République **A**lgérienne **D**émocratique et **P**opulaire
Ministère de l'**E**nseignement **S**upérieur et de la **R**echerche **S**cientifique

UNIVERSITE **M**OULOUD **M**AMMERI DE **T**IZI-**O**UZOU

FACULTE DU **G**ENIE **E**LECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Domaine : **Mathématiques et Informatique**
Filière : **Informatique**
Spécialité : **Intitulé de la spécialité**

Présenté par
Yamina SID ALI
Dihia SADI HADDAD

Thème

Analyse, Conception et réalisation d'une application Java EE
Cas : Gestion des achats locaux et imports pour ENIEM.

Mémoire soutenu publiquement le ……/07/ 2016 devant le jury composé de :

Président : M Prénom NOM
Encadreur : M M'hand KERBICHE
Examinateur : M Prénom NOM
Examinateur : M Prénom NOM

REMERCIEMENTS

Tout travail de recherche n'est jamais totalement l'œuvre d'une seule personne, à cet effet, nous tenons à exprimer notre sincère reconnaissance et nos vifs remerciements pour :

- Notre bon Dieu tout puissance et compatissant pour le courage et la patience qu'il nous a donné afin de mener ce projet à terme ;

- Notre promoteur Mr KARBICHE M'hand qui à fait l'honneur de nous accepter, de diriger notre mémoire et pour ses précieux conseils et son soutien ;

- Tous les enseignants du département informatique qui ont contribué à notre formation ;

- A tous ceux qui ont aidé, de prés ou de loin, à l'élaboration de ce modeste travail ;

- Et aussi aux membres du jury pour l'honneur qu'ils nous font en acceptant d'examiner et juger notre travail.

Je dédie ce Modeste travail

À ceux qui sont la source de mon inspiration et de mon courage, à qui je dois de l'amour et de la reconnaissance, mes très chers parents.

A mes frères et sœurs que j'adore énormément.

A mes petits anges 'Lyna, Malek, Houssem Eddine et Yasser'.

A mes deux très chères copines ' Thilleli et Dihia ' ainsi que leurs familles.

A tous mes proches et mes amis surtout Nabil.

A tous ceux qui m'ont aidé à la collaboration de ce travail.

A mon promoteur Mr KARBICHE M'hand.

Et surtout à ma binôme Dihia sans qui ce mémoire n'aurait pas pu être effectué.

YAMINA

Je dédie ce Modeste travail

À ceux qui sont la source de mon inspiration et de mon courage, à qui je dois de l'amour et de la reconnaissance, mes très chers parents.

A mes deux frères que j'aime énormément : Ramdan et Sedik.

A mon adorable petite sœur : Kenza.

A ma très chère copine Feroudja ainsi que toute sa famille.

A tous mes proches et mes amies 'Ghania, Chadia, Tassadite, Houria,…'

A tous ceux qui m'ont aidé à la collaboration de ce travail.

A mon promoteur Mr KARBICHE M'hand.

Et surtout à ma binôme Yamina sans qui ce mémoire n'aurait pas pu être effectué.

Dihia

SOMMAIRE

Sommaire

Introduction générale ..…….p01

Chapitre I : Présentation de la technologie Java EE

Introduction……p03

I.1. Conception architecturale ……p03

I.1.1 Architecture logicielle …..…… p03

I.1.2. Les vues de l'architecture logicielle …….p03

I.1.3. Utilité d'une architecture logicielle ……..p04

I.1.4. L'architecture client/serveur …….p05

I.1.4.1. Définition ……...p06

I.1.4.2. Notions de base ……..p06

I.1.4.3. Classification des architectures client/serveur …….....p07

I.1.4.4. Avantages et inconvénients de l'architecture client/serveur ……p09

I.2. Java Entreprise Edition ……p10

I.2.1. Définition ……..p10

I.2.2. Fonctionnement interne ……p11

I.2.3. L'architecture d'une application Java EE …….p11

I.2.3.1. La couche interface utilisateur ou couche présentation …….p12

I.2.3.2. La couche métier ……....p13

I.2.3.3. La couche d'accès aux données ou couche de persistance ……p15

I.2.3.4. Le modèle MVC « Modèle-Vue-Contrôleur » ……..p15

I.2.4. L'environnement d'exécution des applications Java EE ……..p16

I.2.4.1. Les conteneurs ……...p17

I.2.5. Les avantages de l'utilisation de Java EE …….p17

Conclusion……..p17

Sommaire

Chapitre II : Présentation de l'organisme d'accueil

Introduction……p19

II. Présentation de l'organisme d'accueil ENIEM ……p19

II.1. Situation géographique ……...p19

II.2. Historique ……...p20

II.3. Champ d'activité de l'ENIEM ……...p20

II.4. Missions et objectifs de l'entreprise ……...p21

II.4.1. Missions ……...p21

II.4.2. Objectifs ……..p22

II.5. Organisation générale de l'ENIEM ……p23

II.5.1 Organigramme de l'entreprise ENIEM ……p23

II.5.2 Organigramme de l'unité de prestation technique ……p24

II.5.3 Organigramme de département commercial ……p25

II.5.4 Organigramme de service d'achat ……p26

II.6. Situation informatique ……p26

Conclusion ……p27

Chapitre III : Analyse et conception

Introduction ……p30

III.1. Etude de l'existant ……..p30

III.1.1. Quelques définitions ……p30

III.1.2. Etudes des postes de travail ……p31

III.1.3. Etude des documents……p34

III.1.4. Diagramme de flux……..p35

III.1.4.1. Concepts du diagramme……..p35

III.1.4.2. Présentation du diagramme de flux……..p37

Sommaire

III.1.4.3. Description des flux d'information …….p38

III.1.5. Critiques de l'existant et suggestions ……..p40

III.1.5.1. Niveau organisationnel ……p40

III.1.5.2. Niveau informationnel …….p41

III.1.5.3. Niveau technique …….p41

III.1.5.4. Solution informatique …….p42

III.2. Analyse et conception ……p43

III.2.1. Présentation des différentes phases ……p43

III.2.1.1. Phase d'Initialisation …….. p43

III.2.1.2. Phase d'Elaboration ……p44

III.2.1.3. Phase de Construction ……p44

III.2.1.4. Phase de Transition …… p44

III.2.2. Identification des acteurs …….. p44

III.2.2.1. Diagramme de contexte ……..p45

III.2.2.2. Représentation des diagrammes de cas d'utilisation ……p46

III.2.2.3. Les cas d'utilisations ……..p48

III.2.3. Définition des itérations ……p51

III.2.3.1. Réalisation des itérations ……..p52

Conclusion …….p59

Chapitre 4. Réalisation

Introduction…….p61

IV.1. Les outils et l'environnement de développement …….p61

IV.1.1. Netbeans IDE ……p61

IV.1.2. Hibernate Framework ……..……..p62

IV.1.3. JSF ……p63

IV.1.3.1. Structure d'une application JSF ……p63

IV.1.4. Le serveur Web Apache Tomcat ……...p64

IV.1.5. Oracle Database Express ……..p64

IV.2. Les langages de programmation ……...p65

IV.2.1. Java ……p65

IV.2.2. XHTML …….p66

IV.2.3. Java Entreprise Edition (Java EE) …….p66

IV.3. Le modèle relationnel ……...p66

IV.3.1. Le modèle physique de données ……p68

IV.4. Présentation de quelques interfaces de l'application ……...p79

Conclusion …….p80

Conclusion général ……....p81

Liste des figures :

Chapitre I :

Figure I.1 : Dialogue client/serveur……..p7

Figure I.2 : Architecture client/serveur à deux tiers….....p7

Figure I.3 : Architecture client/serveur à trois tiers…… p8

Figure I.4 **:** architecture multi-niveaux…….p9

Figure I.5 : MVC avec Java EE……..p16

Chapitre II :

Figure II.1 : Vue satellitaire du site de l'entreprise……p19

Figure II.2 : Organigramme de l'entreprise ENIEM…….p23

Figure II.3 :Organigramme de l'unité de prestation technique……..p24

Figure II.4 :Organigramme du département commercial……...p25

Figure II. 5 : Organigramme de service d'achat ……p26

Chapitre III :

Figure III.1 : représentation des flux d'informations……p36

Figure III.2 : diagramme de flux…….p37

Figure III.3 : Architecteur globale du modèle proposée……p42

Figure III.4: Diagramme de contexte……..p45

Figure III.5 : Diagramme de cas d'utilisation…….p47

Figure III.6 : Diagramme de séquence du cas d'utilisation : «Authentification»……p55

Figure III.7 : Diagramme de séquence du cas d'utilisation : «Changer password»……p54

Figure III.8 : Diagramme de séquence du cas d'utilisation : « supprimer une commande».p56

Figure III.9 : Diagramme de séquence du cas d'utilisation : « modifier une facture»…….p57

Figure III.10 : diagramme de classe pour l'itération2……p58

Chapitre IV

Figure IV.1 : l'interface de NetBeans IDE……..p62

Figure IV.2 : l'interface de Hibernate…….p63

Figure IV.3 : l'interface oracle……p65

Figure IV.4 : page d'authentification…….p79

Figure IV.5 : page ajouter DAI……...p79

Figure IV.6 : page suppression d'une facture……...p80

Liste des tableaux :

Chapitre III :

Tableau III.1 : Fiche d'analyse poste N°01 ……p32

Tableau III.2 : Fiche d'analyse poste N°02 ……p32

Tableau III.3 : Fiche d'analyse poste N°03 …...…….p33

Tableau III.4 : Fiche d'analyse poste N°04 ……p33

Tableau III.5 : Liste des documents ……...p34

Tableau III.6 : Description des flux d'information …….p39

INTRODUCTION GENERALE

Introduction générale

L’Entreprise est une entité économique financièrement indépendante réunissant des moyens de production en vue de créer des biens et services pour la satisfaction de sa clientèle et la réalisation d’un profit. Sa croissance passe par une organisation établie sur des bases bien définies et des objectifs clairement fixés, tenant compte de son environnement et de la gestion rigoureuse de ses différentes fonctions. L’entreprise permet d’avoir une bonne politique de gestion, lui assurant des produits à moindre coût et de bonne qualité.

Par ailleurs, l'informatique ayant envahi tous les domaines de la vie. Dans notre cas, étant affectés à l’Entreprise Nationale des Industries en Electroménager (ENIEM) afin de développer une application pour garantir le bon fonctionnement du service gestion des achats locaux et imports.

Notre travail consiste à l’informatisation de la gestion des achats de l’entreprise ENIEM et le projet porte sur le développement d’une application JavaEE qui répondra aux contraintes de fiabilité, efficacité et surtout de disponibilité.

Notre travail se décompose en 4 chapitres qui sont :

- Chapitre I : Présentation de l’organisme d’accueil.
- Chapitre II : La Technologie de JavaEE.
- Chapitre III : Analyse et conception.
- Chapitre IV : La Réalisation.

Chapitre I :

Présentation de la technologie Java Entreprise Edition

Introduction

De nos jours, le génie logiciel nous offre des nouvelles technologies qui ont les potentialités de révolutionner le monde des conceptions des systèmes informatiques. Un bon logiciel ne verra jamais le jour sans utiliser une bonne architecture logicielle qui respecte les critères de fiabilité, sécurité, portabilité…

Dans ce chapitre on va étudier une technologie d'architecture logicielle (Java Entreprise Edition) qui est nécessaire pour la réalisation des applications des entreprises.

I.1. Conception architecturale [1]

I.1.1. Architecture logicielle :

L'architecture logicielle décrit d'une manière symbolique et schématique les différents éléments d'un ou de plusieurs systèmes informatiques, leurs interrelations et leurs interactions. Contrairement aux spécifications produites par l'analyse fonctionnelle, le modèle d'architecture, produit lors de la phase de conception, ne décrit pas ce que doit réaliser un système informatique mais plutôt comment il doit être conçu de manière à répondre aux spécifications. L'analyse décrit le « quoi faire » alors que l'architecture décrit le « comment le faire ».

I.1.2. Les vues de l'architecture logicielle :

Selon P.Kruchten, l'architecture logicielle ce présente selon « 4+1 »vues :

- **Vue logique :** elle se concentre sur la modélisation des principaux éléments d'architecture et mécanismes logiciels, la définition des éléments du domaine (objet métier, diagrammes d'états de ces objets…). Elle comprend en particulier les modèles d'analyse et de conception du système, c'est la vue des équipes de conception et développement.

- **Vue composant ou vue implémentation :** elle permet d'identifier les modules (composants logiciels) qui implémentent les éléments définis dans la vue logique, de les regrouper en composants logiciels, d'identifier les dépendances d'intégration entre ces composants ; c'est la vue des équipes d'intégration.

- **Vue processus :** en environnement multi-tâches, cette vue de l'architecture permet de définir les processus, la coordination et la synchronisation des processus, les threads d'exécution… Cette vue est optionnelle et n'est utile que dans le cas d'architecture complexes multi-tâches.

- **Vue déploiement :** elle précise l'architecture de production (ressources matérielles, implantation des composants, pilotage…). Cette vue permettra par exemple de s'assurer que l'application répond aux contraintes de déploiement, aux exigences de qualité de service (montée en charge, temps de réponse, haute-disponibilité…) et s'intègre aux infrastructures de supervision, etc.

- **Vue use-cases (vue « +1») :** elle se concentre sur un sous-ensemble des cas d'utilisations qui ont une influence significative sur l'architecture du système. Ces use-cases structurants permettent d'identifier les fonctionnalités et contraintes importantes, les risques majeurs de l'architecture, ce sont eux qui guident l'élaboration des quatre autres vues de l'architecture, de la conception à la mise en production de l'application.

I.1.3. Utilité d'une architecture logicielle :

L'utilité d'une architecture logicielle est :

- Compréhension : facilite la compréhension des grands systèmes complexes en donnant une vue de haut-niveau de leurs structures et de leurs contraintes. Les motivations des choix de conception sont ainsi mises en évidence.

- Réutilisation : favorise l'identification des éléments réutilisables, parties de conception, composants, caractéristiques, fonctions ou données communes.

- Construction : fournit un plan de haut-niveau du développement et de l'intégration des modules en mettant en évidence les composants, les interactions et les dépendances.

- Évolution : met en évidence les points où un système peut être modifié et étendu. La séparation composant/connecteur facilite une implémentation du type« plug-and-play»

- Analyse : offre une base pour l'analyse plus approfondie de la conception du logiciel, analyse de la cohérence, test de conformité, analyse des dépendances.

- Gestion : contribue à la gestion générale du projet en permettant aux différentes personnes impliquées de voir comment les différents morceaux de casse-tête seront agencés. L'identification des dépendances entre composants permet d'identifier où les délais peuvent survenir et leurs impacts sur la planification générale.

I.1.4. L'architecture client/serveur [3] :

Client-serveur est une expression que tout informaticien utilise aujourd'hui, autant pour afficher sa connaissance des techniques actuelles que parce qu'elle est vraiment incontournable. Cette expression fait référence à une mosaïque de conseils qui touche de nombreux domaines en informatique.

Au début des années 90, le monde de client-serveur n'était une réalité que pour quelques rares entreprises à la pointe de l'innovation.

Aujourd'hui, le nom est devenu familier mais il est souvent mal employé. L'architecture client/serveur est considérée comme l'évolution la plus fondamentale depuis l'arrivée du microordinateur.

I.1.4.1. Définition :

L'architecture client/serveur désigne un mode de communication entre deux processus. Le premier appelé client, demande des services au deuxième processus qui est le serveur, ce dernier exécute les requêtes du client et envoie les résultats en retour.

I.1.4.2. Notions de base :

- **Client :** Processus qui demande l'exécution d'une opération à un autre processus serveur en émettant un message décrivant l'opération à exécuter et attendant la réponse à cette opération par un message en retour.

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

- **Requête :** Message transmis par un client à un serveur décrivant l'opération à exécuter pour le compte client.

- ***Réponse* :** Message transmis par un serveur à un client suite à l'exécution d'une opération contenant les paramètres de retour de l'opération.

- **Middleware :** Un des composant clé de l'architecteurs client/serveur est le middleware qui est simplement un ensemble de services logiciels qui jouent le rôle d'intermédiaire entre les clients et les serveurs en assurant le dialogue entre ces derniers d'une manière transparente permettant de cacher l'hétérogénéité des composantes mis en jeux (réseaux, SGBD…)

  Exemple d'un middleware : SOAP qui est un protocole d'échange de messages, permettant d'invoquer des applications sur des réseaux distribués et décentralisés. Simple et facile à implémenter dans les serveurs web ou dans les serveurs d'applications.

En résumer, la figure suivante illustre ces notions :

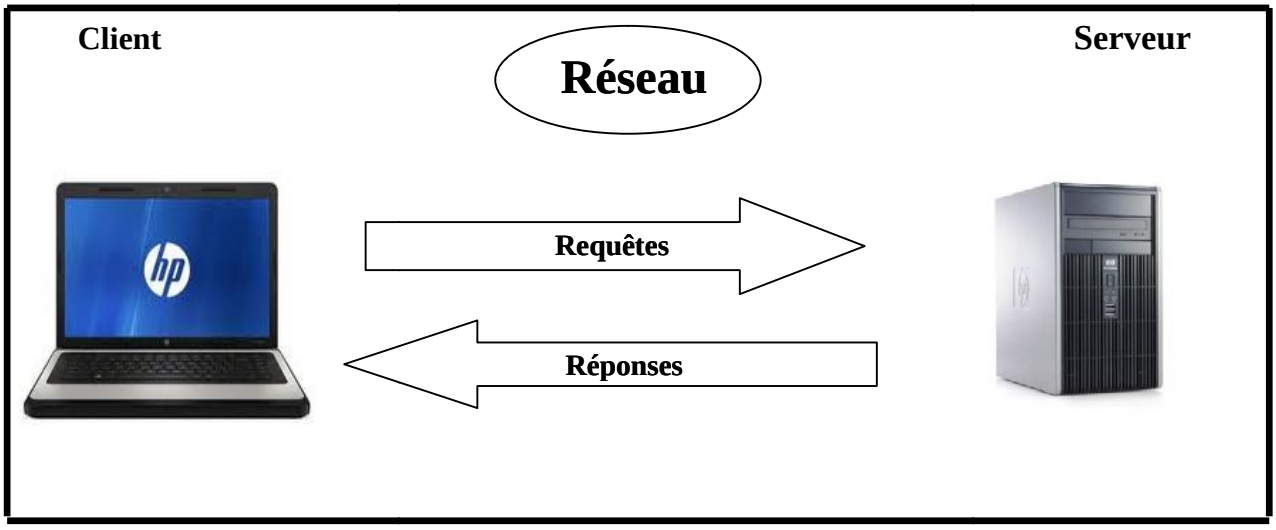


Figure I.1 : Dialogue client/serveur.

I.1.4.3. Classification des architectures client/serveur [4] :

On distingue plusieurs types d'architectures client/serveur :

A. Architecture à deux niveaux :

L'architecture à deux niveaux (aussi appelée *architecture 2-tiers)*, caractérise les systèmes clients/serveurs dans lesquels le client demande une ressource et le serveur la lui fournit directement. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir le service. Le schéma qui le présente :

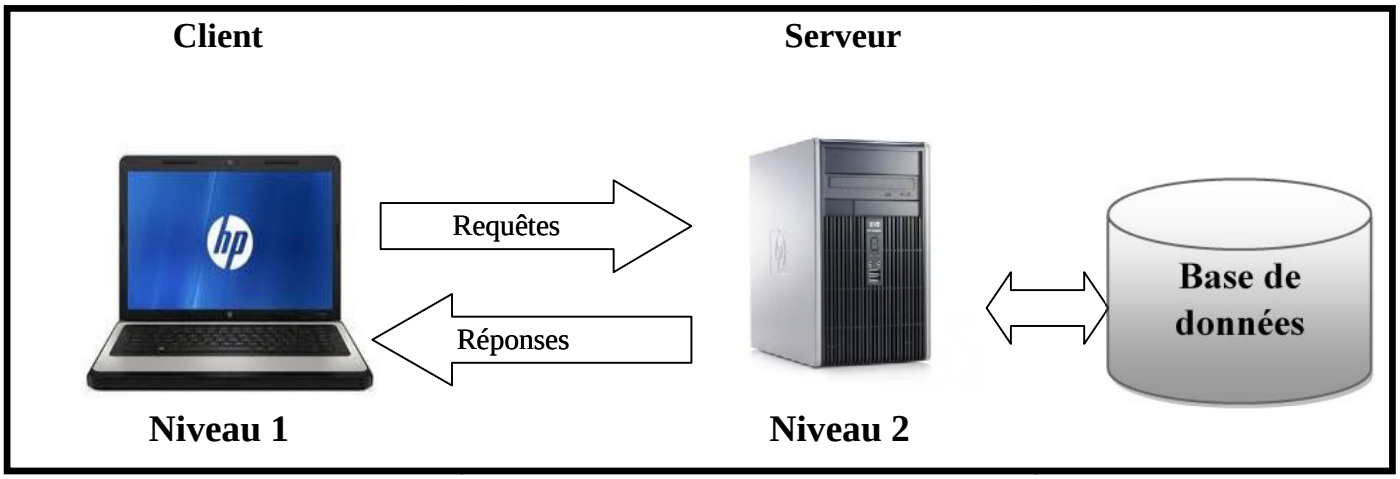


Figure I.2 : Architecture client/serveur à deux tiers

B. Architecture à trois niveaux :

Dans l'architecture à 3 niveaux (appelée *architecture 3-tiers*), il existe un niveau intermédiaire, c'est-à-dire que l'on a généralement une architecture partagée entre:

- **Le client :**

  Le demandeur de ressources, équipé d'une interface utilisateur (généralement un navigateur Web) chargé de la présentation.

- **Le serveur d'application** (appelé également **middleware**) :

  Chargé de fournir la ressource mais en faisant appel à un autre serveur.

- **Le serveur de données** :

  Fournissant au serveur d'application les données dont il a besoin.

Le schéma qui le présente :

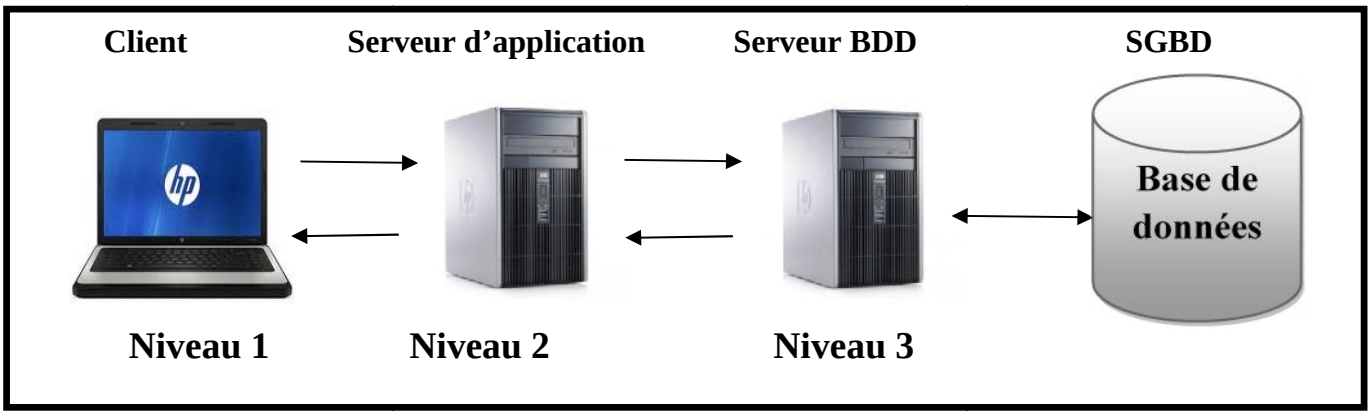


Figure I.3 : Architecture client/serveur à trois tiers

C. Architecture multi niveaux :

L'architecture multi niveaux (aussi appelée architecture n-tiers) a été pensée pour pallier aux limitations des architectures trois tiers et concevoir des applications puissantes et simples à maintenir. Dans cette architecture chaque serveur (niveaux 2 et 3) effectue une tâche spécialisée. Un serveur peut donc utiliser les services d'un ou plusieurs autres serveurs afin de fournir son propre service. Par conséquent, l'architecture à trois niveaux est potentiellement une architecture à N niveaux.

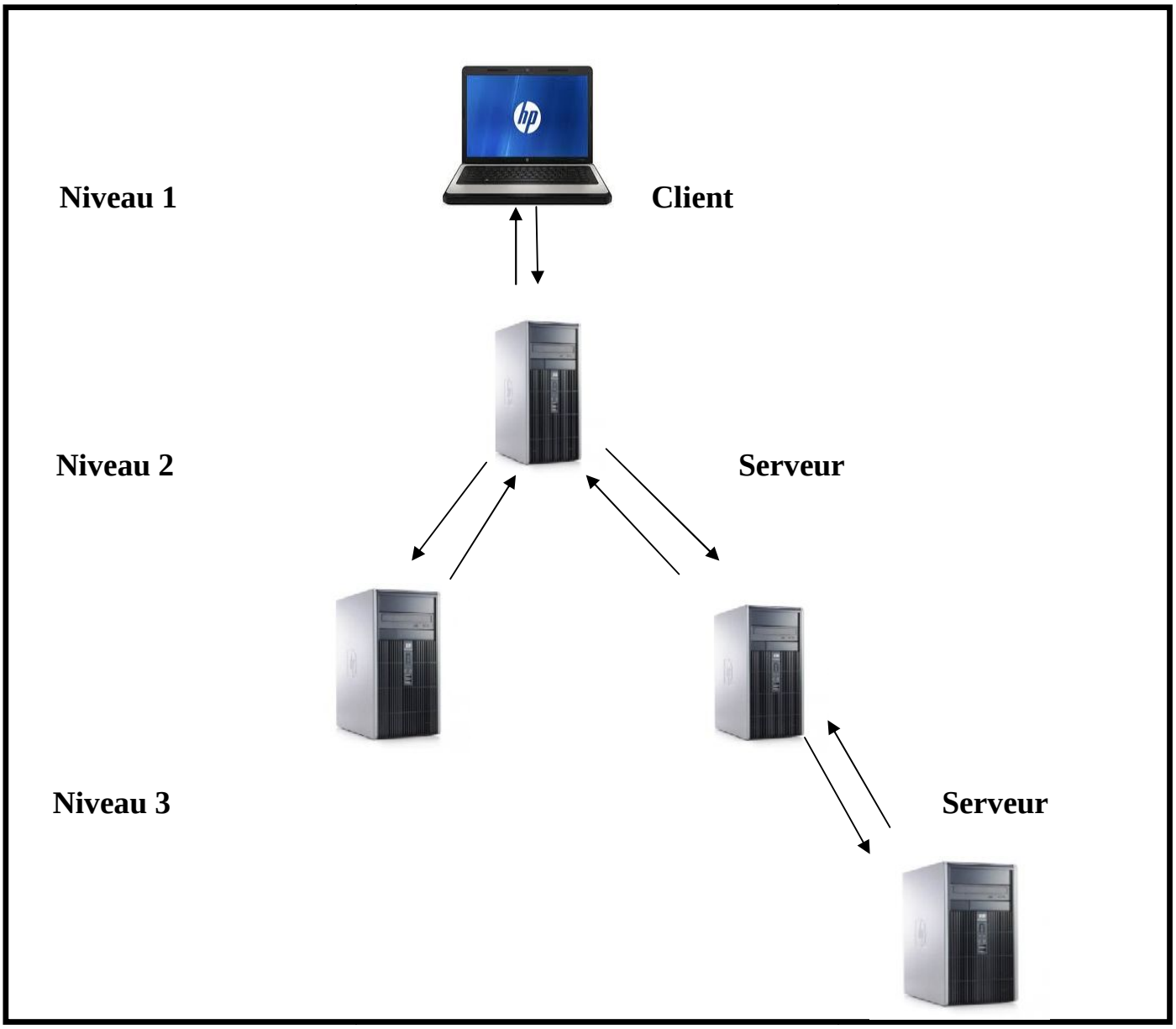


Figure I.4 : architecture multi-niveaux.

I.1.4.4. Avantages et inconvénients de l'architecture client/serveur [5]:

- **Les avantages :**
  - ***Des ressources centralisées*** : Étant donné que le serveur est au centre du réseau, il peut gérer des ressources communes à tous les utilisateurs, comme par exemple une base de données centralisées afin d'éviter les problèmes de redondance et de contradiction.

- ***Une meilleure sécurité*** : car le nombre de points d'entrée permettant l'accès aux données est moins important.
- ***Une administration au niveau serveur*** : les clients ont peu d'importance dans ce modèle, ils ont moins besoin d'être administrés.
- ***Un réseau évolutif*** : grâce à cette architecture, on peut supprimer ou rajouter des clients sans perturber le fonctionnement du réseau et sans modifications majeures.

➢ **Les inconvénients :**

- ***Un coût élevé*** : Dû à la technicité du serveur.
- ***Un maillon faible*** : Le serveur est le seul maillon faible du réseau, étant donné que tout le réseau est construit autour de lui.

I.2. Java Entreprise Edition [6]:

Dans le monde actuel de l'informatique et des technologies de l'information et de la communication, les applications sont de plus en plus complexes, et doivent être réalisées en un minimum de temps et de coût et portable à travers les déférents systèmes utilisés.

Le triptyque de gestion de projets Délais-Moyens-Objectifs est toujours d'actualité et demande aux sociétés informatiques de créer des applications dans un temps donné avec les ressources adaptées dans un souci de productivité. Sun Microsystems, désormais Oracle Corporation, propose pour cela la plate-forme Java Entreprise Edition ou Java EE, véritable référence pour le développement d'applications d'envergure et supportant en standard la programmation, la persistance des données, les transactions, la sécurité ou encore le déploiement de projet sur un serveur applicatif.

I.2.1. Définition :

Java Entreprise Edition est une spécification pour la technique Java de Sun plus particulièrement destinée aux applications d'entreprise. Ces applications sont considérées dans une approche multi-niveaux basées sur des composants.

Dans ce but, toute implémentation de cette spécification contient un ensemble d'extensions au framework Java standard (JSE, Java Standard Edition) afin de faciliter la création d'applications réparties.

I.2.2. Fonctionnement interne :

Le langage Java, sur lequel les librairies Java EE sont utilisées, met à disposition un compilateur et une machine virtuelle (JVM-Java Virtuel Machine) qui se charge de créer un environnement standard pour le lancement de l'application sur tout type de système opérationnel. Le compilateur compile le code source et produit le byte code, soit un code intermédiaire qui sera ensuite lu par la machine virtuelle Java. Chaque système opérationnel majeur possède une JVM expressément codée.

I.2.3. L'architecture d'une application Java EE [7]:

Java EE ajoute des nombreuses couches de niveau entreprise au-dessus de la plate-forme JSE (Java Standard Edition). Chaque couche est conçue pour supporter une différente technologie de développement.

- **Technologie web application :** technologie liée à la production des interfaces web dynamiques, par exemple JSP (Java Server Page) et servlet.

- **Technologie entreprise application :** technologie plus liée à la logique du business : EJP (Entreprise Java Bean), Java Mail, JMS (Java Message Service), JTA (Java Transaction), etc.

- **Technologie web services :** technologie utile au développement des applications adhérentes au paradigme SOA (Service Oriented Architecture), web services,
  JAX-WS (Java API for XML-based web services), JAX-RPC (Java API for XML-based RPC) .

- **Technologie management and security :** technologie liée à la gestion de la technologie d'entreprise afin de réaliser l'accès et l'échange d'information entre machines et services distribués : JAAS (Java Authentification and Authorization Service)**,** JCA (Java Connector Architecture).

Pour bien expliquer l'utilisation de ces technologies on peut imaginer que les technologies entreprise sont utilisées pour gérer l'accès aux données (généralement une ou plusieurs bases de données), les technologies web application sont utilisées pour montrer les données aux utilisateurs génériques. Dans un contexte Business to Business, les technologies web service seront utilisées pour échanger les informations avec les partenaires commerciaux et les technologies de gestion gèrent tous les processus informationnels assurant la sécurité des transactions.

L'architecture d'une application Java EE se découpe idéalement en au moins 3-tiers (trois couches) :

- La couche interface utilisateur ou couche présentation.
- La couche métier.
- La couche d'accès aux données ou couche de persistance.

I.2.3.1. La couche interface utilisateur ou couche présentation :

C'est la couche qui dialogue avec l'utilisateur, via une interface graphique Swing, une interface console ou une interface web.
Cette couche permet par l'intermédiaire de la couche dite « Business» d'envoyer et de recevoir des informations de la couche d'accès aux données. La présentation peut être exploitée sous plusieurs formes : client lourd (application graphique ou textuelle), client léger (navigateur web), applications mobiles.
Pour utiliser ce niveau logique de l'architecture Java EE, plusieurs technologies sont utilisées dont on cite :

- **Les pages JSP (Java Server Page) [8] :**

JSP est une technologie Java qui permet de générer dynamiquement, aussi bien coté client que serveur, du code HTML, XML ou autres.

Une page JSP se présente comme un fichier contenant du code HTML et des fragments de code Java exécutés sur le moteur de servlet comparable aux langages côtés serveur de type PHP, ASP, les pages JSP sont converties en servlet par le moteur de servlet lors du premier appel à la JSP.

- **Java Server Faces (JSF) :**

JSF est un Framework java pour le développement d'application web. Il utilise les JSP et il est basé sur la notion de composant.

I.2.3.2. La couche métier :

C'est la partie de l'architecture 3-tiers qui implémente le métier de l'entreprise. Ce niveau permet le traitement des interactions entre les utilisateurs et le système tout en tenant compte des règles de gestion et de contrôle de l'entreprise.

Cette couche met en œuvre des services métiers et applicatifs. Elle permet de faire un relais entre la couche de présentation et celle d'accès aux données. Dans l'architecture 3-tiers, un niveau ne peut communiquer qu'avec son voisin direct ce qui justifie que cette couche est bien l'intermédiaire des deux autres couches. Dans le « Business tier » la plate-forme Java EE utilise un certain nombre de technologies telles que : EJB (Entreprise JavaBeans), JPA (Java Persistence API), servlet.

- **Entreprise JavaBeans (EJB) [9] :**

Les « EJB » sont des composants métier qui s'exécutent côté serveur. On distingue trois types d'entités : EJB session, EJB entité et EJB message.

- **EJB session :** Cette partie définit un ensemble de service exposé sous forme de fonctions. Les EJB session sont sans état (stateless) ou avec état (statefull) entre deux appels.
- **EJB entité :** Ces EJB permettent de faire le « mapping » des entités d'une base de données relationnelle.
- **EJB message :** Les EJB message donne la possibilité d'échange de messages asynchrone entre objet sous forme de publication (en queue ou topic).

- **Java Persistence API (JPA) :**

  Java Persistence API permet de faire le mapping objet/relation (O/R) donnant ainsi au développeur un niveau d'abstraction plus élevé que celui de JDBC pour l'accès aux informations d'une base de données relationnelles. Cette API est initialement implémentée dans le cadre des EJB 3.0 mais peut être aussi utilisée dans le cadre d'une application Java SE (Java Standard Edition).

  L'API s'appuie sur de simples POJOs annotés et sur un gestionnaire de ces entités pour manipuler (CRUD : créer, lire, mettre à jour et supprimer) les informations d'une base de données relationnelles.

  L'API propose un langage de manipulation de données d'une base relationnelle appelé JPQL pour Java Persistence Query Langage. Ce langage ressemble au langage SQL mais utilise les objets dans les requêtes.

- **Servlets [10] :**

Une servlet est une classe java qui s'exécute côté serveur pour générer dynamiquement des données exploitables par un navigateur web. Ces données sont souvent en format HTML ou XML.

Les servlets ont de nombreux avantages :

- ✓ Elles sont performantes et rapides car elles sont chargées en mémoire dès le premier appel. Elles sont portables et évolutives.
- ✓ Elles sont disponibles car elles s'exécutent dans un contexte multitâche (une seule instance créée).

I.2.3.3. La couche accès aux données ou couche de persistance :

Cette couche regroupe les informations manipulées par la plate-forme. Ces informations peuvent être sous différentes formes telles que des bases de données, des fichiers et tout autre type de stockage de données (texte, XML, Bases De Données, Excel).

I.2.3.4. Le modèle MVC « Modèle-Vue-Contrôleur » :

En anglais design pattern, un modèle de conception (ou encore patron de conception) est une simple bonne pratique, qui répond à un problème de conception d'une application, elle s'applique particulièrement bien à la conception d'applications Java EE

Le modèle MVC permet de séparer la couche interface utilisateur des autres parties du système (car les interfaces utilisateurs sont beaucoup plus susceptibles de changer que la base de connaissances du système), Il découpe littéralement l'application en couches distincte

- tout ce qui concerne le traitement, le stockage et la mise à jour des données de l'application doit être contenu dans la couche nommée "Modèle" (le M de MVC) ;
- tout ce qui concerne l'interaction avec l'utilisateur et la présentation des données (mise en forme, affichage) doit être contenu dans la couche nommée "Vue" (le V de MVC) ;
- tout ce qui concerne le contrôle des actions de l'utilisateur et des données doit être contenu dans la couche nommée "Contrôle" (le C de MVC).

Ce modèle peut être représenté par la figure suivante.

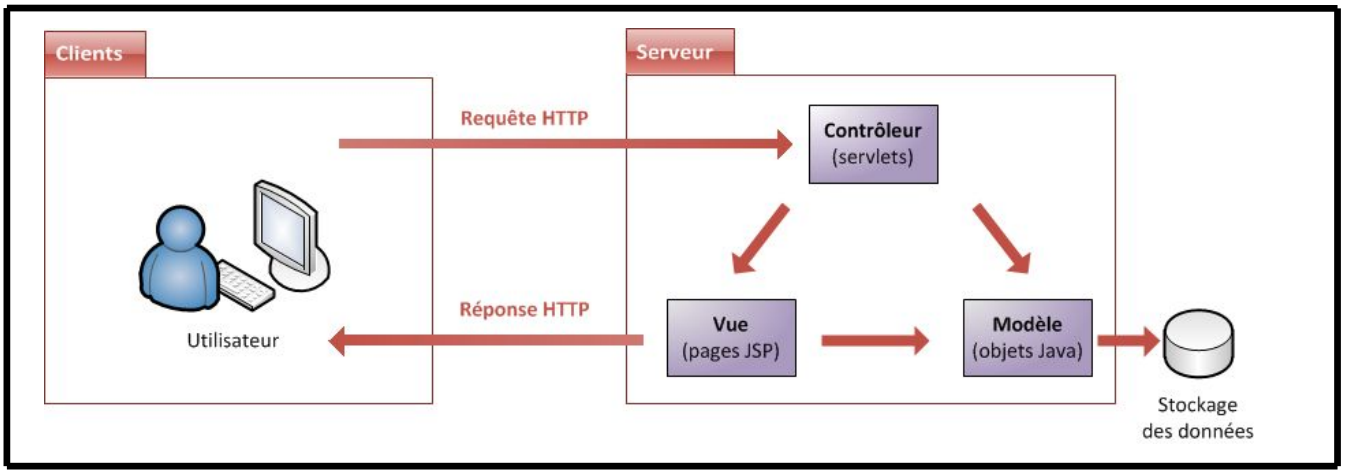


Figure I.5 : MVC avec Java EE.

I.2.4. L'environnement d'exécution des applications Java EE :

Java EE propose des spécifications pour une infrastructure dans laquelle s'exécutent les composants. Ces spécifications décrivent les rôles de chaque élément et précisent un ensemble d'interfaces pour permettre à chacun de ces éléments de communiquer.

Ceci permet de séparer les applications et l'environnement dans lequel elles s'exécutent. Les spécifications précisent à l'aide des API un certain nombre de fonctionnalités que doit implémenter l'environnement d'exécution. Ces fonctionnalités sont de bas niveau ce qui permet aux développeurs de ce concentrer sur la logique métier.

Pour exécuter ces composants de natures différentes, Java EE définit des conteneurs pour chacun d'eux. Il définit pour chaque composant des interfaces qui leur permettront de dialoguer avec les composants lors de leurs exécutions. Les conteneurs permettent aux applications d'accéder aux ressources et aux services en utilisant les API.

Les appels aux composants se font par des clients en passant par les conteneurs, les clients n'accèdent pas directement aux composants mais sollicitent le conteneur pour les utiliser.

I.2.4.1. Les conteneurs :

Les conteneurs assurent la gestion du cycle de vie des composants qui s'exécutent en eux. Les conteneurs fournissent des services qui peuvent être utilisés par les applications lors de leurs exécutions.

Il existe plusieurs conteneurs définit par Java EE :

- **Conteneur web :** pour exécuter les servlets et les JSP.
- **Conteneur d'EJB :** pour exécuter les EJB.
- **Conteneur client :** pour exécuter des applications standalone sur les postes qui utilisent des composants Java EE.

I.2.5. Les avantages de l'utilisation de Java EE :

L'utilisation de Java EE pour développer et exécuter une application offre plusieurs avantages :

- Une architecture d'applications basée sur les composants qui permet un découpage de l'application et donc une séparation des rôles lors du développement.
- La possibilité de s'interfacer avec le système d'information existant grâce à de nombreuses API : JDBC, JNDI, JMS, JCA…
- La possibilité de choisir les outils de développement et le ou les serveurs d'applications utilisés qu'ils soient commerciaux ou libres.

Conclusion :

Au cours de ce chapitre, nous avons présenté la technologie Java EE, nous avons parlé aussi de l'architecteur client /serveur, puis nous avons terminé par une description des architecteurs distribuées Java EE, Ces notions seront utilisées pour le développement de notre projet.
Dans le chapitre suivant nous allons présenter l'organisme d'accueil.

Chapitre II :

Présentation de l’organisme d’accueil

Introduction

L'étude préalable nous permettra d'analyser la situation informatique actuelle de l'entreprise, d'étudier ses lacunes et de proposer des solutions.

Cette étape nous permettra d'avoir une vue générale de l'organisme d'accueil et analyser la situation informatique afin de comprendre comment le parc informatique est exploité, de déceler certains problèmes et de proposer des solutions.

II. Présentation de l'organisme d'accueil ENIEM :

II.1. Situation géographique :

L'entreprise ENIEM (Entreprise Nationale des Industries de Electroménagers) se trouve au sein de la zone industrielle AISSAT-IDIR OUED – AISSI à 10 Km de TIZI-OUZOU, s'étalant une surface totale de 55 Hectares, sa direction générale se trouve au chef-lieu de TIZI-OUZOU à proximité de la gare ferroviaire.

Figure II.1 : Vue satellitaire du site de l'entreprise

II.2. Historique :

E.N.I.E.M (Entreprise National des Industries de l'Electroménager) est une entreprise publique économique constitué le 2 janvier 1983 mais qui existe depuis 1974 sous tutelle de l'entreprise SONELEC.

Leader de l'électroménager en Algérie. Doté d'un capital d'1 milliard et demi de dollars, d'un savoir-faire reconnu et d'une main d'œuvre performante, elle a su se construire une réputation solide dans le domaine du froid, de la cuisson, de la climatisation.

II.3. Champ d'activité de l'ENIEM :

L'ENIEM est le leader de l'électroménager en Algérie. Elle possède une expérience de 40 ans dans le développement et la fabrication d'appareils dans différentes branches de l'électroménager notamment :

- les appareils ménagers domestiques.
- Les appareils de collectivités.
- Les petits appareils ménagers.
- Les produits sanitaires.

L'activité de l'ENIEM sera concentrée sur la fabrication de réfrigérateurs, cuisinières, et climatiseurs. Cette activité sera assurée par plusieurs unités de production :

- **Unité froid :**

  Elle est chargée de la production et le développement des produits froids.

- **Unité cuisson :**

  Chargée de la production et le développement des produits de cuisson, ses activités sont :

- La transformation des tôles
- Assemblage

- **Unité climatisation**

Les capacités existantes sont de 60.000 climatiseurs sous licence AIWELL – France – 1977, dont les modèles sont :

- Climatiseurs type fenêtre – 9000, 12000 et 15000 BTU/h
- Climatiseurs Split système S320 – 11250 BTU/h
- Climatiseurs Split système S530 – 18000 BTU/h
- Climatiseurs Split système S430 – 14950 BTU/h
- Machine à laver 07 Kg
- Chauffe-eau 10 litre à GN et GB

- **Unité commerciale**

Ses activées sont :

- La distribution et l'exploitation des produits ENIEM
- Le service après – vente (à travers ses moyens propres et un réseau d'agent agrées).

- **Unité prestations techniques**

Cette unité assure les fonctions de soutien aux unités de production dans les domaines de :

- Réparation des outils de moules
- Conception et réalisation d'outillages
- Fabrication de pièce de rechange mécanique
- Gestion des énergies et fluides
- Gardiennage et sécurité
- Travaux d'imprimerie

II.4. Missions et objectifs de l'entreprise :

II.4.1. Missions :

La mission de L'ENIEM est d'assurer la production, le montage, la commercialisation, le développement et la recherche dans les différentes branches de l'électroménager notamment :

- Les appareils de cuisson par l'unité cuisson
- Les appareils de climatisation par l'unité climatisation
- Les produits sanitaires par l'unité d'AIN DEFLA

II.4.2. Objectifs :

- L'amélioration de la qualité des produits.
- La maîtrise des coûts de production
- L'augmentation des capacités d'études et de développement.
- L'amélioration de la maintenance de l'outil de production des installations.
- La valorisation des ressources humaines.
- L'augmentation des taux d'intégration (Interne et Externe).
- L'augmentation du volume de production.

II.5. Organisation générale de l'ENIEM :

II.5.1 Organigramme de l'entreprise ENIEM :

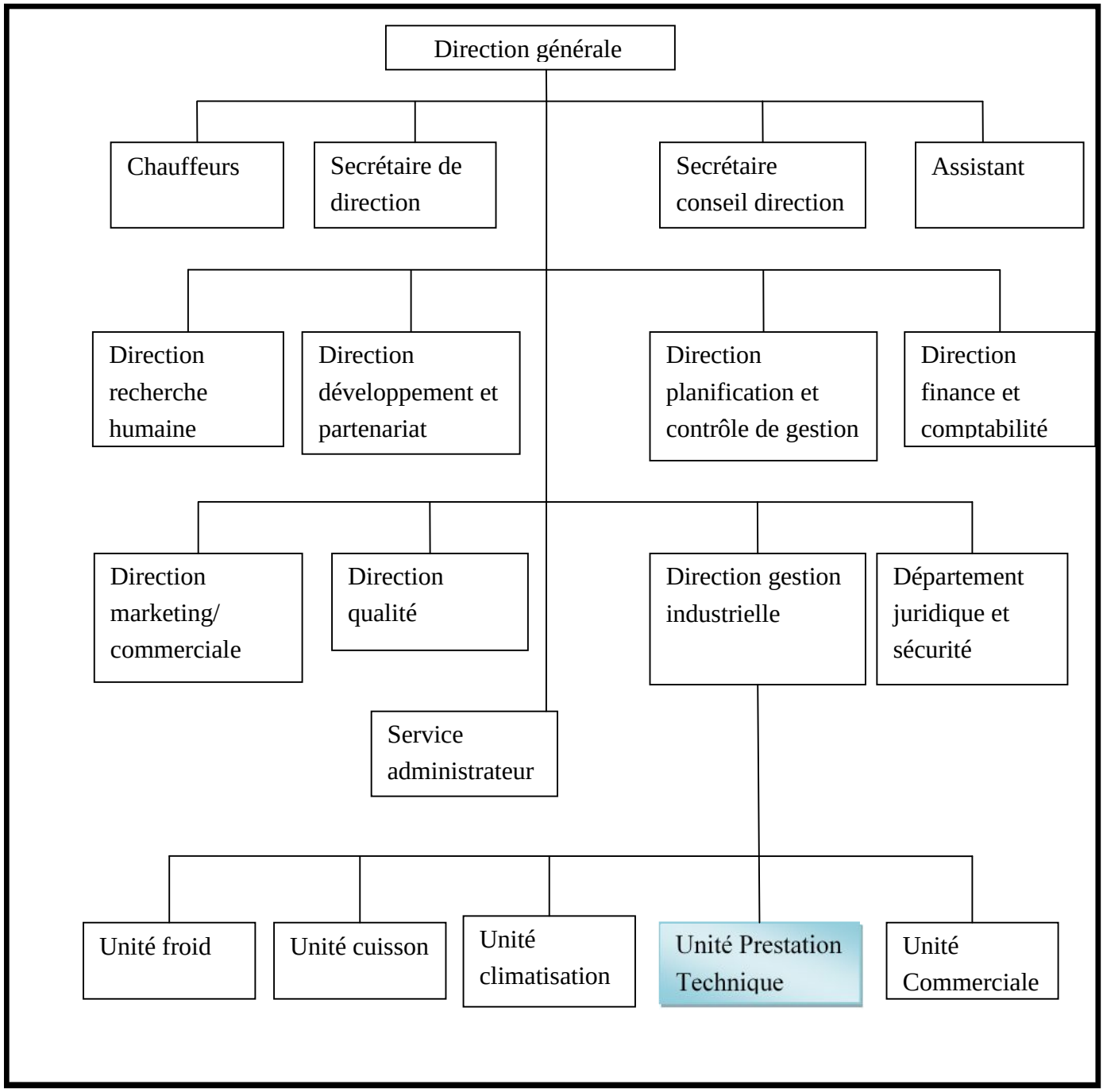


Figure II.2 : Organigramme de l'entreprise ENIEM

II.5.2 Organigramme de l'unité de prestation technique :

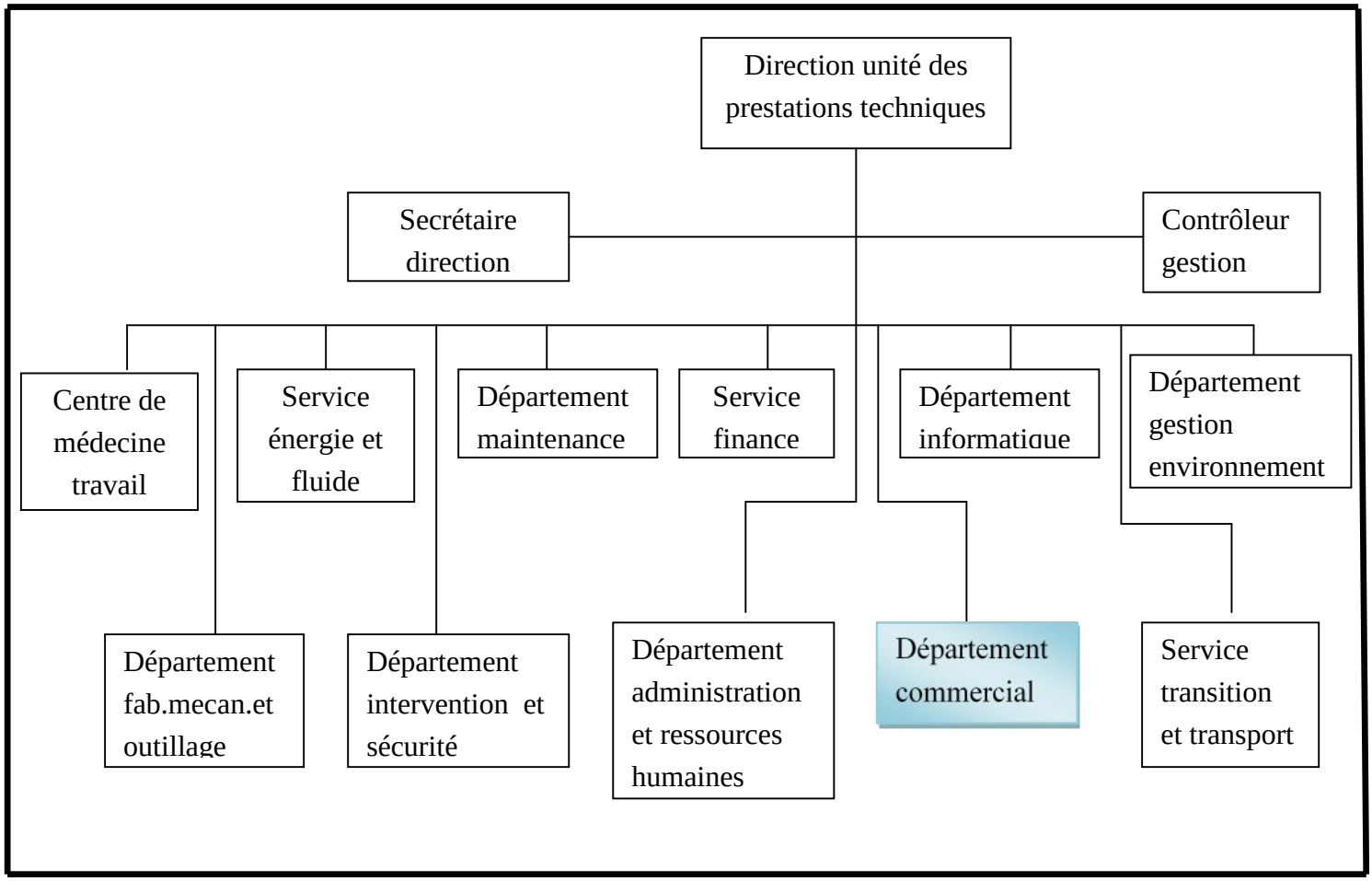


Figure II.3 :Organigramme de l'unité de prestation technique

II.5.3 Organigramme de département commercial :

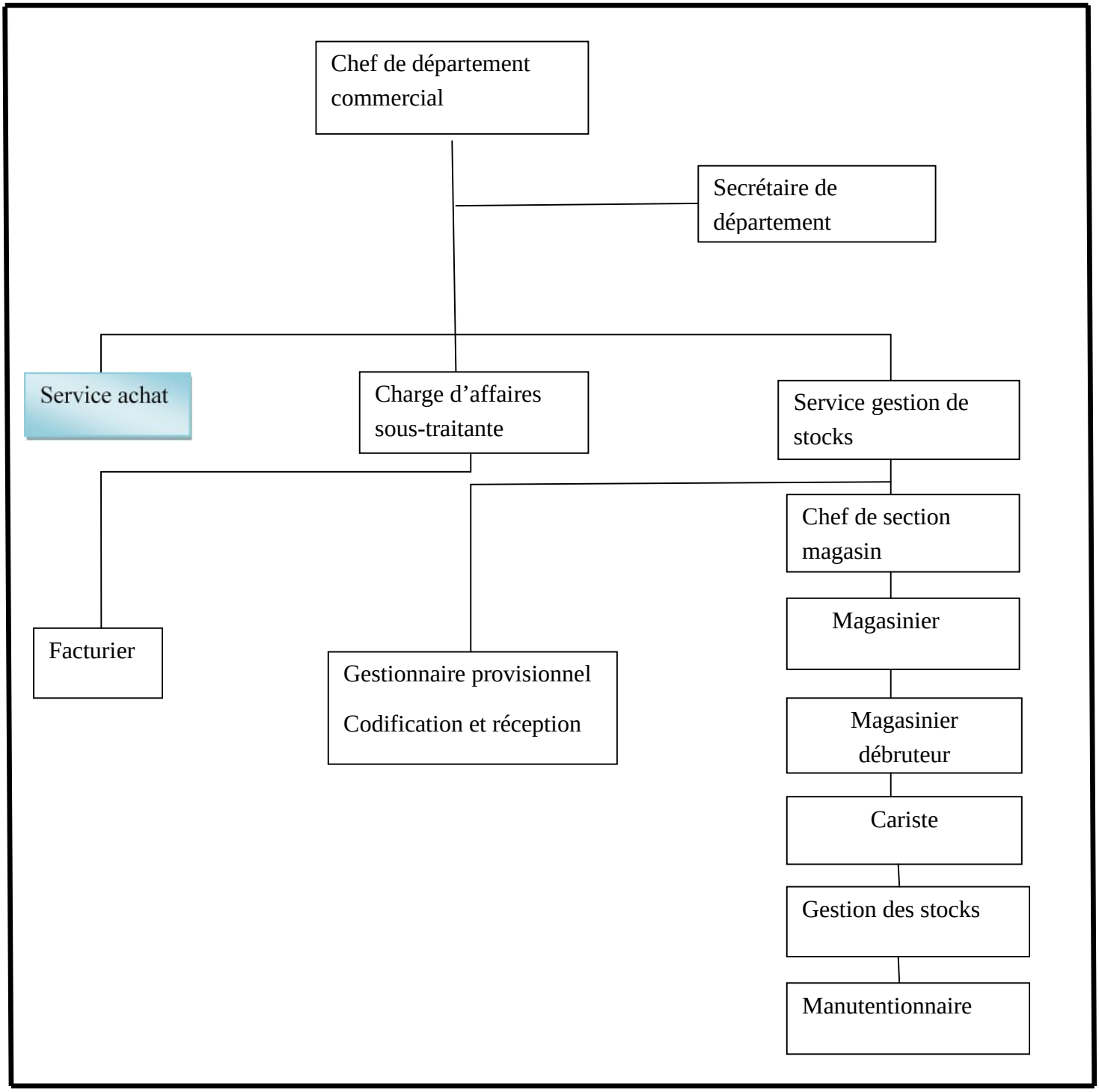


Figure II.4 : Organigramme du département commercial

Champ d'étude : Service achat

II.5.4 Organigramme de service d'achat :

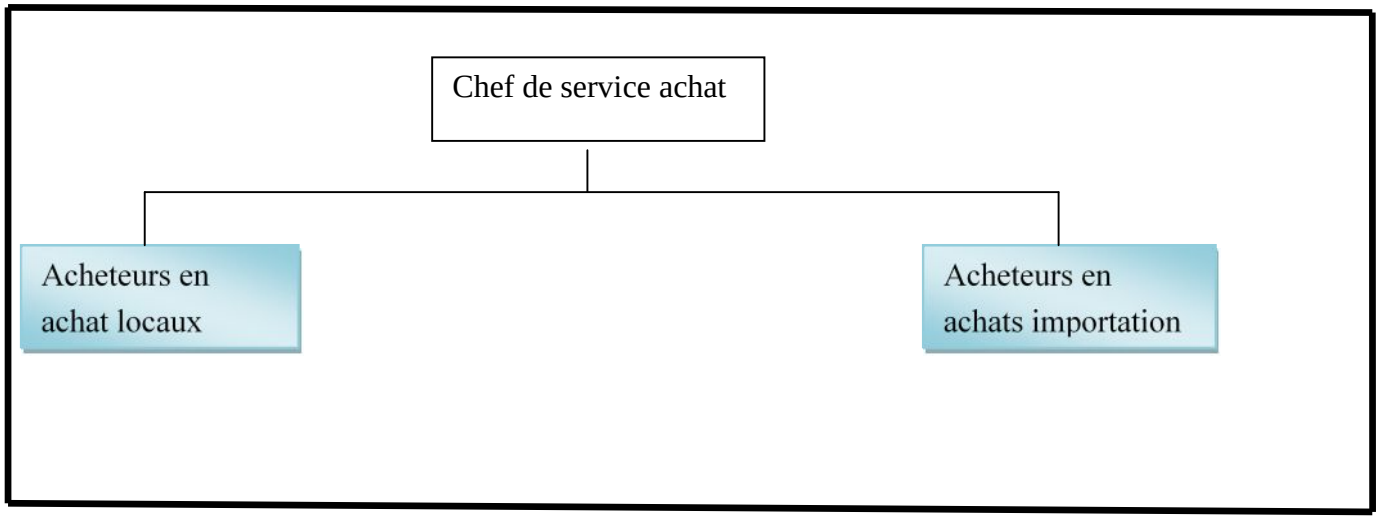


Figure II. 5 : Organigramme de service d'achat

Champ d'étude :

II.6. Situation informatique :

Au début des années 80, ENIEM, a commencé à s'organiser dans le but d'informatiser certains services, jugés prioritaires telles les ressources humaines et la comptabilité ; en optant d'abord à un recrutement du personnel qui aura la charge d'asseoir un projet informatique.

Vers 1985, les prémices d'une solution intégrant la comptabilité et la paie a vu le jour et sera finalisé vers 1986 par l'achat d'un mini-ordinateur Hp3000 modèle 52 dont la configuration était de 1 MO de mémoire centrale et de 800 MO de disque. Quatorze terminaux et quatre imprimantes desservant les structures de gestion comptabilité et paie forment le réseau informatique de l'époque.

Vers 1990, après une organisation tournée vers la gestion de production assistée par ordinateur (GPAO) et un système « achats », l'ENIEM s'est équipée d'un nouveau mini-ordinateur Hp3000/70 modèle 70 dont la configuration est de 8 MO de mémoire RAM et de 4 disques de 571 MO, muni d'une solution réseau LAN intégrant une interface réseau vers le

Hp3000/52 déjà en place. Si ajoutent 96 terminaux et 14 imprimantes installés soit aux ateliers aux fins de gestions de production et de stocks, soit dans les structures de gestion des achats et autres transits.

Vers 1997, l'obsolescence de la solution informatique commence à se ressentir sérieusement. En effet, le constructeur Hewlett - Packard ayant délaissé les systèmes sous OS MPE (Multi–Programming - Equipment), générant ainsi une maintenance trop cher payée, et un suivi sous label Remarkete (vente de cartes électroniques sous maintenance). De plus, le problème du passage à l'an 2000 et les très fortes indisponibilités de la pièce de rechange auront fortement agi sur la décision d'acquérir de nouvelles machines capables à la fois de la prise en charge de l'existant et de permettre une ouverture vers les systèmes dits ouverts.

- Solution préconisée et mise en œuvre actuelle

Plusieurs axes sont imaginés. Le schéma directeur final sera matérialisé par une solution de continuité, à savoir :

- Une machine Hp3000 série A500 pouvant immédiatement prendre en charge l'existant tout en ayant la possibilité de devenir Serveur Hp9000, moyennant un kit agissant sur le hardware et le software ;
- Un serveur Hp9000 servant de plateforme de base pour une solution future.

Les deux systèmes, une fois l'intégration du passif terminée, formeront un couple de serveurs, qui, grâce au memoring (sauvegardes à chaud durant les périodes de travail), permettra à ENIEM de bénéficier de la sécurité de ses données.

Le nombre d'utilisateurs, actuellement porté à 90 ports de terminaux départagés entre les ateliers et l'administration, se verra augmenté suffisamment pour arriver à connecter le complexe dans un cadre de réseau Internet en utilisant toutes les solutions et moyens adéquats.

Grâce à la solution préconisée, ENIEM gagnera à acquérir toutes les solutions existantes sur le marché et pouvant améliorer ses modes de gestion.

Conclusion

Dans ce chapitre nous avons présenté l'organisme d'accueil qui nous a permis de mettre en évidence notre domaine d'étude.

Dans le chapitre suivant nous allons étudier le domaine d'étude et essayer d'améliorer son fonctionnement et par la suite présenter l'analyse et la conception.

Chapitre III :

Analyse et conception

Introduction

Tout analyse doit commencer par l'étude de l'existant pour avoir une bonne connaissance du système existant qui nous permet d'éclairer ses avantages ainsi que ses inconvénients, ce qui permettra de chercher les meilleurs solutions à travers notre étude.

III.1. Etude de l'existant :

III.1.1. Quelques définitions :

- **Demande d'achat interne (DAI) :** C'est une demande d'achat informatisée, établie par le service gestion des stocks (magasin) à la suite d'une naissance d'un besoin. Elle est transmise au service achats après avoir été approuvée et signée par :

✓ **Chef de section magasin**

✓ **Chef de service magasin**

✓ **Chef de département commerciale**

✓ **Directeur de l'unité.**

- **Demande d'offre(DO) :** Document contenant la liste des produits que l'entreprise souhaite acheter (d'après la DAI) cette demande est envoyée aux fournisseurs.
- **Les offres (Facture pro forma abrégé FPF) :** Représente les offres des fournisseurs suite à la réception d'une demande d'offre.
- **Le tableau comparatif des offres(TCO) :** Document établi par les acheteurs après réception des différentes offres des fournisseurs dans le but de sélectionner le meilleur fournisseur après comparaison des offres.
- **Une commande :** Exprimée par un bon de commande (BC) contenant la liste des produits, les quantités et caractéristique voulues.
- **Fiche de vérification (FV):** C'est un document contenant les informations du fournisseur retenu.
- **Facture :** Document commerciale justifiant l'opération d'achat, elle est établie par le fournisseur à destination du client.
- **Bon de livraison (BL):** Document établi par le fournisseur contenant la liste des produits livrés.

- **Bon de réception(BR) :** Document établi par le service magasin contenant la liste des produits reçus.

III.1.2. Etudes des postes de travail :

Cette étude a pour objectif de recenser les différents postes de travail entrant dans la gestion des achats locaux et imports de l'ENIEM, cela tout en précisant les tâches assurées par chaque poste et les différents documents manipulées (informations, entrées, sorties, …etc.).

Le service des achats appartenant à l'UPT (L'unité de prestation technique) représente le maillon principal dans la chaîne des achats locaux et imports, ce qui mène à nous concentrer sur ce service lors de l'étude de l'existant.

Dans notre étude il existe quatre (04) postes de travail qui sont :

1. Chef de département commercial (administrateur)

2. Acheteurs

3. Chef de service stock

4. Chef de service transit et douane

Fiche d'analyse poste N°01
Désignation : Chef de département commercial **Rattachement :** Unité prestation technique **Effectif : 01** **Responsabilité de poste :** Il est responsable du département commerciale, regroupant les activités d'achats, relations client, gestion des stocks magasin
Tâches et responsabilités
-Coordonner et superviser les activités d'achats -Assurer le suivi de l'évaluation des prix et détecter les nouvelles conditions d'achats et les nouvelles sources d'approvisionnement -Négocier et suivre les commandes et contrats importants - Approuver les différents documents commandés -Donner l'autorisation de paiement des factures.

Tableau III.1 : Fiche d'analyse poste N°01

Fiche d'analyse poste N°02
Désignation : Acheteur **Effectif : 01** **Responsabilité de poste :** -Assure l'achat de matières premières et auxiliaires, composantes, pièces de rechange et outillage. -Gestion des dossiers d'achats.
Tâches et responsabilités
-Prospecter le marché (nouvelles sources d'approvisionnement). -Choisir les fournisseurs à consulter. -Lancer les consultations dès la réception d'une DAI. -Lancer la procédure d'achat et établir les commandes. -Elaborer le tableau comparatif des offres. -Suivre les dossiers d'achats conformément à la procédure du service. -établir des documents nécessaires au service de gestion des stocks. -Suivre les livraisons. -Sorties en missions.

Tableau III.2 : Fiche d'analyse poste N°02

Fiche d'analyse poste N°02
Désignation : Chef de service Stock **Effectif : 01** **Responsabilité de poste :** Gère les entrées et les sorties journalière des produits finis
Tâches et responsabilités
-Etablir et vérifier le bon de chargement -Assurer la livraison et la distribution dans les délais exigés. -Etablir et vérifier le bon de réception. -Etablir le bulletin de livraison interne. -Assurer et vérifier l'entrée et la sortie des produits.

Tableau III.3 : Fiche d'analyse poste N°03

Fiche d'analyse poste N°02
Désignation : Chef de service transit et douane **Effectif : 01** **Responsabilité de poste :** Veille au respect des formalités relatives au dédouanement des marchandises importées par le complexe.
Tâches et responsabilités
-veiller à ce que le dossier réceptionné soit complet pour procéder au dédouanement suivant la réglementation douanière. -procéder à la localisation et à la reconnaissance des marchandises. -ordonner le dédouanement suivant les urgences signalées. -veiller à ce que la position tarifaire soit correcte.

Tableau III.4 : Fiche d'analyse poste N°04

III.1.3. Etude des documents :

Il est nécessaire de connaitre les différents documents utilisés dans les systèmes.

Dans le cadre de notre étude, nous avons recensé une liste de dix (10) documents.

N° du document	Code du document	Désignation du document
01	DAI	Demande d'achat interne
02	DO	Demande d'offre
03	FPF	Facture pro format
04	TCO	Tableau comparatif des offres
05	BC	Bon de commande
06	FV	Fiche de vérification
07	F	Facture
08	BL	Bon de livraison
09	BR	Bon de réception
10	/	Demande de chèque

Tableau III.5 : Liste des documents**.**

III.1.4. Diagramme de flux :

Le diagramme de flux est le moyen d'expression qui permet la présentation graphique d'un ensemble des flux d'informations échangés entre les différents acteurs du domaine d'étude ainsi qu'avec l'environnement.

III.1.4.1. Concepts du diagramme:

L'analyse des flux s'exprime avec deux concepts qui sont :

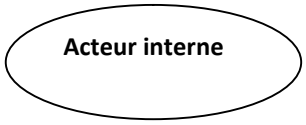
- **Acteur:**

Un acteur est un émetteur ou un récepteur d'un flux d'information lié à une activité au sein du système d'information d'une organisation. Selon le cas, il peut s'agir d'une catégorie de personne, d'un service ou du système d'information d'une autre organisation. Un acteur reçoit un flux d'information, qui lui permet d'agir en transformant l'information et en renvoyant un ou plusieurs autres flux d'information à d'autres acteurs. Les acteurs sont représentés par leurs rôles dans l'activité étudiée.
On distingue :

Les acteurs internes: qui font partie du domaine d'étude.
Les acteurs externes: qui ne font pas partie pas du domaine mais qui on des échanges avec les acteurs internes dans le cadre de l'activité étudiée.

Dans la notation que nous retiendrons, un acteur externe est représenté par un cercle tracé en pointillé :

Acteur interne est représenté par un cercle au tracé continu. Le nom de l'acteur est placé à l'intérieur du cercle :



- **Flux:**

Représente un échange de données entre deux acteurs, il peut être interne ou bien externe.

Flux interne: il décrit l'échange d'informations entre deux acteurs interne dans le champ d'étude.

Flux externe: il décrit l'échange d'informations entre un acteur interne et externe dans le champ d'étude.

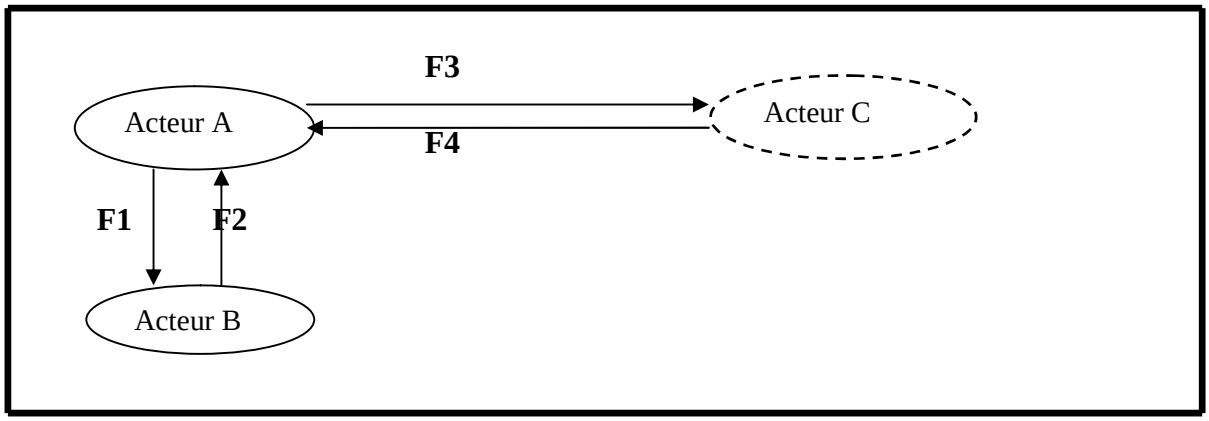


Figure III.1 : représentation des flux d'informations.

Acteur A, Acteur B : sont des acteurs internes.

Acteur C : C'est un acteur externe.

F1, F2 : sont des Flux interne.

F3, F4 : sont des Flux externes.

III.1.4.2. Présentation du diagramme de flux :

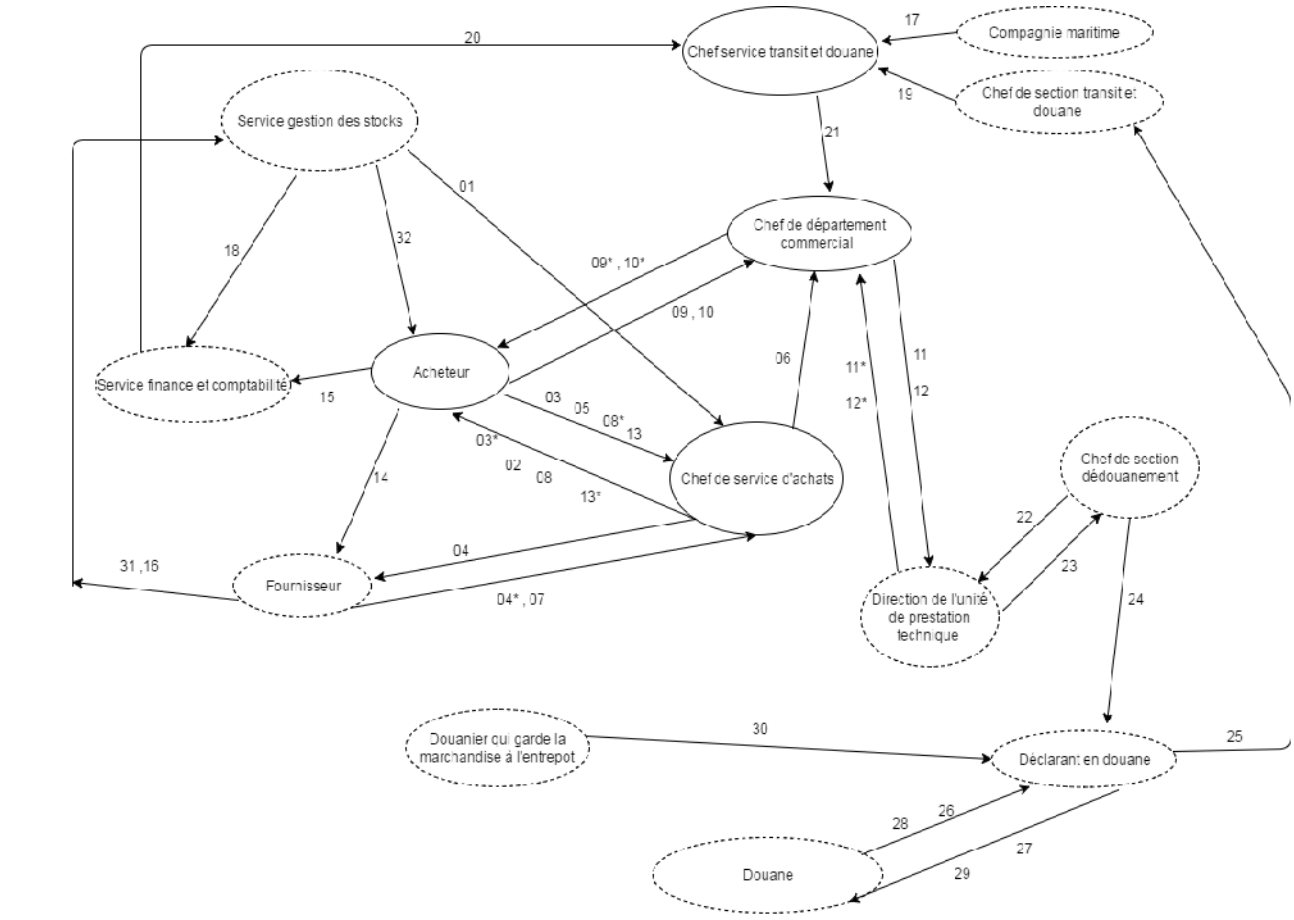


Figure III.2 : diagramme de flux.

III.1.4.3. Description des flux d'information :

N° Flux	Descriptif des flux d'information
01	DAI
02	Liste des fournisseurs
03	Demande d'offre +liste des fournisseurs à consulter
03*	Demande d'offre signée
04	Demande d'offre
04*	Facture pro format
05	TCO
06	TCO+rapport de présentation
07	Facture pro format après négociation
08	Fiche de vérification (montant<8MDA)
08*	Fiche de vérification signée
09	Bon de commande (montant<8MDA)
09*	Bon de commande signée
10	Fiche de vérification (montant>=8MDA)
10*	Fiche de vérification signée
11	Bon de commande
11*	Bon de commande signée
12	Fiche de vérification
12*	Fiche de vérification signée
13	Dossier (DAI, demande d'offre, pro format, TCO, fiche de vérification)
13*	Dossier validé

14	Bon de commande
15	Bon de commande
16	Bon de commande
17	Avis d'arriver de la marchandise
18	Demande de chèque
19	Ordre de mission
20	Chèques
21	Ordre de mission
22	Bon de transport
23	Bon de transport+bon de réception
24	Dossier (bon de transport, bon de réception, chèques)
25	Bon à délivrer
26	Un exemplaire D11
27	Dossier (D11, D3, connaissement, avis d'arriver)
28	Montant à payer
29	Chèques
30	Arrivée des marchandises
31	Bulletin de réception

Tableau III.6 : Description des flux d'information.

III.1.5. Critiques de l'existant et suggestions:

L'étude de l'existant nous a conduit à noter quelques insuffisances dans le système actuel de L' ENIEM causés principalement par le manque de communication et de circulation de l'information entre les différents acteurs.

En effet, nous les avons classés en trois catégories:

- Insuffisances d'ordre organisationnel : elles concernent généralement les postes de travail, classement et archivage de documents.
- Insuffisances d'ordre informationnel: elles concernent généralement les rubriques des documents utilisés par les services du département d'achats de L'ENIEM.
- Insuffisances d'ordre technique: elles concernent en général les outils et le matériel informatique utilisé par les services du département d'achat de L'ENIEM.

III.1.5.1. Niveau organisationnel :

- **Problème 1 :** Le volume élevé des archives et sa non indexation ne permet pas une recherche optimale des documents.
- **Cause** : Absence d'une bonne structure d'archivage.
- **Suggestion :** Sauvegarder les informations sur des supports numériques pour permettre une recherche rapide des informations.

- **Problème 2 :** Chaque acheteur a sa propre méthode de travail.
- **Exemple :** Le classement des documents dans le dossier d'achat.
- **Cause** : Les acheteurs ne sont pas soumis à une méthode fixe de travail. Chacun des acheteurs est satisfait pour sa propre méthode de travail.
- **Suggestion :** Etablissement d'un plan unique et optimal pour spécifier la méthode de classement des documents, et pour toute méthode de travail.

III.1.5.2. Niveau informationnel:

- **Problème 1 :** Redondance de certaines informations sur le même document
- **Cause :** Manque de généralisation des rubriques des documents.
- **Suggestion :** Généraliser la rubrique et supprimer les redondances

- **Problème 2 :**Non remplissage de certaines rubriques des documents
- **Exemple :**
  - N° de demande d'offre sur le bon de commande.
  - N° de l'article sur le TCO.
- **Cause** :
  - Mal conception des documents
  - Absence de mise à jour de la structure du document en réponse au changement de procédure.
  - Utilisation d'un même document pour plusieurs fonctions (achats locaux, imports).
- **Suggestion :** Remplir les rubriques utiles et éliminer les autres jugés inutiles.

III.1.5.3. Niveau technique :

Après l'étude de l'existant nous avons pu déceler quelques anomalies au niveau de la solution informatique utilisée.

- **Problème 1 :** Lenteur dans le traitement de certaines informations effectuées manuellement.
- **Suggestion :** Automatiser les tâches concernées.

- **Problème 2 :** L'obsolète application utilisée ne permet pas une prise en main facile de l'outil, son manque d'ergonomie : utilisation uniquement au clavier cause des difficultés de manipulation qui ont un impact direct sur la productivité de l'acheteur.
- **Suggestion :** Refonte totale de l'application avec les technologies modernes afin d'offrir une interface plus conviviale et une manipulation plus souple.

III.1.5.4. Solution informatique :

La solution proposée est une solution réseau, qui consiste à mettre en place une application 3-tiers pour la gestion d'achat locaux et importation en matières premières offrant aux différents acteurs du service des interfaces conviviales et simples à utiliser en s'appuyant sur une base de données centralisés. Voici l'architecteur globale de cette solution.

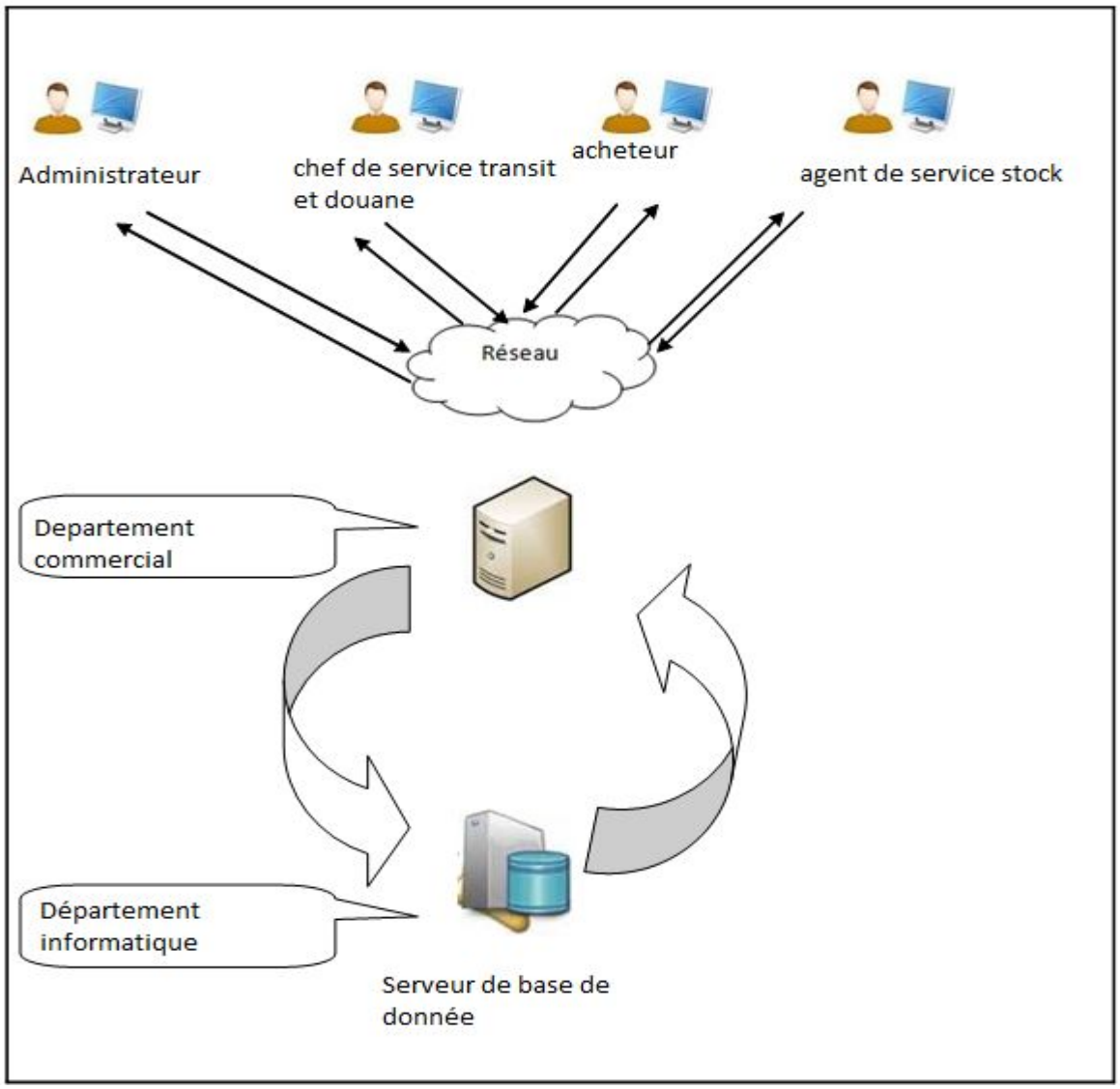


Figure III.3 : Architecteur globale du modèle proposée

III.2. Analyse et conception :

Afin d'aboutir à une meilleure organisation, nous devons tout naturellement avoir recours à un formalisme de conception orientée objet (RUP, Rational Unified Process) qui est une démarche de développement souvent utilisée conjointement au langage UML qui va nous permettre de comprendre et de décrire les besoins, de spécifier et documenter le système ainsi que d'esquisser les architectures logicielles.

Pour le développement de notre application de gestion des achats locaux et importation cas: ENIEM nous suivons les phases suivantes :

Initialisation : Définition du problème.
Elaboration : Planification des activités, affectation des ressources, analyse.
Construction : Développement du logiciel par incréments successifs.
Transition : déploiement.

III.2.1. Présentation des différentes phases :

III.2.1.1. Phase d'Initialisation :

Une analyse approximative de notre projet permet d'élaborer les points suivants :

- Notre application touche à la gestion des achats locaux et imports qui est un domaine connu.

- Notre application est de type Client/serveur, ce genre d'applications a connu déjà un succès considérable et ces derniers se ressemblent d'un point de vu outil.

- Notre application est isolée à internet, le risque de sécurité ne présente pas un énorme problème.

III.2.1.2. Phase d'Elaboration :

Notre projet porte sur la mise en place d'une application client/serveur 3-tiers, pour l'automatisation des activités de la gestion des achats locaux et importation.
Dans le but de maitriser le suivi des mouvements des documents et dossiers entre les différents services du champ d'étude, l'application doit offrir un environnement interactif et convivial aux différents acteurs de tous les services de la direction de l'unité de prestation technique leur permettant de travailler et de communiquer mutuellement, dans l'objectif de la gestion, et ce via un réseau local (intranet).

III.2.1.3. Phase de Construction :

Pour chaque itération réalisée précédemment, nous allons suivre un cycle de vie en cascade pour aboutir à un diagramme de classe en passant par un diagramme de séquence ou de collaboration.

III.2.1.4. Phase de Transition :

Exécution des bêtas tests, déploiement et test de réception.

III.2.2. Identification des acteurs :

Acteur : Un acteur est un utilisateur type qui a toujours le même comportement vis-à-vis d'un cas d'utilisation.

Durant la période du stage au sein de l'ENIEM, nous avons procédé en premier lieu à la localisation des centres d'activité de la gestion et le suivi des achats de la matière première.
En second lieu on a identifié les principaux acteurs qui seront les futurs utilisateurs de l'application interpréter par :

- **Administrateur (ADMIN) :** gère principalement les comptes utilisateurs et entretient le système (dans notre cas c'est le chef du département commercial).

- **Agent de service stock (ASS) :** il s'occupe de la gestion des stocks, des magasins, et encore des emplacements dans ces magasins.

- **Acheteur (A) :** est chargé de la mise en place des contrats commerciaux et de l'élaboration des conventions avec les fournisseurs.

- **Chef de service transit et douanes(CSTD) :** veille au respect des formalités relatives au dédouanement des marchandises importées par le complexe.

III.2.2.1. Diagramme de contexte :

Le diagramme de contexte est un modèle conceptuel de flux qui permet d'avoir une vision globale des interactions entre le système et les liens avec l'environnement extérieur. Il permet aussi de bien d'éliminer le champ d'étude. Pour notre cas diagramme de contextes est donné par la figure suivante:

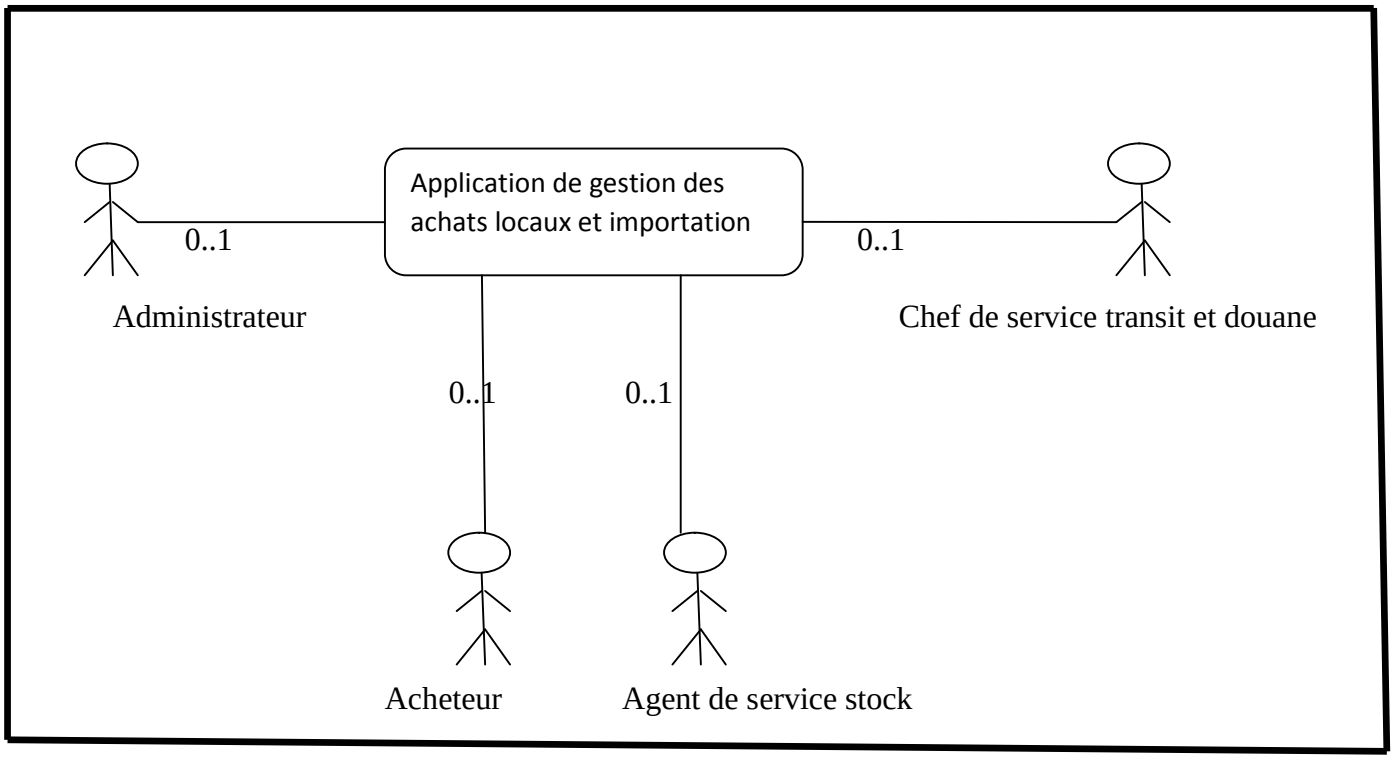


Figure III.4: Diagramme de contexte

III.2.2.2. Représentation des diagrammes de cas d'utilisation [11] :

Le Diagramme de cas d'utilisation est le premier diagramme du modèle UML utilisé pour la modélisation des besoins des utilisateurs.

Les cas d'utilisations décrivent le comportement du système étudié du point de vue de l'utilisateur, et décrivent les possibilités d'interactions fonctionnelles entre le système et les acteurs, ils permettent de définir les limites et les relations entre le système et son environnement.

Il est destiné à structurer les besoins des utilisateurs et les objectifs par rapport au système**.**

- **Représentation du diagramme de cas d'utilisation globale**

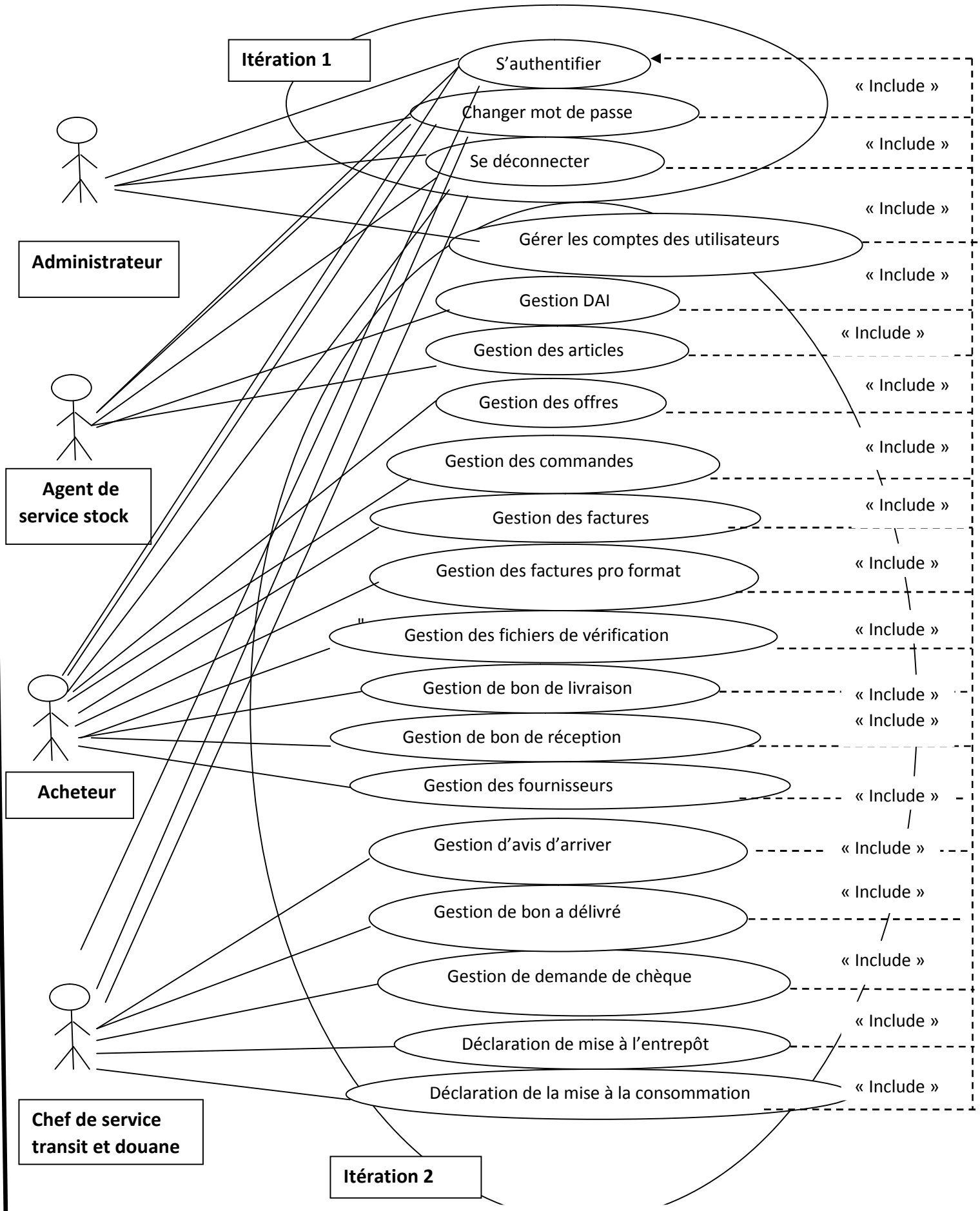


Figure III.5 : Diagramme de cas d'utilisation.

III.2.2.3. Les cas d'utilisations :

Cas d'utilisation relatifs au chef département commercial (administrateur(ADMIN)) :

- S'authentifie.
- Gérer les comptes utilisateurs :
  - Ajouter un utilisateur.
  - Consulter un utilisateur.
  - Modifier un utilisateur.
- Modifier password.

Cas d'utilisation relatif à l'agent de service stock (ASS) :

- S'authentifier.
- Gestion des DAI.
  - Ajouter DAI.
  - Consulter DAI.
  - Modifier DAI.
  - Supprimer DAI.

- Gestion des articles.
  - Ajouter article.
  - Consulter article.
  - Modifier article.
  - Supprimer article.

- Changer password.

Cas d'utilisation relatif à l'Acheteur (A) :

- S'authentifie.
- Gestion des offres.
  - Ajouter offre.
  - Consulter offre.
  - Modifier offre.
  - Supprimer offre.

- Gestion des commandes.
  - Ajouter commande.
  - Consulter commande.
  - Modifier commande.
  - Supprimer commande.

- Gestion de facturation d'achats.
  - Ajouter facture.
  - Consulter facture.
  - Modifier facture.
  - Supprimer facture.

- Gestion de facture pro format.
  - Ajouter facture pro format.
  - Consulter facture pro format.
  - Modifier facture pro format.
  - Supprimer facture pro format.

- Gestion des fiches de vérification.
  - Ajouter fiche de vérification.
  - Consulter fiche de vérification.
  - Modifier fiche de vérification.
  - Supprimer fiche de vérification.

- Gestion du bon de livraison.

- Ajouter un bon de livraison.
- Consulter un bon de livraison.
- Modifier un bon de livraison.
- Supprimer un bon de livraison.

- Gestion fournisseurs.
  - Ajouter fournisseur.
  - Consulter fournisseur.
  - Supprimer fournisseur.
  - Modifier un fournisseur.

- Gestion de bon de réception.
  - Ajouter un bon de réception.
  - Consulter un bon de réception .
  - Modifier un bon de réception.
  - Supprimer un bon de réception.

- Tableau comparatif des offres (TCO).
  - Ajouter un TCO.
  - Consulter un TCO.
  - Modifier un TCO.
  - Supprimer un TCO.

- Changer password.

Chef de service transit et douane :

- S'authentifie.
- Gestion de bon à délivre.
  - Ajouter un bon à délivre.
  - Consulter un bon à délivre.
  - Modifier un bon à délivre.
  - Supprimer un bon à délivre.

- Gestion d'avis d'arriver.
  - Ajouter un avis d'arriver.
  - Consulter un avis d'arriver.
  - Modifier un avis d'arriver.
  - Supprimer un avis d'arriver.

- Gestion de demande de chèque.
  - Ajouter une demande de chèque
  - Consulter une demande de chèque.
  - Modifier une demande de chèque.
  - Supprimer une demande de chèque.

- Déclaration de mise à la l'entrepôt (D11).
  - Ajouter un d11.
  - Consulter un d11.
  - Modifier un d11.
  - Supprimer un d11.

- Déclaration de mise à la consommation (D3).
  - Ajouter un d3.
  - Consulter un d3.
  - Modifier un d3.
  - Supprimer un d3.

- Changer password.

III.2.3. Définition des itérations :

Le critère de définition des itérations se base sur les deux points suivants :

1. Ressemblance entre cas d'utilisation
2. Dépendance entre cas d'utilisation

III.2.3.1. Réalisation des itérations:

- **Diagramme de séquence[12] :**

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.

Voici quelques notions de base du diagramme **:**

-**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 de l'information avec l'intention de déclencher une réaction chez le récepteur.

- **Exemple de diagramme de séquence pour l'itération 1 :**

Diagramme de séquence du cas d'utilisation : «Authentification»

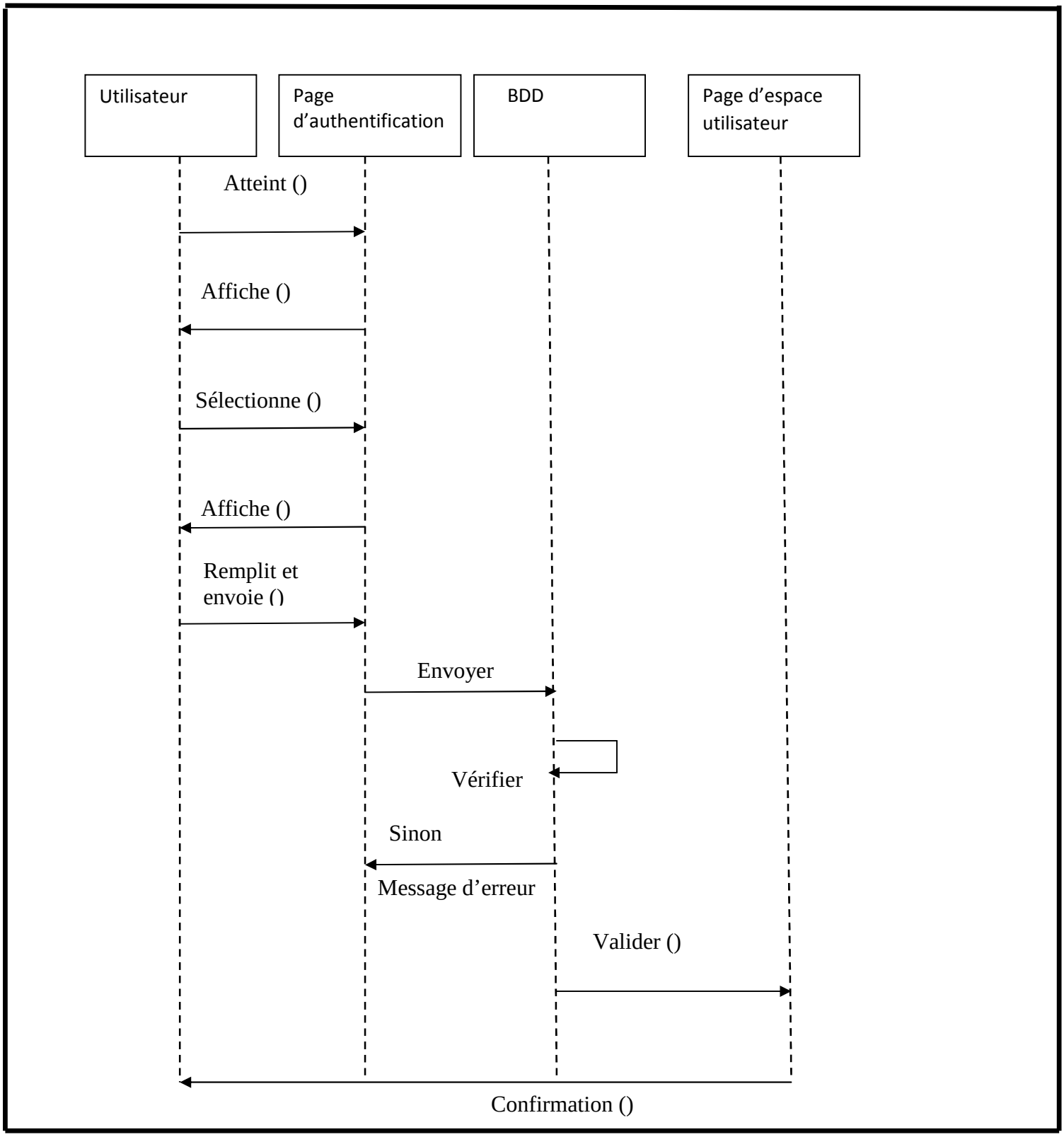


Figure III.6 : Diagramme de séquence du cas d'utilisation : «Authentification»

- **Diagramme de séquence du cas d'utilisation :** «Changer password»

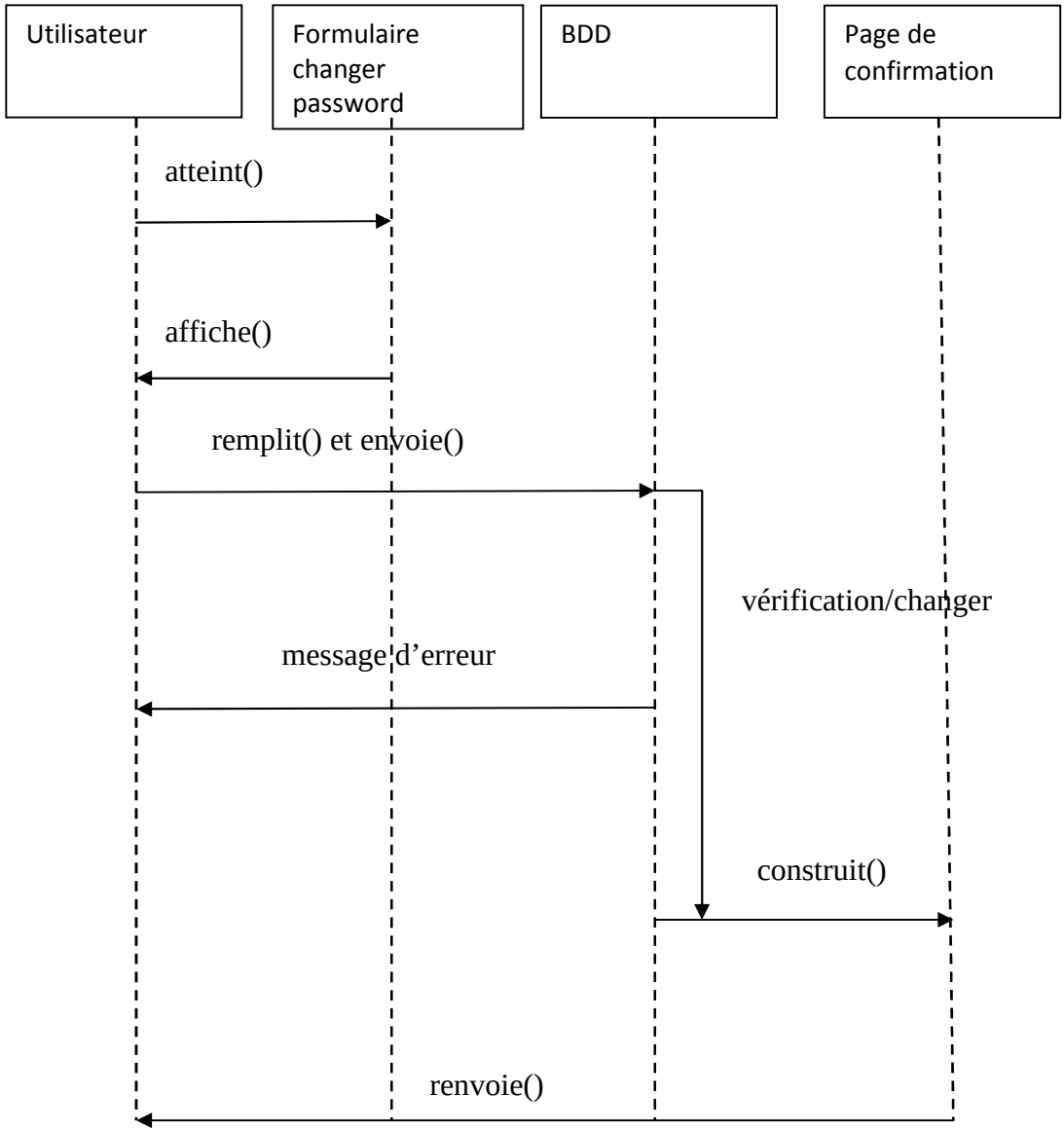


Figure III.7 : Diagramme de séquence du cas d'utilisation : «changer password»

- **Diagrammes de classe pour l'itération 1 :**

Utilisateur
Id_utilisateur Login password
Ajouter () Supprimer () Modifier () Editer ()

- **Exemple de diagramme de séquence pour l'itération 2 :**

Diagramme de séquence du cas d'utilisation : « supprimer une commande»

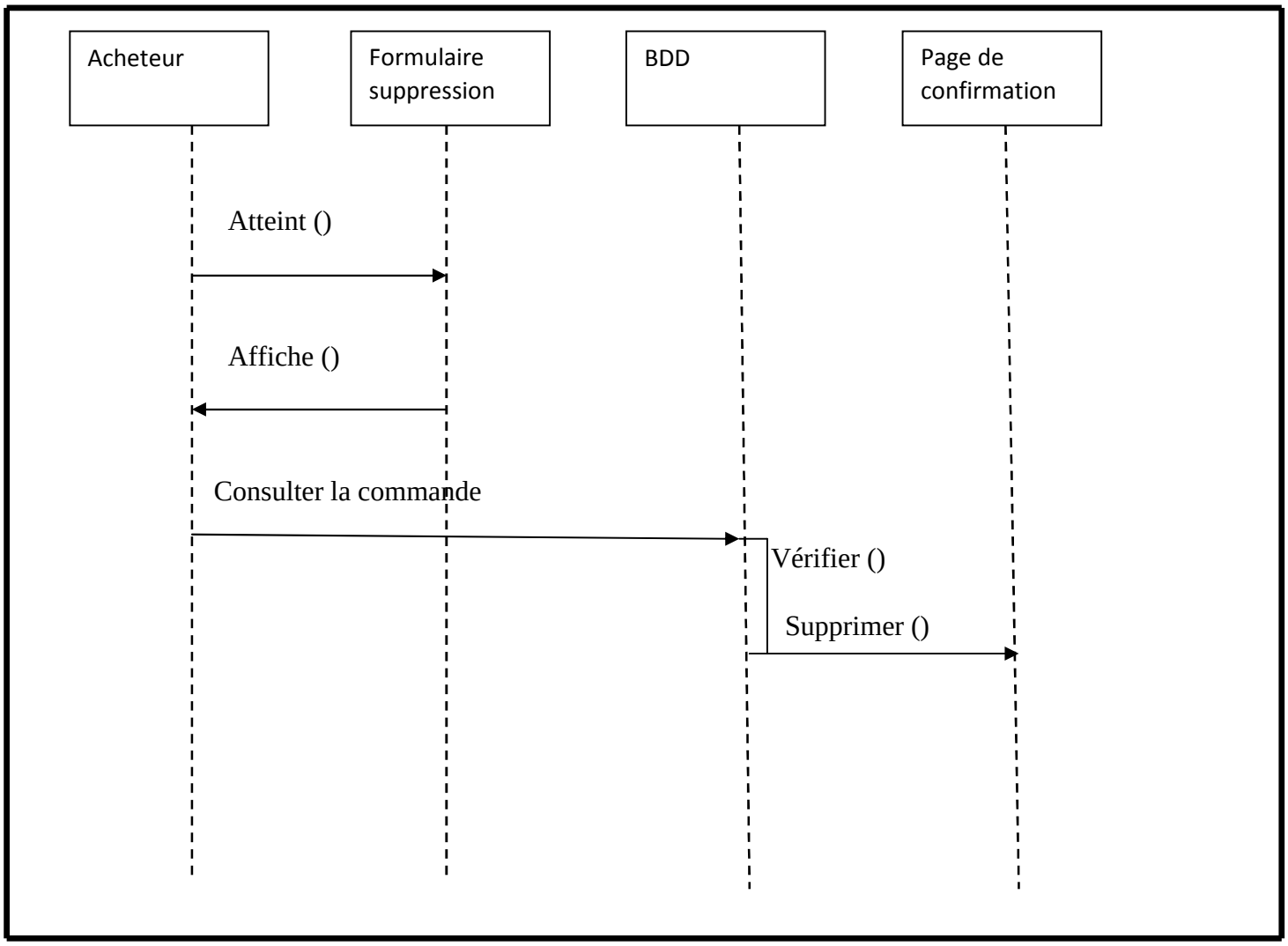


Figure III.8 : Diagramme de séquence du cas d'utilisation **: « supprimer une commande»**

- **Diagramme de séquence du cas d'utilisation : « modifier une facture»**

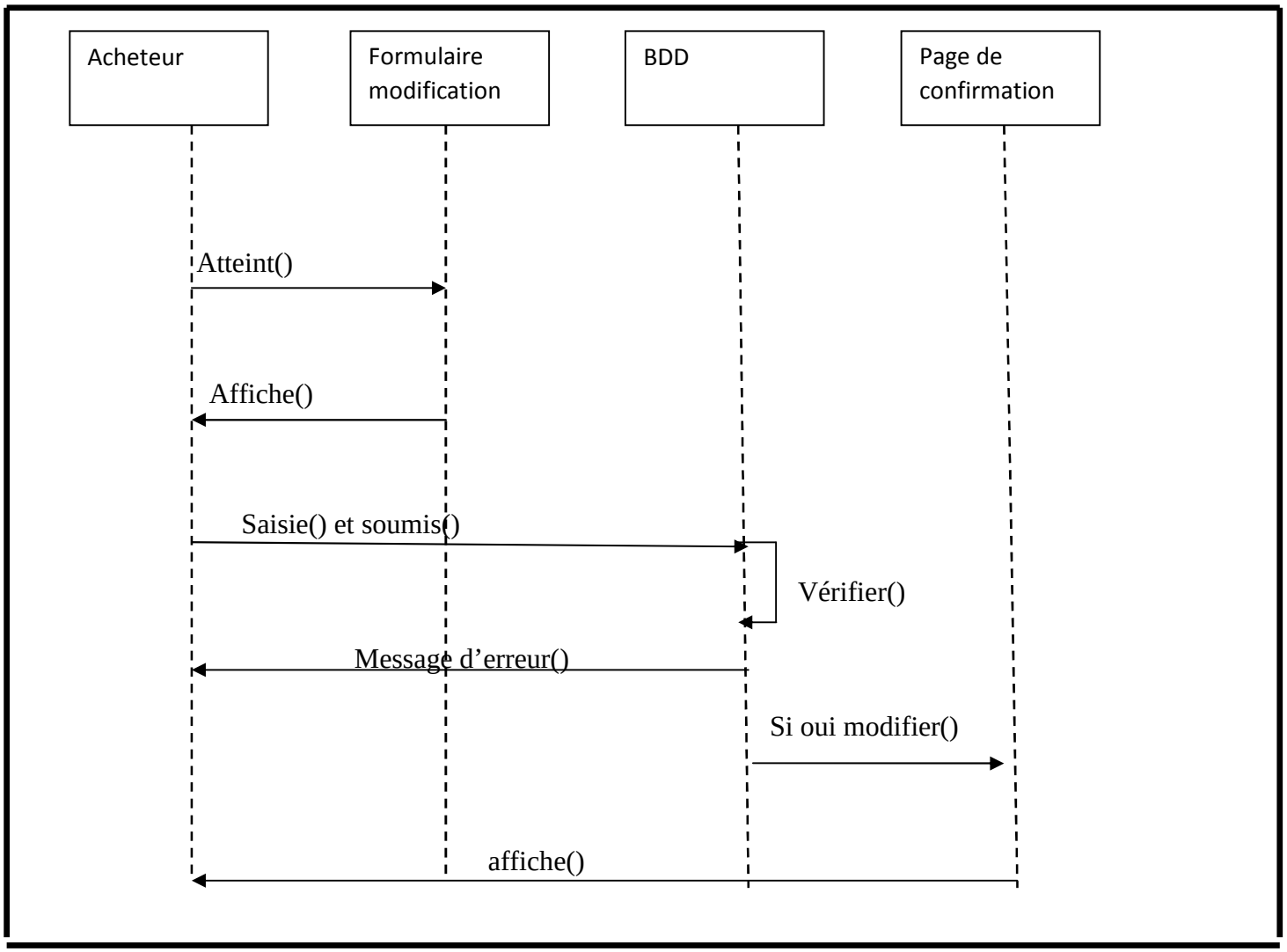


Figure III.9 : Diagramme de séquence du cas d'utilisation **: « modifier une facture»**

Diagrammes de classe pour l'itération2

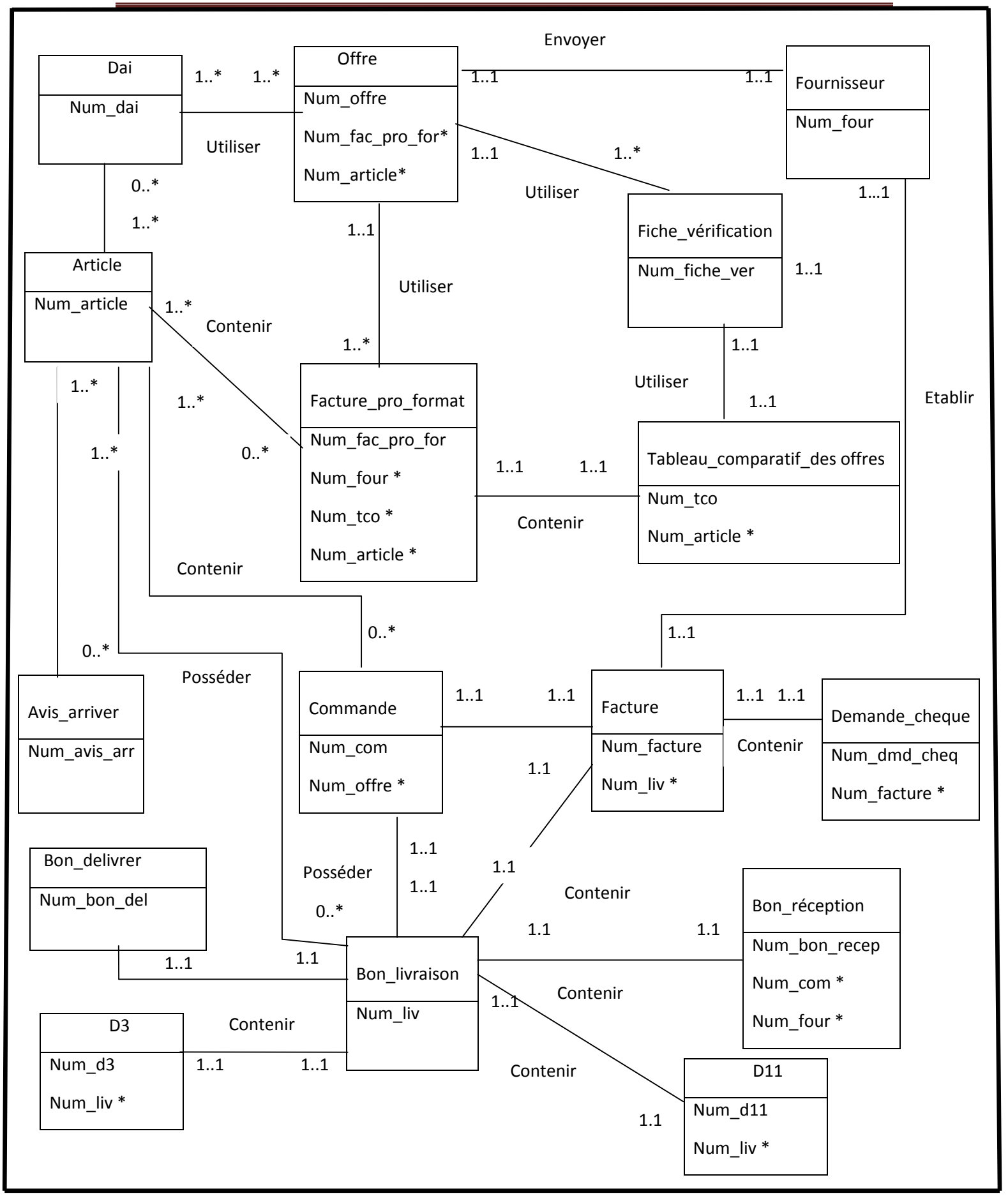


Figure III.10 : diagramme de classe pour l'itération2

Conclusion :

Tout au long de ce chapitre nous avons traité la phase d'analyse et de conception. Faite grâce à des diagrammes UML : diagrammes de séquences, de classes et de cas d'utilisation. La conception de la base de données est faite à l'aide d'un schéma relationnel.

L'étude détaillée nous a permis de décrire en détail la solution proposée et sa conformité aux choix de gestion et aux choix techniques, ce qui nous permettra de passer à la partie suivante qui est la réalisation de la solution informatique retenue.

Chapitre IV:

Réalisation

Introduction

Après la présentation de l'analyse et de la conception de notre application nous allons présenter dans ce chapitre l'environnement de développement ainsi que les différents outils utilisés et nous finirons par la présentation de quelques interfaces du logiciel.

IV.1. Les outils et l'environnement de développement :

IV.1.1. Netbeans IDE [13] :

Netbeans est l'environnement de Développement Intégré (EDI) supporté par SUN. Il est particulièrement bien adapté pour le développement d'applications WEB. Il remplace l'IDE Java Studio Creator. C'est un IDE moderne offrant un éditeur avec des codes couleurs et un ensemble de signes, des modèles de projets multi-langage et de différents types (application indépendante, distribuée, plugin, mobiles, ...), le refactoring, l'éditeur graphique d'interfaces et de pages web pour supporter le programmeur dans son travail. Il permet d'accéder rapidement à la documentation détaillée, de naviguer dans les sources et de faire des recherches d'usage des classes, méthodes et propriétés. Netbeans indique à l'utilisateur les erreurs et fait des propositions pour y remédier. Un déboggeur permet l'exécution pas à pas. Un suivi des ressources utilisées (cpu, mémoire) par le logiciel développé peut être fait via un profiler. Un framework de test unitaire tel que Junit peut être utilisé.

L'EDI NetBeans fournit des outils pour construire tous les composants Java EE, ce qui inclut les Enterprise Java Beans (EJBs), les pages web, les servlets, et les services web. Il intègre le serveur d'application Glassfish, ce qui permet de facilement développer des EJB et de les déployer. Il intègre la norme WebService JAX-WS. Il est aisé de lier un WS avec un EJB pour faire son implémentation.

Netbeans est également un moteur graphique servant de base logicielle pour le développement de RAD (Rich Application Desktop). Il offre un framework RCP (Rich Client Programming) pour développer tout type d'application graphique en utilisant l'environnement graphique Swing de Java.

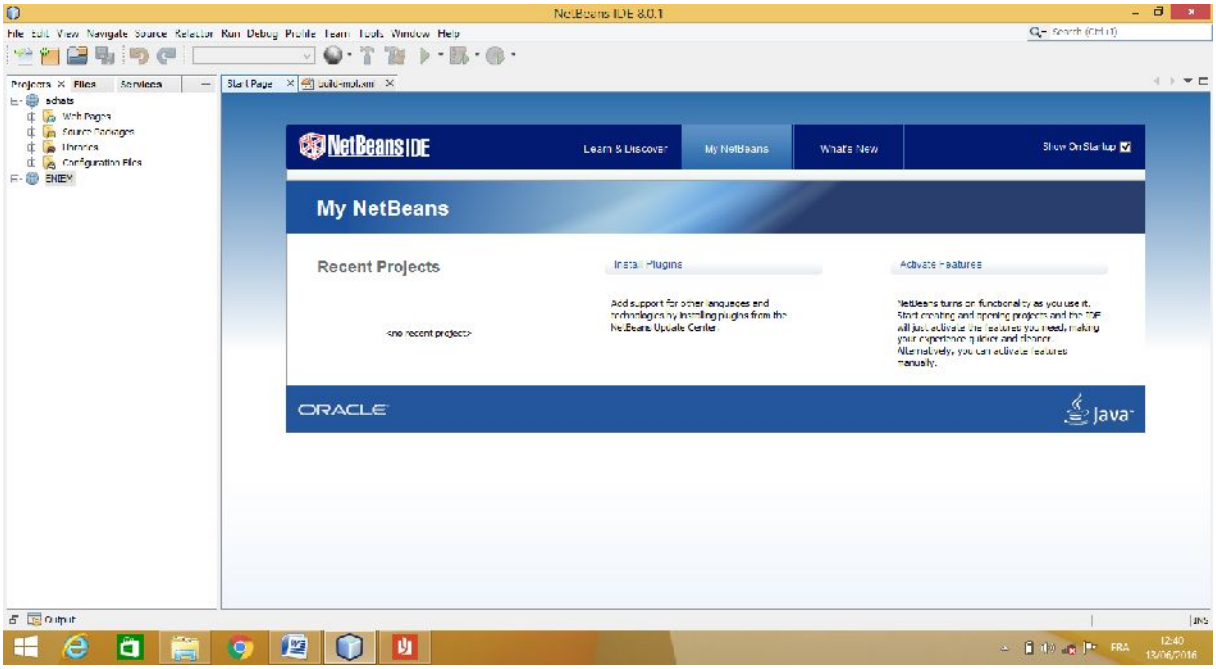


Figure IV.1 : l'interface de NetBeans IDE

IV.1.2. Hibernate Framework [14] :

Hibernate est un framework, écrit en java, qui permet de faire le mapping entre Objets Java et objet stockés en base relationnelle en assurant la persistance, et aussi s'occupe du transfert des classes Java dans les tables de la BDD et des types de données dans les types de données SQL.

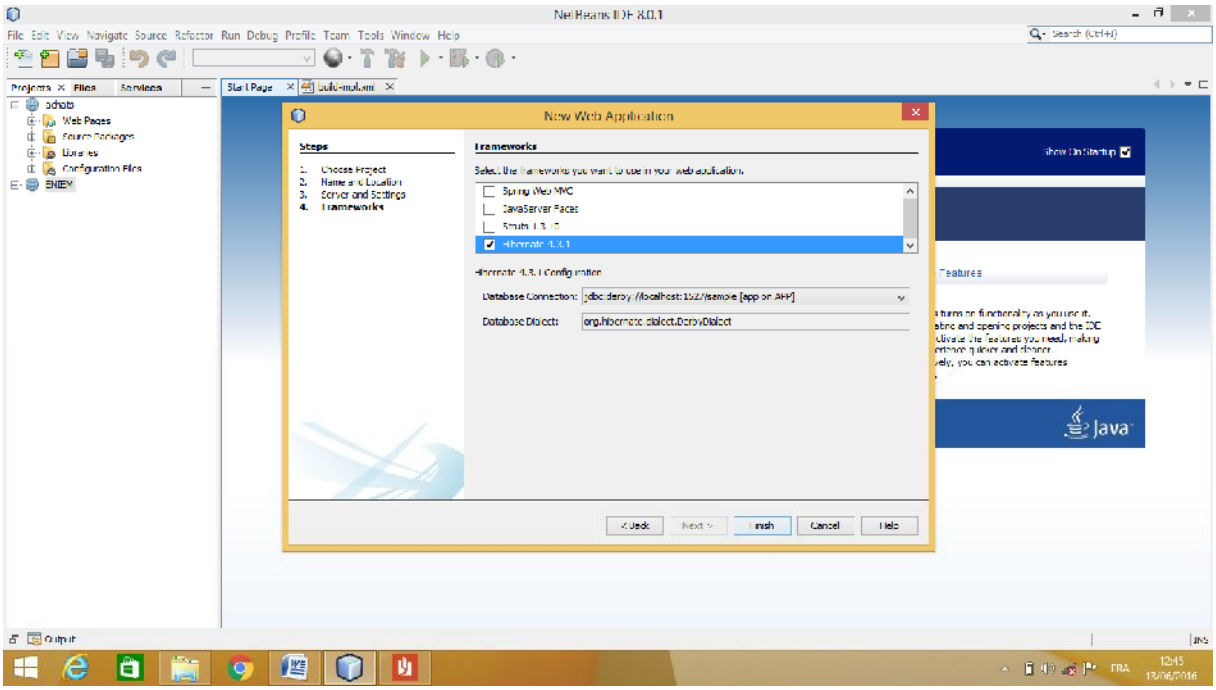


Figure IV.2 : l'interface de Hibernate

IV.1.3. JSF :

Est un *framework* web MVC qui se focalise sur la simplification de la construction d'interfaces utilisateur, autrement dit la vue ou présentation. Il propose nativement plus d'une centaine de balises à cet égard, et facilite la réutilisation des composants d'une interface utilisateur à l'autre.

IV.1.3.1. Structure d'une application JSF :

Étudions brièvement comment se construit une application basée sur JSF. Cela ne devrait pas bouleverser nos habitudes, la structure globale est très similaire à celle d'une application web Java EE MVC traditionnelle :

- la vue est généralement assurée par des pages JSP ou par des pages XHTML (on parle alors de Facelets).
- le modèle est assuré par des entités ou des JavaBeans.
- le contrôleur, auparavant incarné par nos servlets, est dorénavant décomposé en deux éléments :

- une unique servlet mère servant de point d'entrée à toute requête, la FacesServlet .
- un JavaBean particulier, déclaré via une annotation et désigné par le terme *managed-bean*.

IV.1.4. Le serveur Web Apache Tomcat [15] :

Apache Tomcat est une implémentation open source d'un conteneur web qui permet donc d'exécuter des applications web reposant sur les technologies servlets et JSP.

Tomcat est diffusé en open source sous une licence Apache. C'est aussi l'implémentation de référence des spécifications servlets jusqu'à la version 2.4 et JSP jusqu'à la version 2.0 implémentées dans les différentes versions de Tomcat.

IV.1.5. Oracle Database Express :

Oracle est un SGBD (système de gestion de base de données) édité par la société du même nom (oracle corporation). Il permet d'assurer la définition, la manipulation, la cohérence, la confidentialité et l'intégrité des données.
La solution oracle est un véritable environnement de travail constitué de nombreux logiciels permettant notamment une administration graphique d'oracle, de s'interfacer avec des produits divers et d'assistants de création de base de données et de configuration de celles-ci.

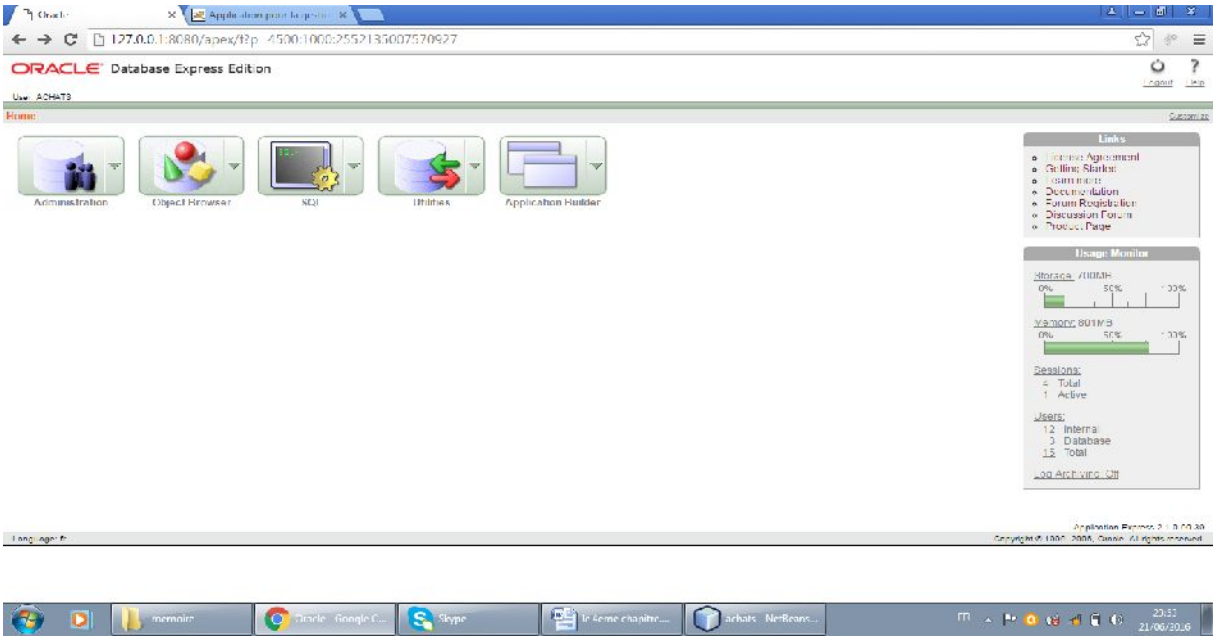


Figure IV.3 : l'interface oracle.

IV.2. Les langages de programmation :

IV.2.1. Java [16] :

Le langage Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Jc, (cofondateur de Sun Microsystems en 1982), présenté officiellement au *SunWorld.*

Le langage Java reprend en grande partie la syntaxe du langage C++, très utilisée par les informaticiens. Néanmoins, Java a été épurée des concepts les plus subtils du C++ et à la fois les plus déroutants, tels que les pointeurs et références, ou l'héritage multiple contourné par l'implémentation des interfaces. Les concepteurs ont privilégié l'approche orientée objet de sorte qu'en Java, tout est objet à l'exception des types primitifs (nombres entiers, nombres à virgule flottante, etc.).

Java a donné naissance à un système d'exploitation (JavaOS), à des environnements de développement (eclipse/JDK), des machines virtuelles (MSJVM (en), JRE) applicatives multiplate-forme (JVÎ.I), une déclinaison pour les périphériques mobiles/embarqués (J2ME), une bibliothèque de conception d'interface graphique (AWT/Swing), des applications lourdes (Jude, Oracle SQL Worksheet, etc.), des technologies web (servlets, applets) et une déclinaison pour l'entreprise (J2EE). La portabilité du bytecode Java est assurée par la machine virtuelle Java, et éventuellement par des bibliothèques standard incluses dans un

JRE. Cette machine virtuelle peut interpréter le bytecode ou le compiler à la volée en langage machine. La portabilité est dépendante de la qualité de portage des JVM sur chaque OS.

IV.2.2. XHTML [17] :

XHTML (Extensible Hyper Text Markup Language) est un langage de balisage servant à écrire des pages pour le word wide web. Conçu à l'origine comme le successeur de HTML, XHTML se fonde sur la syntaxe définie par XML. Il s'agissait en effet à l'époque de tirer parti des bénéfices techniques attendus de la simplification offerte par XML.

IV.2.3. Java Entreprise Edition (Java EE) [18]:

Java Enterprise Edition, ou Java EE (anciennement J2EE), est une spécification pour la technique Java d'Oracle plus particulièrement destinée aux applications d'entreprise. Ces applications sont considérées dans une approche multi-niveaux. Dans ce but, toute implémentation de cette spécification contient un ensemble d'extensions au *framework* Java standard (JSE, *Java Standard Edition*) afin de faciliter notamment la création d'applications réparties.

IV.3. Le modèle relationnel :

Le passage vers le modèle relationnel donne lieu a un modèle logique de données qu'on écrit sous une forme relationnelle comme suit:

Article : (num_article, date_etab_article, prix_achat, prix_vente, qte_delivrer, date_liv, lieu_liv, designation)

Avis_ariver :(num_avis_arr, date_etab_aarrive, nom_banque, nom_navire, heur_arr_march, num_liv*)

Bon_delivrer :(num_bon_del, date_etab_bon_del, lieu_etab_bon_del, num_liv*)

Bon_livraison : (num_liv, date_etab_liv, adresse_liv, raison_sociale, march_remise, date_deliv, lieu_deliv, nom_produit, qte, observation, reste_a_liv)

Bon_reception :(num_bon_recep, date_etab_bon_recep, date_etab_com, designation, date_etab_liv, mer, air, num_com*, num_liv*, num_four*)

Commande :(num_com, adr_com, date_etab_com, adr_envoi,date_consultation, designation, qte_com, prix_unitaire, montant_total, tva, montant_rendu, num_offre*)

Dai :(num_dai, date_etab_dai, date_emission, objectif_dai, nbr_article)

Demande_cheque: (num_dmd_cheq, date_etab_cheq, lieu_etab_cheq, montant_cheq, num_facture*)

D3 :(num_d3, date_etab_d3, heur_etab_d3, l ieu_etab_d3, num_liv*)

D11 :(num_d11, date_etab_d11, heur_etab_d11, lieu_etab_d11, num_liv*)

Facture :(num_facture, qte, montant_hors_tax, taux, montant_tva, montant_ttc, tva, total_hors_tax, total_ttc, num_liv*)

Facture_pro_format :(num_fac_pro_for, date_etab_fac_pro_for, lieu_etab_fac_pro_for, num_art*, code_produit, designation, qte, prix_uitaire, tva, ttc, num_four*, num_tco*)

Fiche_de_verification:(num_fiche_ver, nom_four, montant, qualite, rigueur, norme, evolution_prix, reference, contrat, commande, p1, p2)

Fournisseur :(num_four, nom_four, prenom_four, adresse_four, fax_four, tel_four, type_four)

Offre :(num_offre, date_etab_offre, description, qte, prix_unitaire, tva, total, total_avant, num_fac_pro_for*, num_article*)

Tco : (num_tco, date_etab_tco, designation, qte, unite, nom_four1, nom_four2, nom_four2, devise1, devise2, devise3, dinars1, dinars2, dinars3, four_retenu, num_article*)

Utilisateur :(id_utilisateur, login, password, profil, nom_utilisateur, prenom_utilisateur, adresse_utilisateur, matricule, date_naissance)

IV.3.1. Le modèle physique de données :

Ce modèle nous donne la représentation physique de l'ensemble des tables de la base de données du système étudié.

- **description des tables**

- La table « Article »

Nom du champ	Désignation	Types de données	Clé(s)
Num_article	Numéro article	Number(10)	Primaire
Date_etab_article	Date établissement article	Date	
Prix_achat	Le prix d'achat de la marchandise	Number(20)	
Prix_vente	Le prix vente de la marchandise	Number(20)	
Qte_delivrer	La quantité délivrée	Number(10)	
Date_liv	La date de livraison	Date	
Lieu_liv	Lieu de livraison	Varchar2(50)	
Désignation	Désignation	Varchar2(20)	

- La table « avis_arriver »

Nom du champ	Désignation	Types de données	Clé(s)
Num_avis_arr		Number(10)	Primaire
Date_etab_aarrive		Date	
Nom_banque		Varchar2(30)	
Nom_navire		Varchar2(30)	
Heure_arr_march		Time	
Num_liv		Number(20)	Etrangère

- La table « bon_delivrer »

Nom du champ	Désignation	Types de données	Clé(s)
Num_bon_del	Numéro de bon délivré	Number(10)	Primaire
Date_etab_bon_del	Date d'établissement du bon délivré	Date	
Lieu_etab_bon_del	Lieu d'établissement du bon délivré	Varchar2(50)	
Num_liv	Numéro de livraison	Number(20)	Etrangère

- La table « bon_livraison »

Nom du champ	Désignation	Types de données	Clé(s)
Num_liv	Numéro de livraison	Number(20)	Primaire
Date_etab_liv	Date d'établissement du bon de livraison	Date	
Adresse_liv	Adresse livarison	Varchar2(50)	
Raison_sociale	Raison sociale	Varchar2(20)	
March_remise	Marchandise remise	Varchar2(30)	
Date_deliv	Date de livraison	Date	
Lieu_deliv	Lieu de livraison	Varchar2(20)	
Nom_produit	Nom de produit	Varchar2(20)	
Qte	Quantité à livrer	Number(20)	
Observation	Observation	Varchar2(20)	
Reste_a_liv	Reste à livrer	Number(20)	

- La table « bon_reception »

Nom du champ	Désignation	Types de données	Clé(s)
Num_bon_recep	Numéro du bon de récéption	Number(20)	Primaire
Date_etab_bon_recep	Date d'établissement du bon de récéption	Date	
Désignation	Désignation	Varchar2(20)	
Date_etab_com	Date d'établissement de la commande	Date	
Date_etab_liv	Date d'établissement de la livraison	Date	
Mer	Mer	Varchar2(10)	
Terre	Terre	Varchar2(10)	
Num_com	Numéro de la commande	Number(20)	Etrangère
Num_four	Numéro de fournisseur	Number(20)	Etrangère
Num_liv	Numéro de livraison	Number(20)	Etrangère

- La table « commande »

Nom de champ	Désignation	Types de données	Clé(s)
Num_com	Numéro de commande	Number(20)	Primaire
Adresse_com	Adresse de la commande	Varchar2(50)	
Date_etab_com	Date d'établissement de la commande	Date	
Adresse_envoi	Adresse d'envoi	Varchar2(50)	
Date_consultation	Date de consultation	Date	
Designation	Désignation	Varchar2(30)	
Qte_com	Quantité commandée	Number(11)	
Prix_unitaire	Prix_unitaire	Number(11)	
Monatnt_total	Monatnt_total	Number(11)	
Tva	Tva	Number(11)	
Num_offre	Numéro d'offre	Number(20)	Etrangère

- La table « DAI » (Demande d'Achat Interne)

Nom du champ	Désignation	Types de données	Clé(s)
Num_dai	Numéro de la demande d'achat interne	Number(20)	Primaire
Date_emission	Date d'émission	Date	
Nbr_article	Nombre d'articles demandés	Number(10)	
Date_etab_dai	Date d'établissement de la demande	Date	
Objectif_dai	Objectif de la dai	Varchar2(50)	

- La table « demande_cheque »

Nom du champ	Désignation	Types de données	Clé(s)
Num_dmd_cheq	Numéro de demande de cheque	Number(20)	Primaire
Date_etab_cheq	Date d'établissement de demande de cheque	Date	
Lieu_etab_cheq	Lieu d'établissement de demande de cheque	Varchar2(50)	
Monatant_cheq	Montant cheque	Number(11)	
Num_facture	Numéro de la facture	Number(20)	Etrangère

- La table « D11 » (la mise à l'entrepôt)

Nom du champ	Désignation	Types de données	Clé(s)
Num_dmd_d11	Numéro d11	Number(20)	Primaire
Date_etab_d11	Date d'établissement d11	Date	
Lieu_etab_d11	Lieu d'établissement d11	Varchar2(50)	
Heur_etab_d11	Heure d'établissement d11	Time	
Num_liv	Numéro de livraison	Number(20)	Etrangère

- La table « D3 » (la mise à la consommation)

Nom du champ	Désignation	Types de données	Clé(s)
Num_dmd_d3	Numéro d3	Number(20)	Primaire
Date_etab_d3	Date d'établissement d3	Date	
Lieu_etab_d3	Lieu d'établissement d3	Varchar2(50)	
Heur_etab_d3	Heure d'établissement d3	Time	
Num_liv	Numéro de livraison	Number(20)	Etrangère

- La table « facture »

Nom du champ	Désignation	Types de données	Clé(s)
Num_facture	Numéro facture	Number(20)	Primaire
Qte	Quantité	Number(11)	
Monatant_hors_tax	Monatant hors taxe	Number(11)	
Prix_unitaire	Prix unitaire	Number(11)	
Monatant_tva	Monatant tva	Number(11)	
Monatant_ttc	Monatant_ttc	Number(11)	
Tva	Tva	Number(11)	
Total_hors_tax	Total hors taxe	Number(11)	
Totl_ttc	Total ttc	Number(11)	
Num_liv	Numéro livraison	Number(20)	Etrangère

- La table «facture_pro_format »

Nom du champ	Désignation	Types de données	Clé(s)
Num_fac_pro_for	Numéro facture pro format	Number(20)	Primaire
Date_etab_fac_pro_for	Date établissement de la facture pro format	Date	
Lieu_etab_fac_pro_for	Lieu établissement de la facture pro format	Varchar2(50)	
Code_produit	Code produit	Varchar2(20)	
Designation	désignation	Varchar2(30)	
Monatant_ttc	Monatant_ttc	Number(11)	
Qte	Qte	Number(10)	
Prix_unitaire	Prix unitaire	Number(10)	
Tva	tva	Number(10)	
Ttc	ttc	Number(10)	
Num_four	Numéro fournisseur	Number(20)	Etrangère
Num_tco	Numéro de tableau comparatif des offres	Number(20)	Etrangère
Num_article	Numéro de l'article	Number(20)	Etrangère

- La table « fiche_vérification »

Nom du champ	Désignation	Types de données	Clé(s)
Num_fiche_ver	Numéro fiche de vérification	Number(20)	Primaire
Nom_four	Nom fournisseur	Varchar2(20)	
Monatant	montant	Number(10)	
Qualité	Qualité de produit	Varchar2(20)	
Rigueur	Rigueur dans les délais	Varchar2(20)	
Evolution_prix	Evolution de prix	Varchar2(20)	
Reference	Référence	Varchar2(20)	
Commande	Commande	Varchar2(20)	
Contrat	Contrat	Varchar2(20)	
P1	Fournisseur (agréé, qualifié, nouveau)	Varchar2(20)	
P2	Fournisseur (agréé, qualifié, nouveau)	Varchar2(20)	
Norme	Norme	Varchar2(20)	
Num_four	Numéro fournisseur	Number(20)	Etrangère

- La table « fournisseur »

Nom du champ	Désignation	Types de données	Clé(s)
Num_four	Num fournisseur	Number(20)	Primaire
Nom_four	Nom fournisseur	Varchar2(30)	
Prenom_four	Prénom fournisseur	Varchar2(30)	
Adresse_four	Adresse fournisseur	Varchar2(50)	
Fax_four	Fax fournisseur	Number(10)	
Tel_four	Téléphone	Number(10)	
Type_four	Type fournisseur	Varchar2(30)	

- La table « offre »

Nom du champ	Désignation	Types de données	Clé(s)
Num_four	Num d'offre	Number(20)	Primaire
Date_etab_offre	Date établissement de l'offre	Date	
Description	Description	Varchar2(50)	
Qte	Quantité	Number(11)	
Prix_unitaire	Prix unitaire	Number(11)	
Tva	Tva	Number(10)	
Total	Total	Number(20)	
Num_fac_pro_for	Numéro facture pro format	Number(20)	Etrangère
Num_article	Numéro article	Number(20)	Etrangère

- La table « utilisateur»

Nom du champ	Désignation	Types de données	Clé(s)
Id_utilisateur	Identifiant de l'utilisateur	Varchar2(30)	Primaire
Login	Utilisateur	Varchar2(30)	
Password	Mot de passe	Varchar2(30)	
Profil	Profil	Varchar2(30)	
Nom_utilisateur	Nom utilisateur	Varchar2(30)	
Prenom_utilisateur	Prenom_utilisateur	Varchar2(30)	
Adresse_utilisateur	Adresse utilisateur	Varchar2(30)	
Num_tel	Téléphone	Number(10)	
Matricule	Matricule	Varchar2(30)	

- La table « TCO » (Tableau Comparatif des Offres)

Nom du champ	Désignation	Types de données	Clé(s)
Num_tco	Numéro de tableau comparatif des offres	Number(20)	Primaire
Date_etab_tco	Date établissement tco	Date	
Designation	Désignation	Varchar2(30)	
Qte	Quantité	Number(11)	
Unite	Unité de mesure	Varchar2(10)	
Nom_four1	Nom fournisseur a	Varchar2(30)	
Nom_four2	Nom fournisseur b	Varchar2(30)	
Nom_four3	Nom fournisseur c	Varchar2(30)	
Devise1	Devise a	Number(10)	
Devise2	Devise b	Number(10)	
Devise3	Devise c	Number(10)	
Dinars1	Dinars a	Number(10)	
Dinars2	Dinars b	Number(10)	
Dinars3	Dinars c	Number(10)	
Four_retenu	Fournisseur retenu	Varchar2(20)	

IV.4. Présentation de quelques interfaces de l'application :

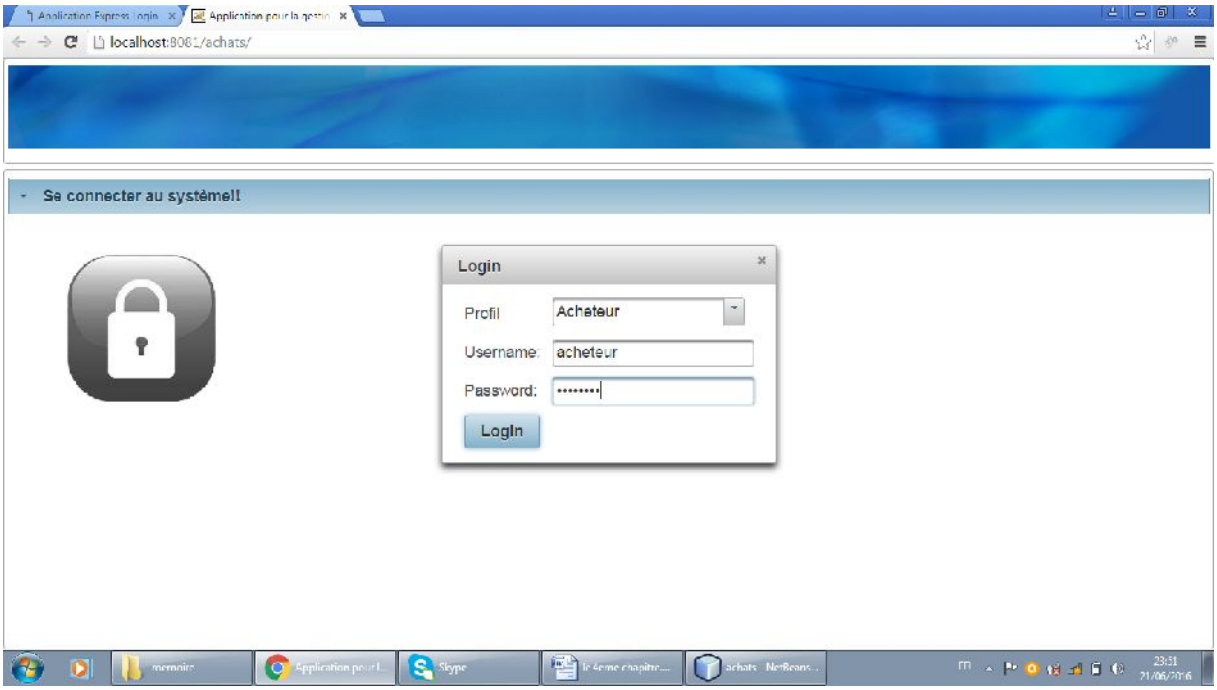


Figure IV.4 : page d'authentification

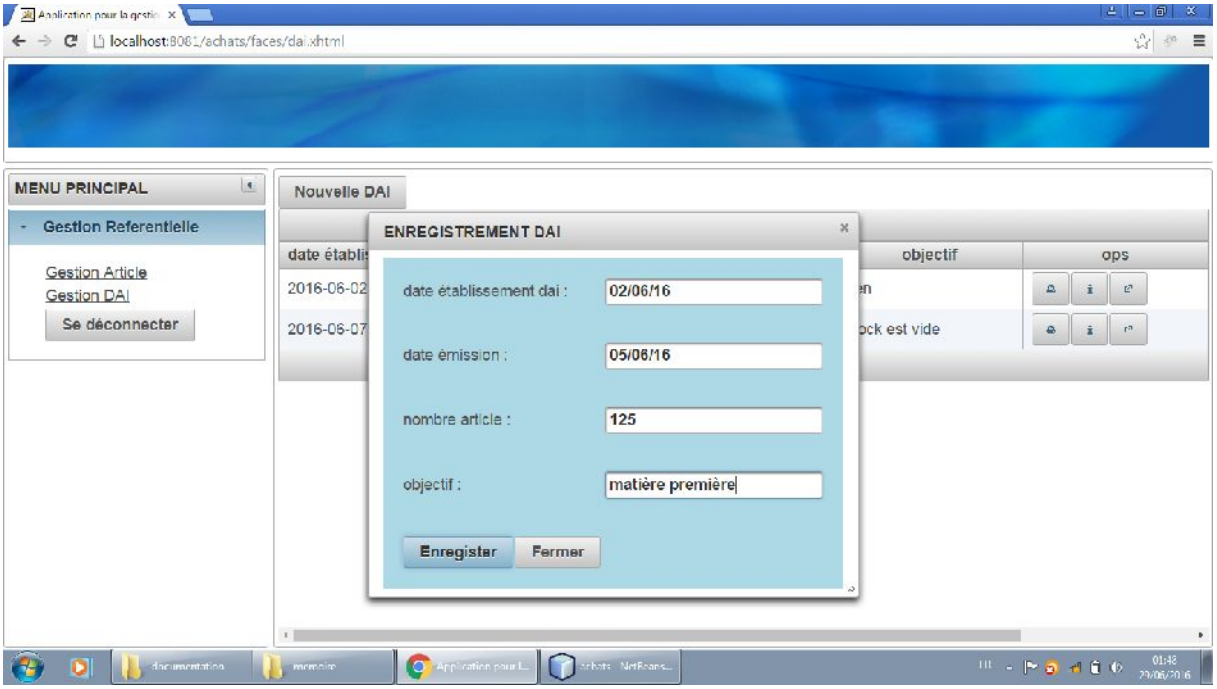


Figure IV.5 : page « ajouter DAI »

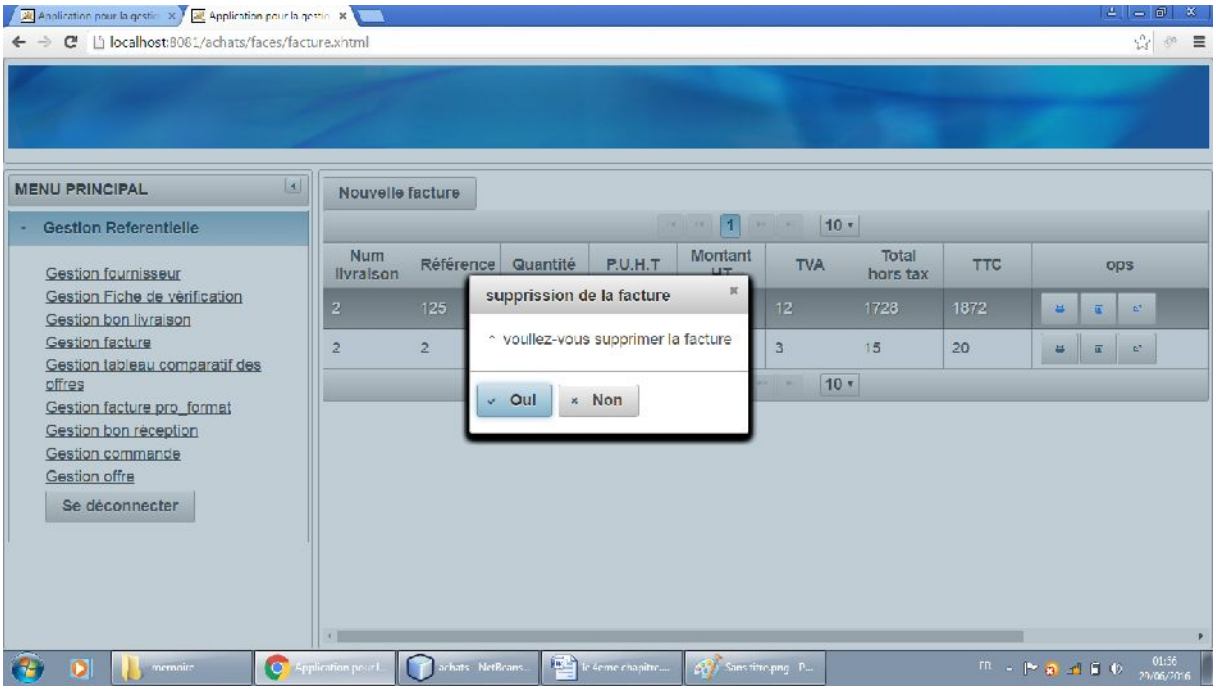


Figure IV.6 : page « suppression d'une facture »

Conclusion

Dans ce chapitre nous avons cité les différents utils utilisés ainsi que l'environnement de développement et les langages de programmation utilisés et cela à travers l'implémentation de notre application présentée par quelques interfaces illustrant ces différentes fonctionnalités.

CONCLUSION GENERALE

Notre travail consistait à réaliser une application Java EE pour la gestion des achats locaux et import. Ces derniers nous ont permis de découvrir le domaine des achats et de voir de prés les tâches complexes aux quelles le personnel des trois services (service achat, service stock et service transit et douane) font face.

Après le passage par les différentes étapes, à savoir l'analyse, la conception et la réalisation, nous avons réussi à réaliser une application qui offre plusieurs services : elle rend aisée la tâche de gestion à tous les niveaux. Elle permet aux utilisateurs d'accéder aux informations en temps réel.

L'étude et le développement de notre projet informatique nous ont permis de mettre en pratique des connaissances théoriques acquises durant notre cursus universitaire en particulier dans le domaine de la conception des systèmes d'informations, de la conception objet (UML, JEE, …), de la programmation objet et des bases de données. En outre nous avons acquis des connaissances pratiques en matière de systèmes d'informations et de gestions.

Cependant, malgré les diverses fonctionnalité qu'offre notre application, quelques améliorations et compléments peuvent être apportés. En guise de perspectives :

- Réalisation des états de sorties ;
- Mettre au point un module autorisant un acheteur à passer une commande en ligne avec un mécanisme d'authentification et de sécurisation de l'opération.

Enfin nous espérons que notre travail sera de grands intérêts pour l'ENIEM et un guide efficace pour les nouvelles promotions.

BIBLIOGRAPHIE

Webliographie/Bibliographie

[1] : IFT2255 :Génie logiciel, Julie Vachon 2008

[2] : http://fr.wikipedia.org/wiki/Architecture_logicielle

[3] : *Le modèle client-serveu*r Olivier Aubert .pdf

[4] : http://www.commentcamarche.net/cs/peer.php3

[5] : Université Hassiba Ben Bouali Chlef /Faculté des sciences et sciences de l'ingénieur / Département TCT/support du cours /architecture réseaux

[6] : www.comment-ça-marche.net

[7] : Mémoire promotion 2012/2013 réalisé par :amranesamia et amranehassina (conception et réalisation d'une application web avec J2EE pour la gestion de scolarité)

[8] :http://www.ai.univ-paris8.fr/~maa/M1_J2EE_Jsp.pdf

[9] : http://www.jmdoudoux.fr/java/dej/chap-ejb.htm

[10] :http://infocenter.sybase.com/archive/index.jsp?topic=/com.sybase.stf.powe ramc.docs_12.0.0/html/dcgu/dcgup811.html

[11] : Laurent Aulibert, UML 2 de l'apprentissage à la pratique(cours et exercices),édition elipses.

[12] :Définition et caractéristique d'UML,2008

[13] : html://fr.Wikipedia.org/Wiki/netbeanes

[14] : html://fr.Wikipedia.org/Wiki/Hibernate

[15] :html://fr.Wikipedia.org/Wiki/ApacheTomcat

[16] : html://fr.Wikipedia.org/java-(langage).

[17] : html://fr.Wikipedia.org/Wiki/Xml-Hypertexte-markup-langage

[18] : *http://www.framasoft.net/article3991.html*

Annexes

Introduction

La conception de toute solution logicielle doit être traitée d'une manière précise et détaillée, précédée d'une analyse profonde et bien réfléchie, car elle est le reflet du futur système avant même sa réalisation. Dans le but d'avoir une meilleure analyse et de rendre la conception de notre application plus complète, nous avons opté le langage UML qui permet de bien représenter l'aspect statique et dynamique d'une application par une série de diagramme qu'il offre.

1. Définition UML :

Le langage de modélisation unifié, de l'anglais *UnifiedModelingLanguage* (**UML**), est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet. L'UML est le résultat de la fusion de précédents langages de modélisation objet : Booch, OMT, OOSE. Principalement issu des travaux de GradyBooch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard adopté par l'Object Management Group (OMG).

2. Utilité d'UML

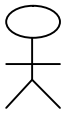
UML est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet. UML offre un standard de modélisation, pour représenter l'architecture logicielle. Les différents éléments représentables sont :

· Activité d'un objet/logiciel
· Acteurs
· Processus
· Schéma de base de données
· Composants logiciels
· Réutilisation de composants

Grâce aux outils de modélisation UML, il est également possible de générer automatiquement une partie de code, par exemple en langage Java, à partir des divers documents réalisés.

3. Quelques notions de base :

Acteur : un acteur représente un ensemble de rôles joués par une personne, une machine ou un autre système, il peut interagir avec un cas d'utilisation et participer à son scénario. Il est représenté par un petit bon homme avec son rôle inscrit dessous.

Acteur 

Tache: est un ensemble de fonctions qu'un acteur bien définit peut accéder.
Scénario: C'est une succession particulière d'enchainement, s'exécutant du début à la fin du cas d'utilisation.

4. les diagrammes d'UML :

UML est un langage graphique et repose sur neuf types de diagrammes représentant autant de vues distinctes pour représenter des concepts particuliers du système d'information.
UML permet de visualiser un modèle, á l'aide des diagrammes.

4.1. Les vues d'UML :

On distingue trois types de vues :

- **Vue statique (structurelle) :** elle représente le système physiquement et elle s'appuie sur les diagrammes suivants :
  Diagramme de classe
  Diagramme d'objet
  Diagramme de composant
  Diagramme de déploiement

- **vue dynamique (comportementaux) :** elle s'appuie sur les diagrammes suivants :
  Diagramme d'activité
  Diagramme de séquence
  Diagramme d'étude
  Diagramme de collaboration

- **vue fonctionnelle :** elle s'appuie sur les diagrammes suivants :
  Diagramme de cas d'utilisation

Pour le cas de notre application les diagrammes utilisés sont :

- ✓ Diagramme de classe
- ✓ Diagramme de cas d'utilisation
- ✓ Diagramme de séquence

4.1.1. Les diagrammes de cas d'utilisation :

Les diagrammes de cas d'utilisation décrivent sous la forme d'actions et de réactions le comportement d'un système du point de vue d'un utilisateur, ils modélisent une interaction entre le système informatique à développer et un utilisateur ou acteur interagissant avec le système.
Les diagrammes de cas d'utilisation représentent un ensemble de cas d'utilisation, et d'acteurs et les différentes relations entre eux c'est la vue statique des cas d'utilisation d'un système.

4.1.2. Les diagrammes de séquence:

Le diagramme de séquence représente la succession chronologique des opérations réalisées par un acteur. Il indique les objets que l'acteur va manipuler et les opérations qui font passer d'un objet à l'autre. On peut représenter ces opérations par un diagramme de communication dont les nœuds sont des objets et les arcs (numérotés selon la chronologie) les échanges des messages entre objets. En fait, diagramme de séquence et diagramme de communication sont deux vues différentes mais logiquement équivalentes d'une même chronologie.
L'ordre d'envoi d'un message est déterminé par sa position sur l'axe verticale du diagramme, le temps s'écoule de haut en bas de cet axe.

4.1.3. Les diagrammes de classe :

Un diagramme des classes constitue le coeur du langage UML, il est considéré le plus important dans un développement orienté objets.il décrit les classes que le système utilise, ainsi que leurs liens, il permet de définir quelle serons les composantes du système finale, il permet aussi de construire le système de manière correcte.

L'intérêt majeur de ce diagramme est de représenter les entités du système d'information.

Présentation de la démarche RUP :

Introduction

RUP (Relationel Uniffied Process) dérivée de UP

Processus unifié (UP) est un processus de développement logiciel : il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel.

Caractéristiques essentielles du processus unifié :

- Le processus unifié est à base de composants
- Le processus unifié utilise le langage UML (ensembles d'outils et de diagrammes)
- Le processus unifié est piloté par les cas d'utilisation
- Centré sur l'architecture
- Itératif et incrémenté

1. RUP (Rationel Unified Process)

- Dérivée d'UP (UP a été créée en 1996)
- Commercialisée par IBM en 1998

2. Méthodes de développement logiciel :

- Itérative
- Incrémental
- Pilotée par le cas d'utilisation
- Centrée sur l'architecture et la réduction des risques
- Produit de qualité

3. Phases de RUP :

3.1. Phase d'initialisation :

- Définir la portée du projet
- Spécification
- Affectation des taches
- Evaluation des risques
- Cycle de vie

3.2. Phase d'élaboration :

- Planifier le projet
- Spécifier les fonctionnalités
- Construire l'architecture
- Spécification du produit
- Conception de l'architecture
- Planification activités-ressources

3.3. Phase de construction :

- Construire le produit
- Implémentation du produit
- Tests
- Produit opérationnel

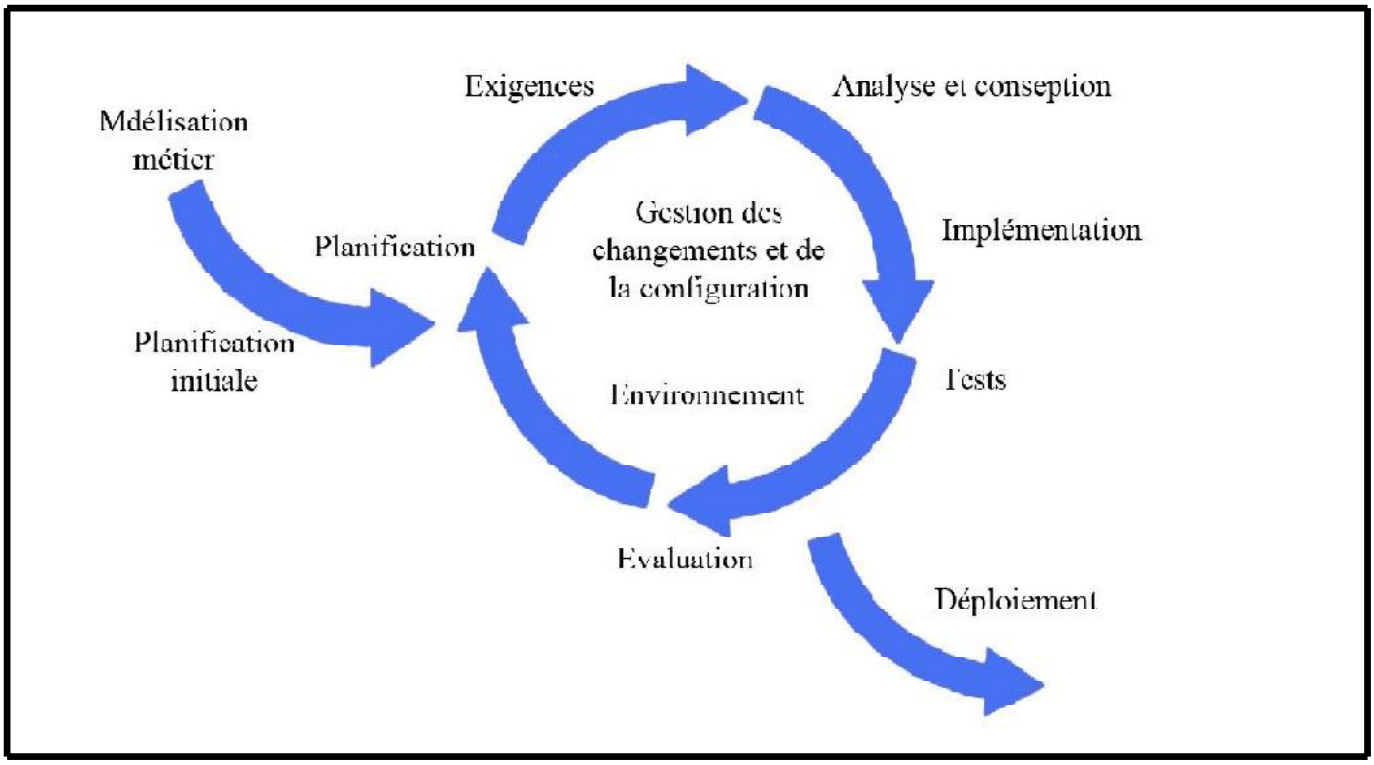
3.4. Phase de transition :

- Transition du produit vers les utilisateurs
- Livraison
- Formation
- Qualité

4. Cycle de vie de RUP :

- **Exigence :** détermination des besoins :
  - Fonctionnel (ce que l'on attend du système)
  - Non fonctionnel (fiabilité, temps de réponse, environnement distribué, etc)
- **Analyse et conception** : évaluer depuis la spécification des besoins jusqu'à une solution informatique
  - Analyse des besoins fonctionnels
  - Conception intègre aussi les besoins non fonctionnels
- **Implémentation :**
  - Transcription dans un langage de programmation ou de base de données
  - Utilisation de composants existants

- **Tests :**
  - Estimer si les besoins sont satisfaits, s'il y a des erreurs/défauts à corriger renforcer et stabiliser l'architecture
- **Déploiement :**
  - Distribuer le logiciel dans son environnement opérationnel, installation test formation des utilisateurs, migration des données



Cycle de vie de RUP

5. Différence entre incrémental et itératif

- **RUP est incrémental**

Le produit final est livré en incréments

- Chaque incrément livre une fonctionnalité ayant une valeur pour l'utilisateur final
- Chaque incrément consiste en un ensemble de cas d'utilisation
- Les incréments doivent être de courte durée (quelques semaines à quelques mois)
- Les incréments sont des mini-projets dans un projet

- **RUP est itératif**

Une itération est une séquence d'activités qui répond à un plan et à des critères d'évaluation et qui produit une version exécutable

6. Forces de la méthode RUP :

- Traçabilité à partir des uses-cases jusqu'au déploiement
- Approche basée sur l'architecture
- Gestion des risques dans les projets

7. Faiblesses de la méthode RUP :

- Cout de personnalisation souvent élevés
- Très axé processus

Conclusion

Cette partie nous a permet de prendre connaissance le processus RUP (Relationnel Unified Process) qui est une méthode de développement pour logiciel orienté objet, c'est une méthode générique itérative et incrémentale.