

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEINGNEMENT SUPERIEUR ET DE LA
RECHERCHE SIENTIFIQUE



Université Mouloud Mammeri de Tizi-Ouzou
Faculté de Génie Electrique et d'Informatique
Département d'Informatique



Mémoire de fin d'étude

En vue de l'obtention d'un diplôme en Master informatique

Spécialité « Systèmes informatiques »

Thème :

conception et réalisation d'une Application pour la gestion d'un Cabinet Médical
cas: cabinet de Dermatologie

Réalisé par :

- M^{elle} Aouchiche Khedidja
- M^{elle} Ararbi Amina

Dirigé par :

Mr S. Sadou

Promotion : 2016-2017

Remerciements

Nous remercions tout d'abord Dieu qui nous a donné la foi et le courage pour accomplir ce projet.

Nous tenons aussi à exprimer notre reconnaissance et profonde gratitude à notre promoteur Mr Sadou pour nous avoir encadré durant cette année, pour sa forte présence et sa disponibilité, pour son exigence scientifique et ses précieuses orientations méthodologiques, pour son encouragement et sa patience.

Que les membres du jury trouvent ici nos plus vifs remerciements pour avoir accepté d'honorer par leur jugement notre travail. Aussi, nous adressons nos remerciements à tous nos enseignants de l'UMMTO pour nous avoir appris le goût de l'effort et du travail.

Un grand merci aussi à toute personne qui de près ou de loin a contribué à ce que cet humble travail voit le jour.

Dédicaces

Je dédie ce modeste travail

A la mémoire de mon père

A celle qui as éclairé ma vie, ma très chère mère pour son aide et son soutien tout au long de mes études, et qui as fait de moi ce que je suis aujourd'hui

A mes frères

A ma sœur, son mari et leurs ange Ritedj

A mon cher oncle et sa femme

A mes cousins et cousines

A mes ami(e)s particulièrement groupe « par hasard »

A ma binôme Khedidja

Amina

Dédicaces

*Je dédie ce modeste travail
À mes très chers parents.*

À mes chers frères et sœurs : Mohamed, Hocine,

Fatima, Djamilâ

À mon cousin préféré Mourad ainsi que toute ma famille.

À Mes ami(e)s.

À Ma binôme Amina.

Khedidja

Sommaire

Sommaire

Introduction générale.....	1
----------------------------	---

Chapitre I : Etude de l'existant

I.1 Introduction :.....	2
I.2. Présentation de l'organisme d'accueil :	2
I.2.1-Présentation de l'organisme d'accueil :	2
I.2.1.1.Cabinet médical :	2
I.2.1.1.1-Définition d'un cabinet médical :	2
I.2.1.1.2.-Classification des cabinets médicaux :	3
I.2.1.1.2.1-Le cabinet médical individuel :	3
I.2.1.1.2.2-Le cabinet médical communautaire :	3
I.2.1.1.2.3-Le cabinet médical collectif :	3
I.2.1.1.2.4-Le cabinet du médecin chef :	4
I.2.2-Présentation du domaine d'étude :	4
I.3. Etude de l'existant :.....	5
I.3.1-Etude des postes de travail :	5
I.3.2-Etude des documents existants :	7
I.4. Graphe des flux d'information :.....	12
I.4.1- Définition:.....	12
I.4.2-Concepts utilisés :	12
a-Acteur :	13
b-flux d'information:.....	13
I.4.3Graphedesflux d'information de l'organisme d'accueil :	13
I.5. Critiques de l'existant :.....	14
I.6. Proposition de solution :.....	15

I.7. Conclusion :	15
-------------------	----

Chapitre II : Analyse Et Conception

II.1. Introduction :	16
II.2. Les Méthodes de conception :	16
II.2.1- Présentation d'UML:	17
II.2.1.1- Définition :	17
II.2.1.2- Les diagrammes d'UML :	17
1-Diagrammes fonctionnels :	17
2- Diagrammes statiques :	17
3- Diagrammes dynamiques:	18
II.2.2-Utilités d'UML :	18
II.2.3- Formalisme d'UML :	18
II.3. Démarche de modélisation à suivre :	19
II.4. L'Analyse ::	20
II.4.1-Identification des acteurs ::	20
II.4.2- Diagramme de contexte :	20
II.4.2.1- Définition :	20
II.4.2.2- Diagramme de contexte de notre application.....	21
II.4.3- Elaboration de modèles de cas d'utilisation :	21
II.4.3.1- Diagramme des cas d'utilisation :	21
II.4.3.1.1- Définition:	21
II.4.3.1.2- Identification des cas d'utilisation	23
II.4.3.1.3- Diagramme de cas d'utilisation général :	25

II.4.4- Description textuelle des cas d'utilisation :.....	25
II.5. La conception :.....	43
II.5.1- Diagramme de séquence :.....	43
II.5.1.1- Définition :.....	43
II.5.1.2- Diagramme de Séquence de quelques cas d'utilisation :.....	44
II.5.2- Diagramme de classe :.....	49
II.6. Conclusion :.....	51

Chapitre III : Implémentation et Réalisation

III.1.	
Introduction :.....	52
III.2. Le modèle physique des données : :.....	52
III.3. Description de l'environnement de travail :	57
III.3.1-Les outils de développement :.....	57
III.3.1.1-Le langage de programmation (java) :.....	57
III.3.1.2-IDE (NetBeans) :.....	59
III.3.1.3-L'interface phpMyadmin :.....	60
III.3.1.4-WampServer :.....	60
III.3.1.5-Le Langage de requêtes SQL ::.....	61
III.4. Implémentation de L'application :.....	61
III.4.1-Partie stockage :	61
III.4.2-Partie traitement :	62
III.4.2.1- Le Projet Cabinetmedical :.....	62
III.4.3-PartiePrésentation :.....	63
III.5. Conclusion :.....	71

Conclusion générale72

Références

Introduction Générale

Introduction Générale

L'informatique est une science qui est impliquée aujourd'hui dans la plupart des domaines de la vie, son utilisation connaît de nos jours une grande expansion. Elle intervient dans les différents secteurs tels que l'enseignement, l'administration, le commerce ...et le secteur de la santé ne fait pas défaut. Les difficultés rencontrées au quotidien dans la gestion du nombre important de patients et leurs dossiers, dans la communication au sein du corps médical pourrait être contournée par l'utilisation de l'informatique.

Dans ce cadre s'inscrit notre projet de fin d'études qui consiste à réaliser une application pour la gestion d'un cabinet médical.

Afin de mener à bien notre projet, nous avons réparti le contenu de notre travail en chapitres, ainsi :

- Le premier chapitre s'intitule « Etude de l'existant » Dans ce chapitre nous présenterons la cabinet médical dermatologique « Lasers Dermatologiques » qui a été notre organisme d'accueil, nous allons décrire ses postes de travail ainsi que les différents documents existants puis nous représenterons le diagramme de flux d'informations échangé entre les acteurs du système.
- Le deuxième chapitre sous le nom « analyse et conception », qui est consacré à l'analyse et la conception de notre application, pour cela nous avons opter pour le langage UML, pour se faire, nous avons utilisée quelques diagrammes :
 - Diagramme de cas d'utilisation pour la représentation des besoins des utilisateurs
 - Diagramme de séquence pour la représentation des cas d'utilisation
 - Diagramme de classe pour la représentation des objets du système
- Le troisième et le dernier chapitre « Réalisation et implémentation », comporte quand à lui la présentation de l'environnement dont lequel notre application a été réalisé, les outils utilisés et quelques interfaces de notre application.

Chapitre I

Etude de l'existant

I.1. Introduction :

L'étude de l'existant est la première étape de la conception d'un système d'information, car pour proposer des solutions, il est impératif de bien comprendre le fonctionnement du système existant. Le but de cette étape est d'avoir une vue générale sur l'organisme d'accueil et de prendre en compte tous les moyens de traitement d'information utilisés, la circulation de ces informations, ainsi que les paramètres humains et organisationnels de la structure. Cette étude se conclue par un diagnostic de la structure actuelle pour pouvoir apporter des solutions aux problèmes rencontrés.

Ce chapitre contient essentiellement :

- ☆ Une présentation de l'organisme d'accueil.
- ☆ Etude de l'existant.
- ☆ Critique de l'existant.
- ☆ Proposition de solution.

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

I.2.1. Présentation de l'organisme d'accueil :

Lasers Dermatologiques est un cabinet médical communautaire spécialisé dans la dermatologie, situé à: 1Rue des frères BEGGAZ-bloc C n°2 – Nouvelle ville TIZI-OUZOU.

Ce cabinet a une importante activité autour des maladies de la peau et la chirurgie dermatologique.

I.2.1.1. Cabinet médical :

I.2.1.1.1. Définition d'un cabinet médical:

Un cabinet médical est un local où est exercé une profession libérale liée aux soins de santé (exemples : radiologue, chirurgien-dentiste, dermatologue). Le médecin proposera dans ses locaux des consultations liées à sa spécialité.

Un cabinet médical sera toujours constitué d'au moins une salle de consultation (qui fera parfois office de bureau pour le médecin spécialiste), d'une salle d'attente pour les patients, et parfois d'un /une secrétaire médical(e) sera présente pour renseigner prendre les rendez vous des patients.[1]

I.2.1.1.2. Classification des cabinets médicaux:

Les cabinets médicaux sont classifiés comme suit :

I.2.1.1.2.1. Le cabinet médical individuel :

Le médecin dispose de son propre cabinet, de locaux qui correspondent à ses besoins d'équipements et d'appareils. Il est le seul responsable du fonctionnement du cabinet, et emploie selon sa taille un ou plusieurs employés ou assistantes médicales.

I.2.1.1.2.2. Le cabinet médical communautaire :

Deux ou plusieurs médecins exerçant dans des branches similaires dirigent ensemble un grand cabinet médical en partageant le personnel et en utilisant une infrastructure. Comme (exemple : laboratoire, salle de radiologie). Ils forment une société simple ou une société en nom collectif et établissent en son nom un contrat avec les patients, travaillent ensemble directement ou indirectement et émettent des factures communes.

I.2.1.1.2.3. Le cabinet médical collectif :

Le cabinet collectif est le modèle de cabinet le plus « classique » et rassemble deux ou trois médecins exerçant dans différentes branches ou de même spécialisation.

Un cabinet collectif est le regroupement de plusieurs médecins qui vont partager la même infrastructure sans pour autant coordonner leurs pratiques médicales. Chaque médecin exerce alors pour son propre compte. Il établit un contrat avec le patient en son seul nom, décide seul des traitements à prescrire et émet ses propres factures.

Chapitre 1 : Etude de l'existant

Seuls l'infrastructure et le personnel du cabinet médical (laborantine, assistante en radiologie, secrétaire et/ou assistante médicale) sont partagés. Chaque médecin dispose souvent d'une ou de plusieurs assistantes médicales qui gèrent les consultations, l'accueil et l'administration. Ce modèle autorise aussi la collaboration avec d'autres professions médicales. Il n'y a aucune obligation de lien entre les spécialités médicales représentées dans un cabinet collectif doté d'une petite infrastructure. Les membres d'un cabinet collectif peuvent ainsi avoir chacun son propre groupe de patients.

I.2.1.1.2.4. Le cabinet du médecin chef :

Le cabinet est ici dans un hôpital. Le médecin assure des soins ambulatoires et stationnaires et ses horaires de consultation sont par conséquent limités à quelques jours. Les analyses en laboratoire, radios et thérapies sont effectuées par le personnel de l'hôpital. L'assistante médicale s'occupe ici principalement de tâches administratives et organisationnelles. Les patients sont en général envoyés par autre médecin.

I.2.2. Présentation du domaine d'étude :

Un cabinet médical contient des médecins et des secrétaires chacun a ces activités

☆ Les activités du médecin :

- ✓ L'auscultation des patients.
- ✓ La mise à jour des dossiers des patients.
- ✓ Délivrance des ordonnances.
- ✓ Délivrance des certificats médicaux.
- ✓ La direction du cabinet médical.
- ✓ Superviser et veiller au bon fonctionnement du cabinet.

☆ Les activités de la secrétaire médicale :

- ✓ L'accueil des patients, et l'organisation de leurs passages dans la salle des consultations.

Chapitre 1 : Etude de l'existant

- ✓ Préparation et classification des dossiers des patients.
- ✓ Noter les rendez-vous.

I.3. Etude de l'existant :

I.3.1. Etude des postes de travail :

L'étude des postes de travail nous aidera à comprendre l'organisation qui existe et peut être nous aider à déceler les postes surchargés, et les tâches effectuées, les responsabilités et les documents qui passent par les différents postes, ainsi que les principaux défauts de l'organisation existants.

Pour rendre notre étude plus bénéfique pour les informations que nous traitant, une fiche d'étude des postes de travail sera utilisée et contiendra les informations suivantes :

☆ **Fiches d'analyse des postes de travail :**

La figure suivante représente la fiche d'analyse du poste de travail N°1 (Médecin) :

Fiche d'analyse de poste N°1	
Code : DRS Désignation : Médecin. Service de rattachement : / Effectifs : 02 Tâches à accomplir -Soigne les malades : <ul style="list-style-type: none">▪ Soigne les maladies de la peau et allergiques.▪ Effectue les chirurgies dermatologiques.▪ Effectue une épilation électrique.▪ Etablie les soins esthétiques.▪ Etablie une cryothérapie.▪ Création des dossiers médicaux. -Délivre des ordonnances. -Délivre des certificats médicaux	
Documents entrants au poste	
Désignation	Origine

Chapitre 1 : Etude de l'existant

-Le dossier médical avant la mise à jour. -Les fiches résultat d'analyse.	-Secrétaire. -Patient.
Documents sortants du poste	
Désignation	Destinataire
-Ordonnance. -Le dossier médical après la mise à jour. -Certificat médical. -Fiche d'analyse.	-Patient -Secrétaire. -Patient. -Patient.

Figure I.1 : Fiche d'analyse du poste de travail n°1 (Médecin)

La figure suivante représente la fiche d'analyse du poste de travail N°2 (Secrétaire) :

Fiche d'analyse de poste de travail N° 02	
Code : SC	
Désignation : Secrétaire.	
Service de rattachement : Médecin.	
Effectifs : 01	
Tâches à accomplir :	
<ul style="list-style-type: none"> ▪ Accueille et aide le malade. ▪ Prise de rendez-vous. ▪ Orientation des malades destinés à la chirurgie. ▪ Classement des dossiers médicaux par ordre alphabétique. ▪ Classement des fiches d'analyse effectuées par les patients. 	
Documents entrants au poste	
Désignation	Origine
-Les fiches d'analyse des patients. -Le dossier médical après la mise à jour	-Médecin -Médecin
Documents sortants du poste	
Désignation	Destinataire
-Le dossier médical avant la mise à jour	-Médecin

Figure I.2 : Fiche d'analyse du poste de travail n°2 (Secrétaire)

I.3.2. Etude des documents existants :

Pour mieux comprendre la circulation des flux d'informations dans l'organisation il est important d'étudier tous les supports d'information utilisés, à savoir les documents. Il existe deux types de documents :

-Documents internes : Ce sont des documents utilisés seulement à l'intérieur du champ d'étude.

-Documents externes : Ce sont des documents qui sont destinés utilisés ou qui proviennent de l'extérieur du champ d'étude.

Cette étude va nous permettre d'avoir une vue global sur les informations manipulées et d'évaluer le cas du système existant.

☆ Liste des documents existants :

Numéro	Désignation
01	Dossier médical
02	Ordonnance
03	Certificat médical
04	Fiche d'analyse
05	Fiche résultat d'analyse

Figure I.3 : Tableau de la liste des documents existants

Les abréviations utilisées pour les types de données :

A : Alphabétique

N : Numérique

AN : Alphanumérique

D : Date

☆ **Fiches d'analyse des documents :**

La figure suivante représente la fiche d'analyse du document N°1(Dossier Médical) :

Fiche d'analyse du document N°1			
I. Identification			
Code : DM.			
Désignation : Dossier Médical.			
Nature de document : interne.			
Nombre d'exemplaire : 01.			
Utilité : présente le patient			
II. Description des rubriques			
Rubrique	Type	Taille	Observation
Nom patient	A	10	/
Prénom patient	A	10	/
Age	N	2	/
Fonction	A	20	/
Adresse	AN	20	/
Nom maladie	A	10	/
Nom de traitement	A	100	/

Figure I.4 : Fiche d'analyse du Dossier Médical

Chapitre 1 : Etude de l'existant

La figure suivante représente la fiche d'analyse du document N°2(Ordonnance) :

Fiche d'analyse du document N°2			
I. Identification			
Code : OR			
Désignation : Ordonnance.			
Nature de document : externe.			
Emetteur : médecin.			
Destinataire : patient.			
Nombre d'exemplaire : 01.			
Utilité : décrit le traitement nécessaire.			
II. Description des rubriques			
Rubrique	Type	Taille	Observation
Nom médecin	A	10	/
Prénom médecin	A	10	/
N° tel médecin	A	10	/
Adresse	AN	20	/
Nom patient	A	10	/
Prénom patient	A	10	/
Age	N	2	/
Date	D	10	JJ/MM/AAAA
Nom de traitement	A	100	/

Figure I.5 : Fiche d'analyse de L'Ordonnance

Chapitre 1 : Etude de l'existant

La figure suivante représente la fiche d'analyse du document N°3(Certificat Médical) :

Fiche d'analyse du document N°3			
I. Identification			
Code : CM. Désignation : Certificat Médical.. Nature de document : externe. Emetteur : médecin. Destinataire : patient. Nombre d'exemplaire : 01. Utilité : justifier les absences de travail, d'étude...			
II. Description des rubriques			
Rubrique	Type	Taille	Observation
Nom médecin	A	10	/
Prénom médecin	A	10	/
Adresse	AN	20	/
N° tel médecin	N	10	/
Nom patient	A	10	/
Prénom patient	A	10	/
Date	D	10	JJ/MM/AAAA
Nbr de jours d'arrêt de travail	N	2	Date fin-Date début d'arrêt
Date début d'arrêt	D	10	JJ/MM/AAAA
Date fin d'arrêt	D	10	JJ/MM/AAA >date début d'arrêt

Figure I.6 : Fiche d'analyse de la Certificat Médical

Chapitre 1 : Etude de l'existant

La figure suivante représente la fiche d'analyse du document N°4(Fiche d'analyse) :

Fiche d'analyse du document N°4			
I. Identification			
Code : FA Désignation : Fiche d'analyse. Nature de document : externe. Emetteur : médecin. Destinataire : patient. Nombre d'exemplaire : 01. Utilité : Effectuer des analyses..			
II. Description des rubriques			
Rubrique	Type	Taille	Observation
Nom médecin	A	10	/
Prénom médecin	A	10	/
N° tel médecin	A	10	/
Adresse	AN	20	/
Nom patient	A	10	/
Prénom patient	A	10	/
Age	N	2	/
Date	D	10	JJ/MM/AAAA
Nom d'analyse	A	30	/

Figure I.7 : Fiche d'analyse de la Fiche d'analyse

Chapitre 1 : Etude de l'existant

La figure suivante représente la fiche d'analyse du document N°5(Fiche Résultat d'analyse) :

Fiche d'analyse du document N°5			
I. Identification			
Code : FRA Désignation : Fiche résultat d'analyse. Nature de document : externe. Emetteur : patient. Destinataire : médecin. Nombre d'exemplaire : 01. Utilité : représentent les résultats d'analyses.			
II. Description des rubriques			
Rubrique	Type	Taille	Observation
Nom médecin	A	10	/
Prénom médecin	A	10	/
Adresse	AN	20	/
N° tel médecin	N	10	/
Nom patient	A	10	/
Prénom patient	A	10	/
Age	N	2	/
Date	D	10	JJ/MM/AAAA
Numéro patient	AN	20	/
N° tel patient	N	10	/
Date de prélèvement	D	10	JJ/MM/AAAA
Résultat d'analyse	AN	100	/

Figure I.8 : Fiche Résultat d'analyse

I.4. Graphe des flux d'information :

I.4.1. Définition :

Diagramme des flux est une représentation des interactions, entre les acteurs du champ d'étude (acteur interne) et les autres domaines (acteur externe) en termes de flux d'information échangé.[2]

I.4.2. Concepts utilisés :

a-Acteur : est un agent capable d'échanger une information avec les autres acteurs, il peut être interne ou externe

➤ **Formalisme graphique :**

Pour représenter un acteur on utilise un ovale ou un rectangle à l'intérieur du quelle on écrit le nom de l'acteur.

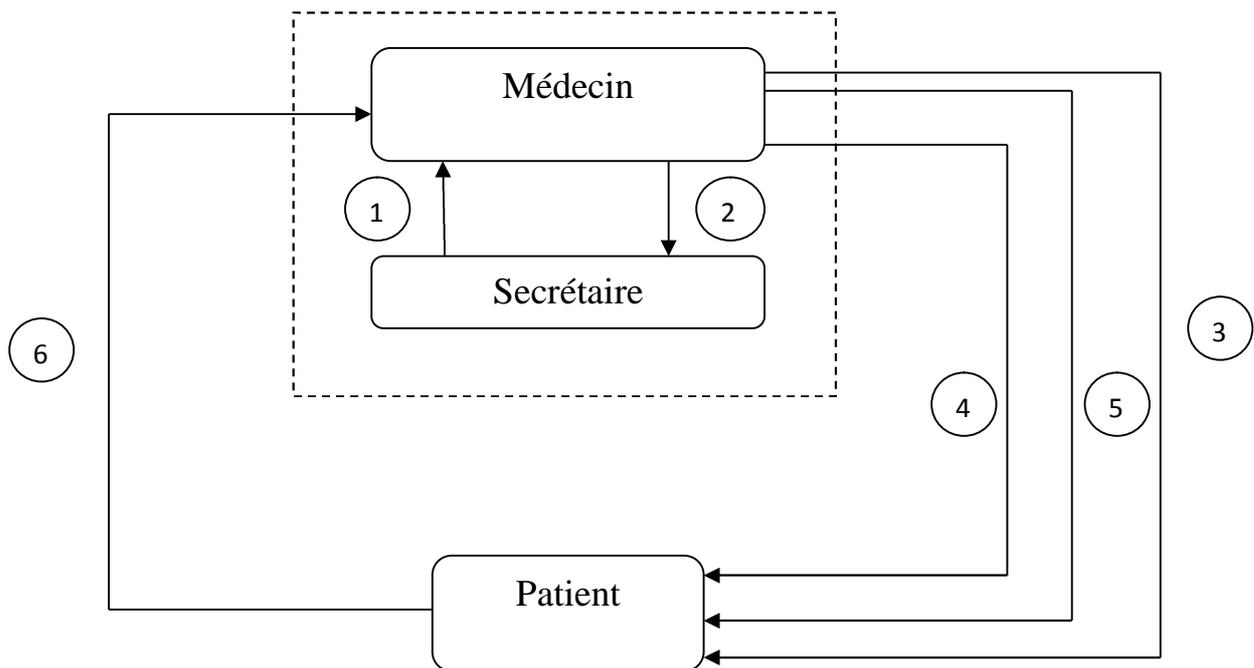
b-Flux d'information : il décrit l'information qui circule entre deux acteurs, dans le cadre du système d'information concerné.

➤ **Formalisme graphique :**

Le flux d'information est représenté par une flèche orienté de l'acteur émetteur vers l'acteur qui le reçoit. La désignation du flux est inscrite à côté de la flèche.

I.4.3. Graphe des flux d'information de l'organisme d'accueil :

La figure suivante illustre le graphe des flux d'information au sein de l'organisme d'accueil :



 Le champ d'étude

Figure I.9 : Le Graphe des flux de données

Description des flux d'information :

1. Dossier médical avant la mise à jour.
2. Dossier médical après la mise à jour.
3. Fiche d'analyse.
4. Ordonnance.
5. Certificat médical.
6. Fiche résultat d'analyse.

I.5. Critiques de l'existant :

Le cabinet médical ne dispose d'aucun outil informatique ce qui engendre beaucoup de problèmes dont :

- Travaux manuels élevés, lourds et pénibles qui se présente d'une façon répétitive à savoir l'archivage, la mise en œuvre et la consultation des fiches médicales.
- Absence d'un moyen de recherche rapide : pour chercher un dossier, il faut faire une recherche manuelle dossier par dossier par nom du patient, ce qui engendre une perte de temps.
- Processus très long avec probabilité de perte de documentation : puisqu'un dossier médical englobe un ensemble de documents tels -que, fiche médicale, ordonnance et les feuilles qui contiennent les dates des RDV. Il est possible qu'un document qui appartient à tel dossier soit rangé par erreur dans un autre dossier lors de l'organisation et le stockage dans les boites d'archives.
- Encombrement et non clarté de la fiche médicale qui contient plusieurs informations à cause de sa petite taille, 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 médical 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 le dossier.

- La gestion des RDV, se fait d'une manière manuelle ce qui provoque un risque d'oubli ou chevauchement des RDV.
- La perte de temps est remarquable en cas d'augmentation du nombre des patients pour la consultation.

I.6. Proposition de solution :

Dans le cadre de ce stage au sein du cabinet Lasers Dermatologiques. On a eu l'opportunité d'effectuer ce projet de fin d'étude qui consiste à développer une application pour la gestion d'un cabinet médical et en particulier un cabinet de dermatologie.

En effet, cette application permet de gérer les patients d'un cabinet, de rentrer les observations de la consultation en cours, de préparer les ordonnances, de les sauvegarder et de gagner le temps d'écriture en utilisant les listes prédéfinies de médicaments, de signes cliniques, d'antécédents familiaux et médicaux, etc.

I.7. Conclusion :

Tout au long de ce chapitre, on a pu donner une présentation du cadre de travail. Pour cela, on a commencé par la présentation de l'organisme d'accueil et ses différents postes. Puis, on a décrit l'application qu'on vise à développer.

Le prochain chapitre est consacré à l'analyse et la conception de notre système en utilisant UML. .

Chapitre II

Analyse et Conception

II.1. Introduction:

Pour mener à bien le développement d'une application, il convient de suivre une démarche méthodologique pour la planification et la conception de cette dernière, et cela en mettant en évidence tous les objectifs tracés pour la bonne élaboration du projet souhaité.

Dans ce cas précis, le principal objectif à atteindre est le développement d'une application permettant la gestion d'un cabinet médical. Pour cela l'utilisation d'UML semble être l'outil le plus approprié.

Ce chapitre est structuré comme suit : la première section est consacrée à la présentation de l'UML.

Dans les deux sections suivantes nous présentons respectivement l'analyse et la conception de notre application.

II.2. Les Méthodes de conception :

Les méthodes utilisées dans les années 80 pour organiser la programmation fonctionnelle (notamment MERISE) étaient fondées sur une modélisation séparée des données et des traitements.

Lorsque la programmation objet prend de l'importance au début des années 90, la nécessité d'une méthode qui lui soit adaptée devient évidente. Plus de cinquante méthodes apparaissent entre 1990 et 1995, mais uniquement trois d'entre elles se sont détachées nettement après quelque années : OMT, BOOCH et OOSE.

Chaque méthode avait ses avantages et ses partisans. Le nombre de méthode en compétition s'était réduit mais le risque d'un éclatement subsistait : la profession pouvait se diviser entre ses trois méthodes, créant autant de continents intellectuel qui auraient eu du mal à communiquer. C'est pour cela que l'unification de ses trois méthodes c'est avérée indispensable.

Pour la modélisation de notre application, nous allons utiliser le langage de modélisation unifié **UML**. [3]

II.2.1. Présentation d'UML :

II.2.1.1. Définition :

UML (Unified Modeling language), se définit comme un langage de modélisation graphique et textuel destiné à comprendre et à définir des besoins spécifiques et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. Il véhicule en particulier :

Les concepts des approches par objets : classe, instance, classification ...etc.

Intégrant d'autres aspects : association, fonctionnalités, événements, états, séquences, etc. [5]

II.2.1.2. Les diagrammes d'UML : [4]

UML définit treize types de diagrammes devisés en trois catégories :

1. Diagrammes fonctionnels :

- Diagramme de cas d'utilisation.
- Diagramme de contexte.

2. Diagrammes statiques (structurels) :

- Diagramme de classe.
- Diagramme d'objet.
- Diagramme de composant.
- Diagramme de déploiement.
- Diagrammes de structure composite.
- Diagrammes de package.

3. Diagrammes dynamiques (comportementaux) :

- Diagramme d'activité.

- Diagramme de séquence.
- Diagramme d'état transition.
- Diagramme de collaboration.
- Diagramme de timing.

II.2.2. Utilités d'UML :

UML est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet, UML offre un standard de modélisation, pour représenter l'architecture logicielle. Les différents éléments représentables sont :

- Activité d'un objet /logiciel.
- Acteurs.
- Processus.
- Schéma de base de données.
- Composants logiciel.
- Réutilisation de composants.

II.2.3. Formalisme d'UML :

UML se décompose en plusieurs sous ensembles :

- Les **vues** : les vues sont les observations du système. Elles décrivent le système d'un point de vue donné, qui peut être organisationnel, dynamique, temporel, architectural, géographique, logique, etc. En combinant toutes ses vues, il est possible de définir(ou retrouver) le système complet.
- Les **diagrammes** : Les diagrammes sont des éléments graphiques. Ceux-ci décrivent le contenu des vues, qui sont des notions abstraites. Les diagrammes peuvent faire partie de plusieurs vues.
- Les **modèles d'élément** : Les modèles d'élément sont les briques des diagrammes UML, ces modèles sont utilisés dans plusieurs types de diagrammes

Chapitre 2 : Analyse et conception

Exemple d'élément : cas d'utilisation, classes, association, etc.

II.3. Démarche de modélisation à suivre :

L'UML n'est pas une méthode et par conséquent il nous propose la démarche de modélisation représentée graphiquement comme suit:

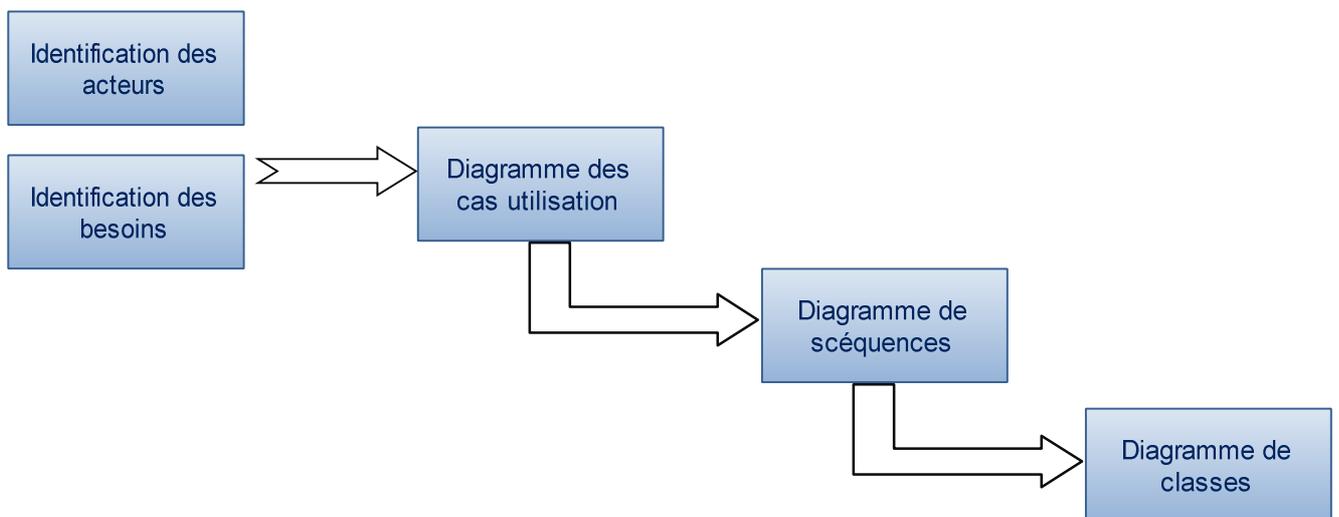


Figure II.1 : Représentation graphique de la Démarche de modélisation

II.4. L'Analyse :

Consiste à déterminer ce que l'application devra faire et l'expression de son comportement et son architecture ainsi que l'examen des cas d'utilisation et leurs scénarios.

La phase d'analyse débute par la mise en évidence des différents acteurs qui interviennent dans le système, et leurs besoins.

II.4.1. Identification des acteurs :

Un acteur représente une entité externe qui interagit directement avec le système étudié, en échangeant de l'information (en entrée et en sortie). On trouve les acteurs en observant les utilisateurs directs du système, les responsables de la maintenance, ainsi que les autres systèmes qui interagissent avec lui.

Pour notre système, on peut distinguer deux acteurs principaux:

- ✓ Secrétaire.
- ✓ Médecin
- ✓

II.4.2. Diagramme de contexte :

II.4.2.1. Définition :

C'est un modèle conceptuel de flux qui permet d'avoir une vision globale des interactions entre le système et les liens avec l'environnement extérieur .il permet aussi de bien délimiter le champ d'étude.[5]

II.4.2.2. Diagramme de contexte de notre application :

La figure suivante illustre le diagramme de contexte de notre application :

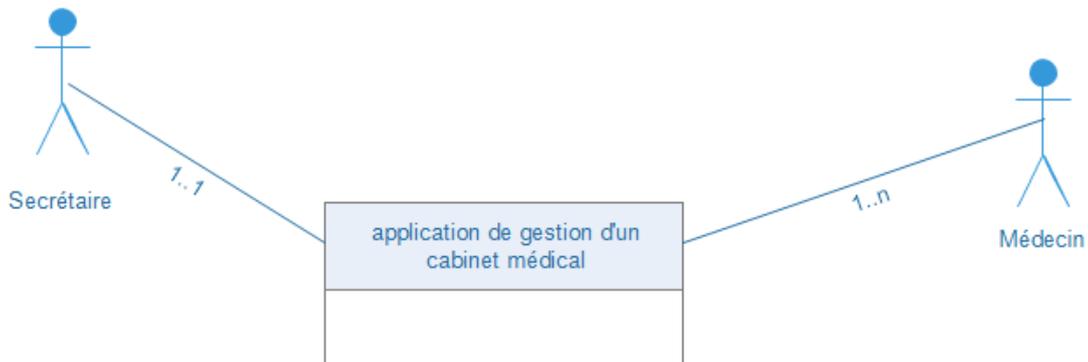


Figure II.2 : Diagramme de contexte

II.4.3. Elaboration de modèles de cas d'utilisation :

Les modèles des cas d'utilisation permettent d'avoir une représentation de l'ensemble des fonctionnalités complètes du système.

Le modèle de cas d'utilisation comprend les acteurs, le système et les cas d'utilisation eux-mêmes.

Cette section se décompose en deux parties :

- Le diagramme de cas d'utilisation.
- Description textuelle des cas d'utilisation.

II.4.3.1. Diagramme des cas d'utilisation :

II.4.3.1.1. Définition :

Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. C'est le premier diagramme du modèle UML. Celui où s'assure la relation entre l'utilisateur et les objets que le système met en œuvre.

Les éléments des diagrammes de cas d'utilisation :

Acteur

Un acteur est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système.

Cas d'utilisation

Un cas d'utilisation est un moyen de représenter les différentes possibilités d'utiliser un système. Il modélise un service rendu par le système, sans imposer le mode de réalisation de ce service.

Il exprime toujours une suite d'interaction entre un acteur et l'application.

Il définit une fonctionnalité utilisable par un acteur.

Relation entre cas d'utilisation :

- **L'inclusion :**

La relation d'inclusion "include" précise qu'un cas d'utilisation contient le comportement défini dans un autre cas d'utilisation.

- **L'extension :**

La relation d'extension "extends" signifie qu'un cas d'utilisation peut dans certains cas augmenter le comportement d'un autre cas d'utilisation. Une condition devra valider cette augmentation.

- **La relation «Use»:**Lorsqu'une ou plusieurs tâches sont utilisées régulièrement, on peut les factoriser dans un même use case et faire de telle sorte que d'autres use cases l'utilisent en pointant par une flèche traitée.

II.4.3.1.2. Identification des cas d'utilisation :

Pour chacun des acteurs cités, notre application doit donc offrir un ensemble de fonctionnalités. Ces fonctionnalités sont classées par acteur :

Médecin :

- ✓ S'authentifier
- ✓ Gestion des patients
 - Ajouter un patient
 - Consulter la liste des patients
 - Modification des coordonnées du patient
 - Suppression d'un patient
- ✓ Gestion du dossier médical
 - Visualiser le dossier médical
 - Consulter la liste des consultations médicales d'un patient
 - Ajout d'une nouvelle consultation
 - Visualisation des détails d'une consultation
 - Modification d'une consultation
 - Suppression d'une consultation
 - Création d'une ordonnance
 - Impression d'une ordonnance
 - Ajout d'un certificat médical
 - Impression d'un certificat médical
 - Ajout d'une lettre d'orientation
 - Impression de lettre d'orientation

Chapitre 2 : Analyse et conception

- ✓ Gestion de la bibliothèque médicale
 - Ajout d'un médicament
 - Modifier un médicament
 - Suppression d'un médicament

- ✓ Gestion des rendez-vous
 - Ajouter un rendez-vous
 - Modifier un rendez-vous
 - Supprimer un rendez-vous

- ✓ Gérer les coordonnées du médecin
 - Modifier les coordonnées

Secrétaire :

- ✓ S'authentifier
- ✓ Gestion des patients
- ✓ Gérer les rendez-vous

II.4.3.1.3. Diagramme de cas d'utilisation général :

La figure suivante représente le diagramme de cas d'utilisation général de notre application :

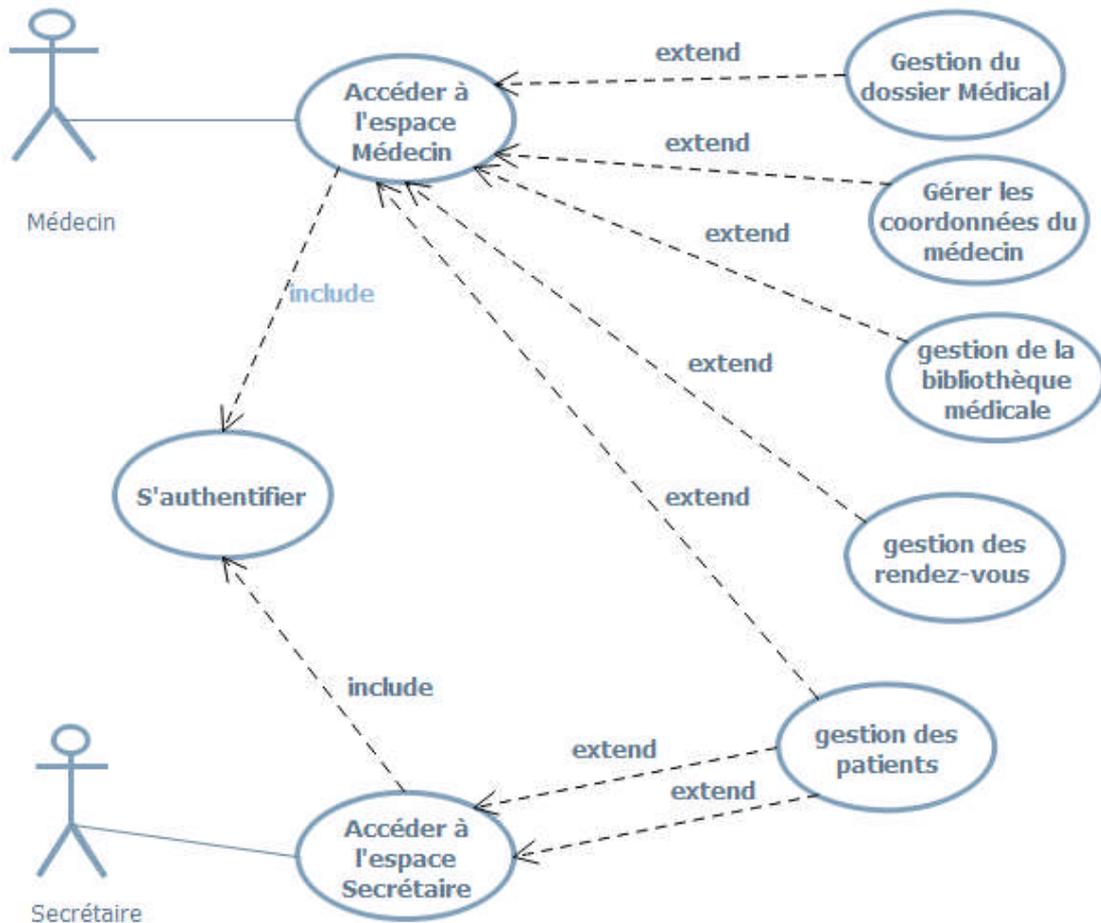


Figure II.3 : Diagramme de cas d'utilisation général

II.4.4. Description textuelle des cas d'utilisation :

Chaque résultat possible de l'exécution d'un cas d'utilisation est appelé scénario, c'est un chemin logique traversant le cas d'utilisation. Un scénario est une instance de cas d'utilisation. Un ensemble de scénarios pour un cas d'utilisation identifie tout ce qu'il peut arriver lorsque ce cas d'utilisation est en œuvre.

➤ Cas d'utilisation « s'authentifier »

Titre : s'authentifier

Résumé : chaque utilisateur doit taper son propre nom d'utilisateur et son mot de passe pour accéder à l'interface qui le concerne.

Acteurs : médecin et secrétaire.

Pré conditions : l'utilisateur n'est pas authentifié donc il ne peut pas accéder à l'application.

Post conditions : l'utilisateur est authentifié, il accède à l'interface qui le concerne.

Scénario nominal :

1. L'utilisateur demande le formulaire d'authentification
2. Le système affiche le formulaire
3. L'utilisateur tape son propre nom d'utilisateur et son mot de passe puis clique sur le Botton « connexion »
4. Le système vérifie la validité des coordonnées de l'utilisateur.
5. L'utilisateur accède à la session qui le concerne.

Scénario alternatif :

A1 : l'enchaînement démarre à l'étape 4 du scénario nominale. Le système indique la non validité des coordonnées de l'utilisateur, alors il ne peut pas accéder à sa propre session. L'enchaînement reprend à l'étape 3 du scénario nominal.

➤ Cas d'utilisation « Ajouter un nouveau patient »

Titre : Ajouter un nouveau patient

Acteurs : Utilisateur (médecin ou secrétaire)

Chapitre 2 : Analyse et conception

Pré condition : L'utilisateur est connecté, le patient n'existe pas

Post condition : patient créé

Scénario nominale :

1. L'utilisateur demande le formulaire d'ajout d'un nouveau patient
2. Le système affiche le formulaire
3. L'utilisateur remplit le formulaire
4. L'utilisateur valide le formulaire
5. Le système enregistre la demande d'ajout

Scénario alternatifs :

A1 : Les champs du formulaire ne sont pas remplis

L'enchaînement A1 démarre au point 4 du scénario nominal

5. Le système indique que le formulaire n'est pas bien rempli

Le scénario nominal reprend au point 3

A2 : Le patient existe déjà

L'enchaînement A2 démarre au point 4 du scénario nominal

5. Le système affiche un message d'erreur (patient existe déjà)

Le scénario nominal reprend au point 3

➤ **Cas d'utilisation « Consulter la liste des patients »**

Titre : Consulter la liste des patients

Acteurs : Utilisateur (médecin ou secrétaire)

Pré condition : L'utilisateur est connecté,

Scénario nominale

1. L'utilisateur accède à son espace après l'authentification
2. L'utilisateur clique sur « gestion des patient » puis sur liste des patients
3. Le système affiche la liste des patients.

➤ **Cas d'utilisation « modification des coordonnées d'un patient »**

Titre : modification des coordonnées d'un patient

Acteurs : Utilisateur (médecin ou secrétaire)

Pré condition : L'utilisateur est connecté, le patient existe

Post condition : coordonnées patient modifiées

Scénario nominale :

1. L'utilisateur demande l'affichage de la liste des patients
2. Le système affiche la liste
3. L'utilisateur sélectionne un patient
4. L'utilisateur Clique sur le bouton de modification
5. Le système affiche le formulaire de modification
6. L'utilisateur effectue les modifications puis valide
7. Le système enregistre les modifications puis affiche un message de confirmation

➤ **Cas d'utilisation « Suppression d'un patient »**

Titre : Suppression d'un patient

Acteurs : Utilisateur (médecin ou secrétaire)

Pré condition : L'utilisateur est connecté, le patient existe

Post condition : patient supprimé

Scénario nominale :

1. L'utilisateur demande l'affichage de la liste des patients
2. Le système affiche la liste
3. L'utilisateur saisit le nom du patient à supprimer et il valide la suppression en cliquant sur « Supprimer ».
4. Le système enregistre la demande de suppression et affiche un message de confirmation de suppression.

➤ **Cas d'utilisation « Visualiser le dossier médical »**

Titre : Visualiser le dossier médical

Acteurs : médecin

Pré condition : le médecin est connecté, le patient existe, le dossier médical existe

Scénario nominale :

1. Le médecin clique sur le menu Dossier Médical dans la page d'accueil
2. Le système affiche la liste des patients
3. Le médecin sélectionne un patient puis clique sur « voir dossier »

Chapitre 2 : Analyse et conception

4. Le système affiche les détails du dossier médical du patient

Scénario alternatifs :

A1 : Le dossier médical est vide

L'enchaînement A1 démarre au point 4 du scénario nominal

4. Le système affiche un dossier médical vide

➤ **Cas d'utilisation « Consulter la liste des consultations d'un patient »**

Titre : Consulter la liste des consultations d'un patient

Acteurs : médecin

Pré condition : le médecin est connecté, le patient existe

Scénario nominale :

1. L'utilisateur demande l'affichage de la liste des patients
2. Le système affiche la liste
3. L'utilisateur sélectionne un patient puis clique sur « voir Dossier »
4. Le système affiche les détails du dossier médical du patient
5. Le médecin clique sur « Consultation »
6. Le système affiche la liste des consultations du patient sélectionné

➤ **Cas d'utilisation « Ajout d'une nouvelle consultation »**

Titre : Ajout d'une nouvelle consultation

Acteurs : médecin

Pré condition : le médecin est connecté, le patient existe

Post condition : consultation ajoutée

Scénario nominale :

1. L'utilisateur demande l'affichage de la liste des patients
2. Le système affiche la liste
3. L'utilisateur sélectionne un patient puis clique sur « voir Dossier »
4. Le système affiche les détails du dossier médical du patient
5. Le médecin clique sur « nouvelle consultation »
6. Le système affiche un formulaire d'ajoute d'une consultation
7. Le médecin remplit le formulaire puis valide
8. Le système affiche un message pour confirmer l'enregistrement de la consultation

➤ **Cas d'utilisation « Visualisation des détails d'une consultation »**

Titre : Visualisation des détails d'une consultation

Acteurs : médecin

Pré condition : le médecin est connecté, le patient existe, la consultation est déjà créée

Scénario nominale :

1. L'utilisateur demande l'affichage de la liste des patients
2. Le système affiche la liste
3. L'utilisateur sélectionne un patient puis clique sur « voir Dossier »
4. Le système affiche les détails du dossier médical du patient

5. Le médecin clique sur le menu « consultation »
6. Le système affiche la liste des consultations du patient sélectionné
7. Le médecin sélectionne une consultation
8. Le système affiche les détails de la consultation

➤ **Cas d'utilisation « Suppression d'une consultation »**

Titre : Suppression d'une consultation

Acteurs : médecin

Pré condition : le médecin est connecté, la consultation existe

Post condition : consultation supprimé

Scénario nominale :

1. Le médecin demande l'affichage de la liste des consultations d'un patient
2. Le système affiche la liste des consultations
3. Le médecin sélectionne la consultation puis clique sur « supprimer »
4. Le système enregistre la demande de suppression et affiche un message de confirmation de suppression.

➤ **Cas d'utilisation « Création d'une ordonnance »**

Titre : Création d'une ordonnance

Acteurs : médecin

Pré condition : le médecin est connecté, le patient existe, consultation créé

Post condition : Ordonnance ajoutée

Scénario nominale :

1. Le médecin demande l'affichage de la liste des patients
2. Le système affiche la liste
3. Le médecin sélectionne un patient puis clique sur « voir Dossier »
4. Le système affiche les détails du dossier médical du patient
5. Le médecin clique sur «nouvelle Consultation»
6. Le système affiche un formulaire d'ajout d'une nouvelle consultation
7. Le médecin remplit le formulaire puis clique sur « Délivrer Ordonnance »
8. Le système affiche un formulaire d'ajout des médicaments à l'ordonnance
9. Le médecin remplit l'ordonnance puis valide
10. Le système effectue l'enregistrement

➤ **Cas d'utilisation « Impression d'une ordonnance »**

Titre : Impression d'une ordonnance

Acteurs : médecin

Pré condition : le médecin est connecté, l'ordonnance existe

Post condition : Ordonnance imprimé

Scénario nominale :

1. Le médecin demande l'affichage de la liste des patients
2. Le système affiche la liste
3. Le médecin sélectionne un patient puis clique sur «voir Dossier»
4. Le système affiche les détails du dossier médical du patient

5. Le médecin clique sur le menu « ordonnance »
6. Le système affiche la liste des ordonnances du patient
7. Le médecin sélectionne une ordonnance puis clique sur « impression »

➤ **Cas d'utilisation « Ajout d'un certificat médical »**

Titre : Ajout d'un certificat médical

Acteurs : médecin

Pré condition : le médecin est connecté, le patient existe, le dossier médical est créé

Post condition : Certificat médical ajoutée

Scénario nominale :

1. Le médecin demande l'affichage de la liste des patients
2. Le système affiche la liste
3. Le médecin sélectionne un patient puis clique sur « Voir Dossier »
4. Le système affiche les détails du dossier médical du patient
5. Le médecin clique sur «Nouvelle Consultation»
6. Le système affiche un formulaire d'ajout d'une nouvelle consultation
7. Le médecin remplit le formulaire puis clique sur « Certificat Médical »
8. Le système affiche un formulaire d'ajout d'un certificat médical
9. Le médecin remplit le certificat médical puis clique sur « enregistrer »
10. Le système affiche un message de confirmation d'ajout

➤ Cas d'utilisation « Impression d'un certificat médical »

Titre : Impression d'un certificat médical

Acteurs : médecin

Pré condition : le médecin est connecté, le patient existe, certificat médical existe

Post condition : Certificat médical imprimé

Scénario nominale :

1. Le médecin demande l'affichage de la liste des patients
2. Le système affiche la liste
3. Le médecin sélectionne un patient puis clique sur « Voir Dossier »
4. Le système affiche les détails du dossier médical du patient
5. Le médecin clique sur « Certificat Médical »
6. Le système affiche la liste des certificats du patient
7. Le médecin sélectionne le certificat puis clique sur « impression »

➤ Cas d'utilisation « Ajout d'une lettre d'orientation »

Titre : Ajout d'une lettre d'orientation

Acteurs : médecin

Pré condition : le médecin est connecté, le patient existe, le dossier médical est créé

Post condition : lettre d'orientation ajoutée

Scénario nominale :

1. Le médecin demande l'affichage de la liste des patients
2. Le système affiche la liste
3. Le médecin sélectionne un patient puis clique sur «Voir Dossier»
4. Le système affiche les détails du dossier médical du patient
5. Le médecin clique sur «Nouvelle Consultation»
6. Le système affiche un formulaire d'ajout d'une nouvelle consultation
7. Le médecin remplit le formulaire puis clique sur « Lettre d'orientation»
8. Le système affiche un formulaire d'ajout d'une lettre d'orientation
9. Le médecin remplit la lettre puis clique sur « enregistrer »
10. Le système affiche un message de confirmation d'ajout

➤ **Cas d'utilisation « Impression de lettre d'orientation »**

Titre : Impression d'une lettre d'orientation

Acteurs : médecin

Pré condition : le médecin est connecté, le patient existe, la lettre d'orientation existe

Post condition : lettre d'orientation imprimée

Scénario nominale :

1. Le médecin demande l'affichage de la liste des patients
2. Le système affiche la liste des patients
3. Le médecin sélectionne un patient puis clique sur « Voir Dossier »

Chapitre 2 : Analyse et conception

4. Le système affiche les détails du dossier médical du patient
5. Le médecin clique sur le menu «lettre d'orientation »
6. Le système affiche la liste des lettres du patient
7. Le médecin sélectionne une lettre d'orientation puis clique sur « impression »

➤ **Cas d'utilisation « Ajouter un médicament »**

Titre : Ajouter un médicament

Acteurs : médecin

Pré condition : le médecin est connecté, le médicament n'est pas enregistré dans la base des médicaments

Post condition : médicament ajouté

Scénario nominale :

1. Le médecin demande le formulaire d'ajout d'un nouveau médicament
2. Le système affiche le formulaire
3. Le médecin remplit le formulaire
4. Le médecin valide le formulaire
5. Le système enregistre la demande d'ajout et affiche un message de confirmation d'ajout du médicament.

Scénario alternatifs :

A1 : Les champs du formulaire ne sont pas remplis

L'enchaînement A1 démarre au point 4 du scénario nominal

5. Le système indique que le formulaire n'est pas bien rempli

Le scénario nominal reprend au point 3

A2 : Le médicament existe déjà

L'enchaînement A2 démarre au point 4 du scénario nominal

5. Le système affiche un message d'erreur (médicament existe déjà)

Le scénario nominal reprend au point 3

➤ **Cas d'utilisation « Modifier un médicament »**

Titre : Modifier un médicament

Acteurs : médecin

Pré condition : Le médecin est connecté, le médicament existe

Post condition : médicament modifié

Scénario nominale :

1. Le médecin demande l'affichage de la liste des médicaments
2. Le système affiche la liste des médicaments
3. Le médecin sélectionne un médicament
4. Le médecin Clique sur le bouton de modification
5. Le système affiche le formulaire de modification
6. Le médecin effectue les modifications puis valide
7. Le système enregistre les modifications puis affiche un message de confirmation de modification

➤ Cas d'utilisation « Suppression d'un médicament »

Titre : Suppression d'un médicament

Acteurs : médecin

Pré condition : Le médecin est connecté, le médicament existe

Post condition : médicament supprimé

Scénario nominale :

1. Le médecin demande l'affichage de la liste des médicaments
2. Le système affiche la liste des médicaments
3. Le médecin saisit le nom du médicament à supprimer et il valide la suppression en cliquant sur « Supprimer ».
4. Le système enregistre la demande de suppression et affiche un message de confirmation de suppression.

➤ Cas d'utilisation « Ajouter un rendez-vous »

Titre : Ajouter un rendez-vous

Acteurs : L'utilisateur (Secrétaire)

Pré condition : L'utilisateur est connecté, le patient existe

Post condition : rendez-vous ajoutée

Scénario nominale :

1. L'utilisateur demande l'affichage de la liste des patients
2. Le système affiche la liste
3. Le médecin sélectionne un patient puis clique sur « Rendez-vous »

Chapitre 2 : Analyse et conception

4. Le système affiche un formulaire d'ajout d'un nouveau rendez-vous
5. L'utilisateur remplit le formulaire puis valide
6. Le système affiche un message de confirmation d'ajout du rendez-vous.

➤ **Cas d'utilisation « Ajouter un rendez-vous »**

Titre : Ajouter un rendez-vous

Acteurs : L'utilisateur (Médecin)

Pré condition : L'utilisateur est connecté, le patient existe

Post condition : rendez-vous ajoutée

Scénario nominale :

1. L'utilisateur demande l'affiche de la liste des patients
2. Le système affiche la liste des patients
3. Le médecin sélectionne un patient puis clique sur « Voir Dossier»
4. Le système affiche les détails du dossier médical du patient
5. L'utilisateur Clique sur le bouton « nouveau RDV »
4. Le système affiche un formulaire d'ajout d'un nouveau rendez-vous
5. L'utilisateur remplit le formulaire puis valide
6. Le système affiche un message de confirmation d'ajout du rendez-vous.

➤ Cas d'utilisation « Modifier un rendez-vous »

Titre : Modifier un rendez-vous

Acteurs : Utilisateur (médecin ou secrétaire)

Pré condition : L'utilisateur est connecté, le patient existe, le rendez-vous existe

Post condition : rendez-vous modifié

Scénario nominale :

1. L'utilisateur demande l'affichage de la liste des rendez-vous
2. Le système affiche la liste des rendez-vous
3. L'utilisateur sélectionne un rendez-vous puis clique sur « modifier»
4. Le système affiche un formulaire de modification du rendez-vous
5. L'utilisateur effectue les modifications puis valide
6. Le système affiche un message de confirmation de modification

➤ Cas d'utilisation « Supprimer un rendez-vous »

Titre : supprimer un rendez-vous

Acteurs : Utilisateur (médecin ou secrétaire)

Pré condition : L'utilisateur est connecté, le patient existe, le rendez-vous existe

Post condition : rendez-vous supprimé

Scénario nominale :

1. L'utilisateur demande l'affichage de la liste des rendez-vous
2. Le système affiche la liste des rendez -vous
3. L'utilisateur sélectionne un rendez-vous puis clique sur « supprimer»

4. Le système effectue la suppression puis affiche un message de confirmation de modification

➤ **Cas d'utilisation « Modifier mot de passe du médecin »**

Titre : Modifier mot de passe du médecin

Acteurs : médecin

Pré condition : Le médecin est connecté

Post condition : mot de passe modifié

Scénario nominale :

1. Le médecin clique sur le menu le bouton « modifier mot de passe » dans le menu « Gestion des coordonnées » de la page d'accueil
2. Le système affiche le formulaire de modification du mot de passe
3. Le médecin effectue les modifications puis valide
4. Le système enregistre les modifications puis affiche un message de confirmation

➤ **Cas d'utilisation « Modifier nom d'utilisateur du médecin »**

Titre : Modifier nom d'utilisateur du médecin

Acteurs : médecin

Pré condition : Le médecin est connecté

Post condition : nom d'utilisateur modifié

Scénario nominale :

Chapitre 2 : Analyse et conception

1. Le médecin clique sur le menu le bouton « modifier nom d'utilisateur » dans le menu « Gestion des coordonnées » de la page d'accueil
2. Le système affiche le formulaire de modification du nom d'utilisateur
3. Le médecin effectue les modifications puis valide
4. Le système enregistre les modifications puis affiche un message de confirmation

II.5. La conception :

La conception est la phase la plus complexe du projet, Dans laquelle une nouvelle vue du modèle fait son apparition. Cette vue exprime les modules et les exécutable physiques sans aller à la réalisation concrète du système.

Après avoir décrit textuellement les différents cas d'utilisations concernant notre système, nous allons les présenter formellement à l'aide des diagrammes

II.5.1. Diagramme de séquence :

II.5.1.1. Définition :

Le diagramme de séquence représente la succession chronologique des opérations réalisées par un acteur. Il indique les objets que l'acteur va manipuler et les opérations qui font passer d'un objet à un autre.

II.5.1.2. Diagramme de Séquence de quelques cas d'utilisation :

Voici Les diagrammes de séquence de quelques cas d'utilisation :

➤ Diagramme de séquence pour le cas d'utilisation « Authentification »

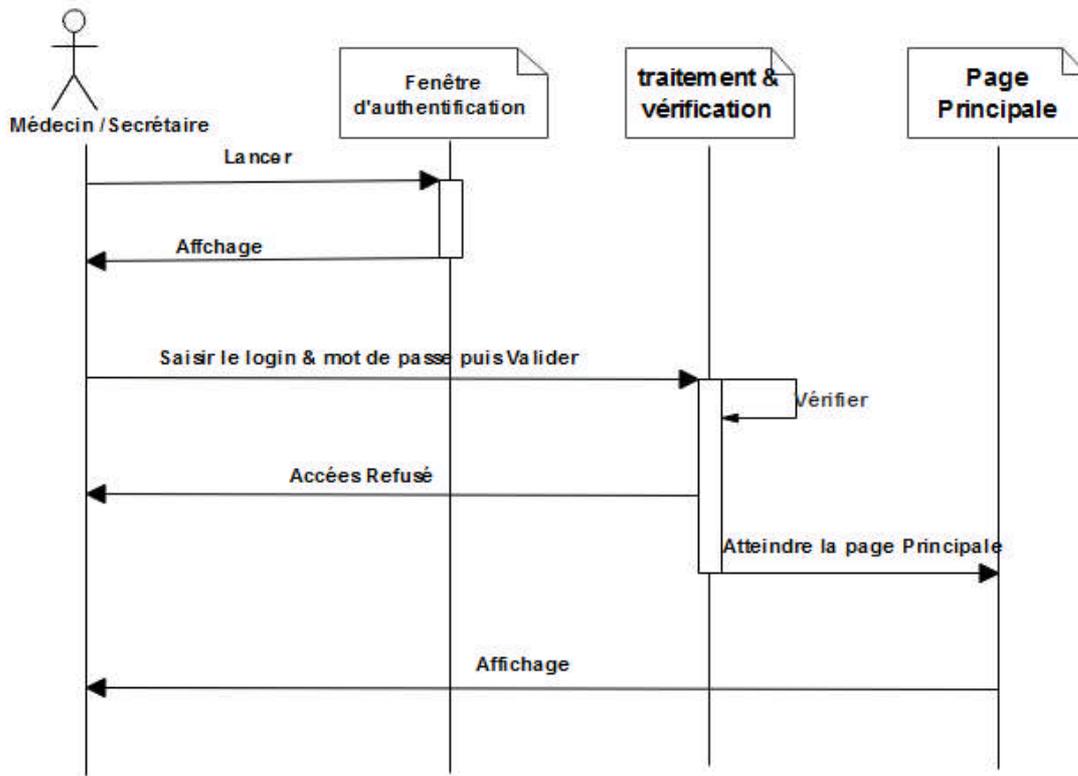


Figure II.4 : Diagramme de séquence « S'authentifier »

Chapitre 2 : Analyse et conception

➤ Diagramme de séquence pour le cas d'utilisation « Nouvelle consultation »

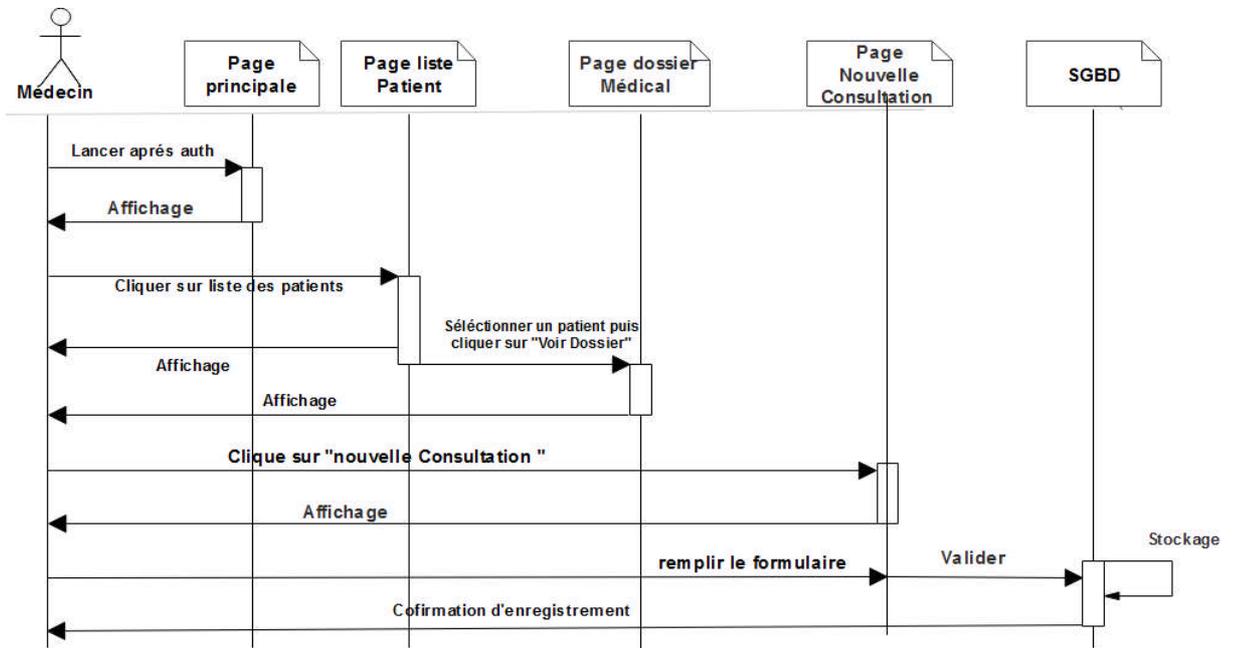


Figure II.5 : Diagramme de séquence « Nouvelle Consultation »

➤ Diagramme de séquence pour le cas d'utilisation « Ajouter un Patient »

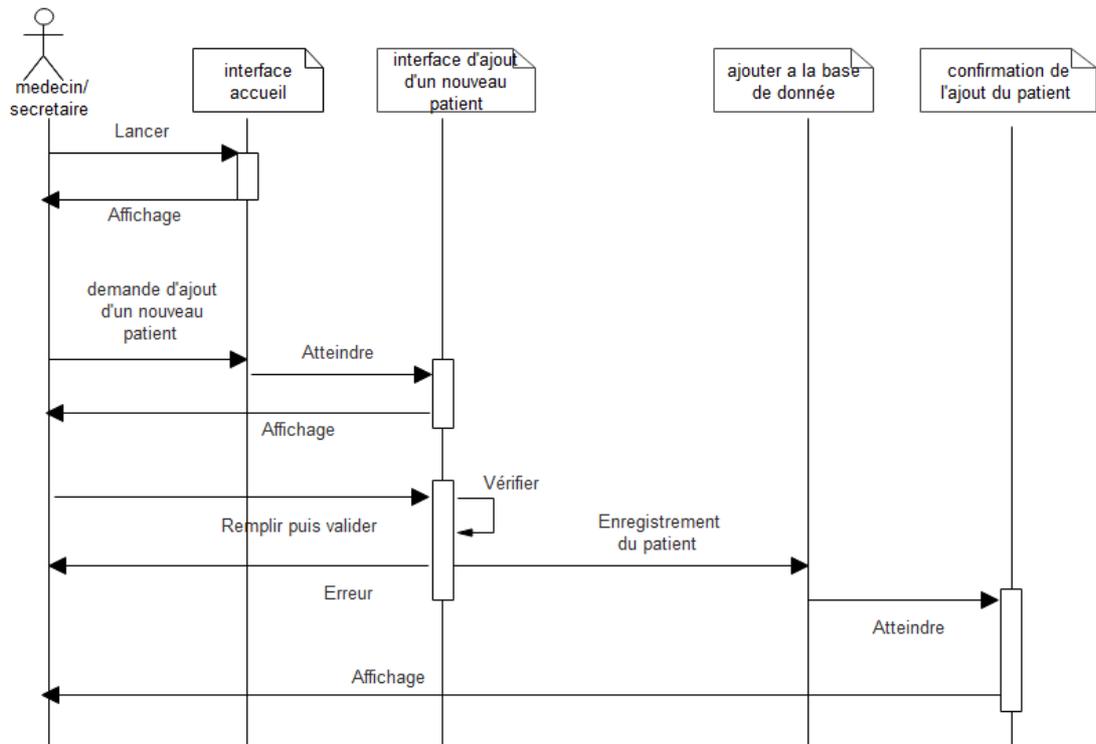


Figure II.6 : Diagramme de séquence « Ajouter un Patient »

➤ Diagramme de séquence pour le cas d'utilisation « Modifier Patient »

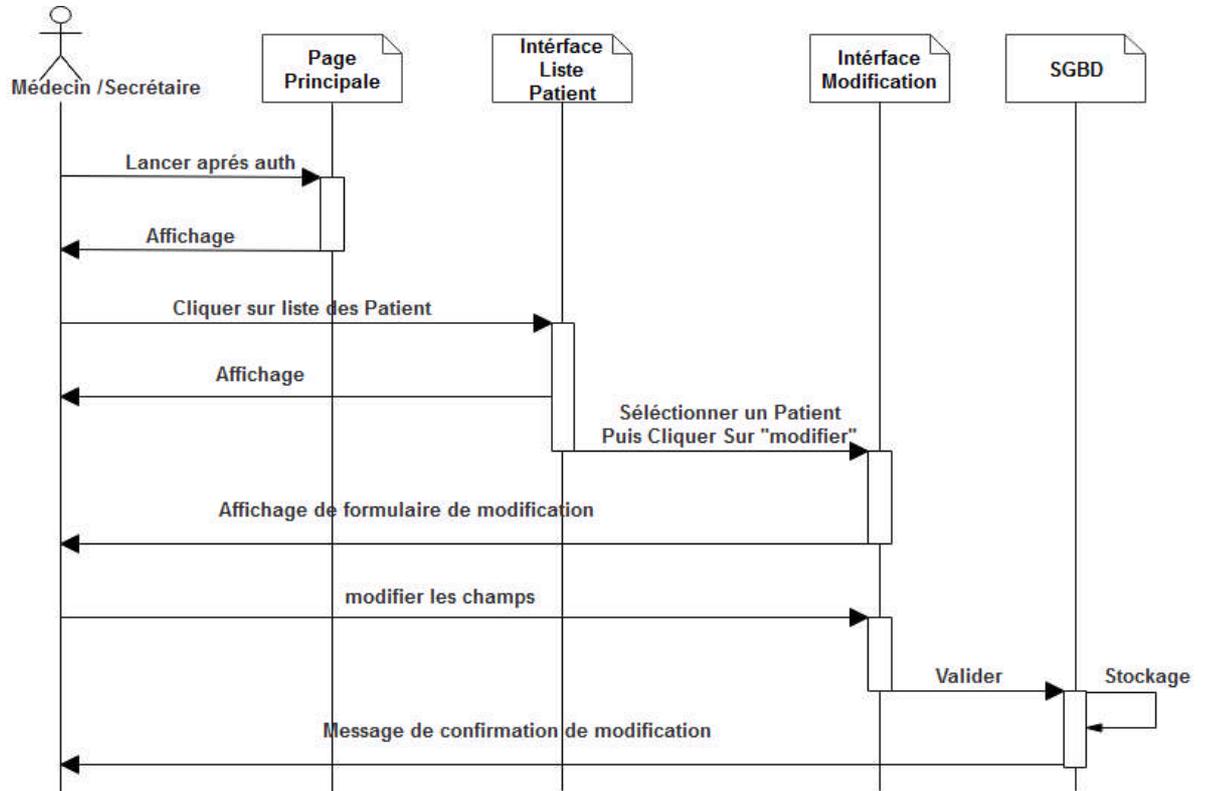


Figure II.7 : Diagramme de séquence « Modifier Patient »

➤ Diagramme de séquence pour le cas d'utilisation « Ajouter Rendez-vous »

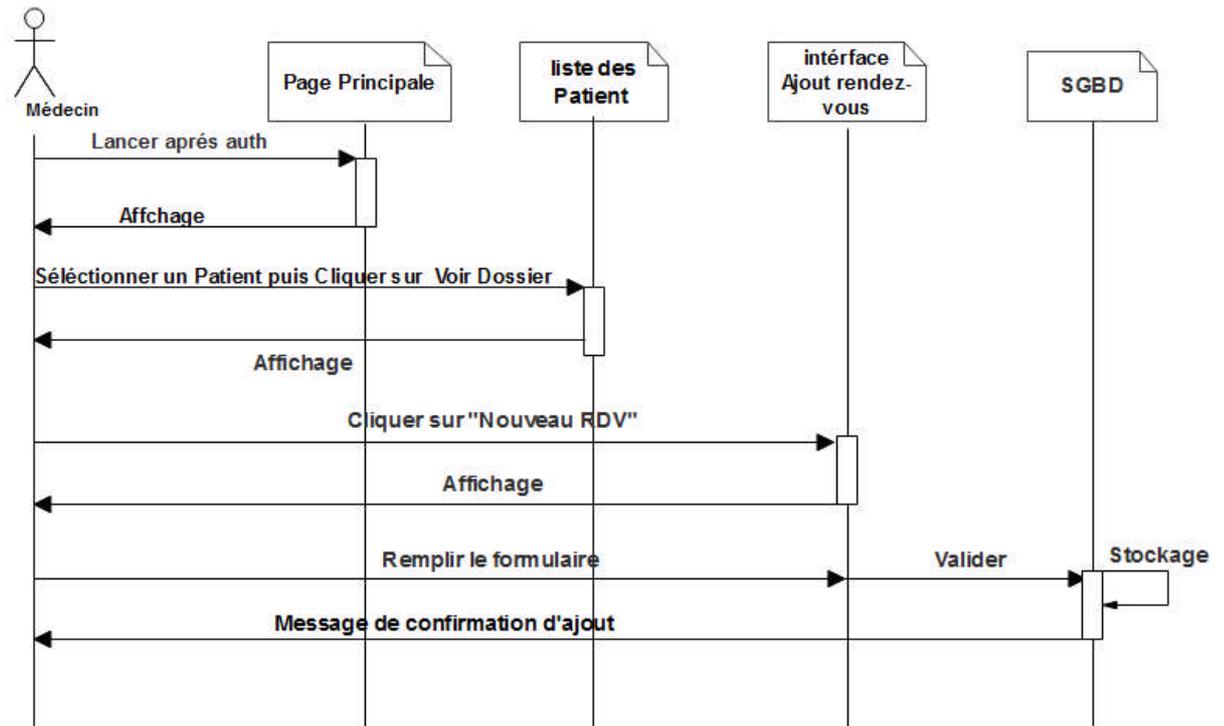


Figure II.8 : Diagramme de séquence « Ajouter Rendez-vous »

II.5.2. Diagramme de classe :

Le diagramme de classes est généralement considéré comme le plus important dans un développement orienté objet. Il représente l'architecture conceptuelle du système. Il décrit les classes que le système utilise, ainsi que leurs liens.

La figure suivante représente le diagramme de class détaillé de notre application :

➤ **Diagramme de classe détaillé :**

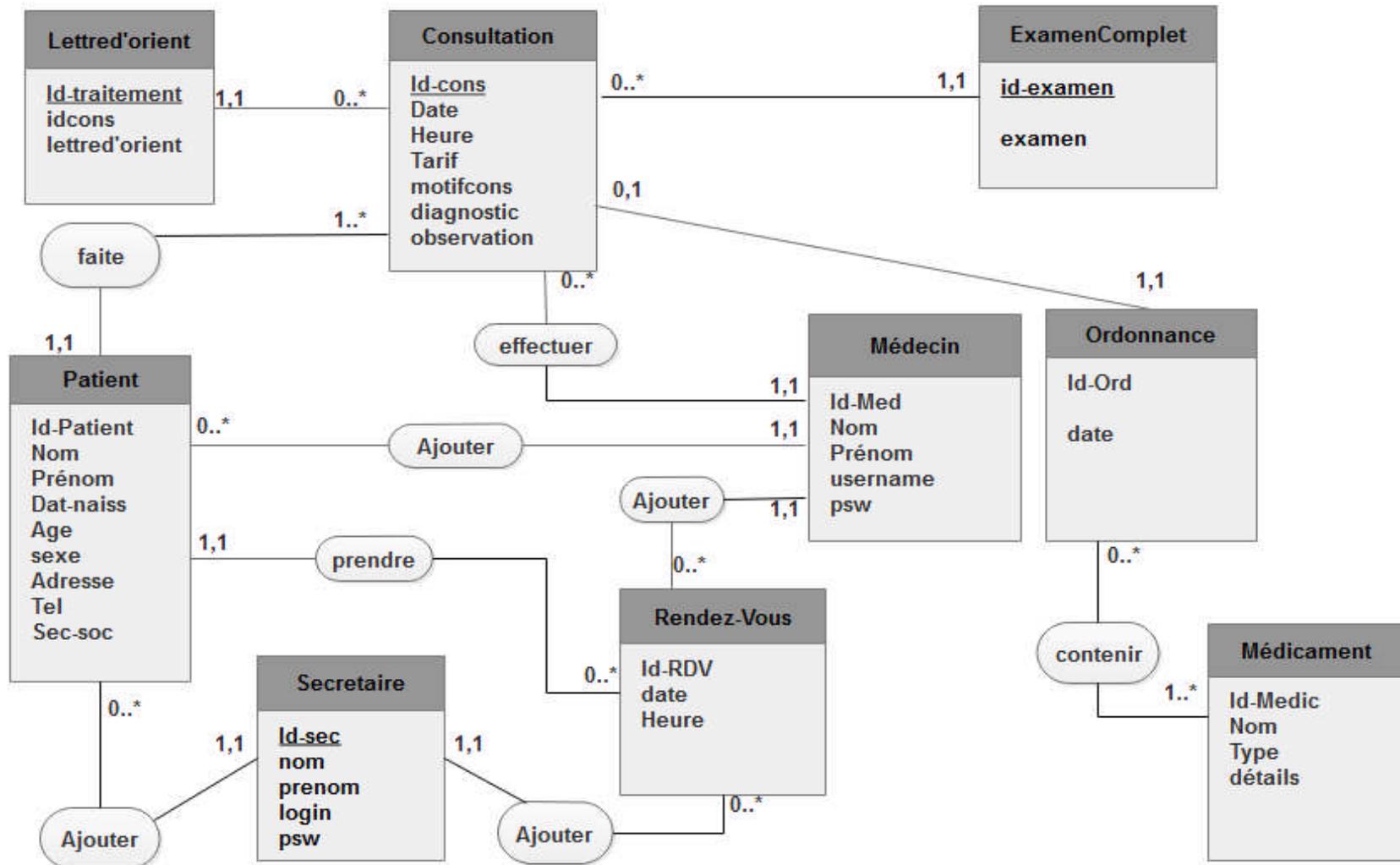


Figure II.9 : Diagramme de classe détaillé

La base de données :

Après avoir modélisé notre application avec les différents diagrammes offerts par le langage de modélisation UML, une mise en œuvre d'une base de données est nécessaire car elle permet d'étendre les possibilités d'interaction avec les utilisateurs et de mettre des données à la disposition d'utilisateur pour une consultation, une saisie ou une mise à jour tout en s'assurant des droits accordés à ces derniers.

Le Modèle relationnel :

Un schéma relationnel est un ensemble de relations correspondant à la structure de données manipulées par une application. On peut déduire notre modèle relationnel à partir du diagramme de classes, en fonction des cardinalités.

Schema Relationnel :

table _medecin (idmed, nommed , prenommed , psw , username)

table_Secretaire (idsec , nomsec ,prenomsec , login , psw)

table _patient (idpatient , nom , prenom , datenaiss , sexe , adresse , tel , secsoc , age , idmed*)

table_consultation (idcons , date , tarif , heure , motifcons , diagnostic , observation , idpatient* , idmed*)

table_ordonnance (idord , dateord , idcons*)

table_medicament (idmedicament , nommedic , typemedic , detailmedic)

table-contenir (traitement , durée , quantite , idmedic* , idord*)

table_rendzvous (idrdv , heurerdv , daterdv , idpatient* , idmed*)

table_lettre_orient (idorient , lettre , idcons*)

table_certificat (idcert , certificat , idcons*)

examen complet (idexamen, examen , idcons*)

II.6.Conclusion :

Dans ce chapitre nous avons présenté l'architecture conceptuelle de notre application en utilisant le proche objet, et cela en s'aidant des différents outils de modélisation du langage UML qui facilite la maîtrise et la production de logiciels.

Pour la phase analyse, nous avons défini les différents cas d'utilisation puis, nous les avons traduit au travers la construction des diagrammes de séquence, puis on a construit le diagramme de classes dans la phase de conception.

Enfin nous avons défini le modèle relationnel.

Le chapitre suivant sera consacré à l'implémentation et la réalisation de notre application.

Chapitre III

Implémentation

et

Réalisation

III.1. Introduction :

La réalisation vient couronner le travail de l'étude conceptuelle. Elle présente la dernière étape et elle est très importante puisque grâce à elle le projet informatique va exister réellement, sa réussite est conditionnée par une multitude de choix essentiellement d'ordre technique concernant l'exécution de ce qui a été conçu et proposé comme solution afin de répondre aux besoins des utilisateurs et de remédier aux insuffisances perçues.

Nous allons présenter dans ce dernier chapitre les tables de la base de données, l'environnement de développement, les outils qui ont servi à la réalisation de notre application, et nous terminerons par la présentation de ses fonctionnalités à travers ses différentes interfaces.

III.2. Le modèle physique des données :

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

Table medecin :

Attribut	Description	Type	Clé
idmed	Numéro d'identification du médecin	Int(2)	Primaire
nommed	Nom du médecin	Varchar(20)	
prenommed	Prénom du médecin	Varchar(20)	
psw	Mot de passe du médecin	Varchar(10)	
username	Nom d'utilisateur du médecin	Varchar(10)	

Figure III.1 : Table medecin

Chapitre 3: Implémentation et réalisation

Table Secretaire :

Attribut	Description	Type	Clé
idsec	Numéro d'identification de la secrétaire	Int(2)	Primaire
nomsec	Nom de la secrétaire	Varchar(20)	
prenomsec	Prénom de la secrétaire	Varchar(20)	
login	Login de la secrétaire	Varchar(10)	

Figure III.2 : Table Secretaire

Table patient :

Attribut	Description	Type	Clé
idpatient	Numéro d'identification du patient	Int(11)	Primaire
nom	Nom du patient	Varchar(20)	
prenom	Prénom du patient	Varchar(20)	
idmed	Numéro d'identification du médecin	Int(2)	Secondaire
datenaiss	Date de naissance du patient	date	
sexe	Sexe du patient	Varchar(5)	
adresse	Adresse du patient	Varchar(30)	
tel	Numéro de téléphone du patient	Int(10)	
secsoc	Numéro de sécurité sociale du patient	Int(12)	
age	Age du patient	Int(2)	

Figure III.3 : Table patient

Chapitre 3: Implémentation et réalisation

Table consultation :

Attribut	Description	Type	Clé
idcons	Numéro d'identification de la consultation	Int(11)	Primaire
idmed	Numéro d'identification du médecin	Int(2)	Secondaire
idpatient	Numéro d'identification du patient	Int(11)	Secondaire
date	Date de la consultation	date	
tarif	Tarif de la consultation	Int(4)	
heure	Heure de la consultation	time	
motifcons	Motif de la consultation	text	
diagnostic	Diagnostic de la consultation	text	
observation	Observation	text	

Figure III.4 : Table consultation

Table ordonnance :

Attribut	Description	Type	Clé
idord	Numéro d'identification de l'ordonnance	Int(11)	Primaire
idcons	Numéro d'identification de la consultation	Int(11)	Secondaire
dateord	Date de l'ordonnance	date	

Figure III.5 : Table ordonnance

Chapitre 3: Implémentation et réalisation

Table médicament :

Attribut	Description	Type	Clé
idmedicament	Numéro d'identification de médicament	Int(11)	Primaire
nommedic	Nom du médicament	Varchar(20)	
typemedic	Type du médicament	Varchar(10)	
detailmedic	Détails sur le médicament	text	

Figure III.6 : Table médicament

Table contenir :

Attribut	Description	Type	Clé
idmedic	Numéro d'identification de médicament	Int(11)	Secondaire
idord	Numéro d'identification de l'ordonnance	Int(11)	Secondaire
traitement	Traitement	text	
durée	Durée	Int(3)	
quantite	Quantité du médicament	Int(3)	

Figure III.7 : Table contenir

Table rendezvous :

Attribut	Description	Type	Clé
idrdrv	Numéro d'identification du rendez-vous	Int(11)	Primaire
idpatient	Numéro d'identification du patient	Int(11)	Secondaire
daterdrv	Date du rendez-vous	date	
heurerdrv	Heure du rendez-vous	time	
idmed	Numéro d'identification du médecin	Int(2)	Secondaire

Figure III.8 : Table rendezvous

Chapitre 3: Implémentation et réalisation

Table lettreorientation :

Attribut	Description	Type	Clé
idorient	Numéro d'identification de la lettre d'orientation	Int(11)	Primaire
idcons	Numéro d'identification de la consultation	Int(11)	Secndaire
lettre	Lettre d'orientation	text	

Figure III.9 : Table lettreorientation

Table CertificatMédicale :

Attribut	Description	Type	Clé
idcert	Numéro d'identification du certificat Médicale	Int(11)	Primaire
idcons	Numéro d'identification de la consultation	Int(11)	Secndaire
certificat	Certificat médicale	text	

Figure III.10 : Table CertificatMédicale

Table Examencomplet :

Attribut	Description	Type	Clé
Idexamen	Numéro d'identification de l'examen complet	Int(11)	Primaire
idcons	Numéro d'identification de la consultation	Int(11)	Secndaire
examen	Examen Complet	text	

Figure III.11 : Table Examencomplet

III.3. Description de l'environnement de travail :

Afin de réaliser notre projet on a travaillé sous le système d'exploitation Microsoft Windows 8 et on a opté pour la programmation en java sous l'environnement de développement : NetBeans IDE 8.2

III.3.1. Les outils de développement :

III.3.1.1. Le langage de programmation (java) :

JAVA est un langage orienté objet créé par James Gosling et Patrick Naughton employés de Sun Microsystems avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au Sun World.

Les concepts de JAVA sont hérités du C++ dont il est issu. JAVA possède des bibliothèques de classes qui fournissent les types de données de base, les possibilités d'entrées/sorties du système, la gestion des réseaux et autres fonctions utilitaires. Ces bibliothèques sont écrites en java et de plus JAVA est doté de mécanismes puissants tels que :

- **Le GarbageCollector (GC)** asynchrone (ramasse miettes), le GC libère toutes les instances en mémoire qui ne sont plus référencées par aucune variable.
- **Un mécanisme d'exception** contraignant dans lequel une exception ne peut pas être ignorée. Une méthode provoquant une exception doit soit la traiter, soit indiquer explicitement qu'elle ne la traite pas et la fait remonter. Une exception non capturée par aucune méthode provoque l'arrêt de la Machine virtuelle (JVM).
- **Le multi-Thread** : toute instance peut être gérée dans un nouveau Thread (tâche), les accès concurrents sont gérés par des moniteurs transparents pour le programmeur.
- **La sécurité** est assurée par trois couches :

Chapitre 3: Implémentation et réalisation

- a) Le “vérifier “ est intégré à la JVM et vérifie que le pseudo code qu’il reçoit est valide (il vérifie par exemple qu’un entier n’est pas utilisé comme une adresse mémoire).
 - b) Le “classloader “ vérifie que le chargement des classes est valide (il vérifie par exemple qu’une classe ne se fait pas passer pour une autre ou qu’une classe possède une signature valide).
 - c) Le “security manager“ contrôle l’accès d’un programme aux ressources (entrées/sorties, accès réseaux, chargement de bibliothèques binaires, etc.).
- **La portabilité de JAVA :** La compilation java ne prend pas en compte les caractéristiques systèmes de la machine sur laquelle elle est effectuée (on parle de pseudo code) l’interpréteur JAVA interprète alors le pseudo code et exécute les applications.

Ceci fait de JAVA un langage totalement indépendant des plates-formes et donc portable d’un système à l’autre. L’inconvénient du pseudo code est sa faible vitesse d’exécution.

Les navigateurs compatibles JAVA (Netscape Navigator, Microsoft internet Explorer,...etc.) intègrent un interpréteur de pseudo code (JVM) pour exécuter les programmes JAVA dans des pages Web (applets). [6]

Avantages de langages java :

- La portabilité des programmes écrits en JAVA : ainsi ils peuvent être exécutés sur différentes plates-formes et sous des systèmes d’exploitation distincts.
- L’accès à des bases de données à l’aide de l’API JDBC : les programmes peuvent consulter différentes bases de données qui utilisent chacune un SGBD différent.
- La possibilité de générer dynamiquement des pages Web personnalisées.
- La sécurité offerte par le langage.
- Le multi-thread : Un programme peut lancer et gérer l’exécution de plusieurs threads.

Chapitre 3: Implémentation et réalisation

- La gestion des exceptions : un programme écrit en JAVA peut prévoir dans son code l'arrivée d'une erreur (dans le réseau par exemple) et la traiter en cas où elle survient.
- La puissance de développement offerte par la programmation orienté objet.
- La possibilité de créer des applications distribuées.

III.3.1.2. IDE (NetBeans) :

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 et SPARC), Mac OS X et Open VMS.[7]

L'image suivante est une capture d'écran montrant l'environnement de développement de notre application :

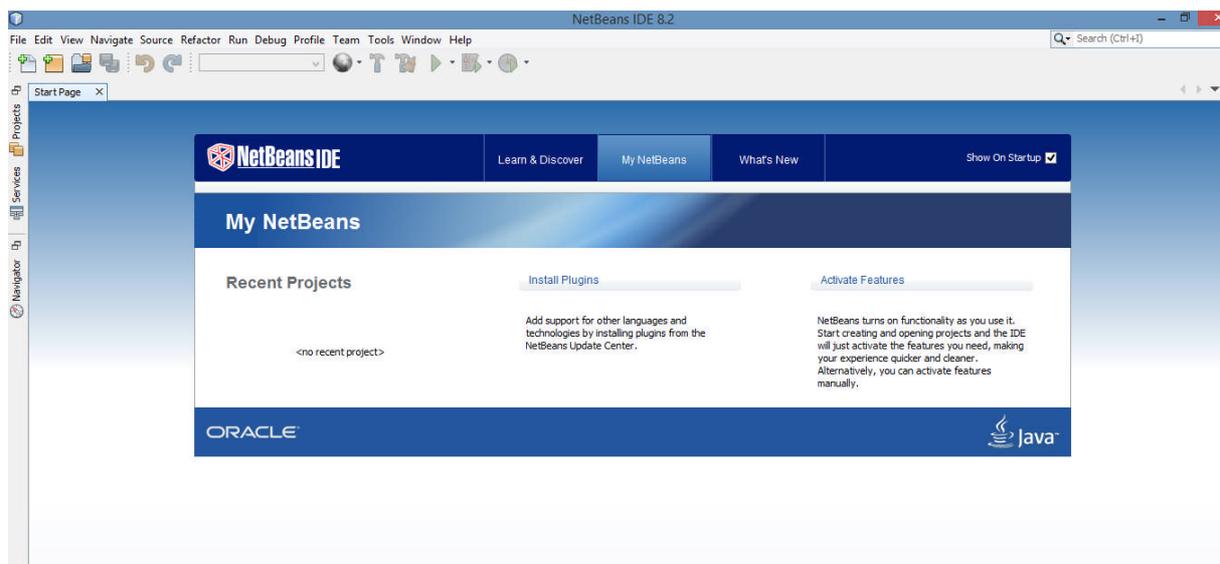


Figure III.12 : L'environnement de développement NetBeans

III.3.1.3. L'interface phpMyadmin :

PhpMyAdmin est une application web qui permet de gérer un serveur de bases de données MySQL. Dans un environnement multiutilisateur, cette interface permet également de donner à un utilisateur un accès à ses propres bases de données.

La figure suivante montre une copie d'écran de la page d'accueil de phpMyAdmin. L'écran est divisé en deux parties. Sur la gauche on peut afficher toutes les bases de données gérés par le serveur. la partie droite présente l'ensemble des opérations disponibles en fonction du contexte.



Figure III.13 : Page d'accueil de phpMyAdmin

III.3.1.4. WampServer :

WampServer est une plateforme de développement Web, permettant de faire fonctionner localement (sans avoir à se connecter à un serveur externe) des Scripts PHP. *WampServer* n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL.

Chapitre 3: Implémentation et réalisation

Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs au travers d'un *tray icon* (icône près de l'horloge de Windows).

La grande nouveauté de WampServer 2 réside dans la possibilité d'y installer et d'utiliser n'importe quelle version de PHP, Apache ou MySQL en un clic. Ainsi, chaque développeur peut reproduire fidèlement son serveur de production sur sa machine locale. [8]

III.3.1.5. Le Langage de requêtes SQL :

Pour communiquer avec une base de données, on a besoin de lui envoyer des commandes ou instructions appelées requêtes, Que ce soit pour la création, la suppression d'une table, la modification, l'insertion ou la sélection de données , le langage standard de requêtes est SQL.

SQL ou (Standard Query Language) est un langage permettant d'interroger les bases de données d'une manière simple. Il est doté d'une syntaxe particulière que l'on doit respecter pour que la communication avec la base se passe au mieux son succès est dû à sa simplicité et au fait qu'il énonce des requêtes en laissant le SGBD responsable de la stratégie d'exécution.

SQL couvre les trois fonctions indispensables à la mise en œuvre et à l'exploitation de base de données relationnelles :

- ✓ La création des données.
- ✓ La manipulation des données.
- ✓ Le contrôle des données.

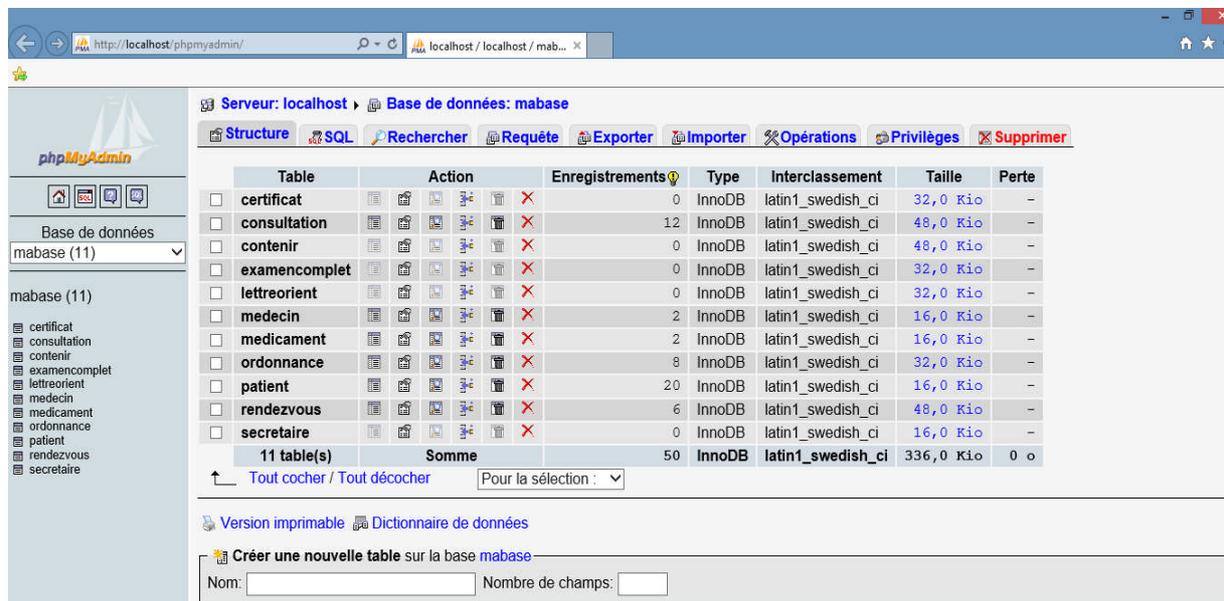
III.4. Implémentation de L'application :

III.4.1. Partie stockage :

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

Chapitre 3: Implémentation et réalisation

L'image suivante est une capture d'écran montrant les différentes tables de notre base de données :



The screenshot shows the phpMyAdmin interface for a database named 'mabase' on localhost. The main area displays a table with columns: Table, Action, Enregistrements, Type, Interclassement, Taille, and Perte. The table lists 11 tables: certificat, consultation, contenir, examencomplet, lettreorient, medecin, medicament, ordonnance, patient, rendezvous, and secretaire. A summary row shows 11 tables, 50 records, and a total size of 336,0 Kio.

Table	Action	Enregistrements	Type	Interclassement	Taille	Perte
certificat		0	InnoDB	latin1_swedish_ci	32,0 Kio	-
consultation		12	InnoDB	latin1_swedish_ci	48,0 Kio	-
contenir		0	InnoDB	latin1_swedish_ci	48,0 Kio	-
examencomplet		0	InnoDB	latin1_swedish_ci	32,0 Kio	-
lettreorient		0	InnoDB	latin1_swedish_ci	32,0 Kio	-
medecin		2	InnoDB	latin1_swedish_ci	16,0 Kio	-
medicament		2	InnoDB	latin1_swedish_ci	16,0 Kio	-
ordonnance		8	InnoDB	latin1_swedish_ci	32,0 Kio	-
patient		20	InnoDB	latin1_swedish_ci	16,0 Kio	-
rendezvous		6	InnoDB	latin1_swedish_ci	48,0 Kio	-
secretaire		0	InnoDB	latin1_swedish_ci	16,0 Kio	-
11 table(s)	Somme	50	InnoDB	latin1_swedish_ci	336,0 Kio	0

Figure III.14 : Les tables de la base de données

Après avoir créé cette base de données, on a établi une connexion NetBeans /MYSQL à l'aide de pilote OJDBC.

III.4.2.Partie traitement :

III.4.2.1.Le Projet Cabinetmedical :

La figure suivante est un capteur d'écran de notre projet :

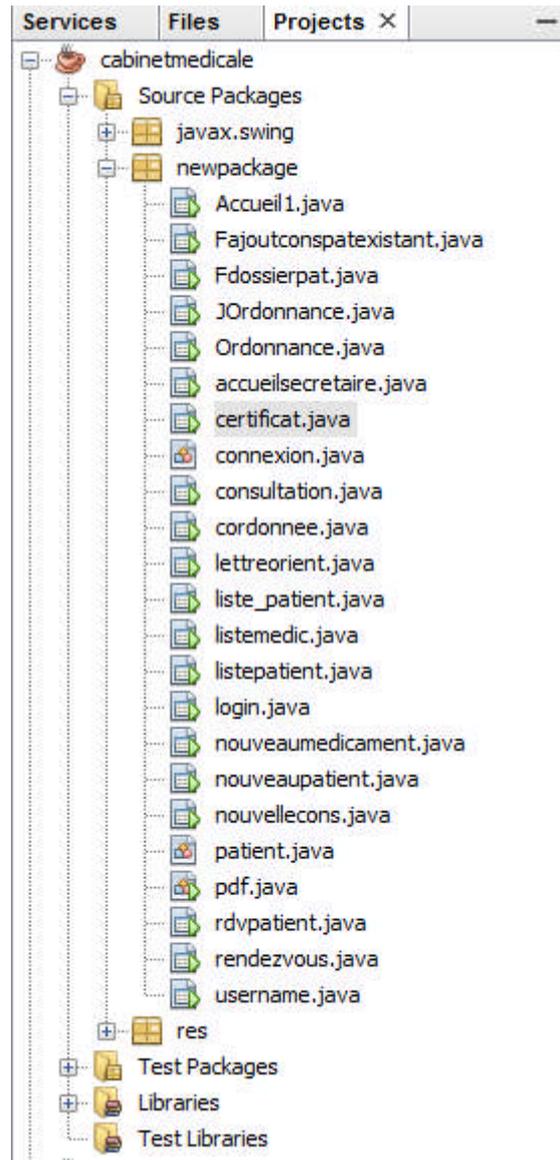


Figure III.15 : Implémentation du Projet

III.4.3. Partie Présentation :

À travers les interfaces présentées ci-dessous, nous visons à donner une vue générale de notre application conçue.

Fenêtre d'authentification :

C'est la première fenêtre visible par notre application, cette fenêtre conduit l'utilisateur à s'authentifier par son nom d'utilisateur et son mot de passe .Cette étape lui permet d'accéder à son propre espace



Figure III.16 : Fenêtre d'authentification

Fenêtre 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

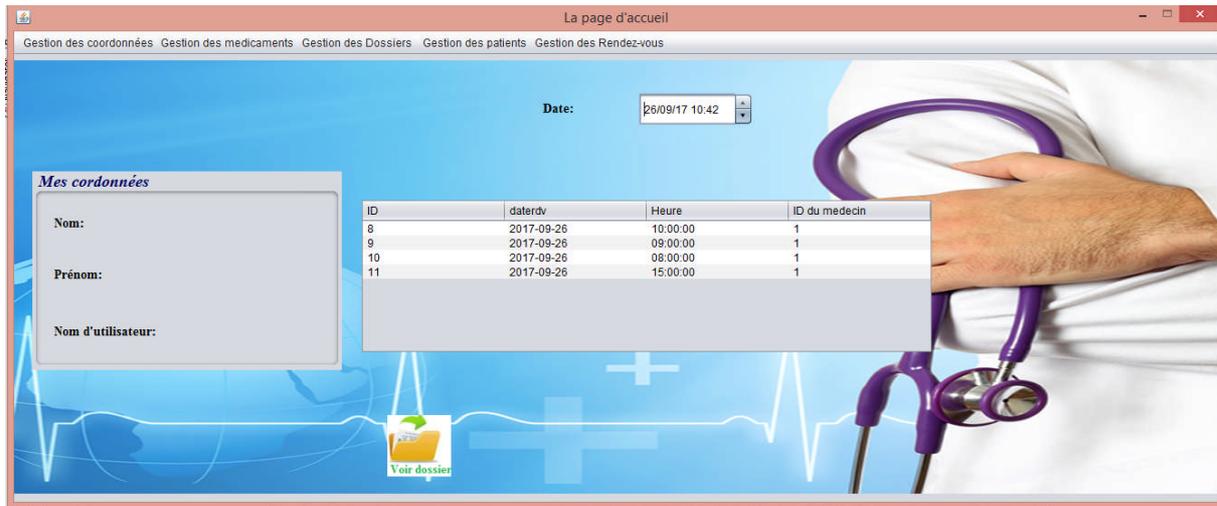


Figure III.17 : Fenêtre espace médecin

Fenêtre nouveau patient :

Cette fenêtre s'affiche en cliquant sur bouton Ajouter 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.



The image shows a software window titled "Nouveau patient" with a blue header. The window contains several input fields and buttons. The fields are: "Nom:" (text), "Prénom:" (text), "Date de naissance:" (text with a calendar icon), "Sexe:" (radio buttons for "Homme" and "Femme"), "Adresse:" (text), "Numéro de téléphone:" (text), "Age:" (text), and "Numéro de sécurité sociale:" (text). At the bottom, there are three buttons: "Quitter" (with a red X icon), "Ajouter" (with a green checkmark icon), and "modifier" (with a pencil icon).

Figure III.18 : Fenêtre Ajout d'un nouveau Patient

Fenêtre liste des Patient :

Cette fenêtre s'affiche en cliquant sur bouton liste des patient dans le menu de la fenêtre espace médecin

Chapitre 3: Implémentation et réalisation

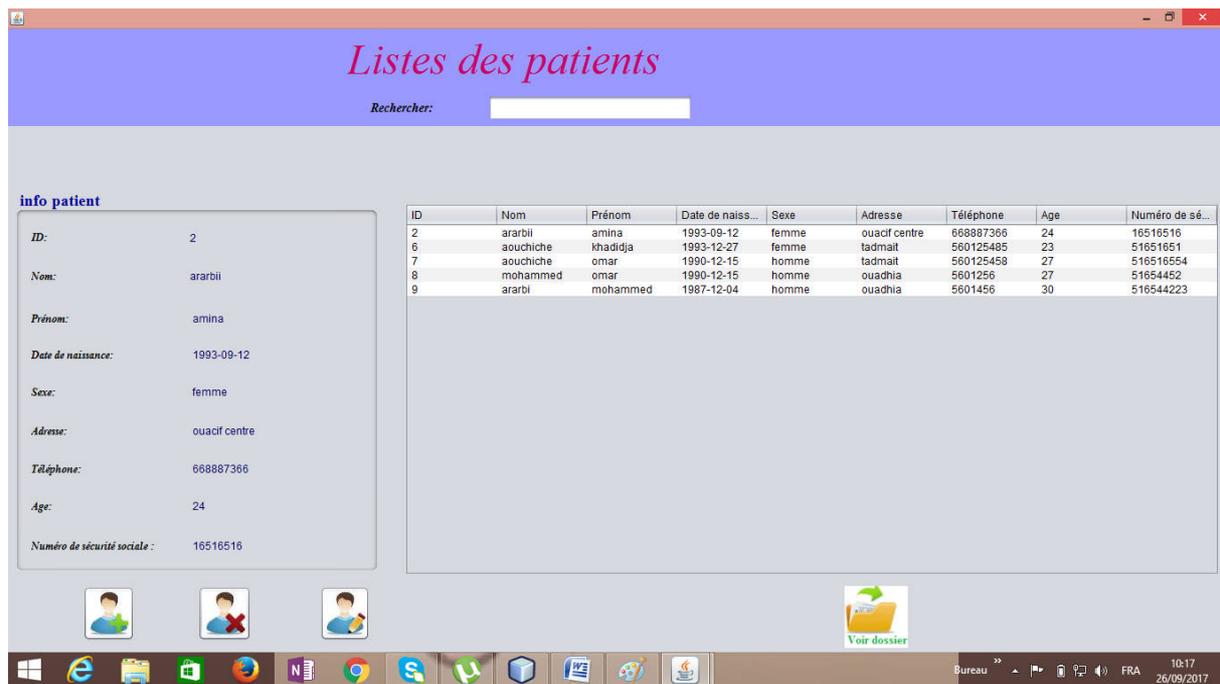


Figure III.19 : Fenêtre Liste des Patient

Fenêtre dossier médical :

Cette fenêtre s'affiche en cliquant sur bouton « détails » dans le la fenêtre liste patient après avoir sélectionné un patient. Elle contient les informations personnel, les consultations les ordonnances ...etc.

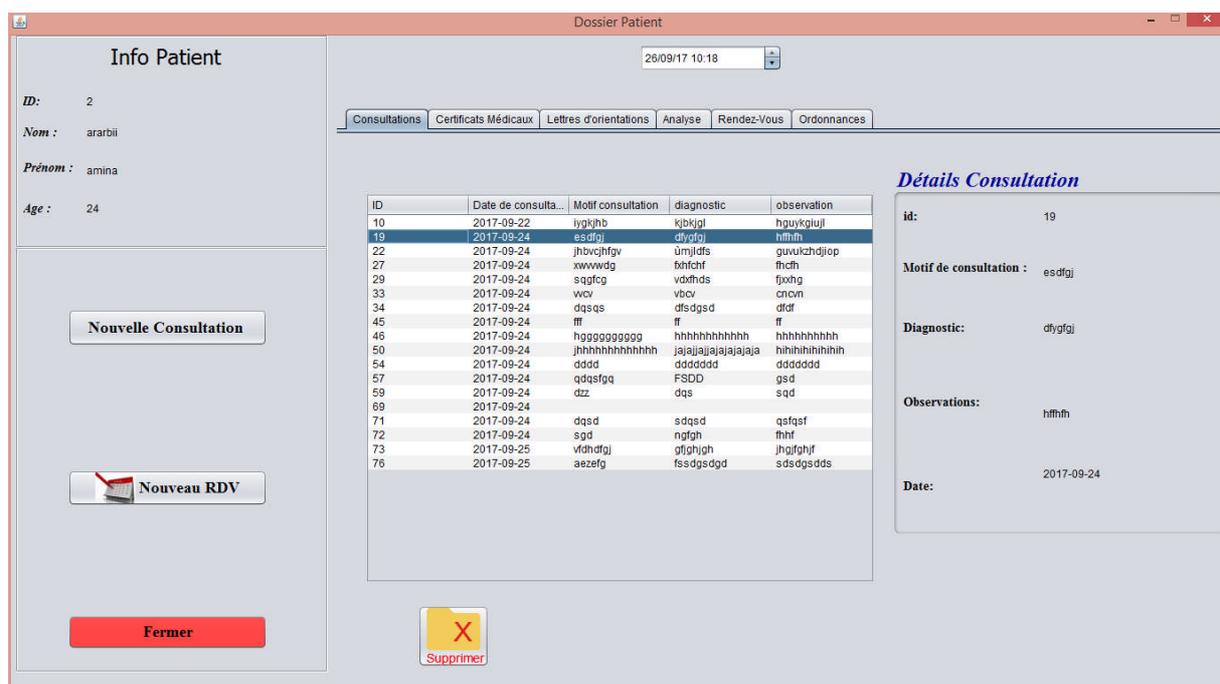
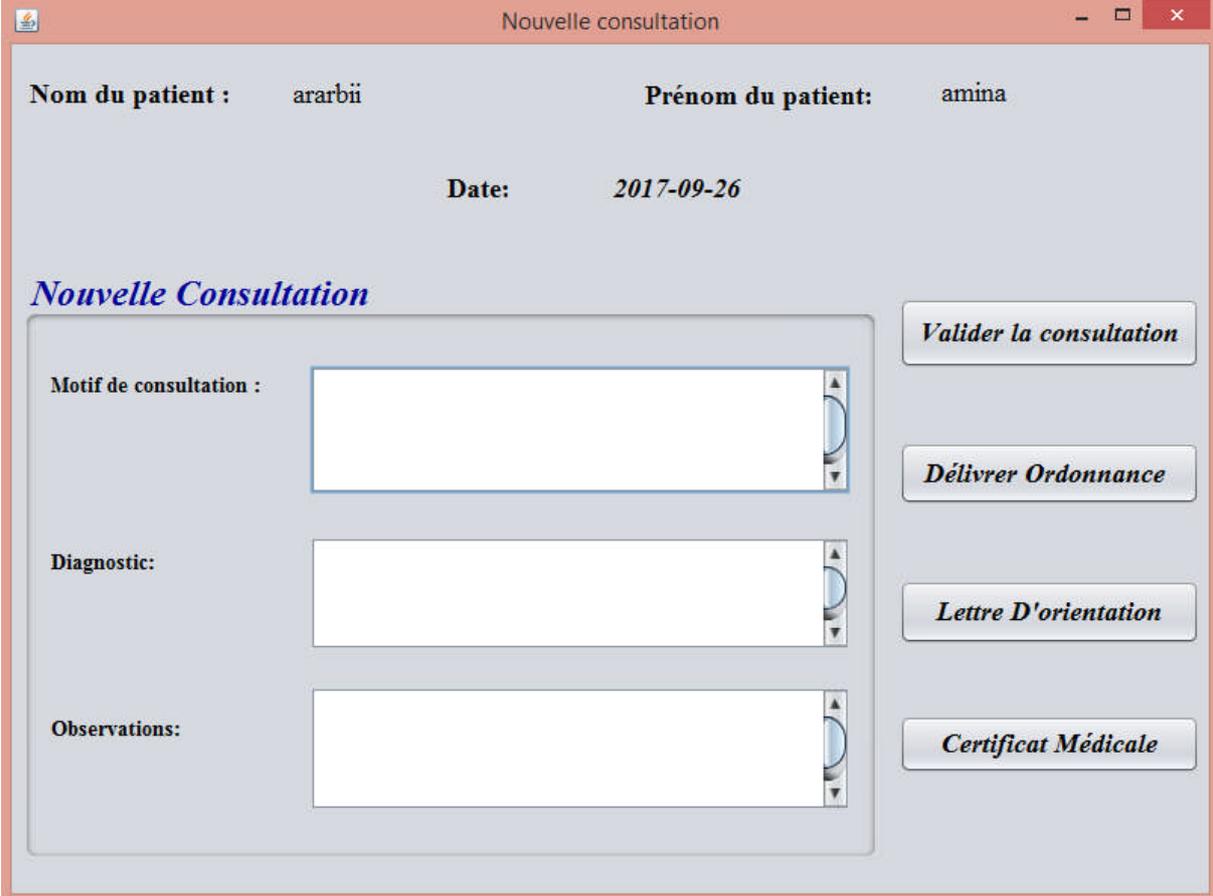


Figure III.20 : Fenêtre Dossier médical

Fenêtre nouvelle consultation :

Cette fenêtre s'affiche en cliquant sur bouton nouvelle consultation dans le menu de la fenêtre dossier médical



The screenshot shows a window titled "Nouvelle consultation" with a light gray background. At the top, it displays patient information: "Nom du patient : arabii" and "Prénom du patient: amina". Below this, the date is shown as "Date: 2017-09-26". The main area is titled "Nouvelle Consultation" in blue italicized text. It contains three text input fields: "Motif de consultation :", "Diagnostic:", and "Observations:". To the right of these fields are four buttons: "Valider la consultation", "Délivrer Ordonnance", "Lettre D'orientation", and "Certificat Médicale".

Figure III.21: Fenêtre nouvelle consultation

Fenêtre liste des consultations :

Cette fenêtre s'affiche en cliquant sur consultation dans le menu de la fenêtre dossier médical. Elle contient la liste des consultations du patient

Chapitre 3: Implémentation et réalisation

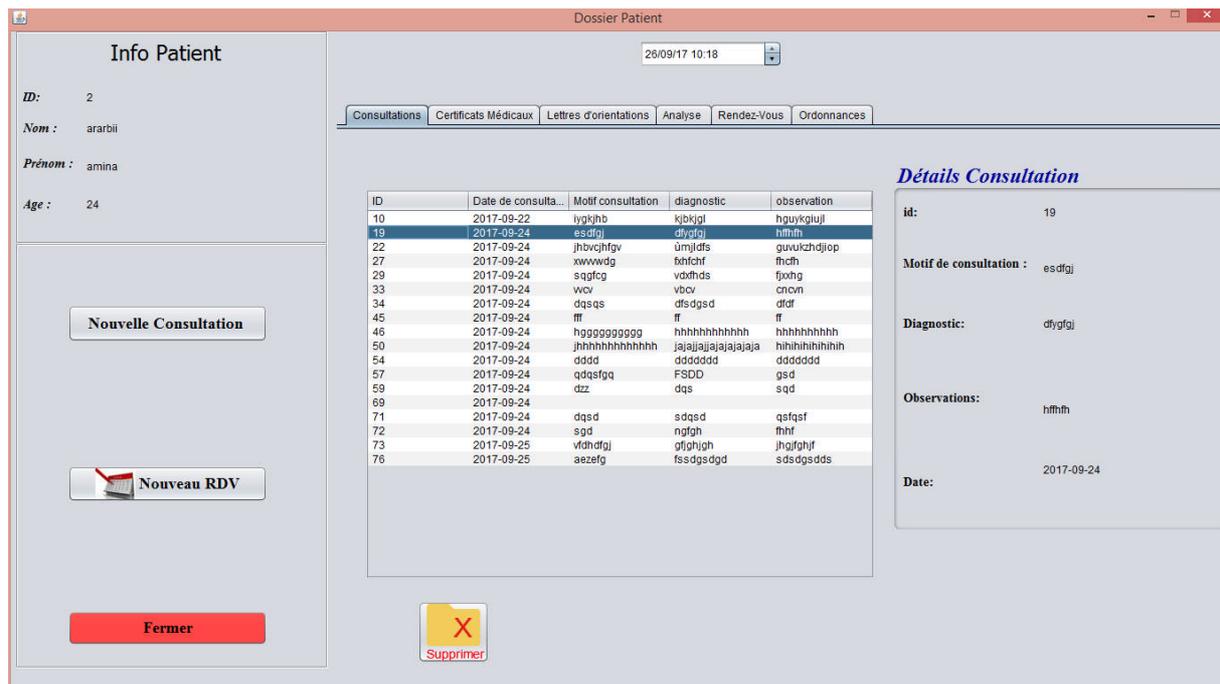


Figure III.22 : Fenêtre Liste des consultations

Fenêtre Détails d'une consultation :

Cette fenêtre s'affiche en sélectionnant une consultation dans la liste des consultations d'un patient. Elle contient les détails de la consultation.

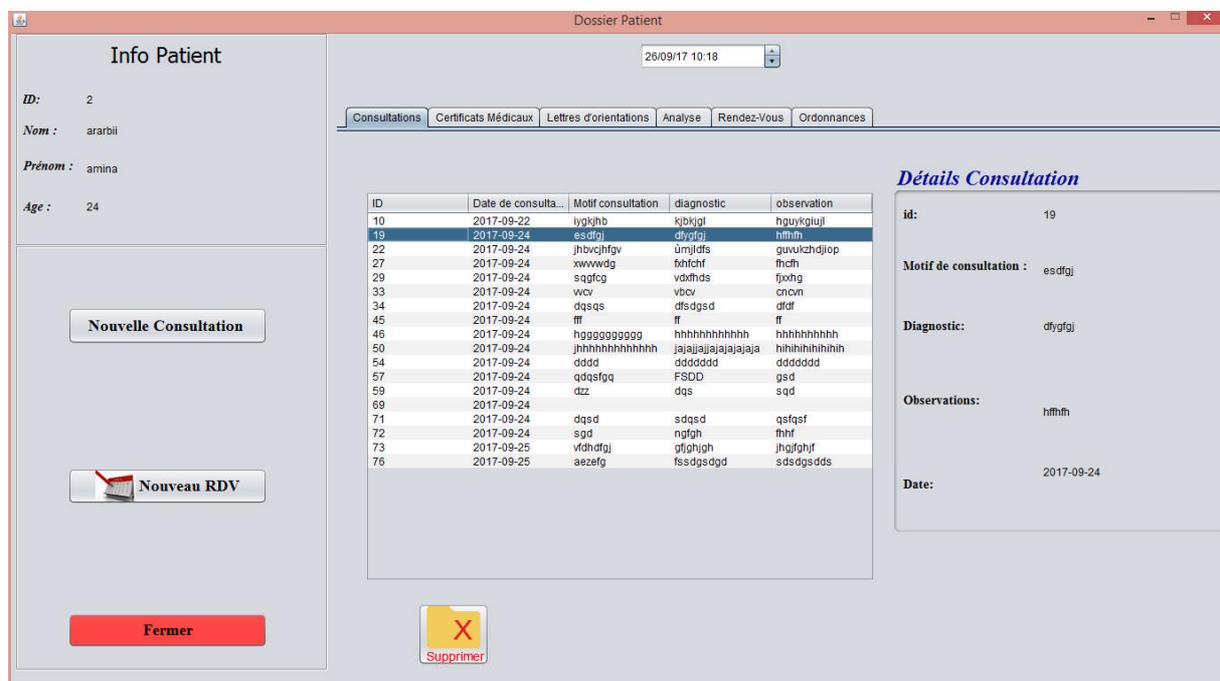
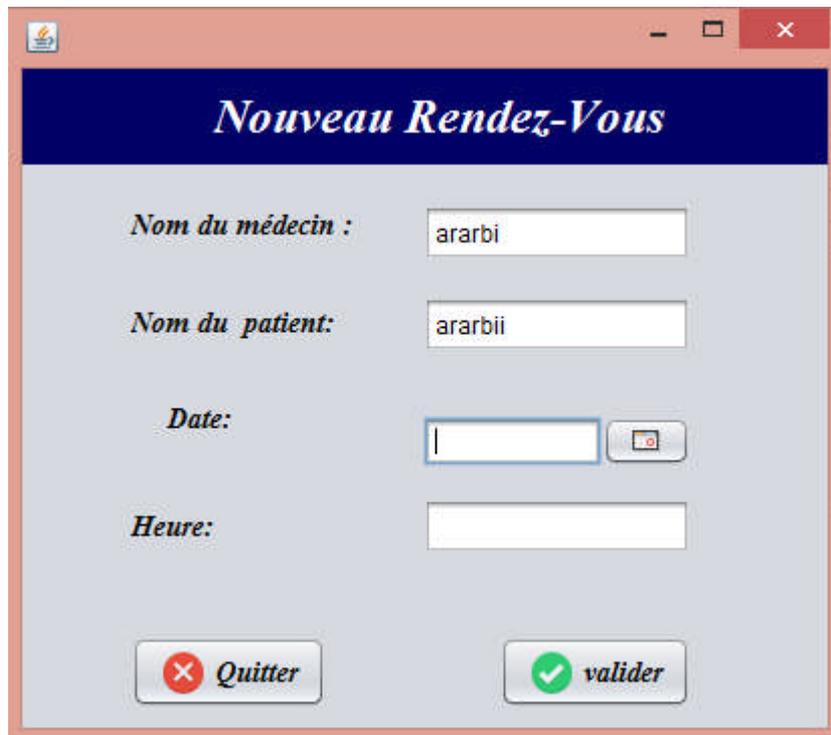


Figure III.23 : Fenêtre Détails Consultation

Fenêtre Nouveau Rendez-vous :

Cette fenêtre s'affiche en cliquant sur bouton « Rendez-vous » dans le la fenêtre liste patient après avoir sélectionné un patient. Elle permet d'ajouter un rendez-vous pour le patient sélectionné.



The screenshot shows a software window titled "Nouveau Rendez-Vous". The window has a dark blue header bar with the title in white italicized font. Below the header, the form is set against a light gray background. It contains four labeled input fields: "Nom du médecin :" with the text "ararbi", "Nom du patient:" with "ararbi", "Date:" with an empty field and a calendar icon, and "Heure:" with an empty field. At the bottom, there are two buttons: "Quitter" with a red 'X' icon and "valider" with a green checkmark icon.

Figure III.24 :Fenêtre nouveau Rendez-vous

Fenêtre Liste des médicaments :

Cette fenêtre s'affiche en cliquant sur bouton liste des médicaments dans le menu de la fenêtre espace médecin



Figure III.25 : Fenêtre Liste des médicaments

III.5. 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

Conclusion Générale

Le travail réalisé tout au long de notre stage au niveau de la cabinet médical « Lasers Dermatologiques » nous a permis de découvrir le domaine médical dermatologique et de voir sur le terrain l'ensemble des tâches effectuées par les médecins et la secrétaire

L'objectif de ce travail était de concevoir une application informatique pour la gestion des dossiers médicaux de la cabinet médical dermatologiques « Lasers Dermatologiques ».

Ce travail nous a permis :

- D'acquérir une expérience dans la conception avec la notation UML.
- D'approfondir nos connaissances en matières programmation JAVA, SQL.
- S'adapter a l'environnement de développement Netbeans.

Notre application qui a traité la gestion des dossiers médicaux, offre une meilleure communication entre les médecins et la secrétaire.

Notre application a un effet structurant sur l'organisation des Système d'information de la cabinet, quoi qu'elle présente diverses fonctionnalités (gestion des dossiers médicaux, gestion des patients, gestion de la bibliothèque médicale ...) sera toujours perfectible pour une meilleure disponibilité des informations sur les patients.

LA TABLE DES FIGURES

Figure I.1 : Fiche d'analyse du poste de travail n°1 (Médecin)

Figure I.2 : Fiche d'analyse du poste de travail n°2 (Secrétaire)

Figure I.3 : Tableau de la liste des documents existants

Figure I.4 : Fiche d'analyse du Dossier Médical

Figure I.5 : Fiche d'analyse de L'Ordonnance

Figure I.6 : Fiche d'analyse de la Certificat Médical

Figure I.7 : Fiche d'analyse de la Fiche d'analyse

Figure I.8 : Fiche Résultat d'analyse

Figure I.9 : Le graphe des flux de données

Figure II.1: Représentation graphique de la Démarche de modélisation

Figure II.2: Diagramme de contexte

Figure II.3: Diagramme de cas d'utilisation général

Figure II.4: Diagramme de séquence du cas d'utilisation « S'authentifier »

Figure II.5: Diagramme de séquence du cas d'utilisation « Nouvelle Consultation »

Figure II.6: Diagramme de séquence du cas d'utilisation « Ajouter un patient »

Figure II.7: Diagramme de séquence du cas d'utilisation « Modifier un Patient »

Figure II.8: Diagramme de séquence du cas d'utilisation « Ajouter Rendez-vous »

Figure II.9: Diagramme de classe détaillé

Figure III.1 : Table medecin

Figure III.2 : Table Secretaire

Figure III.3 : Table patient

Figure III.4 : Table consultation

Figure III.5 : Table ordonnance

Figure III.6 : Table medicament

Figure III.7 : Table contenir

Figure III.8: Table rendezvous

Figure III.9 : Table lettreorientation

Figure III.10 : Table CertificatMédicale

Figure III.11 : Table Examencomplet

Figure III.12 : L'environnement de développement NetBeans

Figure III.13 : Page d'accueil de phpMyAdmin

Figure III.14 : Les tables de la base de données

Figure III.15 : Implémentation du Projet

Figure III.16 : Fenêtre d'authentification

Figure III.17 : Fenêtre espace médecin

Figure III.18 : Fenêtre Ajout d'un nouveau Patient

Figure III.19 : Fenêtre Liste des Patient

Figure III.20 : Fenêtre Dossier médical

Figure III.21 : Fenêtre nouvelle consultation

Figure III.22 : Fenêtre Liste des consultations

Figure III.23 : Fenêtre Détails Consultation

Figure III.24 : Fenêtre nouveau Rendez-vous

Figure III.25 : Fenêtre Liste des médicaments

Bibliographie

1. MAST 88 année 2012 conception et réalisation d'une application pour la gestion d'un cabinet médical
2. LIC 69 année 2013 conception et réalisation d'une application pour la gestion du dossier médical
3. Michel lai « penser objet avec UML et java » Dunod 2004
4. « Modélisation avec UML Partie2 » Samira SI-SAID CHERFI

Netographie

- 5 . <http://www.memoireonline.com>
6. <https://java.developpez.com/cours/>
7. <https://netbeans.org/kb/trails/platform.html>
8. <https://fr.wikipedia.org/wiki/WampServer>