

**République Algérienne Démocratique et Populaire.**  
**Ministère de L'Enseignement Supérieur et de la Recherche Scientifique.**  
**Université Mouloud Mammeri de Tizi Ouzou.**

*Faculté des Sciences de*  
**Génie Electrique & d'Informatique**



*Mémoire*  
*de fin d'études*  
*de fin d'études*

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

*Option : System Informatique*

**Thème :**

*Conception et réalisation d'une application*  
*Client Serveur pour le service personnel*  
*Cas : NAFTAAL Oued-Aissi*  
*Tizi-Ouzou*

**Proposé et dirigé par :**

*M<sup>r</sup> Kerbiche.M 'hand*

**Réalisé par :**

*M<sup>r</sup> Moukli Aghiles*

*M<sup>r</sup> Mansouri Sofiane*

*Promotion 2013/2014*

**Promotion 2013/2014**

# *Remerciements*

*Quelle meilleure opportunité que ce mémoire qui marque l'achèvement des études universitaires, pour exprimer les plus vifs remerciements à notre promoteur Monsieur Kerbiche M'hani pour nous avoir encadrés et guidés tout au long de notre projet.*

*Nous tenons à remercier chaleureusement le personnel du Service Ressource Humaine de NAFTA.*

*Nos plus vifs remerciements vont aussi à tous les membres de jury pour avoir accepté d'honorer par leur jugement notre travail.*

*Sans oublier nos parents qui ont sacrifié jusqu'aujourd'hui, aussi leurs encouragements incessants tout le long de notre parcours.*

*Ainsi que toute personne qui, de près ou de loin, a contribué à la réalisation de ce travail.*

## *Dédicaces*

*Aghiles*

*A mes très chers parents*

*A mes frères*

*A tous mes amis(es)*

*Sofiane*

*A la mémoire des mes grands- pères*

*A mes grands-mères*

*A mes très chers parents*

*A mes frères & Sœurs*

*A toute la famille*

*A tous mes amis(es)*

# Sommaire

## Chapitre I :Client Serveur

I.1. Introduction :.....	1
I.2. Présentation de l'architecture Client / Serveur :.....	1
I.2.1. Définition :.....	1
Client :.....	1
Serveur : .....	1
Requête .....	1
Client-serveur.....	1
I.2.2. Caractéristiques d'un client, d'un serveur et du client-serveur:.....	1
a. Caractéristiques d'un serveur :.....	1
b. Caractéristiques d'un client :.....	2
c. Caractéristiques du modèle client-serveur :.....	2
I.2.3. Fonctionnement d'un système CLIENT SERVEUR :.....	3
I.2.4. L'Interaction client/serveur :.....	4
I.2.5. Architecture client-serveur :.....	4
I.2.5.1. Présentation de l'architecture à 2 niveaux :.....	4
I.2.5.2. Présentation de l'architecture à 3 niveaux :.....	5
I.2.5.3. Comparaison des deux types d'architecture :.....	6
I.2.5.4 L'architecture multi niveaux :.....	6
I.2.6. Avantages de l'architecture c /s :.....	7
I.2.7. Conception d'une application C/S :.....	7
I.2.8. Les protocoles :.....	7
I.2.9. Mode de communication client-serveur :.....	8
I.2.9.1. Transport fiable en mode connecté :.....	8
I.2.9.2. Transport non fiable en mode non connecté :.....	9
I.2.10. Communications client/serveur :.....	10
I.2.10.1. Les sockets :.....	10
I.2.10.2. Les RPC (Remote Procedure Call):.....	11
I.2.11. Type de serveurs :.....	12
I.2.12. Type Client :.....	13
I.3. Conclusion : .....	13

## **Chapitre II : Organisation d'accueil**

II.1. Introduction :.....	15
II.2. Présentation de l'entreprise :.....	15
II.2.1 .Historique :.....	15
II.2.2.Organisation générale de NAFTAL.....	16
II.2.3. Présentation du district COM Tizi-ouzou : .....	17
II.2.4 Mission et activité de l'organisme d'accueil :.....	17
II.2.5. Organigramme général de l'organisme d'accueil : District Com De Tizi-Ouzou .....	18
II.2.6. Description de l'organigramme général :.....	18
II.2.6.1. Présentation des différents départements : .....	18
II.2.6.2. Présentation du centre Multi Produit (CMP) : .....	22
II.2.6.3. Présentation du centre de stockage et de distribution (CSD) :.....	22
II.2.7. Etude des documents et fiches informatiques:.....	22

## **Chapitre III : Analyse et Conception**

III.1. Introduction :.....	36
III.2. Analyse .....	36
III.2.1. Définition des besoins : .....	36
III.2.2. Identification des acteurs : .....	36
III.2.3. Identification des besoins :.....	36
III.2.4. Diagramme du contexte :.....	38
III.2.5. Identification des espaces : .....	38
III.3. Conception : .....	39
III.3.1.La démarche de conception .....	39
III.3.2. Besoins et cas d'utilisations.....	40
III.3.3. Représentation de cas d'utilisation .....	41
III.3.4. Diagrammes de séquences :.....	43
III.3.5. Diagrammes d'activités : .....	47
III.3.6. Diagramme de classe :.....	52
III.3.6.1. Quelques diagramme de classe du cas d'utilisation :.....	52
III.3.6.2 Diagramme de la classe global :.....	54
III.3.7 conception de la base de données :.....	55
III.3.7.1. Dictionnaire de données : .....	55

III.3.7.2. Modèle relationnel : .....	59
III.3.7.3. Le modèle physique des données : .....	61
III.7. Conclusion : .....	70
<b>Chapitre IV :Realisation</b>	
IV.1. Introduction : .....	71
IV.2. Présentation des outils utilisés : .....	71
IV.2.1. Oracle 10g : .....	71
IV.2.2. NetBeans : .....	72
IV.3. Préparation de l'environnement de travail :.....	73
IV.4. Présentation de quelques interfaces : .....	73
IV.5. Conclusion : .....	80
<b>Anexxe A : Description du langage UML</b>	
1. Introduction :.....	81
2. Présentation d'UML :.....	81
3. modélisation avec UML : .....	81
3.1. Définition d'un modèle.....	81
3.2 La modélisation UML.....	82
4. Eléments d'UML : .....	82
4.1 Les éléments structurels.....	83
4.1.1. Modélisation d'une classe.....	83
4.1.2. Les classes-associations.....	83
4.1.3. Cas d'utilisation.....	84
4.2. Les éléments comportementaux :.....	84
4.2.1. Les interactions : .....	84
4.2.2. Les éléments de regroupement : .....	85
4.2.3. Les éléments d'annotation : .....	85
5. Relations dans UML : .....	86
6. Définition des besoins : .....	87
7. Les acteurs : .....	87
8. Diagramme de contexte : .....	88
9. Conclusion .....	88

## Annexe B: Présentation SGBD Oracle

1. Introduction :	89
2. Définition d'une base de données :	89
3. Définition d'un SGBD :	89
3.1. Caractéristiques d'un SGBD :	89
4. Présentation d'Oracle :	90
4.1. Historique :	90
4.2. Fonctionnalités d'Oracle :	91
4.3. Les composants d'Oracle :	92
4.3.1. Les outils d'administration d'Oracle :	92
4.3.2. Les outils de développement d'Oracle :	92
4.4. SQL (Structured Query Language) :	93
4.4.1. Les objets de base :	93
4.4.1.1. Les objets internes :	93
4.4.1.2. Objets internes aux zones de mémoire :	97
4.4.1.3. Objets externes :	100
4.4.2. Le langage de définition de données (LDD) :	101
4.4.3. Le langage de manipulation de données (LMD) :	102
4.4.4. Le langage PL/SQL :	103
4.4.5. Les outils d'Oracle Forms :	104
4.4.5.1. Le navigateur d'objet « <i>Objet Navigator</i> » :	104
4.4.5.2. L'éditeur de représentation « <i>Layout Editor</i> » :	104
4.4.5.3. La palette des propriétés :	104
4.4.5.4. L'éditeur PL/SQL « <i>PL/SQL Editor</i> » :	104
5. Oracle Entreprise Manager :	105
6. L'interface SQL*Plus :	106
7. L'administration d'Oracle 10g :	107
7.1. Rôle de l'administrateur :	107
7.2. Outils d'administration de données :	107
7.3. Vue globale de SQL*DBA :	107
8. Dictionnaire de données :	108
9. L'informatique distribuées et Oracle 10g :	108
9.1. Oracle Net :	108
10. Le serveur de données :	110

11. Le i d'Oracle 10g (internet) :.....	110
12. Les nouveautés d'oracle 10g :.....	110
13. Conclusion .....	111

## Liste des tables

### Chapitre II : Organisme d'accueil

Tableau.II.1 : Liste des documents .....	23
Tableau.II.2 : Liste des registres .....	23
Tableau.II. 3 : Appel à Candidature.....	24
Tableau.II.4 Appel à Candidature Interne .....	25
Tableau.II.5. Attestation de travail .....	26
Tableau.II.6. Bilan Mensuel : Mutation District. ....	27
Tableau.II.7. Bilan Mensuel : Absentéisme .....	28
Tableau.II.8. Bilan Mensuel : <i>Promotion</i> . ....	29
Tableau.II.9. Bilan Annuel : Actions de Formation réalisées.....	30
Tableau.II.10. Contrat de Travail à Durée Indéterminée .....	32
Tableau.II.11. Contrat de Travail à Durée Déterminée.....	33
Tableau.II.12. Contrat de Formation Fidélité.....	34
Tableau.II.13. Titre de Congé .....	35

### Chapitre III: Analyse et conception

Tableau.III.1 Tableau de décisionnaire de données» .....	59
Tableau.III.2 Table employé .....	61
Tab.III.3 Table utilisateur .....	62
Tableau.III.4 Table enfant .....	62
Tableau.III.5 Table conjoint.....	62
Tableau.III.6 Table fiche d'absence.....	63
Tableau.III.7 Table congé .....	63
Tableau.III.8 Table décision.....	63
Tableau.III.9 Table heure supplémentaire .....	64
Tableau.III.10 Table période d'essai .....	64
Tableau.III.11 Table poste.....	64
Tableau.III.12 Table sanction .....	65
Tableau.III.13 Table formation .....	65
Tableau.III.14 Table contrat travail.....	65
Tableau.III.15 Table diplôme.....	66
Tableau.III.16 Table grade.....	66
Tableau.III.17 Table attestation travail .....	66
Tableau.III.18 Table certificat travail .....	67

Tableau.III.19 Table s'absenter .....	67
Tableau.III.20 Table prendre .....	67
Tableau.III.21 Table occupe.....	67
Tableau.III.22 Table mutation .....	68
Tableau.III.23 Table promotion .....	68
Tableau.III.24 Table réintégration .....	68
Tableau.III.25 Table posséder.....	68
Tableau.III.26 Table classe .....	69
Tableau.III.27 Table service.....	69
Tableau.III.28 Table département .....	69
Tableau.III.29 Table unité .....	69
Tableau.III.30 Table apprentis.....	70
Tableau.III.31 Table cattg_socioprof.....	70

#### Annexe B

Tableau.B1. langage de définition de données (LDD) .....	102
Tableau.B2. langage de manipulation de données (LMD).....	102

## Liste de figures

### Chapitre I : Client Serveur

Figure I.1. : Fonctionnement du client/serveur .....	3
Figure I.2. : Architecture a deux niveaux.....	5
Figure I. 3 : Architecture a trois niveaux.....	5
Figure I. 4 : Architecture multi niveaux. ....	6
Figure I.5 : transport en mode connecté.....	9
Figure I.6: transport en mode non connecté. ....	9
Figure I. 7: fonctionnement sockets.....	10
Figure I. 8 : fonctionnement d'un RCP.....	11

### Chapitre II: Organisme d'accueil

Figure II.1.Schéma de l'organisation générale de NAFTAL.....	16
Figure II. 2. L'organigramme Du DISTRICT COM TIZI-OUZOU.....	18

### Chapitre III: Analyse et conception

Figure III.1. Diagramme du contexte de l'application.....	38
Figure III.2. Démarche de modélisation de l'application.....	40
Figure III. 3. Représentation de cas d'utilisation.....	42
Figure III.4 diagramme de séquence de cas utilisation « Ajouter Contrat de travail» .....	43
Figure III.5 diagramme de séquence de cas utilisateur « Modifier Contrat de travail».....	44
Figure III.6 diagramme de séquence de cas utilisation « Modifier Décision».....	45
Figure III.7 diagramme de séquence de cas utilisation « Ajoute Décision» .....	46
Figure III.8 Diagramme d'activité du cas d'utilisation « Modifier mot de passe » .....	47
Figure III.9. Diagramme d'activité du cas d'utilisation «Ajouter employé» .....	48
Figure III.10. Diagramme d'activité du cas d'utilisation «Ajouter décision».....	49
Figure III.11. Diagramme d'activité du cas d'utilisation «Ajouter congé» .....	50
Figure III.12. Diagramme d'activité du cas d'utilisation «Ajouter fiche d'absences» .....	51
Figure.III.13 Diagramme de classe du cas d'utilisation « modifier mot de passe » .....	52
Figure.III.14 Diagramme de classe du cas d'utilisation ajouter employé » .....	52
Figure.III.15 Diagramme de classe du cas d'utilisation ajouter enfant » .....	53
Figure.III.16 Diagramme de classe du cas d'utilisation « consulter enfant ».....	53
Figure III.17:Diagramme de classe global.....	54

## Chapitre IV: Réalisation

Figure IV.1 : Interface d'Oracle 10 g .....	71
Figure IV.2 : Interface de NetBeans. ....	72
Figure IV.3 : Page d'accueil.....	73
Figure IV.4 : Fenêtre « Identification Administrateur » .....	74
Figure IV.5 : Fenêtre « Identification Utilisateurs » .....	75
Figure IV.6 : Fenêtre « Espace Gestion Section Administratif».....	76
Figure IV.7 : Fenêtre « Employé » .....	77
Figure IV.8 : Fenêtre «Page Fiche d'absence».....	78
Figure IV.9 : Fenêtre « Changer Mot de passe ».....	79

## Annexe A

Figure A1 : Représentation d'une classe UML.....	83
Figure A2 : Classe association.....	84
Figure A3 : Cas d'utilisation .....	84
Figure A4 : Exemple de message .....	85
Figure A5 : Exemple de paquetage.....	85
Figure A6 : Exemple de note.....	85
Figure A7 : Démarche de modélisation .....	87
Figure.A8 : Acteur .....	87
Figure A9 : Diagramme de contexte.....	88

## Annexe B

Figure.B1: Dictionnaire de données d'oracle.....	108
Figure.B2 : Emplacement des fichiers listener.ora et tnsnames.ora.....	109

# *Introduction*

## *Général*

---

# INTRODUCTION GÉNÉRAL

---

La technologie de l'information, a envahi tous les domaines de notre vie quotidienne, surtout aux seins des entreprises économiques, qui doivent s'armer de la plus récente des technologies, Car soumise à une rude et virulente concurrence.

La gestion et la maîtrise de l'information sont devenues des préoccupations de premier ordre.

La mise en œuvre de cette technologie moderne de traitement de l'information s'avère nécessaire, pour assurer une gestion plus fiable, plus rigoureuse, moins fastidieuse, ainsi pour minimiser le risque d'erreurs et d'assurer la disponibilité de l'information à toute éventuelle demande.

Dans toute organisation, la gestion des ressources humaines est une partie déterminante, sachant que de bons employés et de bons équipements contribuent en grande partie à sa réussite.

Actuellement l'entreprise économique qui veut marquer sa présence sur le marché, doit admettre que l'information n'est pas une fin en soi et qu'elle doit être menée en harmonie avec d'autres actions liées pour l'essentiel à la formation de son potentiel humain.

Le travail qui nous a été confié par le District Commercialisation de Tizi ouzou consiste à mettre en œuvre un système d'information pour la gestion des ressources humaines.

Le projet porte sur la mise en œuvre d'un système d'information qui procurera un environnement convivial aux acteurs afin de répondre aux contraintes de fiabilité, efficacité et surtout de disponibilité.

Pour la réalisation de ce système d'information, on a opté pour l'utilisation de l'architecture *Client/serveur* sous *Oracle* qui va nous permettre de réaliser une application déployée sous un réseau LAN.

Notre travail se décompose en 4 chapitres qui sont :

Chapitre 1 : Chapitre1 : Le client / serveur.

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

Chapitre 3 : Analyse et conception

Chapitre 4: Réalisation.

Et se termine par une conclusion générale et deux annexes (A et B) portant sur le langage UML et sur SGBD Oracle.

# *Chapitre 1*



*Client Serveur*

## I.1. Introduction :

Tout utilisateur dans l'entreprise doit pouvoir accéder à toute information utile à sa tâche dès lorsque celle-ci est autorisée par les règles de confidentialité et de sécurité en vigueur. L'accès doit être instantané et doit pouvoir être à partir de n'importe quel poste de travail, en plus l'accès à l'information doit avoir lieu par une interface aussi simple que possible. Pour cela une solution est apportée dans les années 90, qui est le modèle client/serveur et qui met en jeu un processus client qui demande l'exécution d'un service et un processus serveur qui réalise ce service.

## I.2. Présentation de l'architecture Client / Serveur : [1]

### I.2.1. Définition :

#### a. Client :

On appelle logiciel client un programme qui utilise le service offert par un serveur. Le client envoie une requête et reçoit la réponse. Le client peut être raccordé par une liaison temporaire.

#### b. Serveur :

On appelle logiciel serveur un programme qui offre un service sur le réseau. Le serveur accepte des requêtes, les traite et renvoie le résultat au demandeur. Le terme serveur s'applique à la machine sur lequel s'exécute le logiciel serveur. Pour pouvoir offrir ces services en permanence, le serveur doit être sur un site avec accès permanent et s'exécuter en permanence.

#### c. Requête :

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

#### d. CLIENT SERVEUR :

Architecture de réseau dans laquelle toutes les informations sont localisées sur des ordinateurs ayant le rôle de serveur, et accessible à l'ordinateur client.

## I.2.2. Caractéristiques d'un client, d'un serveur et du client-serveur: [2]

### a. Caractéristiques d'un serveur :

- ✓ il est initialement passif (ou esclave, en attente d'une requête) ;
- ✓ il est à l'écoute, prêt à répondre aux requêtes envoyées par des clients ;
- ✓ dès qu'une requête lui parvient, il la traite et envoie une réponse.

### b. Caractéristiques d'un client :

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

Le client et le serveur doivent bien sûr utiliser le même protocole de communication.

Un serveur est généralement capable de servir plusieurs clients simultanément.

### c. Caractéristiques du modèle client-serveur :

On peut essayer de préciser ce que l'on entend par modèle client-serveur, bien qu'il n'y ait pas de consensus sur une définition, en donnant les caractéristiques suivantes :

#### ✓ **Service :**

Le serveur est fournisseur de services. Le client est consommateur de services.

#### ✓ **Protocole :**

C'est toujours le client qui déclenche la demande de service. Le serveur attend passivement les requêtes des clients.

#### ✓ **Partage des ressources :**

Un serveur traite plusieurs clients en même temps et contrôle leurs accès aux ressources.

#### ✓ **Localisation :**

Le logiciel client-serveur masque aux clients la localisation du serveur.

#### ✓ **Hétérogénéité :**

Le logiciel client-serveur est indépendant des plates-formes matérielles et logicielles.

#### ✓ **Redimensionnement :**

Il est possible d'ajouter et de retirer des stations clientes. Il est possible de faire évoluer les serveurs.

#### ✓ **Intégrité :**

Les données du serveur sont gérées sur le serveur de façon centralisée. Les clients restent individuels et indépendants.

#### ✓ **Souplesse et adaptabilité :**

On peut modifier le module serveur sans toucher au module client. La réciproque est vraie. On modifie le module client (en améliorant l'interface, par exemple) sans modifier le module serveur.

**I.2.3. Fonctionnement d'un système CLIENT SERVEUR : [3]**

Dans le modèle client/serveur, il Ya 3 principaux acteurs qui sont : **le serveur, le client et le middleware**. Le dialogue entre un client et un serveur se fait sous forme de requêtes/réponses par le biais du middleware.

- **Client** : c'est une entité (processus, programme, ordinateur...) qui demande l'exécution d'une opération à une autre entité par envoi d'un message contenant le descriptif de l'opération à exécuter et attendant la réponse à cette opération par un message en retour.
- **Serveur** : c'est une entité (processus, programme...) qui accomplit une opération sur demande d'un client et transmet la réponse à ce dernier.
- **Middleware** : ensemble de services logiciels construits au-dessus d'un protocole de transport afin de permettre l'échange de requêtes et des réponses associées entre client et serveur de manière transparente.
  - ✓ **Réponse** : message transmis par un serveur à un client suite à l'exécution d'une opération contenant les paramètres de l'opération.
  - ✓ **Requête** : message transmis par un client à un serveur décrivant l'opération à exécuter pour le compte du client.

Un système client/serveur fonctionne selon le schéma suivant :

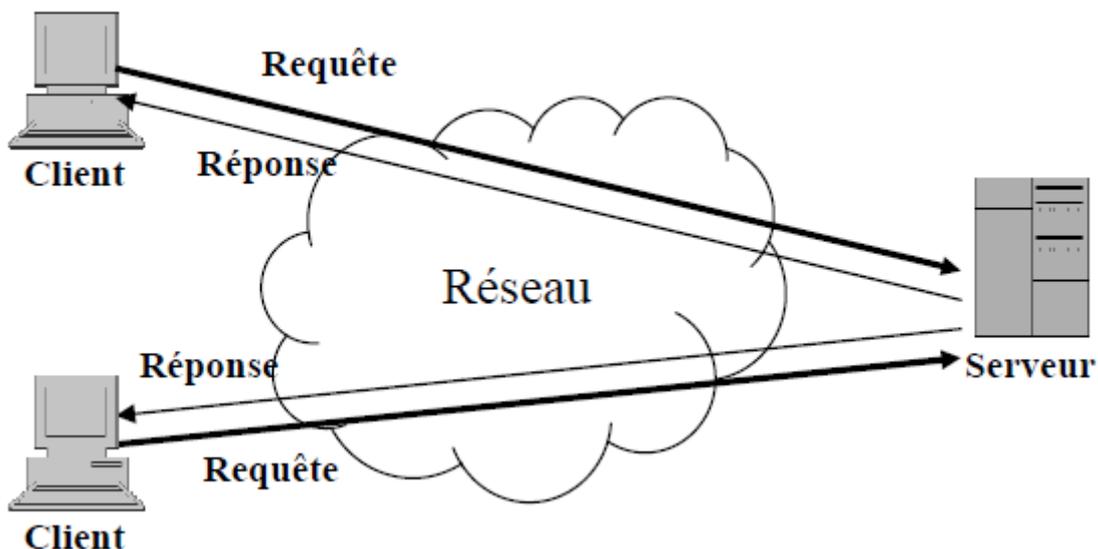


Figure I.1. : Fonctionnement du client/serveur

### I.2.4. L'Interaction client/serveur :

Lorsque le client émet, via le réseau, une requête qui est exécutée par le serveur, ce dernier émet en retour l'ensemble des lignes du résultat.

Dans une telle interaction :

- les machines et les logiciels peuvent être hétérogènes.
- les protocoles réseaux peuvent être distincts.

Mais quand il s'agit d'une architecture comprenant des systèmes hétérogènes, un problème de transparence et de portabilité se pose. C'est pour pallier à ce problème que vient s'ajouter le principe de logiciels intermédiaire (Middleware)

- Le middleware est un ensemble de services logiciels permettant l'échange de requêtes et de réponses associées entre le client et le serveur de manière transparente.

Il assure la médiation entre le (s) client (s) et le (s) serveur (s). Il possède aussi de multiples fonctions telles que l'homogénéisation des formats de requêtes et de données, l'optimisation des accès...etc.

### I.2.5. Architecture client-serveur : [4]

L'architecture client-serveur C'est la description du **fonctionnement coopératif** entre le serveur et le client. Les services Internet sont conçus selon cette architecture. Ainsi, chaque application est composée de logiciel serveur et logiciel client. A un logiciel serveur, peut correspondre plusieurs logiciels clients développés dans différents environnements: Unix, Mac, PC...; la seule obligation est le respect du protocole entre les deux processus communicants. Ce protocole étant décrit dans un RFC (Request For Comment)

#### I.2.5.1. Présentation de l'architecture à 2 niveaux :

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

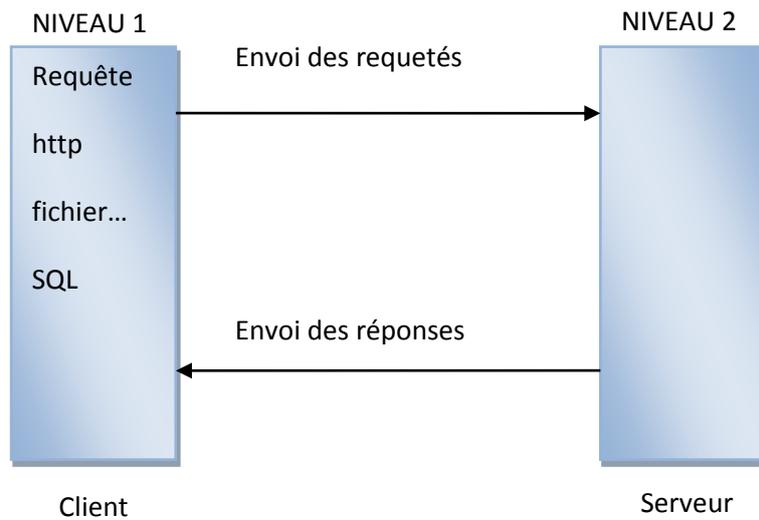


Figure I.2. : Architecture a deux niveaux

**I.2.5.2. Présentation de l'architecture à 3 niveaux :**

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

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

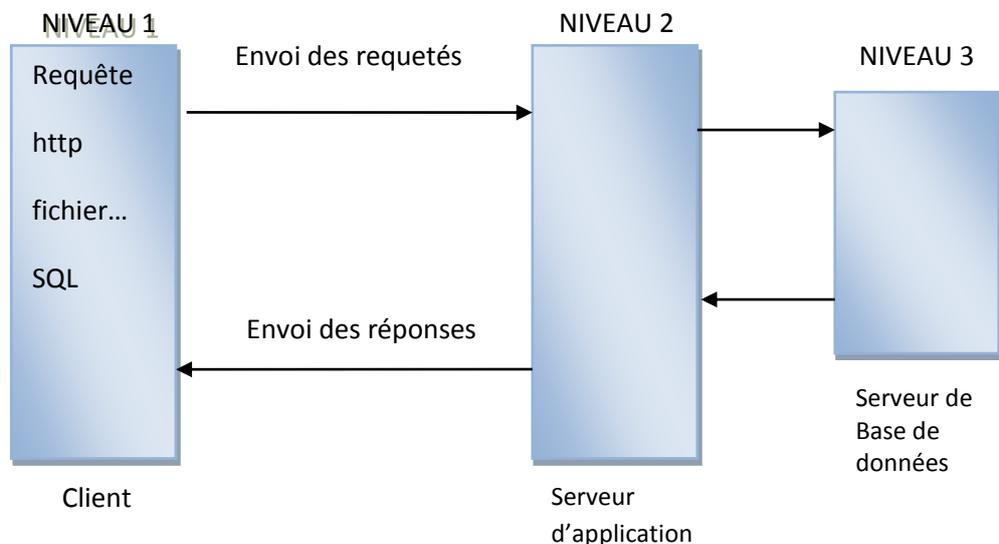


Figure I. 3 : Architecture a trois niveaux.

**I.2.5.3. Comparaison des deux types d'architecture :**

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

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

L'architecture à trois niveaux permet :

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

**I.2.5.4 L'architecture multi niveaux :**

Dans l'architecture à 3 niveaux, chaque serveur (niveaux 2 et 3) effectue une tâche (un service) spécialisée. Un serveur peut donc utiliser les services d'un ou plusieurs autres serveurs afin de fournir son propre service. Par conséquent, l'architecture à trois niveaux est potentiellement une architecture à N niveaux...

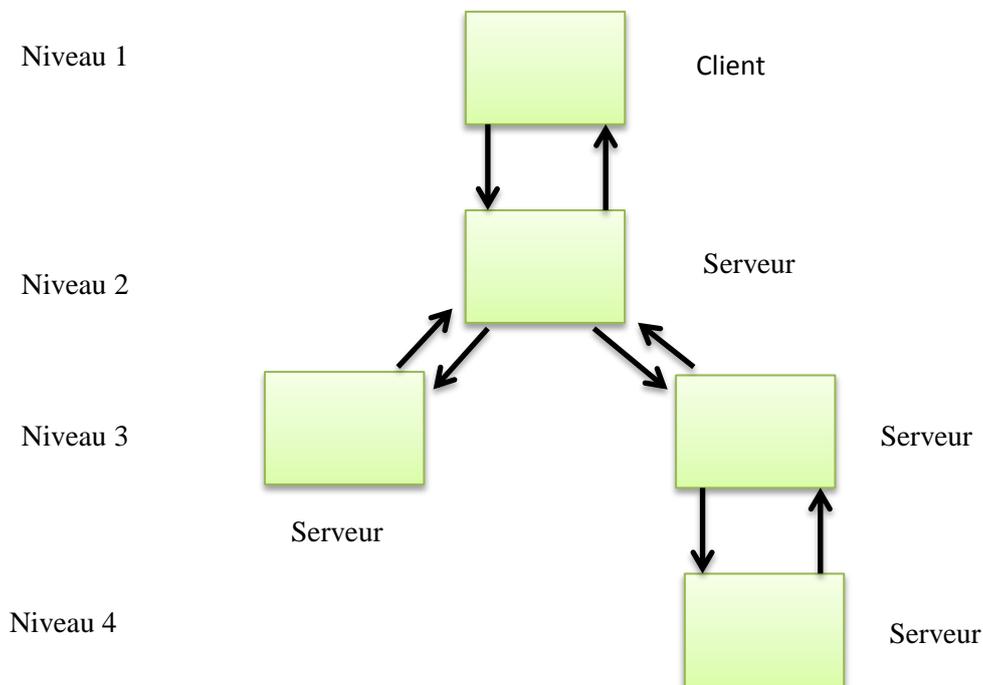


Figure I. 4 : Architecture multi niveaux.

### I.2.6. Avantages de l'architecture c /s :

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

- ✓ **des ressources centralisées** : étant donné que le serveur est au centre du réseau, il peut gérer des ressources communes à tous les utilisateurs, comme par exemple une base de données centralisée, afin d'éviter les problèmes de redondance et de contradiction
- ✓ **une meilleure sécurité** : car le nombre de points d'entrée permettant l'accès aux données est moins important
- ✓ **une administration au niveau serveur** : les clients ayant peu d'importance dans ce modèle, ils ont moins besoin d'être administrés
- ✓ **un réseau évolutif** : grâce à cette architecture, il est possible de supprimer ou rajouter des clients sans perturber le fonctionnement du réseau et sans modification majeure.

### I.2.7. Conception d'une application C/S :

Une application informatique est représentée selon un modèle en trois couches :

- ✓ la couche présentation (interface Homme/Machine) gestion de l'affichage...
- ✓ la couche traitements (ou logique) qui assure la fonctionnalité intrinsèque de l'application (algorithme).
- ✓ la couche données qui assure la gestion des données de l'application (stockage et accès).

### I.2.8. Les protocoles : [5]

Un protocole de communication établit les règles selon lesquelles les ordinateurs communiquent et coopèrent

Un protocole de communication établit le format des requêtes envoyées au serveur, et des réponses de celui-ci.

Le schéma qui suit essaie de montrer où se situent divers protocoles de la pile TCP/IP dans le modèle OSI de l'ISO : un modèle qui décrit les fonctionnalités nécessaires à la communication et l'organisation de ces fonctions.

7	<b>Application</b>	ex. HTTP, HTTPS, SMTP, SNMP, FTP, NFS
6	<b>Présentation</b>	ex. XDR, ASN.1, SMB, AFP
5	<b>Session</b>	ex. ISO 8327 / CCITT X.225, RPC,
4	<b>Transport</b>	ex. TCP, UDP, RTP, SPX, ATP
3	<b>Réseau</b>	ex. IP (IPv4 ou IPv6), ICMP, IGMP, X.25, CLNP, ARP, OSPF, RIP, IPX, DDP
2	<b>Liaison</b>	ex. Ethernet, , PPP, HDLC, Frame relay, RNIS , Bluetooth
1	<b>Physique</b>	ex. techniques de codage du signal (électronique, radio, laser, ...) pour la transmission des informations sur les réseaux

### Protocoles de la couche Application :

Le protocole applicatif définit :

- ✓ le format des messages échangés entre les processus Émetteur et Récepteur.
- ✓ les types de messages : requête, réponse...
- ✓ l'ordre d'envoi des messages.
- ✓ Exemples de protocoles applicatifs :
- ✓ HTTP pour le Web, POP/IMAP/SMTP pour le courrier électronique, SNMP pour l'administration de réseau, ...

### I.2.9. Mode de communication client-serveur : [6]

#### I.2.9.1. Transport fiable en mode connecté :

- ✓ point à point, bidirectionnel : entre deux adresses de transport (@IP src, port src) --> (@IP dest, port dest)
- ✓ transporte un flot d'octets (ou flux)
- ✓ l'application lit/écrit des octets dans un tampon
- ✓ assure la délivrance des données en séquence
- ✓ contrôle la validité des données reçues
- ✓ organise les reprises sur erreur ou sur temporisation
- ✓ réalise le contrôle de flux et le contrôle de congestion.

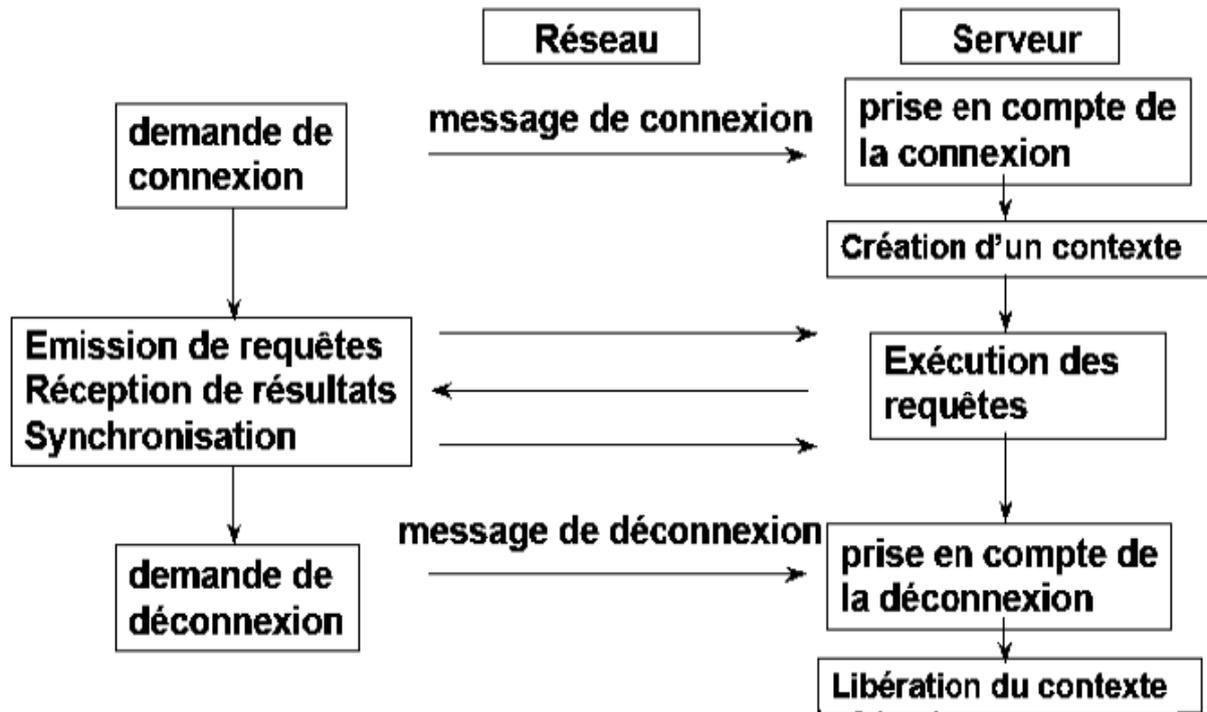


Figure I.5 : transport en mode connecté.

**I.2.9.2. Transport non fiable en mode non connecté :**

- ✓ chaque segment UDP est traité indépendamment des autres
- ✓ les segments UDP peuvent être perdus
- ✓ les segments UDP peuvent arriver dans le désordre

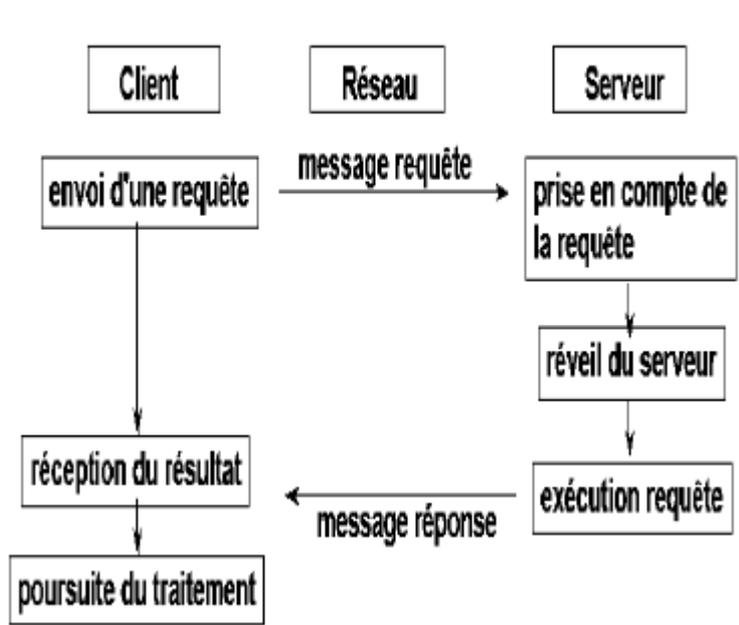


Figure I.6: transport en mode non connecté.

**I.2.10. Communications client/serveur :****I.2.10.1. Les sockets :**

- ✓ Deux processus communiquent en émettant et recevant des données via les sockets.
- ✓ Les sockets sont des portes d'entrées/sorties vers le réseau (la couche transport)
- ✓ Une socket est identifiée par une adresse de transport qui permet d'identifier les processus de l'application concernée
- ✓ Une adresse de transport = un numéro de port (identifie l'application) + une adresse IP (Identifie le serveur ou l'hôte dans le réseau)
- ✓ Le serveur doit utiliser un numéro de port fixe vers lequel les requêtes clientes sont dirigées
- ✓ les clients utilisent un port quelconque entre 1024 et 65535 à condition que le triplet <transport/@IP/port> soit unique
- ✓ ils communiquent leur numéro de port au serveur lors de la requête (à l'établissement de la connexion TCP ou dans les datagrammes UDP)
- ✓ Une socket est un fichier virtuel avec les opérations d'ouverture, fermeture, écriture, lecture, ...

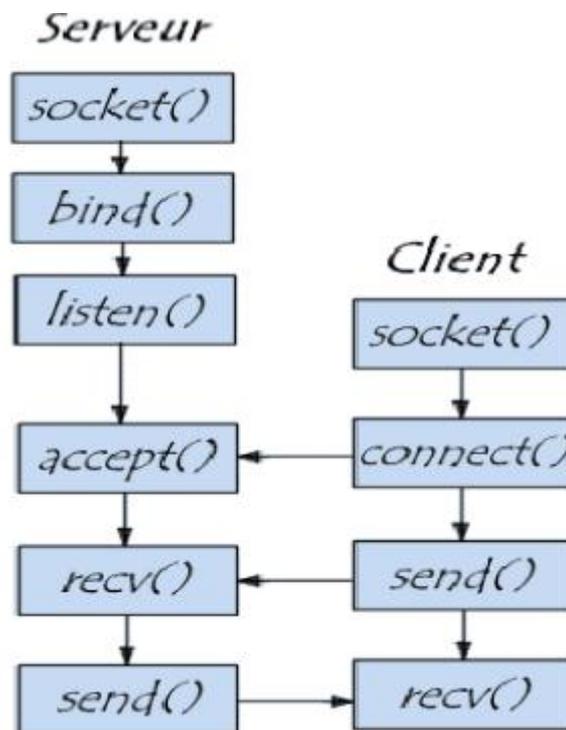


Figure I. 7: fonctionnement sockets.

**I.2.10.2. Les RPC (Remote Procedure Call):**

- ✓ permettre à un processus de faire exécuter une fonction par un autre processus se trouvant sur une machine distante
- ✓ se traduit par l'envoi d'un message contenant l'identification de la fonction et les paramètres
- ✓ une fois le traitement terminé, un message retourne le résultat de la fonction à l'appelant
- ✓ L'objectif des RPC est de faire en sorte qu'un appel distant ressemble le plus possible à un appel local
- ✓ Le processus client (l'appelant) est lié à une petite procédure de bibliothèque, appelée stub client, qui représente la procédure du serveur dans l'espace d'adressage du client
- ✓ Le processus serveur (l'exécutant) est lié à un stub serveur qui représente l'exécution du client
- ✓ Dissimule le fait que l'appel de la procédure n'est pas local : le programmeur de l'application utilise un appel de procédure "normal"

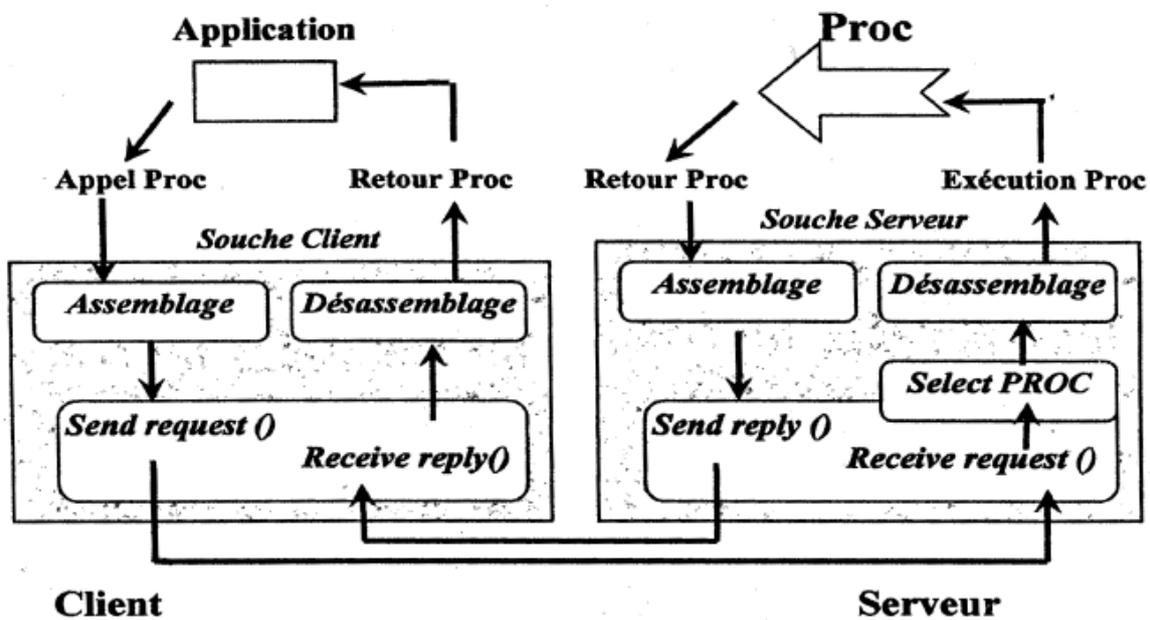


Figure I. 8 : fonctionnement d'un RCP.

### Le protocole RPC :

- ✓ Il doit définir le format du call (message du client vers le serveur), le format des arguments de la procédure, le format du reply (résultats).
- ✓ Il doit permettre d'identifier la procédure à exécuter par le serveur quand un call arrive
- ✓ Il doit permettre d'authentifier la demande (problèmes de sécurité)
- ✓ Quelles machines distantes sont autorisées à exécuter la procédure?
- ✓ Quels utilisateurs sont autorisés à exécuter la procédure ?
- ✓ Les clients RPC ne connaissent que l'identifiant du programme RPC distant et le numéro de Procédure.
- ✓ Un processus spécial, le démon portmap maintient une base de données renseignant les associations locales entre numéro de port et programme RPC.

### I.2.11. Type de serveurs :

- ✓ **Le serveur de fichiers** : est utilisé pour le stockage et le partage de fichiers. Les fichiers placés dans les mémoires de masse du serveur peuvent être manipulés simultanément par plusieurs clients ;
- ✓ **le serveur d'impression** : est utilisé comme intermédiaire entre un ensemble de clients et un ensemble d'imprimantes. Chaque client peut envoyer des documents à imprimer aux imprimantes reliées au serveur ;
- ✓ **le serveur de base de données** : est utilisé pour stocker et manipuler des données contenues dans une ou plusieurs bases de données et partagées entre plusieurs clients ;
- ✓ **le serveur de courrier messagerie** : est utilisé pour stocker et transmettre du courrier électronique. Le courrier partant est transmis au serveur du destinataire. Le courrier arrivé est stocké sur le serveur et est consultable par le client ;
- ✓ **le serveur web** : serveur http stocke et manipule les pages d'un site Web et les transmet sur demande au client (voir navigateur web) ;
- ✓ **Le serveur de jeu** : arbitre et suit l'évolution d'un jeu. Il met en communication les différents joueurs d'un jeu en ligne multi-joueurs ;
- ✓ **le serveur d'applications** : est un terme générique pour désigner un serveur qui effectue les traitements d'un ou plusieurs logiciels applicatif à architecture client/serveur ;
- ✓ **Le serveur de nom** : permet d'établir la correspondance entre le nom de domaine et l'adresse IP des machines d'un réseau.

- ✓ **le serveur mandataire** : reçoit des demandes, les contrôle, puis les transmet à d'autres serveurs.

### I.2.12. Type Client :

#### I.2.12.1. Par usage : on a

- ✓ **Un client de messagerie** : un logiciel qui sert à lire et envoyer des courriers électroniques.
- ✓ **Un client http (navigateur web)** : un logiciel conçu pour se connecter à un serveur http. Dans le cas le plus courant, il s'agit d'un navigateur Web.
- ✓ **Client IRC (chat)** : un logiciel pour se connecter à un serveur irc et dialoguer avec d'autres internautes.
- ✓ **Client SSH (Secure Shell)** : Il est à la fois un programme informatique et un protocole de communication sécurisé.
- ✓ **Client Jabber/XMPP (messagerie instantanée)** : C'est un protocole basé sur une architecture client/serveur permettant les échanges décentralisés de messages instantanés ou non, entre clients, au format XML.

#### I.2.12.2. Par ressource :

- ✓ **Un client léger** : c'est un ordinateur qui, dans une architecture client-serveur, n'a presque pas de logique d'application. Il dépend donc surtout du serveur central pour le traitement
- ✓ **Un client lourd** : c'est un logiciel qui propose des fonctionnalités complexes avec un traitement autonome. La notion de client s'entend dans une architecture client-serveur. Et contrairement au client léger, le client lourd ne dépend du serveur que pour l'échange des données dont il prend généralement en charge l'intégralité du traitement. Les solutions client lourd sont également caractérisées comme étant des solutions très coûteuses tant au niveau de la maintenance que du déploiement et de la formation.
- ✓ **Client riche** : applications effectuant une grande partie du traitement en JavaScript, donc côté client.

### I.3. Conclusion :

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

Tout utilisateur dans l'entreprise doit pouvoir accéder à toute information utile à sa tâche dès lors que celle-ci est autorisée par les règles de confidentialité et de sécurité en vigueur.

L'accès doit être instantané et doit être possible à partir de n'importe quel poste de travail.

En plus il doit avoir lieu par une interface aussi simple que possible. Pour cela une solution est apportée dans les années 1990, à travers le modèle client/serveur qui met en jeu un processus client qui demande l'exécution d'un service et processus serveur qui réalise ce service.

Le chapitre qui suit est l'organisme d'accueil de l'Entreprise NAFTAL.

# *Chapitre 2*



*Organisme d'accueil*

# *Chapitre 3*



*Analyse et conception*

**III.1. Introduction :**

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

Dans ce chapitre on a mis en évidence les différents acteurs intervenants dans le système, ainsi leurs besoins durant la phase d'analyse ensuite on arrive à la conception qui est consacrée essentiellement à la réalisation des résultats obtenus pendant d'analyse en utilisant le langage UML (Unified Modeling Language).

**III.2. Analyse****III.2.1. Définition des besoins :**

Notre projet consiste à concevoir et réaliser une application client/serveur à deux niveaux sous ORACLE pour l'automatisation de la gestion des ressources humaines de **NAFTAL**.

Dans le but de maîtriser le suivi des mouvements des documents et dossiers entre les différents services du champ d'étude, l'application doit offrir un environnement interactif et convivial aux différents acteurs de tous les services de la direction des ressources humaines de **NAFTAL**, leur permettant de travailler et de communiquer mutuellement, dans l'objectif de la gestion.

**III.2.2. Identification des acteurs :**

Un acteur représente une entité qui joue un rôle d'interagir directement avec le système. Les différents acteurs de notre système sont :

- ✓ **Administrateur (ADMIN)** : il gère les comptes utilisateur.
- ✓ **Gestionnaire Section Administratif(GSA)** : sa mission principale est d'assurer la gestion administrative du personnel.
- ✓ **Charger d'étude (CHE)** : sa mission principale est de gérer les dossiers relatifs aux allocations familiales.
- ✓ **Cadre d'étude (CE)** : sa mission principale est d'entreprendre toute étude et travaux ayant trait à la fonction ressources humaines à la demande de la hiérarchie.
- ✓ **Chef de service Ressources Humaines (CSRH)** : sa mission principale est de contrôler et diriger les activités de la ressource humaine.

**III.2.3. Identification des besoins :**

- ✓ **Cas d'utilisation relatifs à l'administrateur(ADMIN)** :
  - Gérer des comptes utilisateurs : Ajouter, supprimer, Consulter, Recherche, Modifier
  - Modifier mot de passe.

- ✓ **Gestionnaire Section Administratif(GSA) :**
  - Gérer des attestations de travail: Ajouter, supprimer, Consulter, Recherche, Modifier
  - Gérer des contrats de travail: Ajouter, supprimer, Consulter, Recherche, Modifier
  - Gérer des certificats de travail: Ajouter, supprimer, Consulter, Recherche, Modifier
  - Gérer des employés: Ajouter, supprimer, Consulter, Recherche, Modifier
  - Gérer des heures supplémentaire: Ajouter, supprimer, Consulter, Recherche, Modifier
  - Modifier mot de passe.
- ✓ **Charger d'étude (CHE) :**
  - Gérer des congés: Ajouter, supprimer, Consulter, Recherche, Modifier
  - Gérer des conjoints: Ajouter, supprimer, Consulter, Recherche, Modifier
  - Gérer des enfants: Ajouter, supprimer, Consulter, Recherche, Modifier
  - Modifier mot de passe
- ✓ **Cadre d'étude (CE) :**
  - Gestion des apprentis: Ajouter, supprimer, Consulter, Recherche, Modifier
  - Gérer des fiches d'absences: Ajouter, supprimer, Consulter, Recherche, Modifier
  - Gestion des formations: Ajouter, supprimer, Consulter, Recherche, Modifier
  - Modifier mot de passe
- ✓ **Chef de service Ressource Humains (CSRH)**
  - Gestion des décisions: Ajouter, supprimé, Consulter, Recherche, Modifier
  - Gestion des promotions: Ajouter, supprimé, Consulter, Recherche, Modifier
  - Gestion des réintégrations: Ajouter, supprimé, Consulter, Recherche, Modifier
  - Gestion des mutations: Ajouter, supprimé, Consulter, Recherche, Modifier

- Gestion des sanctions: Ajouter, supprimé, Consulter, Recherche, Modifier
- Gestion des périodes d'essai: Ajouter, supprimé, Consulter, Recherche, Modifier
- Modifier mot de passe
- Vérifier/rechercher les contrats de travail ;
- Consulter/rechercher employés ;
- Consulter/rechercher les fiches d'absences ;
- Vérifier/rechercher les attestations de travail ;
- Vérifier/rechercher les certificats de travail ;
- Consulter/rechercher les congés ;
- Vérifier/rechercher les formations ;

#### III.2.4. Diagramme du contexte :

Le diagramme du contexte est un modèle de conceptuel qui permet d'avoir une vision globale de l'interaction entre le système et le client avec l'environnement extérieure. Il permet aussi de déterminer le champ d'étude.

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

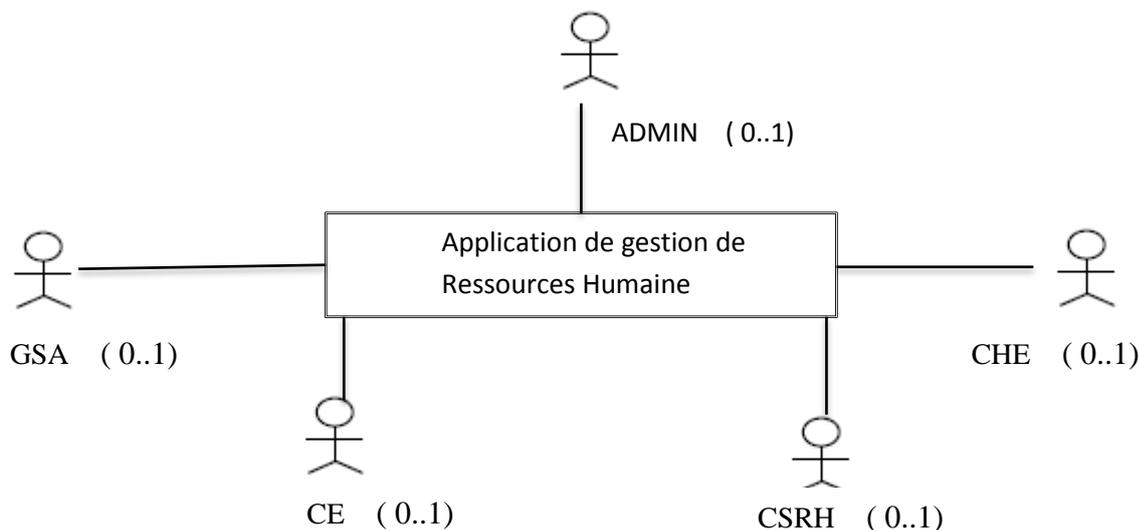


Figure III.1. Diagramme du contexte de l'application

#### III.2.5. Identification des espaces :

A chaque acteur est attribué un espace regroupant toutes les tâches et actions qu'il peut effectuer. Pour notre cas nous avons identifié cinq espaces qui sont :

- ✓ Espace administrateur.

- ✓ Espace gestionnaire section administration
- ✓ Espace chargé d'étude
- ✓ Espace cadre d'étude
- ✓ Espace chef de service ressource humaine

### **III.3. Conception :**

#### **III.3.1. La démarche de conception**

Le processus de conception de notre système comprend deux niveaux :

- ❖ Le Niveau applicatif.
- ❖ Le niveau de données.

##### **III.3.1.1. Le Niveau applicatif**

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

A cet effet, nous avons adopté la démarche suivante :

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

On formalise graphiquement le ou les scénarios qui décrivent chaque cas d'utilisation à l'aide des diagrammes de séquence.

A partir des diagrammes de séquences, on identifie les classes, ensuite on élabore les diagrammes de classes généraux.

##### **III.3.1.2. Le Niveau données**

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

Nous avons adopté la démarche suivante pour la conception de l'application :

- ✓ Identification des acteurs et des besoins.
- ✓ Elaboration du diagramme de contexte du système à étudier.
- ✓ Identification et représentation des cas d'utilisation.
- ✓ Élaboration des diagrammes de séquence.
- ✓ Élaboration des diagrammes d'activités.
- ✓ Élaboration du diagramme de classes.

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

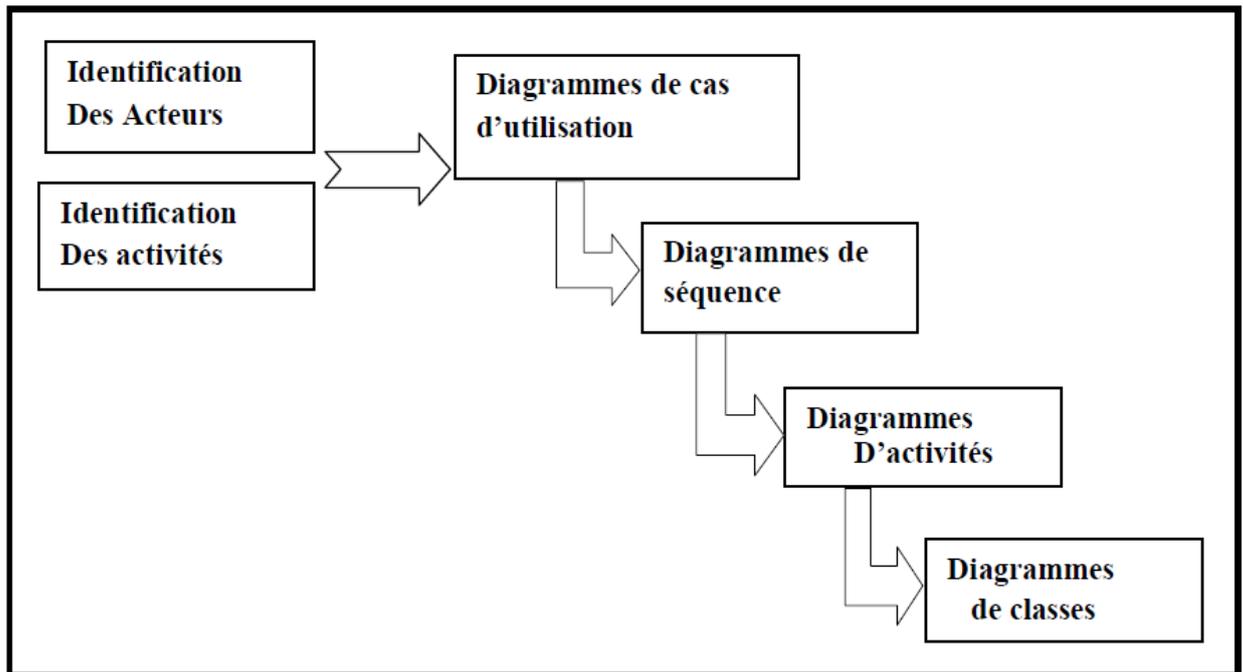


Figure III.2. Démarche de modélisation de l'application.

### III.3.2. Besoins et cas d'utilisations

#### III.3.2.1. Besoins

C'est un énoncé de ce que le système devrait faire (répondre à la question quoi), ces besoins peuvent être fonctionnels ou non fonctionnels.

**Besoins fonctionnels** : expriment une action que doit effectuer le système en réponse à la demande.

**Besoins non fonctionnels** : peuvent être classés dans les catégories suivantes :

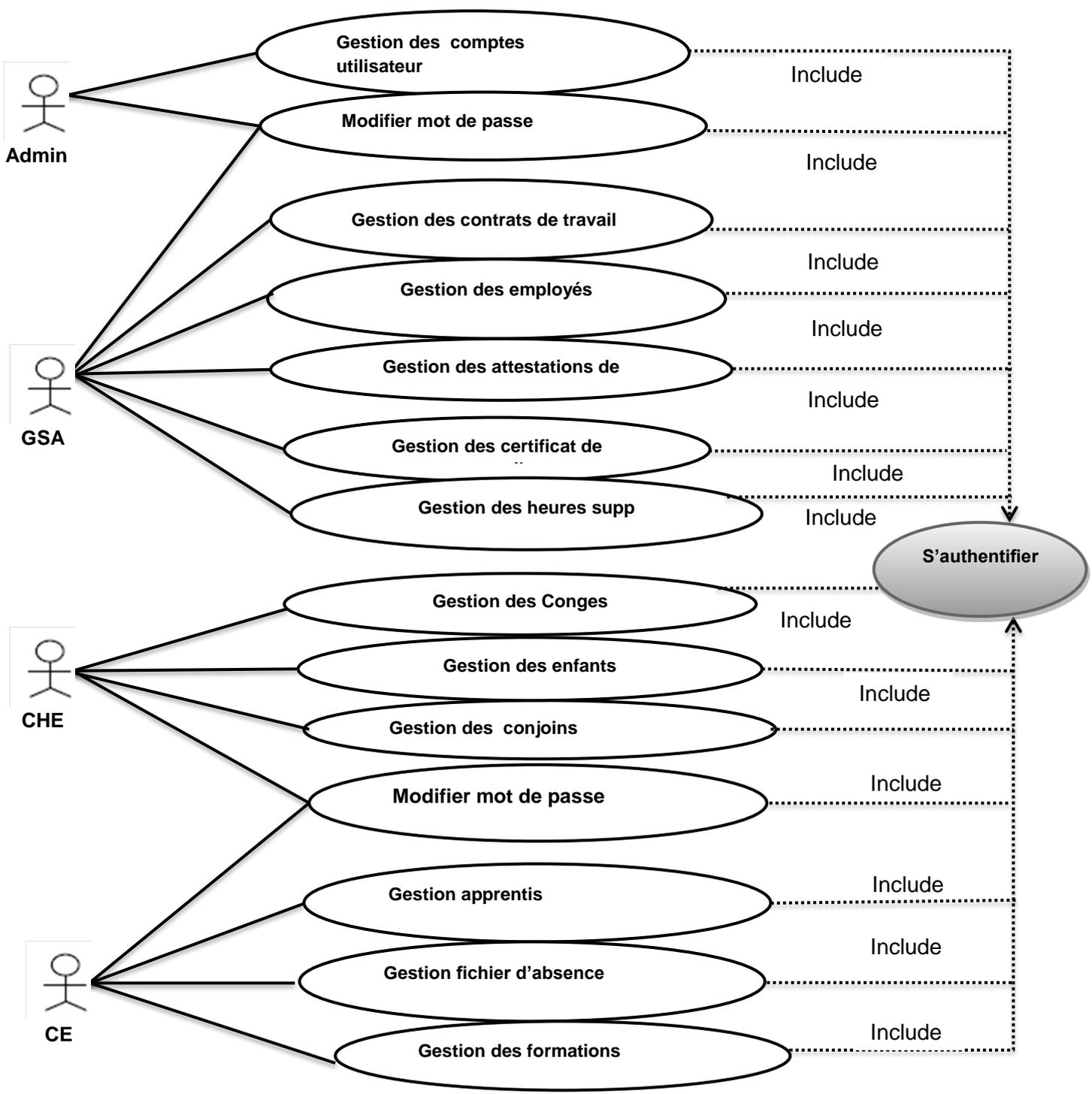
- ✓ Besoins d'utilisabilité.
- ✓ Besoins matériels (configuration, ...).
- ✓ Besoins de performance.
- ✓ Besoins de sécurité.

#### III.3.2.2. Cas d'utilisation

Les cas d'utilisation sont des outils formels qui permettent de consigner et d'exprimer les interactions et les dialogues des utilisateurs (acteurs) avec le système. Ils spécifient ce que le système devrait effectuer sans préjuger de la manière.

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

III.3.3. Représentation de cas d'utilisation:



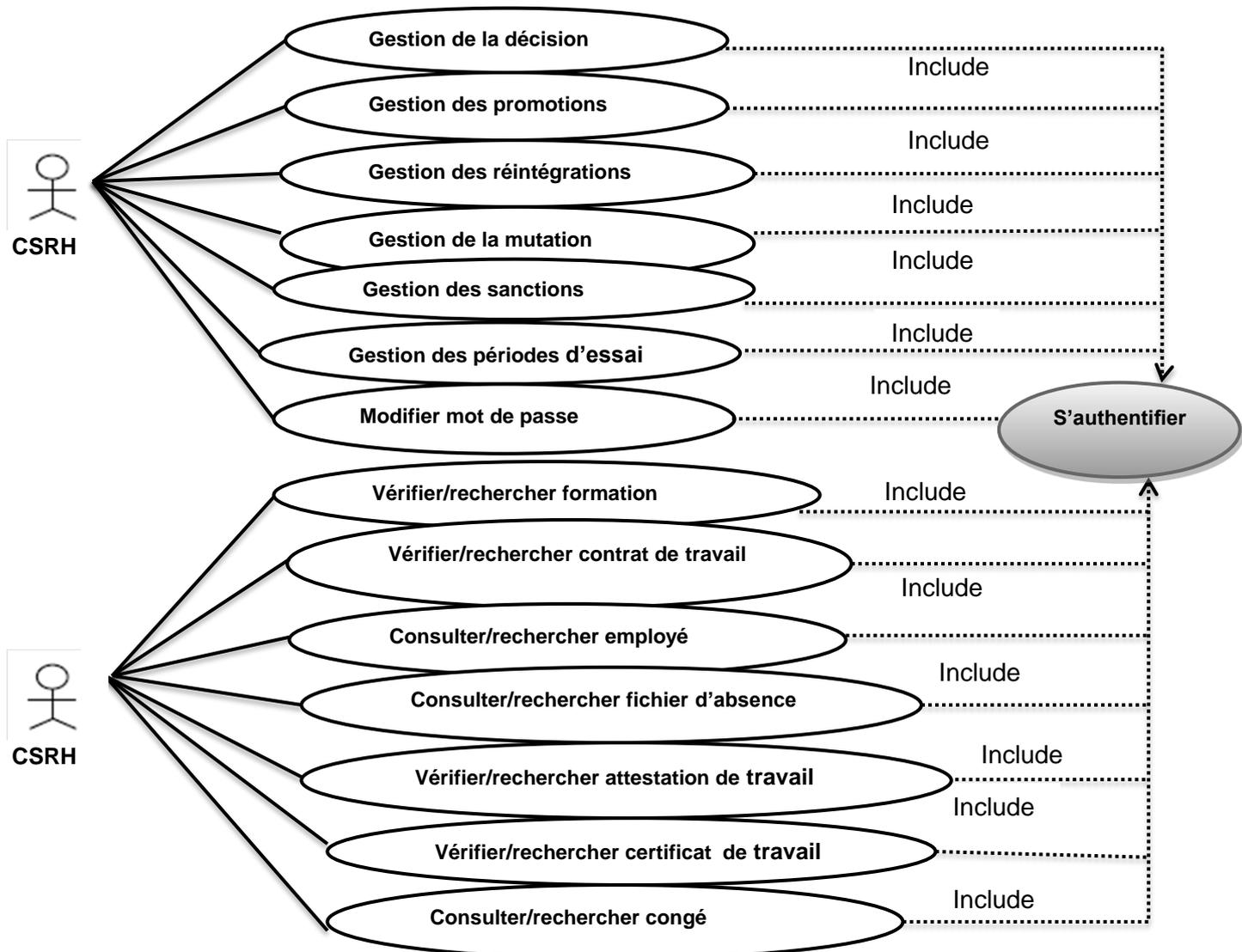


Figure III. 3. Représentation de cas d'utilisation.

## III.3.4. Diagrammes de séquences :

Indique l'interaction entre plusieurs partenaires de communication, également appelés lignes de vie. Les principales informations contenues dans les diagrammes de séquences sont les messages échangés entre les lignes de vie. Un diagramme de séquences met toujours l'accent sur l'ordre chronologique des messages.

Ces diagrammes peuvent être utilisés pour modéliser les responsabilités et les collaborations sans prendre en compte les mécanismes définis par l'architecture du système

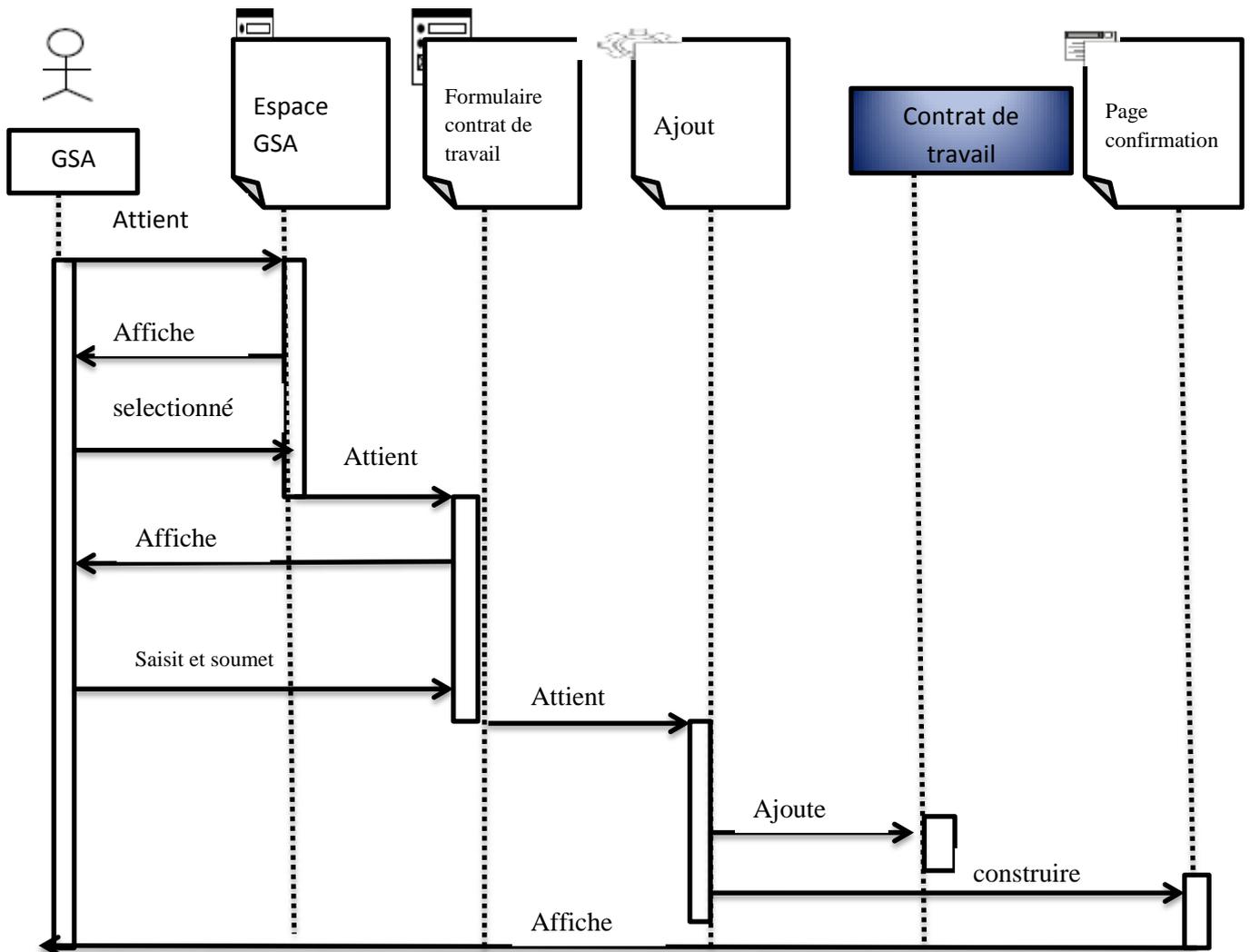


Figure III.4 diagramme de séquence de cas utilisation « Ajouter Contrat de travail »

## Scénario :

- ✓ GSA accède à son espace après identification ;
- ✓ GSA sélectionne le formulaire contrat de travail ;
- ✓ GSA saisit et soumet les informations du formulaire ;
- ✓ Le formulaire est alors ajouté s'il n'existe pas ;
- ✓ Une page de confirmation est ensuite affichée.

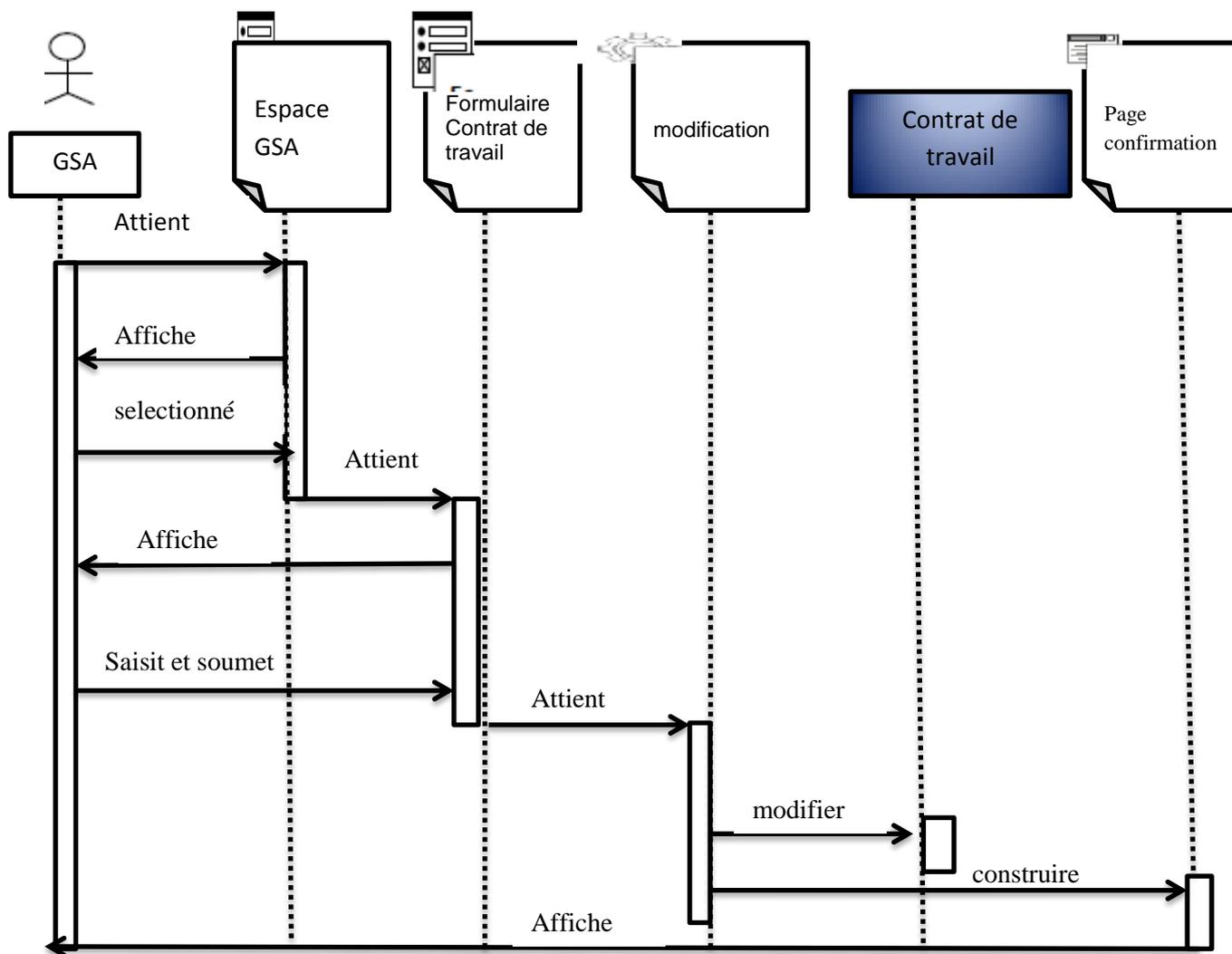


Figure III.5 diagramme de séquence de cas utilisateur « Modifier Contrat de travail »

## Scénario :

- ✓ GSA accède à son espace après identification ;
- ✓ GSA sélectionne le formulaire contrat de travail ;
- ✓ GSA saisit et soumet les informations du formulaire ;
- ✓ Le formulaire est alors modifié après confirmation de la modification ;
- ✓ Une page de confirmation est ensuite affichée.

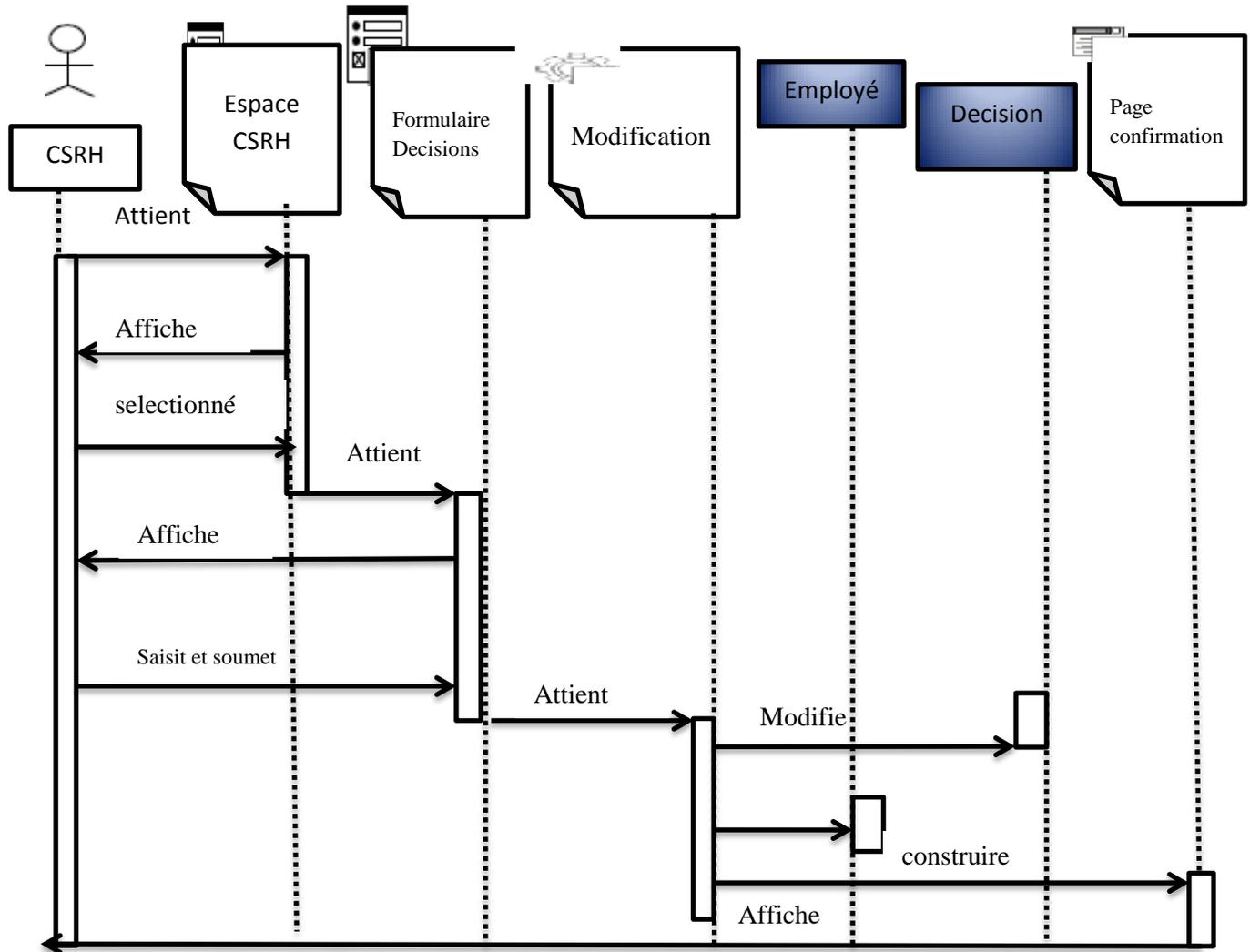


Figure III.6 diagramme de séquence de cas utilisation « Modifier Décision »

## Scénario :

- ✓ CSRH accède a son espace après identification ;
- ✓ CSRH sélectionne le formulaire décision;
- ✓ CSRH saisit et soumet les informations du formulaire ;
- ✓ La décision est alors modifié après confirmation de la modification;
- ✓ L'employé concerné est modifié selon la décision;
- ✓ Une page de confirmation est ensuite affichée.

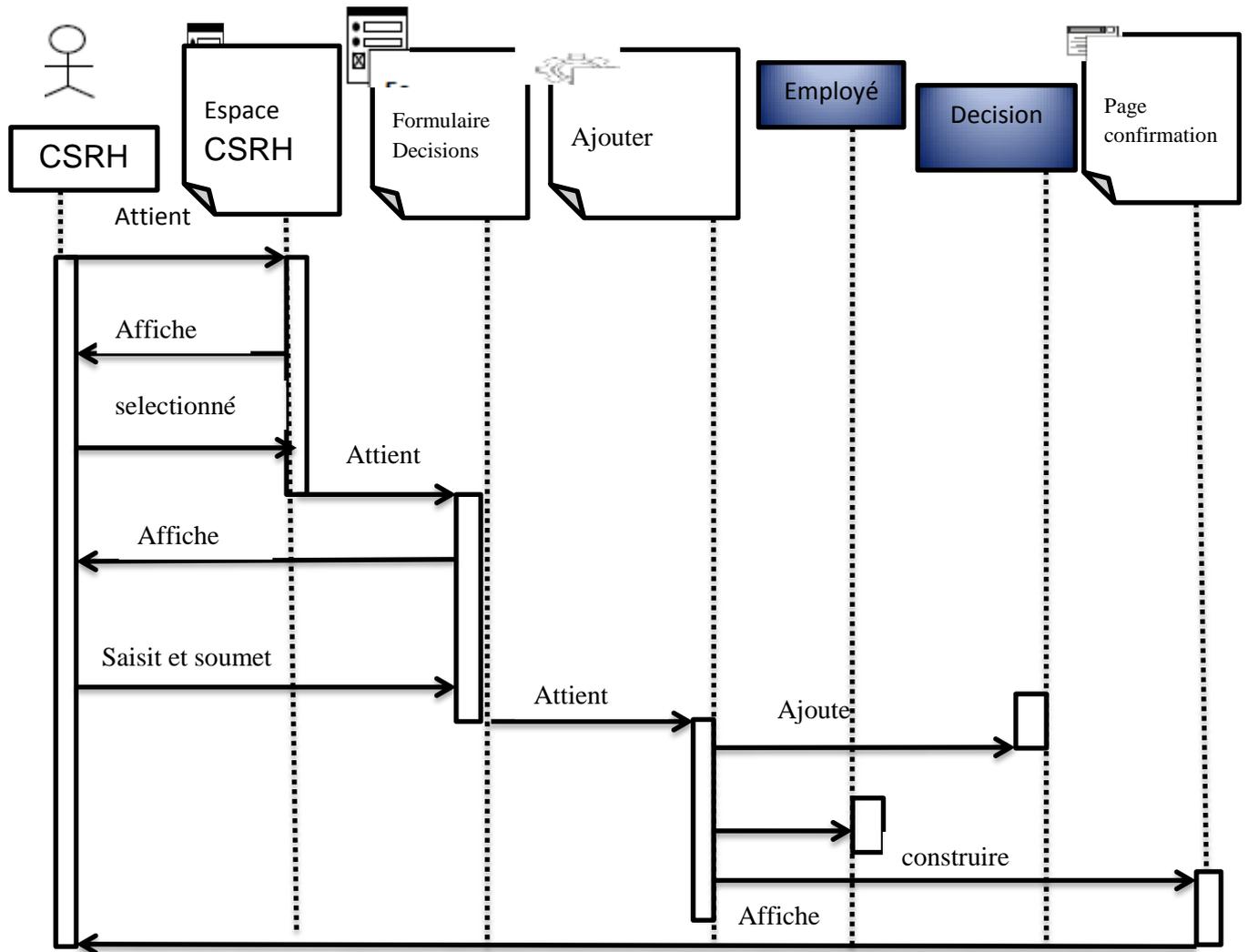


Figure III.7 diagramme de séquence de cas utilisation « Ajoute Décision »

## Scénario :

- ✓ CSRH accède a son espace après identification ;
- ✓ CSRH sélectionne le formulaire décision;
- ✓ CSRH saisit et soumet les informations du formulaire ;
- ✓ La décision est alors ajoutée si elle n'existe pas ;
- ✓ L'employé concerné est modifié selon la décision;
- ✓ Une page de confirmation est ensuite affichée.

## III.3.5. Diagrammes d'activités :

Apporte un point de vue complémentaire à l'aspect dynamique de la modélisation. Il offre un pouvoir d'expression très proche des langages de programmation objets. Il est donc bien adapté à la spécification détaillée des traitements en phase de réalisation. Un diagramme d'activités se concentre plutôt sur les activités entre les objets, c'est-à-dire, il met en évidence l'activité qui a lieu dans le temps, donc les opérations transmises entre les objets.

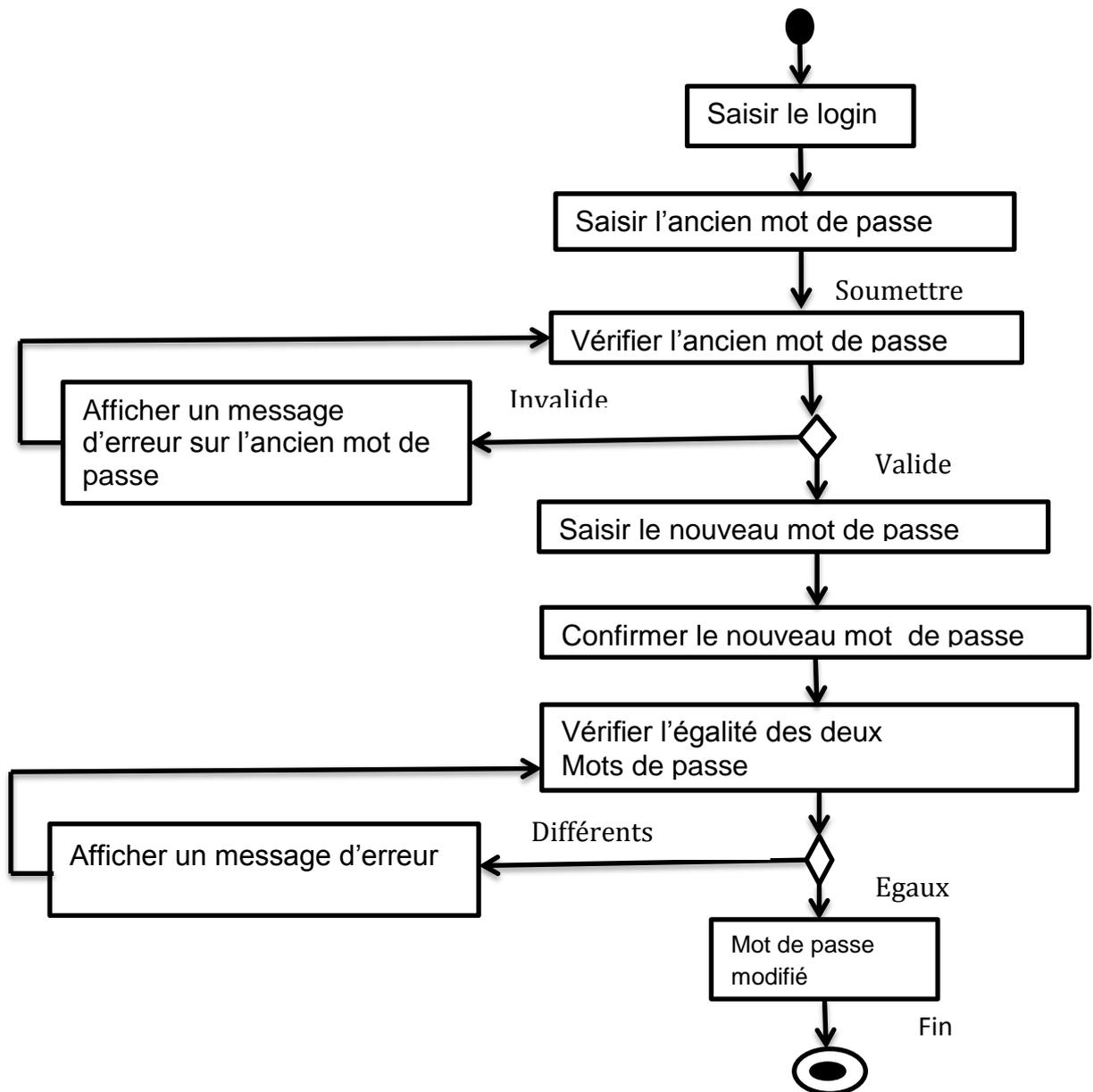


Figure III.8 Diagramme d'activité du cas d'utilisation « Modifier mot de passe »

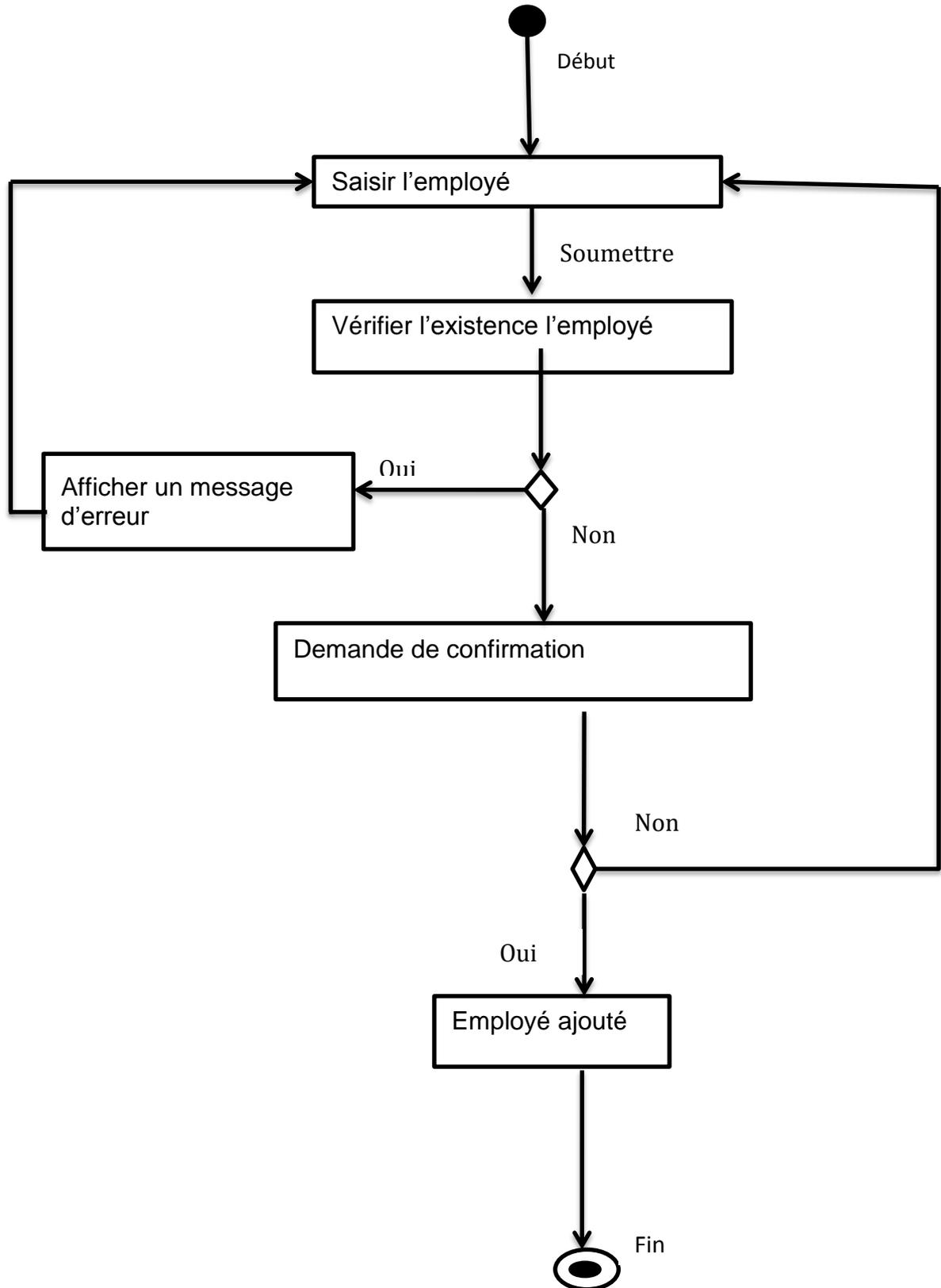


Figure III.9. Diagramme d'activité du cas d'utilisation «Ajouter employé»

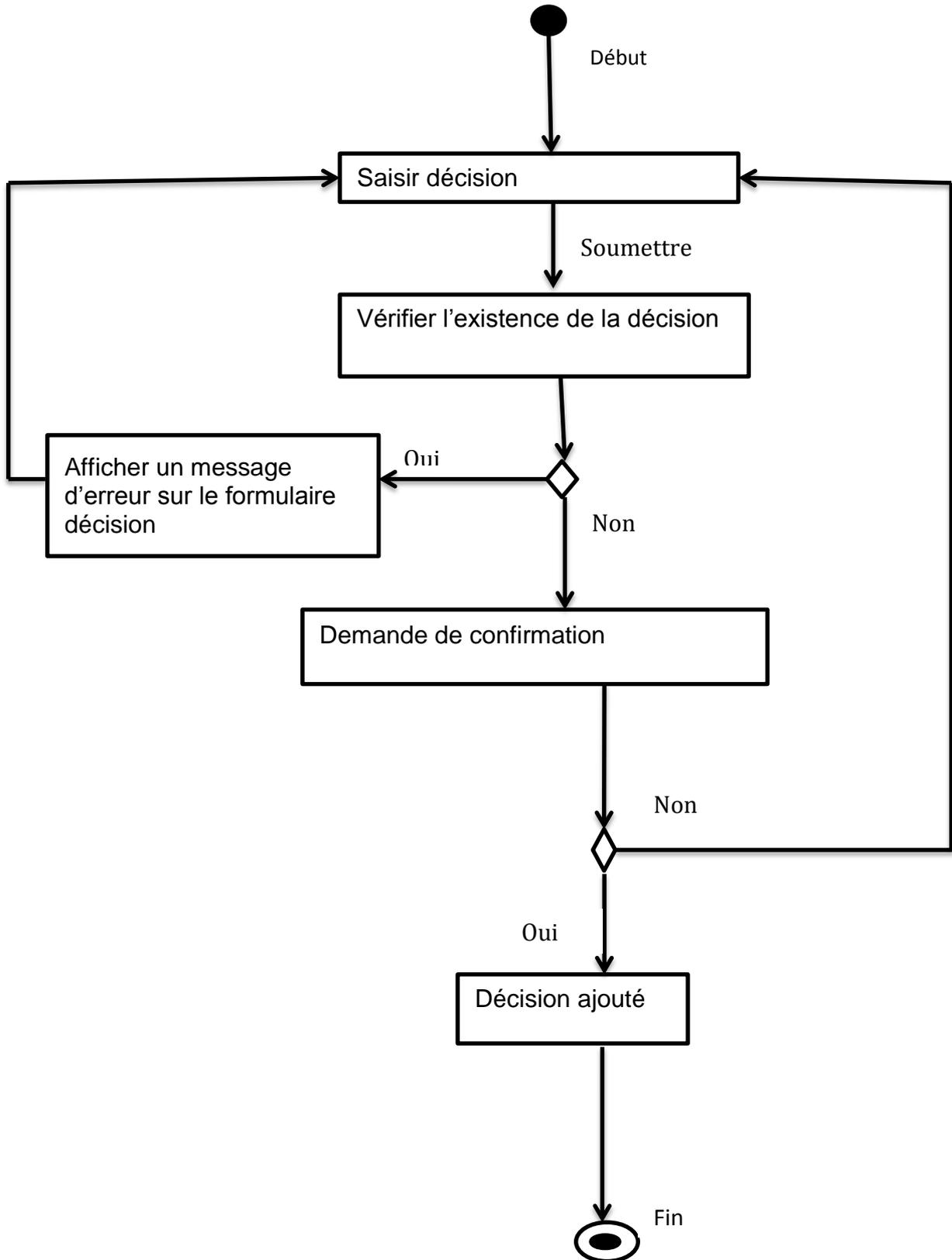


Figure III.10. Diagramme d'activité du cas d'utilisation «Ajouter décision»

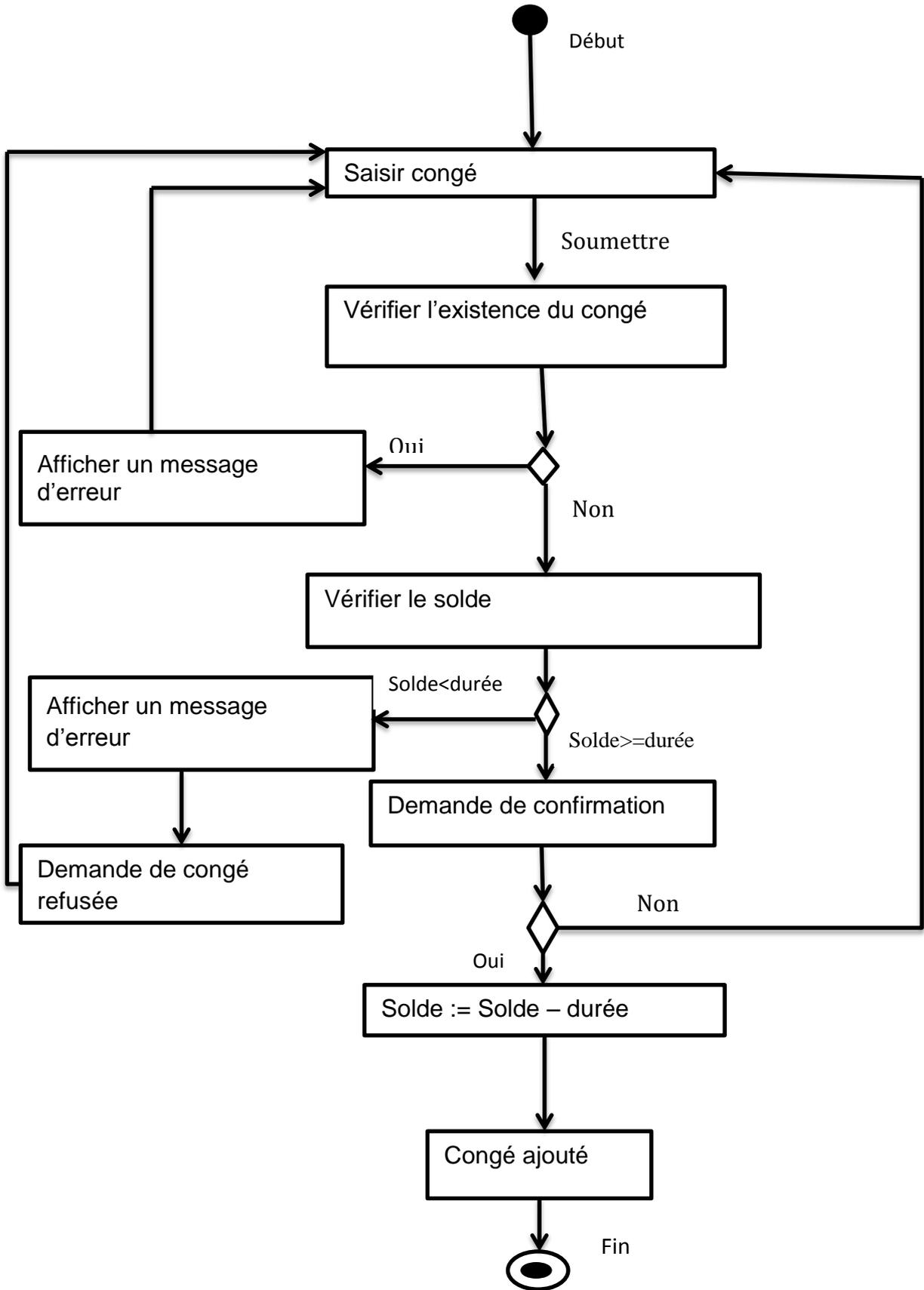


Figure III.11. Diagramme d'activité du cas d'utilisation «Ajouter congé»

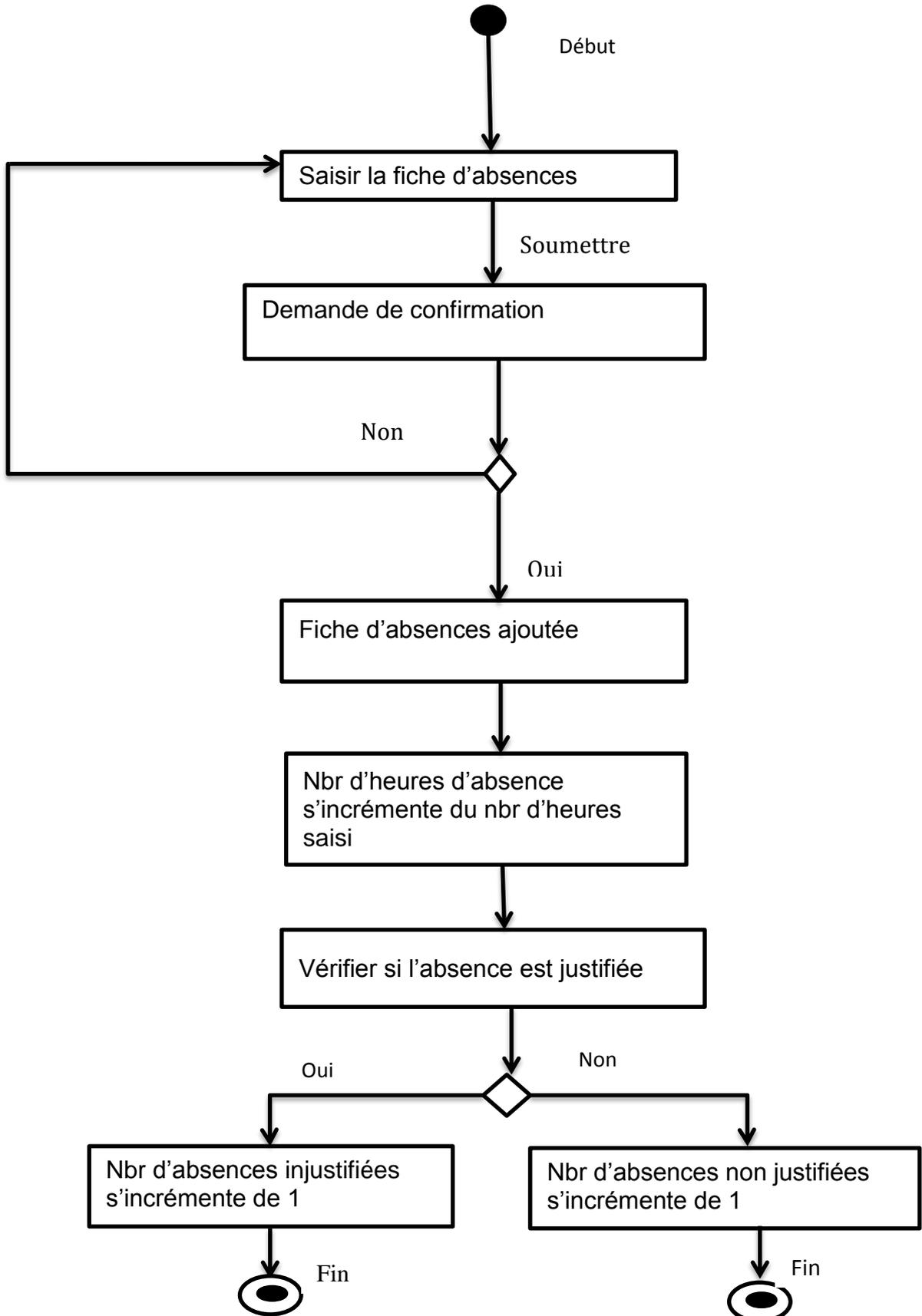


Figure III.12. Diagramme d'activité du cas d'utilisation «Ajouter fiche d'absences»

## III.3.6. Diagramme de classe :

### III.3.6.1. Quelques diagramme de classe du cas d'utilisation :

Les diagrammes de classe sont les plus fréquents dans la modélisation des systèmes orientés objet. Dans ce qui nous nous présenter les diagrammes classe de quelque cas d'utilisation.

#### ✓ Diagramme de classe du cas d'utilisation « modifier mot de passe » :

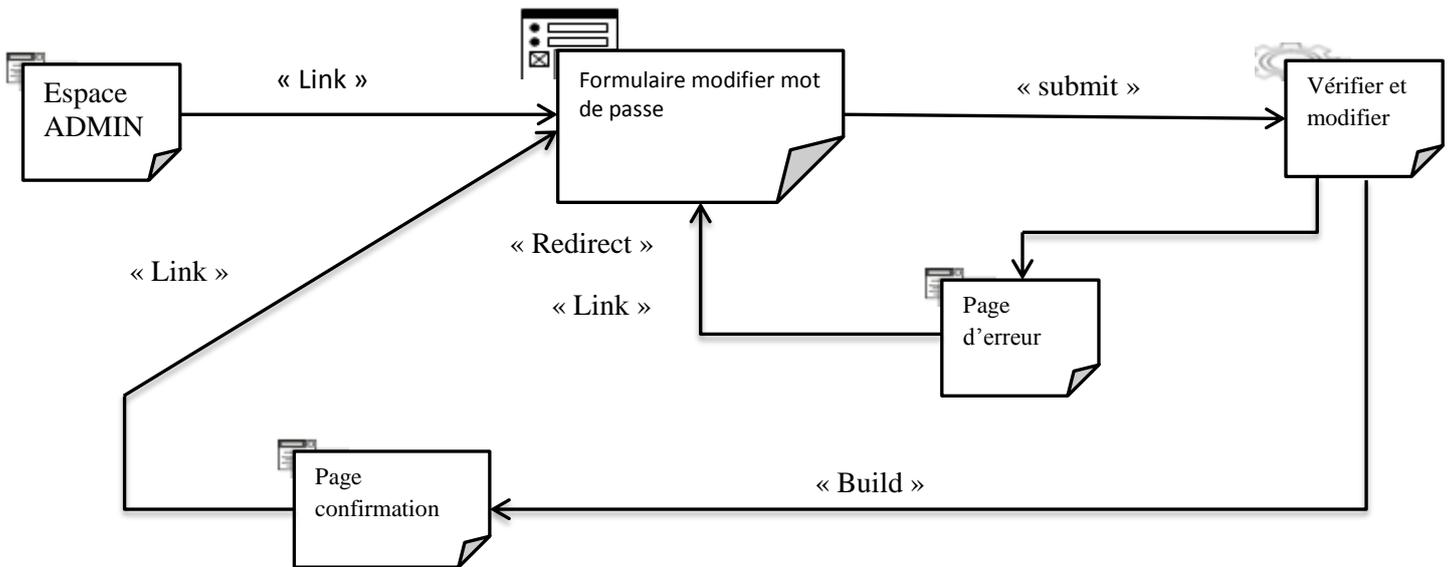


Figure.III.13 Diagramme de classe du cas d'utilisation « modifier mot de passe »

#### ✓ Diagramme de classe du cas d'utilisation « ajouter employé » :

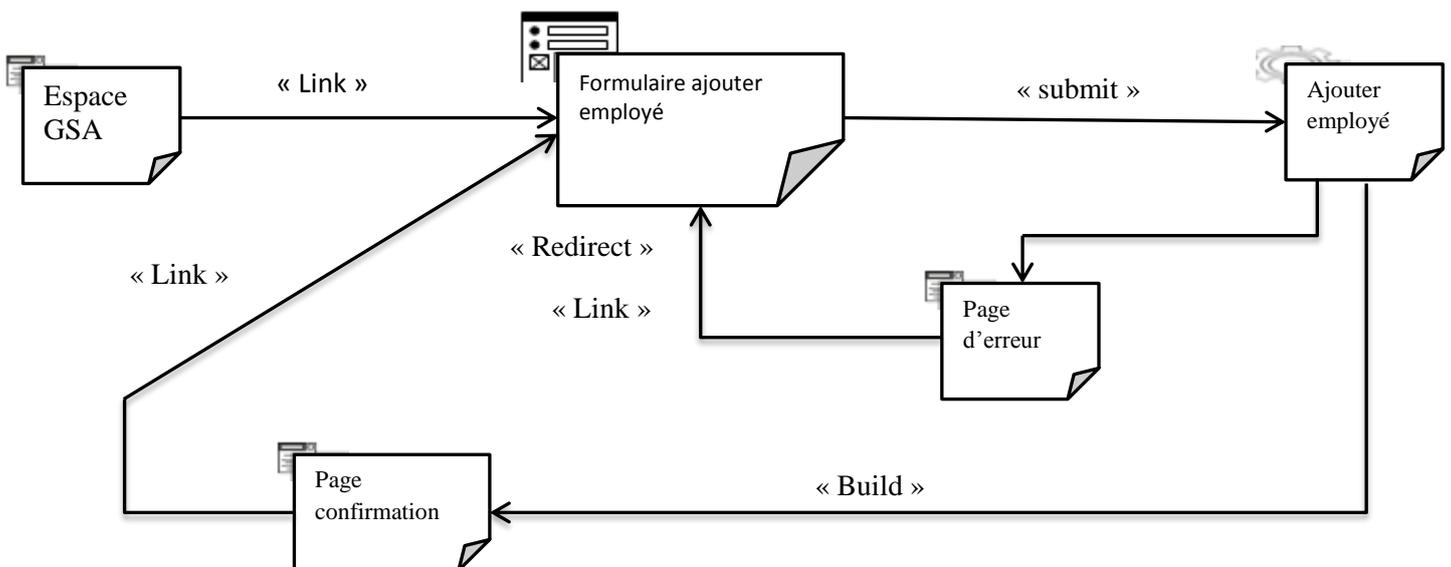


Figure.III.14 Diagramme de classe du cas d'utilisation « ajouté employé »

## ✓ Diagramme de classe du cas d'utilisation « ajouter enfant »

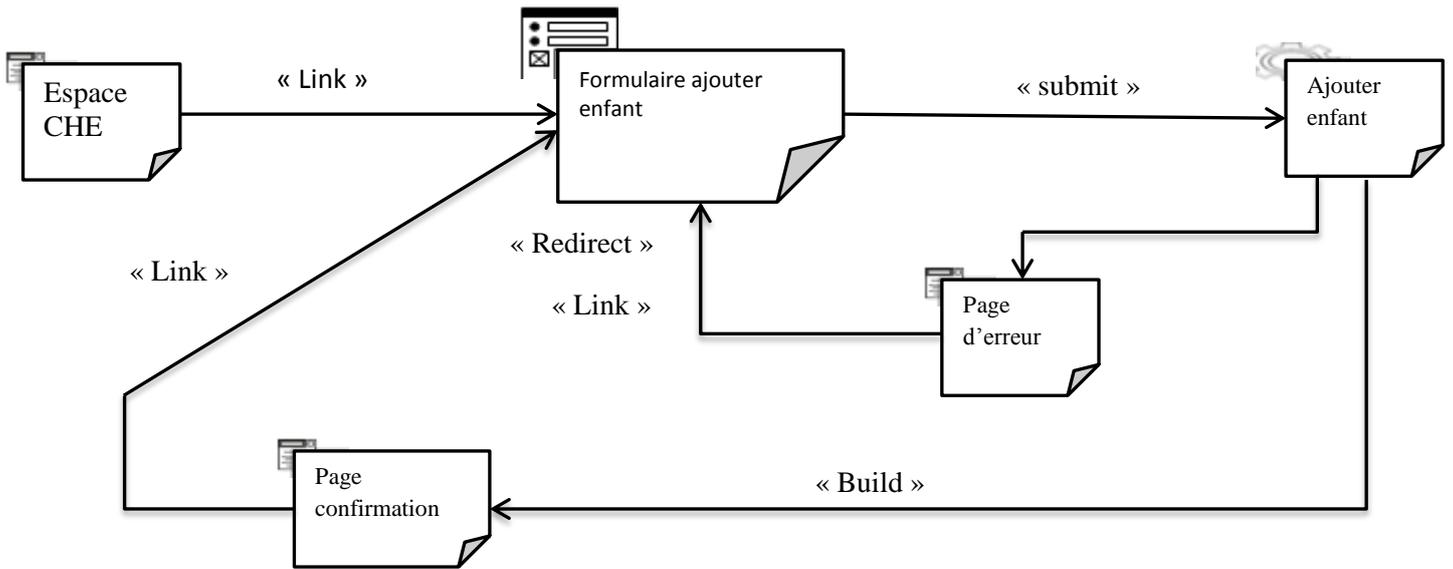


Figure.III.15 Diagramme de classe du cas d'utilisation ajouter enfant »

## ✓ Diagramme de classe du cas d'utilisation « consulter enfant »

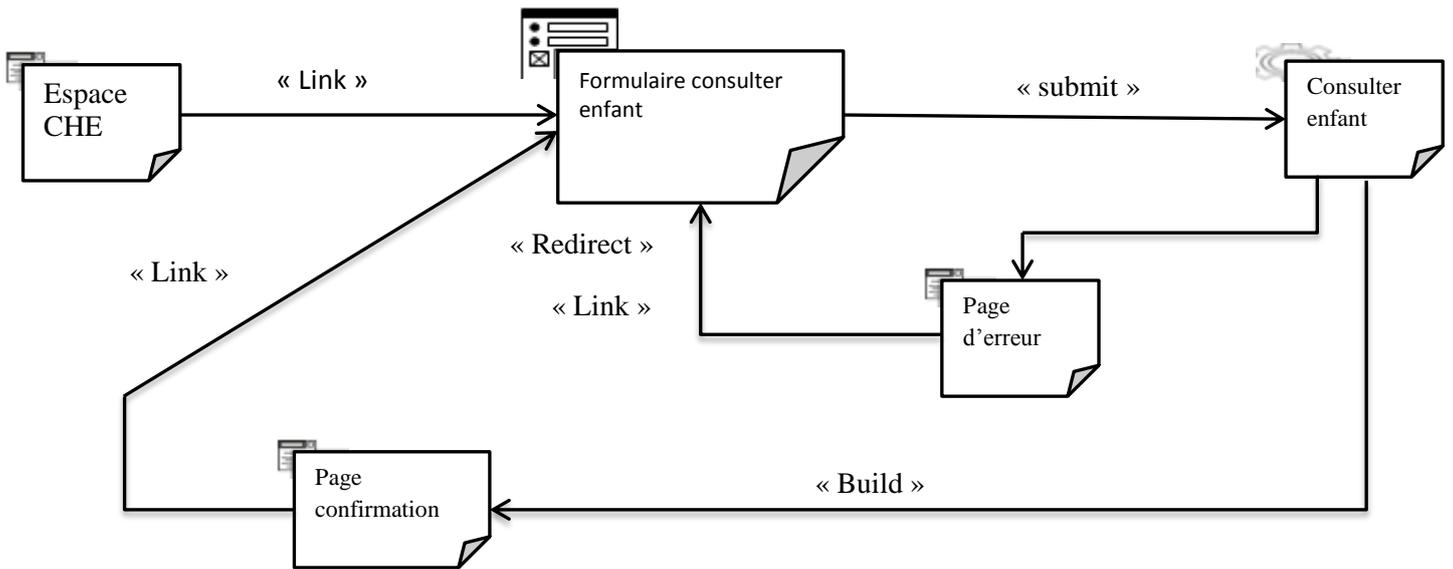


Figure.III.16 Diagramme de classe du cas d'utilisation « consulter enfant »

III.3.6.2 Diagramme de la classe global :

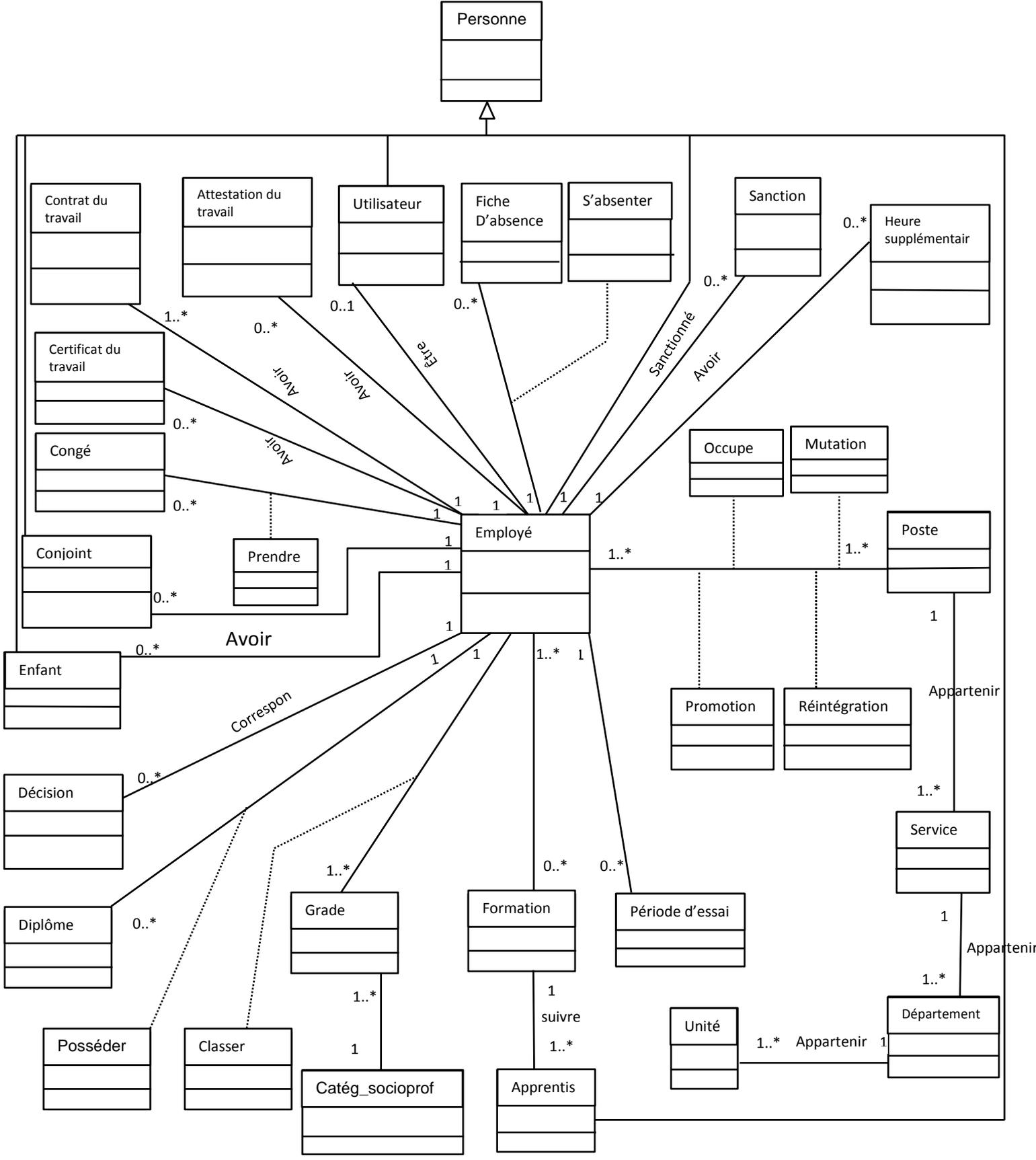


Figure III.17:Diagramme de classe

**III.3.7 conception de la base de données :**

**III.3.7.1. Dictionnaire de données :** Le tableau suivant représente un descriptif du codage de l'information utilisé dans les différents modèles.

✓ **Table des abréviations :**

<b>Libellé</b>	<b>Abréviation</b>
Date (jj /MM/AAAA)	<b>Date</b>
Numérique	N
Alphabétique	A
Alphanumérique	AN

✓ **Dictionnaire de donner :**

<i>Code</i>	<i>Désignation</i>	<i>Type</i>	<i>Taille</i>
MAT_EMP	Matricule de l'employé	AN	06
NOM_EMP	Nom de l'employé	AN	25
PRENOM_EMP	Prénom l'employé	AN	25
DAT_NAIS_EMP	Date de naissance de l'employé	DATE	
LIEU_NAIS_EMP	Lieu de naissance de l'employé	AN	100
ADRESSE_EMP	Adresse de l'employé	AN	100
NAT_EMP	Nationalité de l'employé	AN	25
SEXE_EMP	Sexe de l'employé	A	1
SIT_FAM_EMP	Situation familiale de l'employé	A	1
NBR_ENF_EMP	Nombre d'enfants de l'employé	N	2
EMAIL_EMP	Email de l'employé	AN	40
FONCT_EMP	Fonction de l'employé	AN	100
DATE_RECRUT_EMP	Date de recrutement de l'employé	DATE	
SAL_BASE	Salaire de base	N	8
NUM_ATTEST_TRAV	Numéro de l'attestation de travail	N	7
DAT_ATTEST_TRAV	Date de l'attestation de travail	DATE	

DUR_FONCT	Durée de la fonction	AN	25
NOM_RESP	Nom du responsable de l'employé	AN	25
PRENOM_RESP	Prénom du responsable de l'employé	AN	25
NOM_ADJ_RESP	Nom de l'adjoint du responsable	AN	25
PRENOM_ADJ_RESP	Prénom de l'adjoint du responsable	AN	25
NUM_CERTIF	Numéro du certificat de travail	N	7
DAT_DEBUT_CARRIERE	Date de début de carrière de l'employé	DATE	
DAT_FIN_CARRIERE	Date de fin de carrière de l'employé	DATE	
FONCT_OCCUP	Fonction occupée par l'employé	AN	100
DAT_CERTIF	Date du certificat de travail	DATE	
COD_FOR	Code de la formation	AN	10
TYP_FOR	Type de la formation	AN	50
NOM_FORMATEUR	Nom du formateur	AN	25
DAT_DEBUT_FOR	Date début de la formation	DATE	
DUR_FOR	Durée de la formation	AN	25
FAMILLE_FOR	Famille de la formation	AN	40
SESSION_FOR	Session de la formation	AN	40
COD_CONGE	Code du congé	AN	15
OBJET_CONGE	Objet de la demande de congé	AN	40
DATE_DEBUT_CONGE	Date début du congé	DATE	
DUREE_CONGE	Durée du congé	AN	25
COD_CONJ	Code du conjoint	AN	5
NOM_CONJ	Nom du conjoint	AN	25
PRENOM_CONJ	Prénom du conjoint	AN	25
DAT_NAIS_CONJ	Date de naissance du conjoint	DATE	
SIT_PROF_CONJ	Situation professionnel du conjoint	A	1
COD_CONT	Code du contrat	AN	10

TYP_CONT	Type du contrat	AN	25
DUR_CONT	Durée du contrat	AN	25
DAT_DEBUT_CONTRAT	Date début du contrat	DATE	
COD_ENF	Code de l'enfant	AN	2
NOM_ENF	Nom de l'enfant	AN	25
PRENOM_ENF	Prénom de l'enfant	AN	25
DAT_NAIS_ENF	Date de naissance de l'enfant	DATE	
COD_SCOL_ENF	Code scolaire de l'enfant	AN	40
SEXE_ENF	Sexe de l'enfant	A	1
NUM_FICHE_ABS	Numéro de la fiche d'absences	N	10
DATE_FICHE_PRES	Date de la fiche d'absences	DATE	
NBR_HEUR_ABS	Nombre d'heures d'absence	N	10
JUSTIFICATIONS	Justifications de l'absence	AN	200
NUM_HEUR_SUP	Numéro d'heures supplémentaires	N	10
DATE_HEUR_SUP	Date d'heures supplémentaires	DATE	
NBR_HEUR_SUP	Nombre d'heures supplémentaires	N	10
COD_SANC	Code de la sanction	AN	10
TYP_SANC	Type de la sanction	AN	40
DAT_SANC	Date de la sanction	DATE	
DUR_SANC	Durée de la sanction	AN	25
COD_DECIS	Code décision	AN	15
DAT_DECIS	Date décision	DATE	
DES_DECIS	Désignation de la décision	AN	50
DAT_DEM_DECIS	Date demande de la décision	DATE	
COD_POST	Code de poste	AN	15

DAT_OCC_POST	Date occupation de poste	DATE	
TACH_POST	Taches du poste	AN	50
NIV_ETUD_REQ	Niveau d'étude requis	AN	25
DAT_PROM	Date de la promotion	DATE	
MTF_REIN	Motif de réintégration	AN	50
DAT_REIN	Date de réintégration	DATE	
DAT_MUT	Date de mutation	DATE	
COD_DIPL	Code de diplôme	AN	05
DAT_OB_DIPL	Date obtention de diplôme	DATE	
DES_DIPL	Désignation de diplôme	AN	50
COD_GRAD	Code de grade	AN	05
DES_GRAD	Désignation de grade	AN	50
DAT_OB_GRAD	Date obtention grade	DATE	
COD_PER_ESS	Code période d'essai	AN	10
DAT_DEB_PER_ESS	Date début période d'essai	DATE	
DUR_PER_ESS	Duré période d'essai	AN	15
COD_SER	Code service	AN	05
DES_SER	Désignation de service	AN	40
COD_DEP	Code de département	AN	07
DES_DEP	Désignation de département	AN	35
COD_UNIT	Code d'unité	AN	07
DES_UNIT	Désignation d'unité	AN	35

MAT_APP	Matricule de l'apprenti	AN	07
NOM_APP	Nom de l'apprenti	A	25
PREN_APP	Prénom de l'apprenti	A	25
SEXE_APP	Sexe de l'apprenti	A	1
DAT_NAIS_APP	Date de naissance de l'apprenti	DATE	
COD_CSCP	Code de catégorie socio professionnel	AN	05
DES_CSCP	Désignation de catégorie socio professionnel	AN	40
ID_USER	Identifiant de l'utilisateur	N	2
LOGIN_USER	Login de l'utilisateur	AN	25
PASSWORD_USER	Mot de passe de l'utilisateur	AN	25

Tableau.III.1 Tableau de décisionnaire de données»

### III.3.7.2. Modèle relationnel :

*Employé* (MAT\_EMP, NOM\_EMP, PRENOM\_EMP, DATE\_NAIS\_EMP, LIEU\_NAIS\_EMP, ADRESSE\_EMP, NAT\_EMP, SEXE\_EMP, SIT\_FAM\_EMP, NBR\_ENF\_EMP, EMAIL\_EMP, FONCT\_EMP, DATE\_RECRUT\_EMP,).

*Attestation travail* (num attest tray, date\_attest\_trav, durée\_fonct, nom\_resp, prenom\_resp, nom\_adj\_resp, prenom\_adj\_resp, #mat\_emp).

*Certificat travail* (num certif, date\_debut\_carrière, date\_fin\_carrière, fonct\_occup, date\_certif, nom\_responsable, prenom\_responsable, #mat\_emp).

*Congé* (code congé, objet\_congé, date\_debut\_congé, durée\_congé).

*Prendre* (mat emp, code congé).

*Conjoint* (code conj, nom\_conj, prenom\_conj, date\_nais\_conj, sit\_prof\_conj, #mat\_emp).

*Contrat travail* (code contrat, type\_contrat, durée\_contrat, date\_debut\_contrat, #mat\_emp).

**Decision** (cod decis, dat\_decis, des\_decis, dat\_dem\_decis, #mat\_emp).

**Enfant** (code enf, nom\_enf, prenom\_enf, date\_nais\_enf, sexe\_enf, code\_scolaire\_enf, #mat\_emp).

**Fiche absences** (num fiche abs, date\_fiche\_abs, #mat\_emp, nbr\_heur\_abs, justifications).

**S'absenter**(mat emp, num fiche abs).

**Formation** (code for, type\_for, nom\_formateur, date\_debut\_for, durée\_for, famille\_for, session\_for, #mat\_emp).

**Apprentis** (mat app, nom\_app, pren\_app, sexe\_app, dat\_nais\_app, #code\_for).

**Sanction** (code\_sanction, type\_sanction, date\_sanction, durée\_sanction, #mat\_emp).

**Poste** (cod post, tach\_post, niv\_etud\_req).

**Service** (cod ser, des\_ser, #\_cod\_post).

**Département** (cod dep, des\_dep, #\_cod\_ser).

**Unité**(cod unit, des\_unit, #\_cod\_dep).

**Occupe** (mat emp, cod post, dat\_occ\_post).

**Mutation** (mat emp, cod post, dat\_mut).

**Promotion** (mat emp, cod post, dat\_prom).

**Réintégration** (mat emp, cod post, dat\_rein, mtf\_rein).

**Diplôme** (cod dipl, des\_dipl).

**Posséder** (mat emp, cod dipl, dat\_ob\_dipl).

**Grade** (cod grad, des\_grad, #cod\_cscp).

**Catég\_sosioprof** (cod cscp, des\_cscp).

**Classe** (mat emp, cod grad, sal\_base, dat\_ob\_grad).

**Periode d'essai**(cod per ess, dat\_deb\_per\_ess, dur\_per\_ess,# mat\_emp).

*Utilisateur* (id\_user, login\_user, password\_user, #mat\_emp).

*Heure Supplémentaires* (NUM\_HEUR\_SUP, Dat\_heur\_sup , nbr\_heur\_sup, #mat\_emp).

### III.3.7.3. Le modèle physique des données :

Le modèle physique données est la traduction du modèle logique de données dans un langage de description de données spécifique au système de gestion des base de donnée (table) issue du modèle logique de données en machine afin d'aboutir à la description des fichiers de base de données.

#### Description des tables :

##### ✓ Table employé

Non du champ	Null ?	type	Clé
MAT_EMP	NOT NULL	VARCHAR(6)	PK
NOM_EMP	NOT NULL	VARCHAR(25)	
PRNOM_EMP	NOT NULL	VARCHAR(25)	
DAT_NAIS_EMP	NOT NULL	DATE	
LIEU_NAIS_EMP	NOT NULL	VARCHAR(100)	
ADRESSE_EMP	NOT NULL	VARCHAR(100)	
NAT_EMP	NOT NULL	VARCHAR(25)	
SEXE_EMP	NOT NULL	CHAR(1)	
SIT_FAM_EMP	NOT NULL	CHAR(1)	
NBR_ENF_EMP	NOT NULL	NUMBER(2)	
EMAIL_EMP	NOT NULL	VARCHAR(40)	
FONCT_EMP	NOT NULL	VARCHAR(100)	
DAT_RECRUT_EMP	NOT NULL	DATE	

Tableau.III.2 Table employé

✓ **Table utilisateur**

Non du champ	Null ?	type	Clé
ID_USER	NOT NULL	NUMBER (2)	PK
LOGIN_USER	NOT NULL	VARCHAR(25)	
PASSWORD_USER	NOT NULL	VARCHAR2(25)	
MAT_EMP	NOT NULL	VARCHAR(6)	FK

**Tableau.III.3 Table utilisateur**✓ **Table enfant**

Non du champ	Null ?	type	Clé
COD_ENF	NOT NULL	VARCHAR2(2)	Pk
NOM_ENF	NOT NULL	VARCHAR2(25)	
PRNOM_ENF	NOT NULL	VARCHAR2(25)	
DNS_ENF	NOT NULL	DATE	
COD_SCOL_ENF	NOT NULL	VARCHAR2(40)	
MAT_EMP	NOT NULL	VARCHAR(6)	FK

**Tableau.III.4 Table enfant**✓ **Table conjoint**

Non du champ	Null ?	type	Clé
CODE_CONJ	NOT NULL	VARCHAR2(5)	PK
NOM_CONJ	NOT NULL	VARCHAR2(25)	
PRNOM_CONJ	NOT NULL	VARCHAR2(25)	
DAT_NAIS_CONJ	NOT NULL	DATE	
SIT_PROF_CONJ	NOT NULL	CHAR(1)	
MAT_EMP	NOT NULL	VARCHAR(6)	FK

**Tableau.III.5 Table conjoint**

✓ **Table fiche d'absences**

Non du champ	Null ?	type	Clé
NUM_FICHE_ABS	NOT NULL	NUMBER (10)	PK
DATE_FICHE_PRES	NOT NULL	DATE	
NBR_HEUR_ABS	NOT NULL	NUMBER (10)	
JUSTIFICATIONS	NOT NULL	VARCHAR2(200)	

**Tableau.III.6 Table fiche d'absence**✓ **Table congé**

Non du champ	Null ?	type	Clé
COD_CONGE	NOT NULL	VARCHAR2(15)	PK
OBJET_CONGE	NOT NULL	VARCHAR2(40)	
DATE_DEBUT_CONGE	NOT NULL	DATE	
DUR_CONGE	NOT NULL	VARCHAR2(15)	

**Tableau.III.7 Table congé**✓ **Table décision**

Non du champ	Null ?	type	Clé
COD_DECIS	NOT NULL	VARCHAR2(15)	PK
DAT_DECIS	NOT NULL	DATE	
DES_DECIS	NOT NULL	VARCHAR2(50)	
DAT_DEM_DECIS	NOT NULL	DATE	
MAT_EMP	NOT NULL	VARCHAR2(6)	FK

**Tableau.III.8 Table décision**

✓ **Table heure supplémentaire**

Non du champ	Null ?	type	Clé
NUM_HEUR_SUP	NOT NULL	NUMBER (10)	PK
DATE_HEUR_SUP	NOT NULL	DATE	
NBR_HEUR_SUP	NOT NULL	NUMBER (10)	
MAT_EMP	NOT NULL	VARCHAR2(6)	FK

**Tableau.III.9 Table heure**✓ **Table période d'essai**

Non du champ	Null ?	type	Clé
COD_PER_ESS	NOT NULL	VARCHAR2(10)	Pk
DAT_DEB_PER_ESS	NOT NULL	DATE	
DUR_PER_ESS	NOT NULL	VARCHAR2(15)	
MAT_EMP	NOT NULL	VARCHAR(6)	FK

**Tableau.III.10 Table période d'essai**✓ **Table poste :**

Non du champ	Null ?	type	Clé
COD_POST	NOT NULL	VARCHAR2(15)	PK
DAT_OCC_POST	NOT NULL	DATE	
TACH_POST	NOT NULL	VARCHAR2(50)	
NIV_ETUD_REQ	NOT NULL	VARCHAR2(25)	

**Tableau.III.11 Table poste**

✓ **Table sanction**

Non du champ	Null ?	type	Clé
COD_SANC	NOT NULL	VARCHAR2(10)	PK
TYP_SANC	NOT NULL	VARCHAR2(40)	
DAT_SANC	NOTE NULL	DATE	
DUR_SANC	NOTE NULL	VARCHAR2(25)	
MAT_EMP	NOTE NULL	VARCHAR2(6)	FK

**Tableau.III.12 Table sanction**✓ **Table formation**

Non du champ	Null ?	type	Clé
COD_FOR	NOT NULL	VARCHAR2(10)	PK
TYP_FOR	NOT NULL	VARCHAR2(50)	
NOM_FORMATEUR	NOT NULL	VARCHAR2(25)	
DAT_DEBUT_FOR	NOT NULL	DATE	
DUR_FOR	NOT NULL	VARCHAR2(25)	
FAMILLE_FOR	NULL	VARCHAR2(40)	
SESSION_FOR	NULL	VARCHAR2(40)	
MAT_EMP	NOT NULL	VARCHAR2(6)	FK

**Tableau.III.13 Table formation**✓ **Table contrat du travail**

Non du champ	Null ?	type	Clé
COD_CONT	NOT NULL	VARCHAR2(10)	PK
TYP_CONT	NOT NULL	VARCHAR2(25)	
DUR_CONT	NOT NULL	VARCHAR2(25)	
DAT_DEBUT_CONT	NOT NULL	DATE	
MAT_EMP	NOT NULL	VARCHAR2(6)	FK

**Tableau.III.14 Table contrat**

✓ **Table diplôme**

Non du champ	Null ?	type	Clé
COD_DIPL	NOT NULL	VARCHAR2(5)	PK
DES_DIPL	NOT NULL	VARCHAR2(50)	

Tableau.III.15 Table diplôme

✓ **Table grade**

Non du champ	Null ?	type	Clé
COD_GRAD	NOT NULL	VARCHAR(5)	PK
DES_GRAD	NOT NULL	VARCHAR(50)	
COD_CSCP	NOT NULL	VARCHAR(5)	

Tableau.III.16 Table grade

✓ **Table attestation travail**

Non du champ	Null ?	type	Clé
NUM_ATTEST_TRAV	NOT NULL	NUMBER (7)	PK
DAT_ATTEST_TRAV	NOT NULL	DATE	
DUR_FONCT	NOT NULL	VARCHAR2(25)	
NOM_RESP	NOT NULL	VARCHAR2(25)	
PRENOM_RESP	NOT NULL	VARCHAR2(25)	
NOM_ADJ_RESP	NULL	VARCHAR2(25)	
PRENOM_ADJ_RESP	NULL	VARCHAR2(25)	
MAT_EMP	NOT NULL	VARCHAR2(6)	FK

Tableau.III.17 Table attestation travail

✓ **Table certificat travail**

Non du champ	Null ?	type	Clé
NUM_CERTIF	NOT NULL	NUMBER (7)	PK
DAT_DEBUT_CARRIERE	NOT NULL	DATE	
DAT_FIN_CARRIERE	NOT NULL	DATE	
FONCT_OCCUP	NOT NULL	VARCHAR2(100)	
DAT_CERTIF	NOT NULL	DATE	
MAT_EMP	NOT NULL	VARCHR2(6)	FK

Tableau.III.18 Table certificat travail

✓ **Table s'absenter**

Non du champ	Null ?	type	Clé
MAT_EMP	NOT NULL	VARCHAR(6)	PK
NUM_FICHE_ABS	NOT NULL	NUMBER (10)	PK

Tableau.III.19 Table s'absenter

✓ **Table prendre**

Non du champ	Null ?	type	Clé
MAT_EMP	NOT NULL	VARCHAR(6)	PK
COD_CONGE	NOT NULL	VARCHAR2(15)	PK

Tableau.III.20 Table prendre

✓ **Table occupe**

Non du champ	Null ?	type	Clé
MAT_EMP	NOT NULL	VARCHAR(6)	PK
COD_POST	NOT NULL	VARCHAR2(15)	PK
DAT_OCC_POST	NOT NULL	DATE	

Tableau.III.21 Table occupe

✓ **Table mutation**

Non du champ	Null ?	type	Clé
MAT_EMP	NOT NULL	VARCHAR(6)	PK
COD_POST	NOT NULL	VARCHAR(15)	PK
DAT_MUT	NOT NULL	DATE	

**Tableau.III.22 Table mutation**✓ **Table promotion**

Non du champ	Null ?	Type	Clé
MAT_EMP	NOT NULL	VARCHAR(6)	PK
COD_POST	NOT NULL	VARCHAR(15)	PK
DAT_PROM	NOT NULL	DATE	

**Tableau.III.23 Table promotion**✓ **Table réintégration**

Non du champ	Null ?	type	Clé
MAT_EMP	NOT NULL	VARCHAR(6)	PK
COD_POST	NOT NULL	VARCHAR(15)	PK
MTF_REIN	NOT NULL	VARCHAR(50)	
DAT_REIN	NOT NULL	DATE	

**Tableau.III.24 Table réintégration**✓ **Table posséder**

Non du champ	Null ?	type	Clé
MAT_EMP	NOT NULL	VARCHAR(6)	PK
COD_DIPL	NOT NULL	VARCHAR(5)	PK
DAT_OBT_DIPL	NOT NULL	DATE	

**Tableau.III.25 Table posséder**

✓ **Table classe**

Non du champ	Null ?	type	Clé
MAT_EMP	NOT NULL	VARCHAR(6)	PK
COD_GRAD	NOT NULL	VARCHAR(5)	PK
SAL_BASE	NOT NULL	NUMBER(8)	
DAT_OB_GRAD	NOT NULL	DATE	

**Tableau.III.26 Table classe**✓ **Table service**

Non du champ	Null ?	type	Clé
COD_SER	NOT NULL	VARCHAR(5)	PK
DES_SER	NOT NULL	VARCHAR(40)	
COD_POST	NOT NULL	VARCHAR(15)	FK

**Tableau.III.27 Table service**✓ **Table département**

Non du champ	Null ?	type	Clé
COD_DEP	NOT NULL	VARCHAR(5)	PK
DES_DEP	NOT NULL	VARCHAR(40)	
COD_SER	NOT NULL	VARCHAR(5)	FK

**Tableau.III.28 Table département**✓ **Table unité**

Non du champ	Null ?	type	Clé
COD_UNIT	NOT NULL	VARCHAR(5)	PK
DES_UNIT	NOT NULL	VARCHAR(40)	
COD_DEP	NOT NULL	VARCHAR(5)	PK

**Tableau.III.29 Table unité**

✓ **Table apprentis**

Non du champ	Null ?	type	Clé
MAT_APP	NOT NULL	VARCHAR(6)	PK
NOM_APP	NOT NULL	VARCHAR(25)	
PREN_APP	NOT NULL	VARCHAR(25)	
SEXE_APP	NOT NULL	CHAR(1)	
DAT_NAIS_APP	NOT NULL	DATE	
COD_FOR	NOT NULL	VARCHAR(25)	FK

**Tableau.III.30 Table apprentis**✓ **Table catg\_socioprof**

Non du champ	Null ?	type	Clé
COD_CSCP	NOT NULL	VARCHAR(5)	PK
DES_CSCP	NOT NULL	VARCHAR(40)	

**Tableau.III.31 Table cattg\_socioprof****III.7. Conclusion :**

Ce chapitre est consacré à l'étude conceptuelle de l'application avec le langage UML. Cette partie a mis en évidence les différents composants de l'application et cela dans le but de la construction de notre base de données.

A ce stade du développement, on est prêt à mettre en œuvre l'application, c'est ce qu'on présentera dans le chapitre qui suit.

# *Chapitre 4*



*Réalisation*

### IV.1. Introduction :

Dans ce présent chapitre nous allons entamer la quatrième étape de notre travail, qui consiste en « réalisation d'une application client-serveur pour le service personnel de NAFTA L ».

Pour cela nous avons utilisés des outils, en l'occurrence Oracle Database 10g Express Edition comme système de gestion de base de données, NetBeans comme environnements de développement.

Dans un premier temps, nous allons présenter ces différents outils, en suite nous allons présenter les interfaces de notre application.

### IV.2. Présentation des outils utilisés :

#### IV.2.1. Oracle 10g :

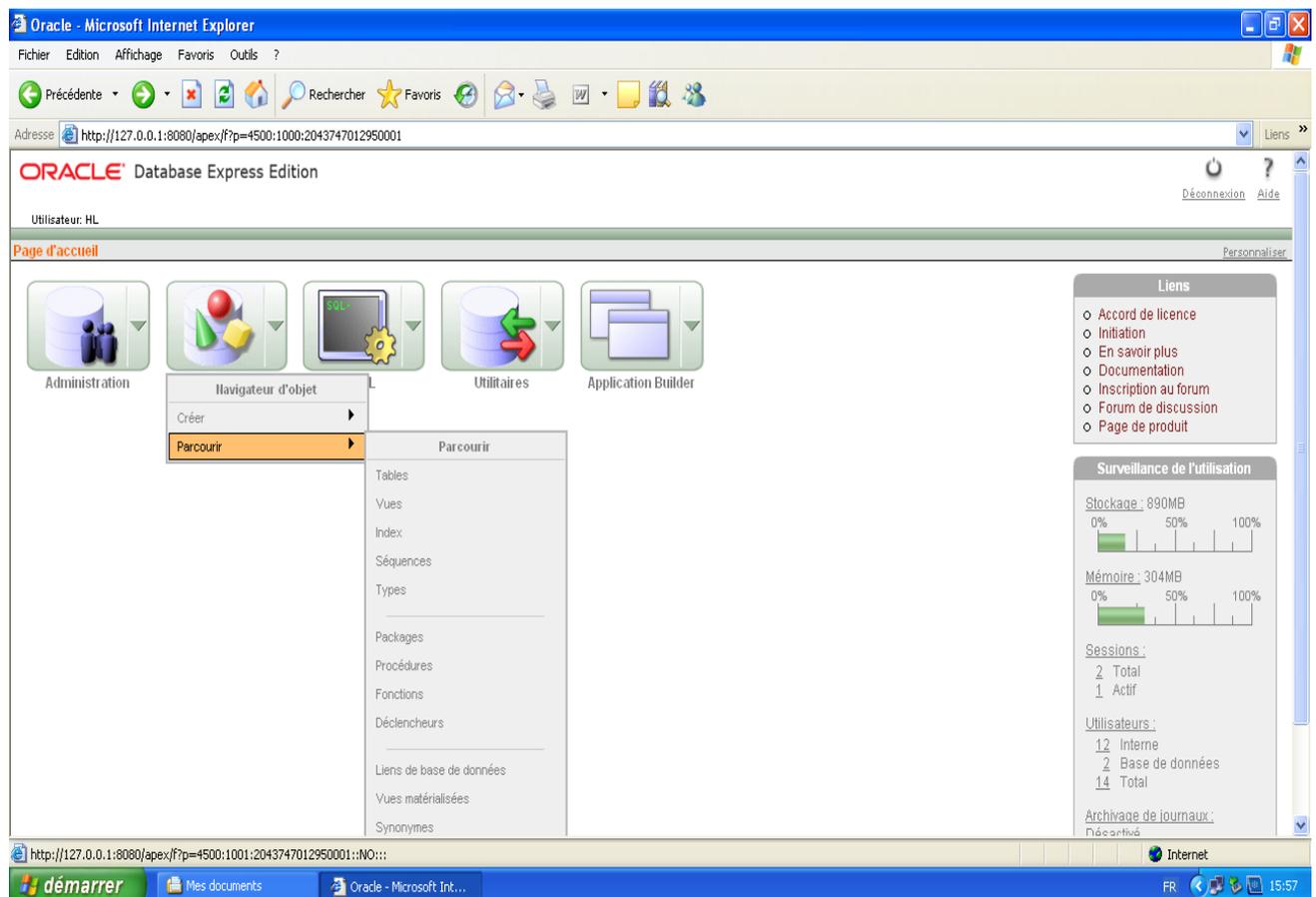


Figure IV.1 : Interface d'Oracle 10 g.

### IV.2.2. NetBeans :

#### a. Historique :

En 1997, NetBeans naît de Xelfi, un projet d'étudiants dirigé par la faculté de mathématiques et de physique de l'Université Charles de Prague. Plus tard, une société se forme autour du projet et édite des versions commerciales de l'EDI NetBeans, jusqu'à ce qu'il soit acheté par Sun en 1999. Sun place le projet sous double licence CDDL et GPL v2 en juin de l'année suivante.

#### b. Définition :

NetBeans est un environnement de développement intégré (EDI) pour Java, 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++, 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 et Open VMS.

#### c. Interface de NetBeans:

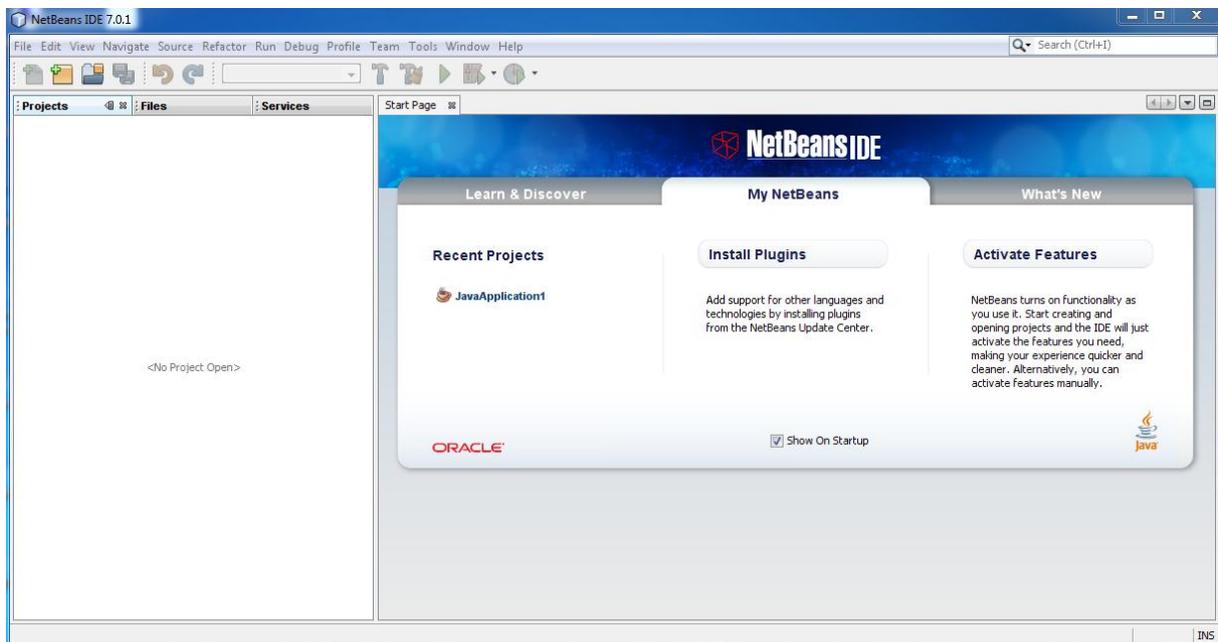


Figure IV.2 : Interface de

### IV.3. Préparation de l'environnement de travail :

Avoir un environnement de travail complet pour le développement de toute application est essentiel. De fait la préparation de l'environnement de travail est une étape primordiale qu'il faut réaliser avec soin, pour éviter des problèmes lors du développement.

En effet notre environnement est préparé comme suit :

- **NetBeans** : nous avons utilisé la version 6.8 qui est gratuite sur le site de Sun Microsystems « <http://java.sun.com> »
- **JDK** (Java Development kit): téléchargeables sur le site de SUN Microsystems (fondateur du langage) « <http://java.sun.com/javase/downloads/index.jsp> » ensuite les installer sur la machine
- **Driver Oracle** : nous l'avons copier sur le répertoire suivant : C:\Program Files\Java\jdk1.6.0\_18\jre\lib\ext Ces outils nous les avons utilisés sous le système d'exploitation Windows XP, et avec comme langage de programmation Java pour NetBeans et le SQL pour Oracle 10g.

### IV.4. Présentation de quelques interfaces :

Vu le nombre important, nous n'allons présenter que quelques interfaces :

- **Page d'accueil** : c'est la première page qui apparaît au lancement de l'application.

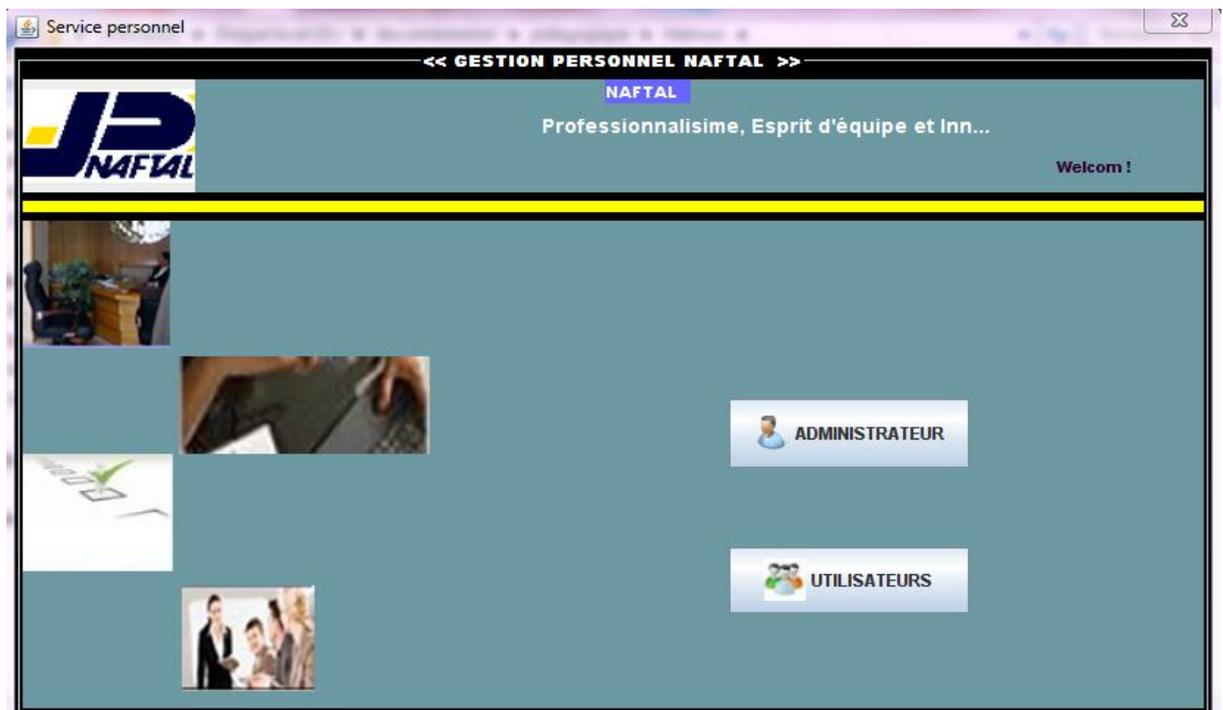


Figure IV.3 : Page d'accueil

- **Identification Administrateur :**

Avant accéder à la fenêtre de gestion utilisateurs nous devons nous identifier, si le mot de passe est incorrect un message d'erreur s'affiche.

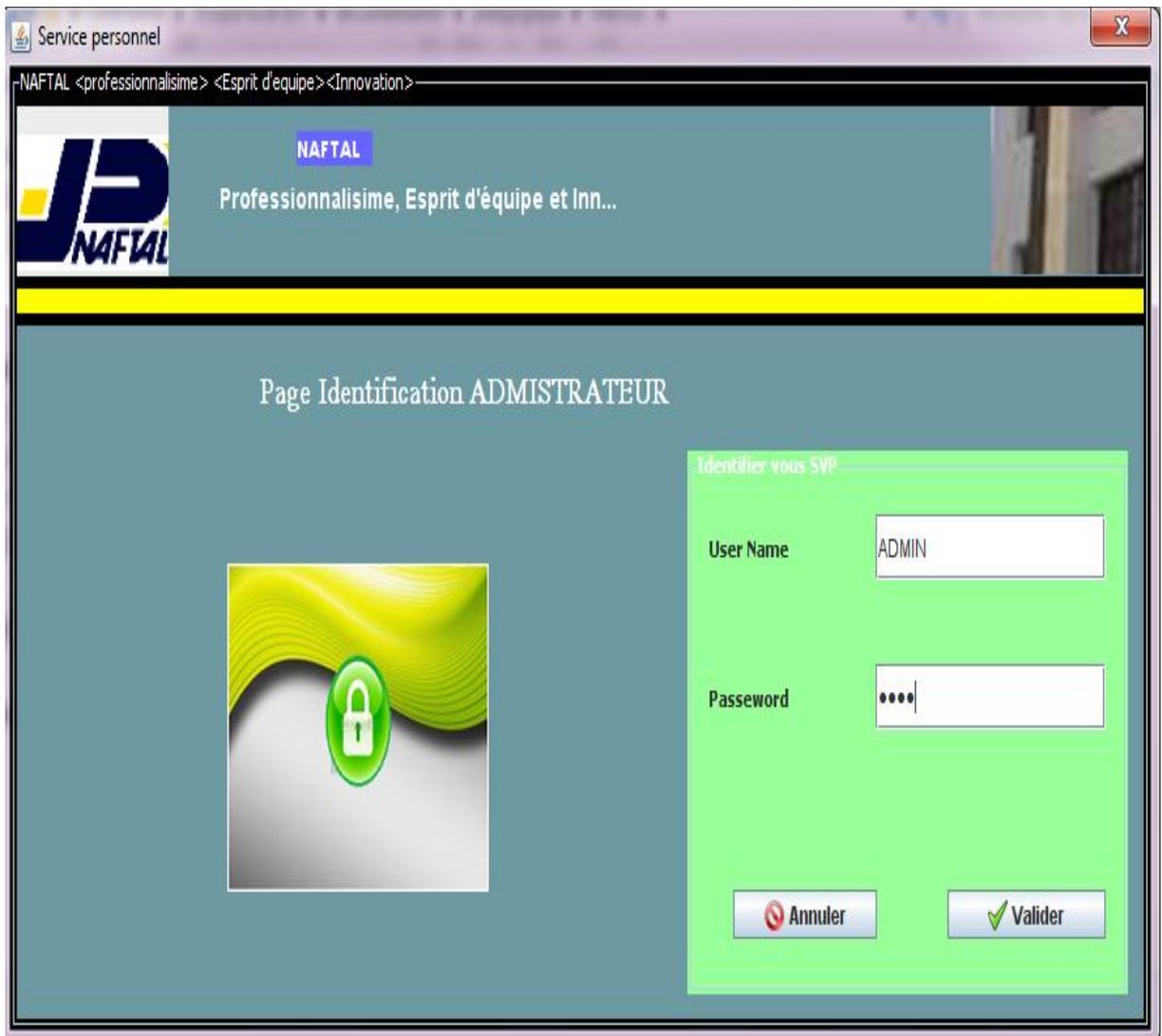


Figure IV.4 : Fenêtre « Identification Administrateur »

- **Identification Utilisateurs :**

Avant d'accéder à la fenêtre MENU, il faut que chaque utilisateur de l'entreprise s'identifie, il choisit son nom dans une liste contenant tous les noms des utilisateurs de l'application.

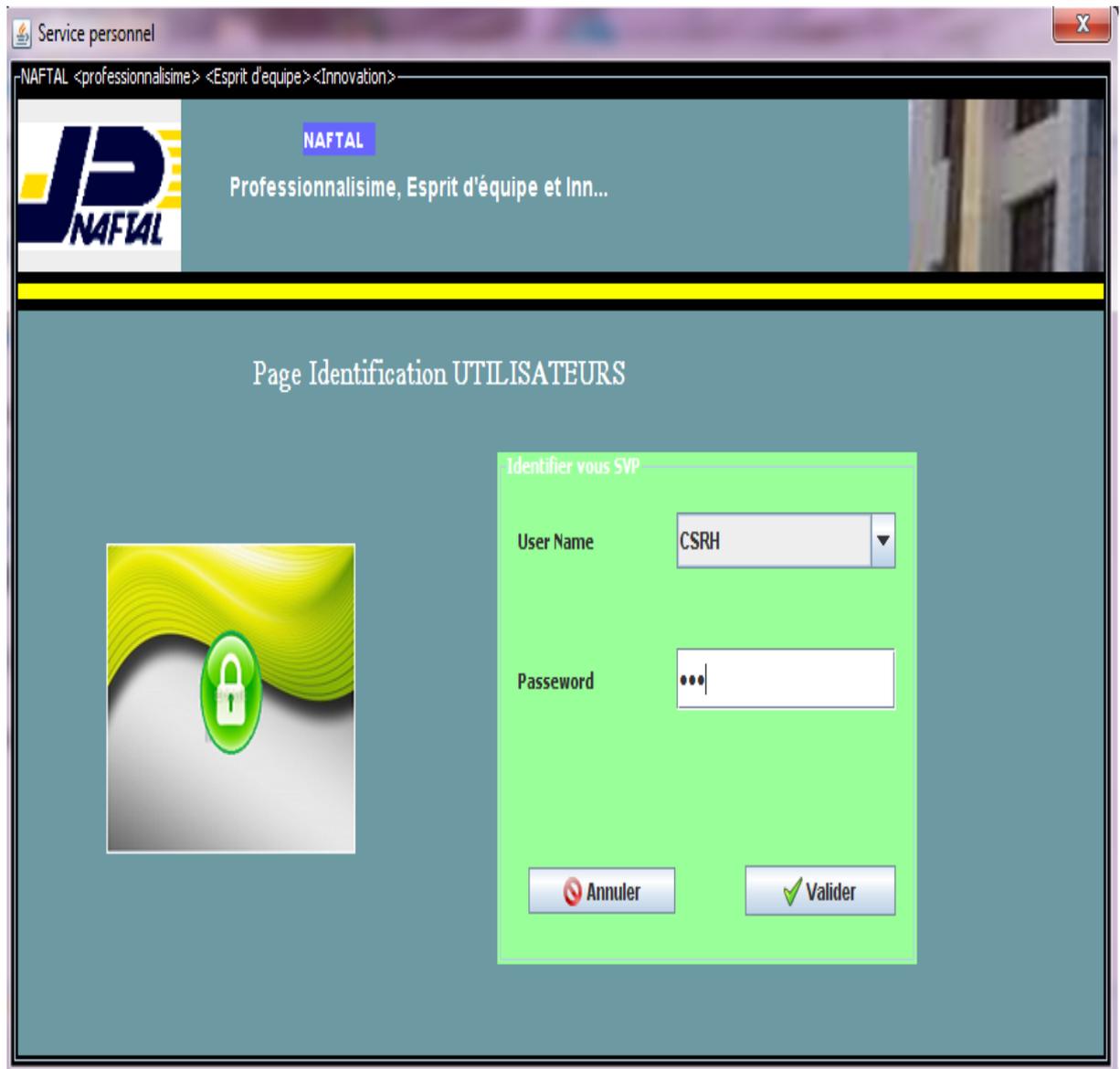


Figure IV.5 : Fenêtre « Identification Utilisateurs »

- **Espace Utilisateurs :**

Dans ce Menu, chacun des neufs utilisateurs pourra accéder aux différentes tables dont il aura besoin et cela selon les privilèges que nous lui avons accordés à base de ses fonctionnalités au sein de l'entreprise.



Figure IV.6 : Fenêtre « Espace Gestion Section Administratif »

- **Page Employé :**

L'utilisateur va choisir dans son propre Menu une table selon les actions qu'il a l'attention d'effectuer (consultation, recherche, affichage, ajout, modification, suppression), ces dernières vont déclencher des traitements dont les résultats seront stockés dans les tables.

Page Employé

Recherche

Ok

Matricule

Nom

Prenom

Date naissance

Lieu naissance

Sex

Natinnalite

Diplome

Date obt diplom

Situation familiale

Nombre d'enfant

Fonction

Date recrute

Email

Adresse

Grade

Date nomination

Salaire de base

Categ\_socioprof

Commande

Ajouter

supprimer

Modifier

Nouveau

Liste

Retour

Figure IV.7 : Fenêtre « Employé »

- Page Fiche d'absence

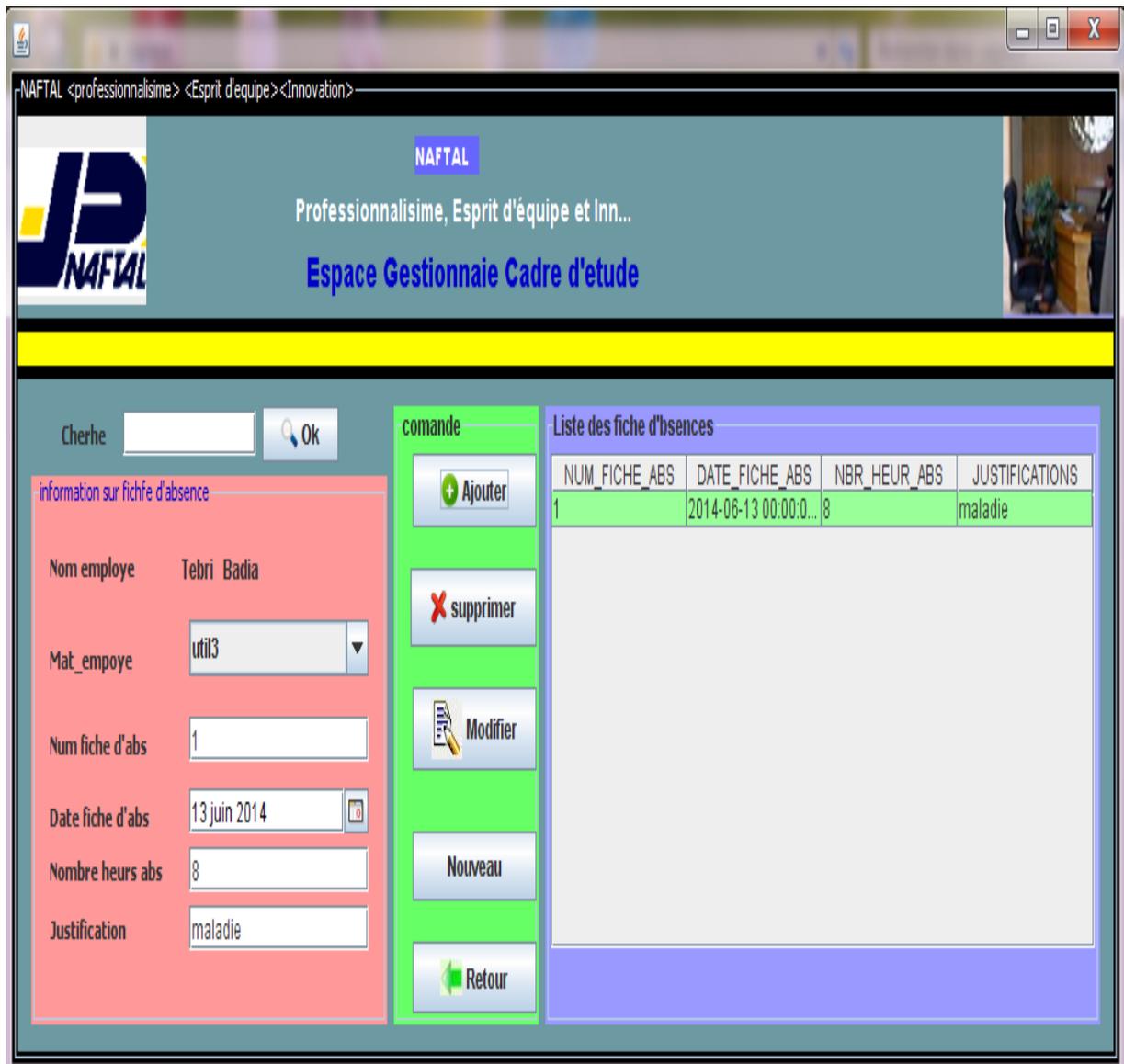


Figure IV.8 : Fenêtre «Page Fiche d'absence»

- **Changer le Mot de passe :**

Chaque utilisateur aura la possibilité de changer son propre mot de passe, il va accéder à la fenêtre changer mot de passe qui choisira dans son propre Menu, ensuite il doit écrire son ancien mot de passe, le nouveau qu'il a choisis et réécrire ce dernier, si l'ancien mot de passe n'existe pas ou si l'écriture et la réécriture du nouveau ne sont pas identiques deux messages d'erreurs différents vont s'afficher pour signaler où réside l'erreur.

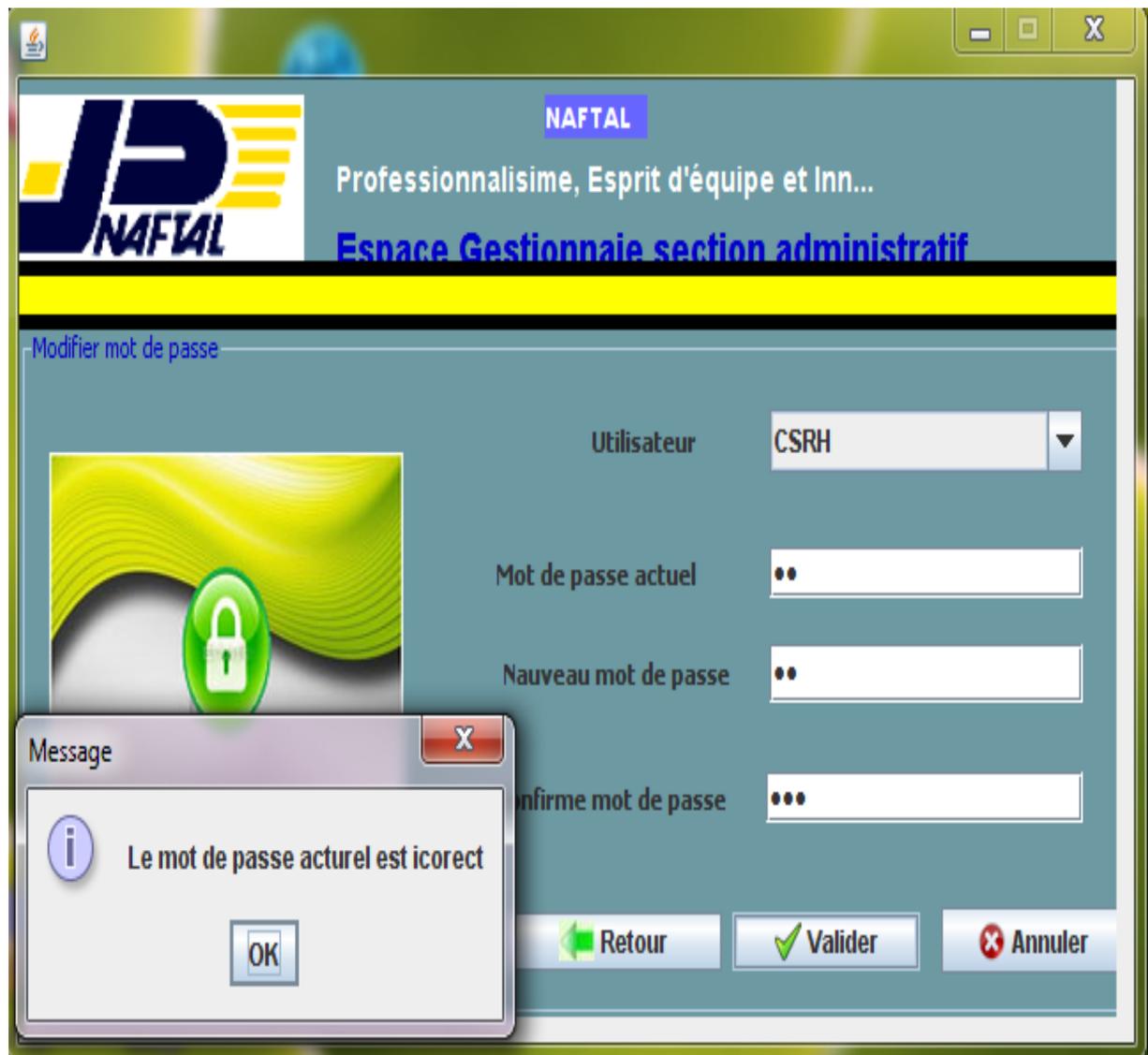


Figure IV.9 : Fenêtre « Changer Mot de passe »

### **V.5. Conclusion :**

Dans cette partie, nous avons présenté quelques exemples de notre application dans le but de vous donner une idée globale et un premier aperçu de notre projet.

# *Annexe A*



## *Description du langage UML*

## 1. Introduction :

Pour faire face à la complexité croissante des logiciels, des nouvelles méthodes et outils de modélisation ont été créés.

La modélisation par objet saisit la structure statique d'un système, en montrant les objets dans le système, leurs relations, ainsi que les attributs et les opérations qui caractérisent chaque classe. Ce mode de conception de système prend son intérêt par rapport à la programmation traditionnelle puisque le modèle objet correspond étroitement au monde réel.

Il est par conséquent plus souple pour les modifications.

Dans ce contexte et devant le foisonnement de nouvelles méthodes de conception « orientée objet », l'Object Management Group (OMG) a eu comme objectif de définir une notation standard utilisable dans les développements informatiques basés sur l'objet. C'est ainsi qu'apparu l'UML.

## 2. Présentation d'UML :

UML (*unified modeling langage*, que l'on peut traduire par “*langage de modélisation unifié*”) est une notation permettant de modéliser un problème de façon standard. Ce langage est né de la fusion des principales méthodes objet : Booch, OMT (*Object Modeling Technique*), et OOSI (Object Oriented Software Engineering). L'UML :

- Est un langage semi formel qui permet de décrire tous les aspects d'une application informatique ;
- Il est issu de la programmation orientée objet. A ce titre il est surtout adapté aux applications réalisées à l'aide d'un langage OO (C++, Java,..) ;
- UML n'est pas une méthodologie. Il ne préconise en effet aucune démarche (n'est pas une méthode mais une notation);
- Encourage un processus itératif et incrémental piloté par le cas d'utilisation et centré sur l'architecture.

## 3. modélisation avec UML :

### 3.1. Définition d'un modèle

Un modèle est une simplification et/ou une abstraction de la réalité. Il doit aider à mieux comprendre, percevoir les relations et les interactions à l'intérieur du système. Il doit permettre de visualiser les conséquences de modifications apportées au système, de visualiser

également les raisons du comportement du système par rapport à une situation donnée. C'est donc un guide pour construire un système stable et fiable. Le modèle doit également aider à documenter le système construit.

### 3.2 La modélisation UML

UML fournit un ensemble d'outils permettant de représenter des éléments du monde objet (classes, objets, ...) ainsi que les liens qui les relient. Toutefois, étant donné qu'une seule représentation est trop subjective, UML fournit un moyen astucieux permettant de représenter diverses projections d'une même représentation grâce aux **vues**. Une vue est constituée d'un ou plusieurs **diagrammes**.

On distingue deux types de vues:

✓ **Les vues statiques :**

Représentant le système physiquement

- diagrammes d'objets
- diagrammes de classes
- diagrammes de cas d'utilisation
- diagrammes de composants
- diagrammes de déploiement

✓ **Les vues dynamiques :** montrant le fonctionnement du système

- diagrammes de séquence
- diagrammes de collaboration
- diagrammes d'états transitions
- diagrammes d'activités

### 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 :** (Modélisation objet avec UML. Edition Eyrolles, 1997) Les éléments structurels sont représentés par des noms dans les modèles UML. Ce sont les parties les plus statiques d'un modèle : ils représentent des éléments conceptuels ou physiques.

#### 4.1.1. Modélisation d'une classe

On appelle classe la structure d'un objet, c'est-à-dire la déclaration de l'ensemble des entités qui composeront un objet. Un objet est donc "issu" d'une classe. En réalité on dit qu'un objet est une **instanciation** d'une classe. Une classe est composée:

- d'attributs: il s'agit des données, dont les valeurs représentent l'état de l'objet
- Les méthodes : il s'agit des opérations applicables aux objets

Une classe se représente avec UML sous forme d'un rectangle divisé en trois sections. Le premier contient le nom donné à la classe (non souligné).

Les attributs d'une classe sont définis par un nom, un type. Les opérations sont répertoriées dans le troisième volet du rectangle.

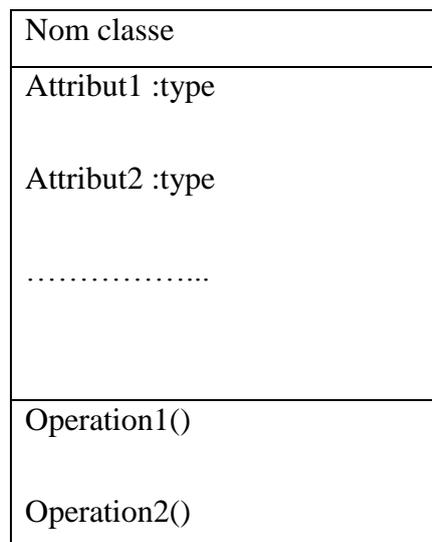


Figure A1 : Représentation d'une classe UML

#### 4.1.2. Les classes-associations :

Une association exprime une connexion sémantique bidirectionnelle entre deux classes. Cette association est représentée par une classe pour ajouter des attributs et des méthodes dans cette association, ce type de classe est appelé classe-association, elle possède au même temps les caractéristiques d'une classe et d'une association.

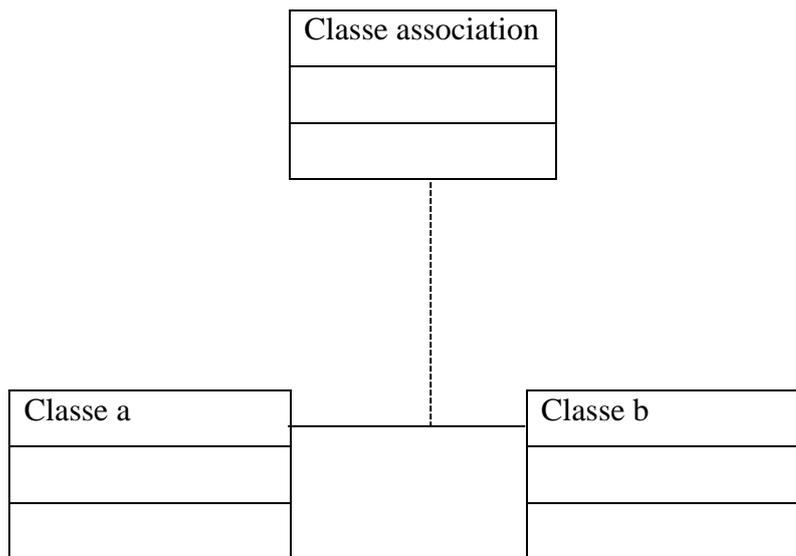


Figure A2 : Classe association

**4.1.3. Cas d'utilisation :** Un cas d'utilisation précise le comportement d'un système ou d'une partie d'un système et décrit un ensemble de séquences d'actions. Les cas d'utilisation servent à saisir le comportement attendu d'un système en cours de développement, sans avoir à préciser la façon dont ce comportement est réalisé.

Il est représenté par une ellipse en traits plein contenant son nom comme le montre la figure suivante :

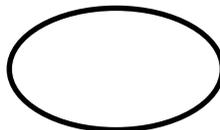


Figure A3 : Cas d'utilisation

## 4.2. Les éléments comportementaux :

Ces éléments représentent les parties dynamiques des modèles UML, ils comprennent les interactions.

### 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 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.



Figure A4 : Exemple de message

#### 4.2.2. 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 ».

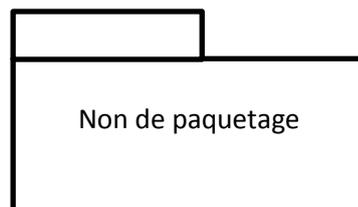


Figure A5 : Exemple de paquetage

#### 4.2.3. Les éléments d'annotation :

Les éléments d'annotation représentent les parties explicatives des modèles UML. Ce sont les commentaires qui peuvent accompagner tout élément dans un modèle, à des fins de description, d'exploitation et de remarque. Il existe un type fondamental d'éléments d'annotation appelé «note» qui est simplement un symbole utilisé pour représenter les contraintes et les commentaires rattachés à un élément ou un ensemble d'éléments. Comme le montre la figure A6, une note est représentée par un rectangle écorné qui contient un commentaire textuel ou graphique



Figure A6 : Exemple de note

## 5. Relations dans UML :

Il existe quatre types de relations dans UML :

- ✓ La dépendance.
- ✓ L'association.
- ✓ La généralisation.
- ✓ La réalisation.

La modélisation objet consiste à créer une représentation abstraite, sous forme d'objets, d'entités. Un objet est caractérisé par plusieurs notions:

- ✓ **Les attributs** (on parle parfois de propriétés): Il s'agit des données caractérisant l'objet. Ce sont des variables stockant des informations d'état de l'objet
- ✓ **Les méthodes** (appelées parfois fonctions membres): Les méthodes d'un objet caractérisent son comportement, c'est-à-dire l'ensemble des actions (appelées opérations) que l'objet est à même de réaliser. Ces opérations permettent de faire réagir l'objet aux sollicitations extérieures (ou d'agir sur les autres objets). De plus, les opérations sont étroitement liées aux attributs, car leurs actions peuvent dépendre des valeurs des attributs, ou bien les modifier
- ✓ **L'identité**: L'objet possède une identité, qui permet de le distinguer des autres objets, indépendamment de son état. On construit généralement cette identité grâce à un identifiant découlant naturellement du problème (par exemple un produit pourra être repéré par un code, une voiture par un numéro de série, ...)

## 6. Définition des besoins :

Notre projet porte sur la mise en place d'une application client/serveur pour le Service Humain.

La figure suivante donne la représentation graphique de la démarche de modélisation :

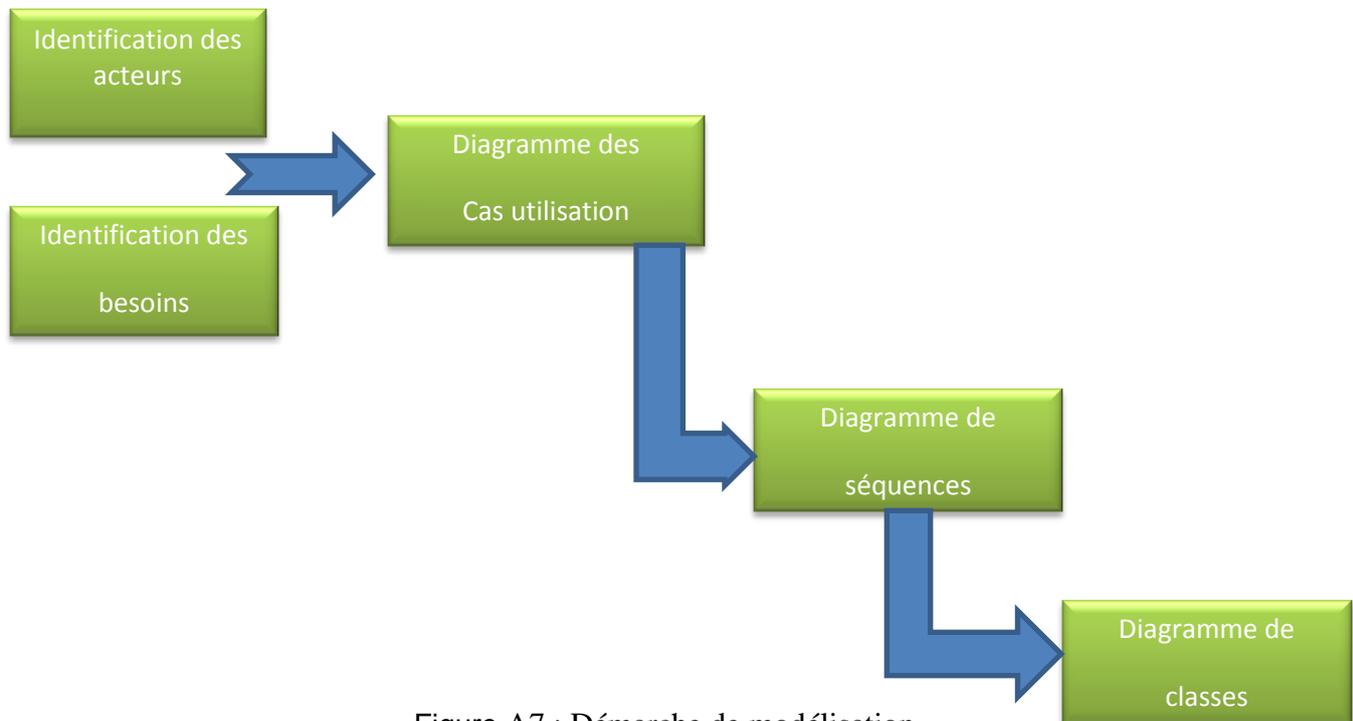


Figure A7 : Démarche de modélisation

## 7. Les acteurs :

Entité externe qui agit sur le système ; Le terme acteur ne désigne pas seulement les utilisateurs humains mais également les autres systèmes. Les acteurs sont des classificateurs qui représentent des rôles au travers d'une certaine utilisation (cas) et non pas des personnes physiques. Ce sont des acteurs types.

L'acteur est représenté dans l'UML par bon homme facile à dessiner :

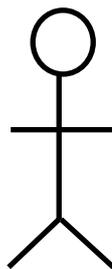


Figure.A8 : Acteur

### 8. Diagramme de contexte :

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

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

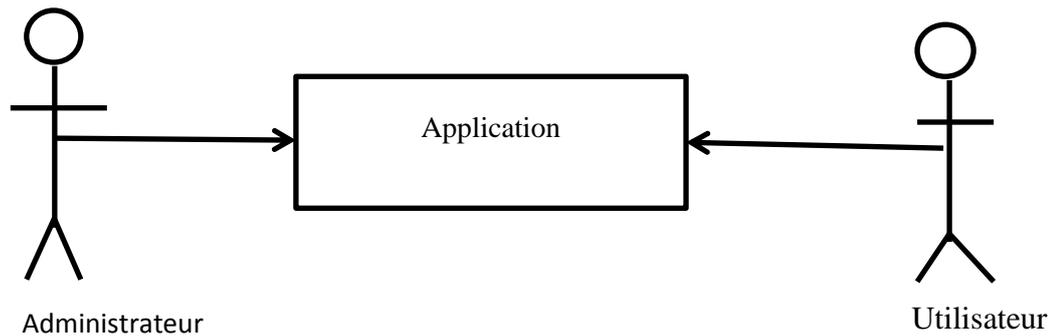


Figure A9 : Diagramme de contexte

### 9. Conclusion

L'UML, comme l'on a vu à travers ce chapitre, ne propose pas une démarche objet mais une notation adaptée 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 à tous les projets informatiques.

# *Annexe B*



*Présentation*

*SGBD Oracle*

## 1. Introduction :

L'informatique est une ressource capitale pour le bon fonctionnement de n'importe quelle structure, ce qui recommande l'utilisation d'un SGBD pour sa sécurisation et sa structuration. Le SGBD oracle que nous présentons dans ce qui suit est l'un des systèmes les plus recommandés.

## 2. Définition d'une base de données :

Une base de données peut être comme étant un ensemble intégré de données modélisant un univers donné. Les données utilisées par des applications diverses sont regroupées dans une même base permettant ainsi d'éviter les problèmes dus à la duplication. A une base de données est associé un schéma appelé schéma conceptuel, qui décrit la structure et le type des données qu'elle contient et éventuellement quelques règles (ou contraintes) qui doivent être toujours vérifiées.

## 3. Définition d'un SGBD : [10]

Un SGBD (Système de Gestion de Bases de Données) permet de définir les structures physique et logique qui constituent une base de données, et qui servent au stockage d'importants volumes d'informations qu'il est ensuite possible d'interroger rapidement et accessibles par des utilisateurs simultanés.

### 3.1. Caractéristiques d'un SGBD :

Parmi les caractéristiques d'un SGBD, on peut citer :

- ✓ **Gestion de gros volume** : Un SGBD peut gérer des tailles importantes de données, ainsi qu'un nombre considérable d'utilisateurs simultanés.
- ✓ **Indépendance logique/physique vis-à-vis des données** : Les données (le plus souvent des tables) sont référencées de manière logique. Un dictionnaire de données permet de retrouver la correspondance avec l'objet physique désiré. Ceci est bien sûr très utile dans les environnements ouverts, et offre une grande souplesse aussi bien lors de la mise en production ou dans la phase de maintenance des applications. La conséquence est que l'on pourra déplacer physiquement des données, par exemple, leurs charger de serveurs, renommer ou retailler un fichier sans pour autant retoucher le code des applications. Il n'y a pas de correspondance bijective entre un fichier et une table.

- ✓ **Langage d'accès standardisé** : Le langage d'accès est normalisé : c'est le SQL. Comme dans toute norme, SGBD est enrichie par les différents fournisseurs de logiciel. Mais si l'on est contraint à croire des procédures respectant la norme ISO/ANSI 92 par exemple, on sera quasiment garanti de pouvoir porter ce code sur Oracle, Informix ou Sybase.
- ✓ **Performance** : Un *SGBD* doit offrir de bonnes performances et des outils lui permettant de les mesurer et de les contrôler via des paramètres de configuration. Des processus d'optimisation en temps réel, des requêtes complexes sont également souvent présentées. Les données peuvent être indexées de manière souple, dynamique et complète (index soit simple, concaténé ou multiple, recherche textuelle, etc.). Un nombre important d'utilisateurs ainsi qu'un volume conséquent de données peuvent être pris en compte.
- ✓ **Disponibilité** : Un *SGBD* se doit d'être fiable et d'offrir une bonne disponibilité des données. Des mécanismes de sauvegarde variés (physique, logique, off line, on line, totale, partielle et incrémentale), ainsi que des mécanismes de journalisation, permettent de restaurer une information cohérente et plus ou moins à jour, voir d'éviter sa perte dans tous les cas de problèmes matériels ou logiciels. Une disponibilité totale des données est possible (temps de reprise nul), il suffit de s'en donner les moyens logiciels et matériels.
- ✓ **Cohérence** : les données réparties doivent être cohérentes ainsi que les données non réparties même si dans ce cas de figure, les mécanismes mis en jeu sont plus complexes.

#### 4. Présentation d'Oracle :

##### 4.1. Historique :

L'histoire d'Oracle commence avec la réalisation d'Oracle Corporation en 1977. La première version conçue à partir des spécifications du système *R d'IBM*, était commercialisée en 1979. La période de 1979 à 1984 était marquée par la commercialisation de divers outils tels que les interfaces avec les langages de troisième génération (*Cobol*, *C*, *Fortran*, etc.), le gestionnaire de grilles d'écrans, le générateur de rapport destiné à faciliter l'utilisation d'Oracle.

L'apparition de la version sur *PC* en 1984 marque le début d'une interopérabilité entre serveurs (mainframes ou mini-ordinateurs) et clients, avec le développement d'un code Oracle sur *PC* qui fonctionne parfaitement après compilation sur un serveur. Dans la version

5 du *SGBDR* apparue en 1986, Oracle introduit des fonctionnalités distribuées avec son module *SQL\*Star* (l'architecture Client/serveur). Deux ans plus tard, la version 6 d'*Oracle* a été commercialisée. Quant à la version 7, elle l'a été en 1992. Cette version constitue une évolution majeure du *SGBD Oracle* car plusieurs nouveautés ont été apportées. Parmi les quelles, on trouve l'intégration des contraintes d'intégrité déclaratives, les procédures stockées, les packages et des améliorations des architectures internes. *Oracle* rejoint *IBM* et *Informix* sur le marché des serveurs universels de données en 1997 avec la version 8, et adopte le système d'exploitation *Linux*. Juste avant, *Oracle* baptise la version 7.3 de son serveur '*Oracle Universal Server*' en s'inspirant d'*Informix* et ajoute au noyau les cartouches contextes, *Media Server* et *Special DATA*. En 1999, *Oracle 8i* (*i* pour *Internet*) sort avec une forte intégration du langage Java, l'Internet est désormais au cœur du *SGBDR*. La version *10g* est apparue en

2003, et ses produits peuvent être répartis en cinq catégories qui sont :

- ✓ La base de données *Oracle 10g* ;
- ✓ *Oracle 10g Application Server* ;
- ✓ Les outils de développement pour *Internet* ;
- ✓ Le *Datawarehousing* et les outils de développement pour l'entreprise ;
- ✓ *Oracle E-Business Suite*.

#### 4.2. Fonctionnalités d'*Oracle* :

Oracle est avant tout un SGBD Relationnel, portable sur une très grande variété de plates-formes matérielles et systèmes d'exploitation. Le SGBD Oracle permet d'assurer :

- ✓ En utilisant les opérateurs relationnels à travers le langage SQL, que les données soient définies et manipulées d'une façon non procédurales ; La cohérence des données en utilisant des unités logiques, des traitements appelés transactions. *Oracle* permet la définition, la validation et l'annulation des transactions ;
- ✓ La confidentialité des données qui est assurée au moyen des concepts de privilèges, de rôles et de vues ;
- ✓ L'intégrité des données : l'une des principales nouveautés de la version 7 est l'enrichissement du langage de définition de données en offrant la possibilité de définir des contraintes d'intégrité ;
- ✓ La sauvegarde et la restauration des données grâce aux techniques de reprise à chaud et à froid à la suite d'une panne matérielle ou logicielle ;

- ✓ La gestion des accès concurrents assurée grâce aux techniques de verrouillage.

#### 4.3. Les composants d'Oracle :

Outre la base de données, la solution Oracle est un véritable environnement de travail constitué de nombreux logiciels permettant notamment une administration graphique d'Oracle, de s'interfacer avec des produits divers de création de bases de données et de configuration de celles-ci :

- ✓ Les outils d'administration.
- ✓ Les outils de développement.
- ✓ Les outils de communication.
- ✓ Les outils de génie logiciel.
- ✓ Les outils d'aide à la décision.

##### 4.3.1. Les outils d'administration d'Oracle :

Oracle est fourni avec de nombreux outils permettant de simplifier l'administration de la base de données. Parmi ces outils, les plus connus sont :

- ✓ Oracle manager (SQL\*DBA).
- ✓ Net Works manager.
- ✓ Oracle entreprise manager.
- ✓ Import /Export : un outil permettant d'échanger des données entre deux bases oracle.

##### 4.3.2. Les outils de développement d'Oracle :

Oracle propose également de nombreux outils de développement permettant d'automatiser la création d'application s'interfaçant avec la base de données. Ces outils de développement sont :

- ✓ **Oracle Designer.**
- ✓ **SQL\*Plus :** Une interface interactive permet d'envoyer des requêtes SQL et PL/SQL à la base de données. SQL\*Plus permet notamment de paramétrer l'environnement de travail (formatage des résultats, longueur d'une ligne, nombre de lignes par page,...).
- ✓ **Oracle développer :** Il s'agit d'une suite de produits destinés à la conception et à la création des applications client/serveur. Il est composé de quatre applications :
- ✓ **Oracle Forms :** (anciennement SQL\*Forms) : un outil permettant d'interroger la base de données de façon graphique sans connaissances préalables du langage SQL. SQL\*Forms permet ainsi de développer des applications graphiques (fenêtres,

formulaire,...) permettant de sélectionner, modifier et supprimer des données dans la base.

- ✓ **Oracle Reports** (SQL\*Report Writer) : un outil permettant de réaliser des états.
- ✓ **Oracle Graphics** : Un outil de génération automatique de graphiques dynamique pour présenter graphiquement des statistiques réalisées à partir des données de la base.
- ✓ **Procédure Builder** : Un outil permettant de développer des procédures, des fonctions et des packages.

#### 4.4. SQL (Structured Query Language) :

Plusieurs langages ont fait leur apparition entre les années 70 et 76 dont les principaux sont : QUEL et SEQUEL implantés respectivement sur INTEGRES et système R.

Le langage SQL provient du langage system R d'IBM. QSL est implanté dans des produits commerciaux à la fin des années 70 avec la première version d'oracle et de SQL/DS d'IBM. SQL est une norme ANSI depuis 1986 et ISO depuis 1987.

La version SQL-2 (appelé aussi SQL-92) est aussi normalisée par ANSI et ISO. Les langages proposés par les SGBDR actuels (DB2, SQL/DS, INGRES, INFORMIX,...) sont proches de cette norme, elle définit trois niveaux de conformité : Le niveau d'entrées, le niveau intermédiaire et le niveau maximum. le langage SQL d'oracle est conforme au premier niveau, il permet de déclarer, de manipuler et d'interroger les données et aussi de contrôler leur accès et d'être intégrées dans les programmes écrits dans un langage quelconque tel que COBOL, C, C++ , JAVA, ....

Avant de présenter les différents langages qu'utilise oracle il est important d'introduire les objets manipulés dans ces langages.

##### 4.4.1. Les objets de base :

Les objets de base peuvent être répartis en 3 catégories : Internes à la base, Internes aux zones mémoire qui inclut les zones de mémoire partagées et les processus d'arrière-plan et externes à la base.

##### 4.4.1.1. Les objets internes :

###### a) Les tables :

Dans Oracle, une table représente l'unité de base utilisée pour stocker des données.

Chaque élément d'information qui est chargé dans la base est placé dans une table. Une table est constituée de colonnes. Chaque colonne reçoit un nom unique dans la table et se voit assigner un type de données (tel que VARCHAR2, DATE, NUMBER, etc.) ainsi qu'une

longueur (qui peut être prédéterminée par le type de données, comme c'est le cas avec DATE).

Afin d'appliquer des règles de gestion aux données d'une table, *Oracle10g* permet d'y associer des déclencheurs et des contraintes d'intégrité qui sont:

- ✓ Chaque colonne d'une table peut être désignée comme étant NULL ou NOT NULL. Lorsque la contrainte NOT NULL s'applique à une colonne, tous les enregistrements insérés dans la table doivent obligatoirement contenir une valeur valide pour cette colonne.
- ✓ La clé primaire d'une table est la colonne (ou les colonnes) qui garantit l'unicité de chaque ligne, une colonne de clé primaire est ainsi définie comme étant NOT NULL.
- ✓ Une colonne doit avoir une contrainte DEFAULT qui génère une valeur par défaut qui lui est insérée lorsqu'une ligne ne contient pas de valeur pour cette colonne.
- ✓ La contrainte CHECK garantit que les valeurs d'une colonne spécifique satisfont certains critères.
- ✓ La contrainte UNIQUE garantit l'unicité des colonnes qui doivent contenir une valeur qui ne fait pas partie de la clé primaire.
- ✓ La contrainte de clé étrangère spécifie la nature de la relation qui existe entre des tables.

Dans une table, une clé étrangère se réfère à une clé primaire préalablement définie ailleurs dans la base.

#### **b) Les vues :**

Une vue est une présentation personnalisée des données contenues dans une ou plusieurs tables, appelées tables de base. Une vue apparaît comme étant une table qui contient des colonnes et qui peut être interrogée de la même manière qu'une table, à la seule différence qu'elle ne contient pas de données mais elle stocke seulement une instruction SQL.

Lorsqu'un utilisateur exécute une requête qui accède à une vue, Oracle interroge le dictionnaire de données (étant donné qu'une vue, qui inclut la requête sur laquelle elle se fonde, les colonnes qu'elle utilise et les propriétés accordées, est stockée dans le dictionnaire de données), extrait l'instruction SQL stockée correspondante, puis l'exécute. Les données renvoyées par la requête sont présentées comme si elles avaient été récupérées directement sans passer par la vue.

En fait, à moins de savoir qu'il accède à des données par l'intermédiaire d'une vue, l'utilisateur pense qu'il interagit directement avec la ou les tables sous-jacentes, d'autant que,

à l'instar d'une table, une vue supporte les opérations d'insertion, de mise à jour, de suppression et de sélection de données. Tous les changements effectués au moyen de la vue sont reportés dans les tables de base.

Les vues ont plusieurs utilités dont on peut citer :

- ✓ Limiter les accès aux données pour les utilisateurs. Prenons le cas d'une table qui stocke des informations sur les salariés d'une entreprise. On peut définir une vue pour faire en sorte que certains employés du service du personnel accèdent uniquement aux données contractuelles contenues dans cette table, excluant ainsi les informations du salaire et tout autre enregistrement d'ordre personnel.
- ✓ Servir à simplifier l'exécution de données et dissimuler la complexité d'une base de données relationnelle. Par exemple, on peut octroyer à des utilisateurs l'accès à une vue dont l'instruction SQL joint plusieurs tables.

#### **c) Les vues matérialisées :**

Contrairement à une vue classique, qui contient seulement une instruction SQL, une vue matérialisée (*materialized view*) inclut les lignes de données qui résultent d'une requête SQL portant sur une ou plusieurs tables.

#### **d) Les séquences :**

Les séquences sont utilisées pour simplifier les tâches de programmation, elles représentent un moyen très efficace pour générer des séries de numéros séquentiels uniques pouvant servir notamment de clé primaire. Les définitions de séquence sont stockées dans le dictionnaire de données.

#### **e) Les synonymes :**

Un synonyme est un autre nom pour une table, une vue, une séquence ou une unité de programme. Les synonymes sont employés généralement dans les situations suivantes :

- ✓ Pour dissimuler le nom du propriétaire d'un objet de base de données ;
- ✓ Pour masquer l'emplacement d'un objet de base de données dans un environnement distribué ;
- ✓ Pour pouvoir se référer à un objet en utilisant un nom plus simple. Un synonyme peut être :
- ✓ Privé, donc il sera accessible uniquement à son propriétaire ainsi qu'aux utilisateurs auxquels ce dernier a accordé une permission.

- ✓ Public, c'est à dire qu'il est disponible pour tous les utilisateurs de la base.

**f) Les index :**

Un index est une structure de base de données utilisée par le serveur pour localiser rapidement une ligne dans une table. Un index étant une copie réduite d'une table, récupérer des données par ce moyen est donc beaucoup plus rapide que d'effectuer une recherche directement dans la table.

**g) Les déclencheurs :**

Les déclencheurs sont des procédures stockées dans la base de données qui s'exécutent implicitement lorsqu'un évènement de base de données survient. Ils peuvent être écrits en PL/SQL, en Java ou encore en C. Ils sont généralement employés à des fins d'audits et de sécurité, mais peuvent aussi servir à empêcher l'exécution de transactions invalides, à faire appliquer des règles de gestion et même à maintenir des tables répliquées.

Il existe deux types de déclencheur :

- ✓ Déclencheur de niveau instructions: est activé une seule fois pour chaque instruction déclenchant.
- ✓ Déclencheur de niveau ligne: est activé pour chaque ligne concernée par une instruction déclenchant.

Pour chaque type de déclencheur, on peut spécifier s'il sera activé avant ou après l'évènement. Les évènements déclenchant incluent des opérations LMD (Langage de Manipulation de Données) telles qu'INSERT (insérer), UPDATE (mise à jour), DELETE (supprimer), et des opérations LDD (Langage de Définition de Données) telles que CREATE (créer), ALTER (modifier), DROP (supprimer). Depuis la version 8 d'Oracle, on peut créer des déclencheurs sur des évènements du niveau système.

**h) Les privilèges et les rôles :**

Pour qu'un utilisateur puisse accéder à un niveau appartenant à un autre utilisateur, il doit avoir reçu le privilège d'y accéder. En général, les vues reçoivent des privilèges de sélection, d'exécution de fonction ou de package, ou encore d'insertion, de mise à jour et de suppression de ligne d'une vue, d'une table ou d'une vue matérialisée. Ainsi, on peut accorder des privilèges à des utilisateurs individuels ou à un groupe public auquel tous les utilisateurs de la base appartiennent.

L'octroi de privilèges devient vraiment ardu, quand on sait que parfois plusieurs utilisateurs doivent accéder à l'application simultanément. Ainsi est apparu la notion de rôle.

Un rôle sert à regrouper des privilèges pour faciliter leur assignation à des utilisateurs et à d'autres rôles. Une fois des privilèges octroyés à un rôle, un utilisateur peut en hériter en devenant membre de ce rôle.

**i) Les fonctions :**

Une fonction est un bloc de code stocké dans le dictionnaire de données, elle est capable de retourner des valeurs au programme appelant. On peut créer des fonctions et les appeler dans des instructions SQL de la même manière qu'on exécute celles fournies par Oracle.

**j) Les procédures :**

Une procédure est un bloc de code stocké dans le dictionnaire de données, mais contrairement à une fonction, une procédure ne retourne pas de valeur au programme appelant.

**k) Les paquetages (les packages) :**

Un paquetage est formé de deux parties, une spécification et un corps, qui servent respectivement à déclarer et à définir toutes les structures qu'il utilise. Les fonctions et les procédures d'un package ne sont pas stockées individuellement dans le dictionnaire de données, mais comme faisant partie du même package. Ils sont très utiles dans les tâches d'administration relatives à la gestion des sous-programmes.

**l) Les schémas :**

L'ensemble des objets qui appartiennent à un compte utilisateur est désigné par le terme schéma.

**4.4.1.2. Objets internes aux zones de mémoire :****a) Utilisation de la mémoire par oracle :**

Une SGA (*System Global Area*) contient les structures de la mémoire d'une instance Oracle, ainsi que les données et les informations de contrôle pour le serveur Oracle. Une SGA est obligatoirement allouée à chaque démarrage d'une instance et est libérée lorsque l'instance est arrêtée. Tous les utilisateurs qui sont connectés à la base Oracle 10g partagent la même zone SGA. Cette zone contient plusieurs structures de mémoire dont nous citons :

**✓ Le pool partagé (*shared pool*)**

Il contient le cache bibliothèque (library cache) qui maintient des informations sur les instructions exécutées dans la base données, et autorise le partage des instructions SQL qui ont été exécutées, et contient aussi le cache du dictionnaire de données composé de tables, dans lequel sont stockées les informations relatives aux objets de la base de données (les descriptions des tables et les privilèges, les noms des comptes utilisateurs, les noms des fichiers de données,...). Lorsqu'une table a besoin des informations précitées, elle lit les tables du dictionnaire.

**✓ Le buffer cache des données**

Il est utilisé pour stocker les données les plus récemment utilisées.

**✓ Le buffer *redo log***

Utilisé pour enregistrer les modifications apportées à la base à l'aide de l'instance.

**✓ Instance Oracle**

Le serveur Oracle se compose d'une SGA et de processus en arrière-plan utilisés par un serveur Oracle pour gérer une base de données. Une instance Oracle, identifiée par ORACLE\_SID au niveau du système d'exploitation, peut ouvrir et utiliser une seule base de données à tout moment.

**b) Les processus :**

Le fonctionnement de la base Oracle est régi par un certain nombre de processus chargés en mémoire permettant d'assurer la gestion de la base de données ; on distingue :

**✓ Processus utilisateur :**

Oracle crée un processus utilisateur qui est exécuté sur la machine à laquelle l'utilisateur est directement connecté afin que ce même utilisateur puisse exécuter un programme applicatif comme Oracle Enterprise Manager (OEM) ou Server Manager. Le processus utilisateur démarre lorsque l'outil démarre et se termine quand l'utilisateur quitte la session en cours.

**✓ Processus serveur :**

Le processus serveur se distingue par les caractéristiques suivantes:

- Il tourne sur la même machine que le serveur Oracle;
- Il est reconnu serveur dédié dans la plus simple des configurations. Chaque processus serveur prend en charge un seul processus utilisateur, et il est créé lorsque l'utilisateur se connecte;

- Chaque processus serveur utilise une zone de mémoire nommée Zone Mémoire du Programme (*Program Global Area: PGA*);
- Le processus serveur utilise l'*Oracle Program Interface (OPI)* qui sert de moyen de communication avec le serveur Oracle à la demande du processus utilisateur;
- Le processus serveur renvoie les résultats au processus utilisateur.

✓ **Processus en arrière-plan :**

Les processus en arrière-plan sont essentiels au fonctionnement d'Oracle 9i. Dans une instance Oracle, ils effectuent les fonctions courantes nécessaires au traitement simultané de plusieurs requêtes utilisateurs et interagissent avec différentes structures de la base de données, sans compromettre l'intégrité et la performance de l'ensemble du système. Plusieurs processus en arrière-plan supportent le fonctionnement d'une instance Oracle 9i.

✓ ***DBWn (Data Base Writer) :***

Le processus DBWn (écriture dans la base de données) écrit dans les fichiers de données sur disque les blocs de données modifiés qui se trouvent dans le cache de données. Il existe un processus DBWn par défaut, mais on peut demander à *Oracle 10g* d'en générer jusqu'à neuf autres (ils sont numérotés de DBW0 à DBW9). Lorsqu'un processus utilisateur a besoin d'accéder à des données qui ne se trouvent pas dans le cache tampon (buffer cache), DBWn permet à la session de disposer d'espace libre dans le cache pour effectuer son traitement. De tous les processus d'arrière-plan, c'est le seul autorisé à écrire des données dans la base.

✓ ***PMON (Process Monitor) :***

Le processus PMON (surveillance de processus) assure la récupération de processus utilisateur et la libération des ressources qu'il exploitait lorsqu'il ne se termine pas correctement et n'est pas donc en mesure de le faire lui-même. Ce genre de situation se produit, par exemple, lorsqu'un utilisateur met fin à une session en appuyant sur Ctrl+Alt+Supp.

✓ ***SMON (System Monitor)***

Le processus SMON (surveillance du système) effectue la récupération d'une instance défaillante lors de son redémarrage. Il peut être activé par d'autres processus en arrière-plan s'ils estiment que celui-ci doit intervenir. Autrement, il s'active de lui-même régulièrement pour voir s'il a du travail. Ce processus est essentiel au bon fonctionnement de n'importe quelle instance Oracle10g.

**✓ LGWR (Log Writer)**

Le processus LGWR (écriture du journal de reprise) est chargé de gérer le tampon du journal de reprise (redo log buffer). Lorsqu'une transaction modifie des données dans la base, ces changements sont consignés en premier lieu sous forme d'entrées dans le tampon redo log et ces entrées sont écrites sur disque. LGWR peut enregistrer sur disque en une seule fois les entrées de plusieurs transactions validées, optimisant ainsi ses performances et limitant les accès au disque.

**✓ CKPT (Check Point)**

Le processus CKPT (point de contrôle) est chargé de signaler au processus DBWn à quel moment écrire sur disque, et aussi de mettre à jour tous les fichiers de données ou de contrôle de la base pour indiquer le point de contrôle le plus récent.

**✓ RECO (Recoverer)**

Le processus RECO (processus de récupération) résout les défaillances dans les configurations de bases de données dans un environnement Oracle 10g distribué, en tentant d'accéder aux bases impliquées dans la transaction distribuée.

**✓ ARCn (Archiver)**

Ce processus effectue une copie automatique du journal redo log qu'il stocke dans un emplacement spécifié par l'administrateur de base de données (DBA) lorsque la base de données fonctionne dans le mode ARCHIVLOG. Il s'agit d'un processus optionnel qui est généré quand le paramètre d'initialisation LOG\_ARCHIVE\_START possède la valeur TRUE.

**4.4.1.3. Objets externes :**

Les fichiers physiques d'une base oracle permettant de stocker de manière persistante les données manipulées par oracle, tandis que la mémoire sert à optimiser la vitesse de fonctionnement de la base de données. On distingue :

**✓ Le fichier *init.ora***

Chaque base de données doit posséder un fichier init.ora associé. Lorsqu'une commande est exécutée pour démarrer la base, les informations contenues dans ce fichier sont examinées afin de déterminer les ressources qui peuvent lui être allouées. Si ce fichier n'inclut aucune ressource particulière, Oracle utilise la configuration par défaut.

**✓ Les fichiers d'une base de données**

Les fichiers composant une base de données contiennent les données utilisateur et les informations supplémentaires nécessaires au fonctionnement des opérations liées à la base de données. Une base de données Oracle se compose des types de fichiers suivants :

**✓ Les fichiers de données**

Ils stockent le dictionnaire de données, les objets utilisateur et les « images avant » de données modifiées par les transactions en cours. Une base de données contient au moins un fichier de données.

**✓ Les fichiers redo log**

Une base de données Oracle 10g possède au minimum deux groupes de fichiers redo log qui forment le journal de reprise (redo log). Ces fichiers contiennent un enregistrement des modifications apportées à la base de données. En cas de défaillance, ils peuvent être utilisés pour assurer une récupération jusqu'au point précédant l'incident. Ce journal garantit que les transactions validées pourront être enregistrées sur disque. Ainsi, aucun travail ne sera perdu.

**✓ Les fichiers de contrôle**

Ils contiennent l'information nécessaire à la mise à jour et à la vérification de l'intégrité des bases de données. Une base de données requiert au moins un fichier de contrôle.

Un serveur Oracle utilise d'autres fichiers outre les fichiers de base de données, parmi lesquels on cite :

**✓ Le fichier des paramètres**

Utilisé pour définir les caractéristiques d'une instance Oracle ;

**✓ Le fichier mot de passe**

Utilisé pour établir l'authenticité des utilisateurs privilégiés de bases de données.

**✓ Le fichier redo log archivé**

C'est une copie du journal de reprise créée et maintenue parallèlement, qui peut être nécessaire à la restauration des données suite à une panne d'un disque. Pour assurer une protection optimale, il importe que la copie du journal soit créée sur une unité de disque séparée.

**4.4.2. Le langage de définition de données (LDD) :**

Il permet la définition (création, modification et suppression) des bases de données et des objets qui les composent. Les principales instructions LDD sont présentées dans le tableau suivant :

Instruction	Description
ALTER PROCEDURE	Recompile une procédure stockée.
ALTER TABLE	Ajoute ou redéfinit une colonne ou bien modifie les paramètres de stockage d'une table existante.
ALTER TABLE ADD CONSTRAINT	Ajoute une contrainte à une table existante.
ANALYZE	Collecte des statistiques de performances sur des objets de base de données pour l'optimiseur fondé coût.
CREATE TABLE	Crée une table.
CREATE INDEX	Crée un index.
DROP TABLE	Supprime une table.
DROP INDEX	Supprime un index.
GRANT	Octroie des privilèges ou des rôles à un utilisateur ou à un autre rôle.
REVOKE	Révoque des privilèges ou des rôles à un utilisateur

Tableau B1. langage de définition de données (LDD)

#### 4.4.3. Le langage de manipulation de données (LMD) :

Dans le langage SQL, les commandes INSERT, UPDATE, DELETE permettent respectivement d'insérer, de modifier et de supprimer des enregistrements d'une table selon des critères définis avec la directive WHERE. Les instructions LMD sont décrites dans le tableau ci-dessous :

Instruction	Description
SELECT	Extrait/sélectionne des données stockées dans une base de données <i>Oracle</i> . Il s'agit de l'instruction <i>SQL</i> la plus utilisée. La création d'un rapport débute toujours par cette instruction.
INSERT	Change/ajoute des lignes de données dans une table.
UPDATE	Met à jour/modifie le contenu d'une table.
DELETE	Supprime des données d'une table.

Tableau B2. langage de manipulation de données (LMD)

#### 4.4.4. Le langage PL/SQL :

##### ✓ Le langage *PL/SQL*

Le langage PL/SQL (*Procedural Language/Structured Query Language*) est le langage de référence d'Oracle depuis la version 6 du SGBD en 1988. Il s'agit d'une extension procédurale du langage SQL. Le PL/SQL permet de combiner des concepts des langages de troisième génération (les structures conditionnelles, les structures répétitives,...) avec ceux des langages non procéduraux de quatrième génération (les requêtes SQL, les procédures et les fichiers stockés,...) dans un même programme.

Il peut être utilisé au sein de différents outils, parmi lesquels nous citons:

- SQL\*PLUS;
- Les outils Developer 2000 (Oracle Forms, Report, Graphics,...);
- PRO\*SQL;

##### ✓ Blocs *PL/SQL*

Dans un environnement Client/serveur, chaque requête SQL donne lieu à l'envoi d'un message du Client vers le Serveur puis du Serveur vers le Client. Il est préférable de travailler avec des blocs PL/SQL plutôt que de lancer successivement une suite d'instructions SQL susceptibles d'encombrer le trafic réseau. En effet, un bloc PL/SQL contenant plusieurs instructions SQL donne lieu à un seul envoi sur le réseau vers le serveur; les résultats intermédiaires sont traités sur place et un résultat unique est retourné au client.

##### ✓ Structure d'un bloc *PL/SQL* Il est composé de trois sections qui sont:

- « DECLARE » (optionnel) ; qui réunit les déclarations des objets de travail tels que les types de données, les Variables,..., cette section est facultative ;
- « BEGIN » jusqu'à « END » : cette section contient toutes les instructions SQL et PL/SQL. Elle est obligatoire et constitue le corps du bloc ;
- « EXCEPTION » (optionnel) : cette section permet la gestion des erreurs captées par le SGBD durant l'exécution des traitements de la section précédente. Comme la première section, celle-là aussi est facultative.

Ainsi, la structure de base de n'importe quel programme PL/SQL aura la forme suivante :

<pre>DECLARE -- Déclarer les variables BEGIN -- Ecrire le code du programme EXCEPTION -- Ecrire le code de gestion des erreurs END ;</pre>
--

✓ **Avantages de PL/SQL :**

- Modularité de développement des programmes ;
- Déclarer des identifiants ;
- Programmer avec des structures de contrôle de langage procédural ;
- Gestion d'erreurs ;
- Portabilité ;
- Intégration ;
- Accroissement des performances ;

**4.4.5. Les outils d'Oracle Forms :**

**4.4.5.1. Le navigateur d'objet « *Objet Navigator* » :**

L'Object Navigator présente les objets de l'application sous forme hiérarchique et permet de créer, de modifier,... des objets etc.

**4.4.5.2. L'éditeur de représentation « *Layout Editor* » :**

Le Layout Editor permet :

- ✓ De concevoir le style, le contenu, la taille et la disposition des objets de l'application ;
- ✓ La création, la suppression et la modification d'objets ;
- ✓ La prévisualisation d'écran ;
- ✓ La synchronisation avec la palette des propriétés et l'Object Navigator ;
- ✓ D'utiliser les fonctions de dessin.

**4.4.5.3. La palette des propriétés :**

Elle permet d'afficher et de modifier les propriétés des objets manipulés.

**4.4.5.4. L'éditeur PL/SQL « *PL/SQL Editor* » :**

L'Editeur PL/SQL permet de :

- ✓ Développer des Triggers, des procédures, des fonctions et des Librairies ;
- ✓ Débugger le code PL/SQL ;
- ✓ Avoir des informations sur le contexte (type, nom,...) ;
- ✓ Visualiser tous les traitements de l'application.

### 5. Oracle Enterprise Manager :

OEM se compose de fonctionnalités utiles à l'administration de base de données (Data Base Administrator), accessible au travers d'une interface graphique GUI (Graphical User Interface), pour interagir avec une base de données Oracle 10g (GUI est utilisée pour faciliter l'interaction avec les utilisateurs).

OEM offre la panoplie d'outils suivante :

- ✓ **Gestionnaire d'instance** : Permet de démarrer ou d'arrêter une base de données Oracle 10g, de gérer les paramètres d'initialisation et de contrôler les sessions utilisateurs lors de leur interaction avec la base de données.
- ✓ **Gestionnaire de schémas** : Permet de travailler avec les différentes catégories d'objets de la base de données.
- ✓ **Gestionnaire de sécurité** : Permet de créer, de supprimer ou de modifier les comptes d'utilisateurs, ainsi que d'attribuer des privilèges spéciaux permettant d'interagir avec la base de données au moyen de certaines fonctionnalités.
- ✓ **Gestionnaire de stockage** : permet d'administrer l'infrastructure d'Oracle 10g, qui comprend entre autres les segments de rollback, les tablespaces et les fichiers redo log en ligne.
- ✓ **Gestionnaire d'environnement de versions (*workspace*)** : Permet de constituer des environnements logiques contenant des tables pour lesquelles la gestion des versions est activée. Dans ce cas, les lignes de ces tables supportent des versions différentes de leurs données. Il est ainsi possible de stocker des versions estampillées des données à des fins de comparaison ou d'analyse de cycles d'exploitation.
- ✓ **Gestionnaire *Data Guard*** : Assume le rôle significatif lors de l'implémentation des fonctions de tolérance aux pannes d'Oracle 10g, telles que les environnements en veille maintenant des copies de bases de données et de journaux de reprise.
- ✓ **Gestionnaire de sauvegarde** : Permet d'accéder à des assistants qui nous guident au travers des opérations de sauvegarde et de récupération d'une base de données Oracle 10g.

- ✓ **Gestionnaire de données** : Offre un accès graphique à certains outils, comme Export ou Import, que l'on utilise pour le transfert de données d'une base vers une autre, ou pour réaliser des copies des données conservées dans des fichiers linéaires du système d'exploitation.
- ✓ **SQL\*Plus Worksheet** : Cet utilitaire offre une interface graphique améliorée à partir de laquelle on peut travailler avec SQL\*Plus et PL/SQL.

Comme on peut le constater, OEM permet d'assurer une gestion centralisée des activités d'administration d'une base de données Oracle 10g.

Parmi les tâches réalisées par OEM, on peut citer:

- Le démarrage de la base de données ;
- L'arrêt de la base de données ;
- La gestion des tablespaces :
- Le redimensionnement d'un fichier de données ;
- L'ajout d'un fichier de données ;
- La réduction de la taille d'un fichier de données ;
- La gestion des utilisateurs ;
- La création d'un utilisateur ;
- Les droits d'exploitation de l'espace des tablespaces ;
- L'octroi de privilèges sur des objets ;
- La gestion des objets de base de données :
- La maintenance d'objets avec SQL\*Plus Worksheet.

## 6. L'interface SQL\*Plus :

SQL\*PLUS permet aux utilisateurs de créer et de manipuler interactivement des objets de la base avec une interface permettant aux utilisateurs de dialoguer avec la base de données à travers le SGBD Oracle. SQL\*PLUS est habituellement utilisé pour formuler une requête SQL et obtenir, sur un écran et de façon immédiate, le résultat attendu.

Outre son utilisation interactive du langage SQL par la formulation de requêtes ponctuelle, SQL\*Plus est un utilitaire puissant offrant d'autres possibilités intéressantes, notamment :

- Le formatage des résultats ;
- L'obtention d'aide en ligne ;
- Le contrôle et l'édition des commandes ;

- La gestion des transactions ;
- Un éditeur de ligne intégré peut être utilisé pour éditer et corriger les requêtes SQL, il est possible d'utiliser tout autre éditeur installé sur l'ordinateur.
- Paramétrage de l'environnement de travail : longueur de ligne, validation automatique.
- Une utilisation de programmes structurés grâce au langage évolué oracle PL/SQL.
- Une mémorisation des commandes SQL, SQL\*Plus, PL/SQL.

### **7. L'administration d'Oracle 10g :**

L'administration effective d'une base de données consiste tout simplement dans la garantie de la stabilité de l'environnement autour de cette base et de la stabilité technique et logique des données qui y sont hébergées.

#### **7.1. Rôle de l'administrateur :**

L'administrateur d'une base de données consiste à définir les objets qu'elle contient et à veiller à leur bonne utilisation. L'administrateur a un double rôle :

- ✓ Un rôle organisationnel qui concerne la définition du schéma conceptuel des données et le partage de ces données par les utilisateurs.
- ✓ Un rôle technique qui consiste à mettre en œuvre ce schéma et ce partage à l'aide des capacités techniques du SGBD, on peut citer comme rôle technique : installation du SGBD et des outils associés, création de la base de données, assurer son évolution et la gestion de privilèges d'accès.

#### **7.2. Outils d'administration de données :**

Les principales tâches d'administration de données sont effectuées en utilisant l'utilitaire SQL\*DBA. On peut aussi utiliser SQL\*loader et IMPORT / EXPORT malgré que ces deux derniers ne sont pas spécifiés à l'administration de données.

#### **7.3. Vue globale de SQL\*DBA :**

SQL\*DBA est l'outil de base pour l'administration de données oracle. Il permet de :

- ✓ Démarrer et arrêter une instance ;
- ✓ Charger /décharger, ouvrir / fermer une base de données ;
- ✓ Sauvegarder et restaurer des données et des journaux ;

- ✓ Exécuter toutes les commandes SQL, PL/SQL.

## 8. Dictionnaire de données :

Le dictionnaire de données d'oracle est composé d'un ensemble de tables et vues. Les tables servant à l'enregistrement des informations relatives aux différents objets de la base de données. Les tables et les vues appartiennent à l'utilisateur qui les a créé.

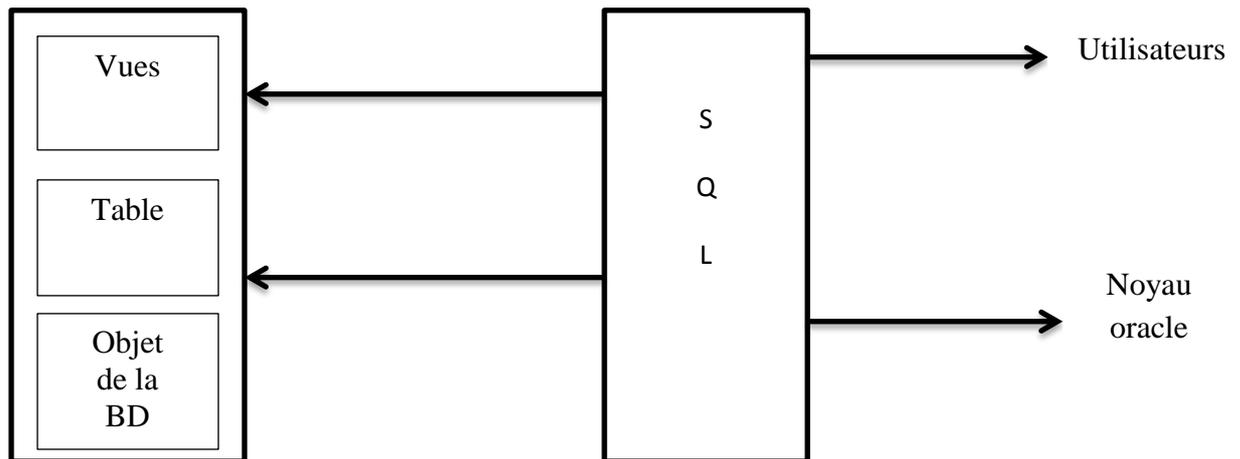


Figure. B1 : Dictionnaire de données d'oracle.

## 9. L'informatique distribuées et Oracle 10g :

Un environnement informatique distribué peut s'appuyer sur le concept de base de données répartie, où des nœuds autonomes participent au système pour permettre aux utilisateurs d'accéder aux données locales ou distantes, ou sur le concept de réplication.

### 9.1. Oracle Net :

Oracle Net est l'épine dorsale de la solution informatique distribuée d'Oracle 10g. Les deux principaux fichiers de configuration utilisés par Oracle Net sont :

- ✓ **Le fichier listener.ora** : considéré comme le coeur d'Oracle Net, il décrit l'environnement dans lequel les requêtes de connexion distantes seront traitées. Un listener est un processus serveur chargé de détecter les requêtes de connexion à un service. Lorsqu'un client émet une requête de connexion à une base de données Oracle, le listener actif sur le serveur intercepte la demande et fait le nécessaire pour mettre en relation le processus utilisateur avec un processus serveur de l'instance

Oracle 10g qui est associée à la base de données sollicitée. Le processus serveur peut être généré à l'occasion ou être déjà existant. Le fichier de configuration *listener.ora* réside sur le serveur sur lequel la base de données Oracle 10g est exécutée.

- ✓ **Le fichier *tnsnames.ora*** : Ce fichier contient les références (liste des alias Oracle Net) permettant d'accéder aux bases de données Oracle 10g au moyen d'Oracle Net. Les entrées du fichier *tnsnames.ora* ont reçu de nombreuses appellations. Lorsque SQL\*NET version 2 a été introduit, aux alentours de 1992, Oracle appelait les premières entrées de ce fichier des descripteurs de service. Certains les nomment les alias Net. Ce fichier doit être placé sur le client qui doit accéder à la base. Qu'on utilise une architecture Client/serveur 2-tiers ou 3-tiers, ce fichier doit être accessible à chacun des clients.

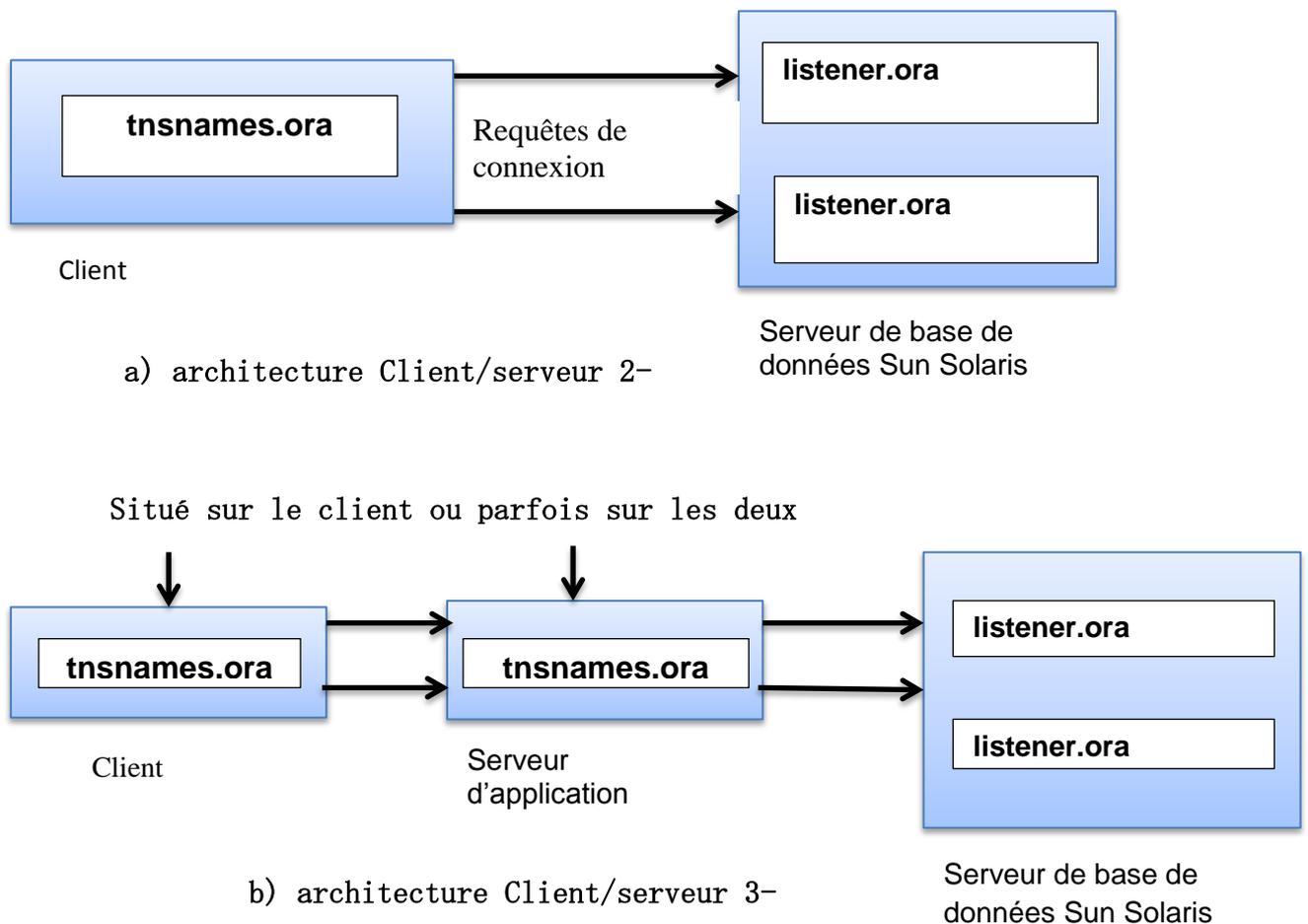


Figure B2 : Emplacement des fichiers listener.ora et

**10. Le serveur de données :**

Oracle a fait sa réputation comme serveur de données sur multiples plates-formes, toutes les applications classiques nécessitant une base de données peuvent être mises en œuvre avec Oracle 10g et Windows. Par sa facilité d'installation et de paramétrage, un serveur Windows relié à des postes Windows ou autres, permet de constituer facilement une configuration

Client/serveur à faible coût.

**11. Le i d'Oracle 10g (internet) :**

L'emploi de la base de données Oracle en tant que plate-forme pour le développement d'applications Internet a commencé avec Oracle 8i. Cette version a intégré un moteur Internet dans la base de données, plaçant ainsi le cœur des applications au niveau de celle-ci. En procédant ainsi, Oracle a proposé une solution évolutive. Lorsque la société Oracle a incorporé

Java pour la première fois dans la base de données (version 8.1.5), elle a introduit un système baptisé JServer. Celui-ci prenait en charge Java à partir de la base de données, démultipliant ainsi l'efficacité de la gestion de la sécurité et de la mémoire par cette manière. Lorsque la version suivante de la base de données a été publiée (8.1.6), JServer a été remplacé par la machine virtuelle Java (Java Virtual Machine, JVM) ; avec Oracle 10g celle-ci a été renommée EJE (Entreprise Java Engine). EJE possède une architecture complexe qui lui permet de supporter de nombreuses méthodes d'accès, des types de clients différents et différents niveaux d'accès utilisateurs. Ainsi, avec Oracle 10g, une procédure externe écrite en C, C++ ou Java peut accéder à une base de données via des protocoles ouverts tels que HTTP ou IIOP basés sur TCP/IP pour assurer les différentes implémentations de CORBA et DCOM. Oracle 10g a donc intégré la technologie Java dans toute son architecture, côté Client avec les Java Beans et côté Serveur d'application et de données avec des composants serveurs Entreprise Java Beans.

**12. Les nouveautés d'oracle 10g :**

Les nouveautés apportées à *Oracle 10g* sont plus qu'une simple évaluation, la gestion interne de la base a été revue intégralement. Pour simplifier l'installation et l'administration de toutes ces nouvelles options, de nombreux assistants sont apparus.

**✓ Le moteur d'Oracle 10g**

Oracle a peut communiquer sur les nouveautés techniques apportées au moteur Oracle10g (qui sont nombreuses). Elles vont dans de multiples directions et concernent tous les aspects de la base de données. Dans ce qui suit, nous listerons quelques-unes de ces nouveautés :

- **Nouveau mode de gestion des tablespaces** : Pour simplifier et améliorer les performances d'accès aux tablespaces, deux modes de gestion existent : le mode Local qui est un nouveau mode, stocke tous les aspects d'allocation de segments à l'intérieur de chaque tablespace, et l'ancien mode qui est le mode Dictionary centralise l'allocation d'espace dans le dictionnaire de données de chaque base.
- **L'exploration des fichiers redo log (Log Miner)** : Oracle propose un outil d'analyse du contenu des fichiers redo log d'une base : le Log Miner. Cet utilitaire analyse de manière séquentielle l'ensemble des opérations réalisées sur une base.
- **La gestion automatique des fichiers par Oracle** : La gestion des fichiers proposés par Oracle offre de multiples options pour faciliter l'administration (accroissement automatique de leur taille, gestion interne, etc.), ainsi que la gestion de l'ensemble des fichiers composants une base.
- **Contrôle dynamique de la mémoire SGA** : Les paramètres d'initialisation qui dimensionnent l'allocation mémoire d'une instance sont dynamiques. Le paramètre dynamique `BD_CACHE_SIZE` remplace l'ancien `BD_BLOCK_BUFFER` qui était statique.
- **L'utilisation d'un fichier mot de passe** : Pour démarrer une base en mode Client/serveur ou à partir d'Oracle Enterprise Manager, Oracle recommande l'utilisation d'un fichier mot de passe. Il autorise les utilisateurs possédant les privilèges `SYSOPER` et `SYSDBA` à administrer une base à distance.

**13. Conclusion**

Cette partie nous a permis de prendre connaissance des produits d'oracle corporation, de voir la puissance de son SGBD et de nous initier dans le détail aux différents langage et technologie de programmation à utiliser pour lancer le développement et l'implémentation des différents composants de l'application (base de données, interfaces et application client/serveur ...) proposé.

# *Conclusion*

## *Général*

---

## ***CONCLUSION GÉNÉRAL***

---

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

La réalisation de ce travail nous a permis d'approfondir nos connaissances en systèmes d'information des entreprises, de découvrir le domaine de gestion des ressources humaines, de voir sur le champ de travail les différentes tâches effectuées par les gestionnaires de ressources humaines. En plus de l'expérience, ce projet à travers les étapes de mise en œuvre notamment l'analyse, la conception et la réalisation, il nous a conduit à l'utilisation des outils de conception « UML » et de réalisation « produit ORACLE » avec tous ses outils et fonctions et de se familiariser avec eux.

L'application à laquelle nous avons abouti, traite la gestion des ressources humaines et qui adopte l'architecture Client/serveur offre aux différents utilisateurs plus de souplesse et d'efficacité dans l'exécution de leurs tâches.

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

# *References*



## Références

- [1] : Olivier GLÜCK Université LYON 1/UFR d'Informatique  
[Olivier.Gluck@enslyon.fr](mailto:Olivier.Gluck@enslyon.fr) <http://www710.univ-lyon1.fr/~ogluck>.
- [2] : livre Cisco
- [3] : Les réseaux Edition 2008 par Goylujolle avec la colaboration de Oliser Selvatori et la contribution de Jacque Nozick
- [4] <http://www.commentcamarche.net/contents/cs/cs3tier.php3>.
- [5] : <http://www.commentcamarche.net/contents/oracle/oracproc.php3>.
- [6] : thèse ingénieur, université Mouloud MAMMERI Tizi-Ouzou,
- [7] [www.Oracle.com](http://www.Oracle.com)
- [8] [www.développeur.com](http://www.développeur.com)
- [9] [www.labo-oracle.com](http://www.labo-oracle.com)
- [10] [www.netbeans.org](http://www.netbeans.org)
- [11] : thèse ingénieur, université Mouloud MAMMERI Tizi-Ouzou,  
conception et réalisation d'une application client-serveur 3tiers sous oracle  
cas : gestion de la comptabilité général de l'ENIEM.,réalisé par NAIT BELAID Ouerdia et HABI Fetta.