

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOULOU D MAMMERI DE TIZI OUZOU

FACULTE DE GENIE ELECTRIQUE ET INFORMATIQUE

DEPARTEMENT INFORMATIQUE

MEMOIRE

En vue de l'obtention du diplôme de Master 2 en informatique.

Thème

*Etude comparatifs des AGL pour la conception et la réalisation d'une application
de gestion de stock*

Cas :Eniem

Réalisé par :

Dirigé par :

TOUAT Massilva

MR : DIB Ahmed

DJERMANE Ferroudja

Promotion 2013/2014

Remerciements

Nous tenons à remercier particulièrement notre encadreur monsieur DIB Ahmed pour nous avoir fait confiance et nous avoir encouragés tout au long de ce projet. On le remercie également pour sa disponibilité , son aide, ses conseils précieux et ses critiques constructives, ses explications et suggestions pertinentes.

Nous tenons aussi à exprimer toute notre gratitude aux membres du jury pour avoir accepté d'évaluer et de juger notre travail.

Nous adressons également nos remerciements à tous les professeurs qui nous ont enseigné durant tous notre cursus universitaire.

Nous remercions aussi tout le personnel de L'ENIEM qui nous a vraiment aidé durant notre stage en particulier Monsieur Bouassem .

Nous remercions aussi tous ceux qui ont contribué de loin ou de près par leurs apports à l'accomplissement et l'aboutissement de ce projet.

DEDICACES

*Je dédie ce modeste travail à mes chères
parents ma famille, mes proches, mes
camarades de master 2, à tous mes amis
(es) et enfin à toutes les personnes ayant
contribué chacune à sa manière au bon
accomplissement de notre projet.*

ferroudja

DEDICACE

*Je dédie ce modeste travail à une personne qui est très chère à mon cœur à la personne qui à toujours était là pour moi qui m'a toujours soutenue et aidée dans tout ce que j'ai fait, qui m'a toujours encouragé à aller de l'avant, à la personne que je regrette tant qu'elle ne soit pas avec moi pour ce jour important et qui aurait été très fière de moi. Je te le dédie à toi **MALIK** dieu ait ton âme, J'aurai aimé que tu sois là pour m'aider comme tu l'as fait pour mon mémoire de licence. Ton image ne me quitte jamais je pense à toi dans tout ce que je fais et je penserai toujours à toi.*

Massilva

Résumé

Le but de cette étude est de répondre aux besoins et préoccupations exprimées par les utilisateurs du service de gestion de stocks de ***l'ENIEM***.

Ce Project nous a permis d'avoir un aperçu sur la vie professionnelle des entreprises et d'améliorer nos connaissances et nous aguerrir à la démarche ***MERISE*** qui est une méthode de conception qui nous a permis de comprendre le système d'information existant, aussi la méthode ***UML (Unified Modeling Language)*** qui par ses briques de base offre de modèle lisible et facile à maîtriser, afin d'assurer une meilleur organisation et une bonne gestion. Pour finir nous avons opté pour l'utilisation de la plate-forme ***Java netbeans 6.8*** et le produit ***MySQL serveur 5.5*** qui est un système de gestion de bases de données relationnelles qui nous a permis d'approfondir nos connaissances dans ces dernières.

Cependant, quoiqu'elle présente de diverses fonctionnalités, quelques améliorations et extensions peuvent être apportés à notre application.

Liste des Figures :

Numéro figure	Titres des figures
Figure 1	Organigramme général de l'ENIEM
Figure2	Organisation de l'unité Commerciale.
Figure 03	unité commerciale.
Figure 04	Dialogue client/serveur
Figure 05	Architecture CLIENT/SERVEUR à deux niveaux
Figure 06	Architecture CLIENT/SERVEUR à trois niveaux.
Figure 07	Architecture CLIENT/SERVEUR multi niveaux.
Figure 08	Représentation graphique de la démarche de modélisation de l'application.
Figure 09	Diagramme de contexte
Figure 10	Diagramme de cas d'utilisation relatif au <<chef de service magasin PF>>.
Figure 11	Diagramme de cas d'utilisation relatif au <<gestionnaire des expéditions >>.
Figure 12	Diagramme de cas d'utilisation relatif au <<chef de service programmation PF >>.
Figure 13	Diagramme de séquence << ajouter quantité d'un produit >>
Figure 14	Diagramme de séquence << consulter facture client >>

Figure 15	Diagramme d'activité du cas d'utilisation <<ajout produit en stock >>.
Figure 16	Diagramme d'activité du cas d'utilisation <<consulter facture produit >>
Figure 17	Diagramme d'activité du cas d'utilisation <<fichier article >>.
Figure 18	Diagramme d'activité du cas d'utilisation <<authentification de l'utilisateur >>.
Figure 19	Diagramme de classe
Figure 20	Indépendance d'un programme en java de toute plate forme.
Figure 21	capture d'écran présentant l'interface de développement netbeans 6.8
Figure 22	Présentation de la base de données.

Liste des tableaux

<i>Numéros tableaux</i>	<i>Titre des tableaux</i>
<i>N°1</i>	<i>Fiche d'analyse du poste de travail n°01</i>
<i>N°2</i>	<i>Fiche d'analyse du poste de travail n°02</i>
<i>N°3</i>	<i>Fiche d'analyse du poste de travail n°03</i>
<i>N°4</i>	<i>Fiche d'analyse du document n°01</i>
<i>N°5</i>	<i>Fiche d'analyse du document n°02</i>
<i>N°6</i>	<i>Fiche d'analyse du document n°03</i>
<i>N°7</i>	<i>Fiche d'analyse du document n°04</i>
<i>N°8</i>	<i>Fiche d'analyse du document n°05</i>
<i>N°9</i>	<i>Fiche d'analyse du document n°06</i>
<i>N°10</i>	<i>Fiche d'analyse du document n°07</i>
<i>N°11</i>	<i>Fiche d'analyse du document n°08</i>
<i>N°12</i>	<i>Fiche d'analyse du document n°09</i>
<i>N°13</i>	<i>Fiche d'analyse du registre n°01</i>
<i>N°14</i>	<i>Fiche d'analyse du registre n°02</i>
<i>N°15</i>	<i>Fiche d'analyse du registre n°03</i>
<i>N°16</i>	<i>Procédure de réception</i>
<i>N°17</i>	<i>Procédure de réintégration et remplacement</i>
<i>N°18</i>	<i>Procédure de reconversion</i>
<i>N°19</i>	<i>Procédure de livraison avec facture</i>
<i>N°20</i>	<i>Procédure livraison sans facture</i>
<i>N°21</i>	<i>Procédure de sortie produit fini</i>
<i>N°22</i>	<i>Description des flux</i>
<i>N°23</i>	<i>Procédure livraison sans facture</i>
<i>N°24</i>	<i>Dictionnaire de données</i>

INTRODUCTION GENERALE :

Dans tous ses domaines d'activité, l'être humain s'est mobilisé pour exploiter le temps d'une manière optimale. Le bon fonctionnement d'une entreprise se base, essentiellement sur une bonne organisation de ses services, et une bonne gestion de ses ressources, comme ses stocks par exemple.

Dans notre organisme d'accueil, l'entreprise nationale des industries en Electroménagers (ENIEM) précisément, l'unité commerciale, qui constitue notre domaine d'étude, le bon fonctionnement des mouvements des produits, est d'une importance capitale. Cela veut dire qu'il faut contrôler à tout moment, la nature de l'opération à effectuer et garantir une bonne circulation de l'information.

Notre travail, consiste en l'informatisation de la gestion des stocks et ce projet porte sur la mise en place d'une application client/serveur qui procurera un environnement convivial aux acteurs commerciaux et répondra aux contraintes de fiabilité, efficacité et surtout de disponibilité.

Pour ce faire, dans notre démarche méthodologique, nous avons utilisé la méthode MERISE et le langage de modélisation unifié UML.

Nous avons utilisé MERISE dans la partie étude préalable et UML dans la partie analyse et conception.

Notre présent mémoire s'articule ainsi autour des cinq chapitres suivant :

Chapitre 1 : Etudes préalable.

Chapitre2 : Le client /serveur.

Chapitre3 : Atelier de génie logiciel.

Chapitre4 : Analyse et conception.

Chapitre5 : Réalisation.

Une conclusion générale clos le mémoire.

BIBLIOGRAPHIE.

ANNEXES.

Dans notre étude préalable, nous avons adopté la méthode MERISE.

Description générale de la méthode <<MERISE>> :

Définition :

MERISE est une méthode d'étude et de réalisation des systèmes d'informations. Le projet qui a permis d'aboutir à sa naissance a été lancé en 1977 par le ministère français de l'industrie, dans l'objectif de doter l'administration d'une méthode de conception devant permettre à ses équipes de réussir leurs projets dans les délais et coûts prévus. La méthode MERISE utilise deux différentes approches qui sont l'approche par étapes et l'approche par niveau.

L'approche par étapes (cycle de vie) :

➤ **Schéma directeur :**

C'est l'orientation générale du développement, budget, contraintes.

➤ **Etude préalable :**

S'effectue par domaine ; analyse ; critique de la situation existante, proposition et évaluation de solution.

➤ **Etude détaillée :**

Spécification complète et détaillée du système d'information futur à partir des choix issus de l'existant.

➤ **Etude technique :**

Spécification complète du système d'information informatisé.

➤ **Production du logiciel :**

Programmation, test, intégration.

➤ **Mise en œuvre :**

C'est l'installation du logiciel réalisé

➤ **Maintenance :**

Son but est d'augmenter la durée de vie des applications et de les maintenir à niveau en tenant compte des évolutions de l'environnement.

L'approche par niveau (cycle d'abstraction) :

➤ **Le niveau conceptuel :**

Au niveau conceptuel, le système d'information est représenté

Indépendamment de son organisation et des moyens physiques et informatique qu'il pourrait utiliser. Son objectif est de répondre à la question quoi ? Les deux modèles proposés à ce niveau sont :

- **Le modèle conceptuel de données (MCD) :**

Spécification des informations sans contraintes techniques ou économique. Il est Réalisé à l'aide des trois concepts du formalisme E/A :

-entité - relation – propriété.

- **Le modèle conceptuel de traitement (MCT) :**

C'est la spécification des activités du domaine sans préciser les ressources ou leur organisation. Il est réalisé à l'aide des concepts suivants :

-processus

- opération qui comprend les concepts d'événement et synchronisation

➤ **le niveau logique / organisationnel**

A ce niveau ce font tous les choix organisationnels afin de déterminer qui fera quoi ? Ou ? Quand et comment les traitements sont réalisés ? Les deux modèles proposés à ce niveau sont :

- **modèle logique de données (MLD) :**

Description des données en tenant compte de leurs conditions et des techniques de mémorisation.

- **le modèle organisationnel de traitement (MOT) :**

Fonctionnement du domaine avec les ressources utilisées et leur organisation.

➤ **le niveau physique / opérationnel :**

A ce niveau se font tous les choix techniques permettant de déterminer les moyens informatiques. Son objectif est de répondre à la question avec quels moyens ? Les deux modèles proposés à ce niveau sont :

- **Le modèle physique de données (MPD) :**

C'est la description de la ou les BDD dans la syntaxe du SGBD.

- **Modèle opérationnel de traitement (MOPT)**

Définit l'architecture technique des programmes.

Etude Préalable

Introduction

L'objectif de cette étude est d'introduire notre organisme d'accueil par une brève description, et de cerner notre champ d'étude par une étude détaillée ,on distingue deux parties :

- Partie 1 : présentation de l'organisme d'accueil.
- Partie 2 : étude de l'existant.

1) Présentation de l'organisme d'accueil <<ENIEM>>

1.1) Historique :

L'entreprise nationale des industries de l'Electroménager (ENIEM) est issue de la restructuration organique de SONELEC (Société Nationale de fabrication et montage du matériel Electronique) en 1983. L'ENIEM à été chargé de la production et de commercialisation des produits électroménagers et disposait à sa création de :

- Complexe d'appareils ménagers (CAM) de Tizi-Ouzou entré en production et la commercialisation des produits en juin 1977.
- L'entreprise lampe de MOUHAMADIA (ULM), entre en production en février 1979.

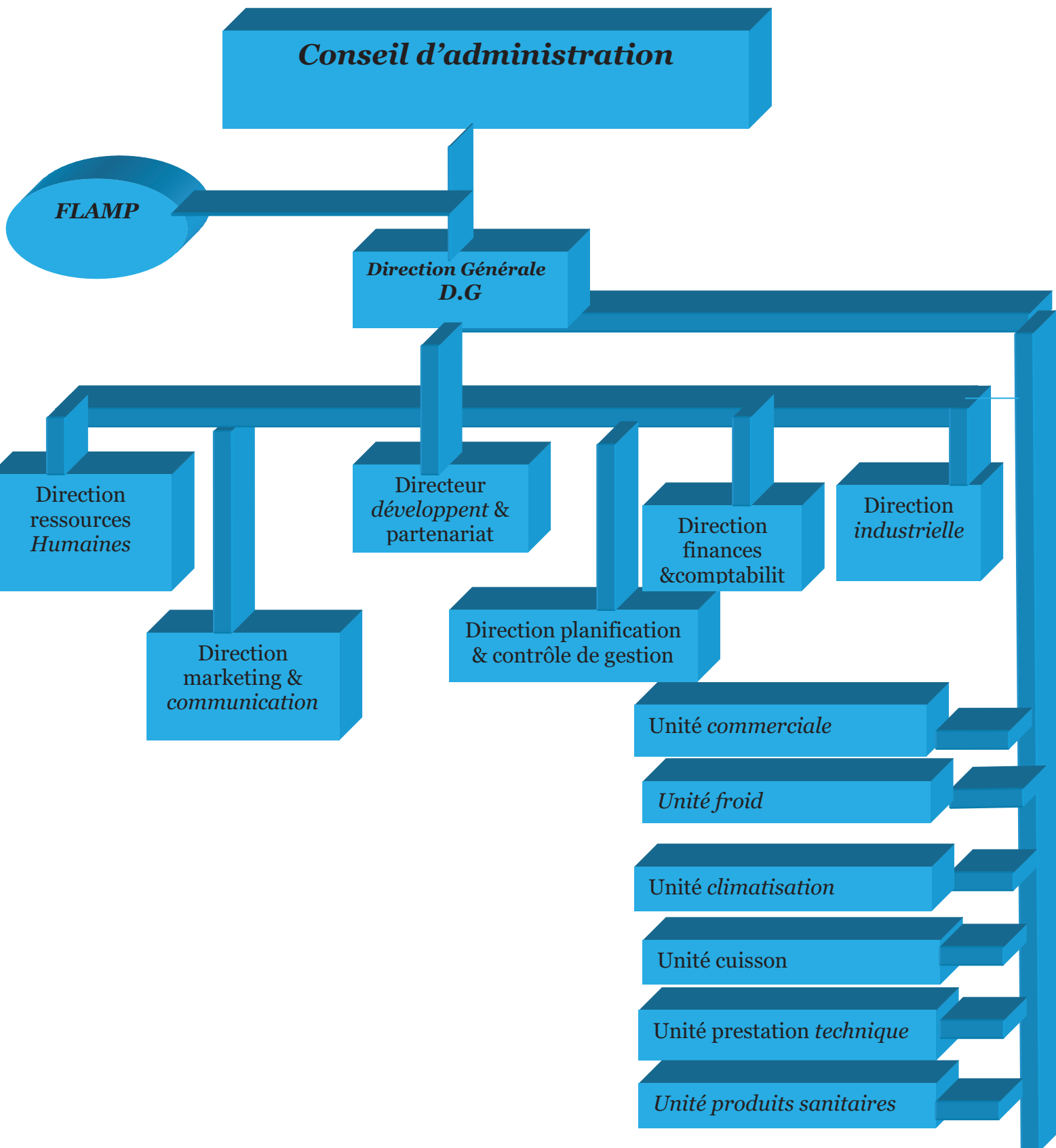
L'entreprise ENIEM est devenue une société par action en1989 .Son capitale sociale est détenue en totalité par INDELEC.

Son siège social se situe au chef lieu de la wilaya de Tizi-Ouzou,

Ses unité de production : froid, cuisson, climatisation, prestation technique, produit sanitaires et FILAMP ainsi que l'unité commerciale.

Elles sont implantées au niveau de la zone industrielle 'AISSAT-IDIR' de Oued AISSI distance de 7KM du chef lieu de la wilaya.

1.2) Mode d'organisation :

1.2.1) Organisation général :**Figure 1 : organigramme général de l'ENIEM.**

1.3) Les directions:

ENIEM est dotée d'une direction générale et de six directions centrales:

- 1.3.1) Direction générale.
- 1.3.2) Direction industrielle.
- 1.3.3) Direction des Ressources Humaines.
- 1.3.4) Direction Développement et Partenariat.
- 1.3.5) Direction des Finances et partenariat.
- 1.3.6) Direction Planification et Contrôle de Gestion.
- 1.3.7) Direction de marketing et de la communication.

1.3.1) Direction Générale :

La direction générale est l'unique qui est responsable de la stratégie et du développement de l'entreprise .Elle exerce son autorité hiérarchique et fonctionnelle sur l'ensemble des directions et des unités.

1.3.2) Direction Industrielle :

Elle est chargée de développer et de mettre en place les moyens et l'organisation industrielle nécessaire à la réalisation de la production en agissant sur l'approvisionnement ; les moyens et les techniques de production.

1.3.3) Direction Ressources Humaines :

En cohérence avec la politique de l'entreprise, la fonction ressources Humaines accroît la mobilisation et la valorisation du personnel dans ses actions au service du client.

Elle pilote le recrutement, l'accueil, l'information.

Elle supervise la gestion administrative et légale pour le personnel.

1.3.4) Direction Développement et Partenariat :

Elle est responsable de :

- Des études et du développement des produits finis,
- Du partenariat et de la sous-traitance,
- Du suivi avec la direction industrielle des actions de développement.

1.3.5) Direction Planification et De contrôle de gestion :

Cette direction est responsable du contrôle de la gestion, de l'audit finance ainsi que du budget de l'entreprise.

-Elle exploite les résultats de l'audit finance, les interprètes et fait les recommandations nécessaires.

-Elle prépare, établit et suit le budget de l'entreprise.

1.3.6) Direction Finances et Comptabilité :

Elle est Composée de deux services :

- Un service comptabilité générale ;
- Un service qui assure la vérification de l'application par la mise en œuvre d'un contrôle interne.

1.3.7) Direction Marketing et Communication :

Met en œuvre la stratégie concurrentielle de l'entreprise. Chargée de la réparation et l'organisation des foires et salons nationaux.

1.4) les unités :

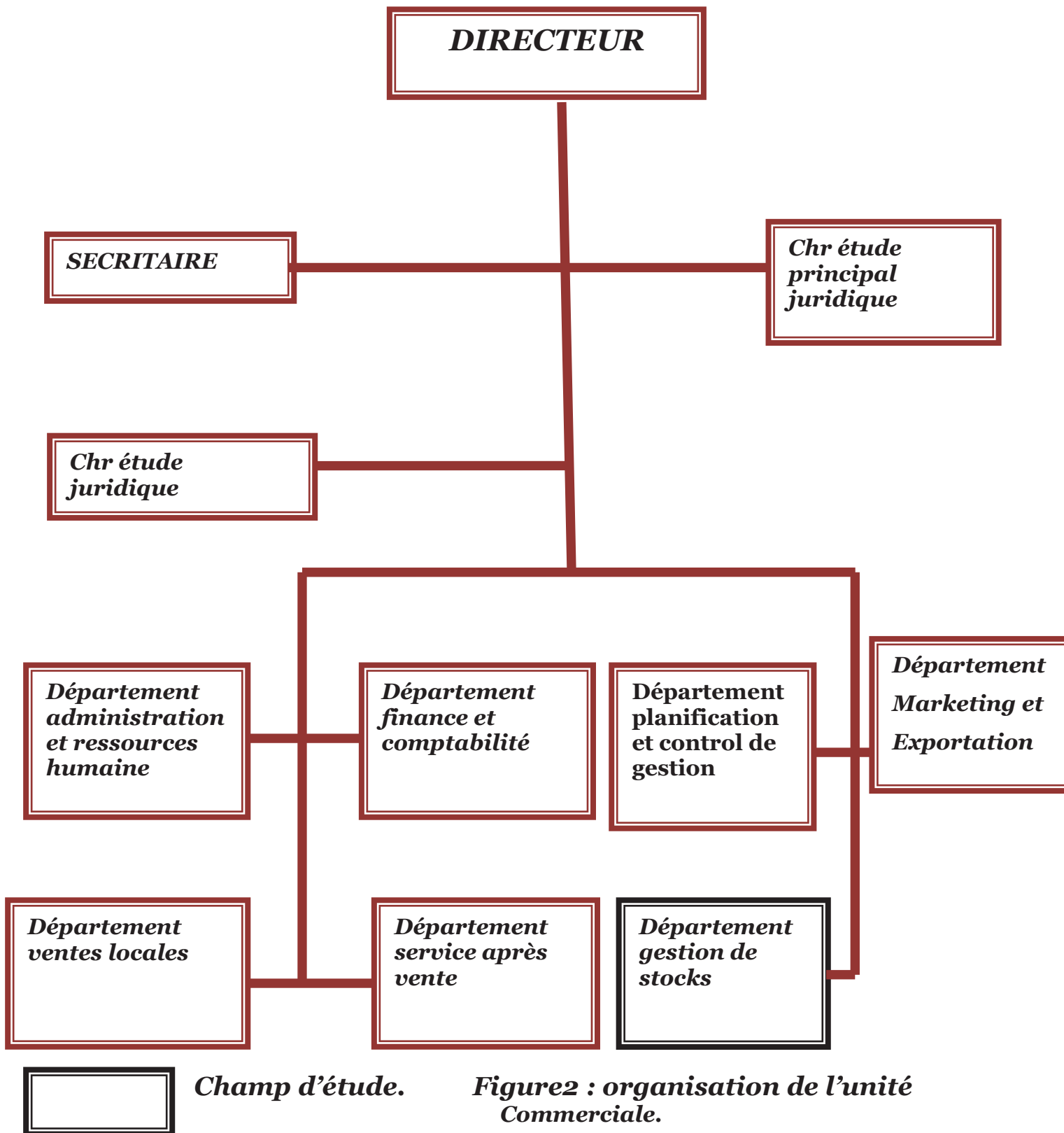
L'entreprise s'est organisée par centres d'activités stratégiques qui se composent de 05 unités de production, et d'une commerciale :

- ✓ Unité froid.
- ✓ Unité cuisson.

- ✓ Unité présentation technique (UTP).
- ✓ Unité produits sanitaires ;
- ✓ La filiale lampes (FILAMP) ;
- ✓ Unité commerciale (UC).

1.5) Présentation de l'unité commerciale :

L'Unité commerciale est chargée de la Commercialisation des produits de l'entreprise ; elle est organisée selon l'organigramme ci de sous :

1.6) Organisation de l'unité commerciale :**Figure2 : organisation de l'unité Commerciale.**

1.7) Organisation générale :

Composée essentiellement, d'une direction commerciale au sommet de la hiérarchie à laquelle sont rattachés le secrétariat, l'assistant qualité et l'assistant juridique.

Cette direction a sous sa tutelle sept départements qui collaborent pour mettre en Œuvre la stratégie commerciale de l'entreprise.

1.7.1) Département vente :

Trois services composent ce département :

➤ **Un service vente** : chargé principalement de :

- -la réception des commandes des clients.
- -La gestion des facturations.
- -la gestion des règlements.
- -le contrôle des livraisons par rapport aux facturations.

➤ **Un service client** : cette structure enregistre les clients et gère

Le réseau des agents agréé a travers le territoire nationale, et ce Par :

- -La définition des critères de sélection des agents.
- -Réception et l'étude des dossiers de soumission.

- -La classification et le suivi des clients.

➤ ***Un service synthèse et recouvrement :***

Le travail de ce Service consiste en le recouvrement des créances qui sont générées soit par les ventes à terme. On distingue :

- Les ventes par facilité,
- Les ventes par trait,
- Les ventes par chèque. Ce service s'occupe également du traitement et du suivi des marchés.

1.7.2) Département gestion de stocks :

Compose de deux services :

➤ ***Service magasin produit finis :*** sa fonction principale est la gestion des stocks de L'entreprise. En outre il doit :

- Alimenter les stocks sur base de la réception des produits en provenance de la production.
- Réintégrer des produits retournés par le client dans le cadre de la garantie, ou depuis les stocks extérieurs.

➤ ***Service programmation :***

Ce service assure les livraisons vers les clients.

Il se charge de l'établissement du bulletin de livraison, en se basant sur la facture après avoir servi le client.

Le Service programmation établit de manière périodique, les situations des stocks ainsi la balance globale des stocks.

1.7.3) Département marketing :

Ce département met en œuvre la stratégie concurrentielle de l'entreprise, il est chargé de la préparation et l'organisation des foires et salons nationaux et internationaux.

1.7.4) Département Service Après Vente :

Ce département est responsable de la gestion du circuit SAV à travers tout le territoire nationale plus de 140 agent agrès forment le réseau SAV de L'ENIEM, ils sont à la charge du département service après vente, qui doit les alimenter en pièce de rechange pour assurer les prestations de service qui s'inscrivent dans le cadre de la garantie sur les produits de l'entreprise. Lors des retours Clients, le département SAV à travers une commission technique spécialisée décide de la réparation ou remplacement du (des) produit (s) retourné (s).

1.7.5) Département finance et comptabilité:

Ce département est composé de deux services: un service comptabilité générale et un service finance .Ils assurent la comptabilité de toutes les opérations permettant d'établir des bilans comptables des résultats, ainsi que la gestion de la trésorière de l'unité commerciale. La finance concerne la couverture des dépenses des autres départements.

1.7.6) Département administration générale et ressource humaine (AGRH):

Le département administration générale et ressource humaine est formé du service gestion du personnel qui gère la formation et les correspondants sociaux, ainsi que les moyens généraux qui se chargent de la gestion des affaires générales, les achats et des relations extérieures.

1.7.7) Département contrôle de gestion :

Ce département est un organe de contrôle de toute la gestion de l'unité commerciale, il réalise et présente tous les travaux permettant de produire une information complète et cohérente des activités de l'unité (opération, commercialisation, approvisionnement et finance) il a l'accès pratiquement à toutes les informations produites par les autres services du département.

1.7.8) Description du département AGRH :

Ce département comme son nom l'indique est chargé de gérer et contrôler tous le personnel de l'unité commerciale.

Ce département comprend :

- Un chef de département et une secrétaire.
- Un service moyens généraux et ces sections.
- Un service gestion du personnel.
- un responsable informatique et un technicien supérieur en informatique de gestion.

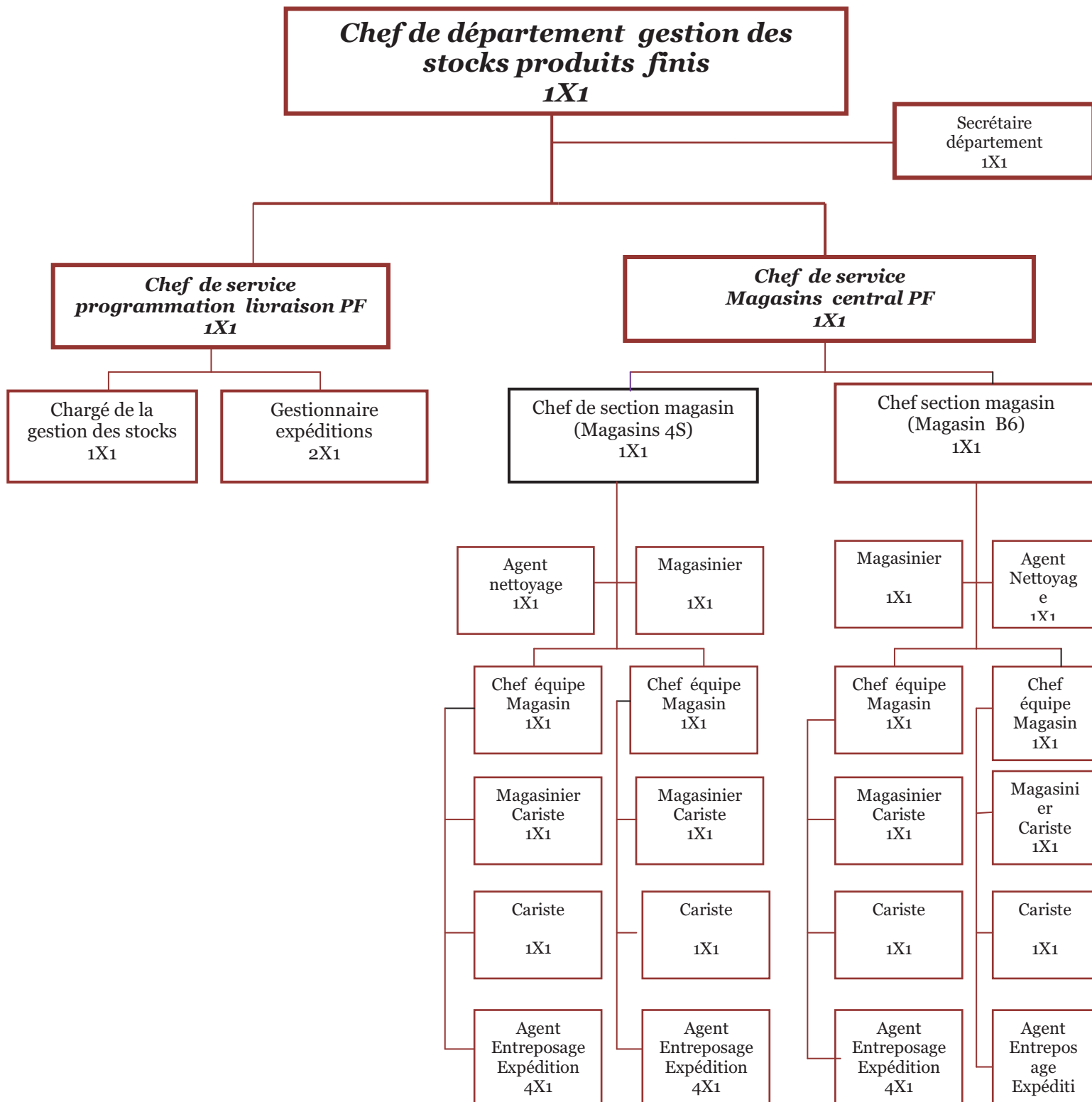
UNITE COMMERCIALE

Figure3 : Unité Commerciale.

ETUDES DES CAPACITES MATERIELLESORDINATEUR

System d'exploitation	Microsoft Windows ME service pack du system
Nom du système	stock 0.3
Nom de l'utilisateur	stock 0.3
Système d'exploitation	MICROSOFT Windows XP Professional
Service pack du système	Service pack 2
Nom du système	STOCK 01
Nom de l'utilisateur	Administrateur
Capacité du disque dur	ST3400 15A (40GO, 5400R)

PERIPHERIQUE

Imprimante	Canon LBP810
Imprimante	Canon LBP-810
Imprimante	Epson LQ-2090 ESC/p2

2) ETUDE DE L'EXISTANT :

L'étude de l'existant a pour objectif de d'écrire en détail le système d'information déjà mis en place.

L'objectif est de comprendre son fonctionnement et de recenser ses dysfonctionnements.

Classiquement, cette étude s'articule autour des points suivants :

- Etude des postes de travaux.
- Etudes des documents.
- Etudes des registres.
- Etudes des procédures.
- Etudes des flux.

2.1) ETUDES DES POSTES DE TRAVAUX :

Dans cette partie, il s'agit d'énumérer les postes de travail directement concernés par le domaine d'étude et de les étudier du point de vue :

- Responsabilités
- Taches exécutées.

LISTES DES POSTES DE TRAVAUX :

Il existe 3 postes de travail dans notre champ d'études qui sont :

- 1) Chef service magasin central produit finit.
- 2) Gestionnaire des expéditions.
- 3) Chef service programmation livraison produits finis.

FICHE D'ANALYSE DU POSTE DE TRAVAIL N°01	
<p>Code : 231.</p> <p>Désignation : chef service magasin central PF.</p> <p>Rattachement : section magasinage.</p> <p>Effectifs : 01.</p>	
RESPONSABILITES	
<p>Coordonne l'activité de sa structure.</p> <p>Assure la gestion des magasins.</p>	
Taches effectués	
<ul style="list-style-type: none"> -Gère le répertoire des produits en stocks -Gère le personnel sous son autorité. -Supervise les activités de réceptions, de tenues des stocks et de livraisons. -Veille à l'utilisation optimale du potentiel (humain et matériel) mis à sa disposition pour la réalisation des objectifs. -Contrôle et approuve les documents comptables d'entrées et sorties avant leur transmission a la gestion des stocks. -contrôle et valide les notes du débit des prestations ou consommations fournies à son service (gasoil entretien des moyens de manutention ou bâtiment...). -Exprime les besoins humains, matériels et aires de stockages nécessaires au bon fonctionnement de son service. -Veille à l'assainissement des stocks des produits détenus dans ses magasins. -Veille à la réparation au niveau des ateliers de production ou ateliers SAV. -Sollicite la commission de réforme. -Intervient au prés des services du complexe pour le règlement de litige ou sollicite des services. -Etablit les objectifs des structures en fonction des orientations données par sa hiérarchie et en accord avec les objectifs des autres structures de l'unité. -Organise le système d'inventaire des produits en magasin, en supervise l'exécution et exploite les résultats avec la gestion des stocks. 	

- Elabore un rapport hebdomadaire d'activité du service relatant notamment les problèmes et anomalies rencontrés.
- Veille à la maintenance des premiers degrés des moyens de manutention mis à sa disposition.
- Veille au respect des instructions de travail et des consignes de sécurité portées à sa connaissance.
- Veille à la propreté des magasins et sanitaires.
- Veille à l'étalonnage et vérifications des moyens de mesure.
- Veille au respect de la procédure de stockage selon les exigences de la norme ISO 14001.

FICHE D'ANLYSE DU POSTE DE TRAVAIL N°02

Code : 709.

Désignation : Gestionnaire des expéditions.

Rattachement : bureau des expéditions.

Effectifs : 02.

LES RESPONSABILITES

Procède à l'opération administrative d'enlèvement des produits finis.

LES TACHES EFFECTUEES

Accueille les clients.

Etablit :

- Le bon à servir.

- Les bons de livraisons.

- Les bons de sorties des produits finis.

Classe :

- Les factures, les bons de livraisons.

- Contrôle les bons de chargement avec les bons de livraisons.

FICHE D'ANLYSE DU POSTE DE TRAVAIL N°03	
Code : 616. Désignation : chef service programmation livraison PF. Rattachement : Bureau de programmation PF. Effectifs : 01.	
LES RESPONSABILITES	
Assure, coordonne et supervise les activités de programmation et de distribution des produits finis. Etablit le programme de distribution des stocks de produits finis.	
LES TACHES EFFECTUEES	
Etablit un programme de distribution des stocks de produits finis sur la base : Des stocks du magasin central. Des stocks des magasins de dépôts. Du programme de production. Des ventes. Des moyens de transports disponibles Intervient pour prendre toutes les décisions d'aménagement du programme de distribution rendu nécessaire par des événements imprévus. Etablit les objectifs de sa structure en fonction des orientations données par sa hiérarchie et en accord avec les objectifs des autres structures de l'Unité. Veille au tenu des stocks. Veille à la transmission des documents de mouvements des stocks aux structures concernées conformément aux procédures en vigueur. Veille au respect des procédures de gestion. Elabore les rapports hebdomadaires et mensuels d'activités. Analyse ces rapports, commente les écarts des prévisions par rapport à la réalisation et exploite l'information recueillie pour la mis a jour du programme de distribution. Sensibilise son personnel sur la norme ISO 14001 et veille à l'application de l'exigence de SME et de la législation environnementale.	

2.2) ETUDES DES DOCUMENTS :**Introduction :**

Cette études permet de comprendre l'utilité des documents et de connaître les données manipulés dans notre champ d'étude .elle se base sur l'étude de contenu des documents et leurs caractéristiques. En distingue deux types de documents :

1) Internes :

Ce document est utilisés à l'intérieure du domaine d'étude ; il existe deux types :

Document interne de liaison.

Document interne de position.

2) Externes :

Ce sont des documents provenant ou destinés a l'extérieur du domaine d'études.

REMARQUE :

Les abréviations utilisées sont :

A : Alphabétique.

N : Numérique.

AN : Alphanumérique.

D : Date.

La liste des documents utilisés dans notre champ d'étude :

Numéros	Codes	Désignation du document	Nombres d'exemplaires
01	/	Bon de réception	04
02	/	Bon de réintégration	05
03	/	Bon de reconversion	04
04	UC 2180	Bon de transfert produit finis.	04

05	UC 2150	Bon de chargement	04
06	UC 2170	Bon à servir	04
07	SC 5050	Bulletin de livraison	05
08	UC 1180	Bon de sorti produit finis	04
09	CL 5120	Facture	04

Fiche d'analyse du document N°01			
IDENTIFICATION			
<p>DESIGNATION : Bon de réception.</p> <p>CODES : sans code.</p> <p>POSTE D'ORIGINE : chef de magasin produit fini.</p> <p>POSTE DESTINATAIRE : comptabilité et finance, département planification.</p> <p>Utilité : confirmer la réception des produits.</p> <p>Nature : interne.</p> <p>Nombre d'exemplaires:04.</p>			
Contenu			
Rubrique	TYPE	TAILLE	OBSERVATION
Numéro du bon de réception	AN	10	Codification
Date d'établissement du bon de réception	D	10	JJ/MM/AAAA

Référence mouvement	AN	25	codification
Numéro de référence mouvement	AN	10	
Nom de fournisseur	A	30	
Code article	N	06	
Désignation article	AN	30	
Quantité de réception	N	06	en dinars
Prix	AN	09	
Nom de dépôt	AN	25	
Total ligne	AN	09	

Fiche d'analyse du document N°02			
IDENTIFICATION			
<p>DESIGNATION : Bon de réintégration.</p> <p>CODES : sans code.</p> <p>POSTE D'ORIGINE : chef d'équipe magasin.</p> <p>POSTE DESTINATAIRE : comptabilité et finance.</p> <p>Utilité : justifier la réintégration.</p> <p>Nature : interne.</p> <p>Nombre d'exemplaire : 05.</p>			
Contenu			
Rubrique	TYPE	TAILLE	OBSERVATION
Numéro de bon de réintégration	AN	10	Codification
Numéro de bon de livraison	AN	10	codification
Date de bon de réintégration	D	10	JJ/MM/AAAA
Code du bon	AN	05	codification
Adresse client	AN	50	
Catégorie	A	06	
Willaya de client	A	16	
Code article			
Désignation article	N	06	Codification
	AN	30	

Unité de mesure			
Nom de dépôt	A	03	Appareil
Quantité rendue	AN	30	
	N	06	

Fiche d'analyse du document N°03			
IDENTIFICATION			
<p>DESIGNATION : Bon de reconversion.</p> <p>CODE : sans code.</p> <p>POSTE D'ORIGINE : chef d'équipe magasin.</p> <p>POSTE DESTINATAIRE : chef de service magasin produit fini.</p> <p>Utilité : confirmer la reconversion.</p> <p>Nature : interne.</p> <p>Nombre d'exemplaires : 04.</p>			
Contenu			
Rubrique	TYPE	TAILLE	OBSERVATION
Numéro bon de reconversion	AN	10	Codification
Date d'établissement du bon	D	10	JJ/MM/AAAA
Source destination	A	30	
code article	A	30	
	N	06	Codification

désignation article	AN	30	
quantité	N	06	
nom du dépôt	AN	30	
prix	AN	09	Dinars
totale ligne	AN	09	Dinars

Fiche d'analyse du document N°04			
IDENTIFICATION			
<p>DESIGNATION : Bon de transfert produit finis.</p> <p>CODES : UC 2180.</p> <p>POSTE D'ORIGINE : gestionnaire des expéditions.</p> <p>POSTE DESTINATAIRE : comptabilité et finance, département planification.</p> <p>Utilité : confirmer le transfert des produits.</p> <p>Nature : interne.</p> <p>Nombre d'exemplaires:04.</p>			
Contenu			
Rubrique	TYPE	TAILLE	OBSERVATION
Numéro de bon de transfert produit finis	AN	10	codification
Nom de département	A	30	
Nom de service	A	30	
Date			

d'établissement de bon de transfert	D	10	JJ/MM/AAAA
Nom de l'expéditeur	A	30	
Raison sociale de l'expéditeur	A	60	
Adresse de l'expéditeur	AN	50	
Numéro de téléphone de l'expéditeur	N	10	
Fax expéditeur	N	30	
Email expéditeur	AN	50	
Numéro de registre de commerce d'expéditeur	N	06	Codification
Nom de destinataire	A	30	
Raison sociale destinataire	A	60	
Adresse destinataire	AN	50	

Numéro de téléphone			
Destinataire	N	10	
Fax du destinataire	AN	30	
Email du destinataire	AN	50	
Numéro de registre de commerce du destinataire	N	06	
Désignation du produit	AN	30	Codification
Quantité transférée	N	06	
Observation	A	30	

Fiche d'analyse du document N°05			
IDENTIFICATION			
<p>DESIGNATION : Bon de chargement.</p> <p>CODES : UC 2150.</p> <p>POSTE D'ORIGINE : chef d'équipe magasin .</p> <p>POSTE DESTINATAIRE : gestionnaire des expéditions.</p> <p>Utilité : confirmer le chargement des stocks.</p> <p>Nature : interne.</p> <p>Nombre d'exemplaires:04.</p>			
Contenu			
Rubrique	TYPE	TAILLE	OBSERVATION
Numéro du bon	N	06	Codification
Fonction	A	30	
Structure	A	30	
Centre des frais	N	03	
Numéro du véhicule	N	10	
Appartenance	A	35	
Produit chargés	A	30	
Quantité chargée	N	06	

Fiche d'analyse du document N°07			
IDENTIFICATION			
<p>DESIGNATION : Bulletin de livraison.</p> <p>CODES : SC 5050.</p> <p>POSTE D'ORIGINE : gestionnaire des expéditions.</p> <p>POSTE DESTINATAIRE : chef service magasin PF, service facturation, comptabilité et finance, client, département planification.</p> <p>Utilité : confirmer la livraison.</p> <p>Nature : interne.</p> <p>Nombre d'exemplaire : 05.</p>			
Contenu			
Rubrique	TYPE	TAILLE	OBSERVATION
Numéro du bulletin	N	06	Codification Exemple : facture /B.E.A.F/O.D.T/F.REMP.
Nom du client	A	30	
Adresse client	AN	50	
Type de livraison	A	30	
Numéro immatriculation	AN	30	
Identificateur fiscal	N	06	
Numéro article fiscal	N	06	
Registre commerciale	AN	10	

Numéro du bon de chargement	N	06	Codification
Observation	A	30	
Référence	N	06	
Désignation du produit	AN	30	
Quantité livrée	N	06	

Remarque :

B.E.A.F : bon d'enlèvement avant facturation.

O.D.T : ordre de transfert.

F.REMP : facture de remplacement.

Fiche d'analyse du document N°08			
IDENTIFICATION			
DESIGNATION : Bon de sorti produit finis. CODES : UC 1180. POSTE D'ORIGINE : gestionnaire des expéditions. POSTE DESTINATAIRE : le client. Utilité : vérification de livraison. Nature : externe. Nombre d'exemplaires:04.			
Contenu			
Rubrique	TYPE	TAILLE	OBSERVATION
Numéro du bon	N	06	Codification

de sorti produit fini	A	30	Codification
Fonction	A	30	
Structure	A	03	
Centre de frais	N	30	
Nom du client	A	30	
Nom du chauffeur	A	30	
Immatriculation	AN	30	
Désignation de produit à faire sortir	AN	30	
Quantité du produit sortant	N	06	
Observation	A	30	
Date de sortie gardiennage	D	10	JJ/MM/AAAA
Heure du gardiennage	AN	05	

Fiche d'analyse du document N°09			
IDENTIFICATION			
<p>DESIGNATION : facture.</p> <p>CODES : CL 5120.</p> <p>POSTE D'ORIGINE : service facturation.</p> <p>POSTE DESTINATAIRE : client.</p> <p>Utilité : confirmer la facturation du produit.</p> <p>Nature : externe.</p> <p>Nombre d'exemplaires:04.</p>			
Contenu			
Rubrique	TYPE	TAILLE	OBSERVATION
Numéro facture.	N	06	Codification
Date d'établissement de la facture.	D	10	JJ/MM/AAAA.
Numéro commande	N	06	Codification
Nom du client	A	30	
Mode de règlement	A	30	
Nature du règlement	A	30	
Numéro identificateur	N	06	Codification

Numéro registre de commerce	N	06	Codification
Code article			
Désignation	AN	06	
Quantité article	A	30	
Prix unitaire	AN	06	
Montant total	N	06	
Total-hors taxe	N	06	
Montant TVA	N	06	
Net à payer	N	06	
	N	06	

2.3) Etudes des registres :

Listes des registres utilisés dans notre domaine d'étude :

Numéro	Code	Désignation
1	01	Mouvements journaliers
2	02	Réintégration et reconversion conforme et non conforme
3	03	Mouvements

Fiche d'analyse du registre N°1		
Identifications		
Désignation : mouvements journaliers. Code : 01. Support : registre. Localisation : sur le bureau. Rôle : enregistrement des mouvements journaliers.		
Operations effectuées		
Rubrique	Réalisateur	Fréquence
-Réception produit. -Réintégration conforme et non conforme. -Sorties des produits vers : client dépôts. - Les reconversions conformes et non conformes. - les retours fournisseurs.	Gestionnaire des stocks	A chaque enregistrement d'un mouvement.

Description du contenu			
Rubrique	Taille	Type	Observation
Date	10	D	JJ/MM/AAAA
N° des bons	15	N	
Provenance	15	A	
Quantité	06	N	

Fiche d'analyse du registre N°2			
Identifications			
<p>Désignation : réintégration et reconversion conforme et non conforme.</p> <p>Code : 02.</p> <p>Support : registre.</p> <p>Localisation : sur le bureau.</p> <p>Rôle : Détailler les intégrations et reconversions conforme et non conforme.</p>			
Operations effectuées			
Rubrique	Réalisateur	Fréquence	
-Année -Réintégration -Reconversion	Gestionnaire des stocks	A chaque enregistrement d'un mouvement.	
Description du contenu			
Rubrique	Taille	Type	Observation
Date d'établissement du registre	10 06	D N	JJ/MM/AAAA Codification
Numéro bon de réception	30	AN	
Désignation produits	06	N	Codification
Numéro de provenance	10	N	Codification
Numéro de série	06	N	Exemple
Statut	06	N	{CF/NCF}

numéro de l'ordre notes	10	D	Codification
date de l'ordre note	06	N	JJ/MM/AAAA
numéro de fiche de remplacement	10	D	
Dates fiche remplacement	06	N	JJ/MM/AAAA
Numéro de fiche de non conformité	10	D	Codification
Date de fiche de non conformité	06	N	JJ/MM/AAAA
Numéro de service après vente	10	D	codification
	30	A	JJ/MM/AAAA
	30	A	JJ/MM/AAAA
Date de service après vente			Exemple {REP/NREP}
Observation	06	N	
Etat de fabrication	10	D	Codification
Numéro de bon de réception manuelle			JJ/MM/AAAA
Date de bon de réception manuel			

Fiche d'analyse du registre N°03		
Identifications		
Désignation : Mouvement. Code : 03. Support : registre. Localisation : sur le bureau. Rôle : enregistrement des mouvements.		
Operations effectuées		
Rubrique	Réalisateur	Fréquence
-Bon de commande		
-Demande de travail		

-Consommation gas-oil -Demande d'intervention -Demande de transport -Fiche de mise à disposition	Gestionnaire des stocks		A chaque enregistrement d'un mouvement.
Description du contenu			
Rubrique	Taille	Type	Observation
-Date d'établissement du registre	10	D	JJ/MM/AAAA
	06	N	Codification
-Numéro bon des mouvements	A	30	
	10	AN	
-Nom du conducteur	30	A	
-Désignation des Clark	10	D	JJ/MM/AAAA
-Désignation des travaux			
-Date de réalisation	06	N	Codification
bonde commande	30	AN	
-Numéro bon de commande	10	D	JJ/MM/AAAA
-Désignation bon de commande	10	D	JJ/MM/AAAA
-Date accusé de réception	06	N	Codification
du bon de commande	10	D	JJ/MM/AAAA
-Date de réalisation du bon de commande	30	AN	
	10	D	JJ/MM/AAAA
-Numéro de la demande de travail	06	N	Codification
	10	D	JJ/MM/AAAA
-Date de demande de travail	30	AN	
-Désignation des travaux	06	N	Codification
-Date de réalisation des travaux	30	A	
	30	A	
-Numéro de bon d'intervention	50	AN	

-Date d'établissement du bon d'intervention	06 10	N D	Codification JJ/MM/AAAA
-Désignation des équipements	10 06	D N	JJ/MM/AAAA JJ/MM/AAAA
-Numéro de l'inventaire	30	AN	
-Anomalie constaté	30	A	
-Observation sur les cas de transports	30	A	CODIFICATION
-Lieu d'enlèvement	50	AN	
-Numéro de demande de transport	06	N	CODIFICATION
-Date de demande de transport	10	D	JJ/MM/AAAA JJ/MM/AAAA
-Date de mise à disposition	10	D	
-Numéro de la fiche de mise à disposition	06	N	
-Désignation de la fiche de mise à disposition	30	AN	
-Observation sur la mise à disposition	50	A	

--	--	--	--

2.3) Etudes des registres :

Listes des registres utilisés dans notre domaine d'étude :

Numéro	Code	Désignation
1	01	Mouvements journaliers
2	02	Réintégration et reconversion conforme et non conforme
3	03	Mouvements

Fiche d'analyse du registre N°1			
Identifications			
Désignation : mouvements journaliers. Code : 01. Support : registre. Localisation : sur le bureau. Rôle : enregistrement des mouvements journaliers.			
Operations effectuées			
Rubrique	Réalisateur	Fréquence	
-Réception produit. -Réintégration conforme et non conforme. -Sorties des produits vers : client dépôts. - Les reconversions conformes et non conformes. -Les retours fournisseurs.	Gestionnaire des stocks	A chaque enregistrement d'un mouvement.	
Description du contenu			
Rubrique	Taille	Type	Observation
Date	10	D	JJ/MM/AAAA
N° des bons	15	N	

Provenance	15	A	
Quantité	06	N	

Fiche d’analyse du registre N°2			
Identifications			
Désignation : réintégration et reconversion conforme et non conforme.			
Code : 02.			
Support : registre.			
Localisation : sur le bureau.			
Rôle : Détailler les intégrations et reconversions conforme et non conforme.			
Operations effectuées			
Rubrique	Réalisateur	Fréquence	
-Année -Réintégration -Reconversion	Gestionnaire des stocks	A chaque enregistrement d’un mouvement.	
Description du contenu			
Rubrique	Taille	Type	Observation
Date d’établissement du registre	10	D	JJ/MM/AAAA
Numéro bon de réception	06	N	Codification
Désignation produits	30	AN	
Numéro de provenance	06	N	Codification
Numéro de série	10	N	Codification
Statut	06	N	Exemple {CF/NCF}
numéro de l’ordre notes	06	N	Codification
date de l’ordre note	10	D	JJ/MM/AAAA
numéro de fiche de remplacement	06	N	
Dates fiche de remplacement	10	D	JJ/MM/AAAA
Numéro de fiche de non conformité	06	N	Codification
Date de fiche de non conformité	10	D	JJ/MM/AAAA
Numéro de service après vente	06	N	codification

DATE de service après vente	10	D	JJ/MM/AAAA
Observation	30	A	Exemple {REP/NREP}
Etat de fabrication	30	A	
Numéro de bon de réception manuelle	06	N	Codification
Date de bon de réception manuel	10	D	JJ/MM/AAAA

Fiche d’analyse du registre N°03			
Identifications			
Désignation : Mouvement.			
Code : 03.			
Support : registre.			
Localisation : sur le bureau.			
Rôle : enregistrement des mouvements.			
Operations effectuées			
Rubrique	Réalisateur	Fréquence	
-Bon de commande -Demande de travail -Consommation gas-oil -Demande d’intervention -Demande de transport -Fiche de mise à disposition	Gestionnaire des stocks	A chaque enregistrement d’un mouvement.	
Description du contenu			
Rubrique	Taille	Type	Observation
-Date d’établissement du registre	10	D	JJ/MM/AAAA
-Numéro bon des mouvements	06	N	Codification
-Nom du conducteur	A	30	
-Désignation des Clark	10	AN	
-Désignation des travaux	30	A	
-Date de réalisation bonde commande	10	D	JJ/MM/AAAA
-Numéro bon de commande	06	N	Codification

-Désignation bon de commande	30	AN	
-Date accusé de réception du bon de commande	10	D	JJ/MM/AAAA
-Date de réalisation du bon de commande	10	D	JJ/MM/AAAA
-Numéro de la demande de travail	06	N	Codification
-Date de demande de travail	10	D	JJ/MM/AAAA
-Désignation des travaux	30	AN	
-Date de réalisation des travaux	10	D	JJ/MM/AAAA
-Numéro de bon d'intervention	06	N	Codification
-Date d'établissement du bon d'intervention	10	D	JJ/MM/AAAA
-Désignation des équipements	30	AN	
-Numéro de l'inventaire	06	N	Codification
-Anomalie constaté	30	A	
-Observation sur les cas de transports	30	A	
-Lieu d'enlèvement	50	AN	
-Numéro de demande de transport	06	N	Codification
-Date de demande de transport	10	D	JJ/MM/AAAA
-Date de mise à disposition	10	D	JJ/MM/AAAA
-Numéro de la fiche de mise à disposition	06	N	JJ/MM/AAAA
-Désignation de la fiche de mise à disposition	30	AN	
-Observation sur la mise à disposition	50	A	

Modèle organisationnel des traitements existant (MOT)**2.4) Etude des procédures :**

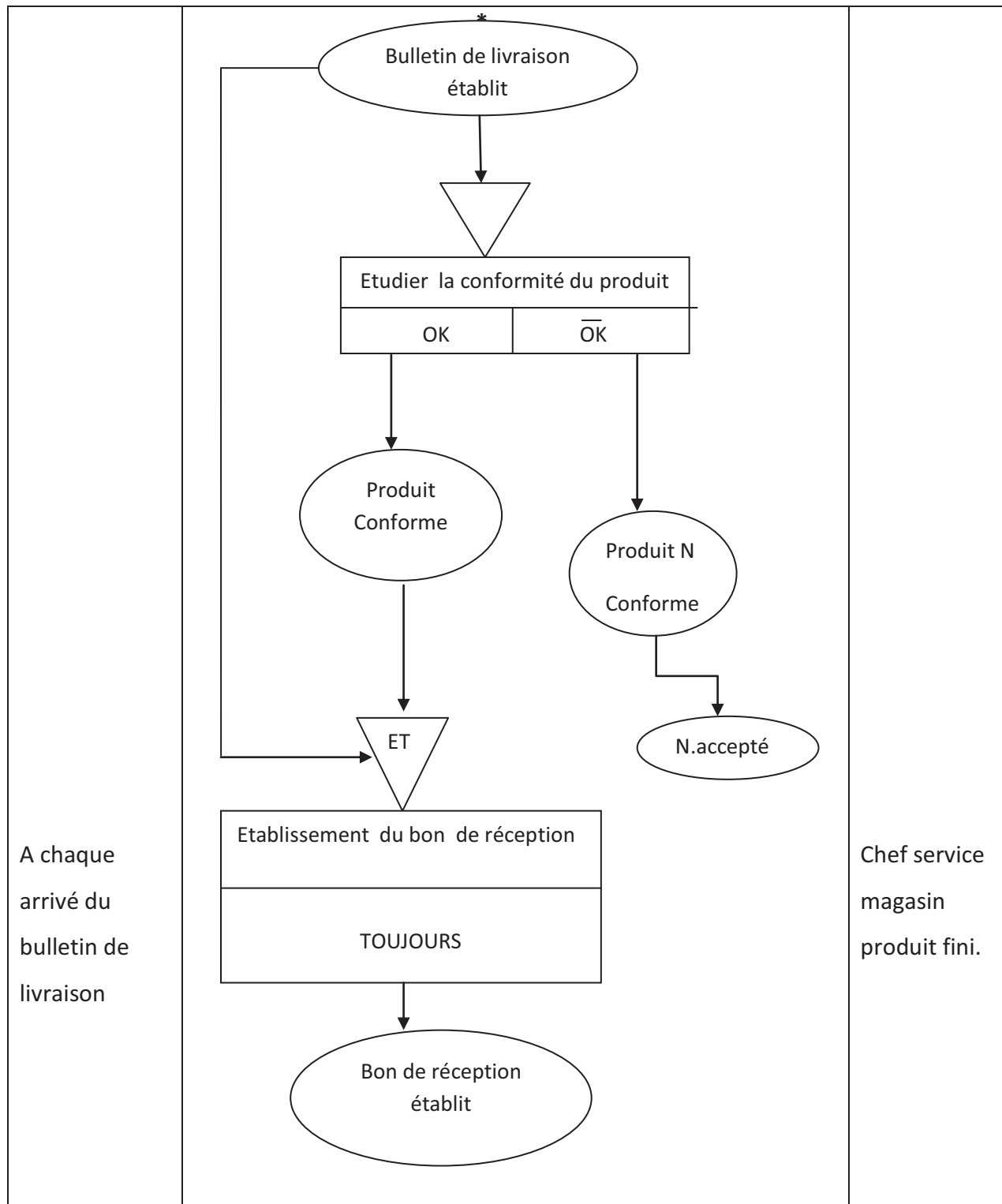
Le but de cette étude est de comprendre le fonctionnement du système d'information mis en place, de détecter les lacunes et de cibler leurs causes.

La liste des procédures :

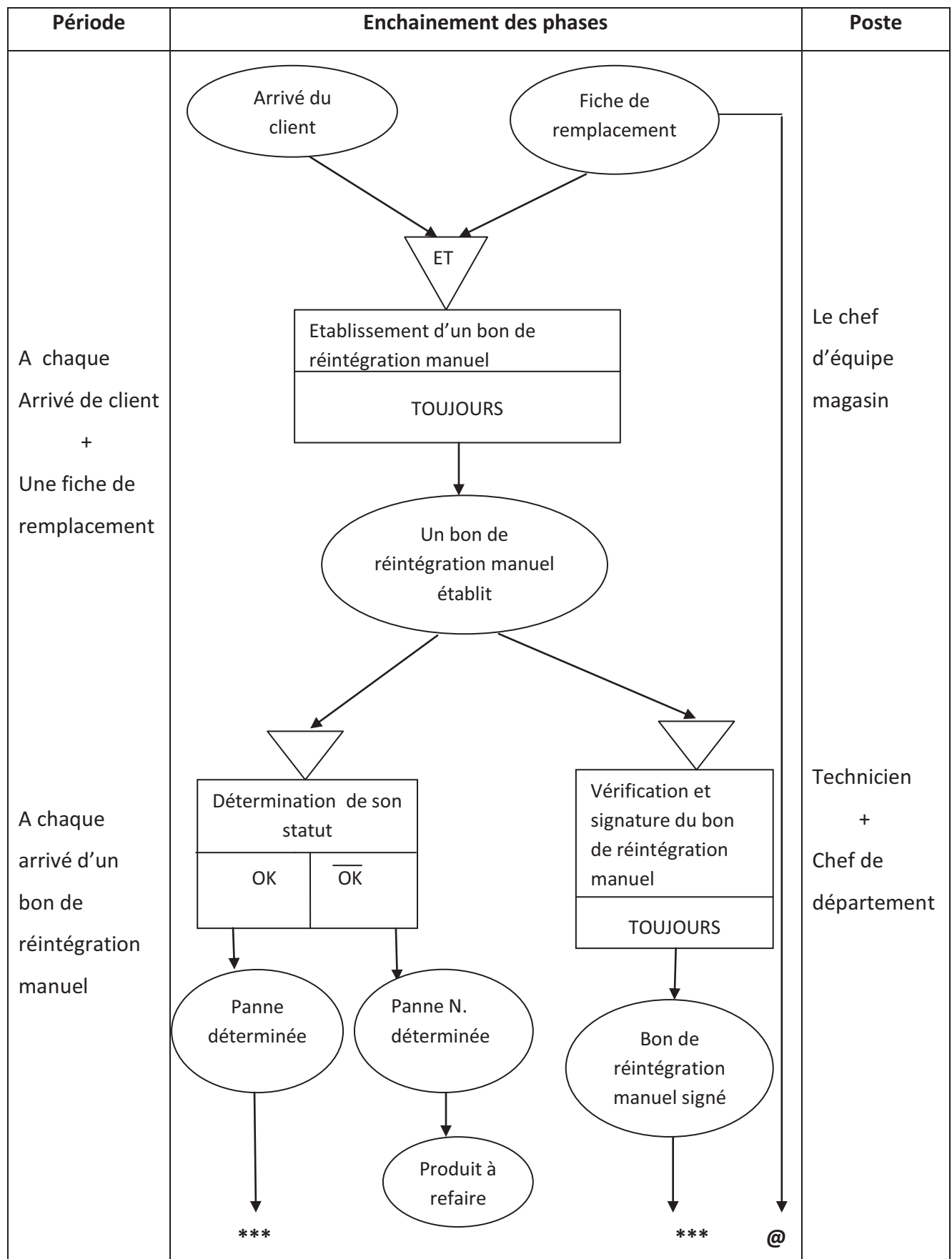
- ❖ Procédure N°01 : procédure de réception.
- ❖ Procédure N°02 : procédure de réintégration et remplacement.
- ❖ Procédure N°03 : procédure de reconversion.
- ❖ Procédure N°04 : procédure de livraison avec facture.
- ❖ Procédure N°05 : procédure de livraison sans facture.
- ❖ Procédure N°06 : procédure de sortie produit fini.

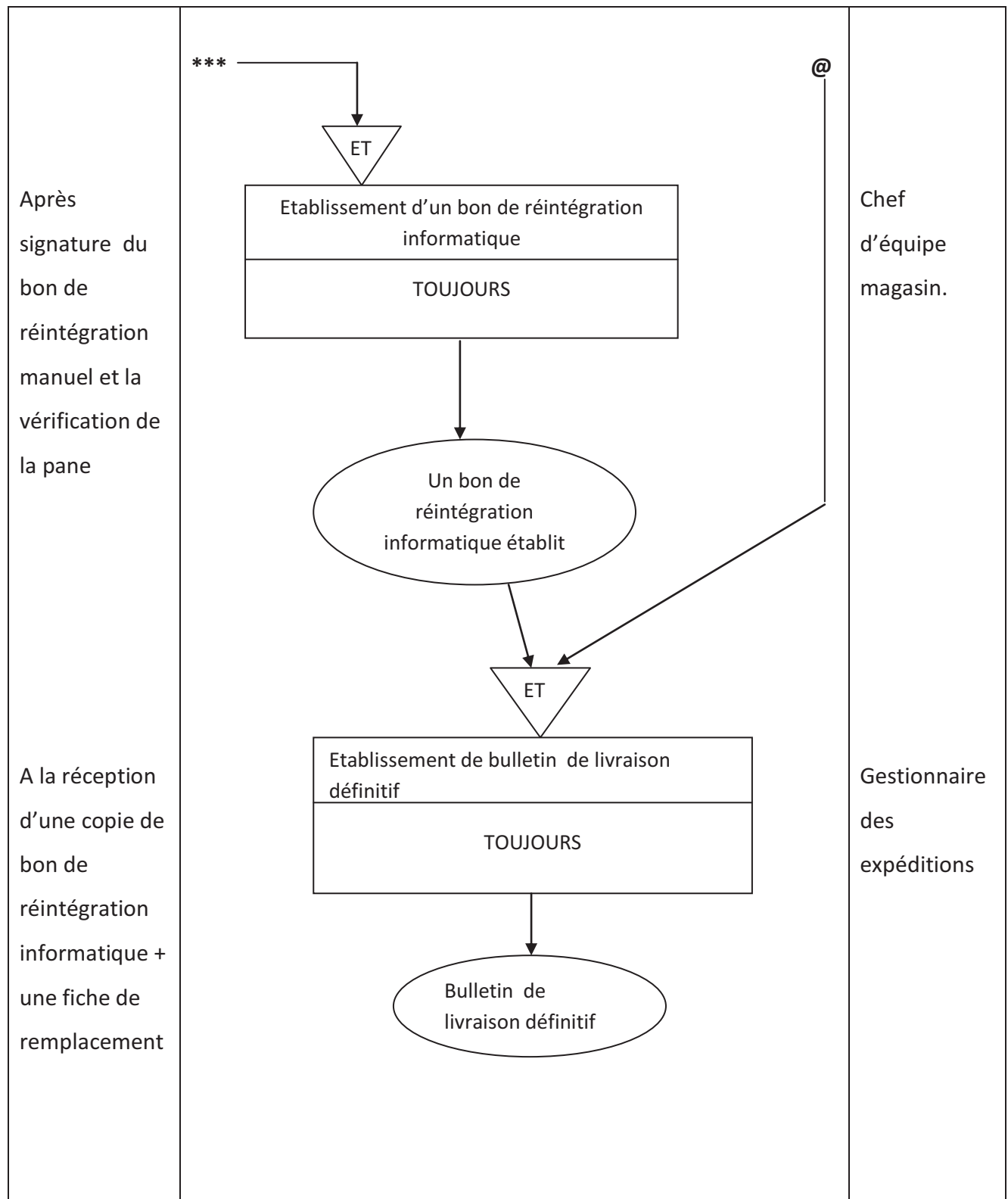
PROCEDURE N°01 : procédure de réception.

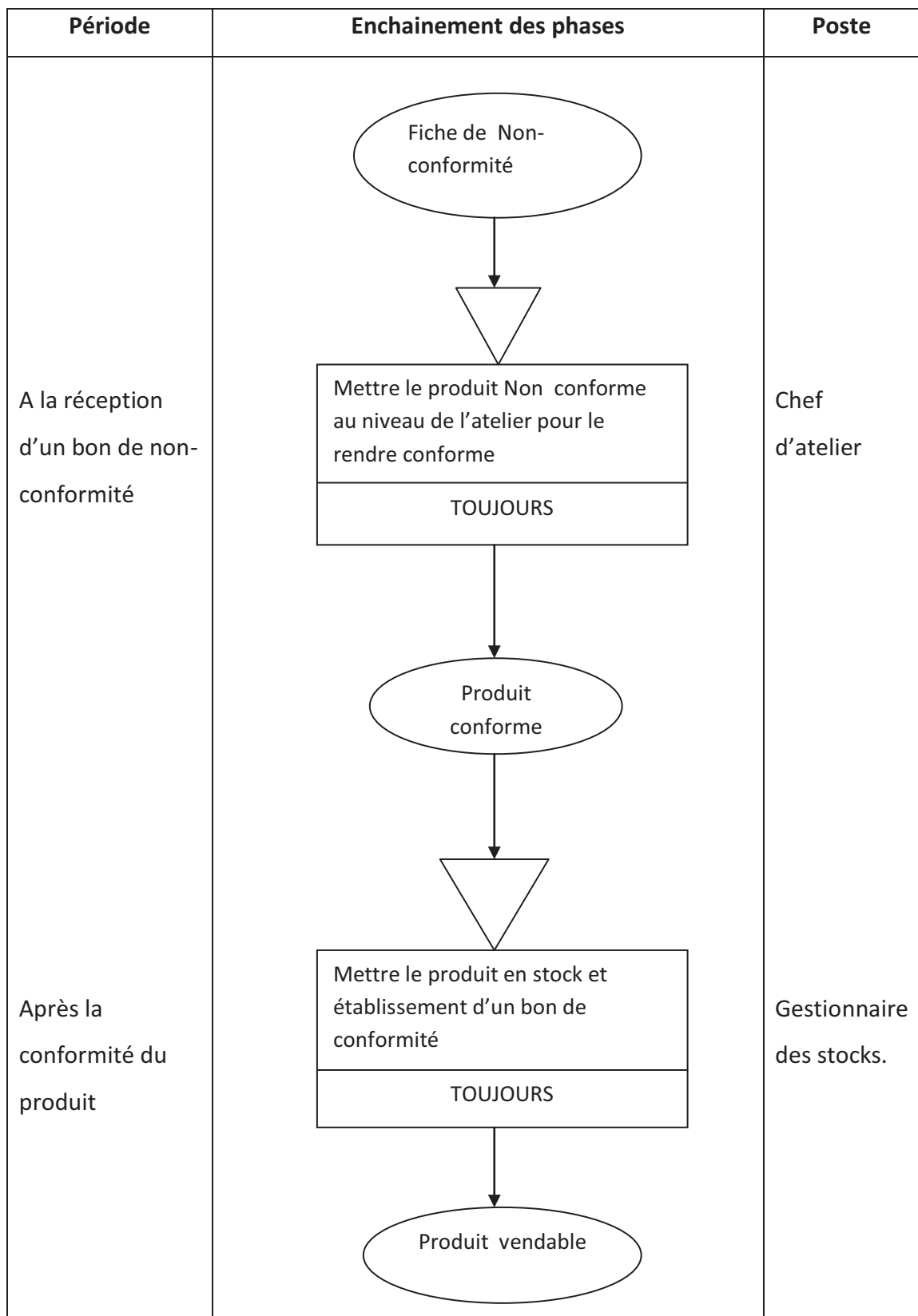
Période	Enchaînement des phases	Poste
A chaque arrivé de bon commande	<pre> graph TD A([Arriver du magasinier cariste]) --> C{ET} B([Bon de commande]) --> C C --> D[-Etablissement du bulletin de livraison TOUJOURS] D --> E[*] </pre>	Gestionnaire des expéditions.

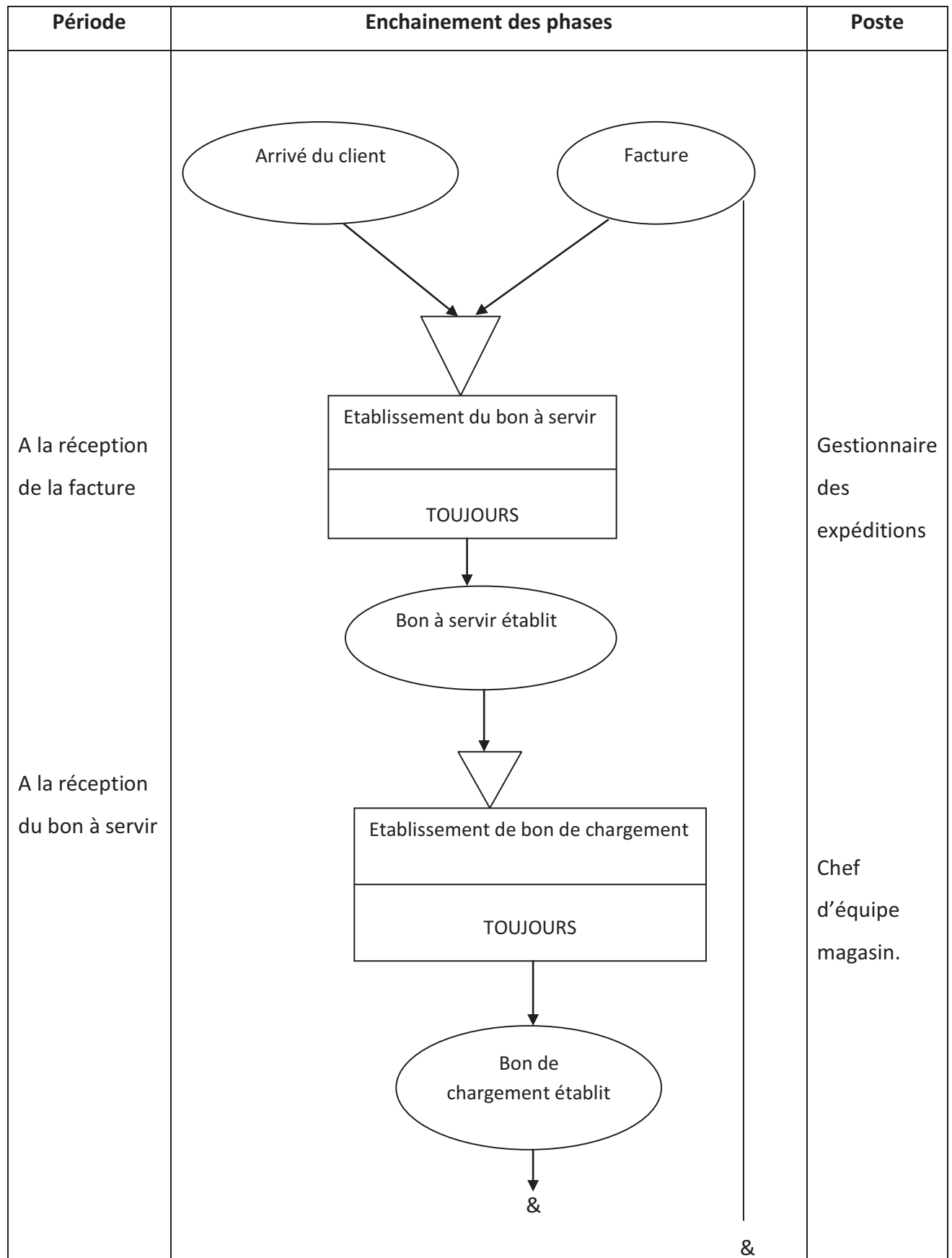


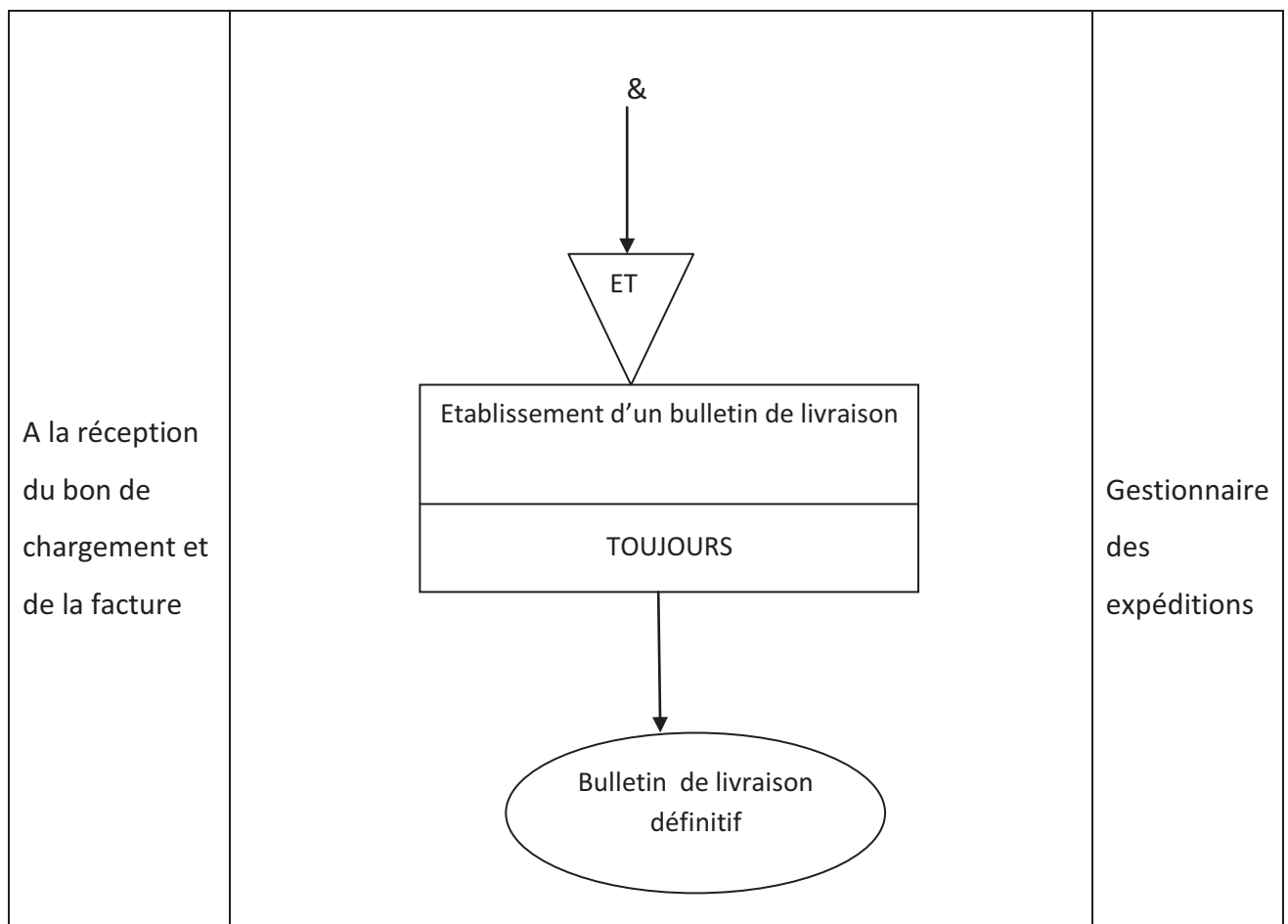
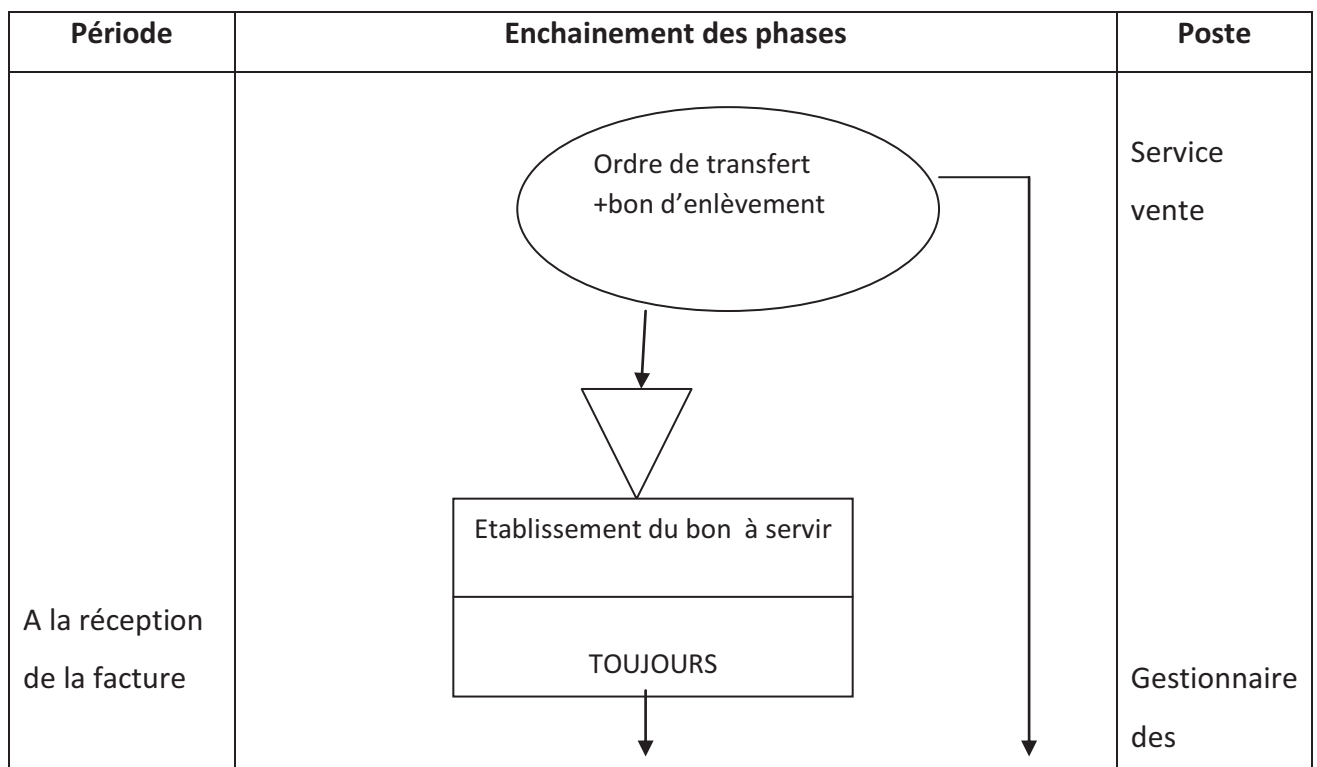
PROCEDURE N°02 : procédure de réintégration et de remplacement.

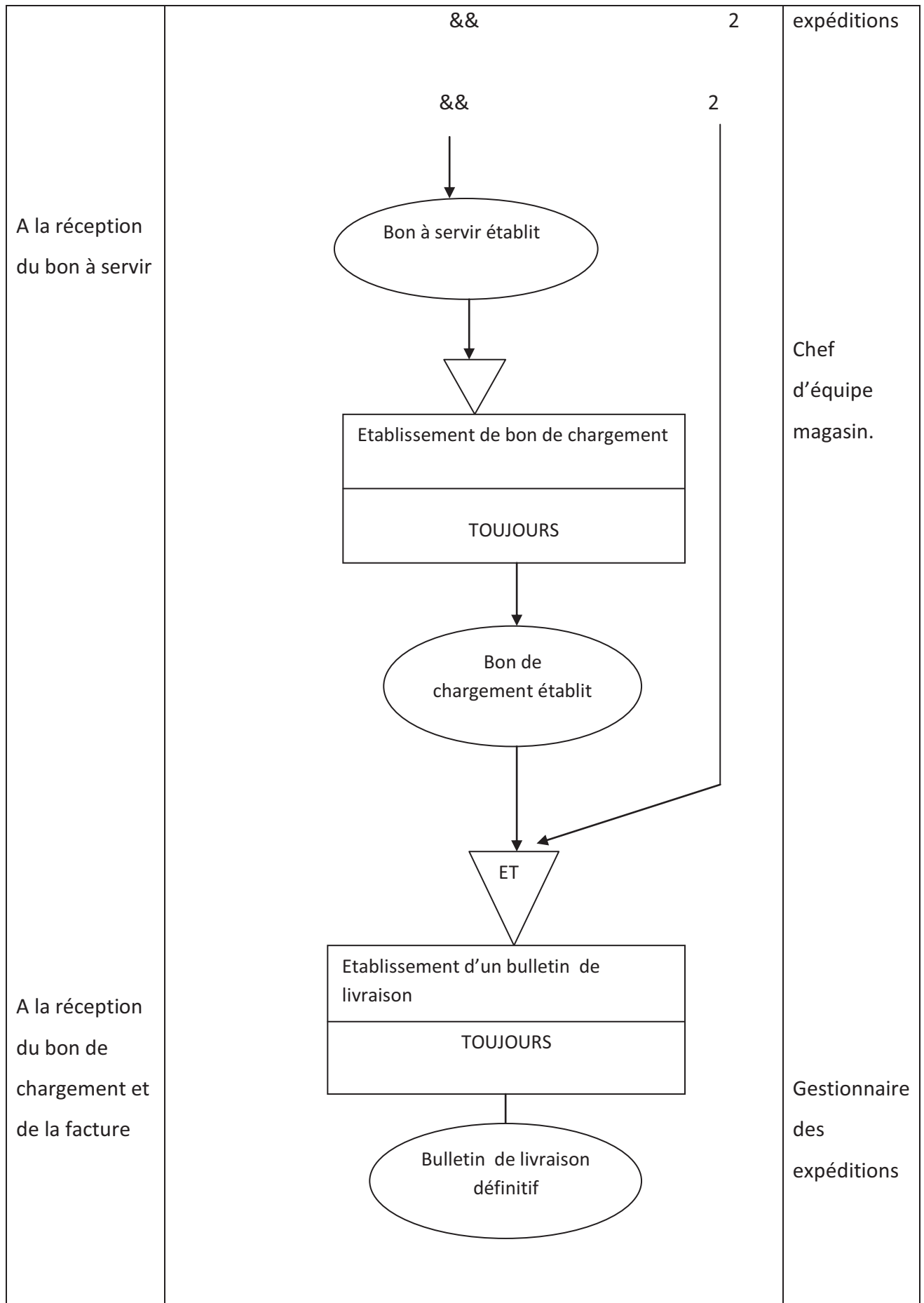


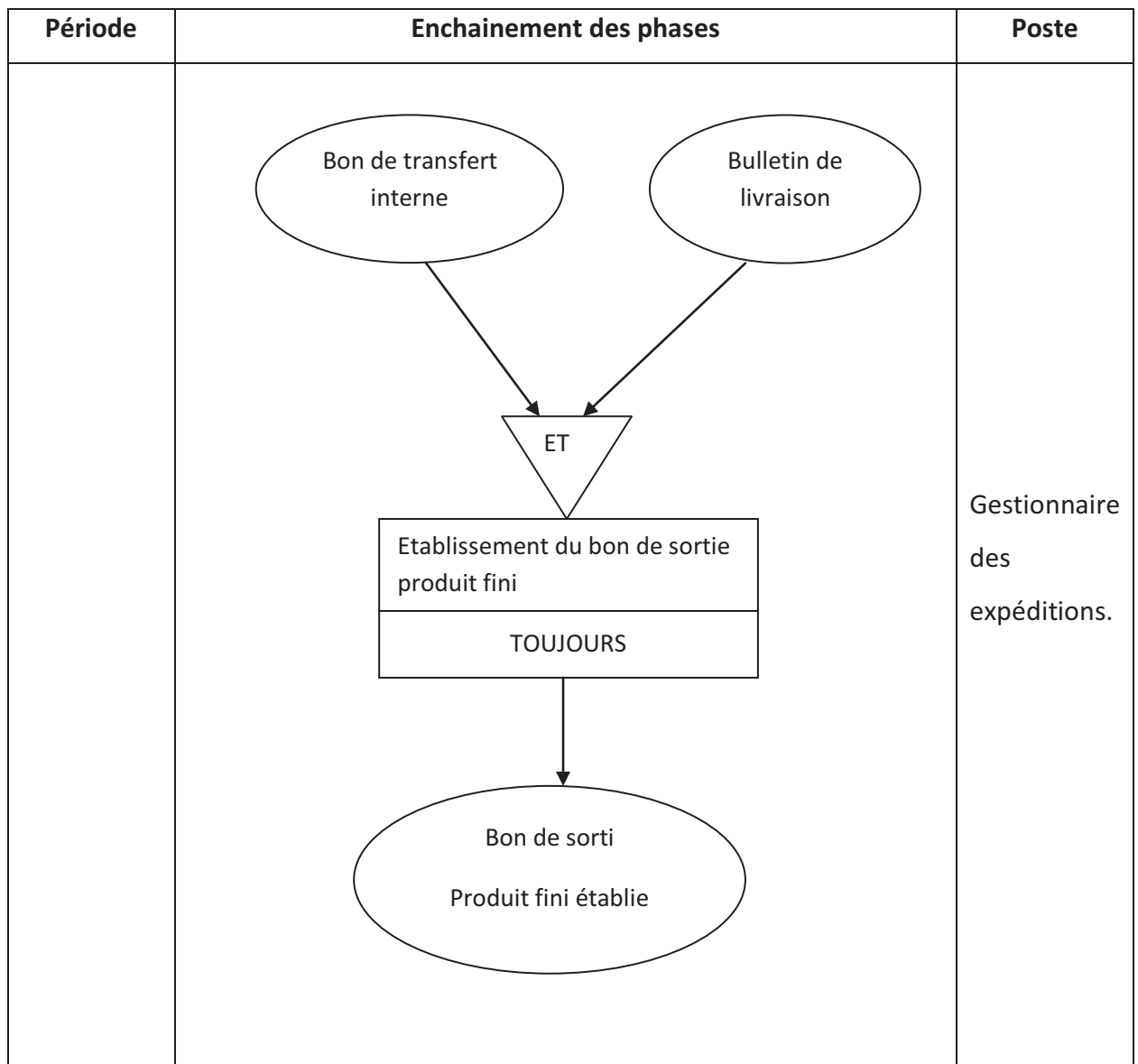


PROCEDURE N°03 : procédure de reconversion.

PROCEDURE N°04 : procédure de livraison avec facture.

**PROCEDURE N°05 :** procédure de livraison sans facture.



PROCEDURE N°06 : procédure de sortie produit fini.

3) La codification :

Définition : La codification est l'action d'associer un code à une information ou un objet. Un code est un nom abrégé associé à un objet.

Différent types de la codification :

Codification séquentielle : elle consiste à affecter des numéros consécutifs aux objets à codifier.

Codification par tranche : elle consiste à réserver une tranche de code à chaque catégorie d'objet, à l'intérieur d'une même tranche la codification est séquentielle.

Codification articulée : chaque code est découpé en plusieurs zones appelées descripteurs, chacun de ces derniers ayant un sens particulier relatif à l'objet à codifier.


Codification par niveau : c'est un cas particulier de la codification articulée dans lequel les descripteurs sont des niveaux qui se succèdent de façon hiérarchique.

Codification mnémonique : elle consiste à associer au nom de l'objet un nom court qui rappelle l'objet codifié, elle porte sur le nom plutôt que sur sa valeur.

Les codifications existantes :

- ❖ Numéro du bon de réception :

Type : codification articulée.

Format : N/AAAA. 

Numéro séquentiel.

Exemple : 1/2011.

- ❖ Code article du bon de réception.

Type : codification séquentiel :

Format : NNNNNN. → Numéro séquentiel.

Exemple : 282598.

❖ Numéro du bon de livraison

Type : codification séquentiel.

Format : NNN/AAAA. ↘ Année d'émission de document.



Numéro séquentiel.

Exemple : 568/2011.

❖ Code du bon de réintégration.

Type : codification articulée.

Format : NN/ANN.

Exemple : 20 M00.

❖ Numéro du bon de transfert produit fini

Type : codification articulée.

Format : NAN/NN.

Exemple : 2A0/11.

❖ Numéro du code de bon de chargement

Type : codification articulée.

Format : AA/NNNN.

Exemple : UC 2150.

5) Critiques de l'existant et suggestions :

Après l'étude de l'existant, nous avons constaté certaines insuffisances liées à l'organisation, aux traitements de l'information et sa circulation. Cette étape servira de base à la définition de la solution répondant aux besoins utilisateurs. Nous présentons dans ce qui suit la synthèse des insuffisances constatées ainsi leurs causes générales, et nous apporterons vers la fin nos suggestions.

- ❖ la manipulation fréquente du papier risque de détériorer le stockage des informations sur support .
- ❖ Insuffisance dans la codification utilisée car un seul caractère numérique pour une codification articulé ne pourra pas suffire aussi cette dernière commence toujours par un numéro.
- ❖ Rapidité d'archivage des documents fréquemment utilisés et non utilisés.
- ❖ Lenteur dans la transmission des documents et des informations ce qui provoque une mauvaise mise à jour.
- ❖ La redondance dans la codification des bons.

Suggestions :

- ❖ Codifier les documents d'une manière a ne pas limiter le nombres de documents et remplacer les numéros du début par des caractères pouvant nous aider a mieux reconnaître la famille du produit .
- ❖ Eliminer les rubriques redondantes.
- ❖ Classer des documents utilisés et archiver les non-utilisés.
- ❖ Eviter le stockage des informations sur papiers en tout informatisant.

Pour pallier ses problèmes de gestion classique, nous proposons conjointement avec la direction le développement d'une application capable d'améliorer le résultat net des services concernés en manière de rapidité et d'efficacité.

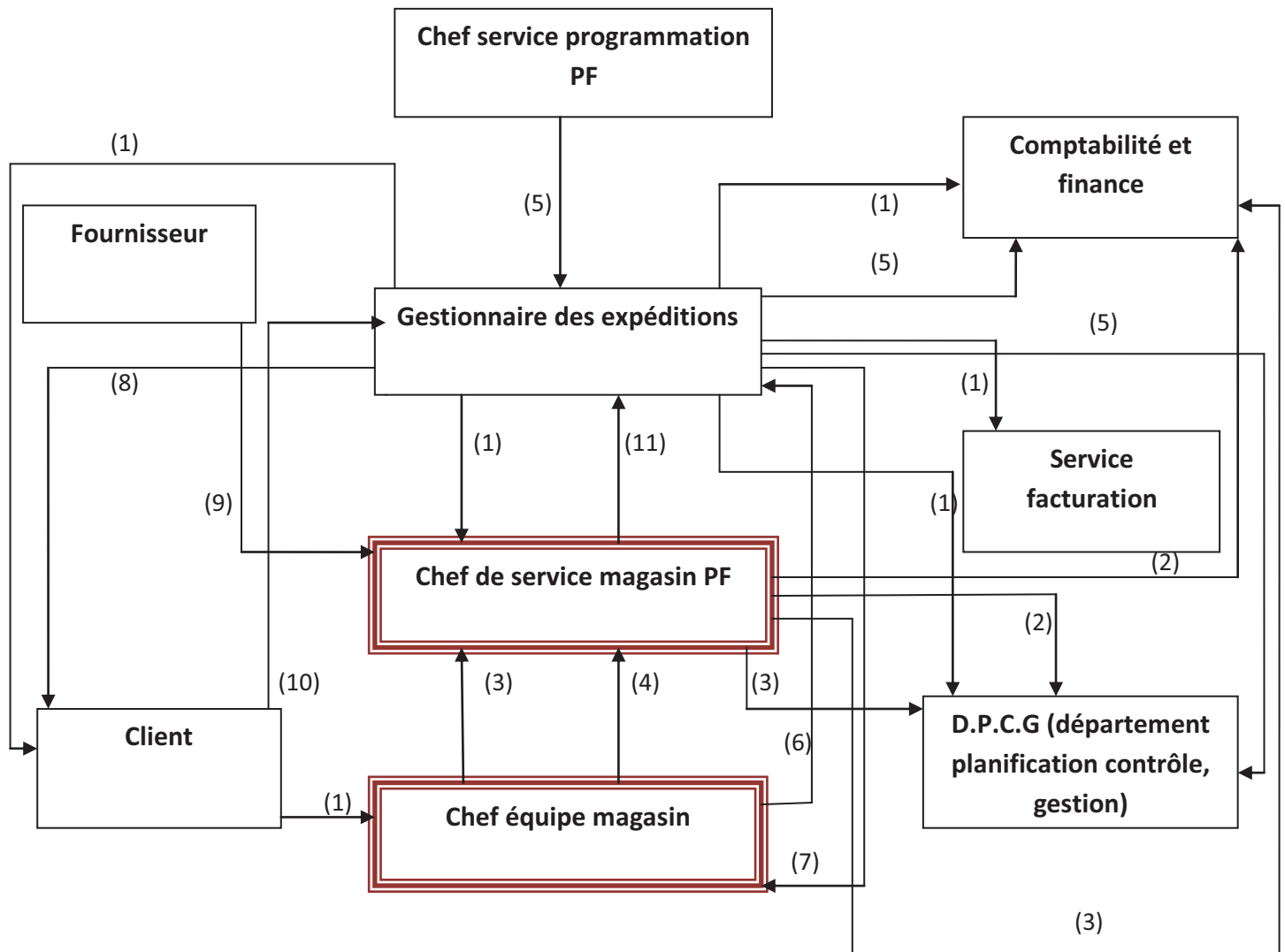
CONCLUSION :

L'étude préalable nous a permis de comprendre le système existant à travers l'étude des différents documents, les postes et les procédures de travail. Au long de cette étape, nous avons recensé les insuffisances informationnelles et organisationnelles du système existant.

Vu l'existence d'un réseau physique intranet à l'entreprise et les échanges d'information entre ce service et d'autres services de l'entreprise, nous avons opté pour la solution qui consiste à mettre en place une application Client/serveur qui sera déployée sur le réseau local existant de l'entreprise .

Chapitre 2

Client / Serveur

Schéma du diagramme des flux existant :

Domaine d'étude.

4) Le diagramme des flux :

Les flux d'information sont des échanges d'informations entre les acteurs internes et avec les acteurs externes au domaine étudié. Un flux entrant dans le système d'éclanche une opération et un flux sortant du système est le résultat d'une opération.

On appelle diagramme des flux , une modélisation qui présente ces flux échangés sans chronologie et sans descriptions des activités associées (en entrée ou en sortie) à ces flux .

4.1) Description des flux :

Numéro	Désignation
1	Bon de livraison
2	Bon de réception
3	Bon de réintégration
4	Bon de reconversion
5	Bon de transfert PF
6	Bon de chargement
7	Bon à servir
8	Bon de sortir produit finit
9	Bon de livraison fournisseur
10	La facture
11	Ordre de remplacement

Introduction :

En général, les serveurs sont des ordinateurs dédiés au logiciel serveur qu'ils abritent, et dotés de capacités supérieures à celles des ordinateurs personnels en termes de puissance de calcul, d'entrées-sorties et de connexion réseau. Les clients sont souvent des ordinateurs personnels ou des appareils individuels (téléphone, tablette), mais pas systématiquement. Un serveur peut répondre aux requêtes d'un grand nombre de clients.

L'environnement **client-serveur** désigne un mode de communication à travers un réseau entre plusieurs programmes ou logiciels : l'un, qualifié de client envoie des requêtes ; l'autre ou les autres, qualifiés de serveurs, attendent les requêtes des clients et y répondent. Par extension, le client désigne également l'ordinateur sur lequel est exécuté le logiciel client et le serveur, l'ordinateur sur lequel est exécuté le logiciel serveur.

L'architecture client /serveur est apparue après un ensemble d'évolutions technologiques ,qui ont été enregistrées dans les dernières années : capacité mémoire , performance des processeurs ,évolution des réseaux et des différents logiciels .

Cette architecture est utilisée d'une manière très large dans plusieurs domaines.

Par conséquent les applications client/serveur sont distribuées sur plusieurs sites d'où la nécessité de les faire communiquer afin qu'elles coopèrent pour la réalisation d'un travail commun.

Dans ce chapitre nous allons proposer les différents architectures du modèle client/serveur ainsi les notions de base nécessaires à la compréhension de ce modèle.

1) Le modèle client /serveur :

1.1) Les techniques de dialogue client/serveur :

Le modèle client/serveur est un mode de dialogue entre deux processeurs, le premier Appelé <<client >>et le second appelé <<serveur>>. La communication client/serveur se fait par la fourniture de service par un processeur serveur à un processeur client, cela est du a la transmission d'une requête à un processus serveur qui exécute l'opération demandée et envoie une réponse en retour . Le programme serveur tourne en permanence attendant des requêtes, et peut répondre à plusieurs clients en même temps.

1.2) Notions de base du modèle client/serveur :

- Client : c'est une entité (processus, programme, ordinateur ...) qui demande l'exécution d'une opération à une autre entité par l'envoi d'un message et attendant la réponse à cette opération par un message en retour .
- Serveur : c'est une entité (processus, programme, ordinateur ...) qui effectue une opération demandé par un client et lui transmette la réponse.
- Requête : Message transmis par un client à un serveur décrivant l'opération à exécuter.
- Réponse : Message transmet par un serveur à un client suite à l'exécution d'une opération demandée par ce client.

La figure si dessous illustre ces notions :

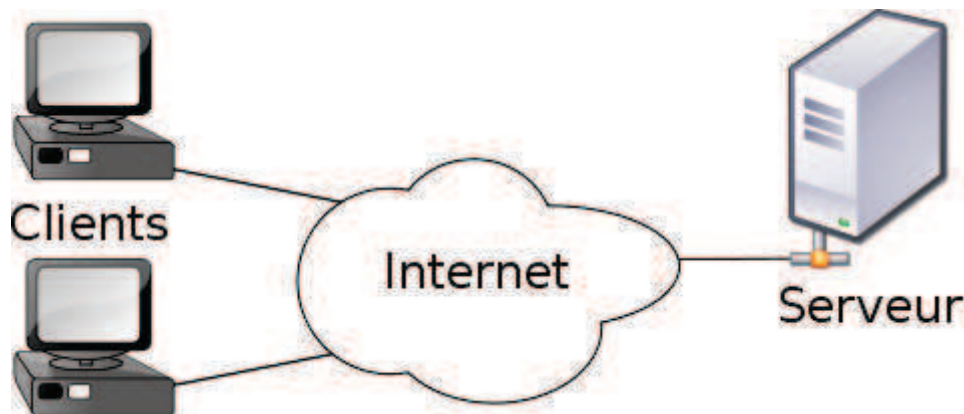


Figure 04 : dialogue client/serveur.

2) MIDDLEWARE :

Le middleware est l'un des composants clé de l'architecture client/serveur , c'est un logiciel assurant la modélisation entre client et serveur dans le cadre d'architecture de système hétérogène .En d'autre terme c'est un ensemble de service logiciels construit au dessus d'un Protocol de transport afin de permettre l'échange des requêtes et des réponses associés entre clients et serveur de manière transparente , permettant de cacher l'hétérogénéité des composants mis en jeu (réseau , SGBD ...).

2-a) Avantages du modèle client/serveur :

Les principaux avantages du modèle client/serveur sont :

- Toutes les données sont centralisées sur un seul serveur, ce qui simplifie
- les contrôles de sécurité, l'administration, la mise à jour des données et des logiciels
- Les technologies supportant l'architecture client-serveur sont plus matures que les autres.
- La complexité du traitement et la puissance de calculs sont à la charge du ou des serveurs, les utilisateurs utilisant simplement un client léger sur un ordinateur terminal qui peut être simplifié au maximum.
- Recherche d'information : les serveurs étant centralisés, cette architecture est particulièrement adaptée et véloce pour retrouver et comparer de vaste quantité d'informations moteur de recherche sur le web, ce qui semble être rédhibitoire pour le P2P beaucoup plus lent, à l'image de Freenet.

2-b) Inconvénients du modèle client/serveur :

Les principaux avantages du modèle client/serveur sont :

- Si trop de clients veulent communiquer avec le serveur au même moment, ce dernier risque de ne pas supporter la charge (alors que les réseaux pair-à-pair fonctionnent mieux en ajoutant de nouveaux participants).
- Si le serveur n'est plus disponible, plus aucun des clients ne fonctionne (le réseau pair-à-pair continue à fonctionner, même si plusieurs participants quittent le réseau).
- Les coûts de mise en place et de maintenance peuvent être élevés.
- En aucun cas les clients ne peuvent communiquer entre eux, entraînant une asymétrie de l'information au profit des serveurs.

3) L'architecture client/serveur multi niveaux :

Les application client/serveur se différencient par la façon dont la communication se partage entre le client et le serveur .le modèle orienté serveur place

plus de fonctionnalités sur le serveur . Tandis que dans le modèle orienté client, le gros de l'application tourne du côté client.

Dans le modèle client/serveur on distingue deux architectures :

Architecture client/serveur à deux niveaux et architectures client /serveur a trois niveau.

3.1) Architecture client /serveur a deux niveaux :

L'architecture a deux niveaux ou a 2-tiers caractérise les systèmes client/serveur dans lesquels le client demande un service au serveur qui le lui rend directement sans faire appel à un autre serveur intermédiaire .cette architecture est simple et populaire grâce a sa souplesse et à la facilité de création d'outils frontaux permettant aux utilisateurs de créer leur application . Comme l'illustre la figure ci-dessous.

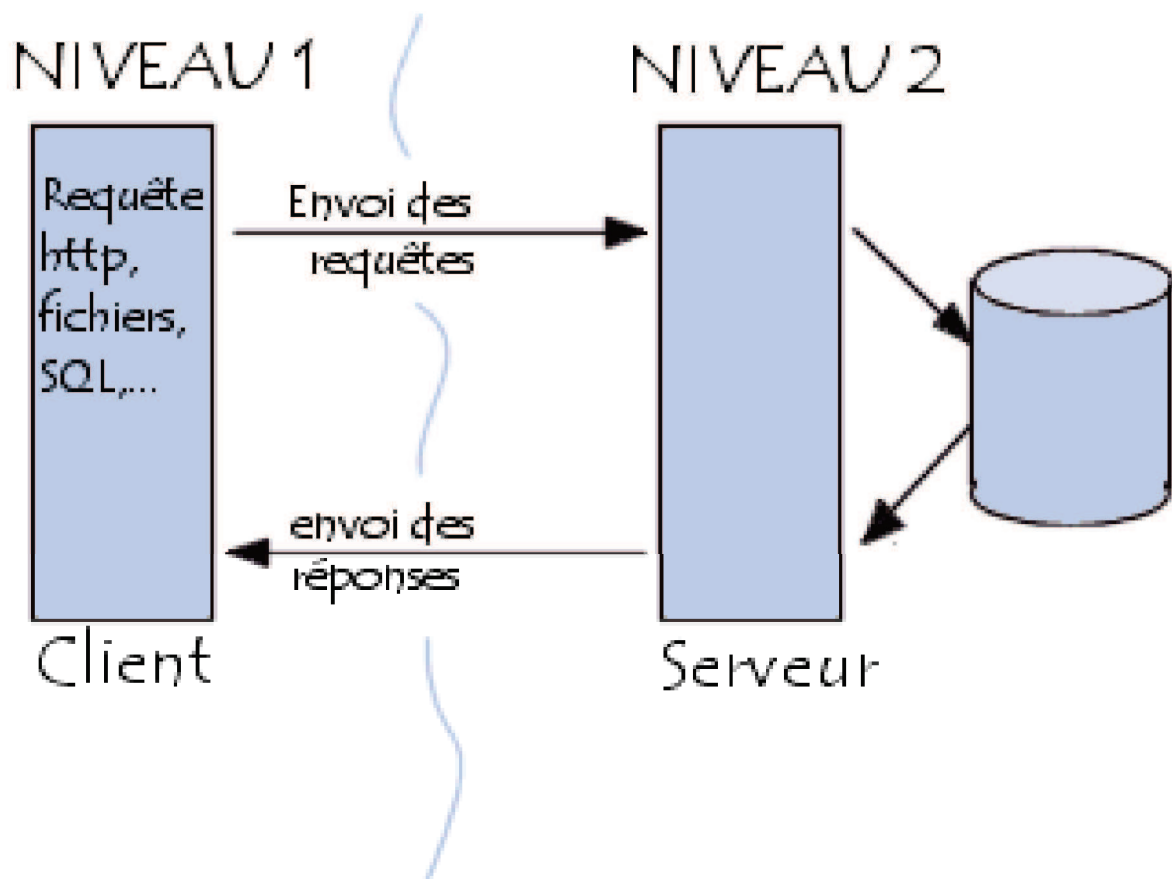


FIGURE 05. Architecture CLIENT/SERVEUR à deux niveaux.

3.2) Architecture client/serveur à trois niveaux :

L'architecture à trois niveaux est aussi appelée architecture à trois tiers. Elle est partagée généralement entre :

1) Client : qui demande la ressource.

2) Serveur d'application (middleware) : qui est chargé de fournir la ressource en faisant appel à un autre serveur.

3) Serveur de base de donnée : qui fournit un service au premier serveur qui est généralement un serveur de base de données.

Les applications de cette architecture à trois niveaux sont plus faciles à déployer et à gérer sur le réseau, elle minimise les charges sur le réseau en créant des niveaux de service.

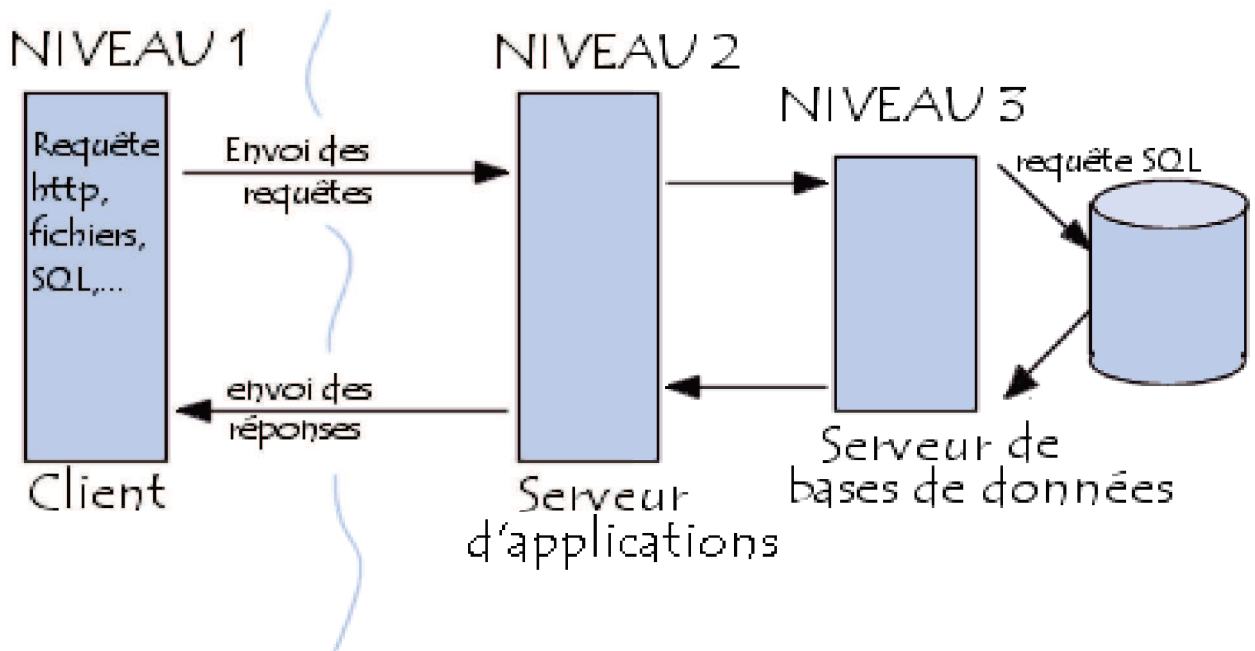


Figure06 : Architecture client/serveur à trois niveaux.

Comparaison entre deux niveaux :

L'architecture à deux niveaux est donc une architecture client/serveur dans laquelle le serveur est polyvalent, c'est-à-dire qu'il est capable de fournir directement l'ensemble des ressources demandées par le client.

Dans l'architecture à trois niveaux par contre, les applications au niveau serveur sont délocalisées, c'est-à-dire que chaque serveur est spécialisé dans une tâche (serveur web/serveur de **base de données** par exemple). L'architecture à trois niveaux permet :

- Une plus grande flexibilité/souplesse ;
- Une sécurité accrue car la sécurité peut être définie indépendamment pour chaque service, et à chaque niveau ;
- De meilleures performances, étant donné le partage des tâches entre les différents serveurs.

3.3) Architecture client/serveur à multi niveaux :

L'architecture à trois niveaux devient de plus en plus à N niveaux, cela est dû au fait que le niveau intermédiaire est implémenté sous forme d'une collection de composants. Chaque composant se charge d'une fonction relativement petite exécutée suite à une demande d'un client, cette dernière peut nécessiter plusieurs composants. D'où le modèle à trois niveaux est en fait à N niveaux.

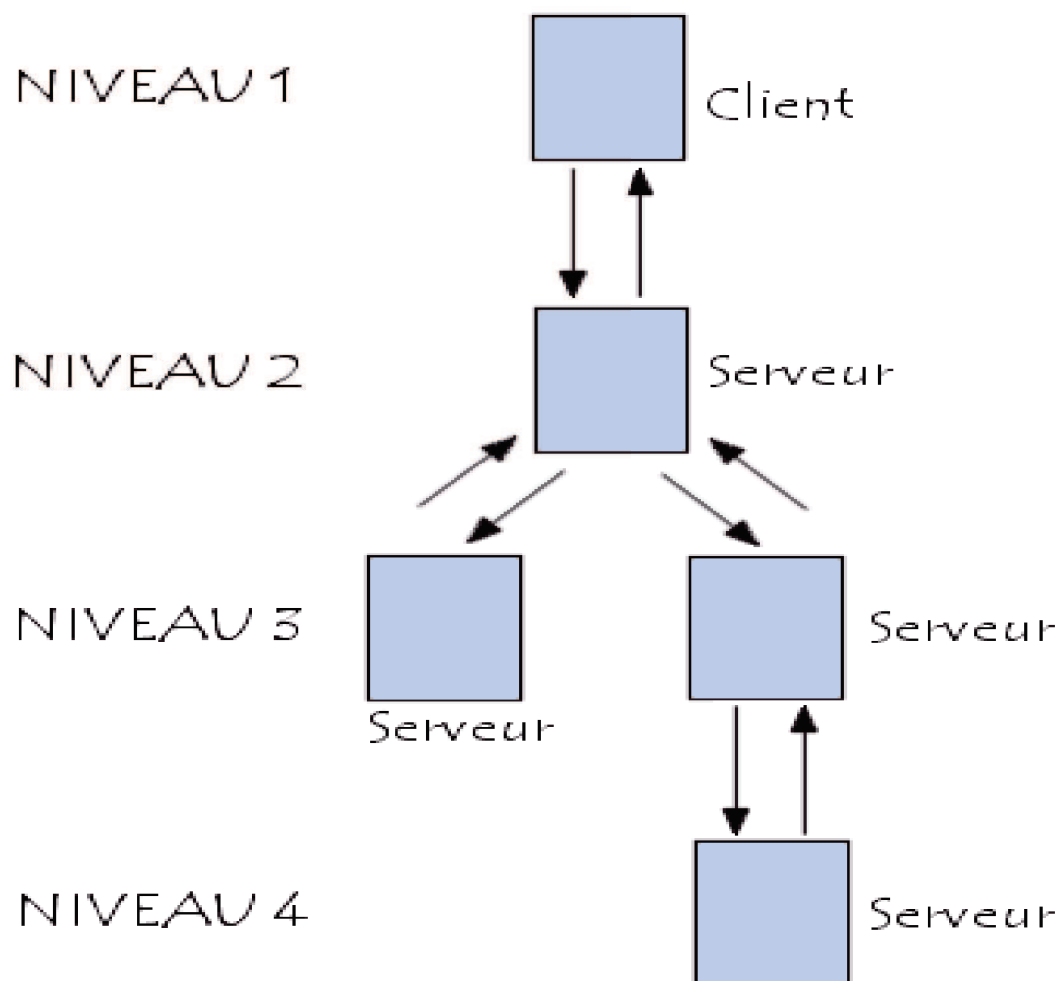


Figure 07 : Architecture client/serveur multi niveaux.

CONCLUSION

Grace à ses différents avantages , le modèle client/serveur est utilisé d'une manière très vaste dans plusieurs domaines du monde informatique , surtout dans les entreprises qui ont comme enjeu de réaliser une intégration de l'informatique personnelle dans le système informatique de l'entreprise avec les objectifs suivants :

- Tout utilisateur dans l'entreprise doit pouvoir accéder à toutes les informations utiles à sa tâche dès lors que celle-ci est autorisée par les règles de confidentialités et de sécurité en rigueur.
- L'accès doit être instantané et doit être possible à partir de n'importe quel poste de travail.
- En plus il doit avoir lieu pour une interface aussi simple que possible.

Pour cela une solution est apportée dans les années 1990. A travers le modèle client/serveur qui met en jeux un processus client qui demande l'exécution d'un service et processus serveur qui réalise ce service.

Chapitre 3

Atelier de

Génie Logiciel

Introduction

Le génie logiciel est l'ensemble des activités de conception et de mise en œuvre des produits et des procédures tendant à rationaliser la production du logiciel et son suivi

Autrement dit, le génie logiciel est « l'art » de produire de bons logiciels, au meilleur rapport qualité/prix.

1) Qu'est ce qu'un atelier de génie logiciel ?

Un AGL (Atelier de Génie Logiciel) ou atelier CASE (Computer Aided Software Engineering) est un logiciel aidant à la réalisation de logiciels. Autrement dit, il s'agit d'un système pour le développement logiciel assisté par ordinateur. Un AGL intègre des outils adaptés aux différentes phases de la production d'un logiciel et facilite la communication et la coordination entre ces différentes phases. Un AGL est basé sur des méthodologies qui formalisent le processus logiciel, et à l'intérieur de ce processus, chacune des phases qui le composent.

Les AGL apportent une réelle solution à certains problèmes du génie logiciel et contribuent nettement à l'amélioration de la productivité et de la qualité du logiciel, notamment en faisant le suivi des différentes phases du processus logiciel et en offrant un cadre cohérent et uniforme de production. Néanmoins, cet enthousiasme doit être modéré: le processus logiciel est encore loin d'être maîtrisé et les différentes formalisations qui en sont proposées font encore l'objet de controverses, et dans tous les cas, sont bien loin d'être totalement automatisables. L'informaticien a encore de belles années de travail devant lui avant d'être supplanté par des AGL...

2) Fonctionnalités d'un AGL

Alors qu'un compilateur, par exemple, ne peut que traduire un langage informatique vers un autre, la réalisation de programmes, notamment au sein d'équipes nombreuses, requiert bien d'autres activités qui sont généralement couvertes par un AGL :

- Conception général du produit, étapes ou phases de réalisation;
- composition et organisation de l'équipe projet;
- calendrier, charges de travail, moyens et budgets;
- conventions de nommage des données et des sous-ensembles de programmes;
- structuration des données;
- aide à l'édition de programmes dans différents langages;
- compilation
- génération de code *optimisé*;
- édition de liens.
- aide aux tests et suivi des corrections;
- bibliothèques de sous-ensembles pouvant être réutilisées dans plusieurs projets;
- documentations
- gestion des versions successives ou des variantes d'un même programme.

Les AGL couvrent donc un champ au-delà des environnements de développements intégrés.

3) Dictionnaire de données

Un AGL est basé sur un dictionnaire de données (qui fait implicitement référence à un méta modèle de données). Il est généralement interfacé avec une interface graphique permettant la modélisation, suivant un ou plusieurs formalismes tels que MERISE ou UML. Parfois, l'outil a l'intelligence pour mettre en forme, et rendre lisible dans un schéma, l'inévitable complexité du « plat de nouilles » issu du dictionnaire de données. Un AGL inclut aussi un système de calcul formel comme Maple, Mathematica ou MuPAD permettant la génération de code optimisé en Fortan, C ou Java

L'AGL permet de « zoomer » et d'enrichir un référentiel d'entreprise suivant plusieurs axes : cycle de vie, domaines projet, cycle d'abstraction, portage technologique sur d'autres environnements de développement...

4) Avantages liés à l'utilisation d'un AGL

Un apport essentiel de l'AGL est de permettre de documenter automatiquement un programme, et de maintenir en permanence à jour cette documentation, et, ce tout au long de sa conception.

L'objectif est souvent d'essayer de trouver le bon niveau de granularité dans la conception de chaque composant applicatif, de manière à essayer d'en améliorer le réemploi.

Certains AGL peuvent aller jusqu'à la génération de code ou à l'inverse peuvent inclure des fonctionnalités de rétro-ingénierie et donc analyser pour modélisation les données contenues dans un programme. Enfin certains peuvent faciliter la génération de jeux de test.

Un AGL facilite la collaboration des différents programmeurs (urbanisme informatique) ainsi que la maintenance ultérieure des programmes en les incitant à partager les mêmes méthodes.

5) Quelques AGL

- AD-Cycle d'IBM,
- AGL Merise/2
- ArgoUML
- ASA & GEODE
- Advantage Plex
- Application Builder d'Enablon
- BOUML
- Dalyo
- DoMIS
- Eclipse
- Envision SART - UML
- Excelerator
- Objecteering
- Optima PLC atelier d'automatisme sur PC (norme IEC 61131-3)
- Pacbase
- Perceptory
- Power AMC de Sybase
- Rational Rose d'IBM
- Visual Studio
- UNIFACE
- WinDev (AGL, UML, FAA, WLangage, Wdscript, gestion de projet, etc.)

5-1) WinDev

WinDev est un atelier de génie logiciel (AGL) édité par la société française PC SOFT et conçu pour développer des applications, principalement orientées données pour Windows 8, 7, Vista, XP, 2008, 2003, 2000, mais également pour Linux, .Net et Java. Il propose son propre langage, appelé le WLangage. La première version de l'AGL est sortie en 1993. Apparenté à Webdev et WinDevMobile

5-1-1) Outils intégrés

L'environnement de développement intégré de WinDev est composé de :

- Éditeur d'analyses et modélisation Merise et UML
- Éditeur d'interface graphique
- Éditeur de dossier technique,
- Éditeur de modèles RAD (également appelés patterns),
- Éditeur de fenêtres,
- Éditeur de chartes graphiques,
- Éditeur de requêtes,
- Éditeur de code,
- Gestionnaire de sources collaboratif
- Éditeur de composants,
- Éditeur d'états et d'étiquettes y compris codes barres,
- Éditeur d'aide en ligne,
- Éditeur d'installation,
- Éditeur de maintenance,
- Éditeur de test de non régression.

WinDev utilise son propre « langage de programmation », le WLangage, ressemblant beaucoup à du pseudo-langage.

L'éditeur d'interface graphique permet de créer des IHM par glisser-déplacer. Il permet également de choisir un modèle de charte graphique parmi un ensemble proposé et d'en créer de nouveaux. Attention cependant, à la différence de la plupart des langages supposés moins évolués, il a longtemps été impossible d'ajouter des champs à partir de rien ou de créer une IHM par pure programmation en WLangage : ceci devait se faire dans l'éditeur d'interface graphique, ou en utilisant un champ invisible hors fenêtre que l'on clonait. À partir de la version 19 les champs peuvent être créés par pure programmation. On peut définir interactivement des « sécurités » de saisie sur

les champs parmi lesquelles : taille, masque, formatage automatique, saisie obligatoire et aide à afficher. Par ailleurs, l'héritage et la surcharge sont gérés.

Bien que prévu initialement pour Windows, WinDev supporte aussi GNU/LINUX pour les applications avec et sans interface graphique, mais aussi MacOSX. Il peut également s'appuyer sur le framework Java pour une partie de ses fonctionnalités, ce qui permet une relative indépendance du fichier exécutable par rapport au système d'exploitation cible.

Ses frères WebDev et WebDevMobile permettent d'utiliser le même langage de programmation (WLangage), et les mêmes concepts (analyse, fenêtre, états, composants, classes...), pour la génération de sites Web et d'applications pour PocketPc, terminaux industriels (Windows CE), smartphones et tablettes (iOS, Android et Windows RT).

WinDev propose également des centres de contrôle intégrés, dont le but est d'organiser et contrôler le développement :

- Centre de contrôle topologique
- Centre de contrôle de planning ;
- Centre de contrôle qualité pour gérer les retours clients (bogues et suggestions, que l'utilisateur final peut envoyer directement via chaque application) ;
- Centre de contrôle des données.

WinDev utilise des formats de fichier propriétaire sous forme binaire. Ce qui implique l'utilisation exclusive du gestionnaire de source maison (GDS) ainsi que le moteur de recherche de l'EDI. Le code ne peut pas être édité via un logiciel classique tel que le bloc-notes et surtout ne peut être mis sous contrôle de version dans un système standard tel que Subversion, Git ou Mercurial, par exemple. En fait, le développeur WinDev est totalement lié à l'environnement de développement vendu par l'éditeur.

5-1-2) Intégration

WinDev permet de lire et aussi de modifier nativement les informations de Excel, Outlook, LotusNet, de l'ERP SAP R/3 (via la technologie BAPI) et depuis la version 14 celles de Salesforce et de diverses applications Google : Maps, Agenda, Contacts, Adwords,... Sont également supportés : ActiveX, appels de DLL, API Windows, OLE, RS 232 et USB, TAPI, SOAP, SNMP, OPC, Flash, Twain,

FTP, Socket, TSE, Citrix, etc. Des fonctions domotiques sont présentes (norme X10).

Pour la manipulation d'un projet, les différents fichiers d'un projet ne peuvent être ouverts qu'avec WinDev (ou WebDev ou WinDev mobile) car ce sont des fichiers binaires et non pas des fichiers texte.

La création d'application multilangue est prise en compte automatiquement.

WinDev permet de s'interfacer avec du code C++, Cobol, Fortran, Pascal.

5-1-3) Prise en charge des bases de données

WinDev gère de nombreux systèmes de gestion de bases de données, que ce soit par l'intermédiaire des protocoles ODBC ou OLEDB ou par accès natif. Il existe un accès natif gratuit pour MySQL, pour PostgreSQL (depuis la version 14) et pour SQLite (depuis la version 16). D'autres modules payants permettant l'accès natif à ORACLE, SQLserver, Sybase, AS /400, DB2, Informix et Progress 4GL

Il existe par ailleurs des accès natifs libres et gratuits comme MySQL4WD et PostgreSQL4WD de Rodolphe Jouannet qui offrent un accès natif à MySQL et PostgreSQL Le module libre et gratuit SQLManagerX de Frederic Emprin permet d'écrire un seul code unifié pour gérer les différents systèmes de gestion de base de données. Par l'ajout d'autres modules, SQLManagerX offre un accès natif à un grand nombre de bases de données

WinDev comporte aussi son propre système de gestion de base de données appelé HyperFileSQL (anciennement nommé HyperFile). Il existe en version locale, client/serveur sous Windows, Linux et mobile. Les versions sont compatibles entre elles.

5-1-4) Les fonctionnalités automatiques des applications développées

Les applications développées avec WinDev bénéficient par défaut de fonctionnalités automatiquement proposées à l'utilisateur final, qui sont néanmoins désactivables par le concepteur. Ces fonctionnalités sont : export des tables vers Word, Excel, Open Office et XML, création de fichier PDF,

possibilité de saisir des macros en code WLangage, possibilités de tri et recherche directe dans les tables, possibilité de mémoriser les valeurs saisies dans un champ de saisie et de les choisir à nouveau (persistance des données), correction orthographique en temps réel (le dictionnaire de OpenOffice.org doit être installé), historique des saisies sur des champs spécifiés, affichage d'un graphique sur une série de données sélectionnées, agrandissement de la taille d'une combo, ajout de somme, moyenne et comptage dans une colonne et le positionnement d'une minuterie sur un bouton.

À partir de la version 10, l'éditeur d'états et de requêtes peut être installé gratuitement sur le poste de chaque utilisateur final, ce qui permet à chaque utilisateur de créer librement ses propres états, étiquettes et requêtes.

5-1-5) Les extensions de fichiers

- NDX : Index HyperFileSQL, permet de lister les catégories des données HyperFileSQL dans WinDev, c'est comme un index dans un livre pour pouvoir trouver la bonne page. Ce fichier est généré automatiquement par l'analyse lors de la création d'une application WinDev.
- MMO : Mémo HyperFileSQL
- FIC : Données HyperFileSQL
- FTX : Index full-text HyperFileSQL (permet une recherche rapide sur plusieurs mots, et obtenir un résultat selon la pertinence)
- WDC : Classe : WinDev permet en effet de générer des classes sur base des principes de la programmation orientée objet
- WDE : État
- WDG : Collection de procédures globales
- WDH : Aide
- WDK : Composant

2) BOUML

Est un logiciel de création de diagrammes UML. Programmé en c++et Qt. Auparavant distribué gratuitement sous licence GPL, il est commercialisé depuis la version 5.0. Il est multilingue, supporte la génération de code et la rétro-ingénierie.

- Permet de dessiner des diagrammes suivant la norme UML 2.0.
- Est capable d'effectuer de la rétro-ingénierie.
- Est multiplateforme.
- Peut être utilisé en français (depuis la version 4.15.1).

En 2007, des analyses de l'outil pointent quelques défaillances -corrigées depuis et concernant en fait l'export XML et non le « reverse »

En 2010, il est le seul outil libre supportant à la fois UML 2.0, les diagrammes d'activité et l'exportation au format XML

BOUML est mentionné en publications scientifiques

Génération de code et rétro-ingénierie

L'ergonomie et la richesse graphique de BOUML sont parfois limitées, tout au moins en comparaison de modeleurs UML plus conventionnels comme Start UML ou Argo UML. Par contre, il est jugé extrêmement efficace — et surpasse dans ce domaine la plupart des autres gratuits UML — pour la « rétro-modélisation » (créer un modèle UML à partir de codes sources) et pour le développement roundtrip (faire des aller-retour entre modèle UML et code source). BOUML permet ainsi de générer automatiquement du code à partir des diagrammes de classes UML, ainsi que de reconstruire ces derniers à partir de sources existantes, pour les langages C++Java, PHP et MYSQL. Pour les langages Python et IDL, il peut générer du code mais ne permet pas la rétro-conception UML.

Changement de licence

En Septembre 2010 Bruno Pagès, auteur du logiciel, annonce qu'il arrête définitivement de développer BOUML suite à un conflit sur Wikipédia. Une version « ultime » est néanmoins mise en ligne en novembre 2010, suivie par des patches correctifs en 2011. Suite à la sortie des patches correctifs, l'auteur décide de porter le projet en Qt version 4 et de changer la licence.

Le logiciel est disponible en version payante en février 2012. Un logiciel de visualisation gratuit est disponible.

3) *Rational Rose*

Est un logiciel édité par l'entreprise *Rational Machines* (plus tard renommée *Rational Software*) pour créer et éditer les différents diagrammes du modèle UML (Unified Modeling Language) d'un logiciel.

Rational Software a été vendu pour 2,1 milliards de dollars à IBM le 20 février 2003.

Rational Rose permet également de sauvegarder et d'imprimer ces diagrammes, ainsi que de générer le code source ou C++ qui leur correspondent.

4) *Produits Uniface*

La plateforme Uniface offre une gamme d'outils de développement et de déploiement, dont la plupart est regroupée dans l'Uniface Application Platform Suite (APS). L'APS comprend:

- Uniface : environnement de développement. C'est un ensemble intégré d'outils pour la modélisation, la mise en œuvre, la compilation, le débogage et la distribution d'applications.
- Uniface Flow : environnement pour développer et déployer des processus métier.
- Uniface View : portail d'intégration d'applications qui fournit un bureau organisé de documents Web, des rapports de base de données, des applications d'entreprise, et d'autres ressources de l'entreprise.

Les applications Uniface (y compris l'environnement de développement, Uniface Flow et Uniface View) utilisent une infrastructure d'exécution commune, composée de :

- Uniface Runtime Engine : une machine virtuelle spécifique à la plateforme qui interprète et exécute les composants d'application et les bibliothèques.
- Uniface Router : un processus multi-thread responsable de la communication interprocessus dans les applications Uniface. Il démarre et arrête Uniface Server. Il effectue l'équilibrage des charges (load balancing), et transmet les messages entre les différents processus Uniface.

- Uniface Server : un processus qui permet aux clients d'accéder aux ressources Uniface à distance ou d'exécuter des composants distants. Il agit comme un serveur d'applications, un serveur de données et un serveur de fichiers.
- Référentiel Uniface : une base de données relationnelle utilisée pour stocker les définitions et propriétés des objets de développement, des modèles de processus et de l'organisation, et les définitions du portail.
- Uniface Web serveur : le serveur Apache Tomcat est retenu pour développer et tester des applications Web, en production. N'importe quel serveur web peut être utilisé.
- Servlets Java : servlets utilisées pour la gestion des échanges de communication entre le serveur web et le serveur d'applications Uniface pour les interfaces Web et les Web services.
- Connecteurs de base de données : Pilotes qui gèrent les connexions entre Uniface et une grande variété de bases de données.
- Les outils d'intégration : Pilotes, composants et API qui gèrent la communication entre Uniface et des applications tierces et/ou des technologies différentes : Java, CICS, IMS, LDAP, SMTP/POP, des commandes du système, COM etc.

En outre, Uniface JTi (Java Thin Client Interface) permet d'accéder à des applications client/serveur Uniface depuis le web.

Uniface possède un langage de procédure propriétaire appelé "Proc" qui est utilisé pour coder le comportement des applications. Le noyau d'Uniface et le code par défaut automatisent la plupart des opérations d'entrée/sortie. Si bien que les opérations de base n'ont pas besoin d'être codées.

Applications Uniface

Les applications Uniface sont basées sur des composants logiciels indépendants de l'infrastructure, des bases de données et des systèmes de fichiers. Elles peuvent être des applications composites qui comprennent des éléments non-Uniface créés à l'aide d'autres outils de développement, elles peuvent être déployées dans les environnements client/serveur, web, mobiles, services web, et mainframe. Uniface comprend plusieurs de types de composants qui sont

destinés à être utilisés dans les différentes couches d'une architecture d'application multi-tiers.

Les composants pour la couche de présentation sont responsables de l'interface utilisateur, et comprennent:

- Les "Form" : Écrans ou formulaires interactifs pour l'affichage et la mise à jour des données dans un environnement client/serveur.
- Les "Server Page" : Pages HTML interactives pour l'affichage et la mise à jour des données dans un environnement web.
- Les "Report" : Mise en page pour la présentation des données en vue d'une impression.

Les composants pour la couche logique métier traitent des règles commerciales et du comportement spécifiques lié à une tâche et n'ont pas d'interface utilisateur:

- Les "Services" : Assurent le traitement et la logique métier lorsqu'ils sont appelés par d'autres composants, que ce soit localement ou à distance.
- Les "Session Services" : Centralisent les règles métier complexes qui touchent plusieurs entités de données, tels que les comportements spécifiques d'une tâche, les transactions et l'intégrité référentielle.
- Les "Entity Services " : Centralisent les règles liées à une seule entité de données.

Le tiers d'accès aux données contient les structures de base de données physiques définies dans le modèle d'application Uniface. Uniface assure l'accès physique aux données en encapsulant les ordres SQL dans ses connecteurs SGBD. La gestion réseau et les accès aux middlewares sont encapsulés par les pilotes middleware et le routeur Uniface. Le moteur d'exécution exécute les composants de l'application. Il affiche les composants de présentation en utilisant le connecteur d'interface utilisateur approprié (soit l'interface graphique, ou en mode caractère), et envoie ou reçoit les données via un connecteur SGBD.

Le développement d'applications

Avec Uniface, le développement est guidé par le modèle et se conçoit à partir de composants. La structure de données, les règles métier et le comportement par défaut de l'application sont définis dans le modèle d'application. La définition des modèles peut être réutilisée et héritée par les composants, qui

ensuite peuvent remplacer les définitions héritées et fournir un comportement spécifique. La création de « Gabarits » ou « Template » permet d'améliorer la productivité et de garantir la cohérence dans la définition des modèles.

Modèle d'application

Le modèle d'application définit les entités (tables), les attributs, les clés primaires, les indices et les associations entre les relations pour définir l'intégrité référentielle. Chaque entité et chaque attribut dans le modèle ont des propriétés et un ensemble de déclencheurs. Les règles métier sont ajoutées au modèle de façon déclarative en définissant les propriétés et le traitement associé, en ajoutant du code Proc (langue de procédure d'Uniface) dans les déclencheurs. Les déclencheurs sont des conteneurs pour le code. Ils représentent des événements activés par l'utilisateur ou le système. Exemples : "Occurrence Get Focus", "Read" ou "Leave Field". D'autres portent sur des traitements tels que la validation, ou agissent comme conteneur pour les méthodes associées à l'objet. L'utilisation de déclencheurs au niveau du modèle permet de recueillir les propriétés et le comportement métier des objets. La séparation entre structures logiques et données physiques permet de définir le comportement par défaut une seule fois et de le réutiliser systématiquement. Ce procédé accélère le développement et facilite la réalisation d'applications n-tiers.

Composants

Les objets décrits dans le modèle d'application sont réutilisés par les composants. Les développeurs incorporent des objets du modèle sur la grille canevas de constitution de l'écran représentant la mise en page (pour les composants de présentation), ou en les insérant dans une arborescence définissant la structure des composants. Ils peuvent également, au niveau des composants, ajouter des objets qui ne sont pas dans le modèle d'application, tels que des champs de contrôle, des menus, et des variables de composant. Les propriétés et les déclencheurs définis dans le modèle d'application sont hérités par copie dans le composant. Les définitions peuvent être modifiées au niveau des composants pour fournir des fonctionnalités spécifiques. Cela rompt le lien entre le modèle d'application et le composant (même s'il est possible de restaurer le lien vers le modèle). Si le code ou les propriétés sont modifiés au niveau du modèle, tous les composants utilisant ces spécifications seront mis à jour lors de leur prochaine compilation. Cette propagation facilite la maintenance et assure que les règles associées à l'objet sont disponibles partout où il est utilisé.

Le Référentiel d'Uniface

Uniface maintient une base de données de ses propres métadonnées afin de réutiliser les modèles d'application, les définitions de composants, les schémas de composants, du code de procédure, et ainsi de suite. Le référentiel est propriétaire et dédié à l'environnement de développement Uniface, qui en assure l'intégrité. Cependant, la structure du référentiel est documentée, ce qui rend possible (mais non recommandé) d'interagir directement avec lui, pour des rapports par exemple. Le référentiel peut être partagé par des équipes de développeurs. De plus, Uniface s'intègre à n'importe quel système de contrôle de code source (SCCS) prenant en charge les spécifications du Microsoft Common Source Code Control Interface. Les fonctionnalités disponibles dans SCCS d'Uniface dépendent du SCCS utilisé, les éditeurs de logiciels ont interprété et mis en œuvre les spécifications du Microsoft Common Source Code Control Interface différemment. En conséquence, les fonctionnalités et le comportement pris en charge varient en fonction du fournisseur du CSSC.

Déploiement d'applications

Les applications Uniface peuvent être déployées sans changement de code sur des plateformes aussi différentes que le « mainframe » et le « mobile ». Les composants (et d'autres objets tels que les scripts de démarrage, les menus, les barres d'outils (panneaux), les glyphes, les variables globales et les « include proc ») sont compilés et produisent des objets exécutables qui peuvent être empaquetés dans des fichiers zip et déployés sur n'importe quelle plateforme. Les objets exécutables sont activés à l'aide d'une machine virtuelle dédiée à chaque plateforme (Java et d'autres langages ont, plus tard, suivie Uniface à cet égard). Les composants peuvent être compilés dans un environnement et exécutés dans un autre.

Connexion aux bases de données

Uniface permet l'accès à de nombreuses bases de données et systèmes de fichiers par le biais de connecteurs (ou drivers). Les connecteurs SGBD convertissent les données Uniface au format le plus approprié en fonction de la nature du SGBD. À l'exécution, il est possible de passer des paramètres pour invoquer (ou désactiver) des extensions spécifiques de la base de données. Il est possible de convertir des données provenant de différents systèmes de bases de données. Uniface fournit également une API qui peut être utilisée pour créer des connecteurs de base de données personnalisée ou spécifiques.

NetBeans

NetBeans IDE est un environnement de développement intégré (EDI) de premier ordre pour Windows, Mac, Linux et Solaris. Le projet NetBeans consiste en un EDI Open Source et en une plate-forme d'application permettant aux développeurs de créer rapidement des applications Web, d'entreprise, de bureau et mobiles à l'aide de la plate-forme Java, ainsi que de JavaFX, PHP, JavaScript et Ajax, Ruby et Ruby on Rails, Groovy et Grails et C/C++.

Le projet NetBeans est soutenu par une communauté de développeurs dynamique et propose différentes ressources de documentation et de formation, ainsi qu'un choix varié de plug-ins tiers.

NetBeans est aussi une plate forme générique pour le développement d'applications pour stations de travail (bibliothèque Swing (Java)). Elle fournit des ressources pour développer les éléments structurants de ces applications: gestion des menus, des fenêtres, configuration, gestion des fichiers, gestion des mises à jour... Des présentations détaillées sont fournies par le centre de documentation de NetBeans.

L'IDE NetBeans comprend toutes les ressources utiles mais il est aussi possible d'installer la plate forme séparément.

Le développement d'applications sur la base de la plate forme Netbeans consiste en la réalisation de "modules" qui s'insèrent dans la plate forme et en étendent dynamiquement les fonctions.

Un module est un groupe de classes Java classes, de portée variée: elle peut consister en une simple classe Java réalisant des fonctions simples (exemple: ajouter une action dans un menu pour éditer le contenu du presse papier) comme elle peut intégrer une application externe complète (exemple: Java profiling suite). Un module peut s'appliquer à l'IDE NetBeans lui-même.

Comparaison entre les différents AGL

Nom	Créateur	Plateforme / OS	Dernière diffusion stable	Open source ?	Licence logiciel	Langage de programmation utilisé
AmaterasUML		Multiplate-forme (Java)	16 juillet 2012	Oui	EPL v1.0	Java
ArgoUML	Tigris.org	Multiplate-forme (Java)	15 décembre 2011	Oui	EPL v1.0	Java

Nom	Créateur	Plateforme / OS	Dernière diffusion stable	Open source ?	Licence logiciel	Langage de programmation utilisé
Astade		Multiplate-forme	11 mars 2010	Oui	GPL	Basé sur wxWidgets
BOUML (version < 5.0 non maintenu)	Bruno Pagès	Multiplate-forme	1er octobre 2011	Oui	GPL	C++/Qt3
BOUML (version >= 5.0)	Bruno Pagès	Multiplate-forme	15 décembre 2012	Non	Commercial	C++/Qt4
Eclipse UML2 Tools	Eclipse Foundation	Multiplate-forme (Java)		Oui?	GPL?	Java
Enterprise Architect	Sparx Systems	Microsoft Windows, Linux et Mac OS X ¹	19 septembre 2013	Non	Commercial	C++
eUML2 Studio	Soyatec	Multiplate-forme (Java / Eclipse)	31 août 2012	Non	Free, Commercial	Java
Frame UML	Frame	Windows	27 février 2009	Oui	GPL V3	C++
Jink UML	Nether	Multiplate-forme (Java)	27 avril 2009	Oui	MIT	
Modelio	Modeliosoft	Windows, Linux, Mac OS	19 mai 2014	Oui	GPL V3, Apache 2.0	Java, C++
MOSKitt	Valencian Regional Ministry of Infrastructure and	Multiplate-forme (Java)	8 novembre 2012	Oui	EPL	Java

Nom	Créateur	Plateforme / OS	Dernière diffusion stable	Open source ?	Licence logiciel	Langage de programmation utilisé
	Transport					
Objecteering	Objecteering Software	Windows, Linux	25 novembre 2009	Non	Commercial	
Open ModelSphere	Grandite	Multiplate-forme (Java)	13 avril 2012	Oui	GPL	Java
PowerAMC	Sybase	Windows	1er octobre 2008	Non	Commercial	
Rhapsody	Telelogic racheté par IBM	Windows	16 décembre 2011	Non	Commercial	C, C++, Java, C#, Ada
RTDS	PragmaDev	Windows, Linux, Solaris	6 avril 2012	Non	Commercial	C, C++
Software Ideas Modeler	Dusan Rodina	Windows (.NET), Linux (Mono)	20 janvier 2013	Non	Freeware	C#
StarUML	Plastic Software	Windows	30 décembre 2005	Oui	GPL, modifié	Delphi
Topcased	Consortium	Multiplate-forme (Java)	9 mars 2010	Oui	EPL	Java
Umbrello UML Modeller	L'équipe Umbrello	Linux	4 août 2009	Oui	GPL	C++, KDE
UML Designer	Obeo	Multiplate-forme (Java / Eclipse)	23 juin 2013	Oui	EPL	Java

Nom	Créateur	Plateforme / OS	Dernière diffusion stable	Open source ?	Licence logiciel	Langage de programmation utilisé
UmlDesigner	Green Bird Software	Windows		Non		C#
Violet UML Editor	Cay S. Horstmann, Alexandre de Pellegrin (et autres)	Multiplate-forme (Java / Eclipse)	6 février 2013	Oui	Libre	Java
Visual Paradigm for UML	Visual Paradigm Int'l Ltd.	Multiplate-forme (Java)	21 janvier 2013	Non	Commercial avec une édition communautaire gratuite	Java
WhiteStarUML	Janusz Szpilewski	Windows	9 juin 2013	Oui	GPL, modifié	Delphi
WinA&D	Excel Software	Windows, Mac OS X(MacA&D)		Non	Commercial	
Innovator Modeling Platform	MID GmbH			Non	Commercial	
Nom	Créateur	Plateforme / OS	Dernière diffusion stable	Open source ?	Licence logiciel	Langage de programmation utilisé

Comparaison en terme de Fonctionnalités

Nom	UML 2 ?	MDA ?	XMI ?	Patrons ?	Langages générés	Langages de rétro-génération	Intégrable dans	Détails
-----	---------	-------	-------	-----------	------------------	------------------------------	-----------------	---------

Nom	UML 2 ?	MDA ?	XMI ?	Patrons ?	Langages générés	Langages de rétro-génération	Intégrable dans	Détails
Acceleo	?	Oui	?	Oui	JEE, C#, Java, PHP, Python.		Eclipse, EMF	
AmaterasuUML	?	?	?	?	Java	Java		Standard UML
ArgoUML	Oui	Oui	Oui	?	C++, C#, PHP4, PHP5, Java, Ruby, SQL	Java (et autres langages, dont C# en plugins)		Suit de près les standards UML
Astade	?	?	?	?	C, C++			
ATL	?	?	?	?			Disponible via le projet Eclipse M2M (Model to Model).	Peut transformer des modèles UML en d'autres modèles
BOUML	Oui	Oui	Oui	Oui	Java, C++, PHP, Python, IDL et MySQL.	Java, C++, PHP et MySQL.	plug-in : C++ QT4 or Java	
Dia	?	Non	Non	Non	Java, C++, ADA (en utilisant dia2code), PHP (en utilisant uml2php5)			
Eclipse UML2 Tools	Oui	?	?	?	Java	Java	Eclipse	5 types de diagrammes.
Enterprise Architect	Oui	Oui	Oui	OUI Langage de Transformati	ActionScript, C#, C++, Corba IDL,	ActionScript, C#, C++, Corba IDL,	Visual Studio 2005 et	Technologies de modélisation

Nom	UML 2 ?	MDA ?	XMI ?	Patrons ?	Langages générés	Langages de rétro-génération	Intégrable dans	Détails
				on propriétaire	Delphi, Java, PHP, Python, Visual Basic 6 et Visual Basic .NET	Delphi, Java, PHP, Python, Visual Basic 6 et Visual Basic .NET	Eclipse	et extension avec des plugins.
eUML2 Studio	Oui	Oui	Oui		Java 1.4, 1.5 ou +	Java	Eclipse, EMF, GEF, GMF, SWT, UML2	Synchronisation temps réel entre le Modèle et Java, Analyse de dépendance,
Frame UML	Oui	?	?	?	N'importe quel langage qui peut être généré par du JavaScript.	Java (partiel), mais vous pouvez rétro-générer d'autres langages avec du JavaScript.		
Jink UML	?	?	?	?				
Modelio	Oui	Oui	Oui	Oui	Java, C#, C++, XSD, WSDL, BPEL, SQL	Java, C++, C#, SQL	Eclipse, EMF	Supporte tout l'UML2. Intègre BPMN. Génération de document HTML, MS-Word, OpenDocument. Modélisation et application de Design

Nom	UML ?	MDA ?	XMI ?	Patrons ?	Langages générés	Langages de rétro-génération	Intégrable dans	Détails
								Patterns.
MOSKitt	?	Oui	?	?	N'importe quel langage textuel résultant d'une transformation Modèle Vers Texte.		Eclipse, EMF, GMF, SWT, DataTools ...	Outil servant de support à la méthodologie de processus de développement de logiciel gvMétrica. Transformation de modèle, traçabilité, synchronisation, édition de modèle par forme ou graphiquement.
Objecteerin g	Oui	Oui	Oui	?	Java, C++, C#, SQL DDL, CORBA IDL et Fortran		Eclipse	
Open ModelSphere	Non	Non	Non	Oui	Java, DDL	Java		Modélisation de données, modélisation de processus métier, conception et rétro-ingénierie. Architecture en plug-in.
PowerAMC	Oui	Oui	Oui	Oui	Java, C# et VB .NET		Eclipse	Modélisation de données, modélisation de processus

Nom	UML ?	MDA ?	XMI ?	Patrons ?	Langages générés	Langages de rétro-génération	Intégrable dans	Détails
								métier, conception et rétro-conception.
Rhapsody	Oui	Oui	Oui (XMI Toolkit)	?	C, C++, Java, C#, Ada	C, C++, Java, C#, Ada	Eclipse, VxWorks workbench	Modélisation de données, conception et rétro-conception, gestion des exigences, roundtrip.
RTDS	?	?	Import	?	C, C++	C, C++	RTOS: VxWorks, FreeRTOS, ThreadX, CMX, OSE Delta, OSE epsilon, uITRON 3, uITRON 4, Nucleus, posix, win32. Débogueur: gdb (GNU Debugger), Multi (Green Hills Software), Tornado (Wind River), Multi (Green	Profil pour les applications temps réel embarquées basé sur le standard de l'ITU-T SDL et SDL-RT. Intègre le standard de test TTCN-3.

Nom	UML ?	MDA ?	XMI ?	Patrons ?	Langages générés	Langages de rétro-génération	Intégrable dans	Détails
							Hills Software), Tasking Cross View Pro (Altium).	
Software Ideas Modeler	Oui	?	Oui	?	C#, VB.NET et SQL DDL			UML, diagramme de flux de données, CRC, ERD et diagramme mixte.
StarUML	?	Oui	Import	?	java, C++, C#, PHP		C#	Architecture en plugin : C++, Delphi, C#, VB. Export : doc, excell, ppt, txt. Plus maintenu depuis 2005.
Topcased	Oui	Oui	Oui	?	Java, Python, C++		Eclipse, EMF	Un outil complet et puissant où vous pouvez écrire des transformations de modèles, exécuter des simulations de diagramme d'état, générer du code et de la rétro-

Nom	UML 2 ?	MDA ?	XMI ?	Patrons ?	Langages générés	Langages de rétro-génération	Intégrable dans	Détails
								conception. Inclut le langage de modélisation SysML.
UML Designer	Oui	Oui	Oui	Oui (via ATL)	via Acceleo : Java, JavaEE, C/C++, et n'importe quel langage textuel	via tout outil de rétro-génération UML compatible EMF	Eclipse	Supporte le format Eclipse UML2 nativement, et la majorité des diagrammes UML2. Possibilité de créer ses propres diagrammes.
UmlDesigner	?	?	?	?	C#			
Umbrello UML Modeller	?	?	Import	?	16 langages : C++, Java, Perl, PHP, Python...	C++, IDL, Pascal/Delphi, Ada, Python, Java; import XMI et RoseMDL	KDE	
Violet UML Editor	Non	Non	Non	?			Eclipse	Outil simple et rapide d'utilisation, loin de vouloir être complet.
Visual Paradigm for UML	Oui	Oui	?	?	Java, C#, C++, PHP, Ada et Action Script	Java, C# (binary), C++ et PHP	Eclipse, NetBeans and IntelliJ	Support complet de l'UML 2.1, SysML, ERD, BPMN,

Nom	UML ?	MDA ?	XMI ?	Patrons ?	Langages générés	Langages de rétro-génération	Intégrable dans	Détails
								modélisation de données, modélisation de métier et rétro-conception à partir de code source et de base de données.
WinA&D	?	?	?	?	C, Pascal, Basic, Fortran, Delphi, Java, C#, C++, Objective-C, PHP, Ada et SQL	C, Pascal, Basic, Fortran, Delphi, Java, C#, C++, Objective-C, PHP, Ada et SQL	Générateur de rapport scriptable intégré et extension d'outil	Support complet de l'UML, SA/SD, ERD, Task et support de modèle de système
Nom	UML ?	MDA ?	XMI ?	Patrons ?	Langages générés	Langages de rétro-génération	Intégrable dans	Détails

CONCLUSION

Depuis sa naissance à la fin des années soixante, le génie logiciel a connue beaucoup de développement et d'évaluation. Les outils de développements, les méthodes et les techniques du génie logiciel évoluent rapidement et ont un cycle de vie très court

Du fait de l'importance de maîtrise des coûts de production qui sont souvent imprévisibles un certain nombre de méthodes et techniques permettant d'estimer le coût d'un produit logiciel ont été mises à la disposition des responsables de projet informatiques.



Analyse
Et
Conception

Introduction

Pour le développement d'une application, nous devons suivre une méthode ou une démarche précise afin d'assurer une meilleure organisation et une bonne maîtrise du travail.

Dans ce chapitre nous avons mis en évidence les différents acteurs intervenant dans le système. La phase de conception se base sur les résultats obtenus pendant la phase d'analyse nous utilisons le langage UML (Unified Modeling Language) dans cette page de conception.

Analyse :

Dans cette partie nous allons spécifier l'application de la gestion des stocks d'une manière détaillée. Pour cela nous allons procéder en premier lieu à la détermination d'une manière globale ce qui se trouve dans le champ de cette application.

Le niveau de donnés :

Ce niveau concerne l'organisation conceptuelle, logique et physique des données manipulées.

Durant la partie d'analyse nous avons pu identifier les données nécessaires au bon fonctionnement de l'application et à travers la conception du niveau applicatif , nous allons dégager la classe significative. À partir de là nous pourrons faire la conception de la base de donnée.

Nous avons adopté la démarche suivante pour la conception :

- Identification des acteurs et des besoins.
- Elaboration du diagramme du contexte du système à étudier.
- Identification et représentation des cas d'utilisation.
- Elaboration des diagrammes de séquence.
- Elaboration des diagrammes d'activité.
- Elaboration des diagrammes de classe.

La figure suivante donne la représentation graphique de la démarche de modélisation suivie pour concevoir l'application.

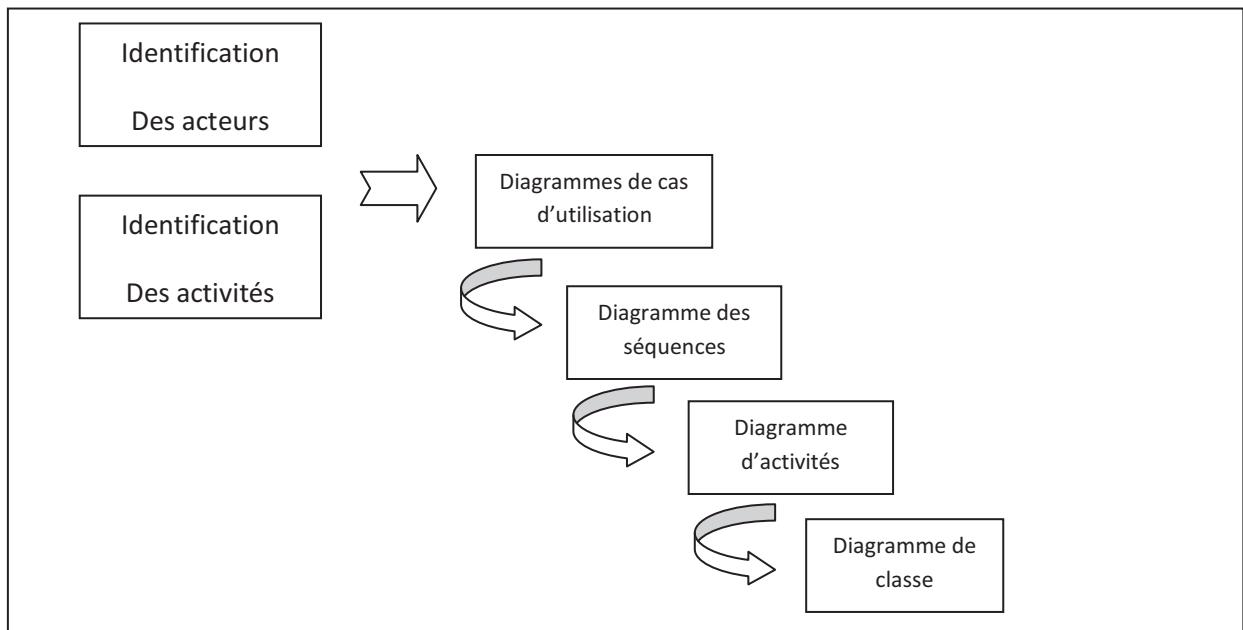


Figure N°08 : Représentation graphique de la démarche de modélisation de l'application.

Définition des besoins :

Notre projet porte sur la mise en place d'une application client serveur à deux niveaux

pour l'automatisation des activités de la gestion des stocks de l'ENIEM de Tizi-Ouzou.

L'application assurera un environnement interactif afin de rendre les tâches accessibles aux différents acteurs et ce via un réseau local dans le but de maîtriser les différentes fonctions concernant la gestion des stocks produit fini.

Identification des acteurs :

Les principaux acteurs qui seront les futurs utilisateurs de notre application sont :

- ✓ Le Chef de département gestion des stocks produits finis (CH-DP-PF) .
- ✓ Le Chef de service programmation livraison PF (CH-S-PROG-PF) .
- ✓ Le Chargé de la gestion des stocks (GE-stock).
- ✓ Le Gestionnaire expéditions (GE-EXP).
- ✓ Le Chef de service Magasins central PF (CH-S-MC).
- ✓ Le Chef de section magasin (CH-SM).

1) Diagramme de contexte :

Le diagramme de contexte est un modèle conceptuel de flux qui permet d'avoir une vision globale des interactions entre le système et les liens. Il permet aussi de bien délimiter le champ d'étude.

Pour notre cas le contexte est donné par la figure suivante :

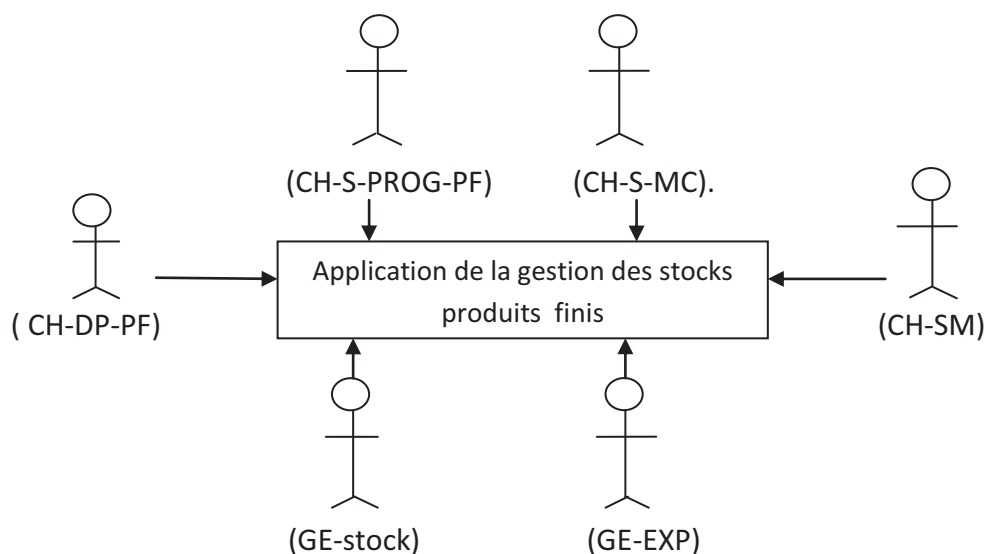


Figure N° 09 : diagramme de contexte.

2) Identification des espaces :

On associe pour chaque acteur un espace regroupant toutes les tâches et actions qu'il peut effectuer.

Pour notre cas nous avons identifié six espaces qui sont :

- ✓ Espace Chef de département gestion des stocks produits finis.

- ✓ Espace Chef de service programmation livraison PF .
- ✓ Espace Chargé de la gestion des stocks.
- ✓ Espace Gestionnaire expéditions.
- ✓ Espace Chef de service Magasins central PF.
- ✓ Espace Chef de section magasin.

Conception :

Le processus de conception de notre système comprend deux niveaux :

- Le niveau applicatif.
- Le niveau de donnée.

le niveau applicatif :

S'appuie essentiellement sur quelques diagrammes de langage de modélisation UML.

A cet effet nous avons adopté la démarche suivante :

Après l'identification des différents acteurs, ainsi que des différentes fonctions du système à concevoir dans la partie d'analyse nous allons mettre en évidence les cas d'utilisations mis en œuvre par les différents acteurs du système, les diagrammes de cas d'utilisation sont alors élaborés.

On formalise graphiquement les scénarios qui décrivent chaque cas d'utilisations à l'aide des diagrammes de séquence. A partir de ses derniers, on identifie les classes, ensuite on élabore les diagrammes des classes généraux.

3) représentation des diagrammes de cas d'utilisation :

Les cas d'utilisations sont des outils formels qui permettent de consigner et d'exprimer les interactions et les dialogues des acteurs avec le système.

Il spécifie ce que le système devrait effectuer sans préjuger de la manière dont cela doit être fait.

Dans ce qui suit nous allons identifier les différents cas d'utilisations.

Les cas d'utilisations :***3.1) Cas d'utilisation relatif au chef de service magasin PF***

Les cas d'utilisation que nous avons recensée sont :

- S'authentifier.
- Modifier le mot de passe.
- Gestion de la livraison ;
 - Ajouter livraison.
 - Supprimer livraison.
 - Modifier livraison.
- Gestion de la réception ;
 - Ajouter réception.
 - Supprimer réception.
 - Modifier réception.

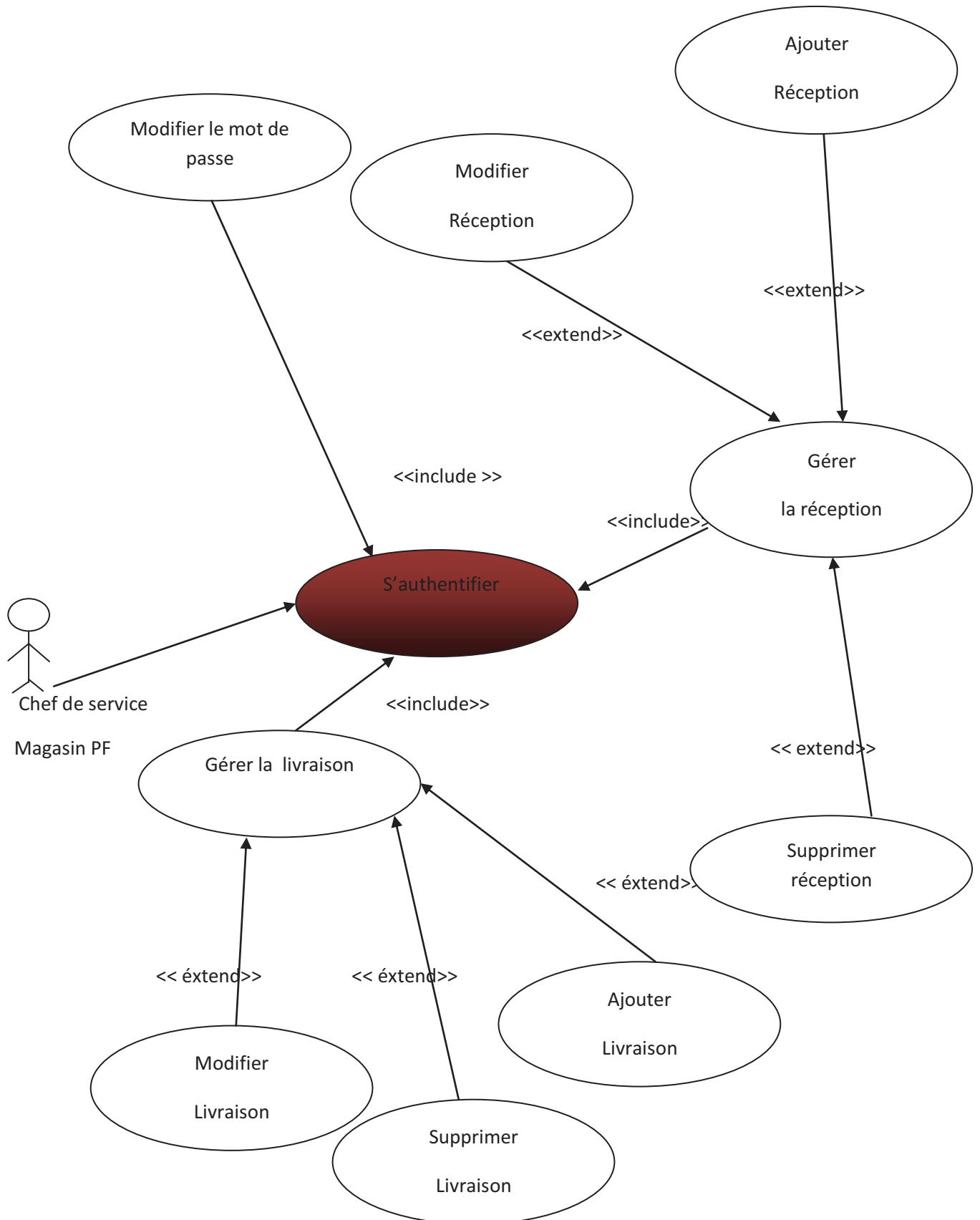


Figure N° 10 : Diagramme de cas d'utilisation relatif au <<chef de service magasin PF>>.

3.2) Cas d'utilisation relatif au gestionnaire des expéditions :

Les cas d'utilisation que nous avons recensée sont :

- S'authentifier.
- Modifier le mot de passe.
- Gestion de la livraison ;
 - Ajouter livraison.
 - Supprimer livraison.
 - Modifier livraison.
- Gestion des bons à servir :
 - Ajouter un bon à servir.
 - Supprimer un bon à servir.
 - Modifier un bon à servir.
- Gestion des bons de sortie :
 - Ajouter des de sortie de PF.
 - Modifier des bons de sorties de PF.
 - Supprimer des bons de sortie de PF.
- vérifier les factures.
- Contrôle des bons de chargements.

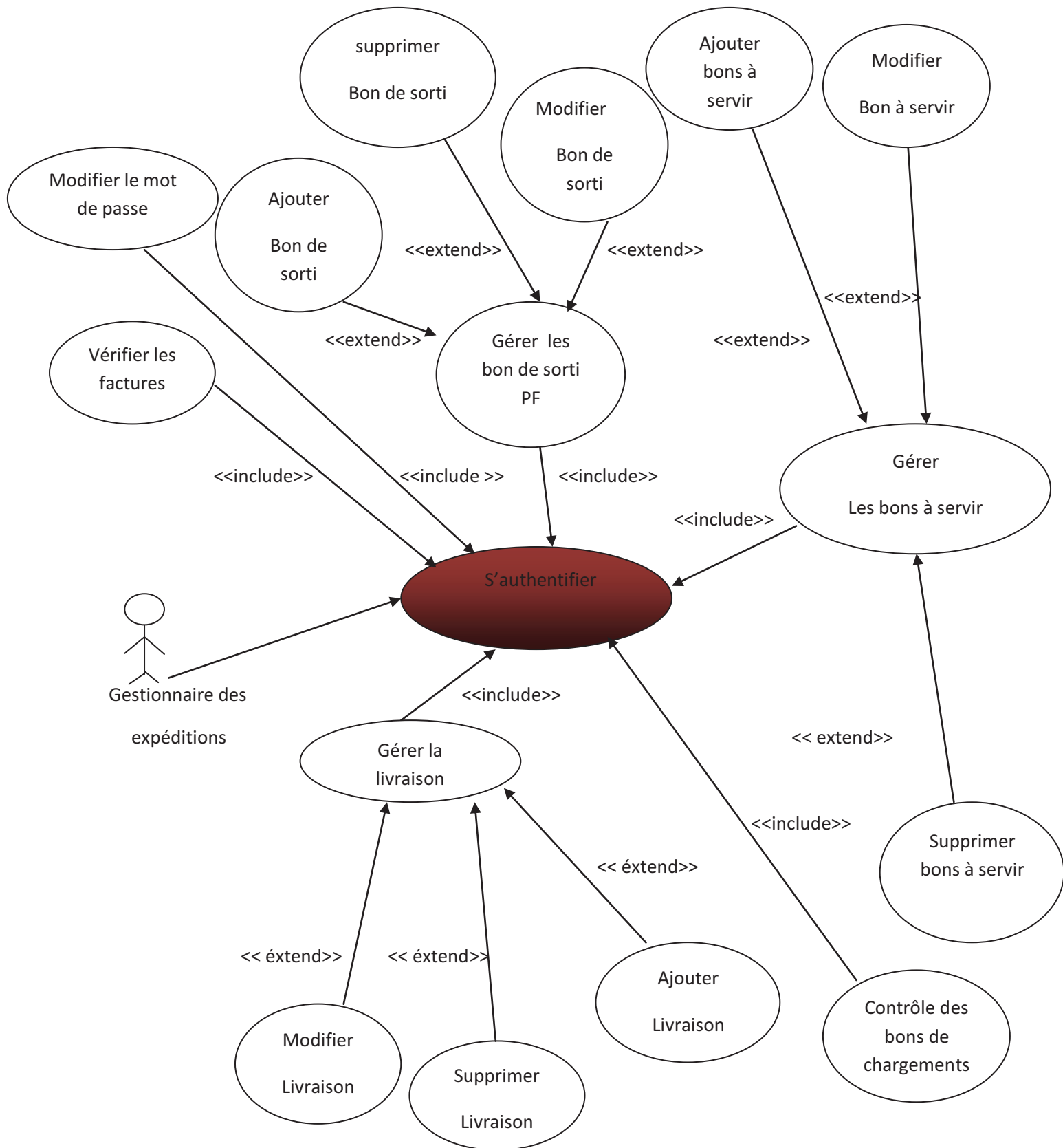


Figure N°11 : Diagramme de cas d'utilisation relatif au <<gestionnaire des expéditions>>.

3.3) Cas d'utilisation relatif au chef de service programmation PF

Les cas d'utilisation que nous avons recensée sont :

- S'authentifier.
- Modifier le mot de passe.
- Gestion des programmes de distribution de produits finis avec comme sous cas d'utilisation :
 - Ajouter programme.
 - Supprimer programme.
 - Modifier programme.

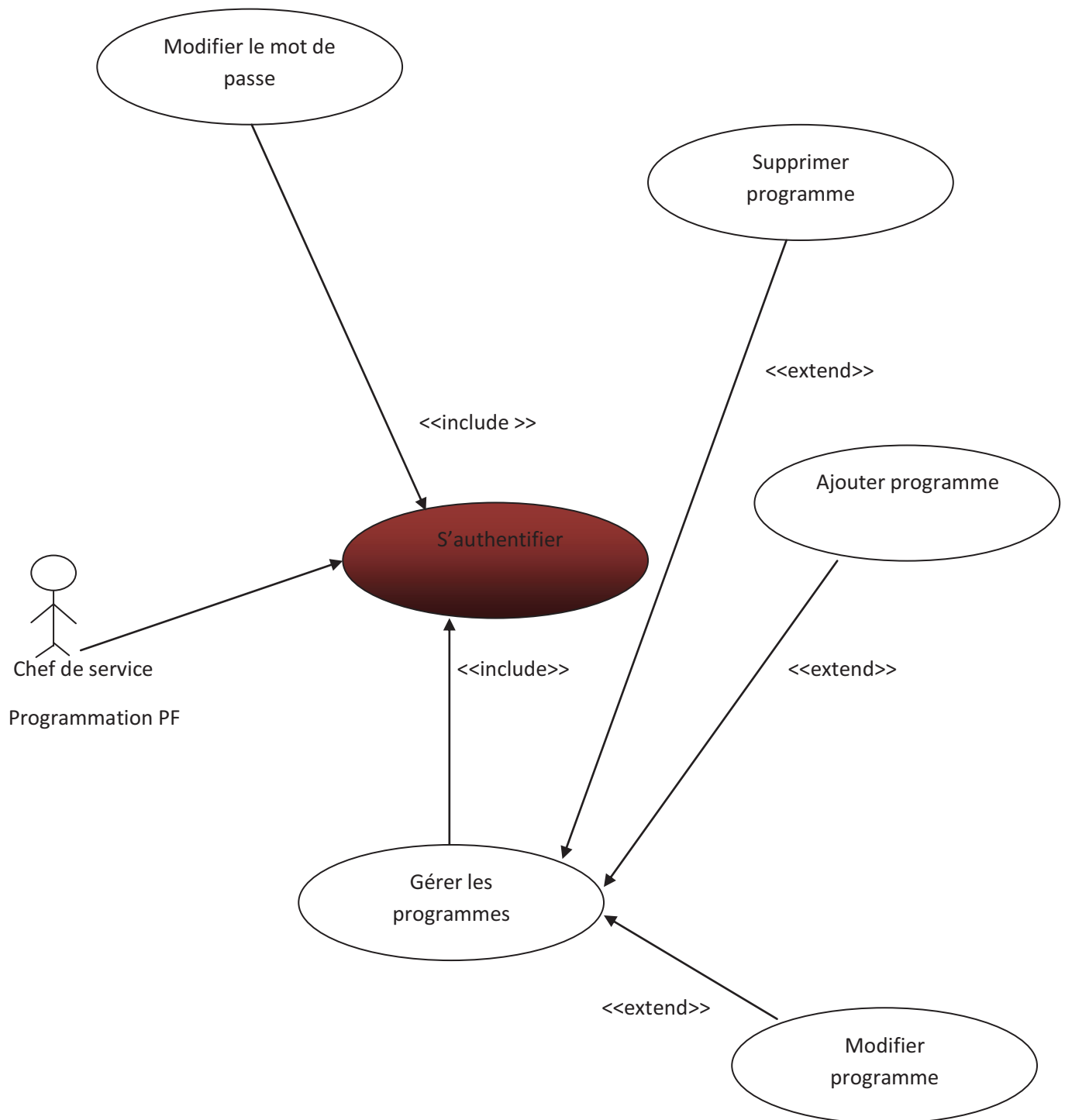


Figure N°12 : Diagramme de cas d'utilisation relatif au <<chef de service programmation PF>>.

4) diagramme de séquence :

A partir des cas d'utilisations détaillés nous pouvons identifier aisément les scénarios, puis élaborer les diagrammes de séquences.

- ✓ Les diagrammes de séquences permettent de représenter les interactions entre objets selon un point de vue temporel. Ils représentent les échanges de messages entre objets au cours du temps. L'axe vertical représente le temps et l'axe horizontal représente les objets qui collaborent, une verticale en pointillé est attachée à chaque objet qui représente sa ligne de vue.
- ✓ Un scénario est une séquence réelle d'événement qui se produit pendant une exécution de système.

Remarque :

Vue le nombre important de cas d'utilisation et de scénario, nous n'allons décrire que deux exemples de cas d'utilisation :

- Ajouter une quantité en stock.
- Consulter facture client.

- Ajouter une quantité en stock.

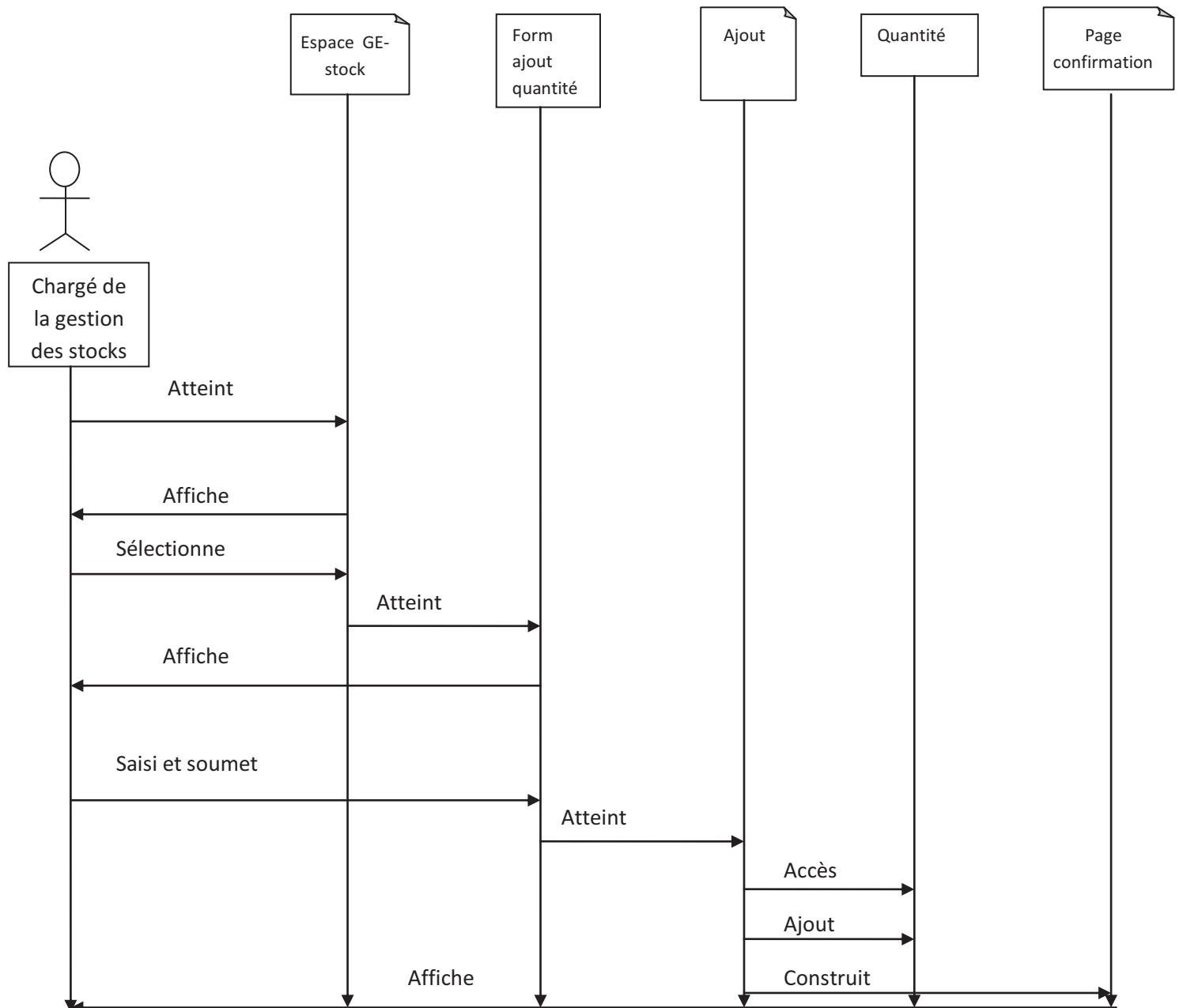


Figure N°13 : Diagramme de séquence << ajouter quantité d'un produit >>

Ajouter quantité :

- 1) le chargé des stocks atteint son espace.
- 2) le système affiche la page correspondante.
- 3) Le chargé des stocks la rubrique (quantité).
- 4) Le système retourne le formulaire ajout.
- 5) le chargé des stocks saisi et soumet les informations.
- 6) Le système affiche une page de confirmation.

➤ Consulter facture.

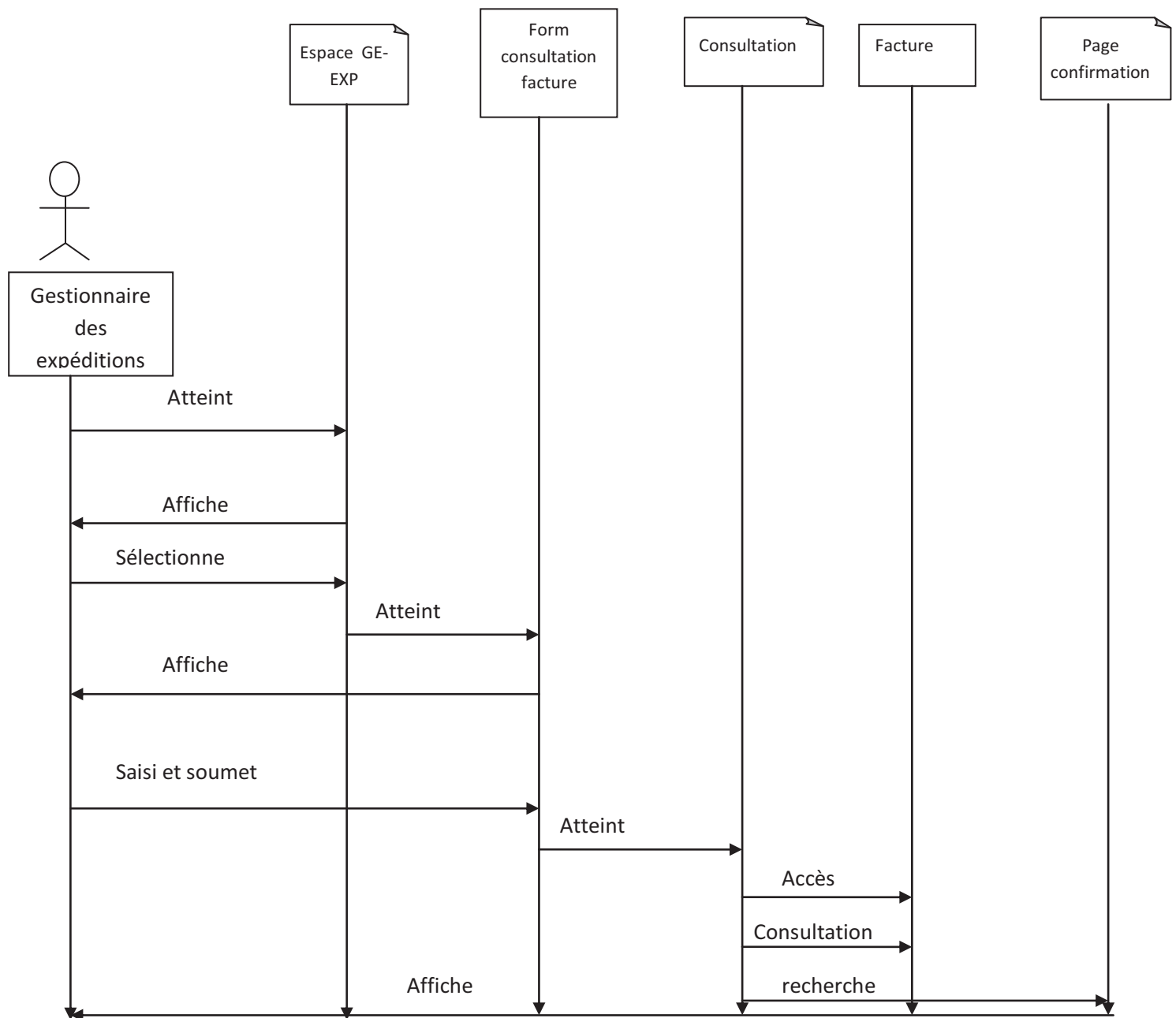


Figure N°14 : Diagramme de séquence << consulter facture client >>

Consulter facture client :

- 1) Le gestionnaire des expéditions atteint son espace.
- 2) Le système affiche la page correspondante.
- 3) Le gestionnaire des expéditions sélectionne la rubrique (facture client).
- 4) Le système retourne le formulaire consultation.
- 5) Le gestionnaire des expéditions saisi et soumet les informations.
- 6) Le système affiche une page de confirmation.

5) Diagramme d'activité :

Le diagramme d'activité fait partie des cinq diagrammes d'UML utilisé pour la modélisation des aspects dynamiques du système. Il est principalement un organigramme qui montre le flot de contrôle d'une activité à l'autre. C'est-à-dire la modélisation des étapes séquentielles dans un processus de calcul.

- Diagramme d'activité du cas d'utilisation <<ajout produit en stock >>.

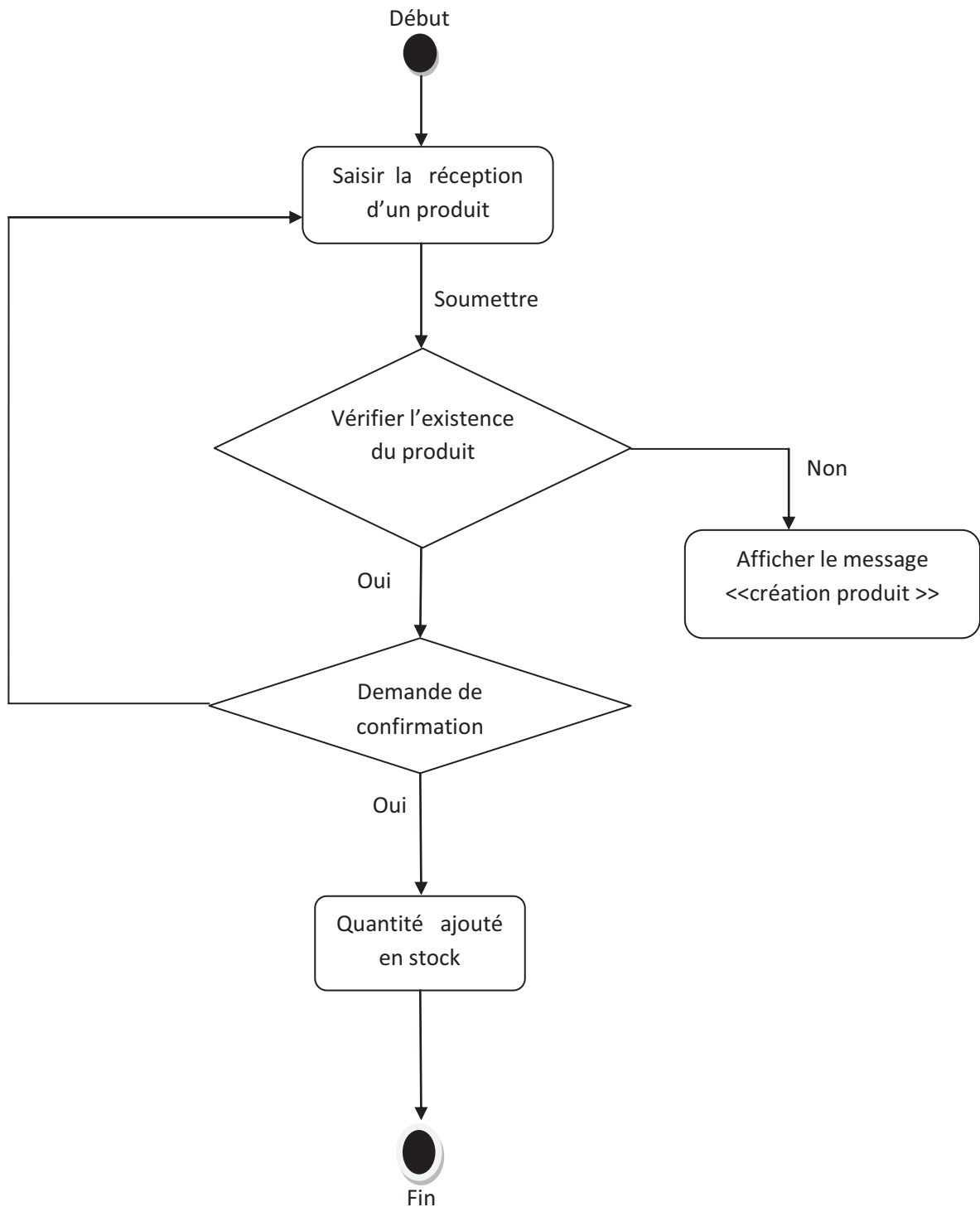


Figure N° 15 : Diagramme d'activité du cas d'utilisation <<ajout produit en stock >>.

- Diagramme d'activité du cas d'utilisation <<consulter facture produit >>.

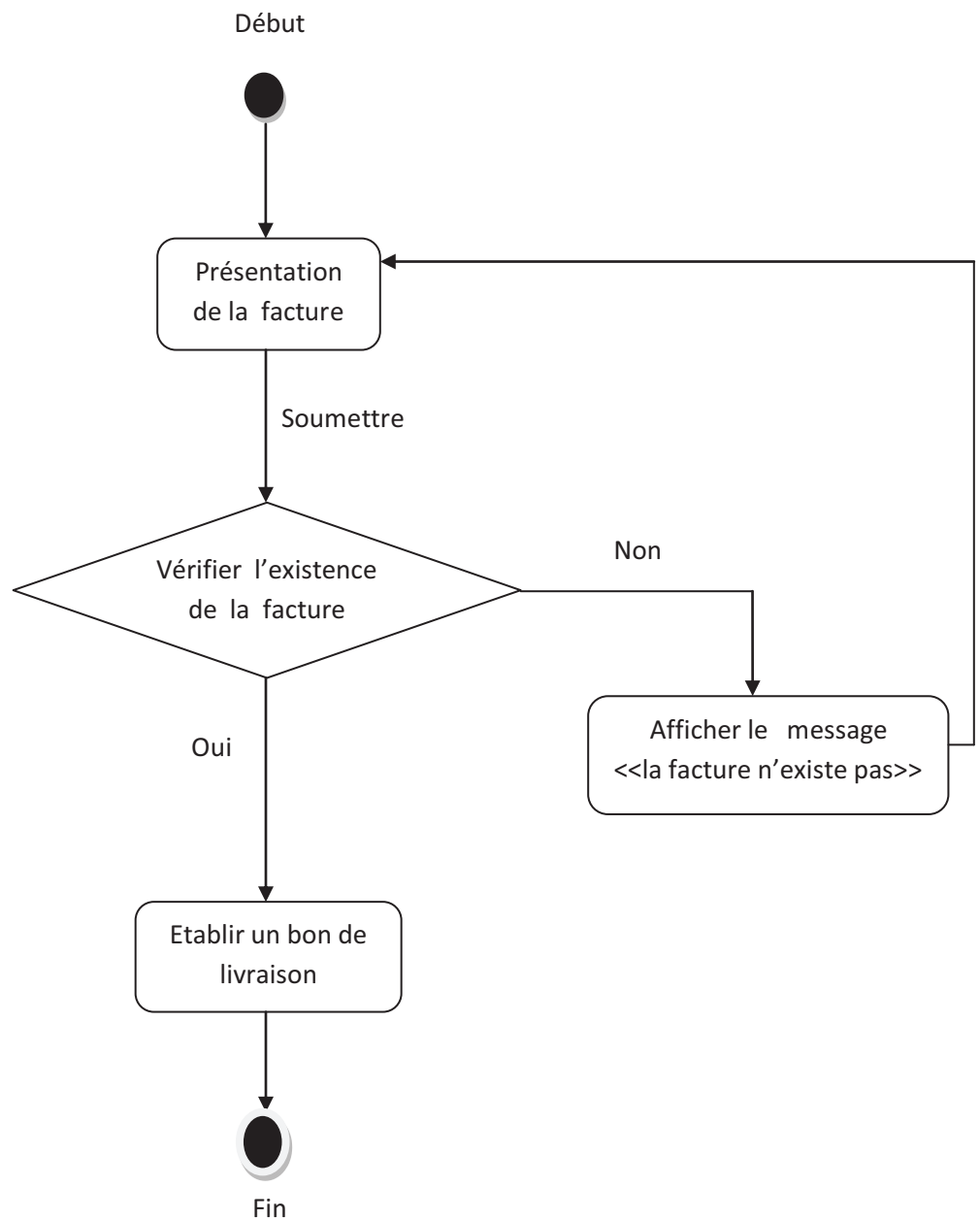


Figure N °16 : Diagramme d'activité du cas d'utilisation <<consulter facture produit>>

- Diagramme d'activité du cas d'utilisation <<fichier article >>.

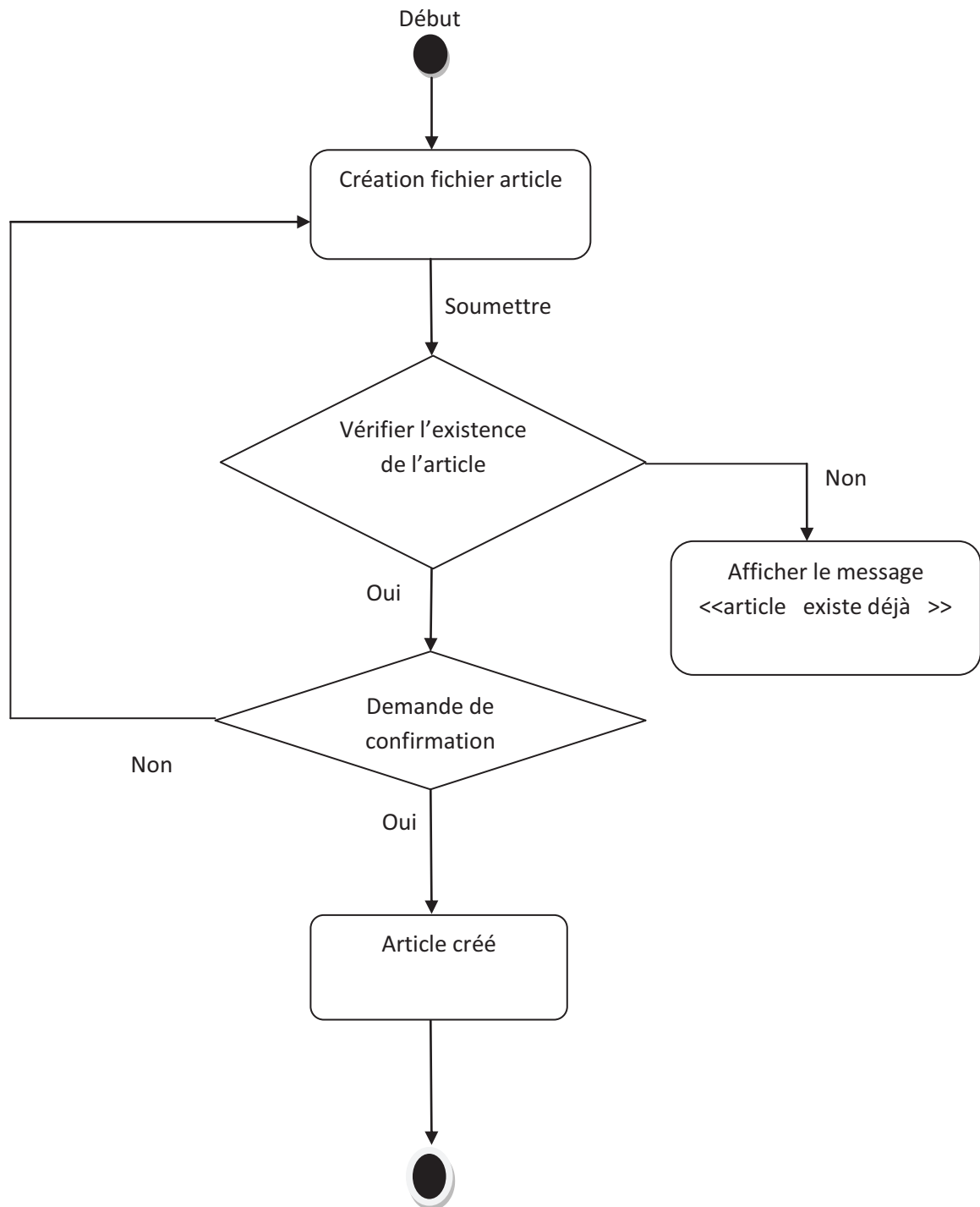


Figure N°17 Diagramme d'activité du cas d'utilisation <<fichier article >>.

- Diagramme d'activité du cas d'utilisation <<authentification de l'utilisateur >>.

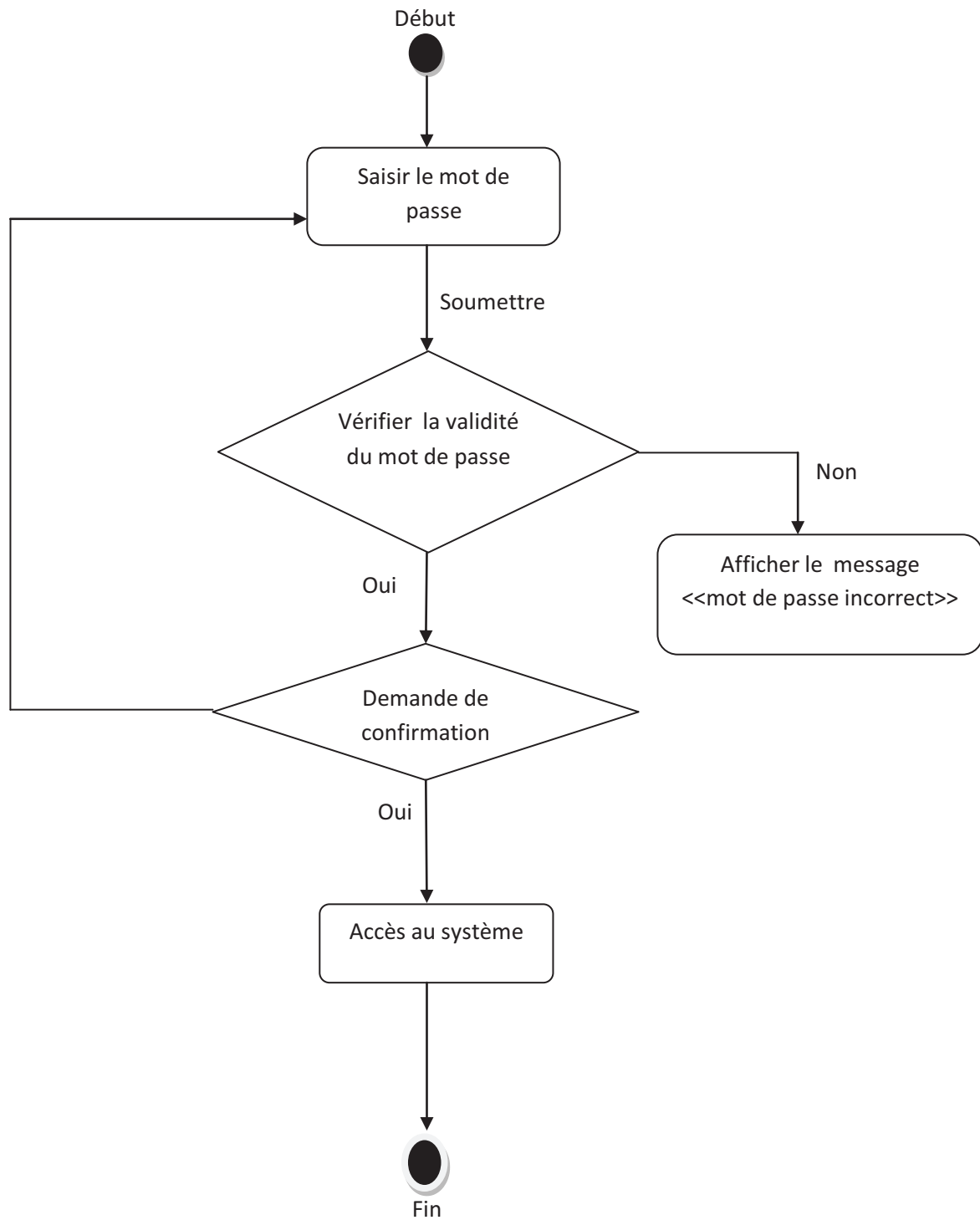


Figure N° 18 Diagramme d'activité du cas d'utilisation <<authentification de l'utilisateur >>

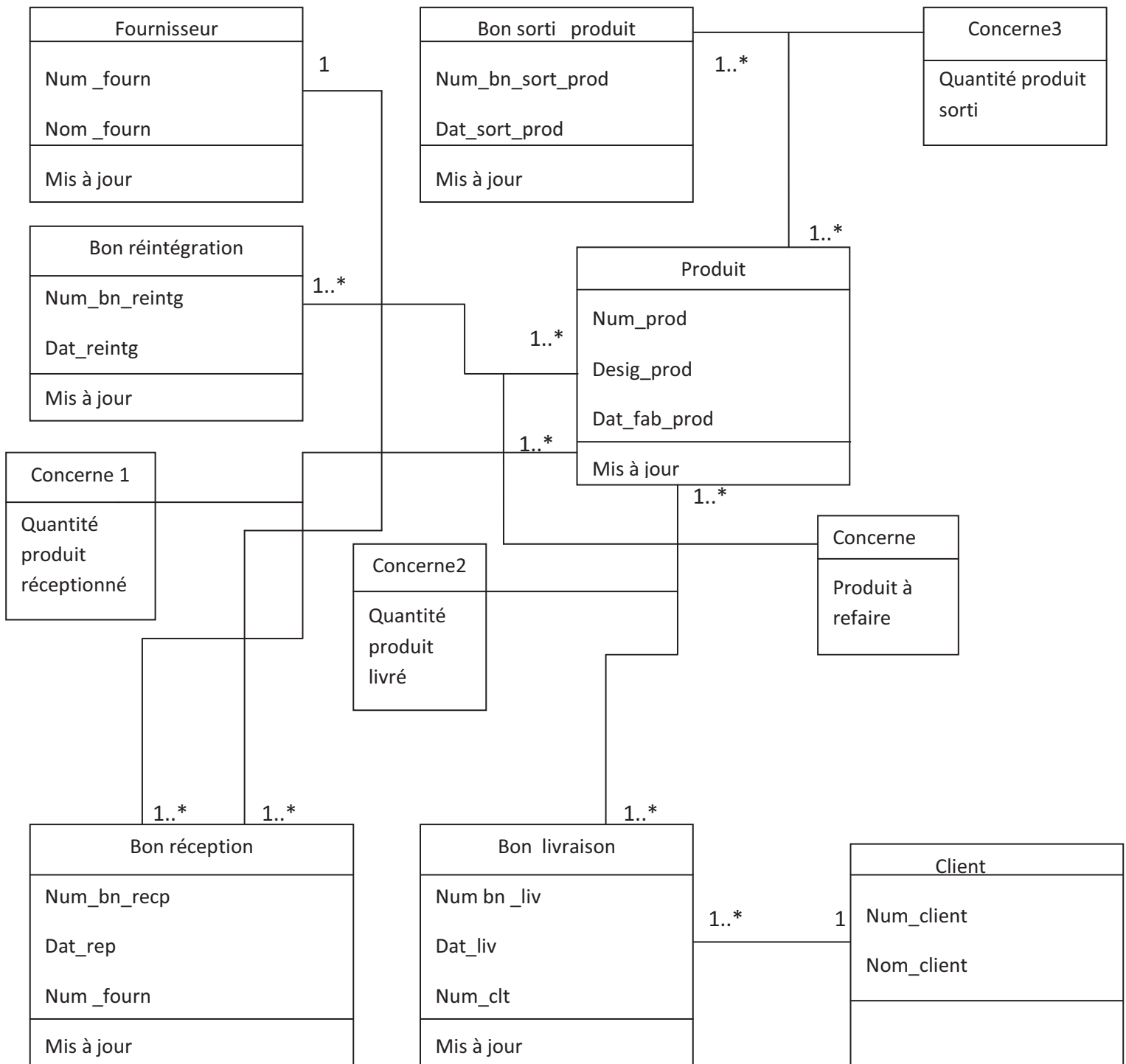
6) *Diagramme de classe :*

Le diagramme de classes représente un ensemble de classe, d'interface et de collaboration ainsi que les relations entre classes. Ils sont les plus fréquents dans la modélisation des systèmes orientés objets.

Dictionnaire de données :

Code de la rubrique	Désignation de la rubrique	Type de la rubrique	taille	Observation
Num_four	Numéro fournisseur	N	06	
Nom_four	Nom fournisseur	A	30	
Num_client	Numéro client	N	06	
Nom_client	Nom client	A	30	
Num_reintg	Numéro bon de réintégration	N	06	
Dat_reintg	Date de réintégration	D	10	JJ/MM/AAAA
Num_bn_recp	Numéro bon de réception	N	06	
Dat_recp	Date de réception	D	10	JJ/MM/AAAA
Num_bn_sort_prod	Numéro bon de Sortie produit	N	06	
Dat_sort_prod	Date sortie produit	D	10	JJ/MM/AAAA
Num_prod	Numéro produit	N	10	
Désig-prod	Désignation produit	A	30	
Dat_fab_prod	Date fabrication produit	D	10	JJ/MM/AAAA

Num_bn_liv	Numéro bon de livraison	N	06	
Dat_liv	Date de livraison	D	10	JJ/MM/AAAA



Figures N°19 : Diagramme de classes.

Produit (Num_prod,Design_prod,Dat_fab_prod,Num_fam_prod).

Bon livraison (Num_bn_liv,Dat_liv,Num_clt).

Bon reception (Num_Bn_rec,dat_rec,Num_fourn).

Bon reintegration(Num_bn_reintg, Dat_reintg).

Bon sorti produit (Num_bn_sort_prod,Dat_sort_prod).

Client (Num_clt,Nom_clt).

Fournisseur (Num_fourn,Nom_fourn).

Concerne1(Num_prod,Num_bn_rec,quantité produit receptionné)

Concern2(Num_prod,Num_bn_liv,quantité produit livré).

Concerne3(Num_prod,Num_bn_sort_prod,quantité produit sorti) .

Concerne(Num_prod,Num_bn _reintg,produit à refaire).

Réalisation

1) Introduction :

Dans ce chapitre nous allons présenter notre plate forme de développement et les outils utilisés pour mener à bien la réalisation de notre application ainsi que , quelques interfaces du logiciel .

2) Outils de développement :

Durant notre réalisation nous avons opté pour l'utilisation de la plate forme Windows avec son système d'exploitation Windows7 , les logiciels de MySql et Java Netbeans 6.8 .

3) Java est un langage de programmation :

Le langage Java : orienté objet issu de la synthèse de plusieurs langages de programmation .En effet il s'inspire du C++ et des techniques approuvées en Small talk et autres langages (organisation en classe , utilisation d'un ramasse miettes (garbag collector), exécution à l'aide d'une machine virtuelle , gestion d'exception...) , tout en supprimant les inconvénients de la pluparts de ses langages .

Le principale atout de java réside dans le fait que se dernier est interprété indépendant de toute plateforme.

En effet la source est compliqué en pseudo code ou bytecode puis exécuté par un interpréteur Java appelé Java Virtual Machine (JVM) . ce concept est a la base du slogan Sun pour Java WORA (WRITE ONCE , RUN Anywhere).En effet , le bytecode , s'il ne contient pas de code spécifique à une plate-forme particulière peut-être exécuté et les obtenir les mêmes résultats sur toute machine disposant d'une JVM , qui existe pour la plupart des systèmes d'exploitation .

De plus il n'ya pas de compilation spécifique pour chaque plate forme . Le code reste indépendant de la machine sur laquelle il s'exécute. Il est donc possible d'exécuté java sur tous les environnements qui possèdent la JVM . Cette indépendance est assurée au niveau du code source grâce à un Unicode et au niveau du Bytecode.

Java Program

```
class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

HelloWorldApp.java

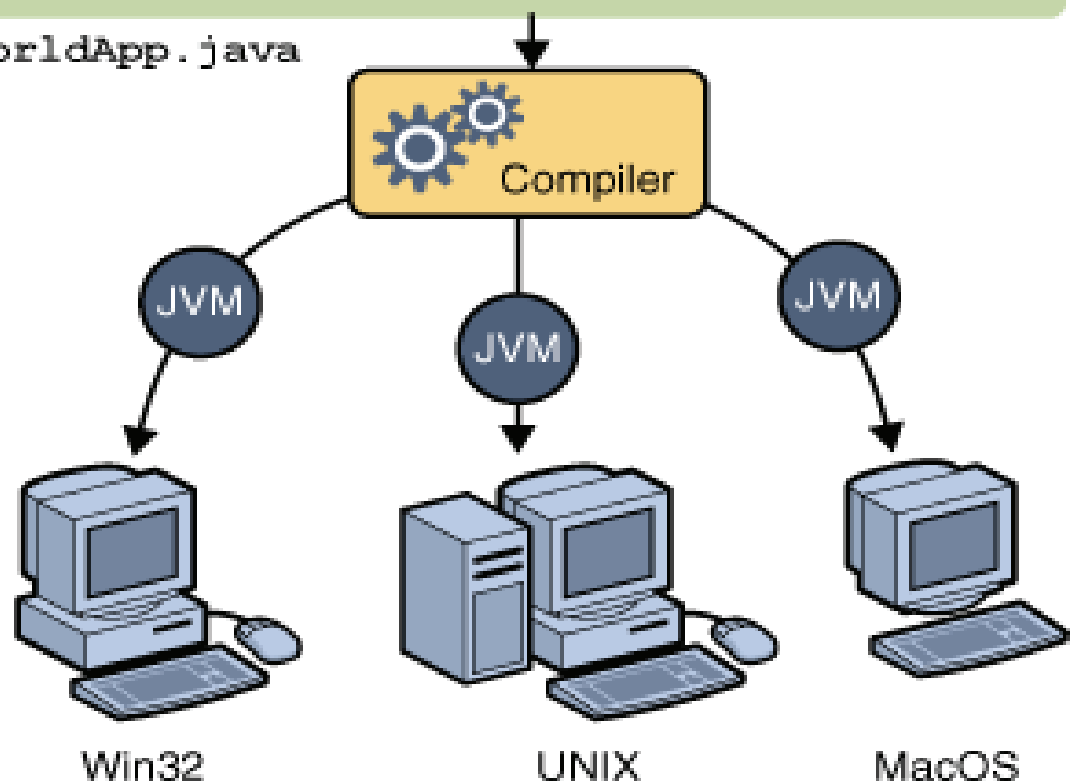


Figure N°20: Indépendance d'un programme en java de toute plate forme.

4) NetBeans 6.8:

NetBeans IDE est un environnement de développement intégré (EDI) de premier ordre pour Windows, Mac, Linux et Solaris. Le projet NetBeans consiste en un EDI Open Source et en une plate-forme d'application permettant aux développeurs de créer rapidement des applications Web, d'entreprise, de bureau et mobiles à l'aide de la plate-forme Java, ainsi que de JavaFX, PHP, JavaScript et Ajax, Ruby et Ruby on Rails, Groovy et Grails et C/C++.

Le projet NetBeans est soutenu par une communauté de développeurs dynamique et propose différentes ressources de documentation et de formation, ainsi qu'un choix varié de plug-ins tiers.

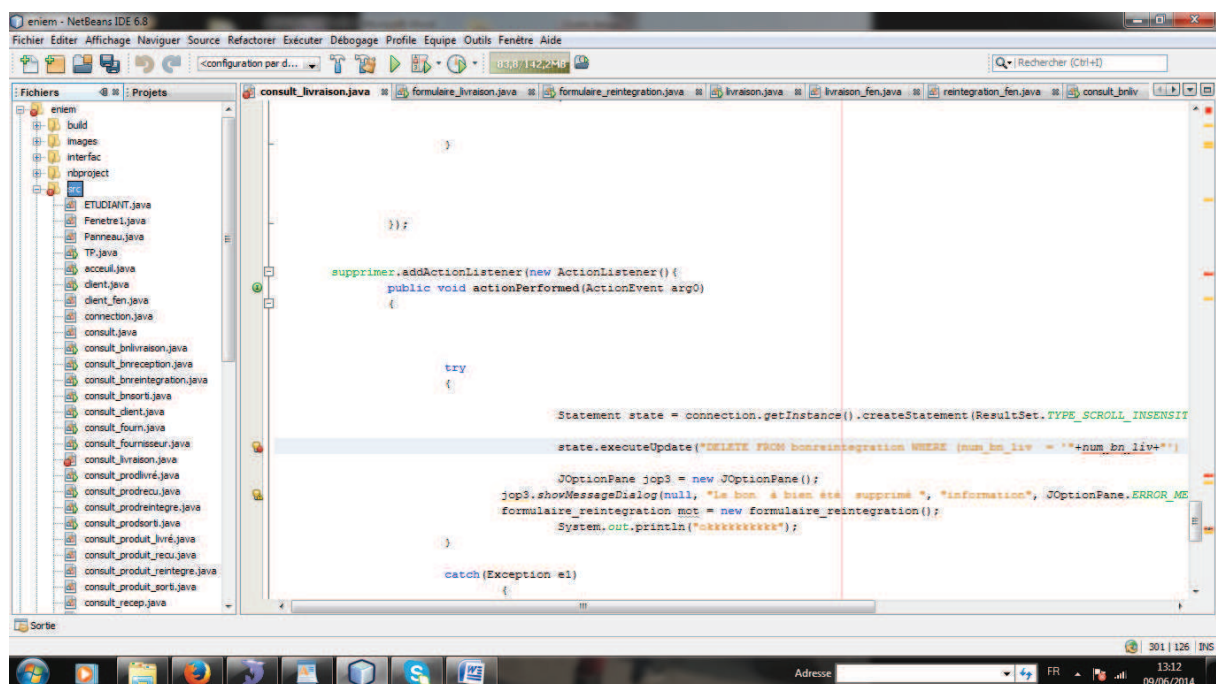


Figure N°21: capture d'écran présentant l'interface de développement NetBeans.

5) *MySql serveur 5.5 :*

MySQL est un serveur de bases de données relationnelles SQL développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est multi-thread et multi-utilisateur.

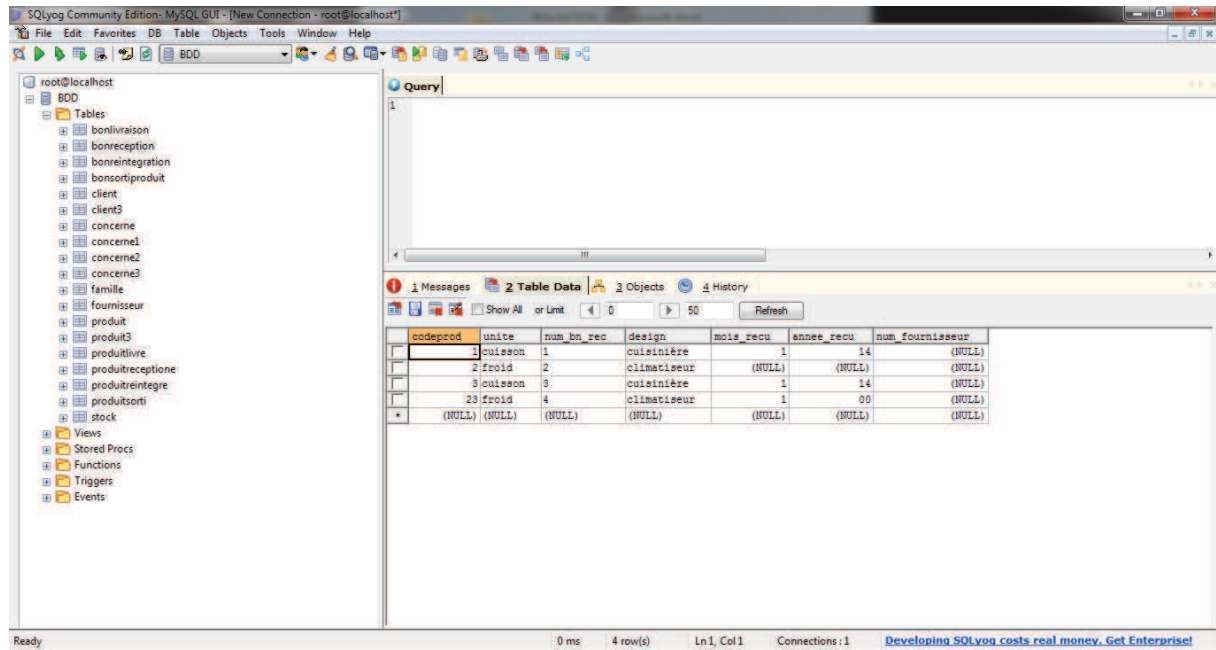
C'est un logiciel libre développé sous double licence selon qu'il est distribué avec un produit libre ou avec un produit propriétaire. Dans ce dernier cas, la licence est payante, sinon c'est la licence publique générale (GPL) qui s'applique. Un logiciel qui intègre du code MySQL ou intégré MySQL lors de son installation devra donc être libre ou acquérir une licence payante. Cependant, si la base de données est séparée du logiciel propriétaire qui ne fait qu'utiliser des API tierces (par exemple en C# ou php), alors il n'y a pas besoin d'acquérir une licence payante MySQL. Ce type de licence double est utilisé par d'autres produits comme le framework de développement de logiciels Qt (pour les versions antérieures à la 4.5).

5.1) Accès aux bases de données :

JDBC est l'acronyme de java Data Base Connectivity et désigne l'API définie par SUN Microsystems pour permettre un accès aux bases de données avec Java. Pour pouvoir utiliser JDBC, il faut un pilote spécifique au SGBD qui contient la base avec laquelle on veut se connecter. Pour ce qui est de MySQL on utilise <<JDBC3 PostgreSQL pilote >> disponible Java.sql et sont inclus dans la JDK depuis la version 1.1 de cette API.

Ce package contient 4 classes principales, chacune correspondant à une étape de l'accès à la base de données.

- ✓ Drivers Managers : permet de changer et de configurer le driver de la base de données.
- ✓ Connexion : établit la connexion et l'authentification de la base de données.
- ✓ Statement : (PreparedStatement) permet de la transmission de la requête à la base de données.
- ✓ ResultSet : les objets de cette classe sont destinés à contenir les informations retournées par la base dans le cas d'un SELECT.

5.2) Présentation de la base de données :

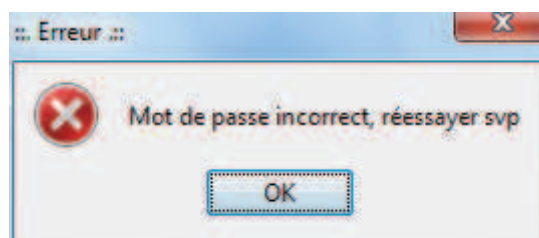
The screenshot shows the SQLyog Community Edition interface. On the left, a tree view displays the database structure for 'root@localhost' and 'BDD'. The 'Tables' folder is expanded, showing a list of tables including 'bonlivraison', 'bonreception', 'bonreintegration', 'bonsortiproduit', 'client', 'client3', 'concerne', 'concerne1', 'concerne2', 'concerne3', 'famille', 'fournisseur', 'produit', 'produit3', 'produitlivre', 'produitreception', 'produitreintegration', 'produitsorti', and 'stock'. The 'Table Data' tab is selected, showing a table with the following columns: 'codeprod', 'unite', 'num_bn_rec', 'design', 'mois_recu', 'annee_recu', and 'num_fournisseur'. The table contains 4 rows of data.

codeprod	unite	num_bn_rec	design	mois_recu	annee_recu	num_fournisseur
1	puisson	1	cuisinière	1	14	(NULL)
2	froid	2	climatiseur	(NULL)	(NULL)	(NULL)
3	puisson	3	cuisinière	1	14	(NULL)
23	froid	4	climatiseur	1	00	(NULL)
*	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Figure N°22: Présentation de la base de données.

5.3) Présentation de quelques interfaces de notre application :

- La fenêtre de la figure ci-dessous montre l'interface du mot de passe qui permet d'accéder au système de l'utilisateur, en cas d'erreur dans la saisie du mot de passe ; une fenêtre d'erreur s'affiche.



- La fenêtre de la figure ci-dessous montre l'interface de l'accueil qui

Comporte plusieurs menus et sous menus qui nous permettent le traitement de la base de données de notre application.



- Cet exemple montre le résultat d'un clic sur le sous menu **bon de réception** dont on pourra faire plusieurs mis à jours.



The image shows a screenshot of a software window titled "bon de reception". The window contains a form with the title "BON DE RECEPTION" in large blue letters. Below the title, there are several input fields and dropdown menus:

- code du bon:** A text input field.
- Designation produit:** A dropdown menu with "cuisinière" selected.
- Unité produit:** A dropdown menu with "cuisson" selected.
- N° Produit :** A text input field.
- Mois reception:** A dropdown menu with "Janvier" selected.
- Annee reception:** A dropdown menu with "2011" selected.

At the bottom of the form, there are three buttons: "Insertion", "RETURN", and "MODIFICATION".



The screenshot shows a window titled 'FOURNISSEUR'. The main heading is 'FOURNISSEUR' in a large, blue, serif font. Below the heading, there are four input fields arranged horizontally, each preceded by a label: 'NOM Fournisseur :', 'NUMERO Fourniss...', 'CHEF d'unité :', and 'NUMERO Produit ...'. At the bottom of the window, there are three buttons: 'Insertion', 'RETURN', and 'MODIFICATION'.



The screenshot shows a window titled 'produit livré'. The main heading is 'PRODUIT Livré' in a large, blue, serif font. Below the heading, there is a single input field preceded by the label 'Num produit livré :'. At the bottom of the window, there are two buttons: 'CONSULTER' and 'RETURN'.

Conclusion générale

Toute entreprise d'aujourd'hui, soucieuse de son efficacité commerciale et de sa performance économique, se doit la maîtrise de l'automatisation de son système d'information. En plus de la coopération de ses différentes structures, l'acquisition et l'exploitation de l'information en temps réel sont plus que jamais des éléments primordiaux pour sa compétitivité commerciale.

Le stage effectué dans l'enceinte de **l'ENIEM**, nous a permis de découvrir le domaine de l'activité des approvisionnements, de voir surtout de près les tâches complexes auxquelles font face les acteurs des différents services et d'acquérir ainsi de larges connaissances de gestion. Ce stage s'est soldé par ce présent travail, dont la réalisation nous a mis face à la réalité de mettre en application en plus de nos connaissances acquises durant notre cursus et de les enrichir par de nouvelles. L'expérience dans le développements d'application, ce travail, à travers les étapes d'analyse, de conception et de réalisation, nous a permis de nous aguerrir à la démarche *UML*, et de travailler avec le produit *MySQL*, qui ne cesse de prouver sa puissance.

Notre application qui a traité la gestion des stocks de **l'ENIEM** des produits finis et qui adopte l'architecture Client/serveur, offre plusieurs services et simplifie la tâche de gestion à tous les niveaux.

Cependant, quoiqu'elle présente diverses fonctionnalités, quelques améliorations et extensions peuvent être apportées à notre application. En guise de perspective, les possibilités d'évolution de ce projet sont nombreuses allant jusqu'à la gestion complète de tous les services en incluant les finances et la comptabilité... tout en mettant l'accent sur l'aspect coopératif entre les différentes structures.

Bibliographie

*[DIGA.01] : DIGLLO Frédéric, Intégration des systèmes client/serveur
CNAM AIX-EN-Provence 2001/2002.*

*[GAR.03] : George et Olivier GARDARIN , le client/serveur , Eyrolles
2003.*

*[HAR.99] : Robert ORFALI , dan HARKEY et Jerry EDWARDS ,
client/serveur*

Guide de service, Edition originale : Jhon WILLYZ SONS , INC 1999.

*[BOR.01] : G.BOOCH ,RUMBAUCH , I.JACOBSON , le guide de
l'utilisateur UML Edition Eyrolles 2001.*

*[GAB.09] : Joseph GABAY et David GABAY , UML 2 Analyse et
conception Edition DUNOD 2009.*

*[TAZ.06] : Conception et réalisation d'une application client/serveur
3-tiers Sous oracle.*

*Cas :Gestion des ressources humaines de L'ENIEM , mémoire
d'ingénieur , Université Mouloud Mammeri TIZI_OUZOU , réalisé
par TAZBOUDJT Said et TOUMERT Mohammed ,2005/2006 .*

*[B EL.09] : Conception et réalisation d'une application Client/serveur
sous Oracle.*

Cas : Gestion de l'approvisionnement de L'électro-industries.

*AZAZGA Tizi-Ouzou réalisé par BELKEBIR NAZIM , ZOUKOULOU
KHELIFA , 2008/2009.*

Les sites :

[w.w.w] : www.développeur.com

www.postgresql.org

www.commentcamarche.com

Table des matières

1. Introduction générale

Chapitre1 : Etude préalable

Description générale de la méthode « MERISE ».....	2
1. Présentation de l'organisme d'accueil <<ENIEM>>	5
1.1. Historique	5
1.2. Mode d'organisation.....	5
1.2.1. Organisation général	6
1.3. Les directions.....	7
1.3.1. Direction générale.....	7
1.3.2. Direction industrielle	7
1.3.3. Direction des ressources humaines	7
1.3.4. Direction développement et partenariat	8
1.3.5. Direction des finances et partenariat.....	8
1.3.6. Direction planification et contrôle de gestion	8
1.3.7. Direction de marketing et de la communication.....	8
1.4. Les unités	8
1.5. Présentation de l'unité commerciale	9
1.6. Organisation de l'unité commerciale.....	10
1.7. Organisation générale	11
1.7.1. Département vente.....	11
1.7.2. Département gestion des stocks	12
1.7.3. Département marketing	13
1.7.4. Département service après vente.....	13
1.7.5. Département finance et comptabilité	13
1.7.6. Département administration générale et ressource humaine (AGR)	13
1.7.7. Département gestion de contrôle.....	14
1.7.8. Description du département AGRH	14
2. Etude de l'existant.....	17
2.1. Etude des postes de travail.....	17
2.2. Etudes ses documents.....	21
2.3. Etude des registres.....	35

2.4. Etude des procédures.....	45
3. La codification.....	54
4. Le diagramme des flux.....	56
4.1. Description des flux.....	57
5. Critique de l'existant et suggestion.....	58
Conclusion.....	59

Chapitre2 : Client/serveur

Introduction

1. Le modèle client/serveur.....	61
1.1. Les techniques de dialogue client/serveur.....	61
1.2. Notion de base du modèle client/serveur.....	61
2. Middleware.....	62
3. L'architecture client/serveur multi niveaux	63
3.1. Architecture client/serveur a deux niveaux	64
3.2. Architecture client/serveur a trois niveaux	65
3.3. Architecture client/serveur a multi niveaux	67
Conclusion.....	68

Chapitre3 : Atelier de génie logiciel

Introduction

1. Qu'est ce qu'un atelier de génie logiciel.....	69
2. Fonctionnalités d'un AGL.....	69
3. Dictionnaire de données	70
4. Avantage liés à l'utilisation d'un AGL.....	70
5. Quelques AGL.....	71
5.1. WinDev.....	71
5.2. BOUML.....	75
5.3. Rational Rose.....	77
5.4. Produits Uniface.....	77
NetBeans.....	82

Conclusion.

Chapitre4 : Analyse et Conception

Introduction

1. Diagramme de contexte	96
2. Identification des espaces	96

3. Représentation des diagrammes de cas d'utilisation	97
3.1 Cas d'utilisation relatif au chef de service magasin PF	98
3.2 Cas d'utilisation relatif au gestionnaire des expéditions	100
3.3 Cas d'utilisation relatif au chef de service programmation PF	102
4 Diagrammes de séquence	104
5 Diagramme d'activité.....	107

Introduction

Chapitre 5: La réalisation

Introduction

2 Outils de développement	112
3 Java est un langage de programmation	112
4 Netbeans 6.8	114
5 MySQL.....	115
5.1 Accès aux bases de données	115
5.2 Présentation de la base de données	116
5.3 Présentation de quelques interfaces de notre application	117

Conclusion générale

Bibliographie.