

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

UNIVERSITE MOULOU D MAMMERRI DE TIZI-OUZOU

• X •ΘΛ•ΣX [://:V •X[•Λ[•O
FACULTE DU GENIE ELECTRIQUE
ET INFORMATIQUE

Mémoire de fin d'études

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

« Option : Conduite de Projet Informatique »

Thème

*Conception et réalisation d'une application
client/serveur pour Le suivi des
installations propane.*

Cas : NAFTAL Thala-Athmane

Dirigé par :

Mr: S.KHEMLICHE

Réalisé et présenté par :

Mr: HADJI Mahdi

Mr: HAMMOUM Nadir

2014/2015

❧ Remerciements ❧

De prime abord, nous tenons à remercier le Bon Dieu tout puissant et miséricordieux de nous avoir donné patience, courage et volonté pour réussir notre travail.

En second lieu, nous remercions vivement notre enseignant et promoteur Mr S.KHEMLICHE d'avoir accepté de nous encadrer tout au long de notre projet, ainsi que pour ses conseils précieux et ses critiques constructives, ses explications et suggestions pertinentes. Aussi nous tenons à lui reconnaître le temps précieux qu'il nous a consacré.

Nous remercions chaleureusement les membres du jury pour l'honneur qu'ils nous font en acceptant de juger ce projet de fin d'études.

Nos remerciements s'étendent également à tout les responsables et personnel de l'Entreprise NAFTAL pour avoir accepté de nous accueillir au sein de l'entreprise dans le cadre du projet de fin d'étude. Plus particulièrement Mr AMMARI pour son accueil et ses conseils précieux

Nous témoignons de notre gratitude et sympathie à toute personne ayant contribué de près ou de loin au bon accomplissement de ce travail.

Nous exprimons, en fin, notre infinie gratitude à nos chers parents et familles en reconnaissance de leurs sacrifices, aides, soutien et encouragement.

Dédicaces

Je dédie ce modeste travail à :

*A mes chers **parents** qui ont toujours été là pour moi, qui m'ont donné un magnifique modèle de labeur et de persévérance, pour leur attention, sacrifice et soutien tout au long de mes études.*

*A ma petite nièce adorable **INES***

*A mes adorables sœurs **Nabila** et **Djamila**.*

*A mes chers frères **Sofiane** et **faycen** et leurs épouses.*

*A mon petit frère **Amar**.*

*A toute ma **famille** & mes **proches**.*

*A mon binôme & ami **Nadir** et toute sa **famille**.*

*A mes **ami(e)s** pour leurs aides, leurs temps, leurs assistance et soutiens.*

*A toute la **promotion** informatique **LMD** 2014-2015*

Mehdi.

Dédicaces

Je dédie ce modeste travail à :

Mes parents pour leur aide précieuse et sacrifices

Mes adorables sœurs Cylia, Lyssia et Ouissam

Toute ma famille & mes proches.

Mon binôme Mehdi et toute sa famille

Mes ami(e) pour leur aide, leur temps, leur assistance et soutien

*Tout se qui ont contribué de près ou de loin a la réalisation de ce
travail.*

Merci infiniment.

Nadir.

Sommaire

Sommaire

Introduction générale.....	1
----------------------------	---

Chapitre I : *Etude de l'existant*

I.1 Introduction et problématique.....	02
I.2 Présentation de l'entreprise	02
I.2.1 Historique de NAFTAL	02
I.2.2 Organisation de NAFTAL.....	03
I.2.3 Missions de NAFTAL.....	04
I.2.4 Organigramme de l'entreprise.....	05
I.2.5. Présentation de la branche GPL	06
I.2.6 Présentation du district GPL Tizi-Ouzou.....	06
I.2.7 Organigramme de la branche GPL Tizi-Ouzou	07
I.3 Présentation du champ d'étude	08
I.3.1 Organigramme du champ d'étude	08
I.3.2 Situation informatique du champ d'étude	09
I.4 Etude de l'existant.....	10
I.4.1 Introduction	11
I.4.2 Etude des postes	14
I.4.3 Etude des documents	22
I.4.4 le model organisationnel de traitement	22
I.4.5 Diagramme des flux	32
I.4.6 Critique et suggestions	34
I.4.7 Proposition de solution	34
I.5 Conclusion	35

Chapitre II : Généralités sur les réseaux et l'architecture client /serveur.

II.1. Introduction	36
II.2. Réseaux informatiques	36
II.1.Définition	36
II.2.Classification des réseaux	36
II.2.1. En fonction de la distance	36
II.2.2. En fonction de la topologie	37
II.3. Architecture des réseaux	38
II.3.1. Architecture d'égal à égal (peer to peer)	38
II.3.2. L'architecture client/serveur	39
II.3.2.1.Définition	39
II.3.2.2.Notions de base	40
II.3.2.3. Fonctionnement d'un système client/serveur	41
II.3.2.4 Les différentes architectures client/serveur	42
II.4. Les protocoles	44
II.4.1. Le modèle OSI	44
II.4.1.1 La couche physique	45
II.4.1.2 La couche liaison de données	46
II.4.1.3 La couche réseau	46
II.4.1.4 La couche transport	46
II.4.1.5 La couche session.....	46
II.4.1.6 La couche présentation.....	46
II.4.1.7 La couche application	46
II.4.2. Le modèle TCP/IP	46
II.4.2.1 La couche accès réseau	47
II.4.2.2 La couche Internet (Inter réseau)	47
II.4.2.3 La couche transport	48
II.4.2.4 La couche Application	48
II.5.Conclusion	48

Chapitre III : *Analyse & Conception*

III.1. Introduction	49
III.2. Analyse	49
III.2.1. Identification des besoins	49
III.2.2. Identification des acteurs	50
III.2.3. Identification des cas d'utilisation	50
III.3. La conception	51
III.3.1. Le niveau applicatif	52
III.3.1.1. Le diagramme des cas d'utilisation.....	52
III.3.1.2. Description des cas d'utilisation avec des scénarios.....	55
III.3.1.3. Les diagrammes de séquences et d'activités.....	58
III.3.1.4. Les diagrammes de classes	65
III.3.2. Le niveau donné	69
III.3.2.1. Le diagramme de classes général.....	70
III.3.2.2. Le modèle logique.....	71
III.4. Conclusion	72

Chapitre IV : *Réalisation et Mise en œuvre*

IV.1. Introduction	73
IV.2. Choix d'outils technologiques	73
IV.2.1. Environnement de développement	73
IV.2.1.1. Eclipse.....	73
IV.2.1.2. Adobe dreamwaver CS5	74
IV.2.2. Outils de développements	75
IV.2.2.1. JAVA	75
IV.2.2.2. HTML.....	79
IV.2.2.3. JavaScript.....	79
IV.2.3. Les serveurs déployés	80
IV.2.3.1. Le serveur Apache	80
IV.2.3.2. Le conteneur de servlets Tomcat	81
IV.2.3.3. Le serveur de données	82
IV.3. Architecture du système	87
IV.4. Présentation de quelques interfaces de notre plate-forme	87

V.5 Conclusion	95
Conclusion générale	96
Bibliographie	97

Liste des figures

Liste des figures

Figure I.1 : Organigramme de NAFTAL.....	05
Figure I.2 : Organigramme du champ d'étude	07
Figure I.3 : Organigramme du département distribution	08
Figure I.4 : Organigramme du département technique & maintenance.....	09
Figure I.5 : Diagramme des flux d'information	33
Figure II.1 : Classification des réseaux informatique selon leur taille	36
Figure II.2 : System peer to peer	39
Figure II.3 : System client/serveur	41
Figure II.4 : L'architecture à deux niveaux	42
Figure II.5 : L'architecture à trois niveaux.....	43
Figure II.6 : L'architecture à multi niveaux	44
Figure II.7 : Modèle OSI	45
Figure II.8 : Modèle TCP/IP	47
Tableau III.1 : identification des cas d'utilisation	50
Figure III.1 : La démarche adoptée pour la modélisation.....	52
Figure III.2 : Le diagramme des cas d'utilisation	53
Figure III.3 : Description du cas d'utilisation « S'authentifier ».....	55
Figure III.4 : Description du cas d'utilisation « Ajouter une demande d'installation »....	56
Figure III.5 : Description du cas d'utilisation « Modifier un ordre de travail ».....	57
Figure III.6 : Description du cas d'utilisation « Supprimer un client».....	57
Figure III.7 : Diagramme de séquence du cas d'utilisation « « S'authentifier ».....	59
Figure III.8 : Diagramme d'activités du cas d'utilisation « S'authentifier ».....	60
Figure III.9 : Diagramme de séquence du cas d'utilisation « Ajouter un client».....	61
Figure III.10 : Diagramme d'activités du cas d'utilisation « Ajout un client».....	62
Figure III.11 : Diagramme de séquence du cas d'utilisation « Modifier le mot de passe»	63
Figure III.12 : Diagramme d'activités du cas d'utilisation « Modifier le mot de passe ».....	64
Figure III.13 : Diagramme de classes général du cas d'utilisation « S'authentifier ».....	65
Figure III.14 : Diagramme de classes général du cas d'utilisation « Ajouter un client».....	66
Figure III.15 : Diagramme d'activités du cas d'utilisation« Modifier le mot de passe»	66
Figure III.16 : Diagramme de classes détaillé du cas d'utilisation « S'authentifier ».....	67
Figure III.17 : Diagramme de classes détaillé du cas d'utilisation « Ajouter un client ».....	68
Figure III.18 : Diagramme de classes détaillé du cas d'utilisation « Modifier le mot de passe »	69
Figure III.19 : Diagramme de classes général	70
Figure IV.1 : Outils retenus.....	73
Figure IV.2 : Interface de d'Eclipse	74
Figure IV.3 : Interface d'Adobe Dreamwaver CS5	75
Figure IV.4 : Interface d'Apache Tomcat.....	81

Figure IV.5 : Interface de PhpMyAdmin	82
Figure IV.6 : Interface d'authentification	88
Figure IV.7 : Interface de modification/suppression d'un client	88
Figure IV.8 : Interface d'ajout d'une d'installation	89
Figure IV.9 : Interface d'ajout d'un nouveau client.....	89
Figure IV.10 : Interface d'ajout d'un ordre de travail.....	90
Figure IV.11 : Interface d'ajout d'une facture	90
Figure IV.12 : Interface d'impression d'une facture.....	91
Figure IV.13 : Consultation des demandes d'installation propane 35 kg	91
Figure IV.14 : Interface d'ajout d'une citerne	92
Figure IV.15 : Interface de consultation des ordres de travaux	92
Figure IV.16 : Interface d'ajout d'un procès verbal.....	93
Figure IV.17 : Interface de modification du mot de passe	93

Introduction

Générale

Actuellement, le monde connaît une avancée technologique considérable dans tous les secteurs et cela grâce à l'informatique qui est une science qui étudie les techniques du traitement automatique de l'information. Elle joue un rôle important dans le développement de l'entreprise.

Avant l'invention de l'ordinateur, on enregistrait toutes les informations manuellement sur des supports en papier ce qui engendrait beaucoup de problèmes tel que la perte de temps considérable dans la recherche de ces informations ou la dégradation et la perte de ces dernières. ..Etc.

Le travail qui nous a été confié par le District GPL de Thala-Athmane, consiste à développer une application Client/serveur 3tiers pour la gestion des installations propane pour les deux services Vente et Maintenance et Installation Fixe(MIF) de l'entreprise NAFTAL.

Cette application sera déployée sous un réseau LAN et permettra au chargé d'étude et au chef de section MIF une meilleure gestion des différentes tâches et des difficultés auxquelles ils sont confrontés quotidiennement.

Et pour mener à bien notre travail, nous avons structuré le présent mémoire comme suit :

- *Le premier chapitre s'intitule « Généralités sur les réseaux et l'architecture client /serveur» comprend des généralités sur les réseaux et la présentation de l'architecture client/serveur.*
- *Le deuxième chapitre s'intitule « Etude de l'existant», consacré à l'organisme d'accueil et à la présentation du champ d'étude.*
- *Le troisième chapitre sous le nom « analyse et conception », qui est consacré à l'analyse et la conception de notre application.*
- *Le quatrième et le dernier chapitre « Réalisation et mise en œuvre », comporte quand à lui la présentation de l'environnement dont lequel notre application a été réalisé, les outils utilisés et quelques interfaces de notre application.*

CHAPITRE

I **Etude de l'existant**

I.1. introduction et problématique :

Le nombre croissant de demande d'installation propane au sein du district NAFTAL GPL de Thala-Athmane engendre plusieurs problèmes aux deux services : Vente et Maintenance & Installation Fixe (MIF) qui s'en chargent de la gestion et de la réalisation de ses installations.

Parmi ses problèmes on peut citer :

- Risque de perte de documents
- Manque de sécurité pour ce qui concerne les documents
- Mauvais archivage des informations
- Lenteur dans la recherche d'information
- Difficultés de suivi de ces dernières une fois les avoir reçus.
- Perte et entassement des piles de documents à cause de la non-informatisation de ceux-là.
- Difficulté de gestion des documents établit (demande d'installation, ordre de travaux, procès verbal et facture).
- Retard dans la réalisation des tâches telle que l'élaboration d'un ordre de travail.

D'où la nécessité d'introduire l'outil informatique afin d'assurer une gestion meilleur et de minimiser le risque d'erreurs. Le but de notre étude est de la conception et la réalisation d'une application client/server pour la gestion des installations propane.

Dans ce mémoire, cette partie sera consacrée à la présentation de l'entreprise « *NAFTAL* ».

I.2. Présentation de l'entreprise :

Cette présentation nous permet de bien connaître l'entreprise en expliquant ses missions et moyens depuis sa naissance.

I.2.1. Historique de NAFTAL :

NAFTAL, société nationale de commercialisation et de distribution des produits pétroliers, filiale de SONATRACH a été créée en 1987. Sa mission essentielle consiste à distribuer et à commercialiser des produits pétroliers sur le marché national. Elle intervient en qualité d'intermédiaire entre les fournisseurs nationaux et étrangers (raffineurs, manufacturiers et autres producteurs) et les utilisateurs de produits pétroliers implantés essentiellement en

Algérie bien que depuis l'année 2002, elle cherche à s'internationaliser en essayant de pénétrer les marchés de certains pays limitrophes.

La distribution consiste à s'approvisionner, stocker, vendre et acheminer le produit vers le client en vue de son utilisation. C'est ainsi que NAFTAL assume deux grandes fonctions :

- La fonction logistique qui comprend la circulation de tous les flux physiques du producteur à l'utilisateur : transport, livraison, stockage et manutention.
- La fonction commerciale qui englobe la gestion du réseau, la vente, les actions promotionnelles et la gestion de la force de vente.

Créée par décret n° 80-101 du 6 avril 1980, l'entreprise ERDP/NAFTAL a été constituée par le transfert des structures, moyens et biens, activités et personnel détenus et gérés auparavant par SONATRACH. L'ERDP/NAFTAL est entrée en activité le 1^{er} janvier 1982. Sa mission consistait à prendre en charge le raffinage et la distribution des produits pétroliers en Algérie. En 1987, elle a connu une autre restructuration instituée par le décret n° 87-189 du 27 août 1987 et qui s'est concrétisée par la création de deux entreprises :

- NAFTEC chargée du raffinage du pétrole ;
- NAFTAL chargée de la distribution et de la commercialisation des produits pétroliers sur le marché national.

I.2.2. Organisation de NAFTAL :

L'organisation de cette entreprise est définie par la décision n°17/CAB du 30 mars 1986 portant organisation transitoire de l'entreprise Nationale NAFTAL modifiée et complétée par décision n°90*/CAB du 05 novembre 1986.

Dans le cadre du plan national de développement économique et social, l'entreprise NAFTAL est chargée de la commercialisation et la distribution des produits pétroliers y compris ceux destinés à l'aviation et à la marine, le GPL, les combustibles, les solvants, les aromatiques, paraffines, bitumes et pneumatiques.

L'identité de l'entreprise NAFTAL est comme suit :

Nom : Entreprise Nationale de Commercialisation Distribution des produits Pétroliers.

Nature : SPA.

Siège : rue des Dunes BP 73, CHERAGA, ALGER

I.2.3. Missions de NAFTAL :

NAFTAL.spa, a pour mission principale, la distribution et la commercialisation des produits pétroliers sur le marché national. Elle intervient dans les domaines :

- de l'enfûtage GPL
- de la formulation de bitumes
- de la distribution, stockage et commercialisation des carburants, GPL, lubrifiants, bitumes, pneumatiques, GPL/carburant, produits spéciaux.
- du transport des produits pétroliers

Pour assurer la disponibilité des produits sur tout le territoire, Naftal met à contribution plusieurs modes de transport :

- Le rail pour le ravitaillement des dépôts à partir des entrepôts.
- Le cabotage et les pipes, pour l'approvisionnement des entrepôts à partir des raffineries.
- La route pour livraison des clients et le ravitaillement des dépôts non desservis par le rail.

I.2.4. Organigramme général de l'entreprise

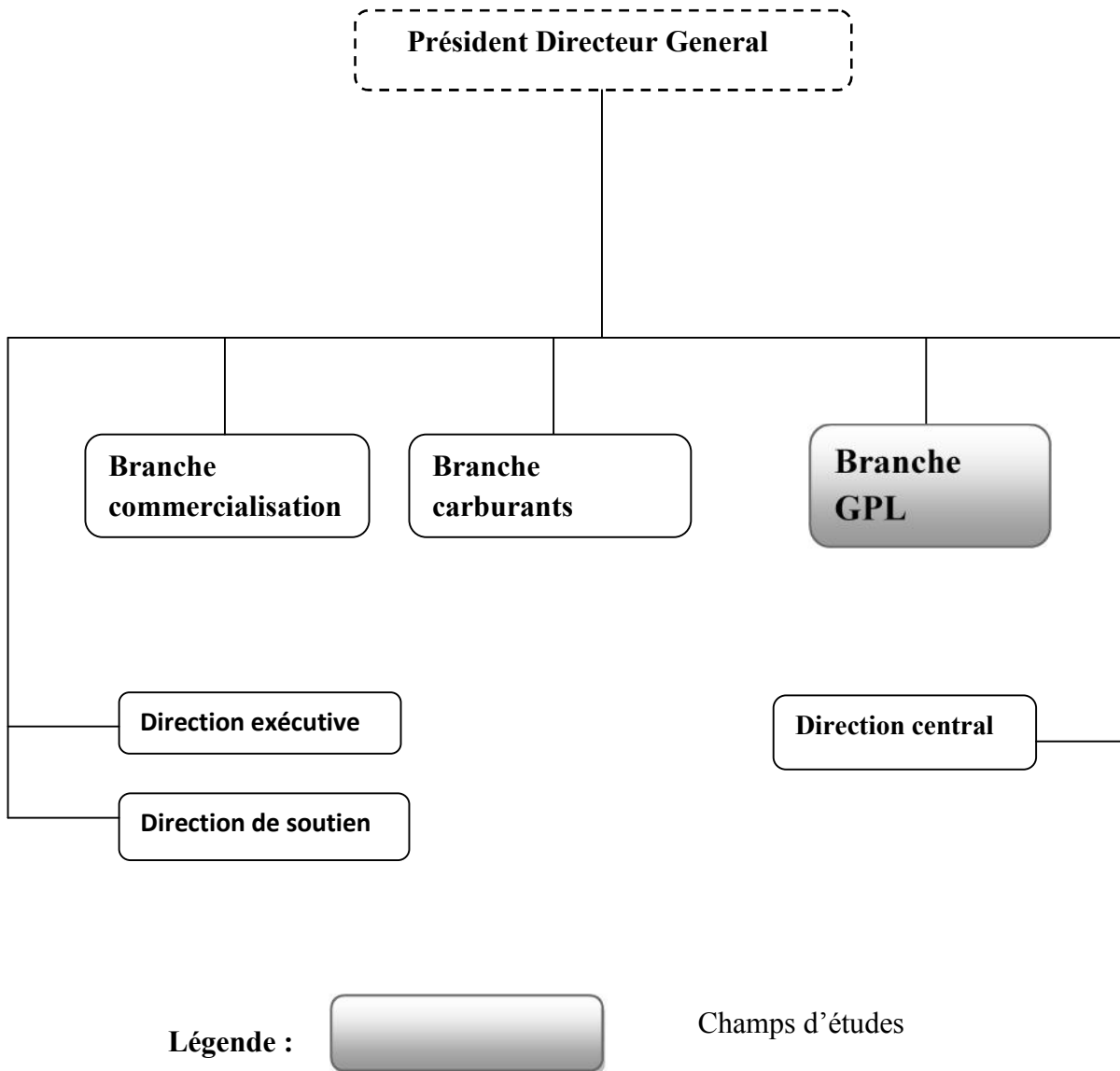


Figure I.1 : Organigramme de l'entreprise NAFTAL

I.2.5 Présentation de la branche GPL

La branche GPL est une structure interne a NAFTAL, chargée totalement de l'activité GPL.

Les principales taches de la branche GPL sont : la commercialisation, le stockage, le transport et le conditionnement des produits GPL.

Le Gaz de Pétrole Liquéfié (GPL), est un mélange d'hydrocarbures légers stocké à l'état liquide et issu du raffinage du pétrole (40% des ressources mondiales) et plus généralement du traitement du gaz naturel (60% des ressources mondiales).

Dans la gamme des GPL, NAFTAL commercialise les trois produits suivants :

- Le Butane
- Le propane
- Le GPL Carburant sous le nom commercial « SIRGHAZ ».

Le butane et le propane sont commercialisés sous deux formes (vrac ou conditionnée).

Le SIRGHAZ est commercialisé sous la forme vrac.

Missions de la branche G.P.L

Elle a pour mission de :

- Veiller au bon fonctionnement des structures conformément aux normes et à la politique de la société.
- Veiller à l'application correcte des règles, procédure et directives de la division
- Assurer des missions d'inspection et de contrôle a tous les niveaux tant fonctionnels qu'opérationnels

I.2.6 Présentation du district GPL de Tizi-Ouzou :

C'est l'unité qui se charge essentiellement de la distribution des produits GPL dans la wilaya de Tizi-Ouzou. Son siège est situé à la localité de Tala-Athmane à 15 km sud de la ville de Tizi-Ouzou.

L'unité a pour objectif d'assurer régulièrement la distribution, et la commercialisation de ses produit par l'intérimaire de ses infrastructures et ses installations diverses.

- Siege de l'unité Tala-Athmane.
- Centre enfuteur 151 Oued-Aissi.
- Mini centre enfuteur 153 Freha.
- Trois dépôts relais : 152 à Tizirt, 154 aux Ouadhias, 155 à Ain El Hammam.

I.2.7. Organigramme de la branche G.P.L « TiziOuzou » :

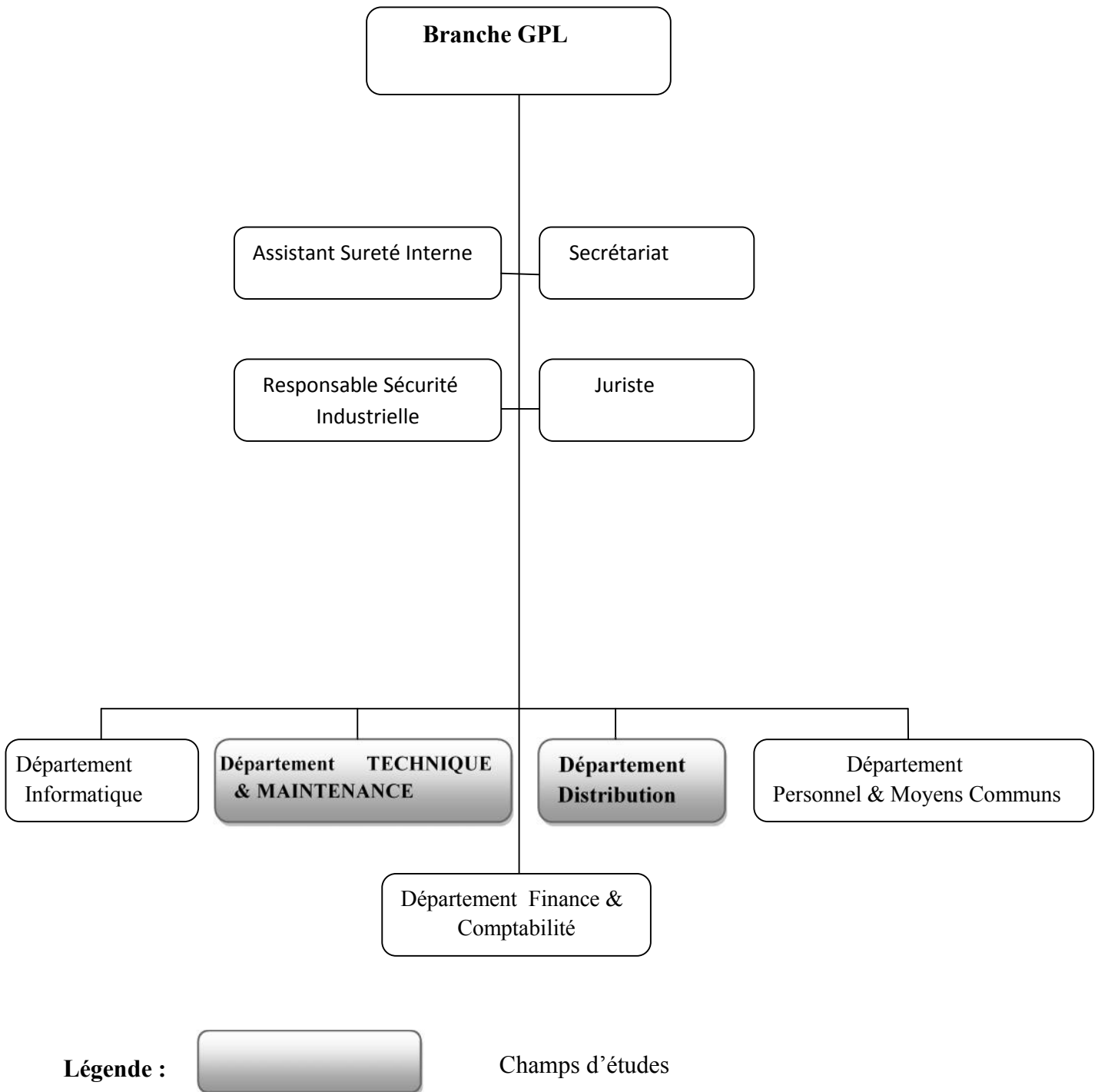


Figure I.2 : Organigramme du champ d'étude

I.3. Présentation du champ d'étude :

Notre champ d'étude comporte les deux services Vente et Maintenance et Installation Fixe (MIF) en particulier les deux postes : le chargé d'études et le chef de section réalisation et installation propane qui se situent dans la branche GPL du district de Thala-Athmane.

I.3.1. Organigramme du champ d'étude

- **Département distribution**

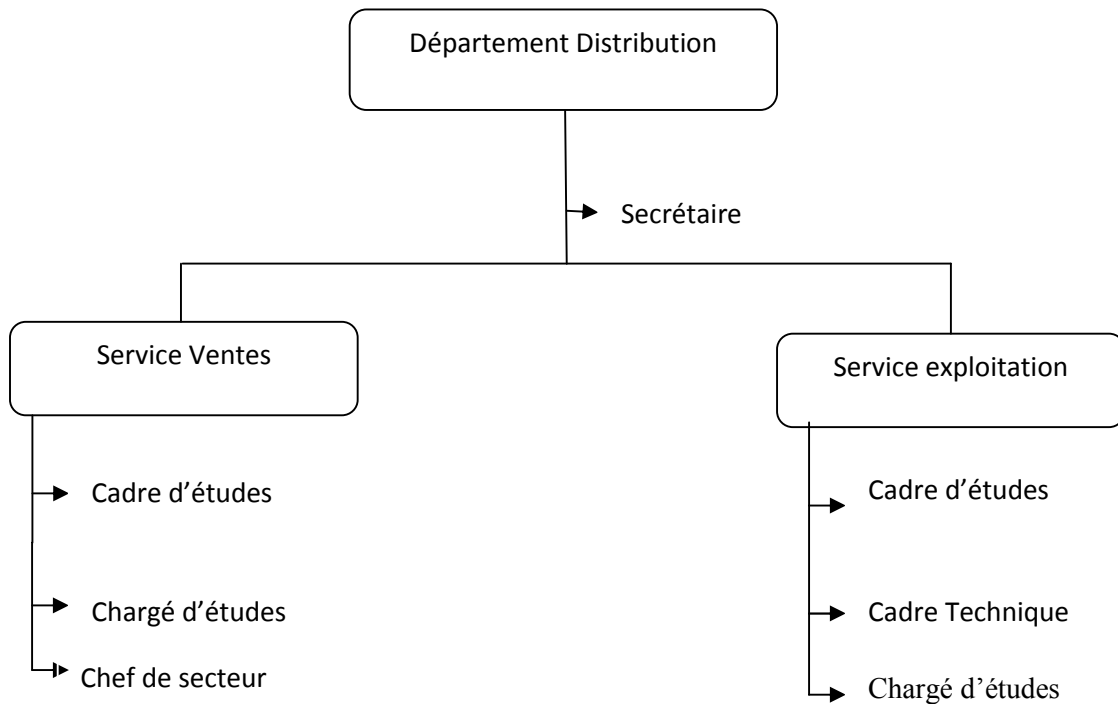


Figure I.3 : Organigramme du département distribution

- **Département technique & maintenance**

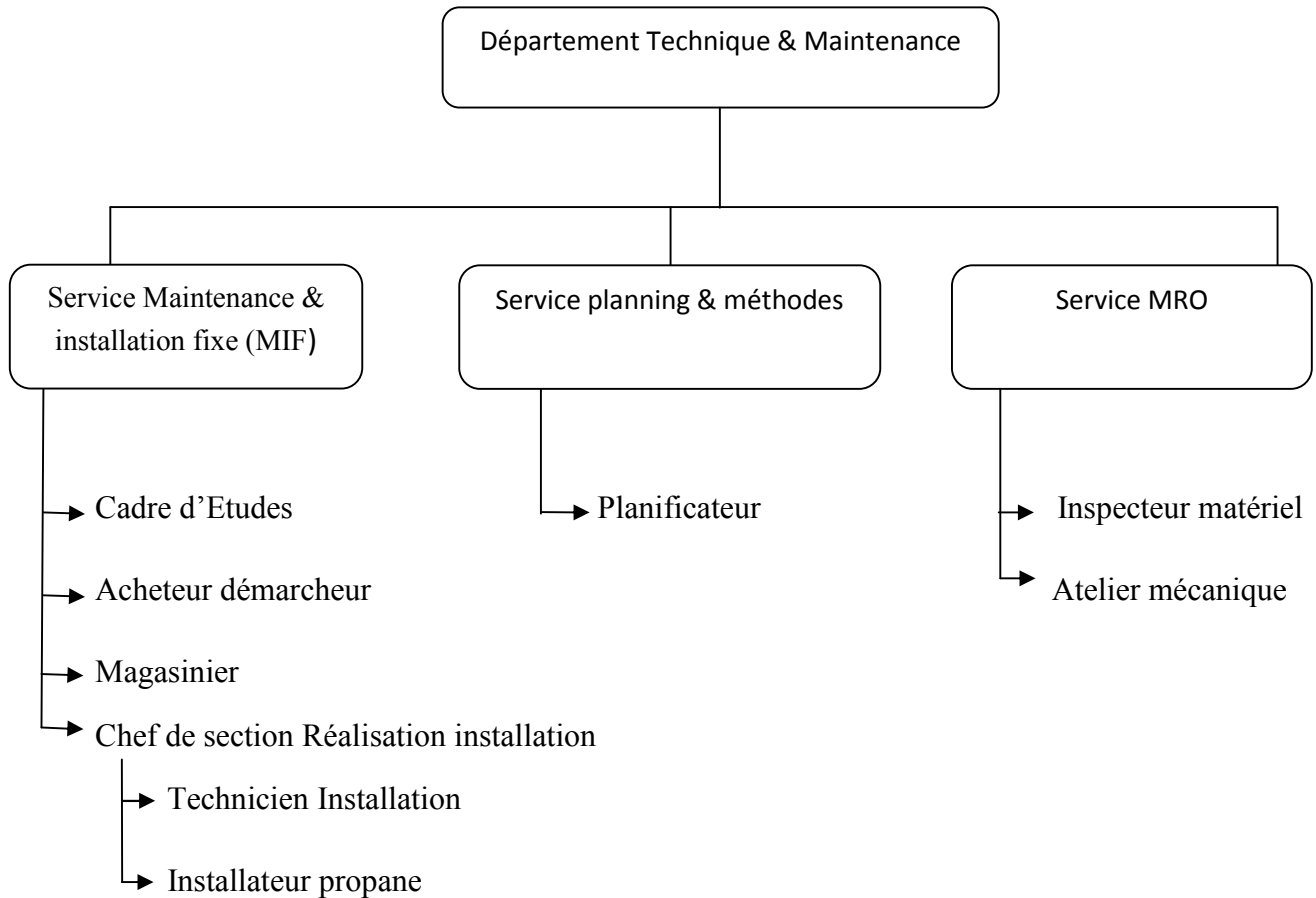


Figure I.4 : Organigramme du département technique & maintenance

I.3.2. Situation informatique du champ d'étude

- **Aspect matériel :**

Un micro-ordinateur HP ayant les caractéristiques suivantes :

- **System d'exploitation :** Windows 7
- **Processeur :** Intel Atom(TM) Cpu D2500 1,86 GHZ 1,86 GHZ
- **Ram :** 1,00 GO
- **Ecran :** 17 pouces
- **Clavier :** azerty.
- **Imprimante :** EPSON : matriciel LQ2090.

Un micro-ordinateur Acer ayant les caractéristiques suivantes :

- **System d'exploitation** : Windows 7
- **Processeur** : Intel core(TM) 2 Duo Cpu E6750 2,66 GHZ 2,66 GHZ
- **Ram** : 1,00 GO
- **Ecran** : 17 pouces
- **Clavier** : azerty.
- **Imprimante** : EPSON : matriciel LQ2090.

Aspect Logiciel :

- **NAFTCOM** : application propre à NAFTAL développé par ses ingénieurs, elle permet de gérer les stocks et de donné la journée comptable.
- **ACCESS** : gère les équipements au niveau du magasin technique.
- **EXCEL, WORD** : pour le traitement des tableaux et des textes
- **POWERPOINT** : pour la planification des schémas de construction des stations services.

I.4. Etude de l'existant

I.4.1 Introduction : L'étude de l'existant est la première phase du processus unifié. Elle a pour but ultime la clarification du champ de notre investigation. On doit dans cette partie prendre en considération :

- ❖ L'ensemble des objectifs fixés.
- ❖ Etudier profondément les traitements pour lesquels on souhaite automatiser le fonctionnement.
- ❖ La capture des besoins et l'analyse.

I.4.2. Etude des postes :

L'étude des postes de travail, est l'étape essentielle pour élaborer un system d'information fiable, la structure organisationnelle de notre champ d'étude est constituée de deux postes de travail qui se situent dans deux services différent.

L'étude se fait en suivant les caractéristiques suivantes :

- Désignation des postes.
- Rattachement
- Responsabilité
- Nombre de personnel
- Les taches
- Les documents utilisés :
 - Document sortant du poste
 - Document entrant au poste

I.4.2.1. La liste des postes de travail dans le champ d'étude :

La liste comporte deux postes :

- Chargé d'Etudes (service vente)
- Chef de section Réalisation installation (service M.I.F)

Fiche d'étude du poste:**Désignation du poste :** Chargé d'études**Section de rattachement :** Service vente**Nombre d'employé :** 01**Responsabilité :** Il assure l'exécution des taches de ce service dans la limite de ces prérogatives (pouvoir)**Taches du poste**

- Participer a la définition des objectifs de vente du District.
- Suivre et contrôler les ventes du District.
- Veiller a la disponibilité permanente du produit.
- Définir toutes les actions de promotion pour augmenter le volume des ventes.
- Développer la force de vente du District.
- Suivre en permanence l'évolution du marché des GPL.
- Assurer le bon suivi des créances et recouvrement relevant des clients du District.
- Veiller sur le suivi de l'activité comptes clients.
- Etablir les réponses aux différentes demandes de la clientèle.
- Veiller a la fiabilité de l'information qu'il communique a sa hiérarchie.
- Engager toute action pour que les attributions du Service soient réalisées dans les délais et ormes requis, avec les moyens humains et matériels mis a sa disposition.
- Veiller au maintien de l'image de marque de l'entreprise en général et de la Branche en particulier.
- Informer la hiérarchie de toute difficulté constatée durant l'exercice de ses fonctions.

Documents entrant	Provenant	Documents sortants	Destination
Demande du client	Client	Demande de travaux	Chef de section Réalisation Installation
PV d'installation	Chef de section Réalisation Installation	Facture Réponse a la demande du client	Client Client

Fiche d'étude du poste :

Désignation du poste : Chef de section réalisation installation

Section de rattachement : service maintenance et installation fixe

Nombre d'employé : 01

Responsabilité : Il assure l'exécution des taches de ce service à la limite de ces prérogatives (pouvoir)

Taches du poste

- Participer à l'élaboration du programme de maintenance des installations fixes.
- Etablir le planning des opérations de contrôle et de visites périodiques et réglementaires.
- Préparer, ordonner et vérifier les travaux effectués.
- Tenir à jour les fichiers historiques des équipements.
- Tenir à jour le registre des contrôles réglementaires.
- Tenir à jour le registre des opérations de maintenance.
- Veiller aux respects des consignes de sécurité et environnement lors des travaux de maintenance sur les équipements.
- Participer à la mise en œuvre et l'amélioration du système QSE.
- Veiller au respect des consignes HSE.

Documents entrant	Provenant	Documents sortants	Destination
Demande de travaux	Chargé d'études	Ordre de travaux	Equipe d'intervention
Fiche des travaux	Equipe d'intervention	PV d'installation	Chargé d'études

I.4.3. Etude des documents :

Au cours de l'étape de la collecte des informations nous avons étudiés les différents documents utilisés entre divers acteurs, et nous avons mis l'accent sur l'importance de ces documents car ils nous ont offert la possibilité de relever des données nécessaires à notre étude, examiner les relations entre les éventuelles rubriques et les contrôles qui peuvent y être appliqués.

Pour chaque document on précisera ses caractéristiques et pour chacune de ses informations, on précisera la longueur, le type et les observations sur éventuelles rubriques.

Le type des rubriques peut être :

- **A** : Alphabétique
- **N** : Numérique
- **AN** : Alphanumérique
- **D** : Date

Les documents sont divisés en deux catégories

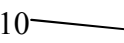
- Documents interne
- Documents externe

- **Documents interne** : C'est tous les documents utilisés à l'intérieur du champ d'étude.
- **Documents interne de position** : sont des documents utilisés uniquement par le créateur.
- **Documents interne de liaison** : sont des documents qui transitent par le champ d'étude.
- **Documents externe** : C'est tous les documents qui proviennent ou destinés à l'extérieur du champ d'étude.

II.4.3.1 Liste des documents existant :

N° Document	Désignation
01	-Demande du client
02	-Réponse a la demande du client
03	-Demande d'installation
04	-Ordre de travaux
05	-PV d'installation
06	-Facture

FICHE D'ANALYSE DU DOCUMENT N° 1**Identification****Désignation** : demande du client**Code** : /**Rôle** : demander une installation propane**Nature** : externe**Nombre d'exemplaire** : 01**Emetteur** : Client**Récepteur** : Chargé d'étude**Désignation**

Rubrique	Type	Taille	Observation
- Nom ou raison social	AN	11	
- Etabli le	D	08	JJ-MM-AAAA
- Localité	A	30	
- Tel /Fax/Lettres/Autres	AN	20	
- Objet de la demande	AN	80	
- Utilisation	A	10	 DOMESTIQUE COMMERCIAL

FICHE D'ANALYSE DU DOCUMENT N° 2**Identification****Désignation** : Réponse à la demande du client**Code** : /**Rôle** : Etablissement d'une réponse à la demande du client**Nature** : externe**Nombre d'exemplaire** : 01**Emetteur** : chargé d'études**Récepteur** : Client**Désignation**

Rubrique	Type	Taille	Observation
- Date	D	08	JJ-MM-AAAA
- Objet	A	60	
- Nom ou raison sociale	A	30	
- Adresse	AN	60	
- Utilisation	A	10	DOMESTIQUE COMMERCIAL
- Num de tel	N	10	
- Autres renseignements	AN	60	
- Avis du technicien	A	12	FAVORABLE DEFAVORABLE

FICHE D'ANALYSE DU DOCUMENT N° 3**Identification****Désignation** : demande d'installation**Code** : /**Rôle** : demande d'installation**Nature** : interne**Nombre d'exemplaire** : 01**Emetteur** : chargé d'études**Récepteur** : chef de section réalisation installation**Désignation**

Rubrique	Type	Taille	Observation
- Code du client	AN	08	
- Nom ou raison sociale	A	30	
- Adresse du projet	AN	60	
- Commune	A	30	
- Daïra	A	30	
- Num de téléphone	N	10	
- Observation	A	80	

FICHE D'ANALYSE DU DOCUMENT № 4			
Identification			
<p>Désignation : Ordre de travail</p> <p>Code : /</p> <p>Rôle : Sélection des agents adapté aux travaux</p> <p>Nature : externe</p> <p>Nombre d'exemplaire : 01</p> <p>Emetteur : chef de section réalisation installation</p> <p>Récepteur : équipe d'intervention</p>			
Désignation			
Rubrique	Type	Taille	Observation
- Numéro l'ordre de travaux	N	11	JJ-MM-AAAA
- Date	D	08	
- Nom du client	A	20	
- Adresse client	AN	30	
- Noms des agents	A	80	

FICHE D'ANALYSE DU DOCUMENT N° 5**Identification****Désignation :** Fiche des travaux**Code :** /**Rôle :** compte rendu des travaux effectués**Nature :** externe**Nombre d'exemplaire :** 01**Emetteur :** équipe d'intervention**Récepteur :** chef de section réalisation installation**Désignation**

Rubrique	Type	Taille	Observation
- Réalisateur	A	30	
- Date de réalisation	D	08	JJ-MM-AAAA
- Nom du Client	A	20	
- Adresse client	AN	30	
- Commune	A	30	
- Daira	A	30	
- Activité du client	A	08	
- Type citerne	AN	30	Pour les citernes
- Nombre de bouteilles	N	02	Pour les installations propane 35kg
- Date de mise en service	D	08	JJ-MM-AAAA
- Observation	AN	80	

FICHE D'ANALYSE DU DOCUMENT N° 6**Identification****Désignation :** Facture**Code :** /**Rôle :** atteste l'achat ou la vente de biens ou services.**Nature :** externe**Nombre d'exemplaire :** 01**Emetteur :** chargé d'études**Récepteur :** client**Désignation**

Rubrique	Type	Taille	Observation
- Nom du Client	A	20	
- Date	D	08	JJ-MM-AAAA
- Adresse client	AN	30	
- Activité	A	08	
- Montant a payé	N	10	En Da
- Mode de paiement	A	06	Espèce Cheque
- Observation	AN	80	

I.4.4. Le Modèle Organisationnel de Traitements (MOT) :

Introduction : Le modèle organisationnel de traitements (MOT) permet de représenter l'ensemble des traitements du domaine d'étude, en tenant compte de son organisation.

Pour construire un MOT, les questions à poser sont :

- **QUI** fait le traitement ?
- **OU est-ce que** les traitements sont-ils exécutés ?
- **QUAND est-ce que** les traitements se sont déroulés ?
- **COMMENT** ?
- **POURQUOI** ?

Et en exprimant successivement :

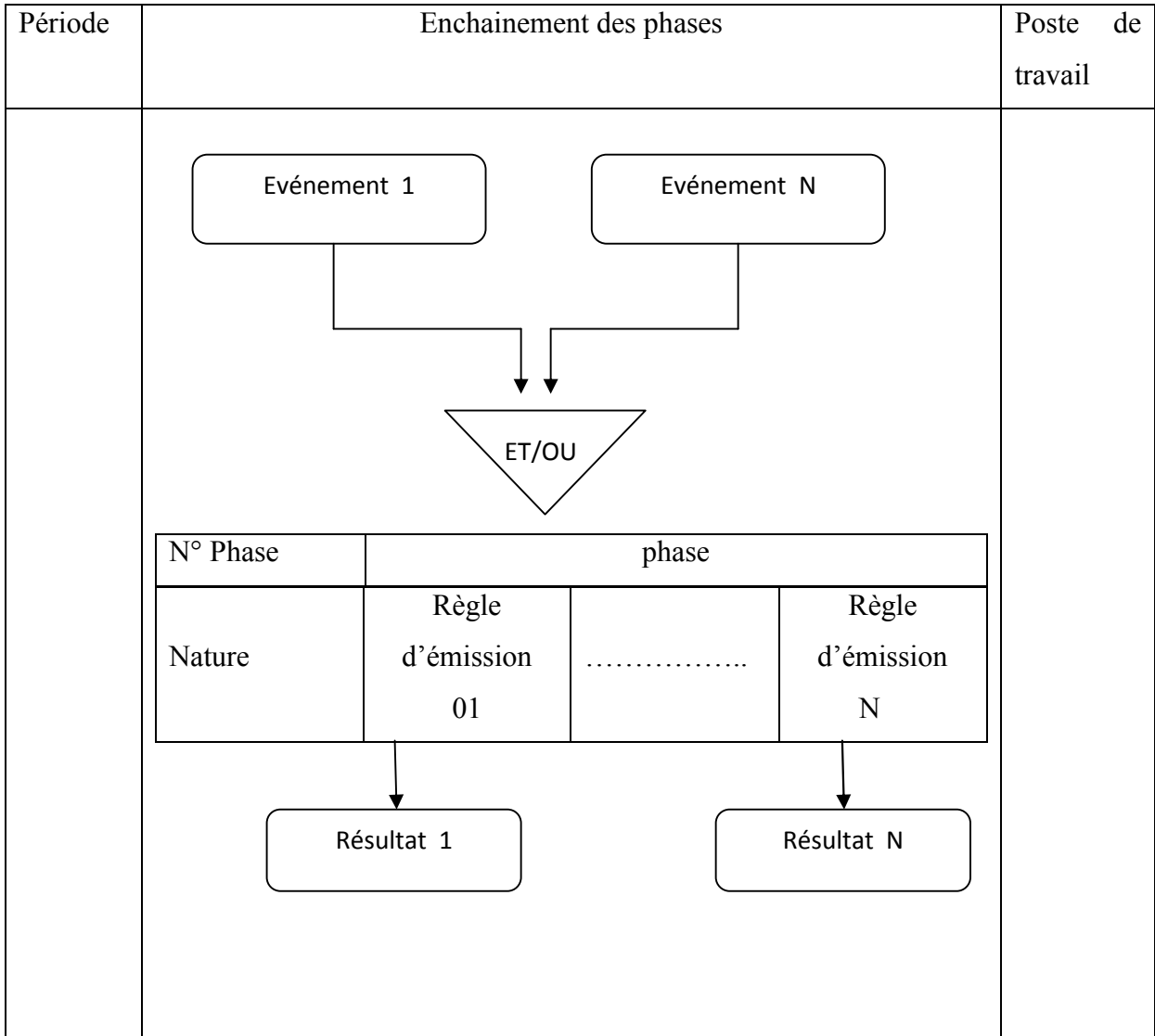
- Nature des traitements qui peuvent être le choix entre un traitement automatique et un traitement manuel.
- Attribution des traitements au poste de travail.
- Périodicité des traitements.
- Enchaînement des traitements.

Les concepts de base :

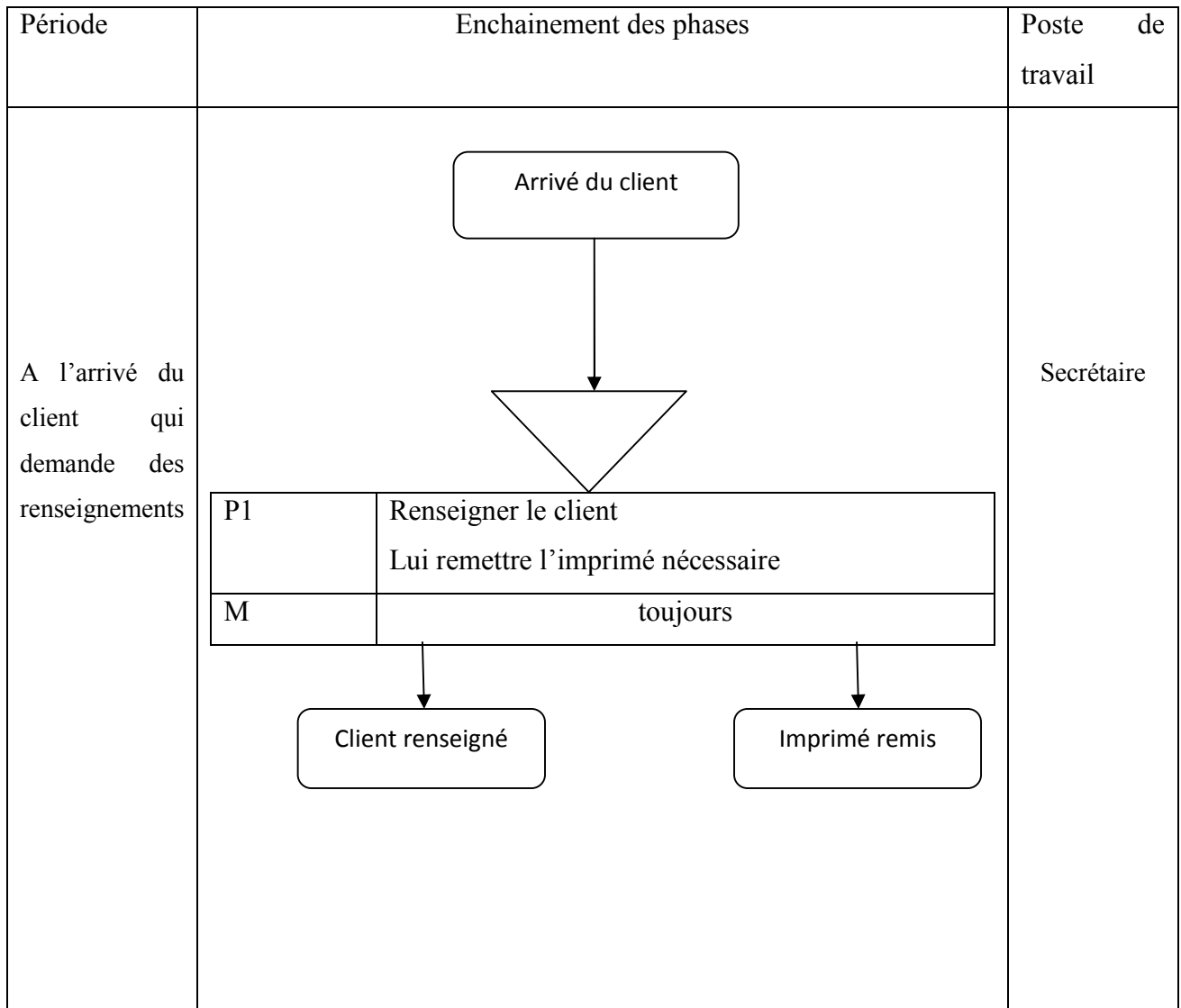
- **Procédure** : c'est un enchaînement de phases déclenchés par un ou plusieurs événements externes.
- **Tache** : c'est un ensemble de traitements élémentaires exécuté à l'intérieur d'une phase.
- **Synchronisation** : c'est une condition booléenne traduisant les règles de gestion et d'organisation pour déclencher l'exécution d'une ou plusieurs tâches.
- **Résultats** : c'est le produit d'exécution d'une phase, il est un fait réel de même nature que l'événement et peut être déclenché par une autre phase.
- **Règles d'émission** : ce sont des règles traduisant les règles de gestion et d'organisation auxquelles est soumise l'émission des résultats d'une phase.
- **Règles d'organisation** : c'est l'expression de l'organisation mise en place en termes de travail, nature de traitements de chronologie.

Le formalisme d'un MOT :

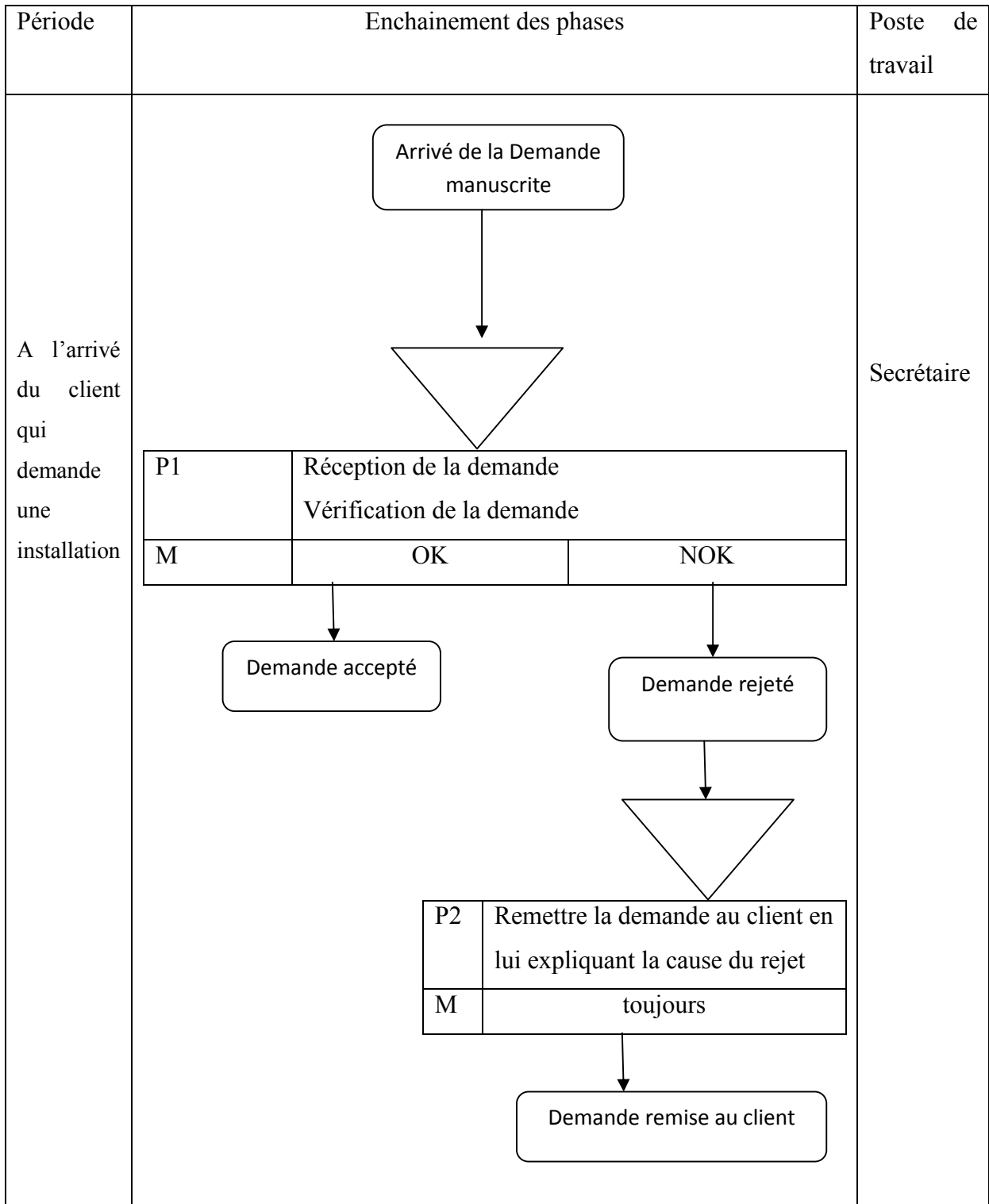
Le formalisme utilisé dans notre étude pour décrire le MOT est représenté par la figure suivante :



Procédure 01 : demande de renseignement



Procédure 02 : demande d'installation



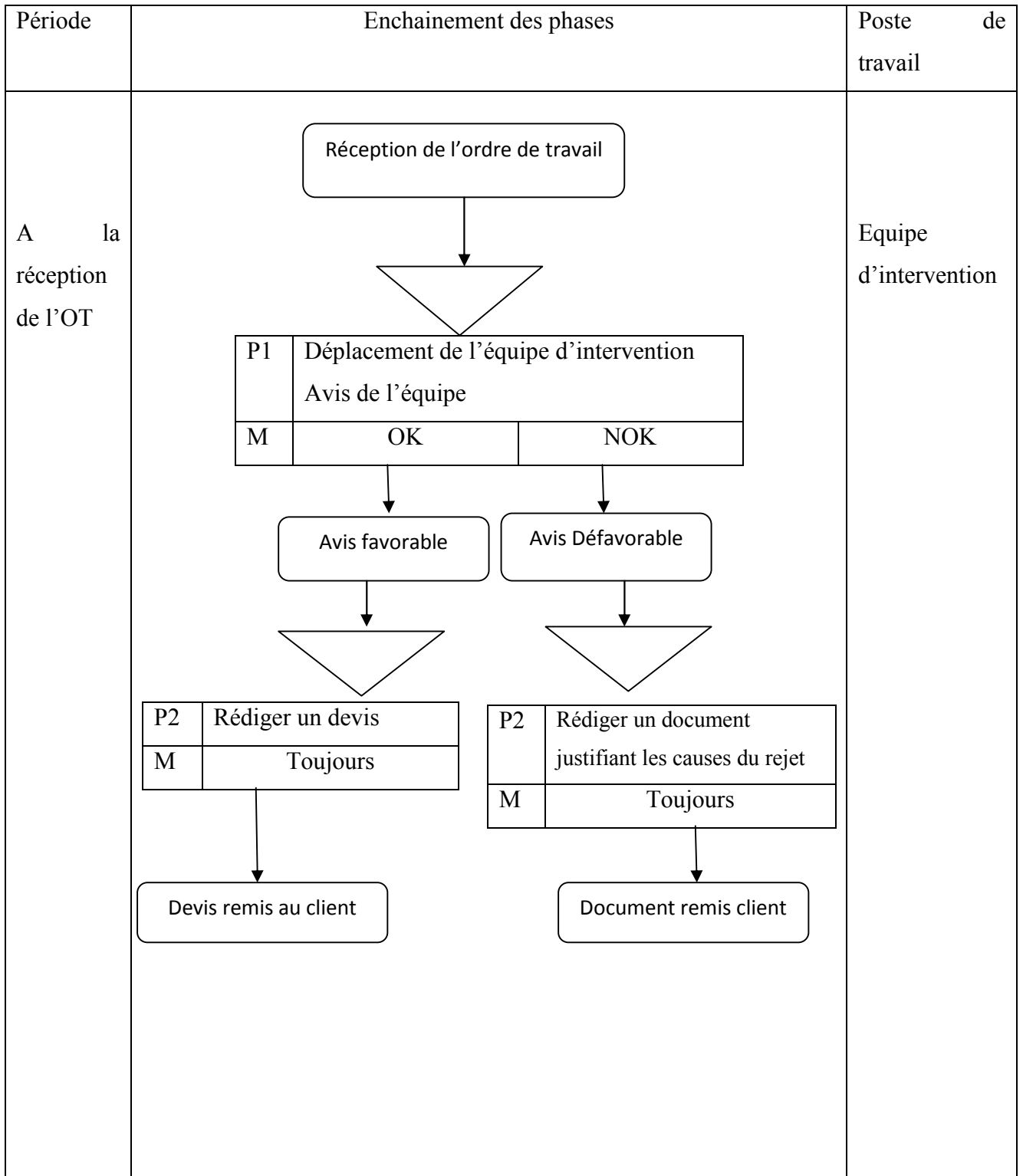
Procédure 03 : transmission de la demande au service MIF

Période	Enchaînement des phases	Poste de travail
<p>Après la réception de la demande</p>	<pre> graph TD A["Demande d'installation établie par le chargé d'étude"] --> B[] subgraph Table B1["P1"] --- B2["envoi du document"] B3["M"] --- B4["toujours"] end B --> C["Demande transmise au service MIF"] </pre>	<p>Chargé d'étude</p>

Procédure 04: Réception de la demande par le service MIF

Période	Enchaînement des phases	Poste de travail				
<p>Après avoir accepté la demande</p>	<pre> graph TD A(Arrivé de la demande d'installation) --> B() B --> C[Table] C --> D(OT transmis a l'équipe d'intervention) </pre> <table border="1" data-bbox="423 781 1088 1050"> <tr> <td data-bbox="423 781 498 991">P1</td> <td data-bbox="498 781 1088 991"> Réception de demande Etablissement d'un ordre de travaux (OT) Sélection des agents Transmission de l'OT a l'équipe d'intervention </td> </tr> <tr> <td data-bbox="423 991 498 1050">M</td> <td data-bbox="498 991 1088 1050">toujours</td> </tr> </table>	P1	Réception de demande Etablissement d'un ordre de travaux (OT) Sélection des agents Transmission de l'OT a l'équipe d'intervention	M	toujours	<p>chef de section réalisation & installation</p>
P1	Réception de demande Etablissement d'un ordre de travaux (OT) Sélection des agents Transmission de l'OT a l'équipe d'intervention					
M	toujours					

Procédure 05: Déplacement de l'équipe sur le terrain



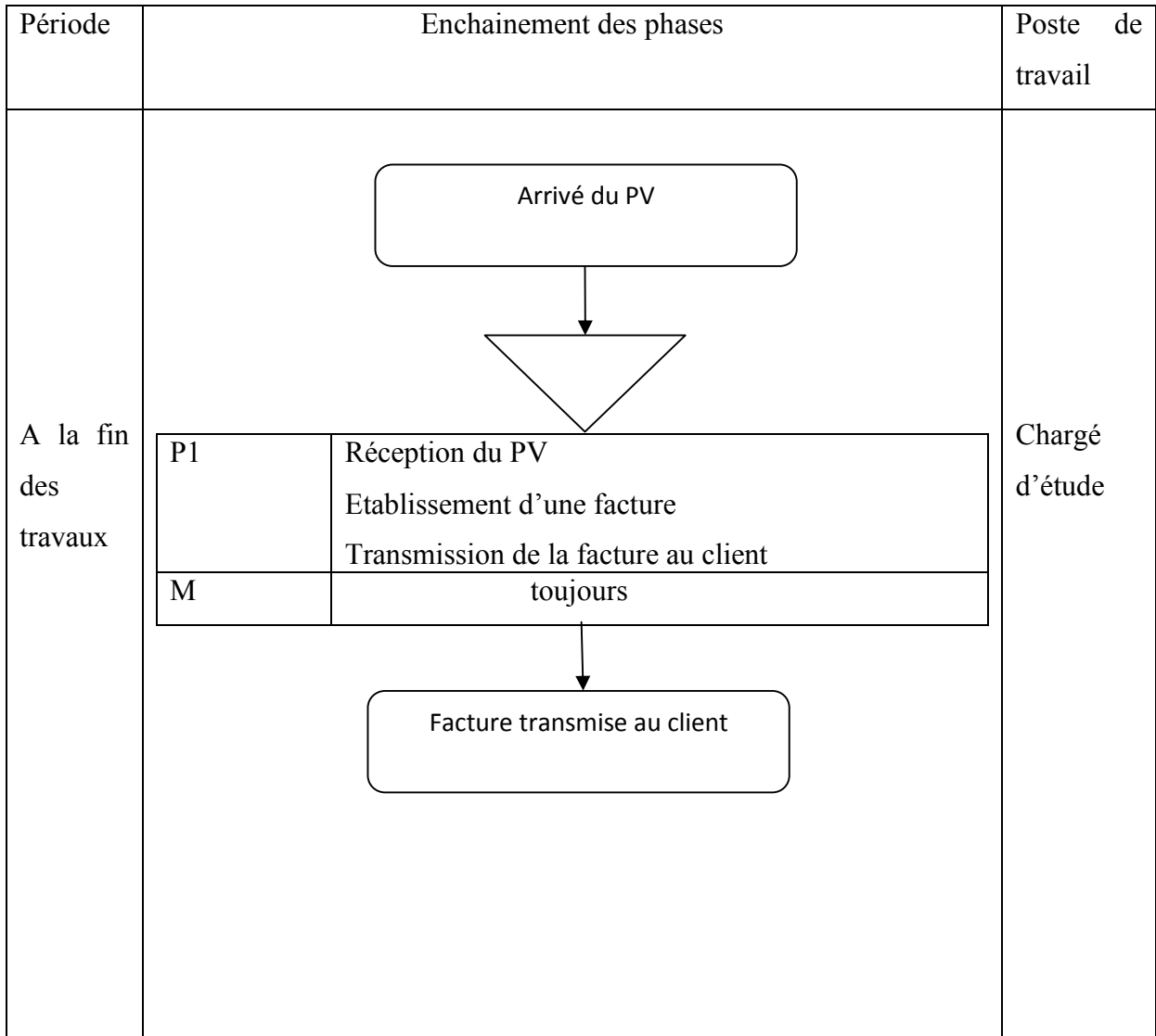
Procédure 06 : Réalisation des travaux

Période	Enchaînement des phases	Poste de travail
<p>Après le paiement du devis</p>	<pre> graph TD A[Devis payé] --> B[] subgraph Box [] direction TB C[P1] --- D[Réalisation des travaux Etablissement d'une fiche de travaux (FT) Transmission de la FT au service MIF] E[M] --- F[toujours] end B --- C F --- G[Fiche des travaux transmise au service MIF] </pre> <p>The flowchart starts with a rounded rectangle 'Devis payé'. An arrow points down to an inverted triangle. Below this is a large rectangular box containing two smaller boxes. The top box is labeled 'P1' and contains the text: 'Réalisation des travaux', 'Etablissement d'une fiche de travaux (FT)', and 'Transmission de la FT au service MIF'. The bottom box is labeled 'M' and contains the text 'toujours'. An arrow points down from the bottom of the large box to a final rounded rectangle: 'Fiche des travaux transmise au service MIF'.</p>	<p>Equipe d'intervention</p>

Procédure 07 : Fin des travaux

Période	Enchaînement des phases	Poste de travail
A la fin des travaux	<pre> graph TD A[Fiche des travaux transmise au service MIF] --> B[] B --> C subgraph C [] direction TB C1[P1] --- C2[Réception de la FT Etablissement du procès verbal (PV) de l'intervention Transmission du procès verbal au chargé d'étude] C3[M] --- C4[toujours] end C --> D[PV transmis au service distribution] </pre>	Chef de section réalisation & installation)

Procédure 08 : Facturation



I.4.5. Diagramme des flux

Afin de bien cerner notre champ d'étude, il convient de recenser les différents flux d'informations circulants entre les différents acteurs

Le diagramme des flux permet :

- De mieux identifier les acteurs internes et externes par rapport au domaine d'étude
- Identifier les flux échangés entre les acteurs.
- Délimiter le domaine d'étude.

I.4.5.1. Concepts utilisés :

Domaine d'étude : est définie par un ensemble de personnes qui analyse la circulation de l'information au sein de l'organisme. Il délimite l'étude à une ou plusieurs activités précises.

Acteur : est un émetteur ou un récepteur d'informations liées à une activité. Selon le cas, il peut s'agir d'une catégorie de personnes ou d'un service. Un acteur reçoit une information qui lui permet d'agir en transformant l'information et en la renvoyant à un, ou plusieurs autres acteurs. Les acteurs sont représentés par leur rôle dans l'activité étudiée. On distingue :

- les acteurs internes qui font partie du domaine d'étude
- les acteurs externes qui ne font pas partie du domaine mais qui ont des échanges avec les acteurs internes dans le cadre de l'activité étudiée.

Flux : il est représenté par une flèche orientée de l'acteur émetteur vers l'acteur récepteur.

Le libellé du flux est inscrit à côté de la flèche tracée.

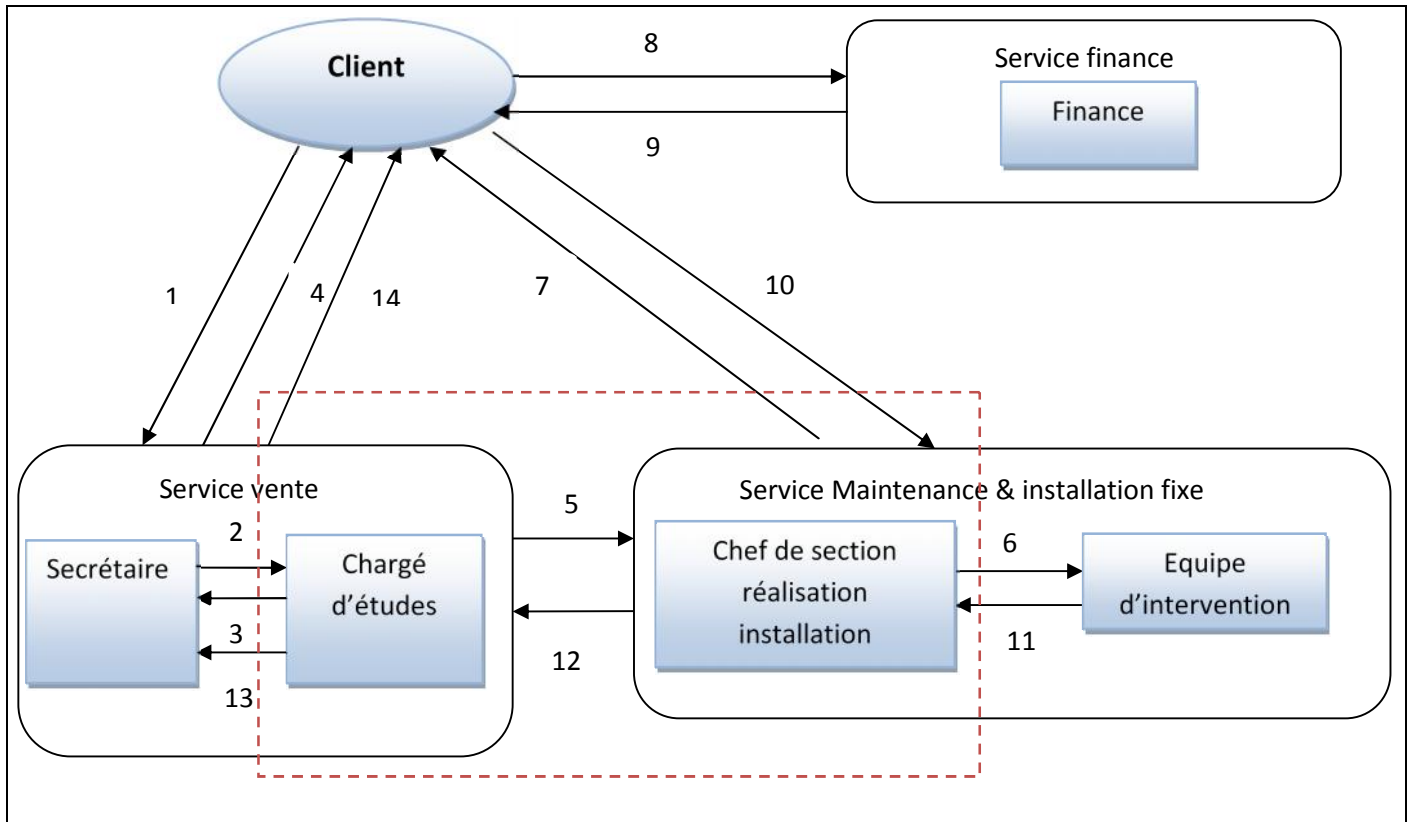
Domaine connexe : c'est un composant de notre organisme interagissant avec notre domaine d'étude. C'est un acteur interne à l'organisme, mais externe à notre domaine d'étude.

Formalisation graphique : un flux de données est représenté graphiquement par une flèche orientée du composant émetteur du flux vers le composant récepteur. Le numéro du flux est inscrit sur la flèche tracée.



I.4.5.2. Diagramme des flux d'informations :

Le schéma suivant représente le diagramme des flux d'informations :




 : Champ d'étude

Figure I.5 : diagramme des flux d'information

Liste des flux d'information :

N°	DESCRIPTION
1	Demande du client
2	Transmission de la demande
3	Réponse a la demande
4	Transmission de la réponse
5	Demande d'installation
6	Ordre de travaux

7	Devis
8	Paiement du devis
9	Justificatif de paiement
10	Dépôt du justificatif
11	Fiche de travaux
12	Procès verbal
13	facture
14	Remise de la facture

I.4.6. Critiques et suggestions

Introduction : après avoir étudié le système existant, nous avons identifié certaines insuffisances et anomalies, d'où la nécessité de suggérer et de proposer des améliorations qui serviront à déterminer les solutions futures, on peut séparer les anomalies en deux types :

- **Les anomalies organisationnelles** : elles concernent les méthodes de travail et l'organisation des postes de travail et des tâches.
- **Les anomalies informationnelles** : elles concernent les documents.

I.4.6.1. Critiques d'ordre organisationnel :

Critique : Toutes les procédures sont faites manuellement, ce qui entraîne l'absence de la précision dans le travail et les risques d'erreurs.

Suggestion : Automatisation de certaines tâches manuelles.

Critique : Transfert manuel des documents d'un service à un autre.

Suggestion : Utilisation du réseau local pour le transfert des documents.

Critique : utilisation du faxe pour l'envoi de la demande de création du code client a la direction général de NAFTAL

Suggestion : Introduire les nouvelles technologies de communication et d'information (Internet...)

I.4.6.2. Critiques d'ordre informationnel :

Critique : Manipulation d'une masse très importante de documents, ce qui entraîne la difficulté concernant la recherche des informations et aussi l'insécurité des documents(les documents peuvent se perdre facilement).

Suggestion : Utiliser une base de donné pour archiver les données et faciliter leurs recherche.

Critique : Saisie manuel des documents d'où l'illisibilité et l'ambiguïté des écritures.

Suggestion : Utiliser des contrôles automatiques lors de la saisie des informations.

I.4.7. Proposition de solution :

Notre solution consiste à réaliser un logiciel client/serveur connecté à une base de donnée sur laquelle seront enregistrées toutes les informations nécessaire pour la gestion des installations propane. Ce logiciel sera installé au poste du chargé d'études (poste administrateur de l'application) et sera relié également au poste du chef de section réalisation et installation ; de ce fait ces derniers n'auront plus à remplir les informations manuellement la seule chose manuelle qui devra être faite est la signature des documents après impression

❖ Les avantages d'une solution informatique client/serveur :

L'architecture client serveur s'appuie sur un poste central, le serveur, qui envoie des données aux machines clientes et elle offre plusieurs avantages :

- **Unicité de l'information** : Les informations restent identiques. Chaque utilisateur accède aux mêmes informations.
- **Meilleure sécurité** : Lors de la connexion un PC client ne voit que le serveur, et non les autres PC clients. De même, les serveurs sont en général très sécurisés contre les attaques de pirates.
- **Meilleure fiabilité** : En cas de panne, seul le PC serveur fait l'objet d'une réparation, et non le PC client.
- **Facilité d'évolution** : Une architecture client/serveur est évolutive car il est très facile de rajouter ou d'enlever des clients, et même des serveurs.

- **Les qualités intrinsèques** : La souplesse et la puissance.

II.5. Conclusion

L'étude de l'existant nous a permis de relever les insuffisances du système existant, ainsi que les différents inconvénients des procédures de travail.

Dans les chapitres qui suivent, nous allons mettre en œuvre toutes nos connaissances acquises afin de remédier aux problèmes évoqués.

CHAPITRE

Généralités sur les réseaux et l'architecture client/serveur

I. Introduction

L'architecture client/serveur est le résultat d'un ensemble d'évolutions technologique apparu dans les vingt dernières années : capacité mémoire, performance des processus et des réseaux et évolution des logiciels. Malgré sa relative jeunesse, l'informatique client/serveur ne laisse intacte aucune facette de l'industrie informatique. Les applications client/serveur sont distribuées sur plusieurs sites par conséquent, il est nécessaire de les faire communiquer afin qu'elles coopèrent pour la réalisation d'un travail commun.

Dans ce chapitre nous proposons une vue d'ensemble de l'architecture client/serveur et nous définissons des notions de base indispensable a la compréhension du modèle client/serveur.

II. Réseaux informatiques:

II.1. Définition : Ensemble d'ordinateurs reliés entre eux grâce à des lignes physiques et échangeant des informations sous forme de données numériques.

Un réseau informatique peut servir plusieurs buts distincts :

- Le partage de ressources (fichiers, applications ou matériels, connexion à internet etc.).
- La communication entre personnes (courrier électronique, etc.).
- La communication entre processus (entre des ordinateurs industriels par exemple).
- La garantie de l'unicité et de l'universalité de l'accès à l'information (bases de données en réseau).

II.2. Classification des réseaux : [1]

Les réseaux sont généralement classifiés suivant le type de liaisons utilisées (la topologie) et la surface géographique qu'ils couvrent.

II.2.1. En fonction de la distance :

La principale classification des réseaux est faite selon leur taille. On distingue : réseau local (LAN), réseau métropolitain (MAN) et réseau étendu (WAN).

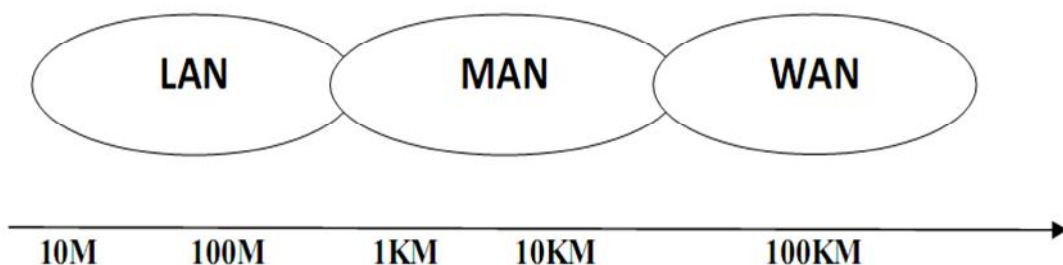


Figure II.1 : Classification des réseaux informatiques selon leur taille

- a) **Réseau local LAN (Local Area Network):** Il est formé d'un ensemble de stations situées dans une même zone géographique limitée (qui ne dépasse pas 1 kilomètre), il peut se développer dans une petite entreprise, un immeuble...etc.
- b) **Réseau métropolitain MAN (Metropolitan Area Network) :** Souvent utilisé pour interconnecter un ensemble de réseaux locaux géographiquement dispersés, il peut couvrir la taille d'une ville.
- c) **Réseau étendu WAN (Wide Area Network) :** Il relie plusieurs réseaux locaux entre eux. Les réseaux étendus couvrent une zone géographique importante (grande distance à l'échelle d'un pays, d'un continent, voire de la planète entière), Le plus grand WAN est le réseau Internet.

II.2.2. En fonction de la topologie : [2]

Un réseau informatique est constitué d'ordinateurs reliés entre eux grâce à des lignes de communication (câbles réseaux, etc.) et des éléments matériels (cartes réseau, ainsi que d'autres équipements permettant d'assurer la bonne circulation des données). L'arrangement physique, c'est-à-dire la configuration spatiale du réseau est appelé topologie physique. On distingue :

a) Topologie en bus :

Une topologie en bus est l'organisation la plus simple d'un réseau. En effet, 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.

Les réseaux en bus permettent de relier simplement de multiple clients, mais pose des problèmes quand deux clients veulent transmettre des données au même moment sur le même bus. Les systèmes qui utilisent une topologie en bus, ont normalement un gestionnaire de collision qui gère l'accès au bus.

b) Topologie en étoile :

Dans une topologie de réseau en étoile, les équipements du réseau (nœuds) sont reliés à un système matériel central appelé concentrateur (en anglais hub, littéralement "moyen de roue" - à rayons -). Celui-ci a pour rôle d'assurer la communication entre les différents équipements du réseau.

c) Topologie en anneau :

Toutes les entités sont reliées entre elles dans une boucle fermée. Les données circulent dans une direction unique, d'une entité à la suivante. Une entité n'accepte une

donnée en circulation sur l'anneau que si elle correspond bien à son adresse. Dans le cas contraire, l'entité en question fait passer la donnée à l'entité suivante.

En réalité, dans une topologie en anneau, les ordinateurs ne sont pas reliés en boucle, mais sont reliés à un répartiteur (appelé MAU, Multistation Access Unit) qui va gérer la communication entre les ordinateurs qui lui sont reliés en attribuant à chacun d'entre eux un temps de parole.

d) Topologie en arbre :

Aussi connu sous le nom de topologie hiérarchique, le réseau est divisé en niveaux. Le sommet, le haut niveau, est connectée à plusieurs nœuds de niveau inférieur, dans la hiérarchie. Ces nœuds peuvent être eux-mêmes connectés à plusieurs nœuds de niveau inférieur. Le tout dessine alors un arbre, ou une arborescence.

e) Topologie en maillée : [3]

Une topologie en maillée, est une évolution de la topologie en étoile, elle correspond à plusieurs liaisons point à point. Une unité réseau peut avoir (1, N) connexions point à point vers plusieurs autres unités. Chaque terminal est relié à tous les autres.

Cette topologie se rencontre dans les grands réseaux de distribution (Exemple : Internet). L'information peut parcourir le réseau suivant des itinéraires divers, sous le contrôle de puissants superviseurs de réseau, ou grâce à des méthodes de routage réparties. L'armée utilise également cette topologie, ainsi, en cas de rupture d'un lien, l'information peut quand même être acheminée.

II.3. Architecture des réseaux :

Les réseaux sont structurés du point de vue fonctionnel en deux catégories :

- Les réseaux Poste à Poste (Peer to Peer).
- Les réseaux organiser au tour de serveur (client / serveur).

II.3.1. Architecture d'égal à égal (peer to peer) : [5]

Contrairement à une architecture de réseau de type client/serveur, dans une architecture d'égal à égal (peer to peer), il n'y a pas de serveur dédié. Chaque ordinateur est à la fois client et serveur. Le "égal à égal" ou encore poste à poste représente une alternative plus simple et moins chère au réseau classique, de type Novell ou Windows NT.

Il est donc libre de partager ses ressources (disques, connexion Internet...)

❖ Avantage :

- Un coût réduit : Les coûts engendrés par un tel réseau sont d'ordre matériel (postes de travail + cartes réseau + câbles).

- Faible maintenance.
- Une simplicité à toute épreuve.

❖ **Inconvénient :**

- Ce système n'est pas du tout centralisé, ce qui le rend difficile à administrer.
- La sécurité est très peu présente.
- les réseaux d'égal à égal ne sont valables que pour un petit nombre d'ordinateurs (généralement une dizaine), et pour des applications ne nécessitant pas une grande sécurité.

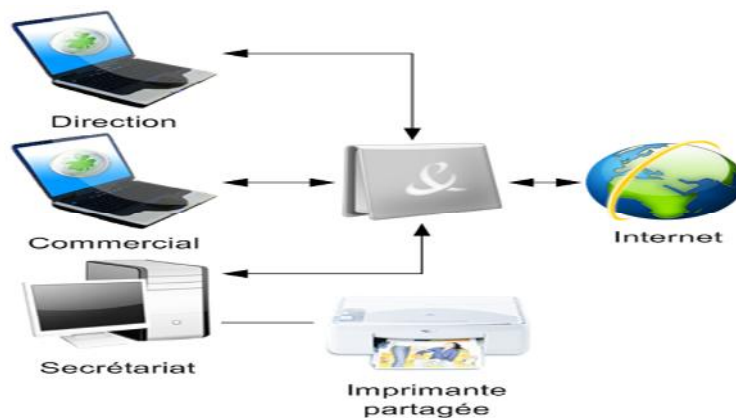


Figure II.2: Système peer to peer

II.3.2. L'architecture client/serveur :

II.3.2.1. Définition : [6]

Le client serveur est avant tout un mode de dialogue entre deux processus. Le premier appelé client demande l'exécution de services au second appelé serveur. Le serveur accomplit les services et envoie en retour des réponses. En général, un serveur est capable de traiter les requêtes de plusieurs clients. Un serveur permet donc de partager des ressources entre plusieurs clients qui s'adressent à lui par des requêtes envoyées sous forme de messages.

II.3.2.2. Notions de base [07] :

- **Le client** : processus demandant l'exécution d'une opération à un autre processus serveur par l'envoi d'un message contenant le descriptif de l'opération à exécuter et attendant la réponse à cette opération par un message en retour.
- **Le serveur** : processus accomplissant une opération sur demande d'un client et transmettant la réponse à ce client
- **Requête** : message transmis par un client à un serveur décrivant l'opération à exécuter pour le compte du client.
- **La réponse** : message transmis par un serveur à un client suite à l'exécution d'une opération contenant le paramètre de l'opération.

➤ **Caractéristiques d'un serveur :**

Le serveur est considéré comme le centre d'un réseau. C'est le cerveau du réseau.

- Il est initialement passif (ou esclave, en attente d'une requête) ;
- Il est à l'écoute, prêt à répondre aux requêtes envoyées par des clients ;
- Dès qu'une requête lui parvient, il la traite et envoie une réponse ;
- Il est composé des mêmes sous-ensembles qu'un ordinateur standard, mais ces sous-ensembles sont beaucoup mieux optimisés ;
- Il contient plus de mémoire vive ;
- Son contrôleur de disques est de très bonne qualité (SCSI, voire Wide ou Ultra Wide SCSI) ;
- Disques durs de très grande capacité ;
- Microprocesseur(s) de dernière génération ;
- Capacités de gestion de réseau ;

➤ **Caractéristiques d'un client :**

Un client est capable de traiter des informations qu'il récupère auprès du serveur :

- Il est actif en premier (ou maître).
- Il envoie des requêtes au serveur.
- Il attend et reçoit les réponses du serveur.

II.3.2.3. Fonctionnement d'un système client/serveur :

Un système client/serveur fonctionne selon le schéma suivant :

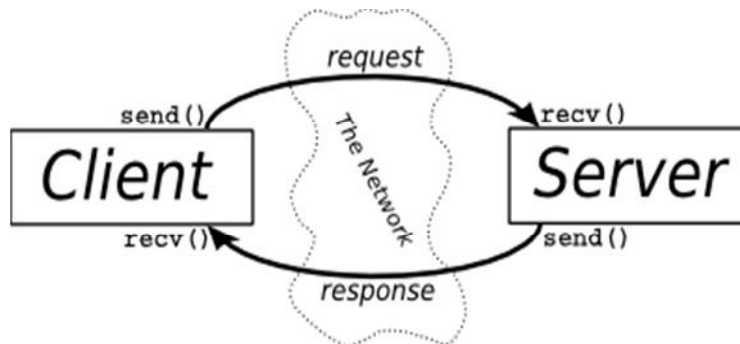


Figure II.3 : Système client/serveur

- ✓ Le client émet une requête vers le serveur grâce à son adresse IP et le port, qui désigne un service particulier du serveur ;
- ✓ Le serveur reçoit la demande et répond à l'aide de l'adresse de la machine cliente et son port ;

❖ **Avantage :**

Le modèle client/serveur est particulièrement recommandé pour des réseaux nécessitant un grand niveau de fiabilité, ses principaux atouts sont :

- **Des ressources centralisées :** comme le serveur se trouve au centre du réseau, il peut alors gérer les ressources communes à tous les utilisateurs, par exemple une base de donnée, et cela pour éviter les problèmes de redondance et de contradiction.
- **Capacité de traitement :** le modèle client/serveur s'appuie sur les progrès réalisés dans le monde du microprocesseur. Les techniques qu'il met en œuvre nécessitent des capacités de traitement importantes que les microprocesseurs d'aujourd'hui ont capable de fournir.
- **Une sécurité meilleure :** car le nombre de point d'entrée (les points de sécurité) permettant l'accès aux données est moins important.
- **L'administration au niveau du serveur :** dans ce modèle les clients ayant peu d'importance, puisque ils ont moins besoins d'être administrés.

- **La capacité de mémorisation** : puisque les postes serveurs doivent gérer simultanément les contextes applicatifs de plusieurs clients, les serveurs disposent de grandes capacités mémoire.

- **Un réseau évolutif** : il est possible de rajouter ou supprimer des clients sans modification majeure et sans perturbation du fonctionnement du réseau grâce à cette architecture.

❖ **Inconvénient :**

L'architecture client/serveur a tout de même quelques lacunes parmi lesquelles :

- **Un coût élevé** : du à la technicité du serveur.
- **Un maillon faible** : le seul maillon faible du réseau dans cette architecture est le serveur, étant donné que tout le réseau est construit autour de lui.

- **Fragilité relative du système** : les données sont concentrées en un seul endroit puisque elles sont dans un environnement centralisé, et le contexte client/serveur peut nécessiter l'installation des données sur des serveurs différents. C'est alors à cause de cette répartition géographique des données et l'utilisation d'un réseau que le système fragilises et ainsi il pose problème de sécurité de données en environnement réparti.

II.3.2.4 Les différentes architectures client/serveur : [2]

a) L'architecture à 2 niveaux :

L'architecture à deux niveaux (aussi appelée architecture 2-tiers, tiers signifiant rangée en anglais) caractérise les systèmes clients/serveurs pour lesquels le client demande une ressource et le serveur la lui fournit directement, en utilisant ses propres ressources. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir une partie du service.

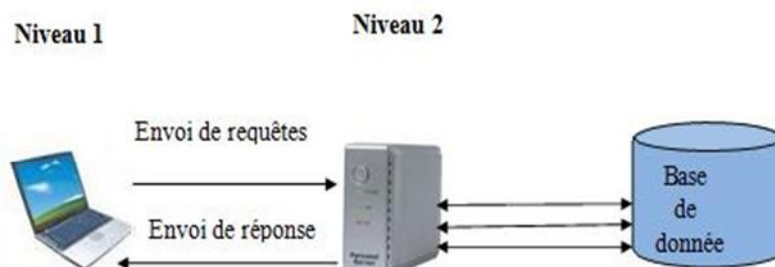


Figure II.4 : L'architecture à 2 niveaux

b) L'architecture à 3 niveaux :

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

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

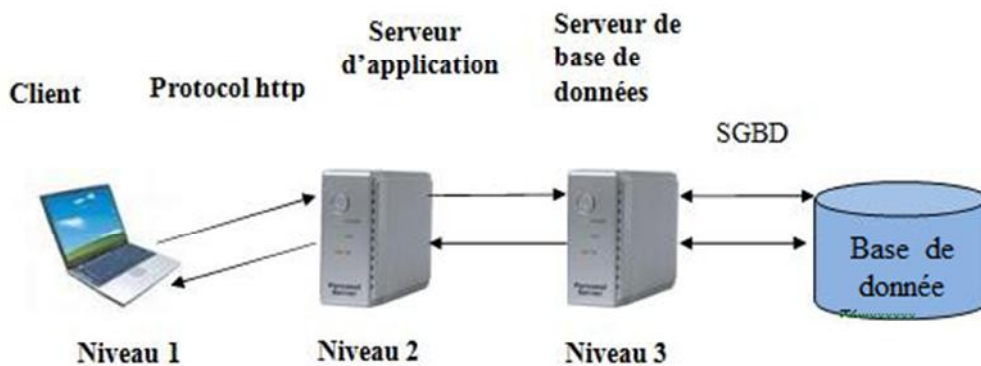


Figure II.5 : L'architecture à trois niveaux.

Le middleware : Est le complément de services du réseau permettant la réalisation du dialogue client/serveur, il prend en compte les requêtes de l'application cliente, les transmet de manière transparente à travers le réseau jusqu'au serveur et prend en compte les données résultants du Serveur vers l'application.

- ✓ L'objectif essentiel du middleware est d'offrir aux applications une interface unifiée permettant l'accès à l'ensemble des services disponibles sur le réseau API.

Remarque :

Etant donné l'emploi massif du terme d'architecture à 3 niveaux, celui-ci peut parfois désigner aussi les architectures suivantes :

- Partage d'application entre client, serveur intermédiaire, et serveur d'entreprise ;
- Partage d'application entre client, serveur d'application, et serveur de base de données d'entreprise qui est l'architecture utilisée dans notre application (serveur d'application « Apache », serveur de base de données « MySQL » et le client « Navigateur »).

c) L'architecture multi niveaux :

Dans l'architecture à 3 niveaux, chaque serveur (niveaux 2 et 3) effectue une tâche (un service) spécialisée. Un serveur peut donc 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...

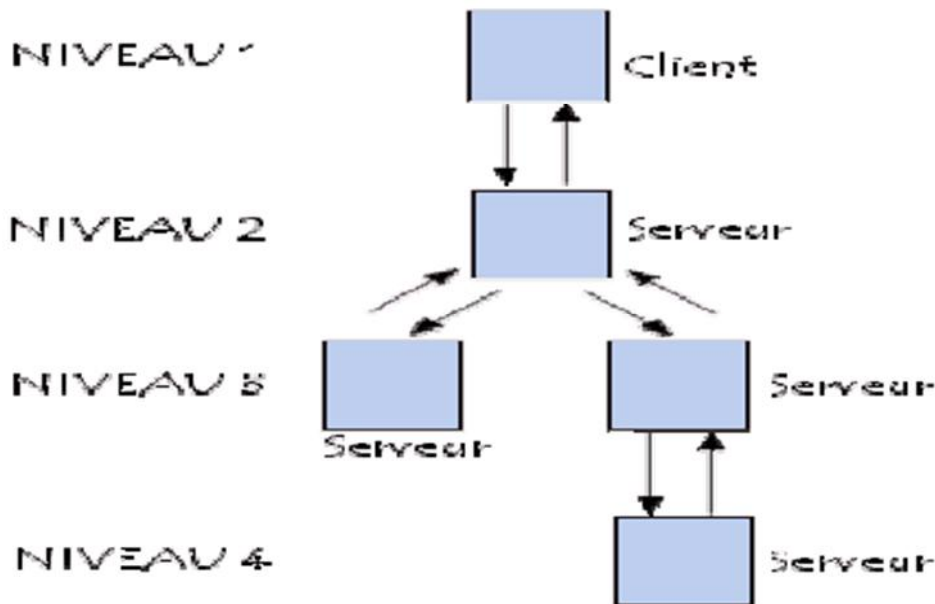


Figure II.6 :L'architecture à multi niveaux.

II.4. Les protocoles : [2]

Définition : Un protocole est un ensemble de règles et de procédures à respecter pour émettre et recevoir des données sur un réseau. Il en existe plusieurs selon ce que l'on attend de la communication. Certains protocoles seront par exemple spécialisés dans l'échange de fichiers (le FTP), la messagerie (le SMTP) d'autres pourront servir à gérer simplement l'état de la transmission et des erreurs (protocole ICMP)...

II.4.1. Le modèle OSI : [8]

Open System Interconnexion c'est-à-dire interconnexion des systèmes ouverts, est un modèle de base qui a été défini par l'ISO (International Standard Organisation) en 1977.

Le modèle de référence est un schéma théorique qui décompose les communications réseaux (l'ensemble des protocoles) en sept couches distinctes où sont définis deux types de relations :

- Relations verticales entre les couches d'un même système (interfaces).
- Relations horizontales relatives au dialogue entre deux couches de même niveau.

Chaque ordinateur du réseau utilise une série de protocoles pour effectuer les fonctions spécifiques à chaque couche.

L'emploi de ce modèle permet de décomposer un processus complexe en unités facilement compréhensibles.

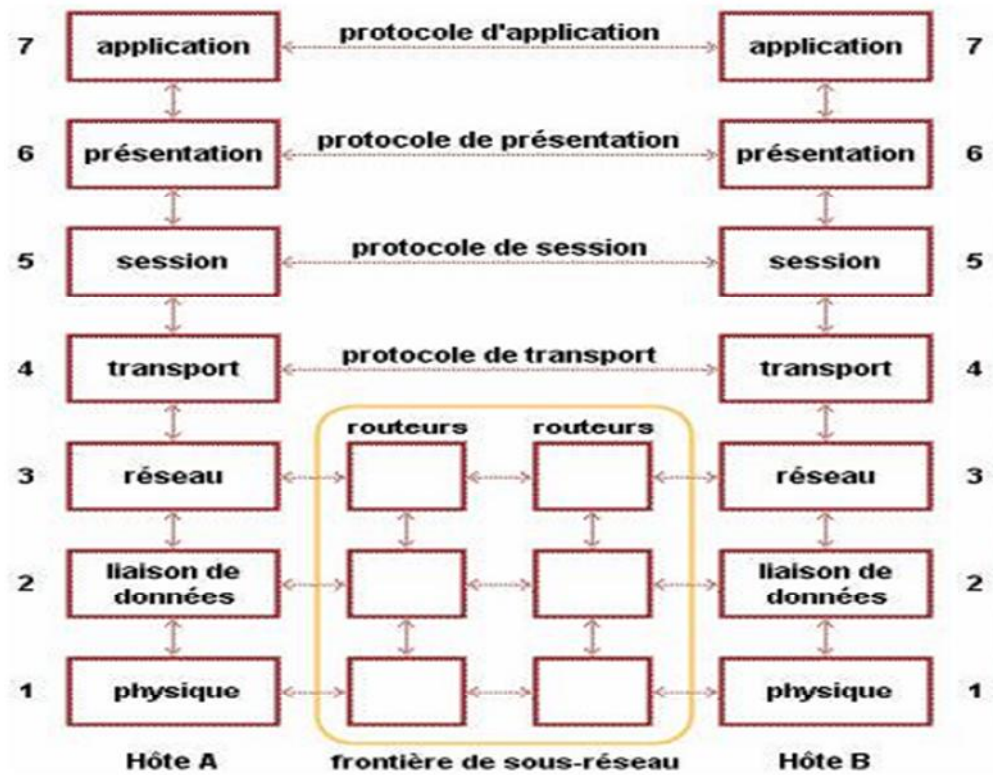


Figure II.7 : Le modèle OSI.

Les sept couches du modèle OSI peuvent être divisées en deux catégories :

- Les couches supérieures (5, 6,7) : s’occupant de tout ce qui est application (traitement).
- Les couches inférieures (1, 2, 3,4) : prennent en charge le transport des données.

Chacune d’elle possède ses propres fonctions :

II.4.1.1. Couche physique (1) :

Gère les connexions matérielles, assure la transmission fiable du bit sur un canal de transmission (ça peut-être un câble coaxial, paires sur RJ45, onde radio, fibre optique, ...) et définit la façon dont les données sont converties en signaux numériques.

II.4.1.2. Couche liaison de données (2) :

Le rôle principal de cette couche est de fournir à la couche supérieure (couche réseau) un moyen de communication fiable sans erreurs de transmission, Elle assure la synchronisation de la transmission et contrôle d'accès au réseau.

II.4.1.3. Couche réseau (3) :

Le rôle essentiel de cette couche est de déterminer la manière dont les paquets sont routés de la source à la destination (ROUTAGE).

II.4.1.4. Couche transport (4) :

Le rôle principal de cette couche est d'accepter des données de la couche supérieure, de les découper en paquets si nécessaire, de les transmettre à la couche réseau, et d'assurer qu'elles arrivent correctement à destination.

II.4.1.5. Couche session (5) :

Elle permet de créer, d'utiliser, d'achever une connexion entre deux ordinateurs. Contrôle le dialogue.

II.4.1.6. Couche présentation(6) :

Elle s'occupe de la mise en forme des données éventuellement de l'encryptage, et de la compression de ces dernières, par exemple mise en forme des textes, images et vidéos.

II.4.1.7. Couche application (7) :

Cette couche gère l'accès des applications aux services réseaux, elle permet de gérer les communications directement entre les programmes et logiciels via différents mécanismes.

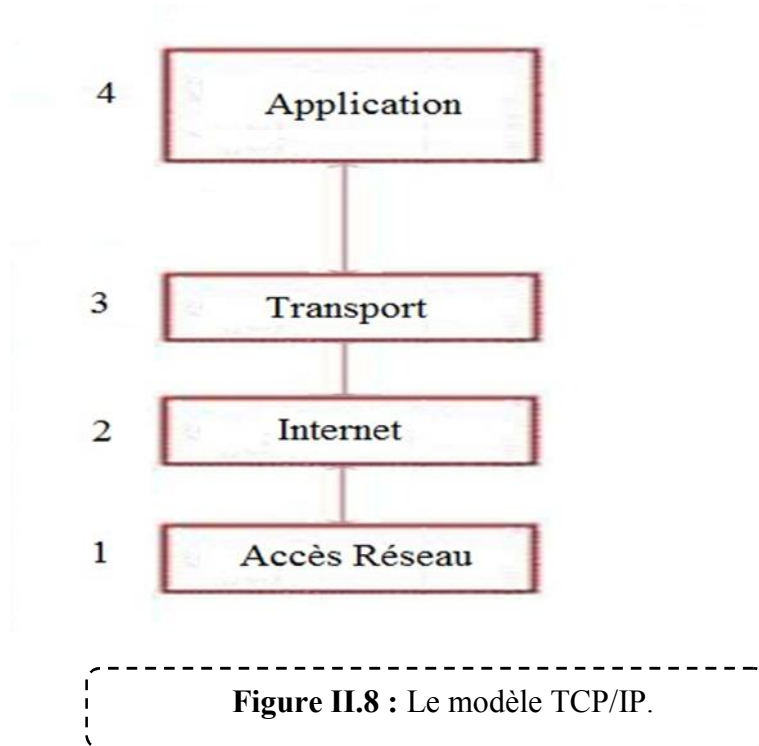
II.4.2. Le modèle TCP/IP : [9]

Le but principal de la conception de TCP/IP est de permettre une interconnexion entre des réseaux d'ordinateurs indépendants pour assurer des services de communication universelles formant ainsi inter network ou internet qui du point de vue de l'utilisateur est un seul réseau, il est inspiré du modèle OSI.

Il reprend l'approche modulaire (utilisation des couches) mais en contient uniquement quatre. C'est actuellement le modèle le plus utilisé. A chaque niveau, le paquet de données change d'aspect, car on lui ajoute un en-tête, ainsi les appellations changent suivant les couches :

- Le paquet de données est appelé message au niveau de la couche application.

- Le message est ensuite encapsulé sous forme de segment dans la couche transport. Il est donc découpé en morceaux avant l'envoi.
- Le segment une fois encapsulé dans la couche internet prend le nom de datagramme.
- Enfin on parle de trame au niveau de la couche accès au réseau.



II.4.2.1. La couche hôte réseau (1) :

Cette couche est assez "étrange". En effet, elle semble "regrouper" les couches physique et liaison de données du modèle OSI. En fait, cette couche n'a pas vraiment été spécifiée ; la seule contrainte de cette couche, c'est de permettre un hôte d'envoyer des paquets IP sur le réseau. L'implémentation de cette couche est laissée libre.

II.4.2.2. La couche internet (2) :

Cette couche est la clé de voûte de l'architecture. Cette couche réalise l'interconnexion des réseaux (hétérogènes) distants sans connexion. Son rôle est de permettre l'injection de paquets dans n'importe quel réseau et l'acheminement de ces paquets indépendamment les uns des autres jusqu'à destination. Comme aucune connexion n'est établie au préalable, les paquets peuvent arriver dans le désordre ; le contrôle de l'ordre de remise est éventuellement la tâche des couches supérieures.

La couche internet possède une implémentation officielle : le protocole IP (Internet Protocol).

II.4.2.3. La couche transport (3) :

Son rôle est le même que celui de la couche transport du modèle OSI : permettre à des entités paires de soutenir une conversation.

II.4.2.4. La couche application (4) :

Cette couche contient tous les protocoles de haut niveau, comme par exemple Telnet, TFTP (trivial File Transfer Protocol), SMTP (Simple Mail Transfer Protocol), HTTP (HyperText Transfer Protocol). Le point important pour cette couche est le choix du protocole de transport à utiliser.

II.5. Conclusion :

Dans le présent chapitre, nous avons présenté le modèle client/serveur, qui est un environnement que nous allons utiliser pour la réalisation d'une application de gestion des installations propane au sein de l'entreprise NAFTAL.

A ce stade, l'étape analyse et conception du système peut être lancée. Elle sera l'objet du chapitre qui suit.

CHAPITRE

Analyse & Conception

III

III.1.Introduction :

La modélisation est la base de toute activité qui de développement de logiciels de qualité. Les modèles sont construits pour spécifier la structure et le comportement attendu d'un système, pour visualiser et contrôler son architecture et pour mieux comprendre son organisation.

Dans ce chapitre nous allons entamer le développement de notre application par une analyse qui mettra en évidence les différents acteurs intervenant 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.

De nombreux efforts 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 (Unified Modelling language) qui permet de bien représenter la dynamique d'une application par la série des diagrammes qu'il offre.

III.2.Analyse :

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

III.2.1.Identification des besoins :

C'est une étape primordiale au début de chaque démarche de développement. Son but est de veiller à développer une application adéquate, sa finalité est la description générale des fonctionnalités du système, en répondant à la question : Que doit faire notre système?

Notre système doit répondre aux exigences suivantes :

1. Sécurisation de l'application.
2. Le système doit pouvoir éditer :
 - les demandes d'installation : document établi par le chargé d'étude du service distribution.
 - les ordres de travaux(OT) : document établi après réception d'une demande d'installation, pour ordonner l'exécution des travaux.
 - Les procès verbaux (PV): rapport des travaux effectués lors de l'intervention.
 - les Factures : document indiquant la nature, la quantité et le prix de la marchandise vendue.
3. Consulter les demandes d'installation, ordre de travaux, PV et facture.
4. Impression des divers documents : demande d'installation, OT, PV et facture.
5. Enregistrer les informations à propos des interventions et toutes autres entités, pour mettre à jour la base de données.
6. La suppression et la modification de certaines entités.
7. Gérer la base de données.

Dans l'étape d'analyse, les besoins du client doivent être recensés, formalisés ainsi que l'ensemble des contraintes, puis l'estimation de la faisabilité de ces besoins.

III.2.2. Identification des acteurs :

Définition [10]: Un acteur représente un rôle que peut jouer l'utilisateur dans le système.

L'acteur est associé à un cas d'utilisation, c'est-à-dire qu'il peut interagir avec lui et participer à son scénario, il est représenté par un personnage stylisé.

Pour notre système, on peut distinguer un seul acteur principal qui est le cadre technique.

III.2.3. Identification des cas d'utilisation :

Définition [10]: Un cas d'utilisation représente un ensemble de séquences qui sont réalisées par le système en réponse à une action d'un acteur et qui produisent un résultat observable. Les cas d'utilisation décrivent ce que le système devra faire sans spécifier comment le faire.

Le tableau suivant récapitule les cas d'utilisation de chaque acteur de notre application :

Acteur	Tâches
Chargé d'étude	T01 :S'authentifier. T02 : Accéder au tableau de bord T04 : Gérer les demandes d'installation P35 (Ajouter, consulter) T05 : Gérer les demandes d'installation citerne (Ajouter, consulter) T06 : Gérer les clients P35 et les clients citerne (Ajouter, Modifier, Supprimer) T07 : Gérer les clients citerne (Ajouter, Modifier, Supprimer) T08 : Gérer les facteurs (Ajouter, Consulter) T09 : Suivi des installations citerne, P35, P11 T10 : Impression des documents (Factures, liste des clients par daïra, ..) T12 : Gérer le compte (Modifier le mot de passe, Déconnection)

<p>Service MIF(chef de section réalisation installation)</p>	<p>T01 :S'authentifier. T02 : Accéder au tableau de bord T13 : Gérer les ordres de travaux P35et les ordres de travaux citerne (Ajouter, modifier, consulter) T14 : Consultation des demandes d'installation citerne T15 : Gestion des PV (Ajouter, consulter) T16 : Impression des documents (ordre de travaux,...) T12 : Gérer le compte (Modifier le mot de passe, Déconnection)</p>
---	---

Tableau III.1 : identification des cas d'utilisation

III.3.Conception :

Le processus de conception de notre projet se caractérise par deux niveaux : le niveau applicatif et le niveau donné.

Le niveau applicatif s'appuie essentiellement sur quelques diagrammes de l'extension du langage de modélisation UML pour les applications web. A cet effet nous avons adopté la démarche suivante :

- Après l'identification des différents acteurs ainsi que les cas d'utilisation qui sont mis en œuvre par ces acteurs, le diagramme de cas d'utilisation est élaboré.
- Chaque cas d'utilisation se traduit par un ou plusieurs scénarios. Chaque scénario fait l'objet d'une description sous forme graphique à l'aide d'un diagramme de séquence et un digramme d'activité.
- Une identification des classes est fournie par la synthèse des diagrammes de séquence, ainsi le diagramme de classe sera élaboré.

Le niveau donné concerne l'organisation conceptuelle, logique et physique des données manipulées. Durant la partie analyse nous avons pu identifier les données nécessaires et indispensables au bon fonctionnement de l'application et à travers la conception du niveau applicatif nous allons dégager les classes significatives, dès lors on peut élaborer la conception de la base de données.

La figure suivante donne la représentation graphique de la démarche de modélisation adoptée pour concevoir l'application:

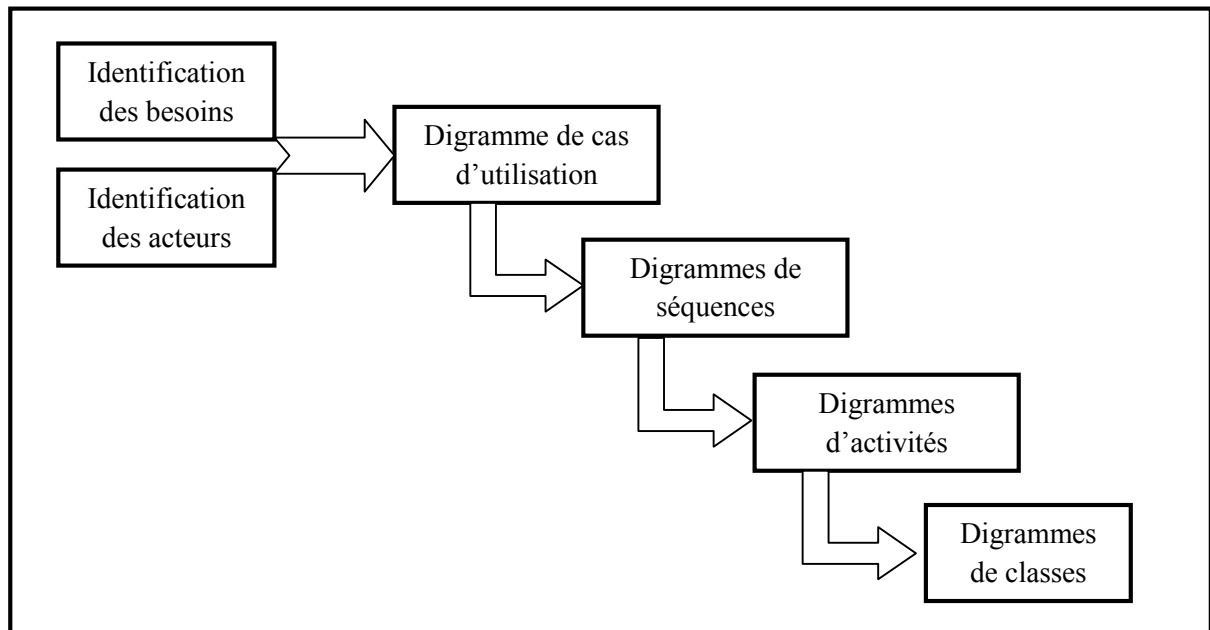


Figure III.1 : La démarche adoptée pour la modélisation

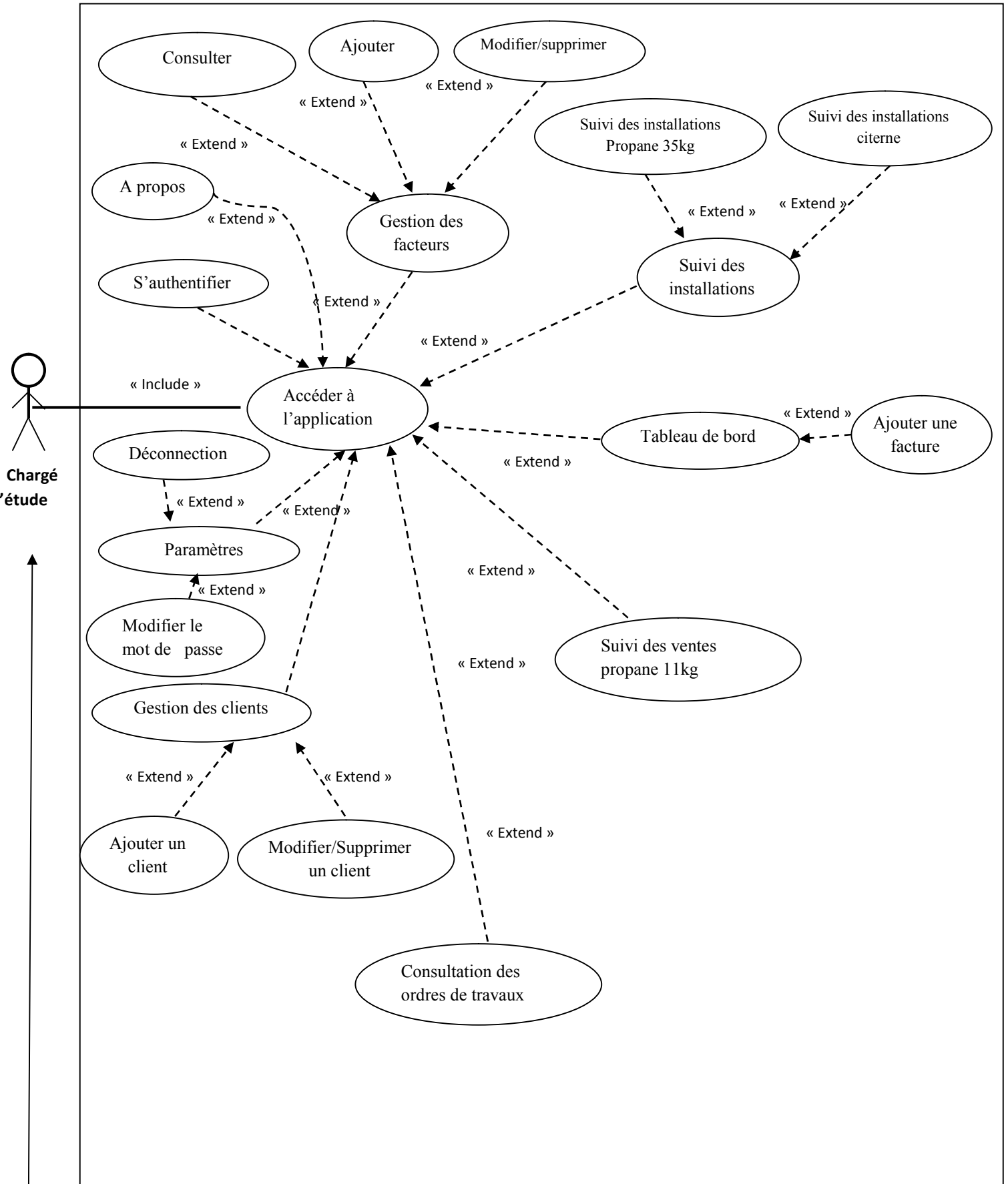
III.3.1. Le niveau applicatif :

III.3.1.1. Le diagramme des cas d'utilisation :

Lors de la phase d'analyse nous avons pu identifier les acteurs ainsi que les cas d'utilisation associés à ces derniers. Ce qui nous donne l'opportunité d'élaborer le diagramme des cas d'utilisation.

Définition [15]: Les diagrammes de cas d'utilisation permettent de représenter un ensemble de cas d'utilisation, d'acteurs et leurs relations. Ils présentent la vue statique des cas d'utilisation d'un système et sont particulièrement importants dans l'organisation et la modélisation des comportements d'un système.

- **La relation d'inclusion (include) [11]:** Elle indique que le cas d'utilisation source contient aussi le comportement décrit dans le cas d'utilisation destination. Cette relation permet de décomposer des comportements et de définir les comportements partageables entre plusieurs cas d'utilisations.
- **La relation d'extension (Extend) [11]:** Elle indique que le cas d'utilisation source ajoute son comportement au cas d'utilisation destination. L'extension peut être soumise à des conditions.



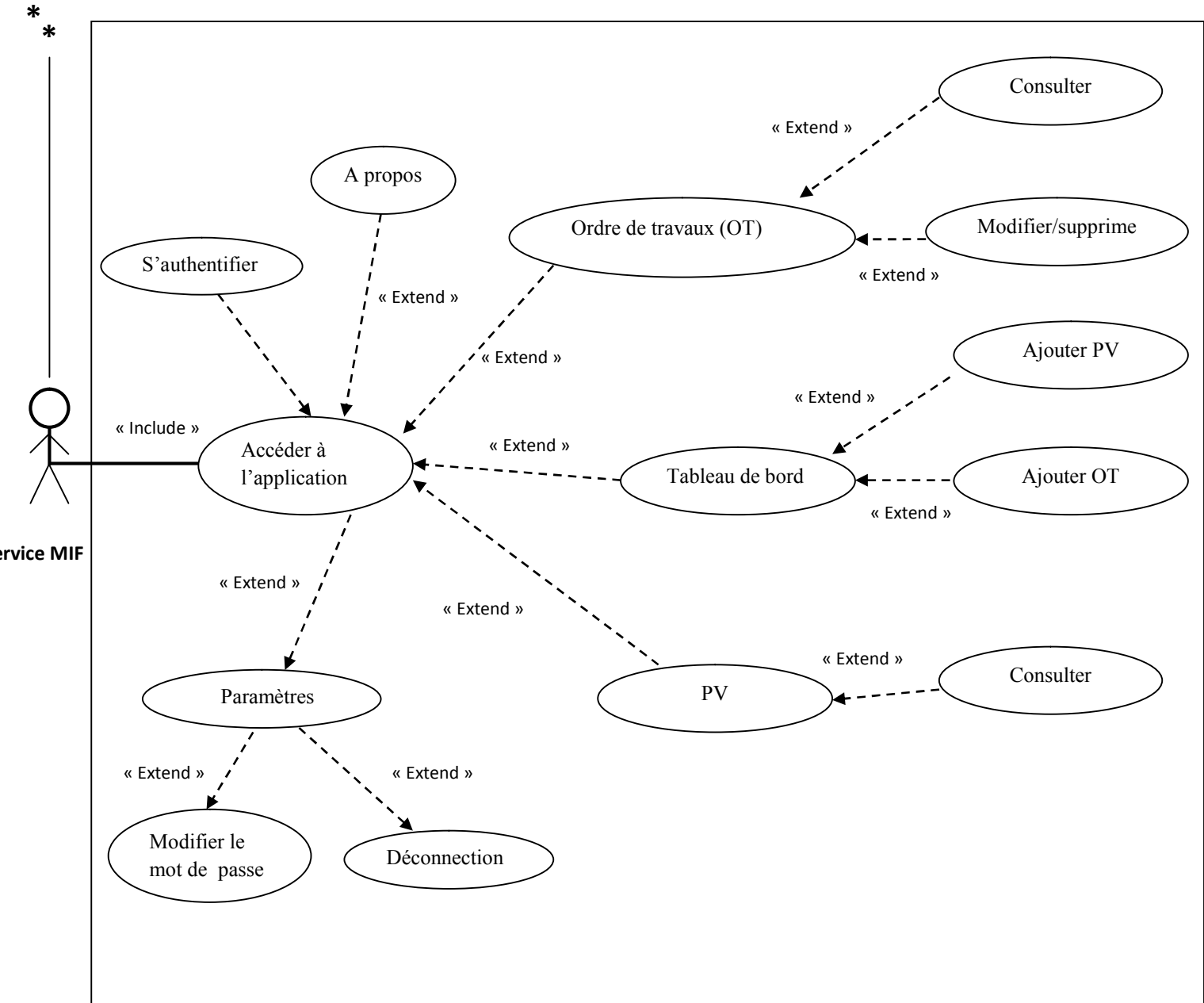


Figure III.2 : Le diagramme des cas d'utilisation

III.3.1.2. Description des cas d'utilisation avec des scénarios :

Un scénario est une requête de cas d'utilisation, tout résultat possible de l'exécution d'un cas d'utilisation est appelé scénario.

Les scénarios doivent être définis par les utilisateurs eux même, chacune des tâches effectuées par un ou plusieurs acteurs sera décrite par un ensemble des scénarios.

Voici quelques scénarios pour quelques cas d'utilisations :

Cas utilisation : S'authentifier.

Acteur : Chargé d'étude.

Résumé : Ce cas d'utilisation permet à l'utilisateur de s'authentifier et d'accéder à la page principale.

Enchaînement :

Ce cas d'utilisation commence lorsque l'utilisateur ouvre l'application

Scénario nominal :

- 1- L'utilisateur saisit son mot de passe et son login puis clique sur le bouton « Continuer ».
- 2- Le système vérifie la conformité du mot de passe et du login.
- 3- Le système affiche la page principale de l'application.

Scénario alternatif : Le mot de passe saisi ou le login est incorrect.

L'enchaînement commence au point 1 du scénario nominal.

- 3- Le système affiche un message d'erreur sur le formulaire d'authentification demandant à l'utilisateur de ressaisir le mot de passe et le login.

Figure III.3 : Description du cas d'utilisation « S'authentifier »

Cas utilisation : Ajouter une demande d'installation.

Acteur : Chargé d'étude.

Résumé : Ce cas d'utilisation permet au cadre technique d'ajouter une demande d'installation.

Enchaînement :

Ce cas d'utilisation commence lorsque l'utilisateur clique sur le lien « Ajouter demande d'installation ».

Scénario nominal :

- 1- Le système affiche la liste des clients.
- 2- L'utilisateur choisit un client puis clique sur valider.
- 3- Le système affiche le formulaire de renseignement.
- 4- L'utilisateur remplit le formulaire puis clique sur le bouton « Enregistrer ».
- 5- Le système vérifie la validité des données saisies.
- 6- Le système crée une nouvelle demande puis affiche un message de confirmation.

Scénario alternatif : L'utilisateur a oublié de remplir un champ obligatoire.

L'enchaînement commence au point 1 du scénario nominal.

Figure III.4 : Description du cas d'utilisation « Ajouter une demande d'installation »

Cas utilisation : Modifier un ordre de travaux

Acteur : Chef de section installation et réalisation.

Résumé : Ce cas d'utilisation permet au chef de section installation et réalisation de modifier un ordre de travaux dans l'application.

Enchaînement :

Ce cas d'utilisation commence lorsque l'utilisateur clique sur le lien « Modifier/consulter ».

Scénario nominal :

- 1- Le système affiche le formulaire de l'ordre de travaux déjà établie.
- 2- L'utilisateur modifie les données du formulaire puis clique sur le bouton « Enregistrer ».
- 3- Le système vérifie la validité des données ressaisies.
- 4- Le système enregistre le nouvel ordre de travaux puis affiche un message de confirmation.

Scénario alternatif : L'utilisateur a oublié de remplir un champ obligatoire.

L'enchaînement commence au point 1 du scénario nominal.

Figure III.5 : Description du cas d'utilisation « Modifier un ordre de travaux »

Cas utilisation : Supprimer un client

Acteur : Chargé d'étude.

Résumé : Ce cas d'utilisation permet à un utilisateur de supprimer client.

Enchaînement :

Ce cas d'utilisation commence lorsque l'utilisateur clique sur le lien « modifier/Supprimer un client ».

Scénario nominal :

- 1- Le système affiche un la liste de tous les clients.
- 2- Le cadre technique sélectionne un client en cliquant sur le lien « supprimer ».
- 3- Le système affiche un message de confirmation de suppression.

Figure III.6 : Description du cas d'utilisation « Supprimer un client »

III.3.1.3. Les diagrammes de séquences et d'activités :

Après avoir décrit les cas d'utilisation avec des scénarios, nous allons traduire ces derniers en diagrammes de séquences et en diagrammes d'activités.

- **Un diagramme de séquences [12]:** indique l'interaction entre plusieurs partenaires de communication, également appelés lignes de vie. Les principales informations contenues dans les diagrammes de séquences sont les messages échangés entre les lignes de vie. Un diagramme de séquences met toujours l'accent sur l'ordre chronologique des messages.
- **Un diagramme d'activités [12]:** apporte un point de vue complémentaire à l'aspect dynamique de la modélisation. Il offre un pouvoir d'expression très proche des langages de programmation objets. Il est donc bien adapté à la spécification détaillée des traitements en phase de réalisation. Un diagramme d'activités se concentre plutôt sur les activités entre les objets, c'est-à-dire, il met en évidence l'activité qui a lieu dans le temps, donc les opérations transmises entre les objets

➤ Diagramme de séquence du cas d'utilisation « Authentification »

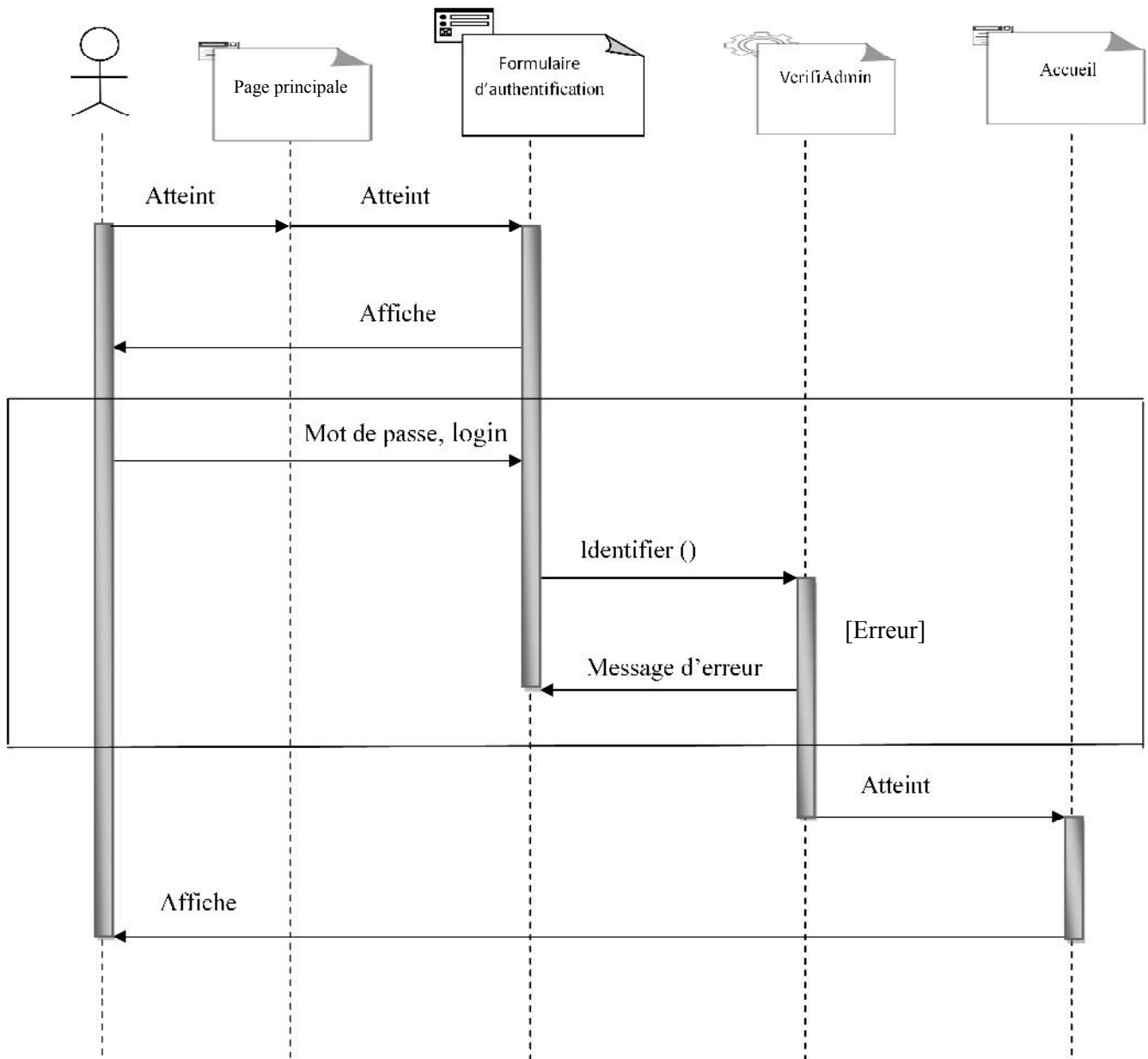


Figure III.7: Diagramme de séquence du cas d'utilisation « S'authentifier »

➤ **Diagramme d'activité de l'authentification :**

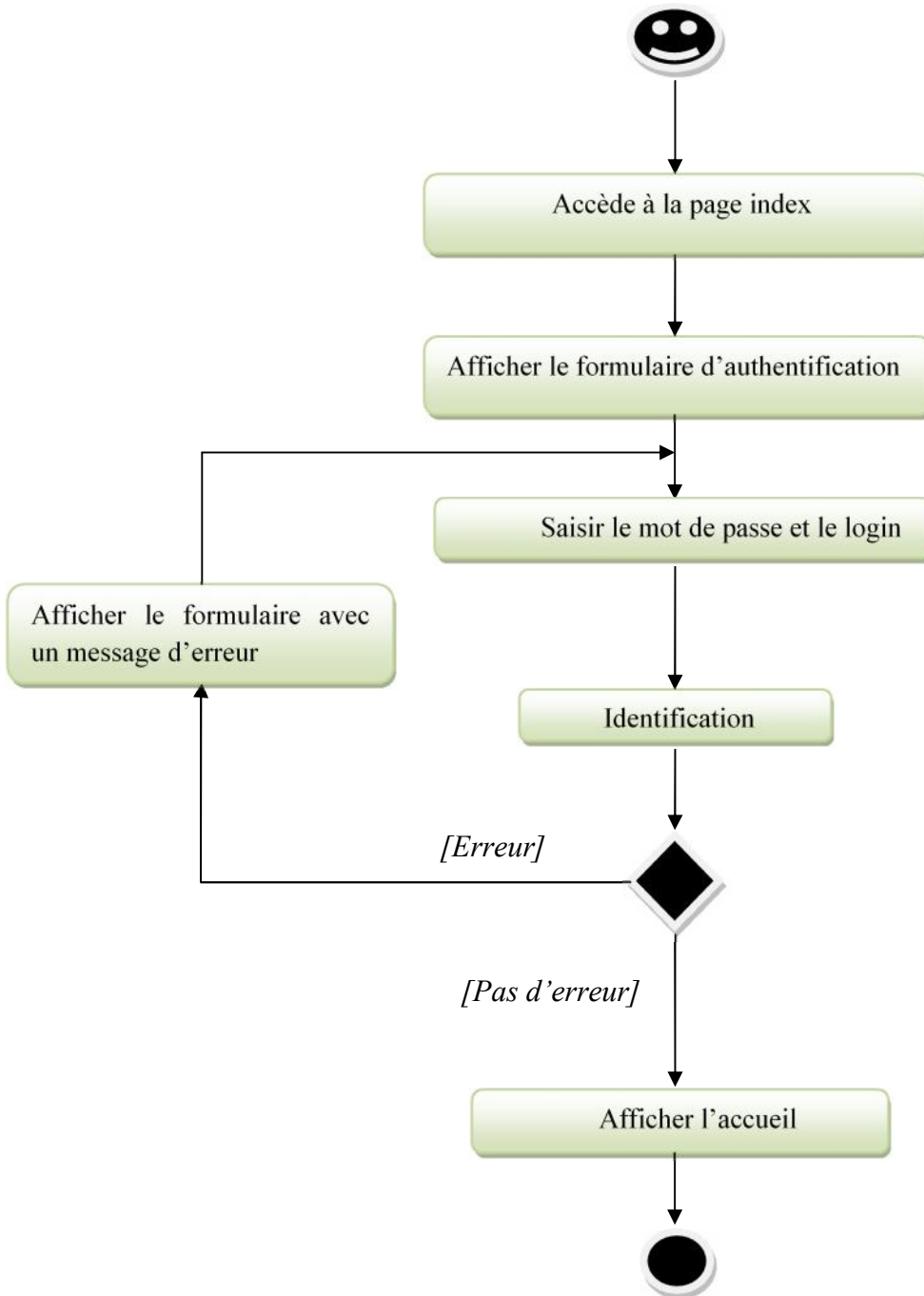


Figure III.8 : Diagramme d'activités du cas d'utilisation « S'authentifier »

➤ Diagramme de séquence «Ajouter un client» :

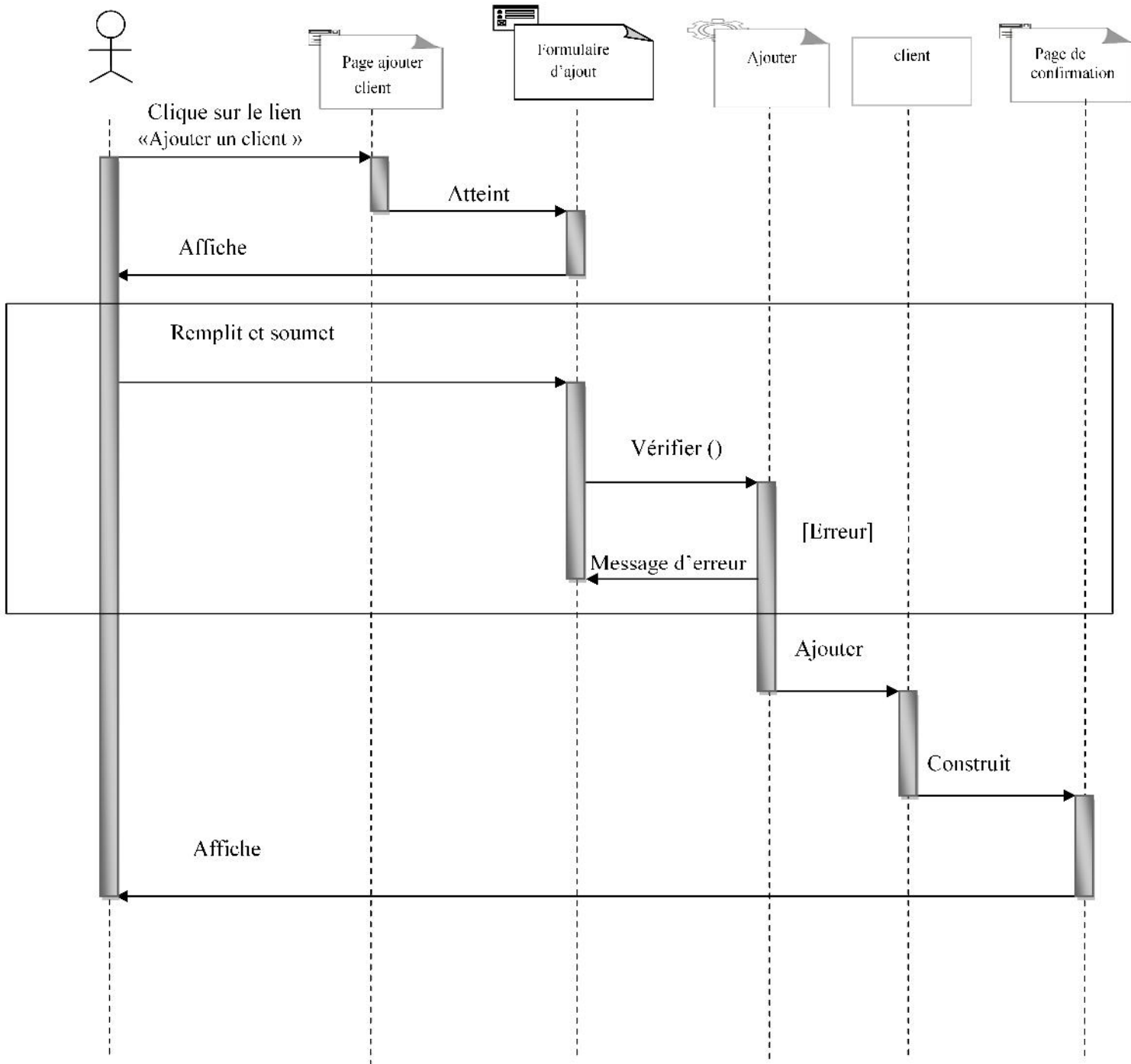


Figure III.9 : Diagramme de séquence du cas d'utilisation « Ajouter un client »

➤ **Diagramme d'activité d'ajout d'un client :**

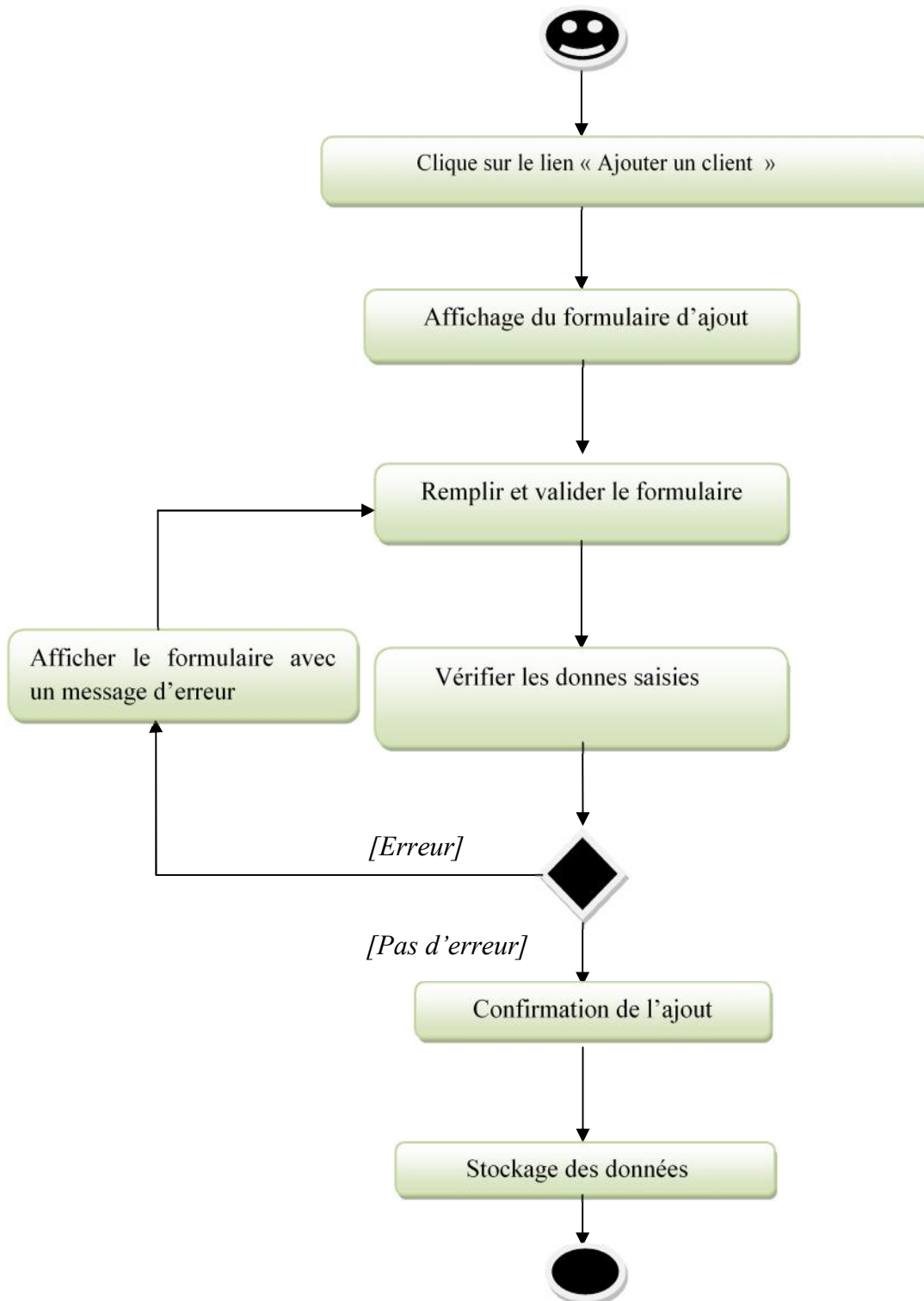


Figure III.10 : Diagramme d'activités du cas d'utilisation « Ajout un client »

➤ Diagramme de séquence « cas Modifier le mot de passe » :

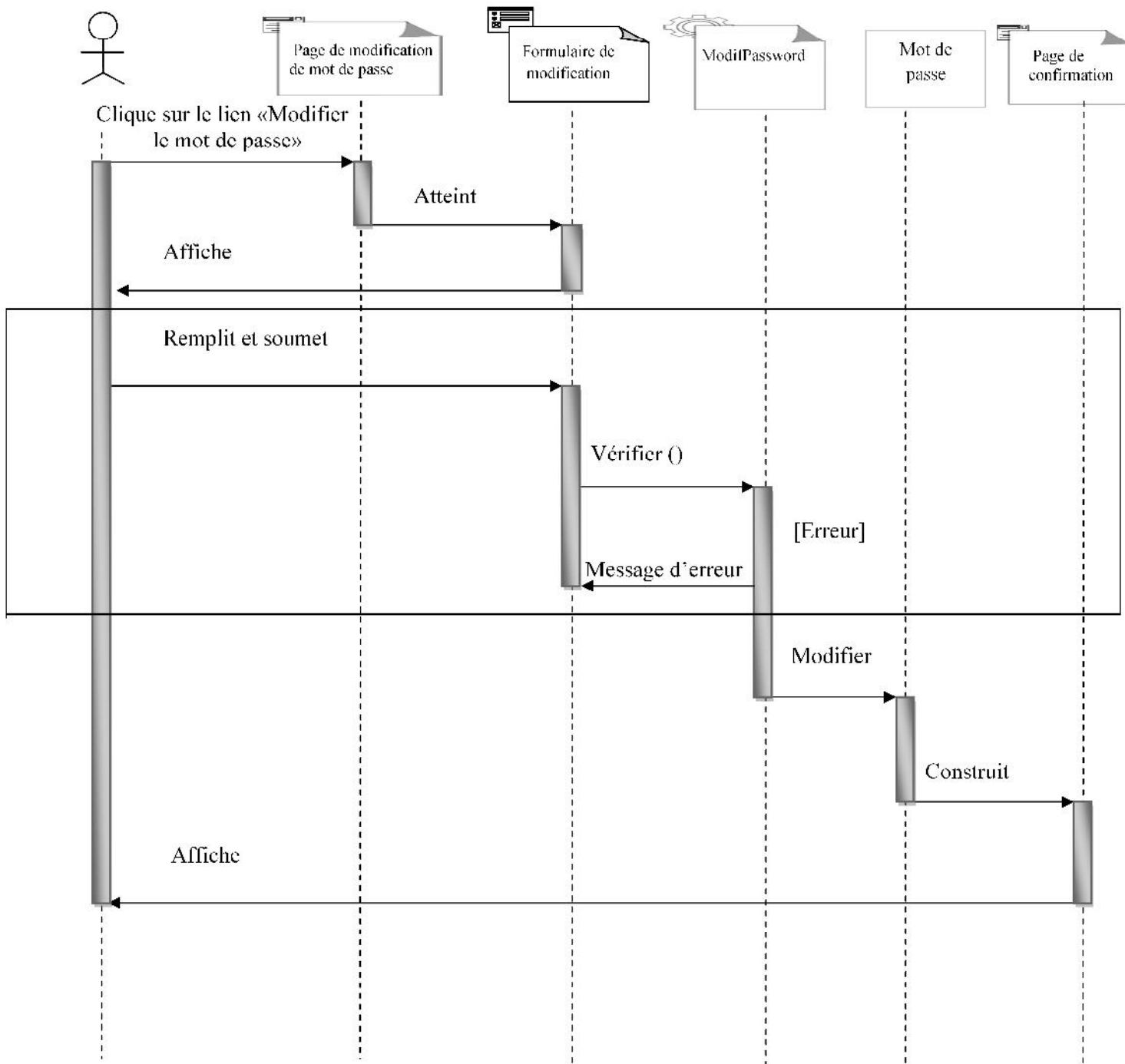


Figure III.11: Diagramme de séquence du cas d'utilisation « Modifier le mot de passe»

➤ Diagramme d'activité de modification de mot de passe :

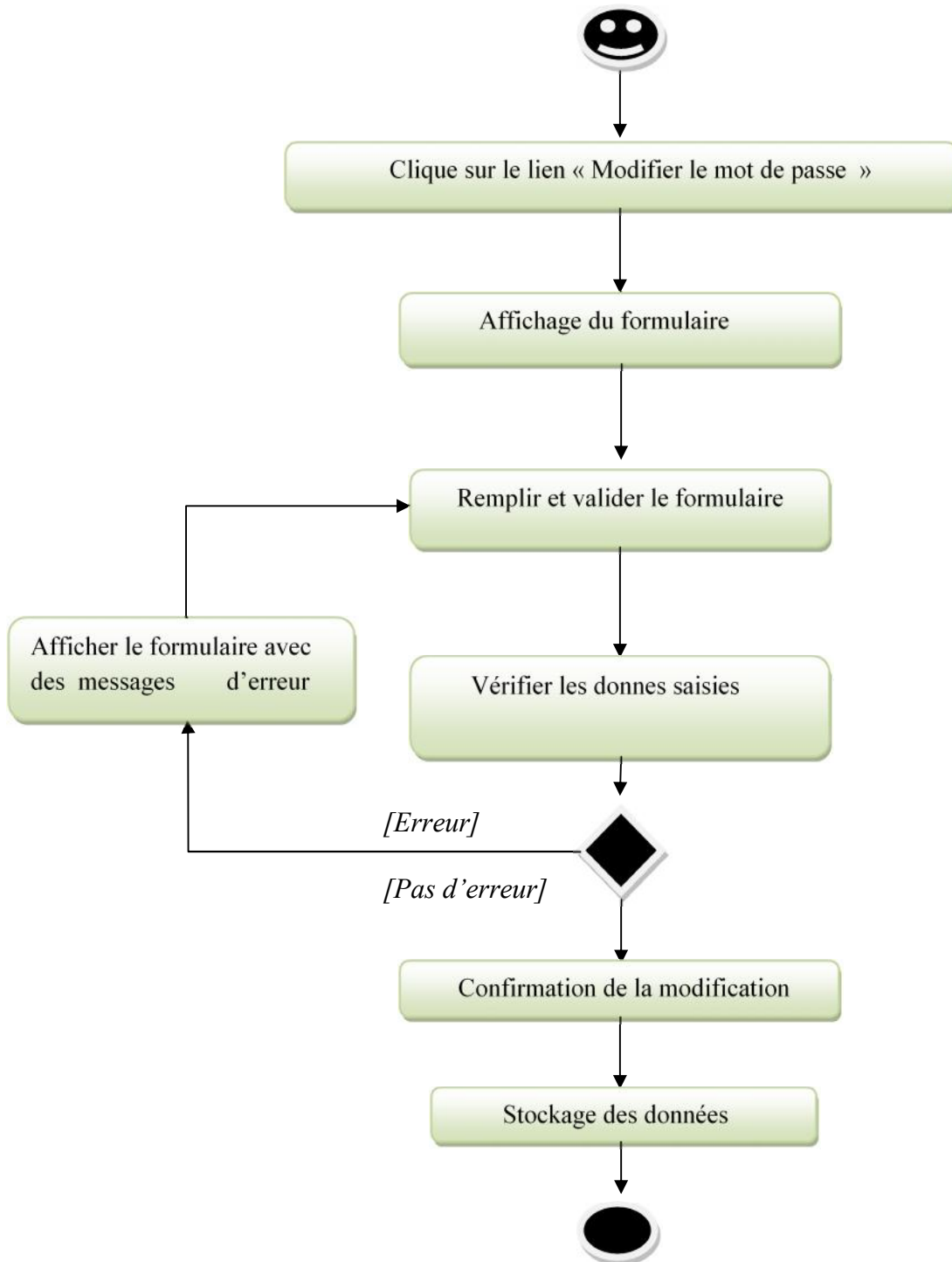


Figure III.12 : Diagramme d'activités du cas d'utilisation « Modifier le mot de passe »

III.3.1.4. Les diagrammes de classes :

Dans cette partie, nous allons passer à la modélisation de l'aspect statique de notre application, c'est-à-dire nous allons modéliser l'intérieur de notre système. Pour ce faire, nous allons utiliser les diagrammes de classes.

Le diagramme de classe est considéré comme le plus important de la modélisation orientée objet. Alors que les diagrammes précédents (diagrammes de séquences et diagrammes d'activités) montrent le système du point de vue dynamique, le diagramme de classe en montre la structure interne. Il contient principalement des classes reliées par des associations et chaque classe contient des attributs et des opérations.

Après avoir élaboré les diagrammes de séquences et les diagrammes d'activités pour les cas d'utilisation déjà décrit, nous allons élaborer des diagrammes de classes généraux puis détaillés pour chacun d'eux.

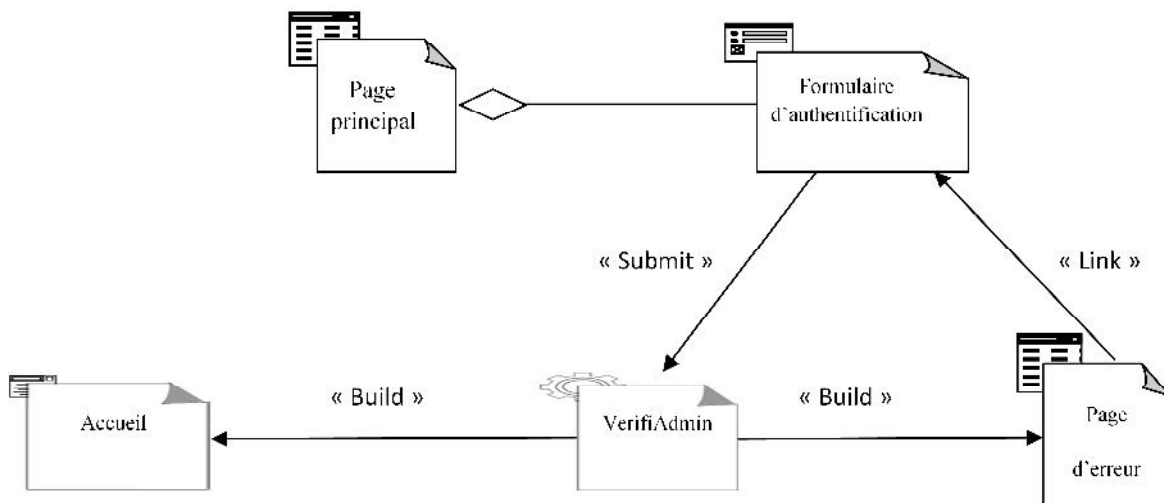


Figure III.13 : Diagramme de classes général du cas d'utilisation « S'authentifier »

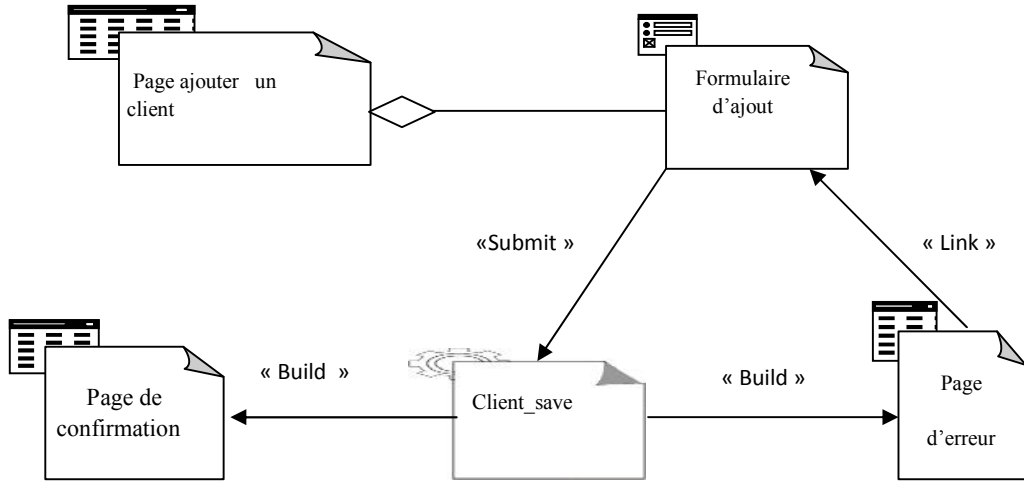


Figure III.14 : Diagramme de classes général du cas d'utilisation « Ajouter un client »

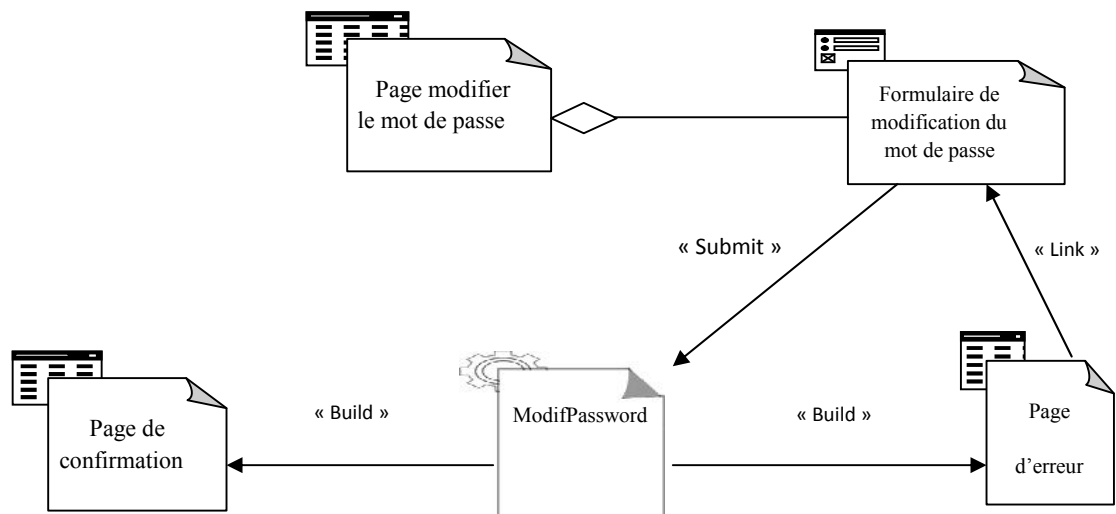


Figure III.15 : Diagramme de classes général du cas d'utilisation « Modifier le mot de passe »

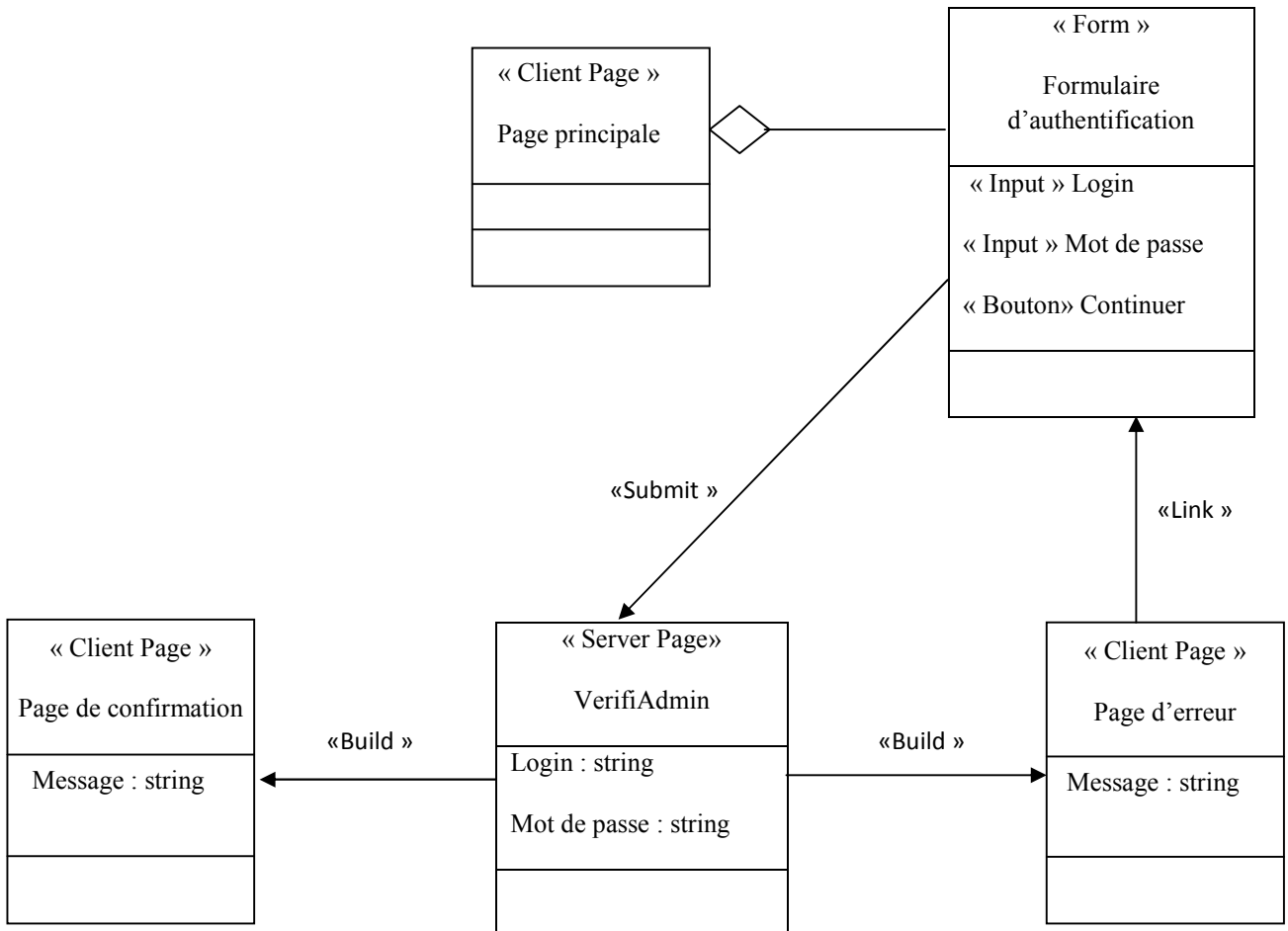


Figure III.16 : Diagramme de classes détaillé du cas d'utilisation « S'authentifier »

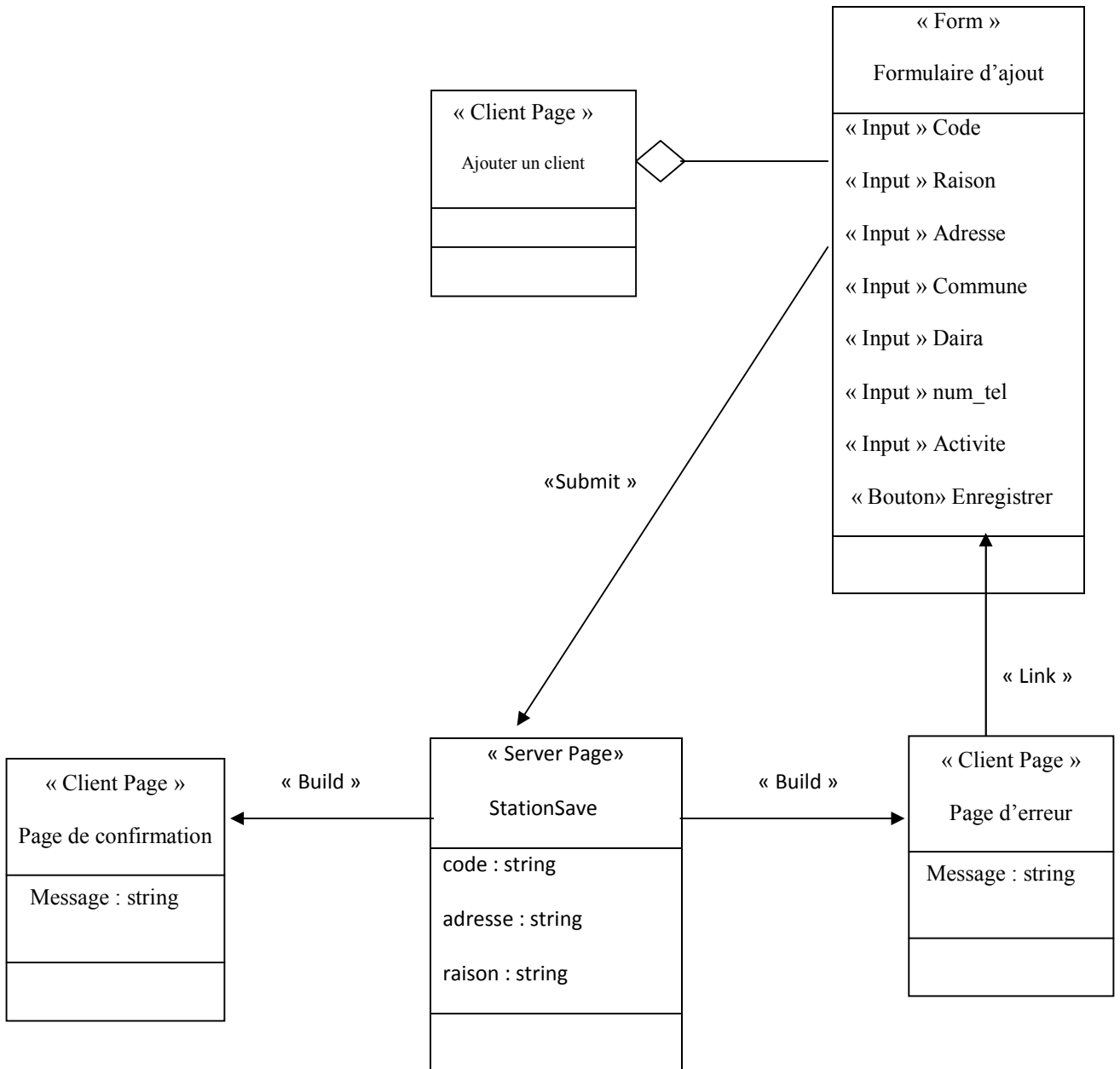


Figure III.17 : Diagramme de classes détaillé du cas d'utilisation « Ajouter un client »

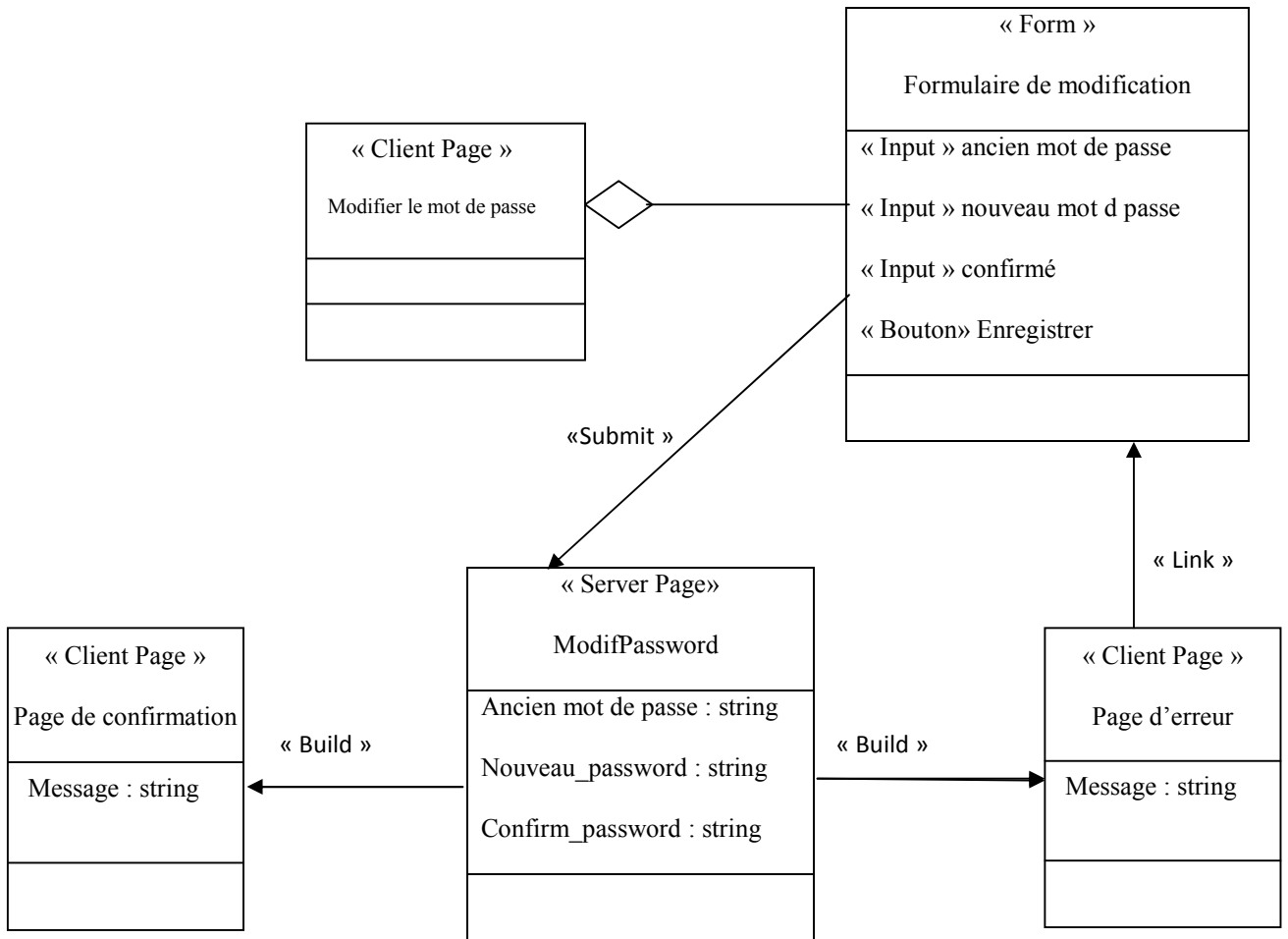


Figure III.18 : Diagramme de classes détaillé du cas d'utilisation « Modifier le mot de passe »

III.3.2. Le niveau donné :

III.3.2.1. Le diagramme de classes général :

Le schéma conceptuel de la base de données est représenté par le diagramme de classe général suivant :

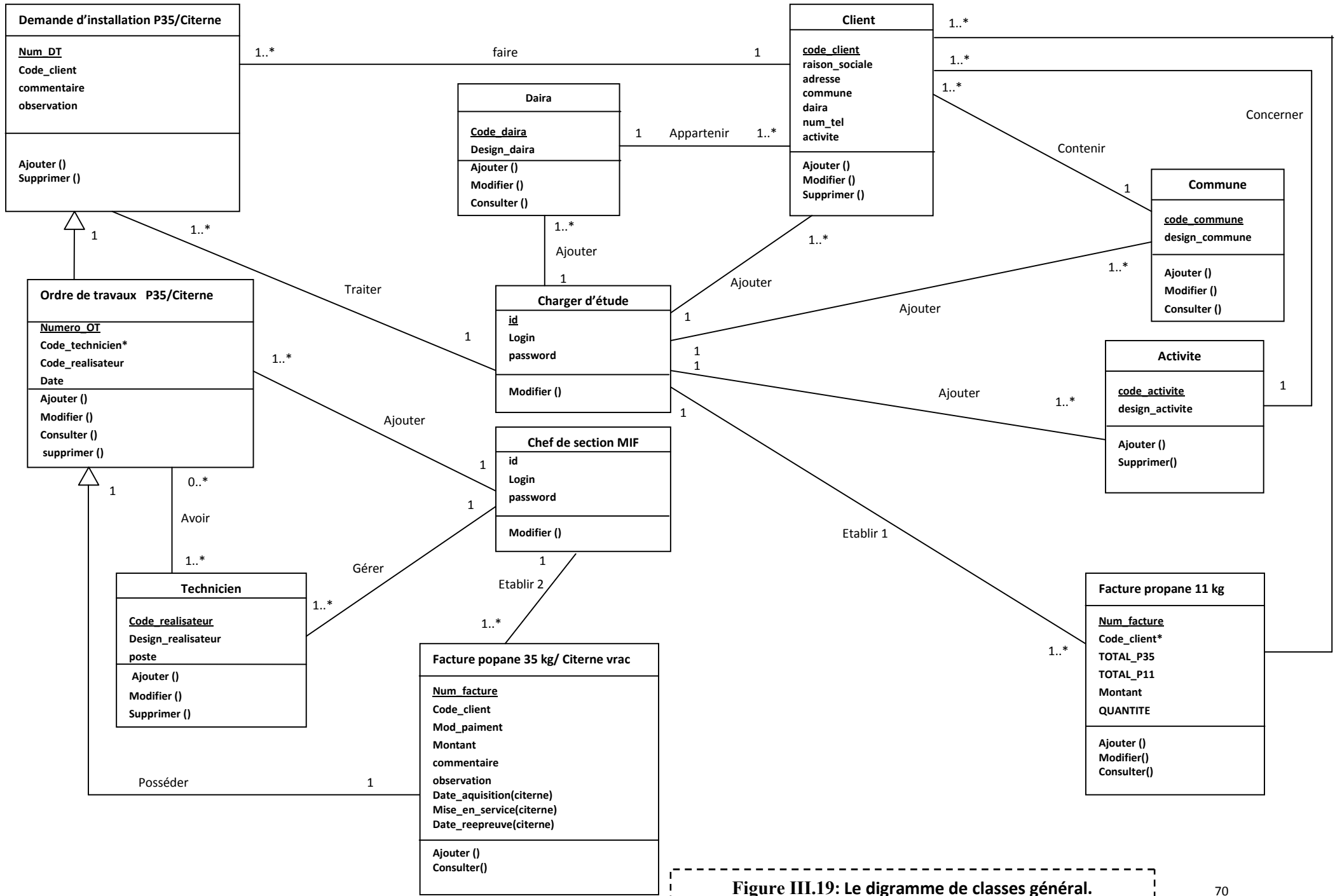


Figure III.19: Le digramme de classes général.

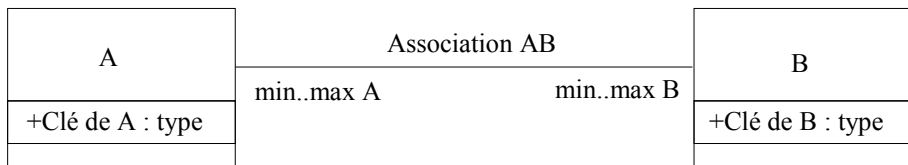
III.3.2.2. Le modèle logique : [1 3]

Afin de d’assurer la cohérence de la base de données lors du passage du modèle conceptuel (le digramme de classes) au modèle logique, un certain nombre de règles doivent être respectées. Ces règles sont les suivantes :

A- Les classes : La règle est assez simple, « chaque classe devient une relation ». Chaque attribut de la classe devient un attribut de la relation. Concernant la clé, soit il existe une clé pour la classe qui deviendra la clé de la relation, ou bien, on ajoute une clé artificielle à la relation.

B- Traduire les associations : L’objectif est de mémoriser les liens entre les objets des classes reliées par l’association. La solution à cette mémorisation dépend de la cardinalité de l’association, plus exactement de ses maximums.

Soient les deux classes A et B suivantes, qui sont reliées par une association AB :



Le tableau suivant examine les cas possibles concernant le maximum des cardinalités de A et de B :

MaxA \ MaxB	1	>1
1	Ajouter la clé de B dans la relation de A comme attribut ou Ajouter la clé de A dans la relation de B comme attribut.	Ajouter la clé de B dans la relation de A comme attribut.
>1	Ajouter la clé de A dans la relation de B comme attribut.	Créer une relation AB ayant comme attribut la clé de A et la clé de B.

En se basant sur les règles citées précédemment pour le passage du modèle conceptuel au modèle logique, nous obtenons les relations suivantes :

client (code_client, id_activite*,raison_sociale,adresse,commune,daira,num_tel) ;

activite (id_activite,design_activite) ;

citerne (code_citerne,designation_citerne,montant) ;

facture_p11(code_commande,code_client* ,commune,total_p35,total_p11,quantite) ;

realisateur(code_realisateur,design_realisateur) ;

demande_installation_citerne/p35(num_demande,code_client*,observation) ;

ot_citerne/p35(num_ot,num_demande*,code_realisateur* ,date_ot) ;

user(id,Login,password) ;

commande_p35(code_commande,num_ot*,total_p35,capacite ,realisation,mise_en_service, observation)

pv_citerne(code_pv,num_ot*,code_realisateur*,code_citerne*,mise_service,date_repreuve, date_acquisition, observation)

pvp35(code_pv,num_ot*,code_realisateur*,capacite,total_p35,mise_service,date_repreuve,d ate_acquisition, observation)

facture p11(code_facture , code_client*,paiement ,total_p35,total_p11,quantite,montant)

facture citerne/p35(code_facture,code_commande* , paiement ,date_installation,montant)

repreuve(num_demande,code_client*,observation) .

___ : Un attribut souligné signifie qu'il s'agit d'une clé primaire.

* : Un attribut suivi d'une étoile signifie qu'il s'agit d'une clé étrangère.

III.4.Conclusion :

Ce chapitre a été consacré à l'analyse et à la conception de l'application avec le langage UML.

Nous avons présenté le processus de conception de notre application en deux niveaux, le niveau applicatif et le niveau de données. En premier lieu, nous avons commencé l'analyse et la conception par le niveau applicatif qui concerne les fonctionnalités et les traitements de l'application, ensuite nous sommes passés au niveau données qui nous a permis d'avoir le modèle logique de la base de données.

Le chapitre suivant sera donc la traduction de cette conception en un ensemble de scripts, en ce qui concerne partie applicative, et en une base de données physique pour la partie données.

CHAPITRE

**Réalisation & Mise en
œuvre**

IV.1. Introduction :

Dans ce chapitre, consacré à la réalisation et la mise en œuvre de notre application de gestion de maintenance, nous allons présenter les outils de développement adoptés dans notre application ainsi nous présenterons aussi certaines interfaces développées pour illustrer notre application.

IV.2. Choix d’outils technologiques :

La figure suivante décrit notre choix pour les outils technologiques :

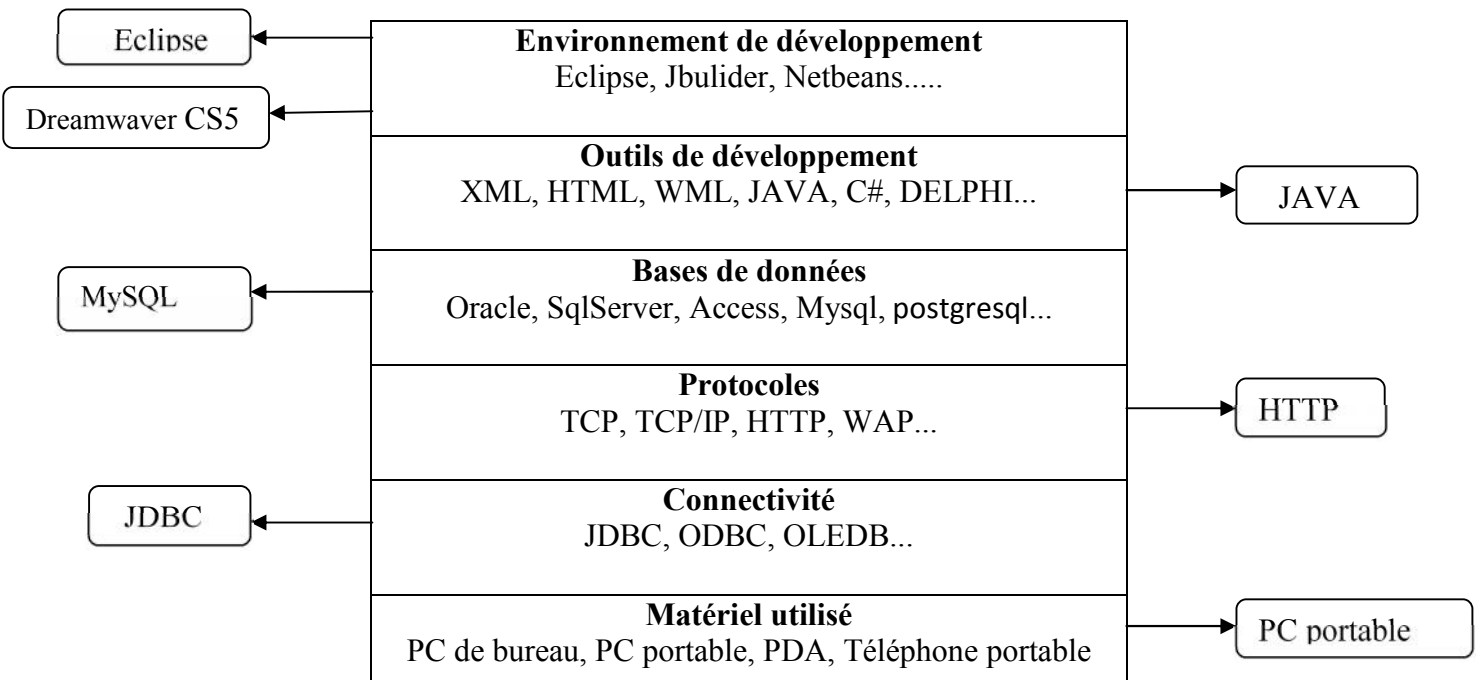


Figure IV.1 : Outils retenus

IV.2.1. Environnement de développement :

En matière d’environnement de développement, le marché offre un choix très large. Pour la réalisation de notre application nous avons utilisé l’environnement de développement suivant :

IV.2.1.1.Eclips : [14]

Eclipse IDE est un environnement de développement intégré libre (le terme *Eclipse* désigne également le projet correspondant, lancé par IBM) extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java (à l'aide de la

bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.

La spécificité d'Eclipse IDE vient du fait de son architecture totalement développée autour de la notion de plug-in (en conformité avec la norme OSGi) : toutes les fonctionnalités de cet atelier logiciel sont développées en tant que plug-in.

Eclipse a passé depuis son développement par plusieurs versions depuis la première version 1.0 en 2001 et jusqu'aujourd'hui. Dans notre travail nous avons utilisées la version **4.2** appelée **Juno**.

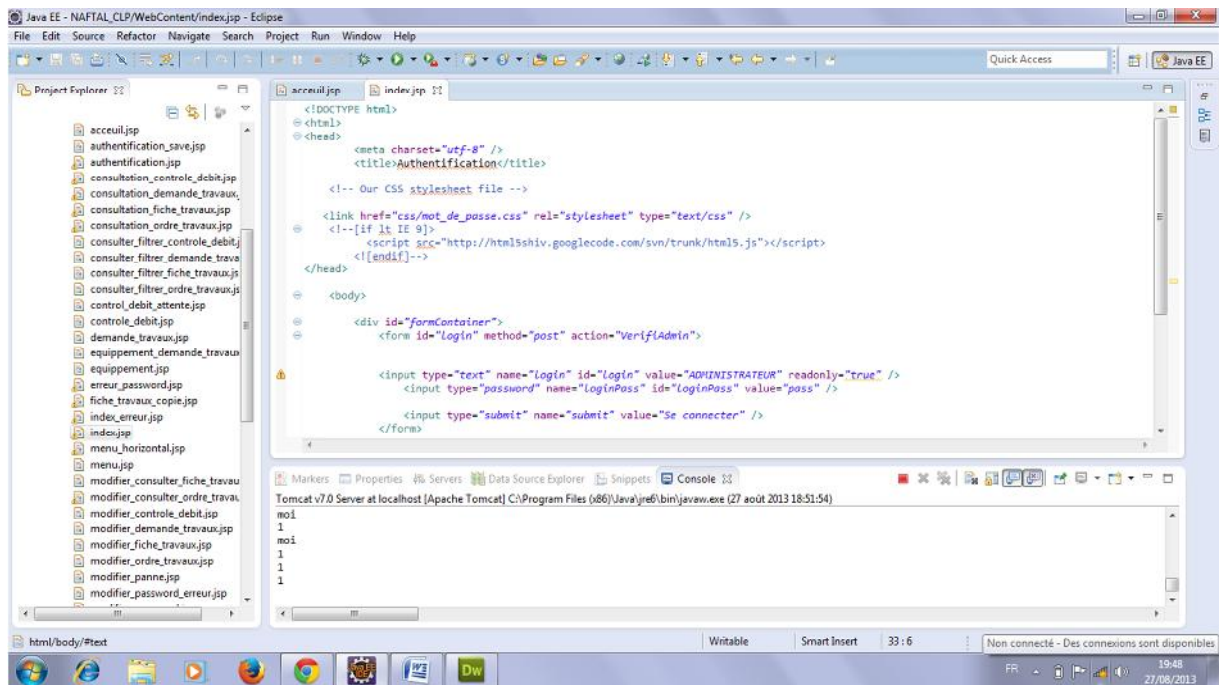


Figure IV.2 : Interface d'Eclipse

IV.2.1.2. Adobe dreamwaver CS5 :

Dreamweaver est le système professionnel par excellence pour la création de sites Web et d'applications. Sa puissante combinaison d'outils de mise en forme visuelle, de fonctions de développement d'applications et d'édition de code permet aux développeurs et aux concepteurs de créer des sites et des applications visuellement attrayants et normalisés.

Qu'il s'agisse de la prise en charge des concepts basés sur CSS ou de fonctionnalités de codage manuel, Dreamweaver fournit aux professionnels les outils dont ils ont besoin dans un environnement intégré et optimisé.

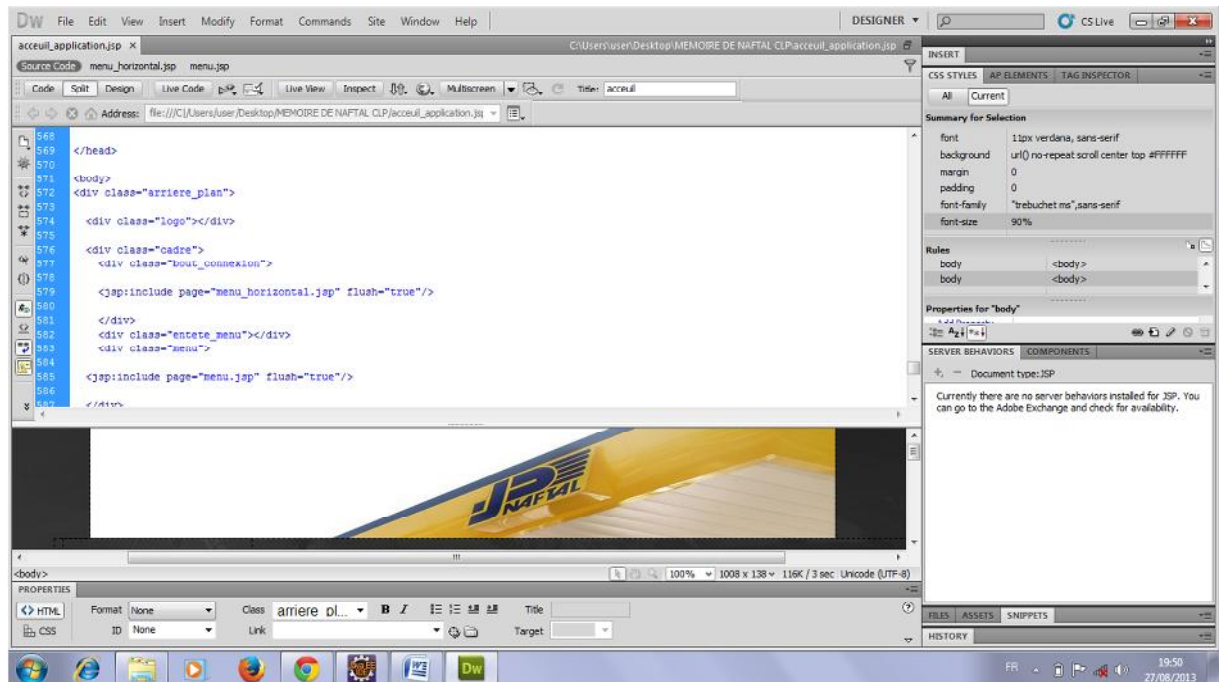


Figure IV.3 : Interface d'Adobe Dreamwaver CS5

IV.2.2. Outils de développements :

IV.2.2.1. JAVA : [15][16]

Java est un langage de programmation moderne orienté objet, développé par Sun Microsystems (aujourd'hui racheté par **Oracle**). Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitations (Windows, Linux, Macintosh, Solaris). Java donne aussi la possibilité de développer des programmes pour téléphones portables et assistants personnels. Enfin, ce langage peut-être utilisé sur internet pour des petites applications intégrées à la page web (applet) ou encore comme langage serveur (jsp). Java est aujourd'hui devenu un outil indispensable qui permet aux développeurs :

- d'écrire des logiciels sur une plate-forme et de les exécuter sur pratiquement toutes les autres plates-formes,
- de créer des programmes qui peuvent être exécutés dans un navigateur Web et accéder aux services Web disponibles,
- de développer des applications côté serveur pour des forums, des magasins et des sondages en ligne, pour le traitement de formulaires HTML, etc.,
- de combiner des applications ou des services basés sur le langage Java pour créer des applications ou des services très personnalisés,
- d'écrire des applications puissantes et efficaces pour les téléphones portables, les processeurs à distance, les produits de consommation et tous les autres types de dispositif électronique.

Dans notre mémoire nous avons choisi de développer en J2EE (Java 2 Entreprise Edition) : développement des applications Web en JAVA grâce aux servlets et jsp.

❖ Présentation de J2EE : [17] [18]

J2EE (*Java 2 Enterprise Edition*) est une norme proposée par la société Sun, portée par un consortium de sociétés internationales, visant à définir un standard de développement d'applications d'entreprises multi-niveaux, basées sur des composants.

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'application, 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'application 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*).

Dans la mesure où J2EE s'appuie entièrement sur le Java, il bénéficie des avantages et inconvénients de ce langage, en particulier une bonne portabilité et une maintenabilité du code.

De plus, l'architecture J2EE repose sur des composants distincts, interchangeables et distribués, ce qui signifie notamment :

- qu'il est simple d'étendre l'architecture ;
- qu'un système reposant sur J2EE peut posséder des mécanismes de haute-disponibilité, afin de garantir une bonne qualité de service ;
- que la maintenabilité des applications est facilitée.

L'architecture J2EE permet 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 XML, dans des annuaires d'entreprise ou encore dans des systèmes d'information complexes.

❖ Les Servlets : [19]

Une **servlet** est une classe Java qui permet de créer dynamiquement des données au sein d'un serveur HTTP. Ces données sont le plus généralement présentées au format HTML, mais elles peuvent également l'être au format XML ou tout autre format destiné aux navigateurs web. Les servlets utilisent l'API **Java Servlet** (package `javax.servlet`).

Une servlet s'exécute dynamiquement sur le serveur web et permet l'extension des fonctions de ce dernier, typiquement : accès à des bases de données, transactions d'e-commerce, etc. Une servlet peut être chargée automatiquement lors du démarrage du serveur web ou lors de la première requête du client. Une fois chargées, les servlets restent actives dans l'attente d'autres requêtes du client. Une servlet est une application développée dans un contexte client-serveur.

Voici un exemple de code d'une servlet :

```
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class DoIt extends HttpServlet
{
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException
    {
        out.println("<h1>Méthode doGet</h1>");
    }

    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException
    {
        out.println("<h1>Méthode doPost</h1>");
    }
}
```

❖ Les jsp : [20]

Les JSP (Java Server Pages) sont une technologie Java qui permettent la génération de pages web dynamiques.

La technologie JSP permet de séparer la présentation sous forme de code HTML et les traitements sous formes de classes Java définissant un bean ou une servlet. Ceci est d'autant plus facile que les JSP définissent une syntaxe particulière permettant d'appeler un bean et d'insérer le résultat de son traitement dans la page HTML dynamiquement. La technologie JSP mélange la puissance de Java côté serveur et la facilité de mise en page d'HTML côté client.

Une JSP est habituellement constituée :

- De données et de tags HTML
- De tags JSP
- De scriptlets (code Java intégré à la JSP)

Les fichiers JSP possèdent par convention l'extension .jsp.

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
    <title>Insert title here</title>
  </head>
  <body>
    <%
      --Ceci est un commentaire JSP --
      Out.print(" Voici un exemple d'une JSP");
    %>

  </body>
</html>
```

❖ **JavaBean : [21]**

Souvent raccourci en "bean", un JavaBean désigne tout simplement un composant réutilisable. Il est construit selon certains standards, définis dans les spécifications de la plateforme et du langage Java

La technologie java beans propose de simplifier et faciliter la création et l'utilisation de composants.

Les java beans possèdent plusieurs caractéristiques :

- la persistance : elle permet grâce au mécanisme de sérialisation de sauvegarder l'état d'un bean pour le restaurer. Ainsi si on assemble plusieurs beans pour former une application, on peut la sauvegarder.
- la communication, grâce à des événements, qui utilise le modèle des écouteurs introduit par Java 1.1
- l'introspection : ce mécanisme permet de découvrir de façon dynamique l'ensemble des éléments qui composent le bean (attributs, méthodes et événements) sans avoir le code source.
- la possibilité de paramétrer le composant : les données du paramétrage sont conservées dans des propriétés.

IV.2.2.2. HTML : [22]

HTML (Hyper Text Markup Language) est un langage de description (et non un langage de programmation) qui va nous permettre de décrire l'aspect d'un document, d'y inclure des informations variées (textes, images, sons, animations) et d'établir des relations entre ces informations grâce aux liens hypertextes. HTML est un langage qui définit la structure logique d'un document web. Il a été comme un langage de structuration logique et utilisé comme langage de présentation et de mise en page.

Les pages produites avec ce langage portent le nom de page HTML. Elles sont en général stockées sur un serveur Web, qui les fournira à la demande. Du côté client, un navigateur est chargé de l'interprétation de pages HTML et de leur affichage. On dit donc que le navigateur affiche des pages web.

V.2.2.3. JavaScript : [23]

JavaScript (souvent abrégé JS) est un langage de programmation de scripts principalement utilisé dans les pages web interactives mais aussi côté serveur. . JavaScript fait ce que HTML ne peut pas. Il permet de traiter les saisies de l'utilisateur faites au clavier ou avec la souris et d'en réagir avec des sorties à l'écran ou des modifications dynamiques dans la

page Web affichée. L'utilisateur introduit donc quelques valeurs, JavaScript fait les calculs et affiche le résultat sans qu'il y ait d'échanges entre le client et le serveur web.

La structure classique d'une page HTML contenant du JavaScript est la suivante :

```
<script language="JavaScript" type="text/javascript" >
function verif ( )
{
    mes="";
    if(document.form.login.value=="") {mes=mes+"\n-Veuillez saisir votre login";}
    else
        if(document.form.login.value=='Login') {mes=mes+"\n-Veuillez saisir votre login";}
    if(document.form.mdp.value=="") {mes=mes+"\n-Veuillez saisir votre mdp";}
    else
        if(document.form.mdp.value=='Mot de passe') {mes=mes+"\n-Veuillez saisir votre mdp";}
    if(mes!="")
    {
        alert(" Attention :"+mes);
        return false;
    }
}
</script>
```

IV.2.3. Les serveurs déployés :

IV.2.3.1. Le serveur Apache :

Le serveur Apache est le serveur HTTP le plus répandu avec près de 60% du marché. Tomcat est un projet open source, réalisé par la communauté Jakarta, au sein du groupe Apache. C'est un serveur d'applications Java. Les applications web qu'il est capable d'exécuter sont écrites en Java. Pour traiter les requêtes entrantes, il existe des connecteurs capables de véhiculer les requêtes du serveur web frontal au serveur d'applis.

Le serveur Apache est le serveur HTTP le plus répandu avec près de 60% du marché, soit le double de son principal concurrent Microsoft-IIS. Ces principaux avantages tiennent de son mode de distribution et à son mode de licences. L'avantage principal de ce produit n'est pas la gratuité, mais ce serveur http est le plus libre, et offre une totale indépendance à l'utilisateur pour la maintenance et le développement, il est aujourd'hui le serveur web le plus utilisé au monde.

IV.2.3.2. Le conteneur de servlets Tomcat :

Le conteneur de servlets choisi est le moteur Tomcat 6.0 développé par la fondation Apache. Le dialogue entre le moteur de servlets et le serveur web s'effectue à l'aide d'un module logiciel appelé connecteur. Tomcat peut fonctionner sur d'autres serveurs Web mais seul le couple Tomcat/ Apache a été testé. Le module Tomcat du serveur Apache a été développé à partir des sources de Sun Microsystems. Il représente une implémentation de référence pour les servlets. Tomcat peut fonctionner seul, mais cela n'est pas une solution efficace. En exploitation il est préférable d'associer Tomcat avec un serveur HTTP plus puissant, qui se chargera du contenu statique. Tomcat pourra ainsi être mis à contribution uniquement pour les requêtes mettant en œuvre des servlets, et nous opterons pour ce principe.

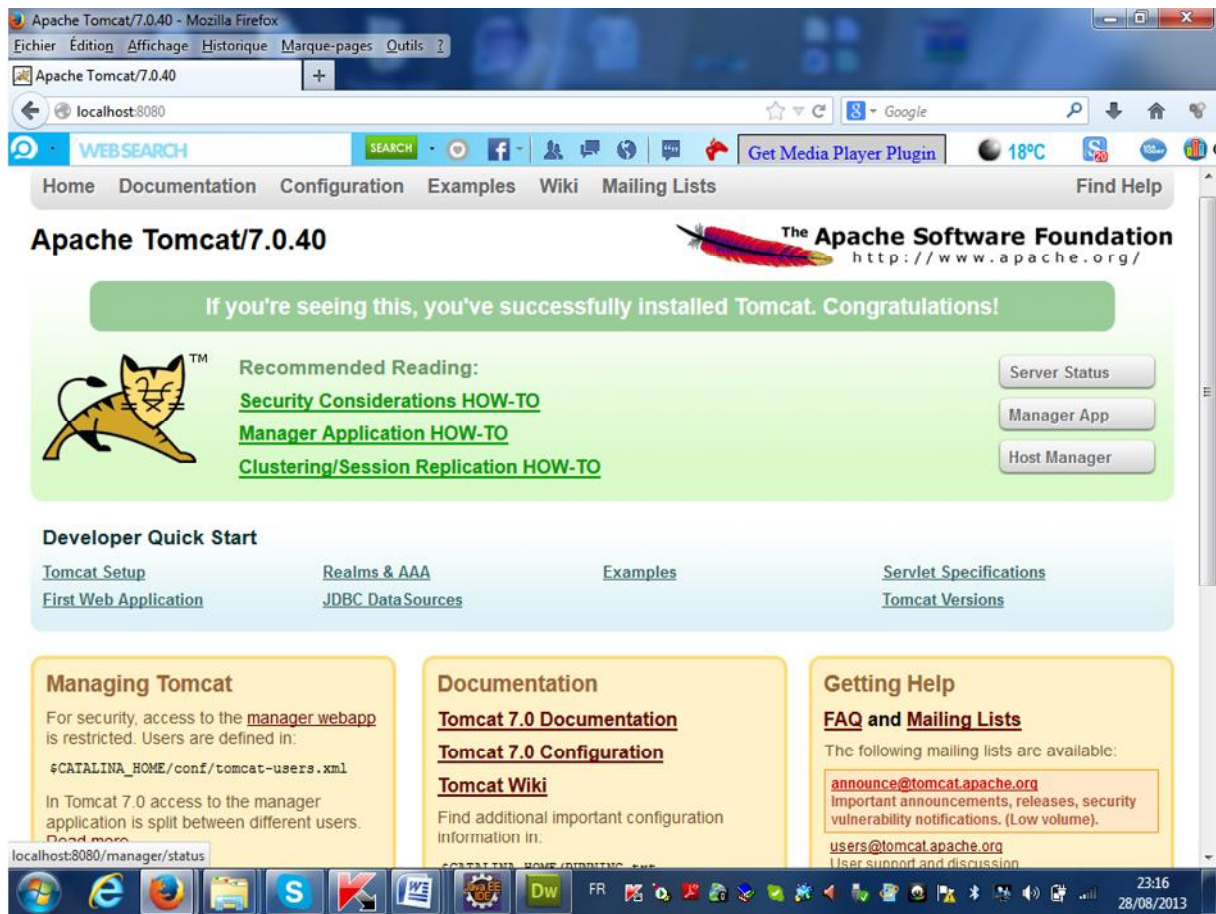


Figure IV.4 : Interface d'Apache Tomcat

IV.2.3.3. Le serveur de données :

❖ Wampserver :

wampserver est un paquetage contenant à la fois deux serveurs (Apache et MySQL), un interpréteur de script (PHP), les deux bases SQL PhpMyAdmin et SQLiteManager pour gérer plus facilement les bases de données. Il permet d'installer automatiquement et facilement une plateforme permettant l'exploitation d'un site web en PHP qui éventuellement aurait besoin d'un accès à une base de données.

❖ L'interface PHPmyadmin :

phpMyAdmin est une application web qui permet de gérer un serveur de bases de données MySQL. Dans un environnement multiutilisateur, cette interface écrite en PHP permet également de donner à un utilisateur un accès à ses propres bases de données.

La figure suivante montre une copie d'écran de la page d'accueil de PHPmyadmin. l'écran est divisé en deux parties. sur la gauche on peut afficher toutes les bases de données gérés par le serveur. la partie droite présente l'ensemble des opérations disponibles en fonction du contexte.

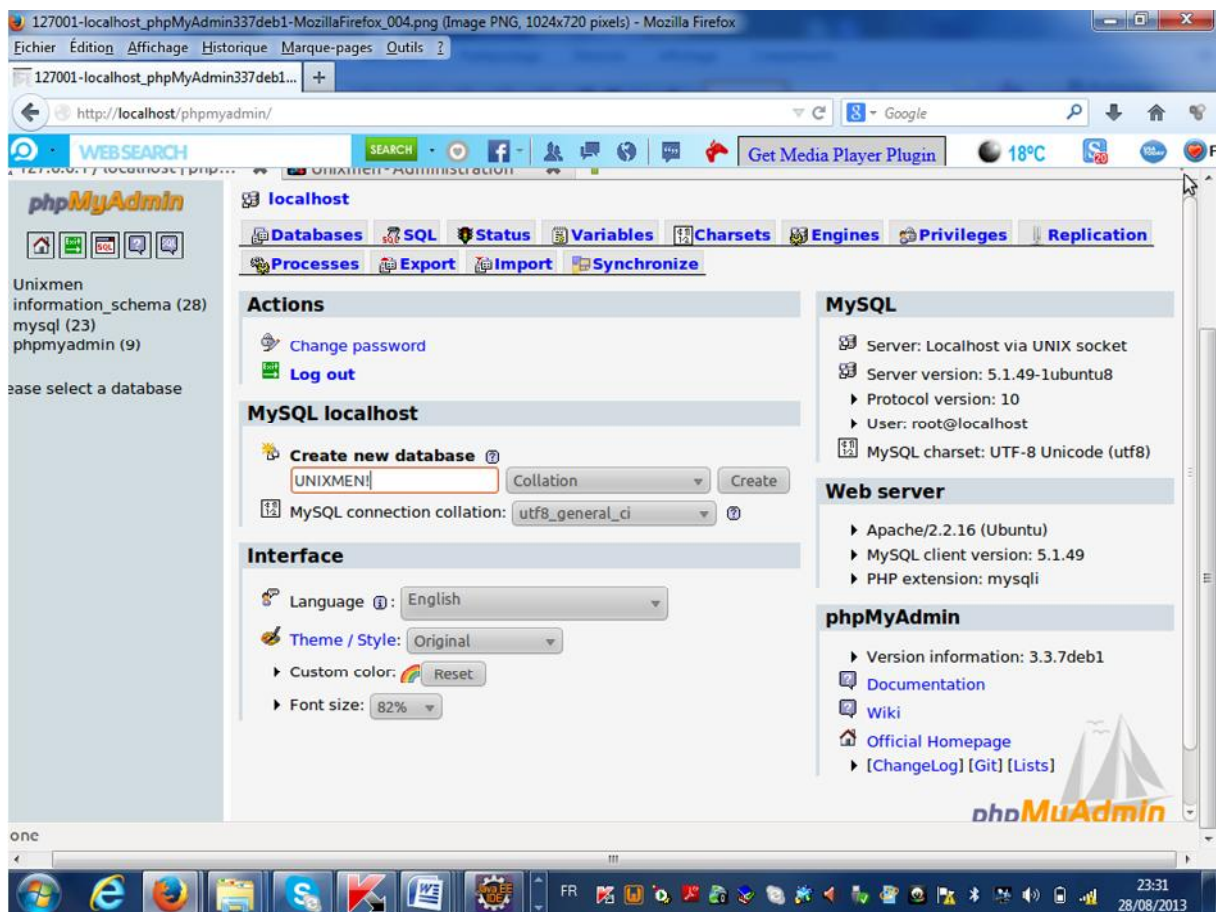


Figure IV.5 : Interface de PhpMyAdmin

❖ Structure des tables de la base de donnée :

➤ Table user

Nom	Type de données	Taille	Description
id	Int	02	Identificateur
login	Text	30	Nom d'utilisateur (fixe)
password	Text	30	Mot de passe

➤ Table client

Nom	Type de données	Taille	Description
Code_client	Text	10	Identificateur
Raison_social	Text	30	
Adresse	Text	60	
Commune	Varchar	30	
Daïra	Varchar	30	
Num_tel	Int	10	Numero de telephone
Activité	Text	15	

➤ Table Demande_installation:

Nom	Type de données	Taille	Description
Num_demande	Int	10	Numero de la demande
Code_client	Text	10	
Des_client	Text	30	Designation client
adresse	Text	60	
commune	Varchar	30	
daira	Varchar	30	
activité	Text	15	

tel	Int	10	
observation	Text	90	

➤ **Table Ordre_travaux :**

Nom	Type de données	Taille	Description
Num_OT	Int	10	Numéro de l'ordre de travaux
Code_client	Text	10	
Des_client	Text	30	Designation client
Adress	Text	30	
commune	Varchar	30	
daira	Varchar	30	
activité	Text	15	
agent	Text	60	
Date_ot	Date	08	Date de l'ordre de travaux

➤ **Table PV_propane(p11&p35)**

Nom	Type de données	Taille	Description
Code_pv	Int	10	
Code_client	Text	10	
Designation	Text	30	
commune	Varchar	30	
daira	Varchar	30	
activité	Text	30	
realisateur	Text	60	
Date_realisation	date	08	
Nbre_P11	int	02	Nbre de bouteilles

Nbre_P35	int	02	Nbre de bouteilles
Montant	int	10	En da
Observation	Text	90	
Mise_en_service	Date	08	Date de mise en service

➤ **Table PV_CIETRNE**

Nom	Type de données	Taille	Description
Code_pv	Int	10	
Code_client	Text	10	
Des_client	Text	30	Designation client
commune	Varchar	30	
daira	Varchar	30	
activité	Text	30	
realisateur	Text	60	
Code_citerne	text	10	
Capacité_citerne	text	04	En litre
Montant	int	10	En da
Observation	Text	90	
Mise_en_service	Date	08	Date de mise en service
Date_reepreuve	Date	08	
Date_acquisition	Date	08	

➤ **Table Facture:**

Nom	Type de données	Taille	Description
Code_facture	Text	10	
Des_client	Text	30	Designation client
Adresse	Text	60	
Commune	Varchar	30	
Daira	Varchar	30	
Activité	Text	15	
paiement	Text	10	Mode de paiement
date	Date	08	
installation	Text	60	
montant	int	10	

➤ **Table reepruve**

Nom	Type de données	Taille	Description
Code_pv	Int	10	
Code_client	Text	10	
Raison_social	Text	30	
Adress	Text	60	
Commune	Varchar	30	
daira	Varchar	30	
Activtè	Text	15	
Realisateur	Text	90	
Date_reepreuve	Date	08	
Duree	int	08	

observation	Text	90	
-------------	------	----	--

➤ **Table citerne**

Nom	Type de données	Taille	Description
Code_citerne	Int	10	
capacité_citerne	Text	04	En litre
prix_citerne	int	10	

➤ **Table realisateur**

Nom	Type de données	Taille	Description
Code_realisateur	Text	10	
Des_realisateur	Text	30	

➤ **Le middleware JAVA Data Base Connectivity (JDBC) :**

La technologie JDBC (*Java DataBase Connectivity*) est une API fournie avec Java permettant de se connecter à des bases de données, c'est-à-dire que JDBC constitue un ensemble de classes permettant de développer des applications capables de se connecter à des serveurs de bases de données (SGBD).

IV.3. Architecture du système :

Le principe de fonctionnement de la solution proposée est basé sur une architecture Client-Serveur trois tiers. Le premier niveau est le client demandeur de la ressource, le second niveau est le serveur web utilisant le conteneur de servlet (Apache Tomcat), le troisième niveau est le serveur de BD (Mysql).

IV.4. Présentation de quelques interfaces de notre plate-forme :

Pour réaliser une application Web, il faut tenir compte de la qualité de l'interface homme/machine et permettre une meilleure adéquation de celle-ci aux besoins des différents utilisateurs.

Dans ce qui suit nous proposons quelques interfaces de notre Application :

IV.4.1 Interface d'authentification :

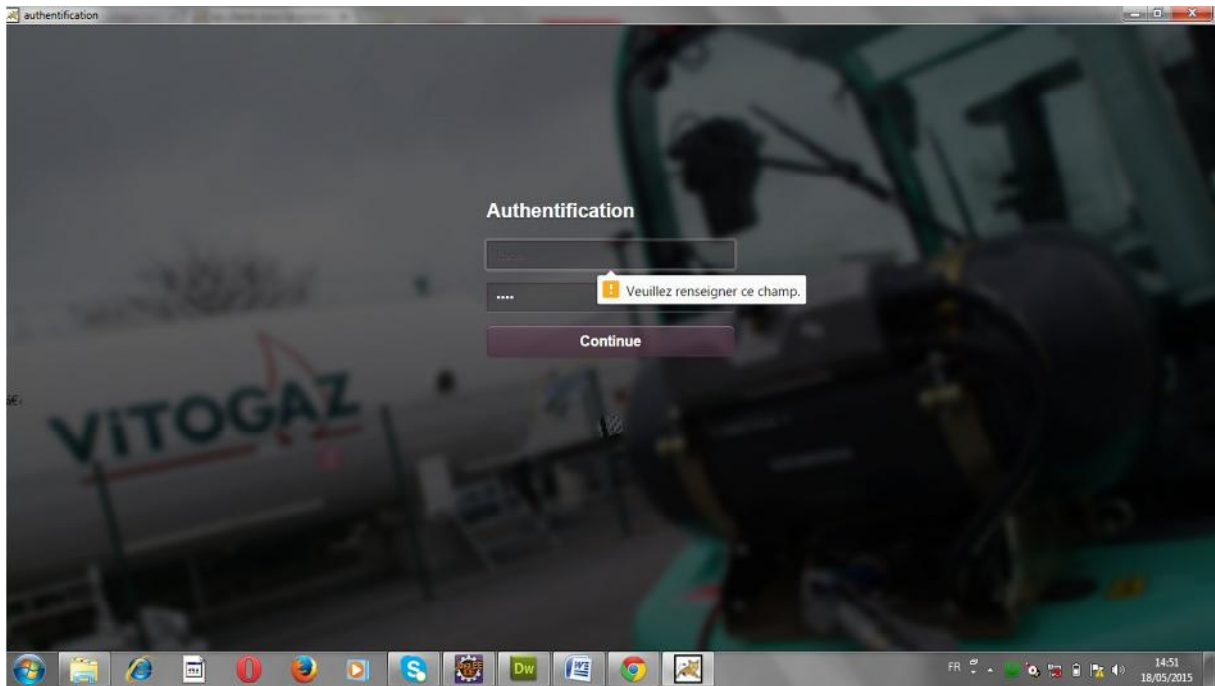


Figure IV.6 : Interface d'authentification

IV.4.2 Interface de modification/suppression d'un client

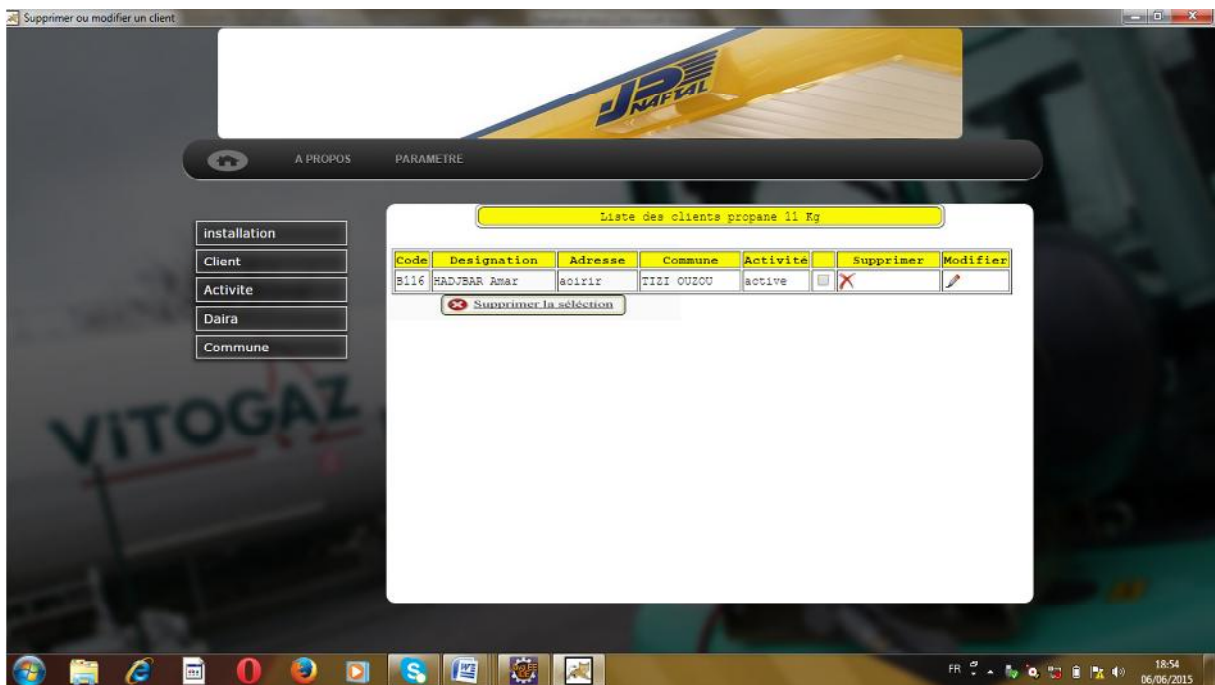
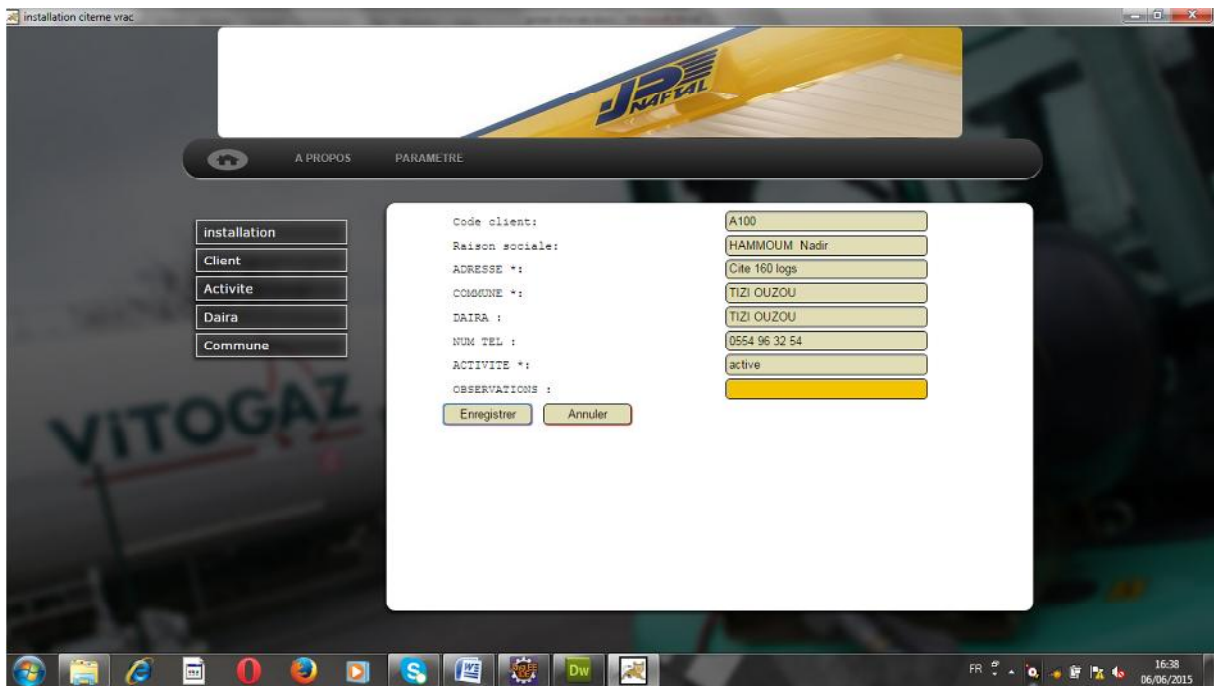


Figure IV.7 : interface de modification ou de suppression d'un client

IV.4.3 interface d'ajout d'une demande d'installation



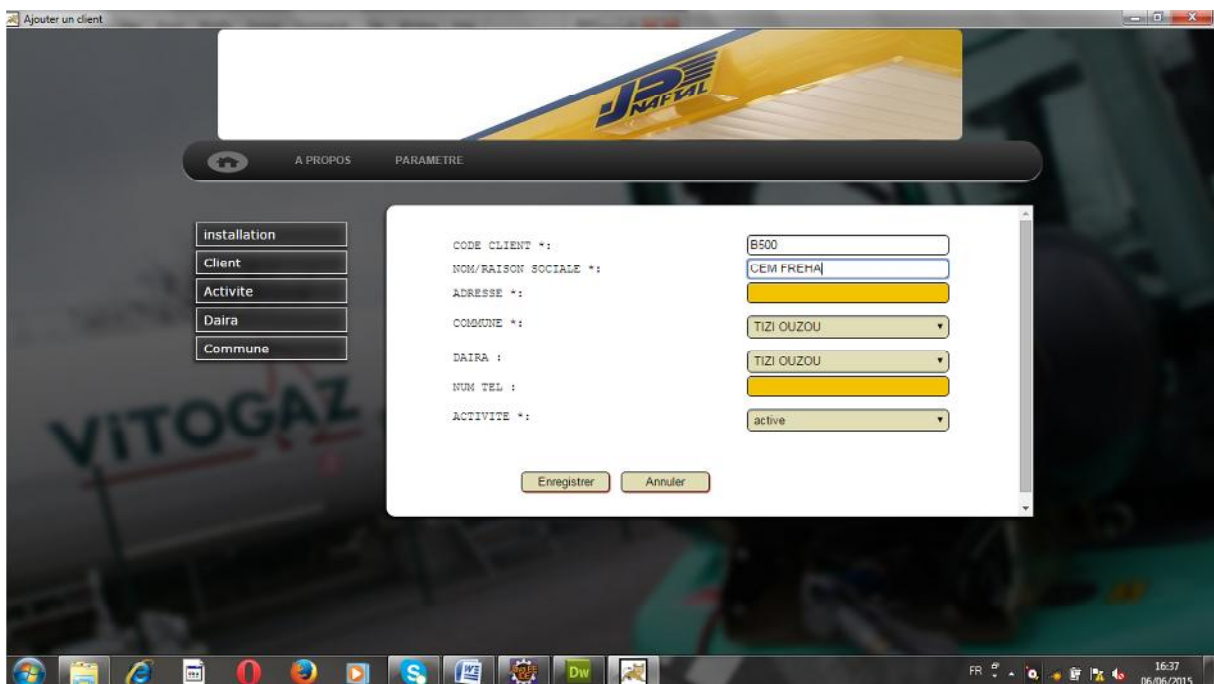
The screenshot shows a web application window titled "installation citerne vrac". The interface features a dark header with a logo and navigation links "A PROPOS" and "PARAMETRE". On the left, there is a vertical menu with buttons for "installation", "Client", "Activite", "Daira", and "Commune". The main content area contains a form with the following fields:

Code client:	A100
Raison sociale:	HAMMOUM Nadir
ADRESSE *:	Cite 160 logs
COMMUNE *:	TIZI OUZOU
DAIRA :	TIZI OUZOU
NUM TEL :	0554 96 32 54
ACTIVITE *:	active
OBSERVATIONS :	

At the bottom of the form are two buttons: "Enregistrer" and "Annuler". The Windows taskbar at the bottom shows the date and time as 16:38 on 06/06/2015.

Figure IV.8 : Interface d'ajout d'une d'installation

IV.4.4 Interface d'ajout d'un nouveau client



The screenshot shows a web application window titled "Ajouter un client". The interface features a dark header with a logo and navigation links "A PROPOS" and "PARAMETRE". On the left, there is a vertical menu with buttons for "installation", "Client", "Activite", "Daira", and "Commune". The main content area contains a form with the following fields:

CODE CLIENT *:	B500
NOM/RAISON SOCIALE *:	CEM FREMA
ADRESSE *:	
COMMUNE *:	TIZI OUZOU
DAIRA :	TIZI OUZOU
NUM TEL :	
ACTIVITE *:	active

At the bottom of the form are two buttons: "Enregistrer" and "Annuler". The Windows taskbar at the bottom shows the date and time as 16:37 on 06/06/2015.

Figure IV.9 : Interface d'ajout d'un nouveau client

IV.4.5 Interface d'ajout d'un ordre de travail

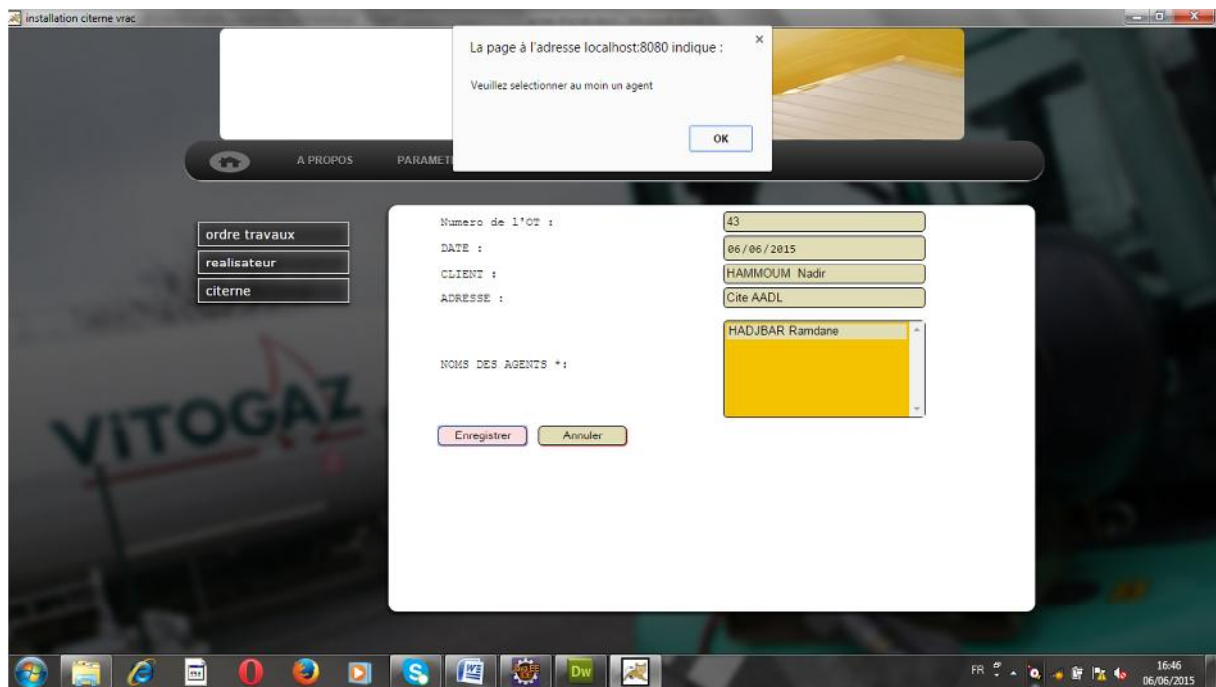


Figure IV.10 : Interface d'ajout d'un ordre de travail

IV.4.6 Interface d'ajout d'une facture

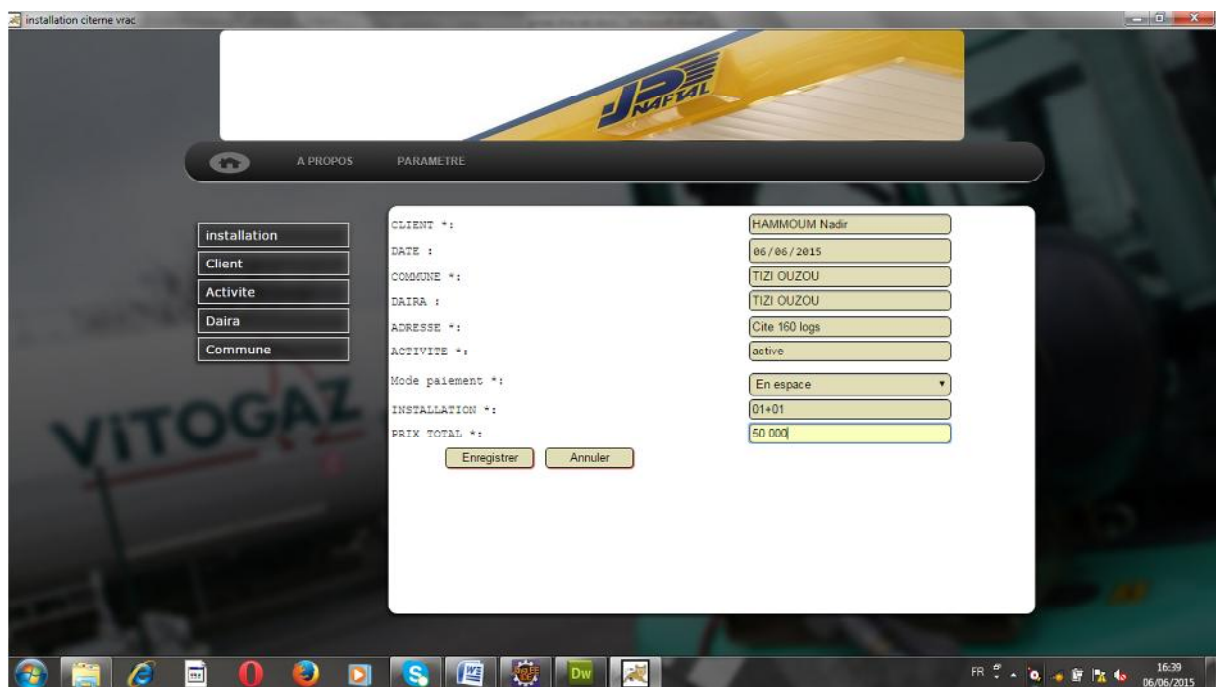


Figure IV.11 : Interface d'ajout d'une facture

IV.4.7 Interface d'impression d'une facture

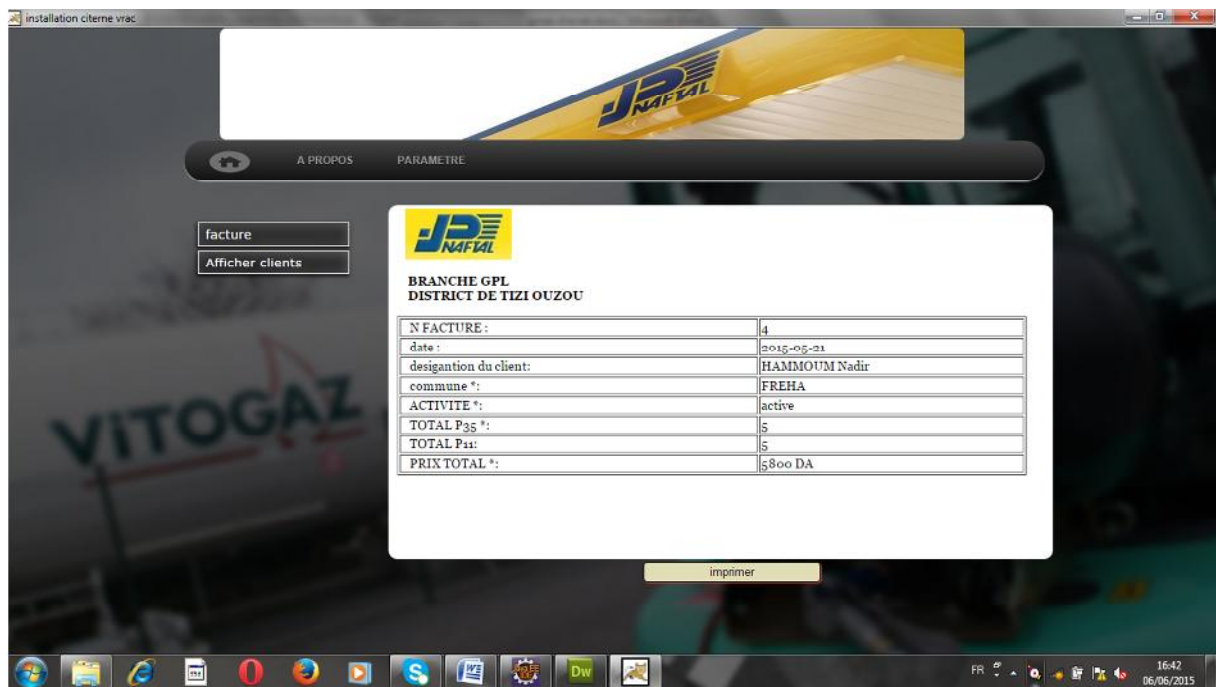


Figure IV.12 : Interface d'impression d'une facture

IV.4.8 Consultation des demandes d'installation propane 35 kg

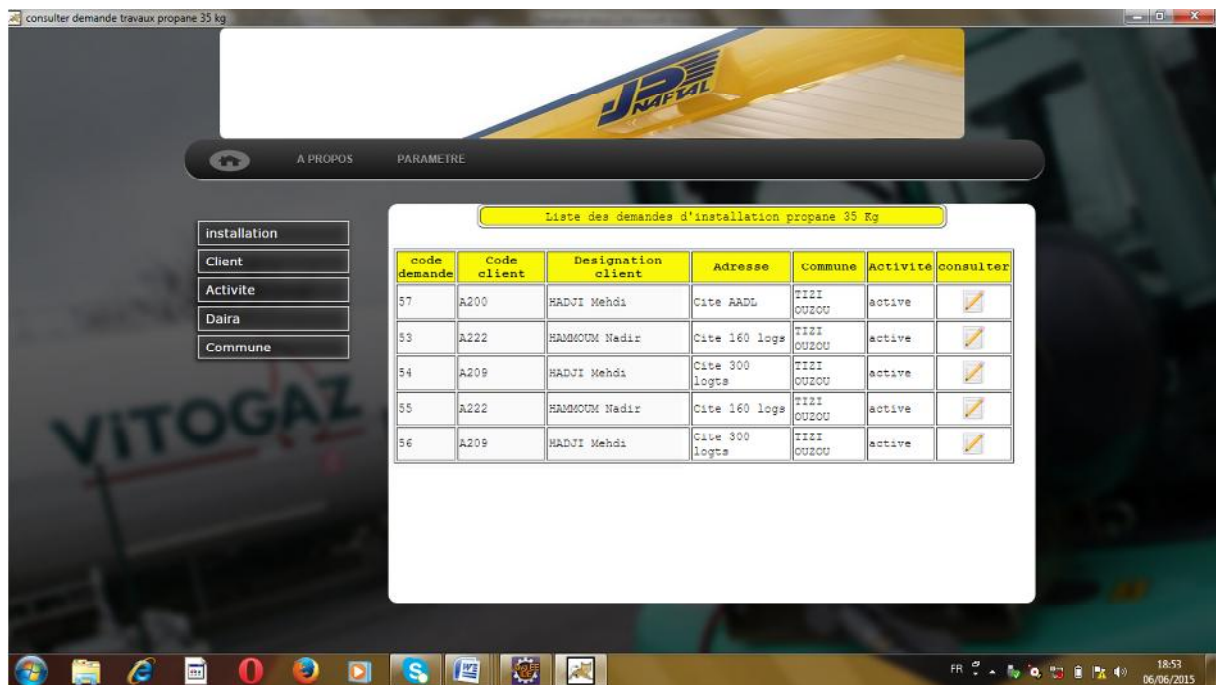


Figure IV.13 : Consultation des demandes d'installation propane 35 kg

IV.4.9 interface d'ajout d'une citerne

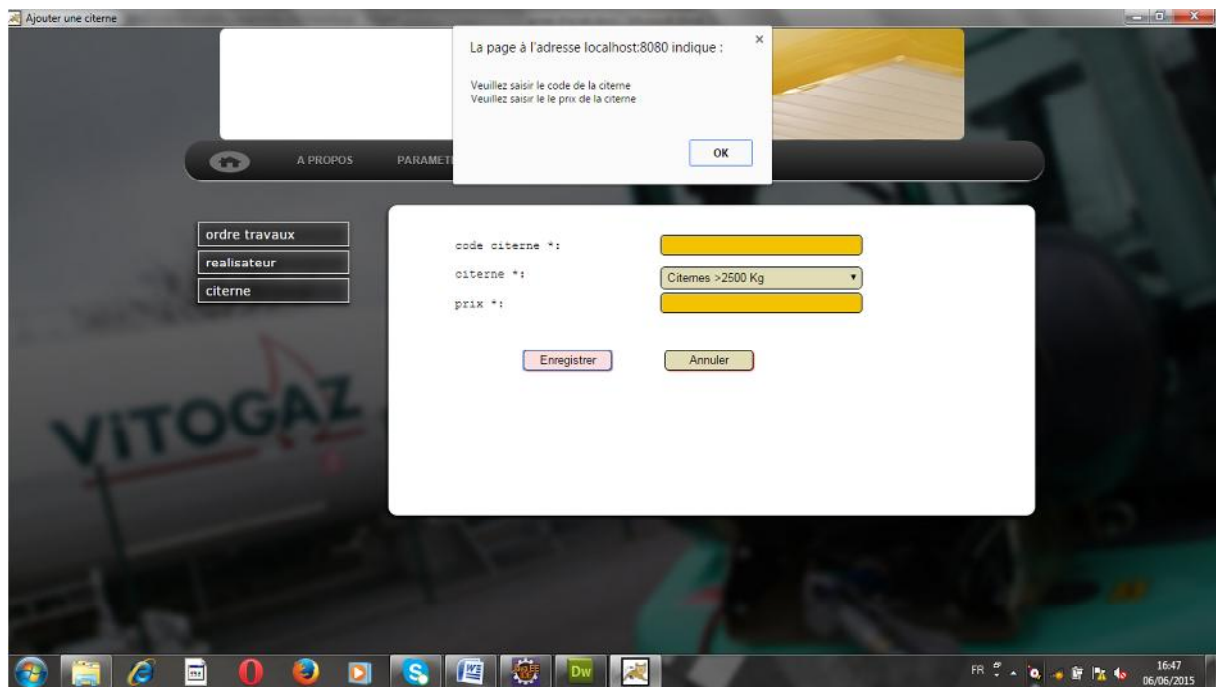


Figure IV.14 : Interface d'ajout d'une citerne

IV.4.10 Interface de consultation des ordres de travaux

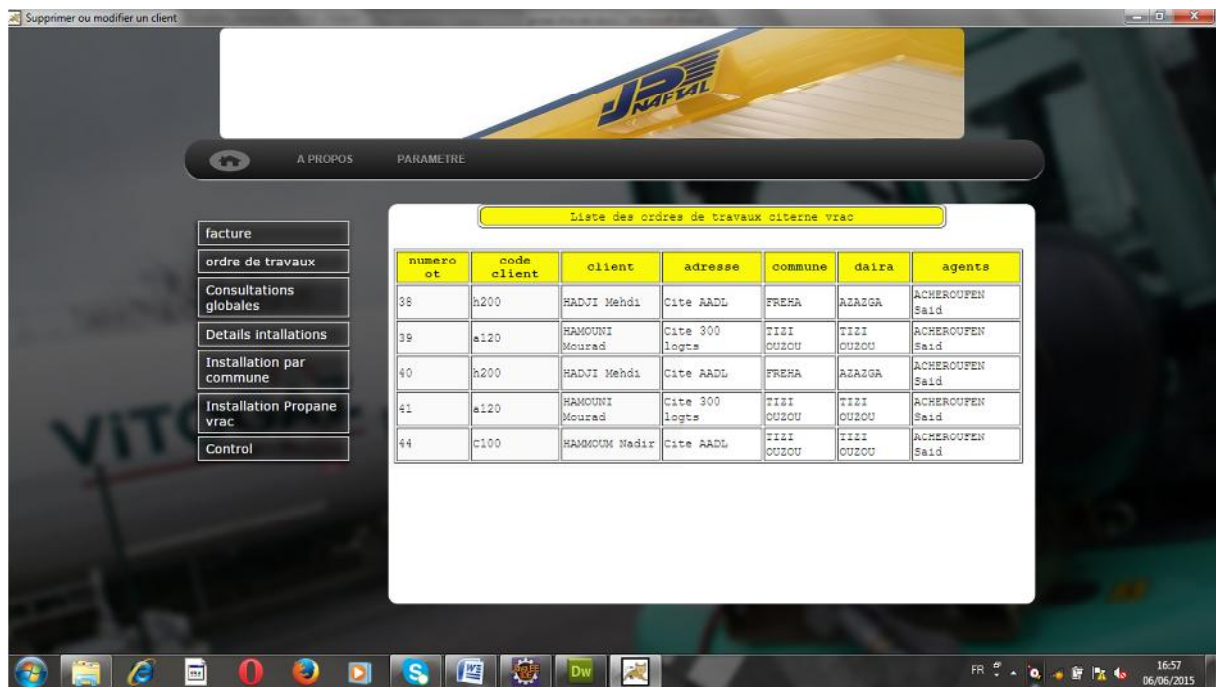


Figure IV.15 : Interface de consultation des ordres de travaux

IV.4.11 Interface d'ajout d'un procès verbal

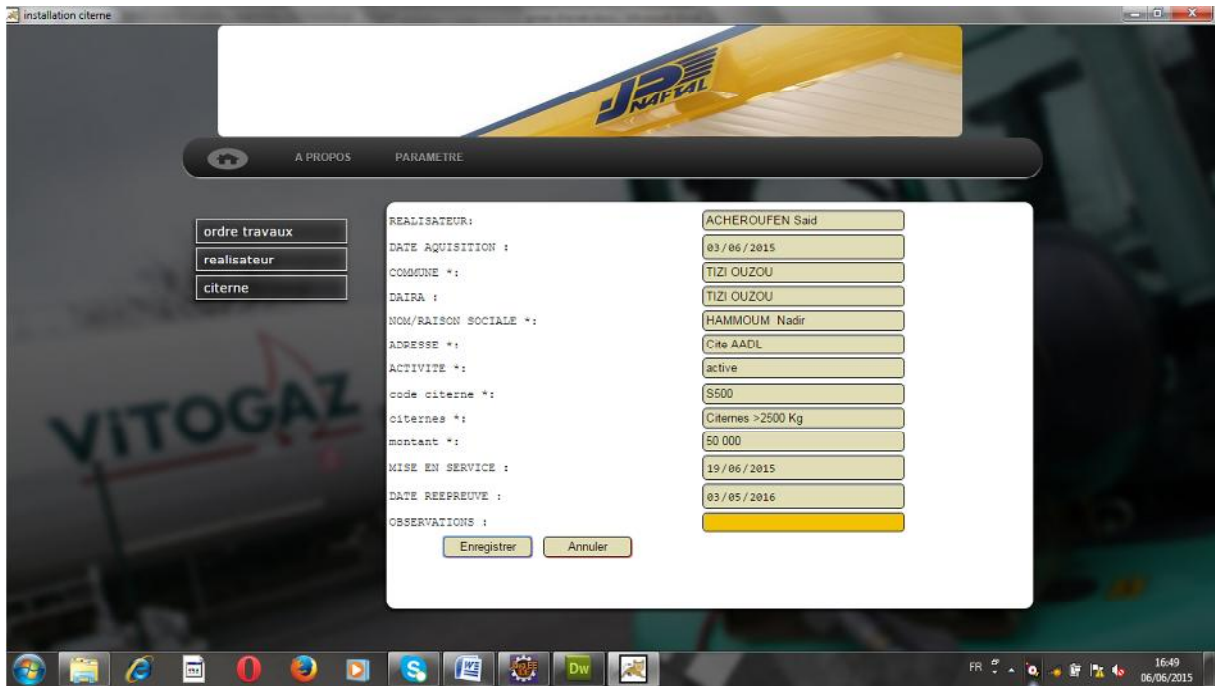


Figure IV.16 : Interface d'ajout d'un procès verbal

IV.4.12 Interface de modification de mot de passe

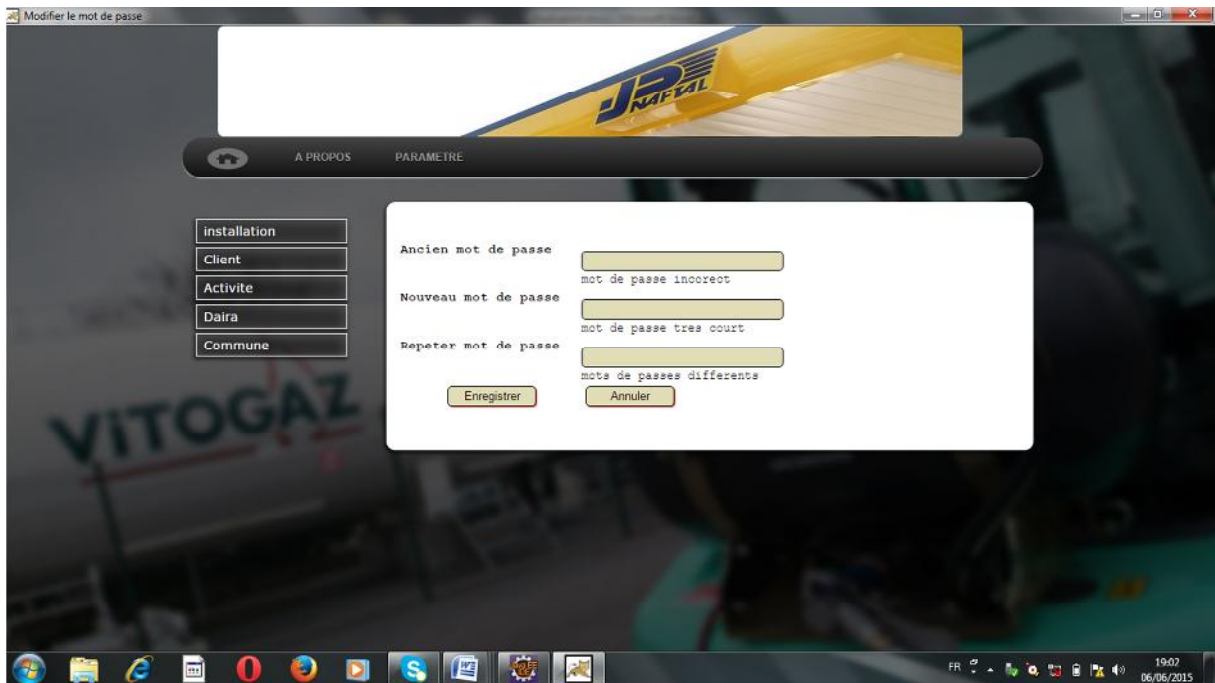


Figure IV.17 : Interface de modification du mot de passe

IV.6 Conclusion :

Ce dernier chapitre a été consacré à la présentation de l'étape réalisation de notre application. Ainsi, nous avons présenté les différentes tables de la base de données utilisées et les outils logiciels qui nous ont permis de réaliser cette application, à savoir l'environnement de développement, les langages de programmation et les logiciels nécessaires pour notre travail. Enfin nous avons donné quelques exemples d'interfaces pour illustrer les principales fonctionnalités de notre application



Conclusion générale



Toute entreprise, quelle que soit sa vocation et son caractère, doit se mettre au diapason de la progression technologique et faire face par l'automatisation de ses structures et la formation de son personnel afin d'améliorer son rendement et son service et d'assurer sa place sur le marché.

La réalisation de ce travail nous a permis de découvrir le domaine de la gestion des installations propane, de voir sur le champ de travail les différentes tâches effectuées par le chargé d'étude et le chef de section MIF. En plus de l'expérience, ce projet à travers les étapes de mise en œuvre notamment l'analyse, la conception et la réalisation, nous a conduit à l'utilisation des outils de conception « UML » et de réalisation « produit Java Entreprise Edition » avec tous ses outils et fonctions et de se familiariser avec eux,

L'objectif de notre projet était de concevoir une application Client/serveur pour la gestion des installations propane. Cette application offre aux différents utilisateurs plus de souplesse et d'efficacité dans l'exécution de leurs tâches en conformités avec les exigences de l'entreprise NAFTAL .

Cependant, quoiqu'elle présente diverses fonctionnalités, quelques améliorations et extensions peuvent être apportées à cette application. En guise de perspectives, les possibilités d'évolutions de ce projet sont nombreuses allant jusqu'à la gestion complète de tous les départements et inclure ainsi les services finances et comptabilités, et la gestion commerciale tout en mettant l'accent sur l'aspect coopératif de ces structures. Ainsi nous espérons que notre travail sera de grand intérêt pour NAFTAL et un support d'aide éventuel aux lecteurs.

Références Bibliographiques

Bibliographie :

- [1] : <http://www.apprendre-informatique.com/forum/toile/reseau/type-reseaux/>
- [2] : <http://www.commentcamarche.net/contents/initiation/topologi.php3>
- [3] : http://fr.wikiversity.org/wiki/Topologie_de_reseaux/Maillé
- [4] : Polycopié complément de cours Communication de données et réseaux.
Mr DJEMAH « 4ème année Ingénieur Promotion 2008 /2009 ».
- [5] : Architecture des réseaux .Auteur : Bertrand Petit, Edition 2008.
- [6] : Mémoire (Ingénieur) réalisé par Melle BENCHABANE Bahia et Melle BALHADJ Sonia.
Thème : Conception et réalisation d'un portail web collaboratif. Cas : La direction
Coordination Groupe Finance de SONATRACH (DG). Promotion
2008/2009.
Etablissement : Université Mouloud Mammeri de Tizi-Ouzou.
- [07] : P.Yves, Architecture client serveur internet et intranet, DOUD 2000
- [8] : http://fr.wikipedia.org/wiki/Model_OSI
- [9] : http://fr.wikiversity.org/wiki/Modèle_TCP-IP
- [10] : « Le guide de l'utilisateur UML » (Grady Booch, James Rumbaugh, Ivar Jacobson).
Edition: Eyrolles 2003
- [11] : « UML par la pratique » (Pascal Roques). Edition Eyrolles 2004
- [12]: <http://www.commentcamarche.net/contents/entreprise/groupware.php3>
- [13] : « Conception et réalisation des bases de données: De UML à SQL » (Jacques Guyot).
Editions systèmes et information 2002.

- [14] : <http://www.techno-science.net/?onglet=glossaire&definition=517>
- [15] : <http://www.futura-sciences.com/magazines/high-tech/infos/dico/d/internet-java-485>
- [16] : <http://www.java.com/fr/about/>
- [17] : <http://www.commentcamarche.net/contents/548-j2ee-java-2-enterprise-edition>
- [18] : <http://jmdoudoux.developpez.com/cours/developpons/java/chap-j2ee-javaee.php>
- [19] : <http://fr.wikipedia.org/wiki/Servlet>
- [20] : <http://www.commentcamarche.net/contents/601-introduction-aux-java-server-pages>
- [21] : <http://www.jmdoudoux.fr/java/dej/chap-javabean.html>
- [22] : Saleh, « les langages de développement web », Edition HERMES SCIENCE, 1996.
- [23] : <http://www.toutjavascript.com/savoir/savoir00.php3>