

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**

**Ministère de l'enseignement et de la recherche scientifique**

**UNIVERSITE MOULOUD MAMMARI DE TIZI OUZOU**

**Faculté de Génie Electrique et Informatique**

**Département d'informatique**



# **Mémoire de Fin de Cycle**

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

*Option : conduite de projet informatique*

## **Thème**

**Conception et Implémentation**

**d'une Application Client/Serveur**

**Cas : gestion du dossier du malade pour la Consultation  
spécialisée CHU de Tizi-Ouzou**

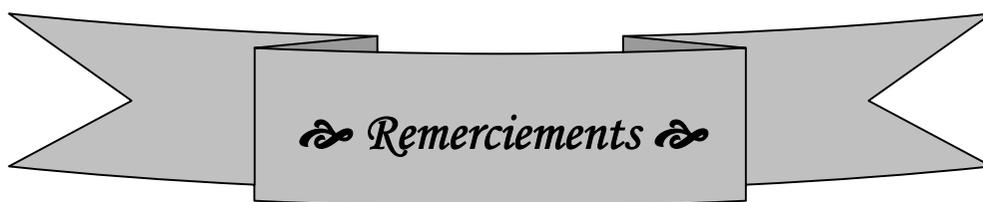
Réalisé par :

➤ **HAMMADI Samir**

Dirigé par :

**Mme: Sini**

**Promotion LMD 2013/2014**



*Je tiens à exprimer ma profonde gratitude à ma promotrice,  
Madame Sini, qui m'a fait l'honneur  
de diriger ce travail et ses précieux conseils furent d'un  
apport considérable.*

*Aussi je tiens à lui reconnaître le temps qu'elle m'a  
consacré.*

*Je tiens à remercier tous les membres du jury d'avoir accepté de  
juger notre travail.*

*Mes sincères sentiments vont à tous ceux qui, de près ou de  
loin, ont contribué à la réalisation de ce projet, en  
particulier ma famille et mes amis (es).*

*Samir*

*Dédicaces*

*Je dédie ce modeste travail :*

*A mes parents adorés*

*A mes frères et ma sœur*

*A mes grands parents*

*A toute ma famille*

*A tous mes ami(e)s*

.....*Samir*

# *Sommaire*

# Sommaire

1 .Introduction générale .....	01
<b>Chapitre I : l'organisme d'accueil</b>	
Introduction .....	02
1. Présentation .....	02
2. Historique du CHU de Tizi-Ouzou .....	02
3. Mission .....	03
3.1 en matière de santé .....	03
3.2 en matière de formation .....	03
3.3 en matière de recherche .....	03
4. Unités du CHU de Tizi-Ouzou.....	03
5. Champ d'étude .....	04
6. Présentation de la consultation spécialisée .....	04
7. Mission de la consultation spécialisée .....	05
8. Organigramme de la consultation spécialisée .....	05
9. Problématique .....	06
10. Objectifs .....	07
<b>Etude de l'existant :</b>	
1. Introduction .....	08
2. Liste des postes de travail.....	08
3. Fiche d'étude des postes de travail.....	09
4. Etude des documents :	
- 1. Introduction .....	12
- 2. Abréviations utilisées dans les types de données .....	12
- 3. Liste des documents.....	12
5. Critiques et suggestions :	
a. Critiques organisationnelles .....	16
b. Critiques informationnelles .....	16
c. suggestions .....	16
6. Proposition de la solution informatique :	
a. La solution informatique retenue .....	17
b. Les avantages de la solution.....	17
7. Conclusion .....	18

## Chapitre II : le client serveur

1. Introduction .....	19
2. Définition .....	19
3. Architecture réseaux .....	19
3.1. Architecture mainframe .....	19
3.1.1 Avantage .....	20
3.1.2. Inconvénients .....	20
3.2. Architecture d'égal à égal .....	20
3.2.1. Avantage de l'architecture d'égal à égal .....	21
3.2.2. Inconvénients des réseaux d'égal à égal .....	21
3.2.3. Mise en œuvre d'un réseau peer to peer .....	21
3.2.4. Administration d'un réseau poste a poste .....	22
3.2.5. La sécurité dans les réseaux poste a poste .....	22
3.3. Client serveur .....	22
3.3.1. Fonctionnement des applications client/serveur .....	23
3.3.2. Les objectifs du client/serveur .....	24
3.3.3. Type client .....	24
3.3.3.1 Par usage.....	24
3.3.3.2. Par ressource .....	25
3.3.4. Type de serveur .....	25
3.3.5. Architecture client/serveur .....	26
3.3.5.1 .Architecture deux tiers .....	26
3.3.5.1.1. Présentation .....	26
3.3.5.1.2. Limite du client serveur 2 tiers .....	27
3.3.5.2. L'architecture a 3 niveaux .....	28
3.3.5.2.1. Présentation .....	28
3.3.5.2.2 Limites du client-serveur trois-tiers .....	28
3.3.5.3 L'architecture multi niveaux .....	29
3.3.6 Conception d'une application client serveur .....	30
3.3.7 Le modèle OSI .....	30
3.3.7.1 Définition modèle OSI .....	30
3.3.7.2 Couches du modèle OS .....	31
3.3.8 Protocoles .....	31
3.3.8.1 Définition protocoles .....	32
3.3.8.2 Exemples à différents niveaux pour comprendre les protocoles .....	32

3.3.9 Le Middleware .....	32
3.3.9.1 Définition .....	32
3.3.9.2 Fonctions d'un Middleware .....	33
3.3.10 Client/serveur comment ça marche .....	34
3.3.11 Mode de communication client/serveur .....	34
3.3.11.1 Mode connecté .....	34
3.3.11.2 Mode non connecté.....	34
3.3.11.3 Points communs entre le client et le serveur .....	34
3.3.11.4 Caractéristiques mode connecté .....	35
3.3.11.5 Contraintes mode connecté.....	35
3.3.11.6 Caractéristiques mode non connecté .....	36
3.3.11.7 Contraintes mode non connecté.....	36
3.3.12 Avantages et inconvénients de l'architecture client/serveur.....	36
3.3.12.1 Avantages de l'architecture client/serveur .....	36
3.3.12.2 Inconvénients du modèle client/serveur .....	37
4. Conclusion .....	38

### **Chapitre III. Analyse et conception.**

1. Introduction .....	39
2. Présentation UML .....	39
2.1 Définition UML .....	39
2.2 Pourquoi modéliser .....	39
2.3 Historique d'UML .....	40
2.4 Les 9 Diagrammes .....	40
3. Les diagrammes à utiliser pour notre application .....	41
3.1 Liste des intervenants .....	42
3.2 Diagramme de cas d'utilisation .....	43
3.3 Diagramme de séquence .....	44
3.4 Diagramme d'état du dossier de malade .....	52
3.5 Diagramme d'activité .....	53
3.6. Dictionnaire de données épuré .....	62
3.7 Diagramme de classe .....	64
3.8 Identification des classes du diagramme .....	65
3.9 Diagramme de déploiement .....	66
4. Conclusion .....	67

## Chapitre IV : réalisation

1. Introduction .....	68
2. Présentation des outils de développement .....	68
2.1. Présentation du langage java .....	68
- Présentation le JDK .....	68
2.2. Oracle .....	68
2.3. Oracle SQL Developer .....	69
2.4. Eclipse.....	70
2.5. Apache Tomcat .....	71
3. Présentation des interfaces .....	71
4. Conclusion .....	80
Conclusion générale .....	81
Références bibliographiques.	

*Introduction*

*Générale*

## Introduction générale

Aujourd'hui, la relation entre l'entreprise et son système d'information est encore plus forte. Le système d'information est au cœur de l'activité de l'entreprise, et son efficacité en conditionne les performances.

Le SI fournit les données dont les dirigeants de l'entreprise ont besoin pour une exploitation efficiente et une gestion efficace.

Ainsi il contribue à réduire l'incertitude dans le pilotage de l'entreprise et à maîtriser la complexité des situations de gestion.

Suite à la demande des médecins de la consultation spécialisée d'une application de gestion des dossiers des malades poursuivre leurs patients, la responsable informatique m'a chargé de cette dernière.

Ce projet porte sur la conception et la réalisation d'un système d'information afin de répondre au différents besoins des utilisateurs.

Pour la réalisation de ce système d'information, nous avons opté à l'utilisation de l'architecture *Client-serveur* 3 tiers. La méthode MVC qui est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leur architecture respective. Et les outils suivants: Le serveur d'application apache tomcat, JDK, IDE eclips, oracle 11g et enfin IDE sqldeveloper.

Pour mener a bien ce travail, nous avons décompose ce mémoire en 4 chapitres qui sont :

- Chapitre I : Etude préalable;
- Chapitre II : Etude des réseaux ;
- Chapitre III : analyse et conception;
- Chapitre IV : réalisation ;

Et en finune conclusion générale.

# Chapitre 1 :

*Etude préalable*

**Introduction :**

L'étude préalable, construit une étape importante dans le développement d'un logiciel, elle permet d'étudier l'existant et de proposer des solutions et enfin de choisir une à développer.

**1. Présentation de l'organisme d'accueil :**

Le Centre Hospitalo-universitaire de Tizi-Ouzou est un établissement public à caractère administratif, doté de la personnalité morale et de l'autonomie financière. Il a été créé par décret exécutif, sur proposition conjointe du ministère chargé de la santé et du ministère chargé de l'enseignement supérieur et de la recherche scientifique. Il est placé sous la tutelle administrative du ministère chargé de la santé et la tutelle pédagogique du ministère chargé de l'enseignement supérieur.

Le Centre Hospitalo-universitaire de Tizi-Ouzou est caractérisé par son envergure régionale puisqu'il couvre quatre wilayas (Tizi-Ouzou, Bejaia, Boumerdes, et Bouira) dont la demande en soin est sans cesse croissante du fait de transitions démographique, épidémiologique que subit la région à l'instar des autres régions du pays mais aussi socioéconomique.

Centre de référence pour l'ensemble des établissements de santé des quatre wilayas, il offre aussi des soins spécialisés pour une population de 03 millions d'habitants. Il est par ailleurs le terrain de formation pour l'ensemble des étudiants en médecine et résidents relevant de l'université de Tizi-Ouzou, ainsi qu'à ceux émanant de l'école paramédicale de Tizi-Ouzou et depuis la rentrée universitaire 2003-2004 aux étudiants en pharmacie.

**2. Historique du CHU de TIZI-OUZOU :**

- L'hôpital de Tizi-Ouzou a été inauguré le 28 juillet 1955, il a été nommé l'hôpital civil NAEGLIN, dont la direction a été confiée à des français.
- A cette époque, l'hôpital comportait un nombre restreint de disciplines médicales.
- En 1974, l'hôpital régional de TIZI-OUZOU devient un secteur sanitaire grâce aux différentes unités de santé qui lui étaient reliées.
- En 1982, le secteur sanitaire de TIZI-OUZOU se voit transformer en secteur sanitaire universitaire (SSU) et ceci par l'ouverture d'une formation biomédicale pluridisciplinaire.
- En 1982, Le CHU de Tizi-Ouzou est né. Ce dernier est une institution à caractère administratif rattaché au ministère de la santé, crée par le décret N°86/25 du 11 février 1986, complété et modifié par le décret N°86/294 DU 16 décembre 1986.
- Le siège du CHU de TIZI-OUZOU est fixé à l'hôpital NEDIR MOHAMMED.

**3. Missions :**

Le CHU de TIZI-OUZOU est chargé, en relation avec l'établissement d'enseignement et de formation supérieure en sciences médicales concerné, des missions de diagnostic, d'exploration, de soins, de prévention, de formation, d'étude et de recherche.

**3.1 En matière de santé :**

Il est chargé :

- d'Assurer les activités de diagnostic, de soins, d'hospitalisation et des urgences médicochirurgicales, de prévention ainsi que de toute activité concourant à la protection et à la promotion de la santé de la population
- d'Appliquer les programmes nationaux, régionaux et locaux de santé
- de Contribuer à la promotion et à la protection de l'environnement dans les domaines relevant de la prévention, de l'hygiène, de la salubrité et de la lutte contre les nuisances et les fléaux sociaux.

Outre ces missions, le CHU assure pour la population résidant à proximité et non couverte par les secteurs sanitaires environnants, les missions dévolues aux secteurs sanitaires.

**3.2 En matière de formation:**

- Assurer, en collaboration avec l'établissement d'enseignement supérieur de formation en sciences médicales, la formation graduée et post-graduée en sciences médicales et de participer à l'élaboration et à la mise en œuvre des programmes y afférents.
- De participer à la formation, au recyclage et au perfectionnement des personnels de santé.

**3.3 En matière de recherche:**

- Effectuer, dans le cadre de la réglementation en vigueur des travaux d'études et de recherche dans le domaine des sciences de santé,
- Organiser des séminaires, colloques, journées d'études et autres manifestations techniques et scientifiques en vue de promouvoir les activités de soins, de formation et de recherche en science de santé.

**4. Unités du CHU de Tizi-Ouzou:**

Le CHU est constitué des unités suivantes :

- Unité NEDIR MOHAMMED.
- Unité de BELLOUA.
- Clinique dentaire.
- Médecine de travail.
- Unité des consultations spécialisées.

**5. Champ d'étude:**

Notre champ d'étude se limite à Unité des consultations spécialisées.

**6. Présentation de la consultation spécialisée :**

La Consultation spécialisée est un service de suivi des malades, doté de plusieurs types de consultations qui sont assurées par des médecins spécialistes.

C'est l'une des unités du centre hospitalo-universitaire de Tizi-Ouzou qui est caractérisé par son envergure régionale dont elle prend en charge quatre wilayas (Tizi-Ouzou, Bouira, Bejaia, Boumerdes).

L'unité des consultations spécialisées est dotée aussi d'un service « contact RDV » dont celui-ci est mis spécialement pour les citoyens afin de leur permettre de prendre des RDV à distance.

L'unité des consultations spécialisées comporte plusieurs services qui sont réparties comme suit :

**➤ Services de médecine :**

- Médecine interne.
- Cardiologie.
- Néphrologie.
- Hématologie.
- Maladies infectieuses.
- Endocrinologie diabétologie.
- Anesthésie réanimation.

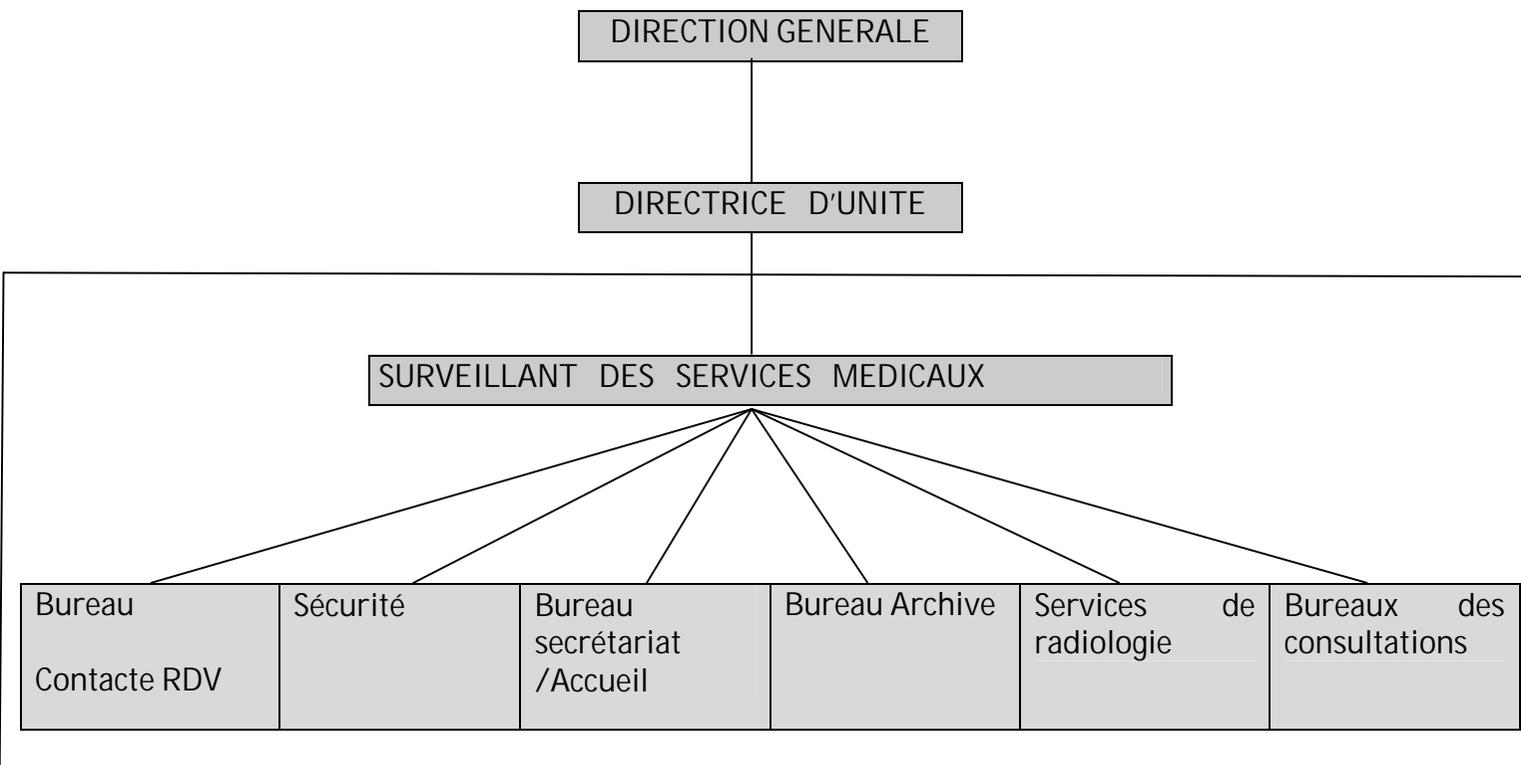
**➤ Services de chirurgie :**

- Orthopedie-traumatologie.
- Chirurgie générale.
- Chirurgie infantile.
- Urologie.
- Neurochirurgie.

**7. Mission de la consultation spécialisée :**

- Elle est chargée du suivi et de la prise en charge des malades ayant subi une intervention chirurgicale ou une hospitalisation dans l'un des services de chirurgie ou de médecine et cela en leur établissant des cartes ou des dossiers afin de les suivre jusqu'à leur guérison.
- Elle est chargée aussi du premier contact des nouveaux patients qui se présentent avec des lettres d'orientations vers des médecins spécialistes qui assureront leur suivi et les programmer pour d'éventuel prise en charge ou intervention chirurgicale.

**8. Organigramme de la consultation spécialisée :**



**Figure1.1 : Organigramme de la consultation spécialisée.**

: Domaine d'étude.

- **Bureau contact RDV** : s'occupent de la réception des appels téléphonique des patients et des attributions d'éventuelles RDV.
- **Sécurité** : permet d'intervenir en cas de conflit et d'effectuer des rondes de prévention afin d'assurer la sécurité du personnel et des patients.
- **Bureau secrétariat /accueil** : s'occupent de l'accueil et l'orientation du patient.
- **Bureau archive** : le respect de la confidentialité et veille au respect des diverse lois régissant le dossier et l'accès aux informations qui 'y sont contenues.
- **Service de radiologie** : diagnostiquer la maladie et réalise des prélèvements biopsies de tissus pour pouvoir les envoyer a un autre spécialiste (l'anatomo pathologiste), surveille les lésions en cours de traitement en particulier lors des lésions cancéreuse.
- **Bureaux des consultations** : consultation auprès du médecin pour divers problèmes de santé actuels.

### **9. Problématique:**

- Les secrétaires perdent beaucoup de temps lors de la recherche des dossiers des malades dans la salle d'archive.
- Les secrétaires ne peuvent pas trouver le dossier d'un malade qui a oublié sa carte de rendez vous vue que la recherche de dossier se fait seulement par numéro de dossier.
- Difficultés de savoir le nombre de rendez vous programmé par un service pour une journée donnée, D'ou la difficulté de programmer des rendez vous pour les nouveaux patients.
- Si les secrétaires font une grève, les médecins ne vont pas travailler car n'y a personne qui va leur remmener les dossiers des malades de la salle d'archive et ainsi tous les patients qui se sont déplacés, la plus part d'entre eux âgés, seront déçues vue qu'ils ne feront pas de consultations.
- Il est impossible pour un médecin d'un service donné de consulter les dossiers d'un patient des autres services afin d'avoir un maximum d'informations sur le patient.
- Difficultés de savoir le nombre de dossiers créés dans un service par année ou par trimestre ou par mois.

**10. Objectifs:**

- La recherche rapide d'un dossier de malade.
- Donner aussi la possibilité de rechercher un dossier de malade par nom ou prénom du patient si le patient à oublié sa carte de rendez vous ne posera pas de problème.
- Donner la possibilité aux médecins et secrétaires de voir les rendez vous programmés dans une journée par service cela va les aider à planifier les rendez vous.
- Permettre aux médecins de travailler même sans secrétaire et accéder rapidement au planning du jour.
- Si un patient possède plusieurs dossiers dans des services différents alors donner la possibilité aux médecins de consulter ces autre dossiers afin d'avoir le maximum d'informations.
- Donner la possibilité à l'administrateur de voir le nombre de dossiers créés dans chaque service soit par année ou par mois.

Pour atteindre ces objectifs et répondre à la problématique posée, il est impératif de bien étudier l'existant.

## **Introduction :**

Le but de l'étude de l'existant est d'analyser la situation actuelle et de collecter les informations manipulées par la consultation spécialisée.

## **2) Liste des postes de travail:**

L'étude des postes de travail concerne trois postes à savoir:

- L'administrateur.
- Secrétaire.
- Le médecin.

**Fiche d'étude du poste de travail N°1**

**Désignation** : administrateur.

**Nombre de personne** : 01.

**Taches et responsabilités:**

**Responsabilité** :

- répondre à l'urgence, d'assurer la permanence de l'administration et la continuité du fonctionnement des services.

**Tachesactuelles:**

- Gere lasécurité ;
- Transcrire un RDV ;
- Modifier un RDV ;
- Supprimé un RDV ;
- Consulter un RDV ;
- transcrire les informations de l'état civile d'un patient ;
- Modifier les informations de l'état civile d'un patient ;
- Rechercher un patient ;

**Fiche d'étude du poste de travail N°2**

**Désignation :** secrétaire ;

**Nombre de personne :** 03 ;

**Tâches et responsabilités:**

**Responsabilité:**

- l'accueil physique et téléphonique des patients;

**Tâches actuelles:**

- transcrire un RDV ;
- Modifier un RDV ;
- Supprimé un RDV ;
- Consulter un RDV ;
- transcrire les informations de l'état civil du patient ;
- Modifier les informations de l'état civil du patient ;
- Rechercher un patient ;

Fiche d'étude du poste de travail N° 3

**Désignation :** médecin ;

**Nombre de personne :** 20 ;

**Tâches et responsabilités:**

**Responsabilités:**

- Consultation dumalade ;

**Tâches actuelles:**

- Transcrire un RDV ;
- Supprime un RDV ;
- Modifier un RDV ;
- Consulte un RDV ;
- Consulter un dossier malade ;

## **Etude des documents:**

### **1-Introduction:**

Cette étude permet de faire un inventaire de tous les documents utilisés par notre domaine d'étude ;

### **2-Abréviation utilisée dans les types de données :**

A : Alphabétique ;

N : Numérique ;

AN : Alphanumérique ;

D : Date ;

### **3 .Liste des documents :**

**Document n°01** : carte de RDV ;

**Document n°02** : Dossier de Malade ;

**Document n°03** : fiche technique ;

**Fiche d'étude du document N°1**

**Code :** carte D RDV ;

**Désignation :** Carte de RDV ;

**Nature:** /

**Nombre d'exemplaire :** 1 ;

**Rôle :** pour ne pas oublier le RDV et être à l'heure.

**En-tête :** CENTRE HOSPITALO-UNIVERSITAIRE DE TIZI-OUZOU.

**Contenu du document :**

<b>Rubrique</b>	<b>Type</b>	<b>Taille</b>	<b>Observation</b>
- Nom patient	<b>A</b>	<b>15</b>	
- Prénom patient	<b>A</b>	<b>15</b>	
- Age patient	<b>AN</b>	<b>03</b>	
- Domicile patient	<b>AN</b>	<b>25</b>	
- Dossier N° patient	<b>AN</b>	<b>07</b>	

## Fiche d'étude du document N°2

**Code :** Doss de Mal ;

**Désignation:** Dossier de Malade;

**Nombre d'exemplaire :** 01 ;

**Rôle :** il constitue une partie du dossier d'hospitalisation qui est un document permettant la connaissance du malade, regroupant toutes les données administratives et médicales, les informations relatives aux soins, traitements, examens ou interventions pratiques

**En-tête :** CENTRE Hospitalo-Universitaire de TIZI-OUZOU

**Continue du document:**

Rubrique	Type	Taille	Observation
- Nom Service	AN	25	
- Nom du patient	A	15	
- Prénom du patient	A	15	
- Profession du patient	A	20	
- Date de naissance du patient	D	10	JJ/MM/AAAA
- Sexe du patient	A	10	F O U M
- Lieu de naissance du patient	AN	30	
- Domicile du patient	AN	25	
- Salle	AN	15	
- Date d'entrée le	D	10	JJ /MM /AAAA
- Date de Sortie le	D	10	JJ /MM /AAAA
- Hospitalisation	A	20	
- Adressé par	A	20	
- Diagnostique	AN	60	
- Etat a l'entrée	AN	50	
- Etat a la sortie	AN	50	
- Traitements a la sortie	AN	80	
- Date de l'examen	D	10	JJ/MM/AAAA
- Nature de l'examen	AN	60	
- Résultats de l'examen	AN	60	

**Fiche D'étude Du Document N°3**

**Code :** fiche thec ;

**Désignation :** Fiche technique ;

**Nombre d'exemplaire :** 01 ;

**Rôle :** fiche thérapeutique du patient.

**Entête :** CENTRE HOSPITALO-Universitaire de TIZI-OUZOU.

**Contenu du document :**

<b>Rubrique</b>	<b>Type</b>	<b>Taille</b>	<b>Observation</b>
- Nom Service	<b>AN</b>	<b>20</b>	
- Nom patient	<b>A</b>	<b>15</b>	
- Prénom patient	<b>A</b>	<b>15</b>	
- Date de naissance patient	<b>D</b>	<b>10</b>	JJ/MM/AAAA
- Lieu	<b>AN</b>	<b>30</b>	
- Adresse du patient	<b>AN</b>	<b>20</b>	
- Profession patient	<b>A</b>	<b>25</b>	
- Médecin consultant	<b>A</b>	<b>20</b>	
- Date	<b>D</b>	<b>10</b>	JJ/MM/AAAA
- Observation et traitement	<b>AN</b>	<b>60</b>	

**Critiques et suggestions:**

Après l'analyse de l'existant menée au sein de la consultation spécialisée nous avons constaté les insuffisances et les anomalies pour les quelles nous avons apporté des suggestions ci-après.

**a) Critiques organisationnelles :**

- Absence de traçabilité des opérations (ex : en cas de perte de documents).
- Aucune sécurité des informations en cas de destruction accidentelle ou volontaire.

**b) Critiques informationnelles :**

- perte de temps dans la recherche de l'information à cause de l'archivage volumineux
- en amont vis-à-vis du personnel qui devra avoir les informations nécessaires pour l'accomplissement de leurs tâches au sein de la consultation spécialisée et pour un bon fonctionnement des services.
- Une mauvaise circulation de l'information peut donner lieu à des problèmes de prise en charge du patient notamment en période de crise, de pandémie.
- Manque de communication entre les trois acteurs à savoir le médecin l'administrateur et le secrétaire.

**c) Suggestions:**

- Enregistré le mouvement et la mobilité de chaque document.
- Garder les copie sur des supports magnétique et garder les copies dans des lieux sure.
- Edition automatique des documents importants.
- Mise en place des outils (informatique) pour assuré la communication interpersonnel.

**Proposition de la solution informatique :**

Notre solution consiste à mettre en place une application informatique au niveau de la consultation spécialisée qui répondra aux insuffisances du système existant en respectant les suggestions organisationnelles, informationnelles.

**a) La solution informatique retenue :**

Cette solution consiste à installer une application client /serveur avec deux ordinateurs pour chacun des deux postes (le médecin, le secrétaire) qui seront des clients, reliés à un serveur qui se situera au niveau de l'administrateur par un réseau local.

**b) Les avantages de la solution :**

- Automatisation des tâches effectuées par chaque poste de travail.
- Sécurité et confidentialité des informations.
- Rapidité et précision d'exécution des traitements.

**7. Conclusion :**

Dans ce chapitre nous avons d'abord présenté l'organisme d'accueil qui est le CHU de TIZI-OUZOU ensuite, nous avons présenté la consultation spécialisée. Après ceci nous avons posé la problématique et les objectifs. L'étude de l'existant nous a permis de comprendre le fonctionnement de notre champ d'étude ainsi de pouvoir dégager les anomalies existantes et de proposer une solution informatique répondant à ces anomalies. Dans le chapitre qui suivra nous parlerons sur les réseaux.

# Chapitre 2 :

## Etude des réseaux

## 1. Introduction :

Client et serveur, voici deux mots que nous pouvons rencontrer dans la vie courante. Dans un café, par exemple, Un client est une personne qui demande quelque chose au serveur: le client demande un café au serveur, qui lui apporte. En informatique, le principe est le même: un client va demander quelque chose au serveur.

## 2. Définitions :

- ✓ **Client** : processus demandant l'exécution d'une opération à un autre processus par envoi d'un message contenant le descriptif de l'opération et attendant la réponse à cette opération par un message en retour.
- ✓ **Serveur** : processus accomplissant une opération sur demande d'un client et transmettant la réponse à ce dernier.
- ✓ **CLIENT SERVEUR** : Architecture de réseau dans laquelle toutes les informations sont localisées sur des ordinateurs ayant le rôle de serveur, et accessible aux ordinateurs clients.
- ✓ **Requête** : message transmis par un client à un serveur décrivant l'opération à exécuter pour le compte du client.

## 3. Architectures réseaux :

### 3.1. Architecture mainframe :

Avant que n'apparaisse le mode client-serveur, les réseaux informatiques étaient configurés autour d'un ordinateur central appelé mainframe auquel étaient connectés des terminaux passifs (écran adjoint d'un clavier sans unité centrale). Tous les utilisateurs sont alors connectés sur la même unité centrale.

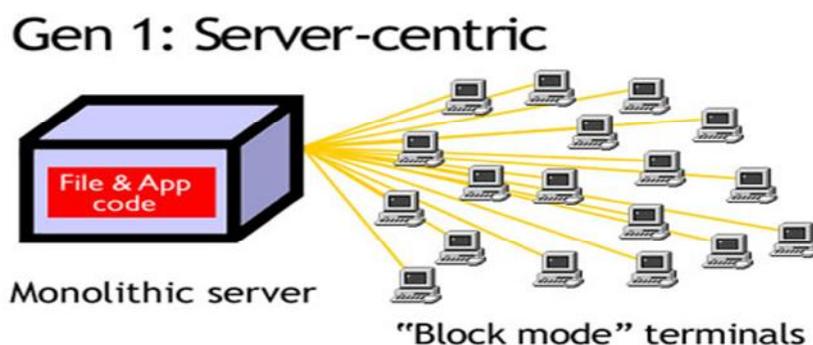


Figure 2.1 : Architecture mainframe.

Le **mainframe** n'affiche que du texte à l'écran sans graphisme (pas de bouton, pas de fenêtre,...). Il est spécialisé dans la gestion d'informations de masse auquel il peut appliquer des instructions simples (addition, soustraction...) mais avec une grande vélocité. Ainsi, plusieurs milliers de personnes peuvent travailler sur cette unité centrale sans ralentissement.

Aujourd'hui, les anciens terminaux passifs ont été remplacés par des émulations logicielles installées sur les PC.

### 3.1.1 Avantages :

- ✓ Gestion des données et des traitements centralisée.
- ✓ Maintenance matériel minime.
- ✓ Grande vélocité sur des grands volumes de données et de traitements.

### 3.1.2 Inconvénients :

- ✓ Interface homme-machine minimaliste.
- ✓ Utilisation de langages de programmation anciens.
- ✓ Calcul scientifique complexe impossible.

### 3.2 L'architecture d'égal à égal :

Dans une architecture d'**égal à égal** (en anglais *peer to peer*), contrairement à une architecture de réseau de type client/serveur, il n'y a pas de serveur dédié. Ainsi chaque ordinateur dans un tel réseau est un peu serveur et un peu client. Cela signifie que chacun des ordinateurs du réseau est libre de partager ses ressources. Un ordinateur relié à une imprimante pourra donc éventuellement la partager afin que tous les autres ordinateurs puissent y accéder via le réseau.

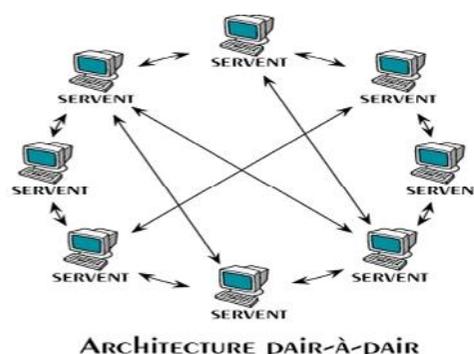


Figure2.2 : Architecture égal à égal.

### 3.2.1 Avantages de l'architecture d'égal à égal :

L'architecture d'égal à égal a tout de même quelques avantages parmi lesquels :

- ✓ un coût réduit (les coûts engendrés par un tel réseau sont le matériel, les câbles et la maintenance)
- ✓ une simplicité à toute épreuve!

### 3.2.2 Inconvénients des réseaux d'égal à égal :

Les réseaux d'égal à égal ont énormément d'inconvénients :

- ✓ ce système n'est pas du tout centralisé, ce qui le rend très difficile à administrer.
- ✓ la sécurité est très peu présente.
- ✓ aucun maillon du système n'est fiable.

Ainsi, les réseaux d'égal à égal ne sont valables que pour un petit nombre d'ordinateurs (généralement une dizaine), et pour des applications ne nécessitant pas une grande sécurité (il est donc déconseillé pour un réseau professionnel avec des données sensibles).

### 3.2.3 Mise en œuvre d'un réseau peer to peer :

Les réseaux poste à poste ne nécessitent pas les mêmes niveaux de performance et de sécurité que les logiciels réseaux pour serveurs dédiés. On peut donc utiliser Windows NT Workstation, Windows pour Workgroups ou Windows 95 car tous ces systèmes d'exploitation intègrent toutes les fonctionnalités du réseau poste à poste.

La mise en œuvre d'une telle architecture réseau repose sur des solutions standards :

- ✓ Placer les ordinateurs sur le bureau des utilisateurs.
- ✓ Chaque utilisateur est son propre administrateur et planifie lui-même sa sécurité.
- ✓ Pour les connexions, on utilise un système de câblage simple et apparent.

Il s'agit généralement d'une solution satisfaisante pour des environnements ayant les caractéristiques suivantes :

- ✓ Moins de 10 utilisateurs.

- ✓ Tous les utilisateurs sont situés dans une même zone géographique.
- ✓ La sécurité n'est pas un problème crucial.
- ✓ Ni l'entreprise ni le réseau ne sont susceptibles d'évoluer de manière significative dans un proche avenir.

### 3.2.4 Administration d'un réseau poste à poste :

Le réseau poste à poste répond aux besoins d'une petite entreprise mais peut s'avérer inadéquat dans certains environnements. On désigne par le terme **Administration** :

- ✓ Gestion des utilisateurs et de la sécurité ;
- ✓ Mise à disposition des ressources ;
- ✓ Maintenance des applications et des données ;
- ✓ Installation et mise à niveau des logiciels utilisateurs.

Dans un réseau poste à poste typique, il n'y a pas d'administrateur. Chaque utilisateur administre son propre poste. D'autre part tous les utilisateurs peuvent partager leurs ressources comme ils le souhaitent (données dans des répertoires partagés, imprimantes, cartes fax etc.)

### 3.2.5 La sécurité dans les réseaux poste à poste :

La politique de sécurité minimale consiste à mettre un mot de passe à une ressource. Les utilisateurs d'un réseau poste à poste définissent leur propre sécurité et comme tous les partages peuvent exister sur tous les ordinateurs, il est difficile de mettre en œuvre un contrôle centralisé.

## 3.3 Client serveur :

On peut essayer de préciser ce que l'on entend par modèle client-serveur, bien en donnant les caractéristiques suivantes :

**Service** : Le serveur est fournisseur de services. Le client est consommateur de services.

**Protocole** :

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

**Partage des ressources :**

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

**Localisation :**

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

**Hétérogénéité :**

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

**Redimensionnement :**

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

**Intégrité :**

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

**Souplesse et adaptabilité :**

On peut modifier le module serveur sans toucher au module client. La réciproque est vraie. Si une station est remplacée par un modèle plus récent, on modifie le module client (en améliorant l'interface, par exemple) sans modifier le module serveur.

**3.3.1 Fonctionnement des applications client-serveur :**

C'est l'application cliente qui prend l'initiative du dialogue, le programme dit «Client» demande, par l'envoi d'un message, un service extérieur à un autre programme dit «Serveur». Une fois le service rendu, le programme «Serveur» renvoi un message au programme «Client».

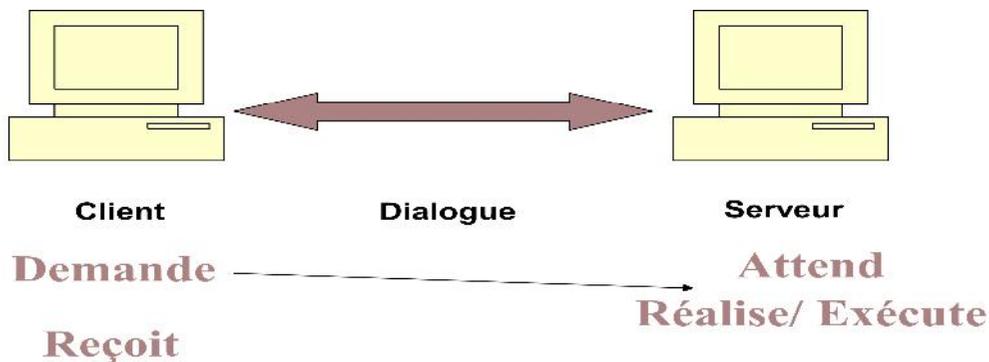


Figure 2.3 : Fonctionnement client-serveur.

### 3.3.2 Les objectifs du client-serveur :

Répartir les tâches entre le client et le serveur :

- ✓ décharger le serveur de l'exploitation des données.
- ✓ décharger la station cliente de la gestion des données.
- ✓ réduire le trafic sur le réseau.

L'idée du client-serveur est d'associer les demandes, l'informatique centralisée (le monde des mainframes) et la micro-informatique (le monde des PC), pour que l'utilisateur dispose d'une fenêtre unique sur les systèmes d'informations de son organisation. Le système informatique tire ainsi parti de la convivialité du micro-ordinateur et de la puissance des systèmes centraux.

### 3.3.3 Type Client :

#### 3.3.3.1 Par usage : on a

- ✓ **Un client de messagerie** : un logiciel qui sert à lire et envoyer des courriers électroniques.
- ✓ **Un client http (navigateur web)** : un logiciel conçu pour se connecter à un serveur http. Dans le cas le plus courant, il s'agit d'un navigateur Web.
- ✓ **Client IRC (chat)** : un logiciel pour se connecter à un serveur irc et dialoguer avec d'autres internautes.

- ✓ **Client SSH (Secure Shell)** : Il est à la fois un programme informatique et un protocole de communication sécurisé.
- ✓ **Client Jabber/XMPP (messagerie instantanée)** : C'est un protocole basé sur une architecture client/serveur permettant les échanges décentralisés de messages instantanés ou non, entre clients, au format XML.

### 3.3.3.2 Par ressource :

- ✓ **Un client léger** : c'est un ordinateur qui, dans une architecture client serveur, n'a presque pas de logique d'application. Il dépend donc surtout du serveur central pour le traitement.
- ✓ **Un client lourd** : c'est un logiciel qui propose des fonctionnalités complexes avec un traitement autonome. La notion de client s'entend dans une architecture client serveur. Et contrairement au client léger, le client lourd ne dépend du serveur que pour l'échange des données dont il prend généralement en charge l'intégralité du traitement.

Les solutions client lourd sont également caractérisées comme étant des solutions très coûteuses tant au niveau de la maintenance que du déploiement et de la formation.

- ✓ **Client riche** : applications effectuant une grande partie du traitement en JavaScript, donc côté client.

### 3.3.4 Type de serveurs :

- ✓ **Le serveur de fichiers** : est utilisé pour le stockage et le partage de fichiers. Les fichiers placés dans les mémoires de masse du serveur peuvent être manipulés simultanément par plusieurs clients ;
- ✓ **Le serveur d'impression** : est utilisé comme intermédiaire entre un ensemble de clients et un ensemble d'imprimantes. Chaque client peut envoyer des documents à imprimer aux imprimantes reliées au serveur ;
- ✓ **Le serveur de base de données** : est utilisé pour stocker et manipuler des données contenues dans une ou plusieurs bases de données et partagées entre plusieurs clients ;

- ✓ **Le serveur de courrier messagerie** : est utilisé pour stocker et transmettre du courrier électronique. Le courrier partant est transmis au serveur du destinataire. Le courrier arrivé est stocké sur le serveur et est consultable par le client ;
- ✓ **Le serveur web** : serveur http stocke et manipule les pages d'un site web et les transmet sur demande au client (voir navigateur web) ;
- ✓ **Le serveur de jeu** : arbitre et suit l'évolution d'un jeu. Il met en communication les différents joueurs d'un jeu en ligne multi-joueurs ;
- ✓ **Le serveur d'applications** : est un terme générique pour désigner un serveur qui effectue les traitements d'un ou plusieurs logiciels applicatif à architecture client/serveur ;
- ✓ **Le serveur de nom** : permet d'établir la correspondance entre le nom de domaine et l'adresse IP des machines d'un réseau ;
- ✓ **Le serveur mandataire** : reçoit des demandes, les contrôle, puis les transmet à d'autres serveurs.

### 3.3.5 Architecture client-serveur :

#### 3.3.5.1 L'architecture deux tiers :

##### 3.3.5.1.1 Présentation :

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

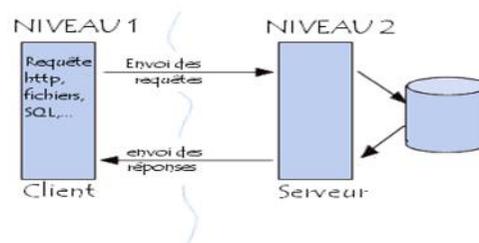


Figure 2.4 : L'architecture deux tiers.

### 3.3.5.1.2 Limites du client-serveur deux tiers :

L'expérience a démontré qu'il était coûteux et contraignant de vouloir faire porter l'ensemble des traitements applicatifs par le poste client. On en arrive aujourd'hui à ce que l'on appelle le client lourd, ou fat client.

L'architecture client-serveur de première génération s'est heurtée à ce constat à l'heure des premiers bilans :

- ✓ on ne peut pas soulager la charge du poste client, qui supporte la grande majorité des traitements applicatifs,
- ✓ le poste client est fortement sollicité, il devient de plus en plus complexe et doit être mis à jour régulièrement pour répondre aux besoins des utilisateurs,
- ✓ la conversation entre client et serveur est assez bruyante et s'adapte mal à des bandes passantes étroites. De ce fait, ce type d'application est souvent cantonné au réseau local de l'entreprise,
- ✓ les applications se prêtent assez mal aux fortes montées en charge car il est difficile de modifier l'architecture initiale,
- ✓ la relation étroite qui existe entre le programme client et l'organisation de la partie serveur complique les évolutions de cette dernière,
- ✓ ce type d'architecture est grandement rigidifié par les coûts et la complexité de sa maintenance.

Malgré tout, l'architecture deux tiers présente de nombreux avantages qui lui permettent de présenter un bilan globalement positif :

- ✓ elle permet l'utilisation d'une interface utilisateur riche,
- ✓ elle a permis l'appropriation des applications par l'utilisateur,
- ✓ elle a introduit la notion d'interopérabilité.

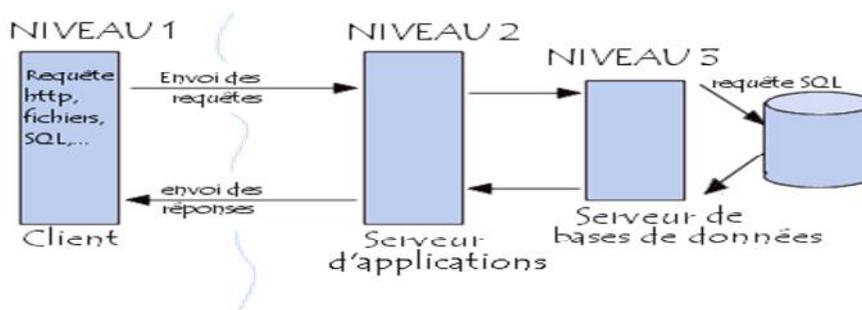
Pour résoudre les limitations du client-serveur deux tiers tout en conservant ses avantages, on a cherché une architecture plus évoluée, facilitant les forts déploiements à moindre coût. La réponse est apportée par les architectures distribuées.

### 3.3.5.2 L'architecture à 3 niveaux :

#### 3.3.5.2.1 Présentation :

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

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



**Figure 2.5 : L'architecture trois tiers.**

L'architecture trois tiers applique les principes suivants :

- ✓ les données sont toujours gérées de façon centralisée,
- ✓ la présentation est toujours prise en charge par le poste client,
- ✓ la logique applicative est prise en charge par un serveur intermédiaire.

#### 3.3.5.2.2 Limites du client-serveur trois-tiers :

L'architecture trois tiers a corrigé les excès du client lourd en centralisant une grande partie de la logique applicative sur le serveur. Le poste client, qui ne prend à sa charge que la présentation et les contrôles de saisie, s'est trouvé ainsi soulagé et plus simple à gérer. Par contre, le serveur constitue la pierre angulaire de l'architecture et se trouve souvent fortement

sollicité et il est difficile de répartir la charge entre client et serveur. De plus, les solutions mises en œuvre sont relativement complexes à maintenir et la gestion des sessions est compliquée. Les contraintes semblent inversées par rapport à celles rencontrées avec les architectures deux tiers : le client est soulagé, mais le serveur est fortement sollicité.

Le juste équilibrage de la charge entre client et serveur semble atteint avec la génération suivante : les architectures n-tiers.

### **3.3.5.3 L'architecture multi niveaux :**

L'architecture n-tiers a été pensée pour pallier aux limitations des architectures trois tiers et concevoir des applications puissantes et simples à maintenir. Ce type d'architecture permet de distribuer plus librement la logique applicative, ce qui facilite la répartition de la charge entre tous les niveaux.

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

Théoriquement, ce type d'architecture supprime tous les inconvénients des architectures précédentes

- ✓ elle permet l'utilisation d'interfaces utilisateurs riches,
- ✓ elle sépare nettement tous les niveaux de l'application,
- ✓ elle offre de grandes capacités d'extension,
- ✓ elle facilite la gestion des sessions.

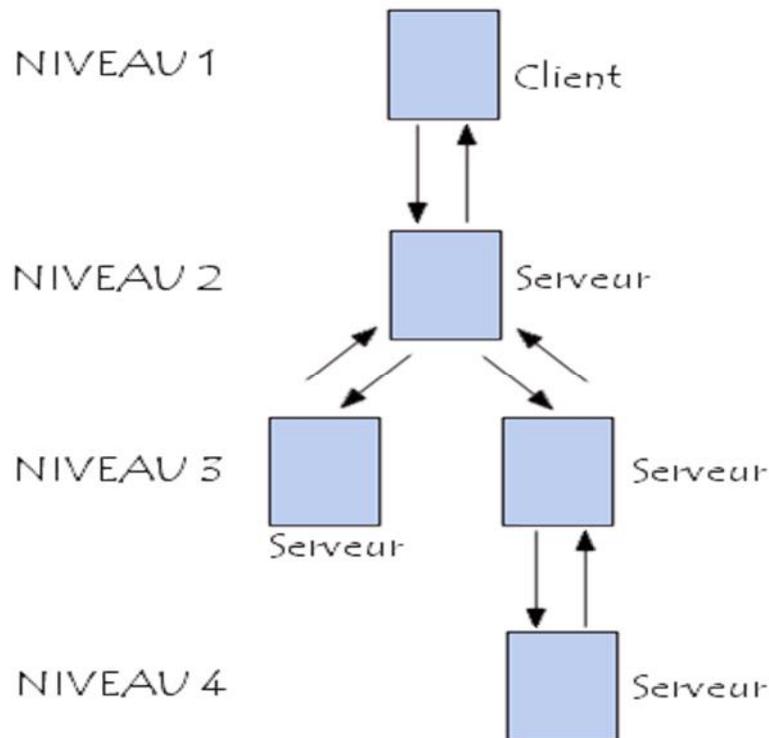


Figure 2.6 : L'architecture multi niveaux.

### 3.3.6 Conception d'une application client serveur :

Dans une application client serveur, il faut décider de l'emplacement des composantes de :

- ✓ **présentation** : interfaces textuelles ou graphiques, interactions, entrée des données, validation, etc.
- ✓ **logique d'application** : traitements associés à l'application.
- ✓ **accès aux données**: stockage et accès aux données (base de données, serveur web, ect).

### 3.3.7 Le modèle OSI :

#### 3.3.7.1 Définition modèle OSI :

Open Systems Interconnection est un standard de communication, en réseau, de tous les systèmes informatiques. C'est un modèle de communications entre ordinateurs proposé par l'ISO qui décrit les fonctionnalités nécessaires à la communication et l'organisation de ces fonctions.

### 3.3.7.2 Couches du modèle OSI

Les différentes couches du modèle OSI sont définies ainsi :

1. **Physique** : chargé de la transmission effective des signaux électriques ou optiques entre les interlocuteurs.
2. **Liaison** : gère la communication entre 2 machines adjacentes, c'est-à-dire directement reliées entre elles par un support physique.
3. **Réseau** : fonction de routage et de relayage.
4. **Transport** : gère les communications de bout en bout entre les processus.
5. **Session** : synchronisation des communications et gestion des transactions.
6. **Présentation** : codage des données applicatives.
7. **Application** : point d'accès aux services réseaux.

### Les 7 couches du modèle OSI

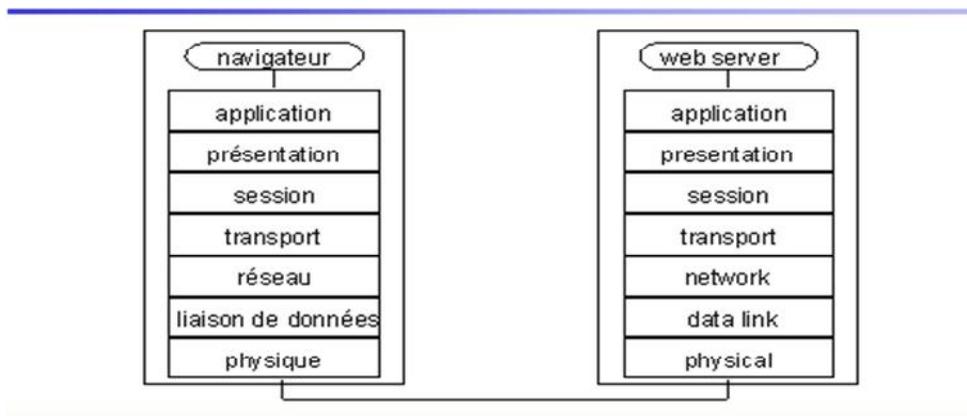


Figure 2.7: Modèle OSI.

### 3.3.8 Protocoles :

A tous les niveaux il ya des protocoles Communiquer entre humains / programmes / machines.

### 3.3.8.1 Définition protocoles :

Ensemble de règles et de procédures à respecter pour pouvoir échanger des données sur un réseau.

**Remarque :** exemple de la vie courante :

Ca va ? (→) / Oui (←) / Je n'ai pas entendu (→) / Je répète : « Oui » (←)

### 3.3.8.2 Exemples à différents niveaux pour comprendre les protocoles :

#### ✓ Niveau programmes (A – B) :

Tu fonctionnes ? (→) / Oui (←) / Envoie-moi le fichier toto.doc (→) / Ok c'est parti (←) / toto.doc (←) / Bien reçu (→) / Au revoir (←)

#### ✓ Niveaux système d'exploitation :

Toi, système d'exploitation de la machine machine1.univ-lyon1.fr, passe le message «Tu fonctionnes?» au programme qui écoute sur le port 3422 (→)

#### ✓ Niveau cartes :

Toi carte réseau, regarde passer des paquets de données sur le câble, attrape ceux qui sont pour toi, et passe-en le contenu au système d'exploitation.

### 3.3.9 Le Middleware :

#### 3.3.9.1 Définition :

C'est un ensemble de services logiciels construits au dessus d'un protocole de transport afin de permettre l'échange de requête/réponse entre le client et le serveur de manière transparente.

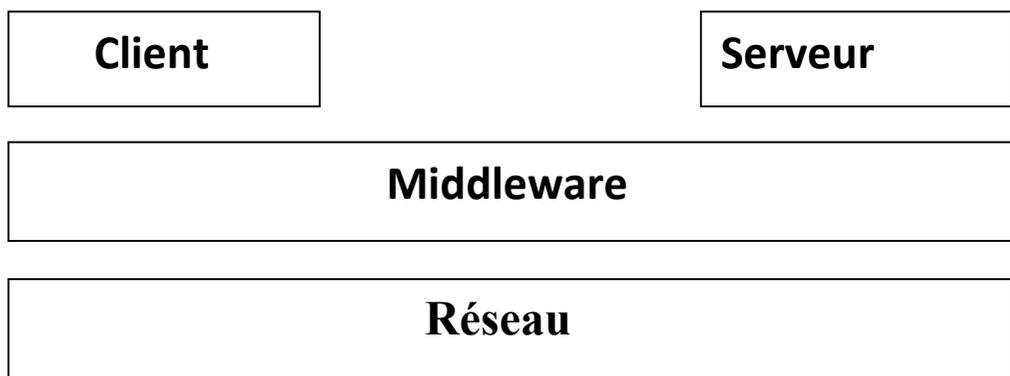


Figure 2.8 : Le Middleware.

Le Middleware s'appuie côté client et côté serveur sur :

• **FAP (Format And Protocols) pilote les échanges à travers le réseau :**

- ✓ synchronisation des échanges selon un protocole de communication
- ✓ mise en forme des données échangées selon un format connu de part et d'autre

• **API (Application Programming Interfaces) :**

Les fonctions encapsulées dans l'API permettent à l'application de faire appel aux services proposés par le serveur.

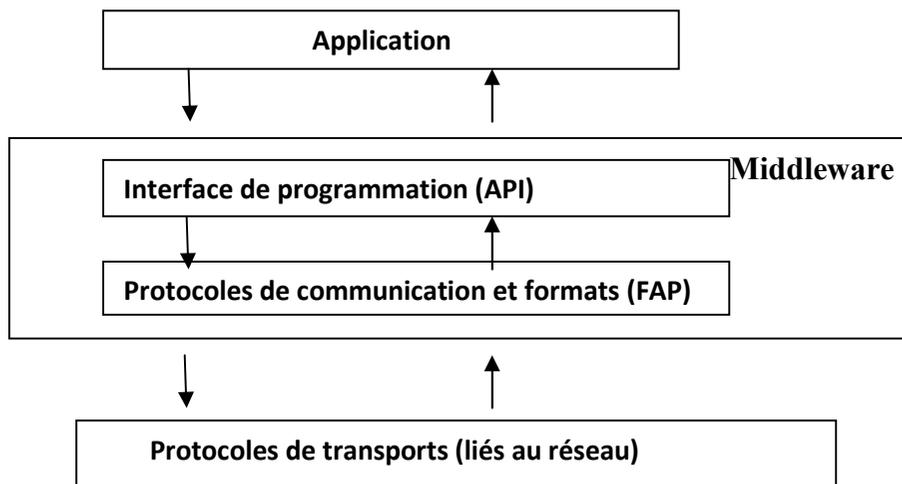


Figure 2.9: Inter Process Communication Middleware.

**3.3.9.2 Fonctions d'un Middleware :**

- ✓ Procédures d'établissement/fermeture de connexion.
- ✓ Exécution des requêtes, récupération des résultats.
- ✓ Initiation des processus sur différents sites.
- ✓ Services de répertoire.
- ✓ Accès aux données à distance.
- ✓ Gestion d'accès concurrents.
- ✓ Sécurité et intégrité (authentification, cryptage, ...).
- ✓ Monitoring (compteurs, ...).
- ✓ Terminaison de processus.
- ✓ Mise en cache des résultats, des requêtes.

**3.3.10 Client-serveur comment ça marche ?****Les ports :**

- ✓ Coté serveur: chaque serveur (application) ouvre un port de communication identifié par un n°.
- ✓ Coté client: chaque client (application) ouvre un port de communication identifié par un n°.
- ✓ La communication s'établit entre deux application identifiées (ports) sur deux machines identifiées (@IP).

**3.3.11 Mode de communication client serveur :****Deux réalisations possibles :**

- ✓ Mode connecté (protocole TCP).
- ✓ Mode non connecté (protocole UDP).

**3.3.11.1 Mode connecté :**

- ✓ Ouverture d'une liaison, suite d'échange, fermeture de la liaison.
- ✓ Le serveur préserve son état entre deux requêtes.
- ✓ Garanties de TCP : ordre, contrôle de flux, fiabilité.
- ✓ Adapté aux échanges ayant une certaine durée (plusieurs messages).

**3.3.11.2 Mode non connecté :**

- ✓ Les requêtes successives sont indépendantes.
- ✓ Pas de préservation de l'état entre les requêtes.
- ✓ Le client doit indiquer son adresse à chaque requête (pas de liaison permanente).
- ✓ Pas de garanties particulières.
- ✓ Adapté aux échanges brefs (réponse en 1 message) et pour faire de la diffusion.

**3.3.11.3 Points communs entre le client et le serveur :**

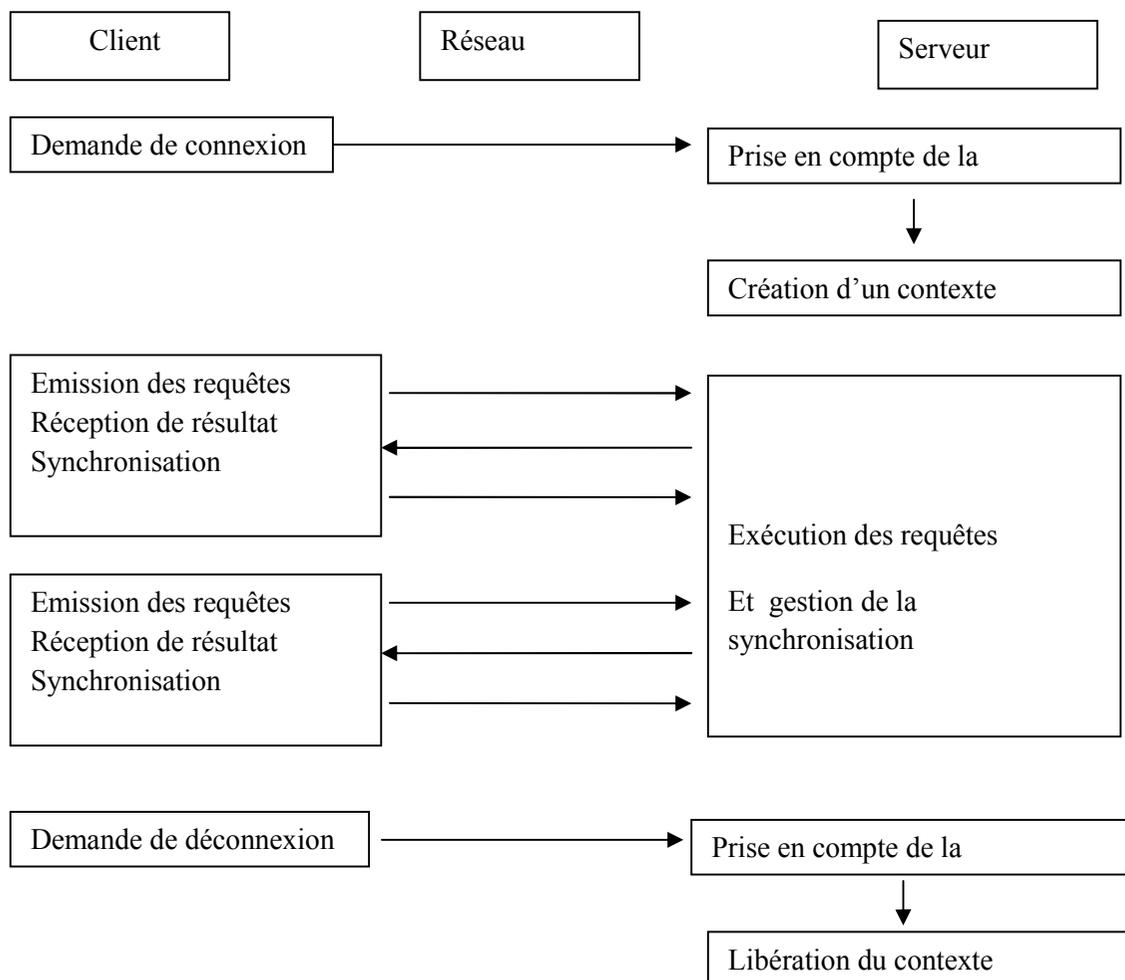
- ✓ Le client à l'initiative de la communication et le serveur doit être à l'écoute.
- ✓ Le client doit connaître la référence du serveur (adresse IP, numéro de port).
- ✓ Le serveur peut servir plusieurs clients.

**3.3.11.4 Caractéristiques mode connecté :**

- ✓ Etablissement préalable d'une connexion (circuit virtuel) : le client demande au serveur s'il accepte la connexion.
- ✓ Fiabilité assurée par le protocole (TCP).
- ✓ Possibilité d'émettre et de recevoir des caractères urgents.
- ✓ Après initialisation, le serveur est passif : il est activé lors de l'arrivée d'une demande de connexion du client.
- ✓ Mode d'échange par flots d'octets : le récepteur n'a pas connaissance du découpage des données effectué par l'émetteur.
- ✓ Un serveur peut répondre aux demandes de service de plusieurs clients : les requêtes arrivées et non traitées sont stockées dans une file d'attente.

**3.3.11.5 Contraintes mode connecté :**

Le client doit avoir accès à l'adresse du serveur (adresse IP, numéro de port).



**Figure 2.10 : Mode connecté**

### 3.3.11.6 Caractéristiques mode non connecté :

- ✓ Pas d'établissement préalable d'une connexion.
- ✓ Pas de garantie de fiabilité.
- ✓ Adapté aux applications pour lesquelles les réponses aux requêtes des clients sont courtes (1 message).
- ✓ Le récepteur reçoit les données selon le découpage effectué par l'émetteur.

### 3.3.11.7 Contraintes mode non connecté :

- ✓ Le client doit avoir accès à l'adresse du serveur (adresse IP, numéro de port).
- ✓ Le serveur doit récupérer l'adresse de chaque client pour lui répondre.

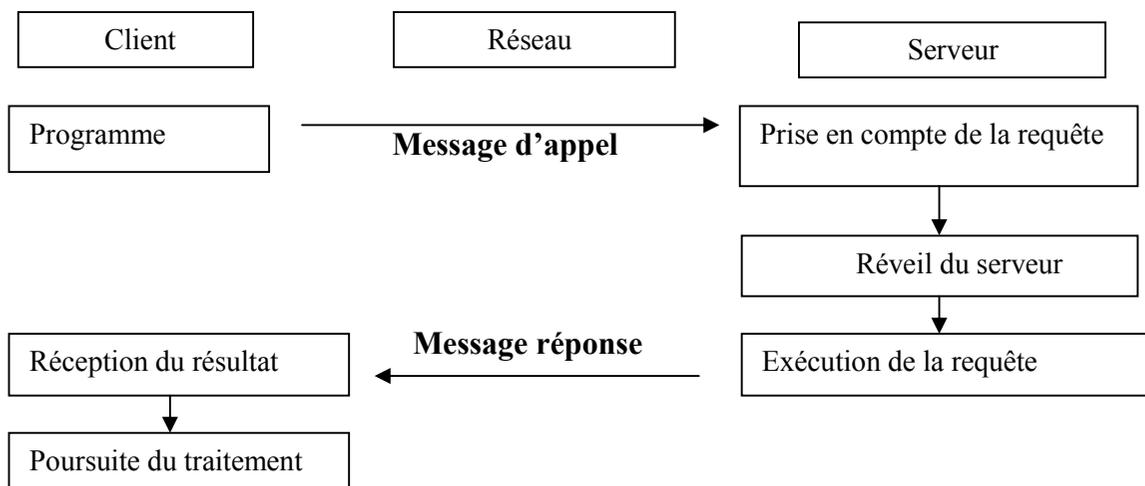


Figure 2.11 : Mode non connecté

### 3.3.12 Avantages et inconvénients de l'architecture client/serveur :

#### 3.3.12.1 Avantages de l'architecture client/serveur

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

- ✓ **des ressources centralisées** : étant donné que le serveur est au centre du réseau, il peut gérer des ressources communes à tous les utilisateurs, comme par exemple une base de données centralisée, afin d'éviter les problèmes de redondance et de contradiction.

- ✓ **une meilleure sécurité** : car le nombre de points d'entrée permettant l'accès aux données est moins important.
- ✓ **une administration au niveau serveur** : les clients ayant peu d'importance dans ce modèle, ils ont moins besoin d'être administrés.
- ✓ **un réseau évolutif** : grâce à cette architecture il est possible de supprimer ou rajouter des clients sans perturber le fonctionnement du réseau et sans modification majeure.

### 3.3.12.2 Inconvénients du modèle client/serveur :

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

- ✓ **un coût élevé** dû à la technicité du serveur
- ✓ **un maillon faible** : le serveur est le seul maillon faible du réseau client/serveur, étant donné que tout le réseau est architecturé autour de lui ! Heureusement, le serveur a une grande tolérance aux pannes.

**4. Conclusion :**

Dans ce chapitre nous a permis d'étudier les différents architecture client serveur, ce qui nous à permis de choisir le type d'architecture pour notre système qui est la trois tiers. Le chapitre qui suit sera donc consacré à l'analyse et conception de notre application.

# Chapitre 3 :

*Analyse*

*Et*

*Conception*

**1. Introduction :**

Le génie logiciel et la méthodologie s'efforcent de couvrir tous les aspects de la vie du logiciel. Issus de l'expérience des développeurs, concepteurs et chefs de projets, ils sont en constante évolution, parallèlement à l'évolution des techniques informatiques et du savoir-faire des équipes. Comme toutes les tentatives de mise à plat d'une expérience et d'un savoir-faire, les méthodologies ont parfois souffert d'une formalisation excessive, imposant aux développeurs des contraintes parfois contre-productives sur leur façon de travailler.

Avec la mise en commun de l'expérience et la maturation des savoir-faire, on voit se développer à présent des méthodes de travail à la fois plus proches de la pratique réelle des experts et moins contraignantes. UML, qui se veut un instrument de capitalisation des savoir-faire puisqu'il propose un langage qui soit commun à tous les experts du logiciel, va dans le sens de cet assombrissement des contraintes méthodologiques.

**2. Présentation UML :****2.1 Définition UML :**

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue.

UML unifie à la fois les notations et les concepts orientés objet. Il ne s'agit pas d'une simple notation graphique, car les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage.

UML unifie également les notations nécessaires aux différentes activités d'un processus de développement et offre, par ce biais, le moyen d'établir le suivi des décisions prises, depuis l'expression de besoin jusqu'au codage.

**2.2 Pourquoi modéliser ?**

- ✓ Un modèle est une simplification de la réalité qui permet de mieux comprendre le système à développer.
- ✓ Il permet :
  1. De visualiser le système comme il est ou comme il devrait l'être.
  2. De valider le modèle vis à vis des clients.
  3. De spécifier les structures de données et le comportement du système.
  4. De fournir un guide pour la construction du système.
  5. De documenter le système et les décisions prises.

2.3 Historique d'UML :

- ✓ Fin des années 80 : compétition des méthodes d'analyse et de conception OO
- ✓ Booch : particulièrement adaptée au design et à l'implémentation
- ✓ OOSE (Jacobson) : expression des besoins.
- ✓ OMT-2 (Rumbaugh) : analyse et applications orientées-données
- ✓ 1994 : Rumbaugh rejoint Booch chez Rational.
- ✓ 1995 : Jacobson rejoint Rational
- ✓ 14 novembre 1997 : UML adopté par l'OMG.
- ✓ Différentes améliorations continuent d'être apportées au standard UML, donnant naissance à 4 révisions : UML 1.1, 1.3, 1.4, 1.5. UML 1.5 est la dernière révision avant le passage à la version UML 2.0.

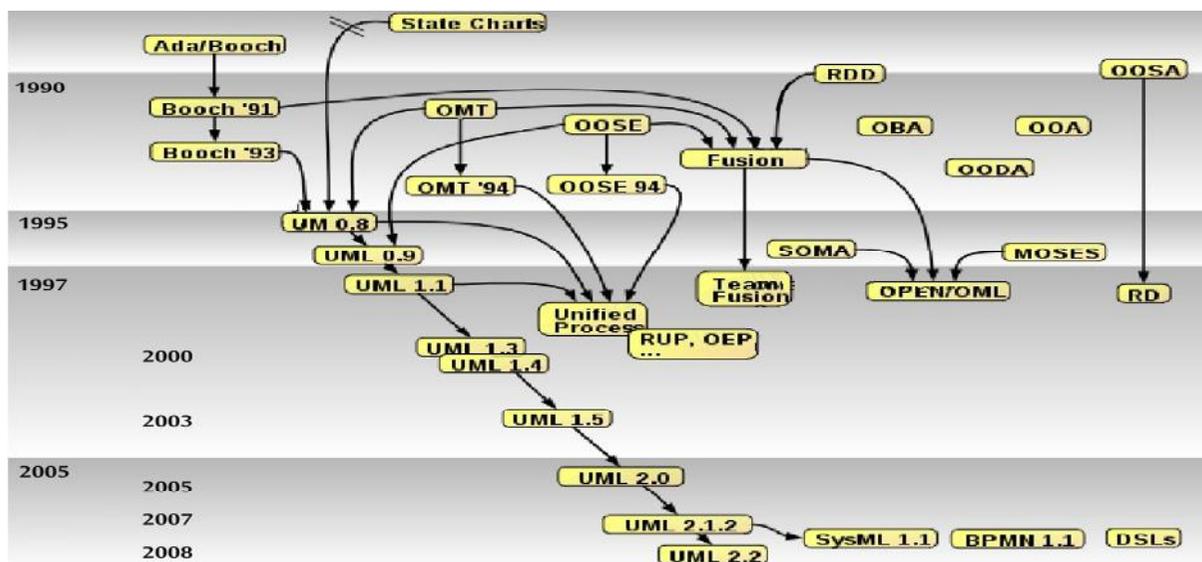


Figure historique UML

2.4 Les 9 Diagrammes :

Besoins des utilisateurs :

✓ Diagramme des cas d'utilisation :

Permettent De connaître le comportement du système sans spécifier comment ce Comportement sera réalisé et De définir les limites précises du système.

Au développeur ils lui permettent de bien comprendre l'attente des utilisateurs et les experts du domaine.

Structure statique :

✓ Diagramme de classes :

un des plus importants en conception, fournit la modélisation des entités du système. ils expriment structure statique du système en terme de classes et de relations entre ces classes. ce sont des extensions du modèle Entité-Association par l'introduction de : la généralisation

relation entre une classe plus générale et une classe plus spécifique, l'agrégation une forme spéciale d'association de la forme « partie de », et la spécification d'opérations et contraintes au niveau des entités, nommées ici Classes.

✓ **Diagramme objet :**

illustre le diagramme de classes dans des cas complexes.

**Dynamique des objets :**

✓ **Diagramme états-transition:**

il montre la manière dont l'état du système (ou de sous-parties) est modifié en fonction des événements du système.

✓ **Diagramme d'activités:**

règles d'enchaînement des activités du système ("organigramme" amélioré).

**Interactions entre objets:**

✓ **Diagramme de séquence:**

Le scénario d'un cas d'utilisation peut être représenté en utilisant un diagramme de séquences qui permet de visualiser l'aspect temporel des interactions et de connaître le sens des interactions (acteur vers système ou contraire)

✓ **Diagramme de collaboration :**

Le diagramme de collaborations sous une forme distincte du diagramme de Séquences représente les interactions entre classes en mettant moins en Évidence l'aspect temporel.

L'aspect temporel n'est pas complètement caché car chaque message est numéroté.

**Réalisation et déploiement :**

✓ **Diagramme de composants :**

Ce modèle définit l'architecture logicielle du système dans un environnement de développement donné. Il est issu de la conception et permet de représenter le système et les sous systèmes du modèle physique de l'architecture logicielle à réaliser.

✓ **Diagramme de déploiement :**

Ce modèle définit le diagramme de l'architecture matérielle du système ou de l'application. Il représente les différents processeurs, périphériques et la répartition du système sur ces différents éléments. Il montre les liens de communication entre ces différentes entités.

**3. Les diagrammes à utiliser pour notre application :**

Diagramme des cas d'utilisation pour décrire les besoins des utilisateurs.

Diagramme de classes pour décrire la structure statique.

Diagramme états-transition, Diagramme d'activités pour décrire la dynamique des objets.

Diagramme de séquence pour décrire les interactions entre objets.

Diagramme de déploiement, diagramme composants pour décrire réalisation et déploiement.

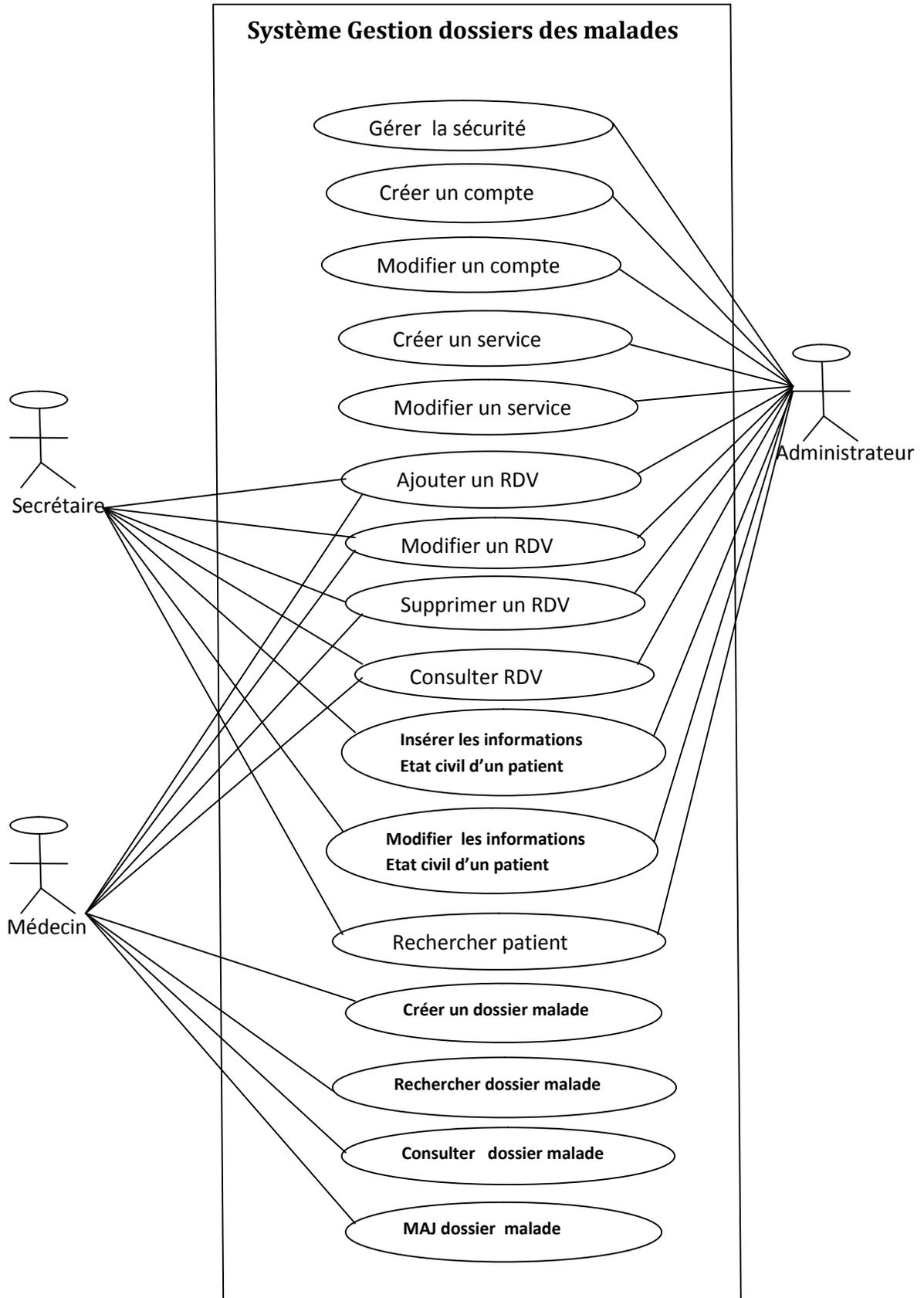
### 3.1 Liste des intervenants :

**Le médecin :** il soigne les maladies, pathologies et blessures. Il est très lié avec d'autres professionnels de la santé comme le psychologue, le pharmacien, l'infirmier ou le dentiste.

**L'administrateur :** répondre à l'urgence, d'assurer la permanence de l'administration et la continuité du fonctionnement des services.

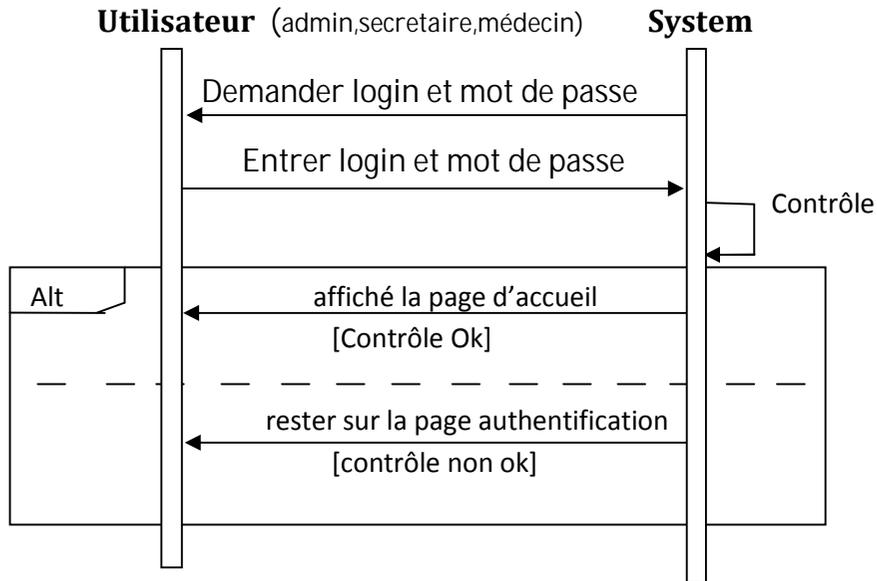
**Le secrétaire :** l'accueil physique et téléphonique des patients.

3.2 Diagramme de cas d'utilisation :

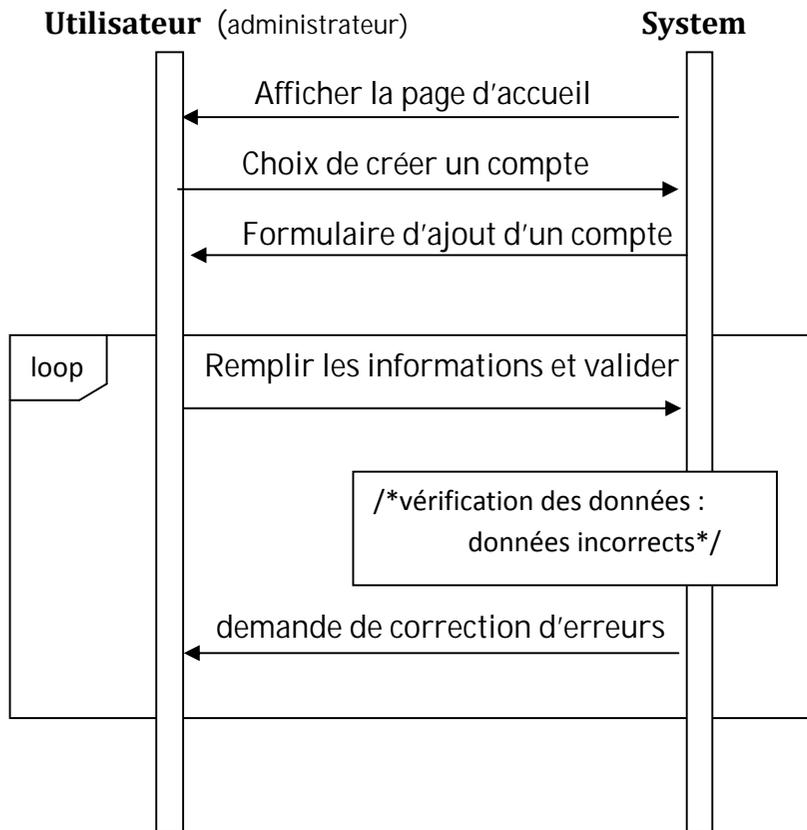


3.3 Diagramme de séquence :

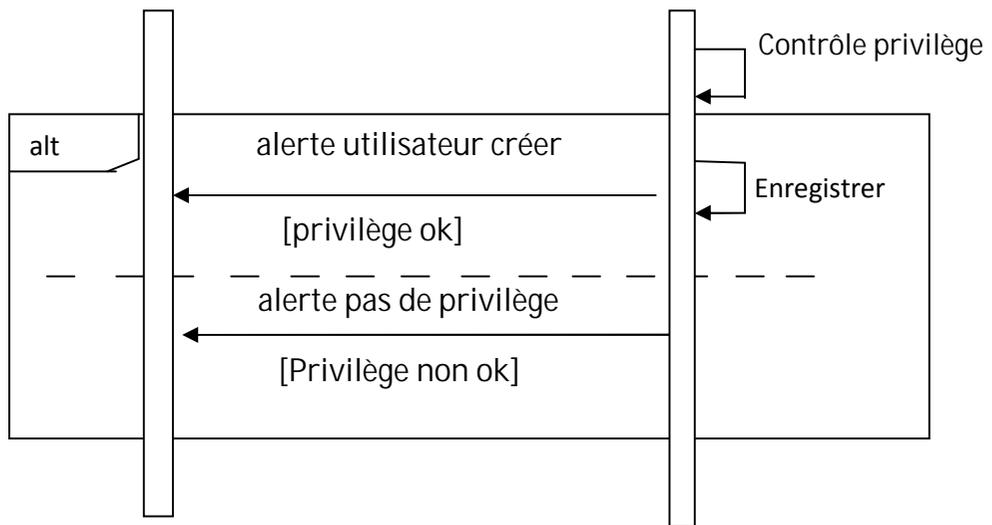
1. authentification d'utilisateur



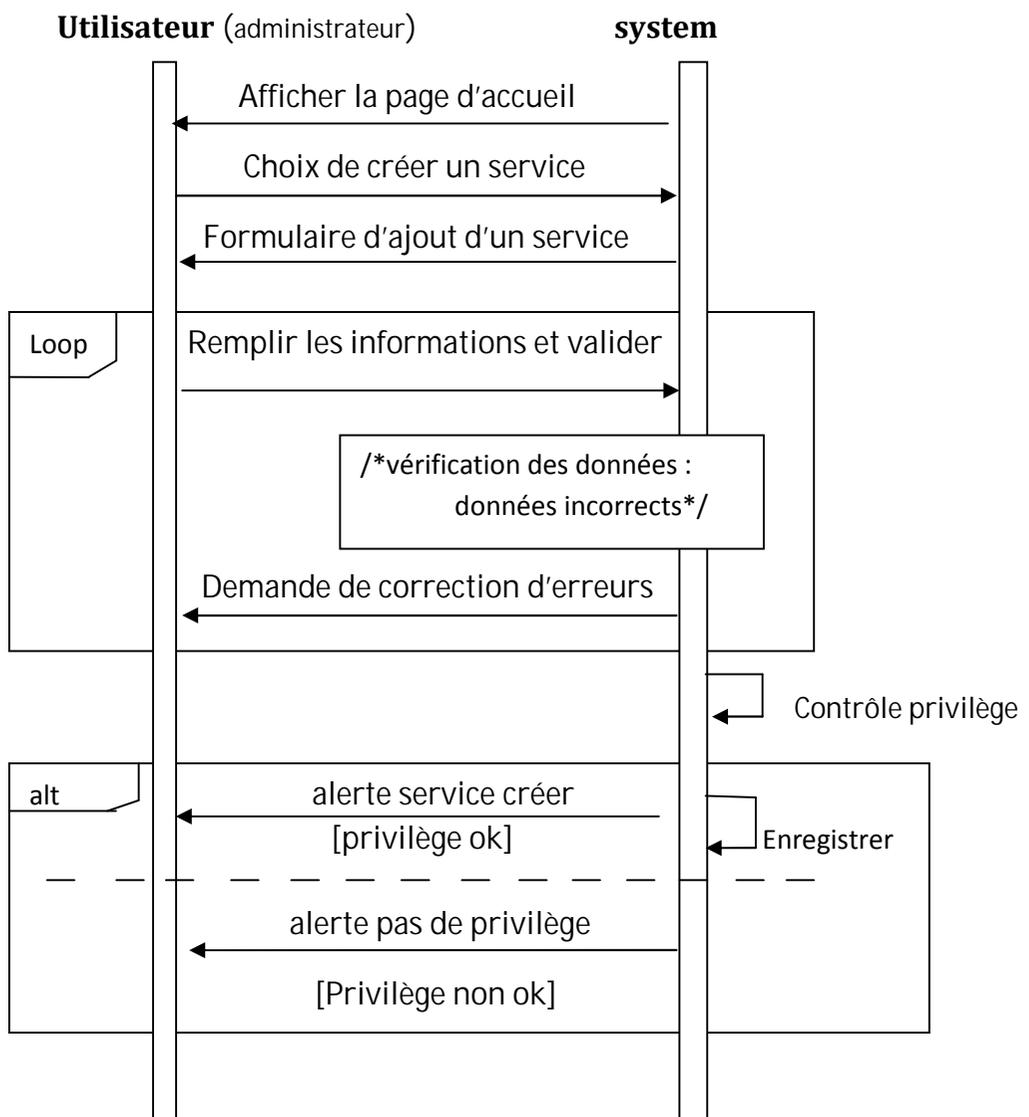
2. créer un compte :



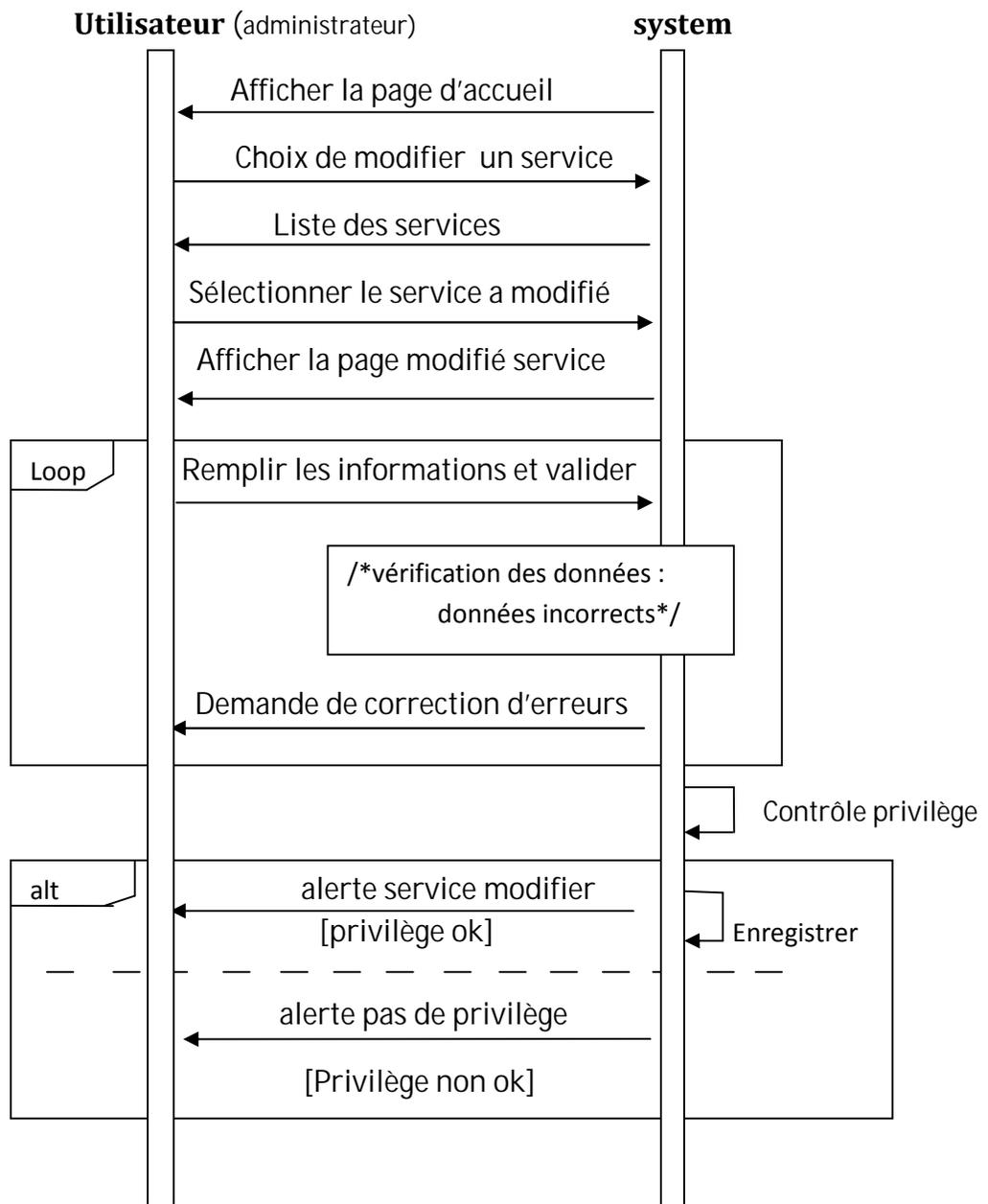
Suite du diagramme de séquence créer un compte :



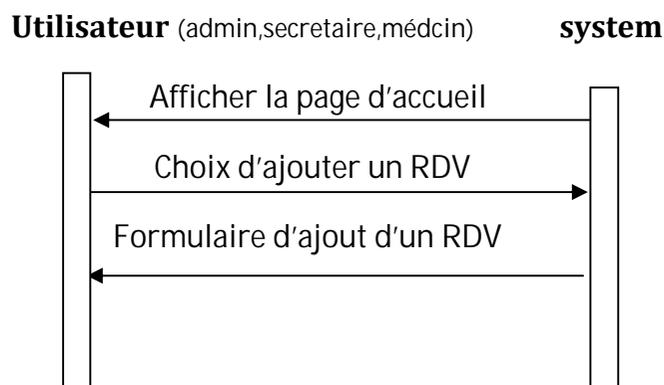
3. Diagramme de séquence créer un service :



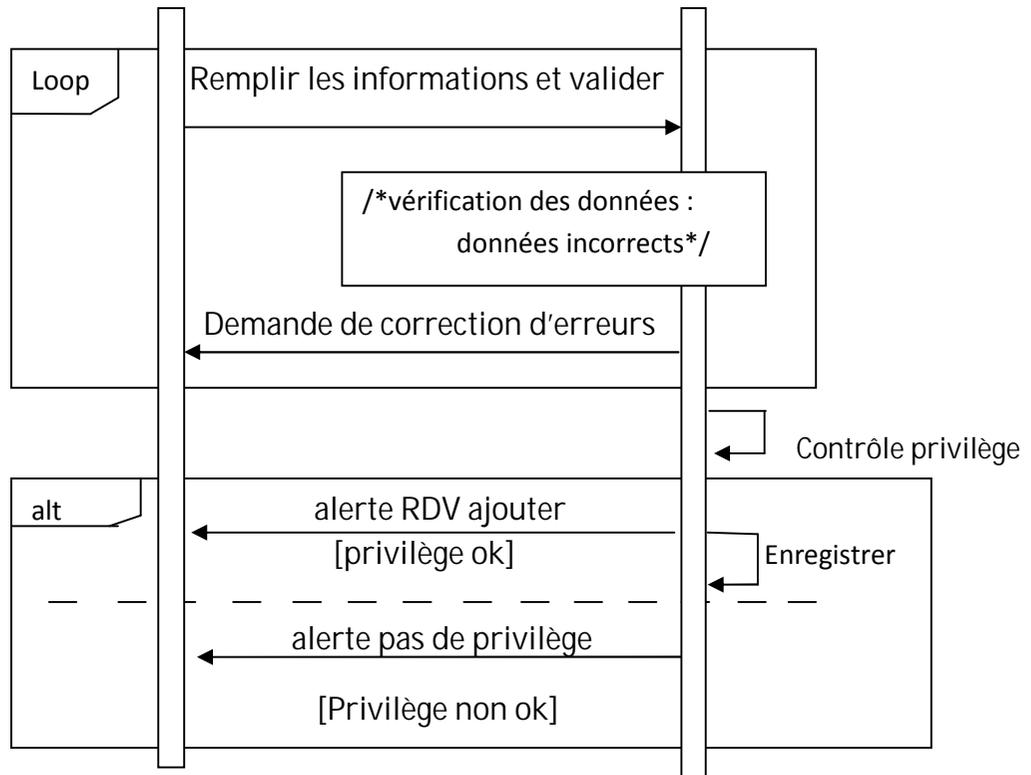
4. Diagramme de séquence modifier un service :



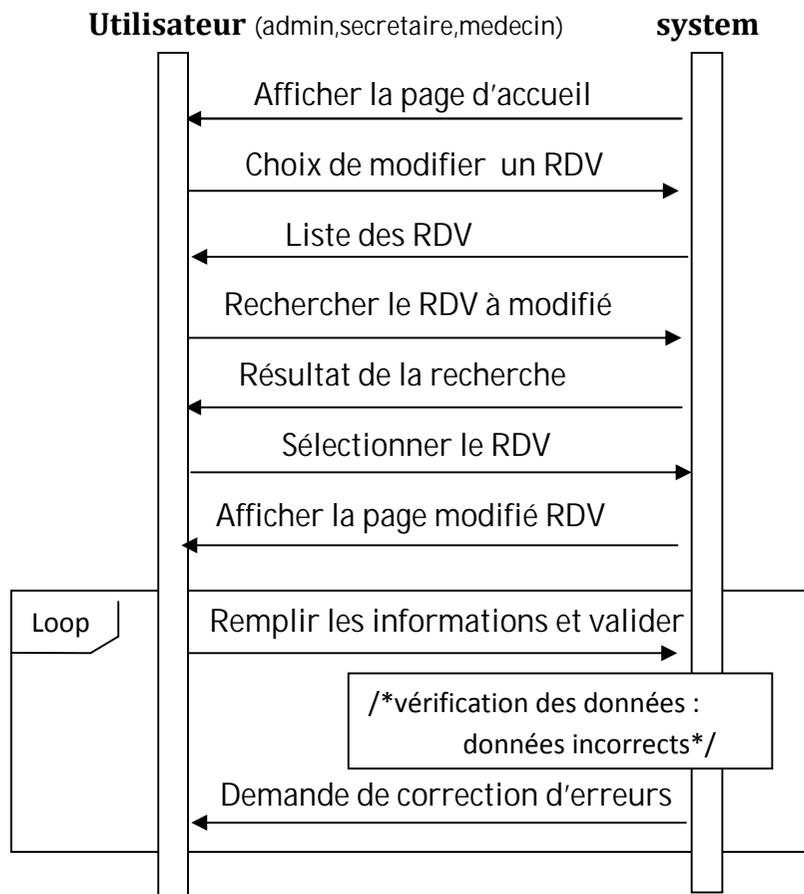
5. Diagramme de séquence ajouter un RDV :



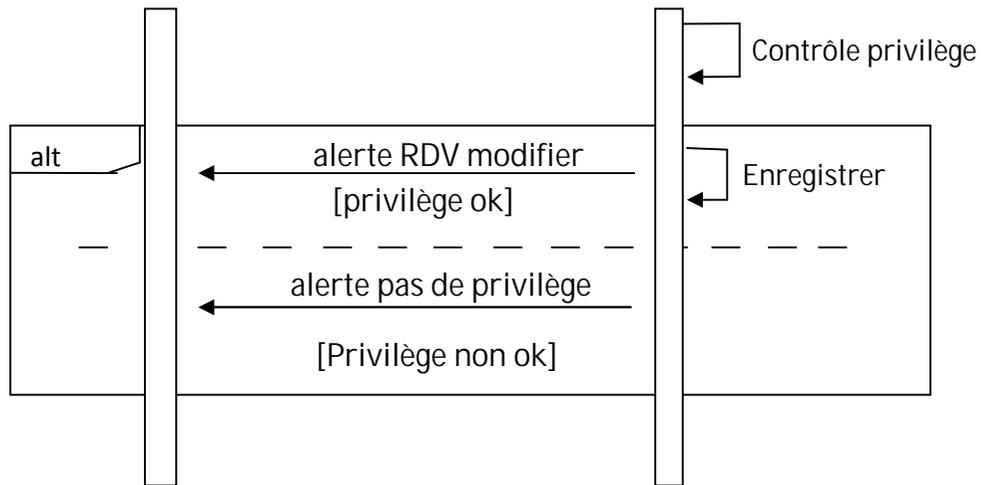
Suite Diagramme de séquence ajouter un RDV :



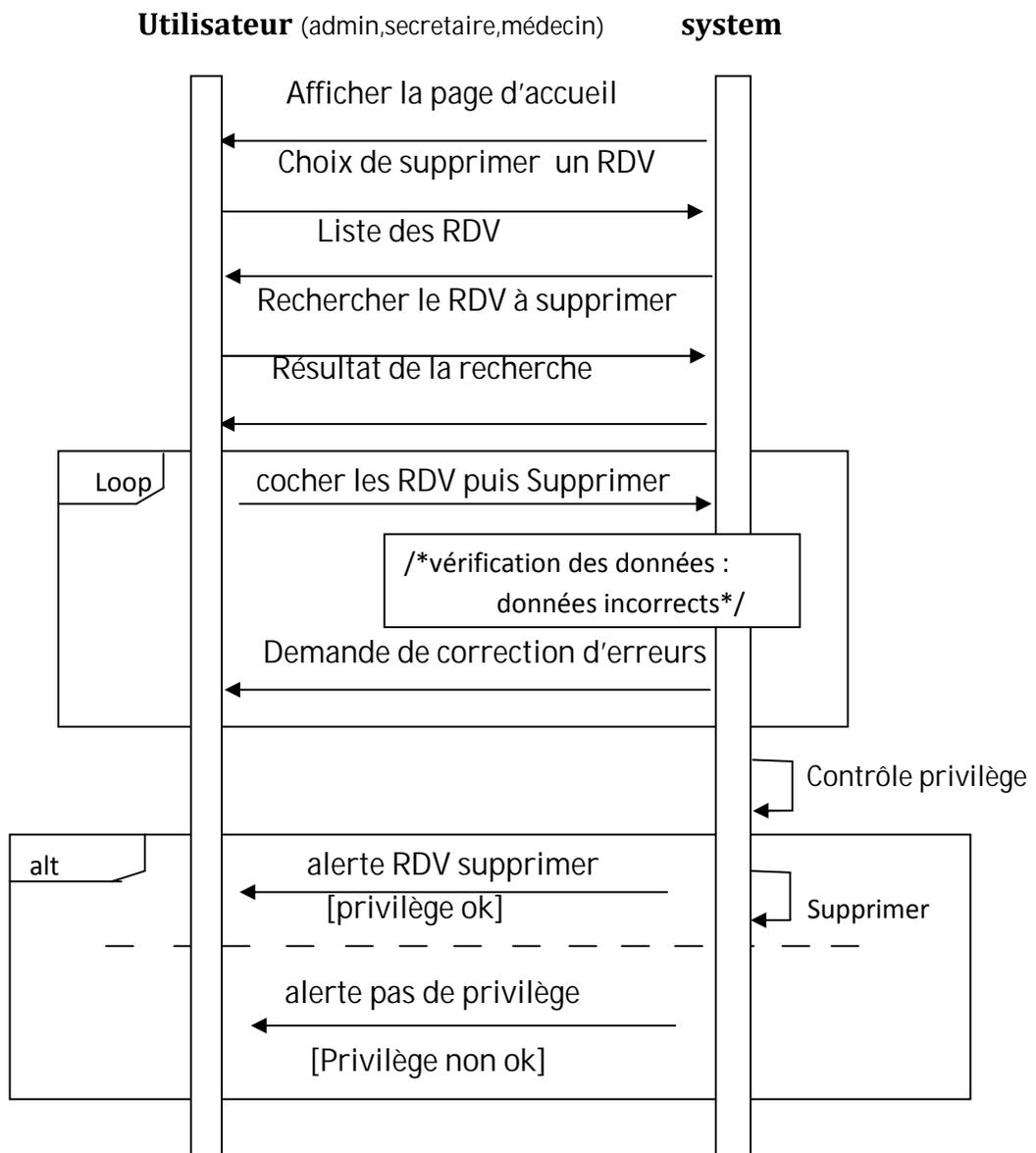
6. Diagramme de séquence modifier un RDV:



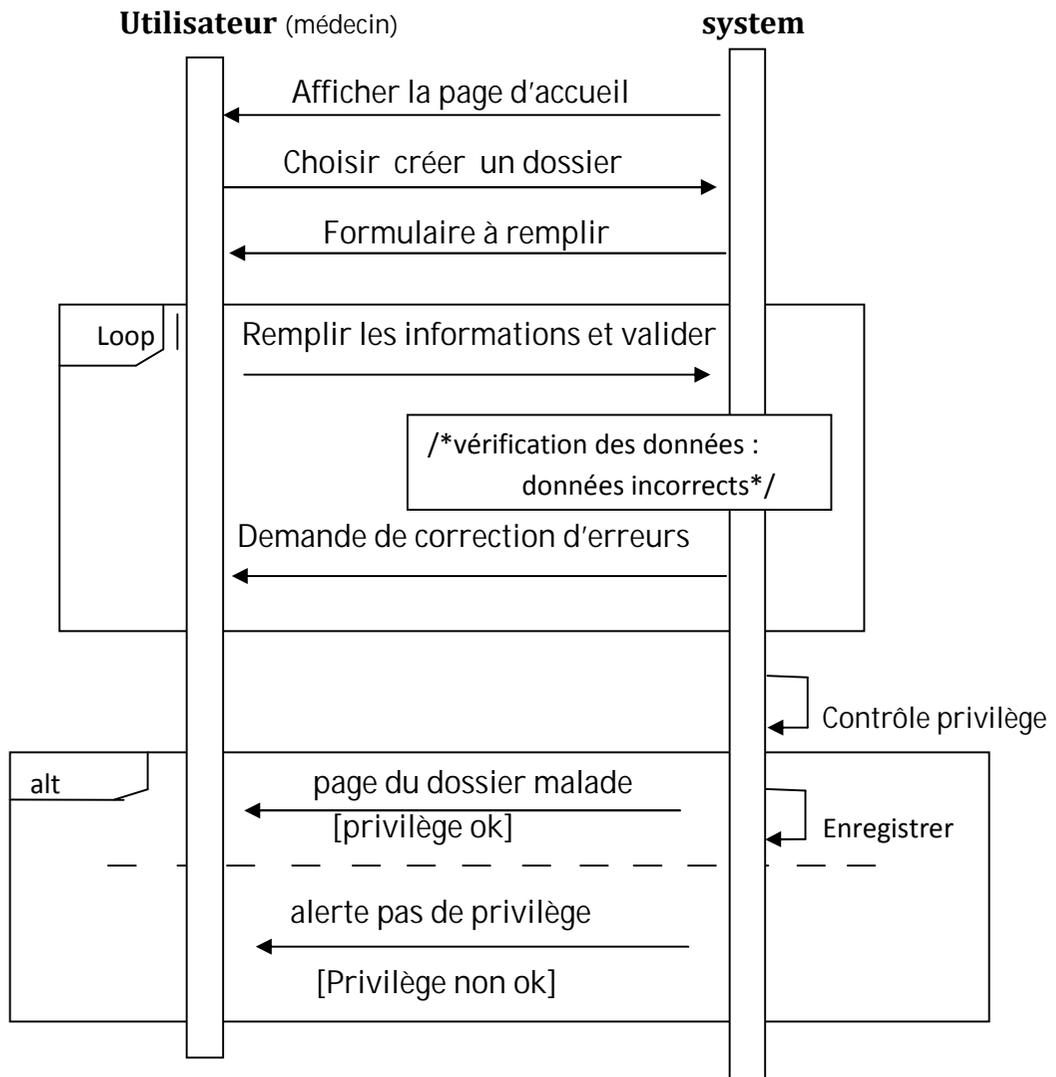
Suite Diagramme de séquence modifier un RDV:



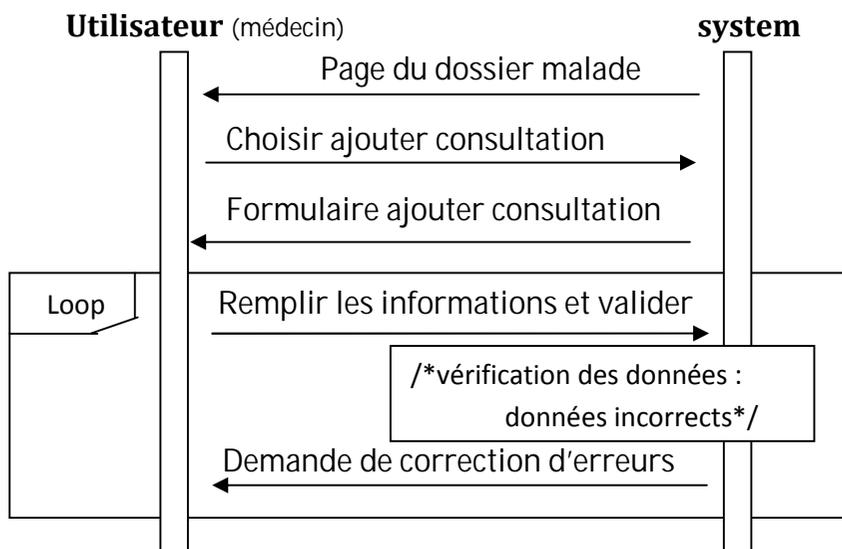
7. Diagramme de séquence supprimer un RDV



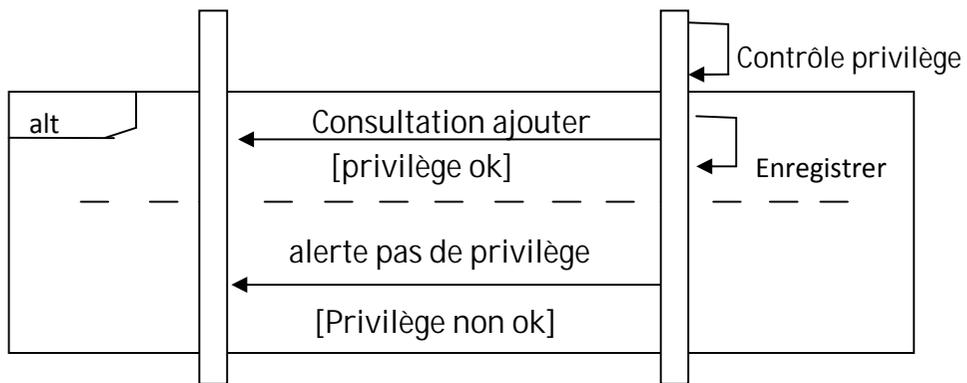
8. Diagramme de séquence créer un dossier :



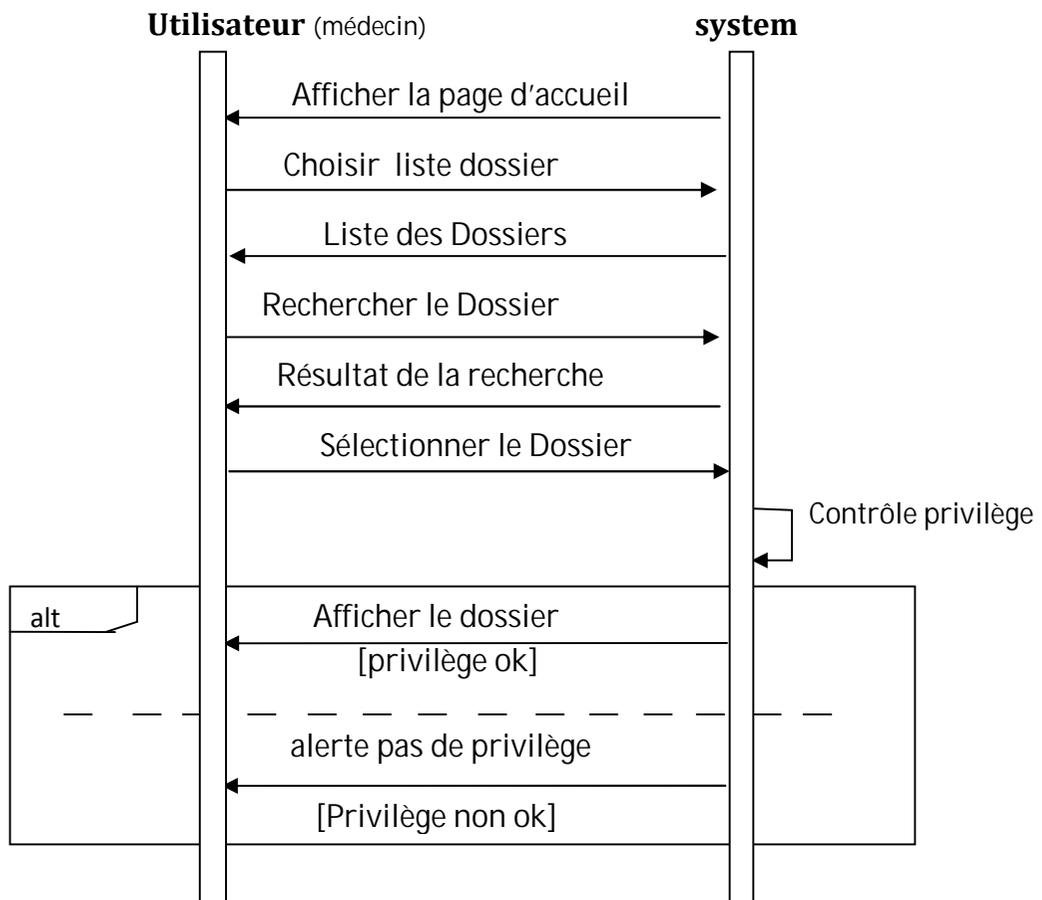
9. Diagramme de séquence ajouter une consultation :



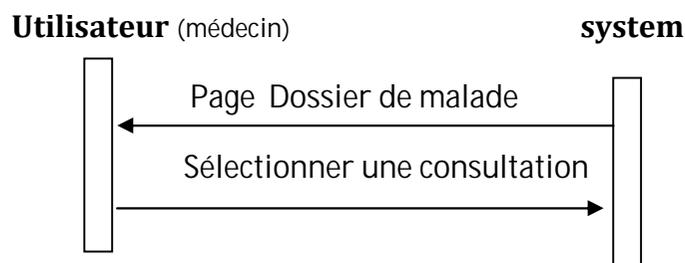
Suite Diagramme de séquence ajouter une consultation :



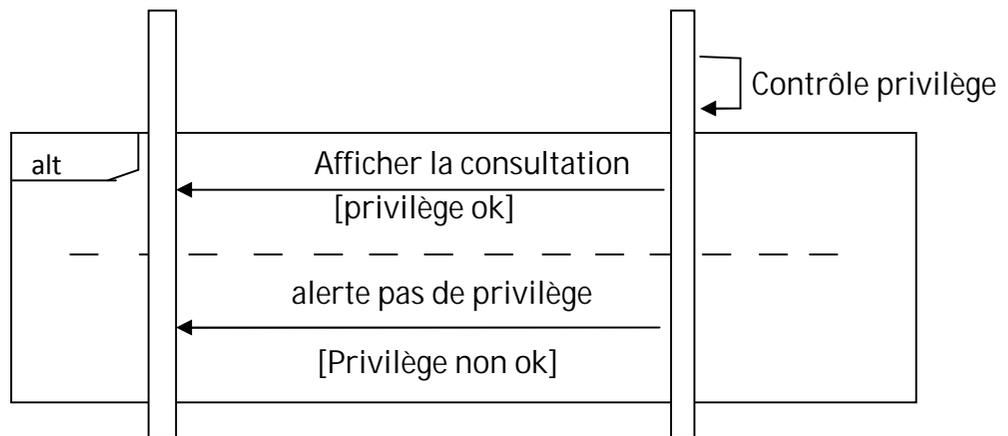
10. Diagramme de séquence consulter dossier malade :



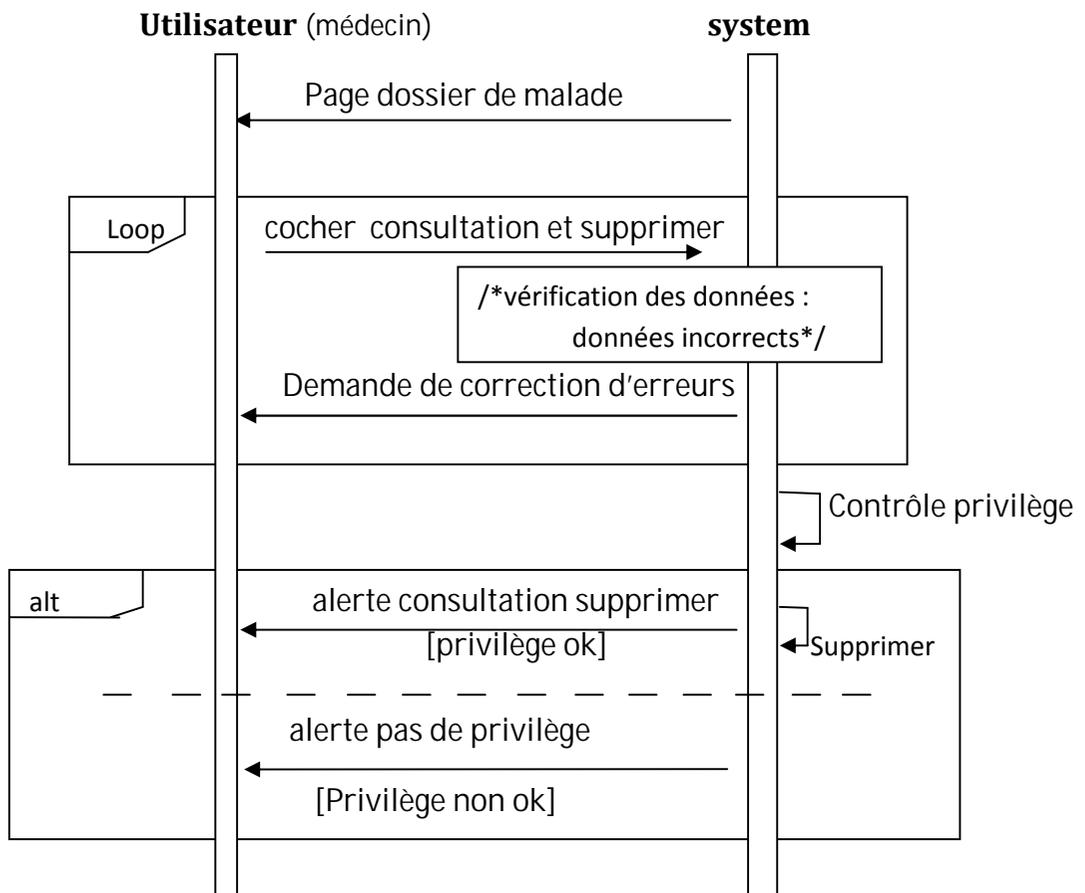
11. Diagramme de séquence voir une consultation :



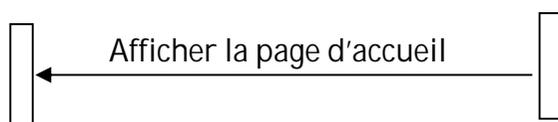
Suite diagramme de séquence voir consultation :

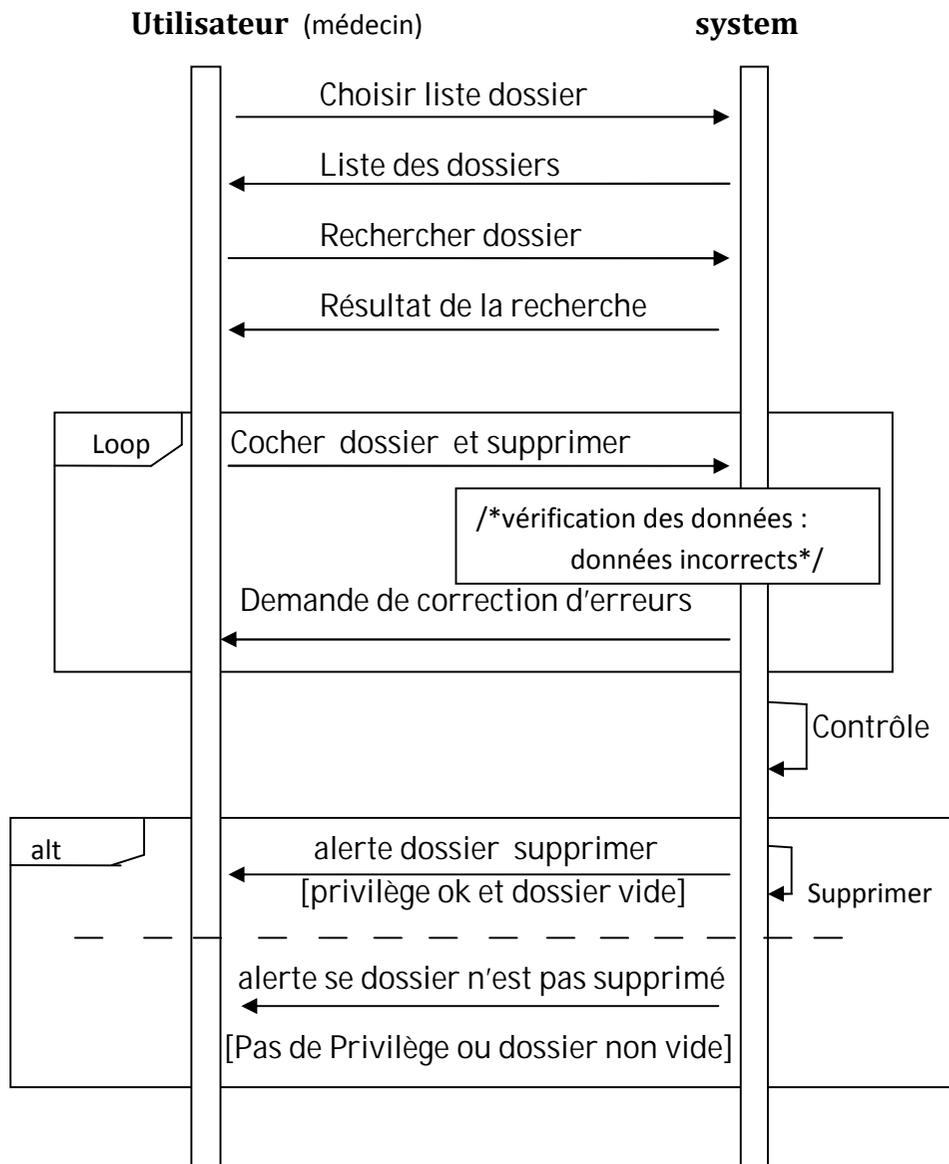


12. Diagramme de séquence supprimer une consultation d'un dossier:

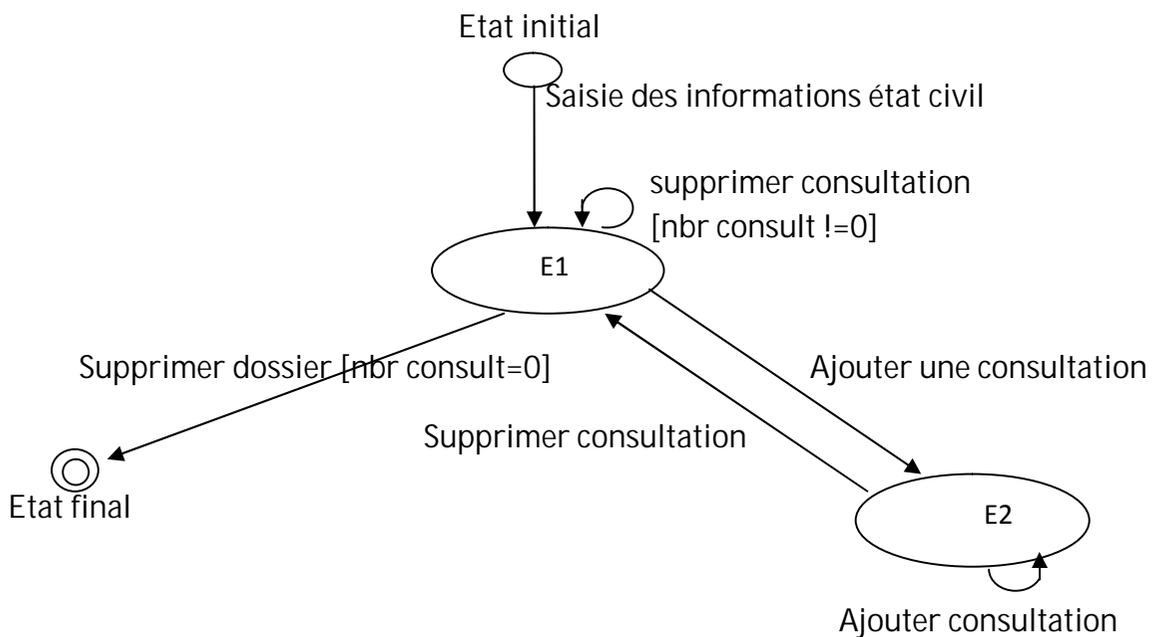


13. Diagramme de séquence supprimer dossier :



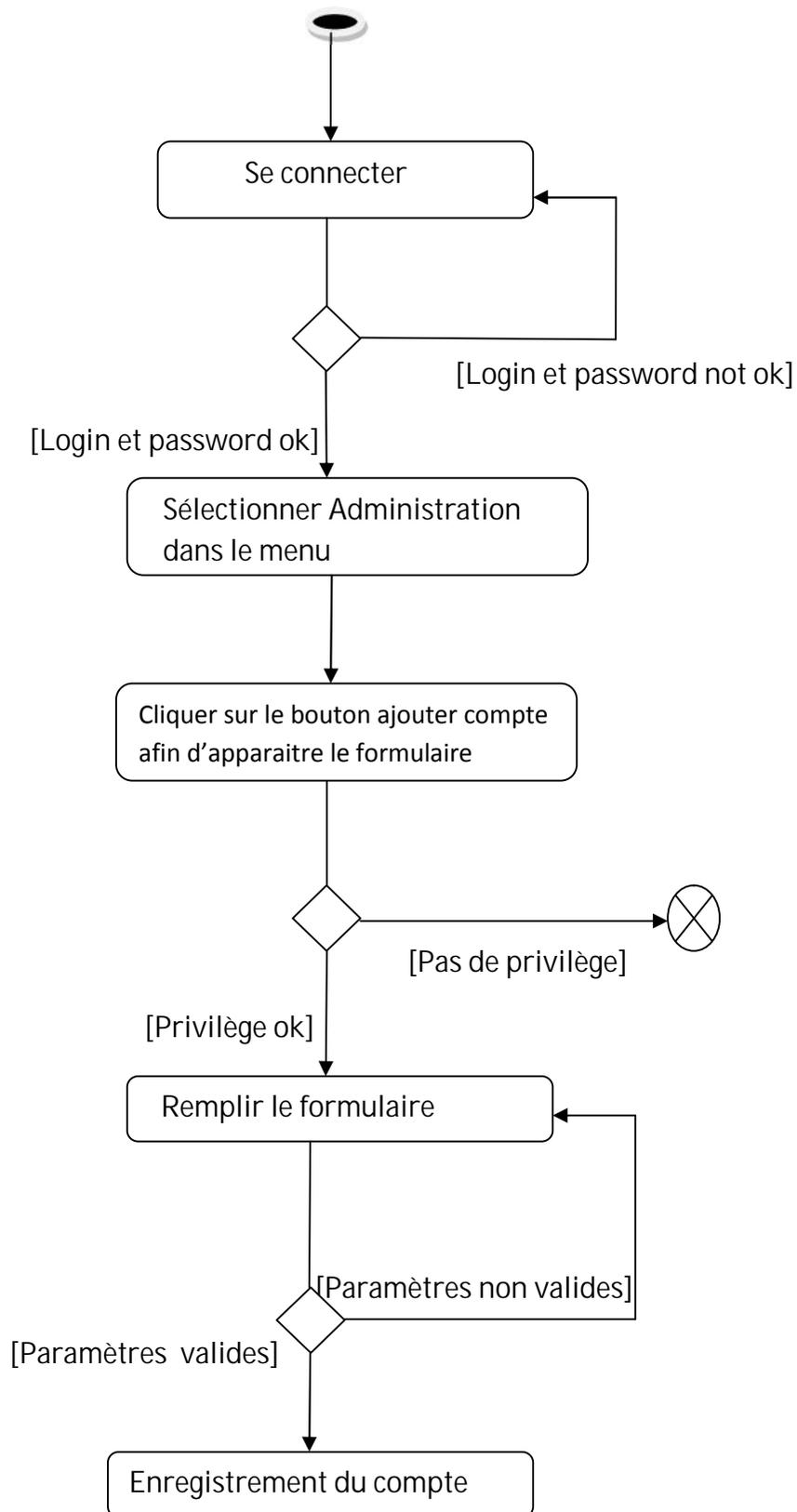


3.4 Diagramme d'état du dossier de malade :

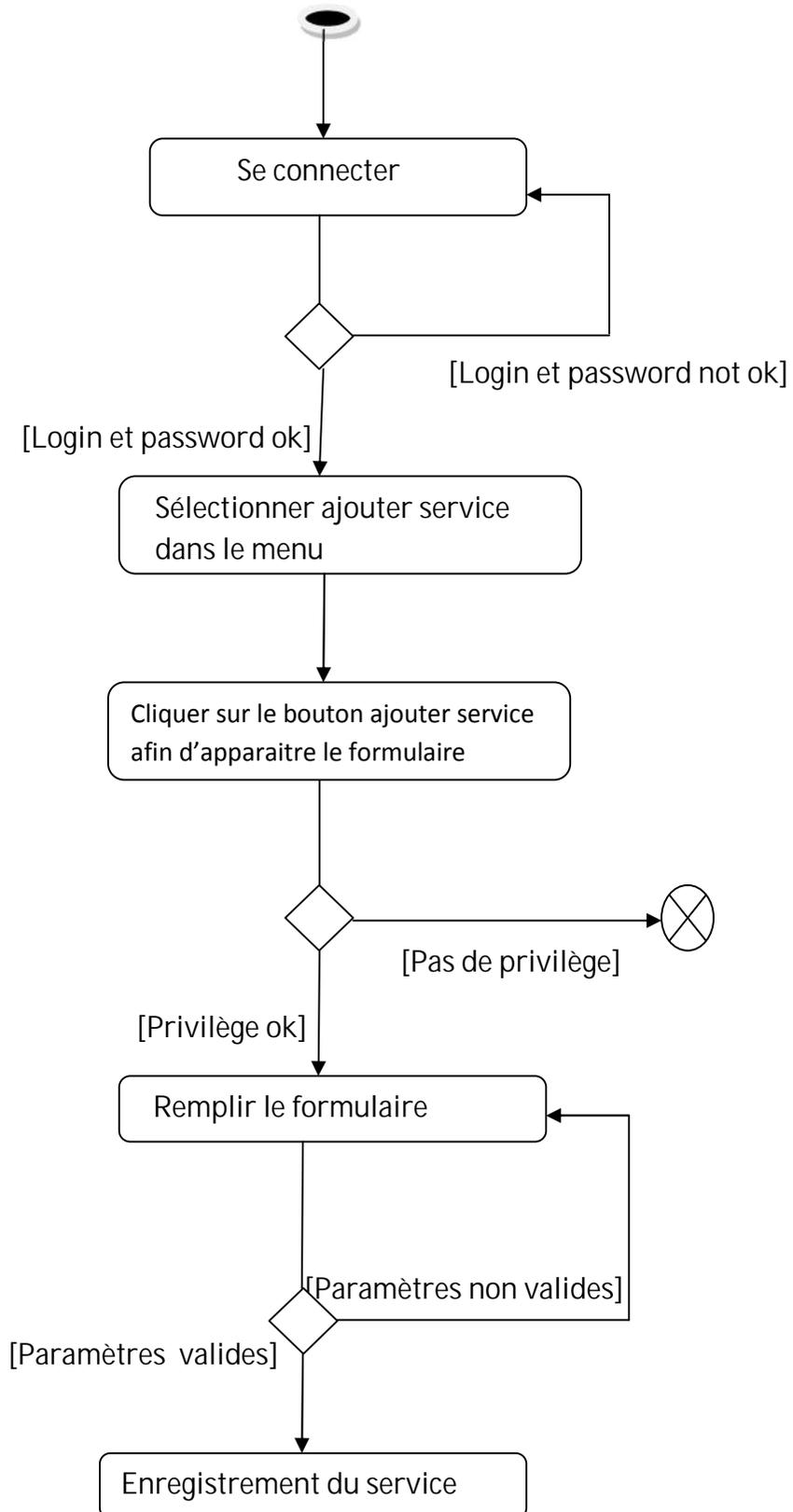


3.5 Diagramme d'activité :

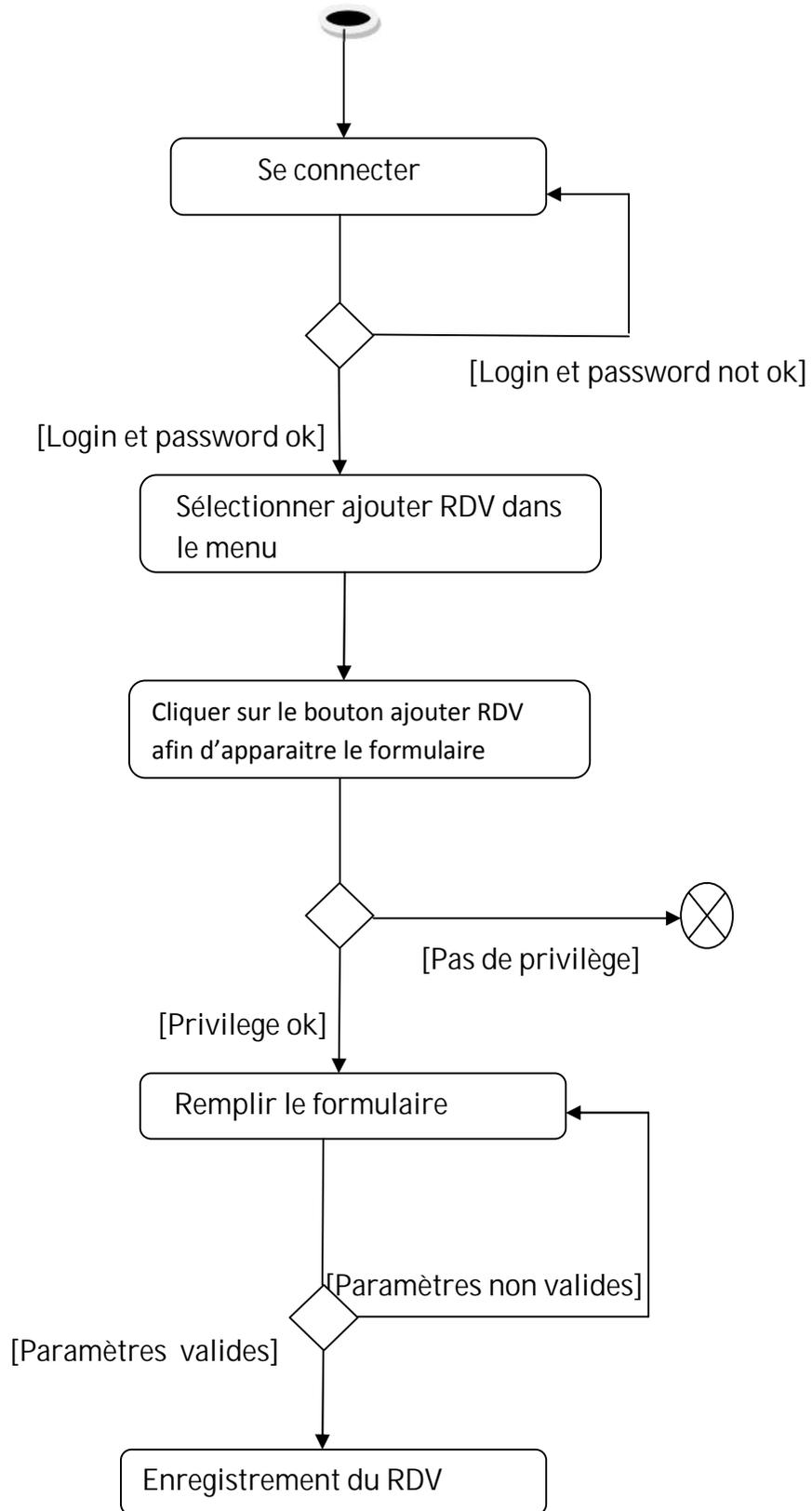
1. Diagramme d'activité ajouté un compte (administrateur):



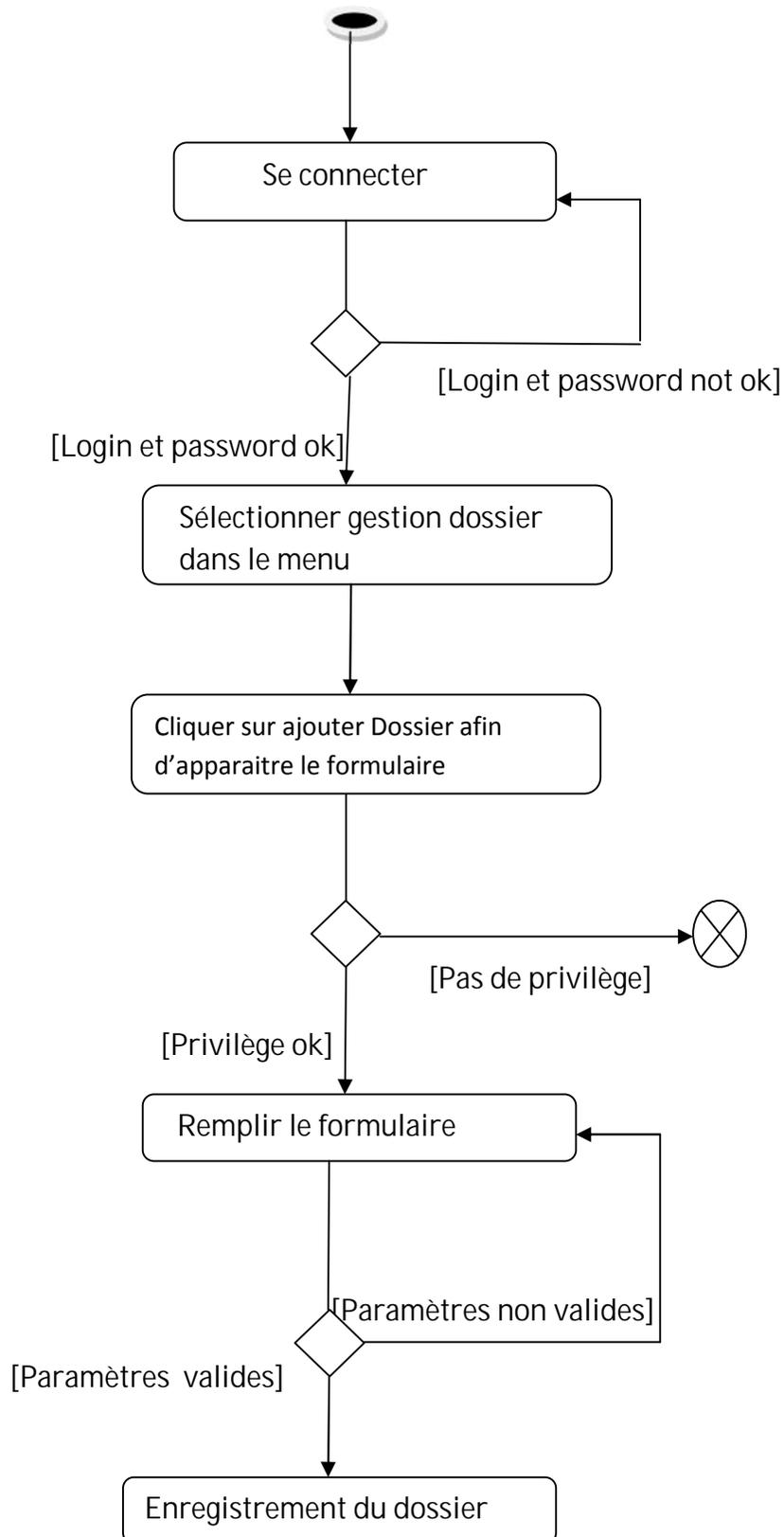
2. Diagramme d'activité ajouté un service (administrateur) :



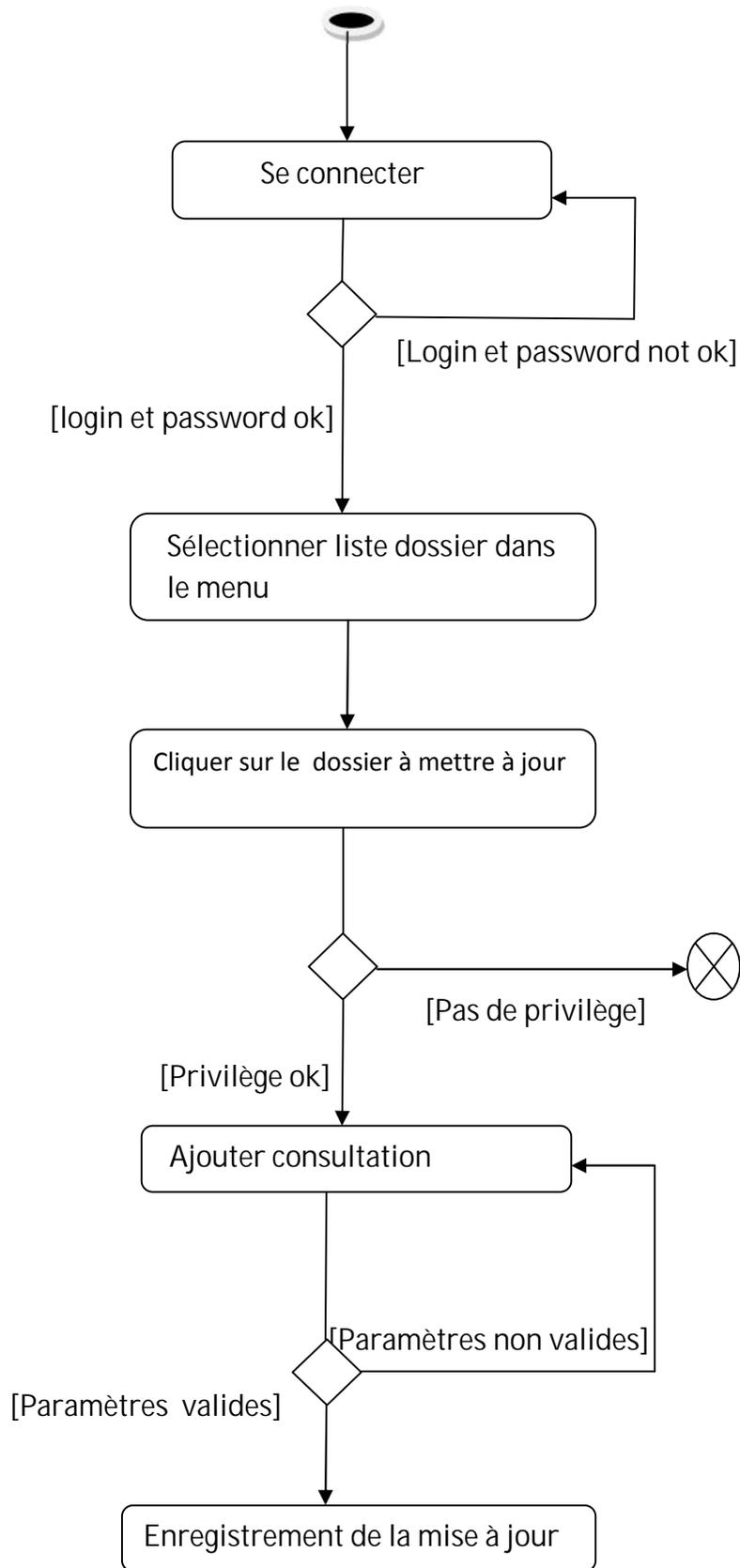
3. Diagramme d'activité ajouté un RDV (administrateur, secrétaire, médecin):



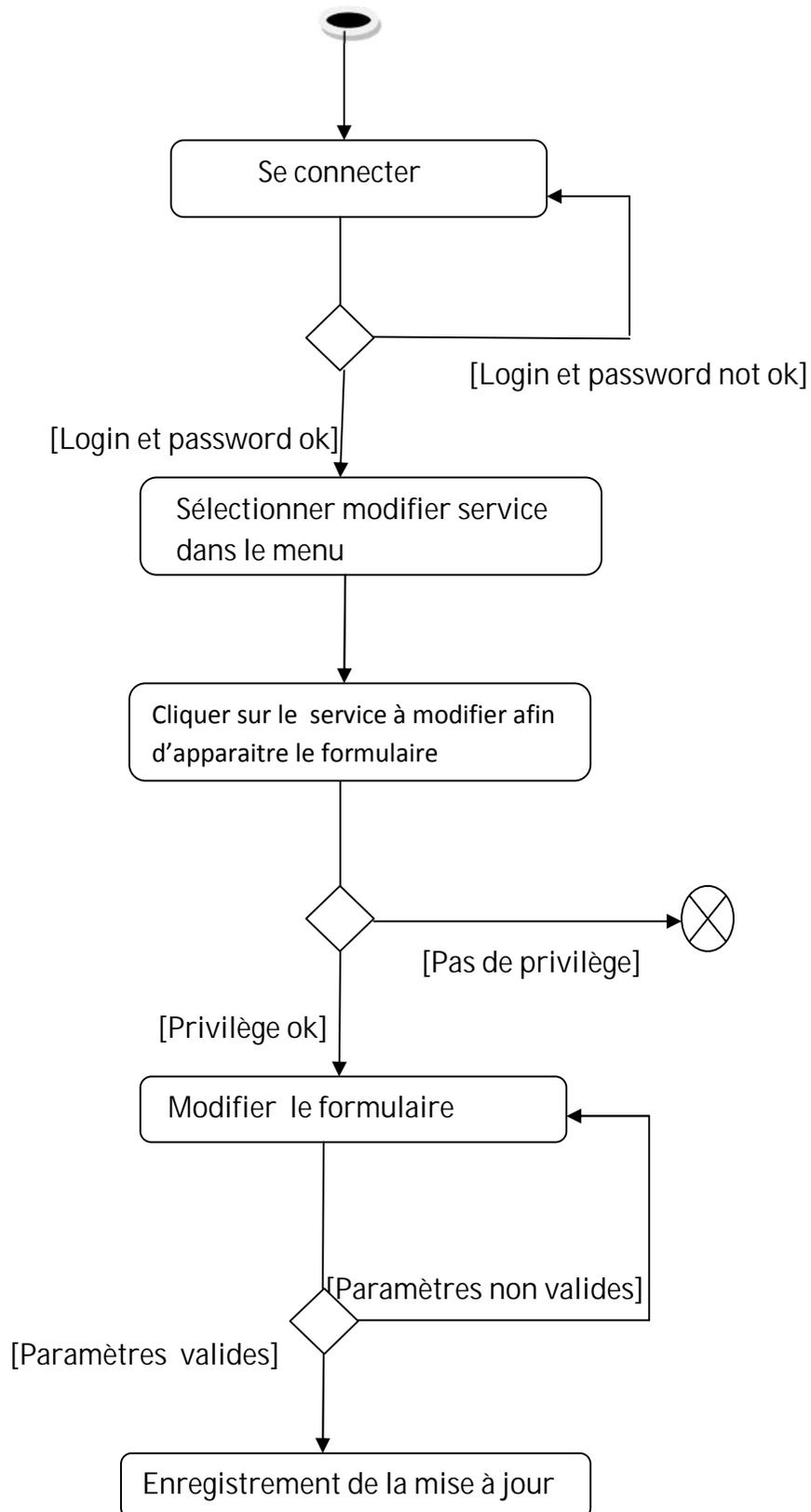
4. Diagramme d'activité ajouté un Dossier (médecin):



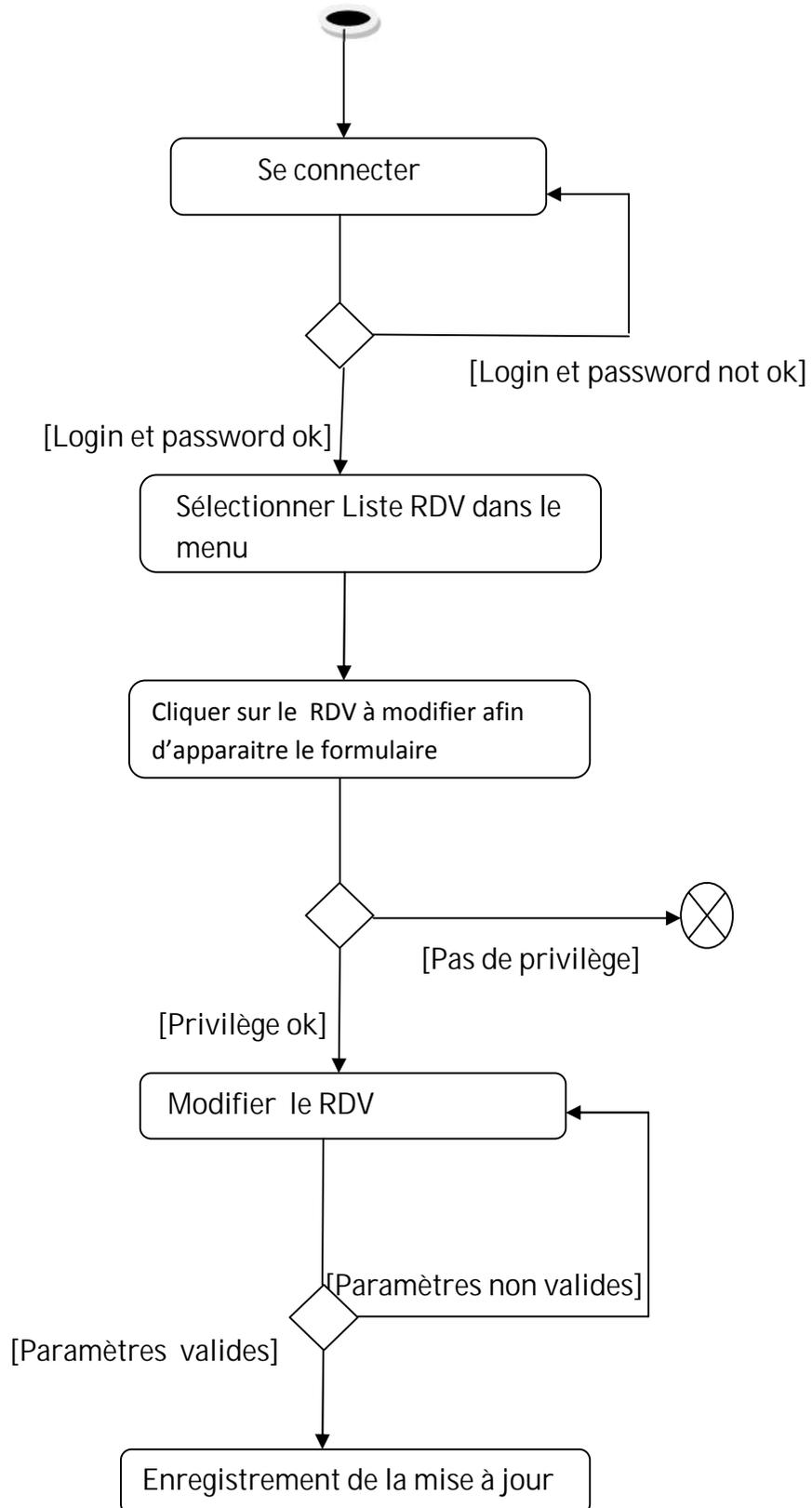
5. Diagramme d'activité Ajouter une consultation (médecin) :



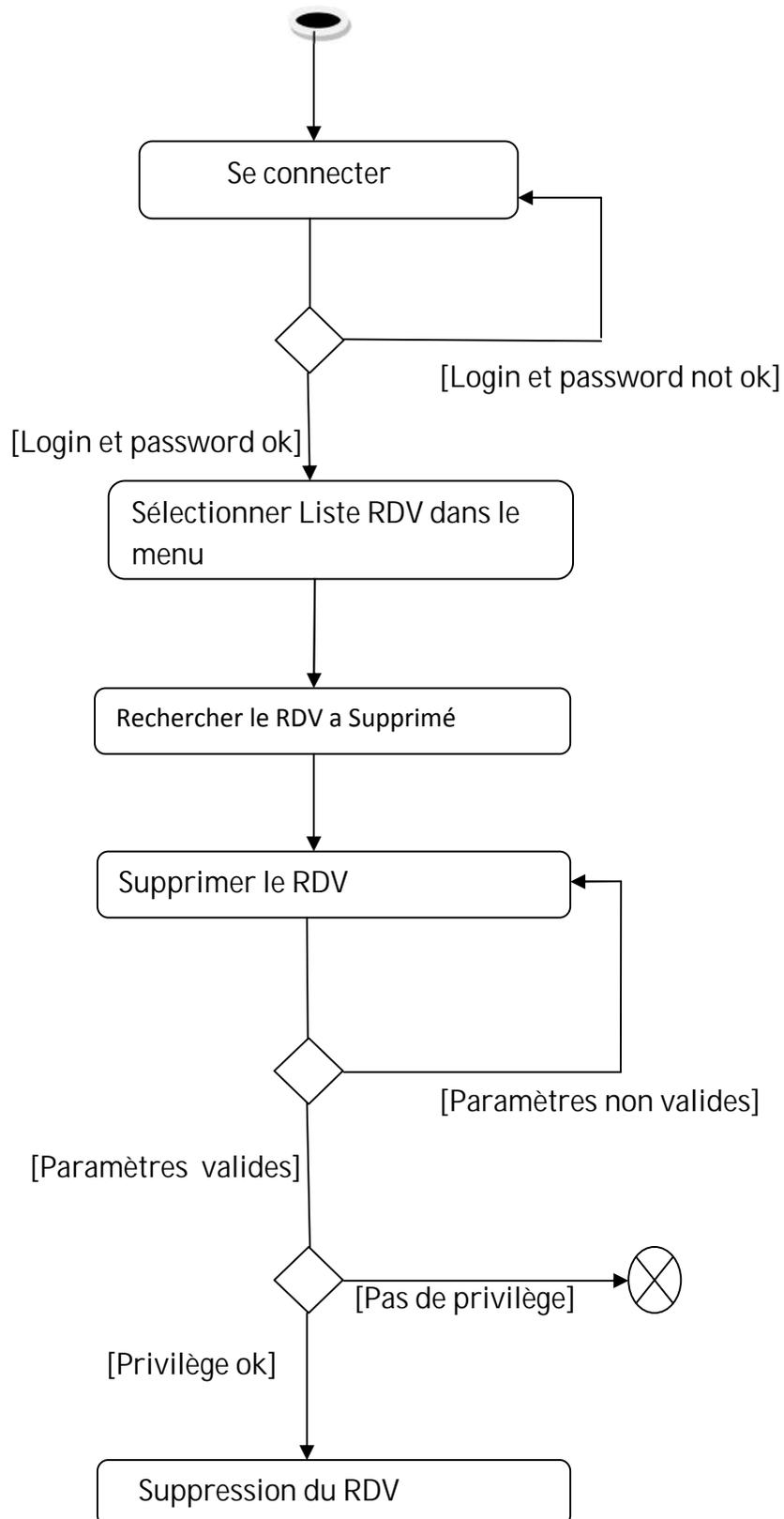
6. Diagramme d'activité modifier un service (administrateur) :



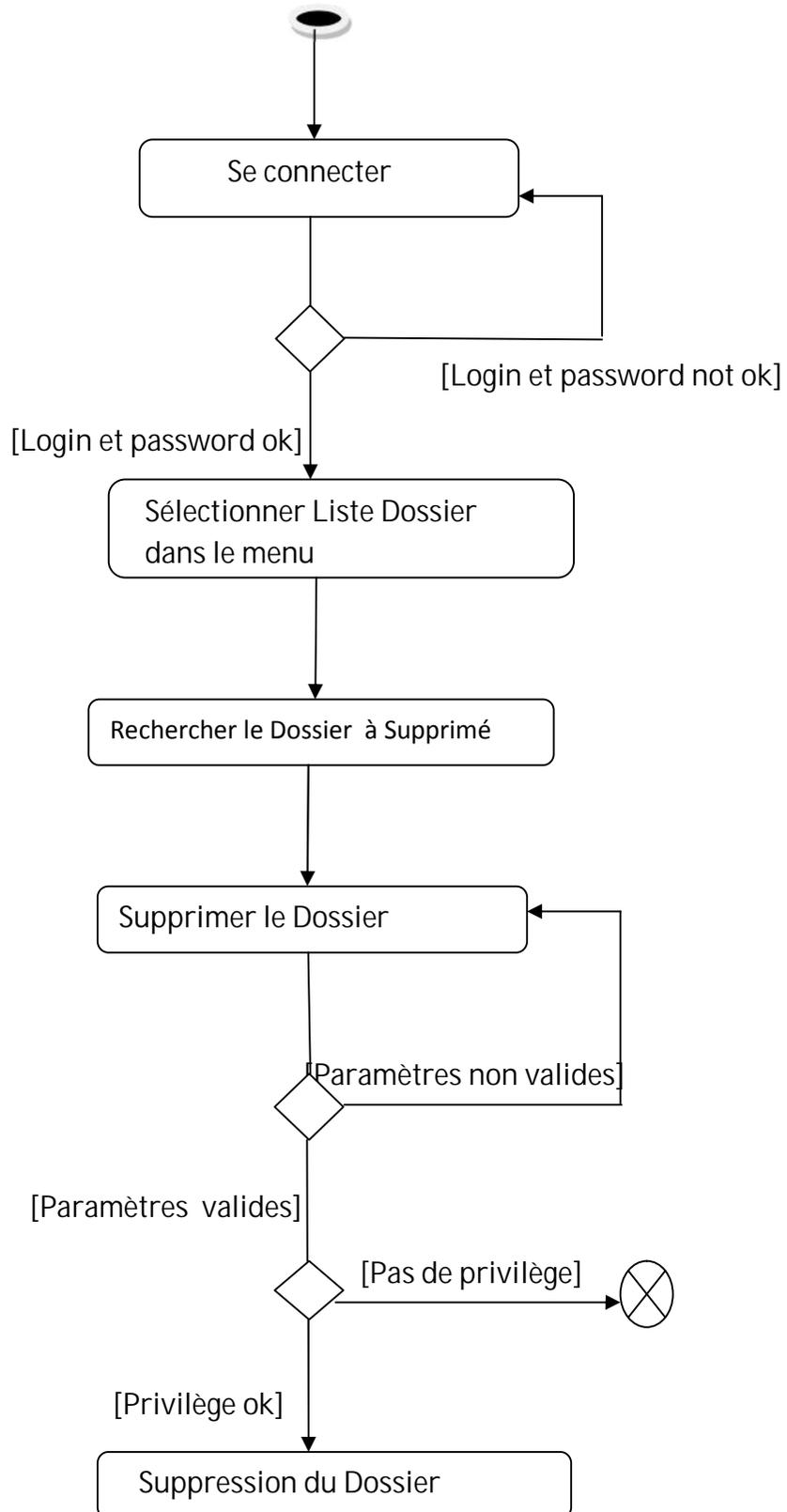
7. Diagramme d'activité modifier un RDV (administrateur, secrétaire, médecin):



8. Diagramme d'activité supprimé un RDV (administrateur, secrétaire, médecin):



9. Diagramme d'activité supprimé un Dossier (médecin) :



**3.6 Dictionnaire de données épuré :**

**Le DDD :** est un document qui fournit des détails sur toute entité enregistré par un système. Pour avoir un DDD épuré, il faut éliminer éventuelles incohérence qui sont de différents types :

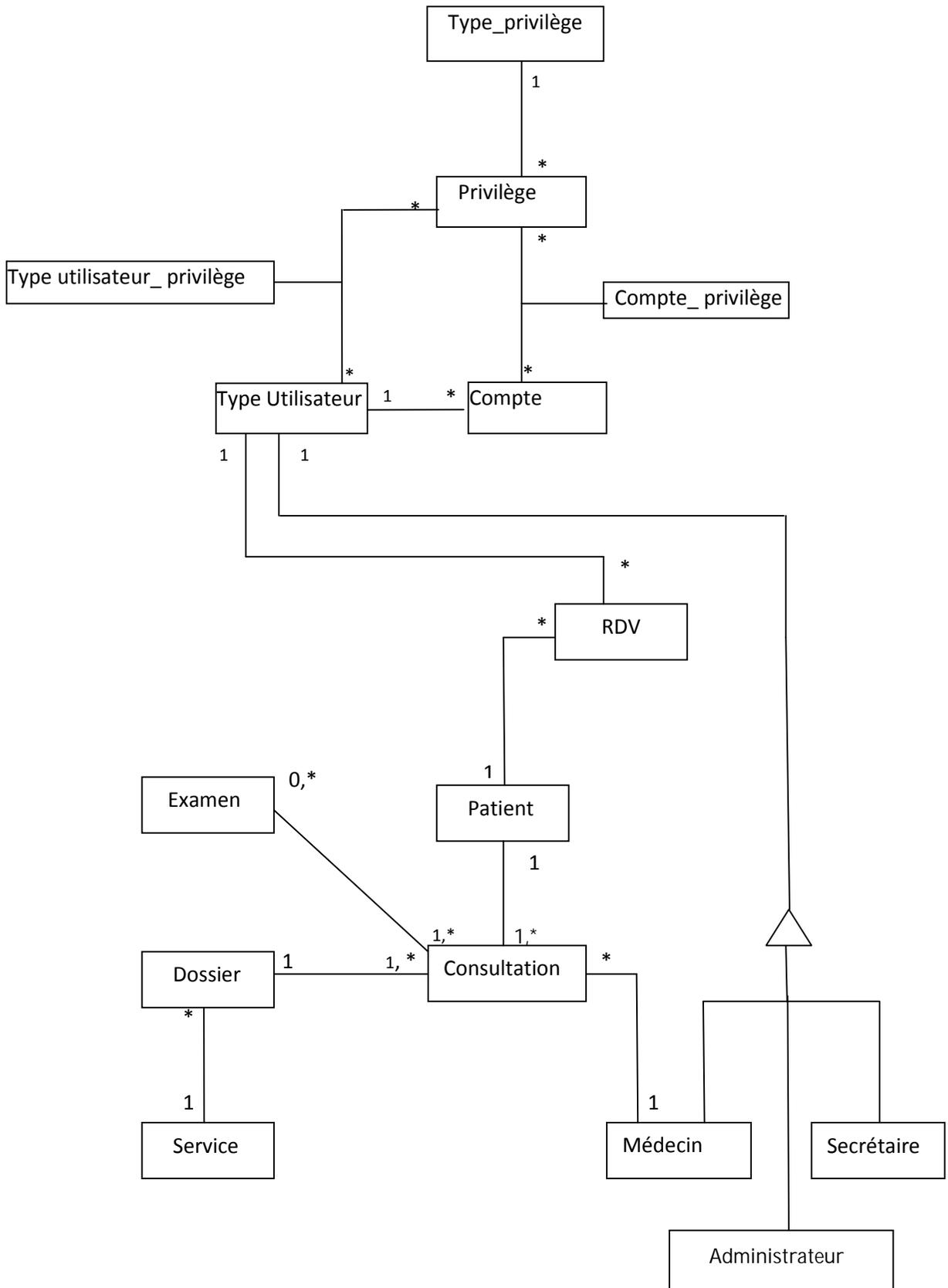
- **Les synonymes :** sont des noms différents qui désignent la même réalité.
- **Les polysémies :** sont des mêmes noms qui désignent des réalités différents.
- **Les redondances :** une même propriété utilisée plusieurs fois.

Le DDD épuré contient des données recensées des documents utilisés dans le champ d'étude.

## Dictionnaire des données épuré :

N°	code	Désignation rubrique	type	taille	observation
01	A -patient	Age patient			
02	Ad-par	Adressé par	A	20	
03	Ad-patient	Adresse patient	AN	20	
04	Dom-patient	Domicile patient	AN	25	
05	Doss N°pat	Dossier numéro patient	AN	07	
06	D-Naiss-pati	Date naissance patient	D	10	JJ/MM/AAAA
07	Dom-pati	Domicile patient	AN	25	
08	D-entr-l	Date entrée le	D	10	
09	D-sort-l	Date sorti le	D	10	
10	D-exam	Date examen	D	10	
11	E-entr	Etat a l'entrée	AN	50	
12	E-sort	Etat a la sortie	AN	50	
13	E-bio	Examens biologique	A	15	
14	E-imag-médi	Examens imagerie médical	A	15	
15	Hospita	Hospitalisation	AN	20	
16	Lieu-N-pat	Lieu naissance patient	AN	30	
17	Med-consu	Médecin consultant	A	20	
18	N-pat	Nom patient	A	15	
19	N-serv	Nom service	AN	25	
20	N-exam	Nature examens	AN	60	
21	N-doc	Nom document	AN	40	
22	N-privi	Nom privilège	A	10	
23	N-utilisa	Nom utilisateur	A	15	
24	N-typ-exam	Nom type examens	A	30	
25	N-exam	Nom examens		20	
26	Observ-trai	Observation et traitement	AN	60	
27	Pren-pat	Prénom patient	A	15	
28	Prof-pat	Profession patient	A	25	
29	Res-exam	Résultat examens	AN	60	
30	Sex-pat	Sexe patient	A	10	F O U M
31	Sall	Salle	AN	15	
32	Trait_sort	Traitement sortie	AN	80	

3.7 Diagramme de classe :



**3.8 Identification des classes du diagramme :**

Type privilège (id\_typ, libellée)

Privilège (id\_pri, libellé, n°compte\*, id type privilège)

Type utilisateur (id\_type\_utili, nom utilisateur)

Compte (n°compte, login, MDP, id\_type\_utili\*, id\_groupe\*)

RDV (n°RDV, date RDV, n°patient\*, id typ\_utili\*).

Examens (id\_examen, libellée examens).

Patient (id\_patient, Nom\_pati, prénom, date\_naissance, adresse, fonction, groupe S)

Dossier (N°dossier, id\_patient\*, id service\*, date\_consultation, id\_utilisateur\*)

Consultation (n°consultation, id\_medecin\*, n°type\_utili\*, n°patient\*, n°dossier\*, id\_dossier\*)

Service (id\_service, nom\_serv)

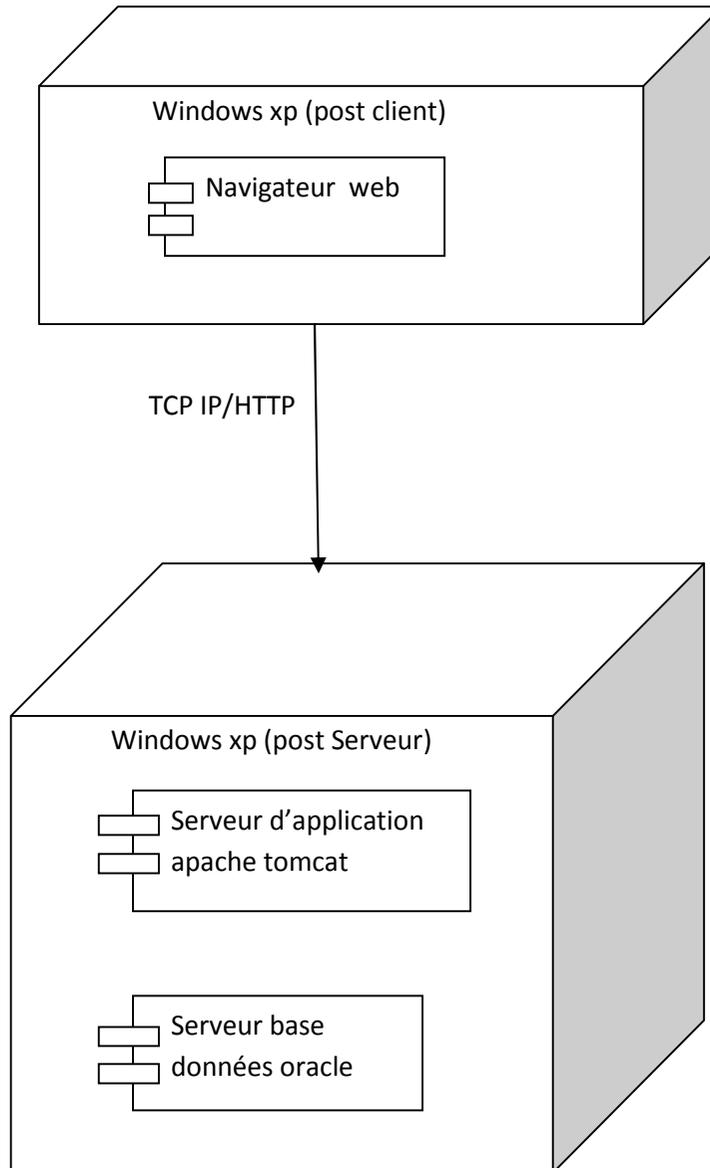
Compte\_privilège (id\_compte, id\_privilege).

Type\_utilisateur\_privilege (id typ utili, id\_privilege).

Consultation\_examens (n°consultation, n°examen).

3.9 Diagramme de déploiement :

Dans ce diagramme nous parlons du déploiement du logiciel. Le navigateur web va se trouver au niveau des clients à savoir le médecin et le secrétaire, et le serveur va se situer au niveau de l'administrateur comme l'illustre le schéma ci-dessous :



**4. Conclusion :**

Ce chapitre nous a permis de présenter les différents diagrammes d'UML. Pour notre cas, en premier nous avons utilisé le diagramme des cas d'utilisation afin de spécifier les besoins des utilisateur, ensuite les diagrammes de séquence qui consistent à décrire les interactions entre les objet, le diagramme d'état pour les objets plus complexe tel que (dossier malade) et nous avons enchaîner avec le diagramme d'activité qui nous permet de décrire la dynamique des objet, enfin le diagramme de classe qui va nous permettre de concevoir notre base de donnée. Le chapitre suivant se portera sur la réalisation.

# Chapitre 4 :

## Réalisation

**1. Introduction :**

Dans ce présent chapitre nous allons entamer la quatrième étape de notre travail, qui consiste en « réalisation d'une application client-serveur 3 tiers pour la gestion du dossier du malade ».

Pour cela nous avons utilisés des outils, en l'occurrence Oracle Database 11g comme système de gestion de base de données, Oracle SQL Developer, Java, eclipse, serveur d'application apache tomcat.

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

**2. Présentation des outils de développement :****2.1Présentation du langage JAVA :**

Java est un langage typé et orienté objet. Il est compilé et basé sur une architecture logicielle très particulière nécessitant une machine virtuelle Java. Il utilise les notions usuelles de la programmation orientée objet : la notion de classe, d'encapsulation, d'héritage, d'interface, de virtualité, de généricité,... Il est accompagné d'un ensemble énorme de bibliothèques standard couvrant de très nombreux domaines, notamment des bibliothèques graphiques. C'est un langage qui présente d'excellentes propriétés de portabilité du code. Son gros point faible est une relative lenteur, surtout si on le compare à des langages comme le C++. Cependant, ce défaut a été résolu en grande partie par l'introduction de la technologie JIT (compilateur *Just-In-Time*, en anglais « juste à temps »), qui compile le code à la première exécution, permettant une exécution quasiment aussi rapide qu'en C/C++.

**Présentation le JDK :**

Le JDK, Java Development Kit, contient toutes les bibliothèques logicielles permettant de programmer dans le langage JAVA, ainsi que les outils permettant de compiler le langage JAVA en bytecode qui est un langage compilé traduisible par la JVM.

**2.2 Oracle :**

Oracle est un SGBDR qui existe depuis de nombreuses années. Disponible sous Solaris (SUN), HP/UX (Hewlett Packard), Aix(IBM), Windows, Linux, etc. Très présent dans le monde industriel. Dispose d'un grand nombre d'outils lui permettant désormais de supporter la programmation en JAVA embarquée, le standard Corba, etc.

Le SGBD Oracle a bâti sa réputation autour de 3 points essentiels: (Pierre Marcenac consultant expert Oracle).

- La fiabilité, la robustesse de son moteur et la tolérance aux pannes ;
- L'acceptation de la charge, en parfaite adéquation avec des applications nécessitant de fortes capacités transactionnelles ;
- Et sa disponibilité sur un grand nombre de systèmes et de plateformes (Unix, Linux, Windows, systèmes 64-bits).

Dans Oracle y'a plusieurs versions à savoir : 7, 8, 8i, 9i, 10g ensuite la 11g et on a opté pour la 11g.

Avec Oracle 11g l'utilisation de ces nouvelles fonctions permet de simplifier étonnamment à la fois les applications et leur suivi, introduit une importante variété de nouveautés, dans les domaines de SQL, PL/SQL, l'administration, les performances et le tuning, la sécurité et le cryptage ou encore l'amélioration des outils de la console d'administration et de surveillance.

**2.3 Oracle SQL Developer :** est un environnement de développement intégré (EDI) multi plateforme, fourni gratuitement par Oracle Corporation et utilisant la technologie Java (Java Développement Kit). C'est un outil graphique permettant d'interroger des bases de données Oracle à l'aide du langage SQL.

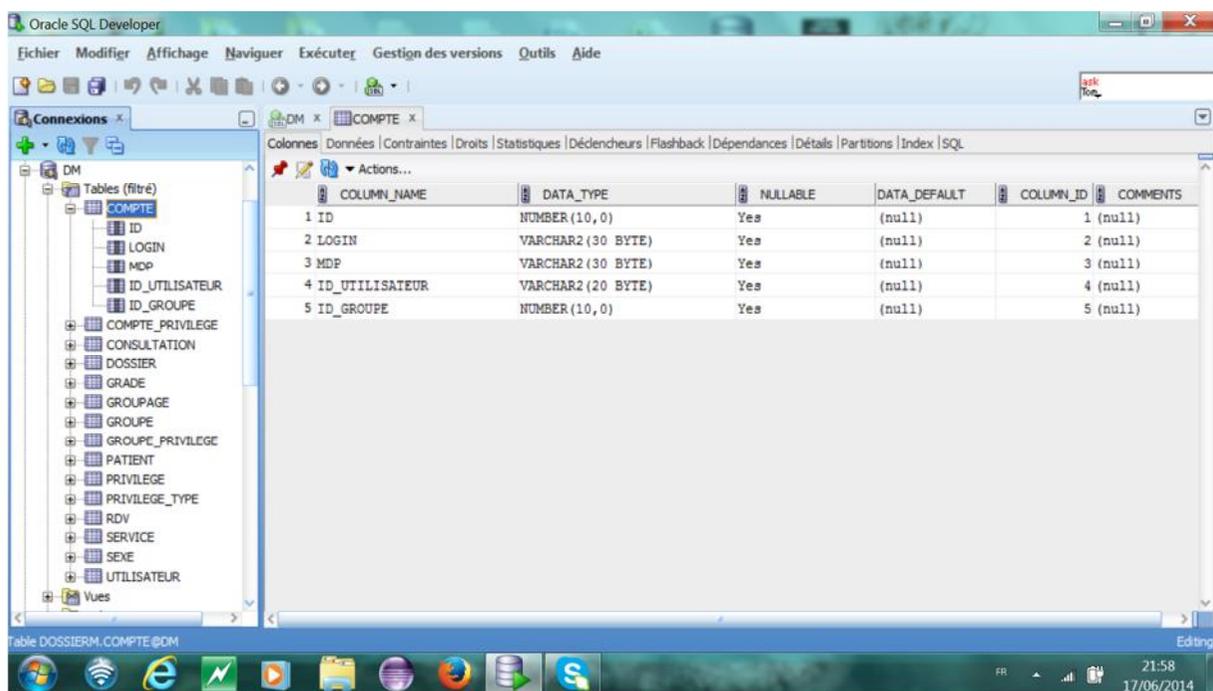


Figure : Oracle SQL Developer

**2.4 Eclipse (eclipse foundation)** : est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la Fondation Eclipse visant à développer un environnement de production de logiciels libres qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java.

Il existe plusieurs versions 3.3, 3.5, 3.6, 3.7, 4.2 et enfin La version 4.3 d'Eclipse appelée Kepler dans notre réalisation on a opter pour kepler cette version apporte quelque nouveautés majeur :

- Support de Java EE 7 pour Eclipse Web Tools (WTP);
- Déploiement d'Orion en version 3.0 pour les outils de développement web ;
- Nouvelle suite pour la gestion des processus métier Eclipse Stardust 1.0 ;
- Nouveau support pour BigData Eclipse Birt 4.3 intégrant le support des base de donnée MongoDB et Cassandra ;
- Amélioration de l'intégration du processus de revu de code avec Mylyn 3.9 ;
- Amélioration de l'intégration de Maven pour le développement JavaEE.

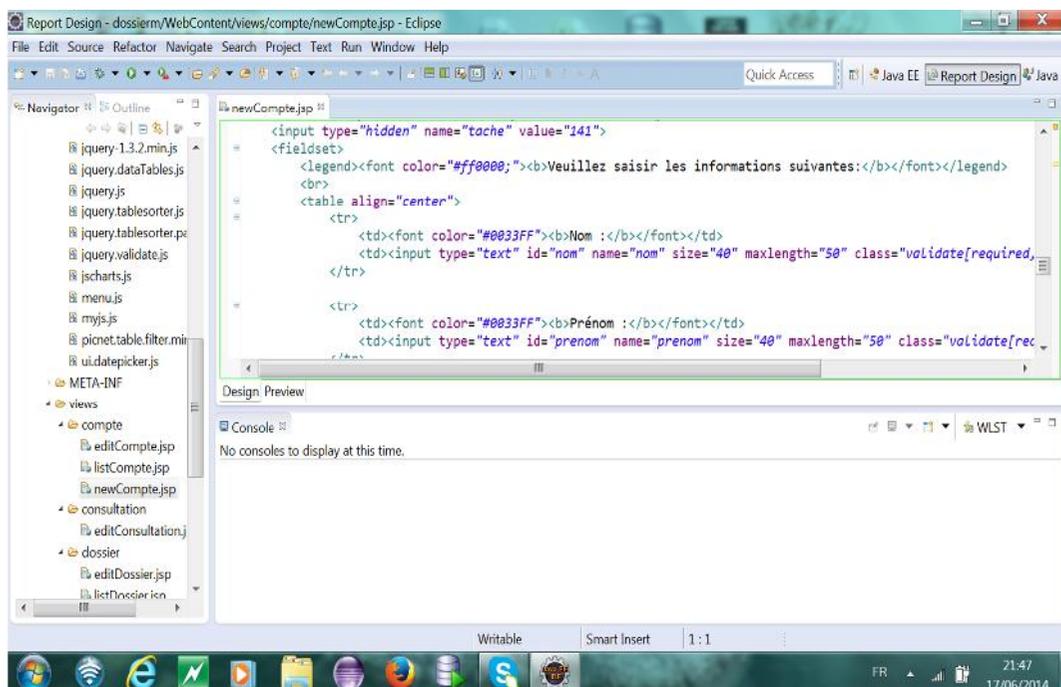
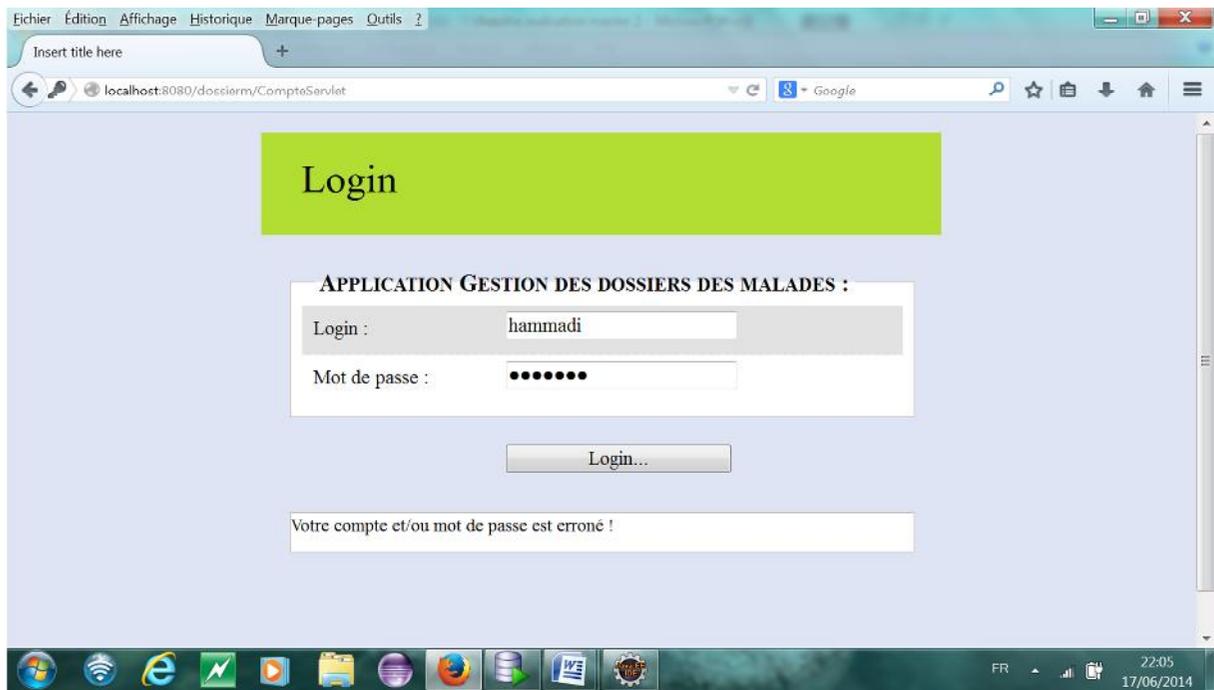


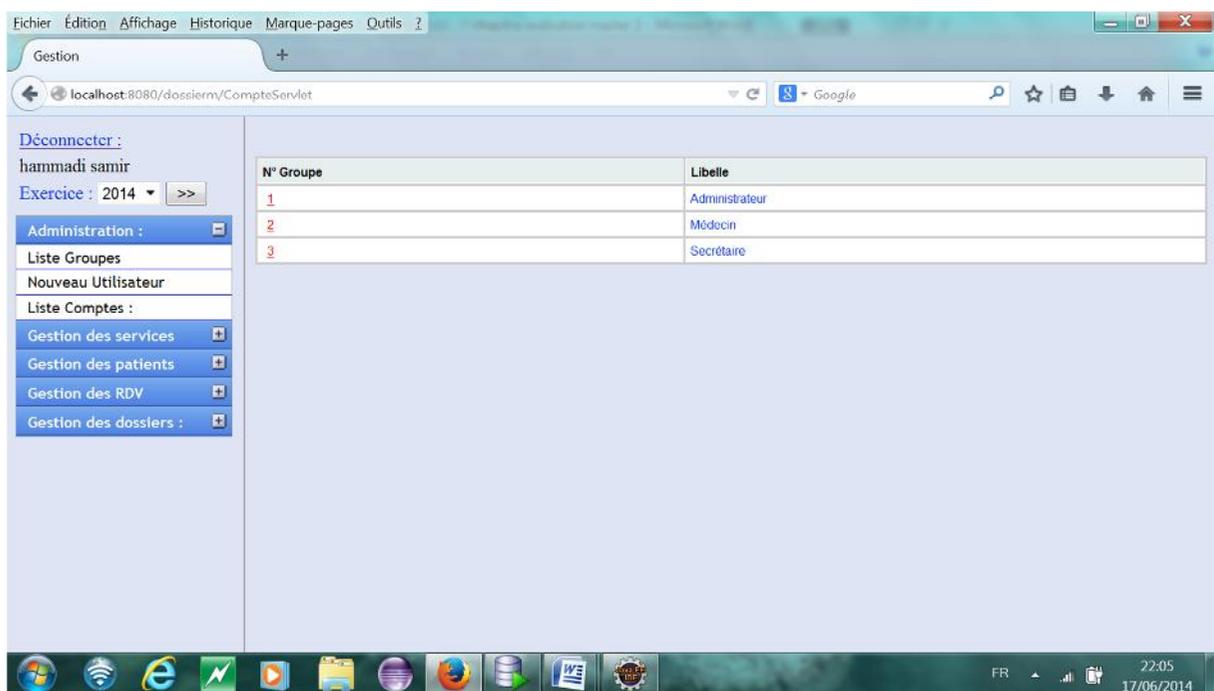
Figure : Eclipse JUNO

**2.5 Apache Tomcat** : est un conteneur web libre de servlets et JSP Java EE. Issu du projet Jakarta, c'est un projet principal de l'Apache Software Foundation. Il implémente les spécifications des servlets et des JSP du Java Community Process, est paramétrable par des fichiers XML et de propriétés, et inclut des outils pour la configuration et la gestion. Il comporte également un serveur\_HTTP.

**3. Présentation des interfaces :**



**Figure: authentification**



**Figure: liste Groupes**

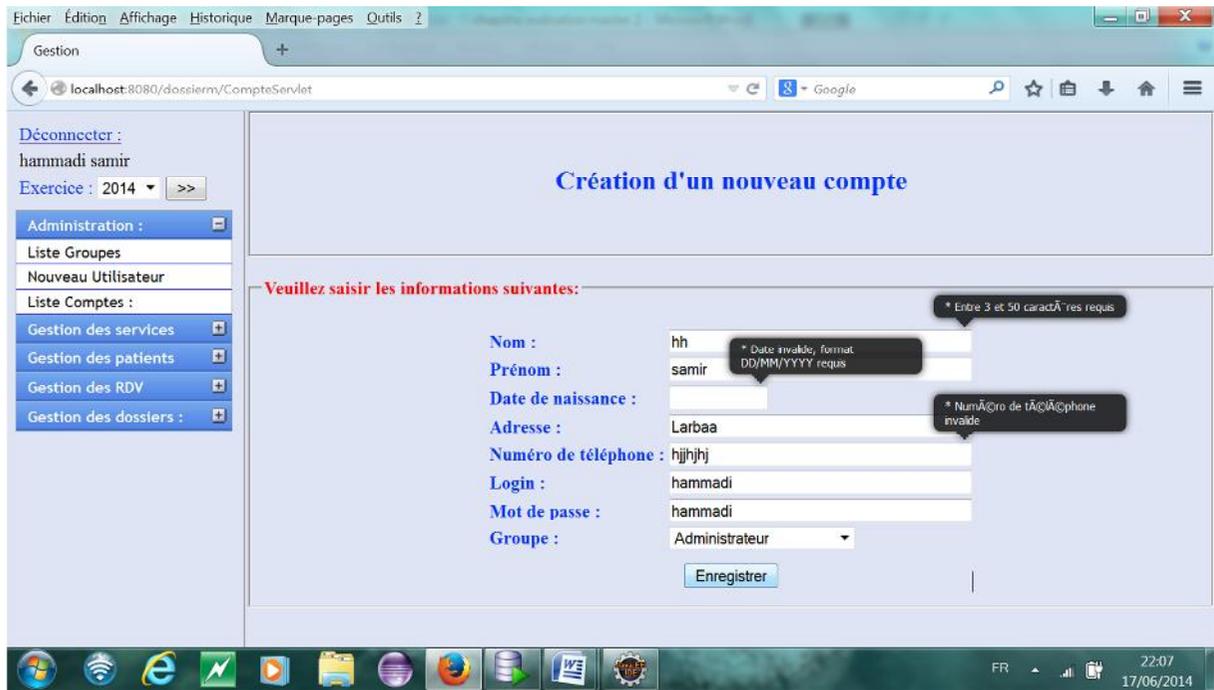


Figure : Nouveau utilisateur

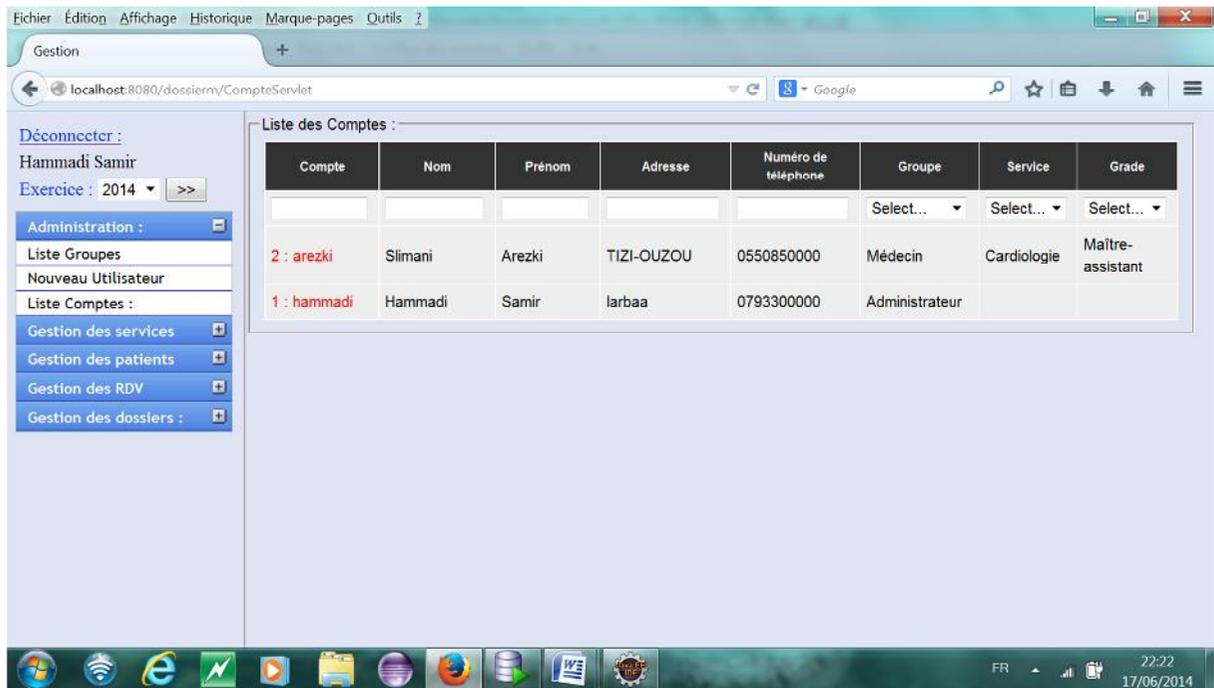


Figure : Liste des comptes

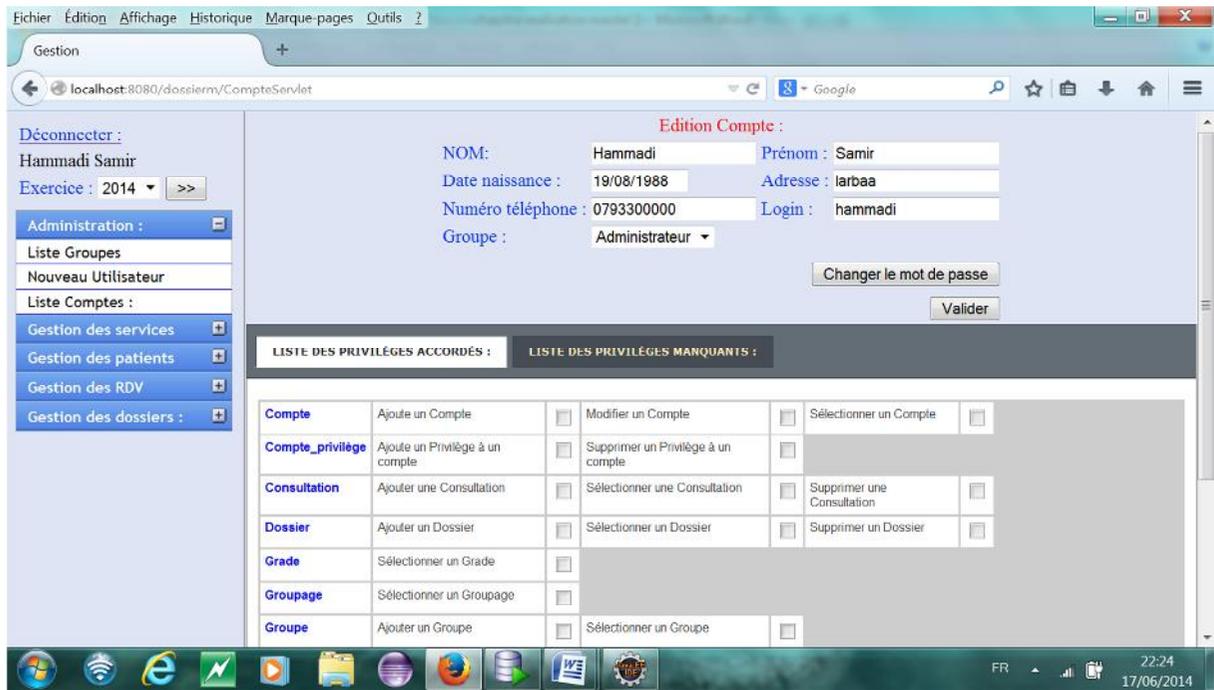


Figure : Modifier un compte

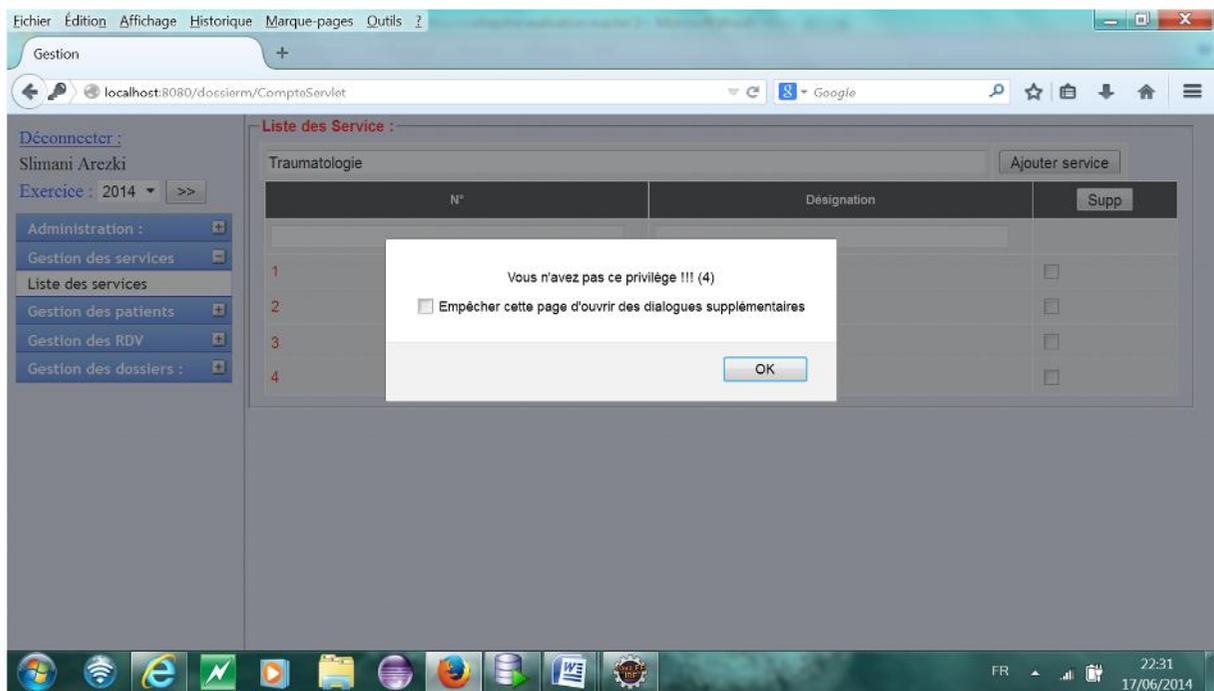


Figure : Nouveau service

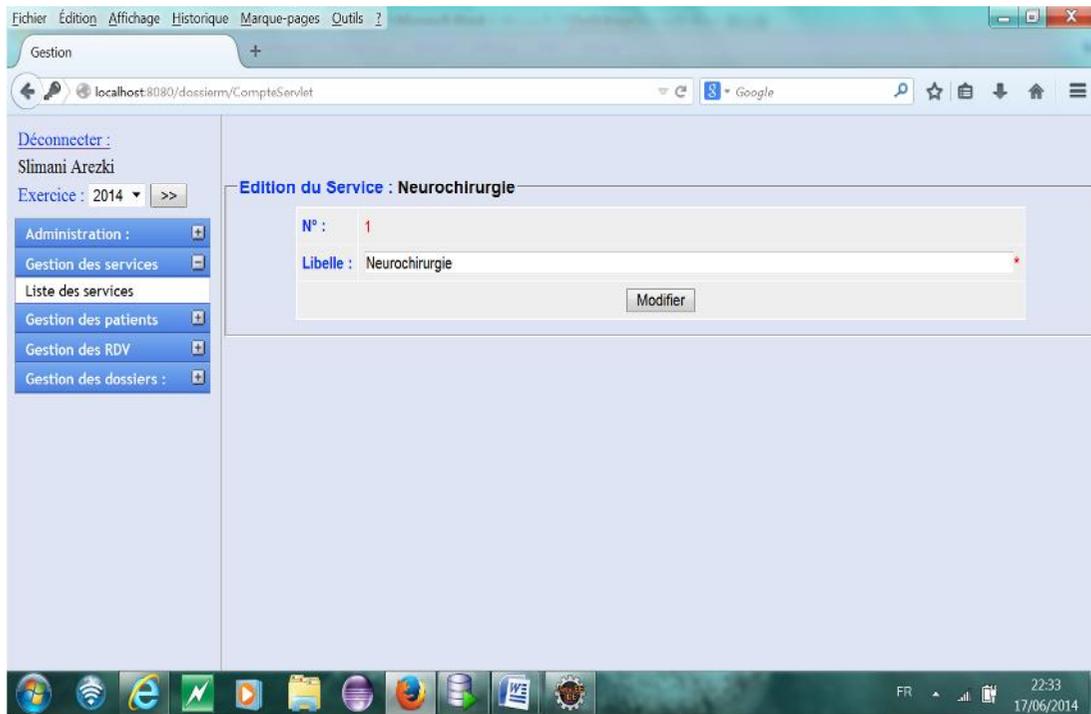


Figure : Modifier service

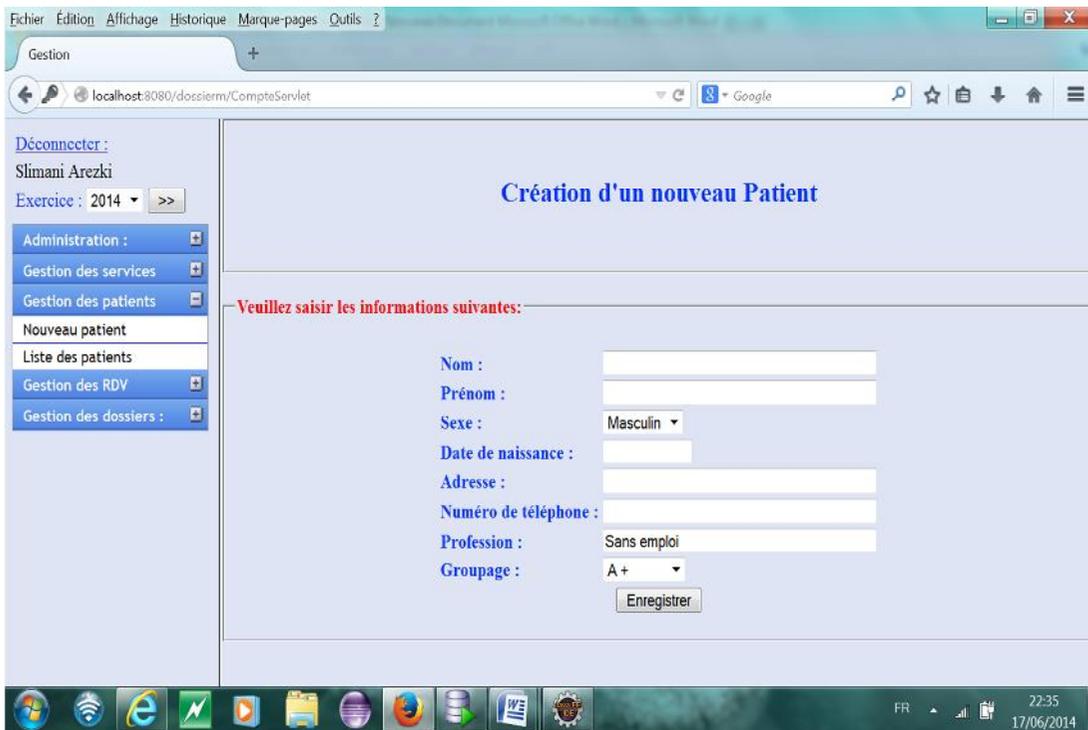


Figure : Nouveau patient

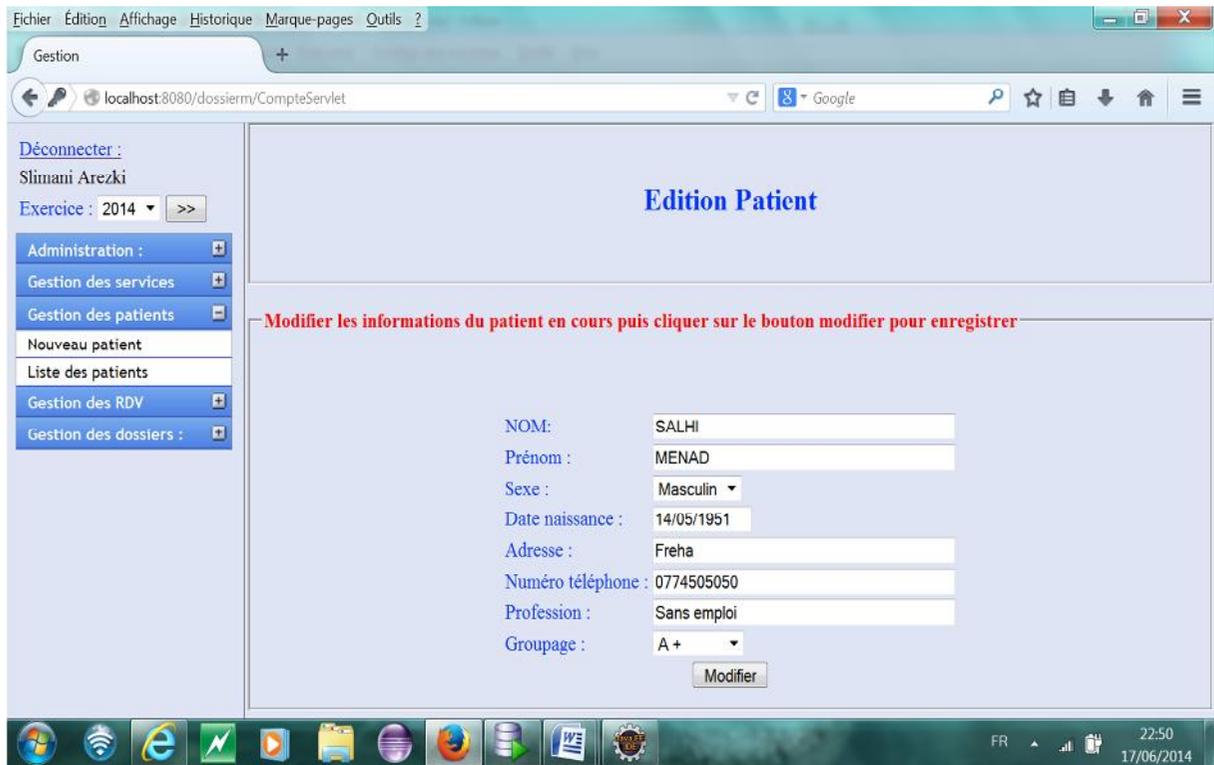


Figure : Modifier patient

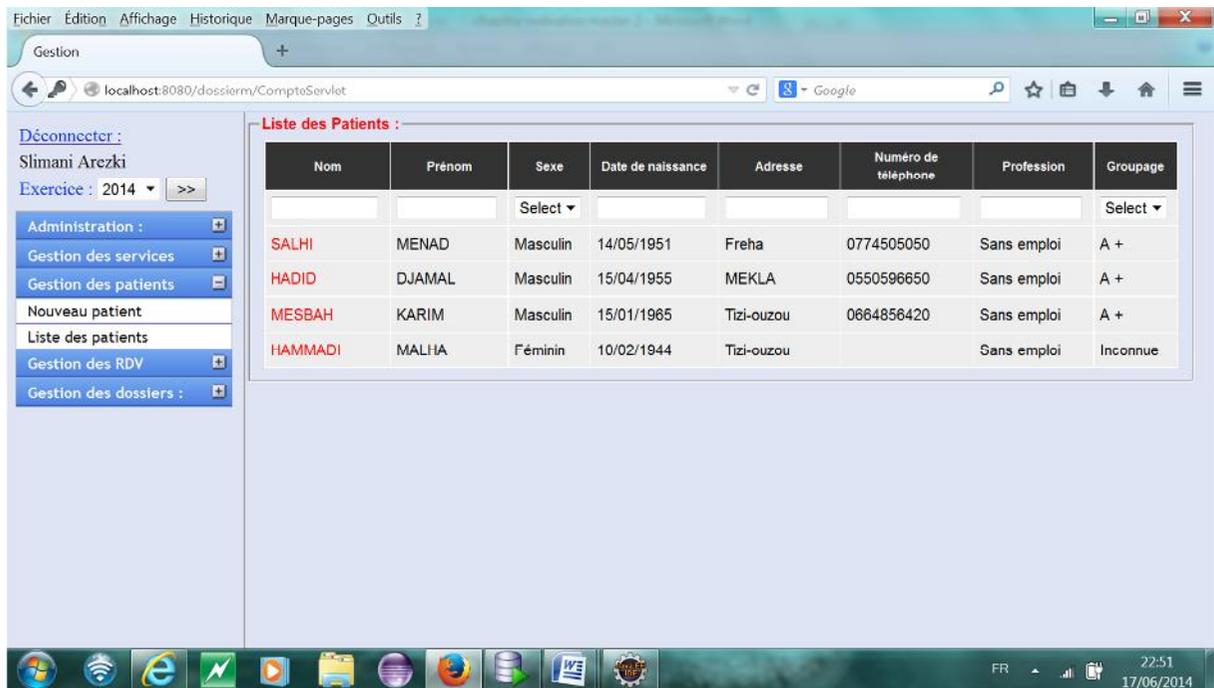


Figure : Liste des patients

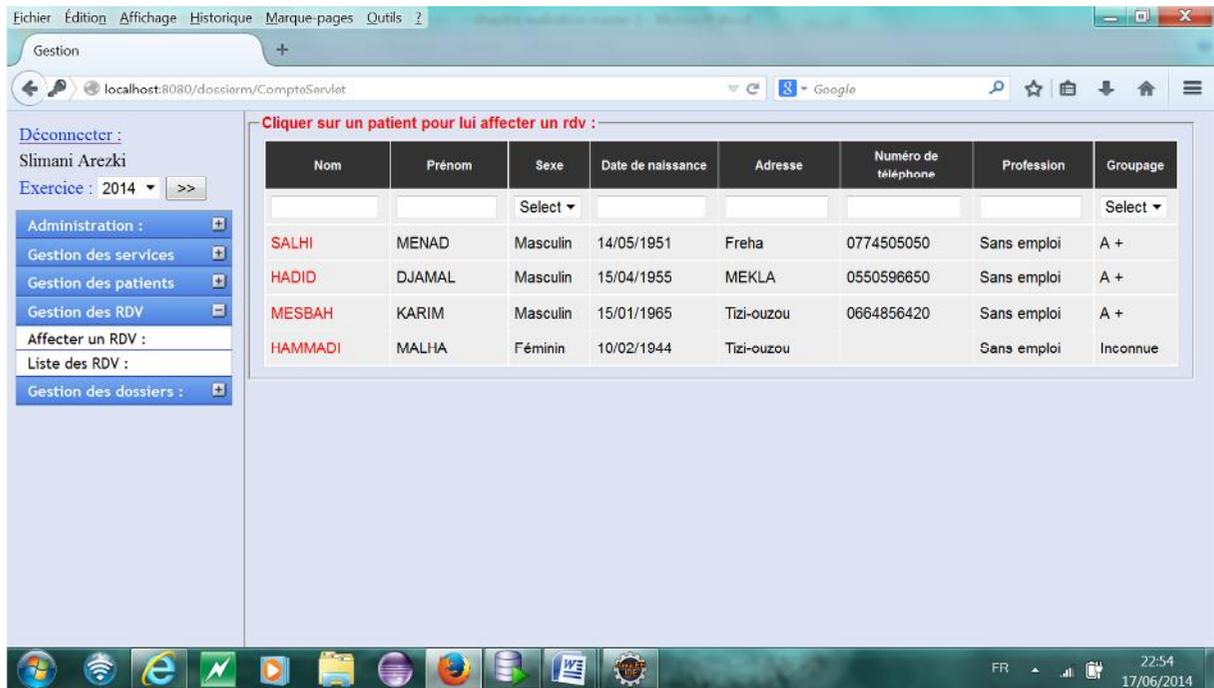


Figure : Liste des patients pour leur affecter des RDV



Figure : Nouveau RDV

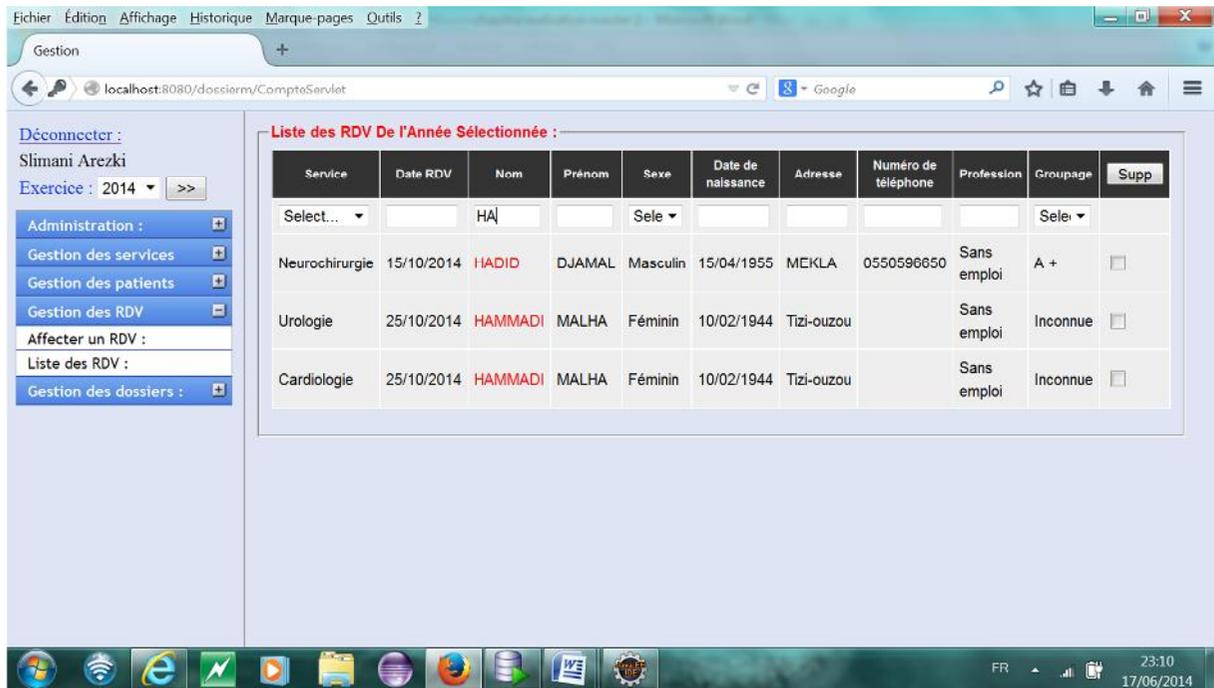


Figure : Liste des RDV de l'année sélectionnée

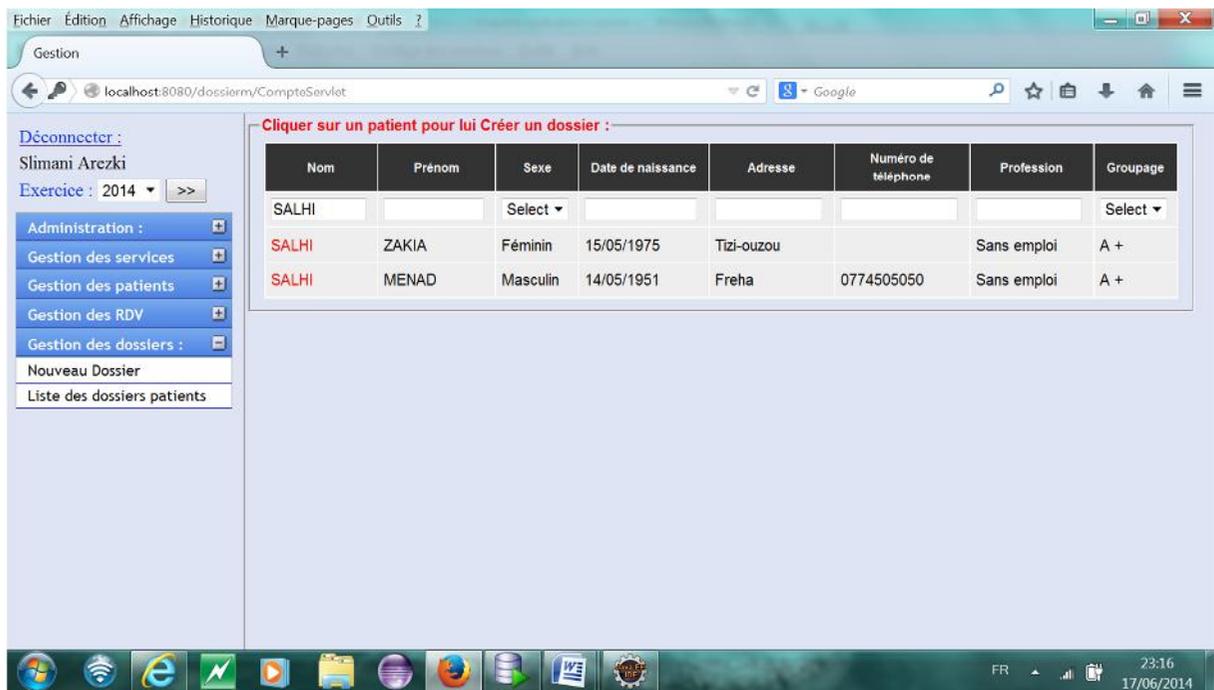


Figure : Liste des patients pour leur créer un dossier

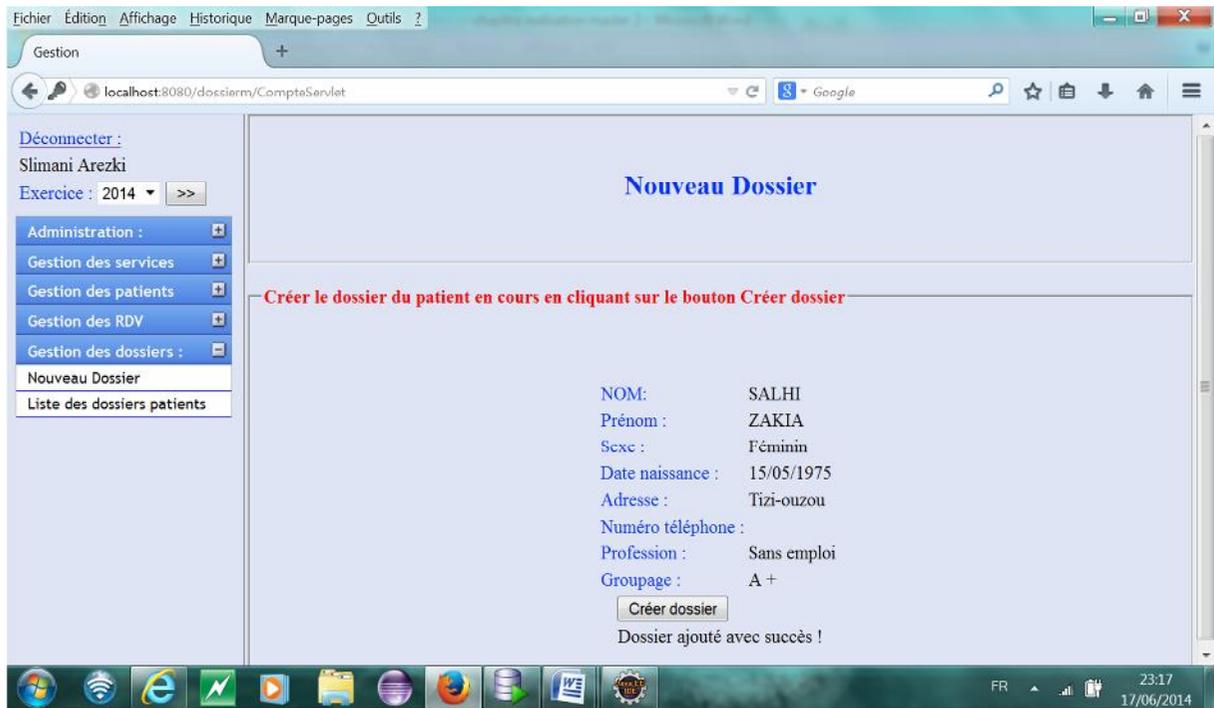


Figure : Nouveau Dossier

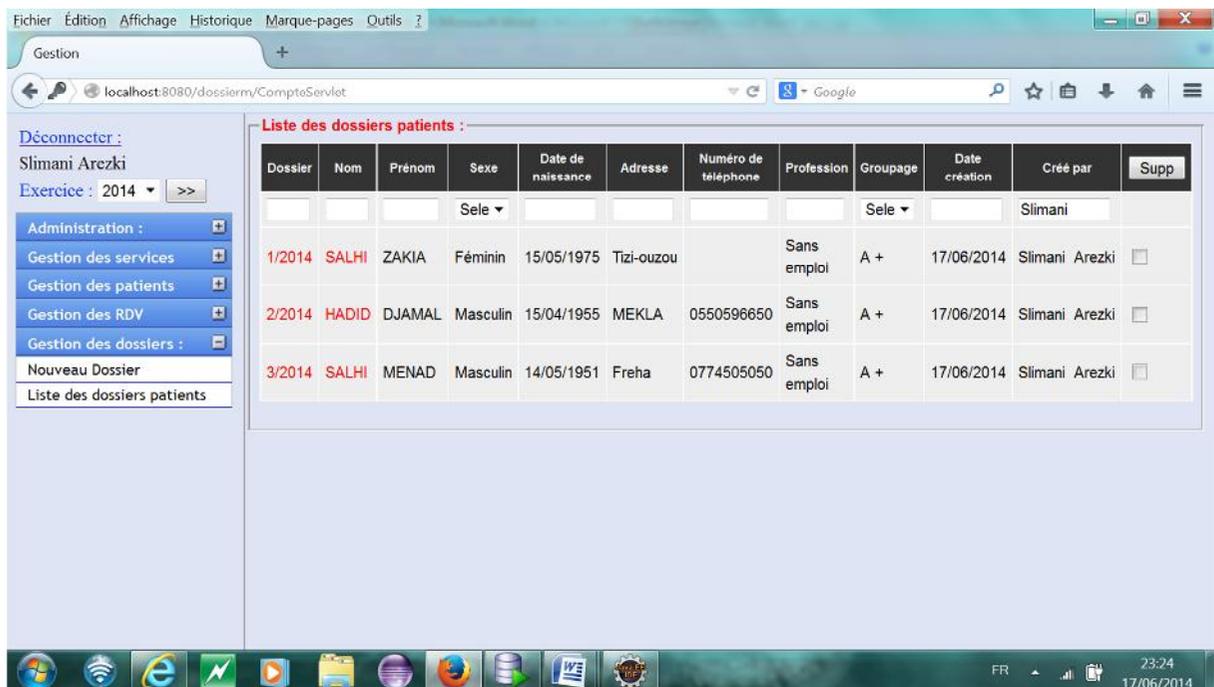


Figure : Liste des dossiers

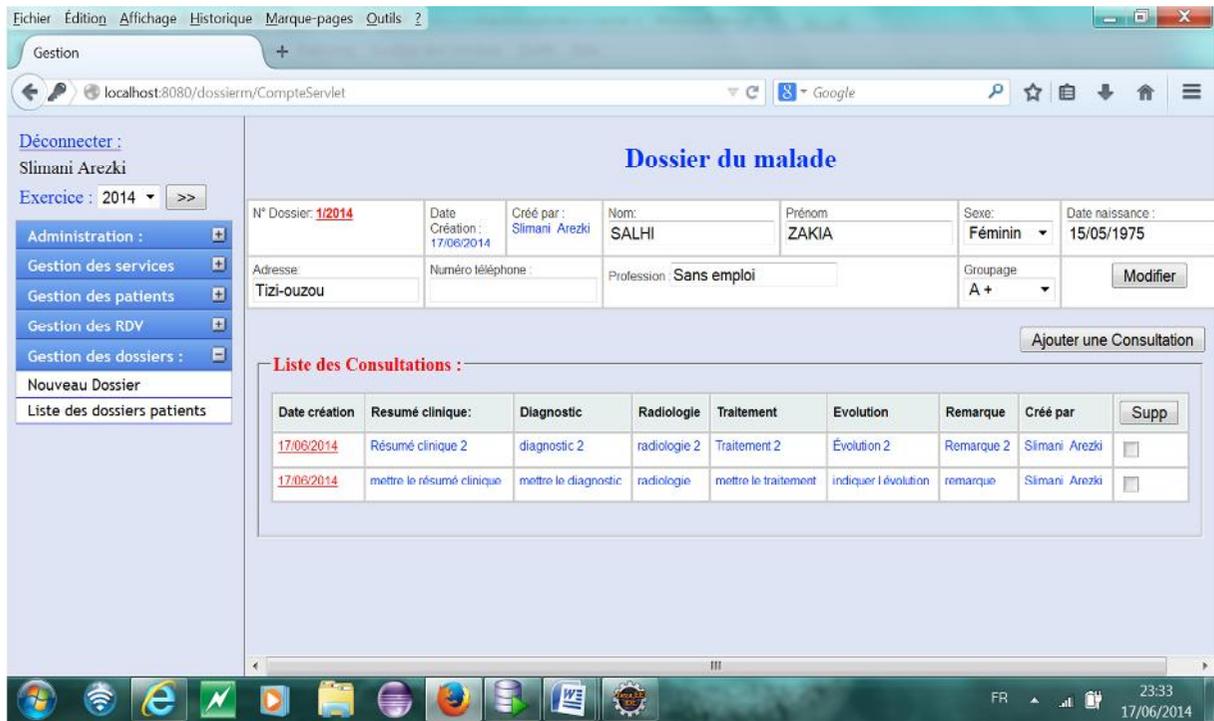


Figure : Dossier d'un malade

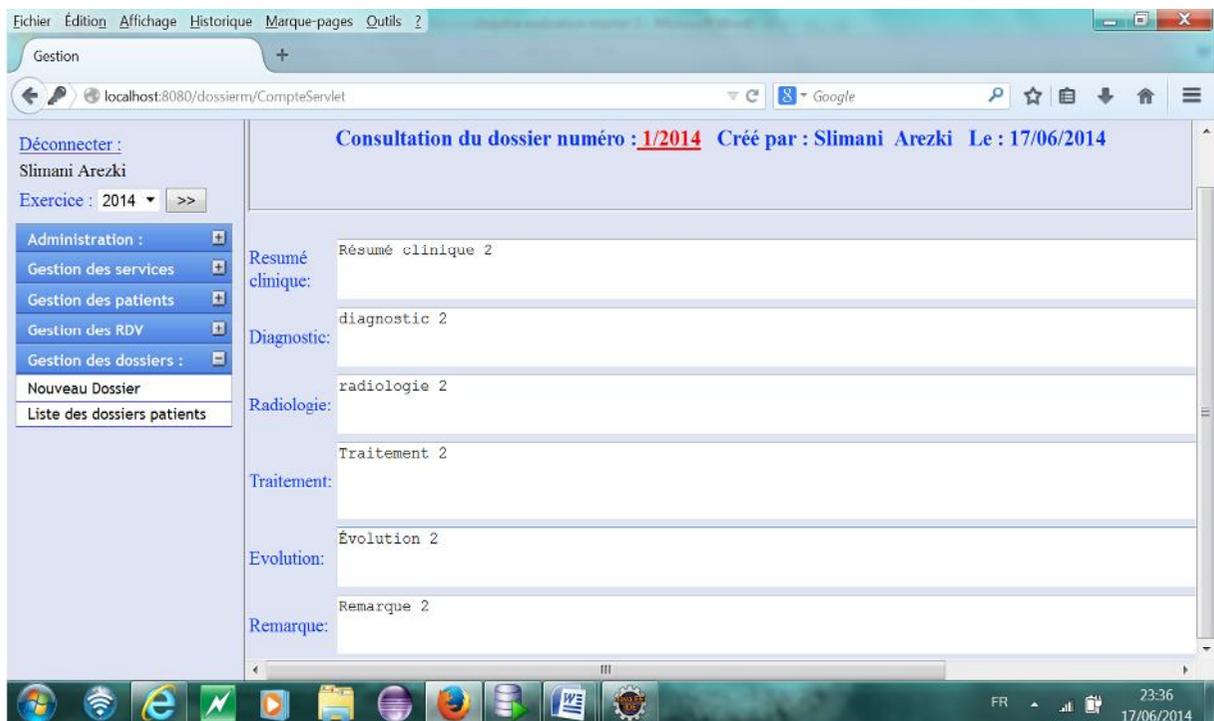


Figure : Contenu d'une Consultation

**4. Conclusion :**

Dans la première partie de ce chapitre, nous avons décrit les différents outils de développement à savoir le serveur de base de données Oracle, IDE SQL Developer, le JAVA, IDE Eclipse JUNO, ensuite le serveur d'application Apache Tomcat, en seconde partie j'ai présenté les interfaces de l'application.

*Conclusion*

*Générale*

## CONCLUSION GENERALE

Dans ce travail, nous avons développé une application permettant aux utilisateurs du service consultation spécialisée de gérer facilement certaines tâches. Pour le faire,

Nous avons d'abord présenté le domaine d'étude qui est la consultation spécialisée ensuite nous avons approfondi son étude en élaborant une étude de l'existant qui se traduit par l'étude des postes, étude des documents, et après avoir mené l'analyse de l'existant auprès de la consultation spécialisée nous avons constaté les insuffisances et les anomalies auxquelles nous avons apporté des suggestions, ensuite nous avons proposé une solution informatique de type client/serveur avec deux ordinateurs pour chacun des deux postes (le médecin, le secrétaire) qui seront des clients, reliés à un serveur qui se situera au niveau de l'administrateur par un réseau local, ensuite, nous avons présenté les réseaux ou nous avons défini les différents réseaux qui existent ce qui nous a amené à opter pour l'architecture 3 tiers qui se définit par l'utilisation d'un navigateur web, serveur d'application et une base de données. Pour développer notre application nous sommes passés à l'analyse et conception où nous avons défini différents diagrammes UML grâce auxquels nous avons réalisé notre conception et en fin nous sommes passés à la réalisation dans laquelle nous avons défini les différents outils de développement Eclipse, JDK, Oracle, SQL Developer, le langage de programmation en l'occurrence Java.

Ce travail nous a fait bénéficier sur les différents plans d'abord la mise en pratique de nos connaissances théoriques que nous avons acquises durant notre cursus plus particulièrement UML, nous a permis d'avoir une expérience dans le domaine des entreprises. D'avoir une expérience dans la conception orientée objet et aussi de programmer en Java qui est un langage orienté objet.

Références

bibliographiques

## Références bibliographiques

**Chapitre I:** site web CHU :<http://www.chutodz.net/consultation/consultation.php>.

**Chapitre II:** Client-serveur :le modèle client-serveur Olivier Aubert

Architectures Client-Serveur Bernard ESPINASSE Professeur à l'Université d'Aix-Marseille 2011

Le modèle client-serveur introduction Christian Bulfone

C.Crochepeyre applications CS

### **Chapitre III:**

Introduction à UML Shebli Anvar – DSM/DAPNIA – CEA Saclay François Terrier,  
Sébastien Gérard DRT/LIST – CEA/Saclay.

Robert Ogor [www.enst-bretagne.fr](http://www.enst-bretagne.fr)

Conception orientée-Objetsavec UML Rémi Bastide IRIT – CUFR J.F. Champollion  
<http://ihcs.irit.fr/bastide>

<http://commons.wikimedia.org/wiki/File:OO-historie.jpg>

Introduction à UML 2.0 F.-Y. Villemin, CNAM

Pascal ROQUES. – UML 2 par la pratique

### **Chapitre IV :**

Oracle: claude duvallet Maître de Conférences en Informatique.

eclipse fondateur : eclipse foundation ; site web : [www.eclipse.org](http://www.eclipse.org).

Apache tomcate fondateur : Fondation Apache ; site web:[tomcat.apache.org](http://tomcat.apache.org).

Java fondateur : Oracle Corporation; site web:[www.java.com](http://www.java.com).