

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique
Université A. MIRA de Béjaïa



Faculté Des Sciences Exactes
DEPARTEMENT D'INFORMATIQUE

Projet de fin de cycle

En vue de l'obtention du diplôme de master professionnel
Option : Administration et sécurité des réseaux

Thème

**Conception et réalisation d'un système de messagerie
électronique en intranet**

Cas d'étude : Cevital Agroalimentaire, Béjaïa.

Réalisé par :

BOURDACHE Katia.
OUALI Siham.

Soutenu le 02-07-2016 devant le jury composé de :

| | |
|---|-------------------------------|
| <u>Président</u> : M ^f NAFI Mohammed | Université A. MIRA de Béjaïa. |
| <u>Examineur</u> : M ^f ATMANI Mouloud | Université A. MIRA de Béjaïa. |
| <u>Encadrant</u> : M ^f LARBI Ali | Université A. MIRA de Béjaïa. |

Année universitaire 2015-2016

Remerciements

Nous tenons à remercier notre promoteur Monsieur LARBI Ali, et lui exprimer notre profonde gratitude pour sa disponibilité, ses orientations et ses conseils.

Nous remercions également le directeur de système d'information de CEVITAL, Monsieur Ounsser Farouk et notre Co-encadrant Monsieur AIT IKHLEF Nourdine pour leur collaboration au long de la période de stage.

Nos remerciements s'adressent particulièrement aux membres de jury qui ont accepté d'examiner notre travail, ainsi que tous les enseignants qui ont contribué à notre formation.

Enfin, Nous remercions aussi tous nos amis et collègues qui nous ont soutenu et tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Dédicaces

Je dédie ce travail :

Aux êtres les plus chères à mon cœur

À mes très chères parents pour leurs soutiens et conseils, que Dieu tout puissant vous préserve et vous procure santé et longue vie à la fin que je puisse à mon tour vous combler.

À ma très chère sœur Naoual d'avoir été toujours là pour moi et son mari Ahmed.

À mes chers frères : David, Snoussi et sa femme Kahina, Cherif et sa femme Rahima.

À mes neveux : Sami et Ishak que j'aurai aimé qu'ils soient présents et à mes nièces Céline, Sarah et notre nouveau née Sarah.

À tous mes cousins et cousines chacun à son nom.

À tous mes oncles et mes tantes.

À tous mes amis: Kouta, Tina, Thinhinane, Timouche, Tissou, Sylvia, Zouzou, Cylia, Kahina, Samou, Nizia, Bilal, Rafik, Rachid, Athmane, Chikou, Djamel, Nassim et Salim.

Siham

Dédicaces

Je dédie ce travail :

A mes chers parents, que nulle dédicace ne peut exprimer mes sincères sentiments, pour leur patience Illimitée, leur encouragement continu et leur aide.

A mon cher mari : Amazigh pour son soutien.

A mes très chers sœurs : Sílía, Thíziri, Lína, Amel et son mari Bilal, Nessrine et son mari Idir.

A mon adorable neveu : Ales.

A ma grand-mère: Faroudja à qui je souhaite une longue vie.

A ma belle-famille BOUABOUD.

A mes chers amis : Siham, Tina, Helima, Samou, Timouch, Bilal, Mimi.

KATIA

TABLES DE MATIERES

| | |
|------------------------------|-----|
| Liste des figures ----- | I |
| Liste des tableaux ----- | II |
| Liste des abréviations ----- | III |
| INTRODUCTION GENERALE ----- | 1 |

CHAPITRE01 : GENERALITES SUR LES RESEAUX

| | |
|--|---|
| I.1. Introduction ----- | 3 |
| I.2. Réseaux informatiques ----- | 3 |
| I.2.1. Qu'est-ce qu'un réseau informatique ? ----- | 3 |
| I.2.2. Classification des réseaux informatiques ----- | 3 |
| I.2.2.1. Réseau PAN() ----- | 3 |
| I.2.2.2. Réseau LAN (Local Area Network) ----- | 3 |
| I.2.2.3. Réseau MAN (Mitropolitain Area Network) ----- | 3 |
| I.2.2.4. Réseau WAN (Wide Area Network) ----- | 3 |
| I.2.2.5. Réseau Internet ----- | 4 |
| I.2.2.5.1. Définition ----- | 4 |
| I.2.2.5.2. Les services de l'Internet ----- | 4 |
| I.2.2.5.3. Architecture Client / Serveur ----- | 4 |
| I.2.2.6.1. Définition ----- | 5 |
| I.2.2.6.2. L'utilité d'un réseau Intranet ----- | 6 |
| I.2.2.7. Extranet ----- | 6 |
| I.2.3. Modèle OSI ----- | 6 |
| I.2.3.1. Couches du modèle OSI ----- | 7 |
| I.2.3.2. Processus de transmission/réception ----- | 8 |
| I.2.4. Modèle TCP/IP ----- | 8 |
| I.2.4.1. Historique ----- | 8 |
| I.2.4.2. Définition ----- | 8 |

| | |
|--|-----------|
| I.2.4.3. Caractéristique de TCP/IP ----- | 9 |
| I.2.4.4. Couches du modèle TCP/IP ----- | 9 |
| I.2.4.5. Protocoles du modèle TCP/IP ----- | 10 |
| I.2.5. Comparaison du modèle OSI et modèle TCP/IP ----- | 11 |
| I.2.5.1. Les similitudes ----- | 11 |
| I.2.5.2. Les différences ----- | 11 |
| I.3. La messagerie électronique ----- | 12 |
| I.3.1. Définition ----- | 12 |
| I.3.2. Architecture du service de messagerie ----- | 12 |
| I.3.3. Etape d'envoi d'un courriel électronique ----- | 13 |
| I.3.4. Protocoles de communication ----- | 13 |
| I.3.5. Trois types d'outils pour consulter les mails ----- | 14 |
| I.4. Conclusion ----- | 14 |

CHAPITRE02 : ANALYSE ET CONCEPTION

| | |
|--|-----------|
| II.1. Introduction ----- | 15 |
| II.2. Présentation d'UML ----- | 15 |
| II.2.1. Définition ----- | 15 |
| II.2.2. Les Diagrammes UML ----- | 15 |
| II.2.2.1. Diagrammes des cas d'utilisation ----- | 17 |
| II.2.2.2. Diagrammes de séquences ----- | 17 |
| II.2.2.3. Diagrammes de classes ----- | 17 |
| II.3. Processus de développement logiciel ----- | 18 |
| II.3.1. Processus unifié (UnifiedProcess) ----- | 18 |
| II.3.2. Adaptation du processus unifié ----- | 18 |
| II.3.2.1. Le processus 2TUP ----- | 18 |
| II.3.2.2. Les phases du processus 2TUP ----- | 19 |
| II.4. L'Analyse ----- | 20 |
| II.4.1. Présentation du projet ----- | 20 |
| II.4.2. Présentation de Cevital agro-alimentaire ----- | 20 |
| II.4.2.1. Présentation de CEVITAL Agro-industrie ----- | 20 |
| II.4.2.2. Organigramme de CEVITAL Agro-alimentaire ----- | 21 |

| | |
|--|-----------|
| II.4.2.3. Besoins fonctionnels | 22 |
| II.4.2.3.1. Besoins de point de vue utilisateur | 22 |
| II.4.2.3.2. Besoins de point de vue administrateur | 23 |
| II.4.2.4. Besoins non fonctionnels | 23 |
| II.5. Diagrammes des cas d'utilisation | 23 |
| II.5.1. Description des acteurs | 23 |
| II.5.2. Description des cas d'utilisation | 23 |
| II.5.2.1. Partie client | 23 |
| II.5.2.2. Partie administrateur | 26 |
| II.5.3. Diagramme de cas d'utilisation général | 28 |
| II.6. La conception | 29 |
| II.6.1. Diagrammes de séquences | 29 |
| II.6.1.2. La partie administrateur | 32 |
| II.6.2. Diagramme de classes | 34 |
| II.6.2.1. Règles de gestion | 34 |
| II.6.2.2. Le dictionnaire de données | 34 |
| II.6.2.3. Réalisation du diagramme | 35 |
| II.6.2.4. Le modèle relationnel | 36 |
| II.7. Conclusion | 36 |

CHAPITRE 03 : REALISATION

| | |
|--|-----------|
| III.1. Introduction | 37 |
| III.2. L'environnement de développement | 37 |
| III.2.1. Langages de programmation | 37 |
| III.2.2.1. Le langage de programmation JAVA | 37 |
| III.2.2.2. Le langage SQL | 37 |
| III.2.3 Outils de programmation | 38 |
| III.2.3.1. Netbeans IDE 8.0.2 | 38 |
| III.2.3.2. EasyPHP | 38 |
| III.2.3.3. Apache | 38 |
| III.2.3.4. MySQL | 38 |
| III.2.3.5. PhpMyAdmin | 39 |
| III.3. Les principales interfaces graphiques | 39 |

| | |
|--|----|
| III.3.1. Interfaces client ----- | 39 |
| III.3.1.1. Espace d'authentification ----- | 39 |
| III.3.1.2. Boite de réception ----- | 40 |
| III.3.1.3. Boite d'envoi----- | 41 |
| III.3.1.4. Nouveau message ----- | 42 |
| III.3.1.5. Messages supprimés ----- | 43 |
| III.3.1.6. Gérer le profil----- | 44 |
| III.3.2. Interfaces administrateur ----- | 45 |
| III.4. Conclusion ----- | 46 |
| CONCLUSION GENERALE ----- | 47 |
| Références bibliographiques ----- | 48 |

Liste des figures

| | |
|---|----|
| Figure1 : Architecture Client / Serveur----- | 5 |
| Figure2 : Architecture à trois niveaux ----- | 5 |
| Figure3 : Processus de transmission/réception----- | 8 |
| Figure4 : Couches du modèle TCP/IP ----- | 9 |
| Figure5 : Comparaison entre les deux modèles OSI et TCP/IP ----- | 11 |
| Figure6 : Etape d'envoi d'un courriel électronique ----- | 13 |
| Figure7 : Les diagrammes d'UML ----- | 16 |
| Figure8 : Processus de développement 2TUP ----- | 19 |
| Figure9 : Organigramme de CEVITAL Agro-alimentaire ----- | 22 |
| Figure10 : Diagramme de cas d'utilisation générale ----- | 28 |
| Figure11 : Diagramme de séquence du cas d'utilisation « s'authentifier » ----- | 29 |
| Figure12 : Diagramme de séquence du cas d'utilisation « envoyer un message » ----- | 30 |
| Figure13 : Diagramme de séquence de cas d'utilisation « Consulter/ lire les messages reçus»----- | 31 |
| Figure14 : Diagramme de séquence du cas d'utilisation « s'authentifier » ----- | 32 |
| Figure15 : Diagramme de séquence du cas d'utilisation « supprimer/ajouter un utilisateur» ----- | 35 |
| Figure16 : Diagramme de classes----- | 36 |
| Figure17 : Interface de la page authentification d'un client ----- | 40 |
| Figure18 : La page d'interface de la boîte de réception ----- | 41 |
| Figure19 : La page d'interface de la boîte d'envoi----- | 42 |
| Figure20 : La page d'interface de nouveau message ----- | 43 |
| Figure21 : La page d'interface des messages supprimés----- | 44 |
| Figure22 : La page d'interface gérer le profil ----- | 45 |
| Figure23 : La page d'interface d'accueil de l'administrateur----- | 46 |

Liste des tableaux

| | |
|---|-----------|
| Tableau1 : Quelques protocoles du modèle TCP | 11 |
| Tableau2 : Le dictionnaire de données | 35 |

LISTE DES ABREVIATIONS

PAN Personal Area Network

LAN Local Area Network

MAN Mitropolitain Area Network

WAN Wide Area Network

TCP Transmission Control Protocol

IP Internet Protocol

OSI Open System Interconnexion

UDP Unit Data Protocol

HTTP HypertText Transfert Protocol

FTP File Tansfert Protocol

SMTP Simple Mail Transfert Protocol

DNS *Domaine Name Server*

ARP Address Resolution Protocol

ATM Asynchronous Transfer Mode

DHCP Dynamic Host Configuration Protocol

MUA Mail User Agent

MTA Mail Transfert Agent

MDA Mail Delivery Agent

IMAP Interactive Mail Access Protocol

POP3 Post Office Protocol version 3

Introduction générale

INTRODUCTION GENERALE

Depuis les origines, l'homme a eu besoin de communiquer. Pour cela, il a mis au point des codes, des alphabets et des langages (Parole, gestes de la main, tam-tam, document écrit,...etc.) pour véhiculer le message. La communication n'a pas cessé de se développer, devenant ainsi de plus en plus technique. Il s'agit d'une notion moderne, qui n'a fait qu'évoluer jusqu'à devenir aujourd'hui un élément fondamental de la stratégie d'entreprise.

Pour cette raison les technologies de l'Information et de la Communication (TIC) ont bouleversé les modes traditionnels de circulation de l'information en apportant de nouvelles formes de circulation de l'information, un partage plus facile, une diffusion plus rapide qui est la messagerie électronique.

Notre thème consiste donc à développer un système de messagerie en intranet au sein de l'entreprise Cevital, qui sera composé de deux applications distinctes : l'application Client et l'application Serveur chacune avec des fonctionnalités bien précises.

Durant notre stage, nous avons pu constater qu'il y a un nombre important d'anomalies et d'ambiguïtés induit par le grand nombre de documents émis et reçus qui circulent quotidiennement à l'intérieur de l'entreprise. De plus, une coupure d'internet dans l'entreprise handicape tout le système de communication interne. Nous avons remarqué aussi que malgré cette dépendance à l'Internet, les postes de travail de l'entreprise ne disposent pas tous d'une connexion Internet. De ce fait, l'entreprise doit se doter d'une meilleure solution afin d'assurer une vitesse de transmission et de confidentialité des données circulant sur le réseau intranet.

L'intranet permet aussi une transformation majeure dans la circulation des flux d'informations au sein de l'entreprise. Assimilable à un réseau isolé et sécurisé, il est administré et contrôlé par une entité qui a les droits et les privilèges nécessaires.

Une autre raison qui peut favoriser l'intégration d'un intranet est le coût. En effet, une application intranet est moins chère à développer, et surtout moins chère à maintenir.

Pour la réalisation de cette application, notre choix s'est porté sur le processus unifié, qui représente une solution de développement logiciel adaptée à tout type de projet. La phase de modélisation sera réalisée en utilisant le langage UML qui est une partie intégrante de la démarche 2TUP.

Pour l'implémentation, le choix du langage de programmation a été dicté par rapport, au type de projet à réaliser. Ainsi, le choix s'est porté sur le langage de programmation orienté objet JAVA. La base de données est implémentée avec EasyPhp.

Notre mémoire est subdivisé en trois grands chapitres.

Le premier chapitre est intitulé « Généralités sur les réseaux », nous y présenterons quelques définitions sur le réseau Internet, l'Intranet et l'Extranet.

Le deuxième chapitre est intitulé «Analyse et conception», qui commencera par la présentation du langage de modélisation unifié (UML), suivi d'une analyse des principaux objectifs attendus du futur système à concevoir. Ces objectifs seront décrits en premier lieu par le diagramme des cas d'utilisation, puis les diagrammes de séquences et enfin le diagramme de classes qui est le plus précis conceptuellement et qui vise principalement à préciser le modèle de telle sorte qu'il puisse être implémenté.

Dans le dernier chapitre nommé «Réalisation», nous présenterons deux parties: la première partie consiste à la présentation des différents outils et langages de programmation, la deuxième sera consacrée à la présentation de quelques interfaces illustrant les différentes fonctionnalités de notre système.

Enfin, nous terminons notre travail par une conclusion et quelques perspectives.

CHAPITRE I

GENERALITES SUR LES RESEAUX

I.1. Introduction

L'informatique s'est historiquement développée autour de la saisie, du traitement et de la mise à disposition de données structurées. Mais avec l'évolution de l'intranet et des technologies de l'information et de la communication, il ne s'agit plus de traiter des données : les ordinateurs sont utilisés pour envoyer du courrier (messagerie), partager les agendas, diffuser la documentation, etc.

La messagerie électronique constitue alors un moyen de communication, utilisés dans différents environnements professionnels : entreprises privées et publiques, collectivités locales, administrations, institutions et ministères.

Avant d'entamer ce concept, nous allons voir quelques généralités sur les réseaux, en particulier le réseau Internet ainsi que ses services, l'Intranet, et pour finir nous présenterons l'Extranet.

I.2. Réseaux informatiques

I.2.1. Qu'est-ce qu'un réseau informatique ?

C'est un ensemble d'objets (ordinateurs, stations de travail, cartes réseau, imprimantes...) reliés entre eux par différents supports de communication selon une architecture physique bien définie. Le but des réseaux informatiques est d'assurer le partage des différentes ressources matériels et logiciels (fichiers, applications, imprimantes...) et la communication par l'échange des informations d'une manière simple, rapide et efficace entre les stations [11].

I.2.2. Classification des réseaux informatiques

Nous distinguons différents types de réseaux selon leur taille (en termes de nombre de machines), leur vitesse de transfert des données ainsi que leur étendue.

La classification la plus connue est basée sur l'étendu et elle donne les types suivants :

I.2.2.1. Réseau PAN (Personal Area Network)

Désigne un type de réseau informatique restreint en termes d'équipements, généralement mis en œuvre dans un espace d'une dizaine de mètres. D'autres appellations pour ce type de réseau sont: réseau domestique ou réseau individuel.

I.2.2.2. Réseau LAN (Local Area Network)

Il s'agit d'un ensemble d'ordinateurs appartenant à une même organisation et reliés entre eux dans une petite aire géographique, il peut s'étendre de quelques mètres à quelques kilomètres et avec un débit de transmission compris entre 4 et 100 Mbits/s.

I.2.2.3. Réseau MAN (Metropolitan Area Network)

Les réseaux métropolitains permettent l'interconnexion de plusieurs réseaux locaux répartis sur différents sites en utilisant des commutateurs et des routeurs et couvrent une surface importante avec une distance comprise entre 1 et 10 Km et une vitesse de transmission qui atteint quelques centaines de Kbits/s.

I.2.2.4. Réseau WAN (Wide Area Network)

Un réseau étendu est un réseau informatique reliant différents LAN entre eux sur une zone géographique importante dans un même pays ou dans le monde. Les débits

disponibles sur un WAN peuvent être faibles à cause de l'augmentation des coûts des liaisons utilisées. Ce type de réseau fonctionne grâce à des passerelles qui permettent d'établir la fonction de routage des messages de l'expéditeur au récepteur [19].

I.2.2.5. Réseau Internet

I.2.2.5.1. Définition

L'Internet est un système d'interconnexion de machines qui constitue un réseau informatique mondial, utilisant un ensemble standardisé par un protocole de transfert de données (TCP/IP : Transfert Control Protocol / Internet Protocol). Ce modèle peut être considéré comme un langage commun par des ordinateurs aux architectures diverses pour échanger des informations sur un réseau.

L'Internet n'est pas en lui-même un réseau primitif d'ordinateurs mais représente plutôt une multitude des réseaux reliés entre eux au fil du temps pour constituer un réseau des réseaux (Interconnected Networks) [11].

I.2.2.5.2. Les services de l'Internet

L'internet est un outil capable de nous rendre un certain nombre de services.

Voici une liste des différents services les plus utilisés sur internet, les plus importants seront détaillés dans la suite du rapport [28]:

- **Le courrier électronique (e-mail)** : permet aux internautes qui possèdent une adresse e-mail de s'expédier mutuellement des messages dans le monde entier. Les messages sont expédiés et reçus grâce à deux protocoles (SMTP et POP).
- **FTP (File Transfer Protocol)**: est un protocole de transfert de fichiers. Ce transfert s'effectue en établissant une connexion entre un serveur FTP et un client FTP situé sur votre ordinateur. Les fichiers échangés sont des fichiers informatiques de tous types (texte, images, sons, logiciels, ...).
- **News** : ce sont des espaces de discussions où des groupes discutent en temps différé. Les news sont regroupés par thèmes d'intérêt, appelés newsgroups ou forums.
- **Telnet et SSH**: ces services permettent à un internaute de se connecter et donc d'utiliser à distance une machine comme s'il se trouvait face à elle. Cela ouvre par exemple des possibilités pour le travail à domicile, puisqu'il devient possible d'utiliser les machines se trouvant sur son lieu de travail depuis chez soi [13].
- **Le World Wide Web (WWW)** : ce service permet d'accéder à des pages, appelées pages Webs. Une page Web est écrite en langage HTML et peut contenir du texte, des images, des séquences vidéo, du son et des liens. Ces liens permettent de passer d'une page Web à une autre. Le protocole utilisé est http (HyperText Transfer Protocol) [30].

Le système fonctionne selon une architecture client/serveur, qui désigne le mode de communication entre plusieurs composants.

I.2.2.5.3. Architecture Client / Serveur

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.

Les services sont exploités par des programmes, appelés programmes clients, s'exécutant sur les machines clientes. On parle ainsi de client (client FTP, client de messagerie, etc.) [18].

Un système client/serveur fonctionne selon le schéma suivant :

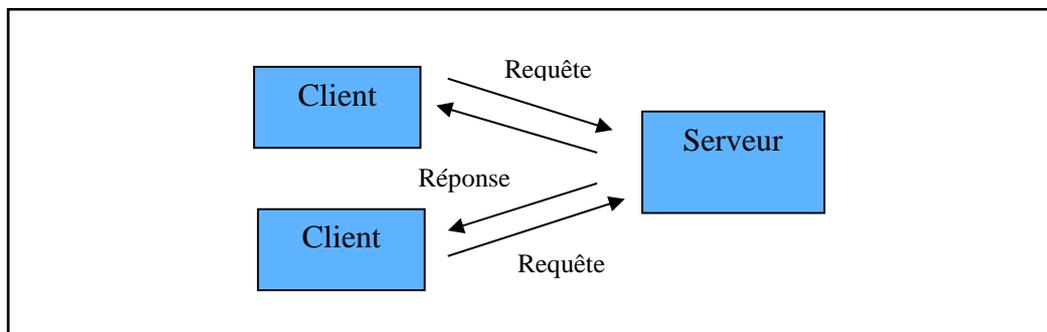


Figure1 : Architecture Client / Serveur.

I.2.2.6. Réseau Intranet

I.2.2.6.1. Définition

Un intranet est un ensemble de services internet (par exemple un serveur web) internes à un réseau local, c'est-à-dire accessibles uniquement à partir des postes d'un réseau local, ou bien d'un ensemble de réseaux bien définis, et invisibles de l'extérieur. Il consiste à utiliser les standards client-serveur de l'internet (en utilisant les protocoles TCP/IP), pour réaliser un système d'information interne à une organisation ou une entreprise [18].

Un intranet repose généralement sur une architecture à trois niveaux, composée :

- de clients (navigateur internet généralement) ;
- d'un ou plusieurs serveurs d'application (middleware): un serveur web permettant d'interpréter des scripts CGI, PHP, ASP ou autres, et les traduire en requêtes SQL afin d'interroger une base de données ;
- d'un serveur de bases de données.

De cette façon, les machines clientes gèrent l'interface graphique, tandis que les différents serveurs manipulent les données. Le réseau permet de véhiculer les requêtes et les réponses entre clients et serveurs, comme la montre la figure2.

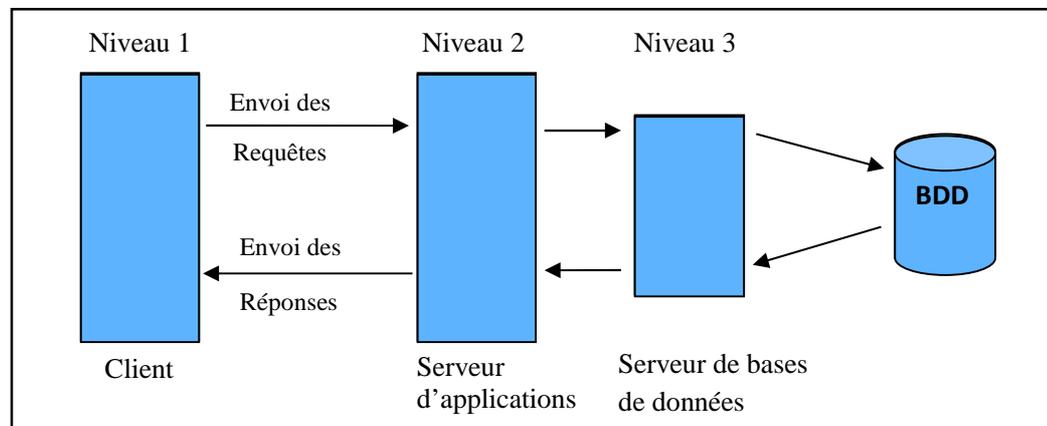


Figure2 : Architecture à trois niveaux.

I.2.2.6.2. L'utilité d'un réseau Intranet

Un intranet dans une entreprise permet de mettre facilement à la disposition des employés des documents divers et variés; cela permet d'avoir un accès centralisé et cohérent à la mémoire de l'entreprise [18].

De cette façon, il est généralement nécessaire de définir des droits d'accès pour les utilisateurs de l'intranet et une authentification afin de leur permettre un accès personnalisé à certains documents.

Voici quelques fonctions qu'un intranet peut réaliser :

- Mise à disposition d'informations sur l'entreprise (panneau d'affichage).
- Mise à disposition de documents techniques.
- Moteur de recherche de documentations.
- Un échange de données entre collaborateurs.
- Annuaire du personnel.
- Gestion de projet, aide à la décision, agenda, ingénierie assistée par ordinateur.
- Messagerie électronique.
- Forum de discussion, liste de diffusion, chat en direct.
- Visioconférence.
- Portail vers internet.

Ainsi, intranet favorise la communication au sein de l'entreprise et limite les erreurs dues à la mauvaise circulation d'une information.

I.2.2.7. Extranet

Un extranet est une extension du système d'information de l'entreprise à des partenaires situés au-delà du réseau.

L'accès à l'extranet doit être sécurisé dans la mesure où cela offre un accès au système d'information à des personnes situées en dehors de l'entreprise.

Il peut s'agir soit d'une authentification simple (authentification par nom d'utilisateur et mot de passe) ou d'une authentification forte (authentification à l'aide d'un certificat). Il est conseillé d'utiliser HTTPS pour toutes les pages web consultées depuis l'extérieur afin de sécuriser le transport des requêtes et des réponses HTTP et d'éviter notamment la circulation du mot de passe en clair sur le réseau.

Un extranet n'est donc ni un intranet, ni un site internet. Il s'agit d'un système supplémentaire offrant par exemple aux clients d'une entreprise, à ses partenaires ou à des filiales, un accès privilégié à certaines ressources informatiques de l'entreprise par l'intermédiaire d'une interface Web [18].

I.2.3. Modèle OSI

Le modèle OSI (Open System Interconnected) est une norme ISO (International Standard Organization) créé comme une architecture descriptive, qui définit un ensemble de spécifications pour une architecture réseau permettant la connexion d'équipements hétérogènes. Ce modèle normalise la manière dont les matériels et les logiciels coopèrent pour assurer la communication réseau, il décompose les systèmes de communication en 7 couches chacune illustrant une fonction réseau bien précise. Une couche (N) offre des

services à la couche supérieure (décapsulation) et à la couche inférieure (encapsulation) tout en lui dissimulant les détails d'implémentation [12,21].

Le découpage du réseau en sept couches présente les avantages suivants :

- Il permet de diviser les communications sur le réseau en éléments plus petits et plus simples.
- Il uniformise les éléments du réseau afin de permettre le développement et le soutien multi constructeur.
- Il permet à différents types de matériel et de logiciel réseau de communiquer entre eux.
- Il empêche les changements apportés à une couche d'affecter les autres couches, ce qui assure un développement plus rapide.
- Il divise les communications sur le réseau en éléments plus petits, ce qui permet de les comprendre plus facilement [26].

I.2.3.1. Couches du modèle OSI

Le modèle OSI est constitué de sept couches distinctes. Dans chacune de ces couches opèrent un certain nombre de protocoles :

- **Couche application** : joue le rôle d'une interface d'accès des applications au réseau, c'est donc elle qui va apporter à l'utilisateur les services de base offerts par le réseau permettent notamment de transférer des fichiers, de rédiger un mail, d'établir une session à distance, et de visualiser une page web. Plusieurs protocoles assurent ces services, dont FTP, Telnet, SMTP, et http [20,12].
- **Couche présentation** : cette couche s'intéresse à la syntaxe et à la sémantique des données transmises en traitant l'information de manière à la rendre compatible entre tâches communicantes [20].
- **Couche session** : la principale fonction de cette couche est de fournir une interface permettant de gérer la synchronisation de la communication en établissant une connexion logique appelé session entre des entités pour tenir un dialogue entre eux [29].
- **Couche transport** : cette couche manipule des unités de données appelées segments, Elle se situe à la jonction des couches orientées traitement et celles orientées communication, elle contrôle l'optimisation de la gestion des ressources réseau en terme de performances et tente d'assurer la communication par l'établissement de la connexion entre l'émetteur et le récepteur afin de garantir une qualité de service acceptable. Parmi ses protocoles on a TCP et UDP [11].
- **Couche réseau** : elle assure la connectivité et l'acheminement de paquets d'une extrémité à une autre, autrement dit, il a pour objectif de gérer l'adressage de transmission des données entre l'émetteur et le récepteur. Le protocole le plus utilisé à ce niveau est bien sûr le protocole IP [23].
- **Couche liaison de données** : elle a pour rôle d'assurer la transmission des trames entre les systèmes d'une manière correcte. Donc, c'est elle qui se charge de la détection et la correction des erreurs de transmission en appliquant des protocoles

de contrôle. Quelques exemples de protocoles de cette couche : Ethernet, PPP et HDLC [26].

- **Couche physique** : elle assure l'interface entre l'équipement informatique et le support physique de communication. En d'autres termes, La couche physique s'occupe de la transmission des bits représentés par des signaux électriques (numérique ou analogique) en respectant certains protocoles bien définis [10].

I.2.3.2. Processus de transmission/réception

Afin de permettre l'acheminement des données entre l'ordinateur source et l'ordinateur de destination, chaque couche du modèle OSI au niveau de la source doit communiquer avec sa couche homologue sur l'ordinateur de destination par l'intermédiaire de protocoles. Cette forme de communication est appelée communication d'égal à égal [25].

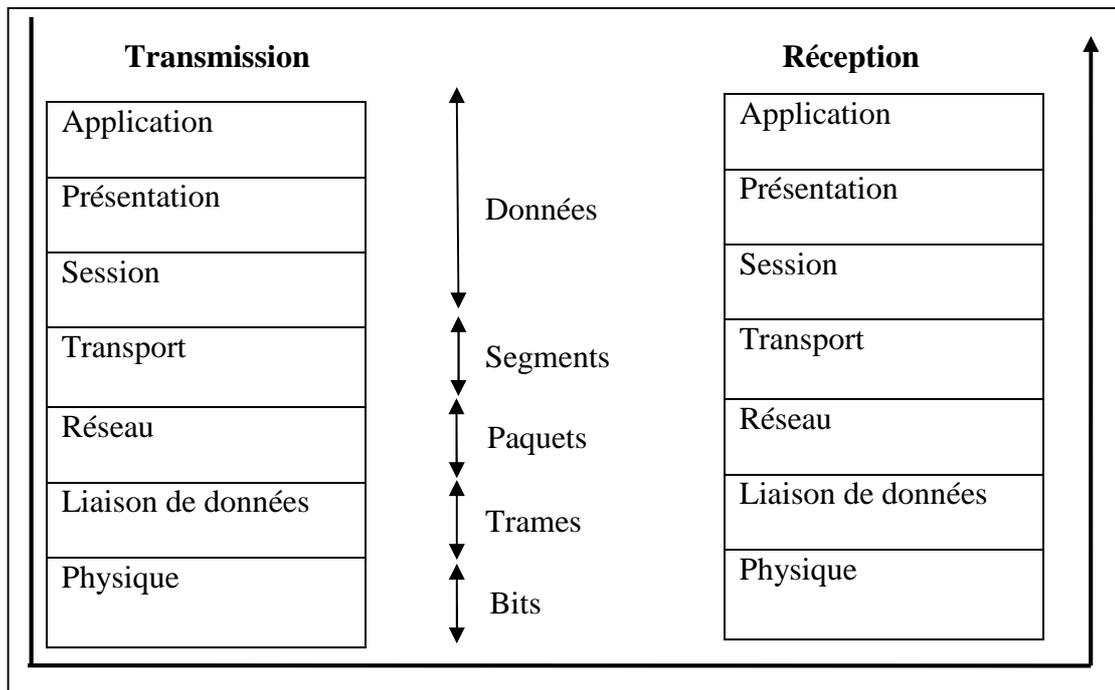


Figure3 : Processus de transmission/réception.

I.2.4. Modèle TCP/IP

I.2.4.1. Historique

Le ministère américain de la Défense a créé le modèle TCP/IP parce qu'il avait besoin de concevoir un réseau pouvant résister à toutes les conditions, même à une guerre nucléaire. Dans un monde connecté par différents types de médias de communication tels que les fils de cuivre, micro-ondes, fibres optiques et liaisons satellite, le ministère de la défense souhaitait une transmission de paquets capable d'aboutir à coup sûr et sous n'importe quelle condition. Ce problème de conception extrêmement ambitieux a conduit à la création du modèle TCP/IP [29].

I.2.4.2. Définition

L'architecture des réseaux informatiques TCP/IP qui porte le nom des principaux protocoles qui constituent ce modèle, est un ensemble de protocoles définis en quatre

couches qui permet de décrire les fonctions qui interviennent à chaque couche de protocoles au sein de la suite TCP/IP. En effet, contrairement au modèle OSI, le modèle TCP/IP est né d'une implémentation ; par la suite vient la normalisation [17].

I.2.4.3. Caractéristique de TCP/IP

Le succès de TCP/IP, s'il vient d'abord d'un choix du gouvernement américain, s'appuie ensuite sur des caractéristiques intéressantes :

- C'est un protocole ouvert, les sources (C) sont disponibles gratuitement. Et ils sont théoriquement transportables sur n'importe quel type de plate-forme.
- Ce protocole est indépendant du support physique du réseau. Cela permet à TCP/IP d'être véhiculé par des supports et des technologies aussi différents qu'une ligne série, un câble coaxial Ethernet, une liaison louée, un réseau token-ring, une liaison radio, infrarouge, ATM, fibre optique, la liste des supports et des technologies n'est pas exhaustive...
- Le mode d'adressage est commun à tous les utilisateurs de TCP/IP. Si l'unicité de l'adresse est respectée, les communications aboutissent même si les hôtes sont aux antipodes.
- Les protocoles de haut niveau sont standardisés ce qui permet des développements largement répandus et interopérables sur tous types de machines [23].

I.2.4.4. Couches du modèle TCP/IP

Bien qu'il n'y a aucune convention universelle décrivant en couches le modèle TCP/IP, on le représente généralement par une hiérarchie qui contient les quatre couches suivantes : la couche accès réseau, la couche Internet, la couche transport et la couche application exactement comme l'illustre la figure suivant [11]:

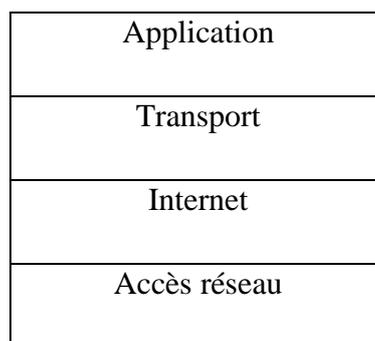


Figure4 : Couches du modèle TCP/IP.

- **Couche Application** : est la couche qui gère les protocoles de haut niveau, les questions de représentation, le code et le contrôle du dialogue. Le modèle TCP/IP regroupe en une seule couche tous les aspects liés aux applications et suppose que les données sont séparées de manière adéquate pour la couche suivante. Ce niveau représente en réalité les différentes applications permettant d'exploiter les services d'un réseau en utilisant les différents protocoles comme : Telnet, FTP, HTTP, SMTP et DNS [28].

- **Couche Transport** : son rôle est le même que celui de la couche transport du modèle OSI, d'abord elle gère les communications de bout en bout entre processus, ensuite elle divise les données en segments, ce qui facilitera le transport des données par la couche inférieure. Enfin cette couche permet le multiplexage des sessions d'un côté et d'effectuer un contrôle d'erreur sur les données du segment dans l'autre côté. Elle contient deux protocoles permettant à deux applications d'échanger des données indépendamment du type de réseau emprunté (c'est-à-dire indépendamment des couches inférieures...), il s'agit des protocoles TCP et UDP [11].
- **Couche internet** : cette couche est chargée d'interconnecter des réseaux distants sans connexion. Son rôle est de permettre l'injection de paquets dans n'importe quel réseau et l'acheminement de ces paquets indépendamment les uns des autres jusqu'à destination. Elle gère aussi la notion d'adressage IP, qui est une adresse logique permettant d'identifier des périphériques situés sur le même réseau ou sur des réseaux différents. La couche Internet contient plusieurs protocoles de base, à titre d'exemple : le protocole IP, ARP, ICMP et RARP [12,20].
- **Couche accès réseau** : le nom de cette couche a un sens très large et peut parfois prêter à confusion. On lui donne également le nom de couche hôte-réseau. Cette couche se charge de tout ce dont un paquet IP a besoin pour établir une liaison physique, puis une autre liaison physique. Cela comprend les détails sur les technologies LAN et WAN, ainsi que tous les détails dans la couche physique et liaison de données du modèle OSI [28].

Elle prend en charge les notions suivantes :

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

I.2.4.5. Protocoles du modèle TCP/IP

Le tableau suivant représente quelques protocoles essentiels correspondants à chaque couche du modèle TCP/IP :

| Couches | Protocoles | L'unité d'échange |
|--------------|------------------------------------|-------------------|
| Application | Telnet, FTP, SMTP, HTTP, DNS, DHCP | Messages |
| Transport | TCP, UDP | Segments |
| Internet | IP, ICMP, ARP, RARP | Paquets |
| Accès réseau | Ethernet, Token Ring | Trames, Bits |

Tableau1 : Quelques protocoles du modèle TCP/IP.

I.2.5. Comparaison du modèle OSI et modèle TCP/IP

Le modèle TCP/IP et le modèle OSI ont beaucoup de points communs. Tous les deux sont basés sur le concept d'empilement de protocoles indépendants. De plus les différentes couches ont des systèmes de fonctionnement similaires.

La figure suivante représente la comparaison entre les deux modèles OSI et TCP/IP :

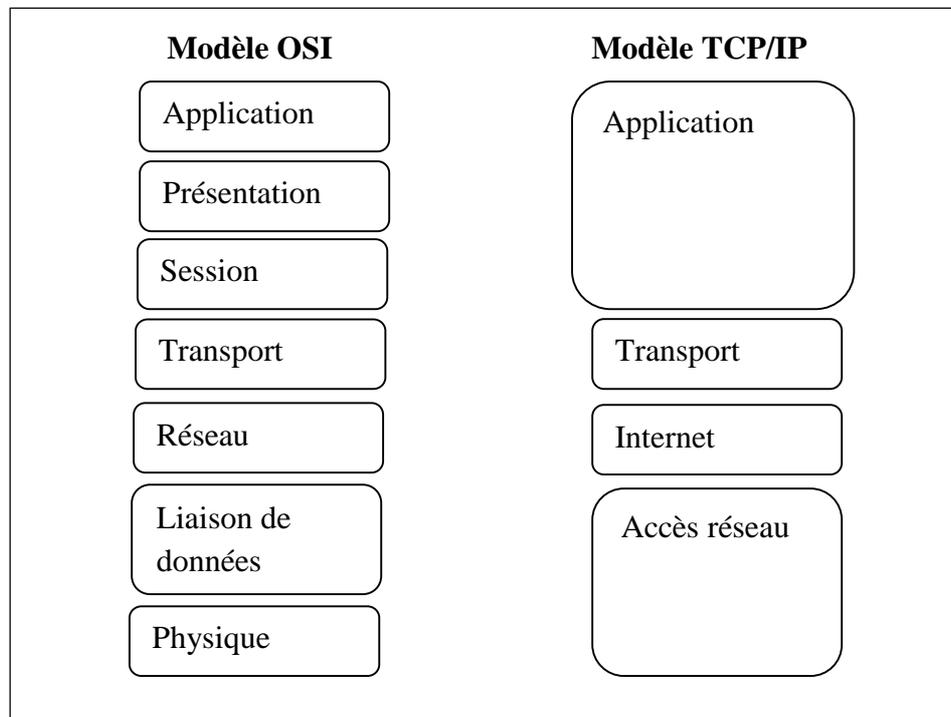


Figure5 : Comparaison entre les deux modèles OSI et TCP/IP.

I.2.5.1. Les similitudes

- Les deux modèles comportent des couches.
- Ils comportent une couche application, bien que chacune fournisse des services très différents.
- Ils comportent des couches réseau et transport comparables.
- Ils supposent l'utilisation de la technologie de communication de paquets (et non de communication circuits) [28].

I.2.5.2. Les différences

- La chose la plus flagrante est le nombre de couches entre les deux modèles. Pour le modèle OSI, il y a 7 couches ; alors que pour le modèle TCP/IP nous sommes à seulement 4 couches.
- TCP/IP intègre la couche présentation et la couche session dans sa couche application.
- TCP/IP regroupe les couches physiques et liaison de données OSI au sein d'une seule couche.
- OSI est un modèle théorique, par contre TCP/IP est un modèle applicatif [28].

-

I.3. La messagerie électronique

I.3.1. Définition

La messagerie électronique n'est pas un service « point à point », ce qui signifie que les machines émettrices et réceptrices des messages n'ont pas besoin d'être reliées ensemble directement pour pouvoir communiquer.

Un serveur de messagerie est un ordinateur-serveur dédié (c'est-à-dire une machine plus des logiciels, offrant des services aux utilisateurs). Ce serveur de messagerie offre des espaces disques pour le stockage des comptes de messagerie personnels et assure la transmission et la réception des courriers électroniques.

Ainsi, une fois parvenu au serveur de messagerie, le message est enregistré dans une boîte aux lettres électronique jusqu'à ce que le destinataire le récupère. Ce serveur joue, en quelque sorte, le rôle de « bureau de poste » [24].

I.3.2. Architecture du service de messagerie

Le service de messagerie est constitué de trois entités distinctes qui coopèrent et communiquent par le biais de protocoles bien défini afin d'assurer un service entre utilisateurs.

- **MUA (*Mail User Agent*)** : c'est le client de messagerie, c'est un programme qui permet de lire et écrire les messages. Il formate les messages en partance afin de les donner au MTA, et les messages de la boîte aux lettres afin de les afficher à l'écran.
- **MTA (*Mail Transfert Agent*)** : lors de l'envoi, l'e-mail est envoyé vers un premier serveur de messagerie appelé : **MTA** ou serveur de messagerie sortant. Son rôle est d'établir en fonction du nom domaine contenu dans l'adresse mail du destinataire quel est le serveur chargé de le prendre en charge. Il lui envoie alors l'e-mail. Le protocole de communication utilisé est le SMTP (Simple mail Transfer Protocol).
- **MDA (*Mail Delivery Agent*)** : ce serveur est aussi appelé Serveur de courrier entrant, les MDA servent à récupérer le courrier électronique. Ils distribuent le courrier dans les boîtes des utilisateurs spécifiés [7].

I.3.3. Etape d'envoi d'un courriel électronique

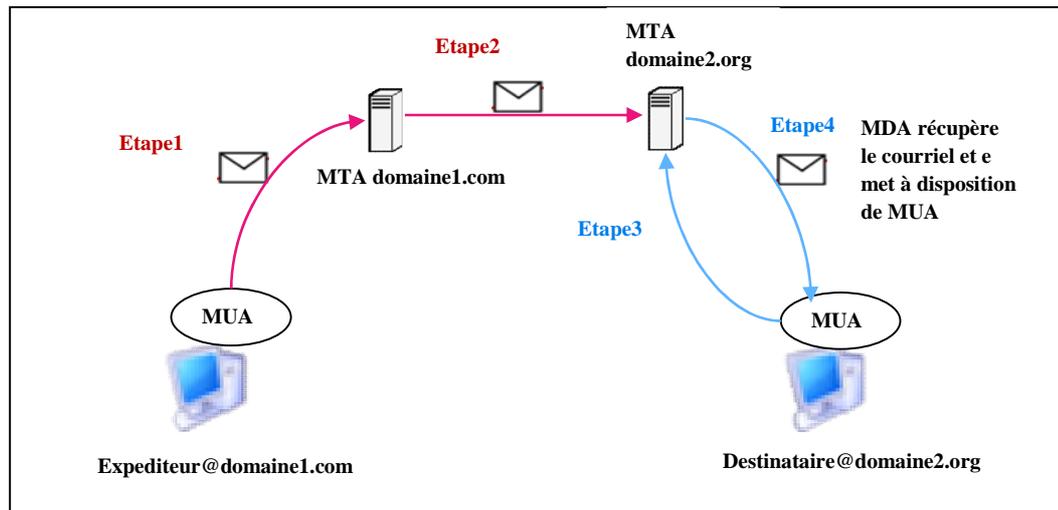


Figure6 : Etape d'envoi d'un courriel électronique.

- Le MUA de l'expéditeur envoie le message vers son serveur de courrier sortant, Le MUA utilise pour cela soit le protocole SMTP soit IMAP.
- Le serveur de messagerie sortant de l'expéditeur, à la réception du message, extrait le domaine domaine2.org de l'adresse du destinataire et effectue une recherche DNS de type MX pour chercher l'adresse du serveur de messagerie du domaine domaine2.org. Une fois trouvée l'adresse IP de ce serveur, il ouvre une connexion avec lui et lui transfère le message de user1 et ferme la connexion.
 - Le MTA du destinataire délivre le message vers le MDA du domaine domaine2 avec le protocole LMTP.
 - Souvent le MDA est un logiciel sur la même machine que le MTA. Le MDA met le message reçu dans la boîte de destinataire.
- L'utilisateur destinataire utilise son MUA pour récupérer auprès de son MDA, ses messages entrants. L'échange se fait soit en POP3 (Post Office Protocole), soit en IMAP.
- Le MDA envoie le message au MUA du destinataire [7].

I.3.4. Protocoles de communication

Le fonctionnement du courrier électronique repose sur une série de protocoles de communication destinés à envoyer ses messages, de serveur à serveur, à travers l'Internet. Parmi ces principaux protocoles on a :

- **Le protocole SMTP (Simple Mail Transfert Protocol):** est le protocole standard permettant de transférer le courrier entre deux serveurs de messagerie ; celui de l'expéditeur et celui du destinataire.

Pour y arriver, il analyse dans un premier temps la partie de l'adresse située à droite du @, pour trouver le domaine du destinataire. Si ce domaine le concerne, il cherche alors la boîte aux lettres du destinataire en regardant la partie de l'adresse

située à gauche du @. Si le domaine du destinataire ne le concerne pas, il va chercher le serveur SMTP qui gère ce domaine, au moyen des champs MX du DNS (*Domaine Name Server*) du domaine destinataire et transmet le message à ce serveur. Le SMTP utilise TCP et le port 25.

- **Le protocole POP3 (Post Office Protocol 3):** permettant de relever le courrier sur un MDA ou (serveur POP). Ce protocole est nécessaire pour les personnes qui ne sont pas connectées en permanence à l'Internet.

Ainsi, POP3 permet le traitement hors-ligne de ses emails. Il suffit de se connecter périodiquement à son serveur de messagerie, via un logiciel spécifique, pour rapatrier sur sa machine le courrier en attente. Les messages récupérés sont ensuite effacés du serveur de messagerie. POP3 est basé sur TCP (port 110).

L'évolution du courrier électronique vers le multimédia et le manque de flexibilité de POP favorisent l'émergence d'un nouveau protocole : l'IMAP.

- **Le protocole IMAP (Interactive Mail Access Protocol):** offre quant à lui des possibilités plus avancées que POP. Il peut assurer un traitement hors-ligne des messages mais sa qualité principale réside dans sa capacité à traiter les courriels en ligne. Tous les courriers et dossiers de messages restent sur le serveur comme il permet aussi de gérer les fichiers dans les dossiers distants (suppression, déplacement d'un dossier à un autre,...), et d'accéder à un compte de plusieurs postes simultanément [22].

I.3.5. Trois types d'outils pour consulter les mails

Pour aller sur le Web, on utilise un navigateur. Mais pour consulter (et envoyer) des mails, il y a deux façons de procéder : soit avec un logiciel spécialement conçu pour la gestion de mail, soit également via le navigateur Internet. Par ailleurs, il existe ce que l'on appelle des clients mail, qui permettent de traiter des courriers électroniques.

- **Client léger :** est une application cliente entièrement gérée par un serveur, de la gestion au stockage des données. Les utilisateurs de l'application auront accès aux données par un portail sécurisé depuis leur navigateur, c'est-à-dire accéder via un site web (Hotmail et Gmail sont typiquement des clients mail en ligne).
- **Client lourd :** le poste client doit comporter un système d'exploitation capable d'exécuter en local une partie des traitements. Le traitement de la réponse à la requête du client utilisateur va mettre en œuvre un travail combiné entre l'ordinateur serveur et le poste client.

Les plus connus des logiciels sont Thunderbird (Mozilla) et Outlook (Microsoft).

I.4. Conclusion

Dans ce chapitre, nous avons présenté quelques notions de base sur les réseaux informatiques, par la suite nous avons vu le réseau Internet et ses différents services, la définition de l'Intranet et son utilité et enfin un petit aperçu sur l'Extranet.

Nous pouvons ainsi entamer la phase de présentation de notre travail dans le prochain chapitre, qui consiste à la modélisation du système et cela à l'aide de différents outils de modélisation du langage UML.

CHAPITRE II

ANALYSE ET CONCEPTION

II.1. Introduction

Au fil des années, l'intranet s'affirme de plus en plus comme un outil indispensable en entreprise. En effet lorsque l'information circule bien, elle favorise la communication interne dans l'entreprise et devient, de ce fait un facteur de cohésion, de motivation, de décision efficace et de créativité.

Or, la conception d'un intranet pose plusieurs questions comme la nature des informations à traiter, les personnes qui vont être concernées et les besoins fonctionnels de système. Pour répondre à ces questions il faut prendre en compte plusieurs critères.

Dans ce chapitre, nous allons suivre une méthode d'analyse et de conception qui a pour objectif de permettre de formaliser les étapes préliminaires du développement d'un système afin de le rendre plus fidèle aux besoins des utilisateurs.

II.2. Présentation d'UML

Durant les années 70 jusqu'à 90 un nombre de méthodes objets ont été développées, nous pouvons citer la méthode OMT (Object Modeling Technique) de Rumbaugh, BOOCH de GradyBooch ainsi que OOSE d'Yvar Jacobson, l'inconvénient majeur de toutes ces méthodes était leur divergence sur certains exemples. C'est de là que naquit le besoin de standardisation qui a conduit à l'apparition de l'UML (UnifiedModelingLanguage) suite à des travaux d'unification [1].

II.2.1. Définition

UML est un langage ou formalisme de modélisation orienté objet, il est basé sur la modélisation graphique et textuelle via un certain nombre de diagrammes. Ces derniers nous permettent de comprendre et décrire des besoins, spécifier et documenter des systèmes, concevoir des solutions et communiquer des points de vue.

UML n'est pas un langage à proprement parler, mais plutôt comme une boîte d'outils qui sert à améliorer les méthodes de travail [3].

II.2.2. Les Diagrammes UML

UML dans sa version 2 comporte treize types de diagrammes, qui permettent de définir une application selon plusieurs points de vue. Ces diagrammes sont regroupés dans deux grands ensembles: les diagrammes structurels (statiques) et les diagrammes de comportement [25].

- **Diagrammes structurels** : ces diagrammes, au nombre de six, ont vocation à représenter l'aspect statique d'un système (classes, objets, composant).
- **Diagrammes de comportement** : ces diagrammes représentent la partie dynamique d'un système réagissant aux événements et permettant de produire les résultats attendus par les utilisateurs. Sept diagrammes sont proposés par UML 2.

L'ensemble des treize types de diagrammes UML 2 peut ainsi être résumé par la figure ci-après :

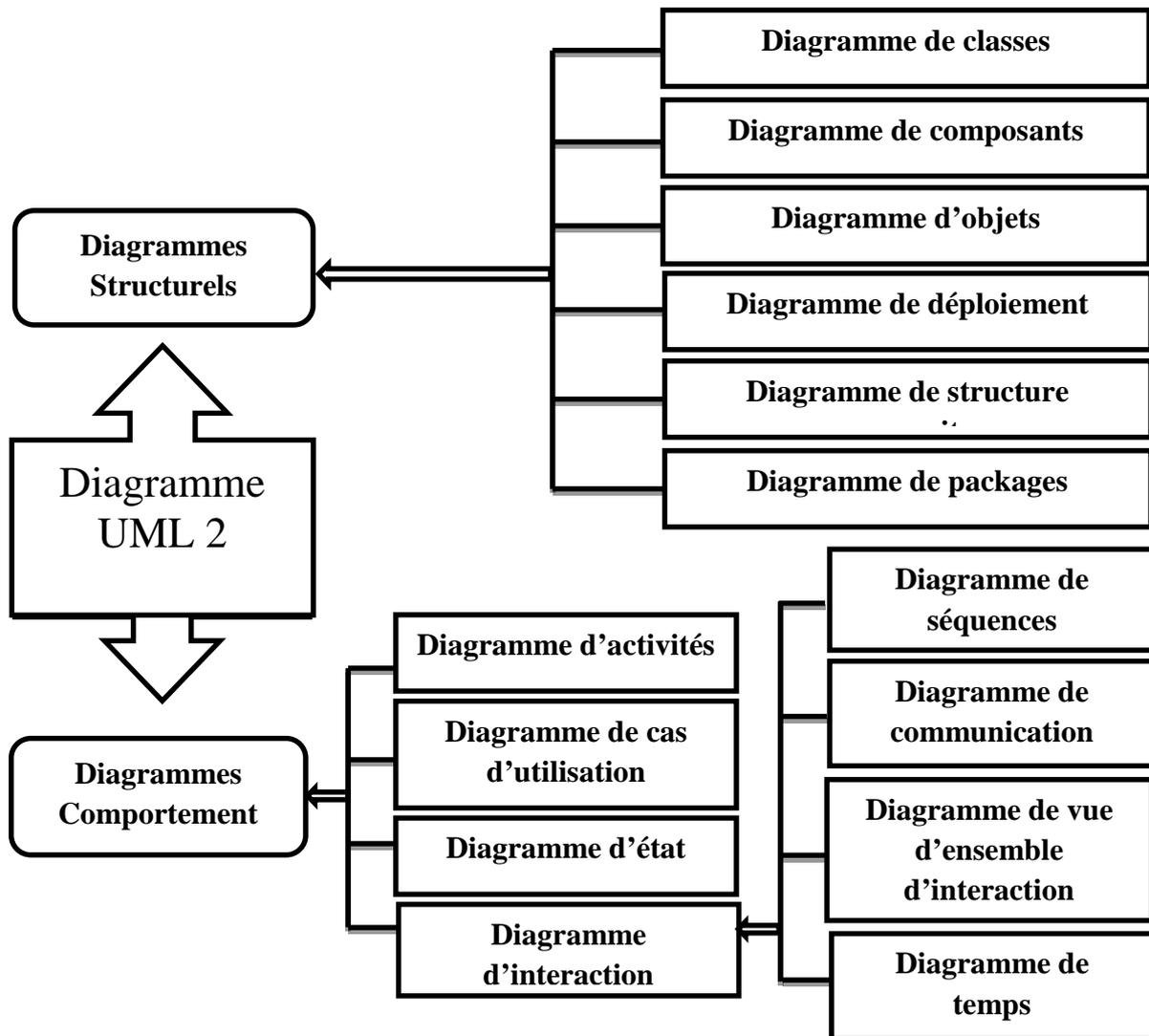


Figure7 : Les diagrammes d'UML.

Ces diagrammes, d'une utilité variable selon les cas, ne sont pas nécessairement tous produits à l'occasion d'une modélisation. Les plus utiles pour la maîtrise d'ouvrage sont les diagrammes d'activités, de cas d'utilisation, de classes, d'objets, de séquences et d'état-transition. Les diagrammes de composants, de déploiement et de communication sont surtout utiles pour la maîtrise d'œuvre à qui ils permettent de formaliser les contraintes de la réalisation et la solution technique [9].

Nous présentons ci-dessous les diagrammes UML 2, que nous avons utilisés dans le cadre de ce projet et quelques notions de base qui leurs sont associées.

II.2.2.1. Diagrammes des cas d'utilisation

Ce sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils montrent les interactions entre un utilisateur (humain ou machine) et un système [4].

Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs, et la suite logique des actions qui constituent ce cas est appelée scénario.

- **Acteurs** : un acteur est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système, qui peut être [9] :
 - **Primaire** : pour lequel l'objectif du cas d'utilisation est essentiel.
 - **Secondaire** : qui interagit avec le cas d'utilisation mais dont l'objectif n'est pas essentiel [1].
- **Cas d'utilisation** : un cas d'utilisation est une unité cohérente représentant une fonctionnalité visible de l'extérieur. Il réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l'acteur qui l'initie. Un cas d'utilisation modélise donc un service rendu par le système, sans imposer le mode de réalisation de ce service [9].

II.2.2.2. Diagrammes de séquences

Ils décrivent la dynamique du système en représentant les échanges de messages entre objets et permettent de mieux visualiser la séquence des messages par une lecture de haut en bas [4].

- **Scénario**: représente une succession particulière d'enchaînements, s'exécutant du début à la fin du cas d'utilisation, un enchaînement étant l'unité de description de séquences d'actions [16].
- **Ligne de vie**: représente l'ensemble des opérations exécutées par un objet [2].
- **Message**: un message est une transmission d'information unidirectionnelle entre deux objets, l'objet émetteur et l'objet récepteur. Dans un diagramme de séquence, deux types des messages peuvent être distingués :
 - **Message synchrone** : dans ce cas l'émetteur reste en attente de la réponse à son message avant de poursuivre ses actions.
 - **Message asynchrone** : dans ce cas, l'émetteur n'attend pas la réponse à son message, il poursuit l'exécution de ses opérations.
- **Période d'activités** : correspondant au temps pendant lequel un objet effectue une action [2].

II.2.2.3. Diagrammes de classes

Les diagrammes de classes sont généralement considérés comme les plus importants dans un développement orienté objet. Ils représentent l'architecture conceptuelle du système : ils décrivent les classes que le système utilise, les relations entre objets, les attributs, et les opérations qui caractérisent chaque classe d'objets [4].

- **Une classe:** représente la description abstraite d'un ensemble d'objets possédant les mêmes caractéristiques. On parle également de type.
- **Un attribut:** représente un type d'information contenu dans une classe.
- **Une opération:** représente un élément de comportement (un service) contenu dans une classe.
- **Une association:** représente une relation sémantique durable entre deux classes [16].
- **Un objet:** est une entité identifiable du monde réel, tout objet possède un ensemble d'attribut (sa structure) et un ensemble de méthodes (son comportement). Un objet est une instance (ou occurrence) d'une classe [1].

II.3. Processus de développement logiciel

Un processus définit une séquence d'étapes, en partie ordonnées, qui concourent à l'obtention d'un système logiciel ou à l'évolution d'un système existant.

L'objet d'un processus de développement est de produire des logiciels de qualité qui répondent aux besoins de leurs utilisateurs dans des temps et des coûts prévisibles.

II.3.1. Processus unifié (UnifiedProcess)

Le processus unifié est un processus de développement logiciels orientés objets, centré sur l'architecture, guidé par des cas d'utilisation et orienté vers la diminution des risques. C'est une méthode générique, itérative et incrémentale [15].

La méthode UP est générique c'est à dire qu'elle définit un certain nombre de critères de développement, que chaque société peut par la suite personnaliser afin de créer son propre processus plus adapté à ses besoins. Itératif et incrémental vu que les projets à réaliser sont de plus en plus complexes et grands, l'idée est de découper le travail en mini projets, Chacun d'entre eux représente une itération qui donne lieu à un incrément. Les itérations désignent des étapes de l'enchaînement d'activités, tandis que les incréments correspondent à des stades de développement du produit [8].

II.3.2. Adaptation du processus unifié

Il existe plusieurs processus de développement qui implémente l'UP dont le plus intéressant est le 2TUP (2TracksUnifiedProcess).

II.3.2.1. Le processus 2TUP

Est un processus unifié qui apporte une réponse aux contraintes de changement continu imposé aux systèmes d'information des entreprises. De ce fait tout développement peut être décomposé et traité selon un axe fonctionnel et un axe technique ; la réalisation du système consiste à la fusion de ces deux branches de processus [5].

La schématisation du processus de développement correspond alors à un Y. Les deux perspectives se rejoignant lors de la phase de conception-réalisation.

II.3.2.2. Les phases du processus 2TUP

Le 2TUP propose un cycle de développement qui dissocie les aspects techniques des aspects fonctionnels et propose une étude parallèle des deux branches : fonctionnelle (étude de l'application) et la technique (étude de l'implémentation). Illustré sur la figure suivante, le processus 2TUP s'articule autour de trois branches :

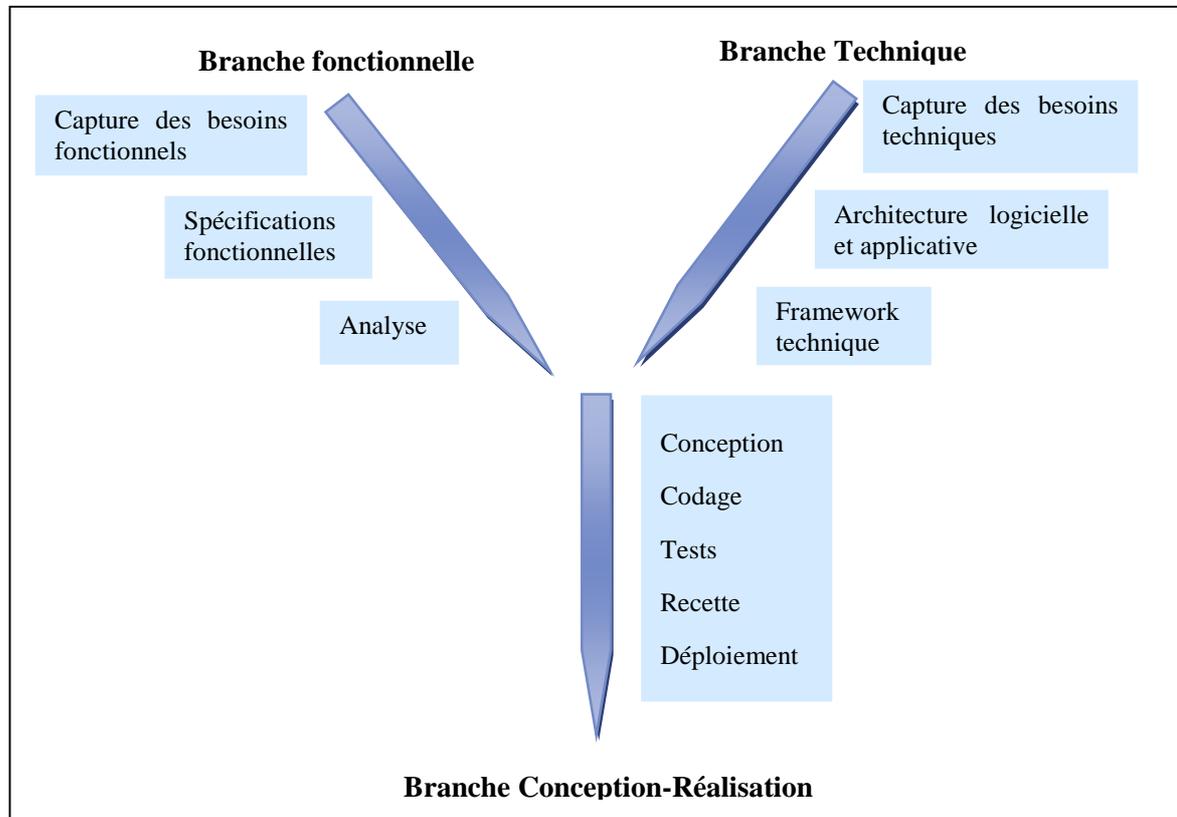


Figure8 : Processus de développement 2TUP.

- **Branche fonctionnelle** : les principales étapes de la branche fonctionnelle se présentent comme suit :
 - L'étape capture des besoins fonctionnels produit le modèle des besoins focalisé sur le métier des utilisateurs. Elle qualifie, au plus tôt le risque de produire un système inadapté aux utilisateurs.
 - L'étape d'analyse consiste à étudier précisément les spécifications fonctionnelles de manière à obtenir une idée de ce que va réaliser le système en terme de métier.
- **Branche technique**: les principales étapes de la branche technique se présentent comme suit :
 - L'étape capture des besoins techniques recense toutes les contraintes sur les choix de dimensionnement et la conception du système. Les outils et le matériel sélectionnés ainsi que la prise en compte des contraintes d'intégration avec l'existant (pré requis d'architecture technique). Cette étape permet de définir le

modèle d'analyse technique. Le rôle de ce dernier est d'établir les couches logicielles et y spécifie les activités techniques attendues.

–L'étape conception générique définit ensuite les composants nécessaires à la construction de l'architecture technique. Cette conception est complètement indépendante des aspects fonctionnels. Elle permet de générer le modèle de conception technique qui définit les Framework. Ces derniers, délivrant les services techniques, assurent la réponse aux exigences opérationnelles du système.

- **Branche conception-réalisation:** les principales étapes de cette branche se présentent comme suit :

–L'étape conception préliminaire est une étape délicate, car elle intègre le modèle d'analyse fonctionnelle dans l'architecture technique de manière à tracer la cartographie des composants du système à développer. Cette étape permet de produire le modèle de conception système. Ce dernier organise le système en composants, délivrant les services techniques et fonctionnels. Ce modèle regroupe les informations des branches technique et fonctionnelle.

– L'étape conception détaillée permet d'étudier comment réaliser chaque composant. Cette étape produit le modèle de conception des composants. Ce modèle fournit l'image prête à fabriquer du système complet. C'est dans l'étape de codage que s'effectue la production des composants et les tests des unités de code au fur et à mesure de leur réalisation.

–L'étape de recette consiste à valider les fonctionnalités du système développé[5].

II.4. L'Analyse

L'analyse permet une formalisation du système à développer en réponse à l'expression des besoins formulés par les utilisateurs. Cet activité débute par la définition des cas d'utilisation et de leurs scénarios, la définition des différents acteurs participant à la réalisation de ces scénarios ainsi que les besoins fonctionnel de système qui aboutira à un ensemble de diagrammes formés avec des classes d'analyse qui représentent la dynamique de système précédemment détaillée dans les cas d'utilisation.

II.4.1. Présentation du projet

La finalité de ce projet est de réaliser un système de messagerie intranet au sein de l'entreprise Cevital agro-alimentaire, ayant pour objectif de permettre de réduire le nombre d'appels internes, de rencontres entre la direction et le personnel. Il est aussi un excellent moyen de coordination d'une équipe ou d'un service. Outre ces possibilités, il permet le partage de fichiers, avec la fonction de joindre des fichiers au message.

II.4.2. Présentation de Cevital agro-alimentaire

II.4.2.1. Présentation de CEVITAL Agro-industrie

CEVITAL Agro-industrie (ou SAP) est une filiale du groupe CEVITAL, créée en 1998, elle est régie par le code de commerce modifié et complété par le décret législatif n°08/93 du 25 avril 1993 et l'ordonnance n°27/96 du 9 décembre 1996 et l'ensemble des textes à caractère législatif et réglementaires ultérieurs. Implantée au sein du port de Bejaia.

CEVITAL Agro-industrie est le leader du secteur agroalimentaire en Algérie. Il est composé de plusieurs unités de production : deux raffineries de sucre, une unité de sucre liquide, une raffinerie d'huile, une margarinerie, une unité de conditionnement d'eau minérale, une unité de fabrication et de conditionnement de boisson rafraichissante et une conserverie.

1999: entrée en production de la raffinerie d'huile et lancement de la 1ère marque d'huile de table de haute qualité, 100% tournesol "FLEURIAL".

2001: lancement de la 1ère marque de margarine de table "FLEURIAL".

2003: entrée en production de la raffinerie de sucre (sucre blanc cristallisé « SKOR » aux normes de l'Union Européenne) et lancement de la margarine de feuilletage "LA PARISIENNE".

Avril 2005: Lancement de trois nouveaux projets (verre plat, fabrication industrielle de produits manufacturés en béton et l'acquisition des eaux minérales Lalla Khadîdja).

2006: Acquisition de COJEK: Jus et conserves.

II.4.2.2. Organigramme de CEVITAL Agro-alimentaire

Voici le schéma général du groupe Cevital, dont chaque direction a pour but d'assurer le après bon fonctionnement de chaque partie du groupe comme le montre la figure citée si.

Notre étude est faite à la direction de Système d'informations qui permet :

- D'assurer la mise en place des moyens des technologies de l'information nécessaires pour supporter et améliorer l'activité, la stratégie et la performance de l'entreprise.
- De veiller à la cohérence des moyens informatiques et de communication mise à la disposition des utilisateurs, à leur mise à niveau, à leur maîtrise technique et à leur disponibilité et opérationnalité permanente et en toute sécurité.
- De définir, également les évolutions nécessaires en fonction des objectifs de l'entreprise et des nouvelles technologies.

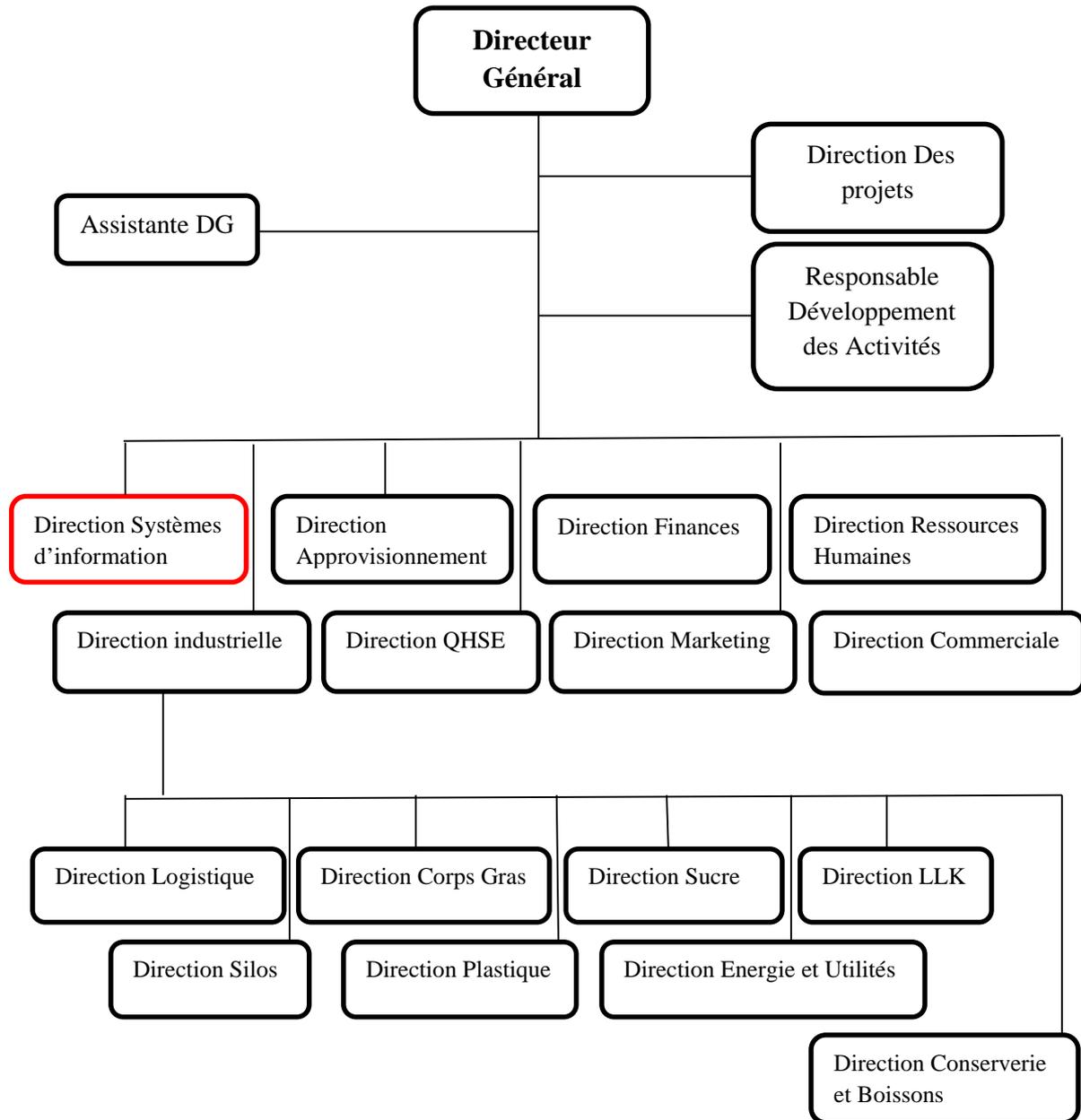


Figure9 : Organigramme de CEVITAL Agro-alimentaire.

II.4.2.3. Besoins fonctionnels

Les besoins fonctionnels ou besoin métiers représentent les actions que le système doit exécuter, il ne devient opérationnel que s'il les satisfait.

Cette application doit couvrir principalement les besoins fonctionnels suivants :

II.4.2.3.1. Besoins de point de vue utilisateur

L'application doit répondre aux critères suivants:

- Gérer son profil.
- Envoyer des messages à d'autres utilisateurs avec ou son pièce jointe.
- Recevoir et visualiser des messages.

- Diffusion d'un message à un ensemble d'utilisateurs.
- Récupérer le mot de passe oublié. Si l'utilisateur oublie son mot de passe, il sera automatiquement redirigé vers une page de récupération de ce dernier.

II.4.2.3.2. Besoins de point de vue administrateur

- Gérer les comptes des inscrits.
- Créer des boîtes aux lettres pour les utilisateurs.

II.4.2.4. Besoins non fonctionnels

Ce sont des exigences qui ne concernent pas spécifiquement le comportement du système mais plutôt identifient des contraintes internes et externes du système.

Les principaux besoins non fonctionnels de notre application se résument dans les points suivants :

- **Ergonomie des interfaces:** la solution doit présenter une interface ergonomique englobant toutes les fonctionnalités offertes. La manipulation de l'interface ne doit pas nécessiter des connaissances poussées en informatique, elle doit être simple et claire afin de s'adapter aux connaissances informatiques des utilisateurs.
- **Robustesse et maintenabilité:** l'application doit permettre le stockage des informations concernant tous les internautes inscrits et les différents traitements utiles pour le fonctionnement correct, ainsi qu'assurer une gestion exhaustive des erreurs.
- **Fiabilité et rapidité :** notre système doit garantir la rapidité et la fiabilité de la recherche des informations, ainsi qu'une gestion optimale des ressources.
- **Le code** doit être clair pour permettre des futures évolutions ou améliorations.

II.5. Diagrammes des cas d'utilisation

II.5.1. Description des acteurs

Les acteurs qui interagissent avec le système sont :

- **Les utilisateurs :** il s'agit des personnes qui ont le droit de consulter leurs boîtes aux lettres après être authentifié.
- **L'administrateur :** c'est la personne qui s'occupe de l'administration du système, il a pour rôle d'effectuer de différentes mises à jour sur la liste des utilisateurs.

II.5.2. Description des cas d'utilisation

II.5.2.1. Partie client

Les figures qui suivent représentent une description de certains cas d'utilisation de notre système :

✓ Cas d'utilisation « **S'authentifier et afficher l'application** »

Utilisation : s'authentifier.

Acteur : utilisateur.

Objectif : cette fonctionnalité permet à l'utilisateur d'accéder à l'application.

Description :

1. Saisir le login et le mot de passe puis valider en cliquant sur le bouton « Connexion ».
2. Le système envoie les données au serveur, ce dernier envoie une requête à la BDD pour vérifier la véracité des informations saisies, En cas d'erreur, il le signale à l'utilisateur sinon il affiche la boîte de réception.

✓ Cas d'utilisation « **Consulter/lire les messages reçus** »

Utilisation : consulter les messages reçus.

Acteur : utilisateur.

Objectif : cette fonctionnalité permet aux utilisateurs de consulter les messages reçus.

Description :

1. L'utilisateur sélectionne la consultation des messages reçus.
2. Le système envoie une requête au serveur, ce dernier interroge la base de données puis il retourne la liste des messages reçus, et le système l'affiche à l'utilisateur.
3. L'utilisateur peut lire un message, comme il peut le supprimer.

✓ Cas d'utilisation « **Ecrire un message** »

Utilisation : écrire un message.

Acteur : utilisateur.

Objectif : cette fonctionnalité permet aux utilisateurs d'écrire un nouveau message pour pouvoir l'envoyer, lui attacher une pièce jointe s'il y a une nécessité.

Description :

1. L'utilisateur accède à « nouveau_message », et écrit le message dans la zone contenu, ainsi que l'objet et le destinataire. L'utilisateur peut aussi ajouter une pièce jointe.
2. Le système vérifie le contenu de message à envoyer, en cas d'erreur, il le signale à l'utilisateur sinon il envoie le message au serveur via le socket client pour l'enregistrer dans la base de données. Le récepteur pourra récupérer le message au moment de sa connexion.

✓ Cas d'utilisation « **Récupérer le mot de passe oublié** »

Utilisation : récupérer le mot de passe oublié.

Acteur : utilisateur.

Objectif : cette fonctionnalité permet à l'utilisateur de récupérer son mot de passe.

Description :

1. Si un utilisateur oublie son mot de passe, il peut le récupérer en cliquant sur « mot de passe oublié ». L'utilisateur remplit le formulaire, puis le valide.
2. Le système envoie la requête au serveur pour vérifier l'identité du client dans la base de données, puis affiche au client son mot de passe.

II.5.2.2. Partie administrateur

- ✓ Cas d'utilisation « **S'authentifier et faire connecter le serveur** ».

Utilisation : s'authentifier.

Acteur : administrateur.

Objectif : cette fonctionnalité permet à l'administrateur d'accéder à ses privilèges, et de lancer le serveur.

Description :

2. Saisir le login et le mot de passe puis valider en cliquant sur le bouton « Connexion ».
3. Le système vérifie la véracité des données, les compare avec celles de la BDD puis il affiche l'espace administrateur et lance le serveur. En cas d'erreur, il le signale.

- ✓ Cas d'utilisation « **Ajouter un utilisateur** »

Utilisation : ajouter un utilisateur.

Acteur : administrateur.

Objectif : cette fonctionnalité permet à l'administrateur d'ajouter un nouvel utilisateur.

Description :

1. L'administrateur saisit les informations de l'utilisateur dans un formulaire qui apparaît à gauche de son interface, puis il clique sur le bouton « Ajouter ».
2. Le système vérifie les informations saisies, en cas d'erreur il le signale à l'administrateur, sinon l'utilisateur sera enregistré dans la base de données.

- ✓ Cas d'utilisation « **Supprimer un utilisateur** »

Utilisation : supprimer un utilisateur.

Acteur : administrateur.

Objectif :cette fonctionnalité permet à l'administrateur de supprimer un utilisateur de la liste des utilisateurs et non pas de la BDD.

Description :

1. Après l'authentification de l'administrateur, le système affiche la liste des utilisateurs dans un tableau.
2. L'administrateur sélectionne la ligne de l'utilisateur à supprimer puis il clique sur le bouton « Supprimer », le système demande une confirmation.Si l'administrateur confirme, l'utilisateur est supprimé, sinon il annule la suppression.

II.5.3. Diagramme de cas d'utilisation général

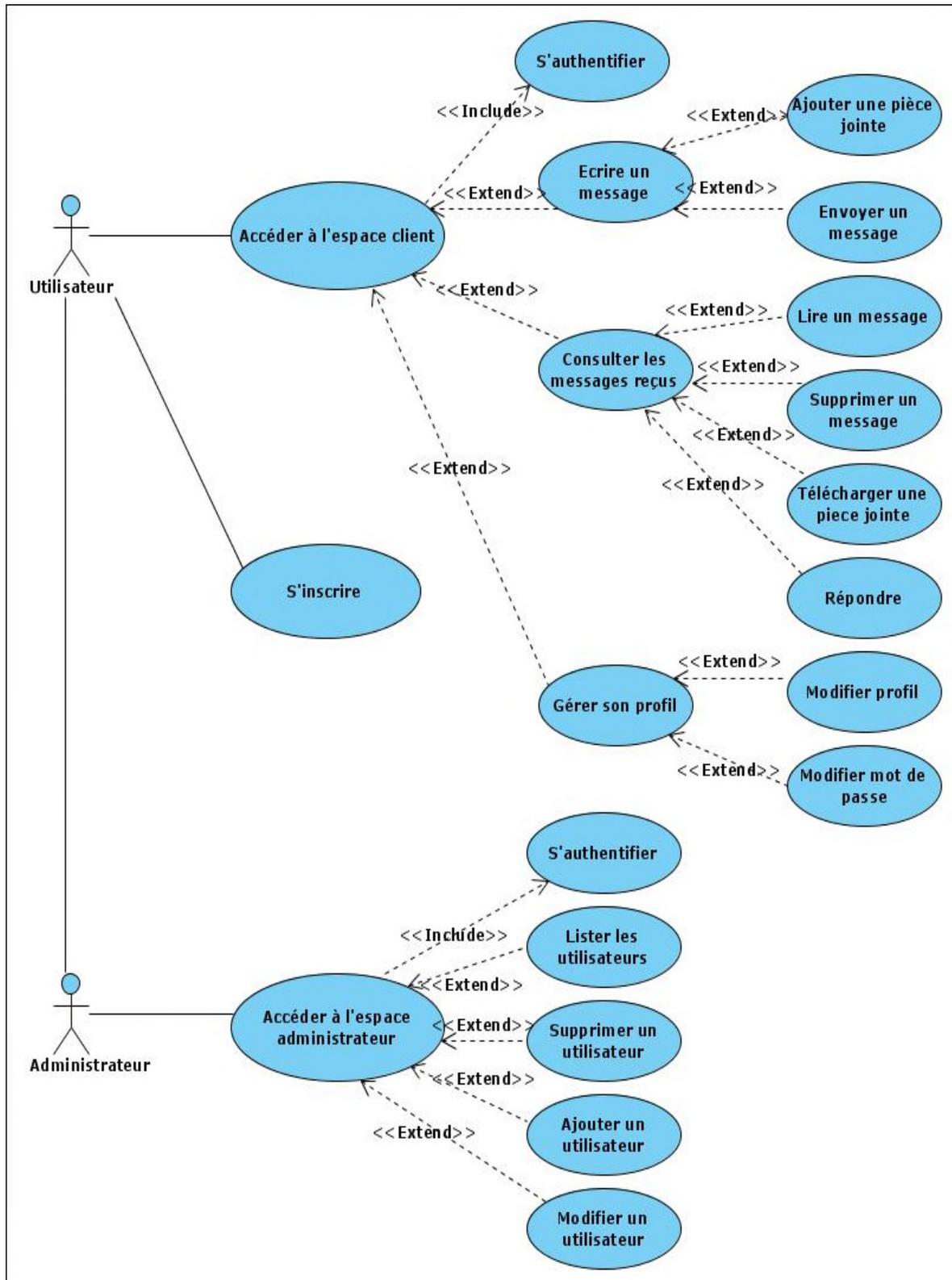


Figure10 : Diagramme de cas d'utilisation général.

II.6. La conception

La conception est la phase la plus importante dans un projet. Elle vise principalement à décrire le modèle de telle sorte qu'il puisse être bien implémenté. Pour cela, nous allons définir les diagrammes de séquence et le diagramme de classe qui permettront de décrire les fonctionnalités obtenus durant la phase d'analyse.

II.6.1. Diagrammes de séquences

Les diagrammes de séquences représentent la dynamique du fonctionnement de l'application. Ces diagrammes permettent essentiellement de décrire les scénarios de chaque cas d'utilisation, en mettant l'accent sur la chronologie des opérations en interaction avec les objets [14].

Dans la description des cas d'utilisation nous avons pu identifier les scénarios. Dans ce qui suit, nous allons traduire ces derniers en diagrammes de séquence.

II.6.1.1. La partie client

a- Diagramme de séquence du cas d'utilisation « s'authentifier »

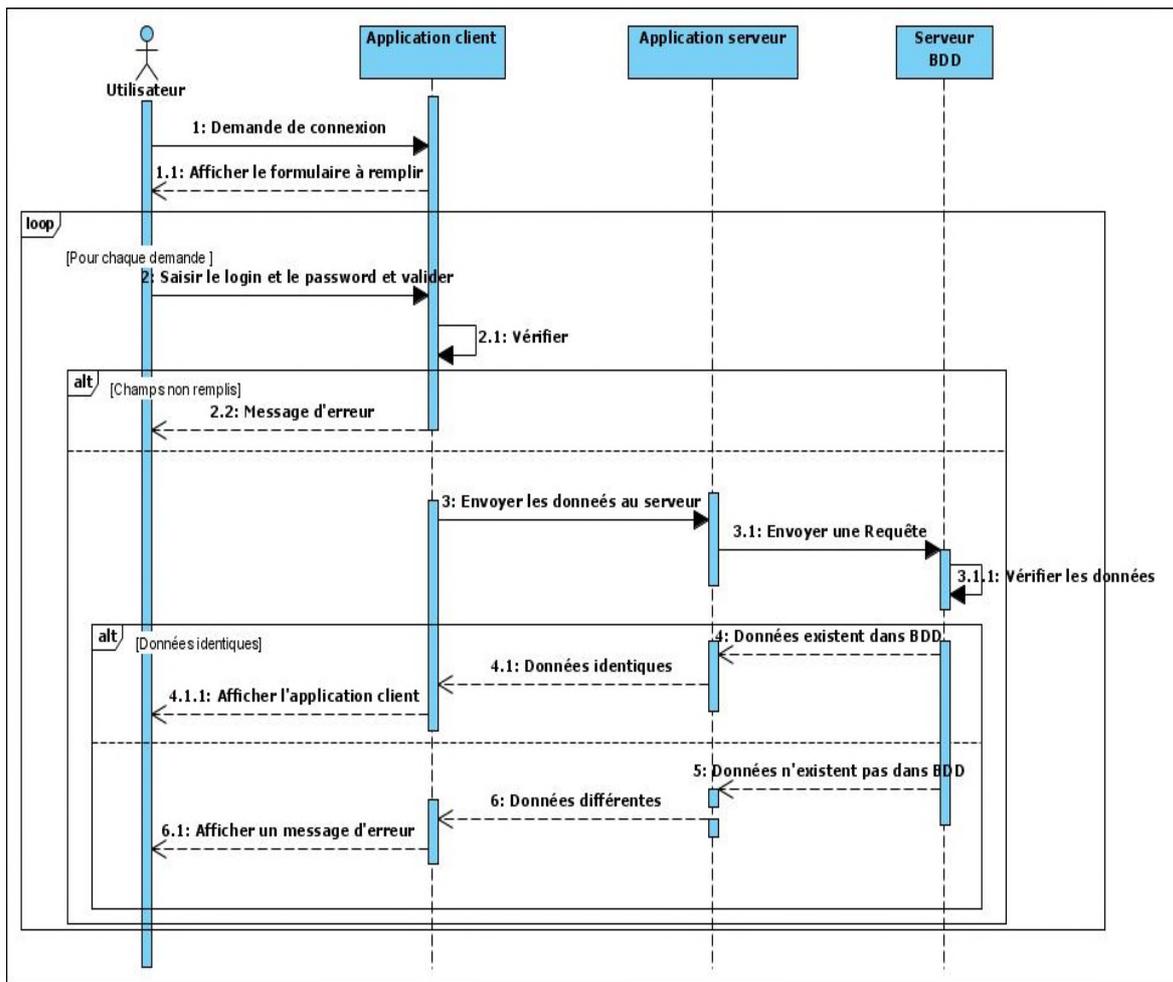


Figure 11 : Diagramme de séquence du cas d'utilisation « s'authentifier ».

b- Diagramme de séquence du cas d'utilisation « écrire un message »

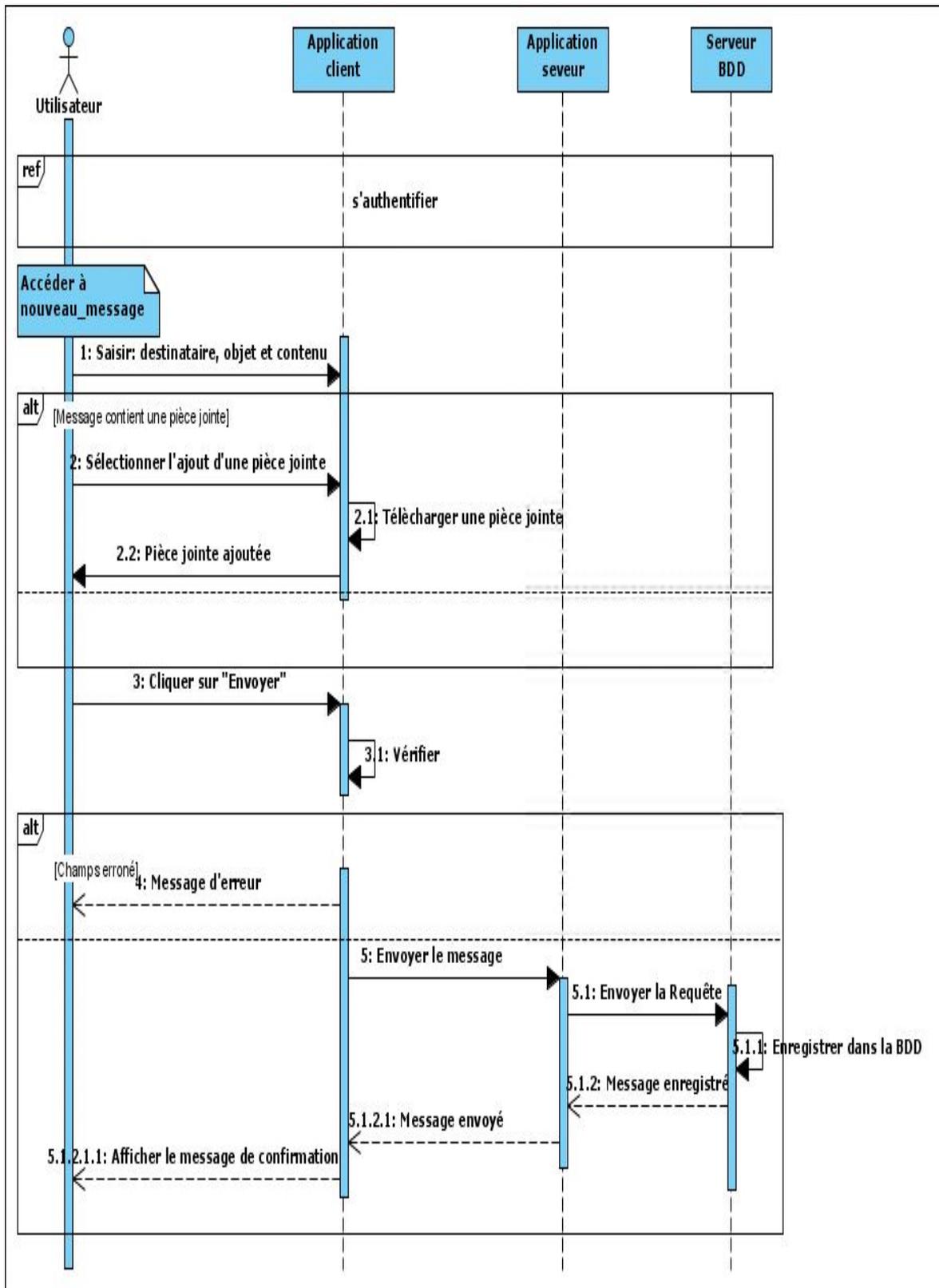


Figure12 : Diagramme de séquence du cas d'utilisation « écrire un message ».

c- Diagramme de séquence du cas d'utilisation « Consulter/lire/supprimer les messages reçus »

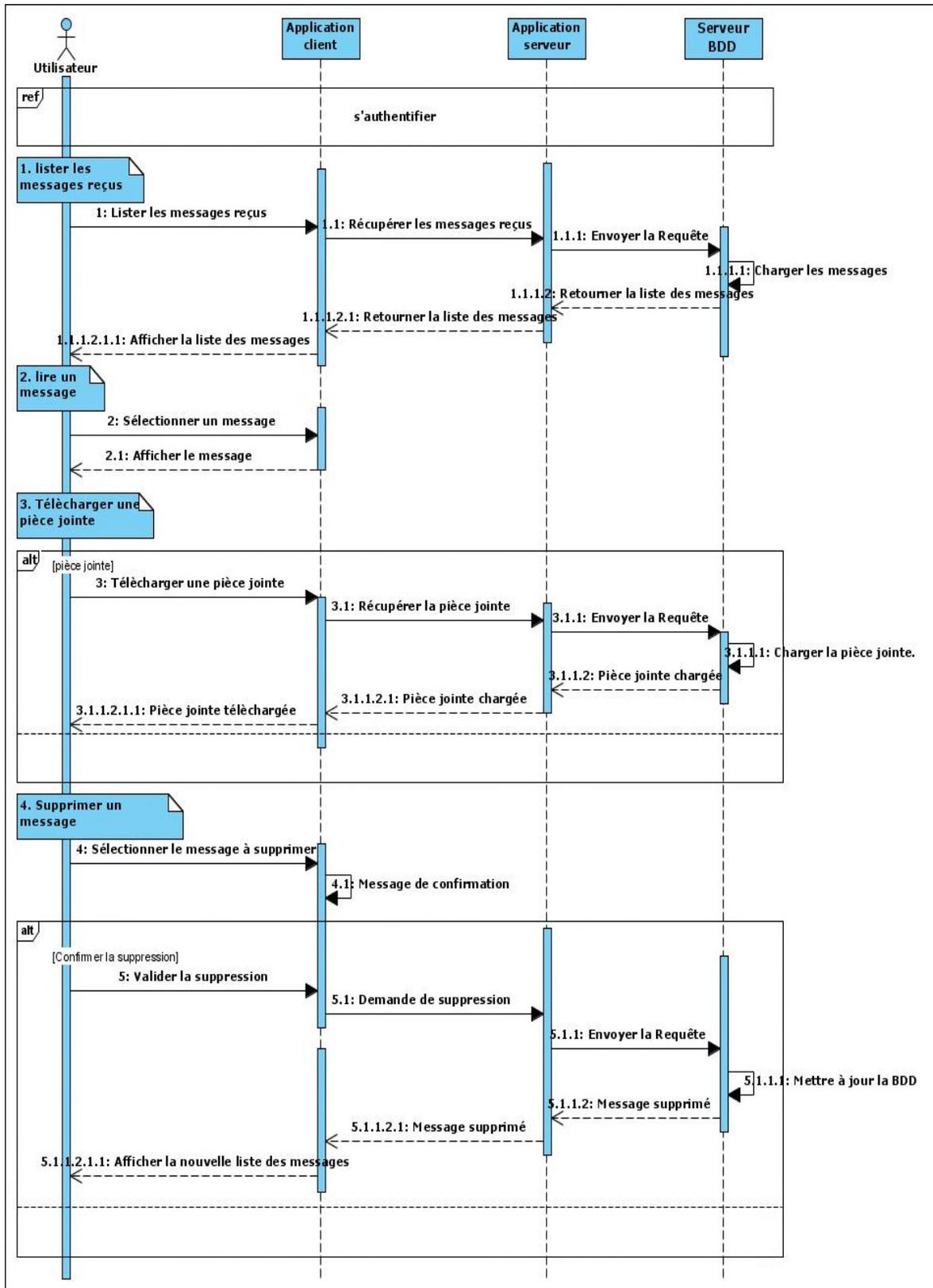


Figure13 : Diagramme de séquence de cas d'utilisation « Consulter/lire les messages reçus ».

II.6.1.2. La partie administrateur

a- Diagramme de séquence du cas d'utilisation « s'authentifier »

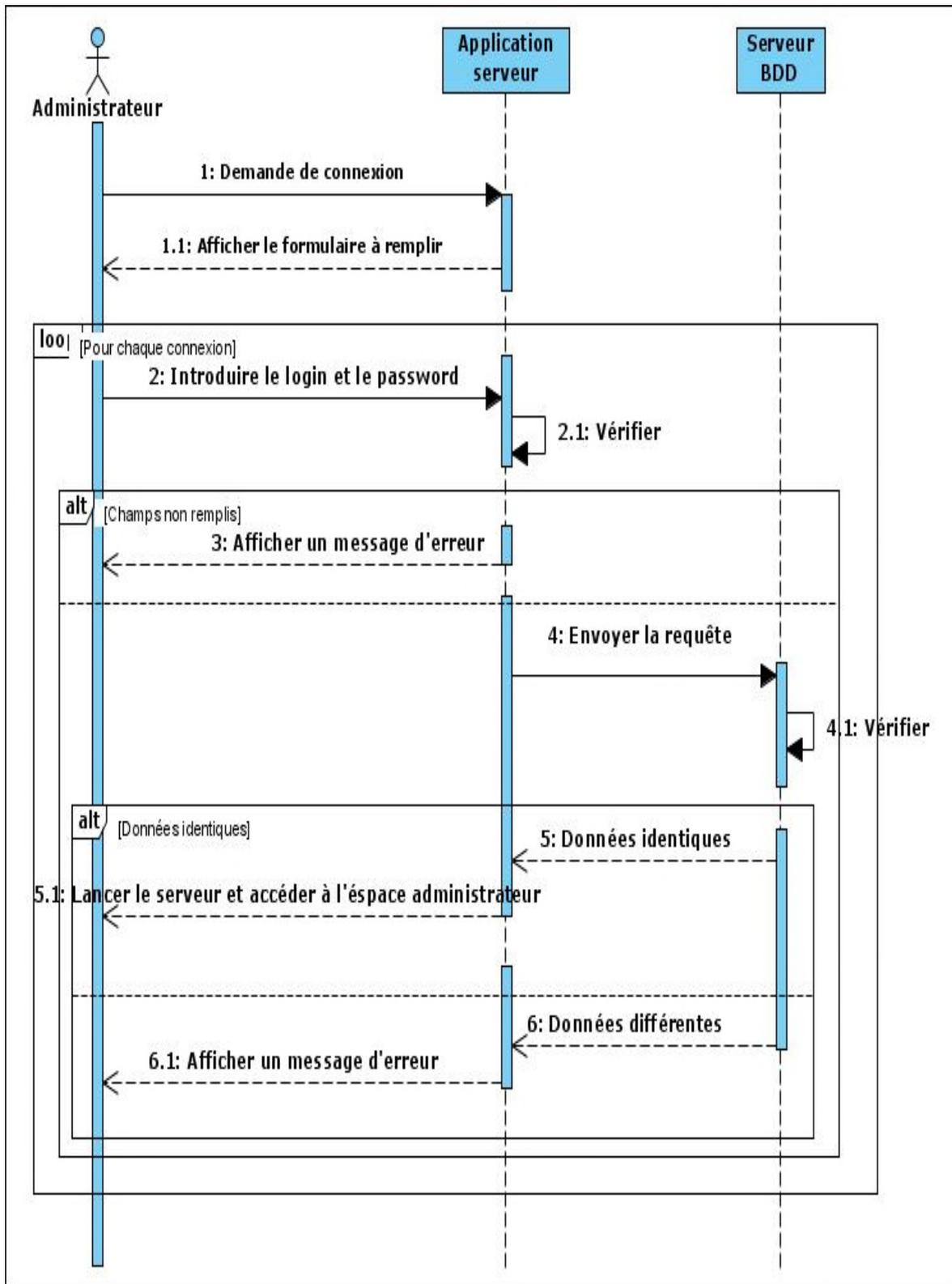


Figure14 : Diagramme de séquence du cas d'utilisation « s'authentifier ».

b- Diagramme de séquence du cas d'utilisation « supprimer/ajouter un utilisateur »

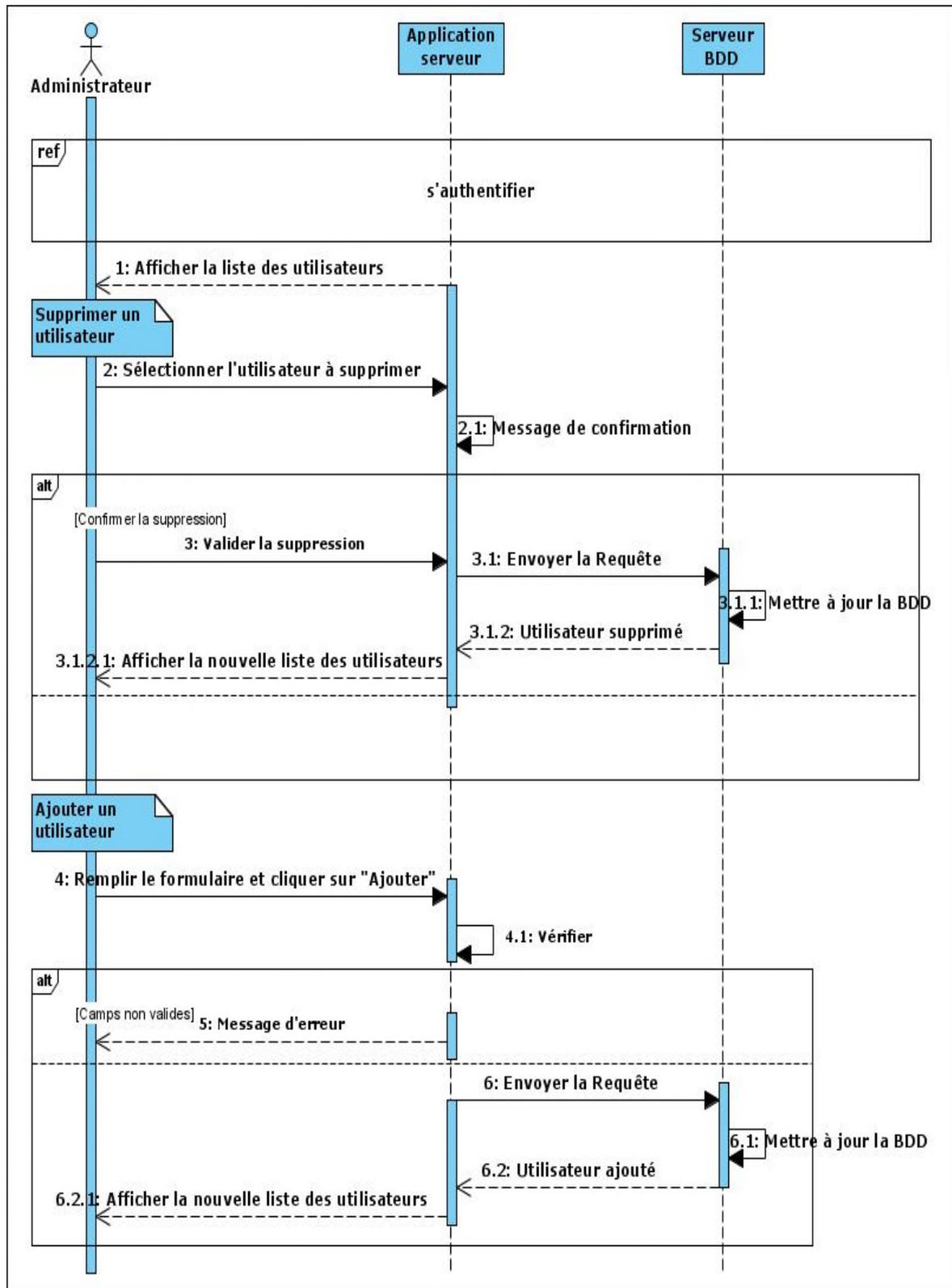


Figure15 : Diagramme de séquence du cas d'utilisation « supprimer/ajouter un utilisateur ».

II.6.2. Diagramme de classes

Les diagrammes de classes sont les diagrammes les plus connus du langage UML. Ils permettent d'appréhender, d'un point de vue logique, la structure statique du système en indiquant :

- La structure des objets composant de l'application.
- Les liens structurels entre les objets.

La réalisation du diagramme de classe se base sur le dictionnaire de données et les règles de gestion. L'analyse sémantique des données du dictionnaire permet de les regrouper dans des entités. Les liens qui les relient tiennent compte des règles de gestion.

II.6.2.1. Règles de gestion

Le diagramme de classe du système étudié est basé sur les règles de gestion suivantes :

- R01** : un utilisateur peut ne pas envoyer un message, ou envoyer plusieurs messages.
- R02** : un message peut être envoyé par un et un seul utilisateur.
- R03** : un utilisateur peut ne pas recevoir un message ou recevoir plusieurs messages.
- R04** : un message peut être reçu par un et un seul utilisateur.
- R05** : une pièce jointe est attaché à zéro ou à plusieurs messages.
- R06** : un message peut ne pas attacher une pièce jointe, comme il peut attacher une seul.
- R07** : un utilisateur peut ne pas avoir de contact, comme il peut avoir plusieurs contacts
- R08** : un contact peut ne pas être ajouté par un utilisateur, comme il peut être ajouté par plusieurs utilisateurs.

II.6.2.2. Le dictionnaire de données

Le dictionnaire de données permet de rassembler les informations nécessaires. Il précise le nom de la classe et ses attributs, désignation, le type et la taille. Le tableau suivant représente le dictionnaire de notre diagramme de classe :

| Classe | Attributs | Désignation | Type | Taille |
|--------------------|------------------|------------------------------------|------|--------|
| Utilisateur | id_utilisateur | L'identificateur de l'utilisateur. | N | 11 |
| | nom | Le nom de l'utilisateur. | A | 25 |
| | prenom | Le prénom de l'utilisateur. | A | 25 |
| | login | Le login de l'utilisateur. | AN | 25 |
| | questin_secret | Une réponse à une question. | AN | 15 |
| | password | Le mot de passe de l'utilisateur. | AN | 15 |
| | service | Le service d'un employé. | AN | 15 |
| | mail_externe | L'email de l'utilisateur. | AN | 15 |
| Message | id_msg | L'identificateur de message. | N | 10 |
| | user_source | La source de message. | AN | 25 |
| | user_destiantion | La destination de message. | AN | 25 |

| | | | | |
|---------------------|--------------|--|----|----|
| | Contenu | Le contenu de message. | AN | |
| | Objet | L'objet de message. | AN | 25 |
| | date_envoi | Date d'envoi de message. | D | |
| | heur_envoi | Heure d'envoi de message. | T | |
| | etat | Les messages lus ou non lus. | A | 6 |
| | etat_message | Message supprimé ou non supprimé. | A | 3 |
| Piece_jointe | id_piece | Identificateur de la pièce jointe | N | 11 |
| | path | Le chemin dont la pièce est enregistré dans le serveur | AN | 70 |
| Corbeille | | | | |
| Contacts | id_contact | Identificateur du contact. | N | 11 |
| | nom_contact | Nom du contact. | AN | 25 |
| | date_ajout | Date d'ajout du contact. | D | |

Tableau2 : Le dictionnaire de données.

NB : AN=alphanumérique, N=numérique, T=time, D : date, Taille : taille du champ de saisi.

II.6.2.3. Réalisation du diagramme

La figure suivante représente le diagramme de classe qui correspond au système étudié :

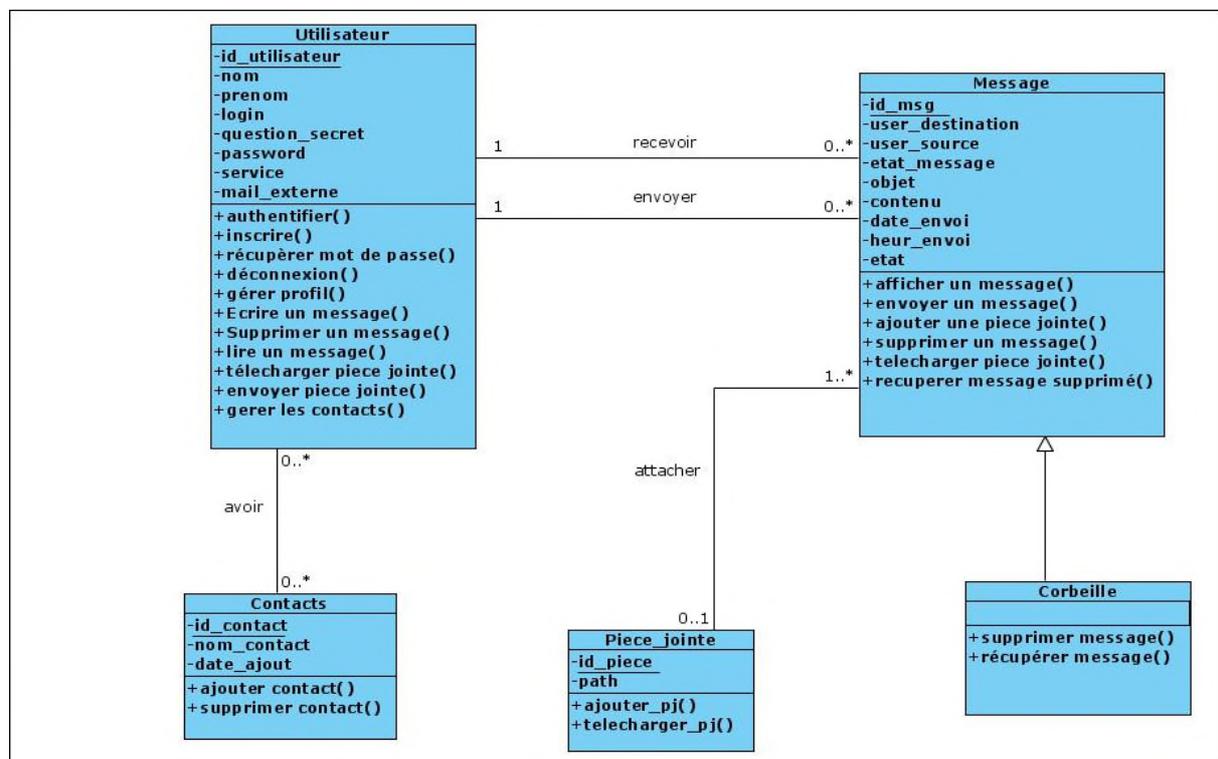


Figure16 : Diagramme de classes.

D'après les règles de passage, on a abouti au modèle relationnel suivant :

II.6.2.4. Le modèle relationnel

Utilisateur (id_utilisateur, nom, prenom, login, question_secret, password, service, mail_externe).

Message (id_msg, user_source, user_destiation, contenu, objet, date_envoi, heur_envoi, lu, etat_message, #id_user_source, #id_user_dest, #id_piece).

Piece_jointe(id_piece,path).

avoir(#id_utilisateur, #id_contact).

Contacts (id_contact, nom_contact, date_ajout).

Corbeille (#id_msg).

II.7. Conclusion

Dans cette partie, nous avons présenté le langage de modélisation UML, ses diagrammes sont utilisés dans chaque phase et activité du processus unifié. Nous avons commencé par la présentation des diagrammes des cas d'utilisation qui montrent l'enchaînement des cas d'utilisation au sein du processus ; puis les diagrammes de séquences qui montrent l'enchaînement chronologique des opérations à l'intérieur de chaque cas d'utilisation ; enfin le diagramme de classes qui est le plus précis conceptuellement.

Cette étude nous permettra d'entamer directement la réalisation de notre système de messagerie intranet avec le logiciel NetBeans qui représente l'environnement de développement.

CHAPITRE III

REALISATION

III.1. Introduction

Tout développement d'application nécessite le choix des technologies et outils adéquats pour faciliter sa réalisation. Dans ce présent chapitre, nous allons décrire en premier lieu les outils de développement utilisés. Ensuite, nous présenterons notre système de messagerie électronique et cela en illustrant les principales interfaces qu'il contient.

III.2. L'environnement de développement

III.2.1. Langages de programmation

Pour satisfaire les besoins de notre application, nous avons utilisé les langages suivants :

III.2.2.1. Le langage de programmation JAVA

C'est un langage de programmation orienté objet, développé par Sun Microsystems et dont la syntaxe est proche du C. Il permet de créer des logiciels compatibles et facilement portables sur plusieurs systèmes d'exploitation tels qu'UNIX, Windows, Mac OS ou DNU/Linux.

JAVA peut être utilisé sur internet pour développer des applications client-serveur. Côté client, les applets sont à l'origine de la notoriété du langage, Mais c'est côté serveur qui s'est imposé dans le milieu de l'entreprise grâce aux servlets [12,6].

Le Java Development Kit (JDK) désigne l'ensemble des bibliothèques logicielles de base du langage de programmation Java, ainsi que les outils avec lesquels le code peut être compilé, transformé en byte code destiné à la machine virtuelle Java.

Il existe plusieurs éditions de JDK, selon la plate-forme Java considérée :

- JSE pour la Java 2 Standard Edition également désignée J2SE.
- JEE, sigle de Java Enterprise Edition également désignée J2EE.
- JME 'Micro Edition', destinée au marché mobiles.

À chacune de ces plateformes correspond une base commune de Development Kits, plus des bibliothèques additionnelles spécifiques selon la plate-forme Java que le JDK cible [32].

III.2.2.2. Le langage SQL

Le langage SQL ou (Standard QueryLanguage) est un langage permettant d'interroger les bases de données de manière simple. Il est doté d'une syntaxe particulière que l'on doit respecter pour que la communication avec la base se passe au mieux.

Le succès du langage SQL est dû essentiellement à sa simplicité et au fait qu'il s'appuie sur le schéma conceptuel pour énoncer des requêtes en laissant le SGBD responsable de la stratégie d'exécution.

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

- la création des données.
- la manipulation des données.
- le contrôle des données.

A part le fait d'envoyer directement les requêtes SQL telles quelles au SGBD, le but ultime de l'utilisation de SQL sera aussi d'inclure ces requêtes SQL dans un programme écrit dans un autre langage tel est le cas de MySQL avec JAVA [15].

III.2.3 Outils de programmation

III.2.3.1. Netbeans IDE 8.0.2

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

NetBeans constitue par ailleurs une plate forme utilisée comme brique logicielle pour la création d'applications spécifiques (bibliothèque Swing (Java)). Les partenaires privilégiés fournissent des modules à valeurs rajoutées qui s'intègrent facilement à la Plateforme et peuvent être utilisés pour développer ses propres outils et solutions [33].

III.2.3.2. EasyPHP

EasyPHP est le premier package WAMP à voir le jour (1999). Il s'agit d'une plateforme de développement Web, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. EasyPHP n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (un serveur web Apache et un serveur de bases de données MySQL), un interpréteur de script (PHP), ainsi qu'une administration SQL phpMyAdmin. Il dispose d'une interface d'administration permettant de gérer les alias (dossiers virtuels disponibles sous Apache), et le démarrage/arrêt des serveurs [34].

III.2.3.3. Apache

C'est le serveur le plus répandu sur Internet, permettant la configuration de l'environnement d'exécution de pages web. Il s'agit d'une application fonctionnant à la base sur les systèmes d'exploitation de type Unix, mais il a désormais été porté sur de nombreux systèmes, dont Microsoft Windows grâce à sa conception modulaire (morceaux de code) qui correspond à différents aspects ou fonctions du serveur.

Cette conception autorise le développeur à choisir quelles fonctionnalités seront incluses dans le serveur en sélectionnant les modules à charger soit à la compilation, soit à l'exécution.

Elle lui permet aussi d'écrire son propre morceau de code qui pourra ensuite être facilement intégré dans le serveur Web Apache [15].

III.2.3.4. MySQL

MySQL est un système de gestion de bases de données relationnelles (SGBDR) libre fonctionnant sous diverses plates-formes telles et permettant de manipuler des instructions adressées à la base de données sous forme de requêtes SQL. Et il fait partie des logiciels de gestion de base de données les plus utilisés au monde [11].

III.2.3.5. PhpMyAdmin

PhpMyAdmin est une application écrite en PHP, qui offre une interface permettant de gérer depuis un navigateur des bases de données MySQL. Avec cet outil, on peut :

- Créer et supprimer des BDD.
- Créer, modifier et supprimer des tables.
- Ajouter, éditer et supprimer des champs (attributs) à une table.
- Exécuter des requêtes SQL.

III.3. Les principales interfaces graphiques

Avant d'entamer la phase de présentation des principales interfaces graphiques, nous tenons à préciser que la réalisation de notre système repose sur :

- L'utilisation de protocole TCP: définie dans le but de fournir un service de transfert de données de haute fiabilité et qui fonctionne en mode connecté.
- L'utilisation des sockets : pour établir un canal de communication permettant l'envoi et la réception d'informations entre des processus qui s'exécutent dans un environnement distribué.

Afin d'avoir un serveur performant pouvant traiter plusieurs requêtes à la fois, nous avons utilisé des threads.

Dans ce qui suit, nous présenterons les interfaces les plus importantes de notre système de messagerie :

III.3.1. Interfaces client

III.3.1.1. Espace d'authentification

La page d'authentification permet à un client qui est déjà inscrit de s'authentifier pour accéder à l'application client. Une fois ses informations sont reconnues, le système affiche la page appropriée. Dans le cas contraire, le système réaffiche la page d'authentification avec un message d'erreur.

The screenshot shows a web interface for client authentication. At the top, the title "Espace authentication" is centered. Below it, there are two input fields: "Login:" with the value "katia_bourdache@informatique" and "Password:" with masked characters "••••••". To the right of these fields are two buttons: "Connexion" and "Quitter". Below the buttons, there are two links: "->Inscription" and "->Mot de passe oublié?". At the bottom of the interface, there is a separate form with three input fields: "Nom:", "Prenom:", and "Quel est le nom de votre ami préféré :". Below these fields are two buttons: "Confirmer" and "Annuler".

Figure17 : Interface d'authentification d'un client.

III.3.1.2. Boite de réception

Une fois le client est authentifié, le système affiche la boite de réception pour pouvoir visualiser la liste des messages reçus.

Dans cette partie, le client peut lire les messages, télécharger une pièce jointe, supprimer le message ou le fermer.

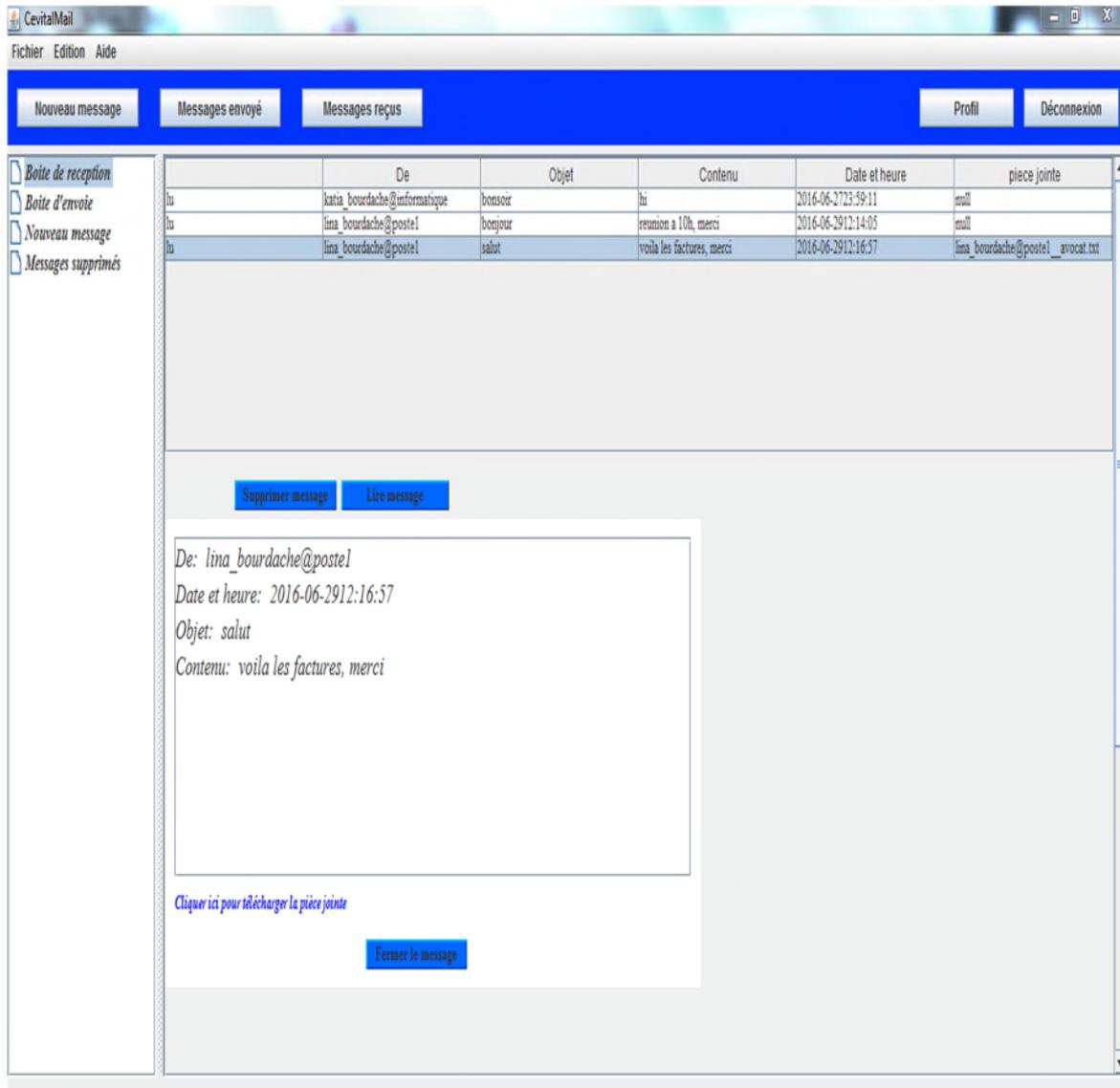


Figure18 : interface de la boîte de réception.

III.3.1.3. Boîte d'envoi

C'est la partie où un client peut voir la liste des messages envoyés, lire un message ou le supprimer.

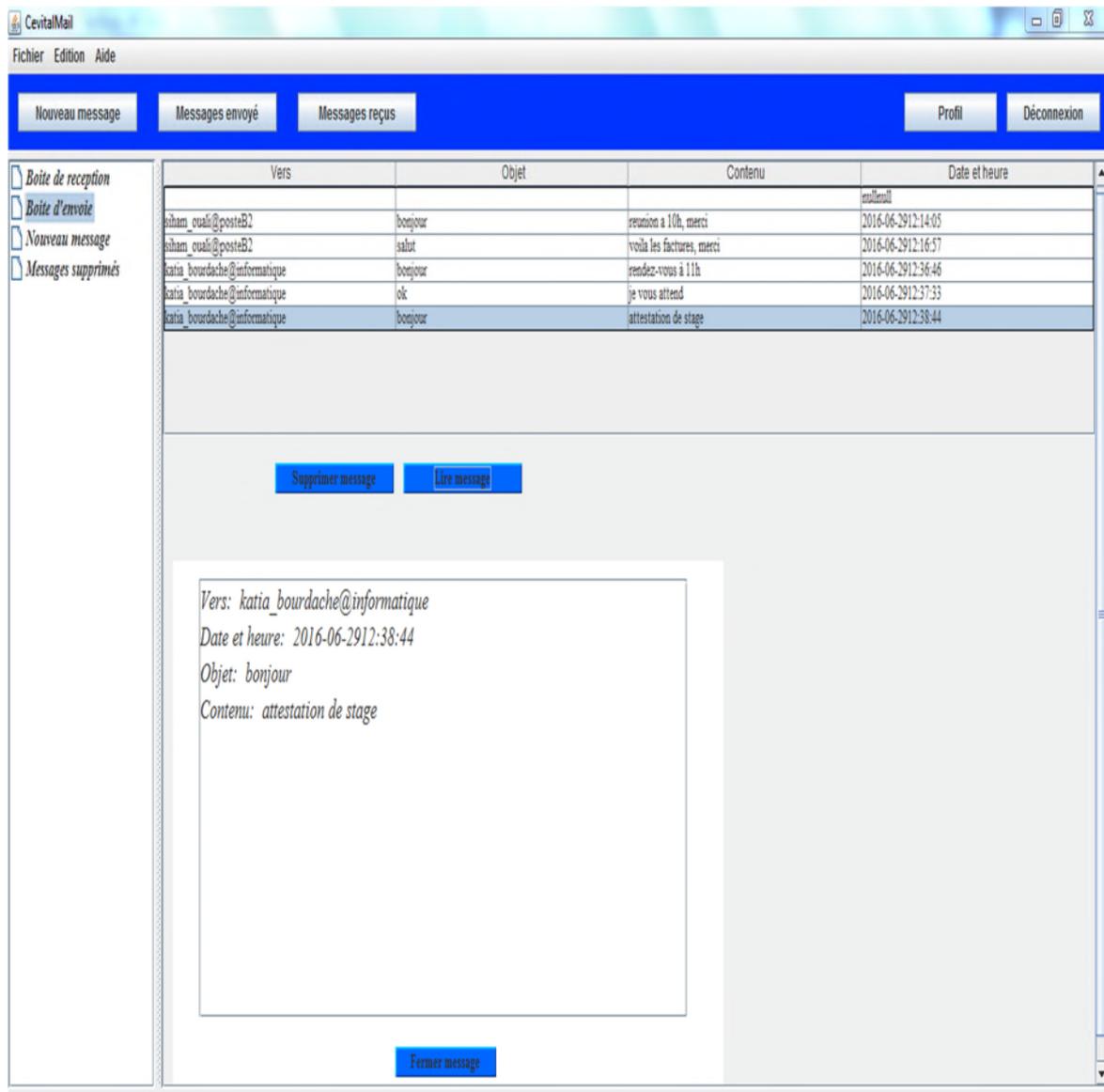


Figure19 : interfacede la boite d’envoi.

III.3.1.4. Nouveau message

Dans cette page, le client peut envoyer un message au destinataire souhaité en saisissant son login, ou simplement le sélectionné dans la partie ‘liste des contacts’, il peut aussi attacher une pièce jointe à son message.

Dans la liste des contacts, le client peut effectuer de différentes mises à jour, comme ajouter ou supprimer un contact.

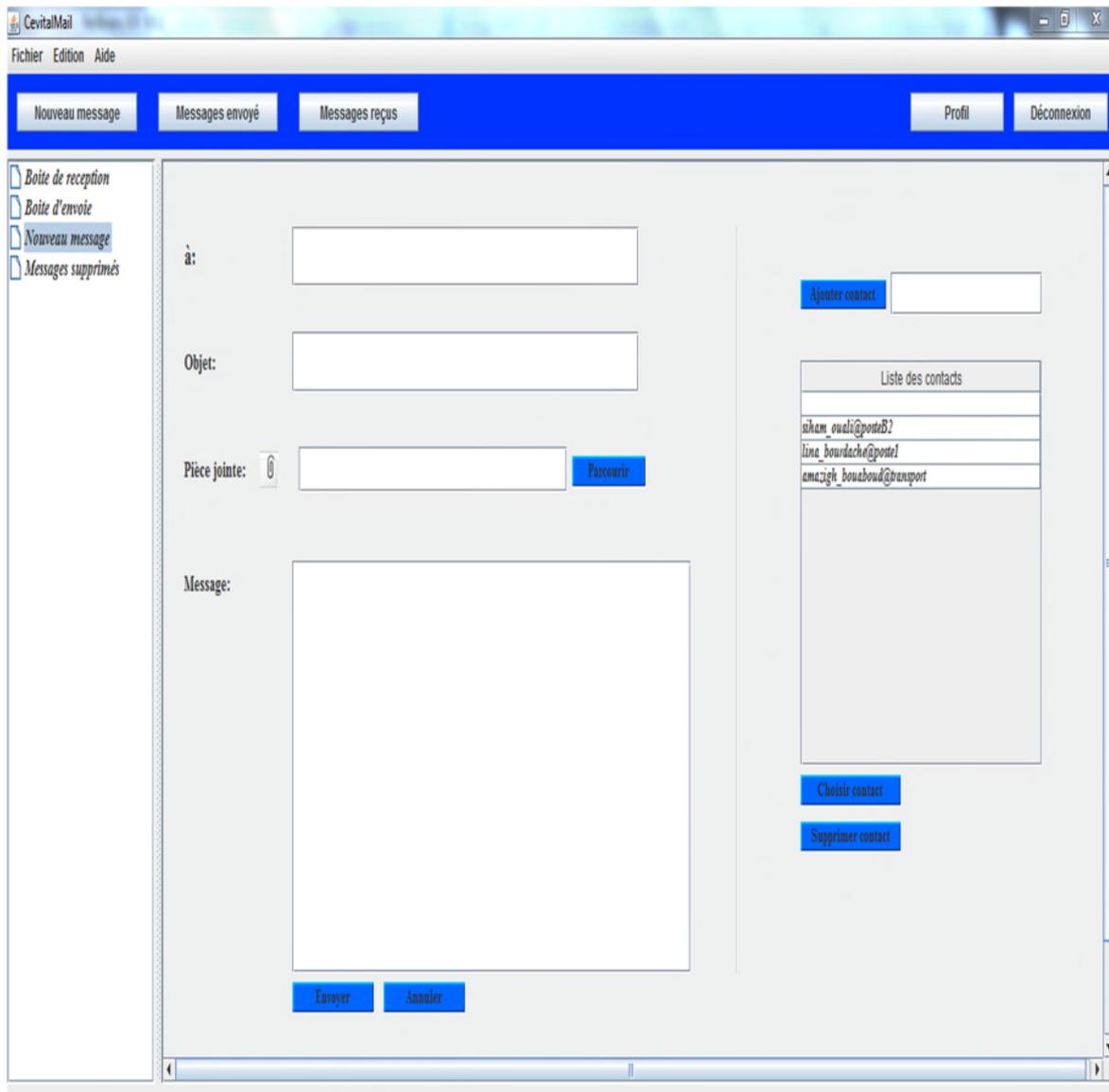


Figure20 : interface de nouveau message.

III.3.1.5. Messages supprimés

Cette page permet à un client de supprimer un message définitivement de sa boîte email, ou de le récupérer en cas de besoin.

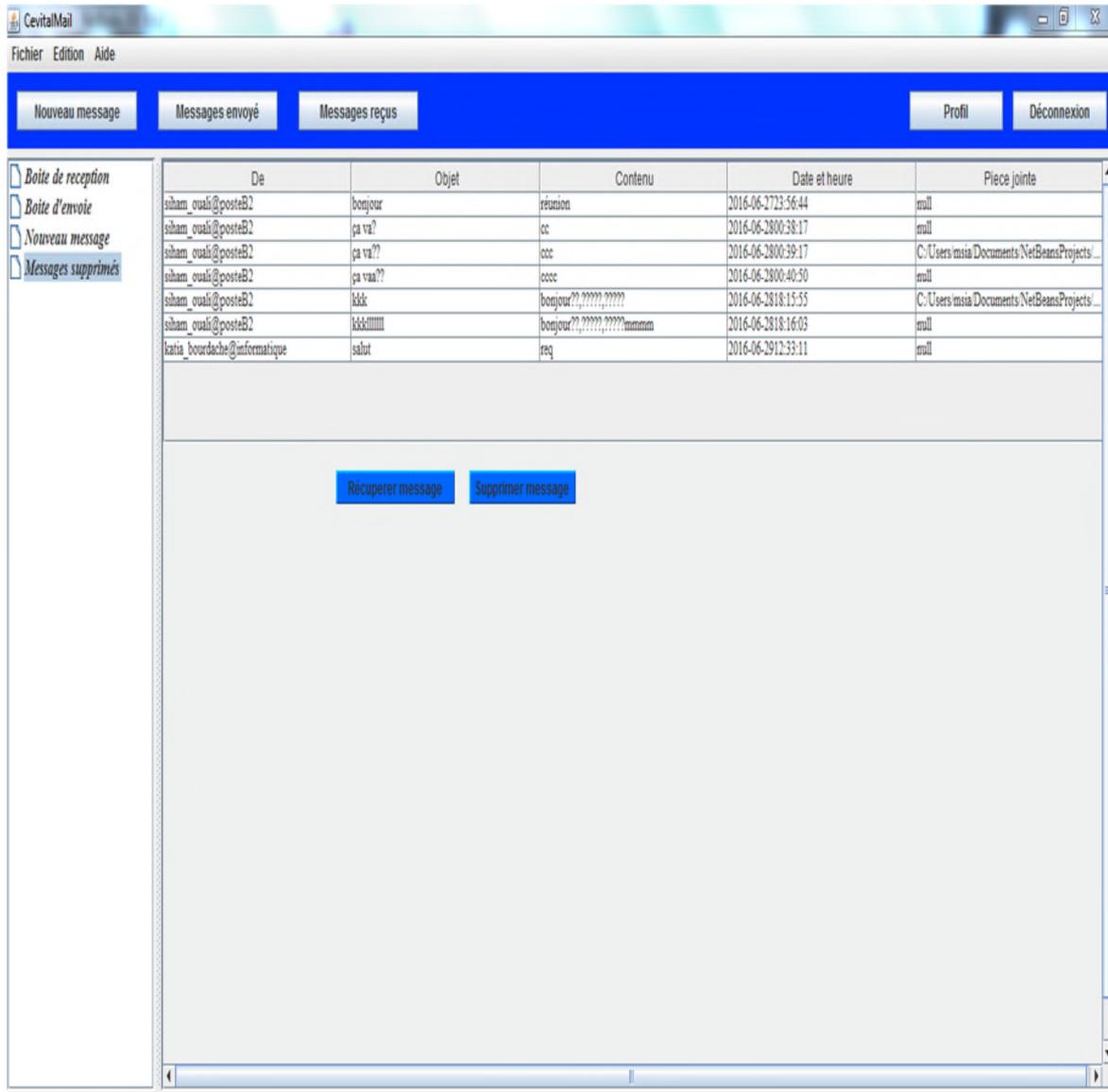


Figure21 : interface des messages supprimés.

III.3.1.6. Gérer le profil

Permet à un client de consulter ses informations personnelles et de les modifier.

The screenshot shows the CevitalMail web application interface. At the top, there is a navigation bar with buttons for 'Nouveau message', 'Messages envoyés', 'Messages reçus', 'Profil', and 'Déconnexion'. Below this, a sidebar on the left contains a tree view with folders: 'Boîte de réception', 'Boîte d'envoi', 'Nouveau message', and 'Messages supprimés'. The main content area is divided into two sections. The left section is a form for updating profile information with the following fields and values:

| | | |
|--|---|---|
| Nom : | <input type="text" value="bourdache"/> | |
| Prenom : | <input type="text" value="kalia"/> | |
| Login : | <input type="text" value="kalia_bourdache@informatique"/> | <input type="button" value="Modifier"/> |
| Service : | <input type="text" value="informatique"/> | <input type="button" value="Modifier"/> |
| Quel est le nom de votre ami préféré : | <input type="text" value="amazigh"/> | <input type="button" value="Modifier"/> |
| Email externe : | <input type="text" value="kalia@hotmail.com"/> | <input type="button" value="Modifier"/> |

Below the form is a blue button labeled 'Enregistrer les modifications'. The right section is titled 'Modifier le mot de passe' and contains three password input fields with masked characters (dots):

- Mot de passe:
- Nouveau mot de passe:
- confirmer nouveau mot de passe:

At the bottom of this section are two blue buttons: 'Confirmer' and 'Annuler'.

Figure22 : interface gérer le profil.

III.3.2. Interfaces administrateur

La figure ci-dessous, présente l'application serveur, c'est dans cette interface là que l'administrateur peut lancer le serveur en cliquant sur bouton « Connecter le serveur »,

L'application gère aussi la gestion des utilisateurs, l'administrateur peut consulter la liste des utilisateurs et effectuer les différentes mises à jour : ajouter un utilisateur, modifier ses informations ou de le supprimer de la base de données.

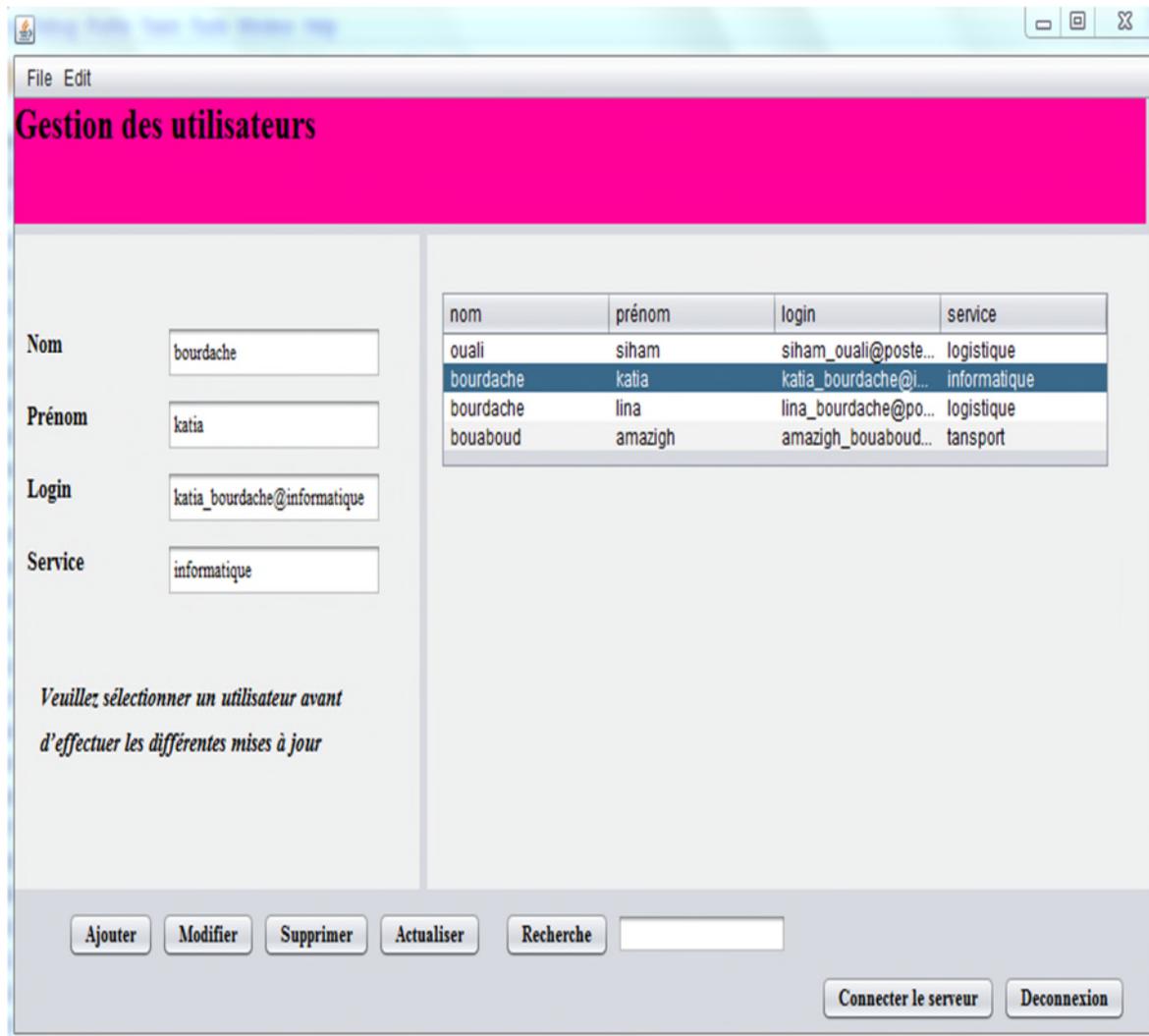


Figure23 : interface d'accueil de l'administrateur.

III.4. Conclusion

A travers ce chapitre, nous avons cité les différents outils de développement de notre système et quelques interfaces graphiques que nous avons jugé plus importantes ainsi ses différentes fonctionnalités tout en respectant la conception élaborée.

Conclusion générale et

perspectives

CONCLUSION GENERALE ET PERSPECTIVES

Notre stage de fin de cycle, s'est déroulé au sein de l'entreprise Cevital Agro-alimentaire à Béjaïa. Notre travail consiste à réaliser un système de messagerie électronique en intranet, l'objectif principal est de rendre service aux employés de l'entreprise. Notre application offre la possibilité d'échanger des messages et de partager des documents via une pièce jointe, ce qui rend leur travail plus coopératif, en assurant une meilleure coordination entre les services.

Durant la réalisation de ce projet, nous avons essayé de couvrir le maximum de besoins et d'assurer l'efficacité des fonctionnalités du système tout en ayant une interface conviviale et facile à utiliser.

Pour la réalisation de ce travail, nous avons opté pour l'utilisation de langage Java qui nous a permis d'exploiter des notions de programmation orientée objet, nous avons également fait usage de nouveaux outils de développement tels que NetBeans et EasyPhp.

A la fin de la réalisation de ce mémoire, nous avons accumulé une masse importante de connaissances aussi bien sur le plan théorique que sur le plan pratique, et cela en ayant l'opportunité de mettre en pratique nos connaissances acquises durant notre parcours universitaire, de profiter de l'environnement professionnel et des compétences de cet organisme par l'échange d'idées et d'informations.

Ainsi nous pensons avoir répondu aux besoins exprimés par l'entreprise et nous souhaitons que le système développé puisse servir comme un outil d'aide et de documentation pour les étudiants à venir.

Notre projet reste ouvert aux autres extensions, et peut être enrichi par :

- Lister les contacts selon les services disponibles.
- La diffusion de messages à un certain groupe de service.
- Ajout de la fonction transférer et répondre à un message.
- La possibilité de passer des appels vidéo entre les utilisateurs.

BIBLIOGRAPHIE

- [1] DEBRAUWER L., « UML2 Modélisation des objets », Amazon France, éditions ENI, 2006.
- [2] GABAY J. et GABAY D., « Mise en œuvre guidée avec études de cas », Paris, édition : Dunod, 2008.
- [3] JULIARD F., « UML Unified Method Language », Journal Université de Bretagne Sud UFR SSI-IUP Vannes, édition : 2001-2002.
- [4] MORLY C., HUNGUES J. et BERLAND B., « UML pour l'analyse d'un système d'information », édition: Dunod, 2002/2006.
- [5] ROQUES P. et VALLÉE F., « UML en action de l'analyse des besoins à la conception en Java », Groupe Eyrolles, 2003, 2^{ème} édition, 2003.
- [6] Claude D., « Programmer en java », Paris, 5^{ème} édition d'EYROLLES, 2008.
- [7] SIDER A., Chapitre 5: « Service de messagerie Internet et Protocoles associés » [PDF], Béjaïa, 2015-2016, www.univ-bejaia.dz.
- [8] NKENLIFACK M J., « Ingénierie des Systèmes Logiciels » [PDF], Université de Doual, 2009.
- [9] AUDIBERT L., « Introduction à la modélisation objet », Institut Universitaire de Technologie de Villetaneuse: Département Informatique, Villetaneuse, 2006, www-lipn.univ-paris13.fr.
- [10] VEHENT J., « Intranet pour la sécurité des systèmes d'information » [PDF], Université de Poitiers, Avril 2007.
- [11] AUDIBERT L., « Base de Données et langage SQL », Institut Universitaire de Technologie de Villetaneuse : Département Informatique, Villetaneuse.
- [12] CYRILL H., « Apprenez à programmer en java, la programmation professionnelle à la portée de tous », France, 2^{ème} édition, imprimé par Imprim Vert, Novembre 2012.
- [13] MOKRANI B. et SADI C. et *al.*, « Configuration des protocoles de routage RIPv2 et OSPF pour une communication inter VLAN au sein d'un réseau WAN », mémoire de fin d'étude : Spécialité Informatique, Université Abderrahmane. MIRA, Béjaïa, juin 2014.
- [14] BOURDACHE K. et OUALI S. et *al.*, « Conception et réalisation d'une application web pour la gestion d'une salle des fêtes », mémoire de fin d'étude : Spécialité Informatique, Université Abderrahmane. MIRA, Béjaïa, juin 2014.
- [15] BONI W., ENDERSON Y., « Mise en place d'un système de messagerie: cas de l'ITA (Institut des technologies d'Abidjan) », mémoire de fin d'étude : Informatique et télécommunications, Institut des technologies d'Abidjan, 2011.
- [16] BELKHICHENE A. et TOUAHRI K. et *al.*, « Conception et réalisation d'une application pour la gestion des prêts immobiliers », mémoire de fin d'étude : Spécialité Informatique, Université Abderrahmane. MIRA, Béjaïa, juin 2012.

WEBLIOGRAPHIE

- [17] www.bastien.smeuh.org
- [18] www.commentcamarche.net
- [19] www.courstechinfo.be
- [20] www.frameip.com
- [21] www.hautrive.free.fr
- [22] www.igm-univ-mlv.fr
- [23] www.laissus.developpez.com
- [24] www.les-infostrateges.com
- [25] www.lifl.fr
- [26] www.mescoursinfo.free.fr
- [27] www.openclassrooms.com
- [28] www.opendoc.net
- [29] www.supinfo.com
- [30] www.viviani.org
- [31] www.ybet.be
- [32] www.code4pi.com
- [33] www.netbeans.org
- [34] www.geronimo.developpez.com

Résumé

La communication et le système de transmission de l'information sont devenus des moyens à grande importance pour les entreprises. Nous avons fait un tour sur l'une des plus importantes technologies de communication qui est la messagerie électronique. Ce travail a pour objectif la conception et la réalisation d'un système de messagerie électronique en intranet, qui permet de relier les employés de l'entreprise Cevital dans le but de faciliter leur communication, le partage des ressources et la possibilité de la mise en place d'un travail collaboratif. Pour l'implémentation de notre application, nous avons défini quelques concepts de base et généralités sur les réseaux, pour passer ensuite à la modélisation par la méthode UML guidée par le processus 2TUP et enfin l'utilisation de langage de programmation Java.

Mots clés: communication, information, messagerie électronique, intranet, réseaux, modélisation, UML, 2TUP, Java.

Abstract

The communication and the transmission system of the information became ways with big importance for companies. We made a turn on one of the most important communication technologies that is e-mail. This work aims at the conception and the realization of an electronic mail system in intranet, which links the Cevital company employees in order to facilitate communication, the resource sharing and the possibility of the establishment of collaborative work. For the implementation of our application, we define some basic concepts and general information on networks, and then move to modeling by the UML method guided by 2TUP process and finally the use of Java programming language.

Keywords : communication, information, e-mail, intranet, networks, modeling, UML, 2TUP, Java.