



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU
FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE



DEPARTEMENT D'INFORMATIQUE

Mémoire de fin d'études

En vue de l'obtention du diplôme de Master II en informatique
Options : Ingénierie des Systèmes d'information.

Thème :

Conception et réalisation d'une application web Avec J2EE

Cas gestion du papier pour la DGSI/SONELGAZ.

Proposé et dirigé par :

M^r HAMMACHE (UMMTO)

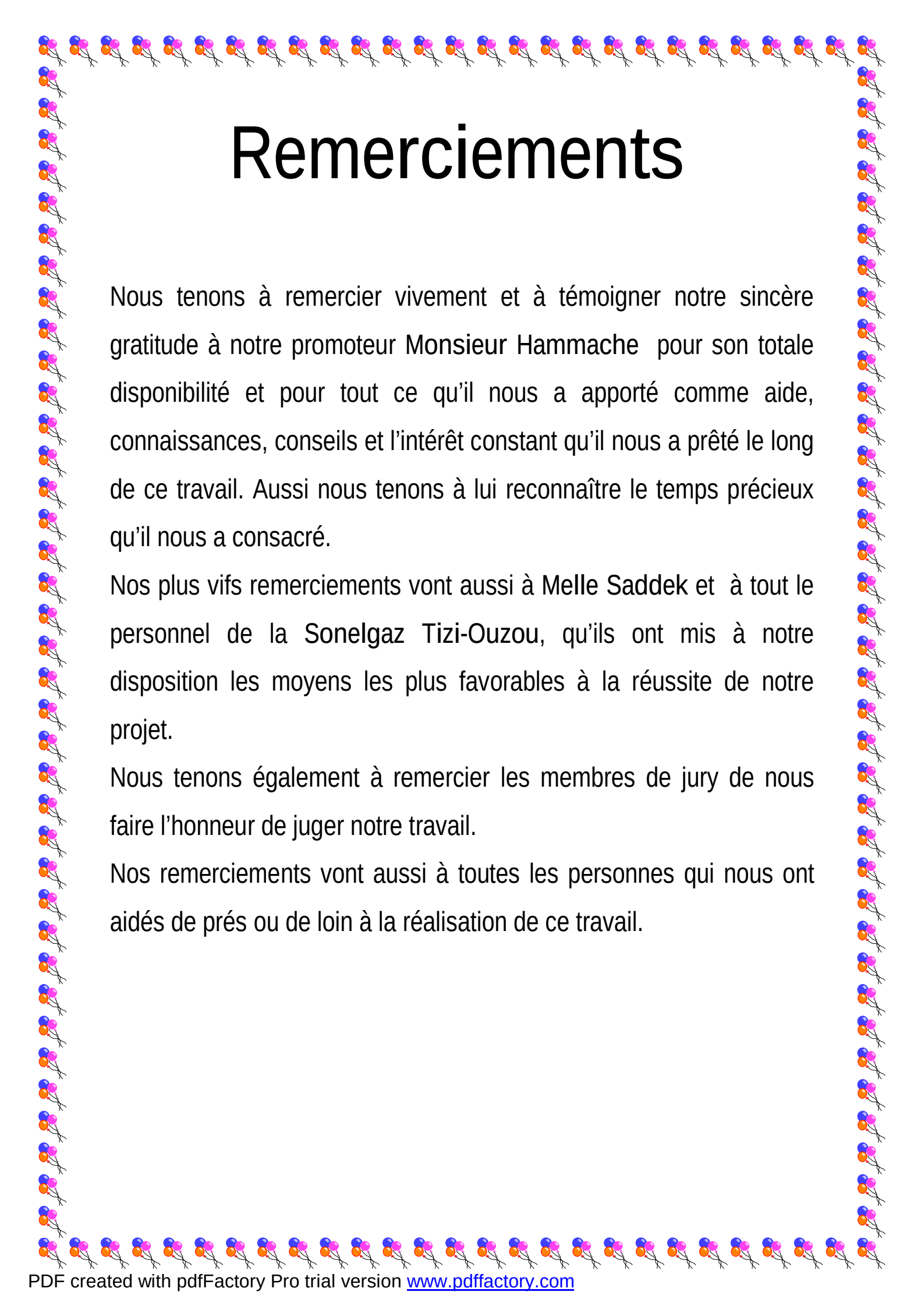
M^{elle} SADDEK (SONELGAZ)

Réalisé par :

M^r BENASSIL Mouloud

M^r ADJAUD Meziane

Promotion : 2012/2013



Remerciements

Nous tenons à remercier vivement et à témoigner notre sincère gratitude à notre promoteur Monsieur Hammache pour son totale disponibilité et pour tout ce qu'il nous a apporté comme aide, connaissances, conseils et l'intérêt constant qu'il nous a prêté le long de ce travail. Aussi nous tenons à lui reconnaître le temps précieux qu'il nous a consacré.

Nos plus vifs remerciements vont aussi à Melle Saddek et à tout le personnel de la Sonelgaz Tizi-Ouzou, qu'ils ont mis à notre disposition les moyens les plus favorables à la réussite de notre projet.

Nous tenons également à remercier les membres de jury de nous faire l'honneur de juger notre travail.

Nos remerciements vont aussi à toutes les personnes qui nous ont aidés de près ou de loin à la réalisation de ce travail.

D édicaces

A mes très chers parents

A mes sœurs et mes frères

A tous mes amis(es)

A toute la famille

M alad

D é d i c a c e s

Je dédie ce modeste travail à :

La lumière de ma vie qui n'a jamais cessé de m'éclairer et à qui Je dois tout (Dieu tout puissant) ;

A ma très chère mère et mon défunt père que Dieu lui offre le paradis eternal ;

A ma sœur et frères (Karima, Achour, Ramdane);

A tous mes cousins et cousines ;

A tous (tes) mes amis (es) ;

A mon binôme Mouloud et à toute sa famille;

A toutes les personnes qui pensent à moi de près ou de loin;

A toutes les personnes qui me connaissent ;

A tous les étudiants et personnels de département informatique en particulier la promotion (ISI 2012/2013).

Meziane

Résumé du Mémoire

Dans ce mémoire nous avons réalisé une application web pour la gestion du papier pour le Service « **la Division Gestion des Systèmes Informatiques DGSi** » de **SONELGAZ** au niveau de la Direction de Distribution de Tizi Ouzou.

Cette application fait la gestion de tout le papier entrant et sortant de stock et la mise à jours des types de papier (ajout de nouveau type de papier, réforme un type de papier non utilisé et la mise à jour la base de données), ainsi que des gérer les commandes de papier et ces factures.

Dans cette application nous avons choisi la technologie J2EE pour développer cette application web à l'aide des JSP et Servlets. Pour ce qui est du mémoire nous l'avons devisé en 4 chapitres qui sont organisés comme suit :

- Ø Un premier chapitre, intitulé « Présentation de l'existant » présente l'organisme d'accueil, ainsi que la problématique.
- Ø Un second chapitre, intitulé « Technologies web » donne un aperçu général sur les nouvelles technologies et de la communication (NTIC).
- Ø Un troisième chapitre, intitulé « Analyse et conception » est consacré à la conception de l'application en utilisant le langage de modélisation UML
- Ø Un quatrième chapitre, intitulé « Réalisation » qui détail principalement le fonctionnement de l'application et les outils utilisés pour son développement.

Sommaire

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

I : Organisme d'accueil

I.1 Définition et historique	2
I.2 Société de Distribution de l'Electricité et du Gaz du Centre (SDC)	4
I.3 La Direction de Distribution de Tizi-Ouzou	5
I.3.1 Missions de La Direction de Distribution.....	5
I.3.2 Organigramme général de la direction de distribution	7
I.3.3 Mode d'organisation de la direction de distribution	8
a. Chargé de la Communication.....	8
b. Chargé de la Sécurité.....	8
c. Chargé de la Sécurité interne	9
d. Chargé des Affaires Juridiques.....	9
e. Division Relations Commerciales	10
f. Division étude d'exécution des travaux d'Electricité et du Gaz	10
g. Division Finance et Comptabilité.....	11
h. Division Exploitation (Electricité ou Gaz).....	12
i. Division Ressource Humain.....	12
j. Divisions des Affaires Générales	12
k. Division Gestion des Systèmes Informatiques	13
I.3.4 Présentation du système informatique existant	14
I.3.5 Problématique.....	14
I.3.6 Objectifs.....	15
I.4 Conclusion	15

Chapitre II : Développement d'application

II.1 Introduction	16
II.2 quelques concepts sur les réseaux informatique	16
II.2.1 Définition d'un réseau	16
II.2.2 caractéristique d'un réseau	16
II.2.3 Le modèle OSI.....	17
II.2.4 Le modèle TCP/IP	19
a. Couche application.....	19
b. Couche transport	20
c. Couche Internet	20
d. Couche Accès Réseaux.....	21
II.2.5 Sécurité et administration.....	21

II.3L 'architecture client/serveur.....	22
II.3.1 Définition.....	22
II.3.2 Notions de bases	22
II.3.3 Fonctionnement d'un système client/serveur.....	23
II.3.4 Les types de serveurs	23
a. Serveur de bases de données.....	23
b. Serveur d'objet.....	24
c. Serveur de transaction	24
d. Le serveur web	24
e. Les serveurs de groupware	24
f. Un serveur de fichiers.....	24
II.3.5 Types d'architectures Client – Serveur.....	25
II.3.5.1 Architecture à 2 niveaux	25
II.3.5.2 Architecture à 3 niveaux.....	26
II.3.5.3 Comparaison entre client/serveur 2 tiers et client/serveur 3 tiers	27
II.3.5.4 Architecture Client/serveur n-tiers.....	28
II.3.6 Caractéristiques de l'architecture Client/Server	28
II.3.7 Avantages et inconvénients du modèle client/serveur	29
A. Avantages	29
B. Inconvénients.....	29
II.4 Les plates formes de développement d'application client/serveur	30
II.4.1 introduction.....	30
II.4.2 LES ARCHITECTURES DISTRIBUEES.....	30
II.4.3 Architecture J2EE	31
II.4.3.1 Définition.....	31
II.4.3.2 Fonctionnement interne	32
II.4.3.3 Architecture	32
II.4.3.4 Le serveur d'application.....	35
II.4.3.5 Outils de programmation.....	36

II.4.4 Architecture .NET.....	36
II.4.4.1 Définition.....	36
II.4.4.2 Fonctionnement interne.....	36
II.4.4.3 Architecture	36
II.4.4.4 L'environnement d'exécution.....	37
II.4.4.5 Serveur d'application	38
II.4.4.6 Outils de programmation.....	38
II.5 Conclusion	39
 Chapitre III : Analyse et Conception	
III.1 Introduction.....	40
III.2 Langage de modélisation UML.....	40
III.2.1 Définition	40
III.2.2 UML en œuvre	40
III.3 Analyse et conception.....	41
A. Analyse	41
A.1 Quelques définitions de bases	41
A.2 Spécification des exigences d'après les cas d'utilisation	42
A.2.1 Identification des acteurs	42
A.2.2 Identification des cas d'utilisation.....	43
A.2.3 Spécification des scénarios	44
A.2.4 Spécification des cas d'utilisations.....	49
A.2.5 Diagramme de cas d'utilisation générale.....	55
A.2.6 Diagramme de cas d'utilisation.....	56
B Conception	58
B.1 Elaboration des diagrammes de séquences	58
B.2 Elaboration des diagrammes de classe	63
B.3 Diagramme de classes.....	67
B.4 Le modèle physique des données	68
Conclusion.....	70

Chapitre IV : Réalisation

IV.1	Introduction.....	71
IV.2	Environnement et outils de développement.....	71
IV.2.1	Le Système d'exploitation Windows 7.....	71
IV.2.2	IDE NetBeans 6.8.....	71
IV.2.3	Serveur Web.....	73
IV.2.4	Serveur Web (Apache).....	74
IV.2.5	Le conteneur de Servlets Tomcat	74
IV.2.6	Présentation de MySQL.....	75
IV.5.	Le langage de programmation utilisé (Java)	76
IV.2.8	Le Servlets.....	77
IV.2.9	La Notion de JSP	79
IV.2.10	JavaScript	81
IV.3	Présentation de quelques interfaces et fonctionnalités de l'application	82
IV.3.1	L'interface Principale Page d'accueil.....	82
IV.3.2	La Page d'Authentification Administrateur.....	83
IV.3.3	L'interface Principale Administrateur	84
IV.3.4	La page d'ajout d'un nouveau type de papier	86
IV.3.5	La page Supprimer de papier	87
IV.3.6	La page Consultation de Stock.....	87
IV.3.7	La page Entrer de papier	88
IV.3.8	La page Historique d'Entrer de papier.....	90
IV.3.9	La page Historique Sortie de papier	91
IV.4	Conclusion	92

Conclusion générale	93
Bibliographie & Webliographie	94

Table des illustrations

1. Les figures

FIGURE I- 1 : LA SOCIETE DE DISTRIBUTION DE L'ELECTRICITE ET DU GAZ DU CENTRE(SDC).....	5
FIGURE I- 2 : ORGANIGRAMME GENERAL DE LA DIRECTION REGIONALE DISTRIBUTION	7
FIGURE I- 3 : ORGANIGRAMME DE LA DIVISION RELATIONS COMMERCIALES	10
FIGURE I- 4 : ORGANIGRAMME DE LA DIVISION ETUDE D'EXECUTION DES TRAVAUX D'ELECTRICITE ET DU GAZ	11
FIGURE I- 5 : ORGANIGRAMME DE LA DIVISION FINANCE ET COMPTABILITE.....	11
FIGURE I- 6 : ORGANIGRAMME DE LA DIVISION RESSOURCE HUMAINE	12
FIGURE I- 7 : ORGANIGRAMME DE LA DIVISION GESTION DES SYSTEMES INFORMATIQUES	13
FIGURE II- 1 : FONCTIONNEMENT DU CLIENT/SERVEUR.....	23
FIGURE II- 2 : ARCHITECTURE A 2 NIVEAUX.....	26
FIGURE II- 3 : ARCHITECTURE A 3 NIVEAUX	27
FIGURE II- 4 : ARCHITECTURE CLIENT/SERVEUR N-TIERS	28
FIGURE II- 5: EXEMPLE D'UNE ARCHITECTURE DISTRIBUEE	31
FIGURE II- 6 : EXEMPLE D'UNE REQUETE D'UN CLIENT LEGER J2EE.....	33
FIGURE II- 7 : ARCHITECTURE D'UNE SERVLET	34
FIGURE II- 8 : CONVERSION D'UNE JSP	35
FIGURE II- 9 : ARCHITECTURE GENERALE J2EE	36
FIGURE II- 10 : EXEMPLE D'UN CLIENT LEGER .NET	38
FIGURE II- 11 : ARCHITECTURE GENERALE .NET	39
FIGURE III- 1 : DEMARCHE DE MODELISATION	42
FIGURE III- 2 : DIAGRAMME DE CAS D'UTILISATION POUR L'UTILISATEUR.	56
FIGURE III- 3 : DIAGRAMME DE CAS D'UTILISATION POUR L'ADMINISTRATEUR	57
FIGURE III- 4 : DIAGRAMME DE SEQUENCE DU CAS D'UTILISATION « CREER UN COMPTE OU INSCRIPTION UTILISATEUR »	59
FIGURE III- 5 : DIAGRAMME DE SEQUENCE DU CAS D'UTILISATION « AUTHENTIFICATION»	60
FIGURE III- 6 : DIAGRAMME DE SEQUENCE DU CAS D'UTILISATION « GERER LE PAPIER »	61
FIGURE III- 7 : DIAGRAMME DE SEQUENCE DU CAS D'UTILISATION « GERER LES UTILISATEURS »	62
FIGURE III- 8 : DIAGRAMME DE CLASSE DU CAS D'UTILISATION « S'INSCRIRE UTILISATEUR».....	63
FIGURE III- 9 : DIAGRAMME DE CLASSE DU CAS D'UTILISATION « AUTHENTIFICATION UTILISATEUR »... ..	64
FIGURE III- 10 : DIAGRAMME DE CLASSE DU CAS D'UTILISATION « SUPPRIMER PAPIER ».....	64
FIGURE III- 11 : DIAGRAMME DE CLASSE DU CAS D'UTILISATION « GERER LES ENTREES/SORTIES PAPIER.....	65
FIGURE III- 12 : DIAGRAMME DE CLASSE DU CAS D'UTILISATION « GERER LES AGENCES/SERVICES ».....	66
FIGURE III- 13 : DIAGRAMME DE CLASSE GLOBALE	67
FIGURE IV- 1 : INTERFACE PRINCIPALE D'IDE NETBEANS 6.8.....	73
FIGURE IV- 2 : INTERFACE D'APACHE TOMCAT 7.0.34.....	74
FIGURE IV- 3 : INTERFACE PRINCIPALE MYSQL SERVEUR 5.4.....	76
FIGURE IV- 4 : PAGE D'ACCUEIL PRINCIPALE	82
FIGURE IV- 5 : PAGE D'AUTHENTIFICATION ADMINISTRATEUR.	83
FIGURE IV- 6 : L'INTERFACE GENERALE ADMINISTRATEUR.....	85
FIGURE IV- 7 : PAGE D'AJOUT D'UN NOUVEAU TYPE DE PAPIER	86
FIGURE IV- 8 : PAGE SUPPRIMER DE PAPIER.....	87
FIGURE IV- 9 : PAGE CONSULTATION DE STOCK	88
FIGURE IV- 10 : PAGE ENTREE DE PAPIER.....	89
FIGURE IV- 11 : PAGE HISTORIQUE PAPIER ENTREE	90
FIGURE IV- 12 : PAGE HISTORIQUE SORTIE DE PAPIER	91

2. Les tableaux

TABLEAU II- 1: LES COUCHES DE MODELE OSI	18
TABLEAU II- 2 : LES MODELE OSI ET TCP/IP	19
TABLEAU II- 3 : TABLEAU DE COMPARAISON ENTRE ARCHITECTURE CLIENT/SERVEUR 2 TIERS ET 3 TIERS	27
TABLEAU III- 1 : LES DIAGRAMMES D'UML	41
TABLEAU III- 2 : TABLEAU DE SPECIFICATION DES TACHES.	43
TABLEAU III- 3: TABLEAU SPECIFICATION DES SCENARIOS POUR L'UTILISATEUR.	46
TABLEAU III- 4: TABLEAU SPECIFICATION DES SCENARIOS POUR L'ADMINISTRATEUR.	48

Introduction générale

Tout au long de l'histoire de l'informatique, les technologies ont dessiné et fait évoluer l'organisation de nos sociétés. Aujourd'hui les nouvelles technologies, et notamment celles de l'information et de la communication (NTIC) ouvrent la voie à des modèles organisationnels d'entreprise plus adaptés.

Ceci laisse dire que quelle que soit la taille d'une entreprise grande ou petite, l'inclusion des nouvelles technologies informatiques est devenue indispensable et s'avère nécessaire pour son bon fonctionnement surtout dans le domaine de la gestion de l'information. Elles permettent d'améliorer, d'assurer une gestion plus fiable, plus rigoureuse, de minimiser le risque d'erreurs et d'assurer la disponibilité de l'information à toute éventuelle demande.

Comme toute entreprise se voulant concurrente sur le marché national et international, **SONELGAZ** « Société Nationale de l'Electricité et du GAZ » spécialisée dans le domaine de la fourniture des énergies électriques et gazières a pris soin d'améliorer les insuffisances imputées à leur système informatique traditionnel (hérité).

Pour ce faire, ce travail nous a été confié afin d'introduire des nouveautés au système existant et qui consistent en :

- Amélioration l'ancien système vers un nouveau système permettant la bonne gestion de papiers.

Pour la bonne organisation de notre travail, nous avons adopté la structure chapitrée suivante:

- ✚ Le premier chapitre, intitulé « Présentation de l'existant » présente l'organisme d'accueil, ainsi que la problématique.
- ✚ Le second chapitre, intitulé « Technologies web » donne un aperçu général sur les nouvelles technologies et de la communication (NTIC).
- ✚ Le troisième chapitre, intitulé « Analyse et conception » est consacré à la conception de l'application en utilisant le langage de modélisation UML.
- ✚ Le quatrième chapitre, intitulé « Réalisation » en détaillant principalement le fonctionnement de l'application et les outils utilisés pour son développement.

I. Organisme d'accueil

I.1 Définition et historique

La Société « **SONELGAZ** » est l'opérateur historique dans le domaine de la fourniture des énergies électrique et gazière en Algérie. Ses missions principales sont la production, le transport et la distribution de l'électricité ainsi que le transport et la distribution du gaz par canalisations.

Ses nouveaux statuts lui confèrent la possibilité d'intervenir dans d'autres segments d'activités présentant un intérêt pour l'entreprise et notamment dans le domaine de la commercialisation de l'électricité et du gaz à l'étranger.

« **SONELGAZ** » existe depuis plus d'un demi-siècle, voici un petit historique illustrant la création et le développement de la Société « **SONELGAZ** » :

- Ø **En 1947** : Suite au décret du 5 juin 1947 qui porte création de l'établissement public national « EGA » (Electricité et Gaz d'Algérie), chargé de la production, du transport et de la distribution de l'Electricité et du Gaz.
- Ø **En 1969** : Création de SONELGAZ, c'est l'ordonnance n° 69-59 du 28 juillet 1969 (parue au journal officiel n° 63 du 1^{er} août 1969) qui porte dissolution d'EGA «Electricité et Gaz d'Algérie » et création de la nouvelle Société Nationale de l'Electricité et du GAZ « **SONELGAZ** » Elle lui a notamment attribué le monopole de la production, du transport, de la distribution, de l'importation et de l'exportation de l'électricité et du gaz manufacturé. L'ensemble des biens de l'ex-EGA lui a été légué. En cette année, SONELGAZ était déjà une entreprise de taille importante dont le personnel est de quelque 6000 agents. Elle desservait déjà 700 000 clients.
- Ø **En 1991** : la SONELGAZ change de nature juridique et devient un Etablissement Public à Caractère Industriel et Commercial « EPIC » (décret exécutif n°91-475 du 14 Décembre 1991).
- Ø **En 1995** : Le décret exécutif n° 95-280 du 17 septembre 1995 confirme la nature de Sonelgaz en tant qu'Etablissement Public à caractère Industriel et Commercial.
 - § Sonelgaz est placée sous la tutelle du Ministre chargé de l'énergie
 - § Sonelgaz est dotée de la personnalité morale tout en jouissant de l'autonomie financière.
 - § Sonelgaz est gérée par les droits publics dans ces relations avec l'état.

Le présent décret définit les missions de Sonelgaz :

- ü Assurer la production, le transport et la distribution de l'énergie électrique de qualité.
- ü Assurer la distribution publique du Gaz, dans le respect des conditions de sécurité et au moindre coût.

Ø **En 1996** : Sonelgaz a engagé dès 1996 une réorganisation de l'entreprise dont la stratégie consiste à se centrer de plus en plus sur ses métiers de bases. C'est-à-dire la production, le transport et la distribution de l'électricité et Du Gaz, ce qui a donné naissance à un certain nombre de filiales dont elles sont à 100% actionnaires pour le moyen terme.

Le 1er janvier 1998, neuf filiales ont vu le jour. Il s'agit de:

- Une filiale en charge de la maintenance des équipements industriels : MEI
- Trois filiales en charge de la réparation des transformateurs TRANSFO (Centre, Est et Ouest)
- Une filiale en charge des travaux d'imprimerie : SAT Info
- Quatre filiales en charge de la maintenance et prestations véhicules : MPV (Alger, Constantine, Oran et Ouargla)

Ø **En 2011** : Les statuts de Sonelgaz, adoptés en 2002, ont été révisés et approuvés par le conseil des Ministres le Lundi 02 Mai 2011, et deviennent de ce fait en conformité avec le dispositif de la loi n°02/01 du 05 février 2002, relative à l'électricité et la distribution du Gaz par canalisations. Désormais, Sonelgaz –spa est organisée en "société Holding " sans création d'une personne morale nouvelle et prend la dénomination de Sonelgaz. Par ailleurs, la société holding et ses sociétés filiales forment un ensemble dénommé "Groupe Sonelgaz". Dans les statuts amendés, Sonelgaz conserve le rôle de détenteur du portefeuille d'actions constituant le capital social de ses filiales. Les conseils d'administration des filiales constituent les relais incontournables permettant à la société holding de suivre et d'orienter l'opération de pilotage des filiales.

Parmi les filiales métiers de base assurent la production, le transport et la distribution de l'électricité et de gaz on trouve la Société de Distribution de l'Electricité et du Gaz du Centre (SDC) dans la quelle la direction de distribution de Tizi-Ouzou fait parti.

I .2 Société de Distribution de l'Electricité et du Gaz du Centre (SDC)

La Société de Distribution de l'Electricité et du Gaz du Centre

(Par abréviation SDC -Spa) est une société par actions, filiale du groupe Sonelgaz. Elle a pour missions :

- l'exploitation et la maintenance du réseau de distribution de l'électricité et du gaz,
- le développement des réseaux électricité et gaz permettant le raccordement des nouveaux clients,
- la commercialisation de l'électricité et du gaz, dans les meilleures conditions de sécurité, de qualité de service au moindre coût.

La (SDC) est composée de treize (13) directions de distribution couvrant les territoires de douze (12) wilayas : Blida, Bouira, Médéa, Tizi Ouzou, Djelfa, Ouargla Urbain, Ouargla Rural, Biskra, El Oued, Laghouat, Ghardaïa, Illizi et Tamanrasset.

Créée en Janvier 2006, elle dispose d'un réseau de distribution d'une consistance de :

Pour l'électricité :

- **28 542 Km** en HTA
- **31 115 Km** en Basse Tension.

Pour le gaz

- **11 699 KM** en Moyenne pression.

Elle gère un portefeuille client à fin 2010, de :

Pour l'Electricité :

- **1 464 542** clients Basse Tension.
- **11 945** clients Moyenne Tension.
- **26** clients Haute Tension

Pour le Gaz :

- **571 644** clients Basse Pression.
- **771** clients Moyenne Pression.
- **36** clients Haute Pression.

Elle dispose d'un réseau commercial constitué de **65 agences commerciales**.

La SDC emploie un effectif global de **4 637 Agents**.

Elle a réalisé un Chiffre d'Affaires de **34.7 Milliards de DA en 2010**.

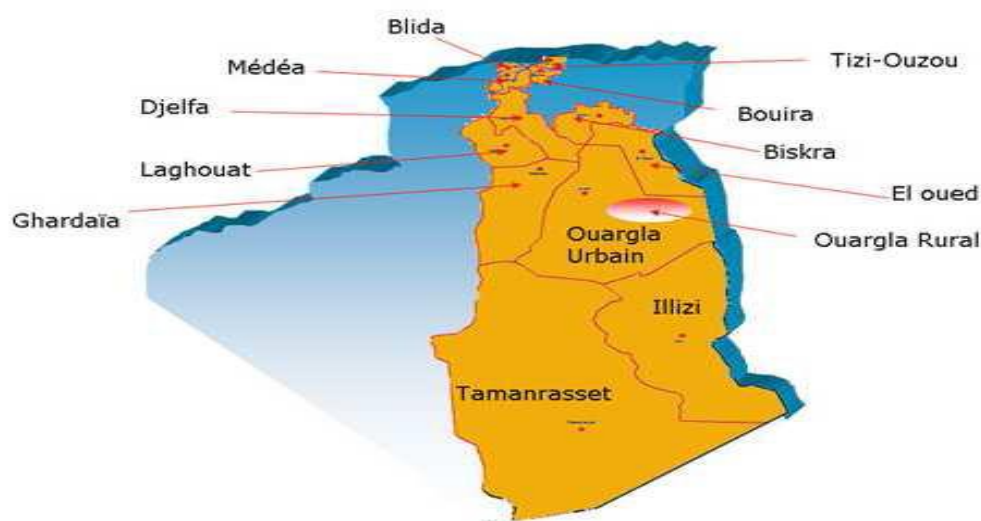


Figure I.1 : La Société de Distribution de l'Electricité et du Gaz du Centre(SDC)

I .3 La Direction de Distribution de Tizi-Ouzou

L'organisme d'accueil qui est la Direction de Distribution de Tizi-Ouzou dépend de la Société de Distribution Centre (SDC) filiale du groupe Sonelgaz.

Nous faisons la représentation de l'organisation générale de cette structure d'accueil comme suit :

I .3. 1 Missions de La Direction de Distribution

La Direction de Distribution de Tizi-Ouzou participe à l'élaboration de la politique de la direction générale de distribution (en matière présentation rendus aux Clients, développement des ventes, recouvrement des Créances...), elle est aussi chargée, dans les limites de ses attributions, de la distribution de l'énergie électrique et gazière et de la satisfaction des besoins de la clientèle aux conditions requises de coût, de qualité de service et de sécurité.

Plus spécifiquement elle accomplit les tâches suivantes :

- Ø La garantie de la qualité et la continuité de service.
- Ø Mettre en œuvre la politique commerciale de l'entreprise et en contrôler l'application.
- Ø Satisfaire dans les meilleures conditions de coûts et de délais aux demandes de raccordement des clients MT/BT (Moyenne Tension/Basse Tension), MP/BP (Moyenne Pression/Basse Pression) et leur apporter conseil et assistance.
- Ø Assurer la gestion (conduite, exploitation, et maintenance) et le développement des réseaux MT/BT et MP/BP et des installations annexes.

Chapitre I : Organisme d'accueil

- Ø Elaborer et maitre en œuvre le développement de la construction, la maintenance et l'exploitation des ouvrages.
- Ø Etablir les programmes travaux qui se rapportent à ces missions et en assurer la maitrise d'œuvre.
- Ø Assurer la gestion et le développement de la ressource humaine et des moyens matériels nécessaires au fonctionnement du centre.
- Ø L'exploitation et la maintenance des réseaux de distribution de l'électricité et du gaz.
- Ø Le développement des réseaux électricité et gaz permettant le raccordement des clients nouveaux.
- Ø Assurer la sécurité des personnels et des biens en rapport avec les activités de la distribution.
- Ø La garantie de l'équilibre entre l'offre et la demande en matière d'énergie.
- Ø La commercialisation de l'électricité et du gaz.
- Ø Assurer la représentation de Sonelgaz au niveau local.

I.3.2 Organigramme général de La Direction de Distribution

La direction de distribution de Tizi-Ouzou est composée de plusieurs subdivisions et l'organigramme suivant montre sa structure générale.

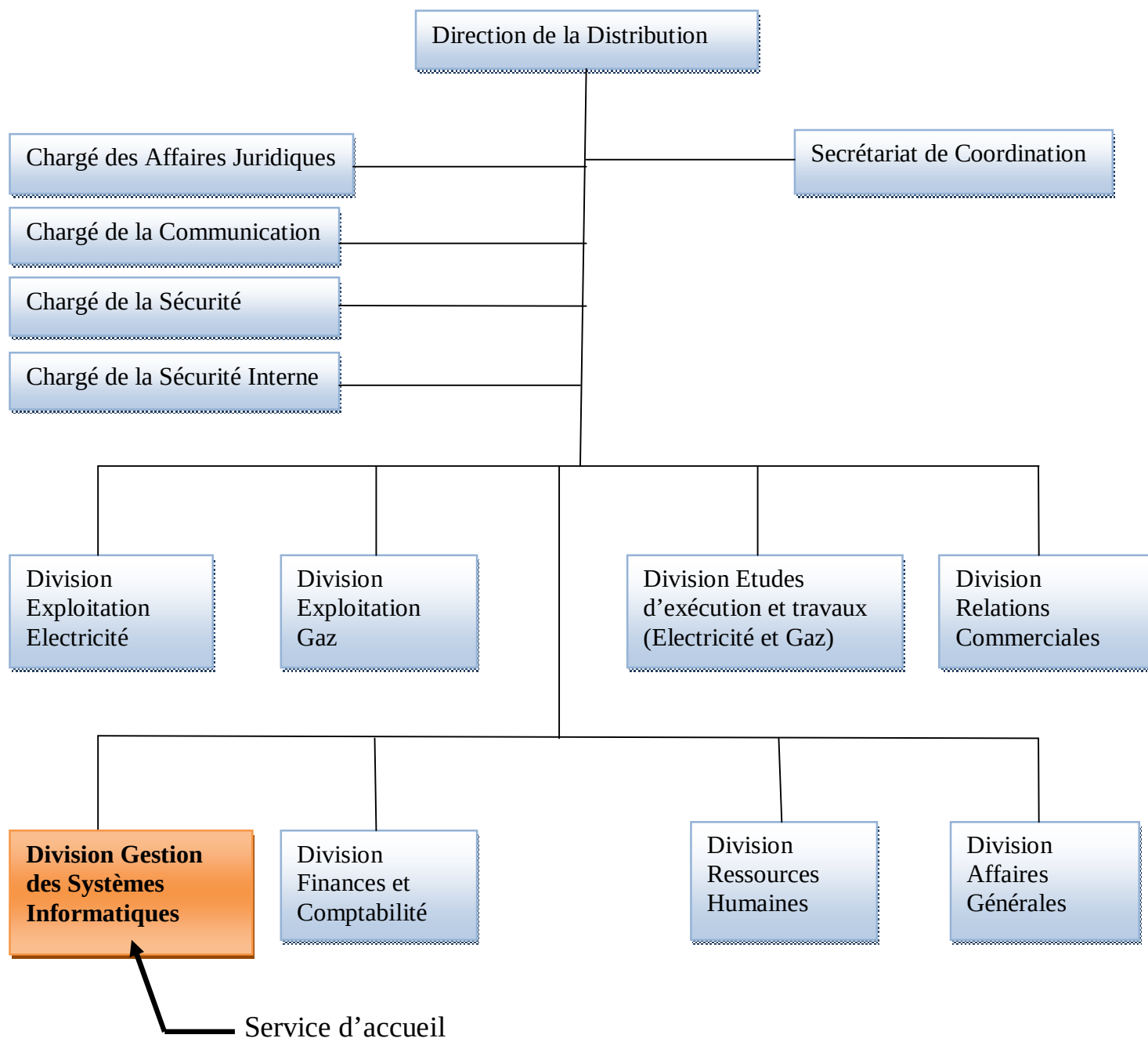


Figure I.2 : Organigramme général de la Direction Régionale Distribution.

I .3. 3 Mode d'Organisation de La Direction de Distribution

La direction de distribution est composée de :

a. Chargé de la Communication

Il a pour rôle de :

- Concevoir et organiser l'information destinée au public et à la clientèle en utilisant les supports appropriés (dépliants, affiches, presse, radio locales, brochures ...), en s'appuyant sur la politique arrêtées par la société.
- Participer avec la direction générale de la Société (SD) aux manifestations commerciales.
- Proposer des thèmes sur la publicité et l'information de la clientèle sur la base d'observations locales.
- Entretenir des relations étroites avec les medias (TV, Radios, Presse)

b. Chargé de la Sécurité

Ses tâches sont les suivantes :

- Suivre les accidents de travail, entreprises sous traitantes et tiers.
- Etablir les tableaux de bord et bilans de la direction de distribution.
- Mettre en œuvre des campagnes de sensibilisation des tiers sur les risques liés à la mauvaise utilisation de nos énergies.
- Participer pour le compte de la direction de distribution aux échanges dans le domaine de l'hygiène, la sécurité et la protection de l'environnement avec les organismes spécialisés locaux.
- Assurer le secrétariat de la CHS/Unité DD, et suivi des recommandations.
- Diffuser les affiches d'accidents typiques.
- Faire un planning des visites avec programmation des actions de sensibilisation.
- Visiter les chantiers (ouvrages neufs et installations existante).
- Préparer des simulations d'incident gaz et électricité avec les districts.
- Mettre en application toutes les directives et prescriptions concernant la sécurité.
- Diffuser les fiches d'accidents types.
- Veiller à la mise en œuvre des plans d'actions annuels.
- Participer aux prévisions du matériel de sécurité.

c. Chargé de Sécurité Interne

L'assistant du directeur de distribution, chargé de la sécurité interne d'établissement a pour mission d'assister le directeur de distribution dans la coordination des mesures et actions relatives à la sûreté interne des structures et établissements territorialement subordonnés.

L'assistant du directeur de distribution, chargé de la sûreté interne d'établissement est tenu de :

- Veiller à l'application de toutes les instructions et orientations émanant de la hiérarchie en matière de sûreté interne des établissements.
- S'assurer de la disponibilité technique et opérationnelle des moyens nécessaires à l'exercice de la sûreté interne au niveau des établissements.
- Elaborer et effectuer un programme d'inspection des établissements.
- Elaborer des rapports d'inspection en préconisant toutes les mesures qui s'imposent à l'amélioration de l'efficacité de la sûreté interne des établissements.
- Veiller à ce que toutes les mesures de sécurisation et de protection des ressortissants étrangers activant au profit de l'entreprise sur site ou lors de leurs déplacements soient prises. A ce titre, il veille à ce que les autorités locales en charge de la sécurité soient avisées conformément aux procédures établies au niveau du groupe.
- Assurer la représentation de la direction en matière de sûreté interne des établissements auprès des autorités locales.
- Rendre compte sans délais au directeur de distribution de toute anomalie ou fait saillant relatifs à la sûreté interne.
- Etablir des comptes rendus périodiques des activités de sûreté interne au directeur de distribution.

d. Chargé des Affaires Juridiques

Il a pour tâche de prendre en charge :

- les affaires d'ordre juridique de la direction régionale et les mesures permettant d'assurer le recouvrement des créances de toutes natures.
- Examiner et traiter les demandes d'indemnisation ainsi que le suivi de l'exécution des décisions de justice.

e. Division Relations Commerciales

Ce service est chargé de l'inspection et du contrôle. Son rôle consiste à élaborer des actions commerciales et le développement des ventes, apporter assistance aux clients. Il a aussi pour rôle de repérer les clients potentiels à travers les différents circuits d'informations, contrôler la relève, la facturation et le recouvrement de la clientèle.

Organigramme de Division Relations Commerciales

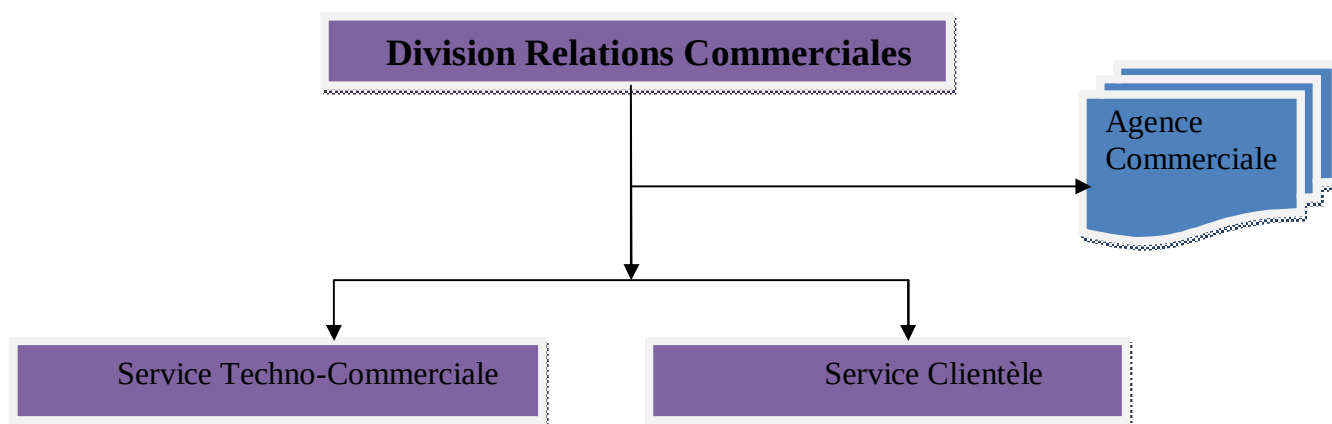


Figure I.3 : Organigramme de la Division Relations Commerciales.

f. Division étude d'exécution des travaux d'Electricité et du Gaz

Ce service est chargé de :

- L'étude et travaux d'Electricité et Gaz.
- La maîtrise d'œuvre.
- La gestion des investissements en matière de crédit et d'ordonnancement.

Organigramme de division étude d'exécution des travaux d'électricité et du Gaz

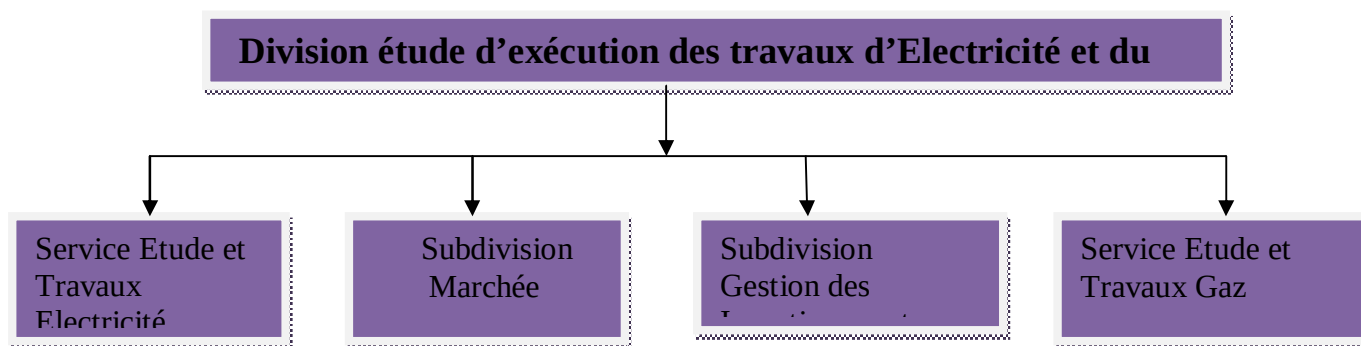


Figure I.4 : Organigramme de division étude d'exécution des travaux d'électricité et du Gaz.

g. Division Finance et Comptabilité

Cette division assure les règlements décentralisés, elle procède au suivi des rapprochements des comptes bancaires et CCP, d'élaboration des tableaux de bords et du budget annuel de la direction régionale et d'assurer les travaux de contrôle et de comptabilité de toutes les opérations de la direction.

Organigramme de division finance et comptabilité

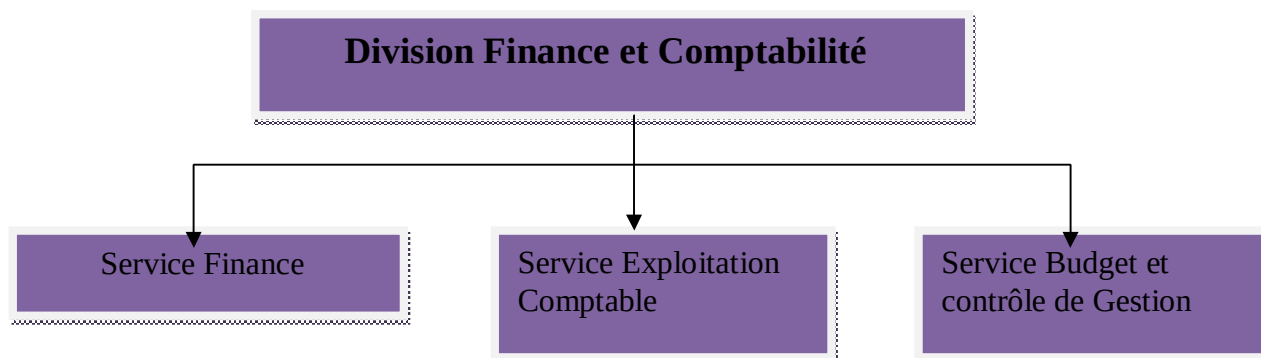


Figure I.5 : Organigramme de la Division Finance et Comptabilité.

h. Division Exploitation (Electricité ou Gaz)

Cette division est chargée de la supervision, conduite et la surveillance des installations de production et de l'élaboration des programmes d'essais, d'analyse et de contrôle des équipements et du matériels et d'amélioration de leurs performances, et elle est aussi chargée d'assurer la continuité de service et la disponibilité des moyens de production.

i. Division Ressource Humain

Ce service est chargé de gérer les agents et les apprentis, d'élaboration des prévisions budgétaires et des plans de formations et de carrières.

Il est aussi chargé de la préparation et de la réalisation des plans de recrutement de l'unité.

Organigramme de division ressource humain

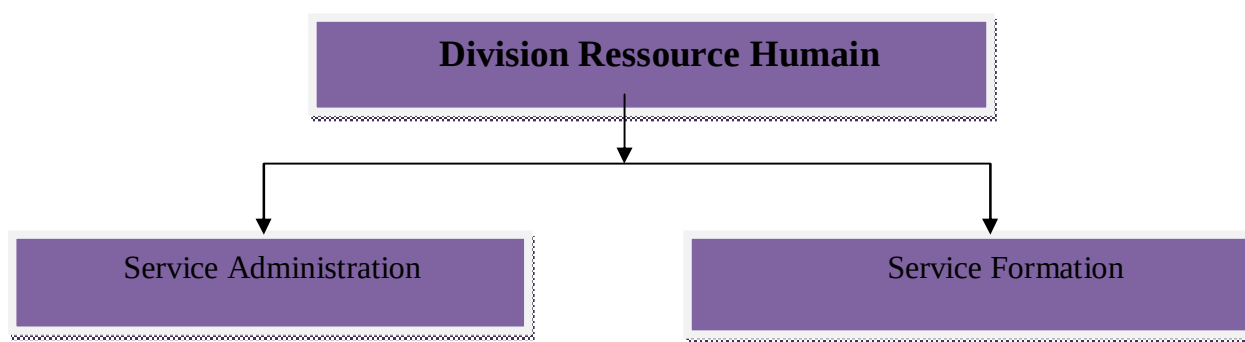


Figure I.6 : Organigramme de la division ressource humaine.

j. Divisions des Affaires Générales

Cette division est chargée de la gestion des moyens internes et des affaires générales de l'unité. Cette division assure les tâches suivantes :

- Les travaux de reproduction.
- La gestion immobilière.
- La gestion du parc véhicules de service de l'unité et le transport des agents.
- La gestion de courriers, archives et de la documentation.

k. Division Gestion des Systèmes Informatiques(DGSI)

Ce service est chargé d'accomplir les tâches suivantes :

- Gérer tout les matériels informatiques et périphériques affectés à la direction régionale.
- Gérer le centre du traitement informatique et la promotion des systèmes au niveau de la direction.
- Développer et améliorer les applications propres à la direction régionale.
- Approvisionner et contrôler la fourniture en consommables.
- Veiller et assurer la maintenance des systèmes.

Organigramme de division gestion des systèmes informatiques

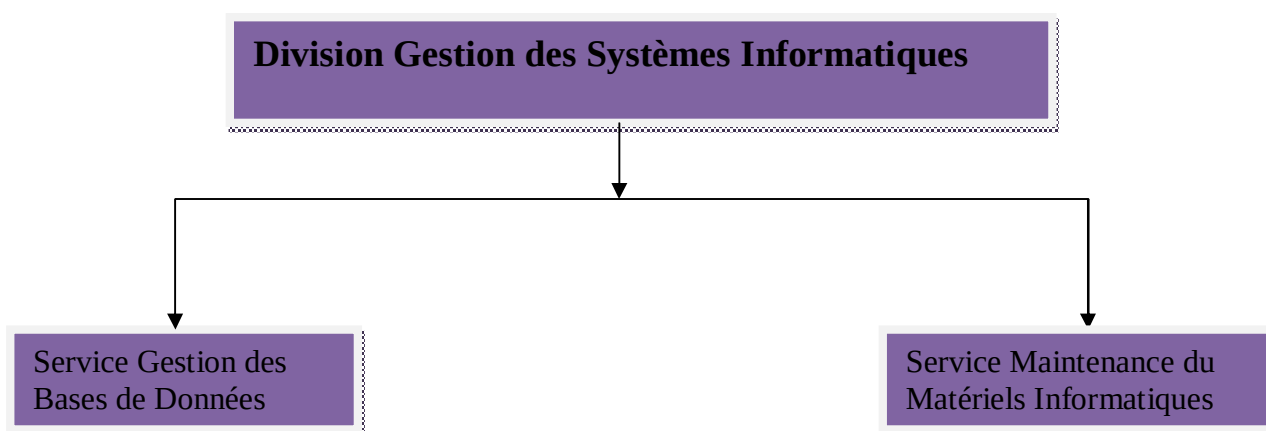


Figure I.7 : Organigramme de division gestion des systèmes informatiques.

Le travail réalisé dans le cadre de ce mémoire est destiné à la Division Gestion des Systèmes Informatiques(DGSI), consiste à la conception d'un système (informatique) permettant la gestion de papier. Pour cela, nous devons avoir une bonne connaissance du système existant, cela en faisant une analyse complète et détaillée de la situation actuelle, pour nous permettre de cerner les éventuelles carences, les défaillances et les failles du système et donc d'avoir une meilleure connaissance de l'envergure du problème étudié.

I .3. 4 Présentation du système informatique existant

La direction de distribution de Tizi-Ouzou est constituée de douze (12) agences répartie sur différents régions de wilaya, et d'autres services internes, pour lesquels fait la livraison des différents types de papiers reçus a la demande de celle-ci au niveau de la société de distribution de l'électricité et du gaz du centre (SDC). Ces agences comme (441 : Tizi-Ouzou, 442 : Draa Ben-Khedda... etc.), et services (DRH, DEG... etc.), sont connectés par des connexions VPN « Virtual Private Network » au serveur AIX (Advanced Interactive eXecutive) installé dans le service d'accueil, à préciser que toutes les applications (traitements, consultationsetc.) sont déployées dans ce serveur selon une architecture client/serveur.

Ces agences gèrent différents types d'abonnés :

a. Clients BT/BP :

- **BT/BP** : Basse tension (KW) / Basse pression

Il y'a deux types de clients BT/BP :

§ **Clients FSM** « Facture sur mémoire » : sont les clients ayant plusieurs lieux de consommation.

§ **Abonnés ordonnés** : ce sont d'abonnées ménages et abonnés non ménages.

b. Clients MT/MP :

- **MT/MP** : Moyenne tension / Moyenne pression ce sont les clients ayant des postes de transformation (GAZ, ELECTRECITE).

Dans le cas de rupture de stock, la DGSI établie une commande sur les différents types de papiers comme (BTBP1, MP1...etc.), une fois la livraison est effectué, le papier sera stocké, ce dernier sera livré aux différents services agences selon les besoins.

Au niveau de ce service (service d'accueil) on trouve une application qui fait la gestion de papiers mais celle-ci ne répond pas parfaitement aux exigences demandées.

I .3. 5 Problématique

L'exploitation du système existant pour la gestion de papier, nous amené à constater certains défaillances d'information au niveau de la base de données, lors des transactions (entrées, sortie) sur les différents types de papier, la mise à jour, la reformulation de papier non utilisé, le contrôle des commandes et des factures.

Quand la DGSI établie une commande sur un ou plusieurs type(s) de papier avec une telle quantité, à la livraison on constate par fois que la quantité livré n'accomplit pas la quantité demandé, et dans ce cas on doit garder une trace sur cette information, ainsi quand les agences liées à La Direction de Distribution de Tizi-Ouzou établirent des commandes pour un type de

papier, on se retrouve dans la même situation décrite au paravent, et pour remédier à ce problème, la DGSI sauvegarde les quantités manquantes manuellement, souvent elle trouve des failles pour gérer les bons de commandes et les factures.

- La quantité de tous les types de papiers se fait manuellement au niveau de la DGSI. D'où la difficulté de gérer le seuil minimum de papier, et c'est pour cela qu'elle se retrouve souvent dans des situations critiques.
- Au niveau de la gestion de reformulation la difficulté se trouve dans la mise à jour des types de papiers non utilisés, et les nouveaux types de papiers délivrés par le groupe SONELGAS.

I.3.6 Objectifs

L'objectif primordial de notre projet est de concevoir une application pour une bonne gestion automatique de papiers, des bons de commandes, des factures ainsi que la reformulation de différents types de papiers.

Plus précisément, l'application permettra de :

- Ø Automatiser toutes les procédures des entrées, sorties et réintégration et reformulation de papiers.
- Ø Mise à jour automatique de la fiche stock de papier.
- Ø Edition automatique des états mensuels suivants :
 - § Etat des consommations mensuelles.
 - § Récapitulation finale des consommations pour bien gérer les seuils de papiers.
- Ø Automatiser la procédure des commandes, factures et éviter les failles et les pertes d'informations.

I.4 Conclusion

Dans ce premier Chapitre, nous avons fait la présentation de l'organisme d'accueil, une analyse profonde de la DGSI nous a permis de cerner les défaillances trouvées au niveau de la gestion de papier (les commandes, les factures, la mise à jour des types de papiers). Pour faire une proposition d'une solution architecturale et logicielle plus adaptée aux nouvelles technologies de l'information et de communication, et faire une réalisation d'un système d'information qui fait gérer les problèmes constatés au sein de la DGSI, nous ferons d'abord la présentation de ces nouvelles technologies de communication dans le prochain chapitre.

II.1 Introduction

Toute entreprise utilise son propre système d'information pour son fonctionnement, et fait la communication avec le monde externe grâce à des flux d'information circulant dans l'organisation associé aux moyens mis en œuvre pour les gérer.

Pour traiter l'information, les humains mettent en œuvre des règles et des procédures comme Les « Nouvelles Technologies de l'Information et de la Communication ».

Le NTIC représentent l'ensemble des technologies informatiques qui contribuent à une véritable révolution socioculturelle, surtout leurs applications dans le domaine économique.

L'échange d'information entre ordinateurs et la liaison entre ces machines (réseau informatique) est en plein d'évolution, le modèle OSI et le modèle TCP/IP sont deux modèles très utilisés dans les réseaux informatiques et beaucoup d'outils qui mise en œuvre dans l'utilisation des réseaux. En plus la programmation par objet est vue davantage comme un paradigme, un ensemble d'instruments conceptuels, plutôt que comme une simple technique de programmation. Les plates formes J2EE et .NET ont des avantages, en particulier une bonne portabilité et une maintenance du code. De plus, J2EE et .NET reposent sur des composants distincts, interchangeables et distribués, ce qui signifie notamment qu'un système reposant sur J2EE ou .NET peut posséder des mécanismes de haute disponibilité, afin de garantir une bonne qualité de service et que la maintenance des applications soit très facile.

II.2 Quelques concepts sur les réseaux informatiques

II.2.1 Définition d'un réseau

Un réseau est un ensemble de matériels et de logiciels permettant de connecter deux ou plusieurs ordinateurs géographiquement dispersés afin d'échanger des données. Les données circulent dans le réseau sous forme de flux de bits. Ces bits sont transférés sous forme de paquets de données qui se déplacent rapidement et qui traversent plusieurs réseaux pour arriver à la destination. Quelque soit la durée de déplacement, l'information doit circuler d'une manière fiable et atteindre sans erreurs la destination voulue.

II.2.2 Caractéristique d'un réseau

Les réseaux locaux sont des infrastructures complexes et pas seulement des câbles entre stations de travail. Si l'on énumère la liste des composants d'un réseau local, on sera surpris d'en trouver une quantité plus grande que prévue:

1. Le **câblage** constitue l'infrastructure physique, avec le choix entre paires téléphoniques, câble coaxial ou fibre optique. Il détermine le type de concentrateurs (switch, HUB, point d'accès Wifi, ...etc.) utilisé. Ces équipements constituent les nœuds dans le cas de réseaux en étoile.
2. La **méthode d'accès** décrit la façon dont le réseau arbitre les communications des différentes stations sur le câble: ordre, temps de parole, organisation des messages. Elle dépend étroitement de la topologie et donc de l'organisation spatiale des stations les unes par rapport aux autres. La méthode d'accès est essentiellement matérialisée dans les cartes d'interfaces, qui connectent les stations au câble.
3. Les **protocoles** de réseaux sont des logiciels qui "tournent" à la fois sur les différentes stations et leurs cartes d'interfaces réseaux. C'est le langage de communication. Pour que deux structures connectées sur le réseau, ils doivent "parler" le même protocole.
4. Le **système d'exploitation** du serveur réseau, souvent nommé gestionnaire du réseau, est installé sur le ou les serveurs. Il gère les partages, droits d'accès, ...etc. Pour Microsoft, on retrouve Windows NT serveur, Windows 2000 serveur, Windows 2003, 2008. Ce sont des versions spécifiques. Linux est utilisé sous différentes versions serveurs. Novell Netware est un système dédié principalement efficace comme serveur de fichier.
5. Le **système de sauvegarde** est un élément indispensable qui fonctionne de diverses manières soit en recopiant systématiquement tous les fichiers du ou des serveurs, soit en faisant des sauvegardes régulières, éventuellement automatisées.
6. Un **pont**, un **routeur** ou **passerelle** constituent les moyens de communication qui permettent à un de ses utilisateurs de "sortir" du réseau local pour atteindre d'autres réseaux locaux ou des serveurs distants, Internet ou autres.
7. Le **système de gestion et d'administration** du réseau envoie les alarmes en cas d'incidents, comptabilise le trafic, mémorise l'activité du réseau et aide le superviseur à prévoir l'évolution de son réseau. Cette partie est typiquement software.

II.2.3 Le modèle OSI

Le modèle OSI (Open System Interconnection Model) défini en 1977 régit la communication entre 2 systèmes informatiques selon 7 niveaux. A chaque niveau, les deux systèmes doivent communiquer "compatibles". En matériel réseau, nous n'utilisons que les couches inférieures, jusqu'au niveau 3. Ces niveaux sont également appelés couches.

L'OSI est un modèle de base normalisé par l'International Standard Organisation (ISO).

Chapitre II : Développement d'application

Ce tableau montre les différentes couches de modèle OSI :

Application	↓	Couche Application	7	Couche Application	↑	
	↓	Couche Présentation	6	Couche Présentation	↑	
	↓	Couche Session	5	Couche Session	↑	
Transport des données	↓	Couche Transport	4	Couche Transport	↑	
	↓	Couche Réseau(Network)	3	Couche Réseau(Network)	↑	Paquet
	↓	Couche liaison de données (Data Link)	2	Couche liaison de données (Data Link)	↑	Trames
	→	Physique(Physical)	1	Couche Physique(Physical)	→	BIT
		Support de communication				

Tableau II.1 : les couches de modèle OSI.

- **Niveau 7 (application):** gère le format des données entre logiciels.
- **Niveau 6 (présentation):** met les données en forme, éventuellement de l'encryptage et de la compression, par exemple mise en forme des textes, images et vidéo.
- **Niveau 5 (session):** gère l'établissement, la gestion et coordination des communications
- **Niveau 4 (transport):** s'occupe de la gestion des erreurs, notamment avec les protocoles UDP et TCP/IP
- **Niveau 3 (réseau):** sélectionne les routes de transport (routage) et s'occupe du traitement et du transfert des messages: gère par exemple les protocoles IP (adresse et le masque de sous-réseau) et ICMP. Utilisé par les routeurs et les switches manageables.
- **Niveau 2 (liaison de données):** utilise les **adresses MAC**. Le message Ethernet à ce stade est la trame, il est constitué d'un en-tête et des informations. L'en-tête reprend l'adresse MAC de départ, celle d'arrivée + une indication du protocole supérieur.
- **Niveau 1 (physique):** gère les connections matérielles et la transmission, définit la façon dont les données sont converties en signaux numériques: ça peut-être un câble coaxial, paires sur RJ45, onde radio, fibre optique, ...etc.

II.2.4 Le modèle TCP/IP

Le modèle TCP/IP s'inspire du modèle OSI auquel il reprend l'approche modulaire mais réduit le nombre à quatre. Les trois couches supérieures du modèle OSI sont souvent utilisées par une même application. Ce n'est pas le cas du modèle TCP/IP. C'est actuellement le modèle théorique le plus utilisé.

Le tableau suivant montre les deux modèles OSI et TCP/IP :

Protocoles utilisés	Modèle TCP/IP	correspondance en OSI
	Couche application	Application
		Présentation
		Session
TCP / UDP, gestion des erreurs	Couche Transport	Transport
IP / ARP et RARP /ICMP / IGMP	Couche Internet	Réseau
	Couche Accès réseau	Liaison de donnée
		Physique

Tableau II.2 : les modèle OSI et TCP/IP.

De nouveau, on ajoute à chaque niveau un entête, les dénominations des paquets de données changent chaque fois:

- 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 message est donc découpé en morceau avant envoi pour respecter une taille maximum suivant le MTU.
- Le segment une fois encapsulé dans la couche Internet prend le nom de **datagramme**
- Enfin, on parle de **trame** envoyée sur le réseau au niveau de la couche accès réseau

Les **couches du modèle TCP/IP** sont plus générales que celles du modèle OSI.

A. Couche application

La **Couche Application** reprend les applications standards en réseau informatique et Internet:

- **SMTP**: "Simple Mail Transport Protocol" gère le transfert de mails entre serveurs
- **POP**: gère le transfert des mails entre un serveur de messagerie et un ordinateur client
- **TELNET**: connexion sur une machine distante (serveur) en tant qu'utilisateur
- **FTP** (File Transfert Protocol), transfert des fichiers via Internet et beaucoup d'autres.

B. Couche transport

La **Couche transport** permet le transfert des données et les contrôles qui permettent de vérifier l'état de la transmission.

Les protocoles des couches suivantes permettent d'envoyer des données issues de la couche application. On ne définit pas réellement les logiciels qui communiquent, mais des numéros de **ports** associés au type d'application (numéro variant de 0 à 65535, 2^{16}). Par exemple, la navigation Internet utilise le port TCP 80, l'http, le 443, le FTP utilise le 21, ...etc.

La couche transport gère 2 protocoles de transport des informations, indépendamment du type de réseau utilisé:

- **TCP** est orienté connexion (il vérifie la bonne transmission de données par des signaux d'accusés de réception –a cknowledge - du destinataire), il assure ainsi le contrôle des données
- **UDP**, archaïque et non orienté connexion, n'assure aucun contrôle de transmission des données, par exemple utilisé en streaming.

Ces 2 types (orienté connexion ou pas) sont une notion utilisée pour les firewalls. Si vous fermez un port en TCP, l'envoi d'un message ne renvoie pas de signal de retour (a cknowledge), faisant croire que l'adresse IP est libre, non utilisée. En UDP au contraire, un port fermé ne renvoie pas d'informations, faisant croire à une adresse IP utilisée. Le protocole UDP renvoie uniquement un message si le port est en erreur (ne répond pas)

C. Couche INTERNET

La **couche INTERNET** est chargée de fournir le paquet des données. Elle définit les datagrammes et gère la décomposition / recomposition des segments.

La couche Internet utilise 5 protocoles, seuls les 3 premiers sont importants):

1. Le **protocole IP**: gère les destinations des messages, adresse du destinataire
2. Le **protocole ARP** (Adresse Resolution Protocol): gère les adresses des cartes réseaux et la correspondance avec l'adresse IP. Chaque carte a sa propre adresse MAC d'identification codée sur 48 bits.
3. Le **protocole ICMP** (Internet Control Message Protocol) gère les informations relatives aux erreurs de transmission. ICMP ne les corrige pas, il signale uniquement que le message contient des erreurs, utilisé par exemple par la commande DOS Ping.
4. Le **protocole RARP** (Reverse Address Resolution Protocol) gère l'adresse IP pour les équipements réseaux qui ne peuvent en récupérer une automatiquement par lecture

d'information dans un fichier de configuration ou via un serveur DHCP. Lorsqu'un équipement réseau démarre, le gestionnaire réseau lit l'adresse IP à utiliser, ce qui est impossible pour certains équipements qui ne possèdent pas de disques durs (principalement les terminaux)

5. Le protocole **IGMP** (Internet Group Management Protocol) permet d'envoyer le même message à des ordinateurs qui font partie d'un groupe. Il permet aussi à ces machines de s'abonner et se désabonner d'un groupe. La principale application HARDWARE de l'IGMP se retrouve dans les SWITCH manageables. Ce protocole permet de regrouper des stations.

D. Couche Accès réseau

La **couche Accès réseau** spécifie la forme sous laquelle les données doivent être transmises. Elle prend en charge les notions suivantes:

- type de réseaux (Ethernet, Token Ring, ...), y compris les cartes réseaux
- transfert des données
- Synchronisation de la transmission de données
- Mise en forme (format) des données
- Conversion analogique/numérique pour les modems téléphoniques
- Contrôle des erreurs

II.2.5 Sécurité et administration

Un des aspects important d'un réseau informatique local est la centralisation de l'administration des données. Ceci permet de sauvegarder et sécuriser les données sur une seule machine. La sécurité reprend un ensemble de mesures contre les intrusions et virus, la gestion des privilèges et droits d'accès, la sauvegarde quotidienne des données, des équipements redondants en cas de panne, ...

Il n'y a pas de solutions idéales pour la sécurité des réseaux (et pour la sécurité informatique en générale). Trois solutions sont envisageables: les solutions matérielles que nous verrons, des solutions basées sur Linux et des solutions basées sur Windows ou des programmes rajoutés sur ces stations Windows. Le mélange de plusieurs solutions est possible dans certains cas. Certaines solutions sont d'ailleurs complémentaires. Sur un gros réseau "sensible", mettre un VPN hardware n'est pas suffisant. Une sécurité logicielle complémentaire incluant des contrôles d'accès au niveau administration serveur (serveur,

dossier, droits d'accès) et logiciels de sécurités vérifiant le trafic sur le réseau interne n'est pas superflu.

- Les **routeurs** peuvent être remplacés par une solution basée sur un serveur 2003 ou 2008, par un logiciel proxy comme WinGate ou par un ordinateur configuré spécifiquement en Linux
- Un **serveur proxy** est parfois intégré dans les routeurs (mais généralement sous Windows ou Linux)
- Les **firewalls** sont intégrés dans certains routeurs mais des logiciels assurent (presque) des fonctions équivalentes, souvent intégrés dans l'anti-virus (ex.:Symantec, ZoneAlarm, Mcafee au niveau des stations)
- Les **réseaux privés intégrés (VPN)** sont intégrés dans certains systèmes d'exploitation serveurs mais peuvent également être des équipements spécifiques.
- Les **anti-virus** sont le plus souvent des logiciels, mais peuvent être implantés dans des routeurs qui vérifient tout le trafic extérieur.

II.3 L'architecture client/serveur

II.3.1 Définition

De nombreuses applications fonctionnent selon un environnement Client/serveur, cela signifie que des machines clientes (des machines faisant partie du réseau) contactent un serveur, une machine généralement très puissante en termes de capacités d'entrée-sortie, qui leur fournit des services. Ces services sont des programmes fournissant des données telles que des fichiers, une connexion, etc. Les services sont exploités par des programmes, appelés programmes clients, s'exécutant sur les machines clientes. On parle ainsi de client lorsque l'on désigne un programme tournant sur une machine cliente, capable de traiter des informations qu'il récupère auprès d'un serveur.

II.3.2 Notions de bases

- ü **Client** : C'est le processus demandant l'exécution d'une opération à un autre processus par envoi d'un message contenant le descriptif de l'opération à exécuter et attendant la réponse à cette opération par un message en retour.
- ü **Serveur** : C'est un processus accomplissant une opération sur demande d'un client.
- ü **Requête** : C'est un message transmis par un client à un serveur décrivant l'opération à exécuter pour le compte d'un client.

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

ü **Middleware** : C'est le logiciel qui est au milieu assure les dialogues entre les clients et les serveurs souvent hétérogènes.

II.3.3 Fonctionnement d'un système client/serveur

Un système client/serveur fonctionne selon le schéma suivant :

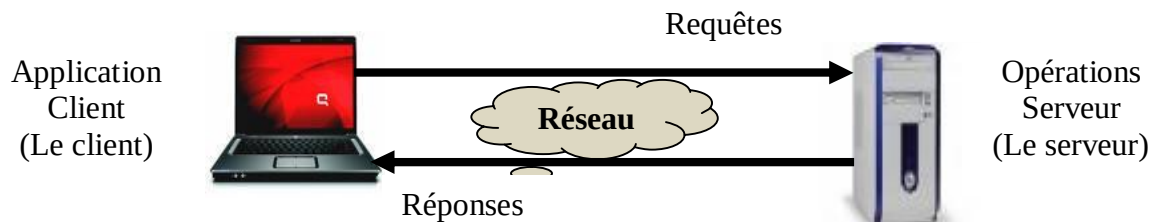


Figure II.1 : fonctionnement du client/serveur.

- Le client émet une requête vers le serveur grâce à son adresse IP et le port, qui désigne un service particulier du serveur.
- Le serveur reçoit la demande et répond à l'aide de l'adresse de la machine cliente et son port.

II.3.4 Les types de serveurs pour le système client/serveur

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

a) **Serveur de bases de données**

Dans le cas d'un serveur de bases 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.

b) Serveur d'objet

Dans ce type de serveur, l'application Client/serveur est écrite sous forme d'un jeu d'objets communicants. Les objets clients communiquent avec les objets serveurs au moyen d'un courtier d'objet 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 envoie les résultats à l'objet client. Les architectures résultantes sont appelées CORBA.

c) Serveur de transaction

Dans ce modèle le client invoque des procédures distantes résidant sur le serveur qui comporte un moteur de base de données SQL. Ces procédures distantes exécutent un ensemble d'instructions SQL. L'échange sur le réseau consiste en un seul message requête/réponse (contrairement à l'application serveur base de données pour laquelle le message requête/réponse est émis pour chaque instruction SQL). Pour ce type de serveurs l'application client/serveur nécessite du code source au niveau du serveur.

d) Le serveur web

Le serveur web est un processus capable de traiter des requêtes http issus des clients web; c'est-à-dire des navigateurs. Ce processus est présent sur un serveur physique accessible par les ordinateurs du réseau via TCP/IP. Ce serveur contient les informations qui peuvent être mise à disposition des utilisateurs sous forme de fichiers HTML localisés dans un ou plusieurs périphériques de stockage, généralement des disques durs.

e) Les serveurs de groupware

Ce type de serveur s'intéresse à la gestion d'informations semi-structurées. Il est fondé sur trois technologies de base : la gestion de documents multimédia, l'ordonnancement des tâches et le courrier électronique.

f) Un serveur de fichiers

Permet de partager des données à travers un réseau. Le terme désigne souvent l'ordinateur (serveur) hébergeant le service applicatif. Il possède généralement une grande quantité d'espace disque où sont déposés des fichiers. Les utilisateurs peuvent ensuite les récupérer au moyen d'un protocole de partage de fichier.

On utilise généralement l'un des trois protocoles suivant :

- Ø **FTP** : protocole de transfert de fichiers, c'est un protocole de communication dédié à l'échange de fichiers informatique.
- Ø **CIFS** : anciennement nommé SMB, c'est un protocole permettant le partage de ressources (fichiers et imprimantes) sur des réseaux locaux.

Ø **NFS** : Network File System, est un protocole développé par Sun Microsystems qui permet à un ordinateur d'accéder à des fichiers via un réseau. le choix du protocole dépend principalement de la méthode d'accès des utilisateurs.

CIFS est utilisé par les systèmes d'exploitation Microsoft Windows, **NFS** est répandu dans le milieu UNIX.

Toutefois des implémentations de ces protocoles sont disponibles pour tout type de système.

Ces deux protocoles permettant d'établir des liaisons permanentes entre le client et le serveur.

FTP est utilisé pour des connexions ponctuelles lorsque le client n'a pas besoin d'être connecté en permanence au serveur de fichier.

II.3.5 Types d'architectures Client – Serveur

II.3.5.1 Architecture à 2 niveaux

L'architecture à deux niveaux (aussi appelée architecture 2-tiers, tiers 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.

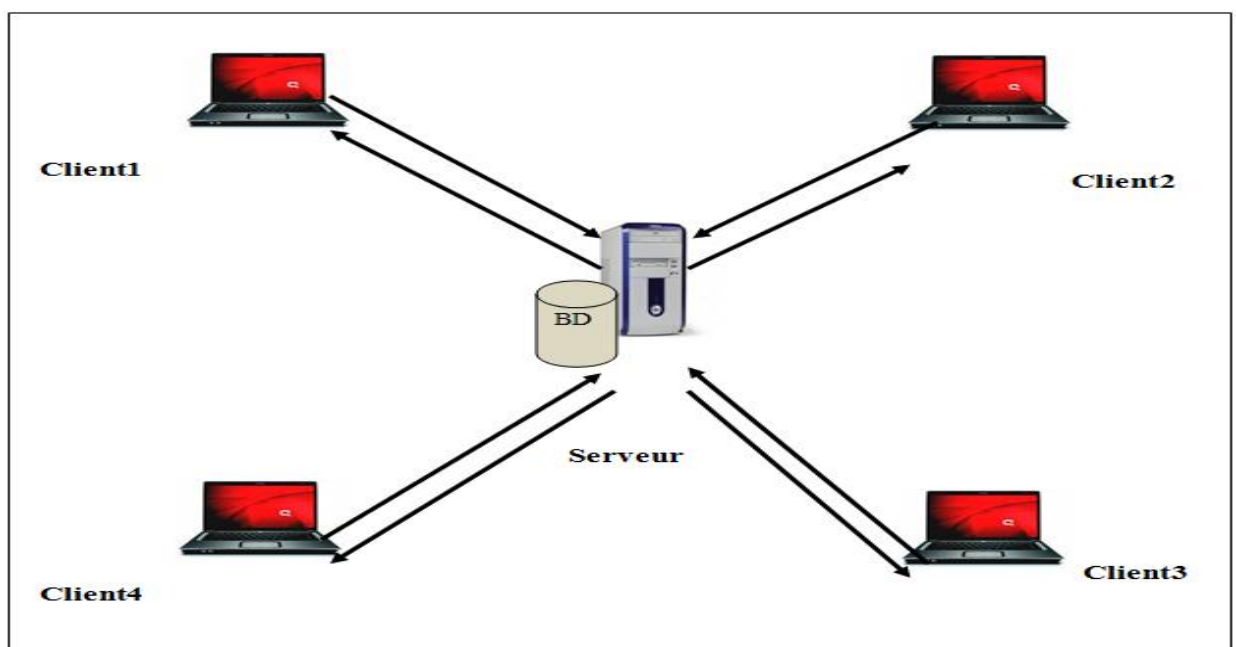


Figure II.2: Architecture à 2 niveaux

II.3.5.2 Architecture à 3 niveaux

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

- ü **Un client**, c'est-à-dire l'ordinateur demandeur de ressources, équipée d'une interface utilisateur (généralement un navigateur web) chargée de la présentation;
- ü **Le serveur d'application** (appelé également middleware), chargé de fournir la ressource mais faisant appel à un autre serveur ;
- ü **Le serveur de données**, fournissant au serveur d'application les données dont il a besoin.

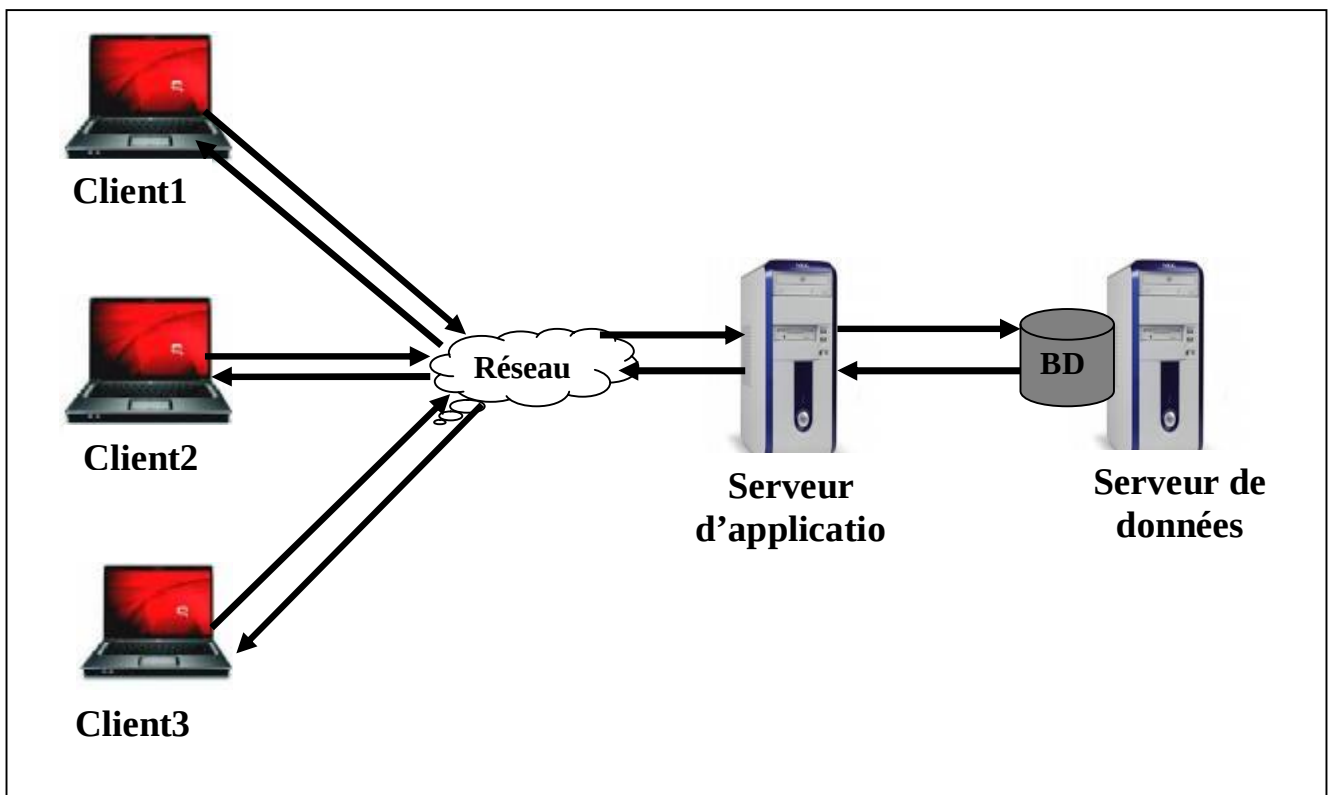


Figure II.3: Architecture à 3 niveaux.

Chapitre II : Développement d'application

Voici un tableau qui illustre en détail la différence entre une architecture 2 tiers et une architecture 3 tiers :

II.3.5.3 Comparaison entre une architecture client/serveur 2 tiers et 3 tiers

	Architecture 2 tiers	Architecture 3 tiers
Administration du système	Complexe	Moins complexe
Sécurité	Faible : sécurité au niveau des données	Importante : au niveau du service, méthode ou type d'objet.
Encapsulation des données	Faible : les tables de la base de données sont accessibles.	Importante : le client invoque des services ou méthode.
Performance	Faible : de nombreuses instructions SQL transite sur le réseau	Bonne : seules les requêtes et réponse sont échange
Réutilisation	Faible : le client est monolithique	Important : réutilisation possible
Facilité de développement	Importante	En voie d'amélioration vu l'utilisation d'outils
Support de base de données hétérogènes	Non	Oui : utilisation de plusieurs bases de données
Support de l'internet	Faible : limitation de la bande passante	Excellente : utilise de petits clients facilement téléchargés (Beans....)
Choix de communication	Non : seule les appels synchrones orientés connexion	Oui : support d'appels de type RPC, file d'attente,etc.
Souplesse de l'architecture matérielle	Limitée.	Excellente : le second niveau peut être repartitionné sur plusieurs serveurs, le deuxième et le troisième sur la même machine.
Disponibilité	Faible.	Excellente : relance possible du niveau intermédiaire sur d'autres serveurs.

Tableau II.3 : Tableau de comparaison entre architecture client/serveur 2 tiers et 3 tiers.

II.3.5.4 Architecture Client/serveur n-tiers

L'architecture à n-tiers a été pensée pour pallier aux limitations des architectures 3 tiers et concevoir des applications puissantes et simples à maintenir. Ce type d'architecture permet de distribuer plus librement la logique applicative, ce qui facilite la répartition de la charge entre tous les niveaux. Cette architecture est illustrée dans la figure suivante :

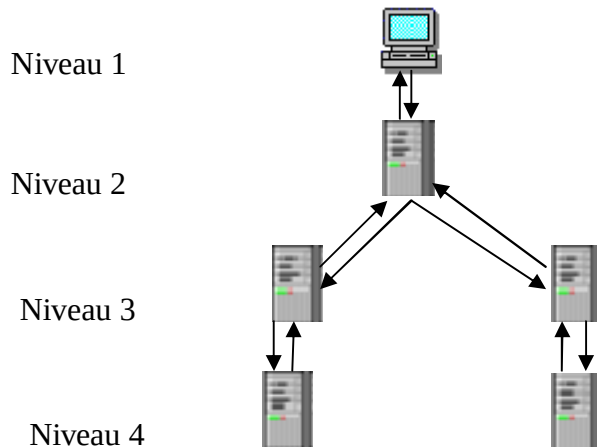


Figure II.4: architecture client/serveur n-tiers.

II.3.6 Caractéristiques de l'architecture client/serveur

L'architecture client/serveur est caractérisée par :

- ▼ **Service** : le modèle Client/serveur est essentiellement une relation entre des processus. Le processus serveur est un fournisseur de services et le client en est le consommateur, le modèle client/serveur établit ainsi une séparation claire des rôles à partir de la notion du service.
- ▼ **Partage de ressources** : un serveur est censé savoir traiter plusieurs demandes clients à la fois et leurs accès aux ressources.
- ▼ **Asymétrie des protocoles** : la relation entre le client et le serveur est de type plusieurs vers un, toutefois, le client est le déclencheur du dialogue en demandant un service alors que le serveur attend passivement les requêtes.
- ▼ **Assemblage multi-vendeur** : le client/serveur est indépendant de la plate forme matérielle ou de système d'exploitation. On doit toujours pouvoir mélanger et apparier les plates formes client/serveur.

- v Echange de messages :** le client et le serveur sont des systèmes à la liaison épisodique qui interagissent au moyen de messages. Le message est un mécanisme d'émission de demandes de services et de réponses à celles-ci.
- v Encapsulation des services :** le serveur est un spécialiste, un message lui indique quel service est requis et c'est à lui de décider comment rendre ce service. Les serveurs peuvent être mis à niveau sans effet sur les clients tant que l'interface des messages reste la même.
- v Intégrité :** le code et les données du serveur sont gérés de façon centralisée, ce qui garantit un moindre coût de maintenance et une meilleure intégrité des données tandis que les clients restent individuels et indépendant.
- v Souplesse et adaptabilité :** on peut modifier le module serveur sans toucher au module client et vice versa. Si une station est remplacée par un modèle plus récent, on modifie le module client sans modifier le module serveur.

II.3.7 Avantages et inconvénients du modèle client/serveur

a. Avantages

Les principaux avantages de ce modèle sont :

- ü Des ressources centralisées : Etant donné que le serveur est au centre du réseau, il peut gérer des ressources communes à tous les utilisateurs, comme par exemple une base de données centralisée afin d'éviter les problèmes de redondance et d'incohérence.
- ü Une meilleure sécurité : Car le nombre de points d'accès aux données est moins important.
- ü Une administration au niveau serveur : Les clients ont moins d'importance dans ce modèle, ils ont besoin d'être administrés.
- ü Un réseau évolutif : Grâce à cette architecture, on peut supprimer ou ajouter des clients sans perturber le fonctionnement du réseau et sans modifications majeures.

b. Inconvénients

L'architecture client/serveur a tout de même quelques inconvénients parmi lesquels :

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

II.4 Les plates formes de développement d'application client/serveur

II.4.1 Introduction

Les technologies J2EE et .NET sont des plates formes de développement OO (Orientés Objet) utilisés pour la création d'applications commerciales multi-niveaux.

Le but de tout langage de programmation consiste à fournir aux programmeurs des instruments d'abstraction pour exprimer, de façon le plus efficace possible, le fonctionnement d'un programme.

Les langages OO (orientés objet) ont permis l'introduction de certains concepts d'abstraction plus puissants par rapport à ceux qui étaient disponibles auparavant : c'est à dire les concepts de Classe et Objet.

Classe: Structure de données formées par un ensemble de variables et opérations (appelées méthodes) qui peuvent être exécutées sur ces données

Objet: Élément d'une classe qui est créé dans un programme pour être en suite utilisé à travers les méthodes que la classe définit. On dit qu'un objet est une instance d'une classe.

Aujourd'hui, la programmation par objet est vue davantage comme un paradigme, un ensemble d'instruments conceptuels, plutôt que comme une simple technique de programmation. Les langages Java et .NET font partie de la famille de langages OO bien que plusieurs différences existent au niveau technique et de mise en œuvre.

II.4.2 Les Architectures Distribuées

Une application distribuée est une application (typiquement de grandes dimensions) pour laquelle les différents composants ont été distribués sur différents niveaux de l'environnement d'élaboration du réseau par rapport aux fonctions qu'ils offrent dans l'architecture globale. Ces applications sont complexes et sont constituées de nombreux programmes logiciels qui sont exécutées sur différents supports matériels. Les « niveaux » (tiers) d'une application distribuée sont les niveaux entre sa propre application et les données qu'elle gère. Ces niveaux respectent les règles suivantes :

- Chaque niveau ne communique qu'avec le niveau sous-jacent et dépend seulement de ce dernier
- Chaque niveau n'a pas connaissance d'aucun niveau en dehors du niveau sous-jacent

En général il peut y avoir plus que trois niveaux entre l'application et la base de données. Cela permet d'avoir une faible dépendance entre les composants, par exemple, on pourra changer la base de données avec peu de modification. Une application multi-tiers

permet de distribuer la charge de travail. Ça permet d'améliorer l'utilisation du réseau et éviter la saturation du réseau même et des composants.

La figure suivante montre en détail une architecture distribuée.

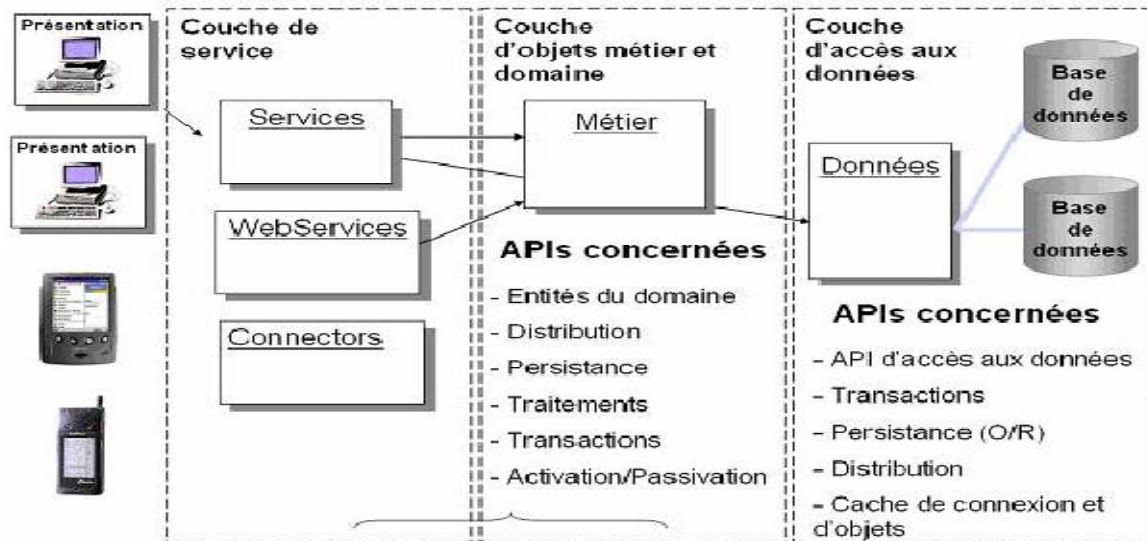


Figure II.5: Exemple d'une Architecture Distribuée.

- **La couche de présentation** contient les différents types de clients, léger (Web, ASP, JSP) ou lourd (Swing, WinForm)
- **La couche de service** contient les traitements (contrôleurs de Use Case UML) représentant les règles métier (créer un compte, rechercher un client, calculer un amortissement, ...)
- **La couche d'objets métier** est représentée par les objets du domaine, c'est à dire l'ensemble des entités persistantes de l'application (Facture, Bon de Commande, Client, ...)
- **La couche d'accès aux données** contient les usines d'objets métier, c'est à dire les classes chargées de créer des objets métier de manière totalement transparente, indépendamment de leur mode de stockage (SGBDR, Objet, Fichiers, Legacy, ...)

II.4.3 Architecture J2EE

II.4.3.1 Définition

Java 2 Enterprise Edition (J2EE) est une spécification pour le langage de programmation Java de Sun destinée aux applications d'entreprise. J2EE offre une plateforme de développement et déploiement en langage Java pour les applications distribuées à plusieurs niveaux.

On parle généralement de « plate-forme J2EE » pour désigner l'ensemble constitué des services API (Application Programmable Interface) offerts et de l'infrastructure d'exécution.

Dans la mesure où J2EE s'appuie entièrement sur Java, il bénéficie des avantages de ce langage, en particulier une bonne portabilité et une maintenance du code. De plus, l'architecture J2EE repose sur des composants distincts, interchangeables et distribués, ce qui signifie notamment qu'un système reposant sur J2EE peut posséder des mécanismes de haute-disponibilité, afin de garantir une bonne qualité de service et que la maintenance des applications est facilitée.

II.4.3.2 Fonctionnement interne

Le langage Java, sur lequel les librairies J2EE sont utilisées, met à disposition un compilateur et une machine virtuelle (JVM – Java Virtual Machine) qui se charge de créer un environnement standard pour le lancement de l'application sur tout type de système opérationnel. Le compilateur compile le code source et produit le bytecode, soit un code intermédiaire qui sera en suite lu par la machine virtuelle Java. Chaque système opérationnel majeur possède une JVM expressément codée.

II.4.3.3 Architecture

J2EE ajoute de nombreuses couches de niveau entreprise au-dessus de la plate-forme J2SE - Java

Standard Edition. Chaque couche est conçue pour supporter une différente technologie de développement.

- **Technologie web application:** technologies liées à la production des interfaces web dynamiques, par exemple JSP (Java server Pages) et Servlets



Figure II.6: Exemple d'une requête d'un client léger J2EE.

- **La notion de Servlet**

C'est un Composant logiciel écrit en Java fonctionnant du côté serveur, qui Permet de gérer des requêtes HTTP et de fournir au client une réponse HTTP, son exécution se fait dans un moteur de Servlet ou conteneur de Servlet permettant d'établir le lien entre le Servlet et le serveur Web.

Le schéma suivant montre l'architecture générale d'un Servlet :

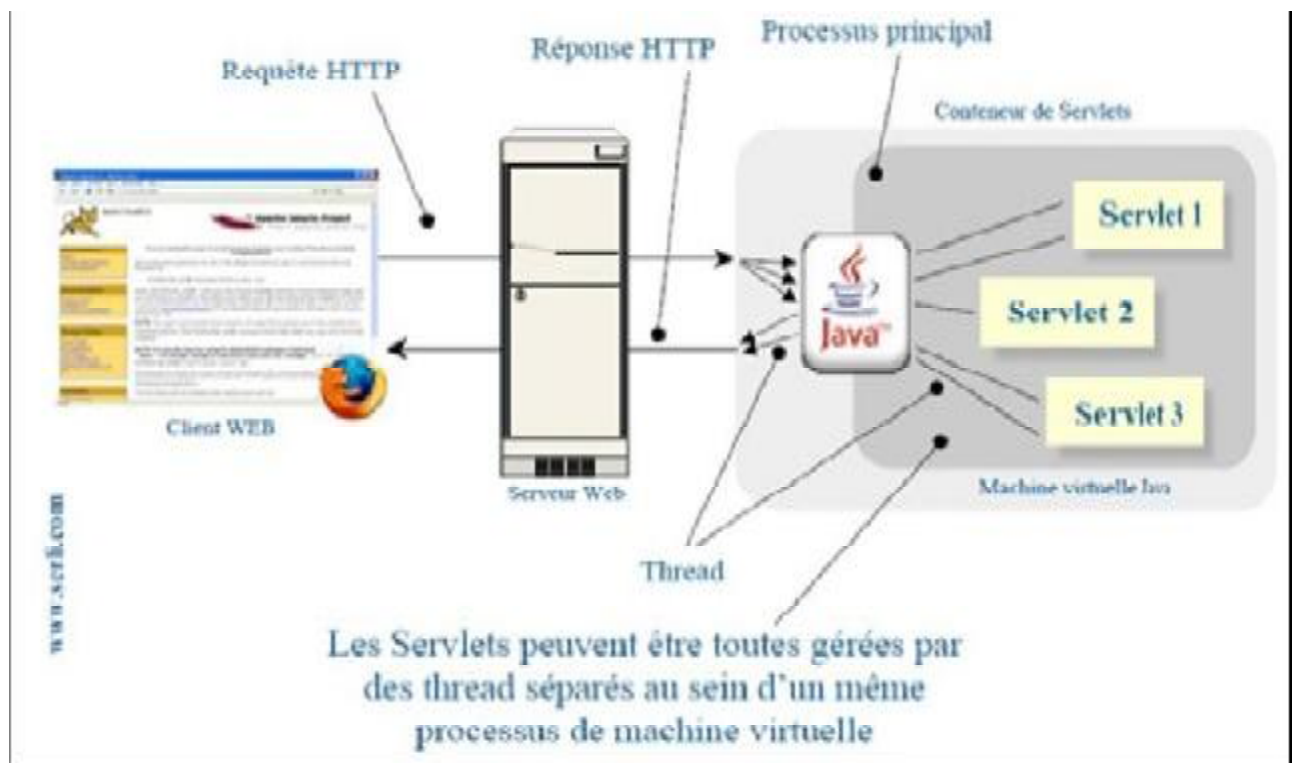


Figure II.7 : Architecture d'un Servlet

- **La notion de JSP**

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

Ce schéma suivant illustre les étapes d'exécution d'une JSP :

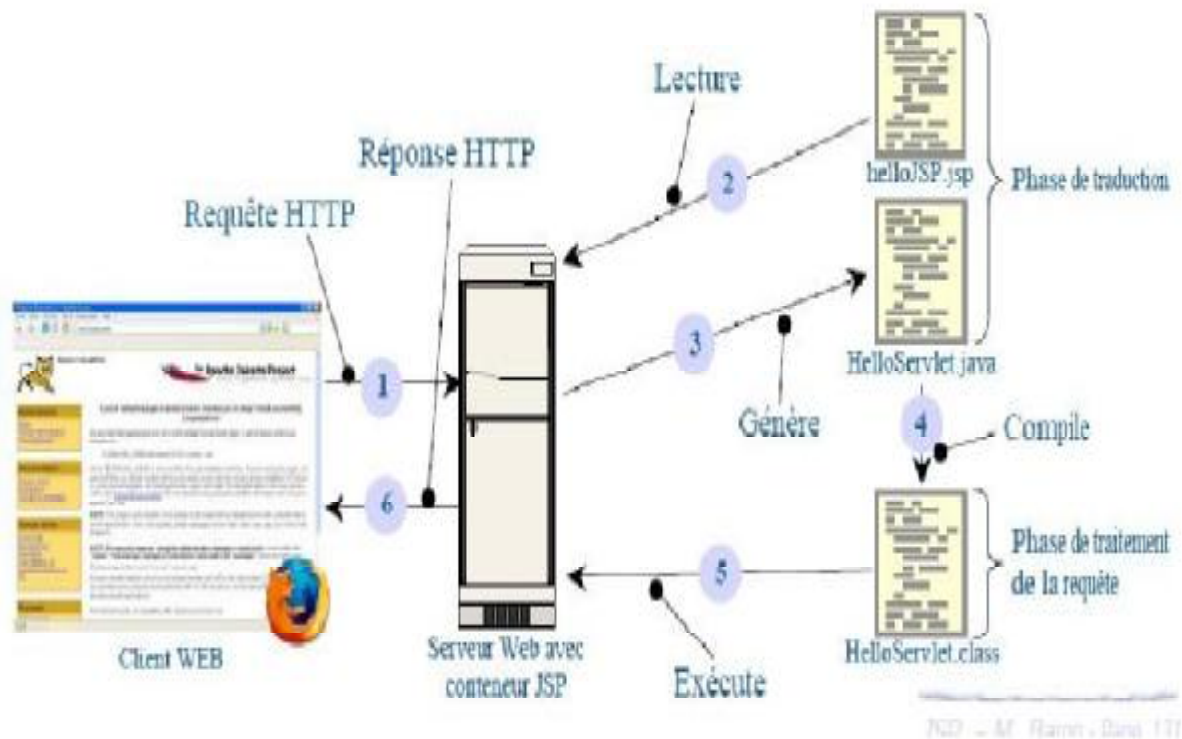


Figure II.8 : conversion d'une JSP

- **Technologie entreprise application:** technologies plus directement liées à la logique de business : EJB (Enterprise Java Bean), JavaMail, JMS (Java Message Service), JTA (Java Transaction), etc.
- **Technologie web services:** technologies utiles au développement des applications adhérentes au paradigme SOA (Service Oriented Architecture) : web services, JAX-WS (java API for XML-based web services), JAX-RPC (Java API for XML-Based RPC)
- **Technologie management and Security:** technologies liées à la gestion de la technologie entreprise afin de réaliser l'accès et l'échange d'information entre machines et services distribués : JAAS (Java Authentication and Authorization Service), JCA (Java Connector Architecture)

Pour expliquer l'utilisation de ces technologies on peut imaginer que les technologies entreprise sont utilisées pour gérer l'accès aux données (généralement un ou plus DataBase), les technologies web application sont utilisées pour montrer les données aux utilisateurs

Chapitre II : Développement d'application

génériques. Dans un contexte Business to Business, les technologies web service seront utilisées pour échanger les informations avec les partenaires commerciales et les technologies de gestion gèrent tous les processus informationnels assurant la sécurité des transactions.

Ce schéma illustre en détail l'architecture générale J2EE.

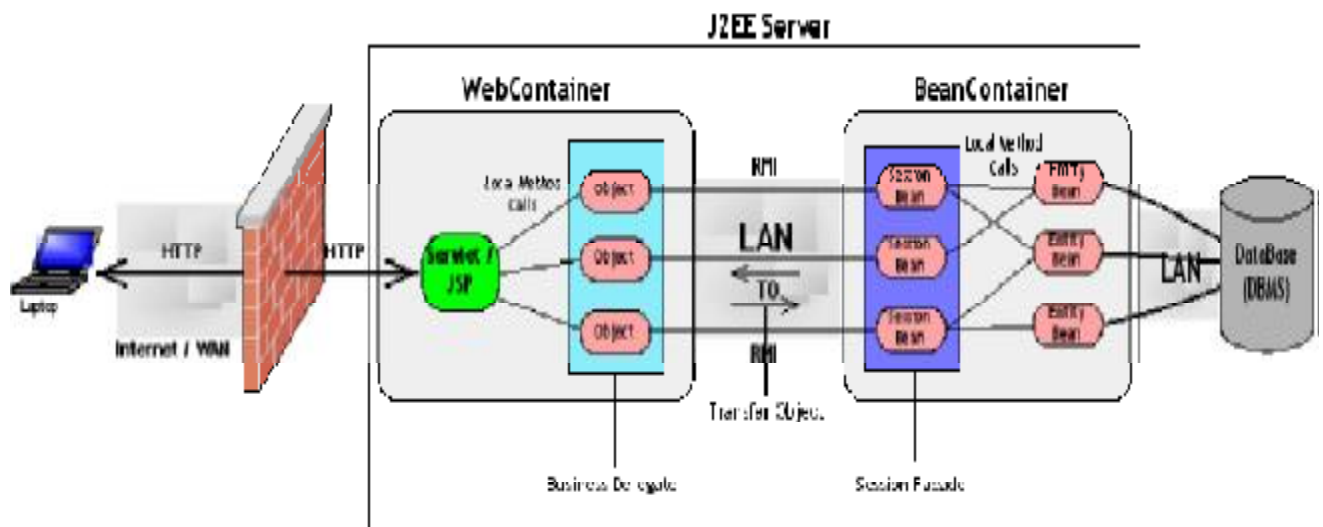


Figure II.9: Architecture Générale J2EE.

L'architecture J2EE permet ainsi de séparer la couche présentation, correspondant à l'interface homme-machine (IHM), la couche métier contenant l'essentiel des traitements de données en se basant dans la mesure du possible sur des API existantes, et enfin la couche de données correspondant aux informations de l'entreprise stockées dans des fichiers, dans des bases de données relationnelles ou XML, dans des annuaires d'entreprise ou encore dans des systèmes d'informations complexes.

II.4.3.4 Le serveur d'application

Les API J2EE ont été projetés pour travailler avec un particulier type de serveur appelé J2EE Application Server. Un serveur d'application est une couche software résident sur une machine serveur qui fournit les services que la technologie J2EE nécessite. Il y a plusieurs applications serveurs. Parmi les produits commerciaux on rappelle Bea WebLogic, IBM Web sphere, Sun Application Serveur, etc. Parmi les produits libres le plus connu est JBOSS. Les différences principales entre les différents serveurs d'applications sont relatives aux opérations de deploy, clustering, etc.... Par contre toutes les fonctionnalités qui concernent strictement le fonctionnement des applications J2EE adhèrent aux spécifications proposées

par la Sun. Un serveur d'application s'installe et se lance comme un normal serveur web (de fait il met à disposition des services typiques d'un serveur web). En plus il possède un panneau d'administration accessible par un poste distant à travers lequel on peut définir les différents services. On peut considérer le serveur d'application comme une application Java standalone exécutée par une J2SE qui dessert les requêtes provenant des différents clients.

II.4.3.5 Outils de programmation

Plusieurs Environnements de développement intégrés (IDE) sont disponibles pour la plate-forme J2EE. Parmi les logiciels libres on rappelle Eclipse de IBM, qui met à disposition, grâce au système des plug-ins, un riche environnement de développement. Encore on retrouve NetBeans de la Sun que, à différence d'Eclipse, a été créé exclusivement pour faciliter le développement d'applications J2EE, c'est l'outil que nous avons choisi pour le développement de notre application.

II.4.4 Architecture .NET

II.4.4.1 Définition

.NET est un produit proposé par la société Microsoft, pour le développement d'applications d'entreprises multi-niveaux, basées sur des composants Microsoft.

.NET constitue ainsi la réponse de Microsoft à la plate-forme J2EE de Sun. Annoncé pour l'année 2000, la première version du langage a été livrée en 2002.

II.4.4.2 Fonctionnement interne

Un moteur d'exécution, appelé CLR (Common Language Runtime) permet de compiler le code source de l'application en un langage intermédiaire, baptisé MSIL (Microsoft Intermediate Language) ou CIL (Common Intermediate Language). Lors de la première exécution de l'application, le code MSIL est à son tour compilé à la volée en code spécifique au système grâce à un compilateur JIT (Just In Time). .NET est de fait un Framework qui contient différents langages de programmations intégrés entre eux. .NET a été conçu pour les systèmes Windows et bientôt il sera disponible pour toutes les systèmes opérationnels.

II.4.4.3 Architecture

La plate-forme .NET a été élaborée en s'appuyant sur une communauté d'utilisateurs et a abouti à l'élaboration de spécifications. Ces spécifications ont été ratifiées par un organisme international de standardisation, l'ECMA (European Computer Manufacturers Association), ce qui en fait un standard. Ainsi l'effort de standardisation a permis l'émergence de plates-

formes portées par des entreprises tierces et disponibles sous un grand nombre de systèmes d'exploitation.

Ce schéma montre un client léger .NET :



Figure II.10 : Exemple d'un client léger .NET.

On parle généralement de «Framework» pour désigner l'ensemble constitué des services (API) offerts et de l'infrastructure d'exécution. Le Framework NET comprend notamment :

II.4.4.4 L'environnement d'exécution

- La machine virtuelle (CLR)
- Un environnement d'exécution d'applications et de services web, appelé ASP .NET ;
- Un environnement d'exécution d'applications lourdes, appelé WinForms.
- Des services, sous forme d'un ensemble hiérarchisé de classes appelé Framework Class Library (FCL). La FCL est ainsi une librairie orientée objet, fournissant des fonctionnalités pour les principaux besoins actuels des développeurs. Le SDK (Software Development Kit) fournit une implémentation de ces classes.

Le schéma suivant de l'architecture générale de .NET :

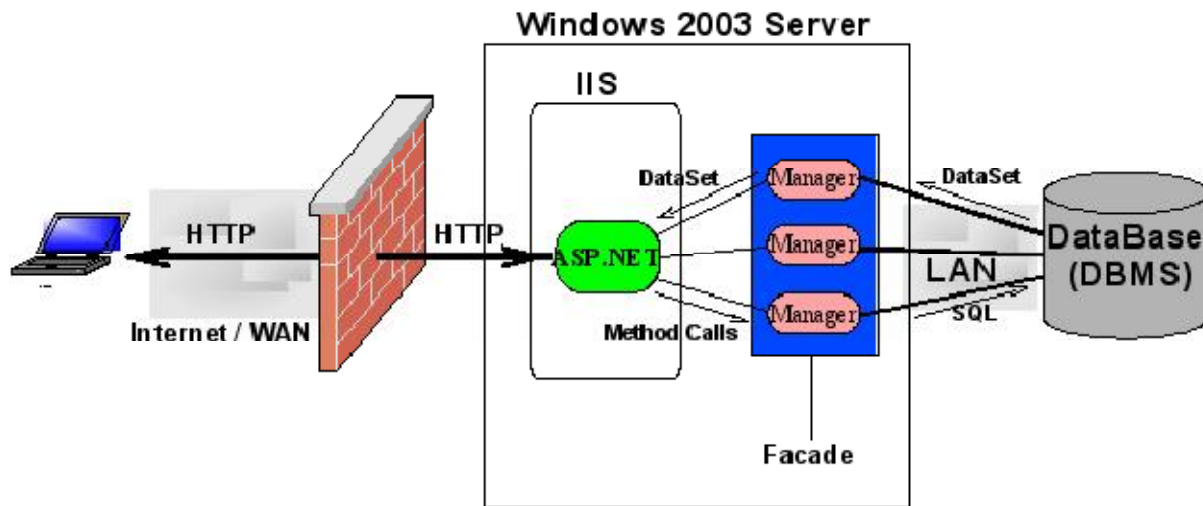


Figure II.11 : Architecture Générale .NET.

II.4.4.5 Serveur d'application

Le serveur d'application de .NET est directement le système d'exploitation Windows Server 2003. Ce système d'exploitation s'accompagne du Framework .NET et des middlewares sous-jacents (IIS10, COM+11 et MSMQ Microsoft Message Queuing (MSMQ) 12).

Pour les autres implémentations, par exemple, Mono, il n'y a pas à proprement parler de serveur d'application. La partie web tourne sur un serveur web tel que celui d'Apache, et les parties métier et accès aux données sont tout simplement exécutées avec Mono (**supporte actuellement une dizaine de langages de programmation différents**).

II.4.4.6 Outils de programmation

L'outil de développement de .NET par excellence est Visual Studio (VS) de Microsoft. Toutefois, il n'est pas sans concurrence, et d'autres IDE ont fait leur apparition, notamment Borland C# Builder qui n'a rien à envier à son concurrent de chez Microsoft, Visual Studio (VS) de Microsoft cet IDE très complet, s'occupe du développement d'application web, Borland C# Builder et Visual Studio Express...etc.

II.5 Conclusion

Au cours de ce chapitre, nous avons présenté la notion de base des réseaux informatiques qui sont un moyen pour minimiser les coûts de transport des informations et d'augmenter les performances des systèmes, nous avons parlé des modèles OSI et TCP/IP, et nous avons parlé aussi de l'architecture client/serveur, puis nous avons terminé par une description des architectures distribuées J2EE et .NET. Ces notions seront utilisées pour le développement de notre application. Dans le chapitre suivant, nous aborderons la conception de l'application en utilisant le langage de modélisation UML.

III.1 Introduction

La description de la programmation par objets a fait ressortir l'étendue du travail conceptuel nécessaire : définition des classes, de leurs relations, des attributs et méthodes, des interfaces,...etc.

Pour programmer une application il faut d'abord organiser ses idées, les documenter, puis organiser la réalisation en définissant les modules et les étapes de la réalisation. C'est cette démarche antérieure à l'écriture que l'on appelle modélisation ; son produit est un modèle.

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

III.2 Langage de Modélisation UML

III.2.1 définition

UML (Unified Modeling Language, que l'on peut traduire par "langage de modélisation unifié") est une notation permettant de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes existant auparavant, et est devenu désormais la référence en terme de modélisation objet, à un tel point que sa connaissance est souvent nécessaire pour obtenir un poste de développeur objet.

III.2.2 UML en œuvre

UML est un langage de modélisation avec plusieurs objectifs qui en font un véritable outil de communication:

- Comprendre et décrire les besoins,
- Spécifier (un système),
- Etablir l'architecture logicielle.

Les diagrammes d'UML vont mettre en place deux parties comme le montre le tableau suivant :

Statique (structure)	Dynamique (comportement)
Diagramme de cas d'utilisation Diagramme de classe Diagramme à objet Diagramme de composants Diagramme de déploiement	Diagramme de collaboration Diagramme de séquence Diagramme état/ transition Diagramme d'activités

Tableau III.1 : les diagrammes d'UML.

III.3 Analyse et conception

A. Analyse

Cette activité commence par la mise en évidence des différents acteurs intervenants dans le système cible ainsi que leurs besoins. Ensuite, la phase conception donnera la modélisation des objectifs à atteindre en s'appuyant sur les résultats de la phase analyse.

A.1 Quelques définitions de bases

Acteur : entité externe qui agit sur le système ; Le terme acteur ne désigne pas seulement les utilisateurs humains mais également les autres systèmes. Les acteurs sont des entités qui représentent des rôles au travers d'une certaine utilisation (cas) et non pas des personnes physiques.

Cas d'utilisation : un cas d'utilisation « **use case** » représente un ensemble d'actions réalisées par le système en réponse à une action d'un acteur. On distingue plusieurs types des cas d'utilisations :

- les cas d'utilisation peuvent être structurés.
- Ils peuvent être organisés en paquetages.
- l'ensemble des cas d'utilisation décrit les objectifs (buts) du système.

Scénario : un scénario représente une succession particulière d'enchaînement qui s'exécute du début à la fin du cas d'utilisation, un enchaînement étant l'unité de description de séquences d'actions.

Un cas d'utilisation contient en général, un scenario nominal et plusieurs scenarios alternatifs (qui se terminent d'une façon normale) ou d'erreurs (qui se terminent en échec).

La figure suivante donne la représentation graphique de la démarche de modélisation :

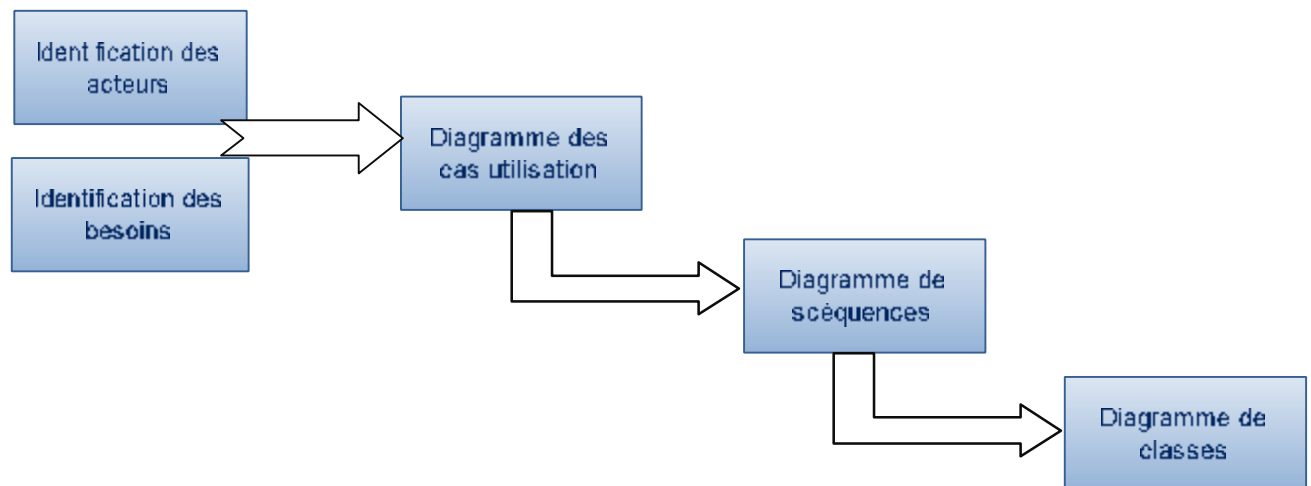


Figure III.1 : Démarche de modélisation.

A.2 Spécification des exigences d'après les cas d'utilisation

Modèle des cas d'utilisation

En détaillant les différentes étapes de la démarche que nous allons adopter afin d'aboutir au modèle des cas d'utilisations.

- Identification des acteurs.
- Identification des cas d'utilisation.
- Ajout des relations entre cas d'utilisation (spécification des scénarios).
- Finalisation des diagrammes de cas d'utilisation.

A.2.1 Identification des acteurs

Dans notre application, nous recensons deux (2) acteurs :

- § **Utilisateur** : Toute personne, inscrit et validé par l'administrateur du site.
- § **Administrateur** : La personne chargé de l'administration du site.

A.2.2 Identification des cas d'utilisation

Les acteurs définis précédemment effectuent un certain nombre de tâches, ces dernières sont résumées dans le tableau ci-dessous :

Acteurs	Tâches
Utilisateur	T₀ : Accéder à la page Utilisateur. T₁ : S'inscrire. T₂ : Quitter l'application. T₃ : Enregistrer de papier en entrée. T₄ : Enregistrer de papier en sortie. T₅ : Effectuer une commande. T₆ : Consulter la liste de papier. T₇ : Consulter la liste des commandes. T₈ : Consulter l'historique de papier entré. T₉ : Consulter l'historique de papier sorti. T₁₀ : Réformer un papier. T₁₁ : Consulter la liste de papier réformé. T₁₂ : Consulter la liste des agences/services.
Administrateur	De T₀ à T₁₂ T₁₃ : Accéder à la page Administrateur. T₁₄ : Ajouter un nouveau type de papier. T₁₅ : Supprimer un type de papier. T₁₆ : Supprimer une entrée de papier. T₁₇ : Supprimer une sortie de papier. T₁₈ : Ajouter une Agence ou un service. T₁₉ : Supprimer une Agence ou un service. T₂₀ : valider un Utilisateur. T₂₁ : Supprimer un Utilisateur. T₂₂ : Voir la liste des Utilisateur.

Tableau III.2 : tableau de spécification des tâches.

A.2.3 Spécification des scenarios

Un cas d'utilisation peut avoir une ou plusieurs instances représentées par des scénarios chacun des acteurs qu'on a définis, effectue des tâches, et chaque tâche est décrite par des scénarios qu'on résume comme suit :

ü Pour un Utilisateur

Tâches	Scénarios
T₀ : Accéder à la page Utilisateur.	S₀ : Dans la barre de menu de la page d'accueil cliquer sur le lien «Accéder Utilisateur » S₁ : Remplir le formulaire de la page « authentification Utilisateur». S₂ : Cliquer sur le bouton « Connecter ».
T₁ : S'inscrire.	S₃ : Dans la barre de menu de la page d'accueil cliquer sur le lien «S'inscrire». S₄ : Remplir le formulaire de la page « inscription Utilisateur » S₅ : Cliquer sur le bouton « s'inscrire»
T₂ : Quitter l'application.	S₆ : Dans la barre de menu de la page d'accueil cliquer sur le lien «Quitter».
T₃ : Enregistrer de papier en entrée.	S₇ : A la page (Utilisateur/Administrateur) dans la barre de menu cliqué sur le lien « Entrer Papier ». S₈ : Remplir le formulaire de la page « papier en entrée » S₉ : Cliquer sur le bouton « valider »

Chapitre III : Analyse et Conception

T₄ : Enregistrer de papier en sortie.	S₁₀ : A la page (Utilisateur/Administrateur) dans la barre de menu cliqué sur le lien « Sortir Papier ». S₁₁ : Remplir le formulaire de la page « sortir du papier» S₁₂ : Cliquer sur le bouton « valider »
T₅ : Effectuer une commande.	S₁₃ : A la page (Utilisateur/Administrateur) dans la barre de menu cliquer sur le lien « faire une commande ». S₁₄ : Choisir les types de papier a commandé dans la page « commande » puis validé. S₁₅ : entré les quantités pour les papiers sélectionnés. S₁₆ : Cliquer sur le bouton « valider »
T₆ : Consulter la liste de papier.	S₁₇ : A la page (Utilisateur/Administrateur) dans la barre de menu cliquer sur le lien « liste de papier ». S₁₈ : cliqué sur le bouton imprimer pour imprimer la page.
T₇ : Consulter la liste des commandes.	S₁₉ : A la page (Utilisateur/Administrateur) dans la barre de menu cliquer sur le lien « liste des commandes ». S₂₀ : cliqué sur le bouton imprimer pour imprimer la page.
T₈ : Consulter l'historique de papier entré	S₂₁ : A la page (Utilisateur/Administrateur) dans la barre de menu cliqué sur le lien « Historique papier entries ».
T₉ : Consulter l'historique de papier sorti.	S₂₂ : A la page d'accueil Utilisateur/administrateur dans la barre de menu cliqué sur le lien « Historique papier sorties ».

T₁₀ : Réformer un papier.	S₂₃ : A la page (Utilisateur/administrateur) dans la barre de menu cliqué sur le lien « Réformé un papier ». S₂₄ : Remplir le formulaire de la page « Réforme papier » S₂₅ : Cliquer sur le bouton « valider »
T₁₁ : Consulter la liste de papier réformé.	S₂₆ : A la page (Utilisateur/administrateur) dans la barre de menu cliqué sur le lien « liste de papier réformé ».
T₁₂ : Consulter la liste des agences/services.	S₂₇ : A la page (Utilisateur/administrateur) dans la barre de menu cliqué sur le lien « liste des agences ».

Tableau III.3 : tableau spécification des scénarios pour l'utilisateur.

ü Pour L'Administrateur

Tâches	Scenarios
T₁₃ : Accède à la page Administrateur « S'Authentifier Admin ».	S₂₈ : Dans la page d'accueil Cliquer sur le lien « Accès administrateur ». S₂₉ : Saisir le login et le mot de passe . S₃₀ : Cliquer sur le bouton « Connecter ».
T₁₄ : Ajouter un nouveau type de papier.	S₃₁ : Dans la page Administrateur Cliquer sur le lien « Ajouter un nouveau type papier ». S₃₂ : Remplir le formulaire de la page « Ajouter un nouveau type papier ». S₃₃ : Cliquer sur le bouton « Ajouter ».

Chapitre III : Analyse et Conception

T₁₅ : Supprimer un type de papier.	S₃₄ : Dans la barre de menu de la page « Administrateur » cliquer sur le lien « Supprimer un papier ». S₃₅ : Sélectionner le code de papier à supprimer. S₃₆ : Cliquer sur le bouton « Supprimer ».
T₁₆ : Supprimer une entrée de papier.	S₃₇ : Dans la barre de menu de la page « Administrateur » cliquer sur le lien « Supprimer une entrée papier ». S₃₈ : Sélectionner l'entrée papier à supprimer. S₃₉ : Cliquer sur le bouton « Supprimer ».
T₁₇ : Supprimer une sortie de papier.	S₄₀ : Dans la barre de menu de la page « Administrateur » cliquer sur le lien « Supprimer une sortie papier ». S₄₁ : Sélectionner la sortie papier à supprimer. S₄₂ : Cliquer sur le bouton « Supprimer ».
T₁₈ : Ajouter une Agence ou un service.	S₄₃ : A la page « Administrateur » dans la barre de menu cliquer sur le lien « Ajout une nouvelle Agence/Service ». S₄₄ : Remplir le formulaire de la page « Ajout une nouvelle Agence/Service ». S₄₅ : Cliquer sur le bouton « Ajouter »
T₁₉ : Supprimer une Agence ou un service.	S₄₆ : A la page « Administrateur » dans la barre de menu cliquer sur le lien « Supprimer d'Une Agence/Service ». S₄₇ : Sélectionner « l'Agence/Service » à Supprimer. S₄₈ : Cliquer sur le bouton « Supprimer ».

Chapitre III : Analyse et Conception

T₂₀ : valider un Utilisateur.	S₄₉ : A la page « Administrateur » dans la barre de menu cliquer sur le lien « valider un Utilisateur ». S₅₀ : Sélectionner « l'Utilisateur » à valider. S₅₁ : Cliquer sur le bouton « Accepté ».
T₂₁ : Supprimer un Utilisateur.	S₅₂ : A la page « Administrateur » dans la barre de menu cliquer sur le lien « Supprimer d'un Utilisateur ». S₅₃ : Sélectionner Un Utilisateur à Supprimer. S₅₄ : Cliquer sur le bouton « Supprimer ».
T₂₂ : Voir la liste des Utilisateur.	S₅₅ : A la page « Administrateur » dans la barre de menu cliquer sur le lien « Liste des Utilisateurs ».

Tableau III.4 : tableau spécification des scénarios pour l'Administrateur.

A.2.4 Spécification des cas d'utilisations

Les figures si dessous présentent une description de quelque cas d'utilisations de notre projet :

Ø **Cas d'utilisation: « S'authentifier »** pour un Utilisateur.

Cas d'utilisation: S'authentifier.

Scenarios: S₀, S₁ et S₂.

Acteurs : Utilisateur.

Résumé : Cette fonctionnalité permet aux **Utilisateurs** d'accéder à leurs espaces personnels.

Description :

- L'Utilisateur accède à l'application.
- Le système affiche la page d'accueil d'application.
- L'Utilisateur clique sur le lien « accéder Utilisateur » sur la page d'accueil.
- L'Utilisateur saisie son login et son mot de passe sur la page d'authentification.
- Il clique sur le bouton « **Connexion** » pour envoyer ses informations.
- Le système vérifie la validité des informations envoyées.
- Si les informations sont correctes le système affiche la page d'accueil Utilisateur, sinon il lui affiche un message d'erreur, et le renvoyer vers la page d'authentification.

Use case : « **S'authentifier** »

Ø Use case : « **entrée de papier** » pour un Utilisateur/Administrateur.

Use case: enregistre le papier en entrée.

Scenarios: S₇, S₈ et S₉.

Acteurs : Utilisateur/Administrateur.

Résumé : Cette fonctionnalité permet, aux Utilisateurs de l'application de l'entreprise, d'enregistrer l'arrivée de papier et l'ajouter dans le stock.

Description :

- Après authentification.
- Le système affiche la page d'espace d'accueil Utilisateur ou Administrateur.
- Cette page contient le formulaire d'entrée de papier.
- L'Utilisateur remplit le formulaire.
- A chaque fois qu'il choisit un type de papier à entrer avec une quantité Le système vérifie la validité des informations envoyées.
- Si les informations sont correctes, le système lui affiche un message de confirmation d'enregistrement de la tâche, sinon il lui affiche un message d'erreur.

Use case : « **entrée de papier** »

Ø Use case : « **Sortir de papier** » pour un Utilisateur.

Use case: Faire enregistrer le papier en sortie.

Scenarios: S₁₀, S₁₁ et S₁₂.

Acteurs : Utilisateur/Administrateur.

Résumé : Cette fonctionnalité permet, aux Utilisateurs de l'application de l'entreprise, de faire enregistrer la tâche de papier en sortie aux différents agences/services liées à l'organisme et de l'enlevé de stock.

Description :

- Après authentification.
- Le système affiche la page d'espace accueille Utilisateur/Administrateur.
- Cette page contient le formulaire sortir de papier.
- L'Utilisateur remplit le formulaire.
- A chaque fois qu'il choisit un type de papier à sortir avec une quantité Le système vérifie la validité des informations envoyées.
- Si les informations sont correctes, le système lui affiche un message de confirmation d'enregistrement de la tâche, sinon il lui affiche un message d'erreur.

Use case : « **sortir de papier** »

Ø Use case : « **Gérer les Utilisateurs** » pour un Administrateur.

Use case : Gérer les Utilisateur.

Scenarios: S₄₉, S₅₀, S₅₁, S₅₂, S₅₃, S₅₄, et S₅₅.

Acteurs : Administrateur.

Résumé : Cette fonctionnalité à l'administrateur de valider un nouveau Utilisateur, supprimer ou consulter la liste des Utilisateurs.

Description :

- Après authentification.
- Le système affiche la page d'accueil « **administrateur** ».
- L'administrateur sélectionne dans le menu le lien « **Gérer les Utilisateurs** ».
- Il clique sur « **liste des Utilisateurs** » pour afficher la liste des Utilisateurs.
- Il clique sur « **Supprimer un Utilisateur** » pour supprimer un Utilisateur.
- Il clique sur « **valider un Utilisateur** » pour ajouter un Utilisateur.

Use case : « **Gérer les Utilisateurs** ».

Ø Use case : « **Ajout d'un nouveau type de papier** » pour un Administrateur.

Use case : Ajout d'un nouveau type de papier.

Scenarios: S₃₁, S₃₂ et S₃₃.

Acteurs : Administrateur.

Résumé : Cette fonctionnalité à l'administrateur d'ajouter un nouveau type de papier.

Description :

- Après authentification.
- Le système affiche la page d'accueil « **administrateur** ».
- L'administrateur sélectionne dans le menu le lien «**Ajout d'un nouveau type de papier**».
- Il clique sur « **Ajout d'un nouveau type de papier**».
- Le système affiche la page « **Ajout d'un nouveau type de papier**».
- Cette page contient le formulaire d'ajout un nouveau type de papier.
- L'Administrateur remplit le formulaire.
- A chaque fois qu'il fait remplit les informations nécessaires avec une quantité et un seuil sur le type de papier, Le système vérifie la validité des informations envoyées.
- Si les informations sont correctes, le système lui affiche un message de confirmation d'enregistrement de la tâche, sinon il lui affiche un message d'erreur.

Use case : « **Ajout d'un nouveau type de papier**».

Ø Use case : « **Gérer les Agences/services** » pour un Administrateur.

Use case : Gérer les Agences/Services.

Scenarios: S₄₃, S₄₄, S₄₅, S₄₆, S₄₇, S₄₈.

Acteurs : Administrateur.

Résumé : Cette fonctionnalité à l'administrateur d'ajouter ou de supprimer un service/agence.

Description :

- Après authentification.
- Le système affiche la page d'accueil « **administrateur** ».
- L'administrateur sélectionne dans le menu le lien «**Gérer les Agences/services**».
- Il clique sur « **liste des Agences/services** » pour afficher la liste des agences.
- Il clique sur « **Supprimer une Agence/service** » pour supprimer une agence.

Use case : « **Gérer les Utilisateurs** ».

A.2.5 Diagramme de cas d'utilisation générale :

Il décrit le comportement du système du point de vue utilisateur sous la forme d'actions et de réactions. Il existe deux concepts fondamentaux dans la modélisation par cas d'utilisation :

- Les acteurs qui utilisent le système.
- Les cas d'utilisation qui représentent l'utilisation du système par les acteurs.

Chaque cas d'utilisation indique une fonctionnalité du système déclenchée par un acteur externe du système. Ce genre de diagramme permet de mettre en place et de comprendre les besoins des utilisateurs.

Afin d'optimiser la formalisation des besoins en ayant notamment recours à la réutilisation des cas d'utilisation, trois relations ont été décrites entre cas d'utilisation:

- **Relation d'inclusion « include »** : Une relation d'inclusion d'un cas d'utilisation A par rapport à un cas d'utilisation B signifie qu'une instance de A contient le comportement décrit dans B. A ne peut être exécuté seul.
- **Relation d'extension « extend »** : Une relation d'extension d'un cas d'utilisation A par rapport à un cas d'utilisation B signifie qu'une instance de A peut être étendue par le comportement décrit dans B.
- **Relation Use « uses »** : Lorsqu'une ou plusieurs tâches sont utilisées régulièrement on peut les factoriser dans un même use case, et faire de telle sorte que d'autres use cases l'utilisent en le pointant par une flèche.

A.2.6) Diagramme de cas d'utilisation:

Ø Utilisateur

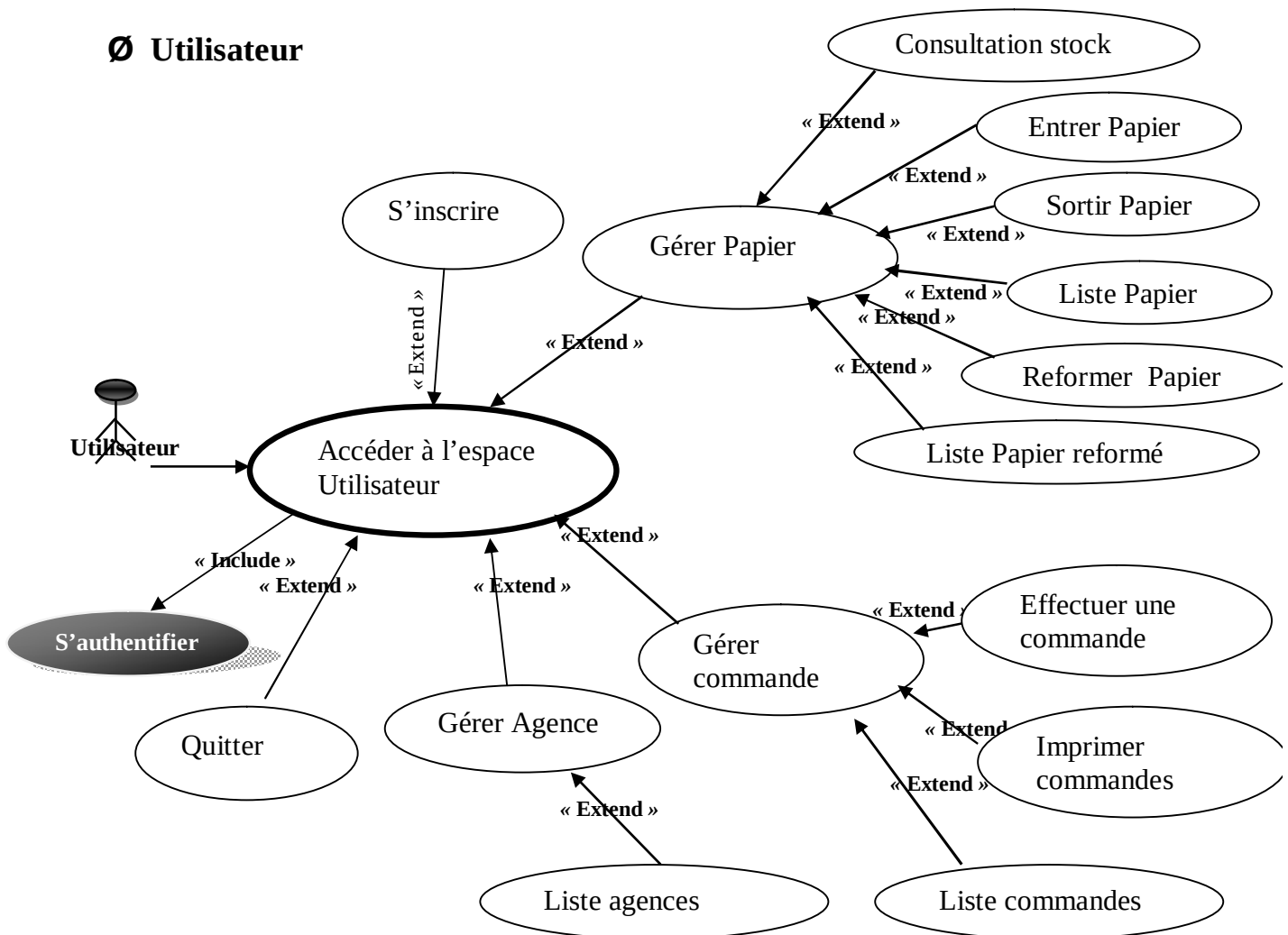


Figure III.2 : Diagramme de cas d'utilisation pour l'Utilisateur.

Ø Administrateur

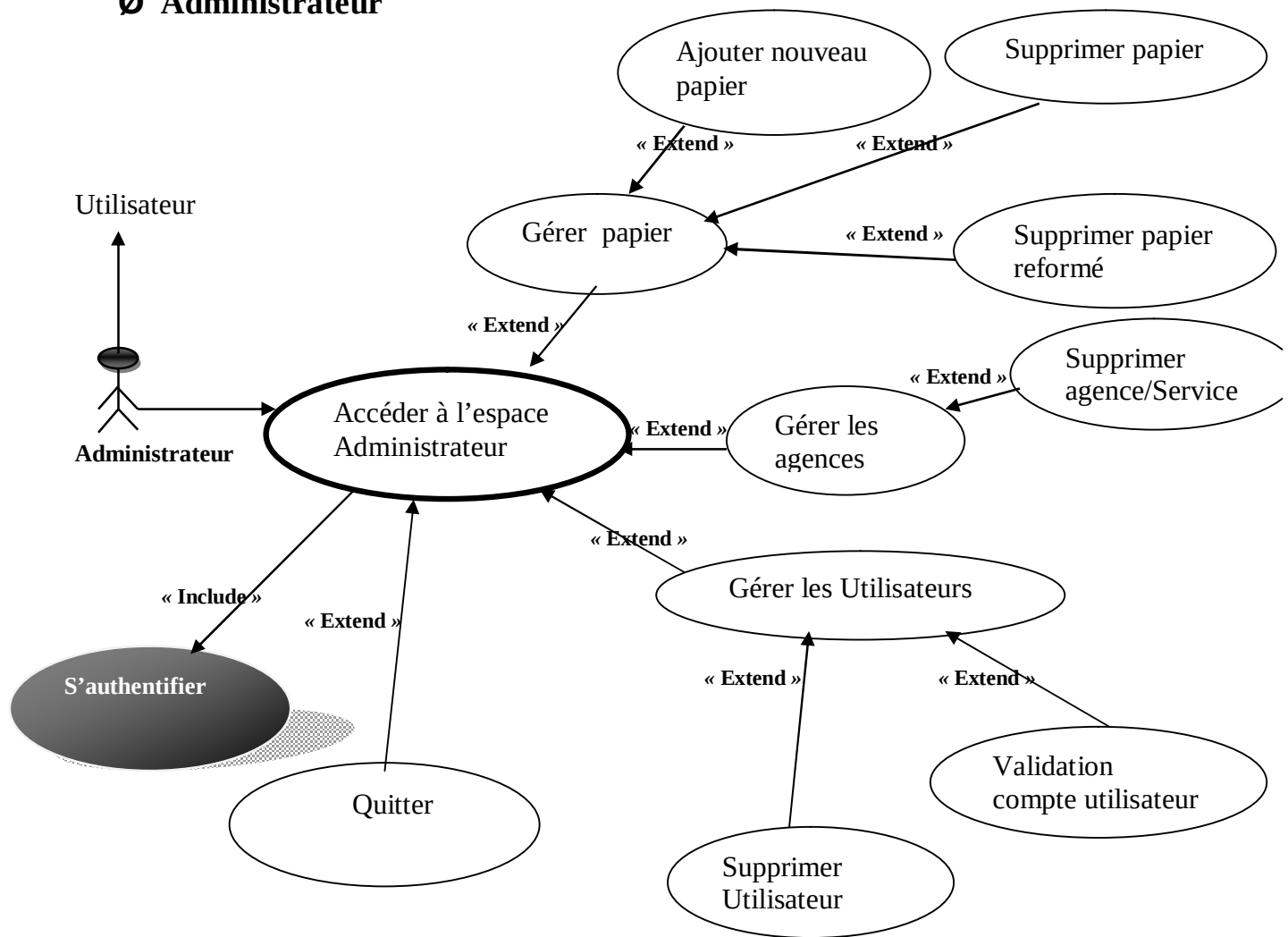


Figure III.3 : Diagramme de cas d'utilisation pour l'Administrateur.

B. Conception

Après avoir déterminé les cas d'utilisations et spécifié les besoins de l'application, nous allons passer à la phase de conception du projet, qui consiste à affiner les spécifications d'après leurs contenus.

Notre étude se base toujours sur le formalisme UML qui fournit un ensemble de diagrammes permettant d'affiner le modèle d'analyse jusqu'à l'obtention d'une nouvelle vue du modèle qui sera réalisable sous forme logicielle.

B.1 Elaboration des diagrammes de séquences

Cette étape se traduit par la détermination des pages web et les relations qu'elles entretiennent entre elles ainsi qu'avec les autres objets du système. Durant cette étape les objets système des diagrammes de séquence créés pendant l'analyse se transforment en objet et en pages web.

Le diagramme de séquence représente la succession chronologique des opérations réalisées par un acteur : saisir une donnée, consulter une donnée, lancer un traitement ; il met en évidence les objets manipulés ainsi que les opérations qui font passer d'un objet à l'autre.

Nous présentons dans ce qui suit quelques diagrammes de séquence pour les cas d'utilisation précédemment définis :

Ø Diagramme de séquences « inscription Utilisateur/Créer un compte Utilisateur »

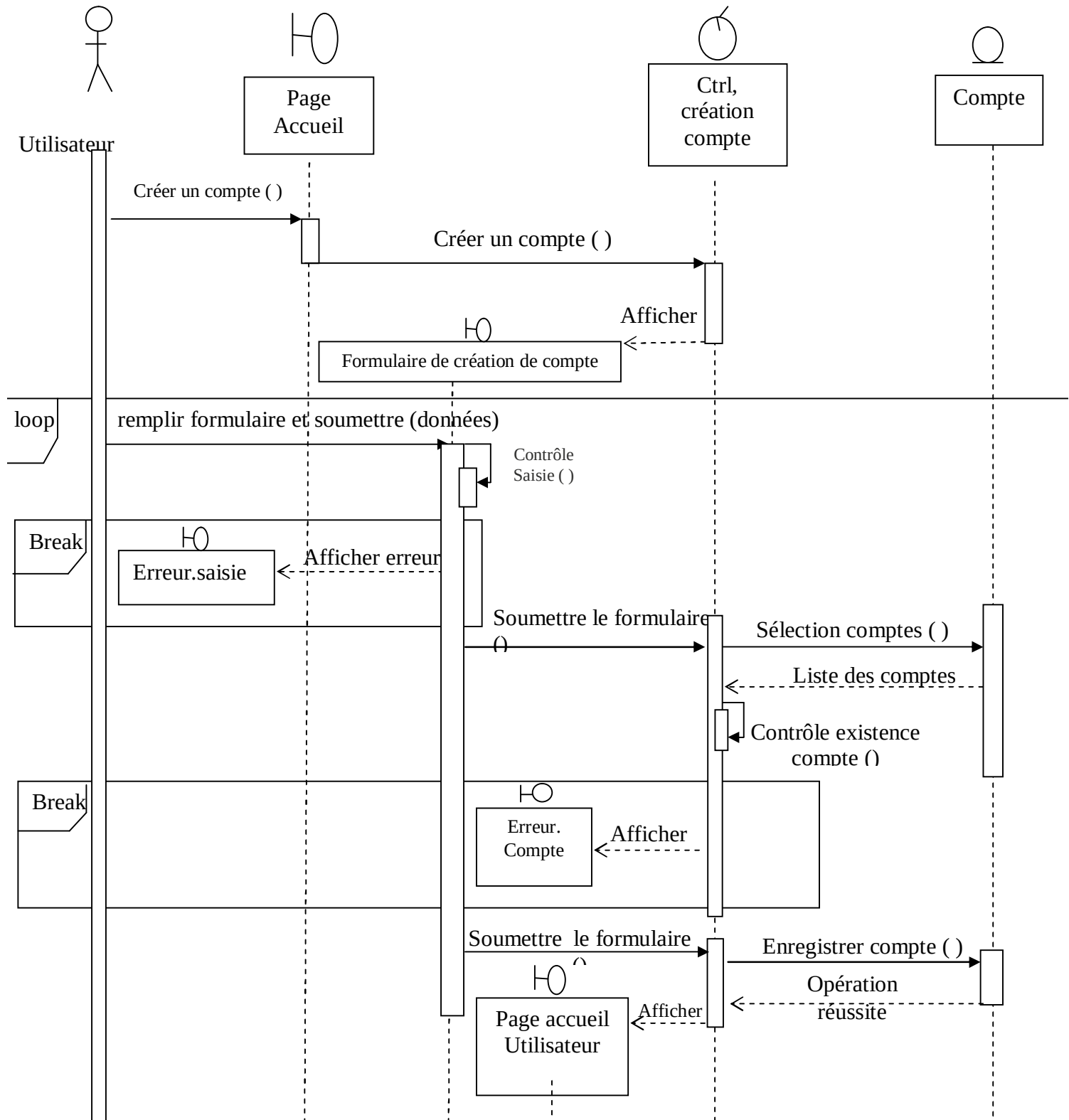


Figure III.4 : Diagramme de séquence du cas d'utilisation « Créer un compte ou Inscription Utilisateur »

Ø Diagramme de séquences « authentification administrateur »

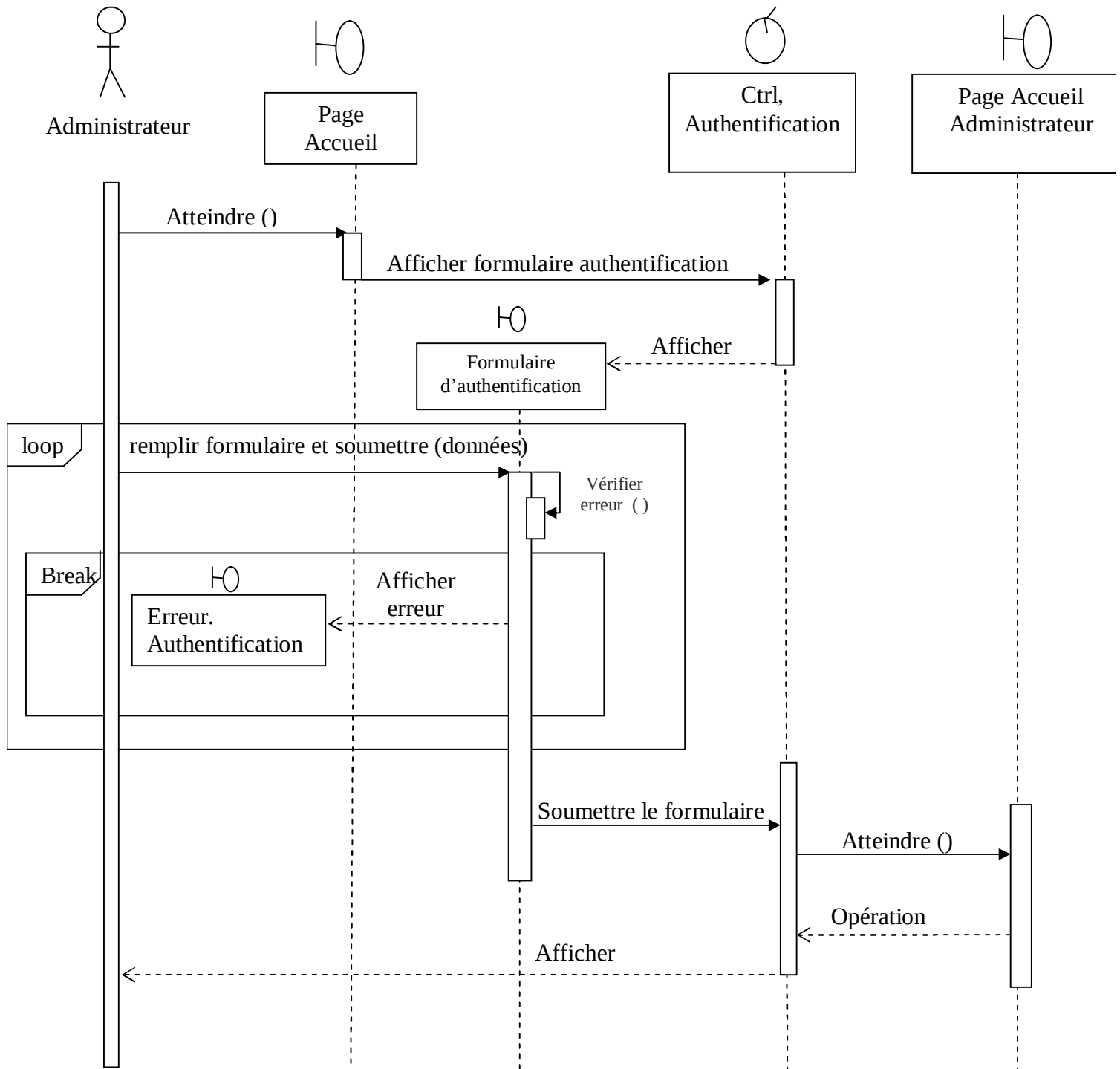


Figure III.5 : Diagramme de séquence du cas d'utilisation « Authentification »

Ø Diagramme de séquences « gérer le papier »

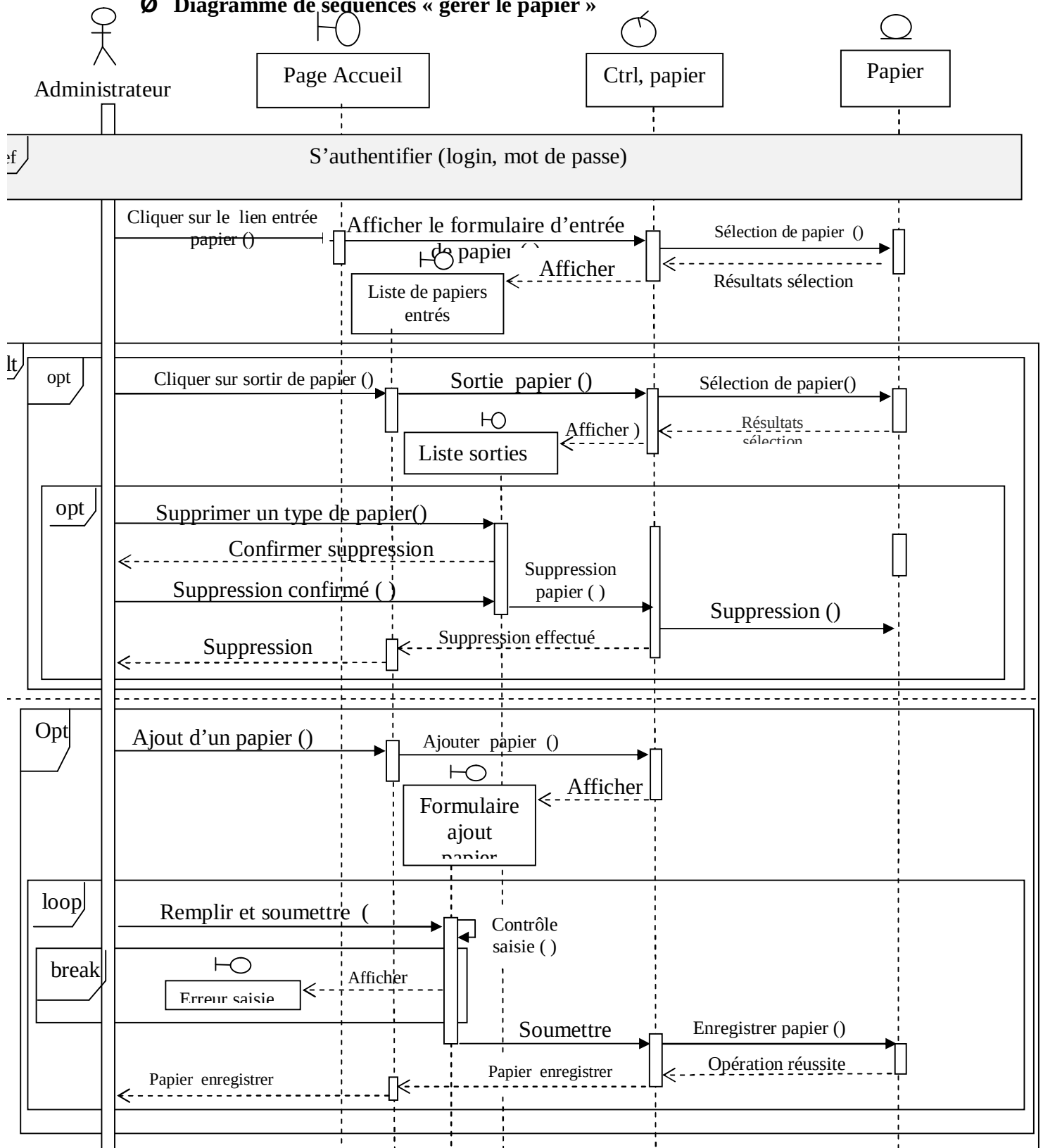


Figure III.6 : Diagramme de séquence du cas d'utilisation « Gérer le papier »

Ø Diagramme de Séquence « gérer les utilisateurs »

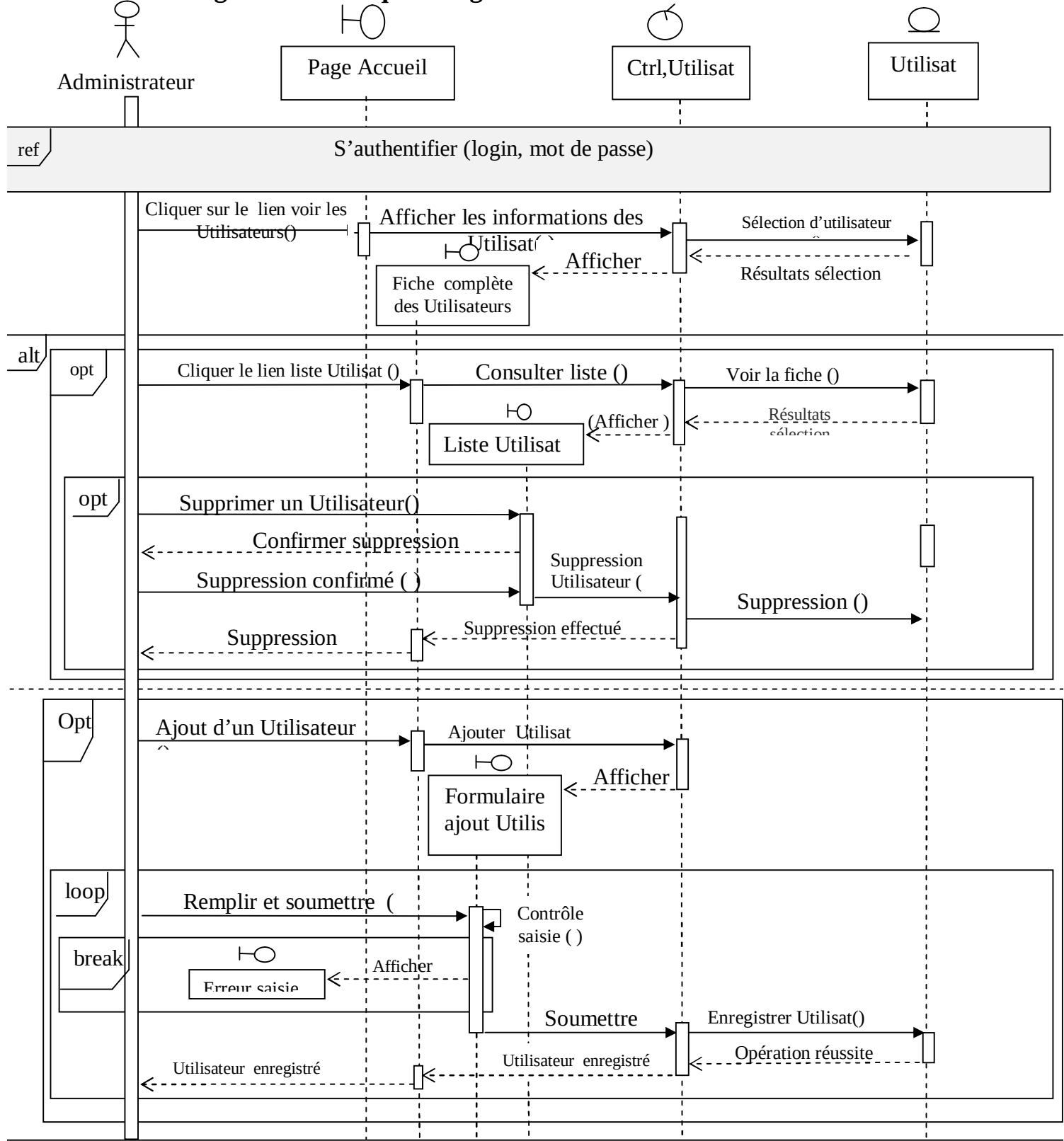


Figure III.7 : Diagramme de séquence du cas d'utilisation « Gérer les Utilisateurs »

B.2 Elaboration des diagrammes de classe

Une fois que les diagrammes de séquence sont élaborés, on passera aux diagrammes de classes. Un diagramme de classes est une collection d'éléments de modélisation statiques qui exprime de manière générale la structure statique d'un système, en termes de classes et de relations entre ces classes. Le but de ce diagramme est d'expliquer ce qu'il faut réaliser plutôt que d'exprimer comment le réaliser.

L'intérêt majeur du diagramme de classe est de modéliser les entités du système.

Ainsi, Il permet de montrer les relations existantes entre les pages client et les pages serveur des cas d'utilisations déjà décrits :

Ø Diagramme de Classe du cas d'Utilisation « S'inscrire Utilisateur ».

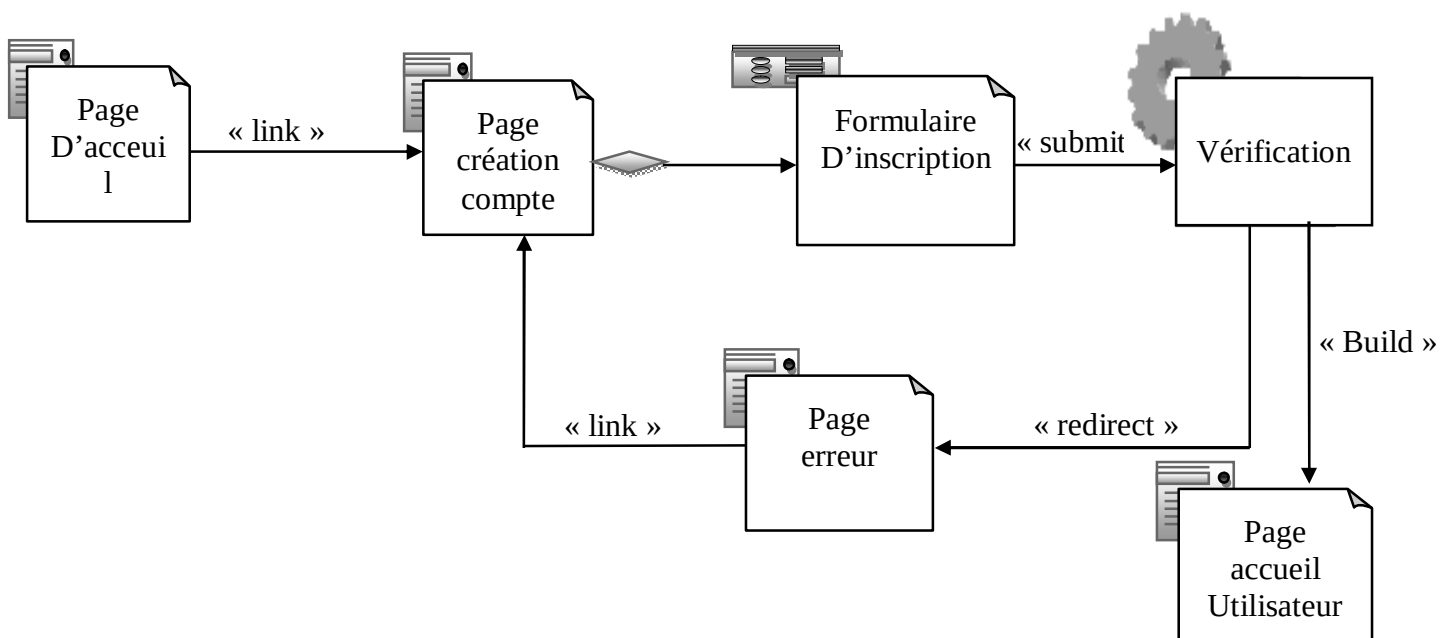


Figure III.8 : Diagramme de Classe du cas d'Utilisation « S'inscrire Utilisateur ».

Ø Diagramme de Classe du cas d'Utilisation « Authentification Utilisateur ».

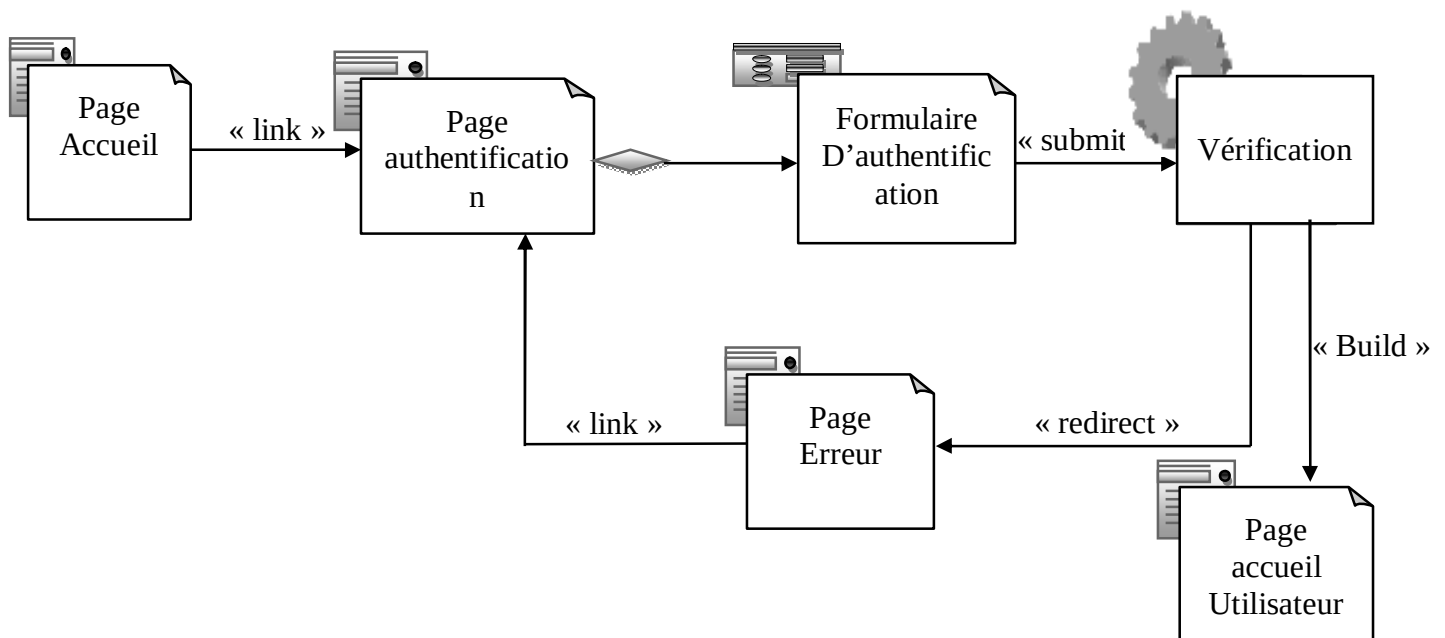


Figure.III.9 : Diagramme de Classe du cas d'Utilisation « Authentification Utilisateur ».

Ø Diagramme de Classe du cas d'Utilisation « faire une commande ».

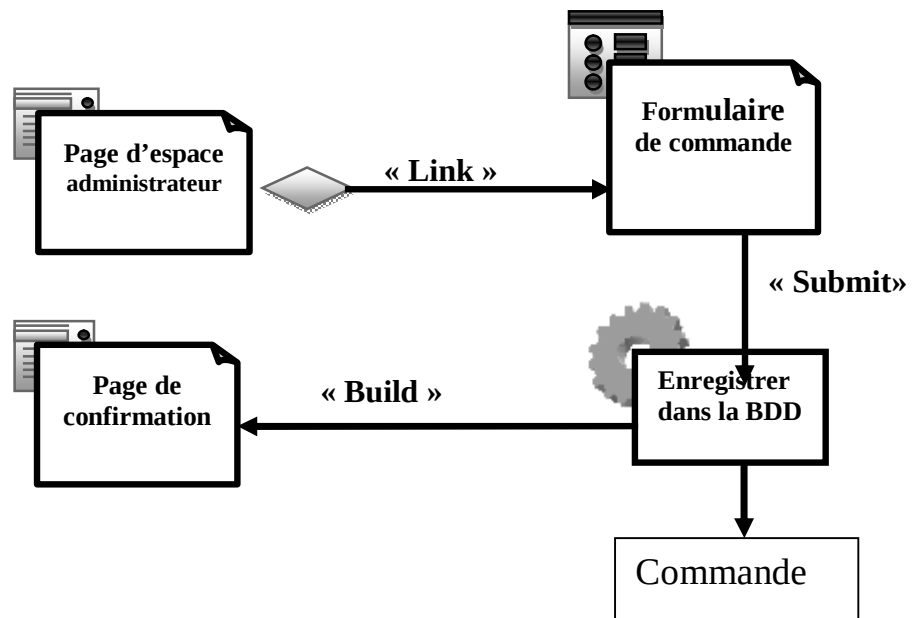


Figure.III.10 Diagramme de Classe du cas d'Utilisation « Supprime Papier ».

Ø Diagramme de Classe du cas d'Utilisation « gérer les Entrées /Sorties Papier ».

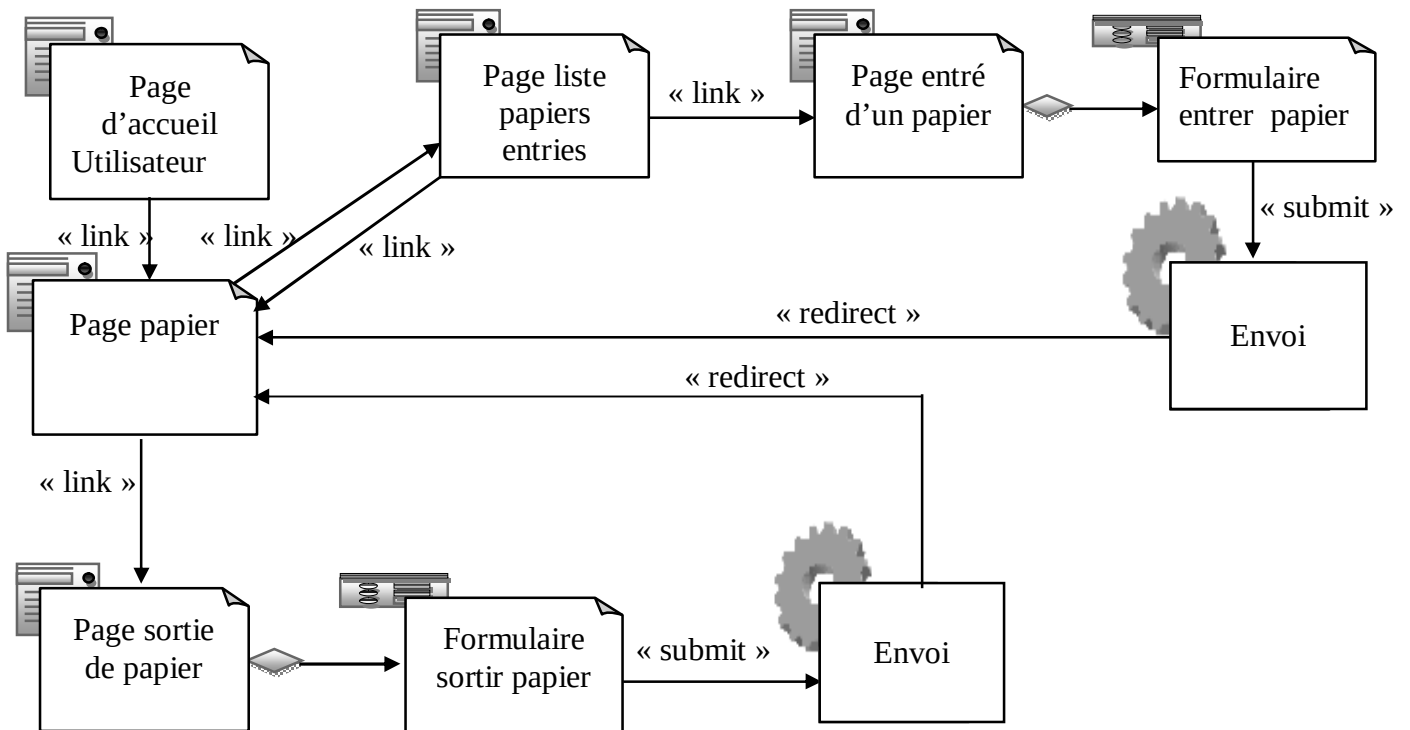


Figure.III.11 Diagramme de Classe du cas d'Utilisation « Gérer les Entrées/Sorties Papier ».

Ø Diagramme de Classe du cas d'Utilisation « gérer les Agences/Services ».

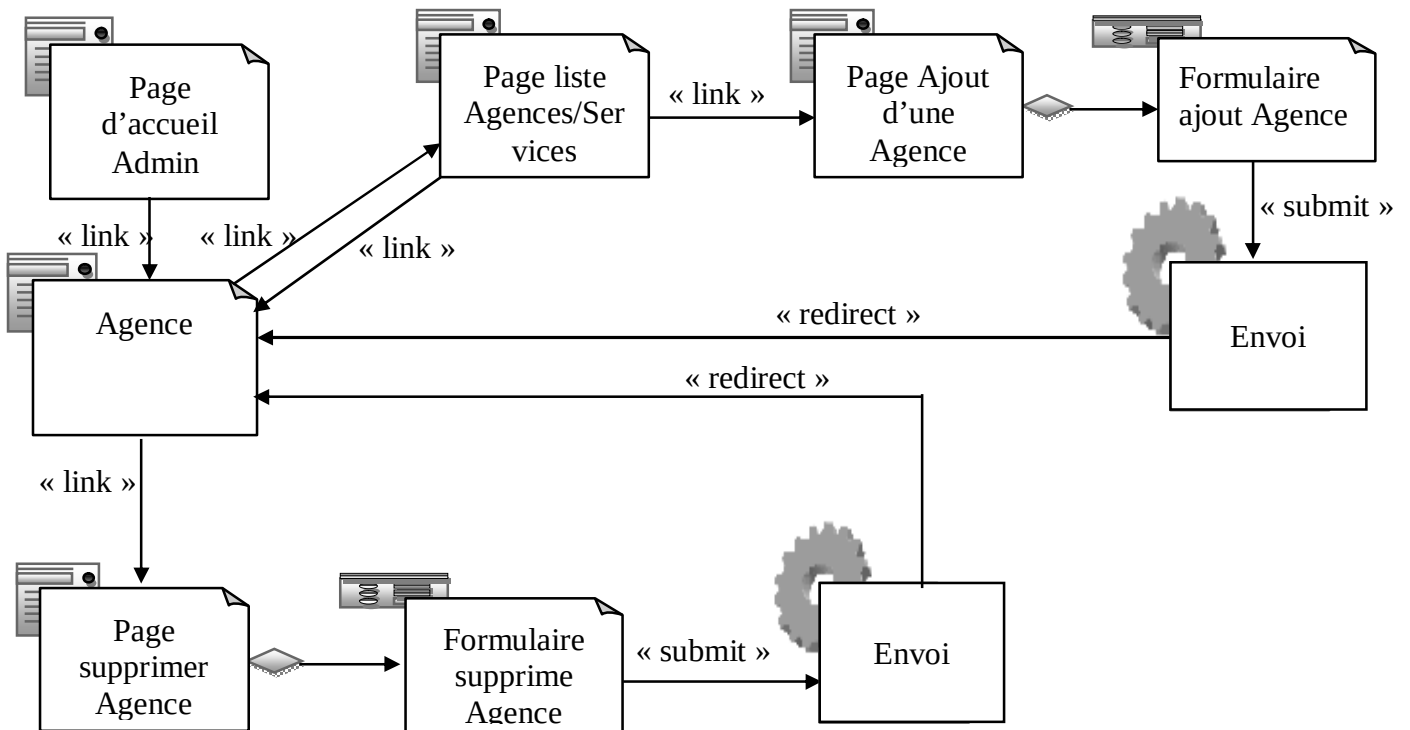


Figure.III.12 Diagramme de Classe du cas d'Utilisation « Gérer les Agences/Services ».

B.3 Diagramme de classes

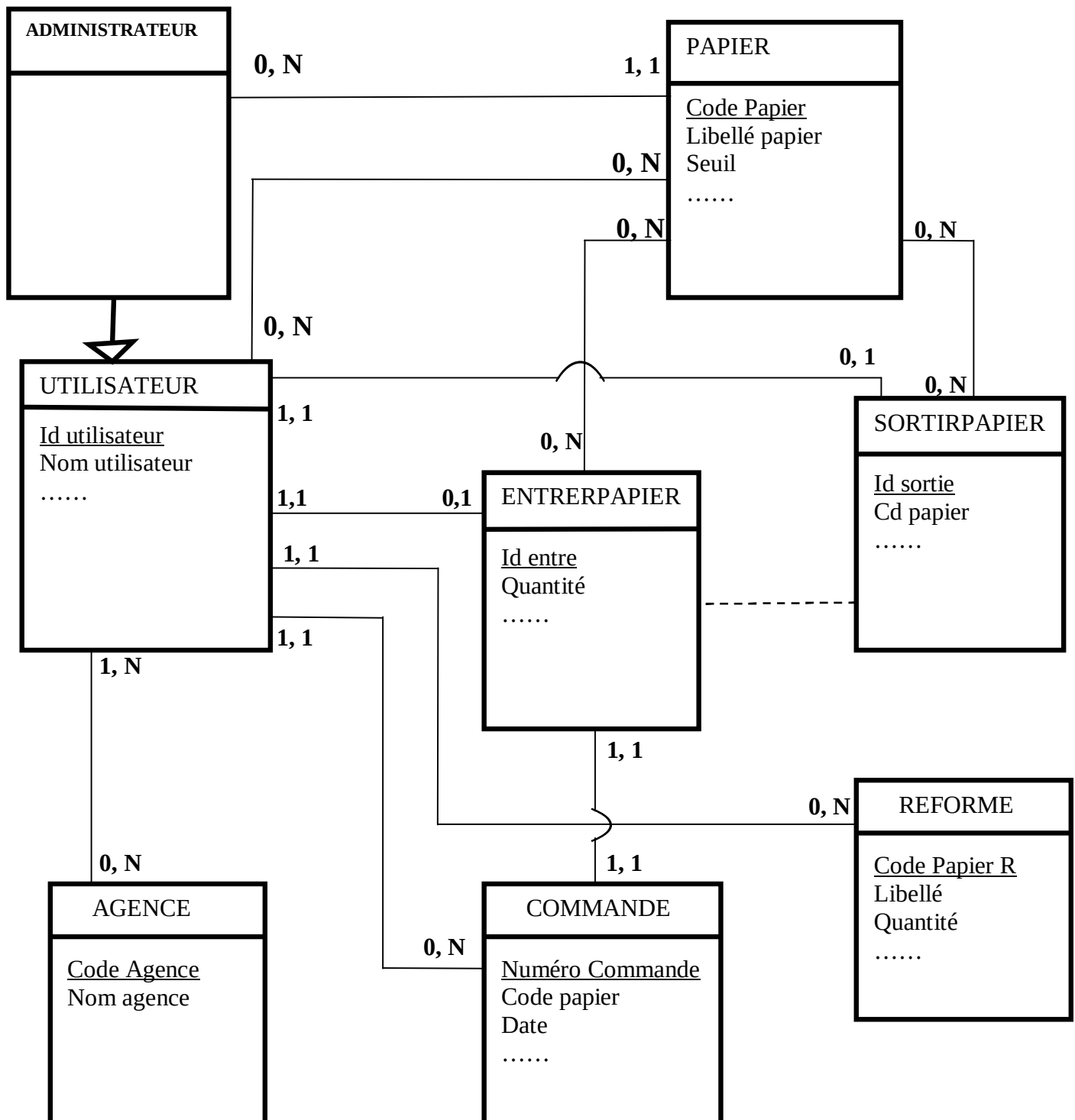


Figure.III.13 Diagramme de Classe Globale

B.4 Le modèle physique des données

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

Ø Les abréviations

- P : clé primaire : identifie d'une façon unique une ligne dans une table ;
- E : clé étrangère : dépend d'une clé primaire située dans une autre table et migrée à partir de cette table ;
- U : unique : index dans lequel deux lignes ne peuvent pas avoir la même valeur.

Ø Description des tables

ü Table Utilisateur

Nom du champ	Code	NULL?	Type	P	E	U
IdUtilisateur	IdUtilisateur	NOTNULL	INT(11)	Oui	Non	Oui
Nom	Nom	NOTNULL	VARCHAR(45)	Non	Non	Non
Prenom	Prenom	NOTNULL	VARCHAR(45)	Non	Non	Non
Login	Login	NOTNULL	TEXT	Non	Non	Oui
Password	Password	NOTNULL	TEXT	Non	Non	Oui

ü Table Papier

Nom du champ	Code	NULL?	Type	P	E	U
CodePapier	CodePapier	NOTNULL	VARCHAR(45)	Oui	Non	Oui
Libelle	Libelle	NOTNULL	VARCHAR(45)	Non	Non	Oui
SeuilPapier	SeuilPapier	NOTNULL	INT(11)	Non	Non	Non
Quantité	Quantité	NOTNULL	INT(11)	Non	Non	Non

ü Table Entrée Papier

Nom du champ	Code	NULL?	Type	P	E	U
IdEntrée	IdEntrée	NOTNULL	INT(11)	Oui	Non	Oui
NumCommande	NumCommande	NOTNULL	VARCHAR(45)	Non	Oui	Oui
CodePapier	CodePapier	NOTNULL	VARCHAR(45)	Non	Oui	Oui
Quantité	Quantité	NOTNULL	INT(11)	Non	Non	Non
NomFournisseur	NomFournisseur	NOTNULL	VARCHAR(45)	Non	Non	Non
Date	Date	NOTNULL	DATETIME	Non	Non	Non
IdUtilisateur	IdUtilisateur	NOTNULL	INT(11)	Non	Oui	Oui
Observation	Observation	NULL	TEXT	Non	Non	Non

ü Table Sortie Papier

Nom du champ	Code	NULL?	Type	P	E	U
IdSortie	IdSortie	NOTNULL	INT(11)	Oui	Non	Oui
CodePapier	CodePapier	NOTNULL	VARCHAR(45)	Non	Oui	Oui
Quantité	Quantité	NOTNULL	INT(11)	Non	Non	Non
NomAgent	NomAgent	NOTNULL	VARCHAR(45)	Non	Non	Non
NomAgence	NomAgence	NOTNULL	VARCHAR(45)	Non	Oui	Oui
Date	Date	NOTNULL	DATETIME	Non	Non	Non
IdUtilisateur	IdUtilisateur	NOTNULL	INT(11)	Non	Oui	Oui
Observation	Observation	NULL	TEXT	Non	Non	Non

ü Table Agence

Nom du champ	Code	NULL?	Type	P	E	U
CodeAgence	CodeAgence	NOTNULL	VARCHAR(45)	Oui	Non	Oui
NomAgence	NomAgence	NOTNULL	VARCHAR(45)	Non	Non	Oui

ü Table Commande

Nom du champ	Code	NULL?	Type	P	E	U
IdCommande	IdCommande	NOTNULL	INT(11)	Oui	Non	Oui
CodePapier	CodePapier	NOTNULL	VARCHAR(45)	Non	Oui	Oui
Quantité	Quantité	NOTNULL	INT(11)	Non	Non	Non
Libelle	Libelle	NOTNULL	VARCHAR(45)	Non	Non	Non
Date	Date	NOTNULL	DATETIME	Non	Non	Non
IdUtilisateur	IdUtilisateur	NOTNULL	INT(11)	Non	Oui	Oui
Observation	Observation	NULL	TEXT	Non	Non	Non

ü Table Réforme

Nom du champ	Code	NULL?	Type	P	E	U
IdReforme	IdReforme	NOTNULL	INT(11)	Oui	Non	Oui
CodePapier	CodePapier	NOTNULL	VARCHAR(45)	Non	Oui	Oui
Quantité	Quantité	NOTNULL	INT(11)	Non	Non	Non
Date	Date	NOTNULL	DATETIME	Non	Non	Non
Observation	Observation	NULL	TEXT	Non	Non	Non

III.4 Conclusion

Dans ce chapitre, nous avons présenté l'analyse et la conception de notre application. Nous avons d'abord présenté l'approche globale de conception que nous avons suivi pour l'élaboration de notre application. Par la suite nous avons cité les fonctionnalités qu'elle offrait aux différents acteurs. Nous nous sommes également appuyé sur les diagrammes offerts par le langage UML afin de nous approfondir dans l'analyse et la conception et ce en modélisant graphiquement certains cas d'utilisation de l'application, et pour conclure ce chapitre une représentation des tables de la base de données utilisée par notre application est nécessaire.

Dans Le chapitre suivant, nous allons définir les outils nécessaires au développement de l'application. Ainsi qu'une présentation des interfaces de l'application réalisée.

IV.1 Introduction

On s'intéressera dans ce chapitre à l'implémentation de notre application et les différentes fonctionnalités qu'elle nous offre ainsi qu'aux différents outils que nous avons utilisés lors de sa réalisation. En effet, nous verrons les langages utilisés ainsi que les différentes technologies utilisées pour son développement (Système d'exploitation, outils de conception web, IDE etc....), et nous aborderons ensuite son fonctionnement et ses différentes interfaces.

IV.2 Environnement et outils de développement

IV.2.1 Le Système d'exploitation Windows 7

Malgré sa récente sortie sur le marché des systèmes d'exploitation, cette dernière version éditée par le géant Microsoft est très appréciée du grand public pour sa rapidité, son efficacité et surtout sa maniabilité.

IV.2.2 IDE NetBeans 6.8

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...etc. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

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

En 1997, NetBeans naît de Xelfi, un projet d'étudiants dirigé par la Faculté de mathématiques et de physique de l'Université Charles de Prague. Plus tard, une société se forme autour du projet et édite des versions commerciales de l'EDI NetBeans, jusqu'à ce qu'il soit acheté par Sun en 1999. Sun place le projet sous double licence CDDL et GPL v2 en juin de l'année suivante.

L'environnement de base comprend les fonctions générales suivantes :

- configuration et gestion de l'interface graphique des utilisateurs,
- support de différents langages de programmation,
- traitement du code source (édition, navigation, formatage, inspection..),
- fonctions d'import/export depuis et vers d'autres IDE, tels qu'Eclipse ou JBuilder,
- accès et gestion de bases de données, serveurs Web, ressources partagées,
- gestion de tâches (à faire, suivi,...etc.).
- documentation intégrée.

L'éditeur intégré propose des fonctions de complétion, de contrôles syntaxiques et sémantiques, d'avertissements et de conseils, de reprise de codes ("refactoring": renommage, changement des méthodes, gestion des classes, ...), de sauvegarde et reprise.

Il supporte les principaux langages suivants :

- Java (Java SE, Java ME JavaFX Java EE), Java doc,
- JSP, JSF
- Ruby et Ruby on Rails: Ruby 1.9, Ruby on Rails 3, JRuby 1.4,
- Groovy et Grails,
- PHP (dont les environnements Zend et Symfony)
- JavaScript C, C++, Fortran. Netbeans ne requiert pas l'utilisation d'un compilateur particulier. A noter le support des bibliothèques Qt. Les plates formes supportées sont Microsoft Windows, Linux, Mac OS, Solaris 10 et OpenSolaris,
- Python,
- HTML, XHTML, RHTML,
- XML,
- DTD,
- CSS,etc.

NetBeans comprend un explorateur de bases de données qui supporte toutes les bases relationnelles pour lesquelles un connecteur JDBC existe (selon les versions des gestionnaires de bases de données): JavaDB (Derby) MySQL, PostgreSQL, Oracle, Microsoft SQL, PointBase, jTDS, IBM Redistributable DB2, ...

L'explorateur comprend un éditeur de requêtes, un gestionnaire intégré de bases de données MySQL.

Ø Applications sur serveurs (applications Web et JAVA EE)

NetBeans supporte une importante variété d'environnements pour l'exécution d'applications web et Java EE: Java Server Pages (JSP), Java Server Faces (JFS), Enterprise JavaBeans (EJB 2.1, EJB 3, EJB 3.1), Apache Struts, Spring Web MVC, Hibernate ... Il supporte les standards Java EE 6, Java EE 5, J2EE 1.4, Java Persistence API (JPA 2.0), Java Servlet API. Il permet le déploiement d'applications Java et d'Applets via Java Web Start (JNLP).

NetBeans intègre les fonctions de Hudson (Maven et projets Java SE pour Ant) et de Maven (créer, exécuter, tester et déboguer des projets Maven Apache).

La figure suivante montre l'interface de principale d'IDE :

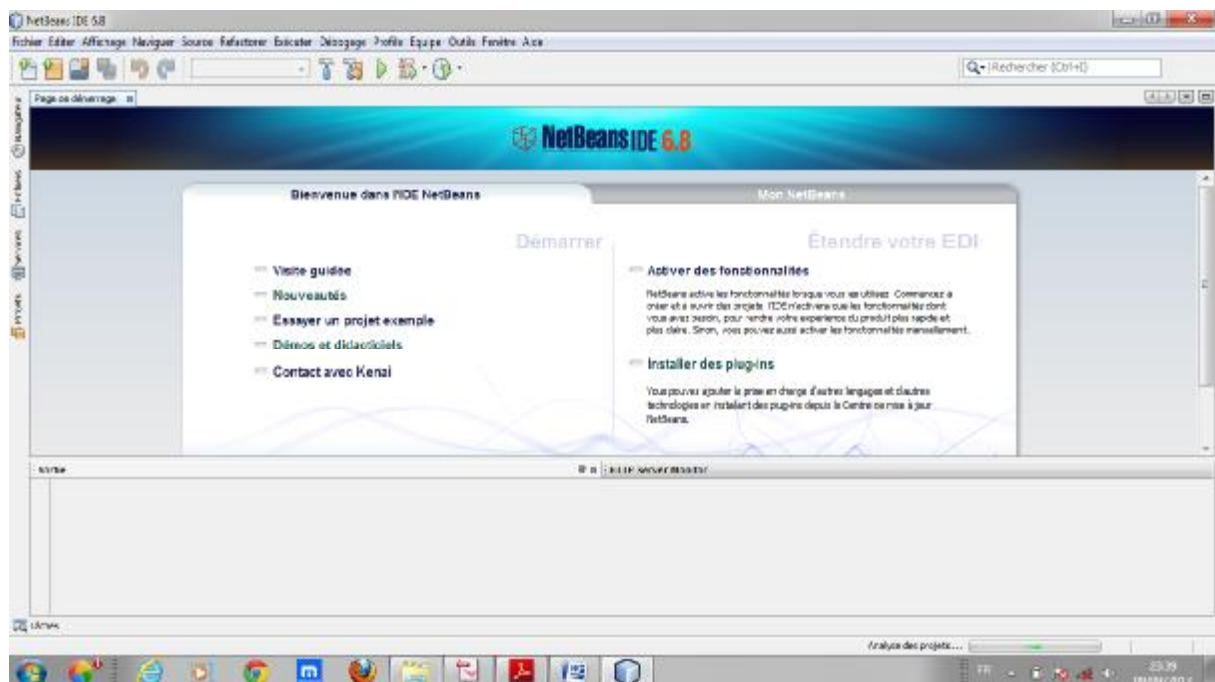


Figure IV .1: Interface Principale d'IDE NetBeans 6.8

IV.2.3 Serveur Web

Pour exécuter des applications Web, il faut disposer d'un serveur. Un serveur Web est un logiciel permettant à des clients d'accéder à des pages web, à partir d'un navigateur (aussi appelé browser). Un serveur Web est donc un simple logiciel capable d'interpréter les requêtes http arrivants sur le port associé au protocole http et de fournir une réponse avec ce même protocole.

Exemple de serveur web :

- Microsoft IIS (Internet Information Serveur)
- Apache
- Microsoft PWS (Personal Web Server)

Notre application est gérée par le serveur Apache.

IV.2.4 Serveur Web (Apache)

Le serveur http apache est le fruit de travail d'un groupe de volontaires, (The Apache Groupe), qui a voulu réaliser un serveur web du même niveau que les produits commerciaux sous forme de logiciels libre. L'équipe d'origine a été rejointe par des centaines d'utilisateurs qui, par leurs idées, leurs tests et leurs lignes de code, ont contribué à faire d'apache le serveur web le plus utilisé dans le monde actuellement. Notre choix pour apache est motivé par les nombreux avantages qu'il présente :

- Extrême portabilité (fonctionne sous les différentes plates-formes UNIX et Windows).
- Modestie des exigences matérielles requises pour un niveau élevé de performance.
- Gratuit.
- Extensible, modulaire et configurable.
- ... etc.

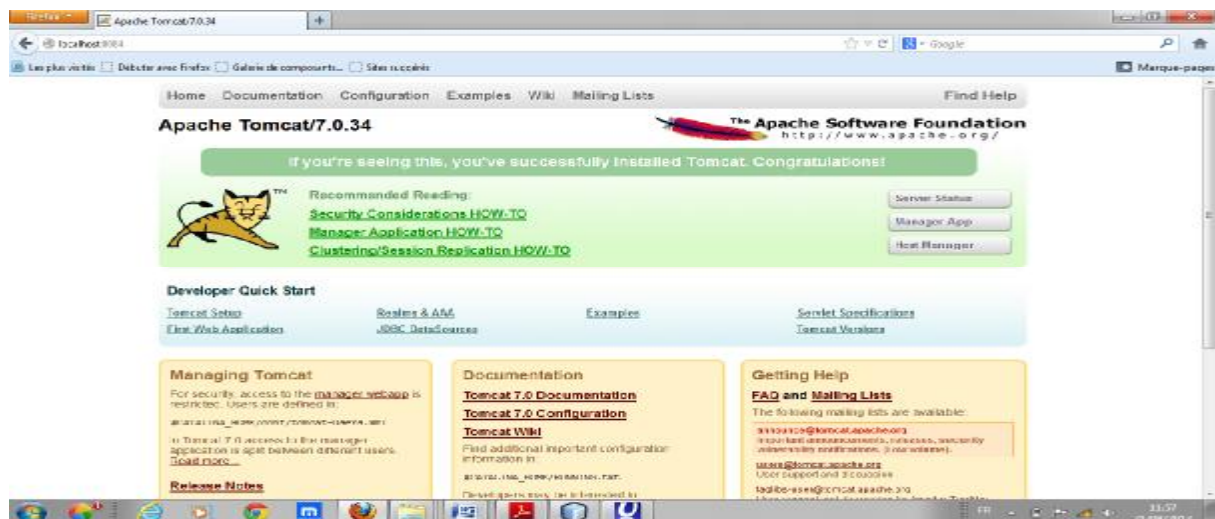


Figure IV .2: Interface d'Apache Tomcat 7.0.34.

IV.2.5 Le conteneur de Servlets Tomcat

Le conteneur de Servlets choisi est le moteur Tomcat 6.0 développé par la fondation

Apache. Le dialogue entre le moteur de Servlets et le serveur web s'effectue à l'aide d'un module logiciel appelé connecteur. Tomcat peut fonctionner sur d'autres serveurs Web mais seul le couple Tomcat/ Apache a été testé. Le module Tomcat du serveur Apache a été développé à partir des sources de Sun Microsystems. Il représente une implémentation de référence pour les Servlets. Tomcat peut fonctionner seul, mais cela n'est pas une solution efficace. En exploitation il est préférable d'associer Tomcat avec un serveur HTTP plus puissant, qui se chargera du contenu statique. Tomcat pourra ainsi être mis à contribution uniquement pour les requêtes mettant en œuvre des Servlets, et nous opterons pour ce principe.

IV.2.6 Présentation de MySQL

Le logiciel MySQL est un serveur de base de données SQL très rapide, multithreads, et robuste. Il est le logiciel open source le plus utilisé dans le monde. Le serveur MySQL est destiné aux missions stratégiques et aux systèmes de production à forte charge, ainsi qu'à l'intégration dans des logiciels déployés à grande échelle. MySQL est une marque déposée de MySQL AB.

C'est une configuration Client –Serveur ce qui consiste en un serveur démon MySQLd, différents programmes clients et des bibliothèques.

MySQL est basé sur une bibliothèque de gestion de données éprouvées depuis de nombreuses années et faisant appel à des index d'arbres binaires. Grâce à cela, le cœur du système peut afficher une performance remarquable, tout particulièrement dans les accès indexés. Les principaux Atouts de MySQL sont la rapidité la robustesse, et la facilité d'utilisation car il est trois à quatre fois plus rapide que la plupart des autres bases de données commerciales et ceci grâce à l'implémentation des fonctions SQL qui s'est faite à travers des classes bibliothèques extrêmement optimisées. , la robustesse, et la facilité d'utilisation.

MySQL est caractérisé par ses fonctionnalités multiples comme le multitraitement en utilisant plusieurs CPU ou bien la diversification de l'outil linguistique qui peut de ce fait afficher au client les messages d'erreurs en plusieurs langues.

Nous avons donc opté pour l'utilisation d'une base de données de type MySQL qui est simple à installer et à utiliser, et pour ce faire nous avons utilisé la version 5.5.20 de MySQL Serveur.

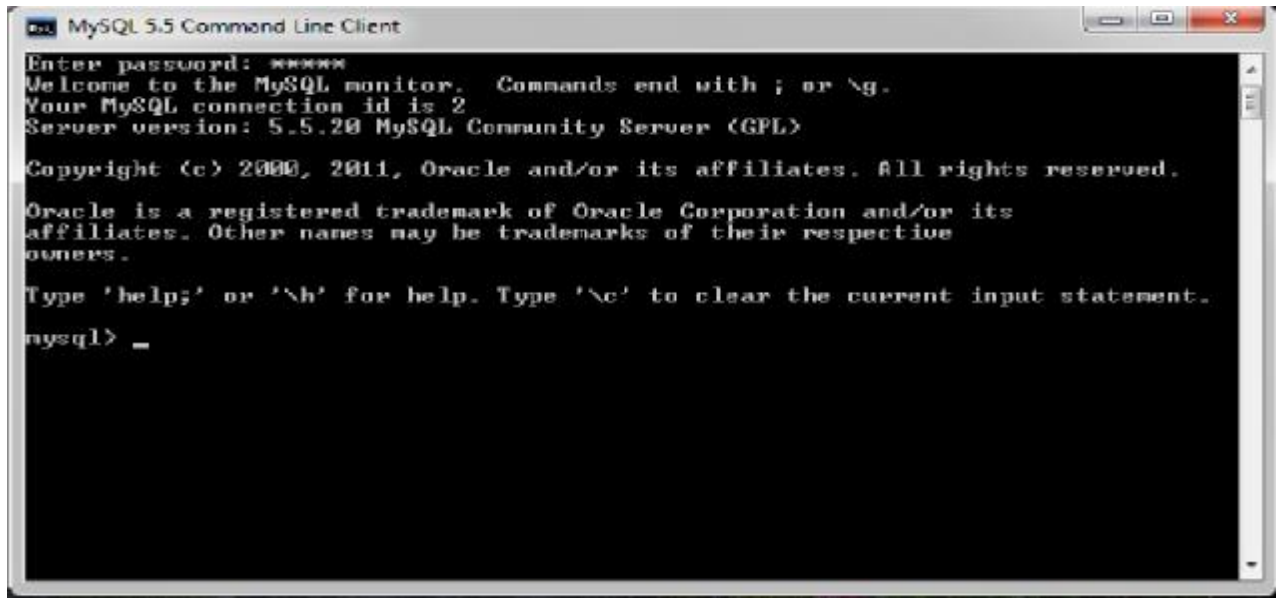


Figure IV .3 : Interface principale MySQL Serveur 5.4

IV.2.7 Le langage de programmation utilisé (Java)

Le langage de programmation que nous avons choisi pour le développement de notre application est le langage JAVA, développé par SUN Microsystems, il est disponible pour les principales plates formes du marché (LINUX, Windows ou autres) et il est totalement gratuit.

JAVA possède de nombreuses caractéristiques :

- Simple du faite que sa syntaxe soit basée sur celle du C++, mais dépouillée de tous les mécanismes complexes, redondants et inutiles (pointeurs,...).
- Performant, puissant, java est une plate forme de développement comportant une bibliothèque de classes très riche et de nombreux outils d'interfaces de programmations applicatifs (API).
- Interprété, portable et indépendant des architectures matérielles : Cette caractéristique est un avantage primordiale pour java face à des applications transmises par un réseau et exécutées sur des machines hétérogènes. Un programme java est successivement compilé pour fournir un code intermédiaire indépendant de la plate forme d'exécution (le byte code) simple et rapide à traduire en langage machine.
- Riche : Un des aspects importants de l'environnement java est la richesse de ses librairies. Java permet de développer de nombreux sortes de programmes dont :

- ü Des applications, sous forme de fenêtre ou de console.
- ü Des applets, qui sont des programmes JAVA incorporés à des pages web.
- ü Servlets et JSP pour le développement d'applications web.

Dans notre mémoire nous avons choisi pour le développement de l'application le J2EE (Java 2 Entreprise Edition) : développement des applications Web en JAVA grâce aux Servlets et JSP.

IV.2.8 Les Servlets

Une servlet est un programme java qui fonctionne sur un serveur Web compatible J2EE et dont le rôle est d'apporter une réponse à une requête. Elle reçoit une requête du client, elle effectue des traitements et renvoie le résultat. La technologie des Servlets n'est qu'un ensemble de classes. Pour fonctionner convenablement elles ont besoin d'une machine virtuelle JAVA et de l'ensemble des autres classes intégrées à l'API standard du langage JAVA. Une Servlet est une application développée dans un contexte client-serveur.

Voici un exemple de code d'une servlet extrait de la page « Authentification Administrateur » qui permet la connexion pour un Administrateur :

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package servlets;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 *
 * @author micromedia
 */
public class adminServ extends HttpServlet {

    /**
     * Processes requests for both HTTP GET and POST methods.
     * @param request servlet request
     * @param response servlet response
     */
}
```

```
* @throws ServletException if a servlet-specific error occurs
* @throws IOException if an I/O error occurs
*/
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    String username = request.getParameter("username");
    String password = request.getParameter("password");
    if((username.equals("admin"))&&(password.equals("admin")))
    {
        HttpSession session=request.getSession();
        session.setAttribute("id", "admin");
        out.println("<h1>BIENVENNUE A LA PAGE ADMINISTRATEUR</h1>");
        request.getRequestDispatcher("/Administrateur.jsp").include(request, response);
    }else
    {
        out.println("<h1> Administrateur Non Valide ...</h1>");
        request.getRequestDispatcher("/loginAdmin.jsp").include(request, response);
    }

    try {
        /* TODO output your page here
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet adminServ</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Servlet adminServ at " + request.getContextPath () + "</h1>");
        out.println("</body>");
        out.println("</html>");
        */
    } finally {
        out.close();
    }
}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the
left to edit the code.">
/**
 * Handles the HTTP <code>GET</code> method.
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    processRequest(request, response);
}
```

```
/**
 * Handles the HTTP <code>POST</code> method.
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>
}
```

IV.2.9 La notion de JSP

JSP = Java Server Pages, est un fichier contenant du code HTML et des fragments de code Java exécutés sur le moteur de Servlets Comparable aux langages côté serveur de type PHP, ASP, ...etc.

Les JSP permettent d'introduire du code Java dans des tags prédéfinis à l'intérieur d'une page HTML. Elles permettent donc de mélanger la puissance de Java côté serveur et la facilité de mise en page d'HTML côté client. Les fichiers JSP possèdent par convention l'extension (.jsp).

Le code suivant est le code source de la JSP de la page d'accueil de l'application :

```
<%--
  Document   : index
  Created on : 29 mai 2013, 18:20:21
  Author      : micromedia
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <link rel="stylesheet" media="screen" type="text/css" href="lestyle.css" />
    <title>Page d'accueil </title>
    <script type="text/javascript">
//----- LOCALIZEABLE GLOBALS-----
var d=new Date();
monthname=
Array("Janvier","Février","Mars","Avril","Mai","Juin","Juillet","Août","Septembre","Octobr
e","Novembre","Décembre");
//Ensure correct for language. English is "January 1, 2004"
var TODAY = monthname[d.getMonth()] + " " + d.getDate() + ", " + d.getFullYear();
//----- END LOCALIZEABLE -----
    </script>
  </head>
  <body>
    <div id="conteneur">
      <div id="haut">
        
        <tr bgcolor="CCFF99">
          <td colspan="7" id="dateformat" height="25">
            <script language="JavaScript" type="text/javascript">
              document.write(TODAY);
            </script>
          </td>
        </tr>
        <marquee><h2><font color="green"> Bienvenue Sur Le Site Web De Gestion
</font></h2></marquee>
      </div>
      <div id="navMenu" align="center">
        <ul>
          <li><a href="#">Administrateur </a>
          <ul>
            <li><a href="loginAdmin.jsp">Accée Administrateur </a></li>
          </ul>
          </li>
          <ul>
            <li><a href="#">Utilisateur </a>
            <ul>
              <li><a href="LogUtilisateur.jsp">Acceder Utilisateur</a></li>
              <li><a href="Inscription.jsp">S'inscrire</a></li>
            </ul>
            </li>
          </ul>
        </ul>
      </div>
    </div>
  </body>
</html>

```

```
<li><a href="#">Liens </a>
<ul>
<li> <a href="www.sdc.dz">www.sdc.dz </a></li>
</ul>
</li>
</ul>
<ul>
<li><a href="#">Quiter l'application </a>
<ul>
<li> <a href="#">Exit </a></li>
</ul>
</li>
</ul>
</div>
<div id="melieu">

<p>voici un peut d'ecriture </p>

</div>
</body>
</html>
```

IV.2.10 JavaScript

C'est un langage de scripts principalement utilisé dans les pages web interactives.

JavaScript fait ce que HTML ne peut pas. Il permet de traiter les saisies de l'utilisateur faites au clavier ou avec la souris et d'en réagir avec des sorties à l'écran ou des modifications dynamiques dans la page

Web affichée. L'utilisateur introduit donc quelques valeurs, JavaScript fait les calculs et affiche le résultat sans qu'il y ait d'échanges entre le client et le serveur web.

IV.3 Présentation de quelques interfaces et fonctionnalités de l'application

Nous allons vous présenter dans cette partie les interfaces principales de notre application, pour mieux vous expliquer le manuel d'utilisation de l'application.

IV.3.1 L'interface principale « page d'accueil »

L'interface d'accueil, contient les liens nécessaires et indispensables pour la bonne exploration de l'application par les utilisateurs.

Cette page contient un menu dont les rubriques suivantes :

1. Administrateur : contient lien de connexion pour l'administrateur.
2. Utilisateur : contient deux liens :
 - Accéder utilisateur (pour un utilisateur inscrit et validé par l'administrateur).
 - S'inscrire (pour un nouvel utilisateur)
3. Liens : contient l'adresse internet de la direction de distribution du centre (www.sdc.dz).
4. Quitter l'application : pour sortir de l'application.

La figure suivante montre la «page d'accueil » de l'application



Figure IV.4 : Page d'accueil principale

IV.3.2 La Page « Authentification Administrateur »

C'est la page qui permet d'accéder à l'interface générale administrateur, elle contient un formulaire d'authentification, un champ pour le nom d'administrateur et un autre champ pour le mot de passe, et deux boutons « Valider » pour accéder a son interface « Administrateur », et « Annuler » pour revenir a la « page d'accueil principale ».

La figure suivante montre cette page « d'authentification administrateur ».

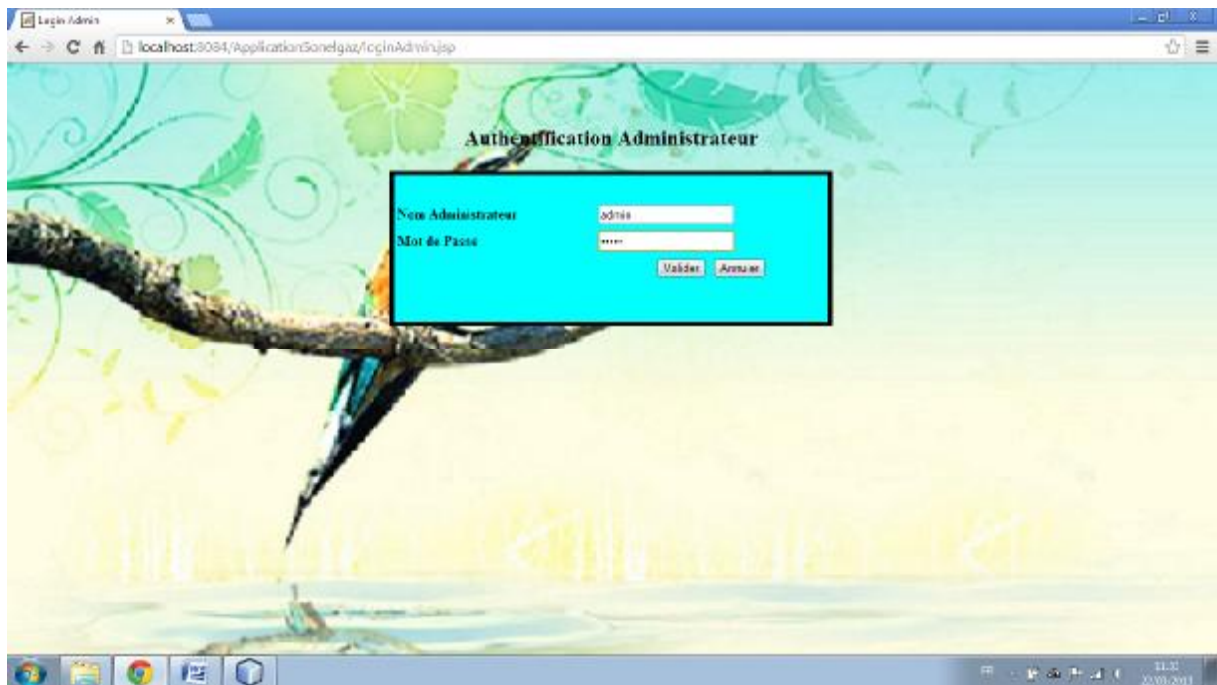


Figure IV.5 : Page d'Authentification Administrateur.

IV.3.3 L'interface Principale Administrateur

Cette page contient un menu avec plusieurs rubriques :

1. Gestions du papier : cette rubrique contient plusieurs liens :
 - Ajout un nouveau papier : pour ajouter un nouveau type de papier.
 - Supprimer papier : pour la suppression de papier.
 - Liste du papier : pour la consultation de liste complète de papier.
 - Entrer du papier : pour enregistrer le papier arriver a la DGSi et de le stocker.
 - Sortir du papier : pour sortir du papier de stock.
 - Consultation (stock) : pour consulter le papier existant dans le stock.
 - Reforme de papier : pour reformer un papier non utilisé dans le stock.
 - Supprimer un papier reforme : pour la suppression du papier reformés.
 - Liste du papier reformes : pour consulter la liste du papier reformés.
2. Gestions des commandes : contient les liens suivants :
 - Faire une commande : pour faire une nouvelle commande du papier.
 - Imprimer commande : pour l'impression d'une commande.
 - Liste des commandes : pour consulter la liste des commandes.
 - Supprimer une commande : pour la suppression d'une commande erronées.
3. Gestions des agences/services : cette rubrique contient les liens suivants :
 - Ajout d'une agence/service : pour ajouter d'une nouvelle agence/service dont on fait la sortie du papier.
 - Supprimer une agence/service : pour la suppression d'une agence/service.
 - Liste agences/services : pour consulter toutes les agences/services.
4. Historique sur le papier : contient deux liens :
 - Papier en entrée : pour la consultation de la fiche complète d'entrée du papier.
 - Papier en sortie : pour la consultation de la fiche complète de sortie du papier.
5. Gestion des utilisateurs : cette rubrique contient des liens suivants :
 - Validation compte : liens pour valider un utilisateur inscrit et non validé encore par l'administrateur.
 - Liste utilisateur : liens pour consulter les utilisateurs de l'application.
 - Supprimer utilisateur : liens pour la suppression d'utilisateurs.
6. Quitter : rubrique contient un lien pour la quitter la page administrateur et revenir a la page principale d'accueil.

La figure suivante montre la page « interface administrateur » .



Figure IV.6 : l'interface générale Administrateur.

IV.3.4 La page d'ajout d'un nouveau type de papier

Cette page permet à l'administrateur d'ajouter un nouveau type de papier », en précisant le code du papier, son libellé (nom) ainsi que son seuil.

La figure suivante montre la page « ajout d'un nouveau type de papier ».

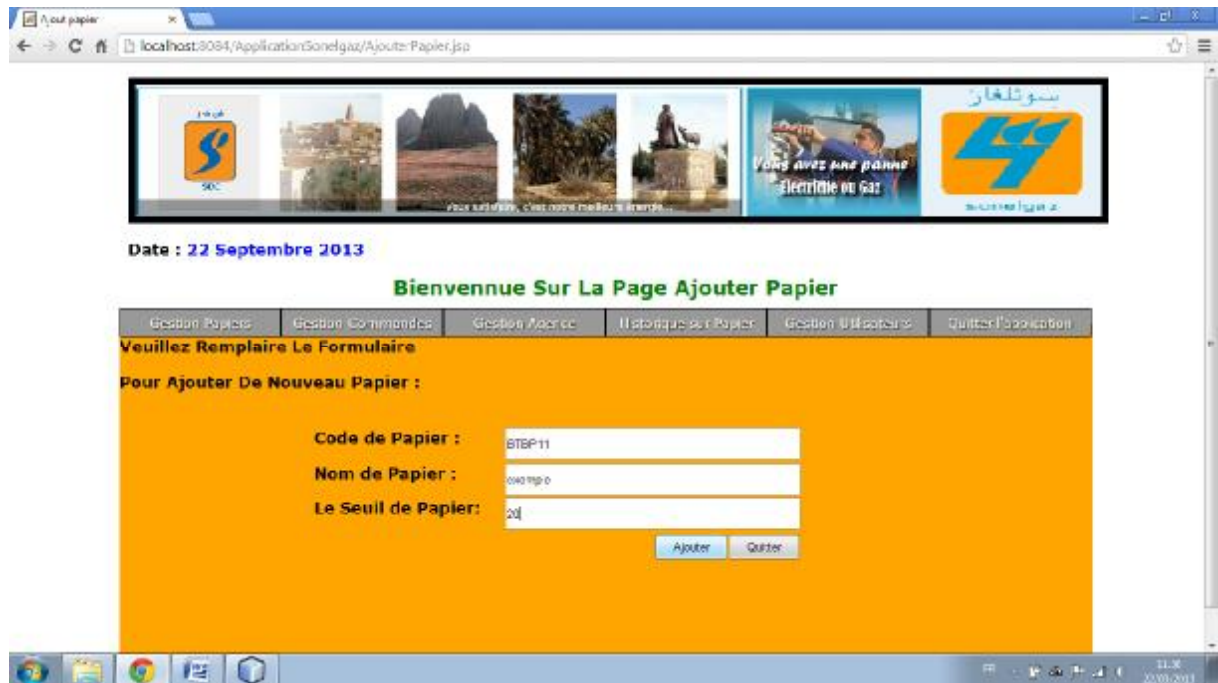


Figure IV.7 : Page d'Ajout d'un nouveau type de papier.

IV.3.5 La page Supprimer de papier

Elle permet à l'administrateur de supprimer un papier donné, et cela en choisissant (sélectionnant) le code du papier, puis en cliquant sur le bouton « supprimer ».

La figure suivante montre la page « supprimer de papier ».

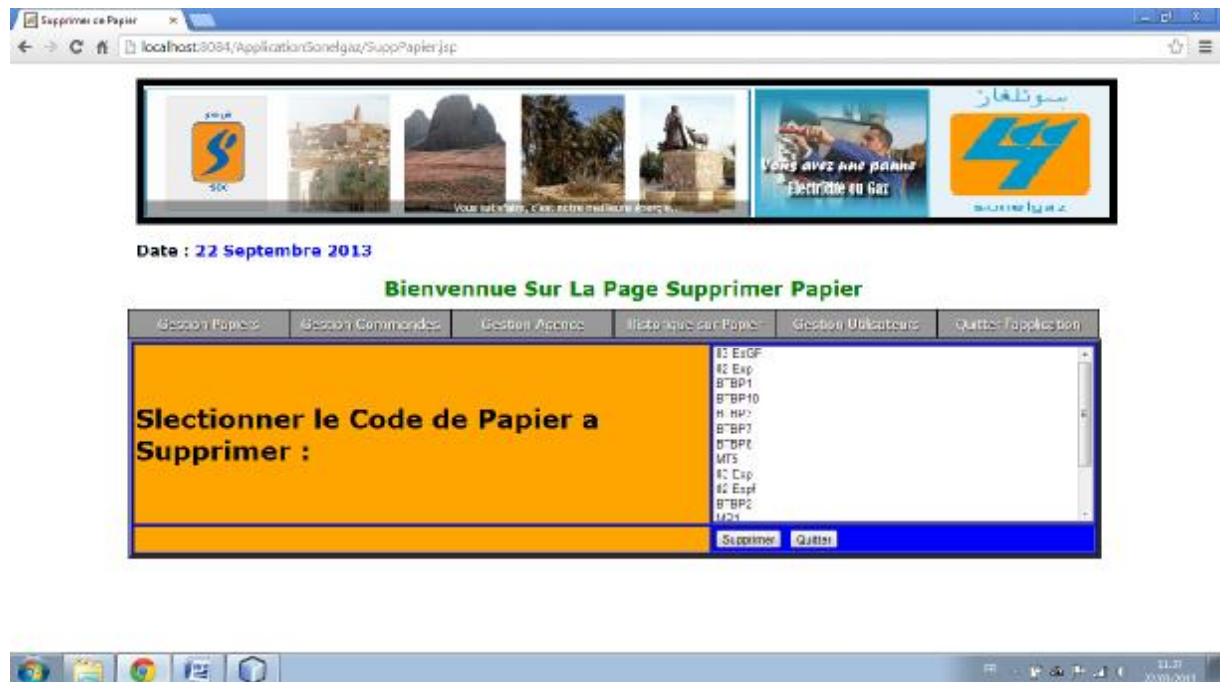


Figure IV.8 : Page Supprimer de papier.

IV.3.6 La page Consultation de Stock

Cette page permet aux utilisateurs ou administrateur de visualiser la liste complète du papier existante au niveau de stock, cette liste contient le code du papier, son nom, sa quantité ainsi que son seuil minimum.

La figure suivante montre la page « consultation de stock »



Date : 22 Septembre 2013

Bienvenue Sur Le Page Consultation Stock

Code Papier	libelle(NomPapier)	Quantite	Stock
03 ExGP	Listing 03 Exemplaires(G.F)	200	50
02 Exp	Listing 02 Exemplaires(G.F)	400	40
BTBP1	Facture Abonnes Ordinaires	300	60
BTBP10	Facture Devis	700	30
BTBP3	Facture Abonnés Domiciles	150	50
BTBP2	Facture Memoires FGM	470	40
BTBP8	Facture Lettres Acquit	520	50
MT5	Fiches de relevé	340	20
03 Exp	Listing 03 Exemplaires(P.F)	260	10
02 Expt	Listing 02 Exemplaires(P.F)	500	15
BTBP2	Factures D'Avaloir	400	20
MP1	Factures Moyenne Pression	330	15
MT1	Factures Moyenne Tension	700	12
MT6	Traites MT/MP	120	25

Imprimer Quitter

Figure IV.9 : Page Consultation de Stock.

IV.3.7. La page Entrer de papier

Elle permet aux utilisateurs de faire enregistrer le papier en entrant au stock, cette page contient un formulaire avec des champs suivant :

- Numéro de commande.
- Code du papier.
- La quantité du papier entré.
- Nom d'agent qui a fourni le papier
- Date de l'opération
- Et le nom d'utilisateur de l'application comme un trace pour cette opération.

La figure suivante montre la page « entrer de papier ».

The screenshot shows a web browser window with the address bar displaying 'localhost:8084/ApplicationSone/gaz/PapierEntree.jsp'. The page features a header banner with logos and images, including a logo with the number '50' and a logo with the number '49'. Below the banner, the text 'Utilisateur : admin' and 'Date : 22 Septembre 2013' are displayed. The main heading is 'Bienvenue Sur La Page Entrer Papier'. A navigation bar contains the following links: 'Gestion Papiers', 'Gestion Commandes', 'Gestion Adresse', 'Historique sur Papier', 'Gestion Utilisateurs', and 'Quitter l'application'. The main form area has an orange background and contains the following fields and labels:

- Numero De Commande : C001
- Sélectionner Code Papier : 03 ExGF
- La Quantité : 1000
- Livrer Par : 1780001
- Ajouter une Observation : 010

At the bottom of the form are two buttons: 'Entrer' and 'Quitter'.

Figure IV.10 : Page Entrée de Papier.

IV.3.8 La page Historique d'Entrer de papier

Elle permet à l'administrateur de faire une consultation complète sur le papier en entré, cette page contient des liens pour faire un choix de consultation :

1. Par utilisateur : pour savoir les utilisateurs qui ont fait l'opération entré du papier.
2. Par numéro du commande : consultation par le numéro du commande.
3. Par papier : consultation les entrées par le code du papier.
4. Par fournisseur : visualiser les entrées du papier par le nom d'agent qui a fait la fourniture de papier.
5. Par date : consultation mensuelle des entrées du papier.

La figure suivante montre la page « historique entré de papier »



Figure IV.11 : Page Historique Papier Entrée.

IV.3.9 La page Historique Sortie de papier

Elle permet à l'administrateur de visualiser la fiche complète des sorties du papier vers les différentes agences / services, cette page contient des liens suivants :

1. Par agence : pour la consultation des sorties par agence/service.
2. Par papier : pour faire la consultation des sorties par type du papier.
3. Par date de sortie : consultation des sorties par mois (consultation mensuelle).
4. Par utilisateur : consultation des sorties du papier par les utilisateurs de l'application.

La figure suivante montre la page « historique sortie de papier »



Figure IV.12 : Page Historique Sortie de Papier.

IV.4 Conclusion :

Au niveau de ce chapitre nous avons présenté l'environnement de développement et l'implémentation de notre application web, puis nous avons décrits quelques interfaces pour l'utilisation d'application et à la fin, nous avons montré un peu le manuel d'utilisation pour avoir une idée générale sur les tâches de notre applications.

Conclusion générale

Toute organisation, doit suivre l'évolution de la technologie et se mettre à jour pour trouver des solutions efficaces à des systèmes informatiques déjà existants, afin d'améliorer le rendement, la qualité de service et le temps de réponse et de satisfaire la clientèle qui ne cesse d'augmenter ainsi que la bonne gestion de son travail et éviter de tomber dans des situations critiques.

L'étude et le développement de ce projet informatique nous ont permis, non seulement de nous familiariser avec le domaine professionnel (notamment les entreprises algériennes), mais aussi d'acquérir une certaine maîtrise, à savoir les outils de développement tels que NetBeans, le serveur web Apache (Tomcat) et le serveur de base de données MySQL, coté conception orienté objet nous avons opté pour le langage UML, tandis que pour la réalisation nous avons utilisé le langage de programmation JAVA web (J2EE) (Servlets, JSP) pour l'application web, ainsi JavaScript et SQL.

Pour réaliser notre projet nous l'avons décomposé en deux parties :

- Ø Etudier l'application existante qui est été développé en Delphi et analyser ces erreurs, ces défaillances et ces failles de gestion.
- Ø Création d'une nouvelle application web J2EE qui fait la bonne gestion de papiers et les transactions (sortie et entrée de papier au niveau de Stock) ainsi que la réforme de l'ancien papiers non utilisable qui est dans le stock et faire un historique complet sur l'entée et la sortie de papiers pour éviter les situations critiques pour la consommation de différents types de papiers.

En effet, notre application offre plusieurs services et rend facile la tâche de la gestion de papiers, et offre la possibilité de contrôler les utilisateurs de l'application.

Pour en finir, nous espérons avoir répondu aux attentes et aux exigences de l'entreprise, et que l'application informatique réalisée sera utile pour ses utilisateurs.

BIBLIOGRAPHIE

Mémoires

[ING INF 2009/2010] AOUDIA.AMINA-BELHOCINE.THANINA, *Mise en oeuvre d'un middleware métier J2EE pour la DGS/SONELGAZ*, mémoire d'ingénieurs, Université MOULOUD Mammeri, département d'informatique.

Livres & document

Livre **à UML pour la modélisation d'un système d'information** [le cahier des charges du maître d'ouvrage] chantal Morley, Jean Hugues, Bernard Leblanc..... 2^e édition. Mai 2002.

Document **à L'orienté objet, cours et exercices en UML**, fichier PDF créé par Hugues Bersini, Ivan Wellesz et publié par Groupe Eyrolles. 2002.

Document **à Méthodologie des systèmes d'information – UML**, fichier PDF créé par DI GALLO Frédéric.2000-2001

Document **à Comparaison des architectures J2EE et .NET**, Jean-Philippe FORESTIER. Copyright OSYX 2003

Cours **à NFE107 Urbanisation & Architecture des Systèmes d'Information**, Leonardo AMODIO. (Conservatoire Nationale des Arts et Métiers, Centre d'enseignements de Grenoble, Année Universitaire: 2008-2009)

WEBLIOGRAPHIE

[http://www. Introduction aux réseaux informatiques, modèle OSI et TCP_IP.htm](http://www.Introduction%20aux%20réseaux%20informatiques,%20modèle%20OSI%20et%20TCP_IP.htm)

http://www.ybet.be/hardware2_ch2/hard2_ch2.htm

[http://www. uml.free.fr](http://www.uml.free.fr)

<http://www.sdc.dz>

[http://www. Sonelgaz Spa.dz](http://www.Sonelgaz%20Spa.dz)

Des cours et articles intéressants :

<http://www.developpez.com>