

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

UNIVERSITE MOULOUD MAMMERRI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire de Fin d'Etudes MASTER ACADEMIQUE

Domaine : **Mathématiques et Informatique**

Filière : **Informatique**

Spécialité : **Conduite de projet informatique**

Thème

Conception et réalisation d'une application java EE pour la gestion des achats d'une clinique médicale « Cas : clinique les amandiers »

Mémoire soutenu publiquement le 23/09/2019 Devant le jury composé de :

Président : M^r RAHMANI R

Encadreur : M^r HABET M-S.

Examinatrice : M^{elle} BOUGCHICHE L.

Examinatrice : M^{elle} YESLI Y.

Présenté par :

) *M^r KECHAOUI Mohammed*

) *M^r HASSINI Ali*

Remerciements

Nous tenons à exprimer notre profonde gratitude à notre promoteur, pour nous avoir encadrés et guidés tout au long de notre projet, pour ses conseils judicieux et minutieusement prodigués.

*Nos plus vifs remerciements vont aussi à tous le personnel de
L'EHP Les Amandiers.*

Nous remercions tous les enseignants du département informatique de la faculté de Génie Electrique et Informatique de l'université MOULOUD MAMMERI pour leurs disponibilités, leurs conseils tout au long de notre cursus au sein de ce département.

Que les membres du jury trouvent ici nos remerciements les plus vifs pour avoir accepté d'honorer par leur jugement notre travail.

Nos sincères sentiments vont à tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce projet. En particulier nos chères familles et nos amis (es).

Dédicace

Nous dédions ce modeste travail à :

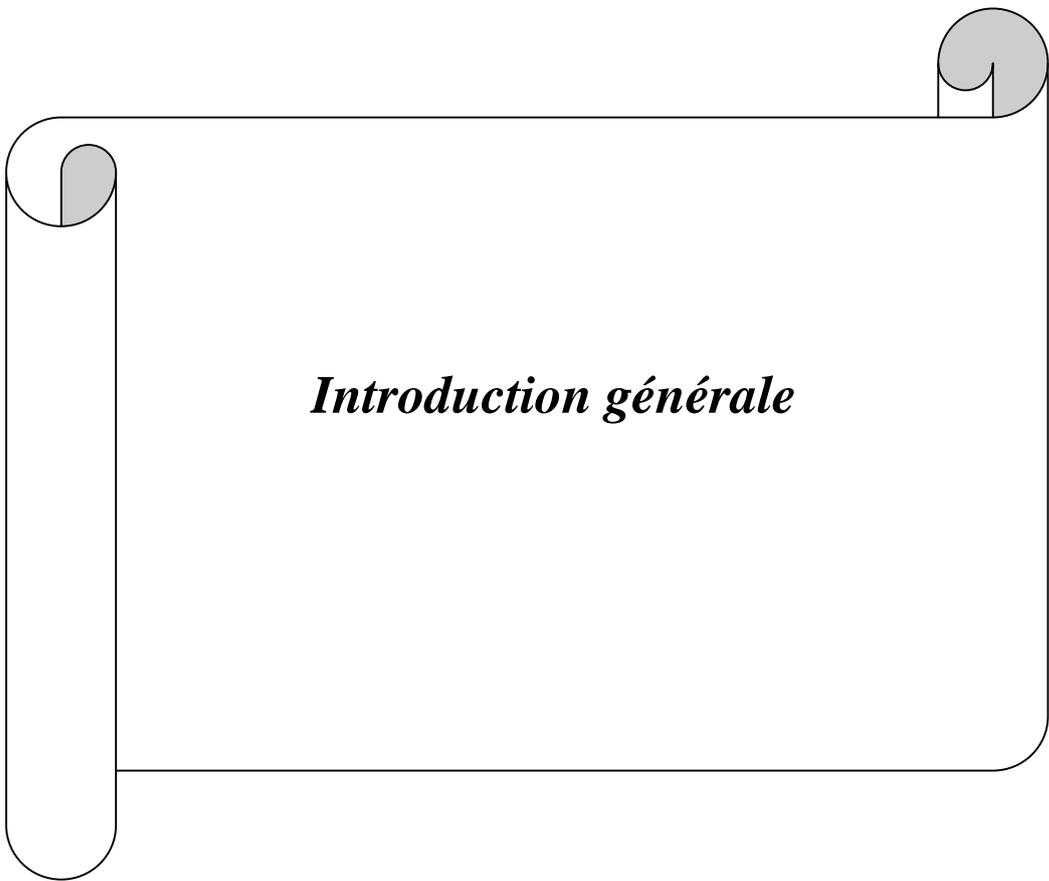
Nos chers parents,

Qui m'ont toujours fait confiance et n'ont jamais cessé

De m'encourager et me soutenir,

Ainsi qu'à Toutes nos familles sans exception,

Nos amis et nos camarades



Introduction générale

Introduction Générale

A l'aube du troisième millénaire le monde a connu une exponentielle avancée technologique dans tous les domaines précisément dans l'informatique, ce qui a donné naissance à d'innombrables applications dans tous les secteurs, l'administration, la gestion, l'industrie, la recherche ou même dans le domaine pédagogique et la santé.

L'informatique est l'art, la technique ou la science qui consiste à manipuler des informations. Elle a pour objet de définir des algorithmes qui permettent de modifier la vision que l'on a d'un problème, ou d'extraire d'une grande quantité d'informations mal structurées, à de nouvelles connaissances plus utiles.

Aujourd'hui, vu l'intérêt croissant de vouloir gagner en temps, de conserver les données, de limiter le nombre d'employés et pas mal d'autres raisons, ont poussé les petites, moyennes et grandes entreprises à chercher des solutions informatiques capables de répondre à leurs besoins. Notamment, celles qui relèvent de la professionnalisation des activités touchant au domaine médical.

Les applications et les réseaux informatiques représentent un atout pour les établissements sanitaires vu le taux élevé des informations qui circulent, la redondance, et la difficulté d'archivage. Dans ce cadre s'inscrit notre projet de fin d'étude qui consiste à réaliser une application web pour la gestion d'un centre médical.

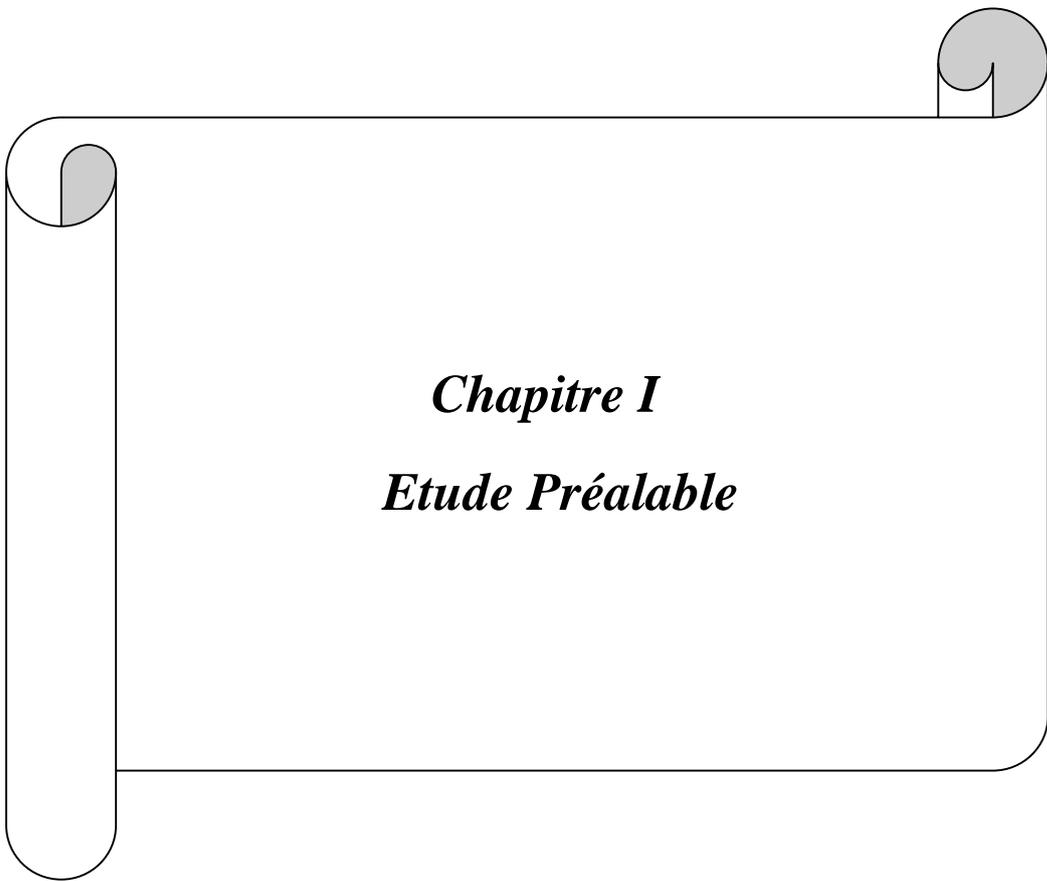
L'objectif de notre projet est de concevoir et réaliser un système informatique servant à gérer automatiquement les flux d'informations et les interactions entre les différents services existants au niveau d'un Etablissement Hospitalier Privé (EHP).

Pour atteindre cet objectif nous avons organisé notre mémoire en quatre chapitres :

- ❖ Chapitre I : « L'étude préalable » a pour objectif de présenter l'organisme d'accueil, son fonctionnement et ces difficultés, ainsi que la solution préconisée.
- ❖ Chapitre II : « Généralités sur les applications web » a pour objectif de présenter la technologie web, son fonctionnement, ainsi que le choix de l'architecture adoptée et outils utilisés pour le développement de notre application.
- ❖ Chapitre III : « Analyse et conception », ce chapitre est consacré à l'analyse et la conception de notre système à travers son architecture globale puis détaillée en définissant les différentes fonctionnalités attendues. Et pour modéliser notre solution nous avons utilisé le langage UML.

Introduction Générale

- ❖ Chapitre III : « Réalisation », décrit la réalisation en listant les outils utilisés ainsi que l'architecture générale de l'application, ainsi que quelque capture d'écrans décrivant quelques fonctionnalités sont données.



Chapitre I
Etude Préalable

1. Introduction

L'étude de l'opportunité (Etude préalable) dans un projet informatique est une étape qui consiste principalement à recenser l'existant c'est-à-dire les solutions informatiques déjà mises en œuvre dans l'entreprise et à recenser les besoins notamment en termes de fonctionnalités nouvelles.

Le but de l'étude préalable est de présenter l'organisme d'accueil et d'étudier l'existant. Cette dernière permet de regrouper tous les besoins des responsables, recenser toutes les tâches effectuées par les différents services ainsi que les documents en circulation. Une fois l'étude de l'existant réalisée, nous faisons une évaluation de la situation actuelle en faisant ressortir les besoins d'amélioration. Enfin, arrêter les nouvelles orientations de la solution proposée.

2. Présentation de l'organisme d'accueil

2.1. Historique de l'entreprise

L'Etablissement Hospitalier Privé « les Amandiers » qui a été créé le 01/02/2003 par le Docteur MELBOUCY Abdelaziz.

2.2. Fiche technique et coordonnées de l'EHP¹

-) **Nom** : Etablissement hospitalier privé les Amandiers.
-) **Secteur** : Privé.
-) **Directeur** général : Docteur MELBOUCY Abdelaziz.
-) **Téléphone** : (+213) 026 37 00 02
-) **Adresse** : Chaib ,Mekla , TiziOuzou , Algérie.

¹ EHP: entreprise hospitalier privé



Figure I.1 : Localisation de l'EHP.

2.3. Présentation des différents services sanitaires existants

Chirurgie générale:^[2] Branche de la chirurgie qui regroupe les opérations ne nécessitant pas une chirurgie spécialisée comme cela est parfois requis dans le cas d'opérations pratiquées sur un organe spécifique ou un système particulier.

Note : Il s'agit principalement de la chirurgie digestive, car en sont exclues la chirurgie thoracique et cardiaque, la neurochirurgie, l'orthopédie, l'urologie et l'obstétrique.

Pédiatrie:^[3] Branche spécialisée de la médecine qui étudie le développement psychomoteur et physiologique normal de l'enfant, ainsi que toute la pathologie qui y a trait, de la naissance à la période postpubertaire où il devient adulte. Le médecin spécialisé en pédiatrie s'appelle le pédiatre.

Chirurgie maxillo-faciale:^[4] Est la spécialité de la chirurgie prenant en charge les pathologies de la cavité buccale et de la face, tant dans le domaine chirurgical que le domaine médical.

Gynéco-Obstétrique:^[5] S'intéresse à la femme enceinte et à son enfant à naître. L'obstétricien a donc la charge de deux patients, et doit maîtriser un certain nombre de techniques

² <https://fr.wikipedia.org/wiki/Chirurgie>

³ <https://fr.wikipedia.org/wiki/P%C3%A9diatrie>

⁴ https://fr.wikipedia.org/wiki/Chirurgie_maxillo-faciale

complexes : médecine et chirurgie de la femme, mais aussi, souvent, échographie, pour le diagnostic prénatal de certaines affections du fœtus. Il ne doit pas être confondu avec la sage-femme, qu'on peut considérer comme le spécialiste de la grossesse normale, alors que le gynécologue obstétricien est spécialisé dans la prise en charge de la grossesse pathologique.

Chirurgie Viscérale:^[6] Est une des branches essentielles de la chirurgie traitant essentiellement les affections des organes abdomino-pelviens. Elle est également désignée sous les termes de chirurgie générale et chirurgie digestive. Ainsi, les chirurgiens viscéraux traitent les affections de l'appareil digestif mais également de la paroi abdominale ou encore d'autres organes intra-abdominaux non-digestifs tels que la rate ou des glandes endocrines comme la thyroïde.

ORL et Chirurgie Cervico-faciale:^[7] L'oto-rhino-laryngologie (ORL) est une spécialité médicale et chirurgicale dédiée au diagnostic et traitement des maladies de l'oreille, du nez, de la gorge et du larynx. La chirurgie cervico-faciale est une sous-spécialité de l'ORL qui s'intéresse à la prise en charge chirurgicale des affections de la face et du cou.

Chirurgie gynécologique:^[8] La chirurgie gynécologique est une chirurgie qui s'intéresse aux organes génitaux de la femme : vulve, vagin, utérus et ses annexes (ovaires et trompes), seins.

Chirurgie Orthopédique:^[9] L'orthopédie est la spécialité chirurgicale qui a pour objet la prévention et la correction des affections de l'appareil locomoteur, qui recouvrent les déformations et les malformations des os, des articulations, des muscles, des tendons et des nerfs. Le traitement chirurgical porte sur les membres supérieurs (épaule, coude et main), les membres inférieurs (hanche, genou et pied) et le rachis.

Imagerie Médicale:^[10] Spécialité appartenant aux sciences médicales, la radiologie désigne une technique d'imagerie médicale qui emploie soit des rayonnements ionisants, soit des rayons X pour obtenir des images de différentes structures anatomiques. Le recours à ce type d'applications médicales est principalement indiqué pour obtenir des clichés du squelette, des poumons, de la cavité abdominale ou encore des seins.

⁵https://fr.wikipedia.org/wiki/Gyn%C3%A9cologie_obst%C3%A9trique

⁶https://fr.wikipedia.org/wiki/Chirurgie_visc%C3%A9rale

⁷<https://www.agepa.org/quest-ce-quun-ork/>

⁸https://fr.wikipedia.org/wiki/Chirurgie_gyn%C3%A9cologique

⁹<https://fr.wikipedia.org/wiki/Orthop%C3%A9die>

¹⁰https://fr.wikipedia.org/wiki/Imagerie_m%C3%A9dicale

Oncologie:^[11] Ou carcinologie ou cancérologie est la spécialité médicale d'étude, de diagnostic et de traitement des cancers. Un médecin qui pratique cette discipline est appelé oncologue ou cancérologue.

Anapath:^[12] Est la discipline médicale qui permet la reconnaissance des anomalies des cellules et des tissus d'un organisme, appelées lésions, pour effectuer le diagnostic des maladies, porter un pronostic et, plus généralement, en comprendre les causes et les mécanismes.

Laboratoire D'analyse Médicale:^[13] Est un lieu où sont prélevés et analysés divers fluides biologiques d'origine humaine sous la responsabilité des biologistes médicaux, qui en interprètent les résultats dans le but de participer au diagnostic et au suivi de certaines maladies.

Médecine nucléaire:^[14] Comprend l'ensemble des applications médicales de la radioactivité en médecine. On administre au patient (par injection intraveineuse, ingestion, inhalation...) des médicaments radio-pharmaceutiques ; ceux-ci peuvent être des radionucléides isolés (comme l'iode 123 pour la glande thyroïde) ou être constitués d'un vecteur et d'un radionucléide. Une molécule « vectrice » qui fait partie du métabolisme humain va être attirée vers un organe cible ou tracer une fonction de l'organisme. Le radionucléide-traceur rattaché émet des rayonnements ionisants qui peuvent être détectés (imagerie ou scintigraphie) ou servir à détruire des cellules (radiothérapie interne vectorisée). Cette méthode est très sensible et permet des diagnostics ou des traitements.

Endoscopie:^[15] L'endoscopie ou fibroscopie est une méthode d'exploration visuelle médicale ou industrielle de l'intérieur d'une cavité inaccessible à l'œil. L'instrument utilisé, appelé endoscope ou fibroscope, est composé d'un tube optique muni d'un système d'éclairage. L'endoscopie médicale consiste en des examens spécifiques par organe. Elle peut être utilisée soit pour un diagnostic, soit pour une intervention.

¹¹ <https://sante.journaldesfemmes.fr/quotidien/2540946-oncologue-cancerologue-medecin-definition/>

¹² <http://www.chups.jussieu.fr/polys/anapath/Cours/POLY.Chp.1.html>

¹³ <https://sante-medecine.journaldesfemmes.fr/faq/20654-laboratoire-d-analyse-medecine-definition>

¹⁴ https://fr.wikipedia.org/wiki/Médecine_nucléaire

¹⁵ <https://fr.wikipedia.org/wiki/Endoscopie>

Hémodialyse:^[16] Technique permettant d'épurer le sang avec un filtre (rein artificiel) éliminant les déchets toxiques chez des sujets ayant une insuffisance rénale grave. Elle consiste à faire passer le sang du sujet au travers des membranes d'un circuit d'épuration extracorporelle. Celui-ci filtre le sang et le débarrasse des déchets normalement éliminés par les reins avant de le réinjecter au patient insuffisant rénal.

Pharmacie:^[17] La pharmacie hospitalière est une branche de la pharmacie se pratiquant à l'hôpital ou dans une collectivité (clinique privée).

Mission de service pharmacie :

- ✓ L'approvisionnement, le stockage, la distribution et la gestion d'un nombre considérable de produits pharmaceutiques.
- ✓ Le contrôle des factures et des mouvements des produits.
- ✓ Le contrôle des médicaments du point de vue conditionnement, date de péremption.
- ✓ Hygiène (contrôle de l'environnement...).
- ✓ La garde des produits toxiques et leur comptabilité selon le règlement en vigueur.

2.4. Présentation des différents services extra-sanitaires existants

Achats : Est un service bien structuré et bien organisé commence par une bonne définition de ses missions, de ses objectifs, et des fonctions (postes) des collaborateurs impliqués.

Missions du service Achat :

- ✓ Le cœur du métier consiste en premier lieu à satisfaire les besoins des clients internes, en obtenant la meilleure performance des fournisseurs (Qualité / Coût / Délais / Services) et en optimisant les coûts complets.
- ✓ Piloter l'amélioration continue au travers des relations fournisseurs.
- ✓ Protéger l'entreprise (gestion des ruptures d'approvisionnement, litiges, incidents).

Buanderie : Est un local plus ou moins vaste réservé à la lessive, aux lavages domestiques, artisanaux ou industriels. La buanderie a pour fonction principale d'être une laverie, cette

¹⁶ <https://fr.wikipedia.org/wiki/H%C3%A9modialyse>

¹⁷ <https://fr.wikipedia.org/wiki/Pharmacie>

première tâche implique différentes applications du simple lavage du linge jusqu'à son séchage en passant par le stockage et le repassage.

Comptabilité : La comptabilité au niveau de l'EHP est réalisée grâce à un comptable interne, qui fait la journalisation de toutes les entrées et sorties de l'EHP.

Mission de service comptabilité :

- ✓ La saisie de la fiche journalière de jour dans un registre contient une fiche de recette journalière de moi.
- ✓ Réalisation de la comptabilité chaque fin de mois.
- ✓ Calcul de la paie des employés.
- ✓ Comptabilisation des produits achetés par les différents services de l'EHP.
- ✓ Il permet d'établir à tout moment la situation financière de l'entreprise.
- ✓ Il permet les analyses des conditions d'exploitation en vue de réduire, de supprimer les faits et les pertes de l'entreprise et améliorer les diverses sources de bénéfices.

Cuisine : Manger c'est se procurer les nutriments indispensables au bon fonctionnement de l'organisme mais c'est aussi prendre du plaisir, s'inscrire dans le cadre social, relationnel et affectif, autant de facteurs indispensables au processus de guérison. La restauration doit alors être considérée non plus seulement comme une fonction logistique marginale, mais comme un service essentiel à rendre au patient

Mission de service cuisine :

- ✓ Répondre aux besoins nutritionnels de chaque patient.
- ✓ Adapter un régime pour un patient donné.
- ✓ La réalisation d'un autocontrôle sur les matières premières et produits finis.
- ✓ L'achat de certains produits.

Services personnel : Il s'agit d'un groupe de personnes qui travaillent en permanence pour satisfaire certaines tâches qui sont définies dans ce qui suit :

- ✓ Le suivi des nouveaux stagiaires et les dirigés vers les différents services de l'EHP.

- ✓ Accueil permanent des employés, marquage de leurs présences grâce à un appareil détection d'empreinte intégré à un logiciel qui enregistre toutes les informations concernant tous les employés de l'établissement.
- ✓ Recrutement de personnels manquant en cas de besoins.

2.5. Effectifs de l'EHP

Effectif médical : il est composé de :

-) Deux (2) chirurgiens pédiatres.
-) Deux (2) radiologues.
-) Cinq (5) généralistes.
-) Deux (2) anesthésistes.
-) Un (1) réanimateur.
-) Deux (2) néphrologues.
-) Un (1) médecin biologiste.
-) Un (1) oncologue.
-) Un (1) médecin anapath.

Effectif paramédical : ce service comprend :

-) Un (1) surveillant médical chef.
-) Quinze (15) infirmiers.
-) Six (6) infirmiers brevetés.
-) Une (1) aide-soignante.
-) Un (1) chef de bloc opératoire.

Effectif administratif : ce service comprend :

-) Huit (8) secrétaires.
-) Douze (12) agents administratifs.

Les services généraux et économique : ils se composent de :

-) Cinq (5) cuisiniers.
-) Une (1) buanderie.

-) Deux (2) chauffeurs.

2.6. Etat informatique de l'EHP

L'informatique dans les services :

La plupart des services de l'établissement hospitalier Privé (EHP) les amandiers disposent d'un logiciel pour traiter les informations d'une façon autonome des autres services.

Dans ce qui suit, les différents services avec les logiciels utilisés :

Hémodialyse

Logiciel(s) utilisé(s): CHIFA_DIALYSE.

Développé par : la CNAS.

Les Fonctions de Base de logiciel :

-) Gérer les patients dans le centre d'hémodialyse.
-) Gérer les rendez-vous des patients.
-) Gestion des séances d'hémodialyse.
-) Sauvegarder un dossier médical pour chaque patient se présentant au centre d'hémodialyse.
-) Gérer les Utilisateurs du logiciel CHIFA_DIALYSE.
-) La lecture de la Carte CHIFA et la Clé du Professionnel de santé.
-) La sauvegarde des actes prescrits sur la carte CHIFA par une Signature électronique.
-) Remboursement du Professionnel de santé par le transfert du Bordereau et Factures signées à la CNAS.

Configuration :

-) Système d'exploitation : Windows XP ou 7 32/64 bits.
-) Framework 4.0 et 3.5 pour XP.
-) Espace disque disponible supérieur à 500Mo.
-) SGBD : SQL Server 2005.



Figure I.2 : Logiciel CHIFA_DIALYSE

Radiologie

Logiciel(s) utilisé(s) : NX Viewer.

Développé par : Agfa-Gevaert.

Les Fonctions de Base de logiciel :

-) Traitement d'image.
-) Archivages des données patients.
-) Récupération de données à partir des équipements.



Figure I.3 : Logiciel NX Viewer.

Médecine Nucléaire

Logiciel(s) utilisé(s) : Scintron 7 MiE.

Développé par :

Les Fonctions de Base de logiciel :

-) Imagerie et scintigraphie.
-) Traitement d'image.
-) Archivages des données patients.
-) Récupération de données à partir des équipements.



Figure I.4 : Logiciel Scinttron 7 MiE.

Pharmacie

Logiciel(s) utilisé(s) : IntelliX.

Développé par : IntelliXgroup.

Les Fonctions de Base de logiciel :

-) Gestion des stocks.
-) Statistique.

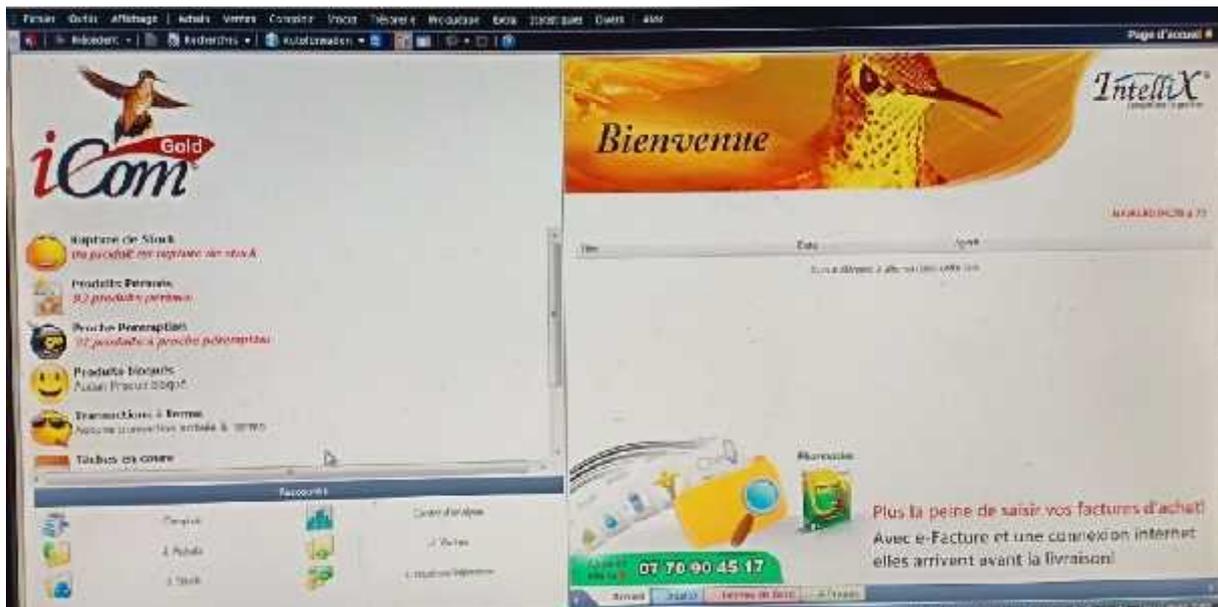


Figure I.5 : Logiciel IntelliX

Remarque : Pour le reste des services non citer dans ce qui précède, soit ils n'ont pas de logiciel pour traiter les flux d'informations ou ils utilisent des logiciels basiques (Word, Excel, ...).

3. L'étude de l'existant

Dans notre cas de recherche on s'intéresse à la partie Achat et Comptabilité de L'EHP les Amandiers, et les flux d'informations liées à ses deux services.

4. Organigramme de l'EHP

La Figure I.6 montre l'organisation générale de l'EHP (cas les amandiers) qui est représentée par un organigramme.

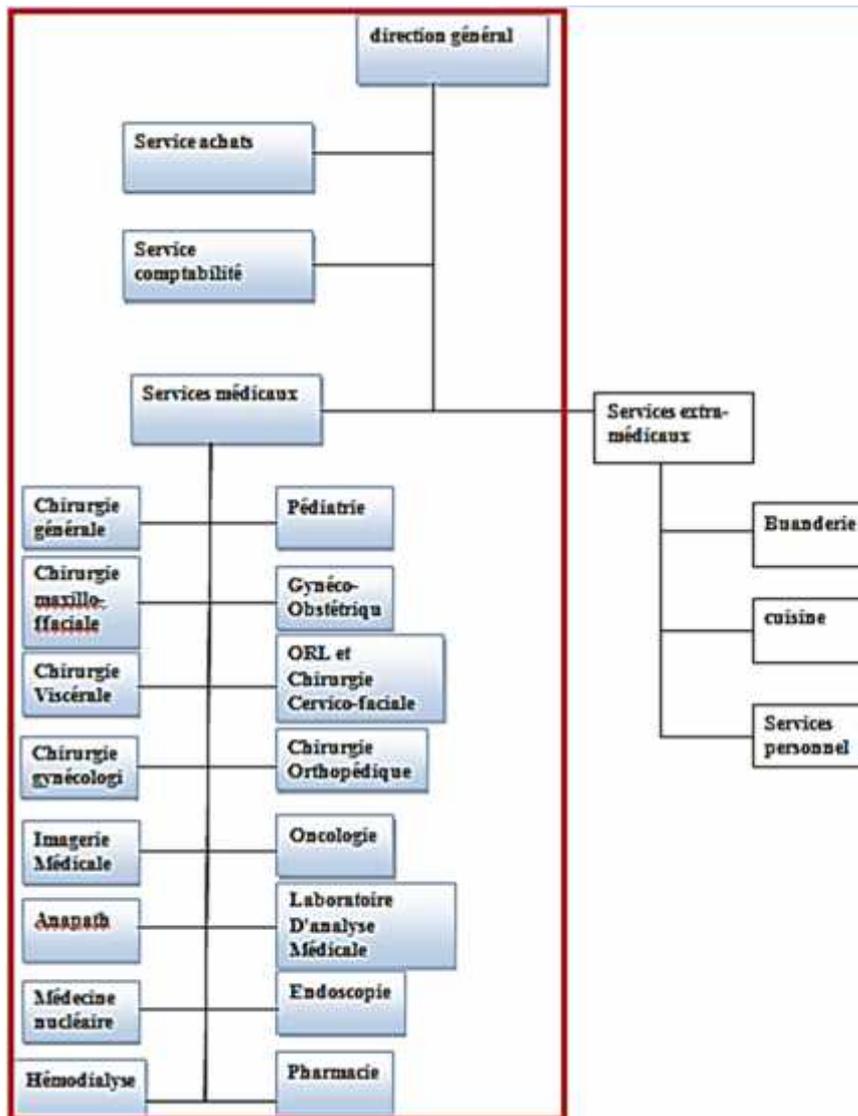


Figure I.6: Organigramme de l'EHP.

Comme la procédure d'achat à un usage quotidien aux sein de l'EHP est non informatisé, nous sommes focalisées sur cette dernière dans l'objectif est d'automatiser la communication entre le service achat et les autres services qui l'entoure.

Dans notre cas d'étude nous essayerons de représenter les interférences entre les différents service avec le service achats. Et représenté le flux d'information qui circulent.

5. Etude des Postes de travail

Le but de cette étape est de définir les différents postes qui sont liés à notre champ d'étude. Elle nous permet d'étudier le rôle de chaque poste. Les postes identifiés dans notre champ d'étude sont les suivants :

1. Service achats.
2. Service comptabilité.
3. La direction générale.
4. Autres Services.

Remarque : la procédure d'achats est la même au niveau des services demandeurs (service hémodialyse, la pharmacie, radiologie, médecine nucléaire...).

L'EHP les Amandiers est constitué de quatre (4) acteurs Principaux dans ce qui concerne les achats et la comptabilité (service achats, service comptabilité, La direction générale, Services demandeurs) chacun à un rôle spécifique qui est présenté ci-dessous :

Service achats :

- ✓ Etablissement de factures proforma.
- ✓ Contacte les Fournisseurs.
- ✓ Etablissement de bon de commande externe.
- ✓ Négociation des délais de paiement et de livraison.
- ✓ Etablissement de la demande de paiement.

Service comptabilité :

- ✓ Paiement des Fournisseurs.
- ✓ Réception des demandes de paiements.

La direction générale :

- ✓ Validation des demandes d'achats.
- ✓ Modification des demandes d'achats.
- ✓ Signature des demandes de paiement.
- ✓ Gestion opérationnelle.
- ✓ Représentation et promotion.

Service demandeur :

- ✓ Etablissement de Formulaire de demande d'achats.

- ✓ Identification des besoins.
- ✓ Réception de la livraison.

Dans le cadre de cette étude nous devons décrire pour chaque poste :

- ❖ Le code.
- ❖ Désignation de poste.
- ❖ Service de rattachement.
- ❖ Nombre d'effectifs.
- ❖ Responsabilités de tâches.
- ❖ Les documents établis, et diffusés par le poste.

5.1. Etude du poste de travail "Service achats"

Les détails concernant le poste Service achats sont présentés ci-après :

Fiche d'analyse du poste n°1

- Code : achat
- Désignation : Service achats
- Nombre d'effectifs :04

Responsabilités et Taches :

- Gestion des achats et des approvisionnements.

Documents établis par le poste

- Demande de factures proforma.
- Bon de commande externe.
- Demande de paiement.

Documents arrivants au poste :

- Formulaire de demande d'achats.
- Factures proforma.
- Bon de réception
- Bon de livraison + Facture avec prix
- Demande de paiement signée.

Origine :

- Services restants.
- Fournisseur.
- Services restants.
- Fournisseur.
- La direction générale.

Documents diffusés par le poste :

- Demande de Factures proforma.
- Bon de commande externe.
- Demande de paiement.
- Dossier de paiement

Destination :

- Fournisseur +La direction générale.
- Fournisseur.
- La direction générale.
- La comptabilité

5.2. Etude du poste de travail "Service Comptabilité"

Les détails concernant le poste Service Comptabilité sont présentés ci-après :

Fiche d'analyse du poste n°2

- Code : Compt.
- Désignation : Service Comptabilité
- Nombre d'effectifs :03

Responsabilités et Taches :

- La comptabilité

Documents établis par le poste

- Cheque de paiement.

Documents arrivants au poste :

- Dossier de paiement (Formulaire de demande d'achats, demande de facture proforma, facture proforma, bon de commande externe, la facture, Bon de réception, bon de livraison)

Origine :

- Services Achats.

Documents diffusés par le poste :

- Cheque de paiement.

Destination :

- Fournisseur.

5.3. Etude du poste de travail "Direction générale

Les détails concernant le poste Direction générale sont présentés ci-après :

Fiche d'analyse du poste n°3

- Code : DG.
- Désignation : Direction générale.
- Nombre d'effectifs :01

Responsabilités et Taches :

- Validation des demandes d'achats.
- Modification des demandes d'achats.
- Signature des demandes de paiement.

Documents établis par le poste

Documents arrivants au poste :

- Facture proforma.
- Dossier de paiement.

Origine :

- Services Achats.
- Services Achats.

Documents diffusés par le poste :

- Facture proforma signée.
- Dossier de paiement signé.

Destination :

- Services Achats.
- Services Achats.

5.4. Etude du poste de travail "Service demandeur "

Les détails concernant le poste Autres services sont présentés ci-après :

Fiche d'analyse du poste n°4

- Code : Autres.
- Désignation : Autres services.
- Nombre d'effectifs : /

Responsabilités et Taches :

- Identification des besoins.
- Réception de la livraison.

Documents établis par le poste

- Formulaire de demande d'achats.
- Bon de réception de livraison.

Documents arrivants au poste :

/

Origine :

/

Documents diffusés par le poste :

- Bon de réception de livraison.
- Formulaire de demande d'achats.

Destination :

- Service Achats
- Service Achats

6. Etude des procédures

Une procédure est un enchaînement de phases dont les opérations originales appartiennent au même processus et permettant de parcourir celui-ci en totalité. Dans notre cas d'étude nous intéressent à la procédure Achats au sein de l'EHP et les phases reliées à cette procédure.

6.1. Procédure Achats

Dans le tableau suivant, nous introduisons les tâches présentées dans la procédure Achat :

Numéro de la phase	Taches à exécuter
01	La réalisation de formulaire de demande d'achats par le service demandeur dans le cas où le stock est atteint un niveau bas (stock de sécurité).
02	La remise de formulaire de demande d'achats au service achats.
03	Etablissement de la demande de facture proforma par le service achats.
04	Contacteur les fournisseurs pour négocier le prix et les délais de livraison.
05	La réponse des fournisseurs sera par la remise d'une facture proforma.
06	Le service achat choisi le meilleur fournisseur par rapport aux contraintes (prix + délais de livraison).
07	Transfert de la facture proforma à la direction générale depuis le service achat pour valider l'achat.
08	La remise de la facture proforma validé par la direction générale aux mains de service achats.
09	Etablissement d'un bon de commande externe par le service achats.
10	Transmission de bon de commande vers le fournisseur.
11	Réception de la commande par le service demandeur.
12	Le service demandeur établie un bon de réception, ce dernier sera remis au service achats.
13	Le fournisseur envoi un bon de livraison + la facture au service achats.
14	Etablissement d'une demande de paiement remplis par le service achats et la transmettre au niveau de la direction générale pour la signer.
15	Remise de la demande paiement signé au service achats.
16	Le service achats établie un dossier composé des copie des documents suivants: formulaire demande achats, demande de facture proforma, facture prforma, bon de commande externe, bon de livraison, bon de réception, et facture qui seront archivés au niveau du service, les originaux seront envoyés au service comptabilité.
17	Service comptabilité va effectuer le paiement de fournisseur.

7. Le diagramme des Flux

Le diagramme des flux sert à représenter, de manière simple, les échanges d'informations entre les acteurs, l'analyse des flux permet de connaître les limites du domaine d'étude, ses activités et les principaux acteurs intervenants aux échanges d'informations.

Les flux sont présentés par une flèche, le nom du flux étant porté sur cette flèche.

L'acteur : c'est un agent capable d'échanger l'information avec les autres acteurs.

C'est l'information échangée entre deux acteurs dans le cadre d'un système d'informations.

Le diagramme des flux de notre domaine d'étude est représenté dans la figure I.9 suivante :

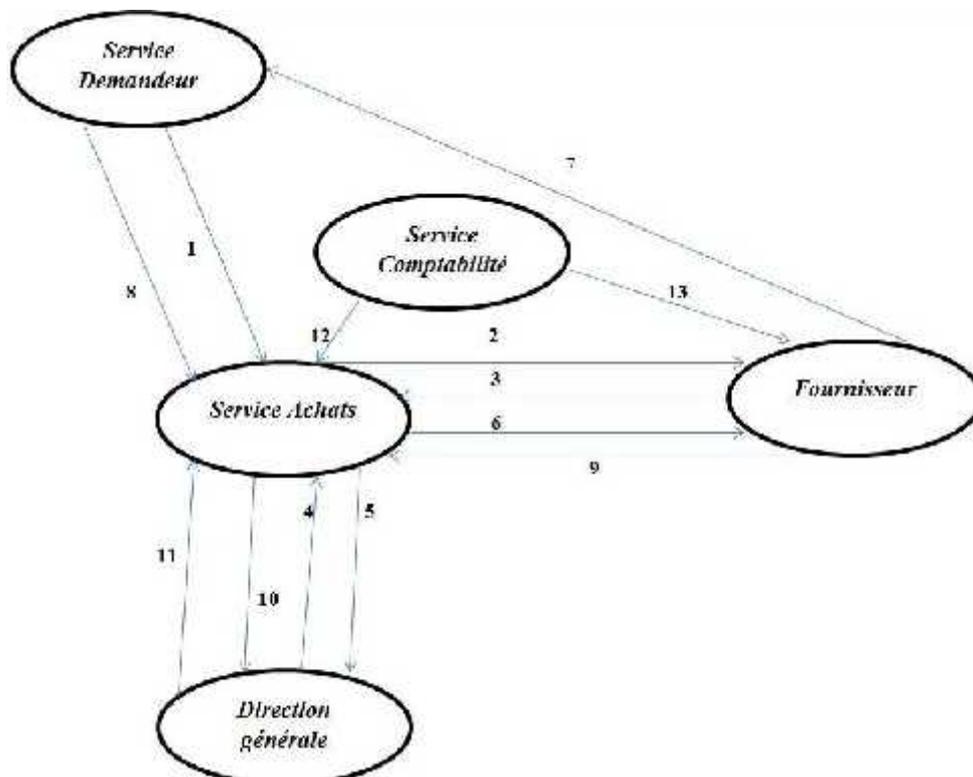


Figure I.7: Diagramme des flux

- 1 : Formulaire de demande d'achats.
- 2 : Demande de facture proforma.
- 3 : Renvoi de facture proforma.
- 4 : Facture proforma à valider.
- 5 : Facture proforma signée.
- 6 : Bon de commande externe.

- 7 : Livraison de la commande.
- 8 : Bon de réception.
- 9 : Bon de livraison+ facture.
- 10 : Demande de paiement.
- 11 : Demande de paiement signée.
- 12 : Dossier de paiement.
- 13 : Paiement (cheque ou espèce).

8. Etude des Documents

Les documents, dans une organisation, constituent les supports de transmission des informations, de leur stockage et de leur contrôle.

Les informations qui y sont contenues sont primordiales. Elles nous permettront de faire ressortir le dictionnaire de données et les relations entre les différentes rubriques et les contrôles qui peuvent être appliqués.

Leur étude nous servira à :

- ✓ Recenser les informations traitées.
- ✓ Couvrir d'éventuelles redondances.
- ✓ Déceler les principales causes du mauvais fonctionnement de la gestion de l'organisme concerné.

Elle se fera en deux parties :

1. Les caractéristiques du document servant à identifier :

- ✓ La désignation du document.
- ✓ Le rôle du document.
- ✓ Le mode d'utilisation du document :
- ✓ La nature du document.
- ✓ Le nombre d'exemplaire.
- ✓ Son origine et sa destination.

2. Éléments d'informations :

- ✓ La désignation de la rubrique.

- ✓ Type de la rubrique :
 - A : Alphabétique.
 - AN : Alphanumérique.
 - N : Numérique.
 - D : Date.
 - H : Heure.
- ✓ Taille.
- ✓ Observation.

La liste des documents à étudier est citée dans le tableau suivant :

La liste des documents à étudier est citée dans le tableau suivant :

Numéro	Désignation
01	Formulaire De Demande d'achat
02	Demande proforma
03	Facture proforma
04	Bon de commande
05	Bon de livraison
06	Facture
07	Bon de réception
08	Demande de paiement

La fiche d'étude du document " formulaire de demande d'achat"

Caractéristiques

Désignation : formulaire de demande d'achat.

Nature : Interne.

Rôle : L'inventaire de stock et la demande d'achats.

Identifiant : N° de la demande.

Nombre d'exemplaire : 02

Contenu

Désignation	Type	Longueur	Observation
N° de la demande	N	08	
Date-demande	D	08	JJ/MM/AA
Identité-demandeur	A	40	
Service	A	40	
Nom-fournisseur	A	40	
Désignation produit	AN	50	
Quantités-demandés	N	08	
Unité	A	30	
Etat actuel des stocks	N	08	
Quantité accordées par le DG	N	08	

La fiche d'étude du document " demande proforma"**Caractéristiques**

Désignation : demande proforma.

Nature : Externe.

Rôle : demande de renseignement sur les produits demander par le service demandeur.

Identifiant : numéro demande proforma.

Nombre d'exemplaire : 02

Contenu

Désignation	Type	Longueur	Observation
N° demande	N	08	
Fournisseur	A	40	
Adresse	A	50	
Tél	N	10	
Numéro produit	N	08	
Dénomination	A	50	
DCI	AN	50	
Quantité	N	08	
Unité	A	30	

La fiche d'étude du document " facture proforma"**Caractéristiques**

Désignation : facture pro-forma.

Nature : externe.

Rôle : présentation des produits demandé avec leur prix.

Identifiant : numéro de la facture

Nombre d'exemplaire : 02

Contenu

Désignation	Type	Longueur	Observation
Fournisseur	AN	40	
Numéro-facture	N	08	
Date	D	08	JJ/MM/AA
Destination	A	50	
REF	AN	30	
Désignation	AN	50	
PUHT	N	10	
QTE	N	08	
PTHT	N	10	
TOTAL HT	N	10	
TVA	N	10	
TOTAL TTC	N	10	

La fiche d'étude du document " bon de commande "**Caractéristiques**

Désignation : bon de commande.

Nature : Interne.

Rôle : choisir un fournisseur et procéder à l'achat.

Identifiant : Numéro bon de commande.

Nombre d'exemplaire : 02

Contenu

Désignation	Type	Longueur	Observation
Référence	AN	30	
N°BC	N	08	
Date	A	08	JJ/MM/AA
Fournisseur	AN	40	
Adresse	A	50	
Tél	N	10	
FPN°	N	08	
N° Produit	N	08	
Désignation-produit	A	50	
QTE	N	08	
Unité	AN	30	

La fiche d'étude du document " bon de livraison "**Caractéristiques**

Désignation : bon de livraison.

Nature : externe.

Rôle : prouvé que le produit est bien livré et réceptionné par le demandeur.

Identifiant : Numéro de bon de livraison.

Nombre d'exemplaire : 02

Contenu

Désignation	Type	Longueur	Observation
Destination	A	50	
Numéro BL	N	08	
Date	A	08	JJ/MM/AA
Référence produit	AN	30	
Désignation produit	AN	50	
LOT	AN	08	
DDP	A	08	JJ/MM/AA
QTE	N	08	

La fiche d'étude du document " facture"**Caractéristiques**

Désignation : facture.

Nature : externe.

Rôle : prix total de l'achat.

Identifiant : Numéro facture.

Nombre d'exemplaire : 02

Contenu

Désignation	Type	Longueur	Observation
Fournisseur	AN	40	
Numéro facture	A	08	
Date	N	08	JJ/MM/AA
BDC N°	N	08	
Date BDC	A	08	JJ/MM/AA
Désignation-produit	AN	50	
QTE	N	08	
TVA	N	10	
PUHT	N	10	
PTHT	N	10	
TOTAL TTC	N	10	

La fiche d'étude du document " Bon de réception"**Caractéristiques**

Désignation : Bon de réception.

Nature : Interne.

Rôle : L'accusé de réception des produits demandés.

Identifiant : N° de la demande de facture Proforma.

Nombre d'exemplaire : 02

Contenu

Désignation	Type	Longueur	Observation
Date	A	08	JJ/MM/AA
N° Proforma	N	08	
N° facture	N	08	
N° Bon de livraison	N	08	
Désignation	A	50	
Unité	AN	30	
Quantité	A	08	
N° LOT	AN	08	
DDP	N	08	JJ/MM/AA
Fournisseur	AN	40	
Fabriquant	AN	50	
Le réceptionniste	A	40	

La fiche d'étude du document " demande de paiement"**Caractéristiques**

Désignation : demande de paiement.

Nature : interne.

Rôle : Demande de paiement.

Identifiant : N° demande paiement.

Nombre d'exemplaire : 02.

Contenu

Désignation	Type	Longueur	Observation
Nom	A	20	
Prénom	A	20	
Référence	AN	30	
Date	A	08	JJ/MM/AA
Somme en chiffre	N	10	
Somme en lettre	A	80	
Fournisseur	A	40	
Service bénéficiaire	A	40	
Mode de paiement	A	35	
Demande d'achat N°	A	08	
DDPN°	N	08	
FPN°	N	08	
BDCN°	N	08	
BDLN°	N	08	
FN°	N	08	

9. Etude des registres

Parmi les supports d'information dont se servent les postes de travail afin d'archiver les informations exploitées, on trouve les registres. Leur étude se fera en trois parties :

) Les caractéristiques des registres :

- ✓ Désignation.
- ✓ Poste utilisateur.
- ✓ Rôle.
- ✓ Périodicité.

) Opérations réalisées :

- ✓ Opération.
- ✓ Réalisateur de l'opération.
- ✓ Période.

) Description du registre :

- ✓ Rubrique.

- ✓ Type.
- ✓ Taille.
- ✓ Observation.

La direction a besoin de deux registres :

- ❖ Le registre des achats.
- ❖ Le registre de la comptabilité.

Le registre des achats est utilisé par le service achats pour mentionner les achats avec un ordre chronologique. Ses caractéristiques, les opérations sur ce registre et son contenu sont illustrés au tableau suivant :

Caractéristiques du registre :			
Désignation : Registre des achats.			
Support : Registre.			
Poste utilisateur : Service achats.			
Rôle : Enregistrer les informations concernant les achats.			
Opération sur le registre :			
Ajout.			
Consultation.			
Etude de contenu du registre			
Désignation	Type	Taille	Observation
Date	N	8	JJ/MM/AA
Service demandeur.	A	40	
Désignation de produit.	A	40	
Quantité en stock.	N	08	
Quantité demandée.	D	08	
Quantité accordé par la Direction générale.	N	08	
Fournisseur.	AN	40	
N° Facture Proforma.	N	08	
N° Bon de réception.	N	08	

Figure I.14: Fiche d'étude du registre des rendez-vous.

Le registre de la comptabilité est utilisé par le service comptabilité pour mentionner les sommes versées à l'ordre des fournisseurs et le type de paiement (espèce, cheque). Ses caractéristiques, les opérations sur ce registre et son contenu sont illustrés au tableau suivant :

Caractéristiques du registre :			
Désignation : Registre de la comptabilité. Support : Registre. Poste utilisateur : Service comptabilité. Rôle : Enregistrer les informations concernant la comptabilité.			
Opération sur le registre :			
Ajout. Consultation.			
Etude de contenu du registre			
Désignation	Type	Taille	Observation
Date	N	8	JJ/MM/AA
Service demandeur.	A	40	
Désignation de produit.	A	40	
Quantité.	N	08	
Fournisseur.	AN	40	
N° Facture Proforma.	N	08	
N° Bon de réception.	N	08	
TOTAL HT	N	10	
TVA	N	10	
TOTAL TTC	N	10	
Type de paiement	A	10	

10. Critiques et suggestion

Gérer les achats est une tâche très difficile tant pour les acteurs de santé.

Critiques

Il existe trois types de critiques :

Critiques d'ordre informationnel :

- Risque de perte de données : difficulté liée à la consultation des fiches archivées des achats et de la comptabilité.
- La redondance d'information.

Critiques d'ordre Organisationnels

- Le mauvais archivage des fichiers : en effet les fichiers sont mal organisés, cela cause une perte de temps dans la recherche.
- La lenteur dans l'exécution des tâches (manuelles).
- Le déplacement des documents à chaque mise à jour ce qui implique une mauvaise localisation des documents.

Critiques d'ordre Techniques

- Recherche difficile sur les registres qui engendre une perte de temps.
- Nombre important des archives accumulées qui engendre une difficulté de stockage.
- Dégradation et régression des archives à force de leur utilisation trop fréquente.

Suggestions

Pour une meilleure gestion des achats au niveau de l'EHP, nous proposons l'introduction de l'outil informatique pour résoudre les problèmes engendrés par l'exécution manuelle de certaines tâches (recherche, ajouts, ct).

11. L'objectif de l'application

Notre application devra assurer les fonctionnalités suivantes :

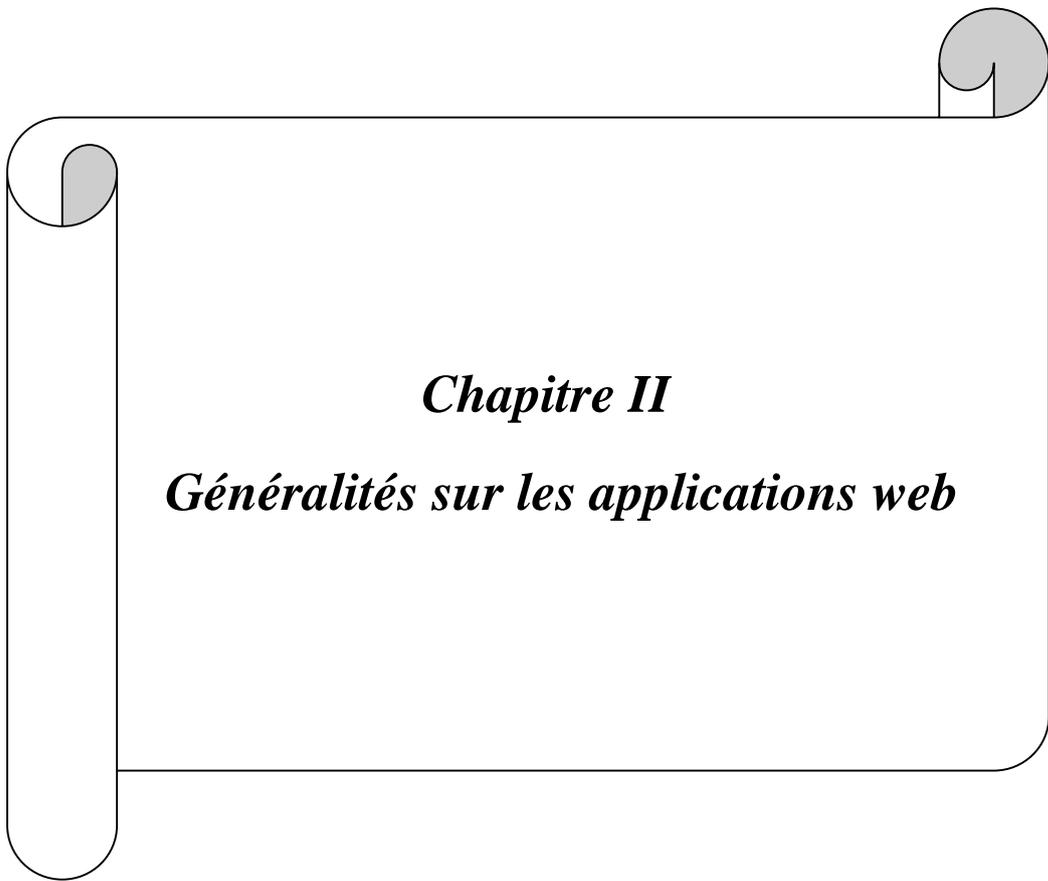
- ✓ Lier tous les services aux services achats.
- ✓ Listes prédéfinies des fournisseurs et des médicaments.
- ✓ Transmission direct des documents sans avoir recours au déplacement entre les services.
- ✓ Gérer les achats.

Les fonctionnalités optionnelles qui sont :

- ✓ Simple à manipuler.
- ✓ Sécurisation de l'information et de l'accès aux données.
- ✓ Un chat entre les services.

12. Conclusion

Nous avons, tout au long de ce chapitre, décrit fidèlement la situation actuelle de notre organisme d'accueil. Cette description nous a aidées d'une part à bien comprendre le fonctionnement général de cet organisme et d'autre part à relever les dysfonctionnements qui seraient à l'origine des difficultés que rencontre l'organisme.



Chapitre II
Généralités sur les applications web

1. Introduction

Notre objectif dans ce chapitre est de présenter l'architecture Logicielle offerte par la plate-forme java EE et les technologies de ses composants pour concevoir et développer des applications informatiques distribuées (comme les sites web dynamiques).

2. Présentation des applications web

2.1. Définition d'une application web dynamique [1]:

Selon Wikipédia une application web est une application manipulable directement en ligne grâce à un navigateur web et qui ne nécessite donc pas d'être installée. De la même manière que les sites web, une application web est généralement placée sur un serveur et se manipule en actionnant des widgets à l'aide d'un navigateur web, via un réseau informatique (Internet, intranet, réseau local, etc.)

Une application Web dynamique est un ensemble de pages statiques et dynamiques dont le contenu de ces pages est partiellement ou entièrement indéterminé. Le contenu final d'une page est déterminé uniquement lorsque l'utilisateur requiert une page depuis le serveur Web, et ce dernier varie d'une requête à une autre en fonction des actions de l'utilisateur.

Cette dernière est plus complexe sur le plan technique, car Elle utilise des bases de données pour charger des informations, et le contenu est mis à jour à chaque fois que l'utilisateur se connecte à l'application. En général, elle a un panneau d'administration depuis lequel l'administrateur peut corriger ou modifier le contenu.

Il existe de nombreux langages de programmation pour le développement d'applications web, parmi ces langages, Java EE celui que nous allons utiliser pour programmer notre application et que nous allons présenter dans la suite de ce chapitre.

Nb : *une **page web statique** n'est pas modifiée lorsqu'un visiteur la consulte : Le serveur Web transmet la page au navigateur qui la sollicite sans la modifier. A l'inverse, une **page web dynamique** est modifiée par le serveur avant d'être transmise au navigateur qui la sollicite. C'est pourquoi cette page est dite dynamique.*

¹ <https://fr.yeeply.com/blog/6-types-developpement-d-applications-web/>

2.2. Fonctionnement d'une application web [2]

Le web ce sont des machines communiquant en réseau à l'aide d'un langage commun. Parmi ces machines, on distingue celles qui proposent des ressources, les serveurs³, et celles qui les utilisent, les clients⁴. Les ressources peuvent être par exemple des documents hypertexte, des images, des fichiers XML ou encore des programmes (PHP, Java, ...) chargés de les générer à la demande.

Quand un client accède à une ressource(consulter un document, modifier des données stockées sur le serveur...etc.).Il doit utiliser le protocole de communication HTTP⁵ pour l'atteindre : ce dernier définit une sémantique très simple (GET, POST, et autres commandes) permettant de formuler des requêtes qui sont interprétées côté serveur par un programme spécifique qui est : le serveur web.

Lorsqu'un serveur Web reçoit une requête de page web statique, il transmet directement cette page au navigateur qui la demande.

Par contre lorsque le serveur Web reçoit une requête de page dynamique, il réagit différemment : Il transmet la page à un logiciel spécial chargé de terminer la page. Ce logiciel spécial est appelé un serveur d'application.

Le serveur d'application lit le code de la page, termine cette page en fonction des instructions figurant dans le code, puis en retire le code. Il en résulte une page statique que le serveur d'application renvoie au serveur Web, lequel transmet alors cette page au navigateur demandeur. Le navigateur reçoit uniquement du code HTML pur lorsque la page lui est transmise. Voici un aperçu du processus :

²<http://cours-fad-public.ensg.eu/mod/imscp/view.php?id=457>

³**Serveur** : processus accomplissant une opération sur demande d'un client et transmettant la réponse à ce dernier.

⁴**Client** : processus demandant l'exécution d'une opération à un processus serveur par l'envoi d'un message contenant un descriptif de l'opération à exécuter et attendant la réponse à cette opération par un message en retour.

⁵**Http** : HyperText Transfer Protocol, protocole de communication entre client et serveur permettant d'accéder à des ressources distantes.

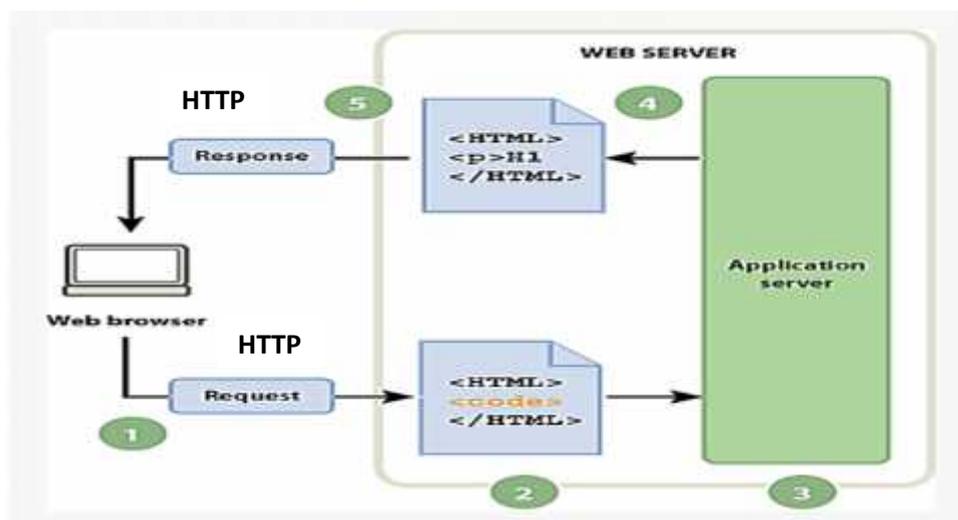


Figure II.1: Fonctionnement d'une application web.

Nb : sur le web, une ressource est identifiée par son url^[6] (chaîne de caractères). Lorsque le client souhaite atteindre une ressource distante, il émet donc une requête HTTP mentionnant l'url de la ressource.

2.3. Architecture web^[7]

2.3.1. Architecture à deux niveaux (client/serveur)

Aussi appelé l'architecture « client-serveur » est assez simple. Vous avez d'un côté le client et de l'autre le serveur. Ce genre d'architecture peut se faire sur tout type d'architecture matérielle interconnectée.

⁶Url : Uniform Resource Locator. Chaîne de caractères normalisée destinée à être utilisée par tout client web pour atteindre une ressource.

⁷ <https://www.supinfo.com/articles/single/574-architecture-2-tiers-vs->

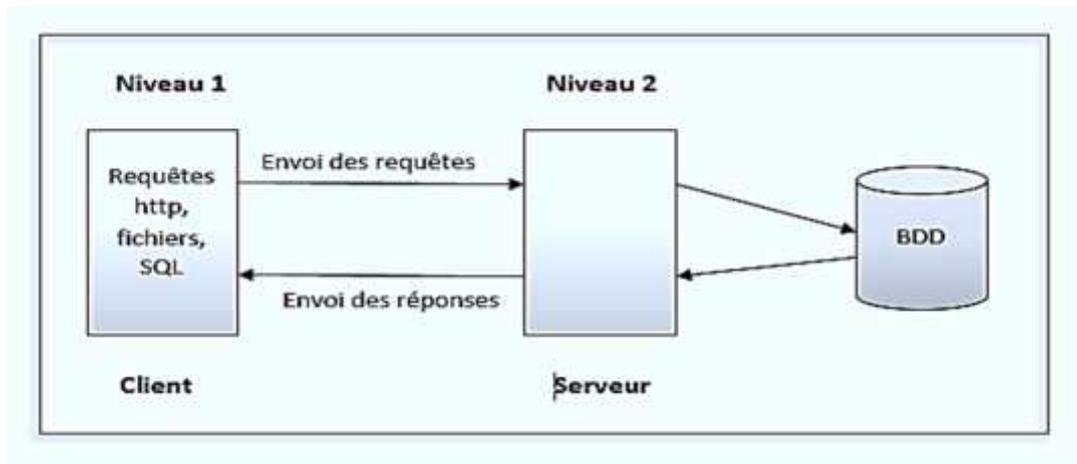


Figure II.1 : Architecture deux tiers.

- Le client demande un service au serveur comme par exemple la page contact.html.
- Le serveur reçoit cette requête http, effectue un traitement, et renvoie la ressource demandée par le client.

a) Avantages

-) C'est que tout d'abord les ressources sont centralisées sur le serveur. Il est donc plus simple de gérer les ressources communes aux utilisateurs comme la base de données par exemple.
-) Ensuite, cette architecture est plus sécurisée étant donné que le client dispose de moins de point d'entrée pour accéder aux données.

b) Inconvénients

-) Vu qu'il est le cœur du réseau, si celui-ci tombe, tout tombe. De plus, comparé à l'architecture à 3 niveaux, celui-ci est beaucoup moins flexible.

2.3.2. Architecture à trois niveaux

Dans l'architecture trois-tiers, un nouveau niveau fait son apparition. En effet, nous avons toujours le premier niveau qui est le client, ce dernier est très léger^[8] étant donné qu'il n'a aucun rôle de traitement. Au deuxième niveau nous avons le serveur d'application et enfin, au dernier niveau le serveur de base de donnée.

⁸Client léger :

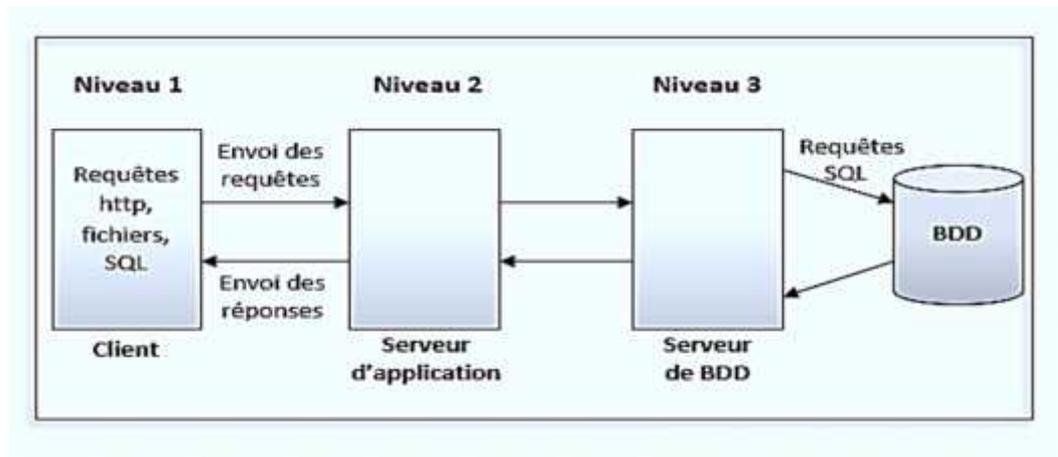


Figure II.2: Architecture à trois tiers.

Les points importants d'une application trois-tiers :

- 1- Le client ne sert qu'à requêter et à afficher les réponses du serveur.
- 2- Le serveur lui s'occupe des calculs et même de requêter des serveurs additionnels.

a) Avantages

-) Cette architecture étant plus divisé permet d'avoir du point de vue du développement, une spécialisation des développeurs selon le niveau de l'application.
-) Cette architecture offre une flexibilité beaucoup plus importante que l'architecture 2-tiers. En effet, la portabilité du tiers serveur permet d'envisager une allocation et ou modification dynamique aux grés des besoins évolutifs au sein d'une entreprise.

b) Inconvénients

-) Le souci c'est que cette architecture est le coût. D'après certaines études une architecture 3-tiers serait plus onéreuse.

2.3.3. Conclusion

Vu que notre application possède un panneau d'administration depuis lequel l'administrateur peut corriger ou modifier le contenu de cette dernière, et que nous allons avoir besoin d'une sécurité élevée.

Par conséquent, nous avons opté pour l'architecture à deux niveaux étant donné que, les ressources sont centralisées sur le serveur, ce qui rend la maintenance plus facile et que cette architecture est plus sécurisée.

2.4. Avantages des applications web^[9] :

-) Il n'est plus nécessaire d'installer un logiciel sur chaque poste, ce qui diminue en grande partie les frais de maintenance et élimine certaines incompatibilités ;
-) Une application bien conçue peut être utilisée par différents types de terminaux (ordinateurs, laptops, tablettes, smartphones) ;
-) Il n'existe plus de contrainte géographique et on peut envisager un accès pour les utilisateurs nomades en toute sécurité grâce au cryptage des données à travers Internet;
-) Des centaines de personnes réparties dans des lieux différents peuvent travailler simultanément sur ce type d'applications, car elles sont conçues pour supporter une grande charge qui est souvent répartie sur plusieurs serveurs ;
-) Certaines parties de ces applications sont ouvertes aux clients et fournisseurs et facilitent ainsi les échanges d'informations entre les partenaires (B2B) ;
-) Ces applications sont capables de gérer des données multimédia (textes, photos, vidéos) et les utilisateurs sont à l'aise car habitués à utiliser un navigateur dans le cadre privé ;
-) Pour mettre à jour l'application, il suffit de modifier l'application sur le serveur et tous les postes accèdent instantanément à la nouvelle version.

2.5. Notions de base ^[10]

2.5.1. Serveur web

Un serveur web n'est qu'un simple serveur de fichiers. Les clients s'adressent à lui via le protocole http afin d'obtenir une ressource. Lorsque le serveur Web reçoit une telle requête http, il extrait simplement de cette dernière, le nom de la ressource demandée ensuite va

⁹ https://www.webexpert.ch/publications/application_web

¹⁰ <https://www.supinfo.com/articles/single/6628-architecture-applications-java-ee>

chercher sur le disque la ressource et l'emballe dans une réponse http, afin de la transmettre au client. C'est le seul travail qu'est capable de réaliser un serveur web.

Le serveur web n'effectue aucun traitement sur la ressource avant de la transmettre au client. Il peut donc indifféremment transférer vers un client une page HTML¹¹, une image, un son voir même un fichier exécutable. Le type de contenu de la ressource demandée lui est totalement indifférent.

2.5.2. Serveur d'application

Le rôle d'un serveur d'application est radicalement différent car les ressources qui lui sont confiées ne sont pas de simples fichiers statiques, mais du code qu'il va se charger d'exécuter pour le compte des clients qui en font la demande. Lorsqu'un serveur d'application reçoit une requête http, il analyse lui aussi cette requête pour déterminer quelle ressource lui est demandée. Généralement la requête concerne du code exécutable hébergé par le serveur. Contrairement à ce que ferait un serveur web dans la même situation il ne transfère pas vers le client le code mais il l'exécute et c'est le résultat de ce code qui est renvoyé au client.

3. JavaEE

3.1. Présentation [12]

La technologie JEE (Java Entreprise Edition) constitue la solution proposée par Sun pour le développement d'applications distribuées. La base de cette solution repose évidemment sur le langage java conçu également par Sun. Ce langage de base est également connu sous le terme JSE (Java standard Edition).

En effet JEE n'est pas une amélioration de JSE. La version JSE est déjà suffisamment riche. JEE peut plutôt être considéré comme une norme décrivant tous les éléments constituant et intervenant pour le fonctionnement d'une application distribuée.

Elle définit par exemple les éléments suivants :

-) Comment doivent être développés les différents composants d'une application (servlet, pages JSP...).

¹¹HTML : (Hyper Text Markup Language) Langage de balisage utilisé pour la création de pages web.

¹²<https://www.supinfo.com/articles/single/6628-architecture-applications-java-ee>

-) Comment ces différents composants peuvent communiquer entre eux ou avec d'autres applications (JDBC, JNDI...).
-) Comment doivent être organisés ces différents composants pour constituer une application (descripteur de déploiement).

3.2. Architecture javaEE

L'architecture Java EE repose sur une architecture en couches qui, repose sur un découpage en couches ou tiers, nous parlons alors d'applications multi-tiers. Trois grands tiers sont ainsi représentés, dans lesquelles sont repartis différents composants que nous allons décrire par la suite.

L'objectif de ce découpage est de proposer une meilleure répartition des rôles (chaque couche a un rôle clairement défini) et une meilleure maintenabilité, étant donné que chaque couche est indépendante des autres.

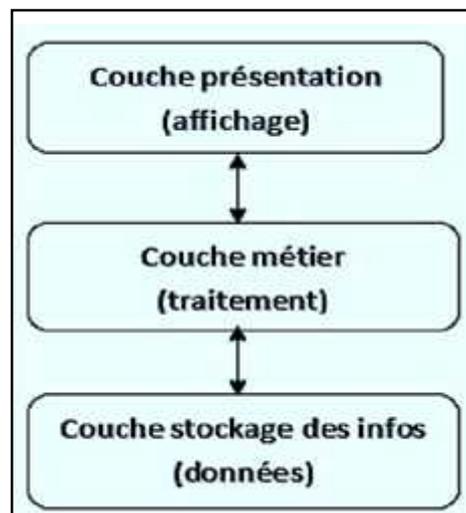


Figure 3: Architecture Java EE.

a) La couche présentation (tiers Web)

Gère l'affichage des données et les interactions de l'application avec l'utilisateur. La séparation de cette couche permet notamment de proposer plusieurs présentations pour une même application : la même couche de traitement peut alors servir pour une application lourde et pour une application légère.

b) La couche métier (tiers Métier ou tiers Business)

Concerne à la fois les tâches à réaliser par l'application sur les données et les traitements nécessaires suite à une action venant de l'utilisateur : vérification d'authentification, calculs divers, etc.

c) La couche stockage des informations (tiers Enterprise Information System)

Regroupe le stockage et les mécanismes d'accès des données de façon à ce qu'elles soient utilisables par l'application au niveau traitement.

3.2.1. Modèle MVC [13]

JEE (Java Enterprise Edition) est une plateforme qui permet de faciliter le développement d'application d'entreprise, en utilisant des bonnes pratiques de développement qu'on appelle les Patrons de conception ou Design Pattern (en anglais).

Un design pattern ou un modèle de conception est en quelque sorte une ligne de conduite qui permet de décrire les grandes lignes d'une solution de développement.

Dans notre application nous avons utilisé un modèle de conception qui est le pattern « MVC » qui nous a permis de bien organiser notre code source. Il nous a aidés à savoir quels fichiers créer, mais surtout à définir leur rôle. Le but de MVC est justement de séparer la logique du code en trois parties que l'on retrouve dans des fichiers distincts qui sont :

a- Modèle

Cette partie gère les données de notre application. Son rôle est d'aller récupérer les informations « brutes » dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur.

b- Vue

Cette partie se concentre sur l'affichage. Elle ne fait presque aucun calcul et se contente de récupérer des variables pour savoir ce qu'elle doit afficher. On y trouve essentiellement du code HTML mais aussi quelques boucles et conditions (Java) très simples, pour afficher par exemple une liste de messages.

¹³ : <https://openclassrooms.com/courses/adoptez-une-architecture-mvc-en-php/comment-fonctionne-une-architecture-mvc>

c- Contrôleur

Cette partie gère la logique du code qui prend des décisions. C'est en quelque sorte l'intermédiaire entre le modèle et la vue : le contrôleur va demander au modèle les données, les analyser, prendre des décisions et renvoyer le texte à afficher à la vue. Le contrôleur contient exclusivement du Java. C'est notamment lui qui détermine si le visiteur a le droit de voir la page ou non (gestion des droits d'accès).

3.2.2. Architecture MVC [14]

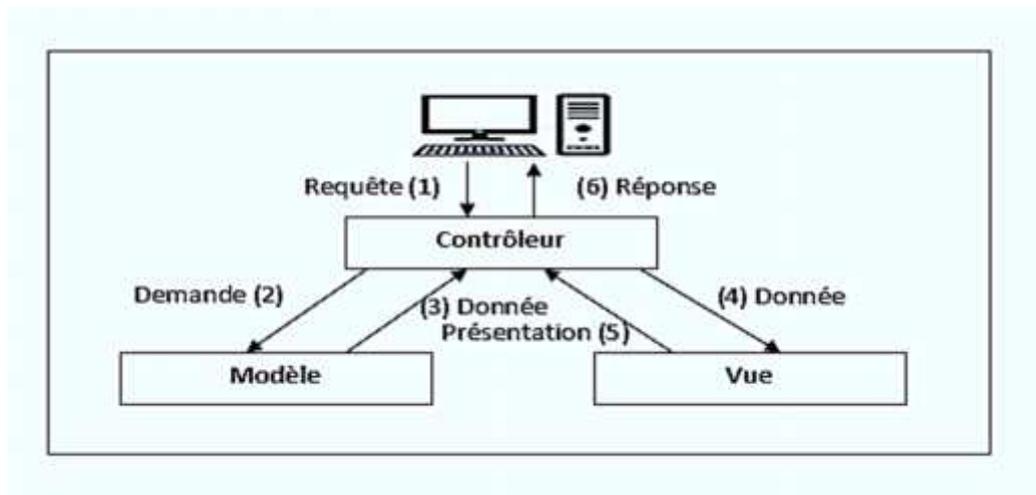


Figure II.4: Architecture MVC.

1. Le visiteur envoie sa requête http, la transmet au serveur d'application qui la transmet directement à la partie de notre code qu'on appelle le Contrôleur.
2. Le contrôleur fait le routage de l'information en décidant qui va récupérer l'information et ensuite la traiter, il appelle en effet le Modèle qui contient les informations structurées.
3. Le modèle va effectuer des calculs ou traitement sur ces informations et les renvoie au Contrôleur.
4. Le contrôleur va demander à la vue de générer une vue (Page Web).

¹⁴ : <https://www.supinfo.com/articles/single/6628-architecture-applications-java-ee>

5. La vue génère une page web selon la demande du contrôleur et la lui transmet.
6. Le contrôleur reçoit la page web qui sera envoyé au visiteur comme résultat de sa requête.

3.3. Les composants JEE [15]

Les applications sont découpées en plusieurs composants réalisant des fonctionnalités spécifiques et installés sur une machine serveur ou sur plusieurs tiers distribués. Les composants Java EE sont des unités autonomes assemblées dans une application Java EE composée de classes Java et de fichiers, et communiquant avec d'autres composants. De même, le code métier écrit est indépendant de la couche de présentation, ce qui est utile pour changer cette dernière ou pour l'afficher sur divers supports (Internet, Téléphonie...).

D'après le schéma ci-après, la spécification Java EE définit les composants suivants :

-) Les composants Clients ou tiers Client sont des applications clientes (logiciel installé en local ou navigateur Web ou Applets) s'exécutant sur la machine des utilisateurs.
-) Les composants Web ou tiers Web sont les technologies Servlets, JavaServer Pages et Java- Server Faces.
-) Les composants métier ou tiers Métier sont des composants Entreprises JavaBeans (EJB) représentant la logique métier, s'exécutant sur le serveur Java EE et dialoguant avec la couche de stockage (EIS : *Enterprise Information System*).

Une application Java EE multi-tiers correspond à un ensemble de couches. Souvent considérée comme étant une architecture trois tiers, celle-ci est distribuée de manière physique sur trois machines :

-) La machine des clients.
-) Le serveur d'applications Java EE.
-) Le serveur de base de données ou de stockage

¹⁵ : <https://www.supinfo.com/articles/single/6628-architecture-applications-java-ee>

3.3.1. Les composants clients ou tiers clients

La plate-forme Java EE propose trois types de clients : les clients Web, les Applets et les applications clientes riches type Java SE.

a- Clients web

Un client Web consiste en des pages Web de différents types (HTML, XHTML, XML, JavaScript ou autres) générées à partir de composants exécutés côté serveur dans un conteneur Web et capables de répondre aux requêtes HTTP en provenance du navigateur Internet. Ces programmes côté serveur sont représentés en Java EE par les Servlets, pages JSP et JSF. Ces programmes serveur ne réalisent en général pas directement les accès aux bases de données mais s'occupent de la logique applicative.

b- Clients Applets

Les Applets sont des interfaces graphiques Java SE exécutées dans un navigateur Web. Ces applications utilisent une interface graphique évoluée de type SWING et sont exécutées dans une machine virtuelle Java installée dans le navigateur. Cependant cette technique est plus contraignante à maintenir, requiert des accès et des droits pour la sécurité et nécessite un plug-in pour les navigateurs. Les clients Web sont donc préférables pour la création du tiers Client.

c- Applications Clientes

Une d'application cliente s'exécute sur un ordinateur client et permet aux utilisateurs de gérer des tâches nécessitant une interface utilisateur plus riche que celle fournie par un langage de balisage. Elle possède généralement une interface utilisateur graphique (GUI) créée à partir de l'API AWT (Swing ou Abstract Window Toolkit), mais une interface de ligne de commande est certainement possible. Les Applications clientes accèdent directement aux beans d'entreprise s'exécutant dans le niveau métier. Toutefois, si les exigences de l'application le justifient, un client d'application peut ouvrir une connexion HTTP pour établir une communication avec une servlet s'exécutant dans la couche Web. Les applications clientes écrites dans des langages autres que Java peuvent interagir avec les serveurs Java EE, ce qui permet à la plate-forme Java EE d'interagir avec les systèmes et clients qui ne sont pas développé en Java.

3.3.2. Les composants Web ou tiers web

Les composants Web Java EE sont des Servlets et/ou des pages JavaServer Pages et/ou des JavaServer Faces. Les Servlets sont des classes Java, capables d'intercepter et de gérer les requêtes du protocole HTTP. Les pages JSP sont des documents textuels exécutés comme des Servlets apportant une solution simplifiée pour la programmation de pages Web. La technologie JavaServer Faces est construite à partir de Servlets et fournit un Framework de développement pour accélérer la création d'applications Web.

3.3.3. Les composants métiers

Le code métier, qui est la logique qui résout ou répond aux besoins d'un domaine d'activité particulier tel que la banque, le commerce de détail ou la finance, est géré par des beans entreprise s'exécutant dans côté métier. La Figure ci-dessous montre comment un enterprise bean reçoit des données de programmes client, les traite (si nécessaire) et les envoie au niveau du système d'information d'entreprise pour le stockage. Un enterprise bean récupère également les données du stockage, les traite (si nécessaire) et les renvoie au programme client.

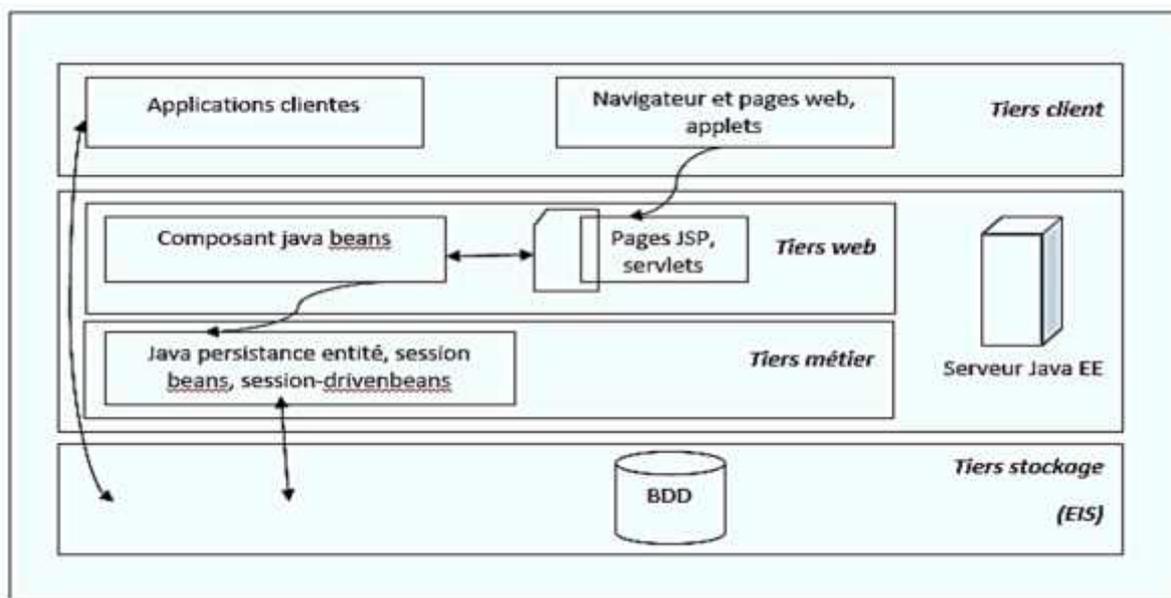


Figure II.5 : Composant JEE.

On constate dans cette architecture qu'on dispose de tous les composants (Client Tiers, Web tiers et Business Tiers) cités plus haut. Cependant, les applications Java EE utilisent un serveur d'application situé entre le serveur Web et la couche base de données. Le Serveur d'application va servir à exécuter les composants métier (EJB) et constitue ce qu'on appelle un environnement d'exécution des composants métier (EJB) qu'on appelle Conteneur d'application qui fournira tout ce qui est nécessaire pour l'exécution des JavaBeans. Le conteneur Web quant à lui servira d'environnement d'exécution des JSP et servlet, etc. On obtient ainsi l'architecture suivante

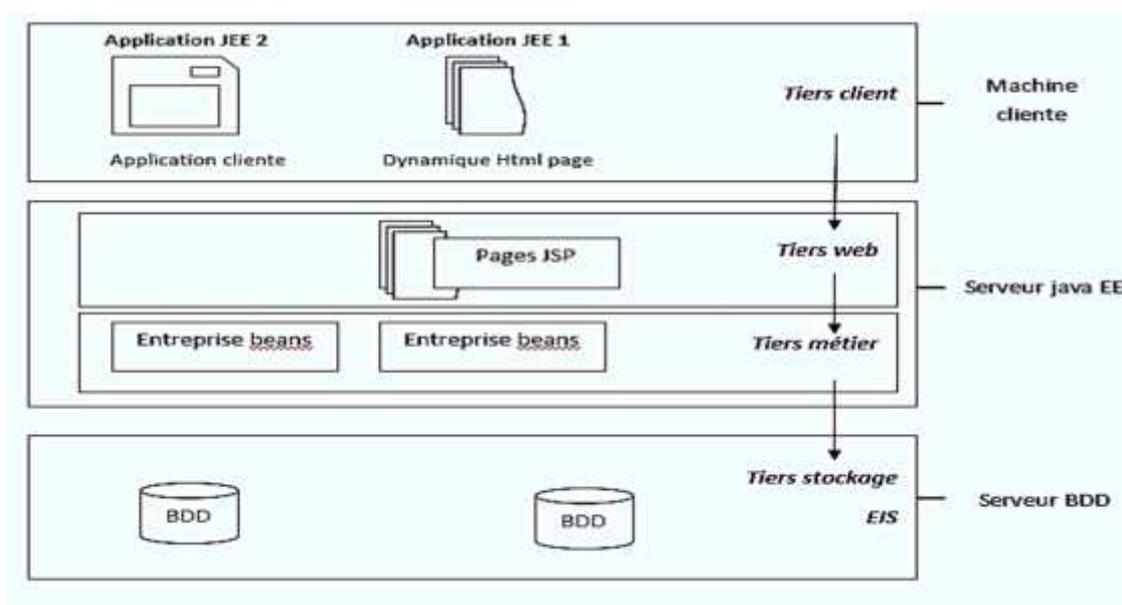


Figure II.6: Composants JEE.

3.4. Les conteneurs JAVA EE [16]

Les serveurs Java EE proposent plusieurs types de conteneurs (containers) pour chaque type de composant. Chaque conteneur a un rôle bien défini et offre un ensemble de services pour les développeurs :

-) L'annuaire de nommage d'accès aux ressources : Java Naming and Directory Interface (JNDI) est une interface unifiée de gestion de nommage pour les services et l'accès à ceux-ci par des applications.
-) L'injection dynamique de ressources.

¹⁶ <https://www.supinfo.com/articles/single/6628-architecture-applications-java-ee>

-) La gestion des accès aux bases de données.
-) Le modèle de gestion de la sécurité.
-) Le paramétrage des transactions.

Les exécutions et développements sont donc directement liés au conteneur utilisé et une application Java EE côté tiers Web ne sera pas programmée comme une application côté tiers EJB, par exemple. Les accès aux ressources nommées, les accès aux bases de données ou encore l'injection dynamique de ressources seront différents. Pour résumer, un conteneur permet de gérer le cycle de vie et la gestion des ressources.

Une application Java EE de type Web nécessite un conteneur Web pour son exécution alors qu'une application utilisant les EJB nécessite un conteneur EJB pour son exécution. Chaque conteneur propose un ensemble de services avec ses avantages et ses contraintes.

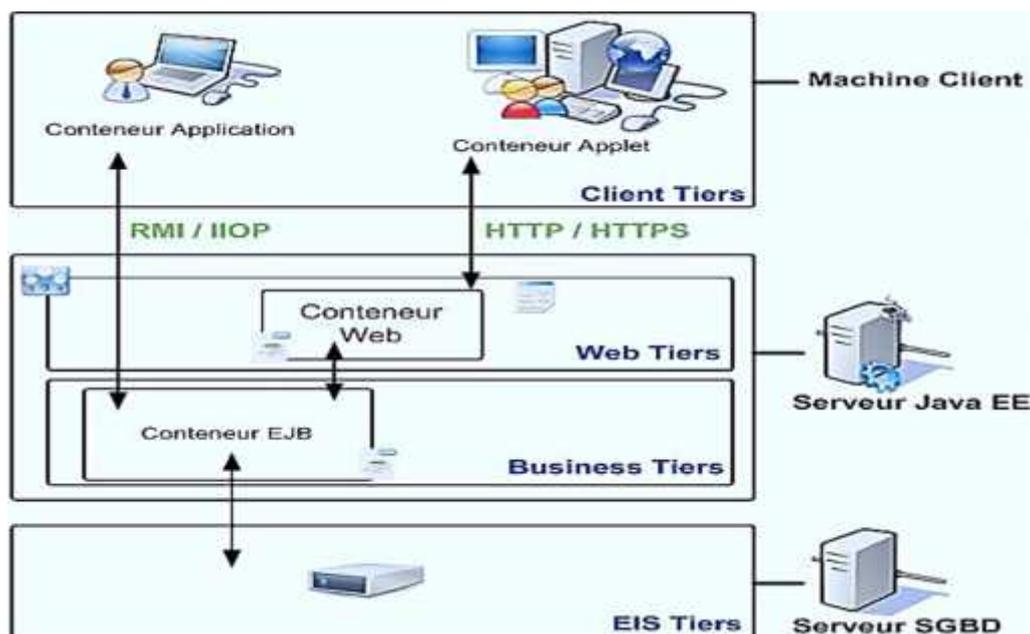


Figure II.7 : Conteneur.

L'architecture Java EE présentée dans la figure ci-dessus est découpée selon plusieurs domaines ou services, appelés des conteneurs ou containers. D'après les précédents schémas, il existe en Java EE cinq types de conteneurs :

- a- **Java EE serveur**, logiciel fournisseur du conteneur Web et EJB.

- b- Conteneur Web**, capable d'exécuter des pages JSP, Servlets et programmes JSF (implémentation JSF nécessaire). Ces composants s'exécutent avec le conteneur serveur Java EE. Ce conteneur supporte les protocoles HTTP et HTTPS et dialogue avec différents types de clients (navigateur, application).
- c- Conteneur EJB**, responsable de la couche métier et de l'exécution des EJB. Ce conteneur fournit le mécanisme de gestion des EJB, la gestion de leur cycle de vie, les transactions, le service de nommage JNDI et la possibilité des déclenchements asynchrones.
- d- Conteneur Applet**, responsable de l'environnement d'exécution des Applets Java. Ce type d'application consiste en un navigateur Web et le plug-in Java, lancés en même temps sur le poste client. Ces programmes sont lancés dans un « bac à sable » aussi nommé « sandbox » afin de gérer la sécurité des accès sur le poste client.
- e- Conteneur des applications clientes(ACC)**, permet l'exécution d'applications Java SE. L'application de type client et son conteneur sont lancés sur le client en simultanément. Le conteneur d'applications client (ACC) propose une bibliothèque de classes, un système de gestion de sécurité et le service de nommage JNDI pour les applications Java SE. La communication directe entre le conteneur client et le conteneur EJB est réalisée avec les technologies RMI-IIOP, alors que la communication avec le conteneur Web est réalisée avec le protocole HTTP pour les services Web, par exemple.

3.5. Les API JEE [17]

La plate-forme Java EE est la plus importante proposée par Oracle et regroupe les services de la plate-forme Java SE. Les services et API Java EE sont donc associés à une plate-forme et un ou plusieurs conteneurs. Les conteneurs étudiés précédemment proposent plusieurs services pour le développement d'applications. En tant que développeur, nous pouvons utiliser la totalité des services ou bien au contraire sélectionner uniquement les techniques nécessaires à la réalisation du projet. Chaque service s'exécute dans un des cinq conteneurs étudiés précédemment et offre d'autres méthodes en fonction de ce conteneur.

¹⁷ <https://www.supinfo.com/articles/single/6628-architecture-applications-java-ee>

3.5.1. Java servlet

Ce service Java EE s'exécute dans le conteneur Web et propose un ensemble de classes de gestion des échanges HTTP de type requête-réponse. Avec la nouvelle version de Java EE, la technologie Servlet 3.0 facilite la configuration et supporte les annotations ou encore les traitements asynchrones.

3.5.2. Java server pages (JSP)

La technologie JavaServer Pages (JSP) s'exécute dans le conteneur Web et utilise le mécanisme de génération de Servlets à partir de documents textuels. Les JSP 2.2 contiennent des données statiques de type HTML, XHTML, JavaScript ou autres, et des balises ou tags JSP nommés JSTL correspondant à du code Java.

3.5.3. Java server pages standard tag library (JSTL)

La technologie JavaServer Pages Tag Library (JSTL) est encapsulée dans des pages JSP et propose une librairie de balises XHTML standardisées. Ces balises proposent de gérer les conditions et itérations, l'internationalisation, les affichages HTML ou encore l'accès aux bases de données.

3.5.4. Java server faces (JSF)

La technologie JavaServer Faces s'exécute dans le conteneur Web et correspond à la réponse du consortium Java EE en matière de framework de développement côté serveur. JSF 2.0 est un framework de développement de projets Web proposant la construction rapide de code serveur, la validation des entrées, la gestion des modèles objets, la configuration de la navigation, les conversions de données, l'utilisation d'annotations, les balises et technologies d'affichages de données pour la partie Vue, le support de la technologie Ajax en natif et bien d'autres fonctionnalités. JSF est orienté composant et repose sur un cycle de vie proche du développement d'applications Java SE (écouteurs, évènements...).

3.5.5. Java DataBase Connectivity (JDBC)

La technologie Java DataBase Connectivity (JDBC) permet de gérer les commandes SQL et les dialogues avec les différents SGBD relationnels. JDBC est utilisée lorsque nous devons accéder à une base de données dans une Servlet, page JSP, page JSF ou autres. L'API JDBC propose un ensemble de classes et interfaces pour se connecter aux différents SGBD du marché mais également les paquetages de manipulation des données. JDBC est une technologie sous-jacente à JPA.

3.5.6. Java Persistence API (JPA)

Java Persistence API (JPA) est un standard Java utilisé pour la persistance des données. Ce mécanisme de persistance utilise le principe de mapping objet/relationnel et relationnel/objet afin de permettre de stocker les objets dans la base de données et inversement de pouvoir lire les données relationnelles et les transformer en objets.

-) L'API Java Persistence 2.0 propose les services suivants :
-) La gestion de la persistance.
-) Un langage de requêtage évolué : Java Persistence Query Language (JPQL).
-) Un mécanisme de mapping objet/relationnel ORM à partir de métadonnées (fichiers XML ou annotations).
-) Une API à typage fort pour le requêtage.

3.5.7. Enterprise JavaBeans (EJB)

Les composants Enterprise JavaBeans s'exécutent dans le conteneur dédié EJB et correspondent à la couche ou tiers Métier de l'application. Ces EJB 3.X sont des classes composées de champs et de méthodes d'accès. Java EE propose deux familles d'EJB : les sessions beans et message-driven beans.

Les sessions beans représentent les conversations entre le client et le serveur, qu'elles soient avec ou sans état. Un Message-Driven Bean (MDB) mélange un session bean et un écouteur de message permettant ainsi la réception asynchrone de messages. Ces services utilisent Java Message Service (JMS) pour leur fonctionnement. La nouvelle version de Java EE offre la possibilité d'empaqueter les EJB directement dans des archives WAR (*Web ARchive*), l'utilisation du Design Pattern Singleton dans les session beans et la possibilité d'utiliser des EJB légers avec un serveur Java ne proposant pas le conteneur EJB.

3.6. Les Framework JEE [18]

Un Framework est un ensemble de composants permettant de réaliser ou de créer l'architecture et les grandes lignes d'une application. Il se présente sous diverse forme qui peut inclure tout une partie des éléments suivants :

-) Un ensemble de classes généralement regroupées sous la forme de bibliothèques pour proposer des services plus ou moins sophistiqués
-) Un cadre de conception reposant sur les design patterns pour proposer tout ou partie d'un squelette d'application
-) Des recommandations sur la mise en œuvre et des exemples d'utilisation
-) Des normes de développement
-) Des outils facilitant la mise en œuvre

En ce qui concerne Java EE, il existe de nombreux Framework, cependant nous allons décrire que ceux que nous avons utilisés durant la réalisation de notre application.

3.6.1. Hibernate[19]

Dans les langages objet, les données étant le plus souvent stockées dans des bases de données relationnelles ainsi l'utilisation d'un Framework de mapping Objet/Relationnel est recommandé pour assurer la rapidité, l'évolutivité et la maintenabilité des développements. Hibernate, issu de la communauté Open Source, répond à ce besoin et connaît depuis quelques années un vif succès. Ce succès s'explique notamment par son architecture parfaitement adaptable à tout type de développements et le support de la majorité des bases de données du marché.

Hibernate est un Framework de persistance utilisé pour gérer la persistance des objets java dans une base de données et plus précisément c'est une surcouche de JDBC qui lui ajoute une dimension objet. Le terme mapping objet/relationnel (ORM) décrit la technique consistant à faire le lien entre la représentation objet des données et sa représentation relationnelle, basée sur un schéma SQL.

Concrètement, Hibernate permet de lier/mapper un objet défini en Java avec une table dans une base de données, via un fichier déclaratif de mapping. Le système peut s'occuper de

¹⁸ <https://www.jmdoudoux.fr/java/dej/chap-frameworks.htm>

¹⁹ http://gardeux-vincent.eu/Documents/ProjetJEE/BVC_Hibernate_Struts2/Hibernate/index.htm

la création des tables en fonction des fichiers de configuration et mettre aussi à jour les tables si nécessaires lors d'un changement dans l'un des fichiers de configuration.

Hibernate possède plusieurs moyens pour effectuer des requêtes. Il est possible d'exprimer des requêtes en SQL, ou en HQL (Hibernate Query Language : langage propre à Hibernate) ou encore en critères orientés objet. On notera que pour les utilisateurs d'Eclipse, il existe un plugin : Hibernate Synchroniser, qui permet de faciliter l'utilisation d'Hibernate.

a) Avantages

-) Base de données indépendante : on n'a pas besoin d'écrire un code spécifique pour la base de données (si on ne change la bd, on ne sera pas obligé à modifier dans le code l'application "requêtes SQL").
-) On peut mettre à jour la base de données sans écrire du code SQL ou HQL.
-) Le Framework Hibernate est open source.

b) Inconvénients

Il est un peu difficile de faire des requêtes complexes avec HQL.

3.6.2. JavaServer Faces (JSF)^[20]

JSF, alias Java Server Faces est un framework de type MVC, destiné aux applications web respectant l'architecture J2EE. Son développement a tenu compte des différentes expériences acquises lors de l'utilisation des technologies standard pour le développement d'applications web (servlet, JSP, JSTL) et de différents frameworks L'objectif de JSF est de fournir un environnement de développement pour la construction d'interfaces de type web sans avoir à toucher au JavaScript ou au HTML. Depuis la version 2 de JSF, les fonctionnalités ont été enrichies grâce à l'ajout de traitement des requêtes de type AJAX et l'utilisation d'autres moteurs de templating. JSF utilise la notion de Managed-bean et les expressions EL pour afficher de la donnée sur les pages web.

En résumé JSF se compose de :

²⁰ : <https://www.supinfo.com/articles/single/6402-primefaces-java-introduction>

-) Un ensemble d'APIs pour la représentation et la gestion des composants, de leur état, des évènements, de la validation des entrées et la conversion des sorties, l'internationalisation et l'accessibilité ainsi que la navigation inter-vues ;
-) Deux jeux de composants standards (affichage de texte, saisie de texte, tables, zones à cocher, etc.) : html et core ;
-) Deux bibliothèques de balises JSP (une pour chaque jeu de composants) pour permettre l'utilisation des JSPs pour la construction de vues JSF ;
-) Un modèle évènementiel côté serveur ;
-) Les Managed-Beans : qui forment la couche contrôle de JSF ;
-) Unified Expression Language : abrégé en EL ou langage d'expressions unifié pour JSF et JSP 2.0. Il permet de lier les composants aux managed-beans.

a) Avantages

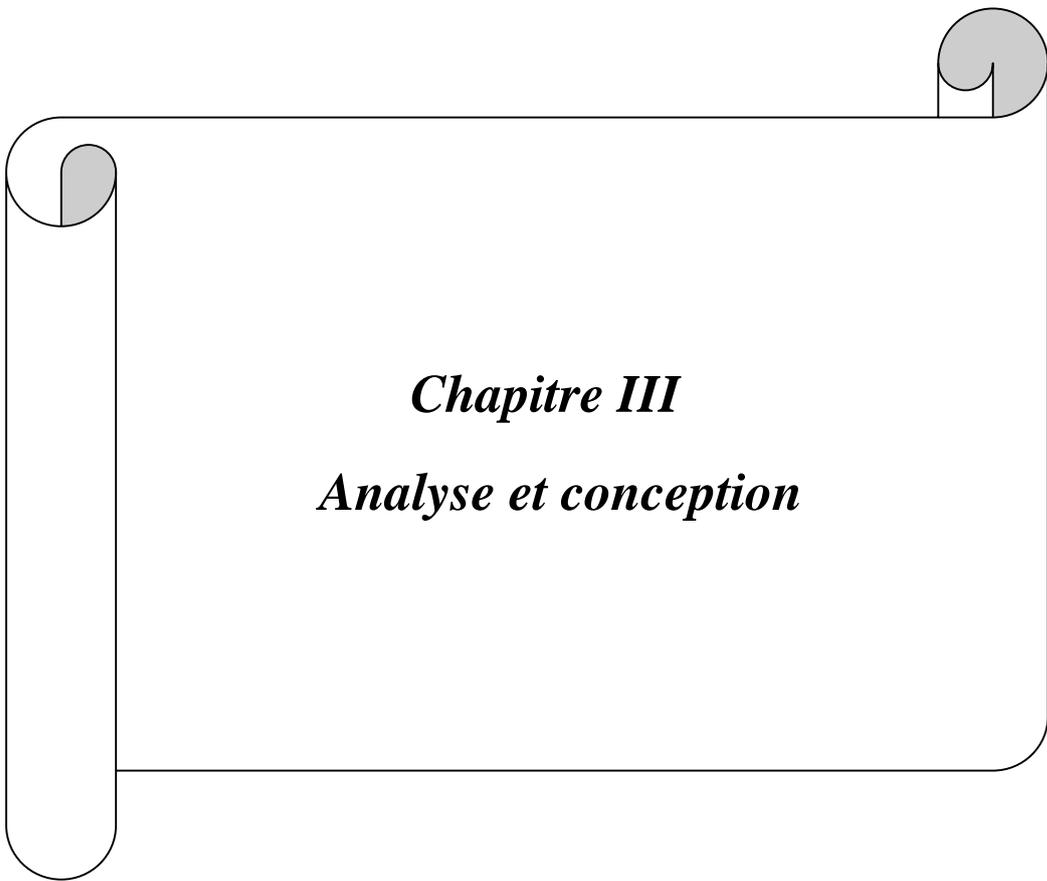
-) Possibilité de créer de nouveaux composants.
-) Accès aux Beans par leurs noms en utilisant les Expressions Language.
-) Simplification des définitions des contrôleurs et des Beans.
-) Simplification du fichier de configuration.
-) L'orientation composants graphiques permet à JSF d'être utilisé de manière plus simple dans les environnements de développement.

b) Inconvénients

-) Pas de moyen facile de commenter le fragment de la page incorrecte.
-) Ne fonctionne pas bien avec REST.
-) Pas si facile pour les concepteurs UX, car les composants prêts à utiliser ont leurs propres styles CSS, qui doivent être écrasés.

3.7. Conclusion

Dans ce chapitre nous avons vu les fondamentaux des applications web ou nous avons parlé sur le fonctionnement, l'architecture et les avantages de ces applications, de même, nous avons présenté la technologie Java EE avec son architecture et ses différents composants ainsi que les Framework que nous avons utilisés.



Chapitre III
Analyse et conception

1. Introduction

Avant toute réalisation d'une application informatique, il convient de suivre une démarche méthodologique et rigoureuse pour planifier et concevoir l'application, en mettant en évidence tous les objectifs tracés pour la bonne élaboration du projet souhaité.

Dans ce chapitre, nous ferons une analyse détaillée qui mettra en évidence les différents acteurs intervenant dans l'application. Ensuite, en s'appuyant sur les résultats de la phase analyse, la phase conception donnera la modélisation des objectifs à atteindre.

Pour ce faire, on va s'appuyer sur le langage de modélisation UML (Unified Modeling Language).

2. Présentation de l'UML

2.1 Historique [1]

Au début des années 90, une cinquantaine de méthodes d'analyse et de conception objet existaient. Parmi elles trois étaient considérées comme les plus importantes : BOOCH et Grady Booch, OMT de James Rumbaugh et enfin OOSE d'Ivar Jacobson. Ces trois acteurs ont ensuite décidé d'unir leurs efforts au sein de la société Rational Software et en 1996 la version 0.9 d'Unified Modeling Language (UML) est proposée. Deux éléments importants sont à noter :

-) Le terme unified signifie que les auteurs ont essayé de regrouper les éléments importants des concepts objets ;
-) Le terme langage montre qu'il s'agit d'un langage de modélisation et nom d'une méthode.

Les principaux acteurs du secteur informatique ont ensuite participé à cet effort, et UML 1.0 a été proposé à l'Object Management Group (OMG) ; Cet organisme international chargé de définir des standards dans le domaine de l'objet normalise UML 1.1 en 1997. Cette norme a continué d'évoluer et nous sommes aujourd'hui à la norme 2.2 sortie en 02/2009. UML est un langage qui permet de modéliser non seulement des applications informatiques ou des

¹ [https://fr.wikipedia.org/wiki/UML_\(informatique\)](https://fr.wikipedia.org/wiki/UML_(informatique))

structures de données, mais également les activités d'un domaine : mécanique, biologie et processus de métier

2.2. Les diagrammes d'UML^[2]

UML, se définit comme un langage de modélisation graphique et textuel destiné à comprendre et à définir des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue.

UML véhicule en particulier les concepts des approches par objets (classe, interface, classification, etc.) intégrant d'autres aspects (associations, fonctionnalités, événements, états, séquences, etc.).

Le langage UML propose treize diagrammes répartis sur les trois axes du niveau conceptuel :

1/ Diagrammes statiques (structurels) : dans lesquels on retrouve les diagrammes suivants :

- ✓ Diagramme de classes,
- ✓ Diagramme de package,
- ✓ Diagramme d'objet,
- ✓ Diagramme de composant,
- ✓ Diagramme de déploiement,
- ✓ Des diagrammes de structure composite.

2/ Diagrammes dynamiques (comportementaux) : dans lesquels on retrouve :

- ✓ Diagramme d'activité,
- ✓ Diagramme de séquence,
- ✓ Diagramme d'état-transition,
- ✓ Diagramme de communication,
- ✓ Diagramme de timing.

3/ Diagrammes fonctionnels : dans lesquels on retrouve les diagrammes suivants :

- ✓ Diagramme de cas d'utilisation
- ✓ Diagramme de collaboration.

² [https://fr.wikipedia.org/wiki/UML_\(informatique\)](https://fr.wikipedia.org/wiki/UML_(informatique))

3. Démarche de modélisation

La modélisation consiste à décrire dans un langage graphique formel (modèle), des besoins et les solutions fonctionnelles et techniques qui leur correspondent.

Grâce au modèle, il est possible de représenter simplement un problème, un concept et le simuler [3].

La démarche d'élaboration de notre projet se base principalement sur la modélisation UML.

Pour réaliser cette perspective nous avons adopté la démarche suivante :

-) A partir de la définition des besoins, nous allons identifier les acteurs et les activités, desquels nous déduirons assez facilement les cas d'utilisation.
-) Ceux-ci nous permettent d'établir un ensemble de scénarios d'utilisation standard à l'aide des diagrammes de séquences.
-) Ces diagrammes nous aideront à identifier les classes métier nécessaire, nous permettant d'aboutir au diagramme de classes.

La figure III.1 est une représentation graphique de cette démarche de modélisation utilisée.

³ [https://fr.wikipedia.org/wiki/UML_\(informatique\)](https://fr.wikipedia.org/wiki/UML_(informatique))

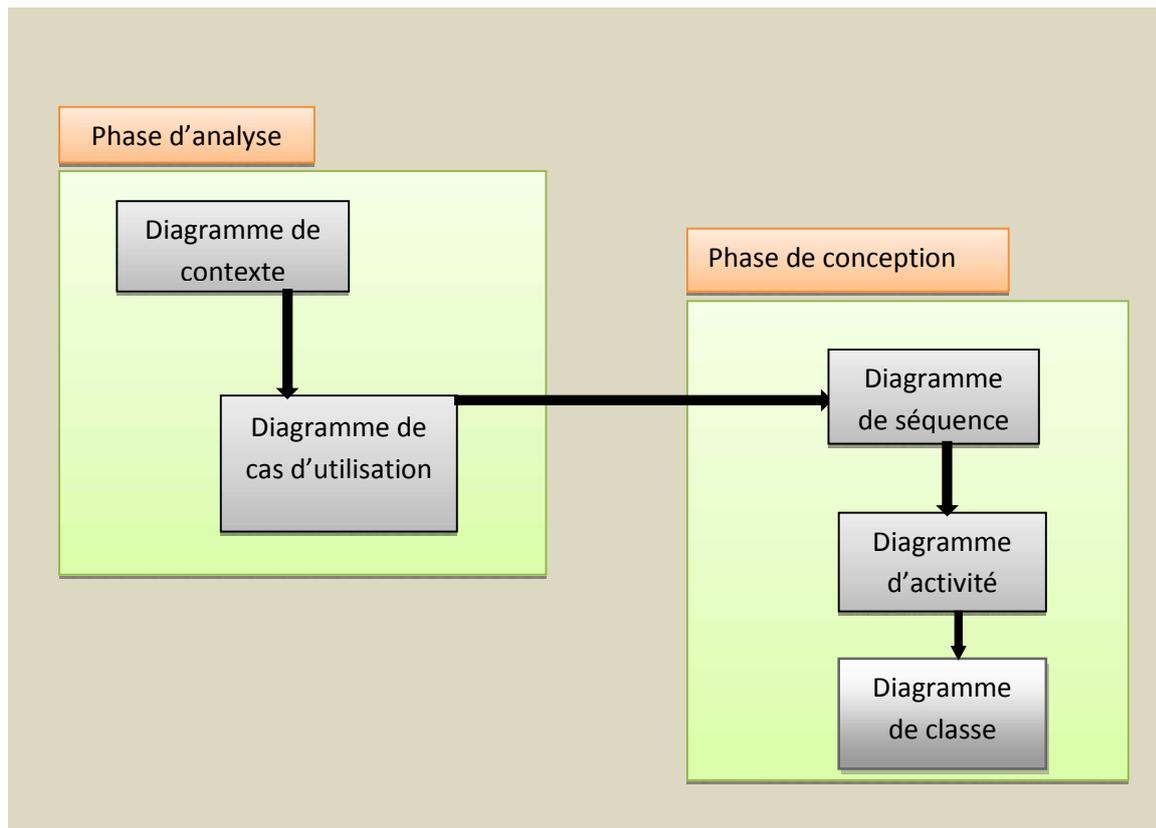


Figure III.1: Démarche de modélisation adoptée.

3.1. Analyse des besoins

Elle offre une description complète, cohérente, lisible et révisable du problème à partir de laquelle on pourra commencer à concevoir une solution [4].

3.2. Spécification des besoins

Comme les bonnes questions représentent la moitié de la réponse dans la plupart des domaines, en informatique une bonne spécification des besoins est primordiale. En effet, elle représente le travail le plus délicat et le plus significatif.

Dans cette section nous allons présenter notre réponse à cette question. Donc, cette phase consiste à mieux comprendre le contexte du système, il s'agit de déterminer les fonctionnalités et les acteurs et d'identifier les cas d'utilisation initiaux [5].

⁴ [https://fr.wikipedia.org/wiki/UML_\(informatique\)](https://fr.wikipedia.org/wiki/UML_(informatique))

3.3. Identification des besoins

A/Les besoins fonctionnels :

Les besoins fonctionnels représentent les principales fonctionnalités du système. Ces besoins proviennent généralement des utilisateurs du système. Cette application devra permettre le bon fonctionnement de la procédure achats qui se résume ainsi :

-) Etablir une commande.
-) Signer et valider les demande d'achats
-) Etablir les bon de commande externe.
-) Paiement des fournisseurs.
-) Consulter l'historique des achats.

B/Les besoins non fonctionnels :

En plus des besoins fondamentaux, notre futur système doit répondre aux critères suivants :

-) La performance : Un logiciel doit être avant tout performant c'est-à-dire à travers ses fonctionnalités, il doit répondre à toutes les exigences des usagers d'une manière optimale.
-) La convivialité : Le futur logiciel doit être facile à utiliser. En effet, les interfaces utilisateurs doivent être conviviales c'est-à-dire simples ergonomiques et adaptées à l'utilisateur.
-) La confidentialité : vu que les données manipulées par notre application sont critiques, nous devons garantir une sécurité optimale. Ainsi, les droits d'accès au système doivent être bien attribués, afin d'assurer la sécurité des données.

4. Acteurs et leurs tâches

Un acteur représente un rôle que peut jouer l'utilisateur ou toute entité externe pouvant interagir avec le système. Autrement dit, c'est un objet actif qui utilise les fonctions du Système [6].

⁵ https://www.memoireonline.com/07/08/1363/m_conception-realisation-application-gestion-centre-kinesie9.html

⁶ https://www.memoireonline.com/07/08/1363/m_conception-realisation-application-gestion-centre-kinesie9.html

Une tâche est l'ensemble des différentes fonctions aux quelles un acteur peut accéder.

Ces tâches sont résumées dans le tableau ci-dessous :

Acteurs	Taches
Service achats	<ul style="list-style-type: none"> ❖ Etablissement de factures proforma. ❖ Contacte les Fournisseurs. ❖ Etablissement de bon de commande externe. ❖ Négociation des délais de paiement et de livraison. ❖ Etablissement de la demande de paiement.
Service comptabilité	<ul style="list-style-type: none"> ❖ Paiement des Fournisseurs. ❖ Réception des demandes de paiements.
La direction générale	<ul style="list-style-type: none"> ❖ Validation des demandes d'achats. ❖ Modification des demandes d'achats. ❖ Signature des demandes de paiement. ❖ Gestion opérationnelle.
Services demandeurs	<ul style="list-style-type: none"> ❖ Etablissement de Formulaire de demande d'achats. ❖ Identification des besoins. ❖ Réception de la livraison.

5. Diagramme de contexte

Le diagramme de contexte permet la représentation des interactions entre objets selon un point de vue temporel, [7]. Une fois la spécification des besoins des acteurs est faite, le diagramme de contexte est élaboré :

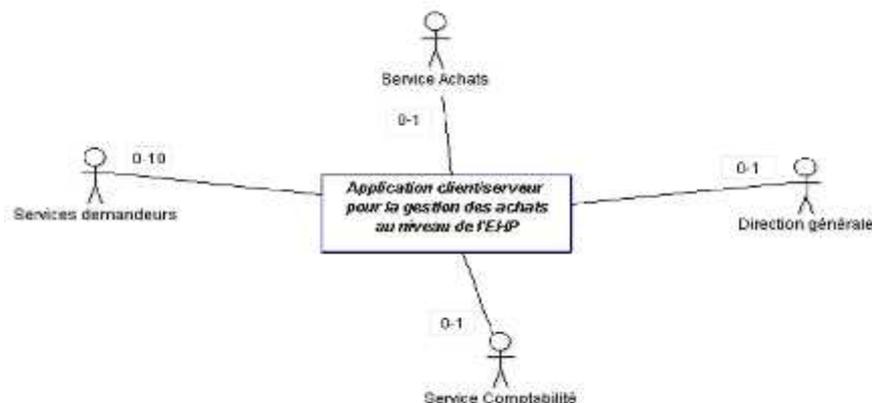


Figure III.2: Diagramme de contexte.

⁷ https://www.memoireonline.com/06/11/4548/m_Conception-et-implementation-dun-systeme-de-gestion-des-centres-dacces-a-internet--rechargemen10.html

6. Diagramme de cas d'utilisation

Bien souvent, les utilisateurs ne sont pas des informaticiens. Il leur faut donc un moyen simple d'exprimer leurs besoins. C'est précisément le rôle des diagrammes de cas d'utilisation qui permette de recueillir, d'analyser et d'organiser les besoins, et de recenser les grandes fonctionnalités d'un système. Il s'agit donc de la première étape UML d'analyse d'un système.

Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous-système, d'une classe ou d'un composant tel qu'un utilisateur le voit. Il scinde la fonctionnalité du système en unités cohérentes, les cas d'utilisation, ayant un sens pour les acteurs.

Il ne faut pas négliger cette première étape pour produire un logiciel conforme aux attentes des utilisateurs, [8].

6.1. Identification des cas d'utilisation

Un cas d'utilisation permet de déterminer les actions et les opérations que chaque acteur est capable de faire. Autrement dit, les cas d'utilisation servent à décrire le comportement du système du point de vue de son environnement. Ils sont de manière spécifique d'utiliser le système.

6.2. Relations entre cas d'utilisation

En plus de la relation de communication qui provoque le déclenchement d'un cas d'utilisation, nous pouvons citer deux types de relations qui sont les plus utilisés : la relation d'inclusion et celle d'extension :

- ❖ Relation d'inclusion (include) : Une relation d'inclusion entre des cas d'utilisation signifie que le cas d'utilisation principal incorpore le comportement d'un autre cas d'utilisation en un point d'intersection bien déterminé. Le cas d'utilisation inclus n'est jamais tout seul, il fait toujours partie d'un cas d'utilisation qui l'englobe.
- ❖ Relation d'extension (extend) : une relation d'extension du cas d'utilisation A à un cas d'utilisation B signifie qu'une instance de A peut être étendue par le comportement décrit dans le cas d'utilisation B. [9]

⁸ https://fr.wikipedia.org/wiki/Diagramme_des_cas_d%27utilisation

Notre application comporte trois cas d'utilisation qui se résument en :

- ✓ Gestion des achats.
- ✓ Gestion de la comptabilité.
- ✓ Gestion de la sécurité de l'application.

6.3. Diagramme de cas d'utilisation global

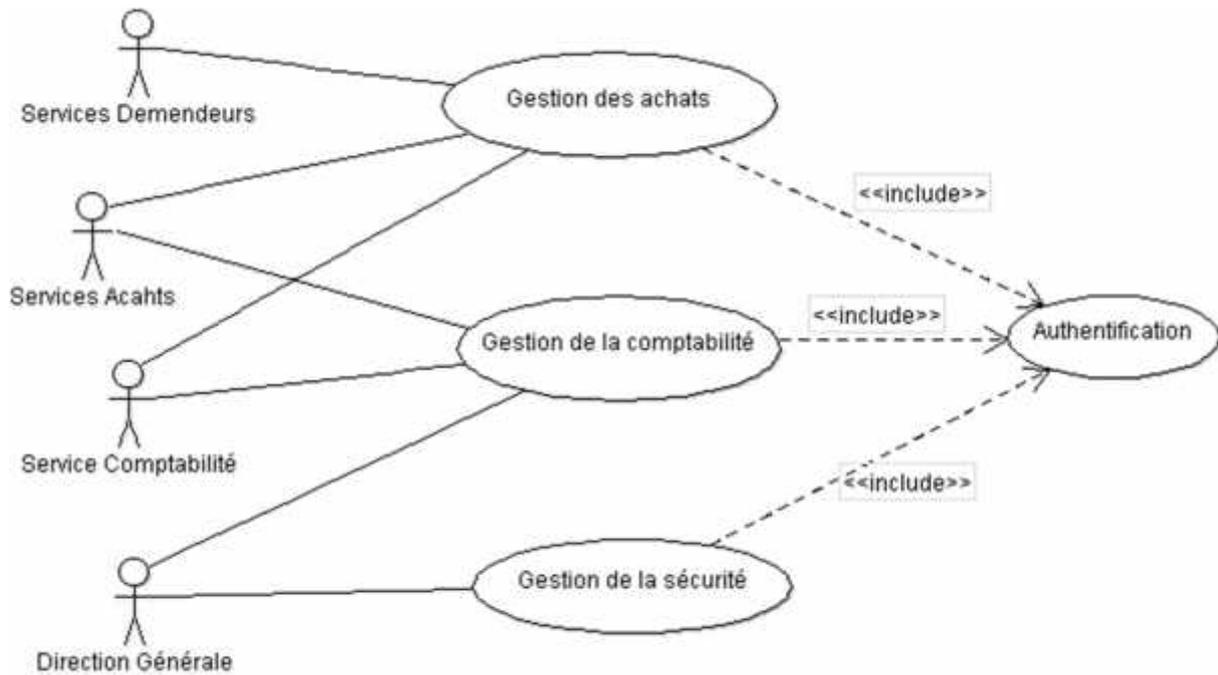


Figure III.3 : Diagramme de cas d'utilisation global.

⁹ https://fr.wikipedia.org/wiki/Diagramme_des_cas_d%27utilisation

6.4. Diagramme de cas d'utilisation détaillé

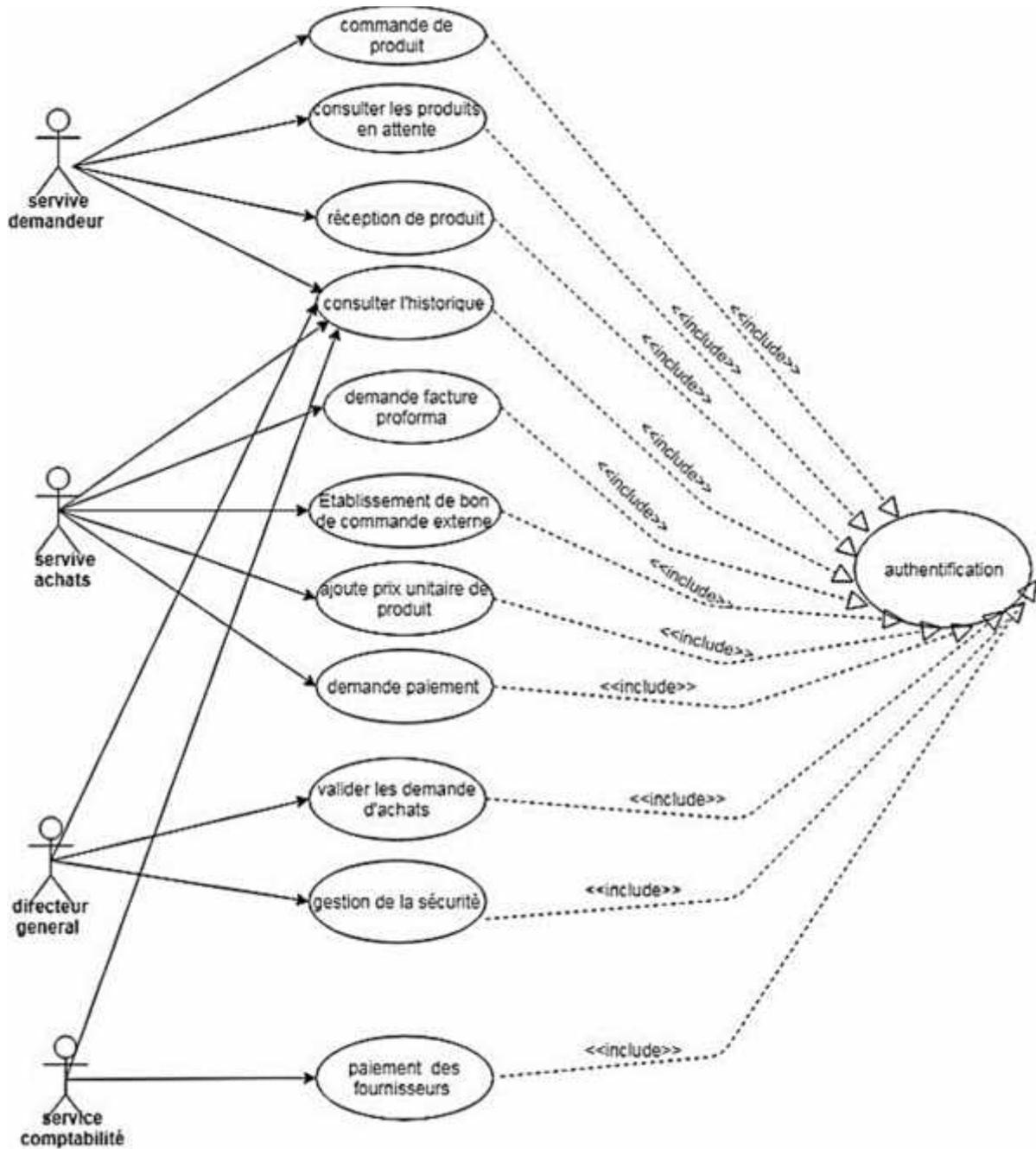


Figure III.4 : Diagramme de cas d'utilisation détaillé.

Les différents scénarios sont expliqués si dessous :

1- Gestion des achats

Gestion des achats
Cas d'utilisation "commande produits"
Acteurs : Utilisateur (Service demandeur). Pré condition : L'utilisateur est connecté. Post condition : demande d'achats.
Scénario nominale : 1. L'utilisateur demande le formulaire de commande produit. 2. Le système affiche le formulaire de commande produit. 3. L'utilisateur remplit le formulaire. 4. L'utilisateur valide le formulaire. 5. Le système enregistre la commande.
Scénario alternatifs : A : Les champs du formulaire ne sont pas remplis L'enchaînement A démarre au point 3 du scénario nominal.
Cas d'utilisation " création de demande de facture proforma"
Acteurs : Utilisateur (Service achats). Pré condition : L'utilisateur est connecté. Post condition : demande de facture proforma.
Scénario nominale : 1. L'utilisateur demande le formulaire de création de facture proforma. 2. Le système affiche le formulaire. 3. L'utilisateur sélectionne un élément afin d'ajouter son prix et le contacte de fournisseur. 4. L'utilisateur remplit le formulaire. 5. L'utilisateur valide le formulaire. 6. Le système enregistre la facture proforma.
Scénario alternatifs : A : Aucun élément sélectionner. L'enchaînement A démarre au point 3 du scénario nominal. B : Les champs du formulaire ne sont pas remplis. L'enchaînement B démarre au point 4 du scénario nominal.
Cas d'utilisation "validation des commandes produits"
Acteurs : Utilisateur (direction générale). Pré condition : L'utilisateur est connecté. Post condition : facture proforma validée.
Scénario nominale : 1. Le système affiche les demandes d'achats non valider. 2. L'utilisateur sélectionne la demande a validé. 3. L'utilisateur valide la demande. 4. Le système enregistre les demandes validées.
Scénario alternatifs : A : Aucun élément sélectionner. L'enchaînement A démarre au point 2 du scénario nominal.

Cas d'utilisation "accusé de réception des produits"
Acteurs : Utilisateur (Service demandeur). Pré condition : L'utilisateur est connecté. Post condition : Bon de réception.
Scénario nominale : 1. Le système affiche les produits non réceptionner. 2. L'utilisateur sélectionne le produit qui viens d'être réceptionner. 3. L'utilisateur valide la réception. 4. Le système enregistre l'accusé de réception.
Scénario alternatifs : A : Aucun élément sélectionner. L'enchaînement A démarre au point 2 du scénario nominal.
Cas d'utilisation "demande de paiement"
Acteurs : Utilisateur (Service achats). Pré condition : L'utilisateur est connecté. Post condition : Demande de paiement.
Scénario nominale : 1. Le système affiche les produits réceptionner non payer. 2. L'utilisateur sélectionne un produit. 3. L'utilisateur remplit le formulaire de demande de paiement. 4. L'utilisateur valide le formulaire. 5. Le système enregistre l'accusé de réception.
Scénario alternatifs : A : Aucun élément sélectionner. L'enchaînement A démarre au point 2 du scénario nominal. B : Les champs du formulaire ne sont pas remplis. L'enchaînement B démarre au point 4 du scénario nominal.
Cas d'utilisation "paiement"
Acteurs : Utilisateur (Service Comptabilité). Pré condition : L'utilisateur est connecté. Post condition : Bon de paiement.
Scénario nominale : 1. Le système affiche les produits non payer. 2. L'utilisateur sélectionne un produit. 3. L'utilisateur valide le paiement. 4. Le système enregistre le paiement.
Scénario alternatifs : A : Aucun produit sélectionner. L'enchaînement A démarre au point 2 du scénario nominal.

2- Gestion de sécurité de l'application

Gestion de sécurité de l'application:

Acteurs : Direction générale

Pré conditions : utilisateur avec droits d'accès initiaux.

Post conditions : utilisateur avec droits d'accès modifiés.

Scénario nominal :

1. La direction générale choisi l'utilisateur à partir de la liste.
2. La direction générale modifie les droits d'accès de l'utilisateur de l'application.
3. Le système enregistre la modification.

Scénario d'exception :

E1 : si la direction générale ne respecte pas la sécurité des droits d'accès, le système va afficher un message d'erreur.

L'enchaînement reprend à l'étape 2 du scénario nominale.

3- Cas d'utilisation « s'authentifier »

Cas d'utilisation « s'authentifier »

Titre : s'authentifier

Résumé : chaque utilisateur doit taper son propre nom d'utilisateur et son mot de passe pour accéder à l'interface qui le concerne.

Acteurs : Direction générale, service Achats, service comptabilité, le reste des services.

Pré conditions : l'utilisateur n'est pas authentifié donc il ne peut pas accéder à l'application.

Post conditions : l'utilisateur est authentifié, il accède à l'espace qui le concerne.

Scénario nominal :

1. L'utilisateur choisi le nom d'utilisateur et tape son mot de passe.
2. Le système vérifie la validité des coordonnées de l'utilisateur.
3. L'utilisateur accède à la session qui le concerne.

Scénario d'exception :

E1 : si l'identifiant de l'utilisateur n'existe pas dans la base de données, le système va afficher un message d'erreur.

L'enchaînement reprend à l'étape 1 du scénario nominale.

7. Diagramme de séquence

Le diagramme de séquence suit le diagramme de cas d'utilisation car il le complète. Il permet de décrire les scénarios (déroulement des traitements entre les éléments du système et les acteurs) de chaque cas d'utilisation en mettant l'accent sur la chronologie des opérations en interaction avec les objets.

En particulier, il montre aussi les objets qui participent à l'interaction par leur "ligne de vie" et les messages qu'ils échangent présentés en séquence dans le temps.

Quelques notions de base du diagramme de séquence :

Scénario : une liste d'actions qui décrivent une interaction entre un acteur et le système.

Interaction : un comportement qui comprend un ensemble de messages échangés par un ensemble d'objets dans un certain contexte pour accomplir une certaine tâche.

Message : un message représente une communication unidirectionnelle entre objets qui transporte l'information avec l'intention de déclencher une réaction chez le récepteur.

7.1. Diagramme de séquence "cas authentification"

L'authentification est une étape fondamentale dans l'application. Pour accéder au système, l'utilisateur introduit son nom d'utilisateur (login) et mot de passe. Si les informations sont correctes une page d'accueil s'affiche selon l'utilisateur, si non le système renvoie un message d'erreur.

Ci-dessous le diagramme de séquence pour le cas d'utilisation « authentification » :

7.2. Diagramme de séquence "cas commande produit"

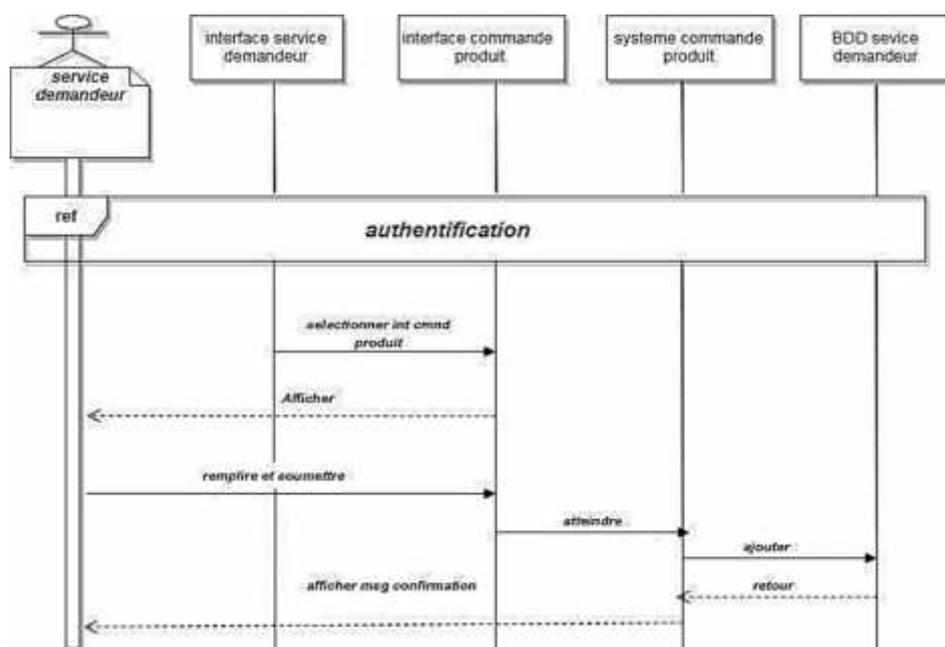


Figure III.5 : Diagramme de séquence "cas commande produit".

7.3. Diagramme de séquence "cas établissement de demande d'achats"

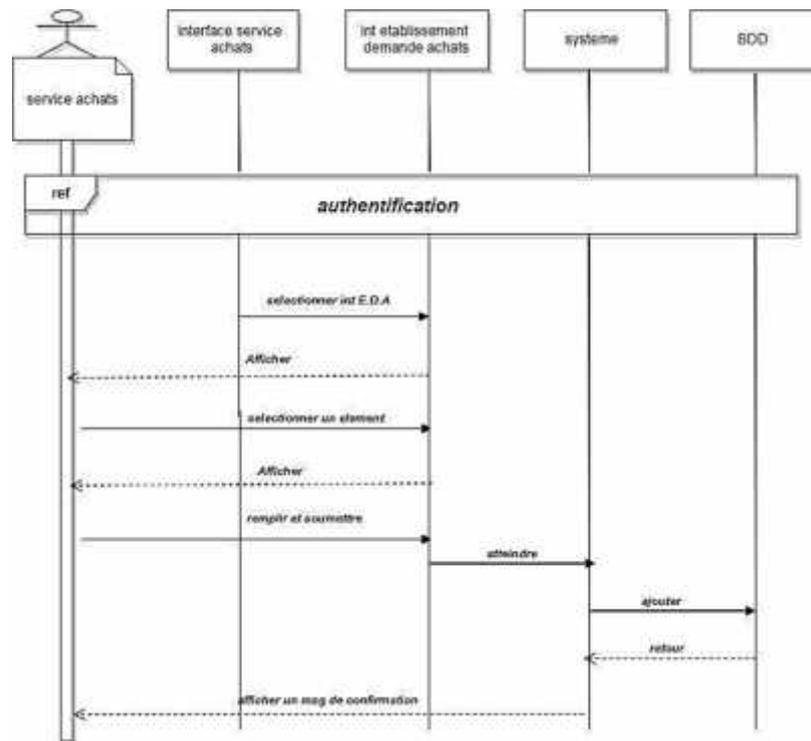


Figure III.6 : Diagramme de séquence "cas établissement de demande d'achats ".

7.4. Diagramme de séquence "cas établissement de bon de commande externe"

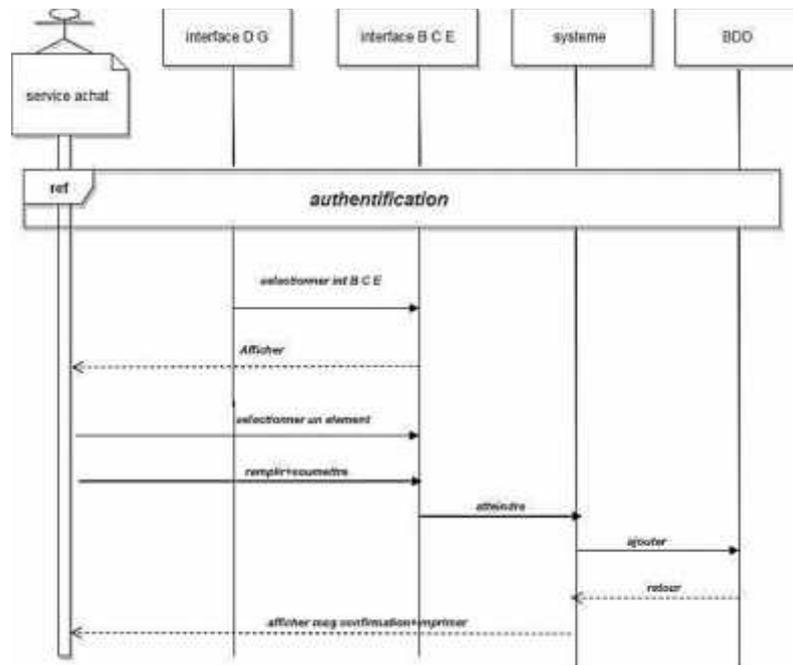


Figure III.7 : Diagramme de séquence "cas établissement de bon de commande externe ".

7.5. Diagramme de séquence "cas valider la demande d'achats"

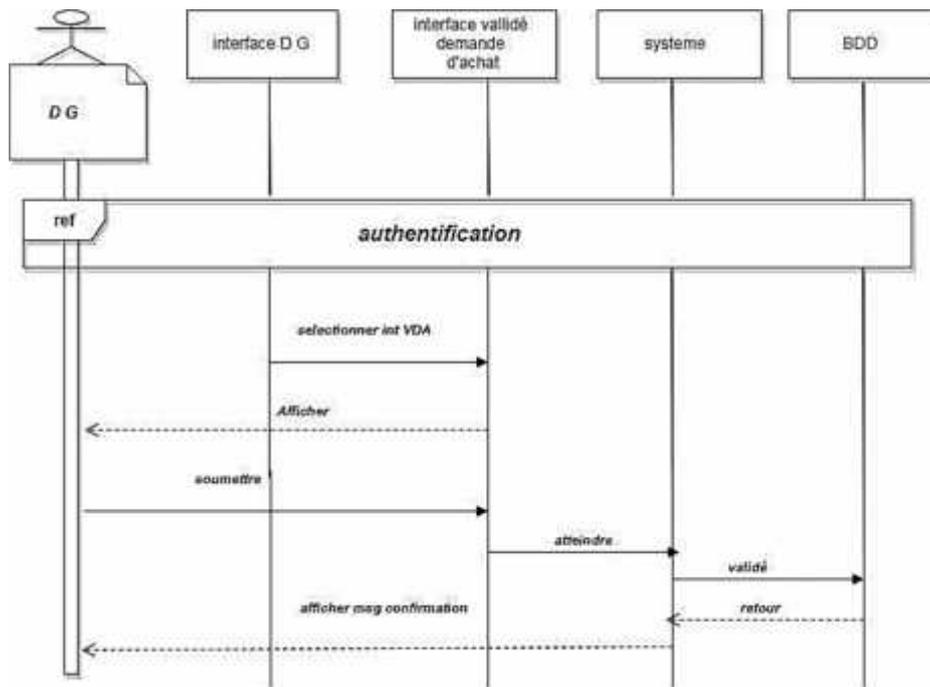


Figure III.8 : Diagramme de séquence "cas valider la demande d'achats ".

7.6. Diagramme de séquence "cas accusé de réception"

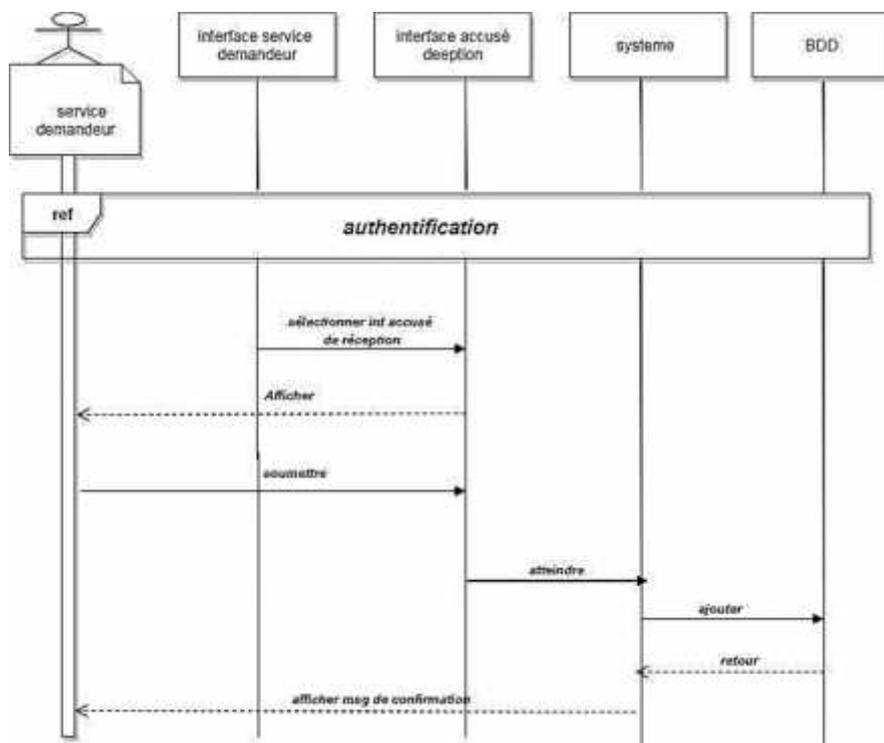


Figure III.9 : Diagramme de séquence "cas accusé de réception ".

7.7. Diagramme de séquence "cas demande de paiement "

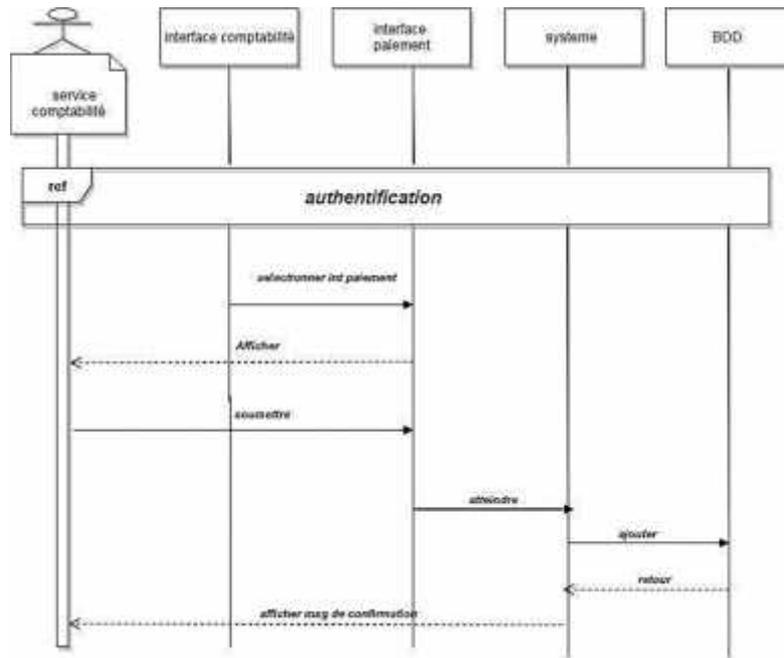


Figure III.10 : Diagramme de séquence "cas demande de paiement ".

7.8. Diagramme de séquence "cas de paiement "

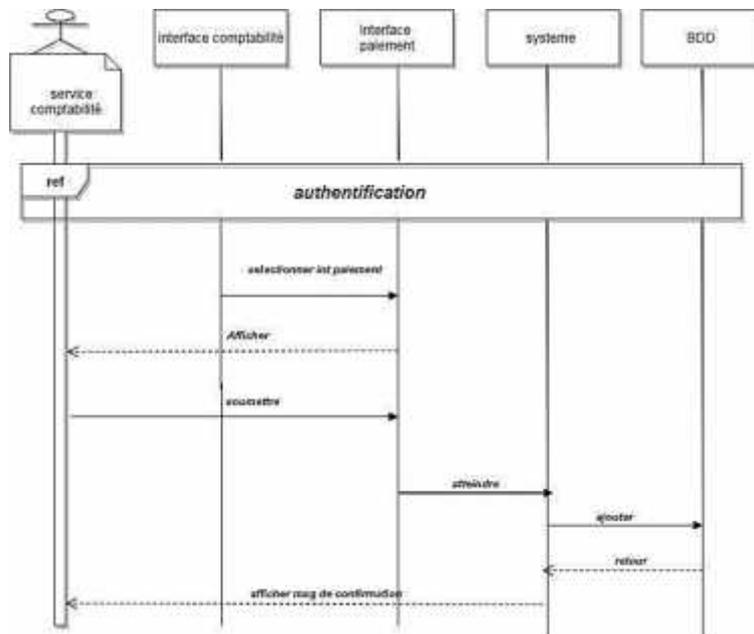


Figure III.11 : Diagramme de séquence "cas de paiement ".

8. Diagramme d'activité

Il donne une vision des enchainements des activités propres à une opération ou à un cas d'utilisation. Un diagramme d'activités permet de mettre l'accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Il indique la part prise par chaque objet dans l'exécution d'un travail. Il sera enrichi par les conditions de séquence. Dans la phase de conception, les diagrammes d'activités sont particulièrement adaptés à la description des cas d'utilisation en venant, par exemple, illustrer et consolider leur description textuelle. Ainsi leur représentation est sous forme d'organigramme, ce qui les rend utiles dans la phase de réalisation.

8.1. Diagramme d'activité "Authentification"

Ci-dessous le diagramme d'activité pour le cas d'authentification :

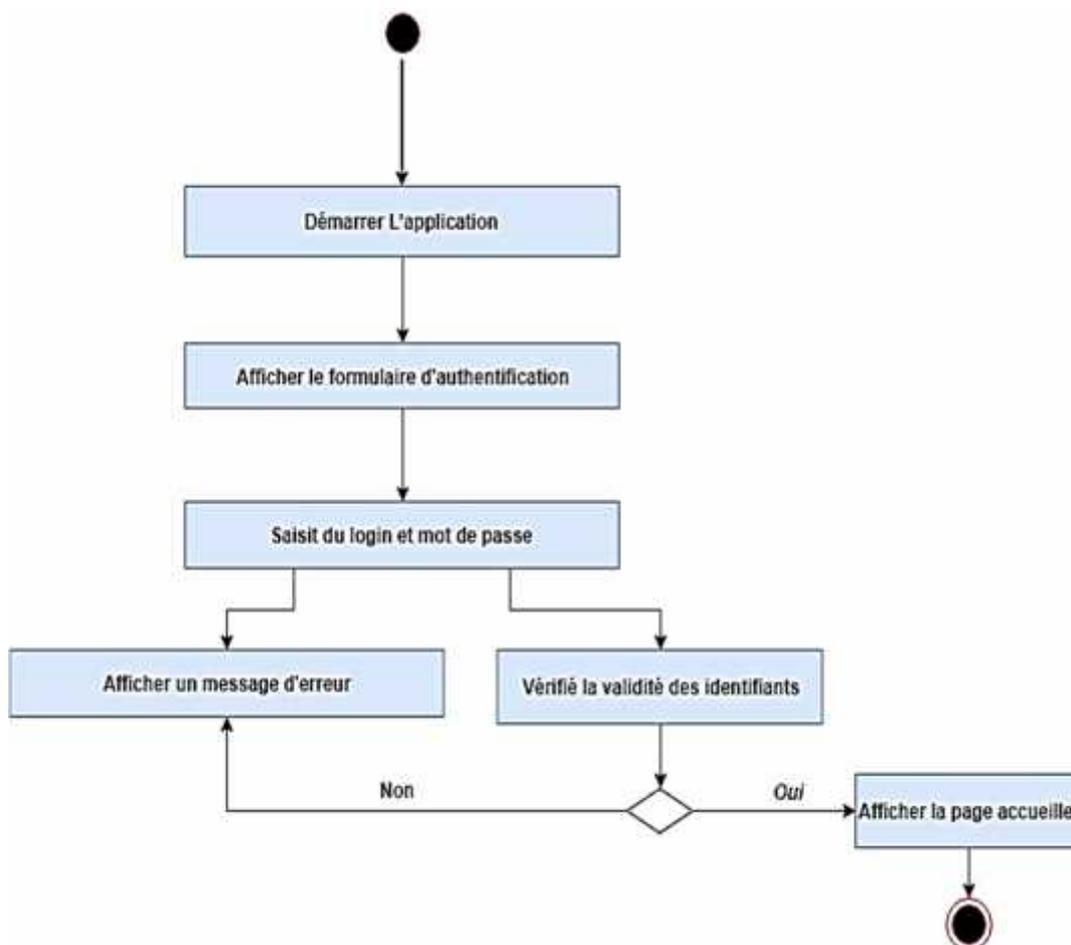


Figure III.12 : Diagramme d'activité « authentification ».

8.2. Diagramme d'activité "Ajouter une commande"



Figure III.13 : Diagramme d'activité « Ajouter une commande ».

8.3. Diagramme d'activité "Modifier une commande"



Figure III.14 : Diagramme d'activité « Modifier une commande ».

8.4. Diagramme d'activité "Supprimer une commande"

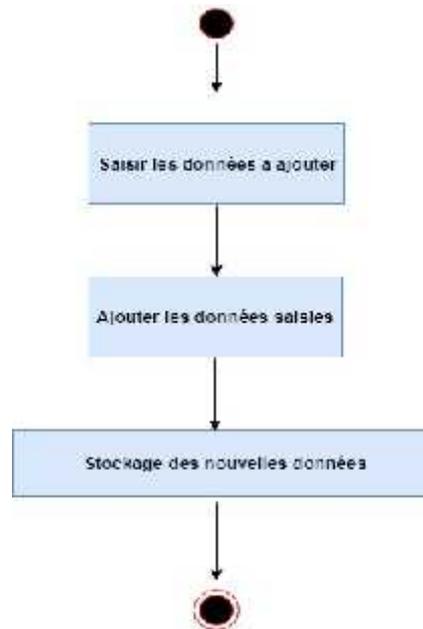


Figure III.15 : Diagramme d'activité « Supprimer une commande ».

8.5. Diagramme d'activité "Etablissement d'une demande d 'achats"

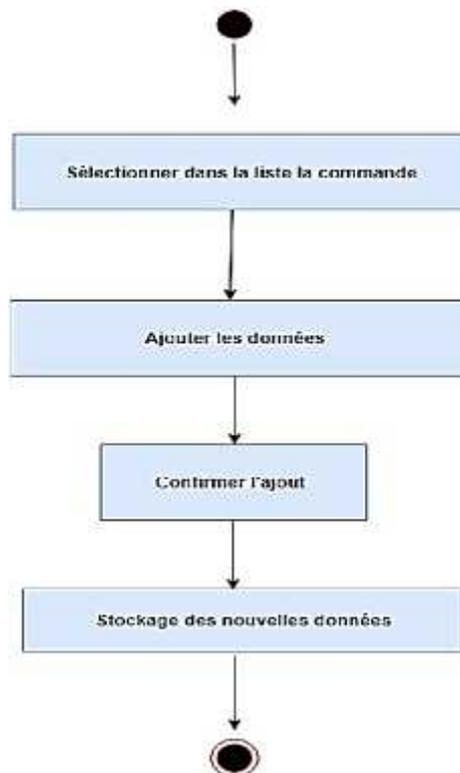


Figure III.16 : Diagramme d'activité « Etablissement d'une demande d 'achats ».

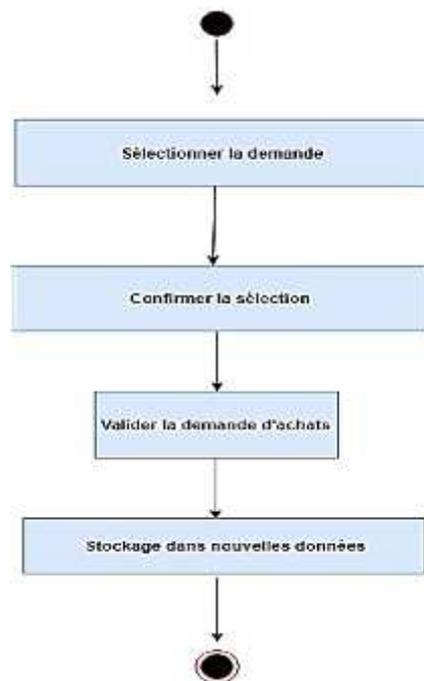
8.6. Diagramme d'activité "Valider une demande d'achats"

Figure III.17 : Diagramme d'activité « Valider une demande d'achats ».

8.7. Diagramme d'activité "Modifier une demande d'achats"

Figure III.18 : Diagramme d'activité « Modifier une demande d'achats ».

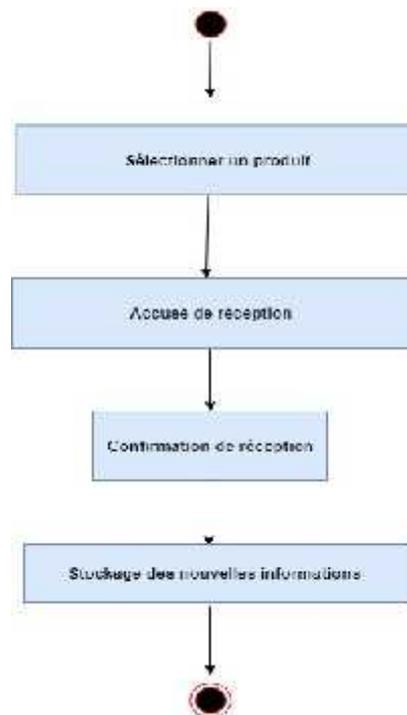
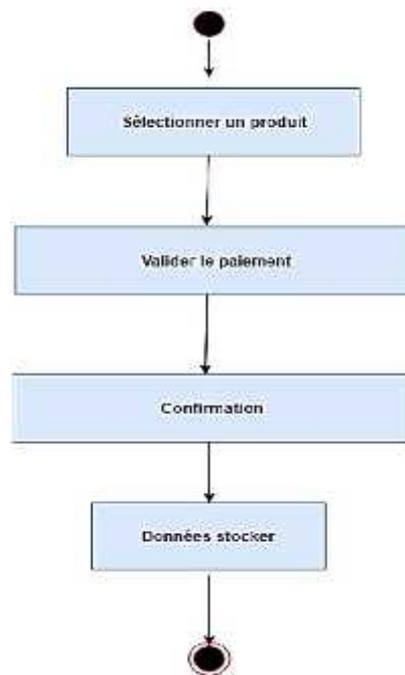
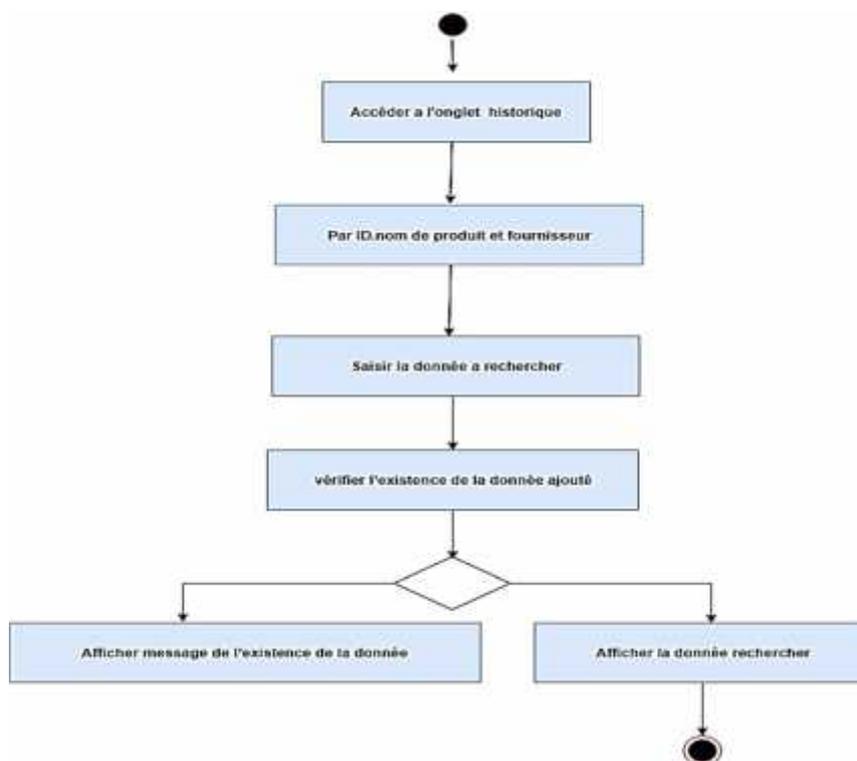
8.8. Diagramme d'activité "Accusé de réception d'un produit"

Figure III.19 : Diagramme d'activité « Accusé de réception d'un produit ».

8.9. Diagramme d'activité "Demande de paiement"

Figure III.20 : Diagramme d'activité « Demande de paiement ».

8.10. Diagramme d'activité "Paieiment"**Figure III.21 :** Diagramme d'activité « paiement ».**8.11. Diagramme d'activité "Recherche dans historique"****Figure III.22 :** Diagramme d'activité « Recherche dans historique ».

9. Diagramme de classe

Le diagramme de classe constitue un élément très important de la modélisation : il permet de définir quelles seront les composants du système final.

Elle représente les classes intervenantes dans le système. Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets, les éléments de cet ensemble sont les instances de la classe.

9.1. Identification des classes

Une classe est une description d'un groupe d'objets partageant un ensemble commun de Propriétés (les attributs), de comportements (les opérations) et de relations avec d'autres objets (les associations et les agrégations).

Une class contient :

Des attributs : (ou champs, ou variables d'instances) : les attributs d'une classe sont une caractéristique d'un objet, décrivent la structure de ses instances (les objets). Un attribut souligné correspond à un attribut de classe.

Des méthodes : (ou opérations de la classe) : les méthodes décrivent les opérations qui sont applicables aux instances de la classe. C'est un service dont un objet peut demander l'exécution.

La multiplicité : sert à compter le nombre minimum et maximum de possibilité que chaque classe contient dans la relation liant deux ou plusieurs classes.

Une agrégation : il s'agit d'une relation entre deux classes, spécifiant que les objets d'une classe sont des composants de l'autre classe. C'est donc une association qui lorsqu'elle est lue dans un sens signifie « est une partie de » et lorsqu'elle est lue dans l'autre sens elle signifie « est composé de ».

Codification	Désignation	Type	Taille	Observation
Idu	Identification Utilisateur	N	01	
Mp	Mot de passe utilisateur	AN	10	
Nom_service	Nom_service	A	25	
Tel_service	Telephone de service	N	10	
Idcommande	Identificateur de la commande	N	04	
Nom_produit	Nom de produit	N	10	
Unité_produit	Unité de produit	D	08	
Quantité_demandée	Quantité demandée	N	08	
Quantité_en_stocks	Quantité en stocks	N	08	
Date_commande	La date de commande	Date		JJ/MM/AAAA
Date_livraison_s	Date souhaitée pour la livraison	Date		JJ/MM/AAAA
Date_reception	Date de réception de produit	Date		JJ/MM/AAAA
Prix_unitaire	Prix unitaire de produit	N	08	
Tht	Prix total hors taxe	N	10	
Fournisseur	Le nom de fournisseur	AN	50	
Tel_fournisseur	Téléphone de fournisseur	N	10	
Tva	Taxe de la valeur ajoutée	N	2	
Ttc	Prix total avec taxe compris	N	08	
Date_paiement	Date de paiement	Date		JJ/MM/AAAA

Ci-dessous sont présentées les différentes classes qui composent notre application :

Utilisateur : cette classe regroupe les utilisateurs de l'application.

Service Achats : cette classe regroupe les utilisateurs l'effectif de service Achats de l'EHP les amandiers, dans notre cas d'étude leurs rôles ce résume à assurer le bon déroulement de la, procédure achats au niveau de l'établissement.

Service Comptabilité : cette classe regroupe l'effectif de service Comptabilité de l'EHP les amandiers, dans notre cas d'étude leurs rôles ce résume à la facturation et à payer les fournisseurs.

Direction Générale : cette classe regroupe le directeur général de l'EHP les amandiers, dans notre cas d'étude leurs rôles ce résume à valider les demandes d'achats.

Service demandeur : cette classe regroupe les utilisateurs qui pouvant effectués des demandes d'achats.

Fournisseur : cette classe représente le fournisseur de produit commander

Bon de commande externe : une instance de cette classe représente les bons de commande externes utilisés par le service achats pour effectuer des demandes aux fournisseurs.

Demande facture proforma : une instance de cette classe représente les produits envoyer par le service achats en direction des fournisseurs afin d'apprendre davantage sur les prix des produits.

Facture proforma : une instance de cette classe représente les produits avec leurs prix unitaires.

Espace achats : cette classe représente les informations sur le produit.

Commande : cette classe contient les informations sur la commande, il contient les produits, la quantité demandée, et le service.

Bon de réception : une instance de cette classe représente les informations concernant la réception ou la livraison des produits.

Formulaire de demande achats : une instance de cette classe représente les informations concernant l'achats.

Bon de livraison : une instance de cette classe représente les informations concernant la livraison.

Facture : une instance de cette classe représente les informations concernant le prix de produit, les taxes et le total à payer.

Demande de Paiement : une instance de cette classe représente les informations sur la demande de paiement.

Paiement : cette classe représente les informations sur le paiement.

Stock : cette classe représente les informations sur l'état des stocks.

Produit : cette classe représente les informations sur le produit.

Signature : une instance de cette classe représente la quantité accorder par la direction générale et la validation de l'achats.

Règles de gestion

- Un espace achats peut établir une ou plusieurs demandes facture proforma.
- Plusieurs demandes de facture proforma peuvent être établit par l'espace achats.
- L'espace achats peut réceptionner une ou plusieurs facture proforma.
- Une ou plusieurs demandes de facture proforma peuvent être réceptionné par un espace achats.
- L'espace achats peut établir plusieurs bon de commande externe.
- Un bon de commande externe peut être établit par un seul espace achats.
- Un ou plusieurs fournisseurs peuvent réceptionner plusieurs demandes de facture proforma.
- Plusieurs facture proforma peuvent être réceptionné par un ou plusieurs fournisseurs.
- Plusieurs factures proforma peuvent être établit par un ou plusieurs fournisseurs.
- Un ou plusieurs fournisseurs peuvent réceptionner un ou plusieurs bon de commande externe.
- Plusieurs bon de commande externes peuvent réceptionner par un ou plusieurs fournisseurs.
- Un fournisseur peut établir un ou plusieurs bon de livraison.
- Un ou plusieurs bon de livraison peuvent être établit par un seul fournisseur.
- Un seul espace achats peut réceptionner un ou plusieurs bon de livraison.
- Un ou plusieurs bon de livraison peuvent être réceptionné par un seul espace achats.
- Un espace achats peut réceptionner un ou plusieurs bon de réception.
- Un ou plusieurs bon de réception peuvent être réceptionné par un espace achats.
- Un espace achats peut réceptionner une ou plusieurs commandes.
- Une ou plusieurs commandes peuvent être réceptionné par un espace achats.
- Un espace achats peut établir une ou plusieurs demandes de paiement.
- Un ou plusieurs demandes de paiement peuvent être établit par un seul espace achats.
- La direction générale peut signer un ou plusieurs formulaires de demande d'achats.
- La direction générale peut signer un ou plusieurs demandes de paiement.
- Le service comptabilité peut réceptionner une ou plusieurs demandes de paiement.
- Un ou plusieurs demandes de paiement peut être réceptionné par un seul service comptabilité.
- Un ou plusieurs fournisseuses peuvent livrer un ou plusieurs livraisons.
- Un ou plusieurs livraisons peuvent être livré par un ou plusieurs fournisseurs.
- Un ou plusieurs fournisseuses peuvent établir une ou plusieurs factures.
- Une ou plusieurs factures peuvent être établit par un ou plusieurs fournisseurs.
- Un ou plusieurs service demandeur peuvent établir une commande.
- Une commande peut être établis par un ou plusieurs services.
- Un services demandeur peut réceptionner un ou plusieurs bon de réception.

- ✚ Plusieurs bon de réception peuvent être réceptionné par un seul service demandeur.
- ✚ Un service demandeur peut réceptionner un ou plusieurs bon de livraison.
- ✚ Un ou plusieurs bon de livraison peuvent être réception" par un seul service demandeur.
- ✚ Un stock contient un ou plusieurs produits.
- ✚ Une livraison contient un ou plusieurs produits.
- ✚ Le service comptabilité à pour paiement un ou plusieurs fournisseurs
- ✚ Un ou plusieurs paiements concerne un ou plusieurs fournisseurs.

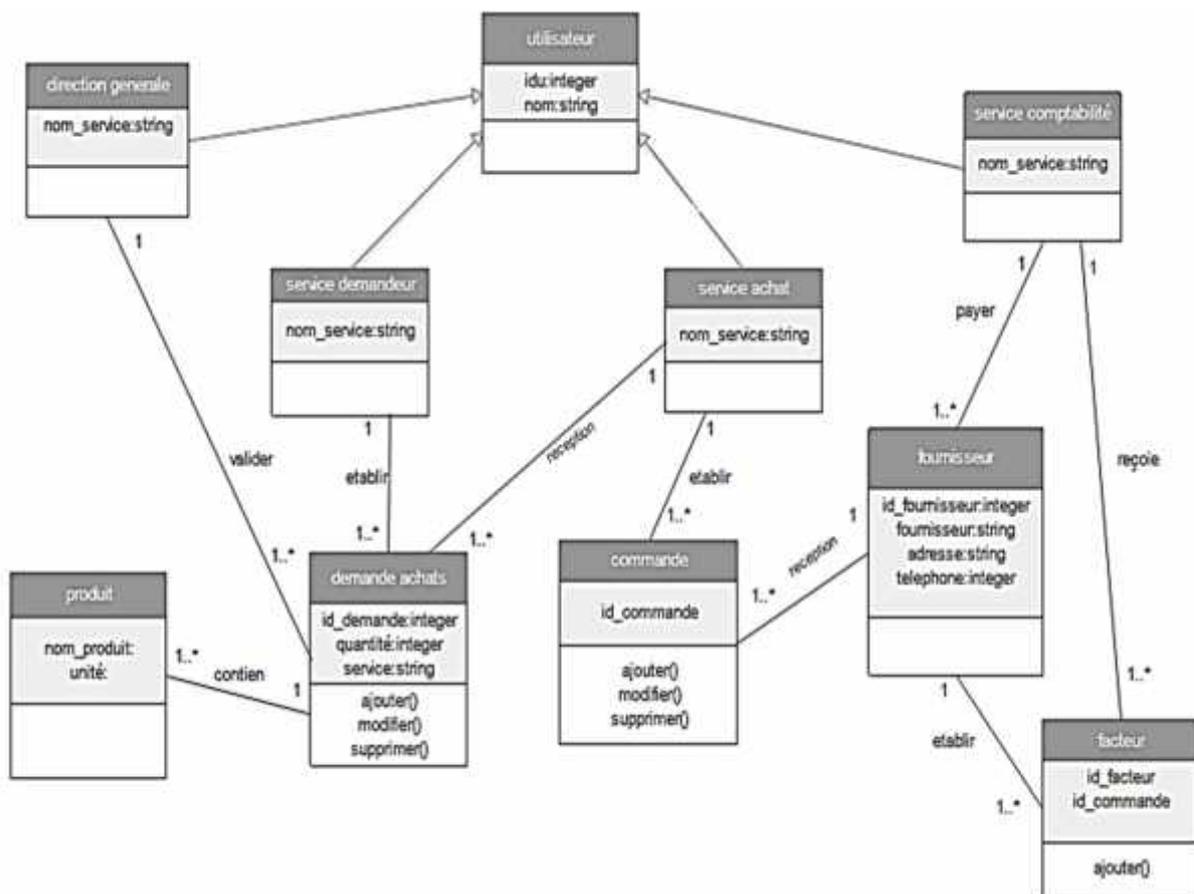


Figure III.23 : Diagramme de classe.

10. Le modèle relationnel

Le modèle relationnel est une manière de modéliser les relations existantes entre plusieurs informations, et de les ordonner entre elles. Cette modélisation qui repose sur des principes

mathématiques, est souvent retranscrite physiquement « Implémentée » dans une base de données [11].

A partir de la description conceptuelle qu'on a effectuée, on peut réaliser le modèle relationnel ; vu que le système d'information ne peut pas le manipuler directement ; et ça en utilisant des règles de passages de l'UML vers le relationnel.

Quelques notions essentielles :

1-Relation : c'est un sous ensemble du produit cartésien d'une liste de domaines. C'est en fait un tableau à deux dimensions dont les colonnes correspondent aux Domaines et dont les lignes contiennent des tuples. On associe un nom à Chaque colonne.

2-Entité : toutes entités sont transformées en table, les propriétés d'entités deviennent attribut de la table, l'identifiant de l'entité devient la clé primaire de la table.

3-Domaine : c'est l'ensemble des valeurs d'un attribut.

4-Attribut : c'est une colonne d'une relation, caractérisé par un nom.

5-Association : relation entre deux ou plusieurs entités.

6-Cardinalité : elle permet de définir les conditions de participation d'une entité à une relation. Toutefois, une entité peut participer à plusieurs relations.

7-L'abrité : est le nombre d'attributs d'une relation.

8-Clé primaire : attribut qui permet d'identifier de manière unique et certaine tout tuples de la table.

9-Clé étrangère : Attribut qui est clé primaire d'une autre entité.

Les règles de passage :

1-Transformation des classes : chaque classe du diagramme UML devient une relation, il faut choisir un attribut de la classe pouvant jouer le rôle de clé. Transformation des associations : il existe trois types d'associations :

2- « Association 1..* » : se traduit par la création d'une clé étrangère dans la relation correspondant à l'entité coté. Cette clé étrangère est la clé primaire de la relation correspondant à l'autre entité (n).

3- « Association 0..* » : Se traduit par la création d'une relation dont La clé primaire est composé des clés étrangères référençant les relations correspondant aux entités liées par l'association

3- « Association 1..1 » : on ajoute un attribut de type clé étrangère dans la relation ayant la multiplicité minimale égale à un, qui porte le nom de la clé primaire de l'autre relation connectée à l'association. Si les deux multiplicités minimales sont à un, il est préférable de fusionner les deux relations en une seule.

Si on se base sur ces règles citées, on arrive au modèle relationnel suivant :

Utilisateur (id_u, mp_u,nom_u)

Service Achats (nom_service,tel)

Service Comptabilité (nom_service,tel)

Direction Générale (nom_service,tel)

Service demandeur (nom_service,tel)

Fournisseur (id_fournisseur, fournisseur, adresse, tel)

Demande Achats (id_demande,id_commande*,id_produit*,id_fournisseur*,

Qte_stock,Qte_demandé ,Date_livraison_s,Date,Service)

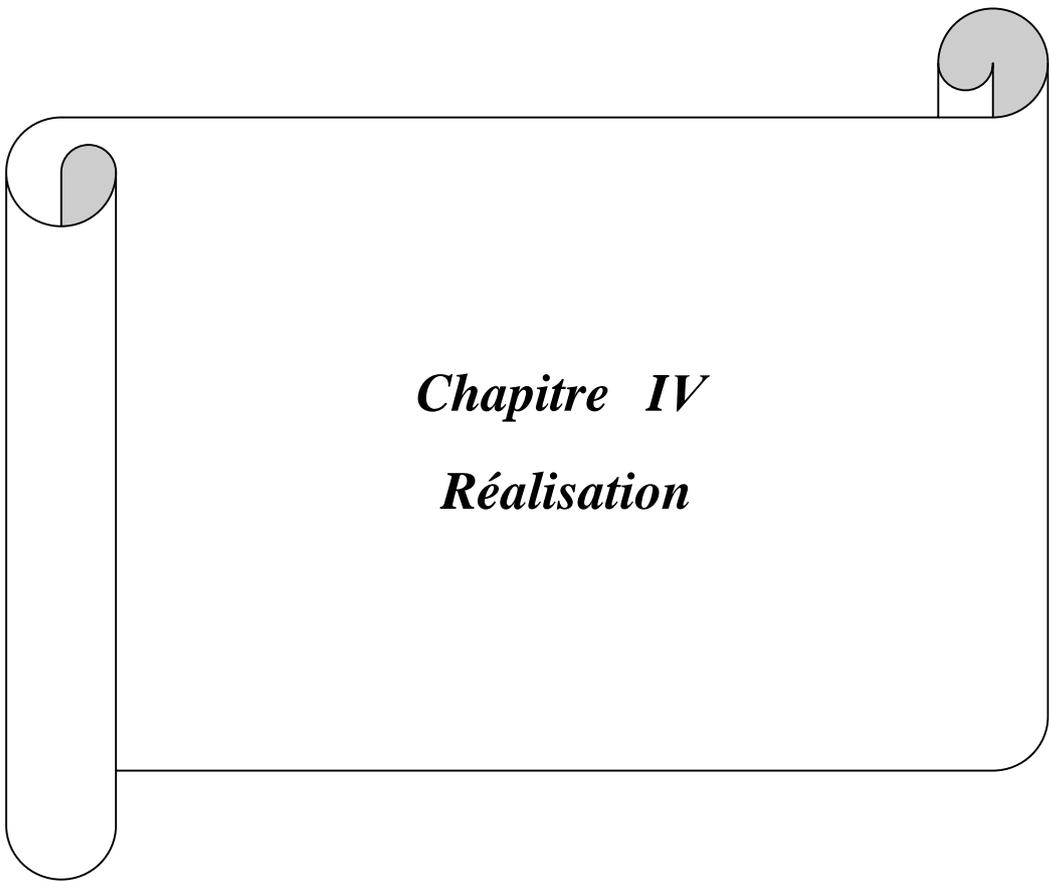
Commande(id_cmnd,prix_unitaire,PTHT,TVA,TTC,Qte_accordé)

Facture(id_facture,id_cmd*,Date_reception)

Produit(id_prod,nom_prod,unité)

11.Conclusion

Dans ce chapitre, nous avons présenté une modélisation détaillée de notre future application de gestion de l'EHP les amandiers. La prochaine étape consistera donc en la concrétisation du modèle de la solution que nous avons proposé, en d'autres termes, la réalisation des différents objets (Modèle de données, programmes, formes, etc.) qui la constituent.



Chapitre IV
Réalisation

1. Introduction

Après avoir présenté dans le chapitre précédent les différentes étapes d'analyse et de conception, nous allons présenter dans ce dernier chapitre l'environnement de

Développement et les langages qui ont servi à l'implémentation qui consiste principalement à la traduction de la conception en code source exécutable.

Premièrement on va représenter tous les outils qui ont servi à la réalisation de notre application, et nous terminerons par la présentation de ses fonctionnalités à travers ses différentes interfaces qui seront présentés sous forme de capture d'écran.

2. Les outils de développement :

2.1. Visual Studio Code [1]

C'est un éditeur de code qui nous aide à développer des technologies dédiées pour le web tel que html, css, javascript car il nous offre un code bien organisé vue l'interface qui nous offre et qui facilite la lisibilité de code quand on a saisi.

2.2. Netbeans [2]

Netbeans est un environnement de développement intégré (EDI), placé en open source par SUN en juin 2000. En plus de java, Netbeans permet également de supporter différents autres langages : C, C++, JavaScript, PHP, etc.

Netbeans est aussi une plateforme générique pour le développement des applications pour des stations de travail. Elle fournit des ressources pour développer les éléments structurants de ces applications : gestion des menus, des fenêtres, configuration, gestion des fichiers, gestion des mises à jour, etc.

¹ https://fr.wikipedia.org/wiki/Visual_Studio_Code

² <https://fr.netbeans.org/edi/articles/concours/presentation-netbeans40-1.html>

La figure suivante illustre l'interface d'accueil de Netbeans :

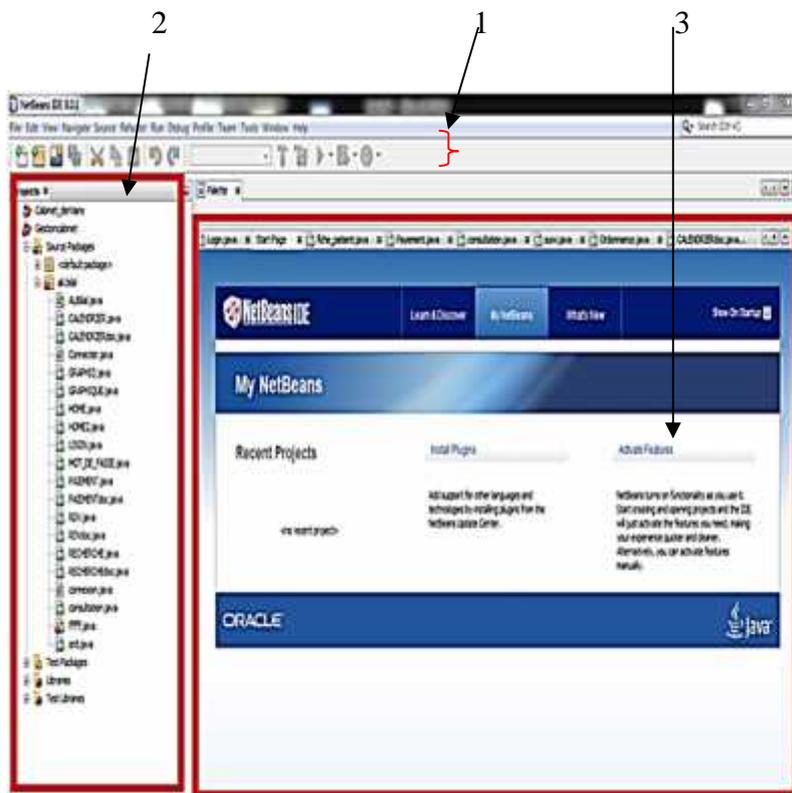


Figure IV.1 : Page d'accueil de Netbeans.

1 : Représente la barre d'outils de Netbeans.

2 : Représente la partie de l'affichage des projets et leur structure (packages, classe, etc).

3 : Représente la partie ou on doit écrire le code (après la fermeture de la « Start page » ou la création d'une nouvelle classe.

3. Les langages d'implémentation utilisés

3.1. HTML : est un langage dit de « balisage » il fait la structure de notre page web ou dont le rôle est de formaliser l'écriture d'un document avec des balises de formatage. Les balises permettent d'indiquer la façon dont doit être présenté le document et les liens qu'il établit avec d'autres documents.

Le langage HTML permet notamment la lecture de documents sur Internet à partir de machines différentes, grâce au protocole HTTP, permettant d'accéder via le réseau à des documents repérés par une adresse URL.^[3]

3.2. CSS : ^[4] c'est les feuilles de style c'est un langage qui permet de gérer la présentation d'une page web et les design des pages web.

3.3. Java script : JavaScript est un langage de programmation de scripts utilisé dans les pages web interactives afin d'améliorer la présentation et l'interactivité des pages Web et de faire des contrôles et des animations coté client

3.4. Le langage de requête SQL :^[5]

Pour communiquer avec une base de données, on a besoin de lui envoyer des commandes ou instructions appelées requêtes. Que ce soit pour la création, la suppression d'une table, la modification, l'insertion ou la sélection de données, le langage standard de requêtes est SQL.

SQL est un langage permettant d'interroger les bases de données de manière simple. Il est doté d'une syntaxe particulière que l'on doit respecter pour que la communication avec la base se passe au mieux. Son succès est dû essentiellement à sa simplicité et au fait qu'il énonce des requêtes en laissant le SGBD responsable de la stratégie d'exécution.

SQL couvre les trois fonctions indispensables à la mise en œuvre et à l'exploitation de bases de données relationnelles :

-) La création des données ;
-) La manipulation des données ;
-) Le contrôle des données

3.5. Le langage java

Java est un langage de programmation informatique orienté. Mais c'est également un environnement d'exécution.

³ https://fr.wikipedia.org/wiki/Hypertext_Markup_Language

⁴ https://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade

⁵ https://fr.wikipedia.org/wiki/Structured_Query_Language

Java peut être séparée en deux parties. D'une part, le programme écrit en langage Java et d'autre part, une machine virtuelle (JVM) qui va se charger de l'exécution du programme Java.

C'est cette plateforme qui garantit la portabilité de Java. Il suffit qu'un système ait une machine virtuelle Java pour que tout programme écrit en ce langage puisse fonctionner.

3.5.1. Java EE [6]

Java Enterprise Edition, ou **Java EE** est une spécification pour la technique Java de Sun plus particulièrement destinée aux applications d'entreprise. Dans ce but, toute implémentation de cette spécification contient un ensemble d'extensions au *Framework* Java standard afin de faciliter la création d'applications interactives et dynamiques.

4. Les serveurs

4.1. Le serveur d'application web GlassFish

GlassFish[7] est le nom du serveur d'applications Open Source Java EE 5 et désormais Java EE 7 avec la version 4.1 qui sert de socle au produit *Oracle GlassFish Server*⁴ (anciennement *Sun Java System Application Server*⁵ de Sun Microsystems). Sa partie *Toplink persistence*⁶ provient d'Oracle. C'est la réponse aux développeurs Java désireux d'accéder aux sources et de contribuer au développement des serveurs d'applications de nouvelle génération.

GlassFish est un serveur Web : il reçoit des requêtes HTTP émises depuis un navigateur, les analyse et envoie des pages Web en réponse au navigateur.

Lorsque les requêtes HTTP concernent des pages JSP (Java Server Pages), du code Java doit être exécuté pour construire les pages Web que GlassFish doit renvoyer au navigateur. Pour cela, GlassFish s'appuie sur un moteur de servlets qui prend en charge cette partie du traitement.

⁶ https://fr.wikipedia.org/wiki/Jakarta_EE

⁷ <https://fr.wikipedia.org/wiki/GlassFish>

4.2. Serveur de base de donnée MYSQL [8]

MySQL est un système de gestion de base de donnée relationnel à la fois robuste et rapide. Le serveur MySQL contrôle l'accès aux données pour s'assurer que plusieurs utilisateurs peuvent servir simultanément de la même base de données, pour y accéder rapidement.

5. Les principales interfaces de l'application :

5.1. Authentification

L'interface d'authentification est la première fenêtre télécharger et visualisé par l'utilisateur. Il devra taper son nom utilisateur, son mot de passe pour pouvoir accéder à la suite de l'application.

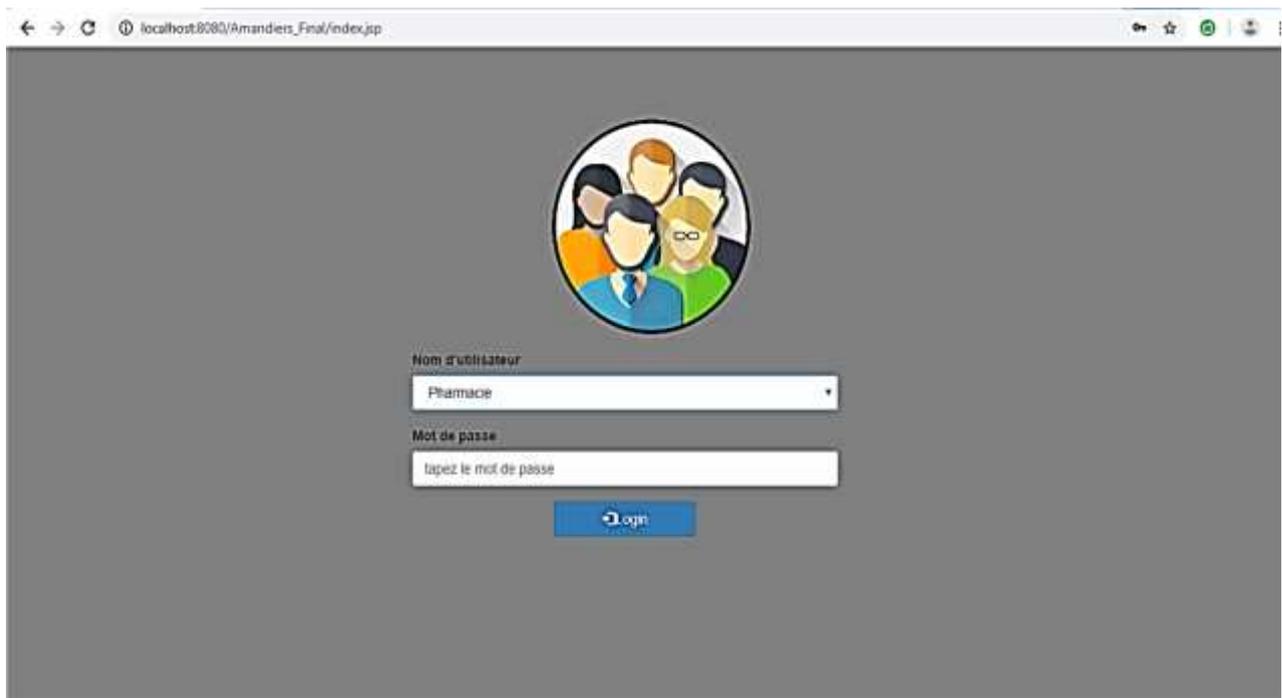
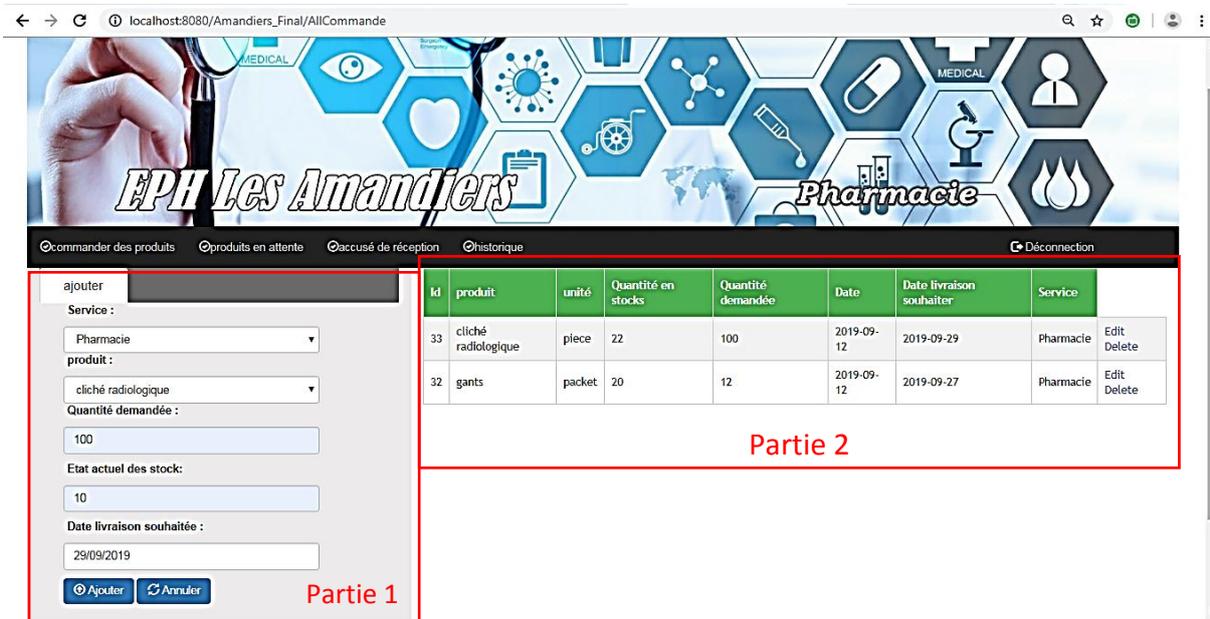


Figure IV.2: L'interface d'authentification.

⁸ <https://fr.wikipedia.org/wiki/MySQL>

5.2. Interface de demande de produit

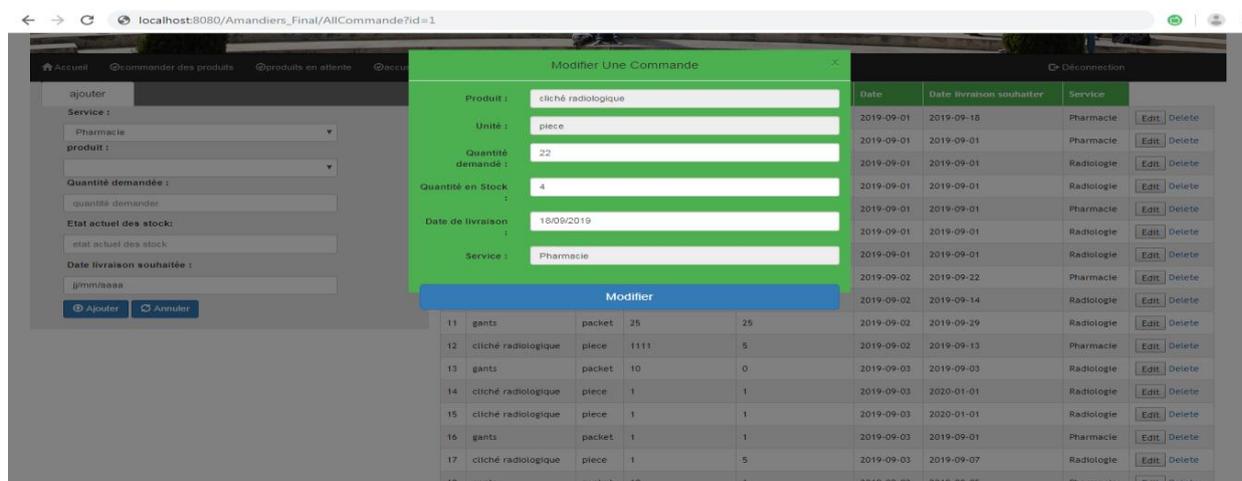
Elle permet d’afficher un formulaire d’ajout d’une demande d’achat (partie1), les détails préinscrits sur le formulaire de la (partie 1) seront affichés sur le tableau (partie 2) ; ce qui permettra à l’utilisateur d’effectuer des modifications ou des suppressions.



FigureIV.3 : Interface demande de produit

5.3. Interface Modifier une demande d'achats

Elle permet d’afficher un formulaire pour la modification des demandes d’achats effectuées précédemment.



FigureIV.4 : Interface Modifier une demande d'achats

5.4. Interface Les Demande D'Achats

Elle permet d'afficher la liste des commandes effectuées précédemment par les services demandeurs (partie 2), et pouvoir ajouter le prix unitaire et les coordonnées de fournisseur dans un formulaire (partie 1).



Figure IV.5 : Interface Les Demande D'Achats

5.5. Interface Valider une demande d'achats

Elle permet d'afficher la liste des demandes d'achats effectuées (partie 2), et pouvoir valider ces derniers ou modifier la quantité demandée (partie 1).



Figure IV.6 : Interface Valider une demande d'achats.

5.6. Interface accusé de réception

Elle permet d'afficher la liste des produits en attente de livraison (partie 1) et accusé la réception des produits. (partie 1).

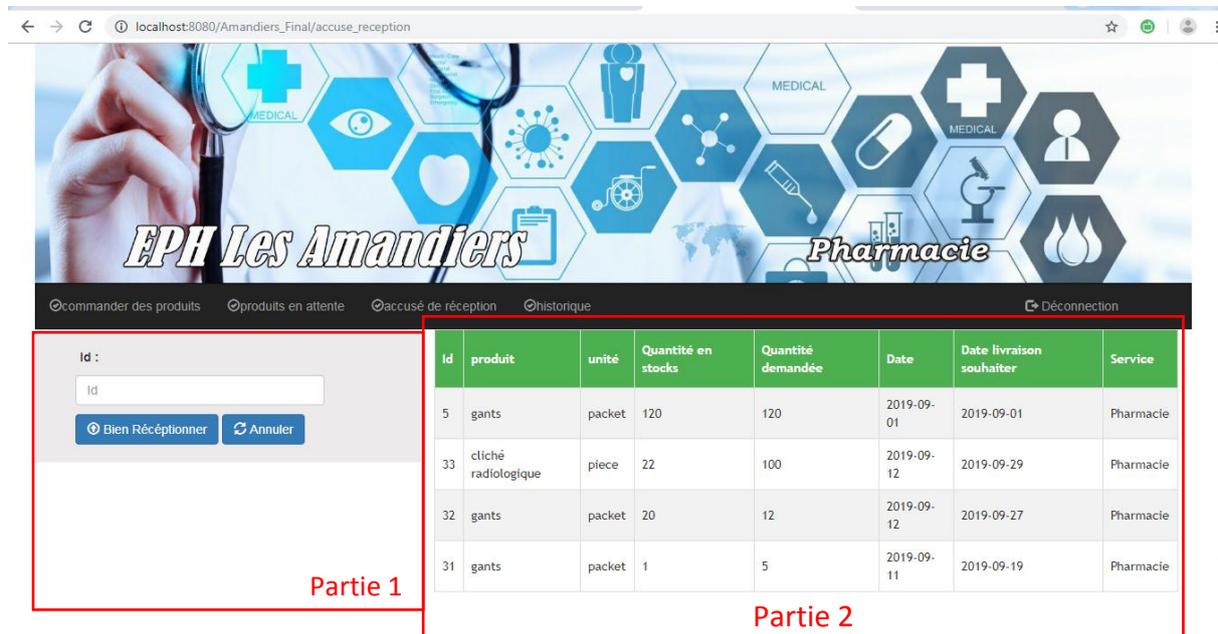


Figure IV. 7 : Interface accusé de réception.

5.7. Interface paiement des fournisseurs.

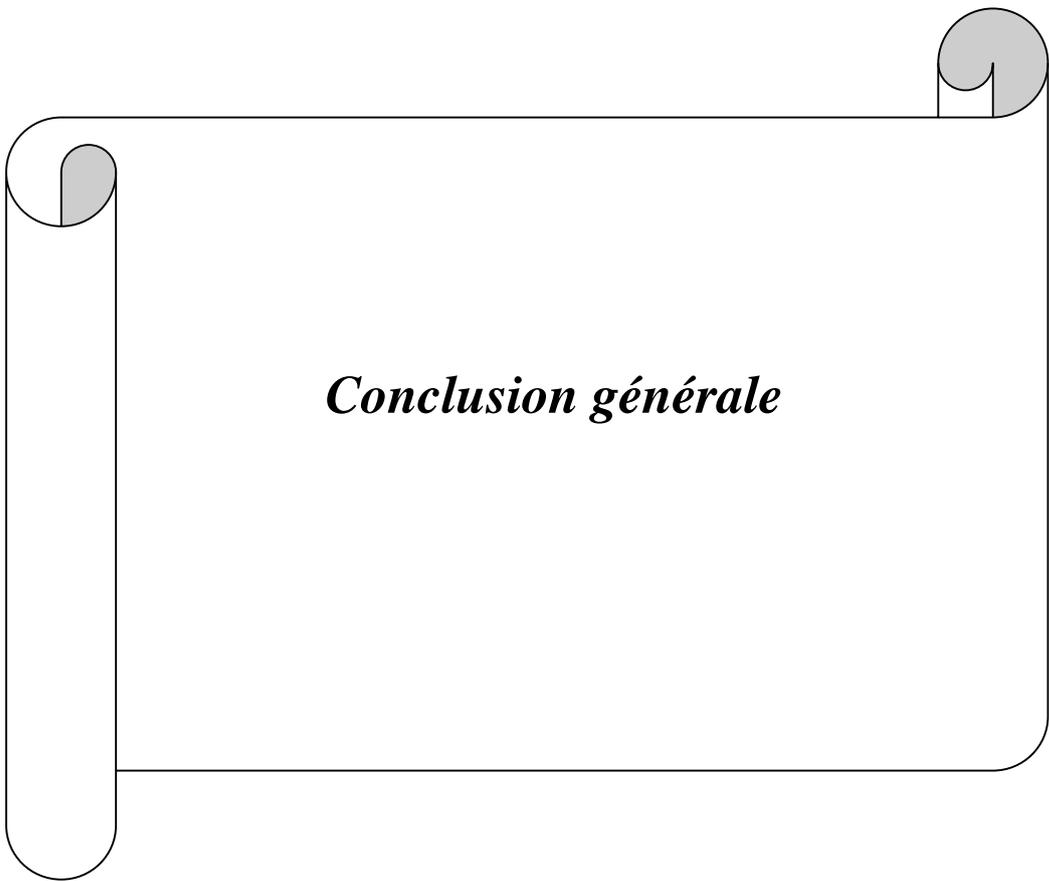
Elle permet d'afficher la liste des produits livrés non payés (partie 1) et valider le paiement une fois que la transaction d'argent est faite (partie 1).



Figure IV.8 : Interface paiement des fournisseurs.

6. Conclusion

Dans ce chapitre nous avons présenté les différents langages ainsi que les outils utilisés (matériels et logiciels) pour le développement de notre application et on a donné quelques explications sur les fonctionnalités de cette dernière par une illustration de certaines interfaces.



Conclusion générale

Conclusion Générale

Le travail que l'établissement hospitalière privé (cas les amandiers) nous a confié et que nous avons réalisé tout au long du semestre nous a permis de découvrir beaucoup d'aspect sur la manière de réalisation d'une application web soit sur le plan de conception ou sur le plan programmation et de bien comprendre et voir de près la complexité des tâches auxquelles le personnel fait face.

Notre travail a consisté le développement une application web pour l'établissement hospitalière privé (cas les amandiers), Ce travail peut se résumer en deux parties :

- ❖ Dans la première partie, nous avons fait une étude théorique sur les langages informatiques. Nous avons défini et suivi, ensuite, le langage de modélisation UML et son extension pour le web afin de bien concevoir notre application.
- ❖ Dans la deuxième partie, nous avons concrétisé notre modèle conceptuel en réalisant une application web qui répond aux objectifs fixés.

Ce travail aussi enrichissant et motivant nous a permet :

- ❖ D'abord d'acquérir de nouvelle expérience dans le développement des applications web.
- ❖ L'acquisition de nouvelle expérience dans la conception avec le langage UML. Ainsi que d'avoir de larges connaissance sur les langages spécialisé web tel que HTML, CSS, JavaScript et ainsi que le langage de programmation JEE qui est un outil très puissant dans la programmation coté serveur.
- ❖ D'approfondir nos connaissances en matière de systèmes d'informations et des bases de données et le SGBD MySQL, le langage SQL ainsi que les technologies d'accès aux bases de données avec JEE.

Cependant, malgré les diverses fonctionnalités qu'offre notre application, quelques améliorations peuvent être apportées.

Introduction générale.....	1
-----------------------------------	----------

Chapitre I : Etude Préalable

1. Introduction	3
2.Présentation de l'organisme d'accueil.....	3
2.1.historique de l'entreprise.....	3
2.2.Fiche technique et coordonnées de l'ehp.....	3
2.3Présentation des différents services sanitaires existants.....	4
2.4. Présentation des différents services extra-sanitaires existants	7
2.4.Effectif de l'ehp	9
2.5.Etat informatif de l'ehp	10
3.Etude de l'extasiant.....	14
4. Organigramme de l'ehp	14
5. Etude des postes de travail	15
5.1.Etude du poste de travail "service achats "	17
5.2. Etude du poste de travail "service comptabilité"	18
5.3. Etude du poste de travail "direction générale"	18
5.4. Etude du poste de travail "service demandeur".....	19
6. Etude des procédures.....	19
6.1. Procédure achats.....	19
7. Le diagramme des flux	21
8. Etude des documents	22
9. Etude des registres.....	29
10.Critiques et suggestion	31
11.Objectif de l'application	32
12.Conclusion.....	32

Chapitre II : Généralités sur les applications web

1. Introduction	33
2.Présentation des application web	33
2.1 Définition d'une application web dynamique.....	33
2.2. Fonction d'un application web	34
2.3. Architecture d'une application web.....	35
2.3.1. Architecture a deux niveaux(client/serveur)	35
2.3.2. Architecture a trois niveaux	36
2.3.3. Conclusion.....	37
2.4. Avantages des applications web.....	38
2.5. Notions de base	38
2.5.1. Serveur web	38
2.5.2. Serveur d'application.....	39
3. Java EE	39
3.1. Présentation	39
3.2. Architecture java EE	40
3.2.1.Modèle MVC.....	41
3.2.2.Architecture MVC.....	42
3.3. Les composants java EE.....	43
3.3.1. Les composants clients ou tiers client.....	44
3.3.2. Les composants web ou tiers web	45
3.3.3. Les composants métiers	45
3.4. Les conteneurs java EE	46
3.5. Les API java EE	48
3.5.1. Servlet.....	49
3.5.2. Java Serve Page(JSP)	49
3.5..3. Java serve pages standard tag library(jstl)	49

3.5.4. Java serve faces(jsf)	49
3.5.5. Java database connectivity(jdbc).....	49
3.5.6. Java persistance api(JPA).....	50
3.5.7. Entreprise Java beans(EJB).....	50
3.6. Les framwork java EE.....	51
3.6.1. Hibernate	51
3.6.2. Java server faces(jsf)	52
3.7. Conclusion.....	54

Chapitre III : Analyse et conception

1. Introduction	55
2. Présentation de l'uml	55
2.1. Historique	55
2.2. Les diagramme d'uml	56
3. Démarche de modélisation	57
3.1. L'analyse.....	58
3.2. Spécification des besoins	58
3.3. Identification des besoins	59
4. Acteurs et leurs taches.....	59
5. Diagramme de contexte.....	60
6. Diagramme de cas d'utilisation	61
6.1 Identification des cas d'utilisation	61
6.2. Relation entre cas d'utilisation	61
6.3. Diagramme de cas d'utilisation global	62
6.4. Diagramme de cas d'utilisation détaillé.....	63
7. Diagramme de séquence.....	66
7.1. Diagramme de séquence "cas authentification"	67
7.2. Diagramme de séquence "cas commande produits"	67

7.3. Diagramme de séquence "établissement de demande achats"	68
7.4. Diagramme de séquence « cas établissement de bon e commande externe" 68	
7.5. Diagramme de séquence "valider la demande achats"	69
7.6. Diagramme de séquence "cas accusé de réception"	69
7.7. Diagramme de séquence "cas demande de paiement"	70
7.8. Diagramme de séquence "cas de paiement"	70
8. Diagramme d'activité.....	71
8.1. Diagramme d'activité "authentification"	71
8.2. Diagramme d'activité "ajouter une commande"	72
8.3. Diagramme d'activité "modifier une commande"	72
8.4. Diagramme d'activité "supprimer une commande"	73
8.5. Diagramme d'activité "établissement d'une commande"	73
8.6. Diagramme d'activité "valider une demande achats"	74
8.7. Diagramme d'activité "modifier demande achats"	74
8.8. Diagramme d'activité " accusé de réception d'un produit "	75
8.9. Diagramme d'activité "demande de paiement"	75
8.10. Diagramme d'activité "paiement"	76
8.11. Diagramme d'activité "recherche dans historique"	76
9. Diagramme de classe	77
9.1 Identification des classes	77
10. Modèle relationnel.....	81
11. Conclusion.....	83

Chapitre IV : Réalisation

1. Introduction	84
2. Les outils de développement	84
2.1. Visual studio code	84
2.2. Netbeans	84

3. Les langages d'implémentation utilisés.....	85
3.1.html	85
3.2. Css	86
3.3. Javascript.....	86
3.4. Le langage de requête SQL.....	86
3.5. Le langage java.....	86
3.5.1. Java EE.....	87
4. Les serveurs.....	87
4.1. Le serveur d'application web glassfish.....	87
4.2. Serveur de base de donnée MYSQL.....	88
5. Les principales interfaces de l'application.....	88
5.1 Authentification.....	88
5.2. Interface de demande de produit.....	89
5.3. Interface Modifier une demande d'achats	89
5.4. Interface Les Demande D'Achats.....	90
5.5. Interface Valider une demande d'achats.....	90
5.6. Interface accusé de réception	91
5.7. Interface paiement des fournisseurs.....	91
6. Conclusion	92
Conclusion générale	93
Bibliographie.	
Annexes.	

Liste des figures

Figure I.1 : Localisation de l'EHP.	4
Figure I.2 : logiciel CHIFA_DIALYSE	11
Figure I.3 : Logiciel NX Viewer.	12
Figure I.4 : Logiciel Scintron 7 MIE	13
Figure I.5 : Logiciel intelliX	14
Figure I.6 : Organigramme de l'EHP.....	15
Figure I.7 : diagramme des flux	21
Figure II.1 : Fonctionnement d'une application web.....	35
Figure II.1 : architecture deux tiers	36
<i>Figure II.2</i> : architecture à trois tiers.....	37
Figure 3 : architecture Java EE.	40
Figure II.4 : architecture MVC.	42
Figure II.5 : composant JEE.	45
Figure II.6 : composants JEE.....	46
Figure II.7 : conteneur.	47
Figure III.1 : Démarche de modélisation adoptée.	58
Figure III.2 : Diagramme de contexte.	60
Figure III.3 : Diagramme de cas d'utilisation global.	62
Figure III.4 : Diagramme de cas d'utilisation détaillé.....	63
Figure III.5 : Diagramme de séquence "cas commande produit"	67
Figure III.6 : Diagramme de séquence "cas établissement de demande d'achats ".	68
Figure III.7 : Diagramme de séquence "cas établissement de bon de commande externe "	68
Figure III.8 : Diagramme de séquence "cas valider la demande d'achats ".	69

Figure III.9 : Diagramme de séquence "cas accusé de réception ".....	69
Figure III.10 : Diagramme de séquence "cas demande de paiement ".....	70
Figure III.11 : Diagramme de séquence "cas de paiement ".....	70
Figure III.12 : Diagramme d'activité « authentification ».....	71
Figure III.13 : Diagramme d'activité « Ajouter une commande».....	72
Figure III.14 : Diagramme d'activité « Modifier une commande».....	72
Figure III.15 : Diagramme d'activité « Supprimer une commande».....	73
Figure III.16 : Diagramme d'activité « Etablissement d'une demande d 'achats».....	73
Figure III.17 : Diagramme d'activité « Valider une demande d'achats».....	74
Figure III.18 : Diagramme d'activité « Modifier une demande d'achats».....	74
Figure III.19 : Diagramme d'activité « Accusé de réception d'un produit».....	75
Figure III.20 : Diagramme d'activité « Demande de paiement».....	75
Figure III.21 : Diagramme d'activité « paiement».....	76
Figure III.22 : Diagramme d'activité «Recherche dans historique».....	76
Figure III.23 : Diagramme de classe.....	81
Figure IV.1 : page d'accueil de Netbeans.....	85
Figure IV.2: l'interface d'authentification.....	88
Figure IV.3: Interface commander produit.....	89
Figure IV.4: Interface Modifier une demande d'achats.....	89
Figure IV.5: Interface les demandes d'achats.....	90
Figure IV.6: Interface Valider une demande d'achats.....	90
Figure IV.7: Interface accusé de réception.....	91
Figure IV.8: Interface paiement des fournisseurs.....	91

Bibliographie

Webographie

<https://fr.wikipedia.org/wiki/Chirurgie>

<https://fr.wikipedia.org/wiki/P%C3%a9diatrie>

https://fr.wikipedia.org/wiki/Chirurgie_maxillo-faciale

https://fr.wikipedia.org/wiki/Gyn%C3%a9cologie_obst%C3%a9trique

https://fr.wikipedia.org/wiki/Chirurgie_visc%C3%a9rale

<https://www.aqepa.org/quest-ce-quun-ortl/>

https://fr.wikipedia.org/wiki/Chirurgie_gyn%C3%a9cologique

<https://fr.wikipedia.org/wiki/Orthop%C3%a9die>

https://fr.wikipedia.org/wiki/Imagerie_m%C3%a9dicale

<https://sante.journaldesfemmes.fr/quotidien/2540946-oncologue-cancerologue-medecin-definition/>

<http://www.chups.jussieu.fr/polys/anapath/Cours/POLY.Chp.1.html>

<https://sante-medecine.journaldesfemmes.fr/faq/20654-laboratoire-d-analyse-medicale-definition>

https://fr.wikipedia.org/wiki/M%C3%A9decine_nucl%C3%A9aire

<https://fr.wikipedia.org/wiki/Endoscopie>

<https://fr.wikipedia.org/wiki/H%C3%a9modialyse>

<https://fr.wikipedia.org/wiki/Pharmacie>

<https://fr.yeepley.com/blog/6-types-developpement-d-applications-web/>

<http://cours-fad-public.ensg.eu/mod/imscp/view.php?Id=457>

<https://www.supinfo.com/articles/single/574-architecture-2-tiers-vs->

https://www.webexpert.ch/publications/application_web

<https://www.supinfo.com/articles/single/6628-architecture-applications-java-ee>

<https://www.supinfo.com/articles/single/6628-architecture-applications-java-ee>

<https://www.supinfo.com/articles/single/6628-architecture-applications-java-ee>

<https://www.supinfo.com/articles/single/6628-architecture-applications-java-ee>

<https://www.supinfo.com/articles/single/6628-architecture-applications-java-ee>

<https://www.jmdoudoux.fr/java/dej/chap-frameworks.htm>

http://gardeux-vincent.eu/Documents/projetjee/BVC_Hibernate_Struts2/Hibernate/index.htm

<https://www.supinfo.com/articles/single/6402-primefaces-java-introduction>

[https://fr.wikipedia.org/wiki/UML_\(informatique\)](https://fr.wikipedia.org/wiki/UML_(informatique))

[https://fr.wikipedia.org/wiki/UML_\(informatique\)](https://fr.wikipedia.org/wiki/UML_(informatique))

[https://fr.wikipedia.org/wiki/UML_\(informatique\)](https://fr.wikipedia.org/wiki/UML_(informatique))

https://www.memoireonline.com/07/08/1363/m_conception-realisation-application-gestion-centre-kinesie9.html

https://www.memoireonline.com/07/08/1363/m_conception-realisation-application-gestion-centre-kinesie9.html

https://www.memoireonline.com/06/11/4548/m_Conception-et-implementation-dun-systeme-de-gestion-des-centres-dacces-a-internet--rechargemen10.html

https://fr.wikipedia.org/wiki/Diagramme_des_cas_d%27utilisation

https://fr.wikipedia.org/wiki/Diagramme_des_cas_d%27utilisation

https://fr.wikipedia.org/wiki/Visual_Studio_Code

<https://fr.netbeans.org/edi/articles/concours/presentation-netbeans40-1.html>

https://fr.wikipedia.org/wiki/Hypertext_Markup_Language

https://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade

https://fr.wikipedia.org/wiki/Structured_Query_Language

https://fr.wikipedia.org/wiki/Jakarta_EE

<https://fr.wikipedia.org/wiki/glassfish>

<https://fr.wikipedia.org/wiki/mysql>