

# *REMERCIEMENTS*

*Nous tenons avant tout à remercier le Très haut qui nous a donné la volonté et le courage pour la réalisation de ce travail.*

*Nous remercions vivement madame SamiaBELATTAF notre promotrice pour l'assistance et la disponibilité qu'elle nous a accordée tout au long de la rédaction de ce mémoire.*

*Que les membres de jury trouvent d'ici l'expression de nos sincères remerciements pour l'honneur qu'ils nous font en acceptant de juger notre humble travail.*

*Nous tenons aussi à remercier Mr KABI et Mr REMDANOU qui nous ont généreusement aidés durant notre stage.*

*Nous aimerons profiter de l'occasion pour exprimer nos gratitude envers tous les membres de nos familles et tous nos amis qui nous ont soutenus et encourager.*

*DU PLUS PROFOND DE NOS  
CŒURS, MERCI*

## ***DEDICACES***

*Je dédie ce travail,*

*A MES CHERS PARENTS*

*Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consenti pour mon instruction et mon bien être.*

*Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours.*

*Que ce modeste travail soit l'exaucement de vos vœux tant formulés, le fruit de vos innombrables sacrifices, bien que je ne vous en acquitterai jamais assez. Puisse Dieu, le Très Haut, vous accorder santé, bonheur et longévité et faire en sorte que jamais je ne vous déçoive.*

*A ma très chère sœur SARAH, son mari HAKIM et leurs filles :  
(DJINANE, RITADJE, INASSE).*

*En témoignage de l'attachement, de l'amour et de l'affection que je porte pour vous. Malgré la distance, vous êtes toujours dans mon cœur.*

*Je vous remercie pour votre hospitalité sans égal et votre affection si sincère.*

*Je vous dédie ce travail avec tous mes vœux de bonheur, de santé et de réussite*

*A ma très chère sœur SABRINA*

*En témoignage de mon affection fraternelle, de ma profonde tendresse et reconnaissance, je vous souhaite une vie pleine de bonheur et de succès et que Dieu, le tout puissant, vous protège et vous garde.*

*Je dédie aussi ce travail ...*

*Aux personnes qui m'ont encouragé, qui étaient toujours à mes côtés, et qui m'ont apporté leur support moral et intellectuel tout au long de ma démarche, mes aimables amis, collègues d'étude.*

***OULKADI SALIM***

## *DEDICACES*

### *A MES CHERS PARENTS*

*Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consenti pour mon instruction et mon bien être.*

*Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours.*

*Que ce modeste travail soit l'exaucement de vos vœux tant formulés, le fruit de vos innombrables sacrifices, bien que je ne vous en acquitterai jamais assez.*

*Puisse Dieu, le Très Haut, vous accorder santé, bonheur et longue vie et faire en sorte que jamais je ne vous déçoive.*

*Je dédie aussi ce travail ...*

*Aux personnes qui m'ont encouragé, qui étaient toujours à mes côtés, et qui m'ont apporté leur support moral et intellectuel tout au long de ma démarche, mes aimables amis, collègues d'étude.*

*Achour BERRAHMA*

## Listes des références :

---

### Bibliographie :

(Auteur, Titre, Edition, Date de publication)

[4] : (Thierry GROUSSARD, le développement d'application web avec JEE6, ENI éditions, mars 2011).

[14] : (GRADY BOOCH, Livre Entreprise Edition, EYROLLES, OCTOBRE 2002).

[15] : (JAMES RUMBAUGH, Programmer facilement avec JAVA EE, Éditions Lefebvre Sarrut, juin 2013).

[16] : (IVAR JACOBSON, développement d'application en JAVA EE, édition Edilivre, janvier 2016).

[11] : Document interne à l'ENIEM.

[13] : Document interne au centre de médecine de travail.

---

### Webographie :

[1] : <https://fr.yeeply.com/blog/6-types-developpement-d-applications-web/>. Mars 2018.

[3] : <https://www.supinfo.com/articles/single/574-architecture-2-tiers-vs-juin> 2018.

[2] : <http://cours-fad-public.ensg.eu/mod/imscp/view.php?id=457>. Mars 2018.

[5] : <https://www.supinfo.com/articles/single/6628-architecture-applications-java-ee>. septembre 2018.

[6] : <https://openclassrooms.com/courses/adoptez-une-architecture-mvc-en-php/comment-fonctionne-une-architecture-mvc>. Mars 2018.

[7] : <https://helpx.adobe.com/fr/dreamweaver/using/web-applications.html> juin 2018.

[8] : [http://gardeuxvincent.eu/Documents/ProjetJEE/BVC\\_Hibernate\\_Struts2/Hibernate/index.htm](http://gardeuxvincent.eu/Documents/ProjetJEE/BVC_Hibernate_Struts2/Hibernate/index.htm). Mars 2018.

[9] : [https://www.webexpert.ch/publications/application\\_web](https://www.webexpert.ch/publications/application_web). avril 2018.

[10] : <https://www.supinfo.com/articles/single/6402-primefaces-java-introduction>. avril 2018.

[12] : [www.eniem.com.dz](http://www.eniem.com.dz). Mars 2018.

[17] : <https://www.oracle.com/technetwork/java/javaee/>. aout 2018.

[18] : <https://www.jmdoudoux.fr/java/dej/chap-hibernate.htm>. aout 2018.

[19] : <https://www.oracle.com/technetwork/.../javaserverfaces>, aout 2018.

[20] : <https://getbootstrap.com/docs>, aout 2018.

[21] : [www.oracle.com/technetwork/middleware/glassfish](http://www.oracle.com/technetwork/middleware/glassfish), aout 2018.

[22] : <https://fr.netbeans.org>, juin 2018.

[23] : <https://www.oracle.com/fr/mysql>, aout 2018.

[24] : [nhibernate.info/doc/nhibernate-reference/queryhql.html](http://nhibernate.info/doc/nhibernate-reference/queryhql.html), aout 2018.

[25] : [www.la-grange.net/w3c/xhtml](http://www.la-grange.net/w3c/xhtml), aout 2018.

[26] : <https://www.w3.org/XML>, aout 2018.

## Table des matières :

### Premier chapitre :

Introduction générale .....	1
I. Introduction .....	3
II. Présentation des applications web .....	3
II.1 Définition d'une application web dynamique : .....	3
II.2 Fonctionnement d'une application web: .....	3
II.3 Architectures web .....	5
II.3.1 Architecture à deux niveaux (client/serveur) : .....	5
II.3.2 Architecture à trois niveaux : .....	6
II.4 Avantages des applications web : .....	7
III. Présentation de Java EE : .....	7
III.1 Définition de Java EE: .....	7
III.2 Architecture java EE : .....	8
III.2.1 Modèle MVC : .....	9
III.2.2 Architecture MVC : .....	9
III.3 Les composants JEE : .....	10
III.3.1 Les composants clients ou tiers clients : .....	11
III.3.2 Les composants Web ou tiers web : .....	11
III.3.3 Les composants métiers : .....	12
III.4 Les conteneurs JAVA EE : .....	13
III.5 Les API JEE : .....	15
III.5.1 Java servlet : .....	15
III.5.2 Java server pages (JSP) : .....	15
III.5.3 Java server pages standard tag Library (JSTL) : .....	15
III.5.4 Java server faces (JSF) : .....	15
III.5.5 Java Data Base Connectivity (JDBC) : .....	15
III.5.6 Java Persistence API (JPA) : .....	16
III.5.7 Enterprise JavaBeans (EJB) : .....	16
III.6 Les Framework JEE : .....	16
III.6.1 Hibernate : .....	17
III.6.2 Java Server Faces (JSF) : .....	18
IV. Conclusion : .....	19

## Deuxième chapitre

I.	Introduction :	20
II.	Présentation générale de l'entreprise ENIEM	20
II.1	Situation géographique :	20
II.2	Historique :	20
II.3	Les missions principales de l'ENIEM :	21
II.4	Les objectifs de l'ENIEM :	22
II.5	Organisation générale :	22
III.	Etude des postes de travail	25
III.1	Définition d'un poste de travail :	25
III.2	Les postes de travail étudiés :	25
III.3	Fiche d'analyse des postes	26
III.3.1	Poste de travail N°01 : Médecin	26
III.3.2	Poste de travail N°02 : infirmier(e)	26
IV.	Etude des documents	27
IV.1	Liste des documents utilisés :	27
IV.2	Fiches d'analyse des documents :	28
V.	Problématique	31
V.1	Présentation du travail demandé :	32
V.2	Spécification des besoins :	32
V.3	Les besoins fonctionnels :	33
V.4	Les besoins non-fonctionnels :	33
VI.	Conclusion	33

## Troisième chapitre

I.	Introduction :	34
II.	Rappel sur UML :	34
III.	Démarche de modélisation	35
III.1	Phase d'analyse :	35
III.1.1	Diagramme de contexte :	36
III.1.2	Identification des acteurs :	36
III.1.3	Identification des cas d'utilisation :	38
III.1.4	Identification des scénarios :	42
III.1.5	Diagramme de cas d'utilisation :	47
III.2	Conception	53
III.2.1	Les diagrammes de séquences	53

III.2.2	Diagrammes de classe .....	55
III.2.3	Modèle relationnel : .....	60
IV.	Conclusion .....	62

## Quatrième chapitre

I.	Introduction : .....	63
II.	Environnement de développement : .....	63
II.1	La plateforme Java Entreprise Edition : .....	63
II.2	Framework : .....	64
II.2.1	Hibernate : .....	64
II.2.2	JSF (Java Server Faces) : .....	64
II.2.3	Bootstrap : .....	65
III.	Outils de développement: .....	66
III.1	Serveur d'application (glassfish) : .....	66
III.2	Netbeans : .....	66
III.3	MySQL: .....	67
III.3.1	MySQL est un système de gestion de base de données : .....	67
III.3.2	MySQL est un serveur de base de données relationnelle : .....	67
III.3.3	Bref historique : .....	67
IV.	Langage de programmation : .....	68
IV.1	Langage coté serveur : .....	68
IV.2	Langage coté client : .....	68
V.	Modèle physique des données : .....	68
VI.	Présentation de quelques interfaces de notre application : .....	76
VII.	Conclusion : .....	79



## Liste des figures :

Figure I. 1 : Fonctionnement d'une application web .....	5
Figure I-2 : architecture 2 tiers.....	6
Figure I-3: architecture à 3 tiers.....	6
Figure I.4: Architecture Java EE.....	8
Figure I-5: architecture MVC. ....	9
Figure I-6 : composant JEE .....	12
Figure I-7: les différents composants de JEE.....	13
Figure I-8 : conteneur.....	14
Figure II.9 Organigramme général de l'ENIEM.....	23
Figure III-10 : démarche de modélisation.....	35
Figure III.11 : diagramme de contexte. ....	36
Figure III.12: diagramme de cas d'utilisation général. ....	48
Figure III.13: diagramme de cas d'utilisation de l'infirmier. ....	49
Figure III.14: diagramme de cas d'utilisation du médecin spécialiste .....	50
Figure III.15: diagramme de cas d'utilisation du médecin généraliste .....	51
Figure III.16: diagramme de cas d'utilisation de l'administrateur. ....	52
Figure III-17 : Diagramme de séquences du cas d'utilisation "s'authentifier" .....	53
Figure III-18 diagramme de séquences du cas d'utilisation "ajouter un dossier médical" .....	54
Figure III-19 : diagramme de séquences du cas d'utilisation "consulter un dossier médical" ..	55
Figure III.20 : diagramme de classe. ....	57
Figure IV.21 architecture d'Hibernate.....	64
Figure IV.22 : interface de l'IDE Netbeans .....	67
Figure IV.23 Interface d'authentification. ....	76
Figure IV.24 Interface liste des patients. ....	77
Figure IV.25 Interface liste des consultations. ....	78
Figure IV. 26 détail de la visite.....	79

## Liste des tableaux :

Tableau II.1 : Historique de l'ENIEM .....	21
Tableau II.2: Poste du médecin.....	26
Tableau II-3: Poste d'infirmier .....	27
Tableau II.4 dossier médical. ....	29
Tableau II-5: Fiche de renseignement .....	29
Tableau II-6 : Fiche pour les visites périodiques, redéploiement et d'embauche .....	29
Tableau II-7 : Fiche pour la visite spontanée .....	30
Tableau II-8 : Fiche pour les visites d'accident de travaux.....	30
Tableau II-9: Fiche de demande d'examens .....	31
Tableau II-10 : Fiche d'ordonnance. ....	31
Tableau III.11 : structuration de cas d'utilisation globale.....	39
Tableau-III.12 : structuration des cas d'utilisations en package.....	41
Tableau III.13 : récapitulatif des scenarios par taches de l'infirmier. ....	43
Tableau III.14 : récapitulatif des scenarios par taches du médecin spécialiste. ....	45
Tableau III.15 récapitulatif des scénarios par taches d'un médecin généraliste. ....	46
Tableau III.16 : récapitulatif des scénarios par taches d'un administrateur .....	47
Tableau 17: dictionnaire de classes et des attributs .....	60
Tableau IV-18 : Table Médecin_spécialiste .....	68
Tableau IV-19 : Table Médecin_généraliste .....	69
Tableau IV-20 : Table Médecin_généraliste .....	69
Tableau IV-21 : Table Patient.....	70
Tableau IV-22 : Table Antécédents.....	70
Tableau IV-23 : Table Traitement.....	70
Tableau IV-24 : Table Examen .....	70
Tableau IV-25 : Table Visite_générale.....	71
Tableau IV-26 : Table Consultation .....	71
Tableau IV-27 : Table Opht.....	71
Tableau IV-28 : Table Orl .....	72
Tableau IV-29 : Table Locomoteur.....	72
Tableau IV-30 : Table Locomoteur.....	72
Tableau IV-31 : Table Locomoteur.....	73
Tableau IV-32 : Table Cardio_vasc .....	73
Tableau IV-33 : Table Digestif .....	73
Tableau IV-34 : Table Génito-urinaire .....	74
Tableau IV-35 : Table neuro-psychisme .....	74
Tableau IV-36 : Table Hémato .....	75
Tableau IV-37 : Table Endocrino .....	75
Tableau IV-38 : Table Exp_fonc .....	75

## Liste des abréviations :

**AJAX** :Asynchronous JavaScript And XML.

**CSS** :Cascading Style Sheet.

**EJB** :Enterprise JavaBeans.

**EL** : Unified Expression Language

**JDBC** : Java DataBase Connectivity.

**JEE** :Java Entreprise Edition.

**SGBD** :Système de Gestion de Base de Données.

**JMS** :Java Message Service.

**JNDI** :Java Naming and Directory Interface.

**JPA** :Java Persistence Api.

**JPQL** :Java Persistence Query Language.

**JSF** :Java Server Faces

**JSP** :Java Server Pages.

**JSTL** :Java Server Pages Standard Tag Library.

**HQL** :Hibernate Query Language.

**HTTP** :HyperText Transfer Protocol.

**HTTPS** : HyperText Transfer Protocol Secure.

**HTML** : Hyper Text Markup Language.

**PHP** : Hypertext Preprocessor.

**SQL** :Structured Query Language.

**URL** : Uniform Resource Locator.

**XML** : eXtensible Markup Language.

**XHTML** : eXtensible HyperText Markup Language.

# Introduction générale

**Premier chapitre :  
Introduction à la technologie  
Java EE**

**Deuxième chapitre :**  
**Organisme d'accueil et étude**  
**de l'existant**

# **Troisième chapitre : Analyse et conception**

# **Quatrième chapitre : Réalisation**



# **Bibliographie**

## Introduction générale

Actuellement, le monde connaît une avance technologique considérable dans tous les domaines, et cela grâce à l'informatique qui s'est imposée d'une manière très impressionnante dans les entreprises, cela est dû à son apport extraordinaire dans le traitement automatique de l'information. En effet, l'informatique désigne l'automatisation du traitement de l'information par un système concret « machine » ou abstrait. L'informatique est de plus en plus utilisée dans tous les domaines d'activités y compris celui de la médecine de travail auquel nous rattacherons d'ailleurs notre étude.

Dans le cadre de notre projet de fin d'étude nous avons intégré le centre de médecine de travail de l'entreprise nationale « ENIEM » sise à OUED AISSI.

A l'heure actuelle, l'ensemble des traitements effectués au sein de ce centre se fait manuellement où le personnel de ce dernier utilisent des supports en papier pour la gestion de différentes procédures, visites et consultations. Cette situation engendre un certain nombre de problèmes tels que : la perte de temps considérable dans la recherche des documents et la dégradation de ces derniers, la difficulté de stockage, ainsi que le manque de confidentialité, ...etc.

Notre objectif consiste donc, à la conception et la réalisation d'un système d'information pour le Centre de Médecine du Travail de l'entreprise « ENIEM » qui s'occupe des suivis médicaux de ses employés.

Pour la réalisation de cette tâche, notre choix s'est porté sur une application web dynamique, que nous présentons dans ce document qui se subdivise en quatre principaux chapitres.

Dans le premier chapitre intitulé « Introduction à la technologie Java EE » nous présentons quelques concepts de base sur les applications web et nous donnons un aperçu sur la technologie Java EE.

Puis, au deuxième chapitre « Organisme d'accueil et étude de l'existant », nous allons présenter l'organisme d'accueil, approfondirons la compréhension du contexte du système par un processus continu de collecte d'information auprès des experts du domaine, ensuite nous allons déterminer les inconvénients majeurs de la gestion actuelle du centre de médecine de travail, énumérons des suggestions informatiques qui peuvent remédier aux difficultés rencontrées, et enfin, tenant compte des moyens de l'entreprise, nous proposons la solution qui paraît la plus adéquate.

Au niveau du troisième chapitre intitulé « Analyse et conception », nous allons analyser les principaux objectifs attendus du futur système à concevoir qui seront décrits par le diagramme des cas d'utilisation et nous allons présenter les diagrammes effectués au niveau de l'analyse en y intégrant les aspects techniques les plus proches des préoccupations physique.

Finale­ment dans le dernier chapitre qu'on a nommé « Réalisation » nous allons présenter les outils de développement qui nous ont servi pour le développement de l'application, et enfin l'activité test qui consiste, justement, à la tester dans le but de s'assurer de son bon fonctionnement.

## I. Introduction

Le standard JEE fait partie des technologies dites révolutionnaires de notre époque précisément en développement d'applications d'entreprises de grande envergure. Elle présente plusieurs solutions et elle est en perpétuelle évolution.

L'objectif de ce chapitre est de pouvoir comprendre le fonctionnement des applications web en général et de comprendre le fonctionnement des applications conçue avec Java EE. Par conséquent, nous allons présenter en premier lieu l'application web, son fonctionnement et son architecture, ensuite nous allons présenter la technologie Java EE, son architecture, ainsi que les différents composants utilisés dans cette dernière.

## II. Présentation des applications web

### II.1 Définition d'une application web dynamique :

Une application web est une application manipulable directement en ligne grâce à un navigateur web et qui ne nécessite donc pas d'être installée. De la même manière que les sites web, une application web est généralement placée sur un serveur et se manipule en actionnant des widgets à l'aide d'un navigateur web, via un réseau informatique (Internet, intranet, réseau local, etc.)

Une application web dynamique est un ensemble de pages statiques et dynamiques dont le contenu de ces pages est partiellement ou entièrement indéterminé. Le contenu final d'une page est déterminé uniquement lorsque l'utilisateur requiert une page depuis le serveur web, et ce dernier varie d'une requête à une autre en fonction des actions de l'utilisateur.

Une page statique est une page qui n'est pas modifiée lorsqu'un visiteur la consulte. Le serveur web la transmet au navigateur qui la sollicite sans la modifier, par contre une page dynamique elle est modifiée par le serveur avant d'être transmise au navigateur qui la sollicite.

L'application web dynamique est plus complexe sur le plan technique, car elle utilise des bases de données pour charger des informations, et le contenu est mis à jour à chaque fois que l'utilisateur se connecte à l'application. En général, elle a un panneau d'administration depuis lequel l'administrateur peut corriger ou modifier le contenu [1].

Il existe de nombreux langages de programmation pour le développement d'applications web, parmi ces langages, Java EE celui que nous allons utiliser pour programmer notre application et que nous allons présenter dans la suite de ce chapitre.

### II.2 Fonctionnement d'une application web:

Le web ce sont des machines communiquant en réseau à l'aide d'un langage commun. Parmi ces machines, on distingue celles qui proposent des ressources, les serveurs, et celles

qui les utilisent, les clients. Les ressources peuvent être par exemple des documents hypertexte, des images, des fichiers XML ou encore des programmes (PHP, Java,) chargés de les générer à la demande [2].

Un serveur est un processus accomplissant une opération sur demande d'un client et qui transmet la réponse à ce dernier qui est à son tour un processus demandant l'exécution d'une opération à un processus serveur par l'envoi d'un message contenant un descriptif de l'opération à exécuter et attendant la réponse à cette opération par un message en retour [2].

Quand un client accède à une ressource (consulter un document, modifier des données stockées sur le serveur...etc.). Il doit utiliser le protocole de communication HTTP pour l'atteindre : ce dernier définit une sémantique très simple (GET, POST, et autres commandes) permettant de formuler des requêtes qui sont interprétées côté serveur par un programme spécifique qui est : le serveur web [2].

Un serveur web est un simple serveur de fichiers. Les clients s'adressent à lui via le protocole HTTP afin d'obtenir une ressource. Lorsque le serveur web reçoit une telle requête HTTP, il extrait simplement de cette dernière, le nom de la ressource demandée ensuite va chercher sur le disque la ressource et l'emballe dans une réponse HTTP, afin de la transmettre au client et cela sans effectué aucun traitement sur cette dernière avant de la transmettre au client. Il peut donc indifféremment transférer vers un client une page HTML, une image, un son voir, même un fichier exécutable. Le type de contenu de la ressource demandée lui est totalement indifférent [4].

Lorsqu'un serveur web reçoit une requête de page web statique, il transmet directement cette page au navigateur qui la demande.

Par contre lorsque le serveur web reçoit une requête de page dynamique, il réagit différemment : Il transmet la page à un logiciel spécial chargé de terminer la page. Ce logiciel spécial est appelé un serveur d'application [2].

Un serveur d'application est radicalement différent du serveur web, car les ressources qui lui sont confiées ne sont pas de simples fichiers statiques, mais du code qu'il va se charger d'exécuter pour le compte des clients qui en font la demande. Lorsqu'un serveur d'application reçoit une requête HTTP, il analyse lui aussi cette requête pour déterminer quelle ressource lui est demandée. Généralement la requête concerne du code exécutable hébergé par le serveur. Contrairement à ce que ferait un serveur web dans la même situation il ne transfère pas vers le client le code mais il l'exécute et c'est le résultat de ce code qui sera renvoyé au client [4].

Le serveur d'application lit le code de la page, termine cette page en fonction des instructions figurant dans le code, puis en retire le code. Il en résulte une page statique que le serveur d'application renvoie au serveur web, qui à son tour la transmet au navigateur demandeur. Le navigateur reçoit uniquement du code HTML pur lorsque la page lui est transmise [2].

La figure I.1 représente un aperçu du processus [2] :

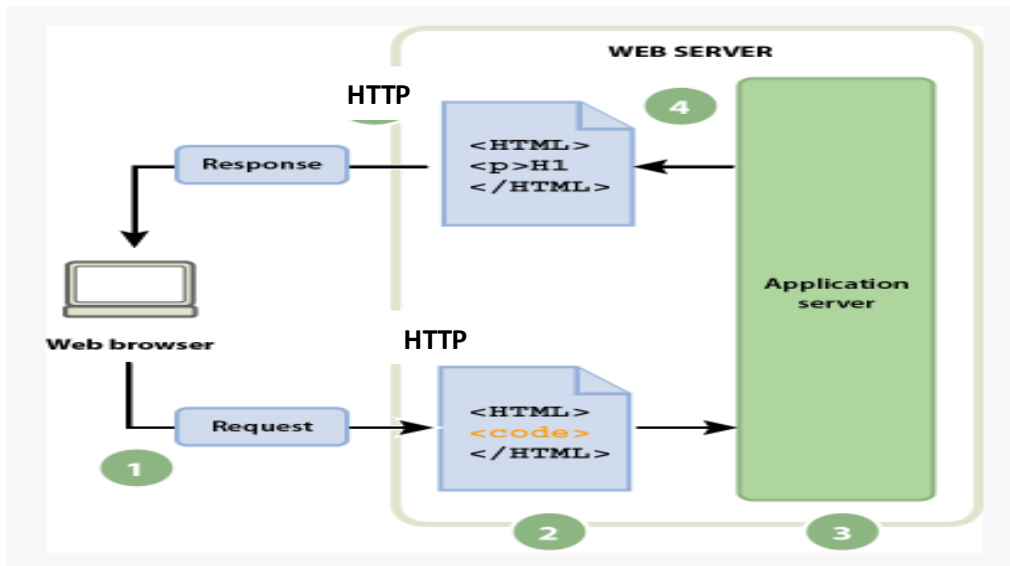


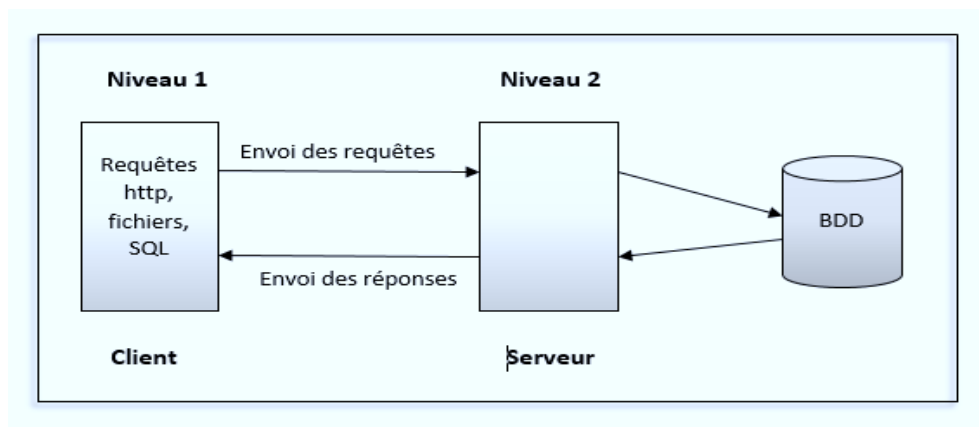
Figure I. 1 : Fonctionnement d'une application web

**Remarque :** sur le web, une ressource est identifiée par son url (chaîne de caractères). Lorsque le client souhaite atteindre une ressource distante, il émet donc une requête HTTP mentionnant l'url de la ressource.

## II.3 Architectures web

### II.3.1 Architecture à deux niveaux (client/serveur) :

Aussi appelé l'architecture « client-serveur » est assez simple. Vous avez d'un côté le client et de l'autre le serveur (figure I.2). Ce genre d'architecture peut se faire sur tout type d'architecture matérielle interconnectée [3].



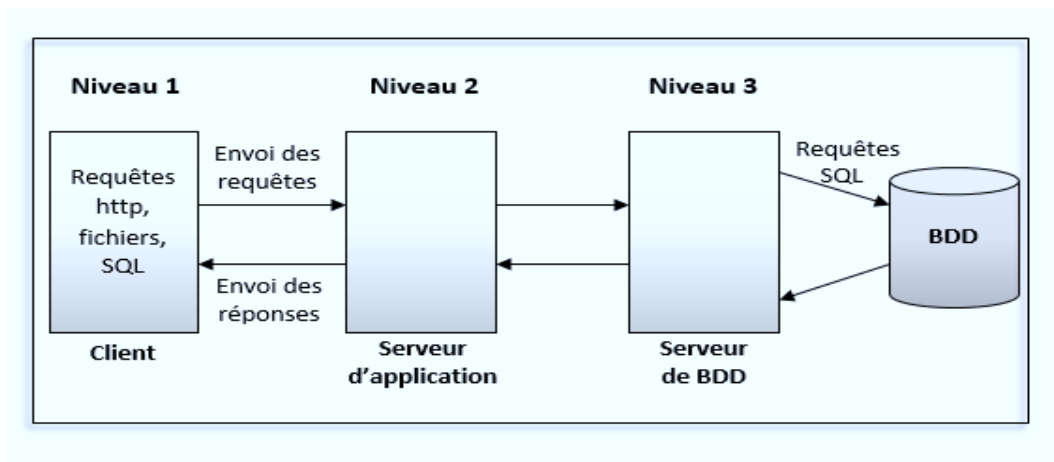
*Figure I-2 : architecture 2 tiers.*

Les points importants d'une application deux-tiers :

- Le client demande un service au serveur comme par exemple la page contact.html.
- Le serveur reçoit cette requête HTTP, effectue un traitement, et renvoie la ressource demandée par le client.
- Le client reçoit la ressource demandée.

### II.3.2 Architecture à trois niveaux :

Dans l'architecture trois-tiers (figure I.3), un nouveau niveau fait son apparition. En effet, nous avons toujours le premier niveau qui est le client, ce dernier est très léger étant donné qu'il n'a aucun rôle de traitement, il utilise l'application mais il dispose juste d'une infime partie de l'application sur son poste, tous les traitements sont alors réalisés côté serveur. Au deuxième niveau nous avons le serveur d'application et enfin, au dernier niveau le serveur de base de données [3].

*Figure I-3: architecture à 3 tiers*

Les points importants d'une application trois-tiers :

- Le client ne sert qu'à requêter et à afficher les réponses du serveur.
- Le serveur d'application lui s'occupe des calculs et même de requêter des serveurs additionnels.
- le serveur de BDD quant à lui s'occupe du stockage de données.

## II.4 Avantages des applications web:

Les avantages des applications web [9]:

- il n'est plus nécessaire d'installer un logiciel sur chaque poste, ce qui diminue en grande partie les frais de maintenance et élimine certaines incompatibilités.
- une application bien conçue peut être utilisée par différents types de terminaux (ordinateurs, laptops, tablettes, smartphones).
- il n'existe plus de contrainte géographique et on peut envisager un accès pour les utilisateurs nomades en toute sécurité grâce au cryptage des données à travers Internet.
- des centaines de personnes réparties dans des lieux différents peuvent travailler simultanément sur ce type d'applications, car elles sont conçues pour supporter une grande charge qui est souvent répartie sur plusieurs serveurs.
- ces applications sont capables de gérer des données multimédia (textes, photos, vidéos) et les utilisateurs sont à l'aise car habitués à utiliser un navigateur dans le cadre privé.
- pour mettre à jour l'application, il suffit de modifier l'application sur le serveur et tous les postes accèdent instantanément à la nouvelle version.

## III. Présentation de JavaEE

### III.1 Définition de Java EE :

La technologie JEE (Java Entreprise Edition) constitue la solution proposée par Sun pour le développement d'applications distribuées. La base de cette solution repose évidemment sur le langage java conçu également par Sun. Ce langage de base est également connu sous le terme JSE (Java standard Edition) [4].

En effet JEE n'est pas une amélioration de JSE. La version JSE est déjà suffisamment riche. JEE peut plutôt être considéré comme une norme décrivant tous les éléments constituant et intervenant pour le fonctionnement d'une application distribuée.

Elle définit par exemple les éléments suivants :

- Comment doivent être développés les différents composants d'une application (servlet, pages JSP...).
- Comment ces différents composants peuvent communiquer entre eux ou avec d'autres applications (JDBC, JNDI...).
- Comment doivent être organisés ces différents composants pour constituer une application (descripteur de déploiement) [4].



### III.2 Architecture javaEE:

L'architecture Java EE repose sur une architecture en couches qui, repose sur un découpage en couches ou tiers, nous parlons alors d'applications multi-tiers. Trois grands tiers sont ainsi représentés, dans lesquelles sont repartis différents composants que nous allons voir dans la suite de ce chapitre.

L'objectif de ce découpage est de proposer une meilleure répartition des rôles (chaque couche a un rôle clairement défini) et une meilleure maintenabilité, étant donné que chaque couche est indépendante des autres[5].

La figure I.4 montre le découpage en couche :

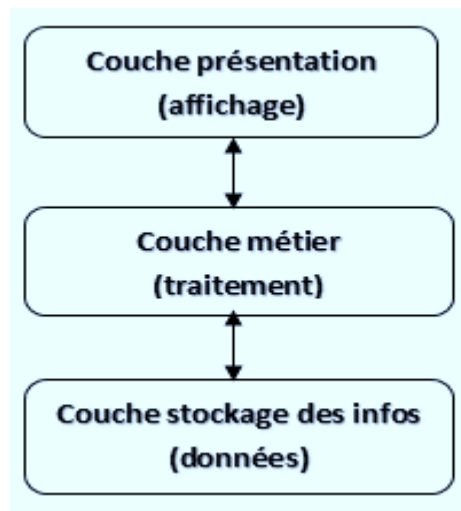


Figure I.4: Architecture Java EE

#### **La couche présentation (tiers Web):**

Cette couche gère l'affichage des données et les interactions de l'application avec l'utilisateur. La séparation de cette couche permet notamment de proposer plusieurs présentations pour une même application : la même couche de traitement peut alors servir pour une application lourde et pour une application légère.

#### **La couche métier (tiers Métier ou tiers Business) :**

Cette couche concerne à la fois les tâches à réaliser par l'application sur les données et les traitements nécessaires suite à une action venant de l'utilisateur : (vérification d'authentification, calculs divers,...).

#### **La couche stockage des informations (tiers Enterprise Information System) :**

Cette couche regroupe le stockage et les mécanismes d'accès des données de façon à ce qu'elles soient utilisables par l'application au niveau traitement.

### III.2.1 Modèle MVC :

JEE (Java Enterprise Edition) est une plateforme qui permet de faciliter le développement d'application d'entreprise, en utilisant des bonnes pratiques de développement qu'on appelle les Patrons de conception ou Design Pattern (en anglais).

MVC est un modèle de conception qui permet de décrire les grandes lignes d'une solution de développement en séparant la logique du code en trois parties que l'on retrouve dans des fichiers distincts qui sont [6] :

#### Modèle :

Cette partie gère les données de l'application. Son rôle est d'aller récupérer les informations « brutes » dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur.

#### Vue :

Cette partie se concentre sur l'affichage. Elle ne fait presque aucun calcul et se contente de récupérer des variables pour savoir ce qu'elle doit afficher. On y trouve essentiellement du code HTML mais aussi quelques boucles et conditions (Java) très simples, pour afficher par exemple une liste de messages.

#### Contrôleur :

Cette partie gère la logique du code qui prend des décisions. C'est en quelque sorte l'intermédiaire entre le modèle et la vue : le contrôleur va demander au modèle les données, les analyser, prendre des décisions et renvoyer le texte à afficher à la vue. Le contrôleur contient exclusivement du Java. C'est notamment lui qui détermine si le visiteur a le droit de voir la page ou non (gestion des droits d'accès).

### III.2.2 Architecture MVC :

L'architecture MVC est représentée par la figure I.5 :

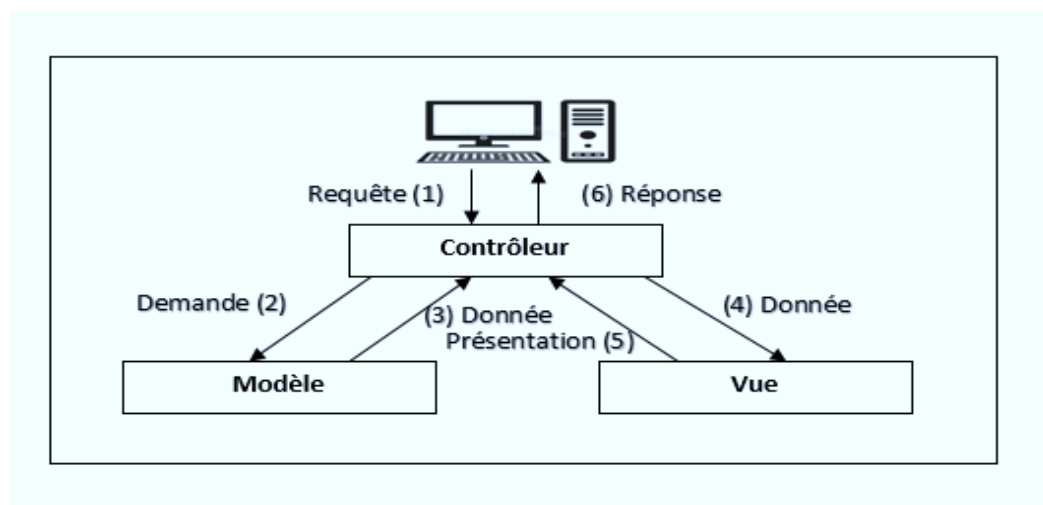


Figure I-5: architecture MVC.

Principe de l'architecture MVC [5] :

1. Le visiteur envoie sa requête HTTP, la transmet au serveur d'application qui la transmet directement à la partie de notre code qu'on appelle le Contrôleur.
2. Le contrôleur fait le routage de l'information en décidant qui va récupérer l'information et ensuite la traiter, il appelle en effet le Modèle qui contient les informations structurées.
3. le modèle va effectuer des calculs ou traitement sur ces informations et les renvoie au Contrôleur.
4. le contrôleur va demander à la vue de générer une vue (Page Web).
5. la vue génère une page web selon la demande du contrôleur et la lui transmet.
6. le contrôleur reçoit la page web qui sera envoyé au visiteur comme résultat de sa requête.

### III.3 Les composants JEE:

Les applications web 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 Java et de fichiers, et communiquant avec d'autres composants. De même, le code métier écrit est indépendant de la couche de présentation, ce qui est utile pour changer cette dernière ou pour l'afficher sur divers supports (Internet, Téléphonie...) [5].

Java EE définit les composants suivants :

- Les composants Clients ou tiers Client sont des applications clientes (logiciel installé en local ou navigateur web ou Applets) s'exécutant sur la machine des utilisateurs.
- Les composants Web ou tiers Web sont les technologies Servlets, JavaServer Pages et Java- Server Faces.
- Les composants métier ou tiers Métier sont des composants Entreprises JavaBeans (EJB) représentant la logique métier, s'exécutant sur le serveur Java EE et dialoguant avec la couche de stockage (EIS : *Enterprise Information System*).

Une application Java EE multi-tiers correspond à 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'applications Java EE.
- Le serveur de base de données ou de stockage.

### III.3.1 Les composants clients ou tiers clients :

La plate-forme Java EE propose trois types de clients : les clients Web, les Applets et les applications clientes riches type Java SE [5].

#### Clients web

Un client Web consiste en des pages Web de différents types (HTML, XHTML, XML, JavaScript ou autres) générées à partir de composants exécutés côté serveur dans un conteneur Web et capables de répondre aux requêtes HTTP, en provenance du navigateur Internet. Ces programmes côté serveur sont représentés en Java EE par les Servlets, pages JSP et JSF. Ces programmes serveur ne réalisent en général pas directement les accès aux bases de données mais s'occupent de la logique applicative.

#### Clients Applets

Les Applets sont des interfaces graphiques Java SE exécutées 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, car elle 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.

#### Applications Clientes

Une d'application cliente s'exécute sur un ordinateur client, et permet aux utilisateurs de gérer des tâches nécessitant une interface utilisateur plus riche que celle fournie par un langage de balisage. Elle possède généralement une interface utilisateur graphique (GUI), créée à partir de l'API AWT (Swing ou Abstract Window Toolkit), mais une interface de ligne de commande est certainement possible. Les Applications clientes accèdent directement aux beans d'entreprise s'exécutant dans le niveau métier. Toutefois, si les exigences de l'application le justifient, un client d'application peut ouvrir une connexion HTTP pour établir une communication avec une servlet s'exécutant dans la couche Web. Les applications clientes écrites dans des langages autres que Java peuvent interagir avec les serveurs Java EE, ce qui permet à la plate-forme Java EE d'interagir avec les systèmes et clients qui ne sont pas développés en Java.

### III.3.2 Les composants Web ou tiers web :

Les composants Web Java EE sont des Servlets et/ou des pages JavaServer Pages et/ou des JavaServer Faces. Les Servlets sont des classes Java, capables d'intercepter et de gérer les requêtes du protocole HTTP. Les pages JSP sont des documents textuels exécutés comme des Servlets apportant une solution simplifiée pour la programmation de pages Web. La technologie JavaServer Faces est construite à partir de Servlets et fournit un Framework de développement pour accélérer la création d'applications Web.

### III.3.3 Les composants métiers :

Le code métier, qui est la logique qui résout ou répond aux besoins d'un domaine d'activité particulier, tel que la banque, le commerce de détail ou la finance, il est géré par des beans entreprise s'exécutant dans côté métier. La Figure I.6 montre comment un entreprise bean reçoit des données de programmes client, les traite (si nécessaire) et les envoie au niveau du système d'information d'entreprise pour le stockage. Un entreprise bean récupère également les données du stockage, les traite (si nécessaire) et les renvoie au programme client.

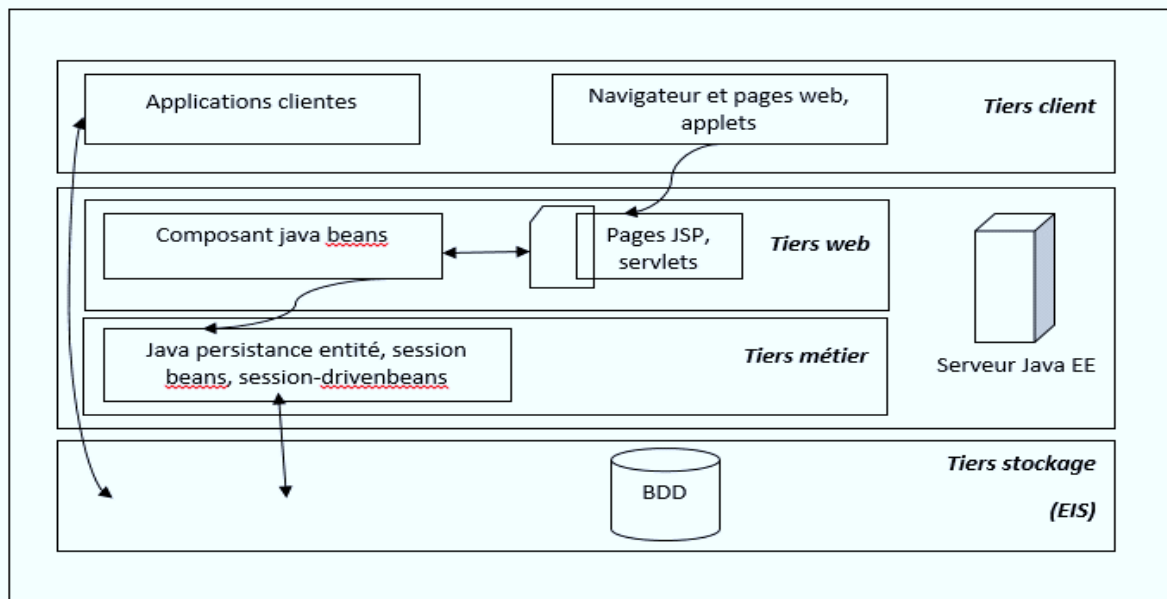


Figure I-6 : composant JEE

On constate dans cette architecture qu'on dispose de tous les composants (Client Tiers, Web tiers et Business Tiers) cités plus haut. Cependant, les applications Java EE utilisent un serveur d'application situé entre le serveur Web et la couche base de données. Le Serveur d'application va servir à exécuter les composants métier (EJB) et constitue ce qu'on appelle un environnement d'exécution des composants métier (EJB) qu'on appelle Conteneur d'application qui fournira tout ce qui est nécessaire pour l'exécution des JavaBeans. Le conteneur Web quant à lui servira d'environnement d'exécution des JSP et servlet, etc.

On obtient ainsi l'architecture présentée par la figure I.7 :

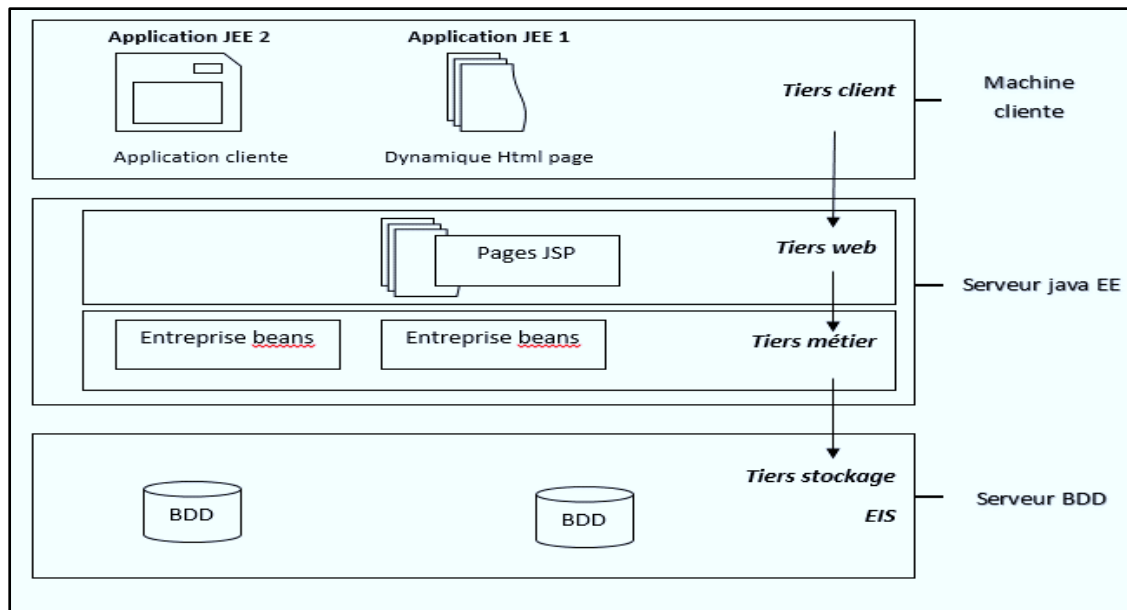


Figure I-7: les différents composants de JEE

### III.4 Les conteneurs JAVA EE :

Les serveurs Java EE proposent plusieurs types de conteneurs (containers) et pour chaque type de composant. Chaque conteneur a un rôle bien défini et offre un ensemble de services pour les développeurs [5]:

- 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 paramétrage des transactions.

Les exécutions et développements 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 nommées, 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 [5].

Les différents conteneurs sont montrés dans la figure I.8 :

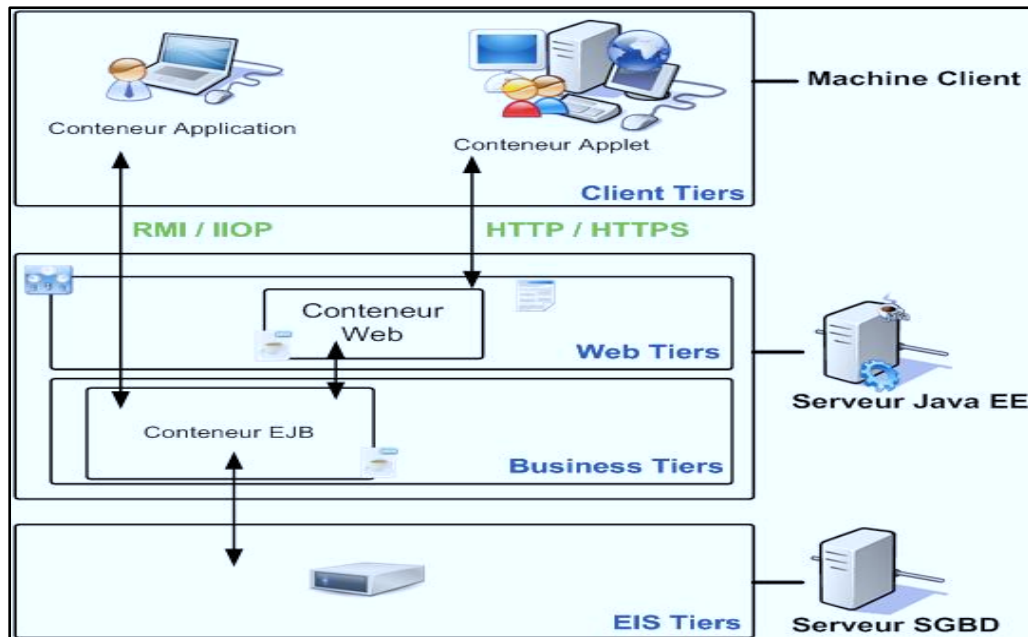


Figure I-8 : conteneur

Il existe en Java EE cinq types de conteneurs qui sont [15] :

- **Conteneur Web** : capable d'exécuter des pages JSP, Servlets et programmes 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).
- **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.
- **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 lancés dans un « bac à sable » aussi nommé « sandbox » afin de gérer la sécurité des accès sur le poste client.
- **Conteneur des applications clientes (ACC)** : 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'applications 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.

Java EE serveur est le conteneur qui fournit le conteneur web et EJB.

### III.5 Les API JEE :

La plate-forme Java EE est proposée par Oracle et regroupe les services de la plate-forme Java SE. Les services et API Java EE sont donc associés à une plate-forme et à un ou plusieurs conteneurs. Les conteneurs étudiés précédemment proposent plusieurs services pour le développement d'applications. On peut 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 [5].

On distingue plusieurs API [5][16] :

#### III.5.1 Java servlet :

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.

#### III.5.2 Java server pages (JSP) :

La technologie JavaServer Pages (JSP) s'exécute dans le conteneur Web et utilise le mécanisme de génération de Servlets à partir de documents textuels. Les JSP 2.2 contiennent des données statiques de type HTML, XHTML, JavaScript ou autres, et des balises ou tags JSP nommés JSTL correspondant à du code Java.

#### III.5.3 Java server pages standard tag Library (JSTL) :

La technologie JavaServer 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 itérations, l'internationalisation, les affichages HTML ou encore l'accès aux bases de données.

#### III.5.4 Java server faces (JSF) :

La technologie JavaServer Faces s'exécute dans le conteneur Web et correspond à la réponse du consortium Java EE en matière de Framework de développement côté serveur. JSF 2.0 est un Framework de développement de projets 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'affichages 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, évènements...).

#### III.5.5 JavaData Base Connectivity (JDBC) :

La technologie Java 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 JSP, page JSF ou autres. L'API JDBC propose un ensemble de classes et interfaces pour se connecter aux différents SGBD du marché mais également les paquetages de manipulation des données. JDBC est une technologie sous-jacente à JPA.

### III.5.6 Java Persistence API (JPA) :

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.

- L'API Java Persistence 2.0 propose les services suivants :
- 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.

### III.5.7 Enterprise JavaBeans (EJB) :

Les composants Enterprise JavaBeans s'exécutent dans le conteneur dédié EJB et correspondent à la couche ou tiers Métier de l'application. Ces EJB 3.X sont des classes composées de champs et de méthodes d'accès. Java EE propose deux familles d'EJB : les sessions beans et 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 (MDB) mélange une session bean et un écouteur de message permettant ainsi la réception asynchrone de messages. Ces services utilisent Java Message Service (JMS) pour leur fonctionnement. La nouvelle version de Java EE offre la possibilité d'empaqueter 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.

## III.6 Les Framework JEE :

Un Framework est un ensemble de composants permettant de réaliser ou de créer l'architecture et les grandes lignes d'une application. Il se présente sous diverse forme qui peut inclure tout une partie des éléments suivants [6]:

- Un ensemble de classes généralement regroupées sous la forme de bibliothèques pour proposer des services plus ou moins sophistiqués

- Un cadre de conception reposant sur les design patterns pour proposer tout ou partie d'un squelette d'application
- Des recommandations sur la mise en œuvre et des exemples d'utilisation
- Des normes de développement
- Des outils facilitant la mise en œuvre

En ce qui concerne Java EE, il existe de nombreux Framework parmi eux :

### III.6.1 Hibernate:

Dans les langages objet, les données étant le plus souvent stockées dans des bases de données relationnelles ainsi l'utilisation d'un Framework de mapping Objet/Relationnel est recommandé pour assurer la rapidité, l'évolutivité et la maintenabilité des développements. Hibernate, issu de la communauté Open Source, répond à ce besoin et connaît depuis quelques années un vif succès. Ce succès s'explique notamment par son architecture parfaitement adaptable à tout type de développements et le support de la majorité des bases de données du marché.

Hibernate est un Framework de persistance utilisé pour gérer la persistance des objets java dans une base de données et plus précisément c'est une surcouche de JDBC qui lui ajoute une dimension objet. Le terme mapping objet/relationnel (ORM) décrit la technique consistant à faire le lien entre la représentation objet des données et sa représentation relationnelle, basée sur un schéma SQL.

Concrètement, Hibernate permet de lier/mapper un objet défini en Java avec une table dans une base de données, via un fichier déclaratif de mapping. Le système peut s'occuper de la création des tables en fonction des fichiers de configuration et mettre aussi à jour les tables si nécessaire lors d'un changement dans l'un des fichiers de configuration.

Hibernate possède plusieurs moyens pour effectuer des requêtes. Il est possible d'exprimer des requêtes en SQL, ou en HQL (Hibernate Query Language : langage propre à Hibernate) ou encore en critères orientés objet. On notera que pour les utilisateurs d'Eclipse, il existe un plugin : Hibernate Synchroniser, qui permet de faciliter l'utilisation d'Hibernate [8].

Les avantages et les inconvénients de Hibernate [16]:

#### Avantages :

- Base de données indépendante : on n'a pas besoin d'écrire un code spécifique pour la base de données (si on ne change la bd, on ne sera pas obligé à modifier dans le code l'application "requêtes SQL").
- On peut mettre à jour la base de données sans écrire du code SQL ou HQL.
- Le Framework Hibernate est opensource.

#### Inconvénients :

- Il est un peu difficile de faire des requêtes complexes avec HQL.

### III.6.2 JavaServer Faces (JSF) :

JSF, alias Java Server Faces est un Framework de type MVC, destiné aux applications web respectant l'architecture J2EE. Son développement a tenu compte des différentes expériences acquises lors de l'utilisation des technologies standard pour le développement d'applications web (servlet, JSP, JSTL) et de différents Framework L'objectif de JSF est de fournir un environnement de développement pour la construction d'interfaces de type web sans avoir à toucher au JavaScript ou au HTML. Depuis la version 2 de JSF, les fonctionnalités ont été enrichies grâce à l'ajout de traitement des requêtes de type AJAX et l'utilisation d'autres moteurs de templating. JSF utilise la notion de Managed-bean et les expressions EL pour afficher les données sur les pages web [10].

En résumé JSF se compose de :

- Un ensemble d'APIs pour la représentation et la gestion des composants, de leur état, des événements, de la validation des entrées et la conversion des sorties, l'internationalisation et l'accessibilité ainsi que la navigation inter-vues.
- Deux jeux de composants standards (affichage de texte, saisie de texte, tables, zones à cocher, ...).
- Deux bibliothèques de balises JSP (une pour chaque jeu de composants) pour permettre l'utilisation des JSPs pour la construction de vues JSF.
- Un modèle événementiel côté serveur.
- Les Managed-Beans qui forment la couche contrôle de JSF.
- Unified Expression Language : abrégé en EL ou langage d'expressions unifié pour JSF et JSP 2.0. Il permet de lier les composants aux managed-beans.

Les avantages et les inconvénients de JSF [15] :

#### Avantages :

- Possibilité de créer de nouveaux composants.
- Accès aux Beans par leurs noms en utilisant les Expressions Language.
- Simplification des définitions des contrôleurs et des Beans.
- Simplification du fichier de configuration.
- L'orientation de composants graphiques permet à JSF d'être utilisé de manière plus simple dans les environnements de développement.

#### Inconvénients :

- Pas de moyen facile de commenter le fragment de la page incorrecte.
- Ne fonctionne pas bien avec REST.
- Pas si facile pour les concepteurs UX, car les composants prêts à utiliser ont leurs propres styles CSS, qui doivent être écrasés.

## IV. Conclusion :

Dans ce chapitre nous avons parlé sur les fondamentaux des applications web, et sur leur fonctionnement, de même, nous avons présenté la technologie Java EE, son architecture, ces composants et quelque conteneurs et APIs.

Quant au chapitre suivant intitulé « L'organisme d'accueil et étude de l'existant », nous allons étudier notre organisme d'accueil.

## I. Introduction :

Afin de se familiariser avec notre domaine d'application, nous présentons dans ce chapitre l'entreprise ENIEM qui est parmi les entreprises qui ont pu résister à tous les changements et toutes les conjonctures que le pays a connues depuis 1977. Elle continue à tout de même à exercer son activité en raison de la subvention et le soutien de l'état.

## II. Présentation générale de l'entreprise ENIEM

### II.1 Situation géographique :

Le siège social de l'ENIEM se situe au chef-lieu de la wilaya de Tizi-Ouzou. Les unités de production sont implantées dans la zone industrielle d'Oued-Aissi, distantes de 7 km du chef-lieu de wilaya, sur l'axe reliant la ville de Tizi-Ouzou et la commune de Tizi-Rached.

### II.2 Historique :

ENIEM est entrée en production à partir de janvier 1977 dont l'activité principale est la fabrication et la commercialisation d'appareils électroménagers. L'entreprise est certifiée ISO 9001/2008 QUALITE et ISO 14001/2004 ENVIRONNEMENT. [12]

Le tableau II.1 représente l'historique de l'entreprise depuis sa création. [11]

Année	Faits marquants
De 1977 à 1981	<ul style="list-style-type: none"> <li>-Démarrage de la fabrication des réfrigérateurs petit modèle en collaboration avec le partenaire allemand RIAG-BOCH/1977.</li> <li>-Démarrage de l'ancienne usine de cuisinière de réchauds à plat en collaboration avec le partenaire allemand RIAG-SPELL-FRENCKE en 1977.</li> <li>-Montage du premier climatiseur de type FRENCK en 1978.</li> <li>-Entrée en production des chauffes eau\bain en collaboration avec le partenaire allemand BOCH-CHAFFOTEAUX, le partenaire français MAURY et le partenaire espagnol en 1979.</li> </ul>
De 1982 à 1986.	<ul style="list-style-type: none"> <li>-Introduction de montage de petits appareils de ménagers MAC, SC en 1982.</li> <li>-Entrée en exploitation de nouvelle usine de réfrigérateur et congélateur grands modèles en collaboration avec le partenaire japonais MITSU-TOSHIBA en 1986.</li> <li>-Elargissement de la gamme de climatisation du SPLIT système en 1986.</li> </ul>
De 1987 à	<ul style="list-style-type: none"> <li>-Abandon de la production de PAM, cédée aux sections privées en 1987.</li> <li>-Montage des armoires frigorifiques (CAF) en 1988.</li> <li>-Arrêt de production de réchaud à plat à deux feux en 1989.</li> </ul>

1989.	-Passage de l'ENIEM à l'autonomie conformément à la loi 88-01 portant sur l'orientation des EPE en octobre 1989.
De 1990 à 1991.	-Arrêt de la production des anciennes cuisinières et vente des équipements en 1990. -Entrée en production de l'usine de cuisinières avec le partenaire italien INTERCOOP-TECHNOGAZ en 1991
De 1992 à 1993.	-Mise en place de l'usine des congélateurs horizontaux (BAHUT) avec le partenaire libanais LEMATIC en 1992. -Reprise des montages des petits appareils ménages de type MAC, SC, THV et mixeurs avec le partenaire espagnol TAUBEN-TAURIS en 1993.
De 1994 à 1997.	-Réalisation des réchauds à plat quatre feux en 1994. -Abandon de la production CAF en 1994. -Entrée en production des réfrigérateurs 5201 en 1995. -Montage de petits appareils ménagers BLENDER et ventilateurs en collaboration avec le partenaire italien ITALISMAMP en 1995. -Entrée en production des PAM (SC et MAC) avec le partenaire italien ITALISMAMP en 1996. -Reprise de la production des comptoirs frigorifiques CAF. -Abandon de la production des réchauds à plat à quatre feux en 1997.
De 1998 à 2018	-Compte tenu de l'importance de son activité, la direction de l'ENIEM a procédé la restriction du CAM en sept unités à savoir : unité siège, unité commerciale, trois unités de production (froid, cuisson et climatisation), deux unités de prestation (technique et service). -Certification ISO-9001 de l'ENIEM par l'organisme français AFAQ-ASSERT. -Suppression de l'unité prestation de service en 2002.

*Tableau II.1: Historique de l'ENIEM*

### II.3 Les missions principales de l'ENIEM :

L'ENIEM est le leader de l'électroménager en Algérie. Elle possède des capacités de production et une expérience dans la fabrication et le développement dans les différentes branches de l'électroménager, notamment : les appareils de collectivités, les petits appareils ménagers, les lampes d'éclairage, les produits sanitaires.

La mission principale de l'ENIEM est d'assurer le montage, le développement et la recherche dans le domaine des différentes branches de l'électroménager. Elle assure aussi la production des appareils de réfrigération, de cuisson et de climatisation avec une faible intégration nationale, des composants sont importés.

Aujourd'hui, avec la transition de l'Algérie vers l'économie de marché et la concurrence exacerbée dans le marché de l'électroménager en Algérie, l'ENIEM est dans l'obligation de reconsidérer ses objectifs et de les orienter dans un but concurrentiel, de tenir compte de son système d'information et de sa veille concurrentielle. [11]

## II.4 Les objectifs de l'ENIEM :

L'objectif principal de l'ENIEM est le maintien de sa position sur le marché national et la pénétration du marché international. A cet objectif principal s'ajoute d'autres objectifs [11] :

- ✓ L'équilibre de sa structure financière.
- ✓ La suppression de sa dépendance financière.
- ✓ La réalisation de la rentabilité financière.
- ✓ Accroître la satisfaction du client.
- ✓ Diversifier les produits.
- ✓ Améliorer la compétence du personnel.
- ✓ Améliorer le chiffre d'affaire et augmenter la valeur de la production.

## II.5 Organisation générale :

A partir de janvier 1998, l'entreprise s'est réorganisée en centre d'activités stratégiques qui s'articulent autour de la restructuration du complexe d'appareils ménagers créant plusieurs unités de production et de soutien.

La figure II.9 montre l'organisation générale de l'ENIEM qui est représentée par un organigramme [12]

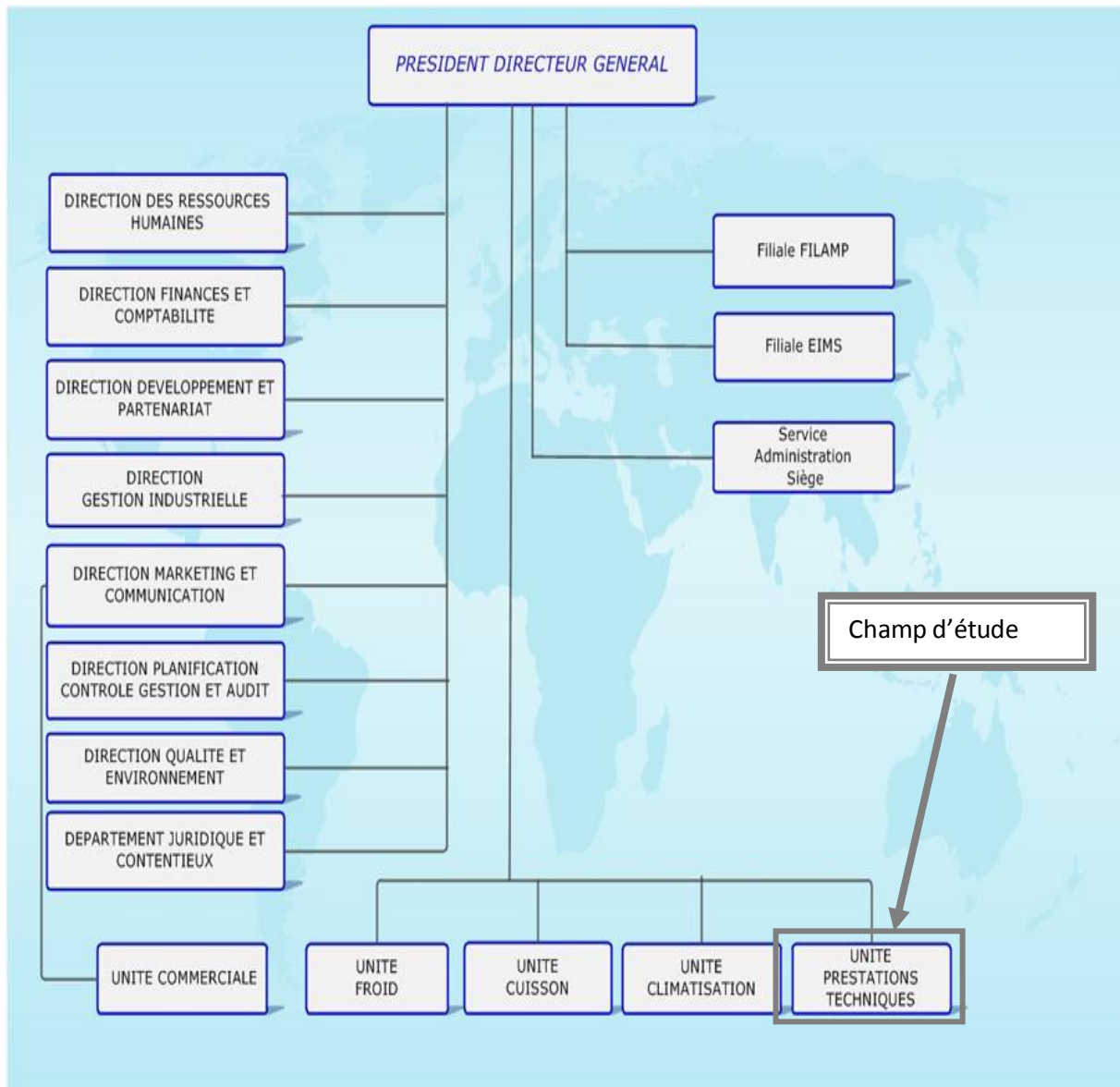


Figure II.9 Organigramme général de l'ENIEM

Dans ce qui suit nous détaillons quelques unités et directions en expliquons leurs tâches et responsabilité au sein de l'entreprise.

Les différentes directions sont :

- **Direction générale :**

Elle est responsable de la stratégie et veille au développement de l'entreprise. Elle exerce son autorité hiérarchique sur les autres directions.

- **Direction des ressources humaines :**

Elle est chargée de mettre en œuvre la gestion des emplois, des contrats, des carrières, des dossiers de pension, de la paie et de la formation de tous les personnels, elle pilote aussi les opérations de gestion collective tel que le recrutement et avancement.



- **Direction marketing et communication :**

Elle permet une analyse des ventes, donne des objectifs commerciaux et stratégiques, participe à la définition du plan marketing.

- **Direction développement et partenariat :**

Elle permet de coordonner toutes les actions visant à développer la collecte de fonds au profit d'actions de recherche et formation de l'entreprise.

- **Direction planification contrôle gestion et audit :**

Veille à améliorer les performances des structures de l'entreprise, le contrôle des achats, approvisionnement, délais de paiement etc.

- **Direction qualité et environnement :**

Elle a comme mission la mise en œuvre de stratégies afin d'atteindre des objectifs de qualité de service.

- **Direction finance et comptabilité :**

Elle a pour objectif d'élaborer les prévisions budgétaires et de tenir une bonne comptabilité.

- **Direction gestion industrielle :**

Elle est chargée de l'élaboration et de la mise en œuvre d'axes d'optimisation en vue d'augmenter la productivité ou d'améliorer les conditions de travail.

Les différentes unités sont :

- **Unité froid**

Elle est chargée de la production et de développement des produits froid.

Les activités de cette unité sont :

- ✓ Transformation de la tôle.
- ✓ Revêtement de surface (peinture, plastification).
- ✓ Fabrication des pièces métalliques.
- ✓ Assemblage.

- **Unité cuisson :**

Elle est chargée de la production et développement des produits de cuisson, ses activités sont :

- ✓ La transformation des tôles.
- ✓ Assemblage.

- **Unité climatisation :**

Cette unité est spécialisée dans la production des équipements de climatisation et de chauffage tels que chauffe bain, radiateur à gaz de butane, comptoirs et armoires frigorifique, équipement de climatisation pour les particuliers et industriels.

- **Unité commerciale :**

Elle assure la commercialisation des produits de l'entreprise, ses activités sont :

- ✓ La distribution et l'exportation des produits de l'entreprise
- ✓ Le service après-vente.
- ✓ Le marketing.
- ✓ La gestion des stocks des produits finis.

- **Unité prestations techniques :**

Chargée de fournir des prestations techniques et des services nécessaires aux unités de production.

Activités :

- ✓ La distribution et la commercialisation des produits électroménagers.
- ✓ Traitement de la vente de produits dans le cadre des marchés suite aux appels d'offres.
- ✓ Le stockage des produits finis.
- ✓ L'assurance de service après-vente pour les produits à travers son réseau d'agents agréés à travers le territoire national.

Notre stage pratique se déroule au sein de cette unité.

### III. Etude des postes de travail

#### III.1 Définition d'un poste de travail :

Un poste de travail est la position d'une ou plusieurs personnes ayant un certain nombre d'opérations à exécuter en utilisant un ensemble d'informations.

#### III.2 Les postes de travail étudiés :

Le poste de responsable : occupé par le médecin qui est responsable de la gestion du centre.

Le poste de communication et traitement : occupé par l'infirmière qui est chargée d'assurer un bon fonctionnement du centre.

On va créer une fiche d'analyse pour chaque poste afin d'étudier :

- Ses responsabilités.
- Ses tâches à accomplir.

- Les documents provenant à ce poste.
- Les documents diffusés par ce poste.

### III.3 Fiche d'analyse des postes

#### III.3.1 Poste de travail N°01 : Médecin

<b>Fiche d'étude du poste N°01</b>	
<b>Désignation</b> : Médecin	
<b>Effectif</b> : 01	
<b>Lieu d'implantation</b> : Le centre médical	
<b>Responsabilités</b> :	
<ul style="list-style-type: none"> <li>➤ Diriger le centre médical</li> <li>➤ Superviser et veiller au bon fonctionnement du centre médical</li> </ul>	
<b>Tâches à accomplir</b> :	
<ul style="list-style-type: none"> <li>➤ Rassurer et soigner les patients.</li> <li>➤ Le suivi et la mise à jour des dossiers des patients.</li> <li>➤ Délivrer les documents médicaux (l'ordonnances, l'arrêt de travail, etc.)</li> </ul>	
<b>Document provenant à ce poste</b>	
<b>Désignation</b>	<b>Provenance</b>
Fiche de renseignement sur les patients	Secrétaire
<b>Documents diffusés par ce poste</b>	
<b>Désignation</b>	<b>Destination</b>
<ul style="list-style-type: none"> <li>➤ La feuille de soins</li> <li>➤ L'ordonnance</li> <li>➤ Compte rendu d'échographie</li> </ul>	<ul style="list-style-type: none"> <li>➤ Le patient</li> <li>➤ Le patient</li> <li>➤ Le patient</li> </ul>
<b>Fichiers utilisés par ce poste</b>	
Dossier patient	

Tableau II.2: Poste du médecin

#### III.3.2 Poste de travail N°02 : infirmier(e)

<b>Fiche d'étude du poste N°02</b>	
<b>Désignation</b> : Infirmier(e)	
<b>Effectif</b> : 02	
<b>Lieu d'implantation</b> : le centre médical	
<b>Responsabilités</b> :	
Assurer le bon fonctionnement du centre médical.	
<b>Tâches à accomplir</b> :	
<ul style="list-style-type: none"> <li>➤ Accueillir les patients et organiser leur passage dans la salle de consultation.</li> <li>➤ Reconnaître une urgence et prévenir immédiatement le médecin.</li> <li>➤ Enregistrement manuel, codification, archivage et classement des dossiers médicaux.</li> </ul>	
<b>Document provenant à ce poste</b>	
Aucun document	
<b>Documents diffusés par ce poste</b>	
<b>Désignation</b>	<b>Destination</b>
Fiche de renseignements	Médecin

Fichiers utilisés par ce poste
Fiches des visites.

*Tableau II-3: Poste d'infirmier*

## IV. Etude des documents

Un document est une pièce de référence pour l'enregistrement d'une réalité. Cette étude permet de faire l'inventaire sur tous les documents utilisés dans le domaine d'études, elle a pour objectif de :

- Justifier la raison d'être de chaque document.
- Recenser l'ensemble des données utilisées.

On peut distinguer deux types de documents :

### ❖ Documents internes

C'est des documents produits par des acteurs internes au champ d'étude grâce à des traitements manuels ou non.

### ❖ Documents externes

Ils portent des informations diffusées ou reçues par les acteurs externes au champ d'étude.

### IV.1 Liste des documents utilisés :

La liste des documents internes et externes utilisés par les différents employés du centre médicale de l'entreprise [13] :

#### Documents internes

- Dossier médical.
- Fiche de renseignement.
- Fiche de visite périodique.
- Fiche de visite redéploiement.
- Fiche de visite d'embauche.
- Fiche de visite spontanée.
- Fiche d'accident de travail.

#### Documents externes

- Ordonnance.
- Demande d'examen.
- Fiche de vaccin.

## IV.2 Fiches d'analyse des documents :

Après l'analyse de chaque document, on désigne dans des tableaux qui suivent la description et les caractéristiques de chacun d'eux.

Le Tableau II.4 représente la fiche d'étude du document : **dossier médical**.

Caractéristique du document
<b>Désignation :</b> dossier médical. <b>Rôle :</b> sert à écrire les renseignements sur le patient <b>Nature :</b> interne <b>Poste créateur :</b> infirmier(e) <b>Poste destinataire :</b> médecin <b>Nombre d'exemplaire :</b> 01
Description du document
Rubrique
<ul style="list-style-type: none"> <li>➤ Informations personnelles             <ul style="list-style-type: none"> <li>• Nom patient</li> <li>• Prénom patient</li> <li>• Situation de famille</li> <li>• Adresse patient</li> <li>• N° tel patient</li> <li>• Les renseignements</li> <li>• Date de naissance</li> </ul> </li> <li>➤ Antécédents personnels             <ul style="list-style-type: none"> <li>• Accidents</li> <li>• Interventions chirurgicales</li> <li>• Intoxication                 <ul style="list-style-type: none"> <li>➤ Visites médicales (embauche, périodique, spontanée, accident de travail, redéploiement)</li> </ul> </li> </ul> </li> <li>• Opht</li> <li>• O.R.L</li> <li>• Locomoteur</li> <li>• Respiratoire</li> <li>• Cardiovasculaire</li> <li>• Digestif</li> <li>• Génito-urinaire</li> <li>➤ Examens             <ul style="list-style-type: none"> <li>• Radiologique</li> <li>• Biologique</li> <li>• Toxicologique</li> </ul> </li> <li>➤ Orientation             <ul style="list-style-type: none"> <li>• Spécialité</li> <li>• Hospitalisation</li> <li>• Avis</li> <li>• Traitement</li> </ul> </li> </ul>

- Service social

Tableau II.4 dossier médical.

Le Tableau II.5 représente la fiche d'étude du document : **fiche de renseignement**

<b>Caractéristique du document</b>
<b>Désignation</b> : fiche de renseignement <b>Rôle</b> : sert à écrire les renseignements sur le patient <b>Nature</b> : interne <b>Poste créateur</b> : infirmier(e) <b>Poste destinataire</b> : médecin <b>Nombre d'exemplaire</b> : 01
<b>Description du document</b>
<b>Rubrique</b>
<ul style="list-style-type: none"> <li>• Nom patient</li> <li>• Prénom patient</li> <li>• Adresse patient</li> <li>• N° tel patient</li> <li>• Les renseignements</li> <li>• Date de naissance</li> </ul>

Tableau II-5: Fiche de renseignement

Le Tableau II.6 représente la fiche d'étude du document : fiche pour visite périodique, redéploiement et d'embauche.

<b>Caractéristique du document</b>
<b>Désignation</b> : visite périodique, redéploiement et d'embauche <b>Rôle</b> : compte rendu d'une consultation général d'un patient <b>Nature</b> : interne <b>Poste créateur</b> : médecin <b>Nombre d'exemplaire</b> : 01
<b>Description du document</b>
<b>Rubrique</b>
<ul style="list-style-type: none"> <li>• Nom médecin</li> <li>• Prénom médecin</li> <li>• Date de consultation</li> <li>• Commentaire</li> <li>• Orl</li> <li>• Locomoteur</li> <li>• Respiratoire</li> <li>• Cardio</li> <li>• Digestif</li> </ul>

Tableau II-6 : Fiche pour les visites périodiques, redéploiement et d'embauche

Le Tableau II.7 représente la fiche d'étude du document : fiche pour visite spontanée.

<b>Caractéristique du document</b>
<b>Désignation :</b> visite spontanée <b>Rôle :</b> consultation d'un patient <b>Nature :</b> interne <b>Poste créateur :</b> médecin <b>Nombre d'exemplaire :</b> 01
<b>Description du document</b>
<b>Rubrique</b>
<ul style="list-style-type: none"> <li>• Nom médecin</li> <li>• Prénom médecin</li> <li>• Date de consultation</li> <li>• Motif</li> <li>• Diagnostic</li> </ul>

*Tableau II-7 : Fiche pour la visite spontanée*

Le Tableau II.8 représente la fiche d'étude du document : fiche pour accident de travail.

<b>Caractéristique du document</b>
<b>Désignation :</b> accident de travail <b>Rôle :</b> consultation d'un patient <b>Nature :</b> interne <b>Poste créateur :</b> médecin <b>Nombre d'exemplaire :</b> 01
<b>Description du document</b>
<b>Rubrique</b>
<ul style="list-style-type: none"> <li>• Nom médecin</li> <li>• Prénom médecin</li> <li>• Date de consultation</li> <li>• Siege lésions</li> <li>• Nature</li> </ul>

*Tableau II-8 : Fiche pour les visites d'accident de travaux*

Le Tableau II.9 représente la fiche d'étude du document : demande d'examens

<b>Caractéristique du document</b>
<b>Désignation :</b> demande d'examens <b>Rôle :</b> nom des examens à faire <b>Nature :</b> externe <b>Poste créateur :</b> médecin

<b>Poste destinataire</b> : laboratoire d'examen
<b>Nombre d'exemplaire</b> : 01
<b>Description du document</b>
<b>Rubrique</b>
<ul style="list-style-type: none"> <li>• Nom médecin</li> <li>• Prénom médecin</li> <li>• Date de consultation</li> <li>• Nom patient</li> <li>• Prénom patient</li> <li>• Date de naissance</li> <li>• Type d'examen</li> </ul>

*Tableau II-9: Fiche de demande d'examens*

Le Tableau II.10 représente la fiche d'étude du document : Ordonnance

<b>Caractéristique du document</b>
<b>Désignation</b> : ordonnance <b>Rôle</b> : prescrire un traitement pour le patient <b>Nature</b> : externe <b>Poste créateur</b> : médecin <b>Poste destinataire</b> : patient <b>Nombre d'exemplaire</b> : 01
<b>Description du document</b>
<b>Rubrique</b>
<ul style="list-style-type: none"> <li>• Nom médecin</li> <li>• Prénom médecin</li> <li>• Durée d'ouverture</li> <li>• Date de consultation</li> <li>• Nom du patient</li> <li>• Prénom du patient</li> <li>• Date de naissance du patient</li> <li>• Désignation des médicaments</li> <li>• Posologie des médicaments</li> <li>• Dosage des médicaments</li> </ul>

*Tableau II-10 : Fiche d'ordonnance.*

## V. Problématique

La gestion du nombre important de dossiers des patients dans un centre médical manuellement devient très difficile. Un médecin doit se souvenir des problèmes de ses patients, suivre le développement de leur état médical surtout pour les maladies chroniques et avoir en tête tous ce que se trouve dans leurs dossiers médicaux au moment de la consultation.

Suite aux remarques que nous avons faites au centre médical de l'ENIEM, nous avons trouvé un certain nombre de lacunes dans leur système de gestion :

- La difficulté de retrouver le dossier au moment voulu ce qui engendre un retard dans le traitement de l'information.



- Le manque de communication rapide de l'information et l'utilisation des nouvelles technologies, surtout avec le développement complémentaire des bases de données.
- Nombre important des archives qui engendre une difficulté de stockage.
- Travaux manuels fastidieux ce qui provoque un risque d'erreur et d'oubli ou chevauchement de RDV.
- La perte d'information à cause de l'encombrement et le non clarté de la petite fiche médicale qui contient plusieurs informations, chose qui peut générer l'ajout ou la suppression parfois de certaines informations utiles.
- Absence de la notion de confidentialité à cause de non séparation entre fiche médicale et dossier médical : on remarque que la secrétaire peut accéder aux informations confidentielles du patient, or le respect du secret médical impose que seul le médecin peut consulter tout le dossier.

### V.1 Présentation du travail demandé :

Dans un souci de concevoir une application avec le plus de fonctionnalités possibles et dans le but d'avoir une interface plus conviviale et plus facile à utiliser tout en étant plus efficace et après l'étude que nous avons effectuée et selon les besoins du centre nous opterons pour une application web qui aura pour objectif

- Gérer les dossiers médicaux des patients et sauvegarder dedans tout ce qui est observations des consultations, examens et traitements prescrits.
- L'intégration des bilans et examens demandés aux dossiers médicaux.
- Faciliter de la recherche et l'accès aux informations.
- Automatiser certaines tâches qui se traitent manuellement.
- Eviter la redondance des informations
- Stockage des informations sur le serveur ce qui assurera leur sécurité.

### V.2 Spécification des besoins :

Comme les bonnes questions représentent la moitié de la réponse dans la plupart des domaines, en informatique une bonne spécification des besoins est primordiale. En effet, elle représente le travail le plus délicat et le plus significatif. Une bonne spécification des besoins n'est autre que la question que doit se poser tout ingénieur au début de son travail « Qu'est ce qu'on veut que nous fassions ? ».

Dans cette section nous allons présenter notre réponse à cette question. Donc, cette phase consiste à mieux comprendre le contexte du système, il s'agit de déterminer les fonctionnalités.

L'application envisagée doit satisfaire les besoins fonctionnels qui seront exécutés par le système et les besoins non-fonctionnels qui perfectionneront la qualité logicielle du système.

### V.3 Les besoins fonctionnels :

Les besoins fonctionnels ou besoin métiers représentent les actions que le système doit exécuter, il ne devient opérationnel que s'il les satisfait.

Cette application doit couvrir principalement les besoins fonctionnels suivants

- **Gestion de l'application :** Cette partie permet l'ajout et la suppression des différents utilisateurs.
- **Gestion des dossiers médicaux :** elle s'agit de la gestion des dossiers des patients, les différentes fiches de visite et la sauvegarde des différents examens demandés.
- **Gestion des registres :** archiver toute les différentes visites et soins effectués au centre.

### V.4 Les besoins non-fonctionnels :

Ce sont les exigences qui ne concernent pas spécifiquement le comportement du système mais plutôt identifient des contraintes internes et externes du système.

Les principaux besoins non-fonctionnels de notre application se résument dans les points suivants :

- ✓ Le code doit être clair pour permettre des futures évolutions ou améliorations.
- ✓ L'ergonomie, l'application doit offrir une interface conviviale et facile à utiliser.
- ✓ La sécurité, l'application doit respecter la confidentialité des informations, il faut avoir une interface d'authentification.
- ✓ Garantir l'intégrité et la cohérence des données à chaque mise à jour et à chaque insertion.
- ✓ La convivialité, l'application doit être facile à utiliser, elle doit présenter un enchaînement logique entre les interfaces et un ensemble de liens suffisants pour assurer une navigation rapide et un texte compréhensible et lisible.
- ✓ La disponibilité, lorsque n'importe quel utilisateur désire consulter l'application, elle doit être disponible.

## VI. Conclusion

L'étude de l'existant nous a permis d'avoir une idée bien précise sur le système existant, étudier de plus près les postes de travail occupés ainsi que les documents utilisés et de localiser un certain nombre de défaillances et d'anomalies, soit au niveau organisationnel qu'informationnel et de proposer des solutions adéquates pour y remédier. Le chapitre qui suit sera consacré à l'analyse et la conception de notre application.

## I. Introduction :

Dans le but d'une meilleure organisation et avant toute réalisation d'une application informatique, tout processus de développement d'applications ou systèmes informatiques doit suivre une méthode ou une démarche bien définie, en mettant en évidence tous les objectifs tracés pour la bonne élaboration du projet souhaité.

Dans ce chapitre nous allons entamer la conception de notre application, par une phase d'analyse qui mettra en évidence les différents acteurs intervenants dans le système ainsi que leur besoins, ensuite par une phase de conception, qui s'appuie sur les résultats de cette analyse, et qui donnera la modélisation des objectifs à atteindre.

Dans le but d'avoir une meilleure analyse et de rendre la conception plus complète, nous avons utilisé UML (Unified Modeling language) et cela pour permettre de bien représenter la dynamique de notre application.

## II. Rappel sur UML :

UML (Unified Modeling language) que l'on peut traduire par «langage de modélisation unifié» est une annotation permettant de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes existant auparavant, et est devenu désormais la référence en termes de modélisation objet [14].

Il est basé sur la programmation orienté objet, cette dernière consiste à modéliser informatiquement un ensemble d'éléments d'un domaine en un ensemble d'entités informatique. Ces entités sont appelées objets.

Le méta modèle UML fournit une panoplie d'outils permettant de représenter l'ensemble des éléments du monde objet (classes, objets, ...) ainsi que les liens qui les relient.

Toutefois, étant donné qu'une seule représentation est trop subjective, UML fournit un moyen astucieux permettant de représenter diverses projections d'une même représentation grâce aux vues. Une vue est constituée d'un ou plusieurs diagrammes. On distingue deux types de vues [14]:

- 1) Les vues statiques, c'est-à-dire représentant le système physiquement
  - diagrammes d'objets
  - diagrammes de classes
  - diagrammes de cas d'utilisation
  - diagrammes de composants
  - diagrammes de déploiement
- 2) Les vues dynamiques, montrant le fonctionnement du système
  - diagrammes de séquence

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

### III. Démarche de modélisation

La modélisation comporte deux composantes :

- **L'analyse:** c'est l'étude du problème qui permet de lister les résultats.
- **La conception :** c'est la mise au point d'une solution au problème (décrit l'ensemble des moyens et des procédures qui permettent le développement des applications).

Pour notre cas nous avons adopté la démarche de modélisation représentée par la figure III.10 :

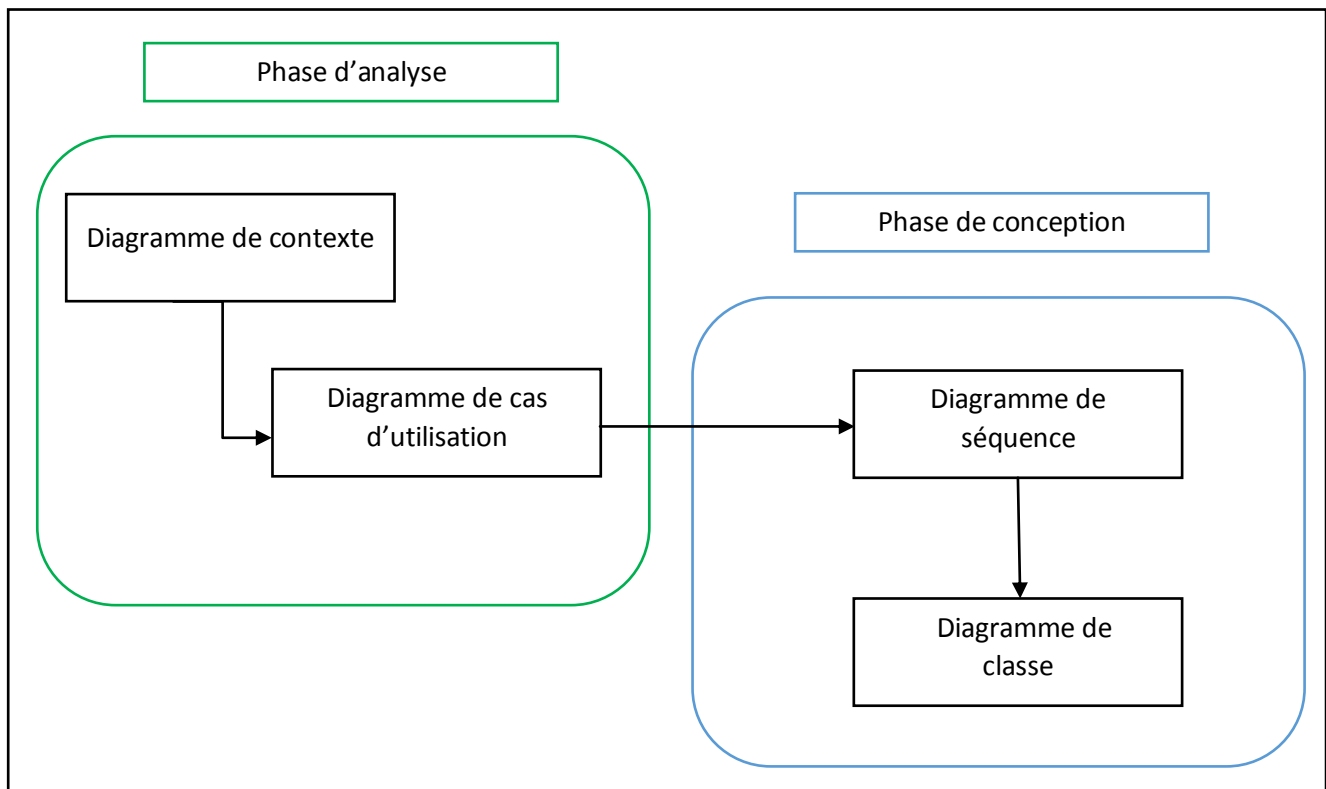


Figure III-10 :démarche de modélisation

#### III.1 Phase d'analyse :

Cette phase s'intéresse à la définition des besoins ainsi qu'au domaine couvert par l'application et cela pour objectif de spécifier de manière claire et détaillé l'application de gestion du centre médical de « l'ENIEM ». Pour se faire, il est nécessaire de déterminer globalement ce qui se trouve dans le champ de l'application.

### III.1.1 Diagramme de contexte :

Le diagramme du contexte statique délimite le domaine d'étude :

- en précisant ce qui est à la charge du système.
- en identifiant l'environnement extérieur au système étudié avec lequel ce dernier communique.

Ses composants sont [14]:

- Les acteurs externes. Un acteur externe est une entité externe au système étudié qui interagit avec le système.
- Un processus unique symbolisant le Système d'information étudié : *Nom du SI*.
- Échange entre le système étudié et son environnement.

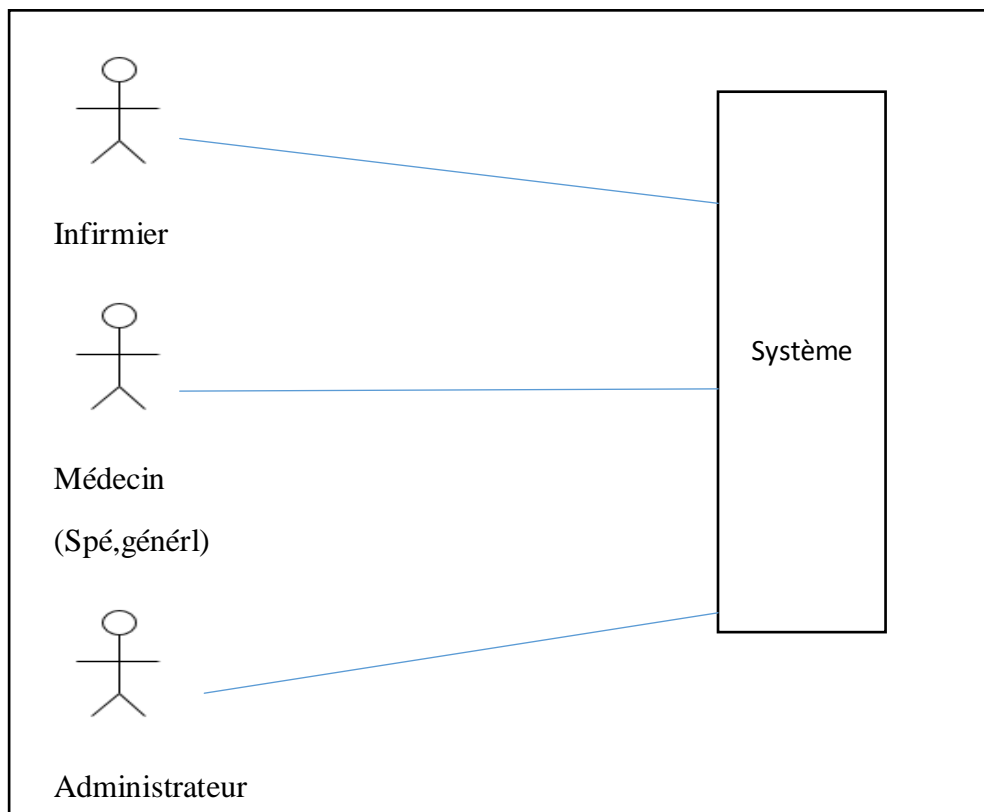


Figure III.11 : diagramme de contexte.

### III.1.2 Identification des acteurs:

Un acteur représente un ensemble cohérent de rôles joués par les utilisateurs des cas d'utilisation en interaction avec ces cas d'utilisation. En règle générale, un acteur représente un rôle qu'un homme, une machine ou même un autre système joue avec le système [1].

Notre système communique avec différents acteurs afin de réaliser un ensemble d'opérations.

Les acteurs de notre système sont :

- ✓ **Médecin spécialiste de la médecine de travail** : assure les visites d'embauches, périodiques et les visites de redéploiements.
- ✓ **Médecin généraliste** : assure les visites spontanées et les visites d'accidents de travail.
- ✓ **Infirmier** : assure les visites infirmières.
- ✓ **Administrateur** : assure la gestion de l'application (gestion des utilisateurs, privilèges,...).

Pour la structuration des informations des dossiers médicaux, nous avons pu dégager les différentes parties que devra contenir notre application.

#### ❖ **Partie identification :**

Cette partie est renseignée par l'infirmier ou le médecin, elle contient les informations personnelles du patient tels que (nom, prénom, âge,...etc.) et des informations sur le travail du patient tels que (unité, service, atelier,...etc.).

#### ❖ **Partie antécédents :**

Cette partie est renseignée par le médecin spécialiste. Elle contient la liste des antécédents personnels et familiaux du patient.

#### ❖ **Partie visite infirmière :**

Cette partie est remplie par l'infirmier, elle contient la liste des examens du patient effectués au niveau de l'infirmerie tel que (l'urine, l'audition, la vision,...) et quelques informations tels que (taille, poids,...etc.).

#### ❖ **Partie contient toutes les visites :**

Cette partie se subdivise en deux sous parties :

- Une partie contient la liste des visites d'embauche, visite périodique et visite de redéploiement qui sont assurées par le médecin spécialiste, et qui sont aussi organisées en plusieurs compartiments que nous avons vu utile de les séparer, afin d'éviter l'encombrement du dossier médicale.  
Ces compartiments sont : visite générale, examens complémentaires, orientation et aptitude au travail.
- Une partie contient la liste de visites d'accident de travail et visite spontanée qui sont assurées par le médecin généraliste.

#### ❖ **Partie traitements :**

Cette partie contient la liste des traitements prescrits aux patients.

#### ❖ **Partie examens :**

Cette partie contient les listes des examens que le patient a passer ou qu'il doit passer

### III.1.3 Identification des cas d'utilisation :

Un cas d'utilisation modélise une interaction entre le système informatique à développer et un utilisateur, autrement dit c'est l'interaction d'un acteur avec le système.

Plus précisément, un cas d'utilisation décrit un ensemble de séquences d'actions y compris des variantes, qu'un système exécute pour produire un résultat tangible pour un acteur [1].

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre system) qui interagit directement avec le système étudié.

Pour notre cas nous avons pu dégager les cas d'utilisation globale qui sont représenté par le Tableau III.11 les suivants :

Acteur	Cas d'utilisation
Infirmier	<ul style="list-style-type: none"> <li>- S'authentifier.</li> <li>- Rechercher un dossier médical.</li> <li>- Ajouter un nouveau dossier médical. <ul style="list-style-type: none"> <li>o Renseigner la partie visite infirmière.</li> <li>o Renseigner la partie la partie identification.</li> </ul> </li> <li>- Consulter un dossier médical. <ul style="list-style-type: none"> <li>o Consulter la partie visite infirmière et antécédents.</li> <li>o Consulter la partie traitements.</li> <li>o Consulter la partie identification.</li> </ul> </li> <li>- Mettre à jour un dossier médical <ul style="list-style-type: none"> <li>o Modifier la partie visite infirmière.</li> <li>o Modifier la partie la partie identification.</li> </ul> </li> </ul>
Médecin spécialiste	<ul style="list-style-type: none"> <li>- S'authentifier.</li> <li>- Rechercher un dossier médical.</li> <li>- Consulter un dossier médical. <ul style="list-style-type: none"> <li>o Consulter toutes les visites et antécédents.</li> <li>o Consulter la partie traitements.</li> <li>o Consulter les examens</li> <li>o Consulter la partie identification</li> </ul> </li> <li>- Ajouter un dossier médical. <ul style="list-style-type: none"> <li>o Renseigner la partie visite infirmière</li> <li>o Renseigner la partie la partie identification</li> </ul> </li> <li>- Mettre à jour un dossier médical. <ul style="list-style-type: none"> <li>o Ajouter une consultation (visite d'embauche, périodique et redéploiement.).</li> <li>o modifier une consultation (visite d'embauche, périodique et redéploiement.).</li> <li>o Ajouter la partie antécédents.</li> <li>o Ajouter examens.</li> <li>o Ajouter traitement.</li> <li>o Modifier la partie antécédents.</li> <li>o Modifie la partie identification.</li> <li>o Modifier la partie visite infirmière.</li> <li>o Modifier examens.</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>○ Modifier traitements.</li> <li>- Scanner les résultats du patient.</li> </ul>
Médecin généraliste	<ul style="list-style-type: none"> <li>- S'authentifier.</li> <li>- Rechercher un dossier médical.</li> <li>- Consulter un dossier médical.               <ul style="list-style-type: none"> <li>○ Consulter toutes les visites et antécédents.</li> <li>○ Consulter la partie traitements.</li> <li>○ Consulter les examens</li> <li>○ Consulter la partie identification.</li> </ul> </li> <li>- Ajouter un dossier médical.               <ul style="list-style-type: none"> <li>○ Renseigner la partie visite infirmière</li> <li>○ Renseigner la partie la partie identification</li> </ul> </li> <li>- Mettre à jour un dossier médical.               <ul style="list-style-type: none"> <li>○ Ajouter une consultation (visite spontané et accident de travail).</li> <li>○ Modifier une consultation (visite spontané et accident de travail).</li> <li>○ Modifier la partie visite infirmière.</li> <li>○ Modifier la partie identification.</li> <li>○ Ajouter examens.</li> <li>○ Ajouter traitement</li> <li>○ Modifier examens.</li> <li>○ Modifier traitement.</li> </ul> </li> <li>- Scanner les résultats du patient.</li> </ul>
Administrateur	<ul style="list-style-type: none"> <li>- S'authentifier.</li> <li>- Gérer les utilisateurs.               <ul style="list-style-type: none"> <li>○ Ajouter un utilisateur.</li> <li>○ Modifier un utilisateur.</li> <li>○ Supprimer un utilisateur.</li> </ul> </li> </ul>

*Tableau III.11 : structuration de cas d'utilisation globale.*

### Structuration des cas d'utilisations en package :

Un package est un élément d'organisation de modèles, il regroupe des éléments de modélisation selon des critères purement logiques. Les éléments contenus dans un package



sont de même nature et de même niveau sémantique et ils doivent représenter un ensemble fortement cohérent.

Dans cette phase nous avons structuré les cas d'utilisations en packages et nous avons obtenu le regroupement représenté dans le Tableau III.12.

Cas d'utilisation	Acteur	Package
S'authentifier	Médecin spécialiste, Médecin généraliste, administrateur.	<b>Connexion</b>
Consulter les examens	Médecin spécialiste, Médecin généraliste,	<b>Consultation</b>
Consulter toutes les visites et antécédents.		
Consulter la partie traitements.		
Consulter la partie visite infirmière et antécédents.	Infirmier.	
Consulter la partie traitements		
Ajouter un utilisateur	Administrateur	<b>Gestion des utilisateurs</b>
Modifier un utilisateur		
Supprimer un utilisateur		
Scanner les résultats du patient.	Médecin généraliste, Médecin spécialiste	<b>Scanner</b>
Rechercher un dossier médical.	infirmier	<b>Gestion des dossiers médicaux</b>
Ajouter un dossier médical.		
Renseigner ou modifier la partie visite infirmière.		
Renseigner ou modifie la partie la partie identification.		
Rechercher un dossier médical.	Médecin spécialiste	

Ajouter un nouveau dossier médical.		Médecin généraliste,
Ajouter ou modifier une consultation (visite d'embauche, périodique et redéploiement.).		
Ajouter ou modifier la partie antécédents et/ou vaccins.		
Modifie la partie la partie identification.		
Modifier la partie visite infirmière.		
Ajouter et /ou modifier des traitements.		
Ajouter ou /et modifier des examens.		
Rechercher un dossier médical.		
Ajouter un nouveau dossier.		
Ajouter ou modifier une consultation (visite spontané et accident de travail).		
Modifier la partie visite infirmière.		
Modifier la partie identification.		
Ajouter et /ou modifier des traitements.		
Ajouter et /ou modifier des examens.		

Tableau-III.12 : structuration des cas d'utilisations en package.

### III.1.4 Identification des scenarios :

Un scenario représente une succession particulière d'enchaînement étant l'unité de description de séquence d'actions. Un cas d'utilisation contient en général un scénario nominal et plusieurs scénarios alternatifs

Dans ce qui suit nous présentons les différents scenarios associés à chaque acteur :

#### Acteur : Infirmier

Tache	Scenario
T01 :s'authentifier	S01 : l'utilisateur demande la connexion. S02 : le système lui affiche un formulaire d'authentification. S03 : l'utilisateur saisit son login et son mot de passe, puis click sur « connecter » S04 : le système vérifie la validité des informations reçues et affiche la page d'accueil de l'espace demandé sinon il retourne vers la page d'accueil.
T02 : Rechercher un dossier médical.	S05 : l'utilisateur demande une recherche d'un dossier. S06 : le système lui affiche une barre de recherche. S07 : l'utilisateur tape le nom du dossier à rechercher. S08 : le système vérifie l'existence du dossier à rechercher et autorise l'accès, sinon affiche un message d'erreur.
T03 : consulter la partie visite infirmière et antécédents.	S09 : l'utilisateur demande la consultation de la partie visite infirmière et/ou antécédents. S10 : le système vérifie l'autorisation d'accès et affiche à l'utilisateur la partie demandée.
T04 : Consulter la partie traitements	S11: l'utilisateur demande la consultation de la partie traitements S12 : le système vérifie l'autorisation d'accès et affiche à l'utilisateur les informations de la partie traitement.
T05 : ajouter un nouveau dossier médical.	S13 : l'utilisateur demande l'ajout d'un dossier. S14 : le système lui affiche un formulaire d'ajout de dossier. S15 : l'utilisateur remplit le formulaire et click sur valider. S16 : le système vérifie les données reçues et affiche un message pour confirmer l'ajout d'un dossier.
T06 : Renseigner la partie visite infirmière.	S17 : l'utilisateur demande la partie visite infirmière. S18 : le système lui affiche le formulaire de la visite demandée. S19 : l'utilisateur remplit le formulaire et click sur valider. S20 : le système affiche un message de confirmation.

T07 : modifier la partie visite infirmière.	S21 : l'utilisateur demande la modification de la partie visite infirmière. S22 : le système lui affiche un formulaire de modification de la partie demandée. S23 : l'utilisateur saisie les informations et click sur valider. S24 : le système vérifie les données reçues et affiche un message pour confirmer la modification.
T08 : Renseigner et/ou consulter la partie identification.	S25 : l'utilisateur demande la partie visite identification. S26 : le système lui affiche le formulaire de la visite demandée. S27 : l'utilisateur remplit le formulaire et click sur valider. S28 : le système vérifie les données reçues et affiche un message pour confirmer la modification.
T09 : modifie la partie la partie identification.	S29 : l'utilisateur demande la modification de la partie identification. S30 : le système lui affiche un formulaire de modification de la partie demandée. S31 : l'utilisateur saisie les informations et click sur valider. S32 : le système vérifie les données reçues et affiche un message pour confirmer la modification.
T10 : mettre à jour le registre infirmier.	S33 : l'utilisateur demande la modification du registre de l'infirmier. S34 : le système vérifie l'autorisation d'accès et affiche les différents champs du registre infirmier. S35 : l'utilisateur remplit les champs et click sur valider. S36 : le système affiche un message de confirmation.

Tableau III.13 : récapitulatif des scénarios par tâches de l'infirmier.

**Acteur : médecin spécialiste.**

Tache	Scenario
T11 : s'authentifier	S01; S02; S03; S04;
T12 : rechercher un dossier médical.	S05; S06; S07; S08;
T13 : ajouter un nouveau dossier médical.	S11; S12; S13; S14;
T14 : Consulter toutes les visites et antécédents.	S37 : l'utilisateur demande la consultation des d'une partie parmi toutes les parties existantes. S38 : le système vérifie l'autorisation d'accès et affiche à l'utilisateur la partie demandée.
T15 : Consulter la partie traitements.	S39 : l'utilisateur demande la consultation de la partie traitements. S40 : le système vérifie l'autorisation d'accès. S41 : le système affiche à l'utilisateur les informations du registre.

T15 : consulter la partie examens	S42 : l'utilisateur demande la consultation de la partie traitements. S43 : le système vérifie l'autorisation d'accès. S44 : le système affiche à l'utilisateur les informations du registre.
T16 : Ajouter une consultation (visite d'embauche, périodique et redéploiement.).	S45 : l'utilisateur demande l'ajout d'une consultation. S46 : le système lui affiche un formulaire d'ajout de consultation. S47 : l'utilisateur remplit le formulaire et click sur valider. S48 : le système vérifie les données reçues et affiche un message pour confirmer l'ajout d'une consultation.
T17 : modifier une consultation (visite d'embauche, périodique et redéploiement.).	S49 : l'utilisateur demande la modification d'une consultation. S50 : le système lui affiche un formulaire de modification de consultation. S51 : l'utilisateur saisie les informations et click sur valider. S52 : le système vérifie les données reçues et affiche un message pour confirmer la modification.
T18 : Ajouter la partie antécédents.	S53 : l'utilisateur demande l'ajout de la partie antécédents et/ou vaccins. S54 : le système lui affiche un formulaire d'ajout de la partie demandée. S55 : l'utilisateur remplit le formulaire et click sur valider. S56 : le système vérifie les données reçues et affiche un message pour confirmer l'ajout.
T19 : modifier la partie antécédents et/ou vaccins.	S57 : l'utilisateur demande la modification de la partie antécédents et/ou vaccins. S58 : le système lui affiche un formulaire de modification de la partie demandée. S59 : l'utilisateur saisie les informations et click sur valider. S60 : le système vérifie les données reçues et affiche un message pour confirmer la modification.
T20 : Modifie la partie identification.	S61 : l'utilisateur demande la modification de la partie identification. S62 : le système lui affiche un formulaire de modification de la partie demandée. S63 : l'utilisateur saisie les informations et click sur valider. S64 : le système vérifie les données reçues et affiche un message pour confirmer la modification.

T21 : Modifier la partie visite infirmière.	S65 : l'utilisateur demande la modification de la partie visite infirmière. S66 : le système lui affiche un formulaire de modification de la partie demandée. S67 : l'utilisateur saisie les informations et click sur valider. S68 : le système vérifie les données reçues et affiche un message pour confirmer la modification.
T22 : ajouter et/ou modifier des traitements.	S69 : l'utilisateur demande l'ajout et/ou la modification des traitements. S70 : le système vérifie l'autorisation d'accès et affiche les différents champs de la partie traitements. S71 : l'utilisateur remplit les champs et click sur valider. S72 : le système affiche un message de confirmation.
T23 : ajouter et/ou modifier des examens.	S73 : l'utilisateur demande l'ajout et/ou la modification des examens S74 : le système vérifie l'autorisation d'accès et affiche les différents champs de la partie traitements. S74 : l'utilisateur remplit les champs et click sur valider. S75: le système affiche un message de confirmation.
T24 : scanner un document.	S76 : l'utilisateur demande de scanner un document. S77 : le système ordonne le scanne et demande à l'utilisateur de le valider. S78 : l'utilisateur valide le scanne et click sur sauvegarder. S79 : le système sauvegarde le document.

Tableau III.14 : récapitulatif des scénarios par tâches du médecin spécialiste.

**Acteur : médecin généraliste.**

Tache	Scenario
T25 : s'authentifier	S01; S02; S03; S04;
T26 : rechercher un dossier médical.	S05; S06; S07; S08;
T27 : ajouter un nouveau dossier médical.	S11; S12; S13; S14;
T28 : Consulter toutes les visites et antécédents.	S37 ; S38 ;
T29 : Consulter la partie antécédents.	S39 ; S40 ; S41.
T29 : Consulter la partie examens.	S42 ; S43 ; S44.
T30 : Ajouter une consultation	S80 : l'utilisateur demande l'ajout d'une consultation

(visite spontané et accident de travail).	(visite spontané et accident de travail). S81 : le système lui affiche un formulaire d'ajout de la consultation demandée. S82 : l'utilisateur remplit le formulaire et click sur valider. S83 : le système vérifie les données reçues et affiche un message pour confirmer l'ajout de la consultation.
T31 : Modifier une consultation (visite spontané et accident de travail).	S84 : l'utilisateur demande la modification d'une consultation (visite spontané et accident de travail). S85 : le système lui affiche un formulaire de modification de consultation demandée. S86 : l'utilisateur saisie les informations et click sur valider. S87 : le système vérifie les données reçues et affiche un message pour confirmer la modification.
T32 : Modifier la partie visite infirmière.	S65 ; S66 ; S67 ; S68
T33 : Modifie la partie identification.	S61 ; S62 ; S63 ; S64
T34 : ajouter et/ou modifier des traitements.	S69 ; S70 ; S71 ; S72
T34 : ajouter et/ou modifier de traitements.	S73 ; S74 ; S75.
T35 : scanner un document.	S76 ; S77 ; S78 ; S79

Tableau III.15 récapitulatif des scénarios par taches d'un médecin généraliste.

**Acteur : administrateur.**

Tache	Scenario
T38 :s'authentifier	S01; S02; S03; S04;
T39 : rechercher un utilisateur.	S88 : l'utilisateur demande une recherche d'un utilisateur. S89 : le système lui affiche une barre de recherche. S90 : l'utilisateur tape le nom de l'utilisateur à rechercher. S91 : le système vérifie l'existence de l'utilisateur à rechercher et autorise l'accès, sinon affiche un message d'erreur.
T40 : ajouter un utilisateur.	S92 : l'utilisateur demande l'ajout d'un utilisateur. S93 : le système lui affiche un formulaire d'ajout de d'utilisateur S94 : l'utilisateur remplit le formulaire et click sur valider. S95 : le système vérifie les données reçues et affiche un message pour confirmer l'ajout.

T41 : modifier un utilisateur.	S61 ; S62 ; S63 ; S64 S96 : l'utilisateur demande la modification de l'utilisateur. le système lui affiche un formulaire de modification d'utilisateur S97 : l'utilisateur remplit le formulaire et click sur valider. S98 : le système vérifie les données reçues et affiche un message pour confirmer la modification.
T42 : supprimer un utilisateur.	S99 ; S62 ; S63 ; S64 S100 : l'utilisateur demande la suppression de l'utilisateur. S101: le système affiche un message pour confirmer la suppression. S102 : l'utilisateur confirme la suppression. S103 : le système supprime l'utilisateur.

*Tableau III.16 : récapitulatif des scénarios par tâches d'un administrateur*

### III.1.5 Diagramme de cas d'utilisation :

Les diagrammes de cas d'utilisation font partie des cinq diagrammes d'UML qui servent à modéliser les aspects dynamiques des systèmes, ces diagrammes sont indispensables à la modélisation du comportement d'un système, d'un sous-système ou d'une classe [1].

On applique les diagrammes de cas d'utilisation pour visualiser, spécifier et documenter le comportement d'un élément. Ils rendent les systèmes, les sous-systèmes et les classes plus faciles à aborder et plus compréhensibles en présentant une vue externe de la façon dont ces éléments peuvent être utilisés en contexte [1].

Un diagramme de cas d'utilisation est un diagramme qui montre un ensemble de cas d'utilisation et d'acteurs, ainsi que leurs relations.



Dans cette phase, nous avons pu dégager un diagramme de cas d'utilisation globale que nous allons présenter en premier lieu par la (figure III.12) et des diagrammes de cas d'utilisation plus détaillés que nous allons voir au cours de ce chapitre.

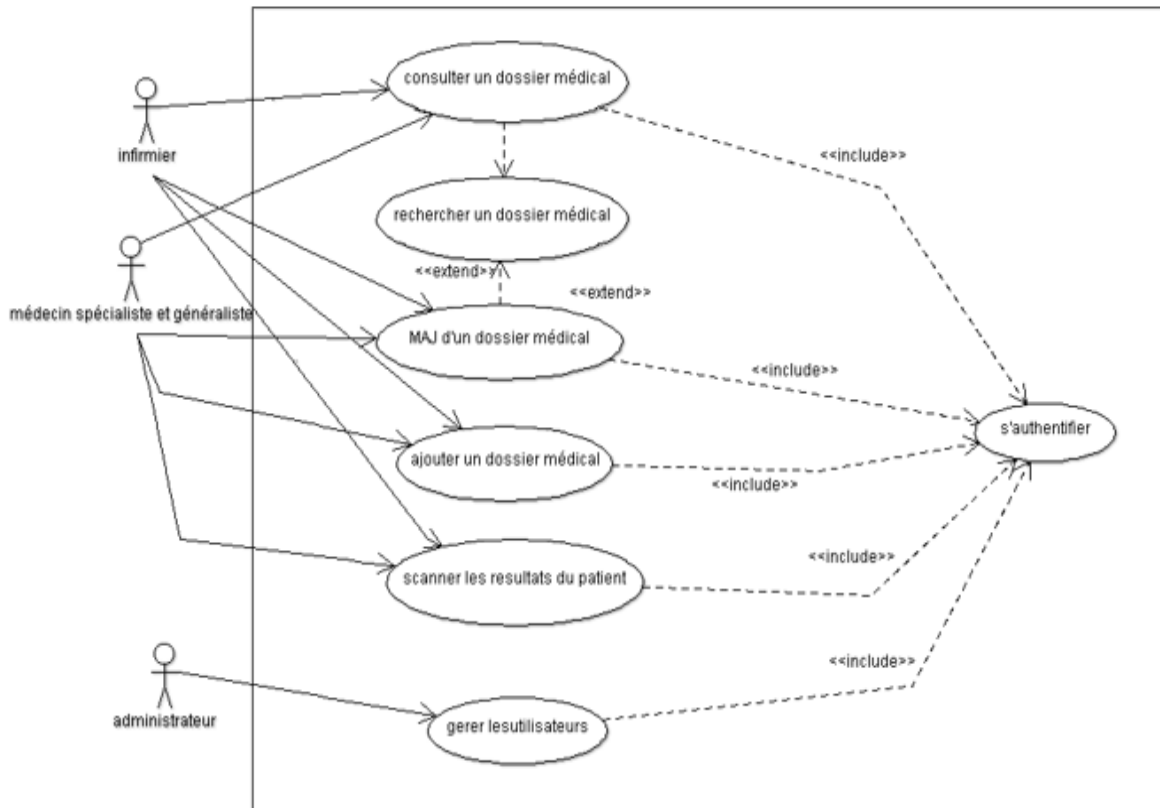


Figure III.12: diagramme de cas d'utilisation général.

Pour mieux voir les différentes fonctionnalités de notre application, nous avons élaboré des diagrammes de cas d'utilisation détaillés.

Nous allons commencer par le diagramme de cas d'utilisation de l'infirmier représenté par la figure III.13.

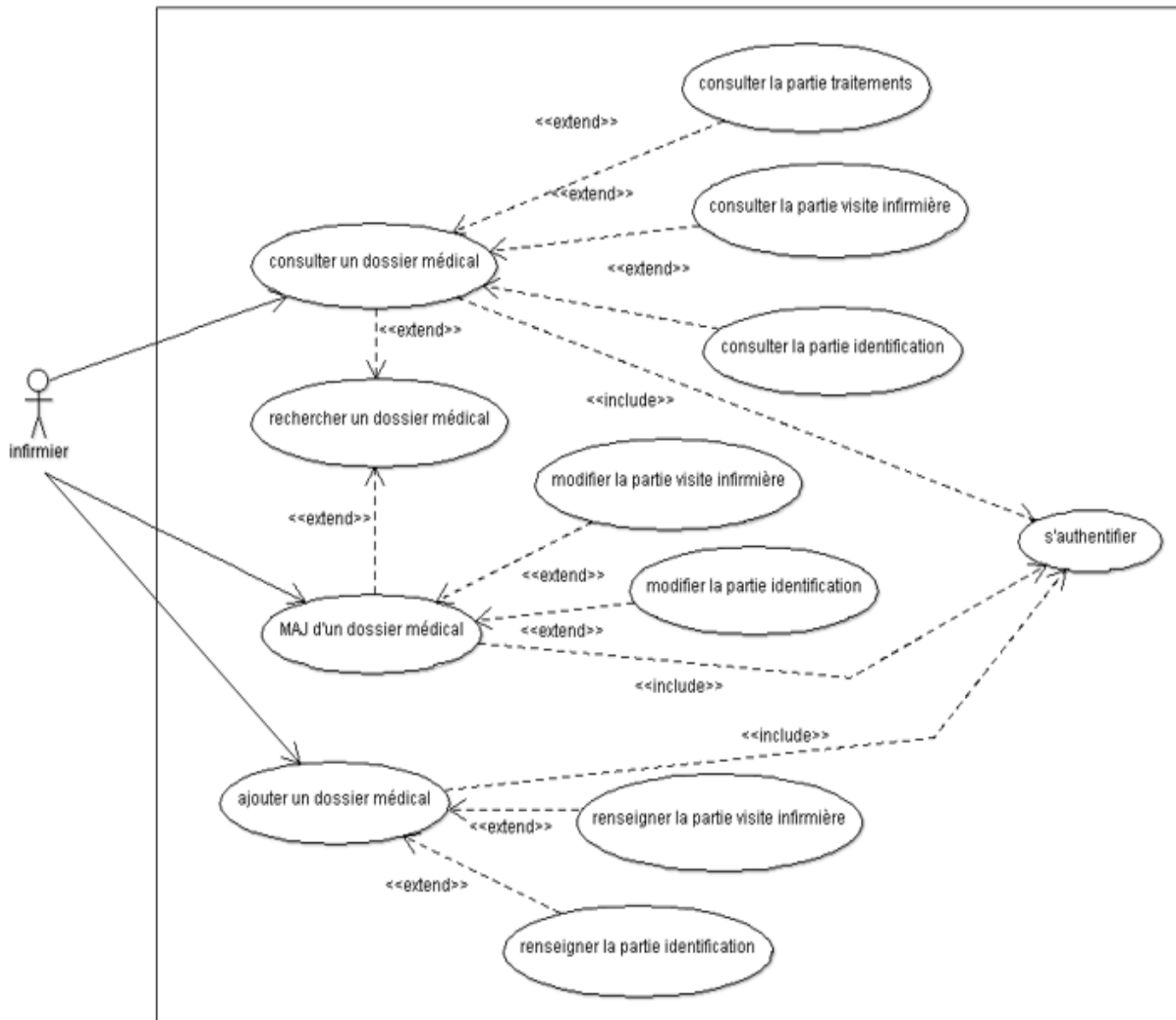


Figure III.13: diagramme de cas d'utilisation de l'infirmier.

Le diagramme de cas d'utilisation du médecin spécialiste est représenté par la figure III.14

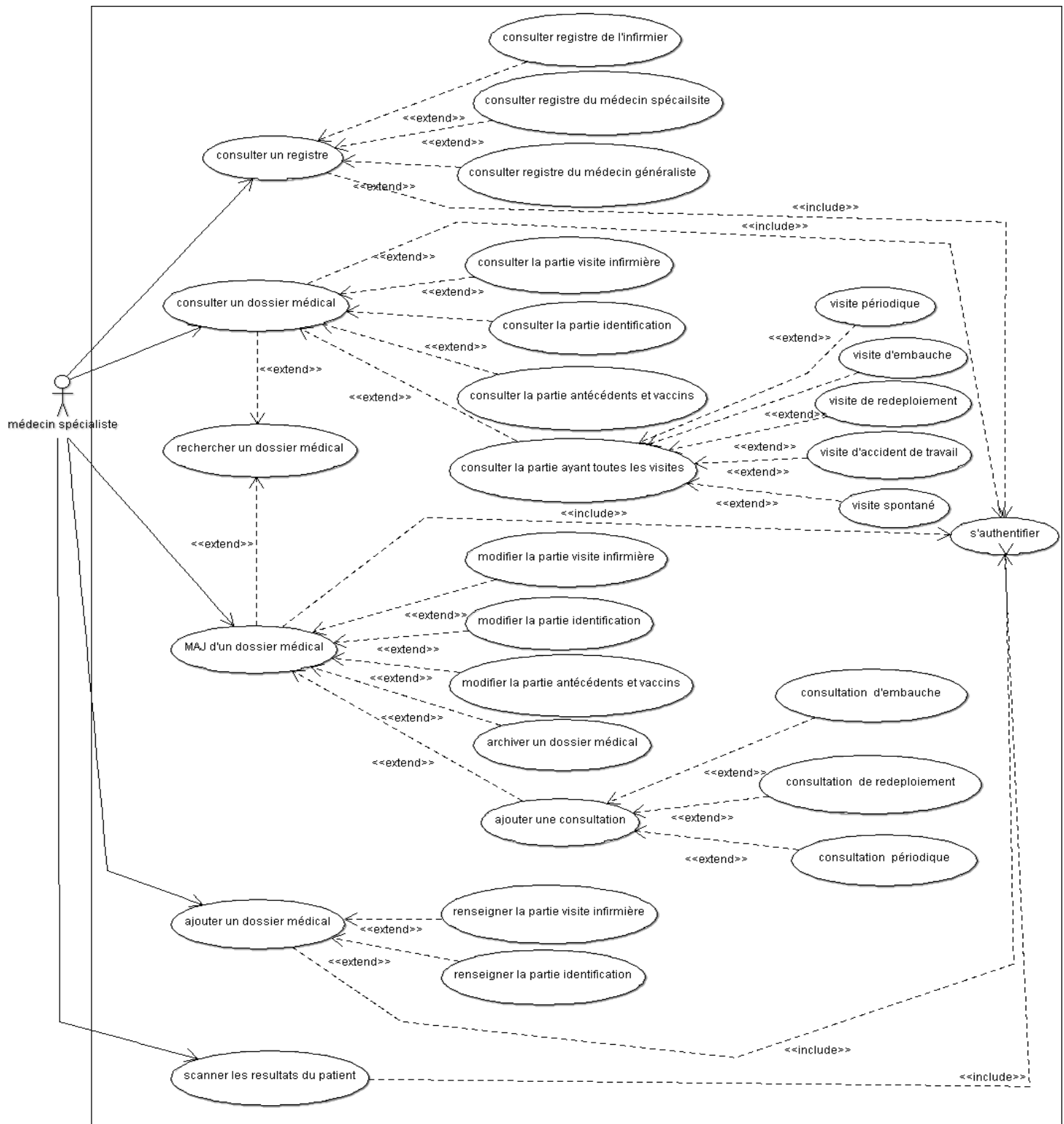


Figure III.14: diagramme de cas d'utilisation du médecin spécialiste

Le diagramme de cas d'utilisation du médecin généraliste est représenté par la figure III.15

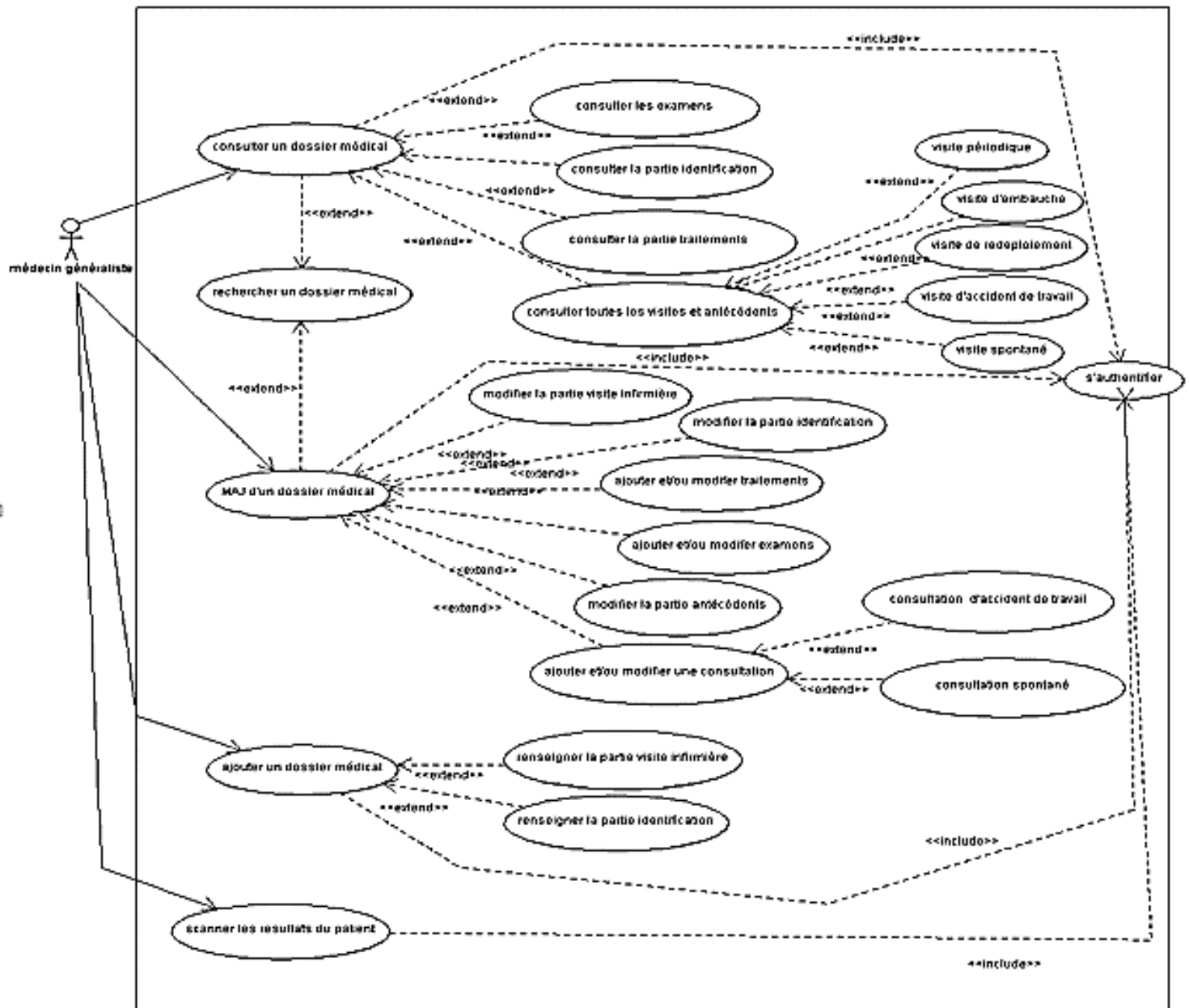


Figure III.15: diagramme de cas d'utilisation du médecin généraliste

Le diagramme de cas d'utilisation de l'administrateur est représenté par la figure III.16

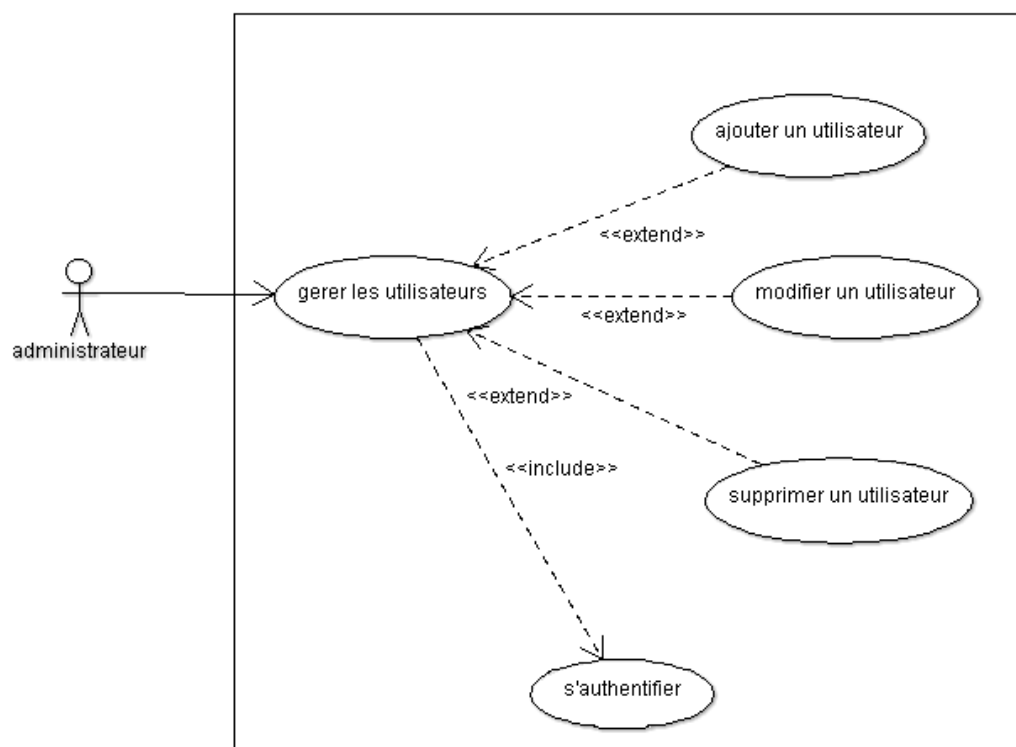


Figure III.16: diagramme de cas d'utilisation de l'administrateur.

## III.2 Conception

La conception est l'étape qui suit l'analyse. Elle consiste à modéliser et à détailler tous les éléments de modélisation issus de la phase d'analyse.

### III.2.1 Les diagrammes de séquences

Les diagrammes de séquences permettent de décrire les scénarios de chaque cas d'utilisation en mettant l'accent sur la chronologie des opérations en interaction avec les objets.

Dans la description des cas d'utilisation nous avons pu identifier les scénarios. Dans ce qui suit nous allons traduire quelques-uns en diagrammes de séquences.

La figure III.17 Représente le diagramme des séquences du cas d'utilisation « s'authentifier »

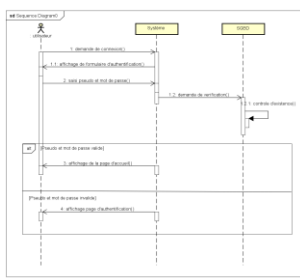
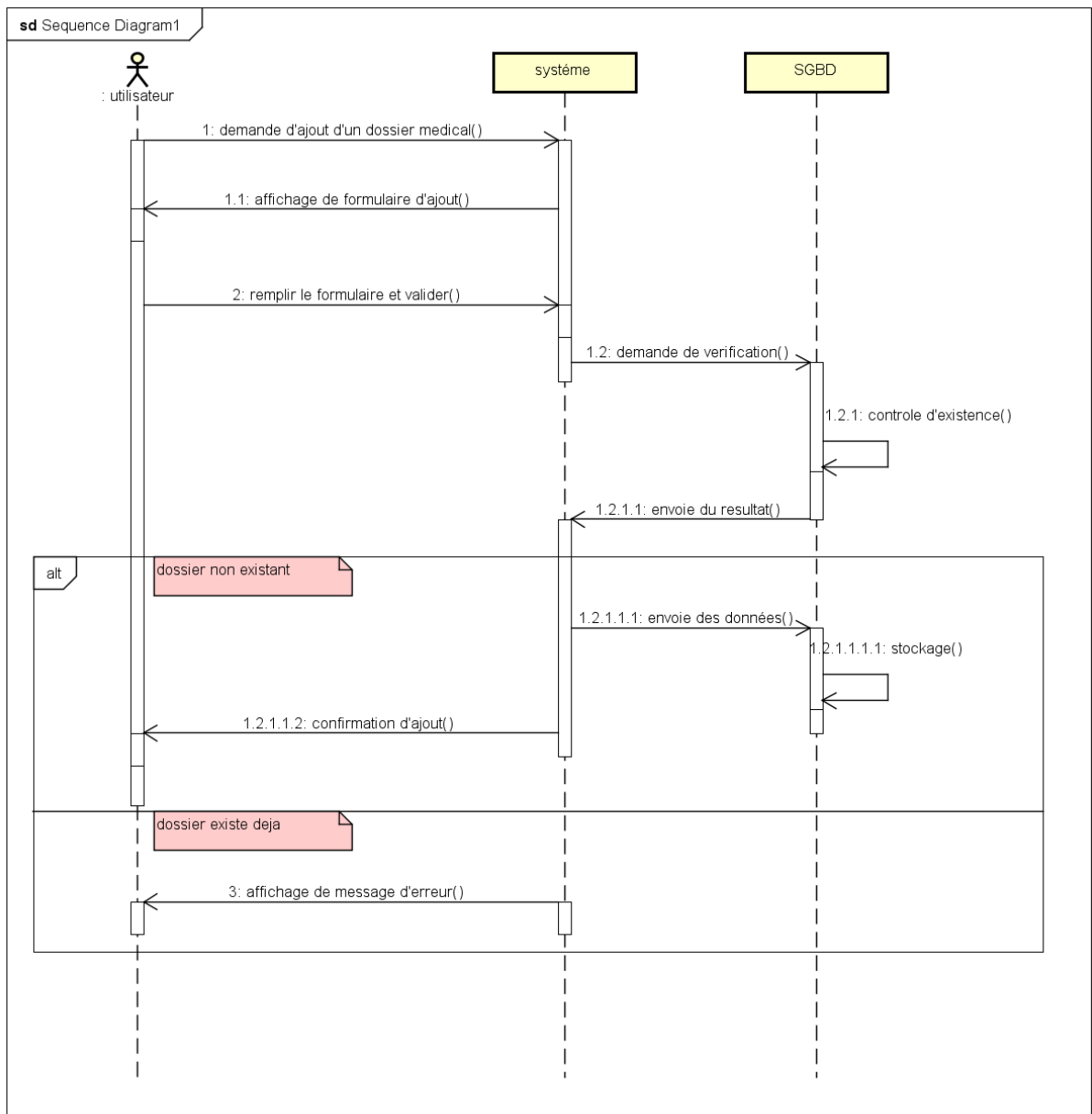


Figure III-17: Diagramme de séquences du cas d'utilisation "s'authentifier"

Lafigure III.18 Représente le diagramme de séquences du cas d'utilisation « ajouter un dossier médical »



powered by Astah

Figure III-18 diagramme de séquences du cas d'utilisation "ajouter un dossier médical"

La figure III.19 Représente le diagramme de séquences du cas d'utilisation « Consulter un dossier médical ».

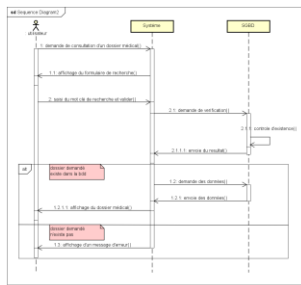


Figure III-19 : diagramme de séquences du cas d'utilisation "consulter un dossier médical"

### III.2.2 Diagrammes de classe

Les diagrammes de classes sont les diagrammes les plus courants dans la modélisation des systèmes orientés. Ils représentent un ensemble de classes, d'interfaces et de collaborations ainsi que leurs relations. On utilise des diagrammes de classes pour modéliser la vue de conception statique d'un système. Pour l'essentiel, cela implique la modélisation de vocabulaire du système, la modélisation de collaboration ou la modélisation de schémas.



Les diagrammes de classes sont importants non seulement pour visualiser, construire, spécifier et documenter des modèles structurels, mais également pour construire des systèmes exécutables [1].

La figure III.20 Montre le diagramme de classes globales qui est conçu à partir des règles de gestion suivantes :

- ✓ Un utilisateur peut être un médecin spécialiste, un généraliste ou bien un infirmier.
- ✓ Un infirmier peut effectuer des traitements.
- ✓ Un médecin spécialiste assure les visites générales.
- ✓ Un médecin généraliste assure les consultations soit spontanée ou accident de travail.
- ✓ Un médecin peut prescrire des examens.
- ✓ Un patient peut passer plusieurs traitements.
- ✓ Un patient peut avoir plusieurs antécédents
- ✓ Un patient peut passer plusieurs consultations
- ✓ Un patient doit passer des visites générales, aux moins une fois lors de son embauche.
- ✓ Un patient peut avoir plusieurs examens à passer.
- ✓ Une visite générale doit englober tous les appareils suivants : peau et muqueuses, ophtalmologique, ORL, locomoteur, respiratoire, cardio vasculaire, digestif, génito-urinaire, neuro psychisme, hémato et ganglions, endocrine et les explorations fonctionnelles.

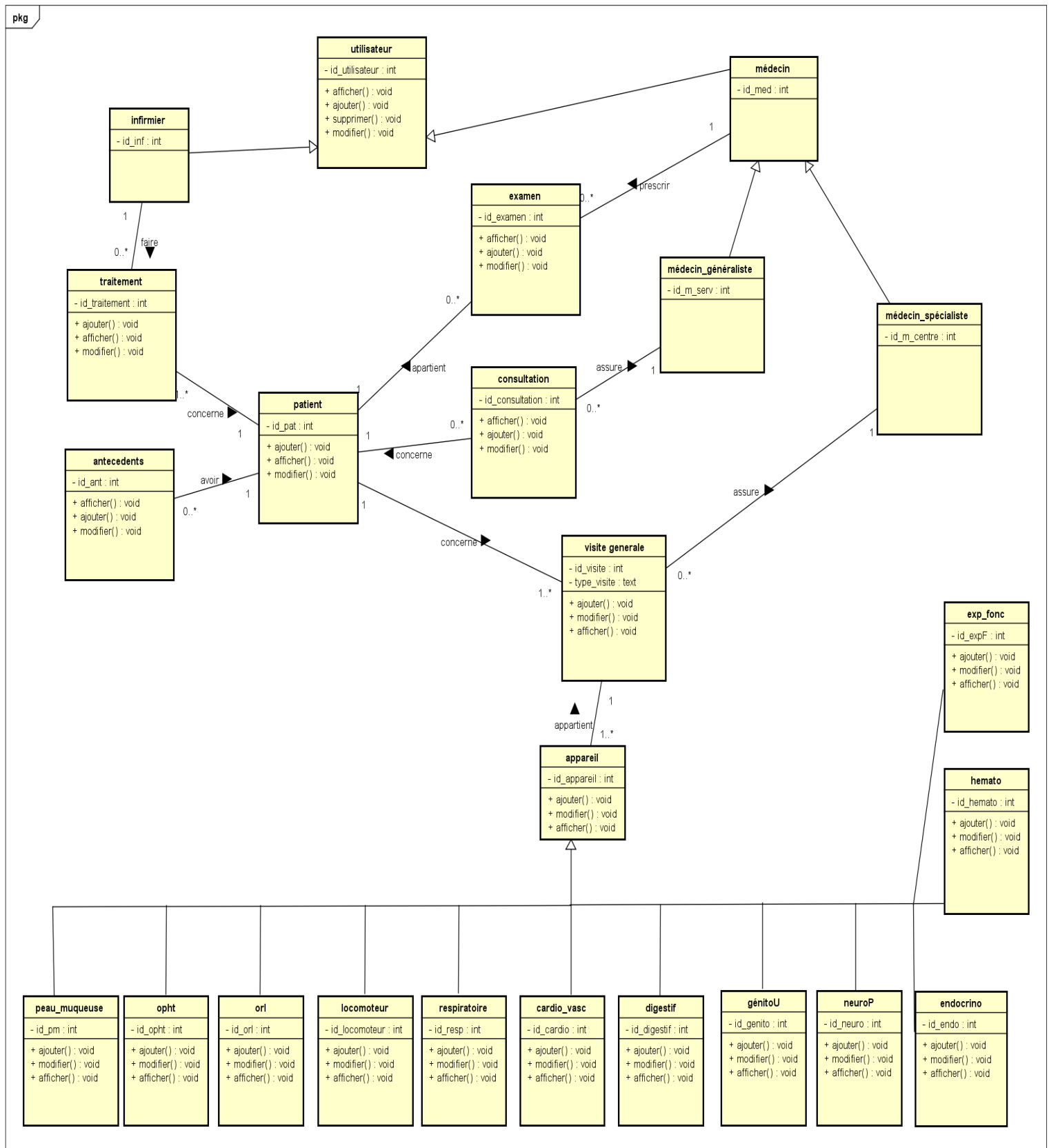


Figure III.20 : diagramme de classe.

Dans le tableau III.16 nous détaillons toutes les classes présentes dans le diagramme avec leurs attributs et méthodes.

CLASSE	ATTRIBUT	METHODES
Utilisateur	Id_utilisateur Nom_utilisateur Prenom_utilisateur MDP_utilisateur Mail_utilisateur	Afficher () Ajouter () Modifier () Supprimer ()
Médecin	Id_medecin	
Medecin_spécialiste	Id_m_Spec Spécialité	
Médecin_généraliste	Id_m_generaliste	
Infirmier	Id_infirmier	
Patient	Id_patient Nom_patient Prenom_patient Sexe Date_naissance Lieu_naissance Adresse Situation_familiale Num_sec_soc Serv_national Service_travail Atelier_travail Formation Aptitude_travail	Afficher () Ajouter () Modifier () Supprimer ()
Antécédents	Id_antecedents Type_ant Description_ant	Afficher () Ajouter () Modifier () Supprimer ()
Traitement	Id_traitement Type_trait Date_traitement	Afficher () Ajouter () Modifier () Supprimer ()
Examen	Id_examen Type_examen Resultat_examen	Afficher () Ajouter () Modifier () Supprimer ()
Visite_generale	Id_visite Type_visite Date_visite Poids Taille Pt Urines Audition_OD Audition_OG Vision_SCorrection	Afficher () Ajouter () Modifier () Supprimer ()

	Vision_ACorrection Orientation Reponse_orientation	
Consultation	Id_consultation Type_Consultation Motif_consultation Observation Traitement Date_consultation	Afficher () Ajouter () Modifier () Supprimer ()
Appareil	Id_appareil	Afficher () Ajouter () Modifier () Supprimer ()
Peau_Muqueuse	Id_PM Allergie	
Opht	Id_opht Larmoiement Douleur_opht Taches Œil_rouge Fatigue Autre_opht	
Orl	Id_orl Sifflement Engines_Répétées Epistaxis Entend_Mal Otorrhées Eternument Rhinorrhée Autre_orl	
Locomoteur	Id_locomoteur Douleur_loc Gêne_mouvement Fatigabilité Autre_loc	
Respiratoire	Id_respiratoire Toux Dysnée Exoectorations Douleurs_thoraciques Tabac Autre_resp	
Cardio_vasc	Id_cardio Palpitations Douleurs_marche Dyspnée Autre_card Pouls TA	

	Varices Cyanose	
Digestif	Id_digestif Appetit Transit Selles Alcool Pyrosis Irritants Vomissement Rectoragies Douleurs_Abdo Autre_digetif	
GenitoU	Id_genito Mictions Pallakiurie Hematurie Brulures Coliques Dysurie Pertes Monstruations Autre_genito	
NeuroP	Id_neuro Sommeil Céphalées Vertiges Peur_vide Perte_connaissance Paréses Paresthésie Autre_neuro	
Hemato	Id_hemato Echymoses Hémorragies	
Endocrino	Id_endo Obésité_familiale Maigreur_familiale	
Exp_fonc (Explorations fonctionnelles)	Id_expF Fonc_respiratoire Fonc_circulatoire Fonc_motrice	

Tableau 17: dictionnaire de classes et des attributs

### III.2.3 Modèle relationnel :

A partir du modèle conceptuel que nous avons effectué, nous avons pu réaliser notre modèle relationnel. En ce qui suit, nous décrivons la liste des tables conçues à partir du diagramme de classe que nous avons cité précédemment avec leurs attribut et clés.

**Infirmier** (Id\_infirmier, Nom\_infirmier, Prenom\_infirmier, MDP\_infirmier, Mail\_infirmier).

**Médecin\_spécialiste** (Id\_m\_spec, Nom\_m\_spec, Prenom\_m\_spéc, MDP\_m\_spéc, Spécialité)

**Médecin\_généraliste** (Id\_m\_generaliste, Nom\_m\_gnr, Prenom\_m\_gnr, MDP\_m\_gnr, Mail\_m\_gnr).

**Patient** (Id\_patient, Nom\_patient, Prenom\_patient, Sexe, Date\_naissance, Lieu\_naissanceAdresse, Situation\_familiale, Num\_sec\_soc, Serv\_national, Service\_travail, Atelier\_travail, Formation, Aptitude\_travail).

**Antécédents** (Id\_antecedents, Type\_ant, Description\_ant, Id\_patient\*).

**Traitement** (Id\_traitement, Type\_traitement, Date\_traitement, Id\_patient\*, Id\_infirmier\*).

**Examen** (Id\_examen, Type\_examen, Resultat\_examen, Id\_patient\*, Id\_m\_spécialiste\*, Id\_m\_généraliste\*).

**Visite\_générale** (Id\_visite, Type\_visite, Date\_visite, Poids, Taille, Pt, Urines, Audition\_OD, Audition\_OG, Vision\_SCorrection, Vision\_ACorrection, Orientation, Id\_m\_spécialiste\*, Id\_patient\*).

**Consultation** (Id\_consultation, Type\_consultation, Date\_consultation, Type\_consultation, Motif\_consultation, Observation, Traitement, Id\_m\_généraliste\*, Id\_patient).

**Opht** (Id\_opht, Larmoiement, Douleur\_opht, Taches, Œil\_rouge, Fatigue, Autre\_opht, Id\_visite\*).

**Orl** (Id\_orl, Sifflement, Engines\_réprétées, Epistaxis, Entend\_mal, Otorrhées, Eternument, Eternument, Rhinorrhée, Autre\_orl, Id\_visite\*).

**Locomoteur** (Id\_locomoteur, Douleur\_loc, Gène\_mouvement, Fatigabilité, Autre\_loc, Id\_visite\*).

**Peau\_muqueuse** (Id\_PM, Allergie, Id\_visite\*).

**Respiratoire** (Id\_respiratoire, Toux, Dyspnée\_resp, Expectorations, Douleurs\_thoraciques, Tabac, Autre\_resp, Id\_visite\*).

**Cardio\_vasc** (Id\_cardio, Palpitations, Douleurs\_marche, Dyspnée\_cardio, Pouls, TA, Varices, Cyanose, Autre\_cardio, Id\_visite\*).

**Digestif** (Id\_digéstif, Appétit, Transit, Selles, Alcool, Pyrosis, Irritants, Vomissement, Rectorragies, Rectorragies, Douleurs\_abdo, Autre\_cardio, Id\_visite\*).

**Génito\_urinaire** (Id\_génito, Mictions, Pollakiurie, Hématurie, Brulures, Coliques, Dysurie, Pertes, Menstruations, Autre\_genito, Id\_visite\*).

**Neuro-psychisme** (Id\_neuro, Sommeil, Céphalées, Vertiges, Peur\_vide, Perte\_connaissance, Parésies, Paresthésie, Autre\_neuro, Id\_visite\*).

**Hémato** (Id\_hemato, Ecchymoses, Hémorragies, Id\_visite\*).

**Endocrino** (Id\_endo, Obésité\_familiale, Maigreur\_familiale, Id\_visite\*).

**Exp\_fonc** (Id\_expF, Fonc\_respiratoire, Fonc\_circulatoire, Fonc\_motrice, Id\_visite\*).

**NB** : chaque mot souligné s'agit d'une clé primaire, et chaque mot suivi d'une étoile s'agit d'une clé étrangère.

## IV. Conclusion

Dans ce chapitre, nous avons proposé une démarche de modélisation pour développer notre application, en se basant sur l'UML. Pour la partie analyse nous avons présenté les divers cas d'utilisations, le diagramme de contexte et identifier les différents scénarios possibles pour chaque tâche d'un utilisateur, puis en phase conception, nous avons élaboré les diagrammes de séquences et le diagramme de classe, et enfin, nous avons défini notre modèle relationnel qui représente les différentes tables qui seront présente dans notre base de données.

Le chapitre suivant sera consacré à la présentation des outils et langages de programmation qui vont nous permettre de mettre en œuvre notre application, ainsi que quelques interfaces qui la décrivent.

## I. Introduction :

Après avoir présenté dans le chapitre précédent les différentes étapes d'analyse et de conception, nous allons présenter dans ce chapitre son implémentation, commençant par présenter les outils et l'environnement de développement adopté dans notre application, ensuite nous présenterons certaines interfaces développées pour illustrer notre application.

## II. Environnement de développement :

### II.1 La plateforme Java Enterprise Edition :

Java Enterprise Edition est une plate-forme fortement orientée serveur pour le développement et l'exécution d'applications distribuées. Elle est composée de deux parties essentielles [17] :

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

Sun propose une implémentation minimale des spécifications de Java Enterprise Edition : le Java EE SDK. Cette implémentation permet de développer des applications respectant les spécifications mais n'est pas prévue pour être utilisée dans un environnement de production. Ces spécifications doivent être respectées par les outils développés par des éditeurs tiers [17].

Java Enterprise Edition permet une grande flexibilité dans le choix de l'architecture de l'application en combinant différents composants. Ce choix dépend des besoins auxquels doit répondre l'application mais aussi des compétences dans les différentes API de Java Enterprise Edition.

L'architecture d'une application se découpe idéalement en au moins trois tiers :

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

L'utilisation de Java Enterprise Edition pour développer et exécuter une application, offre plusieurs avantages :

- Une architecture d'application basée sur des 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,...
- La possibilité de choisir les outils de développement et les serveurs d'applications utilisés qu'ils soient commerciaux ou libres.

## II.2 Framework :

### II.2.1 Hibernate :

Hibernate est une solution open source de type ORM (Object Relational Mapping) qui permet de faciliter le développement de la couche persistance d'une application. Hibernate permet donc de représenter une base de données en objets Java et vice versa[18].

Hibernate facilite la persistance et la recherche de données dans une base de données en réalisant lui-même la création des objets et les traitements de remplissage de ceux-ci en accédant à la base de données. La quantité de code ainsi épargnée est très importante d'autant que ce code est généralement fastidieux et redondant[18].

Hibernate a besoin de plusieurs éléments pour fonctionner :

- une classe de type javabeau qui encapsule les données d'une occurrence d'une table
- un fichier de configuration qui assure la correspondance entre la classe et la table (mapping)
- des propriétés de configuration notamment des informations concernant la connexion à la base de données

Une fois ces éléments correctement définis, il est possible d'utiliser Hibernate dans le code des traitements à réaliser. L'architecture d'Hibernate est donc la suivante :

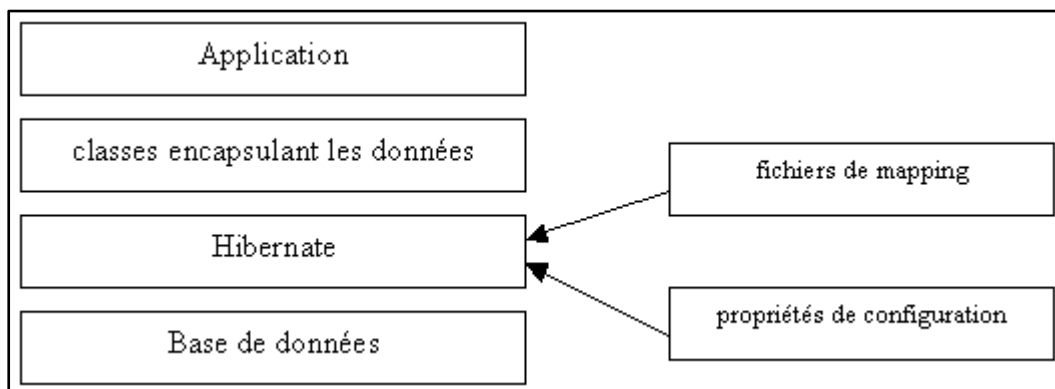


Figure IV.21 architecture d'Hibernate

### II.2.2 JSF (Java Server Faces) :

JSF est un Framework de type MVC, destiné aux applications web respectant l'architecture J2EE.

Le premier objectif de JSF, est de procurer un environnement de développement permettant de construire une interface de type web, sans devoir toucher au code HTML et JavaScript. Ceci est réalisé par la mise en place d'un mapping entre l'HTML et les objets concernés. JSF est donc basé sur la notion de composants, comparable à celle de Swing, où l'état de ces composants est sauvegardé puis restauré au retour de la requête[19].

Le principal atout de JSF est qu'il tient compte des différentes expériences acquises non seulement avec des technologies de type standard comme les Servlets, les JSP,... mais aussi avec les technologies de type Framework comme Struts. Dès lors, JSF propose un Framework qui puisse être mis en œuvre par des outils facilitant le développement d'applications web. Rien n'empêche que JSF puisse être utilisé par codage à la volée, néanmoins, l'utilisation d'un outil approprié est vivement recommandé afin d'exploiter rapidement les possibilités de JSF[19].

L'implémentation de JSF nécessite l'implémentation de deux fichiers qui sont les suivants[19] :

- Face-config.xml :

Le Face-config.xml est le fichier de configuration, qui permet de gérer les beans, et les règles de navigation entre les pages.

- Web.xml :

Le web.xml permet de définir l'emplacement des pages JSP, ainsi que certaines contraintes de sécurité concernant l'accès à des pages. Il établit également l'endroit où se situe le Face Servlet, Face Servlet est la classe motrice de chaque application JSF.

Chaque application JSF a sa propre Faces Servlet, qui gère toutes les informations relatives à la requête courante.

### II.2.3 Bootstrap :

Bootstrap est un Framework CSS, mais pas seulement, puisqu'il embarque également des composants HTML et JavaScript. Il comporte un système de grille simple et efficace pour mettre en ordre l'aspect visuel d'une page web. Il apporte du style pour les boutons, les formulaires, la navigation... Il permet ainsi de concevoir un site web rapidement et avec peu de lignes de code ajoutées[20].

Le projet Bootstrap a été créé au départ par Mark Otto et Jacob Thornton pour répondre à des besoins internes de développement de cette entreprise au niveau de l'uniformisation de l'aspect des pages web. Il s'agissait juste de stylisation CSS, mais le Framework s'est ensuite enrichi de composants JavaScript[19].

Il a été mis à disposition du public sous licence Apache. Le Framework en est actuellement à la version 3. Celle-ci a pris un virage particulier en intégrant l'aspect « responsive » par défaut, alors qu'auparavant cette fonctionnalité faisait l'objet d'un fichier séparé. Cette version est même déclarée comme « mobile-first ». Avec l'utilisation croissante d'appareils mobiles, le Framework s'est adapté pour offrir une solution censée couvrir tous les besoins.

### III. Outils de développement:

#### III.1 Serveur d'application (glassfish) :

GlassFish est le nom du serveur d'applications Open Source Java EE 5 et désormais Java EE 7 avec la version 4.1 dont le développement a été initié et est aujourd'hui encore dirigé par SUN[21] .

Il faut tout d'abord noter que GlassFish est bel et bien un serveur d'application Java EE complet. Il surpasse donc les capacités de Tomcat par exemple qui ne propose qu'un conteneur de JSP/servlet sans le support des EJB entre autres. GlassFish joue donc directement dans la cour des Websphere et autres mais surtout dans celle de JBoss avec lequel il partage son ouverture de code. Le serveur d'application de SUN est disponible sous une double licence Common Development and Distribution License (CDDL) et GNU General Public License (GPL) avec exception du classpath. Le projet glassFish a été initié en 2005 et est aujourd'hui considéré comme un produit stable[21].

#### III.2 Netbeans :

NetBeans IDE est un environnement de développement intégré (IDE) gratuit, open source, il permet de développer des applications de bureau, mobiles et Web. L'IDE prend en charge le développement d'applications dans divers langages, notamment Java, HTML5, PHP et C ++. L'EDI fournit un support intégré pour le cycle de développement complet, de la création de projet au débogage, au profilage et au déploiement. L'EDI s'exécute sous Windows, Linux, Mac OS X et d'autres systèmes UNIX[22].

L'IDE fournit une prise en charge complète des technologies JDK 8 et des améliorations Java les plus récentes. C'est le premier IDE à prendre en charge JDK 8, Java EE 7 et JavaFX 2. L'EDI prend entièrement en charge Java EE en utilisant les dernières normes pour Java, XML, les services Web et SQL et prend entièrement en charge le serveur GlassFish de Java EE[22] .



*Figure IV.22 : interface de l'IDE Netbeans*

### III.3 MySQL:

MySQL fait référence au Structured Query Language, 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, PostgreSQL et Microsoft SQL Server [23].

#### III.3.1 MySQL est un système de gestion de base de données :

Une base de données est un ensemble organisé de données. Cela peut aller d'une simple liste de courses au supermarché à une galerie de photos, ou encore les grands systèmes d'informations des multinationales. Pour ajouter, lire et traiter des données dans une base de données, vous avez besoin d'un système de gestion de base de données tel que le serveur MySQL. Comme les ordinateurs sont très bons à manipuler de grandes quantités de données, le système de gestion de base de données joue un rôle central en informatique, aussi bien en tant qu'application à part entière, qu'intégré dans d'autres logiciels[23].

#### III.3.2 MySQL est un serveur de base de données relationnelle :

Un serveur de base de données stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de l'ensemble. Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête. Le *SQL* dans « MySQL » signifie « *Structured Query Language* » : le langage standard pour les traitements de base de données[23].

#### III.3.3 Bref historique :

La première version de MySQL est apparue le 23 mai 1995. Il a d'abord été créé pour un usage personnel à partir de mSQL en s'appuyant sur le langage de bas niveau ISAM qu'ils trouvaient trop lent et trop rigide. Ils ont créé une nouvelle interface SQL en gardant la même API que mSQL.

MySQL est passé de la version 3.23.19 (juin 2000) à la Version 8.0 : Version (avril 2018)[23].

## IV. Langage de programmation :

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

### IV.1 Langage coté serveur :

- **Java** langage orienté objet pour la création des différentes classes java.
- **HQL (Hibernate Query Language)** : c'est un langage d'interrogation des classes persistantes avec une syntaxe proche du SQL. Cependant les objets manipulés sont les classes et membres du mapping contrairement au SQL qui réalise les requêtes directement sur la base de données. Aussi, le HQL contient des fonctionnalités spécifiques au modèle objet [24].

### IV.2 Langage coté client :

- **XHTML (Extensible Hypertext Markup Language)** : est un langage de balisage servant à écrire des pages pour le World Wide Web. Conçu à l'origine comme le successeur de HTML, XHTML se fonde sur la syntaxe définie par SGML sur laquelle repose HTML. Il s'agissait en effet à l'époque de tirer parti des bénéfices techniques attendus de la simplification offerte par XML [25].
- **XML (eXtensible Markup Language)** : pour la création de fichiers mapping relationnel-objet et pour la déclaration. Il est en quelque sorte un langage HTML amélioré permettant de définir de nouvelles balises. Il s'agit effectivement d'un langage permettant de mettre en forme des documents grâce à des balises [26].

## V. Modèle physique des données :

Ce modèle nous donne la représentation physique des tables de la base de données du système étudié

**Table Médecin\_spécialiste :**

Attribut	Description	Type	Clé
Id_m_spec	Identifiant du médecin spécialiste	Numérique	Primaire
Nom_m_spec	Nom du médecin spécialiste	Caractère	
Prenom_m_spéc	Prénom du médecin spécialiste	Caractère	
MDP_m_spéc	Mot de passe du médecin spécialiste	Caractère	
Mail_m_spéc	Mail du médecin spécialiste	Caractère	
Spécialité	Spécialité du médecin	Caractère	

*Tableau IV-18 : Table Médecin\_spécialiste*

**Table Médecin\_généraliste :**

Attribut	Description	Type	Clé
Id_m_generaliste	Identifiant du médecin généraliste	Numérique	Primaire
Nom_m_gnr	Nom du médecin généraliste	Caractère	
Prenom_m_gnr	Prénom du médecin généraliste	Caractère	
MDP_m_gnr	Mot de passe du médecin généraliste	Caractère	
Mail_m_gnr	Mail du médecin généraliste	Caractère	

Tableau IV-19 : Table Médecin\_généraliste

**Table Infirmier :**

Attribut	Description	Type	Clé
Id_infirmier	Identifiant de l'infirmier	Numérique	Primaire
Nom_infirmier	Nom de l'infirmier	Caractère	
Prenom_infirmier	Prénom de l'infirmier	Caractère	
MDP_infirmier	Mot de passe de l'infirmier	Caractère	
Mail_infirmier	Mail de l'infirmier	Caractère	

Tableau IV-20 : Table Médecin\_généraliste

**Table Patient :**

Attribut	Description	Type	Clé
Id_patient	Identifiant du patient	Numérique	Primaire
Nom_patient	Nom du patient	Caractère	
Prenom_patient	Prénom du patient	Caractère	
Sexe	Sexe	Caractère	
Date_naissance	Date de naissance du patient	Date	
Lieu_naissance	Lieu de naissance du patient	Caractère	
Adresse	Adresse du patient	Caractère	
Situation_familiale	Situation familiale du patient	Caractère	
Num_sec_soc	Numéro de sécurité sociale du patient	Caractère	
Serv_national	Etat du patient par rapport au service national	Caractère	
Service_travail	Service où travaille le patient	Caractère	
Atelier_travail	Atelier où travaille le patient	Caractère	

Formation	Formation	Caractère	
Aptitude_travail	Aptitude au travail	Caractère	

Tableau IV-21 : Table Patient

**Table Antécédents :**

Attribut	Description	Type	Clé
Id_antecedents	Identifiant de l'antécédent	Numérique	Primaire
Type_ant	Type d'antécédent	Caractère	
Description_ant	Description de l'antécédent	Caractère	
Id_patient	Identifiant du patient	Numérique	Étrangère

Tableau IV-22 : Table Antécédents

**Table Traitement :**

Attribut	Description	Type	Clé
Id_traitement	Identifiant du traitement	Numérique	Primaire
Type_traitement	Type de traitement	Caractère	
Date_traitement	Date de traitement	Date	
Id_patient	Identifiant du patient	Numérique	Étrangère
Id_infirmier	Identifiant de l'infirmier	Numérique	Étrangère

Tableau IV-23 : Table Traitement

**Table Examen :**

Attribut	Description	Type	Clé
Id_examen	Identifiant de l'antécédent	Numérique	Primaire
Type_examen	Type d'examen	Caractère	
Resultat_examen	Résultat de l'examen	Caractère	
Id_patient	Identifiant du patient	Numérique	Étrangère
Id_m_spécialiste	Identifiant du médecin spécialiste	Numérique	Etrangère
Id_m_généraliste	Identifiant du médecin généraliste	Numérique	Etrangère

Tableau IV-24 : Table Examen

**Table Visite\_générale :**

Attribut	Description	Type	Clé
Id_visite	Identifiant de la visite	Numérique	Primaire
Type_visite	Type de la visite	Caractère	
Date_visite	Date de la visite	Date	

Poids	Poids	Numérique	
Taille	Taille	Numérique	
Pt	Pt	Caractère	
Urines	Urines	Caractère	
Audition_OD	Audition d'œil droit	Caractère	
Audition_OG	Audition d'œil gauche	Caractère	
Vision_SCorrection	Vision sans correction	Caractère	
Vision_ACorrection	Vision avec correction	Caractère	
Orientation	Orientation	Caractère	
Reponse_orientation	Réponse d'orientation	Caractère	
Id_m_spécialiste	Identifiant du médecin spécialiste	Numérique	Etrangère
Id_patient	Identifiant du patient	Numérique	Étrangère

Tableau IV-25 : Table Visite\_générale

**Table Consultation :**

Attribut	Description	Type	Clé
Id_consultation	Identifiant de la consultation	Numérique	Primaire
Type_consultation	Type de la consultation	Caractère	
Date_consultation	Date de la consultation	Date	
Type_consultation	Type de la consultation	Numérique	
Motif_consultation	Motif de la consultation	Numérique	
Observation	Observation	Caractère	
Traitement	Traitement	Caractère	
Id_m_généraliste	Identifiant du médecin généraliste	Numérique	Etrangère
Id_patient	Identifiant du patient	Numérique	Étrangère

Tableau IV-26 : Table Consultation

**Table Opht :**

Attribut	Description	Type	Clé
Id_opht	Identifiant d'opht	Numérique	Primaire
Larmoiement	Larmoiement	Caractère	
Douleur_opht	Douleur ophtalmologique	Caractère	
Taches	Taches	Caractère	
Œil rouge	Œil rouge	Caractère	
Fatigue	Fatigue	Caractère	
Autre_opht	Autre symptôme ophtalmologique	Caractère	
Id_visite	Identifiant de la visite	Numérique	Etrangère

Tableau IV-27 : Table Opht.



**Table Orl :**

Attribut	Description	Type	Clé
Id_orl	Identifiant d'orl	Numérique	Primaire
Sifflement	Sifflement	Caractère	
Engines_réprétées	Engines répétées	Caractère	
Epistaxis	Epistaxis	Caractère	
Entend_mal	Entend mal	Caractère	
Otorrhées	Otorrhées	Caractère	
Eternument	Eternument	Caractère	
Rhinorrhée	Rhinorrhée	Caractère	
Autre_orl	Autre symptôme orl	Caractère	
Id_visite	Identifiant de la visite	Numérique	Etrangère

*Tableau IV-28 : Table Orl***Table Locomoteur :**

Attribut	Description	Type	Clé
Id_locomoteur	Identifiant d'orl	Numérique	Primaire
Douleur_loc	Sifflement	Caractère	
Gêne_mouvement	Engines répétées	Caractère	
Fatigabilité	Epistaxis	Caractère	
Autre_loc	Autre symptôme locomoteur	Caractère	
Id_visite	Identifiant de la visite	Numérique	Etrangère

*Tableau IV-29 : Table Locomoteur***Table Peau\_muqueuse :**

Attribut	Description	Type	Clé
Id_PM	Identifiant de peau et muqueuse	Numérique	Primaire
Allergie	Allergie	Caractère	
Id_visite	Identifiant de la visite	Numérique	Etrangère

*Tableau IV-30 : Table Locomoteur***Table Respiratoire :**

Attribut	Description	Type	Clé
Id_respiratoire	Identifiant de respiratoire	Numérique	Primaire
Toux	Toux	Caractère	
Dyspnée_resp	Dyspnée respiratoire	Caractère	
Expectorations	Expectorations	Caractère	
Douleurs_thoraciques	Douleurs thoraciques	Caractère	
Tabac	Tabac	Caractère	
Autre_resp	Autre symptôme respiratoire	Caractère	
Id_visite	Identifiant de la visite	Numérique	Etrangère

Tableau IV-31 : Table Locomoteur

**Table Cardio\_vasc :**

Attribut	Description	Type	Clé
Id_cardio	Identifiant de cardiovasculaire	Numérique	Primaire
Palpitations	Palpitations	Caractère	
Douleurs_marche	Douleurs à la marche	Caractère	
Dyspnée_cardio	Dyspnée cardiovasculaire	Caractère	
Pouls	Pouls	Caractère	
TA	TA	Caractère	
Varices	Varices	Caractère	
Cyanose	Cyanose	Caractère	
Autre_cardio	Autre symptôme cardiovasculaire	Caractère	
Id_visite	Identifiant de la visite	Numérique	Etrangère

Tableau IV-32 : Table Cardio\_vasc

**Table Digestif :**

Attribut	Description	Type	Clé
Id_digéstitif	Identifiant de cardiovasculaire	Numérique	Primaire
Appétit	Appétit	Caractère	
Transit	Transit	Caractère	
Selles	Selles	Caractère	
Alcool	Alcool	Caractère	
Pyrosis	Pyrosis	Caractère	
Irritants	Irritants	Caractère	
Vomissement	Vomissement	Caractère	
Rectorragies	Rectorragies	Caractère	
Douleurs_abdo	Douleurs abdominale	Caractère	
Autre_cardio	Autre symptôme cardiovasculaire	Caractère	
Id_visite	Identifiant de la visite	Numérique	Etrangère

Tableau IV-33 : Table Digestif

**Table Génito\_urinaire:**

Attribut	Description	Type	Clé
Id_génito	Identifiant génito-urinaire	Numérique	Primaire
Mictions	Mictions	Caractère	
Pollakiurie	Pollakiurie	Caractère	
Hématurie	Hématurie	Caractère	
Brulures	Brulures	Caractère	
Coliques	Coliques	Caractère	
Dysurie	Dysurie	Caractère	
Pertes	Pertes	Caractère	
Menstruations	Menstruations	Caractère	
Autre_genito	Autre symptôme génito-urinaire	Caractère	
Id_visite	Identifiant de la visite	Numérique	Etrangère

*Tableau IV-34 : Table Génito-urinaire***Table Neuro-psychisme :**

Attribut	Description	Type	Clé
Id_neuro	Identifiant neuro-psychisme	Numérique	Primaire
Sommeil	Sommeil	Caractère	
Céphalées	Céphalées	Caractère	
Vertiges	Vertiges	Caractère	
Peur_vide	Peur du vide	Caractère	
Perte_connaissance	Perte de connaissance	Caractère	
Parésies	Parésies	Caractère	
Paresthésie	Paresthésie	Caractère	
Autre_neuro	Autre symptôme neuro-psychisme	Caractère	
Id_visite	Identifiant de la visite	Numérique	Etrangère

*Tableau IV-35 : Table neuro-psychisme***Table Hémato :**

Attribut	Description	Type	Clé
Id_hemato	Identifiant hémato	Numérique	Primaire
Ecchymoses	Ecchymoses	Caractère	
Hémorragies	Hémorragies	Caractère	
Id_visite	Identifiant de la visite	Numérique	Etrangère

Tableau IV-36 : Table Hémato

**Table Endocrino :**

Attribut	Description	Type	Clé
Id_endo	Identifiant endo	Numérique	Primaire
Obésité_familiale	Obésité familiale	Caractère	
Maigreur_familiale	Maigreur familiale	Caractère	
Id_visite	Identifiant de la visite	Numérique	Etrangère

Tableau IV-37 : Table Endocrino

**Table Exp\_fonc :**

Attribut	Description	Type	Clé
Id_expF	Identifiant explorations fonctionnelles	Numérique	Primaire
Fonc_respiratoire	Fonctions respiratoire	Caractère	
Fonc_circulatoire	Fonctions circulatoire	Caractère	
Fonc_motrice	Fonctions motrice	Caractère	
Id_visite	Identifiant de la visite	Numérique	Etrangère

Tableau IV-38 : Table Exp\_fonc

## VI. Présentation de quelques interfaces de notre application :

- **Interface d'authentification :**

Cette interface permet l'authentification.



The screenshot displays a web interface for authentication. At the top, the title "Authentification" is centered in a bold, black, serif font. Below the title, a prompt "Please enter your email and password" is centered in a smaller, black, serif font. There are two input fields: the first is a light yellow rectangular box containing the text "achour"; the second is a similar light yellow rectangular box containing seven dots, representing a password. Below these two fields is a wide, orange rectangular button with the text "Se connecter" centered in a white, sans-serif font. The entire interface is enclosed in a thin black rectangular border.

Figure IV.23 Interface d'authentification.

- **Interface liste des patients :**

Cette interface nous montre la liste des patients.

Liste des patients					Ajouter un patient
Rechercher..					
identifiant	Nom	Prenom	Securité social	Service	Details
1	berrahma	mouloud	25855555	climatisation	<a href="#">details</a>
2	oulkadi	salim	987987654654	serviceT	<a href="#">details</a>
3	berrahma	achour	02145632	climatiseur	<a href="#">details</a>
4	hassouna	salim	0154623842	psy	<a href="#">details</a>
5	ait amirat	adem	215435821	barbo	<a href="#">details</a>
6	ait amirat	adem	215435821	barbo	<a href="#">details</a>
7	mihoubi	sofiane	214523652	climatiseur	<a href="#">details</a>
8	rouba	siham	548523658	administration	<a href="#">details</a>
9	yefsah	maeva	312522335	climatiseur	<a href="#">details</a>
10	mesdour	smail	789542514	climatiseur	<a href="#">details</a>
11	berrahma	lamine	65498765321	administration	<a href="#">details</a>
12	berrahma	lamine	65498765321	administration	<a href="#">details</a>

Figure IV.24 Interface liste des patients.

- **Interface liste des consultations :**

Cette figure nous montre la liste des consultations.

Type	Motif	Date de consultation	Observation	Traitement	Medecin	Tache
type	motifC	12/05/2018	observation	traitement		<button>modifier</button>
accident	intoxication	2018-09-13	rien	huile d'olive		<button>modifier</button>
type	motif	2018-09-22	dade	aezed		<button>modifier</button>
type	motif	2018-09-22	dade	injection		<button>modifier</button>

Type :

Date de consultation :

Motif :

Observation :

Traitement :

Figure IV.25 Interface liste des consultations.

- **Interface détail de la visite :**

Cette interface nous montre le détail de la visite :

The screenshot displays a web application interface for managing medical visits. At the top, there is a navigation bar with tabs: 'details', 'Visites' (active), 'Consultations', 'Traitements', and 'Examens'. A 'deconnexion' button is located in the top right corner. Below the navigation bar, the main content area is titled 'Details de la visite'. Under this title, there are sub-tabs: 'Général', 'Opht' (active), 'O.R.L.', 'Locomoteur', and 'Respiratoire'. The 'Opht' tab is selected, showing a form for recording eye examination results. The form includes the following fields and checkboxes:

- Larmoiement : ☒
- Douleur : ☐
- Oeil rouge : ☒
- Taches devant les yeux : ☐
- Fatigue : ☒
- Autres :

A blue 'modifier' button is positioned at the bottom right of the form.

Figure IV. 26détail de la visite.

## VII. Conclusion :

Dans ce dernier chapitre, nous avons présenté l'étape de réalisation de notre application ainsi, nous avons présenté les différents outils de développement et les langages de programmations utilisés pour le développement de notre application, par la suite nous avons donné un aperçu de l'application réalisée.



## Conclusion générale :

Le système d'information s'impose actuellement comme un élément stratégique pour l'entreprise, il devient un point d'appui incontournable pour accompagner la mise en place d'une gestion d'information d'une façon relationnelle et efficace. L'objectif assigné à notre travail est de concevoir et réaliser une application web (Java EE) pour la gestion du centre de médecine de travail de l'ENIEM.

Durant notre stage pratique effectué au sein de l'entreprise nationale « ENIEM », nous avons étudié le fonctionnement du centre de médecine de travail de cette même entreprise, ce qui nous a permis de constater, que l'ensemble des traitements effectués au sein de ce centre se fait manuellement, ce qui engendre un certain nombre de problèmes tels que : la difficulté de stockage, retard dans la recherche d'information ainsi, que le manque de confidentialité.

Afin de remédier à ces problèmes, nous avons conçu et réalisé une application web(Java EE) qui permet d'organiser et d'automatiser la gestion de ce centre et qui est destinée au (médecin, infirmier et l'administrateur). Cela dans le but d'assurer l'accès instantané aux données et une sécurisation de ces dernières, ce qui simplifie le travail administratif et réduit l'effort humain.

Pour mener à terme notre projet, nous avons jugé nécessaire de présenter quelques généralités et concepts de base sur les applications web, ainsi que la technologie Java EE.

Ensuite, nous avons présenté notre organisme d'accueil, et étudié le système existant afin de déterminer les anomalies et essayer d'y remédier ainsi, nous avons présenté les différentes étapes de conception et de réalisation qui nous ont permis d'aboutir à une application java EE répondant aux principaux problèmes,

L'application à laquelle nous avons abouti, adopte l'architecture (client / serveur). Elle Est désormais accessible depuis tous les ordinateurs des utilisateurs concernés, facilitant ainsi la gestion du centre et allègent la masse de travail effectué par les utilisateurs et leur offre plus de souplesse et d'efficacité dans l'exécution des tâches.

La réalisation de ce projet nous a permis d'approfondir nos connaissances en système d'information des entreprises, de découvrir le domaine de la médecine de travail, et de voir sur le champ de travail les différentes tâches effectuées par le personnel médical. Comme elle nous a permis de :

- Nous familiariser avec plusieurs langages de programmation à savoir Java EE.
- Enrichir nos connaissances sur les applications java, comment les développer les déployer et les invoquer au sein du serveur d'application GlassFish.
- se familiariser avec les environnements de développement orienté web, à savoir : Netbeans, MySQL pour les bases de données.
- s'initier les concepts et les outils de conception UML.

Cependant quelques améliorations et extensions peuvent être appropriées à cette application.

En guise de perspectives, les possibilités d'évolutions de ce projet sont nombreuses tels que : l'ajout d'un module de statistique et un module pour la rédaction des documents,...etc.

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

## Résumé

Durant ces dernières années l'informatique s'est imposée d'une manière très impressionnante dans les entreprises, et cela grâce à son apport extraordinaire dans le traitement automatique de l'information.

L'informatique est de plus en plus utilisée dans tous les domaines d'activités y compris celui de la médecine de travail auquel nous rattacherons d'ailleurs notre étude.

Dans le cadre de notre projet de fin d'étude nous avons intégré le département d'informatique du service prestation technique de l'entreprise nationale « ENIEM » sise à OUED AISSI, d'où une tâche nous a été confiée qui consiste à la réalisation d'une application web qui permet d'organiser et d'automatiser la gestion de centre de médecine du travail de l'entreprise.