

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

*MINISTERE DE LA RECHERCHE SCIENTIFIQUE ET DE L'ENEIGNEMENT
SUPERIEUR*

UNIVERSITE MOULOUD MAMMERI DE TIZI OUZOU

FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE



MEMOIRE



DEPARTEMENT D'INFORMATIQUE

DE FIN D'ETUDES

EN VUE DE L'OBTENTION DU DIPLOME DE Master2 EN INFORMATIQUE

**Conception et réalisation d'une application client/serveur sous oracle pour la
gestion approvisionnement**

**Proposé par Mr
KERBICHE**

Réalisé par :

M^{elle} IGHEROUSSENE DJEDJIGA

M^{elle} IAZOURENE SOUHILA

PROMOTION : 2014-2015

Remerciements

Nous tenons à témoigner notre reconnaissance à DIEU tout puissant, qui nous a aidé et bénis par sa volonté durant toute cette période.

Notre profonde gratitude et sincères remerciements vont à notre promoteur Mr KERBICHE pour sa précieuse assistance, sa disponibilité et l'intérêt qu'elle a manifesté pour ce modeste travail.

Nos plus vifs remerciements vont à tout le personnel de l'EI Qui nous ont généreusement aidées durant notre stage.

Tous nos remerciement aux membres de jury qui nous font l'honneur de juger se modeste travail.

Nos remerciements vont également à tous ceux qui ont répondu aux questions sur des problèmes rencontrés.

Nos remerciements vont aussi à tous ceux qui ont contribué de près ou de loin à la réalisation de notre travail.

Dédicaces

*J'ai l'honneur de dédier ce modeste travail
à :*

Ma chère mère et Mon père à qui je souhaite une longue vie.

Mes très chers frères :nafaa, hidouche, moussa

Mes très chères sœurs :kahina, fazia, noura, djidja

Toutes mes copines :dalila, farida, samia ,dalia

Mon binôme souhilaet sa famille .

Mon promoteur et sa famille.

Tous ceux qui m'ont aidé à réalisé ce travail.

djidji

Dédicaces

Je dédie ce humble travail à :

- ✚ La mémoire de mes chère cousins Djamila et nana Farroudja .*
- ✚ La mémoire de ma grand-mère hasni .*
- ✚ Ma perle précieuse, ma source de bonheur, volonté et affection, mon adorable maman.*
- ✚ Mon adorable père.*
- ✚ A mon frère Amin et mes sœurs :Amel, Samia, Dahbia .*
- ✚ A mon grand père paternel et maternel .*
- ✚ A mon grand mère paternel et maternel.*
- ✚ Tous mes cousins et cousines en particulier : tahar et Ghania.*
- ✚ Toutes mes copines :dalila, farida,samia ,dalia*
- ✚ Mon binôme djiji et sa famille .*
- ✚ Mon promoteur et sa famille.*
- ✚ A tous ceux qui ont participé de près ou de loin à la réalisation de notre travail (sosso...)*

Souhila

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

Chapitre 1 : généralités sur les réseaux et client/serveur

I.1. Les réseaux :	2
I.1.1. Introduction	2
I.1.2. Généralité sur les réseaux	2
I.1.3. Définition d'un réseau :	2
I.1.4. Objectifs des réseaux.....	3
I.1.5. Types de réseau	3
I.1.5.1. Selon la distance :	3
I.1.5.2. Selon la topologie :	3
I.1.5.2.1. Structure de réseau en bus.....	4
I.1.5.2.2. L'étoile :	4
I.1.5.2.3. L'anneau:	4
I.1.5.2.4. Structure de réseau maillée	5
I.1.6. Types de lignes de transmission.....	6
I.1.6.1. Câbles métalliques:	6
I.1.6.2. Paire torsadée :	6
I.1.6.3. Câble coaxial:	6
I.1.6.4. Fibres optiques:.....	6
I.1.6.5. Faisceaux hertziens:.....	6
I.1.6.6. Liaison satellite:.....	7
I.1.7. Logiciels de réseau	7
I.1.8. Les modèles de réseau	7
I.1.9. Le modèle TCP/IP et ses couches	8
I.1.10. Comparaison entre le modèle OSI et le TCP/IP.....	9
I.2. Le client /serveur :	11
I.2.1. Introduction	11
I.2.2. Le modèle client /serveur	11
I.2.2.1. Définition:.....	11
I.2.2.2. Middleware	12
I.2.3. Pourquoi le client /serveur :	12
I.2.4. Modes de fonctionnement :	13
I.2.5. Fonctionnement d'un système client/serveur :	14

I.2.6. Les protocoles:	14
I.2.7.Caractéristique du client/serveur	17
I.2.8.Types de clients :.....	18
I.2.9.Les type serveur :	18
I.2.10. Architecture client/serveur :	19
I.2.11.Avantages de l'architecture client/serveur :.....	21
I.2.12.Inconvénients de l'architecture client/serveur :	22
Conclusion :.....	23

Chapitre 2 : Présentation de l'organisme d'accueil

II.1.Historique :	24
II.2.Localisation de l'entreprise :	25
II.3.Description de la structure :	25
II.4. Mission et objectif de l'entreprise :	26
II.5. Organigramme générale de l'entreprise :	28
II.6. Présentation du champ d'étude :.....	29
II.6.1. Organigramme détaillé du champ d'étude :.....	29
II.6.2 Présentation des différents services du champ d'étude :	29
II.7. Présentation de l'entreprise	30
II.8 La situation informatique de l'entreprise :	31
II.9.Répartition du matériel informatique (réseau) :	32
II.10.Caractéristiques du matériel informatique.....	34
Conclusion.....	35

Chapitre 3 : Analyse & Conception

III.1.Introduction :	36
III.1.1.Phase d'Initialisation :	36
III.1.2. Phase d'Elaboration :.....	36
III.1.2.1.Identification des acteurs :	37
III.1.2.2.Diagramme de contexte :	37
III.1.2.3.Représentation des diagrammes de cas d'utilisation :	38
III.1.2.4. définition des itérations :.....	40
III.1.3.Construction.....	40
III.1.3.1.réalisation de l'itération 1	40
III.1.3.1.1.Diagrammes de séquence pour itération 1:.....	40

III.1.3.1.2.Diagrammes de classe pour l'itération 1	46
III.1.3.2.réalisation de l'itération 2	47
III.1.3.2.1.Diagrammes de séquence pour itération 2:	47
III.1.3.2.2.Diagrammes de classe pour l'itération 2.....	53
III.1.3.3.réalisation de l'itération 3	54
III.1.3.3.1. Diagrammes de séquence pour itération 3:.....	54
III.1.3.3.2.Diagrammes de classe pour l'itération 3.....	58
III.1.3.4.réalisation de l'itération 4	59
III.1.3.4.1. Diagrammes de séquence pour itération 4:.....	59
III.1.3.4.2 : Diagrammes de classe pour l'itération 4.....	63
Conclusion :.....	64

Chapitre 4 : Réalisation

IV.1.Introduction :.....	65
IV.2. Outils de développement :.....	65
IV.2.1.Configuration logiciel:	65
IV.2.2 Configuration matérielle:	65
IV.2.3 Pour quoi oracle 10g :	65
IV.2.3.1 Qu'est-ce que Oracle Database 10g Express Edition?	66
IV.2.4.Le langage de programmation :.....	66
IV.2.4.1 Définition IDE :	66
IV.3.les Itérations :.....	68
IV.3.1.Itération I :	68
IV.3.2.Itération II :.....	68
IV.3.3.tération III:.....	70
IV.3.4.tération IV :	71
IV.4.Présentation de quelques interfaces :	72
Conclusion :.....	76
Conclusion générale:.....	77
Annexe A.....	78
Annexe B.....	95
Annexe C.....	102

Bibliographie

Chapitre 1 : généralités sur les réseaux et client/serveur

Figure I.1 : Topologie en bus.....	4
Figure I.2 : Topologie en étoile.....	4
Figure I.3: Topologie en anneau.	5
Figure I.4: Topologie en maillée.....	5
Figure I.5: Représentation des couches du modèle OSI	8
Figure I.6: Comparaison entre le modèle OSI et le TCP/IP.....	10
Figure I.7: Fonctionnement d'un système client/serveur	14
Figure I.8: Le client/serveur à deux niveaux	19
Figure I.9: Le client/serveur à deux niveaux	20
Figure I.10: Le client/serveur à trois niveaux.....	21

Chapitre 2 : Présentation de l'organisme d'accueil

Figure II.1 : localisation de l'entreprise.....	25
Figure II.2 : Organigramme générale de l'entreprise.....	28
Figure II.3: Organigramme de la direction des approvisionnements	29
Figure II.4 : l'emplacement des différents bâtiments.....	30

Chapitre 3 : Analyse & Conception

Figure.III.1: diagramme de contexte.....	37
Figure.III.2 : Diagramme de cas d'utilisation globale	39
Figure.III.3 : Diagramme de séquence du cas d'utilisation « Ajouter compte utilisateur »	41
Figure.III.4 : Diagramme de séquence du cas d'utilisation « Supprimer compte utilisateur »	43
Figure.III.5 : Diagramme de séquence du cas d'utilisation « Modifier compte utilisateur »	45
Figure.III.6. Le diagramme de classe pour l'itération 1.....	46
Figure.III.7: Diagramme de séquence du cas d'utilisation « Ajouter un fournisseur homologué »	47
Figure.III.8: Diagramme de séquence du cas d'utilisation « Consulter demande d'offre »	49
Figure.III.9: Diagramme de séquence du cas d'utilisation « Ajouter attestation de non transfert de fonds »	51
Figure.III.10. Diagramme de classes pour l'itération 2.....	53

Figure III .11 Diagramme de séquence du cas d'utilisation «Ajouter une facteur»	54
Figure III .12: Diagramme de séquence du cas d'utilisation «Supprimer une commande»	56
Figure.III.13.Diagramme de classespour l'itération 3.....	58
FigureIII.14 : Diagramme de séquence du cas d'utilisation «Ajouter demande de chèque»	59
FigureIII.15: Diagramme de séquence du cas d'utilisation «Supprimer déclaration en douane »	61
Figure.III.16.Diagramme de classes pour l'itération 4.....	63
Chapitre 4 : Réalisation	
FigureIV.1: NetBeans 7.0.1.....	67
FigureIV.2 : Page d'accueil	71
Figure IV.3 : Page Administrateur	72
Figure IV.4 : Modifie le mot passe.....	72
FigureIV.5 : ASAI	73
FigureIV.6 : Page Gestion des offres	73
FigureIV.7 :Page Gestion des commandes.....	74
FigureIV.8 :Page Gestion de l'homologation des fournisseurs	74
Figure IV.9 :Page agent service transit	75

Introduction

Générale

Introduction générale :

De l'âge de la pierre à nos jours, l'esprit perfectionniste de l'homme n'a cessé de lui permettre d'améliorer sa vie quotidienne. Le passage de la mécanique aux domaines d'informatique, d'électronique et d'automatique a révolutionné la vie journalière de l'être humain. Les nouvelles technologies de l'information et de communication illustrent ce phénomène. Aujourd'hui, vu l'intérêt croissant de vouloir gagner en temps, de conserver les données, de limiter le nombre d'employés et pas mal d'autres raisons, ont poussé petites, moyennes et grandes entreprises à chercher des solutions informatiques capables de répondre à leurs besoins. Le système d'information se doit être la représentation la plus fidèle possible de l'organisation ou de l'entreprise qu'il dessert. Il constitue l'image d'une réalité complexe, et il hérite par conséquent cette complexité. Pour remédier à ce problème des solutions informatiques ont été mises en œuvre et adoptées par les entreprises.

L'EI est l'une des entreprises qui a compris cette problématique, et qui travaille sans relâche pour mettre en œuvre cette nouvelle technologie de traitement, pour assurer une gestion plus fiable, plus rigoureuse, moins fastidieuse et pour minimiser le risque d'erreurs et assurer la disponibilité de l'information à toute éventuelle demande. Notre travail, s'inscrit dans cette démarche et consiste en la mise en place d'un système d'information pour la gestion des approvisionnements de l'EI. Pratiquement, il s'agit de développer une application client /serveur (2-tiers) en utilisant le langage NETbEANS et ORACLE 10.

Pour mener à bien notre travail nous allons le présenter en quatre chapitre qui sont :

Chapitre I : Généralités sur les réseaux et client-serveur

La première partie est consacré à la présentation de quelque généralités sur :les réseaux et c/s

Chapitre II : Présentation de l'organisme d'accueil

Dans cette partie nous allons présenter l'organisme d'accueil en précisant ses différentes directions ainsi que leurs fonctionnalités.

Chapitre III : Analyse et conception

La troisième partie est dédiée a la présentation de la démarche de modélisation pour le développement de notre application et ceci en étudiant quelques diagrammes de cas d'utilisation introduit par le langage UML.

Chapitre IV : Environnement de développement et réalisation

Dans cette dernière partie nous allons présenter les outils utilisés ainsi que quelque interface de l'application.

Résultat obtenus : le traitement automatique de l'information pour chaque service d'approvisionnement.

Chapitre I :
Généralité sur
les
réseaux
&
Client/Serveur

I.1. Les réseaux :

I.1.1. Introduction

Un réseau est un ensemble de moyens matériels et logiciels, géographiquement dispersés, destinés à offrir un service comme les réseaux téléphoniques assurer le transport des données ; Les réseaux devront véhiculer toute sorte d'information avec des contraintes temporelles plus au moins fortes avec des volumes extrêmement variés.

I.1.2. Généralité sur les réseaux [GUY PUJOLLE].

a) Un Réseau informatique: Un réseau informatique est un ensemble de composants matériels et logiciels, permettant d'assurer un service de communication.

b) Un protocole : C'est un ensemble de règles et de messages assurant un service de communication.

c) Nœud : c'est une station de travail, une imprimante, un serveur ou toute entité pouvant être adressée par un numéro unique. l'unicité de l'adresse est garantie par le constructeur d'une carte réseau qui donne un numéro unique ne pouvant être change par une personne.

d) Serveur : ordinateurs qui fournissent des ressources partagées aux utilisateurs par un serveur de réseau

e) Paquet : C'est la plus petite unité d'information pouvant être envoyé sur le réseau. Un paquet contient en générale l'adresse de l'émetteur, l'adresse du récepteur et les données à transmettre.

f) Trame : c'est une unité structuré de bit.

g) Clients : ordinateurs qui accèdent aux ressources partagées fournies par un serveur de réseau.

h) Support de connexion : la façon dont les ordinateurs sont reliés entre eux.

i) Données partagées : fichiers accessibles sur les serveurs du réseau Imprimantes et autres périphériques partagés : fichiers, imprimantes ou autres éléments utilisés par les usagers du réseau.

j) Ressources diverses : autres ressources fournies par le serveur.

I.1.3. Définition d'un réseau : [GUY]

Un réseau informatique est un système de communication destiné à relier des équipements informatiques variés, incluant ordinateurs, imprimantes et périphériques de toutes sortes, rendant ainsi possible le partage des ressources informatiques.

I.1.4.Objectifs des réseaux

Les réseaux permettent :

- Le partage de fichiers.
- Partage de périphériques.
- La communication entre personnes (grâce au courrier électronique, la discussion en directe...)
- Le transfert de données en général (réseaux informatique)
- Le partage d'applications (compilateurs, système de gestion de base de données).
- Le transfert de la parole, de la vidéo et des données.
- Gain du temps.

I.1.5.Types de réseau [MAG.98]**I.1.5.1.Selon la distance :****I.1.5.1.1.Les LAN (Local Area Network)**

Un LAN ou réseau local, est un réseau qui relie des ordinateurs et des périphériques situés à proximité les uns des autres, par exemple dans un même bâtiment. C'est le type de réseau le plus répandu dans les entreprises. Les LAN ne comportent généralement pas plus de 100 ordinateurs.

I.1.5.1.2 Les MAN (Métropolitain Area Network)

Un MAN ou un réseau métropolitain, est une série de réseaux locaux, il relie des ordinateurs situés dans une même ville ou une même agglomération.

I.1.5.1.3. Les WAN (Wide Area Network)

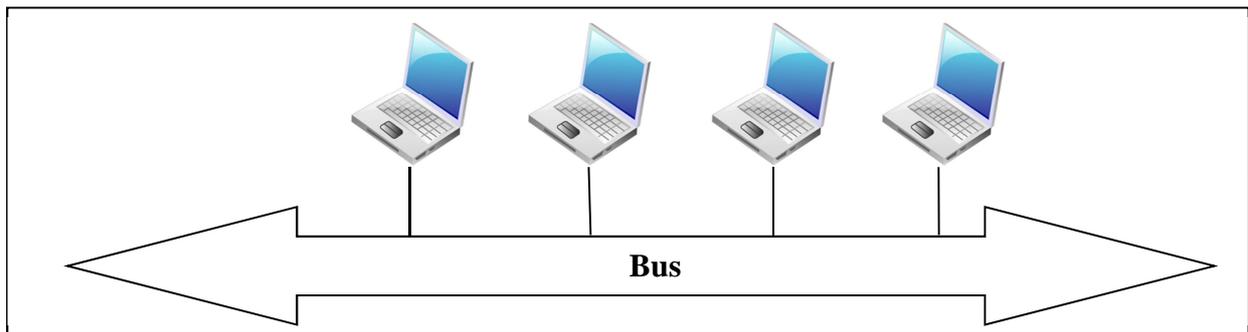
Un WAN ou réseau étendu, sert à relier des LAN et des MAN. Les réseaux qui composent un WAN peuvent être situés dans un même pays ou être dispersés dans le monde.

I.1.5.2. Selon la topologie : indique comment le réseau doit être conçu et organisé tant au niveau physique qu'au niveau logique.

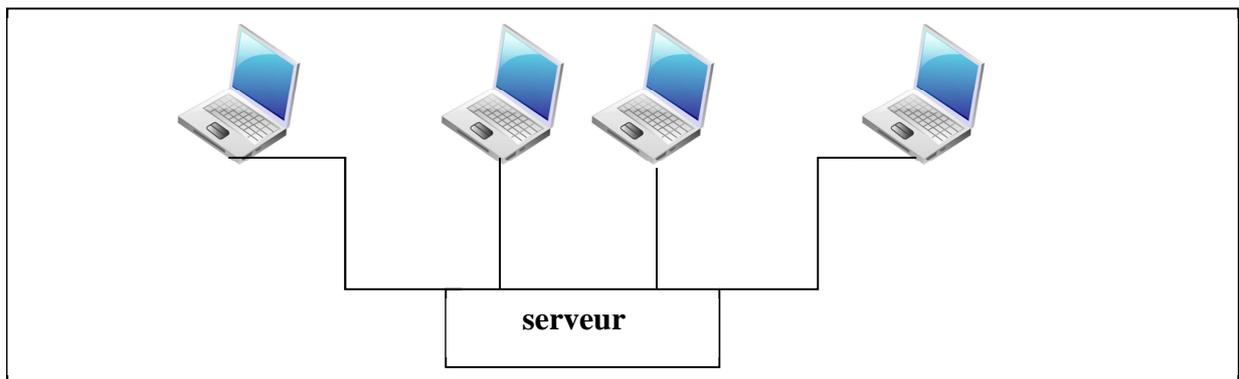
- Le niveau physique correspond aux éléments matériels tel que les ordinateurs, les câbles et les connecteurs qui constituent le réseau, c'est à ce niveau que l'on détermine à quel endroit les différents ordinateurs doivent être placés et comment les différents composants du réseau doivent être reliés les uns aux autres.
- Le niveau logique détermine comment les informations doivent circuler au sein du réseau.

I.1.5.2.1. Structure de réseau en bus

Sur un réseau en bus, les ordinateurs sont reliés par un même câble rectiligne ininterrompu, chaque extrémité du câble possède un bouchon de terminaison qui empêche le signal d'être renvoyé dans l'autre sens quant il arrive à l'extrémité de ce câble, ce qui pourrait provoquer des interférences. Sur un réseau en bus un seul ordinateur peut transférer des informations au même moment. Lorsqu'un ordinateur envoie des informations, celles-ci parcourent l'ensemble du câble, l'ordinateur de destination doit ensuite récupérer les informations à partir du câble.

**Figure I.1: Topologie en bus.****I.1.5.2.2. L'étoile :**

Dans un réseau en étoile, la forme physique du réseau ressemble à une étoile. Une image est plus parlante:

**Figure I.2 : Topologie en étoile**

I.1.5.2.3. L'anneau:

Un réseau en anneau a la forme d'un anneau. Cependant, la topologie physique d'un réseau en anneau est le bus. L'information circule dans une seule direction le long du support de transmission (bus). La topologie de type bus possédait un problème de collision de données : 2 machines ne doivent pas échanger des données en même temps, sinon elles s'entrechoquent. Ce principe est repris dans le réseau en anneau. Sauf que là, le système de token ring utilise la CSMA-CA, une méthode anticollision différente. Le principe est assez simple : une machine connectée au réseau possède un jeton virtuel. Ce jeton, c'est une autorisation de communiquer. Une fois que la machine a transmis ce qu'elle voulait, elle passe le jeton à la machine suivante, et ainsi de suite. Si le détenteur du jeton n'a rien à dire, il le passe au suivant.

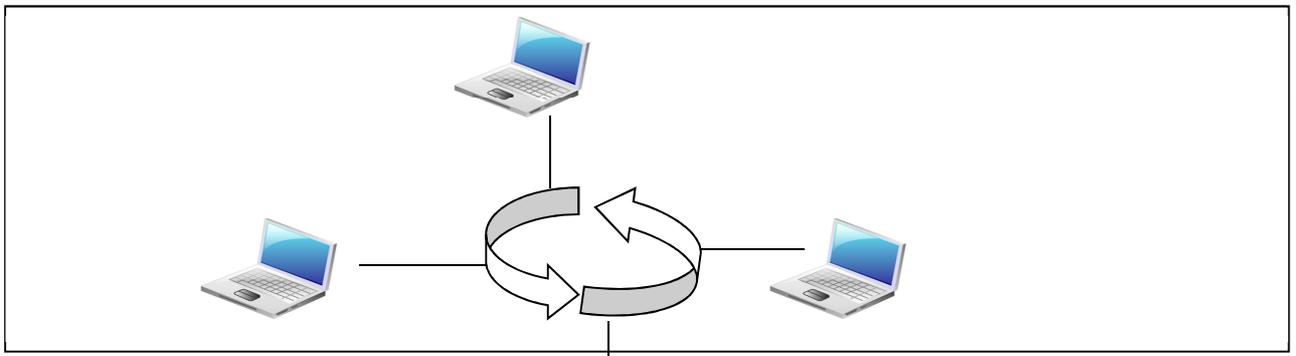


Figure I.3: Topologie en anneau.

I.1.5.2.4. Structure de réseau maillée

Dans une topologie maillée, chaque ordinateur est connecté à chacun des autres par un câble séparé. Son principal avantage, est sa capacité de tolérance de panne. En effet, lorsqu'un câble se rompt, il existe de nombreux autres itinéraires routés. Cette topologie est parfois très coûteuse.

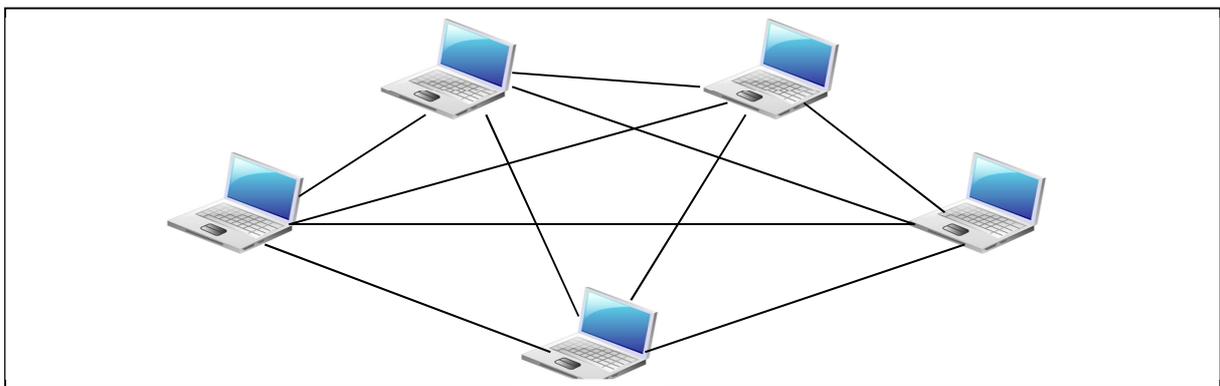


Figure I.4: Topologie en maillée.

I.1.6.Types de lignes de transmission[CHCK]

les lignes de transmission sont choisies selon la distance, le relief géographique et l'utilisation. Une ligne de transmission est l'élément essentiel pour véhiculer l'information d'un émetteur vers un récepteur. On distingue les types suivant:

I.1.6.1.Câbles métalliques: [CHCK]

Constitués de plusieurs paires de conducteurs métalliques (cuivre, bronze, aluminium...) isolé par du plastique, réunies entre elles pour former un câble, utilisés généralement sur des distances courtes, ce qui fait que c'est une solution économique mais limitée.

I.1.6.2. Paire torsadée :

Utilisée pour un câblage dit universel (exemple: téléphonie), il existe des câbles à 2 ou 4 paires, des câbles blindés (Shielded Twisted Pair ou STP) ou non blindés.

I.1.6.3.Câble coaxial:

Composé d'un câble central entouré d'un isolant, lui même recouvert d'une tresse métallique, qui est à son tour recouverte d'un isolant. Il permet des vitesses de transmission beaucoup plus élevées que la paire torsadée ainsi que des connections à plus grandes distances (~1km) , il reste néanmoins assez coûteux.

I.1.6.4. Fibres optiques:

Constituées de plusieurs fibres en verre permettant la transmission de la lumière, les informations sont transmises par le biais d'un courant électrique modulé, c'est un faisceau lumineux modulé qui est utilisé. L'avantage de ces câbles c'est qu'ils présentent un faible affaiblissement. Ils peuvent être utilisés sur de plus grandes distances (jusqu'à 15km) et sont aussi insensibles aux perturbations électromagnétiques.

I.1.6.5 Faisceaux hertziens:

Dans ce cas la transmission se fait par ondes dirigées entre deux antennes paraboliques. Ils supportent de grandes distances et de grandes capacités, pour une propagation en visibilité directe (entre 50 et 80 km), ils prolongent et remplacent les câbles, pour une plus grande souplesse mais aussi une plus grande sensibilité au bruit. Sa portée est de 10km (en fonction de la hauteur des antennes, etc...).

I.1.6.6. Liaison satellite:

Utilise des satellites géostationnaires (orbite à 36 000 km au dessus de l'équateur) en tant qu'antenne de transmission. Les débits accessibles aux utilisateurs : plusieurs Mbit/s. Délai de transmission relativement important (250 ms).

I.1.7. Logiciels de réseau

Les logiciels de réseau sont les programmes qui fonctionnent sur les ordinateurs d'un réseau tel que :

- Le système d'exploitation : c'est lui qui organise et administre toutes les activités du réseau.
- Le pilote de réseau : c'est lui qui permet à l'ordinateur d'utiliser sa carte réseau pour communiquer avec le réseau.
- Logiciels d'application : sont les logiciels qui fonctionnent sur les ordinateurs reliés au réseau, à titre d'exemple logiciels de traitement de texte.
- Logiciels serveurs : permettent au serveur d'effectuer des tâches spécifiques, à titre d'exemple, un logiciel serveur de courrier électronique du réseau permet au serveur de traiter le courrier électronique du réseau. Un serveur est un ordinateur qui met ses données et ses ressources à la disposition des autres ordinateurs du réseau.
- Logiciels d'administration : les réseaux qui relient plus de vingt ordinateurs nécessitent généralement un logiciel d'administration spécial. Le logiciel d'administration permet à l'administrateur système d'administrer et d'organiser le réseau plus efficacement

I.1.8. Les modèles de réseau [CCM 09]:

L'architecture du modèle de référence d'interconnexion des systèmes ouverts (ou modèle de référence OSI) est hiérarchisée en sept couches. Un système ouvert est un ordinateur, un terminal, un réseau, n'importe quel équipement respectant cette norme et donc apte à échanger des informations avec d'autres équipements hétérogènes et issus de constructeurs différents. Les 7 couches du modèle OSI

1. La **couche physique** fournit les moyens mécaniques, électriques, fonctionnels et procéduraux nécessaires à l'activation, au maintien et à la désactivation des connexions physiques destinées à la transmission de bits entre deux entités de liaison de données.
2. La **couche liaison** de données fournit les moyens fonctionnels et procéduraux nécessaires à l'établissement, au maintien et à la libération des connexions de liaison de données entre entités du réseau. Elle détecte et corrige, si possible, les erreurs dues au support physique et signale à la couche réseau les erreurs irrécupérables. Elle supervise le fonctionnement de la transmission et définit la structure syntaxique des messages, la manière d'enchaîner les échanges selon un protocole normalisé ou non.

3. La **couche réseau** assure toutes les fonctionnalités de relai et d'amélioration de services entre entité de réseau, à savoir : l'adressage, le roulage, le contrôle de flux et la détection et correction d'erreurs non réglées par la couche 2.
4. La **couche transport** assure un transfert de données transparentes entre entités de session et en les déchargeant des détails d'exécution. Elle a pour rôle d'optimiser l'utilisation des services de réseau disponibles afin d'assurer au moindre coût les performances requises par la couche session.
5. Les **couches session, présentation et application** constituent les couches hautes du modèle OSI(Open System Interconnexion) et offrent des services orientés vers les utilisateurs alors que les couches basses sont concernées

pas la communication fiable de bout en bout. Elles considèrent que la couche transport fournit un canal fiable de communication et ajoutent des caractéristiques supplémentaires pour les applications.

7- Application
6-Présentation
5-Session
4-Transport
3-Réseau
2-Liaison
1-Physique

Figure I.5 Représentation des couches du modèle OSI

I.1.9.Le modèle TCP/IP et ses couches [USER]

La **suite TCP/IP** est l'ensemble des protocoles utilisés pour le transfert des données sur Internet. Elle est souvent appelée **TCP/IP**, d'après le nom de deux de ses protocoles : TCP (Transmission Control Protocol) et IP (Internet Protocol).TCP/IP, comme son nom l'indique, est en fait constitué de deux protocoles TCP et IP.

TCP (Transmission Control Protocol) se situe au niveau transport du modèle OSI, il s'occupe donc d'établir une liaison virtuelle entre deux ordinateurs. Au niveau de l'ordinateur

émetteur, TCP reçoit les données de l'application dans un buffer, les sépare en datagrammes pour pouvoir les envoyer séparément, l'ordinateur distant (qui utilise le même protocole) à la réception doit émettre un accusé de réception, sans celui-ci, le datagramme est réémis. Au niveau de l'ordinateur récepteur, TCP réassemble les datagrammes pour qu'ils soient transmis à l'application dans le bon ordre.

IP (Internet Protocol) assure l'acheminement de chaque paquet sur le réseau en choisissant la route la plus appropriée. Pour pouvoir s'y retrouver IP va de pair avec un système d'adressage qui identifie de manière unique les réseaux traversés ainsi que chaque entité d'un réseau (appelé aussi nœud: ordinateur, routeur, ...).

La relation entre TCP et IP et la suivante, TCP fait passer à IP un datagramme accompagné de sa destination, IP ne s'occupe pas de l'ordre d'expédition, c'est TCP qui s'occupe de tout remettre en ordre, il se contente de trouver la meilleure route possible.

Souvent les termes « datagrammes » et « paquet » semblent identiques. En fait, on parle de datagramme lorsqu'il est question de TCP (couche 4 de l'OSI), le datagramme est l'unité de données. On parle de paquet pour les couches réseaux (3 IP) et liaison (2 et 1), c'est est une réalité physique, on peut les voir circuler sur le réseau. Généralement, un paquet contient seulement un datagramme, si bien que concrètement, il y a peu de différence entre les deux.

I.1.10.Comparaison entre le modèle OSI et le TCP/IP(chck)

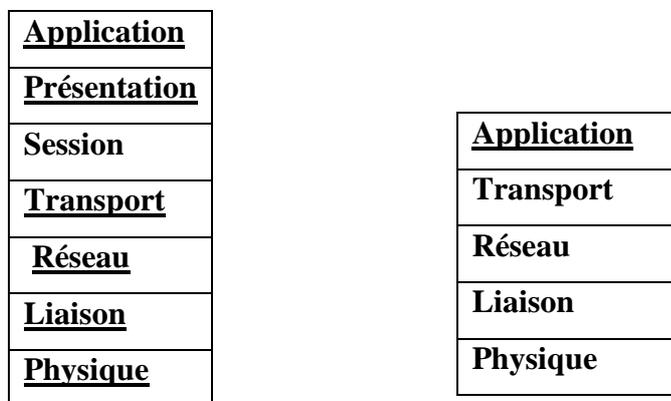
Le terme de couche est utilisé pour évoquer le fait que les données qui transitent sur le réseau traversent plusieurs niveaux de protocoles. Ainsi, les données (paquets d'informations) qui circulent sur le réseau sont traitées successivement par chaque couche, qui vient rajouter un élément d'information (appelé en-tête) puis sont transmises à la couche suivante. Ainsi les appellations changent suivant les couches: Le paquet de données est appelé message au niveau de la couche Application

- Le message est ensuite encapsulé sous forme de segment dans la couche Transport
- Le segment une fois encapsulé dans la couche Internet prend le nom de datagramme
- Enfin, on parle de trame au niveau de la couche Accès réseau Comme les suites de protocoles TCP/IP et OSI ne correspondent pas exactement, toute définition des couches de la pile TCP/IP peut être sujette à discussion...

En outre, le modèle OSI n'offre pas une richesse suffisante au niveau des couches basses pour représenter la réalité ; il est nécessaire d'ajouter une couche supplémentaire d'interconnexion de réseaux (*Internetworking*) entre les couches Transport et Réseau. Les

protocoles spécifiques à un type de réseau particulier, mais qui fonctionnent au-dessus de la couche de liaison de données, devraient appartenir à la couche réseau. ARP, et STP (qui fournit des chemins redondants dans un réseau tout en évitant les boucles) sont des exemples de tels protocoles. Toutefois, ce sont des protocoles locaux, qui opèrent au-dessous de la fonction d'interconnexion de réseaux ; placer ces deux groupes de protocoles (sans parler de ceux qui fonctionnent au-dessus du protocole d'interconnexion de réseaux, comme ICMP) dans la même couche peut prêter à confusion.

Le schéma qui suit essaie de montrer où se situent divers protocoles de la pile TCP/IP dans le modèle OSI de l'ISO :



Le modèle OSI

le modèle TCP/IP

Figure I.6 Comparaison entre le modèle OSI et le TCP/IP

I.2. Le client /serveur :

I.2.1.Introduction

Ce chapitre propose ensemble de l'architecture client-serveur et définit les notions de base indispensable à la compréhension du modèle.

L'enjeu pour les entreprises consiste à réaliser une intégration de l'informatique personnelle dans le système informatique d'entreprise avec les objectifs suivant :

- Tout utilisateur dans l'entreprises doit pouvoir accéder à toute information utile à sa tâche des lors que cet accès est autorisé par les règle de confidentialité et de sécurité vigueur.
- L'accès doit être instantané et doit pouvoir être fait à partir de n'importe quel poste de travail.
- L'accès à l'information doit avoir lieu par une interface que possible.

Une solution est apportée dans les années 90 : le modèle client /serveur (c /s). Une application est bâtie selon le modèle Client/serveur lorsqu'elle est composée de deux parties, coopérant l'une avec l'autre à la réalisation d'un même traitement. La première partie, appelée processeur client, est installée sur un poste de travail alors que la seconde, appelée processus serveur, est implantée sur un ordinateur (ou même des ordinateurs, éventuellement situés dans des lieux géographiques différents) chargé de rendre le service. Dans un environnement que le serveur, c'est un des principaux atouts de ce modèle. Donc, le Client/serveur est un mode de dialogue entre client et serveur, où le client est un processus qui demande l'exécution de services au serveur qui accomplit ces services, et envoie en retour des réponses.

I.2.2. Le modèle client /serveur [ARS 09]

I.2.2.1. Définition: L'architecte informatique client/serveur met en œuvre un ou plusieurs ordinateurs appelés clients, exécutant un programme applicatif communiquant avec un ordinateur distant appelé serveur, qui traite leurs requêtes.

Le client/serveur signifie que des machines clientes (des machines faisant partie du réseau) contactent un serveur, une machine généralement très puissante en terme de capacités d'entrée-sortie, qui leur fournit des services réseau à un autre ordinateur.

Généralement, la notion de serveur recouvre une activité de stockage d'informations. Mais l'un des rôles du serveur est de réagir à une requête en provenance d'un ordinateur. Cette requête concerne l'utilisation d'une ressource ou la récupération d'informations. En réponse à une requête, le serveur se charge d'assurer le partage de la ressource demandée ou initiée le transfert d'information vers le client. Le rôle de l'ordinateur client se borne généralement au lancement des requêtes et à l'affichage des informations obtenues.

I.2.2.2. Middleware

Un des composants clés de l'architecture client/serveur est le middleware qui est simplement un logiciel assurant la médiatisation entre clients et serveurs dans le cadre d'architectures de système hétérogènes. En d'autres termes c'est un ensemble de services logiciels construit au-dessus d'un protocole de transport afin de permettre l'échange des requêtes et des réponses associées entre clients et serveurs de manière transparente, permettant de cacher l'hétérogénéité des composants mis en jeu (réseaux, SGBD...).

I.2.3. Pourquoi le client /serveur :

Le concept du client/serveur s'impose dès lorsque l'on souhaite de décomposer l'exécution d'une application et faire en sorte que différentes machines au sens large (matérielle, logicielles) participent à l'exécution de l'application par opposition aux techniques de centralisées sur mainframe ou tout se fait au niveau du serveur central.

Les intérêts du Client/serveur sont multiples :

- ✚ **Exploitation** : il est alors facile de l'utiliser (prépare des entrées, comprendre les sorties d'un système ou d'un composant)
 - ✚ **Flexibilité** : il est facile de modifier un système ou un composant ;
 - ✚ **Interopérabilité** : Les systèmes ou les composant peuvent échange et utiliser plus facilement les information échange ;
 - ✚ **Mise à l'échelle** : il est alors facile de mettre à l'échelle le système ou le composant selon les facilement les information selon les dimension du problème à résoudre
- Ces intérêts engendrent un certain nombre d'avantage pour l'entreprise :

- ❖ **Mieux maitriser le système d'information** : Pour mieux maitriser un système d'information en le rendant plus intégré et évolutif il faut adopter des systèmes ouverts et éviter de s'enfermer sur un construction . L'intégration et évolutivité passent par l'architecture Client/serveur qui permet l'intégration de données et de traitement communs de l'entreprises au sein du serveur relationnel, et l'évolutivité des application développées sur client
- ❖ **Réutilisation d'applications existantes ou développées dans l'entreprise**: La réutilisation d'applications existantes ou développées dans l'entreprise, avec . . possibilité d'adaptation ou de spécialisation, est aussi un élément déterminant pour le choix d'une architecture client/serveur.
- ❖ **Prendre en compte les évolutions des technologies**: Au-delà du SI, le client/serveur apporte une modularité des composants matériels et logiciels . ce qui est très important.
- ❖ **Réduire les couts**: Le cout de fonctionnement en client/serveur peut s'avérer plus élevé qu'un système fonctionnant sur une architecture centrée serveur (mainframes) avec des client/serveur introduit un facteur de complexité et d'exploitation, mais l'arrivée de la version Internet du modèle comme modèle dominant.

-permettre le partage des données par plusieurs utilisateurs à partir de différents postes .
de travail.

-Décharger le réseau, puisque seules les requetés et les résultats correspondants sont véhiculés.

-Possibilité d'utiliser des outils non disponibles ou non adaptés à la machine serveur tels que les tableurs et les logiciels multimédia.

-Permettre une évolutivité des moyen informatiques.

I.2.4.Modes de fonctionnement :

- **Le mode connecté**

un « canal de communication » se crée entre le client et le serveur (à la demande du client), et les échanges (ordres, accusés de réception, données...) transitent par ce canal, ou alors via d'autres canaux ouverts pour l'occasion

ex : IMAP, FTP...

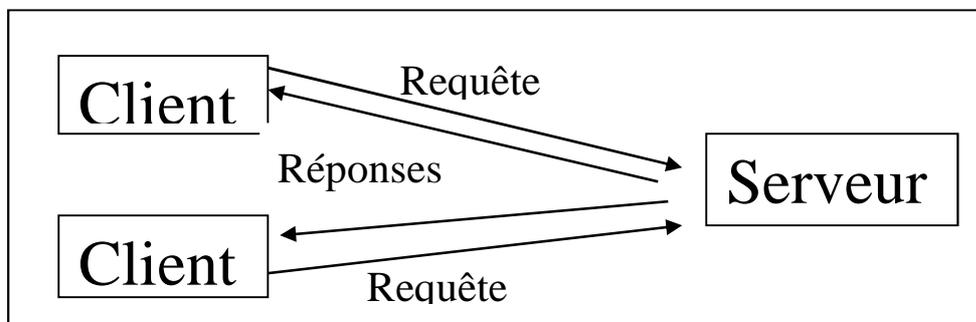
- **Le mode « datagramme », ou « échanges de paquets »**

il s'agit souvent de systèmes plus simples, où le client envoie sa requête (un ordre) dans un paquet, et le serveur lui répond dans un ou plusieurs paquets

ex: TFTP (Trivial File Transfert Protocole),http

I.2.5.Fonctionnement d'un système client/serveur : [CCM]

Un système client/serveur fonctionne selon le schéma suivant :



FigureI.7 Fonctionnement d'un système client/serveur

- **Serveur :** c'est un programme offrant un service sur un réseau (par extension, machine offrant un service).
- **Client :** On appelle logiciel client un programme qui utilise le service offert par un serveur. Le client envoie une requête et reçoit la réponse. Le client peut-être raccordé par une liaison temporaire.
- **Requête :** C'est un message transmis par un client à un serveur décrivant l'opération à Exécuter

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

I.2.6. Les protocoles: [GUP 05,DJADJN]

❖ Qu'est-ce qu'un protocole:

un protocole est une méthode standard qui permet la communication entre deux machines, c'est à dire un ensemble de règles et de procédures à respecter pour émettre et recevoir des données sur un réseau.

- Il en existe plusieurs selon ce que l'on attend de la communication :

✚ **http**: Le protocole HTTP est une implémentation de type client serveur des plus simples qui soit. Le client est généralement votre navigateur web (mozilla, konqueror...) mais l'objectif de cet article est de vous donner toutes les bases pour écrire vos propres clients. Celui-ci envoie une requête à un serveur (apache ...) qui lui répond. C'est un protocole en mode texte, généralement utilisé sur une connexion TCP sur le port 80, dédié au transfert de ressource. Il n'est pas dédié à l'échange de fichiers même si dans la pratique c'est souvent le cas avec les sites web.

✚ **TCP/IP** : Le TCP/IP est une suite de protocoles. Le sigle TCP/IP signifie «Transmission Control Protocol/Internet Protocol». Il provient des noms des deux protocoles majeurs de la suite de protocoles, c'est-à-dire les protocoles TCP/IP. Le TCP/IP représente d'une certaine façon l'ensemble des règles de communication sur internet et se base sur la notion d'adressage IP, c'est-à-dire le fait de fournir une adresse IP à chaque machine du réseau afin de pouvoir acheminer des paquets de données. La suite de protocoles TCP/IP a été créée elle est conçue pour répondre à un certain nombre de critères parmi lesquels :

- Le fractionnement des messages en paquets ;
- L'utilisation d'un système d'adresses ;
- L'acheminement des données sur le réseau (routage) ;
- Le contrôle des erreurs de transmission de donnée.

La suite de protocole TCP/IP est utilisée pour faire communiquer des réseaux reliant des ordinateurs de différents types de réseau. C'est sur cette suite de protocoles que repose le fonctionnement de l'Internet. Elle contient les protocoles suivants : HTTP, FTP, ARP, ICMP, TCP, UDP, IP, SMTP, TELNET...

✚ **LDAP** : Le protocole LDAP définit la méthode d'accès aux données sur le serveur au niveau du client, et non la manière de laquelle les informations sont stockées.

Donc il fournit :

- le protocole permettant d'accéder à l'information contenue dans l'annuaire,
- un modèle d'information définissant le type de données contenues dans l'annuaire,
- un modèle de nommage définissant comment l'information est organisée et référencée,
- un modèle fonctionnel qui définit comment on accède à l'information
- un modèle de sécurité qui définit comment données et accès sont protégés
- un modèle de duplication qui définit comment la base est répartie entre serveurs
- des APIs pour développer des applications clientes,

✚ **SSH (ou Secure SHell) :** est un protocole servant à créer une connexion sécurisée entre deux systèmes. Dans le protocole SSH, un ordinateur client établit une connexion avec un ordinateur serveur.

SSH offre les garanties de sécurité suivantes :

- Après avoir effectué une connexion initiale, le client peut s'assurer de se connecter au même serveur lors des sessions suivantes.
- Le client peut transmettre ses données d'authentification au serveur, telles que son nom d'utilisateur et son mot de passe, en format crypté.
- Toutes les données envoyées et reçues pendant la connexion sont transférées de façon chiffrée, ce qui les rend extrêmement difficiles à déchiffrer et à lire.

Puisque le protocole SSH chiffre tout ce qu'il envoie et reçoit, il peut être utilisé pour sécuriser des protocoles non sûrs. Grâce à la technique de *retransmission de port*, un serveur SSH peut être employé pour sécuriser des protocoles non sûrs,

✚ **SMTP (*Simple Mail Transfer Protocol*, traduisez *Protocole Simple de Transfert de Courrier*) :** est le protocole standard permettant de transférer le courrier d'un serveur à un autre en connexion point à point. Il s'agit d'un protocole fonctionnant en mode connecté, encapsulé dans une trame TCP/IP. Le courrier est remis directement au serveur de courrier du destinataire. Le protocole SMTP fonctionne grâce à des commandes textuelles envoyées au serveur SMTP (par défaut sur le port 25). Chacune des commandes envoyées par le client (validée par la chaîne de caractères ASCII CR/LF, équivalent à un appui sur la touche entrée) est suivi d'une réponse du serveur SMTP composée d'un numéro et d'un message descriptif.

✚ **Le protocole FTP :** permet d'échanger un fichier à la fois, dans les deux sens entre la machine client (celle qui a initié la connexion, donc la machine appelante) et la machine serveur (celle qui fournit le service FTP, donc la machine appelée). Le protocole FTP permet

aussi d'autres actions telles que la création et la suppression de répertoires (à condition qu'il soit vide), le listage des fichiers, la suppression et le renommage de fichiers, etc.

✚ **Le protocole Telnet :** est un protocole standard d'Internet permettant l'interfaçage de terminaux et d'applications à travers Internet. Ce protocole fournit les règles de base pour permettre de relier un client (système composé d'un affichage et d'un clavier) à un interpréteur de commande (côté serveur).

Le protocole Telnet s'appuie sur une connexion TCP pour envoyer des données au format ASCII codées sur 8 bits entre lesquelles s'intercalent des séquences de contrôle Telnet. Il fournit ainsi un système orienté communication, bi-directionnel (half-duplex), codé sur 8 bits facile à mettre en œuvre.

Le protocole Telnet repose sur trois concepts fondamentaux :

- Le paradigme du terminal réseau virtuel (NVT, *Network Virtual Terminal*) ;
- Le principe d'options négociées ;
- Les règles de négociation.

✚ protocole NNTP(Network News Transfer Protocol): NNTP (Network News Transfer Protocol) est le protocole d'échange des news ou forums de discussions à travers Usenet (nom donné au réseau logique constitué des serveurs de news disséminés sur la planète). Il assure l'échange des news entre les serveurs et également la communication entre serveur et postes clients aussi bien pour la lecture que pour l'écriture de messages. Ainsi, lorsqu'un utilisateur poste un article dans un groupe de news, il est dans un premier temps déposé sur le serveur de news auquel le poste client est relié. Puis, ce serveur va réexpédier cet article aux différents serveurs auxquels il est relié, qui eux-mêmes procéderont de la sorte. Ainsi, en quelques heures un message posté à Angers peut se retrouver sur un serveur de news en Australie. Mais, ce processus de diffusion systématique, n'est pas assuré pour tous les groupes de news existant au niveau mondial, car chaque serveur de news n'assure le relai que de certains groupes. En effet, il n'est peut-être pas très utile de diffuser sur les serveurs de news japonais le groupe fr.petites-annonces.automobiles :-). De plus, tout serveur de news fixe pour chaque groupe la durée de conservation des messages sur ses disques durs.

I.2.7. Caractéristique du client/serveur [ASG 08]

- ✚ **Service** : le modèle client/serveur est basé sur une relation entre des processus tournants sur des machines séparées. Le processus serveur est fournisseur de service. Le processus client est un consommateur de services. Le modèle établit ainsi une répartition claire des fonctions à partir de la notion de service.
- ✚ **Partage des ressources** : un serveur peut traiter plusieurs clients en même temps et contrôler leur accès aux ressources.
- ✚ **Transparence à la localisation** : les processus serveur et client peuvent résider sur la même machine ou par l'intermédiaire d'un réseau, sur deux machines différentes interconnectées. Le logiciel client/serveur masque aux clients la localisation du serveur en redirigeant les demandes de service si nécessaire. Un programme peut être client, serveur ou les deux.
- ✚ **Echange de messages** : client et serveur sont des systèmes à liaison épisodique qui interagissent au moyen de messages. Le message est le mécanisme d'émission des demandes de service et des réponses à celles-ci.
- ✚ **Asymétrie des protocoles** : la relation entre clients et serveur est de type plusieurs vers un. C'est toujours le client qui déclenche le dialogue en demandant un service. Les serveurs attendent passivement les requêtes des clients. Notez que, dans certains cas, un client peut transmettre une référence à un objet de type callback lorsqu'il invoque un service. Cela permet au serveur de rappeler le client, lequel devient alors un serveur.

I.2.8. Types de clients : [GEO 00]:

✚ **Client léger** :

Le poste client accède à une application située sur un ordinateur dit « serveur » via une interface et un navigateur Web. L'application fonctionne entièrement sur le serveur, le poste client reçoit la réponse « toute faite » à la demande (requête) qu'il a formulée.

✚ **Client lourd** :

Le poste client doit comporter un système d'exploitation capable d'exécuter en local une partie des traitements. Le traitement de la réponse à la requête du client utilisateur va mettre en œuvre un travail combiné entre l'ordinateur serveur et le poste client.

✚ **Client riche** :

Une interface graphique plus évoluée permet de mettre en œuvre des fonctionnalités comparables à celles d'un client "lourd". Les traitements sont effectués majoritairement sur le

serveur, la réponse "semi-finie" étant envoyée au poste client, où le client "riche" est capable de la finaliser et de la présenter

I.2.9. Les type serveur : [BNOK]

🚦 Serveur de fichiers :

Dans le cas de serveur de fichiers, le client requiert des enregistrements de fichiers en émettant des requêtes au serveur de fichiers. Les serveurs des fichiers sont utiles pour partager des fichiers sur un réseau et ils sont indispensables pour créer des banques de documents, d'images,...etc. Mais l'obtention de l'information nécessite de nombreux échanges de message sur le réseau.

🚦 Serveur de base de données :

Dans le cas de serveur de base de données, le client émet des requêtes SQL sous forme de message en direction du serveur, le résultat est renvoyé au client. Le serveur utilise sa capacité de traitement pour rechercher les données demandées au lieu de transmettre tous les articles au client et de le laisser en faire la sélection, donc la puissance est répartie et utilisée de façon beaucoup plus efficace.

🚦 Serveur groupware :

Le groupware s'intéresse à la gestion d'information semi-structurées tel que le texte, l'image, courrier, messagerie et ordonnancement de tâches.

🚦 Serveur d'application objet :

Dans le cas d'un serveur objet, l'application client/serveur est écrite sous forme d'un jeu d'objets communications. Les objets client communication avec les objets serveur au moyen d'un courtier d'objets ou ORB (Object Request Broker). Le client invoque une méthode sur un objet distant, l'ORB localise une instance de la classe, appelle la méthode demandée et renvoie les résultat à l'objet client.

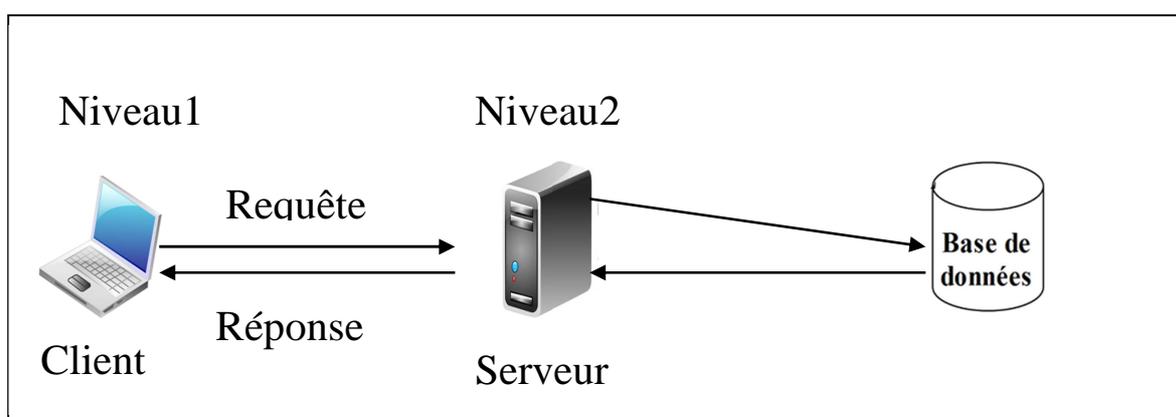
🚦 Serveur d'application web :

Le world wide web est la première application client/serveur, ce modèle consiste en des client léger, portable qui communiquent avec de très gros serveurs, et un serveur web qui renvoie des documents lorsque le client les demande.

I.2.10. Architecture client/serveur : [WIKIPÉDIA,ORD 99]

✚ 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 afin de fournir une partie du service.



FigureI.8 : Le client/serveur à deux niveaux

Dans cette architecture la technologie applicative est enfouie soit dans le serveur, soit dans le client et soit dans les deux.

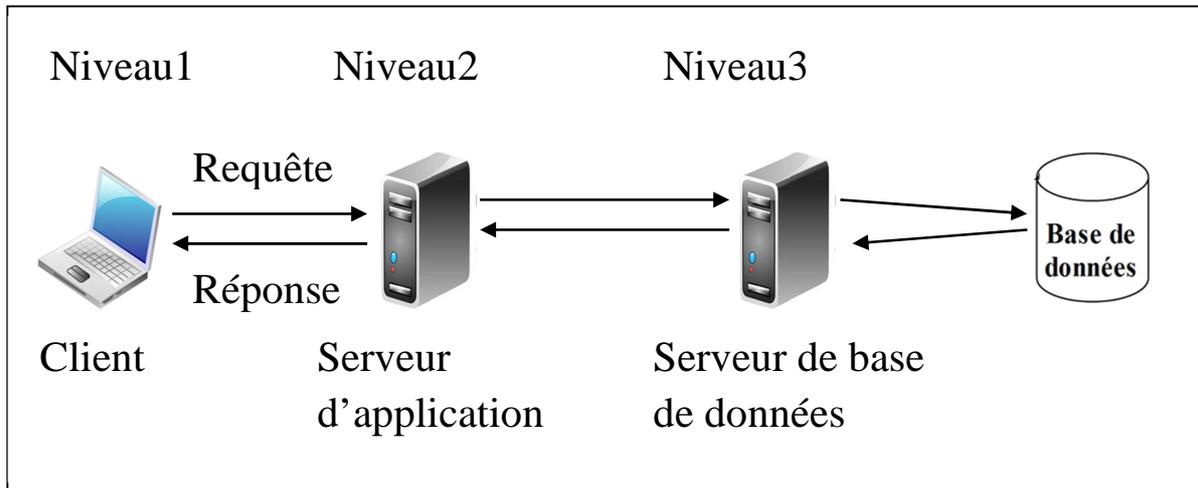
✚ Architecture client/serveur à 3 niveaux :

Cette architecture est généralement constituée par :

- **Un client** : qui demande la ressource.
- **Un serveur d'application** : (middleware) c'est le serveur chargé de fournir la ressource mais en faisant appel à un autre serveur.
- **Un serveur secondaire** : c'est celui qui fournit le service au premier serveur (souvent un serveur de base de données). Dans cette architecture la logique applicative réside dans le niveau intermédiaire et est séparée des données et de l'interface utilisateur. Les serveurs d'application web sont un bon exemple de système client/serveur à trois niveaux, L'architecture internet se décompose en trois niveaux logiques comme suit :

1. l'interface utilisateur gérée par le langage html (hyper texte markup langage) et éventuellement enrichie par l'emploi de html dynamique et de composants java ou ActiveX.

2. serveur web ou http(hyper texte transfert protocole)
3. le serveur de données



FigureI.9: Le client/serveur à deux niveaux

🚦 Architecture à multi niveaux :

Dans l'architecture à 3 niveaux, chaque serveur 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...

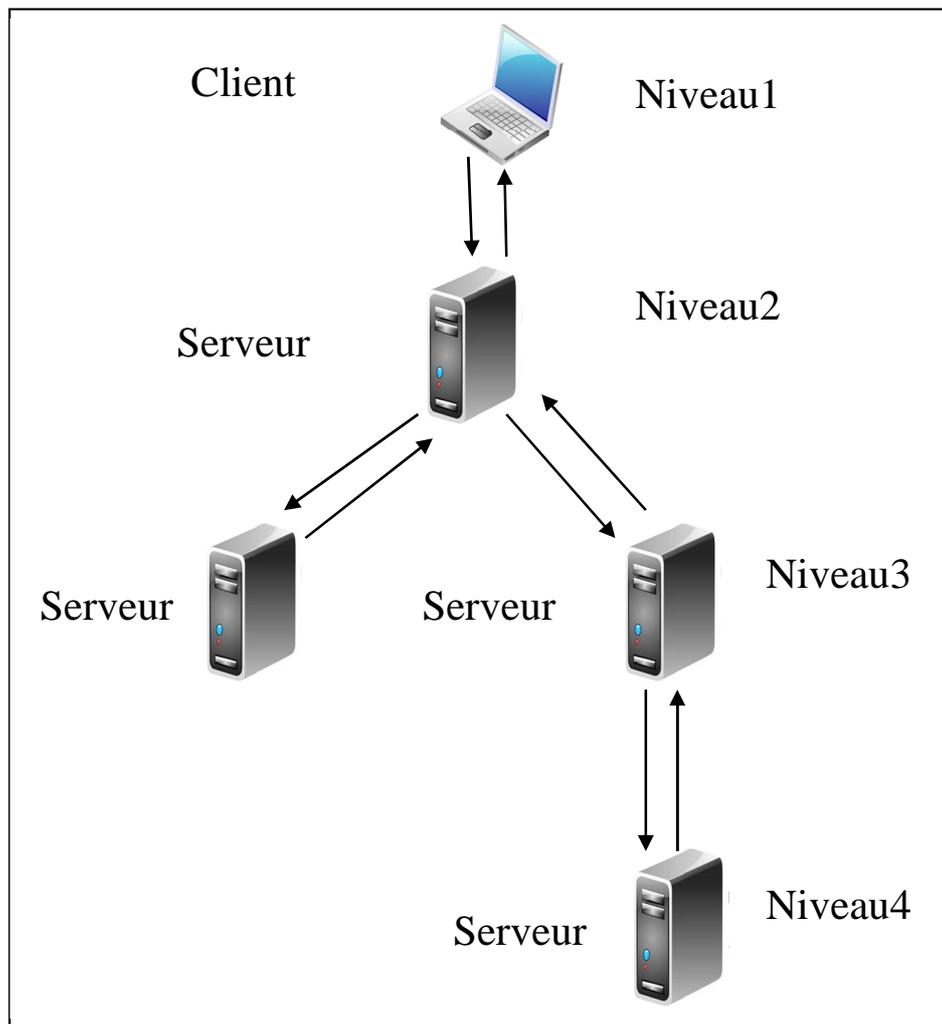


Figure I.10: Le client/serveur à trois niveaux

I.2.11. Avantages de l'architecture client/serveur : [GEO 00]:

- Maintenance matériel minimale.
- Grande vélocité sur des grands volumes de données et de traitements.
- Toutes les données sont centralisées sur un seul serveur, ce qui simplifie les contrôles de sécurité, l'administration, la mise à jour des données et des logiciels.
- Les technologies supportant l'architecture client-serveur sont plus matures que les autres.

- La complexité du traitement et la puissance de calculs sont à la charge du ou des serveurs, les utilisateurs utilisant simplement un client léger sur un ordinateur terminal qui peut être simplifié au maximum.

I.2.12. Inconvénients de l'architecture client/serveur :

- Utilisation de langages de programmation anciens.
- Calcul scientifique complexe impossible.
- Si le serveur n'est plus disponible, plus aucun des clients ne fonctionne (le réseau pair-à-pair continue à fonctionner, même si plusieurs participants quittent le réseau).
- Les coûts de mise en place et de maintenance peuvent être élevés.

Conclusion :

Dans ce chapitre, on a essayé de présenter des généralités sur les réseaux : leurs différents types, modèles de réseau ...etc. et on a présenté un petit aperçu sur que nous avons présenté le paradigme client /serveur, son intérêt, sa mise en œuvre et son fonctionnement.

Dans le chapitre suivant, nous allons essayer de présenter l'organisme d'accueil et de mettre en évidence notre problématique ainsi que les objectifs de notre travail.

Chapitre II :

Présentation

de l'organisme

d'accueil

II.1. Historique :

Electro-industrie est issue de la réorganisation du secteur industriel opéré en Algérie entre 1980 et 2000 qui a conduit en 1999 à la restructuration de l'ancienne **ENEL** (entreprise National des industries Electrotechniques) un certain nombre d'**EPE/SPA**, parmi les quelles figures **ELECTRO-INDUSTRIES**.

L'usine a été réalisée dans le cadre d'un contrat produit en main avec des partenaires ALLEMAND, en l'occurrence, SIEMENS pour le produit et FRITZ WERNER pour l'engineering et la construction, l'infrastructure est réalisée par les entreprises algériennes **ECOTEC, COSIDER** et **BATIMETAL**.

La production prévue initialement porte sur :

- 1500 Transformateurs de 50 à 1600 KVA
- 50.000 Moteurs asynchrones triphasés de 0,18 à 400 KW
- 2.000 Alternateurs 17,5 à 200 KV

La capacité de production de transformateurs de l'entreprise couvre les besoins du marché national à 70% environ. Les ventes de moteurs représentent 30% environ de capacité de production, il est à signaler qu'Electro-Industrie est la seule entreprise fabriquant ces produits en Algérie. En matière de qualité, l'entreprise dispose de ses propres laboratoires d'essai et mesure, de ses produits ainsi pour le *contrôle des principaux matériaux utilisés dans sa fabrication*.

EPE : entreprise publique économique.

SPA : société par action.

Entrée en production :

Le lancement de la production du secteur transformateurs a lieu en 1985, quant au secteur de moteurs/alternateurs n'a pas commencé jusqu'à l'année 1986. La production a été au départ sous licence SIEMENS jusqu'en 1992.

Evolution en 1998 :

En 1998 l'entreprise a connu des améliorations au niveau de production et de dont on cite par Exemple :

- Extension des capacités de production de transformateurs de 1500 à 5000 unités/an
- Développement de la gamme de moteurs monophasés
- Développement de moteurs destinés à la climatisation
- Extension verticale de la gamme de transformateurs (2000Kva)
- Extension horizontale de la gamme du moteur en types et variantes.

II.2. Localisation de l'entreprise :

L'entreprise se situe sur la route nationale N°12 à AZAZGA à 35Km à l'Est du chef-lieu de la wilaya Tizi-Ouzou et à 150Km de la capitale Alger.

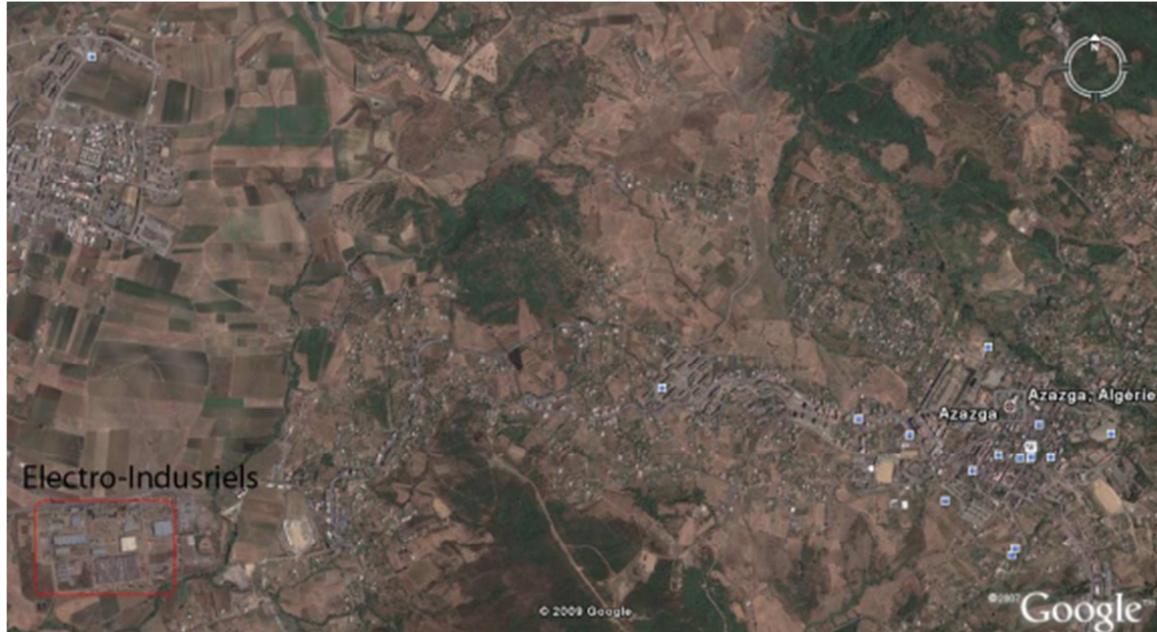


Figure II.1 : localisation de l'entreprise.

II.3. Description de la structure :

La structure d'une entreprise est un élément primordial (essentiel), elle conditionne son bon fonctionnement, son développement et sa réactivité à l'environnement économique. ELECTRO-INDUSTRIES est placée sous l'égide de la société de gestion des participations câbleries et matériels électrique.

Electro-industries compte des directions, des départements, et des assistants:

- ❖ **La direction générale** : son rôle est la coordination des actions des différentes directions ,le suivi et l'évaluation de la réalisation des objectifs tracés.
- ❖ **La direction qualité développement** : son rôle est de coordonner les activités liées au développement des produits et de la qualité au sein de l'entreprise dont l'objectif est de superviser et formaliser la préparation des dossiers du projet de développement des produits en liaison avec les structures concernées de l'entreprise et s'informer des nouvelles techniques visant à une meilleure maîtrise de la qualité et étudie les possibilités d'adaptation de l'entreprise.

- ❖ **Direction de l'administration et de l'emploi** : elle a plusieurs tâches parmi elles :
 - ✚ Gère les ressources humaines
 - ✚ Comptabiliser l'achat et la distribution des fournitures de bureau pour l'ensemble des structures de l'entreprise.
 - ✚ Gérer le magasin des fournitures de bureau et tenir à jour le fichier fournisseur et suivre les prix.
- ❖ **Directeur finance et comptabilité** : ses tâches principales sont d'établir et suivre un planning finance et veiller à ce que l'équilibre financier de l'entreprise ne soit pas mis en péril et s'assurer que l'entreprise dispose toujours d'une trésorerie suffisante pour régler les dettes et les achats de marchandises.
- ❖ **Directeur approvisionnement** : il a pour mission de négocier avec les fournisseurs et acheter la matière première pour la production.
- ❖ **Directeur unité transformateur** : elle a pour mission de la fabrication et de la Commercialisation des transformateurs.
- ❖ **Directeur unité moteur et prestation** : elle a pour mission de la fabrication et de la commercialisation des moteurs et alternateurs et assurer la maintenance des machines et équipements de l'entreprise. En plus des directions et des départements l'organisme dispose de trois organes d'assistance et un service d'informatique.
Ces trois Assistants sont :
 - ✚ Assistant de communication et relations sociales.
 - ✚ Assistant contrôle de gestion et planification
 - ✚ Assistant juridique.

II.4. Mission et objectif de l'entreprise :

a- La mission de l'entreprise : l'entreprise a pour mission d'assurer la production, le montage et la commercialisation des produits fabriqués (transformateurs, moteurs électriques).

b- L'objectif de l'entreprise : l'Electro-Industries vise :

- L'amélioration de la qualité des produits.
- Etablissement des besoins annuel aux matières premières pour satisfaire le programme production.
- Le développement de la recherche dans les différentes branches de l'entreprise.
- Amélioration de la maintenance de l'outil de production.

Pour pallier aux problèmes de la gestion existante, nous comptons doter la direction des approvisionnements d'un système d'information assurant la gestion des approvisionnements de façon automatique. Ce système permettra :

- La prise en charge de toutes les opérations d'approvisionnement depuis l'émission de la demande d'approvisionnement(DA) jusqu'à l'entrée de la marchandise au magasin ;
- Le suivi des stocks ;
- La facturation ;
- La recherche des fournisseurs et leurs homologations ;
- Le règlement ;
- Le dédouanement ;

Vu la distribution des traitements sur les différents services, de la direction des approvisionnements, une solution réseau s'impose.

L'objectif de notre étude est donc de mettre en place une application intranet pour la gestion des approvisionnements qui sera déployée sur le réseau local *de l'entreprise*.

II.5. Organigramme générale de l'entreprise :

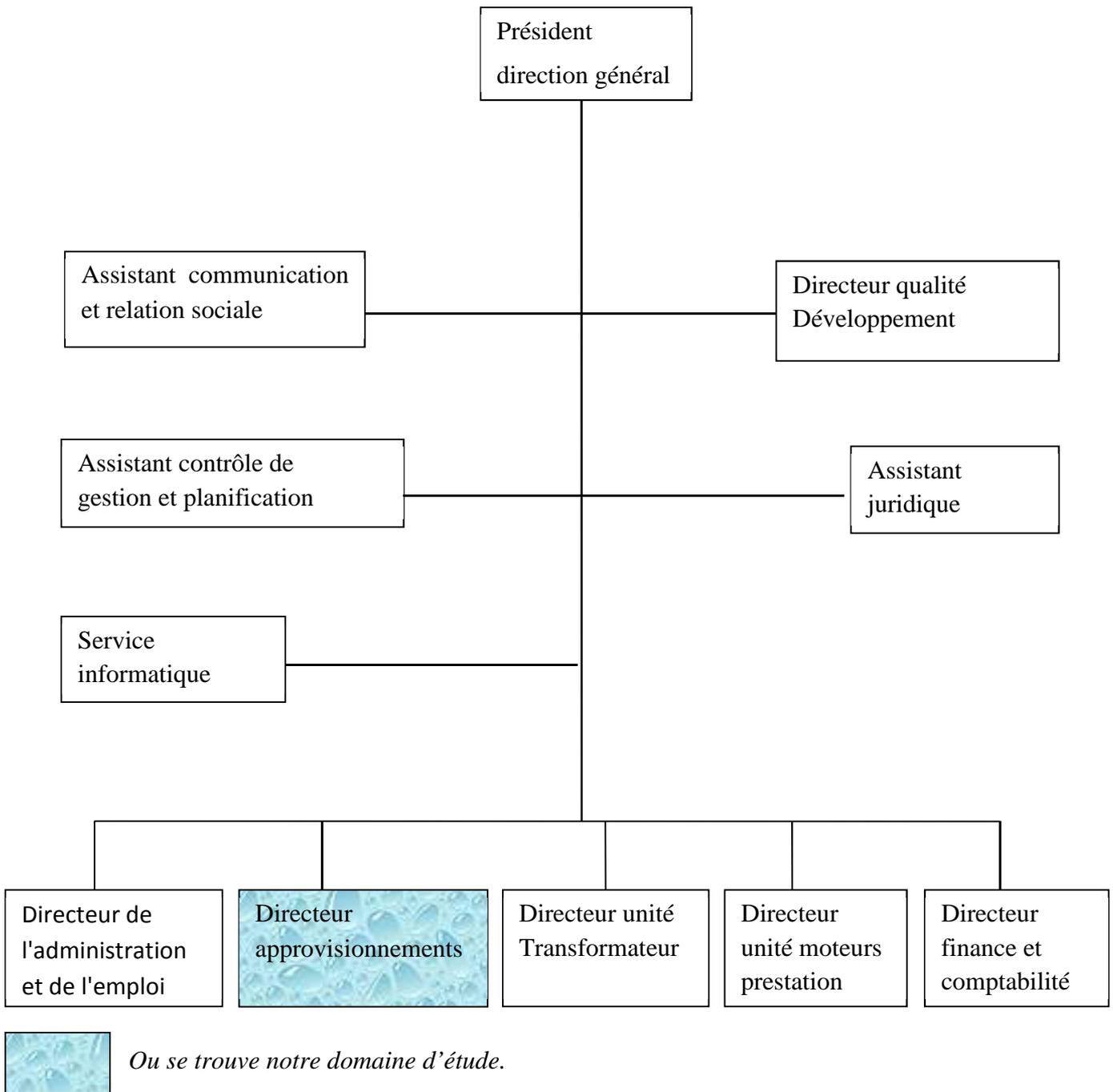


Figure II. 2 : Organigramme générale de l'entreprise

II.6. Présentation du champ d'étude :

II.6.1. Organigramme détaillé du champ d'étude :

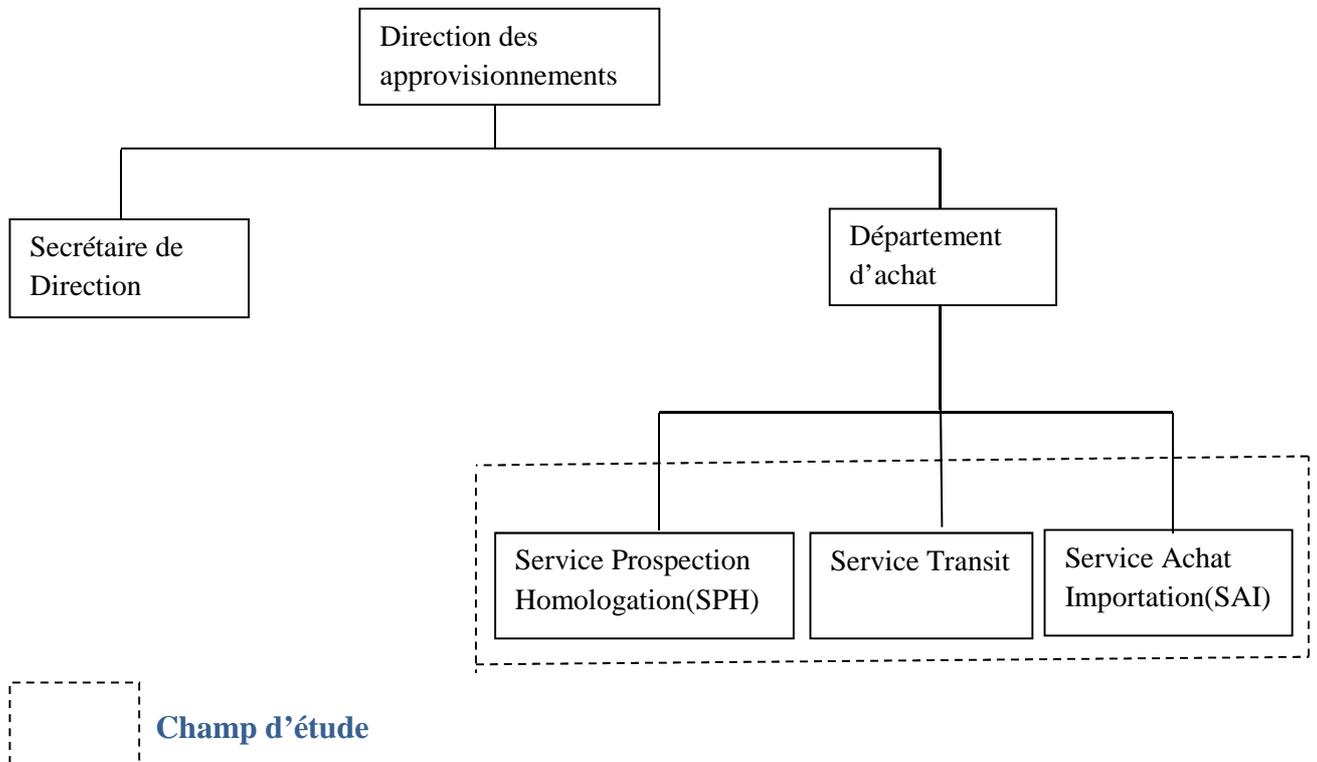


Figure II.3: Organigramme de la direction des approvisionnements

II.6.2 Présentation des différents services du champ d'étude :

- **La direction des approvisionnements:** sa principale tâche est de superviser et de piloter toutes les opérations d'approvisionnement en matières premières (visseries, acier, gaine, cuivre...).
- **La secrétaire de direction : a pour mission de :**
 - Assurer la gestion des courriers départ et arrivée d'une part, et des communications téléphoniques d'autre part;
 - Organiser les réunions de travail.
- **Département achat :** constitué de trois services:
 - Service achats/ importations, service transit et service prospection/homologation.
- **Le service achats/importations :** a pour mission de :
 - ❖ Suivre l'exécution des commandes et contrats ;

- ❖ Veiller au bon suivi des dossiers d'achats ;
- ❖ Exécution des programmes d'approvisionnement dans les meilleures conditions d'efficacité, délais, coûts ... en évitant les ruptures de stock.
- **Le service transit** : sa mission consiste à :
 - ❖ Effectuer le retrait des documents concernant les expéditions ;
 - ❖ Procéder à l'ouverture des dossiers transit ;
 - ❖ Veiller à l'application correcte des règlements et des procédures douanières
- **Le service prospection homologation** : a pour mission :
 - ❖ Recherche, dans le cadre de la Mis en œuvre des programmes d'approvisionnement, de nouveaux fournisseurs, au niveau international, susceptibles de fournir les meilleurs produits dans les meilleures conditions de délais et de coûts ;
 - ❖ Coordonner et évaluer les opérations d'homologation de produits avec les fournisseurs et les services techniques ;
 - ❖ Mettre en place ces programmes (lancement des appels d'offres ou consultations) en respect à la législation et à la réglementation en vigueur.

II.7. Présentation de l'entreprise

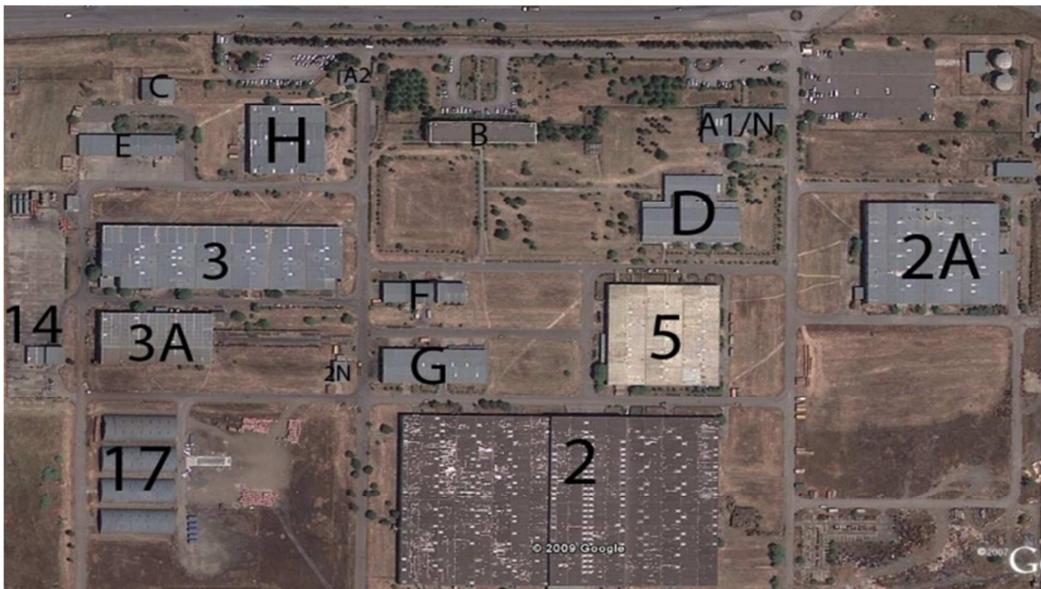


Figure II.4 : l'emplacement des différents bâtiments.

- A2 : service commercial.
- A1/N : Poste de sécurité, CMS
- B : bloc administratif.
- C : central électrique.
- D : restaurant.

- E : Parc auto/ Sécurité et intervention.
- F : chaufferie.
- H : Réparation transformateur, montage des groupes électrogènes.
- G : magasin central.
- 1 : Magasin.
- 2 : Unité des moteurs.
- 2A : Unité des moteurs.
- 3 : Unité des transformateurs.
- 3A : Unité des transformateurs.
- 5 : Atelier central.
- 14 : expédition des transformateurs.
- 17 : hangar de stockage.
- 2N : station de neutralisation.

Cette photo de Google Earth montre l'emplacement des différents bâtiments et tous les obstacles qu'ils y a entre les bâtiments.

II.8 La situation informatique de l'entreprise :

1.Organisation de la fonction informatique :

Le service informatique est constitué dans son état actuel de trois bureaux :

- ✓ Le bureau développement et études
- ✓ Le bureau gestion de production et administration réseau
- ✓ Le bureau exploitation.

2.Ressources humaines affectées :

L'effectif total affecté est de 05 agents dont :

- ✓ 01 responsable informatique
- ✓ 02 informaticiens pour le bureau développement et études
- ✓ 02 informaticiens pour le bureau exploitation.

3.Le matériel informatique existant :

L'entreprise est dotée de deux systèmes informatiques, l'un de type HP3000-68(mini-ordinateur) acquis en 1986, l'autre de type micro-ordinateur installé récemment.

3.1.HP-3000 série 68 :

- Système HP-3000 série 68 avec 8 Mo de RAM.

- Un dérouleur de bandes magnétiques.
- Une imprimante 900 lignes /minute.
- Trois unités de disques de 572 Mo chacune.
- Une cinquantaine de terminaux.

3.2.Réseau de micro :

- Trois serveurs dont deux de moyenne capacité, l'autre d'une grande capacité.
- 76 micro-ordinateurs (station de travail).
- 28 imprimantes dont :
 - 18 imprimantes matricielles(Epson).
 - 10 imprimantes laser (de marque HP).

4.Les logiciels utilisés : les logiciels disponibles au centre informatique sont :

- ✓ Système d'exploitation.
- ✓ Système de gestion des bases de données ;
- ✓ Compilateur : Cobol, Basic, B Basic, Delphi ;
- ✓ Utilitaires : tri, fusion, coupée, éditeur ;
- ✓ Applications existantes :
 - Commande et contrat .
 - Gestion de matière.
 - Gestion de production .
 - Gestion de personnels .
 - Comptabilité générale.

II.9.Répartition du matériel informatique (réseau) :

- ✓ Direction générale

AFFECTATION	MICROS PIII	IMPRIMANTES
Direction générale	02	01
Direction de l'administration et de l'emploi	08	04
Direction approvisionnement	05	03
Assistant contrôle de gestion et planification	03	02
Direction finance et comptabilité	10	04
Direction Informatique	07	03

- ✓ Unités de transformateur

Affectation	Micro-ordinateur	Imprimantes
Direction de l'unité	02	01
Gestion du personnel	01	-
Département production et technique	01	01
Bureau d'étude	02	01
Méthodes	01	-
Ordonnancement	01	-
Control qualité	02	-
Commerciale	02	02

- ✓ Unité moteurs alternateurs

Affectation	Micro-ordinateur	Imprimantes
Direction de l'unité	01	01
Gestion du personnel	01	-
Bureau d'étude	02	-
Control qualité	02	01
Approvisionnement commercial	03	-
Facturation	01	01
Méthodes	01	-
Lancement ordonnance	01	-

- ✓ Unité prestation technique

AFFECTATION	MICROS PIII	IMPREMANTES
Direction de l'unité	02	01
Planning maintenance	01	/
Normalisation	01	/
Méthodes de fabrication	01	/
Finances	01	01
Dept.Energie (DMU)	01	/
Dept.Etudes&Fabrication	01	/
Secrétariat	01	01
Laboratoire	02	/
Magasins 5 et G	02	/
Parc machines	01	/

II.10. Caractéristiques du matériel informatique

Caractéristiques techniques du serveur :

- Windows 2000NT comme système d'exploitation.
- Architecture biprocesseur.
- Bus système Intel Pentium.800MHzcache 256 Ko.
- RAM de 256 Mo ECC 133MHz.
- Contrôleur graphique ATI-TAGE.
- Interface réseau Intel pro/100 + (10/100Mpps).
- Lecteur CD-ROM IDE 52X.
- Lecteur HP sure store DAT 24 512/24Go).
- Deux disques durs host Swap Ultra 160 SCSI 7 KRPM, 10 Go.
- Lecteur de disquette 3'' 1/2.
- Clavier + souris.
- Moniteur 15 « SVGA ».
- Graveur.
- Alimentation redondante Host Swap.
- Imprimantes :

Lieu	Type	Marque	Format	Nombre d'imprimantes
Service informatique	Laser	HP 1200 HP 4V	A4	03
Directeur finance et comptabilité	Matricielles	EPSON LQ-2080	132	04
Directeur unité transformateurs	Matricielles	EPSON LQ-2080	132	04
Directeur unité prestation	Laser	HP jet 1200	132	06

Conclusion :

Dans ce chapitre nous avons présenté notre organisme d'accueil en spécifiant sa structure organisationnelle et des services avec ses fonction.

Dans le chapitre qui suit, nous allons présenter *Analyse & Conception* de notre application.

Chapitre *III*

Analyse et Conception

III.1.Introduction :

L'étude de l'existant nous a permis de collecter les informations qui concernent le fonctionnement du système existant, et ainsi de connaître les besoins et les orientations des utilisateurs.

Pour le développement de notre application de gestion des approvisionnements, nous avons opté pour une démarche de conception orientée objet (RUP, Rational Unified Process), en se basant sur la modélisation UML. Pour cela, nous suivons les phases suivantes.

Initialisation : vision approximative de la finalité du projet, cas d'entreprise, périmètre, estimation globales.

Elaboration : vision plus élaborée, implémentation itérative de l'architecture noyau, résolution des risques élevés, identification de la plupart des besoins et du périmètre réel, estimation plus réalistes.

Construction : implémentation itérative des éléments qui présentent des risques et une complexité moindres et préparation du déploiement.

Transition : Béta tests, déploiement.

III.1.1.Phase d'Initialisation : une analyse approximative de notre projet permet de desseller les points suivants :

- ✓ Notre application touche à la gestion d'approvisionnement qui est un domaine connu,
- ✓ Notre application est de type Client/serveur, ce genre d'applications a connu déjà un succès considérable et ces dernières se ressemblent d'un point de vue outil et langages.
- ✓ Notre application est isolée à internet, le risque de sécurité ne présente pas un énorme problème.

Vu les points précédents de notre analyse de surface, nous pouvons dire que ce projet est abordable

III.1.2. Phase d'Elaboration :

Notre projet porte sur la mise en place d'une application client/serveur 2-tiers, pour l'automatisation des activités de la gestion des approvisionnements de L'ELECTRO-INDUSTRIES.

Dans le but de maîtriser 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

des approvisionnements de L'ELECTRO-INDUSTRIES, leur permettant de travailler et de communiquer mutuellement, dans l'objectif de la gestion, et ce via un réseau local (intranet)

III.1.2.1. Identification des acteurs :

Dans le stage que nous avons fait nous a permis de procéder à l'identification de ces principaux acteurs qui seront les futurs utilisateurs de notre application.

- ✓ **Administrateur(ADMIN)** : il gère les comptes utilisateur.
- ✓ **Responsable Service Prospection/ Homologation(RSPH)** : Coordonner et évaluer les opérations d'homologations de produits avec les fournisseurs et les services techniques.
- ✓ **Agent Service Achats/ Importations (ASAI)** : Vérification et contrôle des dossiers commande.
- ✓ **Agent Service Transit (AST)** : Surveiller l'application correcte des règlements et des procédures douanières.

III.1.2.2. Diagramme de contexte :

Le diagramme de contexte est un modèle conceptuel de flux qui offre une vision globale des interactions entre le système et l'environnement extérieur. Il permet aussi de bien délimiter le champ de l'étude. Pour notre cas, le diagramme de contexte est donné par la figure suivante .

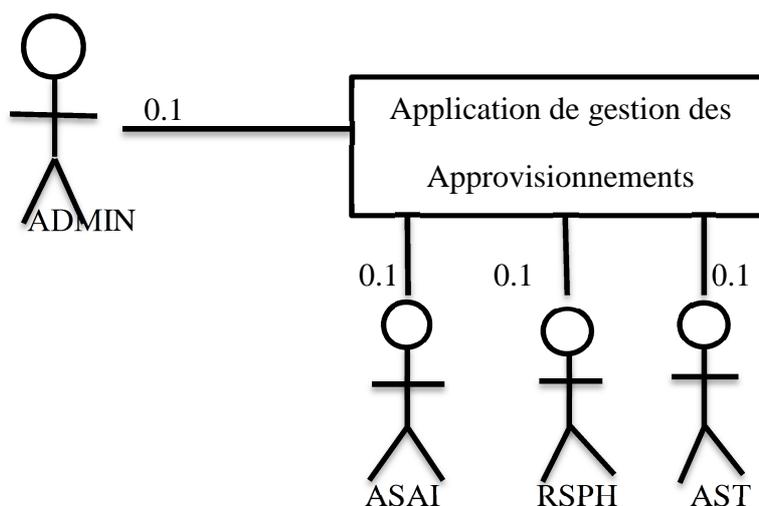
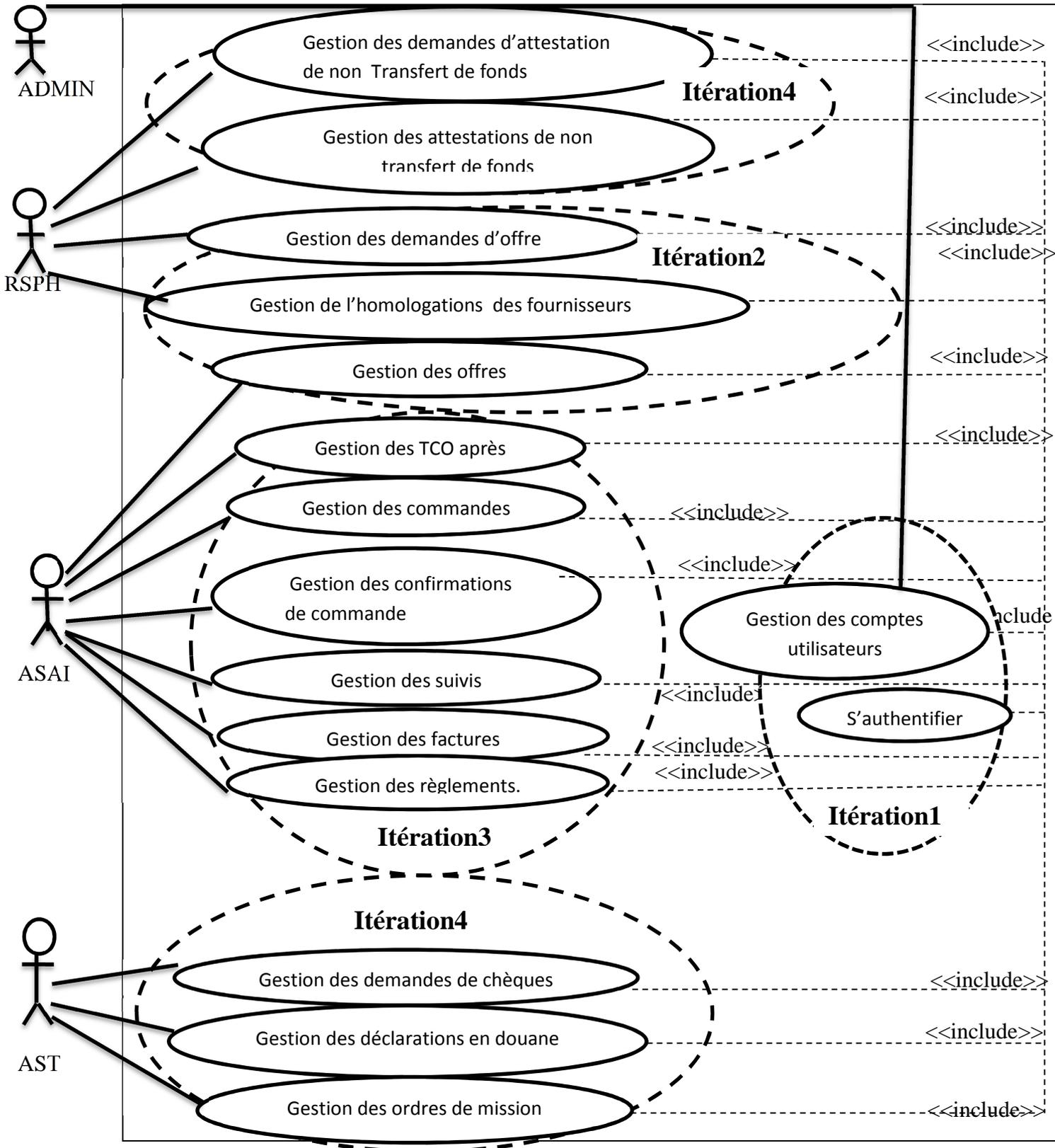


Figure.III.1: diagramme de contexte

III.1.2.3.Représentation des diagrammes de cas d'utilisation :

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet, mais pour le développement, les cas d'utilisation sont plus appropriés. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Il est une unité significative de travail. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs, ils interagissent avec les cas d'utilisation (use cases) les comportements sont modélisés sous la forme de cas d'utilisation via un diagramme de cas d'utilisation

✓ Représentation cas d'utilisation globale :



FigureIII.2 :Diagramme de cas d'utilisation globale

III.1.2.4. 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.1.3.Construction

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

III.1.3.1.réalisation de l'itération 1

III.1.3.1.1.Diagrammes de séquence pour itération 1:Les diagrammes de séquence présentent la coopération entre différents objets. Les objets sont définis et leur coopération est représentée par une séquence de messages entre eux.

Après avoir déterminé nos besoins et réaliser les diagrammes des cas d'utilisations nous allons élaborer quelques diagrammes de séquences de l'itération 1

❖ Ajouter compte utilisateur :

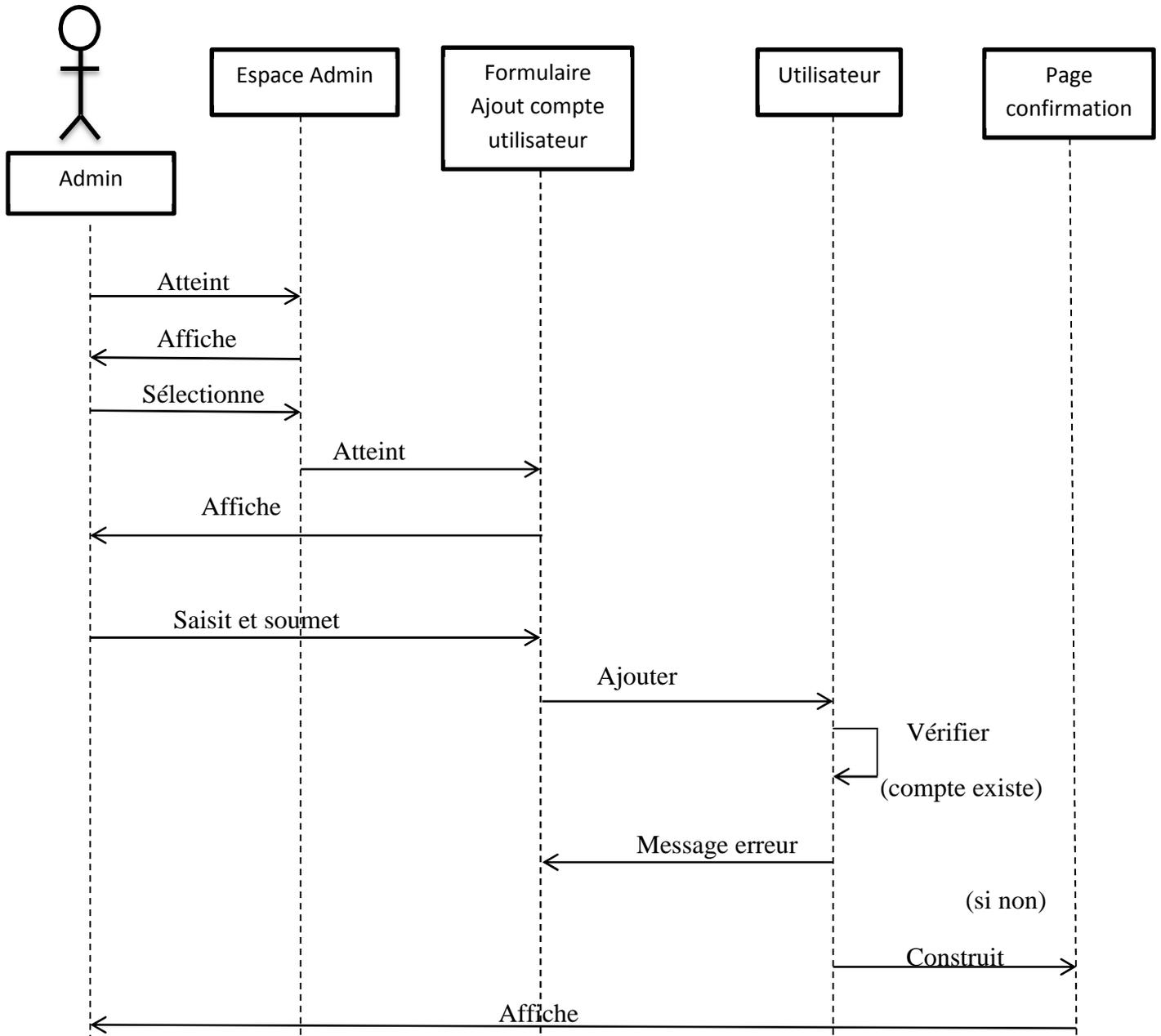


Figure III.3 : Diagramme de séquence du cas d'utilisation « Ajouter compte utilisateur »

Description du cas d'utilisation avec des scénarios:

Cas utilisation : Ajouter compte utilisateur.

Acteur : ADMIN.

Résumé : Ce cas d'utilisation permet à l'administrateur(ADMIN) d'ajouter un compte utilisateur et d'avoir un espace personnel.

Scénario :

- 1-L'ADMIN atteint son espace.
- 2-Le système affiche « espace ADMIN ».
- 3-L'ADMIN sélectionne la rubrique « compte utilisateur».
- 4-Le système retourne le formulaire « compte utilisateur ».
- 5-L'ADMIN saisit et soumet les données.
- 6-Le système retourne une page de confirmation

❖ Supprimer compte utilisateur :

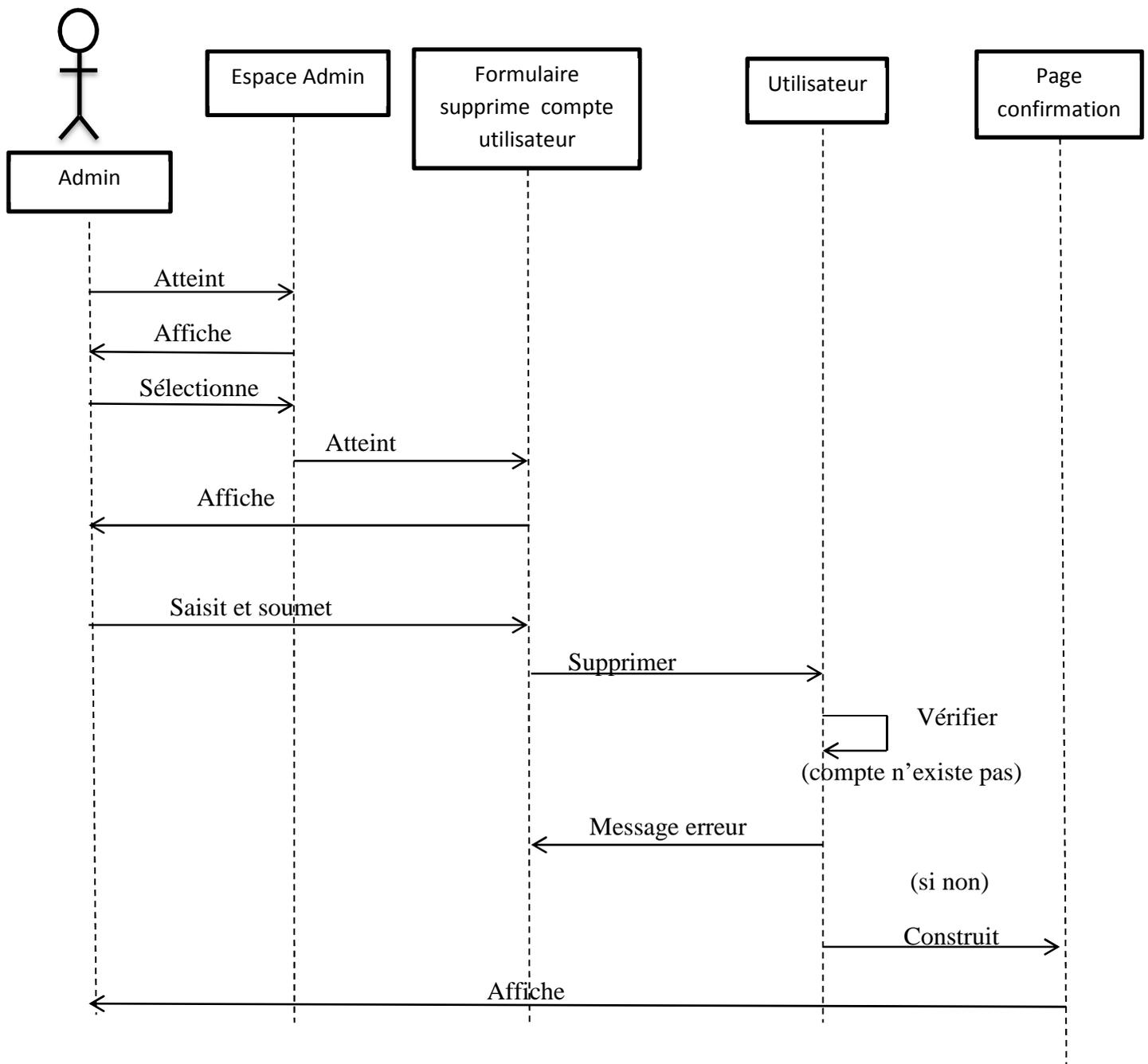


Figure.III.4 : Diagramme de séquence du cas d'utilisation «Supprimer compte utilisateur»

Description du cas d'utilisation avec des scénarios:

Cas utilisation : Supprime compte utilisateur.

Acteur : ADMIN.

Résumé : Ce cas d'utilisation permet à l'administrateur(ADMIN) supprimer un compte utilisateur.

Scénario :

- 1-L'ADMIN atteint son espace.
- 2-Le système affiche « espace ADMIN ».
- 3-L'ADMIN sélectionne la rubrique « supprimer compte utilisateur».
- 4-Le système retourne le formulaire «supprimer compte utilisateur ».
- 5-L'ADMIN saisit et soumet les données.
- 6-Le système retourne une page de confirmation

❖ Modifier compte utilisateur :

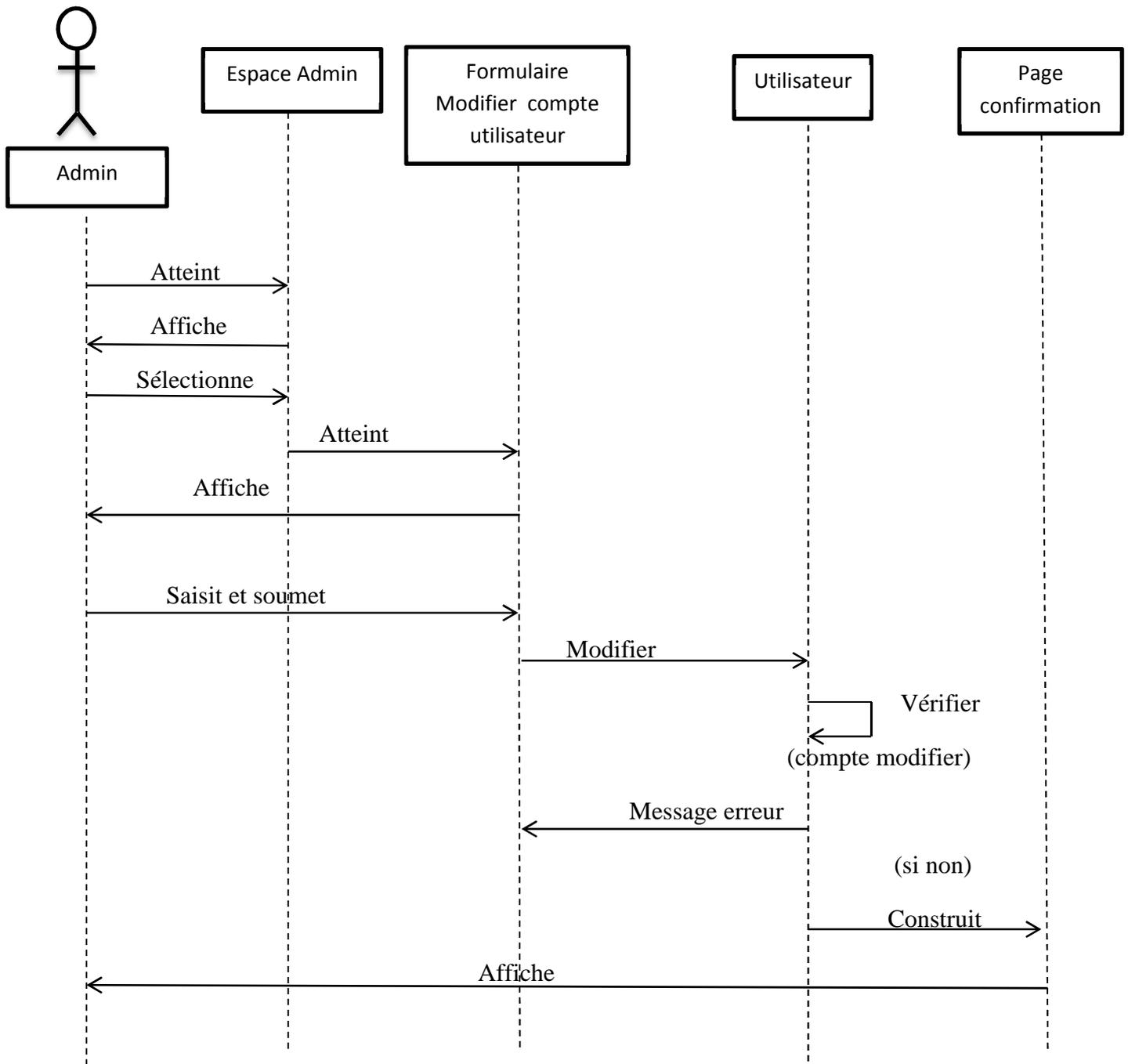


Figure.III.5 : Diagramme de séquence du cas d'utilisation «**Modifier compte utilisateur**»

✚ description du cas d'utilisation avec des scénarios:

Cas utilisation : Supprime compte utilisateur.

Acteur : ADMIN.

Résumé : Ce cas d'utilisation permet à l'administrateur(ADMIN) modifie un compte utilisateur.

Scénario :

- 1-L'ADMIN atteint son espace.
- 2-Le système affiche « espace ADMIN ».
- 3-L'ADMIN sélectionne la rubrique « Modifier compte utilisateur».
- 4-Le système retourne le formulaire « Modifier compte utilisateur ».
- 5-L'ADMIN saisit et soumet les données.
- 6-Le système retourne une page de confirmation

III.1.3.1.2.Diagrammes de classe pour l'itération 1

Utilisateur
Ident_util Mot_passe Service
Ajouter() Supprimer() Consulter() Modifier()

Figure.III.6. Le diagramme de classe pour l'itération 1

III.1.3.2.réalisation de l'itération 2

III.1.3.2.1.Diagrammes de séquence pour itération 2: quelques diagrammes de séquences de l'itération 2

❖ Ajouter un fournisseur homologué :

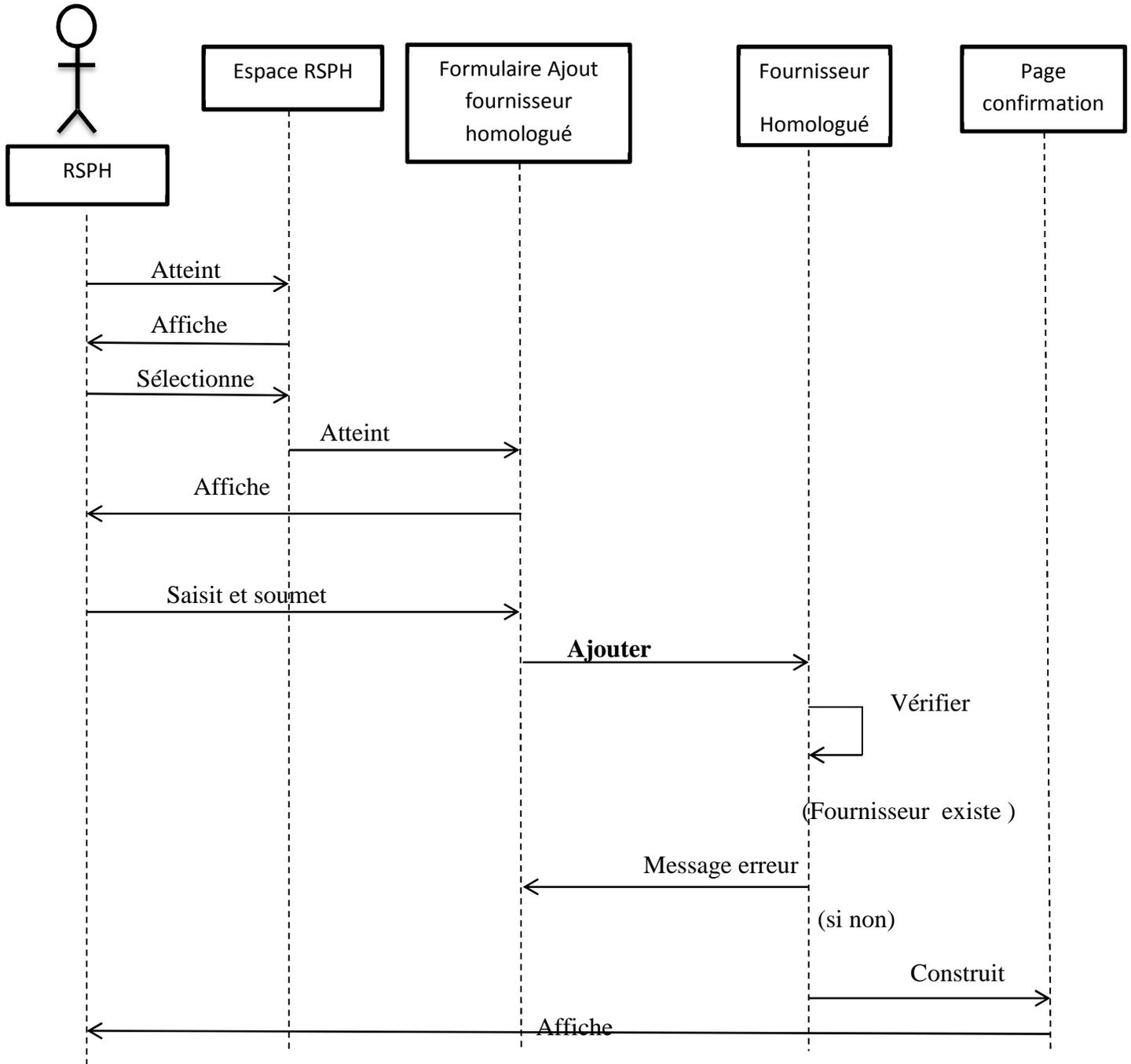


Figure.III.7: Diagramme de séquence du cas d'utilisation «Ajouter un fournisseur homologué»

Description du cas d'utilisation avec des scénarios:

Cas utilisation : Ajouter fournisseur homologué.

Acteur : RSPH.

Résumé : Ce cas d'utilisation permet au responsable service prospection homologation (RSPH) d'ajouter un fournisseur homologué et d'avoir un espace personnel.

Scénario :

- 1- Le RSPH atteint son espace.
- 2- Le système affiche « espace RSPH ».
- 3- L'utilisateur sélectionne la rubrique « Fournisseur ».
- 4- Le système retourne le formulaire « Fournisseur ».
- 5- Le RSPH remplit le formulaire et clique sur le bouton « valider ».
- 6- Le système ajoute un nouveau fournisseur homologué après avoir vérifié la validité des données saisies.

❖ Consulter demande d'offre :

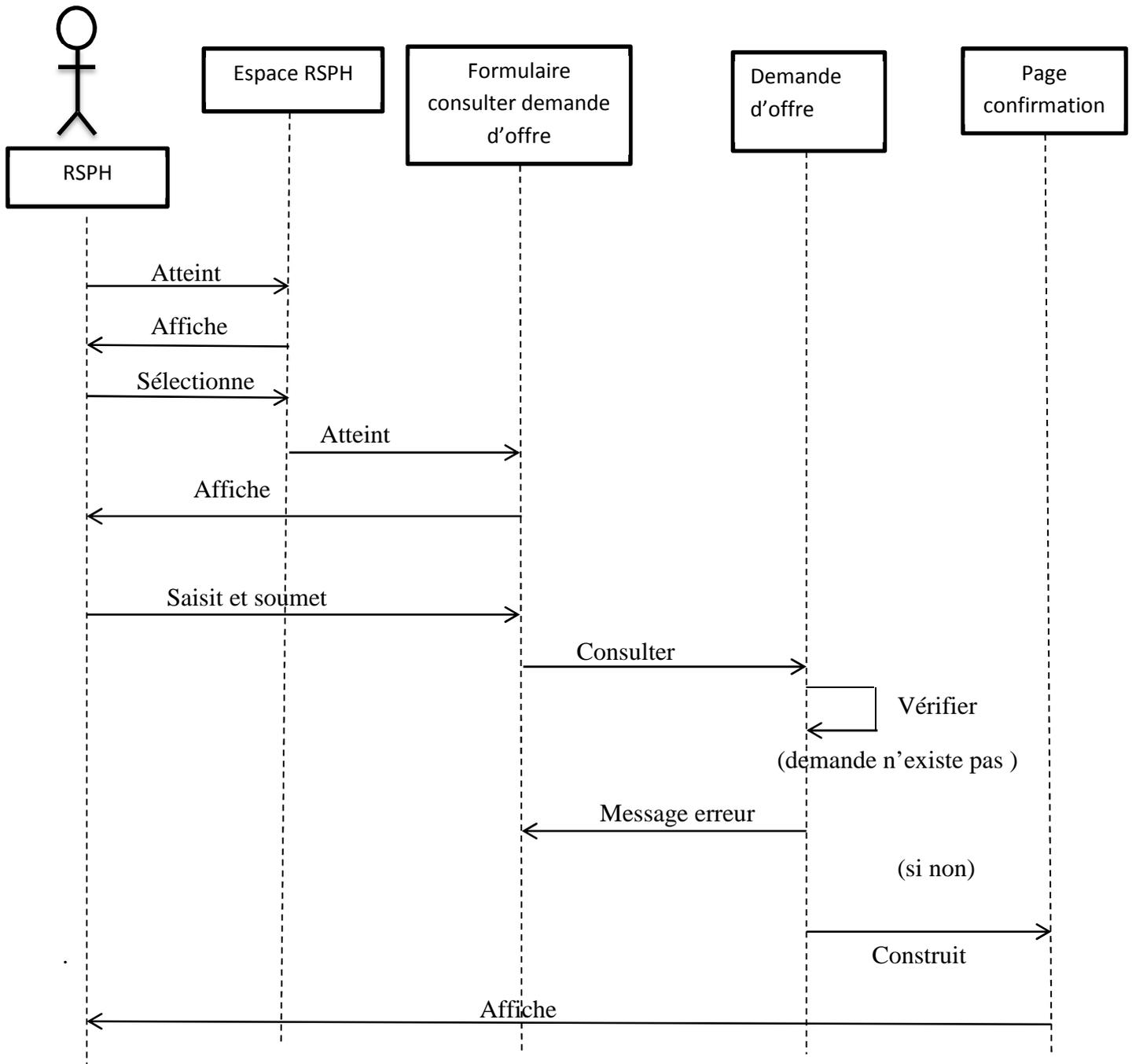


Figure.III.8: Diagramme de séquence du cas d'utilisation «Consulter demande d'offre»

Description du cas d'utilisation avec des scénarios:

Cas utilisation : Consulter demande d'offre.

Acteur : RSPH.

Résumé : Ce cas d'utilisation permet au responsable service prospection homologation (RSPH) de consulter une demande d'offre.

Scénario :

- 1- Le RSPH atteint son espace.
- 2- Le système affiche « espace RSPH ».
- 3- L'utilisateur sélectionne la rubrique « demande d'offre ».
- 4- Le système retourne le formulaire « demande d'offre ».
- 5- Le RSPH remplit le formulaire et clique sur le bouton « valider ».
- 6- Le système retourne une page de confirmation.

❖ Ajouter attestation de non transfert de fonds :

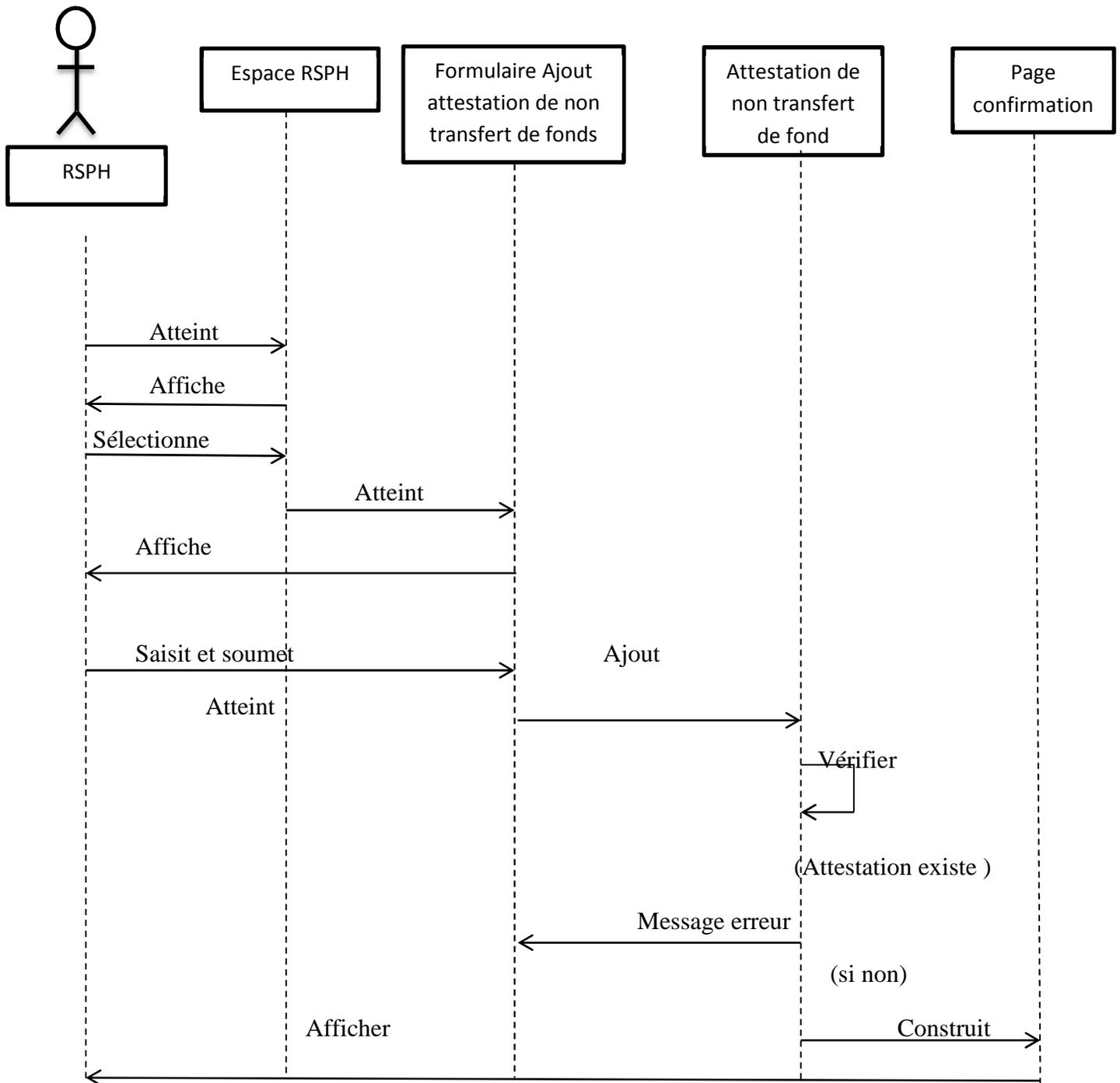


Figure.III.9: Diagramme de séquence du cas d'utilisation «Ajouter attestation de non transfert de fonds »

Description du cas d'utilisation avec des scénarios:

Cas utilisation : Ajouter attestation de non transfert de fonds

Acteur : RSPH.

Résumé : Ce cas d'utilisation permet au responsable service prospection homologation (RSPH) d'ajouter attestation de non transfert de fonds

Scénario :

- 1- Le RSPH atteint son espace.
- 2- Le système affiche « espace RSPH ».
- 3- L'utilisateur sélectionne la rubrique «Ajouter attestation de non transfert de fonds ».
- 4- Le système retourne le formulaire «Ajouter attestation de non transfert de fonds ».
- 5- Le RSPH remplit le formulaire et clique sur le bouton « valider ».
- 6- Le système ajoute un nouveau fournisseur homologué après avoir vérifié la validité des données saisies.

III.1.3.2.2. Diagrammes de classe pour l'itération 2

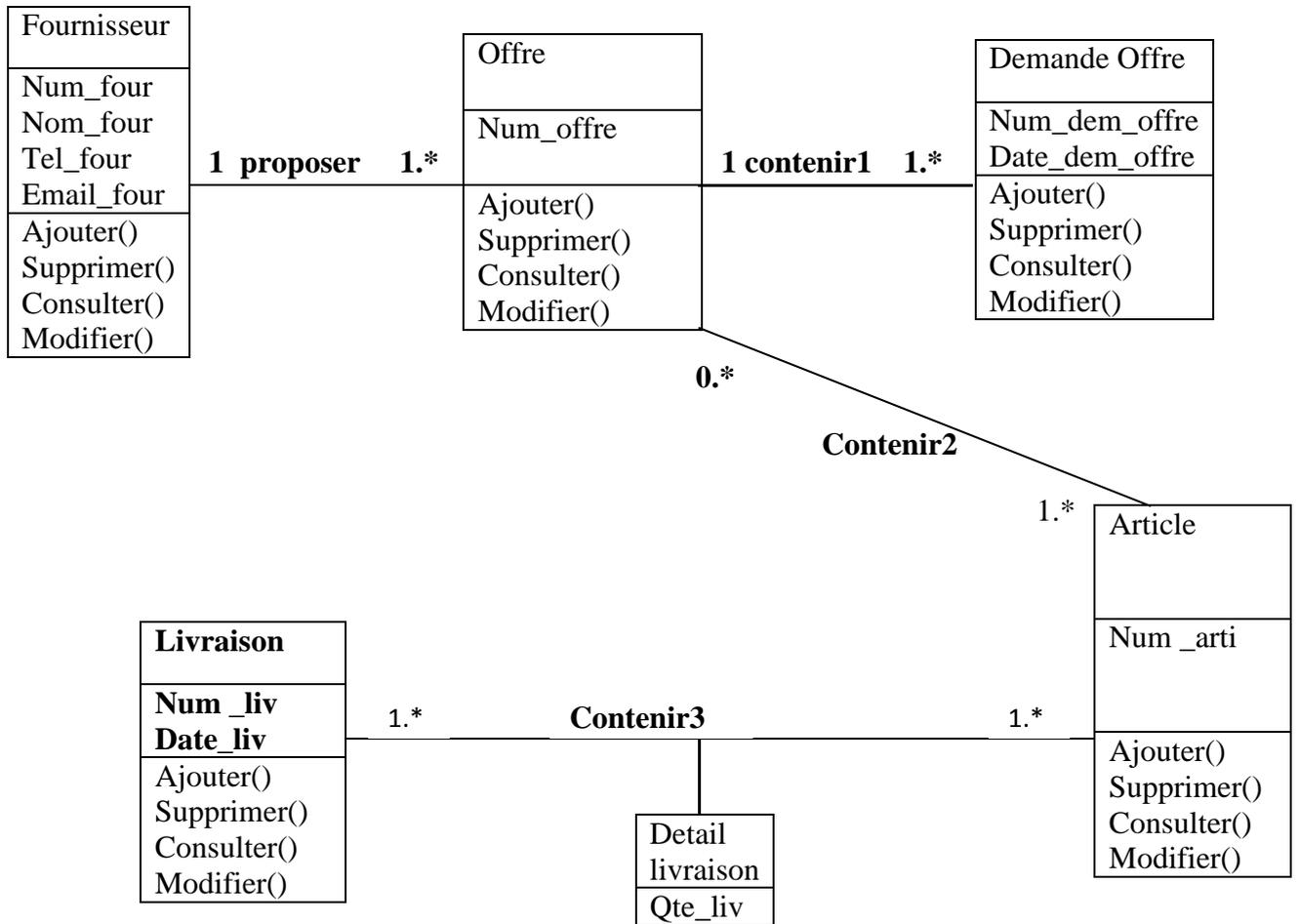


Figure.III.10. Diagramme de classes pour l'itération 2

III.1.3.3.réalisation de l'itération 3

III.1.3.3.1. Diagrammes de séquence pour itération 3: quelques diagrammes de séquences de l'itération 3

❖ Ajouter une facteur:

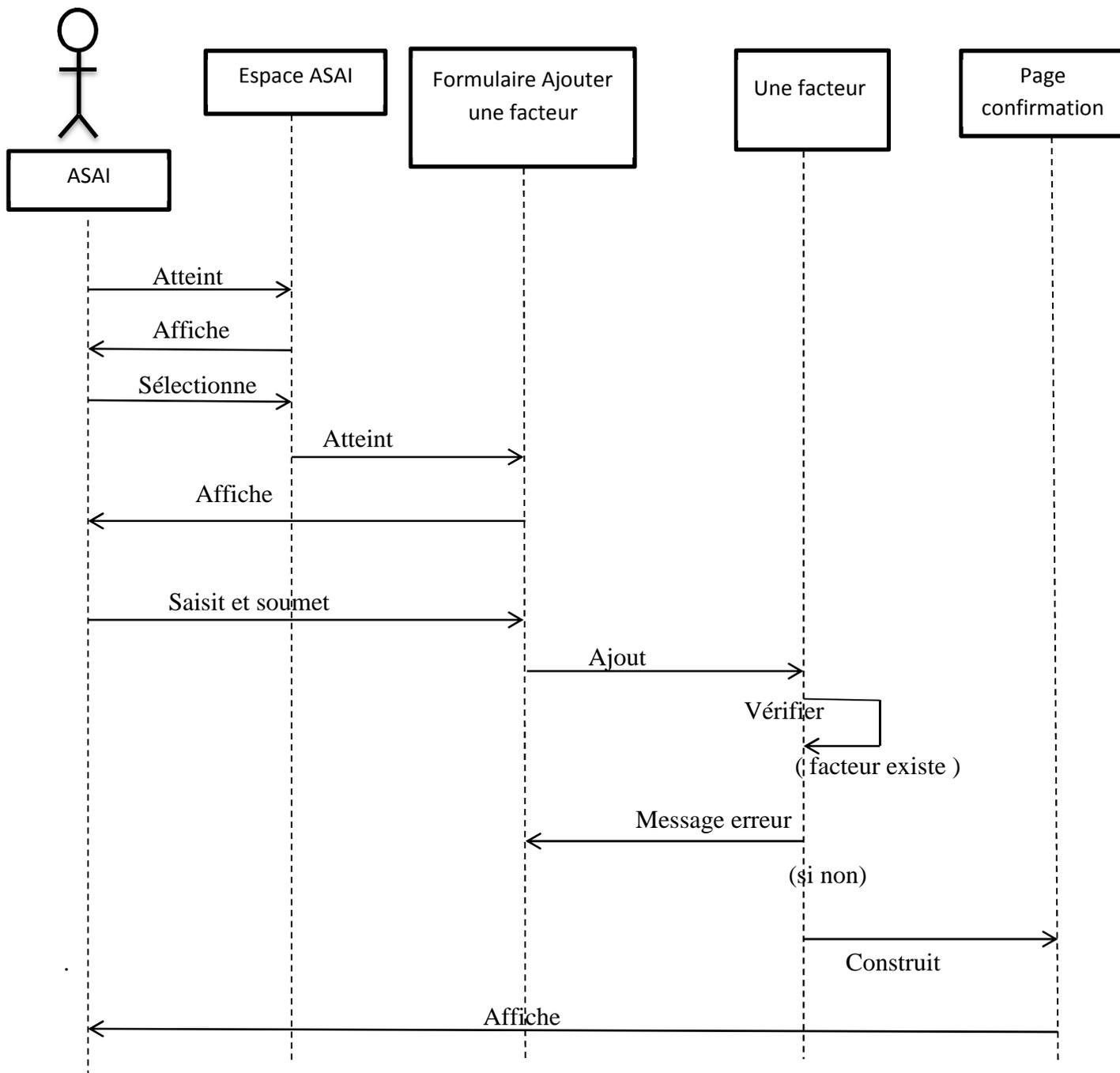


Figure III .11 Diagramme de séquence du cas d'utilisation «Ajouter une facteur»

Description du cas d'utilisation avec des scénarios:

Cas utilisation : Ajouter une facteur.

Acteur : ASAI.

Résumé : Ce cas d'utilisation permet au agent service achat importation d'ajouter une facteur.

Scénario :

- 1- Le ASAI atteint son espace.
- 2- Le système affiche « espace ASAI».
- 3- L'utilisateur sélectionne la rubrique «Ajouter une facteur ».
- 4- Le système retourne le formulaire « Ajouter une facteur ».
- 5- Le ASAI remplit le formulaire et clique sur le bouton « valider ».
- 6- Le système retourne une page de confirmation.

❖ supprimer une commande:

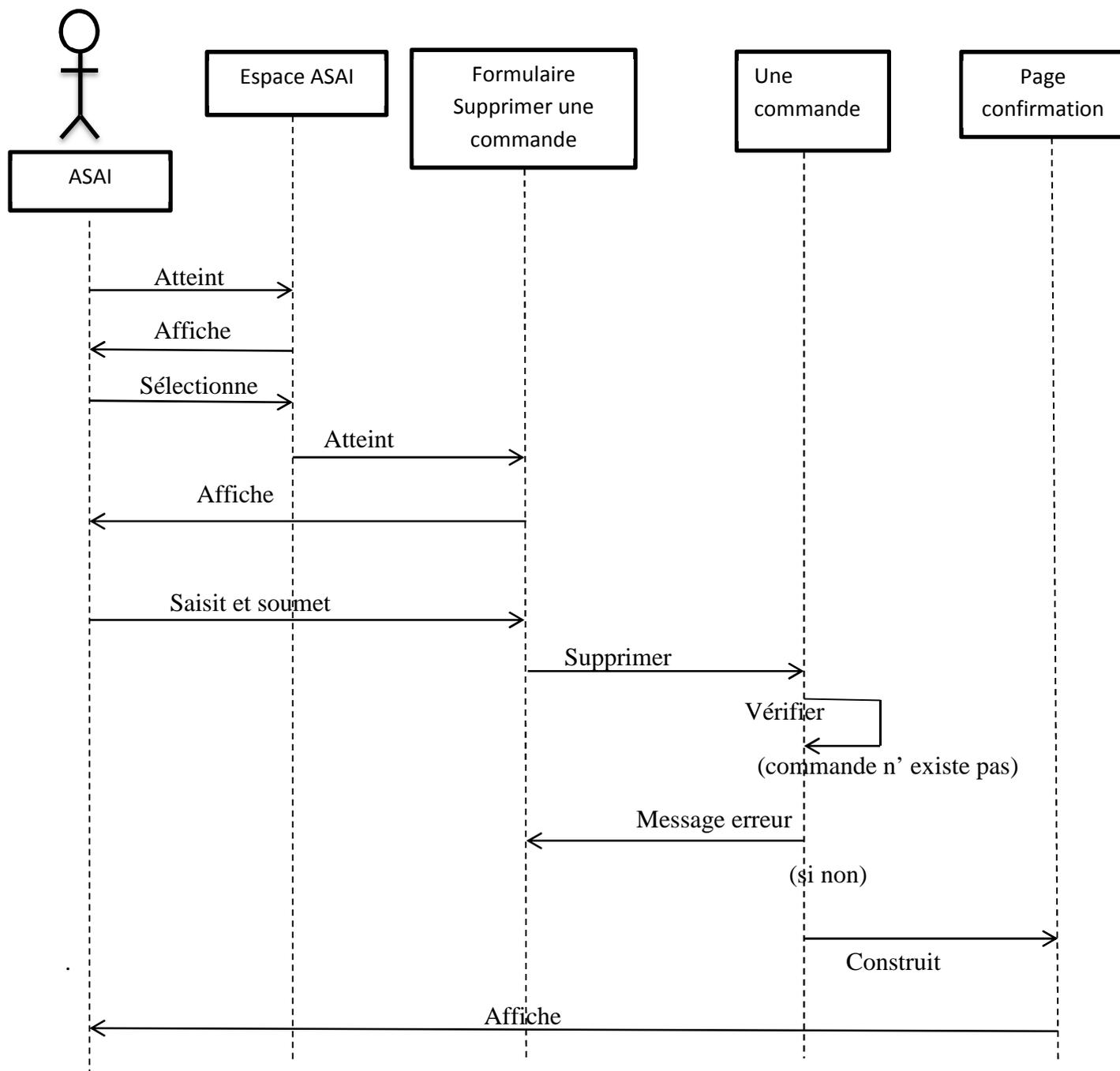


Figure III .12: Diagramme de séquence du cas d'utilisation «Supprimer une commande»

Description du cas d'utilisation avec des scénarios:

Cas utilisation : Supprimer une commande

Acteur : ASAI.

Résumé : Ce cas d'utilisation permet au agent service achat importation Supprimer une commande

Scénario :

- 1- Le ASAI atteint son espace.
- 2- Le système affiche « espace ASAI».
- 3- L'utilisateur sélectionne la rubrique «Supprimer une commande».
- 4- Le système retourne le formulaire «Supprimer une commande».
- 5- Le ASAI remplit le formulaire et clique sur le bouton « valider ».
- 6- Le système retourne une page de confirmation.

III.1.3.3.2. Diagrammes de classe pour l'itération 3

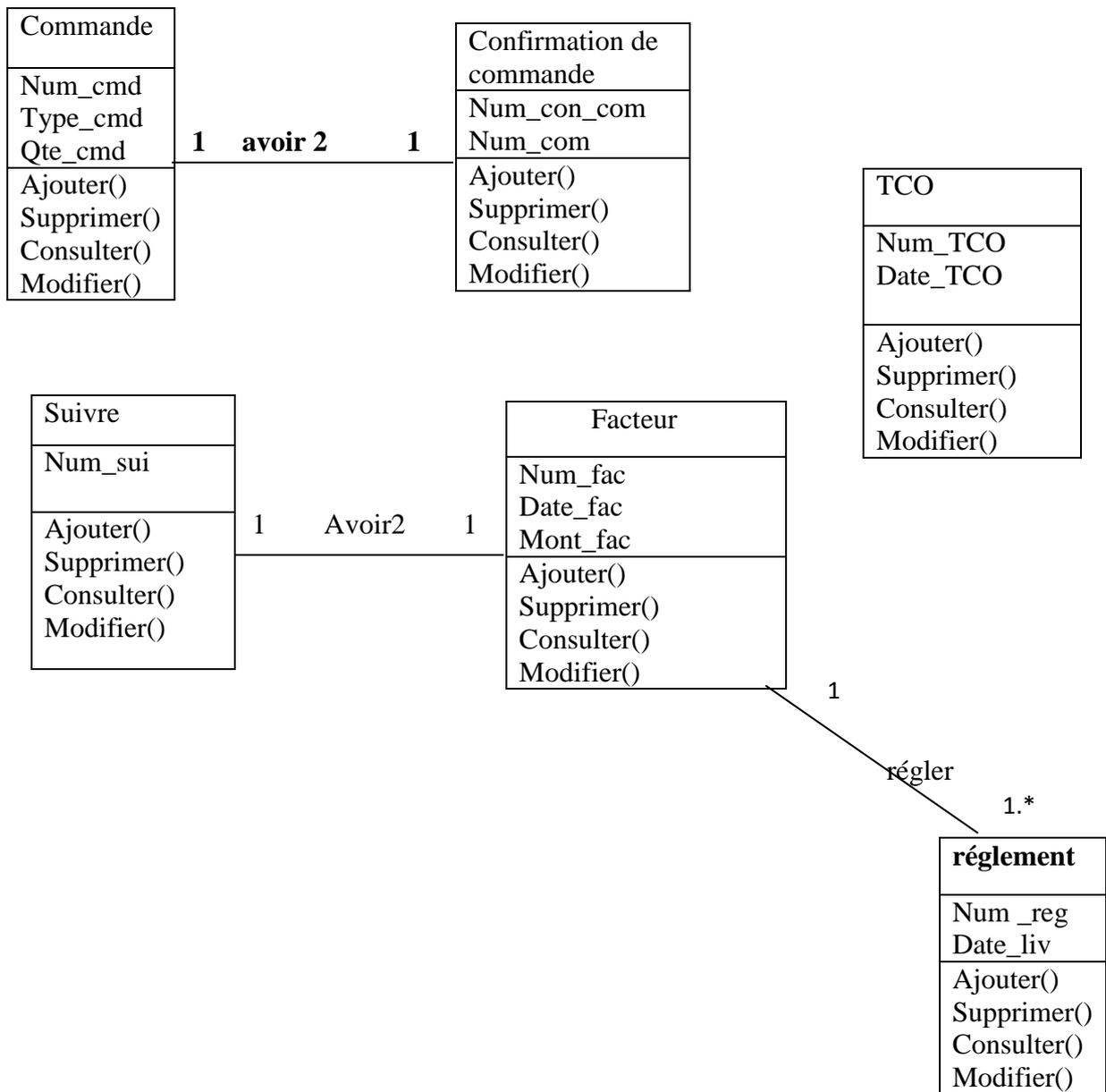


Figure.III.13. Diagramme de classes pour l'itération 3

Remarque :

- La classe **commande** relié avec la classe **article** de l'iteration2 par les cardinalité(1*.....1*).
- La classe **commande** relié avec la classe **livraison** de l'iteration2 par les cardinalité(1.....1).
- La classe **tco** relié avec la classe **offre** de l'iteration2 par les cardinalité(1.....1*).
- La classe **suiivre** relié avec la classe **livraison** de l'iteration2 par les cardinalité(1.....1).

III.1.3.4.réalisation de l'itération 4

III.1.3.4.1. Diagrammes de séquence pour itération 4: quelques diagrammes de séquences de l'itération 4

❖ Ajouter demande de chèque :

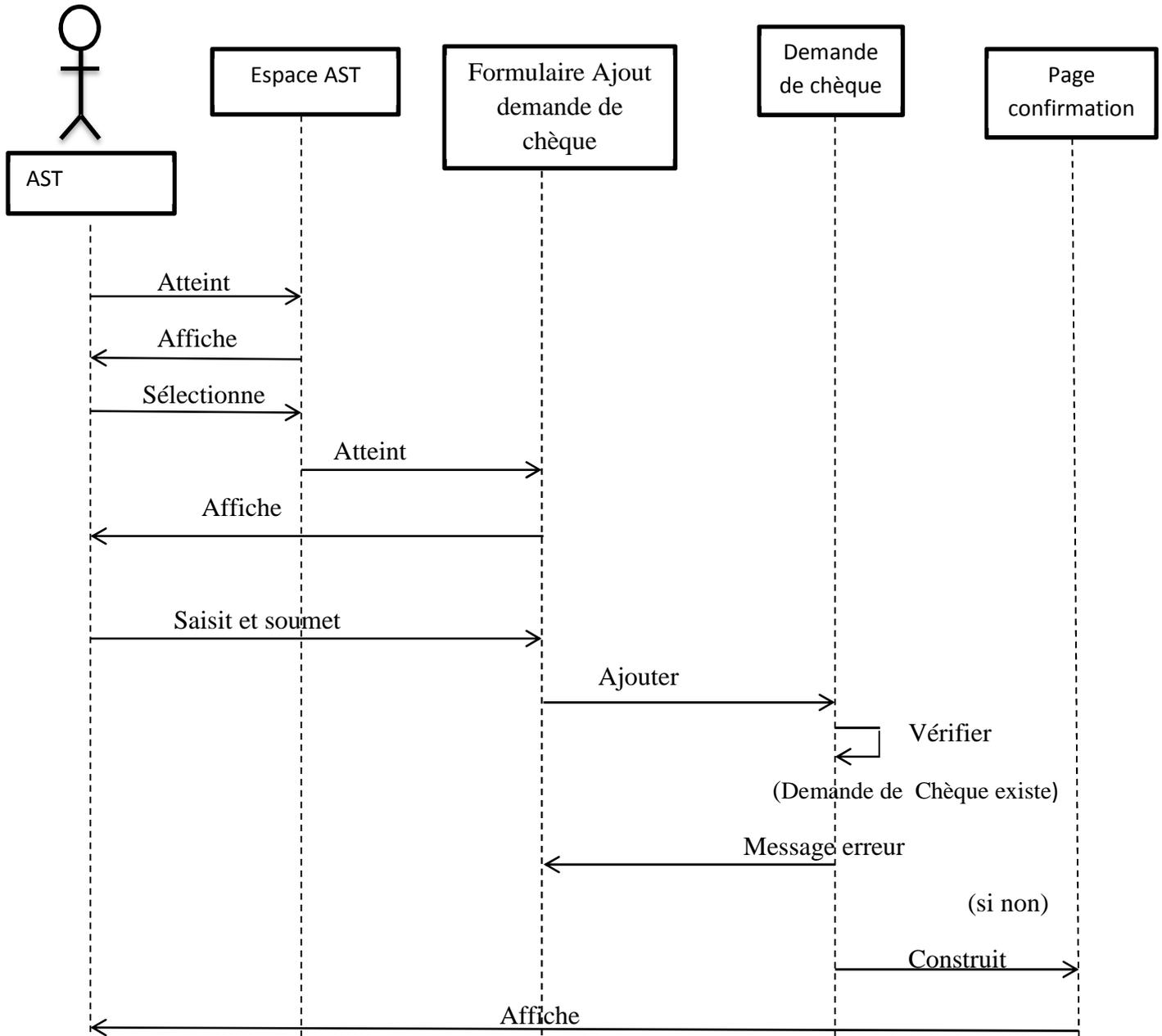


Figure III.14 : Diagramme de séquence du cas d'utilisation «Ajouter une demande de chèque»

Description du cas d'utilisation avec des scénarios:

Cas utilisation : Ajouter une demande de chèque .

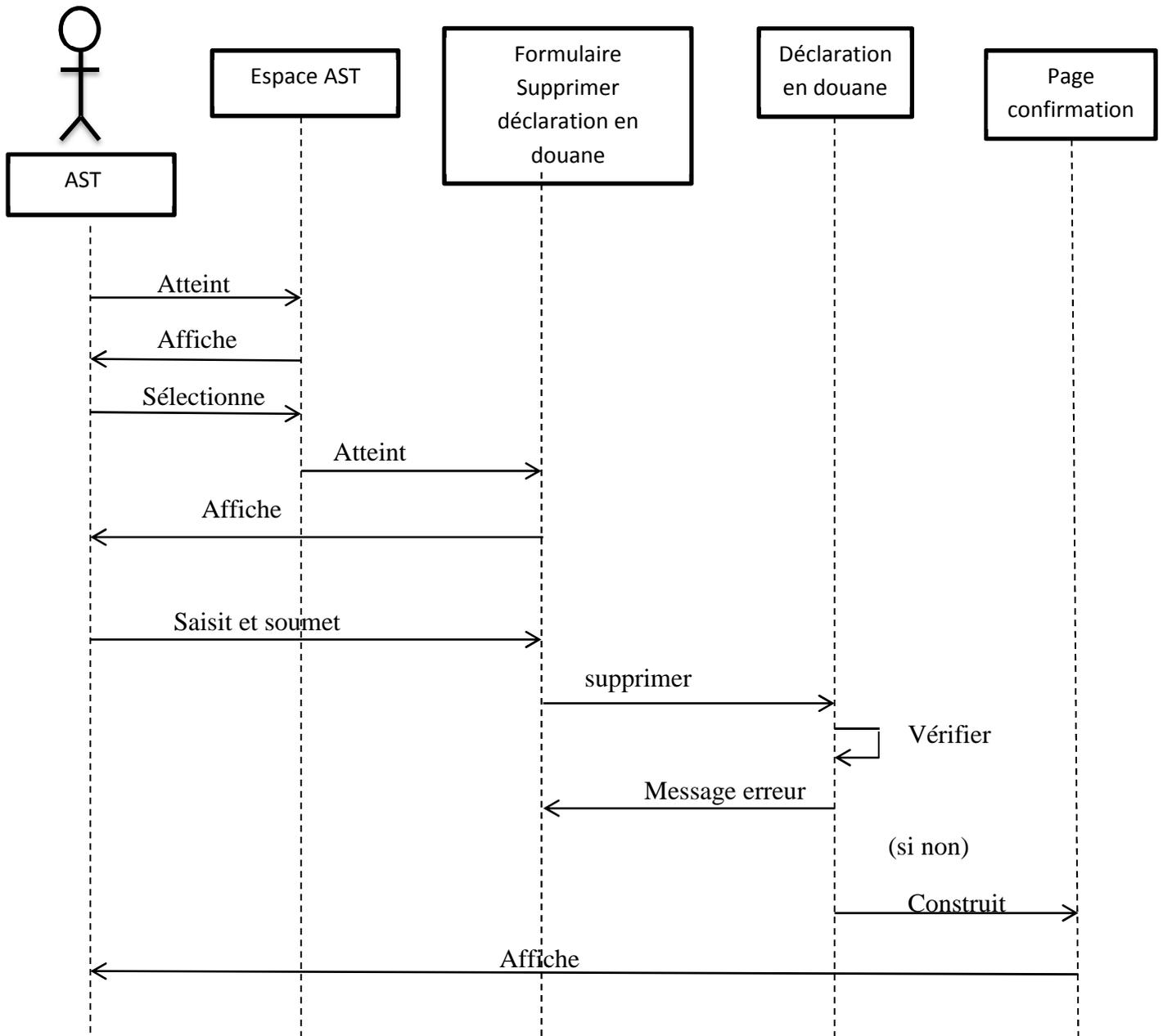
Acteur : AST

Résumé : Ce cas d'utilisation permet au agent service transit d'ajouter demande de chèque

Scénario :

- 1- Le AST atteint son espace.
- 2- Le système affiche « espace AST».
- 3- L'utilisateur sélectionne la rubrique «Ajouter demande de chèque ».
- 4- Le système retourne le formulaire « Ajouter demande de chèque ».
- 5- Le AST remplit le formulaire et clique sur le bouton « valider ».
- 6- Le système retourne une page de confirmation.

❖ Supprimer déclaration en douane :



FigureIII.15: Diagramme de séquence du cas d'utilisation «Supprimer déclaration en douane »

Description du cas d'utilisation avec des scénarios:

Cas utilisation : Supprimer déclaration en douane .

Acteur : AST

Résumé : Ce cas d'utilisation permet au agent service transit Supprimer déclaration en douane

Scénario :

- 1- Le AST atteint son espace.
- 2- Le système affiche « espace AST».
- 3- L'utilisateur sélectionne la rubrique «Supprimer déclaration en douane ».
- 4- Le système retourne le formulaire «Supprimer déclaration en douane ».
- 5- Le AST remplit le formulaire et clique sur le bouton « valider ».
- 6- Le système retourne une page de confirmation.

III.1.3.4.2 : Diagrammes de classe pour l'itération 4

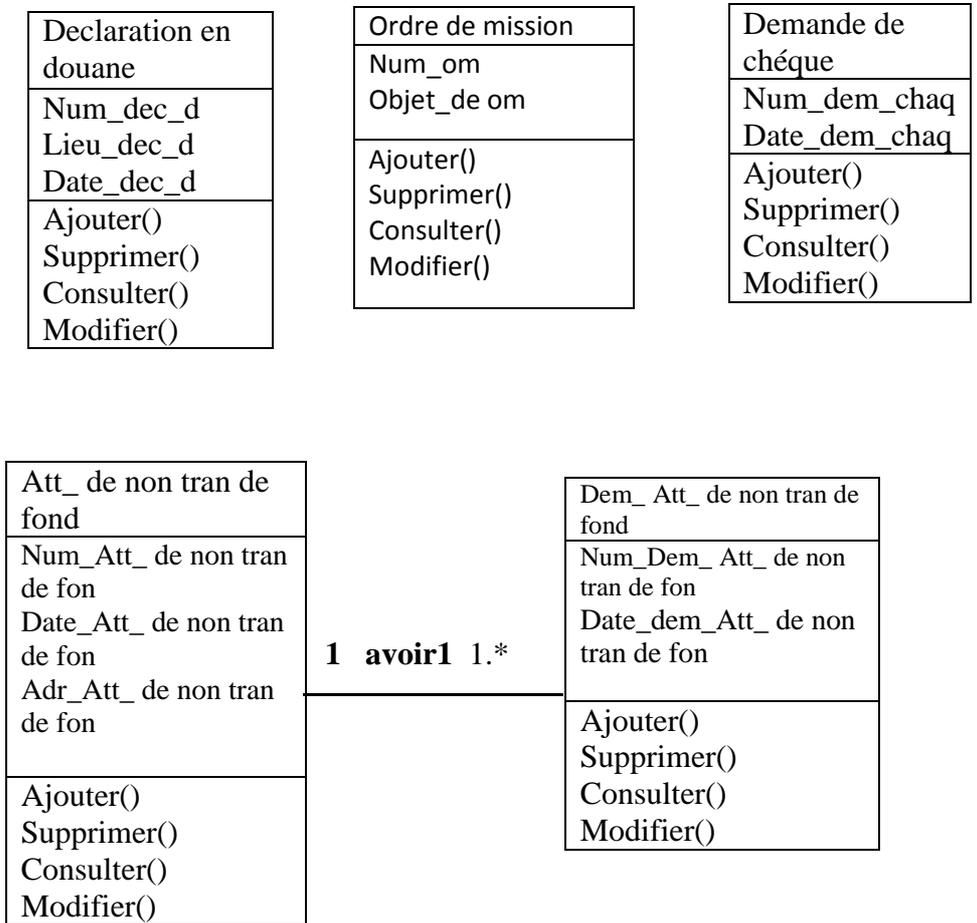


Figure.III.16. Diagramme de classes pour l'itération 4

Remarque :

-La classe **attestations de non transfert de fonds** relié avec la classe **Livraison** de l'iteration3 par les cardinalité(1.....1).

-La classe **attestations de non transfert de fonds** relié avec la classe **Facteur** de l'iteration3 par les cardinalité(1.....1).

La classe **Déclaration en douane** relié avec la classe **Livraison** de l'iteration2 par les cardinalité(1.....1).

La classe **Ordre de mission** relié avec la classe **Livraison** de l'iteration3 par les cardinalité(1*.....1).

La classe **Demande de chaque** relié avec la classe **Facteur** de l'iteration3 par les cardinalité(1.....1).

Conclusion :

Dans ce chapitre nous avons utilisé le processus unifié RUP pour l'analyse et la conception de notre application en utilisant le langage de modélisation UML. Dans le chapitre suivant, qui touche à la réalisation, nous essayons de construire notre application par itération.

Réalisation

IV.1.Introduction :

Après avoir présenté dans le chapitre précédent les différentes étapes d'analyse et de conception, nous allons présenter dans ce dernier chapitre l'environnement de développement, les outils qui ont servi à la réalisation de notre application, pour illustrer quelques fonctionnalités de l'application.

IV.2. Outils de développement :

Pour la mise en place de notre application, nous avons opté pour l'environnement logiciel suivant:

IV.2.1.Configuration logiciel:

Système d'exploitation : le projet est effectué sous une plate forme Windows7

System de gestion de base de données : oracle 10g

Langage de programmation :Net Beans IDE 7.0.1

IV.2.2 Configuration matérielle:

- Un micro processeur Intel pentium inside.
- Fréquence d'horloge 2 GHZ.
- RAM 2GO.
- Disque dure de 120 GO.
- Lecteur DVD.
- Ecran SGA couleur 17'' pouces.
- Clavier azerty 108 touche.

IV.2.3 Pour quoi oracle 10g :

Oracle 10g Il s'agit d'un générateur d'applications transactionnelles basé sur le langage PL/SQL, il permet de construire facilement et rapidement des formulaires complexes interagissent avec les bases de données. la version 10g supporte les expressions rationnelles. Le g signifie grille informatique (en anglais, *grid*) est une infrastructure virtuelle constituée d'un ensemble de ressources informatiques potentiellement partagées, distribuées, hétérogènes, délocalisées et autonomes. Une grille se compose de ressources informatiques : tout élément qui permet l'exécution d'une tâche ou le stockage d'une donnée numérique.

IV.2.3.1 Qu'est-ce que Oracle Database 10g Express Edition?

Oracle Database 10g Express Edition (Oracle Database XE) est une version téléchargeable gratuitement de la plupart du monde base de données relationnelle capable. Oracle Database XE est facile à installer et facile à gérer.

Avec Oracle Database XE, vous utilisez la page d'accueil de la base de données, une interface basée sur un navigateur intuitif, pour administrer le base de données; créer des tables, des vues et autres objets de schéma; importation, l'exportation, et vue les données du tableau; exécuter des requêtes SQL et

son exécution; et générer des rapports.

Oracle Database XE inclut Oracle HTML DB 2.1, un environnement de développement déclaratif, graphique pour la création

applications Web aux bases de données centrée. En plus de HTML DB 2.1, vous pouvez utiliser Oracle populaire et tiers

langues et des outils pour développer vos applications Oracle Database XE.

IV.2.4.Le langage de programmation :

NetBeans est un environnement de développement intégré (EDI), placé en *open source* par Sun en juin 2000 sous licence CDDL et GPLv2 (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages 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). Un environnement Java Development Kit JDK est requis pour les développements en Java.

NetBeans constitue par ailleurs une plate forme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plateforme. Netbeans est un IDE qui supporte une large variété de langages de programmation et d'outils de collaboration.

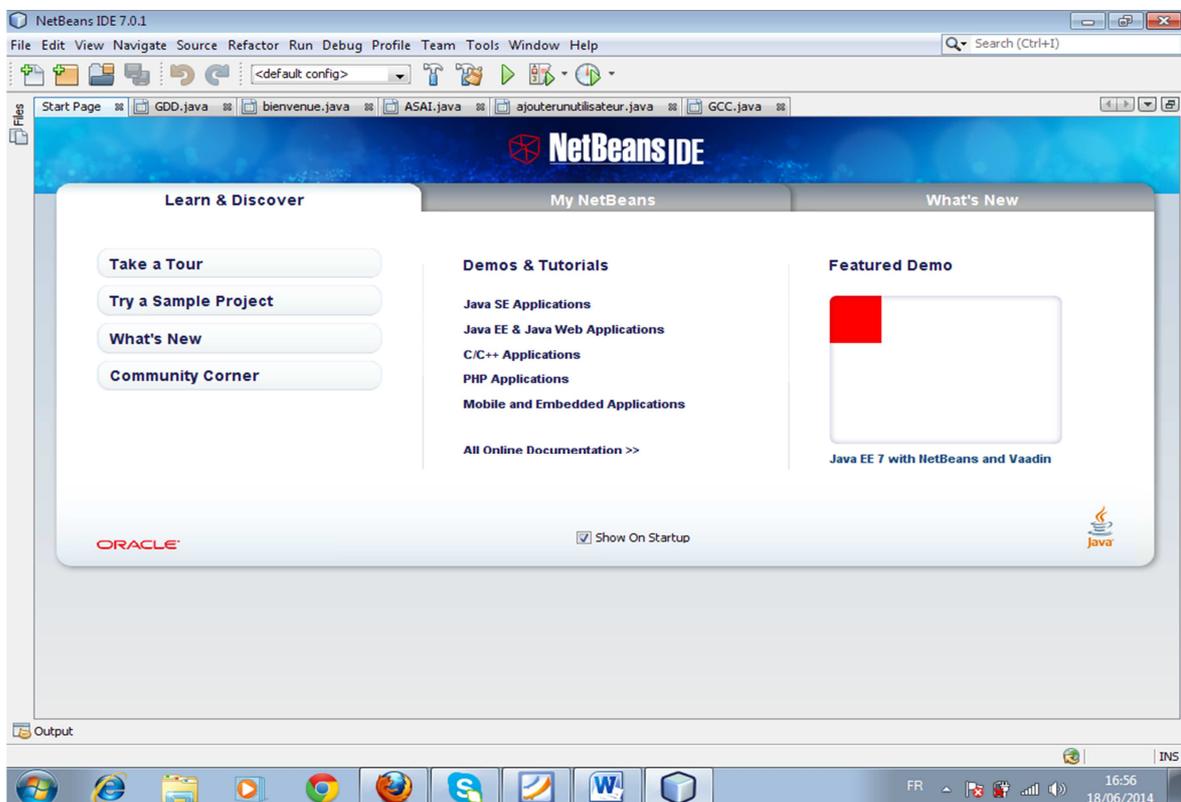
IV.2.4.1 Définition IDE : L'expression *environnement de développement* est apparue dans les années 1980, pour désigner un ensemble d'outils et de procédés destinés à assister et formaliser le travail de création de logiciels. L'objectif visé était de créer des outils qui assistent les développeurs dans toutes les étapes de la réalisation du logiciel: définition,

conception, programmation, test et maintenance. Ces outils tiennent également compte des différents rôles au sein d'une équipe de programmation: programmeur, manager, et responsable qualité

Les premiers outils couvraient les étapes de programmation et de tests, tandis que sont apparus plus tard des outils couvrant également les étapes de définition et de conception. Cette nouvelle génération d'outils de développement, qui couvrent également les étapes préliminaires à la programmation et orientent le développeur vers les bonnes pratiques de génie logiciel a été appelée *Computer Aided Software Engineering* (abr. *CASE*), en français ingénierie logicielle assistée par ordinateur

Un environnement de développement comporte typiquement les outils nécessaires pour analyser, écrire, et déboguer un programme. Dans les modèles *intégrés*, les différents outils ont un *look and feel* identique et peuvent être interconnectés. Par exemple il peut être possible d'ouvrir l'éditeur de texte ou le compilateur à partir du débogueur voire de modifier le code source en cours de débogage directement depuis le débogueur

- NetBeans 7.0.1



FigureIV.1: NetBeans 7.0.1

IV.3.les Itérations :

IV.3.1.Itération I :

✚ Son modèle relationnel :

Utilisateur (Ident_util, Mot_de_passe,service)

✚ Ses tables :

- Table Utilisateur

Column Name	Data Type	Nullable	Default	Primary Key
ID_UTILIS	VARCHAR2(4)	Yes	-	-
MOT_DE_PASSE	NUMBER	Yes	-	-
SERVICE	VARCHAR2(50)	Yes	-	-
				1 - 3

IV.3.2.Itération II :

✚ Son modèle relationnel

Fournisseur:(Nun_four,nom_four,tel_four,fax_four,email_four,adr_four)

Offre: (Num_offre,num_four*,num_tco*,design_off,prix)

Demande d'offre: (Num_dem_offre,date_dem_offre,num_offre*,date_dem_offre,nom_de_fournisseur)

Attestation de non transfert de fon:(Num_att_non_tran_fon,adr_banque,num_livra*,date_att_non_tran_fon)

Demande attestation de non transfert:(Num_dem_att_non_trensf_fond,date_dem_att_non_trensf_fond, Num_att_non_tran_fon*)

livraison:(Num_liv,num_cmd*,date_liv)

Detail livraison:(Qte_liv, Num_non_trensf_fond*,num_liv*)

Article:(Num_art,desig_art,qte)

✚ Ses tables :Quelque tables pour l'itération2:

- Table Fournisseur:

Column Name	Data Type	Nullable	Default	Primary Key
NUM_FOR	VARCHAR2(40)	No	-	1
NOM	VARCHAR2(40)	No	-	-
TEL	NUMBER	No	-	-
FAX	NUMBER	No	-	-
E_MAIL	VARCHAR2(40)	No	-	-
ADR	VARCHAR2(80)	No	-	-
				1 - 6

- Table Offre :

Column Name	Data Type	Nullable	Default	Primary Key
NUM_OFF	VARCHAR2(4)	No	-	1
NUM_FOR	NUMBER	No	-	-
NUM_TCO	NUMBER	No	-	-
DESIG_OFF	VARCHAR2(40)	No	-	-
PRIX	NUMBER	No	-	-
				1 - 5

- Table demande d'offre:

Column Name	Data Type	Nullable	Default	Primary Key
NUM_DEM_OFF	VARCHAR2(4)	No	-	1
NUM_OFF	NUMBER	No	-	-
DATE_DEM_OFF	DATE	No	-	-
NOM_DE_FOURNISSEUR	VARCHAR2(40)	No	-	-
FAX	NUMBER	No	-	-
				1 - 5

- Table article :

Column Name	Data Type	Nullable	Default	Primary Key
NUM_LIV	NUMBER	No	-	1
NUM_CMD	NUMBER	No	-	-
DATE_LIV	DATE	No	-	-
				1 - 3

IV.3.3.itération III:

✚ Son modèle relationnel :

Commande:(Num_comd, designation_comd,pu,montant,qte)

Configuration de commande:(Num_conf_comd,num_comd*,)

Suivre:(Num_suiv,num_liv*)

Facture:(Num_fac, num_suiv*,num_ates_non_tra_fond*, date_fac, prix)

Règlement: (Num_rég,Num_fac*,date_rég)

Tco:(Num_tco,design_offre,nom_de_four,prix)

✚ Ses tables : Quelques tables pour l'itération 3 :

- **Table Commande**

Column Name	Data Type	Nullable	Default	Primary Key
NUM_FAC	VARCHAR2(4)	Yes	-	-
NUM_SUI	NUMBER	Yes	-	-
NUM_ATT_NON_TRA_FON	NUMBER	Yes	-	-
PRIX	NUMBER	Yes	-	-
DATE_FAC	DATE	Yes	-	-
				1 - 5

- **Table règlement**

Column Name	Data Type	Nullable	Default	Primary Key
NUM_REG	VARCHAR2(4)	Yes	-	-
NUM_FAC	NUMBER	Yes	-	-
DATE_REG	DATE	Yes	-	-
				1 - 3

- **Table Suivre :**

Column Name	Data Type	Nullable	Default	Primary Key
NUM_SUI	NUMBER	No	-	1
NUM_LV	NUMBER	No	-	-
				1 - 2

- **Table Facture :**

Column Name	Data Type	Nullable	Default	Primary Key
NUM_FAC	VARCHAR2(4)	Yes	-	-
NUM_SUI	NUMBER	Yes	-	-
NUM_ATT_NON_TRA_FON	NUMBER	Yes	-	-
PRIX	NUMBER	Yes	-	-
DATE_FAC	DATE	Yes	-	-
				1 - 5

IV.3.4.itération IV :

✚ Son modèle relationnel :

Demande de chèque:(Num_dem_chaq,Montant_en chiffre,Montant_en lettres,Num_fac*)

Ordre de mission :(Num_om,moyen_de_transp,date_depart,num_liv*).

déclaration en douane:(num_dd,date_dd,num_liv*)

✚ Ses tables : Quelques tables pour l'itération4

- Demande de chèque :

Column Name	Data Type	Nullable	Default	Primary Key
NUM_DEN_CHAQ	VARCHAR2(4)	No	-	1
NUM_FAC	NUMBER	No	-	-
MONTANT_EN_CHIFFRES	NUMBER	No	-	-
MONTANT_EN_LETTRES	VARCHAR2(60)	No	-	-
				1 - 4

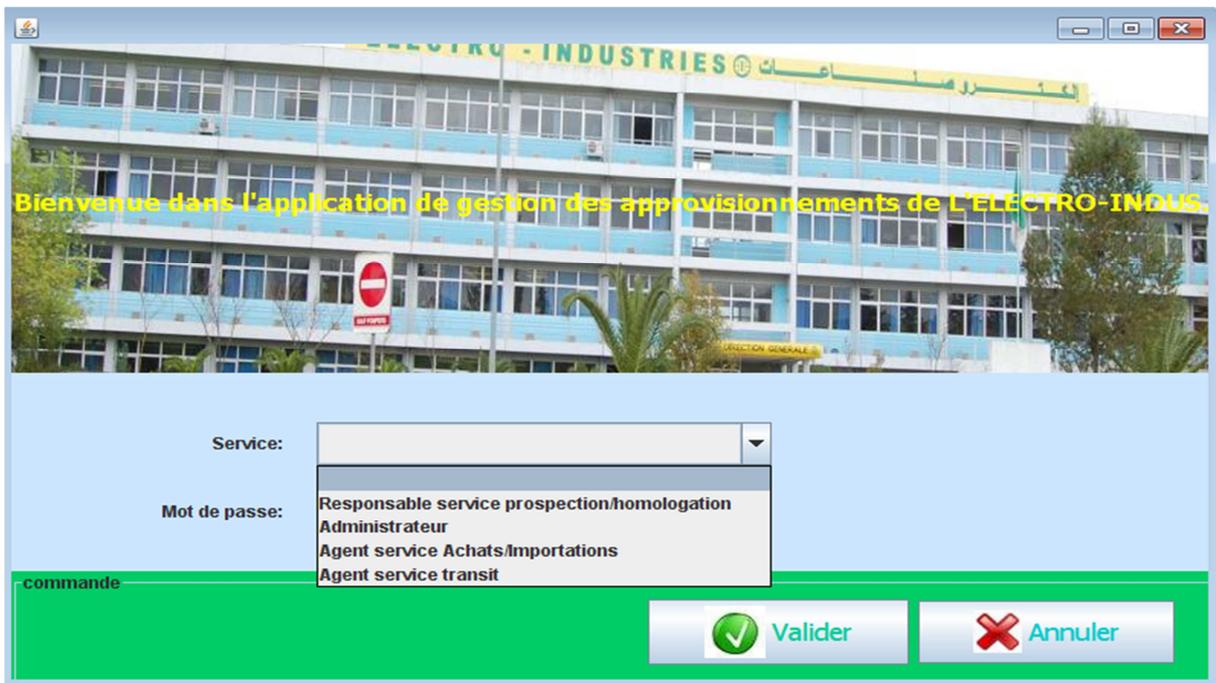
- Table ordre de mission:

Column Name	Data Type	Nullable	Default	Primary Key
NUM_OM	VARCHAR2(4)	No	-	1
NUM_LV	NUMBER	No	-	-
MOYEN_DE_TRANSPORT	VARCHAR2(15)	No	-	-
DATE_DÉPART	DATE	No	-	-
				1 - 4

IV.4. Présentation de quelques interfaces :

Nous n'allons présenter que quelques interfaces :

- Page d'accueil : c'est la première page qui apparaît au lancement de l'application



FigureIV.2 : Page d'accueil

- Page Administrateur :

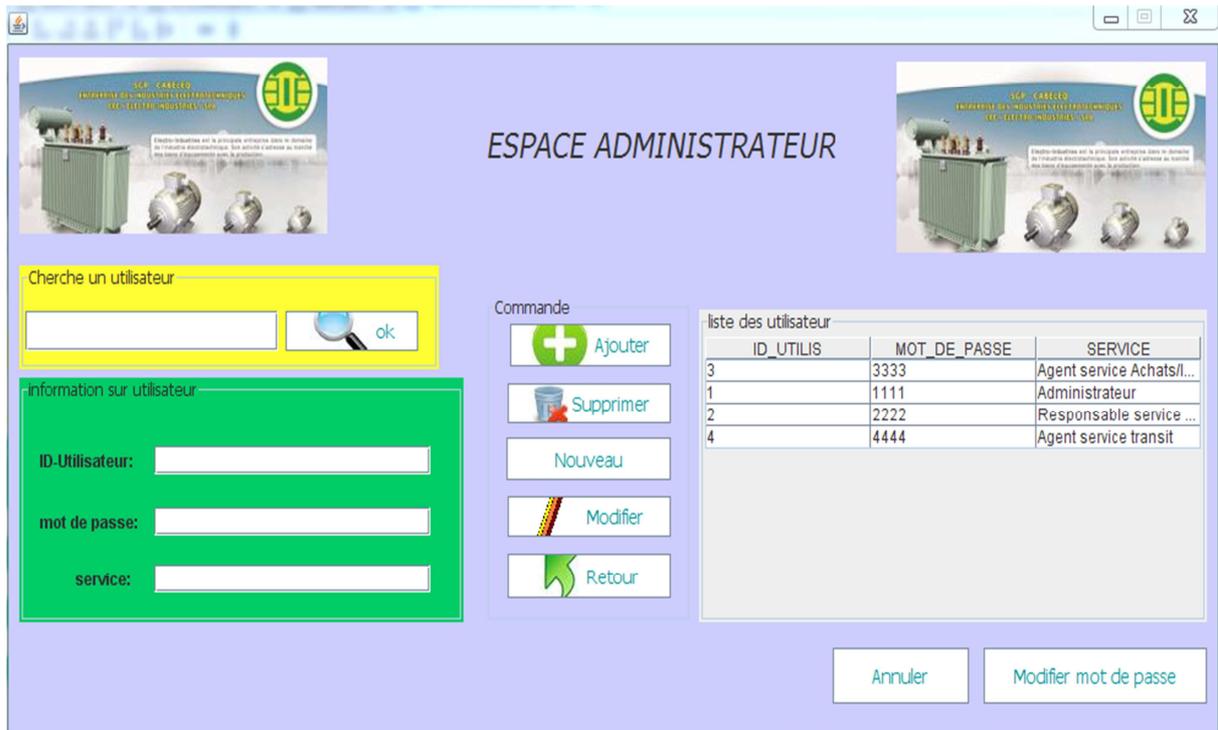


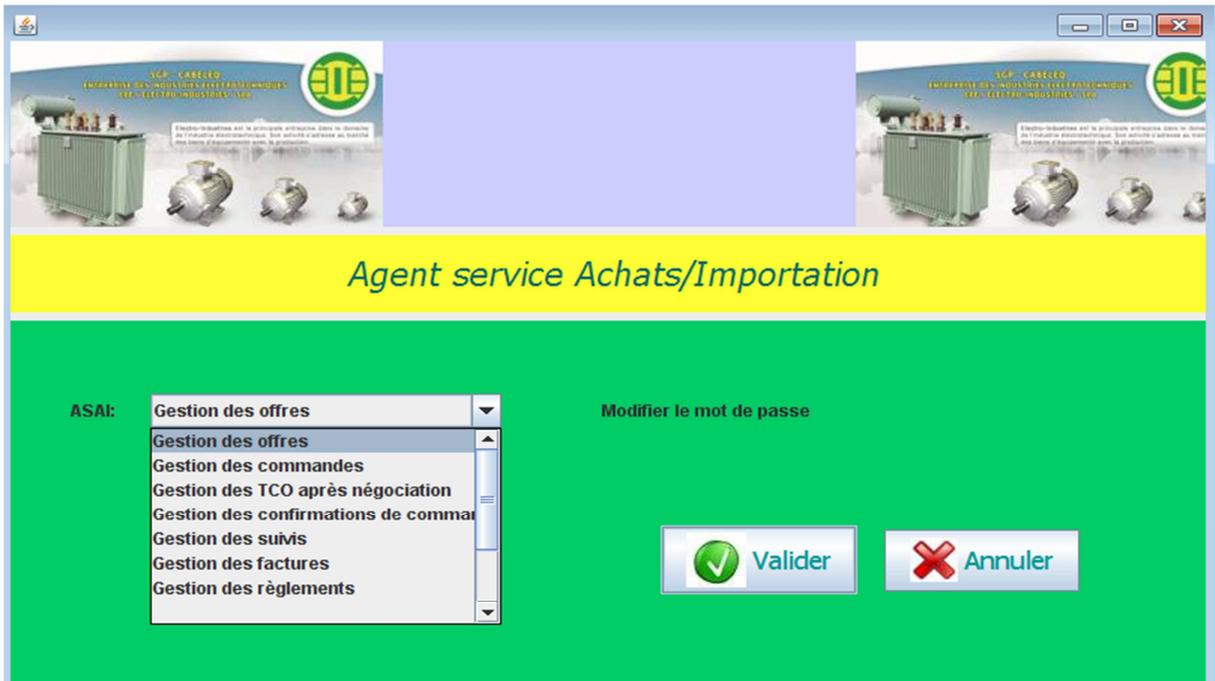
Figure IV.3 : Page Administrateur

- Page modifier le mot passe



Figure IV.4 : Modifie le mot passe

- Page ASAI



FigureIV.5 : ASAI

- Page Gestion des offres :



FigureIV.6 Page Gestion des offres

- Gestion des commandes



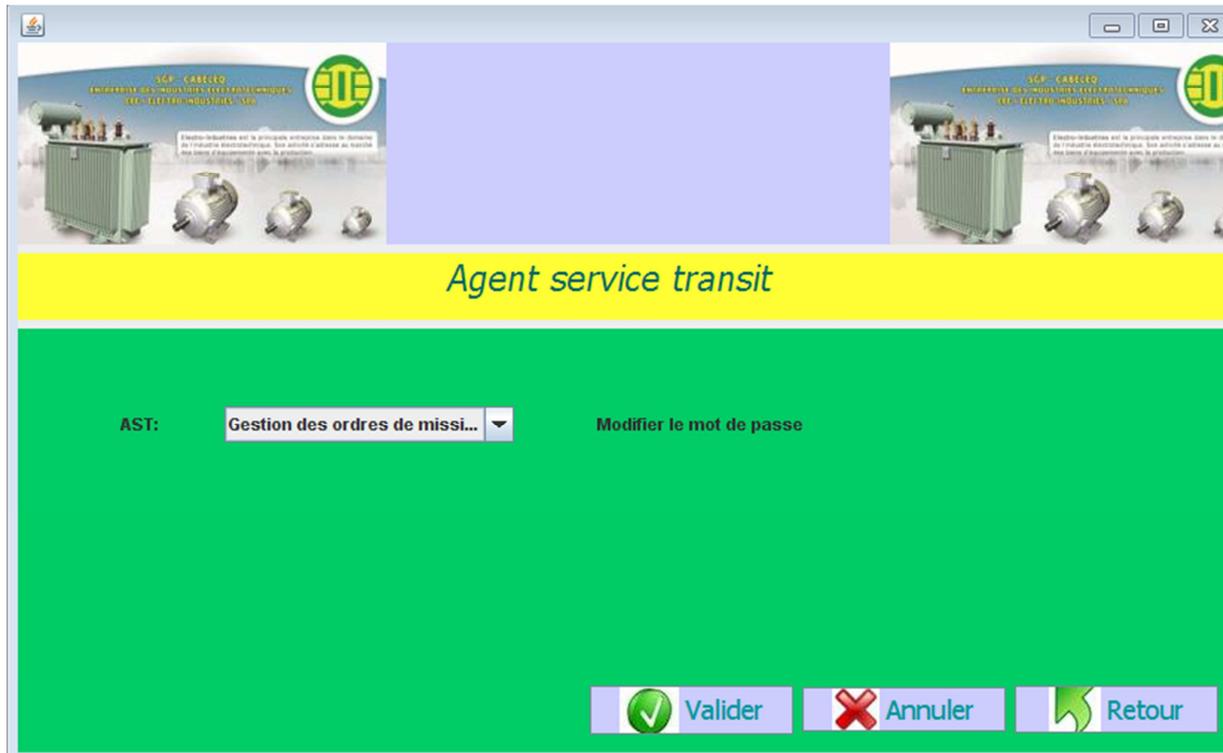
FigureIV.7 Page Gestion des commandes

- Gestion de l'homologation des fournisseurs



FigureIV.8 Page Gestion de l'homologation des fournisseurs

- AST



FigureIV.9.Page agent service transit

Conclusion :

Dans ce chapitre nous avons présenté les outils et l'environnement de développement de l'application.

Nous avons en outre explicité les composants de la base de données, puis pour terminer nous avons présenté quelques interfaces de l'application.

Conclusion

Générale

Conclusion générale :

Nous avons réalisé une application de la gestion des approvisionnements de l'EI basée sur architecture client/serveur 2 tiers. Notre application offre aux différents utilisateurs plus de souplesse et d'efficacité dans l'exécution de leurs tâches, minimise le risque d'erreurs et assure la disponibilité de l'information à toute éventuelle demande.

Le stage que nous avons effectué dans l'enceinte de la direction générale de l'EI nous a permis d'acquérir des connaissances sur la gestion des approvisionnements. De plus, ce projet à travers les étapes de mise en œuvre notamment l'analyse, la conception et la réalisation, nous a conduit à l'utilisation des outils de conception « UML » et de la programmation objet et des bases de données (Oracle en particulier) avec ses outils et fonctions.

Enfin, nous espérons que ce travail puisse servir comme un outil d'aide et de documentation pour les étudiants à venir et un moyen de travail pour le personnel du bureau gestion des approvisionnements.

En guise de perspectives, les possibilités d'évolutions de ce projet sont nombreuses allant jusqu'à la gestion complète de tous les services.

Annexes :

1. Introduction :

Dans ce chapitre, nous allons entamer le processus de développement par une analyse qui mettra en évidence les différents acteurs intervenant dans le système cible ainsi que leurs besoins. La phase conception, s'appuyant sur les résultats de la phase analyse donnera la modélisation des objectifs à atteindre. Pour ce faire, notre démarche va s'appuyer sur le langage UML, qui permet la représentation de la dynamique et la statique du système à travers les différents diagrammes qu'il offre.

2. Définition UML :

UML est le facteur de standardisation, car il est impossible de prétendre imposer une méthode, une manière de faire, à toute l'industrie. En revanche on peut prétendre. Définir un langage de modélisation qui, s'il est suffisamment général, sera adopté comme moyen de communication. UML est un langage qui permet de représenter des modèles, mais il ne définit pas le processus d'élaboration d'un modèle. Cependant, dans le cadre de la modélisation d'une application informatique, les acteurs d'UML préconise d'utiliser une démarche :

- Itérative et incrémentale.
- Guidée par les besoins des utilisateurs de système.
- Centrée sur l'architecture logicielle.

3. La modélisation UML :

UML fournit une panoplie d'outils permettant de présenter l'ensemble des éléments du monde objet (classe ,objet....)ainsi que les liens qui les relie. Toutefois, étant donné qu'une seule représentation est trop subjective ,UML fournit un moyen astucieux permettant de représenter diverse projection d'une même représentation grâce aux vues .une vue est constituée d'un ou plusieurs diagrammes.

On distingue trois types de vues :

✚ Les vues statique : représentant le système physiquement .

- Diagramme de classe
- Diagramme d'objets
- Diagramme de composants
- Diagramme de déploiement

✚ Les vue Dynamique :

- Diagramme d'états
- Diagramme d'activités

- Diagramme de séquence
- Diagramme de collaboration
- ✚ Le vue fonctionnelle :
 - Diagramme de cas d'utilisation
- ✚ UML suit une démarche en trois étapes :
 - Analyse
 - Conception
 - Implémentation
- ✚ Les briques d'UML : composées de :
 - Les éléments
 - Les relations
 - Les diagrammes

4. Eléments d'UML :

Il existe quatre types d'éléments dans UML :

- les éléments structurels.
- les éléments comportementaux.
- les éléments de regroupement.
- les éléments d'annotation.

4.1. Les éléments structurels : Ils représentent les parties statiques du modèle.

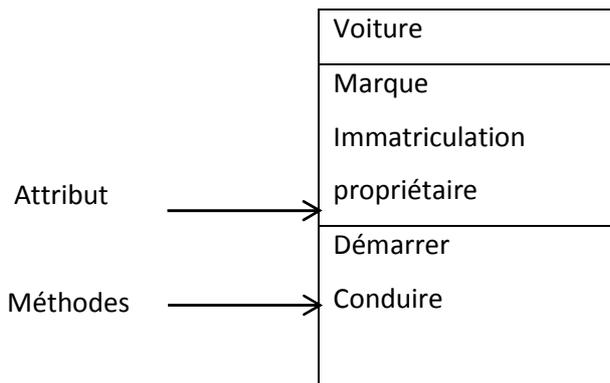
Ceux sont des représentations conceptuelles ou physiques d'un système

Il existe 7 types d'éléments structurels :

- ✚ Classe
- ✚ Interface
- ✚ Collaboration
- ✚ Cas d'utilisation
- ✚ Classe d'activité
- ✚ Composant
- ✚ Nœud

4.1.1. Classe : C'est un ensemble d'éléments ayant les mêmes attributs, les mêmes opérations, les mêmes relations et la même sémantique.

Exemple :



4.1.2. Interface : C'est un ensemble d'opérations définissant la fonction d'un élément ou d'un composant. Elle définit seulement la signature des opérations.

Exemple :



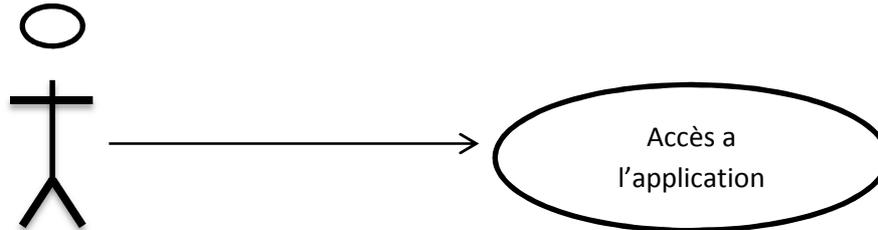
4.1.3 Collaboration : La collaboration Elle définit une interaction entre plusieurs éléments .Elle a un rôle structurel et comportemental.

4.1.4 Cas d'utilisation :

- ❖ Il s'agit de la solution UML pour représenter le modèle conceptuel.
- ❖ Les use cases permettent de structurer les besoins des utilisateurs et les objectifs correspondants d'un système.
- ❖ Ils centrent l'expression des exigences du système sur ses utilisateurs : ils partent du principe que les objectifs du système sont tous motivés.

- ❖ Ils se limitent aux préoccupations "réelles" des utilisateurs ; ils ne présentent pas de solutions d'implémentation et ne forment pas un inventaire fonctionnel du système.
- ❖ Ils identifient les utilisateurs du système (acteurs) et leur interaction avec le système.
- ❖ Ils permettent de classer les acteurs et structurer les objectifs du système.

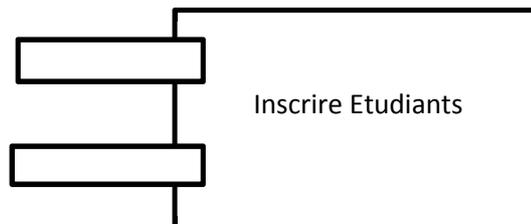
Exemple :



4.1.5 Classe d'activité : Classe particulière dont les objets possèdent un ou plusieurs processus pouvant lancer une activité de commande.

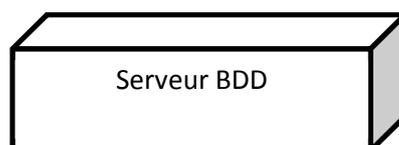
4.1.6 Composant : Le composant est une partie physique d'un système qui conforme a une spécification et fournit la réalisation d'un ensemble d'interface.il peut être un module.exe,dll, com, javaBeans,etc ;il peut aussi être constitué de plusieurs sous composantes.

Exemple :



4.1.7 Nœud : C'est un élément physique. Lors d'une exécution, il représente une ressource ayant une capacité de calcul. En règle général ,il a moins de la mémoire et souvent de capacité de traitement .Un nœud est représenté par un cube.

Exemple :



4.2. les éléments comportementaux,

4.2.1 Les interactions :

Les interactions est l'ensemble des messages échanger entre les éléments du système (résultat de collaboration d'un groupe d'instances), elles sont des messages, des séquences d'actions ou bien des liens (relation entre des éléments). Une interaction peut être visualisée selon le point de vue du temps (diagramme de séquences) ou de celui d'espace (diagramme de collaboration). Un message par exemple est une interaction représenté par une ligne fléchée, indiquant le nom de son opération.

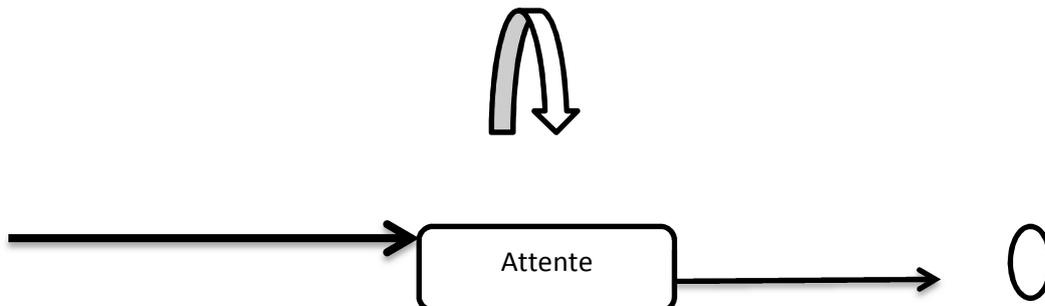
Exemple :



4.2.2 Etat :

La machine à état spécifie le cycle de vie d'un objet quand cet objet a un comportement dynamique. On représente graphiquement un état par un rectangle au coin arrondi (au centre), et on distingue les états de départ (à gauche) et terminaux (à droite). Comme décrit dans figure suivante.

Exemple :



Etat initiale

Etat finale

4.3. Les éléments de regroupement :

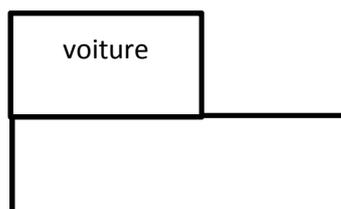
Les éléments de regroupements représentent les parties organisationnelles des modèles UML. Ce sont des boîtes dans lesquelles un modèle peut être décomposé. Il existe un seul type fondamental d'éléments de regroupement : le « paquetage ».

4.3.1 Package :

Un package en UML (ou paquetage en français) est un groupe d'éléments, dans le but de les grouper dans des ensembles cohérents. Un package peut contenir la plupart des éléments UML : classes, objets, cas d'utilisations, composantes, etc. Il peut également contenir des packages, créant une hiérarchie complète.

L'avantage des packages est qu'ils permettent de structurer les diagrammes et donnent une vision globale plus claire.

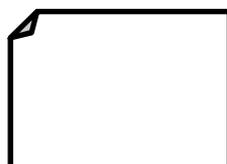
Exemple :



4.3.2 les éléments d'annotation :

Ils représentent les parties explicatives modèles. Ceux des commentaires. Il existe 1 type d'élément d'annotations : Ils font partie des décorations.

Exemple :



5. Les Relation : sont les liens entre les éléments.

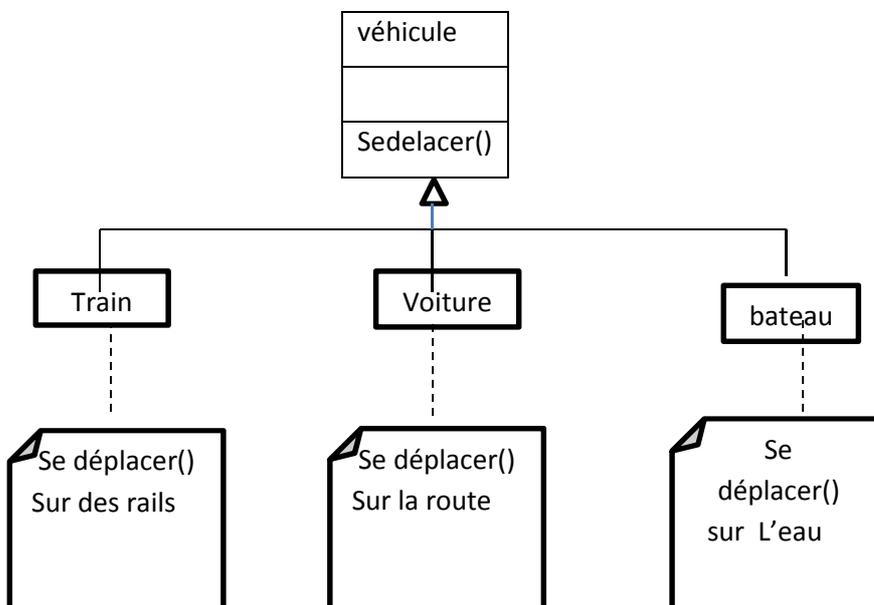
Il existe Cinq types de relation interclasse sont couramment utilisés (il en existe d'autres) :

- héritage
- dépendance
- agrégation
- composition
- Association

- **Héritage :**

- ✓ L'héritage est un mécanisme de transmission des propriétés d'une classe (ses attributs et méthodes) vers une sous-classe.
- ✓ Une classe peut être spécialisée en d'autres classes, afin d'y ajouter des caractéristiques spécifiques ou d'en adapter certaines.
- ✓ Plusieurs classes peuvent être généralisées en une classe qui les factorise, afin de regrouper les caractéristiques communes d'un ensemble de classes

Exemple :



- **Dépendance :**

C'est un lien sémantique entre 2 éléments .Un changement de l'un peut affecter la sémantique de l'autre.

Exemple :



- **Agrégation :**

L'agrégation est un cas particulier de relation d'association qui indique qu'une classe est une partie d'une autre classe. Cette relation comporte également les ordre de multiplicité On la représente graphiquement en décorant la relation d'association par un losange.

Exemple :

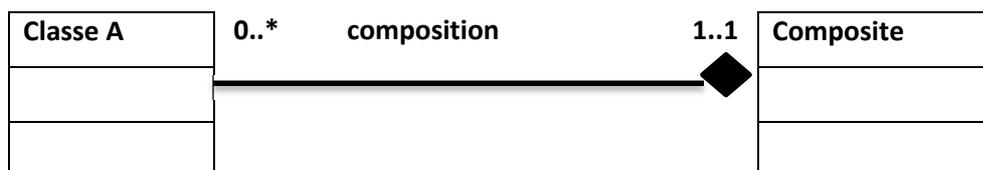


- **Composition :**

Composition Aussi appelée "agrégation forte" ou "agrégation par valeur", il s'agit en fait d'une agrégation à laquelle on impose des contraintes internes : un seul objet peut faire partie d'un composite (l'agrégat de la composition), et celui-ci doit gérer toutes ses parties. En clair, les composants sont totalement dépendants du composite.

En UML, la composition est représentée de la même manière que l'agrégation, mais le diamant est plein.

Exemple :

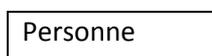


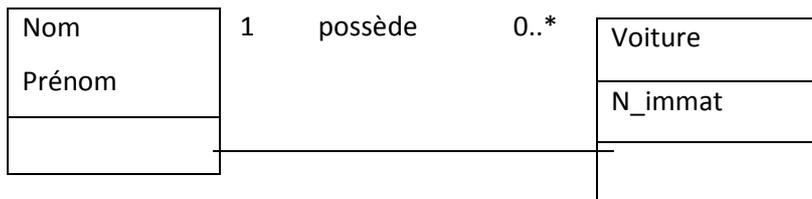
- **Association :**

C'est la relation la plus simple entre deux classes. Elle existe à partir du moment où l'une des deux classes sert de type à un attribut de l'autre, et que cet autre envoie des messages à la première (condition nécessaire pour une association). Simplement, une association indique que deux classes communiquent entre elles (dans un sens ou dans les deux sens).

En UML, une association est représentée par une ligne entre deux classes, possiblement accompagnée d'une flèche si l'association n'est pas bidirectionnelle.

Exemple :



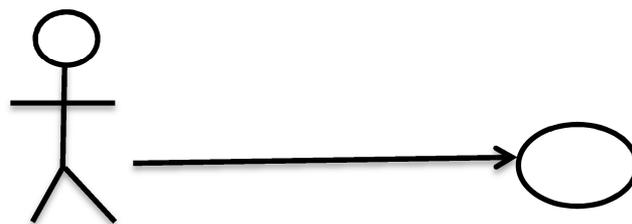


6. Les différents types de diagrammes UML :

6.1 diagramme cas d'utilisation :

- Il s'agit de la solution UML pour représenter le modèle conceptuel. (est de comprendre et structurer les besoins du client.)
- Les use cases permettent de structurer les besoins des utilisateurs et les objectifs correspondants d'un système.
- Ils centrent l'expression des exigences du système sur ses utilisateurs : ils partent du principe que les objectifs du système sont tous motivés.
- Ils se limitent aux préoccupations "réelles" des utilisateurs ; ils ne présentent pas de solutions d'implémentation et ne forment pas un inventaire fonctionnel du système.
- Ils identifient les utilisateurs du système (acteurs) et leur interaction avec le système.
- Ils permettent de classer les acteurs et structurer les objectifs du système.
- Ils servent de base à la traçabilité des exigences d'un système dans un processus de développement intégrant UML.

Exemple :



Acteur A

Cas d'utilisateur x

✓ Intérêt des cas d'utilisation :

- Le but de la conceptualisation est de comprendre et structurer les besoins du client.
- Il ne faut pas chercher l'exhaustivité, mais clarifier, filtrer et organiser les besoins
- Une fois identifiés et structurés, ces besoins :

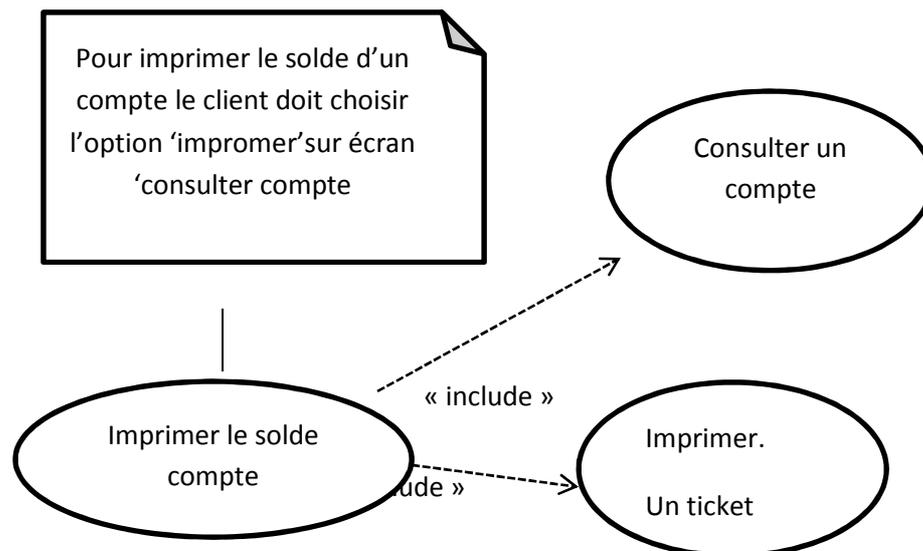
- définissent le contour du système à modéliser (ils précisent le but à atteindre),
- permettent d'identifier les fonctionnalités principales (critiques) du système.
- Le modèle conceptuel doit permettre une meilleure compréhension du système.
- Le modèle conceptuel doit servir d'interface entre tous les acteurs du projet.
- Les besoins des clients sont des éléments de traçabilité dans un processus intégrant UML.

✓ **Les relation entre cas utilisation :**

Il Ya deux type de relation possibles entre cas :

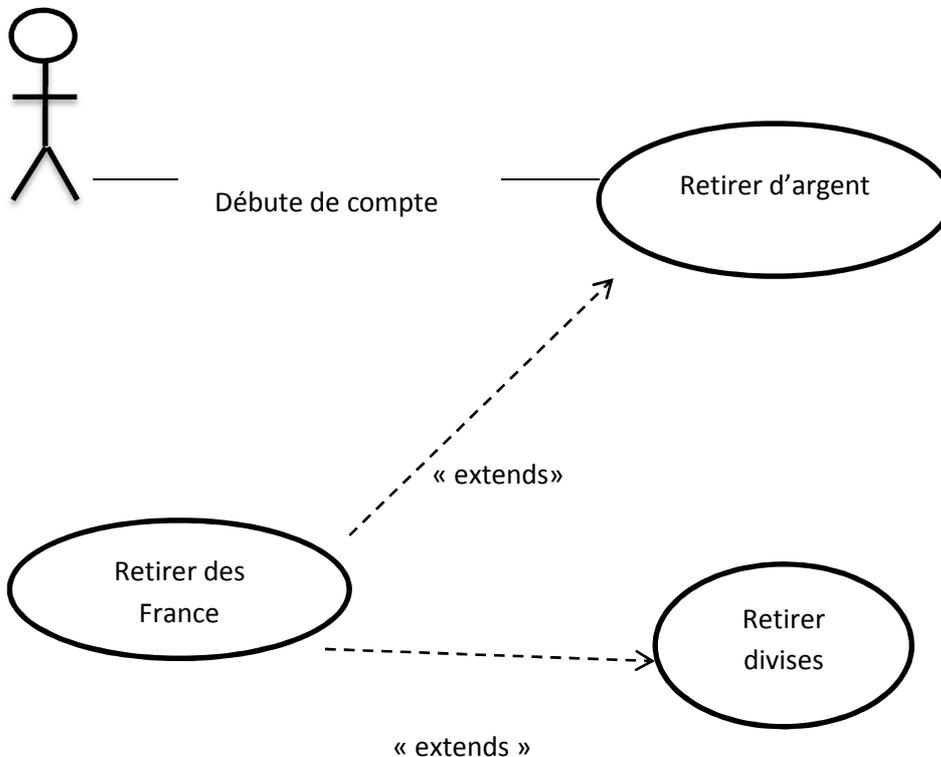
- L'inclusion : quand les cas sources comprend le cas destination
- L'extension : quand le source ajoute son comportement cas.

Exemple :



- ✓ **Relation d'utilisation** : indiquer que le cas d'utilisation source contient aussi le comportement décrit dans le cas d'utilisation destination

Exemple :



Le relation extension indiquer que le cas d'utilisation source étend (précise) les objectifs (le comportement) du cas d'utilisation destination

6.2 diagramme classes :

- ✓ **Diagramme de classes : sémantique**

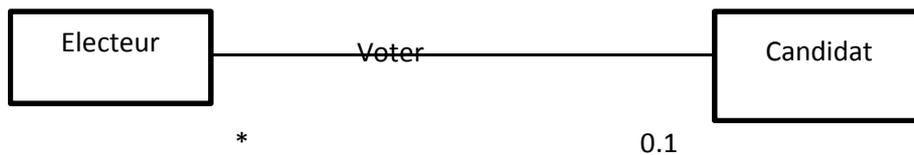
- Un diagramme de classes est une collection d'éléments de modélisation statiques (classes, paquetages...), qui montre la structure d'un modèle.
- Un diagramme de classes fait abstraction des aspects dynamiques et temporels.
- Pour un modèle complexe, plusieurs diagrammes de classes complémentaires doivent être Construits.

On peut par exemple se focaliser sur :

- les classes qui participent à un cas d'utilisation (cf. collaboration),
- les classes associées dans la réalisation d'un scénario précis,
- les classes qui composent un paquetage,

o la structure hiérarchique d'un ensemble de classes.
Les diagramme de classes permettent de représenter l'ensemble des informations formalisés, qui sont regrouper des classes.

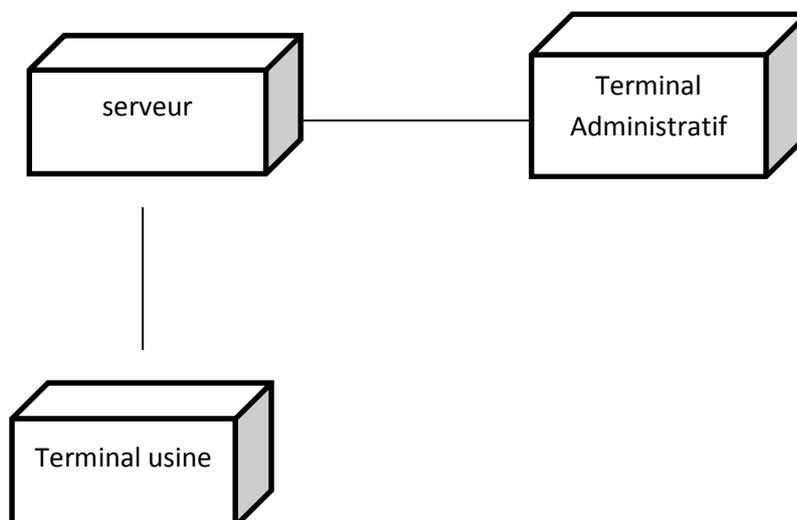
Exemple :



6.3. Diagramme de déploiement :

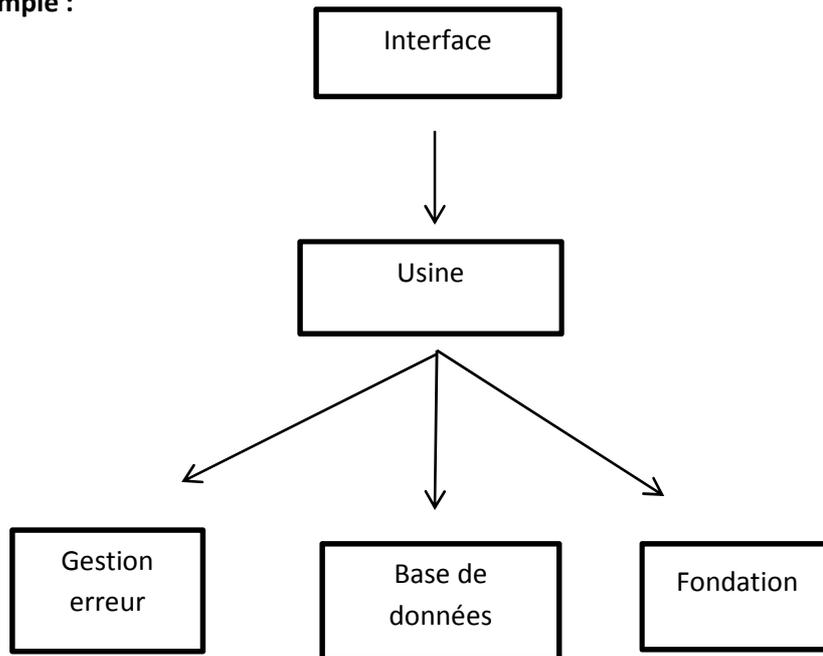
- Les diagrammes de déploiement montrent la disposition physique des matériels qui composent le système et la répartition des composants sur ces matériels.
- Les ressources matérielles sont représentées sous forme de nœuds.
- Les nœuds sont connectés entre eux, à l'aide d'un support de communication. La nature des lignes de communication et leurs caractéristiques peuvent être précisées.
- Les diagrammes de déploiement peuvent montrer des instances de nœuds (un matériel précis), ou des classes de nœuds.
- Les diagrammes de déploiement correspondent à la vue de déploiement d'une architecture logicielle .

Exemple :



6.4 diagramme de composants : Il représente l'organisation et les dépendances des composants. Un composant est soit une (ou plusieurs) classe(s), interface(s) ou collaboration(s).

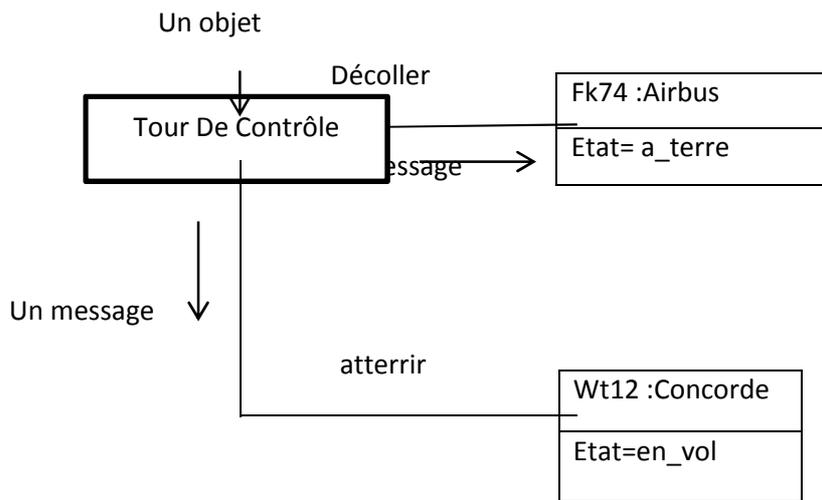
Exemple :



6.5 Diagramme de collaboration

- Les diagrammes de collaboration montrent des interactions entre objets (instances de classes et acteurs).
- Ils permettent de représenter le contexte d'une interaction, car on peut y préciser les états des objets qui interagissent.

Exemple :



6.6 Diagramme de séquence : sémantique

C'est un diagramme d'interaction. Il représente un ensemble d'objets et leurs relations, avec les messages qu'ils échangent. (aspect chronologique des messages).

6.6.1. Les différents types de messages :

- Message simple
- Message minuté
- Message synchrone
- Message asynchrone
- Message déroband

🚦 message simple

Message dont on ne spécifie aucune caractéristique d'envoi ou de réception particulière.

🚦 message minuté (timeout)

Bloque l'expéditeur pendant un temps donné (qui peut être spécifié dans une contrainte), en attendant la prise en compte du message par le récepteur. L'expéditeur est libéré si la prise en compte n'a pas eu lieu pendant le délai spécifié.

🚦 message synchrone

Bloque l'expéditeur jusqu'à prise en compte du message par le destinataire. Le flot de contrôle passe de l'émetteur au récepteur (l'émetteur devient passif et le récepteur actif) à la prise en compte du message.

🚦 message asynchrone

N'interrompt pas l'exécution de l'expéditeur. Le message envoyé peut être pris en compte par le récepteur à tout moment ou ignoré (jamais traité).

✚ message déroband

N'interrompt pas l'exécution de l'expéditeur et ne déclenche une opération chez le récepteur que s'il s'est préalablement mis en attente de ce message.

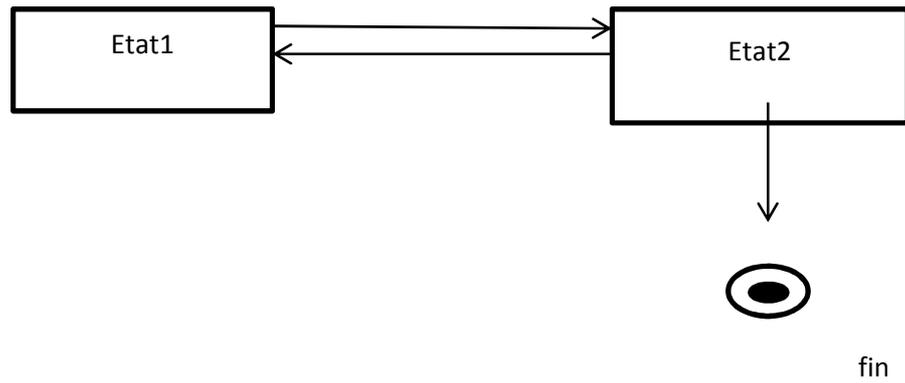
6.7. Diagramme d'états-transitions : sémantique

- Ce diagramme sert à représenter des automates d'états finis, sous forme de graphes d'états, reliés par des arcs orientés qui décrivent les transitions.
- Les diagrammes d'états-transitions permettent de décrire les changements d'états d'un objet ou d'un composant, en réponse aux interactions avec d'autres objets/composants ou avec des acteurs.
- Un état se caractérise par sa durée et sa stabilité, il représente une conjonction instantanée des valeurs des attributs d'un objet.
- Une transition représente le passage instantané d'un état vers un autre.
- Une transition est déclenchée par un événement. En d'autres termes : c'est l'arrivée d'un événement qui conditionne la transition.
- Les transitions peuvent aussi être automatiques, lorsqu'on ne spécifie pas l'événement qui la déclenche.

Exemple :

Début





Conclusion :

L'UML, comme l'on a vu à travers ce chapitre, ne propose pas une démarche objet mais une notation adapté au monde de développement orienté objet. Il nous a donc permis de s'initier aux techniques de modélisation objet. La notation UML peut s'adapter a tous les projets informatiques.

1.Introduction :

Dans les productions informatiques d'aujourd'hui, il est inconcevable de perdre des données. Pourtant, les problèmes matériels et/ou erreurs utilisateurs peuvent toujours exister. Et là... c'est le drame !

Nous savons que nous devons faire des sauvegardes régulières afin de pouvoir retrouver nos données mais s'il faut restaurer les données, cela se traduit par une indisponibilité du service (au moins partielle) tant que la restauration n'est pas terminée.

Il existe plusieurs solutions pour permettre de récupérer rapidement une information et nous allons nous intéresser aux solutions d'infrastructure de résistance aux pannes proposées par Oracle.

2.Définition SGBD :

Un SGBD est un système complexe permettant de gérer de manière efficace, un volume important de données structurées, accessible par des utilisateurs simultanés locaux ou non. On plus SGBD est un ensemble d'outils logiciels permettant la création et utilisation de bases de données

3.Caractéristique d'un SGBD(système de gestion des base de données) :

- **gestion de gros volumes de données** (en consultation et en mise à jour),
- **Sécurité des données, qui se décline en :**

- ✓ **disponibilité**

Un SGBDR(système de gestion des base de données relationnelles) se doit d'offrir une bonne disponibilité des données. Une disponibilité totale des données est possible (temps de reprise nul) il suffit de s'en donner les moyens logiciels et matériels..

- ✓ **fiabilité**

Des mécanismes de sauvegarde variés (physique, logique, off-line, on-line, totale, partielle, incrémentale), ainsi que des mécanismes de journalisation, et de reprise permettent de restaurer une information sans pratiquement aucune perte, dans tous les cas de problème matériel ou logiciel.

- ✓ **confidentialité**

Tout n'est pas accessible à tout le monde! Se connecter à la base de données, donne un certain nombre de droits et de ressources en fonction d'un profil défini et maintenu par un administrateur. La granularité d'accès peut aller jusqu'à la vision unique d'un champ d'un enregistrement d'une table particulière. Encore une fois on ne manipule plus des fichiers..

- ✓ **cohérence**

Que les données soient réparties ou non –dans ce dernier cas les mécanismes mis en jeu seront plus complexes– elles doivent être cohérentes. Cela sous entend, d'une part que les accès concurrents d'utilisateurs, notamment lors de mises à

jour, ne doivent pas compromettre l'intégrité des données et d'autre part que ces dernières satisfassent aux contraintes d'intégrité du modèle, mais aussi aux règles de gestion de l'entreprise.

- **concurrence d'accès en lecture et écriture** (avec une bonne granularité),
- **gestion (efficace) des transactions,**
- **portabilité sur différents OS, des données et du code**
- **Administrable:**
 - existence d'outils d'administration généraux : gestion des données, des utilisateurs, des fichiers physiques, des espaces logiques, des droits, des profils, des ressources systèmes, etc.
 - outils de surveillance en temps réel grâce à un moniteur, si possible graphique ou en temps différé grâce à des journaux ou à des traces paramétrables.
- **performances:**

offrir de bonnes performances et des outils permettant de les mesurer et de les contrôler via des paramètres de configuration. Des processus d'optimisation en temps réel des requêtes complexes sont également souvent présents. Les données peuvent être indexées, de manière souple, dynamique et complète (index simples, concaténés, multiples, tables de hashage, recherche textuelle, etc.). Un nombre important d'utilisateurs, ainsi qu'un volume conséquent de données peuvent être pris en compte.

4. Dans quels cas utiliser Oracle Data base ?

- Traitement d'un gros volume de données (besoin d'optimiser l'utilisation de la mémoire, besoin important de performance sur traitements, ...)
- Entreprises ayant une problématique de sécurité (redondance et accès)
- Traitement des données particulières (Oracle Spatial)
- Nécessite souvent un Administrateur

5. Présentation d'oracle :

5.1. Historique oracle :

Oracle est un **SGBD** (système de gestion de bases de données) édité par la société du même nom (Oracle Corporation), leader mondial des bases de données.

La société *Oracle Corporation* a été créée en 1977 par Lawrence Ellison, Bob Miner, et Ed Oates. Elle s'appelle alors *Relational Software Incorporated (RSI)* et commercialise un Système de Gestion de Bases de données relationnelles (SGBDR ou RDBMS pour *Relational Database Management System*) nommé *Oracle*.

En 1979, le premier prototype (RDBMS - RSI1) intégrant la séparation des espaces d'adressage entre les programmes utilisateurs et le noyau Oracle est commercialisé. Cette version est entièrement développée en langage assembleur. La seconde version (RDBMS - RSI2) est un portage de l'application sur d'autres plates-formes.

En 1984 la première version d'Oracle (Oracle 4) est commercialisée sur les machines IBM.

En 1992, Oracle 7 sort sur les plates-formes UNIX (elle ne sortira sur les plates-formes Windows qu'à partir de 1995). Cette version permet une meilleure gestion de la mémoire, du CPU et des entrées-sorties. La base de données est accompagnée d'outils d'administration (SQL*DBA) permettant une exploitation plus aisée de la base. En 1997, la version Oracle 7.3 (baptisée *Oracle Universal Server*) apparaît, suivie de la version 8 offrant des capacités objet à la base de données

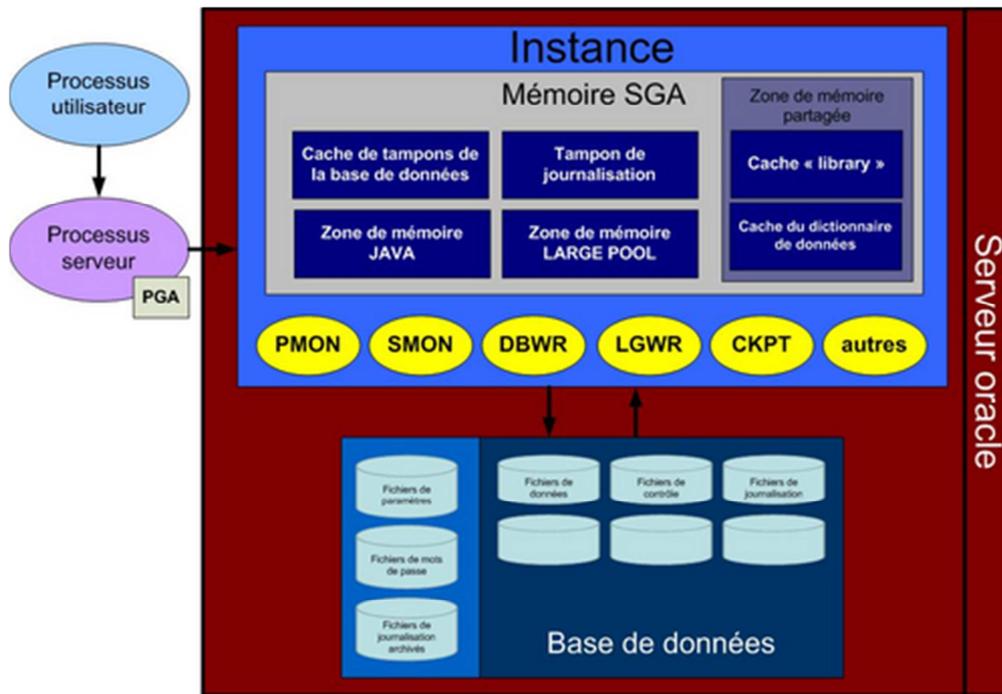
Oracle est écrit en langage C et est disponible sur de nombreuses plates-formes matérielles (plus d'une centaine) dont :

- AIX (IBM)
- Solaris (Sun)
- HP/UX (Hewlett Packard)
- Windows NT (Microsoft)

Oracle depuis la version 8.0.5 est disponible sous Linux

5.2.Définition oracle :

Un serveur **Oracle** est un système qui permet de gérer les bases de données et qui offre moyen de gestion des informations ouvert, complet et intégré. Un serveur **Oracle** est constitué d'une instance et d'une base de données.



5.3.Instance Oracle :

L'instance **Oracle** permet d'accéder à la base de données **Oracle** et ne permet l'ouverture que d'une seule base de données.

L'instance **Oracle** est constituée :

- **des processus d'arrière plan** : qui gèrent et appliquent les relations entre les structures physiques et les structures mémoires. Il en existe deux catégories
 - ✓ les processus d'arrière plan obligatoires : DBWR, PMON, CKPT, LGWR, SMON
 - ✓ les processus d'arrière plan facultatifs : ARCn, LMDn, RECO, CJQ0, LMON, Snnn, Dnnn, Pnnn, LCKn, QMnN
- **structures mémoires** : elle se compose essentiellement de deux zones mémoires :
 - ✓ la zone mémoire allouée à la SGA (System Global Area) : elle est allouée au démarrage de l'instance et qui est représentée une composante fondamentale d'une instance **Oracle**. Elle se compose de plusieurs zones mémoire :
 - ❖ la zone mémoire partagée
 - ❖ le cache de tampons de la base de données
 - ❖ le tampon de journalisation ainsi que d'autres structures pour la gestion de verrous externes (lock), internes (latch), données statistiques, etc...
 - ❖ Eventuellement il est aussi possible de configurer au niveau de la SGA
 - la zone mémoire LARGE POOL
 - la zone mémoire Java

5.4. Les processus oracle :

5.4.1. Le processus utilisateur

C'est le programme qui demande une interaction avec la base de données en démarrant une connexion. Il communique uniquement avec le processus serveur correspondant.

5.4.2. Le processus serveurs

Il représente le programme qui rentre directement en interaction avec le serveur **Oracle**. Il répond à toutes les demandes et renvoi les résultats. Il peut être dédié à un serveur client ou partagé par plusieurs.

5.5. Fonctionnalités oracle :

Oracle est un SGBD permettant d'assurer :

- La définition et la manipulation des données
- La cohérence des données
- La confidentialité des données
- L'intégrité des données
- La sauvegarde et la restauration des données
- La gestion des accès concurrents
- Possibilité de stocker de gros volumes de données
- Partitionnement des grosses tables

5.6. Les composants d'Oracle :

Outre la base de données, la solution Oracle est un véritable environnement de travail constitué de nombreux logiciels permettant notamment une administration graphique d'Oracle, de s'interfacer avec des produits divers et d'assistants de création de bases de données et de configuration de celles-ci.

On peut classer les outils d'Oracle selon diverses catégories :

- Les outils d'administration
- Les outils de développement
- Les outils de communication
- Les outils de génie logiciel
- Les outils d'aide à la décision

5.6.1. Les outils d'administration d'Oracle :

Oracle est fourni avec de nombreux outils permettant de simplifier l'administration de la base de données. Parmi ces outils, les plus connus sont :

- Oracle Manager (SQL*DBA)

- NetWork Manager
- Oracle Enterprise Manager
- Import/Export : un outil permettant d'échanger des données entre deux bases Oracle

5.6.2. Les Outils de développement d'Oracle :

Oracle propose également de nombreux outils de développement permettant d'automatiser la création d'applications s'interfaçant avec la base de données. Ces outils de développement sont :

- Oracle Designer
- Oracle Developer
- SQL*Plus : une interface interactive permettant d'envoyer des requêtes SQL et PL/SQL à la base de données. SQL*Plus permet notamment de paramétrer l'environnement de travail (formatage des résultats, longueur d'une ligne, nombre de lignes par page, ...)
- Oracle Developer : il s'agit d'une suite de produits destinés à la conception et à la création d'applications client-serveur. Il est composé de 4 applications :
- Oracle Forms (anciennement SQL*Forms) : un outil permettant d'interroger la base de données de façon graphique sans connaissances préalables du langage SQL. SQL*Forms permet ainsi de développer des applications graphiques (fenêtres, formulaires, ...) permettant de sélectionner, modifier et supprimer des données dans la base.
- Oracle Reports (SQL*ReportWriter) : un outil permettant de réaliser des états.
- Oracle Graphics : un outil de génération automatique de graphiques dynamiques pour présenter graphiquement des statistiques réalisées à partir des données de la base.
- Procedure Builder : un outil permettant de développer des procédures, des fonctions et des packages

5.7. Avantages d'Oracle Data base :

- Possibilité de choisir entre une installation automatique ou paramétrer son installation à 100%
- Gestion entièrement automatique de la mémoire
- Gestion avancée de la compression des données
- Très performant sur des gros volumes de données
- Vues matérialisées...

Conclusion :

Cette partie nous a permis de prendre connaissance des produits d'oracle corporation, de voir la puissance de son SGBD et de nous initier dans le détail aux différents langages et technologie de programmation à utiliser pour lancer le développement et l'implémentation des différents composants de l'application (base de données, interfaces et application client / serveur ...) proposés.

1. Introduction :

processus unifié (up) Le processus unifié est un processus de développement logiciel : il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel. Caractéristiques essentielles du processus unifié :

- ✓ Le processus unifié est à base de composants,
- ✓ Le processus unifié utilise le langage UML (ensemble d'outils et de diagramme),
- ✓ Le processus unifié est piloté par les cas d'utilisation,
- ✓ Centré sur l'architecture,
- ✓ Itératif et incrémental

2. RUP (Rational Unified Process) :

- Dérivée de UP (UP a été créée en 1996)
- Commercialisée par IBM en 1998.

2.1.Méthode de développement logiciel :

- itérative,
- Incrémentale
- pilotée par les cas d'utilisation.
- centrée sur l'architecture et la réduction des risques
- Produit de qualité

2.2.Phases du RUP :**2.2.1.Phase d'inception :**

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

2.2.2.Phase d'élaboration :

- ✚ Planifier le projet,
- ✚ spécifier les fonctionnalités
- ✚ construire l'architecture
- ✚ Spécification du produit
- ✚ Conception de l'architecture
- ✚ Planification activités-ressources

2.2.3. Phase de construction :

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

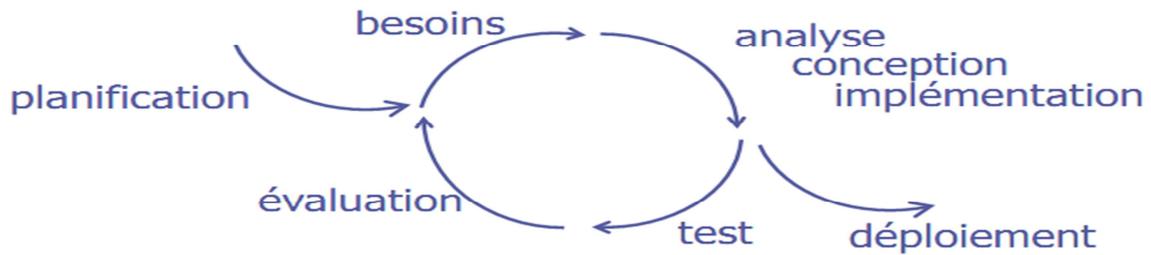
2.2.4. Phase de transition :

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

2.3. Cycle de vie du RUP :

- **Exigence** détermination des besoins :
 - fonctionnels (ce que l'on attend du système)
 - non fonctionnels (fiabilité, temps de réponse, environnement distribué, etc.)
- **Analyse et conception** : évoluer depuis la spécification des besoins jusqu'à une solution informatique
 - analyse besoins fonctionnels
 - conception intègre aussi les besoins non fonctionnels
- **Implémentation** :
 - Transcription dans un langage de programmation ou de base de données .
 - Utilisation de composants existants
- **Test** :
 - Estimer
 - si les besoins sont satisfaits . s'il y a des erreurs/défauts à corriger Renforcer et stabiliser l'architecture
- **Déploiement** :
 - Distribuer le logiciel dans son environnement opérationnel ,Installation,
 - test Formation des utilisateurs ,Migration des données

Cycle de base



Cycle de vie du RUP

2.4. Différence entre incrémental et itératif

✓ RUP est incrémental

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

✓ RUP est itératif

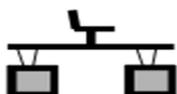
Une itération est une séquence d'activités qui répond à un plan et à des critères d'évaluation et qui produit une version exécutable

Exemple

Différence entre incrémental et itératif



Incrémental



Itératif

❖ **Forces de la méthode RUP**

Traçabilité à partir des Uses Cases jusqu'au déploiement

Approche basée sur l'architecture

Gestion des risques dans les projets

❖ **Faiblesses de la méthode RUP**

Coût de personnalisation souvent élevés

Très axé processus

Conclusion :

Cette partie nous a permis de prendre connaissance le processus rup (rationnel unified process) est une méthode de développement pour logiciel orienté objet ,c'est une méthode générique itérative et incrémentale.

Bibliographie

[ARS 09]: Mr ARHAB Samir & Mr SADOU Samir

« Conception et réalisation d'une application sous oracle, cas : Gestion du budget d'investissement chez SONATRACH, HASSI R'MELL» Mémoire d'ingénieur UMMTO, 2008/2009.

[GUP 05]: Guy Pujolle ; «Les réseaux»; Edition 2005.

[DJADJN 2010]: Mr DJEBRANI Aziz & Mr DJEMA Nacer

« Conception et réalisation d'une application client/serveur sous ORACLE

Cas : Gestion de stock et approvisionnement Du groupe ETRHB HADDAD

[ASG 08] : Mr ASKEUR Akli & GUIDDIR Mohand Akli ;

«Conception et réalisation d'une base de données distribuée, multimédia sous Oracle
9» Mémoire d'ingénieur UMMTO, 2007/2008.

[GEO 00]: George et Olivier Gardain; «client / serveur»; Edition 2000.

[ORD 99] : Orfali Robert, Dan HARKEY et jery EDWARDS ;

Traduction en français LEROY et jean-piere GOUT ;

« client/serveur guide de service » ;

[GEO 00]: George et Olivier Gardain; «client / serveur»;

[BNOK] MR BELKEBIR NAZIM & OUKOULOU KHELIFA

« conception et réalisation d'une application client/serveur sous oracle »

Cas : gestion des approvisionnements de l'électro-industries

Mémoire d'ingénieur UMMTO 2008/2009

[GUY PUJOLLE]. « Les réseaux informatique », Edition 2005

[CHCK] Chikhi hassene & Chaba kamel

« conception et réalisation d'une application client/serveur sous oracle »

Et visuel studio 2008 pour le calcul de la paie Cas : ENIEM

[MAG 98] : MaranGraphics ; «Apprendre les réseaux» ; Edition 1998.

[CCM 09]: <http://www.commentcamarche.net> ; année 2009.

[USER] : www.user.skynet.be/ybet

[WIKIPÉDIA] : <http://www.wikipedia.com>

[CCM] : <http://www.commentcamarche.net>