

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire de Fin d'Etudes de MASTER PROFESSIONNEL

Domaine : **Mathématiques et Informatique**

Filière : **Informatique**

Spécialité : **Ingénierie des systèmes d'information**

Présenté par

HAMACHE Daoud

MEFTAH Said

Thème

Conception et réalisation d'une application Java EE Cas : Gestion du personnel ENTP

Mémoire soutenu publiquement 11/07/2016 devant le jury composé de :

Président : Mr DIB

Encadreur : Mr KERBICHE

Examineur : Mme YESLI

Examineur : Mr CHEBOUBA

Remerciements

Nous remercions Dieu le tout puissant, qui nous à donner la santé et le courage jusqu'à l'aboutissement de notre travail fruit de ce cursus universitaire.

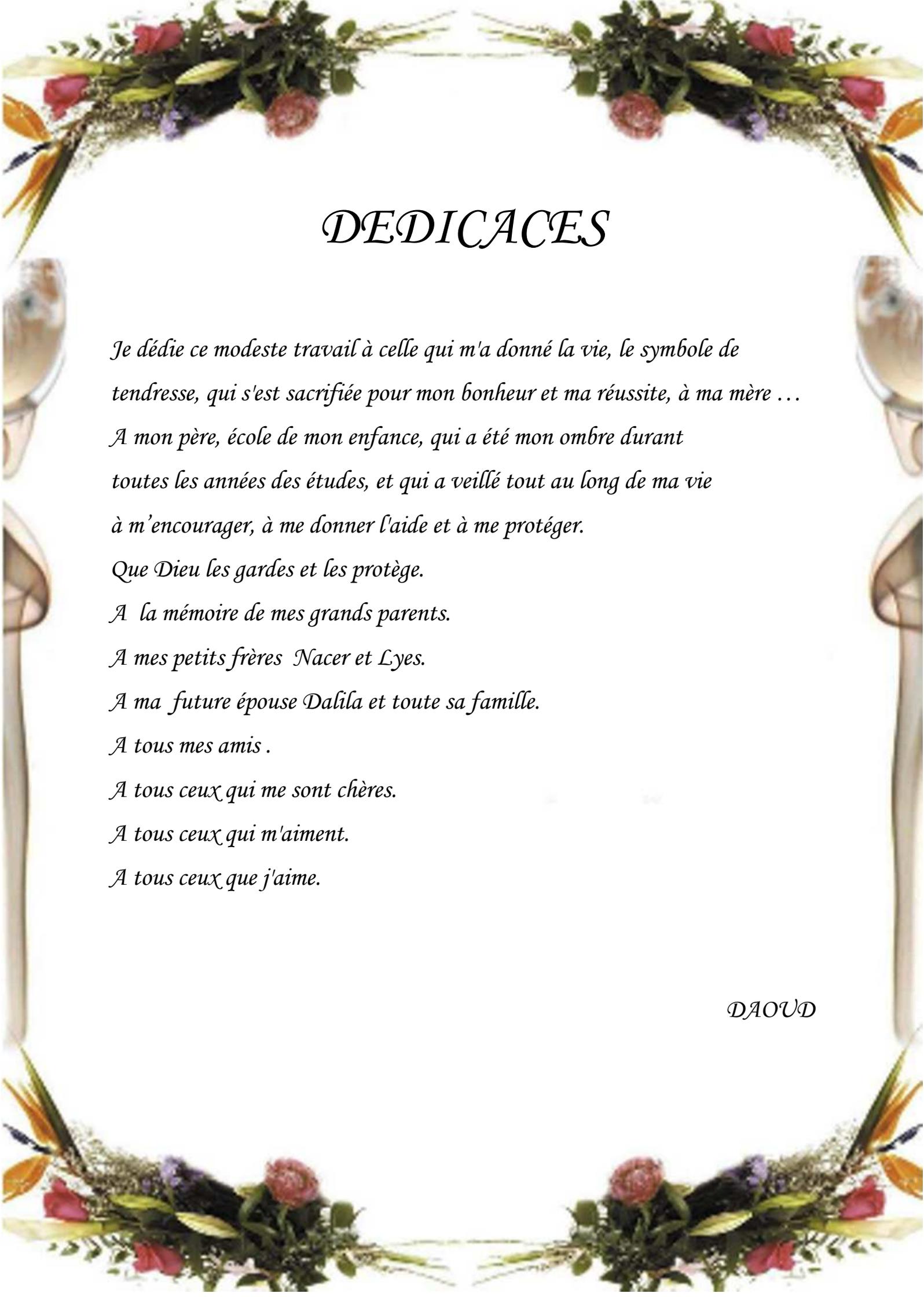
Nous tenons à présenter nos sincères remerciements à notre promoteur Mr KERBICHE qui a accepté de nous encadrer et d'avoir suivi notre travail et pour sa rigueur et sa disponibilité dans le suivi de ce modeste travail.

Je tiens également aremercier très chaleureusement notre encadreur Mr AMAACHE AHMED pour son accueil, son aide, son attention et sa gentillesse tout au long du stage, qui ont fait de ces trois mois un moment très plaisant et intéressant.

Nous remercions également tous ceux qui ont contribué de près ou de loin à la réalisation de ce projet.

Enfin nous remercions les membres du jury qui ont bien voulu nous honoré par leur présence afin d'évaluer nos efforts

Merci 



DEDICACES

Je dédie ce modeste travail à celle qui m'a donné la vie, le symbole de tendresse, qui s'est sacrifiée pour mon bonheur et ma réussite, à ma mère ...

À mon père, école de mon enfance, qui a été mon ombre durant toutes les années des études, et qui a veillé tout au long de ma vie à m'encourager, à me donner l'aide et à me protéger.

Que Dieu les gardes et les protège.

À la mémoire de mes grands parents.

À mes petits frères Nacer et Lyes.

À ma future épouse Dalila et toute sa famille.

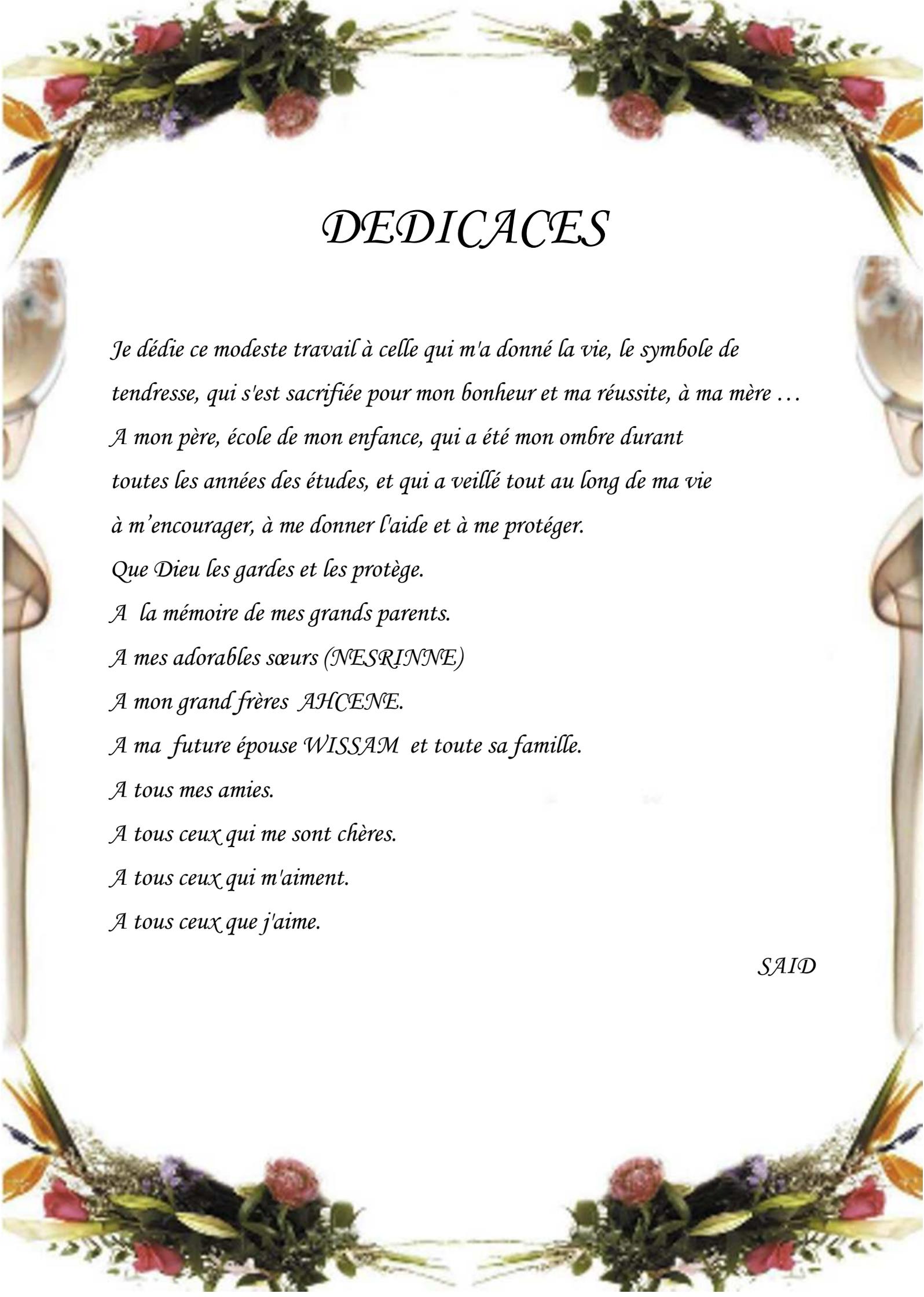
À tous mes amis .

À tous ceux qui me sont chères.

À tous ceux qui m'aiment.

À tous ceux que j'aime.

DAOUD



DEDICACES

Je dédie ce modeste travail à celle qui m'a donné la vie, le symbole de tendresse, qui s'est sacrifiée pour mon bonheur et ma réussite, à ma mère ...

À mon père, écolé de mon enfance, qui a été mon ombre durant toutes les années des études, et qui a veillé tout au long de ma vie à m'encourager, à me donner l'aide et à me protéger.

Que Dieu les gardes et les protège.

À la mémoire de mes grands parents.

À mes adorables sœurs (NESRINNE)

À mon grand frères AHCENE.

À ma future épouse WISSAM et toute sa famille.

À tous mes amies.

À tous ceux qui me sont chères.

À tous ceux qui m'aiment.

À tous ceux que j'aime.

SAID

Sommaire

Introduction Générale

Chapitre I : Présentation de la technologie JavaEE.

I. Introduction.....	1
I.1.Présentation de Java EE.....	2
I.2.L'architecture de Java EE.....	2
I.3. Les couches logicielles JEE.....	3
I.4.Spécification de la plateforme JEE.....	5
I.4.1. Les APIs.....	5
I.4.1.1. Les composants.....	6
I.4.1.1.1. Composants exécutés sur le client(client et applet).....	6
I.4.1.1.2. Composants exécutés sur le serveur.....	7
I.4.1.1.3. Composants métier(EJB).....	9
I.4.1.2. Services d'infrastructures.....	11
I.4.1.3. Services de communication.....	11
I.5. Les serveurs d'application.....	13
I.5.1. Le rôle d'un serveur d'application(ou bien les services).....	14
I.6. Les avantages d'utiliser JEE.....	15
I.7. Architecture client/serveur.....	15
I.7.1. Définition.....	15
I.7.2. Quelques notions de base.....	15
I.7.3. Le fonctionnement du système client/serveur.....	16
I.7.4. Classification des architectures client/serveur.....	16
I.7.5. Caractéristiques de l'architecture client/serveur.....	18
Conclusion.....	18

Chapitre II : Présentation de l'organisme d'accueil

Introduction.....	19
II-1-Présentation de l'ENTP.....	19
II-1-Historique.....	19
II-2-Réglementation.....	20
II-3- Nature des activités de l'entreprise.....	20
II-4- Débouchés et place sur le marché.....	28
II-5- Principales Installations.....	28
II-6- Personnels et effectifs.....	29

Sommaire

II-7- Les Investissements	31
II-8- Les Approvisionnements	31
II-9- Renseignement sur les filiales et participations	32
II-10- Litiges : Néant	32
II-11- Contrats et engagements importants	32
II-12- Facteurs de Risques	33
Conclusion.....	33

Chapitre III : Analyse et conception

III Introduction	34
III.1 Phase d'Initialisation	34
III.2 Phase d'Elaboration	34
III.2.1 Identification des acteurs	35
III.2.2 Diagramme de contexte.....	35
III.2.3 Représentation des diagrammes de cas d'utilisation	35
III.2.4 Définition des itérations	37
III.3 Construction	37
III.3.1 Réalisation de l'itération 1	37
III.3.1.1 Diagrammes de séquence pour itération 1	37
III.3.1.2 Diagrammes de classe pour l'itération 1	40
III.3.2 Réalisation de l'itération 2	40
III.3.2.1 Diagrammes de séquence pour itération 2	40
III.3.2.2 Diagrammes de classe pour l'itération 2	42
III.3.3 Réalisation de l'itération 3	42
III.3.3.1 Diagrammes de séquence pour itération 3	42
III.3.3.2 Diagrammes de classe pour l'itération 3	48
III.3.2 Réalisation de l'itération 4	49
III.3.2.1 Diagrammes de séquence pour itération 4	49
III.3.2.2 Diagrammes de classe pour l'itération 4	54
Conclusion	55

Sommaire

Chapitre IV Réalisation

Introduction	56
IV.1.Les outils de développement.....	56
IV.2 Les langages de programmation	59
IV.3 les Itérations	60
IV.3.1 Itération I	60
IV.3.2 Itération II	61
IV.3.3 Itération III	61
IV.3.4 Itération IV	63
IV.4 Présentation de quelques interfaces	64
IV.4.1 Présentation d'interface authentification	64
IV.4.2 Présentation d'interface page d'accueil	65
IV.3.3 Présentation d'interface d'un employé	65
IV.3.4 Présentation d'interface d'ajouter un employé	66
Conclusion	66

Conclusion générale	67
----------------------------------	-----------

Bibliographie

Annexes

Liste des figure

Chapitre I :

Figure I.1: Architecture et les composants de Java EE.....	3
Figure I.2 : Les couches logicielles d'une application Java EE	5
Figure I.3: Cycle de vie d'une servlet	7
Figure I.4 : L'architecture générale d'un EJB.....	10
Figure I.5 : Architecture Client/serveur	16
Figure I.6 : Architecture a deux niveaux.....	17
Figure I.7 : Architecture à trois niveaux	17
Figure I.8 : Architecture à multi niveaux.....	18

Chapitre II :

Figures II.1 : Volume d'activité de forage en nombre de puits	22
Figures II.2 : Volume d'Activité de Work-Over en nombre de puit.....	23
Figures II.3 : Evolution du chiffre d'affaire Activité Forage et Workover	24
Figures II.4 : Répartition de l'effectif de l'ENTP par métier au 31/12/2015	30
Figures II.5 : Evolution de l'effectif de l'ENTP en nombre d'agent	31

Chapitre III :

Figure III.1: Diagramme de contexte	35
Figure III.2 : Diagramme de cas d'utilisation.....	36
Figure III.3 : Diagramme de séquence du cas d'utilisation : « Authentification »	38
Figure III.4 : Diagramme de séquence du cas d'utilisation : «Modification du mot de passe».....	39
Figure III.5 : Diagramme de classes pour l'itération 1	40
Figure III.6 : Diagramme de séquence du cas d'utilisation : «Ajouter un employée »	41
Figure III.7 : Diagramme de classes pour l'itération2	42
Figure III.8 : Diagramme de séquence du cas d'utilisation : «Ajouter une mission ».....	43
Figure III.9 : Diagramme de séquence du cas d'utilisation : «Modifie une formation ».....	45
Figure III.10 : Diagramme de séquence du cas d'utilisation : «Supprimé un contrat»	47
Figure III.11 : Diagramme de classes pour l'itération3	48
Figure III.12 : Diagramme de séquence du cas d'utilisation : «Ajouter une Absence »	49
Figure III.13 : Diagramme de séquence du cas d'utilisation : «Modifie une sanction »	51
Figure III.14 : Diagramme de séquence du cas d'utilisation : «Supprimé un congé »	53
Figure III.15: Diagramme de classes pour l'itération4	54

Chapitre IV :

Figure IV.1: Interface GlassFish	56
Figure IV.2 : Interface de WampServer	57
Figure IV.3 : Interface dePhpMyAdmin	58
Figure IV.4 : Interface NetBeans	59
Figure IV.5: Interface de JEE.....	60
Figure IV.6 : Interface page d'Authenfication.....	64
Figure IV.7 : Interface page d'accueil	65
Figure IV.8 : Interface d'un employé	65
Figure IV.9 : Interface d'ajout d'un employé	66

Liste des tableaux :

Chapitre II :

Tableaux II.1 :Volume de production en termes de nombre de puits.....	22
Tableaux II.2 :L'évolution des principaux paramètres d'exploitation	23
Tableaux II.3 :L'évolution des paramètres d'exploitation pour les exercices 2014 et 2015.....	25
Tableaux II.4 :La division transport	25
Tableaux II.5 :Le chiffre d'affaires réalisé par l'activité transport durant les exercices 2014 et 2015	26
Tableaux II.6 :Le chiffre d'affaires réalisé par l'activité hôtellerie et le catering durant les exercices 2014 et 2015	27
Tableaux II.7 : Les sites de l'ENTP	29
Tableaux II.8 : Structure de l'ENTP	29
Tableaux II.9 : répartition de l'effectif de l'ENTP	29
Tableaux II.10 :La répartition par métier au 31 décembre 2015	30
Tableaux II.11 :L'évolution des effectifs entre 2010 est 2015	30
Tableaux II.12 : Les investissements de l'ENTP, Durant l'exercice 2015	31

INTRODUCTION GENERALE

Introduction générale

De nos jours, l'intérêt de l'utilisation de l'informatique est primordial dans tous les domaines, dans tous les organismes.

Il est clair que le système d'information joue un rôle important au sein de l'entreprise, ce qui donne une place privilégiée à l'informatique. La mise en œuvre de cette technologie moderne de traitement de l'information s'avère nécessaire, pour assurer une gestion plus fiable ainsi pour minimiser le risque d'erreurs et d'assurer la disponibilité de l'information à tout instant.

L'ENTP (L'Entreprise Nationale des Travaux aux Puits) est une entreprise nationale Algérienne spécialisée dans les activités pétrolières.

Après l'analyse de la méthode de travail de cet organisme et particulièrement la section gestion du personnel, on distingue qu'elle est soumise à un ensemble de problèmes dont:

- Le volume très important des documents entrants et sortants au niveau de ce Service.
- La lenteur de recherche des informations due à l'usage d'un grand nombre de documents ce qui rend ce suivi non fiable engendrant le gestionnaire à commettre des erreurs.
- Traitement manuel des données ce qui entraîne des risques d'erreur.
- La plus grande partie des informations est archivée sur les registres ou des classeurs.

Et pour remédier à ces manques, il nous a été demandé de concevoir un système d'information automatisé pour la gestion du personnel.

Le souci de l'ENTP est de disposer d'un moyen efficace et fiable de traitement, de communication, de centralisation de l'information et d'offrir aux futurs utilisateurs un environnement de communication et d'accès à l'information en temps réel.

Ce travail qui nous a été confié va dans ce sens et consiste en la réalisation d'une application client/serveur pour la gestion du personnel, Pour cela nous avons utilisé une démarche de conception orientée objet (RUP, Rational Unified Process), en se basant sur la modélisation UML, le SGBD MySQL pour l'implémentation de la base de données et Java EE pour la réalisation du logiciel.

Pour mener à bien notre travail, nous avons organisé notre projet en 4 chapitres :

Chapitre 1 : Introduction à la technologie Java EE.

Chapitre 2 : Présentation de l'organisme d'accueil.

Chapitre 3 : Analyse et conception.

Chapitre 4 : Réalisation.

Et on termine par une conclusion générale et une annexe qui porte sur le langage

Chapitre

I

*Introduction a la technologie
Java EE*

I-Introduction :

Dans ce chapitre nous allons présenter l'architecture Java EE ainsi que les notions de base nécessaires à sa compréhension.

I-1-Présentation de JAVA EE :

Java EE est une version de java destinée aux applications d'entreprise, proposée par la société Sun, portée par un consortium de sociétés internationales, dans le but de définir un standard de développement d'applications d'entreprises multi-niveaux, basées sur des composants.

Pour désigner l'ensemble constitué des services (API) offerts et de l'infrastructure d'exécution, nous parlons généralement de «plate-forme Java EE».

JAVA EE comprend notamment :

- Les spécifications du serveur d'application, c'est-à-dire l'environnement d'exécution : Java EE définit les rôles et les interfaces pour les applications ainsi que l'environnement dans lequel elles seront exécutées.
- Des services au travers d'API, c'est-à-dire des extensions Java indépendantes permettant d'offrir en standard un certain nombre de fonctionnalités. Sun fournit une implémentation minimale de ces API appelée Java EESDK (Java EE Software Développement Kit).

JAVA EE s'appuie entièrement sur le Java, il bénéficie des avantages et inconvénients de ce langage, ce qui engendre une bonne portabilité et une maintenabilité du code. L'architecture JAVA EE s'appuie sur des composants différents, interchangeables et distribués, ce qui permet :

- ❖ D'étendre l'architecture d'une façon simple.
- ❖ Que chaque système qui repose sur Java EE peut posséder des mécanismes de haute disponibilité, afin de garantir une bonne qualité de service.
- ❖ La facilité de la maintenabilité des applications.

I.2- L'architecture de JAVA EE : [1]

JAVA EE ajoute de nombreuses couches de niveau entreprise au-dessus de la plateforme JSE-Java Standard Edition. Chaque couche est conçue pour supporter une différente technologie de développement.

1-Technologie web application : est une technologie liée à la création des interfaces web dynamique, par exemple : JSP (Java Server Page) et servlet.

2-Technologie entreprise application : est une technologie liée à la logique de business : EJB (Entreprise Java Bean), javaMail, JMS (Java Message Service)...etc.

3-Technologie web services : est une technologie nécessaire au développement des applications adhérentes au paradigme SOA (Service Oriented Architecture) .Elle permet l'échange des données entre application et systèmes hétérogènes dans des environnements distribués.

4-Technologie management et sécurité: est une technologie liée à la gestion de la technologie d'entreprise afin de réaliser l'accès et l'échange d'informations entre machine et service distribués : JAAS (Java Authentication and AuthorizationService), JCA (Java Connector Architecture).

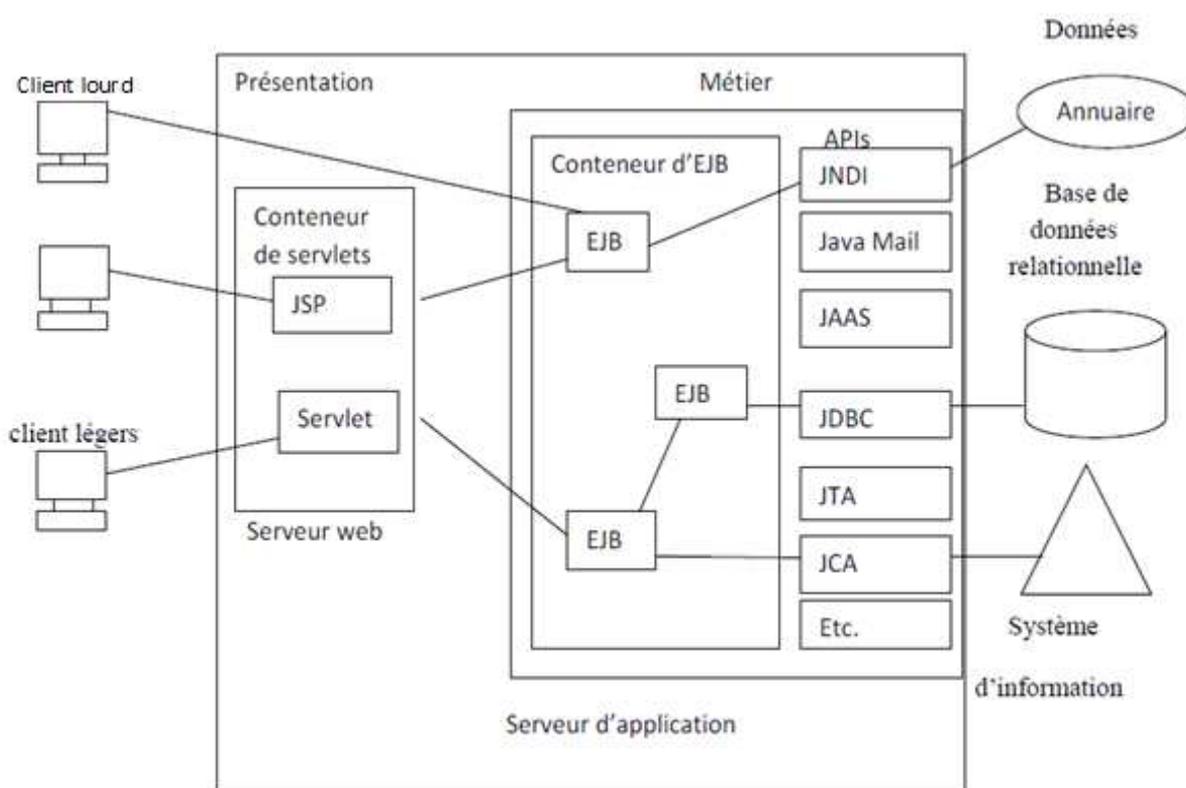


Figure I.1: Architecture et les composants de Java EE

I.3- Les couches logicielles Java EE :

La plateforme Java EE utilise un modèle d'application distribuée multi-tiers pour des applications d'entreprise. La logique d'application est divisée en composants selon la fonction. Les divers composants qui forment l'application Java EE sont installés sur des machines différentes en fonction du niveau de l'environnement Java EE multi tiers auquel le composant d'application appartient, la figure suivante montre deux types d'applications Java EE réparties dans des niveaux décrit comme suit :

1-La couche client : Ses composants s'exécutent sur la machine du client. Ex : logiciel installé en local ou navigateur web.

2-La couche WEB : Ses composants s'exécutent sur le serveur Java EE. Ex :servlet et jsp.

3-La couche métier : Ses composants s'exécutent sur le serveur Java EE. Ex :EJB.

4-La couche EIS (Entreprise Informatique System) : Ses composant s'exécutent sur le serveur EIS (pour le stockage des informations).

Comme le montre la figure suivante, une application Java EE peut être constituée par trois ou quatre couches, les applications multi tiers sont considérées comme les applications 3-tiers car, elles sont distribuées sur trois emplacements :

1-La machine client.

2-La machine pour serveur Java EE.

3-La machine pour serveur de base de données.

Les applications à 3-tiers qui fonctionnent de cette façon étendent le standard 2-tiers : Client/serveur en plaçant un serveur d'application entre l'application de client et le stockage principale.

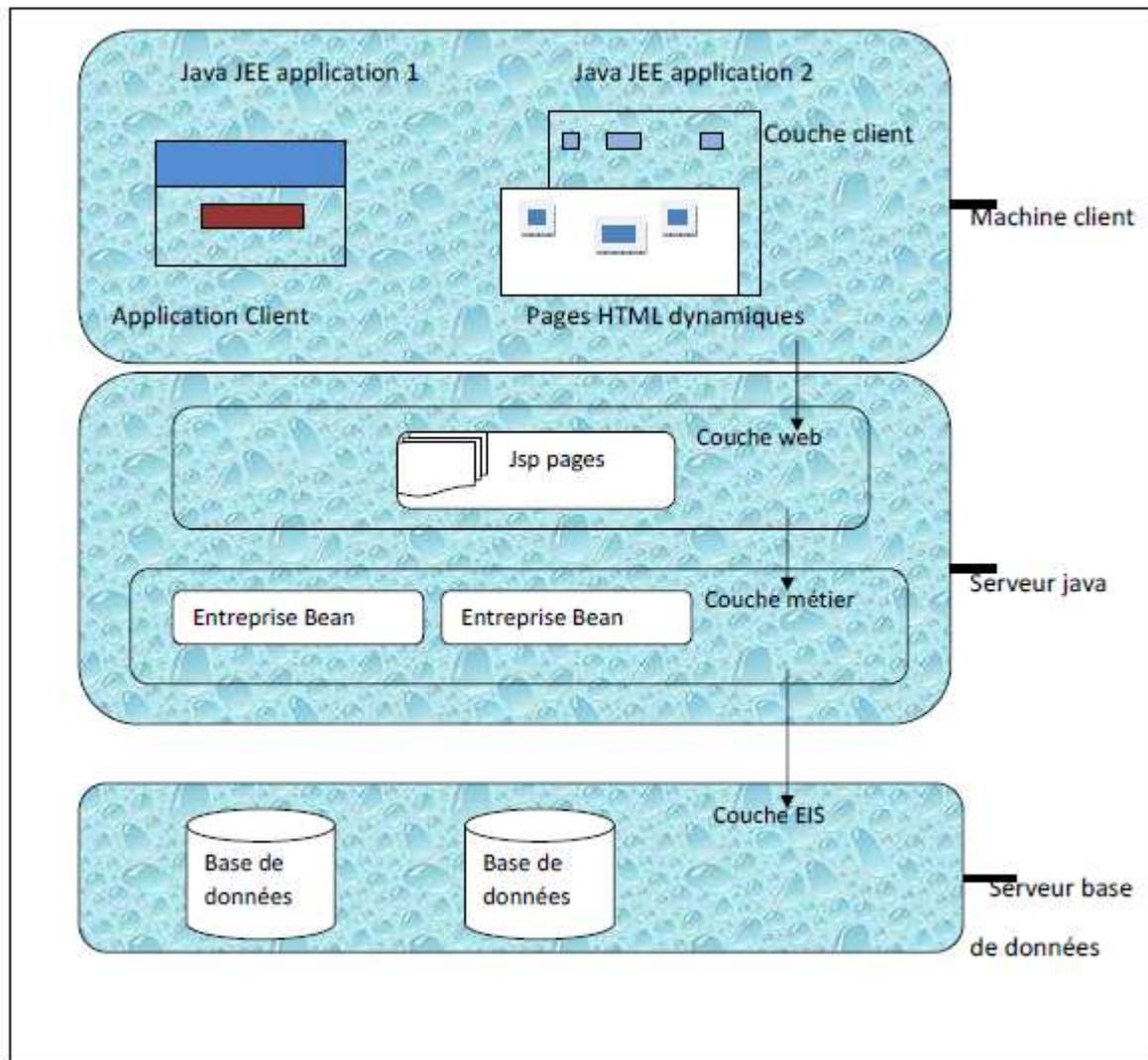


Figure I.2 : Les couches logicielles d'une application Java EE.

I.4- Spécification de la plateforme Java EE :

Java EE compte des spécifications qu'on peut regrouper en deux grandes parties :

- Les APIs.
- Les serveurs d'application.

I.4.1- Les APIs :

Les APIs sont des extensions Java, forment une collection d'outils logiciels permettant d'offrir en standard un certain nombre de fonctionnalités. Sun fournit une implémentation minimale de ces APIs nommée Java EE SDK (Java EE SOFTWARE DEVELOPMENT KIT), ces APIs peuvent être regroupées en trois groupes principaux :

- Composants.
- Services d'infrastructures.
- Services de communication.

I.4.1.1- Les composants :

Un composant est un fichier simple, contenant généralement du code compilé. Il est possible de créer des composants avec la majorité des langages. Aussi il peut être vu comme unité regroupant un certain nombre de fonctionnalités qui peuvent être appelées depuis un programme externe ou client.

Remarque :

Pour les composants JavaEE écrits en java et compilés sont assemblés en une application Java EE déployés, exécutés et gérés par le serveur Java EE. Une séparation peut se faire entre les composants existants selon leur exécution :

I.4.1.1.1-Composants exécutés sur le client (client et applet) :

Peuvent être application client ou applet.

❖ Client JavaEE :

Un client Java EE peut être client web ou application client.

❖ Client web :

Un client web est appelé client léger qui n'a généralement pas de requête de BDD. L'utilisation de client web permet de charger les opérations lourdes au niveau de l'entreprise bean et sont exécutées sur le serveur Java EE ou ils peuvent exploiter les avantages des technologies Java EE (sécurité, la vitesse, les services...) sur le serveur. Un client web a besoin de deux fonctionnalités principales :

- Les pages web dynamiques avec les différents langages à balisage (HTML, XML, ...), ces pages sont générés par les composants web au niveau du client web.
- Le navigateur web qui affiche les pages reçues par le serveur.

3-Application client

L'application client fonctionne au niveau de la couche métier, et peut accéder directement à l'entreprise Bean. Généralement les utilisateurs possèdent une interface graphique (GUI) pour exécuter l'application client (en utilisant l'API Swing ou AbstractWindows Toolkit).

4-Applet :

Une applet est une petite application java qui a les caractéristiques de pouvoir s'exécuter via un navigateur supportant java ou dans une l'appletviewer du JDK. En effet, la plupart des navigateurs aujourd'hui disposent d'une machine virtuelle capable d'interpréter les bytecode inclus dans une page HTML.

Il existe plusieurs différences entre application et applets. En effet, les applets fournissent un certain nombre de fonctionnalités supplémentaires propres à leur contexte d'utilisation, comme la possibilité de communiquer avec le navigateur ou même entre applets. Mais la plus notable réside dans la sécurité qui leur est associée.

I.4.1.1.2- Composants exécutés sur le serveur :

Les composants web sont :

1- **Composants WEB**: Sont des servlets ou des JSP.

2- **Servlet** ou « un peu d'HTML dans beaucoup de Java... » :

Une servlet est un composant coté serveur, considérée comme une classe java qui permet une communication avec un protocole « requête/réponse ». Autrement dit, une servlet est un programme qui s'exécute coté serveur. Elle reçoit une requête du client, elle effectue des traitements et renvoie le résultat.

L'utilisation des servlets permet la génération des pages web dynamiques qui utilisent le protocole HTTP, ou aussi n'importe quel autre protocole reposant sur le principe de servlet. Une servlet est une classe java qui implémente les méthodes `doPost` et/ou `doGet`.

3- **Cycle de vie d'une servlet :**

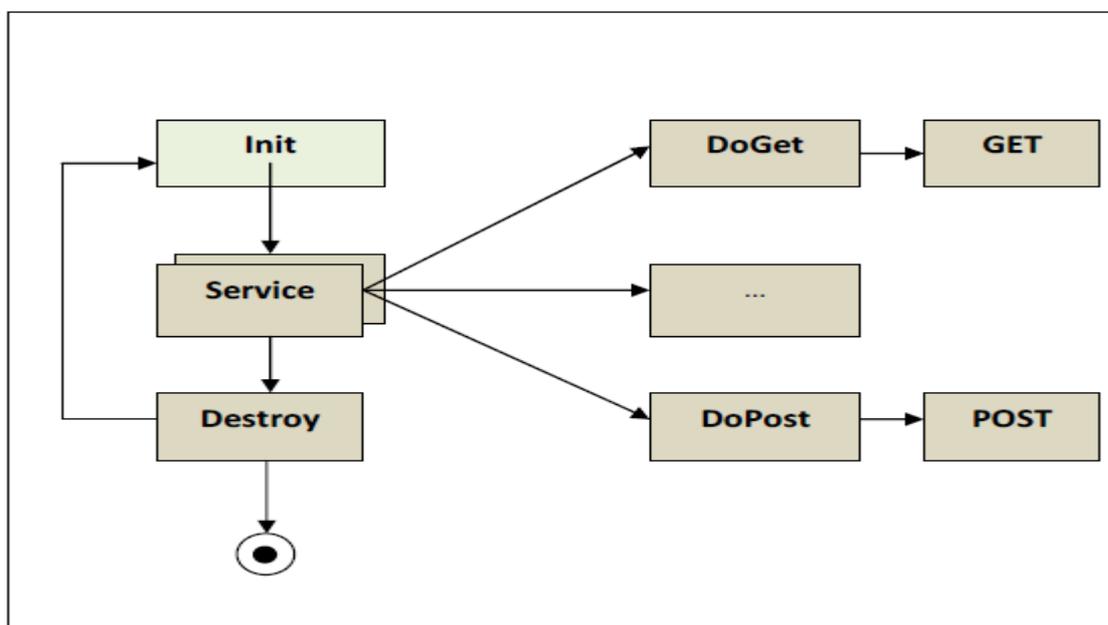


Figure I.3: Cycle de vie d'une servlet

- **Init ()** : initialisation de la servlet chargement du code.

Souvent effectué lors de la première requête cliente (`doGet`, `doPost`) Allocation d'un pool de threads.

- **doGet ()** : Traitement des requêtes HTTP GET et les convertir via la méthode service en Get.
 - **doPut ()** : Traitement des requêtes HTTP PUT.
 - **doPost ()** : Traitement des requêtes HTTP POST et les convertir par la méthode service en Post.
 - **destroy ()** : Destruction de la servlet par le serveur.
- La méthode **Init** est appelée une seule fois au chargement de la servlet.
-La méthode **Service** est appelée à chaque requête http (client).
-La méthode **Destroy** est appelée une seule fois lorsque le conteneur détruit la servlet.

4-Avantages et inconvénients des servlets :

❖ **Avantage :**

- Portabilité (car s'exécute sur une JVM (java Virtual machine)).
- S'exécute cote serveur.
- Maintenabilité.
- Elle maintient des sessions de plusieurs clients http grâce l'objet http session.
- Multithread/contrairement à PHP exécute plusieurs fois.

❖ **Inconvénients :**

- Très difficile d'utiliser les outils d'édition HTML pour traiter les aspects graphiques.

5- JSP : Java Server Page ou « un peu de java dans beaucoup d'HTML... »

Les JSPs : (pour java Server Pages) est une technologie qui permet d'écrire très facilement des pages dynamiques en insérant des portions de code java dans une page html. Destiné à l'origine à la présentation des données, une page JSP permet de définir le design d'une page (en HTML) et d'y intégrer les données (grâce aux portions de code java). cette technologie favorise donc la séparation de la présentation et du contenu.

D'un point de vue plus technique, une page java est donc constituée de code HTML dans lequel sont insérées des codes java délimitées par des balises particulières.

L'intégralité du code est portable puisque tout comme un code java, il sera compilé et exécuté comme Servlet.

A la différence d'une Servlet, il sera par contre compilé au vol (lors de chaque invocation par un moteur de JSP. Enfin, les pages JSP offrent les possibilités d'importer des classes java et d'accéder à des objets java.

6-JSF :

Java Server Faces (JSF) est une technologie dont le but est de proposer un framework qui facilite et standardise le développement d'applications web avec Java. Son développement a tenu compte des différentes expériences acquises lors de l'utilisation des technologies standard pour le développement d'applications web (servlet, JSP, JSTL) et de différents frameworks (Struts, ...). Java Server Faces exécutés sur le serveur, mais sont affichés sur le client.

I.4.1.1.3- Composants métier (EJB) :

1-Entreprise JavaBeans (EJB) :

Est une architecture de composants logiciels côté serveur pour la plateforme de développement Java EE. Cette architecture propose un cadre pour créer des composants distribués (c'est-à-dire déployés sur des serveurs distants) écrit en langage de programmation Java hébergés au sein d'un serveur applicatif permettant de représenter des données (EJB dit entité), de proposer des services avec ou sans conservation d'état entre les appels (EJB dit session), ou encore d'accomplir des tâches de manière asynchrone (EJB dit message). Tous les EJB peuvent évoluer dans un contexte transactionnel.

L'architecture EJB prévoit en de manipuler les composants EJB dans un environnement distribué, assurant une complète comptabilité avec tous les types de clients : servlet, page jsp, autre EJB, applet...

Pour assurer la communication réseau entre différents composants distribués les EJB utilisent un mécanisme basé sur RMI. Ce mécanisme est implémenté dans une API qui offre à une machine virtuelle Java les mécanismes nécessaires pour invoquer des méthodes sur d'autres JVM, qu'elles soient ou non sur le même réseau.

2-Les différents types d'EJB:[2]

Il existe trois types de EJB :

- **Les beans entités :**

Ca correspond à des objets réels tels que ces objets sont généralement des enregistrements persistants stockés dans une BDD. Les beans entités peuvent être des moyens de modélisation des concepts métier pouvant être désignés par des noms, par exemple : un bean entité peut représenter un client ou un objet stocké.

- **Les beans session :**

Ceux sont une extension de l'application client et ils sont responsables de la gestion des processus ou des tâches.

- **Les beans orientés message :**

Un bean session fournit une interface distante afin de définir les méthodes que l'on peut invoquer, alors que bean orienté message attend des messages asynchrones spécifiques auquel il répond.

Donc généralement, un EJB récupère également les données stockées, les traite et les envoie au programme du client.

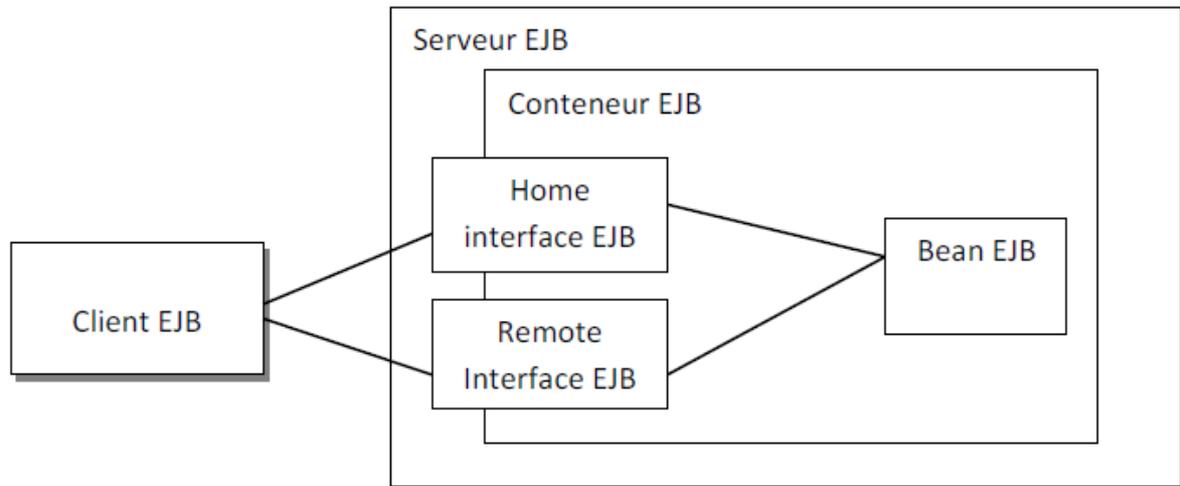


Figure I.4 : L'architecture générale d'un EJB.

3-Serveur EJB :

EJB Server intègre le conteneur EJB assurant les services requis par le composant de ce dernier.

4-Client EJB :

-Le client EJB fournit généralement la logique de l'interface utilisateur sur la machine cliente. Il passe les appels aux composants EJB distants hébergés sur un serveur et doit savoir comment trouver le serveur EJB et interagir avec les composants EJB.

-Un composant EJB peut faire office de client EJB en appelant les méthodes d'un autre composant EJB.

-Un client EJB ne communique pas directement avec un composant EJB.

-Le conteneur fournit les objets proxy mettant en œuvre les Home Interface et Remote Interface des composants. Le Remote Interface des composants définit les méthodes de gestion pouvant être appelées par le client. Ce dernier appelle les méthodes de l'Home Interface pour créer et détruire les proxys pour le Remote Interface.

5-L'interface remote :

L'interface remote permet de définir les méthodes qui contiendront les traitements proposés par le bean. Cette interface doit étendre l'interface `java.ejb.EJBObject`.

```
Package com.jmdoudoux.ejb ;  
import java.rmi.RemoteException ;  
import javax.ejb.EJBObject ;  
public interface MonPremierEJB extends EJBObject {  
public String message () throws RemoteException ;  
}
```

6-L'interface home :

L'interface home permet de définir des méthodes qui vont gérer le cycle de vie d'un bean.

Cette interface doit étendre l'interface EJBHome.

```
Package com.jmdoudoux.ejb ;  
Import java.rmi.RemoteException ;  
Import javax.ejb.CreateException ;  
import javax.ejb.EJBHome ;  
public interface MonPremierEJBHome extends EJBHome {  
public MonPremierEJB create () throws CreateException, RemoteException;  
}
```

I.4.1.2- Services d'infrastructures :[3]

On trouve les services suivants :

- **JDBC « Java DataBase Connectivity »**

C'est une API qui permet aux programmes java d'interagir avec les bases de données SQL. Les serveurs fournissent en plus un mécanisme qui permet de réutiliser les connexions créées avec les utilisations.

- **JTA/JTS « Java Transaction Api/Java Transaction Services »**

C'est une API qui définit des interfaces standard avec un gestionnaire de transaction, c'est-à-dire permettant à des applications et à des serveurs Java EE d'accéder à des transactions.

- **JCA « Java EE Connector Architecture »**

C'est une API de connexion au système d'information de l'entreprise, notamment aux systèmes dits « legacy » tel que les ERP (Entreprise Resource Planning).

- **JMX « Java Management Extension »**

Permet de développer des applications WEB de supervision d'applications à partir de extensions.

- **JNDI « Java Naming and Directory Interface »**

C'est une extension java standard qui fournit une API permettant de localiser et d'utiliser des ressources, il peut y avoir un appel à des services CORBA, DNS, NIS, LDAP...

I.4.1.3. Services de communication :

- **JMS « Java Message Service »**

A pour rôle l'envoi et la réception des messages entre deux composants d'application Java EE.

- **Java Mail TM Technology :**

Permet l'envoi des e-mails d'une manière indépendante de la plateforme et du protocole, donc envoi de courrier électronique.

- **RMI-IIOP « Remote Method Invocation/Internet Inter-ORB Protocol » :**

Cette API permet la communication synchrone entre objets.

- **Web Services :**

Les web services permettent de partager un ensemble de méthodes qui pourront être appelées à distance. Cette technologie utilise XML ce qui permet d'être utilisé par n'importe quel langage et n'importe quelle plateforme.

- **XML :**

N'est pas vraiment une API de Java EE mais son utilisation dans cette plateforme est de plus en plus importante comme son utilisation pour les fichiers de configuration. XML est la base d'un nouveau mode de communication entre les applications, permet de faciliter l'échange d'information sur l'internet.

Les avantages du XML sont multiples :

- **Lisibilité :** il est facile pour un humain de lire un fichier XML car le code est structuré et facile à comprendre. En principe, il est même possible de dire qu'aucune connaissance spécifique n'est nécessaire pour comprendre les données comprises à l'intérieur d'un document XML.
- **Disponibilité :** ce langage est libre et un fichier XML peut être créé à partir d'un simple logiciel de traitement de texte (un simple bloc-note suffit).
- **Interopérabilité :** Quel que soit le système d'exploitation ou les autres technologies, il n'y a pas de problème particulier pour lire ce langage.
- **Extensibilité :** De nouvelles balises peuvent être ajoutées à souhait.
- Plusieurs parseurs XML différents doivent en principe (s'ils sont bien codés) produire le même résultat.

- Tous les navigateurs internet récents intègrent un parseur XML, pour lire les documents de ce langage informatique.

Plusieurs type d'application sont possible avec le XML, tel que: les flux RSS, les documents ODF (utilisé par la suite open office), le MathML, le SVG, les RIA ...

- **Les conteneurs(en anglais containers) :**

Un conteneur est une entité qui fournit aux composants la gestion de cycle de vie, la sécurité, les services de déploiement et d'exécution. Chaque type de conteneur (EJB,WEB, JSP...) fournit aussi des services spécifiques à ces composants.

Il existe plusieurs types de conteneurs qu'on peut citer :

a-Conteneur web :

Il est composé de deux grandes parties : un moteur de servlets et un moteur de JSP, donc ce conteneur permet d'exécuter les composants JSP et les servlets de l'application Java EE.

b-Conteneur EJB :

Ce type de conteneur permet de gérer l'exécution d'EJB.

c-Conteneur d'application client

Il permet de gérer l'exécution des composants de l'application client. Il s'exécute sur le client.

d-Conteneur d'applets

C'est une composition de deux parties : navigateur web et java plug-in qui fonctionnent ensemble sur le client, donc ce conteneur permet de gérer l'exécution des applets.

I.5- Les serveurs d'application:

Les API Java EE ont été projetés pour travailler avec un particulier type de serveur appelé Java EE Application Server. Un serveur d'application est une couche logicielle résidente sur une machine serveur qui fournit les services que la technologie Java EE nécessite. Il y a plusieurs serveurs d'applications. Parmi les produits commerciaux on rappelle Bean WebLogic, IBM Websphere, Sun Application Server,...etc. Parmi les produits libres le plus connu est JBOSS. Les différences principales entre les différents serveurs d'applications sont relatives aux opérations de déploiement, clustering, ...etc. Par contre toutes les fonctionnalités qui concernent strictement le fonctionnement des applications Java EE adhèrent aux spécifications proposées par le Sun.

Un serveur d'application s'installe et se lance comme un normal serveur web (de fait il met à disposition des services typiques d'un serveur web). En plus il possède un panneau d'administration accessible par un poste distant à travers lequel on peut définir les différents services. On peut considérer le serveur d'application comme une application Java standalone exécutée par une Java EE qui dessert les requêtes provenant des différents clients. Les serveurs d'application se sont développés depuis la création de Java EE. Il existe principalement 2 grandes catégories de serveurs :

- **Open Source** : évolue grâce à la communauté par exemple **Tomcat : Apache, Jonas : ObjectWeb, JBoss : JBoss...**
- **Propriétaire**: évolue selon l'éditeur par exemple **WebSphere : IBM, WebLogic : BEA, Web Object : Apple, Oracle Application Server : Oracle...**

I.5.1-Le rôle d'un serveur d'application (ou bien les services) :

Le rôle d'un serveur d'application est d'héberger des applications distribuées, fabriquées à base de composants Java (Servlet, JSP, EJB) et de les rendre accessible à des clients web (navigateurs) et à des applications d'entreprise écrites en java. Le serveur d'application doit être performant et fiable pour satisfaire aux exigences des applications d'entreprise.

Le serveur offre différents services utiles pour les applications :

a. Service de nommage :

C'est le service le plus important dans un serveur d'application implémentant l'interface. SPI (Service Provider Interface) est utilisée généralement le protocole LDAP (Lightweight Directory Access Protocol). Ce dernier fournit des mécanismes de recherche, de localisation et de liaison et il est différent du DNS.

b. Services de gestion de transaction :

Il permet de gérer des transactions locales mais surtout il prend en charge la gestion des transactions distribuées, indispensable pour les applications Java EE.

c. Service de gestion de la disponibilité des applications :

Il permet de gérer un grand nombre d'accès et de requêtes simultanés.

d. Service de sécurité :

L'administrateur du serveur d'application peut configurer des domaines de sécurité en définissant des utilisateurs (nom, mot de passe), des groupes d'utilisateurs, puis les permissions accordées sur différents éléments par rapport aux composants et/ou aux ressources de l'application Java EE.

e. Service d'administration :

L'administrateur a la possibilité de consulter, de modifier, et d'ajouter des éléments à la configuration des applications d'entreprise et du serveur d'application.

f. Service d'accès aux données :

Ce service est une implémentation des extensions de l'API JDBC

g. Service de gestion de messages :

C'est une implémentation de l'API JMS, il permet de gérer une file de messages de traitements asynchrones.

I.6- Les avantages d'utiliser Java EE:

L'utilisation de Java EE pour développer et exécuter une application représente plusieurs avantages :

- Une architecture d'application basée sur les composants qui permet un découpage de l'application et donc une séparation des rôles lors du développement.
- La possibilité de s'interfacer avec le système d'information existant grâce à de nombreuses API : JDBC, JNDI, JMS, JCA...
- La possibilité de choisir les outils de développement et les serveurs d'applications utilisés qu'ils soient commerciaux ou libres.

I.7- Architecture client/serveur :

Il existe en informatique plusieurs applications qui s'exécutent selon une architecture client/serveur. Et cela indique que des machines clientes peut être en contact avec une machine serveur distante, qui répond aux services clients. Ces services sont des programmes qui fonctionnent en parallèle.

I.7.1-Définition : [4]

L'architecture client/serveur est une méthodologie informatique de communication entre plusieurs machines (clients) et une machine serveur. Ce dernier est installé quelque part dans le monde, il s'occupe de l'installation des services, le partage des périphériques, de mémoriser les données sur son disque dur, et les distribuer suite à une demande client.

I.7.2-Quelques notions de base : [5]

1-Client : est un processus qui est connecté à un serveur. Le client envoie une requête et reçoit la réponse immédiatement.

2-Serveur : c'est un ordinateur qui accueille des requêtes, les traite et renvoie des réponses sous forme de pages web aux clients. Il est toujours allumé et il fonctionne 24/24h.

3-Middleware : C'est un logiciel/matériel qui joue le rôle d'intermédiaire entre le client et le serveur et garantit la communication (le dialogue) entre eux.

4-Requête : est un message envoyé par un client à un serveur comportant le fait à exécuter.

5-Réponse : est un message émis par un serveur à un client comprenant les éléments de fait exécuté.

I.7.3-Le fonctionnement du système client/serveur : [6]

Le client envoie une requête vers le serveur à l'aide de son adresse IP et le port, qui désigne un service particulier du serveur. Quand le serveur reçoit la demande, une réponse est envoyée directement à l'aide de l'adresse de la machine client et son port.

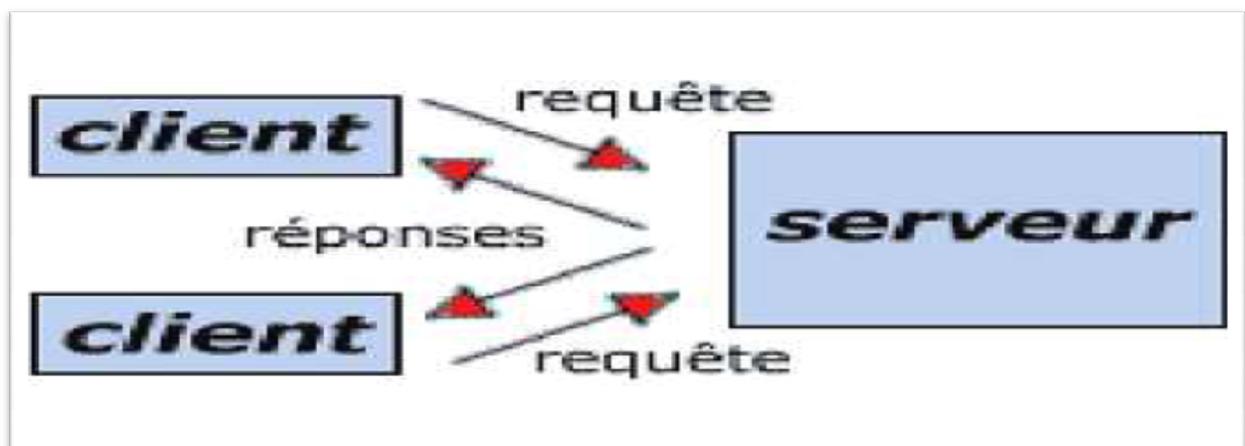


Figure I.5 : Architecture Client/serveur

I.7.4-Classification des architectures clients/serveurs :

I.7.4.1-Présentation de l'architecture à 2 niveaux :

L'architecture à deux niveaux décrit les systèmes clients/serveurs pour lesquels le client demande une ressource et le serveur la lui envoie instantanément, en utilisant ses propres ressources. Sans appeler une autre application pour répondre au service demandé.

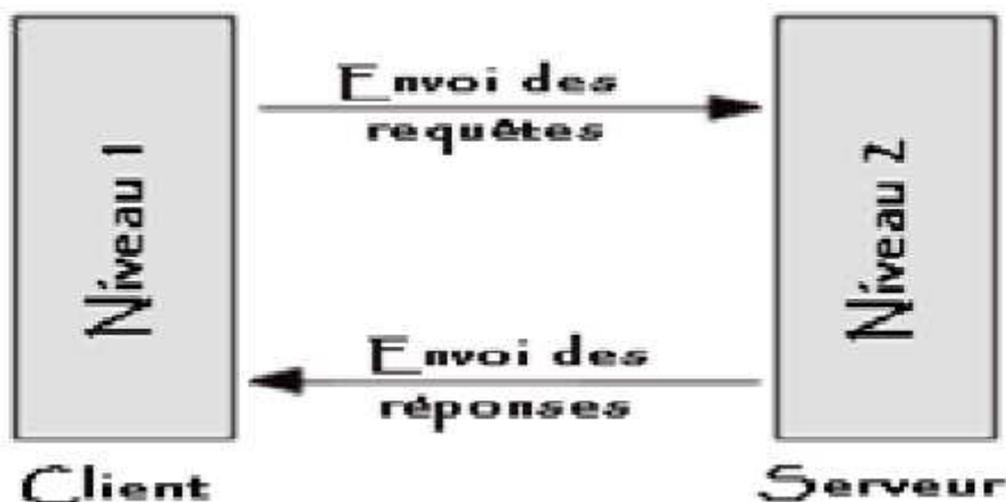


Figure I.6 : Architecture a deux niveaux.

I.7.4.2-Architecture à 3 niveaux (architecture 3-tiers) :

Quand la gestion de la base de données d'application le serveur de base de données permettant d'alléger la charge du serveur d'application.

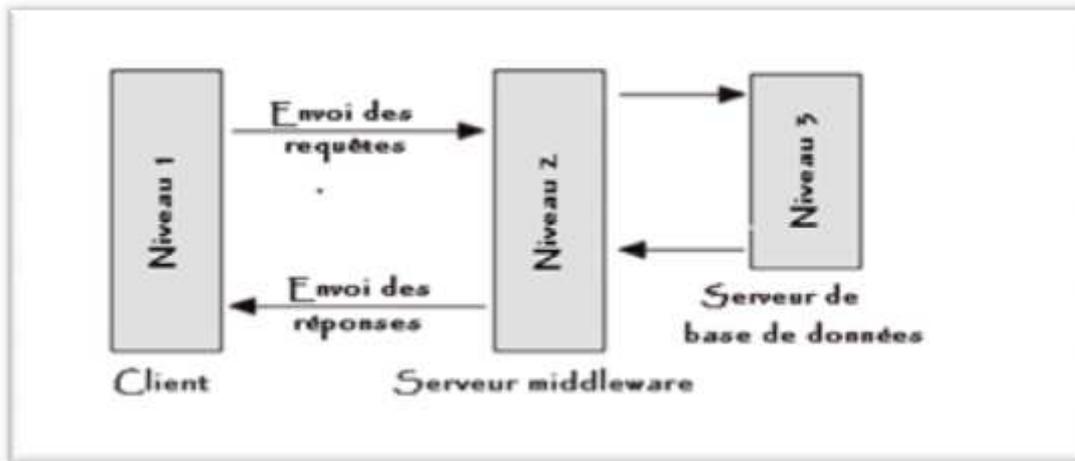


Figure I.7 : Architecture à trois niveaux.

I.7.4.3-Architecture multi niveaux :

Dans les architectures à trois niveaux, chacun des deux serveurs exécute une tâche. Par conséquent, pour fournir un service un serveur peut faire appel à un ou plusieurs serveurs. Donc, l'architecture à trois niveaux est pratiquement un à n niveaux. Comme elle montre la figure suivante :

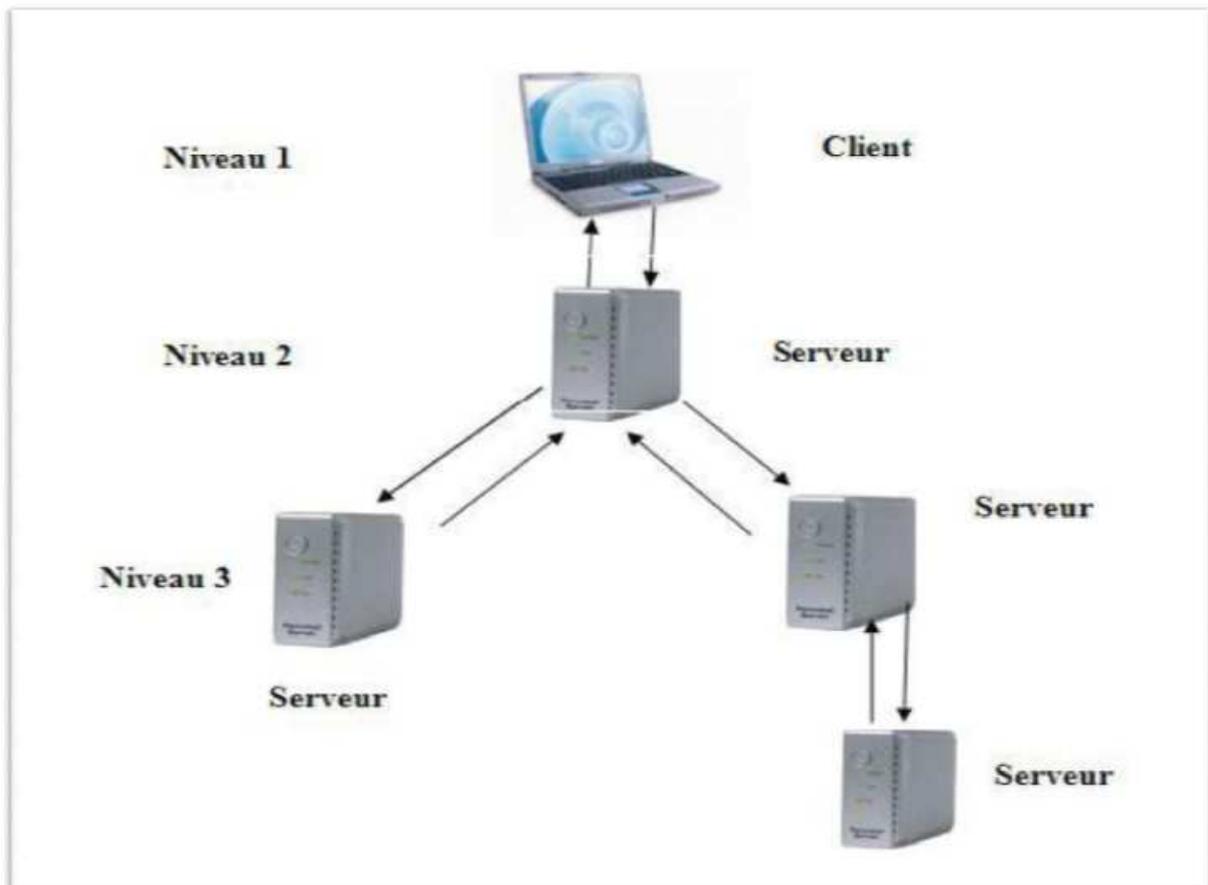


Figure I.8 : Architecture à multi niveaux

I.7.5-Caractéristiques de l'architecture :

- **Le service** : C'est le travail fourni par le serveur suite à la requête du client. Le client est donc consommateur et le serveur fournisseur de services.
- **Les ressources partagées** simultanément et réguler leur accès aux ressources.
- **Échange de message** : Client et serveur sont des systèmes à liaison épisodique qui interagissent au moyen de messages. Le message est le mécanisme d'émission des demandes de services et des réponses à celle-ci.
- **Encapsulation des services** : Le serveur est un spécialiste, un message lui indique quel service est requis. C'est à lui de décider comment rendre ce service. Les serveurs peuvent être mis à niveau sans effet sur les clients tant que l'interface des messages reste la même.

I.8) Conclusion :

Dans ce chapitre, nous avons abordé les notions de base de la plateforme Java EE, certaines d'entre eux seront utilisées pour le développement de notre application et nous avons parlé de l'architecture Client/serveur sa mise en œuvre et son fonctionnement. Dans le chapitre qui suit nous allons présenter l'organisme d'accueil.

Chapitre

III

*Présentation de
l'organisme d'accueil*

Introduction :

La présentation de l'organisme d'accueil est une étape très importante qui nous permet de prendre connaissance de cet organisme qui est l'ENTP (ENTREPRISE NATIONALE DES TRAVAUX AUX PUITES), ainsi que notre champ d'étude qui est le service des ressources humaines, dans ce chapitre on va essayer d'étudier l'existant et de déduire les problématiques et de suggérer des solutions.

II- Présentation de l'ENTP : [7]

II-1-Historique :

Le forage algérien est un produit de SONATRACH depuis l'année 1968. La structure opérationnelle s'appelait Direction des Services Pétroliers (DSP). Keskassa est le premier puits foré et disposait d'un parc de quatre appareils de forage.

En juillet 1972 : DSP prend le nom de Direction des Travaux Pétroliers (DTP).

1er août 1981 : ENTP héritière de la DTP pour les activités de forage et de Work - Over. Créée par décret n° 81-171, ENTP est devenue opérationnelle le premier janvier 1983.

21 Juin 1989 : Transformation du statut juridique de ENTP en société par actions.

Juin 1993 : ENTP est devenue membre de l'IADC (International Association of Drilling Contractors).

30 Mars 1998 : Cession de 51% du capital social ENTP par le holding RGT, en faveur de Sonatrach.

01 janvier 2005 : Transfert des actions détenus par la Société de gestion des participations TRAVEN dissoute, vers la Société de Gestion des participations dénommée "INDJAB"

28 décembre 2005 : Cession à titre gratuit des actions détenues par la SGP INDJAB (49%) en faveur du Holding Sonatrach "SPP Spa". ENTP devient 100% Sonatrach.

25 mars 2006 : Holding Sonatrach Services Para Pétroliers "SPP Spa" devient l'unique Actionnaire de l'ENTP.

II-2-Réglementation :

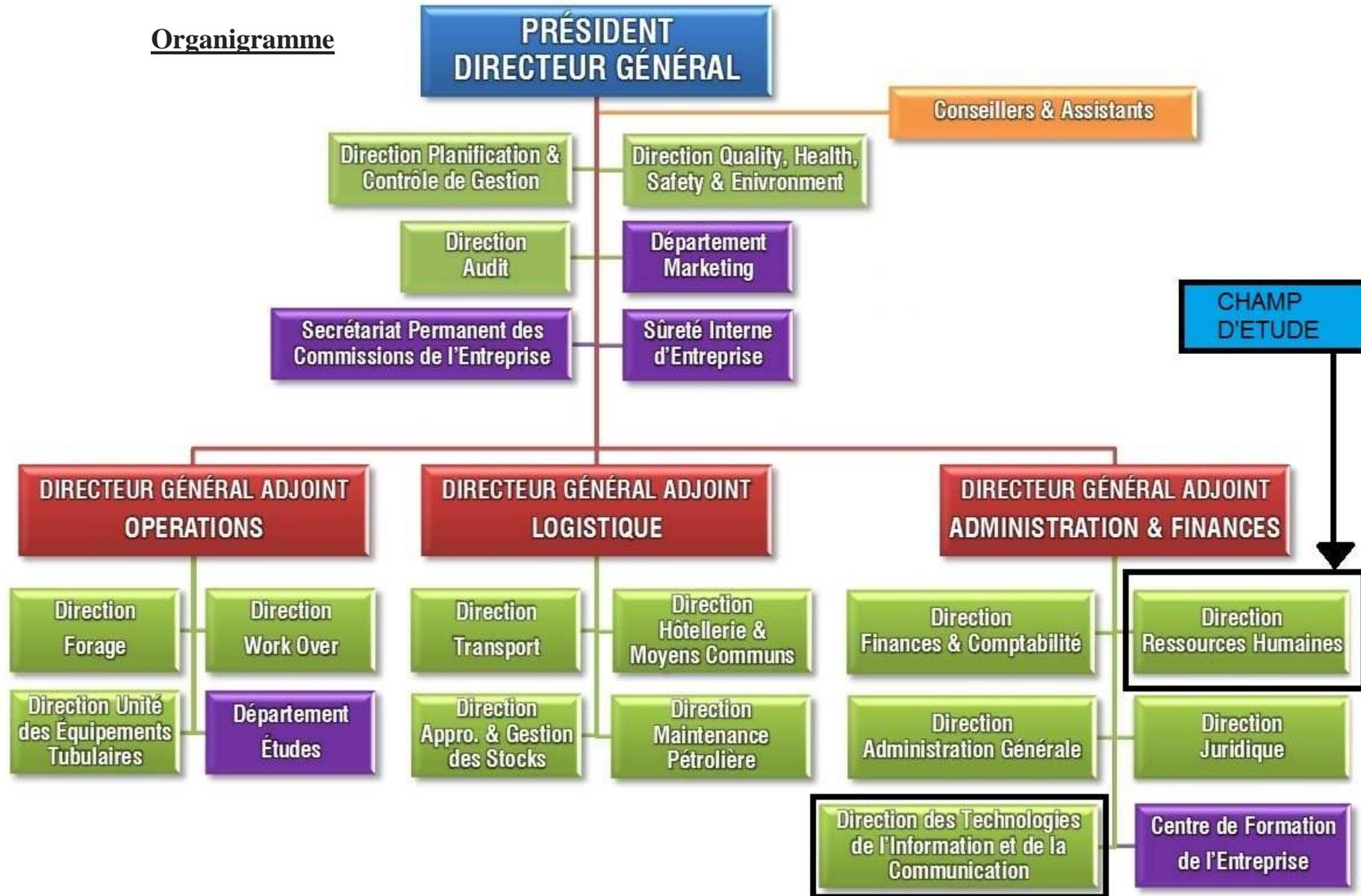
L'ENTP est régie par le cadre juridique suivant :

- le code de commerce modifié et complété par le décret législatif n°08/93 du 25 Avril 1993 et l'ordonnance n° 27 / 96 du 09 Décembre 1996
- le décret législatif 08 / 94 du 26 juin 1994, portant loi de finance complémentaire pour 1994
- l'ordonnance n° 01 / 04 du 20 Août 2001 relative à l'organisation, la gestion et la privatisation des entreprises publiques économiques,
- les dispositions des statuts de l'ENTP et l'ensemble des textes à caractère législatif et réglementaire ultérieurs.

II-3- Nature des activités de l'entreprise :

- Forage des puits d'hydrocarbures.
- Entretien des puits d'hydrocarbures (Work-Over).
- Forage des puits d'eau de grande profondeur.
- Transport (DTM des appareils et camps de forage et rénovation des véhicules)
- Maintenance pétrolière.
- Hôtellerie.

Organigramme



A- L'activité forage et Work-Over :

Les activités de forage et de Work-Over (entretien des puits producteurs d'huile de gaz et d'eau) génèrent plus de 90% des revenus de l'entreprise. Les divisions de Forage et de Work-Over ont pour mission :

- Le forage de puits de pétrole et de gaz, profonds et de moyenne profondeur, la reprise et l'entretien des puits, aussi bien ceux de l'exploration que du développement,
- Le forage des puits d'eau de grande profondeur.

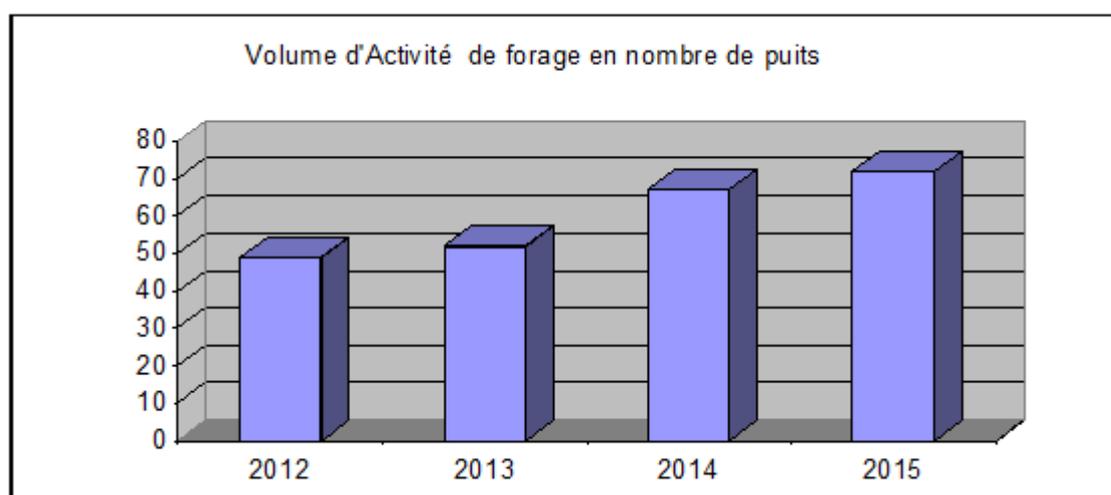
L'ENTP gère au total 38 appareils dotés d'équipements modernes tels que :

- SCR (Silicon Controlled Rectifier)
- Top Driver
- Wireless Network Communication

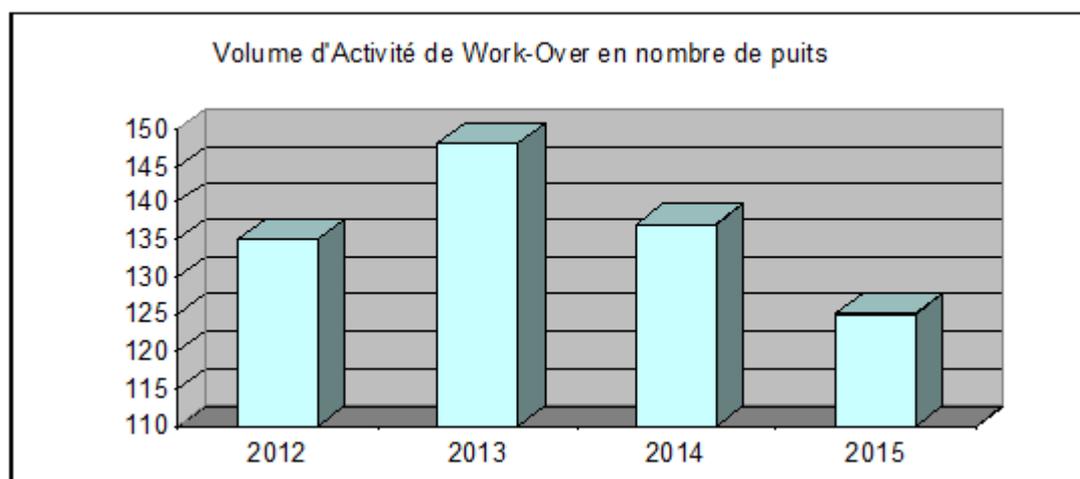
Le volume de production en terme de nombre de puits, est donné dans le tableau ci-dessous

Année	Nombre de puits		total
	Forage Work-Over	Hydraulique	
2012	49	135 -	184
2013	52	148 1	201
2014	67	137 –	204
2015	72	125 1	198

Tableaux II.1 : Volume de production en termes de nombre de puits.



Figures II.1 : Volume d'activité de forage en nombre de puits.

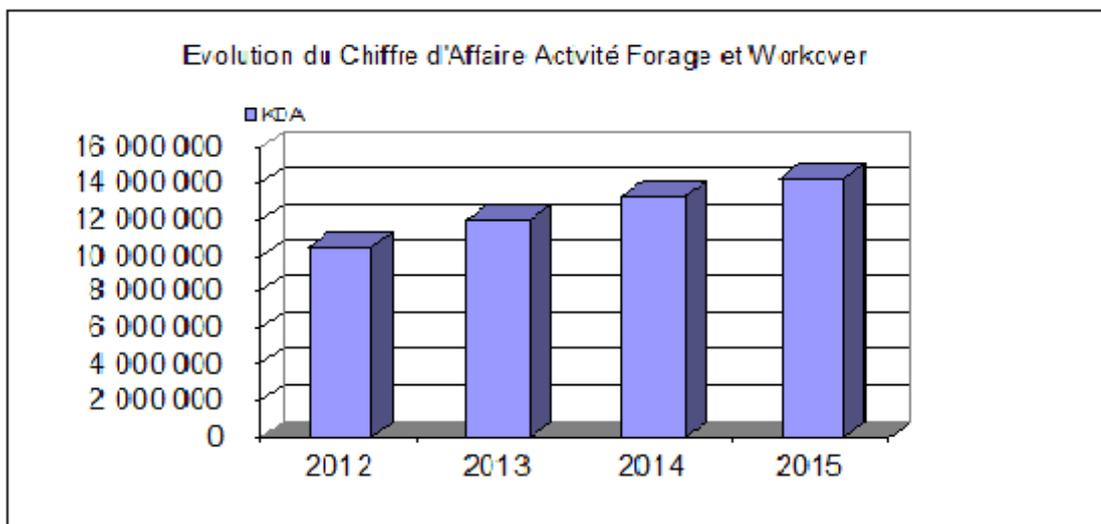


Figures II.2 : Volume d'Activité de Work-Over en nombre de puit.

L'évolution des principaux paramètres d'exploitation est donnée dans le tableau suivant :

	Activité des appareils			
	2012	2013	2014	2015
Potentiel en appareils	36	36	36	36
Capacité réelle mobilisée (Appareils équivalent utilisés)	28,51	29,33	29,70	31,51
Nombre de jours de forage en cadence continue (jours)	10.407	10.706	10.839	11.531
Taux d'utilisation du temps disponible	82,00%	82,22%	82,66%	83,90%
Taux de fonctionnement	76,62%	77,85%	79,79%	80,39%
Longueur total de forage (mètres)	153.966	165.614	29.097	248.887
Rendement en forage de développement (mètres/mois/appareil)	1.296	1.439	1.72	71.890
Rendement en forage d'exploration (mètres/mois/appareil)	945	1.333	1.36	91.242
Chiffre d'Affaires (KDA)	10.516.480	11.959.664	13.250.492	14.254.185

Tableaux II.2 :L'évolution des principaux paramètres d'exploitation.



Figures II.3 :Evolution du chiffre d'affaire Activité Forage et Workover .

B- L'activité maintenance pétrolière:

Cette activité est assurée par la division maintenance pétrolière qui se charge de :

- la rénovation des équipements et des camps de forage,
- la rénovation des appareils électro-froid et électro-ménager
- la fabrication et le reconditionnement des pièces de rechange,
- les interventions sur chantier :
 - prévention et sécurité aux puits
 - mesure et régulation
 - travaux de soudure sur conduite haute pression, circuit boue ou eau et tronçons du mât des appareils de forage.

La Division dispose de 14 ateliers pourvus d'équipements modernes, située à Hassi-Messaoud, occupant une surface couverte de 9.750 m² et une surface noncouverte de 49.800m², ainsi que quatre aires de montage des appareils de forage rénovés.

L'évolution des paramètres d'exploitation en termes de nombre d'heures main d'œuvre et nombre d'heures machine pour les exercices 2014 et 2015 est donnée dans le tableau ci-dessous :

	2014		2015	
	Heures main d'œuvre	Heures Machine	Heures main d'œuvre	Heures Machine
Maintenance des Equipements de Forage	38 109	8 448	33 895	5 260
Maintenance Moteurs Industriels et Electricité	45 524	474	36 925	425
Maintenance des Equipements spéciaux	40 979	4 353	39 708	5 020
Fabrication et chaudronnerie Soudure	135 469	16 805	113 406	53 169
Rénovation et maintenance camps forage	106 249	6 514	104 213	9 041
Total	366 330	36 594	328 147	72 915

Tableaux II.3 :L'évolution des paramètres d'exploitation pour les exercices 2014 et 2015.

C- L'activité transport :

La division transport est assignée :

- Au déménagement des appareils et camps de forage.
- A la livraison des équipements et consommables aux appareils de forage.
- A la maintenance des véhicules et engins de l'entreprise.

La division est dotée des moyens suivants :

Parc Matériel	Type	Capacité (Tonne)	Nombre	Activité
Véhicules spéciaux et Tracteurs	Tracteurs Kenworth 953	40	70	DTM* Livraison et Citernage
	Tracteurs Mercedes	30	27	
Engins de Manutentions	Grues	20 à 30	18	DTM*
		35 à 40	25	
	Chariots élévateurs	14 54 < 14	44	Chantiers de forage

Tableaux II.4 :La division transport.

(*DTM : Déménagement, Transport et Montage)

Le temps d'immobilisation du parc transport pour l'année 2015 enregistre une augmentation de 16%, passant de 14.785 jours en 2014 à 17.090 jours. Le taux d'immobilisation passe de 28% en 2014 à 31% en raison de la maintenance préventive. Le parc transport a exploité 29.018 jours en 2015, soit une augmentation de 15% par rapport à 2014. Le taux d'utilisation passe de 48% du temps disponible en 2014 à 52% en 2015. En termes de nombre d'opérations DTM l'activité transport a réalisé une production physique de

206 DTM (74 DTM pour le forage et 132 pour le Work-Over). Comparée à l'année 2014, le nombre de DTM a baissé de 10 unités.

Le chiffre d'affaires réalisé par l'activité transport durant les exercices 2014 et 2015 est donné dans le tableau suivant :

	En millier de Dinars		Evolution (%)
	2014	2015	
Chiffre d'affaires externe	111 749	153 177	37,07
Chiffre d'affaires interne	682 002	695 524	1,98
Total	793 751	848 701	6,92

Tableaux II.5 :Le chiffre d'affaires réalisé par l'activité transport durant les exercices 2014 et 2015.

D- L'activité hôtellerie et le Catering :

L'activité hôtellerie assure les prestations d'hébergement et de restauration :

- Au personnel opérant sur appareils de forage
- Au personnel opérant sur bases fixes
- Au personnel des compagnies pétrolières

L'activité en moyenne est de 1.500.000 rations par an.

La Division hôtellerie dispose de 3 bases de vie et d'une trentaine de camps de forage dont les capacités d'accueil sont de l'ordre de 4.140 lits. Ces capacités de stockage sont de 860 m en frigorifique et 590 m en sec, ainsi qu'une capacité de transport composée d'une flotte de 32 véhicules de transport frigorifique d'une capacité de 1043 m³.

Le chiffre d'affaires réalisé par l'activité hôtellerie et le catering durant les exercices 2014 et 2015, selon la nature des prestations (internes ou externes) et selon le lieu des prestations (bases de vie ou camps de forage) est donné dans le tableau suivant :

	En milliers de dinars	
	2014	2015
Prestations internes	2 592 085	2 802 549
Prestations externes	3 736	2 170
Total	2 595 821	2 804 719
Bases	585 077	579 681
Camps de forage	2 010 744	2 225 038
Total	2 595 821	2 804 719

Tableaux II.6 :Le chiffre d'affaires réalisé par l'activité hôtellerie et le catering durant les exercices 2014 et 2015.

E- Autres activités :

L'ENTP assure aussi d'autres activités par le biais de sa Direction Engineering et son Unité des Equipements Tubulaires (UET).

La direction Engineering a pour mission :

- Le revamping des appareils de forage (amélioration sur le plan de la forme, remodelage).
- L'assistance technique des divisions opérationnelles en matière de :
 - Spécifications techniques des équipements de forage.
 - Procédures de revamping.
 - Etudes techniques pour l'acquisition d'appareils de forage et des équipements.
 - Mise à disposition de documentation technique et normes des projets de conceptions.

Sur le plan de revamping, depuis la création de l'entreprise et par ses moyens propres, l'ENTP a remis à neuf et rénové 24 appareils de forage pour un coût de 185 549 151 USD.

L'unité des équipements tubulaires (UET) est certifiée ISO 9001/94 depuis juin 2001 puis certifiée ISO 9001/2000.

Ses prestations sont les suivantes :

- L'inspection des équipements tubulaires ENTP.
- La réparation des éléments de la garniture de forage.
- Le rechargement des stabilisateurs.
- Le contrôle non destructif des équipements en Joint-Opération avec TUBOSCOPE-VECT pour les associés de Sonatrach.

L'UET est dotée d'une unité fixe d'inspection électromagnétique (scanner) et de deux (02) unités mobiles acquises en 2003.

L'UET dispose d'ateliers spécialisés pour :

- La fabrication des produits courts tels que Short Drill Collar, Drill StemSubs, Basket Sub, Lift Sub basé sur les spécifications de la norme API(Application Programming Interface ou Interface de ProgrammationApplicative, Interface de programmation offerte aux logiciels en interne pour s'interfacer avec un dispositif).

- La soudure de rechargement des lames de stabilisateurs en carbures de tungstène.

L'UET procède à la réparation d'une gamme de produits : Drill Pipe, Drill Collar, Kelly Heavy Weight, Drill Stem Sub, Basket Sub, Stabiliser, Casing et Tubing.

L'UET dispose d'une capacité de stockage de 40.000 m²

II-4- Débouchés et place sur le marché :

Les principaux clients de l'ENTP sont :

1-SONATRACH.

2-Les associés de SONATRACH :

a. REPSOL YPF

b. BHP PETROLEUM

c. GULF OIL GULF KEYSTONE

d. CEPSA

e. ARCO

f.OXY OCCIDENTAL

g.Eni Group AGIP

h. HALIBURTON

i.SPERRY-SUN Drilling Services

3-Les Wilayas du sud pour les travaux de forage hydrauliques.

SONATRACH est le principal client de l'ENTP avec plus de 90% du chiffre d'affaires.

L'ENTP est le premier contractant de forage en Algérie avec plus de 50% du marché national (Source : SONATRACH).

Ses concurrents sur le marché sont principalement ENAFOR (28 appareils de forage) qui est une autre filiale de SONATRACH, et d'autres entreprises telles que NABORS (5 appareils de forage), SEDCO (3 appareils de forage), SAIPEM(2 appareils de forage) et PRIDE (2 appareils de forage).

II-5- Principales Installations :

ENTP détient en toute propriété les installations présentées dans les tableaux ci-dessous :

a- Les terrains :

Localisation	Surface (m ²)	Observation (propriété, Location)
Alger 29.543,50		Propriété
Hassi Messaoud	1.231.992,79	Propriété
Touggourt 112.646,00		Propriété
In Aménas	149.700,00	Propriété
Hassi R'Mel	54.676,00	Propriété
Mostaganem	3.871,00	Propriété
Total 1.582.429,29		

Tableaux II.7 : Les sites de l'ENTP.

b- Les bâtiments : (usines, entrepôts, succursales, aires de stockage, parking, base de vie)

Localisation	Surface (m ²)	Observation (propriété, Location)
Alger	4.359,54	Propriété
Hassi Messaoud	133.195,40	Propriété
Touggourt 3.140,88		Propriété
In Aménas	441,00	Propriété
Total	141.136,82	

Tableaux II.8 : Structure de l'ENTP.

II-6- Personnels et effectifs :

A fin décembre 2015, le personnel de l'entreprise totalise un effectif de 5.704 agents, il est en augmentation de 146 agents par rapport à 2014.

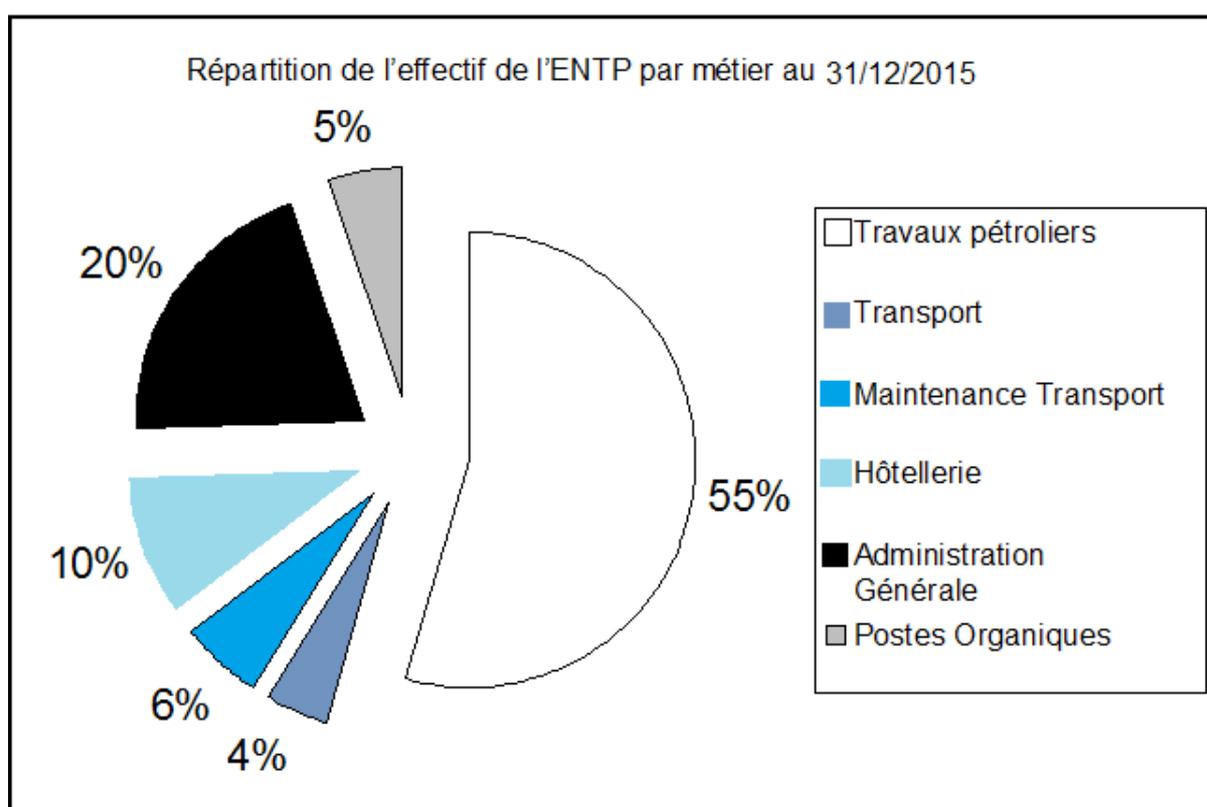
Structures de l'effectif	2014	2015	Ecart
Personnel Permanent	4.023	3.797	-226
- Cadre	1.356	1.317	-39
- Maîtrise	1.491	1.431	-60
- Exécution	1.176	1.049	-127
Personnel Temporaire	1.532	1.904	+372
Personnel Etranger	2	2	0
Pré Emploi	1	1	0
TOTAL ENTREPRISE	5.558	5.704	+146

Tableaux II.9: répartition de l'effectif de l'ENTP.

La répartition par métier au 31 décembre 2015, se présente comme suit :

Désignation	Nombre	Pourcentage
Postes Organiques	296	5,19%
Travaux Pétroliers	3.105	54,44%
Transport	256	4,49%
Maintenance Transport	336	5,89%
Hôtellerie	548	9,61%
Administration Générale	1.163	20,39%
TOTAL	5.704	100,00%

Tableaux II.10 :La répartition par métier au 31 décembre 2015.

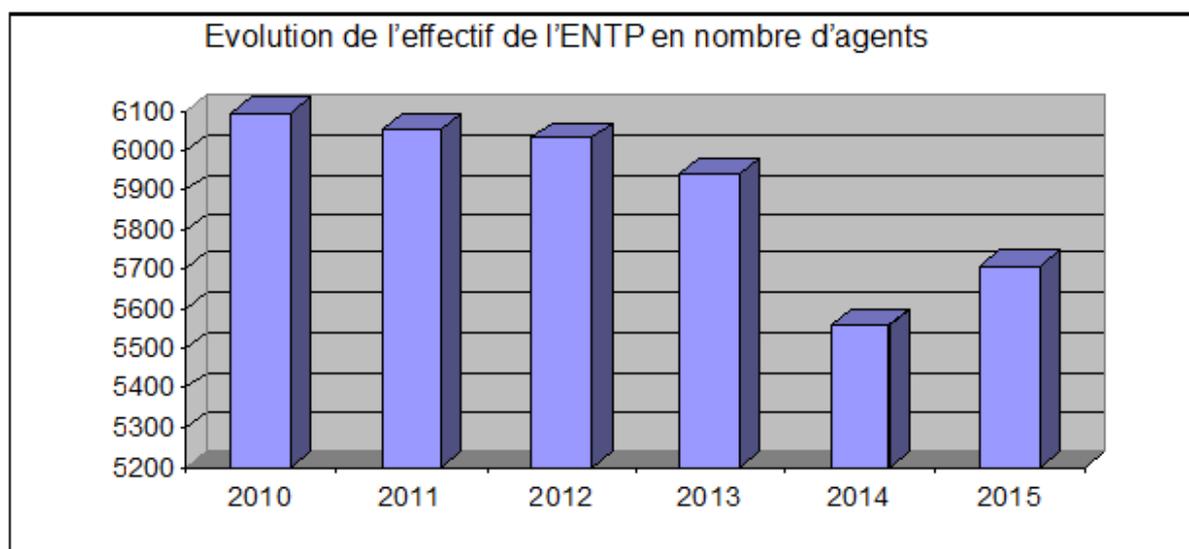


Figures II.4 : Répartition de l'effectif de l'ENTP par métier au 31/12/2015.

L'évolution des effectifs entre 2010 est 2015 est donnée comme suit :

Année	2010	2011	2012	2013	2014	2015
Effectif	6.095	6.055	6.031	5.943	5.558	5.704

Tableaux II.11 :L'évolution des effectifs entre 2010 est 2015.



Figures II.5 : Evolution de l'effectif de l'ENTP en nombre d'agent.

L'ENTP enregistre une baisse constante des effectifs, essentiellement dues aux :

- Départs volontaires à la retraite.
- Départs volontaires avec mesures incitatives.

Deux tiers des effectifs permanents font partie de l'encadrement de l'ENTP.

Le forage et Work-Over occupent plus de 54 % des effectifs

La gestion des ressources humaines enregistre un regain de dynamisme notamment par :

- Le redéploiement d'une partie des effectifs,
- La sélectivité objective et qualitative des recrutements des personnels,
- Le rajeunissement de l'effectif par l'injection de 80 ingénieurs par an depuis 2013.

Dans le domaine de la formation, les besoins sont identifiés et analysés pour concevoir une formation adaptée. Parmi les grands axes de formation élaborés pour l'exercice 2004, figurent des programmes spécifiques pour la QHSE, la GMAO et le Plan de Développement Informatique.

Au cours de l'année 2004, ce sont 3.300 agents qui ont bénéficié d'une formation (contre 2.104 en 2003) et les dépenses de la formation ont atteint 327.863 KDA, soit 6 % de la masse salariale (contre 104.366 KDA, soit 2,10% de la masse salariale).

II-7-Les Investissements :

Durant l'exercice 2015, l'ENTP a réalisé des investissements que nous reprenons ci-après :

Désignation	Montant en milliers de Dinars	
	2014	2015
Machines et Equipements	3 555 147	2 889 718
Dont Forage et Revamping	3 555 147	2 727 647
Véhicules et Engins	-	200 658
Infrastructures	77 631	180 843
Formation	104 367	327 863
Divers	98 866	227 372
TOTAL	3 836 011	3 826 454

Tableaux II.12: Les investissements de l'ENTP, Durant l'exercice 2015.

II-8- Les Approvisionnements :

L'ENTP est soumise pour ses approvisionnements à l'application intégrale de la circulaire 418/SH, relative à la mise en œuvre de la procédure d'appel d'offres BAOSEM (Bulletin d'Appel d'Offres du Secteur des Energies et des Mines) depuis 2013.

II-9- Renseignement sur les filiales et participations :

ENTP détient en titres de participation 20% (60 actions à valeur nominale de 100.000 DA, du capital de Centre de Développement & d'Application des Techniques Pétrolières & Gazières, NAFTOGAZ/spa. L'Assemblée Générale Ordinaire réunie en séance extraordinaire le 2 mars 2016, a décidé la cession des actions détenues par l'ENTP dans le capital de NAFTOGAZ/spa au profit du holding SPP, à leur valeur nominale soit (60 actions x 100.000 DA) 6.000.000 DA. Cette cession se fera en contrepartie de la réduction des créances sur associées et sociétés apparentées détenues par le holding SPP sur l'ENTP.

II-10- Litiges : Néant

A la connaissance de la société, il n'existe pas de litige, arbitrage ou fait exceptionnel susceptible d'avoir ou ayant eu dans un passé récent, une incidence significative sur la situation financière, le résultat, l'activité et le patrimoine de la société

II-11- Contrats et engagements importants :

Des Accords Cadres (avec ou sans garantie bancaire et avec ou sans engagement de l'ENTP) ont été conclus avec les organismes financiers étrangers et nationaux en 2014, 2015 dont nous reprenons ci-après les plus importants :

1-Commande de 04 appareil de forage 2000HP (02) et 1200 HP (02) :

- Contrat du 29 juillet 2015

- Montant : 91 312 612 \$US
- Fournisseur : EDRA (Groupement européen basé en Italie)
- Banque de financement : HSBC (France) 2 appareils 2000HP
- Lettre de crédit à vue : 02 appareils 1200HP
- Livraison : 2000 HP (02) en 2016, 1200HP (02) en 2017

Modalités de paiement : acompte de 5% dès la signature, le reste à la réception finale.

2- Commande de 12 packs SCR (Silicone Control Rectifier)

- Contrat du 30 Avril 2003
- Montant : 27 500 000 \$US
- Fournisseur : IER (Canada)
- Banque de financement : EDC (Canada)

II-12- Facteurs de Risques :

ENTP est une entreprise de forage qui a ses propres risques industriels lesquels sont couverts par des polices d'assurances.

Les facteurs de risque comprennent notamment la conjoncture commerciale et économique mondiale, les activités de forage et le nombre de puits de forage, la volatilité des prix du pétrole et du gaz, la diminution de la demande liée aux produits et services de l'ENTP, les risques liés à l'exploitation, dont les incendies, les explosions et les déversements accidentels de pétrole, entre autres, les litiges imprévus pour lesquels les assurances et les ententes avec les clients ne procurent pas une protection complète, aussi bien que les risques associés à la fluctuation des taux de change. Cette liste des facteurs de risques n'est pas exhaustive. D'autres risques et incertitudes, dont l'ENTP n'a pas actuellement connaissance ou non significatifs pourraient également avoir une incidence négative sur son marché.

L'ENTP a obtenu la certification ISO 9001/2000 en avril 2003. Cette certification a été confirmée après un audit en avril 2004. Elle a finalisé en mai 2005, l'opération de certification QHSE (Qualité, Hygiène, Sécurité, Environnement) afin de mieux prévoir et minimiser les risques.

Conclusion :

Dans ce chapitre nous avons présenté notre organisme d'accueil en spécifiant sa structure organisationnelle et des services avec ses fonctions.

Dans le chapitre qui suit, nous allons présenter Analyse & Conception de notre application.

Chapitre

III

Analyse et conception

Introduction :

L'étude de l'organisme d'accueil nous a permis de collecter les informations qui concernent le fonctionnement du système existant, et ainsi de connaître les besoins et les orientations des utilisateurs.

Pour le développement de notre application de gestion de personnels cas: ENTP nous avons opté pour une démarche de conception orientée objet (RUP, Rational Unified Process), en se basant sur la modélisation UML. Pour cela, nous suivons les phases suivantes :

- Initialisation : vision approximative de la finalité du projet, cas d'entreprise, périmètre, estimation globales.
- Elaboration : vision plus élaborée, implémentation itérative de l'architecture noyau, résolution des risques élevés, identification de la plupart des besoins et du périmètre réel, estimation plus réalistes.
- Construction : implémentation itérative des éléments qui présentent des risques et une complexité moindres et préparation du déploiement.
- Transition : déploiement.

III.1 Phase d'Initialisation : une analyse approximative de notre projet permet de desseller les points suivants :

- Notre application touche à la gestion du personnels cas: ENTP qui est un domaine connu;
- Notre application est de type Client/serveur ;
- Notre application est isolée à internet, le risque de sécurité ne présente pas un énorme problème.

Vu les points précédent de notre analyse de surface, nous pouvons dire que ce projet est abordable.

III.2 Phase d'Elaboration :

Notre projet porte sur la mise en place d'une applicationclient/serveur 3-tiers, pour l'automatisation des activités de la gestion du personnels cas: ENTP services : DTI.

Dans le but de maîtriser le suivi des mouvements des documents et dossiers des employés du service de la direction technique et de l'informatiques de l'ENTP, l'application doit offrir un environnement interactif et convivial aux différents acteurs de ce service, et ce via intranet.

III.2.1 Identification des acteurs :

Le stage que nous avons fait nous a permis de procéder à l'identification des principaux acteurs qui seront les futurs utilisateurs de notre application.

- Administrateur (ADMIN): c'est le chef de service, il peut modifier, ajouter et supprimer un employés, contrats, missions, formations, stage,sanctions ainsi que consulter la liste des employés,d'absences ...etc.
- Chef de départements : son rôle est d'ajouter des absences.

III.2.2 Diagramme de contexte :

Le diagramme de contexte est un modèle conceptuel de flux qui offre une vision globale des interactions entre le système et l'environnement extérieur. Il permet aussi de bien délimiter le champ de l'étude. Pour notre cas, le diagramme de contexte est donné par la figure suivante :

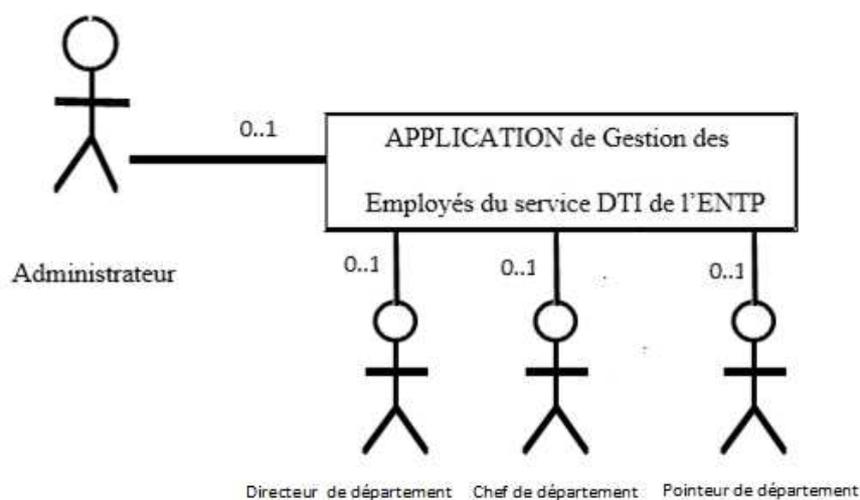


Figure III.1: diagramme de contexte

III.2.3 Représentation des diagrammes de cas d'utilisation :

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs, ils interagissent avec les cas d'utilisation (use

cases), les comportements sont modélisés sous la forme de cas d'utilisation via un diagramme de cas d'utilisation.

- Représentation les cas d'utilisation globale :

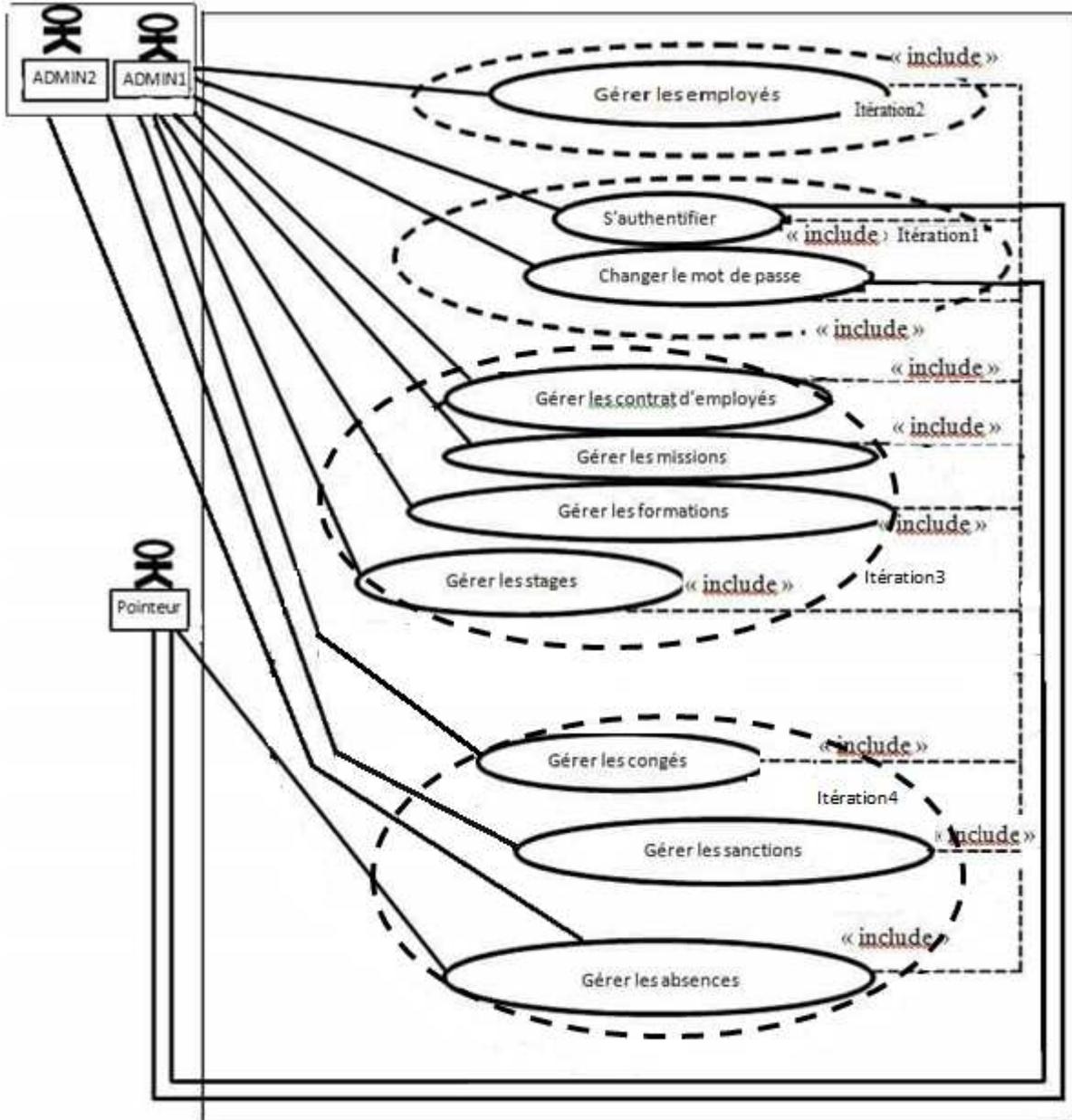


Figure III.2 : Diagramme de cas d'utilisation

III.2.4 Définition des itérations :

Le critère de définition des itérations se base sur les deux points suivants :

1. Ressemblance entre cas d'utilisation.
2. Dépendance entre cas d'utilisation.

III.3 Construction:

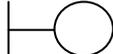
Pour chaque itération réalisée précédemment, nous allons suivre un cycle de vie en cascade pour aboutir à un diagramme de classe en passant par un diagramme de séquence ou de collaboration.

III.3.1 Réalisation de l'itération 1**III.3.1.1 Diagrammes de séquence pour itération 1:**

Les diagrammes de séquence présentent la coopération entre différents objets. Les objets sont définis et leur coopération est représentée par une séquence de messages entre eux.

Après avoir déterminé nos besoins et réaliser les diagrammes des cas d'utilisations nous allons élaborer quelques diagrammes de séquences de l'itération 1. Les classes d'objets utilisées dans la représentation du diagramme de séquence d'analyse peuvent être réparties dans les trois catégories selon les stéréotypes suivant:

- Les objets de type interface : l'interface entre l'acteur et le système, par exemples application web, des pages web complètes.

L'icône utilisée : 

- Les objets de type entité : sont des objets décrits dans un cas d'utilisation, qui peuvent apparaître dans de nombreux cas d'utilisation, et généralement sa durée de vie dépasse celle de toute Interaction où il participe.

L'icône utilisée : 

- Les objets de type contrôle : représentent les processus, c'est-à-dire les activités système, ces objets dirigent les activités des objets entité et d'interface.

L'icône utilisée: 

- Diagramme de séquence du cas d'utilisation : « Authentification

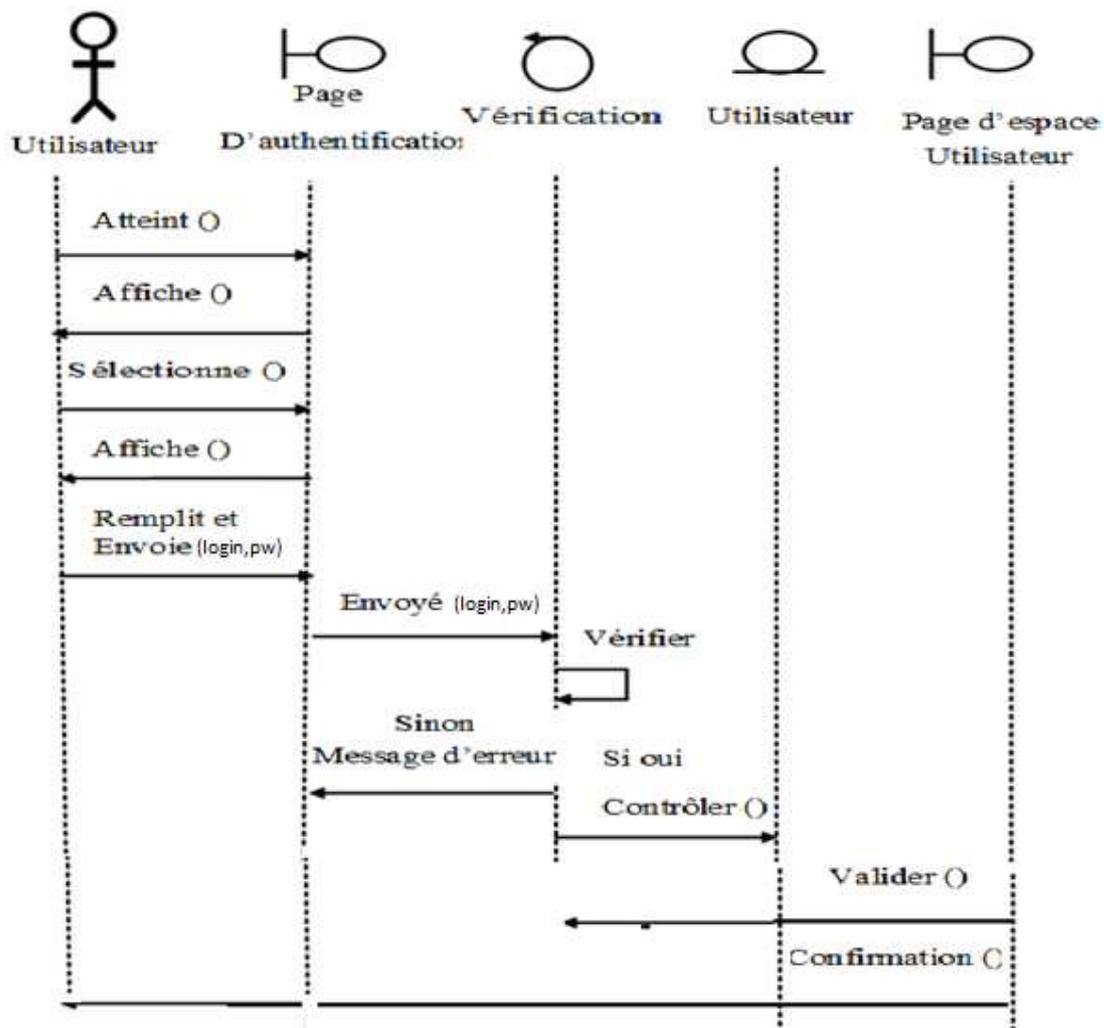


Figure III.3. Diagramme de séquence du cas d'utilisation : « Authentification »

- Description du cas d'utilisation avec des scénarios: Cas utilisation : Authentification.

Acteur : UTILISATEUR

Résumé : Ce cas d'utilisation permet à l'utilisateur (UTILISATEUR) de s'authentifier et d'avoir un espace personnel.

Scénario :

1. Le système retourne le formulaire «authentification ».
2. Le système retourne le formulaire «authentification ».
3. Le système affiche « espace UTILISATEUR »
4. Le système retourne une page de confirmation

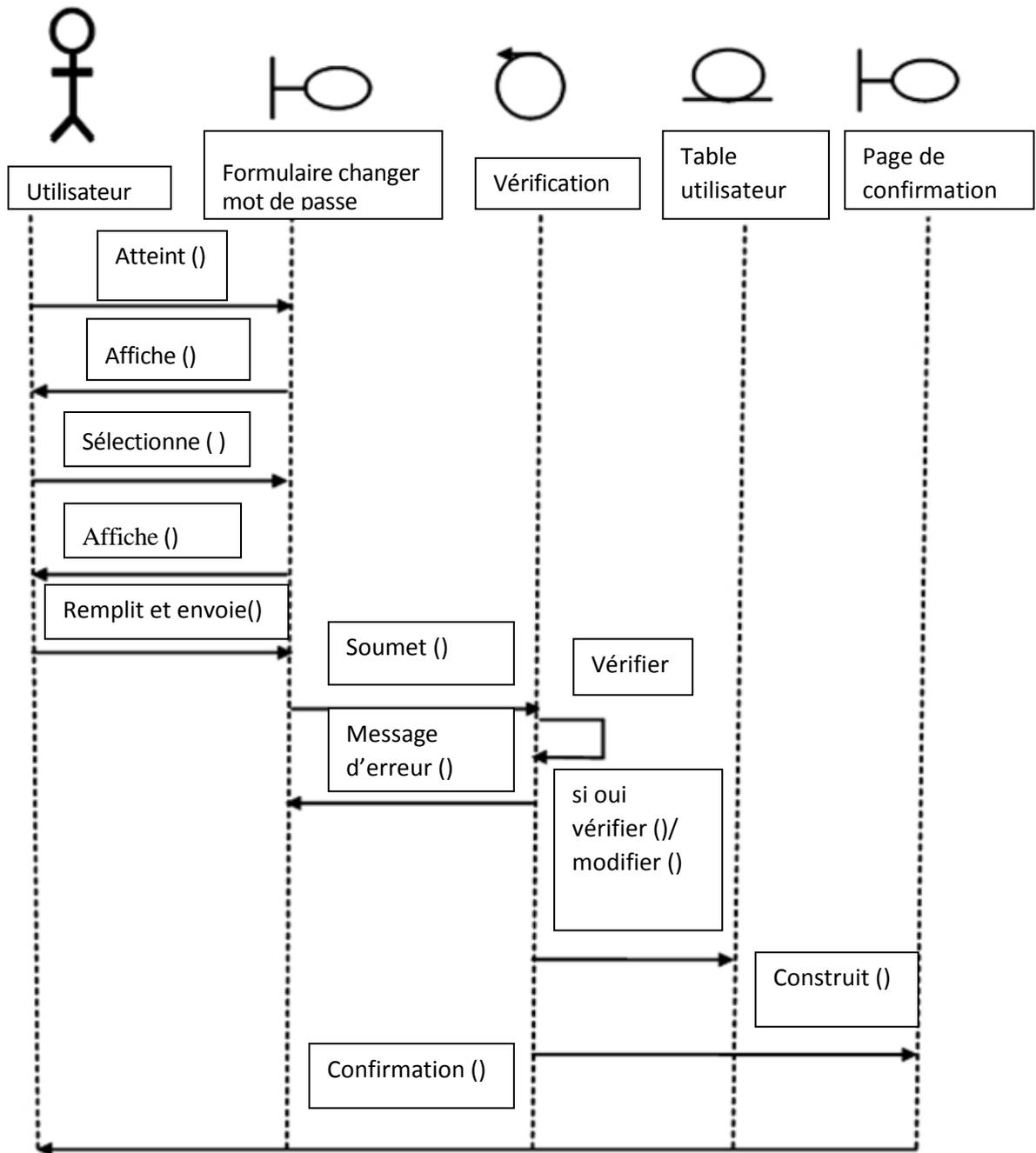


Figure III.4. Diagramme de séquence du cas d'utilisation : «Modification du mot de passe»

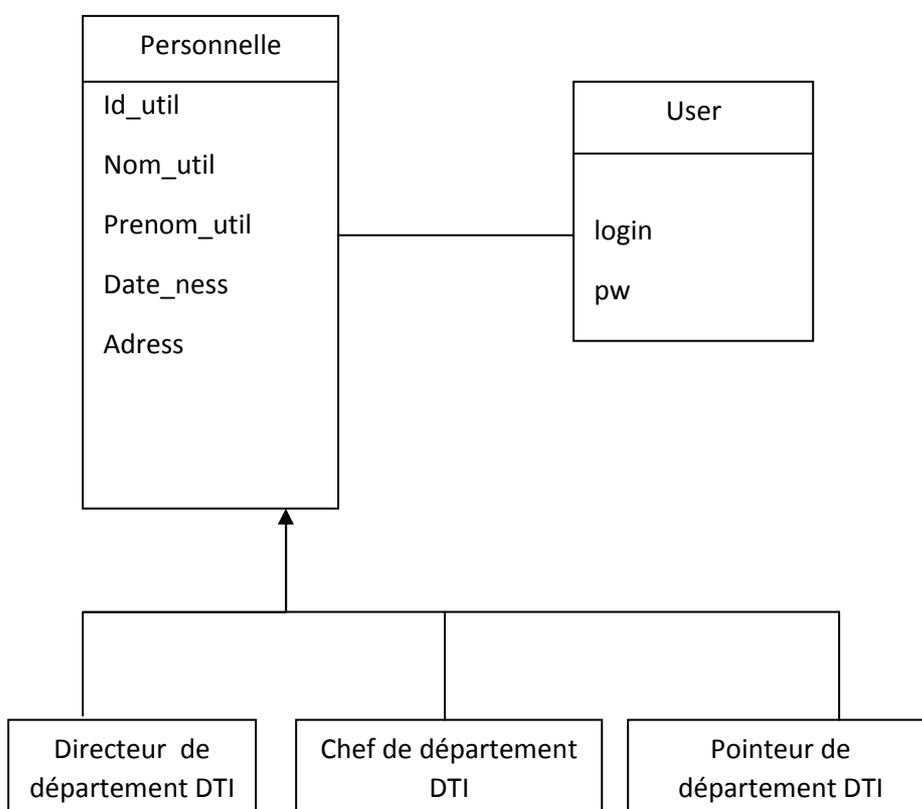
- Description du cas d'utilisation avec des scénarios: Cas utilisation : Authentification

Acteur : UTILISATEUR

Résumé : Ce cas d'utilisation permet à l'utilisateur (UTILISATEUR) modifier son mot de passe

Scénario :

- 1-L'UTILISATEUR atteint son espace.
- 2-Le système affiche « espace UTILISATEUR ».
- 3- L'UTILISATEUR sélectionne la rubrique « changer le mot de passe». 4-Le système retourne le formulaire «changer le mot de passe».
- 5- L'UTILISATEUR saisit et soumet les données.
- 6-Le système retourne une page de confirmation

III.3.1.2 Diagrammes de classe pour l'itération 1**Figure III.5 : Diagramme de classes pour l'itération 1****III.3.2 Réalisation de l'itération 2****III.3.2.1 Diagrammes de séquence pour itération 2:**

- Diagramme de séquence du cas d'utilisation : «Ajouter un employée»

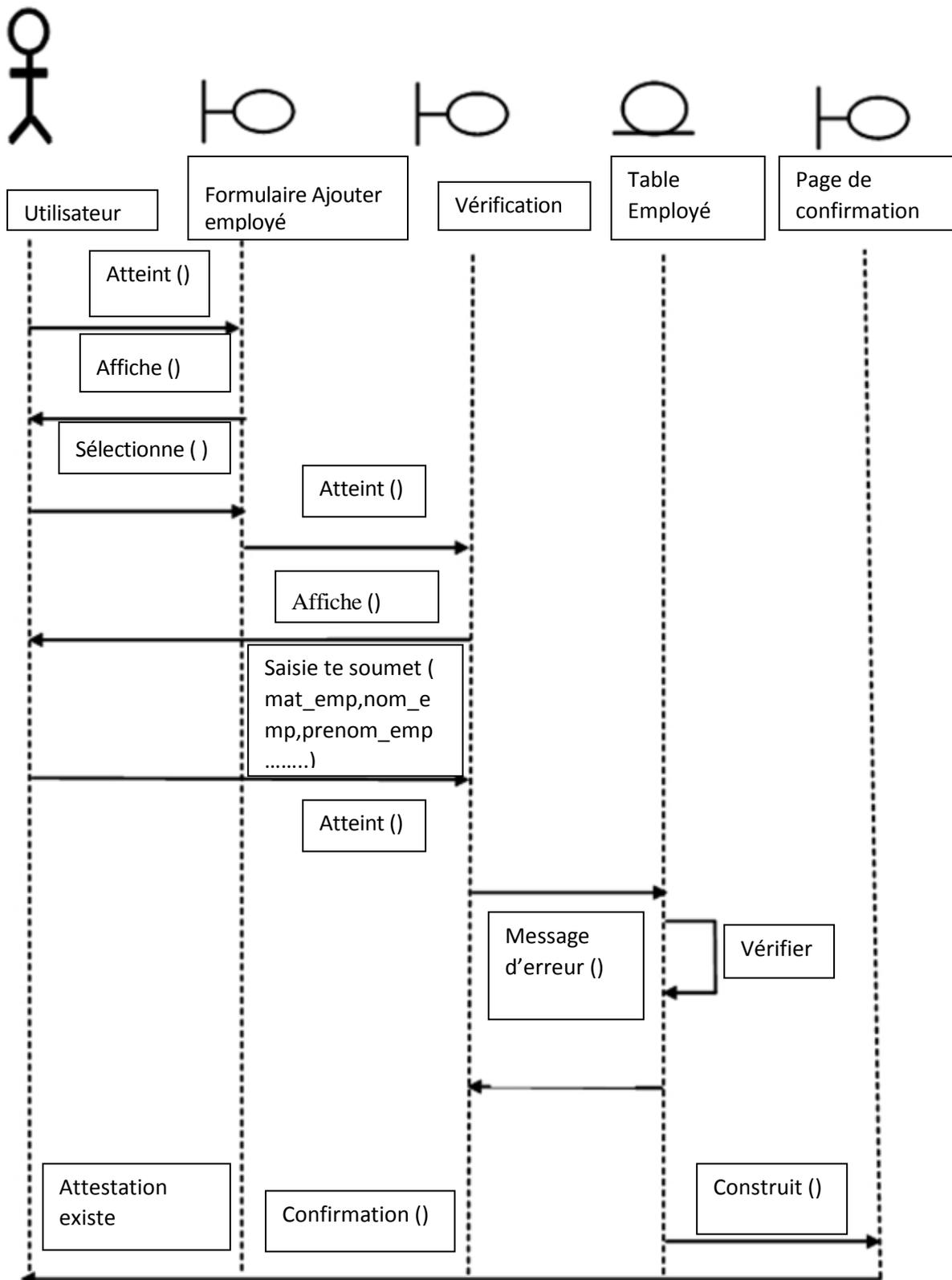


Figure III.6 :Diagramme de séquence du cas d'utilisation : «Ajouter un employée »

- Description du cas d'utilisation avec des scénarios:

Cas utilisation : Ajouter un employé.

Acteur : ADMIN

Résumé: Ce cas d'utilisation permet à l'administrateur (ADMIN) d'ajouter un employeur.

Scénario :

- 1- L' ADMIN atteint son espace.
- 2- Le système affiche « ADMIN ».
- 3- L'utilisateur sélectionne la rubrique « Employée»
- 3- L'utilisateur sélectionne la rubrique « Ajouter employée»
- 4- Le système retourne le formulaire « Ajouter employé».
- 5- L' ADMIN remplit le formulaire et clique sur le bouton « Enregistrer employée ».
- 6- Le système ajoute un nouvel employeur après avoir vérifié la validité Des données saisies.

III.3.2.2 Diagrammes de classe pour l'itération2

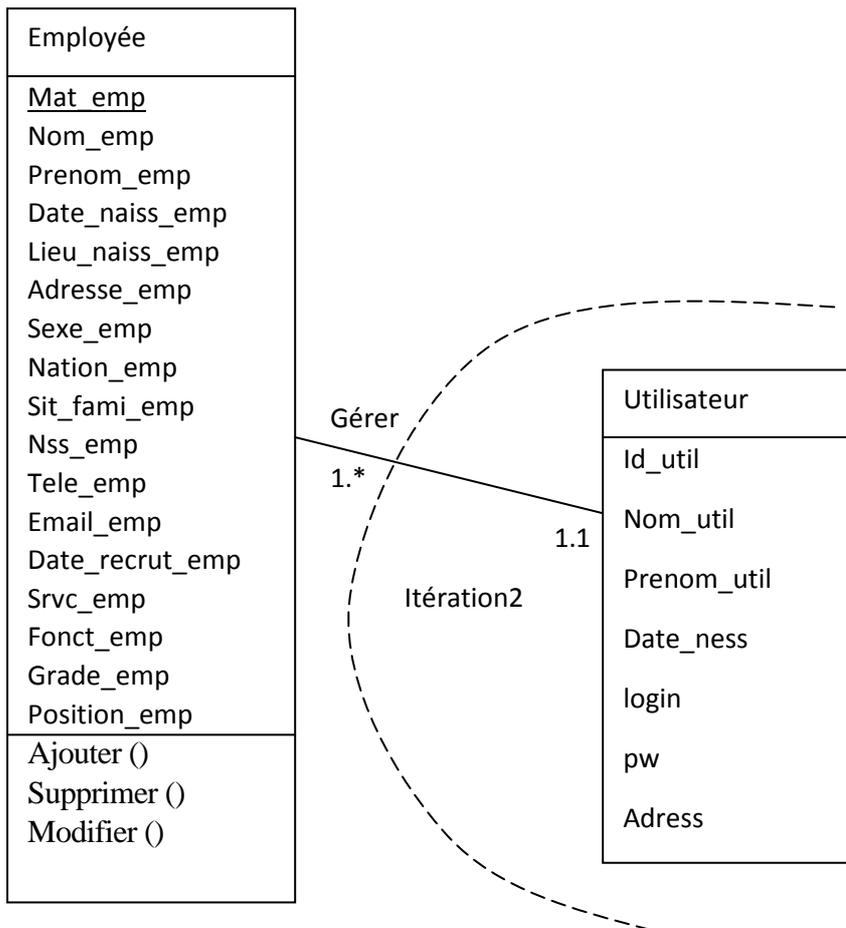


Figure III.7 : Diagramme de classes pour l'itération2

III.3.2 Réalisation de l'itération 3

III.3.2.1 Diagrammes de séquence pour itération 3:

- Diagramme de séquence du cas d'utilisation : «Ajouter une mission»

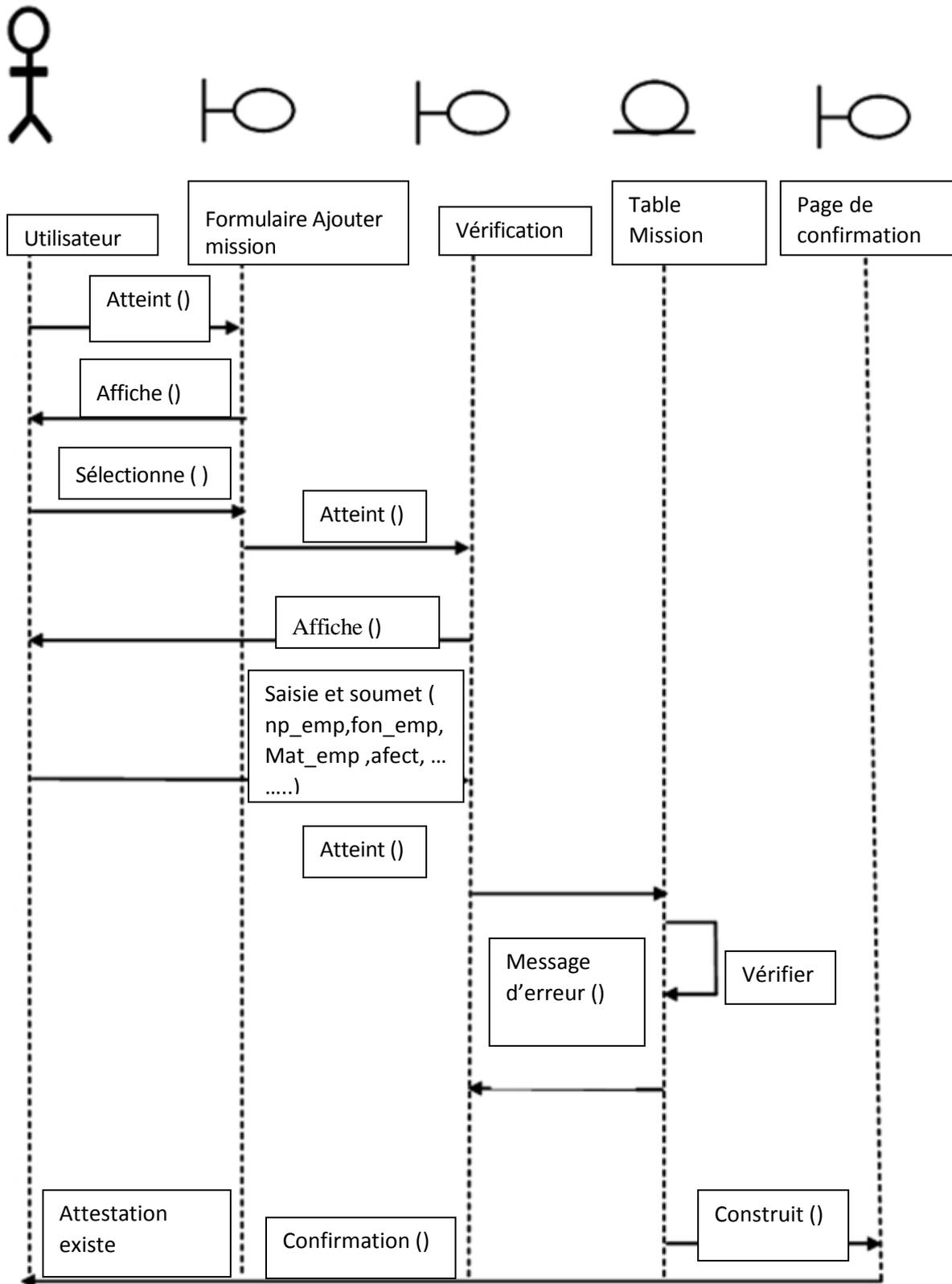


Figure III.8 : Diagramme de séquence du cas d'utilisation : «Ajouter une mission »

- Description du cas d'utilisation avec des scénarios:

Cas utilisation : Ajouter une mission.

Acteur : ADMIN

Résumé: Ce cas d'utilisation permet à l'administrateur (ADMIN) d'ajouter une mission.

Scénario :

- 1- L' ADMIN atteint son espace.
- 2- Le système affiche « ADMIN ».
- 3- L'utilisateur sélectionne la rubrique « mission»
- 3- L'utilisateur sélectionne la rubrique « Ajouter mission»
- 4- Le système retourne le formulaire « Ajouter une mission».
- 5- L' ADMIN remplit le formulaire et clique sur le bouton « Enregistrer mission ».
- 6- Le système ajoute une nouvelle mission après avoir vérifié la validité Des données saisies.

- Diagramme de séquence du cas d'utilisation : «**Modifie une formation**»

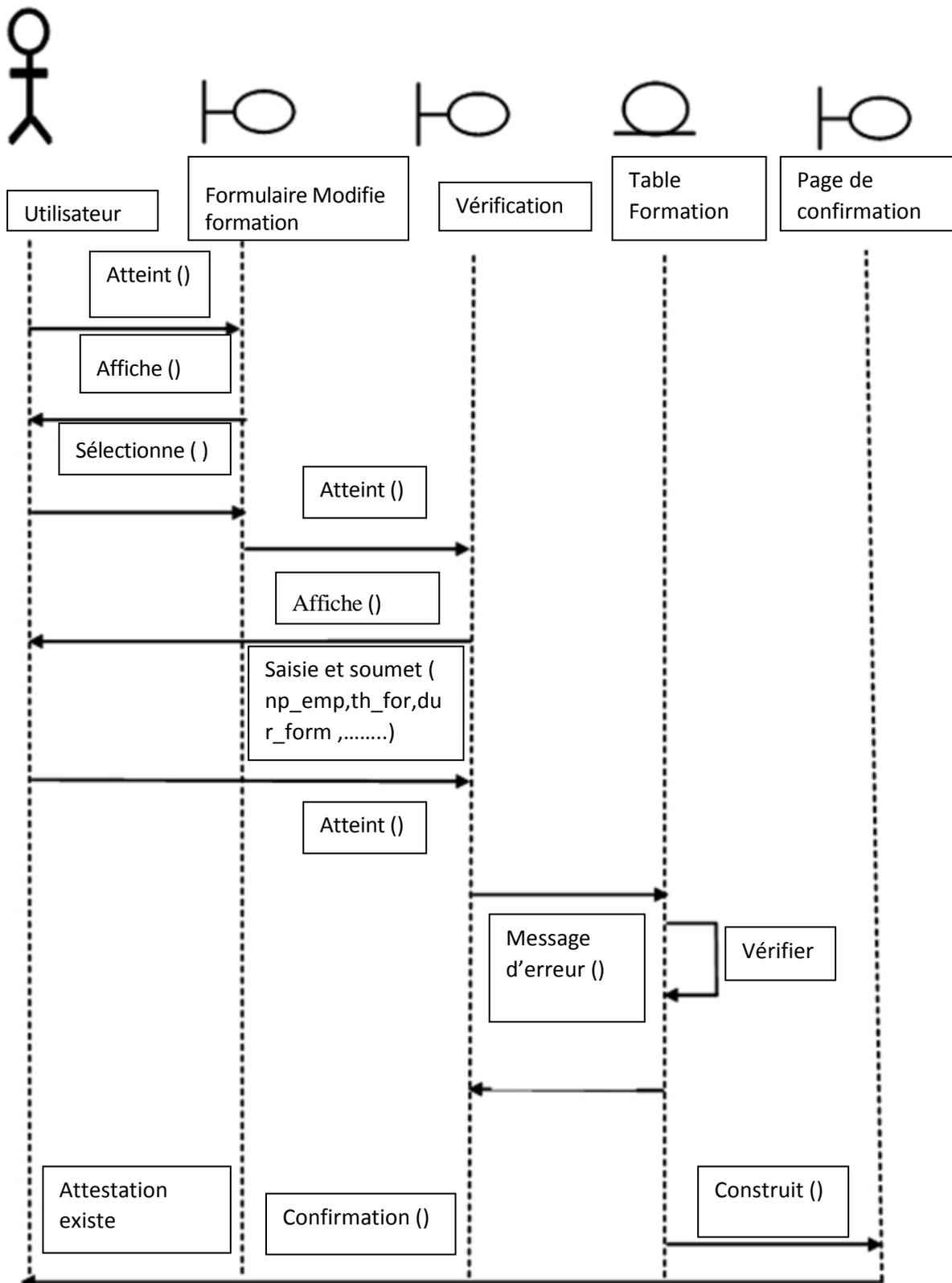


Figure III.9 : Diagramme de séquence du cas d'utilisation : «Modifie une formation »

- Description du cas d'utilisation avec des scénarios:

Cas utilisation : modifie une formation.

Acteur : ADMIN

Résumé: Ce cas d'utilisation permet à l'administrateur (ADMIN) de modifier une formation.

Scénario :

- 1- L'ADMIN atteint son espace.
 - 2- Le système affiche « ADMIN ».
 - 3- L'utilisateur sélectionne la rubrique « liste Formation»
 - 4- Le système retourne la liste des formations.
 - 5- L'utilisateur sélectionne la rubrique « modifie»
 - 6- Le système retourne le formulaire « modifie une formation».
 - 7- L'ADMIN remplit le formulaire et clique sur le bouton « modifie ».
 - 8- Le système va modifier la formation après avoir vérifié la validité des données saisies.
- Diagramme de séquence du cas d'utilisation : «Supprimé un contrat»

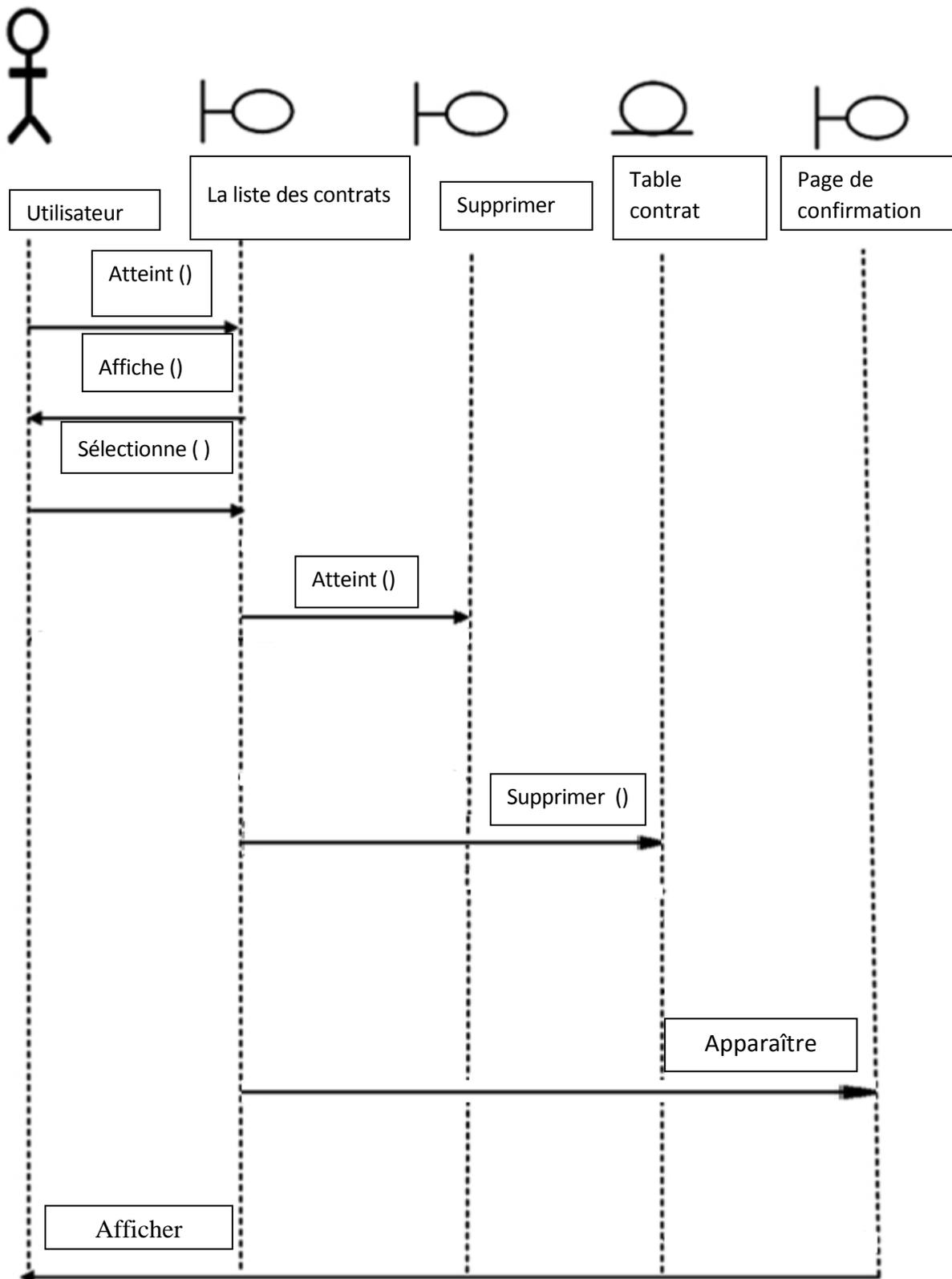


Figure III.10 : Diagramme de séquence du cas d'utilisation : «supprimé un contrat »

- Description du cas d'utilisation avec des scénarios:

Cas utilisation : supprimé un contrat.

Acteur : ADMIN

Résumé: Ce cas d'utilisation permet à l'administrateur (ADMIN) de modifier une formation.

Scénario :

- 1- L' ADMIN atteint son espace.
- 2- Le système affiche « ADMIN ».
- 3- L'utilisateur sélectionne la rubrique « liste contrat»
- 4- Le système retourne la liste des contrats.
- 5- L'utilisateur sélectionne la rubrique «supprimé»
- 6- Le système va supprimer le contrat.

III.3.2.2 Diagrammes de classe pour l'itération3

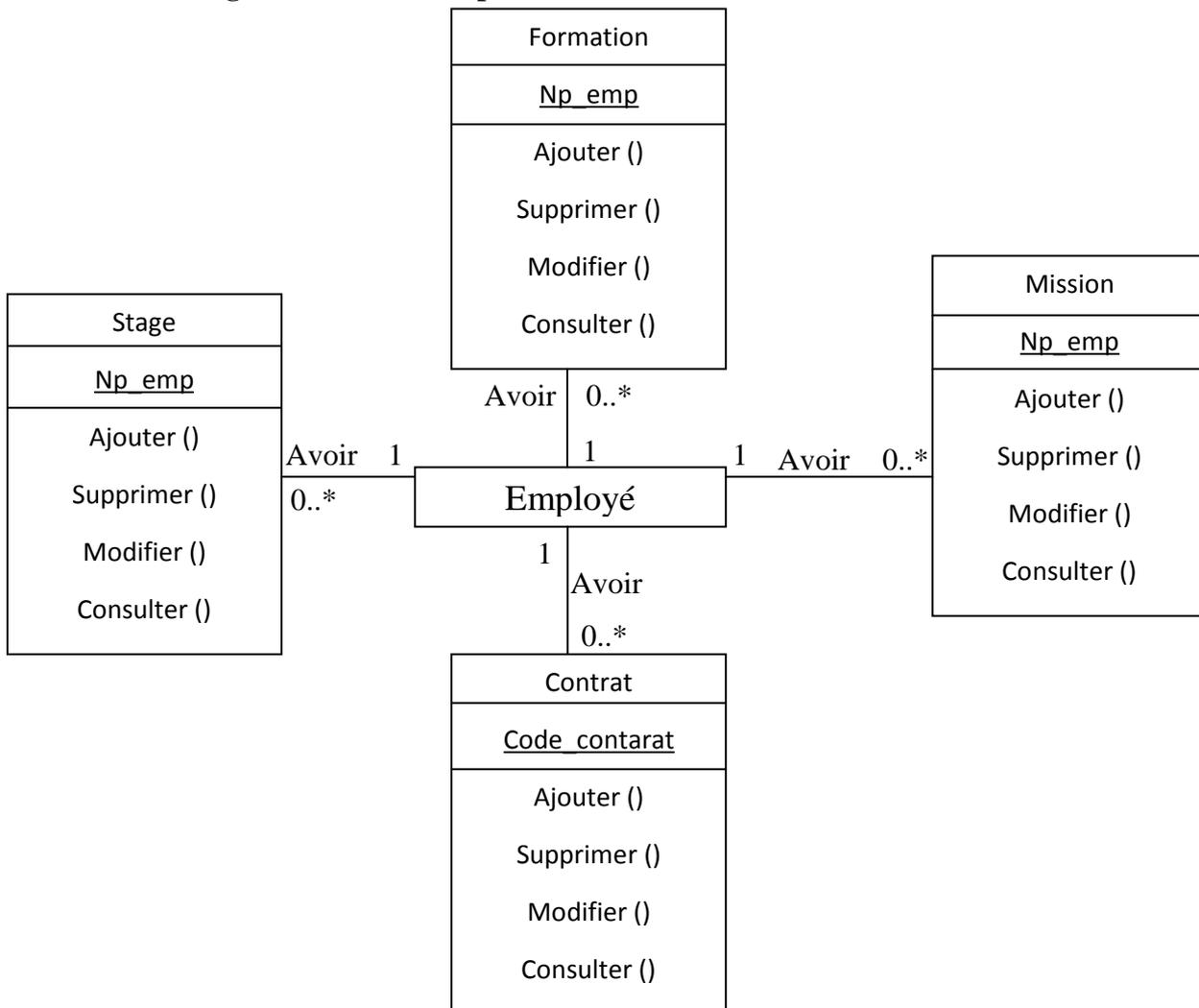


Figure III.11 : Diagramme de classes pour l'itération3

III.3.2 Réalisation de l'itération 4

III.3.2.1 Diagrammes de séquence pour itération 4:

- Diagramme de séquence du cas d'utilisation : «Ajouter une Absence»

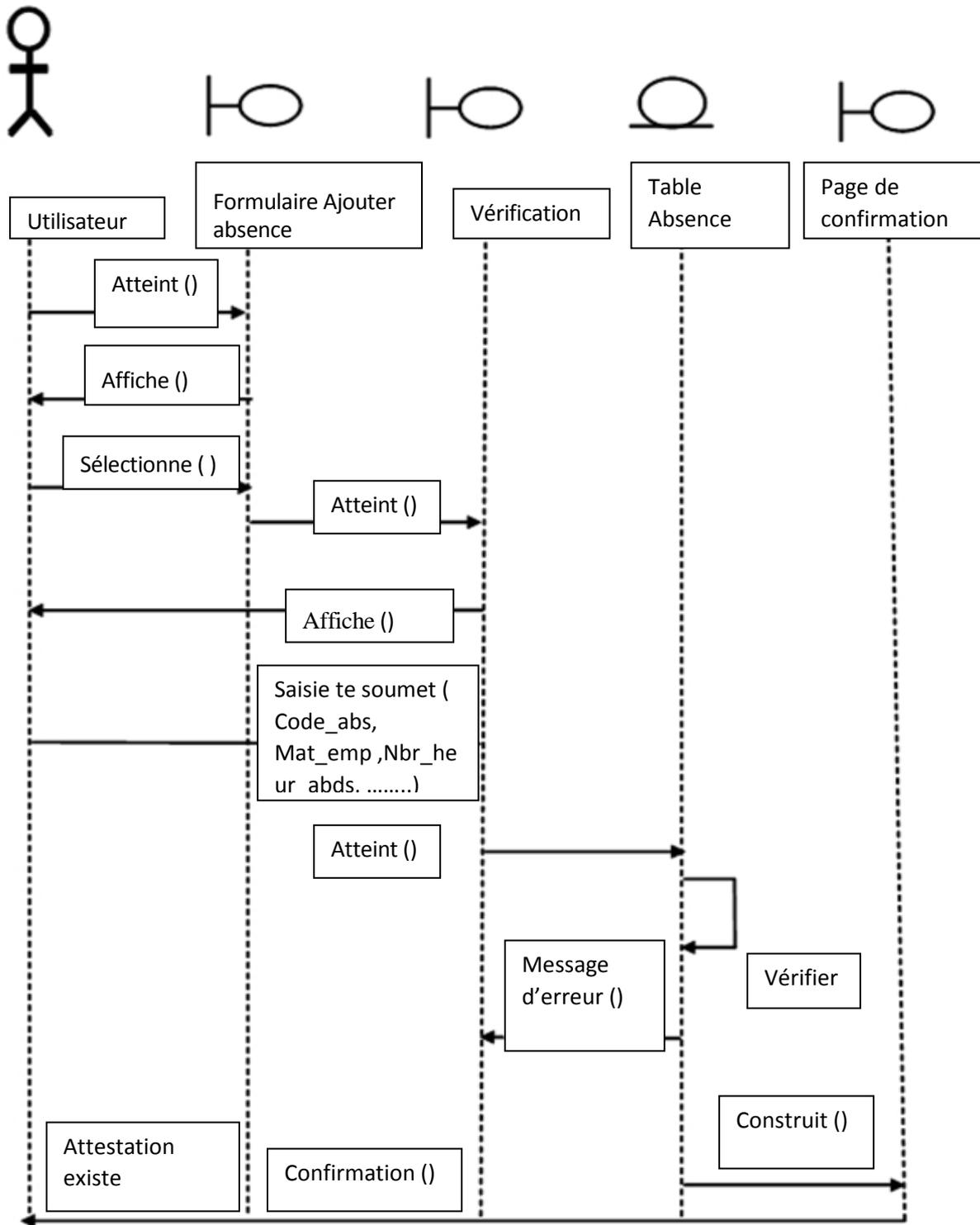


Figure III.12 : Diagramme de séquence du cas d'utilisation : «Ajouter une Absence »

- Description du cas d'utilisation avec des scénarios:

Cas utilisation : Ajouter une mission.

Acteur : ADMIN

Résumé: Ce cas d'utilisation permet à l'administrateur (ADMIN, POINTEUR) d'ajouter une absence.

Scénario :

- 1- L' ADMIN, POINTEUR atteint son espace.
- 2- Le système affiche « ADMIN ».
- 3- L'utilisateur sélectionne la rubrique « Absence»
- 3- L'utilisateur sélectionne la rubrique « Ajouter Absence»
- 4- Le système retourne le formulaire « Ajouter une mission».
- 5- L' ADMIN remplit le formulaire et clique sur le bouton « Enregistrer Absence ».
- 6- Le système ajoute une nouvelle mission après avoir vérifié la validité Des données saisies.

- Diagramme de séquence du cas d'utilisation : «**Modifie une sanction** »

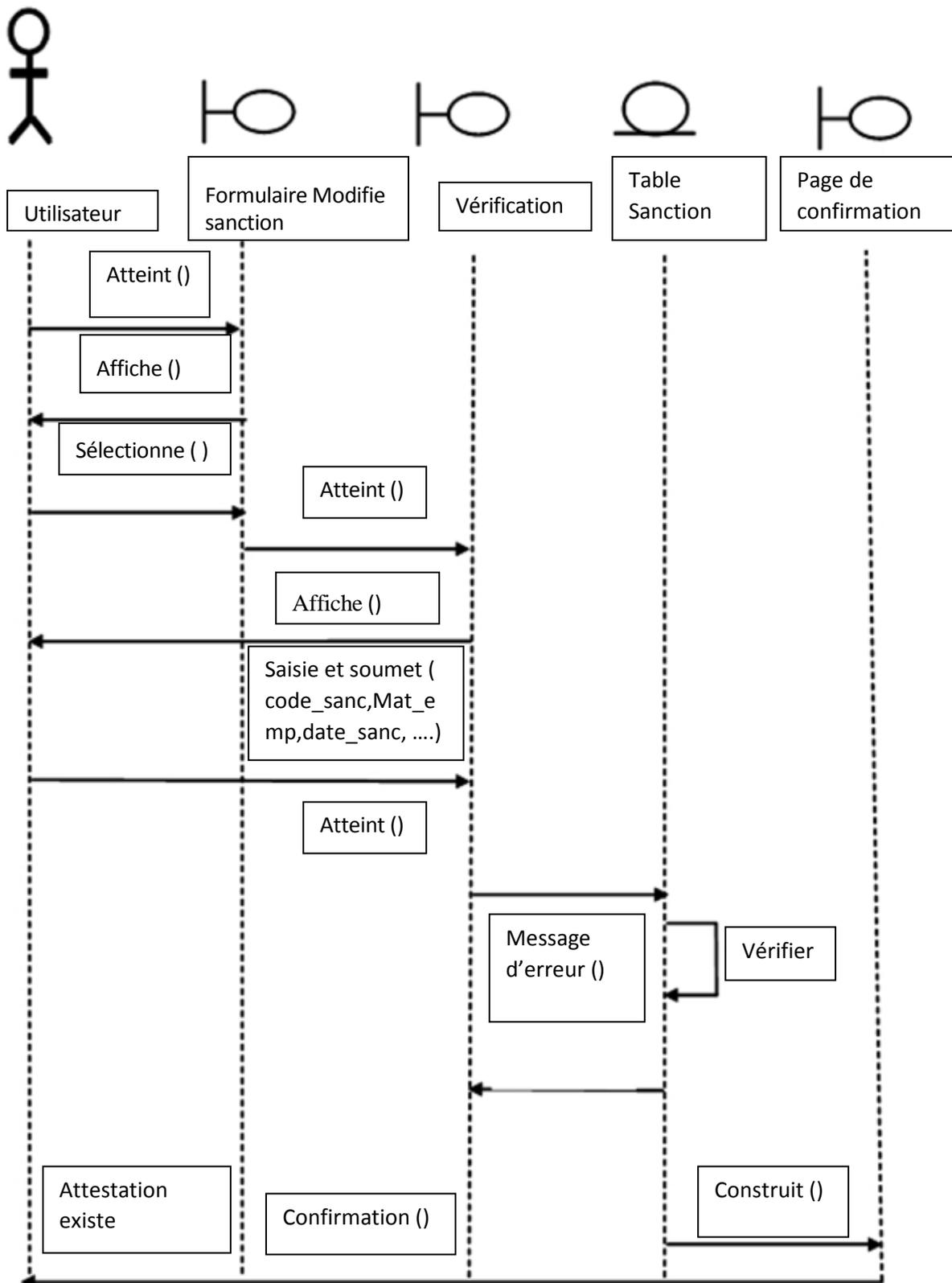


Figure III.13 : Diagramme de séquence du cas d'utilisation : «Modifie une Sanction »

- Description du cas d'utilisation avec des scénarios:

Cas utilisation : modifie une sanction.

Acteur : ADMIN

Résumé: Ce cas d'utilisation permet à l'administrateur (ADMIN) de modifier une formation.

Scénario :

- 1- L'ADMIN atteint son espace.
 - 2- Le système affiche « ADMIN ».
 - 3- L'utilisateur sélectionne la rubrique « liste Sanction»
 - 4- Le système retourne la liste des sanctions.
 - 5- L'utilisateur sélectionne la rubrique « modifie»
 - 6- Le système retourne le formulaire « modifie une Sanction».
 - 7- L'ADMIN remplit le formulaire et clique sur le bouton « modifie ».
 - 8- Le système va modifier la formation après avoir vérifié la validité des données saisies.
- Diagramme de séquence du cas d'utilisation : «Supprimé un congé»

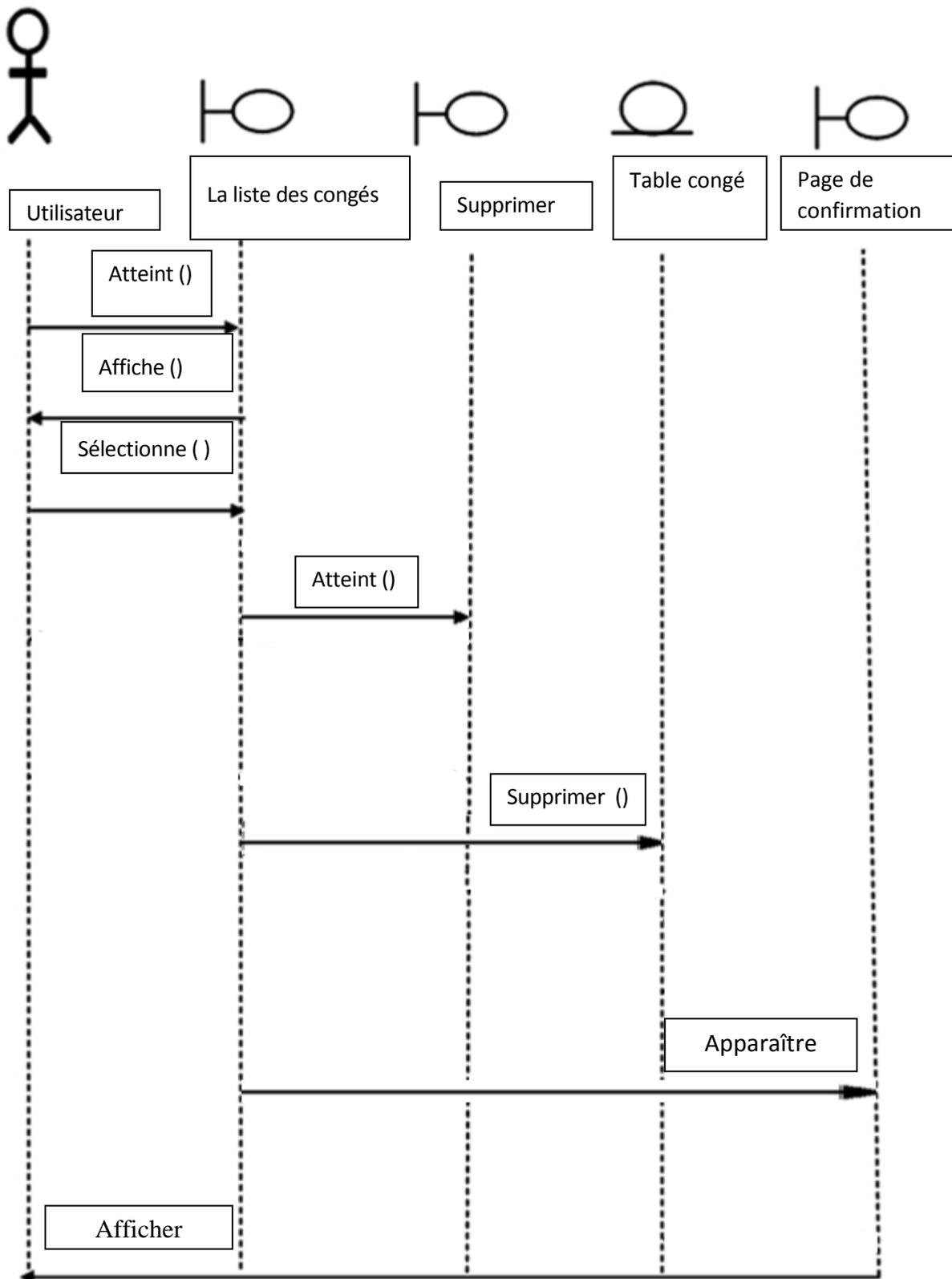


Figure III.14 : Diagramme de séquence du cas d'utilisation : «supprimé un congé »

- Description du cas d'utilisation avec des scénarios:

Cas utilisation : supprimé un congé.

Acteur : ADMIN

Résumé: Ce cas d'utilisation permet à l'administrateur (ADMIN) de modifier une formation.

Scénario :

- 1- L' ADMIN atteint son espace.
- 2- Le système affiche « ADMIN ».
- 3- L'utilisateur sélectionne la rubrique « liste congé»
- 4- Le système retourne la liste des congés.
- 5- L'utilisateur sélectionne la rubrique «supprimé»
- 6- Le système va supprimer le congé.

III.3.2.2 Diagrammes de classe pour l'itération4

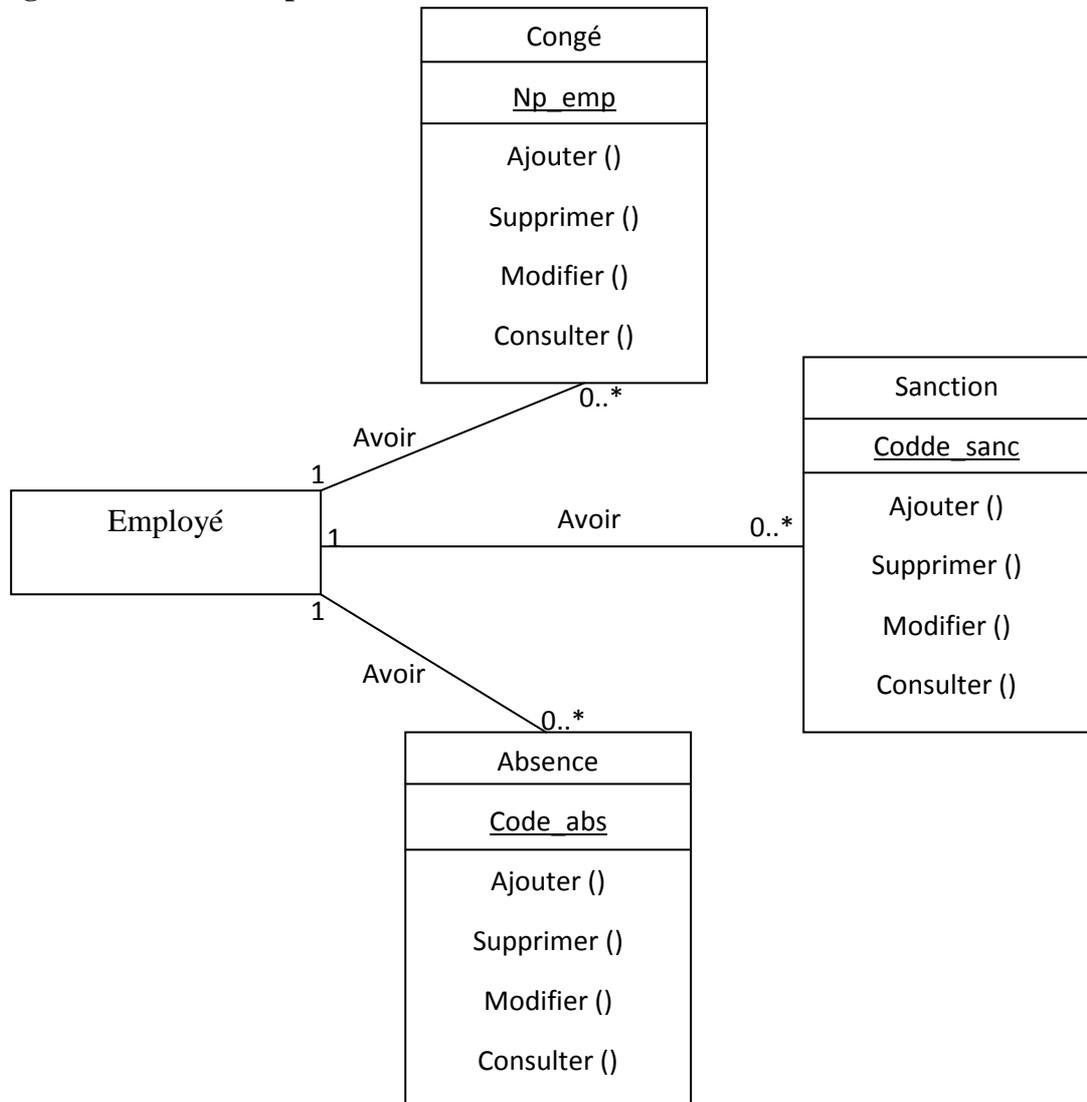


Figure III.15 : Diagramme de classes pour l'itération4

III.4 Conclusion :

Dans ce chapitre nous avons utilisé le processus unifié RUP pour l'analyse et la conception de notre application en utilisant le langage de modélisation UML. Nous avons commencé par la définition des besoins qui ont été traduits par la suite en formalismes graphiques à l'aide des diagrammes offerts par le langage UML s'appuyant sur l'analyse. La partie conception a mis en évidence les différents composants de l'application et cela dans le but de la construction de notre application.

Dans le chapitre suivant, qui touche à la réalisation, nous essayerons de construire notre application par itération.

Chapitre

IV

Réalisation

Introduction

Après avoir présenté dans le chapitre précédent les différentes étapes d'analyse et de conception, nous allons présenter dans ce dernier chapitre l'environnement de développement, les outils qui ont servi à la réalisation de notre application, pour illustrer quelques fonctionnalités de l'application.

V.1 Les outils de développement :

- **GlassFishServeur :**

C'est un logiciel, écrit en java, qui permet de faire le mapping entre ObjetsJava et objets stockés en base relationnelle en assurant la persistance, et aussi s'occupe du transfert des classes Java dans les tables de la BDD et des types de données dans les types de données SQL.

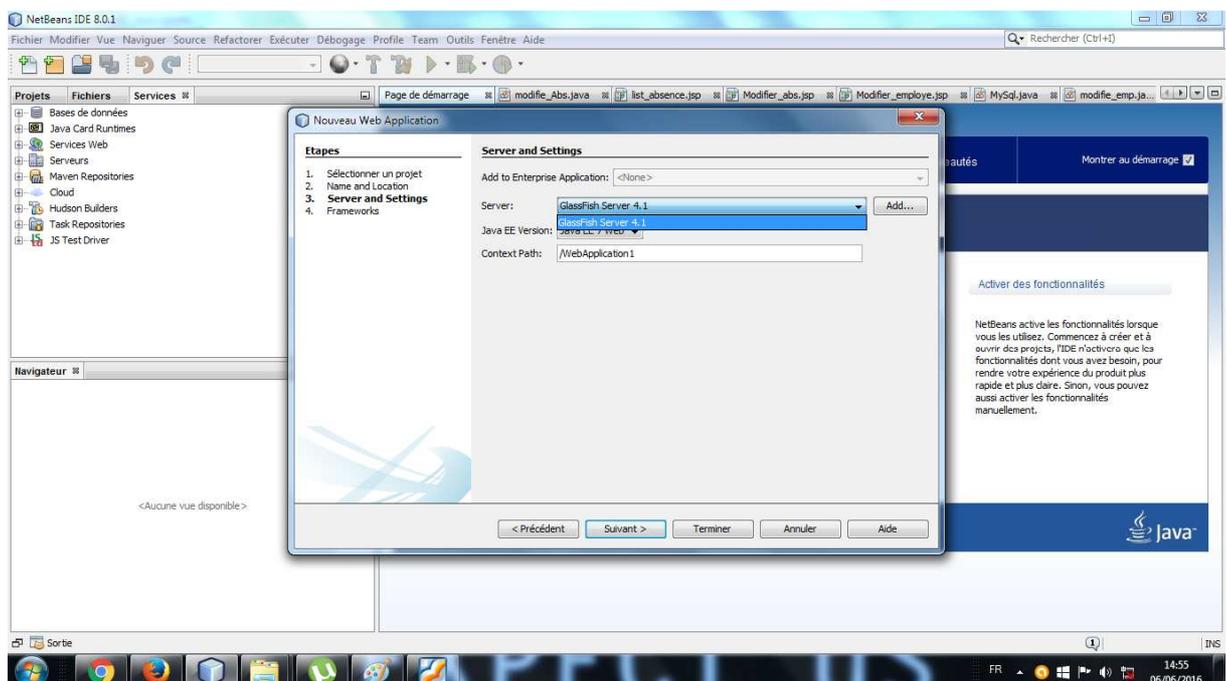


Figure IV.1: Interface Glassfish

WampServer :

WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques. Est un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (php) et PHPMyAdmin pour l'administration web des bases de données.

Il possède également PHPMyAdmin pour gérer plus facilement les bases de données.

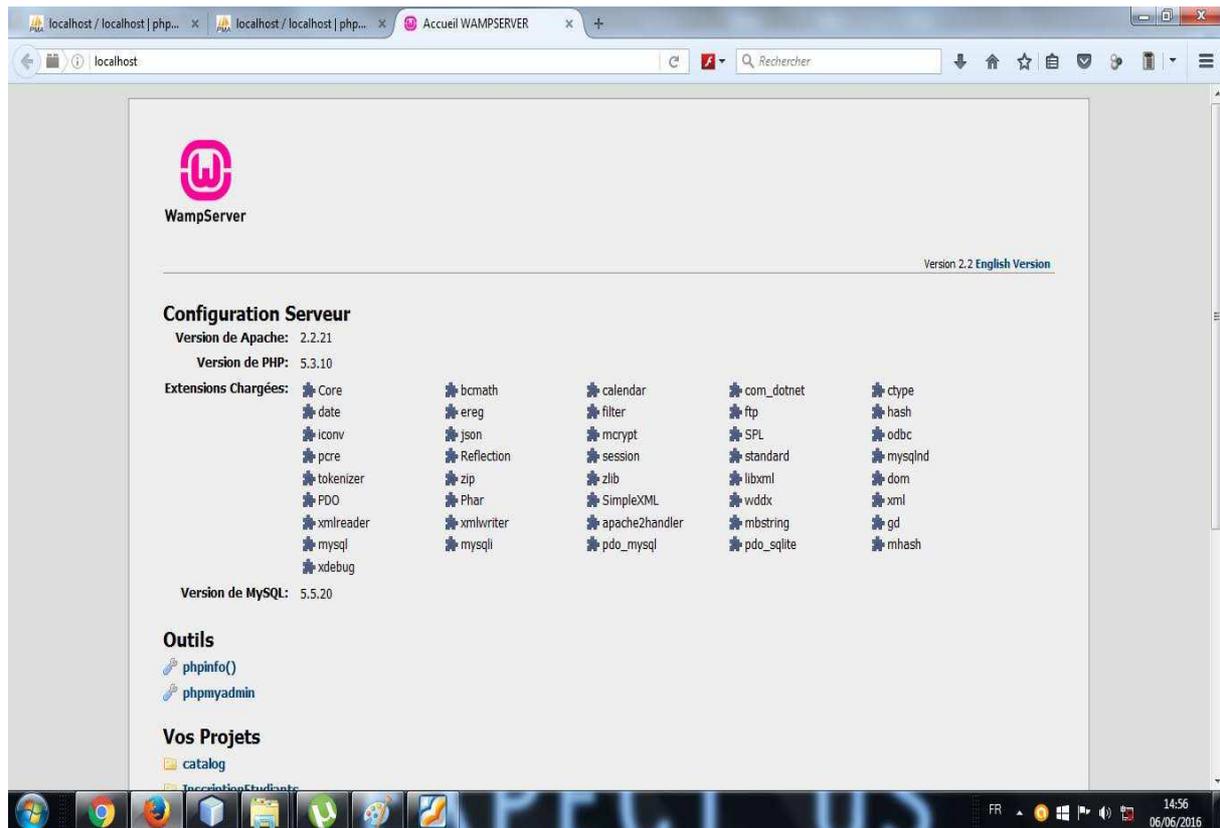


Figure IV.2: Interface de WampServer

- **PhpMyAdmin :**

PhpMyAdmin est une application web permettant de gérer un serveur de bases de données MySQL. Cette interface est écrite en PHP et s'appuie sur le serveur Apache. La figure suivante montre une copie d'écran de la page d'accueil de PhpMyAdmin.

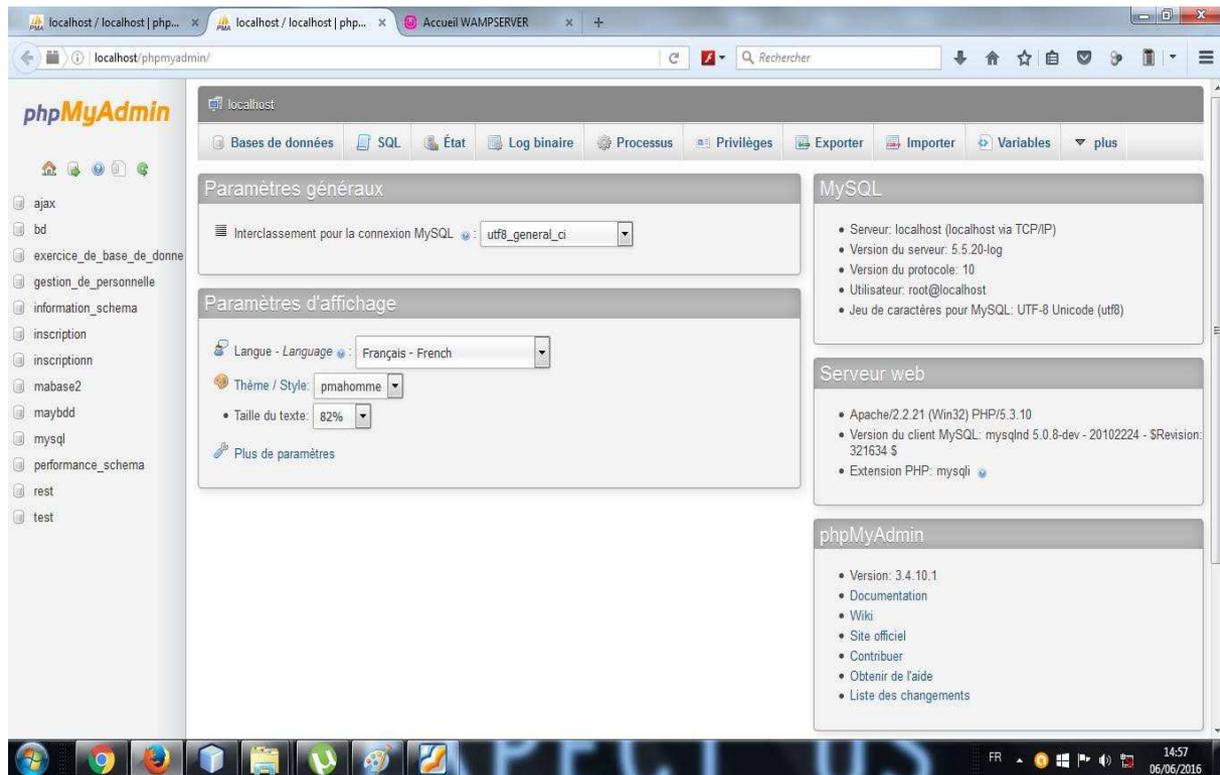


Figure IV.3: Interface dePhpMyAdmin

- **Serveur Apache Tomcat:**

Apache Tomcat est une implémentation open source d'un conteneur web qui permet donc d'exécuter des applications web reposant sur les technologies servlets et JSP.

Tomcat est diffusé en open source sous une licence Apache. C'est aussi l'implémentation de référence des spécifications servlets jusqu'à la version 2.4 et JSP jusqu'à la version 2.0 implémentées dans les différentes versions de Tomcat.

- **Netbeans**

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL et GPLv2 (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java.

NetBeans constitue par ailleurs une plate forme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plate forme.

L'IDE Netbeans s'enrichit à l'aide de plugins. Il comprend toutes les caractéristiques d'un IDE moderne (coloration syntaxique, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages web, etc).

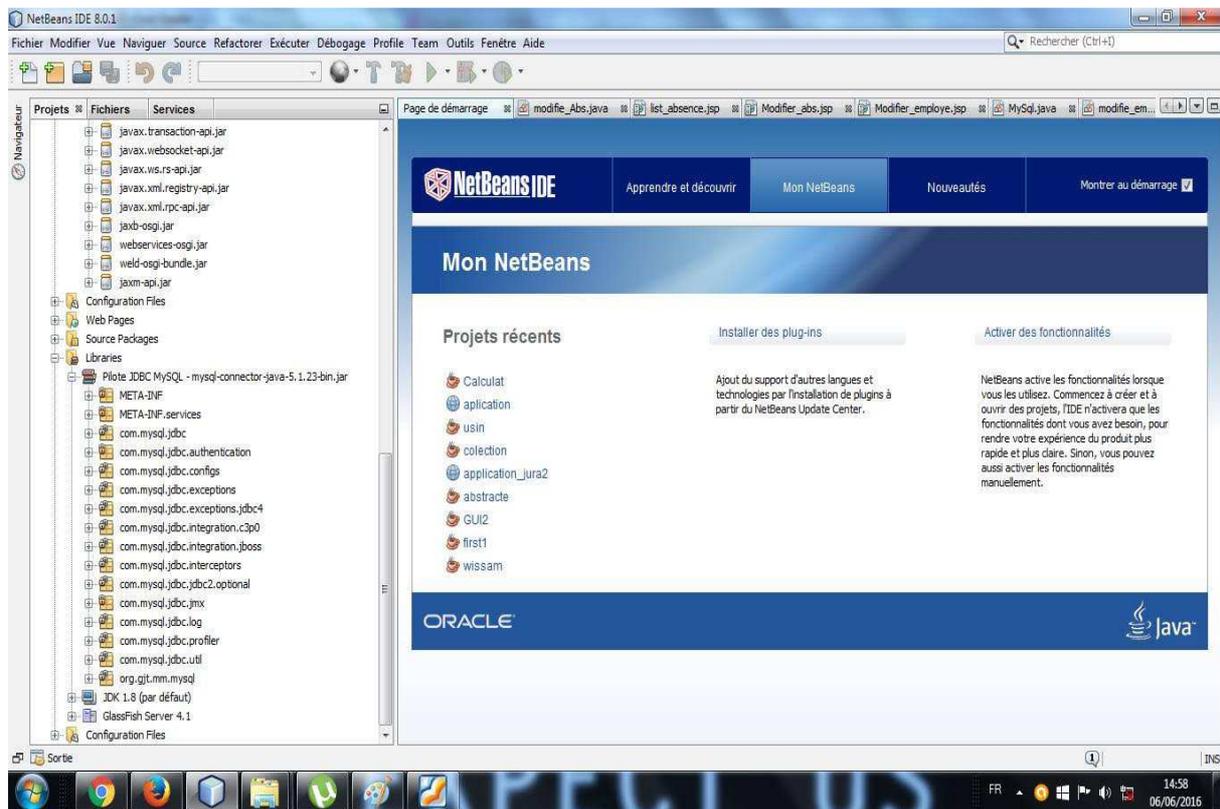


Figure IV.4 : Interface NetBeans

V.2 Les langages de programmation :

- **Java EE**

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

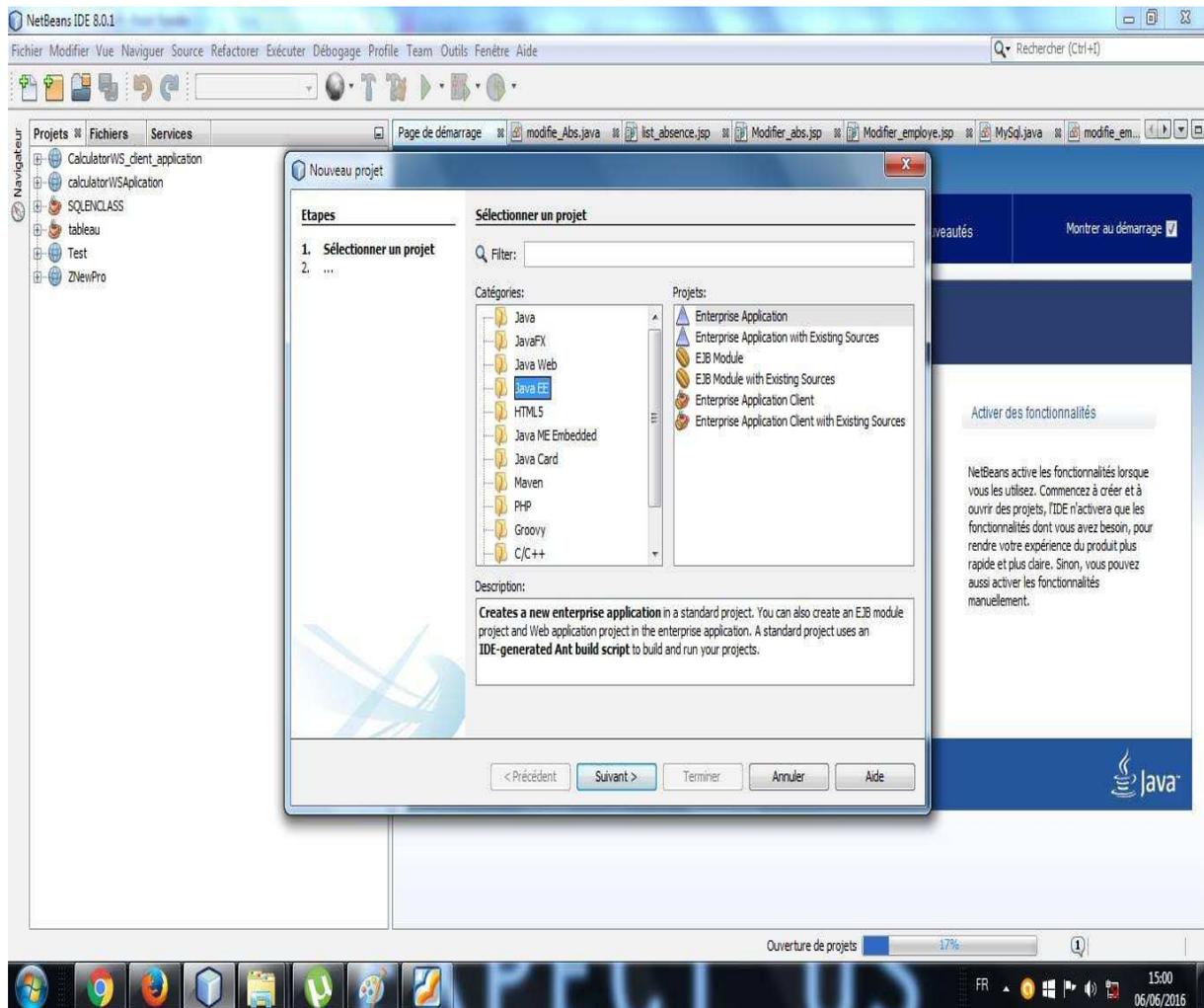


Figure IV.5: L'interface de JEE

- **XHTML** :(Extensible Hyper texte Markup Language)

XHTML est un langage balisé servant à l'écriture de pages du World Wide Web. XHTML est le successeur de HTML (de l'anglais HyperText Markup Language), XHTML respectant la syntaxe définie par XML, plus récente et plus simple que la syntaxe définie par SGML respectée par HTML.

V.3 les Itérations :

V.3.1 Itération I :

- **Son modèle relationnel**

Utilisateur (ID_util, Nom_util, Prenom_util, Date_naiss, login, pw, Adresse, type).

- **Table Utilisateur :**

<input type="checkbox"/>	1	Id_util	int(10)		Non	Aucune	AUTO_INCREMENT		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	2	Nom_util	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	3	Prenom_util	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	4	Date_naiss	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	5	login	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	6	pw	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	7	Adresse	varchar(500)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	8	type	int(1)		Non	Aucune			Modifier		Supprimer	plus ▼

V.3.2 Itération II :

- **Son modèle relationnel**

Employé (Mate_emp, Nom_emp, Prenom_emp, Date_naiss_emp, Lieu_naiss_emp, Adresse_emp, Sexe_emp, Nation_emp, Sit_fami_emp, Nss_emp, Tele_emp, Email_emp, Date_recrut_emp, Srvc_emp, Fonct_emp, Grade_emp, Position_emp).

Utilisateur (ID_util, Nom_util, Prenom_util, Date_naiss, login, pw, Adresse, type).

- **Table Employé**

<input type="checkbox"/>	1	Mat_emp	int(10)		Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	2	Nom_emp	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	3	Prenom_emp	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	4	Date_naiss_emp	varchar(20)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	5	Lieu_naiss_emp	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	6	Adresse_emp	varchar(500)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	7	Sexe_emp	int(1)		Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	8	Nation_emp	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	9	Sit_fami_emp	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	10	Nss_emp	varchar(12)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	11	Tele_emp	varchar(10)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	12	Email_emp	varchar(60)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	13	Date_recrut_emp	varchar(20)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	14	Srvc_emp	varchar(50)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	15	Fonct_emp	varchar(50)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	16	Grade_emp	varchar(50)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	17	Position_emp	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼

V.3.3 Itération III:

- **Son modèle relationnel**

Employé (Mate_emp, Nom_emp, Prenom_emp, Date_naiss_emp, Lieu_naiss_emp, Adresse_emp, Sexe_emp, Nation_emp, Sit_fami_emp, Nss_emp, Tele_emp, Email_emp, Date_recrut_emp, Srvc_emp, Fonct_emp, Grade_emp, Position_emp).

Mission(np_emp, foc_emp, Mate_emp*, afec, Ob_miss, dist, date_deb_miss, date_fin_miss, Moy_tran).

Utilisateur (ID_util, Nom_util, Prenom_util, Date_naiss, login, pw, Adresse, type).

Quelque table pour l'itération3

- **Table Employé :**

<input type="checkbox"/>	1	Mat_emp	int(10)		Non	Aucune		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	2	Nom_emp	varchar(30)	latin1_swedish_ci	Non	Aucune		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	3	Prenom_emp	varchar(30)	latin1_swedish_ci	Non	Aucune		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	4	Date_naiss_emp	varchar(20)	latin1_swedish_ci	Non	Aucune		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	5	Lieu_naiss_emp	varchar(30)	latin1_swedish_ci	Non	Aucune		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	6	Adresse_emp	varchar(500)	latin1_swedish_ci	Non	Aucune		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	7	Sexe_emp	int(1)		Non	Aucune		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	8	Nation_emp	varchar(30)	latin1_swedish_ci	Non	Aucune		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	9	Sit_fami_emp	varchar(30)	latin1_swedish_ci	Non	Aucune		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	10	Nss_emp	varchar(12)	latin1_swedish_ci	Non	Aucune		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	11	Tele_emp	varchar(10)	latin1_swedish_ci	Non	Aucune		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	12	Email_emp	varchar(60)	latin1_swedish_ci	Non	Aucune		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	13	Date_recrut_emp	varchar(20)	latin1_swedish_ci	Non	Aucune		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	14	Srvc_emp	varchar(50)	latin1_swedish_ci	Non	Aucune		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	15	Fonct_emp	varchar(50)	latin1_swedish_ci	Non	Aucune		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	16	Grade_emp	varchar(50)	latin1_swedish_ci	Non	Aucune		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	17	Position_emp	varchar(30)	latin1_swedish_ci	Non	Aucune		Modifier		Supprimer	plus ▼

- **Table absence :**

#	Colonne	Type	Interclassement	Attributs	Null	Défaut	Extra	Action			
<input type="checkbox"/>	1	np_emp	varchar(30)	latin1_swedish_ci	Non	Aucune		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	2	fon_emp	varchar(100)	latin1_swedish_ci	Non	Aucune		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	3	Mat_emp	varchar(5)	latin1_swedish_ci	Non	Aucune		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	4	afect	varchar(4)	latin1_swedish_ci	Non	Aucune		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	5	Ob_miss	varchar(100)	latin1_swedish_ci	Non	Aucune		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	6	dist	varchar(100)	latin1_swedish_ci	Non	Aucune		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	7	Date_deb_miss	varchar(30)	latin1_swedish_ci	Non	Aucune		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	8	Date_fin_miss	varchar(30)	latin1_swedish_ci	Non	Aucune		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	9	Moy_Tran	varchar(100)	latin1_swedish_ci	Non	Aucune		Modifier		Supprimer	plus ▼

- **Table Utilisateur:**

<input type="checkbox"/>	1	Id_util	int(10)		Non	Aucune	AUTO_INCREMENT		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	2	Nom_util	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	3	Prenom_util	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	4	Date_naiss	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	5	login	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	6	pw	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	7	Adresse	varchar(500)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	8	type	int(1)		Non	Aucune			Modifier		Supprimer	plus ▼

V.3.4 Itération IV:

- **Son modèle relationnel**

Employé (Mate_emp, Nom_emp, Prenom_emp, Date_naiss_emp, Lieu_naiss_emp, Adresse_emp, Sexe_emp, Nation_emp, Sit_fami_emp, Nss_emp, Tele_emp, Email_emp, Date_recrut_emp, Srvc_emp, Fonct_emp, Grade_emp, Position_emp).

Absence(code_abs, Mate_emp*, Nbr_heur_abs, Date_deb_abs, Motis_abs).

Utilisateur (ID_util, Nom_util, Prenom_util, Date_naiss, login, pw, Adresse, type).

Quelque table pour l'itération4

- **Table Employé :**

<input type="checkbox"/>	1	Mat_emp	int(10)		Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	2	Nom_emp	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	3	Prenom_emp	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	4	Date_naiss_emp	varchar(20)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	5	Lieu_naiss_emp	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	6	Adresse_emp	varchar(500)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	7	Sexe_emp	int(1)		Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	8	Nation_emp	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	9	Sit_fami_emp	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	10	Nss_emp	varchar(12)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	11	Tele_emp	varchar(10)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	12	Email_emp	varchar(60)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	13	Date_recrut_emp	varchar(20)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	14	Srvc_emp	varchar(50)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	15	Fonct_emp	varchar(50)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	16	Grade_emp	varchar(50)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	17	Position_emp	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼

- **Table absence :**

<input type="checkbox"/>	1	Code_abs	int(10)		Non	Aucune	AUTO_INCREMENT		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	2	Mat_emp	varchar(6)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	3	Nbr_heur_abs	int(4)		Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	4	Dat_deb_abs	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	5	Motif_abs	varchar(300)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼

- **Table Utilisateur:**

<input type="checkbox"/>	1	Id_util	int(10)		Non	Aucune	AUTO_INCREMENT		Modifier		Supprimer	plus ▼
<input type="checkbox"/>	2	Nom_util	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	3	Prenom_util	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	4	Date_naiss	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	5	login	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	6	pw	varchar(30)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	7	Adresse	varchar(500)	latin1_swedish_ci	Non	Aucune			Modifier		Supprimer	plus ▼
<input type="checkbox"/>	8	type	int(1)		Non	Aucune			Modifier		Supprimer	plus ▼

V.4 Présentation de quelques interfaces :

Nous n'allons présenter que quelques interfaces :

V.4.1Présentation d'interface authentification :



Figure IV.6 : Interface page d'Authentification

V.3.2 Présentation d'interface d'accueil:

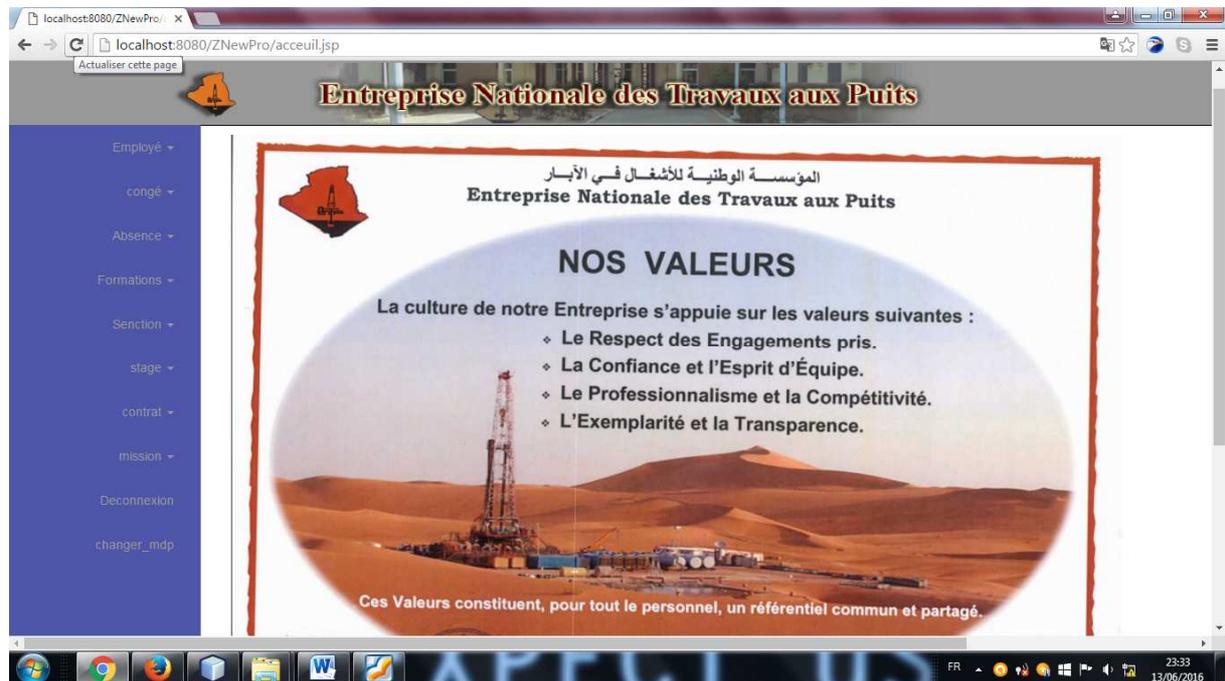


Figure IV.7 : Interface page d'accueil

V.3.3 Présentation d'interface d'un employé:

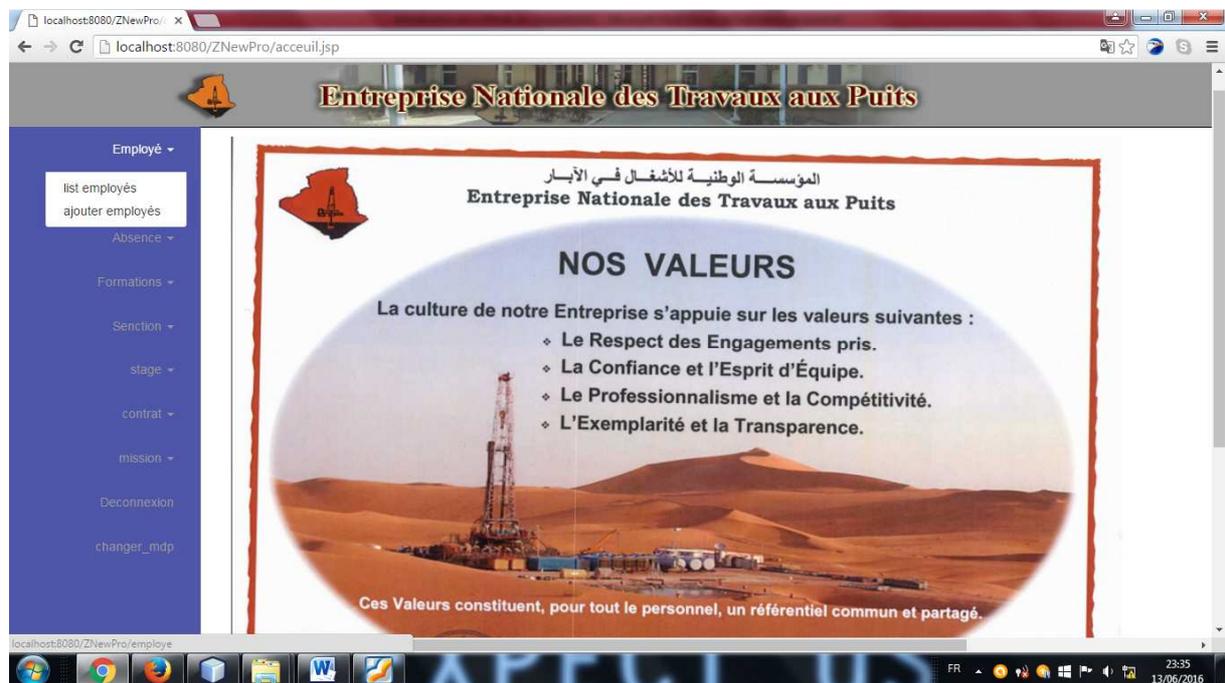


Figure IV.8 : Interface d'un employé

V.3.4 Présentation d'interface Ajout d'un employé :

Entreprise Nationale des Travaux aux Puits

المؤسسة الوطنية للأشغال في الآبار
Entreprise Nationale des Travaux aux Puits
Société Par Actions au Capital Social de 14800000000 DA
Certifiée ISO 9001: 2008 - OHSAS 18001: 2007 - ISO 14001: 2004

AJOUTER EMPLOYE

Matricule: Numéro de sécurité social:

Nom: Téléphone:

Prenom: Email:

Date de naissance: Date de recrutement:

lieu de naissance: Service:

Adress: Fonction:

Figure IV.9 : Interface d'ajout d'un employé**Conclusion :**

Dans ce chapitre nous avons présenté les outils et l'environnement développement de l'application. Nous avons en outre explicité les composants de la base de données, puis pour terminer nous avons présenté quelques interfaces de l'application.

Conclusion Générale

Conclusion générale

La réalisation de ce travail nous a permis d'approfondir nos connaissances en systèmes d'information des entreprises, de découvrir le domaine de gestion du personnel, de voir sur le champs de travail les différentes tâches effectuées par les gestionnaires du personnel et d'acquérir un certain nombre de connaissances et règles de gestion. Comme elle nous a permis de :

- nous familiariser avec plusieurs langages de programmation à savoir Java EE.
- Enrichir nos connaissances sur les applications java, comment les développer, les déployer et les invoquer au sein du serveur d'application GlassFish.
- De se familiariser avec les environnements de développement orienté web, à savoir : Netbeans, MySQL pour les bases de données.
- Ainsi de maîtriser les concepts et les outils de la méthode de conception UML. Ainsi que la démarche de conception orientée objet (RUP, Rational Unified Process).

L'application à laquelle nous avons abouti, traite la gestion du personnel de l'ENTP qui adopte l'architecture client/serveur. Elle offre aux différents utilisateurs plus de souplesse et d'efficacité dans l'exécution de leurs tâches.

Pour mener à terme notre projet, nous avons jugé nécessaire de présenter quelques généralités sur la technologie Java EE, le concept client/serveur, ainsi que les technologies de développement d'application java existantes.

Ensuite, nous avons présenté notre organisme d'accueil, et étudié le système existant afin de déterminer les anomalies et essayer d'y remédier.

Après les étapes de conception et de réalisation, nous avons abouti à une application java EE répondant aux principaux problèmes qui le monoposte, elle est désormais accessible depuis tous les ordinateurs des utilisateurs concernés, facilitant ainsi la gestion du personnel et allégeant la masse de travail effectué par un seul utilisateur auparavant. Cependant quelques améliorations et extensions peuvent être apportées à cette application.

En guise de perspectives, les possibilités d'évolutions de ce projet sont nombreuses allant jusqu'à la gestion complète de tous les départements tout en mettant l'accent sur l'aspect coopératif de ces structures.

Enfin, nous espérons que notre travail sera de grand intérêt pour l'ENTP et un guide pour les nouvelles promotions.

Annexes

1. Introduction :

Dans ce chapitre, nous allons entamer le processus de développement par une analyse qui mettra en évidence les différents acteurs intervenant dans le système cible ainsi que leurs besoins. La phase conception, s'appuyant sur les résultats de la phase analyse donnera la modélisation des objectifs à atteindre. Pour ce faire, notre démarche va s'appuyer sur le langage UML, qui permet la représentation de la dynamique et la statique du système à travers les différents diagrammes qu'il offre.

2. Définition UML :

UML est le facteur de standardisation, car il est impossible de prétendre imposer une méthode, une manière de faire, à toute l'industrie. En revanche on peut prétendre. Définir un langage de modélisation qui, s'il est suffisamment général, sera adopté comme moyen de communication. UML est un langage qui permet de représenter des modèles, mais il ne définit pas le processus d'élaboration d'un modèle. Cependant, dans le cadre de la modélisation d'une application informatique, les acteurs d'UML préconise d'utiliser une démarche :

- Itérative et incrémentale.
- Guidée par les besoins des utilisateurs de système.
- Centrée sur l'architecture logicielle.

3. La modélisation UML :

UML fournit une panoplie d'outils permettant de présenter l'ensemble des éléments du monde objet (classe, objet...) ainsi que les liens qui les relie. Toutefois, étant donné qu'une seule représentation est trop subjective, UML fournit un moyen astucieux permettant de représenter diverse projection d'une même représentation grâce aux vues. une vue est constituée d'un ou plusieurs diagrammes.

On distingue trois types de vues :

-Les vues statique : représentant le système physiquement

- Diagramme de classe
- Diagramme d'objets
- Diagramme de composants
- Diagramme de déploiement

-Les vue Dynamique :

- Diagramme d'états
- Diagramme d'activités
- Diagramme de séquence
- Diagramme de collaboration

Annexe A Langage de modélisation UML

-Le vue fonctionnelle :

- Diagramme de cas d'utilisation

UML suit une démarche en trois étapes :

- Analyse
- Conception
- Implémentation

Les briques d'UML : composées de :

- Les éléments
- Les relations
- Les diagrammes

4. Eléments d'UML :

Il existe quatre types d'éléments dans UML :

- les éléments structurels.
- les éléments comportementaux.
- les éléments de regroupement.
- les éléments d'annotation.

4.1. Les éléments structurels : Ils représentent les parties statiques du modèle.

Ceux sont des représentations conceptuelles ou physiques d'un système

Il existe 7 types d'éléments structurels :

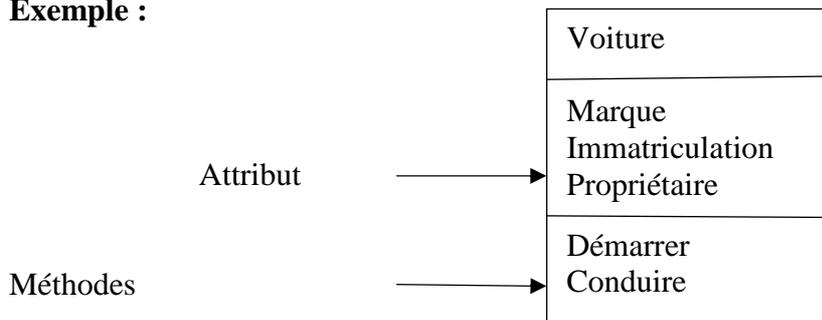
- Classe
- Interface
- Collaboration
- Cas d'utilisation
- Classe d'activité
- Composant
- Nœud

4.1.1. Classe :

C'est un ensemble d'éléments ayant les mêmes attributs, les mêmes opérations, les mêmes relations et la même sémantique.

Annexe A Langage de modélisation UML

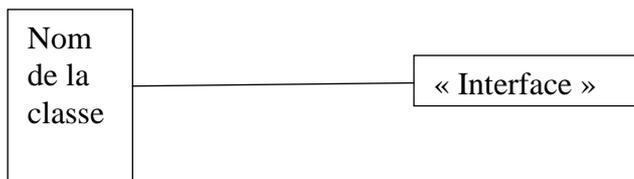
Exemple :



4.1.2. Interface :

C'est un ensemble d'opérations définissant la fonction d'un élément ou d'un composant. Elle définit seulement la signature des opérations.

Exemple :



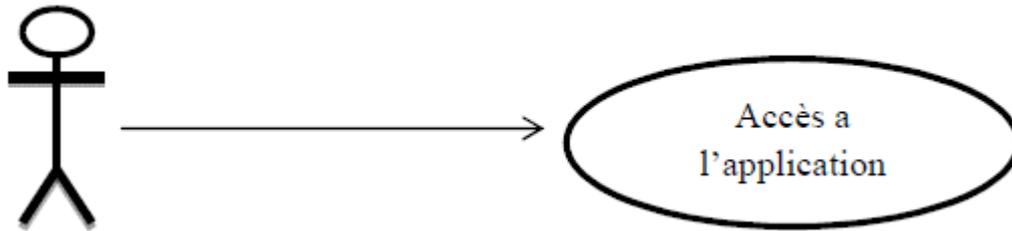
4.1.3 Collaboration :

la collaboration Elle définit une interaction entre plusieurs éléments. Elle a un rôle structurel et comportemental.

4.1.4 Cas d'utilisation :

- Il s'agit de la solution UML pour représenter le modèle conceptuel.
- Les use cases permettent de structurer les besoins des utilisateurs et les objectifs correspondants d'un système.
- Ils centrent l'expression des exigences du système sur ses utilisateurs : ils partent du principe que les objectifs du système sont tous motivés.
- Ils se limitent aux préoccupations "réelles" des utilisateurs ; ils ne présentent pas des solutions d'implémentation et ne forment pas un inventaire fonctionnel du système.
- Ils identifient les utilisateurs du système (acteurs) et leur interaction avec le système.
- Ils permettent de classer les acteurs et structurer les objectifs du système.

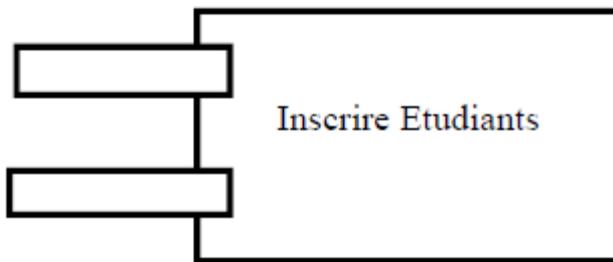
Exemple :



4.1.5 Classe d'activité : Classe particulière dont les objets possèdent un ou plusieurs processus pouvant lancer une activité de commande.

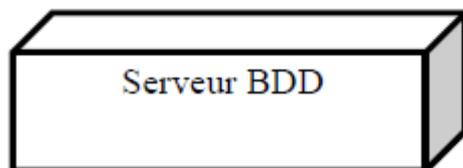
4.1.6 Composant : Le composant est une partie physique d'un système qui conforme à une spécification et fournit la réalisation d'un ensemble d'interface.il peut être un module.exe, com, javaBeans,etc ;il peut aussi être constitué de plusieurs sous composantes.

Exemple :



4.1.7 Noeud : C'est un élément physique. Lors d'une exécution, il représente une ressource ayant une capacité de calcul. En règle général,il a moins de la mémoire et souvent de capacitéde traitement .Un noeud est représenté par un cube.

Exemple :



4.2. Les éléments comportementaux

4.2.1 Les interactions :

Les interactions est l'ensemble des messages échanger entre les éléments du système(résultat de collaboration d'un groupe d'instances), elles sont des messages, des séquences d'actions ou bien des liens (relation entre des éléments). Une interaction peut être visualisée selon le

Annexe A Langage de modélisation UML

point de vue du temps (diagramme de séquences) ou de celui d'espace (diagramme de collaboration). Un message par exemple est une interaction représenté par une ligne fléchée, indiquant le nom de son opération.

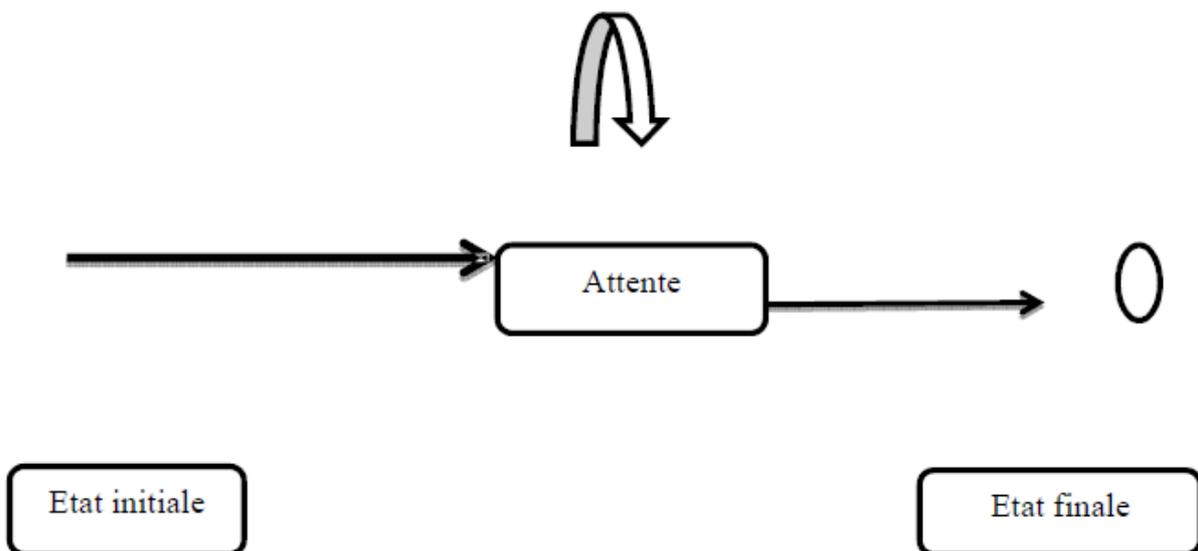
Exemple :



4.2.2 Etat :

La machine à état spécifie le cycle de vie d'un objet quand cet objet a un comportement dynamique. On représente graphiquement un état par un rectangle au coin arrondi (au centre), et en distingue les états de départ (à gauche) et terminaux (à droite). Comme décrit dans figure suivante.

Exemple :



4.3. Les éléments de regroupement :

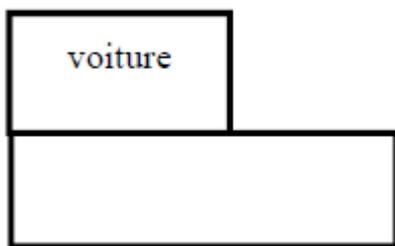
Les éléments de regroupements représentent les parties organisationnelles des modèles UML. Ce sont des boîtes dans lesquelles un modèle peut être décomposé. Il existe un seul type fondamental d'éléments de regroupement : le « paquetage ».

4.3.1 Package :

Un package en UML (ou paquetage en français) est un groupe d'éléments, dans le but de les grouper dans des ensembles cohérents. Un package peut contenir la plupart des éléments UML : classes, objets, cas d'utilisations, composantes, etc. Il peut également contenir des packages, créant une hiérarchie complète.

L'avantage des packages est qu'ils permettent de structurer les diagrammes et donnent une vision globale plus claire.

Exemple :



4.3.2 Les éléments d'annotation :

Ils représentent les parties explicatives modèles. Ceux des commentaires. Il existe 1 type d'élément d'annotations : Ils font partie des décorations.

Exemple :



5. Les Relation : sont les liens entre les éléments.

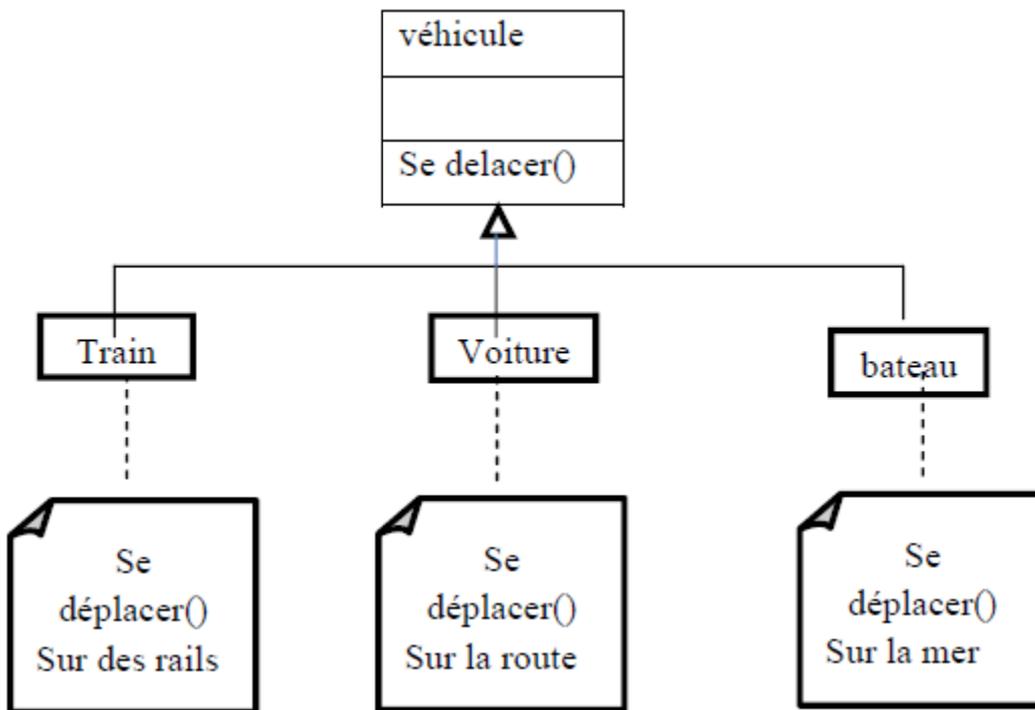
Il existe Cinq types de relation interclasse sont couramment utilisés (il en existe d'autres) :

- Héritage
- Dépendance
- Agrégation
- Composition
- Association

- **Héritage :**

- ✓ L'héritage est un mécanisme de transmission des propriétés d'une classe (ses attributs et méthodes) vers une sous-classe.
- ✓ Une classe peut être spécialisée en d'autres classes, afin d'y ajouter des caractéristiques spécifiques ou d'en adapter certaines.
- ✓ Plusieurs classes peuvent être généralisées en une classe qui les factorise, afin de regrouper les caractéristiques communes d'un ensemble de classes

Exemple :



- **Dépendance :**

C'est un lien sémantique entre 2 éléments .Un changement de l'un peut affecter la sémantique de l'autre.

Exemple :



- **Agrégation :**

L'agrégation est un cas particulier de relation d'association qui indique qu'une classe est une partie d'une autre classe. Cette relation comporte également les ordres de multiplicité. On la représente graphiquement en décorant la relation d'association par un losange.

Exemple :

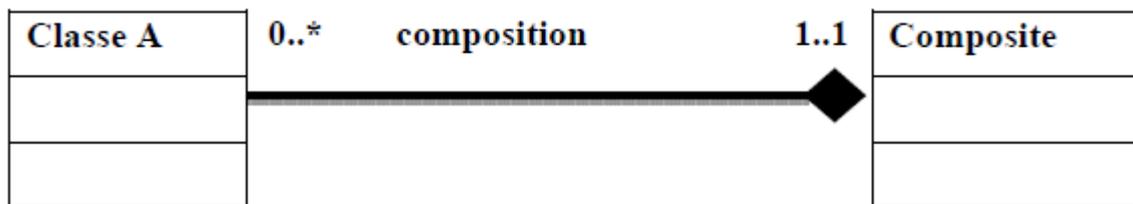


- **Composition :**

Composition Aussi appelée "agrégation forte" ou "agrégation par valeur", il s'agit en fait d'une agrégation à laquelle on impose des contraintes internes : un seul objet peut faire partie d'un composite (l'agrégat de la composition), et celui-ci doit gérer toutes ses parties. En clair, les composants sont totalement dépendants du composite.

En UML, la composition est représentée de la même manière que l'agrégation, mais le diamant est plein.

Exemple :

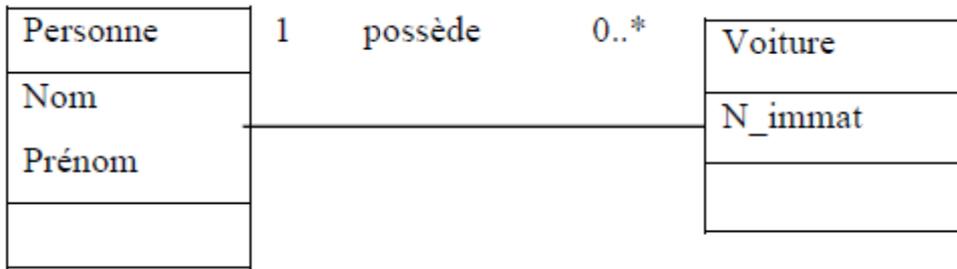


- **Association :**

C'est la relation la plus simple entre deux classes. Elle existe à partir du moment où l'une des deux classes sert de type à un attribut de l'autre, et que cet autre envoie des messages à la première (condition nécessaire pour une association). Simplement, une association indique que deux classes communiquent entre elles (dans un sens ou dans les deux sens).

En UML, une association est représentée par une ligne entre deux classes, possiblement accompagnée d'une flèche si l'association n'est pas bidirectionnelle.

Exemple :



6. Les différents types de diagrammes UML :

6.1 Diagramme cas d'utilisation :

- Il s'agit de la solution UML pour représenter le modèle conceptuel.(est de comprendre et structurer les besoins du client.)
- Les use cases permettent de structurer les besoins des utilisateurs et les objectifs correspondants d'un système.
- Ils centrent l'expression des exigences du système sur ses utilisateurs : ils partent du principe que les objectifs du système sont tous motivés.
- Ils se limitent aux préoccupations "réelles" des utilisateurs ; ils ne présentent pas de solutions d'implémentation et ne forment pas un inventaire fonctionnel du système.
- Ils identifient les utilisateurs du système (acteurs) et leur interaction avec le système.
- Ils permettent de classer les acteurs et structurer les objectifs du système.
- Ils servent de base à la traçabilité des exigences d'un système dans un processus de développement intégrant UML.

Exemple :



Acteur A

Cas d'utilisateur x

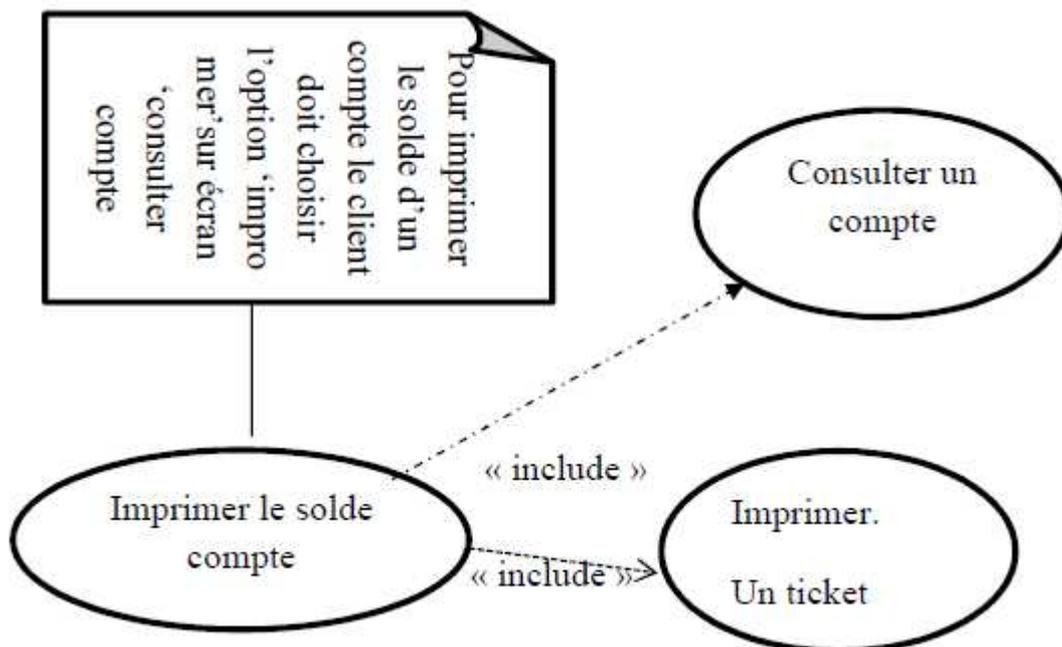
✓ **Intérêt des cas d'utilisation :**

- Le but de la conceptualisation est de comprendre et structurer les besoins du client.
- Il ne faut pas chercher l'exhaustivité, mais clarifier, filtrer et organiser les besoins
- Une fois identifiés et structurés, ces besoins :
 - Définissent le contour du système à modéliser (ils précisent le but à atteindre) .

Annexe A Langage de modélisation UML

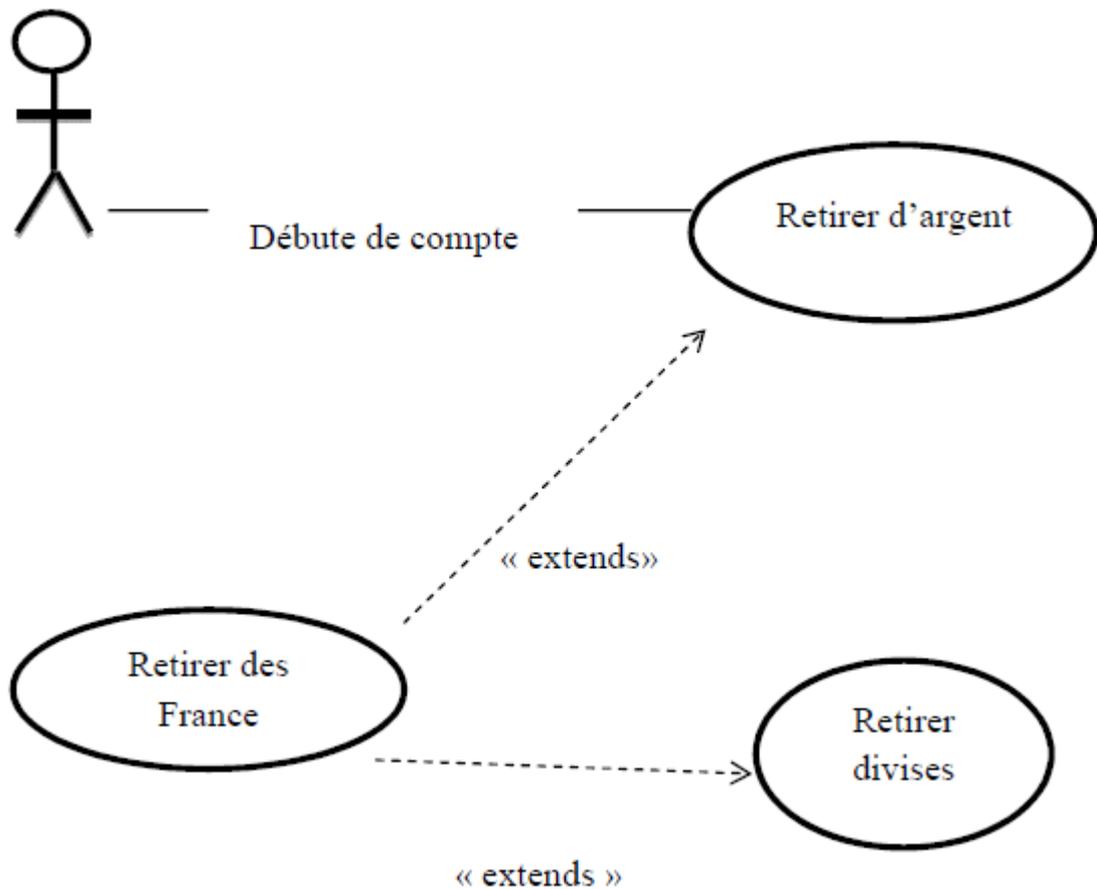
- Permettent d'identifier les fonctionnalités principales (critiques) du système.
 - Le modèle conceptuel doit permettre une meilleure compréhension du système.
 - Le modèle conceptuel doit servir d'interface entre tous les acteurs du projet.
 - Les besoins des clients sont des éléments de traçabilité dans un processus intégrant UML.
- ✓ **La relation entre cas utilisation :**
- Il ya deux types de relation possibles entre cas :
- L'inclusion : quand les cas sources comprend le cas destination.
 - L'extension : quand la source ajoute son comportement cas.

Exemple :



- ✓ **Relation d'utilisation :** indiqué que le cas d'utilisation source contient aussi le comportement décrit dans le cas d'utilisation destination.

Exemple :



Le relation extension indiquer que le cas d'utilisation source étend (précise) les objectifs (le comportement) du cas d'utilisation destination

6.2 Diagramme classes :

✓ **Diagramme de classes : sémantique**

- Un diagramme de classes est une collection d'éléments de modélisation statiques (classes, packages...), qui montre la structure d'un modèle.
- Un diagramme de classes fait abstraction des aspects dynamiques et temporels.
- Pour un modèle complexe, plusieurs diagrammes de classes complémentaires doivent être construits.

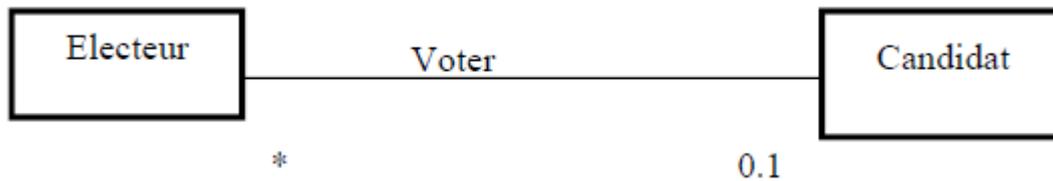
On peut par exemple se focaliser sur :

- Les classes qui participent à un cas d'utilisation (cf. collaboration),
- Les classes associées dans la réalisation d'un scénario précis,
- Les classes qui composent un package,
- La structure hiérarchique d'un ensemble de classes.

Les diagramme de classes permettent de représenter l'ensemble des informations

Annexe A Langage de modélisation UML formalisés, qui sont regrouper des classes.

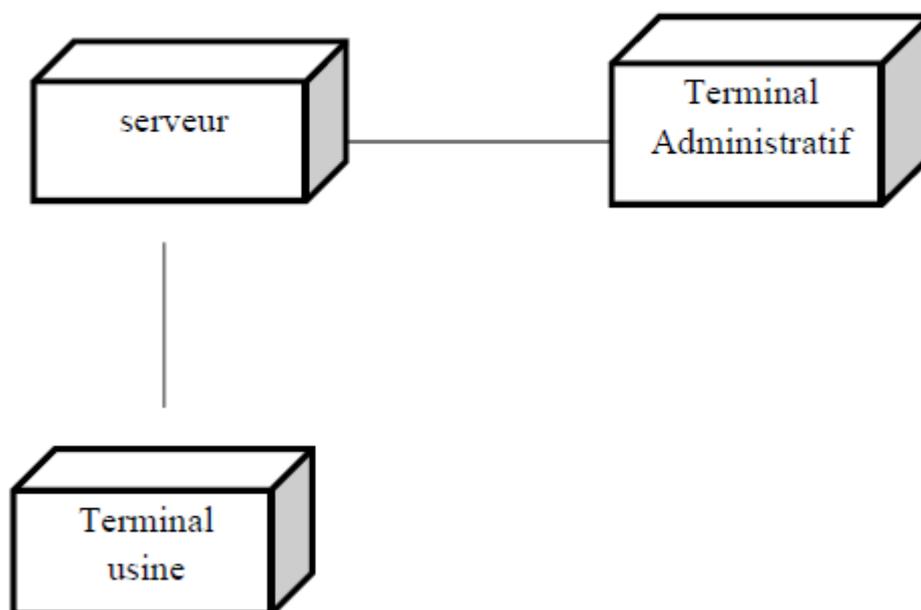
Exemple :



6.3. Diagramme de déploiement :

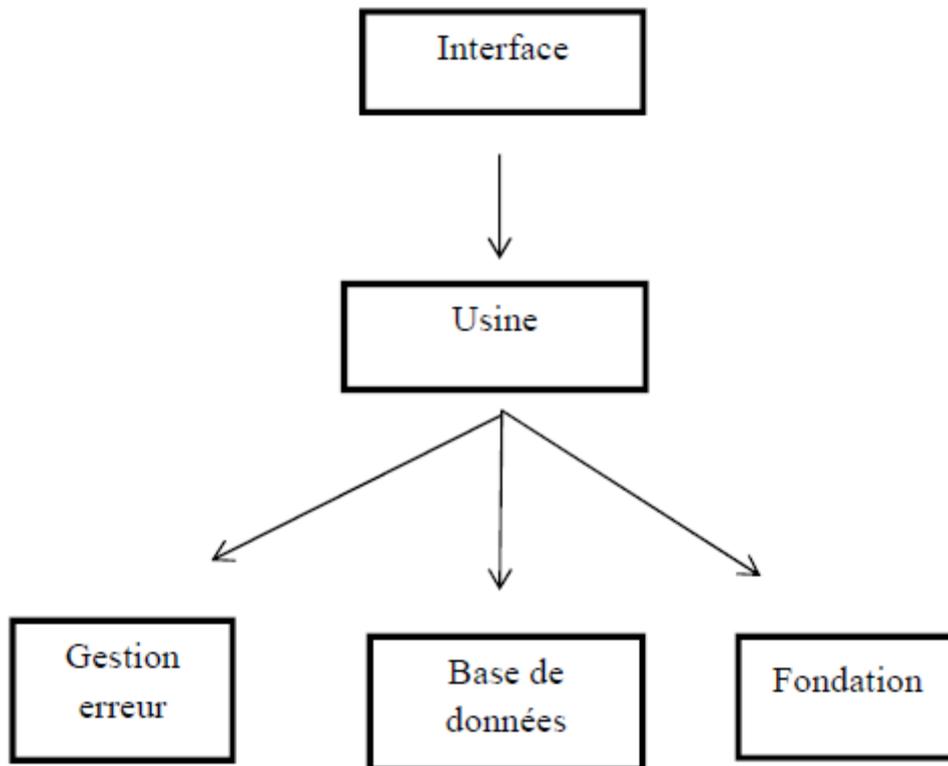
- Les diagrammes de déploiement montrent la disposition physique des matériels qui composent le système et la répartition des composants sur ces matériels.
- Les ressources matérielles sont représentées sous forme de noeuds.
- Les noeuds sont connectés entre eux, à l'aide d'un support de communication. La nature des lignes de communication et leurs caractéristiques peuvent être précisées.
- Les diagrammes de déploiement peuvent montrer des instances de noeuds (un matériel précis), ou des classes de noeuds.
- Les diagrammes de déploiement correspondent à la vue de déploiement d'une architecture logicielle .

Exemple :



6.4 Diagramme de composants : Il représente l'organisation et les dépendances des composants. Un composant est soit une (ou plusieurs) classe(s), interface(s) ou collaboration(s).

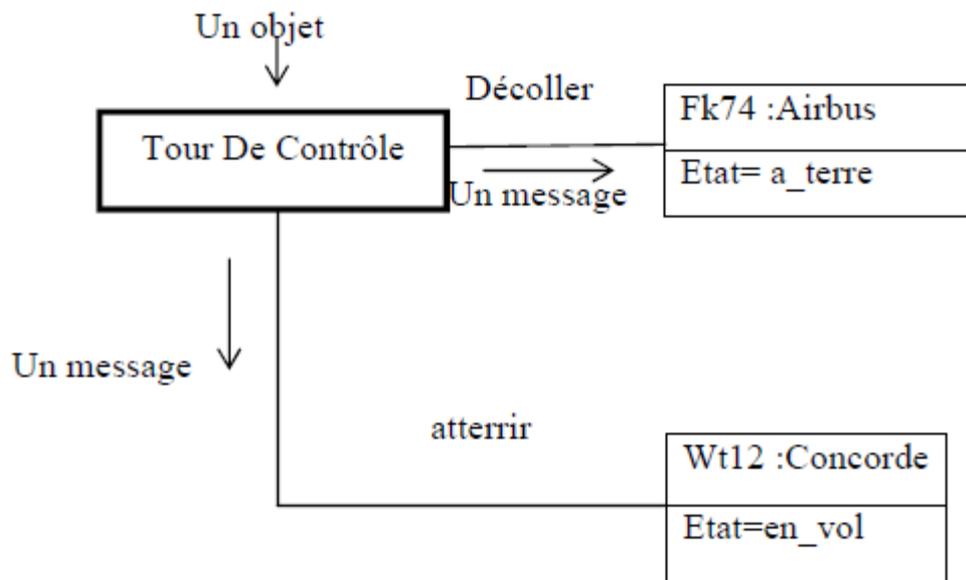
Exemple :



6.5 Diagramme de collaboration

- Les diagrammes de collaboration montrent des interactions entre objets (instances de classes et acteurs).
- Ils permettent de représenter le contexte d'une interaction, car on peut y préciser les états des objets qui interagissent.

Exemple :



6.6 Diagramme de séquence : sémantique

C'est un diagramme d'interaction. Il représente un ensemble d'objets et leurs relations , avec les messages qu'ils échangent. (aspect chronologique des messages).

6.6.1 Les différents types de messages :

- Message simple
- Message minuté
- Message synchrone
- Message asynchrone
- Message déroband

✚ Message simple

Message dont on ne spécifie aucune caractéristique d'envoi ou de réception particulière.

✚ **Message minuté (timeout)**

Bloque l'expéditeur pendant un temps donné (qui peut être spécifié dans une contrainte), en attendant la prise en compte du message par le récepteur. L'expéditeur est libéré si la prise en compte n'a pas eu lieu pendant le délai spécifié.

✚ **Message synchrone**

Bloque l'expéditeur jusqu'à prise en compte du message par le destinataire. Le flot de contrôle passe de l'émetteur au récepteur (l'émetteur devient passif et le récepteur actif) à la prise en compte du message.

✚ **Message asynchrone**

N'interrompt pas l'exécution de l'expéditeur. Le message envoyé peut être pris en compte par le récepteur à tout moment ou ignoré (jamais traité).

✚ **Message déroband**

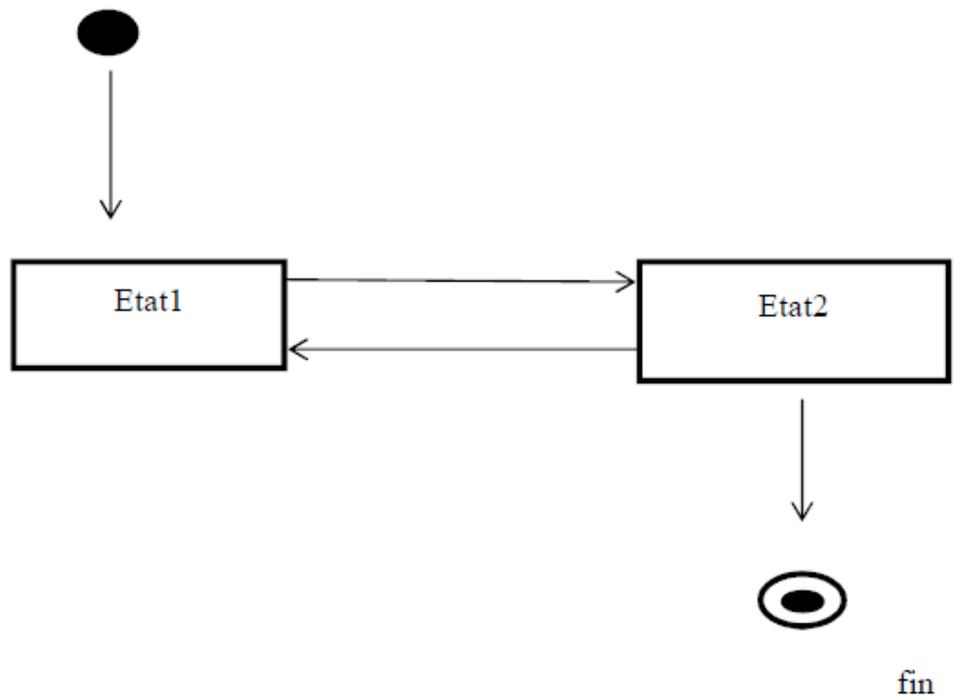
N'interrompt pas l'exécution de l'expéditeur et ne déclenche une opération chez le récepteur que s'il s'est préalablement mis en attente de ce message.

6.7 Diagramme d'états-transitions : sémantique

- Ce diagramme sert à représenter des automates d'états finis, sous forme de graphes d'états, reliés par des arcs orientés qui décrivent les transitions.
- Les diagrammes d'états-transitions permettent de décrire les changements d'états d'un objet ou d'un composant, en réponse aux interactions avec d'autres objets/composants ou avec des acteurs.
- Un état se caractérise par sa durée et sa stabilité, il représente une conjonction instantanée des valeurs des attributs d'un objet.
- Une transition représente le passage instantané d'un état vers un autre.
- Une transition est déclenchée par un événement. En d'autres termes : c'est l'arrivée d'un événement qui conditionne la transition.
- Les transitions peuvent aussi être automatiques, lorsqu'on ne spécifie pas l'événement qui la déclenche.

Exemple :

Début



Conclusion :

L'UML, comme l'on a vu à travers ce chapitre, ne propose pas une démarche objet mais une notation adapté au monde de développement orienté objet. Il nous a donc permis de s'initier aux techniques de modélisation objet. La notation UML peut s'adapter a tous les projets informatiques.

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 UnifiedProcess) :

- 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 de RUP :

2.2.1. Phase d'initialisation :

- 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

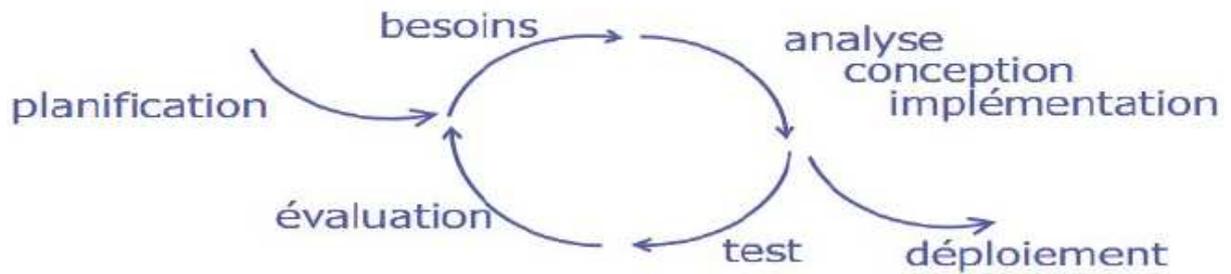
2.2.4. Phase de transition :

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

2.3. Cycle de vie de 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

Cycle de base



Cycle de vie de RUP

2.4. Différence entre incrémental et itératif

✓ RUP est incrémental

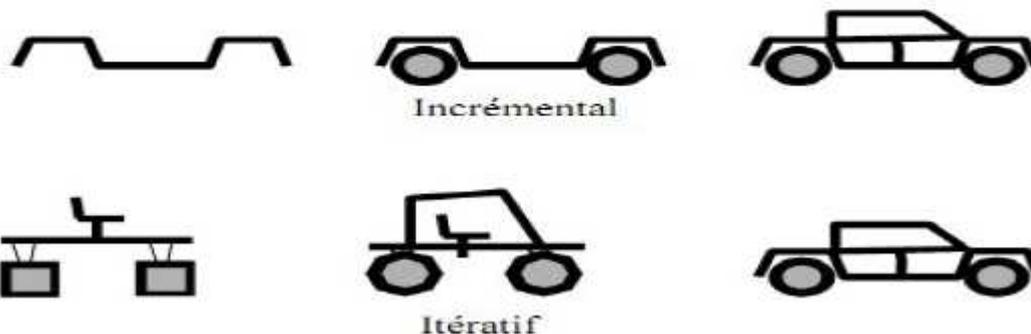
- Le produit final est livré en incréments
 - Chaque incrément livre une fonctionnalité ayant une valeur pour l'utilisateur final
 - Chaque incrément consiste en un ensemble de cas d'utilisation
 - Les incréments doivent être de courte durée (quelques semaines à quelques mois)
 - Les incréments sont 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 :

Différence entre incrémental et itératif



❖ **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

Conclusion :

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

Bibliographie

Bibliographie

- [1] : Java Entreprise Edition (JEE) Développer les applications web avec JEE
Thierry Groussard , François-Xavier Sennesal édition Eni, 14/10/2015
- [2] : Programmer en Java de Claude Delannoy, Eyrolles - 940 pages , 9^e édition,
12 juin 2014.
- [5] : Internet/intranet et les bases de données, G. Gardarin, édition Eyrolles
2000.
- [6] : L'architecture Client/ Serveur. Francois-Marie Colonna édition 2000.
- [7] : Les émetteurs notice ENTP, A.Rezaiguia, Cosob édition 26/06/2015.
- [8] : Introduction à UML de Sinan Si Alhir, édition : O'reilly - 221 pages , 1^{re}
Edition, 1^{er} septembre 2005
- [9] : Introduction au Rational Unified Process, Philippe Kruchten édition
Eyrolles-282 pages, 02/02/2000 .

Webliographie :

- [3]: www.developer.com
- [4] : www.commentcamarch.net.