République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Faculté des Sciences Exactes
Département d'Informatique

Mémoire de fin de cycle
En Vue De L'obtention Du Dipléme du
Master professionel en Informatique

Option: Administration et sécurité des réseaux

Thème

Conception et réalisation d'un système de messagerie interne Cas:"IFRI"

Réalisé par

M^{elle} *OUAZAR* Ouezna
M^{elle} *REDJDAL* Lamia

Encadré par: M^{r} *MEHAOUED* Kamel
Président: M^{r} *AISSANI* Sofiane
Examinateur: M^{r} *SELLAMI* Lynda

Année Universitaire 2015 – 2016

Résumé

La messagerie électronique est l'un des outils les plus répandus dans l'Internet des entreprises ou des particuliers. C'est outil indispensable et le mode de communication le plus utilisé dans les milieux professionnels.

Elle est beaucoup plus développée dans les organisations aux cours de ces dix dernières années, grâce à sa facilité d'utilisation et son utilité perçue.

Notre projet consiste à réaliser un système de messagerie pour l'entreprise d'agro-alimentaire IFRI, un client de messagerie qui permettra l'échange des mails et des différents documents entre les différents utilisateurs en utilisant un serveur de messagerie approprié.

Pour bien mené notre système, nous avons opté pour la méthode d'analyse et de conception de Grady Booch, le langage de programmation Java pour le développement de notre client et le serveur Apache James comme serveur de messagerie après l'avoir configuré.

Mots-clés : La messagerie électronique, un système de messagerie, un client de messagerie, un serveur de messagerie, Grady Booch, Java, Apache James.

Abstract

Email is one of the most popular tools in the internet companies or individuals. It is an essential tool and the mode of communication most used in professional circles.

It is much more developed in organizations with over the last decade, because of its ease of use and perceived usefulness.

Our project is to provide a messaging system for corporate agribusiness IFRI, a mail client that will allow the exchange of mail and various documents between users using an appropriate mail server.

To carried out our system, we opted for the method of analysis and design of Grady Booch, the Java programming language for developing our client and the server as Apache James mail server after setting.

Key-words : E-mail, messaging system, an email client, a mail server, Grady Booch, Java, Apache James.

Remerciements

Tous d'abord, nous tenons à remercier le bon Dieu de nous avoir accordé toute la détermination, la volonté et la force pour qu'on puisse réaliser ce modeste travail.

Nous tenons à exprimer notre gratitude pour notre encadreur Mr MEHAOUED.K qui nous a orienté et conseillé tout au long de ce travail.

Nous remercions aussi le personnel de " IFRI ", de nous avoir acceptés au sein de leur organisme pour effectuer notre stage et leur aimable accueil et pour leur disponibilité.

Nous remercions tout particulièrement les membres de jury qui ont accepté de juger notre travail ainsi que tous les enseignants qui ont contribué à notre formation.

Enfin, Nous remercions aussi nos parents 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 tiens à dédier vivement ce modeste travail à mes très chers parents auxquels je dois ma réussite et auxquels je ne rendrai jamais assez. Je leurs souhaite une longue vie.

A mes soeurs Kahina et sa famille sans oublier le nouveau né "Mazyane".

A mes soeurs Tinou, Kenza, et surtout notre petite adorable Katia et touta ma famille.

A toutes mes amies et leurs familles.

A ma binôme Ouezna et toute sa famille.

A ceux qui m'ont poussé à aller toujours de l'avant.

REDJDAL Lamia

Dédicaces

Je tiens à dédier vivement ce modeste travail à mes très chers parents auxquels je dois ma réussite et auxquels je ne rendrai jamais assez. Je leurs souhaite une longue vie.

A mes fréres Boudjemaa et Djamal.

A ma chére grande-mére et toute ma famille.

A toutes mes amies et leurs familles.

A ma binôme Lamia et toute sa famille.

A ceux qui m'ont poussé à aller toujours de l'avant.

OUAZAR Ouezna

Table des Matières

Liste des figures

Liste des tabeaux

Liste des abréviations

API	**A**pplication **P**rogramming **I**nterfaces
FTP	**F**ile **T**ransfert **P**rotocol
HTTP	**H**yper **T**ext **T**ransfert **P**rotocol
ICMP	**I**nternet **C**ontrol **M**essage **P**rotocol
IDE	**I**ntergrated **D**evelopment **E**nvironment
IMAP	**I**nternet **M**ail **A**ccess **P**rotocol
IP	**I**nternet **P**rotocole
JAF	**J**avaBeans **A**ctivation **F**ramework
JAMES	**J**ava **A**pache **M**ail **E**nterprise **S**erver
JRE	**J**ava **R**untime **E**nvironment
LAN	**L**ocal **A**rea **N**etwork
MAN	**M**etropolitan **A**rea **N**etwork
MDA	**M**ail **D**elivery **A**gent
MIME	**M**ultipurpose **I**nternet **M**ail **M**xtensions
MTA	**M**ail **T**ransfert **A**gent
MUA	**M**ail **U**ser **A**gent
OOA	**O**bject **O**riented **A**nalysis
OOAD	**O**bject **O**riented **A**nalysis and **D**esign
OOD	**O**bject **O**riented **D**esign
OOP	**O**bject **O**riented **P**rogramming
OSI	**O**pen **S**ystems **I**nterconnection
PAN	**P**ersonal de **A**rea **N**etwork
POP	**P**oste **O**ffice **P**rotocol
RBLs	**R**ealtime **B**lackhole **L**ist
SMTP	**S**imple **M**ail **T**ransfert **P**rotocol
SE	**S**tandard **E**dition
SSL	**S**ecure **S**ocket **L**ayer
TCP	**T**ransport **C**ontrol **P**rotocol

TLS	**T**ransport **L**ayer **S**ecurity
UDP	**U**ser **D**atagram **P**rotocol
WAN	**W**ide **A**rea **N**etwork

Introduction générale

Introduction générale

Les technologies de l'information et de la communication représentent la révolution la plus importante et la plus innovante qui a marqué la vie de l'humanité en ce siècle passé. En effet, elles viennent nous apporter de multiples conforts à notre mode de vie en révolutionnant le travail des individus par leur capacité de traitement de l'information, d'une part, et de rapprochement des distances d'une autre part.

Parmi ces technologies, la messagerie électronique, est assez développée dans les organisations aux cours de ces dix dernières années, grâce à sa facilité d'utilisation et son utilité perçue.

La messagerie électronique est devenue l'un des outils les plus répandus dans l'Internet des entreprises ou des particuliers. C'est un service gratuit qui constitue un moyen de communication privilégié entre des personnes à travers Internet. Elle est utilisé pour des applications très variées: personnelles, professionnelles, associatives, politiques, etc..., la messagerie occupe une place de plus en plus primordiale par rapport aux moyens de communication traditionnels. Outre son faible coût, la messagerie électronique a l'avantage d'optimiser la communication et la diffusion d'informations ce qui la rend indispensable au sein d'une entreprise.

L'entreprise d'agro-alimentaire IFRI, comme une entreprise à grande de taille avec ces zones industrielles et ces différents départements a besoin d'un moyen de communication plus rapide et plus sécurisé tel un système de messagerie. Pour cela, nous allons réaliser un système de messagerie interne au sein de cette entreprise.

Afin de réaliser notre système, on s'est inspiré de la méthode de Grady Booch qui est une méthode d'analyse et de conception orienté objet de développement logiciel qui nous permettra de mener une bonne architecture à notre système. Pour l'implémentation, on a opté pour Java comme langage de programmation et pour Apache james server comme serveur de messagerie.

Dans le but de mieux réaliser notre projet, nous avons subdivisé notre travail en quatre chapitres principaux. Dans le premier chapitre intitulé " Généralités ", nous avons donné des généralités sur les réseaux informatiques et plus particulièrement sur les concepts clés de la messagerie électronique.

Dans le second chapitre intitulé " Présentation de l'organisme d'accueil ", nous

avons présenté l'organisme de l'entreprise d'agro-alimentaire IFRI, sa structure ses fonctionnalités et son architecture réseau.

Quant au troisième chapitre " Analyse et Conception ", il est consacré à présenter l'analyse et la conception de notre système après avoir défini la méthode d'analyse et de conception de Grady Booch et ses concepts de l'orienté objet.

Tandis que dans le quatrième et dernier chapitre nommé " Réalisation ", on a identifié les outils et environnement de développement et les étapes de réalisation de notre projet.

Enfin, nous terminons notre mémoire par une conclusion générale.

1

Généralités

1.1 Introduction

La messagerie électronique, appelée aussi «electronic-mail» ou «E-mail» est l'outil le plus répandu dans l'Internet des entreprises ou des particuliers. C'est une architecture client/serveur se basant sur le modèle TCP/IP. Et comme tout système informatique, la messagerie se trouve face à des risques qui touchent à sa sécurité, et pour cela il existe plusieurs solutions pour éviter ces risques.

1.2 Les réseaux informatiques

1.2.1 Définition d'un réseau informatique

Un réseau informatique est un ensemble de moyens matériels et logiciels mis en oeuvre pour assurer les communications entre ordinateurs, stations de travail et terminaux informatiques [1].

1.2.2 Les types des réseaux informatiques

1.2.2.1 Les PAN (Personal Area Network)

Désigne un type de réseau informatique restreint en termes d'équipements, généralement mis en oeuvre dans un espace d'une dizaine de mètres. D'autres appellations pour ce type de réseau sont: réseau domestique ou réseau individuel [2].

1.2.2.2 Les 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 par un réseau, souvent à l'aide d'une même technologie. La vitesse de transfert de données d'un réseau local peut s'échelonner entre 10 Mbps et 1 Gbps. La taille d'un réseau local peut atteindre jusqu'à 100 voire 1000 utilisateurs [3].

1.2.2.3 Les MAN (Metropolitan Area Network)

Il s'agit d'un réseau dont la couverture s'étale à une ville. Le principe est de relier les différents réseaux LAN entre eux. Le principe est identique à celui d'un réseau local, mais les normes de transmissions sont différentes. Un MAN ou réseau métropolitain est une série de réseaux locaux interconnectés à l'échelle d'une ville [4].

1.2.2.4 Les WAN (Wide Area Network)

Un réseau WAN interconnecte des réseaux LAN et MAN pour assurer une couverture et une interconnexion au niveau d'un pays, voire du monde. Ce type de réseau utilise les satellites pour certaine interconnexion [4].

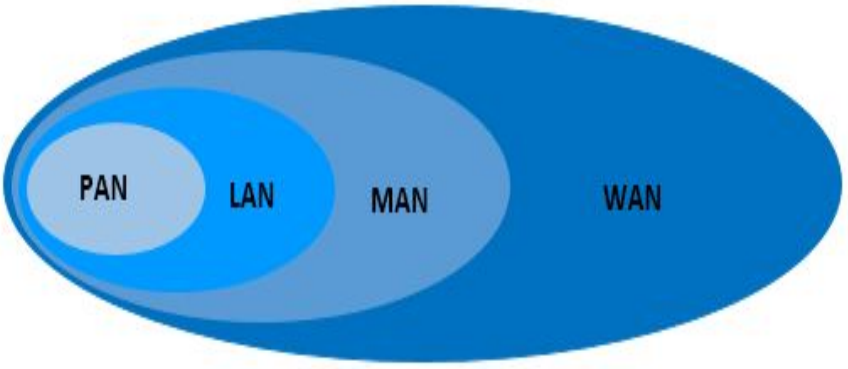

Figure 1.1: Les types des réseaux informatiques.

1.2.3 L'architecture Client/serveur

1.2.3.1 Définition

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

On parle de fonctionnement logiciel dans la mesure où cette architecture est basée sur l'utilisation de deux types de logiciels, à savoir un logiciel serveur et un logiciel client s'exécutant normalement sur 2 machines différentes [5].

1.2.3.2 Les types de l'architecture client/serveur

- **Architecture à deux niveaux**
  L'architecture à deux niveaux 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 (sans passer par un service intermédiaire) [6].

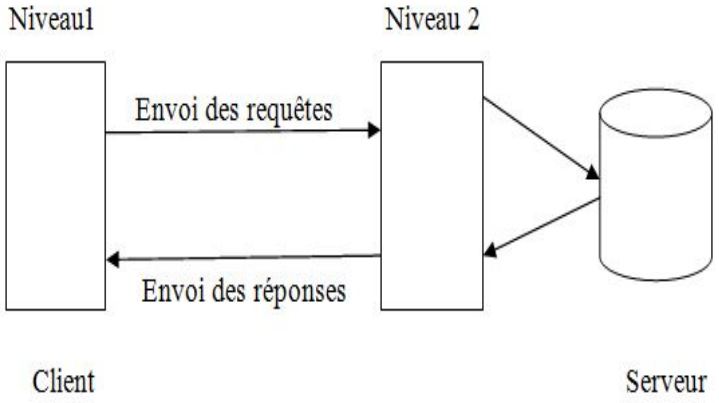

Figure 1.2: Architecture à deux niveaux

- **Architecture à trois niveaux**
  Dans l'architecture à 3 niveaux, il existe un niveau intermédiaire, c'est-à-dire que l'on a généralement une architecture partagée entre :[6]

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

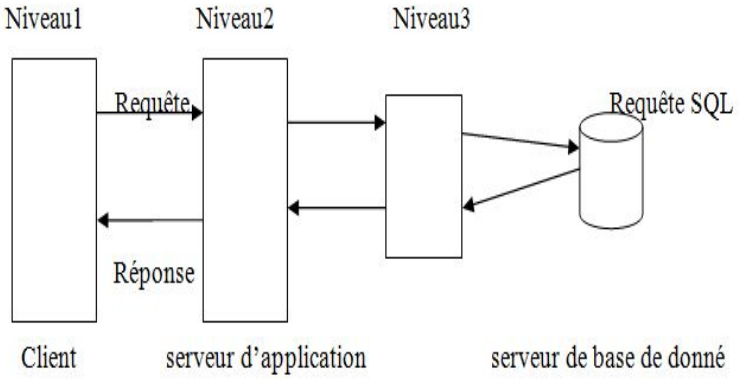

Figure 1.3: Architecture à trois niveaux.

- **Architecture multi-niveaux**
  Dans l'architecture à N niveaux, chaque serveur (niveaux 2 et 3) effectue une tâche (un service) spécifique. Un serveur peut donc utiliser les services d'un ou plusieurs autres serveurs afin de fournir son propre service [6].

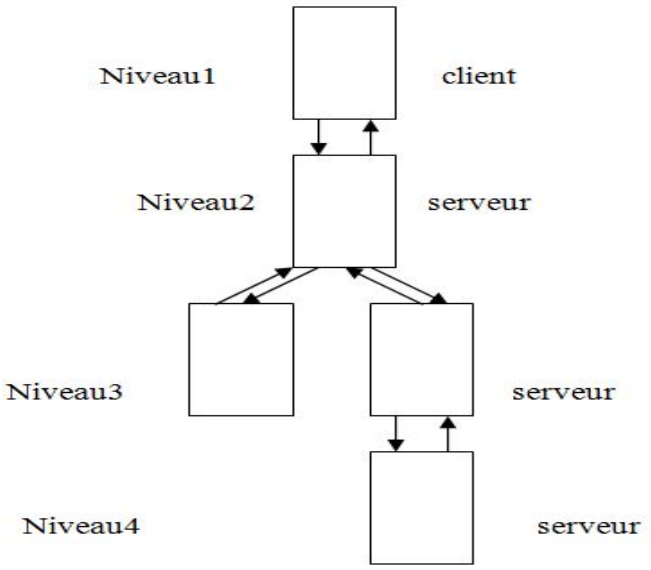

Figure 1.4: Architecture multi niveau.

1.3 TCP/IP

TCP/IP désigne communément une architecture réseau, mais cet acronyme désigne en fait 2 protocoles étroitement liés : un protocole de transport, TCP (Trasmission Control Protocol) qu'on utilise « par-dessus » et un protocole réseau, IP (Internet Protocol). Ce qu'on entend par « modèle TCP/IP », c'est en fait une architecture réseau en 4 couches dans laquelle les protocoles TCP et IP jouent un rôle prédominant, car ils en constituent l'implémentation la plus courante [7].

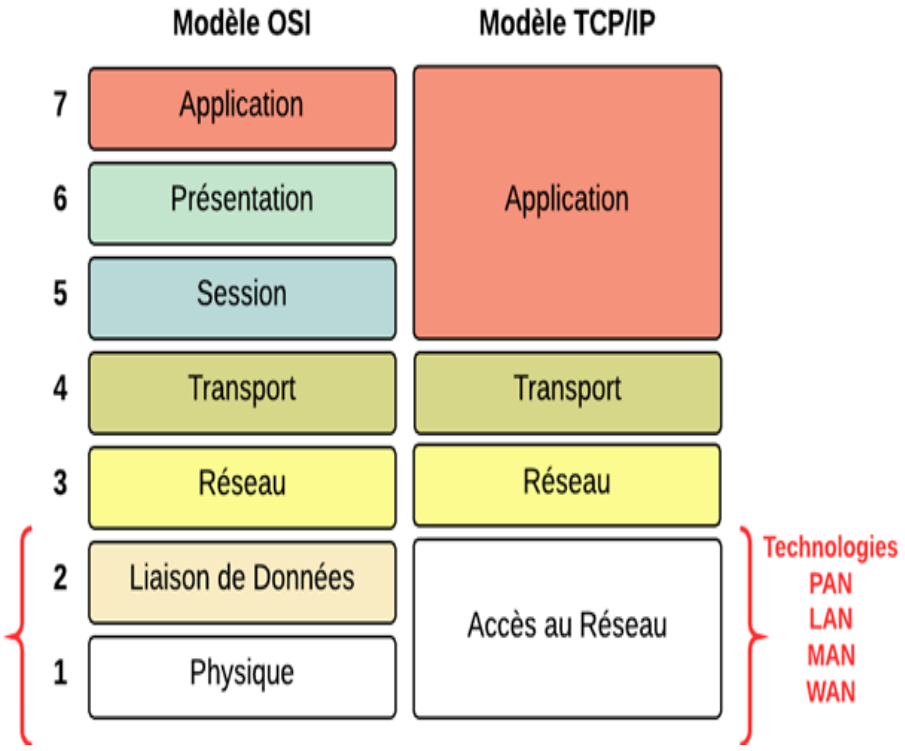

Figure 1.5: Le modèle TCP/IP et le modèle OSI

1.3.1 Les protocoles de Modèle TCP/IP

1.3.1.1 La couche Application

Elle englobe l'application standard du réseau, ses protocoles sont :

HTTP (Hyper Text Transfert Protocol) : Ce protocole est utilisé pour la navigation web entre un serveur HTTP et un butineur. Le protocole assure (normalement) qu'un client comme Internet Explorer ou Netscape Communicator peut envoyer des requêtes et recevoir les réponses de serveurs HTTP comme APACHE ou Internet Information Server sans problèmes particuliers [8].

FTP (File Transfert Protocol): Protocole qui permet d'assurer le transfert de fichiers de façon indépendante des spécificités de NOS (Network Operating System, pour mémoire). Ainsi, un client FTP sous Windows peut télécharger un fichier depuis un serveur UNIX [8].

1.3.1.2 La couche Transport

Elle assure l'acheminement des données, ses protocoles sont:

UDP (User Datagram Protocol) : Est un protocole non orienté connexion de la couche transport du modèle TCP/IP. Ce protocole est très simple étant donné

qu'il ne fournit pas de contrôle d'erreurs (il n'est pas orienté connexion...) [7].
TCP (Transport Control Protocol) : Est l'un des principaux protocoles de la couche transport du modèle TCP/IP. Il permet au niveau des applications, de gérer les données en provenance (ou à destination) de la couche inférieure du modèle (c'est-à-dire le protocole IP).TCP est un protocole orienté connexion, c'est-à-dire qu'il permet à deux machines qui communiquent de contrôler l'état de la transmission [7].

1.3.1.3 La couche Internet (réseau)

Elle est chargée de fournir le paquet de données (datagramme), ses protocoles sont :
IP (Internet Protocole) : Fait partie de la couche Internet de la suite de protocoles TCP/IP. C'est l'un des protocoles les plus importants d'Internet car il permet l'élaboration et le transport des datagrammes IP (les paquets de données) en assurant la livraison. En réalité, le protocole IP traite les datagrammes IP indépendamment les uns des autres en définissant leur représentation, leur routage et leur expédition [7].
ICMP (Internet Control Message Protocol): Est l'un des protocoles fondamentaux constituant la suite des protocoles Internet. Il est utilisé pour véhiculer des messages de contrôle et d'erreur, par exemple lorsqu'un service ou un hôte est inaccessible [2].

1.3.1.4 La couche Accès réseaux

Elle spécifie la forme sous laquelle les données doivent être acheminées quel que soit le type de réseau utilisé, parmi ses protocoles :
Ethernet : Le protocole Ethernet permet de faire transiter dans un réseau local un ensemble de données binaires vers une destination. On appelle cet ensemble de données une trame (ou frame en anglais). Ce protocole est largement déployé dans le monde, a été normalisé par l'organisme IEEE et nommé 802.3 [9].

1.4 Système de messagerie électronique

Un système de messagerie électronique est l'ensemble des éléments contribuant à transmettre un courriel (courrier électronique : message transmis via un réseau informatique) de l'émetteur au récepteur.il y a trois éléments fondamentaux pour assurer les échanges de courriers : le Mail Transfert Agent (MTA), le Mail Delivry

Agent(MDA) et le Mail User Agent (MUA).

1.5 Les concepts clés de la messagerie

1.5.1 Courrier électronique

Le courrier électronique, courriel, e-mail/email ou parfois mail, est un service de transmission de messages envoyés électroniquement via un réseau informatique (principalement l'Internet) dans la boîte aux lettres électronique d'un destinataire choisi par l'émetteur.
Il existe deux moyens pour échanger des courriers électroniques :

- Utiliser un logiciel de messagerie électronique installé sur son ordinateur (Outlook, Thunderbird, Live, ...) avec une adresse de son fournisseur d'accès à Internet par exemple.
- Utiliser un webmail : la consultation des courriels se fait en ligne à partir d'un navigateur web. Les messages sont donc accessibles partout dans le monde [2].

1.5.2 MIME (Multipurpose Internet Mail Extensions)

MIME (Multipurpose Internet Mail Extensions) est un standard qui a été proposé afin d'étendre les possibilités limitées du courrier électronique (mail) et notamment de permettre d'insérer des documents (images, sons, texte, ...) dans un courrier [7].

1.5.2.1 Les fonctionnalitées de MIME

MIME propose de décrire, grâce à des en-têtes, le type de contenu du message et le codage utilisé. MIME apporte à la messagerie les fonctionnalités suivantes :

- Possibilité d'avoir plusieurs objets (pièces jointes) dans un même message.
- Une longueur de message illimitée.
- L'utilisation de jeux de caractères (alphabets) autres que le code ASCII.
- L'utilisation de texte enrichi (mise en forme des messages, polices de caractères, couleurs, etc.) .
- Des pièces jointes binaires (exécutables, images, fichiers audio ou vidéo, etc.), comportant éventuellement plusieurs parties [7].

1.5.2.2 Les formats standards d'un message

MIME utilise des directives d'entête spécifiques pour décrire le format utilisé dans le corps d'un message, afin de permettre au client de messagerie de pouvoir l'interpréter correctement :

- **MIME-Version:** Il s'agit de version du standard MIME utilisée dans le message.
- **Content-type :** Décrit le type et les sous-type des données. Il peut posséder un paramètre « charset », séparé par un point-virtule, définissant le jeu de caractères utilisé.
- **Content-Transfer-Encoding :** Définit l'encodage utilisé dans le corps du message.
- **Content-ID :** Représente un identificateur unique de partie de message.
- **Content-Description :** Donne des informations complémentaires sur le contenu du message.
- **Content-Disposition :** Définit les paramètres de la pièce jointe, notamment le nom associé au fichier grâce à l'attribut filename [7].

1.5.3 Serveur de messagerie

Un serveur de messagerie électronique est un logiciel serveur de courrier électronique. Il a pour vocation de transférer les messages électroniques d'un serveur à un autre. Un utilisateur n'est jamais en contact direct avec ce serveur mais utilise soit un client de messagerie, soit un courrielleur web, qui se charge de contacter le serveur pour envoyer ou recevoir les messages [2].
Des exemples sur les serveurs de messagerie:

1. **Sendmail:** Est un serveur de messagerie dont le code source est ouvert. Il se charge de la livraison des messages électroniques et possède les atouts suivants :[10]
   Avantages:
   - Sendmail est très puissant et résiste beaucoup à la grande charge.
   - Une très bonne sécurité.
   - Code source libre.

- Multi plate-forme de type UNIX (MAC OS, GNU/LINUX).

Inconvénients :

- Sendmail est difficile à configurer car son architecture est vieille.
- Lent.
- Très complexe avec une maintenance difficile.

2. **Postfix:** Il a les caractéristiques suivantes :[10]
   Avantages :

   - Il est adapté aux gros besoins.
   - Facile à installer et à configurer.
   - Maintenance aisée.
   - Sécurisé avec anti-Spam.
   - Codes sources libres.
   - Multi plate-forme de type UNIX (MAC OS, GNU/LINUX).
   - Gratuit.
   - Accessible en mobilité.

   Inconvénients :
   Pas d'inconvénients majeurs.

3. **MS Exchange:** Microsoft Exchange est un logiciel de messagerie qui permet de gérer les Mails, les Calendriers et les Contacts. Il a les caractéristiques suivantes :[10]
   Avantage :

   - Accès en mobilité, c'est-à-dire vous avez accès à vos mails, vos calendriers ou vos contacts via votre ordinateur portable, votre téléphone portable et ce depuis tout endroit connecté à Internet.
   - Une bonne sécurité anti-spam qui sauvegarde et archive vos données critiques.
   - Une bonne efficacité permettant de partager votre agenda et contact professionnel avec vos collègues ou collaborateurs.

   Inconvénients :

   - Les codes sources ne sont pas libres.
   - Uni plate-forme (MS Windows).

1.5.4 Client de messagerie:

Un client de messagerie est un logiciel qui sert à lire et envoyer des courriers électroniques. Ce sont en général des clients lourds mais il existe aussi des applications Web (les webmails) qui offrent les mêmes fonctionnalités.

1.5.4.1 Les clients lourds

Sont des logiciels qui permettent de lire, d'écrire et d'expédier des courriers électroniques. Ils s'installent sur des postes clients qui se connectent au serveur de messagerie.
Les clients lourds ont l'avantage de récupérer nos messages et de les copier sur nos postes locaux, en mode connecté au serveur. Ainsi en mode hors connexion, nous avons accès à nos messages [11].

1. **Thunderbird de Mozilla :** IL se caractérise par:[11]
    - Très léger.
    - Multi plate-forme (Windows, Mac OS, Linux).
    - Rapide.
    - Extensible (peut recevoir de nouvelles fonctionnalités).
    - Les codes sources sont libres d'accès.
    - Installation et configuration simples.
    - Transfert de messages avec pièces jointes.
2. **Zimbra Desktop:** [2]
    - Multi plateforme (Windows, Mac OS, Linux).
    - Codes sources libres.
    - Regroupe tous les comptes dans un seul répertoire.
    - Installation et configuration rapide et facile.
    - Transfert des messages avec des pièces jointes.
    - Extensible.

1.5.4.2 Les clients légers

Des clients de messagerie de type léger sont des logiciels qui sont installés sur des postes clients, permettent de se connecter au serveur de messagerie via un navigateur web. Ils fonctionnent uniquement en mode connecté et ne copie pas en local les messages stockés sur le serveur. Ainsi, en mode hors connexion nous n'avons plus accès à nos courriers [11].

1. **Outlook Web Access:** Outlook Web App est basée sur le Web du client de messagerie de Microsoft Exchange Server 2010. Anciennement connu sous Outlook Web Access dans les itérations précédentes de Exchange Server, Outlook Web App permet aux utilisateurs une expérience similaire à Microsoft Outlook sans nécessiter la présence du client de messagerie complète. Il se caractérise par :
   - Installation rapide et facile.
   - Très léger.
   - Codes sources non libres. [10]
2. **Web Mail Ajax de Zimbra:** Zimbra Collaboration Suite (ZCS) est une suite de logiciels de collaboration, qui comprend un serveur de messagerie et un client Web, actuellement détenue et développée par Zimbra, il se caractérise par :[2]
   - Multi plate-forme (MS Windows, Mac OS, Linux, etc.).
   - Elle est Gratuite.
   - Codes sources libres.
   - Transfère les messages avec pièces jointes.
   - Permet les messages instantanés.

1.5.5 Les agents de la messagerie

1.5.5.1 Le Mail Transfert Agent (MTA)

Le MTA est un programme qui permet d'envoyer le message d'un serveur à un autre. Ce logiciel est situé sur chaque serveur de messagerie. Il est composé d'un agent de routage et d'un agent de transmission. Il envoie le message via un protocole sortant. Notons que les protocoles sortants permettent de gérer la transmission du courrier entre les systèmes de messagerie [12].

1.5.5.2 Le Mail User Agent (MUA)

Sert pour l'expéditeur et le destinataire, est un logiciel client pour le MTA. Il formate les messages en partance afin de les donner au MTA, est un client pour le MDA il formate les messages de la boîte aux lettres afin de les afficher à l'écran [12].

1.5.5.3 Le Mail Delivery Agent (MDA)

Le MDA est un agent qui est en charge de la gestion des boites aux lettres. Il prélève le courrier dans les files d'attentes du MTA et le dépose dans le répertoire de boites aux lettres de l'utilisateur. Pour cela il est souvent considéré comme le point final d'un système de messagerie. Il est possible de placer des fonctions de sécurité à ce niveau : appels antivirus et ou anti-spam [12].

1.5.6 Les protocoles de la messagerie

1.5.6.1 SMTP (Simple Mail Transfert Protocol)

Est le protocole standard permettant de transférer le courrier d'un serveur à un autre en connexion point à point.
Il s'agit d'un protocole fonctionnant en mode connecté, encapsulé dans une trame TCP/IP. Le courrier est remis directement au serveur de courrier du destinataire. Le protocole SMTP fonctionne grâce à des commandes textuelles envoyées au serveur SMTP (par défaut sur le port25). Chacune des commandes envoyées par le client est suivi d'une réponse du serveur SMTP composée d'un numéro et d'un message descriptif [7].

1.5.6.2 POP (Poste Office Protocol)

Permet comme son nom l'indique d'aller récupérer son courrier sur un serveur distant (le serveur POP). Il est nécessaire pour les personnes n'étant pas connectées en permanence à Internet afin de pouvoir consulter les mails reçus hors connexion.
Il existe deux principales versions de ce protocole, POP2 et POP3, auxquels sont affectés respectivement les ports 109 et 110 et fonctionnant à l'aide de commandes textuelles radicalement différentes tout comme dans le cas du protocole SMTP [7].

1.5.6.3 IMAP (Internet Mail Access Protocol)

Le protocole IMAP (Internet Message Access Protocol) est un protocole alternatif au protocole POP3 mais offrant beaucoup plus de possibilités :[7]

- Permet de gérer plusieurs accès simultanés.
- Permet de gérer plusieurs boîtes aux lettres.
- Permet de trier le courrier selon plus de critères.

1.6 Le principe de fonctionnement d'une messagerie électronique

Ce schéma présente le transfert d'un courriel d'un expéditeur à un destinataire [13].

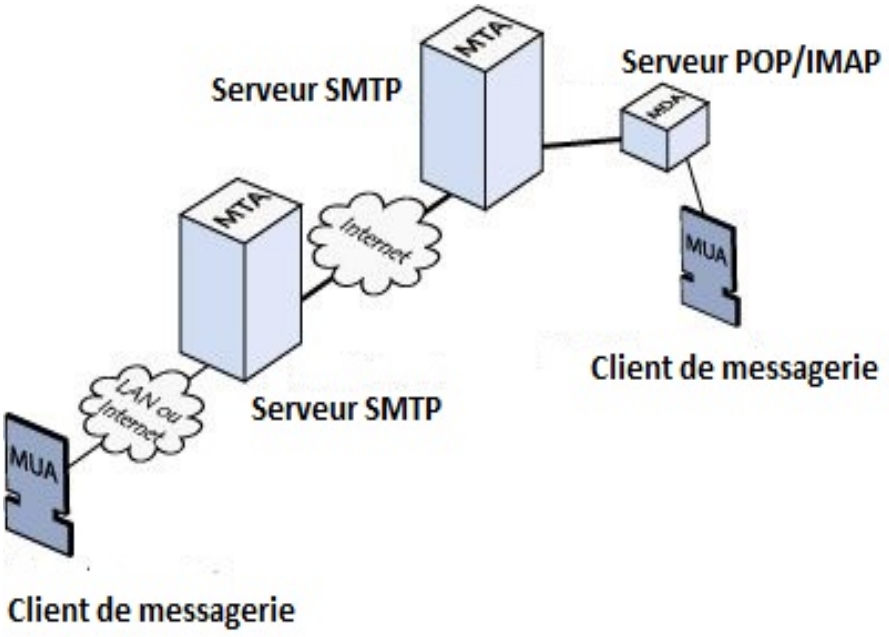

Figure 1.6: Le fonctionnement de la messagerie électronique.

1. L'expéditeur communique son courriel via le MUA.
2. Le MUA transmet ce courriel au MTA (la plupart des MUA intègre des clients SMTP).
3. Le MTA du système de l'émetteur établit un canal de transmission avec le MTA du système du destinataire, par émissions successives de requêtes bidirectionnelles.

4. Une fois le canal établit, le courriel est transmis d'un système à un autre par les MTA.

5. Dans le système du destinataire, Le MTA transmet le courrier reçu au serveur IMAP ou POP3.

6. Le MDA récupère le courriel du serveur IMAP / POP 3, et le met à disposition du MUA.

7. Le MUA dépose le courriel dans les boîtes aux lettres du destinataire qui pourra le consulter à tout moment, sur authentification.

1.7 La sécurité de la messagerie

1.7.1 Menaces et risques

Comme tout système informatique, la messagerie se trouve face à des risques et menaces qui touchent à l'intégrité et la confidentialité des données et tout autre risque, parmi ces risques [14]:

1.7.1.1 Les atteintes aux flux identifiés par l'entreprise

- **Perte d'un e-mail :** Soit au cours de sa transmission ou bien à l'arrivé.
- **Perte de confidentialité :** Se fait par une divulgation accidentelle ou par négligence provoqué par l'émetteur, en envoyant des données et des fichiers sans s'assurer de l'identité des destinataires, ou par un espionnage de message lors de la transmission.
- **Perte d'intégrité :** Un message peut être altéré, accidentellement ou par malveillance pendant sa transmission ou son stockage.
- **Usurpation de l'identité de l'émetteur :** Un utilisateur peut prendre l'identité d'un autre en lui volant son mot de passe et son nom d'utilisateur par exemple.

1.7.1.2 Les atteintes à l'infrastructure et au système d'information

- **Programme malveillants :** La messagerie permet d'introduire des fichiers dans un ordinateur qui peuvent être malveillants comme l'introduction d'un virus par le biais d'une pièce jointe ou bien par un code malicieux dans le

corps même du message et aussi les faux virus (hoax) qui propagent de fausse informations.

- **Spam :** Elle consiste à inonder les boites aux lettres de courriers indésirables et non sollicités, il est aussi utilisé pour diffuser les faux virus.

1.7.2 Solution de sécurité

1.7.2.1 Sécurisation des flux légitimes

- **Chiffrement et signature électronique des messages :** La cryptographie permet d'apporter des réponses efficaces aux problématiques de sécurisation des flux légitimes. Elle permet d'assurer la confidentialité, l'intégrité des messages et l'authentification de l'émetteur.
- **La sécurisation des protocoles :** Permet de sécuriser les communications entre les MTA et entre le serveur et le client de messagerie, parmi ces protocoles :
  - SSL (Secure Socket Layer) qui est développé pour permettre de la communication sécurisée en mode client/serveur pour des applications réseaux utilisant TCP/IP.
  - Le protocole TLS (Transport Layer Security) est une évolution de SSL réalisé par l'IETF et qui sert de base à HTTPS par exemple. [14]

1.7.2.2 Sécurisation des infrastructures

Le filtrage et analyse de contenu se fait par la protection contre les virus et les spam.

- **Protection contre les virus :** Pour qu'elle soit efficace, doit vérifier les points suivants:
  - Le filtrage des e-mails sur les deux niveaux dès leur entrée sur le réseau.
  - Utilisation de plusieurs antivirus.
  - La procédure d'alerte paramétrable.
- **Protection contre les spam :** Les techniques principales sont:
  - L'analyse lexicale et la mise en place d'une liste noire et l'autre blanche.
  - L'utilisation de RBLs (Realtime Blackhole List) pour les adresses ou de relais SMTP.

– L'analyse de signatures.

1.8 Conclusion

La messagerie peut devenir la colonne vertébrale d'une stratégie de communication à l'intérieur de l'entreprise. Durant notre projet, nous allons réaliser un système de messagerie interne pour l'entreprise d'agro-alimentaire IFRI que nous allons présenter dans le chapitre qui suit.

2

Présentation de l'organisme d'accueil

2.1 Introduction

Avant d'entamer notre étude, il convient de commencer par la présentation de l'entreprise et définir son architecture réseau.

2.2 Historique

La SARL IBRAHIM et fils « IFRI» est une société à caractère industriel, évoluant dans le secteur agro-alimentaire. Á l'origine, c'était la « Limonade-rie IBRAHIM » qui existait en 1986, celle-là a été créé sur les fond propres de Mr IBRAHIM LAID, son gérant, dix (10) ans plus tard c'est-à-dire en 1996, elle fut transformée en SNC (société à nom collectif), puis elle s'est fait un statut de SARL (société à responsabilités limitées) composée de plusieurs associés.

La SARL IBRAHIM et fils «IFRI », à caractère familiale, inaugure son premier atelier d'embouteillage d'eau minérale. Elle fût la première entreprise privée dans le secteur des eaux minérales. A cette date, plus de 7.5 millions de litres d'eau minérale sont commercialisés à l'échelle nationale. La production franchira le cap des 504 millions de L (litres) dans toute la gamme des produits ifri en 2011. Avec près de 30% de parts de marché des eaux embouteillées, cette marque est leader dans les eaux minérales.

2.3 Situation géographique

Le complexe de production d'eau minérale naturelle de la SARL IBRAHIM et FILS-ifri est situé dans la commune d'Ighzer- Amokrane- Daira d'Ifri Ouzellague-Wilaya de Bejaia. Il est localisé au sud-ouest de l'agglomération d'Ighzer Amokrane, soit à 400 m au sud de la RN.26

2.4 Identification de la Sarl

FRI est l'une des entreprises du « groupe IFRI », en cours de constitution qui est composé de quatre (04) Sarl :

- La Sarl IBRAHIM et Fils- ifri, spécialisée dans la production d'eau minérale et de boissons diverses.
- La Sarl Bejaia Logistique, assurant le transport de marchandises.

- La Sarl Huileries Ouzellaguen, spécialisée dans le raffinage et le conditionnement des huiles d'origine végétale.
- La Sarl General Plast, Spécialisé dans la production de préforme et bouchons.

2.5 Structure de la SARL IBRAHIMI & FILS

Elle est composée des structures suivantes :

- Direction générale.
- Contrôle de gestion.
- Commerciale & Marketing.
- Approvisionnement.
- Ressources Humaines.
- Industrielle.
- Finance & Comptabilité.
- Logistique.
- Technique (Maintenance).
- Moyens généraux.
- Planification & Ordonnancement.
- Exportation.
- Juridique.
- Service Informatique.

2.6 Service Informatique

La présentation de l'informatique dans l'ensemble des services et son utilisation par près de 90% du personnel donne au service informatique un rôle de plus en plus important dans le bon fonctionnement de la SARL « IBRAHIMI & Fils ».
Il est composé de sinq (05) éléments, un responsable informatique, un ingénieur applicatif et trois (03) informaticiens. Ses missions sont multiples et peuvent être regroupé de la façon suivante :

- Administration des systèmes et bases de données.
- Développement et réalisation des projets informatiques.
- Gestion du parc de micro-ordinateurs.
- Tâches d'exploitation (Traitement des facturations ...).
- Assistance aux utilisateurs et Formation du personnel aux techniques informatiques.
- Maintenance du système informatique.
- L'étude des nouveaux projets.
- Introduction de nouvelles technologies.
- Sauvegarde et archive des données de l'entreprise.

2.7 L'architecture réseau d'IFRI

IFRI se compose de 4 sites principaux : Zone 1 IFRI, Bejaïa logistique, zone III, Général Plast. La zone 1 et Bejaïa logistique sont situé à Ighezar Amokrane et zone 3 et Général Plast sont situé à Taharacht. Les deux parties sont reliées par un réseau virtuel privé (VPN) entre les deux parfeu. Le routage se fait au niveau des parfeu, l'accès internet au niveau des routeurs et leur serveur de messagerie se situe à l'extérieur.

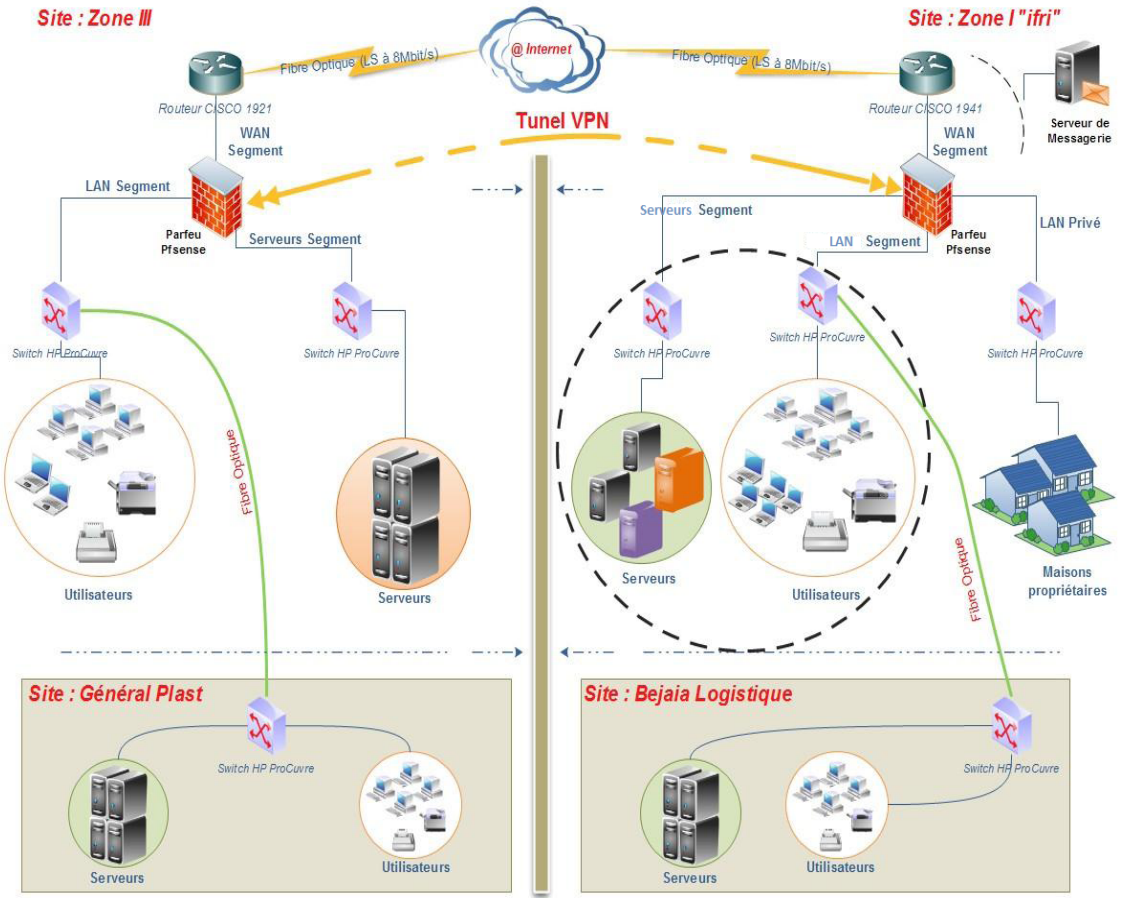

Figure 2.1: Schéma réseau du groupe IFRI

2.8 Ressources matériels

L'entreprise IFRI dispose d'un parc informatique très vaste, on peut toujours tenter de donner un résumé du matériel se trouvant au siège (site centrale) sans compter les autres sites, le tableau suivant est un résumé :

Description de l'Equipment	Quantité	Observation
Serveur IBM	04	Réf : X3550
Serveur HP ProLiant	01	Réf : DL 380
Serveur Dell	01	Réf :
Routeur CISCO Série 1900	01	Modèle : 1941
Micro-ordinateur de Bureau	146	Dell & HP Compaq
Micro-ordinateur Portable	45	Différents Marques
Imprimantes	101	Réseaux et USB
Photocopieur	05	Réseaux en grand model
Switch	41	HP Procuvre & Autres
Point d'accès	12	DLINK, SYSLINK, CISCO ...
Terminaux	10	Scan Code à bar
Standard téléphonique ALCATEL Lucent	01	Alcatel

Table 2.1: Résumé du matériel informatique sur le site central IFRI

2.9 Ressources logicielles

Comme ressources logiciels on peut citer :

- Les systems exploitations : (Windows 2008 server R2, Windows 2003 server R2, windows XP professionnel sp3 et windows 7 professionnel sp1).
- Antivirus Kaspersky Lab version entreprise.
- Le pack Ms Office 2007 et 2010.

2.10 Conclusion

Dans ce chapitre, nous avons présenté notre organisme d'accueil « IFRI » et son architecture réseau ainsi les ressources matériels et logiciels.

3

Analyse et conception

3.1 Introduction

Après avoir présenté l'organisme d'accueil, nous allons présenter le cahier de charge et les différents besoins de notre système, l'architecture système et l'architecture fonctionnelle afin de garantir une bonne conception du futur système.

3.2 Méthode d'analyse et de conception Grady BOOCH (OOAD)

La méthode de Booch est une méthode d'analyse et de conception de développement de logiciel employée pour analyser, modéliser et documenter des conditions de système. Elle a été développée par Grady Booch, basé sur plus de quinze ans d'une expérience pratique de développement avec des application de grandes complexité [15].

La méthodologie de Booch a sa force principale dans la conception du système d'objet. Grady Booch a inclus dans sa méthodologie une analyse qui est orienté objet (OOA) [15].

3.2.1 L'analyse orientée objet(OOA)

L'analyse orientée objet (OOA) est la procédure d'identification des exigences en matière de génie logiciel et l'élaboration de spécifications logicielles en termes de modèle d'objet d'un système logiciel, qui comprend l'interaction des objets [16].

La principale différence entre l'analyse orientée objet et d'autres formes d'analyse est que dans l'approche orientée objet, les exigences sont organisées autour des objets, qui intègrent les données et les fonctions. Ils sont modélisés d'après les objets du monde réel que le système interagit avec. Dans les méthodes d'analyse traditionnelles, les deux aspects, les fonctions et les données sont considérés séparément [16].

Grady Booch a défini OOA comme une méthode d'analyse et de conception qui examine les exigences dans la perspective des classes et des objets trouvés dans le vocabulaire du domaine du problème [16].
Les tâches principales de l'analyse orientée objet (OOA) sont:

- Identification des objets.
- L'organisation des objets en créant le diagramme de modèle d'objet.

- Définir les internes des objets ou des attributs d'objets.
- Définir le comportement des objets, à savoir, objet actions.
- Décrire la façon dont les objets interagissent [16].

3.2.2 La conception orientée objet OOD(Object-Oriented Design)

Object-Oriented Design (OOD) implique la mise en oeuvre du modèle conceptuel produite lors de l'analyse orientée objet. En OOD, les concepts dans le modèle d'analyse, qui sont indépendants de la technologie, sont mappés sur la mise en oeuvre des classes, les contraintes sont identifiées et les interfaces sont conçues, ce qui entraîne un modèle pour le domaine de la solution, à savoir, une description détaillée de la façon dont le système doit être construit sur les technologies concrètes [16].

Grady Booch a défini la conception orientée objet comme une méthode de conception englobant le processus de décomposition orientée objet et une notation pour représenter des modèles à la fois logiques et physiques ainsi que statiques et dynamiques du système en cours de conception [16].
Les détails de mise en oeuvre comprennent généralement:

- Restructuration données de la classe (si nécessaire).
- La mise en oeuvre de méthodes, à savoir, les structures de données internes et des algorithmes.
- La mise en oeuvre du contrôle.
- Mise en oeuvre des associations [16].

3.2.3 La programmation orientée objet (POO)

La programmation orientée objet (POO) est un paradigme de programmation basé sur des objets (ayant à la fois des données et méthodes) qui vise à intégrer les avantages de la modularité et la réutilisabilité. Les objets qui sont généralement des instances de classes, sont utilisées pour interagir les uns avec les autres pour concevoir des applications et des programmes informatiques [16].

Grady Booch a défini la programmation orientée objet comme «une méthode de mise en oeuvre dans laquelle les programmes sont organisés sous forme de collections coopératives d'objets, dont chacune représente une instance d'une classe, et dont les

classes sont tous membres d'une hiérarchie de classes unis par des relations d'héritage » [16].
Les caractéristiques importantes de la programmation orientée objet sont:

- Approche Bottom-up dans la conception du programme.
- Programmes organisés autour des objets, regroupés dans des classes.
- Focus sur les données avec des méthodes pour faire fonctionner sur les données de l'objet.
- Interaction entre les objets grâce à des fonctions.
- Réutilisation de la conception à la création de nouvelles classes en ajoutant des fonctionnalités à des classes existantes [16].

3.2.4 Quelque concepts de l'orienté objet selon Grady booch

3.2.4.1 Un objet

Un objet a un état, un comportement et une identité.

- L'état regroupe les valeurs instantanées de tous les attributs d'un objet et évolue au cours du temps. A un instant donné, l'état d'un objet est la conséquence de ces comportements.
- Le comportement décrit les actions et les réactions d'un objet, il regroupe toutes les compétences d'un objet et il se représente sous la forme d'opérations.
- Tout objet possède une identité qui lui est propre et qui le caractérise, elle permet de distinguer tout objet de façon non ambigüe par rapport aux autres objets de la même classe, indépendamment de l'état [17].

3.2.4.2 Une classe

Une classe est un ensemble d'objets qui partagent une structure commune, un comportement commun. Un seul objet est simplement une instance d'une classe [15].

3.2.4.3 Une abstraction

Une abstraction désigne les caractéristiques essentielles d'un objet qui le distinguent de tous les autres types d'objets et de fournir ainsi une définition conceptuelle limitées, par rapport au point de vue du concepteur [15].

3.2.4.4 Une Encapsulation

Encapsulation est le processus de compartimenter les éléments d'une abstraction qui constituent sa structure et le comportement; l'encapsulation sert à séparer l'interface d'une abstraction et son implémentation [15].

3.2.4.5 Une modularité/Modularisation

Modularisation consiste à diviser un programme en modules qui peuvent être compilés séparément, mais qui ont des connexions avec d'autres modules. Modularité est la propriété d'un système qui a été décomposé en un ensemble de modules cohérents et faiblement couplés [15].

3.2.4.6 Une hiérarchie

La hiérarchie est un ordonnancement des abstractions [15].

3.2.4.7 La persistance

Persistance est la propriété d'un objet à travers lequel son existence transcende le temps (à savoir, l'objet continue d'exister après son créateur cesse d'exister) et / ou l'espace (à savoir, l'emplacement de l'objet se déplace depuis l'espace d'adresses dans laquelle il était créé) [15].

3.2.4.8 Polymorphisme

Le polymorphisme est un concept dans la théorie du type dans lequel des noms peuvent représenter des instances de différentes classes aussi longtemps qu'ils sont liés par certaines superclasses communnes. Tout objet désigné par ce nom est donc en mesure de répondre à un ensemble commun des opérations de différentes manières. Polymorphisme, une opération peut être réalisée différemment par les classes de la hiérarchie [15].

3.2.4.9 Association

Parmi les différents types de rapports de classe, les associations sont le plus général mais aussi les plus sémantiquement faible. L'identification des associations entre les classes est souvent une activité d'analyse et de conception au début, au moment où nous commençons à découvrir les dépendances générales entre nos abstractions. Alors que nous continuons notre conception et la mise en oeuvre, nous allons souvent affiner ces associations faibles en les tournants dans l'un des autres

rapports de classes plus concrètes (composition, agrégation). Degré d'une association désigne le nombre de classes impliquées dans une connexion. Degré peut être unaire, binaire ou ternaire.

- **Une relation unaire** relie les objets de la même classe.
- **Une relation binaire** relie les objets de deux classes.
- **Une relation ternaire** relie les objets de trois ou plusieurs classes [15].

3.2.4.10 L'agrégation

L'agrégation est un type spécialisé d'association. L'agrégation est parfois mieux car il encapsule des parties comme des secrets de l'ensemble. Les liens sont parfois mieux, car ils permettent un couplage lâche entre les objets [15].

3.3 Présentation du projet

La messagerie électronique est l'outil le plus répandu dans l'Internet des entreprises ou des particuliers. Elle tend à prendre une place de plus en plus prépondérante par rapport aux moyens de communication traditionnels.

Dans le cadre de l'entreprise aujourd'hui, la messagerie électronique offre l'opportunité fabuleuse de prendre contact entre utilisateurs, elle permet de réduire le nombre d'appels internes et de rencontres entre direction et personnel. Elle est aussi un excellent moyen de coordination d'une équipe ou d'un service. Comme elle permet aux agents des différents bureaux de se transmettre des informations confidentielles sans passer par un contrôleur ou serveur d'accès à distance. Outre ces possibilités, la messagerie électronique permet de partager des fichiers, c'est la fonction de joindre un fichier au message et d'échanger des documents avec les utilisateurs.

Donc notre projet consiste à réaliser un système de messagerie interne pour l'entreprise d'agro-alimentaire IFRI en utilisant un serveur de messagerie Apache james.

Un système de messagerie fournira aux employés de l'entreprise Ifri la possibilité de pouvoir échanger des messages et des fichiers sans se déplacer et de manière simple et plus rapide. Notre application fournira les fonctionnalités clés suivantes:

- Rédiger et expédier un message et consulter les messages qui lui sont destinés.
- Envoyer un seul message à un groupe entier de personnes.
- Répondre à un message sans avoir à retaper l'entête.

- Transférer un message reçu à un autre distinataire.
- Gérer les utilisateurs par l'administrateur de système.

3.4 Identification des acteurs et des cas d'utilisation

Les acteurs principaux de notre système sont :

- **Administrateur :** Il crée les comptes des utilisateurs ou bien les supprimés.
- **Utilisateur :** Un utilisateur peut envoyer, recevoir et supprimer des messages, consulté sa boite de réception après être authentifié.

3.4.1 Les cas d'utilisation de l'administrateur

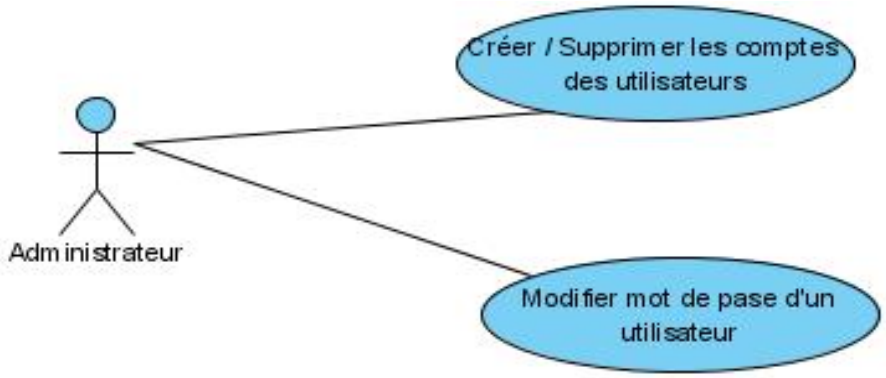

Figure 3.1: Les cas d'utilisation de l'administrateur

Le cas d'utilisation "Créer un compte"
L'administrateur après avoir s'authentifié, il ajoute des utilisateurs en remplissant le formulaire d'ajout.
Le cas d'utilisation "Supprimer un compte"
L'administrateur peut supprimer un compte d'un utilisateur aprés avoir vérifié si le compte existe.
Le même sénario pour le cas d'utilisation "Modifier mot de passe". **Le cas d'utilisation "Modifier un mot de passe"**
L'administrateur peut changer le mot de passe d'un utilisateur donner aprés que se dernier lui demmande.

3.4.2 Les cas d'utilisation de l'utilisateur

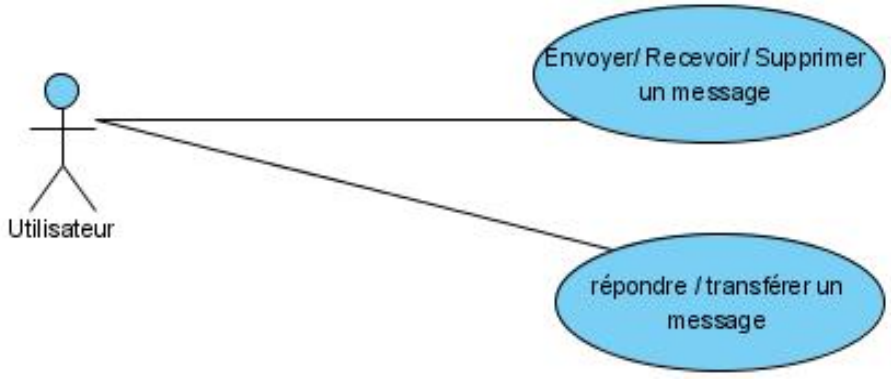

Figure 3.2: Les cas d'utilisation de l'utilisateur

Le cas d'utilisation "Envoyer message"
Ce cas permet d'envoyer un message à un autre utilisateur en remplissant le formulaire de la fenêtre "Nouveau Message" aprés être authentifié.
Le cas d'utilisation "Recevoir message"
L'utilisateur aprés avoir accédé à la boite de réception, il consulte ses messages et à partir de là il peut répondre ou transférer les messages reçu.
Le cas d'utilisation "repondre à un message"
L'utilisateur à la possibilité de repondre à un message donner tout en gardant sont objet et le destinataire apréss l'avoir consulté.
Le cas d'utilisation "transférer un message"
L'utilisateur peut transférer le même message à un autre utilisateur aprés l'avoir consulté en gardant le même objet et le même contenu.

3.5 Conception

Dans ce qui est précédé, nous avons présenté la partie d'analyse de notre système ou on a identifié les besoins fonctionnels ainsi les acteurs de système et les cas d'utilisations de chaque acteurs, on passe maintenant à la phase de conception.

Notre système de messagerie est une application simple, englobant seulement une poignée de classes. En effet, à première vue, le novice orienté objet pourrait être tenté de résoudre ce problème. Pour pouvoir réaliser notre système, on a besoin de définir les différentes classes et objets qui le forme et les relations qui les relient.

3.5.1 Présentaion des différentes classes de notre application

3.5.2 La classe administrateur

Nom de classe : Administrateur.
Rôle : Créer et gérer les comptes des utilisateurs de notre messagerie.
Attributs :

- Login.
- Password.

Opérations :

- Authentification : S'authentifier avant d'accéder à l'espace administrateur.
- Ajouter_Compte : Pour créer les comptes pour les utilisateurs.
- Liste_utilistaeurs : Pour voir la liste des utilisateurs de l'application.
- Supprimer_un_utilisateur : Supprimer un utilisateur du système.
- Modifier_mot_de_passe : Modifier le mot de passe d'un utilisateur.
- Déconnexion : Déconnecter et quitter l'espace administrateur.

3.5.3 La classe utilisateur

Nom de classe : Utilisateur.
Rôle : Utiliser la messagerie en envoyant et recevant des mails.
Attributs :

- Nom.
- Prénom.
- Login.
- Password.

Opérations :

- Récupérer : Récupère et affiche les mails reçus.
- Envoyer_message : Permet d'envoyer les mails.
- Répondre_message : Permet de répondre à un message reçu.

- Transférer_message : Permet de transférer un message vers un autre utilisateur.
- Supprimer_message : Permet de supprimer les messages voulus.
- Ajouter_pj : Permet ajouter une pièce jointe à un message pour l'envoyer.
- Télécharger_pj : Permet de télécharger une pièce jointe et l'enregistrer.
- Deconnexion : Permet de se déconnecter et de quitter l'application.

3.5.4 La classe Message

Nom de classe : Message.
Attributs :

- ID_msg: L'identifiant du message.
- From: L'émetteur de message.
- To: Le ou les destinataires de message.
- Objet: L'objet et le sujet du message.
- Contenu: Le contenu du message.

Opérations :

- Composé_message
- Lire_message.

3.5.5 La classe Pièce jointe

Nom de classe : Pièce jointe.
Rôle : Pour échanger les différents documents entre utilisateurs.
Attributs :

- ID_pj.
- Nom_pj.
- Taille_pj.

Opération:

- Lire_pj.

3.5.6 L'architecture système

Grady Booch offre la possibilité d'implémenter un système à travers une architecture englobant ses différentes classes ainsi que les différentes relations qui les relient.

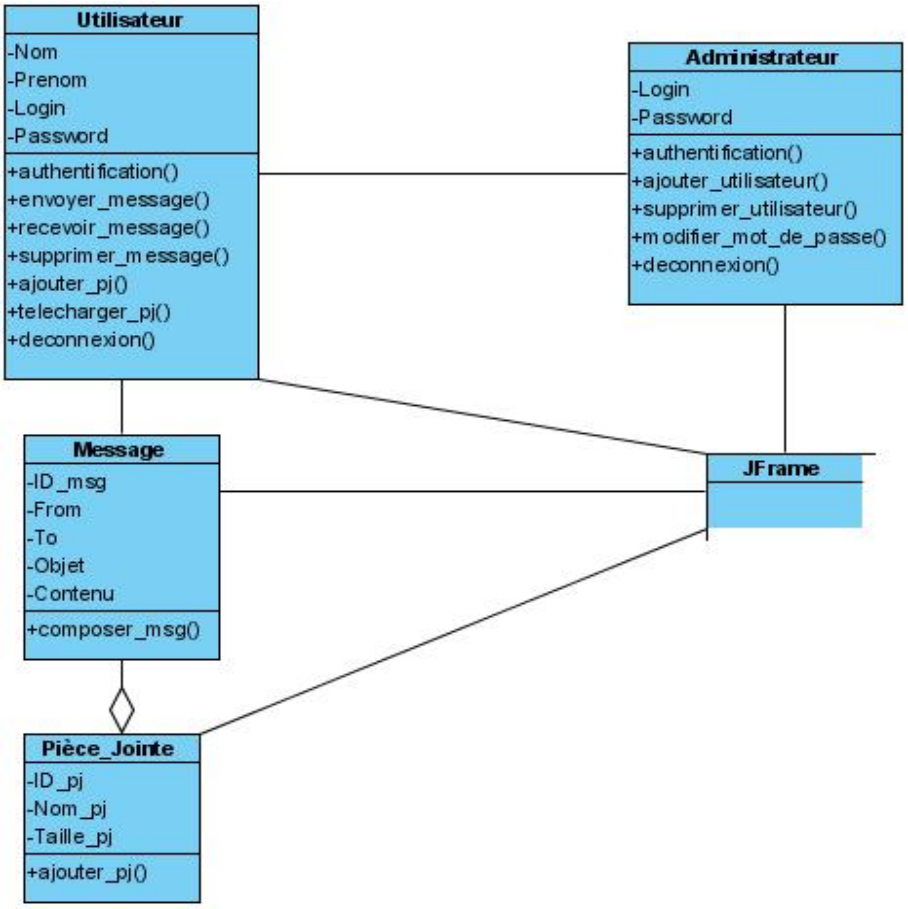

Figure 3.3: L'architecture Système

La classe administrateur gère la classe utilisateur en implémentant ces différentes méthodes, comme elle utilise la classe JFrame pour être vu par l'extérieure.

La classe utilisateur utilise la classe message pour l'envoie et la réception des messages qui peuvent contenir des pièces jointes, elle utilise aussi la classe JFrame pou être vu par l'extérieure.

3.6 L'architecture fonctionnelle

Un système de messagerie étant une spécialisation d'une architecture client / serveur, a minimalement deux noeuds principaux, le serveur et le client de messagerie. Le serveur est un noeud qui a une adresse connue sur un réseau et est

configuré pour écouter les requêtes de client sur un port spécifique. Une application de client de messagerie fait une demande via une interface graphique vue par l'extérieure, à la demande de l'utilisateur.

Dans notre système, le client fait une demande (envoie d'un message, récupérer un message, ...) au serveur de messagerie Apache James qui est configuré pour écouter les requêtes du client sur le port 4555 et lui répond en exécutant ses requêtes.

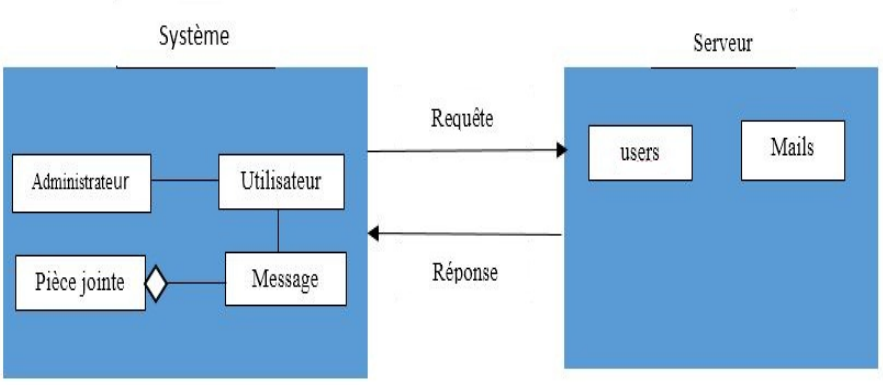

Figure 3.4: L'architecture fonctionnelle

3.7 Conclusion

Pour la conception de notre système, on s'est inspiré de la méthode d'analyse et de conception de Grady Booch qui permet de faciliter l'implémentation de programmes dans des langages de programmation orientée objet, ainsi que de représenter les différentes phases du développement d'un projet analyse et conception.

4

Réalisation

4.1 Introduction

Après avoir conçu notre système, il est temps de réaliser l'application en utilisant des différents outils de développement, java comme langage de programmation en introduisant l'API Javamail pour développer notre système et pour pouvoir faire des test sur le serveur de messagerie, on a opté pour le serveur Apache James server à cause de ces multiple caractéristiques.

4.2 Les outils et environnement de développement

4.2.1 Apache JAMES (Java Apache Mail Enterprise Server)

Le serveur James comme son nom l'indique, est un serveur de messagerie entièrement développé en JAVA. Il intègre un serveur de mail (SMTP, POP3) et un serveur de news (NNTP). James a été développé spécialement pour être une solution Mail complète d'entreprise, portable et personnalisable. C'est un projet de l'Apache Software Foundation il est donc régis par la licence Apache Software License [18].

Ce projet appartient à la communauté OpenSource des projets Apache, il est donc constamment en évolution et a déjà atteint un degré de maturité élevé même si certaines fonctionnalités sont encore expérimentales. La réelle particularité de James par rapport à un autre serveur mail est la possibilité pour un administrateur de changer sa configuration et de la remplacer par la sienne et ce, de manière extrêmement facile. La volonté des auteurs a toujours été dans ce sens et, pour ce faire, ils ont développé leur propre API appelée API Mailet permettant de traiter de manière efficace les différents champs composant un e-mail. Toutes les opérations effectuées sur un message sont basées sur ce que les auteurs appellent «la technologie matcher/mailet» que nous détaillerons par la suite. Celle-ci permet de sélectionner un ensemble de messages arrivant sur le serveur, en fonction des données contenues dans les mails et de leur appliquer une ou plusieurs opérations spécifiées par l'administrateur du serveur. Remarquons que le serveur James a été développé au-dessus des Framework d'Avalon. Signalons également que, même si l'opération est un peu délicate, il est tout à fait envisageable de faire tourner James au dessus d'un autre serveur d'applications [18].

Figure 4.1: Le slogon de apache james

4.2.1.1 Les caractéristiques de Apache james

Parmis les caractéristiques de Apache James, on trouve [18]:

1. **Portabilité:**
   James est portable car il est un Java pur à 100%. Avec James, vous pouvez créer des applications personnalisées de messagerie sur Windows, sous UNIX, sur les téléphones cellulaires, le coupe-herbes, etc.

2. **Complèt et autonome:**
   James est complet car il ne nécessite pas un soutien supplémentaire à partir d'autres serveurs ou applications. Cela rend James autonome, car il peut gérer à la fois le transport du courrier et de stockage comme une application de serveur unique. D'autres serveurs de messagerie peuvent compter sur des applications tierces nécessaires pour obtenir la fonctionnalité dont vous avez besoin. James contient des fonctionnalités d'autres projets Apache, mais tout est emballé comme une application.

3. **Abstraction:** James utilise l'abstraction de haut en bas. Des ressources telles que JDBC (pour spool ou comptes d'utilisateurs stockés dans bases de données) sont extraites comme protocoles. James tire parti des interfaces définies pour chacun de ces ressources pour maintenir la modularité et une bonne conception

4. **Open Source:**
   James est libre, et qui est certainement de bonnes nouvelles pour les organisations et les entreprises avec de petits budgets informatiques. James est pas un serveur e-mail coûteux tels que Microsoft Exchange ou Lotus Domino, mais James rivalise autre serveur e-mail solutions de robustesse et d'évolutivité.

4.2.2 Java

Java est un langage de programmation et une plate-forme informatique qui ont été créés par Sun Microsystems en 1995.Java est sécurisé et fiable. La particularité principale de Java est que les logiciels écrits dans ce langage sont très facilement portables sur plusieurs systèmes d'exploitation tels qu'UNIX, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications [19].

Le langage reprend en grande partie la syntaxe du langage C++, très utilisé par les informaticiens. Néanmoins, Java a été épuré des concepts les plus subtils du C++ et à la fois les plus déroutants, tels que les pointeurs et références, et l'héritage multiple remplacé par l'implémentation des interfaces. Les concepteurs ont privilégié l'approche orientée objet de sorte qu'en Java, tout est objet à l'exception des types primitifs (nombres entiers, nombres à virgule flottante, etc.) [20].

4.2.2.1 Javamail

JavaMail est une extension standard de Java SE. C'est à dire que les spécifications de l'API sont définies, mais elle n'est pas fournie avec la JRE. L'API JavaMail fournit un cadre indépendant de la plateforme et indépendante du protocole pour construire des applications de messagerie. L'API JavaMail fournit un ensemble de classes abstraites définissant des objets qui composent un système de messagerie. C'est un package optionnel (extension standard) pour la lecture, la composition et l'envoi de messages électroniques [21].

4.2.2.2 JavaBeans Activation Framework

JavaBeans Activation Framework est une extension standard, les développeurs qui utilisent la technologie Java peuvent profiter des services standard pour déterminer le type d'une pièce arbitraire de données, encapsuler l'accès, découvrir les opérations disponibles sur elle [22].

Le cadre Activation JavaBeans est implémenté comme une extension standard. Sun fournit une implémentation du logiciel JAF référence libre de droits, sous forme binaire, que les développeurs peuvent utiliser pour développer JAF applications technologiques compatibles pour toute plateforme qui prend en charge la version 1.4 ou ultérieure de Java 2 Standard Edition [22].

4.2.3 Eclipse

L'Éclipse IDE (Intergrated Development Environment) est un environnement de développement libre permettant de créer des programmes dans de nombreux

langages de programmation tels que Java, C++, PHP, etc. Il possède plusieurs caractéristiques dont:

- Les systèmes basés sur Éclipse sont enrichis grâce au nombre impressionnant de plugins qu'il contient.
- Portabilité : Éclipse fournit un support à un grand nombre hétérogène d'environnement clients, du PC traditionnels à des supports plus légers tels que les tablettes, ou des plate-formes mobiles comme les smartphones et les PDA. A partir du moment où une JVM peut être installée.

4.3 Installation et configuration de Apache james

L'installation de Apache james est trés simple, une fois que Apache james est téléchargé, il suffit de décompresser l'archive dans votre répertoire d'installation James.

Après avoir installé le binaire, l'étape suivante consiste à ajuster la configuration initiale. Le serveur doit être arrêté, puis la configuration peut se poursuivre. La configuration la plus essentielle est définie dans le fichier config.xml. Il y a quelques questions qui devraient être traitées immédiatement après l'installation:

- **Compte administrateur RemoteManager :**Avant que le service RemoteManager peut être utilisé pour ajouter des utilisateurs à cette installation de serveur, un compte administrateur doit être créé.
- **Serveurs DNS:** James a besoin d'avoir accès à un serveur DNS pour la résolution de domaine. Lors de la configuration de la boîte suppose qu'il existe un serveur DNS sur localhost. Les administrateurs devront changer la configuration pour pointer vers un serveur DNS valide. Cela peut être fait en ajustant le bloc de configuration dnsserver dans le fichier config.XML.
- **Managed Domain Names / adresses IP :**James ne gère que le courrier envoyé à des destinataires qui se trouve en localhost. Pour permettre à James de gérer les email à des serveurs SMTP distant, il suffit d'ajouter le nom de domaine approprié ou l'adresse IP à la section ServerNames du fichier config.xml.

Une fois que vous avez modifié le fichier de configuration, vous devrez redémarrer James pour que les modifications prennent effet. Lorsque James commence, une liste des services James et les ports sur lesquels ils sont à l'écoute doit être affiché sur

la console. Des informations supplémentaires sur la configuration du système est imprimé dans les fichiers journaux James lors du démarrage. Enfin, une fois la configuration terminée, il sera nécessaire de créer des comptes d'utilisateurs avant que le serveur James sera pleinement opérationnel. Maintenant, reste à lancer James Server. Comme celui-ci va ouvrir des sockets serveur sur des ports réservés (1024), il faut être administrateur pour effectuer cette opération:

Figure 4.2: Lancemebt de serveur Apache james

Après avoir lancer notre serveur maintenant on peut accéder à l'interface qui gère les utilisateurs (RemoteManger) avec la connexion distante « telnet ».

Figure 4.3: La connexion au serveur

Pour pouvoir afficher la liste des fonctions qui peuvent être faite par l'administrateur en doit d'abord s'authentifier en entrant le login et le mot de passe puis la commande « help ».

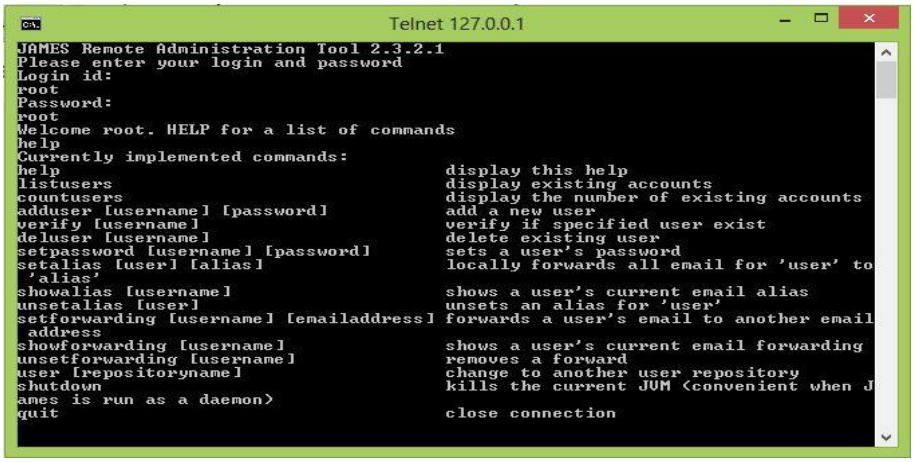

Figure 4.4: Les commandes du serveur Apache James

4.4 Présentation de l'application

4.4.1 La fenêtre graphique d'authentication de l'administrateur

Cette fenêtre nous permet d'authentifier l'administrateur pour accéder à son espace.

Figure 4.5: La fenêtre graphique d'authentication de l'administrateur

4.4.2 La fenêtre graphique de l'ajout d'un utilisateur

Cette fenêtre nous permet d'ajouter un utilisateur.

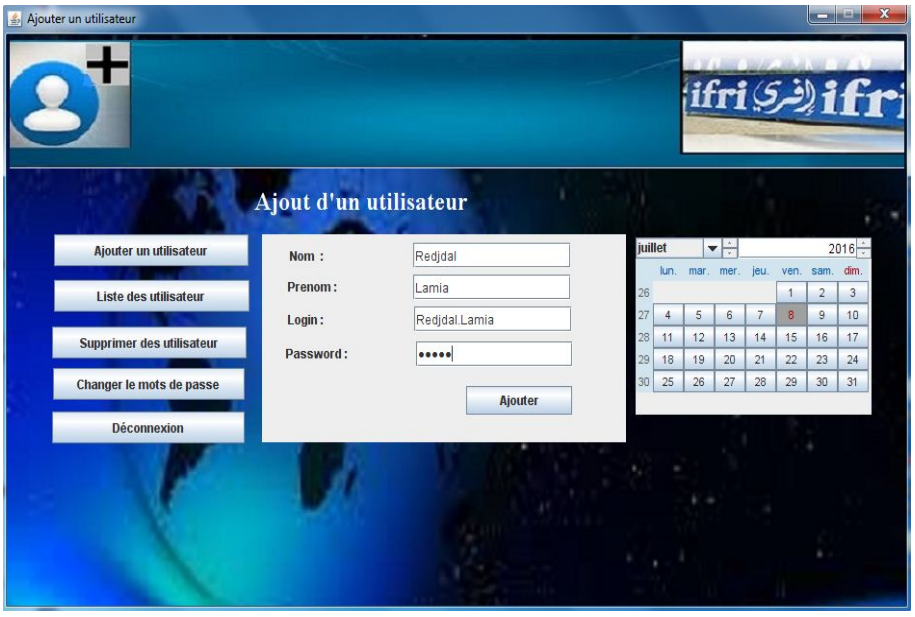

Figure 4.6: La fenêtre graphique d'ajout d'un utilisateur

4.4.3 La fenêtre graphique de la liste des utilisateurs

Cette fenêtre nous permet d'afficher la liste des utilisateurs qui se trouve dans le serveur.

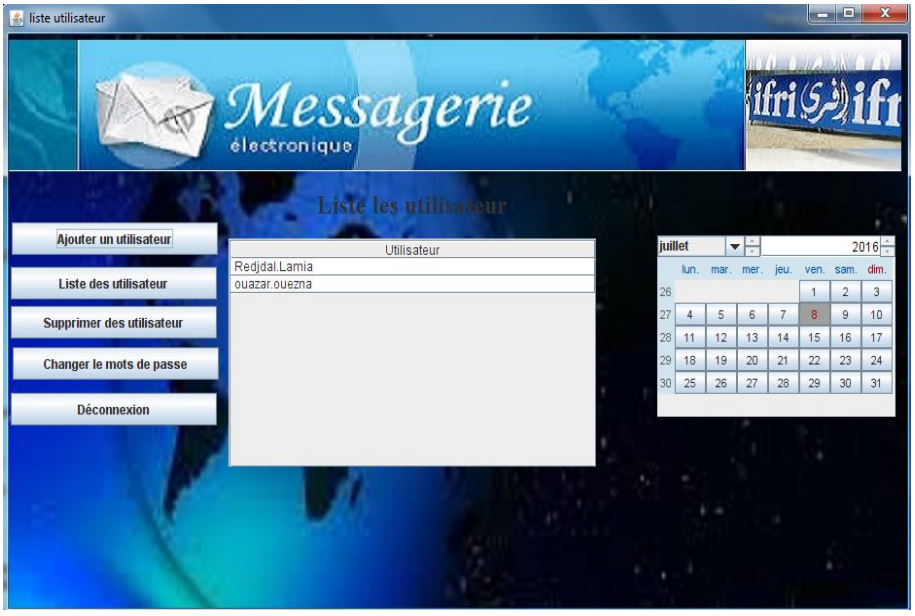

Figure 4.7: La fenêtre graphique de la liste des utilisateurs

4.4.4 La fenêtre graphique d'authentication de l'utilisateur

Cette fenêtre nous permet de s'authentifier pour accéder à son espace utilisateur.

Figure 4.8: La fenêtre graphique d'authentication de l'utilisateur

4.4.5 La fenêtre graphique d'envoie d'un mesage

Cette fenêtre nous permet de composé un nouveau message et puis l'envoyer.

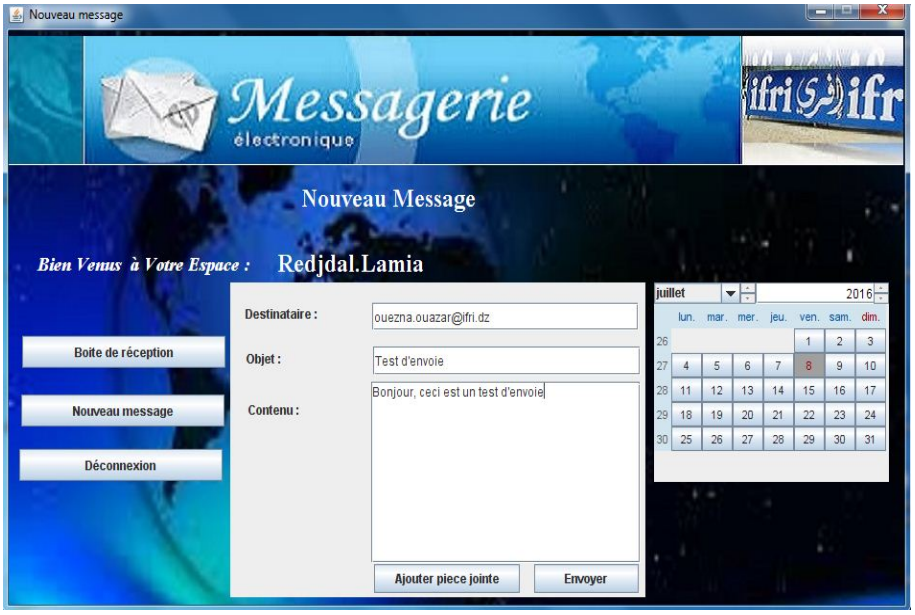

Figure 4.9: La fenêtre graphique d'envoie d'un message

4.4.6 La fenêtre graphique de la boite de réception

Cette fenêtre nous permet de récupérer tous les messages reçus par un utilisateur.

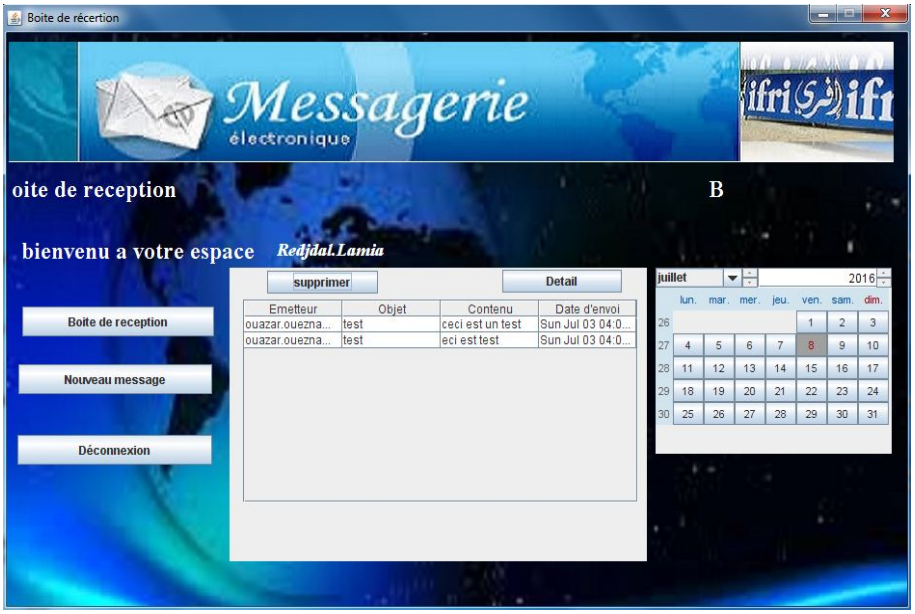

Figure 4.10: La fenêtre graphique de la boite de réception

4.5 Conclusion

Dans ce chapitre nous avons présenté les outils de réalisation de développement de notre application où nous avons utilisé le serveur Apache James et le langage de programmation java. Ainsi nous avons présenté quelques fenêtres graphiques de notre application.

Conclusion générale

Conclusion générale

Dans ce modeste travail, nous avons réalisé un système de messagerie pour l'entreprise d'agro-alimentaire IFRI, qui permet aux différents utilisateurs d'échanger des mails et des différents documents entre eux.

Dans un premier temps, nous avons défini et analysé les besoins de système en effectuant un stage au niveau de l'entreprise IFRI. Ensuite, nous avons entamé la phase de conception qui a permis de structurer et définir les besoins attendus du système en suivant la méthode de Grady Booch comme une méthode d'analyse et de conception. Enfin, nous avons abordé la réalisation en utilisant les outils d'implémentation appropriés; le Java comme langage de programmation et le serveur Apache James comme serveur de messagerie.

Nous espérons que ce travail facilitera et permettra une meilleure communication entre les différents utilisateurs de l'entreprise agro-alimentaire IFRI.

Notre application est limité en nombres de services qu'elle offre à cause des contrainte de serveur, par exemple elle nous permet pas de récupérer les messages envoyés par l'utilisateur.

En guide de perspective nous allons améliorer notre application on offrant la possibilité d'échanger des pièces jointes entre les différents utilisateurs et la possibilité de la télécharger ainsi que l'envoi d'un même message au plusieurs utilisateur.

Bibliographie

[1] Jean-François Pillou. Formation réseau. 2003.

[2] http://www.wikipédia.org, ,.

[3] Cour les réseaux informatique université nice sophia antipolis.

[4] Les réseaux - introduction bts 2éme année, support de cours.

[5] OMARI.K. La réalisation d'une application de contrôle total des processus d'un ordinateur distant,l2 mathematiques et informatique. université pédagogique nationale (upn). 2010.

[6] http://josich.over-blog.com.

[7] http://www.commentcamarche.net.

[8] http://irp.nain-t.net/doku.php/start.

[9] http://reussirsonccna.fr/protocole-ethernet,.

[10] BONI WILLY ENDERSON.M YEYE.Y. Mise en place d'un système de messagerie: cas de l'ita,licence professionnelle en science informatique. institut des technologies d'abidjan. 2011.

[11] DJOB.PN. Mise en place d'un système de messagerie électronique: Cas du fonds de prévoyance militaire,licence professionnelle en science informatique. ifpg - isfpt - ingénieur de conception réseaux et télécoms. 2008.

[12] CHEIKH.P C MACTAR.N S. Mise en place d'un système de messagerie sécurisée pour une pme/pmi ,memoire de maitrise informatisue, section informatique, ufr de sciences appliquées et de technologie. université gaston berger, saint-louis, sénégal. 2010.

[13] http://siguillaume.developpez.com/tutoriels/linux/mise-place-systeme-messagerie-electronique-sous-linux/?page=generalites.

[14] BOUCHE.S BUTEL.A CHAMON.E GONEL.JF BERGEROUN.R, MERTIN.M. Securité de la messagerie. clusif. 2005.

[15] GRADY.B. Object-oriented analysis and design with applications. Avril 2007.

[16] http://www.tutorialspoint.com/object_oriented_analysis_design/ooad_object%_oriented_paradigm.htm.

[17] MULLER.PA. Ensisa. 2001.

[18] http://james.apache.org.

[19] https://www.java.com/fr/download/faq/whatis_java.xml.

[20] ARNOLD.K GOSLING.J, HOLMES.D. Le langage java. 2001.

[21] http://www.oracle.com/technetwork/java/javamail/index-135046.html.

[22] http://www.ossdirectory.com/fr/produits-oss/single/ossproduct/javabeans-activation-frameworkjaf/.