

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU
FACULTE DE GENIE ELECTRIQUE ET DE L'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

MEMOIRE DE FIN D'ETUDES

*En vue de l'obtention du Diplôme de Master
En informatique
Option : Conduite de projet informatique*

THEME

*Conception et réalisation d'une Application Java EE sur la
gestion des achats locaux et importation*

Cas : ENIEM

Dirigé par :

Mr. KERBICHE M'HEND

Proposé par :

Mme DJEMAH LILA

Réalisé par :

Mlle KASHI SAMIRA

Mlle YOUNSI SOUAD

Promotion:2014/2015



REMERCIEMENT

Nous tenons à témoigner notre reconnaissance à dieu tout puissant, qui nous



A aidé et béni par sa volonté durant toute cette période.

*Nous tenons à exprimer nos sincères remerciements à notre
Promoteur monsieur Kerbiche pour l'intérêt et l'aide qu'il nous a donné.*

*Nous tenons également à remercier le personnel de service achat en particulier
monsieur Kessroui, qui nous a pris en charge tout au long De stage pratique.*

*Nos remerciements vont également aux membres du jury pour
L'honneur qu'ils nous font en acceptant d'examiner et de
Juger notre travail.*

*Nous remercions aussi tous ceux et celles qui ont contribué
Deprès ou de loin pour l'accomplissement de
Ce modeste travail*



DEDICACE

A la mémoire de mes grands parents

*A mon père SAÏD et ma mère NOUARA qui ont sacrifié
beaucoup pour moi et grâce à leurs amours et leurs veilles que j'ai pu
arriver à ce stade.*

*A mes chère frères HAKIM, MOHAMMED et NASSIM,
A mes très chère sœurs NOURA et HOURIA avec son marie et ses petits-
enfants AYA et TARIK que j'adore.*

A mes cousins et cousines.

A mes oncles, tantes et à toute la famille Kashi.

*A mes copines : Souad, Dyhia, tassa, Kami, Souhila, Amel, Hassiba,
Radia, Safia, Lynda...*

*A tous les gens qui m'aiment et à ceux qui me sont chères
A tous mes ami(e)s qui mon encouragée pendant la réalisation de ce
modeste travail*

Et surtout à ma binôme SOUAD et sa famille.

K.SAMIRA



DEDICACE

A la mémoire de mes grand parents

*A mon père KHOLLEF et ma mère ELDJIDA qui ont sacrifié beaucoup pour
moi et grâce à leurs amours et leurs veilles que j'ai pu
arriver à ce stade..*

*A mes chère frères TAHAR, MOHAMMED A mes cousins et cousines
Nabila, Kahina, Hanane Lynda et la petite Nadine que j'adore.*

A mes oncles ,tantes et à toute la famille Younsi.

*A mes copines : Souad, Dyhia, tassa, Kami, Souhila, Amel, Hassiba,
Radia, Safia, Lynda...*

*A tous les gens qui m'aiment et à ceux qui me sont chères
A tous mes ami(e) s qui mon encouragée pendant la réalisation de ce modeste
travail*

Et surtout à ma binôme SAMIRA et sa famille.

Y.SOUAD

SOMMAIRE

Introduction général	
----------------------------	--

Chapitre I : La technologie de JavaEE

I. Introduction	p1
I.1. Architecture logicielle	p1
I.1.1. Fonctionnalité de qualité logicielle	p1
I.1.2. Critères de qualité logicielle	p1
I.2. L'architecture client/serveur	p2
I.2.1. Introduction	p2
I.2.2. Définition de l'architecture client/serveur.....	p2
I.2.3.Fonctionnement	p3
I.3. Les différentes modèles de client/serveur	p4
I.4. Définition d'un middleware	p6
I.4.1. Fonctions d'un middleware	p7
I.5.Type de client	p7
I.6. Type de serveurs pour le système client/serveur	p8
I.7. Les différents type d'architecture de client/serveur	p9
I.8. Avantage et inconvénient de l'architecture client/serveur	p11
I.9. Introduction de JavaEE	P12
I.9.1.Définition de JavaEE	P12
I.9.2. L'architecture de JavaEE	P12
I.10. Les Servlets	p14
I.11. Les JSP	P14
I.12. JDBC.....	P14
I.13.EJB	P14
I.14. Les services des API de Java EE	p15
I.14.1. Les services d'infrastructures.....	p15
I.14.2. Les services de communication	p15
I.15. Les avantages d'utiliser Java EE	p16
I.16. Conclusion	p17

Chapitre II : Présentation De L'organisme D'accueil

II .Introduction	p18
II.1.Présentation de l'Enterprise	P18
II.1.1 Situation géographique	P18
II.1.2.Historique	p18
II.1.3.l'organigramme de l'entreprise ENIEM	p19
II.2.Missions et l'objectif de l'entreprise	p20
II.3.Mode d'organisation	p20
II.4.les unité	p24
II.5.Organigramme de l'unité de prestation technique	p25
II.6. Organigramme du département commercial.....	p26
II.7.Organigramme du service d'achat	p27
II.8.Situation informatique des unités de l'entreprise	p27

II.9. Etude d'existante	p28
II.9.1.diagramme de flux	p28
II.9.2.concepts du diagramme	p28
II.9.3.présentation de diagramme de flux	p30
II.9.4.description des flux d'information	p30
II.9.5.critiques de l'existant et suggestions	p32
II.10. Les niveaux de l'entreprise	p33
II.10.1.Niveau organisationnel	p33
II.10.2.Niveau informationnel	p33
II.10.3.Niveau technique	p34
II.10.4.Solution informatique	p34
II.11. Conclusion	p36

Chapitre III. Analyse et Conception

III.introduction	p37
III.1. Phase d'initialisation	p37
III.2. Phase d'élaboration	p37
III.3. Phase de construction	p37
III.4.Phase de transition	p38
III.5. Identification des acteurs	p38
III.5.1. Diagramme de contexte	p38
III.5.2.Représentation de cas d'utilisation	p39
III.5.3. Les cas d'utilisation	p41
III.6. Définition des itérations	p45
III.6.1.réalisation de l'itération 1	p45
III.6.1.1.Diagramme de séquence	p45
III.6.1.2.Diagramme de classe de l'itération1	p50
III.6.1.3.Diagramme de classe de l'itération2	p51
III.6.1.4.Diagramme de classe de l'itération3	p52
III.7. Conclusion	p53

Chapitre IV : La Réalisation

IV. Introduction	p54
IV.1. Outils et développement	p54
IV.1.1. Le serveur WEB Apache TomCat	p54
IV.1.2. Hibernate Framework	p55
IV.1.3. Dreamweaver	p56
IV.1.4.EasyPHP	p56
IV.1.5. PHP Myadmin	p57
IV.1.6. Netbeans	p58
IV.2.Les langages de programmation	p59
IV.2.1. Java	p59
IV.2.2.HTML	p60
IV.2.3.Java EE	P60

IV.3. Le modèle relationnel	p60
IV.3.1.Le modèle physique de données	p62
IV.4. Présentation de quelques interfaces de l'application	p70
IV.5.Conclusion	p74
Conclusion générale	
Annexe	p75

Listes des figures

Figure. I.1.Organigramme client/serveur	P3
Figure. I.2.fonctionnement de modèle client/serveur.....	P3
Figure. I.3.Modèle client/serveur de donnée	P4
Figure. I.4. Modèle de client/serveur de présentation	P5
Figure. I.5.Modèle client/serveur de traitement	P5
Figure.I.6.La structure d'un middleware	P6
Figure. I.7.Présentation de l'architecture a 2 niveau	P9
Figure. I.8.Présentation de l'architecture a 3 niveau	P10
Figure. I.9.L'architecture multi niveau	P11
Figure. I.10.L'architecture de JEE	P13
Figure. I.11.Présentations les services des API deJEE	P16
Figure.II.12.Organigramme générale de l'ENIEM	P19
Figure. II.13.Direction général	P20
Figure. II.14.Organigramme de prestation technique.....	P25
Figure. II.15.Organigramme de département commercial	P26
Figure. II.16.Organigramme de service d'achat	P27
Figure. II.17.Diagramme de flux	P30
Figure. II.18.Architecture globale du modèle proposé	P35
Figure. III.19. Diagramme de contexte	P39
Figure. III.20. Diagramme globale	P41
Figure. III.21. Diagramme de séquence du cas d'utilisation « authentification »	p46
Figure. III.22. Diagramme de séquence du cas d'utilisation « changer le passwoed » ...	p47
Figure. III.23. Diagramme de séquence du cas d'utilisation« ajouter compte utilisateur»	p48
Figure. III.24. Diagramme de séquence du cas d'utilisation « bon de transport »	p49
Figure. III.25. Diagramme de séquence du cas d'utilisation « supprimer commande » ..	p50
Figure. IV.26.Interface Apache Tomcat	p55
Figure. IV.27.Interface Hibernate Framework	p56
Figure. IV.28.Interface Dremweaver	p57
Figure. IV.29.Interface EesyPHP	p58
Figure. IV.30.Interface PHP Myadmin	p59
Figure. IV.31.Interface Netbeans	p60
Figure. IV.32.page d'accueil	p71
Figure. IV.33.Espace administrateur	p73
Figure. IV.34.Changer password	p72
Figure. IV.35.Consulter bon de transport	p73
Figure. IV.36.Supprimer dai	p73
Figure. IV.37.ajouter de bon de commande	p74
Figure. IV.38.Interface Imprimer la facture	p75

Liste des tableaux

Tableau II.1 :situation informatique des unités del'entreprise	p27
Tableau II.2 :description des flux d'information	p30

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 importation.

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 II :

*L'ORGANISME
D'ACCUEIL*

I.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 deux approches de l'architecture logicielle (JEE) qui est nécessaire pour la réalisation des applications des entreprises.

I.1.1.Architecture logicielle[1] :

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. Fonctionnalité de qualité logicielle[2]:

Désigne la capacité d'un produit logiciel à fournir les fonctions qui répondent aux besoins formulés et nécessaires quand le logiciel est utilisé dans des conditions spécifiées. Désigne ce que fait le logiciel pour remplir les besoins utilisateurs.

I.1.3.Critères de qualité logicielle :

L'interopérabilité extrinsèque exprime: la capacité du logiciel à communiquer et à utiliser les ressources d'autres logiciels comme par exemple les documents créés par une certaine application.

L'interopérabilité intrinsèque exprime: le degré de cohérence entre le fonctionnement des commandes et des modules à l'intérieur d'un système ou d'un logiciel.

La portabilité : exprime la possibilité de compiler le code source et/ou d'exécuter le logiciel sur des plates-formes (machines, systèmes d'exploitation, environnements) différents .

La compatibilité : exprime la possibilité, pour un logiciel, de fonctionner correctement dans un environnement ancien(compatibilité descendante) ou plus récent (compatibilité ascendante).

La validité : exprime la conformité des fonctionnalités du logiciel avec celles décrites dans le cahier des charges.

La vérifiabilité exprime: la simplicité de vérification de la validité.

L'intégrité :exprime la faculté du logiciel à protéger ses fonctions et ses données d'accès non autorisés .

La fiabilité :exprime la faculté du logiciel à gérer correctement ses propres erreurs de fonctionnement en cours d'exécution.

La maintenabilité :exprime la simplicité de correction et de modification du logiciel, et même, parfois, la possibilité de modification du logiciel en cours d'exécution.

La réutilisabilité exprime: la capacité de concevoir le logiciel avec des composants déjà conçus tout en permettant la réutilisation simple de ses propres composants pour le développement d'autres logiciels.

L'extensibilité : exprime la possibilité d'étendre simplement les fonctionnalités d'un logiciel sans compromettre son intégrité et sa fiabilité.

L'efficacité : exprime la capacité du logiciel à exploiter au mieux les ressources offertes par la ou les machines ou le logiciel sera implanté.

L'autonomie : exprime la capacité de contrôle de son exécution, de ses données et de ses communications.

La transparence : exprime la capacité pour un logiciel de masquer à l'utilisateur (humain ou machine) des détails inutiles à l'utilisation de ses fonctionnalités.

La composabilité : exprime la capacité pour un logiciel de combiner des informations provenant de sources différentes .

La convivialité : décrit la facilité d'apprentissage et d'utilisation du logiciel par les usagers.

I.2. L'architecture client /serveur [3]:

I.2.1. Introduction :

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 conseil 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 micro-ordinateur.

I.2.2. Définition de l'architecture client-serveur:

L'architecture client-serveur est un modèle de fonctionnement logiciel qui peut se réaliser sur tout type d'architectures matérielle (petites ou grosses machines), à partir du moment où ces architectures peuvent être interconnectées.

On parle de fonctionnement logiciel dans la mesure où cette architecture est basée sur l'utilisation de deux types de systèmes, à savoir un système serveur et un système client s'exécutant normalement sur deux machines différentes. L'élément important dans cette architecture est l'utilisation de mécanismes de communication entre les 2 applications.

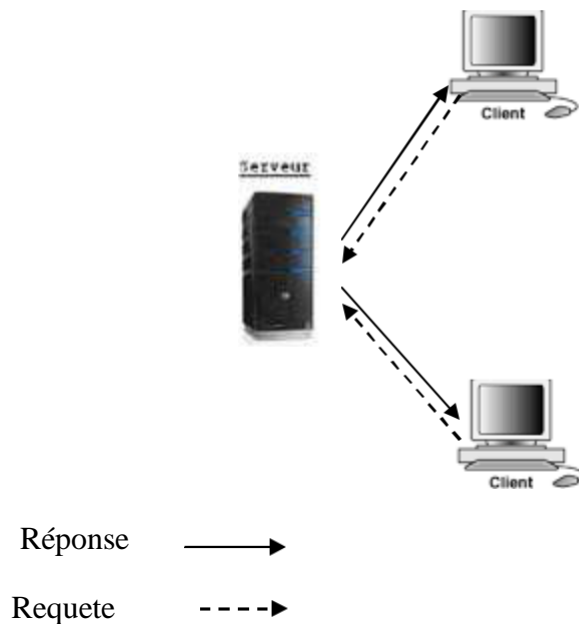


Figure.I .1.Architecture client/serveur.

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

Serveur : processus accomplissant une opération sur demande d'un client, et lui transmettant le résultat.

Requête (Request) c'est: un message transmis par un client à un serveur décrivant l'opération pour le code d'un client.

Réponse (reply) : c'est un message transmis par un serveur à un client à l'exécution d'une opération contenant des paramètres de retour de l'opération

I.2.3. fonctionnement :

Le client émet une requête vers le serveur grâce à son adresse IP et le port, qui désigne un service particulier du serveur.

Le serveur reçoit la demande et répond à l'aide de l'adresse de la machine cliente et son port
Un système client/serveur fonctionne selon le schéma suivant :

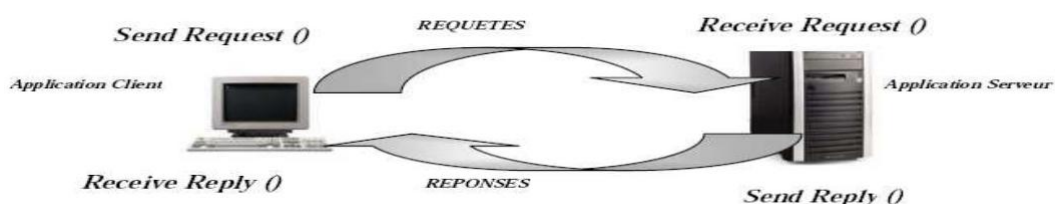


Figure. I.2.fonctionnement de modèle client/serveur.

Les appels aux services de transport mis en jeu sont au nombre de quatre :

Sendrequest () : permet au client d'émettre le message décrivant la requête à une adresse correspondant aux porte d'écoute du serveur.

Receiverrequest () : permet au serveur de recevoir la requête sur sa porte d'écoute.

Sendreply () : permet au serveur d'envoyer la réponse sur la porte d'écoute du client.

Receiverreply () : permet au client de recevoir la réponse en provenance du serveur.

I.3. Les différents modèles de client-serveur:

En fait, les différences sont essentiellement liées aux services qui sont assurés par le serveur. On distingue couramment:

I.3.1. Le modèle client/serveur de données :

Le client /serveur de données est l'exemple encore très répandu de mise en œuvre du client/serveur. Le client effectue la gestion du dialogue, la validation des saisies, la mise en forme des résultats et les traitements (y compris la manipulation des données).

Le serveur gère les données et de plus en plus souvent l'intégrité des données. Il est appelé

serveur de données. Cette mise en œuvre est très répandue car elle a été popularisée par les SGBD relationnels qui ont reposé dès leur origine sur le modèle client/serveur pour ce qui est de leur fonctionnement réparti.

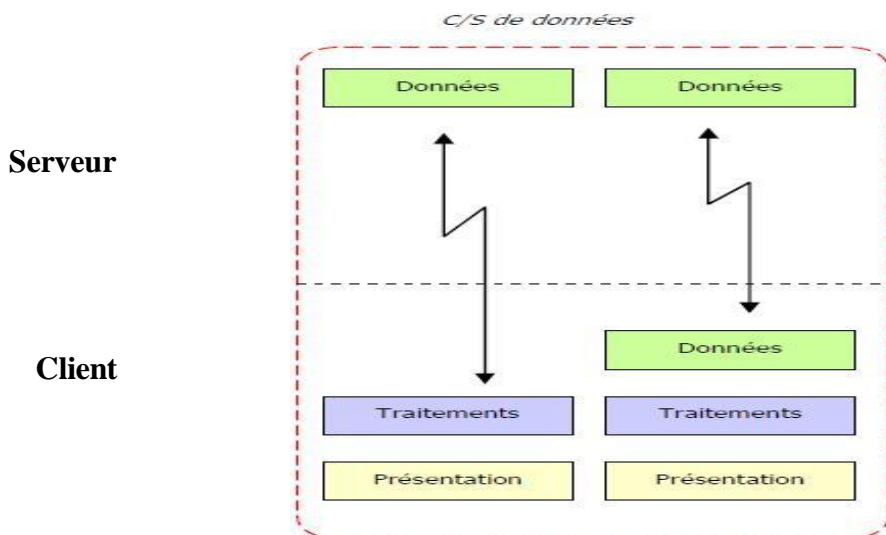


Figure.I.3. le modèle client/serveur de données.

Avantage :

- Interface conviviale,
- Partage des données, réutilisabilité des données stockées sur des gros systèmes.

Inconvénients :

- Trafic réseau encore assez important.

I.3.2 . Le modèle client/serveur de présentation :

Dans ce cas la présentation des pages affichées par le client est intégralement prise en charge par le serveur. Cette organisation présente l'inconvénient de générer un fort trafic réseau

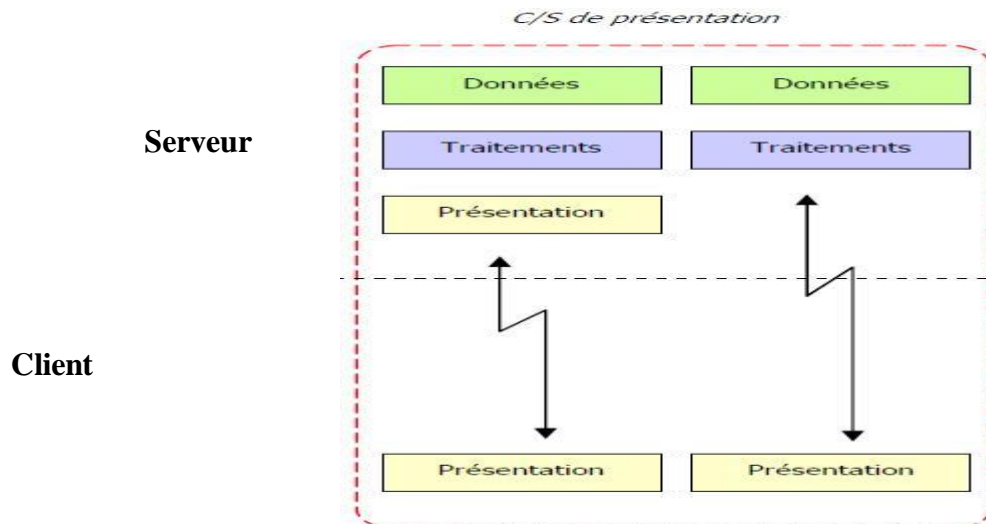


Figure .I.4.le modèle client/serveur de présentation.

I.3.3. Le modèle client/serveur de traitement :

Dans ce cas, le serveur effectue des traitements à la demande du client. Il peut s'agir de traitement particulier sur les données, de vérification de formulaire de saisie, de traitements d'alarmes.

Ces traitements peuvent être réalisés par des programmes installés sur des serveurs mais également intégrés dans des bases de données (triggers, procédures stockées), dans ce cas, la partie donnée et traitement sont intégrés.

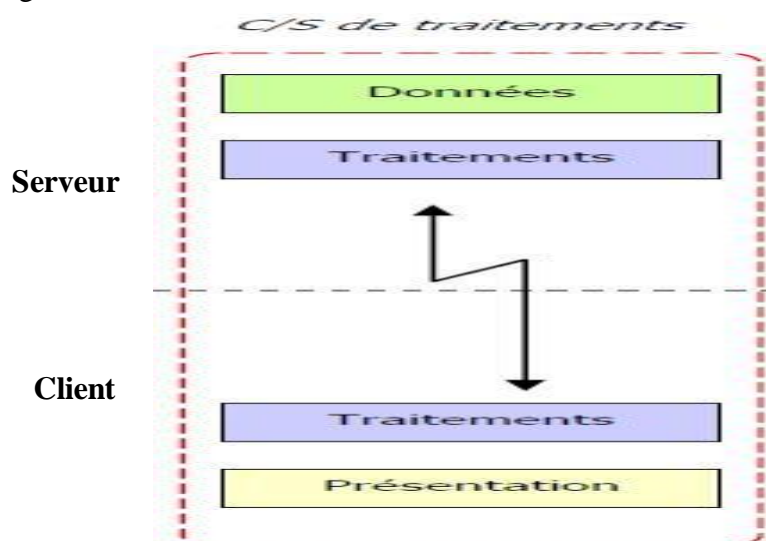


Figure .I.5.le modèle client/serveur de traitement.

Avantage :

les données et les traitements sont idéalement répartis pour équilibrer la charge des serveurs et des clients,
 les données sont proches des utilisateurs,
 le système global reste cohérent malgré l'hétérogénéité des ressources.

Inconvénients :

mise en œuvre plus complexe.

I.4.Définition du Middleware[4] :

On appelle middleware(ou logiciel médiateur en français), littéralement « élément du milieu », l'ensemble des couches réseau et services logiciel qui permettant les dialogues entre les clients et les serveurs. il est souvent hétérogène, en d'autre terme il constitue l'ensemble des services logiciels construits au-dessus du protocole de transport afin de permettre l'échange Requête-réponse de manière transparente en cachant l'hétérogénéité des composants mis en jeu(SGBD, réseau...).

L'objectif de l'ensemble des services disponibles sur le réseau, afin de rendre l'utilisation de ces derniers presque transparente.

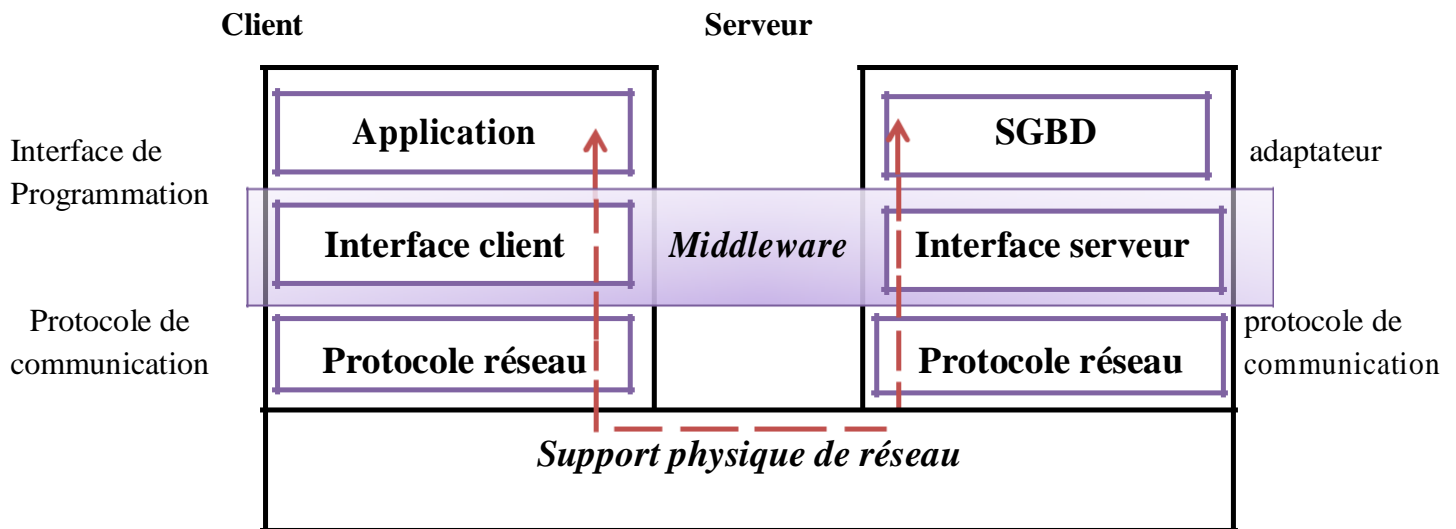


Figure .I.6.la structure d'un Middleware.

I.4.1.Fonctions d'un middleware :

- procédure de fermeture de connexion
- initiation des processus sur différents sites
- services de répertoire (nommage)
- accès aux données à distance
- gestion des accès concurrents
- sécurité et intégrité
- monitoring
- termination des processus
- mise en cache des résultats
- mise en cache des requêtes

I.5.Types de client [5] :**I.5.1.Client léger:**

Le terme « **client léger** » (parfois « client pauvre », en anglais « thin client »), par opposition au client lourd, désigne une application accessible via une interface web (en HTML) consultable à l'aide d'un navigateur web, où la totalité de la logique métier est traitée du côté du serveur. Pour ces raisons, le navigateur est parfois appelé **client universel**.

L'origine du terme lui-même provient de la pauvreté du langage HTML, qui ne permet de faire des interfaces relativement pauvres en interactivité, si ce n'est pas le biais du langage javascript.

Le fait que l'essentiel des traitements soit réalisé du côté du serveur et que l'interface graphique est envoyée au navigateur à chaque requête permet une grande souplesse de mise à jour. En contrepartie, l'application doit s'affranchir des différences d'interprétation du code HTML par les différents navigateurs et l'ergonomie de l'application possède un champ réduit.

I.5.2.Client lourd:

Le terme « **client lourd** » (en anglais « fat client » ou « heavy client »), par opposition au client léger, désigne une application cliente graphique exécutée sur le système d'exploitation de l'utilisateur. Un client lourd possède généralement des capacités de traitement évoluées et peut posséder une interface graphique sophistiquée. Néanmoins, ceci demande un effort de développement et tend à mêler la logique de présentation (l'interface graphique) avec la logique applicative (les traitements).

Ce type d'application étant généralement installé sur le système d'exploitation de l'utilisateur, une nouvelle version doit être installée afin de la faire évoluer. Pour y remédier, les éditeurs d'applications lourdes les dotent généralement d'une fonctionnalité exécutée au lancement de l'application, permettant de vérifier sur un serveur distant si une version plus récente est disponible et le cas échéant propose à l'utilisateur de la télécharger et de l'installer.

I.5.3 Client riche:

Un « **client riche** » est un compromis entre le client léger et le client lourd. L'objectif du client riche est donc de proposer une interface graphique, décrite avec une grammaire de description basée sur la syntaxe XML, permettant d'obtenir des fonctionnalités similaires à celles d'un client lourd (glisser déposer, onglets, multi fenêtrage, menus déroulants).

Les clients riches permettent ainsi de gérer l'essentiel des traitements du côté du serveur. Les données sont ensuite transmises dans un format d'échange standard utilisant la syntaxe XML (SOAP, XML-RPC), puis interprétées par le client riche.

I.6.Types de serveurs pour le système client/serveur :

I.6.1. Serveurs fichiers : dans le cas de serveurs de fichiers, le client requiert des enregistrements de fichiers en émettant des requêtes au serveur de fichiers. Les serveurs de fichiers sont utiles pour partager des fichiers sur un réseau et ils sont indispensables pour créer des banques de documents, d'images.....etc.

I.6.2. Serveurs de base de données: dans le cas de serveurs de bases de données, le client émet des requêtes SQL sous forme de message en direction du serveur. Le résultat de chaque requête est renvoyé au client. Les données ainsi que le code qui traite les requêtes, résident sur la même machine (serveur).

I.6.3.Serveurs de transactions : dans ce modèle, les clients invoquent des procédures distantes résidentes sur le serveur qui comporte en moteur de base de données SQL. Ces procédures exécutent un ensemble d'instruction SQL, l'échange sur le réseau consiste en un seul message de requête/réponse (une réponse pour un bloc de requête SQL).

I.6.4. Serveur de groupware : le groupe s'intéresse à la gestion d'informations semi structurées telles que le texte, l'image, le courrier, la messagerie et l'ordonnancement des tâches. Ces systèmes client/serveur mettent les utilisateurs en contact direct les uns avec les autres. **Microsoft Exchange** est un exemple de ce type.

I.6.5. Serveur d'application objet : dans ce type de serveur, l'application client/serveur est décrite sous forme d'un jeu d'objets communicants. Les objets clients communiquent avec les objets serveurs au moyen d'un courrier d'objet ou O RB (objet request broker). Le client

invoque une méthode sur un objet distant, l'ORB localise une instance de la classe, appelle la méthode demandé et envoie les résultats à l'objet lientc.

I.6.6. Serveur d'application WEB : l'Internet est la plus grande application client/serveur dite intergalactique, ce nouveau modèle consiste à des clients légers et portables qui communiquent avec de très gros serveurs.

Le serveur WEB par exemple, renvoie des documents lorsque le client les demande par leurs noms. Clients et serveurs communiquent via un protocole de type RPC appelé http (Hyper Texte transmission Protocol).

I.7. Les différents types d'architectures de client/serveur[6]:

L'architecture client-serveur permet à une applicat ion de s'adressé à une application physiquement à distance, à travers un protocole d'é change standardisé, pour lui demander de réaliser une tâche pour son propre compte.

I.7.1.Présentation de l'architecture à 2 niveaux:

L'architecture à deux niveaux (aussi appelée *architecture 2-tier*, *tier* signifiant *rangée* en anglais) caractérise les systèmes clients/serveurs pour lesquels le client demande une ressource et le serveur la lui fournit directement, en utilisant ses propres ressources. Cela signifie que le serveur ne fait pas appel à une autre application a fin de fournir une partie du service.



Figure.I.7.Présentation de l'architecture à 2 niveaux.

I.7.2. Présentation de l'architecture à 3 niveaux

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

Un client, c'est-à-dire l'ordinateur demandeur de ressources, équipée d'une interface utilisateur (généralement un navigateur web) chargée de la présentation .

Le serveur d'application (appelé également **middleware**), chargé de fournir la ressource mais faisant appel à un autre serveur .

Le serveur de données, fournissant au serveur d'application les données dont il a besoin.

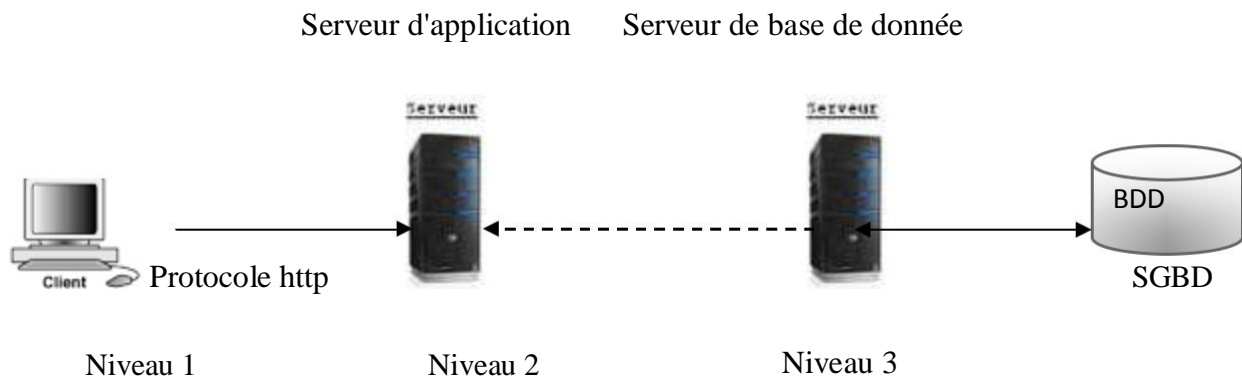


Figure.I.8. Présentation de l'architecture à 3 niveaux

Etant donné l'emploi massif du terme d'architecture à 3 niveaux, celui-ci peut parfois désigner aussi les architectures suivantes :

Partage d'application entre client, serveur intermédiaire, et serveur d'entreprise ;

Partage d'application entre client, serveur d'application, et serveur de base de données d'entreprise.

Comparaison des deux types d'architecture

L'architecture à deux niveaux est donc une architecture client/serveur dans laquelle le serveur est polyvalent, c'est-à-dire qu'il est capable de fournir directement l'ensemble des ressources demandées par le client.

Dans l'architecture à trois niveaux par contre, les applications au niveau serveur sont délocalisées, c'est-à-dire que chaque serveur est spécialisé dans une tâche (serveur web/serveur de base de données par exemple). L'architecture à trois niveaux permet :

Une plus grande flexibilité/souplesse ;

Une sécurité accrue car la sécurité peut être définie indépendamment pour chaque service, et à chaque niveau ;

De meilleures performances, étant donné le partage des tâches entre les différents serveurs.

I.7.3. L'architecteur multi niveaux:

Dans l'architecture à 3 niveaux, chaque serveur (niveaux 2 et 3) effectue une tâche (un service) 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...

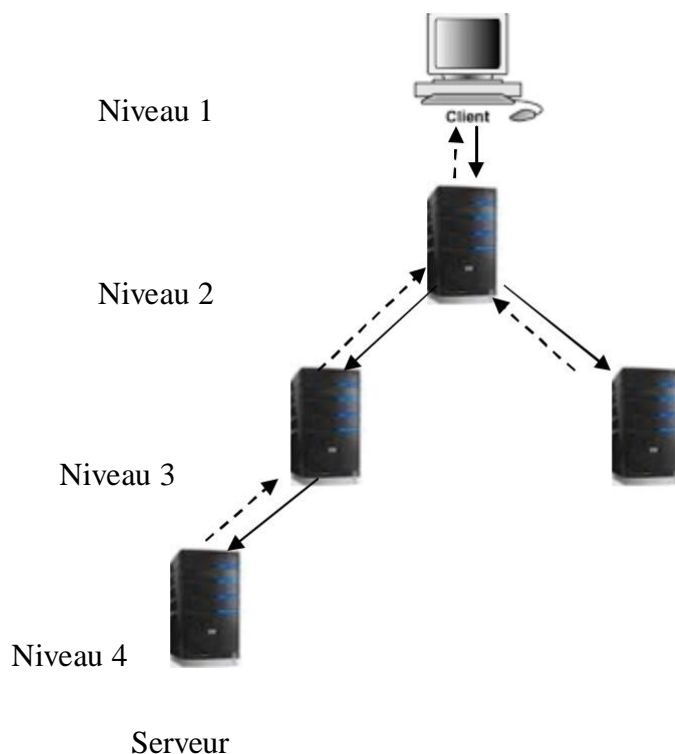


Figure.I.9.L'architecture multiniveaux

I.8.Avantages et inconvénients de l'architecture Client/serveur[7] :

I.8.1. Les avantages de l'architecture client/serveur

Le modèle client/serveur est particulièrement recommandé pour des réseaux nécessitant un grand niveau de fiabilité, ses principaux atouts sont :

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 centralisée, 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 ayant peu d'importance dans ce modèle, ils ont moins besoin d'être administrés

un réseau évolutif: grâce à cette architecture il est possible de supprimer ou rajouter des clients sans perturber le fonctionnement du réseau et sans modification majeure .

I.8.2. Les inconvénients du modèle client/serveur:

L'architecture client/serveur a tout de même quelques lacunes parmi lesquelles :

un coût élevé dû à la technicité du serveur

un maillon faible : le serveur est le seul maillon faible du réseau client/serveur, étant donné que tout le réseau est architecturé autour de lui ! Heureusement, le serveur a une grande tolérance aux pannes (notamment grâce au système RAID).

I.9. Introduction de JavaEE[8] :

Dans le mode 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 portables à travers les différents systèmes utilisés. Sun Microsystems a proposée pour cela la plate-forme Java Entreprise Edition ou Java EE, créée dans le but de faciliter la construction d'applications distribuées, et a doté le langage Java d'une plate-forme logicielle fiable pour répondre aux besoins des entreprises.

I.9.1. Définition de JavaEE :

Java Entreprise Edition (JEE) est une spécification pour le langage de programmation Java de Sun destiné aux applications d'entreprise. JavaEE offre une plate-forme de développement en langage Java pour les applications distribuées à plusieurs niveaux. On parle généralement de « plate-forme JavaEE » pour désigner l'ensemble constitué des services API (Application Programming Interface) offerts et de l'infrastructure d'exécution. Dans la mesure où JavaEE s'appuie entièrement sur Java, il bénéficie des avantages de ce langage, en particulier une bonne portabilité et une maintenance du code. De plus, l'architecture JavaEE repose sur des composants distincts, interchangeables et distribués, ce qui signifie notamment qu'un système reposant sur JavaEE peut posséder des mécanismes de haute-disponibilité, afin de garantir une bonne qualité de service et que la maintenance des applications est facilitée.

I.9.2. L'architecture de JavaEE:[9]

JavaEE 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 : technologies liées à la production des interfaces web dynamiques, par exemple JSP (Java Server Page) et servlet.

Technologie entreprise application : technologies plus directement liées à la logique de business : EJB (Enterprise Java Bean), JavaMail, JMS (Java Message Service), JTA (Java Transaction), etc.

Technologie web services : technologies utiles 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 : technologies liées 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).

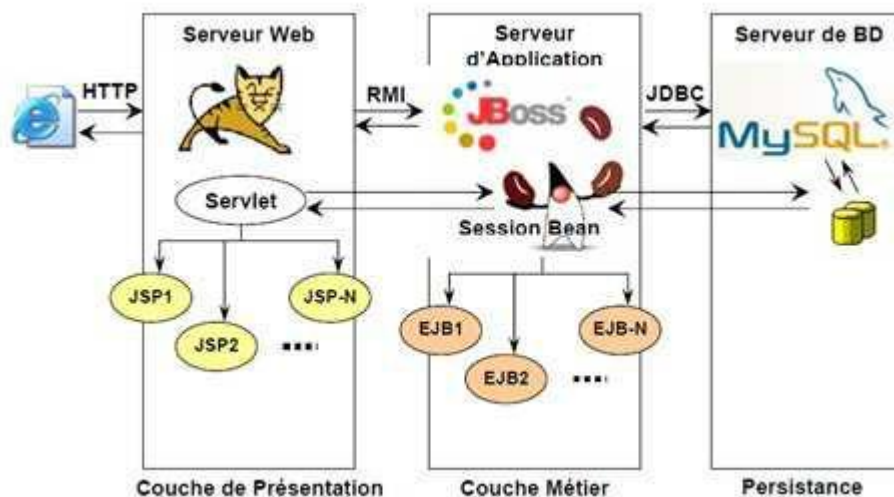


Figure. I.10. l'architecture de JavaEE

Pour 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 data base), 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 entreprises commerciales et les technologies de gestion gèrent tous les processus informationnels assurant la sécurité des transactions.

une application JavaEE est composée principalement de trois couches (pour cela on dit application 3-tiers ou n-tiers) :[10]

Couche présentation : contient les vues que l'utilisateur va interroger soit pour afficher des données ou envoyer des données.

Couche métier : la couche la plus lourde normalement les classes de traitement de données (le langage de l'application).

couche de persistance(ou accès aux données) : la couche basse, c'est elle qui va stocker nos données (texte, XML ,base de données, Excel).

I.10.Les Servelets :[11]**Servlet: organisation de la pile protocolaire**

Les servlets sont des programmes qui aident à la construction d'applications générant des pages Web dynamiques (HTML, XML). Les servlets sont les équivalents Java des scripts CGI qui sont des homologues côté serveur des applets Java situés côté client.

Les servlets sont des classes Java qui mettent en œuvre une interface spécifique et produisent du code HTML en réponse à des demandes GET/POST.

Dans un MOO, un servlet est représenté sous forme de composant et est lié à une classe de servlet qui met en œuvre l'interface requise et assure la mise en œuvre du servlet.

Lorsque vous choisissez le type Servlet pour un composant, la classe servlet appropriée est automatiquement créée, ou bien attachée si elle existe déjà. La classe servlet est initialisée de sorte que les opérations y soient automatiquement ajoutées.

I.11.Les JSP:[12]

JSP(Java Server Pages), c'est un fichier contenant du code HTML et des fragments de code java exécutés sur le moteur de servlet comparable aux langages côté serveur de type PHP, ASP, les pages JSP sont converties en servlet par le moteur de servlet lors du premier appel à la JSP.

Les principales caractéristiques de la technologie JSP sont :

Une langue pour les pages JSP en développement, qui sont des documents textuels qui décrivent comment traiter une demande et construire une réponse

Construit pour accéder à des objets côté serveur

I.12. JDBC:

Java Data Base Connectivity est une API Java permettant d'accéder à des bases de données de façons transparente par rapport à la base de données utilisée. Les procédures de connexion, de la manipulation et d'administration sont les mêmes indépendamment du GSBD : Oracle, Mysql, ODBC ou autre.

I.13.EJB:[13]

Les Enterprise Java Bean ou EJB sont des composants serveurs qui tournent sous un serveur d'application permettant de fournir un certain nombre de services métiers par l'intermédiaire d'un premier type d'EJB, les EJB de session (ou Session Bean). Ils fournissent aussi des services d'accès base de données par l'intermédiaire d'un 2ème type d'EJB : les Beans entité (Entity Beans). Le but des EJB est de faciliter la création d'applications distribuées pour les entreprises. Une des principales caractéristiques des EJB est permettre aux développeurs de se connecter sur les traitements orientés métiers car les EJB est l'environnement dans lequel ils s'exécutent prennent en charge un certain nombre de traitements tel que la gestion des transactions tel que la gestion des transactions, la persistance des données, la sécurité...

I.14. Les services des API de Java EE :

pouvant être classés par catégories :

I.14.1. Les services d'infrastructures :

il en existe un grand nombre, définis ci-dessous:

JDBC (*Java DataBaseConnectivity*) est une API d'accès aux bases de données relationnelles.

JNDI (*Java Naming and Directory Interface*) est une API d'accès aux services de nommage et aux annuaires d'entreprises tels que DNS, NIS, LDAP, etc.

JTA/JTS (*Java Transaction API/Java Transaction Services*) est un API définissant des interfaces standard avec un gestionnaire de transactions.

JCA (*J2EE Connector Architecture*) est une API de connexion au système d'information de l'entreprise, notamment aux systèmes dits «Legacy» tels que les ERP.

JMX (*Java Management Extension*) fournit des extensions permettant de développer des applications web de supervision d'applications.

I.14.2. Les services de communication :

JAAS (*Java Authentication and Authorization Service*) est une API de gestion de l'authentification et des droits d'accès.

JavaMail est une API permettant l'envoi de courrier électronique.

JMS (*Java Message Service*) fournit des fonctionnalités de communication asynchrone (appelées *MOM* pour *Middleware Object Message*) entre applications.

RMI-IIOP est une API permettant la communication synchrone entre objets.

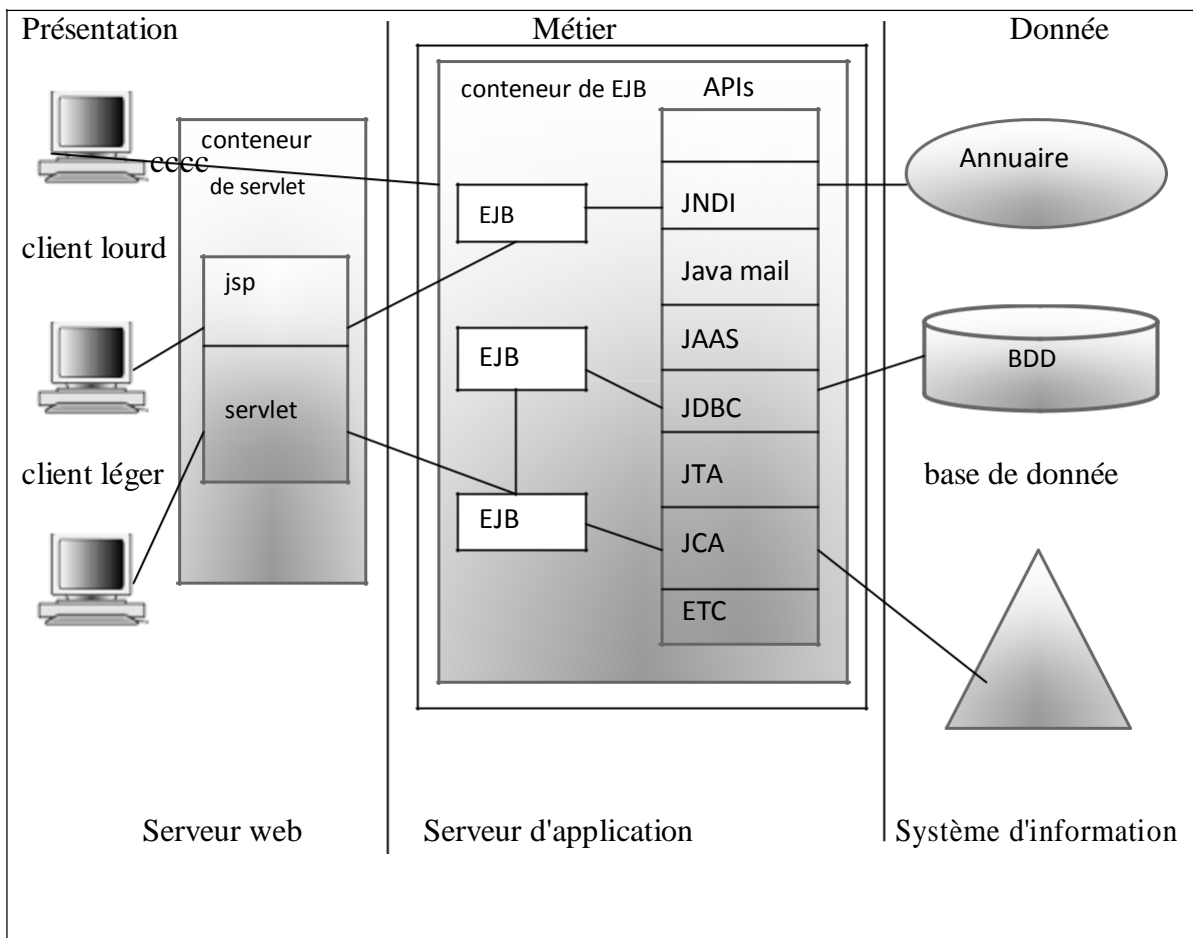


Figure.I.11. Présentation Les services des API de JavaEE

I.15. Les avantages d'utiliser JavaEE :

L'utilisation de JavaEE pour développer et exécuter une application représente plusieurs avantages :

Une architecture d'application basée sur les composants qui permet un découpage de l'application et donc une séparation des rôles lors du développement.

La possibilité de s'interfacer avec le système d'information existant grâce à de nombreuses API : JDBC, JNDI, JMS, JCA...

La possibilité de choisir les outils de développement et les serveurs d'applications utilisés qu'ils soient commerciaux ou libres.

I.16.Conclusion :

Au cours de ce chapitre, nous avons présenté la technologie JavaEE , ,nous avons parlé aussi de l'architecteur client /serveur ,puis nous avons terminé par une description des architecteurs distribués JavaEE, Ces notion seront utilisées pour le développement de notre projet. Dans le chapitre suivant nous allons présenter la conception et réalisation de notre application.

Chapitre I :

*LA TECHNOLOGIE DE
JAVA EE*

II. Introduction:

La présentation de l'organisme d'accueil est une étape importante de l'analyse qui nous permet de prendre connaissance du domaine dans lequel l'organisme souhaite améliorer son fonctionnement. Nous allons commencer par la présentation d'une vue globale sur le domaine d'étude.

II.1. Présentation de l'entreprise:**II.1.1. Situation géographique:**

L'ENIEM (Entreprise Nationale des Industries de l'Électroménager), est une entreprise Publique Economique de droit Algérien (EPE). Son siège social se situe plus exactement à la wilaya de TIZI OUZOU, les unités de productions :froid, cuisson et climatisation sont implantées à la zone industrielle AISSAT IDIR de Oued AISSI, à distance de 10Km de la ville de TIZIOUZOU, la filiale sanitaire est installée à MILIANA, wilaya de AIN DEFLA et la filiale lampe à MOHAMMADIA, wilaya de MASCARA.

II.1.2. Historique :

L'ENIEM est issue de la restructuration organique de la SONELEC (société nationale de fabrication et de montage de matériel électronique et électrique) en 1983. L'entreprise a été chargée de la production et de la commercialisation des produits électroménagers et disposait à sa création de :

Complexe d'appareils ménagers (CAM) de TIZI OUZOU entré en production en juin 1977.

Unité lampe de MOHAMMADIA (ULM) entrée en production en février 1979.

Par la suite elle est devenue une société par action au capital social de 40.000.000 DA en 1989.

Le champ d'activité de l'entreprise ENIEM consiste à la production, le développement, la recherche dans le domaine de l'électroménager, ainsi que la prise en charge de la fonction commerciale, la promotion des exportations et du service après vente. Son siège social est situé au chef lieu de la wilaya de TIZI OUZOU.

Actuellement, l'entreprise, ENIEM est constituée de :

La direction générale.

Unité froid.

Unité cuisson.

Unité climatisation.

Unité prestations techniques (UPT).

Unité commerciale (UC).

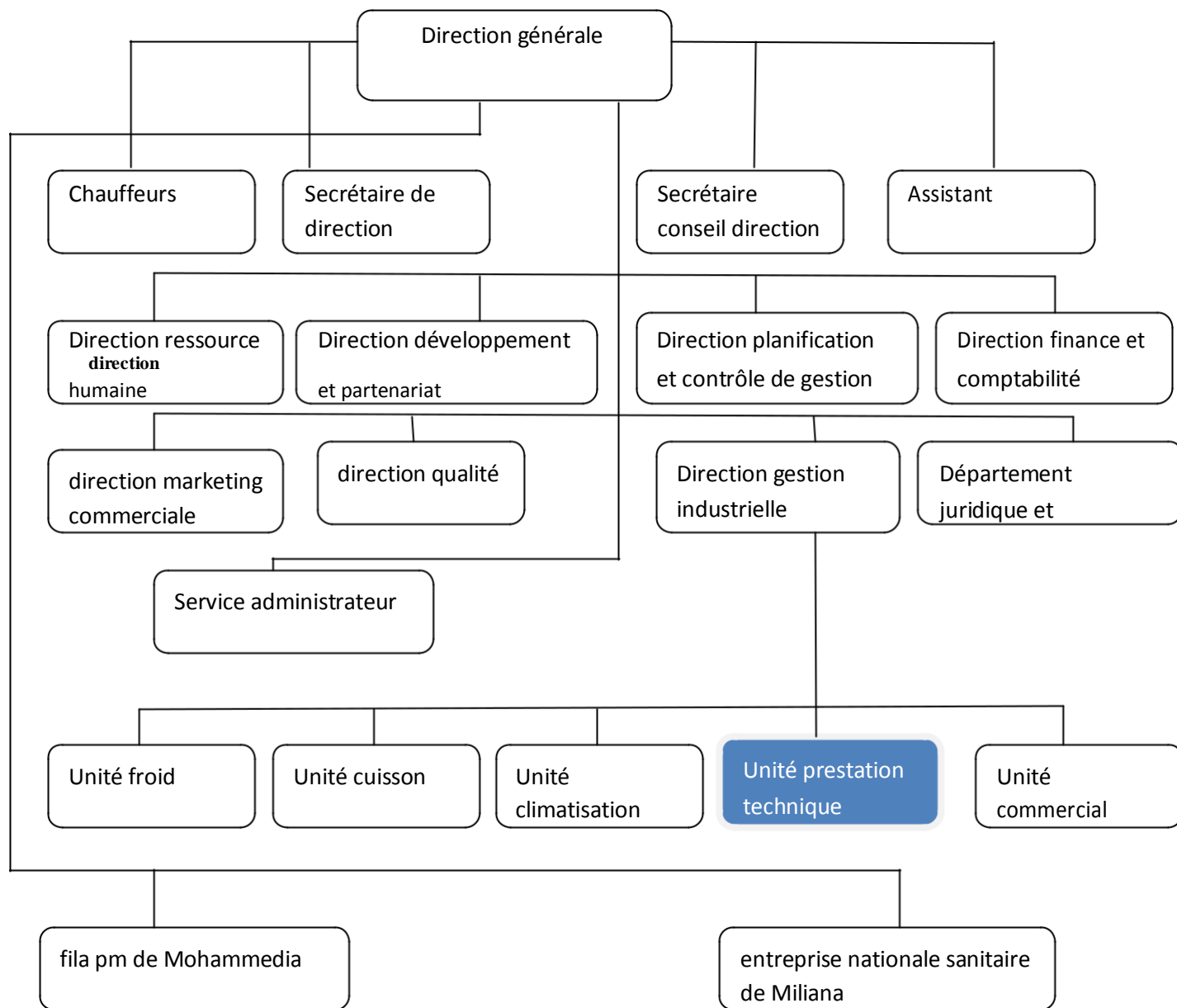
Unité produits sanitaires.

La filiale FILAMP.

Ses unités sont issues de l'ex CAM et sont implantées au niveau de la zone industrielle

AISSAT IDIR (OUED AISSI).

II.1.3. L'organigramme de l'entreprise ENIEM:



Champ d'étude

Unité prestation technique

Figure .II.12: Organigramme générale de l'ENIEM

II.2. Missions et objectifs de l'entreprise:

II.2.1. 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.2.2. 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.3. Mode d'organisation:

II.3.1. Les directions:

II.3.1.1. Direction générale:

La direction générale est l'unique entité qui est responsable de la stratégie et du développement de l'entreprise. Elle exerce son autorité hiérarchique et fonctionnelle sur l'ensemble des directions et unités.



Figure. II.13: Direction général

II.3.1.2. direction planification et contrôle de gestion:

La direction assure le contrôle de gestion, de l'audit financier ainsi que le budget de l'entreprise. Elle assure également les tâches suivantes :

La réalisation et la présentation de tous les travaux permettant de produire une information complète et cohérente des activités de l'entreprise (production, commercialisation, approvisionnement, finances).

L'exploitation et l'analyse des informations relatives aux agrégats de gestion afin de préconiser les actions correctives nécessaires avec toute l'anticipation attendue.

L'exploitation et l'interprétation des résultats des audits financiers, et faire les recommandations nécessaires.

Préparation, établissement et le suivi du budget de l'entreprise.

Planification et organisation du programme annuel d'audit financier ainsi que sa réalisation.

Le contrôle des apports d'activités.

II.3.1.3. Direction de développement et partenariat :

Cette direction assure l'étude et le développement du produit fini ainsi que des actions de partenariat et de sous-traitance. Aussi, elle :

Définit et supervise les actions de développement des produits existants et l'élargissement de la gamme en fonction du marché.

Suit avec la direction industrielle les actions de développement des processus de fabrication et de modernisation de l'outil de production, en vue de l'amélioration de la rentabilité et des conditions de travail .

Participe à la définition de l'organisation de la production dans l'objectif de flexibilité, de réduction des coûts de fabrication.

Définit et concrétise des actions de sous-traitance et de partenariat.

Développe d'autres créneaux pour l'utilisation maximale des capacités technologiques de l'entreprise.

II.3.1.4. Direction des finances et comptabilité:

Cette fonction est auditée au moins une fois par an par un commissaire au compte, sa mission générale est de :

Garant des obligations légales, des règles comptables et des procédures de l'entreprise dont elle vérifie l'application par la mise en œuvre d'un contrôle interne.

Assurer la comptabilisation dans les délais de toutes les opérations permettant l'établissement du bilan, du compte de résultat et des déclarations fiscales de l'entreprise.

Rechercher et mobiliser dans des meilleures conditions de délai et de coût les besoins en ressources financières.

Analyser les équilibres financiers de l'entreprise.

Etudier et met en place la stratégie financière del'entreprise.

Définir la politique bancaire et l'orientation budgétaire.

Gérer la trésorerie (recettes et dépenses) et contrôle les rédactions fiscales périodiques.

Analyser les coûts et les prix de revient et met à la disposition des responsables opérationnels l'information financière nécessaire.

Rédiger, vérifie et approuve les dispositions décrites relatives au fonctionnement efficace de son activité

II.3.1.5. Direction de marketing et communication:

Cette direction assure des politiques commerciales et de communication et les met en œuvre par la conception et l'élaboration des méthodes et outils de gestion nécessaire. Aussi elle :

Conduit les travaux d'études, d'analyses et de synthèse relative aux tendances et évolution des marchés intérieurs et extérieurs.

Elabore en conformité avec la politique commerciale de l'entreprise, toute action concernant les schémas de distribution des produits finis, d'implantation d'antennes de vente au niveau national et international.

Contribue avec les structures concernées de l'entreprise à l'élaboration des plans annuels et pluriannuels de production, de commercialisation et de développement.

Participe à la politique de détermination des barèmes de prix.

Elabore un plan de communication interne et le met en œuvre après approbation de la direction de l'entreprise.

Elabore avec la direction commerciale le plan de communication externe et le met en œuvre après approbation de la direction de l'entreprise.

Etablit les requêtes clients en vue de mesurer eniveau de satisfaction de la clientèle.

Initie et suscite des actions d'amélioration continue de la communication en relation avec la démarche assurance qualité de l'entreprise.

Assure la fonction de porte parole avec l'environnement externe et médiatique de l'entreprise.

Dirige toutes les opérations d'exportation de produit finis vers l'étranger.

II.3.1.6 .Direction industrielle:

La direction industrielle est chargée de développer et de mettre en place les moyens et l'organisation industrielle nécessaire à la réalisation de la production en agissant sur les approvisionnements, les moyennes et les techniques de production. En outre, elle:

Définit les programmes de production en fonction de la demande commerciale et des capacités installées avec le souci de la rentabilité optimale.

Veille à l'optimisation et à l'adéquation des appro visionnements en utilisant au mieux les capacités financières de l'entreprise pour assurer des stocks homogènes et productifs.

Suit la réalisation des programmes de production et préconise des solutions d'adaptation en cas de difficultés.

Améliore la gestion de production en relation avec la structure informatique (GPAO).
Entreprend et suscite des études de modernisation, de renouvellement, d'optimisation et d'installation des moyens de production.

Prend en charge l'industrialisation des nouveaux produits ou modifiés, dans le cadre du développement.

Organise et anime l'industrialisation de produits nouveaux.

Se tient informé des évolutions des techniques de fabrication des appareils électroménagers et étudie avec les unités l'opportunité de leur adoption.

Veille au renforcement des dispositifs de contrôle qualité à tous les stades de la préparation technique, de soutien et de la fabrication des produits et ce, en étroite collaboration avec les responsables qualité.

Définit une politique d'amélioration de la maintenance des équipements de production et en assure le suivi.

II.3.1.7. Direction des ressources humaines:

La direction ressources humaines accroit la mobilisation et la valorisation du personnel dans ses actions aux services du client, Elle :

Pilote le recrutement, l'accueil, l'information et gère le plan de carrière du personnel et les pouvoirs publics en respectant les objectifs de conformité, fiabilité et délais.

Assure la mise en place et l'adaptation des besoins tant quantitatifs que qualitatifs de l'entreprise en matière de ressources humaines et ce, en fonction de son développement et des conjonctures.

Encourage les actions nécessaires à la rationalisation des effectifs et à l'émergence des compétences.

Rédige, vérifie et approuve les dispositions décrites relative au fonctionnement efficace de son activité.

Définit et exécute les plans de formation des besoins de l'entreprise et suivant les niveaux de qualification du personnel.

Gère les éventuels conflits collectifs dans le spectre de la loi et prépare les décisions de direction.

Organise et promouvoit la médecine de travail, gère l'administration du siège de l'entreprise.

En plus des directions centrales, l'ENIEM est composé de quatre unités de production (unité froid, unité cuisson, unité climatisation), d'une unité commerciale et d'une unité de prestations techniques ainsi d'une filiale (FILAMP) sise à MOUHAMMADIA.

II.4. Les unités : L'activité de l'ENIEM sera concentrée sur la fabrication de réfrigérateurs, cuisinières, et climatiseurs. Cette activité sera assurée par plusieurs unités de production:

II.4.1. Unité froid : Elle est chargée de la production et le développement des produits froid

II.4.2. Unité cuisson :

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

- La transformation des tôles
- Assemblage

II.4.3. Unité climatisation: Les capacités existantes sont de 60.000 climatiseur 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

II.4.4. Unité commerciale:

Ses activités sont : La distribution et l'exploitation des produits ENIEM

Le service après – vente (à travers ses moyens propres et un réseau d'agents agréés).

II.4.5. Unité prestations techniques:

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

Réparation des outils de moules

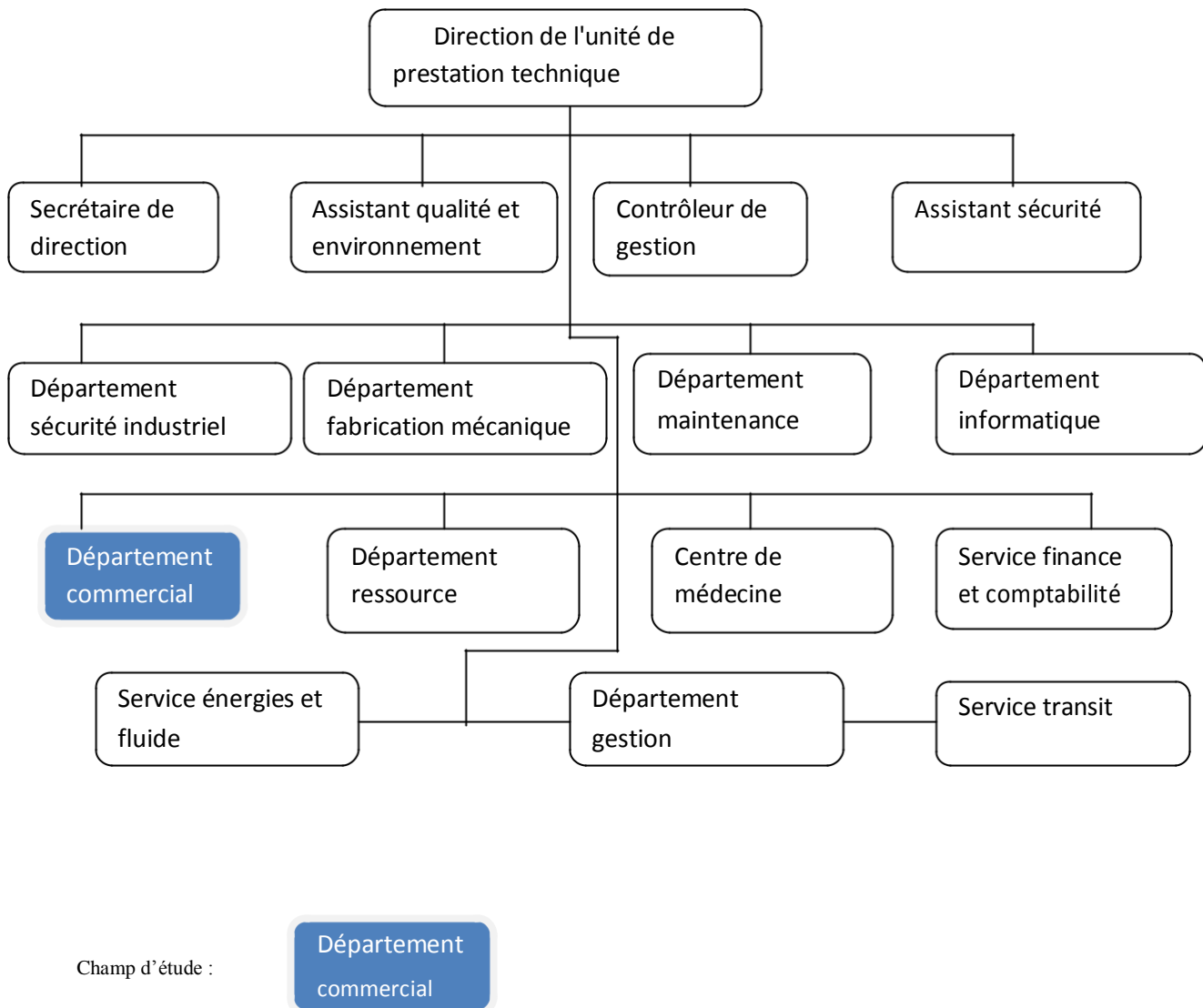
Conception et réalisation d'outillages

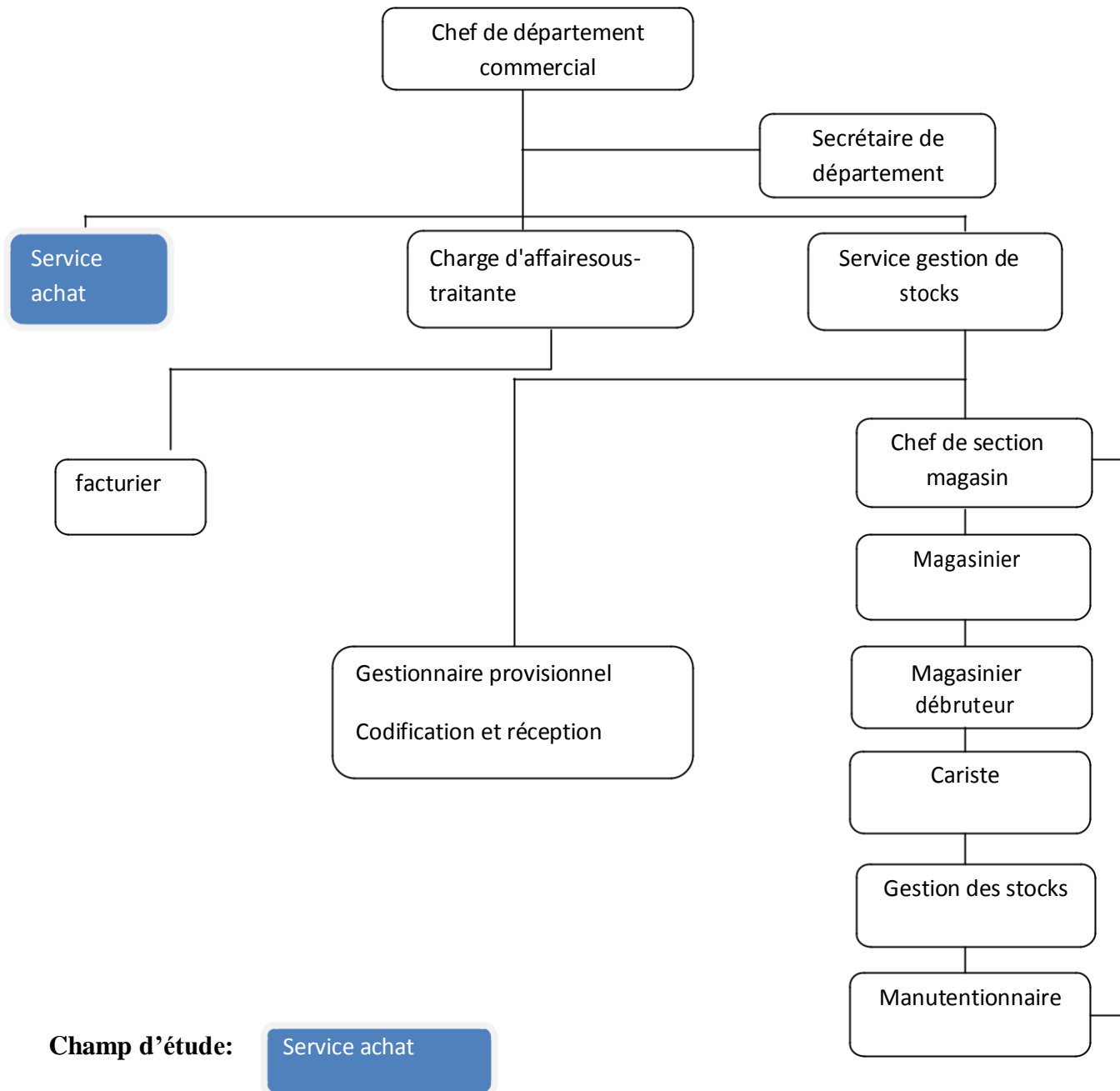
Fabrication de pièces de rechange mécanique

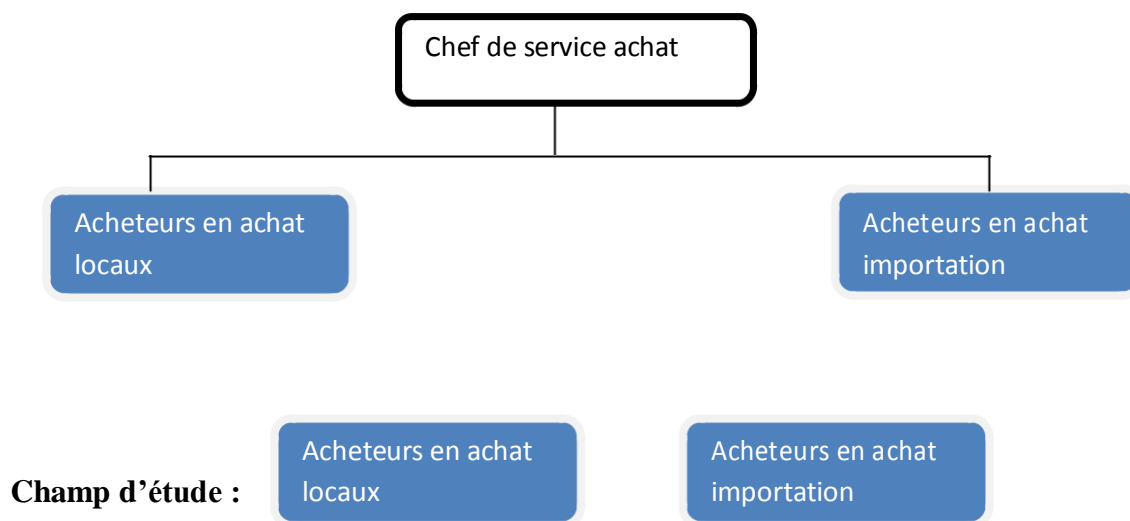
Gestion des énergies et fluides

Gardiennage et sécurité

Travaux d'imprimerie

II.5. Organigramme de l'unité de prestation technique:**Figure.II. 14:Organigramme de l'unité de prestation technique**

II.6. Organigramme de département commercial:**Figure.II.15: Organigramme de département commercial**

II.7. Organigramme de service d'achat:**Figure.II.16: Organigramme de service d'achat:****II.8. Situation informatique des unités de l'entreprise:**

Unité	Matériel	Type
Unité froid	24 pc 13 terminaux 04 pairs modems 03pairs multiplexeurs	PIV et PIII HP D 330 et 700/92A TRT 2334 A 2934 A et 2563 B
Unité climatisation	06 PC 02 Terminaux 02 Imprimantes	PIV 700/92 A 2934A
Unité commerciale	02 PC 02 Pairs modems	PIII TRT
Unité prestation technique	27PC 05 Terminaux 01 Pair modem	PIII et PIV 700/92 A TRT
Unité cuisson	09PC 05 Terminaux 03 Pairs modems	PIV 700/92A TRT

	04 multiplexeurs 03 Imprimantes	2334A 2934A
--	------------------------------------	----------------

Tableau. II.1: Situation informatique des unités de l'entreprise

II.9.Etude d'existante:

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

Cette étape doit être faite avec sérieux en s'appuyant sur: interviews des personnes impliquées (modes opératoires dysfonctionnements, souhaits...).Récupération des documents et leur circulation.

Dans cette partie on va étudier le domaine et essayer d'améliorer son fonctionnement.

II.9.1Diagramme de flux :

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

II.9.2. Concepts du diagramme:

L'analyse des flux s'exprime avec deux concept quisont:

Acteur:

Un acteur est un émetteur ou un récepteur d'un fluxd'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 transforman t l'information et en renvoyant un ou plusieurs autres flux d'information à d'autres acteurs. Les a cteurs sont représentés par leur rôle 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 ont des échanges avec les acteurs internes dans le cadre de l'activité étudiée.

Acteur
externe

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

Acteur
interne

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.

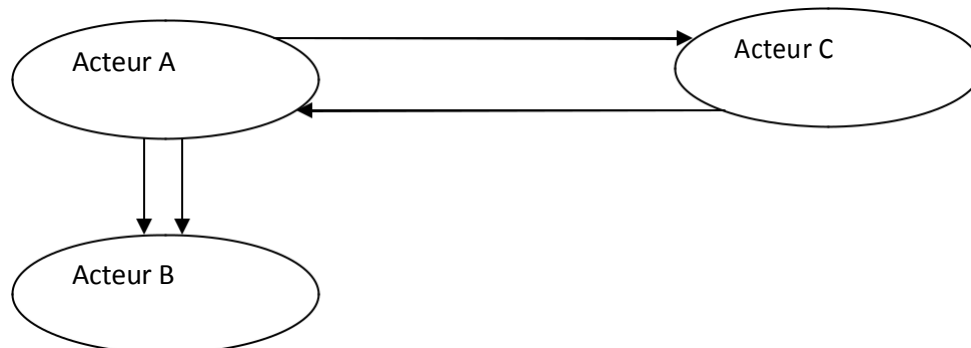
Flux interne: il décrit l'échange d'informations entre deux acteurs interne du le champ d'étude.

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

Formalisme graphique :

F3

F1F2F4



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.

II.9.3. présentation de diagramme de flux:

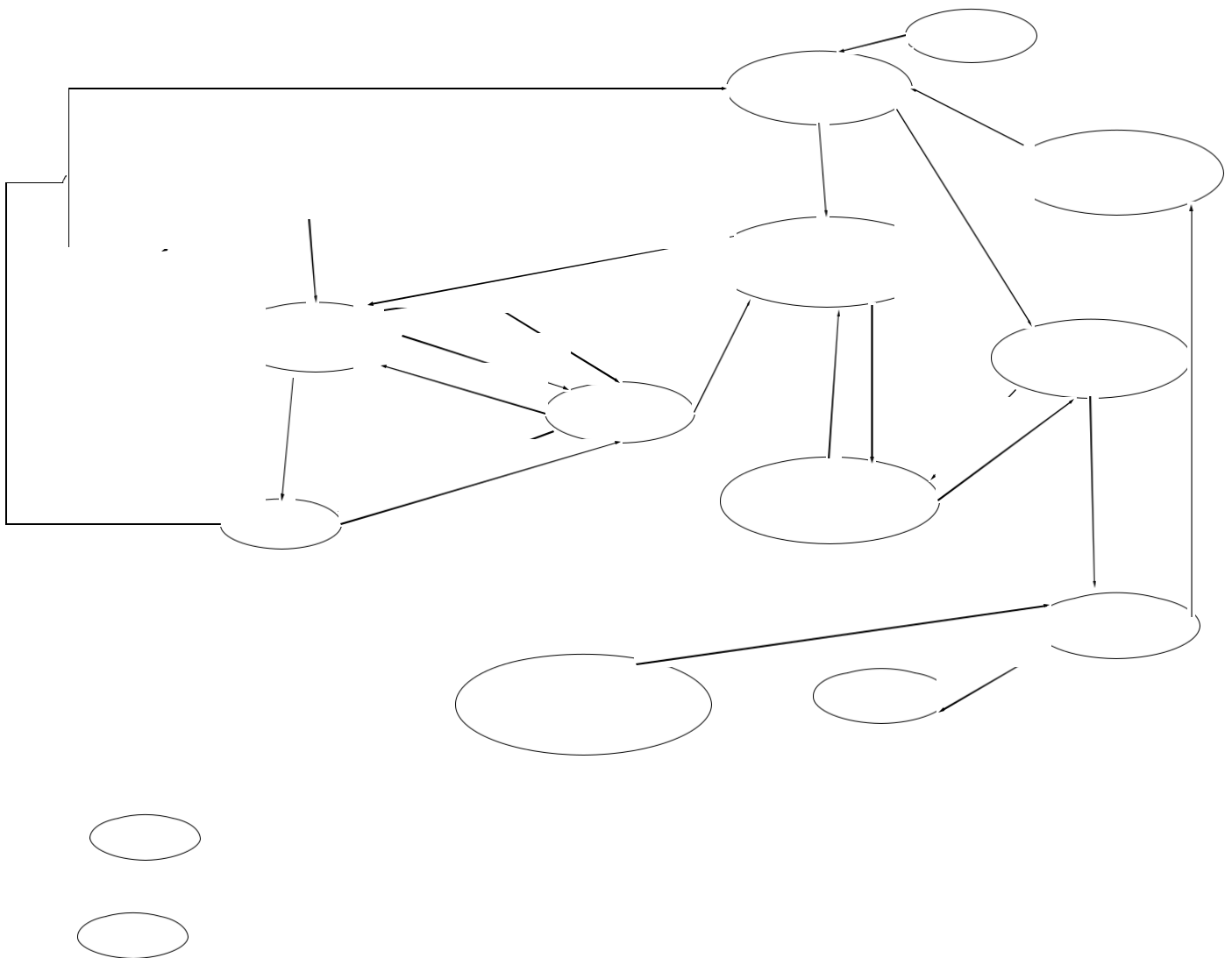


Figure .II.17: Diagramme de flux

II.9.4. Description des flux d'information:

N° Flux	Description des Flux d'information
01	DAI
02	Liste de fournisseurs
03	Demande d'offre + liste de fournisseur à consulter
03*	Demande d'offre signée

04	Demande d'offre
04*	Facture proformat
05	TCO
06	TCO + rapport de présentation
07	Facture pro forma 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 forma, 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, connaissance, avis d'arriver)
28	Montant à payer
29	Chèques
30	Arrivée des marchandises
31	Bulletin de réception
32	Bulletin de réception

Tableau. II.2.Description des flux d'information

II.9.5.Critiques de l'existant et suggestions: L'étude de l'existant nous a conduits à noter quelque insuffisance 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 avons classé en trois catégories:

Insuffisances d'ordre organisationnel : elles concernent généralement les postes de travail, classement et archivage de documents.

Insuffisances d'ordre informationnel: elle concernent généralement les rubriques des documents utilisés par les services du département d'achats de L'ENIEM.

Insuffisances d'ordre technique: elle concernent en général les outils et le matériel informatique utilisé par les services du département d'achat de L'ENIEM.

II.10. Les Niveaux de l'entreprise :

II.10.1. Niveau organisationnel:

Critique 1: Détérioration des documents due aux mauvaises conditions. Les informations sont en effet sur support papier et la manipulation fréquente du papier finit par ce détériorer.

Suggestion 1 : Installation d'une base de données qui permet une recherche plus rapide des informations, leur protection et leur consultation à tout en gardant les documents portant des signatures.

Critique 2 : toutes les tâches se font manuellement ce qui provoque un retard dans la recherche et la manipulation de l'information.

Suggestion 2 : l'outil informatique permettra une grande rapidité dans la recherche et la réalisation des traitements.

Critique 3 : communication limitée entre les services ce qui engendre un retard de transmission de documents.

Suggestion 3 : l'outil informatique permettra le traitement et la manipulation de l'information en temps réel.

Critique 4 : l'information est décentralisée, les services travaillent avec un peu de coopération, ce qui se traduit par des retards.

Suggestion 4 : mettre en place une base de données centralisée accessible par des services en mode contrôlé.

II.10.2. Niveau informationnel:

Critique : mauvaise conception des documents. Par exemple, dans le document "bon de transport" il n'existe pas de rubriques prévues pour : numéro contrat ou bon de commande, numéro de facture .

Suggestion: ajoutez ces rubriques non prévues.

II.10.3. Niveau technique

Critique : les utilisateurs ne sont pas spécialisés en informatique (gestionnaires, financiers....). temps et performance , support papier.

Suggestion : concevoir des logiciels conviviaux et faciles à comprendre même s'ils ne sont pas spécialisés en informatique.

II.10.4. 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 centralisée. Voici l'architecture globale de cette solution

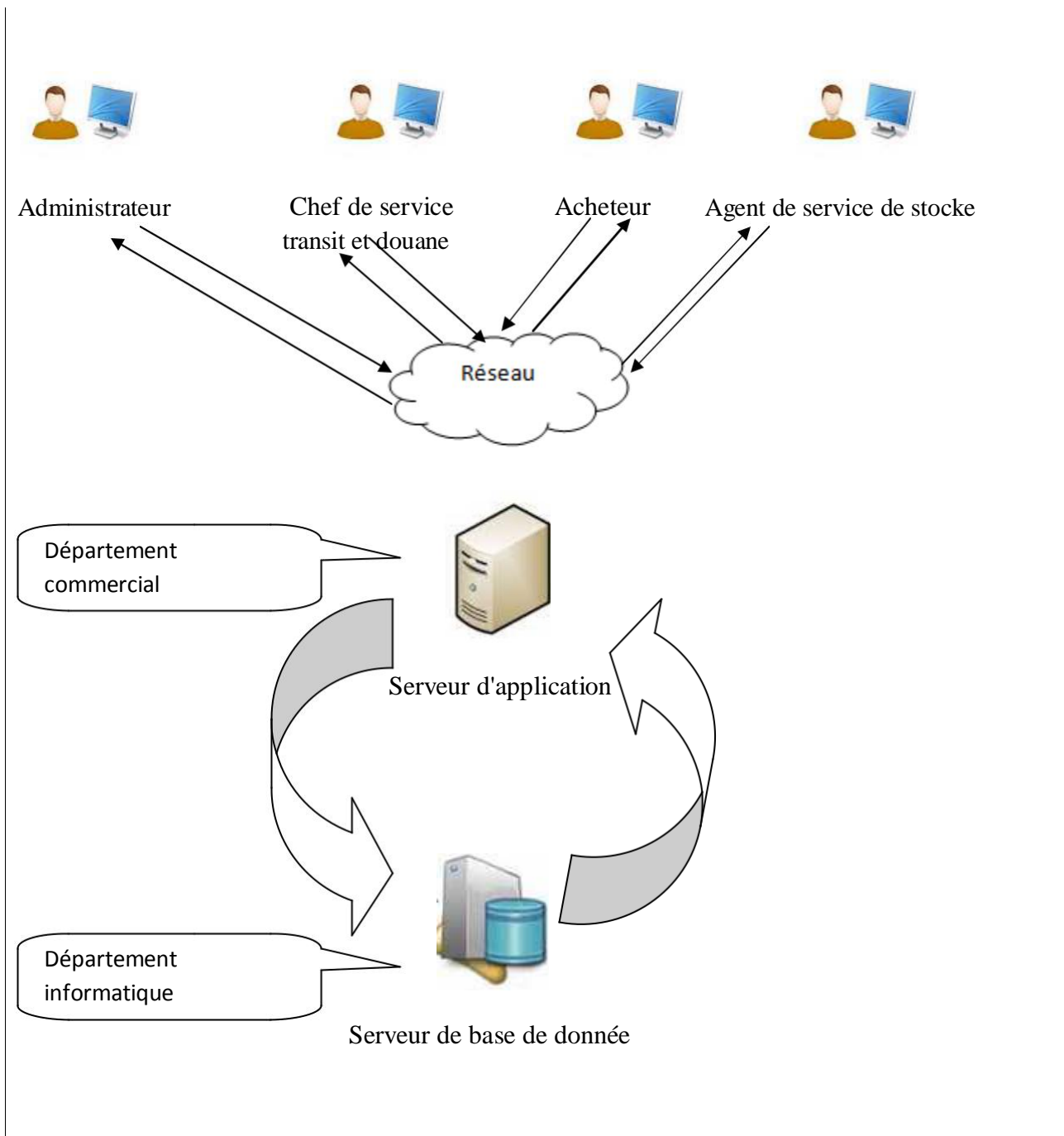


Figure. II.18 : Architecteur globale du modèle proposé

II.11. Conclusion : la présentation de l'organisme d'accueil nous a permis de mettre en évidence notre domaine d'étude .L'étude de ce dernisur le plan information ainsi que le diagramme des flux nous ont permis de mettre en évidence les intervenants, les différents traitements ainsi que les insuffisances et les anomalies qui entravent le bon fonctionnement des services concernés .

Ceci nous a conduit à faire des suggestion ainsi do nc proposer une solution réseau.

Dans le chapitre suivant nous allons présenter l'architecture client/serveur et la technologies de JavaEE.

Chapitre III :

*ANALYSE ET
CONCEPTION*

III. Introduction:

Afin d'aboutir à une meilleure organisation, nous devons tout naturellement avoir recours à un formalisme de conception orientée objet (RUP, Rational UnifiedProcess) qui est une démarche de développement est souvent utilisé conjointement au langage UML qui va nous permettre de comprendre et de décrire les besoin, 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 on va, 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.1.Phase d'Initialisation : une analyse approximative de notre projet permet de d'élaborer les points suivants :

- Notre application touche à la gestion des achats lo caux et importation qui est un domaine connu.
- Notre application est de type Client/serveur, ce genre d'applications a connu déjà un succès considérable et ces dernière se ressemblentd'un point de vu outil .
- Notre application est isolée à internet, le risque de sécurité ne présente pas un énorme problème.

III.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.3. Phase de Construction:

Pour chaque itération réalisée précédemment, nousllonsa suivre un cycle de vie en cascade pour aboutir à un diagramme de classe en passant pa r un diagramme de séquence ou de collaboration

III.4. Phase de Transition:Exécution des bêtas tests ,Déploiement et test de réception .

III.5. 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).

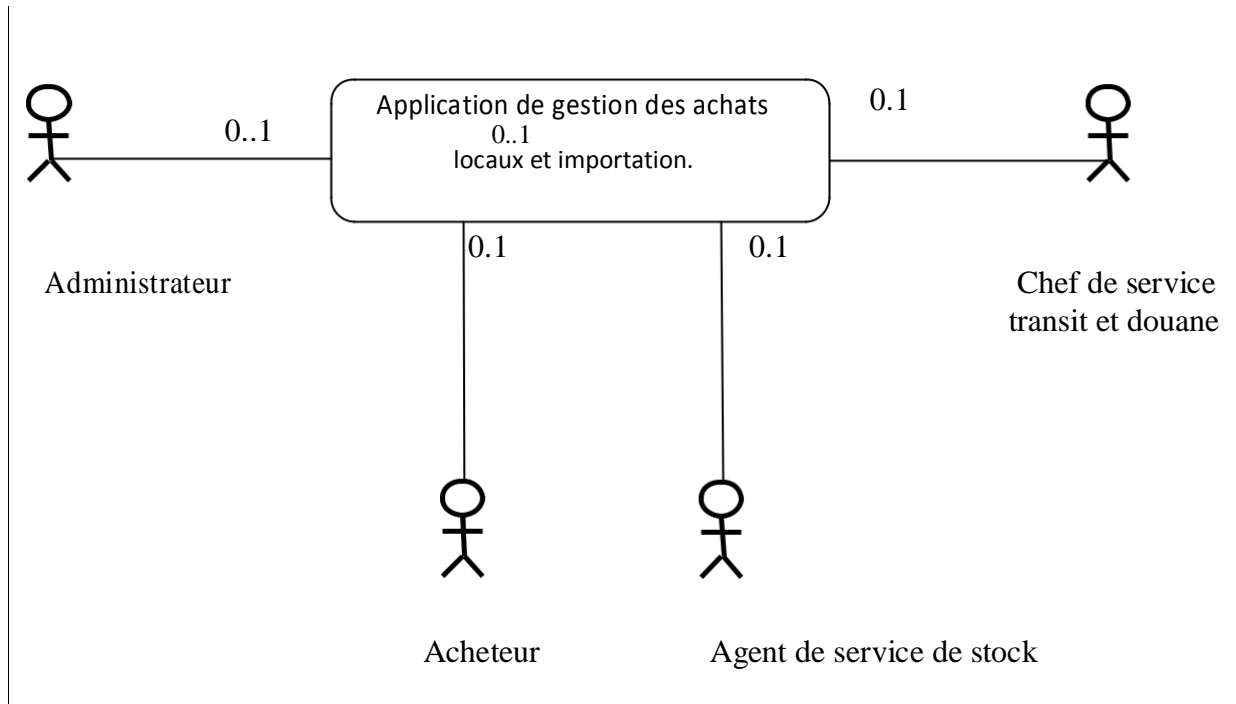
Agent de service stock (ASS) : il s'occupe de la gestion des stocks, des magasins, et encore des emplacements dans ces magasins.

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.5.1. 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 donné par la figure suivante:

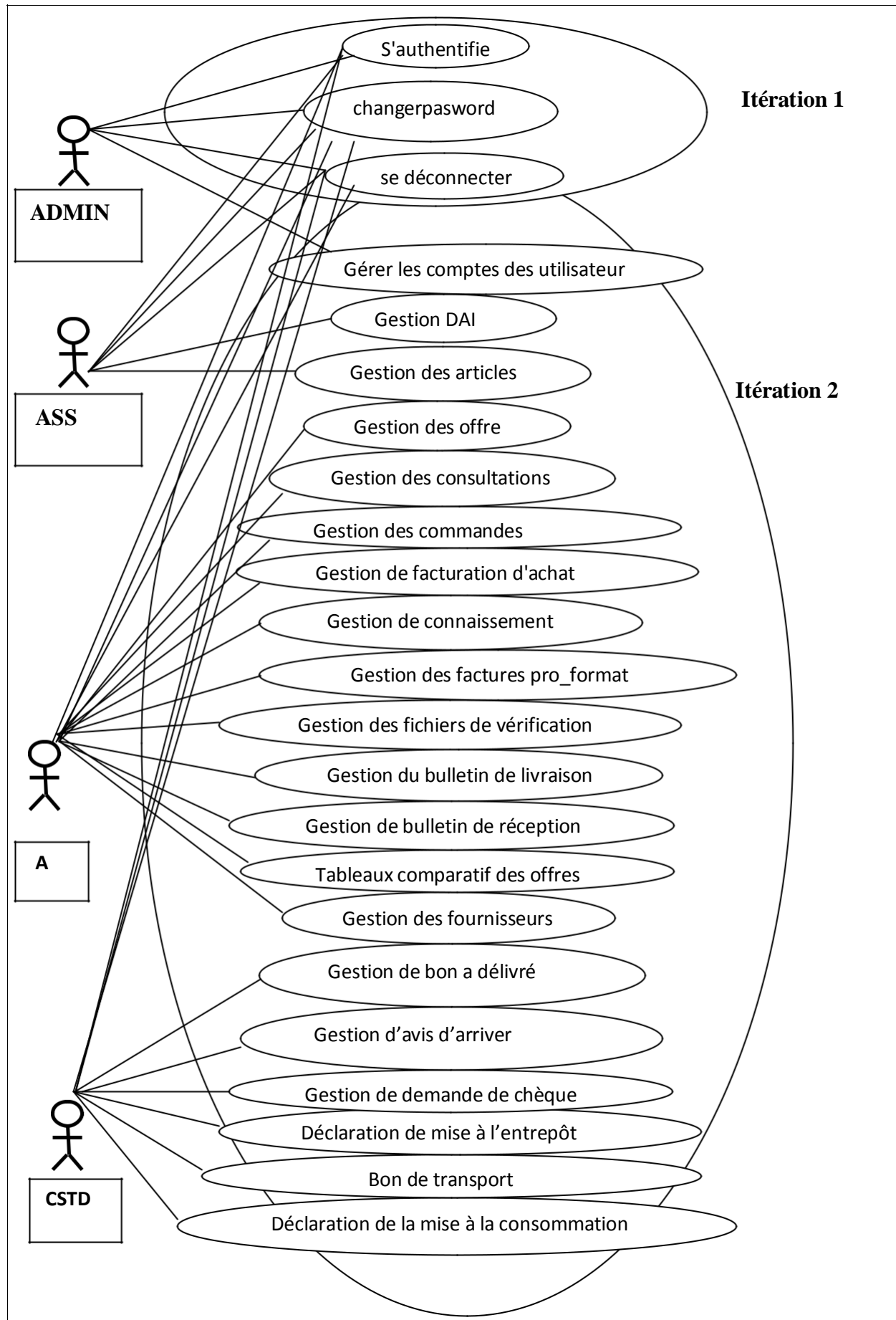


figureIII.19: Diagramme de contexte

III.5.2.Représentation des diagrammes de cas d'utilisation[14] :

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.



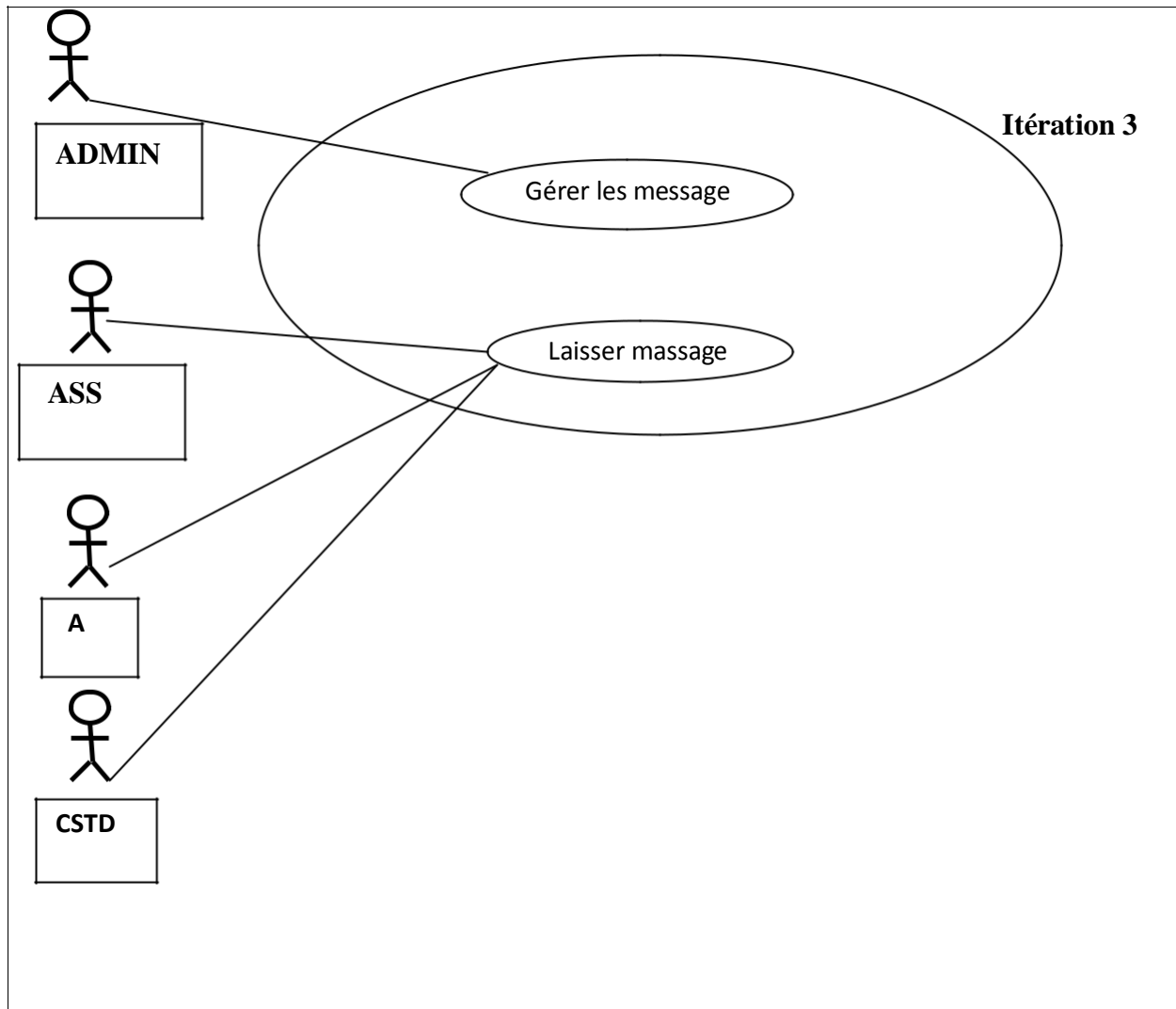


Figure III.20: diagramme globale

III.5.3. Les cas d'utilisations :

Représente un ensemble de cas d'utilisation et d'acteurs et leurs relations. Ils présentent la vue statique des cas d'utilisation d'un système.

Dans ce qui suit nous allons identifier les différents cas d'utilisation.

III.5.3.1. Cas d'utilisation relatifs au chef département commercial (administrateur(ADMN)) :

S'authentifie.

Gérer les comptes utilisateurs :

Ajouter un utilisateur.

Consulter un utilisateur.

Modifier un utilisateur.

Supprimer un utilisateur.

Gestion des
messages. Historique.

Modifier password.

III.3.2.Cas d'utilisation relatif à l'agent de service stock (ASS) :

S'authentifier.

Gestion des DAI.

Ajouter DAI.

Consulter DAI.

Modifier DAI.

Supprimer DAI.

Imprimer DAI.

Gestion des articles.

Ajouter article.

Consulter article.

Modifier article.

Supprimer article.

Imprimer article

Laisser des messages.

Changer password.

III.3.3.Cas d'utilisation relatif à l'Acheteur (A) :

S'authentifie.

Gestion des offres.

Ajouter offre.

Consulter offre.

Modifier offre.

Supprimer offre.

Imprimer offre.

Gestion des consultations .

Ajouter consultation.

Consulter consultation.

Modifier consultation.

Supprimer consultation.

Imprimer consultation.

Gestion des commandes.

Ajouter commande.

Consulter commande.

Modifier commande.

Supprimer commande.

Imprimer commande.

Gestion de facturation d'achats.

Ajouter facture.

Consulter facture.

Modifier facture.

Supprimer facture.

Imprimer facture.

Gestion de facture pro_format.

- Ajouter facture pro_format.
- Consulter facture pro_format.
- Modifier facture pro_format.
- Supprimer facture pro_format.
- Imprimer facture pro_format.

Gestion des fiches de vérification.

- Ajouter fiche de vérification.
- Consulter fiche de vérification.
- Modifier fiche de vérification.
- Supprimer fiche de vérification.
- Imprimer fiche de verification.

Gestion du bulletin de livraison.

- Ajouter un bulletin de livraison.
- Consulter un bulletin de livraison.
- Modifier un bulletin de livraison.
- Supprimer un bulletin de livraison.
- Imprimer un bulletin de livraison.

Gestion fournisseurs.

- Ajouter fournisseur.
- Consulter fournisseur.
- Supprimer fournisseur.
- Imprimer fournisseur.

Gestion des connaissances.

- Ajouter un connaissance.
- Consulter un connaissance.
- Modifier un connaissance.
- Supprimer un connaissance.
- Imprimer un connaissance.

Gestion de bulletin de réception.

- Ajouter un bulletin de réception.
- Consulter un bulletin de réception .
- Modifier un bulletin de réception.
- Supprimer un bulletin de réception.
- Imprimer un bulletin de réception.

Tableau comparatif des offres
(TCO). Ajouter un TCO.

- Consulter un TCO.
- Modifier un TCO.
- Supprimer un TCO.
- Imprimer un TCO.

- Laisser des messages.
- Changer password.

III.3.4.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.

Imprimer 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.

Imprimer un avis d'arriver.

Gestion de demande de chèque.

Ajouter undemande de chèque

Consulter undemande de chèque.

Modifier undemande de chèque.

Supprimer undemande de chèque.

Imprimer undemande de chèque.

Déclaration de mise à la l'entrepôt

. Ajouter un d11.

Consulter un d11.

Modifier un d11.

Supprimer un d11.

Imprimer un d11.

Déclaration de mise à la consommation .

Ajouter un d3.

Consulter un d3.

Modifier un d3.

Supprimer un d3.

Imprimer un d3.

Bon de transport.

Ajouter un Bon de transport.

Consulter un Bon de transport.

Modifier un Bon de transport.

Supprimer un Bon de transport.

Imprimer un Bon de transport.

Laisser messages.

Changer password.

III.6.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

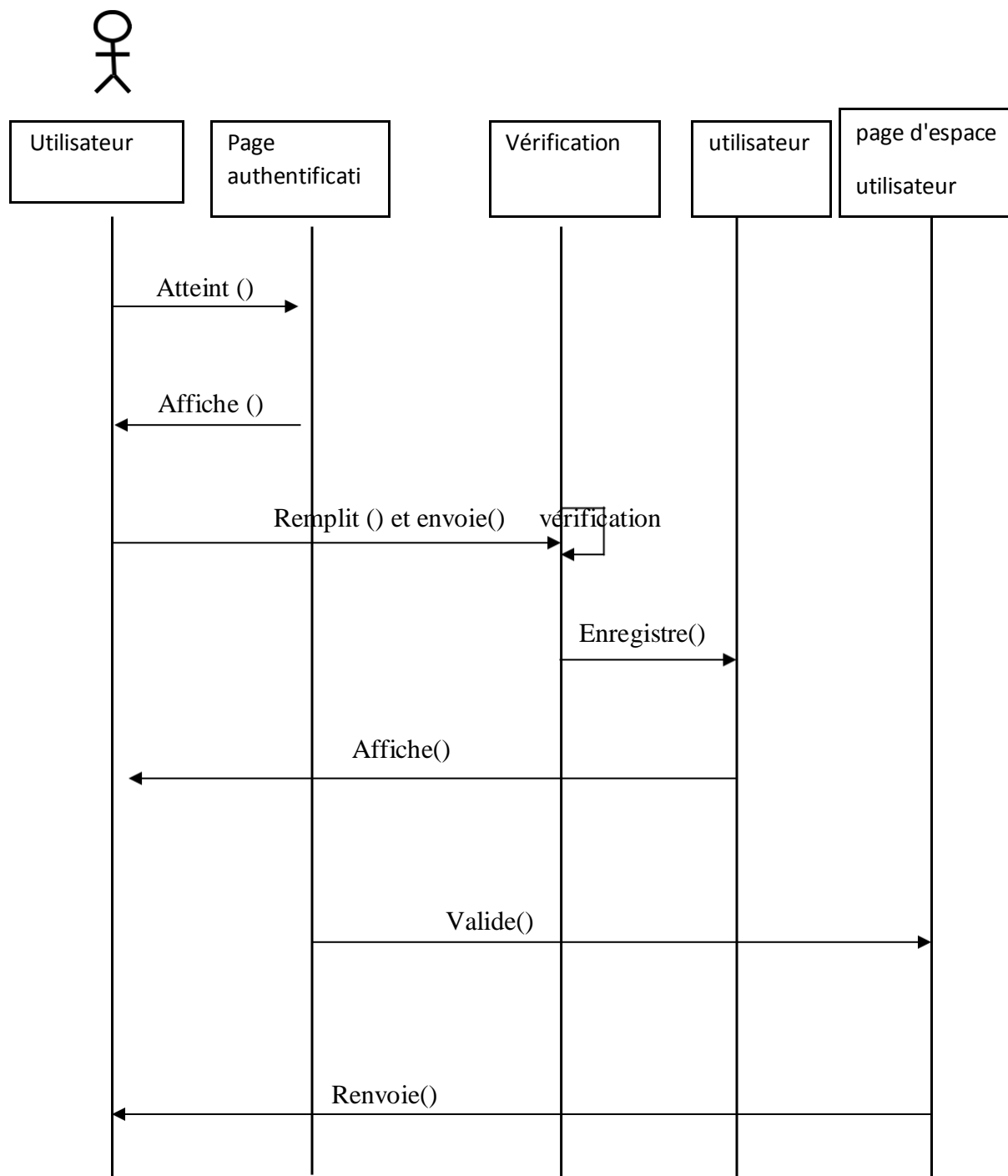
III.6.1. réalisation de l'itération 1:**III.6.1.1. Diagramme de séquence :**

Le diagramme de séquence suit le diagramme de cas d'utilisation car il le complète. Il permet de décrire les scénarios (déroulement des traitements entre les éléments du système et les acteurs) de chaque cas d'utilisation en mettant l'accent sur la chronologie des opérations en interaction avec les objets. En particulier, il montre aussi les objets qui participent à l'interaction par leur "ligne de vie" et les messages qu'ils échangent présentés en séquence dans le temps.

Voici quelques notions de base du diagramme :[15]

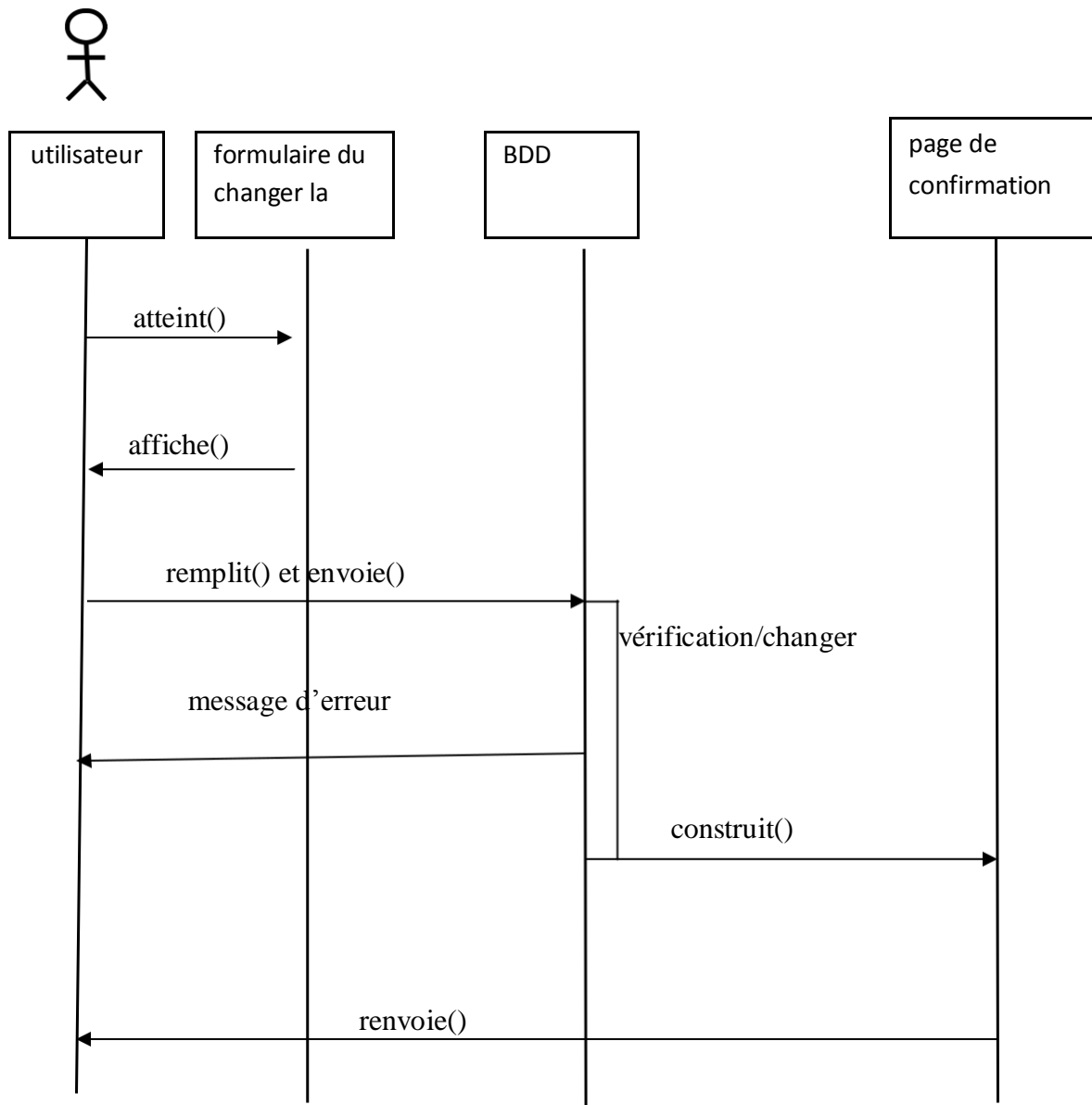
- **Scénario** : une liste d'actions qui décrivent une interaction entre un acteur et le système.
- **Interaction** : un comportement qui comprend un ensemble de messages échangés par un ensemble d'objets dans un certain contexte pour accomplir une certaine tâche.
- **Message** : Un message représente une communication unidirectionnelle entre objets qui transporte de l'information avec l'intention de déclencher une réaction chez le récepteur.

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



FigureIII.21: Diagramme de séquence du cas d'utilisation : « Authentification »

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



FigureIII.22 :Diagramme de séquence du cas d'utilisation : « **changer le password** »

- Diagramme de séquence du cas d'utilisation : « ajouter compte utilisateur »

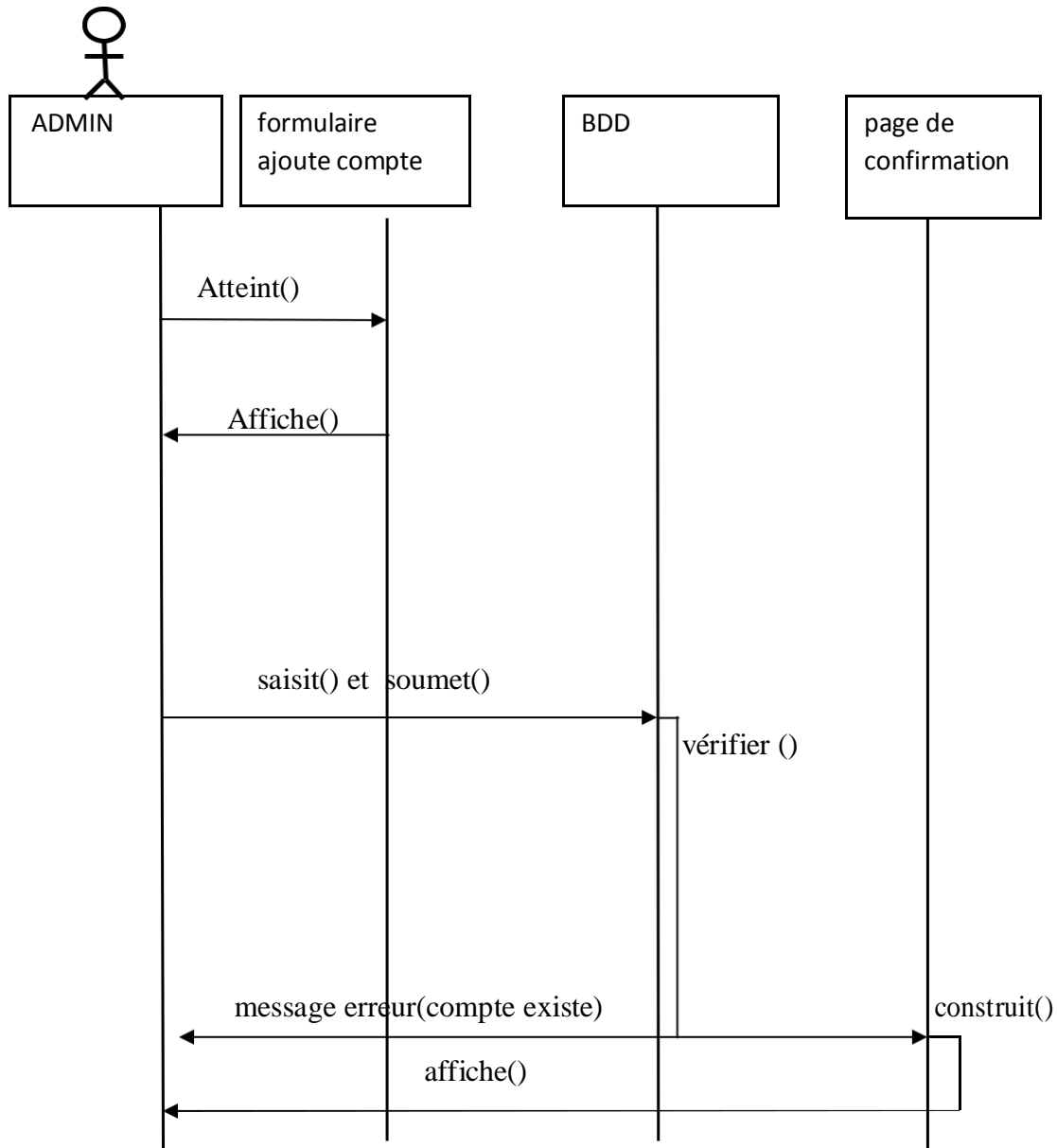


Figure III.23:Diagramme de séquence du cas d'utilisation : « ajouter compte utilisateur »

- Diagramme de séquence du cas d'utilisation : «modifier bon de transport»

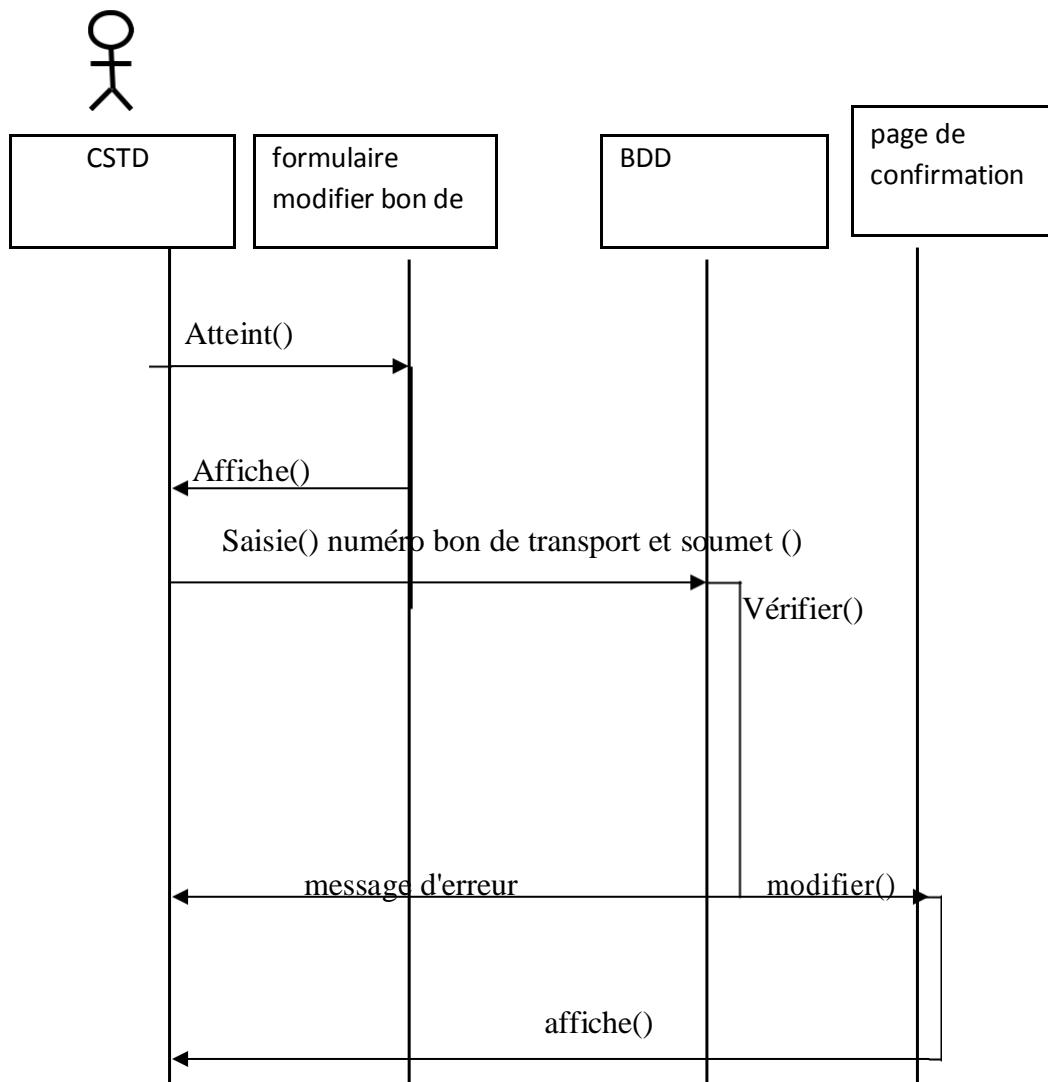


Figure III.24:Diagramme de séquence du cas d'utilisation : « **modifier bon de transport** »

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

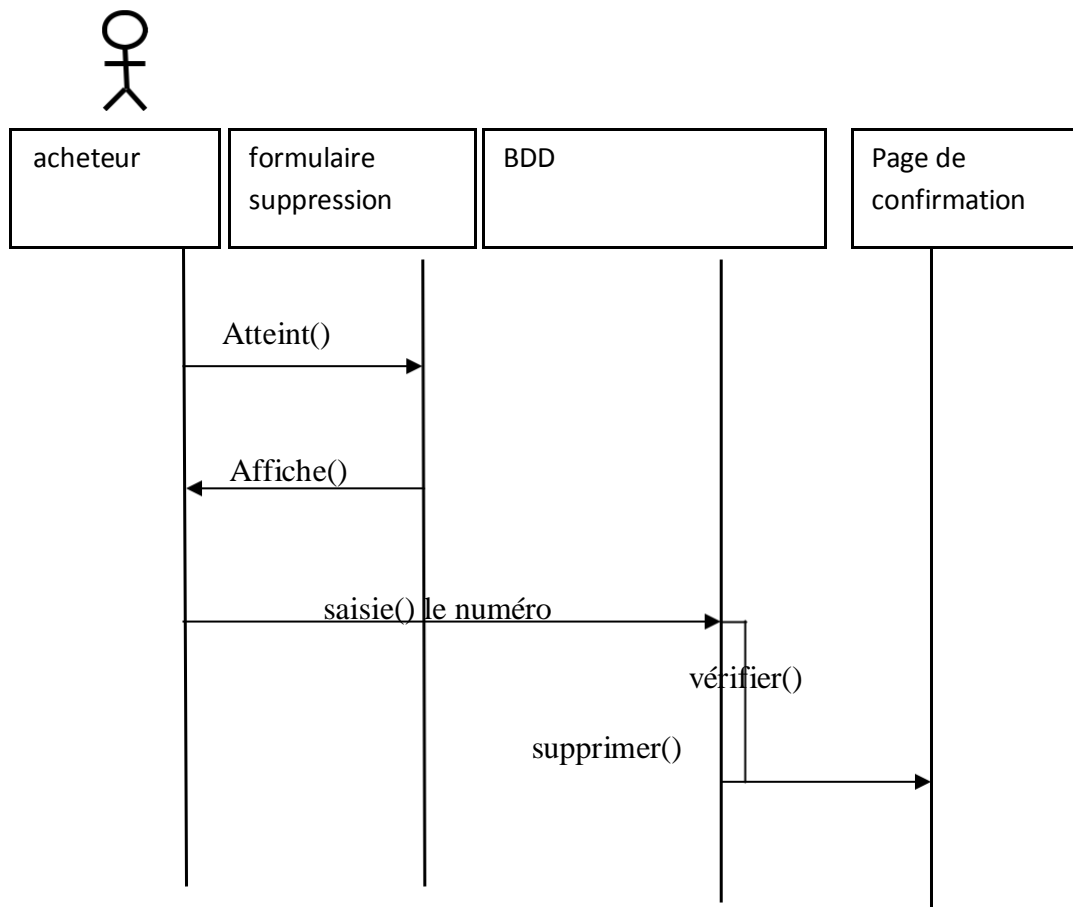
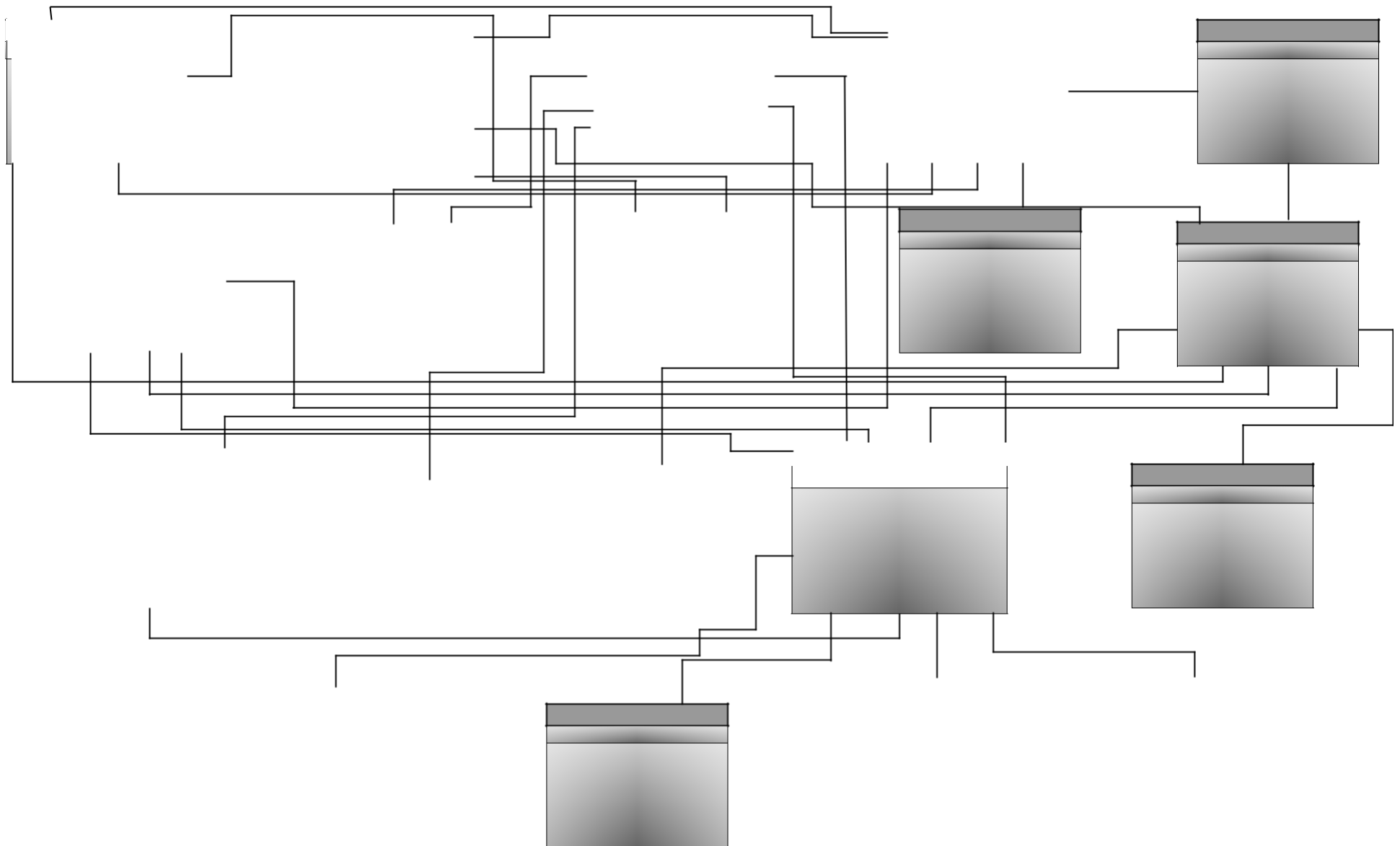


Figure III.25 : Diagramme de séquence du cas d'utilisation : « **supprimer une commande** »

III.6.1.2. Diagrammes de classe pour l'itération 1

Utilisateur
id_ut
Ajouter () Supprimer () Modifier () Consulter()

III.6.1.3. Diagrammes de classe pour l'itération 2



III.6.1.4. diagramme de classe de l'Itération 3:

message
login
Lire message Supprimer message

Remarque de l'itération 3

La classe message relie avec la classe utilisateur de l'itération1 par la cardinalité (0,*1,*)

III.7. Conclusion

Le langage UML nous apporte une aide à toutes les étapes d'un projet, comme il nous offre ainsi de nombreux avantages pour l'analyse et la conception d'un système. Le couple UML et le Processus Unifié de Rational propose une approche pour conduire la réalisation de systèmes orientés objet.

Dans le chapitre suivant nous allons consacrer à la réalisation de notre application ainsi qu'à la présentation des différents outils de développement de notre application.

Chapitre IV :

LA REALISATION

IV. Introduction:

Après avoir présenté dans le chapitre précédents les différentes étapes d'analyse et de conception, nous allons présenter dans ce chapitre l'environnement de développement, et les outils qui ont servi et nous finirons par la présentation de quelques interfaces du logiciel.

IV.1. Outils de développement:

IV.1.1. Le serveur Web Apache Tomcat:[16]

Apache Tomcat est un conteneur Web libre de Servlets et JSP java EE. Il implémente les spécifications des servlets et des JSP du Java Community Process, est paramétrable par des fichiers XML et de propriétés, et inclut des outils pour la configuration et la gestion. Il comporte également un serveur HTTP.

Tomcat ne constitue qu'un conteneur Web, et non un serveur Web à part entière: il gère spécifiquement les servlets et les JSP (un compilateur Jasper compilant les pages JSP pour en faire des servlets). Il peut être également parfois désigné comme moteur de servlet.

Tomcat est en réalité souvent employé en combinaison avec un serveur Web Apache ou d'autres serveurs Web. Tomcat a été écrit en langage java, il peut donc s'exécuter via la machine virtuelle java sur n'importe quel système d'exploitation la supportant.



Figure .IV.26: L'interface Apache Tomcat

IV.1.2.HibernateFramework[17] :

Hibernate est un framework open source gérant la persistance des objets en base de données relationnelle.

Hibernate est adaptable en termes d'architecture, il peut donc être utilisé aussi bien dans un développement client lourd, que dans un environnement web léger de type Apache Tomcat ou dans un environnement Java EE complet : WebSphere, JBoss Application Server et Oracle WebLogic Server.

Hibernate apporte une solution aux problèmes d'adaptation entre le paradigme objet et les SGBD en remplaçant les accès à la base de données par des appels à des méthodes objet de haut niveau.

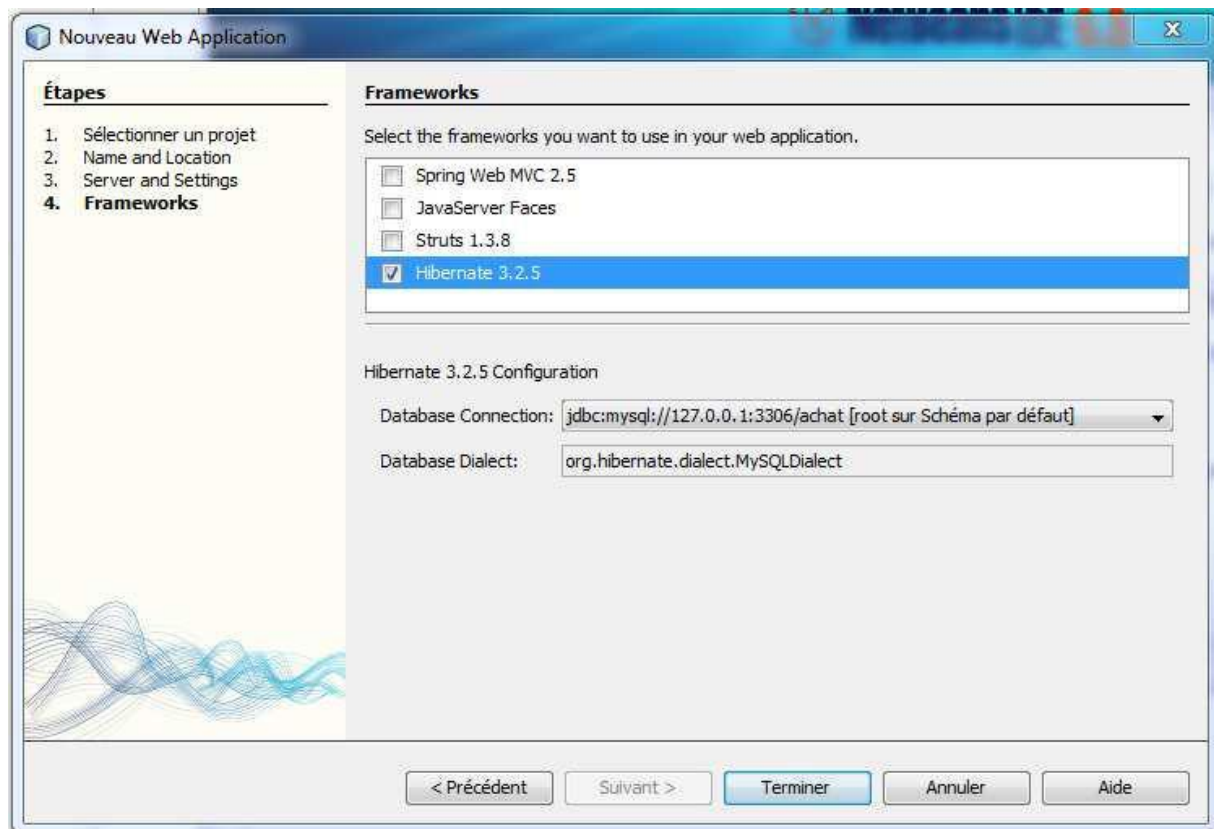


Figure.IV.27.l'interface Hibernate Framework :

IV.1.3. Dreamweaver [18]:

est un éditeur de site Web WYSIWYG pour Microsoft Windows, créée en 1997, Il est l'un des premiers éditeur HTML de type " tel affichage, tel résultat " mais également l'un des premiers à intégrer un gestionnaire de site, Ces innovations l'imposèrent rapidement comme l'un des principaux éditeur de site Web, aussi bien utilisable par le néophyte que par le professionnel.

De plus il offre deux modes de conception par son menu affichage, l'utilisateur peut choisir entre un mode création permettant d'effectuer la mise en page directement à l'aide d'outils simples et un mode code. Il offre aujourd'hui la possibilité de concevoir des feuilles de style. Les liaisons avec des bases de données ont également été améliorées ainsi que le chargement des fichiers sur les serveurs d'hébergement. Il propose en outre l'utilisation de modèles imbriqués de pages Web, selon un format propriétaire.



Figure.IV.28. l'interface Dreamweaver

IV.1.4. EasyPHP[19] :

EasyPHP fut le premier package WAMP à voir le jour (1999). Il s'agit d'une plateforme de développement Web, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. EasyPHP n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (un serveur web Apache et un serveur de bases de données MySQL), un interpréteur de script (PHP), ainsi qu'une administration SQL phpMyAdmin. Il dispose d'une interface d'administration permettant de gérer les alias (dossiers virtuels disponibles sous Apache), et le démarrage/arrêt des serveurs. Il permet donc d'installer en une seule fois tout le nécessaire au développement local du PHP. Par défaut, le

serveur Apache crée un nom de domaine virtuel (en ocal) 127.0.0.1 ou localhost. Ainsi, quand on choisit « Web local » dans le menu d'EasyPHP, le navigateur s'ouvre sur cette URL et affiche la page index.php de ce site qui correspond en fait au contenu du dossier www d'EasyPHP. EasyPHP peut être utilisé comme une *application portable*, c'est-à-dire lancé sur une clé USB.

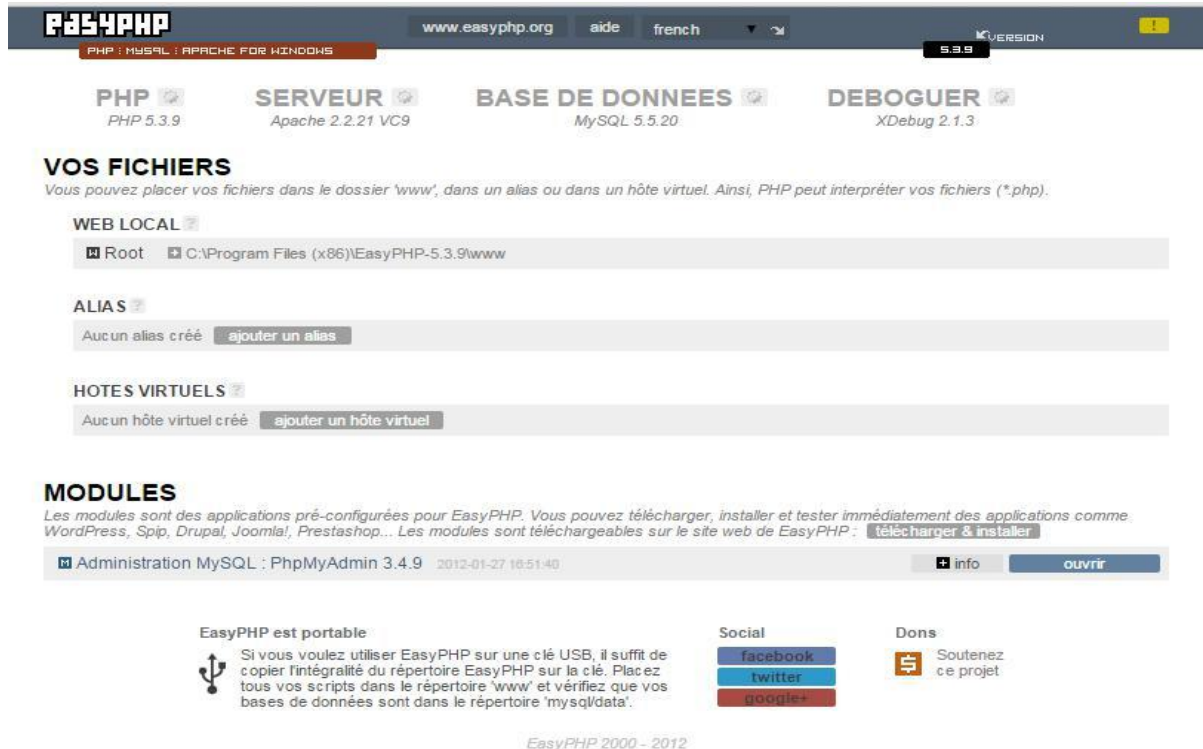


Figure .IV.29.l'interface EasyPHP

IV.1.5. phpMyAdmin:[20]

peut gérer l'ensemble du serveur MySQL (cela nécessite un compte super-utilisateur) aussi bien qu'une seule base de données. Pour ce faire, il est nécessaire d'avoir un droit de lecture/écriture sur la base de données concernée. Il vous appartient de consulter la section appropriée dans le manuel de MySQL *phpMyAdmin*. (PMA) est une application Web de gestion pour les systèmes de gestion de base de données MySQL réalisée en PHP et distribuée sous licence GNU GPL.

Le SGBD MySQL: [21]

MySQL est un gestionnaire de base de données SQL multi utilisateurs et multi traitements. Il se compose d'un langage de définition de données (LDD) et de droits ainsi que d'un langage de manipulation de données (LMD).

MySQL est fondé sur SQL qui est un langage de requêtes vers les bases de données exploitant le modèle relationnel. Il est implémenté sur un mode client serveur, avec un démon mysqld du côté serveur et toute une variété de bibliothèques et de programmes du côté client.

Il a l'avantage d'être portable (il peut être compilé sur plusieurs plates-formes comme Windows, Unix...etc.). De plus, il est facile à utiliser, standard (il utilise SQL), robuste et gratuit. Il permet le stockage et la manipulation de données d'une façon libre, rapide et fiable.

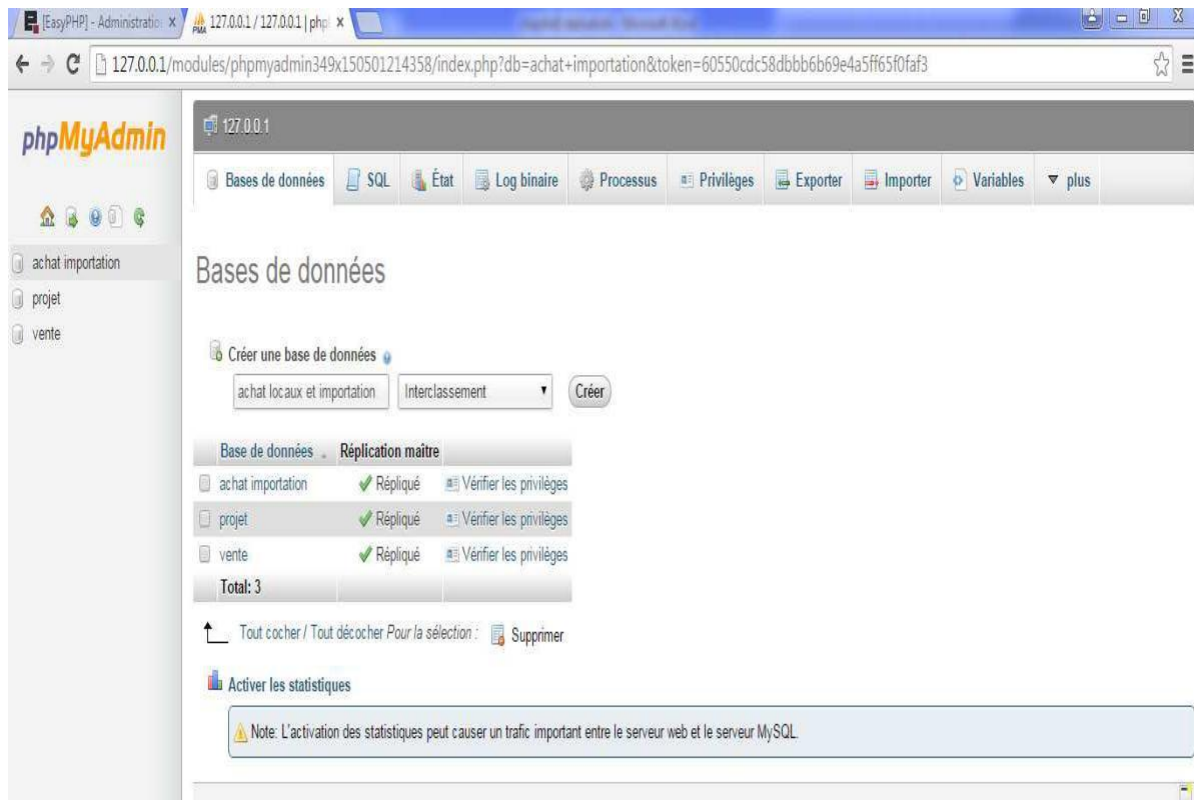


Figure.IV.30.l'interface phpMyAdmin

IV.1.6.Netbeans:[22]

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License) et GPLv2. En plus de Java, NetBeans permet également de supporter différents autres langages, comme C, C++, JavaScript, XML, PHP et HTML de façon native ainsi que bien d'autres (comme Python ou Ruby) par l'ajout de greffons. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java).

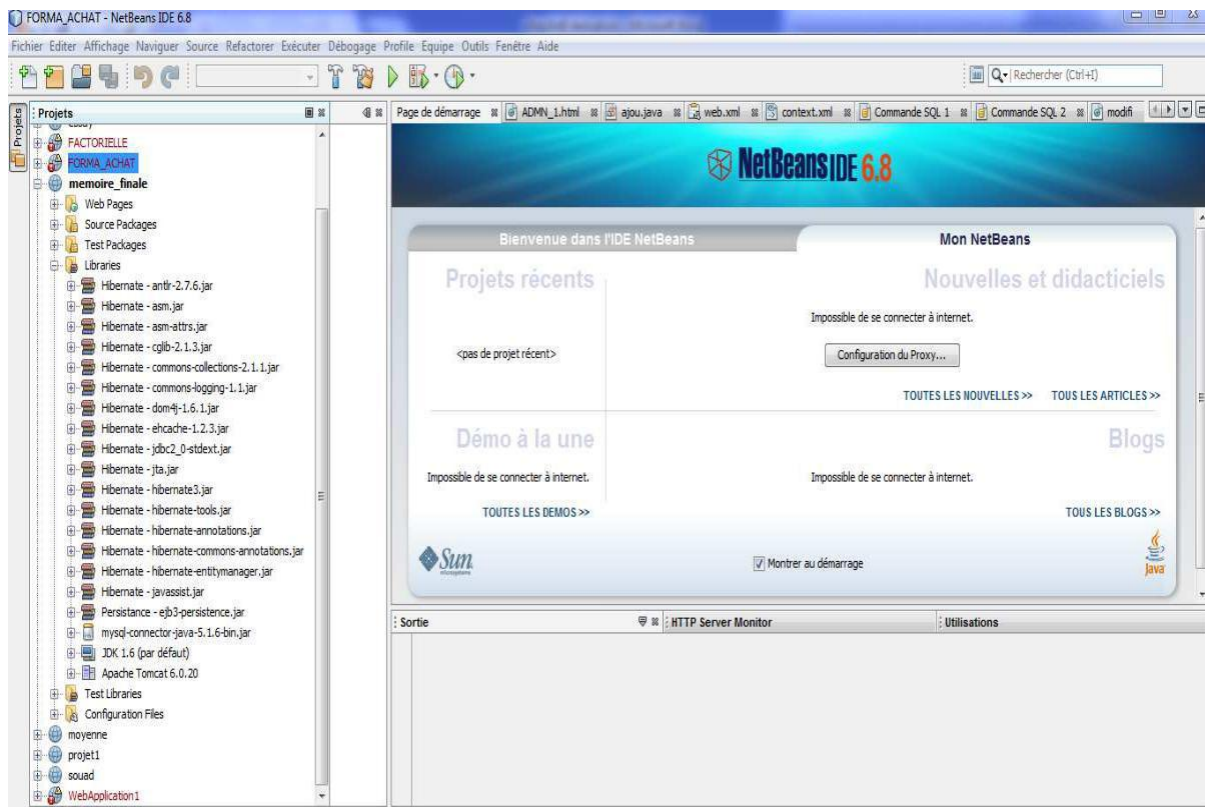


Figure.IV.31. l'interface de Netbeans:

IV.2. Les langages de programmation :

IV.2.1.Java:[23]

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 Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au *SunWorld*.

La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java.

La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que UNIX, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications. Pour cela, divers plateformes et frameworks associés visent à guider, sinon garantir, cette portabilité des applications développées en Java.

IV.2.2. HTML: [24]

Le HTML (« *HyperText Mark-Up Language* ») est un langage dit de « marquage » (de « structuration » ou de « balisage ») dont le rôle est de formaliser l'écriture d'un document avec des balises de formatage. Les balises permettent d'indiquer la façon dont doit être présenté le document et les liens qu'il établit avec d'autres documents.

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

IV.2.3. java EE:[25]

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 à un modèle logique de données qu'on écrit sous une forme relationnelle comme suit:

utilisateur:(id_ut, login, pwd_ut).

article: (num_art, date_etab_art, prix_achat, prix_vente, qte_delivre, date_livraison, lieu_livraison, designation).

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, obj_dai, nbre_article).

offre:(num_offre, date_etab_offre, description, qte, prix_unitaire, tva, total, total_avant, total_fina, num_fac_pro_for*, num_art*).

facture:(num_fac, qte, montant_ht, taux, montant_tva, montant_ttc, tva, total_ht, total_ttc, num_liv*)

bon_livraison: (num_liv _____, date_etab_liv, adr_liv, liv_sage, raison_sociale, marchandise_remise,

cni_pc, date_deliv, camion, immat, lieu_delivre, nom_produit, condi, qte, obs, rest, solde).

fournisseur: (num_four, nom_four, prenom_four, adr_four, fax_four_tel_four, type_four).

bon_de_transport: (num_bon_tran, date_etab_bon_tran, lieu_etab_bon_trans, nom_navire, num_liv*).

bulletin_de_reception: (num_bul_recep, date_etab_bul_recep, num_com*, date_etab_com, num_four*, designation, num_liv*, date_etab_liv, mer, air, terre).

connaissance: (num_con, date_etab_con, lieu_etab_con, nom_charg, cout_fret_march, nom_cap_nav, org_charg_march, date_charg_march, heur_charg_march, cout_frais_march, num_liv*).

avis_darivee: (num_a_a, date_etab_a_a, nom_banq, nom_navire, heur_a_march, num_liv*).

bon_delivrer: (num_bon_del, date_etab_bon_del, lieu_etab_bon_del, 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*).

d3: (num_d3, date_etab_d3, heur_etab_d3, lieu_etab_d3, num_liv*).

d11: (num_d11, date_etab_d11, heur_etab_d11, lieu_etab_d11, num_liv*).

fiche_de_verification: (num_fiche_ver, nom_four, montant, qualite, c_qualite, qualite_c, rigueur, norme, evolution_prix, reference, contrat, commande, p1, p2).

consultation: (num_con, date_etab_con).

tco: (num_tco, date_etab_tco, num_art, designation, qte, unite, noma, nomb, nomc, devisea, deviseb, devisec, dinarsa, dinarsb, dinarsc, four_retenue).

demande_cheque: (num_dmd_cheq, date_etab_cheq, lieu_etab_cheq, mont_cheq, num_fac*).

message(login, id_ut, commentaire).

historique (num_his, num_com, date_etab_com) ;

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 table article:

Nom du champ	Désignation du champ	Type de donnée	Clé(s)
num_art	Numéro d'article	Int(10)	primaire
date_etab_art	Date établissement article	Date	
prix_achat	Le prix d'achat de la marchandise	Int(20)	
prix_vente	Le prix vente de la marchandise	Int(20)	
qte_delivre	La quantité délivrer	Int(10)	
date_livraison	La date de livraison de la marchandise	Date	
lieu_livraisn	Le lieu de livraison de la marchandise	Varchar(50)	
designation	La désignation	Varchar(20)	

Table avis_darriver:

Nom du champ	désignation	Type de donnée	clé
num_a_a	Numéro d'avis d'arrivée	Varchar(20)	primaire
date_etab_a_a	Date établissement avis d'arrivée	date	
nom_banq	Nom de banque	Varchar(30)	
nom_navire	Nom de navire	Varchar(30)	
heur_a_march	Heur d'arrivée de marchandise	time	
num_liv	Numéro delivraison	Varchar(20)	étranger

Table bon_delivrer:

Nom du champ	désignation	Type de donnée	clé
num_bon_del	Numéro de bon de delivre	Varchar(20)	primaire
date_etab_bon_del	Date etablissement de bon delivre	date	
lieu_etab_bon_del	Lieu de etablissement de bon delivre	Varchar(50)	
num_liv	Numéro de livraison	Varchar(20)	étranger

Table bon_de_transport:

Nom du champ	Designation du champ	Type de donnée	Clé(s)
num_bon_tran	Numéro de bon de transport	Varchar(20)	primaire
date_etab_bon_tran	Date établissement de bon de transport	date	
lieu_etab_bon_trans	Le lieu d'établissement de bon de transport	Varchar(50)	
nom_navire	Le nom de navire	Varchar(30)	
num_liv	Numéro de livraison	Varchar(20)	Etranger

Table bon_livraison :

Nom du champ	désignation	Type de donnée	clé
num_liv	Numéro de livraison	Varchar(20)	primaire
date_etab_liv	Date établissement de livraison	date	
adr_liv	Adresse de livraison	Varchar(50)	
liv_sage	Numéro de bon de livraison sage	Varchar(20)	
raison_sociale	Raison sociale	Varchar(20)	
marchandise_remise	Marchandise remise	Varchar(30)	
date_deliv	Date delivre	date	
camion	camion	Varchar(30)	
immat	immatriculation	Varchar(30)	
lieu_delive	Lieu delivrer	Varchar(20)	
nom_produit	Nom de produit	Varchar(20)	
condi	condition	Varchar(20)	
qte	quantité	int(20)	
obs	observation	Varchar(20)	
rest	Reste a livrer	int(20)	
solde	solde	int(20)	

Table bulletin_de_reception :

Nom du champ	Designation du champ	Type de donnée	Clé(s)
num_bul_recep	Le numéro de bulletin de réception	Varchar(20)	primaire
date_etab_bul_recep	La date d'établissement de bulletin de réception	Date	
num_com	Numéro commande	Varchar(20)	étranger
date_etab_com	Date établissement de la commande	Date	
num_four	Numéro fournisseur	Varchar(20)	étranger
designation	Désignation	Varchar(20)	
num_liv	Numéro de la livraison	Varchar(20)	étranger
date_etab_liv	Date établissement de la livraison	Date	
mer	Mer	Varchar(10)	
air	Air	Varchar(10)	
terre	Terre	Varchar(10)	

table commande:

Nom du champ	Designation du champ	Type de donnée	Clé(s)
num_com	Numéro de commande	Varchar(20)	primaire
adr_com	Adresse de la commande	Varchar(50)	
date_etab_com	Date d'établissement de la commande	Date	
adr_envoi	Adresse d'envoi	Varchar(50)	
date_consultation	Date consultation	Date	
num_offre	Numéro offre	Varchar(20)	étranger
designation	Désignation	Varchar(30)	
qte_com	Quantité commande	Int(11)	
prix_unitaire	Prix unitaire	Int(11)	
montant_total	Montant total	Int(11)	
tva	La tva	Int(11)	
montant_rendu	Montant rendu	Int(11)	

table connaissance :

Nom du champ	désignation	Type de donnée	clé
num_con	Numéro de connaissance	Varchar(30)	primaire
date_etab_con	Date établissement de connaissance	date	
lieu_etab_con	Lieu établissement de connaissance	Varchar(50)	
nom_charg	numéro de changement	Varchar(20)	
cout_fret_marche	Cout de fret des marchandises	Varchar(20)	
nom_cap_nav	Nom de capitaine de navire	Varchar(20)	
org_charg_march	origine de chargement de la marchandise	Varchar(50)	
date_charg_march	date de changement de marchandise	date	
heur_charg_march	Heur de changement de marchandise	time	
cout_frais_march	Cout de frais des marchandises	Varchar(20)	
num_liv	Numéro de livraison	Varchar(30)	étranger

table consultation:

Nom du champ	Designation du champ	Type de donnée	Clé(s)
num_con	Le numéro de connaissance	Varchar(20)	primaire
date_etab_con	La date d'établissement de connaissance	date	

table d3:

Nom du champ	Designation du champ	Type de donnée	Clé(s)
num_d3	Le numéro de la d3	Varchar(20)	primaire
date_etab_d3	La date d'établissement de la d3	date	
heur_etab_d3	Heur d'établissement de la d3	Time	
lieu_etab_d3	Lieu d'établissement de la d3	Varchar(50)	
num_liv	Le numéro de la livraison	Varchar(20)	étranger

table d11 :

Nom du champ	Designation du champ	Type de donnée	Clé(s)
num_d11	Le numéro de la d11	Varchar(20)	primaire
date_etab_d11	La date d'établissement de la d11	date	
heur_etab_d11	Heur d'établissement de la d11	Time	
lieu_etab_d11	Lieu d'établissement de la d11	Varchar(50)	
num_liv	Le numéro de la livraison	Varchar(20)	étranger

Table dai:

Nom du champ	désignation	Type de donnée	clé
num_dai	Numéro de demande d'achat interne	Varchar(20)	primaire
date_emission	Date émission	date	
nbre_article	Nombre article	Varchar(10)	
date_etab_dai	Date d'établissement	date	
obj_dai	Objectif de dai	Varchar(50)	

Table demande_cheque:

Nom du champ	Designation du champ	Type de donnée	Clé(s)
num_dmd_cheq	Numéro de demande de cheque	Varchar(20)	Primaire
date_etab_cheq	Date d'établissement de cheque	date	
lieu_etab_cheq	Lieu d'établissement de cheque	Varchar(50)	

mont_cheq	Montant de cheque	Int(11)	
num_fac	Numéro de la facture	Varchar(20)	étranger

table facture:

Nom du champ	désignation	Type de donnée	clé
num_fac	Numéro de facture	Varchar(20)	primaire
qte	quantité	Int(11)	
montant_ht	montant hors taxe	Int(11)	
taux	taux	Int(11)	
montant_tva	montant de taxe du la valeur à ajouter	Int(11)	
montant_ttc	montant de tous taxe comprise	Int(11)	
tva	taxe de la valeur ajouter	Int(11)	
total_ht	total de hors taxe	Int(11)	
total_ttc	total de tous taxe comprise	Int(11)	
num_liv	Numéro de livraison	Varchar(20)	étranger

Table facture_pro_format:

Nom du champ	Désignation du champ	Type de donnée	Clé(s)
num_fac_pro_for	Numéro de la facture proformat	Varchar(20)	primaire
date_etab_fac_pro_for	Date d'établissement de la facture proformat	Date	
lieu_etab_fac_pro_for	Lieu d'établissement de la facture proformat	Varchar(50)	
num_four	Numéro de fournisseur	Varchar(20)	étranger
num_tco	Numéro de tableau de comparative des offres	Varchar(20)	étranger
num_art	Numéro article	Varchar(20)	étranger
code_produit	Code de produit	Varchar(20)	
designation	La designation	Varchar(30)	
qte	La quantité	Varchar(10)	
prix unitaire	Le prix unitaire	Varchar(10)	
tva	La tva(taxe de la	Varchar(10)	

	valeur ajouter)		
ttc	La ttc(tous taxe comprise)	Varchar(10)	

Table fihe_de_verification:

Nom du champ	désignation	Type de donnée	clé
num_fiche_ver	Numéro de fiche de vérification	Varchar(30)	primaire
nom_four	Nom de fournisseur	Varchar(20)	
montant	montant	Int(10)	
qualite	quantité	Varchar(20)	
c_qualite	Constance da la qualité	Varchar(20)	
qualite_c	qualité de l'assistance	Varchar(20)	
rigueur	rigueur dans les délais	Varchar(20)	
evolution_prix	Evolution de prix	Varchar(20)	
reference	reference	Varchar(20)	
commande	commande	Varchar(20)	
contrat	contrat	Varchar(20)	
P1	fournisseur(agrée, qualifier, nouveau)	Varchar(20)	
P2	mode de passation(avis d'appel offre ouvert, consultation	Varchar(20)	
norme	norme	Varchar(20)	

table fournisseur:

Nom du champ	Designation du champ	Type de donnée	Clé(s)
num_four	le numéro de fournisseur	Varchar(20)	primaire
nom_four	Le nom de fournisseur	Varchar(30)	
prenom_four	Le prenom de fournisseur	Varchar(30)	
adr_four	L'adresse de fournisseur	Varchar(50)	
fax_four	Le fax de fournisseur	Int(11)	
tel_four	Téléphne de fournisseur	Int(11)	
type_four	Le type de	Varchar(30)	

	fournisseur		
--	-------------	--	--

table offre:

Nom du champ	désignation	Type de donnée	clé
num_offre	Numéro de l'offre	Varchar(20)	primaire
date_etab_offre	Date d'établissement de l'offre	date	
description	description	Varchar(50)	
qte	Quantité de l'offre	Varchar(30)	
prix_unitaire	Le prix unitaire de l'offre	Varchar(20)	
tva	La tva	Varchar(20)	
total	Total de l'offre	Varchar(20)	
total_avant	Total avant de l'offre	Varchar(20)	
total_final	Total final de l'offre	Varchar(20)	
num_fac_pro_for	Numéro de facture proformat	Varchar(20)	Etranger
num_art	Numéro article	Varchar(20)	Etranger

Table tco:

Nom du champ	Designation du champ	Type de donnée	Clé(s)
num_tco	Numéro de tableau comparatif des offres	Varchar(20)	primaire
date_etab_tco	Lieu d'établissement de tableau comparatif des offres	Date	
num_art	Numéro d'article	Varchar(20)	étranger
designaton	Designation	Varchar(30)	
qte	Quantité	Int(11)	
unite	Unité	Varchar(10)	
noma	Nom de fournisseur a	Varchar(20)	
nomb	Nom de fournisseur b	Varchar(20)	
nomc	Nom de fournisseur c	Varchar(20)	
devisea	Devise a	Int(10)	
deviseb	Devise b	Int(10)	
devisec	Devise c	Int(10)	
dinarsa	Dinars a	Int(10)	
dinarsb	Dinars b	Int(10)	
dinarsc	Dinars c	Int(10)	
four_retenu	Le fournisseur retenu	Varchar(20)	

table utilisateur:

Nom du champ	Designation du champ	Type de donnée	Clé(s)
id_ut	Identifiant utilisateur	Varchar(30)	primaire
login	Le login utilisateur	Varchar(30)	
pwd_ut	Le password utilisateur	Varchar(30)	

table message :

Nom du champ	Designation du champ	Type de donnée	Clé(s)
login	Le login utilisateur	Varchar(30)	primaire
id_ut	Identifiant utilisateur	Varchar(30)	
commentaire	Le commentaire laisser	Varchar(50)	

table historique :

Nom du champ	Designation du champ	Type de donnée	Clé(s)
num_his	Le numéro historique	Int(5)	primaire
num_com	Numéro de commande	Varchar(30)	
date_etab_com	Date établissement de la commande	date	

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

page d'authentification de l'application :

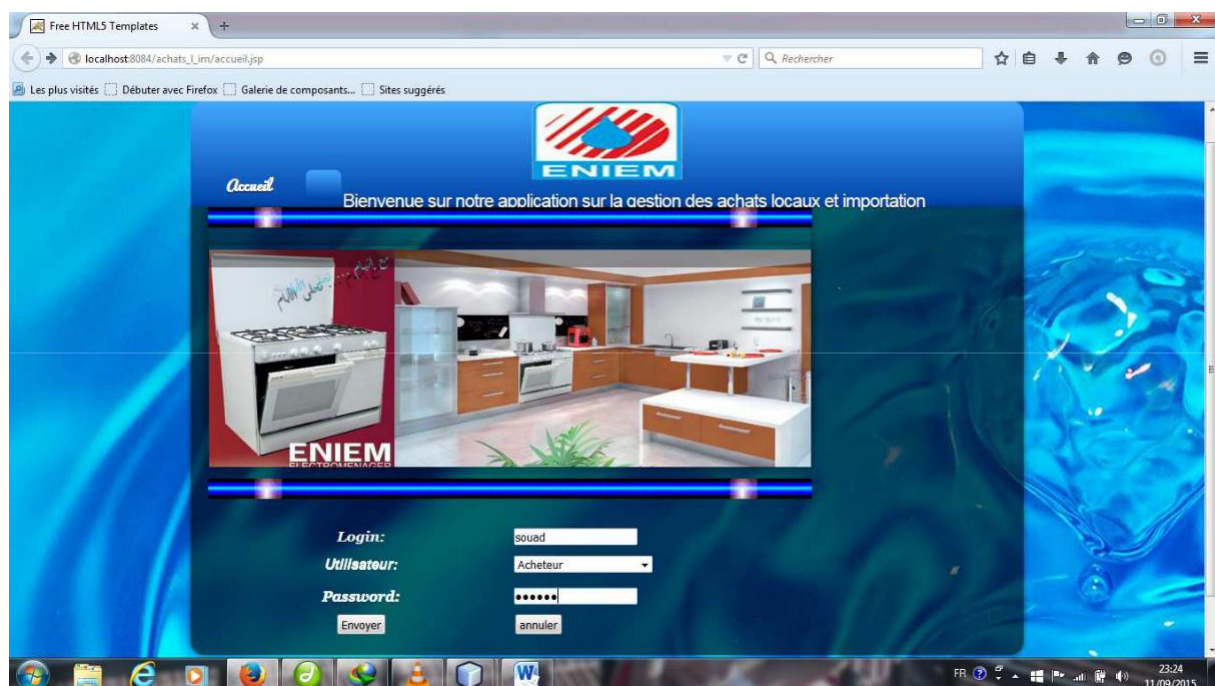


Figure .IV.32. page d'authentification.

Espace Administrateur

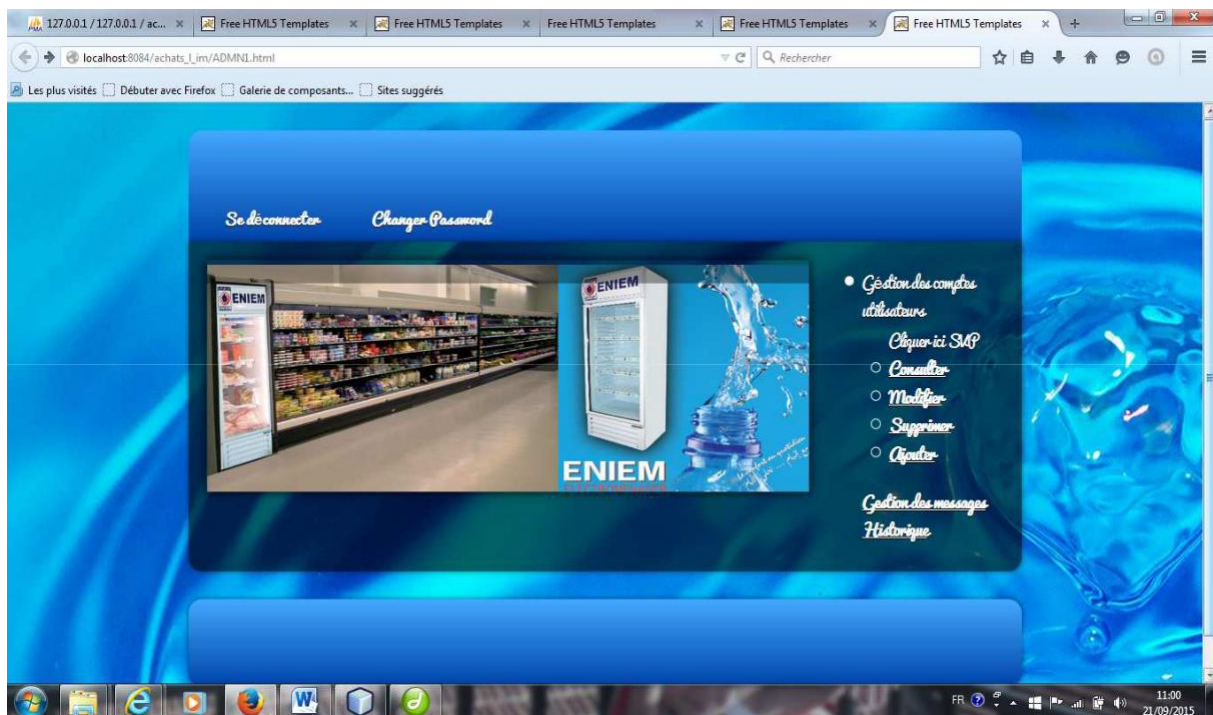


Figure .IV.33.espace Administrateur

Changer password

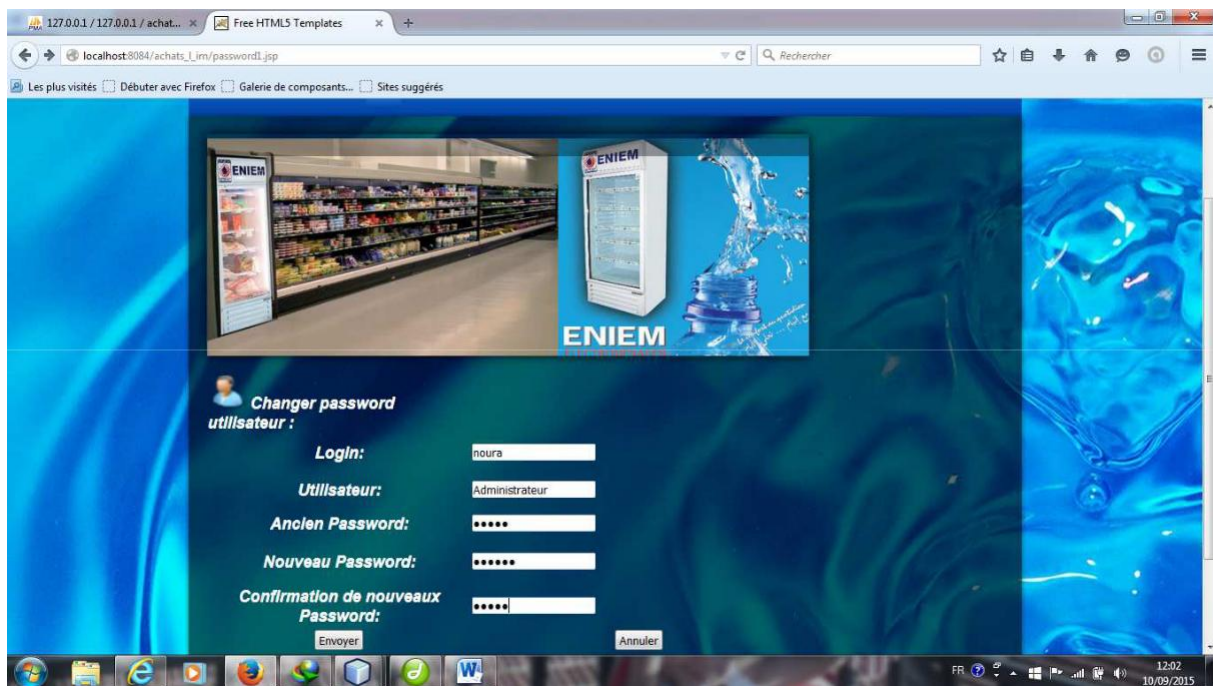


Figure .IV.34. changerpassword

Consulter bon de transport :



Figure .IV.35.consulter bon de transport

Supprimer dai :

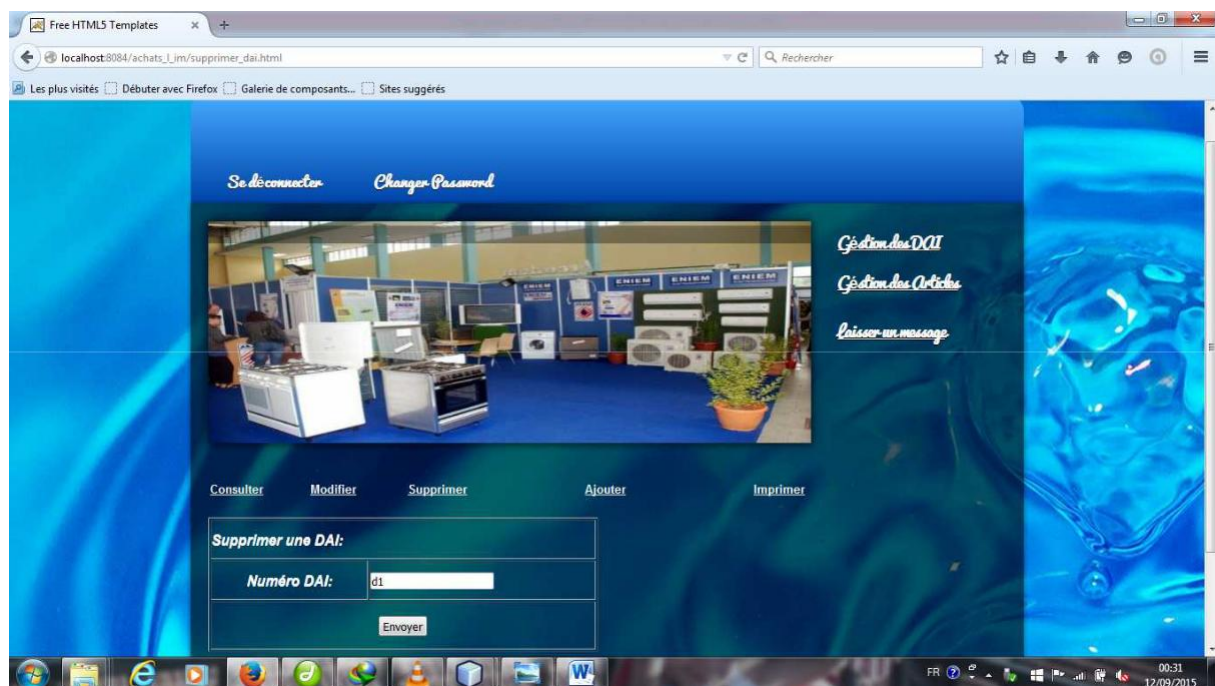


Figure .IV.36.supprimer dai

Ajouter Bon de commande :

ENIEM		oued aissi	
UNITE FROID		le	
Zone Industrielle de Oued-Aissi		2014-06-15	
BP 108 Poste Chikhi		BON DE COMMANDE	
Tizi-ouzou-Algérie		N° 14/00094	
Tél: 213 26 22 50 91		(A rappeler dans toutes correspondances)	
Fax: 213 26 22 54 92/50 19			
Notre consultation du 2014-04-30		DISTRIM SPA	
Votre Offre N° 101456		Zone Industrielle de	
		Réghaia-16908	

Code ENIEM	Désignation et spécification	U.M	Quantité	Prix Unitaire DA/Pièce	Montant total
300 005	Soude Caustique perlée	Kg	13100	71.00	930100.00

NB: La marchandise objet du présent bon de commande doit être conforme à la fiche technique ENIEM en votre possession.

- Mode de paiement : cheque à 45 jours date facturation	Montant total H.T	930 100.00
Type d'emballage : En sacs de 25 kg	Taxe T.V.A 17%	158 117.00
-délai de livraison :Suivant planning ENIEM ci-joint		

Marquage:

Commande N° 14/00094	Montant T.T.C Rendu Usine	1 088 217.00
ENIEM /Unité Froid		/
N° des caisses/Colis		/
Poids Net:		/
Poids Brut:		

					Nom:
					Fonction:
					Signature:

Ajouter Imprimer cette page

Figure . IV.37.ajouter bon de commande

Imprimer la facture :

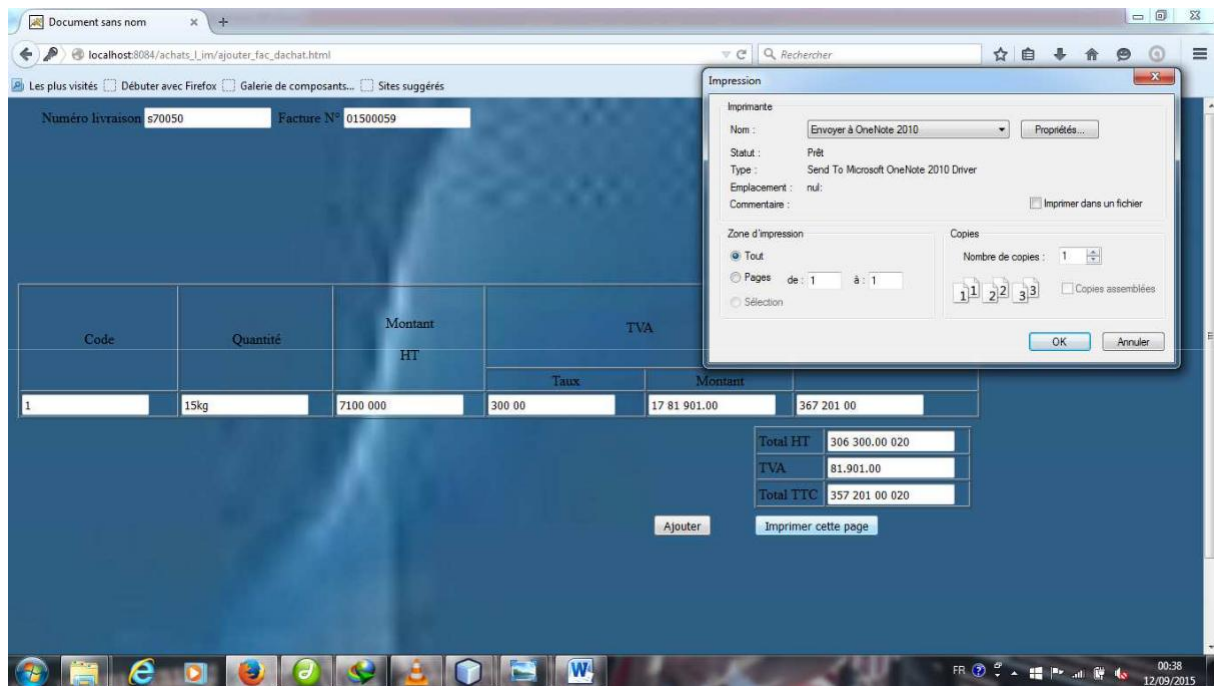


Figure .IV.38.imprimer la facture

IV.5. Conclusion :

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

CONCLUSION

GENERALE

Conclusion générale :

Nous voici au terme de notre mémoire dont l'intitulé est : « conception et réalisation d'une application JavaEE sur la gestion des achats locaux et importation ». Dans ce travail, nous avons voulu de trouver des solutions a des problèmes de performance , de temps et de disponibilité ... pour des utilisateurs ne sont pas spécialisés en informatique.

Le stage que nous avons effectué au sein de l'entreprise nationale des industries de l'électroménager (ENIEM) ainsi que l'étude et le développement de notre projet, nous a permis de mettre en pratique les notions théoriques acquises durant notre cursus d'études universitaires et d'acquérir de nouvelles connaissances (conception avec UML,HTML et JavaEE) et de mettre entre les mains des utilisateurs du service achat de l'entreprise ENIEM un outil qui leur facilitera le suivi des achats locaux et importation.

Notre application offre plusieurs services ; et cela en permettant l'accès rapide aux données et ainsi de faciliter la collaboration et la communication entre les différents acteurs de l'entreprise.

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

*BIBLIOGRAPHIE
ET
WEBLIOGRAPHIE*

Webliographie/Bibliographie

- [1] : www.wikipedia.org
- [2] : http://fr.wikipedia.org/wiki/Architecture_logicielle
- [3] : <http://www.framasoft.net/article3991.html>
- [4] : *Le modèle client-serveur* Olivier Aubert .pdf
- [5] : http://caron.nicolas.free.fr/cnam/Projet%20Methodologie/PASSAGE_DU_CLIENT_LOURD_AU_CLIENT_LEGER.ppt
- [6] : <http://www.commentcamarche.net/cs/peer.php3>
- [7] : Université Hassiba Ben Bouali Chlef /Faculté des sciences et sciences de l'ingénieur /
Département TCT/support du cours /architecture réseaux [8] : www.comment-ça-marche.net
- [9] : Mémoire promotion 2012/2013 réalisé par :amranesamia et amrane hassina (conception et réalisation d'une application web avec J2EE pour la gestion de scolarité)
- [10] : Mémoire promotion 2011/2012 ;réalisé par M^rLEHAD OMAR ET M^rsaddedinemohamed ;dirigé par M^m bourkache ;le thème conception et réalisation d'une application j2ee web service cas : gestion de la scolarité du département informatique.
- [11] : http://infocenter.sybase.com/archive/index.jsp?topic=/com.sybase.stf.powershell/docs_12.0.0/html/dcg/dcgup811.html
- [12] : http://www.ai.univ-paris8.fr/~maa/M1_J2EE_Jsp.pdf
- [13] : <http://www.jmdoudoux.fr/java/dej/chap-ejb.htm>
- [14] : Laurent Aulibert, UML 2 de l'apprentissage à la pratique(cours et exercices),édition elipses.
- [15] : Définition et caractéristique d'UML,2008
- [16] : <http://fr.Wikipedia.org/Wiki/ApacheTomcat>
- [17] : <http://fr.Wikipedia.org/Wiki/Hibernate>
- [18] : <http://fr.Wikipedia.org/Wiki/Adobe-DreamWeaver>
- [19] : <http://fr.Wikipedia.org/Wiki/EasyPHP>
- [20] : <http://fr.Wikipedia.org/Wiki/phpMyadmin>
- [21] : [http://WWW.memoireonline.Com/10/12/6388/m-conception-et-realisation-du-application-java.card-pour-la-recherche-de-billet_d'avion-le-m0\(\).html](http://WWW.memoireonline.Com/10/12/6388/m-conception-et-realisation-du-application-java.card-pour-la-recherche-de-billet_d'avion-le-m0().html)
- [22] : <http://fr.Wikipedia.org/Wiki/netbeans>
- [23] : [http://fr.Wikipedia.org/java-\(langage\)](http://fr.Wikipedia.org/java-(langage)).
- [24] : <http://fr.Wikipedia.org/Wiki/Hypertexte-markup-langage>
- [25] : <http://fr.Wikipedia.org/Wiki/java.EE>

ANNEXE

I. 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 pour le langage UML qui permet de bien représenter l'aspect statique et dynamique d'une application par une série de diagrammes qu'il offre.

II. Définition UML :

Le **langage de modélisation unifié**, de l'anglais *Unified Modeling Language (UML)*, est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. 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 Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard adopté par l'Object Management Group (OMG).

III. 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.

IV. 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 bonhomme avec son rôle inscrit dessous.



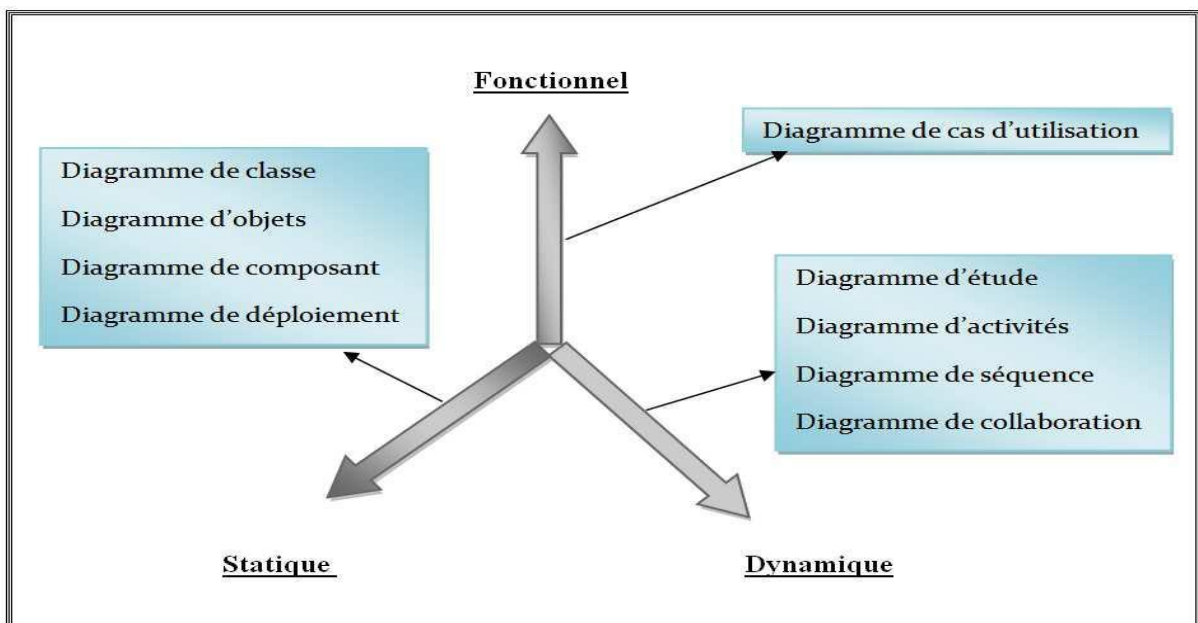
Acteur

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

V. 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, à l'aide des diagrammes.

Les vues d'UML :



5.1.1. Vue statique (structurelle) : S'appuie sur les diagrammes suivants :

Diagramme de classe

Diagramme d'objet

Diagramme de composant

Diagramme de déploiement

5.1.2. **vue dynamique (comportementaux)** : S'appuie sur les diagrammes suivants

Diagramme d'activité

Diagramme de séquence

Diagramme d'étude

Diagramme de collaboration

5.1.3. **vue fonctionnel**:

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

Les diagrammes de classe :

Un diagramme des classes constitue le cœur 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 quelles seront les composantes du système final, 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.



Figure 1 représentation graphique de diagramme de classe

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.

La relation include :

Une relation d'inclusion d'un cas d'utilisation A par rapport à un cas d'utilisation B signifie qu'une instance de A contient le comportement décrit dans B.

La relation extend :

Une relation d'extension d'un cas d'utilisation A par un cas d'utilisation B signifie qu'une instance de A peut être étendue par le comportement décrit dans B.

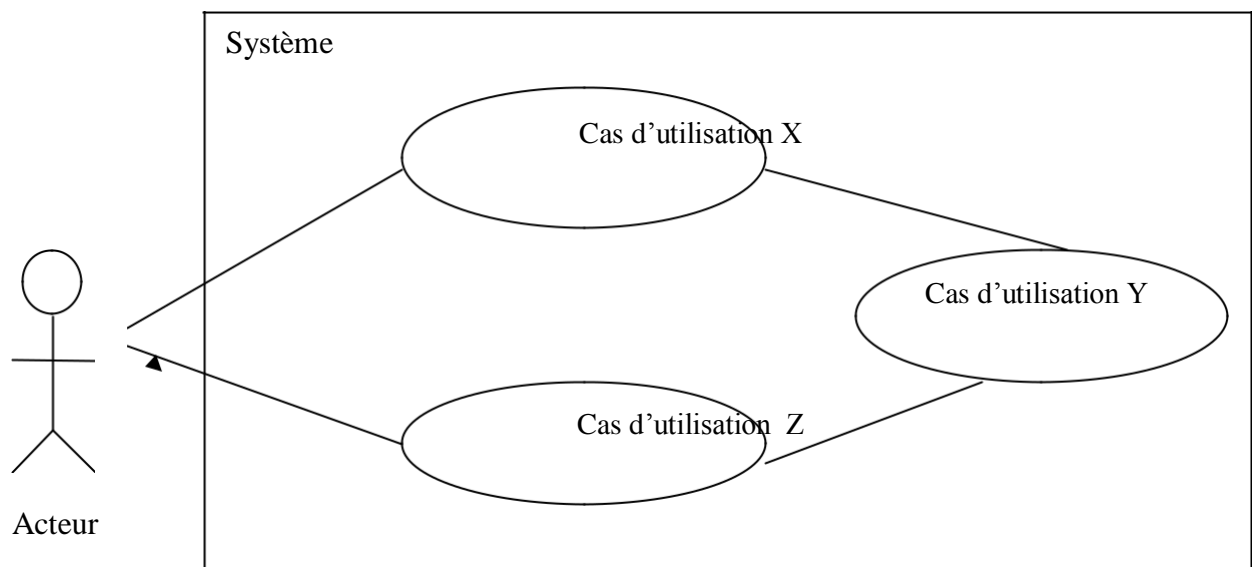


Figure2 : représentation graphique d'un diagramme de cas d'utilisation

Les diagrammes de séquence:

Le diagramme de séquence représente la succession hronologique des opérations réalisées par un acteur. Il indique les objets quel'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 noeuds sont des objets et les arcs (numérotés selon la chronologie) les échanges des messages entre objets. En fait, diagramme de séquence et diagramme de communicationsont deux vues différentes mais logiquement équivalentes d'une mêmechronologi.

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.

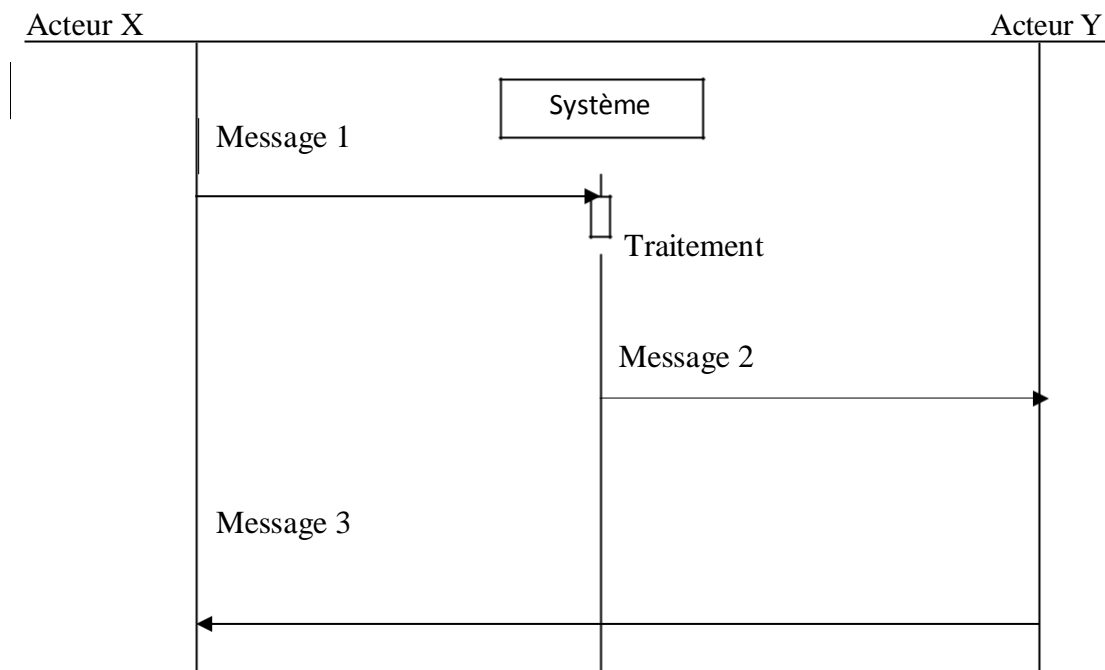


Figure 3 : Représentation graphique du diagramme de séquence

VI. CONCLUSION :

Comme UML n'impose pas de méthode de travail particulière, il peut être intégré à n'importe quel processus de développement logiciel de manière transparente. UML est une sorte de boîte à outils, qui permet d'améliorer progressivement vos méthodes de travail, tout en préservant vos modes de fonctionnement .

Intégrer UML par étapes dans un processus, de manière pragmatique, est tout à fait possible. La faculté d'UML de se fondre dans le processus courant, tout en véhiculant une démarche méthodologique, facilite son intégration et limite de nombreux risques (rejet des utilisateurs, coûts...).