

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU



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

Mémoire de Fin d'Etudes
de MASTER ACADEMIQUE

Domaine : Mathématiques et Informatique

Filière : Informatique

Spécialité : Système Informatique

Présenté par

TEBRI Djedjiga

M'ZYENE Katia

Thème

Conception et réalisation d'une application javaEE pour la gestion du centre médical au sein de l'ENIEM.

Mémoire soutenu le 04/juillet/2016 Devant le jury composé de :

Président : M^r RAHMANI RABAH

Encadreur : M^r HABET Mohammed-Said

Examineur: M^{me} BOUSNINA Lila

Examineur : M^{elle} YESLI Yasmine

Sommaire

Introduction	1
Chapitre I : Java EE et Framework	
I.1- Introduction :	3
I.2- JavaEE :	4
I.2.1. Définition :	4
I.2.2. Le serveur d'application :	4
I.3- Les architectures Web.....	5
I.3.1- Les types d'architectures :	5
I.3.1.1 - L'architecture 2tiers :	6
I.3.1.2- l'architecture 3tiers :	6
I.3.2- Les types d'architectures en Java EE	7
I.4- Le modèle MVC :	7
I.4.1- Le modèle MVC type1.....	8
I.4.2- Le modèle MVC de type2 :	9
I.5. Framework :	10
I.5.1- Définition :	10
I.5.2- Objectif d'un framework :	11
I.5.3- Composants d'un framework :	11
I.5.3.1- Bibliothèque :	11
I.5.3.2- Structure :	11

I.5.4- Framework Hibernate:	12
I.5.4.1- Definition Hibernate:	12
I.5.4.2- Architecture d'Hibernate :	12
I.5.4.3- Avantages et inconvénients d'Hibernate:	12
I.5.5- Le Framework Spring	13
I.5.5.1-:Définition	13
I.6.EJB	13
I.6.1.Définition	13
I.6.2. les types d'EJB.....	13
I.7- Conclusion :	14

Chapitre II : L'organisme d'accueil et étude de l'existant

Parie I : L'organisme d'accueil :	16
II .1- Introduction	16
II.2- : Présentation générale de l'entreprise ENIEM	16
II.2.1 Historique :	16
II.2.2 - Les activités principales de l'ENIEM	16
II.2.3- Les missions principales de l'ENIEM	16
II.2.4- Les objectifs de l'ENIEM	17
II.2.5- Activités de l'ENIEM :	17
II.3- Organisation générale de l'ENIEM.....	18
II.3.1- L'Organigramme de l'unité prestation technique	19
II.3.2- L'organigramme de centre médecine :	20

Partie II : Etude de poste.	20
II.1- Définition d'un poste de travail :	20
II.2. Les postes de travail étudiés	20
II.3- Fiche d'analyse des postes :	20
II.4- Diagramme des flux	22
II.4.1- Définition	22
II.4.2- Formalisme graphique	22
II.4.3- Table explicative des acteurs de domaine d'étude.....	22
II.4.4- Représentation du diagramme des flux existant	23
II.4.5- Description des flux	24
II.5- Etude des documents et des registres	24
II.5.1- Introduction.....	24
II.5.2- Liste des documents utilisés	24
II.5.3- Etude des documents utilisés :	25
II.6- Problématique.....	30
II.7- Objectifs de cette étude	30
II.8- Conclusion.....	31

Chapitre III : Analyse et Conception

III.1-Introduction.....	33
III.2-Rappel sur l'UML	33

III.3-Phase d'analyse :	34
III.3.1- Identification des acteurs :	34
III.3.2- Les fonctionnalités du système :	35
III.3.3- Identification des cas d'utilisation	35
III.3.4- Structuration des cas d'utilisation en packages :	36
III.3.5- Identification des scenarios	38
III.3.6- Diagramme de cas d'utilisation	41
III.4-Conception	46
III.4.1 Diagramme de séquences.....	46
III.4.2- Diagramme de classe	51
III.5Le modèle physique :	53
III.6- Conclusion	57

Chapitre IV : Réalisation

IV.1- Introduction.....	58
Partie 1 : Description des outils de développement	58
I- Outils de développement	58
I-1- Langage de programmation (Java)	58
I-2- IDE (Netbeans) :	59

I-3- JSF :	59
I-3-1- Structure d'une application JSF :	59
I-4- Les serveurs déployés :	60
I-4-1- Le serveur d'application (appache tomcat) :	60
I-4-2- Le serveur de données(Oracle) :	60
I-6- Le framework (Hibernate)	61
I-7- EJB	61
Partie 2 : Présentation de l'application	62
IV .2- Conclusion	72
Conclusions générale.....	74
Références bibliographique	

Liste des Figures

Chapitre I : Java EE et Framework

Figure I.1 : Schéma modèle MVC de type1.

Figure I.2 : Schéma le modèle MVC de type2.

Chapitre II: L'organisme d'accueil et étude de l'existant

Figure II.1 : organigramme générale de l'ENIEM.

Figure II.2 : organigramme de l'unité prestation technique.

Figure II.3 : Organigramme du centre médical.

Figure II.4. Le schéma du diagramme des flux.

Figure II.5. : Diagramme des flux existant.

Chapitre III : Conception et réalisation.

Figure III.1. : Composantes de la modélisation UML.

Figure III.2 : Diagramme de cas d'utilisation général.

Figure III.3. : Diagramme de cas d'utilisation pour le personnel paramédicale.

Figure III.4. : Diagramme de cas d'utilisation pour le médecin

Figure III.5 : Diagramme de cas d'utilisation pour l'Administrateur

Figure III.6 : Diagramme de séquence simple de cas d'utilisation « S'authentifier ».

Figure III.7 : Diagramme de séquence simple de cas d'utilisation « Ajouter un nouveau Patient.

Figure III.8 :Diagramme de séquence simple de cas d'utilisation « Supprimer un compte Utilisateur».

Figure III.9: Diagramme de séquence simple de cas d'utilisation « Consulter un dossier Médical

Figure III.10: Diagramme de classe.

Chapitre IV : Réalisation

Figure IV.1. : Présentation de l'application.

Figure IV.2. : La connexion établie et les différentes tables de notre BDD.

Figure IV.3. : La connexion établie et les différentes tables de notre BDD.

Figure IV.4. : Interface Authentification de l'administrateur.

Figure IV.5. : Interface Espace administrateur.

Figure IV.6. : Interface Gestion des utilisateurs.

Figure IV.7. : Interface Gestion des utilisateurs.

Figure IV.8. : Interface Espace médecin.

Figure IV.9. : Interface Ajout nouveau patient.

Figure IV.10. : Interface visite d'embauche.

Figure IV.11. : Interface visite périodique.

Figure IV.12. Interfaces du Secrétaire.

Liste des tableaux

Chapitre II: L'organisme d'accueil et étude d l'existant

Tableau II.1.: Poste du médecin.

Tableau II.2. : Poste de l'infirmière.

Tableau II.3 : Description des flux.

Tableau II.4. : Liste des documents utilisés.

Tableau II. 5. : Fiche de Renseignement.

Tableau II. 6. : Fiche pour l'Ordonnance.

Tableau II.7. : Fiche pour les visites périodiques, redéploiement, d'embauche.

Tableau II.8. : Fiche pour la visite spontanée.

Tableau II.9. : Fiche pour la visite Accident de travail.

Tableau II.10. : Fiche pour les Examens.

Tableau II.11. : Fiche pour les Vaccins.

Chapitre III :Analyse et conception.

Tableau III.1. : Structuration des cas d'utilisation globale.

Tableau III.2.: Structuration des cas d'utilisation en package.

Tableau III.3.: Récapitulatif des scénarios par tâches d'un personnel paramédical.

Tableau III.4 : Récapitulatif des scénarios par tâches d'un Personnel médical.

Tableau III.5. : Récapitulatif des scénarios par tâches d'un Administrateur.

Listes des Références:

- [1] : François-Marie Colonna « L'architecture Client – Serveur » Octobre 2002.
- [2] : George et Olivier Gardain, " le client/serveur", Edition: Eyrolles 2000.
- [2] : <http://sardes.inrialpes.fr/people/krakowia>.,
- [2] : Robert ORFALI, Dan HARKEY et Jery EDWARDS, traduction de François Leroy et Jean-pierre GOUT ; « Client/Serveur Guide de service ».
- [3] : <http://sardes.inrialpes.fr/people/krakowia>.
- [4] : <http://sardes.inrialpes.fr/people/krakowia>.
- [5] : SUPPORT DE COURS IUT de Nice – Département INFORMATIQUE.
- [6] : <http://cours-pdf.com/cours-informatique/guide-pratique-de-netbeans.html>].
- [7] : www.javafr.com.
- [8] : SUPPORT DE COURS IUT de Nice – Département INFORMATIQUE.
- [9] : http://www.iai-france.org/projets/batinterop/RT_Nicolle_0210_c.pdf.
- [9] : Jérôme_LAFOUSSE_javaEE_guide_de_developpement (BOOK_zz.org).
- [10] : Document interne a l'ENIEM.
- [11] : Document interne a l'unité prestation technique.
- [12] : Document interne au centre de médecine travail.
- [13] : Roques Pascal et Lai Michel « UML 2.0 par la pratique, UML ; la notion unifiée de la Modélisation objet.» Edition 2004.
- [13] : www.wikipedia.com.fr
- [14] : www.javafr.com
- [15] : <http://cours-pdf.com/cours-informatique/guide-pratique-de-netbeans.html>]]
- [16] : SUPPORT DE COURS IUT de Nice – Département INFORMATIQUE.
- [17] : <http://www-igm.univ-mlv.fr/~dr/XPOSE2004/msamson>
- [18] : [www.objj.com /formation.../tutoriel-hibernate.premiere-application.html](http://www.objj.com/formation.../tutoriel-hibernate.premiere-application.html).
- [18] : Richard monson-Haefel 2002.
- [19] : <http://book.google.com/books?isbn=284177>

Remerciements

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

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

Nos plus vifs remerciements vont à tout le personnel de l'Entreprise ENIEM Qui nous ont généreusement aidés durant notre stage.

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

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

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

Introduction

Actuellement, le monde connaît une avance technologique considérable dans tous les secteurs et cela grâce à l'informatique qui est une science qui étudie les techniques du traitement automatique de l'information. Elle joue un rôle **important dans le développement de travail**.

Avant l'invention de l'ordinateur, on enregistrait toutes les informations manuellement sur des supports en papier ce qui engendrait beaucoup de problèmes tel que la perte de temps considérable dans la recherche de ces informations ou la dégradation de ces dernières...etc.

Ainsi, jusqu'à présent, l'ordinateur reste le moyen le plus sûr pour le traitement et la sauvegarde de l'information. Cette invention a permis d'informatiser les systèmes de données des entreprises, ce qui est la partie essentielle dans leur développement aujourd'hui.

Les centres médicaux font partie intégrante des établissements que l'informatique pourra beaucoup aider. Jusqu'à ce jour, la manière de gérer manuellement est encore dominante, d'où la nécessité d'introduire l'informatique dans les gestions des centres médicaux.

L'exercice de la profession médicale ne se résume pas à faire un diagnostic, donner des conseils et proposer des traitements à son patients .

Après avoir effectué un stage de trois mois au sein de l'Enterprise national des electro-ménager « ENIEM », nous avons en un premier temps fait une étude sur le suivi informatisé des données médicales des patients de cette dernière ; étude qui nous a permis de concevoir un logiciel de gestion pour le CMT.

Dans ce présent document, après avoir présenté le centre médical de l'entreprise « ENIEM », nous donnerons un aperçu sur l'architecture javaEE le langage de programmation utilisé, après on entame le chapitre conception la ou on parle sur les outils utilisé et vers la fin on fini par la réalisation de notre logiciel qui est fiable pour le CMT de l'ENIEM.

I. 1. Introduction

Java Entreprise Edition est destiné aux gros systèmes d'entreprises. Les librairies utilisées fonctionnent difficilement sur un simple PC et requièrent une puissance beaucoup plus importante (notamment au niveau de la mémoire).

Java Entreprise Edition est apparue à la fin des années 90. Cette évolution apporte au langage Java une plateforme logicielle robuste et complète pour le développement. La plateforme Java EE a souvent été remise en cause, mal utilisée et mal comprise. Des outils Open Source sont venus la concurrencer. Ces remarques et la concurrence ont permis à Sun (le concepteur de Java) d'améliorer son produit et d'éditer des versions de plus en plus abouties. Java EE ne remplace en aucun cas J2SE.

La plateforme J2SE offre des outils de développement d'applications client/serveur, applications graphiques fenêtrées et Applets.

La plateforme J2SE est composée des éléments suivants :

- **La machine virtuelle Java (JVM)** : permet d'exécuter des applications Java. Elle constitue une passerelle et permet une portabilité entre les architectures (Windows, Linux, Mac...).
- **La bibliothèque de classes Java** : un ensemble de composants logiciels prêt à l'emploi.
- **Les outils de développement** : le compilateur javac , un interpréteur Java nommé java, le générateur de documentation javadoc, la console de supervision Jconsole... La plateforme Java EE est une extension de la Plateforme J2SE. Elle permet un développement d'applications qui vont s'exécuter sur un serveur d'applications.

Dans ce chapitre nous allons présenter l'essentiel des connaissances permettant de développer des applications web en s'appuyant sur JavaEE.

I.2. Java EE**I.2.1. Définition**

Java entreprise Edition (JavaEE) est une spécification pour le langage de programmation Java de Sun destinée aux applications d'entreprise. J2EE offre une plateforme de développement et déploiement en langage Java pour les applications distribuées à plusieurs niveaux. On parle généralement de « plate-forme J2EE » pour désigner l'ensemble constitué des services API (Application Programmable Interface) offerts et de l'infrastructure d'exécution.

I.2.2. Le serveur d'application [01]

Ils existent plusieurs définitions d'un serveur d'application parmi lesquelles on cite :

❖ Généralement, un serveur d'application est un logiciel d'infrastructure offrant un contexte d'exécution pour des domaines des applications WEB, dans le sens strict où les composants hébergés par le serveur d'application ne sont pas de simples procédures ou scripts mais de réels composants logiciels conformes (identique) à un modèle de composants (EJB, COM, Fractal,...etc.).

Les clients des serveurs d'applications sont des programmes autonomes (stand-alone application), des applets ou d'autres composants.

La structure en couches des différents composants mis à disposition par le serveur d'application permet une prise en compte des besoins métier et des interactions avec les utilisateurs et connexions avec les bases de données.

❖ Les serveurs d'applications peuvent être définis comme suit : Ils sont des logiciels occupant la couche centrale dans une architecture multi couches, qu'elle soit classique 3-tiers (postes client, serveur de données, serveur d'application) ou étendue (n-tiers) lorsqu'elle intègre des serveurs d'acquisitions (données de terrain, données de traitement, de back-office, etc.)

Donc généralement, un serveur d'application est l'environnement d'exécution des applications côté serveurs. Il prend en charge l'ensemble des fonctionnalités qui permettent à plusieurs clients d'utiliser une même application.

Les serveurs d'application se sont développés depuis la création de J2EE. On peut distinguer principalement 2 grandes catégories de serveurs :

En open source: Evolue grâce à la communauté.

A. TOMACT:

C'est un serveur d'application intégrant le serveur web APACHE. Il implémente les technologies suivantes : Servlet, JDBC, JNDI.

B. Jonas

C. JBOSS

D. GLASS FISH

En propriétaire: Evolue selon l'éditeur.

A. Web sphere d'ibm.

B. Web object d'apple

C. Oracle application server

I.3. Les architectures Web

I.3.1. Les types d'architectures

Dans les applications Web, la communication entre le client et le serveur est réalisée selon le protocole TCP/IP qui est chargé du routage des données. Le transit des informations s'effectue selon le protocole HTTP pour le Web, les données sont alors transmises entre le client et le serveur via TCP/IP. On distingue alors deux types de clients :

A- Le client léger [2]

il est aussi appelé client Web car le module d'exécution est alors un navigateur. Les applications clientes sont composées de pages HTML/XHTML voire DHTML avec l'utilisation du langage client JavaScript.

B- Le client lourd [2]

Il s'agit d'une application composée d'une interface graphique évoluée ou en mode console.

Dans l'idéal, les clients lourds communiquant ne contiennent que la logique présentation (affichage des données). Tous les traitements sont délégués à des composants métier distants.

I.3.1.1. L'architecture 2tiers [2]

Est composée de deux éléments, un client et un serveur. Cette architecture physique simple peut être représentée de cette façon :

Cette architecture peut aussi être représentée avec un serveur de base de données (SGBD), Dans ce type d'architecture, le client assume les tâches de présentation et communique uniquement avec le serveur d'applications. Le client est dit "lourd". Ce type d'architecture peut être développé très rapidement en fonction de la complexité du projet.

I.3.1.2. L'architecture 3tiers [2]

Le client est constitué d'un simple navigateur Internet et communique avec le serveur. Cette architecture est composée de trois éléments ou trois couches. La couche présentation ou affichage est le client "léger" dans la mesure où il ne fait aucun traitement. La couche fonctionnelle ou métier est en général un serveur Web. Et enfin, la couche de données est liée au serveur de bases de données (SGBD).

a- **La couche présentation** (de premier niveau) souvent appelée IHM (*Interface Homme Machine*) correspond à la partie visible et interactive. Cette partie est réalisée pour le Web en HTML en général avec JavaScript, Flash...

b- **La couche métier** (de second niveau) correspond à la partie fonctionnelle de l'application. Les opérations à réaliser, les fonctions d'accès aux données et les traitements sont mis à la disposition des utilisateurs et invoqués par leurs requêtes. Pour fournir ces services, elle s'appuie parfois sur la couche accès aux données et en retour renvoie à la couche présentation les résultats qu'elle a calculés.

c- **La dernière couche** (La couche de données) gère l'accès aux données du système. Ces données peuvent être stockées sur le même système (fichiers, fichiers XML, base de données, images...) ou sur d'autres systèmes.

L'accès aux données est transparent pour la couche métier et correspond uniquement à la préoccupation de la couche accès aux données.

D'une manière générale cette abstraction **améliore la maintenance du système**. Parmi les avantages de cette architecture, la flexibilité de l'ensemble peut être citée. La partie client est composée uniquement d'affichage (pas de programmation, de requêtes SQL...). De fait, des modifications peuvent être réalisées au niveau du SGBD sans que cela apporte un impact sur la couche client. De même, par la suite toute nouvelle technologie peut être introduite sans tout remettre en question. Du point de vue développement, la séparation entre le client, le serveur et le SGBD permet une spécialisation des développeurs et une meilleure répartition des tâches et fonctions (développeur de modèle/designer, programmeur, administrateur de bases de données...).

Le gros inconvénient de ce modèle (et le principal), est l'expertise qu'il est nécessaire d'avoir et qui est assez longue à obtenir pour bien maîtriser chaque tiers et interconnexions. Les coûts de développement d'une architecture 3tiers sont plus élevés que pour du 2tiers.

I.3.2. Les types d'architectures en Java EE [3]

Dans ce cas, l'application cliente peut être développée avec des composants graphiques (Swing par exemple) et faire appel à des règles métier EJB qui accèdent à une base de données.

Il est aussi possible de trouver des applications clientes avec JSP, EJB et base de données. Le client est un navigateur Web. Les pages JSP accèdent aux règles métier et construisent le contenu HTML fourni au navigateur.

Un dérivé de l'architecture précédente est une Applet cliente avec JSP et base de données.

Dans ce cas, le client est encore un navigateur mais une Applet permet d'obtenir une interface utilisateur plus complexe (mais aussi plus lourde) et plus interactive. Cette Applet peut accéder au contenu produit par des JSP. Celles ci obtiennent des données nécessaires grâce à JDBC.

Le dernier type d'architecture repose sur l'intégration de services Web. Une application cliente accède aux données grâce à un service Web programmé en Java.

Les architectures proposées par Java EE reposent sur le découpage des applications en plusieurs tiers aux responsabilités clairement séparées.

Les programmeurs vont développer des composants qui seront hébergés par un serveur d'application Java EE (Tomcat, JBoss...). Les applications distribuées (réalisées sous forme de composants distincts) permettent de diviser le logiciel en plusieurs couches appelées tiers (chaque couche représente un tier). Le modèle le plus courant étant l'architecture 3tiers/ntiers.

Cette division facilite la maintenance et l'adaptabilité du produit.

I.4. Le modèle MVC :

Le Model-View-Controller (MVC) est un modèle de conception logicielle très répandu et fort utile. il est aujourd'hui fortement recommandé dans l'univers JavaEE. Néanmoins il faut retenir que c'est un modèle de conception, et il est donc indépendant du langage de programmation.

Ce modèle est un concept d'architecture qui propose une séparation en trois entités des données, des traitements et de l'interface :

- le modèle
- la vue
- le contrôleur

Chacun de ces trois composants a un rôle bien défini.

- ⇒ Le **modèle** représente les données et les règles métiers. C'est dans ce composant que s'effectuent les traitements liés au cœur du métier. Les données peuvent être liées à une base de données, des EJBs, des services Web, ... Il est important de noter que les données sont indépendantes de la présentation. En d'autres termes, le modèle ne réalise aucune mise en forme. Ces données pourront être affichées par plusieurs vues. Du coup le code du modèle est factorisé : il est écrit une seule et unique fois puis réutilisé par chaque vue.
- ⇒ La **vue** correspond à l'IHM. Elle présente les données et interagit avec l'utilisateur. Dans le cadre des applications Web, il s'agit d'une interface HTML, mais n'importe quel composant graphique peut jouer ce rôle.
- ⇒ Le **contrôleur**, quant à lui, se charge d'intercepter les requêtes de l'utilisateur, d'appeler le modèle puis de rediriger vers la vue adéquate. Il ne doit faire aucun traitement. Il ne fait que de l'interception et de la redirection.

I.4.1. Le modèle MVC type1

Dans ce modèle, chaque requête est traitée par un contrôleur sous la forme d'une servlet. Celle-ci traite la requête, fait appel aux éléments du model si nécessaire et redirige la requête vers une JSP qui se charge de créer la réponse à l'utilisateur.

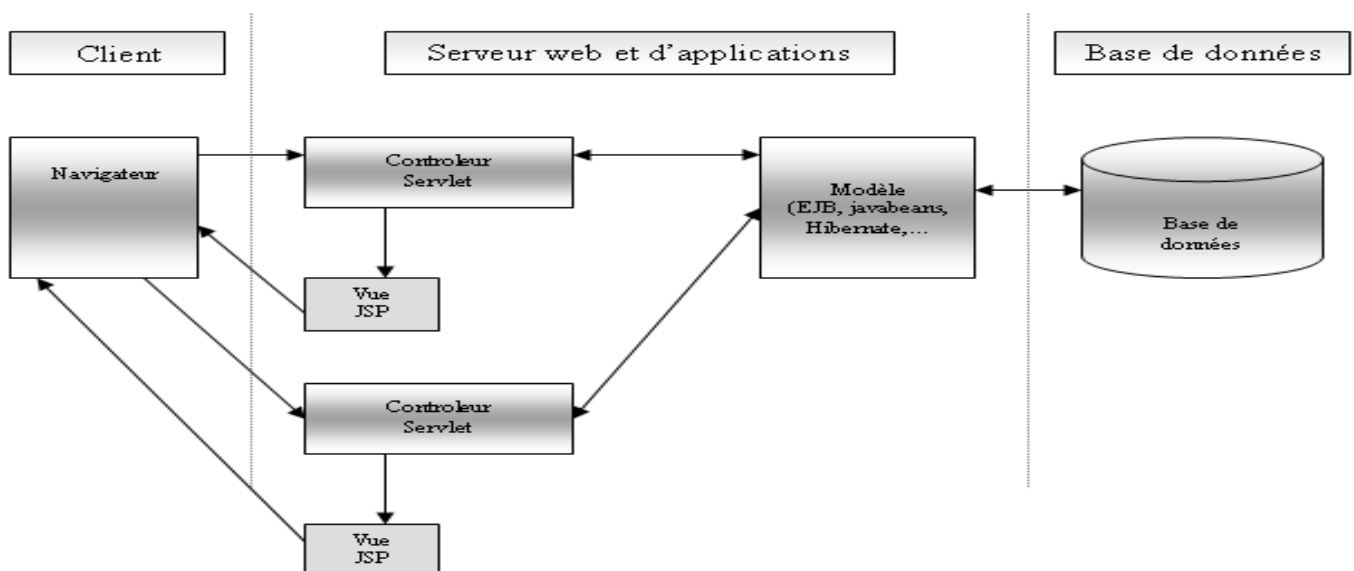


Figure1.1. : Schéma modèle MVC de type1.

L'inconvénient est donc une multiplication du nombre de servlets nécessaires à l'application : l'implémentation de plusieurs servlets nécessite beaucoup de code à produire d'autant que chaque servlet doit être déclarée dans le fichier web.xml.

I.4.2. Le modèle MVC de type2

Le principal défaut du modèle MVC est le nombre de servlets à développer pour une application. Pour simplifier les choses, le modèle MVC model 2 ou MVC2 de Sun propose de n'utiliser qu'une seule et unique servlet comme contrôleur. Cette servlet se charge d'assurer le workflow des traitements en fonction des requêtes http reçues.

Le modèle MVC 2 est donc une évolution du modèle 1 : une unique servlet fait office de FFF la forme d'un fichier au format XML.

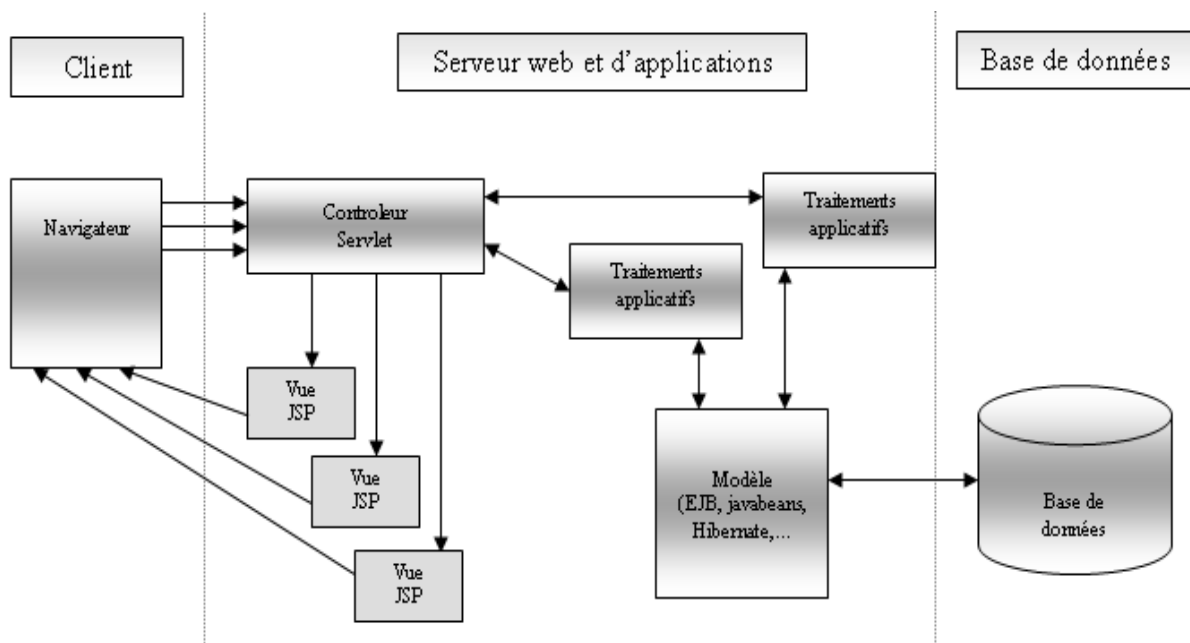


Figure I.2. : Schéma le modèle MVC de type2.

Le modèle MVCII conserve les principes du modèle MVC, mais il divise le contrôleur en deux parties en imposant un point d'entrée unique à toute application (première partie du contrôleur) qui déterminera à chaque requête reçue les traitements applicatifs à invoquer dynamiquement (seconde partie du contrôleur).

Une application web implémentant le modèle MVC de type II utilise une servlet comme contrôleur traitant les requêtes. En fonction de celles-ci, elle appelle les traitements dédiés généralement encapsulés dans une classe.

Dans ce modèle, le cycle de vie d'une requête est le suivant :

1. Le client envoie une requête à l'application, requête est prise en charge par la servlet faisant office de contrôleur.
2. La servlet analyse la requête et appelle la classe dédiée contenant les traitements
3. Cette classe exécute les traitements nécessaires en fonction de la requête, notamment, en faisant appel aux objets métiers.
4. En fonction du résultat, la servlet redirige la requête vers la page JSP
5. La JSP génère la réponse qui est renvoyée au client.

I.5. Framework [4]

I.5.1. Définition

Le terme framework est fréquemment utilisé dans des contextes différents mais il peut être traduit par cadre de développement.

Les frameworks se présentent sous diverses formes, qui peuvent inclure tout ou une partie des éléments suivants :

- 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 une partie d'un squelette d'application
- des recommandations sur la mise en oeuvre et des exemples d'utilisation
- des normes de développement
- des outils facilitant la mise en œuvre.

I.5.2. Objectif d'un framework

L'objectif d'un framework est de faciliter la mise en œuvre des fonctionnalités de son domaine d'activité. Il doit permettre au développeur de se concentrer sur les tâches spécifiques de l'application à développer plutôt que sur les tâches techniques récurrentes telles que :

- l'architecture de base de l'application.
- l'accès aux données.
- l'internationalisation.
- la journalisation des événements (logging).
- la sécurité (authentification et gestion des rôles).
- le paramétrage de l'application.

La mise en œuvre d'un framework permet notamment :

- de capitaliser le savoir-faire sans "réinventer la roue"
- d'accroître la productivité des développeurs une fois le framework pris en main
- d'homogénéiser les développements des applications en assurant la réutilisation de composants fiables
- donc de faciliter la maintenance notamment évolutive des applications.

I.5.3. Composants d'un framework [5]

I.5.3.1. Bibliothèque

Un *Framework* peut être ou peut contenir une ou des bibliothèques. Une bibliothèque, en informatique, est un ensemble d'algorithmes qui peut être répétitif. Donc les bibliothèques sont utilisées pour que nous n'ayons pas à les réécrire à chaque nouveau projet ou plusieurs fois dans le même projet. Une bibliothèque peut aussi être un ensemble d'algorithmes offert par des développeurs pour d'autres développeurs pour leur simplifier la vie. Par exemple, la bibliothèque *Math* qui contient une grande quantité de formules mathématiques.

I.5.3.2. Structure

En plus de contenir ou non des bibliothèques, un Framework peut avoir une structure prédéfinie pour concevoir des programmes. Par exemple, la fenêtre de l'application et ses composantes déjà conçues et pouvant être modifiées selon les goûts et les besoins des programmeurs. La structure peut aussi être une architecture du code permettant une utilisation facile de conception complexe.

I.5.4. Framework HibernateSpring

I.5.4.1. Définition 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.

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.

Hibernate est très populaire notamment à cause de ses bonnes performances et de son ouverture à de nombreuses bases de données.

Les bases de données supportées sont les principales du marché : DB2, Oracle, MySQL, PostgreSQL, Sybase, SQL Server, Sap DB, Interbase, ...

I.5.4.2. Architecture d'Hibernate [6]

Hibernate a besoin de plusieurs éléments pour fonctionner : une classe de type `javabeen` 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.

I.5.4.3. Avantages et inconvénients d'Hibernate [7]

Les avantages

- Hibernate génère le code SQL nécessaire, ce qui rend l'application plus portable(s'adapte à la base de données).
- La persistance est transparente. Nous pouvons faire nos classes métiers des classes persistantes sans ajout de code.
- La récupération de données est optimisée. On peut interroger la base de données de plusieurs façons (requête SQL, langage HQL...).
- Portabilité du code en cas de changement de base de données.

- Propose une API performante et robuste
- mature et dispose d'un support fiable et d'une documentation abondante Supporte la gestion des transactions
- S'intègre facilement à Eclipse et Spring.

Les inconvénients

- Il est dure de faire des requête complexe avec HQL (ex : sous requêtes).
- Etant une technologie jeune, il reste des problèmes à résoudre (ex : les fichiers de mapping ne sont pas toujours bien formés)
- Nécessite un module pour la connectivité aux bases Oracle
- Possibilité de perte.

I.5.5. Framework Spring [8]

I.5.5.1. Définition Spring

C'est un framework qui permet de « remplacer » la lourdeur des serveurs d'application lourds. En effet, on parle de « conteneur léger ». Il prend en charge la création et la mise en relation d'objets par l'intermédiaire d'un fichier de configuration décrivant l'ensemble de ces relations. L'un des avantages principal est qu'il n'impose pas d'hériter ou d'implémenter une quelconque interface ou classe contrairement aux EJB.

I.6. Enterprise JavaBeans (EJB) [9]

I.6.1. Définition Enterprise JavaBeans (EJB)

Est une architecture de composants logiciels côté serveur pour la plateforme de développement [Java EE](#).

I.6.2. les 3 types d'EJB

Les composants côté serveur des Entreprise JavaBeans sont de trois types: les **beans entité**, les **beans session** et les **beans orientés message**. Les beans session et les beans entité sont des composants coté serveur basés sur **RMI** (Remote Method Invocation). On y accède à l'aide des protocoles d'objets distribués. Les beans orientés message disponibles depuis la version 2.0 sont des composants côté serveur asynchrones qui répondent aux messages **JMS** (Java Message Service) asynchrones.

- **Les beans entité** modélisent des concepts métier pouvant être désignés par des noms. Par exemple un bean entité peut représenter un client, un objet d'un stock. Autrement dit, les bean entité correspondent à des objets réel. Ces objets sont dans la plupart des cas des enregistrements persistants stockés dans une base de données.
- **Les beans session** sont une extension de l'application client et ils sont responsable de la gestion des processus ou des tâches.

- **Les beans orientés message** des EJB 2.0 sont responsables de la coordinations des tâches impliquant d'autres bean session et entité. La différence entre les beans orientés message et les beans entité réside dans la façon de d'y accéder. Un Bean session fournit une interface distante afin de définir les méthodes que l'on peut invoquer, alors que les Beans orientés message attend des messages asynchrones spécifiques auxquels il répond.

I.7. Conclusion

Au cours de ce chapitre, nous avons abordé les notions de base de l'architecture J2EE et le Model MVC. Ensuite on a parlé sur l'objectif d'un Framework où on a choisir Hibernate avec une justification de notre choix.

A. Partie I : L'organisme d'accueil**II.1 Introduction**

L'ENIEM 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 exercer son activité en raison de la subvention et le soutien de l'ETAT.

II.2. Présentation générale de l'entreprise ENIEM**II.2.1. Historique [10]**

L'ENIEM est une entreprise publique de droit algérien constitué le 02 janvier 1983 par décret n° 83-19-1983 suite à la restructuration de la société nationale de fabrication et du montage du matériel électrique et électronique (SONELEC), elle est transformée juridiquement en société par action le 08/octobre 1989. Son capital social est de 2 957 500 000DA ou de 31 462 766 euros qui sont détenus en totalité (100% par la société de gestion des participations industrielles électrodomestique (SGP. INDELEC).

Face aux enjeux économiques, l'ENIEM a mis en oeuvre un système de management de qualité basé sur l'amélioration continue de la qualité du produit pour accroître la satisfaction client.

EN 2013, elle s'est vue décernée le prix algérien de la qualité, récompensant tous les efforts qu'elle a entrepris pour la satisfaction des clients.

II.2.2. Les activités principales de l'ENIEM [10]

L'ENIEM est 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.

II.2.3. Les missions principales de l'ENIEM [10]

L'ENIEM a progressé dans un environnement où la dimension financière n'est pas considérée et la finalité de la réalisation de bénéfice n'a jamais été une priorité. Les objectifs sociaux ont tout de même dominé et la loi avantageait les valeurs sociales (augmentation des salaires et le maintien des emplois). 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.

II.2.4. Les objectifs de l'ENIEM [10]

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 :

- ✓ 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.2.5. Activités de l'ENIEM [10]

L'activité de l'ENIEM est concentrée sur la fabrication de réfrigérateurs, cuisinières, et climatiseurs. Cette activité sera assurée par Cinq unités de productions :

a) Unité froid (UF) : 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.

b) Unité cuisson (U CUI) : Elle est chargée de la production et développement des produits de cuisson, ses activités sont :

- ✓ La transformation des tôles.
- ✓ Assemblage.

c) Unité climatisation (U CL) : Elle est chargée de la production et du développement des produits de climatisation.

d) Unité commerciale (U C) : 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.

e) Unité prestation technique (UPT) : Qui est chargée de fournir des prestations techniques.

- Notre stage pratique se dérouler au sien de cette unité.

II.3. Organisation générale de l'ENIEM [10]

L'Organigramme de l'ENIEM

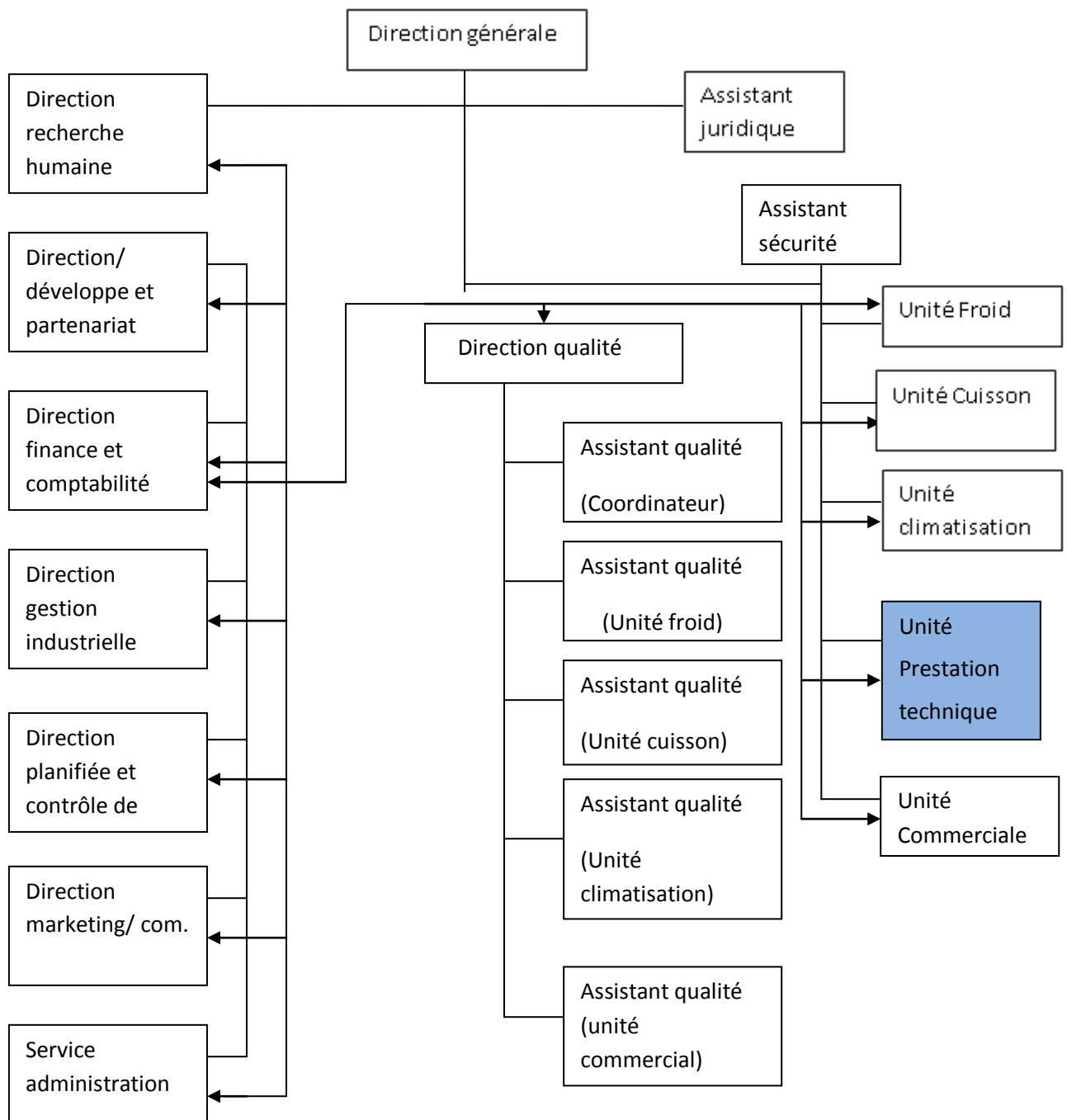


Figure II.1 : organigramme générale de l'ENIEM.

Source : document interne à l'ENIEM.

II.3.1. L'Organigramme de l'unité prestation technique

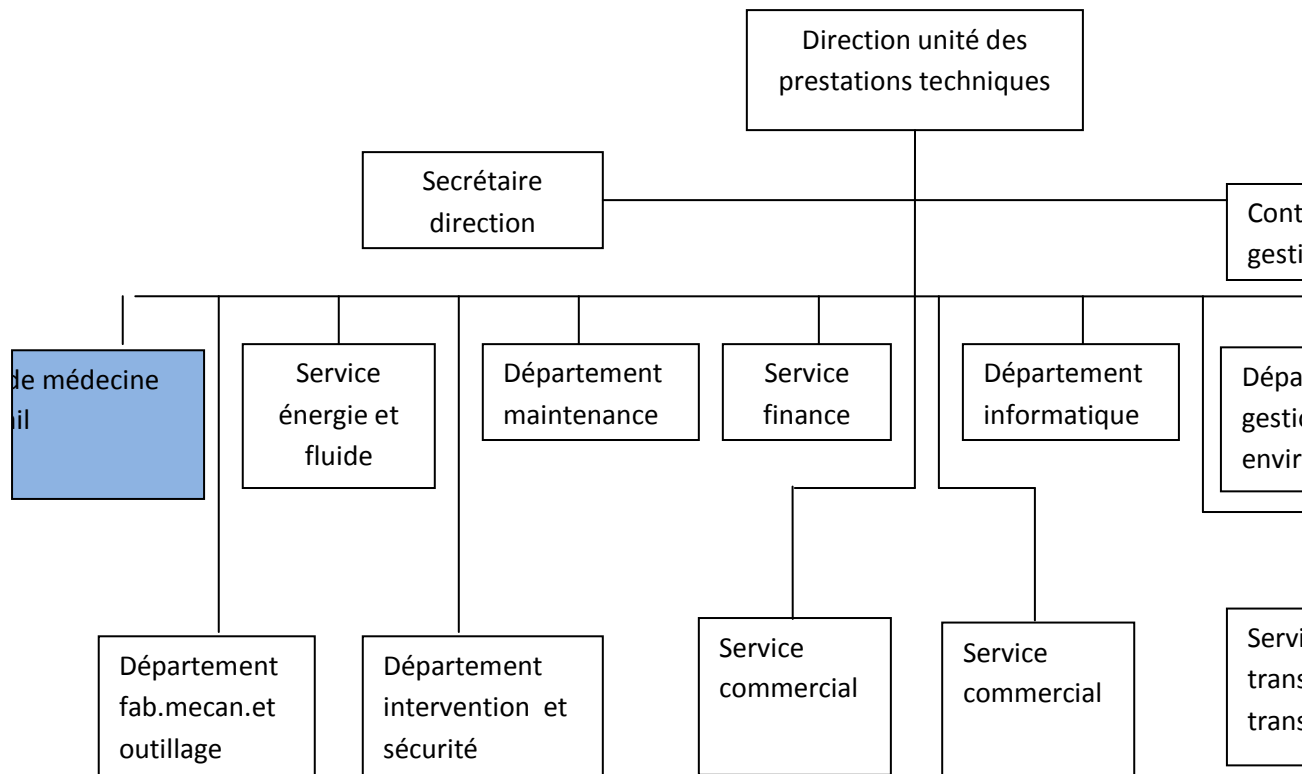


Figure II.2 : organigramme de l'unité prestation technique.

Source : centre médical de l'ENIEM.

Champ d'étude :

Centre de médecine de travail

II.3.1.1. L'organigramme de centre médecine [11]

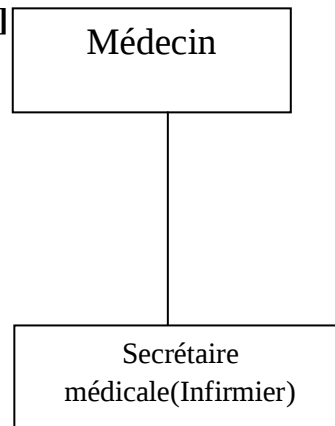


Figure II.3 : Organigramme du centre médical.

B. Partie II : Etude de poste**II.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.

II.1.1. Les postes de travail étudiés [12]

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

Le poste de communication et traitement : occupé par le Secrétaire (infirmière) (qui est chargée d'assurer un bon fonctionnement pour le 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.

II.1.2. Fiche d'analyse des postes :**II.1.2.1. Poste de travail N°01 : Médecin [12]**

Fiche d'étude du poste N°01
Désignation : Médecin
Effectif : 02
Lieu d'implantation : Le centre médical
Responsabilités : -Diriger le centre médical -Superviser et veiller au bon fonctionnement du centre médical
Tâches à accomplir : -Rassurer et soigner les patients. -Le suivi et la mise à jour des dossiers des patients. -Délivrer les documents médicaux (l'ordonnances, l'arrêt de

travail, etc.)	
Document provenant à ce poste	
Désignation	Provenance
-Fiche de renseignement sur les patients	-Secrétaire
Documents diffusés par ce poste	
Désignation	Destination
-La feuille de soins -L'ordonnance -Compte rendu d'échographie	-Le patient -Le patient -Le patient
Fichiers utilisés par ce poste	
- dossier patient	

Tableau II.1. : Poste du médecin.

II.1.2.2. Poste de travail N°02 : infirmière (Secrétaire) [12]

Fiche d'étude du poste N°02	
Désignation : Infirmière	
Effectif : 01	
Lieu d'implantation : Le centre médical	
Responsabilités :	
- Assurer le bon fonctionnement du centre médical.	
Tâches à accomplir :	
-Accueillir les patients et organiser leur passage dans la salle de consultation.	
-Reconnaître une urgence et prévenir immédiatement le médecin.	
-Enregistrement manuel, codification, archivage et classement des dossiers médicaux.	
Document provenant à ce poste	
Aucun document	
Documents diffusés par ce poste	
Désignation	Provenance
Fiche de renseignements	-Médecin
Documents remplis par ce poste	
Désignation	Destination
Fiche de renseignements	-Médecin
Fichiers utilisés par ce poste	
Fichier des visites.	

Tableau II.2. : Poste de l'infirmière.

II.2. Diagramme des flux

II.2.1. Définition

L'acteur : c'est une unité active intervenant dans le fonctionnement du système opérant, pour l'échange de l'information avec les autres acteurs.

L'acteur peut être **interne ou externe**.

Le flux : Un flux d'information ou de données est un échange d'informations entre deux acteurs dans le cadre de système d'information concerné.

Le diagramme des flux est une représentation graphique des différents **acteurs** (Interne, externe) au domaine d'étude et les **flux** d'information échangés entre eux, cette représentation a pour but de délimiter le champ du projet.

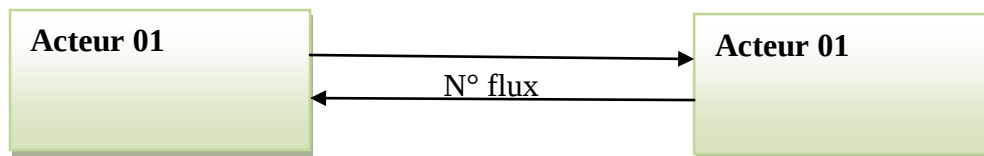
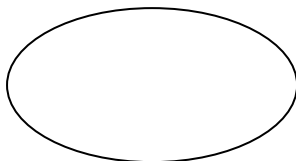


Figure II.4. : Le schéma du diagramme des flux.

II.2.2. Formalisme graphique :



Acteur interne au champ d'étude.



Acteur externe au champ d'étude.



Sens de circulation de l'information.

II.2.3. Table explicative des acteurs de domaine d'étude : Dans notre cas d'étude nous avons recensé les acteurs suivants :

⇒ **Acteurs internes**

-Médecin.

-Infirmière.

⇒ **Acteurs externes**

-Patient : toute personne représentant au centre pour des soins , une personne qui n'est pas en bonne santé.

-Hôpital : Établissement public ou privé ayant passé certaines conventions avec l'État et où peuvent être admis tous les malades pour y être traités.

II.2.4. Représentation du diagramme des flux existant [12]

Ce diagramme représente la circulation de l'information entre les acteurs internes et externes.

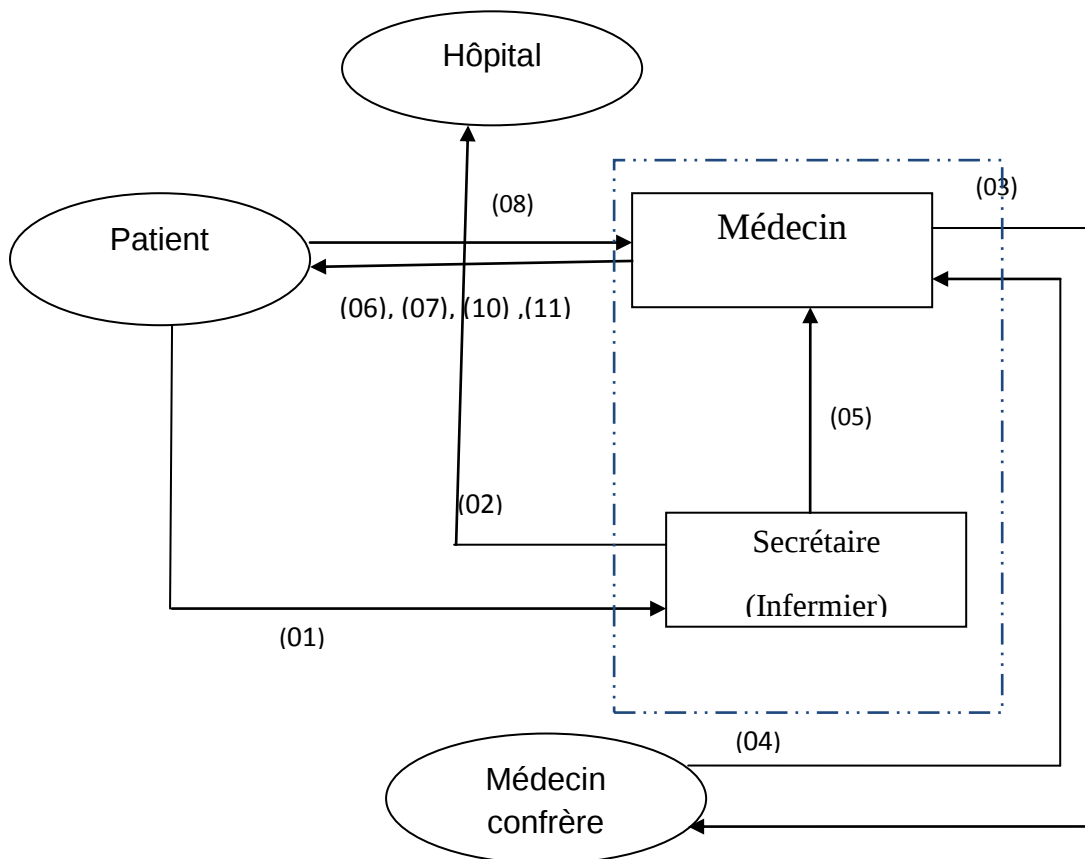


Figure II.5. : Diagramme des flux existant.

II.2.5. Description des flux

N° flux	Description
01	Demande de consultation .
02	Consultation prioritaire (cas d'urgence).
03	Lettre d'orientation.
04	Lettre d'orientation.
05	Fiche de renseignement + consultation prioritaire
06	Ordonnance
07	Demande d'analyse (laboratoire d'analyse)
08	Résultat d'analyse
09	Compte rendu d'échographie

Tableau II.3 : Description des flux.

II.3. Etude des documents et des registres [12]

II.3.1. Introduction

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 (2) 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.

II.3.2. Liste des documents utilisés

N° de document	Désignation	code	Nature
01	Fiche de renseignement	FR	Interne
02	Ordonnance	OrD	Externe
03	visite périodique, redéploiement, d'embouche.	VPRE	Interne
04	Visite spontanée	VS	Interne
05	Accident de travail.	AT	Interne
06	Demande d'examen	DE	Externe
07	fiche de vaccin	FV	Externe

Tableau II.4. : Liste des documents utilisés.

Les abréviations

A : Alphabétique

N : Numérique

AN : Alphanumérique

D : Date

II.3.3. Etude des documents utilisés [12]

II.3.3.1. Fiches d'analyse des documents

Fiche d'étude du document N° 01 : **Fiche de Renseignement**

Caractéristique du document			
Code : FR Désignation : fiche de renseignement Rôle : Sert à écrire les renseignements sur le patient Nature : interne Poste créateur : Infermière(Secrétaire) Poste destinataire : Médecin Nombre d'exemplaire : 01			
Description du document			
Rubrique	Type	Taille	
-Nom patient	A	25	JJ/MM/AAAA
-Prénom patient	A	25	
-Adresse patient	AN	45	
-N° tel patient	N	10	
-les renseignements	A	70	
-Date de naissance.	D	10	

Tableau II. 5. : Fiche de Renseignement.

Fiche d'étude du document N° 02 : **Ordonnance**

Caractéristique du document			
Code : Ord. Désignation : Ordonnance Rôle : Prescrire un traitement pour le patient. Nature : Externe. Poste créateur : Médecin. Poste destinataire : Patient. Nombre d'exemplaire : 01.			
Description du document			
Rubrique	Type	Taille	
Nom médecin	A	25	JJ/HH JJ/MM/AAAA
-Prénom médecin	A	25	
-Durée d'ouverture	AN	20	
-Date consultation	D	10	JJ/MM/AAAA
-Nom patient	A	25	
-Prénom patient	A	25	
-Date de naissance patient	D	10	

-Désignation médicament	A	20	
-Posologie médicament	A	10	
-Dosage médicament	A	10	

Tableau II. 6. : Fiche pour l'Ordonnance.

Fiche d'étude du document N° 03 : fiche pour : **visite périodique, redéploiement, d'embouche.**

Caractéristique du document			
Code : VPRE Désignation : visite périodique, redéploiement, d'embouche. Rôle : consulter un patient Nature : interne Poste créateur : Médecin Nombre d'exemplaire : 01			
Description du document			
Rubrique	Type	Taille	
-Nom médecin	A	25	JJ/MM/AAAA
-Prénom médecin	A	25	
Date de consultation	D	10	
Commentaire	A	15	
Orl	A	08	
Locomoteur	A	10	
Respiratoire	A	09	
Cardio	A	15	
Digestif	A	15	

Tableau II.7. : Fiche pour les visites périodiques, redéploiement, d'embouche.

Fiche d'étude du document N° 04 : fiche pour **visite spontanée.**

Caractéristique du document			
Code : VS Désignation : visite spontanée. Rôle : consulter un patient Nature : interne Poste créateur : Médecin Nombre d'exemplaire : 01			
Description du document			
Rubrique	Type	Taille	
-Nom médecin	A	25	JJ/MM/AAAA
-Prénom médecin	A	25	
Date de consultation	D	10	

Motif	AN	15	
Diagnostic	AN		

Tableau II.8. : Fiche pour la visite spontanée.

Fiche d'étude du document N° 05 : fiche pour **Accident de travail**.

Caractéristique du document			
Code : AT Désignation : Accident de travail. Rôle : consulter un patient Nature : interne Poste créateur : Médecin Nombre d'exemplaire : 01			
Description du document			
Rubrique	Type	Taille	
-Nom médecin	A	25	JJ/MM/AAAA
-Prénom médecin	A	25	
Date de consultation	D	10	
Siege lésions	AN	30	
Nature	AN	30	

Tableau II.9. : Fiche pour la visite Accident de travail.

Fiche d'étude du document N° 06 : **Demande d'examen**

Caractéristique du document			
Code : DE Désignation : Demande d'examen Rôle : Nom des examens à faire Nature : Externe Poste créateur : Médecin Poste destinataire : laboratoire d'examen Nombre d'exemplaire : 01			
Description du document			
Rubrique	Type	Taille	
-Nom médecin	A	25	JJ/MM/AAAA
-Prénom médecin	A	25	
-Date consultation	D	10	
-Nom patient	A	25	
-Prénom patient	A	25	
-Date de naissance	D	10	
- type d'examen	AN	25	

Tableau II.10. : Fiche pour les Examens.

Fiche d'étude du document N° 07 : fiche de vaccin

Caractéristique du document			
Code : FV Désignation : vaccin Rôle : Aider le médecin lors de la consultation Nature : Externe Poste créateur : Médecin Poste destinataire : laboratoire d'analyse. Nombre d'exemplaire : 01			
Description du document			
Rubrique	Type	Taille	
-Nom patient	A	25	JJ/MM/AAAA
-Prénom patient	A	25	
-date de naissance	D	10	
-Type vaccin	A	10	
-Commentaire	A	40	

Tableau II.11. : Fiche pour les Vaccins.

II.4. 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. Dans ce but, nous avons pensé à développer une application de gestion d'un centre médical pour faire face à de problème comme :

- ✓ La difficulté de retrouver le dossier au moment voulu ce qui engendre un retard dans le traitement de l'information: nous avons constaté que le temps de recherche d'un Dossier médical est très important (allant de 10 à 15 minutes), donc si les informations ne peuvent être repérer à temps, la prise en charge du malade devient difficile.
- ✓ Les besoins de communication rapide de l'information et l'utilisation des nouvelles technologies, surtout avec le développement complémentaires des bases de données.
- ✓ Nombre important des archives qui engendre une difficulté de stockage.
- ✓ Travaux manuels élevés, lourds et pénibles ce qui provoque un risque d'erreur et d'oubli ou chevauchement des 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.

II.5. Objectifs de cette étude

De ce fait la problématique se pose :

Comment assurer au personnel médical un accès rapide aux dossiers malades tout en assurant la sécurité et l'intégrité des informations ainsi que la préservation des dossiers à long terme ?

L'informatique propose une solution d'archivage et de suivi des patients très claire et offre plusieurs moyens pour préserver l'information en toute sécurité par authentification et de la récupérer au cas de perte.

La solution proposée doit assurer les fonctions suivantes :

- Rapidité dans l'établissement des différents documents.
- Faciliter de la recherche et l'accès aux informations.
- Automatiser certaines tâches qui se traitent manuellement (Saisir les données personnelles et les détails sur les consultations ; la saisie des autres fiches médicales telle que l'ordonnance et les recherches effectuées sur les dossiers des patients, etc.)

Stockage des informations sur des supports magnétiques ce qui assurera leur sécurité.

II.6. Conclusion

L'étude de l'existant nous a permis d'avoir une idée bien précise sur le système existant 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.

III.1-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 démarche bien définie, en mettant en évidence tous les objectifs tracés Pour la bonne élaboration du projet souhaité.

Dans ce chapitre nous allons entamer le développement de notre application par une analyse qui mettra en évidence les différents acteurs intervenants dans le système ainsi que leurs besoins.

La phase conception, s'appuyant sur les résultats de cette analyse, donnera la modélisation des Objectifs à atteindre.

Des progrès énormes ont été consentis dans le but d'avoir une meilleure analyse et de rendre la conception plus complète.

L'approche objet, s'est avérée un modèle d'analyse et de conception très puissant et se trouve de plus en plus utilisée. Pour cela nous avons adopté la conception avec l'UML (Unified Modelling language) qui permet de bien représenter la dynamique d'une application par la série des diagrammes qu'il offre.

III.2-Rappel sur l'UML [9]

La description de la programmation par objets a fait ressortir l'étendue du travail conceptuel nécessaire : définition des classes, de leurs relations, des attributs et méthodes, etc. Pour réaliser une application, il ne convient pas de se lancer tête baissée dans l'écriture du code il faut d'abord organiser ses idées, les documenter, puis organiser la réalisation en définissant les modules et étapes de la réalisation. C'est cette démarche antérieure à l'écriture que l'on appelle modélisation son produit est un modèle.

UML (Unified Modelling Language) est la synthèse des différentes notations que l'on retrouve dans OMT, OOSE, Booch, qui recouvrent à elles trois plus de la moitié du marché des méthodes objets. UML que ses concepteurs ont voulu simple, est plus intuitive, plus expressive, plus cohérent, plus homogène que les autres méthodes. Elle est en particulier conçue pour être lisible sur des supports courants et variés, comme le papier, les écrans d'ordinateur, les tableaux blancs, etc.

UML propose 9 diagrammes de modélisation, réparties sur trois axes du niveau conceptuel

- Fonctionnel.
- Statique.
- Dynamique.

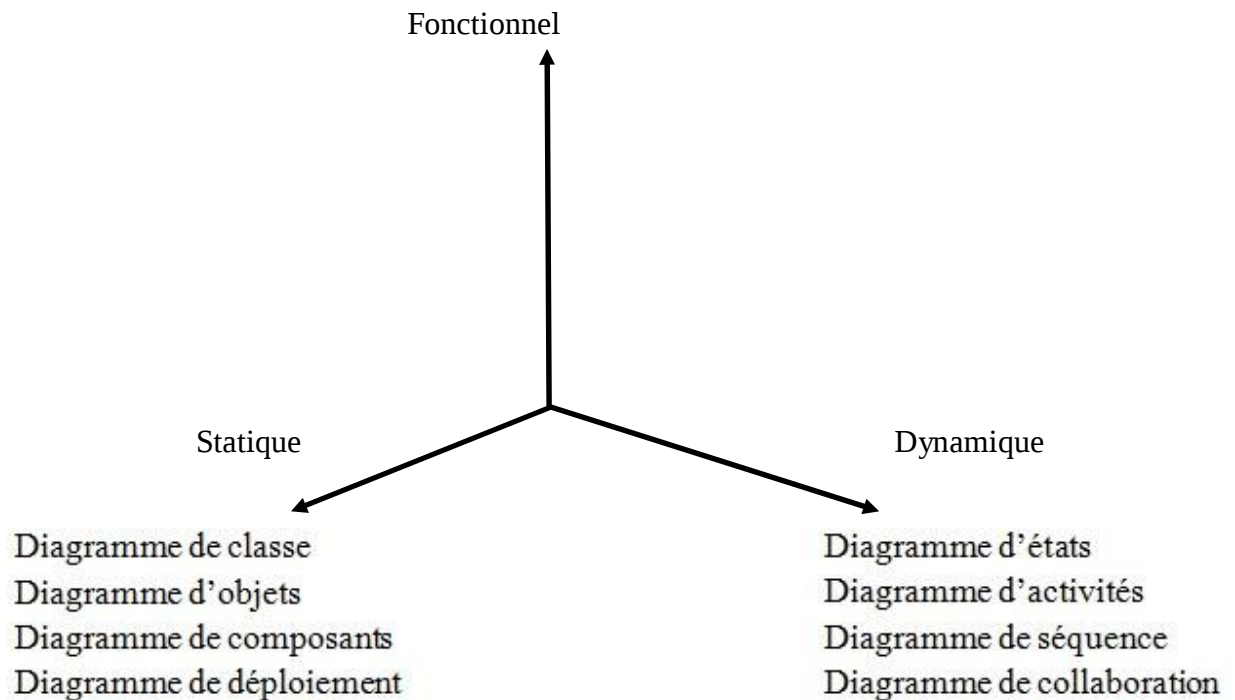


Figure III.1 : Composantes de la modélisation UML.

III.3-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 médicale de l'ENIEM. Pour se faire, il est nécessaire de déterminer globalement ce qui se trouve dans le champ de l'application.

III.3.1- Identification des acteurs :

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

Un acteur présente un rôle que peut jouer l'utilisateur dans le système. L'acteur est associé à Un cas d'utilisation, c'est-à-dire qu'il peut interagir avec lui et participer à son scénario, il est Représenté par un personnage stylisé.

Les acteurs de notre système sont :

- ✓ Médecin : c'est les médecins spécialistes, et généralistes qui jouent le même rôle.
- ✓ Infirmier (secrétaire médicale) : c'est les surveillants médicaux chefs, Surveillants médicaux, aides soignant Qui jouent le même rôle.
- ✓ Administrateur : la personne qui rentre dans la gestion du personnel médical et paramédical (médecin, Infirmier, secrétaire médicale).

III.3.2- Les fonctionnalités du système :

Pour la structuration des informations des dossiers médicaux, nous avons pu dégager les Différentes parties que devra contenir notre dossier médical :

❖ Partie identification :

Cette partie devra contenir des champs pour l'identification des patients. Il devra y avoir le champ «Nom et Prénom» qui sera l'identifiant du patient dans notre application et qui le suivra durant toute sa vie. Les champs nom, prénom, date de naissance, adresse et sexe peuvent être considérés, comme des champs obligatoires à remplir car ils apparaissent dans pratiquement tous les documents du dossier médical (fiche visite spontanée, fiche visite d'accident de travail...).

❖ Partie antécédent, Partie vaccin, Partie allergie:

Cette partie est remplie par le Medecin traitant, elle contiendra la liste des antécédents Personnels (Chirurgicaux et médicaux), la liste des vaccins et aussi l'allergie qu'a eue le patient ainsi que les antécédents familiaux.. Cette partie devra être mise à jour à chaque découverte d'un antécédent, allergie ou vaccin.

❖ Partie visite infirmière:

Cette partie est remplie par l'infirmier traitant, elle contiendra la liste des champs du visite. Cette partie devra être mise à jour.

❖ Partie contient toutes les visites:

Cette partie est remplie par le Médecin traitant, elle contiendra la liste des champs de visite d'embauche, visite périodique, visite redéploiement . Elle donne la possibilité de consulter tous les visites du patient. Cette partie devra être mise à jour à chaque nouvelle visite.

III.3.3- Identification des cas d'utilisation :

Un cas d'utilisation (use case) modélise une interaction entre le système informatique à Développer et un utilisateur ou acteur interagissant avec le système.

Plus précisément, un cas d'utilisation décrit une séquence d'actions réalisées par le système qui produit un résultat observable pour un acteur.

Pour notre cas, nous avons pu dégager les cas d'utilisation suivants :

Acteur	Cas d'utilisation
Personnel paramédical (infirmière, secrétaire médicale)	- S'authentifier. - Rechercher un patient. - Consulter un dossier médical. - o Consulter les antécédents, vaccin , allergie. - Mise à jour un dossier médical.
Personnel médical (médecin)	- S'authentifier. - Rechercher un patient. - Consulter un dossier médical. - o Consulter les antécédents, vaccin , allergie. - Mise à jour un dossier médical. - o Ajouter une nouvelle consultation. - o Ajouter les antécédents, vaccin , allergie. - Editer les documents d'un patient. - o Editer la lettre d'orientation. - o Editer l'ordonnance. - o Editer la carte examen. - Gérer les patients.
Administrateur	- S'authentifier. - Gérer les utilisateurs. - o Ajouter un utilisateur. - o modifier un utilisateur. - o Supprimer un utilisateur.

Tableau III.1 : Structuration des cas d'utilisation globale.

Remarque :

L'opération « Supprimer un dossier médical » n'est pas autorisée pour aucun acteur.

III.3.4- Structuration des cas d'utilisation en packages :

Dans cette phase nous allons structurer les cas d'utilisation en Packages. 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 doivent représenter un ensemble fortement cohérent et sont généralement de même nature et de même niveau sémantique.

Dans notre cas on obtient le regroupement suivant :

Cas d'utilisation	Acteur	Package
S'authentifier	Personnel paramédical, personnel médical, Administrateur	Connexion
Rechercher un patient	Personnel paramédical, personnel médical	Recherche
Ajouter un nouveau Patient	Personnel medical	Gestion des Patients
Consulter les examen	Personnel paramédical, personnel médical	Consultation
Consulter les antécédents	Personnel paramédical, personnel médical	
Voir l'existence de l'allergie.	Personnel medical, Personnel paramédical	
Ajouter un vaccin	personnel médical	MAJ les données
Ajouter une nouvelle Consultation	Personnel medical	
Ajouter une Allergie	Personnel medical	
Ajouter un antécédent	Personnel medical	
Editer l'ordonnance	Personnel medical, Personnel paramédical.	Edition
Editer la lettre d'orientation	Personnel medical, Personnel paramédical	
Editer la carte examen	Personnel medical, Personnel paramédical	
Ajouter un utilisateur	Administrateur	Gestion des Utilisateurs
Modifier un utilisateur	Administrateur	
Supprimer un utilisateur	Administrateur	

Tableau III.2.: Structuration des cas d'utilisation en package.

III.3.5- Identification des scenarios

Un scénario 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 (qui se terminent d'une façon normale) ou d'erreurs (qui se terminent en échec).

Dans ce qui suit, on présente dans des tableaux, les différents scénarios associés à chaque acteur.

Tâche	Scénarios
T01 : S'authentifier	S01 : L'utilisateur demande la connexion. S02 : Le système lui affiche un formulaire d'authentification. S03 : L'utilisateur saisit son nom utilisateur et son mot de passe, puis il click sur « connexion ». S04 : Le système vérifie la validité des informations reçues et affiche la page d'accueil de l'espace demandé, sinon il retourne a la page d'accueil.
T02 : Consulter les différentes visites.	S05 : L'utilisateur demande la consultation d'une visite S06 : Le système lui affiche un formulaire de recherche du patient. S07 : L'utilisateur remplit le formulaire. S08: Le système vérifie la validité des informations reçues et autorise l'accès au dossier du patient recherché, sinon affiche un message d'erreur.
T03 : Consulter les examen	S09 : L'utilisateur demande la visualisation des examens en cliquant sur « visualiser les examens ». S10 : Le système affiche la liste des examens qu'a eus le patient.
T04 : Consulter les vaccins	S11 : L'utilisateur demande la visualisation des vaccins en cliquant sur « visualiser les vaccins ». S12 : Le système affiche la liste des vaccins qu'a eus le patient.
T05: Consulter l'Alergie	S13 : L'utilisateur demande la visualisation de l'Alergie en cliquant sur « visualiser les Alergie». S14 : Le système affiche la liste d' Alergie qu'a eus le patient.

T06 : Consulter les antécédents.	S15: L'utilisateur demande la consultation des antécédents du patient en cliquant sur « Visualiser les antécédents ». S 16: Le système affiche la liste des antécédents (familiaux et personnels).
T 07: Consulter la lettre d'orientation.	S17: L'utilisateur demande la consultation de la lettre d'orientation. S18 : Le système lui affiche un formulaire de recherche du patient. S19 : L'utilisateur remplit le formulaire.
T08 : Ajouter la visite infirmier.	S20: L'utilisateur demande l'ajout d'une visite. S 21: Le système affiche le formulaire d'ajout des visite. S22: L'utilisateur click sur « Ajouter ». S 23: Le système affiche un message pour confirmer l'ajout de la visite.
T09: Consulter l'ordonnance.	S24: L'utilisateur demande de la consultation d'ordonnance S 25: Le système affiche le formulaire de chercher le patient S26: L'utilisateur remplit le formulaire et click sur Afficher.

Tableau III.3.: Récapitulatif des scénarios par tâches d'un personnel paramédical.

Tâche	Scénarios
T10 : Idem que les tâches d'un personnel	S01 ; S26 ;
T11: Ajouter un nouveau patient.	S27 : L'utilisateur demande l'ajout d'un nouveau patient. S28 : Le système affiche un formulaire d'ajout. S29 : L'utilisateur remplit le formulaire. S30 : Le système vérifie les données reçues et retourne un message pour confirmer l'ajout d'un nouveau patient.

T12: Ajouter un vaccin	S32 : L'utilisateur demande l'ajout d'un vaccin en cliquant sur « Nouveau vaccin ». S33 : Le système lui affiche un formulaire d'ajout d'un vaccin. S34 : L'utilisateur saisit les informations demandées puis il click sur «Ajouter».
T 13: Ajouter un examen	S35 : L'utilisateur demande l'ajout d'examen en cliquant sur « Nouveau examen ». S36 : Le système lui affiche un formulaire d'ajout d'un examen. S37 : L'utilisateur saisit les informations demandées puis il click sur «Ajouter».
T 14: Ajouter une Allergie	S38 : L'utilisateur demande l'ajout d'examen en cliquant sur « Nouvelle Allergie ». S39 : Le système lui affiche un formulaire d'ajout d'une Allergie. S40 : L'utilisateur saisit les informations demandées puis il click sur «Ajouter».
T 15: Ajouter un antécédent.	S41 : L'utilisateur demande l'ajout d'un antécédent en cliquant sur « Mettre à jour la liste des antécédents». S42 : Le système lui affiche un formulaire d'ajout d'un antécédent. S43 : L'utilisateur saisit les informations demandées et il click sur « Valider ». S44 : Le système affiche, un message pour confirmer l'ajout de l'antécédent.
T16 Editer l'ordonnance.	S45 : L'utilisateur demande l'édition de l'ordonnance en cliquant sur « Editer l'ordonnance ». S46 : Le système affiche l'ordonnance.
T17 : Editer la lettre d'orientation.	S 47: L'utilisateur demande l'édition de la lettre d'orientation en cliquant sur « Editer la lettre d'orientation ». S48 : Le système affiche la lettre d'orientation.
T18 : Editer la fiche d'examen.	S49 : L'utilisateur demande l'édition de la fiche d'examen en cliquant sur « Editer la fiche d'examen ». S50 : Le système affiche la fiche d'examen
T19 : Editer la fiche du vaccin	S51 : L'utilisateur demande l'édition de la fiche d'examen en cliquant sur « Editer la fiche d'examen ». S52 : Le système affiche la fiche d'examen
T20 : Editer la fiche de l'Allergie	S53 : L'utilisateur demande l'édition de la fiche d'examen en cliquant sur « Editer la fiche de l'Allergie ». S54 : Le système affiche la fiche de l'Allergie

Tableau III.4 : Récapitulatif des scénarios par tâches d'un Personnel médical.

Tâches	Scénarios
T24 : S'authentifier.	S01 ; S04;
T25 : Créer un compte utilisateur.	S58 : L'utilisateur demande l'ajout d'un utilisateur en cliquant sur « Ajouter un professionnel ». S59 : Le système affiche un formulaire d'ajout. S60 : L'utilisateur remplit le formulaire et valide l'ajout en cliquant sur « Enregistrer ». S61 : Le système vérifie la validité des informations reçues et affiche un message pour confirmer l'ajout.
T26 : Supprimer un compte utilisateur.	S62 : L'utilisateur demande la suppression d'un utilisateur en cliquant sur « Supprimer un professionnel ». S63 : Le système lui retourne la liste des utilisateurs ainsi leurs informations. S64 : L'utilisateur saisit l'identifiant de l'établissement à supprimer et il valide la suppression en cliquant sur « Supprimer ».

Tableau III.5. : Récapitulatif des scénarios par tâches d'un Administrateur.

III.3.6- Diagramme de cas d'utilisation

Bien souvent, les utilisateurs ne sont pas des informaticiens. Il leur faut donc un moyen simple d'exprimer leurs besoins. C'est précisément le rôle des diagrammes de cas d'utilisation qui permettent de recueillir, d'analyser et d'organiser les besoins, et de recenser les grandes fonctionnalités d'un système. Il s'agit donc de la première étape UML d'analyse d'un système.

Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit. Il scinde la fonctionnalité du système en unités cohérentes, les cas d'utilisation, ayant un sens pour les acteurs.

Il ne faut pas négliger cette première étape pour produire un logiciel conforme aux attentes des utilisateurs.

Le diagramme de cas d'utilisation de notre application est représenté par la figure ci-dessous :

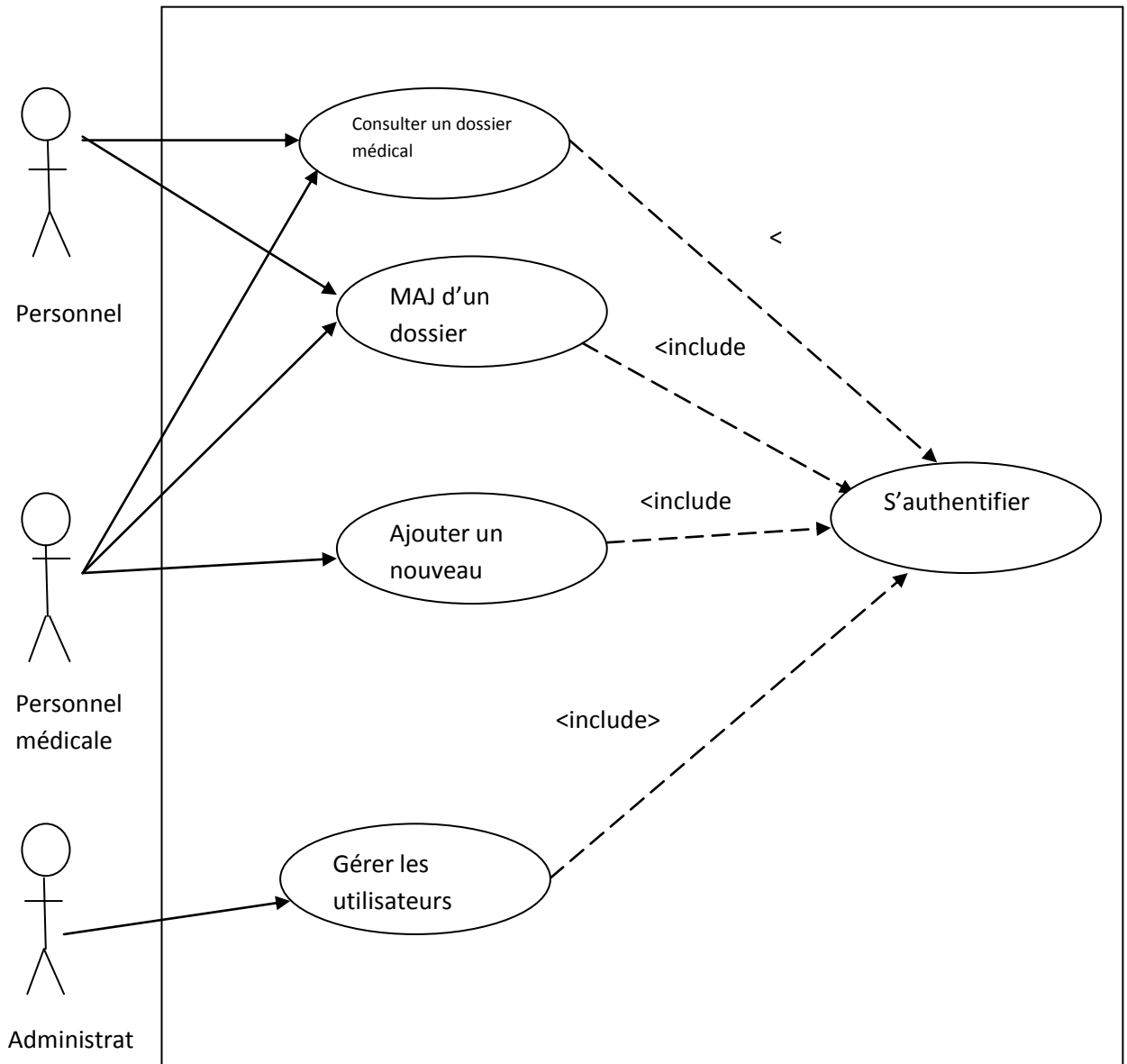


Figure III.2 : Diagramme de cas d'utilisation général.

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

nous allons commencé par le diagramme de cas d'utilisation du personnel paramédicale

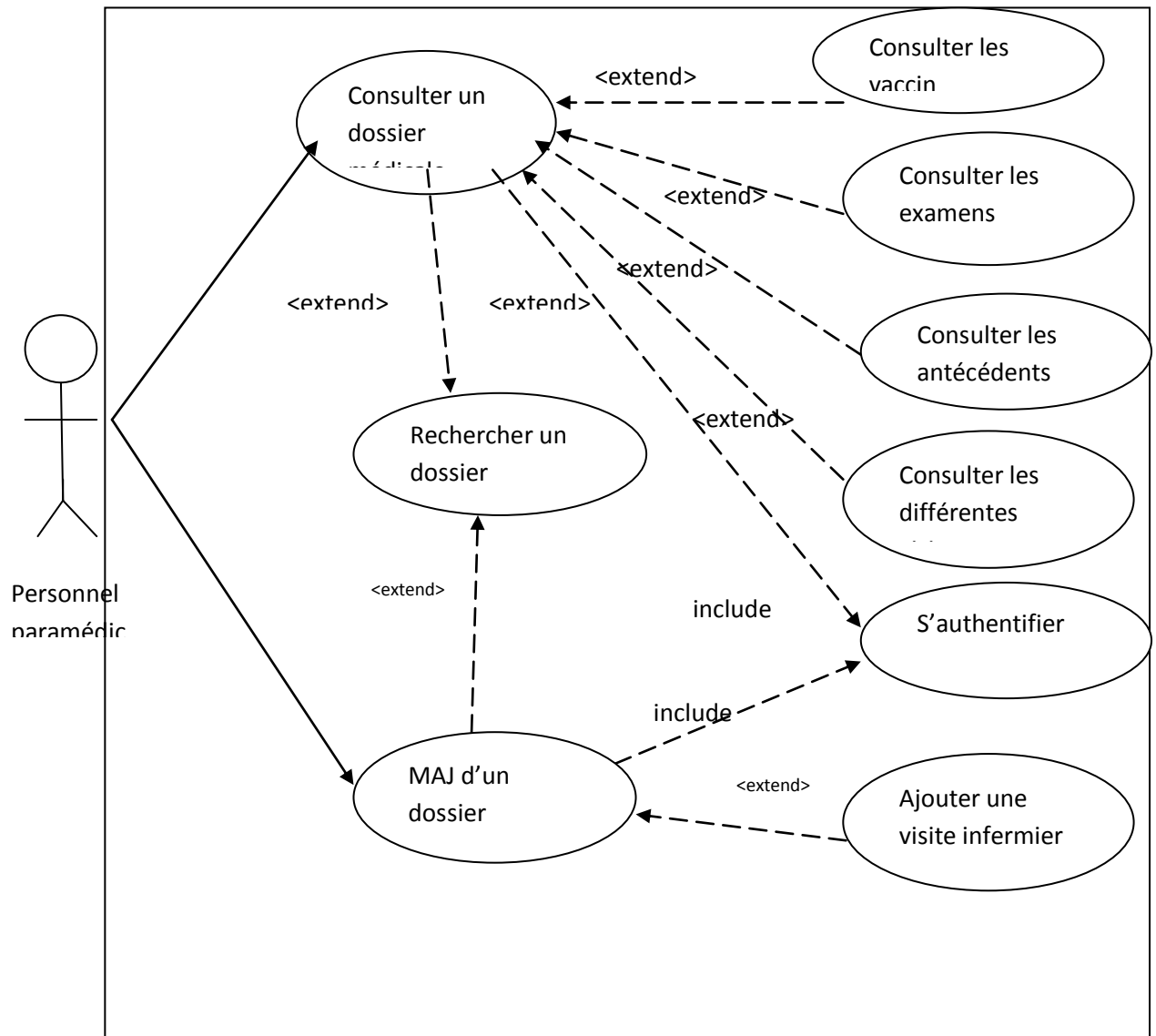


Figure III.3. : Diagramme de cas d'utilisation pour le personnel paramédical

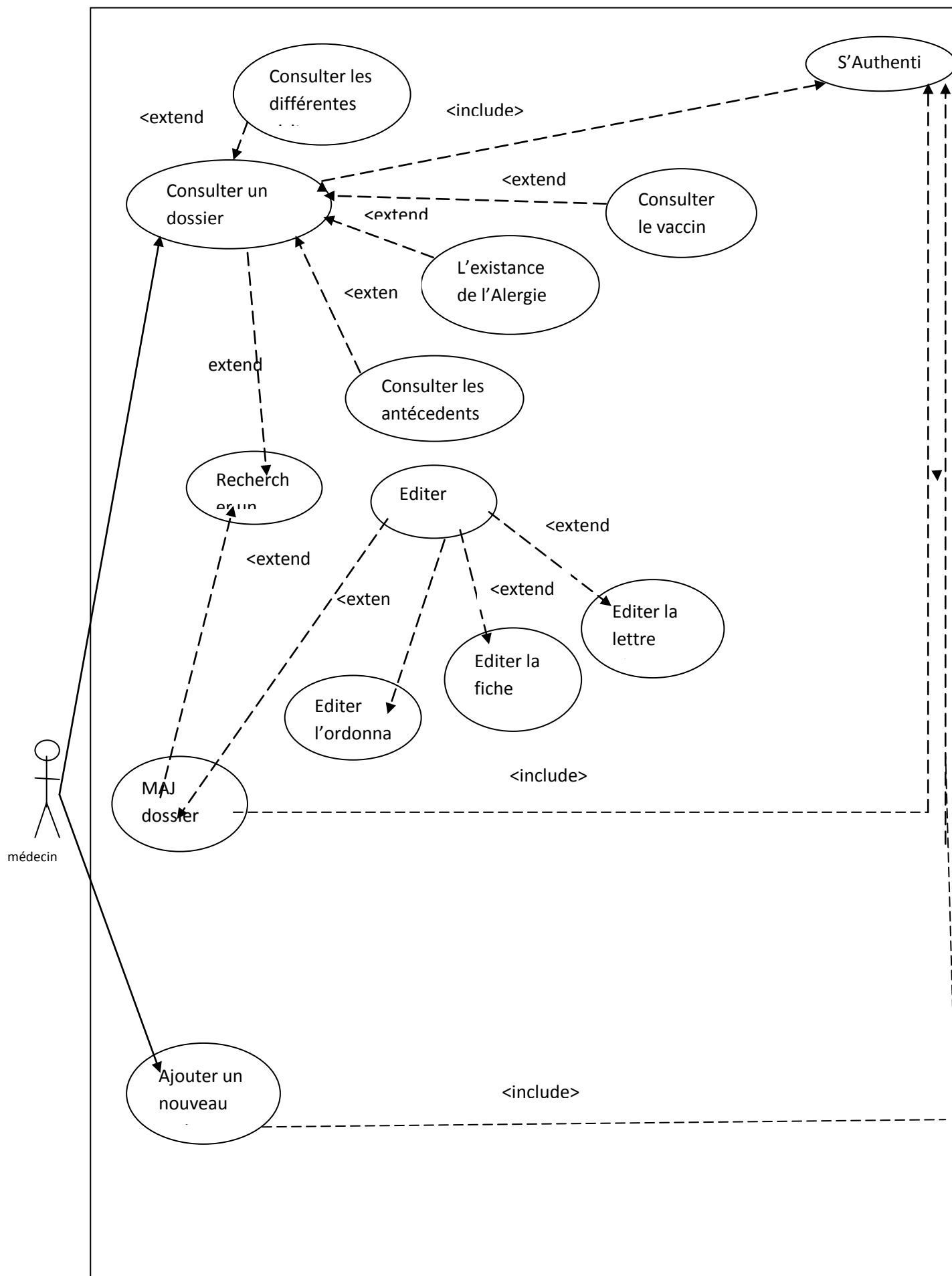


Figure III.4. : Diagramme de cas d'utilisation pour le médecin

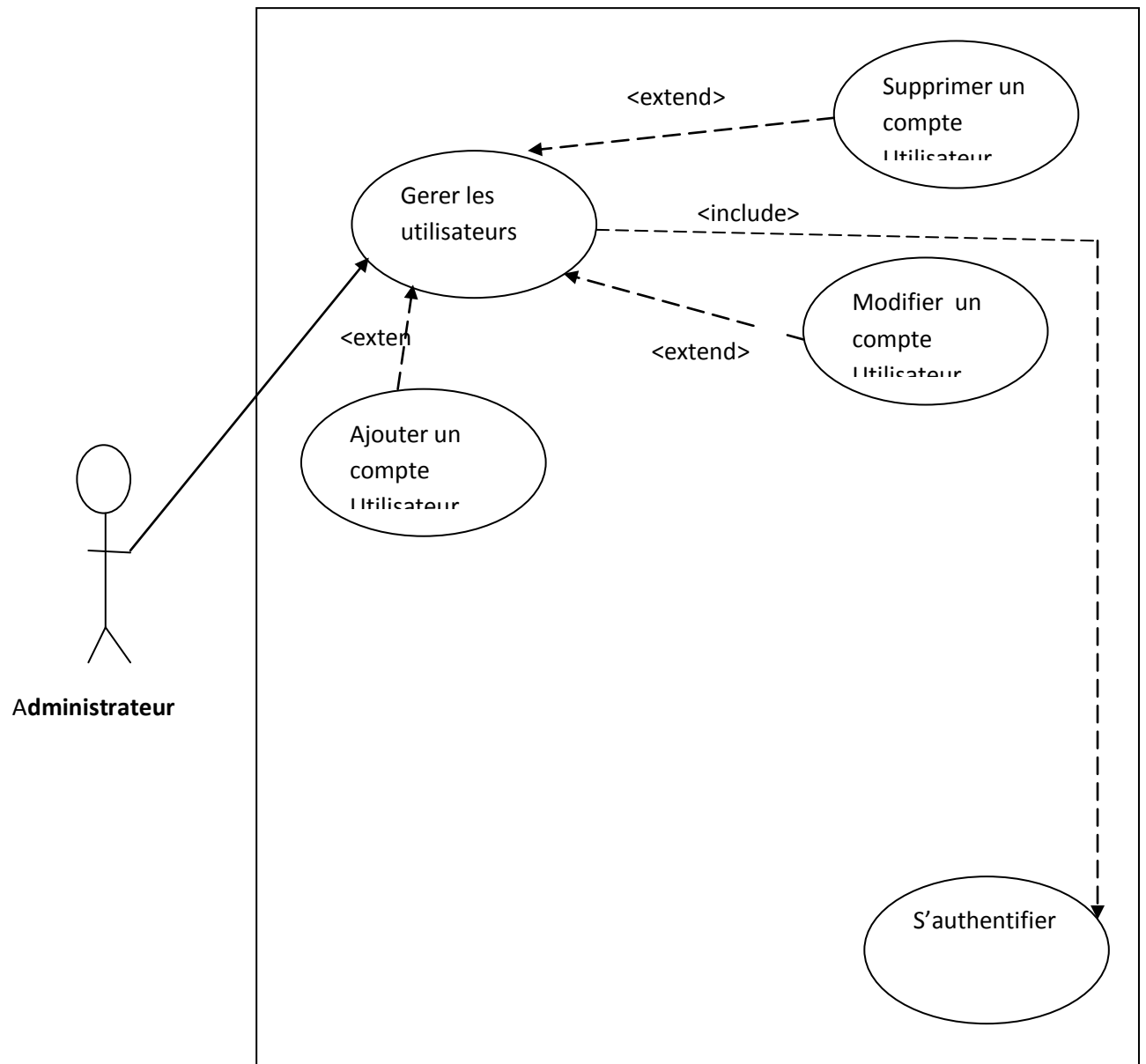


Figure III.5 : Diagramme de cas d'utilisation pour l'Administrateur

III.4-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 analyse.

III.4.1 Diagramme de séquences

Le diagramme de séquence montre des interactions entre objets selon un point de vue temporel. Ce type de diagramme sert à modéliser les aspects dynamiques des systèmes temps réels et des scénarios complexes mettant en œuvre peu d'objets.

Les principales informations contenues dans un diagramme de séquence sont les messages échangés entre les lignes de vie, présentés dans un ordre chronologique.

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

- ✓ S'authentifier ;
- ✓ Ajouter un nouveau patient
- ✓ Supprimer un compte utilisateur
- ✓ Consulter un dossier médical.

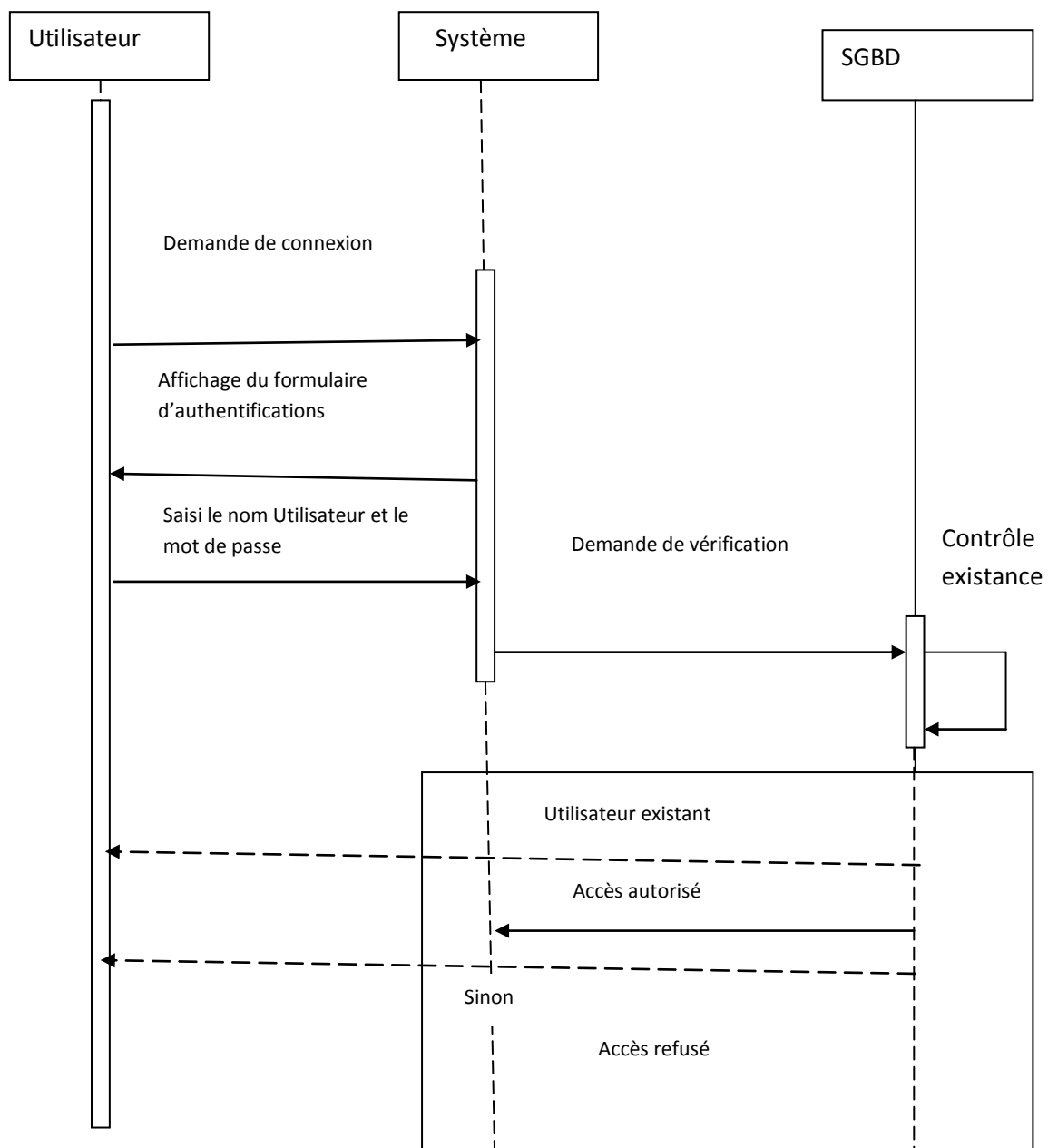
Diagramme de séquence de cas d'utilisation « s'authentifier » :**Figure III.6 : Diagramme de séquence simple de cas d'utilisation « S'authentifier ».**

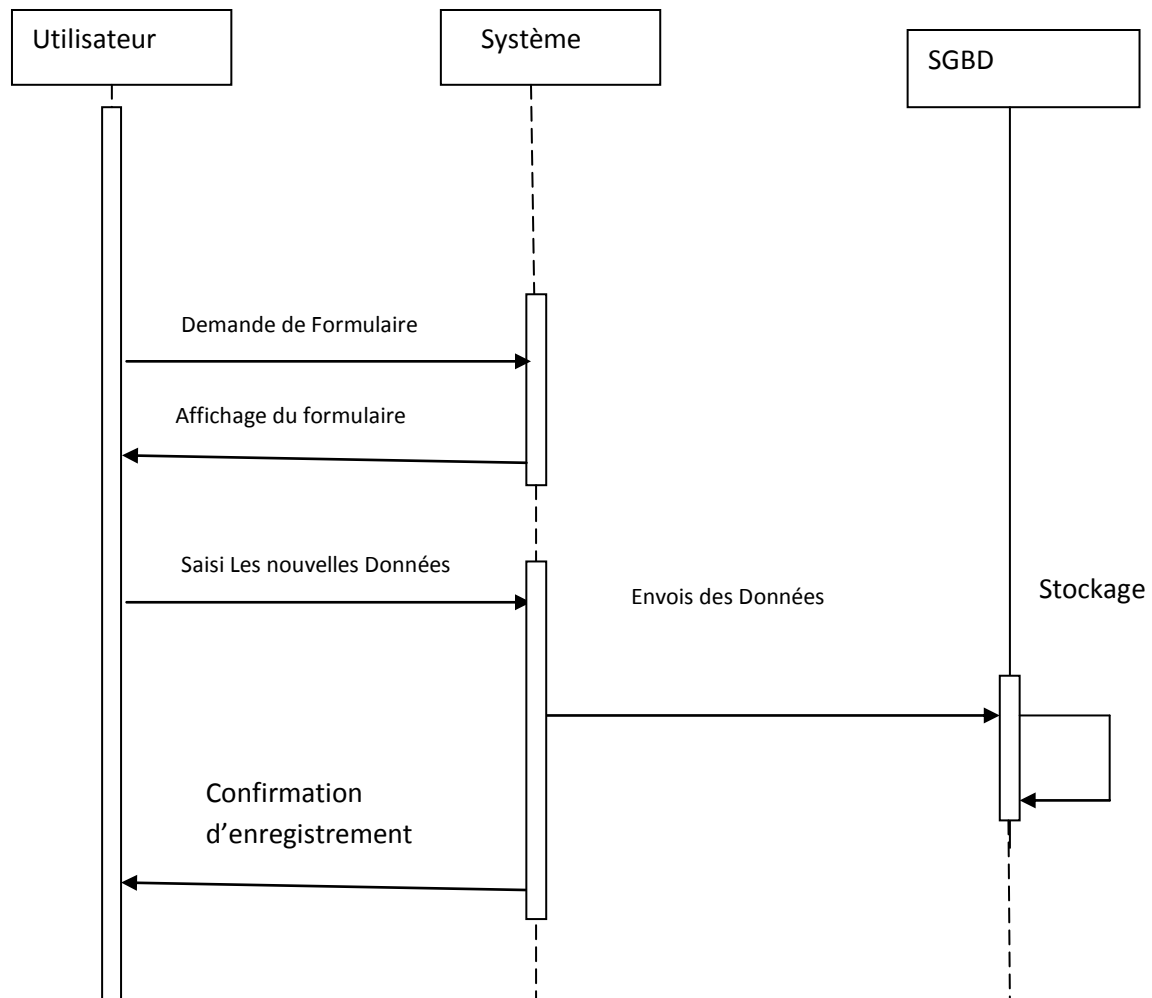
Diagramme de séquence de cas d'utilisation « Ajouter un nouveau patient »

Figure III.7 : Diagramme de séquence simple de cas d'utilisation « Ajouter un nouveau Patient.

Diagramme de sequence simple de cas d'utilisation « Supprimer un compte utilisateur »

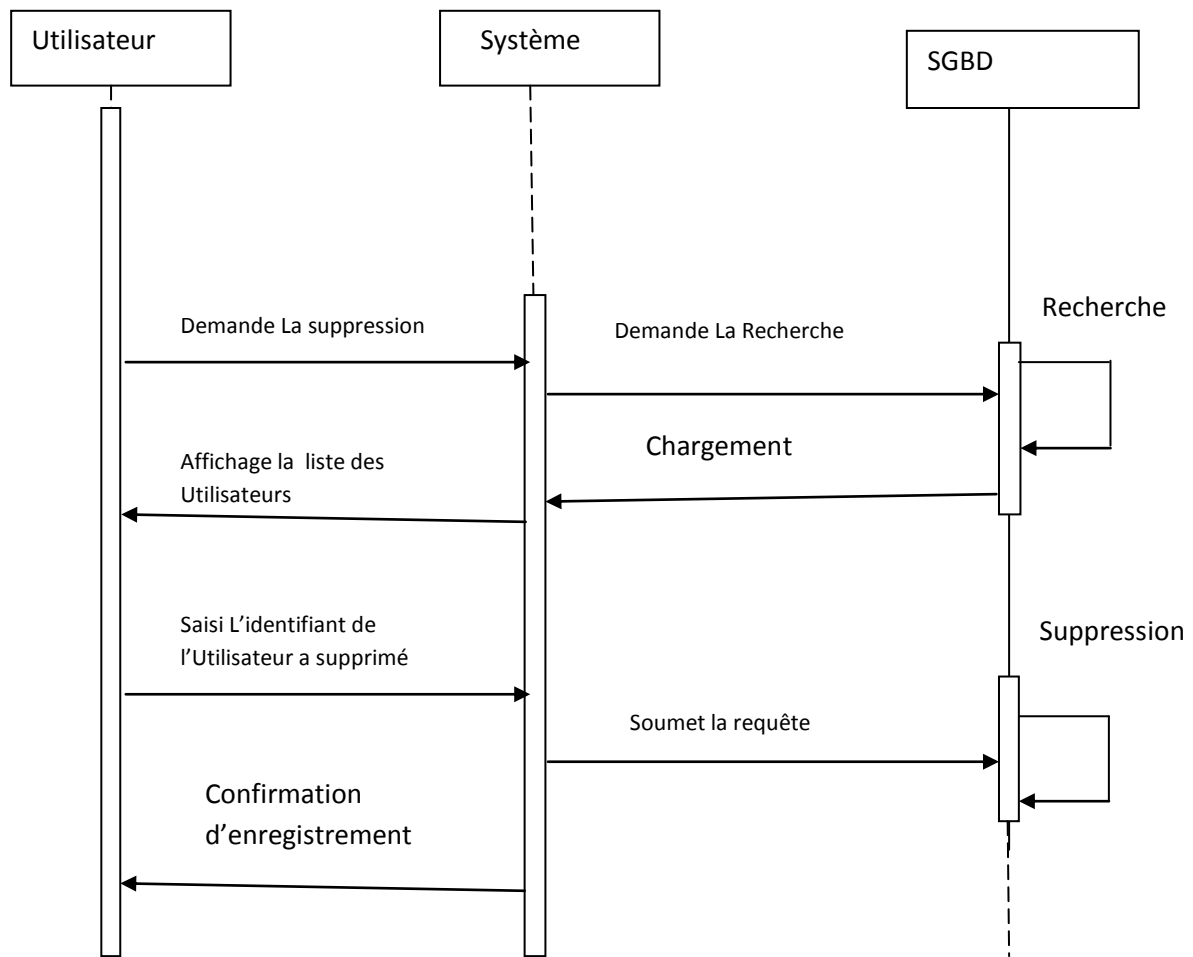


Figure III.8 : Diagramme de séquence simple de cas d'utilisation « Supprimer un compte Utilisateur ».

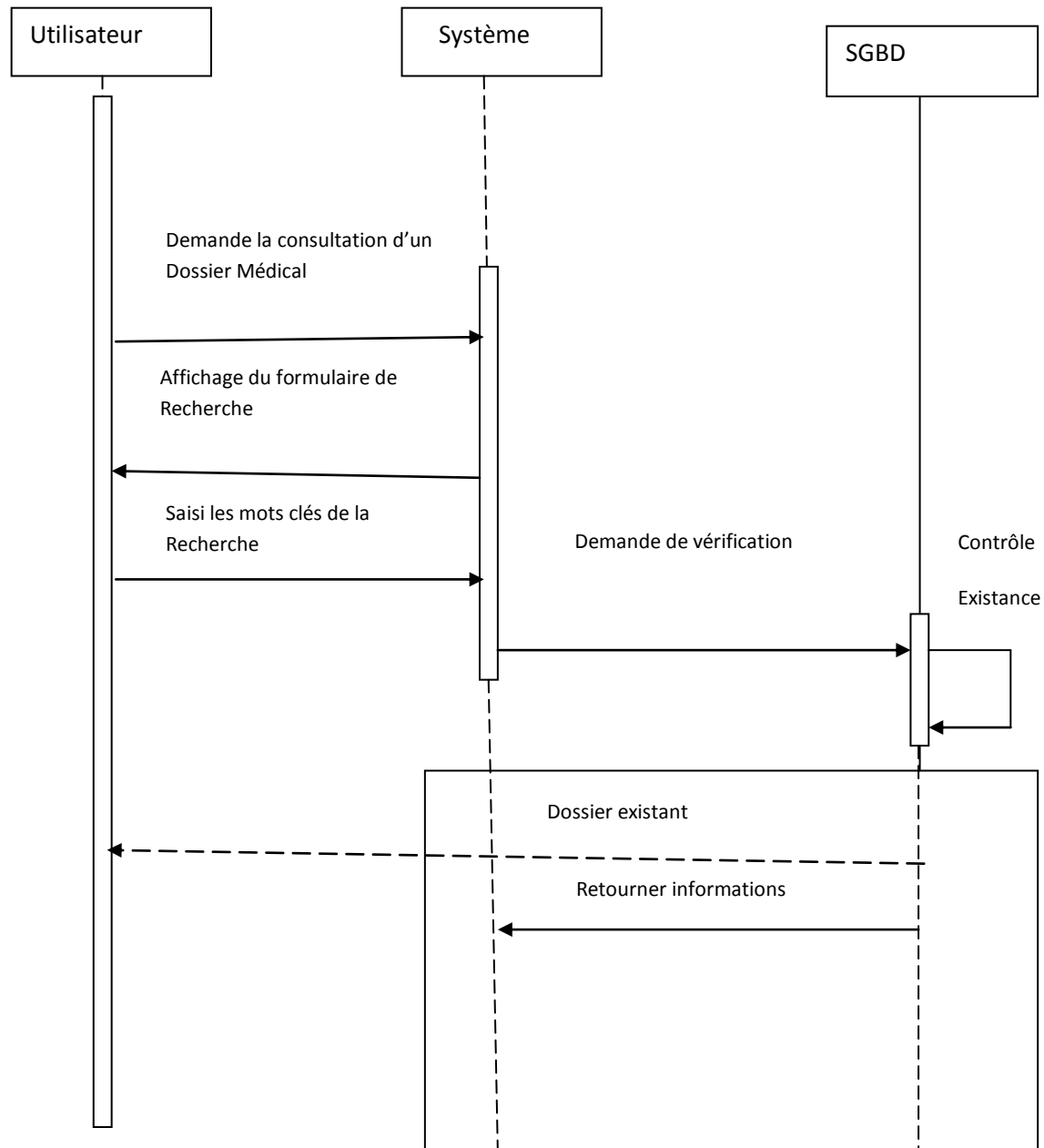
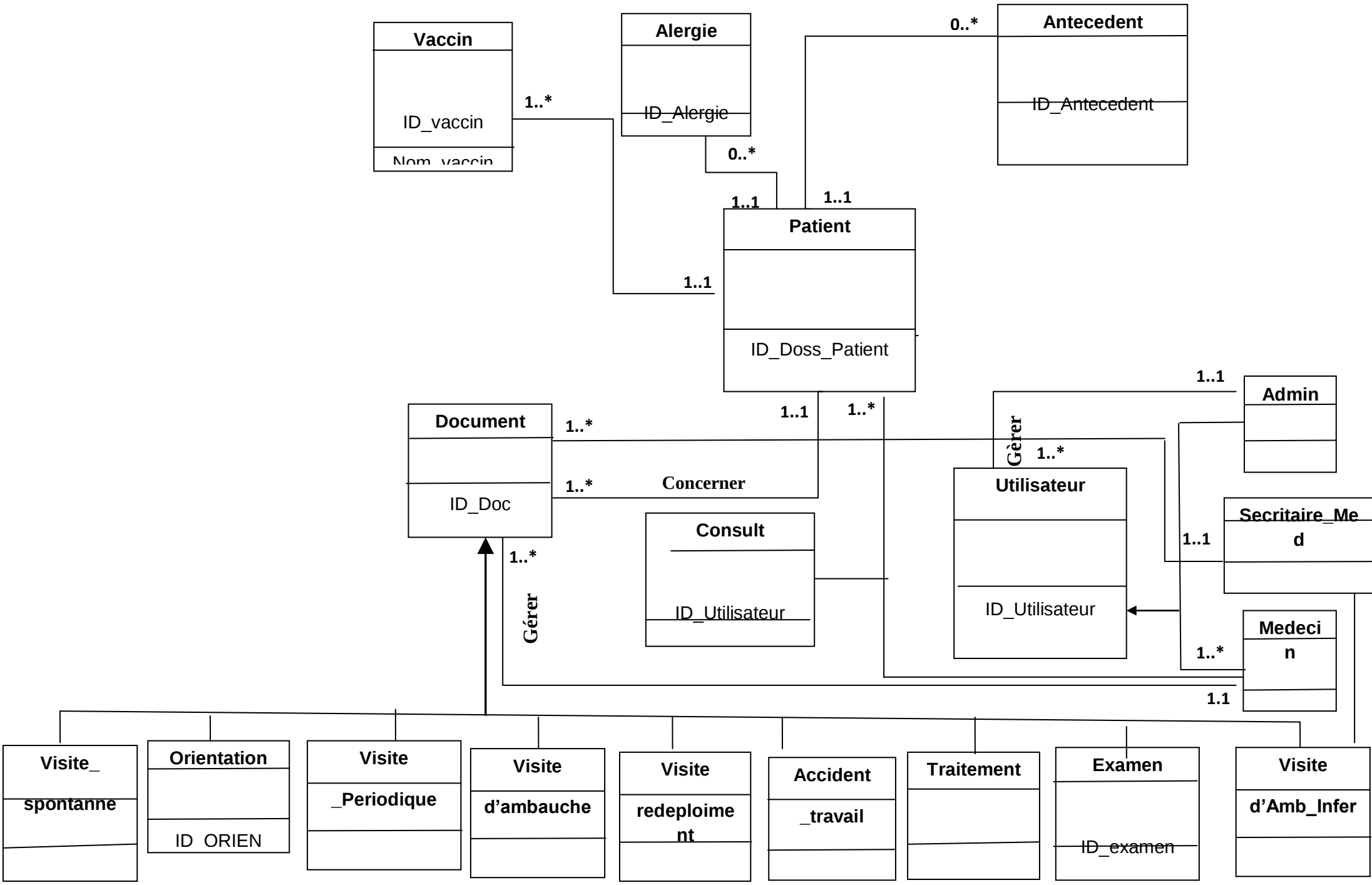
Diagramme de séquence de cas d'utilisation «Consulter un dossier médical » :

Figure III.9: Diagramme de séquence simple de cas d'utilisation « Consulter un dossier Médical».

III.4.2- Diagramme de classe

Le diagramme de classe montre la structure interne du système. Il permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d'utilisation.

Figure III.10 :Diagramme de classe



III.7-- Le modèle physique :

Ce modèle nous donne la représentation physique de l'ensemble des tables de la base des Données du système étudié.

Table Allergie :

Attribut	Description	Type	Clé
ID_ALERGIE	Numéro d'identification d'allergie	Int(10)	Primaire
ID_DOSSIER_PATIENT	Numéro d'identification du patient	Int(10)	Etrangère
NOM_ALERGIE	Nom de l'allergie	Varchar(30)	
COMMENTAIRE	Commentaire	Text	

Table Patient

Attribut	Description	Type	Clé
Id_Doss_Patient	Numéro d'identification du patient	Int(10)	Primaire
Nom_Patient	Nom du patient	Varchar(30)	
Prenom_Patient	Prénom du patient	Varchar(30)	
Date_Naissance_p	Date de naissance du patient	Date	
Adr_Patient	Adresse du patient	Varchar(50)	
Tel_Patient	Numéro de téléphone du patient	Int(15)	
Profession_Patient	Profession du patient	Varchar(20)	
Sexe	Sexe	Varchar(8)	
Situation	Situation familiale du patient	Varchar(8)	

Table Antecedent

Attribut	Description	Type	Clé
ID_Antecedent	Numéro de l'antécédent	Int(10)	Primaire
Type_Antecedent	Type de l'antécédent	Varchar(50)	
Id_Doss_Patient	Numéro d'identification du patient	Int(10)	Etrangère
Commentaire	Commentaire sur l' antécédent	text	

Table pour Accident de travail :

Attribut	Description	Type	Clé
ID_At	Numéro de Accédant	Int(10)	Primaire
Id_Document	Numéro d'identification du Document	Int(10)	Etrangère

Siège _lèsion	Siège _lèsion	text	
Date	Date	Date	
Nature	Nature	Text	

Table pour la visite d'embauche :

Attribut	Description	Type	Clé
ID_Visite_Amb	Identifiant de la visite	Int(10)	Primaire
Id_Document	Numéro d'identification du Document	Int(10)	Etrangère
Date	Date	Date	
Opht	Opht	Varchar(50)	
Locomoteur	Locomoteur	Varchar(50)	
Respiratoire	Respiratoire	Varchar(50)	
Cardio_Vasculaire	Cardio_Vasculaire	Varchar(50)	
Digestif	Digestif	Varchar(50)	

Table pour la visite Périodique:

Attribut	Description	Type	Clé
ID_Visite_Per	Identifiant de la visite	Int(10)	Primaire
Id_Document	Numéro d'identification du Document	Int(10)	Etrangère
Date	Date	Date	
Opht	Opht	Varchar(50)	
Locomoteur	Locomoteur	Varchar(50)	
Respiratoire	Respiratoire	Varchar(50)	
Cardio_Vasculaire	Cardio_Vasculaire	Varchar(50)	
Digestif	Digestif	Varchar(50)	

Table pour la visite Redéploiement :

Attribut	Description	Type	Clé
ID_Visite_Amb	Identifiant de la visite	Int(10)	Primaire
Id_Document	Numéro d'identification du Document	Int(10)	Etrangère
Date	Date	Date	
Opht	Opht	Varchar(50)	
Locomoteur	Locomoteur	Varchar(50)	
Respiratoire	Respiratoire	Varchar(50)	
Cardio_Vasculaire	Cardio_Vasculaire	Varchar(50)	
Digestif	Digestif	Varchar(50)	

Table pour la visite_Infirmier :

Attribut	Description	Type	Clé
ID_visite_infer	Identifiant de la visite	Int(10)	Primaire
Id_document	Identifiant de la table	Int(10)	Etrangère

	document		
Date_visite	Date_visite	Date	
Poids	Poids	Varchar(50)	
Taille	Taille	Varchar(50)	
Pt	Pt	Varchar(50)	
Urinnes	Urinnes	Varchar(50)	
Audition	Audition	Varchar(50)	
Vision_Sans_correction	Vision_Sans_correction	Varchar(50)	
Avec_Correction	Avec_Correction	Varchar(50)	

Table Pour la Visite Spontanée :

Attribut	Description	Type	Clé
Id_VS	Identifiant de la visite	Int(10)	Primaire
Id_document	Identifiant de la table document	Int(10)	Etrangère
Diagnostic	Diagnostic	Text	
Date	Date	Date	

Table utilisateur:

Attribut	Description	Type	Clé
Id_Utilisateur	Identifiant de l'utilisateur	Int(10)	Primaire
Nom	Nom de l'utilisateur	Varchar(50)	
Prenom	premier de l'utilisateur	Varchar(50)	
Date_Naiss	Date de naissance de l'utilisateur	Date	
Adresse	Adresse de l'utilisateur	Varchar(50)	
Téléphone	Numéro de téléphone utilisateur	Int(10)	
Type_Utilisateur	Nom utilisateur qui permet aux utilisateurs d'accéder à notre système.	Varchar(30)	
Pw_Utilisateur	Le mot de passe de l'utilisateur.	Varchar(10)	

Table Traitement :

Attribut	Description	Type	Clé
Id_Traietement	Identifiant de l'utilisateur	Int(10)	Primaire
Id_document	Identifiant du document	Int(10)	Etrangère
Date	Date	Date	
Nom_Trait	Nom de traitement	Varchar(10)	
Dosage	La dose	Varchar(10)	
Periode	La période de traitement	Varchar(10)	

Table vaccin :

Attribut	Description	Type	Clé
Id_Vaccin	Identifiant de l'utilisateur	Int(10)	Primaire
Id_Doss_Patient	Numéro d'identification du patient	Int(10)	Etrangère
Nom_vaccin	Nom du vaccin	Varchar(50)	
Commentaire	Commentaire sur Le Vaccin	Text	
Date_vaccin	Date du vaccin	Date	

Table Examen:

Attribut	Description	Type	Clé
Id_Examen	Identifiant du l'examen	Int(10)	Primaire
Id_Document	Identifiant du document	Int(10)	Etrangère
Date	Date	Date	
Commentaire	Commentaire sur l'examen	text	

Type_Examen	Type de l'examen à faire	Varchar(50)	
-------------	--------------------------	-------------	--

Table Document :

Attribut	Description	Type	Clé
Id_Document	Identifiant du document	Int(10)	Primaire
Id_Utilisateur	Identifiant du patient	Int(10)	Etrangère
Id_doss_Patient	Identifiant du dossier	Int(10)	Etrangère

Table Orientation :

Attribut	Description	Type	Clé
Id_Orientation	Identifiant de l'orientation	Int(10)	Primaire
Id_Document	Identifiant du document	Int(10)	Etrangère
Destinataire	Destinataire	Varchar(10)	
Date	Date	Date	

Table Consult:

Attribut	Description	Type	Clé
Id_Doss_Patient	Identifiant de Patient	Int(10)	Etrangère
Id_Utilisateur	Identifiant de l'utilisateur	Int(10)	Etrangère
Date	Date	Date	

III.6- Conclusion

Dans ce chapitre nous avons suivi une démarche de modélisation pour développer notre application, en se basant sur l'UML. Ainsi nous avons commencé par la spécification

des besoins et les divers cas d'utilisations, puis en phase de conception nous avons élaboré le diagramme de séquence et enfin nous avons défini les tables de notre base de données.

Le chapitre suivant sera consacré à la partie réalisation de notre application et sa mise en œuvre.

Introduction

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

Ce chapitre sera divisé en deux parties : la première partie sera consacrée à la description des outils de développement de notre application. La deuxième partie contiendra les détails techniques d'implémentation des différentes fonctionnalités offertes par notre système.

A. Partie 1 : Description des outils de développement**IV.1. Outils de développement****IV.1.1. Langage de programmation (Java) [14]**

Le langage de programmation que nous avons choisi pour le développement de notre application est le langage JAVA, développé par SUN Microsystems, il est disponible pour les principales plates-formes du marché (LINUX, Windows ou autres) et il est totalement gratuit. JAVA possède de nombreuses caractéristiques :

- Simple du fait que sa syntaxe soit basée sur celle du C++, mais dépouillée de tous les mécanismes complexes, redondants et inutiles (pointeurs,...).
- Performant, puissant, java est une plate-forme de développement comportant une bibliothèque de classes très riche et de nombreux outils d'interfaces de programmations applicatifs (API).
- Interprété, portable et indépendant des architectures matérielles : Cette caractéristique est un avantage primordial pour java face à des applications transmises par un réseau et exécutées sur des machines hétérogènes. Un programme java est successivement compilé pour fournir un code intermédiaire indépendant de la plate-forme d'exécution (le byte code) simple et rapide à traduire en langage machine.
- Riche : Un des aspects importants de l'environnement java est la richesse de ses librairies.
- Java permet de développer de nombreux sortes de programmes dont :
 - Des applications, sous forme de fenêtre ou de console .
 - Des applets, qui sont des programmes JAVA incorporés à des pages web .

Dans notre mémoire nous avons choisi de développer avec JavaEE (Java 2 Entreprise Edition).

IV.1.2. IDE (Netbeans) [15]

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

NetBeans est disponible sous Windows, Linux, Solaris (sur x86 SPARC), Mac OS X et Open VMS.

IV.1.3. JSF [16]

Est un *framework* web MVC qui se focalise sur la simplification de la construction d'interfaces utilisateur, autrement dit la vue. Il propose nativement plus d'une centaine de balises à cet égard, et facilite la réutilisation des composants d'une interface utilisateur à l'autre.

IV.1.3.1. Structure d'une application JSF

Étudions brièvement comment se construit une application basée sur JSF. Cela ne devrait pas bouleverser nos habitudes, la structure globale est très similaire à celle d'une application web Java EE MVC traditionnelle :

- la vue est généralement assurée par des pages JSP ou par des pages XHTML (on parle alors de Facelets) .
- le modèle est assuré par des entités ou des JavaBeans .
- le contrôleur, auparavant incarné par nos servlets, est dorénavant décomposé en deux éléments :
 - une unique servlet mère servant de point d'entrée à toute requête, la FacesServlet ;
 - un JavaBean particulier, déclaré via une annotation et désigné par le terme *managed-bean*.

Nous observons que la différence majeure qui ressort jusqu'à présent est l'absence de servlets spécifiques à chaque page.

IV.1.4. Les serveurs déployés [17]

IV.1.4.1. Le serveur d'application (apache tomcat)

Le serveur Apache est le serveur HTTP le plus répandu avec près de 60% du marché, Soit le double de son principal concurrent Microsoft-IIS. Ces principaux avantages tiennent de son mode de distribution et à son mode de licences. L'avantage principal de ce produit n'est pas la gratuité, mais ce serveur http est le plus libre, et offre une totale indépendance à L'utilisateur pour la maintenance et le développement, il est aujourd'hui le serveur web le plus utilisé au monde.

IV.1.4.2. Le serveur de données(Oracle)

Le SGBD (Système de Gestion de Base de Données) Oracle est édité par la société Oracle Corporation, est un système de gestion de base de données qui offre une méthode de gestion des informations ouverte, complète et intégrée. Il permet d'assurer :

- ✓ La définition et la manipulation des données
- ✓ La cohérence des données
- ✓ La confidentialité des données
- ✓ L'intégrité des données
- ✓ La sauvegarde et la restauration des données
- ✓ La gestion des accès.

IV.1.5. Le framework (Hibernate) [18]

Hibernate est un logiciel écrit sous la responsabilité de Gavin King, qui fait entre autre partie de l'équipe de développement de JBOSS.

L'ensemble des données nécessaires au fonctionnement de l'application sont sauvegardées dans une base de données. La manipulation des données peut se faire de différentes manières : Par l'accès directement à la base en écrivant les requêtes SQL adéquates. Utiliser un outil d'ORM (objectRelationnalMapping) permettant de manipuler facilement les données et d'assurer leur persistance. Il en existe plusieurs. Hibernate, présenté dans ce document, est l'un des plus utilisés.

Pourquoi ajouter une couche entre l'application et la base de données ? L'objectif est de réduire le temps de développement de l'application en éliminant une grande partie du code SQL à écrire pour interagir avec la base de données et en encapsulant le code SQL résiduel. Les développeurs manipulent les classes dont les données doivent être persistantes comme des classes Java normales. Seules une initialisation correcte d'hibernate doit être effectuée, et quelques règles respectées lors de l'écriture et de la manipulation des classes persistantes.

IV.1.6. EJB [19]

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

B. Partie 2 : Présentation de l'application

Pour la réalisation proprement dite de notre application, on a opté pour la technologie JSF pour la couche graphique cliente.

Notre application suivra donc l'architecture JEE en couche suivante :

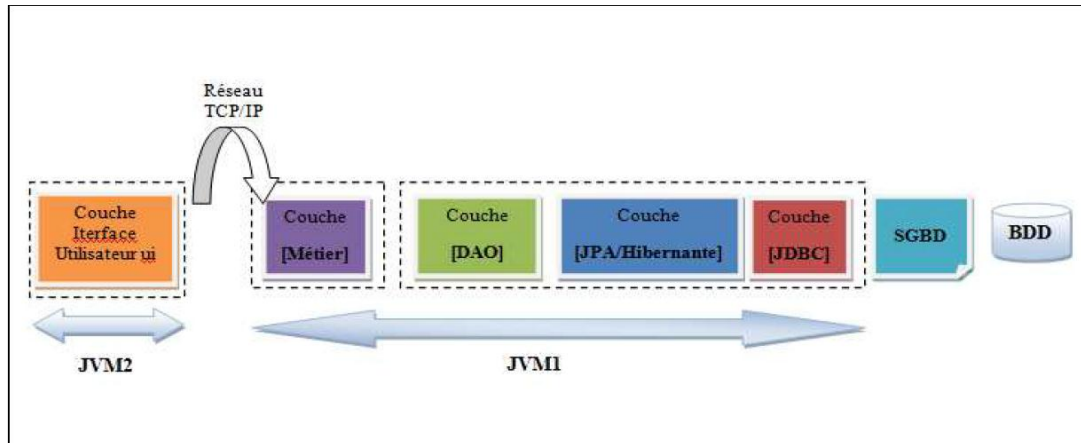


Figure IV.1. : Présentation de l'application.

Elle est constituée essentiellement de trois grandes parties :

1. Une partie stockage : cette partie contient une base de données qui nous permet de stocker et sauvegarder nos données.

2. Une parties traitement : dont laquelle on pourra effectuer les différents traitements sur nos données, cette dernière sera décomposée selon l'architecture JEE utilisées aux couches suivantes :

- ❖ **La couche [JDBC] :** (Java DataBaseConnectivity) est la couche standard utilisée en Java pour accéder à des bases de données. C'est ce qu'on appelle habituellement le pilote Jdbc du SGBD.
- ❖ **La couche [JPA/Hibernate] :** (Java Persistence API) se place entre la couche [dao] écrite par le développeur et la couche [JDBC]. Hibernate est un ORM (Object RelationalMapping), cette couche fait le pont entre le monde relationnel des bases de données et celui des objets manipulés par Java.
- ❖ **La couche [dao] :** (Data Access Object) est la couche qui fournit à la couche [metier] des données pré-enregistrées (fichiers, bases de données, ...) et qui enregistre certains des résultats fournis par la couche [metier].
- ❖ **La couche [metier] :** est la couche qui applique les règles dites métier, c.à.d la

logique spécifique de l'application, sans se préoccuper de savoir d'où viennent les données qu'on lui donne, ni où vont les résultats qu'elle produit.

3. **Une partie présentation** : Comme son nom l'indique, cette partie s'occupe de la présentation graphique (interfaces graphiques) du système pour les utilisateurs; elle est implémentée par des clients java.

Implémentation de notre application

Partie stockage :

Pour l'implémentation de cette partie, Nous avons créé avec le SGBD une base de données [master] contenant les différentes tables de chapitre conception.

Après avoir créé cette base de données, on a établie une connexion NetBeans /oracle à l'aide de pilote JDBC.

L'image suivante est une capture d'écran montrant la connexion établie, ainsi que les différentes tables de notre BDD.

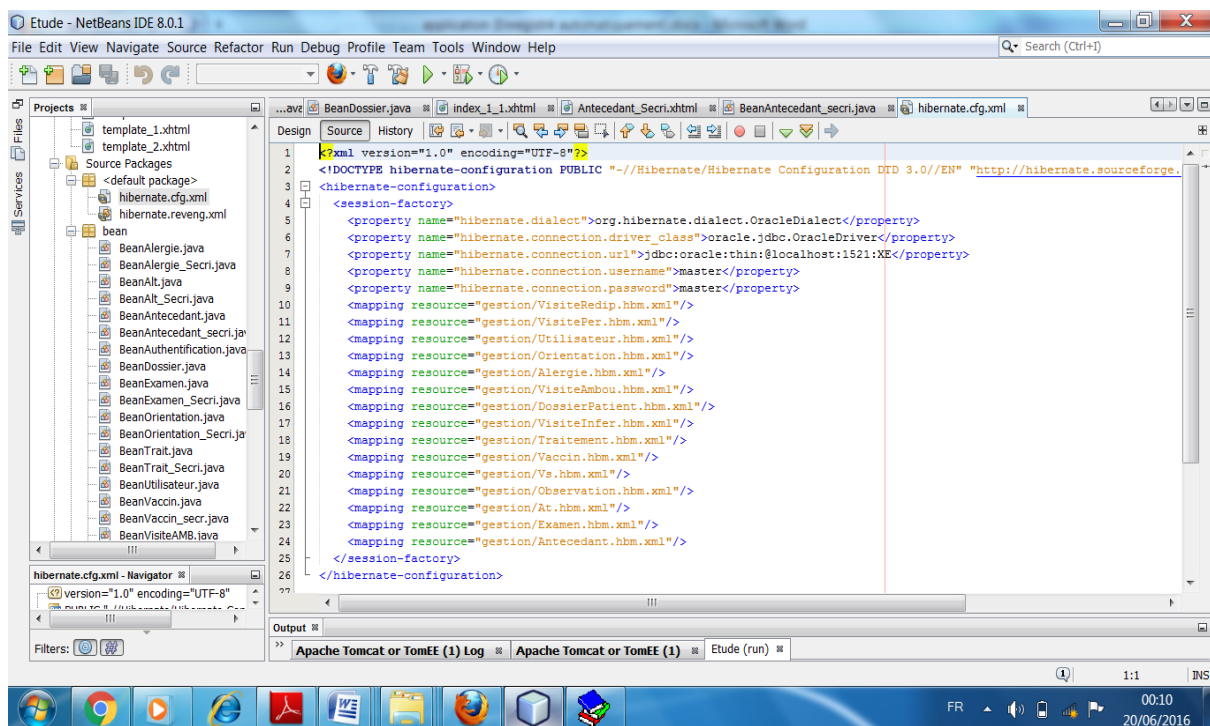


Figure IV.2. : La connexion établie et les différentes tables de notre BDD.

Partie présentation :

Elle sera consacrée pour donner une vue générale de notre application, et cela à travers les interfaces présentées ci-dessous.

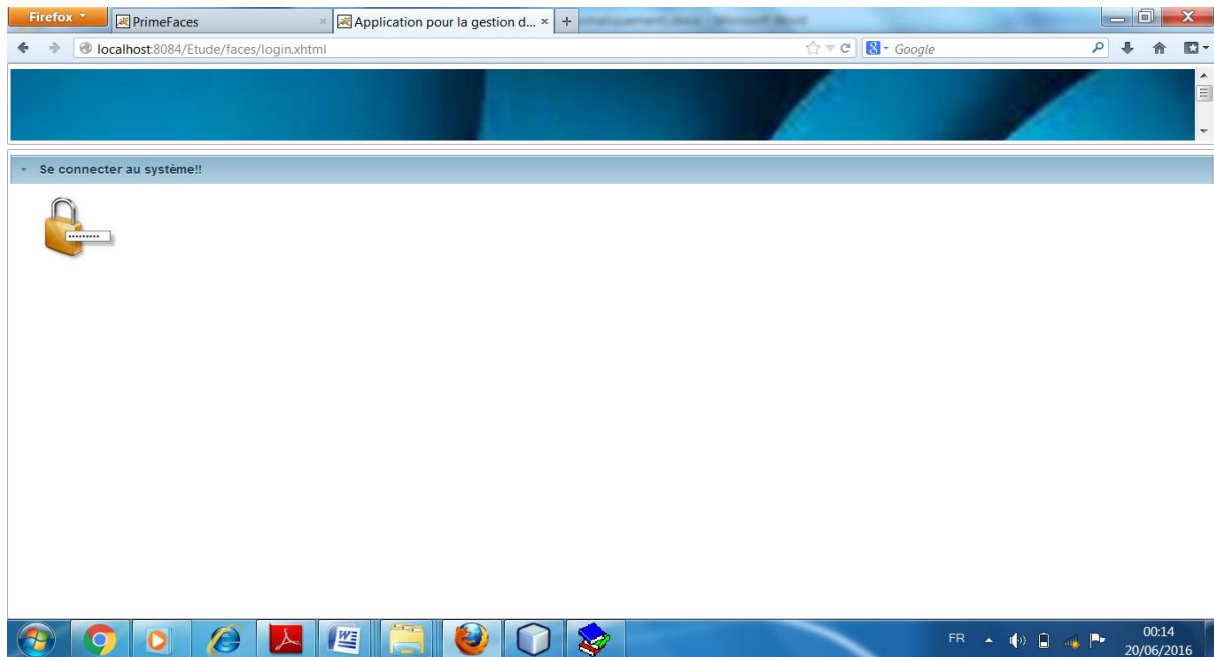
Interface accueil :

Figure IV.3. : La connexion établie et les différentes tables de notre BDD.

C'est la première fenêtre visible par notre application, elle nous permet d'aller vers la fenêtre D'authentification de l'administrateur et des utilisateurs (médecin et Secrétaire).

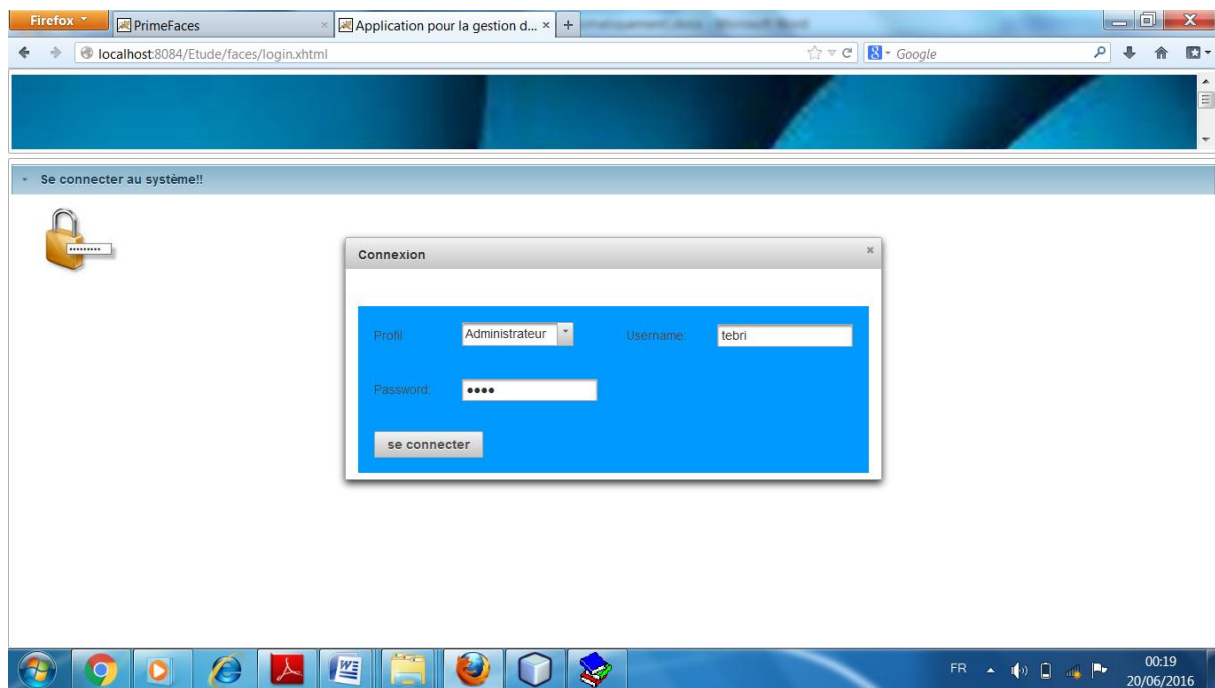
Interface Authentification de l'administrateur :

Figure IV.4. : Interface Authentification de l'administrateur.

C'est l'étape où l'administrateur pourra accéder à son espace en saisissant son 'nom D'utilisateur' et son 'Passeword'.

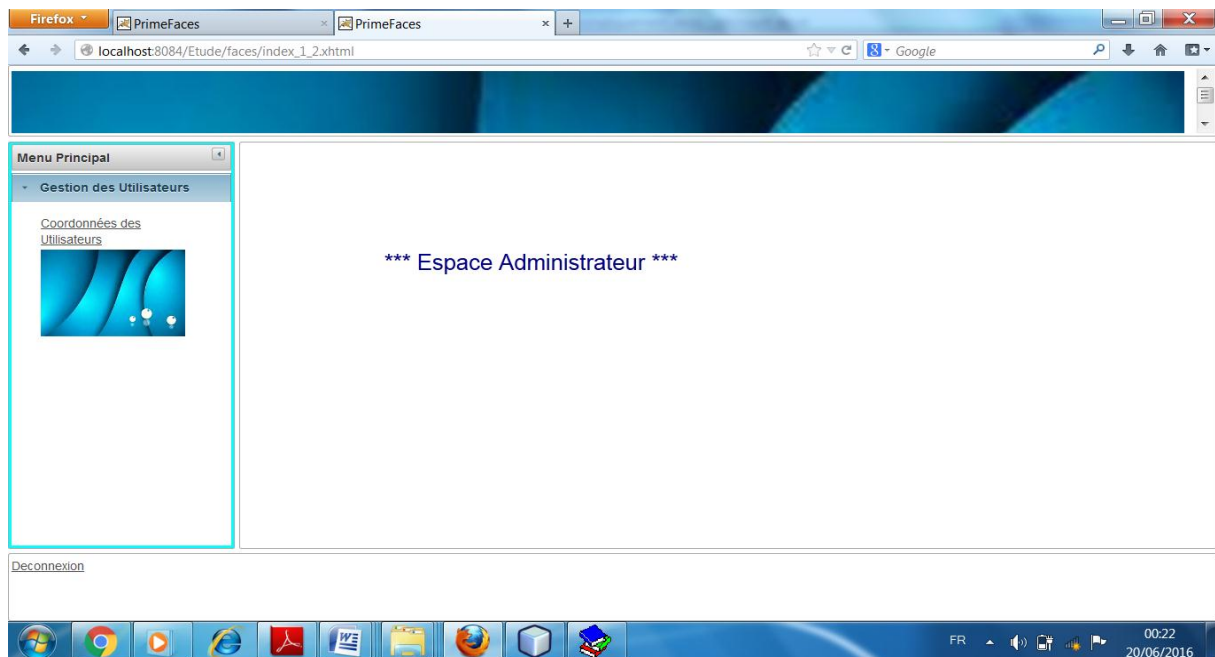
Interface Espace administrateur :

Figure IV.5. : Interface Espace administrateur.

Cette fenêtre s'affiche après la réussite de connexion de l'administrateur, elle lui permet

D'accéder aux différentes fenêtres de son propre espace afin d'effectuer les fonctionnalités Appropriées.

Interface Gestion des utilisateurs :

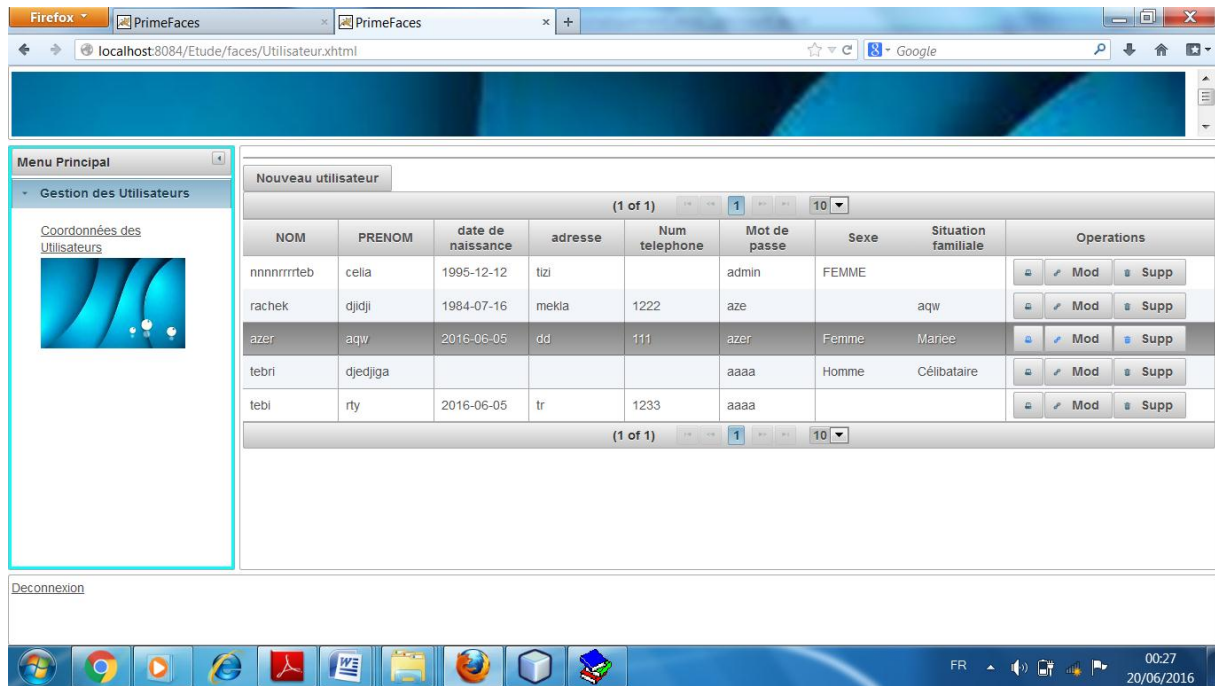


Figure IV.6. : Interface Gestion des utilisateurs.

Cette fenêtre s'affiche en cliquant sur bouton gestion des utilisateurs dans le menu de la fenêtre espace administrateur, elle permet de créer, d'afficher, modifier et supprimer un utilisateur.

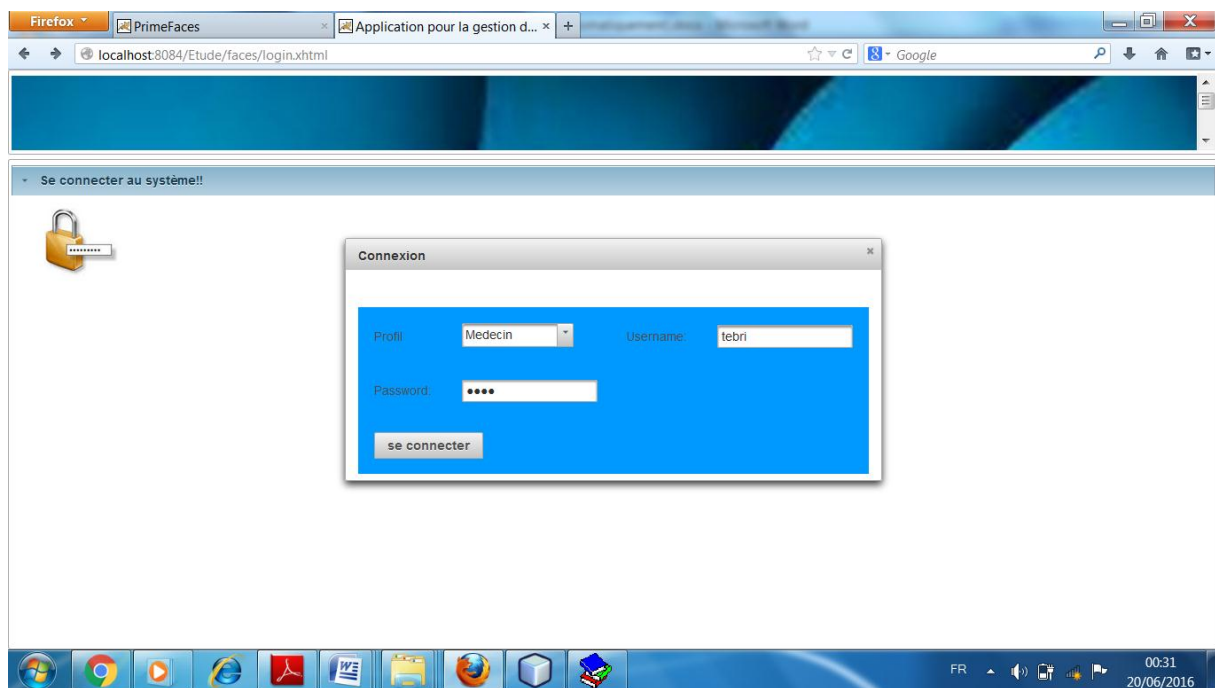
Interface d'Authentification des utilisateurs (médecin et infirmier) :

Figure IV.7. : Interface Gestion des utilisateurs.

Cette fenêtre conduit l'utilisateur à s'identifier par son nom utilisateur et son mot de passe on choisissant aussi le type de l'utilisateur (médecin ou Secrétaire). Cette étape lui permet d'accéder à son propre espace.

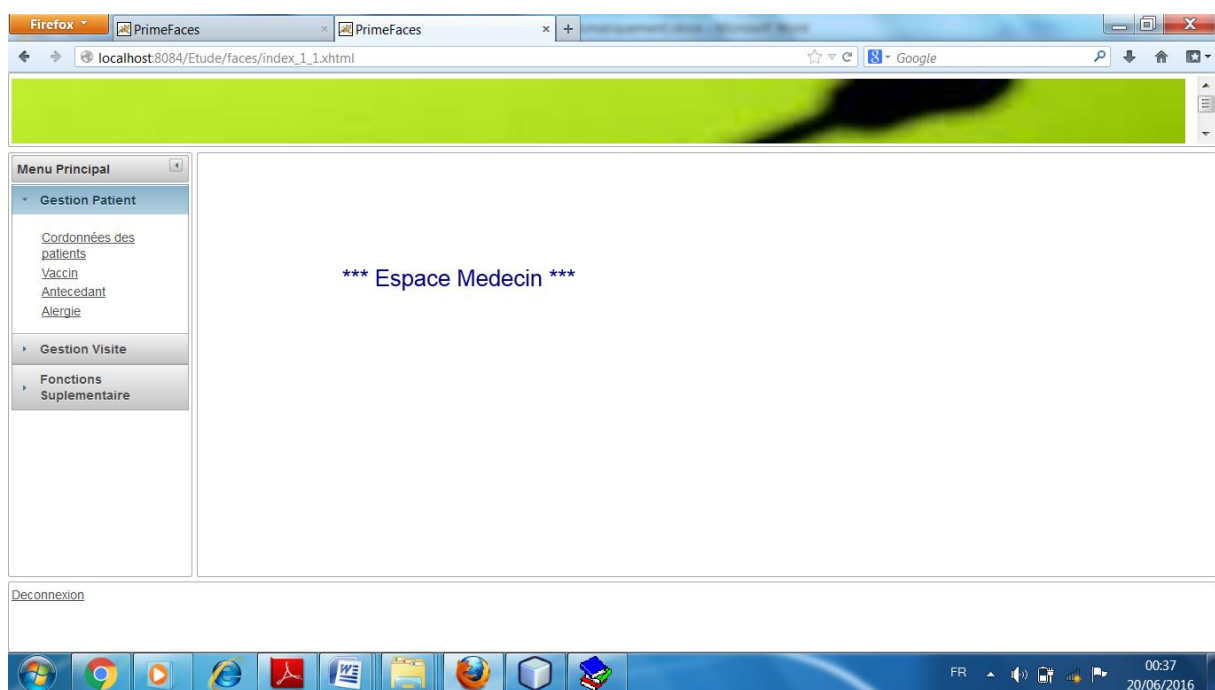
Interface espace médecin :

Figure IV.8. : Interface Espace médecin.

Cette fenêtre s'affiche après la réussite de connexion du médecin, elle lui permet d'accéder aux différentes fenêtres de son propre espace afin d'effectuer certains traitements après avoir tapé son nom d'utilisateur et son mot de passe en tant que médecin. **Interface**

Ajout nouveau patient :

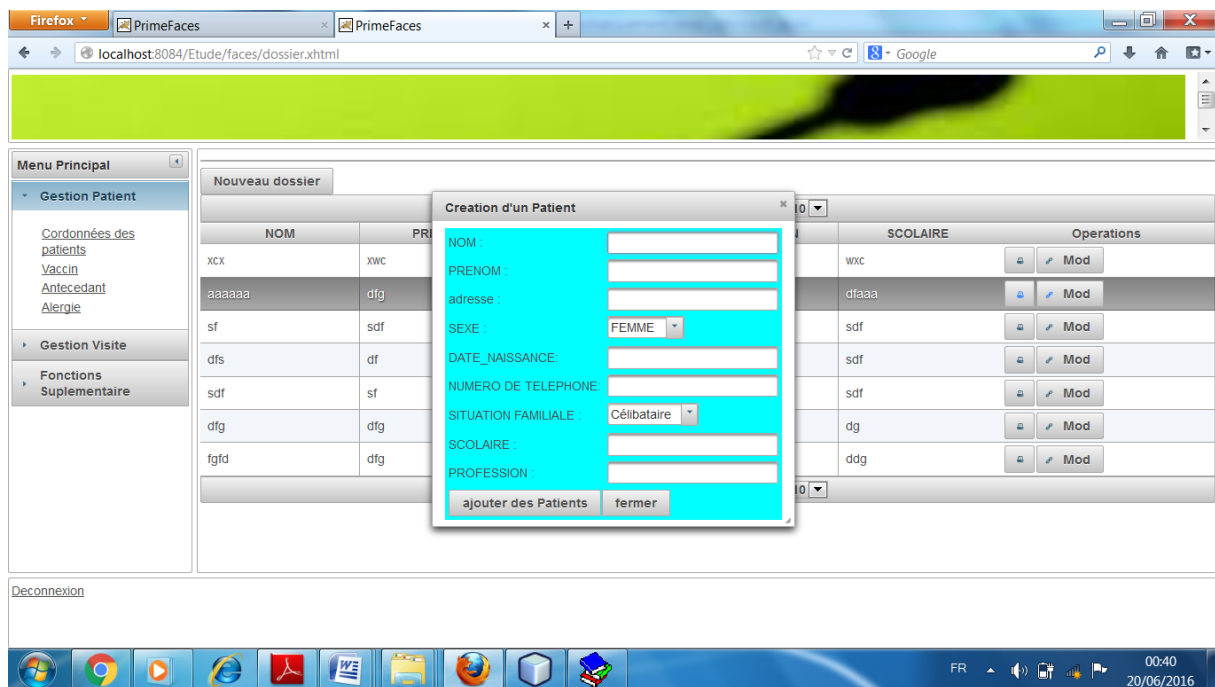


Figure IV.9. : Interface Ajout nouveau patient.

Cette fenêtre s'affiche en cliquant sur bouton nouveau patient dans le menu de la fenêtre espace médecin, elle permet de créer un nouveau dossier d'un patient en remplissant toutes les différentes informations.

Remarque : Comme on a plusieurs visites on donne juste quelques interfaces.

1- Interface visite d'embauche :

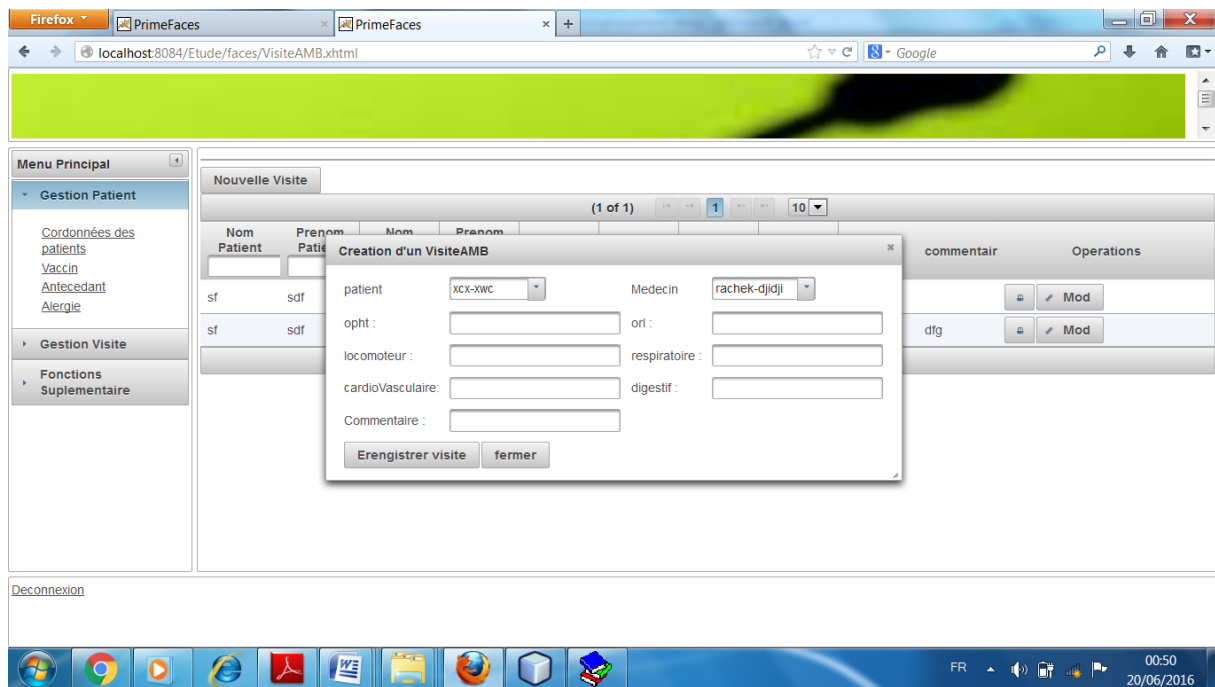


Figure IV.10. : Interface visite d'embauche .

2- Interface visite périodique :

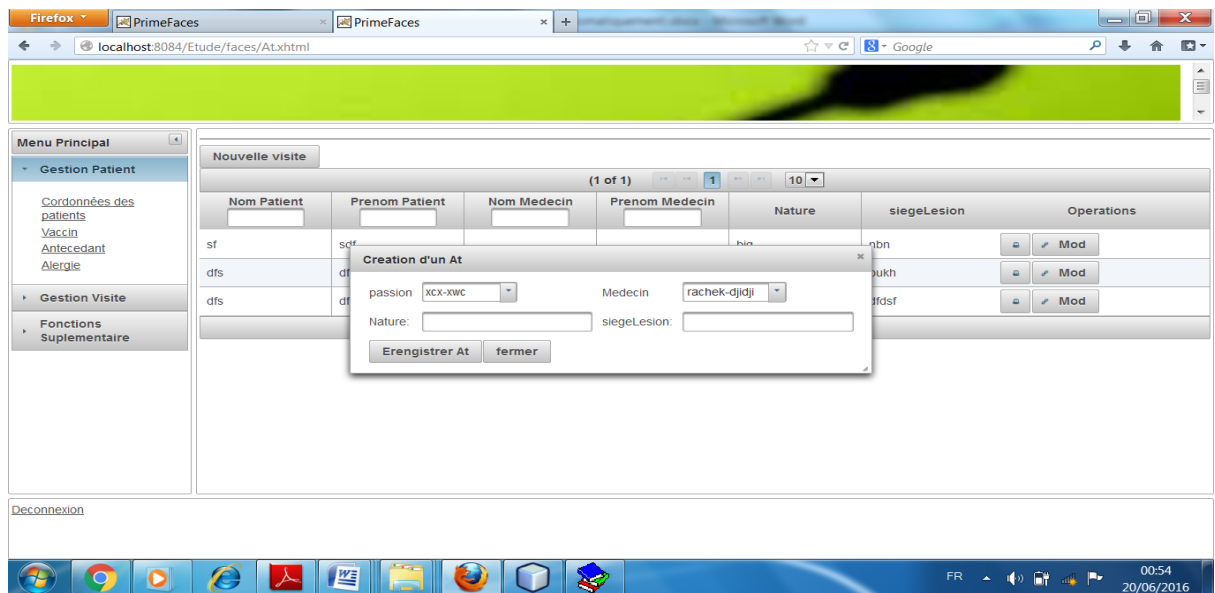


Figure IV.11. : Interface visite périodique .

Interfaces du Secrétaire

Le Secrétaire et le médecin ont des tâches presque similaires, mais le Secrétaire n'a pas le droit ni de prescrire ni de faire une consultation, ils sont des accompagnateurs des médecins, alors ils ont le droit que voir les dossiers médicaux sans rien y est changer. D'où les interfaces du secrétaire et celles du médecin sont presque identique. On prend le cas de l'interface Dossier médical :

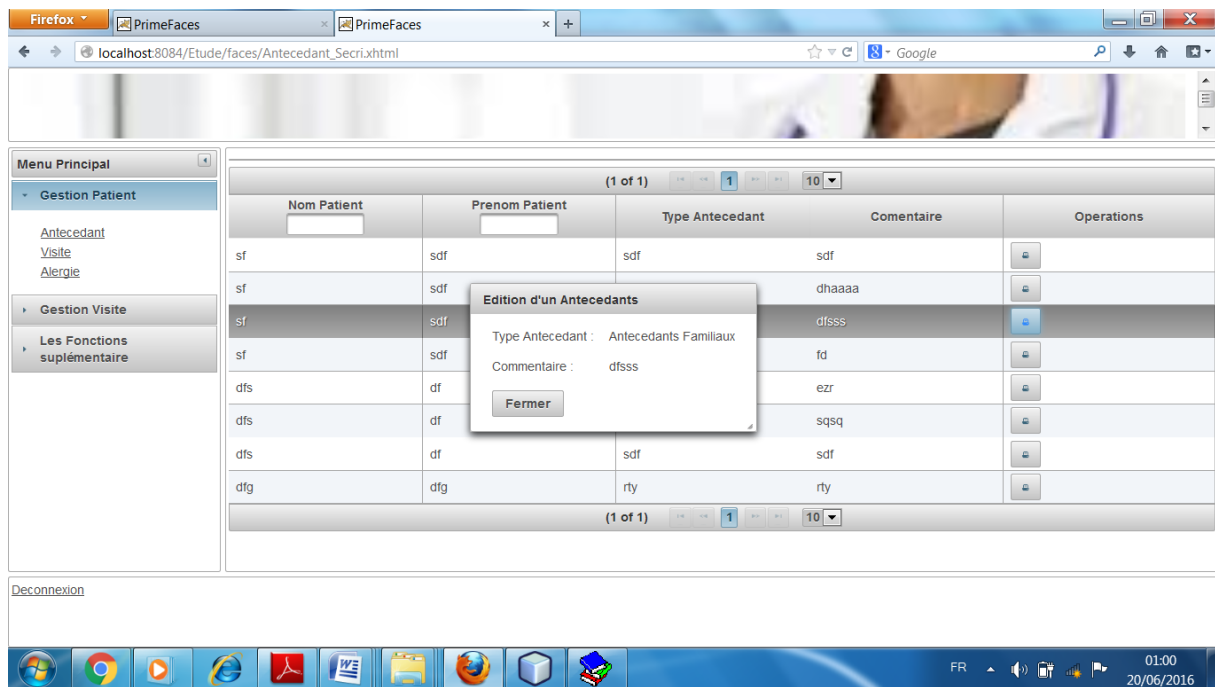


Figure IV.12. Interfaces du Secrétaire.

Conclusion :

Dans ce chapitre nous avons présenté les outils utilisés pour le développement de notre application, ensuite nous avons donné une description du fonctionnement de notre application en présentant ces différentes interfaces et les différents traitements que peut faire chaque utilisateur.

Conclusion générale

Le travail réalisé tout au long de notre stage au niveau de l'entreprise de L'ENIEM nous avons permis de découvrir le domaine médical et de voir sur le terrain l'ensemble des tâches effectuées par les professionnels de santé.

L'objectif de notre projet est la mise en œuvre d'une application permettant l'accès aux dossiers médicaux des patients, Ce travail nous a permis :

- ✓ D'acquérir une expérience dans la conception avec la notation UML.
- ✓ D'approfondir nos connaissances en matière de programmation JAVA.
- ✓ S'adapter à l'environnement de développement Netbeans.

Notre application qui a traité la gestion des dossiers médicaux, et qui adopte l'architecture Client-serveur 3-tiers offre une meilleure communication entre les professionnels de la santé, d'éviter des pertes de temps, d'améliorer la qualité des soins, finalement pour assurer une gestion plus rapide et plus efficace.

