

Université Mouloud Mammeri Tizi-Ouzou

Faculté de Génie Electrique et d'Informatique
Département d'Informatique



MEMOIRE

De fin d'études

Master 2 Option : conduite de projets informatiques

Conception et réalisation d'une application client-serveur pour la gestion des stocks

Cas : compagnie maritime d'affrètement_compagnie maritime générale

(CMA_CGM) de Bâb Ezzouar

Proposé et dirigé par :

Mr. HAMMACHE Arezki.

Réalisé par :

Melle. MORSLI Nacera.

Melle. ZAOUACHE Kamelia.

2013/2014

Remerciements

Nos remerciements vont tout premièrement au bon dieu tout puissant de nous avoir donné le courage, la volonté et la patience pour réaliser ce travail.

Nous tenons à témoigner notre profonde gratitude et remerciements les plus sincères à Mr. HAMMACHE pour avoir dirigé notre travail pour son soutien et pour le temps qu'il a consacré au bon déroulement de ce travail.

Nos remerciements distingués à l'ensemble du personnel du groupe CMA CGM.

Nous remercions également le président et les membres de Jury pour nous avoir fait l'immense privilège d'évaluer équitablement notre travail.

On souhaite remercier toute personne ayant contribué de près ou de loin à l'élaboration de ce mémoire.

A tous merci.

❧ Dédicaces ❧

Je dédie ce travail à :

*Mes très chers parents qui ont toujours été là pour moi et à qui je ne pourrais jamais assez
exprimer l'amour que je leur porte,*

A mes chères sœurs, Rebiha et Ouzna

A mes chers frères Ahmed, Ali et Sofiane,

A mes chers amis (Amirouche, Zohra, Tassadit, Lydia, Nina, Katia, Nel et Sabrina),

A tous mes camarades

A ma chère copine Kamelia et toute sa famille,

A un ami exceptionnel qui nous a quittés trop tôt IMMOUN Malik paix à son âme

Nacera

❧ Dédicaces ❧

Je dédie ce travail à :

*Mes très chers parents qui ont toujours été là pour moi et à qui je ne pourrais jamais assez
exprimer l'amour que je leur porte,*

A ma chère sœur Nadia,

A mes chers frères Samir et Karim,

A mes chers amis (Amirouche, Zohra, Tassadit, Nina, Katia, Lydia , Maya, Nel et Sabrina),

A tous mes camarades

A ma chère copine Nacera et toute sa famille,

A mon cher ami qui nous a quittés trop tôt IMMOUN Malik paix à son âme

Kamelia

Table des matières

Introduction générale.....	6
Chapitre I : Concepts de base sur le développement d'une application 3-tiers	8
I.1 Introduction.....	9
I.2 Généralités sur les réseaux et le web.....	9
I.2.1 Réseaux [AD 11].....	9
I.2.1.1 classification des réseaux	9
I.2.1.2 Couches protocolaires du modèle OSI.....	13
I.1.1.3. Le réseau Internet, intranet, extranet.....	17
I.1.1.4 Classement par relation fonctionnelle.....	18
I.2.2 Le web.....	19
I.2.2.1 Présentation du web	19
I.2.2.2 Fonctionnement du web	20
I.2.2.3 Les composants du web	20
I.2.3 Le modèle Client / Serveur	24
I.2.3.1 Topologie des architectures client-serveur de données.....	25
I.2.4 Outils de développement d'une application 3-tiers.....	27
I.2.4.1 La plateforme .NET	27
I.2.4.2 La plateforme J2Entreprise Edition.....	29
I.2.4.3 PHP	31
I.2.4.4 Comparaison des trois outils	32
I.3 Conclusion	35
Chapitre II : Description de l'organisme d'accueil et étude de l'existant	36
II.1 Introduction.....	37
II.2 Présentation de l'organisme d'accueil	37
II.2.1 Historique du groupe CMA CGM	37
II.2.2 Organigramme général	38
II.2.3 Missions de CMA-CGM.....	40
II.2.4 Situation informatique	40
II.2.5 Problématiques	40
II.2.6 Objectifs.....	41
II.3 Étude de l'existant	41
II.3.1 L'organigramme du champ d'étude.....	41

II.3.2 Diagramme des flux.....	42
II.3.2.1 Définition.....	42
II.3.2.2 Les concepts de base.....	42
II.3.2.3 Description des acteurs internes	42
II.3.2.4 Présentation des acteurs externes.....	43
II.3.2.5 Description des flux	43
II.3.3 Etude des postes de travail.....	44
II.3.3.1.Définition d'un poste de travail	44
II.3.3.2 Liste des postes de travail	44
II.3.3.3 Documents manipulé par les postes.....	46
II.3.4 Etude des documents	47
II.3.4.1 Liste des documents.....	47
II.3.5 Procédures.....	53
II.3.5.1 Achat.....	53
II.3.5.2Gestion des affectations	54
II.3.5.3Réapprovisionnement des articles	54
II.3.5.4Inventaire	54
II.3.6 Critiques et suggestions	54
II.4 Conclusion	55
Chapitre III : Analyse et conception.....	56
III.1 Introduction	57
III.2 Présentation d'UML [HB 06].....	57
III.2.1 Historique	57
III.2.2 Diagrammes d'UML	58
III.2.3 Modélisation avec l'UML	60
III.3 Analyse.....	61
III.3.1 Spécification des besoins.....	61
III.3.2 Identification des acteurs.....	61
III.3.3 Diagramme de contexte.....	61
III.3.4 Représentation des cas d'utilisation	62
III.3.4.1 Définition d'un cas d'utilisation.....	62
III.3.4.2 Diagramme de cas d'utilisation	62
III.4 Conception.....	67
III.4.1 Diagramme de séquence.....	67

III.4.2 Les règles de gestion du diagramme de classes.....	71
III.4.3 Diagramme de classes	71
III.4.4 Liste des classes.....	Erreur ! Signet non défini.
III.4.5 Liste des classes d'association.....	Erreur ! Signet non défini.
III.4.6 Modèle relationnel.....	72
III.5 Conclusion.....	72
Chapitre IV : Réalisation.....	74
IV.1 Introduction.....	75
IV.2 Les outils de développement.....	75
IV.2.1 NetBeans	75
IV.2.2 Serveur Web Apache tomcat.....	75
IV.2.3 Dreamweaver	76
IV.2.4 Langages web utilisés.....	76
IV.2.5 JSP.....	76
IV.2.6 Feuille de style en cascade	77
IV.2.7 JavaScript.....	77
IV.2.8 Le Java Development Kit (JDK).....	77
IV.2.9 Système de gestion de bases de données.....	77
IV.2.10 Langages d'interrogation de bases de données utilisés.....	78
IV.3 Architecture Modèle-Vue-Contrôleur	80
IV.4 Quelques interfaces de notre application	81
IV.4.1.Interface NetBeans	82
IV.4.2 Interface d'accueil.....	82
IV.4.3 Interface d'authentification	83
IV.4.4 Interface de gestion des comptes.....	84
IV.4.5 Interface de page de confirmation.....	84
IV.4.6 Interface de gestion des affectations	85
IV.4.7 Fenêtre de consultation des articles affectés	85
IV.4.8 Interface de gestion d'achat.....	85
IV.4.9 Fenêtre de consultation des articles.....	86
IV.4.10 Interface d'édition de la liste des articles	86
IV.5 Conclusion.....	87
Conclusion générale	88
Références	Erreur ! Signet non défini.

Liste des figures

Figure I.1 Classification des réseaux en fonction de la distance	10
Figure I.2 Topologie en bus.	11
Figure I.3 Topologie en anneau.	12
Figure I.4 Topologie en étoile.	12
Figure I.5 Modèle OSI.....	14
Figure I.6 La navigation web.....	20
Figure I.7 Architecture client-serveur.	24
Figure I.8 Architecture 2-tiers.....	26
Figure I.9 Architecture 3-tiers.....	27
Figure I.10 Architecture Dot NET.	28
Figure I. 11 Architecture J2EE.	30
Figure I.12 Schéma fonctionnel du PHP.	32
Figure II.1 Organigramme générale du groupe CMA-CGM.....	39
Figure II.2 Organigramme du champ d'étude.....	42
Figure II.3 Diagramme des flux.	44
Figure III.1 Représentation graphique de la démarche de modélisation.	60
Figure III.2 Diagramme de contexte.....	62
Figure III.3 Diagramme de cas d'utilisation générale.....	64
Figure III.4 Diagramme de cas d'utilisation détaillé relatif au « FM ».	65
Figure III.5 Diagramme de cas d'utilisation détaillé relatif à « L'ACHETEUR ».	66
Figure III.6 Diagramme de cas d'utilisation détaillé relatif au « chef de département des moyens généraux ».	67
Figure III.7 Diagramme de séquence «Authentification».....	68
Figure III.8 Diagramme de séquence «Ajouter un article».	69
Figure III.9 Diagramme de séquence «Modifier mot de passe ».	70
Figure III.9 Diagramme de classe global.	71
Figure IV.2 Interface NetBeans.	82
Figure IV.3 Interface d'accueil.	83
Figure IV.4. Interface d'authentification.	83
Figure IV.5 Interface de gestion des comptes.....	84
Figure IV.6 Interface de page de confirmation de l'ajout.....	84
Figure IV.7 Interface de gestion des affectations.	85
Figure IV.8. Fenêtre de consultation des articles affectés.....	85
Figure IV.9 Interface de gestion des achats.	86
Figure IV.10 Fenêtre de consultation des articles.	86
Figure IV.11 Fenêtre d'édition de la liste des articles.	87

Liste des tableaux

Tableau I.1 Comparaison des outils de programmation	35
Tableau II.1 Situation informatique.	40
Tableau II.2 Description des flux.....	43
Tableau II.3 Etude du poste « Facility manager ».	45
Tableau II.4 Etude du poste « Acheteur ».	46
Tableau II.5 Etude du poste « Chef du département des moyens généraux ».....	46
Tableau II.6 Documents manipulés par les différents postes.	47
Tableau II.7 Liste des documents.	47
Tableau II.8 Etude du document « Bon de commande ».	48
Tableau II.9 Etude du document « bon de livraison ».	49
Tableau II.10 Etude du document « Demande de paiement ».....	50
Tableau II.11 Etude du document « Facture ».....	51
Tableau II.12 Etude du document « Cheque ».....	52
Tableau II.13 Etude du document « Facture pro forma ».	53
a. Diagramme de cas d'utilisation.....	59
b. Diagramme de séquence	59
d. Diagramme de classes	59
Tableau III.1 les cas d'utilisations.	63
Tableau IV.1 Table article.	78
Tableau IV.2 Table fournisseur.	78
Tableau IV.3 Table service.	79
Tableau IV.4 Table affecter.	79
Tableau IV.5 Table livrer.	79
Tableau IV.6 Table contenir.....	79
Tableau IV.7 Table utilisateur.....	79

Introduction générale

L'informatique désigne l'automatisation du traitement de l'information par un système concret « machine » ou abstrait. On entend également par « l'informatique » l'ensemble des sciences et techniques en rapport avec le traitement de l'information.

L'informatique est de plus en plus utilisée dans tous les domaines d'activités y compris celui de la gestion des stocks auquel nous rattacherons d'ailleurs notre étude, et cela pour une meilleure gestion des différents traitements exigés par cette activité de gestion des stocks.

En effet, Nous avons constaté pendant nos recherches que l'ensemble des traitements au sein du département facilities du groupe CMA-CGM se fait manuellement ce qui engendre un certain nombre de problèmes tel que la non existence d'une base de données globale pour permettre l'enregistrement des articles ce qui engendre la difficulté de se renseigner sur l'état du stock ; donc la meilleure solution pour pallier aux problèmes est l'informatisation afin d'assurer un accès instantané aux données , ce qui simplifie le travail administratif.

Pour la réalisation de cette tâche, notre choix s'est porté sur Le langage de modélisation UML (Unifier ModelingLanguage), qui est une partie intégrante de la démarche UP. Ses diagrammes sont largement utilisés dans chaque étape et phases de ce processus de développement.

Pour l'implémentation, le choix du langage de programmation a été dicté par le type de l'application qui devrait être réalisée d'une part, et être accessible via le réseau Intranet de l'entreprise d'autre part. Ainsi, le choix s'est porté sur la plate-forme JEE. La base de données est implémentée avec MySQL qui est largement compatible avec JEE.

Ayant présenté les outils et la méthode adoptée, nous allons maintenant exposer le plan du mémoire qui se subdivisera en quatre principaux chapitres.

Dans le premier chapitre intitulé « Concepts de base sur le développement d'une application 3-tiers » nous présentons quelques concepts de base qui nous ont servi pour l'accomplissement de notre projet.

Puis, au sein de « Présentation de l'organisme d'accueil et l'étude de l'existant », deuxième chapitre de ce travail, nous commençons à présenter l'organisme d'accueil, approfondir la compréhension du contexte du système par un processus continu de collecte d'information auprès des experts du domaine, ensuite déterminer les inconvénients majeurs de la gestion actuelle du service d'accueil, énumérer des suggestions informatiques qui peuvent remédier aux difficultés rencontrées, et enfin, tenant compte des moyens du service facilities, nous proposons la solution qui paraît la plus adéquate.

Au niveau du troisième chapitre intitulé « Analyse et conception », nous analysons les principaux objectifs attendus du futur système à concevoir, et qui seront décrits par le diagramme des cas d'utilisation puis nous établirons les diagrammes de séquence suivie du diagramme de classe.

Finalement dans le dernier chapitre qu'on a nommé « Réalisation » nous présentons les outils de développement qui nous ont servi pour la réalisation de l'application, puis nous terminerons par présenter quelques interfaces de l'application.

Chapitre I : Concepts de base sur le développement d'une application 3-tiers

I.1 Introduction

Comme le but de notre travail consiste en la réalisation d'une application de gestion des stocks de type client/serveur à trois niveau nous avons jugé utile de présenter dans ce chapitre les généralités sur les réseaux, une vue globale sur le web ainsi les différents outils de développement d'application 3-tiers.

I.2 Généralités sur les réseaux et le web

I.2.1 Réseaux [1]

Un réseau informatique est un ensemble d'équipements interconnectés selon une architecture leur permettant ainsi de communiquer afin d'assurer des échanges d'informations sous forme de données numériques.

Les premiers réseaux informatiques permettaient de relier des ordinateurs centraux, volumineux et fragiles, à des terminaux passifs (des postes de travail avec clavier et écran mais sans puissance de calcul).

Actuellement, ils permettent l'interconnexion de tous types d'ordinateurs, comme: les serveurs, les stations de travail, les ordinateurs personnels, etc.

I.2.1.1 classification des réseaux

Les réseaux peuvent être classifiés en fonction de différents critères:

- L'étendue
- La topologie
- Le support de connexion.

a. Classification selon l'étendu

On distingue différents types de réseaux, selon leur taille (nombre de machines), leur vitesse de transfert des données ainsi que leur localisation. On peut distinguer quatre types de réseaux

- PAN (*Personal Area Network*)
- LAN (*Local Area Network*)
- Man (*Métropolitain Area Network*)
- WAN (*Wide Area Network*)

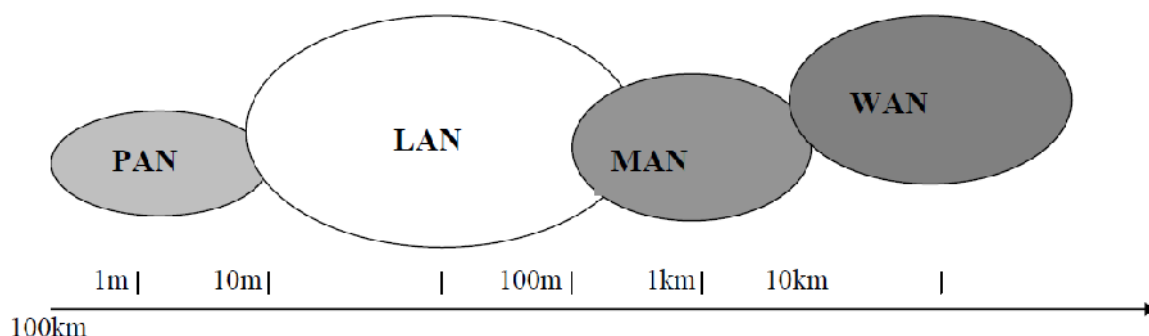


Figure I.1 Classification des réseaux en fonction de la distance

a.1 Réseaux PAN (Personal Area Network)

C'est la plus petite taille de réseau. Ils interconnectent sur quelques mètres les équipements personnels tels que GSM, portable, organisateur d'un même utilisateur.

a.2 Réseaux local ou (LAN : Local Area Networks)

Il s'agit d'un ensemble d'ordinateurs appartenant à une même organisation et reliés entre eux dans une petite aire géographique par un réseau, souvent à l'aide d'une même technologie (la plus répandue étant Ethernet).

Un réseau local est donc un réseau sous sa forme la plus simple. La vitesse de transfert de données d'un réseau local peut s'échelonner entre 10 Mbps (pour un réseau ethernet par exemple) et 1 Gbps (en FDDI ou Gigabit Ethernet par exemple). La taille d'un réseau local peut atteindre jusqu'à 100 voir 1000 utilisateurs.

a.3 Réseaux métropolitains ou MAN (Metropolitan Area Networks)

Les MAN interconnectent plusieurs LAN géographiquement proches (au maximum quelques dizaines de km) à des débits importants. Ainsi un MAN permet à deux noeuds distants de communiquer comme si ils faisaient partie d'un même réseau local.

Un MAN est formé de commutateurs ou de routeurs interconnectés par des liens hauts débits (en général en fibre optique).

a.4 Réseaux étendus WAN (Wide Area Networks)

Un WAN interconnecte plusieurs LANs à travers de grandes distances géographiques.

Les débits disponibles sur un WAN résultent d'un arbitrage avec le coût des liaisons (qui augmente avec la distance) et peuvent être faibles.

Les WAN fonctionnent grâce à des routeurs qui permettent de "choisir" le trajet le plus approprié pour atteindre un nœud du réseau.

b. Classification selon la topologie

Un réseau informatique est constitué d'ordinateurs reliés entre eux grâce au matériel (câblage, cartes réseau, ainsi que d'autres équipements permettant d'assurer la bonne circulation des données). L'arrangement physique de ces éléments est appelé topologie physique

b.1 Topologie en bus

Dans une topologie en bus tous les ordinateurs sont reliés à une même ligne de transmission par l'intermédiaire de câble, généralement coaxial. Le mot "bus" désigne la ligne physique qui relie les machines du réseau.

Cette topologie a pour avantages d'être facile à mettre en œuvre et de fonctionner facilement, par contre elle est extrêmement vulnérable étant donné que si l'une des connexions est défectueuse, c'est l'ensemble du réseau qui est affecté.

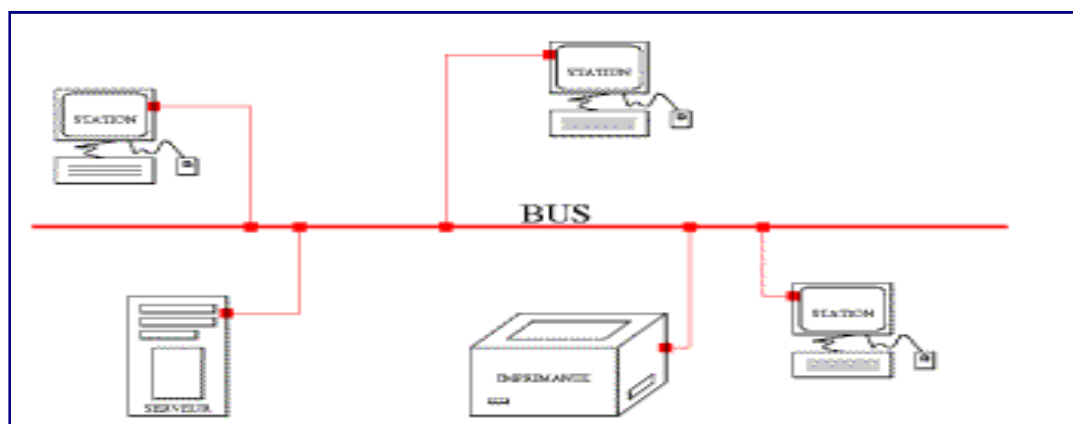


Figure I.2 Topologie en bus.

b.2 Topologie en anneau

Dans un réseau en topologie en anneau, les ordinateurs communiquent chacun à leur tour, nous avons donc une boucle d'ordinateurs sur laquelle chacun d'entre eux accède à la communication dès que le média de transmission se libère.

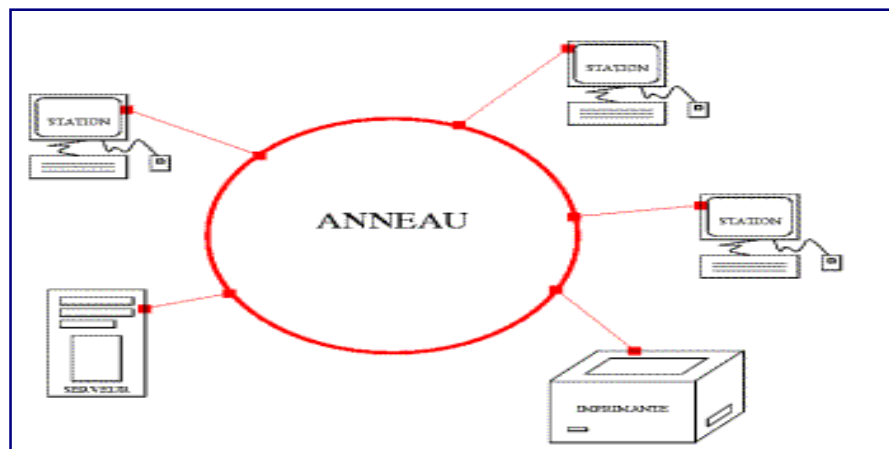


Figure I.3 Topologie en anneau.

b.3 Topologie en étoile

Dans une topologie en étoile, les ordinateurs du réseau sont reliés à un système matériel appelé Hub ou concentrateur.

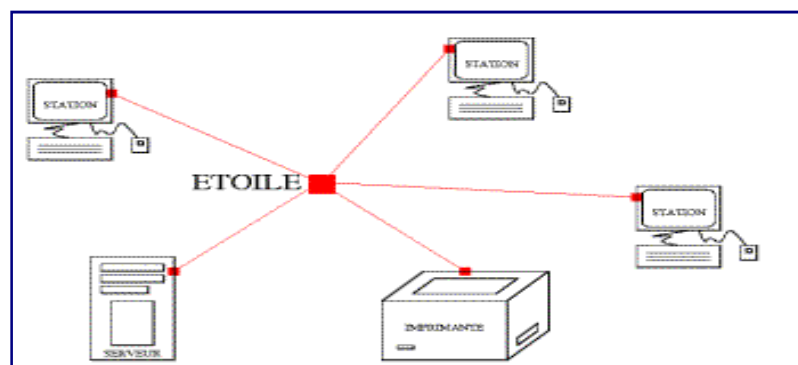


Figure I.4 Topologie en étoile.

Contrairement aux réseaux construits sur une topologie en bus, ces réseaux sont moins vulnérables aux pannes car nous pouvons aisément retirer une des connexions en la débranchant du concentrateur sans pour autant paralyser le reste du réseau.

c. Classification selon le support de transmission

On cite deux types de réseaux :

- Réseaux filaires
- Réseaux sans fil

c.1 Réseaux filaires

On distingue deux grandes familles de supports qui sont les câbles métalliques et les câbles optiques.

c.2 Réseaux sans fil

Est un réseau informatique ou numérisé qui connecte différents postes ou systèmes entre eux par ondes radio. Il peut être associé à un réseau de télécommunications pour réaliser des interconnexions entre nœuds. Nous pouvons citer quelques types : Bluetooth, wifi, wimax, GSM (*Global System for Mobile Communication*), GPRS (*General Packet Radio Service*), UMTS (*Universal Mobile Telecommunication System*).

I.2.1.2 Couches protocolaires du modèle OSI

La première évolution des réseaux informatiques a été des plus anarchiques, chaque constructeur développant sa propre technologie. Le résultat fut une quasi-impossibilité de connecter différents réseaux entre eux. Pour pallier à ce problème d'interconnexions, l'ISO (*International Standards Organisation*) décida de mettre en place un modèle de référence théorique décrivant le fonctionnement des communications réseaux.

Ainsi fût créé le modèle OSI, à partir des structures réseau prédominantes de l'époque : DECNet (*Digital Equipment Corporation's Networking développé par digital*) et SNA (*System Network Architecture développé par IBM*). Ce modèle a permis aux différents constructeurs de concevoir des réseaux interconnectables.

Le modèle OSI est un modèle conceptuel. Il a pour but d'analyser la communication en découpant les différentes étapes en sept couches, chacune de ces couches remplissant une tâche bien spécifique.

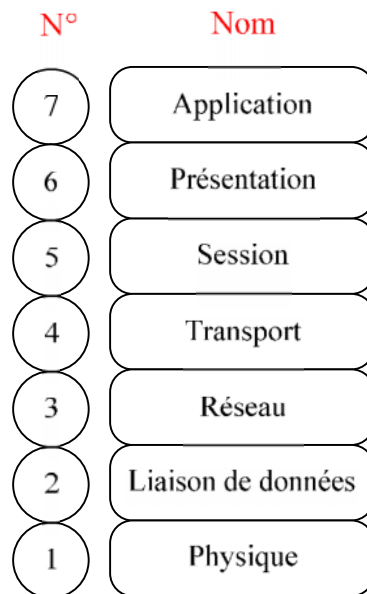


Figure I.5 Modèle OSI.

a. Couche physique

Dans le domaine des réseaux informatiques, la couche physique est la première couche du modèle OSI. Elle est chargée de la transmission effective des signaux électriques ou optiques entre les interlocuteurs. Son service est généralement limité à l'émission et la réception d'un bit ou d'un train de bits continu (notamment pour les supports synchrones comme la fibre optique). Cette couche est chargée de la conversion entre bits et signaux électriques ou optiques. On appelle cette conversion la modulation. Elle est en pratique toujours réalisée par un circuit électronique spécifique.

b. Couche liaison de données

Dans le domaine des réseaux informatiques, la **couche de liaison de données** est la seconde couche des sept couches du modèle OSI. La couche de liaison de données est la couche de protocole qui transfère les données entre les nœuds adjacents d'un réseau étendu (WAN) ou entre des nœuds sur le même segment d'un réseau local (LAN). La couche de liaison de données fournit les moyens fonctionnels et procéduraux pour le transfert de données entre des entités d'un réseau et, dans certains cas, les moyens de détecter et potentiellement corriger les erreurs qui peuvent survenir au niveau de la couche physique.

Parmi les protocoles s'exécutant au niveau de cette couche, nous citons : Ethernet pour les réseaux locaux (multi-nœuds), le protocole point à point (PPP), HDLC (*High-Level Data Link Control*) et ADCCP (*Advanced Data Communication Control Procedures*) pour des

connexions points à points (double nœud) sont des exemples de protocoles de liaison de données.

c. Couche réseau

Cette couche gère l'adressage de niveau trois, la sélection du chemin et l'acheminement des paquets au travers du réseau. Ses apports fonctionnels principaux sont donc:

- **Le Routage :** Déterminer un chemin permettant de relier les 2 machines distantes suivant des critères bien précis;
- **L'Acheminement :** Retransmission d'un PDU (*Protocol Data Unit* ou *Unité de données de protocole*) dont la destination n'est pas locale pour le rapprocher de sa destination finale.
- **Le Contrôle des flux :** Représente un asservissement du débit binaire de la source vers le puits. Quand une machine qui a un débit montant supérieur au débit descendant de la destination, la source diminue son débit pour ne pas submerger le puits de requêtes (obligeant parfois, vu que le puits ne peut pas les traiter, la réémission de ces dernières).

Cette couche est donc la seule à être directement concernée par la topologie du réseau. C'est aussi la dernière couche supportée par toutes les machines du réseau pour le transport des données utilisateur : les couches supérieures sont réalisées uniquement dans les machines d'extrémité. Le PDU (*Protocol Data Unit*) de cette couche est souvent appelé « paquet ». La fonction de « relayage » (terme OSI) est parfois appelée « acheminement ».

d. Couche transport

La couche transport gère les communications de bout en bout entre processus. Cette couche est souvent la plus haute couche dont on se préoccupe de la correction des erreurs (exception connue dans le monde IP : utilisation de DNS sur UDP). C'est-à-dire que le service de niveau transport consiste généralement en un service en mode connecté offrant le transfert de messages ou d'octets bruts garantis sans corruption, pertes, réordonnancement, duplication. En particulier, c'est le service offert par les protocoles TCP (dans le monde IP).

En fonction des protocoles, le PDU est appelé « message », « paquet » ou, uniquement dans le cas TCP, « fragment » (ou « segment » en Internet).

On peut aussi parler du multiplexage au niveau de la couche transport.

e. Couche session

Les deux services originaux de la couche session sont la synchronisation des communications (n'importe quel intervenant peut émettre à tout moment) et la gestion des « transactions ». Un service cependant a été rajouté, c'est un mécanisme de correction des erreurs de traitement par restauration d'un état antérieur connu.

Les services de transport sont des services de communication point à point, c'est-à-dire avec deux interlocuteurs. Mais le modèle OSI doit aussi convenir aux communications multipoints. Deux genres de communications multipoints sont explicitement mentionnées dans la norme : les communications en étoile où une session est un ensemble de communications point à point avec un interlocuteur engagé dans tous les échanges ; et la diffusion où tous les interlocuteurs reçoivent tous les messages. Des mécanismes de synchronisation sont alors requis pour savoir par exemple qui répond à quoi. C'est le rôle des protocoles de cette couche.

Quelques protocoles de la couche session : AppleTalk, NetBIOS, RPC.

f. Couche présentation

La couche présentation est chargée du codage des données applicatives. Les couches 1 à 5 transportent des octets bruts sans se préoccuper de leur signification. Mais ce qui doit être transporté en pratique, c'est du texte, des nombres et parfois des structures de données arbitrairement complexes. Un protocole de routage par exemple doit transporter un graphe représentant au moins partiellement la topologie du réseau. Le rôle de la couche présentation est donc de convertir les données applicatives manipulées par les programmes en chaînes d'octets effectivement transportées par le réseau.

j. Couche application

La couche application est surtout, du point de vue du modèle, le point d'accès aux services réseaux. Comme le modèle n'a pas pour rôle de spécifier les applications, il ne spécifie pas de service à ce niveau.

La couche d'application représente des données pour l'utilisateur ainsi que du codage et un contrôle du dialogue : des mécanismes de communication offerts aux applications de l'utilisateur.

I.1.1.3. Le réseau Internet, intranet, extranet

a. Internet

Le mot Internet vient du terme anglais " Internetwork ", " inter" pour "interconnexion " et "net" pour "network". C'est un réseau international d'ordinateurs, il englobe plusieurs réseaux de taille moindre, qui communiquent entre eux grâce à un protocole d'échange de données, standard (TCP/IP). Les différents ordinateurs branchés au réseau Internet communiquent entre eux et la démarche est transparente pour l'utilisateur.

a.1 Les services Internet

La disponibilité d'une infrastructure de télécommunication comme Internet a permis le déploiement de devers services :

a.1.1 le WEB

Le Web ou le World Wide Web (WWW) est un système hypertexte qui rend au public une lecture plus dynamique, fonctionnant sur l'Internet. Le principe du web repose sur l'utilisation des hyperliens qui lient les pages web entre elles, pour naviguer entre des documents multimédias (pages de textes enrichies de sons, graphiques, images fixes et animées, vidéos,...) grâce à un logiciel appelé navigateur. Ce dernier permet d'effectuer une demande par l'intermédiaire d'une adresse (URL) et de lire le résultat de cette recherche. [II.2]

a.2 Les protocoles d'Internet

a.2.1 Le protocole TCP /IP[6]

Ce protocole est en fait une combinaison de deux protocoles agissant chacun à un niveau différent.

Le TCP : Fonctionne en mode déconnecté, c'est-à-dire qu'une liaison est établie entre deux ordinateurs. C'est un protocole fiable dans la mesure où il est capable de déterminer si un paquet est arrivé ou s'il est interrompu, s'occupe de la transmission des paquets, vérifie l'ordre d'arrivée et l'intégrité des données.

L'IP : est le nœud de transmission des données au-dessous de TCP. Il s'occupe de transport des données, mais il ne contient aucun outil de contrôle de données ni de vérification des paquets, il permet d'adresser les paquets au bon endroit.

a.2.2 Le HTTP (HyperText Transport Protocol)

C'est un protocole léger et rapide, utilisé pour délivrer des fichiers multimédia et hypertextes appelés plus généralement ressources, en utilisant Internet. Il gère la totalité des échanges réalisés entre le client et le serveur. Pour chaque demande d'accès à l'URL, une requête http est émise indépendamment par le client.

a.2.3 Le DNS (Domaine Name Service)

C'est un ensemble de protocoles permettant aux utilisateurs d'un réseau TCP/IP d'accéder aux hôtes à l'aide des noms conviviaux hiérarchisés, le serveur DNS assure la transformation des noms des domaines en adresses IP et inversement, car la structure des adresses IP est complexe à manipuler pour cela on utilise le DNS, sans celui-ci les utilisateurs devraient mémoriser les adresses IP.

b. Intranet

Lorsque les services Internet sont exploités dans un réseau local (à l'intérieur d'une entreprise, d'une administration ou une institution publique), on parle alors d'Intranet.

Intranet est le synonyme de «réseau privé» s'appuie sur de nombreux services de communication et de travail coopératif : (la messagerie, circulation de documents, travail coopératif en temps partagé).

c. Extranet

Un extranet est une extension du système d'information de l'entreprise offrant à des partenaires situés au-delà du réseau un accès privilégié à certaines ressources informatiques de l'entreprise via une interface Web.

I.1.1.4 Classement par relation fonctionnelle

Suivant la relation fonctionnelle (hiérarchique presque) entre les éléments constituant le réseau, on peut identifier deux grandes **architectures réseaux** :

a. Architecture Client / Serveur

L'architecture **client / serveur** désigne un mode de communication entre plusieurs ordinateurs d'un réseau. On y distingue un ou plusieurs **clients** du **serveur** généralement unique. Ainsi, chaque client peut envoyer des requêtes à un serveur qui a pour rôle d'être passif dans l'attente d'une requête client.

Le serveur peut être divisé en plusieurs entités plus spécialisées créant ainsi plus de deux types d'ordinateurs et c'est ce qu'on appelle l'architecture multi-tiers. A titre d'exemple, on peut imaginer un serveur web qui lui-même est client d'un serveur d'application qui à son tour est le client d'un serveur de base de données, etc.

Pour pouvoir communiquer, le client et le serveur doivent utiliser le même protocole de communication.

b. Architecture Peer to Peer

Dans le modèle Peer To Peer, ou Pair à Pair, on s'oppose radicalement au modèle Client / Serveur qui distingue plus d'une catégorie d'ordinateurs). En effet, dans le P2P, tous les éléments du réseau sont égaux et sont tour à tour client et serveur.

I.2.2 Le web

I.2.2.1 Présentation du web

Né en suisse en 1990, le web représente le véritable point de départ de la démocratisation d'Internet.

En effet avant cette date les services d'Internet étaient essentiellement à la base d'UNIX, et la recherche des informations sur Internet supposait la connaissance des commandes de ce système.

Les utilisateurs connectés à Internet peuvent désormais avoir accès à des grandes quantités de documents de façon plus au moins interactives par des interfaces conviviales.

Dès lors l'information existante sur les réseaux est devenue accessible aux personnes pas Forcement initiées au transfert de données, cet accès se fait tout simplement à l'aide de la souris en cliquant sur les liens hypertexte ou des icônes particulières via un navigateur (browsers) standards quel que soit la plateforme cliente.

Ainsi le web deviendra de plus en plus un outil de masse avec le développement et l'apparition des pc connectés à Internet dans les foyers.

I.2.2.2 Fonctionnement du web

Le web est fondé sur le modèle **Client/serveur**. Dans ce modèle, deux programmes, le client web et le serveur web, travaillent ensemble pour accomplir une tâche spécifique en saisissant une **URL (Uniform Resource Locator)** dans la barre d'adresse.

Le serveur web Est un programme capable de traiter des requêtes HTTP issues des clients web.

Le client web ou navigateur (appelé **Browser** en anglais) est un programme qui permet d'afficher des pages web et de naviguer entre elles. Le rôle d'un navigateur web est principalement de permettre la consultation des informations disponibles sur le web. Pour cela, il suit le principe illustré dans la figure suivante :

Réponse du serveur : réponse du http contenant le code HTML de la page et des entêtes http



Requête http : le client demande au serveur une page (composée

D'entités, de l'URL de la page et éventuellement de paramètres)

Figure I.6 La navigation web.

I.2.2.3 Les composants du web

- le web est un système hypermédia distribué fonctionnant en mode client/serveur sur Internet.
- Il permet de mettre des informations sous forme de documents hypertextes.
- Pour accéder au web, il est nécessaire de disposer d'un logiciel appelé navigateur web.
- L'accès à un document est exprimé sous forme d'url (uniforme resource locator).

Les clients et les serveurs dialoguent sur le web en utilisant le protocole http (hyper text transfert Protocol). Pour mettre un document à disposition sur le web il est nécessaire de l'écrire dans le langage html (HyperText markup language).

a. L'hypertexte

L'hypertexte est une représentation de l'information, sous la forme d'un réseau, de bien que les utilisateurs peuvent consulter de manière non linéaire.

Prenons l'exemple de deux livres un roman et un dictionnaire, le roman se parcourt ligne par ligne, page par page, par contre dans un dictionnaire, la consultation n'est pas linéaire la définition d'un mot renvoie à un autre mot qui lui-même renvoie à un autre mot c'est un système hypertexte manuel.

b. L'hypermédia

Méthode de représentation des informations en unité numérique connectées par des liens, des informations peuvent être présentes de différentes façons, par exemple : une documentation peut être exécutable seulement on apparaît comme un texte contenant des graphiques, des sons, des clips vidéo, des animations ou des images.

c. L'adresse URL

L'URL (Uniform Resource Locator) est un format de nommage universel permettant d'identifier d'une manière unique une ressource sur le web. Il se décompose en cinq parties :

- Le nom du protocole : c'est le langage utilisé pour communiquer sur le réseau. Le protocole le plus largement utilisé est le protocole http, un protocole permettant d'échanger des pages web au format html, de nombreux protocoles sont utilisés (ftp, news, mail, Gopher).
- Identifiant et mot de passe : permet de spécifier les paramètres d'accès à un serveur sécurisé.
- Le nom du serveur : il s'agit d'un nom de domaine, noter qu'il est toutefois possible d'utiliser l'adresse IP dans celui-ci ; mais l'url devient tout de suite beaucoup moins lisible.
- Le numéro de port : il s'agit d'un numéro associé à un service permettant au serveur de savoir quel type de ressource est demandé.

- Le chemin d'accès à la ressource : cette dernière partie permet au serveur de connaître l'emplacement auquel la ressource est située .c'est-à-dire de manière générale l'emplacement (répertoire) et le nom du fichier demande.

d. Requête http

Une requête http est un ensemble de lignes envoyées au serveur par le navigateur .elle

Comprend :

- Une ligne de requête : c'est une ligne précisant le type de document demandé, la méthode qui doit être appliquée, et la version du protocole utilise. La ligne comprend trois éléments devant être séparés par un espace :

- La méthode.
- L'URL.
- La version du protocole utilisé par le client.

- Les champs d'en-tête de la requête : il s'agit d'un ensemble de lignes facultatives permettant de donner des informations supplémentaire sur la requête et/ou le client (navigateur, système d'exploitation,...) chacune de ces lignes est composée d'un nom qualifiant le type d'en-tête, suivie de deux points [:] est de valeur de l'en-tête.

- Le corps de requête : c'est un ensemble de ligne optionnelles doivent être séparées des lignes précédentes par une ligne vide et permettant par exemple un envoi de données par la commande post lors de l'envoi de données au serveur par un formulaire. À titre d'exemple :
`https://www.google.com/`

e. Réponse http

Une réponse http est un ensemble de lignes envoyées au navigateur par le serveur, elle comprend :

- Une ligne de statut : c'est une précision de la version du protocole utilisé est l'état du traitement de la requête à l'aide d'un code et d'un texte explicatif.

- Les champs d'en-tête de la réponse : Il s'agit d'un ensemble de lignes facultatives permettant de donner des informations supplémentaires sur la réponse et /ou le serveur. Chacune de ces lignes est composée d'un qualifiant, le type d'en -tête, suivi de deux point et de la valeur de l'en-tête.

Le corps de la réponse : contient le document demandé.

f. Le langage html

Les pages sur le www comportent du texte, des liens hypertextes, des objets multimédia, des formulaires. Afin que tout navigateur soit capable d'afficher une page web, un langage universel non propriétaire de publication est utilisé html. Un document html comporte d'une part le texte qui sera affiché dans le navigateur, et d'autres parts des balises ou tage.

h. Le langage XML

XML est en quelque sort un langage html amélioré permettant de définir de nouvelles balises, il s'agit effectivement d'un langage permettant de mettre en forme des documents grâce à des balises (markup).

Contrairement à html, qui est considéré comme un langage défini et figé (avec un nombre de balise limité), XML peut être considéré comme un métalangage permettant de définir d'autres langages.

La force de XML réside dans sa capacité à pouvoir décrire n'importe quel domaine de données grâce à son extensibilité.

j. Le navigateur web

Les navigateurs sont des applications en mode texte, ou en mode graphique destinés à la lecture et l'affichage de document html, c'est le seul logiciel directement manipulé par l'utilisateur. Grâce au navigateur, on peut afficher différents types de données (texte ou image) et communiquer avec le serveur http, quelle que soit la machine qui l'héberge sur lequel un navigateur est disponible.

h. Le serveur web

Un serveur web est un logiciel permettant à des clients d'accéder à des pages web, c'est-à-dire en réalité des fichiers html, à partir d'un navigateur installé sur leur ordinateur distant.

Un serveur web est donc un simple logiciel capable d'interpréter les requêtes http et de fournir une réponse avec ce même protocole. Les principaux serveurs web sur le marché sont :

- Apache.
- Microsoft IIS (Internet Information Server).

- Microsoft PSW (Personnel Web Serveur).

I.2.3 Le modèle Client / Serveur[5]

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.

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 connexions 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.

Il existe une grande variété de logiciels serveurs et de logiciels clients en fonction des besoins à servir : un serveur web publie des pages web demandées par des navigateurs web ; un serveur de messagerie électronique envoie des mails à des clients de messagerie ; un serveur de fichiers permet de stocker et consulter des fichiers sur le réseau ; un serveur de données à communiquer des données stockées dans une base de données, etc.

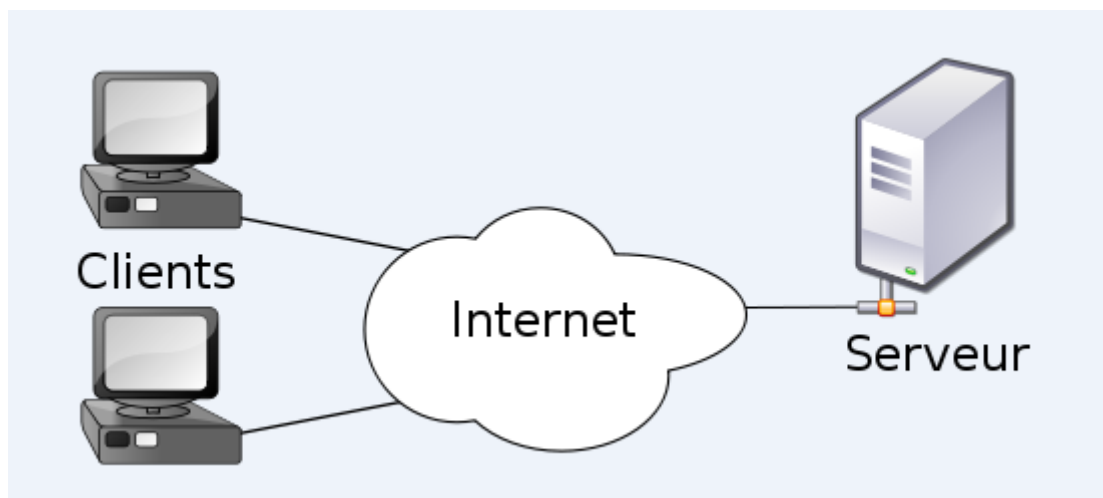


Figure I.7 Architecture client-serveur.

Client : C'est une entité (processus, programme, ordinateur, etc.) qui demande l'exécution d'une opération à une autre entité par envoi d'un message contenant le descriptif de l'opération à exécuter et attendant la réponse à cette opération par un message en retour.

Serveur : C'est une entité (processus, programme, etc.) qui accomplit une opération sur demande d'un client et transmet la réponse à ce client.

Requête : Message transmis par un client à un serveur décrivant l'opération à exécuter pour le compte du client.

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

I.2.3.1 Topologie des architectures client-serveur de données

On distingue trois types d'architecture client-serveur :

- Architecture client-serveur à deux niveaux.
- Architecture client-serveur à trois niveaux.
- Architecture multi niveaux.

a. Architecture client/serveur à deux niveaux

L'architecture client/serveur à deux niveaux (2 tiers) est l'architecture la plus classique dont la communication se fait directement entre le client et le serveur en utilisant deux approches.

Dans la première, les traitements sont assurés par le client et les données sont gérées par le serveur. Par contre dans la deuxième approche les traitements et les données de l'interface utilisateur sont séparés, c'est le serveur qui assure les traitements, le client s'occupe seulement de l'interface utilisateur.

L'architecture client/serveur à deux niveaux se schématise comme suit :

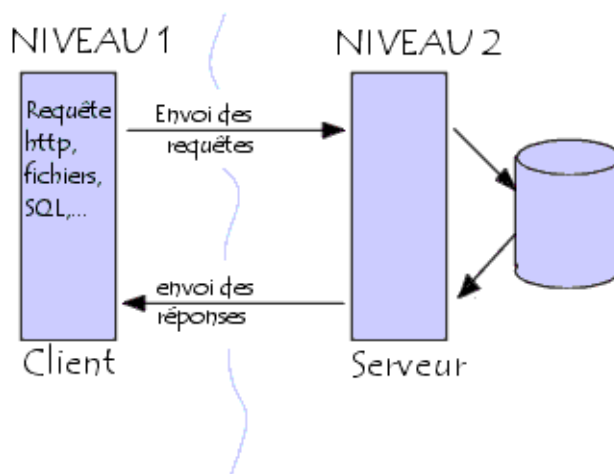


Figure I.8 Architecture 2-tiers.

b. Architecture client/serveur à trois niveaux

Dans l'architecture à trois niveaux, il existe un niveau intermédiaire. C'est une architecture partagée entre :

- Le demandeur de ressources qui est le client.
- Le fournisseur de ressources qui est le serveur d'application (Middleware) en faisant appel à un autre serveur.
- Le fournisseur de services au premier serveur (serveur secondaire qui est généralement un serveur de bases de données).

L'architecture à trois niveaux est recommandée dans des applications possédant les caractéristiques suivantes :

- Une plus grande flexibilité (souplesse).
- Une plus grande sécurité (la sécurité peut être définie pour chaque service).
- De meilleures performances (les tâches sont partagées).
- L'installation d'un gestionnaire de base de données n'est obligatoire que sur le serveur.
- Indépendance des couches par rapport à la localisation physique ; le niveau traitement pouvant être redistribué à l'infini, ce qui permet le déploiement d'applications à grande échelle.

L'architecture client/serveur à trois niveaux se présente comme suit :

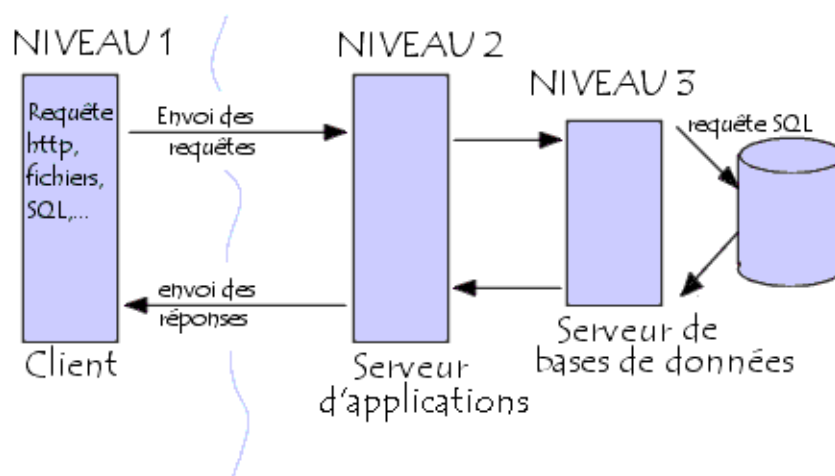


Figure I.9 Architecture 3-tiers.

Middleware : C'est le logiciel situé au milieu qui assure les dialogues entre les clients et le serveur. Il est souvent hétérogène, en d'autres termes il constitue l'ensemble des services logiciels construits au-dessus du protocole de transport afin de permettre l'échange de requête-réponse de manière transparente en cachant l'hétérogénéité des composants mis en jeu (SGBD, réseau, etc.)

c. Architecture multi niveaux

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

I.2.4 Outils de développement d'une application 3-tiers

Il existe une multitude d'outils pour la mise en œuvre des applications 3-tiers, nous présentons dans cette section les principaux outils utilisés.

I.2.4.1 La plateforme .NET

.NET (prononcez «Dotnet») est un standard proposé par la société Microsoft, pour le développement d'applications d'entreprises multi-niveaux, basées sur des composants. Microsoft .NET constitue ainsi la réponse de Microsoft à la plate-forme J2EE de Sun. La plate-forme .NET a été élaborée en s'appuyant sur une communauté d'utilisateurs et a abouti à l'élaboration de spécifications. Ces spécifications ont été ratifiées par un organisme international de standardisation, l'ECMA (European Computer Manufacturers Association), ce qui en fait un standard. Ainsi l'effort de standardisation a permis l'émergence de plates-formes

Chapitre I Concepts de base sur le développement d'une application 3-tiers

portées par des entreprises tierces et disponibles sous un grand nombre de systèmes d'exploitation. On parle généralement de «Framework» (traduisez «socle») pour désigner l'ensemble constitué des services (API) offerts et de l'infrastructure d'exécution.

b. Architecture .NET

Un moteur d'exécution, appelé **CLR** (Common Language Runtime), permettant de compiler le code source de l'application en un langage intermédiaire, baptisé **MSIL** (Microsoft Intermediate Language) et agissant telle la machine virtuelle Java. Lors de la première exécution de l'application, le code MSIL est à son tour compilé à la volée en code spécifique au système grâce à un compilateur JIT (Just In Time).

-un environnement d'exécution d'applications et de services web, appelé **ASP .NET** ;

-un environnement d'exécution d'applications lourdes, appelé **WinForms**.

Des **services**, sous forme d'un ensemble hiérarchisé de classes appelé **Framework Class Library** (FCL).

La FCL est une librairie orientée objet, fournissant des fonctionnalités pour les principaux besoins actuels des développeurs. Le SDK (Software Development Kit) fournit une implémentation de ces classes.

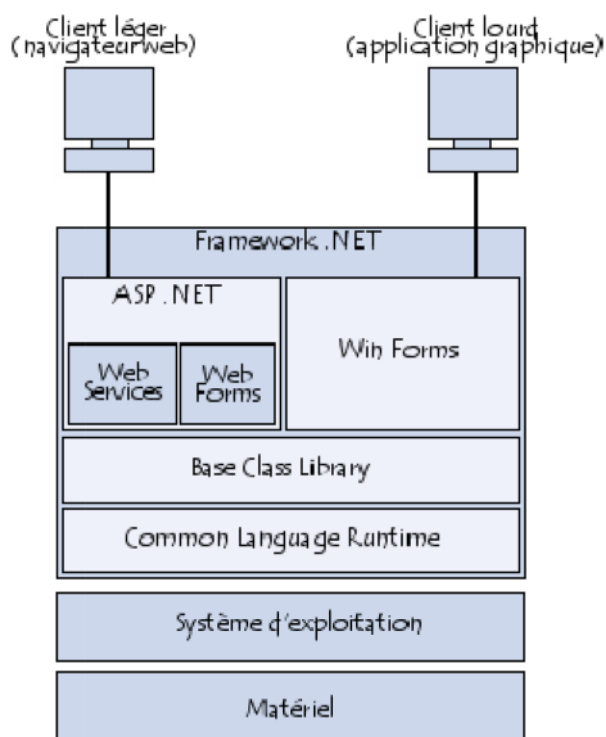


Figure I.10 Architecture Dot NET.

a. Les langages de programmation

Grâce au CLR, la plate-forme .NET est indépendante de tout langage de programmation et supporte nativement un grand nombre de langages de programmation, parmi lesquels : le langage c #.

I.2.4.2 La plateforme J2Entreprise Edition

Java Enterprise Edition, ou **Java EE** (anciennement **J2EE**), est une spécification pour la technique Java de Sun plus particulièrement destinée aux applications d'entreprise. Ces applications sont considérées dans une approche multi-niveaux. Dans ce but, toute implémentation de cette spécification contient un ensemble d'extensions au *Framework* Java standard (JSE, *Java Standard Edition*) afin de faciliter notamment la création d'applications réparties.

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

- Les spécifications du serveur d'applications, c'est-à-dire de l'environnement d'exécution: J2EE définit finement les rôles et les interfaces pour les applications ainsi que l'environnement dans lequel elles seront exécutées. Ces recommandations permettent ainsi à des entreprises tierces de développer des serveurs d'applications conformes aux spécifications ainsi définies, sans avoir à redévelopper les principaux services.
- Des services, au travers d'API, c'est-à-dire des extensions Java indépendantes permettant d'offrir en standard un certain nombre de fonctionnalités. Sun fournit une implémentation minimale de ces API appelée J2EE SDK (J2EE Software Development Kit).

a. Les composants de J2EE

J2EE fournit:

- Un modèle de développement de composants Web (Servlets, JSP) et de composants métier (EJB) sous la forme d'APIs JAVA.

Chapitre I Concepts de base sur le développement d'une application 3-tiers

- Un ensemble de services (JDBC, JTA, JNDI, JMS, RMI / IIOP, JavaMail, XML), utiles aux composants, sous la forme d'APIs JAVA.
- Un modèle de création de modules Web (.war) de modules EJB (.jar) et de modules d'entreprises (.ear), associés à des descripteurs de déploiement au format XML, utiles pour le déploiement des applications d'entreprise.
- Des conteneurs Web et EJB pour la mise en œuvre des composants.

a. Architecture J2EE

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

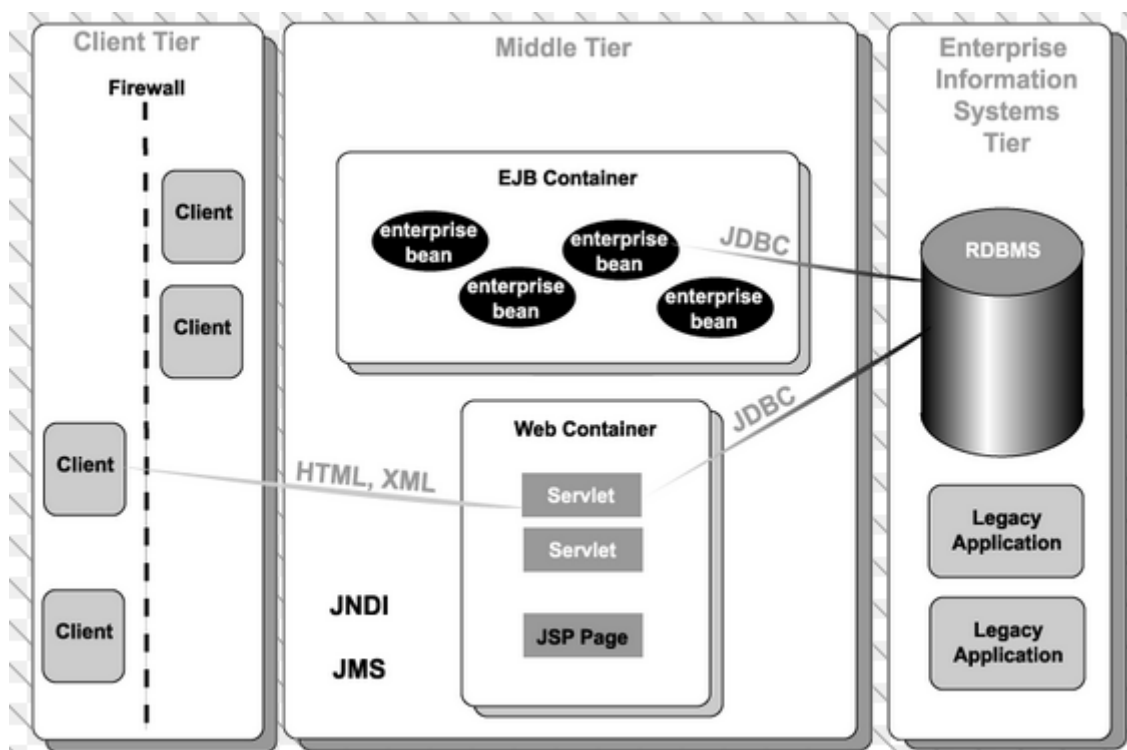


Figure I. 11 Architecture J2EE.

a. Environnement d'exécution de J2EE

J2EE se contente de regrouper un certain nombre d'API, mais il présente également la caractéristique remarquable de faire abstraction de l'**infrastructure d'exécution**.

Informatique distribuée "traditionnelle" = souvent problèmes liés non pas à la logique propre à l'application mais à la mise en œuvre de services complexes (threading, transactions, sécurité...).

J2EE introduit la notion de **conteneur**, et via les **API J2EE**, il élabore un **contrat** entre le conteneur et les applications.

C'est le vendeur du conteneur qui se charge de mettre en œuvre les services pour les développeurs d'applications J2EE, dans le respect des **standards**.

e. Langages de programmation

Le langage Java est, bien sûr, le langage des développements J2EE.

Né en 1995, le langage Java est aujourd'hui très largement utilisé et apprécié des développeurs.

d. Avantage de J2EE

- Pas d'installation sur les postes clients (hormis le navigateur lui-même)
- Facilité de mise-à-jour et maintenance.
- Protocole d'échange standardisé (HTTP, HTTPS)
- Approche multi-plate-forme et multi-éditeurs
- Plus de 30 éditeurs implémentent totalement ou partiellement J2EE
- Existence d'implémentations open source (JBoss, Tomcat, ...)
- Portabilité entre implémentations J2EE

I.2.4.3 PHP

Le langage PHP est principalement un langage de programmation web côté serveur ce qui veut dire que c'est le serveur (la machine qui héberge le site web en question) qui va interpréter le code PHP et générer du code qui pourra être interprété par un logiciel. Le plus souvent, le code généré est le HTML afin d'être lu par un navigateur mais il peut être utilisé pour d'autres langages ou formats tels que WML, GIF, PDF, SVG, etc.

Il a été conçu pour permettre la création d'applications dynamiques, le plus souvent dédiées au web. PHP peut être installé sur les principaux serveurs web du marché, les plus répandus étant IIS et Apache. Ce couplage permet de récupérer des informations issues d'une base de données, d'un système de fichiers (contenu de fichiers et de l'arborescence) ou plus simplement des données envoyées par le navigateur afin d'être interprétées ou stockées pour une utilisation ultérieure. C'est un langage peu typé et souple et donc facile à apprendre par un débutant mais, de ce fait, des failles de sécurité peuvent rapidement apparaître dans les applications. Son utilisation commence avec le traitement des formulaires puis par l'accès aux bases de données. L'accès aux bases de données est aisé une fois l'installation des modules correspondant effectuée sur le serveur. La force la plus évidente de ce langage est qu'il est devenu au fil du temps un incontournable des offres d'hébergement.



Figure I.12 Schéma fonctionnel du PHP.

Fonctionnement

Dans une utilisation Web, l'exécution du code PHP se déroule ainsi : lorsqu'un visiteur demande à consulter une page Web, son navigateur envoie une requête au serveur http correspondant. Si la page est identifiée comme un script PHP (généralement grâce à l'extension .php), le serveur appelle l'interpréteur PHP qui va traiter et générer le code final de la page (constitué généralement d'HTML ou de XHTML, mais aussi souvent de CSS et de JS). Ce contenu est renvoyé au serveur HTTP, qui l'envoie finalement au client.

I.2.4.4 Comparaison des trois outils

Nous allons présenter les différents avantages et inconvénients de trois outils de développements, les différents langages utilisés par ces derniers et le taux d'utilisations à l'échelle internationale en mars 2014 :

Chapitre I Concepts de base sur le développement d'une application 3-tiers

	Langage	Avantages	Inconvénients	Popularité des langages de programmation en mars 2014
J2EE	JAVA	<ul style="list-style-type: none"> -approche multi-plateformes et multi-éditeurs -existence d'implémentation open source (Tom cat) -portabilité entre implémentation J2EE -nombreuses références clients -modèle de programmation plus avancée. 	<ul style="list-style-type: none"> -architecture complexe nécessitant un temps d'apprentissage conséquent -les services web ne sont supportés que dans la version J2EE 1.4 -de nombreuses solutions propriétaires implémentent toutefois les services web. 	<p>16.406%</p> <p>[http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html]</p>
PHP	HTML XML JavaScript	<ul style="list-style-type: none"> -La gratuité et la disponibilité du code - La simplicité d'écriture de scripts - Couplé à un Framework + MVC, on arrive à de très bonnes structures. - Présent chez de très nombreux hébergeurs. - Le temps de développement est diminué 	<ul style="list-style-type: none"> - Une gestion partielle de la POO (bien que ça se soit grandement amélioré avec la version 5) - Aucun typage des variables, c'est très pénible pour la documentation et les contrôles des fonctions/méthodes. - Des noms de 	<p>3.698%</p> <p>[http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html]</p>

Chapitre I Concepts de base sur le développement d'une application 3-tiers

		par rapport à bon nombre de langages.	fonctions par cohérents (lacunes du passé) - Nécessite l'activation de caches PHP si on veut parler un minimum de performances (normal, la compilation des sources à chaque page ou presque est un processus couteux) - Gestion de l'Unicode un peu foireuse (expérience personnelle avec l'UTF-8 + BOM) - Nécessite une bonne connaissance des failles.	
--	--	---------------------------------------	---	--

DOTNET	C#	<ul style="list-style-type: none"> -support natif des services web -multiplicité des langages de programmation -indépendance vis-à-vis du langage de développement -simplicité d'utilisation -efficacité en termes de productivité de développement -interopérabilité entre les langages -gestion des versions des composants exécutables. 	<ul style="list-style-type: none"> -changement technologique important pour les développeurs VB et ASP actuels -limité à la plateforme Windows -pas d'équivalent dans .net des EJB -migration d'applications Windows existantes pas forcément triviales. 	<p style="text-align: center;">5.572%</p> <p>[http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html]</p>
--------	----	---	--	--

Tableau I.1 Comparaison des outils de programmation

I.3 Conclusion

Les progrès réalisés dans les technologies de l'information nous ont fait entrer dans une nouvelle ère, celle de l'informatique ouverte au grand public.

Au cours de ce chapitre, nous avons présenté certains concepts des nouvelles technologies de l'information et de la communication, entre autre : les réseaux informatique, l'internet et le web, l'architecture client/serveur...Qui nous aiderons à mieux comprendre notre mode d'opération et les notions fondamentales pour mener à bien notre application

Chapitre II : Description de l'organisme d'accueil et étude de l'existant

II.1 Introduction

L'objectif de ce chapitre est d'introduire notre organisme d'accueil par une brève description et de cerner notre champ d'étude à travers une étude détaillée du système existant. Le présent chapitre est organisé comme suit :

- Présentation de l'organisme d'accueil.
- L'étude de l'existant.

II.2 Présentation de l'organisme d'accueil

Dans cette section, nous allons présenter l'organisme, ces différentes missions et un petit aperçu de la situation informatique.

II.2.1 Historique du groupe CMA CGM

Le groupe **CMA-CGM**, abréviation de **Compagnie maritime d'affrètement - Compagnie générale maritime**, est issu de la fusion en 1996 de la Compagnie générale maritime (CGM) et de la Compagnie maritime d'affrètement (CMA).

Le nom de la compagnie CMA- CGM est récent, mais ses origines remontent à l'année 1851 quand les Messageries Maritimes (MM) ont été fondées. Vers la même époque en 1855 une autre compagnie, la Compagnie Générale Maritime (CGM), fut fondée puis renommée en Compagnie générale transatlantique en 1961. La Compagnie générale transatlantique surnommée la French Line, a assuré le transport transatlantique régulier de passagers avec des navires comme *le « Normandie »* ou *le « France »*. Ces deux sociétés fusionneront en Compagnie Générale Maritime en 1973. En 1996, CGM est privatisé. D'un autre côté Jacques Saadé, créa la CMA (Compagnie Maritime d'Affrètement) en 1978.

La Compagnie Maritime d'Affrètement (CMA) et la Compagnie Générale Maritime (CGM) fusionnent pour former la CMA CGM. La fusion est finalisée en 1999.

Sous ce même nom la compagnie rachètera en 1998 Australian National Lines (ANL), puis le concurrent Delmas fondé en 1867 basé au Havre appartenant à Bolloré group en septembre 2005.

Ces acquisitions placent la compagnie française basée à Marseille comme la troisième plus grande compagnie maritime de transport de conteneurs au monde derrière A.P. Moller-Mærsk et Mediterranean Shipping Company.

En 2006 la compagnie relance une activité de transport de passager haut-de-gamme avec la Compagnie du Ponant, revendu en 2012 au fonds d'investissement Bridge point. Il est également possible d'embarquer comme passager sur ses cargos.

En 2012, CMA-CGM comptait :

- 414 navires
- 2 155 000 conteneurs evp.
- 160 000 conteneurs réfrigérés (ou reefer).

Le groupe possède des navires performants, comme le *CMA CGM Christophe Colomb* (13 800 evp) le deuxième porte conteneur mondial le *CMA CGM Jules Verne* (16000 evp) et le *CMA CGM Andromeda* (11 400 evp), équipés des dernières technologies environnementales et permettant de réduire les émissions atmosphériques. CMA-CGM préserve l'environnement marin en mettant en place des mesures impliquant notamment la gestion écologique des eaux de ballast et la prévention des pollutions marines.

Le groupe CMA-CGM comporte plusieurs agences en Algérie :

- CMA-CGM Alegria quartier des affaires BabEzouar
- CMA-CGM Alegria/Skikda
- Ghazaouet
- Annaba
- Bejaia
- Mostaganem
- Oran

Dans notre cas on s'intéresse plutôt à l'agence CMA-CGM de Bab Ezzouar.

II.2.2 Organigramme général

La figure suivante illustre l'organigramme du groupe CMA-CGM

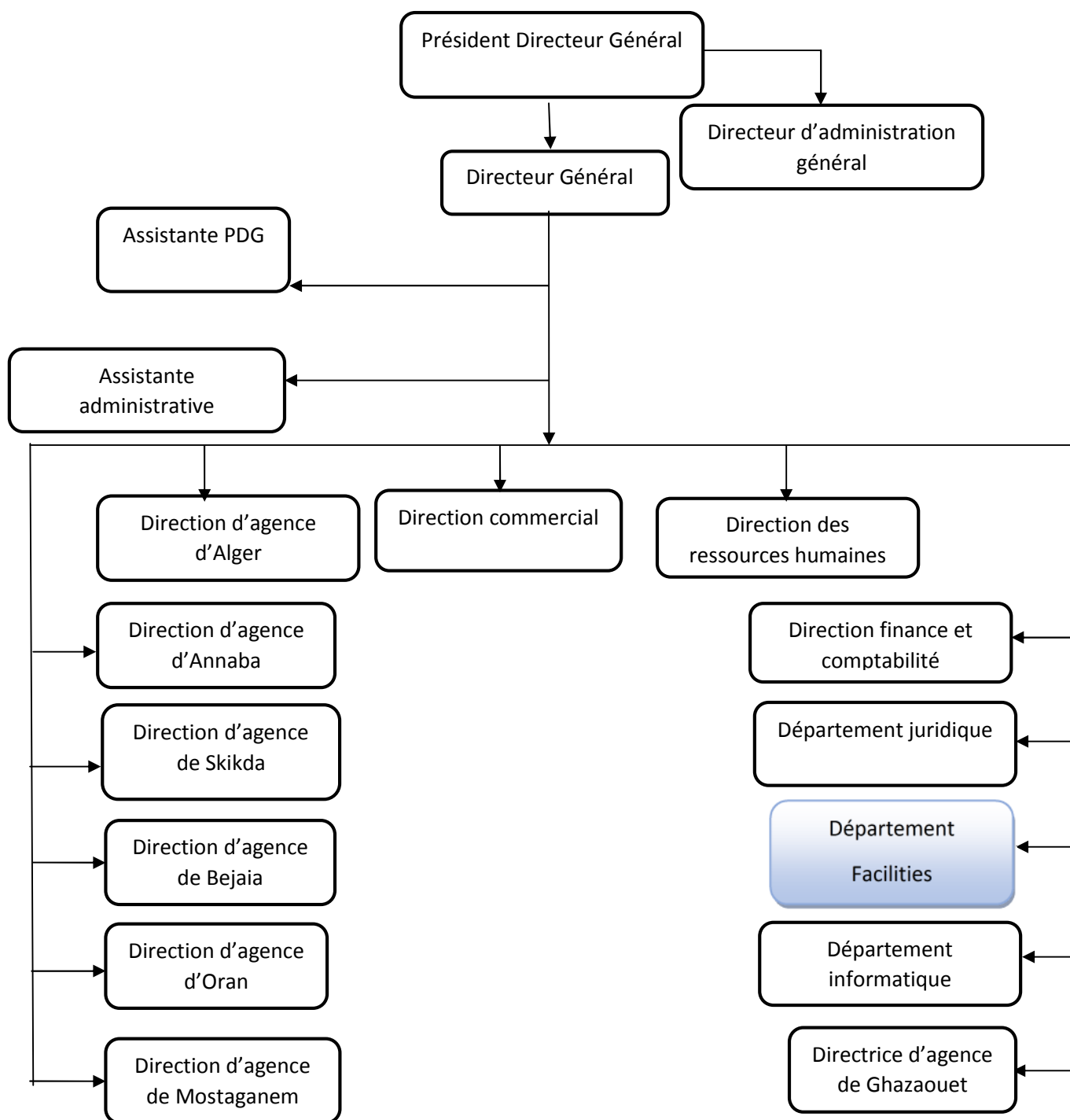


Figure II.1 Organigramme générale du groupe CMA-CGM.

 : Champ d'étude

II.2.3 Missions de CMA-CGM

Le groupe CMA-CGM est le troisième armateur mondial de transport maritime en conteneurs et le premier français.

Son offre globale de transport intègre :

1. le transport maritime,
2. la manutention portuaire
3. la logistique terrestre.

CMA-CGM propose des solutions de transport intermodal complètes, combinant le maritime, le rail, le fluvial et le routier. Ses filiales spécialisées dans l'intermodal sont : CMA Rail / Rail Link Europe, River Shuttle containers (RSC), Land Transport International (LTI) et CMA CGM Logistics.

II.2.4 Situation informatique

Le département facilities dispose de matériels suivant :

Aspect matériel	Aspect logiciel
-Deux cent Micro-ordinateurs dont PC Bureau, Micro Portables et Trois Serveurs ; -Cent Micro Portable de marque Dell: Une RAM (Random Access Memory) de 4GB -Processeur : Intel core i5 3320M -Fréquence d'horloge : 1600MHz -Système d'exploitation : Windows Seven -Imprimante Laser Jet HP et Laser Toner GT	-Microsoft office Word 2007, Excel 2007

Tableau II.1 Situation informatique.

II.2.5 Problématiques

Le suivi des matériels utilisables consommables (papier, stylo, chemise etc.) et non consommables (voitures, micro-ordinateur etc.) dans l'entreprise pose des problèmes au niveau du service facilities. Parmi ces problèmes on cite :

- La non existence d'une base de données pour permettre l'enregistrement et le meilleur suivi des stocks en temps réel.

-La mal organisation de la procédure d'affectation .s

II.2.6 Objectifs Pour résoudre les problèmes cités, nous nous proposons de mettre en place une application automatisée de suivi des stocks des articles utilisable par ce groupe et pour cela les objectifs de notre étude sont les suivants :

- La mise en place d'une base de données des articles utilisés dans l'entreprise ;
- L'automatisation des différentes tâches de la gestion des stocks, particulièrement la gestion d'achat, la gestion des affectations et l'inventaire.

II.3 Étude de l'existant

Cette partie (Etude ou Analyse du Système Existant) nous permettra de bien comprendre l'organisation de l'organisme et mieux définir et analyser la situation existante au sein du département facilities.

L'analyse de l'existant permet :

- La définition et la représentation du domaine d'étude.
- L'étude et l'analyse des postes de travail.
- L'étude et l'analyse des documents et des fichiers (registres)
- L'étude et l'analyse des procédures.

II.3.1 L'organigramme du champ d'étude

Le département facilities se charge principalement de l'achat des articles pour l'entreprise, les différents acteurs participants sont : Facility manager, Acheteur et Chef département des moyens généraux.

La figure suivante illustre l'organigramme du département Facilities :



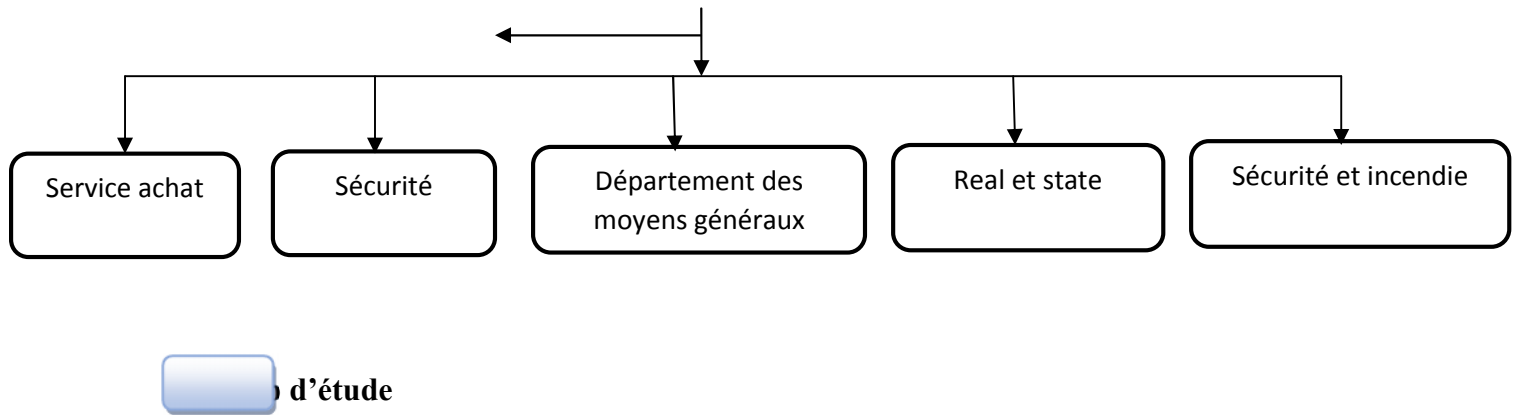


Figure II.2 Organigramme du champ d'étude.

II.3.2 Diagramme des flux

II.3.2.1 Définition

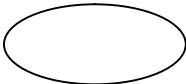

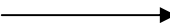
Le diagramme des flux est une représentation simple de l'organisation d'une entreprise, il fait apparaître les différents échanges entre les différents acteurs du domaine étudié ainsi qu'avec leur environnement.

II.3.2.2 Les concepts de base

Acteur : un acteur représente une unité active, c'est un agent capable d'échanger des informations avec d'autres agents .Il peut être interne ou externe. Il est interne s'il appartient au domaine d'étude et externe dans le cas contraire.

Flux d'information: il représente un échange de données entre deux acteurs dans le cadre du système d'information concerné.

Légende

Acteur interne	
Acteur externe	
Flux d'information	

II.3.2.3 Description des acteurs internes

Acheteur

Il est chargé de recevoir et d'établir les différents documents nécessaires pour passer une commande et réception des articles.

Facility manager

Le Facility manager s'occupe de la signature des bons de commandes, bons de livraison ainsi les demandes des besoins.

Chef de département des moyens généraux

Le chef de département des moyens généraux a pour mission d'effectuer l'inventaire.

Coordinateur

La principale tâche du coordinateur est la réception et remise de la facture du service comptabilité et finances au département des moyens généraux

II.3.2.4 Présentation des acteurs externes

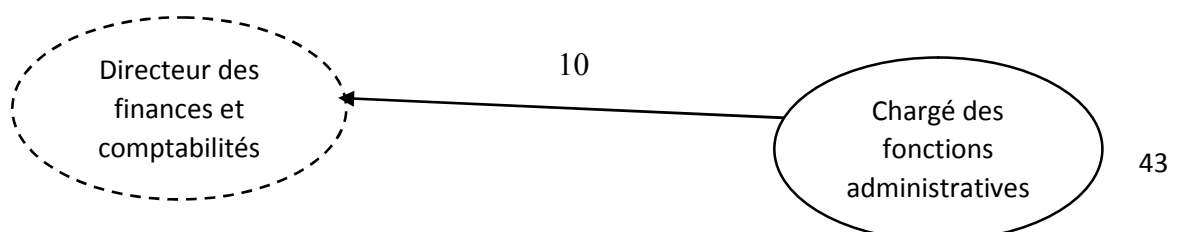
Les acteurs externes figurants dans le diagramme de flux d'information sont :

- Directeur des finances et comptabilités
- Fournisseurs
- Chef du département informatique

II.3.2.5 Description des flux

Numéro	Désignation des flux
01	Demande des besoins
02	Facture pro forma (Fiche descriptive +Devis)
03	Bon de commande non signé
04	Bon de commande signé
05	Bon de livraison non signé
06	Bon de livraison signé
07	Facture
08	Demande de paiement non signée
09	Demande de paiement signée
10	Dossier (copie de bon de commande, bon de livraison, facture et demande de paiement)
11	Chèque

Tableau II.2 Description des flux.



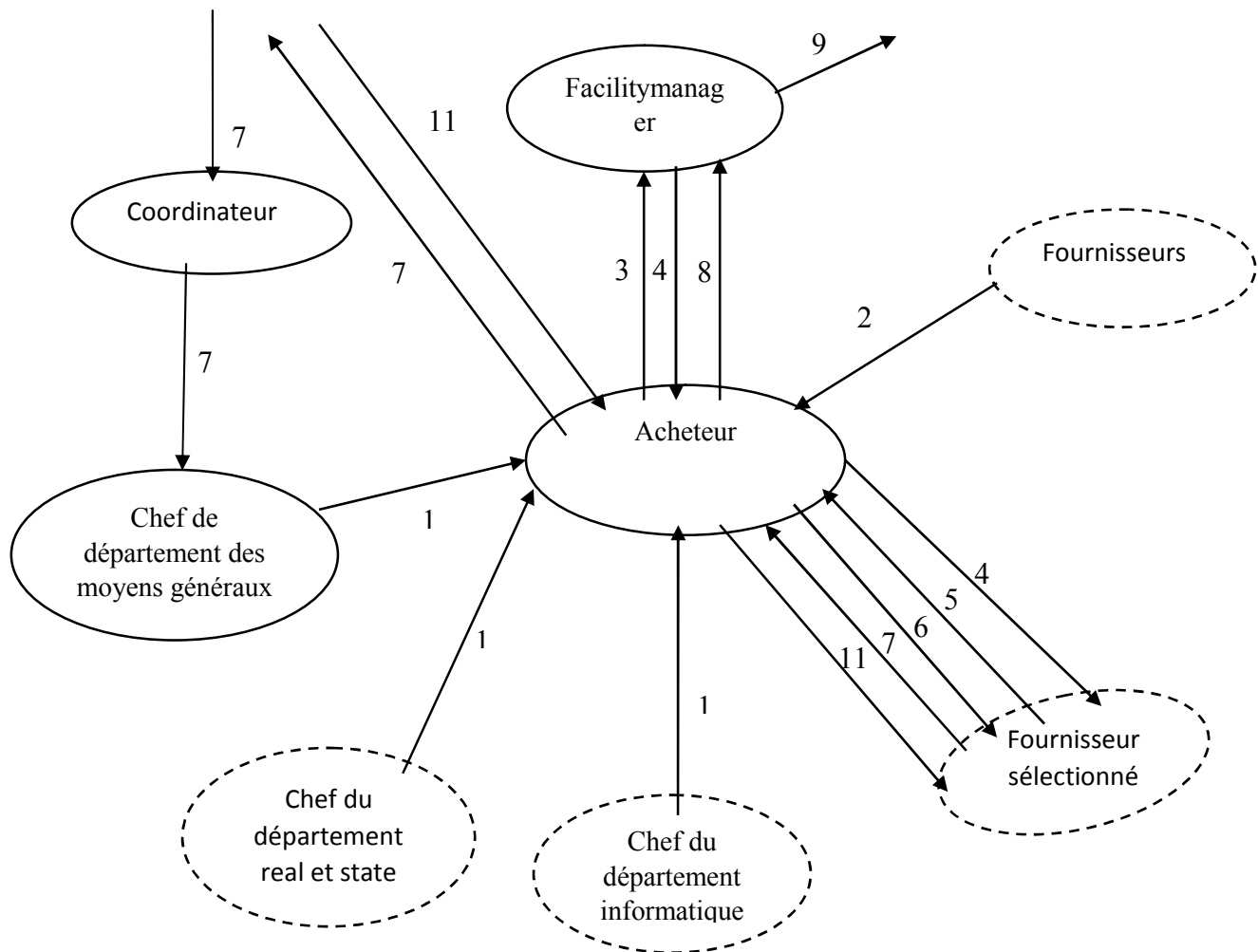


Figure II.3 Diagramme des flux.

II.3.3 Etude des postes de travail

Dans cette partie nous allons définir les différents postes de travail participant aux procédures achat, affectation et inventaire.

II.3.3.1. Définition d'un poste de travail

Un poste est une composante organisationnelle situé sur un site. Il est constitué d'un ou plusieurs agents ayant un certain nombre de tâches à exécuter et des informations à collecter ou à transmettre. Dans ces postes il s'agit d'énumérer les postes directement concernés par le champ d'étude.

II.3.3.2 Liste des postes de travail

Notre domaine d'étude contient trois postes de travail qui sont :

- Département facilities.
- Service achat.

- Département des moyens généraux

Fiche d'étude du poste N°1		
Code du poste :		
Désignation du poste : Facilitymanager		
Service de rattachement : département facilities.		
Responsabilité :		
<ul style="list-style-type: none"> - signature des différents documents (bon de commande, demande de paiement) ; - validation de la demande des besoins. 		
Nombre d'agent occupant le poste : 01		
Tâche du poste		
Désignation	Fréquence	Délai
-validation de la demande des besoins. -signature du bon de commande. -signature de la demande de paiement.	à chaque opération d'achat	Variable

Tableau II.3 Etude du poste « Facility manager ».

Fiche d'étude du poste N°2		
Code du poste :		
Désignation du poste : Acheteur		
Service de rattachement : Service achat.		
Responsabilité :		
<ul style="list-style-type: none"> - établissement des bons de commande ; - signature des bons de livraison ; 		
Nombre d'agent occupant le poste : 01		
Tâche du poste		
Désignation	Fréquence	Délai
-consultation des différents fournisseurs pour choisir un. -établissement d'un bon de commande en 2 exemplaires -Remise des 2 exemplaires au Facilitymanager pour signature -Remise d'un bon de commande signé au fournisseur -signature du bon de livraison.	à chaque opération d'achat	Variable

-réception de la facture.		
-remise du chèque au fournisseur.		

Tableau II.4 Etude du poste « Acheteur ».

Fiche du poste N°3		
Code du poste :		
Désignation du poste : chef du département des moyens généraux.		
Service de rattachement : département des moyens généraux.		
Responsabilité : établissement de la demande des besoins.		
Nombre d'agent occupant le poste : 01		
Tâche du poste		
Désignation	Fréquence	Délai
-établissement de la demande des besoins.	-a chaque opération d'achat.	
-établissement de la liste des articles utilisés.	-annuellement	

Tableau II.5 Etude du poste « Chef du département des moyens généraux ».

II.3.3.3 Documents manipulé par les postes

Documents entrants	
Au poste	Désignation du document
Facility manager	-Bon de commande. -Demande de paiement.
Acheteur	-Facture pro forma (fiche descriptive +devis) -Bon de livraison -Facture -Chèque.
Directeur des finances et comptabilité	-Dossier contenant la facture, le bon de livraison, une copie du bon de commande et la demande de paiement.
Chef du département des moyens généraux	-liste des articles achetés. -facture.
Documents sortant	
Désignation	Vers le poste
Demande des besoins	Acheteur
Bon de commande	Fournisseur

Demande de paiement	Facility manager
Chèque	Fournisseur

Tableau II.6 Documents manipulés par les différents postes.

II.3.4 Etude des documents

L'étude des documents permet de recenser tous les documents manipulés et de les analyser.

Une fiche d'étude sera utilisée à cette fin qui comportera toutes les caractéristiques du document : code, désignation, rôle, nombres d'exemplaires etc.

Chaque information sera décrite par sa taille, son type et des observations s'il ya lieu. Le type de rubriques peut être :

A : Alphabétique, N : Numérique, AN : Alphanumérique, DATE : date

L'objectif de cette étude est de :

- Déceler l'origine et la destination de chaque document,
- Voir si les documents répondent aux besoins pour lesquels ils ont été créés.

Parmi les documents, on distingue :

- **Document interne de position** : on appelle document interne de position, tout document utilisé dans le service même où il a été créé.
- **Document interne de liaison** : on appelle document interne de liaison, tout document transitant entre les services de l'organisme.
- **Document externe** : on appelle document externe, tout document provenant de l'environnement externe de l'organisme ou diffusé vers celui-ci.

II.3.4.1 Liste des documents

Nous avons recensé les documents suivants :

Code	Désignation	Nombre d'exemplaire
BDC	Bon de commande	02
BDL	Bon de livraison	04
DDP	Demande de paiement	01
CHEQ	Chèque	01
FACT	Facture	01
FACTPRO	Facture pro forma	01

Tableau II.7 Liste des documents.

A présent nous présentons la fiche d'étude de chaque document

Fiche d'étude document N°1			
Code document : BDC			
Désignation : Bon de commande			
Nature : interne			
Rôle : passer une commande			
Poste émetteur : acheteur			
Poste récepteur : Fournisseur			
Nombre exemplaire : 02			
Description des données			
Désignation	Type	Taille	Observation
-Numéro de bon de commande	N	10	JJ/MM/AAAA
-Date de bon de commande	DATE	10	
-Nom du fournisseur	A	20	
-Adresse fournisseur	AN	20	
-Adresse du client	AN	40	
-Numéro de téléphone du client	N	10	
-Fax du client	N	10	
-Référence article	AN	20	
-Désignation article	AN	30	
-Quantité commandé	N	10	
-observation	A	50	
Total		234	

Tableau II.8 Etude du document « Bon de commande ».

Fiche d'étude du document N°02
Code du document : BDL
Désignation : bon de livraison
Nature : externe
Rôle : justification de la marchandise
Poste émetteur : fournisseur
Poste récepteur : Acheteur

Nombre d'exemplaire : 04			
Description des données			
Désignation	Type	Taille	Observation
-Numéro de bon de livraison	N	10	JJ/MM/AAAA
-Date de bon de livraison	Date	10	
-Nom de fournisseur	A	20	
-Adresse du fournisseur	AN	30	
-Numéro de téléphone du fournisseur	N	10	
-Numéro de fax du fournisseur	N	10	
-Nom du client	A	20	
-Adresse du client	AN	30	
-Numéro du registre commercial	AN	10	
-Matricule fiscale	N	20	
-Article d'imposition	N	15	
-Référence article	AN	10	
-Désignation article	AN	30	
-Quantité livrée	N	10	
Total		239	

Tableau II.9 Etude du document « bon de livraison ».

Fiche d'étude du document N°03			
Code du document : DDP			
Désignation : demande de paiement			
Nature : interne			
Rôle : paiement de la marchandise			
Poste émetteur : Facilitymanager			
Poste récepteur : chargé des fonctions administratives			
Nombre d'exemplaire : 01			
Description des données			
Désignation	Type	Taille	Observation

-Référence de la demande de paiement	AN	10	JJ/MM/AAAA
-Service émetteur	AN	15	
-Date d'établissement	DATE	10	
-Mois de comptabilisation	AN	10	
-N° de journal et folio	N	08	
-N° de pièce	N	08	
-Nature et n° de pièce	AN	20	
-Règlement	AN	200	
-Montant	N	20	
-Total HT	N	15	
-TVA	N	10	
-Total TTC	N	15	
-Montant en chiffres et en lettres	AN	30	
-Bénéficiaire	AN	30	
-Imputation	AN	20	
-Référence et nature de règlement	AN	10	
Total		421	

Tableau II.10 Etude du document « Demande de paiement ».

Fiche d'étude du document N° 04			
Code du document : FACT			
Désignation : Facture			
Nature : externe			
Rôle : justification de l'achat			
Poste émetteur : fournisseur			
Poste récepteur : Acheteur			
Nombre d'exemplaire : 01			
Description des données			
Désignation	Type	Taille	Observation

-Numéro de la facture	N	10	
-Date de la facture	Date	10	
-Nom de fournisseur	A	20	JJ/MM/AAAA
-Adresse du fournisseur	AN	30	
-Numéro de téléphone du fournisseur	N	10	
-Numéro de fax du fournisseur	N	10	
-Nom du client	A	20	
-Adresse du client	AN	30	
-Numéro du registre commercial	AN	10	
-Matricule fiscale	N	20	
-Article d'imposition	N	15	
-Référence article	AN	10	
-Désignation article	AN	30	
-Quantité livrée	N	10	
-Prix unitaire	N	10	
-Montant HT	N	12	
-Total HT	N	17	
-Total TVA	N	10	
-Total à payer	N	3	
-Timbre	N	20	
Total		301	

Tableau II.11 Etude du document « Facture ».

Code du document : CHEQ			
Désignation : chèque			
Nature : externe			
Rôle : paiement			
Poste émetteur : chef de département des moyens généraux			
Poste récepteur : fournisseur			
Nombre exemplaire : 01			
Description des données			
Désignation	Type	Taille	Observation
-Numéro du compte	AN	08	
-Clé du chèque	N	02	
-Nom du client	A	20	
-Prénom du client	A	20	
-Adresse du client	AN	30	
-Montant du chèque	N	10	
-Date du chèque	Date	10	JJ/MM/AAAA
Total		100	

Tableau II.12 Etude du document « Cheque ».

Fiche d'étude du document N°06			
Code du document : FACTPRO			
Désignation : Facture pro forma			
Nature : externe			
Rôle : description du devis et délai			
Poste émetteur : acheteur			
Poste récepteur : fournisseur			
Nombre exemplaire : 01			
Description des données			
Désignation	Type	Taille	Observation
- Adresse du fournisseur	AN	30	
- Numéro de téléphone du fournisseur	N	10	
- Adresse électronique du fournisseur	AN	30	

-Nom du client	A	20	
- Adresse du client	AN	30	
- Numéro de la facture pro forma	N	10	
- Date d'établissement	DATE	10	JJ/MM/AAAA
- Numéro du bon de commande	N	10	
- Date d'établissement du bon de commande	DATE	10	JJ/MM/AAAA
- Numéro de téléphone du client	N	10	
- Numéro de fax du client	N	10	
- Référence de l'article	AN	10	
- Désignation de l'article	A	20	
- Quantité livré	AN	10	
- Prix unitaire hors taxe	N	10	
- Montant hors taxe	N	10	
- Total hors taxe	N	10	
-TVA	N	06	
-Timbre	N	06	
-Net à payer	N	10	
-Mode de règlement	AN	20	
-Validité de l'offre	AN	30	
- Délai de livraison	AN	30	
- Garantie	AN	30	
Total			

Tableau II.13 Etude du document « Facture pro forma ».

II.3.5 Procédures

Elle est Constituée d'un ensemble de traitements déclenché par un ou plusieurs événements externes. Elle appartient à un seul processus et dans notre cas il existe quatre procédures :

II.3.5.1 Achat

Les différents chefs de sous service du département Facilities envoient des demandes de besoins dans une boite générique que consulte par la suite le Facility manager pour donner son accord et l'acheteur pour enfin consulter les fournisseurs.

A la réception des offres, l'acheteur sélectionne un fournisseur par rapport au délai, qualité et prix de la marchandise, par la suite, il établit un bon de commande en deux exemplaire (un gardé à son niveau et l'autre remis au fournisseur sélectionné) qu'il remet au Facility manager pour signature puis le renvoi au fournisseur. Ce dernier établit et envoi un bon de livraison à l'acheteur pour signature. Après signature celui-ci envoie à son tour le bon de livraison signé pour récupérer la facture accompagné du bon de livraison et de la marchandise.

Après réception de la marchandise l'acheteur établit une demande de paiement qui doit être signé par le Facility manager et au tour du chargé des fonctions administratives de la récupérer pour à la fin établir un dossier contenant une copie du bon de commande, le bon de livraison, la facture et la demande de paiement, enfin l'envoi à son tour au directeur des finances et comptabilités pour régler la facture en préparant un chèque.

II.3.5.2 Gestion des affectations

A la réception de la marchandise chaque service doit recevoir sa liste des articles commandés par rapport à sa demande des besoins, dans ce cas le CDMGX se charge d'effectuer cette tâche.

II.3.5.3 Réapprovisionnement des articles

L'acheteur reçoit le bon de livraison du fournisseur qu'il comparera par la suite avec le bon de commande et si ces derniers ne se correspondent pas, l'acheteur renvoie le bon de commande accompagné du bon de livraison pour signaler le problème puis le fournisseur établit à son tour un autre bon de livraison pour livrer le reste de la commande.

II.3.5.4 Inventaire

A chaque opération d'achat le coordinateur récupère la facture de chez le directeur des finances et comptabilités pour la remettre au chef du département des moyens généraux. Celui-ci calcule les articles existant au sein de l'entreprise et les compare avec les factures précédentes pour établir une liste réelle. Le chef du département des moyens généraux affecte un code barre pour chaque bureau et chaque article de l'entreprise.

II.3.6 Critiques et suggestions

L'étude de l'existant nous a permis de recenser des insuffisances du système existant que nous classons en deux catégories :

Insuffisances d'ordre organisationnel : Elles concernent généralement les postes de travail, classement et archivage de documents.

Insuffisances d'ordre technique : Elles concernent en général les outils et le matériel informatique utilisé par les services du département des moyens généraux.cds

a. Niveau organisationnels

Critique1 : la non existence d'une base de données

Suggestion1 : installer une base de données qui permet l'enregistrement et le suivi des stocks des articles et leurs consultations.

Critique2 : difficulté de se renseigner sur le stock ce qui engendre la non connaissance de la quantité à commander.

Suggestion2 : consultation des différents articles restants faciliter l'opération d'achat

b. Niveau technique

Critique : Non installation d'un logiciel qui effectue les différentes tâches.

Suggestion : Mise en œuvre d'une application informatique.

II.4 Conclusion

A travers ce chapitre, nous avons introduit une brève présentation de l'organisme d'accueil et avons présenté notre champ d'étude puis élaboré une étude de l'existant et sa problématique générale. Enfin nous avons proposé des solutions aux insuffisances constatées ; entre autre la mise en place d'une application pour le suivie de la gestion d'achat des articles ; dans le chapitre suivant nous décrivons la conception de cette dernière.

Chapitre III : Analyse et conception

III.1 Introduction

Dans le but d'une meilleure organisation et avant toute réalisation d'une application informatique, tout processus de développement d'applications ou systèmes informatiques doit suivre une méthode ou démarche bien définie, en mettant en évidence tous les objectifs tracés pour la bonne élaboration du projet souhaité. Dans ce chapitre nous allons entamer le développement de notre application par une analyse qui mettra en évidence les différents acteurs intervenants dans le système ainsi que leurs besoins.

La phase conception, s'appuyant sur les résultats de cette analyse, donnera la modélisation des objectifs à atteindre. Des progrès énormes ont été consentis dans le but d'avoir une meilleure analyse et de rendre la conception plus complète. L'approche objet, s'est avérée un modèle d'analyse et de conception très puissant et se trouve de plus en plus utilisée. Pour cela nous avons adopté la conception avec l'UML (UnifiedModellinglanguage) qui permet de bien représenter la dynamique d'une application par la série des diagrammes qu'il offre.

III.2 Présentation d'UML[2]

III.2.1 Historique

Les langages de modélisation orientés objets ont faits leurs apparition entre le milieu des années soixante-dix et la fin des années quatre-vingt quand les spécialistes méthode, confrontés à un nouveau genre de langages de programmation orientés objet à des applications de plus en plus complexes, ont commencé à expérimenter de nouvelles approches d'analyse et de conception. Entre 1989 et 1994 le nombre de méthodes orientées objet est passé de moins de 10 à plus de 50.

La majorité des utilisateurs ne trouvaient pas de langage de modélisation qui réponde entièrement à leurs besoins ; ils ont alors déclenché la "guerre des méthodes". Basées sur l'expérience acquise, de nouvelles générations de ces méthodes sont apparues. Certaines d'entre elles se sont nettement détachées du lot et plus spécialement la méthode Booch, OOSE (Object-Oriented Software Engineering) de Jacobson et OMT (Object Modeling Technique) de Rumbaugh.

Fusion, Shaler-mellor et Coad-Yourdon en faisaient également partie. Chacune de ces méthodes constituait une méthode complète mais présentait aussi bien des avantages que des inconvénients. Pour résumer, la méthode Booch était particulièrement expressive lors des

phases de conception et de construction des projets, OOSE représentait un excellent outil pour les cas d'utilisation en matière de définition des exigences, d'analyse et de conception générale et OMT-2 était plus particulièrement utile à l'analyse et de conception et aux systèmes d'information contenant une grande quantité de données.

Certaines idées déterminantes ont émergé progressivement au milieu des années quatre-vingt-dix, lorsque GradyBooch (Rational Software Corporation), Ivar Jacobson (Objectory) et James Rumbaugh (General Electric) ont chacun commencé à adopter des idées des deux autres.

Leurs trois méthodes ont été reconnues au niveau mondial comme étant des méthodes orientées objet incontournables.

III.2.2 Diagrammes d'UML

Un diagramme est la représentation graphique d'un ensemble d'éléments qui constituent le système. UML 2 s'articule autour de treize types de diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel. Ces types de diagrammes sont répartis comme suit :

Diagrammes structurels ou diagrammes statiques (*UML Structure*)

- diagramme de classes (*Class diagram*)
- diagramme d'objets (*Object diagram*)
- diagramme de composants (*Component diagram*)
- diagramme de déploiement (*Deployment diagram*)
- diagramme de paquetages (*Package diagram*)
- diagramme de structures composites (*Composite structure diagram*)

Diagrammes comportementaux ou diagrammes dynamiques (*UML Behavior*)

- diagramme de cas d'utilisation (*Use case diagram*)
- diagramme d'activités (*Activity diagram*)
- diagramme d'états-transitions (*State machine diagram*)

Diagrammes d'interaction (*Interaction diagram*)

- diagramme de séquence (*Sequence diagram*)

- diagramme de communication (*Communication diagram*)
- diagramme global d'interaction (*Interaction overview diagram*)
- diagramme de temps (*Timing diagram*)

Pour la modélisation des besoins, nous utilisons les diagrammes suivant :

a. Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation est un graphe d'acteurs, un ensemble de cas d'utilisation englobés par la limite du système, des associations de communication entre les acteurs et les cas d'utilisation, et des généralisations entre cas d'utilisation.

Il est destiné à représenter les besoins des utilisateurs par rapport au système.

b. Diagramme de séquence

Il permet de décrire les scénarios de chaque cas d'utilisation en mettant l'accent sur la chronologie des opérations en interaction avec les objets.

Un diagramme de séquence montre une interaction présentée en séquence dans le temps. En particulier, il montre aussi les objets qui participent à l'interaction par

Leur "ligne de vie" et les messages qu'ils échangent présentés en séquence dans le temps.

Voici quelques notions de base du diagramme :

Scenario : représente une succession particulière d'enchaînements s'exécutent du début à la fin du cas d'utilisation.

Un enchaînement étant l'unité de description de séquences d'actions.

Un ensemble de scénarios pour un cas d'utilisation identifié tous ce qu'il peut arrivé lorsque ce cas d'utilisation est mis en œuvre.

Interaction : un comportement qui comprend un ensemble de messages échangés par un ensemble d'objets dans un certain contexte pour accomplir une certaine tâche.

Message : Un message représente une communication unidirectionnelle entre objets qui transporte de l'information avec l'intention de déclencher une réaction chez le récepteur.

d. Diagramme de classes

Le diagramme de classes est généralement considéré comme le plus important dans un développement orienté objet. Il représente l'architecture conceptuelle du système : il décrit les

classes que le système utilise, ainsi que leurs liens, que ceux-ci représentent un emboîtement conceptuel (héritage) ou une relation organique (agrégation).

III.2.3 Modélisation avec l'UML[3]

UML permet de représenter des modèles, mais il ne définit pas de processus, d'élaboration de modèle. Les acteurs d'UML conseillent tout de même une démarche pour favoriser la réussite d'un projet, cette démarche doit être :

- **Une démarche itératif et incrémentale :**
Pour comprendre et représenter un système complexe, pour analyser par étape, pour favoriser le prototype pour réduire et maîtriser l'inconnu.
- **Une démarche guidée pour les besoins des utilisateurs :**
Tout est basé sur le besoin des utilisateurs du système, le but du développement lui-même est de répondre à leur besoin. Chaque étape sera affinée et validée en fonction des besoins des utilisateurs.
- **Une démarche centrée sur l'architecture logicielle :**
C'est la clé de succès d'un développement, les choix stratégiques définiront la qualité du logiciel.

La figure ci-dessous montre la représentation graphique de la démarche de modélisation que nous avons choisie pour concevoir notre application.

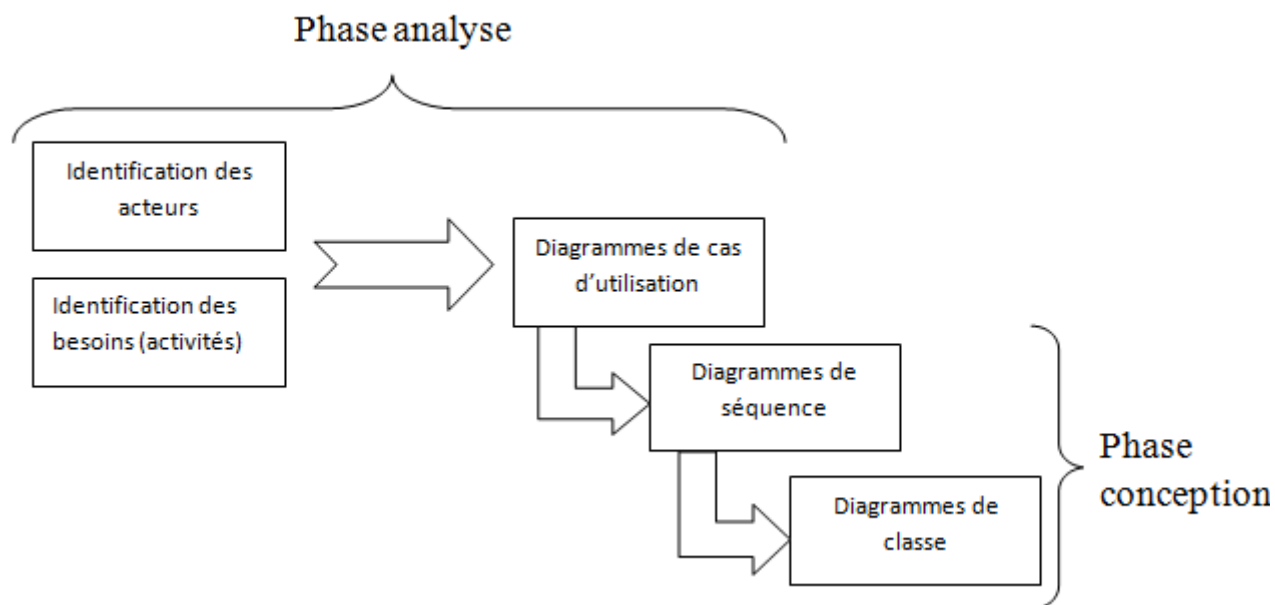


Figure III.1 Représentation graphique de la démarche de modélisation.

III.3 Analyse

Cette partie comprend la spécification des besoins fonctionnels du système, des acteurs et leurs interactions avec le système ainsi que les cas d'utilisation.

III.3.1 Spécification des besoins

Notre projet porte sur la mise en place d'une application client/serveur trois-tiers pour l'automatisation des activités de la gestion des stocks dans le but de maîtriser le suivi des mouvements des documents et dossiers entre les différents services du champ d'étude, l'application doit offrir un environnement interactif et convivial aux différents acteurs de tous les services de la direction facilities, leur permettant de travailler et de communiquer mutuellement, dans l'objectif de la gestion, et ce via un réseau local (intranet).

III.3.2 Identification des acteurs

Un acteur représente une entité qui joue un rôle d'interagir directement avec le système.

Pendant le stage que nous avons effectué au niveau de la compagnie maritime d'affrètement-compagnie maritime général CMA-CGM, nous avons procédé à l'identification des acteurs qui seront les futurs utilisateurs de l'application de la gestion des stocks des articles achetés au sein de cette dernière ces acteurs sont énumérés ci-dessous :

Facility manager (FM) : il gère les comptes utilisateurs, l'ajout des services ou agences.

Acheteur : il se charge de la gestion des achats, la gestion des fournisseurs et l'inventaire.

Chef de département des moyens généraux (CDMGX): s'occupe de la gestion des affectations.

III.3.3 Diagramme de contexte

Le diagramme de contexte est un modèle conceptuel de flux qui offre une vision globale des interactions entre le système et l'environnement extérieur. Il permet aussi de bien délimiter le champ de l'étude.

Dans notre cas le diagramme de contexte est représenté par la figure suivante :

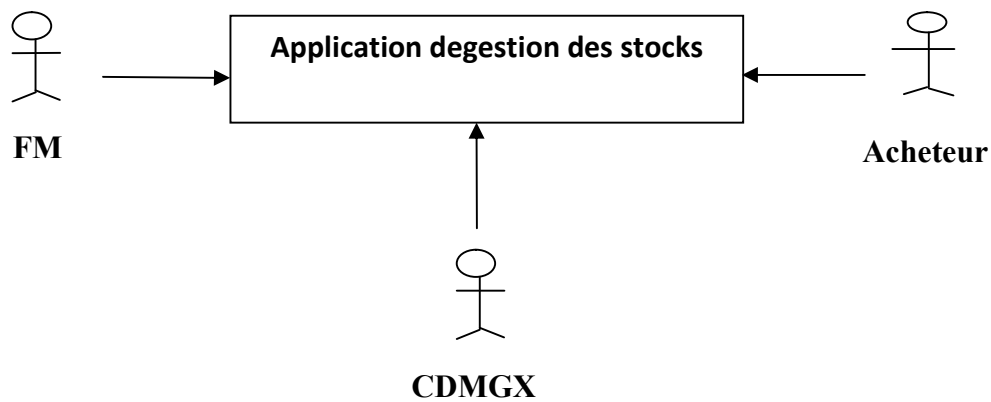


Figure III.2 Diagramme de contexte.

III.3.4 Représentation des cas d'utilisation

III.3.4.1 Définition d'un cas d'utilisation

Un cas d'utilisation précise le comportement d'un système ou d'une partie d'un système et décrit un ensemble de séquences d'actions. Les cas d'utilisation servent à saisir le comportement attendu d'un système en cours de développement, sans avoir à préciser la façon dont ce comportement est réalisé. En UML, tous les comportements sont modélisés sous la forme de cas d'utilisation via un diagramme de cas d'utilisation.

III.3.4.2 Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation permet de représenter graphiquement les cas d'utilisation. Le fait qu'un acteur déclenche un cas d'utilisation est représenté par une flèche entre ces deux derniers.

a. Les cas d'utilisation

Acteur	Cas d'utilisation	Sous cas d'utilisation
FM	S'authentifier	
	Gérer les comptes utilisateurs	Ajouter un utilisateur
		Consulter les utilisateurs
	Ajouter un service	
	Modifier le mot de passe	
	Se déconnecter	
Acheteur	S'authentifier	

	Gestion des achats	Ajouter article
		Réapprovisionner article
	Inventaire	Consulter la liste des articles restants
		Editer la liste des articles
	Gestion des fournisseurs	Ajouter fournisseur
		Consulter les fournisseurs
Se déconnecter		
CDMGX	S'authentifier	
	Gestion des affectations	Affecter un article à un service
		Consulter les articles affectés
	Se déconnecter	

Tableau III.1 les cas d'utilisations.

b. Représentation des diagrammes de cas d'utilisation

Le diagramme de cas d'utilisation de notre application est représenté par la figure ci-dessous

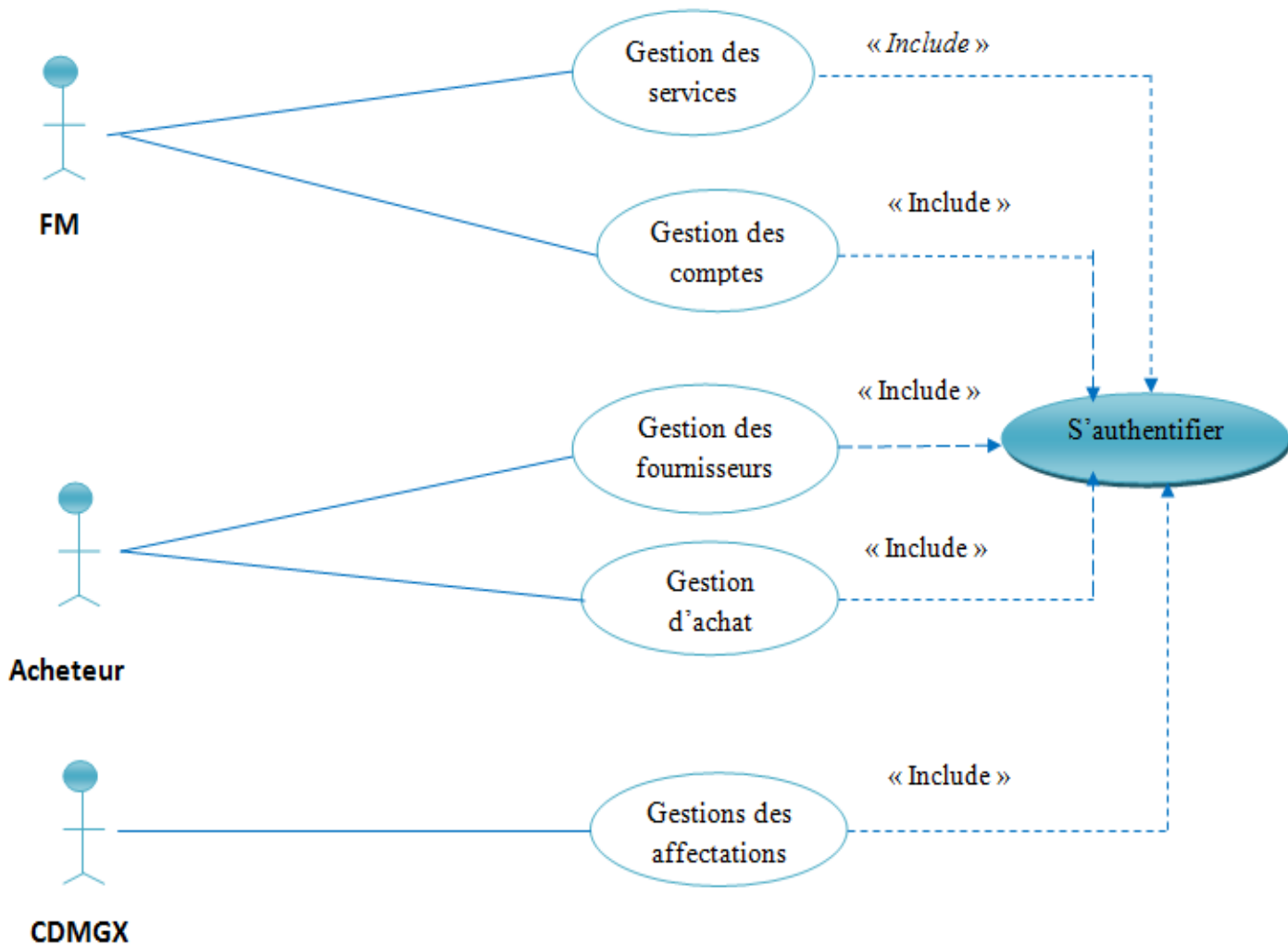


Figure III.3 Diagramme de cas d'utilisation générale.

Pour mieux voir les différentes fonctionnalités de notre application, nous avons élaboré des diagrammes de cas d'utilisation détaillés :

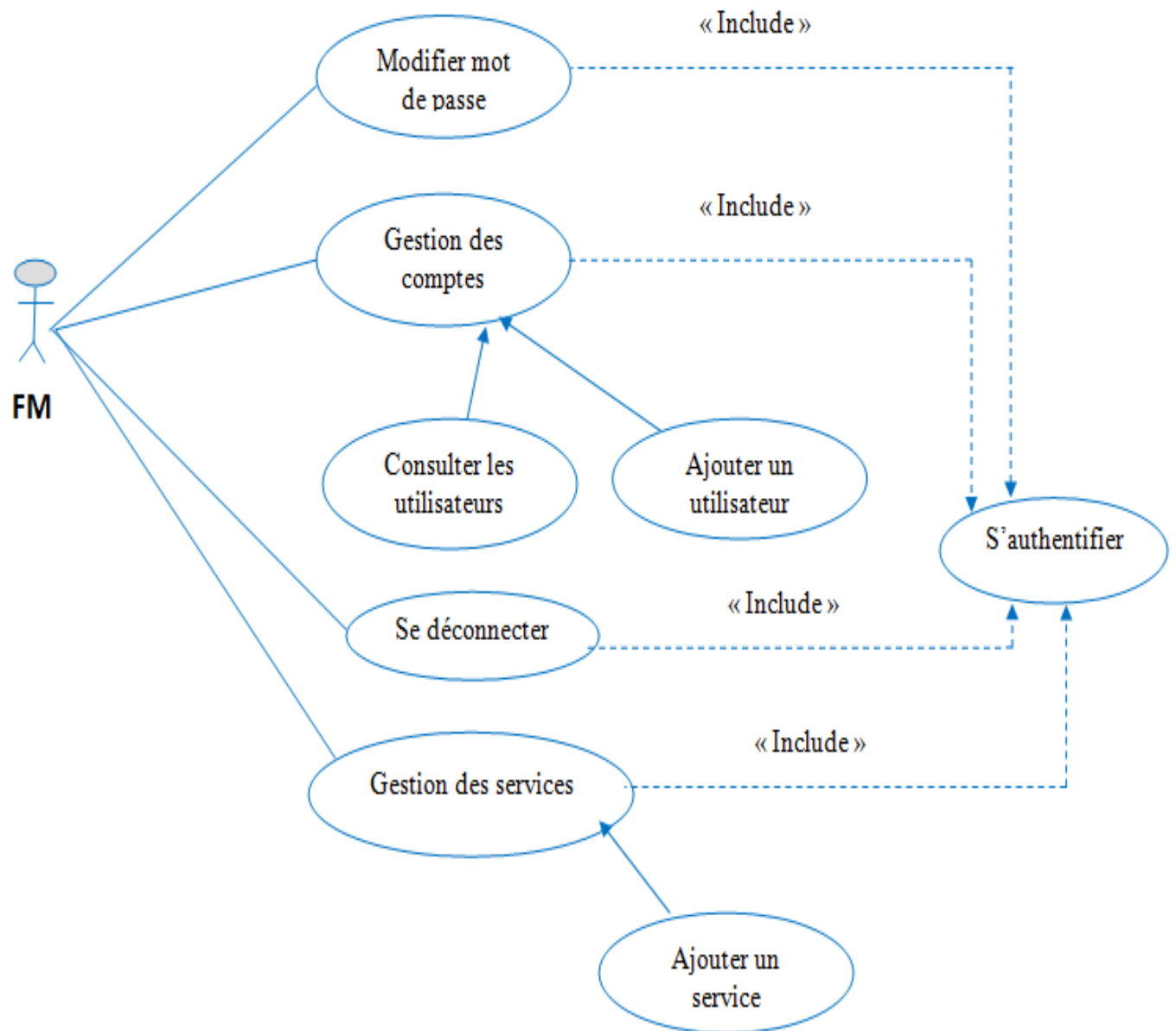


Figure III.4 Diagramme de cas d'utilisation détaillé relatif au « FM ».

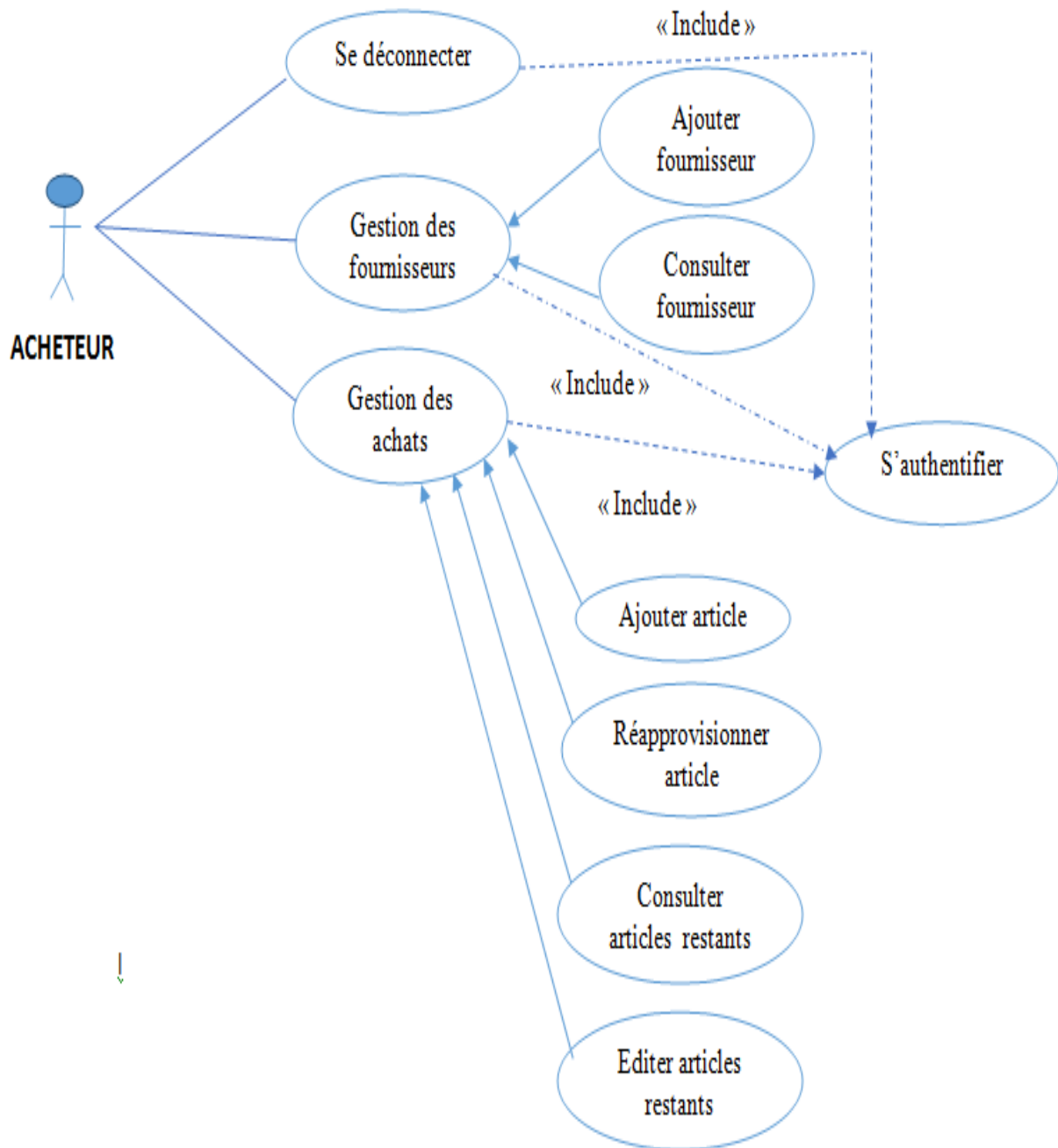


Figure III.5 Diagramme de cas d'utilisation détaillé relatif à « L'ACHETEUR ».

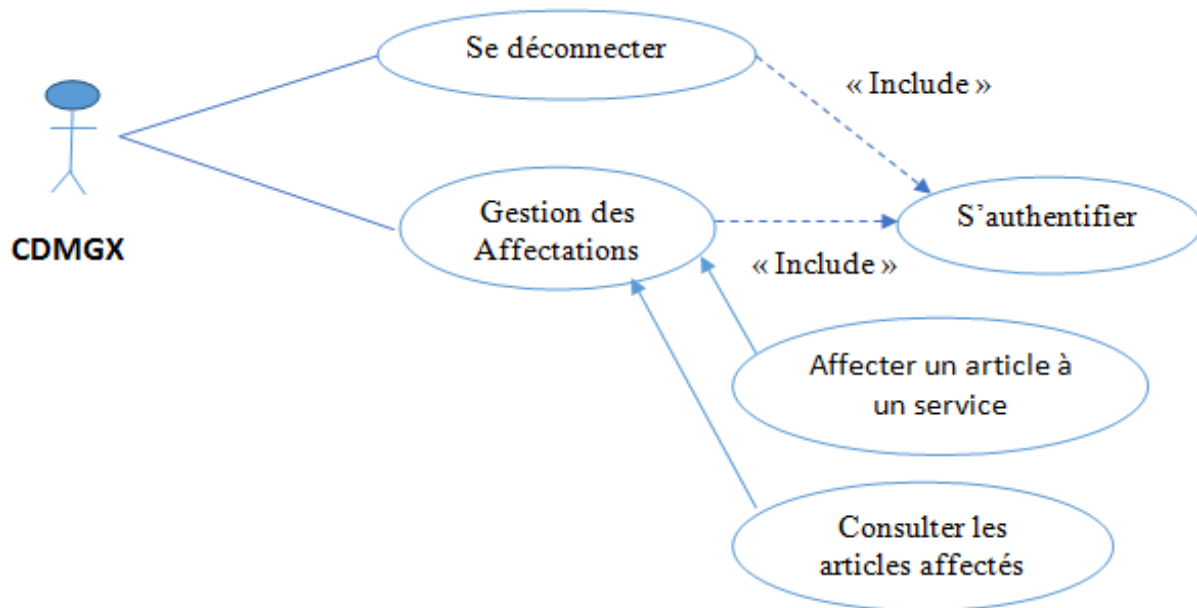


Figure III.6 Diagramme de cas d'utilisation détaillé relatif au « chef de département des moyens généraux ».

III.4 Conception

Le but de la conception est de rendre le modèle réalisable sous forme de logiciel. Après avoir recensé les cas d'utilisation relatifs à chaque acteur de notre système, nous pouvons dès à présent établir les diagrammes de séquence suivie du diagramme de classe.

III.4.1 Diagramme de séquence

Le diagramme de séquence montre des interactions entre objets selon un point de vue temporel. Ce type de diagramme sert à modéliser les aspects dynamiques des systèmes temps réels et des scénarios complexes mettant en œuvre peu d'objets.

Les principales informations contenues dans un diagramme de séquence sont les messages échangés entre les lignes de vie, présentés dans un ordre chronologique.

Dans ce qui suit nous allons établir quelques diagrammes de séquence pour les cas : « s'authentifier », « Ajouter un article », « Modifier mot de passe ».

a. Diagramme de séquence « **Authentification** »

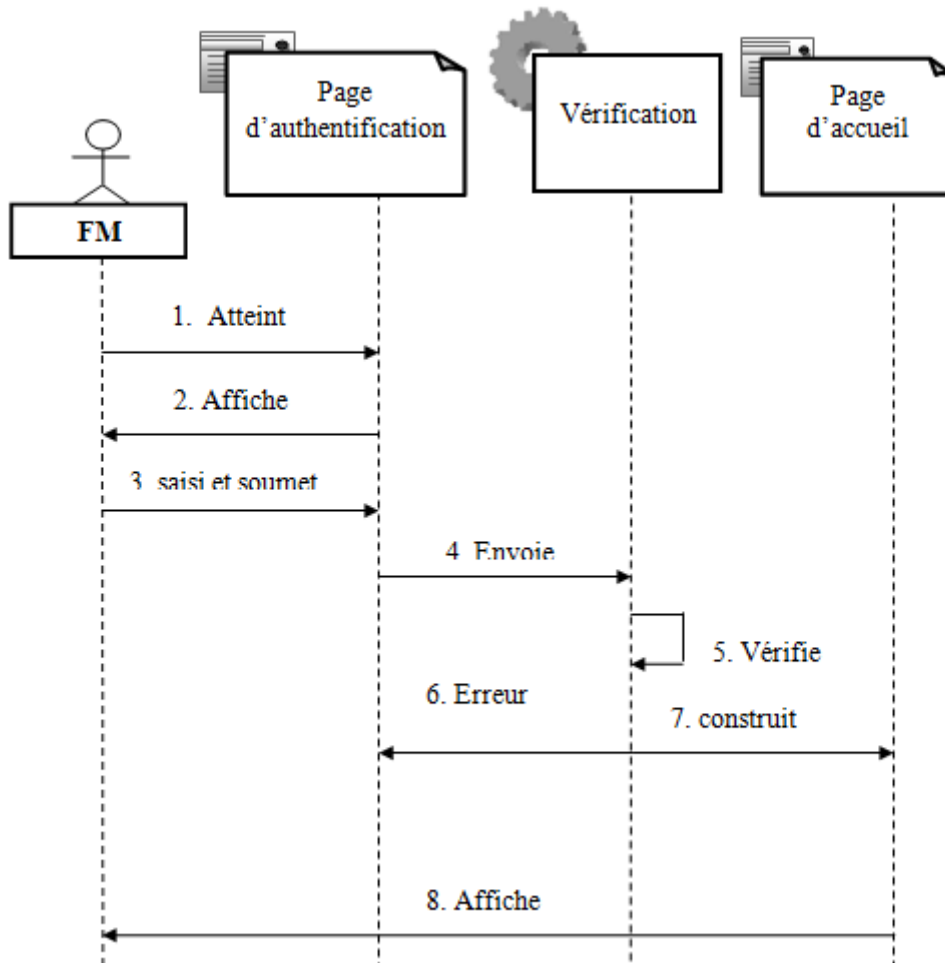


Figure III.7 Diagramme de séquence «Authentification».

Scénario « **Authentification** »

1. Le Facilitymanager atteint la page d'authentification.
2. Le système affiche « la page d'authentification ».
3. Le Facility manager saisit et soumet les données l'identifiant.
4. La page envoie les données au système.
5. Le système vérifie les données.
6. Le système retourne une erreur au cas de l'incompatibilité des données.
7. Le système construit la page d'accueil dans le cas contraire.
8. Le système affiche la page d'accueil.

b. Diagramme de séquence « ajouter article »

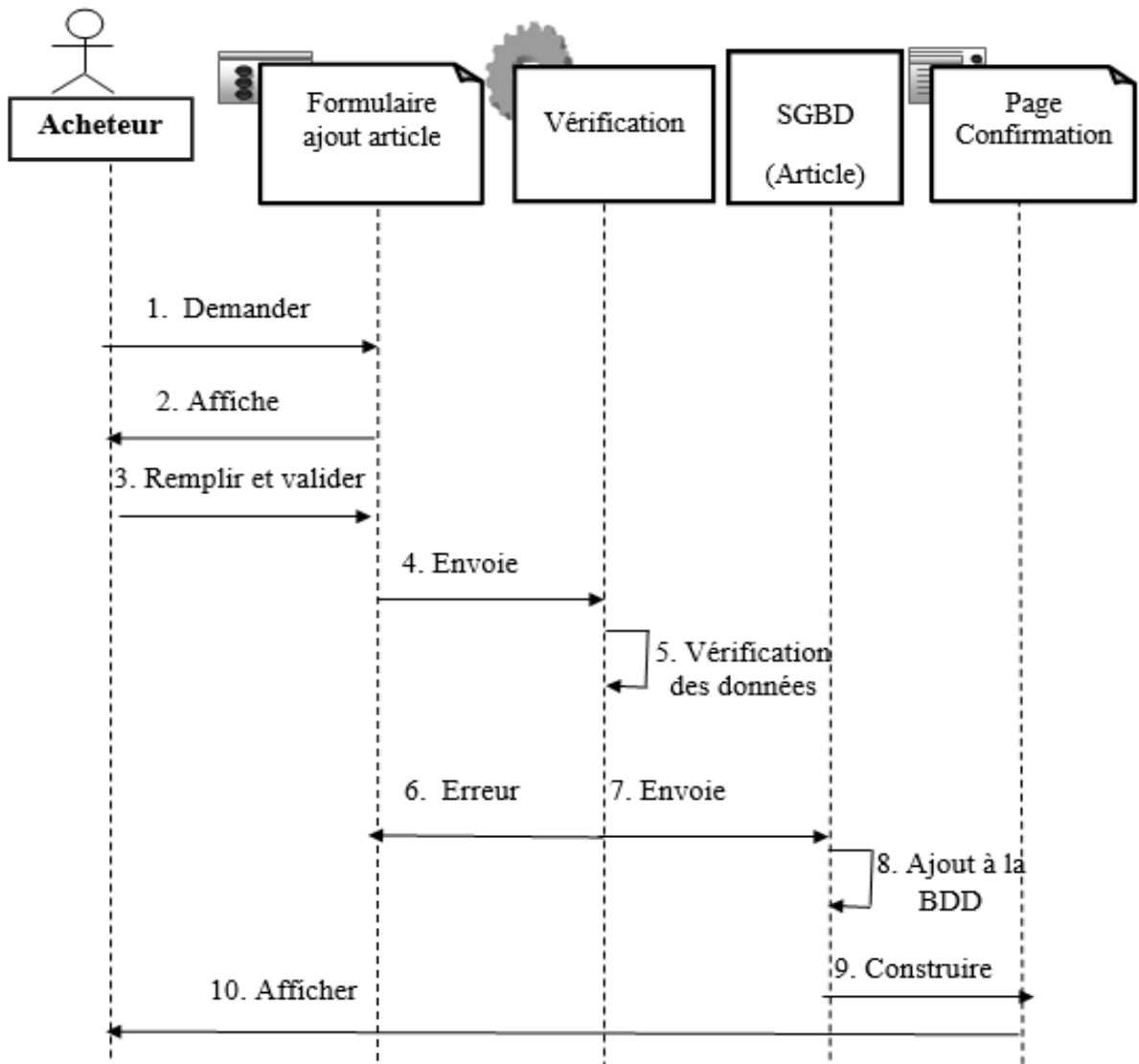


Figure III.8 Diagramme de séquence «Ajouter un article».

Scénario « Ajouter un article »

1. l'acheteur demande le formulaire d'ajout d'un article.
2. affichage du formulaire d'ajout.
3. l'acheteur saisit les données et valide.
4. envoi des données pour vérification.
5. control des données.
6. afficher un message d'erreur.
7. envoi de requête pour stocker les données.

- 8. ajout des données dans la BDD.
- 9. construire un message de confirmation.
- 10. afficher la confirmation d'enregistrement.

c.Diagramme de séquence «**Modifier mot de passe**»

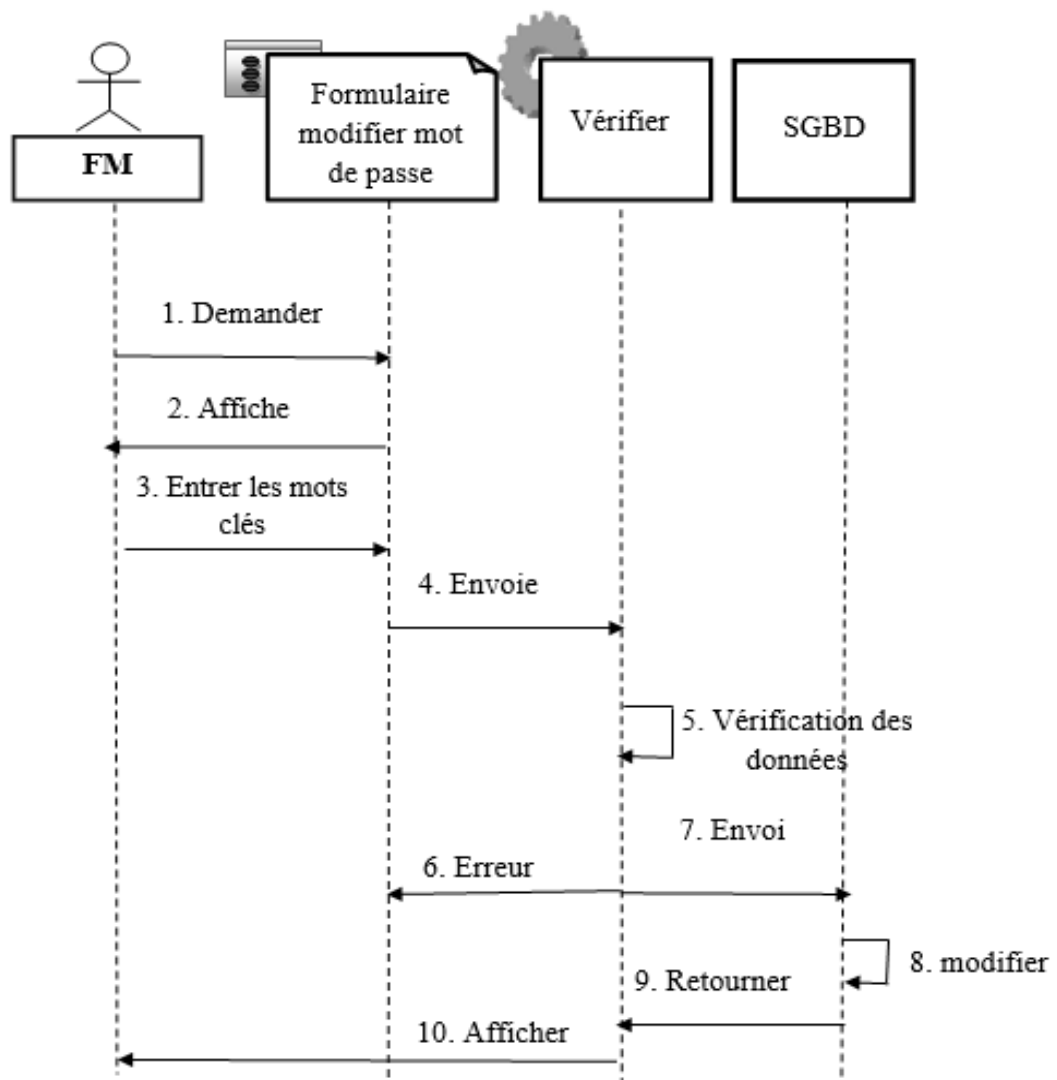


Figure III.9 Diagramme de séquence «*Modifier mot de passe* ».

Scénario «**Modifier mot de passe**»

- 1. le FM demande le formulaire de modification.
- 2. le système affiche le formulaire.

3. le FM saisit les mots clés.
4. l'envoi des données pour vérification.
5. control des données.
6. envoie de requête pour chercher les données.
7. afficher un message d'erreur si données non trouvé.
8. Modifier si données trouvées.
9. la base de données retourne les données au système.
10. le système affiche un message de confirmation.

III.4.2 Les règles de gestion du diagramme de classes

Ces règles vont nous permettre d'établir le diagramme de classes

- Un article peut être affecté à un ou plusieurs services.
- Un fournisseur peut faire une ou plusieurs livraisons.
- Une ou plusieurs livraisons contient un ou plusieurs articles.

III.4.3 Diagramme de classes

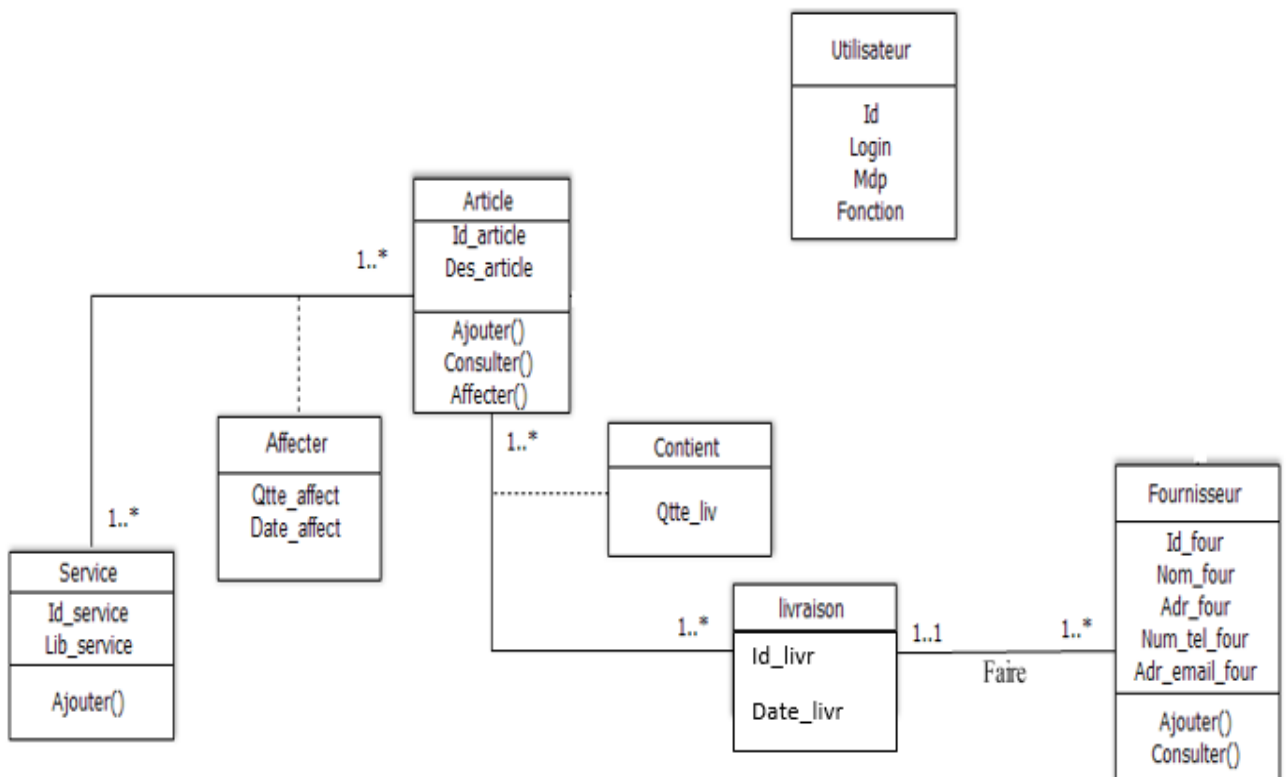


Figure III.9 Diagramme de classe global.

III.4.4 Modèle relationnel

Le modèle relationnel une manière de modéliser les informations contenues dans une base de données qui repose sur des principes mathématiques mis en avant par E.F.Codd. Pour le passage au modèle relationnel, Nous avons suivi les règles suivantes :1- La classe devient une relation.2- L'attribut d'une classe devient une colonne dans une table.3- Association à ordre 0 ou 1 - * : clé étrangère dans la table ayant * comme terminaison d'association.

4- Association à ordre * - * : la classe d'association devient une relation contenant deux clés étrangères.

Article (id_article, Des_article)

Fournisseur (Id_four, Nom_four, Adr_four, Num_tel_four, Num_fax_four, Adr_email_four, id_article).

Livraison (id_livr, date_livr, Id_four*).

Affecter (Qtte_affect, Date_affect, id_article*, Id_serv*).

Service (Id_serv, Lib_serv).

Contient (id_article*, Id_livr*, qtte_livr).

Utilisateur(Id, Login, Mdp, Fonction)

: Clé primaire

*** : Clé étrangère**

III.5 Conclusion

Dans ce chapitre nous avons présenté l'analyse et la conception de notre application en utilisant le langage UML pour le Web. Nous avons commencé par la définition des besoins

qui ont été traduits par la suite en formalismes graphiques à l'aide des diagrammes offerts par le langage UML s'appuyant sur l'analyse.

La partie conception a mis en évidence les différents composants de l'application et cela dans le but de la construction de notre base de données.

Dans le chapitre qui suit nous allons le consacrer à présenter les différents outils et technologies de développement de notre application.

Chapitre IV : Réalisation

IV.1 Introduction

Dans ce chapitre, nous allons présenter les outils de développements utilisés pour mener à terme la réalisation de notre application ainsi que, quelques interfaces de notre application.

IV.2 Les outils de développement

Nous allons présenter dans cette partie les différents outils de développement dont nous nous sommes servis pour réaliser notre application.

IV.2.1 NetBeans

NetBeans est un environnement de développement intégré (EDI), placé en *open source* par Sun en juin 2000 sous licence CDDL et GPLv2 (*Common Development and Distribution License*). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML.

La version utilisée est 7.4

IV.2.2 Serveur Web Apache tomcat

L'ancêtre d'apache est le serveur libre développé par le (National Center for Supercomputing Applications) de l'université de l'Illinois. L'évolution de ce serveur s'est arrêtée lorsque le responsable a quitté le NCSA en 1994, les utilisateurs ont continué à corriger les bugs et à créer des extensions qu'ils distribuaient sous forme de " patches " (bouts de programmes ajoutés par les utilisateurs des NCSA pour étendre les fonctionnalités d'Apache) d'où le nom "a patchee Server" la version 1.0 d'Apache a été disponible le 1 décembre 1995.

Nous choisissons et présentons ce serveur par les avantages qu'il offre et que nous allons détailler ci-après :

- C'est un serveur gratuit (peut être téléchargé à partir du site du groupe Apache à l'adresse '<http://www.apache.org>').
- Un niveau élevé de performances pour des besoins matériels modeste.
- Extensible, modulaire et configurable.
- Robuste.

- Très portable contrairement à IIS (Internet Information Services) de Microsoft qui tourne seulement sous Windows, le serveur Web apache dispose d'une version pour chaque plateforme (Linux, Windows, ...).

IV.2.3 Dreamweaver

Est un éditeur de site web WYSIWYG pour Microsoft Windows, et Mac OS X créé en 1997, commercialisé par Macromedia puis Adobe Systems sous licence utilisateur final.

Dreamweaver fut l'un des premiers éditeurs HTML de type « tel affichage, tel résultat », mais également l'un des premiers à intégrer un gestionnaire de site. Ces innovations l'imposèrent rapidement comme l'un des principaux éditeurs de site web, aussi bien utilisable par le néophyte que par le professionnel.

IV.2.4 Langages web utilisés

- **Le langage HTML** : pour la création des pages statiques de l'application qui constitueront l'interface utilisateur.
- **JSP (JAVA SERVER PAGE)** : pour la partie dynamique du site. Il constitue le langage pilier de l'application, c'est-à-dire, que c'est sur lui que reposent le dynamisme de l'application. c'est la raison pour laquelle nous allons mieux le détailler dans ce qui suit :

IV.2.5 JSP

Les Java Server Pages, JSP pour les intimes, représentent le V dans le modèle M-V-C c'est-à-dire "la vue". C'est un standard permettant de développer des applications Web interactives, donc, dont le contenu est dynamique. C'est-à-dire qu'une page web JSP (repérable par l'extension *.jsp*) aura un contenu pouvant être différent selon certains paramètres (des informations stockées dans une base de données, les préférences de l'utilisateur, etc.).

Son fonctionnement :

Une page utilisant les Java Server Pages est exécutée au moment de la requête par un moteur de JSP, fonctionnant généralement avec un serveur Web ou un serveur applicatif. Lorsqu'un utilisateur appelle une page JSP, le serveur Web appelle le moteur de JSP qui crée un code source Java à partir du script JSP, compile la classe afin de fournir un fichier compilé (d'extension **.class**), c'est-à-dire qu'il constitue en fait une servlet à partir du script JSP..

Fragment de code en JSP

```
<%  
java.util.Date dateToday = new java.util.Date();  
%>  
  
<html>  
<body>  
<p>La date est <%= dateToday %></p>  
</body>  
</html>
```

IV.2.6 Feuille de style en cascade

CSS est un langage informatique qui sert à décrire la présentation des documents HTML et XML. Son principal objectif est de séparer la structure d'un document de ses styles de présentation.

IV.2.7 JavaScript

Le JavaScript est un langage de script incorporé dans un document HTML. Historiquement il s'agit même du premier langage de script pour le Web. Ce langage de programmation permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client, c'est-à-dire au niveau du navigateur et non du serveur web.

IV.2.8 Le Java Development Kit (JDK)

La version du *JDK* utilisé pour le système est *7.0.450.18*

IV.2.9 Système de gestion de bases de données

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server.

Son nom vient du prénom de la fille du Co créateur Michael Widenius, MySQL fait allusion au *StructuredQueryLanguage*, le langage de requête utilisé.

MySQL AB a été acheté le 16 janvier 2008 par Sun Microsystems pour un milliard de dollars américains. En 2009, Sun Microsystems a été acquis par Oracle Corporation, mettant entre les mains d'une même société les deux produits concurrents que sont Oracle Data base et MySQL. Ce rachat a été autorisé par la Commission européenne le 21 janvier 2010.

Depuis mai 2009, son créateur Michael Widenius a créé Maria DB pour continuer son développement en tant que projet Open Source.

IV.2.10 Langages d'interrogation de bases de données utilisés

SQL Sigle de StructuredQueryLanguage, en français langage de requête structurée est un langage informatique normalisé servant à exploiter des bases de données relationnelles. La partie langage de manipulation des données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles.

IV.3 Structure de la base de données

IV.3.1 Article

Nom du champ	Type de donnée	Description	Clé
id_article	Varchar	Identifiant article	Primaire
Des_article	Varchar	Désignation article	

Tableau IV.1 Table article.

IV.3.2 Fournisseur

Nom du champ	Type de donnée	Description	Clé
Id_four	Varchar	Identifiant fournisseur	Primaire
Nom_four	Varchar	Nom du fournisseur	
Adr_four	Varchar	Adresse fournisseur	
Num_tel_four	Int	Numéro de téléphone du fournisseur	
Num_fax_four	Int	Numéro de fax du fournisseur	
Adr_email_four	Varchar	Adresse email du fournisseur	

Tableau IV.2 Table fournisseur.

Chapitre IV Réalisation

IV.3.3 Service

Nom du champ	Type de donnée	Description	Clé
Id_serv	Varchar	Identifiant service	Primaire
Lib_serv	Varchar	Libellé service	

Tableau IV.3 Table service.

IV.3.4 Affecter

Nom du champ	Type de donnée	Description	Clé
id_article	Varchar	Identifiant article	Etrangère
Id_serv	Varchar	Identifiant service	Etrangère
Qtte_affec	Varchar	Quantité affectée	

Tableau IV.4 Table affecter.

IV.3.5 La table livraison

Nom du champ	Type de donnée	Description	Clé
id_livr	Varchar	Identifiant livraison	Primaire
Date_livr	Date	Date de livraison	

Tableau IV.5 Table livrer.

IV.3.6 la table contenir

Nom du champ	Type de donnée	Description	Clé
Id_article	Varchar	Identifiant article	Etrangère
Id_livr	Varchar	Identifiant livraison	Etrangère
qtte_livr	Varchar	Quantité livré	

Tableau IV.6 Table contenir.

IV.3.6 la table utilisateur

Nom du champ	Type de donnée	Description	Clé
id_uti	Varchar	Identifiant utilisateur	Primaire
login	Varchar	login	
mdp	Varchar	Mot de passe	
fct	Varchar	Fonction utilisateur	

Tableau IV.7 Table utilisateur.

IV.3 Architecture Modèle-Vue-Contrôleur

Le M-V-C plus connu sous Modèle-Vue-Contrôleur est une architecture très puissante qui aide à bien structurer les applications graphiques. Ce modèle de conception permet de séparer les données (le modèle), l'interface homme-machine (la vue) et la logique de contrôle (le contrôleur).

Ce paradigme impose donc une séparation en 3 couches :

Le modèle : Il représente les données de l'application. Il définit aussi l'interaction avec la base de données et les traitements de ses données.

La vue : Elle représente l'interface utilisateur, ce avec quoi il interagit. Elle n'effectue aucun traitement, elle se contente simplement d'afficher les données que lui fournit le modèle. Il peut tout à fait y avoir plusieurs vues qui présentent les données d'un même modèle.

Le contrôleur : Il gère l'interface entre le modèle et le client. Il va interpréter la requête de ce dernier pour lui envoyer la vue correspondante. Il effectue la synchronisation entre le modèle et les vues.

Le modèle MVC est illustré par la figure suivante :

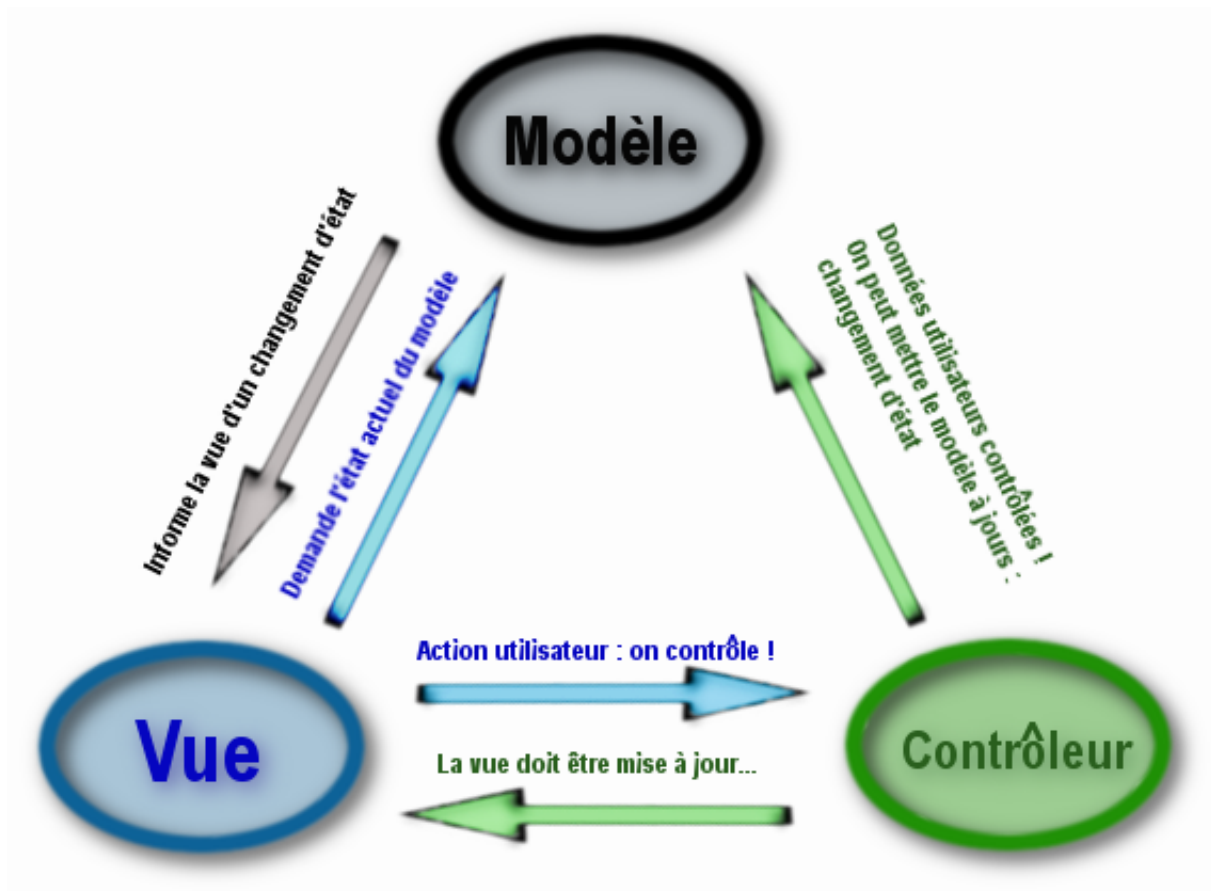


Figure IV.I Architecture du modèle-vue-contrôleur « MVC ».

IV.4 Quelques interfaces de notre application

Dans cette section nous allons présenter quelques interfaces de notre application

IV.4.1. Interface NetBeans

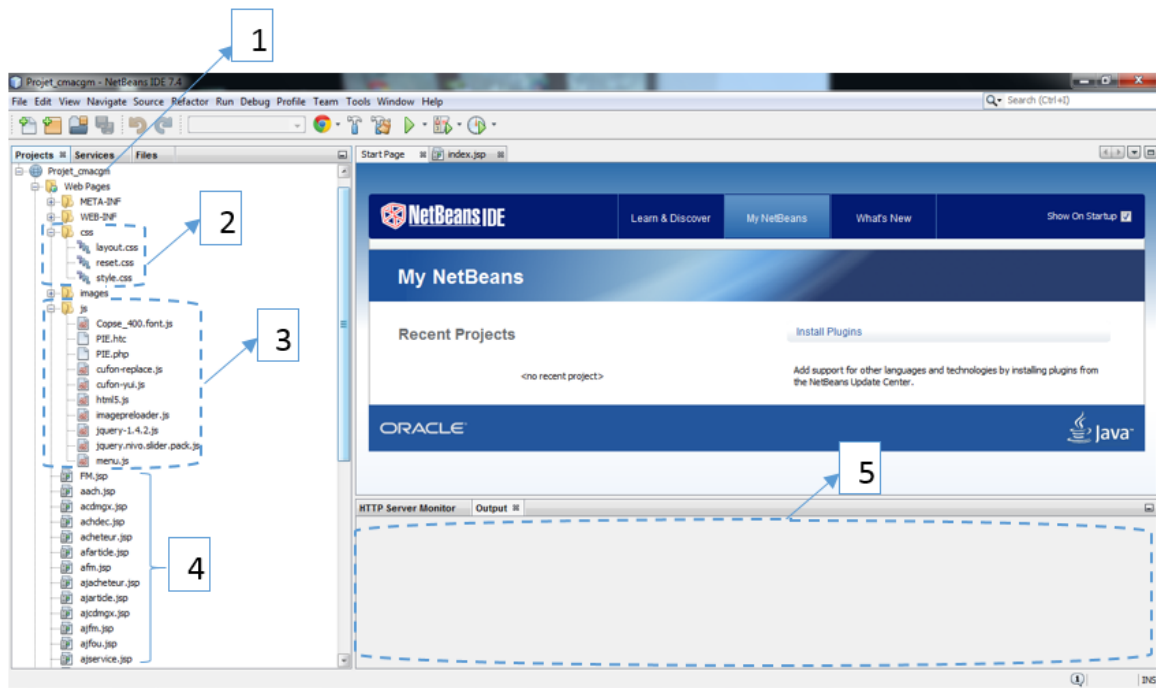


Figure IV.2 Interface NetBeans.

1 : notre projet intitulé « projet_cmacgm ».

2 : le dossier CSS et ces différentes feuilles de styles en cascades composant ce dossier.

3 : le dossier java script qui contient les fichiers JavaScript utilisés dans notre application qui se distinguent des autres fichiers grâce à l'extension *.js*.

4 : le contenu de notre projet.

5 : la zone où apparaissent les erreurs éventuelles de nos programmes.

IV.4.2 Interface d'accueil

Chaque acteur de notre système doit choisir la rubrique qui lui correspond pour atteindre sa page d'authentification.



Figure IV.3 Interface d'accueil.

IV.4.3 Interface d'authentification

Avant d'entrer dans l'environnement de travail, un acteur doit s'authentifier en saisissant un login et un mot de passe.

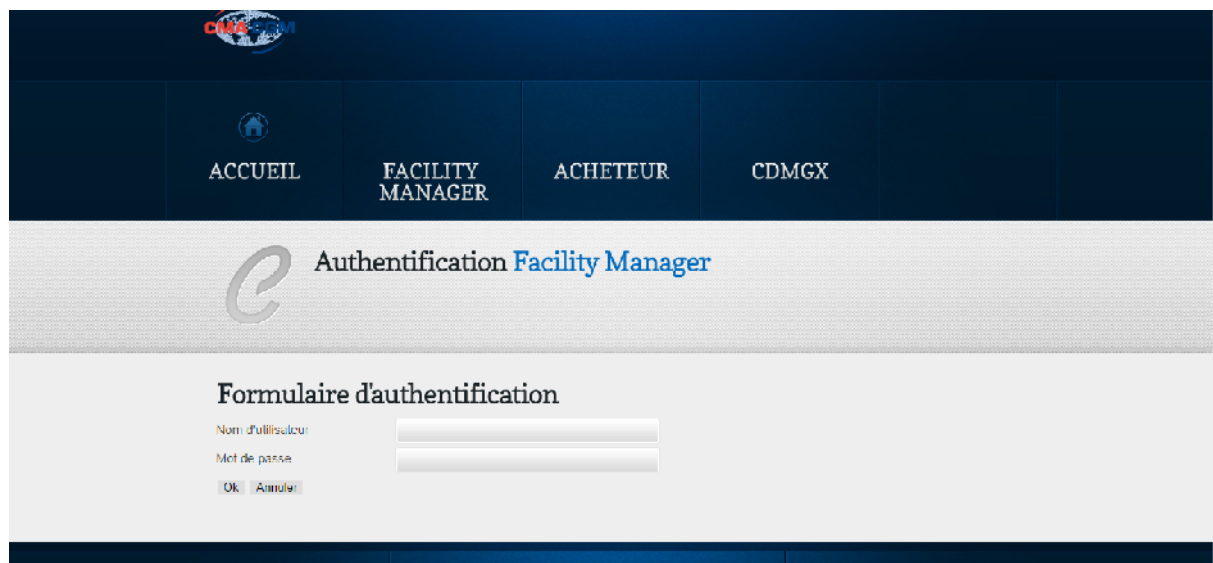
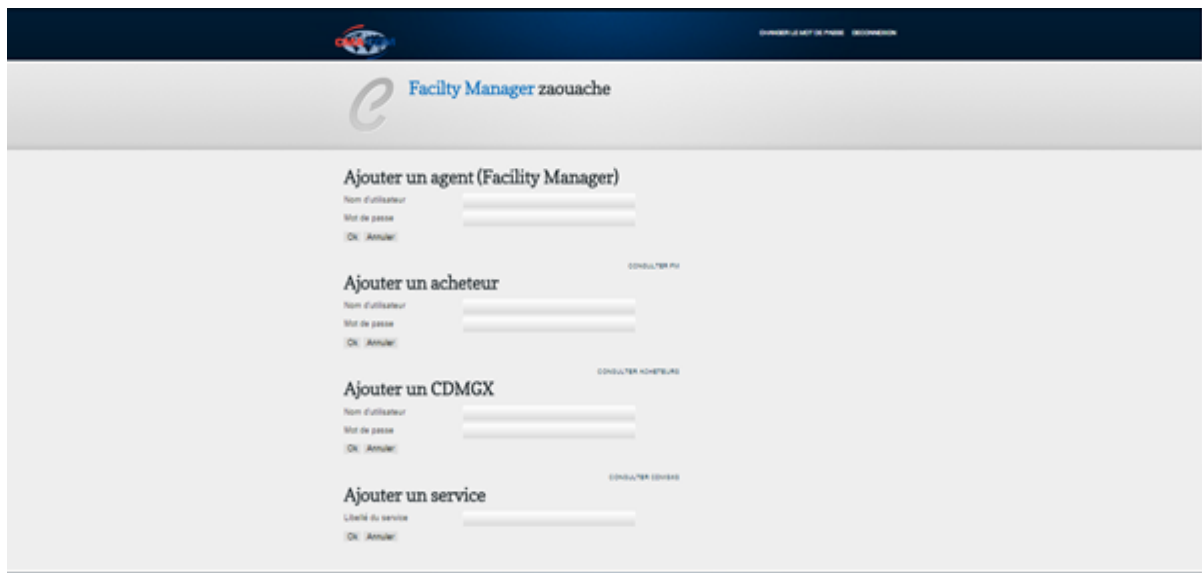


Figure IV.4. Interface d'authentification.

IV.4.4 Interface de gestion des comptes

Cette fenêtre permet à l'acteur Facility manager d'ajouter des utilisateurs dans la base de données en saisissant les informations relatives à celui-ci comme il peut consulter les différents acteurs dans le formulaire présenté ci-dessous :



The screenshot displays the 'Facility Manager zaouache' web interface. It features a dark blue header with a logo and navigation links. Below the header, the main content area is divided into four sections, each with a title and a form:

- Ajouter un agent (Facility Manager)**: Includes fields for 'Nom d'utilisateur' and 'Mot de passe', and buttons for 'OK', 'Annuler', and 'CONSULTER FMI'.
- Ajouter un acheteur**: Includes fields for 'Nom d'utilisateur' and 'Mot de passe', and buttons for 'OK', 'Annuler', and 'CONSULTER ACHETEUR'.
- Ajouter un CDMGX**: Includes fields for 'Nom d'utilisateur' and 'Mot de passe', and buttons for 'OK', 'Annuler', and 'CONSULTER CDMGX'.
- Ajouter un service**: Includes a field for 'Liberté du service' and buttons for 'OK', 'Annuler', and 'CONSULTER SERVICE'.

Figure IV.5 Interface de gestion des comptes.

IV.4.5 Interface de page de confirmation

Cette fenêtre confirme l'ajout d'un utilisateur dans la base de données.



The screenshot shows a confirmation page titled 'formulaire enregistré'. The page has a dark blue header with a logo. The main content area is light gray and contains a central message box with the following text:

Votre formulaire a été bien enregistré!
Veuillez repeter l'operation en cliquant sur ce lien.

Figure IV.6 Interface de page de confirmation de l'ajout.

IV.4.6 Interface de gestion des affectations

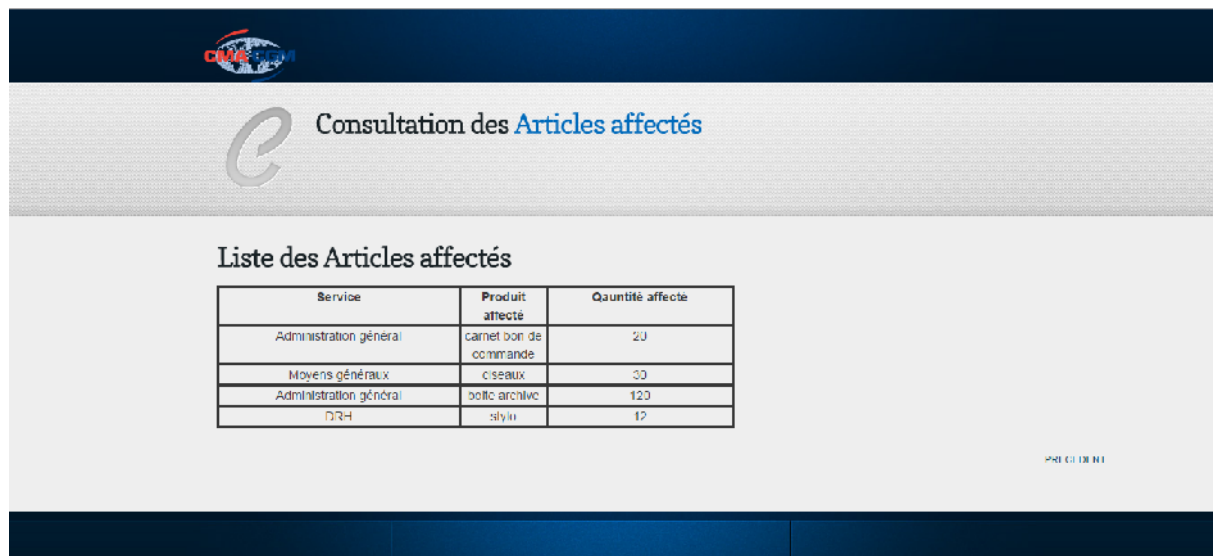
Cette fenêtre permet à l'acteur CDMGX d'affecter des articles à un service en cas de besoin de ce dernier.



Figure IV.7 Interface de gestion des affectations.

IV.4.7 Fenêtre de consultation des articles affectés

Cette fenêtre permet d'afficher la liste des articles affectés aux différents services.



Service	Produit affecté	Quantité affectée
Administration générale	carton bon de commande	20
Moyens généraux	ciseaux	00
Administration générale	boite archive	120
DRH	stylo	12

Figure IV.8. Fenêtre de consultation des articles affectés.

IV.4.8 Interface de gestion d'achat

Cette fenêtre permet à l'acheteur d'ajouter un fournisseur, un article dans la base de données en saisissant les informations relatives à ces derniers dans le formulaire présenté ci-dessous :

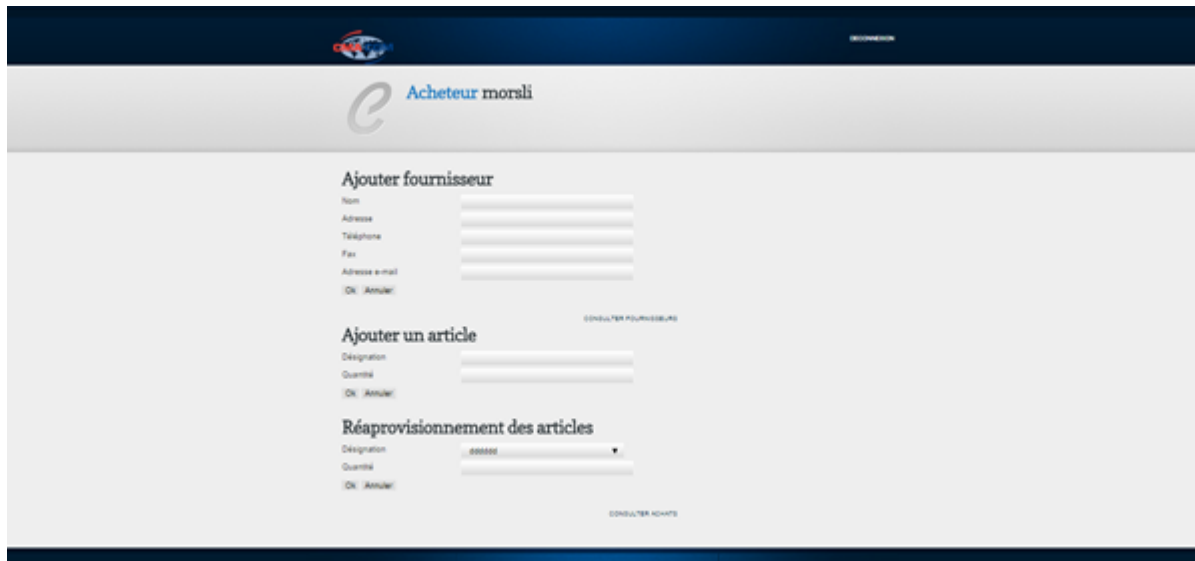


Figure IV.9 Interface de gestion des achats.

IV.4.9 Fenêtre de consultation des articles

Cette fenêtre permet d'afficher la liste des articles et d'éditer ce document.

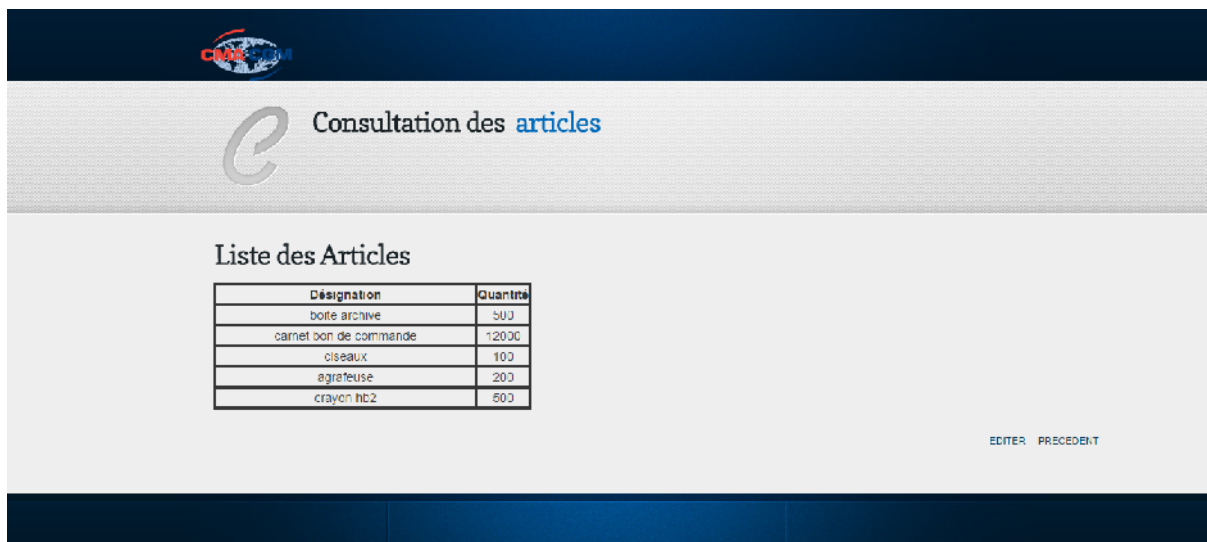


Figure IV.10 Fenêtre de consultation des articles.

IV.4.10 Interface d'édition de la liste des articles

Cette fenêtre permet à l'acheteur d'imprimer la liste des articles en cas de besoin

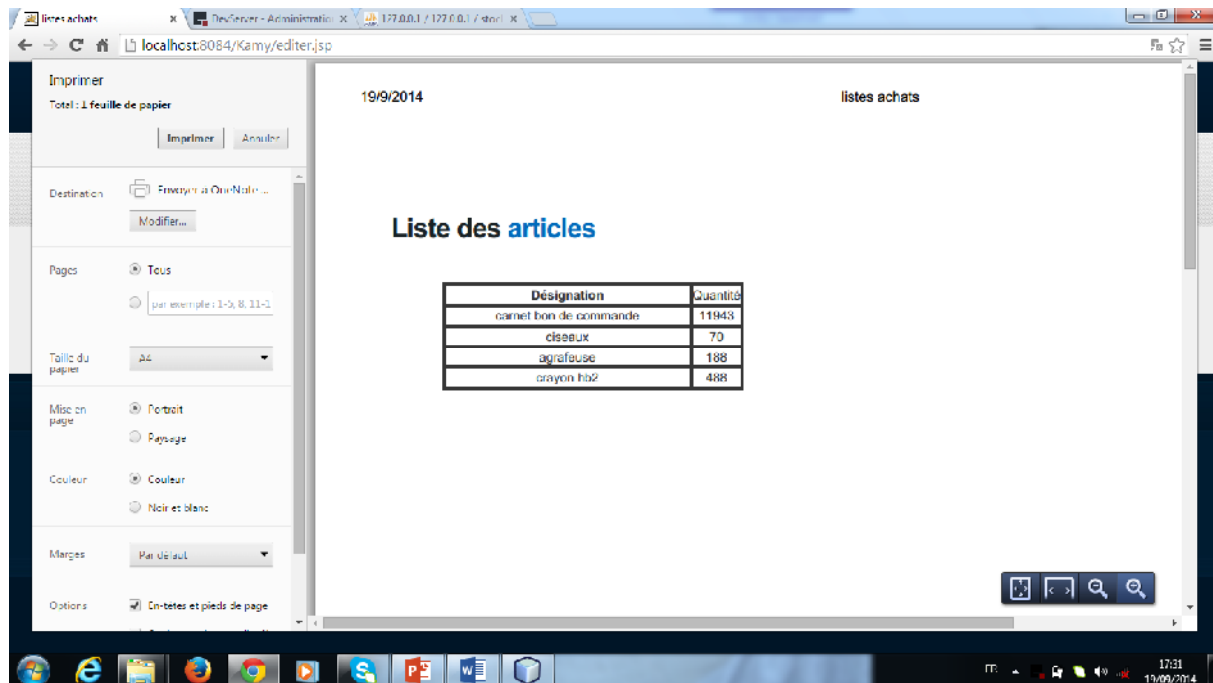


Figure IV.11 Fenêtre d'édition de la liste des articles.

IV.5 Conclusion

Dans ce chapitre nous avons présenté les différents outils qui ont servis à la réalisation de ce projet l'architecture adoptée et pour finir, nous avons présenté quelques interfaces de l'application.

Conclusion générale

Aujourd'hui, la disponibilité de systèmes de gestion de base de données fiables permet aux organisations de toutes tailles de gérer des données efficacement, de déployer des applications utilisant ces données et de les stocker. Les bases de données sont actuellement au cœur du système d'information des entreprises. La quasi-totalité des applications requièrent une base de données pour le stockage de l'information comme simple source d'informations.

La réalisation de notre application client/serveur sur la gestion automatisée du stock a été un projet très bénéfique pour nous car elle nous a permis d'approfondir nos connaissances apprises tant théoriques que pratiques durant notre parcours universitaire à travers la conception et la modélisation des systèmes d'information et le développement des logiciels d'application.

Durant notre stage, nous avons pu combler certaines zones d'ombres qui régnaient dans notre compréhension. Ces nouvelles connaissances acquises nous permettent d'entrevoir de nouvelles perspectives. Ce travail que nous avons réalisé n'est que le début de notre carrière en développement logiciel que nous espérons très longue.

Nous avons réalisé ce projet dans le but de faciliter et d'améliorer la gestion du stock et l'inventaire des articles au sein du groupe **CMA CGM**.

Bibliographie

[1] : Réseaux Andrew Tanenbaum et David Wetherall, Editeur Pearson 26/08/2011

[2] : UML2 Compact de HeideBalzert, Editeur Eyrolles 16/02/2006

[3] : UML2 Pratique de la modélisation de Benoît Charroux, AomarOsmani, Yann Thierry-Mieg Editeur Pearson 26/06/2010

[4] : Le guide de l'utilisateur UML GradyBooch, James Rumbaugh, Editeur Eyrolles 24/04/2003

[5] : François-Mariecolonna « l'architecture client-serveur » Octobre 2002.

[6] : TCP/IP architecture, protocoles et applications Douglas Comer, Editeur Pearson 01/01/2009.

Webographie

<http://www.commentcamarche.net>

<http://www.developpez.com>

<http://www.uml.free.fr>

<http://www.memoireonline.com>

<http://www.futura-sciences.com/fr>

<http://java.developpez.com/>

<http://delphi.developpez.com/cours/>