TCP/IP nécessaires. Par exemple, si l'auditeur de HTTP fonctionne sur le port 8080, configurer le pare-feu pour autoriser les requêtes HTTP sur le port 8080 seulement. De même, si les requêtes HTTPS sont mises en place pour le port 8081, vous devez configurer le pare-feu pour permettre les requêtes HTTPS sur le port 8081.
- Si l'invocation de méthodes à distance directes sur Internet (RMI-IIOP, Inter-ORB Protocol accès à partir de l'Internet) pour les modules EJB est nécessaire, ouvrir le port d'écoute RMI-IIOP.

  Remarque : L'ouverture du port d'écoute RMI-IIOP est fortement déconseillée, car elle crée des risques de sécurité.
- En architecture à double pare-feu, vous devez configurer le pare-feu externe pour permettre les transactions HTTP et HTTPS. Vous devez configurer le pare-feu interne pour permettre le plug-in de serveur HTTP de communiquer avec GlassFish Server derrière le pare-feu.

3.4.5 Certificats et SSL (HTTPS)

Les Certificats : également appelés certificats numériques, sont des fichiers électroniques qui identifient les personnes et les ressources sur Internet. Les Certificats permettent aussi une communication sécurisée et confidentielle entre deux entités. Il existe différents types de certificats :

- Les certificats personnels sont utilisés par des individus.
- Les certificats de serveur sont utilisés pour établir des sessions sécurisées entre le serveur et les clients grâce à Secure Sockets Layer (SSL).

Les certificats sont basés sur la cryptographie à clé publique, qui utilise des paires de clés numériques (très longs numéros) pour crypter, ou encoder, des informations afin qu'elles puissent être lues que par son destinataire. Le destinataire décrypte ensuite (décode) les informations à lire. Une paire de clés comporte une clé publique et une clé privée. Le propriétaire distribue la clé publique et la rend accessible à tous. Mais le propriétaire ne distribue jamais la clé privée, qui est toujours gardée secrète. Parce que les clés sont mathématiquement liées, les données chiffrées avec une clé ne peuvent être déchiffrées avec l'autre clé de la paire.

Les certificats sont émis par un tiers de confiance appelée autorité de certification (CA). Le CA est analogue à un bureau des passeports : il valide l'identité du titulaire du certificat et signe le certificat de sorte qu'il ne peut pas être falsifié ou altéré. Après qu'un CA signe un certificat, le titulaire peut présenter comme preuve d'identité et d'établir des communications confidentielles cryptées. Plus important encore, un certificat lie la clé publique du propriétaire à l'identité du propriétaire.

En plus de la clé publique, un certificat contient généralement des informations telles que les suivantes :

- ✓ Le nom du titulaire et une autre identification, telles que l'URL du serveur Web en utilisant le certificat ou l'adresse électronique d'un individu.
- ✓ Le nom de l'autorité de certification ayant émis le certificat.

- ✓ Une date d'expiration.

Les certificats sont régis par les spécifications techniques du format X.509. Pour vérifier l'identité d'un utilisateur dans le certificat royaume, le service d'authentification vérifie un certificat X.509, en utilisant le champ de nom commun du certificat X.509 que le nom principal.

Secure Sockets Layer : Secure Sockets Layer (SSL) est la norme la plus populaire pour la sécurisation des communications et transactions par Internet. Applications Web sécurisées utilisent HTTPS (HTTP sur SSL). Le protocole HTTPS utilise des certificats pour assurer les communications confidentielles et sécurisées entre le serveur et les clients. Dans une connexion SSL, les données envoyées par le client sont cryptées et seront décryptées lors de la réception.

Quand un navigateur web (client) veut se connecter à un site sécurisé, une négociation SSL se produit, comme ceci :

1. Le navigateur envoie un message sur le réseau demandant une session sécurisée (typiquement, en demandant une URL qui commence par https au lieu de http).
2. Le serveur répond en envoyant son certificat (y compris sa clé publique).
3. Le navigateur vérifie que le certificat du serveur est valide et est signé par une autorité dont le certificat est dans la base de données du navigateur (et qui est approuvé). Il vérifie également que le certificat de CA n'est pas expiré.
4. Si le certificat est valide, le navigateur génère une seule fois, l'unique session de clé et la chiffre avec la clé publique du serveur. Le navigateur envoie alors la clé de session cryptée sur le serveur de telle sorte qu'ils ont une copie.
5. Le serveur déchiffre le message en utilisant sa clé privée et récupère la clé de session.

Après la poignée de main, le client vérifie l'identité du site Web, et seul le client et le serveur Web détiennent une copie de la clé de session. En avant, le client et le serveur utilisent la clé de session pour crypter toutes leurs communications avec l'autre. Ainsi, leurs communications sont assurées d'être sécurisées.

La nouvelle version de la norme SSL est appelée Transport Layer Security (TLS). Le GlassFish Server prend en charge le protocole SSL 3.0 et les protocoles de chiffrement TLS 1.0.

Pour utiliser SSL, GlassFish Server doit avoir un certificat pour chaque interface externe ou l'adresse IP qui accepte des connexions sécurisées. Le service de HTTPS de la plupart des serveurs Web ne fonctionnera que si un certificat a été installé.

3.4.6 Outils de gestion de la sécurité du système

GlassFish Server fournit les outils suivants pour gérer la sécurité du système :

- **Console d'administration**

La console d'administration est un utilitaire basé sur le navigateur utilisé pour configurer la sécurité pour l'ensemble du serveur. Les tâches comprennent la gestion des certificats, les utilisateurs, les groupes et les royaumes. D'autres tâches de sécurisation de l'ensemble du système sont aussi effectuées. Pour une introduction générale à la console d'administration, nous avons ci-dessus l'interface de la console d'administration.

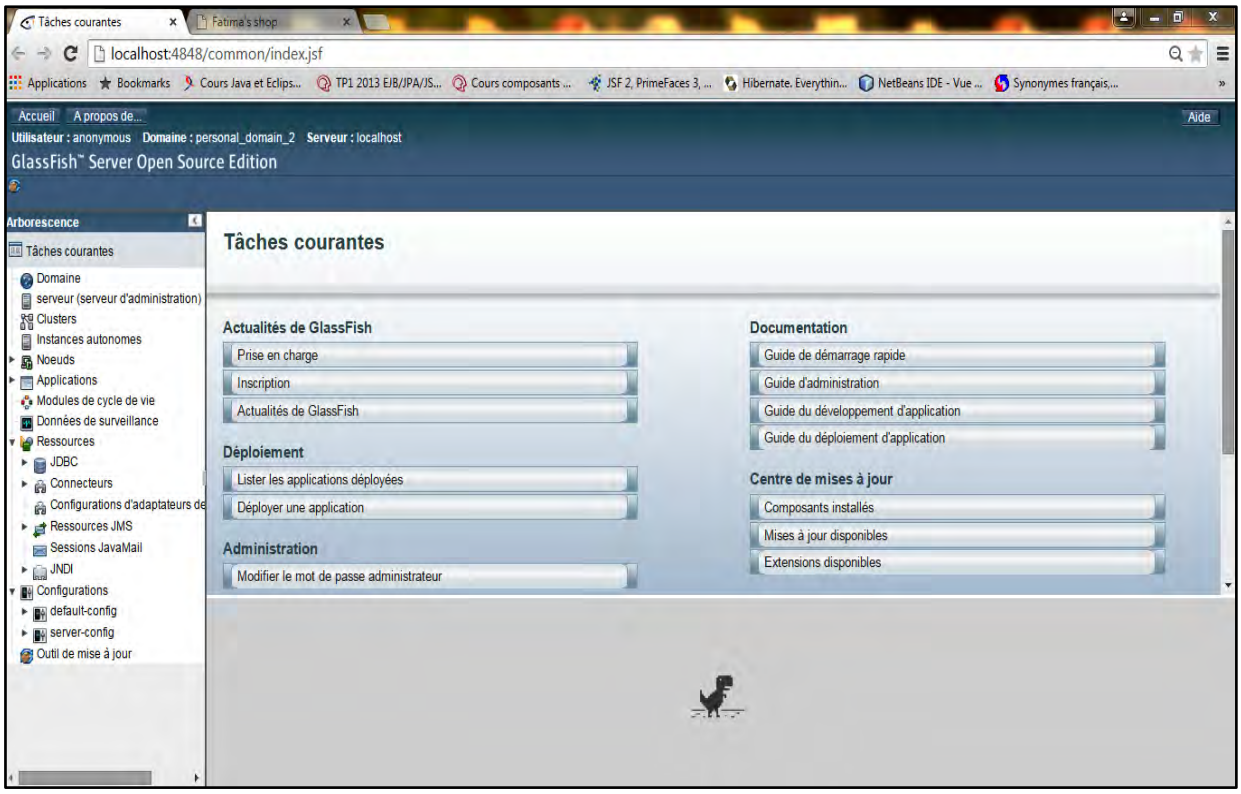


Figure 3-1 : Page d'accueil de la console d'administration du serveur GlassFish

Figure 3-2 : Partie configuration du serveur GlassFish

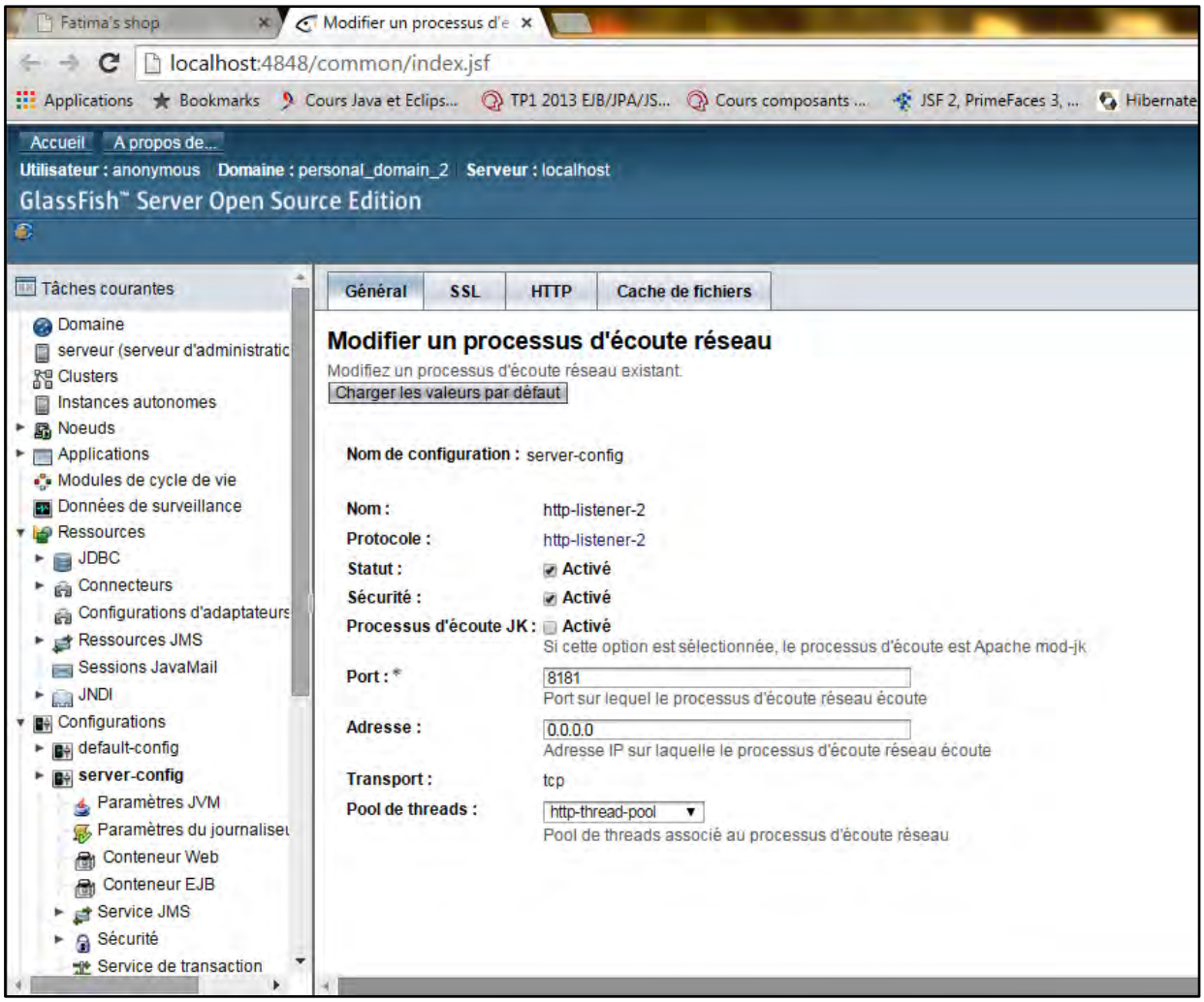


Figure 3-3 : Modification d'un processus d'écoute

- **Le asadmin utilitaire**

Le asadmin utilitaire de ligne de commande effectue un grand nombre de tâches identiques à celles de la console d'administration. Ainsi vous pourriez être en mesure de faire des choses avec le asadmin utilitaire si vous ne pouvez pas faire avec la console d'administration.

- **Le keytool utilitaire**

Le keytool Java Platform, Standard Edition (Java SE) utilitaire de ligne de commande est utilisé pour la gestion des certificats numériques et des paires de clés.

- **Le policytool utilitaire**

Le policytool utilitaire graphique de J2SE est utilisé pour gérer les politiques de sécurité de Java échelle du système. En tant qu'administrateur, vous utilisez rarement policytool.

CONCLUSION

Nous avons effectué ce travail dans le but d'obtenir notre diplôme de Master II en Transmission des Données et Sécurité de l'Information.

A cet effet, nous avons choisi le sujet « conception et implémentation d'une application E-commerce » dans le but d'exposer, à travers le monde, nos produits disponibles dans notre boutique afin d'encourager nos sœurs musulmanes à s'habiller décemment et avec élégance.

Ainsi, ce travail nous a permis d'une part de mettre en pratique certaines de nos connaissances acquises durant notre formation et d'autres part, il nous a donné l'opportunité de découvrir d'autres techniques et technologies de Java EE.

Ce qui nous a vraiment suscité à aimer plus la technologie Java EE et nous incite à vouloir améliorer notre application, en y ajoutant d'autres fonctionnalités.

BIBLIOGRAPHIE ET WEBOGRAPHIE

WEBOGRAPHIE

1. http://www.adie.sn/index.php/e-gouvernement/reglementation-tic/textes (03/2015)
2. http://www.developpez.net/ (05/2015)
3. http://www.ekladata.com/mba.eklablog.com/perso/architecture20logicielle/exposes2012/hibernate%20.jpa.docx (05/2015)
4. http://www.glossaire.infowebmaster.fr (2015)
5. http://www.horstmann.com/corejsf/jsf-tags.html (2015)
6. http://www.jebossedansleweb.com/programmation-quel-meilleur-langage-debuter/ (05/2015)
7. http://www.jmdoudoux.fr/java/dej/chap-spring.html (2015)
8. http://www.jsftoolbox.com/documentation/facelets/ (2015)
9. http://www.leppf.fr/spip.php?article35 (2015)
10. http://www.modatoi.com (12/2014)
11. https://www.netbeans.org/kb/docs/web/security-webapps.html (12/2014)
12. http://www.open-source-guide.com/Actualites/Les-langages-de-programmation-les-plus-populaires (05/2015)
13. http://www.oracle.com/technetwork/java/javase/jaas/index.html (03/2015)
14. http://www.primefaces.org/ (12/2014)
15. https://www.rocq.inria.fr/secret/Anne.Canteaut/crypto_moderne.pdf (05/2015)
16. http://www.soocurious.com/fr/programmation-internet-langage/ (05/2015)
17. http://www.sourceforge.net/projects/nbpfcrudgen (12/2014)
18. http://www.techno-science.net/?onglet=glossaire&definition=11378 (05/2015)
19. http://www.tutorialspoint.com/jsf/jsf_page_navigation.htm (2015)
20. http://www.visionsdeveloper.com/blog/page/ejb3-vs-spring-framework.jsp (05/2015)
21. http://www.wikipedia.org/ (05/2015)
22. https://www.youtube.com (05/2015)

BIBLIOGRAPHIE

1. 285022-creez-votre-application-web-avec-java-ee (télécharger dans le site http://openclassrooms.com)
2. Livre blanc sur la persistance (http://www.ippon.fr)
3. Les Cahiers Du Programmeur Java.EE.5 (Eyrolles)
4. Mémoire de Mouhamed Rassoul SARR (un ancien de Master 2 TDSI)