

**République Algérienne Démocratique et Populaire**  
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**  
**Université Mouloud MAMMERRI, Tizi-Ouzou**



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

**Mémoire de Fin d'Etudes**

*En vue de l'obtention du diplôme*

*Master en Informatique*

*Thème*

**Conception et réalisation d'une application J2EE pour la  
gestion de la maintenance**

**Cas : NAFTAL TIZI OUZOU**

**Encadré par :**

**M<sup>er</sup> KARBICHE**

**Réalisé par :**

**Meziani Sonia  
OUAMER Tassadit**

**Promotion : 2013-2014**

# Dédicaces

Je dédie ce modeste travail à :

- ✚ Ma très chère mère et mon père à qui je souhaite une longue vie.
- ✚ Mon adorable sœur Melissa.
- ✚ Mes très chers frères.
- ✚ Mon cher fiancé RABEh.
- ✚ Tous mes amis.
- ✚ Mon binôme Tassadit et sa famille.
- ✚ Tous ceux qui m'aiment et que j'aime.

MEZIANI Sonia

# Dédicaces

Je dédie ce modeste travail à :

- ✚ Ma très chère mère et mon père qui ont toujours été la pour moi à qui je souhaite une longue vie.
- ✚ Mon adorable sœur Mounia.
- ✚ Mon cher frère Nabil a qui je souhaite la réussite.
- ✚ Mon future mari ma source de bonheur khelifa pour son aide, son temps, son assistance et soutien.
- ✚ Tous mes amis .
- ✚ Mon binôme Sonia et sa famille.
- ✚ Tous ceux qui m'aiment et que j'aime.

OUAMER Tassadit

# Sommaire

## Introduction générale

### Chapitre I : Réseaux, Internet et web

I.1 Introduction .....	2
I.2 Les réseaux .....	2
I.2.1 Définition du réseau .....	2
I.1.2. Objectifs des réseaux .....	2
I.1.3 Classification des réseaux .....	2
I.1.3.1 Classification des réseaux selon l'étendu.....	2
I.1.3.2 Classification des réseaux selon la topologie .....	3
I.2.3.2.1 La topologie en bus .....	3
I.2.3.2.2 La topologie en étoile .....	4
I.2.3.2.3 La topologie en anneau.....	4
I.1.4 Architecture des réseaux .....	5
I.1.4.1 L'architecture OSI (Open Systems Interconnexion).....	5
I.1.4.2 L'architecture TCP/IP.....	7
I.1.4.1.3 Architecture client/ serveur.....	9
I.1.5 La sécurité des réseaux.....	11
I.3 Internet.....	11
I.3.1 Définition.....	11
I.3.2 Intérêt de l'internet.....	11
I.3.3 Les services d'internet.....	12
I.4 Le web.....	13
I.4.1 Définition.....	13
I.4.2 Les concepts du web.....	13

I.4.3 Les langages du web.....	14
I.5 Conclusion.....	15

## Chapitre II : Etude de l'existant

II.1 Introduction .....	16
III.2 Présentation de l'entreprise.....	16
III.2.1 Historique de L'entreprise.....	16
III.2.2 Organisation de NAFTAL.....	16
III.2.3 Mission de NAFTAL .....	17
III.2.4 Les objectifs de NAFTAL .....	17
III.2.5 Moyens.....	17
III.2.6. Organigramme général de l'entreprise.....	19
III.2.7 Présentation du champ d'étude .....	20
III.2.7.1 Organigramme du champ d'étude .....	20
III.2.7.2 Situation informatique du champ d'étude.....	20
III.3 Etude de l'existant.....	21
III.3.1 Introduction.....	21
III.3.2 Etude des postes.....	21
III.3.3 Etude des documents .....	23
III.3.3.1 Liste des documents existant.....	23
III.3.4 Etude des procédures existantes.....	28
III.3.4.1 Introduction.....	28
III.3.4.2 Description des procédures existantes.....	28
III.3.5 Diagramme des flux.....	28
III.3.5.1 Concepts utilisés.....	28
III.3.5.2 Diagramme des flux d'information.....	29
III.3.5.3 Liste des flux d'information.....	29

III.3.5 Problématique du champ d'études.....	30
--	----

III.4 Conclusion.....	30
-----------------------	----

## Chapitre III : Analyse et conception

III.1 Introduction.....	31
-------------------------	----

III.2 Phase d'Initialisation.....	31
-----------------------------------	----

III.3 Phase d'Elaboration.....	31
--------------------------------	----

III.3.1 Identification des acteurs.....	32
---	----

III.3.2 Spécification des tâches.....	32
---------------------------------------	----

III.3.3 Diagramme de contexte.....	34
------------------------------------	----

III.3.4 Spécification de quelques cas d'utilisation avec scénarios.....	35
---	----

III.3.5 Représentation des Diagrammes de cas d'utilisation.....	42
---	----

III.4 Construction .....	44
--------------------------	----

III.4.1.Réalisation de l'itération 1.....	44
---	----

III.4.1.1.Diagrammes de séquence pour itération 1.....	44
--	----

III.4.1.2 Diagrammes de classe pour l'itération 1.....	47
--	----

III.4.2.réalisation de l'itération 2.....	49
---	----

III.4.2.1.Diagrammes de séquence pour itération 2.....	49
--	----

III.4.2.2. Diagrammes de classe pour l'itération 2.....	51
---	----

III.4.3. Réalisation de l'itération 3.....	51
--	----

III.4.3.1. Diagrammes de séquence pour itération 3.....	51
---	----

III.4.3.2. Diagrammes de classe pour l'itération 3.....	55
---	----

III.5. Conclusion.....	56
------------------------	----

## Chapitre IV : Réalisation et mise en œuvre

IV.1 Introduction .....	57
IV.2 Environnement de développement .....	57
IV.2.1 La plateforme J2EE .....	57
IV.2.2 Le Framework Hibernate .....	58
IV.2.2 .1 Mise en œuvre du framework hibernate .....	59
IV.2.3 Le Framework JSF/prime faces .....	64
IV.2.4 Architecture de notre application selon le modèle MVC.....	65
IV.3 Outils de développement .....	67
IV.3.1 Matériels.....	67
IV.3.1 Logiciels.....	67
IV.3.1.1 Environnement de développement intégré : Eclipse.....	67
IV.3.1.2 Le serveur d'application.....	69
IV.3.1.3 Le serveur de données.....	70
IV.4 Langages de programmation.....	76
IV.5 Présentation des interfaces de l'application.....	77
IV.6 Conclusion.....	84

**Conclusion générale.....85**

**Bibliographie**

**Annexe A UML**

**Annexe B Présentation de RUP**

# Liste des figures

# Liste des figures

---

## Chapitre I : Réseaux, Internet et web

**Figure I.1** Classification des réseaux.

**Figure I.2** Topologie en bus.

**Figure I.3** Topologie en étoile.

**Figure I.4** Topologie en anneau.

**Figure. I.5.** Le modèle OSI.

**Figure I.6.** L'architecture.

**Figure I.7.** Architecture à deux niveaux.

**Figure I.8.** Architecture à trois niveaux.

**Figure I.9.** Architecture à multi niveaux.

## Chapitre II : Etude de l'existant

**Figure II.1 :** Organigramme de NAFTAL.

**Figure II.2** Organigramme du champ.

**Figure II.3 :** diagramme des flux

## Chapitre III : Analyse et conception

**Figure.III.1.** Diagramme de contexte

**Figure.III.2.** Le diagramme de Cas d'utilisation globale

**Figure III.3.** Le diagramme de séquence du cas d'utilisation « **Ajouter compte utilisateur** »

**Figure III.4.** Le diagramme de séquence du cas d'utilisation « **Supprimer compte utilisateur** »

**Figure III.5.** Le diagramme de séquence du cas d'utilisation « **Modifier compte utilisateur** »

**Figure.III.6.** Le diagramme de classe pour l'itération 1

**Figure III.7.** Le diagramme de séquence du cas d'utilisation « **Ajouter une station**»

**Figure III.8.** Le diagramme de séquence du cas d'utilisation « **Consulter l'inventaire des équipements** »

**Figure III.9.** Le diagramme de classe pour l'itération 2

# Liste des figures

---

**Figure III.10.** Cas d'utilisation « **Ajouter une Demande de travail (OT)** »

**Figure III.11.** Le diagramme de séquence du cas d'utilisation « **Modifier l'ordre de travail (OT)** »

**Figure III.12.** Diagramme de classe pour l'itération

## Chapitre IV : Réalisation et mise en œuvre

**Figure IV.1** Architecture Hibernate

**Figure IV.2** Le fichier hibernate.cfg.xml

**Figure IV.3** Les fichiers **xml de mapping**

**Figure IV.4** Javabeen

**Figure IV.5** La classe HibernateUtil

**Figure IV.6** l'architecture de notre application suivant le modèle MVC

**Figure IV.7** Interface de l'IDE Eclipse.

**Figure IV.8** Apache Tomcat.

**Figure IV.9** L'interface PHPmyadmin

**Figure IV.10** L'interface d'authentification

**Figure IV.11** L'interface d'accueil pour l'administrateur

**Figure IV.12** L'interface d'accueil pour le cadre technique

**Figure IV.13** L'interface d'accueil pour le chef d'équipe d'intervention

**Figure IV.14** L'interface A-propos

**Figure IV.15** L'interface de création d'un nouvel utilisateur

**Figure IV.16** L'interface d'accueil pour le chef d'équipe d'intervention équipement

**Figure IV.17** L'interface de modification d'un équipement

# **Introduction Générale**

# Introduction générale

---

Au cours de ces dernières décennies, les réseaux d'entreprises ont connus une grande croissance. Ainsi, on assiste à une utilisation accrue d'application sur le web permettant d'accéder à l'information à tout moment et n'importe où, avec une connexion Internet. Cela est rendu possible grâce aux progrès technologiques dans le domaine de l'informatique. C'est ainsi que Internet est venu révolutionner ce domaine. Cet outil indispensable est accessible au grand public. Aujourd'hui, vu l'intérêt croissant de vouloir gagner en temps, de conserver les données, de limiter le nombre d'employés et pas mal d'autres raisons, ont poussés petites, moyennes et grandes entreprises à chercher des solutions informatiques capables de répondre à leurs besoins.

Le Web comme premier en la matière de l'information et de communication est vite devenu le domaine de commerce et de publicité pour ces entreprises. Le travail qui nous a été confié par le District Commercialisation de Tizi Ouzou consiste à développer une application web pour la gestion de la maintenance pour la section maintenance réseau de l'entreprise NAFTAL de Tizi Ouzou.

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

- ❖ Le premier chapitre s'intitule « **Réseaux, Internet et web** »  
Comprend des généralités sur les réseaux, Internet et quelques services offerts par le web.
- ❖ 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 s'intitule « **Analyse et conception** »: consacré à l'analyse et à la conception de notre application.
- ❖ Le cinquième et dernier chapitre s'intitule « **Réalisation et mise en œuvre** »: consacré à l'analyse à la présentation de l'environnement dont lequel notre application a été réalisée, les outils utilisés et quelques interfaces de notre application

L'objectif de notre application est de permettre au cadre technique qui est le chef de la section maintenance réseau et au chef d'équipe d'intervention une meilleure gestion des différentes tâches et difficultés auquel ils sont affrontés quotidiennement.

# **Chapitre I**

**Réseaux, Internet et web.**

**I.1 Introduction :**

La croissance constante des réseaux en général et de l'Internet en particulier constitue l'un des phénomènes les plus intéressants et passionnants dans le domaine des réseaux. Il y a une vingtaine d'années l'Internet était un projet de recherche qui concernait quelques dizaines de sites. Il est devenu aujourd'hui un système de communication indispensable.

Ce chapitre a pour objectif d'expliquer le fonctionnement d'Internet en commençant par une présentation des inter-réseaux et leurs architectures ainsi que le protocole de communication TCP/IP sur lequel se fonde le fonctionnement de l'Internet et ses différents services, notamment le World Wide Web qui est considéré comme une démocratisation d'Internet sans oublier l'architecture client/serveur sur laquelle se base son fonctionnement.

**I.2 Les réseaux**

**I.2.1 Définition du réseau :**

Un réseau informatique est un ensemble interconnecté d'ordinateurs autonomes permettant d'échanger des informations sous forme de données numériques. Cette connexion ne nécessite pas forcément un câble en cuivre, mais elle peut être réalisée par ondes courtes ou par satellite.

**I.2.2. Objectifs des réseaux :**

Les réseaux informatiques sont faits pour résoudre beaucoup de problèmes

- ✚ Partage des ressources entre les machines.
- ✚ Augmentation de la fiabilité.
- ✚ Rendre les coûts moins chers grâce au partage des données et des périphériques.
- ✚ Facilite la communication et l'échange des informations entre les personnes qui sont éloigné géographiquement.
- ✚ La garantie de l'unicité et de l'universalité de l'accès à l'information (base de données en réseau).
- ✚ Communication entre les processus (exemple : entre des ordinateur industriel).

**I.2.3 Classification des réseaux :**

**I.2.3.1 Classification des réseaux selon l'étendu :**

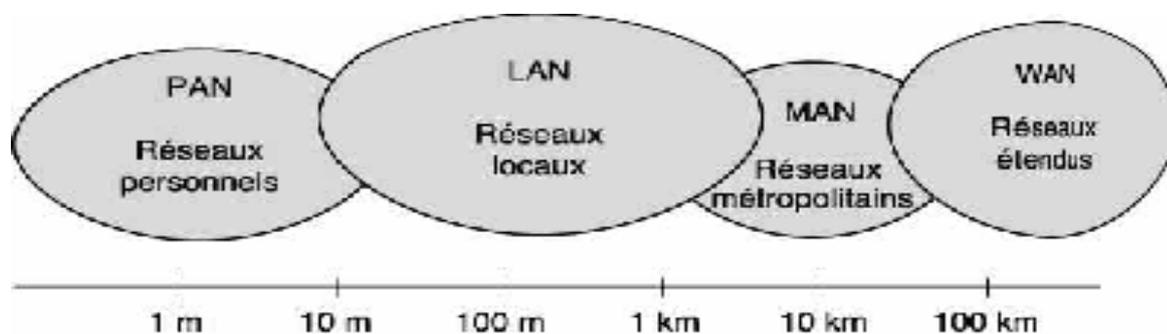


Figure I.1 Classification des réseaux.

Les réseaux en général sont classés selon l'étendue et on a :

✚ **PAN** : Personal Area Network

Ces réseaux personnels interconnectent sur quelque mètre les équipements personnels tels que le portable...

✚ **LAN** : Local Area Network

Un réseau local s'étend de quelque mètre à quelques kilomètres, correspondent par leur taille aux réseaux intra entreprise. Ils servent au transport de toutes les informations numériques de l'entreprise. La technologie la plus répandue dans ce type est Ethernet.

✚ **MAN** : Metropolitan Area Network

Les réseaux métropolitains permettent l'interconnexion des entreprises ou éventuellement des particuliers sur un réseau spécialisé à haut débit qui est géré à l'échelle d'une métropole.

✚ **WAN** : Wide Area Network

Les réseaux étendus sont destinés à transporter des données numériques sur des distances à l'échelle d'un pays ou de la planète entière.

### I.2.3.2 Classification des réseaux selon la topologie :

Une topologie de réseau en informatique est une définition de l'architecture d'un réseau. Elle donne une certaine disposition des différents postes informatiques du réseau et une hiérarchie de ces postes, on reconnaît trois topologies les plus utilisées, qui sont :

#### I.2.3.2.1 La topologie en bus :

Toute information diffusée sur un noeud sera propagée sur l'ensemble du bus pour atteindre les autres noeuds qui reconnaîtront chacun l'information qui lui est destinée en reconnaissant son adresse. L'inconvénient de cette topologie est la difficulté de reconnaissance d'une rupture dans le réseau. Et son principal avantage est la simplicité de son installation et si une station tombe en panne elle ne perturbe pas le réseau.

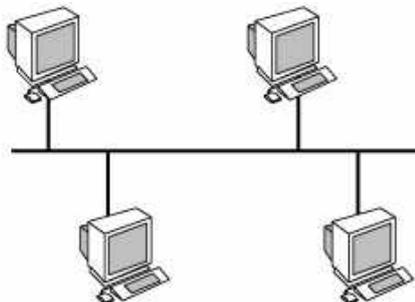


Figure I.2 Topologie en bus.

### I.2.3.2.2 La topologie en étoile :

C'est la topologie la plus courante actuellement. Omniprésente, elle est aussi très souple en matière de gestion et dépannage de réseau : la panne d'un noeud ne perturbe pas le fonctionnement global du réseau. En revanche, l'équipement central (un concentrateur (hub) et plus souvent sur les réseaux modernes, un commutateur (switch)) qui relie tous les noeuds constitue un point unique de défaillance : une panne à ce niveau rend le réseau totalement inutilisable. Le réseau ethernet est un exemple de topologie en étoile. L'inconvénient principal de cette topologie réside dans la longueur des câbles utilisés.

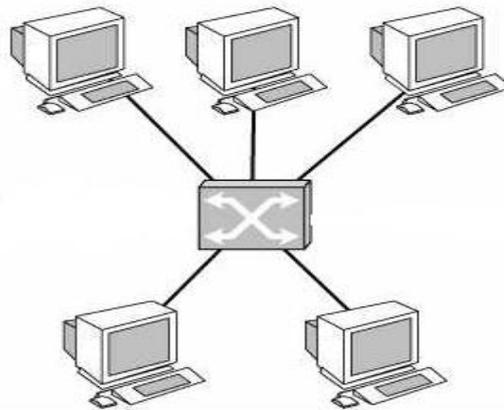


Figure I.3 Topologie en étoile.

### I.2.3.2.3 La topologie en anneau :

Dans cette topologie, tous les noeuds sont disposés sur une boucle fermée pour former un anneau qui se compose d'une série de liaisons entre les différents noeuds du réseau. L'inconvénient majeur de cette topologie est que le réseau devient inopérant en cas de rupture du câble. Et à cause du rôle actif de toutes les stations dans la transmission des informations une seule panne dans une de ces stations provoquera l'arrêt de la propagation.

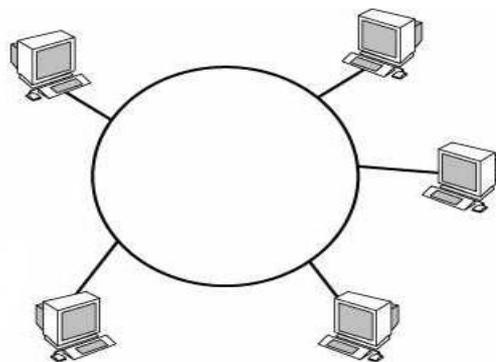


Figure I.4 Topologie en anneau

### I.2.4 Architecture des réseaux :

Le transport des données d'une extrémité à l'autre d'un réseau nécessite un support physique de communication. Cependant, pour que ces données arrivent correctement au destinataire, avec la qualité de service exigée, il faut une architecture logicielle. L'ensemble de protocoles nécessaires constitue une architecture. Un protocole est l'ensemble de règles qui définissent les modalités de fonctionnement d'une communication entre deux ordinateurs.

#### I.2.4.1 L'architecture OSI (Open Systems Interconnection):

A fin de mettre en place un standard de communication entre les ordinateurs d'un réseau et pour faciliter l'interconnexion, un modèle dit d'interconnexion des systèmes ouverts, appelé encore OSI (Open Systems Interconnexion) a été mis en place par l'ISO (Interconnexion l'Standards Organisation).

Le modèle OSI répartit les protocoles utilisés selon sept couches, définissant ainsi un langage commun pour le monde des télécommunications et de l'informatique. Chaque couche regroupe des dispositifs matériels (dans les couches basses) ou logiciels (dans les couches hautes). Entre couches consécutives sont définies des interfaces sous forme de primitives de service et d'unités de données rassemblant les informations à transmettre et les informations de contrôle rajoutées.

Voici l'architecture de ce modèle :

<b>Couche 7: Application</b>
<b>Couche 6 : Présentation</b>
<b>Couche 5 : Session</b>
<b>Couche 4 : Transport</b>
<b>Couche 3 : Réseau</b>
<b>Couche 2 : Liaison</b>
<b>Couche 1 : Physique</b>

**Figure. I.5.** Le modèle OSI

**Les 7 couches de l'OSI**

- **Couche 1 : physique**

La couche physique rassemble les moyens électriques, mécaniques, optiques ou hertziens par lesquels les informations sont transmises. Les unités de données sont donc des bits 0 ou 1. donc l'objectif de la conception de cette couche est d'assurer une transmission fiable d'un bit d'une extrémité à une autre.

- **Couche 2 : liaison**

La couche liaison gère la fiabilité du transfert de bits d'un nœud à l'autre du réseau, comprenant entre autres les dispositifs de détection et de correction d'erreurs. L'unité de données à ce niveau est appelée trame.

- **Couche 3 : réseau**

La couche réseau définit l'unité de données de base transférée sur un réseau entre deux sites extrêmes et inclus les concepts de routage. Plus généralement cette couche assure la qualité de service (délais, temps de transit) L'unité de données s'appelle en général un paquet.

- **Couche 4 : transport**

La couche transport regroupe les règles de fonctionnement de bout en bout en permettant à un processus destinataire de communiquer directement avec le processus source, assurant ainsi la transparence du réseau vis-à-vis des couches supérieures. Elle traite notamment l'adressage, l'établissement des connexions et la fiabilité du transport.

- **Couche 5 : session**

La couche session définit la manière dont les protocoles sont organisés pour fournir toutes les fonctionnalités dont les programmes d'applications se servent. Alors elle réunit les procédures de dialogue entre les applications ainsi la gestion des transactions.

- **Couche 6 : présentation**

La couche présentation traite les formes de représentation des données, permettant la traduction entre machines différentes car elle s'intéresse à la syntaxe et à la sémantique des informations elle est destinée à supporter les fonctions dont beaucoup de programme ont besoin tel que la compression de texte, la conversion d'image graphique.

- **Couche 7 : application**

Source et destination de toutes les informations à transporter, la couche application rassemble toutes les applications qui ont besoin de communiquer par le réseau : messagerie électronique, transfert de fichiers, gestionnaire de bases de données, etc.

❖ **Avantages du modèle OSI :**

- Réduit la complexité, puisque cela subdivise la communication en plus petites couches.
- Standardise bien sur les interfaces.
- Permet un meilleur développement et une meilleure évolution, car il suffit d'interagir sur la couche qui doit être modifiée.

**I.2.4.2 L'architecture TCP/IP :**

Le modèle TCP/IP s'inspire du modèle OSI auquel il reprend l'approche modulaire mais réduit le nombre à quatre. Les trois couches supérieures du modèle OSI sont souvent utilisées par une même application. Ce n'est pas le cas du modèle TCP/IP. C'est actuellement le modèle théorique le plus utilisé.

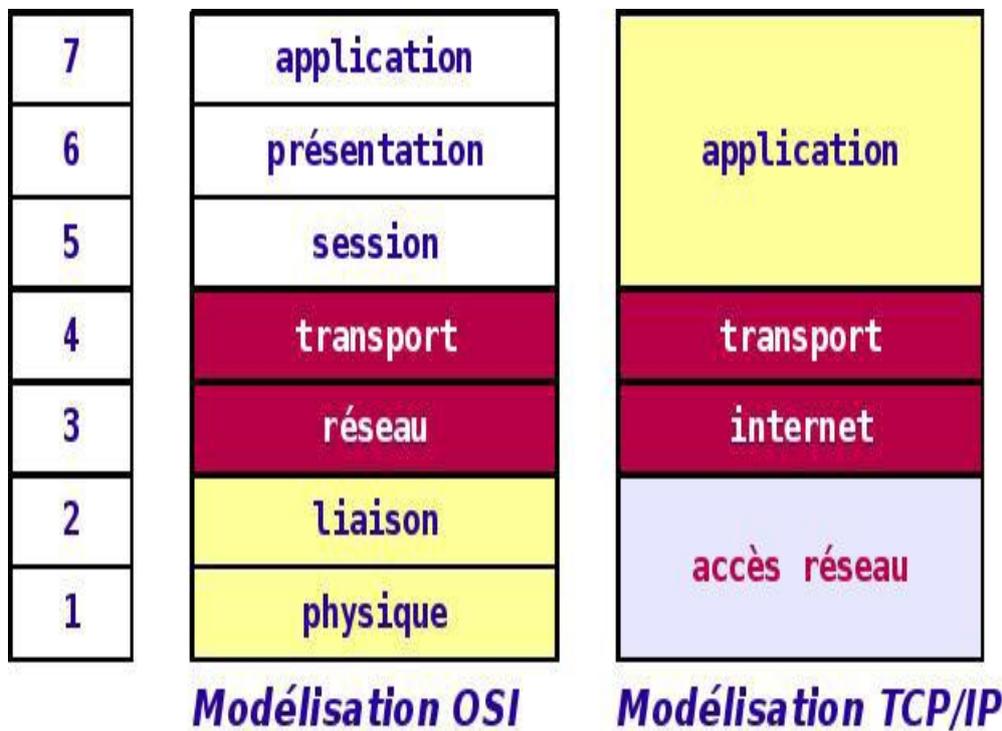


Figure I.6 L'architecture TCP/IP

❖ **La couche application :**

Elle englobe les applications standard du réseau :

- ✚ **SMTP (Simple Mail Transport Protocol) :** permet d'échanger du courrier entre deux serveurs de messagerie.

- ✚ **TELNET** : Protocole permettant de se connecter sur une machine distante (serveur) en tant que utilisateur.
- ✚ **FTP (File Transfert Protocol)** : Est un protocole permettant d'échanger des fichiers via Internet.
- ✚ **DNS (Domain Name Service)** : Service de nom du domaine. Il permet de convertir le nom d'une machine en une adresse réseau (ou IP) et vice versa.

#### ❖ **La couche transport :**

Elle assure l'acheminement des données sur le réseau local et les mécanismes permettant de connaître l'état de la transmission. Les protocoles de cette couche permettent d'envoyer des informations d'une machine à une autre.

- ✚ **TCP** : Il assure le contrôle de données, orienté connexion.
- ✚ **UDP (User Datagram Protocol)** : Archaïque et non orienté connexion, il n'assure aucun contrôle de transmission de données.

#### ❖ **Couche Internet :**

La couche Internet est chargée de fournir le paquet de données. Elle contient cinq protocoles, les trois premiers sont les plus importants :

- ✚ **IP** : gère les destinations des messages, adresse du destinataire.
- ✚ **ARP (Address Resolution Protocol)** : gère les adresses des cartes réseau. Chaque carte a sa propre adresse d'identification codée sur 48 bits.
- ✚ **ICMP (Internet Control Message Protocol)** : gère les informations relatives aux erreurs de transmission. Il ne corrige pas les erreurs, mais signale aux autres couches que le message contient des erreurs.
- ✚ **RARP (Reverse Address Resolution Protocol)** : gère l'adresse IP pour les équipements, qui ne peuvent s'en procurer une, par lecture d'information dans un fichier de configuration. En effet, lorsque le PC démarre, la configuration réseau lit l'adresse IP qu'elle va utiliser. Ceci n'est pas possible dans certains équipements qui ne possèdent pas de disques durs (imprimantes, terminaux).
- ✚ **IGMP (Internet Group Management Protocol)** : permet d'envoyer le même message à des machines faisant parti d'un groupe. Ce protocole permet également à ces machines de s'abonner ou de se désabonner d'un groupe. Ceci est utilisé par exemple dans la vidéo conférence à plusieurs machines, envoi de vidéos,...

### I.2.4.1.3 Architecture client/ serveur :

Dans un monde où la course à la productivité conduit les technologies à évoluer de plus en plus vite, le client/serveur s'est imposé comme une architecture puissante. Dans cette partie, nous proposons une vue d'ensemble sur l'architecture client/serveur tout en définissant les notions de base indispensables à la compréhension du modèle.

- **Définition du client/serveur :**

L'architecture **client/serveur** désigne un mode de communication entre plusieurs ordinateurs d'un réseau qui distingue un ou plusieurs clients du serveur : chaque logiciel client peut envoyer des requêtes à un serveur. Un serveur peut être spécialisé en serveur d'applications, de fichiers, de terminaux, ou encore de messagerie électronique.

- **Client :** Caractéristiques d'un client :

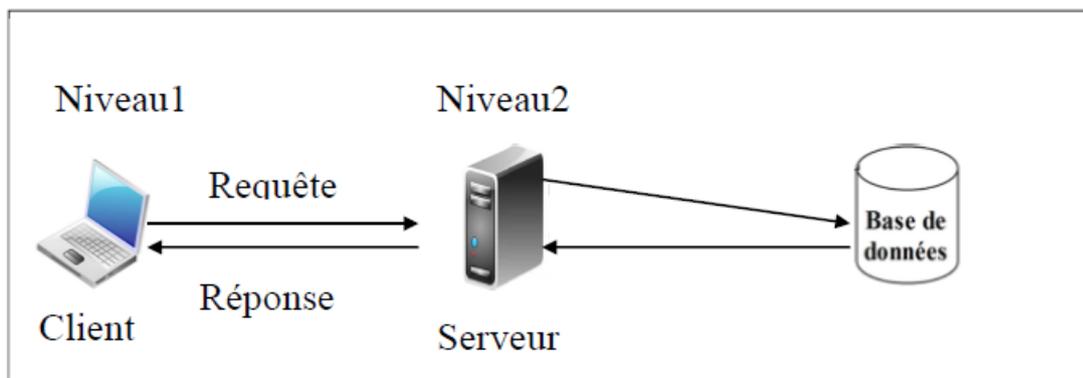
- il est actif le premier (ou maître) ;
- il envoie des requêtes au serveur ;
- il attend et reçoit les réponses du serveur.

- **Serveur :** Caractéristiques d'un serveur :

- 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.
- Un serveur est généralement capable de servir plusieurs clients simultanément.

- **Les architectures client/serveur :** On distingue deux architectures :

- **Architecture à deux niveaux :** L'architecture à deux niveaux (aussi appelée architecture 2-tiers, tier 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. (voir Figure I.6).



**Figure I.7.** Architecture à deux niveaux

- **Architecture à trois 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.

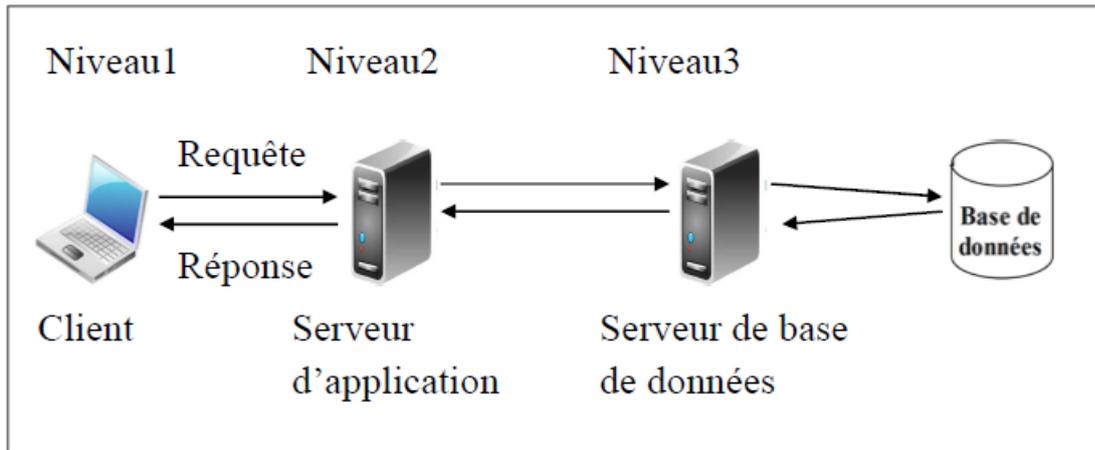


Figure I.8. Architecture à trois niveaux

- **Architecture à multi niveaux** : Dans l'architecture à 3 niveaux, chaque serveur 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...

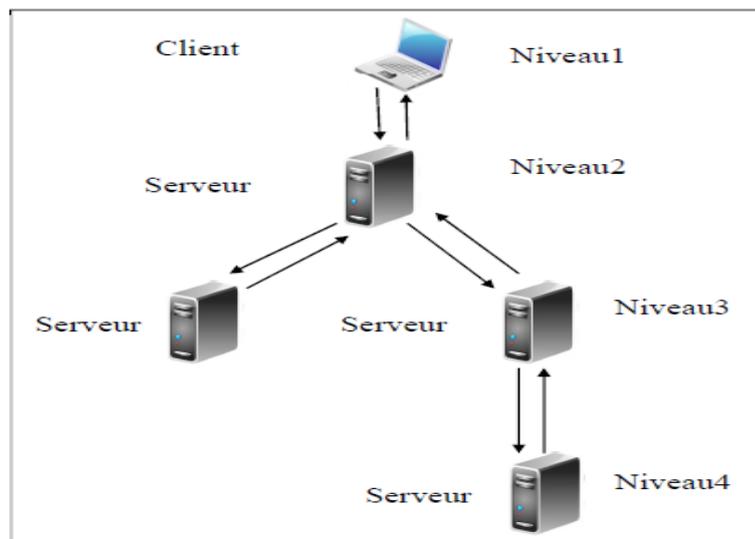


Figure I.9. Architecture à multi niveaux

- **Avantages de l'architecture client/serveur :**

- Maintenance matériel minimale.
- Grande vélocité sur des grands volumes de données et de traitements.
- Toutes les données sont centralisées sur un seul serveur, ce qui simplifie les contrôles de sécurité, l'administration, la mise à jour des données et des logiciels.
- Les technologies supportant l'architecture client-serveur sont plus matures que les autres.

### **I.1.5 La sécurité des réseaux :**

La sécurité des réseaux est une opération impérative et importante pour protéger les machines au sein du réseau des utilisations malveillantes.

Ce processus de sécurité se résume par la réalisation des actions suivantes :

- ✓ Empêcher les personnes non autorisées à effectuer des opérations malveillantes dans le système.
- ✓ Sécuriser les données du système ainsi garantir sa continuité.
- ✓ Protéger les différents systèmes des attaques extérieures comme le piratage et l'espionnage.

Chaque ordinateur connecté dans un réseau quelconque est exposé à ces différentes menaces afin de les protéger, il existe plusieurs mécanismes de protection comme le **(PAR FEU)** qui sert de passerelle filtrante et qui protège les ordinateurs des intrusions émanant des réseaux.

### **I.3 Internet :**

#### **I.3.1 Définition :**

Internet est le réseau informatique mondial qui rend accessibles au public des services variés comme le courrier électronique, la messagerie instantanée et le World Wide Web, en utilisant le protocole de communication IP (internet protocol). Son architecture technique qui repose sur une hiérarchie de réseaux lui vaut le surnom de réseau des réseaux.

Internet ayant été popularisé par l'apparition du World Wide Web, les deux sont parfois confondus par le public non averti. Le World Wide Web n'est pourtant que l'une des applications d'Internet.

L'accès à Internet peut être obtenu grâce à un fournisseur d'accès à Internet via divers moyens de télécommunication : soit filaire (réseau téléphonique commuté (bas débit), ADSL, fibre optique jusqu'au domicile...), soit sans fil (WiMAX, Internet par satellite, 3G+...). Un utilisateur d'Internet est désigné en français par le néologisme « internaute ».

#### **I.3.2 Intérêt de l'internet :**

- ❖ Internet permet aux utilisateurs d'accéder à des centaines de milliers de giga-octets de données, sous forme de textes, d'images, de vidéos et de sons. Il permet également de communiquer entre eux.
- ❖ Le réseau est abondamment utilisé par les scientifiques qui ont constamment besoin d'échanger des informations.
- ❖ Internet intéresse également les entreprises de toute taille en tant qu'outil de travail (travail à distance) et en tant qu'outil commercial (diffusion d'informations commerciales, messagerie pour correspondre les clients et les fournisseurs, de façon plus efficace et plus économique qu'avec la télécopie et le téléphone).

### I.3.3 Les services d'internet :

Les principaux services internet sont liés à la communication.

#### + **Mail :**

La messagerie électronique (*e-mail* ou encore *mail*), c'est utiliser l'Internet comme on utilise la poste. Il est possible de déposer un message dans la boîte aux lettres de son correspondant, qu'il soit ou non devant une machine. Ce dernier sera capable, à sa prochaine connexion, de consulter sa boîte aux lettres pour lire ou envoyer des messages à ses correspondants.

#### + **Ftp :**

Le service d'échange de fichiers permet de déposer des fichiers sur une machine distante, mais aussi, et c'est le plus fréquent de télécharger des fichiers sur sa machine. La distribution de logiciels gratuits, la diffusion d'images, de sons, de notes de cours pour les étudiants ou d'articles scientifiques sont parmi les utilisations les plus courantes de ce service.

#### + **News :**

À l'inverse du *mail* où la discussion est réalisée de 1 à 1, de l'émetteur vers le destinataire, les forums de discussion (ou *News*) sont des moyens de discussion entre plusieurs personnes. La métaphore la plus précise est celle du kiosque à journaux, à la seule différence que tous les utilisateurs d'Internet sont des journalistes potentiels. Tous sont capables de rédiger un article qui pourra être diffusé dans le groupe de discussion de son choix. Chacun peut lire le (ou s'abonner au) groupe de discussion qu'il désire. Il existe des groupes de discussion sur tout et n'importe quoi. Des utilisateurs de Windows par exemple aux fanatiques des séries télévisées. C'est souvent dans ce cadre que se sont posées des questions éthiques quant à l'utilisation d'Internet, lorsque se forment des groupes au sujet par exemple du racisme ou de la pédophilie.

#### + **Telnet et ssh :**

Ces services permettent à un internaute (utilisateur d'Internet) de se connecter et donc d'utiliser à distance une machine comme si il se trouvait face à elle. Cela ouvre par exemple des possibilités pour le travail à domicile, puisqu'il devient possible d'utiliser les machines se trouvant sur son lieu de travail depuis chez soi. Telnet est plus ancien, et commence à être abandonné au profit de ssh pour des raisons de sécurité: avec Telnet toutes les informations transmises transitent en clair sur le réseau, y compris les mots de passe, ce qui permet à un éventuel pirate simplement à l'écoute du réseau d'intercepter toutes les données sensibles qui transiteraient de cette manière. À l'inverse pour ssh toutes les données transmises sont cryptées et donc illisibles par toute autre personne que le destinataire.

#### + **Web :**

Le service le plus connu, le plus récent et maintenant le plus utilisé de consultation d'hyperdocuments. C'est le service d'Internet. C'est ce service qui l'a rendu attrayant, et «commerciallement utile».

## I.4 Le web :

### I.4.1 Définition :

Le Web ou le World Wide Web (WWW) est un système hypertexte qui rend au public une lecture plus dynamique, fonctionnant sur l'Internet. Le principe du web repose sur l'utilisation des hyperliens qui lient les pages web entre elles, pour naviguer entre des documents multimédias (pages de textes enrichies de sons, graphiques, images fixes et animées, vidéos,...) grâce à un logiciel appelé navigateur.

Ce dernier permet d'effectuer une demande par l'intermédiaire d'une adresse (URL) et de lire le résultat de cette recherche.

### I.4.2 Les concepts du web :

- **URL :**(Uniform Resource Locator)

C'est l'adresse qui permet de localiser une ressource ou un document sur Internet et qui indique la méthode pour y accéder (le nom du serveur et le chemin à l'intérieur du serveur exemple <http://www.culture.gouv.fr/dglf/acceuil.htm>).

- **client web:**

Le client web, appelé aussi navigateur ou browser, peut être soit un PC, soit un macintosh ou une plate forme UNIX. Les requêtes sont lancées à partir du client web, la réponse qui est constitué d'un document HTML sera analysée par le client web puis affichée à l'utilisateur.

De point de vu de l'utilisateur, la connexion s'effectue de façon totalement transparente.

- **serveur web :**

Le serveur web est un processeur (programme) capable de traiter les requêtes http issues des clients web, c'est-à-dire des navigateurs. Un système web est appelé serveur http, ce processus est présent sur un serveur physique (une machine) accessible par les ordinateurs du réseau via TCP/IP. Ce serveur physique contient les informations qui peuvent être mises a disposition des utilisateurs sous forme de fichier HTML.

- **Site web :**

Un site Web est un ensemble de pages qui peuvent être consultées en suivant des hyperliens à l'intérieur du site. L'adresse Web d'un site correspond en fait à l'URL d'une page Web, prévue pour être la première consultée : la page d'accueil. La consultation des pages d'un site s'appelle une « visite », car les hyperliens devraient permettre de consulter toutes les pages du site sans le quitter (sans devoir consulter une page Web hors du site). Une visite peut commencer par n'importe quelle page, particulièrement lorsque son URL est donnée par un moteur de recherche. Techniquement, rien ne distingue la page d'accueil d'une autre page.

- **Site web statique :**

Un site Web statique est un site où chacune des pages est créée en HTML. Un ordinateur qui se connecte au serveur, demande une page, celle-ci lui est directement servie (elle est stockée toute prête sur le serveur).

- **Site web dynamique :**

Un site Web dynamique est un site dont les pages sont générées dynamiquement à la demande. Le contenu est obtenu (par exemple) en combinant l'utilisation d'un langage de scripts ou de programmation et une base de données. Il s'agit souvent de PHP pour le langage et MySQL pour la base de données.

- **Page web :**

Une page Web est une ressource du World Wide Web conçue pour être consultée par des visiteurs à l'aide d'un navigateur Web. Elle a une adresse Web. Techniquement, une page Web est souvent constituée d'un document en Hypertext Markup Language (HTML) (ou XHTML) et d'images. Cependant, tout type de ressources ou d'assemblage de ressources, textuelles, visuelles, sonores, logicielles, peuvent constituer une page Web.

- **Un hébergeur Web :**

Entreprise de services informatiques hébergeant (mettant en ligne) sur ses serveurs Web les ressources constituant les sites web de ses clients

- **Surfer, Naviguer :**

Parcourir le Web en suivant les hyperliens, de page Web en page Web.

- **Moteur de recherche :**

Un moteur de recherche est un serveur qui indexe un certain nombre de sites web et permet généralement de rechercher les informations qui intéressent l'internaute à l'aide de mots clés.

- **Hyperlien :**

Lien logique, unidirectionnel (codé par une balise HTML) reliant une page Web à une autre. Se sont des mots ou des phrases (soulignés souvent de couleur différente) ou bien des icônes, boutons ou images qui peuvent servir de marques activables pour concrétiser les hyperliens. En cliquant sur une telle zone on active le lien qui désigne l'adresse d'un document qui se trouve sur un serveur web. Un lien peut être externe au document ou bien interne.

Un hyperlien interne se situe dans le même document dont l'intérêt est de faciliter l'accès à une autre partie de ce document.

Un hyperlien externe c'est un lien envers un document qui se trouve sur le même serveur ou bien sur serveur distant.

### I.4.3 Les langages du web :

Les langages utilisés par le Web peuvent être classés en deux grandes catégories

- **Coté client :**

- **HTML (Hyber Text Mark-up Language) :** C'est le langage qui décrirait la structure des documents du monde Web. Et permet de créer des pages Web, il utilise une structure formé avec des balises permettant la mise en forme du texte. Nécessite un navigateur web pour la visualisation.

- **XML (Extensible Mark up Langage) :** C'est un langage simple et élégant pour la création de documents auto descriptifs. Utilise des balises (Mark up) pour décrire des données afin que d'autres application (ou outils) puissent les lire et les traiter et il s'occupe également de la standardisation du langage HTML et des feuilles de style entre autres.

- **Applets :** Petite application, écrite généralement en langage java, envoyée par un serveur web et exécutée par un navigateur.

- **Java script :** Le Java script est un langage de script incorporé dans dans un document HTML. Historiquement il s'agit même du premier langage de script pour le Web. Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client.

 **Coté serveur :**

- **CGI (Common gateway interface) :** Spécification permettant de définir une interface de programmation utilisée pour permettre à un serveur web de communiquer avec un programme permettant d'effectuer des traitements de données.
- **API (Application programmable interface) :** C'est un ensemble de fonctions permettant d'accéder aux services d'une application, par l'intermédiaire d'un langage de programmation.
- **ASP (Active Server Page) :** est un standard mis au point par Microsoft en 1996 permettant de développer des applications web interactives, c'est-à-dire dont le contenu est dynamique. Ainsi une page web ASP (fichier repérable par l'extension.asp) aura un contenu pouvant être différent selon certains paramètres (des informations stockées dans une base de données, les préférences de l'utilisateur, ...) tandis qu'une page web « classique » (dont l'extension est .html ou .htm) affichera continuellement la même information.
- **PHP (Hypertext PerProcessor) :** C'est langage de programmation orienté objet permettant la programmation de serveur web. Il Permet également de créer des pages( ou sites ) Web dynamiques en gérant une base de données comme MYSQL.
- **Servlet :** Les servlets sont des applications Java car elles permettent de gérer des requêtes http et de fournir aux clients une repense http dynamique.

**I.5 Conclusion :**

Dans ce chapitre, nous avons donné quelques notions générales sur les interconnexions réseaux, le modèle OSI ainsi que le protocole TCP/IP puis nous avons présentés Internet et ses services. Pour expliquer le fonctionnement des applications réseaux il a fallut définir et expliquer l'architecture client/serveur ainsi que les notions indispensables à la compréhension de ce modèle, comme on a aussi parlé du web et ses différentes caractéristiques.

Dans le chapitre suivant, nous allons essayer de présenter l'organisme d'accueil et de mettre en évidence notre problématique ainsi que les objectifs de notre travail.

# Chapitre II

*Etude de l'existant*

**II.1 Introduction :**

Dans le chapitre précédent, on a fait un récapitulatif sur les concepts importants des réseaux informatique ainsi que l'internet et le web, toutes ces notions sont indispensables dans notre projet afin de réaliser une application web. Elles nous seront utiles après la réalisation de notre application.

Dans ce chapitre nous allons présenter généralement l'entreprise en spécifiant ses différents services et activités. Cette partie de présentation nous permet de mieux comprendre le fonctionnement de l'organisme d'accueil dans lequel notre pratique se fera à son niveau.

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

**II.2 Présentation de l'entreprise :**

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

**II.2.1 Historique de L'entreprise :**

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 le 1<sup>er</sup> 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.

**II.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, NAFTAL est chargé de la commercialisation et de développement économique et social, NAFTAL est chargé de la commercialisation et de distribution des produits pétroliers y compris ceux destinés à l'aviation et à la marine, le GPL, les combustibles, les solvants, es 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, CHERAGA, ALGER.

### **II.2.3 Mission 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, l'approvisionnement des entrepôts à partir des raffineries.
- ✚ La route pour livraison des clients et le ravitaillement des dépôts non desservis par le rail.

### **II.2.4 Les objectifs de NAFTAL :**

A travers son plan de développement, NAFTAL vise plusieurs objectifs :

- ❖ Poursuivre la de distribution des produits pétroliers.
- ❖ Améliorer la qualité des produits et services proposés.

Le développement durable reste cependant un des défis majeurs pour NAFTAL. C'est la raison pour laquelle l'entreprise s'est fixé un ensemble d'objectifs visant à instaurer les principes de la pratique responsable et citoyenne :

- ❖ Diminuer les rejets liquides et gazeux ainsi que la production de déchets.
- ❖ Promouvoir le carburant écologique GPL/c.
- ❖ Améliorer la sécurité industrielle.
- ❖ Diminuer la consommation énergétique.
- ❖ Promouvoir l'image de « l'entreprise verte ».
- ❖ Réduire les accidents de route incluant les camions de transports NAFTAL.

### **II.2.5 Moyens :**

Avec un personnel de 30 000 agents, Natale est le premier distributeur de produits pétroliers en Algérie.

Elle contribue à hauteur de 51% de l'énergie finale en fournissant 10 millions de tonnes de produits pétroliers par an sous forme de :

- ❖ Carburants (8 millions de TM).
- ❖ Gaz de pétrole liquéfiés (plus de 1.6 million de TM).
- ❖ Bitumes (plus de 0.5 million de TM).

- ❖ Lubrifiants (plus de 70 000 TM).

Représentant pour l'exercice 2007 :

- ❖ Un chiffre d'affaire toutes taxes comprises de 208 milliards de DA.
- ❖ Une valeur ajoutée de 30 milliards de DA.
- ❖ Un résultat d'exploitation de 6 milliards de DA.

Pour cela elle dispose de :

- ❖ 49 centres et dépôts de distribution et de stockage de carburants.
- ❖ 22 centres et 27 magasins lubrifiants et pneumatiques.
- ❖ 26 centres et dépôts aviation, 06 centres marine.
- ❖ 49 dépôts relais de stockage GPL.
- ❖ 41 centres d'emplissage GPL d'une capacité d'enfûtage de 1.2 millions tonnes/an.
- ❖ 3 centres vrac GPL.
- ❖ 15 unités bitumes d'une capacité de formulation de 360.000 tonnes/an. - 3 000
- ❖ Véhicules de distribution et 800 engins de manutention et de maintenance.
- ❖ 730 Km de canalisation

Et son réseau de distribution s'étend sur :

- ❖ 1 952 stations service dont 671 en toute propriété
- ❖ 7925 points de vente GPL.

II.2.6. Organigramme général de l'entreprise

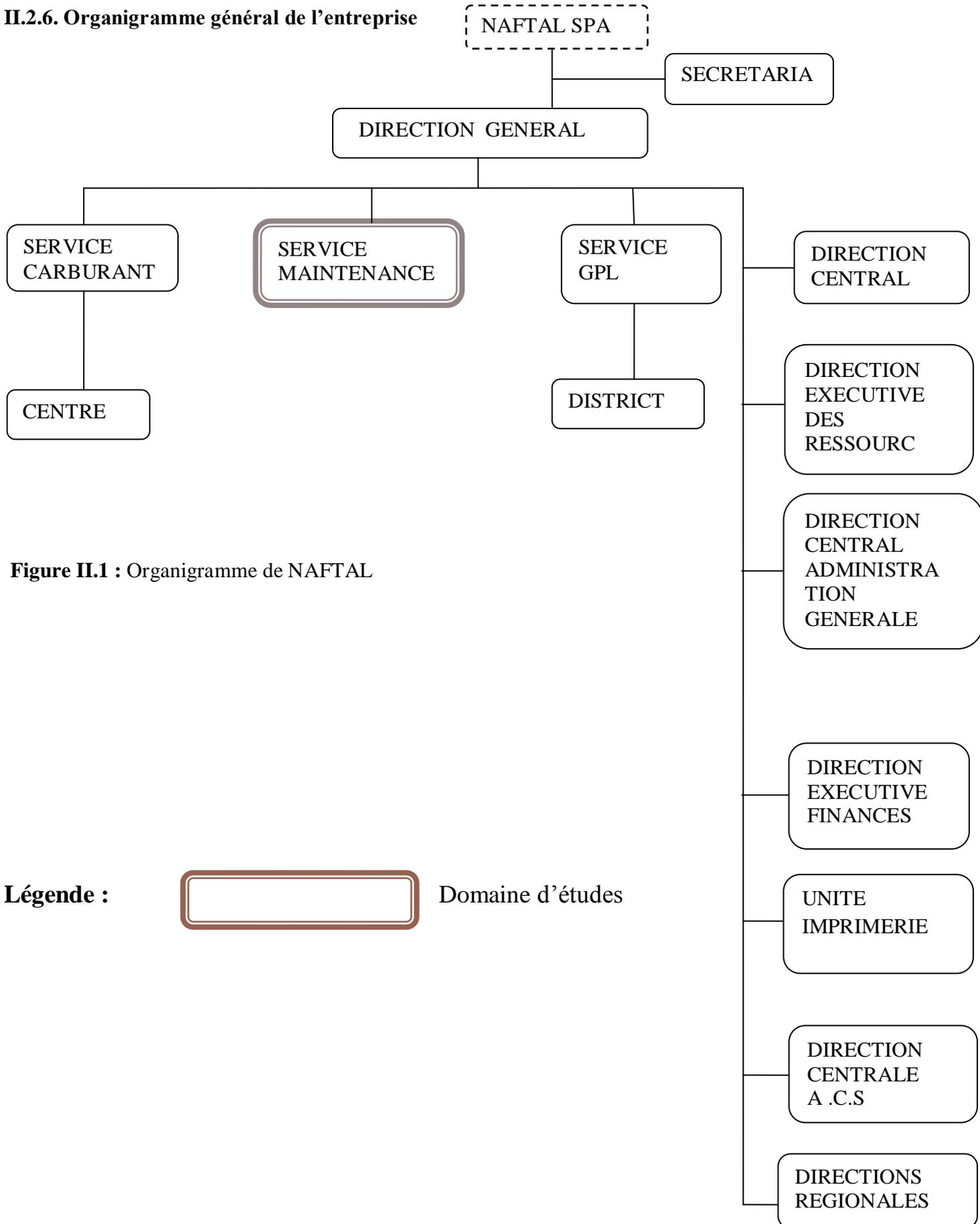


Figure II.1 : Organigramme de NAFTAL

Légende :

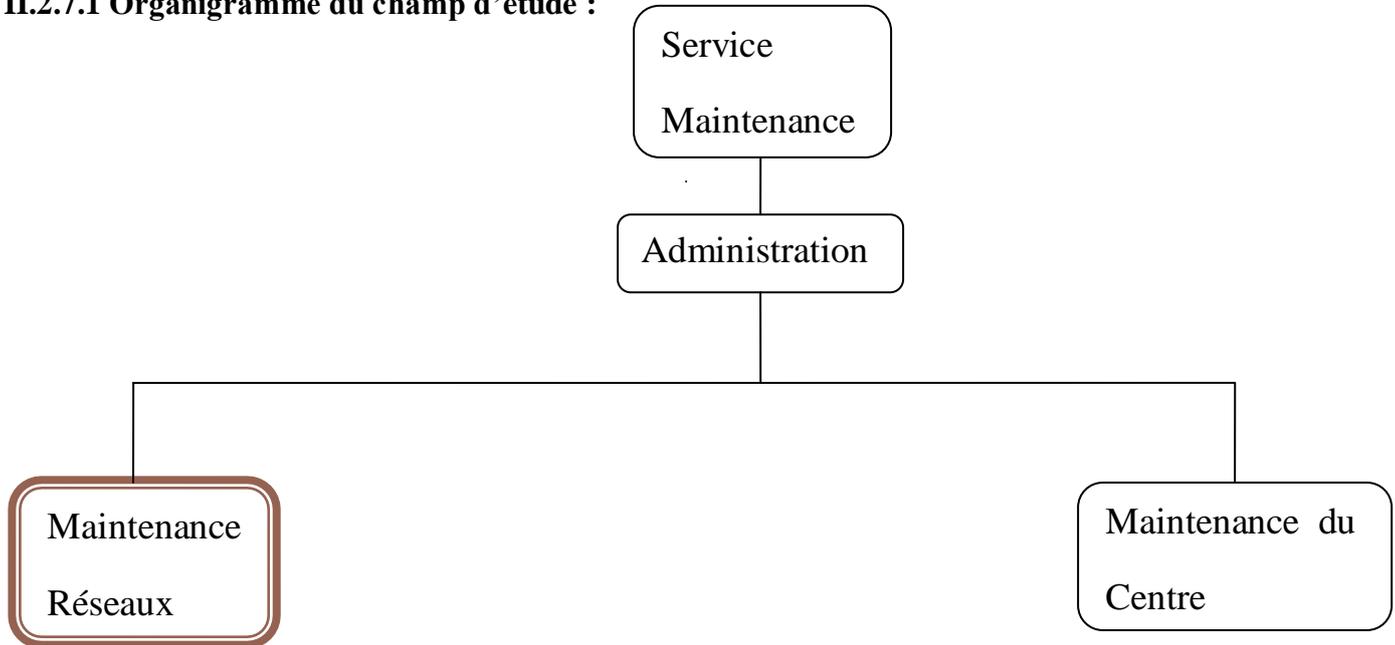


Domaine d'études

**II.2.7 Présentation du champ d'étude :**

Notre champ d'étude est la section maintenance réseaux, qui se trouve au niveau du service Maintenance du district Oued Aissi Tizi Ouzou.

**II.2.7.1 Organigramme du champ d'étude :**



**Figure II.2** Organigramme du champ

**Légende :**  Champs d'études

**II.2.7.2 Situation informatique du champ d'étude :**

 **Aspect matériel :**

Un micro-ordinateur HP ayant les caractéristiques suivantes :

- ❖ **System d'exploitation :** Windows 7
- ❖ **Processeur :** Intel core™ 2 Duo Cpu E6750 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 donner la journée comptable.
- ❖ **WINGMR STOCK** : utiliser pour la gestion des stocks du magasin.
- ❖ **WINGMR EQUIPEMENT** : pour la gestion des équipements des stations services.
- ❖ **IMMOSYS** : 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.
- ❖

**II.3 Etude de l'existant :****II.3.1 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.

**II.3.2 Etude des postes :**

L'étude des postes de travaux, est l'étape essentielle pour élaborer un système d'information fiable, la structure organisationnelle de notre champ d'étude se consiste d'un seul poste de travail.

L'étude se fait en suivant les caractéristiques suivantes :

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

✚ **La liste des postes de travail dans le champ d'étude :** La liste des postes n'est que le cadre technique de la section maintenance réseaux.

<b>Fiche d'étude du poste :</b>			
<p><b>Désignation du poste :</b> Cadre technique  <b>Section de rattachement :</b> Section Maintenance Réseaux  <b>Nombre d'employé :</b> 01  <b>Responsabilité :</b> Il assure l'exécution des taches de service à la limite de ces prérogatives (pouvoir)</p>			
<b>Taches du poste</b>			
<ul style="list-style-type: none"> <li>-Intervenir à n'importe quel moment dans un poste ou autre, pour un soutien ou un remplacement en cas de besoin.</li> <li>-Recueillir et interpréter les remarques et souhaits exprimés par le client.</li> <li>-S'assurer du bon fonctionnement des équipements de protection individuels.</li> <li>-Gérer son propre stock de pièces détachées.</li> <li>-Rédiger un rapport d'intervention interne ou tout autre document de d'informations</li> <li>-Identifier une situation à risque ou une situation de presque accident et remédier à la situation.</li> <li>-Faire les essais métrologiques et ajuster un mesureur de carburant.</li> <li>-Procéder à la mise en route de matériels de stations-service ainsi qu'à premier niveau du client utilisateur.</li> <li>-Réaliser des interventions de maintenance préventive</li> <li>-Réaliser des interventions de maintenance curative.</li> </ul>			
<b>Document entrant</b>	<b>Provenant</b>	<b>Documents sortants</b>	<b>Destination</b>
Demande de travaux	Département Commercial	Ordre des travaux	Equipe d'intervention
Fiche de travaux	Equipe d'intervention		
Control débit	Equipe d'intervention		

### II.3.3 Etude des documents :

Au cours de l'étape de la collecte des informations nous avons étudiés les différents documents utilisés entre divers acteurs, ainsi nous avons extrais 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 appliqué.

Pour chaque document on précisera ses caractéristiques et on le décrit avec des indications pour la réparation des informations et pour chacune, on désignera la longueur, le type et des observations sur des rubriques si elles existent et ca en utilisant une fiche.

Le type des rubriques peut être :

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

Les documents sont divisés en catégories :

- ❖ Document interne
- ❖ Document externe

**Document interne** : C'est tous les documents utilisés à l'intérieur du champ d'étude.

**Document interne de position** : sont des documents utilisé que dans le poste ou ils sont créer.

**Document interne de liaison** : sont des documents qui transitent entre

Les postes de champ d'étude :

**Document externe** : C'est tous les documents qui proviennent ou destinés à l'extérieur du champ d'étude.

#### II.3.3.1 Liste des documents existant :

<i>N<sup>o</sup></i> Document	Désignation
<b>01</b>	<b>Demande de travaux</b>
<b>02</b>	<b>Ordre de travaux</b>
<b>03</b>	<b>Contrôle débit</b>
<b>04</b>	<b>Fiche de travaux</b>

FICHE D'ANALYSE DU DOCUMENT N° 1			
Identification			
<p><b>Désignation :</b> demande de travaux  <b>Code :</b> /  <b>Rôle :</b> signaler une anomalie  <b>Nature :</b> externe  <b>Nombre d'exemplaire :</b> 01  <b>Emetteur :</b> département commercial  <b>Récepteur :</b> cadre technique</p>			
Désignation			
Rubrique	Type	Taille	Observation
- Numéro de la demande de travaux	N	11	GD : Gestion Directe GL : Gestion Lire RO : Revendeur Ordinaire PVA : Point de Vente Agrée
-Client	A	03	
- Demande faite par le P.V	A	20	
-CODE	AN	05	
-Etabli le	D	08	
-Localité	AN	30	JJ-MM-AAAA
- Tel /Fax/L1/Lettres/ Autres	AN	30	
-Pannes signalées	AN	60	
-Reçu le	D	08	
-Par	A	30	JJ-MM-AAAA

FICHE D'ANALYSE DU DOCUMENT N° 2			
Identification			
<p><b>Désignation :</b> ordre de travaux  <b>Code :</b> /  <b>Rôle :</b> Sélection des techniciens adaptés aux besoins  <b>Nature :</b> externe  <b>Nombre d'exemplaire :</b> 01  <b>Emetteur :</b> cadre technique  <b>Récepteur :</b> équipe d'intervention</p>			
Désignation			
Rubrique	Type	Taille	Observation
- Numéro de l'ordre de travaux	N	11	JJ/MM/AAAA
-Code de la station	AN	05	
-Pannes signalées	AN	60	
-Date des travaux	D	08	
- Noms des agents	A	60	
-Véhicule	AN	30	

FICHE D'ANALYSE DU DOCUMENT № 3			
Identification			
<p><b>Désignation :</b> Fiche de travaux  <b>Code :</b> /  <b>Rôle :</b> Etablir un compte rendu de la mission  <b>Nature :</b> externe  <b>Nombre d'exemplaire :</b> 01  <b>Emetteur :</b> équipe d'intervention  <b>Récepteur :</b> cadre technique</p>			
Désignation			
Rubrique	Type	Taille	Observation
-Numéro de la fiche de travaux	N	11	JJ-MM-AAAA
-Code de la station	AN	30	
-Pannes signalées	AN	60	
-Noms des agents	A	60	
-Travaux effectués	AN	50	
-Désignation pièces	AN	50	
-Code pièces	AN	05	
-Durée des travaux	N	02	
-Date des travaux	D	08	
-Observation du client	AN	60	

FICHE D'ANALYSE DU DOCUMENT N° 4			
Identification			
<b>Désignation :</b> contrôle débit			
<b>Code :</b> /			
<b>Rôle :</b> contrôler l'état des équipements			
<b>Nature :</b> externe			
<b>Nombre d'exemplaire :</b> 01			
<b>Emetteur :</b> Equipe technique			
<b>Récepteur :</b> cadre technique			
Désignation			
Rubrique	Type	Taille	Observation
- Numéro control débit	N	11	
-Code station	AN	05	
-Numéro du VC	AN	06	JJ-MM-AAAA
-Date	D	08	
-SN VC duplex	N	05	OUI
-Contrôle plombage	A	03	NON
-Num VC. Coté A. Essai 1. Essence	N		
-Num VC. Coté A. Essai 1.Esuper	N		
- Num VC.Cotè A. Essai 1. ESP	N	05	
-Num VC. Coté A. Essai 1.Gasoil		05	Le VC contient 2 coté. Le contrôle s'effectue sur 2 essais.
-Observation			
-Observation de l'agent	AN		
-Commentaire technique	AN		
-Observation du client	AN	50	
-Nom client	AN	50	
Nom des agents techniques	AN	10	
-Remise en cuve .ESSENCE	N	50	
-Remise en cuve .E.SUPER	N	04	
-Remise en cuve .E.SP	N	04	
-Remise en cuve. GAZOIL	N	04	En litres
		04	En litres
			En litres
-VC DUPLEX. Essai 1.Gazoil 1		05	E n litres
-VC DUPLEX. Essai 1.Gazoil 2	N		

### II.3.4 Etude des procédures existantes

#### II.3.4.1 Introduction :

Une procédure c'est ensemble de traitement déclenché par un ou plusieurs événements existants au sein de notre champ d'étude à l'extérieure.

#### + Liste des procédures :

D'après toutes les informations recueillies dans ce service, nous avons recensé deux procédures qui sont les suivantes :

1. **Procédure d'intervention.**
2. **Procédure de maintenance préventive.**

#### II.3.4.2 Description des procédures existantes :

##### + Description de la procédure de maintenance curative :

Après la réception de la demande de travaux, le cadre technique établit un ordre de travaux qui sera transmis à l'équipe d'intervention, en fin de l'intervention cette dernière établira une fiche de travaux qui inclura les détails de l'intervention.

##### + Description de la procédure de maintenance préventive :

Chaque mois, le cadre technique transmet un contrôle débit avec des champs à remplir à l'équipe technique, après la vérification des équipements de la station du réseau, l'équipe technique retransmet le contrôle débit rempli sur l'état des équipements.

### II.3.5 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.

#### II.3.5.1 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. En distingue :

- ❖ les acteurs internes qui font partie du domaine mais qui ont des échanges avec

- ❖ les acteurs externes qui ne font pas parti 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 orienté de l'acteur émettant vers l'acteur recevant.

Le libellé du flux inscrit à coté de la flèche tracée.

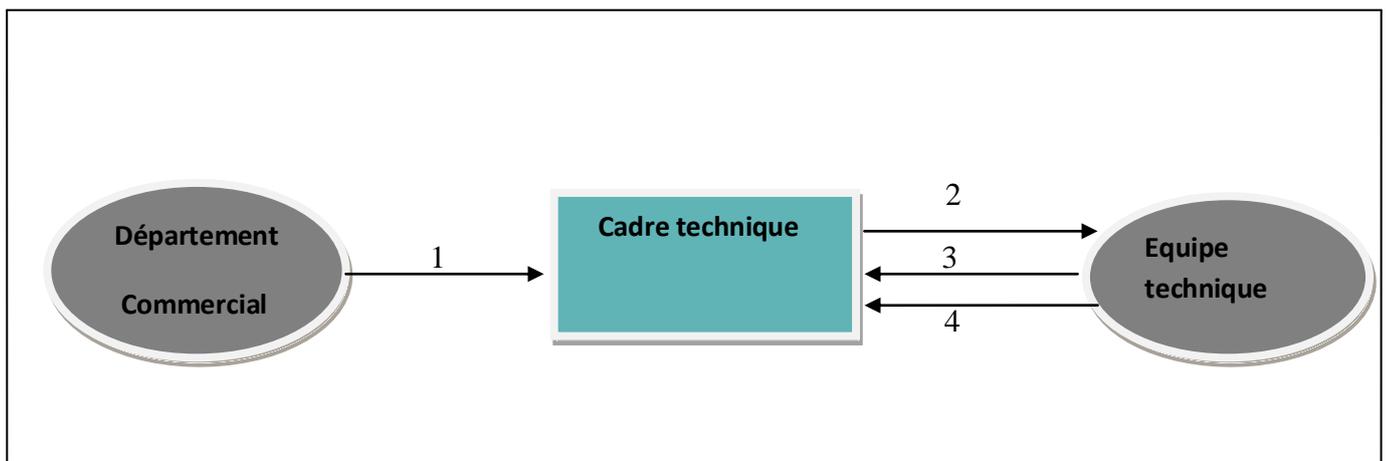
**Domaine connexe** : c'est un composant de notre organisme interagissant avec notre domaine d'étude. C'est un acteur interne à l'organisme, mais externe à notre domaine d'étude.

**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.



**II.3.5.2 Diagramme des flux d'information :**

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



**Figure II.3** : diagramme des flux

**II.3.5.3 Liste des flux d'information :**

N°	DESCIPTION
1	Demande de travaux
2	Ordre de travaux
3	Fiche de travaux
4	Control débit

**II.3.5 Problématique du champ d'études :**

En conclusion de ce chapitre il est utile de rappeler que l'un des principaux objectifs de l'étude de l'existant est de définir les différentes anomalies qui perturbent le bon fonctionnement du système et de déterminer les limites de l'application courante.

**Problématique :**

Le service Maintenance Réseaux est confronté à plusieurs problèmes dans la gestion de ses tâches, tels que :

- 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 dernière une fois les avoir reçu.
- Perte et entassement des piles de documents à cause de la non-informatisation de ceux-là.
- Difficulté de gestion des documents établit (demande de travaux, ordre de travaux, fiche de travaux et contrôle débit).
- Retard dans la réalisation des tâches telle que l'élaboration d'un ordre de travaux.

**Proposition de solution :**

Notre solution consiste à réaliser un application web connecté a une base de donnes sur les quelles seront enregistré toutes les informations nécessaire pour la gestion de la maintenance réseau de ce fait les employé n'aurons plus a remplir les informations manuellement.

**Les avantages de la solution :**

- Faciliter la recherche de l'information.
- Rapidité du traitement de l'information
- Edition des déférents documents.
- Informatisation des taches
- La rapidité dans la consultation des archives
- Faciliter le control des interventions.

**II.4 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 ces connaissances acquises afin de remédier aux problèmes évoqués.

# Chapitre III

*Analyse et conception.*

## IV.1 Introduction :

Après avoir présenté dans le chapitre précédent les différentes étapes d'analyse et de conception, nous allons présenter dans ce dernier chapitre, consacré à la réalisation et la mise en œuvre de notre application de gestion de maintenance, les outils de développement adoptés dans notre application ainsi que nous présenteront aussi certaines interfaces développées pour illustrer notre application.

## IV.2 Environnement de développement :

### IV.2.1 La plateforme J2EE :

L'élaboration de notre application s'appuie sur la J2EE qui est une plate-forme fortement orientée serveur pour le développement et l'exécution d'applications distribuées. Elle est composée de deux parties essentielles :

- un ensemble de spécifications pour une infrastructure dans laquelle s'exécutent les composants écrits en Java : un tel environnement se nomme serveur d'applications.
- un ensemble d'API qui peut être obtenues et utilisées séparément. Pour être utilisées, certaines nécessitent une implémentation de la part d'un fournisseur tiers.

Sun propose une implémentation minimale des spécifications de J2EE : le J2EE SDK. Cette implémentation permet de développer des applications respectant les spécifications mais n'est pas prévue pour être utilisée dans un environnement de production. Ces spécifications doivent être respectées par les outils développés par des éditeurs tiers.

J2EE permet une grande flexibilité dans le choix de l'architecture de l'application en combinant les différents composants. Ce choix dépend des besoins auxquels doit répondre l'application mais aussi des compétences dans les différentes API de J2EE. L'architecture d'une application se découpe idéalement en au moins trois tiers :

- la partie cliente : c'est la partie qui permet le dialogue avec l'utilisateur. Elle peut être composée d'une application standalone, d'une application web ou d'applets
- la partie métier : c'est la partie qui encapsule les traitements (dans des EJB ou des JavaBeans)
- la partie donnée : c'est la partie qui stocke les données.

Ce choix est justifié par plusieurs facteurs à savoir :

- ❖ La maturité et la richesse de cette technologie.
- ❖ La possibilité de la réutilisation des différents composants qui en font partie.
- ❖ La séparation forte qu'offre la plupart des Frameworks relevant de cette architecture.
- ❖ la possibilité de choisir les outils de développement et le ou les serveurs d'applications utilisés qu'ils soient commerciaux ou libres

- **Le Framework :** C'est un ensemble de composants qui servent à créer l'architecture et les grandes lignes d'une application. On peut le voir comme une boîte à outils géante, conçue par un ou plusieurs développeurs et mise à disposition d'autres, afin de faciliter leur travail. Il existe des Frameworks dans beaucoup de langages et plateformes. Ce n'est pas un concept propre à Java EE, nous pouvons par exemple citer JSF, Spring, Struts ou encore Hibernate. Toutes ces solutions sont des Frameworks que les développeurs sont libres d'utiliser ou non dans leurs projets.

#### IV.2.2 Le Framework Hibernate :

Hibernate est un Framework Java de persistance qui permet de faire correspondre des tables de base de données relationnelles avec des objets java simples (POJO ou «Plain Old Java Object»). Une fois la correspondance entre les deux mondes définie, le programme Java peut manipuler toutes les données en utilisant que des JavaBean, masquant alors totalement la base de données sous-jacente et ses spécificités. Le Framework assure le remplissage de ces objets et la mise à jour de la base en se basant sur leur contenu.

Hibernate 3 franchit un pas supplémentaire. De la même façon que les POJO permettent de manipuler des objets dont les données sont stockées dans une base de données, cette nouvelle fonctionnalité permet de s'affranchir des solutions XML propriétaires. En effet, chaque fournisseur, se devant de se préoccuper de l'importation/exportation des données XML, fournit une solution qui lui est propre à défaut de se baser sur un standard.

Le framework Hibernate permet d'obtenir une représentation XML du résultat d'une requête. Il permet aussi de rendre persistant, c'est-à-dire d'insérer ou de mettre à jour des données dans la base de données depuis des fragments de document XML de façon très simple et similaire à la manipulation de POJO.

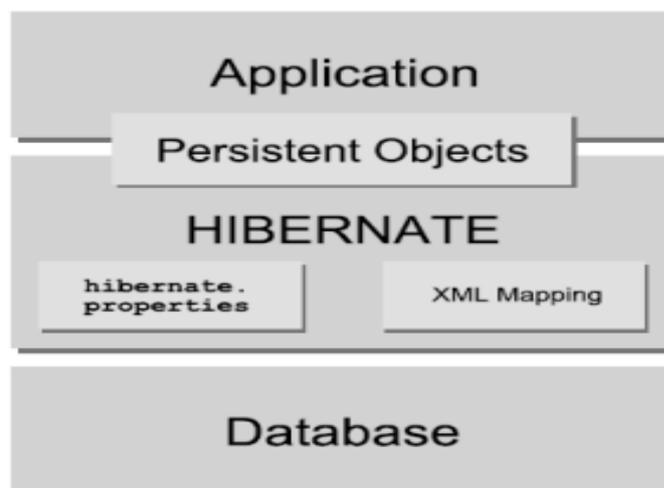
### IV.2.2 .1 Mise en œuvre du framework hibernate :

Hibernate a besoin de plusieurs éléments pour fonctionner :

- Création du projet Java supportant Hibernate sous Eclipse
- L'ajout du support de Hibernate revient à rajouter un ensemble de Jar du répertoire lib de Hibernate.
- Une classe de type JavaBean qui encapsule les données d'une table donnée nommée « classe de persistance ».
- Un fichier de correspondance qui configure la correspondance entre la classe et la table.
- Des propriétés de configuration, notamment des informations concernant la connexion à la base de données.

Une fois ces éléments correctement définis, il est possible d'utiliser Hibernate dans le code des traitements à réaliser.

L'architecture d'Hibernate est donc la suivante :



**Figure IV.1** Architecture Hibernate

## A. Création du fichier de configuration : hibernate.cfg.xml

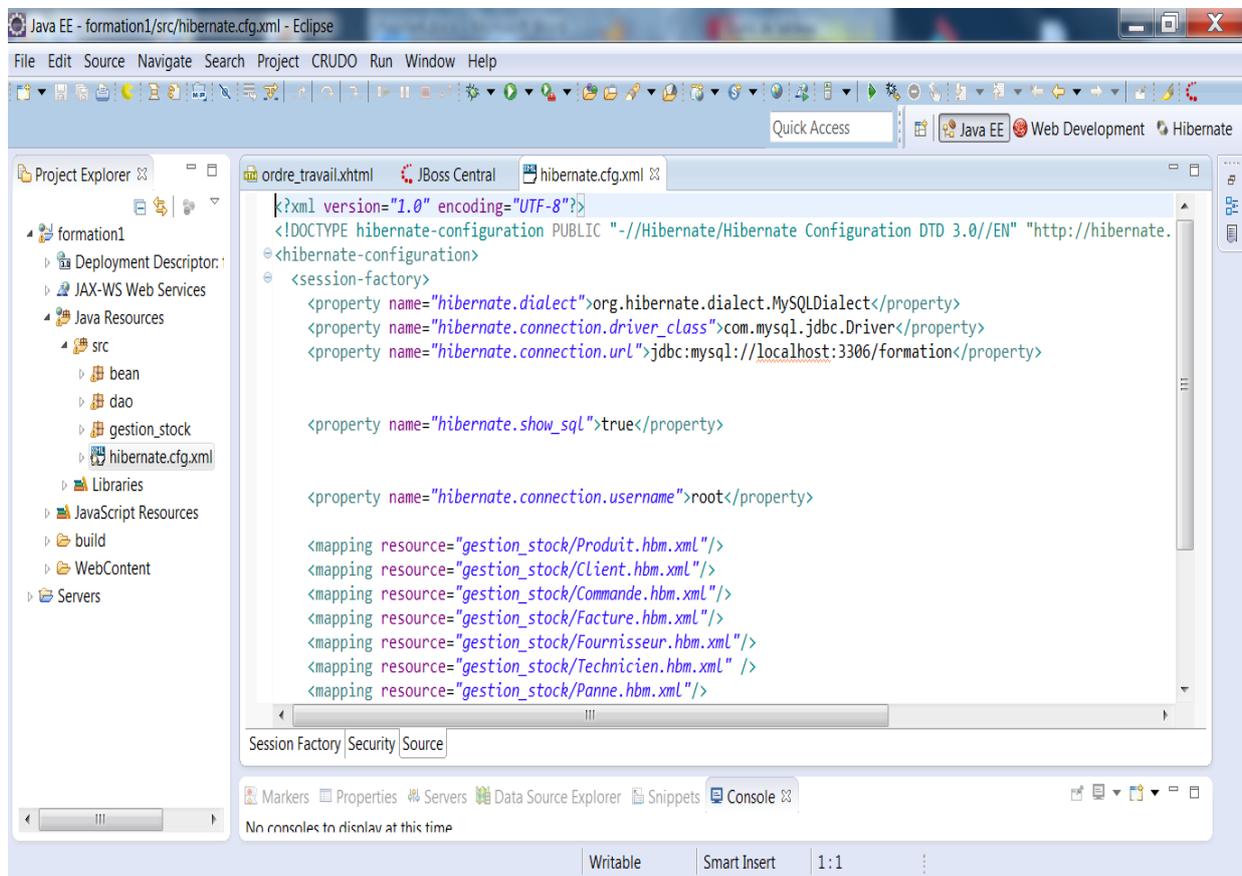
Hibernate propose des classes qui héritent de la classe Dialect pour chaque base de données supportée. C'est le nom de la classe correspondant à la base de données utilisée qui doit être obligatoirement fourni à la propriété hibernate.dialect.

Les propriétés sont alors définies par un tag <property>. Le nom de la propriété est fourni grâce à l'attribut

« name » et sa valeur est fournie dans le corps du tag.

Il est possible de fournir les propriétés de configuration « Hibernate » dans un fichier hibernate.properties.

Le fichier **hibernate.cfg.xml** se présente comme suit :



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN" "http://hibernate.
<hibernate-configuration>
  <session-factory>
    <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
    <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
    <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/formation</property>

    <property name="hibernate.show_sql">>true</property>

    <property name="hibernate.connection.username">root</property>

    <mapping resource="gestion_stock/Produit.hbm.xml"/>
    <mapping resource="gestion_stock/Client.hbm.xml"/>
    <mapping resource="gestion_stock/Commande.hbm.xml"/>
    <mapping resource="gestion_stock/Facture.hbm.xml"/>
    <mapping resource="gestion_stock/Fournisseur.hbm.xml"/>
    <mapping resource="gestion_stock/Technicien.hbm.xml"/>
    <mapping resource="gestion_stock/Panne.hbm.xml"/>
```

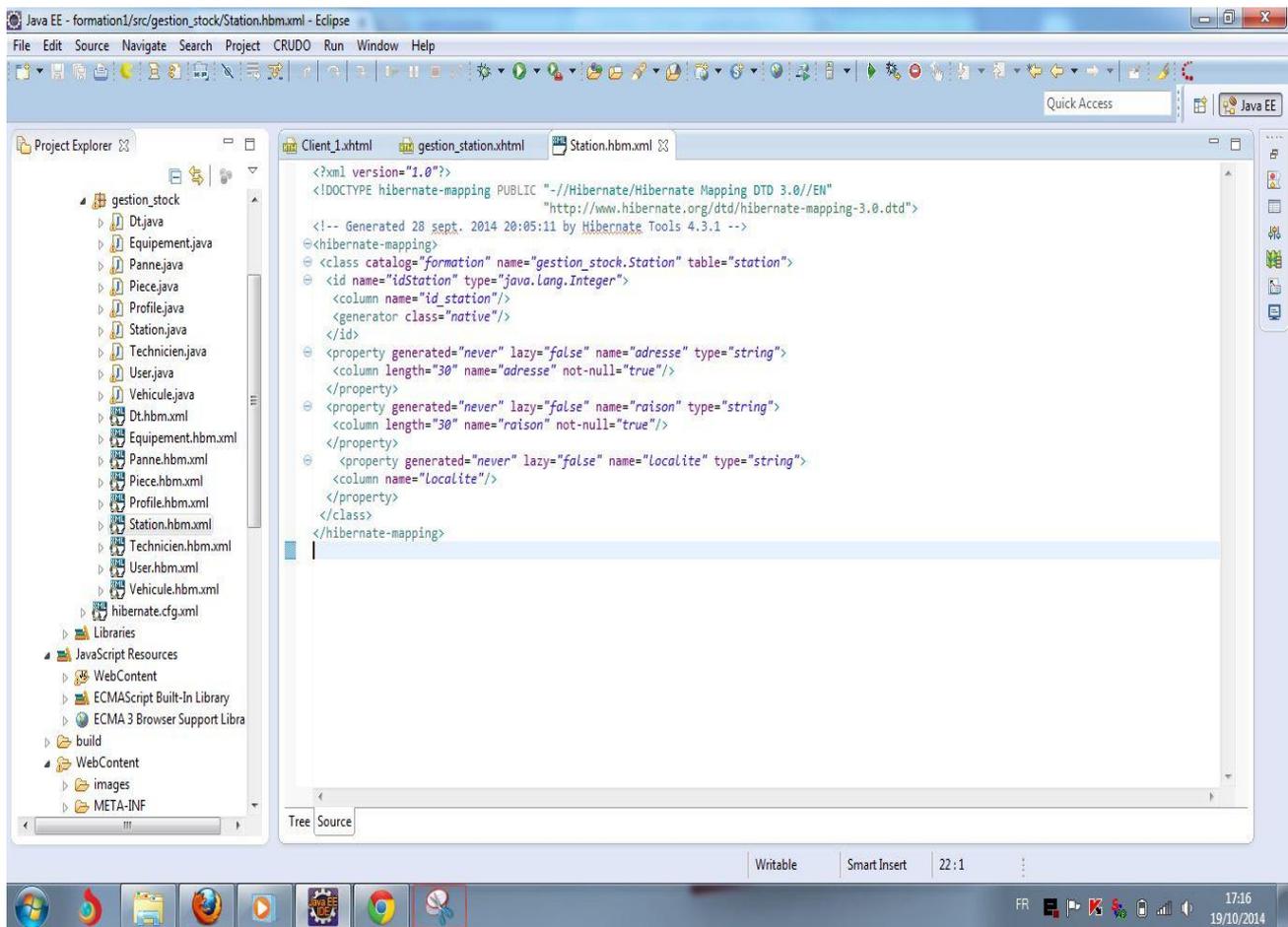
**Figure IV.2** Le fichier hibernate.cfg.xml

**B. Création des fichiers xml de mapping :**

Ces fichiers sont des éléments majeurs puisqu'ils vont permettre à Hibernate de faire le pont entre les classes de persistance et les sources de données.

Dans cette partie, on va présenter le fichier de mapping relatif à la table «Station » nommée Station.hbm.xml

Les fichiers **xml de mapping** se présentent comme suit :



**Figure IV.3** Les fichiers xml de mapping

Il est absolument indispensable d’ajouter la référence du fichier « Station.hbm.xml » dans le fichier de configuration « hibernate.cfg.xml » au niveau de la balise <mapping /> comme suit:

```
<mapping resource="nom BDD /Station.hbm.xml" />
```

### C. Création des Classes de données :

Une classe de données est un Javabean qui va encapsuler les propriétés de la table dans ses champs **private** avec des getters et setters et qui a un constructeur par défaut

Les Classes de données se présentent comme suit :

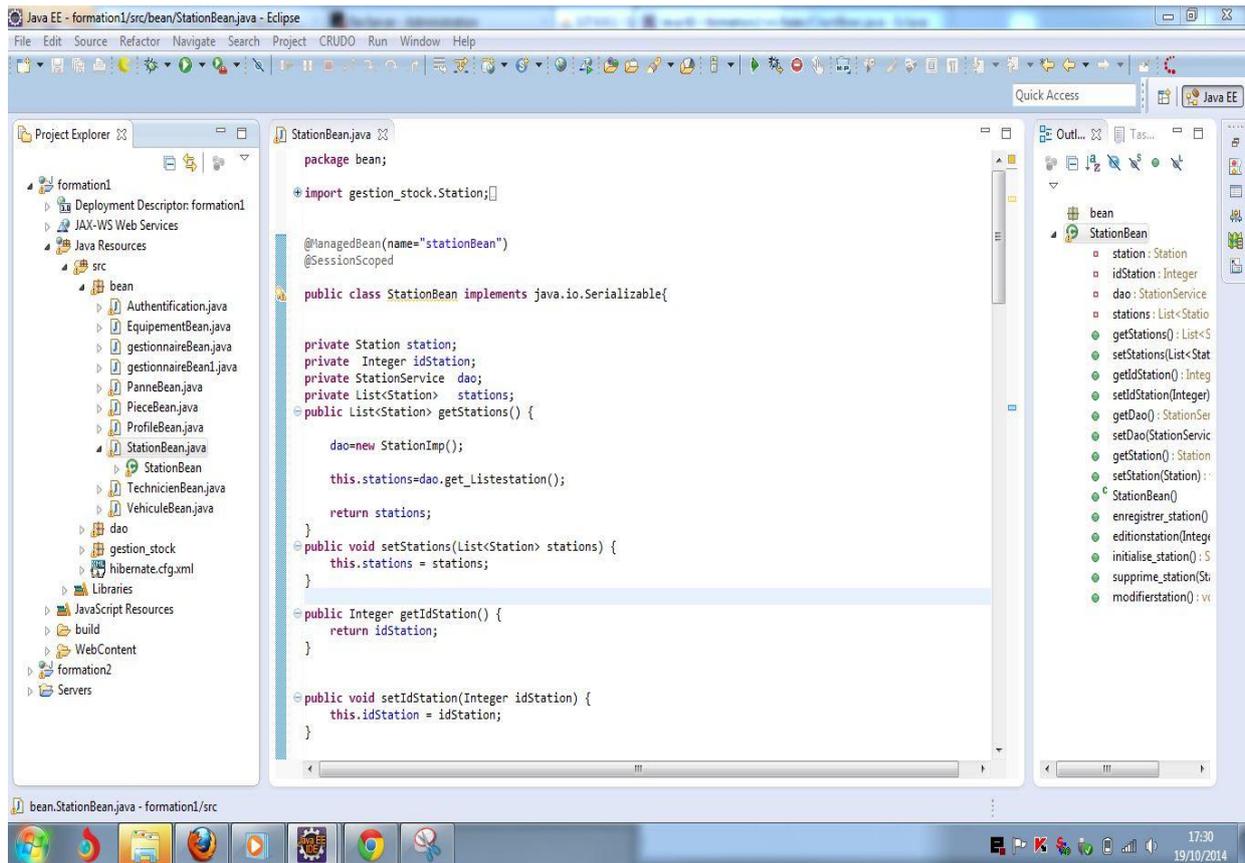


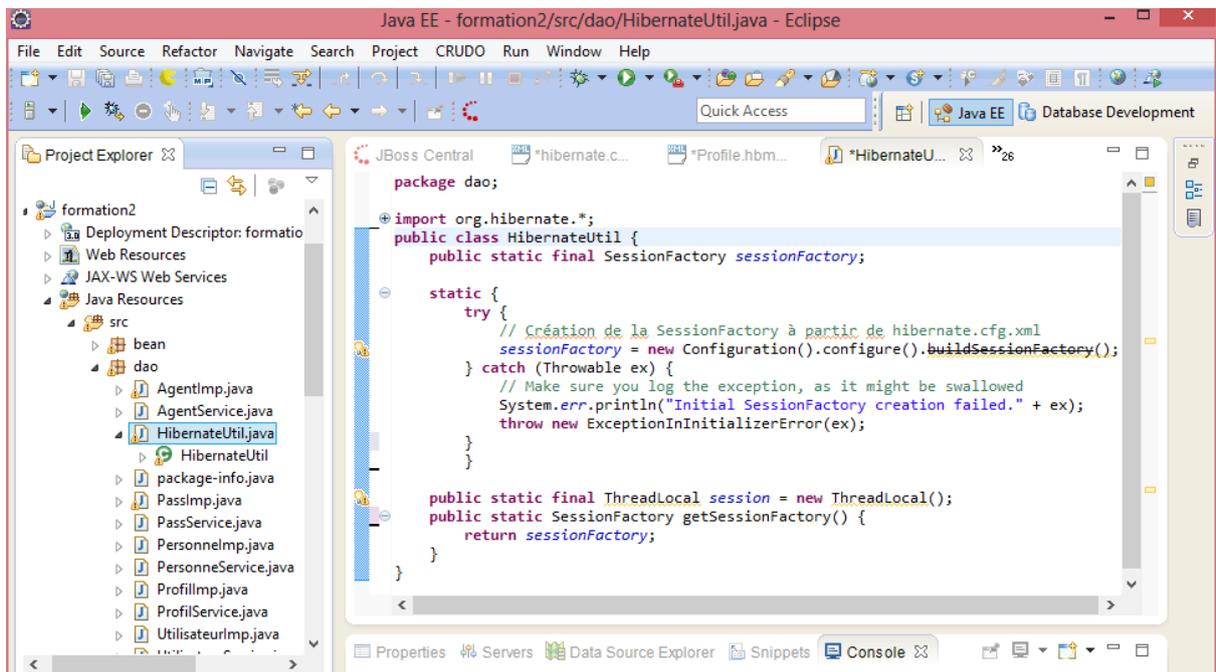
Figure IV.4 Javabean

**D. La classe HibernateUtil :**

La classe Hibernate nommée SessionFactory permet d'établir la connexion avec la source de données à partir du fichier de configuration « hibernate.cfg.xml ». On remarque que la classe

SessionFactory serait instanciée autant de fois qu'il y a de threads, il est donc plus adapté de rendre une même instance de SessionFactory accessible par les threads. Cette classe possède une méthode appelée currentSession() qui retourne la session hibernate en cours si elle existe sinon elle se charge d'ouvrir une nouvelle session.

La classe HibernateUtil se présente comme suit :



**Figure IV.5** La classe HibernateUtil

### IV.2.3 Le Framework JSF/prime faces :

Java server faces est un Framework de développement d'applications web en java permettant de respecter le model d'architecture MVC et basé sur des composantes coté présentation.

- Architecture MVC pour séparer l'interface utilisateur, la couche de persistance et les processus métiers, utilisant la notion d'événement, conversion des données, validation des données (par exemple des champs de formulaires requis).
- Automatisation de l'affichage des messages d'erreur en cas de problèmes de conversion ou de validation.

Le premier objectif de JSF, est de procurer un environnement de développement permettant de construire une interface de type web, sans devoir toucher au code HTML et JavaScript. Ceci est réalisé par la mise en place d'un mapping entre l'HTML et les objets concernés. JSF est donc basé sur la notion de composants, comparable à celle de Swing, ou l'état de ces composants est sauvegardé puis restauré au retour de la requête.

L'implémentation de Jsf nécessite l'implémentation des deux fichiers suivants :

➤ **faces-config.xml :**

Le faces-config.xml, est le fichier de configuration, qui permet de gérer les beans, et les règles de navigation entre les pages. Vous pouvez apercevoir, ci-dessous un bref, exemple de fichier de configuration.

➤ **web.xml : descripteur de déploiement**

Le web.xml qui permet de définir l'emplacement des pages JSP, ainsi que de possible contraintes de sécurité concernant l'accès à des pages. Il établit également l'endroit où se situe le Faces Servlet. Faces Servlet, est la classe, qui est le moteur de chaque application JSF. Chaque application JSF à sa propre Faces Servlet, qui gère toutes les informations relatives à la requête courante.

➤ **Prime faces :**

Prime faces c'est l'un des composants de JSF c'est une bibliothèque open source de composantes JSF il est basé coté serveur sur l'API standard de ce dernier. Elle permet de mettre à disposition des interfaces clients riches dans les applications web JEE.

Primefaces est un jar facile à ajouter dans votre projet jsf pour obtenir un jeu de composants additionnel beaucoup avancé qui prend en compte les nouveautés des technologies du web surtout Ajax. Il existe des alternatives à Primefaces dont les plus connues et les plus anciennes sont **RichFaces** de la communauté JBoss et **IceFaces** qui s'est suicidé à partir de sa version 3 en copiant en totalité le code de primefaces.

#### IV.2.4 Architecture de notre application selon le modèle MVC :

##### ❖ Le modèle MVC :

Un modèle de conception, en anglais design pattern (ou encore patron de conception) est une simple bonne pratique, qui répond à un problème de conception. C'est en quelque sorte une ligne de conduite qui permet de décrire les grandes lignes d'une solution. De tels modèles sont issus de l'expérience des concepteurs et développeurs d'applications : c'est en effet uniquement après une certaine période d'utilisation que peuvent être mises en évidence des pratiques plus efficaces que d'autres, qui sont alors structurées en modèle et considérées comme standard.

Java EE permet plus ou moins de coder son application comme on le souhaite, en d'autres termes, on peut coder n'importe comment, et dans Java EE, il y'a « Entreprise », et le développement en entreprise implique entre autre :

- ❖ Que l'on puisse être amené à travailler à plusieurs contributeurs sur un même projet ou une même application (travail en équipe).
- ❖ Que l'on puisse être amené à faire évoluer une application que l'on n'a pas créée soi-même.

Pour toutes ces raisons, il est nécessaire d'adopter une architecture plus ou moins standard, que tout développeur peut reconnaître, c'est-à-dire dans lesquels tous développeurs sait se repérer.

Il a été très vite remarqué qu'un modèle permettait de à ces besoins, et qu'il s'appliquait particulièrement bien à la conception d'application Java EE : le modèle MVC (Modèle - Vue - Contrôleur). Il découpe littéralement l'application en couches distinctes, et de ce fait impacte très fortement l'organisation du code. Voici ce qu'impose MVC :

- ✓ Tout ce qui concerne le traitement, le stockage et la mise à jour des données de l'application doit être contenu dans la couche nommée « Modèle » (le M de MVC).
- ✓ Tout ce qui concerne l'interaction avec l'utilisateur et la présentation des données (mise en forme, affichage) doit être contenu dans la couche nommée « Vue » (le V de MVC).
- ✓ Tout ce qui concerne le contrôle des actions de l'utilisateur et des données doit être contenu dans couche nommée « Contrôle » (le C de MVC).

La figure suivante montre l'architecture de notre application :



### IV.3 Outils de développement :

#### IV.3.1 Matériels

- PC portable (OS Windows, Intel core I3, 4GO RAM ).

#### IV.3.1 Logiciels :

##### IV.3.1.1 Environnement de développement intégré : Eclipse :

Eclipse est un environnement de développement intégré libre extensible, universel et polyvalent, permettant 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 est 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 plugin (en conformité avec la norme OSGI) : toutes les fonctionnalités de cet atelier logiciel sont extensions.

Eclipse massivement utilisé en entreprise, est un outil puissant, libre et multiplateforme. Les avantages d'un IDE dans le développement d'applications web Java EE sont multiples, et en voici quelque uns :

- Intégration des outils nécessaires au développement et au déploiement d'une application.
- Paramétrage aisé et centralisé des composantes d'une application.
- Multiples moyens de visualisation de l'architecture d'une application.
- Génération automatique de portion de code.
- Assistance à la volée lors de l'écriture du code.
- Outils de débogage...

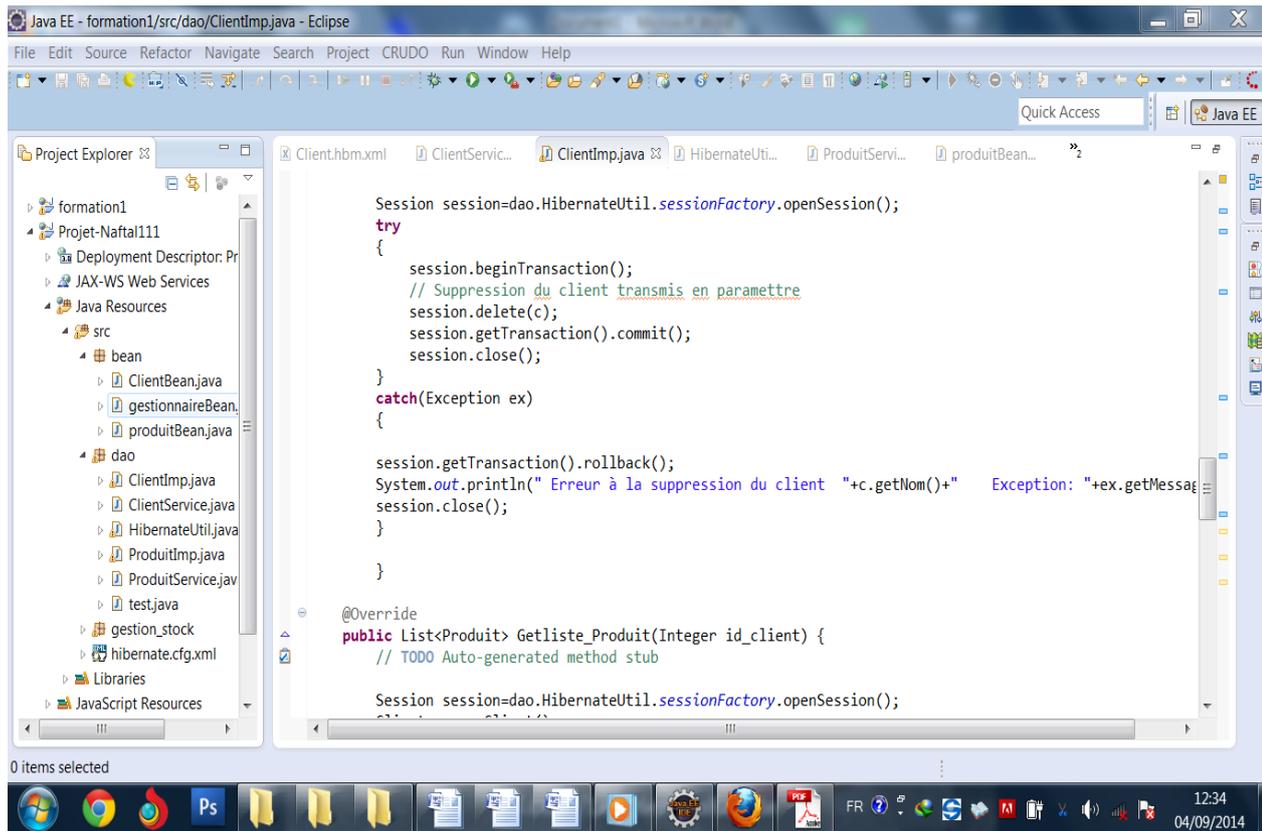


Figure IV.7 Interface de l’IDE Eclipse.

### IV.3.1.2 Le serveur d'application :

#### ❖ Apache Tomcat

Pour faire fonctionner une application web Java EE, nous avons besoin de mettre en place un serveur d'application. Il en existe beaucoup sur le marché, on a choisi d'utiliser Tomcat, car c'est un serveur léger, gratuit, libre, multiplateforme et assez complet pour ce que nous allons développer. On le rencontre d'ailleurs très souvent dans des projets en entreprise, en phase de développement comme en production.

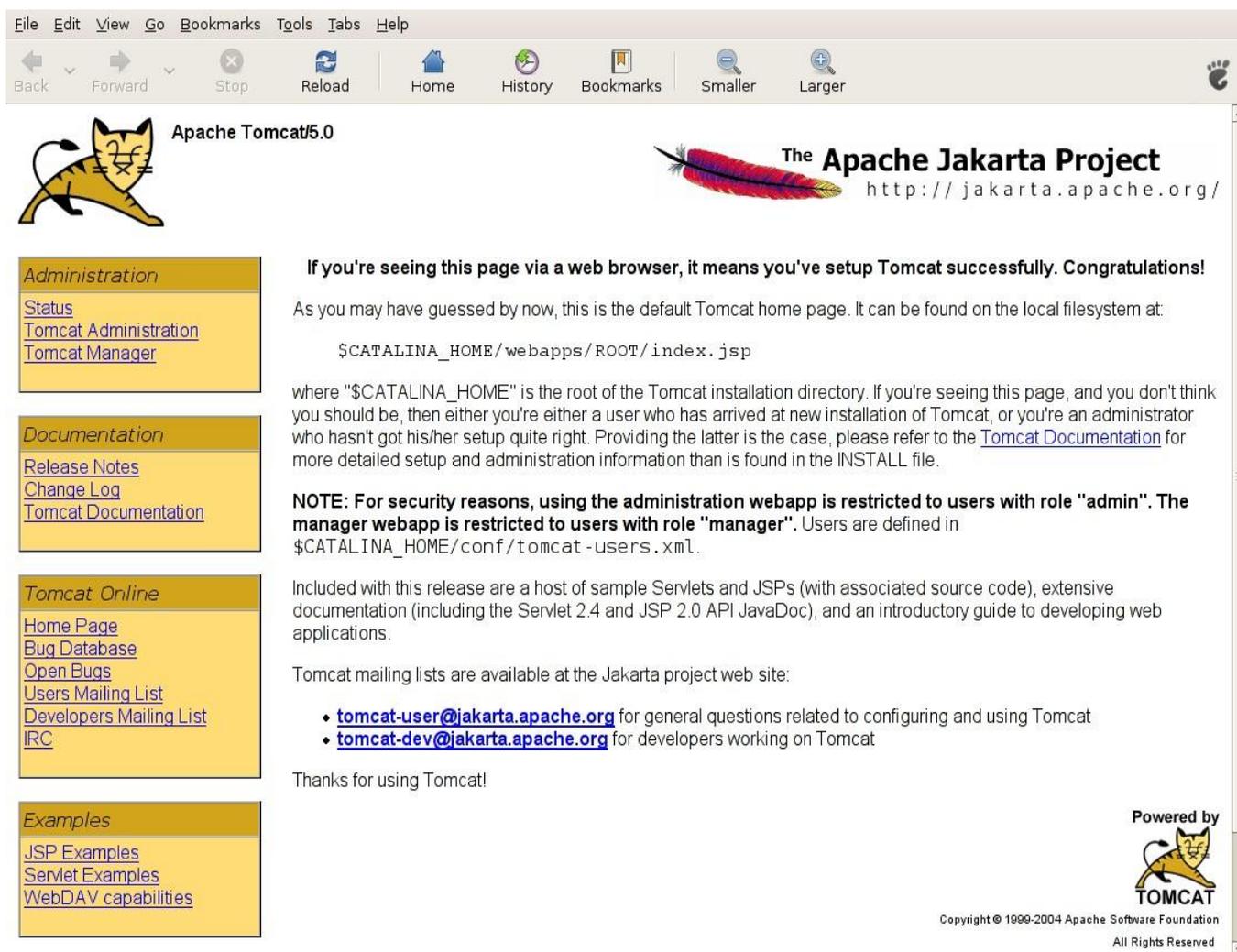


Figure IV.8 Apache Tomcat.

**IV.3.1.3 Le serveur de données :**

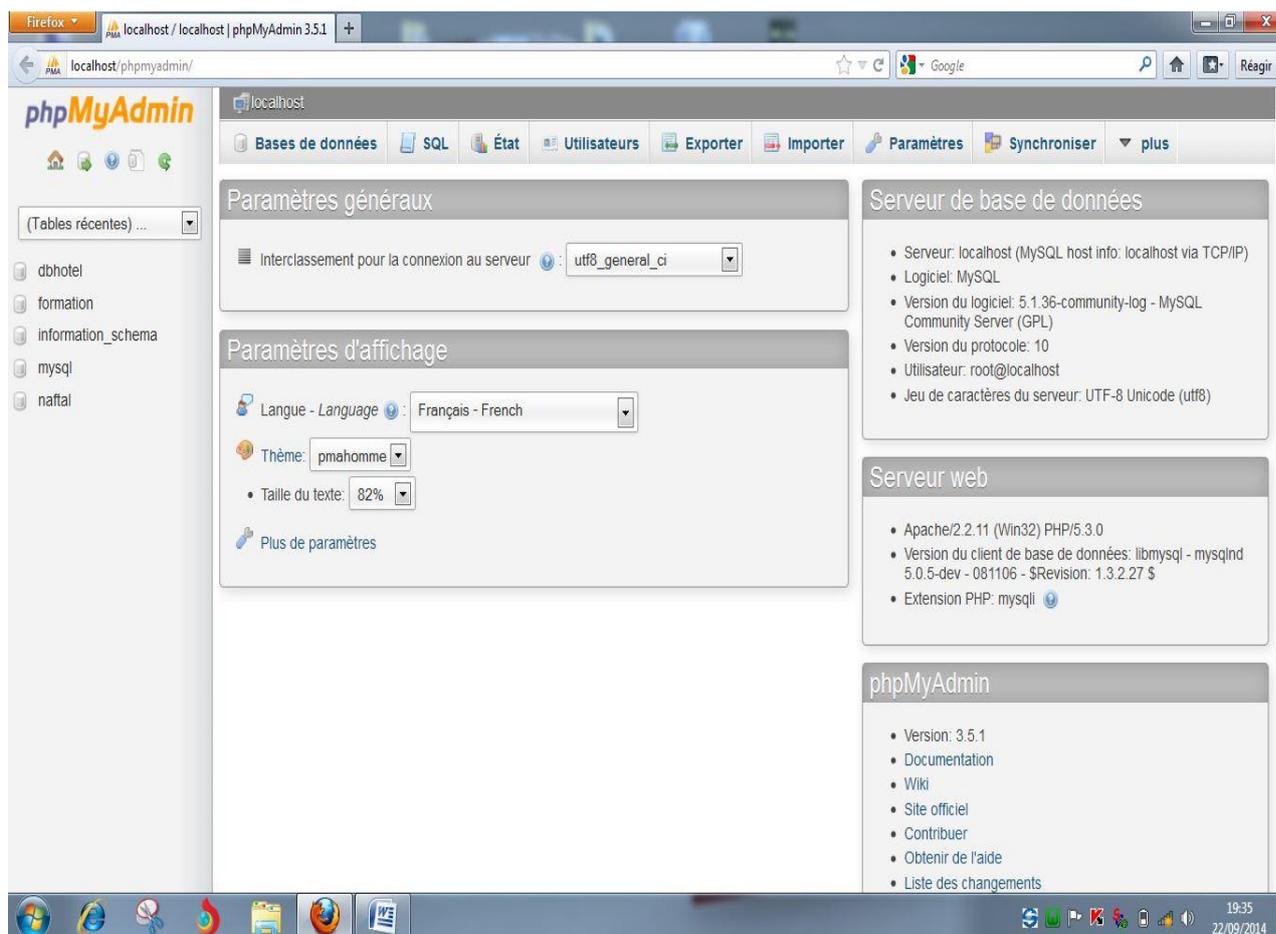
❖ **Wampserver :**

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 a 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 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. Sue 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.9** L'interface PHPmyadmin

❖ **Construction de la base de données selon les itérations :**

- **Itération 1 :** voilà son modèle relationnel :

✚ **User** (Id\_user, login,mdp , profile )

Voilà les table : de l'**itération 1** :

- **Table Utilisateur**

Nom du champ	Type de données	Taille	Description
Id_user	int	11	identificateur
login	Varchar	30	Nom de l'utilisateur
mdp	Varchar	30	Mot de passe de l'utilisateur
profile	varchar	30	Type d'accès

- **Itération 2 :**

voilà son modèle relationnel :

✚ **Station** (id\_station, adresse, raison )

✚ **Technicien** ( id\_tec, id\_OT\*, nom\_tec, prenom\_tec, adresse, tel\_tec)

✚ **Véhicule** ( id\_OT, id\_Technicien, marque , modele, immatriculation)

✚ **Equipement** ( id\_equipement, id\_station\*, designation, année\_fabrication)

✚ **DT** (id\_DT, id\_station\*, panne-signalé, priorité,date, commentaire)

✚ **Piece** ( id\_piece, id\_FT\*,emb, designation\_piece)

Voilà les tables de l'itération 2 :

- **Table Station**

Nom du champ	Type de données	Taille	Description
id_station	int	11	Identificateur station
adresse	varchar	50	Adresse de la station
raison	vararch	30	raison

- **Table Technicien**

Nom du champ	Type de données	Taille	Description
id_tec	int	11	Identificateur Technicien
id_OT	int	11	Identificateur OT
nom_tec	varchar	30	Nom du technicien
prenom_tec	varchar	30	Prénom du technicien
adresse	varchar	50	Adresse du technicien
tel_tec	varchar	15	Téléphone du technicien

- **Table Véhicule**

Nom du champ	Type de données	Taille	Description
id_OT	int	11	Identificateur de OT
Id_Technicien	int	11	Identificateur du technicien
marque	varchar	30	Marque du véhicule
modele	varchar	30	Modèle du véhicule
immatriculation	varchar	15	Immatriculation du véhicule

• **Piece** ( id\_piece, id\_FT\*,emb, designation\_piece)

Nom du champ	Type de données	Taille	Description
id_piece	int	11	Identificateur piece
id_FT	int	11	Identificateur FT
qte	int	10	Quantité en stock
designation_piece	vchar	30	Designation de piece
cds	vchar	30	Code referential au magasin

• **Equipement**

Nom du champ	Type de données	Taille	Description
id_equipement	int	11	Identificateur équipement
id_station	int	11	Identificateur station
designation	vchar	30	Désignation station
année_fabrication	int	4	Année fabrication équipement

• **DT**

Nom du champ	Type de données	Taille	Description
id_DT	int	11	Identificateur Demande de Travail
id_station	int	11	Identificateur station
panne-signalé	vchar	35	Panne signalé par la station
priorité	vchar	25	Priorité de la demande selon l'urgence
date	date		Date de la demande de travail
commentaire	vchar	80	Commentaire du client

- **Itération 3 :**

voilà son modèle relationnel :

✚ **Panne** ( id\_panne, id\_equipement\*, designation\_panne, remède)

✚ **OT** ( id\_OT, id\_DT\*, date, agent, vihecule)

✚ **FT**(id\_FT, id\_OT\* , travaux\_effec, piece, durée\_travaux, date\_travaux, -  
agents, observation\_agent )

✚ **CD** (id\_CD, id\_equipement\*, date\_control, observation, agents)

Voilà les tables de l'itération 3 :

- **Panne**

Nom du champ	Type de données	Taille	Description
<u>id_panne</u>	int	11	Identificateur panne
id_equipement	int	11	Identificateur equipement
nom_panne	varchar	30	Nom de la panne
remède	varchar	30	Remède de la panne

- **OT**

Nom du champ	Type de données	Taille	Description
<u>id_OT</u>	int	11	Identificateur Ordre de de Travail
id_DT	int	11	Identificateur Demande de Travail
date	date		Date de OT
agent			Technicien chargé de la mission
vihecule			

- **FT**

Nom du champ	Type de données	Taille	Description
id_FT	int	11	Identificateur fiche d'interven
id_OT	int	11	Identificateur Ordre de Travail
travaux_effec	varchar	60	Travaux effectués
id-piece	int	11	Pièces changés
durée_travaux	varchar	30	Durée de l'intervention
date_travaux	date		Date de l'intervention
agents	varchar	30	Agents chargés de la mission
observation_agent	varchar	60	Observation de l'agent

- **CD**

Nom du champ	Type de données	Taille	Description
id_CD	int	11	Identification control debit
<u>id_equipement</u>	int	11	Identification equipment
date_control	date		Date du controle
observation	varchar	60	observation
agents	varchar	30	Agent chargé du controle

#### IV.4 Langages de programmation :

Pour satisfaire les besoins de l'application à réaliser, nous avons utilisé les langages suivants :

##### 1-Le langage coté serveur :

- ❖ Le langage orienté objet **Java** pour la création des différentes classes Java.
- ❖ **HQL** (Hibernate Query language) : Hibernate Query Langage est un langage d'interrogation des classes persistantes avec une syntaxe proche du SQL. Cependant les objets manipulés sont les classes et membres du mapping contrairement au SQL qui réalise les requêtes directement sur la base de données. Aussi, le HQL contient des fonctionnalités spécifiques au modèle objet.
- ❖ Le langage de requêtes MySQL pour la création de la base de données.

##### 2- Le langage côté client

- ❖ Le langage XHTML (Extensible Hypertext Markup Language) pour la création des pages web qui constitueront les interfaces de l'application.  
C'est une évolution de l'HTML, rendant sa syntaxe plus stricte en appliquant les règles de l'XML. En effet, l'HTML est un langage très peu strict (ce qui signifie que les navigateurs Internet arriveront souvent à interpréter l'HTML même s'il n'est pas correctement écrit). Dans des utilisations sophistiquées, ce manque de rigueur peut être problématique. L'XHTML essaye donc de palier à ce problème en apportant beaucoup plus de rigueur à l'HTML.

- ❖ **XML** pour la création de fichiers mapping relationnel-objet et pour la déclaration.

XML (eXtensible Markup Language), est le standard soutenu par le W3C pour le balisage de documents. Il définit une syntaxe générique utilisée pour formater des données avec des balises simples et compréhensibles par l'homme. Ce format est suffisamment souple pour être adapté à des contextes aussi variés que les sites WEB, l'échange de données électroniques, les dessins vectoriels, les menus déroulants de logiciels, les descripteurs de déploiement, les procédures d'appel à distance (XML-RPC), etc

- ❖ **Le Java Script : [15]**

Le JavaScript est un langage de script incorporé dans un document **HTML**. Historiquement il s'agit du premier langage de script pour le web, mis au point par Netscape en **1995**. Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage **HTML** en permettant d'exécuter des commandes du côté client, c'est-à-dire au niveau du navigateur et non du serveur web.

Ainsi, ce langage est fortement dépendant du navigateur appelant la page web dans laquelle le script est incorporé, mais en contrepartie il ne nécessite pas de compilateur, contrairement au langage *Java*, avec lequel il a longtemps été confondu

#### IV.5 Présentation des interfaces de l'application :

L'objectif de cette partie du chapitre est de donner un aperçu de l'interface homme/machine de notre application :

##### ❖ L'interface d'authentification :

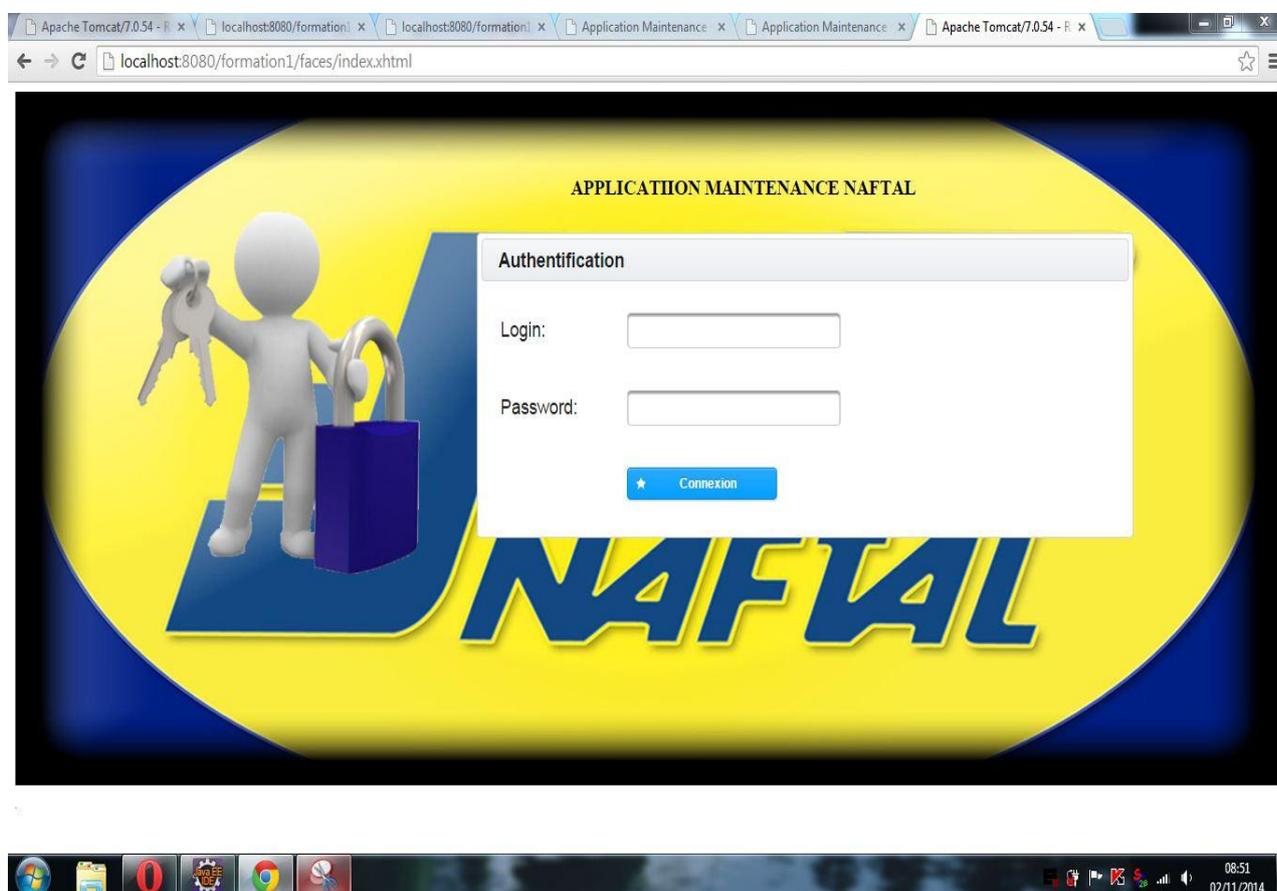


Figure IV.10 L'interface d'authentification

❖ L'interface d'accueil pour l'administrateur :



Figure IV.11 L'interface d'accueil pour l'administrateur

❖ L'interface d'accueil pour le cadre technique :



Figure IV.12 L'interface d'accueil pour le cadre technique

❖ L'interface d'accueil pour le chef d'équipe d'intervention :



**Figure IV.13** L'interface d'accueil pour le chef d'équipe d'intervention

❖ L'interface A-propos :

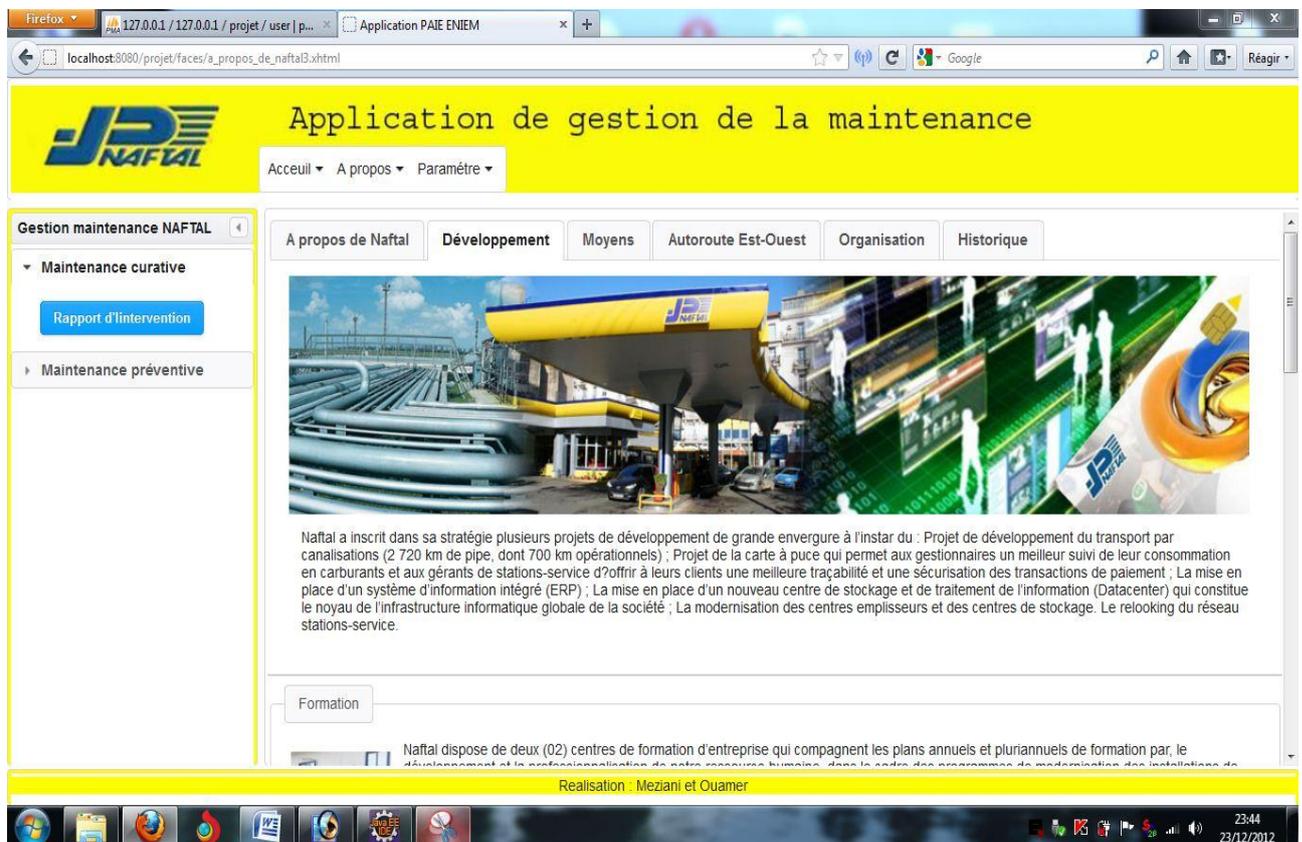


Figure IV.14 L'interface A-propos

❖ L'interface de création d'un nouvel utilisateur :

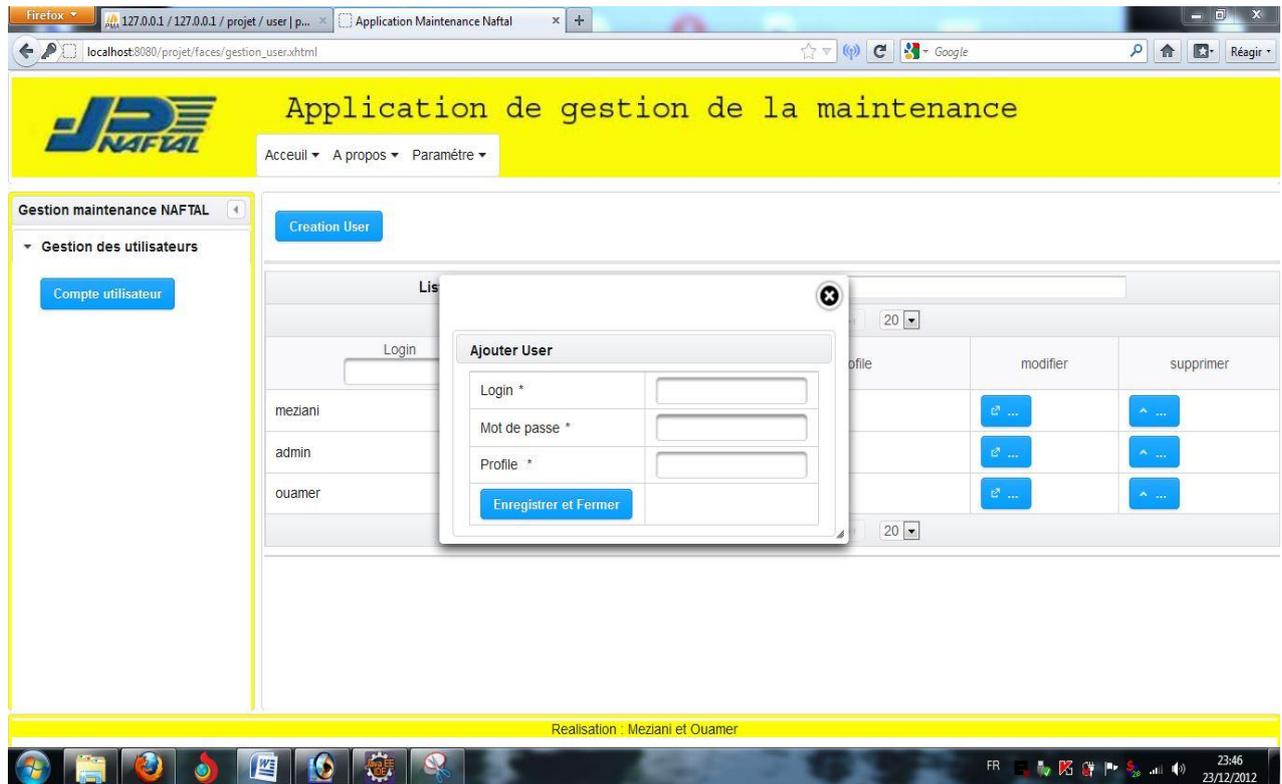


Figure IV.15 L'interface de création d'un nouvel utilisateur

❖ L'interface de gestion des techniciens:

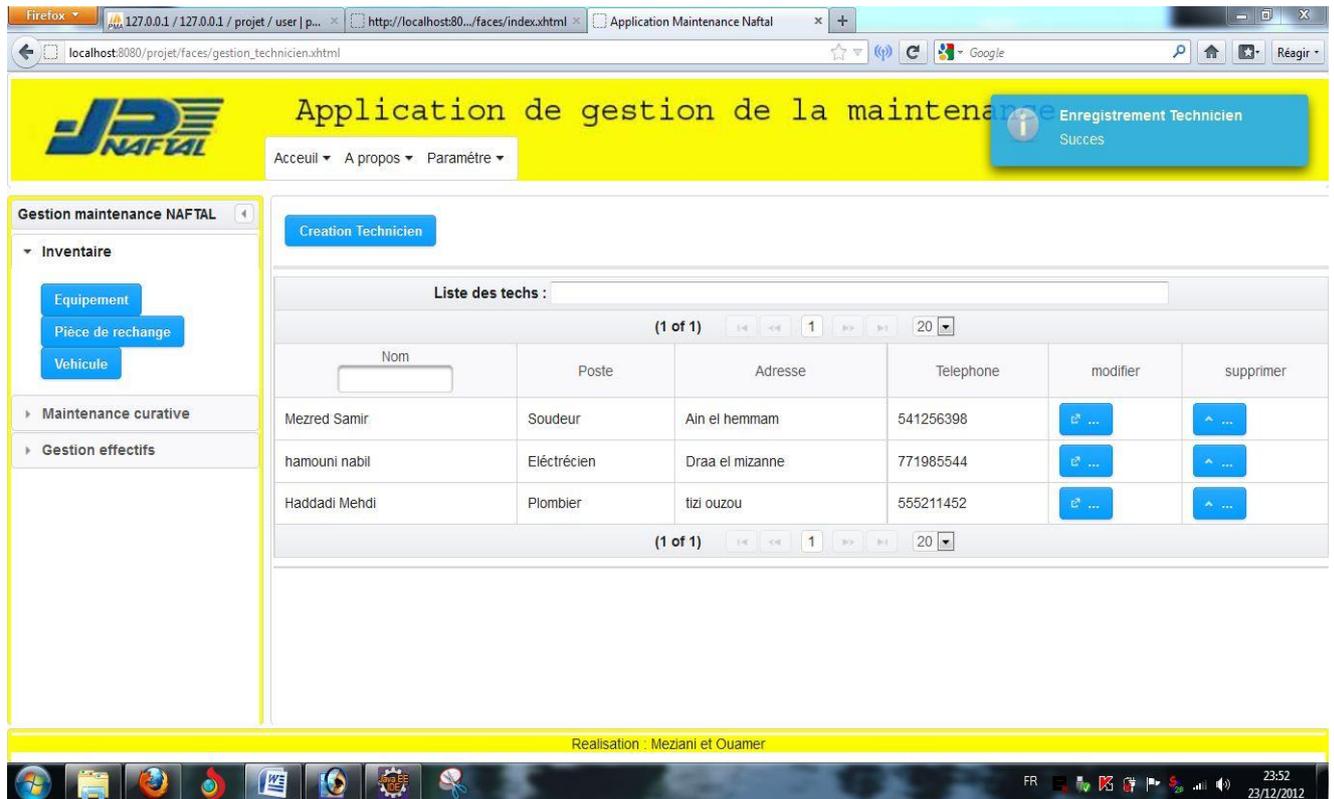


Figure IV.16 L'interface d'accueil pour le chef d'équipe d'intervention

❖ L'interface de modification d'un équipement :

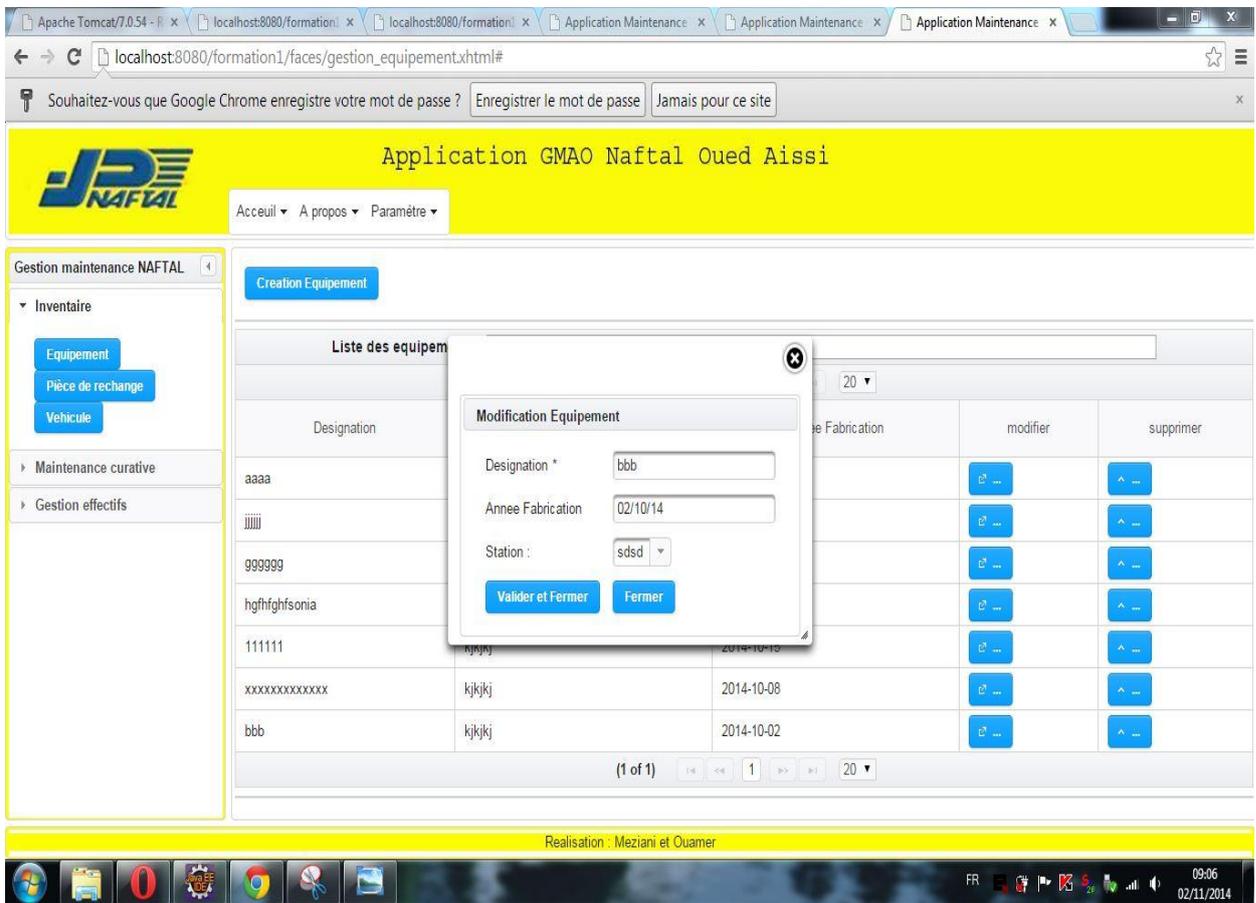


Figure IV.17 L'interface de modification d'un équipement

IV.6 Conclusion :

Ce dernier chapitre à été consacré à la présentation de l'étape réalisation de notre application ainsi, nous avons pu voir les exigences technique du système avant de présenté l'environnement et les outils choisis. Par la suite, nous avons donné un aperçu de l'application réalisée.

# Chapitre IV

*Réalisation et mise en œuvre.*

# Conclusion Générale

# Conclusion Générale

# Conclusion Générale

---

L'objectif de notre application était de concevoir une application web pour la gestion de la maintenance, offrant un ensemble de services en conformité avec l'exigence de la section maintenance réseau l'entreprise NAFTAL Tizi Ouzou particulièrement le service Maintenance.

Sur le plan technique nous avons découvert le monde du développement web et nous avons aussi appris les bases de la plateforme J2EE, notamment l'architecture MVC.

La réalisation de ce travail nous a donné l'occasion d'acquérir de nouvelles connaissances et d'en approfondir d'autre sur le développement web tel que le JSF, je citerai le langage orienté objet JAVA précisément sous la plateforme J2EE et avec le langage de modélisation UML et de maîtriser la démarche de développement RUP.

Ça nous a permis aussi de se familiariser avec un certain nombre d'outils informatiques de développement, à titre d'exemple Eclipse, Tomcat et MySQL.

Hibernate quant à lui est beaucoup plus complexe à maîtriser et nécessiterait un projet plus complet pour en maîtriser toutes les facettes. Malgré cela il nous a apparue comme étant un outil exécrément puissant. Son efficacité est d'ailleurs reconnue à tel point que le Framework à été apporté en C# pour être utilisé dans l'environnement .NET c'est l'un des Framework incontournables dans le monde JEE.

Après le passage par les différentes étapes de développement, l'application a abouti à un logiciel fonctionnel qui répond globalement aux critères imposé dans ce domaine, cependant d'éventuelles améliorations et compléments peuvent être apportés.

# **Annexes**

## Introduction

Vu la complexité croissante des systèmes d'information, de différentes méthodes de modalisations ont vu le jour et adaptées à ces derniers.

Dans l'objectif de combler les lacunes constatées, plusieurs méthodes sont apparues. Le groupe OMG (Object Management Group) a développé une notation standard pour le développement informatique basé sur l'objet, ainsi l'apparition de l'UML.

UML, est une fusion des principales méthodes objet : BOOCH, OMT (Object Modeling Technique) et OOSE (Object Oriented Software Engineering), puis normalisé par l'OMG en 1997, est désormais devenu une référence en termes de modélisation.

### I. Présentation de l'UML :

UML est un langage qui permet de représenter des modèles. Il a été conçu pour servir de support à une analyse basée sur les concepts objets. UML propose plusieurs modèles qui sont des descriptions du système étudié, et parmi ces modèles :

- Le modèle de classe
- Le modèle des cas d'utilisation
- Le modèle d'interaction
- Le modèle des états
- Le modèle de réalisation
- Le modèle de déploiement

Ces modèles sont représentés au moyen de diagrammes qui spécifient un aspect précis du système.

On peut organiser une présentation d'UML autour d'un découpage en vues, ou bien en différents diagrammes, Nous adopterons plutôt dans la suite un découpage en diagrammes, mais nous commençons par présenter les différentes vues, qui sont les suivantes :

- ✓ La vue fonctionnelle ou interactive : elle est représentée par des diagrammes de cas ou de séquence, elle cherche à capter les interactions entre les différents acteurs, afin d'atteindre les scénarios.
- ✓ La vue structurelle ou statique : réunit les diagrammes de classes et les diagrammes de packages. Les premiers favorisent la structuration des données et tentent d'identifier les objets/composants constituant le programme, leurs attributs, opérations et méthodes, ainsi que les liens ou associations qui les unissent. Les seconds s'attachent à regrouper les classes fortement liées entre elles en des composants les plus autonomes possibles.
- ✓ La vue dynamique : qui est exprimée par les diagrammes d'états. Cette vue est plus algorithmique et orientée « traitement », elle vise à décrire l'évolution (la dynamique) des objets complexes du programme tout au long de leur cycle de vie. De leur naissance à leur mort. Le diagramme d'activité est une sorte d'organigramme correspondant à une version simplifiée du diagramme d'états. Il permet de modéliser des activités qui se déroulent en

parallèle les unes aux autres, quand ce parallélisme peut poser problème. En général, les diagrammes d'états à eux seuls ne permettent pas de faire apparaître les problèmes spécifiques posés par la synchronisation des processus en concurrence, pour assurer la cohérence du comportement et l'absence d'inter blocage.

## **2. Les diagrammes UML :**

### **2.1. Le diagramme des cas :(vue fonctionnelle)**

Les cas d'utilisation (use case) modélisent une interaction entre le système informatique à développer et un utilisateur ou acteur interagissant avec le système. Plus précisément, un cas d'utilisation décrit une séquence d'actions réalisées par le système qui produit un résultat observable pour un acteur.

### **2.2. Le diagramme de classes : (vue structurelle)**

Un diagramme des classes décrit le type des objets ou données du système ainsi que les différentes formes des relations statiques qui les relient entre eux. On distingue classiquement deux types principaux de relations entre objets :

- ✓ Les associations, utilisées dans les modèles entité/association.
- ✓ Les sous-types, particulièrement en vogue en conception orientée objets, puisqu'ils s'expriment très bien à l'aide de l'héritage en programmation.

### **2.3. Les diagrammes de packages :**

En pratique, il s'agit de regrouper entre elles des classes liées les unes aux autres de manière à faciliter la maintenance ou l'évolution du projet et de rendre aussi indépendantes que possible les différentes parties d'un logiciel. Minimiser les liens inter-packages permet de confier la conception et le développement à des équipes séparées en évitant leurs interactions, donc l'éventuel cumul des délais, chaque équipe attendant qu'une autre ait terminé son travail. Les liens entre paquets sont exprimés par des relations de dépendance et sont représentés par une flèche en pointillé.

### **2.4. Les diagrammes de séquences : (vue fonctionnelle)**

Les diagrammes de séquences mettent en valeur les échanges de messages (déclenchant des événements) entre acteurs et objets (ou entre objets et objets) de manière chronologique, l'évolution du temps se lisant de haut en bas. Chaque colonne correspond à un objet (décrit dans le diagramme des classes), ou éventuellement à un acteur, introduit dans le diagramme des cas. La ligne de vie de l'objet représente la durée de son interaction avec les autres objets du diagramme. Un diagramme de séquences est un moyen semi-formel de capturer le comportement de tous les objets et acteurs impliqués dans un cas d'utilisation. On peut faire apparaître de nombreuses informations de contrôle le long de la ligne de vie d'un objet. Un diagramme des séquences permet de vérifier que tous les acteurs, les classes, les associations et les opérations ont bien été identifiés dans les diagrammes de cas et de classes.

## 2.5. Les diagrammes d'états : (vue dynamique)

Les diagrammes d'états décrivent tous les états possibles d'un objet. Ils indiquent en quoi ses changements d'états sont induits par des événements.

Si les diagrammes de séquences regroupent tous les objets impliqués dans un unique cas d'utilisation, les diagrammes d'états indiquent tous les changements d'états d'un seul objet à travers l'ensemble des cas d'utilisations dans lequel il est impliqué.

C'est donc une vue synthétique du fonctionnement dynamique d'un objet. Les diagrammes d'états identifient pour une classe donnée le comportement d'un objet tout au long de son cycle de vie (de la naissance ou état initial, symbolisée par le disque plein noir, à la mort ou état final, disque noir couronné de blanc).

## 2.6. Le diagramme d'activité :

Le diagramme d'activité est un cas particulier du diagramme d'états, dans lequel à chaque état correspond une activité constituant un élément d'une tâche globale à réaliser.

Le but de ce diagramme est de mettre en évidence les contraintes de « séquentialité » et de parallélisme qui pèsent sur la tâche globale.

## 3. Extension UML pour le Web :

L'extension d'UML pour le web définit un ensemble de stéréotype, d'étiquettes et de contraintes, qui rendent possible la modélisation web. Ces stéréotypes et ces contraintes sont appliqués sur certains des composants propres aux applications Web permettant ainsi de représenter au sein du même modèle et sur les mêmes diagrammes que ceux qui décrivent le reste du système.

### 1. Stéréotype :

C'est une extension du vocabulaire d'UML permettant l'association d'une nouvelle signification à un élément du modèle. Il est représenté par un chaîne de caractère entre guillemet (« »).

#### a) Classes :

- **Page serveur** « server page »

**Description:** elle représente une page web possédant des scripts exécutés par le serveur. Ils interagissent avec des ressources serveur telles que les bases de données.

**L'icône :**



**Contraintes :** Les pages serveur ne peuvent avoir de relation qu'avec des objets sur le serveur

**Étiquette** : Moteur de script qui peut être un langage ou le moteur qui doit être utilisé pour exécuter ou interpréter cette page.

- **Page client : « client page »**

**Description**: Une instance d'une page client est une page web formatée en HTML. Les pages client peuvent contenir des scripts interprétés par les navigateurs lorsque celles-ci sont restituées par ces derniers. Les fonctions des pages clients correspondent aux fonctions des scripts des pages web. Les pages clients peuvent avoir des associations avec d'autres pages clients ou serveurs.

**L'icône** :



**Contraintes**: Aucune.

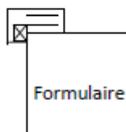
**Étiquette** :

- ✓ Titre (Title) : Titre de la page tel qu'il est affiché par le navigateur.
- ✓ Base (Base) : URL de base pour dé-référencer les URL relatives.
- ✓ Corps (Body) : Ensemble des attributs de la balise <body>, qui définit des caractéristiques par défaut du texte et de l'arrière-plan.

- **Formulaire «form » :**

**Description**: Une classe stéréotype « form » est un ensemble de champs de saisie faisant partie d'une page client. A une classe formulaire correspond une balise HTML « form », les attributs de cette classe correspondent aux éléments de saisie d'un formulaire HTML (zone de saisie, zone de texte, bouton d'option). Un formulaire n'a pas d'options : toute opération qui interagit avec le formulaire appartient à la page qui la contient.

**L'icône** :



**Contraintes** : Aucune.

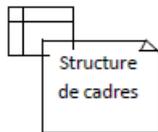
**Étiquette** : les méthodes GET ou POST : utilisée pour soumettre les données à L'URL de l'attribut action de la balise HTML <form>.

- **Structure de cadres : «frameset »**

**Description:** Une structure de cadres est un conteneur de plusieurs pages web.

La zone d'affichage rectangulaire est divisée en cadres rectangulaires, à chaque cadre peut être associé un nom unique de cible « Target ». Une classe stéréotypée « framset » est directement associée à une structure de cadres de pages web par la balise HTML <framset>.

**L'icône :**



**Contraintes:** Aucune.

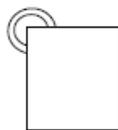
**Étiquette :**

- ✓ Rangées (rows) : Valeur de l'attribut rows de la balise HTML <framset>. C'est une chaîne de pourcentages séparés par des virgules, définissant les hauteurs relatives des cadres.
- ✓ Colonnes (cols) : Valeur de l'attribut cols de la balise HTML <frameset>. C'est une chaîne de pourcentages séparés par des virgules définissant les largeurs des cadres.

- **Cible : « Target »**

**Description:** Une cible est une zone nommée dans la fenêtre du navigateur dans laquelle des pages web peuvent être affichées. Le nom de la classe stéréotypée est ce lui de la cible. Habituellement, une cible est le cadre d'une structure défini dans une fenêtre ; ce pendant, une cible peut être une toute nouvelle instance du navigateur : une fenêtre. Une association « targeted link » spécifie la cible ou une page web doit être affichée.

**L'icône :**



**Contraintes:** Pour chaque client du système le nom de la cible doit être unique. Par conséquent sur un même client, il ne peut exister qu'une seule instance d'une même cible.

**Étiquettes:** Aucune

- **Objet JavaScript « JavaScript Object »**

**Description:** Sur un navigateur compatible java script. Il est possible de simuler des objets personnalisés à l'aide de fonctions JavaScript. Les objets java script ne peuvent exister que dans le contexte de page client.

**Icône:** Aucune.

**Contraintes:** Aucune.

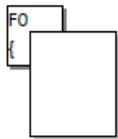
**Etiquettes:** Aucune.

**Icône :** aucune

- **Objet script client « Client Script Object »**

**Description:** Un objet script client est un ensemble qui regroupe des scripts particuliers dans un fichier, lequel est inclus dans une requête distincte du navigateur client. Ces objets regroupent des lots de fonction couramment utilisés à travers une application ou d'une entreprise.

**L'icône :**



*b) Associations :*

- **Lien « link »**

**Description:** Un lien est un pointeur d'une page client vers une autre page.

Dans un diagramme de classe, un lien est une association entre une page client et un autre page client ou une page serveur. A un lien correspond une balise HTML ancre.

**L'icône :** Aucune.

**Contraintes:** Aucune.

**Etiquettes:** La liste des noms de paramètres qui doivent être passés avec la demande de la page liée.

- **Lien cible « targeted link »**

**Description:** similaire à une association lien. Un lien cible est un lien dont la page associée est affichée dans une cible. A un lien cible correspond une balise HTML, dont l'attribut target prend la valeur de la cible.

**L'icône:** Aucune.

**Contraintes:** Aucune.

**Etiquettes:** Liste des de paramètres et le nom de la cible (target name), nom de la cible ou la page vers laquelle pointe le lien, doivent être passés avec la demande de la page liée.

- **Contenu de cadre « fram content »**

**Description :** Une association « contenu de cadre » est une association d'agrégation qui traduit l'apparence d'une page ou d'une cible à un cadre.

Une association contenue de cadre peut aussi pointer vers une structure de cadre aboutissant dans ce cas à des cadres imbriqués.

**Icône :** Aucune.

**Contraintes :** Aucune.

**Étiquette :**

- ✓ Rangée : entier qui indique la rangée du cadre dans la structure de cadres auquel appartient la page, ou la cible associée.
- ✓ Colonne : entier qui indique la colonne du cadre dans la structure des cadres auquel appartient la page, ou la cible associée.

- **Soumet « submit »**

**Description:** Une association de soumission « submit » se trouve toujours entre un formulaire et une page serveur. Les formulaires soumettent les valeurs de leurs champs au serveur pour qu'il les traite. Le serveur web traite la page serveur, qui accepte et utilise les informations du formulaire.

**L'icône:** Aucune.

**Contraintes :** Aucune.

**Étiquettes:** Listes des noms de paramètres qui doivent être passés avec la demande de la page liée.

- **Construit « build »**

**Description:** La relation « build » est une relation particulière qui fait le pont entre les pages clients et les pages serveurs. L'association « build » indique quelle page serveur est responsable de la création d'une page client. C'est une relation orientée, puisque la page client n'a pas connaissance de la page qui est à l'origine de son existence.

Une page serveur peut construire plusieurs pages client, en revanche, une page cliente ne peut être construite que par une seule page serveur.

**L'icône:** Aucune.

**Contraintes:** Aucune.

**Étiquettes:** Aucune.

- **Rediriger « redirect »**

**Description:** Une relation « redirect », est une association unidirectionnelle avec une autre page web, pouvant être dirigée à partir d'une page client ou serveur vers une page client ou serveur.

**L'icône :** Aucune.

**Contraintes:** Aucune.

**Étiquette:** le délai (delay), délai que doit observer une page client avant de se rediriger vers la page destination. Cette valeur correspond à l'attribut content de la balise <META>.

*c) Attributs :*

- **Éléments de saisie « input élément »**

**Description:** Un élément de saisie correspond à la balise <input> d'un formulaire HTML. Les étiquettes associées à cet attribut stéréotype correspondant aux attributs de la balise <input>. Les attributs obligatoires de la balise HTML <input> sont renseignés de la manière suivante : l'attribut « name » prend la valeur du nom de l'élément de saisie et l'attribut « value » prend celle de sa valeur initiale.

**L'icône :** Aucune.

**Contraintes:** Aucune.

**Étiquette :**

- ✓ Type (Type) : Le type de l'élément de saisie : texte, numérique, mot de passe, case à cocher, bouton d'option, bouton « submit » ou bouton « reset ».
- ✓ Taille (Size) : Définit la largeur visible allouée à l'écran en caractères.
- ✓ Longueur max (maxlength) : C'est le nombre maximal de caractères que peut saisir l'utilisateur.
- ✓ Sélection d'éléments « select element »

**Description:** Contrôle de saisie employé dans le formulaire, il permet à l'utilisateur de sélectionner une ou plusieurs valeurs dans une liste. La plupart des navigateurs restituent ce contrôle par une liste d'option ou une liste déroulante.

**L'icône:** Aucune

**Contraintes :** Aucune.

**Étiquette :**

- ✓ Taille(Size) : définit le nombre d'éléments qui doivent être affichés simultanément.
- ✓ Multiple (Multiple) : valeur booléenne qui indique que plusieurs éléments peuvent être sélectionnés conjointement.

- **Zone de texte « text area element »**

**Description:** C'est un contrôle de saisie, employé dans les formulaires, qui permet l'écriture de plusieurs lignes de texte.

**Icône:** Aucune.

**Étiquette :**

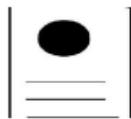
- ✓ Ligne (Rows) : Nombre de lignes de texte visibles.
- ✓ Colonne (cols) : Largeur visible du texte en largeur de caractères moyens.

*d) Composant*

- **Page web « web page »**

**Description:** Un composant page est une page web. Il peut contenir des scripts client ou serveur. Le plus souvent, le composant page est un fichier texte accessible au serveur web, mais il peut être un module compilé, chargé et exécuté par le serveur web. Dans les deux cas, le serveur web produit, à partir du composant page, un document au format HTML, qui est renvoyé en réponse à la requête du navigateur

**L'icône :**



**Contraintes:** Aucune.

**Étiquettes :** Chemin (path), chemin requis pour spécifier la page web sur le serveur web. Cette valeur doit être relative au répertoire racine du site de l'application web.

- **Page ASP « ASP page »**

**Description:** Une page ASP (Active Server Pages) est une page web qui implémente du code ASP coté serveur. Ce stéréotype n'est pertinent que dans un environnement d'application basé sur les ASP.

**L'icône :**



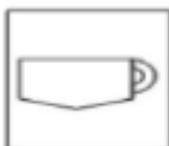
**Contraintes:** Aucune.

**Étiquettes:** Identique à celles de la page web.

- **Page JSP « JSP page »**

**Description:** Une page JSP (Java serveur Page) est une page Web qui implémente du code JSP coté serveur. Ce stéréotype n'est pertinent que dans un environnement d'application basé sur les JSP.

**L'icône :**



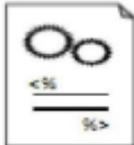
**Contraintes:** Aucune.

**Étiquettes:** Identique à celles de la page web.

- **Servelet « Servlet »**

**Description :** Un servlet est un composant Java. Il n'est pertinent que dans un environnement d'application compatible avec les servlets de SUN.

**L'icône :**



**Contraintes:** Aucune.

**Étiquettes:** Identique à celle de la page web

## 2. Les règles de cohérences sémantiques :

- **Réalisation de composant**

En principe, les composants pages web peuvent réaliser les classes stéréotypées « serveur page », « client page », « form », « JavaScript Object », « client script Object », « frameset » et « target ». Quand l'environnement de développement associé (ASP ou JSP) est en place, les pages web peuvent réaliser, au lieu des pages « serveur page », selon le cas, des classes stéréotypées « ASP page » ou « JSP page ».

- **Généralisation**

Tous les éléments de modélisation impliqués dans une même généralisation doivent être du même stéréotype.

- **Association**

Une page client peut avoir au plus une relation « build » avec une page serveur, mais une page serveur peut avoir plusieurs relations « build » avec différentes pages clients. En plus de la combinaison standard d'UML, les combinaisons de stéréotypes présentées au tableau ci-dessous sont permises :

DE \ A	« Client page »	« Serveur page »	« frameset »	« Target »	« form »
« client page »	« Link » « redirect » « Target Link »	« Link » « redirect » « Target Link »	« Link » « redirect » « Target Link »	Dependance	« agregation »
« Serveur page »	« build » « redirect »	« redirect »	« build » « redirect »		
« frameset »	« Frame content »		« Frame content »	« frame content »	
« Target »					
« form »	« aggregated by »	« Submit »			

Tableau : association entre page.

### 3. Avantages et inconvénient de l'UML :

**3.1. Les points forts de l'UML :** UML est un langage formel normalisé et un support de communication performant.

Son caractère polyvalent et sa souplesse en font un langage universel, il offre :

- ✓ Un gain de précision
- ✓ Un gage de stabilité
- ✓ Le cadrage de l'analyse.
- ✓ Une facilité de compréhension des représentations abstraites complexes.
- ✓ Son caractère polyvalent et sa souplesse en font un langage universel.

**3.2. Les points faibles de l'UML :** Même si UML n'est pas à l'origine des concepts objets, il en constitue une étape majeure, car il unifie les différentes approches et en donne une définition plus formelle. Ceci dit, UML présente tous de même quelques défauts parmi les quel on citera les suivants :

- ✓ La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation.
- ✓ Le processus (non couvert par UML) est une autre clé de la réussite d'un projet.

## 1. Introduction :

**Processus unifié (up)** Le processus unifié est un processus de développement logiciel : il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel.

Caractéristiques essentielles du processus unifié :

Le processus unifié est à base de composants,

- Le processus unifié utilise le langage UML (ensemble d'outils et de diagramme),
- Le processus unifié est piloté par les cas d'utilisation,
- Centré sur l'architecture,
- Itératif et incrémental

## 2. RUP (Rational Unified Process) :

- Dérivée de UP (UP a été créée en 1996)
- Commercialisée par IBM en 1998.

### 2.1. Méthode de développement logiciel :

- itérative,
- Incrémentale
- pilotée par les cas d'utilisation.
- centrée sur l'architecture et la réduction des risques
- Produit de qualité

### 2.2. Phases du RUP :

#### 2.2.1. Phase d'inception :

- Définir la portée du projet
- Spécification
- Affectation des tâches
- Evaluation des risques
- Cycle de vie

#### 2.2.2. Phase d'élaboration :

- Planifier le projet,
- Spécifier les fonctionnalités
- construire l'architecture
- Spécification du produit
- Conception de l'architecture
- Planification activités-ressources

### 2.2.3. Phase de construction :

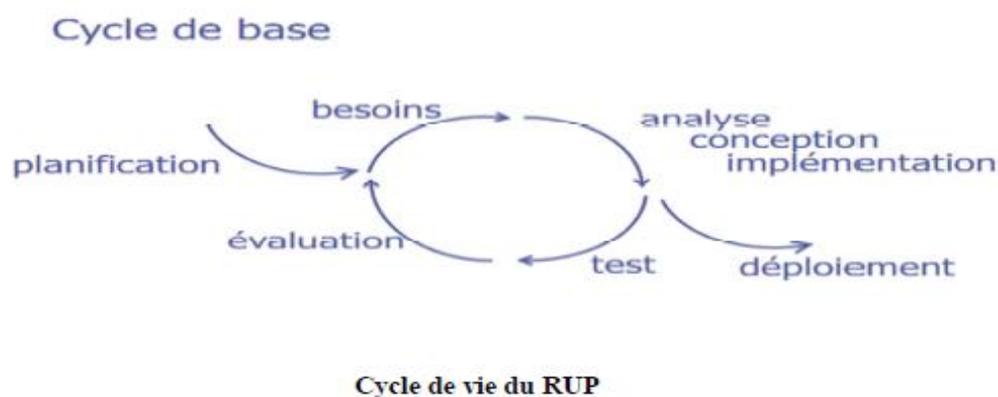
- Construire le produit
- Implémentation du produit
- Tests
- Produit opérationnel

### 2.2.4. Phase de transition :

- Transition du produit vers les utilisateurs
- Livraison
- Formation
- Qualité

### 2.3. Cycle de vie du RUP :

- **Exigence** détermination des besoins :
  - fonctionnels (ce que l'on attend du système)
  - non fonctionnels (fiabilité, temps de réponse, environnement distribué, etc.)
- **Analyse et conception** : évoluer depuis la spécification des besoins jusqu'à une solution informatique
  - analyse besoins fonctionnels
  - conception intègre aussi les besoins non fonctionnels
- **Implémentation** :
  - Transcription dans un langage de programmation ou de base de données.
  - Utilisation de composants existants
- **Test** : Estimer si les besoins sont satisfaits. S'il y a des erreurs/défauts à corriger Renforcer et stabiliser l'architecture
- **Déploiement** : Distribuer le logiciel dans son environnement opérationnel, Installation, test Formation des utilisateurs, Migration des données



#### 2.4. Différence entre incrémental et itératif

- **RUP est incrémental**

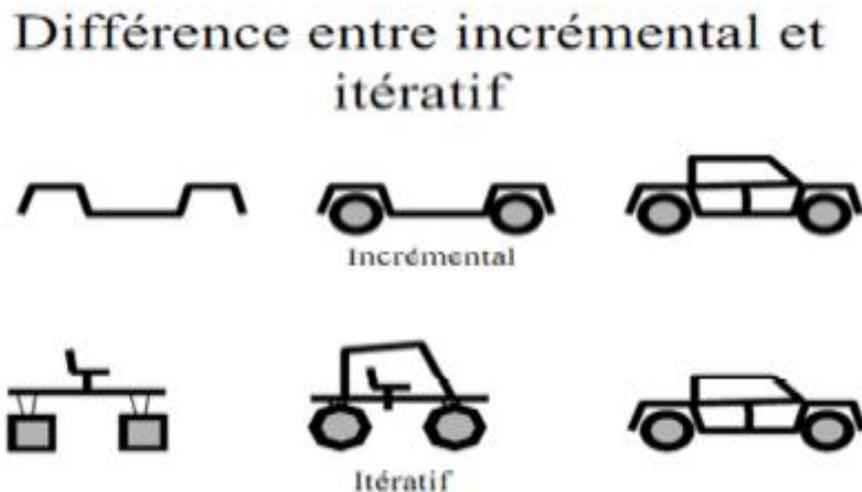
Le produit final est livré en incréments

- ✓ Chaque incrément livre une fonctionnalité ayant une valeur pour l'utilisateur final
- ✓ Chaque incrément consiste en un ensemble de cas d'utilisation
- ✓ Les incréments doivent être de courte durée (quelques semaines à quelques mois)
- ✓ Les incréments sont des mini-projets dans un projet

- **RUP est itératif**

Une itération est une séquence d'activités qui répond à un plan et à des critères d'évaluation et qui produit une version exécutable

#### Exemple



- **Forces de la méthode RUP**
  - ✓ Traçabilité à partir des Uses Cases jusqu'au déploiement
  - ✓ Approche basée sur l'architecture
  - ✓ Gestion des risques dans les projets
- **Faiblesses de la méthode RUP**
  - ✓ Coût de personnalisation souvent élevés
  - ✓ Très axé processus

### **3. Conclusion :**

Cette partie nous a permis de prendre connaissance le processus rup (rationnel unified process) est une méthode de développement pour logiciel orienté objet ,c'est une méthode générique itérative et incrémentale.

# Bibliographie

- Les réseaux. Claude Pichereau. Edition Foucher, 2002.
- Larry L. PETERSON, Bruce S. DAVIE, "Réseaux d'ordinateurs : *une approche système*". Vuibert, 1998.
- Internet Historique et Utilisation Nouvelle édition entièrement revue et augmentée  
Eric Lacher
- Architecture des réseaux par Bertrand Petit, Edition 2008.
- G. Gardarin, "Internet/Intranet et les bases de données Eyrolles, 2000.
  
- Architecture client/serveur par Eric CARION, 2010.
- Le guide de l'utilisateur UML par Grady BOOCH, James RUMBAUGH Edition :  
Eyrolles 2003.
  
- UML 2, Analyse et conception par Joseph Gobay et David Gobay, Edition : Dunod  
2008.
- Développons en java par Jeams Michel Doudoux.
  
- Introduction à SQL et My SQL par Sébastien NAMECHE [2003].
  
- application web et J2EE par pière Gambarotto 2011.

# Webiographie

- <https://www.google.fr/>
- <http://openclassrooms.com/courses?categories=informatique>
- <https://www.hibernate.org/>
- <https://www.eclipse.org/>
- <https://www.devlopper.com/>
- <https://www.devlopper.com/>
- <https://www.devlopper.com/>
- <https://www.MySQL.com/>
- <https://www.tomcat.apache.org/>
- <https://www.primfaces.org/>
- [www.commentcamarche.net/](http://www.commentcamarche.net/)

# *Remerciements*

Nous tenons à témoigner notre reconnaissance à DIEU tout puissant, qui nous a aidé et bénis par sa volonté durant toute cette période.

Notre profonde gratitude et sincères remerciements vont à notre promoteur Mr KERBICHE pour sa précieuse assistance, sa disponibilité et l'intérêt qu'elle a manifesté pour ce modeste travail.

Nos plus vifs remerciements vont à tout le personnel du Section maintenance réseau de l'entreprise Naftal de Tizi Ouzou Qui nous ont généreusement aidées durant notre stage.

Tous nos remerciement aux membres de jury qui nous font l'honneur de juger se modeste travail.

Nos remerciements vont également à tous ceux qui ont répondu aux questions sur des problèmes rencontrés.

Nos remerciements vont aussi à tous ceux qui ont contribué de près ou de loin à la réalisation de notre travail.