

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
*Université Abderahmane Mira-Béjaïa*  
*Faculté des Sciences Exactes*  
*Département d'Informatique*



*Mémoire 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'une application pour la gestion de stock,*

*Cas d'étude : **BOUHZILA***

---

**Réalisé par :**

*Mr Abdessalem SAMAH*

**Soutenu devant le jury composé de :**

<i>Mouloud DEMOUCHE</i>	<i>Président</i>
<i>Abdelouhab ALOUI</i>	<i>Encadreur</i>
<i>Nadir SALHI</i>	<i>Évaluateur</i>
<i>Salah ZEMMOUDJ</i>	<i>Évaluateur</i>

***Promotion: 2015 - 2016***

# *A propos ...*



*La copie finale (après la soutenance) a été déposée dans la bibliothèque des périodiques(université de Université Abderahmane Mira-Béjaïa), ainsi que le logiciel.*

*Si vous avez besoin d'une copie sous forme d'un fichier PDF, contactez*

*moi : [salamo18000@gmail.com](mailto:salamo18000@gmail.com)*

*A touts, un grand Merci.*

# Remerciment

*Après avoir accomplir ce modeste travail; Je tiens à remercier **Allah** qui m'a donnée le courage et la patience pour le continuer, Mr l'encadreur ALOUI Abdelouhab , qui m'a soutenu et pour leur effort durant la réalisation de ce travail...*

*Les jurys, m'ont honorés de leur présence, et je les remercies de bien vouloir juger ce travail...*

*Je remercies également, les enseignants de l'informatique de tous les modules qui ont participés à ma formation soit à l'université de Jijel durant les trois années de licence ,ou à l'université de Béjaia durant mes deux années de Master.*

*A tous un grand merci...*

*Adessalem*

# *Dédicace*

*Je dédié ce modeste travail à la sagesse et à la noblesse à mes  
aimables parents 'Baba' et 'Yemma' qui m'ont donné les meilleurs  
enseignements de la vie, que Dieu les préserver pour moi...*

*A mes sœurs Mariam, Farida et Hadjer .*

*A mon frère Mohammed.*

*A tout la grande famille, toutes mes cousins ....*

*A ma liste ouverte d' amis .....*

*A tous mes collègues de l'informatique...*

*A ceux que j'aime et qui m'aiment.*

*Je dédié ce travail...*

# Table des matières

<b>Table des matières</b>	<b>I</b>
<b>Liste des acronymes</b>	<b>VI</b>
<b>Liste des figures et tableaux</b>	<b>VII</b>
<b>Introduction générale</b> .....	<b>01</b>
<b>Chapitre 01 : Généralités sur les réseaux et les bases de données</b> .....	<b>03</b>
<b>1. Introduction</b> .....	<b>03</b>
<b>2. Les réseaux informatiques</b> .....	<b>03</b>
2.1 Objectif d'un réseau informatique .....	03
2.2 Généralités sur les réseaux .....	03
2.3 Les modes de fonctionnement .....	04
2.4 Classification des réseaux .....	07
2.5 Internet .... ..	08
2.6 Les réseaux locaux d'entreprise .....	08
2.7 Intranet .....	09
<b>3. Les bases de données</b> .....	<b>10</b>
3.1 Définition d'une base de données .....	10
3.2 Utilité d'une base de données .....	10
3.3 Type d'une base de données.....	10
3.4 Objectif de distribution d'une base de données .....	11
3.5 Systèmes de gestion d'une base de données (SGBD) .....	11

# Liste des acronymes

Acronymes	Signification
<b>AGS</b>	<b>A</b> pplication <b>G</b> estion de <b>S</b> tock
<b>SGBD</b>	<b>S</b> ystème de <b>G</b> estion de <b>B</b> ase de <b>D</b> onnées
<b>SGBDR</b>	<b>S</b> ystème de <b>G</b> estion de <b>B</b> ase de <b>D</b> onnées <b>R</b> elationnel.
<b>UML</b>	<b>U</b> nified Modeling <b>L</b> anguage
<b>LAN</b>	<b>L</b> ocal <b>A</b> rea <b>N</b> etwork
<b>SQL</b>	<b>S</b> tructured <b>Q</b> uery <b>L</b> angage
<b>URL</b>	<b>U</b> niform <b>R</b> esource <b>L</b> ocator
<b>WAN</b>	<b>W</b> ide <b>A</b> rea <b>N</b> etwork
<b>WWW</b>	<b>W</b> orld <b>W</b> ide <b>W</b> eb
<b>MAN</b>	<b>M</b> etropolitan <b>A</b> rea <b>N</b> etwork
<b>PHP</b>	<b>H</b> ypertext <b>P</b> reprocessor

# Liste des figures et tableaux

Figure	Description	Page
<b>Chapitre 01: Généralités sur les réseaux et les bases de données</b>		
<b>Figure 1.1</b>	Le modèle client/serveur	4
<b>Figure 1.2</b>	Fonctionnement d'un système client/serveur	5
<b>Figure 1.3</b>	Architecture à deux niveaux.	5
<b>Figure 1.4</b>	Architecture à trois niveaux.	6
<b>Figure 1.5</b>	Principe de fonctionnement d'un P2P centralisé	7
<b>Figure 1.6</b>	Principe de fonctionnement d'un P2P décentralisé	7
<b>Chapitre 02 : Capture des besoins</b>		
<b>Figure 2.1</b>	L'organigramme générale de BOUHZILA	14
<b>Figure 2.2</b>	Diagramme de contexte statique du système à réaliser.	19
<b>Figure 2.3</b>	Représentation graphique de la relation d'inclusion.	27
<b>Figure 2.4</b>	Représentation graphique de la relation d'extension.	28
<b>Figure 2.5</b>	Diagramme de cas d'utilisation associé à l'utilisateur.	29
<b>Chapitre 03 : Analyse et conception</b>		
<b>Figure 3.1</b>	Le diagramme de séquence pour le scénario authentification	33
<b>Figure 3.2</b>	Le diagramme de séquence pour le scénario Ajouter un produit	34
<b>Figure 3.3</b>	Le diagramme de séquence pour le scénario Modifier un produit	35
<b>Figure 3.4</b>	Le diagramme de séquence pour le scénario Supprimer un produit	36
<b>Figure 3.5</b>	Le diagramme de séquence pour le scénario Saisir une commande	37
<b>Figure 3.6</b>	Le diagramme de séquence pour le scénario Etablir un bon de livraison	38
<b>Figure3.7</b>	Le diagramme de séquence pour facturer un bon de livraison	39
<b>Figure3.8</b>	Le diagramme de séquence pour l'ajout d'une nouvelle quantité entré	39
<b>Figure3.9</b>	Le diagramme de classe détaillé	42

# Introduction général

Comme toute entreprise qui consacre de grands efforts pour améliorer la qualité de ces activités, Groupe BOUHZILA est un' entreprise qui veut faire face à l'évolution en mettant en œuvre un système d'information pour l'automatisation de ces tâches.

Vu les avantages d'automatisation des systèmes d'information, l'exploitation des technologies avancées de traitement de l'information est devenue une nécessité dans la gestion des entreprises et cela pour assurer une bonne démarche et une gestion efficace de leurs ressources, par l'utilisation d'un logiciel.

Le bureau du gestionnaire est plus importants des l'entreprise, pour ce la l'informatique doit être capable tout simplement de mieux gérer le stock de l'entreprise. Pour faire face aux problèmes posés par le gestionnaire de stock, nous allons proposer un projet que nous avons appelée "*AGS*" (*Application de Gestion de Stock*), qui doit assurer la gestion des plusieurs éléments.

Pour avoir ce produit logiciel nous somme basé sur un ensemble de méthodes et de langages. Pour la modélisation, nous avons utilisé l'UML, ainsi que le langage de programmation JAVA avec le SGBD MYSQL.

Ce mémoire est organisé en quatre chapitres dans le :

- **Chapitre 1** : Présente généralités sur les réseaux et les bases de données
- **Chapitre 2**: L'étude préalable, ce chapitre contient la capture des besoins fonctionnels et les besoins techniques relatifs à notre domaine d'application.
- **Chapitre 3**: Présente deux parties, une partie d'analyse du système qui englobe le découpage en catégories, et une partie de la conception qui explique comment commencer à concevoir le système, indépendamment de son contenu fonctionnel et décrit le passage du modèle à objet au modèle relationnel.





# Chapitre 01

## Généralités sur les réseaux et les bases de données

- *Introduction*
- *Les réseaux informatiques*
- *Les bases de données*
- *Conclusion*

## 1. Introduction

Les réseaux sont issus de la nécessité de communication entre les machines. Ils sont la conséquence d'un partage équitable des ressources technologiques. Ils sont indispensables dans la vie quotidienne car notre société est à la recherche des technologies fonctionnant au tout numérique. La principale utilisation des réseaux a bien étendu l'accès aux bases de données, elles sont actuellement au cœur du système d'information des entreprises. De plus, ces données sont la plupart du temps de nature diverse et complexe (valeurs numériques, textes, messages, animations, image, vidéo...etc [M 4]).

## 2. Les réseaux informatiques

Les réseaux informatiques sont nés du besoin de faire communiquer des terminaux distants avec un site central puis des ordinateurs entre eux et enfin de connecter des machines telle que des stations de travail avec leurs serveurs [M 4].

### 2.1 Objectif d'un réseau informatique

Un réseau informatique peut servir plusieurs buts distincts : [M 4]

- \_ Partager les fichiers.
- \_ Transférer les fichiers.
- \_ Partager les applications : logiciels ou les bases de données.
- \_ Partager les équipements (imprimantes, modems, fax).
- \_ Dialoguer entre les utilisateurs connectés : messagerie électronique, conférence électronique,..

### 2.2 Généralités sur les réseaux

Un réseau est un ensemble d'objets interconnectés les uns avec les autres qui permettent de faire circuler des éléments entre chacun de ces objets selon des règles bien définies [M 4].

- **Client d'un réseau**

Les clients d'un réseau sont habituellement des ordinateurs de bureau, dotés d'un stockage local et d'une capacité de traitement. Il existe aussi des clients légers moins puissants et moins onéreux que les ordinateurs de bureaux, mais de gestion facile sur les réseaux client/serveur. Ils ne disposent pas de stockage local et dépendent des serveurs du réseau pour leurs applications, leurs données, leurs stockages et d'autres ressources.

- **Serveur d'un réseau**

Les serveurs des réseaux, qui travaillent en coulisse pour fournir les services requis par les ordinateurs client (machines et station de travail de bureau), peuvent se classer en différentes catégories, selon le type de tâche qu'ils accomplissent. Il existe plusieurs types de serveur,

on cite quelque exemple :

- \_ **Serveur de fichier et d'impression** : Ce serveur s'occupe de la gestion des fichiers, il gère les accès des utilisateurs aux fichiers et aux imprimantes partagées.
- \_ **Serveur d'application** : Ces serveurs fonctionnent comme des serveurs d'applications, le téléchargement vers le poste client ne concerne qu'une partie de données.
- \_ **Serveur de communication** : Il gère les échanges des données et des courriers électroniques entre son propre réseau et d'autres réseaux.

- **Protocole d'un réseau**

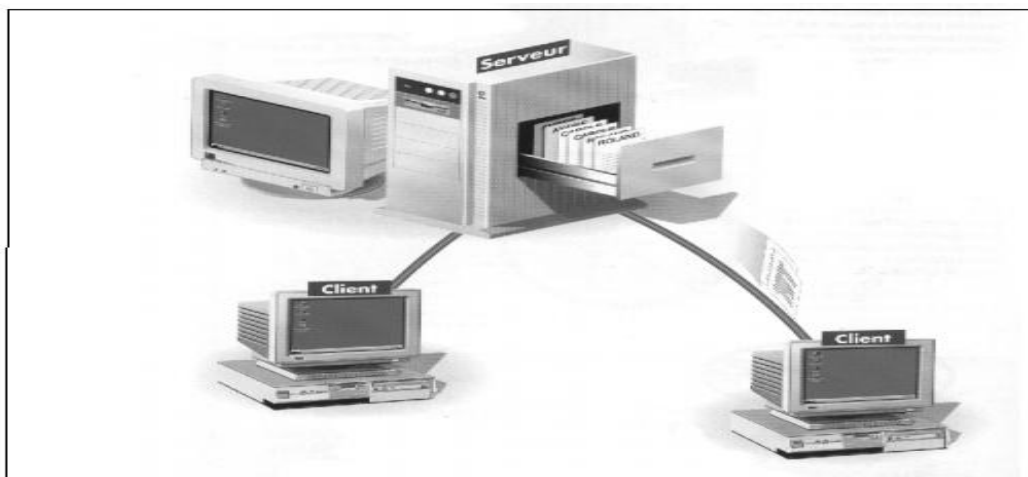
Comme définition générale, le protocole réseau représente le langage utilisé sur la connexion pour communiquer entre les machines. Chaque ordinateur ou périphérique réseau doit utiliser le même protocole pour pouvoir se comprendre, il commande le déroulement des communications. Ils se basent soit sur le modèle OSI, le modèle TCP/IP,....

### 2.3 Les modes de fonctionnement

On distingue deux modes de fonctionnement : [M 4]

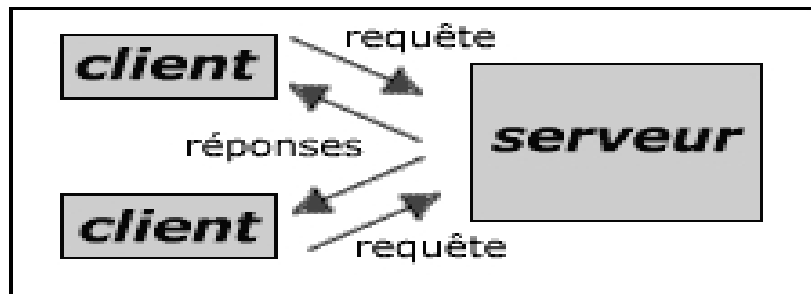
#### ✓ Le modèle client/serveur

**Présentation de l'architecture** : Le modèle client/serveur signifie que des machines clientes contactent un serveur, un serveur est une machine généralement très puissante en terme de capacité d'entrée-sortie, qui leur fournit des services. Ces services sont des programmes fournissant des données telles que l'heure, des fichiers, une connexion...etc. les services sont exploités par des programmes clients s'exécutant sur des machines clientes. Dans un environnement purement client/serveur les ordinateurs du réseau (les clients) ne peuvent voir que le serveur, c'est l'un des principaux atouts de ce modèle.



**Figure 1.1** – Le modèle client/serveur [L3]

**Fonctionnement d'un système client/serveur :** Un système client/serveur fonctionne selon le schéma suivant :



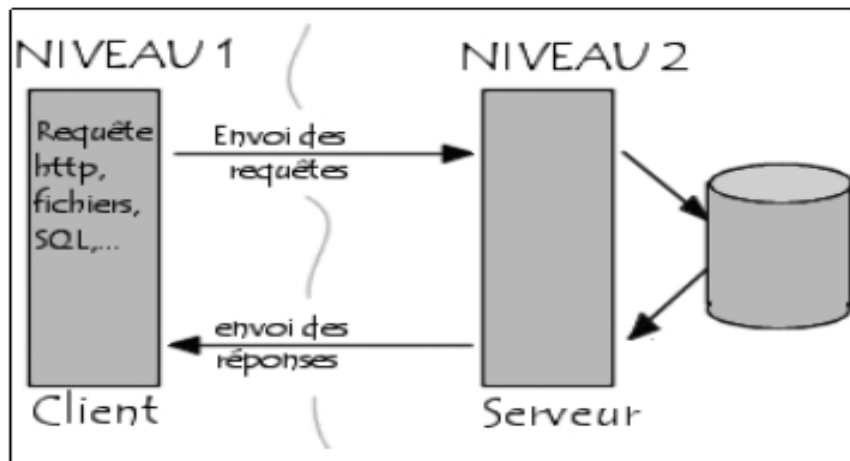
**Figure 1.2** – Fonctionnement d'un système client/serveur [L3]

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

\_ Le serveur reçoit la demande et répond à l'aide de l'adresse de la machine cliente et son port.

**Type d'architecture client/serveur :** On distingue trois types d'architecture :

\_ Architecture à deux niveaux : L'architecture à deux niveaux (aussi appelée architecture 2-tier) caractérise les systèmes clients/serveurs dans lesquels le client demande une ressource et le serveur la lui fournit directement. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir le service.



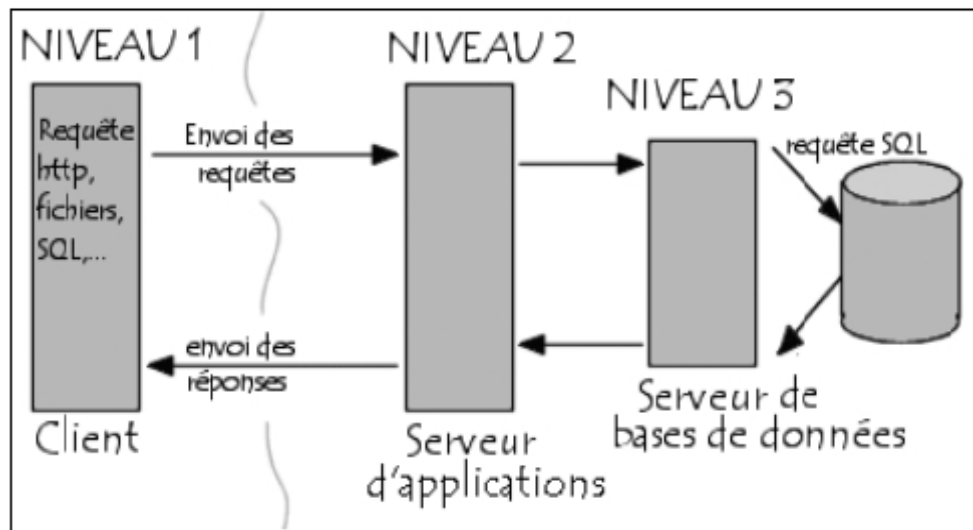
**Figure 1.3** – Architecture à deux niveaux[L3].

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

\_ Le client : le demandeur des ressources.

\_ Le serveur d'application (appelé aussi middleware) : le serveur chargé de fournir la ressource mais faisant appel à un autre serveur.

\_Le serveur secondaire (généralement un serveur de base de données), fournissant un service au premier serveur.



**Figure 1.4** – Architecture à trois niveaux [L3].

● **Architecture multi-niveaux** : Dans l'architecture à trois niveaux, chaque serveur (niveaux 1 et 2) effectue une tâche (un service) spécialisée. Ainsi, un serveur peut utiliser les services d'un ou plusieurs autres serveurs afin de fournir son propre service. Par conséquent, l'architecture à trois niveaux est potentiellement une architecture à N niveaux.

✓ **Le modèle poste à poste (égal à égal)**

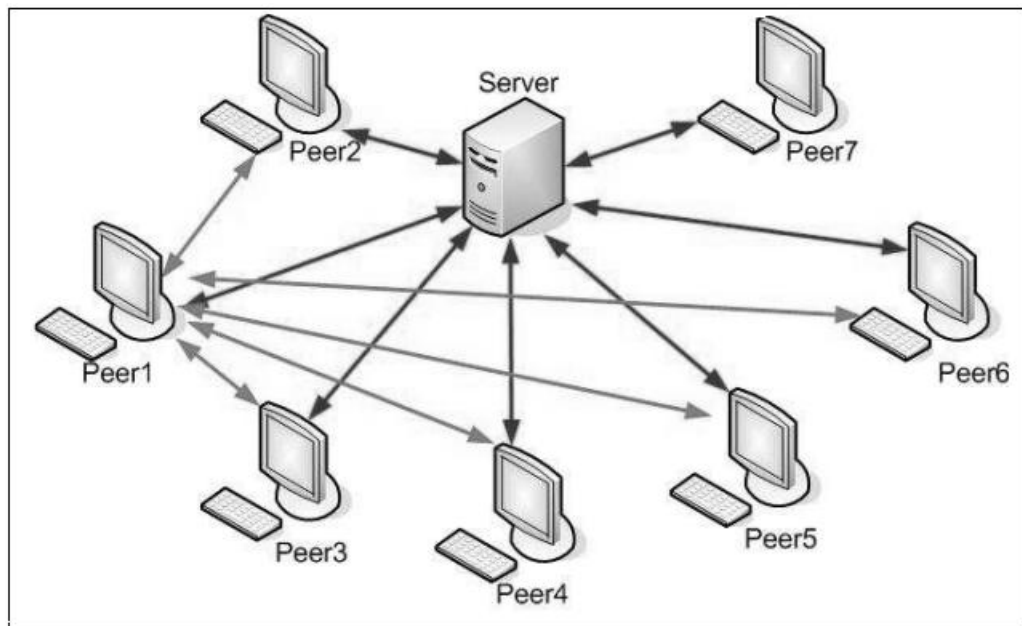
● **Présentation** : Le terme poste à poste désigne un modèle de réseau informatique dont les éléments (les nœuds) sont à la fois clients et serveurs lors des échanges d'informations. Ce modèle permet une décentralisation des réseaux, en offrant une alternative aux traditionnelles architectures client/serveur [M 4].

● **Fonctionnement d'un système poste à poste** : Les réseaux postes à postes sont utilisés dans différents domaines. Le plus connu est le partage de fichiers à travers Internet, en assurant l'échange de fichiers entre les internautes. La différence avec un téléchargement classique c'est que le fichier téléchargé ne se trouve pas sur un serveur ftp (file transmission protocol) accessible via un site web mais directement sur le disque dur d'un autre internaute. Chaque internaute peut ainsi disposer sur son disque dur d'un nombre important de fichiers. On dit alors que les fichiers sont partagés, c'est-à-dire accessibles aux autres internautes [M 4].

● **L'architecture des réseaux poste à poste**

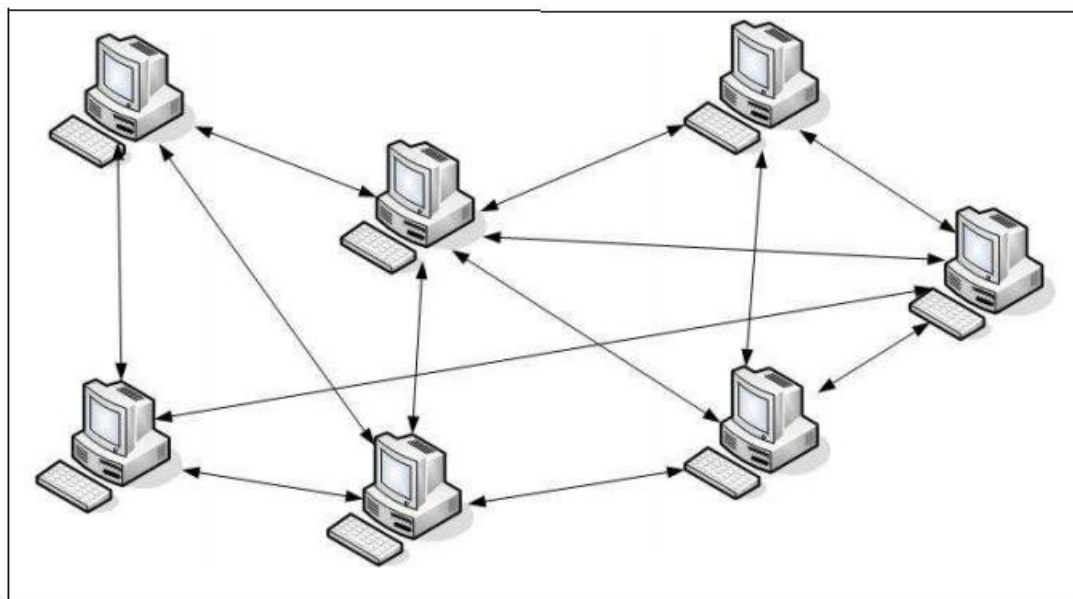
\_ **L'architecture centralisée** : Il existe un serveur central, gérant l'identité des utilisateurs, l'indexation des partages, les recherches et la mise en relation des pairs : le serveur donne à chaque client l'adresse des pairs possédant le fichier recherché. Le client se connecte alors à ce pair pour

échanger directement les données. Les fichiers ne passent donc pas par le serveur. L'index est donc centralisé, mais les fichiers sont décentralisés [M 4].



**Figure 1.5** – Principe de fonctionnement d'un P2P centralisé [L3]

\_ **L'architecture décentralisée** : Les réseaux totalement décentralisés sont dits "pures". Dans ce type d'architecture, l'ensemble des nœuds sont égaux et jouent le même rôle.



**Figure 1.6** – Principe de fonctionnement d'un P2P décentralisé [L3]

## 2.4 Classification des réseaux

On distingue différents types de réseaux selon leurs tailles (en terme de nombre de machines), leurs vitesses de transfert des données ainsi que leurs étendues. On fait généralement trois catégories de réseaux : [M 4]

\_ **Réseau Local** : En anglais Local Area Network (LAN), 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 .

**Métropolitain** : En anglais Area Network (MAN), interconnectent plusieurs LAN géographiquement proches (au maximum quelques dizaines de km) à des débits importants. Ainsi un MAN permet à deux noeuds distants de communiquer comme si ils faisaient partie d'un même réseau local. Un MAN est formé de commutateurs ou de routeurs interconnectés par des liens hauts débits (en général en fibre optique).

**Réseau étendu** : En anglais Wide Area Network (WAN ), interconnecte plusieurs LAN à travers de grandes distances géographiques. Les débits disponibles sur un WAN résultent d'un arbitrage avec le coût des liaisons (qui augmente avec la distance) et peuvent être faibles. Les WAN fonctionnent grâce à des routeurs qui permettent de "choisir" le trajet le plus approprié pour atteindre un noeud du réseau. Le plus connu des WAN est Internet.

## 2.5 Internet

Le réseau internet est un réseau de communication mondial. Les ordinateurs sont reliés entre eux par une ligne téléphonique ou par satellite [M 4].

### ♦ Le World Wide Web (ou WWW ou W3, souvent appelé Web)

Signifie littéralement la "toile d'araignée recouvrant le monde". Cette image représente tous les ordinateurs interconnectés à travers le monde, mais le World Wide Web désigne plus précisément le système hypertexte que supporte le réseau Internet. Les liens hypertextes sont comme les fils d'une toile d'araignée qui relient les pages d'un site à l'autre.

### ♦ URL

L'adresse d'un site (URL) donne de nombreuses informations d'un seul coup d'oeil (si l'URL n'est pas écrite, il suffit de pointer la souris sur le lien, l'URL apparaît en bas du navigateur). Elle se présente de la manière suivante : protocole ://ordinateur/répertoire/document.html ancre

\_ **protocole** : Protocole utilisé pour le dialogue entre les différentes machines connectées. Http est le protocole utilisé pour les pagesWeb. Il existe d'autres protocoles (par exemple FTP, gopher...).

\_ **ordinateur** : Nom de l'ordinateur sur lequel se trouvent les fichiers (nom d'hôte). Ce nom d'hôte est divisé en domaine et sous domaine. Les domaines principaux sont les suivants :

\_com : commercial.

– edu : universités et éducation (US) .

– gov : organismes gouvernementaux .

– int : organisations internationales .

– mil : militaires (US) .



- net : réseaux .
- org : ONG.
- \_ Répertoire : Répertoire dans lequel se trouve, sur le serveur, le document. Souvent, il existe des sous répertoires.
- \_ **document.html** : Nom de la page Web, ou du fichier demandée. -ancre : Détermine un endroit précis de la page Web .

## 2.6 Les réseaux locaux d'entreprise

Un réseau local d'entreprise se caractérise essentiellement par la zone géographique de couverture spécifiquement à l'intérieur d'une entreprise (RLE : réseau Local d'Entreprise). Cette zone de dimension locale a permis à des technologies spécifiques d'être déployées : haut-débit, multipoint avec contrôle d'accès au média,...etc. Par des acteurs essentiellement du monde informatique pour gérer et partager des ressources internes des entreprises. L'organisation de l'entreprise à travers ses services ainsi que les limitations de certaines technologies conduisent à introduire des éléments actifs d'amplification du signal (hub) ou d'interconnexion (pont, routeur, commutateur) dans l'architecture du réseau.

## 2.7 Intranet

L'intranet est un réseau local, c'est-à-dire, un système de communication entre ordinateurs sur une courte distance (quelques centaines de mètres), permettant le partage de ressources informatiques (fichiers, logiciels, imprimantes, scanners). L'ordinateur qui gère les ressources est appelé serveur, l'ordinateur traitant les informations de l'utilisateur est appelé poste de travail ou station. Il est doté des mêmes fonctions qu'Intranet. Son accès est verrouillé par un mot de passe. Il y a une sauvegarde centralisée des données sur le serveur. Ce réseau est surtout utilisé par les entreprises et les collèges. [M 4]

### ▪ L'intérêt de l'Intranet

Des documents de tous types (textes, images, vidéos, sons, ...) peuvent être mis à disposition sur un Intranet. De plus, un Intranet peut réaliser une fonction de "groupware" très intéressante, c'est-à-dire permettre un travail coopératif. Voici quelques unes des fonctions qu'un Intranet peut réaliser :

- \_ Mise à disposition d'informations sur l'entreprise (panneau d'affichage),
- \_ Mise à disposition de documents techniques,
- \_ Un échange de données entre collaborateurs,
- \_ Annuaire du personnel,
- \_ Gestion de projets, aide à la décision, agenda, ingénierie assistée par ordinateur,
- \_ Forums de discussion, listes de diffusions, chat en direct.

De cette façon un Intranet favorise la communication au sein de l'entreprise et limite les erreurs dues à la mauvaise circulation d'une information. L'information disponible sur l'Intranet doit être mise à jour en évitant les conflits de version.

#### ▪ Service et fonctionnement de l'Intranet

L'Intranet n'est pas un écran "libre service" où tout le monde a accès dans l'Entreprise, mais un outil de communication et donc de travail, rapide, utile et destiné à des utilisateurs nommément désignés et disposant d'un niveau d'accès limité suivant leurs postes et leurs fonctions.

### 3. Les bases de données

Apparu au début des années soixante, le concept de base de donnée voulait pallier à la fois les insuffisances des systèmes de gestion des fichiers et les inconvénients liés au mode de développement des applications [M 3].

#### 3.1 Définition d'une base de données

On peut définir une base de données comme une collection de données opérationnelles, enregistrées sur un support adressable et utilisées par des systèmes d'application (les programmes) d'une organisation (humain) particulière. En outre, la collection de données est structurée indépendamment d'une application particulière, elle est cohérente, de redondance minimale et accessible simultanément par plusieurs utilisateurs. Pratiquement une base de données est constituée d'enregistrements qui regroupent un ensemble d'informations(champs) liées et traitées comme une entité unique. L'ensemble d'enregistrements partagent les mêmes champs s'appel une table. enfin une base de données peut contenir plusieurs tables qui sont liées entre elles ou restent indépendantes .

#### 3.2 Utilité d'une base de données

Une base de données permet de mettre des données à la disposition d'utilisateurs pour une consultation, une saisie ou bien une mise à jour, tout en s'assurant des droits accordés à ces derniers. Cela est d'autant plus utile que les données informatiques sont de plus en plus nombreuses. L'avantage majeur de l'utilisation des bases de données est la possibilité de pouvoir être accédées par plusieurs utilisateurs simultanément. L'objectif des bases de données est de pouvoir :

- \_Mémoriser de nouvelles informations .
- \_ Modifier les informations existantes.
- \_ Retrouver facilement des informations mémorisées.

Pour réaliser ces objectifs, on a besoin de structurer les informations [M 3].

#### 3.3 Type d'une base de données

La plupart des cas, on a besoin de manipuler un grand nombre de données. Pour ce faire, il peut être utile de les regrouper dans une base de données. Ces données doivent pouvoir être accédés par un ou plusieurs utilisateurs locaux ou distants. Il existe deux types de base de données :

#### ● Bases de données centralisées

Une base de données centralisée est une collection de données reliées entre elles, stockées ensemble, aussi peu redondante que possible pour être utilisées par une ou plusieurs applications, elle est gérée par un seul SGBD, est stockée dans sa totalité à un emplacement physique unique et ses divers traitement sont confiés à une seule et même unité de traitement.

#### ● Base de données réparties

Une base de données répartie (BDR) est une base de données dont différentes parties sont stockées sur des sites, et gérée par plusieurs processeurs, sites ou SGBD généralement géographiquement distants, reliés par un réseau. La réunion de ces parties forme la base de données répartie. Un système de base de données réparti est suffisamment complet pour décharger les utilisateurs de tous les problèmes de concurrence, fiabilité, optimisation des requêtes ou transaction sur des données gérées par différents SGBD sur plusieurs sites [M 3].

### 3.4 Objectif de distribution d'une base de données

- A l'heure actuelle, de nombreuses entreprises ont des ramifications partout dans le monde. Les états décentralisent leurs administrations. Les bases de données réparties ont donc une architecture plus adaptée leurs organisations.
- Plus de fiabilité : les BDR sont souvent répliquées. La panne d'un site n'est pas très importante pour l'utilisateur, qui s'adressera à autre site.
- Meilleures performances : réduire le trafic sur le réseau est une possibilité d'accroître les performances. Le but de la répartition des données est de les rapprocher de l'endroit où elles sont accédées. Répartir une base de données sur plusieurs sites permet de répartir la charge sur les processeurs et sur les entrées/ sorties.

Faciliter l'accroissement : l'accroissement se fait par l'ajout de machines sur le réseau [M 3].

### 3.5 Systèmes de gestion d'une base de données (SGBD)

Un SGBD est un ensemble des programmes permettant la création, et l'utilisation des bases de données, ainsi que la structuration, le stockage, la mise à jour et la recherche des données dans une base de données. Les SGBD sont une évolution des systèmes de fichiers conventionnels, qui présentent une série d'inconvénients .

#### ✓ Les objectifs d'un système de gestion de base de données (SGBD)

Plus précisément, un SGBD permet de :

**Assurer l'indépendance des programmes aux données indépendance :**

- Physique des programmes aux données,
- Logique des programmes aux données,
- Administration facile,
- Permettre de manipuler par langages SQL.
- \_ Accès par des langages assertionnels :
  - Recherche (le quoi et non le comment),
  - Insertion (en groupes, calculées),
  - Mise à jour (basée sur la recherche).

**Efficacité des accès :**

- Temps de réponse.

**Support de transactions :**

- Atomique (tout ou rien),
- Cohérente (respect de l'intégrité).

**Partageabilité et sécurité des données :**

- Simultanéité lecture/écriture maximum,
- Accès transactionnels et décisionnels,
- Confidentialité (authentification, droits d'accès, cryptage),
- Restauration après pannes (journaux, sauvegardes) [M 3].

**✓ Types de système de gestion de base de données (SGBD)**

Les typologies des SGBD est généralement en fonction des types de structures de données que les SGBD supportent .

**Système de gestion de base de données centralisé :** Logiciel qui gère un ensemble de fichiers qui constituent les bases de données et qui permet à plusieurs utilisateurs de stocker et d'extraire des données de ces fichiers, dans des conditions d'intégrité et de confidentialité. (Protection sur pannes), partageables entre plusieurs utilisateurs et/ou programmes et manipulés indépendamment de leur représentation physique.

**Système de gestion de base de données reparté :** Système gérant une collection de base de données logiquement reliées, repartis sur les différents sites en fournissant un moyen d'accès rendant la distribution transparente.

**Système de gestion de base de données relationnel (SGBDR)**

Ce type de système de gestion de base de données permet de voir une base de données comme un ensemble de table .En outre, il est doté d'une algèbre qui lui confère une assise formelle solide. Les langages relationnels de manipulation de données se caractérisent par leur nature déclarative. Les

SGBD relationnels offrent un langage de manipulation de données standard, SQL en l'occurrence, qui est lui-même fondé sur l'algèbre relationnelle [M 3].

## **4 . Conclusion**

Dans ce premier chapitre, nous avons présenté les réseaux et les base de données en deux parties, la première est portée sur les réseaux informatiques, où on a vu les concepts d'un réseau, leurs objectifs ainsi que leurs types. La deuxième partie est portée sur les bases de données, où on a vu la notion d'une base de données, et les différents types des systèmes de gestion des bases de données.

# Chapitre 02

## Capture des besoins

- *Introduction*
- *Présentation de l'entreprise*
- *Cahier des charges*
- *Modélisation des besoins de système*
- *Diagramme de cas d'utilisation*
- *Conclusion*

# D 1. Introduction

ans les activités de capture des besoins, convient premièrement de considérer le système comme une boîte noire à part entière. L'étude préalable est une étape primordiale et un passage obligatoire, elle s'agit de décrire les besoins de notre système en se servant du langage naturel [M 4].

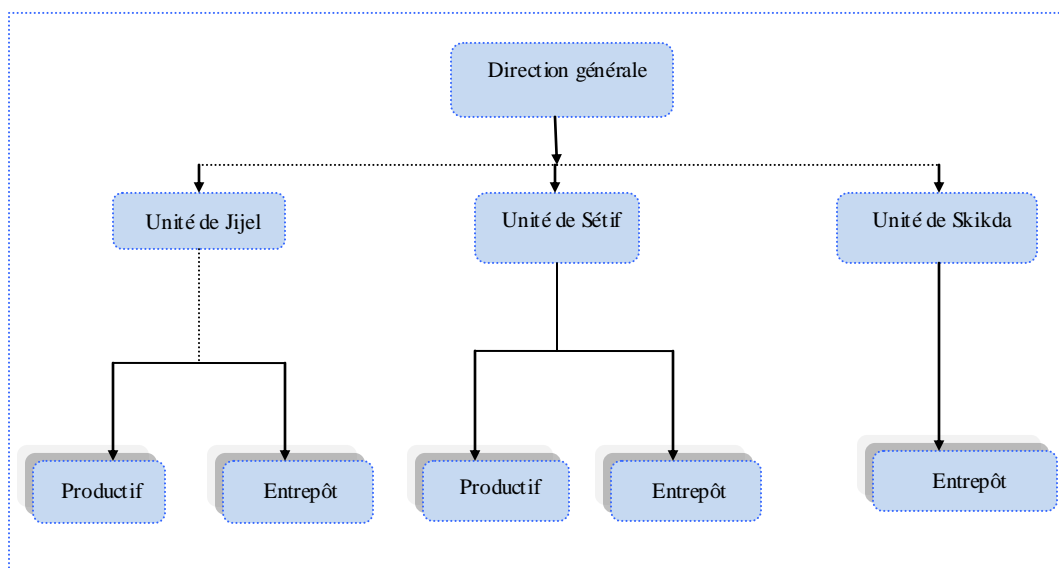
## 2. Présentation de l'entreprise

BOUHZILA est une entreprise priver et commerciale, elle créer en date de 21 décembre 2005 dont le siège social à Sétif, à pour mission prioritaire d'augmenté les capacités de stockage et conditionnement de produit alimentaire (aliments de poules, et de bétails ...) surgelé en grande quantité, destinés au marché local.

### 2.1 . Organigramme de BOUHZILA

- **L'unité de Sétif :** est la détraction générale, constitué d'une unité productif fourni trois entrepôts : l'entrepôt de Sétif, l'entrepôt de Skikda, et de Jijel .
- **L'unité de Jijel :** dont le siège est à Jijel, est constituée d'une unité productive à Taher et un entrepôt à Amir Abdelkader.

La figure ci-dessous montre d'une manière générale l'organigramme de l'entreprise :



**Figure 2.1** : l'organigramme générale de BOUHZILA

## 2.2 Présentation de notre sujet

Notre projet consiste en la réalisation d'une application de gestion de stock de l'entrepôt d'unité de Jijel qui soit parfaitement adaptée au fonctionnement spécifique de l'entreprise BOUHZILA.

Pour cela, je suis rendu sur place dans le cadre d'un stage afin de collecter les données nécessaires pour la réalisation de ce projet.

## 2.3. Problématique

L'entreprise BOUHZILA gère un stock important, ce stock a des aliments chair. La direction de BOUHZILA affronte des difficultés majeures dans la gestion de stock :

- ◆ Les erreurs de saisie et de calcul
- ◆ Les difficultés de gérer tout le stock par les papiers.
- ◆ Une faible sécurité des informations au niveau des documents.
- ◆ Recherche difficile sur les documents qui engendre une perte de temps.

## 3. Cahier des charges

Le cahier des charges est un document qui décrit de façon la plus précise possible, avec un vocabulaire simple, les besoins auxquels le maître d'œuvre doit répondre [M2].

Un besoin exprime un comportement ou une propriété que le système doit respecter chaque énoncé doit traduire la présence d'un comportement très spécifique [M2].

On distingue deux types de besoins les besoins fonctionnels et les besoins non fonctionnels.

### 3.1 Les besoins fonctionnels

Les besoins fonctionnels expriment une action que doit effectuer le système en réponse à une demande [M2].

L'analyse des besoins fonctionnels nous a amené à décomposer ces besoins pour chaque acteur de notre système en trois parties principales :



**a- Utilisateur**

L'utilisateur est le responsable sur le système et doit pouvoir :

- S'authentifier afin d'accéder à l'interface.
- Ajouter, modifier ou supprimer un produit, un chauffeur ou un camion.
- Mettre à jour la liste des clients ou des chauffeurs.
- Gérer et consulter les commandes.
- Etablir le bon de livraison et facturation.

**b- Client**

Le client appelle l'utilisateur et fait une commande du produit.

**c- Chauffeur**

Le chauffeur livre le produit au client.

**3.2. Les besoins non fonctionnels**

Il s'agit des besoins qui caractérisent le système. Ce sont des besoins en matière de performance ou de type matériels. Ces besoins peuvent concerner les contraintes d'implémentation [M2].

Dans le cadre de ce travail, l'application devra être extensible, c'est-à-dire qu'il pourra y avoir une possibilité d'ajouter ou modifier de nouvelles fonctionnalités, et devra aussi répondre aux caractéristiques suivantes :

- L'accès doit se faire au moyen d'un nom utilisateur et d'un mot de passe.
- Le système doit être interactif et fiable
- L'interface doit être simple à cerner par l'utilisateur

**3.3 Objectif de travail**

Le but principal de notre travail est de simplifier les différentes tâches afin de les accélérer et de diminuer le risque d'erreur généré par le recopiage manuel d'un grand nombre d'informations.

Le développement est centré sur la création d'une interface très simple et ergonomique afin que la simplification des tâches soit réelle et importante.

De plus, l'apport de l'outil informatique devait permettre l'utilisation et le traitement de nouvelles informations qui sont trop fastidieuses (car trop volumineuses) à traiter manuellement.

En effet, le travail est basé sur le fait de retrouver au moins autant d'informations qu'avec un traitement manuel et d'avoir une interface très simple afin de ne pas perdre de temps et même aussi possible de gagner le temps.

## 4. Modélisation des besoins du système

Pour la modélisation des besoins, nous avons opté pour les diagrammes suivants : le diagramme de contexte qui va nous permettre d'avoir une vue générale de notre système et les diagrammes de cas d'utilisation qui vont aider à identifier les acteurs à recenser les cas d'utilisation.

### 4.1 Description du contexte de l'application

La description du contexte de l'application se fait à travers les trois étapes suivantes : l'identification des acteurs qui interagissent avec notre système, l'identification des messages qui transiteront entre les acteurs et le système, et le diagramme de contexte.

#### 4.1.1 Identification des acteurs

Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié. Un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données [M2].



Client
Chauffeur

**Tableau 2.1** : Les acteurs du système à réaliser

#### 4.1.2 Identification des messages échangés entre le système et les acteurs

Un message représente la spécification d'une communication entre objets qui transporte les informations avec l'intention de déclencher une activité chez les récepteurs [M2].

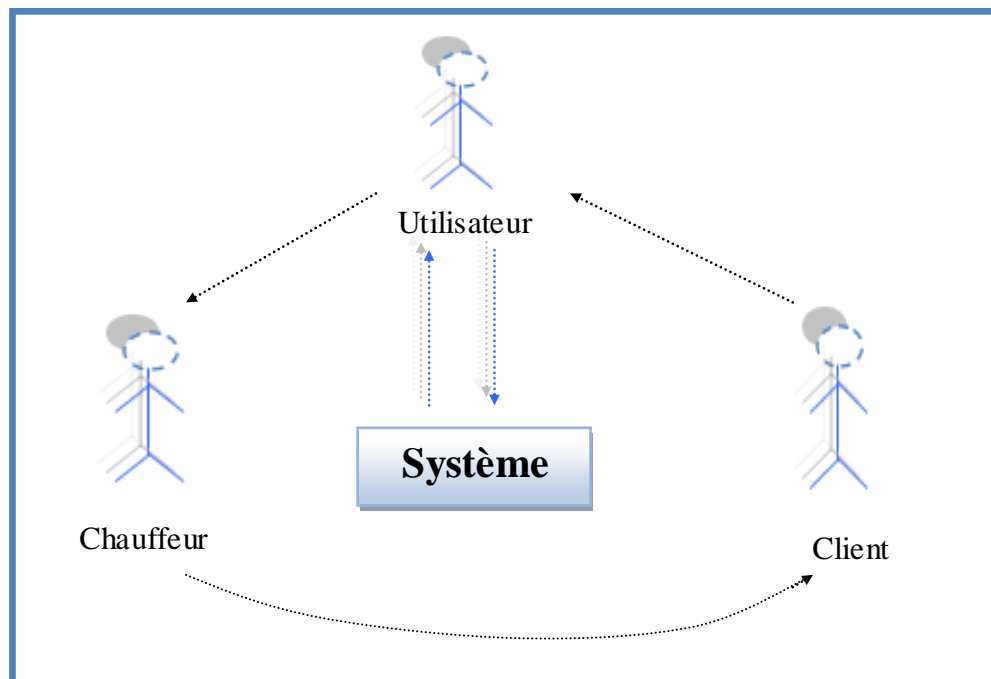
Les messages entre le système et ces acteurs sont :

1. Utilisateur → AGS
<ul style="list-style-type: none"> <li>✉ Demande d'authentification.</li> <li>✉ Demande d'ajout, modifier et supprimer d'un produit.</li> <li>✉ Demande d'ajout, modifier et supprimer d'un client.</li> <li>✉ Demande d'ajout, modifier et supprimer d'un chauffeur.</li> <li>✉ Demande d'ajout, modifier et supprimer d'un camion.</li> <li>✉ Demande de remplir le bon de commande.</li> <li>✉ Demande de remplir le bon de livraison.</li> <li>✉ Demande d'imprimer la facture.</li> </ul>
2. AGS → Utilisateur
<ul style="list-style-type: none"> <li>✉ Formulaire d'authentification.</li> <li>✉ Formulaire d'ajout et modifier et l'interface de suppression d'un produit.</li> <li>✉ Formulaire d'ajout et modifier et l'interface de suppression d'un client.</li> <li>✉ Formulaire d'ajout et modifier et l'interface de suppression d'un chauffeur.</li> <li>✉ Formulaire d'ajout et modifier et l'interface de suppression d'un camion.</li> <li>✉ Formulaire du bon de commande.</li> <li>✉ Formulaire de bon de livraison.</li> <li>✉ La facture imprimée.</li> </ul>

**Tableau 2.2** : Légende des messages des diagrammes de contexte dynamique

#### 4.1.3 Modélisation du contexte

Tous les messages identifiés précédemment peuvent être représentés de façon synthétique sur un diagramme.



**Figure 3.2 :** Diagramme de contexte statique du système à réaliser.

## 4.2 Identification des cas d'utilisation

Un cas d'utilisation représente un ensemble de séquences d'actions réalisés par le système et produisant un résultat observable intéressant pour un acteur particulier. [L1]

	Le cas d'utilisation		Acteur
01	S'authentifier		
02	Gérer les produits	Ajout.	
		Modification.	

		Suppression.	Utilisateur
03	Gérer les clients.	Ajout.	
		Modification.	
		Suppression.	
04	Gérer les commandes	Ajout.	
		Modification.	
05	Modifier la quantité en stock		
06	Etablir un bon de livraison		
07	Facturer un bon de livraison		
08	Gérer les camions	Ajout.	
		Modification.	
		Suppression.	
09	Gérer les chauffeurs	Ajout.	
		Modification.	
		Suppression.	

**Tableau 2.3 :** Les cas d'utilisation du système à réaliser.

### 4.3 Description textuelle des cas d'utilisation

Elle est organisée de la façon suivante :

- **Un sommaire d'identification** : va résumer les propriétés du cas d'utilisation.
- **Une description des enchaînements** : pré conditions au déclenchement du cas d'utilisation doivent être spécifiées, un scénario nominal décrivant celui-ci additionné à des scénarios alternatifs et d'exceptions. [L1]

Cas d'utilisation : <i>Authentification</i>	
Sommaire d'identification	
<b>Titre</b>	Connexion de l'utilisateur au système.
<b>But</b>	Connecter au système.
<b>Acteur</b>	L'utilisateur
Description des enchaînements	
<b>Pré condition</b>	Aucune.
<b>Enchaînements</b>	<ul style="list-style-type: none"> <li>a. L'utilisateur demande une connexion au système.</li> <li>b. Le système demande le nom d'utilisateur et le mot de passe puis valide.</li> <li>c. Le système vérifie l'existence de l'utilisateur et la validité de son mot de passe.</li> <li>d. Le système ouvre une session et affiche l'espace correspondant.</li> </ul>
<b>Exceptions</b>	Dans le cas où les informations fournies sont incorrectes, le système renvoie un message d'erreur, réaffiche le formulaire d'authentification et attend que l'utilisateur ressaisisse ses informations.

Cas d'utilisation: <i>Gérer les produits</i>	
Sommaire d'identification	
<b>Titre</b>	Mettre à jour la base produit
<b>But</b>	Ce cas d'utilisation permet de créer un nouveau produit, de le modifier ou de le supprimer.
<b>Acteur</b>	L'utilisateur
Description des enchaînements	
<b>Pré condition</b>	L'utilisateur est authentifié.
<b>Enchaînements</b>	<ul style="list-style-type: none"> <li>a. L'utilisateur demande d'accéder à l'interface de gestion de produit.</li> <li>b. Le système affiche l'interface de gestion de produit.</li> <li>c. L'utilisateur remplir les champs (cas de saisir) ou sélectionner un produit depuis la table des produits (cas modifier ou supprimer) puis</li> </ul>

	<p>valide l'opération.</p> <p>d. Le système sauvegarde la mise à jour des produits dans la base de données puis, affiche un message de confirmation.</p>
<b>Exceptions</b>	Le système affiche un message d'erreur si l'une des informations sont incorrecte ou ce nom produit est déjà exist dans la base de données.

<b>Cas d'utilisation: Gérer les clients</b>	
<b>Sommaire d'identification</b>	
<b>Titre</b>	Mettre à jour la base client.
<b>But</b>	Ce cas d'utilisation permet de créer un nouveau client, de le modifier ou de le supprimer.
<b>Acteur</b>	L'utilisateur.
<b>Description des enchaînements</b>	
<b>Pré condition</b>	L'utilisateur est authentifié.
<b>Enchaînements</b>	<p>a. L'utilisateur demande d'accéder à l'interface de gestion de client.</p> <p>b. Le système affiche l'interface de gestion de client.</p> <p>c. L'utilisateur remplir les champs (cas de saisir) ou sélectionner un client depuis la table des clients (cas modifier ou supprimer) puis valide l'opération.</p> <p>d. Le système sauvegarde la mise à jour des clients dans la base de données puis, affiche un message de confirmation.</p>
<b>Exceptions</b>	Le système affiche un message d'erreur si l'une des informations sont incorrecte ou le nom de registre commerce est déjà exist dans la base de données.

<b>Cas d'utilisation: Gérer les commandes</b>	
<b>Sommaire d'identification</b>	
<b>Titre</b>	Mettre à jour la base de données commande
<b>But</b>	Ce cas d'utilisation permet de saisir une nouvelle commande ou de la

	modifier.
<b>Acteur</b>	L'utilisateur.
<b>Description des enchaînements</b>	
<b>Pré condition</b>	L'utilisateur est authentifié.
<b>Démarrage</b>	Ce cas d'utilisation commence lorsque le client fait un appel à l'utilisateur et lui fournis des informations sur la commande que veut commander (cas de commander, modifier et annuler).
<b>Enchaînements</b>	<ol style="list-style-type: none"> <li>L'utilisateur demande d'accéder à l'interface commande.</li> <li>Le système affiche l'interface de gestion de commande.</li> <li>L'utilisateur remplir les champs et demande de choisir le client</li> <li>Le système affiche l'interface de liste des clients</li> <li>L'utilisateur choisir le client et valider en suit demande de choisir le produit.</li> <li>Le système affiche l'interface de liste des produits.</li> <li>L'utilisateur choisir le produit et valider.</li> <li>Le système sauvegarde la mise à jour la table commande et contenir dans la base de données puis, affiche un message do confirmation.</li> </ol>
<b>Exceptions</b>	Un message d'erreur est affiché sur l'écran si l quantité commander et supérieur à la quantité stocké.

<b>Cas d'utilisation: Modifier la quantité en stock</b>	
<b>Sommaire d'identification</b>	
<b>Titre</b>	Mètre à jour la quantité des produits en stock.
<b>But</b>	Ce cas d'utilisation permet d'ajout une nouvelle quantité à la quantité déjà existe.
<b>Acteur</b>	L'utilisateur.
<b>Description des enchaînements</b>	
<b>Pré condition</b>	L'utilisateur est authentifié.
<b>Démarrage</b>	Ce cas d'utilisation commence lorsque une nouvelle quantité d'un produit entrant au stock.



<b>Enchaînements</b>	<ul style="list-style-type: none"> <li>a. L'utilisateur demande d'accéder à l'interface des produits.</li> <li>b. Le system affiche la liste des produits.</li> <li>c. L'utilisateur remplir le champ de nouvelle quantité et choisi le produit puis valider.</li> <li>d. Le système sauvegarde la mise à jour de la table produit et affiche un message de confirmation.</li> </ul>
<b>Exceptions</b>	Aucun.

<b>Cas d'utilisation: Etablir un bon de livraison</b>	
<b>Sommaire d'identification</b>	
<b>Titre</b>	Saisir le bon de livraison.
<b>But</b>	Ce cas permet l'utilisateur à effectuer un bon de livraison d'une commande est déjà existe.
<b>Acteur</b>	L'utilisateur.
<b>Description des enchaînements</b>	
<b>Pré condition</b>	L'utilisateur est authentifié.
<b>Enchaînements</b>	<ul style="list-style-type: none"> <li>a. L'utilisateur demande d'établir le bon de livraison</li> <li>b. Le système récupère les informations de liste des commandes.</li> <li>c. L'utilisateur choisi la commande qui veut livrer et demande de choisi le camion et le chauffeur</li> <li>d. Le système affiche les listes des camions et des chauffeurs</li> <li>e. L'utilisateur choisi le camion et le chauffeur puis valider</li> <li>f. Le système sauvegarder sur la table livraison et conduire.</li> </ul>
<b>Exceptions</b>	Aucune.

<b>Cas d'utilisation: Facturer un bon de livraison</b>	
<b>Sommaire d'identification</b>	
<b>Titre</b>	Facturation.
<b>But</b>	Faire l'établissement de la facture.

<b>Acteur</b>	L'utilisateur.
<b>Description des enchaînements</b>	
<b>Pré condition</b>	L'utilisateur est authentifié.
<b>Démarrage</b>	Ce cas d'utilisation commence lorsque le chauffeur demande la facture .
<b>Enchaînements</b>	<ul style="list-style-type: none"> <li>a. L'utilisateur demande l'établissement d'une facture.</li> <li>b. Le système affiche l'interface de liste des bons de livraisons.</li> <li>c. L'utilisateur sélectionner un bon et demande de le facturer.</li> <li>d. Le système affiche le formulaire de la facture.</li> <li>e. L'utilisateur demande d'imprimer la facture</li> <li>f. Le système imprimer la facture et mètre a jour la table facture.</li> </ul>
<b>Exceptions</b>	Aucune.

<b>Cas d'utilisation: Gérer les camions</b>	
<b>Sommaire d'identification</b>	
<b>Titre</b>	Mètre à jour la table camion de la base de données
<b>But</b>	Ce cas d'utilisation permet d'ajoute un nouveau camion, de le modifier ou de le supprimer.
<b>Acteur</b>	L'utilisateur.
<b>Description des enchaînements</b>	
<b>Pré condition</b>	L'utilisateur est authentifié.
<b>Enchaînements</b>	<ul style="list-style-type: none"> <li>a. L'utilisateur demande d'accéder à l'interface de gestion de camion.</li> <li>b. Le système affiche l'interface de gestion de camion.</li> <li>c. L'utilisateur remplir les champs (cas de saisir) ou sélectionner un camion depuis la table des camions (cas modifier ou supprimer) puis valide l'opération.</li> <li>d. Le système sauvegarde la mise à jour des camions dans la base de données puis, affiche un message de confirmation.</li> </ul>
<b>Exceptions</b>	Un message d'erreur est affiché sur l'écran si le matricule du camion est déjà existe dans la base.

Cas d'utilisation: <i>Gérer les chauffeurs</i>	
Sommaire d'identification	
<b>Titre</b>	Mètre à jour la table chauffeur de la base de données
<b>But</b>	Ce cas d'utilisation permet d'ajouter un nouveau chauffeur, de le modifier ou de le supprimer.
<b>Acteur</b>	L'utilisateur.
Description des enchaînements	
<b>Pré condition</b>	L'utilisateur est authentifié.
<b>Enchaînements</b>	<ul style="list-style-type: none"> <li>e. L'utilisateur demande d'accéder à l'interface de gestion de chauffeur.</li> <li>f. Le système affiche l'interface de gestion de chauffeur.</li> <li>g. L'utilisateur remplir les champs (cas de saisir) ou sélectionner un camion depuis la table des chauffeurs (cas modifier ou supprimer) puis valide l'opération.</li> <li>a. Le système sauvegarde la mise à jour des chauffeurs dans la base de données puis, affiche un message de confirmation.</li> </ul>
<b>Exceptions</b>	Un message d'erreur est affiché sur l'écran si le nom chauffeur est déjà existe dans la base.

#### 4.4. Identification des classes candidates

On va préparer maintenant la modélisation orientée objet par l'identification des classes candidat qui aidant à trouver les classes principales du futur modèle statique d'analyse.

- |                |                |
|----------------|----------------|
| 1. Utilisateur | 8. Get_valeurs |
| 2. Client      | 9. Réception   |
| 3. Commande    | 10. Conduire   |
| 4. Produit     | 11. Servir     |
| 5. Livraison   |                |
| 6. Camion      |                |
| 7. Chauffeur   |                |

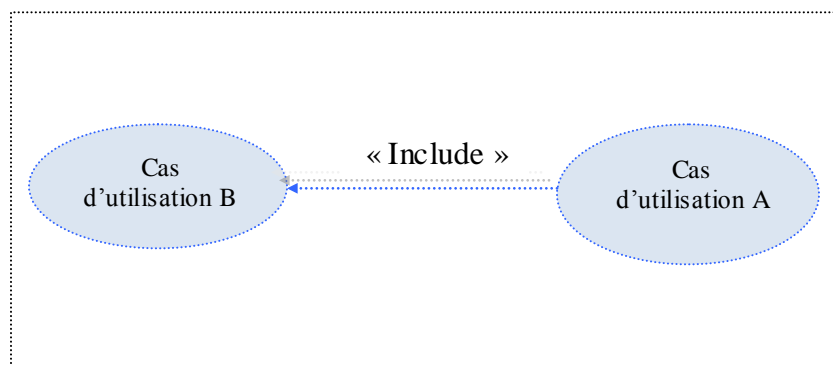
**Tableau 2.4 :** Liste des Classes Candidat du système

## 5. Relations entre cas d'utilisation

Afin d'optimiser la formalisation des besoins en ayant recours notamment à la réutilisation de cas d'utilisation, trois peuvent être décrites entre cas d'utilisation : une relation d'inclusion « include », une relation « extend » et une relation de généralisation [M2] .

### a) Relation d'inclusion (include)

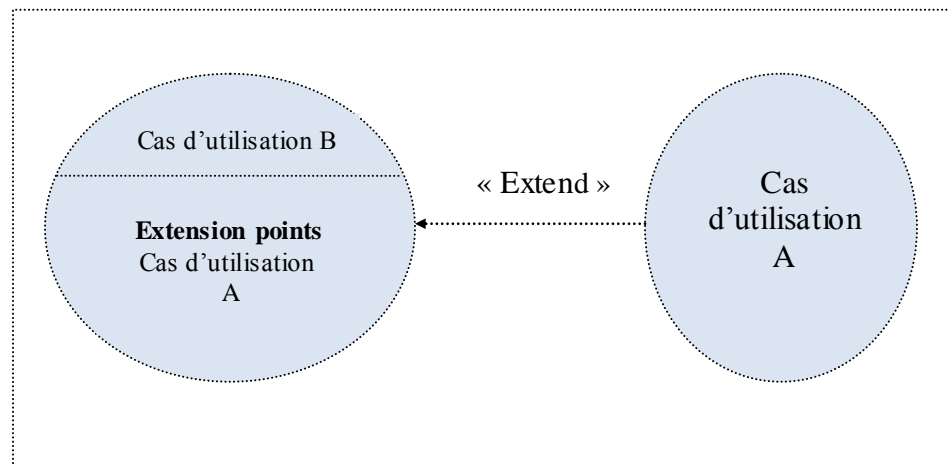
Une relation d'inclusion d'un cas d'utilisation A par rapport à un cas d'utilisation B signifie qu'une instance de A contient le comportement dans B.

**Figure 2.3 :** Représentation graphique de la relation d'inclusion.

### b) Relation d'extension (extend)

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

- Le caractère optionnel de l'extension dans le déroulement du cas d'utilisation standard (A).
- La mention explicite de l'extension dans le cas d'utilisation standard.



**Figure 2.4 :** Représentation graphique de la relation d'extension.

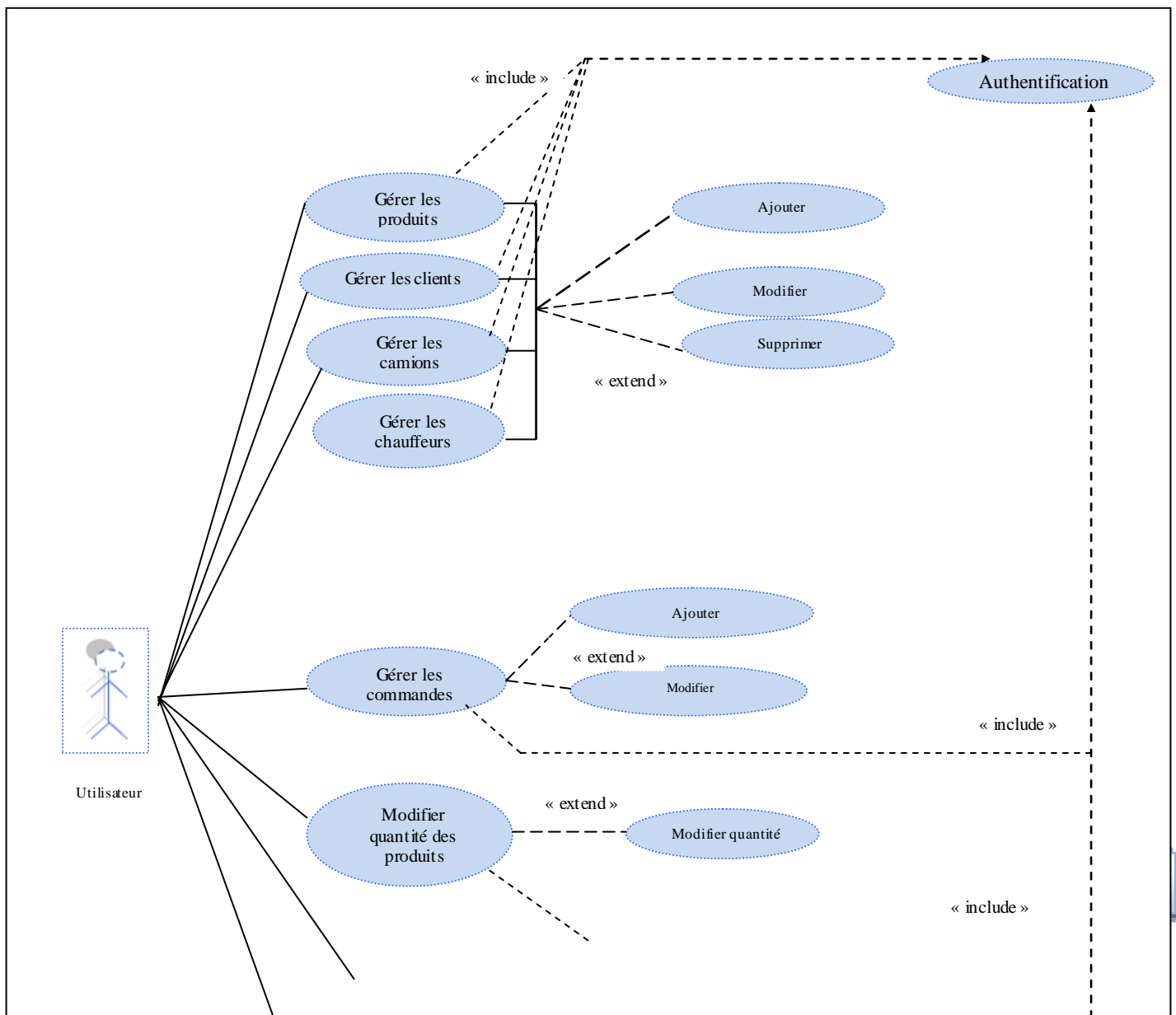
### c) Relation de généralisation

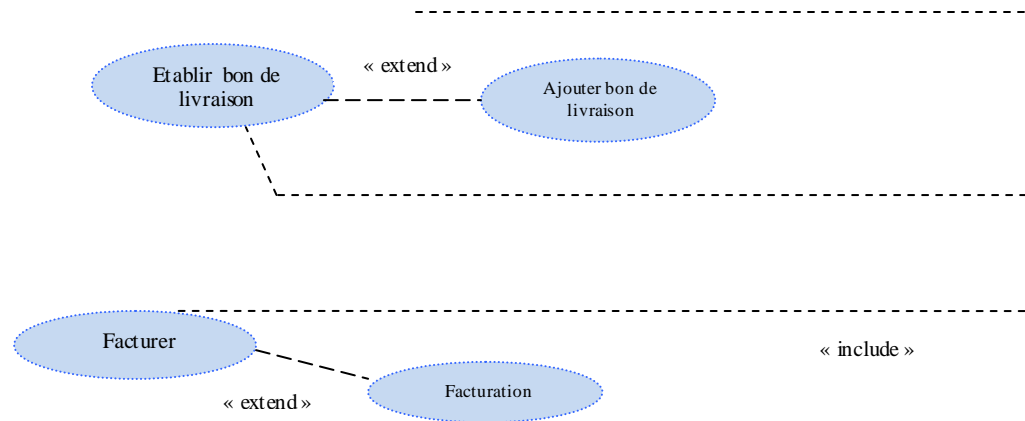
Une génération de cas d'utilisation est utilisée pour exprimer plusieurs variantes d'un cas d'utilisation général.

Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous-système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit.

Il scinde la fonctionnalité du système en unifiés cohérents, les cas d'utilisation ayant un sens les acteurs. Les cas d'utilisation permettent d'exprimer le besoin des utilisateurs d'un système [M2].

Dans ce qui suit, on décrit le diagramme détaillée des cas d'utilisation associées au système réaliser.





**Figure 2.5 :** Diagramme de cas d'utilisation associé à l'utilisateur.

## 6. Conclusion

La capture des besoins est une étape qui travaille d'un côté pour compléter les recueils des besoins opérés pendant la phase d'étude préliminaire, et d'un autre côté, elle donne une première vue pour la prochaine étape concernant l'analyse, afin d'identifier les classes candidats du modèle statique qui présentent une des approches orienté objet.

# Chapitre 03

## *L'Analyse et Conception*

- *Introduction*
- *Analyse*
- *Conception*
- *Conclusion*



## 1. Introduction

L'analyse et la conception sont souvent fortement associées. L'analyse est le processus pour lequel, en partant de l'examen des besoins, produit un modèle conceptuel du système à développer, pour cela nous allons utiliser les diagrammes de séquence. Tandis que la conception traite les résultats produits durant l'analyse en leur appliquant une architecture, dans notre cas nous allons utiliser le diagramme de classe.

## 2. Analyse

L'étape d'analyse permet une formalisation du système à développer en réponse à l'expression des besoins formulés par les utilisateurs à l'analyse des besoins. Cette phase se concrétise par l'élaboration de tous les diagrammes donnant une représentation du système statique (diagramme de classe participantes), que dynamique (diagramme de séquence, de collaboration, état transition,...) [L2].

### 2.1. Développement du modèle statique

Le développement du modèle statique constitue la deuxième activité de l'étape d'analyse.

#### 2.1.1. Affiner les classes, Association, classe Association et leurs attributs

##### ✓ Affiner les classes :

Les classes identifiées lors de l'étude des cas d'utilisation, puis réparties dans les catégories, sont simplement des classes candidates pour l'analyse objet. Il convient désormais de les examiner de manière détaillée, d'en éliminer certaines, ou au contraire d'en ajouter d'autres.

##### ✓ Affiner les associations :

Affiner les associations convient de les valider, les préciser, en éliminer, et en ajouter. Il s'agit d'une activité itérative qui consiste également à utiliser deux notions complémentaires fournies par UML : l'agrégation et la composition, qui sera complétée grâce à l'identification des attributs.

##### ✓ Ajouter les attributs :

Un attribut est une propriété nommée d'une classe qui décrit un domaine de valeurs possible partagé par tous les Objets de la classe. Dans un modèle d'analyse, nous conservons uniquement comme attributs les propriétés simples des classes que le système doit mémoriser et utiliser.

##### ✓ Ajouter les opérations :

Une opération représente un service, traitement qui peut être demandé à n'importe quel objet de la classe. Une opération est l'abstraction de ce qu'on peut réaliser sur un objet, et elle est partagée par tous les objets de la classe.

Après l'affinage des classes, des associations et l'ajout des attributs, nous obtenons le diagramme de classes qu'on aura présenté dans le suivant chapitre.

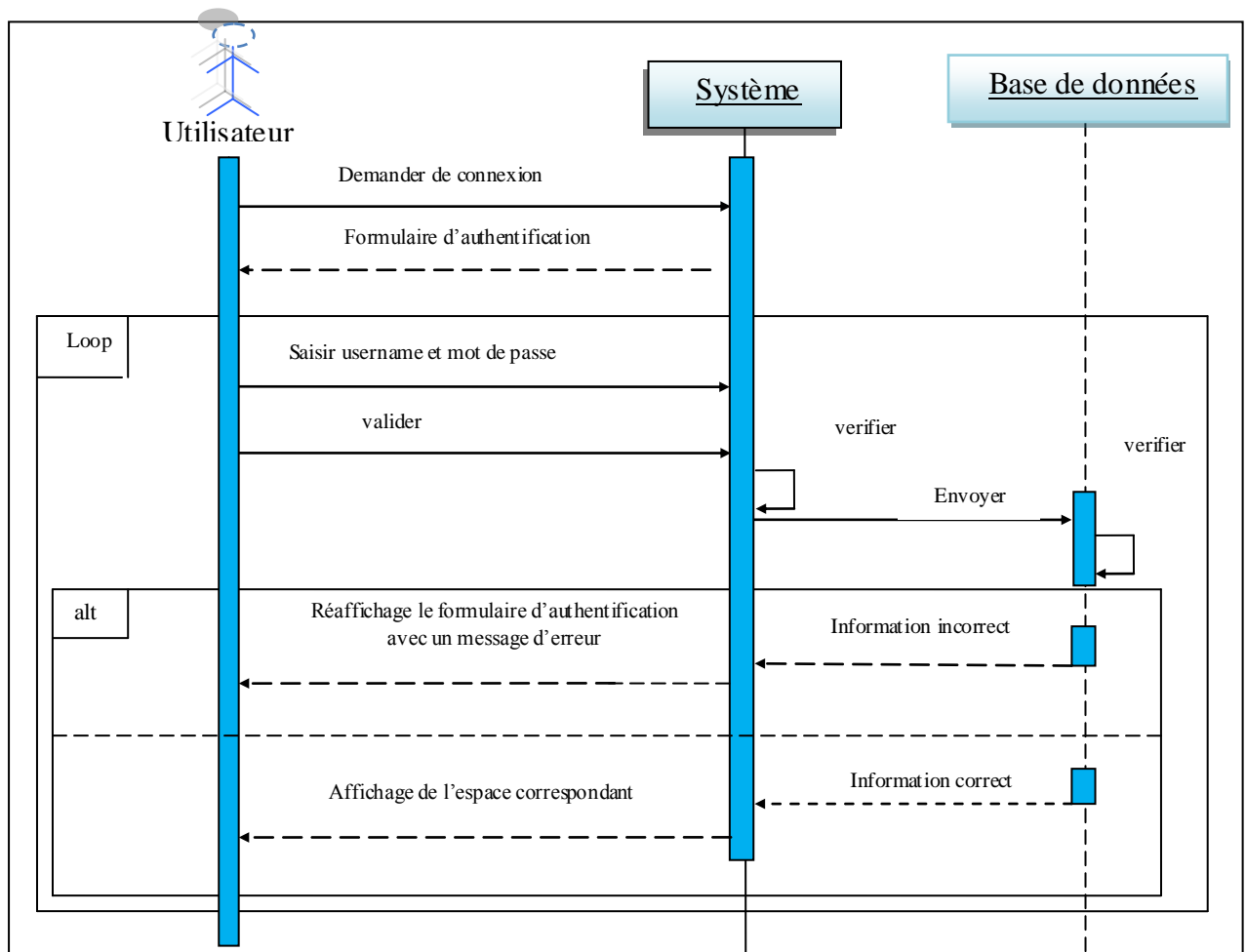
## 2.2. Développement du modèle dynamique

Le développement du modèle dynamique va nous permettre d'illustrer l'utilisation des concepts dynamiques d'UML et des diagrammes associés en phase d'analyse. Dans cette étape, nous verrons comment décrire des scénarios mettant en jeu un ensemble d'objets échangeant des messages. Pour ce là nous avons choisi le diagramme de séquence qui met l'accent sur la chronologie des messages.

- Un scénario décrit une exécution particulière d'un cas d'utilisation du début à la fin. Il correspond à une sélection d'enchaînement d'un cas d'utilisation. [L1]

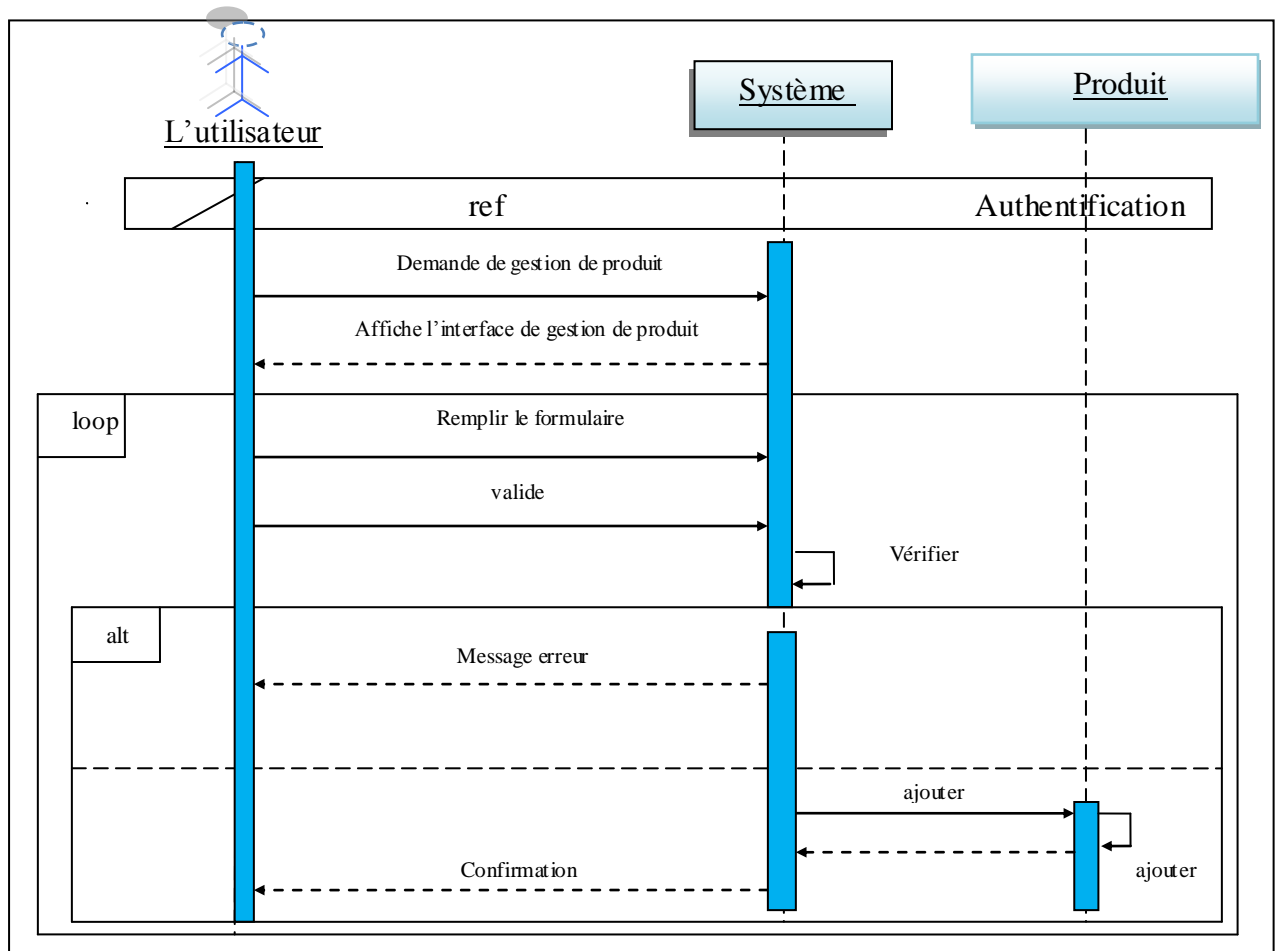
### 2.2.1 Les diagrammes de séquence pour le scénario

- A chaque authentification, l'utilisateur doit saisir son nom d'utilisateur et son mot de passe, d'où l'usage de l'opérateur "loop". L'utilisateur valide sa demande de connexion, deux cas peuvent se présenter : si les informations saisis sont correctes, le système affiche l'espace correspondant de l'utilisateur, si sont incorrectes, le système réaffiche le formulaire d'authentification avec un message d'erreur, ci qui justifie l'utilisation de l'opérateur "alt" comme le scénario ci-dessous :



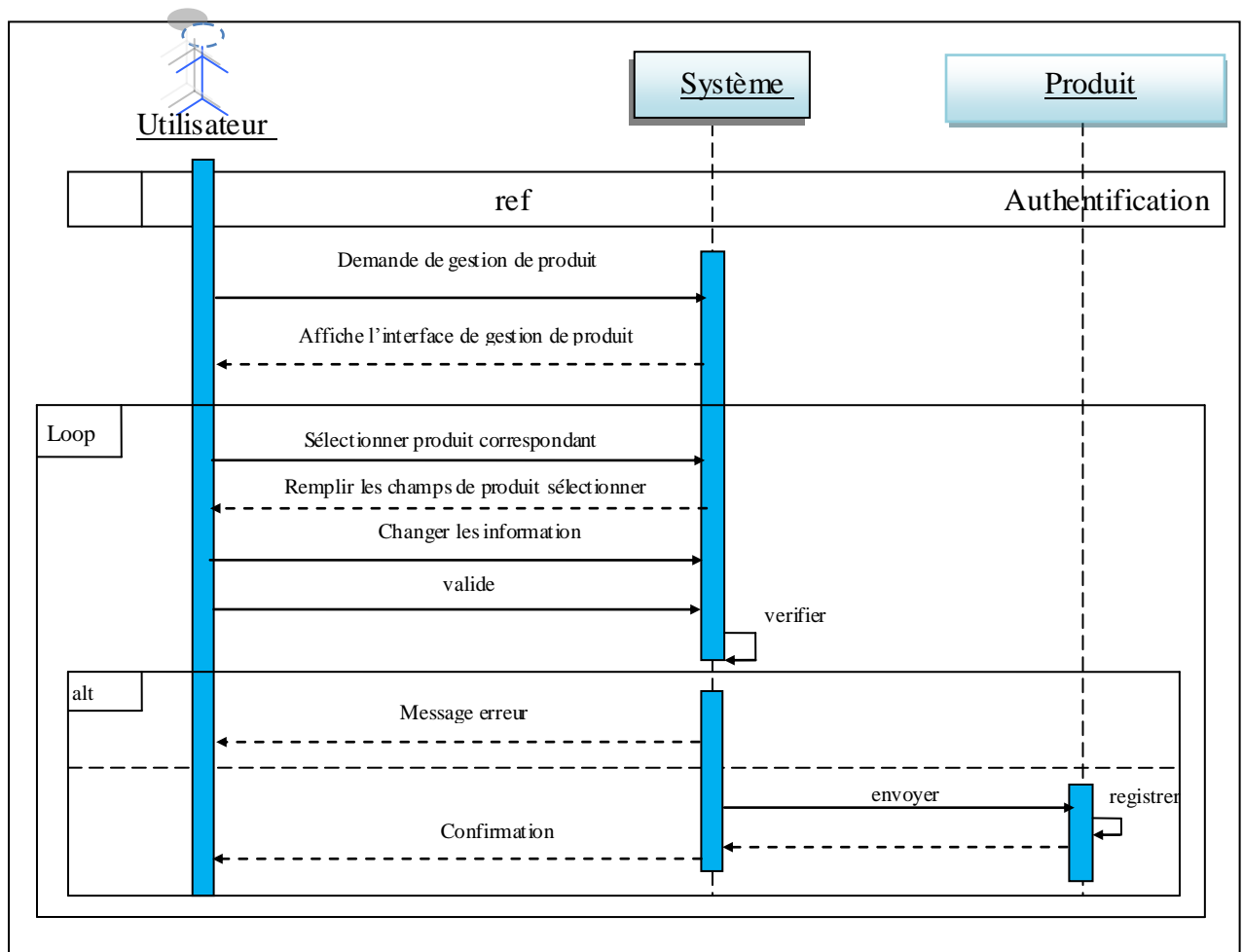
**Figure 3.1 : Le diagramme de séquence pour le scénario "Authentication"**

- Si l'utilisateur veut saisir un produit il va juste saisir les informations de produit, le système va vérifier les informations et répondre soit par un message d'erreur ou bien par un message de confirmation l'ajout.



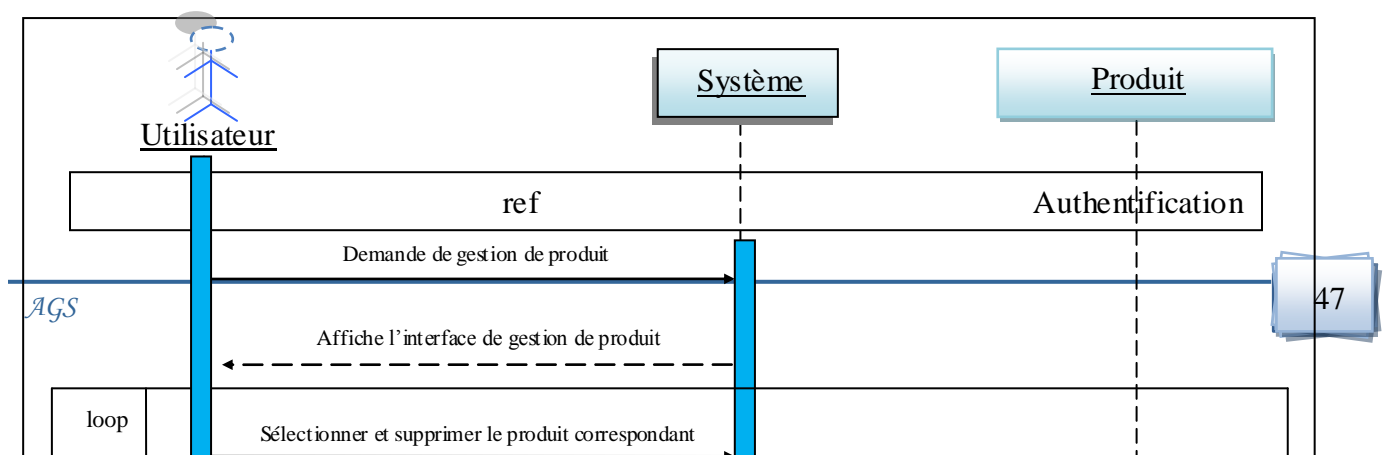
**Figure 3.2 : Le diagramme de séquence pour le scénario Ajouter un produit**

- Si l'utilisateur veut modifier un produit il va juste choisir le produit correspondant, le système va réaffiche les informations du produit sélectionner sur les champs et affiche un message pour confirmer la modification, après l'utilisateur peut changer et confirmer.



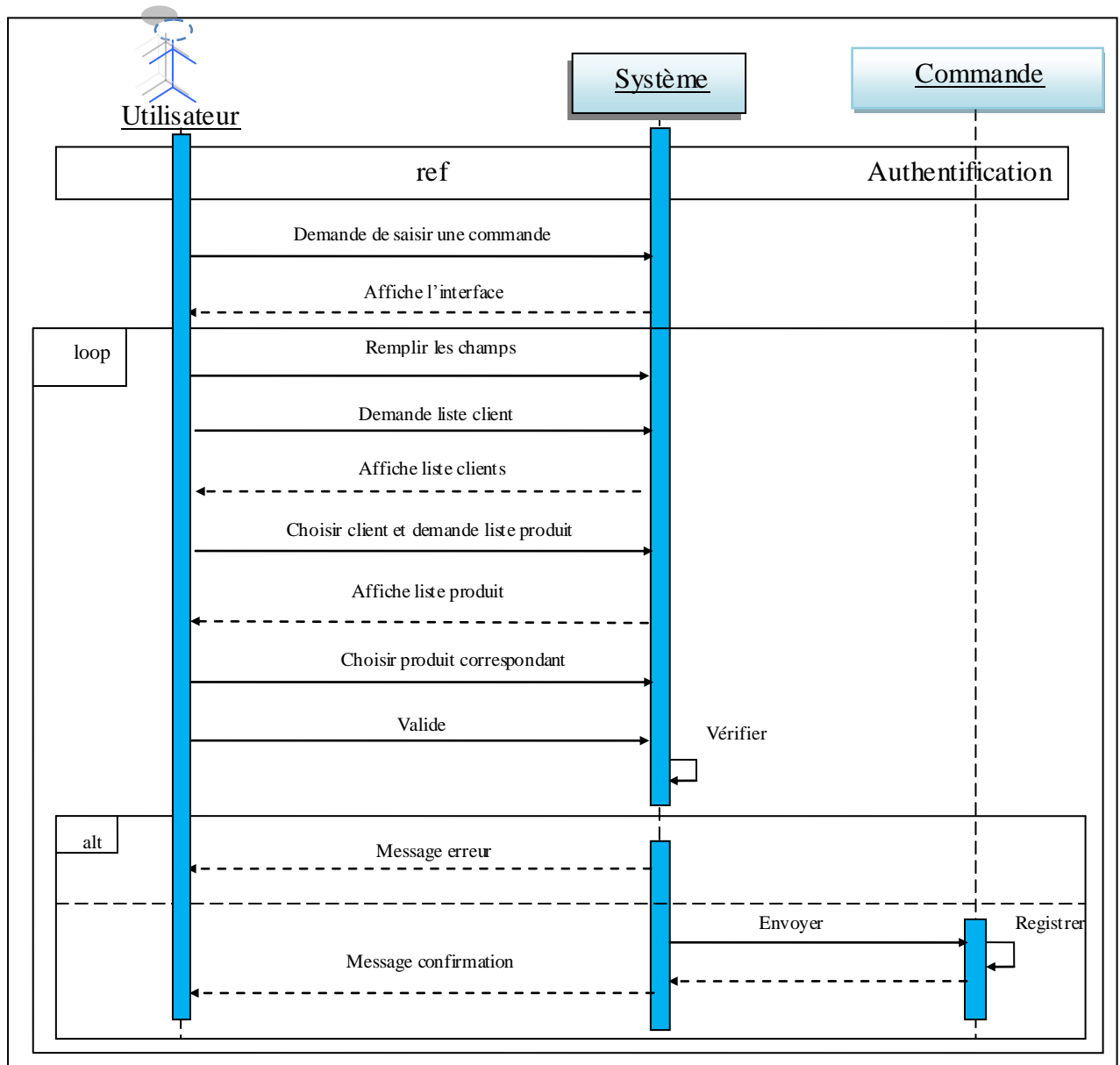
**Figure 3.3 : Le diagramme de séquence pour le scénario Modifier un produit**

- Si l'utilisateur veut supprimer un produit, il va choisir le produit et le supprimer, le système répond par un message de confirmation ou annuler la suppression.



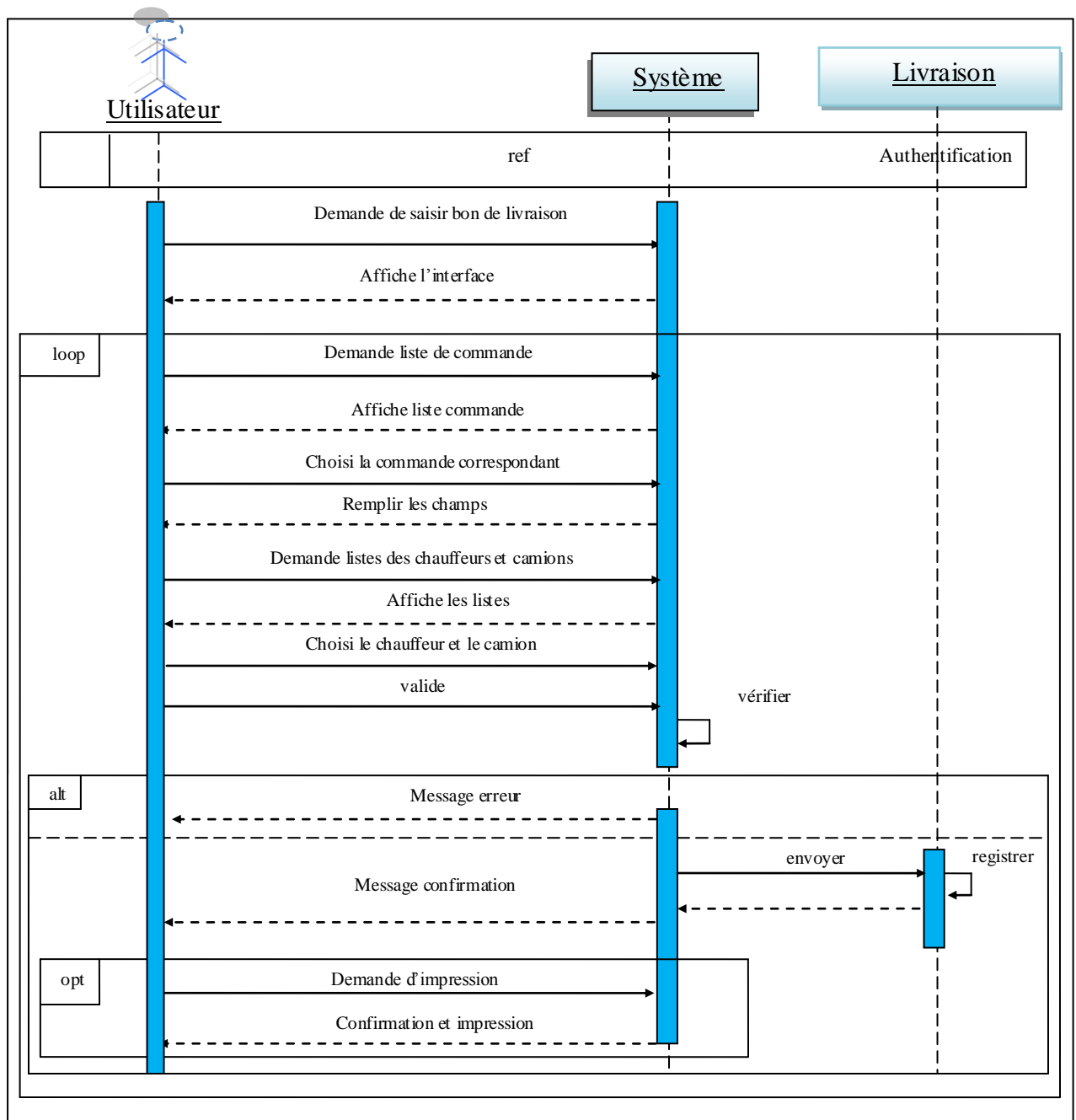
**Figure 3.4 : Le diagramme de séquence pour le scénario Supprimer un produit**

**Remarque :** le diagramme des cas d'utilisations "*ajouter client, camion et chauffeur*", "*modifier client, camion et chauffeur*", et "*supprimer client, camion et chauffeur*" et le même que celui de cas d'utilisation "*ajouter un produit*", "*modifier un produit*" et "*supprimer un produit*".



**Figure 3.5 : Le diagramme de séquence pour le scénario Saisir une commande**

**Remarque :** le diagramme de cas d'utilisation “**modifier une commande**” est le même que celui de cas d'utilisation “**modifier produit**”.

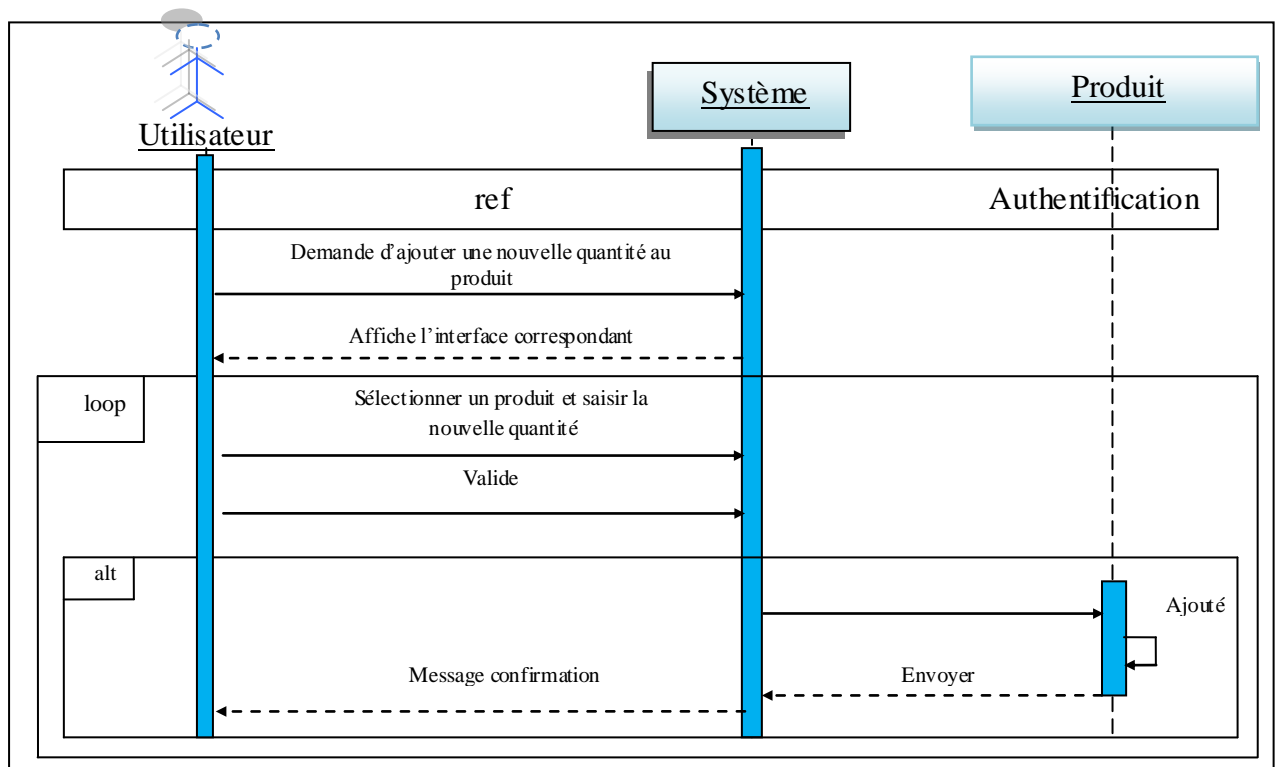


**Figure 3.6 : Le diagramme de séquence pour le scénario Etablir un bon de livraison**





Figure 3.7 : Le diagramme de séquence pour facturer un bon de livraison



### 3. Conception

La conception consiste à modéliser une solution qui résout le problème modélisé dans la phase d'analyse, elle consiste aussi à traduire les besoins en spécifiant comment l'application pourra les satisfaire avant de procéder à sa réalisation avec des outils de développement appropriés.

#### 3.1. Présentation de diagramme de classe

Le diagramme de classe constitue l'un des pivots essentiels de modélisation avec UML. En effet, ce diagramme permet de donner la représentation statique du système à développer. Cette représentation est centrée sur les concepts de classe et d'association. Chaque classe se décrit par les données et les traitements dont elle est responsable pour elle-même et vis-à-vis des autres classes [M2].

La description du diagramme de classe est fondée sur :

- Le concept d'objet.
- Le concept de classe comprenant les attributs et les opérations.
- Les différents types d'association entre classe.

##### 3.1.1. Principes et définition de base

Dans ce qui suit, nous définissons les concepts utilisés dans un diagramme de classe :

- Objet** : Un objet est un concept, une abstraction ou une chose qui a un sens dans le contexte du système à modéliser. Chaque objet a une identité et peut être distingué des autres sans considérer a priori les valeurs de ses propriétés.
- Classe** : Une classe décrit un groupe d'objets ayant les mêmes propriétés (attributs), un même comportement (opérations), et une sémantique commune (domaine de définition).
- Attribut** : Un attribut est une propriété élémentaire d'une classe. Pour chaque objet d'une classe, l'attribut prend une valeur (sauf cas d'attributs multi valeurs).
- Opération** : Une opération est une fonction applicable aux objets d'une classe. Une opération permet de décrire le comportement d'un objet.
- Association** : Une association entre classes représente les liens qui existent entre les instances de ces classes. Chaque association peut être identifiée par son nom.
- Multiplicité** : La multiplicité indique un domaine de valeurs pour préciser le nombre d'instance d'une classe vis-à-vis d'une autre classe pour une association donnée.
- Navigabilité** : la navigabilité si l'association de manière unidirectionnelle ou bidirectionnelle, elle est matérialisée par une ou deux extrémités fléchées. Le non navigabilité se représente par "X".

- h) Agrégation :** L'agrégation est une association qui permet se représenter un lien de type "ensemble" comprenant des "éléments". Il s'agit d'une relation entre une classe représentant le niveau "ensemble" et 1 à n classes de niveau "éléments". L'agrégation représente un lien structurel entre une classe et une ou plusieurs autres classes.
- i) Composition :** La composition est une relation d'agrégation dans laquelle il existe une contrainte de durée de vie entre la classe "composé" implique la suppression de la ou des classes "composant".
- j) Généralisation, spécialisation :** La généralisation est la relation entre une classe et deux autres classes ou plus partageant un sous-ensemble commun d'attributs et/ou d'opération. La classe qui est affinée s'appelle super classe, les classes partir de classes s'appelle la généralisation. Inversement la spécialisation consiste à créer des sous classes à partir d'une classe.
- k) Classe abstraite :** Une classe abstraite est une classe qui n'a pas d'instance directe mais dont les classes descendantes ont des instances. Dans une relation d'héritage, la super classe est par définition une classe abstraite.

## 3.2. Conception détaillée

La conception détaillée s'appuie sur les catégories de conception organisées à la fois suivant les Frameworks techniques et les regroupements propres au métier. Elle consiste à construire et à documenter précisément les classes, les interfaces, les tables et les méthodes qui constituent le codage de la solution. [L1].

### 3.2.1. Diagramme de classe

### **3.2.2. Le modèle relationnel de données**

Le concepteur d'une BD relationnelle doit élaborer ce qu'il est convenu d'appeler le schéma relationnel de la base de données. Cette activité consiste à définir toutes les relations normalisées de la BD et les domaines de leurs attributs. Théoriquement cela consisterait à décrire par intention

chaque relation, définir les domaines de chaque attribut de la relation. Les relations du schéma doivent toutes posséder les propriétés suivantes [M2] :

- Une relation a un nom distinct de toutes les autres du même schéma.
- Chaque attribut d'une relation ne peut recevoir qu'une seule valeur atomique (type de données simple).
- Chaque attribut a un nom distinct.
- Toutes les valeurs d'un attribut font parties du même domaine, même type de données et même contraintes d'intégrité.
- L'ordre des attributs n'a pas importance dans une relation.

### 3.2.3. Passage du modèle objet au modèle relationnel

Dans cette partie nous allons présenter quatre règles permettant de d'écrire un schéma logique dans le modèle objet relationnel à partir d'un diagramme de classe.

#### - Transformation des classes

**R1** : chaque classe du diagramme UML devient une relation, il faut choisir un attribut de la classe pouvant jouer le rôle d'identifiant.

#### - Transformation des associations

Nous distinguons trois familles d'association :

##### – Association 1-\* :

**R2** : il faut ajouter un attribut de type clé étrangère dans la relation fils de l'association. L'attribut porte le nom de la clé primaire de la relation père de l'association.

##### – Association \*-\* :

**R3** : la classe association devient une relation. La clé primaire de cette relation est la concaténation des identifiants des classes connectées à l'association.

##### – Association 1-1 :

**R4** : Il faut ajouter un attribut de type clés étrangères dans la relation dérivée de la classe ayant la multiplicité minimale égale à un. L'attribut porte le nom de la clé primaire de la relation dérivée de la classe connectée à l'association. si les deux multiplicités sont à un, il est préférable de fusionner les deux classes en une seule.

### 3.2.4. Le modèle relationnel

Par l'application des quatre règles précédentes sur le diagramme de classe nous pouvons avoir le modèle objet relationnel suivant :



## **4. Conclusion**

Ce chapitre a été consacré à la conception de notre application, la première section nous a permis de faire une formalisation du système à développer en repense à l'expression des besoins formulés par les utilisateurs. Le modèle structurel d'analyse nécessite un travail d'analyse détaillée de la structure des classes. Celui-ci est considéré comme une base pour le développement du modèle statique et dynamique, et dans la deuxième section nous permettrons de donner les principales bases de notre logiciel, et de descendre dans le niveau d'abstraction et d'approcher au niveau physique (passage du modèle objet vers le modèle relationnel).





# Chapitre 04

## Réalisation

- *Introduction*
- *Modélisation du système*
- *Présentation de l'environnement de programmation*
- *Menu d'application*
- *Mise en œuvre du logiciel*
- *Conclusion*

## Introduction

Dans ce chapitre, consacré à la réalisation et la mise en œuvre de notre application nous allons présenter le langage de modélisation et les outils de développement adoptés ainsi que une aperçu général sur la phase pratique qui consiste à décrire les différentes fonctionnalités disposées lors de l'utilisation du notre logiciel.

## 2. Modélisation du système

UML( *Unified Modeling Language*), est un langage de modélisation graphique, conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en [développement logiciel](#) et en [conception orientée objet](#).

- UML n'est pas une méthode, est un langage qui permet de représenter des modèles, mais il ne définit pas le processus d'élaboration des modèles.
- UML à une notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de solution.
- UML est un langage de modélisation objet, permet d'exprimer et d'élaborer des modèles objet, indépendamment de tout langage de programmation. UML est un langage formel, définit par un méta modèle.
- UML permet de représenter un système selon différentes vues complémentaires (les diagrammes).

## 3. Présentation de l'environnement de programmation

Le logiciel que nous avons réalisé est une application Java qui sera utilisé pour géré un stock de l'entreprise de BOUHZILA.

### 3.1. Outils de développement

#### Langage de programmation

Puisque notre travail est basé sur la méthode objet, nous avons choisi " Java " comme langage de développement parce que :

1. C'est un langage orienté objet.

2. Très proche du langage C++ étant donné qu'il a quasiment la même syntaxe. Toutefois Java est plus simple que le langage C++ bien qu'il s'en inspire, car les caractéristiques critiques du langage C++ (celles qui sont à l'origine des principales erreurs) ont été supprimées. Cela comprend :

- Les pointeurs,
- La surcharge d'opérateurs,
- L'héritage multiple,
- La libération de mémoire est transparente pour l'utilisateur (il n'est plus nécessaire de créer des destructeurs),
- Une meilleure gestion des erreurs,

### Environnement de programmation

Un langage de programmation nécessite forcément un environnement dans lequel on désire développer une application. Eclipse IDE est un environnement de développement intégré libre (le terme Eclipse désigne également le projet correspondant, lancé par IBM) extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions. La spécificité d'Eclipse IDE vient du fait de son architecture totalement développée autour de la notion de

plug-in : toutes les fonctionnalités de cet atelier logiciel sont développées en tant que plug-in [Eclipse].

### Environnement de développement WampServeur

WampServer est une plateforme de développement Web de type WAMP, servant à faire fonctionner localement des scripts PHP.

WampServer n'est pas en soi un logiciel, mais environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script(PHP), ainsi qu'une administration pour les deux bases SQL : PhpMyAdmin et SQLListeManager [M2].

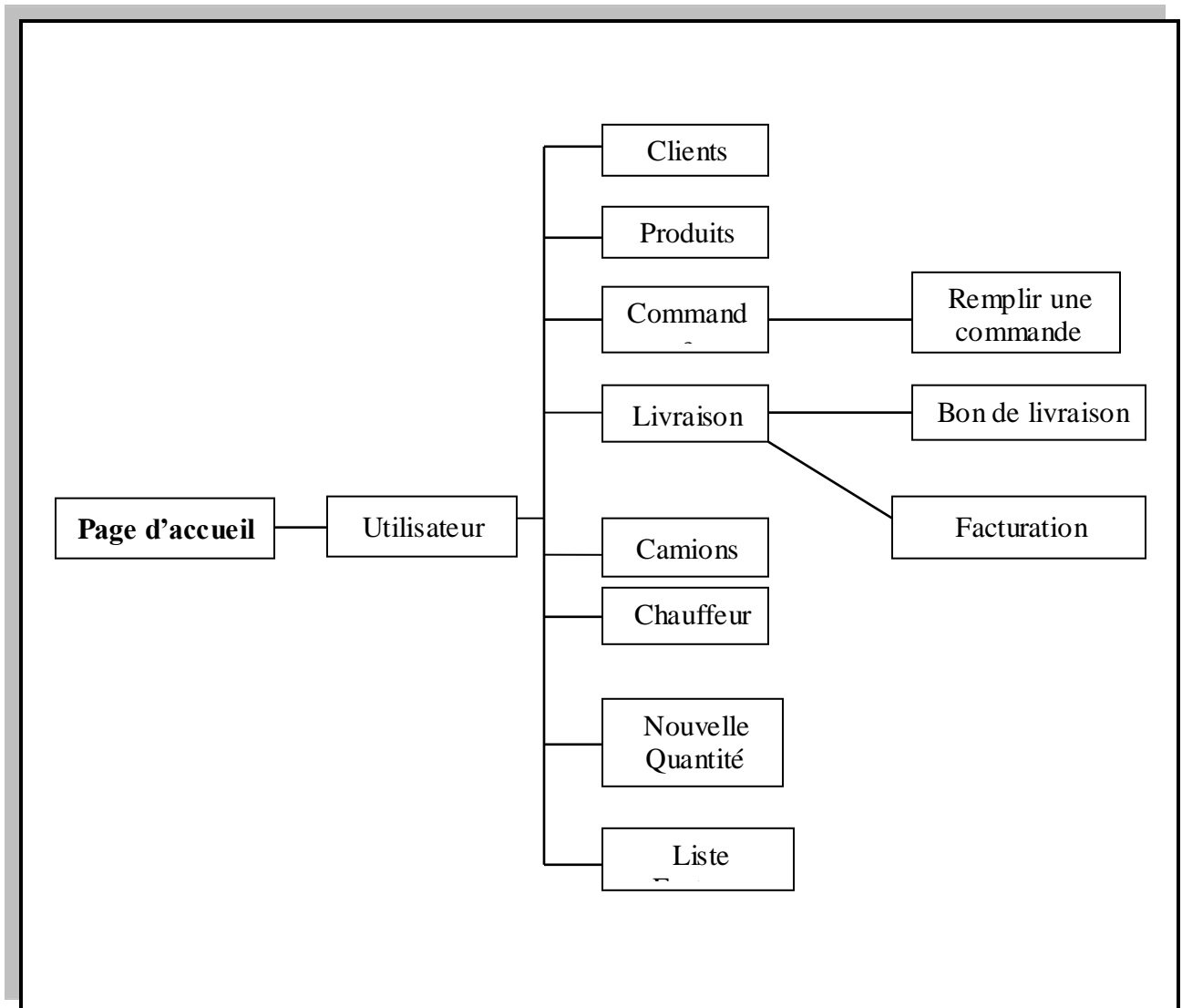
- **MySQL** (*My Structured Query Language*) est un système de gestion de base de données relationnelles SGBDR, multi-tâches, multi-utilisateurs construit autour de l'architecture client/serveur, il prend en charge les fonctionnalités de protection et de sécurité et fournit un ensemble d'interfaces de programmation. Spécialement conçu

pour la vitesse et la stabilité, il est devenu au cours de ces dernières années l'un des plus populaire pour les applications de bases de données à la fois [M2].

Le serveur MySQL contrôle l'accès aux données pour s'assurer que plusieurs utilisateurs peuvent se servir simultanément d'une même base de données pour y accéder rapidement et pour garantir que seuls les utilisateurs autorisés peuvent accéder aux données [M1].

- **PhpMyAdmin** est un logiciel libre, écrit en PHP dédié à la gestion et à l'administration de serveur MySQL sur le Word Wide Web. Les opérations les plus fréquemment utilisés sont pris en charge par l'interface utilisateur (bases de données de gestion, tables, champs, relations, les index, les utilisateurs, les permissions), PhpMyAdmin permet de faire toutes sortes d'opérations comme :
  - Créer et détruire des bases de données (à condition d'avoir les droits).
  - Consulter le contenu des tables, modifier ou supprimer certaines lignes ou les détruire.

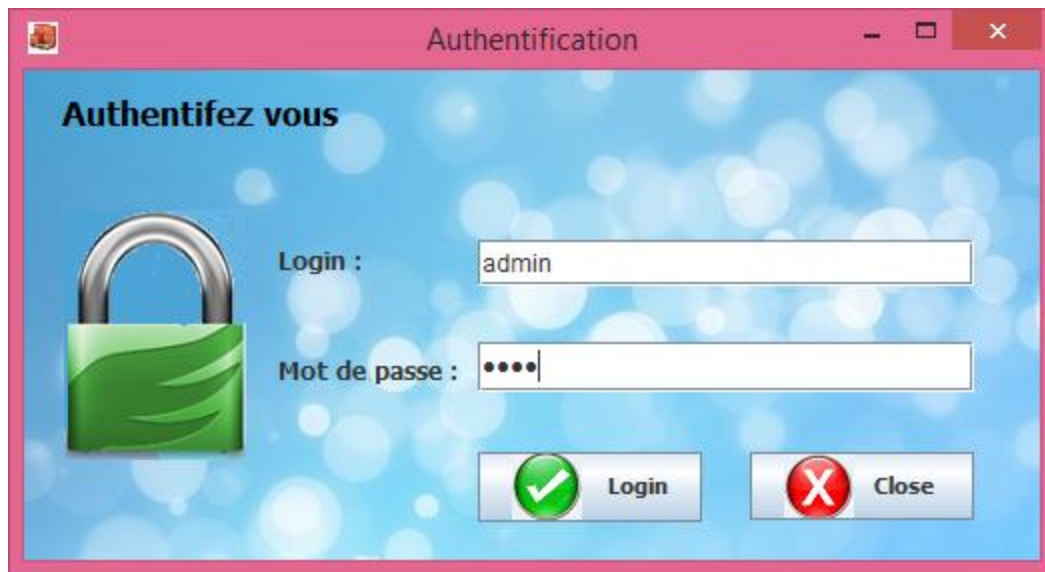
#### **4. Menu de l'application**

**Figure 4.1:** Menu de l'application

## 5. Mise en œuvre du logiciel

Nous allons maintenant entamer la phase pratique qui consiste à décrire les différentes fonctionnalités de notre application :

- ✚ Quand vous lancerez le logiciel, l'interface d'authentification s'apparaîtra sur l'écran, cette étape est pour sécuriser les données et protéger l'application, l'utilisateur doit s'identifier par son nom d'utilisation et son mot de passe pour pouvoir accéder à page d'accueil:



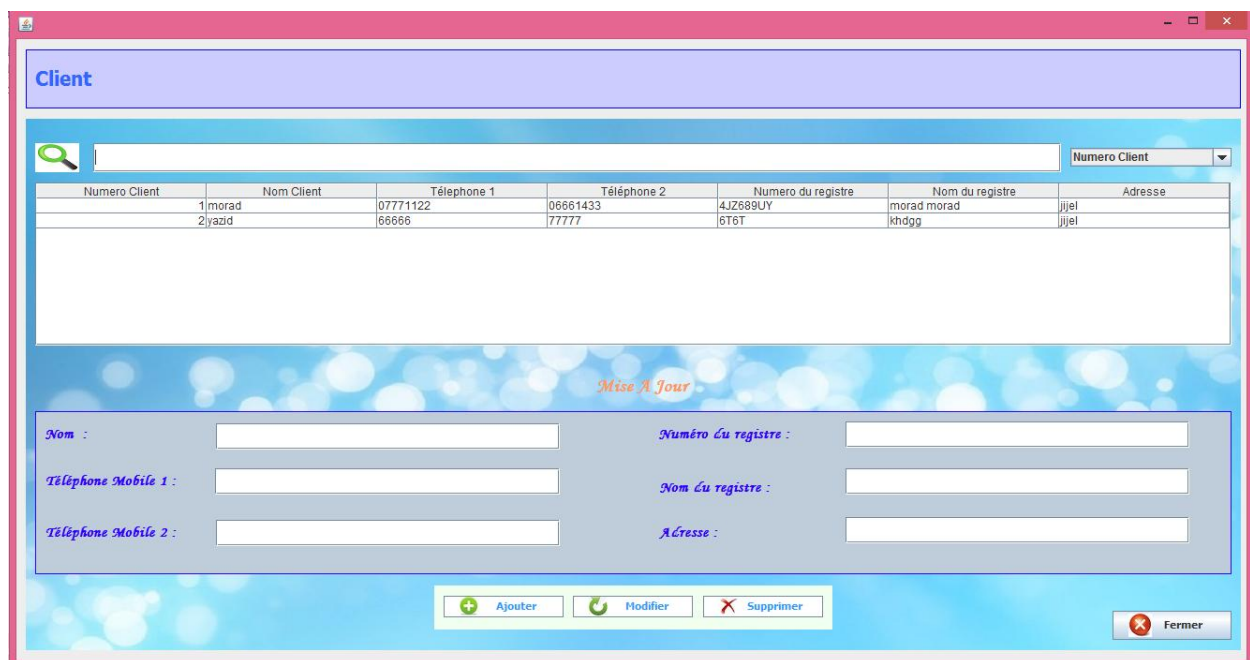
**Figure 4.2:** La fenêtre de l'authentification

- ✚ Si le système valide les mots de passe, l'interface principale sera affichée, où l'utilisateur aura les différentes fonctionnalités pour gérer le stock :



**Figure 4.3:** La fenêtre du menu principal de l'application

- Si l'utilisateur clique sur l'icône Clients, il est dirigé vers la fenêtre de gestion des clients, et lui permet de saisir, modifier ou supprimer un client, et il peut aussi lancer une recherche sur un client.



**Figure 4.4:** La fenêtre gestion client

**Remarque :** Les interfaces produits, commandes, camions, chauffeurs, livraison sont le même principe que l'interface du client.

- Si l'utilisateur choisit la saisie d'une commande, l'interface suivante s'affiche qui lui permet de saisir ou modifier les informations d'une commande :

The screenshot shows a software window titled "Commande" with a subtitle "MISE A JOUR COMMANDE". The window has a pink border and a blue background with a bokeh effect. It contains the following elements:

- Input field for "Client" with a dropdown arrow.
- Input field for "Date" with a calendar icon.
- Input field for "Produit" with a dropdown arrow.
- Input field for "Date livraison" with a calendar icon.
- Input field for "Quantité".
- Buttons for "Ajouter" (green plus icon) and "Modifier" (green circular arrow icon).
- A "Fermer" button (red X icon) in the bottom right corner.

**Figure 4.5:** l'interface du saisir une commande

- Si l'utilisateur choisit d'établir une livraison, l'interface suivante s'affiche qui lui permet d'établir un bon de livraison :



**Bon De Livraison**

Date : 11 juin 2016

Nom Client :

Nom Produit :  Quantité :

Prix Total :

**Livrer**

Camion :  Chauffeur :

Date de conduire :

**Ajouter** **Imprimer** **Fermer**

**Figure 4.6:** La fenêtre du saisir bon de livraison

- Si l'utilisateur veut facturer une livraison, il va sélectionner une livraison dans la liste et lancer l'impression de la facture.

<b>GROUPE BOUHZILA</b>		<b>ALIMENTS DE BETAILS</b> Setif													
Registre de Commerce: 43/00.0322258B99		<b>Facture</b> N° 00004 Du 23/06/2016													
Identifiant Fiscal: 09994303 10 024 37															
Art Imposition : 43030824481															
Phone 031-53-61-46 031-52-73-59		Fax 031-52-32-59													
E-mail : groupbouhzila@yahoo.fr															
Raison sociale	samah yassin	Bon de Commande													
N° de Registre	6T6T	Date Commande													
Adresse	jijel														
<table border="1" style="width: 100%; border-collapse: collapse; margin-top: 20px;"> <thead> <tr> <th style="width: 10%;">Reference</th> <th style="width: 30%;">Designation-Libelle</th> <th style="width: 10%;">Quantité</th> <th style="width: 10%;">Prix Unit</th> <th style="width: 20%;">Montant</th> <th style="width: 10%;">TVA</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00001</td> <td style="text-align: center;">Aliment Croissance</td> <td style="text-align: center;">50</td> <td style="text-align: right;">4000.00</td> <td style="text-align: right;">200000.00</td> <td style="text-align: center;">17</td> </tr> </tbody> </table>				Reference	Designation-Libelle	Quantité	Prix Unit	Montant	TVA	00001	Aliment Croissance	50	4000.00	200000.00	17
Reference	Designation-Libelle	Quantité	Prix Unit	Montant	TVA										
00001	Aliment Croissance	50	4000.00	200000.00	17										
T.V.A				34000.00											
Total				234000.00											
LE RESPONSABLE COMMERCIAL															

Figure 4.7: La facture d'une livraison

## 6. Quelques requêtes SQL

Le SQL est un langage permettant de communiquer avec une base de données. Ce langage informatique est notamment très utilisé par les développeurs web pour communiquer avec les données d'un site web.

- Mise à jour une table :

```

public void Saisir(){
    String query= "INSERT INTO `camion` ( `Mat_cam`, `Marq_cam`, `Cap_cam`) VALUES ('"+matricule+"','"+marque+"','"+capacite+"')";
    executeSQLQuery(query,"ajouter" );
}

public void Modifier(){
    String query = "UPDATE `camion` SET `Mat_cam`='"+matricule+"',`Marq_cam`='"+marque+"',`Cap_cam`='"+capacite+"' WHERE `Mat_cam`= '"+ n
    executeSQLQuery(query,"modifier");
}

public void Supprimer(){
    String query= "DELETE FROM `camion` WHERE `Mat_cam`= '"+matricule;
    executeSQLQuery(query,"supprimer");
}

```

- **La sélection :**

```

int i = 0;

String query="select * from `command` ";
Statement st;
ResultSet rs2,rs3,rs4;
    st = con.createStatement();
    ResultSet rs=st.executeQuery(query);

while(rs.next()){
    String a;
    i=rs.getInt("Num_cmd");
    a=rs.getString("Date_cmd");
    d_l=rs.getString("Date_liv");

    String query2="SELECT Nom_client FROM `client` WHERE Id_client in (SELECT Id_client FROM `command` where Num_cmd = '"+i+"')";
    Statement st2;st2 = con.createStatement(); rs2=st2.executeQuery(query2);
    String a3 = null;
    if(rs2.next()){
        a3=rs2.getString("Nom_client");
    }

    String query3="select Nom_prod from `produit` where Code_prod in (select Code_prod from `contenir` where Num_cmd = '"+i+"') ";
    Statement st3;st3 = con.createStatement(); rs3=st3.executeQuery(query3);
    String a4=null;
}

```

## 7. Conclusion

Dans ce chapitre nous avons présenté l'étape de la réalisation qui contient la description du logiciel ainsi que sa mise en oeuvre, les interfaces principales et ses fonctionnalités.

Durant l'implémentation des différentes fonctionnalités et des différentes fenêtres, nous avons pris soin de créer une application d'un système d'information à des interfaces graphiques simple et interactive.

# Conclusion général

Notre projet « *AGS* » consiste à réaliser un système d'information automatique pour la gestion de stock de l'entreprise BOUHZILA. L'objectif principal visé par ce projet est de faciliter et d'accélérer toutes la gestion de toutes les tâches de l'entreprise.

La contribution de ce modeste travail est présentée en l'application de toutes les étapes de la méthodologie UML dans l'étude, et le langage de programmation JAVA.

Nous avons réalisé un système qui permet :

- Une grande souplesse pour l'organisation.
- La représentation des informations d'une manière fiable, efficace et rapide, surtout dans l'établissement des différents documents.
- La possibilité d'établir les statistiques et de donner des résultats exacts et précis en permettant un grand gain de temps.

Ainsi, le stage que nous avons effectué au sein du l'entreprise a été pour nous une expérience très enrichissante et très intéressante, dans la mesure où il nous a permis de mettre en pratique et de développer les connaissances que nous avons acquises tout au long de notre cursus universitaire.

Enfin, nous espérons que nous avons pu à construire une base de départ pour les améliorations futures.

# Bibliographies et Webographies

## Les ouvrages

- [L1]: Pascal. Roque, Frank. Eyollée « UML en action de l'analyse à la conception en java ».
- [L2]: Joseph Gabay, Merise et UML pour la modélisation des systèmes d'information [dunod, 2000].
- [L3]: Guy Pujolle, «Les réseaux, 2012 ».

## Les mémoires

- [M 1] : AMMOUR F. Conception et réalisation d'une application sous-réseaux local pour suivi des patients au niveaux de l'établissement public de hospitalier de Jijel . Mémoire master de fin d'étude-*Promotion 2015*-Université de Bejaïa.
- [M 2] : DMIMI K et AMIROUCH A. Conception et réalisation d'une application de gestion stock, Cas d'étude MAG\_SOUMMAM. Mémoire de fin d'étude-*Promotion 2014* Université de Bejaïa.
- [M 3] : LAEBANI Y. Conception et réalisation d'un système d'information sous réseau local pour la gestion du service de transport terrestre et sanctions administratives au niveau de direction de transport de la wilaya de Jijel. Mémoire de fin d'étude-*Promotion 2010*-Université de Jijel.
- [M 4] : MANIKH O et ZEIMACH N. Conception et réalisation d'un système d'information sous réseau Intranet pour la gestion de scolarité de la Faculté des Sciences Exactes. Mémoire de fin d'étude-*Promotion 2009*- Université Abderahmane Mira-Bejaïa.

## Les sites web

- [W 1] : [www.techno.science.net/onglet=glossaire & definition](http://www.techno.science.net/onglet=glossaire%20&definition). Date consultation: 12/05/2016 à 03/06/2016.
- [W2]: <http://www.memoireonline.com/08/09/2577/Conception-et-realisation-dune-application-de-suivi-de-patients-dans-un-etablissement-hospitalier.html>. Date consultation:



## Résumé

*L'objectif de ce travail est de mettre en place un système d'information automatisé de base de données pour la gestion de stock de l'entreprise BOUHZILA à Jijel.*

*Notre produit logiciel permet de gérer plusieurs tâches de stock: les clients, les produits, les commandes, la livraison, les camions et les chauffeurs, faciliter les calculs et fournir un outil de qualité fiable et efficace.*

*Pour réaliser cette application, nous avons utilisé le langage UML pour la modélisation objet, JAVA comme un langage de programmation et MYSQL comme SGBD.*

**Mots clés:** *AGS, SQL, UML, Eclipse, Java .*

## Abstract

*The objective of this work is to establish an operational automated information system database for inventory management of the company BOUHZILA in JIJEL.*

*Our software product allow to manage many things of the stock: customers, products, commands, deliveries, camions and drivers, facilitates the calculate and provide a reliable and effective quality tool.*

*To realize this application, We have used UML for object modeling, JAVA as a programming language and MYSQL as DBMS.*

**Key words:** *AGS, SQL, UML, Eclipse, Java.*



