

جامعة أبو بكر بلكايد - تلمسان
Université Abou Bakr Belkaïd de Tlemcen



Faculté de Technologie



Département de Génie Biomédical

Laboratoire de Recherche de Génie Biomédical

Mémoire de fin d'étude pour l'obtention du diplôme

MASTER en Génie Biomédical

Spécialité : Informatique Biomédicale

THEME

Gestion de Cabinet Médical

Présenté par :

Mr. BENEDDRA Rachid

Soutenu le 14 juin 2015 devant le Jury

Mr.	MOUSSAOUI Djillali	Université de Tlemcen	Président
Mr.	GAOUAR Imad	Université de Tlemcen	Examineur
Mr.	<i>ABDERRAHIM Mohammed Alaeddine</i>	Université de Tlemcen	Encadreur

Année Universitaire 2014-2015

Dédicaces

Je dédie ce modeste travail :

A mes très chers parents, *Mohammed* et *Hasnia*, qui ont toujours été là pour moi, « Vous avez tout sacrifié pour vos enfants n'épargnant ni santé ni efforts, vous m'avez donné un magnifique modèle de labeur et de persévérance. Je suis redevable d'une éducation dont je suis fier ».

A mes frères *Abdelkader* et *Hamid*, Samir, Mourad, Kamel, Hichem et mes sœurs *Souhila* et *Karima* pour leur encouragement.

A ma très chère amie Nesrine qui est toujours été la pour moi.

A toutes mes très chères amis Oussama, Youcef, Khaled, Bilel, Mohammed, Amine, Lila, Nassima, Safia.

A tous mes collègues de promotion avec qui j'ai passé mes meilleures années d'études.

A toutes celles et tous ceux qui m'ont aidé dans mes études.

Tous ceux que je connais et que je n'ai pas pu citer.

Remerciements

Je remercie *Dieu* tout puissant de la patience et de la volonté qui m'a donné pour réaliser ce projet de fin d'études.

Je tiens à exprimer toute ma reconnaissance à mon directeur de mémoire *Mr. Abderrahim Mohammed Alaeddine*, je le remercie de m'avoir encadré, orienté, aidé et conseillé.

J'adresse mes sincères remerciements à tous les professeurs, intervenants et toutes les personnes qui par leurs paroles, leurs écrits, leurs conseils et leurs critiques ont guidé mes réflexions et ont accepté à me rencontrer et répondre à mes questions durant mes recherches.

Je voudrais remercier tous ceux qui ont apporté une contribution dans la réalisation de ce projet de fin d'étude.

Ainsi, je remercie les membres *jury* d'avoir accepté d'examiner le projet.

À tous ces intervenants, je présente mes remerciements, mon respect et ma gratitude.

Table des matières

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

Chapitre I : Présentation théorique

Introduction.....	2
I. Présentation du cabinet.....	2
I.1. Etude de l'existant.....	3
I.1.1. Présentation de la mission.....	3
1. Gestion et suivi du Dossier Médical.....	3
2. Gestion des rendez vous.....	4
3. Gestion du Fiche Patients.....	4
4. Gestion des ordonnances.....	4
5. Gestion de Comptabilité.....	4
6. Gestion du Fiche Médecins.....	5
I.1.2. Présentation de l'application.....	5
I.2 Critique de l'existant.....	5
II. La spécification des besoins.....	6
II.1. Identification des besoins.....	6
II.2. Identification des acteurs.....	7
1. Patient.....	7
2. Médecin.....	7
III. Proposition d'une solution.....	7
IV. Conclusion.....	8

Chapitre II : Spécification du système

Introduction.....	9
I. Acteur du système informatisé.....	9
I.1. Les utilisateurs de notre application.....	10
I.2. La base de données pour notre application.....	10
II. Conception des schémas logiques et physique des données.....	10
III. Les diagrammes d'UML.....	13
III.1. Elaboration de modèle des cas d'utilisation.....	13
III.1.1. Diagramme des cas d'utilisation.....	13
III.1.1.1. Identifications des acteurs.....	15
III.1.1.2. Identification des cas d'utilisation.....	15
III.1.1.3. Diagramme de cas d'utilisation général de l'application.....	16
1. Cas d'utilisation pour la gestion du fiche patient.....	16
2. Cas d'utilisation pour la Consultation.....	16
3. Cas d'utilisation des ordonnances.....	17
4. Cas d'utilisation pour la gestion des rendez-vous.....	17
5. Cas d'utilisation pour la gestion de la comptabilité.....	17
III.1.2. Description textuelle des cas d'utilisation.....	17
1. pour cas d'utilisation gestion du Fiche patient.....	18
2. pour cas d'utilisation gestion des Rendez-vous.....	19

3.	pour cas d'utilisation gestion des consultations.....	20
4.	pour cas d'utilisation gestion des Ordonnances.....	22
5.	pour cas d'utilisation gestion des Recettes.....	23
6.	pour cas d'utilisation gestion des fiches médecins.....	24
III.2.	Diagramme de séquence.....	26
1.	Diagramme de séquence du fiche patient.....	26
2.	Diagramme de séquence de la consultation.....	27
3.	Diagramme de séquence des ordonnances.....	27
4.	Diagramme de séquence des Factures.....	28
III.3.	Construction du diagramme de classes.....	28
III.3.1.	La liste des supports d'information.....	29
III.3.2.	Conception des classes.....	32
IV.	Conclusion.....	34

Chapitre III : Implémentation et réalisation

Introduction.....	35	
I.	Outils et langages utilisé.....	35
I.1.	Environnement de réalisation.....	35
I.1.1.	L'environnement matériel.....	35
I.1.2.	L'environnement logiciel.....	36
II.	Architecture générale du projet.....	37
III.	Réalisation du projet.....	39
IV.	Conclusion.....	48
Conclusion générale.....	49	

Liste des figures

Chapitre I : Présentation théorique

Figure I.1 : salle du soin.....	2
---------------------------------	---

Chapitre II : Spécification du système

Figure II.1 : La forme d'un Actor.....	10
Figure II.2 : Diagramme de cas d'utilisation général de l'application.....	16
Figure II.3 : diagramme de séquence du fiche patient.....	26
Figure II.4 : diagramme de séquence de la consultation.....	27
Figure II.5 : diagramme de séquence des ordonnances.....	27
Figure II.6 : diagramme de séquence des Factures.....	28
Figure II.7 : Diagramme de classes.....	33

Chapitre III : Implémentation et réalisation

Figure III.1 : Architecture générale du projet.....	38
Figure III.2 : Fenêtre d authentification.....	39
Figure III.3 : Menu pour médecin.....	40
Figure III.4 : Interface gestion de fiches médecins.....	41
Figure III.5 : Interface gestion du fiche patients.....	42
Figure III.6 : Interface gestion des rendez-vous.....	43
Figure III.7 : Interface gestion des rendez-vous « Nouveau RDV ».....	44
Figure III.8 : Interface pour la page de consultation.....	45
Figure III.9 : Interface Pour la Gestion et suivi du dossier médical.....	45
Figure III.10 : Interface Pour la Gestion des ordonnances.....	46
Figure III.11 : Interface gestion de la comptabilité.....	47

Introduction Générale :

La démarche médicale est fondée sur l'observation du malade, la mémoire du médecin était autrefois suffisante pour enregistrer les données relatives aux patients et servir l'exercice médical, les données médicales étaient rassemblées sous forme d'articles médicaux, de registres à visée épidémiologique, nosologique et administrative, avec la multiplication des effets de l'environnement, de nos jours la bonne tenue d'un dossier exige des moyens informatiques.

L'automatisation du système d'information consiste à structurer et gérer un ensemble de données dont le but de les organiser et d'avoir des résultats rapides.

Dans ce cadre, nous sommes appelés à concevoir, développer et mettre en place un logiciel pour la gestion d'un cabinet médical.

Le logiciel devrait mettre l'organisation et l'automatisation de la gestion d'un cabinet médical, afin d'augmenter la fiabilité, l'efficacité de l'effort humain et faciliter les tâches pénibles au sein du cabinet.

Notre application comprendra les fonctionnalités suivantes :

- Gestion et Suivi du Dossier Médical
- Gestion des Rendez-vous.
- Gestion de la Fiche Patient.
- Gestion de la Fiche Médecin.
- Gestion de la Comptabilité.
- Gestion des Ordonnances.

Le présent projet s'articule autour de trois chapitres qui sont présentés comme suit :

- Le premier chapitre est consacré à la Présentation théorique et les besoins fonctionnels pour le futur système.
- Le deuxième chapitre définit les différents diagrammes d'UML, et la Spécification de notre futur système.
- Le troisième chapitre permet de concevoir, réaliser, et représenter graphiquement les activités des différentes interfaces du système.

Chapitre I

Introduction :

Avec les changements des besoins des utilisateurs et l'apparition de nouvelles technologies de stockait les informations et de le trouver quand on veut, de nombreux personnes cherche à investir dans le domaine des applications de gestion et chaque entreprise souhaite une bonne maitrise de la gestion de ces systèmes d'informations. La gestion du cabinet médical est l'ensemble des règle qui facilite et permet aux utilisateurs de gérer tout les taches de la scolarité afin d'avoir une meilleure organisation du travail pour gagner le temps et éviter l'encombrement des fiches et dossier médicaux du patients.

Durant notre étude on essaye d'analyser le domaine de cabinet médical afin d'automatiser quelques ces aspects.

Ce chapitre d'écrit : les différents règles du cabinet médical, l'organisation d'étude de l'existant de notre travaille, va nous servir à poser les bases des besoins du système à réaliser il consiste à effectuer un premier repérage des besoins fonctionnels et opérationnels.

I. Présentation du cabinet :

Un Cabinet médical est un établissement de soins où un personnel soignant peut prendre en charge des personnes malades pour le diagnostic et / ou la thérapie.

Le médecin et le patient sont les acteurs principaux du cabinet médical.

Voila une image sur la salle du soin dans un cabinet médical :



Figure I.1 : salle du soin

I.1. Etude de l'existant

I.1.1. Présentation de la mission :

Notre Mission dans le cadre de ce projet est de créer une application permettant de gérer le cabinet médical il s'agit de définir les responsabilités de la gestion, mettre à jour les données, organiser des données collectées, de renforcer le contrôle et la confrontation, assurer une meilleure gestion médicale et une cohérence de l'information et enfin faciliter le travail des responsables.

Notre application aura comme principale fonctionnalités :

- Gestion et Suivi du Dossier Médical (détaillé).
- Gestion des Rendez-vous.
- Gestion de la Fiche Patient.
- Gestion des Ordonnances.
- Gestion de la Comptabilité.
- Gestion de la Fiche Médecin

1. Gestion et Suivi du Dossier Médical :

En commençant par la consultation, est l'activité principale du cabinet médical, le patient qui s'adresse à un cabinet médical pour la première fois, une visite en faisant consulter par le médecin, ce dernier lui amène la fiche médicale descriptive ainsi que son dossier médical.

L'écoute attentive et patiente des propos du patient est un moment privilégiée de la consultation.

L'entretien doit se dérouler dans la stricte intimité et confidentialité pour permettre au patient de s'exprimer clairement et sincèrement sur ses préoccupations.

Ensuite le médecin l'examine à l'aide de ses outils (Stéthoscope, Tensiomètre, Thermomètre, etc.), il rédige ensuite l'ordonnance qui contient les noms des médicaments, les formes, les doses et la durée de jour de prise.

Dans le cas où le médecin n'est pas sûr de son diagnostic, il peut demander au patient de faire des examens complémentaires (Bilan biologique ou Bilan radiologique), il donne un rendez vous pour continue le bon diagnostic.

A chaque consultation selon le cas, surtout l'état de santé du patient, si la consultation lui a causé un contretemps, et ou un empêchement de son activité le certificat sera utile pour la justification.

La tenue du dossier médical du malade est une obligation professionnelle pour identifier le patient, assurer un suivi précis de sa pathologie et son évolution.

Le dossier médical est un document médico-légal justifiant la consultation et l'attitude thérapeutique qui en découle, il doit être soigneusement gardé par le médecin dans une enceinte sûre, fermant à clef, sa tenue relève de l'obligation du médecin au secret médical,

le dossier doit être archivé et gardé aussi longtemps que possible car un acte médical peut être remis en cause.

En cas de visite de contrôle ou visite périodique d'un patient en maladie de longue durée, celle-ci sera programmée à un moment précis de la journée.

A chaque consultation un résumé de la nouvelle consultation et du traitement donné sera porté sur le dossier médical.

2. Gestion des rendez vous :

Il peut être nécessaire d'organiser sa consultation sur rendez-vous si le besoin s'en fait sentir et le médecin se doit de les respecter scrupuleusement, le cas échéant, ceci doit être signalé aux patients, cependant il faut tenir compte des urgences qui ne peuvent souffrir aucune attente et admettre également la souplesse et la disponibilité requises.

La prise d'un RDV s'effectue directement ou par une communication téléphonique en donnant le nom, le prénom, la date et l'heure souhaitée, et selon la disponibilité du médecin, un RDV sera fixé. Le médecin est chargé de remplir les renseignements sur la fiche d'un patient (Nom, Prénom).

3. Gestion de la Fiche Patient:

Prise en charge des patients :

S'il s'agit d'un ancien patient, le médecin demande le nom et prénom pour effectuer la recherche de sa fiche parmi les fiches médicales qui sont rangées par ordre alphabétique dans les boîtes d'archives, elle prépare, aussi son dossier médical contenant suivi précis de sa pathologie et son évolution.

La fiche du patient doit contenir : matricule, nom, prénom, date de naissance, Antécédents, adresse, sexe, group sanguin, téléphone.

L'observation médicale rédigée par le médecin doit comprendre les antécédents du patient qui sont les données de son terrain (chirurgicaux, familiales, allergie, maladie chronique...). Ces données sont capitales pour les consultations ultérieures et toute thérapeutiques.

4. Gestion des ordonnances :

Après l'examen clinique le médecin donne un traitement au malade, une ordonnance d'aller acheter.

L'ordonnance comporte les différents médicaments avec leurs formes, posologies, dosages.

5. Gestion de la Comptabilité:

Quand le patient est terminé leur visite, le médecin passe à la facturation : le médecin calcule le montant final de la visite au patient.

6. Gestion de la Fiche Médecin :

La fiche du médecin doit contenir matricule, nom, prénom, photo, spécialité, adresse, téléphone. Chaque consultation elle a un médecin traitant.

I.1.2. Présentation de l'application

Cette application que nous nous sommes proposé de développer pour la gestion de l'ensemble des activités existantes dans ce cabinet, doit permettre de répondre aux exigences de ce dit cabinet. Pour le développement de cette application, nous avons jugé nécessaire d'utiliser les différents outils et méthodes qui sont les suivants :

- ✓ Pour la Programmation, nous utilisons **Netbeans7.4 langage java.**
- ✓ Pour le traitement de texte, nous travaillons avec **Microsoft Office 2007.**
- ✓ Nous utilisons la base de données **PHP MySQL.**

• Objectif à atteindre :

Dans le cadre de notre projet de fin d'études, il nous a été confié de faire l'étude du système informatique relatif à la gestion d'un cabinet médical, enfin de créer une application complète.

Pour cela, nous avons commencé par une critique de l'existant.

I.2 Critique de l'existant :

Cette partie a pour but de dégager les insuffisances et les défaillances du système actuel, relatif à la gestion d'un cabinet médical dont on peut citer :

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 une fiche, le médecin doit faire une recherche manuelle fiche par fiche par nom du patient, ce qui engendre une perte de temps même en cherchant est face au risque du quel les fiches peuvent se mélanger et surtout leurs contenus.

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 un tel dossier soit rangé par erreur dans un autre dossier lors de l'organisation et le stockage dans les boites d'archives.

Absence de la notion de confidentialité à cause de non séparation entre fiche médicale et dossier médical : on remarque que le médecin peut accéder aux informations confidentielles du patient, or le respect du secret médical impose que seul le médecin peut consulter ce dossier.

La gestion des RDV, se fait d'une manière manuelle ce qui provoque un risque d'oubli ou chevauchement des RDV.

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.

La gestion des recettes et dépenses n'est pas bien définie.

La perte de temps qui est remarquable en cas d'augmentation du nombre des patients pour la consultation

La gestion des documents administratives tout à la longueur de la journée qui sont : la saisie des ordonnances médicales encore a chaque fois lors d'élaboration des ordonnances, les médecins ont tendance à regarder une listes des médicaments leurs nom, signification, effets etc.... ce qui est tout a fait gênant a cause du temps et le nombre important des patients en attente.

II. La spécification des besoins :

Dans cette partie on va identifier les besoins et les acteurs principaux de notre futur système.

II.1. Identification des besoins :

On va identifier les besoins de l'arriver du patient au cabinet médical jusqu'à la fin (la sortir du cabinet).

1. Un Patient arrivant dans le cabinet médical et demande un Rendez-vous.
 - Si le patient est un nouveau client le médecin ajouter d'abord ces informations relative puits il donne un rendez vous.
 - Si le patient est un client ancien le médecin lui demander son nom pour faire une recherche sur leur fiche, pour confirmé le rendez-vous.
2. Il demande ensuite de faire une visite médicale / ou une consultation.
3. Procédure de consultation en enregistrant les informations de consultation.
4. Le médecin lui donne un traitement après le diagnostic (une ordonnance d'aller acheter).
5. Le système imprime une ordonnance au patient.
 - Si le traitement n'a pas terminai le médecin lui donne un autre rendez-vous pour complété ce traitement.
 - Si non le médecin passe à la facturation.
6. Le système fait un encaissement des frais de consultation.
7. Le patient demande un reçu.
8. Système imprime le reçu pour le patient.

II.2. Identification des acteurs :

1. **Patient** : Personne qui subit un traitement, diagnostic et / ou la thérapie, etc. Dans le domaine de la médecine, un patient est une personne recevant une attention médicale ou à qui est prodigué un soin. [1]

En médecine, le patient bénéficie d'examens médicaux, de traitements prodigués par un médecin ou autres professionnels de la santé pour faire face à une maladie ou à des blessures. [1]

2. **Médecin** : Personne qui diagnostique et en traitant les maladies. Un médecin est un professionnel de la santé titulaire d'un diplôme de docteur en médecine, il soigne les maladies, pathologies et blessures, il travaille généralement au sein d'une équipe de professionnels de la santé comme le psychologue, le pharmacien, l'infirmière ou le chirurgien-dentiste. [2]

III. Proposition d'une Solution :

Après avoir fait critique de l'existant et détecter les anomalies de la procédure actuelle, une approche de solution qui consiste à concevoir et à développer une application qui facilitera les insuffisances et les défaillances énumérés précédemment.

On propose alors de concevoir une nouvelle application permettant l'organisation et l'automatisation des tâches.

L'informatisation de la gestion du cabinet a des avantages certains et la simplification des tâches n'est plus à discuter tant pour le fichier des malades que pour la tenue d'une comptabilité simplifiée et du traitement de texte pour la correspondance et l'établissement d'ordonnances, de certificats médicaux ou autres :

Mettre en place un logiciel afin de gérer facilement chaque module à part, implanter une base de données complète pour la gestion des RDV, fiches médicales, consultations médicales, dossiers médicaux, assurer une meilleure communication et cohérence de l'information.

Optimiser le temps d'accès aux différentes données, éviter les tâches pénibles et ennuyeuses.

Définir une bonne organisation des données collectées pour faciliter la recherche des documents, aider le médecin pour la prise de décision avec des supports informatisés à l'appui.

Mettre en place un système qui gère tous médicaments de façon détaillée et rapide pour avoir des informations tel que la définition et les effets, quantité prise selon la maladie, etc.

Gérer les droits d'accès afin de permettre un accès sélectif aux différents menus et attribuer des responsabilités à chaque utilisateur : on doit assurer la séparation entre fiche médicale et dossier médical seul le médecin peut consulter le dossier médical.

Donner beaucoup d'importance à l'interface Homme-machine et la simplifier au maximum à l'utilisateur de l'application.

IV. Conclusion :

Dans ce chapitre on a essayé de faire une étude préalable de notre projet dont le but d'avoir une idée sur la gestion de cabinet médical et sur les différentes contraintes de développements.

La Modélisation du métier permet donc de comprendre, d'analyser les différentes activités du cabinet médical et de dégager éventuellement des critiques pour pouvoir proposer des solutions et en construisant un modèle de l'organisation d'un cabinet médical enfin de créer une application.

Pour réaliser une application de gestion, la capture des besoins de cette application fera l'objet du chapitre suivant.

Chapitre II

Introduction :

Ce présent chapitre est de présenter un recueil des besoins fonctionnels et techniques envers le système (l'utilisateur de notre système). Les fonctionnalités et les techniques des besoins du système sont basés sur différents aspects, nous partons depuis l'utilisateur jusqu'au fonctionnement du système en passant par l'authentification, la base de données etc.

Un modèle d'analyse qui nous permet de faire une représentation transitoire entre l'expression des besoins d'une part et le modèle de conception d'autre part, le modèle d'analyse permet de reformuler les besoins sous une forme proche de ce que sera la conception mais tout en s'abstrayant de leurs contraintes techniques. Il existe une variété des méthodes de conceptions.

Ce chapitre décrit les principales règles du langage de modélisation UML avec l'utilisation de cette dernière sur la gestion cabinet médical.

I. Acteur du système informatisé

- *Définition :*

On définit l'acteur du futur système et leur rôle, en faisant le diagramme de classe.

L'acteur est: « **Médecin** »

Le diagramme de classes exprime la structure statique du système en termes de classes et de relations entre ces classes, leur intérêt est de modéliser les entités du système d'information, il permet de représenter l'ensemble des informations finalisées qui sont gérées par le domaine, ces informations sont structurées, c'est-à-dire qu'elles ont regroupées dans des classes.

Cependant UML dispose d'un concept « Le Meta modèle UML » pour déterminer l'acteur du système avec une ressemblance légère de diagramme de classe ce qui nous donne méta classe.

- Un acteur selon le concept méta classe (système informatisé) est :

Celui qui représente l'abstraction d'un rôle joué par des entités externes c'est-à-dire les utilisateurs, ou autre système etc., leur activité sur le système est d'avoir la possibilité de consulter et/ou de modifier directement du système.

- la représentation d'un acteur est sous forme rectangulaire avec le mot clé « Actor ».

Exemple :

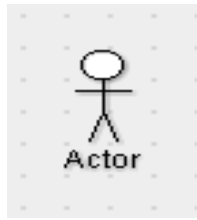


Figure II.1 : La forme d'un Acteur

I.1. Les utilisateurs de notre application:

Le logiciel doit demander au démarrage une identification de l'utilisateur pour assurer la confidentialité et l'intégrité des données.

Le médecin responsable doit pouvoir consulter et manipuler la liste des utilisateurs (Eux-mêmes) qui seront identifiés par un identifiant et un mot de passe.

I.2. La base de données pour notre application :

Une base de données peut être définie selon un seul type, c'est un ensemble de fichiers contenant des données organisées, qui doivent être sauvegardées, nettoyées, réorganisées, sécurisées etc. pour l'utilisateur, c'est un espace, lui permettant d'enregistrer des informations et de les retrouver quand il en a besoin.

Le futur système qui offre à son utilisateur plusieurs fonctionnalités dont la consultation, l'enregistrement, la modification ou même la suppression de données relatives à la gestion d'un cabinet médical.

II. Conception des schémas logiques et physique des données :

Cette partie consiste à transformer les objets du modèle de classes en relations.

Une relation comporte un nom (qui correspondra au nom de la table dans PHP MySQL), des attributs (qui correspondent aux propriétés conceptuelles ou organisationnelles et correspondent au niveau physique aux champs de la table). [3]

Durant la phase d'analyse, on a suivi comme processus de démarche orienté objet le processus de l'UML (**Unified Modeling Language**) comme langage de modélisation. [3]

Puisque nous allons implémenter notre base de données avec PHP MySQL, nous avons besoin de traduire certains objets pour qu'ils soient compatibles à un environnement relationnel, pour cela et pour traduire notre modèle on peut procéder à cette règle :

Règle: Toute classe d'entités du diagramme entité/association est représentée par une **relation** dans le schéma relationnel équivalent. La clé de cette relation est l'identifiant de la classe d'entités correspondante. [3]

Exemple :

Avec la transformation donc :

Patients
Matricule_Patient : Integer
Nom_Patient : Varchar
Prenom_Patient : Varchar
Date_Naissance : Date
Antecedents : Varchar
Adresse : varchar
Sexe : Varchar
Groupe_Sanguine : Varchar
Telephone : Integer
Ajouter()
Modifier()
Supprimer()

Patients (Matricule_Patient, Nom_Patients, Prenom_Patient, Date_Naissance, Lieu_Naissance, Adresse, Sexe, Groupe_Sanguine, Telephone)

- **Construction des schémas physique des données**

Classe		Méthode		
N°	Nom	Champs	Type	
1	Patients	<u>Matricule Patients</u> (*)	Numérique	AJOUTER () MODIFIER () SUPPRIMER ()
		Nom_Patient	Caractère	
		Prenom_Patient	Caractère	
		Date_Naissance	Date	
		Antecedent	Caractère	
		Adresse	Caractère	
		Sexe	Caractère	
		Groupe_Sanguine	Caractère	
		Telephone	Numérique	
2	Médecin	<u>Matricule Med</u> (*)	Numérique	AJOUTER () MODIFIER () SUPPRIMER ()
		Nom_Medecin	Caractère	
		Prenom_Medecin	Caractère	
		Photo_Medecin	Blob	
		Spécialité	Caractère	
		Adresse_Medecin	Caractère	
		Telephone_Medecin	Numérique	

3	Consultation	<u>Num Consu</u> (*)	Numérique	AJOUTER () MODIFIER () SUPPRIMER () IMPRIMER ()
		Nom_Patient	Date	
		Prenom_Patient	Caractère	
		Nom_Medecin	Caractère	
		Prenom_Medecin	Caractère	
		Taille	Caractère	
		Poid	Numérique	
		Température	Numérique	
		Tension	Numérique	
		Examen	Caractère	
		Conclusion	Caractère	
		Date	Date	
		Heure	Caractère	
4	Rendez-vous	<u>Matricule Patient</u> (*)	Numérique	AJOUTER () MODIFIER () SUPPRIMER ()
		Motif	Caractère	
		Nom_Patient	Caractère	
		Prenom_Patient	Caractère	
		Date_Rendez-vous	Date	
		Heure_Rendez-vous	Caractère	
5	Ordonnance	<u>Nom Medicament</u> (*)	Caractère	AJOUTER () MODIFIER () SUPPRIMER () IMPPRIMER ()
		Nom_Patient	Caractère	
		Prenom_Patient	Caractère	
		Date	Date	
		Heure	Caractère	
		Dosage_Medicament	Caractère	
		Forme_Medicament	Caractère	
		Pasologie_Medicament	Caractère	
		Quantité_Medicament	Numérique	
6	Facture	<u>Nom Produit</u> (*)	Caractère	MODIFIER () SUPPRIMER () IMPPRIMER ()
		Nombre	Numérique	
		Prix	Numérique	
		Prix-Total	Numérique	

III. Les diagrammes d'UML

Un diagramme UML est une représentation graphique qui s'intéresse à un aspect précis du modèle, il possède une structure et véhicule, une sémantique précise. Combinés les différents types de diagramme, UML offrent une vue complète des aspects statiques et dynamiques d'un système.

Une caractéristique importante des diagrammes UML est qu'ils supportent l'abstraction, cela permet de mieux contrôler la complexité dans l'expression et l'élaboration des solutions objet.[4]

- **L'utilisation de l'approche UML pour modéliser la gestion du cabinet médical :**

Dans cette partie du travail on essaye de modéliser le maximum de fragments du domaine de la gestion du cabinet médical.

Les diagrammes de cas d'utilisation, de séquences utilisés ne présentent que les cas d'utilisation et les interactions du système avec les différents acteurs, le traitement à l'intérieur du système et la structure interne du système seront décrit en détaille dans les autres diagrammes. [4]

III.1. Elaboration de modèle des cas d'utilisation :

Les modèles des cas d'utilisation permettent d'avoir une représentation de l'ensemble des fonctionnalités complètes du système.ces modèles comprend les acteurs, le système et les cas d'utilisation eux-mêmes. [4]

L'ensemble des fonctionnalités d'un système est déterminé en examinant les besoins fonctionnels de chaque acteur, exprimés sous forme de familles d'interactions dans les cas d'utilisation. [4]

Les acteurs se représentent sous la forme de petits personnages qui déclenche des cas d'utilisation, ces derniers sont représentés par des cercles par le système.

Cette section se compose de deux parties :

- **Les Diagrammes des cas d'utilisation.**
- **Leurs descriptions textuelles des cas d'utilisation.**

III.1.1. Diagramme des cas d'utilisation :

Rappelons que les cas d'utilisation servent à exprimer **les besoins fonctionnels** des utilisateurs d'un système.

D'autres types d'exigences peuvent être joints aux descriptions de cas d'utilisation, notamment les exigences non fonctionnelles qui ne sont pas prises en compte volontairement.

➤ Qu'est-ce qu'un cas d'utilisation ?

Un cas d'utilisation (*use case*) représente un ensemble de séquences d'actions réalisées par le système et produisant un résultat observable intéressant pour un acteur particulier, il modélise un service rendu par le système et qui concerne acteurs et ou système et apporte une valeur ajoutée « notable » à l'acteur concerné. [4]

Chaque cas d'utilisation spécifie un comportement attendu du système considéré comme un tout, sans imposer le modèle de réalisation de ce comportement.

Il permet de décrire ce que le futur système devra faire, sans spécifier comment il le fera, dans le cadre de la branche fonctionnelle, le cas d'utilisation doit mettre en valeur les interactions métier entre les acteurs et le système. [4]

➤ Dans ce chapitre « **Spécification du système** », pourquoi avons-nous recours aux cas d'utilisation ?

- Les cas d'utilisation permettent aux utilisateurs de structurer et d'articuler leurs désirs.
- Ils les obligent à définir de manière dont ils voudraient interagir avec le système.
- Ils favorisent la définition d'un [cahier](#) de charges qui reflète réellement les besoins même en absence d'un système à critiquer.

Nous allons procéder à l'automatisation du système informatique relatif à un cabinet médical.

Notre application aura comme principale fonctionnalités :

- Gestion et Suivi du Dossier Médical.
- Gestion des Rendez-vous.
- Gestion de la Fiche patient.
- Gestion de la Fiche médecin.
- Gestion de la Comptabilité.

Afin de détailler ces fonctionnalités, nous allons utiliser le diagramme de cas d'utilisation du langage de modélisation UML.

Nous allons procéder par les étapes suivantes :

- **Identifications des acteurs.**
- **Identification des cas d'utilisation.**
- **Diagramme de cas d'utilisation**

III.1.1.1. Identifications des acteurs :

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel, ou autre système) 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 le **Médecin** comme acteur principal.

III.1.1.2. Identification des cas d'utilisation :

Pour notre acteur, notre application doit donc offrir un ensemble de fonctionnalités, selon le tableau suivant :

<u>Utilisateur</u>	<u>Cas d'utilisation</u>
Médecin	<ul style="list-style-type: none">· Gestion et Suivi du Dossier Médical· Gestion des rendez-vous.· Gestion de la fiche patient.· Gestion de la fiche médecin.· Gestion de la comptabilité.

III.1.1.3. Diagramme de cas d'utilisation général de l'application

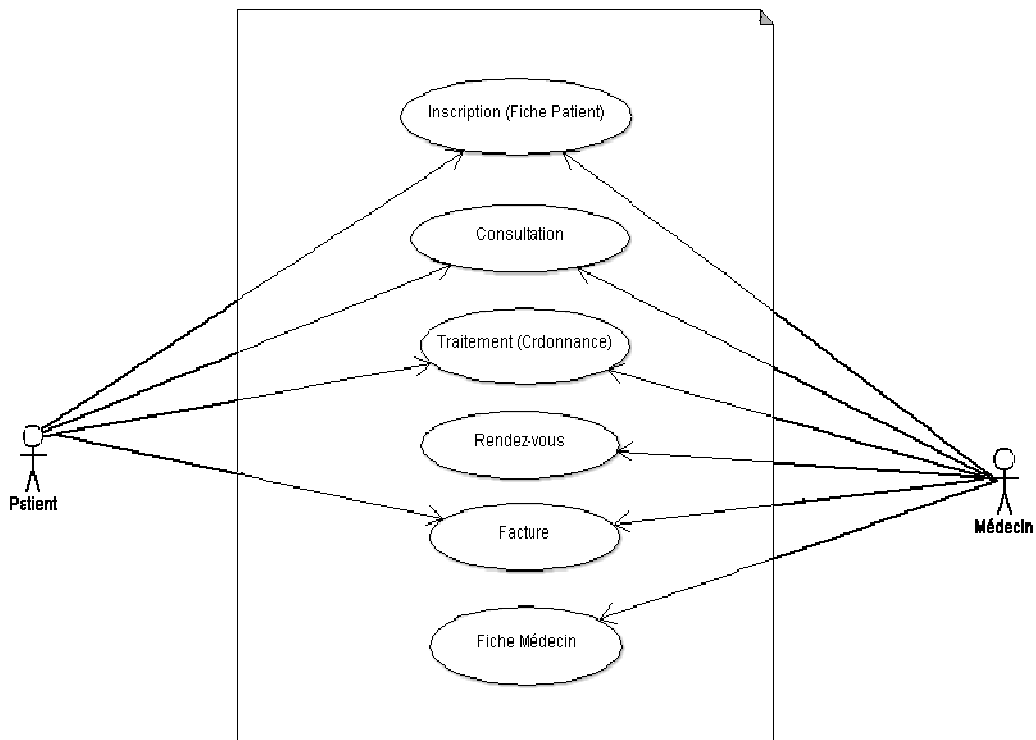


Figure II.2 : Diagramme de cas d'utilisation général de l'application

1. Cas d'utilisation pour la gestion de la fiche patient :

A l'arrivée d'un patient au cabinet, le médecin prend en charge :

- S'il s'agit d'un nouveau patient, une nouvelle fiche sera créée qui comporte toutes les informations nécessaires c'est-à-dire (numéro fiche patient, nom, prénom, date de naissance, lieu de naissance, adresse, sexe, groupe sanguin, numéro de téléphone, etc....).
- S'il s'agit d'un ancien patient, le médecin consulte sa fiche médicale avec une recherche du nom patient et peut modifier quelques informations si c'est nécessaire.

2. Cas d'utilisation pour la Consultation :

La gestion des consultations et des dossiers médicaux s'effectue par le médecin et elle est constituée des informations secrètes et confidentielles du patient, elle englobe les différentes fonctionnalités suivantes : **gestion des consultations, gestion des examens, gestion du motif etc.....** A chaque consultation un résumé de l'observation nouvelle et du traitement institué sera porté sur le dossier, c'est-à-dire un enregistrement pour toutes les informations relatives à un patient

3. Cas d'utilisation des ordonnances :

La gestion des ordonnances s'effectue par le médecin, après la consultation, l'examen clinique et le diagnostic le médecin donne un traitement au malade

4. Cas d'utilisation pour la gestion des rendez-vous :

Ce cas d'utilisation contient les opérations suivantes :

- Afficher la liste des rendez-vous
- Ajouter un rendez-vous
- Modifier rendez-vous
- Supprimer un rendez-vous

5. Cas d'utilisation pour la gestion de la comptabilité :

La gestion de la comptabilité est effectuée par le médecin, en effet, elle présente l'ensemble des dépenses et des recettes.

III.1.2. Description textuelle des cas d'utilisation :

Dans cette section, chaque cas d'utilisation sera décrit de façon exhaustive suivant le format présenté dans les cas d'utilisation précédents.

La description textuelle, pour faire quoi ?

Pour la documentation des cas d'utilisation (les scénarios), et aussi la description textuelle est indispensable, car elle seule permet de communiquer facilement et précisément avec les utilisateurs. Elle est l'occasion d'identifier le contexte d'exécution de l'un ou de l'autre des enchaînements.

Cette partie est composée de deux :

- **Sommaire d'identification** :(titre, but, résumé, acteur).
- **Description de l'enchaînement** :(pré-condition, post-condition, scénario nominale, scénario alternative).
- **Titre** : Cas d'utilisation concerné.
- **But** : l'objectif de ce cas d'utilisation dans le système.
- **Résumé** : c'est le résumé du contenu textuel
- **Pré-condition** : se sont les conditions nécessaire pour déclencher les enchaînements.
- **Post-condition** : représente l'événement futur.
- **Scénario nominale** : représente les événements produits par l'acteur et le système de la façon sans échec (sans erreur).
- **Scénario alternative** : représente les événements après les erreurs produits par l'acteur et le système.

1. Description textuelle pour cas d'utilisation gestion de la Fiche patient :

Sommaire d'identification
<p><u>Titre</u> : Gestion de la fiche patient.</p> <p><u>But</u> : pour avoir les informations des patients.</p> <p><u>Résumé</u> : le médecin établie une fiche patiente, s'il s'agit d'un nouveau patient, sinon il fera la mise à jour nécessaire. (l'ajout, affichage, modification, suppression)</p> <p><u>Acteurs</u> : médecin.</p>
Description de l'enchaînement
<p><u>Pré condition</u> : Accès autorisé</p> <p><u>Post condition</u>: une nouvelle fiche patient sera mise à jour.</p> <p><u>Scénario nominal</u> :</p> <ol style="list-style-type: none">1- Le médecin saisit le login et le mot de passe.2- Le système vérifie le login et le mot de passe.3- Le système affiche le menu.4- Le médecin choisit « Gestion fiche patient ».5- Le système affiche un formulaire.6- Le médecin saisit les informations relatives au patient.7- Le système effectue un contrôle sur les champs saisis.8- Le système vérifie que tous les champs obligatoires sont complets.9- Le système enregistre les informations relatives au patient.10- Le système affiche un message de confirmation. <p><u>Scénario alternatif</u> :</p> <p>1' : Erreur d'identification.</p> <ol style="list-style-type: none">2. Le système affiche une erreur d'identification. <p>Le scénario reprend au point 1.</p> <p>2' : nature des champs saisie incorrecte.</p>

L'enchaînement démarre au point 7.

7. Le système signale une erreur des champs saisis.

Le scénario nominal reprend au point 5.

3' : Champs obligatoires vides.

L'enchaînement démarre au point 8

8. Le système signale l'existence des champs obligatoires vide.

10. Le système réaffiche le formulaire déjà remplis.

Le scénario reprend au point 6.

2. Description textuelle pour cas d'utilisation gestion des Rendez-vous :

Sommaire d'identification

Titre : Gestion des rendez-vous.

But : pour organiser la consultation

Résumé : un patient arrive le médecin prend le rendez-vous. (l'ajout, affichage, modification, suppression)

Acteurs : médecin.

Description de l'enchaînement

Pré condition : Affecter un RDV.

Accès autorisé

Post condition: un nouveau rendez-vous sera enregistré

Scénario nominal :

1- Le médecin saisit le login et le mot de passe.

2- Le système vérifie le login et le mot de passe.

3- Le système affiche le menu du médecin.

4- Le médecin choisit « **Gestion des rendez-vous** ».

5- Le système affiche un formulaire

- 6- Le médecin saisit la date d'un rendez-vous et l'heure voulu.
- 7- Le système affiche la liste des rendez-vous.
- 8- Le médecin saisit un rendez-vous ayant une heure différente à celui des autres rendez-vous.
- 9- Le système vérifie que tous les champs obligatoires sont complets.
- 10- Le système enregistre les informations du nouveau rendez-vous.
- 11- Le système affiche un message de confirmation.

Scénario alternatif

1' : Erreur d'identification.

2. Le système affiche une erreur d'identification.

Le scénario reprend au point 1

2' : champ obligatoire vide.

L'enchaînement démarre au point 9

10. Le système signale l'existence des champs obligatoire vides.

11. Le système réaffiche le formulaire déjà remplis

Le scénario reprend au point 8

3. Description textuelle pour cas d'utilisation gestion des consultations :

Sommaire d'identification
<u>Titre</u> : Gestion des consultations /ou dossiers médicaux.
<u>But</u> : Permette au médecin de gérer les consultations et dossiers médicaux.
<u>Résumé</u> : Gérer une fiche consultation. (Modification, Affichage, l'Ajout, Suppression)
<u>Acteurs</u> : Médecin.
Description de l'enchaînement
<u>Pré condition</u> : Présence d'un patient
Accès autorisé
<u>Post condition</u> : une nouvelle consultation sera enregistrée.

Scénario nominal :

- 1- Le médecin saisit le login et le mot de passe.
- 2- Le système vérifie le login et le mot de passe.
- 3- Le médecin choisit « **gestion des patients /ou gestion des rendez-vous** »
- 4- Le système affiche la liste des patients /ou rendez-vous
- 5- Le médecin choisit le patient.
- 6- Le médecin choisit «**visite médical**».
- 7- Le système affiche un formulaire
- 8- Le système affiche le menu du médecin.
- 9- Le médecin saisit les informations.
- 10- Le système effectue un contrôle sur les champs obligatoires.
- 11- système effectue un contrôle sur les champs saisis.
- 12- Le système vérifie que tous les champs obligatoires sont complets.
- 13- Le système enregistre les informations.
- 14- Le système affiche un message de confirmation.

Scénario alternatif

1' : Erreur d'identification.

Le système affiche une erreur d'identification.

Le scénario reprend au point 1

2' : nature des champs saisie incorrecte.

L'enchaînement démarre au point 12.

12. Le système signale une erreur des champs saisis.

3' : Champs obligatoires vides.

L'enchaînement démarre au point 12

12. Le système signale l'existence des champs obligatoires vide.

Le système réaffiche le formulaire déjà remplis si le numéro de visite et déjà utiliser.

Le scénario reprend au point 1.

4. Description textuelle pour cas d'utilisation gestion des Ordonnances :

Sommaire d'identification

Titre : Gestion des Ordonnances.

But : Permette au médecin de gérer les Ordonnances

Résumé : Etablir une fiche d'ordonnance. (Modification, Affichage, l'Ajout, Suppression)

Acteurs : Médecin.

Description de l'enchaînement

Pré condition : Présence d'un patient

Accès autorisé

Post condition: une nouvelle ordonnance sera établie.

Scénario nominal :

1- Après l'enregistrement des informations de la consultation.

2- Le système passe directement au «**Gestion des Ordonnances**».

3- Le système affiche un formulaire qui comporte le nom et le prénom du patient / médecin

4- Le médecin saisit les informations.

5- Le système effectue un contrôle sur les champs obligatoires.

6- système effectue un contrôle sur les champs saisis.

7- Le système vérifie que tous les champs obligatoires sont complets.

8- Le système imprime les informations.

11- Le système affiche un message de confirmation.

Scénario alternatif

1' : Erreur d'identification.

Le système affiche une erreur d'identification.

Le scénario reprend au point 1

2' : nature des champs saisie incorrecte.

L'enchaînement démarre au point 4.

6. Le système signale une erreur des champs saisis.

3' : Champs obligatoires vides.

L'enchaînement démarre au point 4

7. Le système signale l'existence des champs obligatoires vide.

Le scénario reprend au point 4.

5. Description textuelle pour cas d'utilisation gestion des Recettes :

Sommaire d'identification

Titre : Gestion des recettes.

But : pour contrôler le flux d'entrés.

Résumé : permettre d'enregistrer toutes les recettes du cabinet.

Acteurs : médecin.

Description de l'enchaînement

Pré condition : Accès autorisé

Post condition: toutes les recettes seront enregistrées.

Scénario nominal :

1- Après l'impression d'ordonnance.

2- Le système passe directement au «**gestion des Recettes**».

3- Le système affiche un formulaire qui comporte le nom et le prénom du patient / médecin

4- Le médecin saisit les informations.

5- Le système effectue un contrôle sur les champs obligatoires.

6- système effectue un contrôle sur les champs saisis.

7- Le système vérifie que tous les champs obligatoires sont complets.

8- Le système imprime les informations.

11- Le système affiche un message de confirmation.

Scénario alternatif :

1' : Erreur d'identification.

Le système affiche une erreur d'identification.

Le scénario reprend au point 1

2' : nature des champs saisie incorrecte.

L'enchaînement démarre au point 7.

7. Le système signale une erreur des champs saisis.

3' : Champs obligatoires vides.

L'enchaînement démarre au point 8

8. Le système signale l'existence des champs obligatoires vide.

10. Le système réaffiche le formulaire déjà remplis.

Le scénario reprend au point 5.

6. Description textuelle pour cas d'utilisation gestion de la fiche médecin :

Sommaire d'identification

Titre : Gestion des fiches médecins.

But : pour avoir les informations des médecines.

Résumé : le médecin (responsable) établie une fiche médecin, il fera la mise à jour nécessaire. (l'ajout, affichage, modification, suppression)

Acteurs : médecin.

Description de l'enchaînement

Pré condition : Accès autorisé

Post condition: une nouvelle fiche patient sera mise à jour.

Scénario nominal :

- 1- Le médecin saisit le login et le mot de passe.
- 2- Le système vérifie le login et le mot de passe.
- 3- Le système affiche le menu.
- 4- Le médecin choisit « **Gestion fiche médecins** ».
- 5- Le système affiche un formulaire.
- 6- Le médecin (responsable) saisit les informations relatives au médecin.
- 7- Le système effectue un contrôle sur les champs saisis.
- 8- Le système vérifie que tous les champs obligatoires sont complets.
- 9- Le système enregistre les informations relatives au patient.
- 10- Le système affiche un message de confirmation.

Scénario alternatif :

1' : Erreur d'identification.

2. Le système affiche une erreur d'identification.

Le scénario reprend au point 1.

2' : nature des champs saisie incorrecte.

L'enchaînement démarre au point 7.

7. Le système signale une erreur des champs saisis.

Le scénario nominal reprend au point 5.

3' : Champs obligatoires vides.

L'enchaînement démarre au point 8

8. Le système signale l'existence des champs obligatoires vide.

10. Le système réaffiche le formulaire déjà remplis.

Le scénario reprend au point 6.

III.2. Diagramme de séquence :

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation d'UML, le diagramme de séquence permet de cacher les interactions d'objets dans le cadre d'un scénario d'un Diagramme des cas d'utilisation. [5]

Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système. Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets. [5]

1. Diagramme de séquence de la fiche patient :

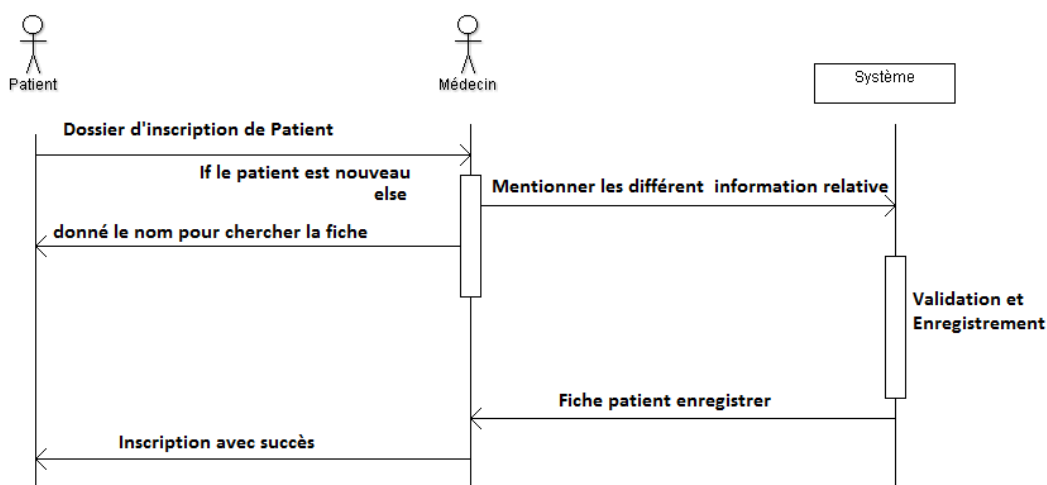


Figure II.3 : diagramme de séquence du fiche patient

- **Description textuelle :**

Le patient donne a le médecin les informations de leur état civile, si le patient est nouveau le médecin mentionner les différentes informations relative a ce patient, le système enregistrer ses informations, si le patient est ancien le patient donne que son nom au médecin.

2. Diagramme de séquence de la consultation :

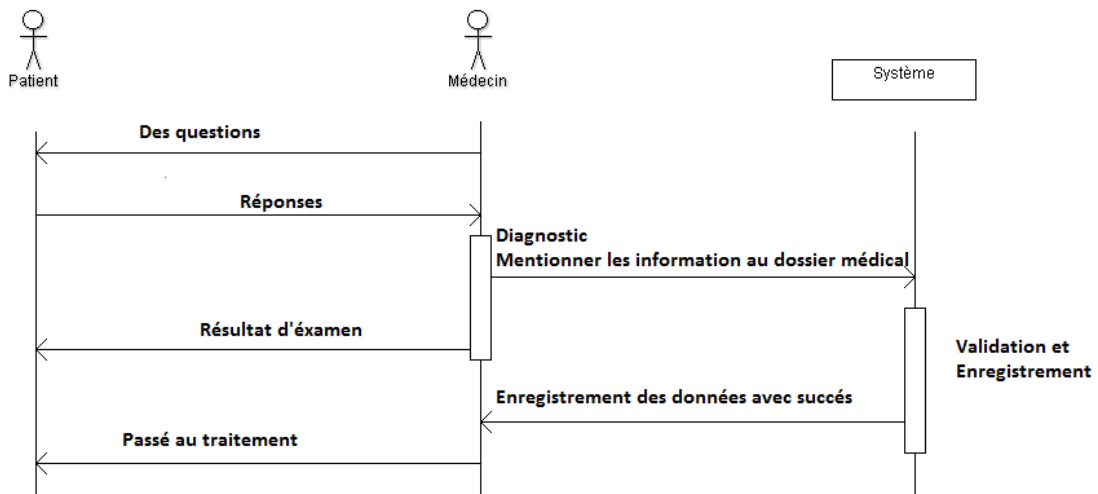


Figure II.4 : diagramme de séquence de la consultation

- **Description textuelle :**

Le médecin pose des questions au patient sur leur douleur, le patient répond au médecin, ce dernier fait des examens sur le patient, après le diagnostic le médecin mentionne les informations au dossier médical de patient, le système enregistre ses informations.

3. Diagramme de séquence des ordonnances :

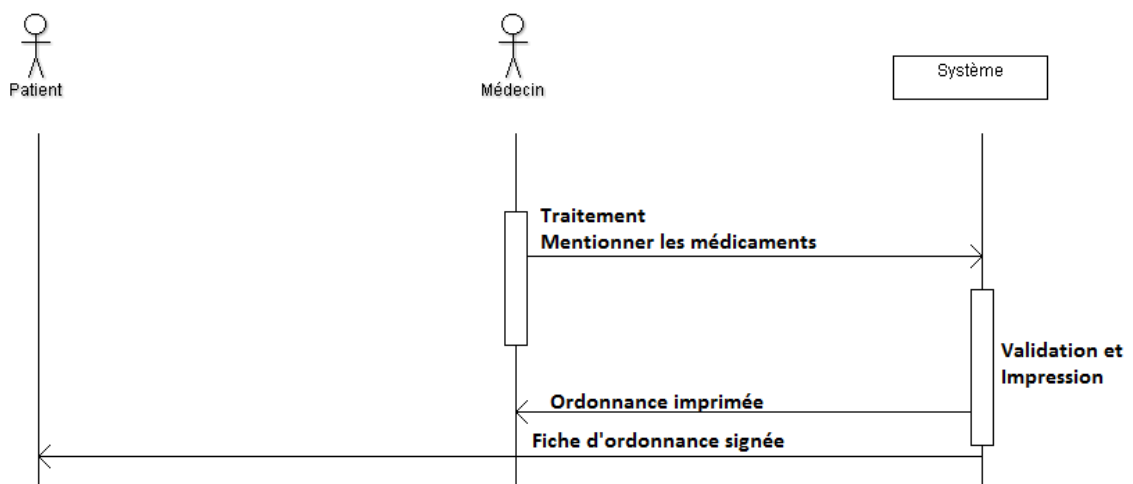


Figure II.5 : diagramme de séquence des ordonnances

- **Description textuelle :**

Le médecin mentionner les médicaments, le système valider et imprimer l'ordonnance au patient, le médecin signée la fiche d'ordonnance et lui a donné au patient.

4. Diagramme de séquence de la Facture :

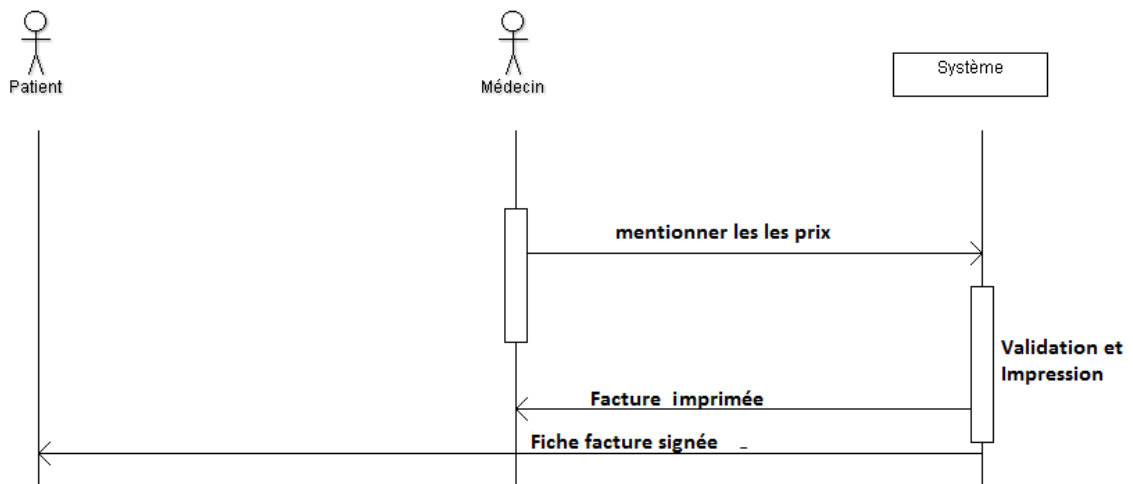


Figure II.6 : diagramme de séquence des Factures

- **Description textuelle :**

Le médecin mentionner les prix du produit utiliser comme Sérum pour un traitement médical, aiguille médical, etc..., et le prix se la visite, le système valider et imprimer la facture au patient, le médecin signée la fiche de facture et lui a donné au patient.

III.3. Construction du diagramme de classes :

Les diagrammes de classes expriment de manière générale la structure statique d'un système, en termes de classes et de relations entre ces classes. [5]

Une classe est composée de nom de la classe, des attributs, permet de décrire un ensemble des objets (attributs et comportement) opérations, le diagramme de classe est un modèle permettant de décrire de manière abstraite et générale les liens entre objets. [5]

Dans cette partie, nous allons présenter :

- La liste des supports d'information
- Conception des classes

III.3.1. La liste des supports d'information :

Support	Informations
Dossier médical	<ul style="list-style-type: none">-Numéro de consultation. -Nom patient -Prénom patient -Nom médecin -Prénom médecin - Motif pour le patient. - Examens complémentaires pour le patient (taille, poids, etc.) - Diagnostic après la consultation. -conclusion des examens -Date visite de patient -Heure visite de patient
Fiche patients	<ul style="list-style-type: none">- Numéro de fiche patient. - Nom de patient. - Prénom de patient - Date de naissance de patient - Antécédents - Adresse de patient - Sexe de patient -Groupe Sanguine de patient - Téléphone de patient
Fiche médecins	<ul style="list-style-type: none">- Numéro de fiche médecin.

	<ul style="list-style-type: none"> - Nom de médecin. - Prénom de patient - photo de médecin - Spécialité - Adresse de médecin - Téléphone de médecin
Rendez-vous	<ul style="list-style-type: none"> - Numéro de fiche patient. - Motif - Nom de patient. - Prénom de patient - Date rendez-vous - Heure rendez-vous
Ordonnance Médicale	<ul style="list-style-type: none"> -Nom patient -Prénom patient -Nom médecin -Prénom médecin - Date ordonnance -Heur ordonnance - Nom de médicament - Dosage de médicament - Forme de médicament - Nombre de fois par jour. - Quantité prise

Facture	-Nom du produit -Nombre du produit -Prix du produit -Prix total des produits
----------------	---

➤ **Description des classes :**

Après une analyse de l'existant, nous avons dégagé les classes nécessaires pour une bonne gestion du cabinet médical.

Les classes sont :

N°	Nom des classes
1	Patients
2	RDV
3	Consultation
4	Ordonnance
5	Facture
6	Médecins

➤ **Description des associations**

Numéro	Association	Classes participantes	Description
01	Avoir rendez-vous	Patient, rendez-vous	Un patient peut avoir un RDV à chaque consultation
02	Consulte	Patient, consultation	Un patient peut effectuer une ou plusieurs consultations
03	Avoir ordonnance	Consultation, ordonnance	Une consultation produire une seule ordonnance
04	Avoir nouveau rendez-vous après la consultation	RDV, consultation	Un RDV est fixé pour une consultation donnée
05	Réglée la caisse	CAISSE, consultation	Une consultation est réglée à travers une caisse

III.3.2. Conception des classes :

C'est une collection d'éléments de modèle statique, tels que des classes, des interfaces et leurs relations, connectés entre eux comme un graphe, il représente la description statique du système en intégrant dans chaque classe la partie dédiée aux données et celle consacrée aux traitements. [5]

C'est le diagramme pivot de l'ensemble de la modélisation d'un système.

- **Identification des classes :**

Une classe est une description d'un groupe d'objets partageant un ensemble commun de propriétés (les attributs), de comportements (les opérations) et de relations avec d'autres objets (les associations et les agrégations). [5]

- **Une classe contient :**

Des attributs (ou champs, ou variables d'instances) : Les attributs d'une classe décrivent la structure de ses instances (les objets).

Des méthodes (ou opérations de la classe) : les méthodes décrivent les opérations qui sont applicables aux instances de la classe.

- **Une agrégation :**

Est une association correspondant à une relation qui lorsqu'elle est lue dans un sens signifie "est une partie de" et lorsqu'elle est lue dans l'autre sens elle signifie "est composé de".

- **Diagramme de classe**

Le diagramme de classe exprime d'une manière générale la structure statique d'un système, en termes de classe et de relation entre ces classes.

Voici le diagramme de classes de notre système :

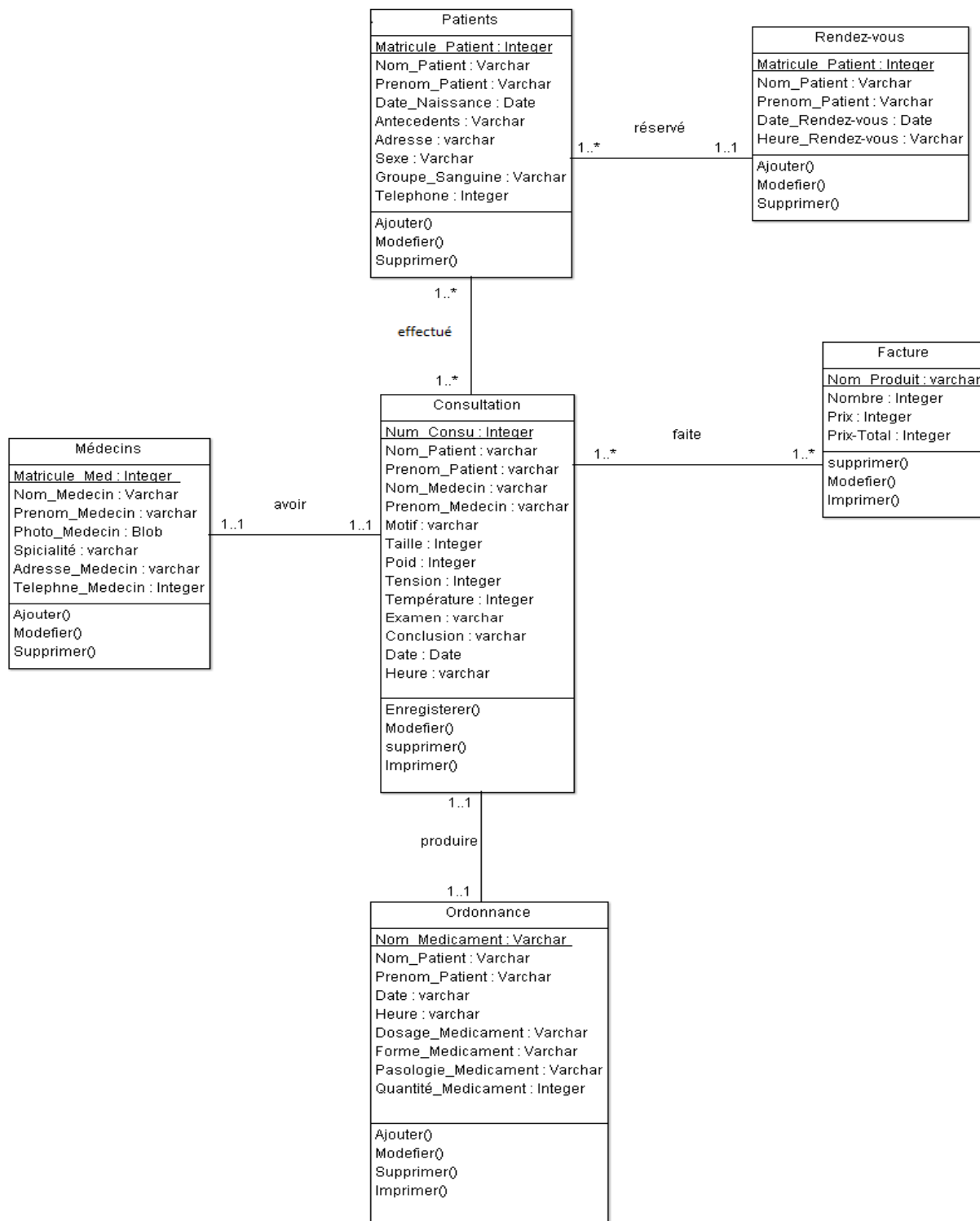


Figure II.7 : Diagramme de classes

IV. Conclusion :

La capture des besoins est la partie pour comprendre les besoins permettant d'identifier et de décrire les *fonctionnalités* d'un logiciel qui sont significatives pour ses utilisateurs (humains, matériels, logiciels) permettant de décrire les interactions du logiciel.

En outre *techniquement*, elle nous a donné plus d'importance à la confidentialité et la sécurité des données en se servant des mots de passe aux données, et avec un maximum de contrôle au moment de la saisie des données et la sécurité de l'information par la définition des règles de contrôle.

Durant ce chapitre on a vu les différents diagrammes d'UML avec leur utilisation sur notre futur application.

Durant la modélisation, il n'est pas nécessaire d'utiliser tous les diagrammes, le diagramme le plus important est le diagramme de classe, le choix des diagrammes dépend des contraintes imposées au développeur.

L'étude d'Analyse de notre application, elle est la partie très importante pour bien maîtriser le sujet à étudier dans ce projet pour cela, on a bien détaillé le diagramme de classes.

Toutefois, rien ne vaut que la pratique et pour y arriver, des étapes de développement sont à suivre. C'est justement, le but de chapitre suivant qui est consacré à l'implémentation et réalisation de notre application et présentés ces différents outils et technologies.

Chapitre III

Introduction :

Ce présent chapitre est de présenter les outils et langage utilisé et une Algorithme générale du projet. Pour bien mener notre application nous allons présenter une représentation graphique de la navigation des différentes interfaces de notre projet Gestion Cabinet Médical.

I. Outils et langage utilisé :

I.1. Environnement de réalisation :

Dans cette partie, nous allons présenter :

- L'environnement matériel
- L'environnement logiciel

I.1.1. L'environnement matériel :

Au cours de la réalisation de cette application, nous avons utilisé le matériel suivant :

	Unité	Caractéristiques
Un micro-ordinateur	Processeur	Intel® Pentium® DUAL CPU
	Mémoire RAM	2040 Mo (2 GB)
	Disque dur	500 Go
	Ecran	15"

I.1.2. L'environnement logiciel :

La réalisation de l'application a été développée avec les outils suivants :

1. NetBeans7.4 :

Est un [environnement de développement intégré](#) (EDI), en plus de Java, NetBeans7.4 permet également de supporter différents autres langages, comme C, C++, JavaScript, XML, [PHP](#) et [HTML](#) de façon native ainsi que bien d'autres (comme [Python](#) ou [Ruby](#)) par l'ajout de greffons, il comprend toutes les caractéristiques d'un IDE moderne ([éditeur en couleur](#), projets [multi-langage](#), [refacturent](#), éditeur graphique d'interfaces et de pages Web). [6]

Un environnement Java développement Kit [JDK](#) est requis pour les développements en Java.

NetBeans7.4 constitue par ailleurs une plate forme qui permet le développement d'applications spécifiques (bibliothèque [Swing \(Java\)](#)).

L'[IDE](#) NetBeans7.4 s'appuie sur cette plate forme. [6]

- **Plate forme Netbeans7.4 :**

NetBeans7.4 est aussi une plate-forme générique pour le développement d'applications pour stations de travail (bibliothèque [Swing \(Java\)](#)). [6]

Elle fournit des ressources pour développer les éléments structurants de ces applications: gestion des menus, des fenêtres, configuration, gestion des fichiers, gestion des mises à jour... des présentations détaillées sont fournies par le centre de documentation de NetBeans7.4. [6]

- **Le langage Java :**

Est un [langage de programmation informatique orienté objet](#).

La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement [portables](#) sur plusieurs [systèmes d'exploitation](#) tels que UNIX, Windows, [Mac OS](#) ou [GNU/Linux](#), avec peu ou pas de modifications. [7]

Pour cela, divers [plateformes](#) et [Framework](#) associés visent à guider, sinon garantir, cette portabilité des [applications](#) développées en Java. [7]

2. PHP MySQL:

MySQL est un [système de gestion de bases de données](#) relationnelles (SGBDR).

Il fait partie des logiciels de gestion de [base de données](#) les plus utilisés au monde¹, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec [Oracle](#), et [PHP](#) MySQL.

MySQL est capable, depuis la version 3.23.15 d'utiliser un système de réplication à sens unique relativement simple à mettre en place : un serveur maître qui contient la base de données source et va recevoir l'ensemble des modifications, et un serveur esclave qui contient une base à structure identique et qui va se connecter au serveur maître pour aller chercher les nouvelles modifications et mettre à jour ses bases.

Un maître peut avoir plusieurs esclaves, et un esclave peut devenir à son tour maître pour un autre serveur esclave, réalisant ainsi une chaîne de réplication. [8]

Toutes les modifications (UPDATE, DELETE, INSERT...) doivent être envoyées au serveur maître et seront automatiquement répercutées sur ses esclaves, mais les requêtes de recherche (SELECT), souvent plus nombreuses et plus gourmandes en ressources pourront être effectuées sur un ou plusieurs esclaves.

Comme les requêtes de modifications doivent être redirigées sur le serveur maître, la conception de l'application doit s'adapter à l'architecture : le programmeur va ouvrir deux connexions dans son programme, une sur le serveur maître, sur lequel il exécutera toutes ses modifications et une sur le serveur esclave sur lequel il exécutera toutes ses recherches. [8]

- **Système d'exploitation** : Microsoft Windows 7.
- **Microsoft WORD 2007** : pour le traitement de texte.
- **Microsoft Power Point 2007** : pour la représentation du projet.

3. ArgoUML :

ArgoUML est un [logiciel libre](#) de création de diagrammes [UML](#).

ArgoUML supporte sept types de diagramme : [cas d'utilisation](#), [classes](#), [séquence](#), [état](#), [activité](#), [collaboration](#), et [déploiement](#). [9]

I. Algorithme UML du projet :

A l'arrivée du patient jusqu'à la sortie du cabinet

Lorsqu'un client arrive la première fois chez un cabinet médical, le médecin crée une fiche contenant les différentes informations relatives à ce patient (nom, prénom, date de naissance, lieu de naissance, adresse, sexe, groupe sanguin, téléphone...)

Si le patient est déjà client du cabinet médical, le médecin lui demande son nom pour chercher sa fiche, il peut aussi faire une recherche sur leur dossier médical pour continuer le traitement si le patient a un rendez-vous de traitement ou le médecin faire une visite médicale / ou une nouvelle consultation au patient.

Après l'examen clinique le médecin lui donne un traitement ou ordonnance d'aller acheter. il peut aussi donner un nouveau rendez-vous si le traitement n'est pas terminé.

A la fin de la visite le médecin donne une facture à régler au malade.

Voila l'algorithmme du projet :

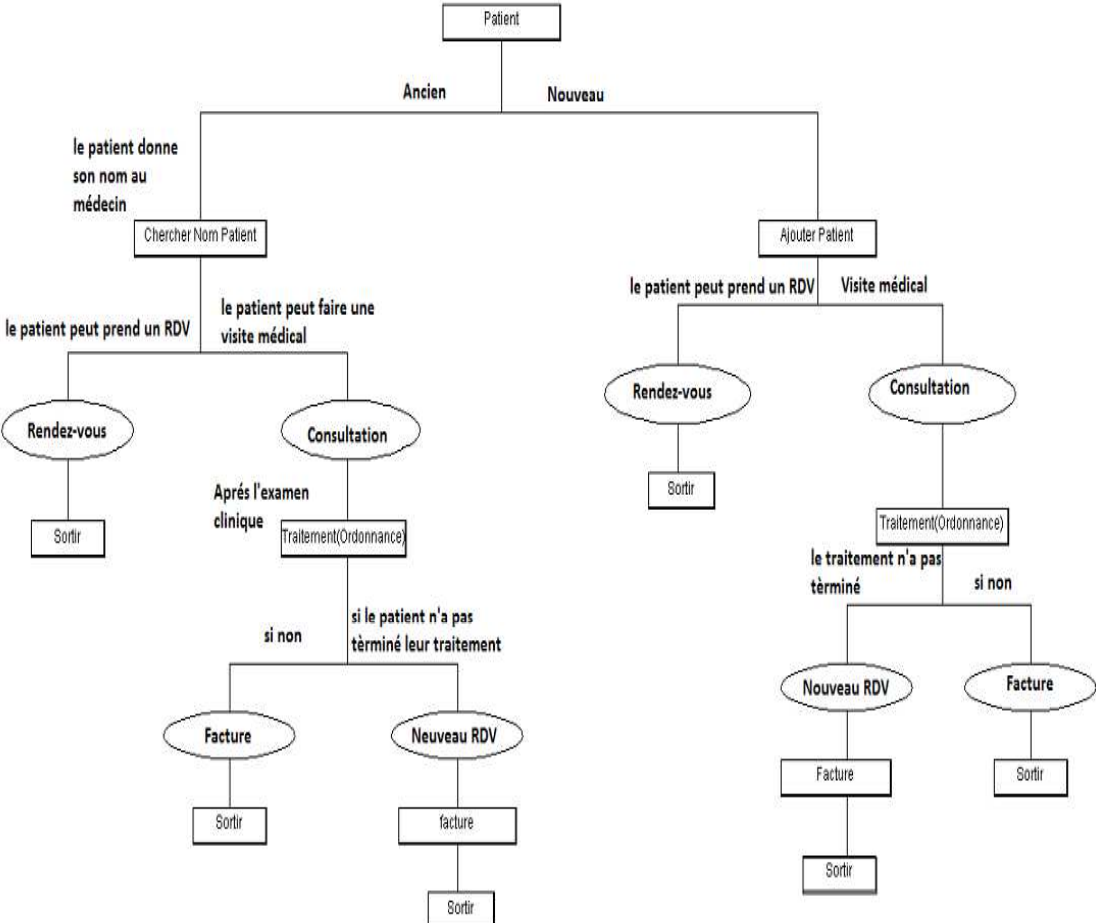


Figure III.1 : l'algorithmme du projet

III.2. Réalisation du projet :

Notre application de gestion d'un cabinet médical sur mesure permet de gérer les consultations, les rendez-vous, le dossier médical, etc.... et d'offrir à l'utilisateur quelques accessoires à savoir la l'heure et la date actuelle.

La multitude des taches que notre application est capable de faire engendrer un grand nombre de fenêtres.

On va essayer de sélectionner quelques fenêtres qui nous paraissent importantes pour les intégrer dans le présent mémoire.

1. Interface authentification

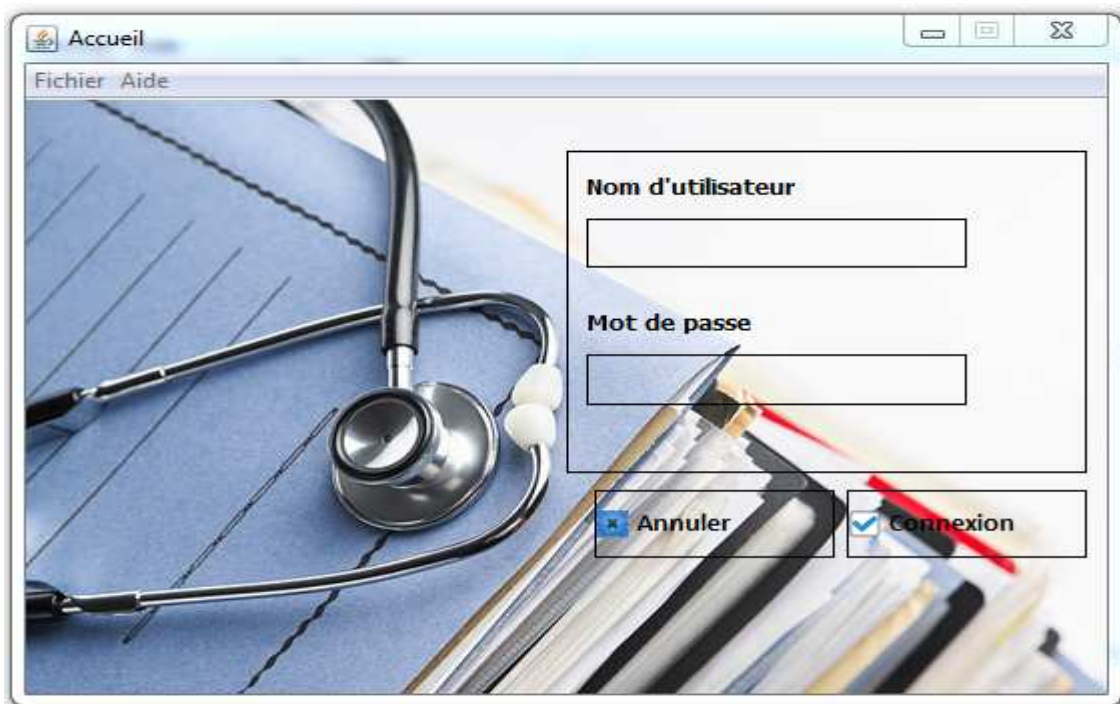


Figure III.2 : Fenêtre d authentification

- **Description**

C'est la première fenêtre qui s'affiche si on exécute l'application, toute personne qui veut bénéficier des services du logiciel doit s'authentifier avec un login et mot de passe.

Cette page comporte aussi deux boutons dont le premier est « Connexion » qui permet l'accès à la fenêtre principale si le login et le mot de passe sont vrais.

Si ces données sont fausses un message d'erreur s'affiche.

Le deuxième bouton est « Annuler » pour annuler l'accès et quitter.

2. Menu principale :



Figure III.3 : Menu pour médecin

- *Description*

Dans le cas où la connexion se fait par le médecin, l'accès est donné toutes les fonctionnalités du système : gestion des rendez-vous, gestion des fiches du patient, gestion des fiches du médecin et gestion du dossier médical. Il peut accéder à n'importe quelles tâches.

3. Gestion de fiches médecins :



Figure III.4 : Interface gestion de fiches médecins

- **Description :**

Pour enregistrer les renseignements personnels pour le médecin comme le matricule, le nom, le prénom, spécialité, la photo, numéro téléphone.

4. Gestion du fiche patients :



Figure III.5 : Interface gestion du fiche patients

- **Description :**

A l'arrivée d'un nouveau patient le médecin remplit les informations relative du patient sur une fiche contiens le matricule, le nom, prénom, adresse, date de naissance, les antécédents, téléphone, sexe, groupe sanguine.

Si est un ancien patient le médecin demander au patient son nom pour chercher leur fiche médical pour passé à la consultation.

5. Gestion des rendez-vous :

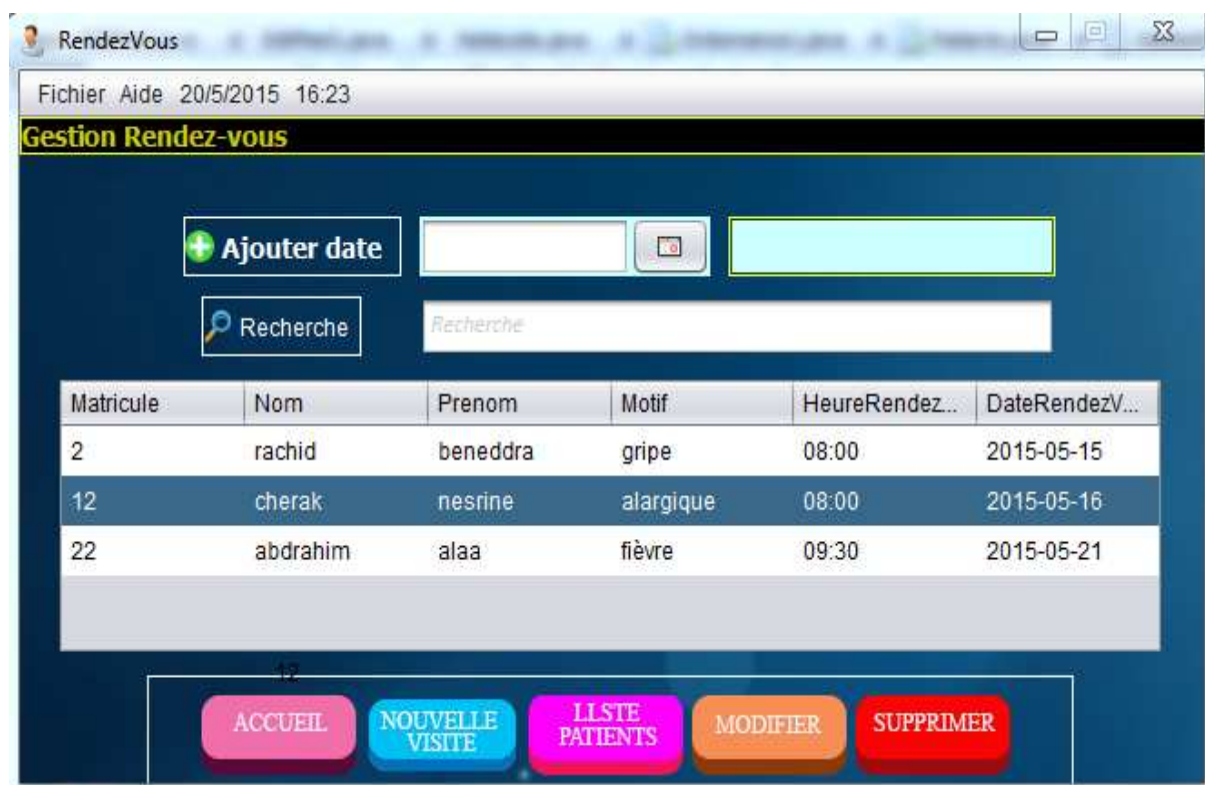


Figure III.6 : Interface gestion des rendez-vous

- **Description**

La gestion des rendez-vous est une tâche essentielle, celle-ci vérifie la disponibilité de la date demandée et par la suite elle ajoute un rendez-vous en saisissant les renseignements nécessaires.

On a deux modes de recherche sur une date réservé : par nom, et par date.

6. Gestion des rendez-vous « Nouveau RDV » :

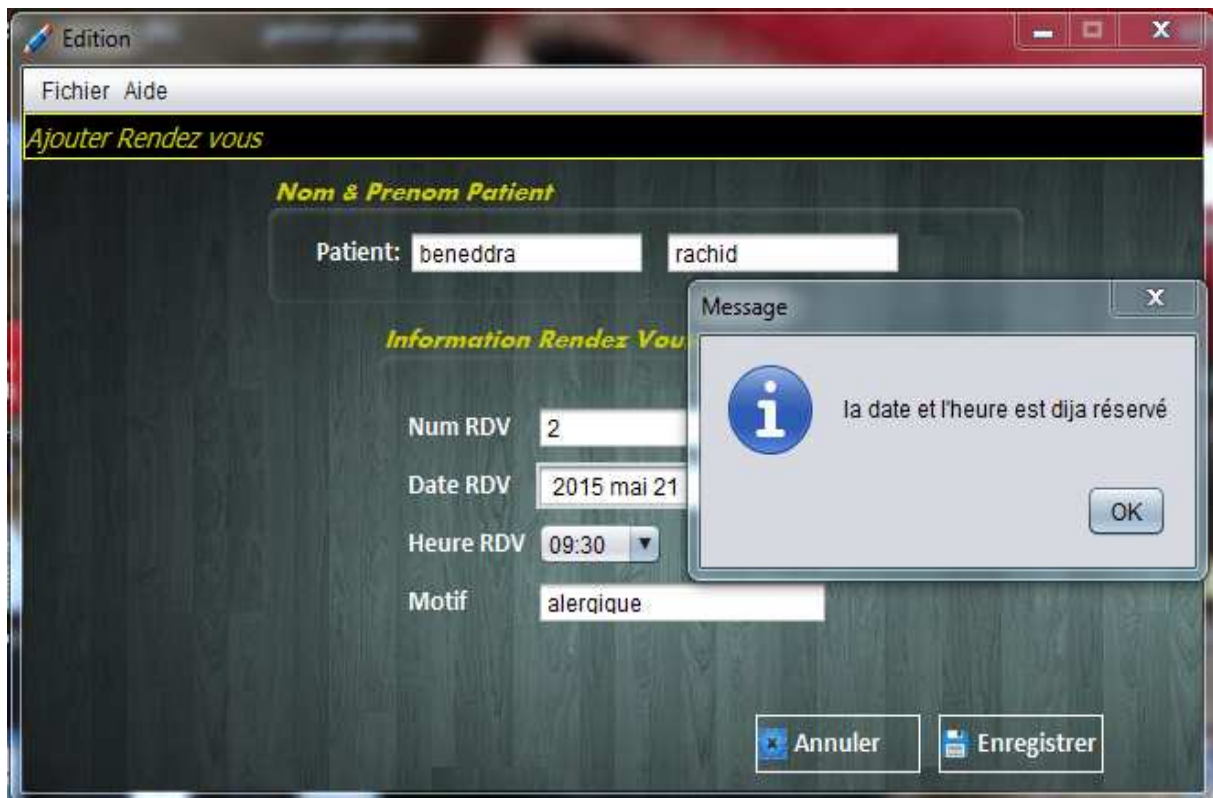


Figure III.7 : Interface gestion des rendez-vous « Nouveau RDV »

- **Description**

Une exception a été déclenchée suite à une opération d'ajout d'un Rendez vous, message d'erreur pour la 'Date et l'heure' qui sont déjà réservées par un patient.

C'est un exemple des messages qui peuvent rencontrer l'administrateur lors de l'utilisation de notre logiciel. Ces messages empêchent les erreurs de saisie, et guident l'utilisateur pour comprendre rapidement le fonctionnement de l'application.

7. Gestion et suivi du dossier médical :

Figure III.8 : Interface pour la page de consultation

num...	No...	Pren...	No...	Pren...	Motif	taille	poids	tens...	tem...	peri...	Exa...	Con...	Date	Heure
22	ben...	rachid	daho	bilal	fatig...	171	66	9	38	12	fible...	doul...	20/5...	15:46
23	ben...	rachid	daho	bilal	alèr...	171	66	10	38	12	fièvr...	grip...	20/5...	16:35
115	cher...	nesr...	daho	bilal	gripe	166	59	12	38	13	fièvre	doul...	9/5/...	17:55

Figure III.9 : Interface Pour la Gestion et suivi du dossier médical

- *Description*

La gestion et suivie du dossier médical qui contient des informations sur le patient et qui facilite la consultation au médecin.

8. Gestion des ordonnances :

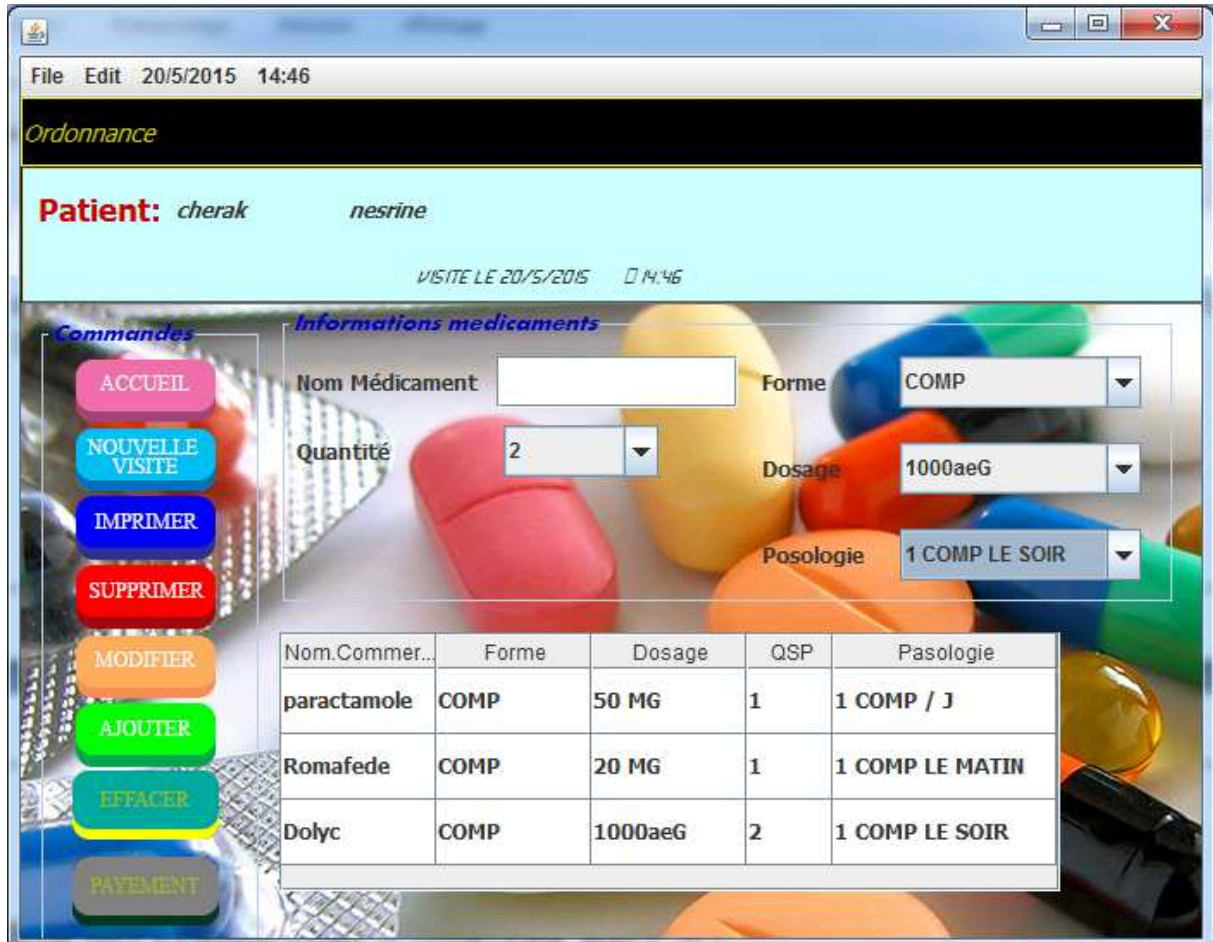


Figure III.10 : Interface Pour la Gestion des ordonnances

- *Description*

La gestion des ordonnances en mode écrire le nom de médicament et en mode sélectionné la quantité, forme, dosage, posologie.

Une exception sera déclenchée suite à une opération pour les médicaments, message d'erreur pour le champ de nom médicament les le entré le nom de médicament.

9. Gestion de la comptabilité :

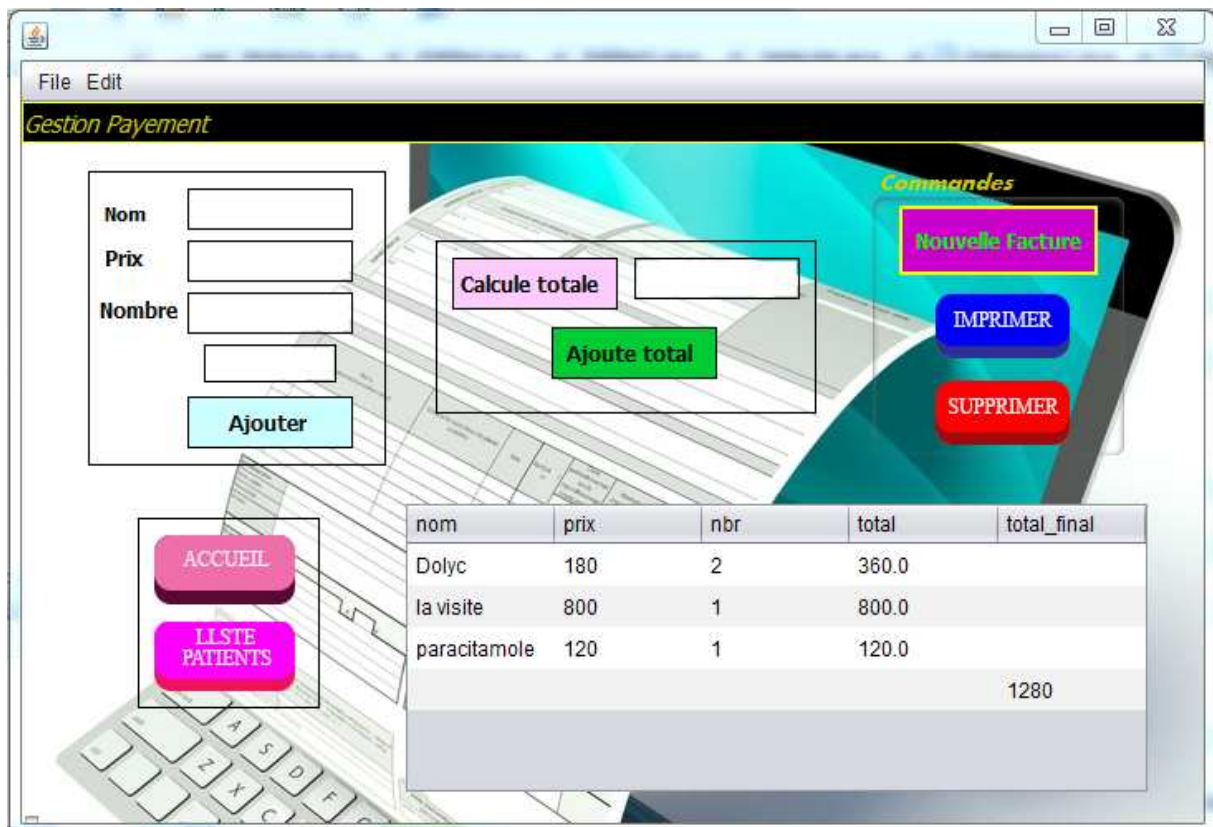


Figure III.11 : Interface gestion de la comptabilité

- **Description :**

Une exception a été déclenchée suite à une opération pour les dépenses, message d'erreur pour le champ de montant, sa valeur n'est pas de type numérique.

Le bouton « Ajouter » calcule le montant de (produit * quantité), et le bouton « Ajoute total » calcule le montant total.

Si vous entrez un mauvais nom ou prix, vous pouvez le supprimer sans revenir à remplir à nouveau la facture, tu peu aussi imprimé cette facture facilement de click sur bouton imprimer.

Une exception sera déclenchée suite à une opération pour les dépenses, Message d'erreur pour le champ de montant, le montant est vide.

Conclusion

Dans ce chapitre, nous avons présentés notre environnement de travail matériel et logiciel, les diagrammes de classes avec des schémas logiques et physiques des données ainsi que les principales interfaces de notre application avec leurs descriptions.

Grâce au schéma des relations, on connaît désormais les tables à créer pour l'application ainsi que les champs.

Le dossier d'analyse ainsi réalisé va permettre de mettre en œuvre une solution au niveau physique l'implantation de la structure de la base grâce au schéma logique des données relationnel.

Conclusion générale :

Ce projet nous a permis d'avoir une approche complète du développement de logiciel et une bonne initiation au cycle complet du développement de logiciel, de la conception à la validation en passant par les différentes étapes incrémentales de codage et de tests et nous a appris aussi à concevoir une base de données complète.

La réalisation de notre application est à présent achevée, elle comporte les fonctionnalités suivantes :

- **Gestion et Suivi du Dossier Médical**
- **Gestion des Rendez-vous.**
- **Gestion de la Fiche Patient.**
- **Gestion de la Fiche Médecin.**
- **Gestion de la Comptabilité.**
- **Gestion des Ordonnances.**

Elle a comporté deux volets à savoir le volet conception et volet réalisation :

- Sur le plan conceptuel nous avons utilisé le langage UML que nous avons bien maîtrisé à travers l'étude effectuée de l'application gestion du cabinet médical.

- Sur le plan pratique, cette application a été réalisée avec le **Système de Gestion Bases de Données de MySQL** serveur, **Netbeans_IDE7.4** comme environnement de développement et langage de programmation **Java**.

Nous avons donc eu l'opportunité d'approfondir nos connaissances que ce soit au plan scientifique ou personnel, pour conclure, on a évalué les principaux avantages et les points forts du logiciel pour améliorer la gestion d'un cabinet médical.

Comme une autre expérience au niveau de l'application des concepts de langages, c'est normal de ne pas pouvoir éviter certains problèmes et difficultés au niveau de la modélisation conceptuelle et au niveau de l'implémentation et la programmation.

Cependant, nous avons essayé de dégager les solutions les mieux adaptées à nos objectifs, nos contraintes et nos moyens disponibles, ces solutions ne prétendent nullement être les meilleures, car en informatique, il n'y a pas de solution absolue.

Il est à noter que cette application peut être améliorée, pour répondre aux besoins des autres spécialités plus appropriées, ainsi que le suivi de rapport d'activités des dossiers médicaux.

Nous avons proposé une solution à notre problématique, citée dans le premier chapitre, qui se résume en la sécurité des informations des patients qui sont confidentielles et il ne faut pas les perdre, et de faciliter la recherche de l'information pour le médecin.

Avec notre solution proposée, incluant les fonctionnalités de la plate forme **Netbeans7.4**, le médecin ne confond plus entre ses patients, surtout quand le nombre est élevé, et ne perd pas de temps dans la recherche des dossiers des patients.

Résumé

Des les premières années de construction des cabinets médicaux, les médecins s'orientaient à un objectif bien déterminé, celui-ci se résumait à comment travailler avec les patients à l'aide de moyens faciles et rapides. Mais malgré tous les efforts faites par les médecins, ils restent des difficultés, parmi les quelles : L'archivage des dossiers des patients sur des feuilles peut engendrer la perte des documents et perdre beaucoup de temps lors de la recherche, la modification d'un élément dans un dossier peut obliger le médecin à refaire un nouveau dossier.

A cause de ces difficultés, les médecins préfèrent s'approprier des bénéfices liés à l'informatisation de la gestion de leurs cabinets médicaux, dans le but de simplifier et de sécuriser leur travail.

Dans ce but, nous avons crée une application basée sur la plateforme netbeans7.4 avec le langage java, comme une solution pour aider le médecin à gérer leur cabinet médical.

Summary

From the early years of building medical cabinets, doctors were oriented to definite goal, it resumed to how to work with patients with quick and easy means. But despite all the efforts made by doctors they remain difficulties, among which: The archiving of patient folders on sheets may cause the loss of documents and lose a lot of time during the search, modifying an item in a folder may require the doctor to make a new folder.

Due to these difficulties, doctors prefer appropriating benefits linked to the computerization of the management of their medical cabinets, in order to simplify and secure their work.

For this aim, we have created an application based on the netbeans7.4 with the Java language as a solution to help doctors to manage their medical center.

ملخص

منذ السنوات الأولى من إنشاء المراكز الصحية، و الأطباء يعملون علي هدف محدد، يعتمد علي كيفية التعامل مع المرضى بطريقة سريعة وسهلة. ولكن علي الرغم من كل الجهود التي بذلت من قبل الأطباء ، إلا أن صعوبات كثيرة لا تزال موجودة : كأرشفة سجلات المرضى علي الأوراق الذي يمكن أن يسبب فقدان هذه الوثائق مع إضاعة الكثير من الوقت عند القيام بالبحث، إعادة تصحيح خطأ في الوثائق الخاصة بالمريض قد تتطلب من الطبيب إعادتها من جديد.

بسبب هذه الصعوبات، فضل الأطباء الاستفادة من الإعلام الآلي لإدارة المراكز الصحية، من أجل تأمين عملهم و ربح الوقت.

تحقيقاً لهذه الغاية ، قمنا بإنشاء تطبيق يستند علي البرنامج نت بينز 7.4 مع لغة جافا ، كحل لمساعدة الأطباء علي إدارة المركز الصحي.