

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

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU



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

Mémoire de Fin d'Etudes de MASTER PROFESSIONNALISANT

Domaine : **Mathématiques et Informatique**

Filière : **Informatique**

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

Présenté par
Azzedine AICI
Soria BOUSNADJI

Thème

**Analyse, Conception et réalisation d'une
application java EE
Cas: La Gestion des ressources
humaines GCB.**

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

Président : M^r Dib

Encadreur : M^r Kerbiche

Examineur : M^{me} Yesli

Examineur : M^r Chebouba

Année universitaire 2015-2016

Remerciements

Nous tenons à témoigner notre reconnaissance à DIEU tout puissant, qui nous a aidé et béni par sa volonté pour réaliser ce travail.

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

Nos plus vifs remerciements vont à tout le personnel de GCB qui nous ont généreusement aidé durant notre stage en particulier à notre encadreur Mr FENNOUH.

Nos remerciements vont également aux membres du jury pour l'honneur qu'ils nous font en acceptant d'examiner et d juger notre travail.

Nous remercions aussi à tous ceux, et celles qui ont contribué de près ou de loin pour l'accomplissement de ce modeste travail.

Dédicace

Je dédie ce modeste travail à mes chers parents

A mon cher et unique frère Cherif,

A mes sœurs Soumia et Annie,

A ma grand mère,

A tous les gens qui m'aiment et à ceux qui me sont chers.

Soria

Dédicace

Je dédie ce modeste travail

*Tous d'abord et avant tous à mes chers parents qui ont veillé
sur moi pour que je me retrouve là où
je suis aujourd'hui.*

A mon petit frère Yassine,

A mes sœurs Kahina et Faiza,

*A tous les gens qui m'aiment et à
ceux qui m'ont aidé de près ou de loin.*

Azzedine

Sommaire

Sommaire

Introduction générale	02
------------------------------------	-----------

Chapitre I: Présentation de la technologie Java EE

Introduction.....	05
I.1.Java EE (Java Entreprise Edition).....	05
I.2. Composants et architecture de JAVA EE	05
I.2.1. Les composants clients ou tiers client.....	07
I.2.1.1. Les clients Web	07
I.2.1.2. Les clients Applets	07
I.2.1.3. Les application client.....	08
I.2.2. Les composants Web (ou tiers Web)	08
I.2.3. Les composants métier ou tiers métier	08
I.3. Les conteneurs Java EE.....	10
I.4. API et services Java EE	12
I.4.1. Java Servlet.....	12
I.4.2. JavaServer Pages (JSP)	13
I.4.3. JavaServer Pages Standar Tag Library (JSTL)	13
I.4.4. JavaServer Faces (JSF)	13
I.4.5. Java DataBase Connectivity (JDBC)	13
I.4.6. Java Persistenc API (JPA)	14
I.4.7. Java Transaction API (JTA)	14
I.4.8. Entreprise JavaBeans (EJB)	14
I.4.9. Java Message Service API (JMS)	15
I.4.10. JavaMail	15
I.4.11.Java Autorization Service Provider Contract For Contrainers (JASPIC)	15

I.4.12.Java Authentification Service Provider Interface For Contrainers (JASPIC)	15
I.4.13.Java Authentification and Autorisation Service (JAAS)	15
I.4.14.Java Naming and Directory Interface (JNDI)	16
I.5.Les avantages d'utiliser JAVA EE.....	16
I.6.Architecture client/serveur.....	16
I.6.1.Définition de l'architecture client/serveur	16
I.6.2.Principe de fonctionnement.....	17
I.6.3.Les concepts client/serveur	17
I.6.3.1.Le concept client	17
I.6.3.2.Le concept serveur.....	18
I.6.4.Classification de l'architecture client/serveur.....	19
I.6.5.Les avantages de l'architecture client/serveur	21
I.6.6.Les inconvénients de l'architecture client/serveur	22
Conclusion.....	22

Chapitre II : Présentation de l'organisme d'accueil

Introduction.....	24
II.1.Présentation générale de l'entreprise	24
II.2.Historique	25
II.3.Implémentation de GCB	25
II.4.Les activités de GCB	26
II.5.Organisation de GCB.....	29
II.5.1.La direction générale (DG)	29
II.5.2.Directions centrales de soutien	30
II.5.3.Directions techniques	30
II.5.4.Directions Régionaux.....	30
II.6.Direction des ressources humaines.....	32

II.6.1.Organigramme de direction des ressources humaines	33
II.6.2.Missions de chaque département	33
II.6.2.1.Département de développement RH.....	33
II.6.2.2.Département Gestion et Control	34
II.6.2.3.Département Relations de travail.....	35
II.7.Présentation de Département Formation	36
II.7.1. Définition et terminologie.....	37
II.7.2. Mission de département formation.....	38
II.7.3.Organigramme de département formation	39
II.4.Service planification et suivie de la formation continue.....	39
II.7.5.Service gestion de la formation continue et de l'apprentissage	40
II.7.6. La problématique	41
II.7.7. Solutions	41
Conclusion.....	42

Chapitre III : Analyse & Conception

Introduction.....	44
III.1. Présentation de la méthode RUP	44
III.1.1. Le cycle de vie du RUP	44
III.2. Phase d'Initialisation.....	45
III.3. Phase d'Elaboration.....	45
III.3.1. Identification des acteurs.....	45
III.3.2 Diagramme de contexte	46
III.3.3. Représentation de diagramme de cas d'utilisation	47
III.3.4 Définition des itérations.....	49
III.4. Construction.....	49
III.4.1. Réalisation de l'itération 1	49
III.4.1.1. Diagrammes de séquence pour l'itération 1	49

III.4.1.2. Diagrammes de classe pour l'itération 1	52
III.4.2. Réalisation de l'itération 2	53
III.4.2.1. Diagrammes de séquence pour l'itération 2	53
III.4.2.2. Diagrammes de classe pour l'itération 2	54
III.4.3 Réalisation de l'itération 3	55
III.4.3.1 Diagrammes de séquence pour itération 3	55
III.4.3.1 Diagrammes de séquence pour itération 3	59
Conclusion.....	60

Chapitre IV : Réalisation

Introduction.....	62
IV.1. Le modèle relationnel	62
IV.2. Description des outils de développement.....	67
IV.2.1. Langage de programmation Java.....	67
IV.2.2. Java EE	67
IV.2.3. IDE (NETBEANS)	67
IV.2.4. Le serveur d'application (Glassfish)	68
IV.3. Les Langages du web	72
IV.3.1. Le coté client	72
IV.3.2. Le coté serveur	72
IV.4.Présentation de quelques interfaces de l'application.....	73
Conclusion.....	79
Conclusion générale	81

Annexe

Bibliographie

Figures et Tableaux

Liste des figures et Tableaux

Les Figures

Chapitre I : Présentation de la technologie JAVA EE.

Figure I.1 : Architecture Java EE standar	06
Figure I.2 : Communication Java EE entre différents tiers	09
Figure I.3 : Serveur Java EE et Conteneurs.....	11
Figure I.4 : Architecture client/serveur	17
Figure I.5 : Les types de serveur	19
Figure I.6 : Architecture client/serveur à 2-tiers	19
Figure I.7 : Architecture client/serveur à 3-tiers	20

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

Figure II.1 : Le logo de l'entreprise GCB	24
Figure II.2 : Implantation de GCB en Algérie.....	26
Figure II.3 : Organigramme générale de GCB	31
Figure II.4 : Organigramme de directions de ressources humaines.....	33
Figure II.5 : Organigramme de département de formation.....	39

Chapitre III : Analyse et Conception.

Figure III.1 : diagramme de contexte	46
Figure III.2: Diagramme de cas d'utilisation globale.....	48
Figure III.3 : Diagramme de séquence du cas d'utilisation:«Authentification»..	50
Figure III.4 : Diagramme de séquence du cas d'utilisation : « Modification de mot de passe »	51
Figure III.5 : Diagramme de classes pour l'itération 1	52
Figure III.6 : Diagramme de séquence du cas d'utilisation:«Ajouter utilisateur»	53

Liste des figures et Tableaux

Figure III.7: Diagramme de classes pour l'itération 2.....	54
Figure III.8 : Diagramme de séquence du cas d'utilisation : «Exprimer un besoin d'une formation»	55
Figure III.9 : Diagramme de séquence du cas d'utilisation : «Evaluer une formation»	57
Figure III.10: Diagramme de classes pour l'itération3	59

Chapitre IV : Réalisation.

Figure IV.1 : Interface de Netbeans	68
Figure IV.2 : Interface GlassFish.....	69
Figure IV.3 : Fenêtre de WampServer	70
Figure IV.4 : Interface PhpMyAdmin	71
Figure IV.5 : Architecture d'une servlet.....	73
Figure IV.6 : Interface page d'accueil.....	74
Figure IV.7 : Interface d'authentification	75
Figure IV.8: Interface Espace chargé de formation	76
Figure IV.9 : Interface du formulaire ajout de formation	77
Figure IV.10 : Interface du formulaire évaluation d'une formation.....	78
Figure IV.11 : Interface du formulaire ajout plan de formation	79

Liste des figures et Tableaux

Les Tableaux

Chapitre I : Présentation de la technologie JAVA EE.

Tableau I.1 : Comparaison de deux modèles (2 tiers et 3 tiers)	21
--	----

Chapitre III : Analyse et Conception.

Tableau III.1 : Description du cas d'utilisation « Authentification »	51
Tableau III.2: Description du cas d'utilisation « Modification de mot de passe »	52
Tableau III.3 : Description du cas d'utilisation : « Ajouter utilisateurs »	54
Tableau III.4 : Description du cas d'utilisation : « Exprimer un besoin d'une formation »	56
Tableau III.5 : Description du cas d'utilisation « Evaluer une formation »	58

Chapitre IV : Réalisation.

Tableau IV.1 : Table Utilisateur.....	63
Tableau IV.2 : Table Participant	63
Tableau IV.3 : Table Directeur_Structure.....	64
Tableau IV.4 : Table Formation.....	64
Tableau IV.5 : Table Plan_Formation.....	65
Tableau IV.6 : Table Evaluation_Froid	65
Tableau IV.7 : Table Evaluation_Chaud	66

Introduction Générale

Introduction générale

Depuis l'apparition de l'informatique, l'être humain a toujours essayé d'exploiter cette science pour automatiser ses tâches quotidiennes de gestion, de communication, de vente...ect

L'arrivée de l'internet marque un nouveau tournant dans l'ère de la communication, c'est une amélioration des méthodes traditionnelles dans le sens qu'il n'est pas géographiquement limité et donne la possibilité de communiquer à l'échelle mondiale à tout utilisateur d'ordinateur. Notamment avec l'apparition du web, le service le plus populaire de l'Internet qui ouvre grand les portes aux grand public sans tenir compte du niveau intellectuel, ni du statut.

Lorsqu'une entreprise possède un grand nombre d'employés, il devient délicat de suivre toutes les opérations faites sur la gestion des ressources humaines. C'est pourquoi il existe des solutions automatisées pour les gérer. Et Comme toute organisation, l'Entreprise National de Génie Civil et Bâtiment (GCB) travaille sans relâche pour mettre en œuvre cette nouvelle technologie de traitement, pour assurer une gestion plus fiable, plus rigoureuse, moins fastidieuse et pour minimiser les risques d'erreurs et assurer la disponibilité de l'information à toute éventuelle demande.

C'est dans ce cadre qui s'inscrit notre étude, qui consiste à la mise en place d'un système d'information pour la gestion des formations existantes dans l'entreprise, Pratiquement, il s'agit de développer une application client/serveur. Pour cela, nous avons utilisé la technologie Java EE.

Introduction générale

Ce mémoire est composé de quatre chapitres, le premier chapitre représente quelque fonctionnement de la technologie Java EE et l'architecture client/serveur, le deuxième chapitre donne une vue globale sur l'entreprise d'accueil ainsi une brève aidée sur notre domaine d'étude et les deux derniers chapitre sont consacrés pour la présentation de notre application et les outils utilisés pour sa réalisation. Nous les détaillerons comme suit :

Chapitre I : Intitulé Présentation de la technologie Java EE, dans ce chapitre nous avons présenté les notions de base de la plateforme JEE, certaines d'entre eux seront utilisées pour le développement de notre application et nous avons parlé de l'architecture Client/serveur sa mise en œuvre et son fonctionnement

Chapitre II : présentation de l'entreprise, dans ce chapitre On a présenté une vue globale sur l'organisme d'accueil, puis on a exploré en détaille le domaine d'étude ainsi que le champ d'étude.

Chapitre III : analyse & conception, nous présentons la démarche adoptée pour la modélisation de notre application en utilisant un processus de développement du logiciel RUP ainsi que la conception générale et la conception détaillée du système.

Chapitre IV: Et pour finir nous enchaînons avec le chapitre de Réalisation & implémentation qui est consacré à la présentation de l'environnement matériel et logiciel utilisé pour la réalisation de notre application, en premier lieu. En second lieu, nous avons présenté les choix techniques adoptés ainsi que la solution proposée tout en s'aidant des interfaces graphiques de notre application.

Chapitre I : Présentation de la technologie Java EE

Introduction.

De nos jours, le génie logiciel nous offre des nouvelles technologies qui ont les potentialités de révolutionner le monde de conception des systèmes informatiques.

Un bon logiciel ne verra jamais le jour sans utiliser une bonne architecture logicielle qui respecte les critères de fiabilité, sécurité, portabilité...

Dans ce chapitre on va étudier une approche de l'architecture logicielle Java Entreprise Edition qui est recommandée pour la réalisation des applications d'entreprises.

I.1. Java EE (*Java Entreprise Edition*). [1]

Java EE (Java Entreprise Edition) est une spécification pour le langage de programmation java d'Oracle plus particulièrement destinée aux applications d'entreprise. Java EE est une plate-forme fortement orientée serveur pour le développement et l'exécution d'applications distribuées. Elle est composée de deux parties essentielles :

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

Dans ce but, toute implémentation de cette spécification contient un ensemble d'extensions au Framework Java standard (J2SE, Java 2 Standard Edition) afin de faciliter la création d'applications réparties.

I.2. Composants et architecture de JAVA EE. [2]

Le développement Java EE repose sur un découpage en couches ou tiers. Nous parlons alors d'application multi-tiers, trois grands tiers sons représenter :

- La couche présentation (tiers web).
- La couche métier (tiers métier ou tiers business).
- La couche stockage des informations (tiers entreprise information system).

CHAPITRE I : Présentation de la technologie JAVA EE

Les applications Java EE sont découpées en plusieurs composants réalisant des fonctionnalités spécifiques et installés sur une machine serveur ou sur plusieurs tiers distribués. Les composants Java EE sont des unités autonomes assemblées dans une application Java EE composée de classes et de fichiers et communiquent avec d'autres composants.

D'après le schéma a-après, la spécification Java EE définit les composants suivants :

- 1) Les composants clients ou tiers clients : sont des applications clientes (logiciel installé en local, navigateur web ou applets).
- 2) Les composants web ou tiers web : sont les technologies Servlets, Java Server Page (JSP) et Java Server Face (JSF).
- 3) Les composants métiers ou tiers métiers : sont des composants entreprise Java beans (EJB) qui représentent la logique métier. S'exécutent sur le serveur Java EE et dialoguent avec la couche de stockage EIS (entreprise information system).

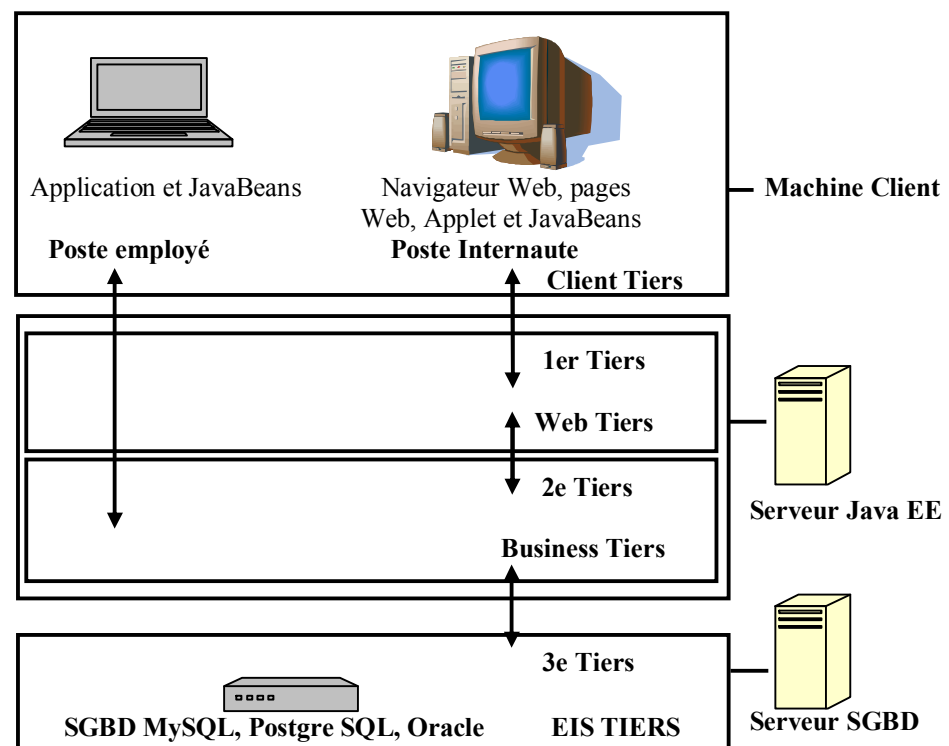


Figure I.1 : Architecture Java EE standard

CHAPITRE I : Présentation de la technologie JAVA EE

Une application Java EE multi-tiers correspond a un ensemble de couches souvent considérée comme étant une architecture trois tiers, celle-ci est distribuée de manière physique sur trois machines :

- La machine des clients.
- Le serveur d'application Java EE.
- Le serveur de base de données ou de stockage.

I.2.1. Les composants clients ou tiers client.

La plate-forme Java EE propose trois types de clients :

- Les clients web.
- Les applets.
- Les applications clients type Java SE.

I.2.1.1. Les clients Web.

Un client web consiste en des pages web de différents types (HTML, XHTML, XML, JavaScript ou autres) générées a partir de composant exécutés côté serveur dans un conteneur Web et capable de répondre aux requêtes HTTP en provenance de navigateur internet. Ces programmes coté serveur sont représentés en Java EE par des servlets, page JSP, et JSF

I.2.1.2. Les clients Applets.

Les applets sont des interfaces graphiques Java EE exécutés dans un navigateur web. Ces applications utilisent une interface graphique évoluée de type SWING et sont exécutées dans une machine virtuelle Java installée dans le navigateur. Cependant cette technique est plus contraignante à maintenir, requiert des accès et des droits pour la sécurité et nécessite un plug-in pour les navigateurs. Les clients Web sont donc préférables pour la création du tiers client.

I.2.1.3. Les application client.

Une application de type client est un logiciel riche, qui s'exécute sur la machine du client et fournit un ensemble de services aux utilisateurs par l'intermédiaire d'une interface graphique évoluée appelée Graphical User Interface.

Ces applications riches sont développées avec les composants SWING, en général exécutées avec Java web Start ou application client container ACC et déployées dans le projet Entreprise Archive (EAR).

I.2.2. Les composants Web (ou tiers Web).

Les composants Web Java EE sont des Servlets et/ou des pages Java Server Page(JSP) et/ou Java Server Face (JSF). Les Servlets sont des classes Java capable d'intercepter et de gérer les requêtes du Protocol http. Les pages JSP sont des documents textuels exécutés comme des Servlets apportant une solution simplifiée pour la programmation de page Web. La technologie Java Server Face (JSF) est construite a partir de Servlet et fournit un Framework de développement pour accélérer la création d'application Web.

I.2.3. Les composants métier ou tiers métier.

Les composants métier ou tiers métier représentent la couche business, avec les données du système, et sont de deux types :

- 1) Les entités beans (entity bean ou bean entity) peuvent être exécutées par un conteneur léger (pas forcément un serveur Java EE) et permettent de réaliser la persistance des données à partir des JavaBeans et de Java Persistence (JPA).
- 2) Les entreprise JavaBeans offrent des possibilités plus riches comme la gestion des transactions, les accès directs par des clients riches ou encore le gestion automatique des sessions utilisateur, mais sont exécuté sur un conteneur lourd, c'est-à-dire compatible Java EE.

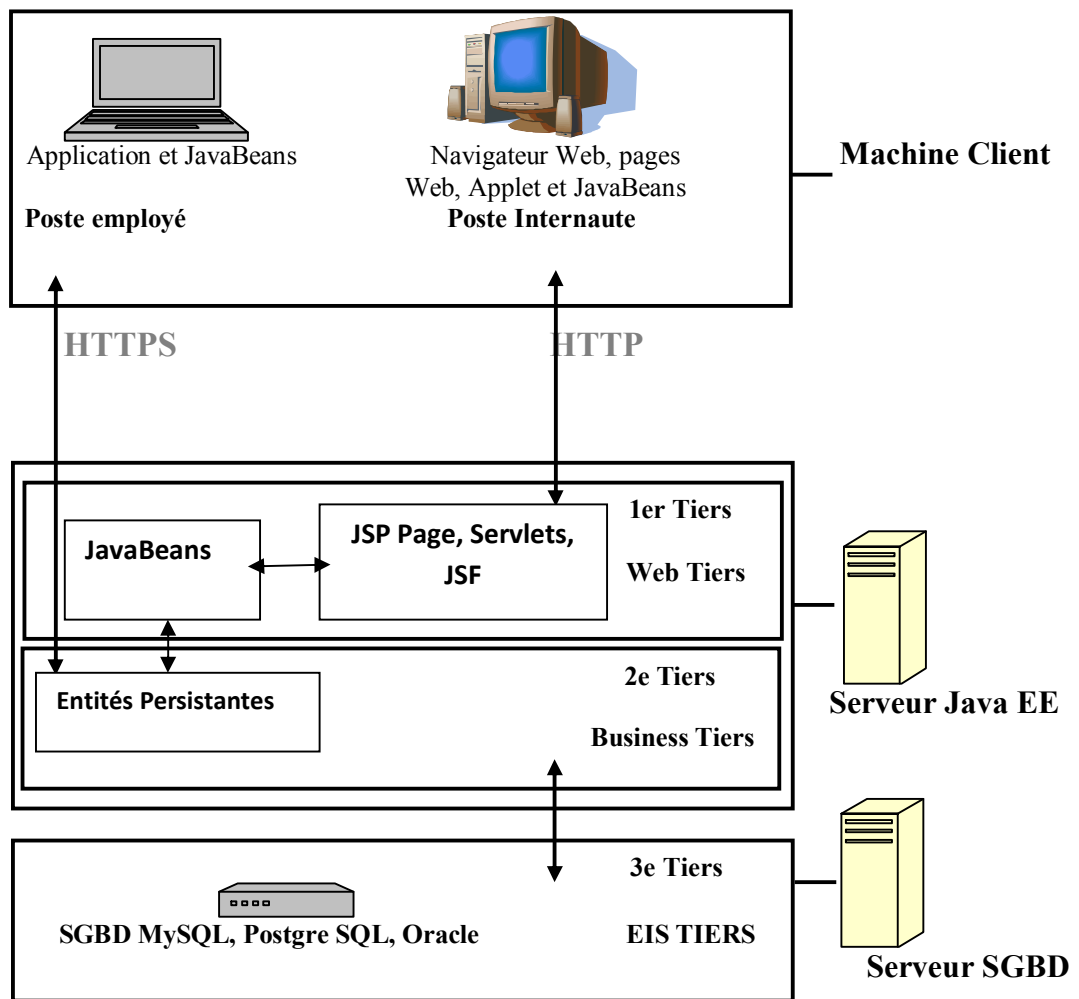


Figure I.2 : Communication Java EE entre les différents tiers.

La figure ci-dessus montre les différents types de clients utilisés dans une plate-forme Java EE. Les clients riches de type application communiquent directement avec le tiers Métier alors que les clients de type navigateur Internet communiquent à travers le tiers Web par l'intermédiaire de page exécutées sur le serveur et générées par des Servlets, JSP OU JSF.

I.3. Les conteneurs Java EE.

Les serveurs Java EE proposent plusieurs types de conteneurs (containers) pour chaque type de composant. Chaque conteneur a un rôle bien défini et offre un ensemble de services pour les développeurs :

- L'annuaire de nommage d'accès aux ressources : Java Naming and Directory Interface (JNDI) est une interface unifiée de gestion de nommage pour les services et l'accès à ceux-ci par des applications.
- L'injection dynamique de ressources.
- La gestion des accès aux bases de données.
- Le modèle de gestion de la sécurité.
- Le paramètre des transactions.

L'exécution et le développement sont donc directement liés au conteneur utilisé et une application Java EE côté tiers Web ne sera pas programmée comme une application côté tiers EJB par exemple. Les accès aux ressources nommés, les accès aux bases de données ou encore l'injection dynamique de ressources seront différents. Pour résumer, un conteneur permet de gérer le cycle de vie et la gestion des ressources.

Une application Java EE de type Web nécessite un conteneur Web pour son exécution alors qu'une application utilisant les EJB nécessite un conteneur EJB pour son exécution. Chaque conteneur propose un ensemble de services avec ses avantages et ses contraintes.

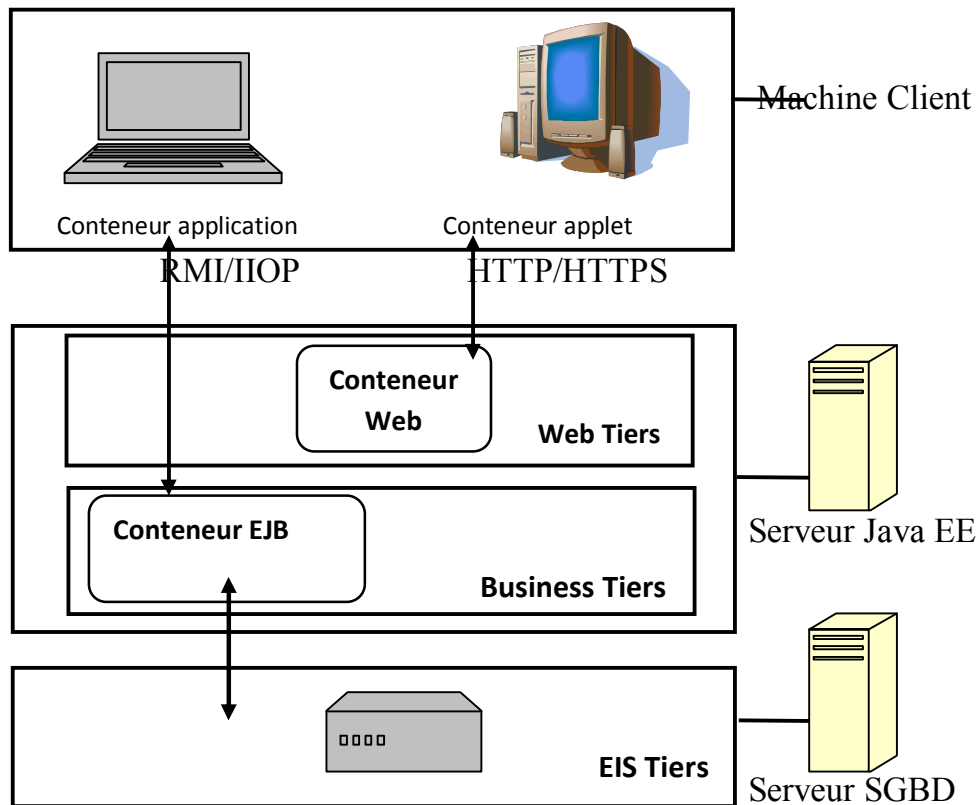


Figure I.3 : Serveur Java EE et Conteneurs.

L'architecture Java EE présentée dans la figure ci-dessus est découpée selon plusieurs domaines ou services, appelés des conteneurs ou containers. D'après les précédents schémas, il existe en Java EE cinq types de conteneurs :

- **Java EE serveur**, logiciel fournisseur du conteneur Web et EJB.
- **Web Container** ou **conteneur Web**, capable d'exécuter des pages JSP, Servlets et JSF (implémentation JSF nécessaire). Ces composants s'exécutent avec le conteneur serveur Java EE. Ce conteneur supporte les protocoles HTTP et HTTPS et dialogue avec différents types de clients (navigateur, application).
- **EJB Container** ou **conteneur EJB**, responsable de la couche métier et de l'exécution des *EJB*. Ce conteneur fournit le mécanisme de gestion des *EJB*, la gestion de leur cycle de vie, les transactions, le service de nommage *JNDI* et la possibilité des déclenchements asynchrones.

- **Applet Container** ou **Conteneur Applet**, responsable de l'environnement d'exécution des Applets Java. Ce type d'application consiste en un navigateur Web et le plug-in JAVA, lancés en même temps sur le poste client. Ces programmes sont dans un « bac à sable » aussi nommé « sandbox » afin de gérer la sécurité des accès sur le poste client.
- **Application Client Contrainer (ACC)** ou **conteneurs des applications clientes**, permet l'exécution d'applications JAVA SE. L'application de type client et son conteneur sont lancés sur le client en simultané. Le conteneur d'application client (ACC) propose une bibliothèque de classes, un système de gestion de sécurité et le service de nommage JNDI pour les applications JAVA SE. La communication directe entre le conteneur client et le conteneur EJB est réalisée avec les technologies RMI-IIOP, alors que la communication avec le conteneur Web est réalisée avec le protocole http pour les services Web, par exemple.

I.4. API et services Java EE. [2]

La plate-forme Java EE est la plus importante proposée par Oracle, regroupe les services de la plate-forme Java SE. Les services et API Java EE sont donc associés à une plate-forme et un ou plusieurs conteneurs. Les conteneurs étudiés précédemment proposent plusieurs services pour le développement d'application. En tant que développeur, nous pouvons utiliser la totalité des services ou bien au contraire sélectionner uniquement les techniques nécessaires à la réalisation du projet. Chaque service s'exécute dans un des cinq conteneurs étudiés précédemment et offre d'autres méthodes en fonction de ce conteneur.

I.4.1. Java Servlet (Technologie Java EE – Conteneur Web)

Ce service Java EE s'exécute dans le conteneur Web et propose un ensemble de classes de gestion des échanges http de type requête-réponse. Avec la nouvelle version de Java EE, la technologie Servlet 3.0 facilite la configuration et supporte les annotations ou encore les traitements asynchrones.

CHAPITRE I : Présentation de la technologie JAVA EE

I.4.2. Java Server Pages (JSP) (Technologie Java EE - Conteneur Web)

La technologie Java Server Pages (JSP) s'exécute dans le conteneur Web et utilise le mécanisme de génération de Servlets à partir de documents textuels. Les JSP contiennent des données statiques de type HTML, XHTML, JavaScript ou autres, et des balises ou tags JSP nommés JSTL correspondant à du code Java.

I.4.3. Java Server Pages Standard Tag Library (JSTL) (Technologie Java EE – Conteneur Web)

La technologie Java Server Pages Tag Library(JSTL) est encapsulée dans des pages JSP et propose une librairie de balises XHTML standardisées. Ces balises proposent de gérer les conditions et les itérations, l'internationalisation, les affichages HTML ou encore l'accès aux bases de données.

I.4.4. Java Server Faces (JSF) (Technologie Java EE - Conteneur Web)

La technologie Java Server Faces s'exécute dans le conteneur Web et correspond à la réponse du consortium Java EE en matière de framework de développement coté serveur.JSF est un Framework de développement de projet Web proposant la construction rapide de code serveur, la validation des entrées, la gestion des modèles objets, la configuration de la navigation, les conversions de données, l'utilisation d'annotations, les balises et technologies d'affichage de données pour la partie Vue, le support de la technologie Ajax en natif et bien d'autres fonctionnalités.JSF est orienté composant et repose sur un cycle de vie proche du développement d'applications Java SE (écouteurs, événements...)

I.4.5. Java DataBase Connectivity (JDBC) (Technologie Java SE - Conteneur Web, Conteneur EJB, Conteneur Application)

La technologie DataBase Connectivity (JDBC) permet de gérer les commandes SQL et les dialogues avec les différents SGBD relationnels. JDBC est utilisée lorsque nous devons accéder à une base de données dans une servlet, page JSF ou autres. L'API JDBC propose un ensemble de classes et interface pour se

CHAPITRE I : Présentation de la technologie JAVA EE

connecter aux différents SGBD du marché mais également les paquetages de manipulations des données. JDBC est une technologie sous-jacente à JPA.

I.4.6. Java Persistence API (JPA) (Technologie Java SE - Conteneur Web, Conteneur EJB et Conteneur Application)

Java Persistence API (JPA) est un standard Java utilisé pour la persistance des données. Ce mécanisme de persistance utilise le principe de mapping objet/relationnel et relationnel/objet afin de permettre de stocker les objets dans la base de données et inversement de pouvoir lire les données relationnelles et les transformer en objets :

- La gestion de la persistance.
- Un langage de requêtage évolué : Java Persistence Query Language (JPQL).
- Un mécanisme de mapping objet/relationnel ORM à partir de métadonnées (fichiers XML ou annotations).
- Une API à typage fort pour le requêtage.

I.4.7. Java Transaction API (JTA) (Technologie Java EE – Conteneur Web, Conteneur EJB)

Java Transaction API (JTA) est un API définissant des interfaces standard avec un gestionnaire de transactions.

I.4.8. Entreprise JavaBeans (EJB) (Technologie Java EE – Conteneur EJB)

Les composants Entreprise JavaBeans s'exécutent dans le conteneur dédié EJB et correspondent à la couche tiers Métier de l'application. Ces EJB sont des classes composées de champs et de méthodes d'accès. Java EE propose deux familles : les sessions beans, message-driven beans.

Les sessions beans représentent les conversations entre le client et le serveur, qu'elles soient avec ou sans état. Un message-Driven Bean (MDN) mélange une session bean et un écouteur de message permettant ainsi la réception asynchrone de messages. Ces services utilisent Java Message Service (JMS) pour le fonctionnement. La nouvelle version de Java EE offre la possibilité d'empaqueter

CHAPITRE I : Présentation de la technologie JAVA EE

les EJB directement dans des archives WAR (Web ARchive), l'utilisation du design Pattern Singleton dans les sessions beans et la possibilité d'utiliser des EJB légers avec un serveur Java ne proposant pas le conteneur EJB.

I.4.9. Java Message Service API (JMS) (Technologie Java EE – Conteneur Web, Conteneur EJB et Conteneur Application)

Java Message Service API (JMS) est le standar permettant aux applications Java EE de créer, d'envoyer, de recevoir et de traiter les messages asynchrones en mode point à point ou multi destinations.

I.4.10. JavaMail (Technologie Java SE – Conteneur Web, Conteneur EJB)

Les applications Java EE utilisent L'API JavaMail pour l'envoi des e-mails. L'API JavaMail fournit pour cela un ensemble de classes de gestion.

I.4.11. Java Autorization Service Provider Contract For containers (JACC) (Technologie Java EE – Conteneur Web, Conteneur EJB)

La sécurité des services est regroupée sous le terme Java Authentication and Autorisation Service (JAAS) et permet de gérer l'authentification et les accès clients. L'API Java Autorization Service Provider Contract For containers (JACC) définit un contrat entre une application Java EE et un fournisseur de services. La spécification Java ACC définit les classes de paquetage `Java.Security.Permission` pour la gestion d'Autorisation.

I.4.12. Java Authentification Service Provider Interface For containers (JASPIC) (Technologie Java EE – Conteneur Web, Conteneur EJB)

L'API Java Authentification Service Provider Interface For containers (JASPIC) propose interface d'authentification pour les dialogues entre le client et le serveur.

I.4.1.13. Java Authentification and Autorisation Service (JAAS) (Technologie Java EE – Conteneur Web, Conteneur EJB)

La technologie Java Authentification and Autorisation Service (JAAS) fournit les mécanismes d'authentification et d'autorisation pour un groupe ou utilisateur spécifique lançant l'application. JAAS est une version de standard Pluggable Authentication Module (PAM).

I.4.14. Java Naming and Directory Interface (JNDI) (Technologie Java SE–Conteneur Web, Conteneur EJB, Conteneur Application)

L'API Java Naming and Directory Interface (JNDI) fournit un mécanisme de nommage de type annuaire pour l'accès aux ressources. Ces ressources peuvent être de différents types mais le but est d'associer les objets à un nom (bind) et de retrouver ces objets dans un annuaire de nommage semblable à LDAP, DNS, ou NIS. Nous pouvons ainsi nommer une source de données de type base de données, une Entreprise JavaBean, une session JavaMail ou autre. JNDI permet une utilisation et un accès depuis une application de type client, un EJB ou un composant Web.

I.5. Les avantages d'utiliser JAVA EE :

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

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

I.6. Architecture client/serveur.

I.6.1. Définition de l'architecture client/serveur. [3]

- **Client :** Un client est un logiciel qui utilise un service offert par un serveur. Le client envoie une requête et reçoit la réponse.
- **Serveur :** Un serveur est un logiciel 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, une machine généralement très puissante, sur laquelle s'exécute le logiciel serveur qui fonctionne en permanence.

- **L'architecture client/serveur** : Est un fonctionnement coopératif entre le client et le serveur qui s'exécutant sur deux machines différentes. Les services internet sont conçus selon cette architecture.

I.6.2. Principe de fonctionnement. [4]

Le client émet des requêtes vers le serveur. Le Serveur attend en permanence les requêtes, les traite et envoie le résultat au client adéquat. Selon le schéma suivant :

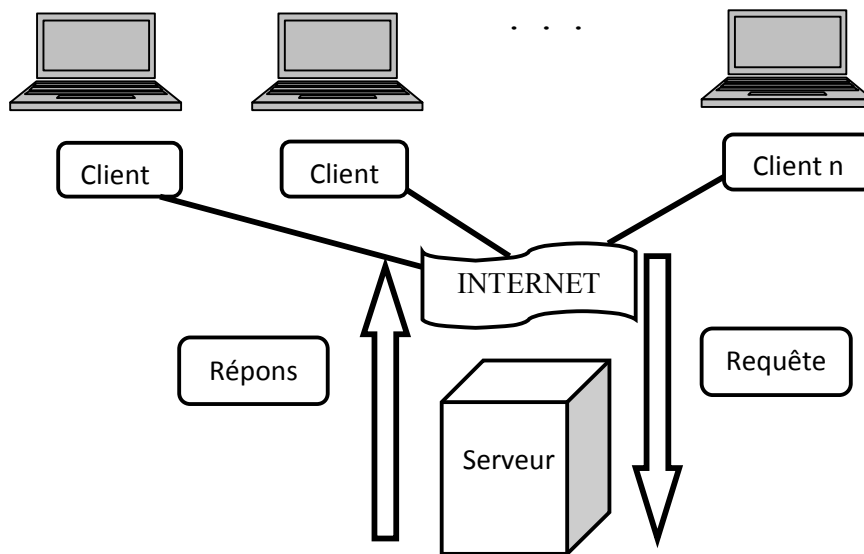


Figure I.4: Architecture client/serveur.

I.6.3. Les concepts client/serveur :

I.6.3.1. Le concept client :

Le composant client, est constituée d'un micro-ordinateur indépendant, fournit à l'utilisateur toute la gamme de ses services pour exécuter des applications.

a) Type de clients : [3]

- **Client lourd** : Le terme « client lourd » désigne une application cliente graphique exécutée sur le système d'exploitation de l'utilisateur. Il possède des capacités de traitement évoluées et une interface graphique sophistiquée.
- **Client léger** : Le terme « client léger » ; désigne une application accessible via une interface web (en HTML) consultable à l'aide d'un navigateur web,

où la totalité du logique métier est traitée du côté du serveur. Pour ces raisons, le navigateur est parfois appelé client universel.

- **Client riche :** Un « client riche » est un compromis entre le client léger et le client lourd. Son objectif est de proposer une interface graphique, permettant d'obtenir des fonctionnalités similaires à celles d'un client lourd (glisser déposer, onglets, multi fenêtrage, menus déroulants).

I.6.3.2. Le concept serveur :

Le composant **serveur**, est constitué soit d'un autre micro-ordinateur, d'un mini-ordinateur ou d'un grand centre informatique, soutient le client en lui offrant, dans un environnement de temps partagé, des services de gestion de données, de partage d'information, d'administration du réseau et de sécurité.

b) Types de serveur :

- **Serveur d'application web :** L'Internet est la plus grande application client/serveur. Ce modèle consiste en des clients légers et portables qui communiquent via le protocole http (HyperText Transmission Protocol) avec des très grands serveurs.
- **Serveur de fichier :** Le serveur s'occupe de la gestion des fichiers. Le client demande l'accès (écriture ou lecture) à des fichiers en émettant des requêtes sur un réseau en direction du serveur.
- **Serveur de bases de données :** Dans ce type de serveur, le client émet des requêtes SQL sous forme de messages en direction du serveur. Les données ainsi que le code qui traite les requêtes résident sur la même machine serveur.

Voici une figure qui résume les différents serveurs :

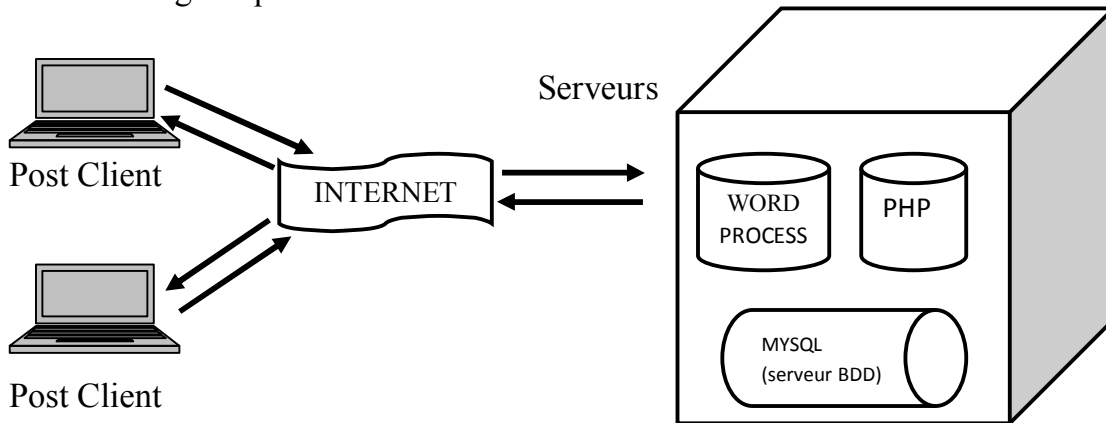


Figure I.5: Les types de serveur.

I.6.4. Classification de l'architecture client/serveur :

a) Architecture à deux niveaux (2-tiers):

Elle se compose d'une interface graphique qui se situe sur le poste client et la base de données est localisée sur le serveur. Pratiquement toutes les charges applicatives s'exécutent sur le client qui devient de ce fait un *client lourd*.

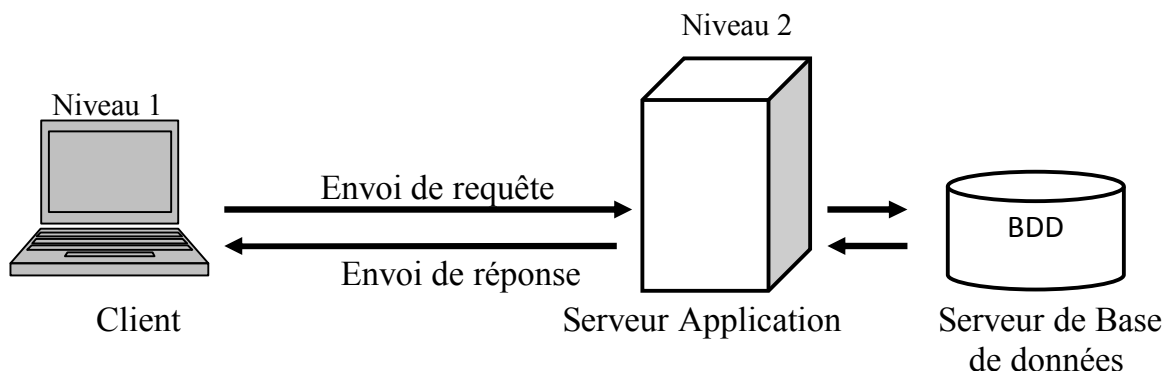


Figure I.6: Architecture client/serveur à 2-tiers.

➤ Avantages du modèle 2-tiers :

- La technique simple du C/S 2-tiers, est excellente pour créer, rapidement des applications à petite échelle.
- Bonne solution lorsque le nombre de clients ne dépasse pas la centaine.

➤ Inconvénients du modèle 2-tiers :

- Manque de modularité: difficulté de réutiliser le code ou de le maintenir.
- Modification de l'application ou de la structure de la base de données nécessite forcément un redéploiement sur les postes clients.
- La charge, difficile à réduire, du poste client qui supporte la grande majorité des traitements applicatifs.

CHAPITRE I : Présentation de la technologie JAVA EE

- Le poste client est forcément sollicité, il devient de plus en plus complexe et doit être mis à jour régulièrement pour répondre aux besoins des utilisateurs.
- La conversation entre le client et le serveur est assez bruyante (en raison de la quantité des données échangées).

b) Architecture à trois niveaux (3-tiers):

L'architecture est partagée entre le client (demandeur de la ressource), le serveur d'application (chargé de fournir la ressource mais faisant appel à un autre serveur) et le serveur secondaire (généralement un serveur de base de données fournissant le service au premier serveur).

Le client devient ainsi léger. Ce modèle répartit donc la charge applicative en trois:

- Le client fournit l'interface graphique et interagit avec le serveur par des appels de services distants.
- le serveur d'application qui exécute la logique applicative qui se trouve dans le niveau médian ; ces traitements deviennent des entités de première importance générées indépendamment de l'interface utilisateur et de la base de données.
- Serveurs de base de données.

Voici une figure qui représente l'architecture client/serveur à 3 tiers :

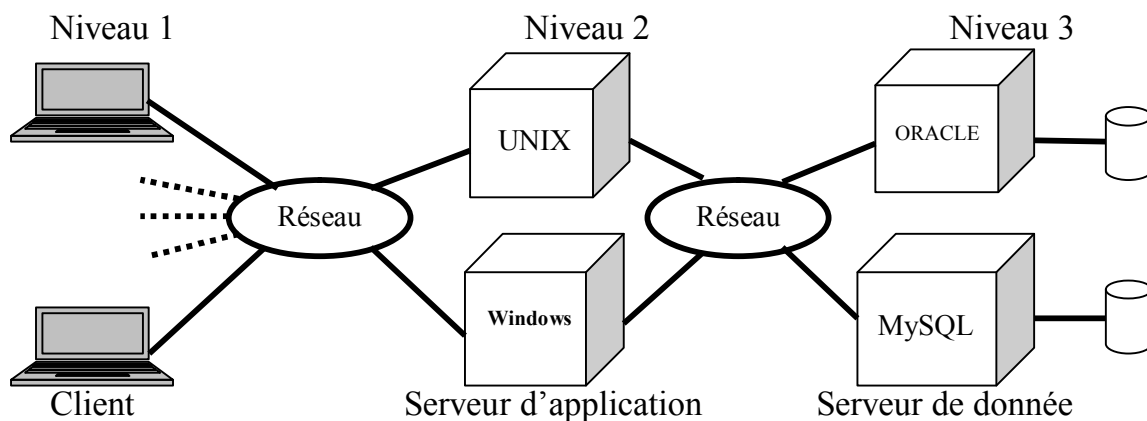


Figure I.7: Architecture client/serveur à 3 tiers

➤ Avantages du modèle 3-tiers :

- Possibilité d'étendre et de développer ce modèle.
- Cette architecture supporte des milliers d'utilisateurs voir des millions.

CHAPITRE I : Présentation de la technologie JAVA EE

- Répartitions géographique des serveurs.
- Facile à gérer et à déployer des solutions. (code centré dans le serveur)
- Plus de sécurité: Le serveur qui accède à la BDD et non pas le client
- Optimisation du nombre de requêtes envoyées par l'utilisateur puisque c'est le serveur qui s'en charge d'appeler le SGBD.

➤ Comparaison de deux modèles (2 tiers et 3 tiers) :

Critères	2-tiers	3-tiers
Administration du système.	Complexe	Moins complexe (Code centralisé sur le serveur)
Sécurité.	Faible (surtout au niveau des données)	Elevée
Encapsulation des données.	Faible (Accède aux données)	Elevée (Invoque des services)
Extensibilité.	Faible	Elevée (Redimensionnement vertical)
Performances	Mauvaise (Envoie de donnée SQL)	Bonne (Invocation de services)
Richesse du choix de communication.	Aucune (Appel de type RPC, synchrone orienté connexion)	Bonne (RPC, délivrance de message, files d'attente, ect.)
Souplesse de l'architecture du matériel.	Limitée (Il ya un client et un serveur)	Elevée (Plusieurs serveurs)
Disponibilité.	Faible (En cas de panne pas de passage à un serveur de secours)	Elevée (On peut redémarrer des services du niveau médian sur d'autres serveurs)

Tableau I.1 : Comparaison de deux modèles (2 tiers et 3tiers)

I.6.5. Les avantages de l'architecture client/serveur :

Le modèle Client/serveur est particulièrement recommandé pour des réseaux nécessitant un grand niveau de fiabilité, ses principaux avantages 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 centralisées 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 ont peu d'importance dans ce modèle, ils ont moins besoin d'être administrés.
- **Un réseau évolutif :** grâce à cette architecture, on peut supprimer ou rajouter des clients sans perturber le fonctionnement du réseau et sans modifications majeures.

I.6.6. Les inconvénients de l'architecture client/serveur :

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

- **Un coût élevé** : Dû à la technicité du serveur.
- **Accessibilité des pages** : les pages web sont accessibles par tous les utilisateurs ; en plus s'il y a trop de clients en même temps, le serveur risque de ne pas supporter la charge.
- Si le serveur devient non disponible, les clients ne pourront plus fonctionner.

Conclusion :

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

Chapitre II :
Présentation de
l'organisme
d'accueil

CHAPITRE II : Présentation de l'organisme d'accueil

Introduction :

La présentation de l'organisme d'accueil est une étape très importante qui nous permet de prendre connaissance de cet organisme qui est la GCB (Société Nationale de Génie Civil et Bâtiment), ainsi que notre champ d'étude qui est le service des ressources humaines.

II.1. Présentation générale de l'entreprise :

GCB (« Société Nationale de Génie Civil et Bâtiment ») est une entreprise nationale algérienne et un acteur majeur dans les projets de Génie civil et du Bâtiment.

L'entreprise est organisée en structures régionales polyvalentes dont chacune dispose des capacités matérielles et managériales pour répondre aux besoins des clients localement en Algérie et pays voisins.

Elle dispose d'un parc important de plus de 2 500 engins lourds et matériels stratégiques en constante modernisation qui lui donne une grande autonomie d'action. L'entreprise GCB emploie environ 6 000 salariés.

Le logo de l'entreprise GCB est comme suit :



Figure II.1 : Le logo de l'entreprise GCB.

CHAPITRE II : Présentation de l'organisme d'accueil

II.2. Historique :

La Société Nationale de Génie Civil et Bâtiment par abréviation GCB est une société issue de la restructuration de Sonatrach, créée le 1^{er} août 1981 par décret N° 81-173 et érigée en société par actions le 12 juillet 1998 avec un capital social actuel de 7.630.000.000 de Dinars Algériens, actions entièrement détenues par Sonatrach.

Certifiée ISO 9001 V 2008, GCB capitalise une expérience de plus de 30 années dans les différents domaines de la construction, une existence couronnée de succès qui lui a valu une maîtrise sereine de ses activités.

II.3. Implantation de GCB :

GCB est organisée en structures régionales polyvalentes dont chacune dispose des capacités matérielles et managériales pour répondre aux besoins des clients localement.

GCB dispose d'infrastructures fixes, à vocation industrielles, administrative et socioprofessionnelles et sont notamment implantés :

- Alger (Zones Industrielles d'El-Harrach et Oued-Smar).
- Boumerdes (Boumerdes et Boudouaou).
- Arzew (Zone Industrielle d'Arzew).
- Hassi-Messaoud, Rhourd Nouss.
- Hassi-r'mel, Ain Salah, Adrar.
- Ain Amenas, Illizi, TFT, Alrar.

GCB dispose d'infrastructures nécessaires à l'installation de bases de vie et de bases industrielles sur les chantiers avec une capacité d'accueil pour la prise en charge de plus de 5000 agents sur les différents sites de projets (Studios, Cabines, d'hébergements, Chalets, Cantines, etc....)

Société (**GCB**) a développé et doté un parc de moyens de transports importants qui lui permettent d'ouvrir des chantiers et de s'installer dans les meilleurs temps et en n'importe quel lieu en Algérie.



Figure II.2 : Implantation de GCB en Algérie.

II.4. Les activités de GCB :

L'activité de **GCB** s'articule autour de neuf branches principales :

- **Activité Terrassement :** Avec ses moyens de production conséquents, avec ses hommes expérimentés et rodés à l'art de ce métier, **GCB** intervient sur des projets dans les contrées les plus reculées l'Algérie, dans les conditions climatiques et reliefs les plus rudes, et y réalise des mouvements de terre colossaux.

GCB intervient sur :

- ✓ Les terrassements en grande masse.
- ✓ Les travaux de rechargement en matériaux choisis de grandes surfaces.
- ✓ L'ouverture d'emprises.
- ✓ La construction de digues de rétention en matériaux choisis

CHAPITRE II : Présentation de l'organisme d'accueil

- **Activité Génie Civil industriel : GCB** intervient sur travaux de génie civil les plus divers et les plus. Ses importantes capacités de production de béton (09 centrales à biton et 10 stations de concassage) la rendent apte à répondre à tout projet qui requiert plus de 100000 m³ de béton.

GCB réalise des complexes industriel de grandes envergure tel que :

- ✓ Les unités de récupération des GPL.
 - ✓ Les centres de traitement des huiles.
 - ✓ Les centres de stockage des hydrocarbures.
 - ✓ Les stations de compression et de pompage.
-
- **Activité Génie Civil Pipe-line** : Il englobe les phases suivantes :
 - ✓ Ouverture de l'emprise.
 - ✓ Réalisation de la piste de service.
 - ✓ Réalisation des tranchées
 - ✓ Mise en œuvre des remblais.
 - ✓ Réalisation des traversées de routes, de pistes, d'oueds et de pipes.
 - ✓ Réalisation du génie-civil des ouvrages concentrés.
-
- **Activité Routes : GCB** a réalisé des centaines de kilomètres de routes à travers les différentes régions d'Algérie ce qui lui a permis la maîtrise de cette activité et lui a ouvert la porte à de nouveaux horizons en la matière en participant dans le cadre d'un groupement à l'édification du grand national de l'automobile Est-Ouest.
-
- **Activité Génie civil hydraulique : GCB** a investi le créneau de l'hydraulique, elle a réalisé des projets de transferts d'eau à partir de barrages, qui comprennent les travaux de terrassements généraux, d'excavation et de remblais de tranchée sur la ligne ainsi que le génie civil des infrastructures d'accompagnement telles que les stations d'épuration, les stations de pompes, les stations électriques, et les réservoirs de stockage d'eau.

CHAPITRE II : Présentation de l'organisme d'accueil

- **Activité Génie civil ferroviaire : GCB** a diversifié ses activités et s'engage dans la construction d'ouvrages de génie civil ferroviaires. Les travaux comprennent l'ouverture de l'emprise, la construction du corps d'assise de la voie, et les ouvrages en béton armé des dalots. Un challenge de plus pour une entreprise tournée vers l'avenir.
- **Activité Génie civil aéroportuaire : GCB** a diversifié ses activités et s'engage dans la construction d'ouvrages de génie civil aéroportuaire. Les travaux comprennent la réalisation et le renforcement des pistes d'atterrissage et des ouvrages annexes.
- **Activité Plate-forme de forage : GCB** réalise des plate-forme de forage dans les endroits les plus reculés d'Algérie, dans les conditions climatiques et reliefs les plus rudes.
- **Activité Bâtiment : GCB** réalise des bâtiments à l'usage d'habitation ou socioprofessionnel en tout corps d'état.
- **Activité Engineering :** Le bureau d'études technique de GCB possède une longue expérience et un savoir-faire dans le domaine d'engineering, allant de l'étude de faisabilité jusqu'aux études de conception générale et de détail, ainsi qu'au suivi et contrôle des travaux de réalisation.
Le bureau d'études technique de GCB offre les services d'une équipe pluridisciplinaire pour la réalisation d'études d'engineering qui englobent diverses activités de constructions tels que :
 - ✓ Le génie civil.
 - ✓ L'architecture.
 - ✓ Les corps d'états secondaires.
 - ✓ La charpente métallique.
 - ✓ Les études topographiques, ainsi que les prestations de maîtrise d'œuvre.

CHAPITRE II : Présentation de l'organisme d'accueil

- **Activité Construction en charpente métallique légère** : GCB à travers son unité de charpente métallique intervient dans :
 - ✓ La fabrication et montage d'infrastructures diverses en charpente métallique tels que les bâtiments, les hangars, les ateliers et les cabines sahariennes.
 - ✓ La fabrication chaudronnerie.
 - ✓ Le montage des toitures et bardage en panneaux sandwich, tôles nervurée ou translucide y compris les accessoires.
 - ✓ Réalisation de réseaux anti-incendie et de tuyauterie.

II.5. Organisation de GCB :

La Société Nationale de Génie Civil et Bâtiment est organisée comme suit :

- ✓ Niveau1 : La direction générale(DG).
- ✓ Niveau2 : les directions centrales de soutien.
- ✓ Niveau3 : Les directions Techniques.
- ✓ Niveau4 : Les directions Régionales.

II.5.1. La direction générale (DG) :

La direction générale est l'unique entité qui est responsable de stratégie et de développement de l'entreprise.

La direction générale de **GCB** est composée comme suit

- Président directeur général PDG :
- Assistants auprès du PDG :
- Direction Audit (AUD).
- Direction Qualité, Santé, Sécurité et Environnement (DMQ).
- Direction Juridique et Passation des Marchés.
- Direction Général Adjoint Technique.

CHAPITRE II : Présentation de l'organisme d'accueil

II.5.2. Directions centrales de soutien.

- Direction Etude et Développement (EDS).
- Direction Finance (FIN).
- Direction Commerciale (COM).
- Direction Achats.
- Direction Ressources Humaines (DRH).

II.5.3. Directions Techniques.

- Direction Affaires Sociales (AFS).
- Direction Moyen Généraux (MGX).
- Direction Matériels.
- Direction Engineering
- Direction Pipeline et Charpente Métallique (UCM).

II.5.4. Directions Régionaux.

- **Trois directions Régionales Nord :**
 - Direction Région Centre (DRC).
 - Direction Région Est (DRE).
 - Direction Région Ouest (DRO).

- **Trois directions Régionales Sud**
 - Direction Sud-ouest (DSO).
 - Direction Sud Est (DSE).
 - Direction Sud In Amenas (DSI).

CHAPITRE II : Présentation de l'organisme d'accueil

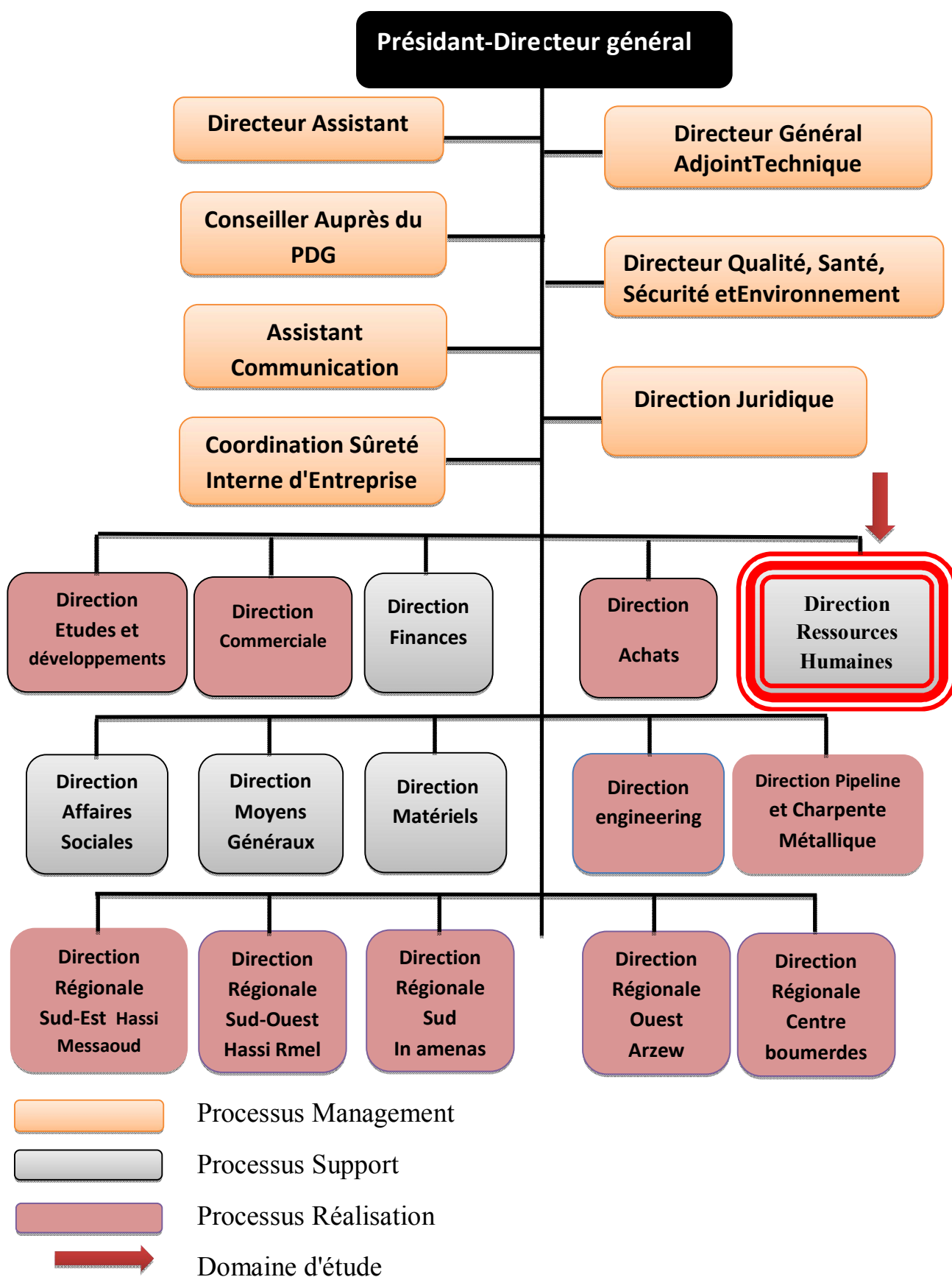


Figure II.3 : Organigramme générale de GCB.

CHAPITRE II : Présentation de l'organisme d'accueil

II.6. Direction des ressources humaines. (Domaine d'étude)

Le Directeur Ressources Humaines a pour objet de :

- ✓ Elaborer et proposer une stratégie de gestion des ressources humaines au service des objectifs globaux de la Société.
- ✓ Mettre en place des dispositifs, systèmes, modes, méthodes et procédures en matière de gestion des ressources humaines :
 - Organisation.
 - Sélection / recrutement.
 - Rémunération.
 - Evaluation des performances des salariés et systèmes de stimulation.
 - Gestion prévisionnelle et gestion des carrières.
 - Gestion des compétences.
 - Formation continue et recyclage.
 - Relations de travail.
- ✓ Définir les modalités d'application, à l'échelle de la Société, des dispositifs, systèmes et procédures de gestion des ressources humaines.
- ✓ Veiller à l'élaboration de plans de formation et d'apprentissage et suivre leur exécution.
- ✓ Veiller à l'élaboration de bilans analytiques de la gestion des ressources humaines à tous les niveaux pour alimenter et préparer les mesures correctives.
- ✓ Mettre en place un dispositif d'application et de contrôle des systèmes, méthodes, procédures et règles de toutes natures édictées en matière de GRH (missions de contrôle programmées et /ou ponctuelles sur tous les sites).
- ✓ Alimenter et entretenir le dialogue social au sein de la Société.
- ✓ Veiller à l'application des textes législatifs et réglementaires.
- ✓ Assister les entités en matière de gestion des ressources humaines, de réglementation et de contentieux.

CHAPITRE II : Présentation de l'organisme d'accueil

- ✓ Veiller au bon fonctionnement du Système Management Qualité à travers, notamment, le pilotage du processus gestion des ressources humaines.

II.6.1. Organigramme de direction des ressources humaines.

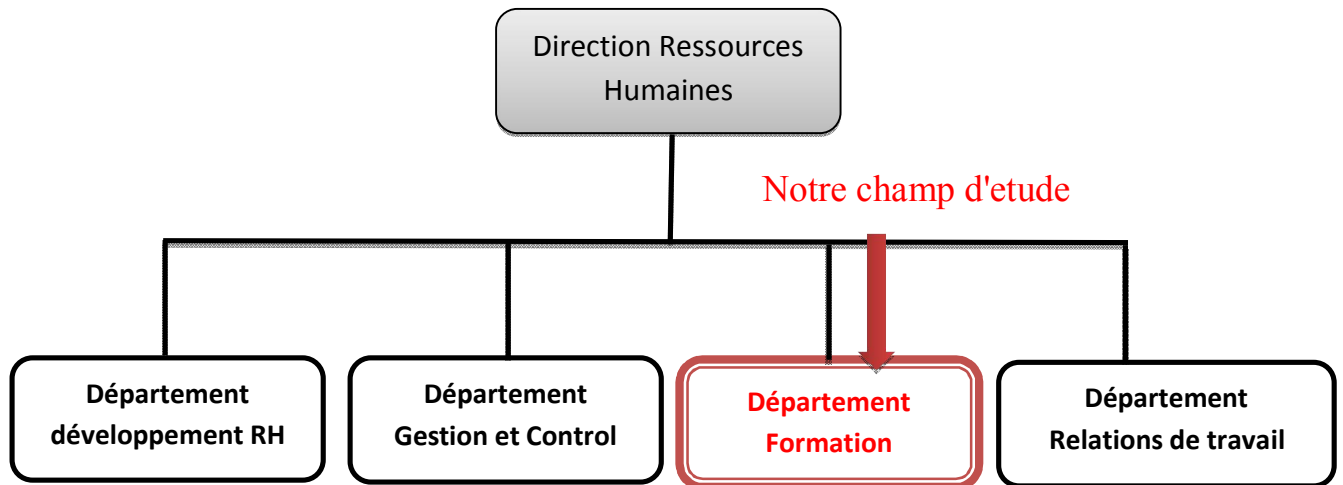


Figure II.4 : Organigramme de direction des ressources humaines.

II.6.2. Missions de chaque département.

II.6.2.1. Département développement RH.

- ✓ Prendre en charge l'acquisition des ressources humaines pour la société.
- ✓ assurer leur formation et la gestion des compétences et des carrières.
- ✓ Prendre en charge la gestion des cadres supérieurs et le suivi de la banque de données des cadres à haut potentiel.
- ✓ Mettre en œuvre des programmes de développement liés au processus de la relève au sein de la Société.
- ✓ Centraliser les plans et programmes de développement des ressources humaines de toutes les entités à l'échelle de la Société, notamment en matière de recrutement des cadres.
- ✓ Développer et implémenter notamment les instruments de :
 - Gestion prévisionnelle en ressources humaines.
 - Gestion des compétences.
 - Gestion des carrières.

CHAPITRE II : Présentation de l'organisme d'accueil

- ✓ Mettre en place et entretenir une banque de données exhaustives et actualisées sur :
 - Les qualifications et les compétences disponibles au sein de la Société.
 - performances collectives et individuelles.
- ✓ Animer et coordonner, à l'échelle de la Société, la mise en œuvre des systèmes d'évaluation et de rétributions professionnelles.
- ✓ Analyser les besoins en recrutement de cadres et développer des plans et programmes annuels de recrutement.
- ✓ Piloter les actions d'études de postes de travail au sein de la Société.
- ✓ Gérer le manuel général d'organisation de la Société.
- ✓ Assister et porter conseil en matière d'organisation et de ressources humaines.
- ✓ Initier toutes mesures susceptibles d'améliorer la qualité des prestations et veiller au respect strict des normes et procédures en vigueur.
- ✓ Assurer la mise en place de la structure d'organisation de son département conformément à l'organigramme approuvé de la direction et recommander, le cas échéant, au Directeur Ressources Humaines toute modification interne susceptible d'améliorer les performances de son département.
- ✓ Mettre en œuvre les moyens humains et matériels nécessaires au bon fonctionnement de son département, préparer les budgets correspondants et diriger les activités de son département dans les limites des budgets approuvés et des pouvoirs qui lui sont délégués.

II.6.2.2. Département Gestion et Control.

- ✓ Prendre en charge la gestion et le contrôle des ressources humaines aux fins de permettre à l'Entreprise de maîtriser ses coûts en matière de frais du personnel.
- ✓ Assurer la planification en matière de ressources humaines.
- ✓ Animer l'élaboration du budget des effectifs.
- ✓ Elaborer le tableau de bord ressources humains ainsi que divers rapports périodiques.

CHAPITRE II : Présentation de l'organisme d'accueil

- ✓ Assurer la mise en place et le suivi d'une banque de données statistiques sur les ressources humaines.
- ✓ Assurer le traitement informatique de la paie par le Service Rémunération et Contrôle de la paie, ainsi que le centre de calcul et de tirage de la paie.
- ✓ Superviser les modalités de rémunération et de l'exécution de la paie.
- ✓ Animer et assurer les opérations de gestion courante des ressources humaines, notamment en matière de : paie, administration, gestion sociale, etc.
- ✓ Assister et porter conseil en matière de gestion des ressources humaines.
- ✓ Initier toutes mesures susceptibles d'améliorer la qualité des prestations et veiller au respect strict des normes et procédures en vigueur.
- ✓ Mettre en œuvre les moyens humains et matériels nécessaires au bon fonctionnement de son département, préparer les budgets correspondants et diriger les activités de son département dans les limites des budgets approuvés et des pouvoirs qui lui sont délégués.
- ✓ Assurer la mise en place de la structure d'organisation de son département conformément à l'organigramme approuvé de la direction et recommander, le cas échéant, au Directeur Ressources Humaines toute modification interne susceptible d'améliorer les performances de son Département.

II.6.2.3. Département Relations de Travail.

- ✓ Mettre en œuvre les textes législatifs et réglementaires régissant les relations de travail afin de préserver les intérêts de la Société et des travailleurs et de contribuer à la sérénité du climat social.
- ✓ Développer la vulgarisation des textes législatifs et réglementaires en matière de relations de travail.
- ✓ Veiller à l'application rigoureuse de la législation et de la réglementation régissant les relations de travail.
- ✓ Veiller à l'actualisation de la réglementation interne à la Société.
- ✓ Conduire des études et des sondages sur le climat social et recommander des actions d'amélioration.
- ✓ Assister les entités de la société en matière de relations de travail.

CHAPITRE II : Présentation de l'organisme d'accueil

- ✓ Entretien des échanges réguliers avec l'environnement professionnel : Inspection du Travail territorialement compétente / Instances syndicales supérieures.
- ✓ Favoriser l'écoute des salariés et le dialogue social.
- ✓ Prendre en charge les doléances et les requêtes des travailleurs et préconiser des solutions appropriées.
- ✓ Initier toutes mesures susceptibles d'améliorer la qualité des prestations et veiller au respect strict des normes et procédures en vigueur.
- ✓ Assurer la mise en place de la structure d'organisation de son département conformément à l'organigramme approuvé de la direction et recommander, le cas échéant, au Directeur Ressources Humaines toute modification interne susceptible d'améliorer les performances de son département.
- ✓ Mettre en œuvre les moyens humains et matériels nécessaires au bon fonctionnement de son département, préparer les budgets correspondants et diriger les activités de son département dans les limites des budgets approuvés et des pouvoirs qui lui sont délégués.

II.7. Présentation de Département Formation (notre champ d'étude).

La présente procédure a pour objet de définir les modalités de gestion de la formation continue au sein de la Société **GCB**. Elle en définit à ce titre, les étapes d'identification et de collecte.

Cette procédure s'applique à l'ensemble des actions de formation réalisées au sein de la Société.

CHAPITRE II : Présentation de l'organisme d'accueil

II.7.1. Définitions et Terminologie :

- **Besoin de formation :**

C'est la nécessité d'adaptation ou d'acquisition de nouvelles connaissances et /ou qualifications découlant de la stratégie de développement et des objectifs définis par la Direction Générale, ou résultant de contraintes inhérentes à l'environnement externe de la Société.

- **Plan de formation :**

C'est un document qui permet de recenser, de consolider et de planifier l'ensemble des besoins de formation approuvés au niveau de la Société, ainsi que les moyens dégagés pour leur satisfaction (budget, effectif).

- **Filière de formation :**

Il s'agit du regroupement des différentes spécialités par processus.

- **Nature de la formation :**

La formation peut être dispensée selon trois (03) types.

- a) La formation de courte durée : Par formation de courte durée, est entendue toute action de formation, qui permet de répondre à une tâche spécifique, dont la durée effective n'excède pas dix (10) jours.
- b) La formation de moyenne durée : Par formation de moyenne durée est entendue toute action de formation dont la durée effective, continue ou alternée, est supérieure à dix (10) jours et inférieure à trente (30) jours.
Les actions de formation de moyenne durée sont généralement des stages de perfectionnement.
- c) La formation de longue durée : Par formation de longue durée est entendue toute action de formation dont la durée effective est supérieure à trente (30) jours.

CHAPITRE II : Présentation de l'organisme d'accueil

II.7.2. Missions de département formation :

- ✓ Mettre en œuvre la politique de l'Entreprise en matière de formation à l'effet de promouvoir le personnel.
- ✓ Développer des actions de formation continue.
- ✓ Elaborer les plans de formation sur la base des besoins exprimés par les différentes structures conformément à la politique de la Direction Générale en la matière.
- ✓ Elaborer les plans d'apprentissage conformément à la réglementation en vigueur.
- ✓ Veiller à la réalisation des plans de formation et d'apprentissage suivant le planning de réalisation dûment arrêté.
- ✓ Coordonner et consolider les programmes de formation, de perfectionnement et d'apprentissage.
- ✓ Veiller à la mise en place d'une banque de données actualisée sur les organismes formateurs.
- ✓ Organiser des réunions avec les organismes formateurs et négocier les contrats de formation.
- ✓ Suivre la programmation et la concrétisation des actions de formation.
- ✓ Contrôler l'analyse des évaluations des formations dispensées et des organismes formateurs à l'issue de chaque action de formation.
- ✓ Initier toutes mesures susceptibles d'améliorer la qualité des prestations et veiller au respect strict des normes et procédures en vigueur.
- ✓ Assurer la mise en place de la structure d'organisation de son département conformément à l'organigramme approuvé de la direction et recommander, le cas échéant, au Directeur Ressources Humaines toute modification interne susceptible d'améliorer les performances de son département.
- ✓ Mettre en œuvre les moyens humains et matériels nécessaires au bon fonctionnement de son département, préparer les budgets correspondants et diriger les activités de son département dans les limites des budgets approuvés et des pouvoirs qui lui sont délégués.

CHAPITRE II : Présentation de l'organisme d'accueil

II.7.3. Organigramme de département Formation.

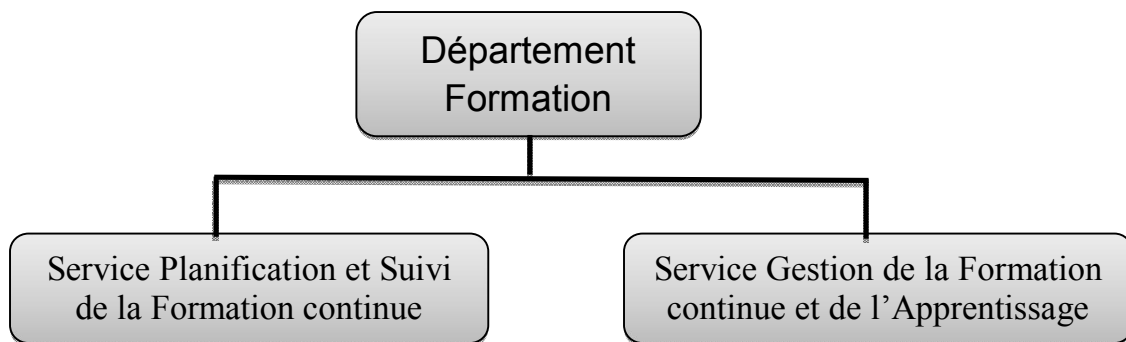


Figure II.5 : Organigramme de département Formation.

II.7.4. Service Planification et Suivi de la Formation continue.

Le service Planification et Suivi de la Formation continue chargé de :

- ✓ Participer à l'élaboration du plan de formation annuel sur la base des besoins exprimés par les différentes structures de l'Entreprise.
- ✓ Diffuser les plans de formation approuvés par le Président -Directeur Général.
- ✓ Coordonner et consolider les programmes de formation et de perfectionnement arrêtés au niveau de l'Entreprise.
- ✓ Tenir à jour une banque de données sur les organismes formateurs.
- ✓ Tenir à jour les dossiers de formation (ingénierie de la formation et post formation).
- ✓ Elaborer les états de suivi de la formation (situations périodiques, tableau de bord SH, reporting , conjoncture trimestrielle, etc...).
- ✓ Elaborer et suivre les indicateurs qualité se rapportant à ses activités.
- ✓ Etablir le rapport d'activités mensuel.
- ✓ Participer à l'élaboration des procédures de travail et veiller à leur application.
- ✓ Proposer et mener toute action tendant à apporter une amélioration en matière de management de la qualité et ce, dans le cadre des normes et procédures y afférentes.

CHAPITRE II : Présentation de l'organisme d'accueil

II.7.5. Service Gestion de la Formation continue et de l'Apprentissage.

Le service Gestion de la Formation continue et de l'Apprentissage chargé de :

- ✓ Réaliser les plans de formation suivant le planning de réalisation dûment arrêté.
- ✓ Coordonner et consolider les programmes de formation et de perfectionnement arrêtés au niveau de l'Entreprise.
- ✓ Suivre l'application correcte des programmes de formation dûment arrêté.
- ✓ Tenir à jour une banque de données sur les organismes formateurs.
- ✓ Tenir à jour les dossiers de formation (ingénierie de la formation et post formation).
- ✓ Programmer les actions de formations et communiquer aux Unités la date, les noms et prénoms des participants à ces formations.
- ✓ Participer aux réunions, avec les organismes formateurs et assister le Chef de Département dans les négociations de contrats de formation.
- ✓ Evaluer les formations dispensées et les organismes formateurs à l'issue de chaque action de formation.
- ✓ Suivre la réalisation des plans d'apprentissage au niveau de chaque unité.
- ✓ Analyser les évaluations des formations dispensées et des organismes formateurs à l'issue de chaque action de formation.
- ✓ Etablir, sur la base des informations recueillies au niveau des différentes Unités, les bilans semestriels afférents à la formation continue et l'apprentissage, et les déposer au niveau du FNAC.
- ✓ Retirer les attestations justifiant l'effort de formation établies par le FNAC.
- ✓ Transmettre à la Direction Finances (Service Fiscalité et Contentieux) les attestations délivrées par le FNAC.
- ✓ Elaborer et suivre les indicateurs qualité se rapportant à ses activités.
- ✓ Participer à l'élaboration des procédures de travail et veiller à leur application.
- ✓ Elaborer le plan d'apprentissage conformément à la réglementation en vigueur en la matière et en fonction des spécialités dispensées par l'Entreprise.
- ✓ Proposer et mener toute action tendant à apporter une amélioration en matière de management de la qualité et ce, dans le cadre des normes et procédures y afférentes.
- ✓ Etablir le rapport d'activités mensuel,

CHAPITRE II : Présentation de l'organisme d'accueil

II.7.6. La problématique

Même avec l'existence d'un grand effectif de travail et une bonne situation informatique, la gestion des formations rencontrent plusieurs problèmes qui ont engendrés le disfonctionnement du département formation, parmi les problèmes rencontrer au sein de se derniers :

- ✓ Etablissement de tous les documents de façon manuelle engendre une perte de temps et une lenteur des services.
- ✓ L'ancienne application qui a était faite ne répond pas aux besoins des utilisateurs vu que l'application ne couvre pas toutes les taches.
- ✓ Plusieurs document sont archivé anarchiquement avec des chemises sur le bureau se qui engendre la difficulté dans la recherche des informations et automatiquement la perte d'information.
- ✓ Insécurité des informations.

II.7.7. Solutions

Le but de notre application consiste à concevoir et réaliser une application web interactif permettant d'offrir une meilleur gestion des formations. Ainsi notre application va permettre :

- D'offrir une interface conviviale et simple d'utilisation.
- D'automatiser les différentes taches liées à l'activité de gestion des formations.
- Gain de temps et simplification de toutes recherches liées à la réalisation des différentes taches du service.
- La sécurisation des informations.

CHAPITRE II : Présentation de l'organisme d'accueil

Conclusion

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

Dans le chapitre qui suit, nous allons présenter l'analyse et conception de notre application.

Chapitre III :

Analyse &

Conception

Introduction

La conception de toute solution logicielle doit être traitée avec précision et détail, précédée d'une analyse profonde et bien réfléchie, car elle est le reflet du futur système avant même sa concrétisation. Dans le but d'avoir une meilleure analyse et de rendre la conception de notre application plus complète, nous avons adopté le processus de développement du logiciel RUP (Rational Unified Process) qui permet de bien représenter l'aspect statique et dynamique d'une application par une série de diagrammes qu'il offre.

III.1. Présentation de la méthode RUP

RUP (Rational Unified Process) est un processus de développement du logiciel couvrant tout le cycle de vie de développement, Son but est d'assurer la production d'un logiciel de grande qualité satisfaisant les besoins de ses utilisateurs finaux dans des délais prédéfini.

Le Rational Unified Process intègre un grand nombre des meilleures pratiques de développement du logiciel moderne sous une forme adaptée à un large éventail de projets et d'organisations.

III.1.1. Le cycle de vie du RUP.

Afin de réaliser notre application de gestion de formation du département gestion des ressources humaines cas: GCB nous avons opté pour une démarche de conception orientée objet RUP, en se basant sur la modélisation UML. Pour cela, nous suivons les phases suivantes :

- ❖ **La phase de création (Inception phase)** consiste à définir l'étendue du projet en spécifiant les cas d'utilisation (périmètre et objectif du projet), et déterminer si la poursuite du projet est possible. Elle se conclut par le jalon LCO (Life cycle Objectif).
- ❖ **La phase d'élaboration** se conclut par le jalon LCA (Life Cycle Architecture). Elle a comme objectif de définir l'architecture du système et l'établir comme référentiel des développements à venir, et aussi d'avoir détaillé 80%des exigences.

Chapitre III : Analyse et Conception

- ❖ **La phase de construction** est concentrée sur la conception détaillée, l'implémentation et les tests. Elle se termine par le jalon IOC (Initial Operation Capability : 1^{ère} version opérationnelle).
- ❖ **La phase de transition** : déploiement.

III.2. Phase d'Initialisation :

Une analyse approximative de notre projet permet de desseller les points suivants :

- Notre application touche la gestion de formation d'une société cas de société national de génie civil et bâtiment.
- Notre application est de type Client/serveur.

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

III.3. Phase d'Elaboration :

Notre projet porte sur la mise en place d'une application client/serveur 2-tiers, pour l'automatisation de la procédure de la gestion des formations cas: Société National de Génie Civil et Bâtiment.

Dans le but de maitriser 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 département formation et les différentes autre départements et directions et de communiquer mutuellement, dans l'objectif de la gestion, et ce via internet.

III.3.1. Identification des acteurs : Durant la période de notre stage au sein du service gestion de formation du département gestion des ressources humaines Cas : GCB à Boumerdes, nous avons pu identifier les principaux acteurs qui seront les futurs utilisateurs de notre application.

Notre système comprend les acteurs suivants :

- Participant.
- Chargé de la Formation RSH.
- Chef de Département formation.
- Directeur de la Structure.

- Directeur des ressources humaines (DRH).
- Président directeur générale (P-DG).

III.3.2 Diagramme de contexte :

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

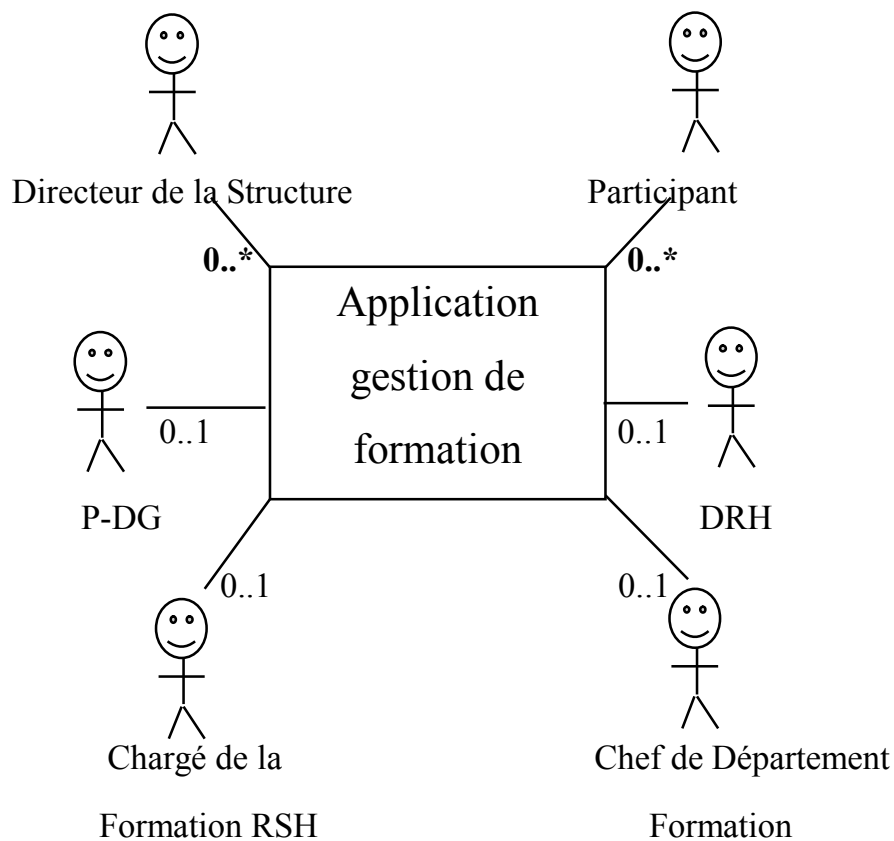


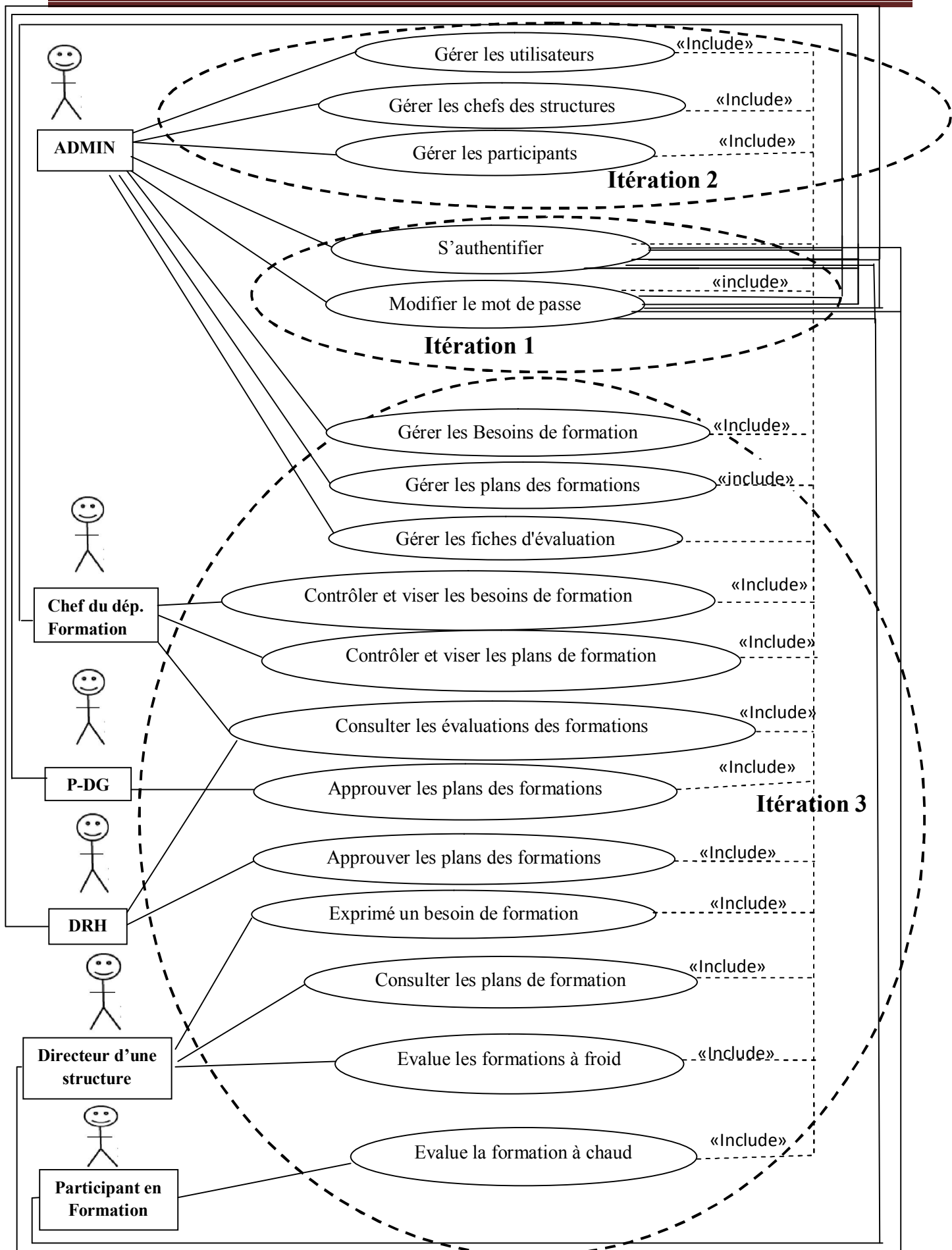
Figure III.1: diagramme de contexte.

III.3.3. Représentation de diagramme de cas d'utilisation

Pour identifier les fonctionnalités du système à concevoir, on utilise le diagramme des cas d'utilisation. Le système n'a qu'un seul acteur externe qui est le développeur. Ce dernier, pour réaliser sa modélisation, il interagit avec le système en utilisant les cas d'utilisation. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs, ils interagissent avec les cas d'utilisation (use cases), les comportements sont modélisés sous la forme de cas d'utilisation via un diagramme de cas d'utilisation.

La figure ci-dessous (III.2) représente le diagramme de cas d'utilisation global de notre application :

Chapitre III : Analyse et Conception



III.3.4 Définition des itérations :

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

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

III.4. Construction :

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

III.4.1. Réalisation de l'itération 1

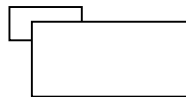
III.4.1.1. Diagrammes de séquence pour l'itération 1:

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

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

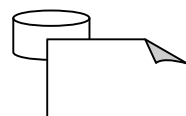
- **Les objets de type interface** : l'interface entre l'acteur et le système, par exemples application web, des pages web complètes.

L'icône utilisée :



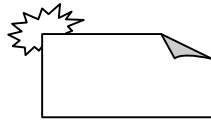
- **Les objets de type entité** : sont des objets décrits dans un cas d'utilisation, qui peuvent apparaître dans de nombreux cas d'utilisation, et généralement sa durée vie dépasse celle de toute Interaction où il participe.

L'icône utilisée :



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

L'icône utilisée :



- ❖ **Diagramme de séquence de cas d'utilisation « S'authentifier » :**

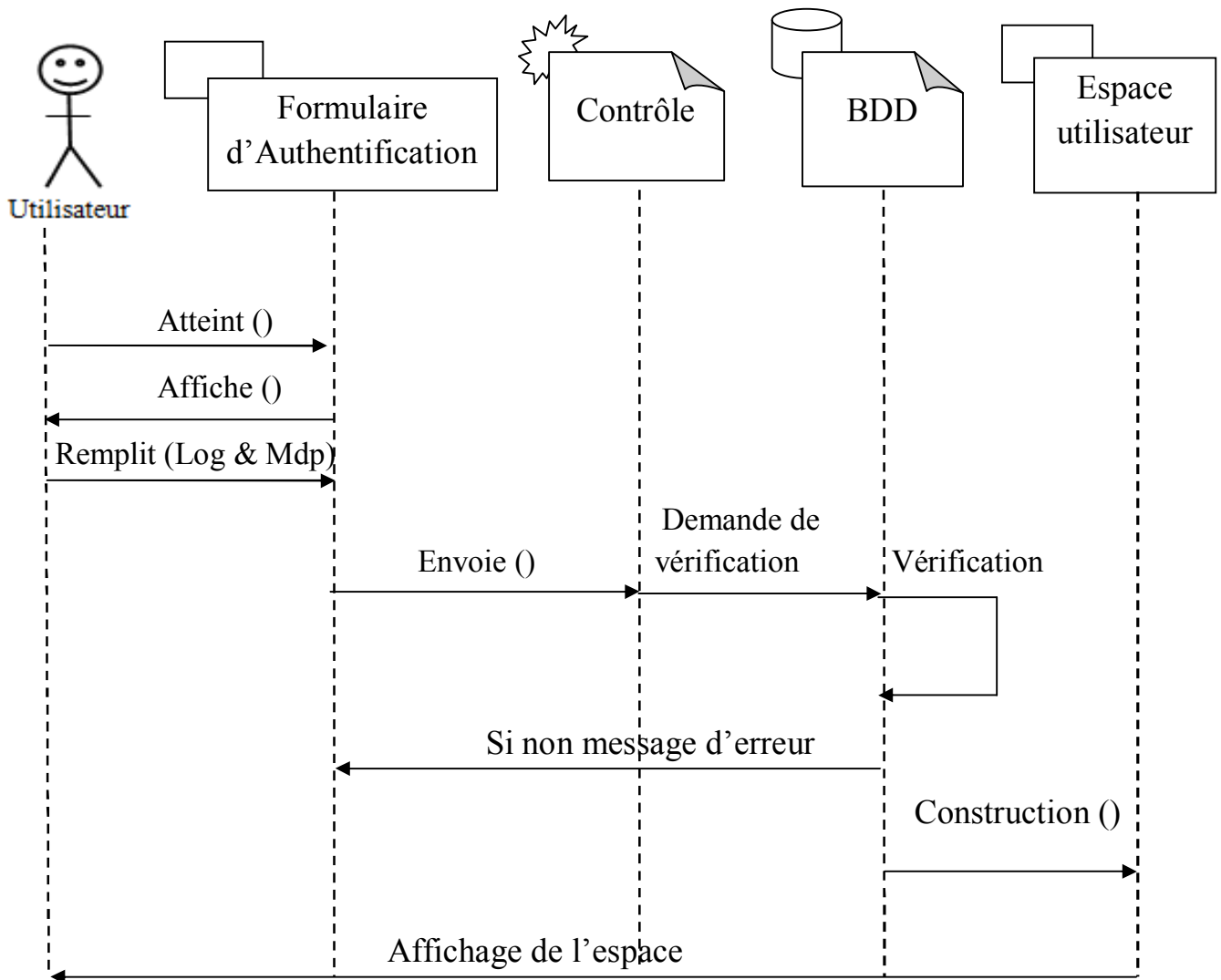


Figure III.3: Diagramme de séquence du cas d'utilisation : « Authentification »

- Description du cas d'utilisation « Authentification » :

Cas utilisation : Authentification

Acteur : Utilisateur

Résumé : Ce cas d'utilisation permet à l'Utilisateur de s'authentifier et d'avoir un espace personnel.

Scénario :

1. L'Utilisateur atteint «espace formation».
2. L'Utilisateur saisit et soumet les données sur la rubrique «authentification» (Identifiant et mot de passe).
3. Le système retourne la page d'utilisateur si les données saisies sont correctes. Si non retourne un message d'erreur.

Tableau III.1 : Description du cas d'utilisation « Authentification »

❖ **Diagramme de séquence de cas d'utilisation « Modification de mot de passe »:**

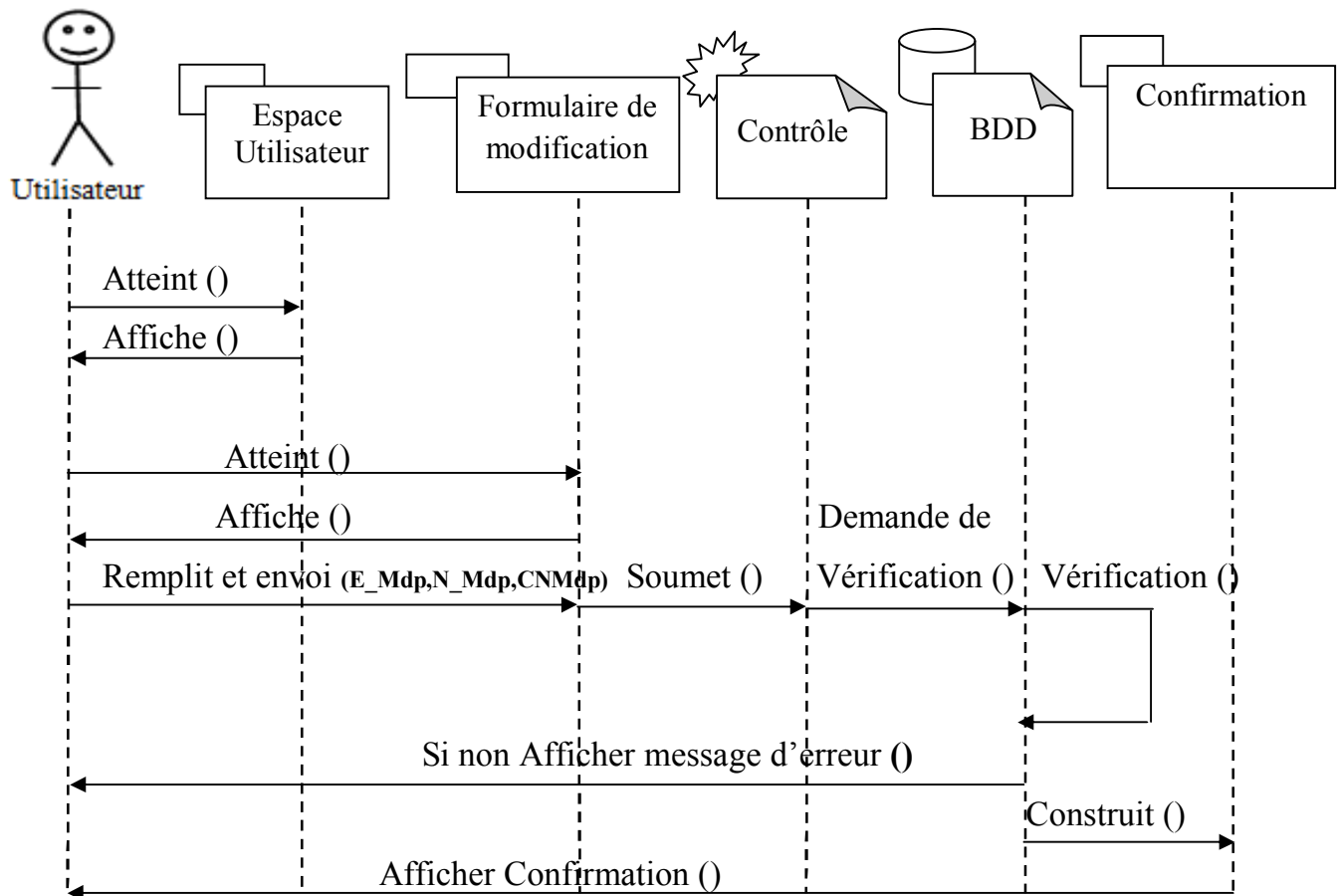


Figure III.4 : Diagramme de séquence du cas d'utilisation :
« Modification de mot de passe »

- **Description du cas d'utilisation « Modification de mot de passe » :**

Cas utilisation : Modification de mot de passe

Acteur : Utilisateurs

Résumé : Ce cas d'utilisation permet à l'Utilisateur de modifier son mot de passe.

Scénario :

1. L'Utilisateur atteint son espace personnel.
2. Le système retourne la page utilisateur
3. L'Utilisateur atteint le formulaire de modification
4. Le système retourne le formulaire
5. L'Utilisateur saisit et soumet les données sur la rubrique mot de passe.
6. Le système retourne un message de confirmation si les données saisit sont correctes. Si non retourne un message d'erreur.

Tableau III.2 : Description du cas d'utilisation « Modification du mot de passe »

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

Utilisateur
LOGIN
Nom
Prénom
MOT-PASS
Fonction
Modifier pw()

Figure III.5 : Diagramme de classes pour l'itération 1

III.4.2 Réalisation de l'itération 2

III.4.2.1 Diagrammes de séquence pour itération 2:

❖ Diagramme de séquence de cas d'utilisation « Ajouter un utilisateur » :

➤ Description du cas d'utilisation : «Ajout d'utilisateur»

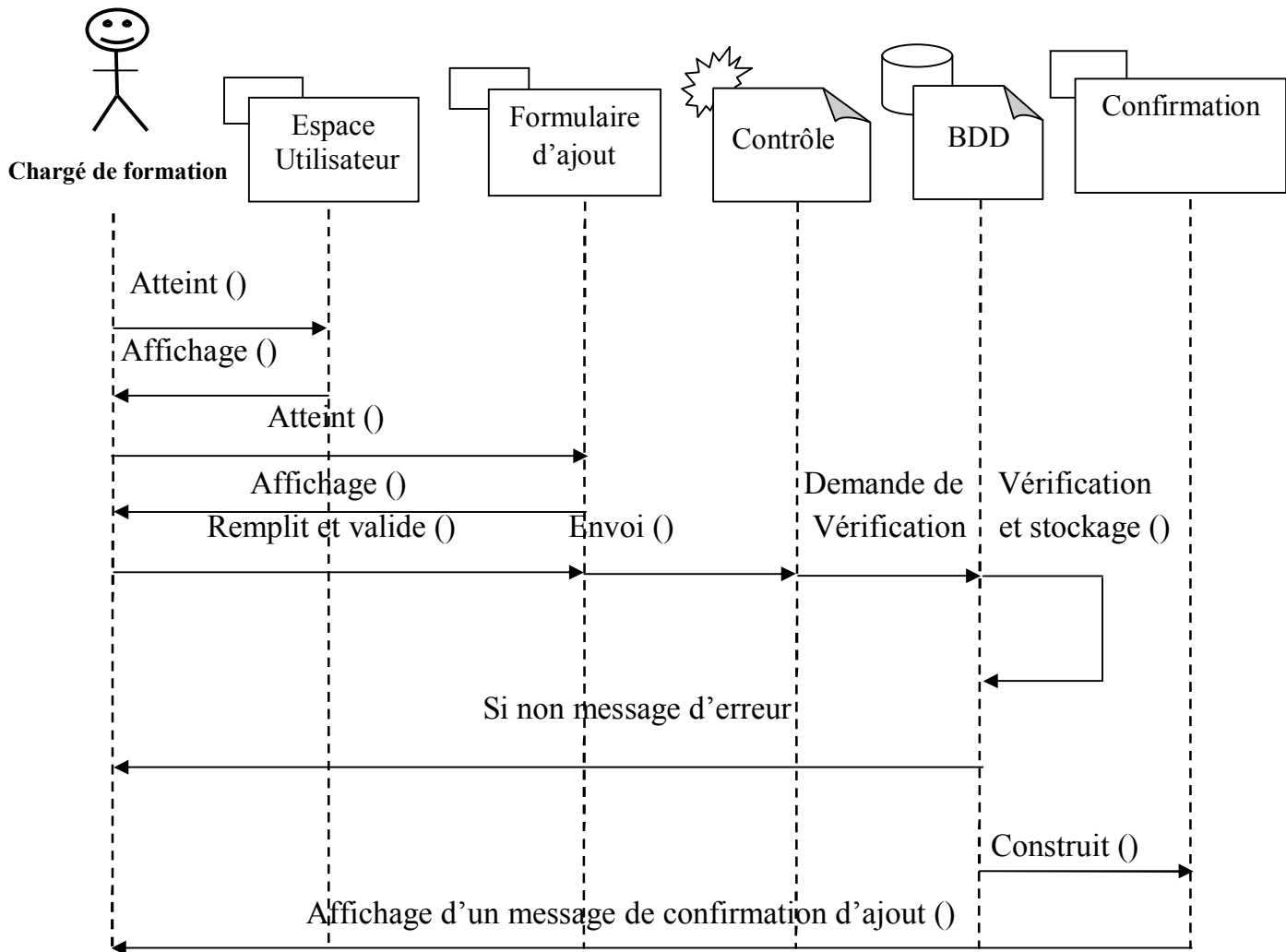


Figure III.6: Diagramme de séquence du cas d'utilisation : « Ajout d'utilisateur »

❖ Diagramme de séquence de cas d'utilisation «Ajout d'utilisateur» :

Cas utilisation : Ajouter d'utilisateur

Acteur : Chargé de formation (Administrateur)

Résumé : Ce cas d'utilisation permet au Chargé de formation (Administrateur) d'ajouter un utilisateur

Scénario :

1. Chargé de formation atteint son espace personnel.
2. Le système retourne la page utilisateur
3. Chargé de formation atteint le formulaire d'ajout d'utilisateur
4. Le système retourne le formulaire
5. Chargé de formation saisit et soumet les données du formulaire.
6. Le système retourne un message de confirmation si les données saisies sont correctes. Si non retourne un message d'erreur.

Tableau III.3 : Description du cas d'utilisation «Ajout d'utilisateur»

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

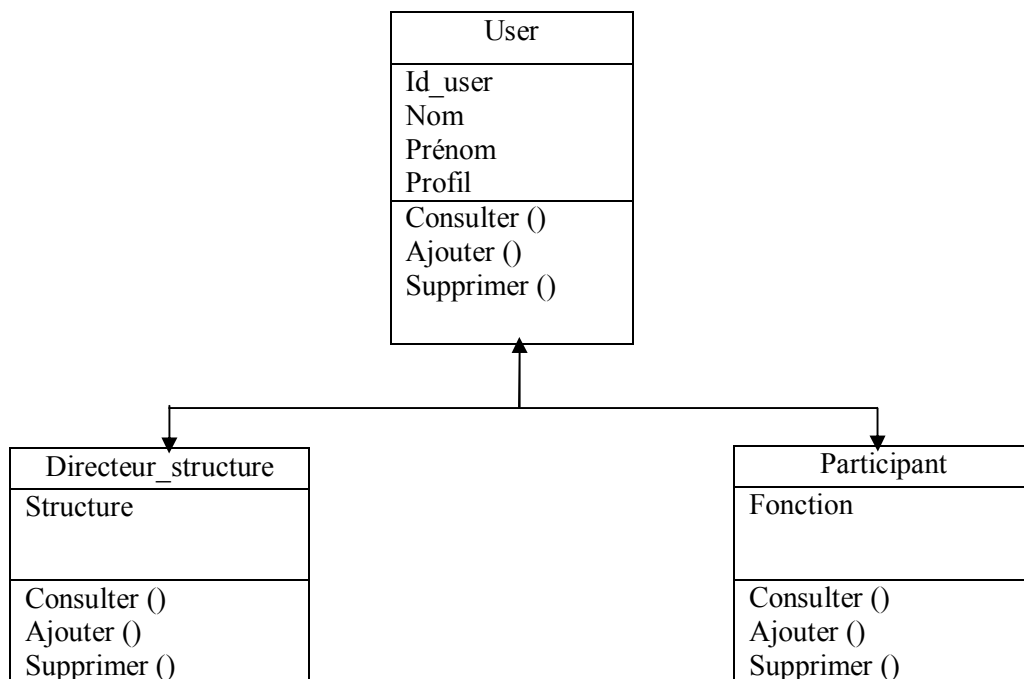


Figure III.7 : Diagramme de classes pour l'itération 2

III.4.3. Réalisation de l'itération 3

III.4.3.1 Diagrammes de séquence pour itération 3:

❖ Diagramme de séquence de cas d'utilisation « Exprimer un besoin d'une formation » :

➤ Description du cas d'utilisation : « Exprimer un besoin d'une formation »

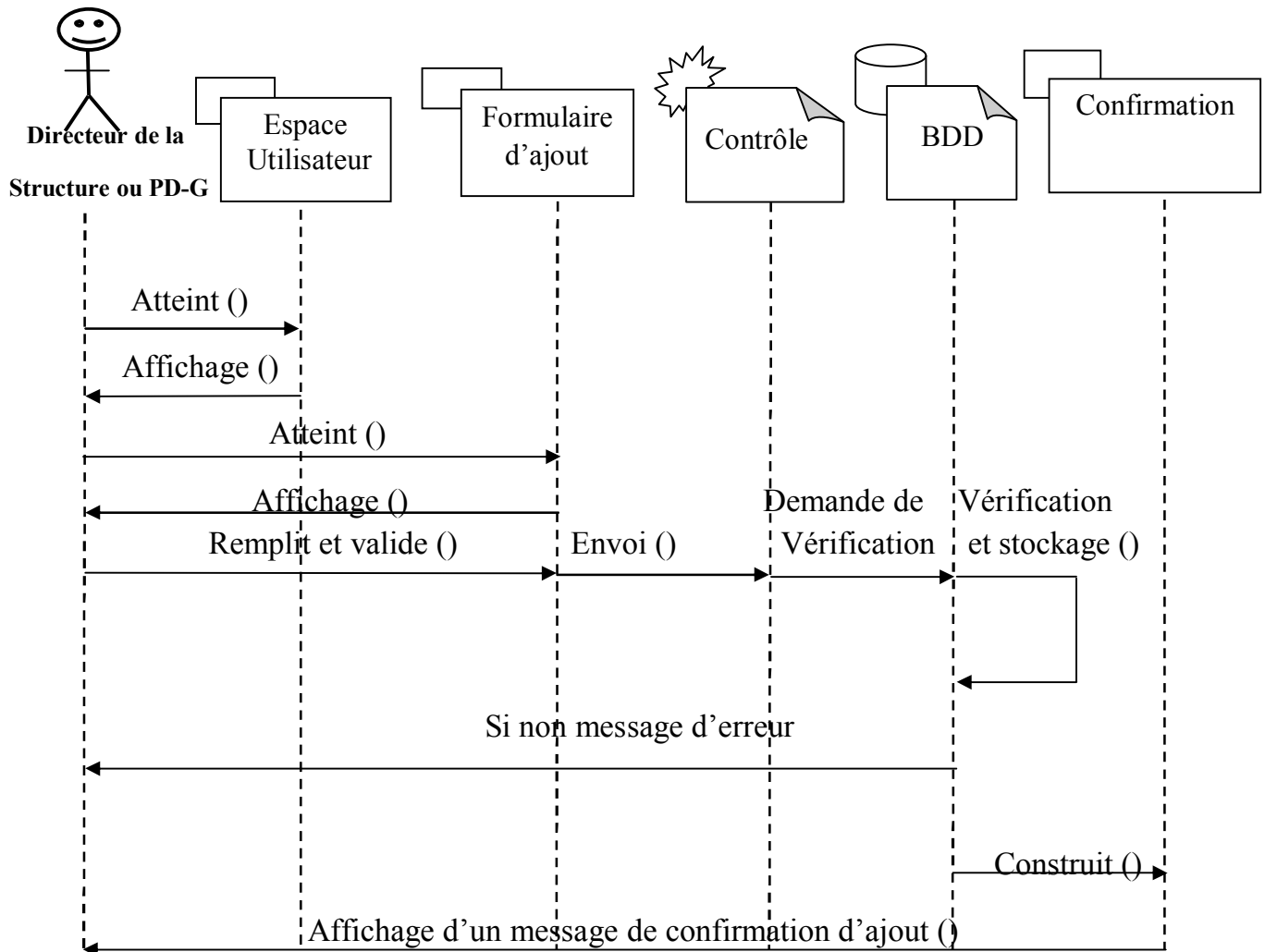


Figure III.8 : Diagramme de séquence du cas d'utilisation : « Exprimer un besoin d'une formation »

➤ Description du cas d'utilisation : « Exprimer un besoin d'une formation »

Cas utilisation : Exprimer un besoin d'une formation

Acteur : Directeur de la structure

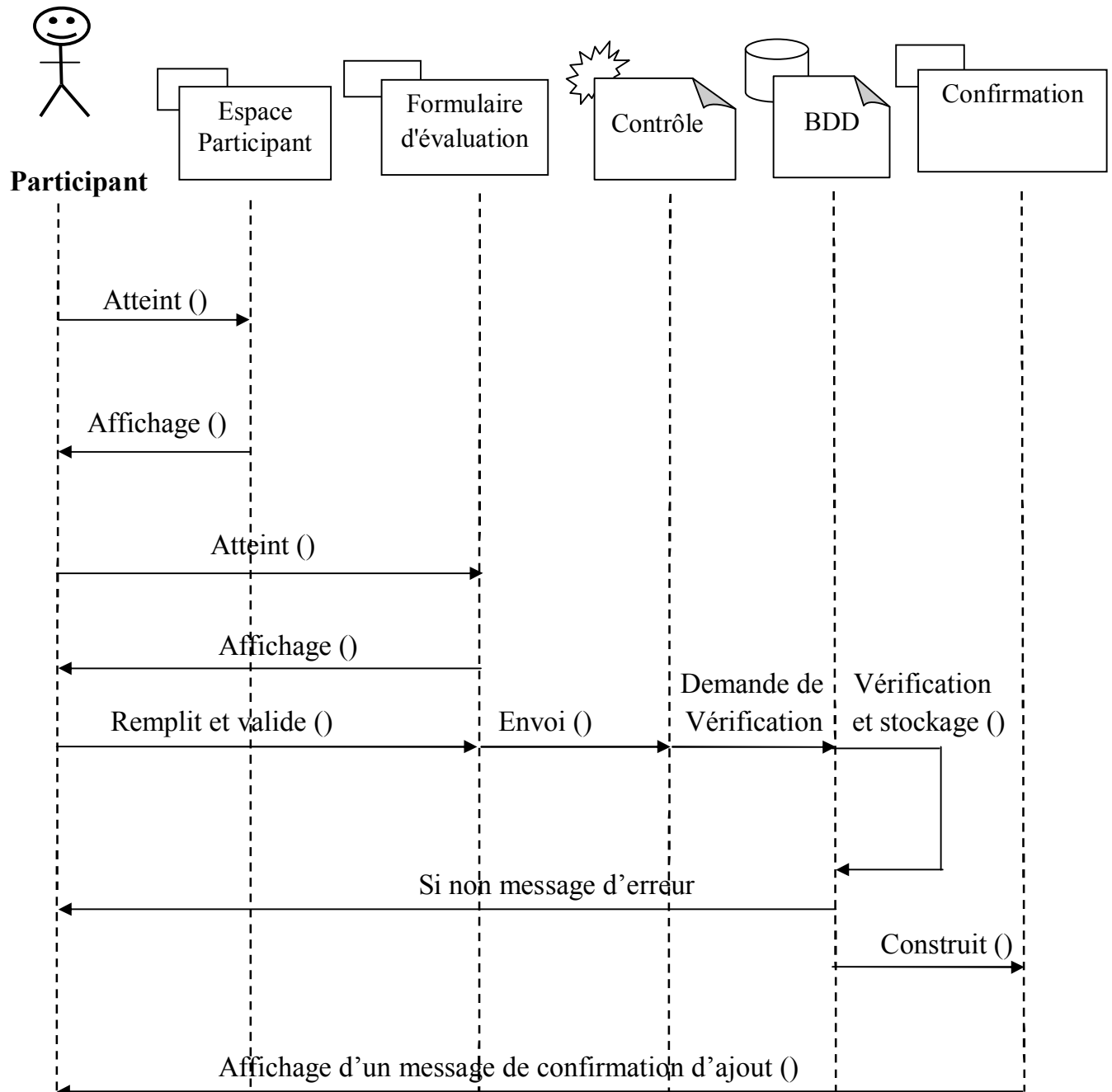
Résumé : Ce cas d'utilisation permet au directeur de la structure (utilisateur) d'exprimer un besoin d'une formation

Scénario :

1. Directeur de la structure atteint son espace personnel.
2. Le système retourne la page utilisateur
3. Directeur de la structure atteint le formulaire d'expression un besoin d'une formation
4. Le système retourne le formulaire
5. Directeur de la structure saisit et soumet les données du formulaire.
6. Le système retourne un message de confirmation si les données saisit sont correctes. Si non retourne un message d'erreur.

Tableau III.4 : Description du cas d'utilisation « Exprimer un besoin d'une formation »

❖ Diagramme de séquence de cas d'utilisation « Evaluer une formation » :



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

➤ **Description du cas d'utilisation : « Evaluer une formation »**

Cas utilisation : Evaluer une formation.

Acteur : Participant.

Résumé : Ce cas d'utilisation permet au Participant d'évaluer une formation.

Scénario :

1. Participant atteint son espace personnel.
2. Le système retourne la page utilisateur
3. Participant atteint le formulaire d'Evaluation.
4. Le système retourne le formulaire d'Evaluation.
5. Participant saisit et soumet les données du formulaire.
6. Le système retourne un message de confirmation si les données saisit sont correctes. Si non retourne un message d'erreur.

Tableau III.5 : Description du cas d'utilisation « Evaluer une formation »

Chapitre III : Analyse et Conception

III.4.3.2. Diagrammes de classe pour l'itération 3 :

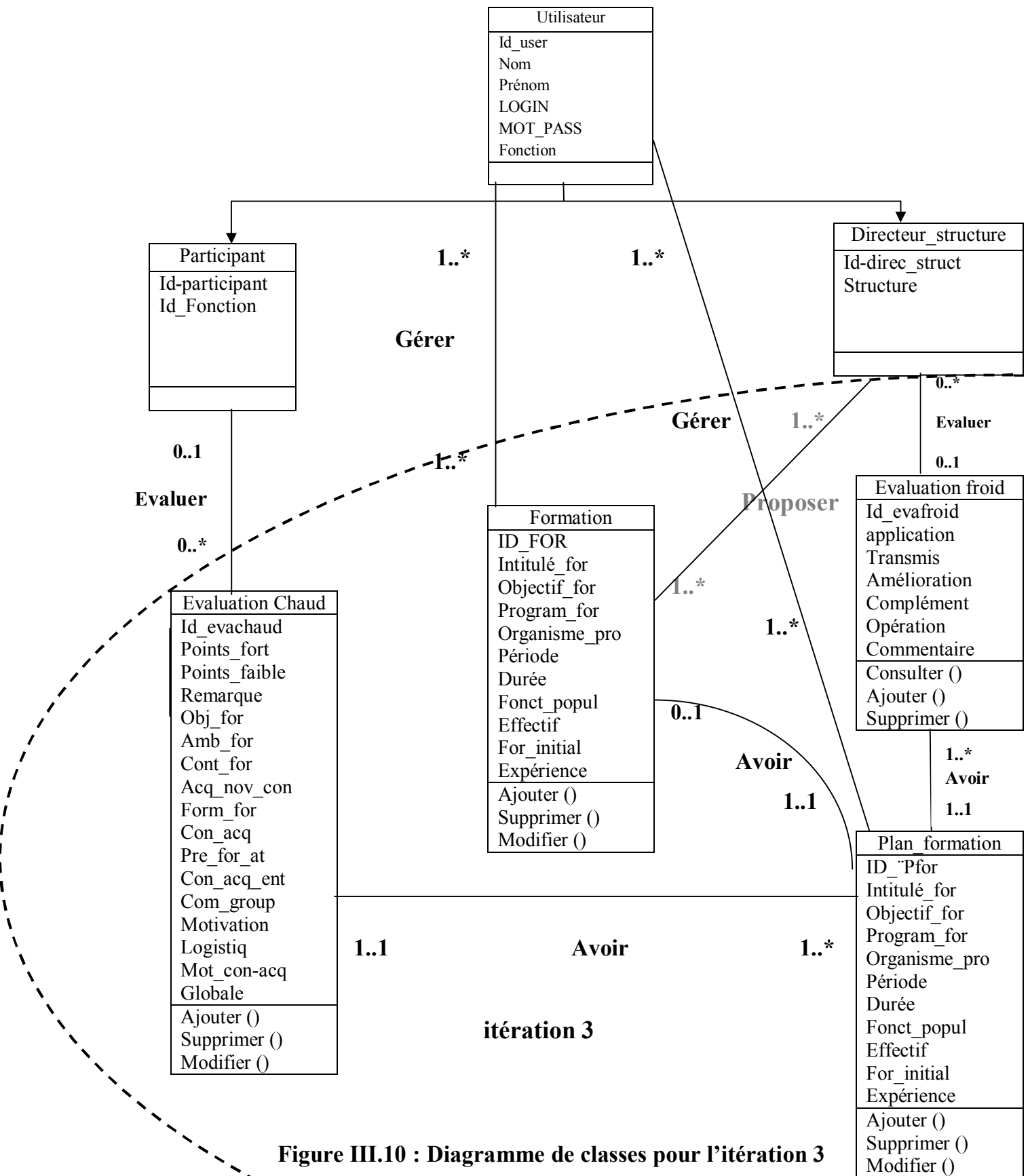


Figure III.10 : Diagramme de classes pour l'itération 3

Chapitre III : Analyse et Conception

Conclusion :

Dans ce chapitre nous avons présenté l'analyse et la conception de notre application en utilisant le processus unifié RUP de langage de modélisation UML. Nous avons commencé par la définition des besoins qui ont été traduits par la suite en formalismes graphiques à l'aide des diagrammes offerts par le processus RUP s'appuyant sur l'analyse.

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

Dans le chapitre qui suit nous présentons la réalisation de notre application.

Chapitre IV :

Réalisation

Introduction

Après avoir présenté dans le chapitre précédent la conception et le fonctionnement global de notre système, nous arrivons dans ce chapitre à la mise en œuvre de notre application.

Ce chapitre sera divisé en deux parties : la première partie sera consacrée à la description de l'environnement et les outils de développement de notre application. La deuxième partie va servir à présenter quelques interfaces.

IV.1. Le modèle relationnel :

Le modèle relationnel présente l'univers des données décrits dans le modèle conceptuel (diagramme de classe), en tenant compte le type de base de données choisie, en d'autre terme le modèle relationnel traduit le modèle conceptuel en un formalisme compressible par la machine.

Le modèle relationnel est basé sur une organisation des données

Utilisateur (Id_user, Nom, Prenom, Login, Mot_passe, Profil).

Participant (Id_part, Fonction, Id_user).

Dir_struct (Id_Direct, Structure, Id_user).

Formation (Id_for, Intit_for, Obj_for, Prog_for, Org_prop, Period_for, Duré_for, Fonc_popu, Effec_popu, For_int, Epr_pro, Id_directeur, Dist).

Plan_for (Id_plan, Intitulé_for, Type_for, Population_con, Effectif, Objectif, Début, Fin, Organisme, Id_directeur, Dist).

Evaluation_froid (Id_froid, connaissance, Transmis, Amélioration, Complément, Commentaire, Id_Direct, Id_plan).

Evalution_chaud (Id_evachaud, Point-for, Point _faible, Remarque, Obj-for, Amb_for, Cont_for, Acq_nov_con, Form_for, Con_acq, Pre_for_at, Con_acq_ent, com_group, Motivation, logistique, Mot-con_acq, Global, Id_plan, Id_par).

Chapitre IV : Réalisation

Remarque:

- Le champ souligné indique la clé primaire de la table
- Le champ suivit du caractère '*' indique une clé étrangère

❖ **Le modèle physique :** Cette étape consiste à implémenter le modèle relationnel dans le SGBD (Système de Gestion de Base de Données), c'est-à-dire le traduire dans un langage de définition de données (LDD).

Le langage utilisé pour cette opération est le SQL. Et plus précisément le langage de définition des données du SQL.

Table 01 : **Utilisateur**

Nom du champ	Description	Type de donnée	Clé(s)
<u>Id_user</u>	Identifiant d'utilisateur	Numérique	Primaire
Login	Login	Caractère(30)	
Nom	Nom de participant	Caractère(30)	
Prenom	Prénom de participant	Caractère(30)	
Mot_passe	Mot de passé	Caractère(30)	
Profil	Profil	Caractère(30)	

Tableau IV.1 : Table Utilisateur.

Table 02 : **Participant**

Nom du champ	Description	Type de donnée	Clé(s)
<u>Id_part</u>	Identifiant de participant	Numérique	Primaire
Fonction	Fonction de participant	Caractère(30)	
Id_user	Identifiant de l'utilisateur	Numérique	étrangère

Tableau IV.2 : Table Participant.

Chapitre IV : Réalisation

Table 03 : **Directeur_structure**

Nom du champ	Description	Type de donnée	Clé(s)
Id_Direct	Identifiant de directeur_structure	Numérique	Primaire
Structure	Fonction de participant	Caractère(30)	
Id_user	Identifiant de l'utilisateur	Numérique	étrangère

Tableau IV.3 : Table Directeur_structure.

Table 04 : **Formation**

Nom du champ	Description	Type de donnée	Clé(s)
Id_for	Identifiant de formation	Numérique	Primaire
Intit_for	Intitulé de l'action de formation	Caractère(255)	
Obj_for	Objectifs de formation	Caractère(255)	
Prog_for	Programme de formation	Caractère(255)	
Org_prop	Organisme propose	Caractère(255)	
Périod_for	Durée du formation	Caractère(30)	
Duré_for	Période du formation	Caractère(30)	
Fonc_popu	Fonction de la Population concernée	Caractère(30)	
Effec_popu	Effectifs de la Population concernée	Numérique	
For_int	Formation initiale de la Population concernée	Caractère(255)	
Epr_pro	Expérience professionnelle de Population concernée	Caractère(255)	
Id_directeur	Identifiant de directeur_structure	Numérique	étrangère
Dist	Destination de la formation	Caractère(30)	

Tableau IV.4 : Table Formation.

Chapitre IV : Réalisation

Table 05 : **Plan_formation**

Nom du champ	Description	Type de donnée	Clé(s)
Id_plan	Identifiant de plan de formation	Numérique	Primaire
Intitul_for	Intitulé du formation	Caractère(255)	
Type_for	Type du formation	Caractère(30)	
Popu_con	Population concernée	Caractère(30)	
Effectif	Effectif de la population concernée	Caractère(30)	
Objectif	Objectifs de formation	Caractère(255)	
Début	Début du formation	Date	
Fin	Fin du formation	Date	
Organisme	Organisme Accueil	Caractère(30)	
Id_directeur	Identifiant de directeur_ structure	Numérique	étrangère
Dist	Distination de la formation	Caractère(30)	

Tableau IV.5 : Table Plan_Formation.

Table 06 : **Evaluation_froid**

Nom du champ	Description	Type de donnée	Clé(s)
<u>Id_froid</u>	Identifiant d'evaluation à chaud	Numérique	Primaire
connaissance	mis en application les connaissances acquises lors de formation	Caractère(3)	
Transmis	transmis les connaissances acquises lors de formation	Caractère(3)	
Amélioration	La formation permis d'améliorer l'efficacité de travail	Caractère(3)	
Complément	compléments seraient nécessaires à la formation suivie, pour l'activité	Caractère(255)	
Commentaire	Commentaires	Caractère(3)	
Id_Direct	Identifiant de directeur de la structure qui évalue la formation	Numérique	étrangère
id_plan	Identifiant de plan de cette formation	Numérique	

Tableau IV.6 : Table Evaluation_Froid.

Chapitre IV : Réalisation

Table 07 : Evaluation_chaud

Nom du champ	Description	Type de donnée	Clé(s)
Id_evachaud	Identifiant d'evaluation à chaud	Numérique	Primaire
Point-for	Points forts du formation	Caractère(255)	
Point_faible	Points faibles du formation	Caractère(255)	
Remarque	Remarques sur la formation	Caractère(255)	
Obj-for	Atteinte des objectifs de la formation	Numérique	
Amb_for	Ambiance générale de la formation	Numérique	
Cont_for	Contenu de la formation	Numérique	
Acq_nov_con	Acquisition de nouvelles connaissances	Numérique	
Form_for	Forme de la formation	Numérique	
Con_acq	Connaissances acquises par rapport à l'attente	Numérique	
Pre_for_at	Prestation du formateur	Numérique	
Con_acq_ent	Connaissances acquises, par rapport aux besoins de l'entreprise	Numérique	
com_group	Composition du groupe	Numérique	
Motivation	motivation à l'application des connaissances acquises	Numérique	
Logistique	Logistique de la formation	Numérique	
Mot-con_acq	motivation à transmettre les connaissances acquises	Numérique	
Global	Appréciation globale de la formation	Numérique	
Id_par	Identifiant de Participant	Numérique	étrangère
Id_plan	Identifiant de plan de formation	Numérique	étrangère

Tableau IV.7 : Table Evaluation_Chaut.

IV.2. Description des outils de développement

IV.2.1. Langage de programmation Java :

Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton de Sun Microsystems. Mais c'est également un environnement d'exécution.

Java peut être séparée en deux parties. D'une part, le programme écrit en langage Java et d'autre part, une machine virtuelle (JVM) qui va se charger de l'exécution du programme Java. C'est cette plateforme qui garantit la portabilité de Java. Il suffit qu'un système ait une machine virtuelle Java pour que tout programme écrit en ce langage puisse fonctionner.

IV.2.2. Java EE

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

IV.2.3. IDE (NETBEANS)

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL et GPLv2 (Common Development and Distribution License). En plus de java, netbeans permet également de supporter différents autres langages, comme python, C, C++, JavaScript, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne

L'environnement de base comprend les fonctions générales suivantes :

- Configuration et gestion de l'interface graphique des utilisateurs.
- Support de différents langages de programmation.
- Traitement du code source (édition, navigation, formatage, inspection...).
- Fonctions d'import/export depuis et vers d'autres IDE, tel qu'Eclipse ou jBuilder.

Chapitre IV : Réalisation

- Accès et gestion de bases de données, serveurs web, ressources partagées.
- Gestion des tâches (à faire, suivi...).
- Documentation intégrée.

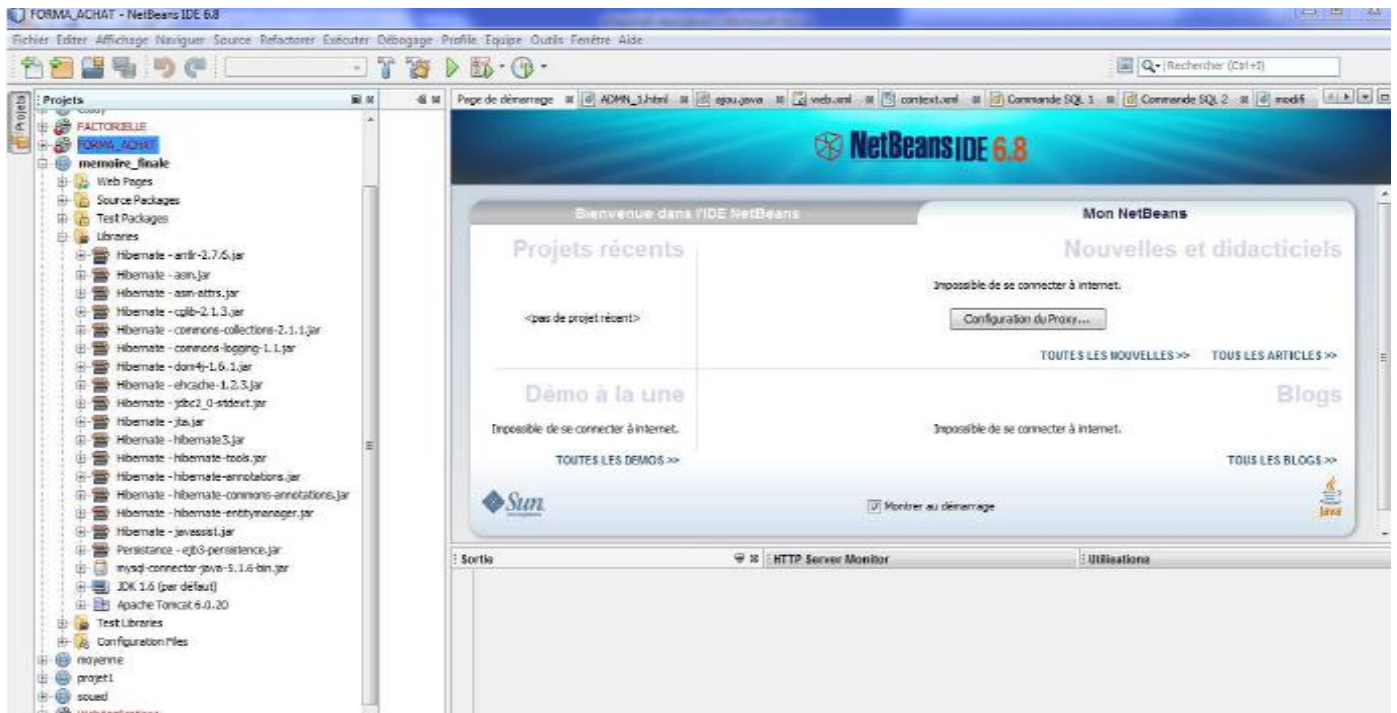


Figure IV.1 : Interface de Netbeans

IV.2.4. Le serveur d'application (Glassfish)

GlassFish est le nom du serveur d'applications Open Source Java EE5 et désormais Java EE 7 avec la version 4.1 qui sert de socle au produit *Oracle GlassFish Server* (anciennement *Sun Java System Application Server*³ de Sun Microsystems).

Sa partie Toplink persistance provient d'Oracle. C'est la réponse aux développeurs Java désireux d'accéder aux sources et de contribuer au développement des serveurs d'applications de nouvelle génération.

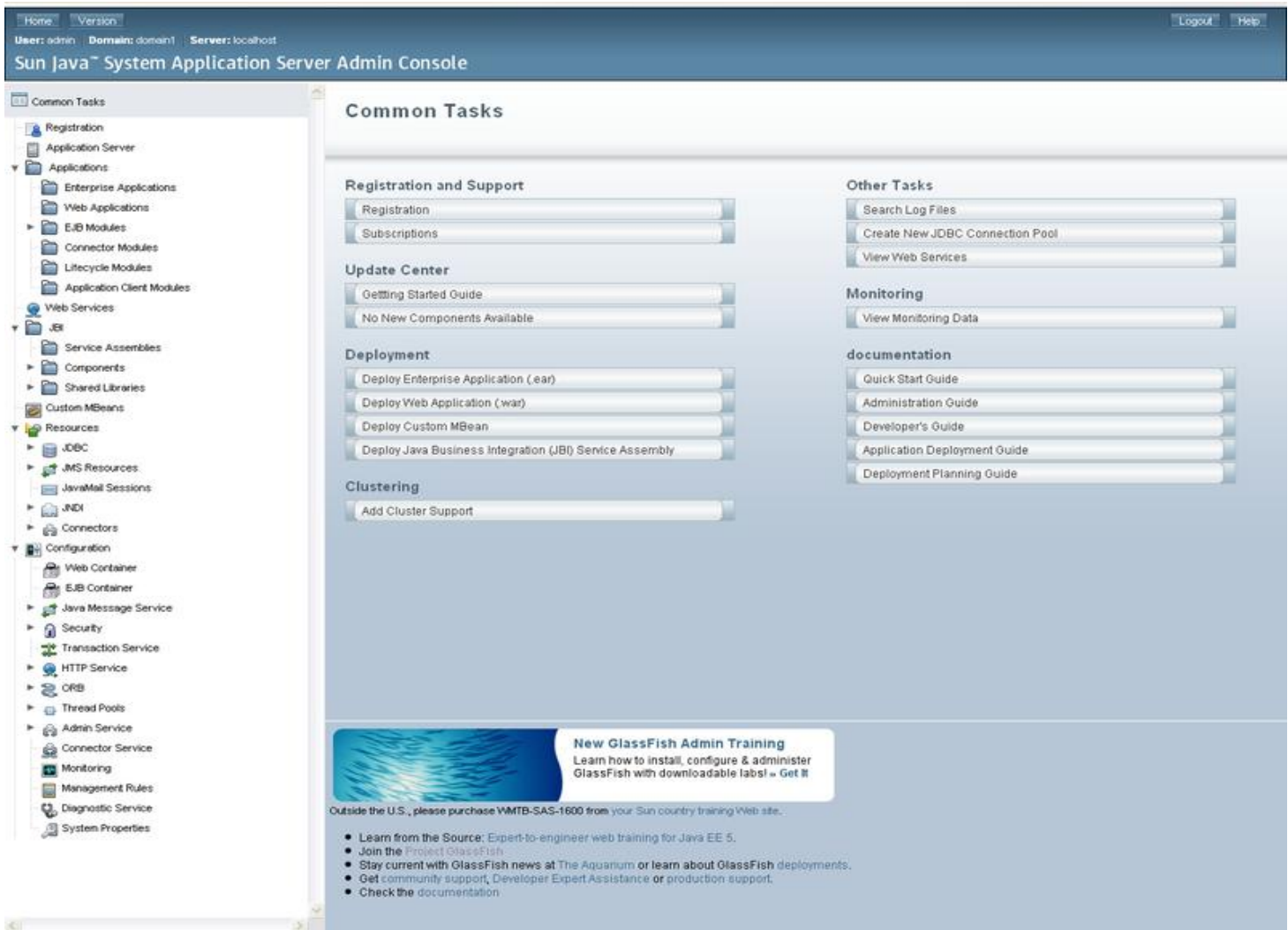


Figure IV.2 : Interface GlassFish

❖ *WampServer* :

WAMP est un acronyme informatique signifiant : (Windows Apache MySQL PHP).

C'est un paquetage qui permet de faire fonctionner des scripts PHP localement sans se connecter à un serveur externe. Ce paquetage est composé à la fois de deux serveurs : Apache et MySQL et un interpréteur de script (PHP), plus deux interfaces: PhpMyAdmin et SQLiteManager pour gérer plus facilement les bases de données.

Les fonctionnalités de WampServer :

- Sont très complètes et très intuitives.
- Gérer les services d'Apache et MySQL.
- Passer en mode online/offline Installer et changer des versions d'Apache.
- MySQL et PHP.
- Gérer les paramètres de configuration des serveurs.
- Accéder aux fichiers de configuration.

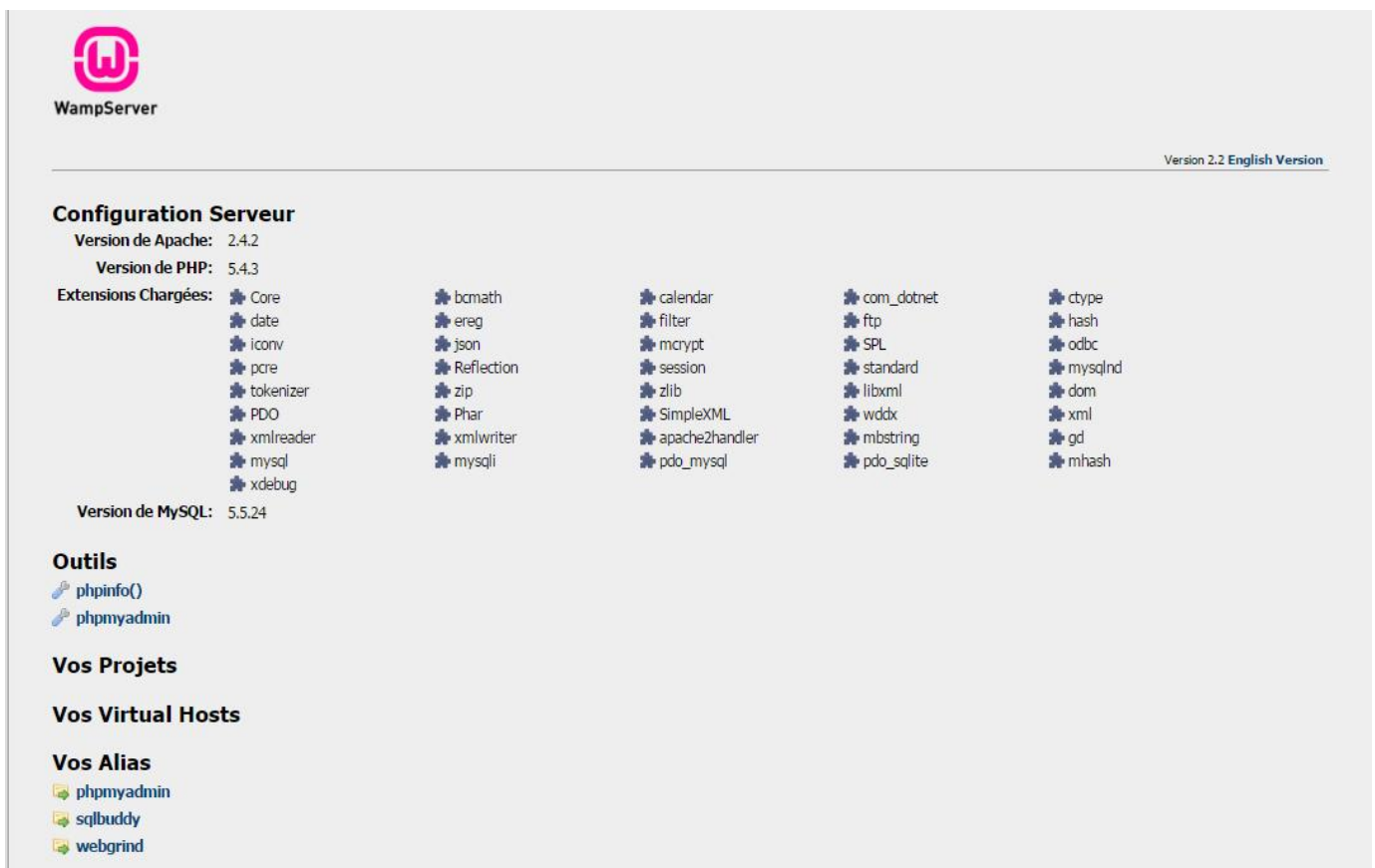


Figure IV.3 : Fenêtre de WampServer

❖ Le serveur MySQL :

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

❖ PhpMyAdmin : [6]

PhpMyAdmin est une application web permettant de gérer un serveur de bases de données MySQL. Cette interface est écrite en PHP et s'appuie sur le serveur Apache.

La figure suivante montre une copie d'écran de la page d'accueil de PhpMyAdmin.

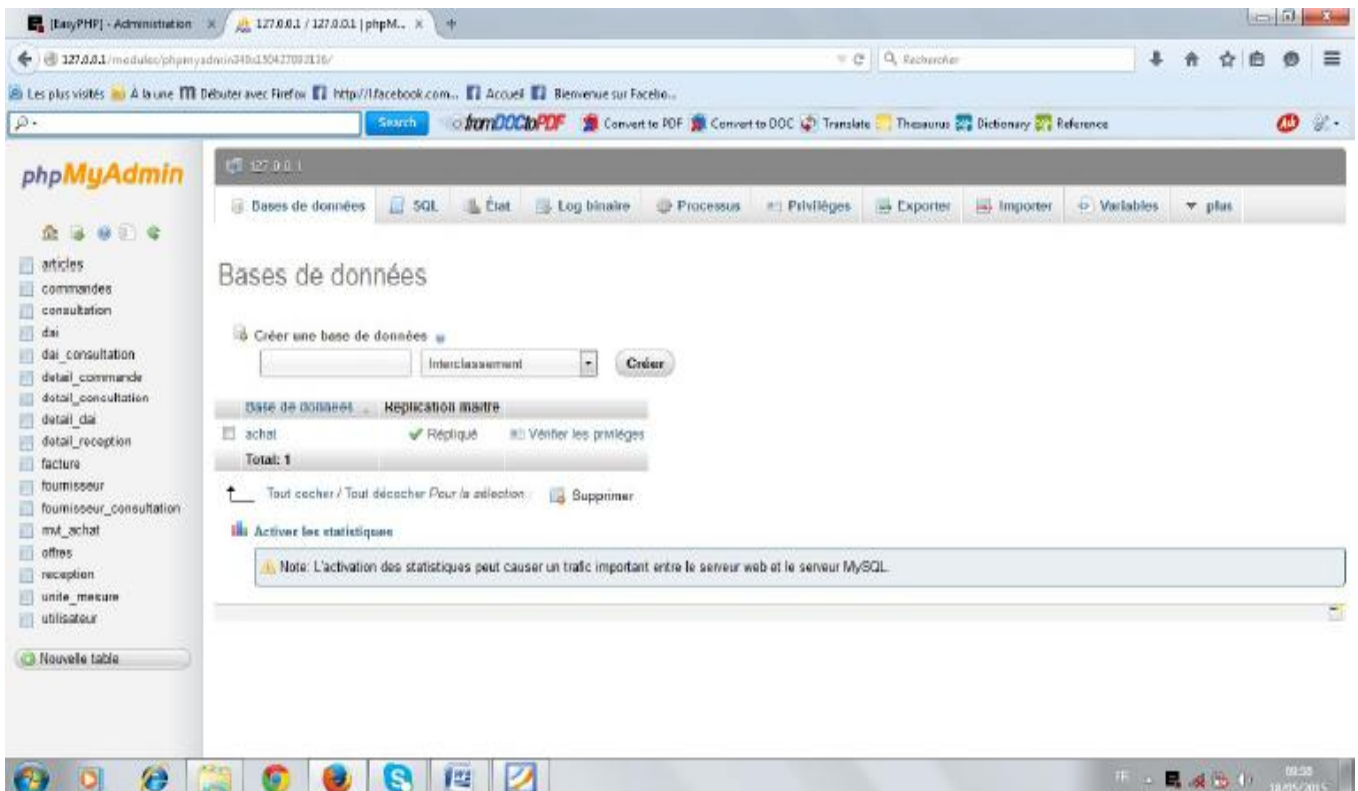


Figure IV.4: Interface PhpMyAdmin.

IV.3. Les Langages du web :

IV.3.1. Le coté client : [7]

- **HTML** : Les informations transmises par le serveur Web au navigateur via http sont portées au regard de l'utilisateur. La présentation et la mise en forme des documents sont codifiées par un langage nommé HTML. Depuis ses premiers développements, Html inclut des éléments qui permettent la communication entre un client web et un serveur web capable de recevoir des informations de ce client, de les stocker et /ou des traitements en dépendant.
- **JAVASCRIPT** : Est un langage de script orienté objet. Il s'appuie sur la DOM développé par le consortium W3. Java script définit des méthodes qu'il est possible d'invoquer sur ces objets.
- **SQL** : Est un langage qui permet de formuler des requêtes pour mettre à jour, interroger les bases de données pour en tirer un certain nombre d'information.
- **CSS :(les feuilles de style)** : Est un ensemble de règles utilisés par le navigateur pour la mise en page et en forme des différents éléments.

IV.3.2. Le coté serveur :

Les dernières technologies Web coté serveur sont ASP (Active Server Page), les Servlets Java, JSP (Java Server Page) et PHP (Personnel Home Page).

Notre choix s'est focalisé sur les JSP et les Servlets dont nous allons donner des brèves définitions.

- **Les Servlets** : Une servlet est un programme java qui utilise des modules supplémentaires figurant dans l'API java. Son exécution se fait dans un moteur de Servlet ou conteneur de Servlet permettant d'établir le lien entre la Servlet et le serveur Web. Les servelets permettent donc de gérer des requêtes HTTP et de fournir au client une réponse HTTP dynamique (donc de créer des pages Web dynamiques).

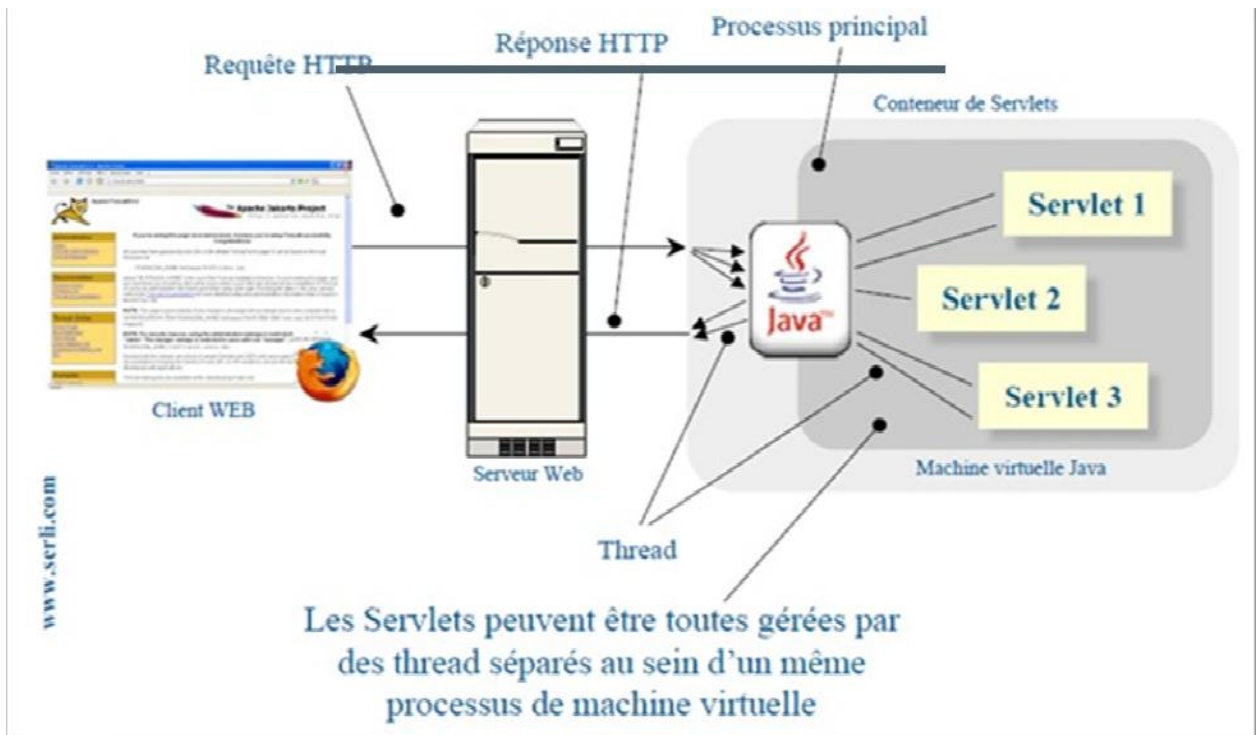


Figure IV.5: Architecture d'une Servlet

- **Les JSP :** Une JSP est un fichier contenant du code HTML et des fragments de code Java exécutés sur le moteur de Servlet. Les JSP sont compilées par un compilateur JSP pour devenir des servlets Java.

IV.4.Présentation de quelques interfaces de l'application

❖ Page d'accueil

C'est la première page qui apparaît au lancement de l'application.



Figure IV.6 : Interface page d'accueil.

❖ Interface d'authentification :

Cette page est nécessaire pour l'authentification des différents acteurs qui vont utiliser cette application, ce qui va leur permettre d'accéder à leur espace personnel.



Figure IV.7 : Interface d'authentification.

❖ **Interface d'Espace chargé de formation (Administration) :**

Cette page permet au chargé de la formation d'accéder aux différentes pages de son propre espace afin d'effectuer les différents traitements.



Figure IV.8: Interface espace chargé de formation

❖ **Formulaire Ajout de formation :**

Elle représente le formulaire d'ajout de formation que le P-DG remplit selon les critères de la formation à ajouté.

Intitulé de l'action de formation :

Objectifs de formation :

Programme de formation :

Organisme proposé :

Durée du formation :

Période du formation :

Population concernée :

A. Fonction :

B. Effectifs :

C. Formation initiale :

D. Expérience professionnelle :

AJOUTER

Copyright@2016 GCB

Figure IV.9 : Interface du formulaire ajout de formation

❖ **Formulaire Evaluation d'une formation :**

Cette page permet au participant d'évaluer la formation.

Merci d'estimer le degré de votre satisfaction et de donner pour chaque critère une note sur 20			
Atteinte des objectifs de la formation	1 ▼	Ambiance générale de la formation (participants, dialogue)	1 ▼
Contenu de la formation (sujets, volume, intensité)	1 ▼	Acquisition de nouvelles connaissances	1 ▼
Forme de la formation (document, projection, durée)	1 ▼	Connaissances acquises par rapport à votre attente	1 ▼
Prestation du formateur	1 ▼	Connaissances acquises, par rapport aux besoins de l'entreprise	1 ▼
Composition du groupe (profil et nombre de participants)	1 ▼	Votre motivation à l'application des connaissances acquises	1 ▼
Logistique de la formation (salle, pause, repas, horaires)	1 ▼	Votre motivation à transmettre les connaissances acquises	1 ▼
Votre appréciation globale de la formation			
1 ▼			
VALIDER			

Copyright©2016 GCB

Figure IV.10 : Interface du formulaire évaluation d'une formation

❖ Formulaire ajout plan de formation :

Ce formulaire permet au chargé de formation d'ajouter un plan de formation.

Intitulé de l'action de formation:

Type de formation:

Population concernée:

Effectif de population:

Objectif de formation:

Début de formation:

Fin de formation:

Organisme de formation:

Copyright © 2016 GCB

Figure IV.11: Interface du formulaire ajout plan de formation.

Conclusion

Nous avons abordé dans ce chapitre les différents outils de développement et les langages de programmations utilisés pour le développement dans notre application ainsi que quelques interfaces graphique qui illustre notre projet.

Conclusion Générale

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

La réalisation de ce travail au sein de l'entreprise GCB nous a permis d'approfondir nos connaissances en systèmes d'information, de découvrir le domaine de gestion des ressources humaines, de voir sur le champ de travail, les différentes tâches effectuées par les gestionnaires des ressources humaines et d'autres part d'enrichir nos connaissances telle que la programmation avec le langage JAVA en se basant sur les concepts d'UML pour la réalisation de notre application.

Nous espérons que l'intégration de notre travail au niveau de service des ressources humaines va répondre aux besoins et va servir d'outil d'aide et d'efficacité pour l'exécution de leur tâches.

Enfin, nous nous souhaitons que cette application soit un support de documentation supplémentaire pour les nouvelles promotions

Annexe

Introduction :

UML n'est pas une méthode, mais un langage. UML n'est pas limité à l'objet, mais peut supporter la majorité des méthodes existantes. Cette double limite explique le succès d'UML. Ce chapitre représente les concepts et les outils de ce langage unifié de modélisation et le processus de développement RUP (Rational Unified Process).

I. UML.

I.1. Historique :

A) Origine : UML est né de la consolidation de trois méthodes objet : OMT (Rumbaugh), booch, et OOSE (Jacobson). Cette consolidation a été marquée par trois étapes :

- ❖ Le regroupement des trois équipes au sein de la société Rational.
- ❖ Le recentrage du projet de standardisation sur le langage de modélisation, les aspects purement méthodologique étant laissés de côté.
- ❖ La décision de l'objet management Group en 1997.

Le succès a été immédiat et UML est aujourd'hui universellement accepté et supporté par l'ensemble des outils de développement. [8]

B) positionnement : Comme son nom l'indique UML est un langage de modélisation. Ce n'est pas une méthode (il y manque la démarche), et UML n'est pas propre à l'approche objet (une part importante des concepts et des outils est empruntée aux méthodes classiques).

Cette double limite explique la rapidité avec laquelle UML a été adopté par l'ensemble des acteurs concernés par le génie logiciel : UML fournit un cadre conceptuel commun mais laisse aux différents acteurs du marché toute liberté pour développer leurs méthodes et leurs outils. [8]

I.2. Définition :

UML (unified modeling language), est un ensemble de notations graphiques (modèles) qui s'appuient sur une syntaxe (méta-modèle). L'objectif initial était de permettre aux informaticiens de représenter un système logiciel et son utilisation prévue dans l'entreprise, afin d'améliorer la qualité des applications informatiques qu'ils développent. Le méta-modèle permet notamment d'assurer que tous les outils de génie logiciel orientés UML présentent une cohérence et autorisent l'interopérabilité des modèles, c'est-à-dire leurs échange entre ateliers.

Aujourd'hui, UML se présente comme un langage pouvant être utilisé dans des domaines variés. [9]

I.3. Modélisation UML :

UML fournit des outils permettant de représenter l'ensemble des éléments du monde réel (classes, objets,.....etc) ainsi les liens qui les relient, il s'articule autour de plusieurs types de diagrammes tel que : [9]

- ❖ **Diagramme de cas d'utilisation** : les diagrammes de cas d'utilisation représentent les cas d'utilisations identifiés et l'acteur associé à chacun. Ils permettent notamment de représenter les processus d'un domaine.
- ❖ **Diagramme de séquence** : le diagramme de séquence représente des échanges de message entre objet. Il permet de mieux visualiser la séquence de messages par une lecture de haut en bas.
- ❖ **Diagramme classe** : le diagramme de classe représente la structure statique d'un système. Il contient principalement de classes, ainsi que leurs associations, mais on peut aussi y trouver des objets.

- ❖ **Diagramme d'activité :** c'est un graphe orienté qui décrit un enchainement de traitements. Le déroulement ainsi présenté est appelé « flot de contrôle ».
- ❖ **Diagramme d'objet :** le diagramme d'objet permet de mettre en évidence des liens entre les objets. Les objets, instances de classes, sont reliés par des liens, instances d'associations. Le diagramme d'objet utilise les mêmes concepts que le diagramme de classe.
- ❖ **Diagramme de paquetage :** le paquetage proposé par UML est une technique qui permet de mettre en œuvre ce partitionnement du modèle tout en préservant la cohérence de l'ensemble.
- ❖ **Diagramme d'état :** représente des états permettant de gérer le domaine étudié. Ces traitements sont positionnés par rapport à des classes et plus précisément par rapport à des états des classes. Il met en évidence l'enchainement des différents états d'une classe, fait ainsi apparaître l'ordonnancement de ces différents travaux.
- ❖ **Diagramme de communication :** le diagramme de communication fait apparaître les interactions entre des objets du système et les messages qu'ils échangent.

II. Méthode RUP de l'UML.

II.1. Présentation de la méthode RUP

RUP (Rational Unified Process) est un processus de développement du logiciel couvrant tout le cycle de vie de développement, Son but est d'assurer la production d'un logiciel de grande qualité satisfaisant les besoins de ses utilisateurs finaux dans des délais prédéfini.

Le Rational Unified Process intègre un grand nombre des meilleures pratiques de développement du logiciel moderne sous une forme adaptée à un large éventail de projets et d'organisations.

RUP reste de loin le processus logiciel le plus utilisé et le plus adapté pour les très grands projets. Il utilise une approche itérative et il est fondé sur les best practices (bonnes pratiques).

RUP est piloté par les cas d'utilisation. Ils sont décrits à l'aide d'une représentation spécifique à RUP, plus riche que celle contenue dans UML. Ses forces sont essentiellement :

- Cadre générique.
- Référentiel de bonnes pratiques.
- Gestion des risques dans les projets.
- Cadre propice à la réutilisation.
- Approche basée sur l'architecture.
- Traçabilité à partir des Uses Cases jusqu'au déploiement.

II.2.Caractéristiques de RUP

Le processus unifié [[http://www-01. ibm.com/software/rational/](http://www-01.ibm.com/software/rational/)] a trois caractéristiques qui sont les suivantes:

- ❖ **Conduit par les cas d'utilisation(Use-Case Driven)** : Le processus emploie les Use Cases à conduire le processus de développement depuis sa création jusqu'au déploiement.
- ❖ **Centré Architecture:** Le processus vise à comprendre les aspects les plus significatifs statique et dynamique en termes d'architecture logicielle. L'architecture est une fonction des besoins des utilisateurs et est capturé dans le cas d'utilisation de base.
- ❖ **Itératif et incrémental:** Le processus reconnaît qu'il est pratique de diviser de grands projets en petits projets ou des mini-projets. Chaque mini-projet comprend une itération qui se traduit par une augmentation d'échelon. Une itération peut englober tous les workflows dans le processus. Les itérations sont prévues, avec les Use Cases.

II.3. Modèle d'un processus RUP.

Un processus décrit qui fait quoi, quand et comment. Le processus unifié de rational est représenté par 5 éléments élémentaires :

- ✓ Rôles : le qui
- ✓ Activités : le comment
- ✓ Artefacts : le quoi
- ✓ Workflows : le quand
- ✓ Discipline s: le conteneur des quatre types d'éléments précédents

III.3.1. Les rôles

C'est le concept essentiel du processus. Il définit le comportement et les responsabilités d'un individu ou d'un groupe d'individus travaillant en équipe. Le comportement est exprimé en termes d'activités que le rôle performe. Les responsabilités sont exprimées en relation d'artefacts que le rôle crée, modifie ou contrôle.

II.3.2. Les activités

Une activité est l'unité de travail qu'un rôle performe et qui produit un résultat significatif dans le contexte du projet. L'activité a un but clair, habituellement exprimé en termes de créer ou mettre à jour des artefacts, tels qu'un modèle, une classe, ou un plan.

II.3.3. Les artefacts (Work products)

Sont utilisés comme entrée par les rôles pour accomplir une activité, et comme résultats pour ces activités.

Les Artefacts sont de plusieurs types:

- **Un modèle**, comme le modèle des use-case ou le modèle conceptuel.
- **Un élément modèle** (un élément dans le modèle) comme une classe, un use case, ou un sous système.
- **Un document**, comme un business case ou software architecture document.
- **Un code Source**
- **Des exécutables**

II.3.4. Les Workflows.

C'est une séquence d'activités qui produit un résultat d'une valeur observable. En termes d'UML, il peut être exprimé comme un diagramme de séquence, de collaboration ou d'activité.

Rational Unified Process utilise trois types de workflow :

- **Le Noyau de workflows (Core workflows)**, associé à chaque discipline. Dans chaque discipline il existe un « core workflow » qui donne globalement le flux des activités.
- **Les détails de Workflow (Workflow details)**, pour affiner le déroulement du core workflow. Chaque « core workflow » couvre beaucoup d'activités, on utilise « Workflow details » pour exprimer un groupe spécifique d'activités étroitement liées.
- **Les plans d'itération (Iteration plans)** qui sont d'autres moyens pour présenter le processus en le décrivant d'avantage dans la perspective de ce qui se produit dans une itération typique. Ils sont réellement les plus proches de ce qu'un moteur de workflow manipulerait. On peut les considérer comme des instanciations du processus pour une itération donnée, choisissant les activités qui seront effectivement gérées pendant l'itération et les répliquées selon les besoins.

II.3.5. Les disciplines.

Les disciplines sont des conteneurs utilisés pour organiser des activités du processus. Il y a neuf disciplines dans le processus RUP, et ils représentent une division de tous les rôles et activités dans des groupements logiques par des sujets de préoccupation ou la spécialité. Les neuf disciplines de noyau sont divisées en six disciplines techniques et trois disciplines de support. Six disciplines techniques "engineering disciplines" (Modélisation d'activités « Business Modeling », Expression des besoins, Analyse et conception, Implémentation, Test et Déploiement) et trois Disciplines de support « supporting disciplines» (Gestion de configuration, Gestion de projet et Environnement).

II.3.6. Des éléments additionnels du processus

Rôles, activités, workflows et artefacts, organisés dans les disciplines, représentent l'épine dorsale (the backbone) de la structure statique du processus RUP. Mais d'autres éléments sont ajoutés à des activités ou des artefacts pour rendre le processus plus facile à comprendre, à utiliser et à fournir des conseils plus complets pour les praticiens. Ces éléments de processus supplémentaires sont : les directives (Guidelines), les modèles (Templates) les experts d'outils (Tool mentors) et les concepts.

II.4. Le cycle de vie du RUP

Un cycle est constitué de quatre phases, exécutées dans l'ordre : création, élaboration, construction et transition.

La phase de création (Inception phase) consiste à définir l'étendue du projet en spécifiant les Cas d'utilisation (périmètre et objectif du projet), et déterminer si la poursuite du projet est possible

La phase d'élaboration Elle a comme objectif de définir l'architecture du système et l'établir comme référentiel des développements à venir, et aussi d'avoir détaillé 80% des exigences.

La phase de construction est concentrée sur la conception détaillée, l'implémentation et les tests

La phase Transition : déploiement

Lorsqu'on finit une itération et que certains éléments sont validés on peut parfois passer à la phase suivante. Rappelons que pour chaque phase on exécute des itérations, et pour chaque itération on exécute l'ensemble des disciplines.

II.5. L'approche RUP.

II.5.1. L'identification de buts et le diagramme d'hierarchie de buts

L'objectif principal de la modélisation avec RUP est d'assister le développeur pendant sa modélisation. Assister est synonyme de guider, fournir des explications et contrôler les intervenants dans un projet au déroulement. On peut identifier les autres buts importants comme suit :

Assister le développeur dans:

A) La réalisation d'un projet

- La réalisation des différentes phases.
- La réalisation des différentes disciplines.
- La réalisation des différentes itérations d'un projet.
- La réalisation des différents groupes d'activités d'une itération.

B) L'assistance d'un rôle

- La réalisation des différentes activités d'une itération.
- La réalisation des différentes étapes d'une activité.
- La production, la consultation ou la mise à jour d'un artefact.

Et éventuellement, assister le développeur (ou une équipe de développeurs) dans ces différentes tâches et prendre des décisions de modélisation, comme l'anticipation de quelques étapes d'une activité...etc. Le diagramme suivant (Figure 01) montre une hiérarchie possible des buts.

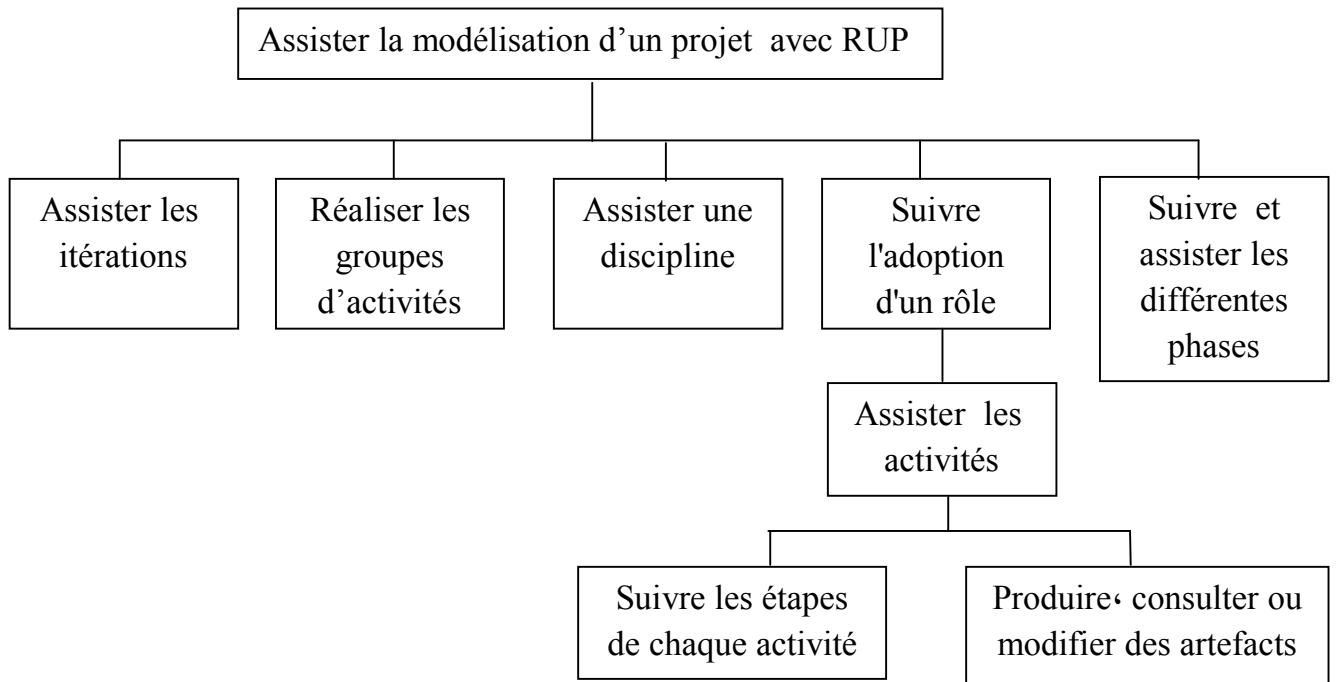


Figure 01 : Diagramme d'hierarchie de buts

II.5.2. Les cas d'utilisation

Pour identifier les fonctionnalités du système à concevoir, on utilise le diagramme des cas d'utilisation. Le système n'a qu'un seul acteur externe qui est le développeur. Ce dernier, pour réaliser sa modélisation, il interagit avec le système en utilisant les cas d'utilisation montrés dans la figure 02.

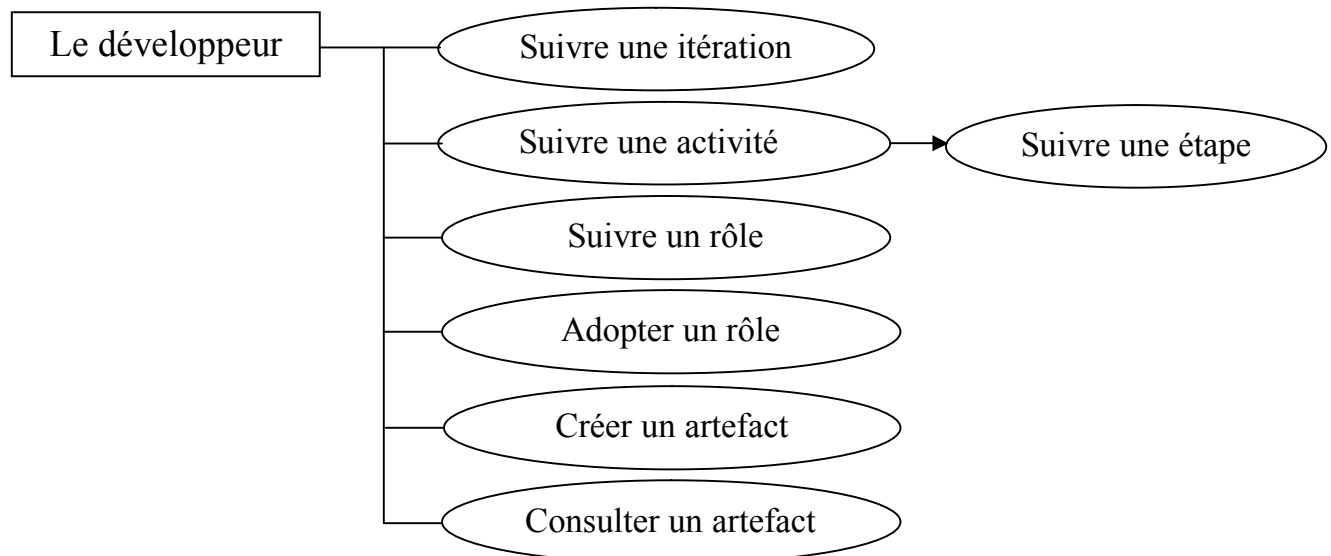


Figure 02 : Diagramme de cas d'utilisation.

II.5.3. Le diagramme de séquence.

Les Cas d'utilisation sont structurés dans un diagramme de séquence comme il est montré dans la figure 03, pour aider la détermination des communications demandées dans le système multi agents. Le diagramme de séquence définit la communication minimale qui doit avoir lieu entre les rôles du système.

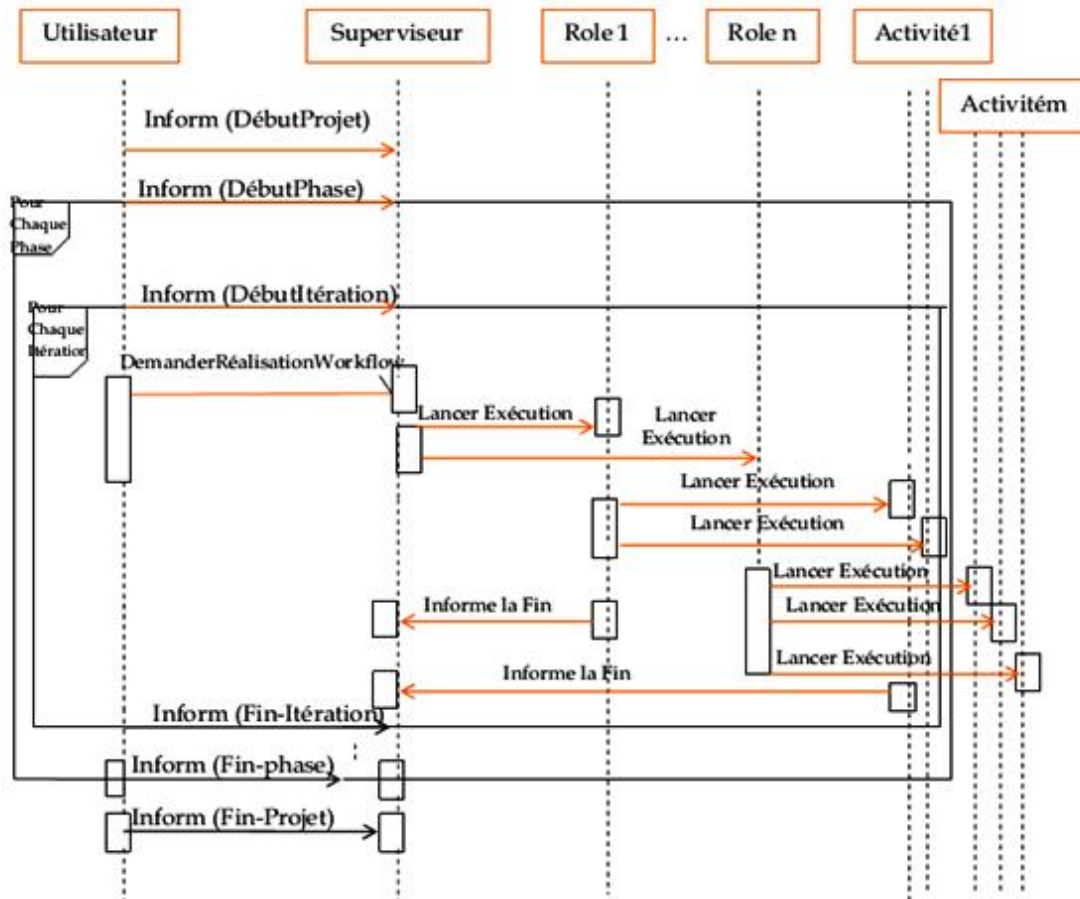


Figure 03 : Le diagramme de séquence.

II.5.4. Le diagramme de classe.

Le diagramme de classes montre les classes d'agents et les relations d'interaction entre elles. Pour notre cas, comme il est montré dans la figure 29, un Projet suit généralement un processus. Le processus est composé des activités et des rôles réalisant ces activités. Les rôles et les activités peuvent appartenir à plusieurs processus à la fois.

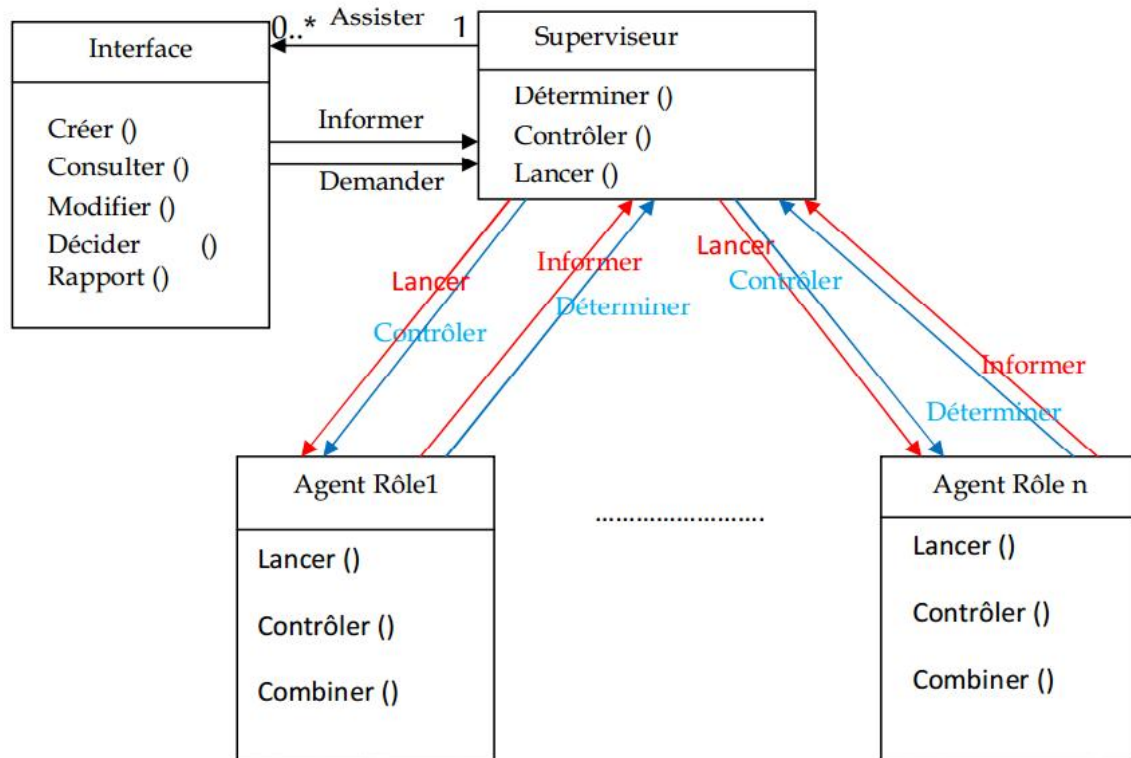


Figure 04 : Diagramme de classes.

Conclusion :

Au cours de ce chapitre nous avons vu l'outil de modélisation UML et est un processus de développement RUP (Rational Unified Process) que nous avons utilisé pour la conception de notre application.

Bibliographie

Bibliographie

[1] : www.comment-ça-marche.net

[2] : Développements n-tiers avec JAVA EE. Préface de Roberto Chinnici Java EE Platform Lead-Oracle Corporation. Collection Jérôme LAFOSSE.2013

[3] : « internet/intranet et les bases de données ».G. Gardarin Edition Eyrolles 2000.

[4] : L'architecture Client/ Serveur. Francois-Marie Colonna Edition 2000.

[5] : http://caron.nicolas.free.fr/cnam/Projet%20Methodologie/PASSAGE_DU_CLIENT_LOURD_AU_CLIENT_LEGER.ppt.

[6] : Pratique de MYSQL et PHP, conception et réalisation de site web dynamique Auteur : Rigaux, Philippe 4ème édition.2009

[7] : Developper une application avec PHP et MySQL , Auteur : etienne vandeput.2005

[8] : UML2 : pour l'analyse d'un système d'information. Auteur : Chantal morlay, Jean hugues, Bernard leblanc.4eme édition.

[9] : UML principes de modélisation. Auteur : Rémy fannader, Hervé Leraux 1er édition

www.google.fr

www.youtube.com

www.sitedezero.com

www.wikipedia.fr

www.developpeur.com