

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche
Scientifique



UNIVERSITÉ D'EL-OUED

FACULTÉ DES SCIENCES ET DE TECHNOLOGIE

Mémoire de fin d'étude

LICENCE ACADEMIQUE

Domaine: Mathématiques et Informatique

Filière: Informatique

Spécialité: Informatique fondamentale

Présenté par: * Guediri Taoufik

* Khechimmadi Abdelaziz

Thème

**Mise en place d'un logiciel pour
la gestion d'un cabinet médical**

Soutenu le juin 2014

Devant le jury composé de:

Mr DJLID Rouf

MA (A) Univ. El Oued Président

Mr BELILA Khaoula

MA (A) Univ. ElOued Examineur

Mr Hadjadj Ismail

MA (B) Univ. ElOued Rapporteur

Année universitaire 2013 – 2014

Remerciements

*Tout d'abord, nous remercions notre dieu qui nous donne la force
pour accomplir ce modeste travail.*

*Nous adressons nos plus sincères remerciements notre encadreur
monsieur Ismail Hadjadj pour son aide et ses conseils précieux.*

*Nous remercions chaleureusement les membres du jury qui m'ont
fait l'honneur de participer à notre jury et d'accepter de juger ce
travail.*

*En fin, nous voulons encore remercier tous ceux qui nous ont
consolidé et nous ont encouragé entre autre les parents et les
amis.*

ملخص

آلية النظام تسمح بهيكلية وتسيير مجموعة من البيانات بهدف التنظيم والحصول على نتائج سريعة وفعالة.

وعلى هذا السياق أنشأنا برنامج لتسيير عيادة طبية.

هذا البرنامج يقوم بتنظيم آلي للعيادة بطريقة سهلة ومبسطة وأكثر حماية للسجلات الطبية.

كما يحتوي هذا البرنامج على الخصائص التالية

- إدارة ومتابعة ملفات المرضى.

- إدارة المواعيد.

الكلمات المفتاحية: المريض, الوصفة, الموعد, فحص المريض.

Résumé

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 et efficace.

Dans ce cadre, nous sommes appelés à concevoir, implémenter 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 des dossiers médicaux.
- Gestion des rendez-vous.

Le mot clé : Patient, Ordonnance, Rendez-vous, Traitement de patient.

Abstract

The automated information system is to structure and manage a data set with the aim to organize and to have fast and effective results.

In this context, we are able to design, implement software for managing a medical practice.

The software should place the organization and automation of the management of a medical practice to increase the reliability, efficiency of human effort and facilitate tedious tasks within the firm.

Our application will include the following features:

- Monitoring and management of medical records.
- Management appointments.

Keyword: Patient, Prescription, Appointment, Patient treatment.

Liste des figures

Figure. I. 1:Peu d'histoir.....	5
Figure. I. 2: Type des diagrammes	6
Figure. I. 3: Les principes de processus unifie.....	10
Figure. I. 4: Le processus de développement en Y.....	12
Figure.II.1 : Diagramme de cas d'utilistion.....	17
Figure.II. 2 : Diagramme de cas d'utilisation « Authentification ».....	17
Figure.II.3: Diagramme d'activité «Authentification».....	18
Figure.II.4: Diagramme de séquence «Authentification».....	19
Figure.II.5: Diagramme de cas d'utilisation «Rendez-vous »	20
Figure.II.6: Diagramme d'activité «Rendez-vous ».....	20
Figure.II.7: Diagramme de séquence «Prenez un rendez-vous».	21
Figure.II.8: Diagramme de séquence «Modifier un rendez-vous».	21
Figure.II.9: Diagramme de séquence «Supprimer un rendez-vous».	21
Figure.II.10: Diagramme de cas d'utilisation « Suivi dossier ».	22
Figure.II.11: Diagramme d'activité «Suivi de dossier».	23
Figure.II.12: Diagramme de séquence «Ajoute un nouveaux patient ».	23
Figure.II.13: Diagramme de séquence «Historique de patient».	24
Figure.II.14: Diagramme de séquence «Modification d'information des patient ».	23
Figure.II.15: Diagramme de cas d'utilisation « Rechercher ».....	25
Figure.II.16: Diagramme d'activité « Rechercher ».....	25
Figure.II.17: Diagramme de séquence « Rechercher ».	26
Figure.II.18: Diagramme de cas d'utilisation «Traitement de patient ».....	26
Figure.II.19: Diagramme d'activité «Traitement de patient ».....	27
Figure.II.20: Diagramme de séquence «Enregistre information de traitement ».	27
Figure.II.21: Diagramme de séquence «Crée(Ordonnance/facture) ».....	28
Figure.II.22: Diagramme de classe.....	29
Figure.III.1:L'interface d'authentification.....	37
Figure.III.2:L'interface principale du partie secretaire.....	38
Figure.III.3:L'interface de rendez-vous d'aujourd'hui.....	38
Figure.III.4:Tableau de rendez-vous.....	39
Figure.III.5:L'interface de traitement.....	40
Figure.III.6:L'interface de suivi des dossiers.....	41
Figure.III.7:L'interface de modification de cabinet.....	42

Liste des tables

Table. II.1 : Liste des cas d'utilisations.	17
Table. II.2: Liste des classes.....	30
Table. II.3: Liste des associations.	31
Table.II.4: Equivalences entre les concepts objets et relationnels.	32
Table.III.5: Liste des tables de la base de données.....	32

Sommaire

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

Chapitre I : La méthode de développement

1. Introduction.....	4
2. L'UML	4
2.1. Définition de l'UML	4
2.2. Un peu d'histoire	5
2.3. Les diagrammes de l'UML	5
a-Diagrammes structurels	6
b-Diagrammes comportementaux.....	7
2.4. Caractéristique d'UML	8
a. UML est un support de communication	8
b. UML n'est pas une méthode ou un processus	8
2.5. Points forts et points faibles de l'UML	9
2.5.1. Points forts d'UML	9
2.5.2. Points faibles d'UML.....	9
3. Le Processus de développement.....	9
3.1. Processus Unifié	10
3.1.1. Définition d'UP	10
3.1.2. Les diagrammes de l'UML	10
3.2. Le Processus 2TUP	11
3.2.1. Définition	11
3.2.2. Les branches du 2TUP	12
a. Branche fonctionnels	12
b. Branche architecture technique.....	13
c. Branche conception	13
4. Conclusion	13

Chapitre II : La modélisation

1. Introduction.....	15
2. Etude préliminaire du contexte	16
2.1. Description du système	16
2.2. Identification des acteurs	16
3. Capture des besoins fonctionnels.....	16
- Identification des cas d'utilisation	16
➤ Cas d'utilisation «Authentification»	18
➤ Cas d'utilisation «Rendez-vous»	19
➤ Cas d'utilisation «Suivi des dossier».....	22
➤ Cas d'utilisation « Rechercher »	24
➤ Cas d'utilisation « Traitement de patient »	19
4. Capture de besoins techniques	28
5. Développement du modèle statique	28
- Diagramme de classe générale	29
6. Conception détaillée.....	29
6.1. Description du modèle des classes	30
6.2. Description du modèle d'association.....	31
6.3. Passage au modèle relationnel	31
7. Conclusion.....	33

Chapitre III : La réalisation du système

1. Introduction.....	35
2. Environnement de travail	35
2.1. Langage de programmation.....	35
2.2. .NET Framework.....	35
2.3. Microsoft Visual Studio.Net	36
2.4. SGBD	36
3. Présentation du logiciel	37
3.1. L'interface authentification	37
3.2. Partie secrétaire.....	37

3.2.1. L'interface principale de partie secrétaire.....	37
3.2.2. Les rendez-vous d'aujourd'hui.....	38
3.2.3. Tableau de rendez-vous.....	39
3.3. Partie médecin	40
3.3.1. L'interface de traitement.....	40
3.3.2. L'interface de suivi de dossier.....	41
3.3.3. Modification l'information de cabinet	42
4. Conclusion.....	42
Conclusion générale	43
Bibliographies	44

Introduction générale

La gestion d'un cabinet médical est une tâche complexe. Cependant, les premières années de construction des cabinets médicaux et 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 des médecins il reste des difficultés, parmi lesquelles:

L'archivage des dossiers des patients sur des feuilles peut engendrer leur perte et perdre beaucoup de temps pour trouver un dossier.

La modification d'un élément dans un dossier peut obliger le médecin à refaire un nouveau dossier.

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

Pour cela les médecins préfèrent s'approprier des bénéfices liés à l'informatisation de la gestion de leurs cabinets médicaux dans le but de simplifier, sécurisé et minimisé leur travail.

Notre projet est découpé en trois chapitres

- Le premier chapitre aborde la méthode de développement qu'on va la suivre et du langage de modélisation unifié UML. Il contient les concepts généraux d'UML et du processus de développement 2TUP (2 Track Unified Processus).
- Le deuxième chapitre concerne la conception du nouveau système à l'aide des diagrammes UML selon la démarche de processus 2TUP.

- Le dernier chapitre présente les outils utilisés lors de la réalisation du système, ainsi que quelques interfaces représentant notre application.

Enfin, nous concluons ce projet par une conclusion générale.

Chapitre I

***La méthode de développement
(UML + 2TUP)***

1. Introduction

Dans ce chapitre nous présentons un outil de modélisation UML et le processus de développement logiciel, tel que le développement avec l'UML demande un processus qui permet d'envisager la marche à suivre lors de la conception.

UML est conçu pour être indépendant du processus de développement et le choix de celui-ci dépend le type de projet, pour notre système on choisit le processus 2TUP, car il traite les projets du domaine de système d'information.

2. L'UML

2.1. Définition de l'UML :

UML ou (Unified Modeling Language) est un langage de modélisation orienté objet le plus connu et le plus utilisé au monde convient à tous les langages objets tel que C++, JAVA et C#. [1]

UML est un langage de modélisation au sens de la théorie des langages. Il contient de ce fait les éléments constitutifs de tout langage, à savoir : des concepts, une syntaxe et une sémantique. De plus, UML a choisi une notation supplémentaire : il s'agit d'une forme visuelle fondée sur des diagrammes. Si l'unification d'une notation est secondaire par rapports aux éléments constituant le langage, elle reste cependant primordiale pour la communication et la compréhension. [2]

2.2. Un peu d'histoire :

Lorsque la programmation par objets prend de l'importance au début des années 1990, la nécessité d'une méthode qui lui soit adaptée devient évidente. Plus de cinquante méthodes apparaissent entre 1990 et 1995 (Booch, Classe-Relation, Fusion, HOOD, OMT, OOA, OOD, OOM, OOSE, etc.) mais aucune ne parvient à s'imposer. En 1994, le consensus se fait autour de trois méthodes:

- OMT de James Rumbaugh (General Electric) fournit une représentation graphique des aspects statique, dynamique et fonctionnel d'un système ;
- OOD de Grady Booch, définie pour le Department of Defense, introduit le concept de paquetage .

- OOSE d'Ivar Jacobson (Ericsson) fonde l'analyse sur la description des besoins des utilisateurs (cas d'utilisation, ou use cases).

Les constructeurs des ces méthodes (les trois Amigos) se mirent d'accord pour définir une méthode commune qui fédérerait leurs apports respectifs.

L'unification a progressé par étapes. En 1995, Booch et Rumbaugh fondèrent la Rational Software.

Corporation pour unir leurs efforts en vue de créer une méthode unifiée (Unified Method) à partir de leurs méthodes. en 1996, Jacobson les a rejoints pour produire UML 0.9 (notez le remplacement du mot méthode par le mot langage, plus modeste). Les acteurs les plus importants dans le monde du logiciel s'associent alors à l'effort (IBM, Microsoft, Oracle, DEC, HP, Rational, Unisys etc.) et UML 1.0 est soumis à l'OMG

L'OMG adopte en novembre 1997 UML 1.1 comme langage de modélisation des systèmes d'information à objets.

La version d'UML en cours est UML 2.X (UML 2.0, publié en 2005), présente quant à lui de profondes modifications et de véritables extensions par rapport aux versions 1.X, comme par exemple les diagrammes d'interaction. Et les travaux d'amélioration se poursuivent.

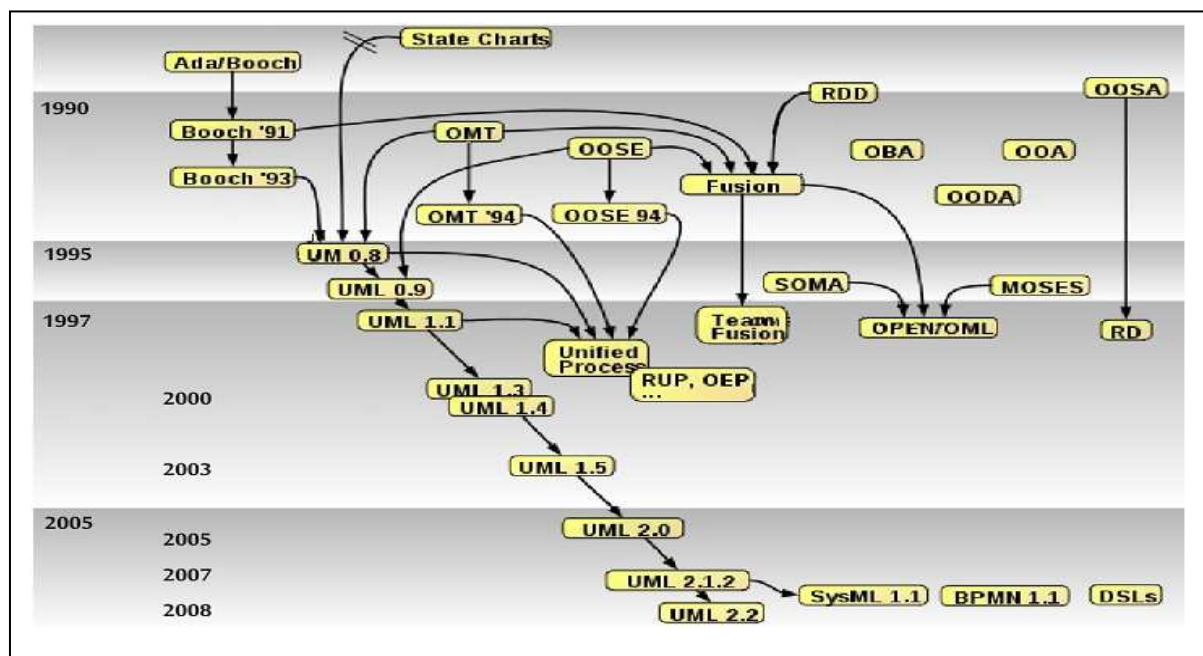


Figure. I. 1: peu d'histoire d'UML[3]

2.3. Les diagrammes de l'UML :

UML 2.0 comporte treize types de diagrammes représentant autant de vues distinctes pour représenter des concepts particuliers du système d'information. Ils se répartissent en deux grands groupes. [1]

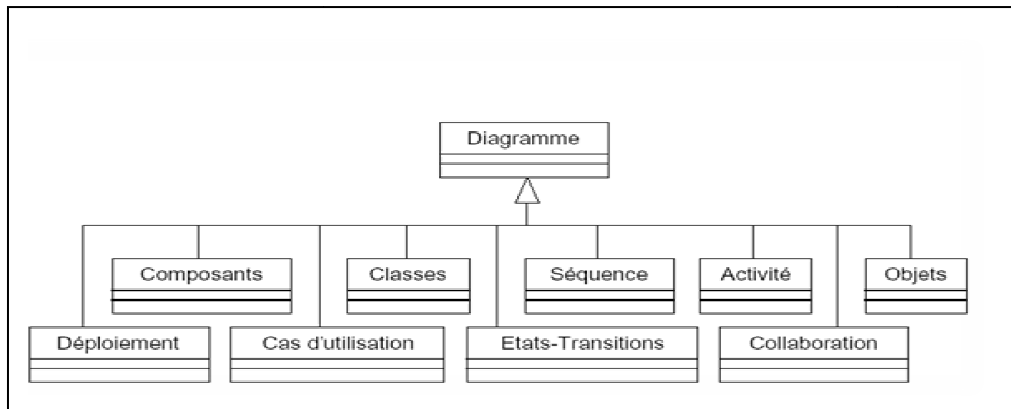


Figure. I. 2: type des diagrammes

Diagrammes structurels ou diagrammes statiques (UML Structure)

- **Le diagramme de classes :** est généralement considéré comme le plus important dans un développement orienté objet. Il montre la structure interne d'un système. Il contient principalement des classes. [1]
- **Le diagramme d'objets :** permet d'éclairer un diagramme de classes en l'illustrant par des exemples. Il est, par exemple, utilisé pour vérifier l'adéquation d'un diagramme de classes à différents cas possibles. [1]
- **Le diagramme de composant :** permet de définir des composants logiciels d'implémentation et leur relation. [1]
- **Le diagramme de structure composite :** permet de décrire sous forme de boîte blanche les relations entre composants d'une classe. [1]
- **Le diagramme de déploiement :** décrit les ressources matérielles et la répartition du logiciel (objets et composants) dans ces ressources. [1]
- **Le diagramme des paquetages :** sert à représenter les dépendances entre paquetage, c'est-à-dire les dépendances entre ensembles de définitions. [1]

Diagrammes comportementaux ou diagrammes dynamiques (UML Behavior)

- **Le diagramme de cas d'utilisation :** représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. Il permet de recueillir, d'analyser et d'organiser les besoins. [1]
- **Le diagramme d'activité :** Les diagrammes d'activités permettent de mettre l'accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de représenter graphiquement le déroulement d'un cas d'utilisation. [1]
- **Le diagramme d'états-transitions :** représente la façon dont évoluent (c'est-à-dire le cycle de vie) les objets appartenant à une même classe. La modélisation du cycle de vie est essentielle pour représenter et mettre en forme la dynamique du système. [1]
- **Le diagramme d'interaction :** est sous-groupes de diagrammes dans cet axe il comporte les diagrammes suivant :
 - **Le diagramme de séquence :** Selon un point de vue temporel, on y met l'accent sur la chronologie des envois de messages. Les diagrammes de séquences peuvent servir à illustrer un scénario d'un cas d'utilisation. Représente les interactions entre les agents et le système selon un ordre chronologique. [1]
 - **Le diagramme de communication :** est basé sur une représentation spatiale des objets. Chaque objet est lié graphiquement aux objets avec lesquels il interagit. [1]
 - **Le diagramme d'interaction globale :** permet de décrire les enchaînements possibles entre les scénarios préalablement identifiés sous forme de diagrammes de séquences (variante du diagramme d'activité). [1]
 - **Diagramme de temps :** permet de décrire les variations d'une donnée au cours du temps. [1]

2.4. Caractéristique d'UML:

a. UML est un support de communication:

- UML est avant tout un support de communication performant, qui facilite la représentation et la compréhension de solutions objet.
- Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de solutions.
- L'aspect formel de sa notation limite les ambiguïtés et les incompréhensions.
- Son indépendance par rapport aux langages de programmation, aux domaines d'application et aux processus, en font un langage universel. [2]

b. UML n'est pas une méthode ou un processus:

Une méthode propose aussi un processus, qui régit notamment l'enchaînement des activités de production d'une entreprise. Or, UML a été pensé pour permettre de modéliser les activités de l'entreprise, pas pour les régir. [2]

✓ UML décrit un méta modèle :

UML est fondé sur un **méta modèle**, qui définit :

- les éléments de modélisation (les concepts manipulés par le langage),
- la sémantique de ces éléments (leur définition et le sens de leur utilisation).

Un méta modèle est une description très formelle de tous les concepts d'un langage. Il limite les ambiguïtés et encourage la construction d'outils.

Le méta modèle d'UML permet de classer les concepts du langage (selon leur niveau d'abstraction ou domaine d'application) et expose sa structure.

Le méta modèle UML est lui-même décrit par un méta-méta modèle (OMG-MOF).

UML propose aussi une notation, qui permet de représenter graphiquement les éléments de modélisation du méta modèle.

✓ UML est un langage de modélisation :

UML est un langage de modélisation au sens de la théorie des langages. Il contient de ce fait les éléments constitutifs de tout langage, à savoir : des concepts, une syntaxe et une sémantique.

De plus, UML a choisi une notation supplémentaire : il s'agit d'une forme visuelle fondée sur des diagrammes. [2]

2.5. Points forts et points faibles de l'UML

2.5.1. Points forts d'UML:

- **UML est un langage formel et normalisé**

Il permet ainsi :

- un gain de précision
- un gage de stabilité
- l'utilisation d'outils
- **UML est un support de communication performant**

Il cadre l'analyse et facilite la compréhension de représentations abstraites complexes. Son caractère polyvalent et sa souplesse en font un langage universel. [2]

2.5.2. Points faibles d'UML:

- La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation.

UML n'est pas à l'origine des concepts objets, mais en constitue une étape majeure, car il unifie les différentes approches et en donne une définition plus formelle.

- Le processus (non couvert par UML) est une autre clé de la réussite d'un projet.

L'intégration d'UML dans un processus n'est pas triviale et améliorer un processus est une tâche complexe et longue. [2]

3. Le Processus de développement

Un processus définit une séquence d'étapes, en partie ordonné, qui concourt à l'obtention d'un système logiciel ou à l'évolution d'un système existant. Pour produire des logiciels de qualité, qui répondent aux besoins des utilisateurs dans des temps et des coûts prévisibles. On découpe le processus en deux axes :

- L'axe de développement technique, qui se concentre principalement sur la qualité de production.

- L'axe de gestion du développement, qui permet la mesure et la prévision des coûts et des délais. [4]

3.1. Processus Unifié (Unifier Processus): (UP)

3.1.1. Définition d'UP :

Le processus unifié est un processus de développement logiciel : il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel. [2]

3.1.2. Les caractéristiques essentielles du processus unifié :

- Le processus unifié utilise le langage UML (ensemble d'outils et de diagramme),
- Le processus unifié est piloté par les cas d'utilisation,
- Centré sur l'architecture,
- Itératif et incrémental.

La figure.II.2 représente les caractéristiques du processus unifié.

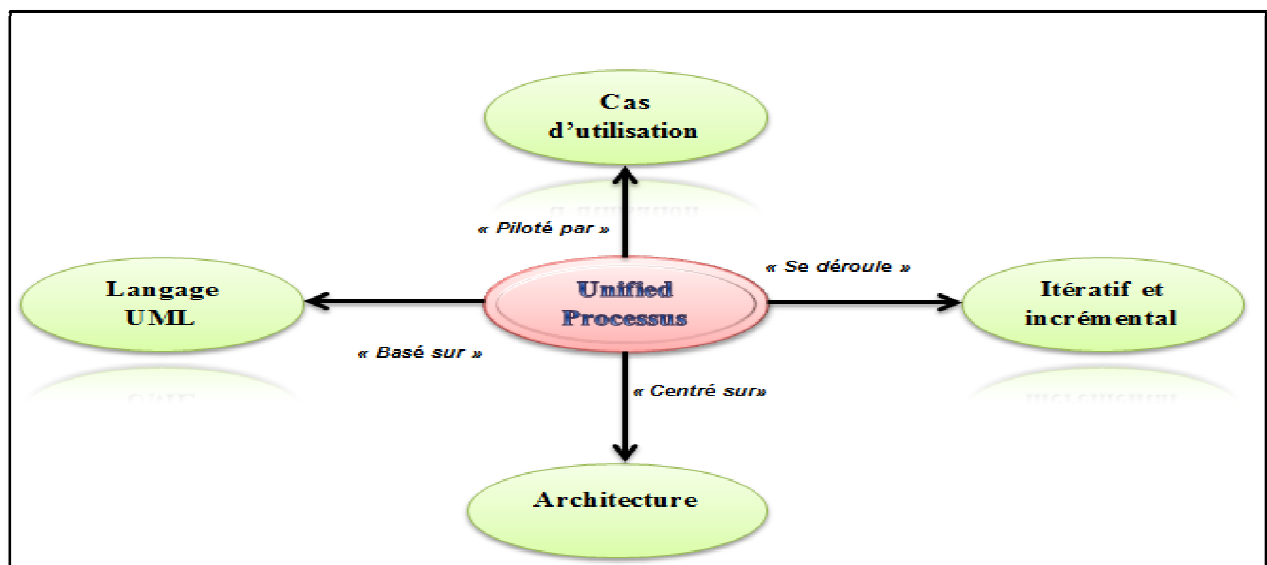


Figure. II. 3: Les principes de Processus Unifié.

a. Piloté par les cas d'utilisation

Les cas d'utilisation ne sont pas un simple outil de spécification des besoins du système. Ils vont complètement guider le processus de développement à travers l'utilisation de modèles basés sur l'utilisation du langage UML. [2]

b. Centré sur l'architecture

Dès le démarrage du processus, on aura une vue sur l'architecture à mettre en place. L'architecture d'un système logiciel peut être décrite comme les différentes vues du système qui doit être construit. L'architecture logicielle équivaut aux aspects statiques et dynamiques les plus significatifs du système. L'architecture émerge des besoins de l'entreprise, tels qu'ils sont exprimés par les utilisateurs et autres intervenants et tels qu'ils sont reflétés par les cas d'utilisation. [2]

c. Itératif et incrémental

Le développement d'un produit logiciel destiné à la commercialisation est une vaste entreprise qui peut s'étendre sur plusieurs mois. On ne va pas tout développer d'un coup. On peut découper le travail en plusieurs parties qui sont autant de mini projets. Chacun d'entre eux représentant une itération qui donne lieu à un incrément.

Une itération désigne la succession des étapes de l'enchaînement d'activités, tandis qu'un incrément correspond à une avancée dans les différents stades de développement.

Le choix de ce qui doit être implémenté au cours d'une itération repose sur deux facteurs :

- Une itération prend en compte un certain nombre de cas d'utilisation qui ensemble, améliorent l'utilisabilité du produit à un certain stade de développement.
- L'itération traite en priorité les risques majeurs. [2]

3.2 Le processus 2TUP

3.2.1. Définition :

Le processus 2TUP (2Tracks Unified Processus) apporte une réponse aux contraintes de changement continu imposées aux systèmes d'information de l'entreprise. [4]

Idée de base du 2TUP : toute évolution imposée au SI peut se décomposer et se traiter parallèlement, suivant 2 axes (« 2 tracks ») : Un axe fonctionnel et Un axe technique

La réalisation du système consiste à fusionner les résultats des deux branches.

A l'issue des évolutions du modèle fonctionnel et de l'architecture technique, la réalisation du système consiste à fusionner le résultat des deux branches. Cette fusion conduit à l'obtention

d'un processus de développement en forme de Y comme illustré dans la figure. I.4 ci-dessous.
[5]

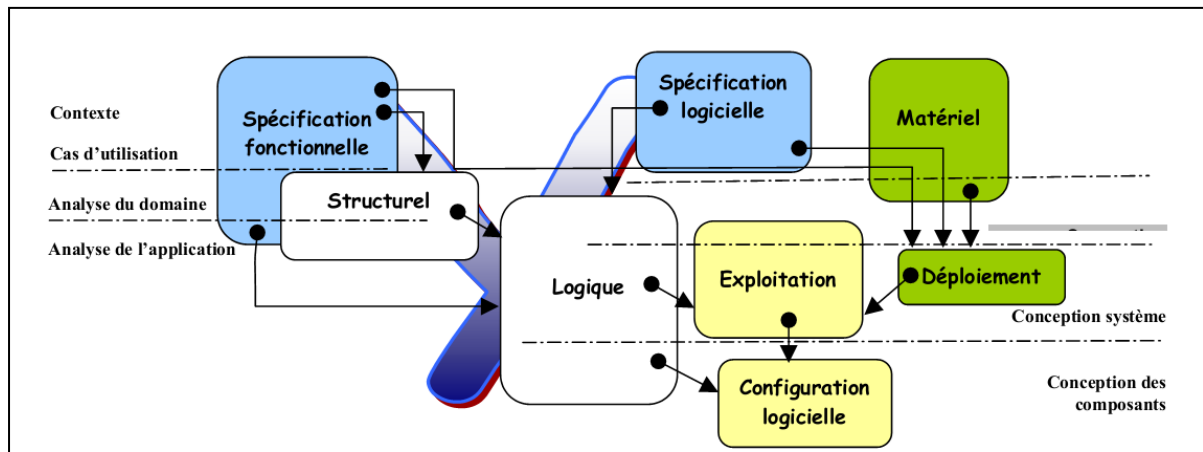


Figure. I. 4:le processus de développement en Y

3.2.2 Les branche du 2TUP :

a. Branche fonctionnels (gauche):

- Capture des besoins fonctionnels, qui produit le modèle des besoins focalisé sur le métier des utilisateurs. Elle qualifie, au plus tôt le risque de produire un système inadapté aux utilisateurs
- L'analyse, qui consiste à étudier précisément la spécification fonctionnelle de manière à obtenir une idée de ce que va réaliser le système en terme de métier. [6]

b. Branche architecture technique (droite) :

- La capture des besoins techniques, qui recense toutes les contraintes sur les choix de dimensionnant et la conception du système. Les outils et le matériels sélectionnés ainsi que la prise en compte des contraintes d'intégration avec l'existant (pré requis d'architecture technique).
- La conception générique, qui définit ensuite les composants nécessaires à la construction de l'architecture technique. Cette conception est complètement indépendante des aspect fonctionnel. Elle a pour objectif de d'uniformiser et de réutiliser les mêmes mécanismes pour tout un système. L'architecture technique construit le squelette du système, son importance est telle qu'il est conseillé de réaliser un prototype. [6]

c. Branche conception (milieu) :

- La conception préliminaire, qui représente une étape délicate, car elle intègre le modèle d'analyse fonctionnelle dans l'architecture technique de manière à tracer la cartographie des composants du système à développer.
- La conception détaillée, qui étudie ensuite comment réaliser chaque composant.
- L'étape de codage, qui produit ses composants et teste au fur et à mesure les unités de code réalisées.
- L'étape de recette, qui consiste enfin à valider les fonctionnalités du système développé. [6]

4. Conclusion

Dans ce chapitre nous avons présenté les principales notions et concepts de l'UML et le Processus Unifié (2TUP), et dans le chapitre suivant en applique ces concepts pour modéliser notre système.

Chapitre II

La modélisation

1. Introduction

Le but de ce chapitre est la modélisation de notre système pour faire une bonne réalisation à la fin de l'étude. La conception de nouveau système nous nous sommes fixés les objectifs suivants :

- ✓ Traitement automatique des dossiers des patients.
- ✓ Facilitation la modification et l'extraction des informations de patient.
- ✓ Garanti l'arrivée des informations dans le temps exacte.

Donc pour garantir tout ça, il faut utiliser une méthode ou démarche de conception. Nous avons voir dans le chapitre précédent que le processus 2TUP s'appuie sur UML tout au long du cycle de développement, car les différents diagrammes de ce dernier permettent de part leur facilité et clarté, de bien modéliser le système à chaque étape. L'UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier, concevoir des solutions et communiquer des points de vue.

Dans ce chapitre on va présenter les étapes suivant du processus 2TUP :

- ✓ Etude préliminaire du contexte.
- ✓ Capture des besoins fonctionnels.
- ✓ Capture de besoins techniques.
- ✓ Développement du modèle statique.
- ✓ Conception détaillée.

2. Etude préliminaire du contexte

L'étude préliminaire (ou Pré-étude) est la toute première étape du processus 2TUP. Elle consiste à effectuer un premier repérage des besoins fonctionnels et opérationnels, en utilisant principalement le texte, ou diagrammes très simples. Elle prépare les activités plus formelles de capture des besoins fonctionnels et de capture techniques. [7]

2.1. Description du système :

L'objectif de notre système est la gestion des cabinets médicaux. Il permet de créer et mettre à jour les fiches des patients. Le médecin peut examiner des différentes informations, le système effectue la recherche dans les dossiers des patients.

2.2. Identification des acteurs :

Nous allons maintenant énumérer les acteurs susceptibles d'interagir avec le système, mais d'abord nous donnons une définition de ce que c'est un acteur.

Définition : 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. [7]

Pour notre système, on peut distinguer deux acteurs principaux:

✓ Médecin



✓ Secrétaire



3. Capture des besoins fonctionnels

Cette phase représente un point de vue « fonctionnel » de l'architecture système. Par le biais des cas d'utilisation, nous serons en contact permanent avec les acteurs du système en vue de définir les limites de celui-ci, et ainsi éviter de trop s'éloigner des besoins réels de l'utilisateur final. [7]

- Identification des cas d'utilisation:

Le cas d'utilisation représente un ensemble de séquences d'actions réalisées par le système et produisant un résultat observable intéressant pour un acteur particulier. Un cas

d'utilisation modélise un service rendu par le système. Il exprime les interactions acteurs/système et apporte une valeur ajoutée « notable » à l'acteur concerné. [7]

Le tableau ci-après récapitule l'ensemble des cas d'utilisation qui seront détaillés par la suite.

Table. II. 1 : Liste des cas d'utilisations.

Cas d'utilisation	Acteurs
Suivi des dossiers Traitement des patients	Médecin
Gestion de RDV	Secrétaire

Dans la figure suivant nous présentons notre diagramme de cas d'utilisation

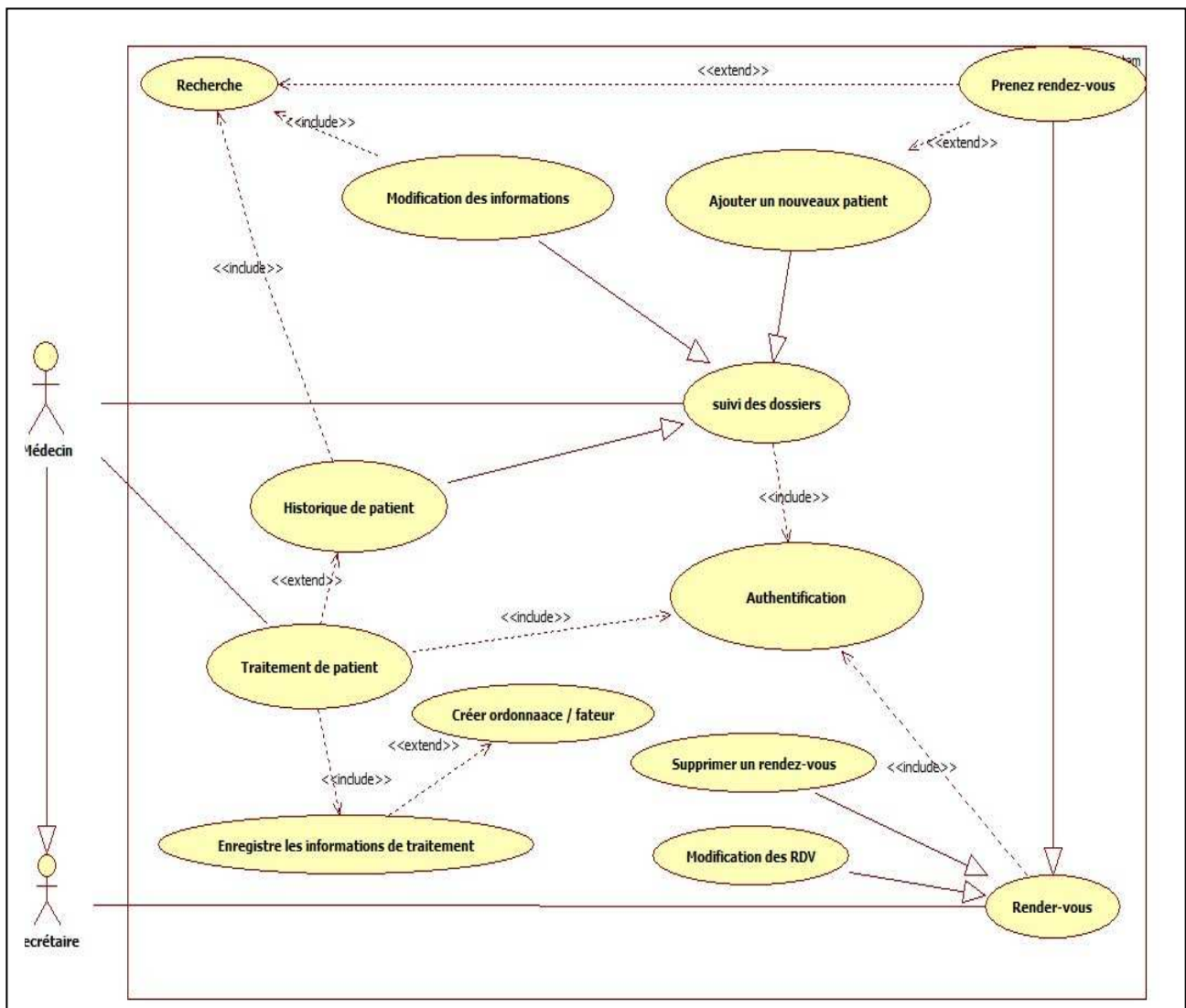


Figure. II. 1 : Le diagramme de cas d'utilisation.

➤ **Cas d'utilisation «Authentification» :**

- Intention: Autorisation d'accès au système.
- Actions:
 - Choisir le type d'utilisateur et saisir le mot de passe.
 - Valider puis connecter.

a. **Diagramme de cas d'utilisation :**

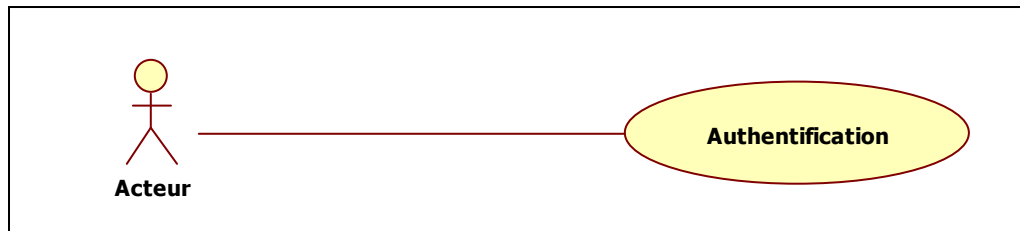


Figure. II.2: Diagramme de cas d'utilisation « Authentification ».

b. **Diagramme d'activité :**

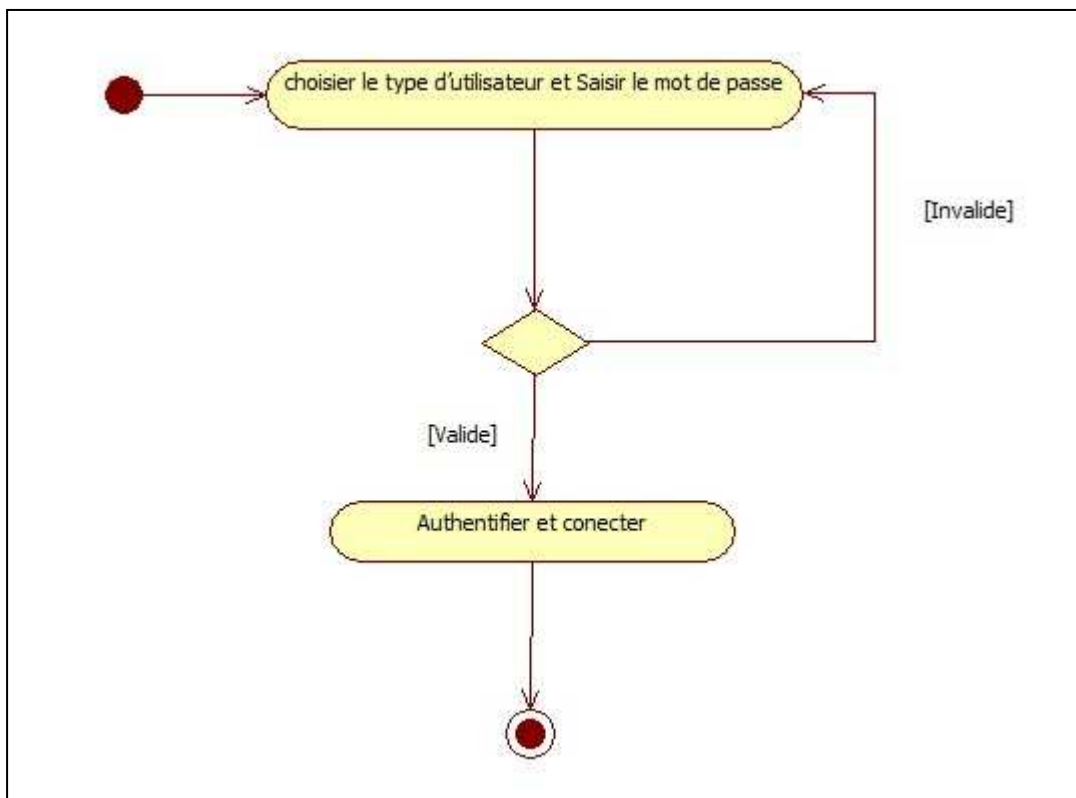


Figure. II.3: Diagramme d'activité «Authentification».

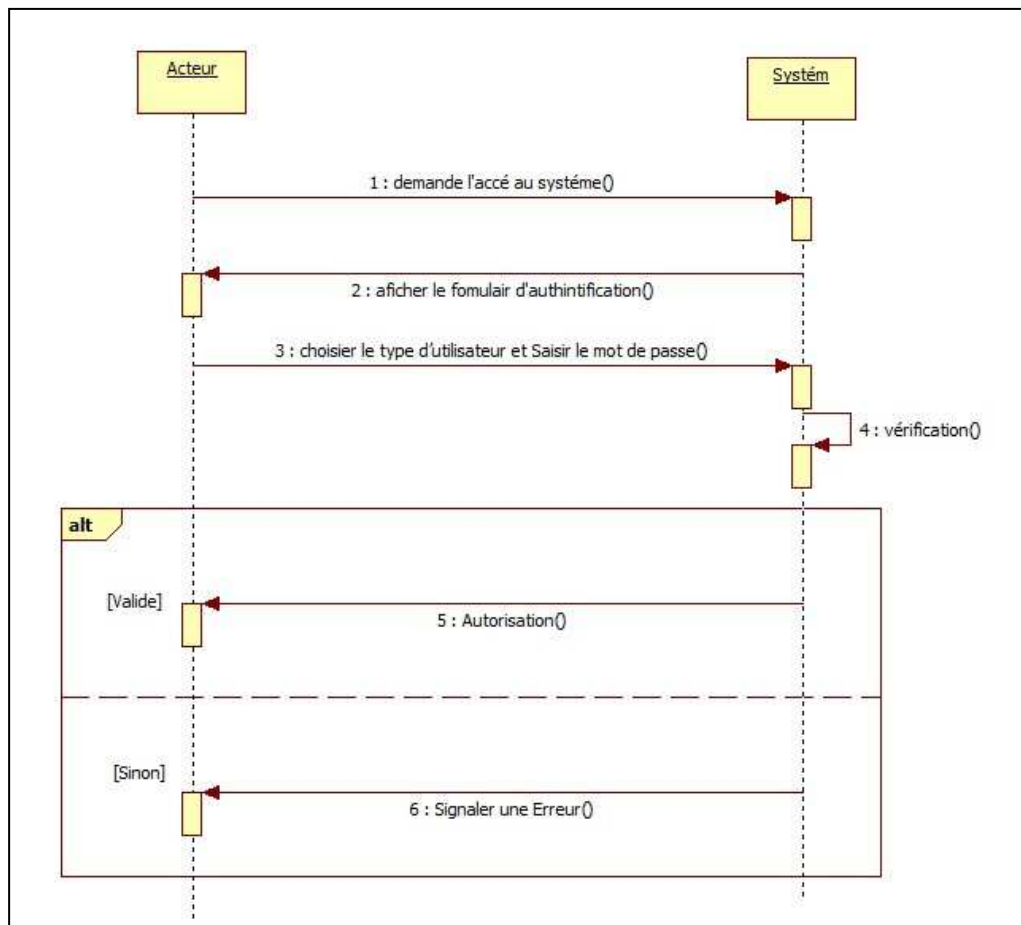
c. Diagramme de séquence : « Authentification »

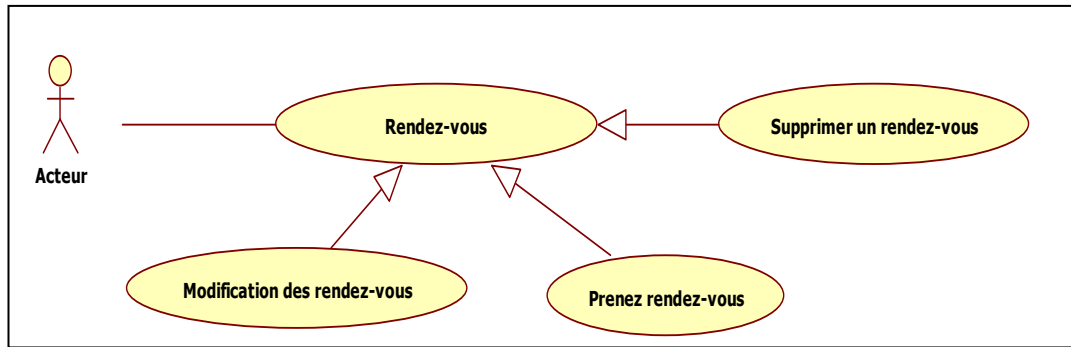
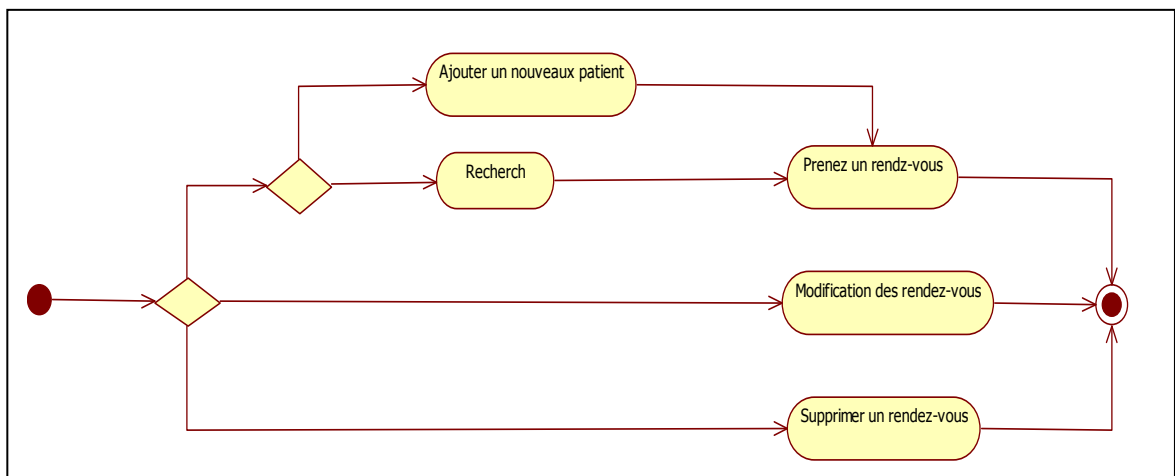
Figure. II.4: Diagramme de séquence «Authentification».

➤ **Cas d'utilisation « Rendez-vous » :**

- Intention : Permet aux les acteurs de notre système la gestion de rendez-vous.
- Actions :

Après l'authentification.

- Prenez un rendez-vous.
- Modifier un rendez-vous
- supprimer un rendez-vous.

a. Diagramme de cas d'utilisation :**Figure. II.5: Diagramme de cas d'utilisation «Rendez-vous».****b. Diagramme d'activité :****Figure. II. 6: Diagramme d'activité «Rendez-vous ».**

c. Diagramme de séquence : « Prenez un rendez-vous »

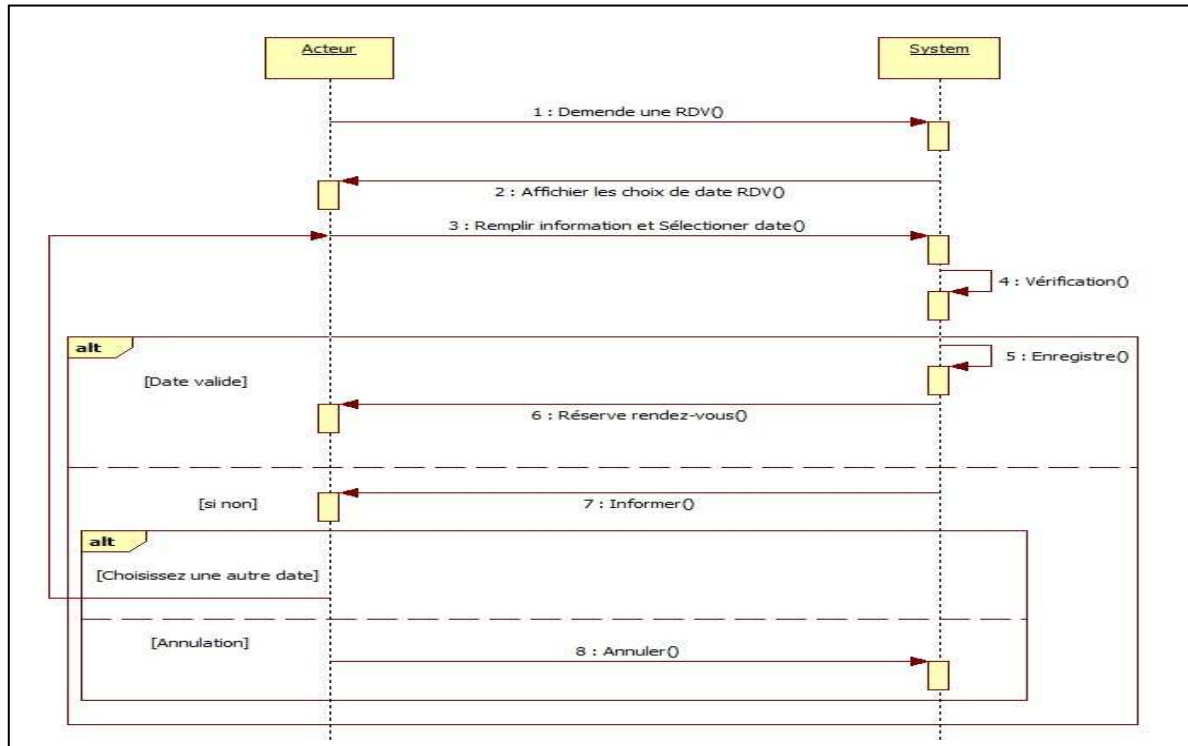


Figure. II.7: Diagramme de séquence «Prenez un rendez-vous».

d. Diagramme de séquence : « Modifier un rendez-vous »

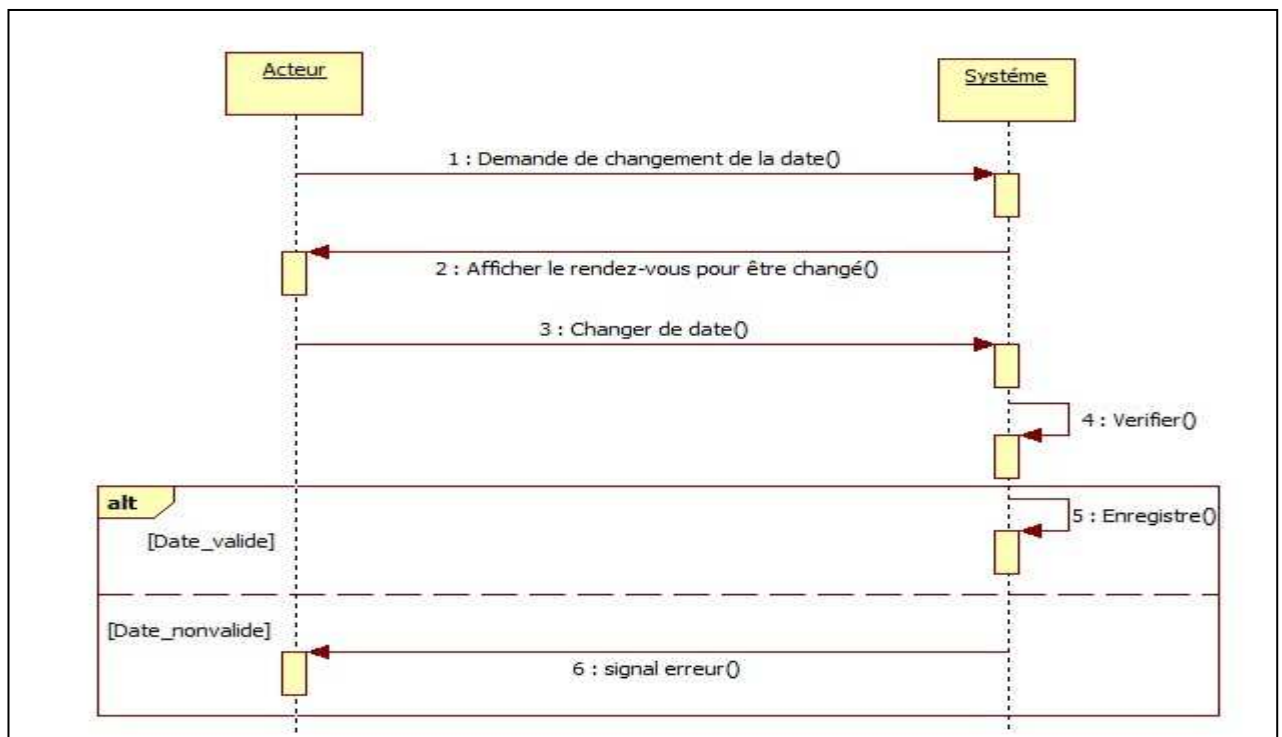


Figure. II.8: Diagramme de séquence «Modifier un rendez-vous».

e. Diagramme de séquence : « supprimer un rendez-vous »

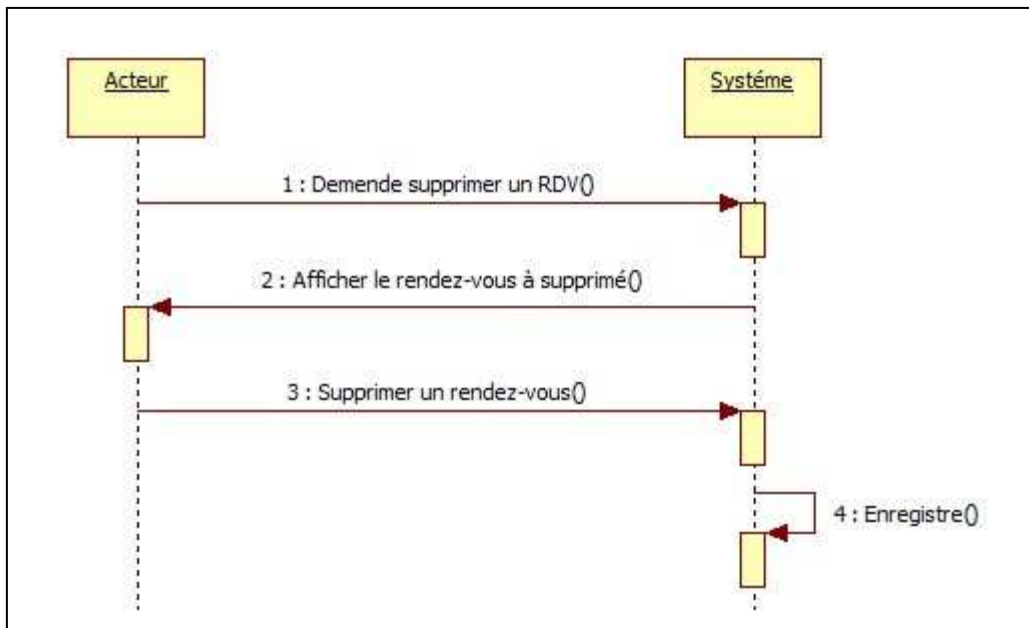


Figure. II.9: Diagramme de séquence «supprimer un rendez-vous».

➤ Cas d'utilisation «Suivi des dossiers» :

- Intention : gérer les information des patients et suivi ses dossiers.
- Actions :
 - Ajouter un nouveaux patient.
 - Modification des information des patients.
 - Historique de patient.

a. Diagramme de cas d'utilisation :

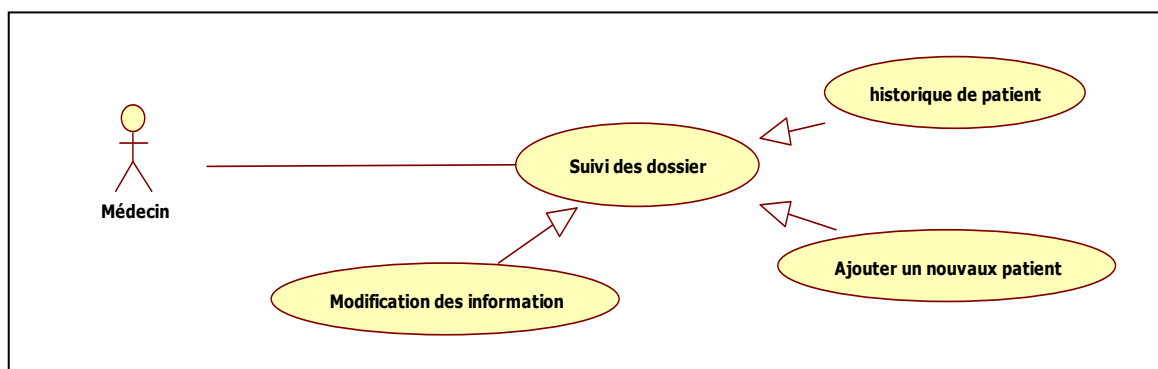


Figure. II.10: Diagramme de cas d'utilisation « Suivi des dossiers ».

b. Diagramme d'activité : «Suivi des dossiers».

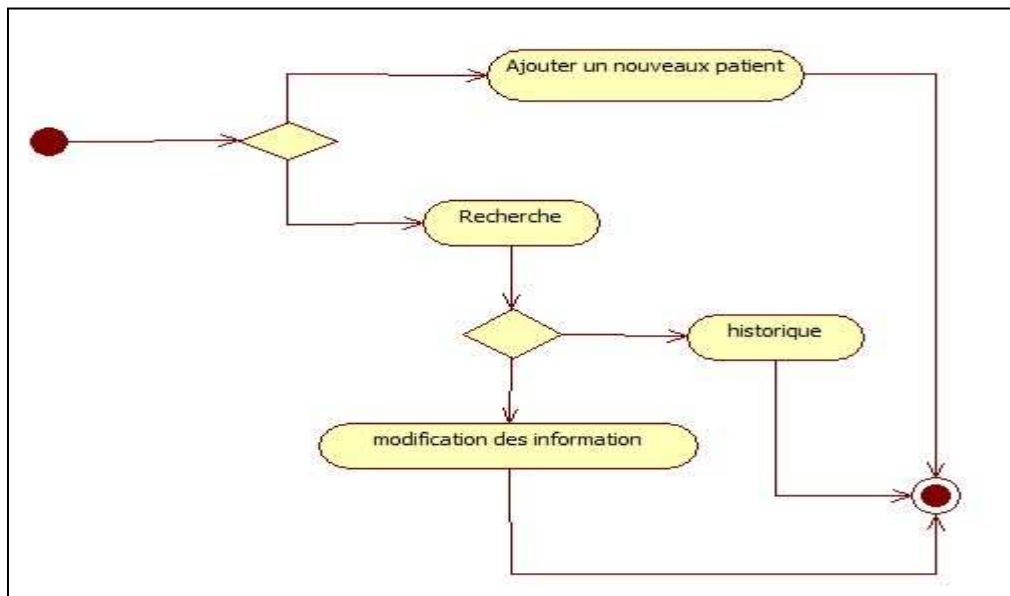


Figure. II.11: Diagramme d'activité «Suivi des dossiers».

c. Diagramme de séquence : " Ajouter un nouveaux patient "

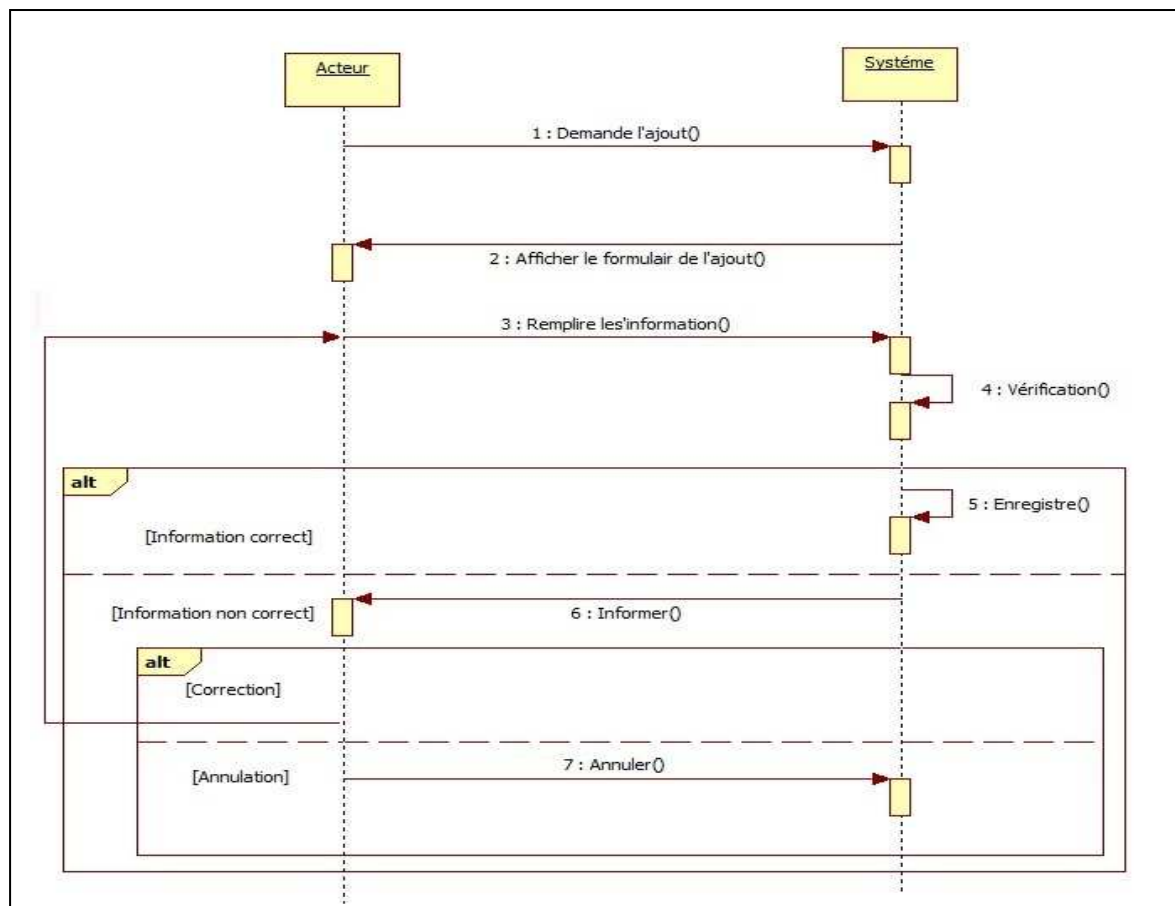


Figure. III.8: Diagramme de séquence «Ajouter un nouveaux patient».

d. Diagramme de séquence : « historique de patient. »

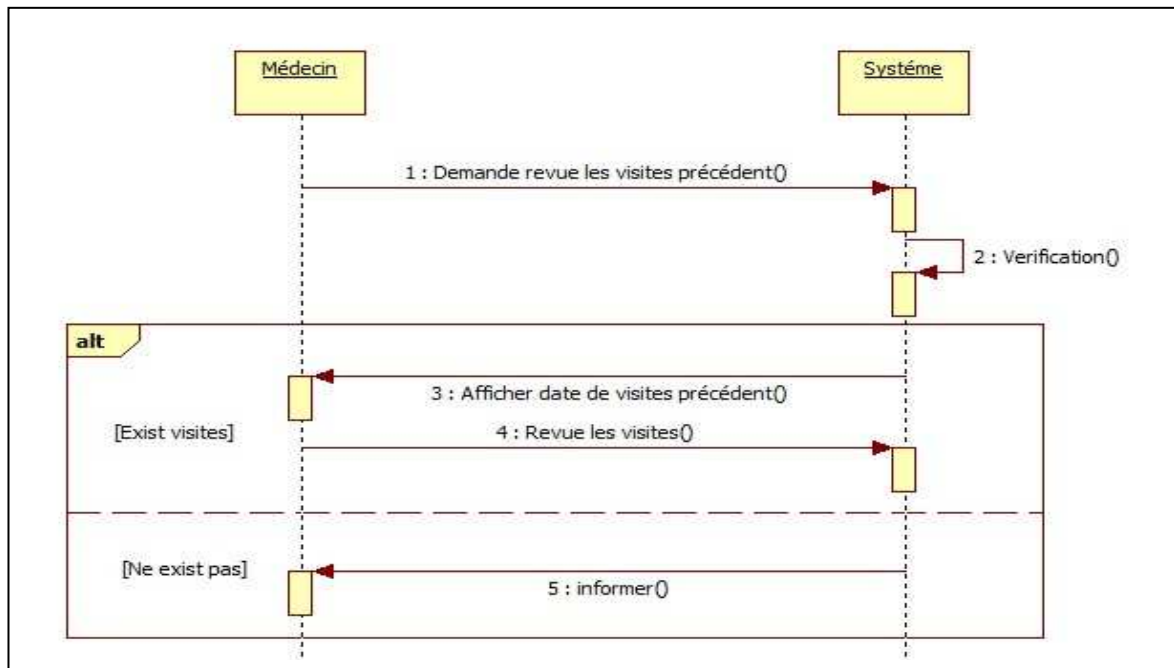


Figure. III.9: Diagramme de séquence «historique de patient».

e. Diagramme de séquence : « Modification des information des patients. »

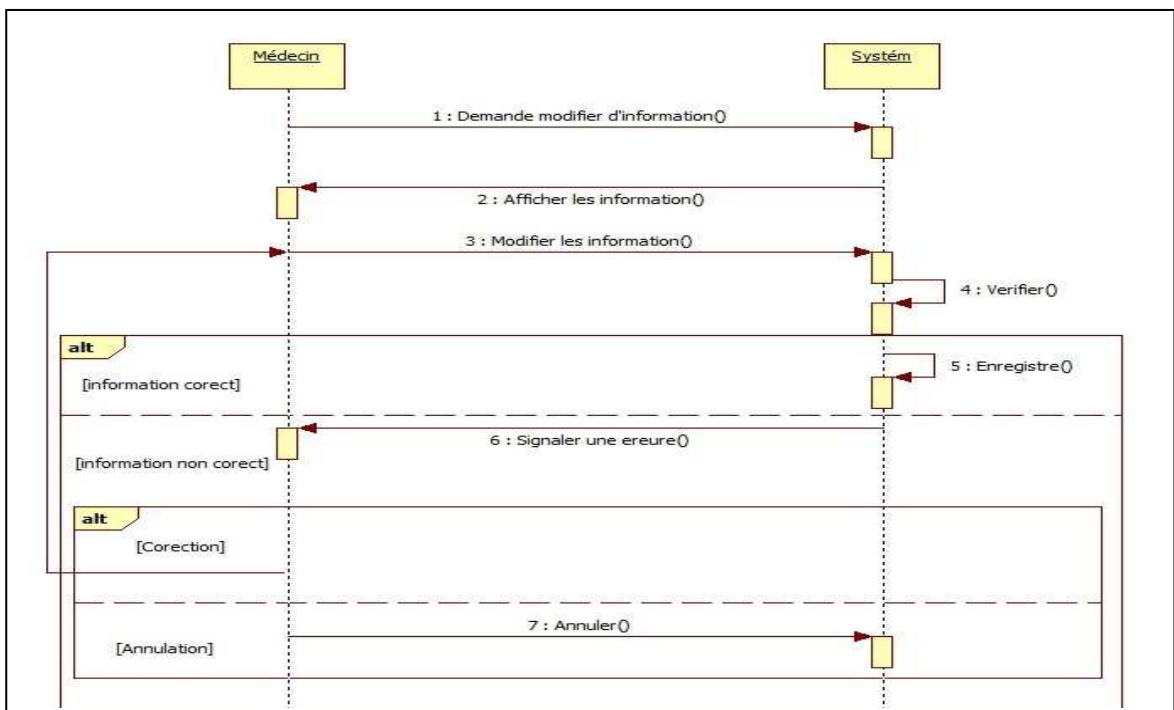


Figure. II.10: Diagramme de séquence «Modification des information des patients.».

➤ **Cas d'utilisation « Rechercher » :**

- Intention : Extraction les informations des patients.
- Actions :
 - Après l'authentification.
 - Saisir l'information de recherche (le numéro de patient, Nom de patient, date de visite).

a. **Diagramme de cas d'utilisation :**

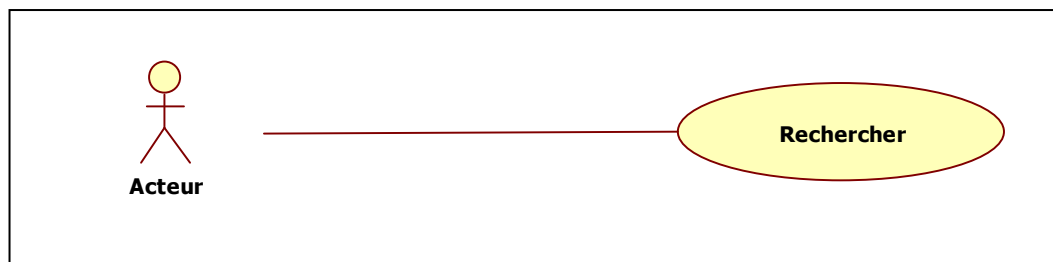


Figure. III.11: Diagramme de cas d'utilisation « Rechercher ».

b. **Diagramme d'activité :**

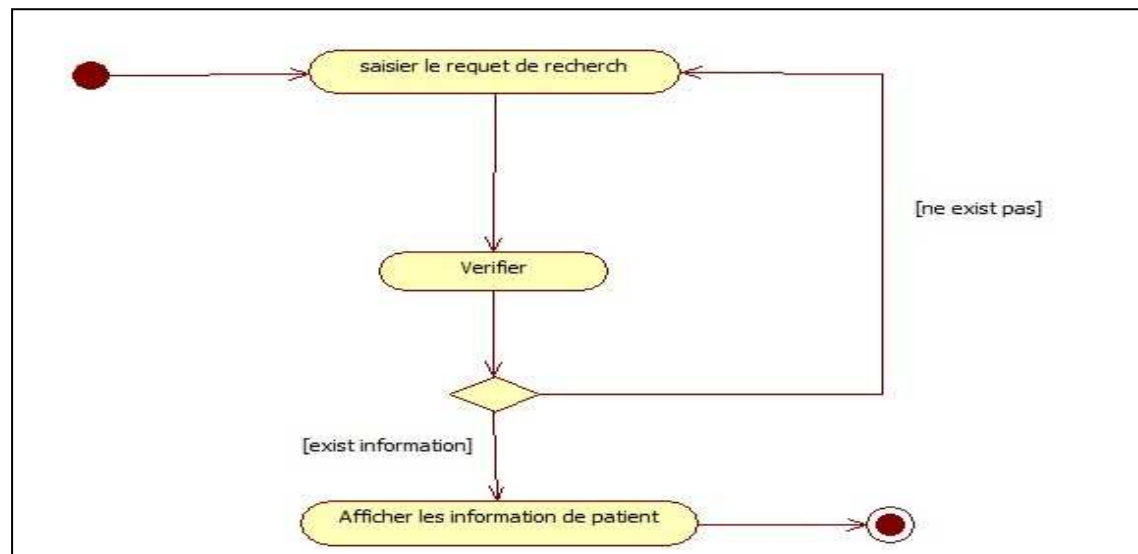


Figure. II.12: Diagramme d'activité « Rechercher ».

c. Diagramme de séquence : « Rechercher ».

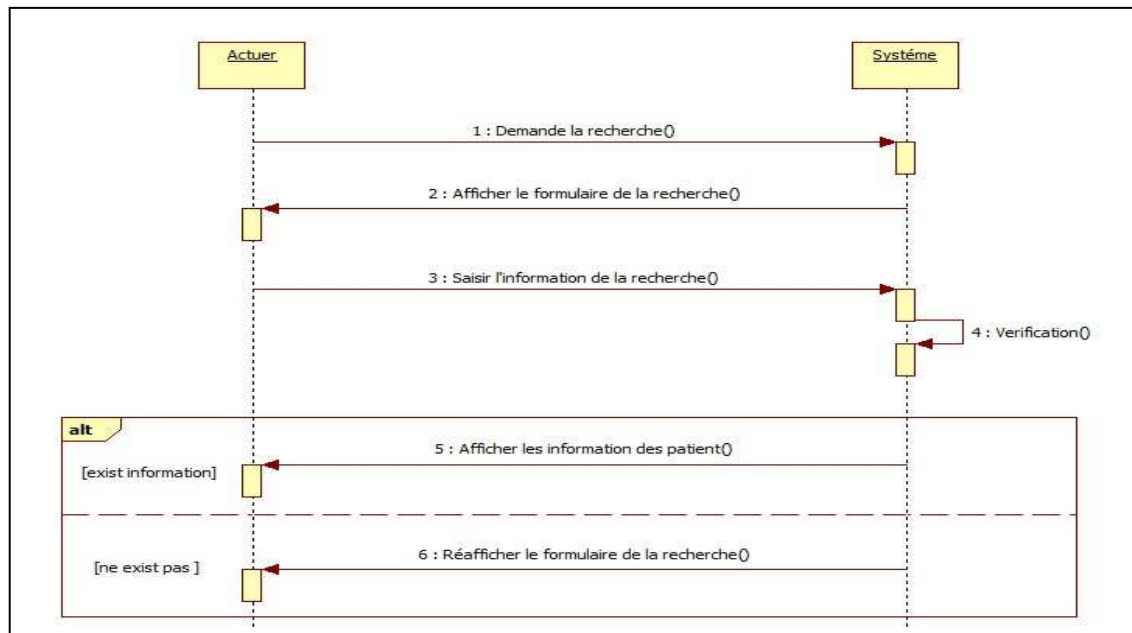


Figure.III.13: Diagramme de séquence « Rechercher ».

➤ Cas d'utilisation « Traitement de patient » :

- Intention : Traitement de patient et enregistre les informations et les documents de traitement (Ordonnance/ Facture).
- Actions :
 - o Après la recherche.
 - o Enregistre information de traitement.
 - o Crée (Ordonnance/ Facture)

a. Diagramme de cas d'utilisation :

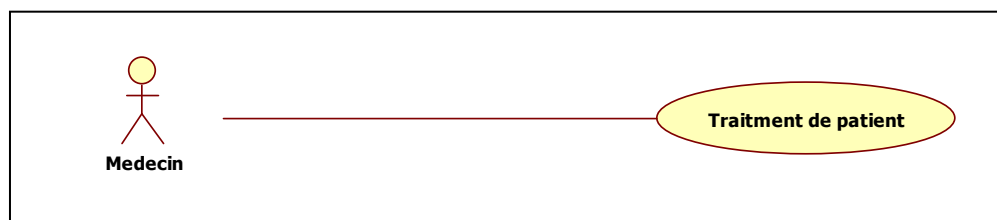


Figure. III.14: Diagramme de cas d'utilisation «Traitement de patient ».

b. Diagramme d'activité :

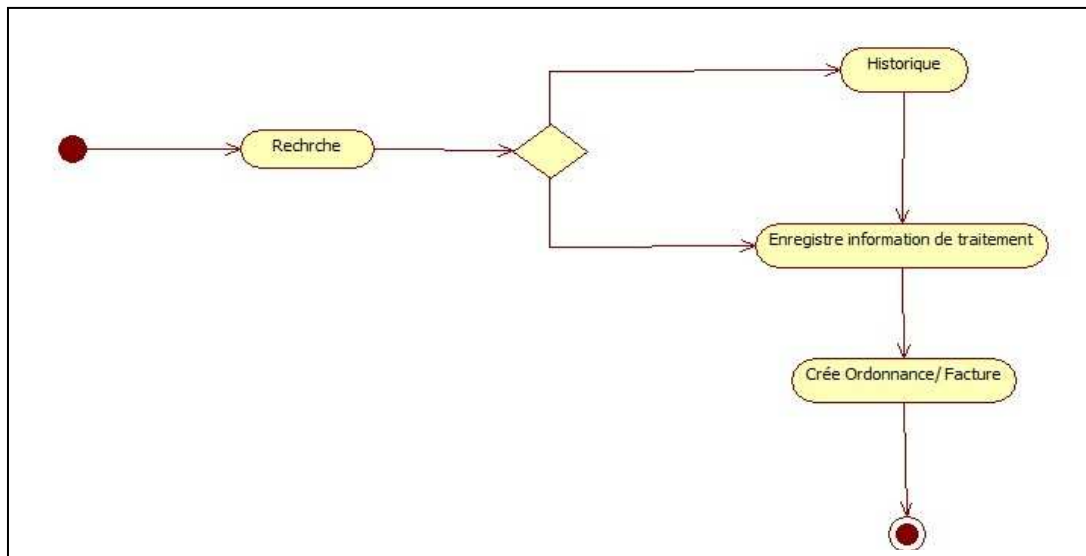


Figure. III.15: Diagramme d'activité «Traitement de patient ».

c. Diagramme de séquence: « enregistre les information de patient »

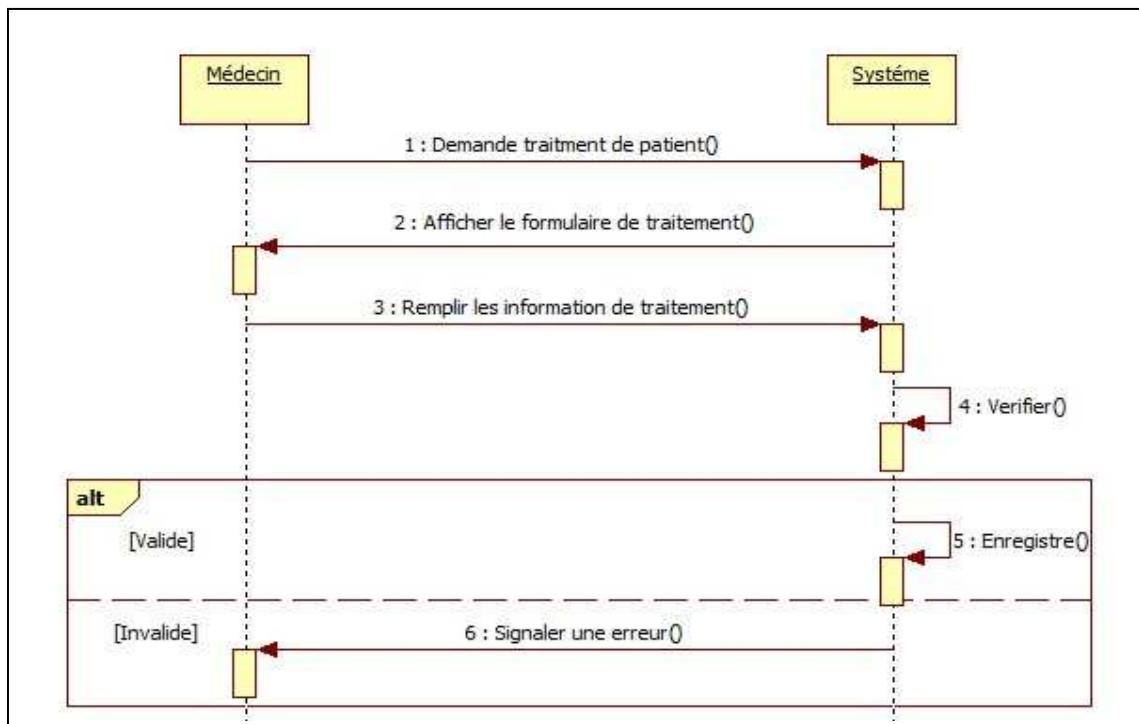


Figure.III.20: Diagramme de séquence « enregistre les information de patient » .

d. Diagramme de séquence: « Crée (Ordonnance/ Facture) »

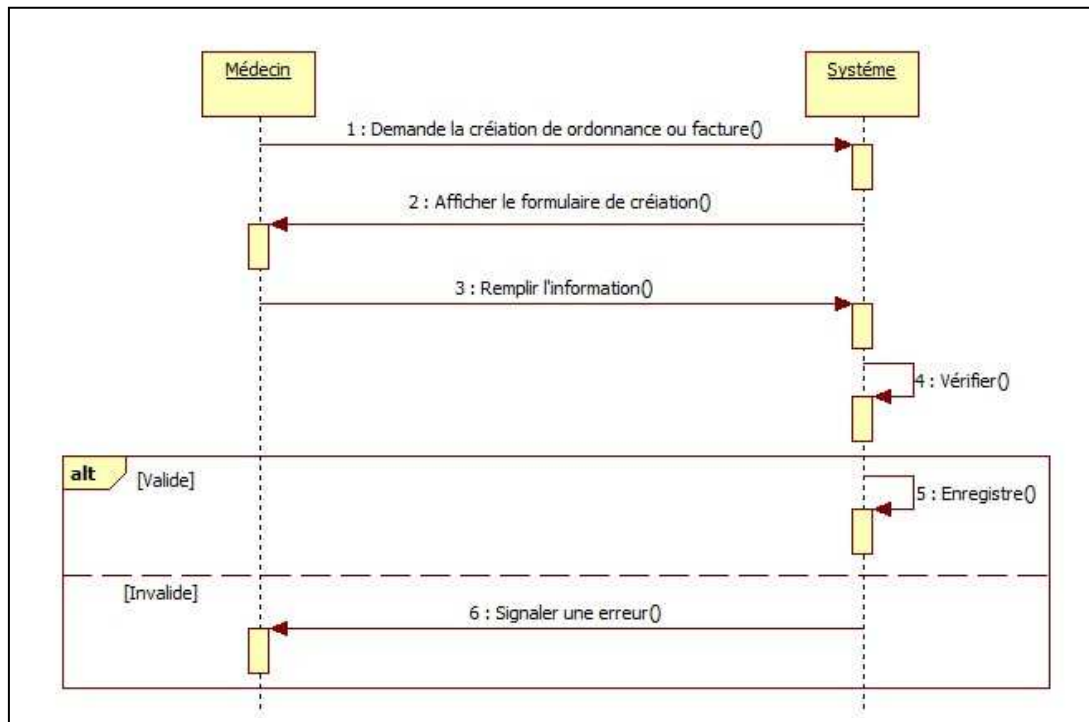


Figure.III.21: Diagramme de séquence « Crée (Ordonnance/ Facture) »

4. Capture de besoins techniques :

La capture des besoins techniques couvre, par complémentarité avec celle des besoins fonctionnels, toutes les contraintes qui ne traitent ni de la description du métier des utilisateurs, ni de la description applicative. Le modèle de spécification s'exprime suivant deux points de vue qui sont : la spécification logicielle et la structure du matériel à exploiter. [9]

5. Développement du modèle statique :

Cette étape nous permettra d'illustrer les principales constructions du diagramme de classes. Ces diagrammes de classes sommairement élaborés dans l'étape d'analyse de besoins fonctionnels seront détaillés, complétés et optimisés.

➤ **Diagramme de classe générale:**

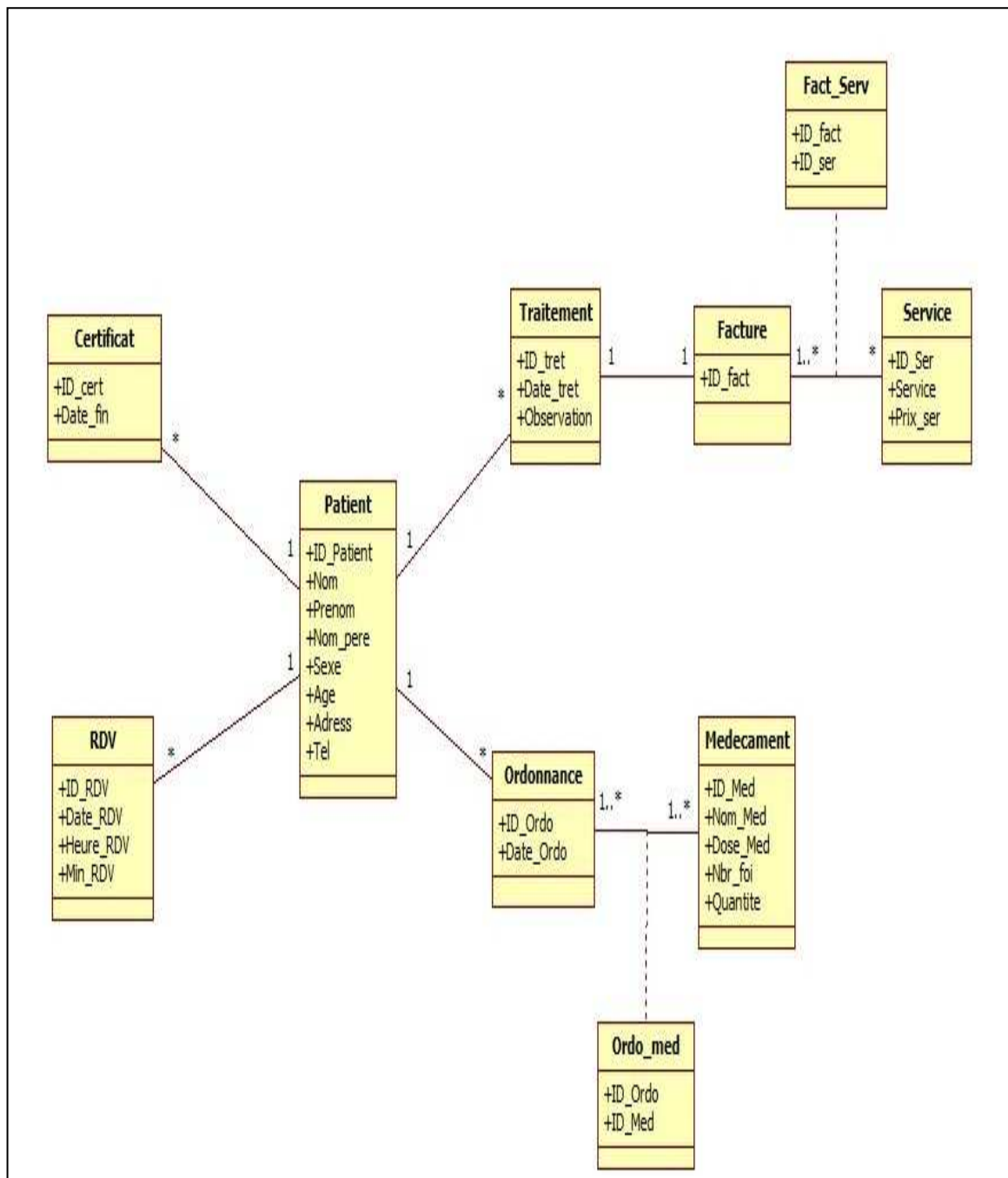


Figure.II.22: Diagramme de classe.

6. Conception détaillée:

La conception détaillée est la phase ultime de la modélisation qui consiste à construire et à documenter précisément les classes, les tables et les méthodes qui constituent le codage de la solution.

6.1. Description du modèle des classes:

Table. II.2: Liste des classes.

Classe	Attributs	Désignation	Type [Taille]
Patient	ID_Patient	Numéro de patient	N [10]
	Nom	Nom de patient	S [20]
	Prenom	Prénom de patient	S [20]
	Nom_Pere	Nom de père de patient	S [20]
	Sexe	Sexe de patient	S [10]
	Age	Age de patient	Int32
	Adresse	Adresse de patient	S [50]
	Tel	Numéro de téléphone de patient	S [10]
Certificat	ID_cert	Code de certificat	N
	Date_fin	Date de fin	D
RDV	ID_RDV	Numéro de rendez-vous	S [20]
	Date_RDV	La date de rendez-vous	D
	Heure_RDV	Le Heure de rendez-vous	S[4]
	Min_RDV	Le minute de rendez-vous	S[4]
Traitement	ID_Tret	Code de traitement	S [10]
	Date_Tret	Date de traitement	D
	Observation	Observation de médecin	S
Facture	ID_fact	Numéro de facture	N [6]
Service	ID_ser	Code service	S [20]

	Service	Service médicale	F
	Prix_Ser	le prix du service	
Ordonnance	ID_Ordo	Code de paient	N[6]
	Date_Ordo	Date d'édite l'ordonnance	D
Médicament	ID_Med	Code de médicament	N
	Nom_Med	Nom de médicament	S [20]
	Dose_Med	Dose de médicament	F
	Nbr_foi	Nombre de fois prenant le médicament	N
	Quantite	la quantité de médicament	N

6.2. Description du modèle d'association:

Table. II.3: Liste des associations.

Associations	Attributs	Désignation	Type [Taille]
Fact_Serv	ID_fact	Numéro de facture	N
	ID_ser	Code service	N
Ordo_Med	ID_Ordo	Code d'ordonnance	N
	ID_Med	Code de médicament	N

6.3. Passage au modèle relationnel :

L'utilisation des SGBDR impose un changement de représentation entre la structure des classes et la structure des données relationnelles. Les équivalences exprimées dans le tableau suivant sont généralement utilisées : [8]

Table.II.4: Equivalences entre les concepts objets et relationnels.

Modèle objet	Modèle relationnel
Classe	Table
Attribut de type simple	Colonne
Attribut de type complexe	Colonnes ou clé étrangère
Instance	T-uplet
OID	Clé primaire
Association	Clé étrangère ou table de liens
Héritage	Clé primaire identique sur plusieurs tables

❖ Listes des tables de la base de données :

Table.II.5: Liste des tables de la base de données.

Table	Identifiant	Attributs
Patient	ID_patient	ID_patient , Nom, Prenom, Nom_pere, Sexe, Age, Adresse, Tel
Certificat	ID_cert	ID_cert , Date_fin <u>ID_patient</u>
RDV	ID_RDV	ID_RDV , Date_RDV, Heur_RDV, Min_RDV <u>ID_patient</u>
Ordonnance	ID_Ordo	ID_Ordo , Date_Ordo <u>ID_patient</u>
Médicament	ID_Med	ID_Med , Nom_Med, Dose_Med, Nbr_foi, Quantite

		<u>ID_Ordo</u>
Traitement	ID_Tret	<u>ID_Tret</u> , Date_tret, Observation, <u>ID_patient</u>
Facture	ID_Fact	<u>ID_Fact</u> , <u>ID_patient</u>
Service	ID_Service	<u>ID_Service</u> , Service, prix_ser, <u>ID_Fact</u>

8. Conclusion

Dans le but de faciliter l'implémentation de notre système de gestion des cabinets médicaux, dans ce chapitre, nous avons présenté une capture des diagrammes utilisés lors de la modélisation suivie d'une démarche de développement. Les résultats de ce chapitre seront enrichis par des détails d'implémentation dans le chapitre suivant pour réaliser notre système.

Chapitre III

La réalisation du système

1. Introduction

Après avoir la conception de notre système dans le chapitre précédent et les différents diagrammes construits, dans ce chapitre on va détailler l'implémentation de notre application, on va voir également l'environnement de travail utilisé pour développer ce projet, puis nous présenterons les interfaces principales de notre logiciel.

2. Environnement de travail :

2.1. Langage de programmation:

Visual Studio 2010 est un environnement de programmation riche en outils comportant toutes les fonctionnalités nécessaires pour créer des projets C# de toute taille.

Le langage star de la nouvelle version de Visual Studio et de l'architecture .NET est C#, un langage dérivé du C++. Il reprend certaines caractéristiques des langages apparus ces dernières années et en particulier de Java. C# peut être utilisé pour créer, avec une facilité incomparable, des applications Windows et Web. C# devient le langage de prédilection d'ASP.NET qui permet de créer des pages Web dynamiques avec programmation côté serveur. [9]

❖ Les points forts de c#:

- libération automatique des objets.
- Le compilateur C#, comme Java, produit un p-code (bytecode) MSIL, Microsoft Intermediate Language, destiné à fonctionner dans une machine virtuelle (managed execution environment).
- C# a une syntaxe plus familière pour les programmeurs C, C++, Java et Perl.
- La syntaxe élégante et économique est empruntée à C et C++. - Grandes capacités d'introspection de classe (réflexion).

2.2..NET Framework:

Les programmes en C# s'exécutent sur le .NET Framework, composant intégral de Windows qui inclut un système d'exécution virtuel appelé Common Language Runtime (CLR) et un jeu unifié de bibliothèques de classes. Le CLR est l'implémentation commerciale de l'infrastructure du langage commun (CLI) de Microsoft, norme internationale constituant la base de toute création d'environnements d'exécution et de développement et assurant le fonctionnement homogène des langages et des bibliothèques. [10]

2.3. Microsoft Visual Studio.Net:

Visual Studio .NET est un jeu complet d'outils de développement permettant de générer des applications Web ASP, des services Web XML, des applications bureautiques et des applications mobiles. Visual Basic .NET, Visual C++ .NET, Visual C# .NET et Visual J# .NET utilisent tous le même environnement de développement intégré (IDE, integrated development environment), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du .NET Framework, qui fournit un accès à des technologies clés simplifiant le développement d'applications Web ASP et de services Web XML. [11]

2.4. SGBD

MS Access:

Microsoft Access (officiellement Microsoft Office Access) est un SGBD relationnel édité par Microsoft. Il fait partie de la suite bureautique MS Office Pro.

MS Access est composé de plusieurs programmes: le moteur de base de données Microsoft Jet, un éditeur graphique, une interface de type (Query by Example) pour manipuler les bases de données, et le langage de programmation (Visual Basic for Applications).

MS Access est un logiciel utilisant des fichiers au format Access (extension de fichier mdb pour Microsoft DataBase) (extension *.accdb depuis la version 2007).

Il est compatible avec les requêtes SQL (sous certaines restrictions) et dispose d'une interface graphique pour saisir les requêtes (QBE - Query by Example - « Requête par l'exemple »).

Il permet aussi de configurer, avec des assistants ou librement, des formulaires et sous-formulaires de saisie, des états imprimables (avec regroupements de données selon

divers critères et des totalisations, sous-totalisations, conditionnelles ou non), des pages html liées aux données d'une base, des macros et des modules VBA.[5]

3. Présentation du logiciel:

3.1 L'interface d'authentification :

Pour connecte au programme, l'utilisateur doit choisir type d'utilisateur puis saisir son mot de passe.

- Si le mot de passe correcte, il connecte au le programme.
- Sinon le système signale à l'erreur.

La figure ci-dessous représenté l'interface de l'authentification.



Figure.III.1: L'interface d'authentification.

3.2 Partie secrétaire

3.2.1 L'interface principale du partie secrétaire

La forme suivent représenter l'interface principale de secrétaire.

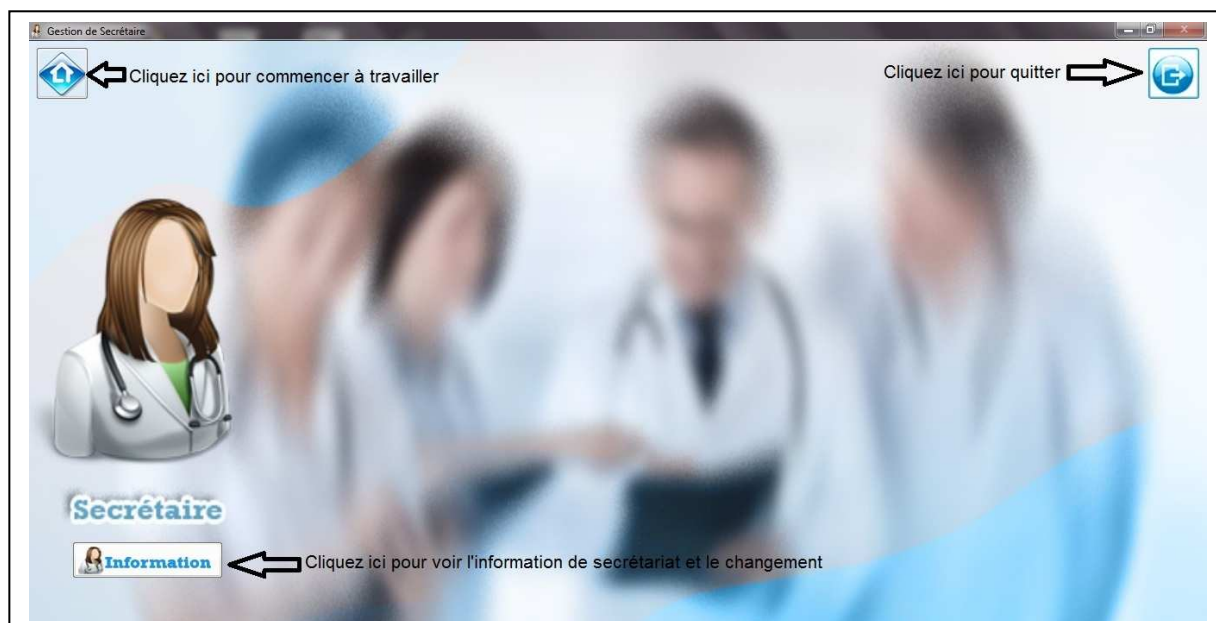


Figure.III.2: L'interface principale du partie secrétaire

3.2.2 Les rendez-vous d'aujourd'hui

Dans cette forme il a représentation des rendez-vous d'aujourd'hui

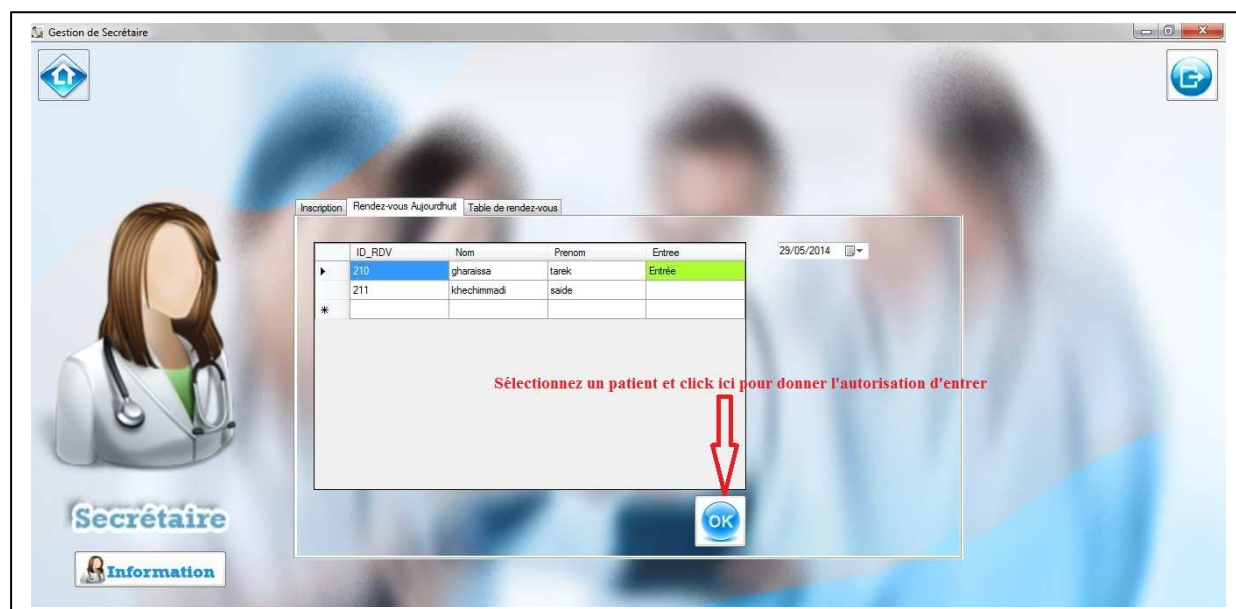


Figure.III.3: L'interface de rendez-vous d'aujourd'hui.

3.2.3 Tableau de rendez-vous

La forme suivante représente le tableau de rendez-vous de d'une semaine.

- Chaque jour, il ya huit heure de travail et chaque heure est divisée en quatre séance.

1- Si vous choisissez une séance non réserve, vous observez la forme de préservation.

2- Si vous choisissez une séance réserver, peu être supprimé un RDV ou non

N° :
 Nom :
 Prénom :
 Nom de père :
 Sexe :
 Age :
 Tel :
 Adresse :
 Date de rendez-vous : 2014- mai -30
 Heure : 8 h Min : 15
☒ Decocher pour recherch par nom

N° : 70
 Nom : lemouchi
 Prenom : nacer
 Tel : 06 6

Figure.III.4: Tableau de rendez-vous.

3.3 Partie médecin

3.3.1 L'interface de traitement

Dans cette forme représenter les rendez-vous d'aujourd'hui qui est avant le traitement et interface de traitement.

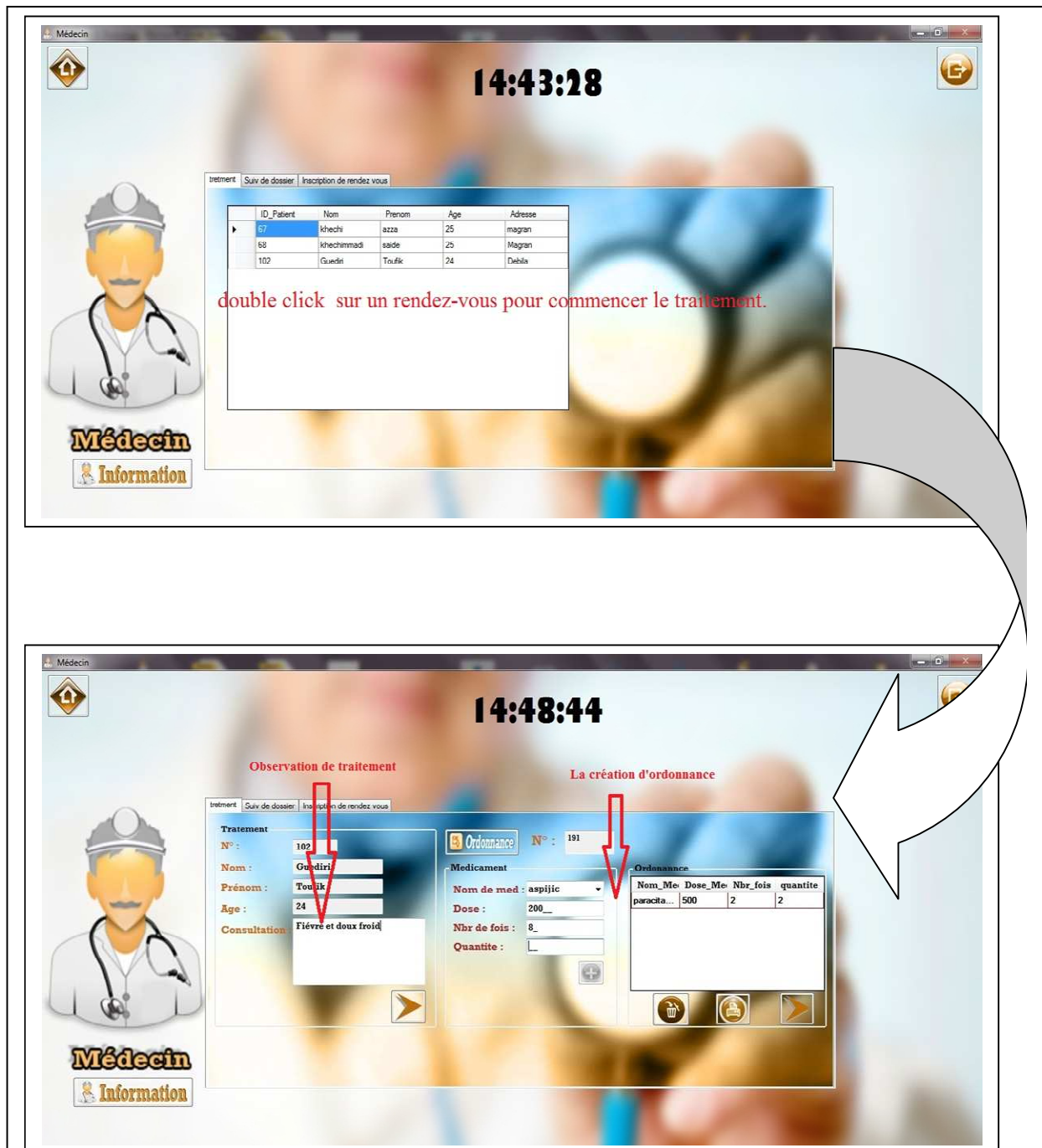


Figure.III.5: L'interface de traitement.

3.3.2 L'interface de suivi des dossiers

Dans ce cas il a la représentation d'interface de suivi des dossiers et l'opération qu'ils contiennent comme:

- Modification d'information de patient.
- Suppression d'un patient.
- Historique.

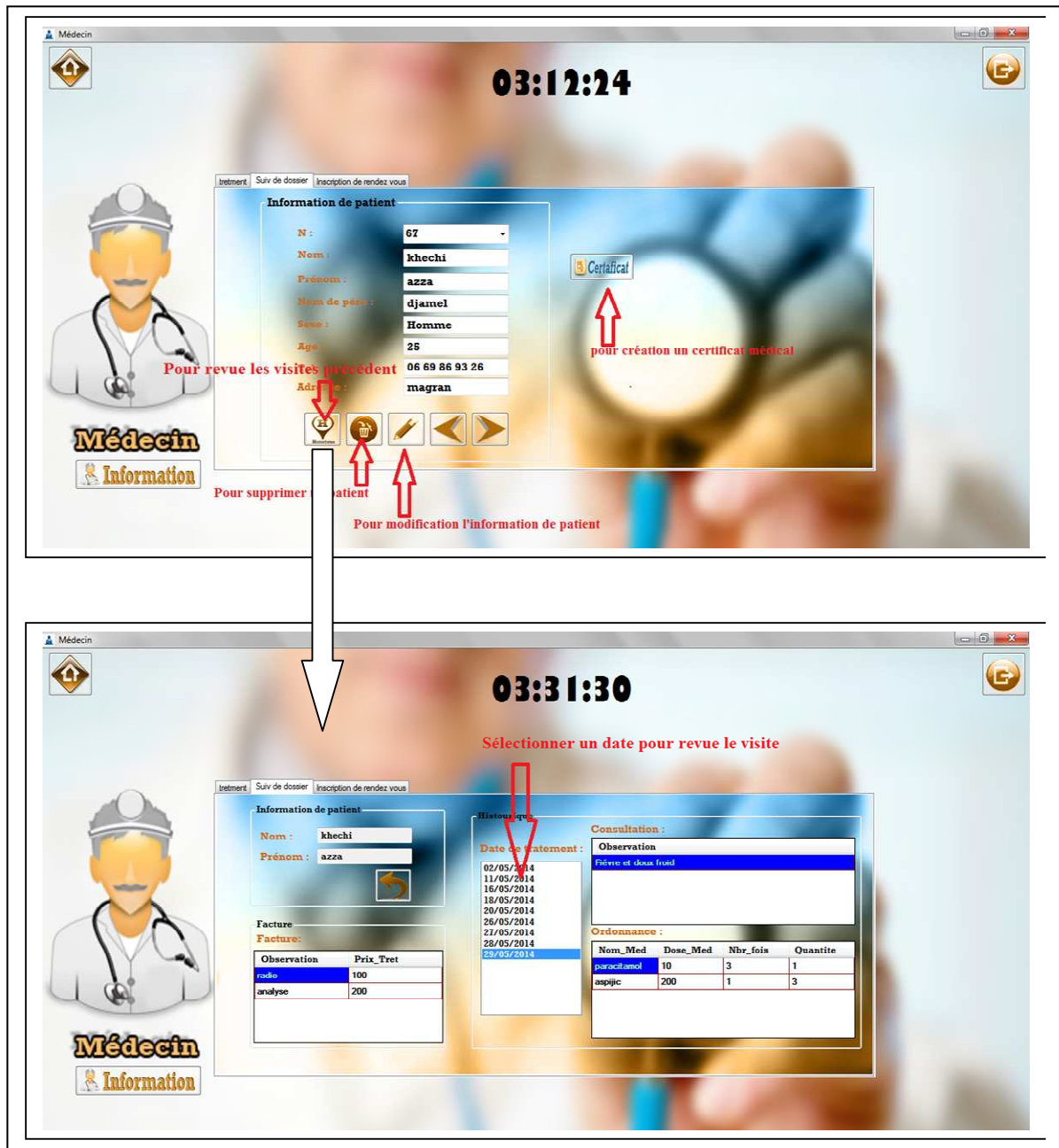


Figure.III.6: L'interface de suivi des dossier.

3.3.3 Modification l'information de cabinet.

Il existe dans la forme suivante l'interface de modification d'information de cabinet et dans ce cas nous pouvons modifier l'information de médecin et le cabinet.

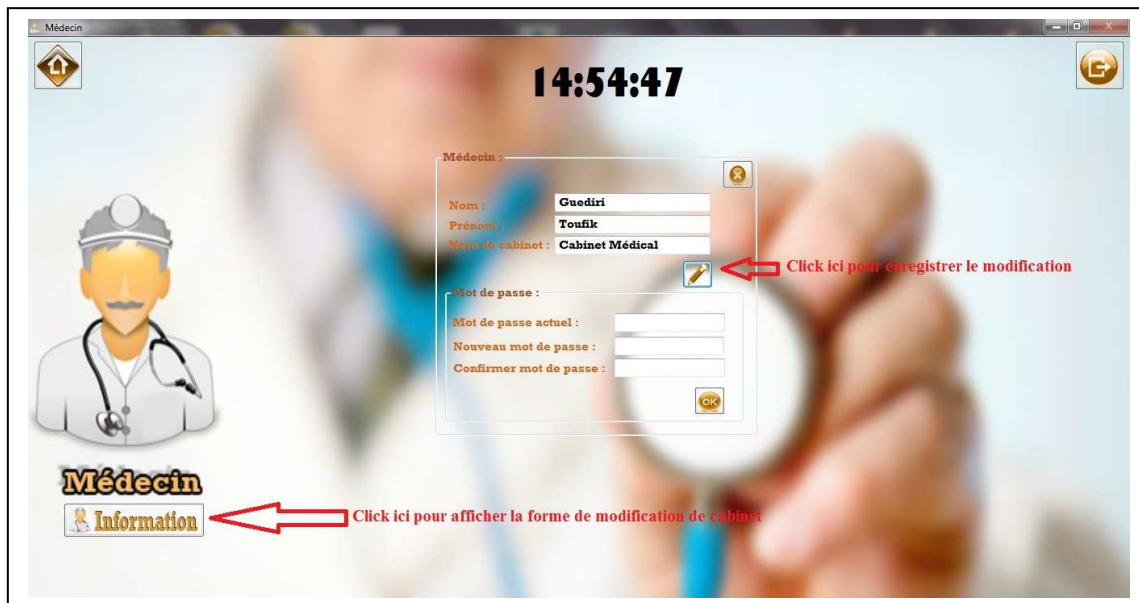


Figure.III.7: L'interface de modification l'information de cabinet..

4. Conclusion

Dans ce chapitre, nous avons présenté la réalisation de notre système et les différentes fonctionnalités ont été développées afin de faciliter l'accès et l'exploitation des données relatives à la gestion des opérateurs économiques.

Conclusion générale

L'objectif de notre travail était la construction d'une application de base de données pour la gestion de cabinet médical.

Nous avons commencé notre travail par la démarche à suivre (2TUP) et le langage de modélisation (UML). Dans le 2ème chapitre nous avons concevoir le nouveau système. Le dernier chapitre présent l'application du système étudié ou nous avons traduit notre modélisation conceptuelle en une implémentation.

Ce projet nous a été très bénéfique, car il nous permis de concrétiser nos connaissances théoriques issues des trois années d'études, de connaitre de près l'environnement de programmation et d'acquérir de nouvelles connaissances.

Bibliographies

[1]: Bali Ahmed /Cours UML/ Chapitre II/ 2013.

[2]: Laurent Piechocki / COURS UML / Document : version 13 / janvier 2003.

[4]: Eric Papet / 2UP_ARCHI_DEV1 / Document: version 1.00/12-12-2000.

[3]: http://fr.wikipedia.org/wiki/Microsoft_Access.

[5]: <ftp://ftp-developpez.com/bassim/ressources/memoire/rapport/02/04/2013>.

[6]: Eric PAPET/ la démarche TOW TRACK UNIFIED PROCESS/ Document:1.00/12/12/2000

[7]: Kazi Aouel Bassim et Rostane Zakaria/Suivie des enseignements du LMD par application de la méthode 2TUP/ Projet de Fin d'Etudes pour l'obtention du Diplôme d'Ingénieur d'Etat en Informatique / 08 -11- 2007.

[8]: Ait-Bachir Ali et Belkada Samir / Conception et réalisation d'un système d'information pour le suivi du parc informatique de B&R-C / Mémoire de fin d'étude pour l'obtention du diplôme d'ingénieur d'état en informatique / 2003-2004.

[9]: G.LebLANC/Livre C# et .NET version 2/ Edition EYROLLES/ 2006.

[10]: <http://msdn.microsoft.com/fr-fr/library/z1zx9t92%28VS.80%29.aspx>. 25/05/2013.

[11]: <http://msdn.microsoft.com/fr-fr/library/aa291755%28v=VS.71%29.aspx>. 25/05/2013,