الجمهورية الجزائرية الديموقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Minstère de l'Enseignement Supérieur et de la Recherche Scientifique

CENTRE UNIVERSITAIRE DE MILA
INSTITUT DES SCIENCES ET DE LA TECHNOLOGIE

Réf. ***/12***

Mémoire de fin d'étude

Présenté pour l'obtention du diplôme de

Licence Académique

Domaine : **Mathématiques et Informatique**

Filière : **Informatique**

Thème

Gestion d'un cabinet médical

Présenté par *:*

1- Boukemara Fadila

2- Houama kenza

Dirigé par *:*

- Abderrezek Samira

Année universitaire 2011-2012

Remerciement

C'est avec l'aide de Dieu qu'a vu les jours ce présent travail.

Ensuite, il n'aurait pas pu être achevé sans le soutien Les conseils

Les encouragements de certaines personnes auxquelles nous tenons ici à exprimer nos sincères remerciements.

En première lieu nous exprimons toute notre gratitude pour notre encadreur Abderrezek Samira pour ses précieux conseils

Ses disponibilités, la confiance qu'elle nous a toujours

Témoigné et la sollicitude dont elle nous a entouré, et ce tout au long de l'élaboration du présent travail.

Nous n'oublions pas non plus nos enseignants qui tout au long du cycle d'étude au centre universitaire de Mila, nous ont transmis leur savoir.

Nous tenons enfin à remercier tous ceux qui ont collaborés de prés ou de loin à l'élaboration de ce travail. Qu'ils acceptent nos humbles remerciements.

Houama kenza

Boukemara Fadila

DEDICACE

Au Début et avant tous,je veux remercier le dieu qui à permet le courage à faire et finir ce modeste travail.

C'est avec un grand plaisir et une réelle joie de fierté que je dédie ce travail.

A celle qui m'a soutenu durant tout mes années d'étude, qui mérité mon amour éternel pour ses conseils précieuse, sa tâchasse, sa patience ...A ma mère maria

A mon Père qui a fait de moi ce que je suis, j'espère que trouveras dans modeste travail toute la fierté que peut éprouver un père pour sa fille.

A ma sœur meriem et son marrie taher. sans oublier bien sur sa petite fille

Lina amale et manale.

A ma sœur nassira et son marri ali .sans oublier bien sur sa petite fille afafe et petite garsont adem.

A tous mes frères abdelmalek ,basseme , walid , abdelghani, mekhlouf

A ma meilleur amie pour la quelle je garde des bons souvenirs et celle qui m'a fait comprendre le vrai sens de l'amitié, et je partage maintenant ce modeste travail à:fadila .

A mes chère amies à la tête de liste: IMEN, Ouiam, FATIMA, LEILA (la femme de mon frère walid).

A la fin, nous remercions tous ceux qui ont aidé de prés ou de loin à réaliser notre travail

Houama kenza

الإهداء

إلى من لم يعرفا اليأس معي يوما، الى من ظنا أن فشلي بداية لأمل طويل" أمي" و"أبي"، إلى اخوتى و أخواتي الدين وقفوا معي ساعات فشلي و تعثري، إلى بنات و أولاد اخوتى، إلى صديقاتي، إلى أساتذتي الذين زرعوا فينا الإرادة و الاخلاص إلي الأستاذة" بوقارورة سامية" الى الأستاذ" كسيس محمد" إلى الأستاذ" دواس بلال" إلى" الأستادة "معوش فضيلة"...

بوكمارة فضيلة

Sommaire

Introduction générale
Introduction générale… ……..……15
Organisation du mémoire……..……..15
CHAPITRE 1 : Le langage de modélisation ……...16
1. Introduction……..……17
2. La modalisation……..……17
2.1 C'est quoi un modèle?.....17
2.2 Pourquoi modéliser ?.....17
3. Les concepts UML……..…18
3.1 Définition……..…18
3.2 Historique……..……18
4. Notions générales d'UML……..…19
4.1 Classeur…….……20
4.2 Paquetage……..…20
4.3 Stéréotype ……....20
4.4 Note ……..21
5. Les points forts d'UML……...…21
6. Les points faibles d'UML……..….21
7. Vues et diagrammes ……..….22
7.1. La vue statique(ou structurel)……...…22
7.2. La vue dynamique(ou comportementale)…….…22
7.3. Diagramme de cas d'utilisation…… 23
6.3.1Objectifs des cas d'utilisation …….……23
7.3.2.Éléments des diagrammes de cas d'utilisation ……..…23
7.3.2.1 Acteur……...23
7.3.3. Cas d'utilisation…….24
7.3.4. Relations dans les diagrammes de cas d'utilisation ……..24
7.3.5. Description textuelle des cas d'utilisation ……....24
7.4. Diagramme de classes ……25
7.4.1 Classe …… …26
7.4.2 Relations entre classes ……...27
7.4.2.1 Association ……..27
7.4.2.2 Multiplicité (cardinalité)……..27
7.4.2.3Les classes-association……..27
7.4.2.4 Agrégation et composition……..28
7.4.2.5. L'héritage …… ..28
7.4.2.6. Les interface ……...29
7.4.2.7. Package :……29
7.5 Diagramme d'objets ……..…29
7.6 Diagramme d'états-transitions ……29
7.7 Diagramme d'activités ……29

7.7.1. Présentation générale et concepts de base……...29
7.7.2 Diagramme de séquence ……30
7.8.2. Stéréotypes de Jacobson…….31
7.8.3. Les cadres d'interaction ……..31
8. Mise en œuvre d'UML ……....31
8.1 Identification des besoins et spécification des fonctionnalités ……32
8.1.1 Identification et représentation des besoins……32
8.1.2 Spécification détaillée des besoins : Diagrammes de séquence système…….33
8.1.3 Maquette de l'IHM de l'application (non couvert par UML) ……34
8.2 Phases d'analyse ……...…34
8.2.1 Analyse du domaine : modèle du domaine ……34
8.2.2 Diagramme de classes participantes……... 35
8.2.3 Diagrammes d'activités de navigation ……36
8.3 Phase de conception……...37
8.3.1 Diagramme d'interaction…….…37
8.4.2Diagramme de classes de conception ……..37
9. Conclusion……38
Chapitre 2 Identification des besoin……..…39
Introduction ……..…40
2. Présentation du projet……..40
2.1 Présentation de l'organisme d'accueil……..…..40
3. Grandes choix techniques ……...….41
3.1 Recueil des besoins fonctionnels ……...…41
3.2 Recueil des besoins opérationnels……..42
3.3.2 La rapidité de traitement…….…42
3.3.3 La convivialité……..…42
3.3 Identification d'acteurs……..….42
4. Diagramme de cas d'utilisation……44
5. Description des cas d'utilisation…….…..45
5.19 S'authentifier ……..45
5.1Ajout fiche de consultation……..…46
5.2. Modifier fiche de consultation……..…46
5.3 Supprimer la fiche de consultation ……47
5.5 Ajout patient……...…47
5.6 Supprime patient ……48
5.7 Modifier la fiche de patient ……48
5.9 Ajout un RDV ……49
5.10 Supprimer un RDV……...…49
5.11 Modifier un RDV……50
5.12 Établir la facture……51
5.13 Etablir l'analyse …….....52
5.15 Etablir ordonnance ……..…53
5.16 RDV en cours ……..…52

5.18 Etablir certificat ……...…54
6. Digrammes de séquence système……...……...…55
6.1 S'authentifier ……..……55
6.2 Ajouter patient ……...55
6.3 Modifier patient ……56
6.4 Supprimer fiche patient ……56
6.5 Crée fiche de consultation ……57
6.6 Modifier fiche de consultation ……..…57
6.8 Supprimer fiche consultation……..…58
6.9 Ajouter RDV……..…58
6.10 Supprimer RDV……....…59
6.11 Modifier RDV……..59
6.12 Etablir certificat ……...…60
6.13 Etablir l'analyse……61
6.14 Etablir ordonnance……..…61
6.15 RDV en cours ……62
7.Conclusion……..…62
Chapitre3… Phase d'analyse……..….63
1. Introduction……..…64
2. Démarche……..64
3. Identification des concepts de domaine……64
4. Modèle de domaine ……..….67
5. Les diagrammes de classes participantes des cas d'utilisations……....68
5.1 S'authentifier……..68
5.2 Créer fiche de consultation……68
5.3 Supprimer fiche de consultation……..…69
5.4 Modifier fiche de consultation……70
5.5 Créer fiche du patient……..70
5.6 Modifier fiche de patient……71
5.7 Supprimer fiche patient……....……71
5.8 Établir facture……...…72
5.9 Établir l'analyse……...…72
5.10 Établir ordonnance……73
5.11 Ajouter RDV……73
5.12 Supprimer RDV……..…..74
5.13 Modifier RDV……..….74
5.14 RDV en cours……...…….75
6. Les diagrammes d'activités de navigation……76
6.1 S'authentifier……..…..76
6.2 Crée fiche de consultation……...77
6.3 Modifier fiche de consultation ……..78
6.4Supprime fiche de consultation……..…79
6.5Ajout patient……80

6.6 Supprime patient……..81
6.7 Modifier patient……..…82
6.8 Établir ordonnance……83
6.9 Établir analyse……84
6.10 Etablir certificat……85
6.11 Ajoute RDV……86
6.12 Supprime RDV……...87
6.13 Modifier RDV……88
7. conclusion……...88
Chapitre4 Phase de conception……..……..89
1. Introduction……90
2 Diagrammes d'interactions……..…90
2.1 S'authentifier…….……...90
2.2 Créer fiche de consultation……91
2.4 Modifier fiche de consultation ……..…92
2.5 Supprimer fiche de consultation……...……93
2.6 Créer patient……..94
2.7 modifier patient……..…95
2.8 supprimer patient……96
2.9 établir certificat……..97
2.10 établir l'analyse……98
2.11 établir la facture……..….99
2.12 Établir l'ordonnance……...100
2.13 supprimer un RDV ……....101
2.14 Modifier un RDV……...102
2.15 ajouter un RDV……...103
2.16 RDV en cours……104
3. Les diagrammes de classes participants des cas d'utilisations……105
3.1 S'authentifier……105
3.2 Créer fiche de consultation……...106
3.3 Supprimer fiche de consultation……...107
3.4 Modifier fiche de consultation……..…108
3.5 Créer fiche du patient……109
3.6 modifier fiche de patient……110
3.7 supprimer fiche patient……..……110
3.8 Établir facture……111
3.9 Établir l'analyse……111
3.10 Établir ordonnance……...112
3.11 Ajouter RDV……112
3.12 Supprimer RDV……...113
3.13 Modifier RDV……..113
3.14 RDV en cours……..114
4. conclusion……...114

Chapitre 5 Implémentation……115
1. Introduction……..…116
2. Implémentation……..…..116
2.1Environnement de développement de l'application……116
2.1.1 utilise Pacestar UML Diagrammer 6.02 ……..116
2.1.2 Utilise Delphi…….117
2.2 Les langages de programmation ……118
2.2.1 Pascal……..118
2.2.2 SQL server……...118
3. Conception de la base de données…….…..118
4. Les interfaces de l'application……..…120
4.1 Fenêtre d'authentification…….120
4.2 Fenêtre d'accueil……...121
4.3 Fenêtre modifier s'authentification……...121
4.4 Fenêtre gestion fiche de consultation……122
4.5 Gestion patient……...…123
4.5Etablir ordonnance…….123
4.6 Etablir certificat…….124
4.7 Etablir analyse……126
4.8 Gestion RDV……..127
6 .Conclusion ……...…127
Conclusion générale……..128
Conclusion générale……..129
Bibliographie…….130

Figure	Page
Chapitre 01	
Figure 1.1 image développeur UML	19
Figure 1.2 classeur	20
Figure1.3 paquetage	20
Figure1.4 stéréotype	20
Figure 1.5 note	21
Figure 1.6 les acteurs	23
Figure1.7 cas d'utilisation	24
Figure 1.8 Description textuelle des cas d'utilisation	25
Figure 1.9 diagramme de classe	26
Figure1.10 association entre classe	27
Figure 1.11 Représentation graphique d'association de type agrégation	28
Figure 1.12 Représentation graphique d'association de type composition	28
Figure 1.13 Représentation graphique de la relation l'héritage	28
Figure 1.14 Représentation graphique de la package	29
Figure 1.15 Diagramme d'activités	30
Figure 1.16 représentation de Lignes de vie	30
Figure 1.17 Représentation d'un message synchrone	30
Figure 1.18 Représentation d'un message asynchrone	31
Figure 1.19 Représentation messages de création et destruction	31
Figure 1.20 Les besoins sont modélisés par un diagramme de cas d'utilisation	32
Figure 1.21 Les diagrammes de séquence système illustrent la description textuelle des cas d'utilisation	33
Figure 1.22 Une maquette d'IHM facilite les discussions avec les futurs utilisateurs	34
Figure 1.23 La phase d'analyse du domaine permet d'élaborer la première version du diagramme de	35
Figure 2.24 Le diagramme de classes participantes effectue la jonction entre les cas d'utilisation, le modèle du domaine et les diagrammes de conception logicielle classes.	36
Figure 2.25 Les diagrammes d'activités de navigation représentent graphiquement l'activité de navigation dans l'IHM.	36
Figure 2.26 Les diagrammes d'interaction permettent d'attribuer précisément les responsabilités de comportement aux classes d'analyse.	37
Figure 2.27 Chaîne complète de la démarche de modélisation du besoin jusqu'au code.	38
Chapitre 02	
Figure 2.1 L'organigramme du cabinet médical	41
Figure 2.2 diagramme de cas d'utilisation	44
Figure 2.3. Diagramme de séquence système du cas d'utilisation « S'authentifier »	55
Figure 2.4. Diagramme de séquence système du cas d'utilisation « ajoute patient»	55
Figure 2.5. Diagramme de séquence système du cas d'utilisation « supprimer fiche de consultation »	56
Figure 2.6. Diagramme de séquence système du cas d'utilisation « crée fiche de consultation »	56

Figure2.7. Diagramme de séquence système du cas d'utilisation « modifier fiche ce consultation»	57
Figure 2.8. Diagramme de séquence système du cas d'utilisation « supprimer fiche consultation»	58
Figure 2.9. Diagramme de séquence système du cas d'utilisation « ajoute RDV »	58
Figure 2.10. Diagramme de séquence système du cas d'utilisation « supprimer RDV »	59
Figure 2.10. Diagramme de séquence système du cas d'utilisation « modifier RDV»	59
Figure 2.11. Diagramme de séquence système du cas d'utilisation « établir certificat»	60
Figure2.12. Diagramme de séquence système du cas d'utilisation « établir l'analyse»	61
Figure2.13. Diagramme de séquence système du cas d'utilisation « établir ordonnance »	61
Figure2.14. Diagramme de séquence système du cas d'utilisation « RDV en cours»	62
Chapitre 03 :	
Figure 3.1 Modèle de domaine	67
Figure 3.2 Diagramme de classe participation du cas d'utilisation<<s'authentification >>	68
Figure 3.3 Diagramme de classe participation du cas d'utilisation<<crée fiche consultation>>	68
Figure 3.4 Diagramme de classe participation du cas d'utilisation<<supprimer fiche consultation>>	69
Figure 3.5 Diagramme de classe participation du cas d'utilisation<<modifier fiche consultation>>	70
Figure 3.6 Diagramme de classe participation du cas d'utilisation<<crée patient >>	70
Figure 3.7 Diagramme de classe participation du cas d'utilisation<<modifier patient >>	71
Figure 3.8 Diagramme de classe participation du cas d'utilisation<<supprimer patient >>	71
Figure 3.9 Diagramme de classe participation du cas d'utilisation<<établir facteur >>	72
Figure 3.10 Diagramme de classe participation du cas d'utilisation<<établir l'analyse>>	72
Figure 3.11 Diagramme de classe participation du cas d'utilisation<<établir ordonnance>>	73
Figure 3.12 Diagramme de classe participation du cas d'utilisation<<ajout RDV>>	73
Figure 3.15 Diagramme de classe participation du cas d'utilisation<<supprimer RDV>>	74
Figure 3.16 Diagramme de classe participation du cas d'utilisation<<modifier RDV>>	74
Figure 3.17 Diagramme d'activité de navigation de cas d'utilisation <<s'authentification>>	76
Figure 3.18 Diagramme d'activité de navigation de cas d'utilisation<<crée fiche consultation>>	77

Figure 3.19 Diagramme d'activité de navigation de cas d'utilisation<<modifier fiche consultation>>	
Figure 3.20 Diagramme d'activité de navigation de cas d'utilisation<<supprimer fiche consultation>>	78
Figure 3.21 Diagramme d'activité de navigation de cas d'utilisation<<ajoute patient>>	79
Figure 3.22 Diagramme d'activité de navigation de cas d'utilisation<<supprimer patient>>	80
Figure 3.23 Diagramme d'activité de navigation de cas d'utilisation<<modifier patient>>	81
Figure 3.24 Diagramme d'activité de navigation de cas d'utilisation<<établir ordonnance>>	82
Figure 3.25 Diagramme d'activité de navigation de cas d'utilisation<<établir analyse>>	83
Figure 3.26 Diagramme d'activité de navigation de cas d'utilisation<<établir certificat>>	84
Figure 3.27 Diagramme d'activité de navigation de cas d'utilisation<<ajoute un RDV>>	85
Figure 3.28 Diagramme d'activité de navigation de cas d'utilisation<<supprimer un RDV>>	86
Figure 3.29 Diagramme d'activité de navigation de cas d'utilisation<<modifier un RDV>>	87
Chapitre 4 :	88
Figure 4.1 diagramme d'interactions du cas d'utilisation<<s'authentifier>>	90
Figure 4.2 diagramme d'interactions du cas d'utilisation <<crée fiche de consultation >>	91
Figure 4.3 diagramme d'interactions du cas d'utilisation <<modifier fiche consultation >>	92
Figure 4.4 diagramme d'interactions du cas d'utilisation <<supprimer fiche consultation>>	93
Figure 4.5 diagramme d'interactions du cas d'utilisation <<crée patient >>	95
Figure 4.6 diagramme d'interactions du cas d'utilisation <<modifier patient >>	96
Figure 4.7 diagramme d'interactions du cas d'utilisation <supprimer patient >	97
Figure 4.8 diagramme d'interactions du cas d'utilisation <<établir certificat >>	98
Figure 4.9 diagramme d'interactions du cas d'utilisation <<établir analyse>>	99
Figure 4.10 diagramme d'interactions du cas d'utilisation <<établir facteur>>	
Figure 4.11 diagramme d'interactions du cas d'utilisation <<établir ordonnance>>	100
Figure 4.12 diagramme d'interactions du cas d'utilisation <<supprimer RDV>>	101
Figure 4.13 diagramme d'interactions du cas d'utilisation <<modifier RDV>>	102
Figure 4.14 diagramme d'interactions du cas d'utilisation <<ajout RDV>>	103
Figure 4.15 Diagramme de classe participation du cas d'utilisation <<s'authentification>>	104
Figure 4.16 Diagramme de classe participation du cas d'utilisation<<crée fiche consultation>>	105 106
Figure 4.17 Diagramme de classe participation du cas d'utilisation<<supprimer fiche consultation>>	107 108
Figure 4.18 Diagramme de classe participation du cas d'utilisation<<modifier fiche consultation>>	109

Figure 4.19 Diagramme de classe participation du cas d'utilisation<<crée patient >>	110
Figure 4.20 Diagramme de classe participation du cas d'utilisation<<modifier patient >>	111
Figure 4.21 Diagramme de classe participation du cas d'utilisation<<supprimer patient >>	
Figure 4.22 Diagramme de classe participation du cas d'utilisation<<établir facteur >>	111
Figure 4.23 Diagramme de classe participation du cas d'utilisation<<établir l'analyse>>	
Figure 4.24 Diagramme de classe participation du cas d'utilisation<<établir ordonnance>>	112
Figure 4. 26 Diagramme de classe participation du cas d'utilisation<<ajout RDV>>	112
Figure 4. 27 Diagramme de classe participation du cas d'utilisation<<supprimer RDV>>	113
Figure 4. 28 Diagramme de classe participation du cas d'utilisation<<modifier RDV>>	113
Chapitre 05	116
Figure 5.1 pacestar UMLdiagrammer 6.02	117
Figure 5.2 interface Delphi	120
Figure 5.3 s'authentification	
Figure 5.4 Ecran Accueil	121
Figure 5.5 modifier s'authentification	121
Figure5.6 gestion fiche du consultation	122
Figure 5.7 gestion fiche patient	123
Figure 5.8 établir ordonnance	124
Figure 5.9 établir certificat	125
Figure 5.10 établir analyse	126
Figure 5.11 Gestion RDV	127

INTRODUCTION GÉNÉRALE

Introduction Général

Notre projet présenté dans ce mémoire concerne la gestion d'un cabinet médicale et en particulier la gestion du cabinet de médecine générale "HAYOUR NOREDDINE".

Durant notre stage au cabinet médical, nous avons dénombré les problèmes suivants **:** la recherche manuelle des informations, la perte de temps pour obtenir les résultats voulus ,le remplissage de documents se fait manuellement ainsi que le manque d'archivage de dossiers des patients et l'absence des moyens rapides de communication.

Ces problèmes ont conduit à proposer une solution pour informatiser la gestion du cabinet médicale. Ce qui sous entend de soulager les membres du personnel du cabinet dans leurs tâches administratives. Cela comprend la gestion des patients, des rendez vous, la gestion de consultation et des divers documents qu'un médecin peut fournir : Certificats médicales et ordonnances. Nous avons choisi de répondre à cette demande par une interface simple et ergonomique car le personnel d'un cabinet médical n'est pas forcément habitué par l'utilisation d'outils informatiques complexes. Tout en respectant cette contrainte, nous ne devons pas pour autant restreindre les possibilités du logiciel.

4- Organisation du mémoire :

Le mémoire commence par une introduction générale du problème en question , ensuite il est organisé en 5 chapitres :

1- Le 1erest consacré au langage de modélisation UML ainsi que la démarche que nous allons utiliser pour modéliser notre application.

2- Le 2^{ième} chapitre est consacré à la phase « Identification des besoins », il présente dans un premier temps le cahier de charges, ensuite il décrit les acteurs et les cas d'utilisation du future système.

3- Le 3^{ième} chapitre concerne la phase d'analyse, c'est-à-dire la construction du modèle du domaine, les diagrammes de classes participantes et les diagrammes d'activités de navigation.

4- Le 4^{ième} chapitre est conçu pour la phase de conception, c'est-à-dire la construction des diagrammes de séquences et de classes conception.

Le 5^{iéme} chapitre est consacré à la phase d'implémentation.

Et enfin, le mémoire se termine par une conclusion générale du travail présenté.

CHAPITRE 1

LE LANGAGE DE MODELISATION

1. Introduction :

La description de la programmation par objets a fait ressortir l'étude du travail conceptuel nécessaire : définition des classes, des attributs et des interfaces etc.

Pour programmer une application il ne convient pas de se lancer tète baisse dans l'écriture du code, il faut d'abord organiser ses idées, les documents, puis organiser en définissant les modules et les étapes de réalisation, c'est cette démarche antérieure a l'écriture que l'on appelle modélisation.

Dons notre projet on va utiliser UML comme un langage de modélisation parce qu'il est non seulement un outil de modélisation intéressant mais une norme qui s'impose en technologie à objets. Aussi il nous faut une démarche permettant de bien guider la modélisation avec UML , et c'est le cas du processus (AUDIBER).

2. La modélisation :

2.1 C'est quoi un modèle?

Un modèle est une représentation abstraite et simplifiée (qui exclut certains détails), d'une entité (phénomène, processus, système, etc.) du monde réel en vue de le décrire, de l'expliquer ou de le prévoir.
Un modèle permet de réduire la complexité d'un phénomène en éliminant les détails qui n'influencent pas son comportement de manière significative.
il reflète ce que le concepteur croit important pour la compréhension du phénomène modélisé.

2.2 Pourquoi modéliser ?

Modéliser un système avant sa réalisation permet de:

- **Faciliter la compréhension du système étudié :** Un modeler étui la complexité du système étudié.
- **Faciliter la communication entre les membres de l'équipe:** Un modèle est un langage commun, précis, qui est connu par tous les membres de l'équipe et il est donc, à ce titre, un vecteur privilégié pour communiquer .Cette communication est essentielle pour aboutir à une compréhension commune

aux différentes parties prenantes (notamment entre la maîtrise d'ouvrage et la maîtrise d'œuvre informatique.

- **Mieux répartir les tâches entre les membres de l'équipe.**
- **Réduire les coûts et les délais :** Par exemple, les plateformes de modélisation savent maintenant exploiter les modèles pour faire de la génération de code (au moins au niveau du squelette).
- **Faciliter la maintenance :** une fois mise en production, l'application va devoir être maintenue, probablement par une autre équipe qui n'est pas nécessairement celle ayant créée l'application.

3. Les concepts UML :

3.1. Définition :

UML (Unified Modeling Language) se définit comme un langage de modélisation graphique qui permet la spécification ,la représentation et la construction des composantes d'un système informatique .

3.2 .Historique :

- Les années 1980: Utilisation de méthodes adaptées à la programmation impérative (notamment Merise)
- Fin 80 / début 90 la programmation par objets prend de l'importance.
- Conséquence naturelle: mise en place de méthodes orientées objet.
  Plus de cinquante méthodes apparaissent entre 1990 et 1995:
  OOD : Object Oriented Design (Booch, 1991)
  HOOD: Hierarchical Object Oriented Design (Delatteet al. 1993)
  OOA: Object Oriented Analysis (Schlaer, Mellor, 1992)
  OOA/OOD: (Coad, Yourdon, 1991)
  OMT: Object Modeling Technique (Rumbaugh, 1991)
  OOSE: Object Oriented Software Engineering (Jacobson, 1992)
  OOM: Object Oriented Merise (Bouzeghoub , Rochfeld, 1993)
  Fusion (Coleman et al. 1994)

- Les méthodes proposées définissent des concepts assez proches avec des notations différentes.

- 1994: le consensus se Fai autour de trois méthodes:

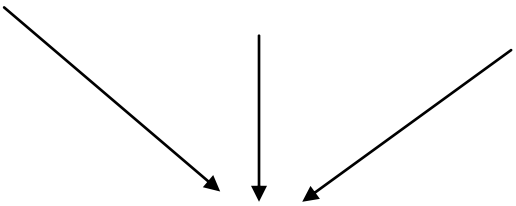
  -OMT (Object Modeling Technique de James Rumbaugh fournit une représentation graphique des aspects statiques, dynamiques et fonctionnels d'un système;

- OOD (Object Oriente Design) de Grad Booch, définie pour le Département of Defense, introduit le concept de paquetage (package);

  -OOSE (Object Oriented Software Engineering) d'Ivar Jacobson fonde l'analyse sur la description des besoins de utilisateurs (cas d'utilisation, ouusecases) .

- 1995 : Fusion des 3 principats les méthodes pour définir un langage de modélisation commun : UML (Unified Modeling Language)

James Rumbaugh Grad Booch, Ivar Jacobson



. Figure 1.1 image développeur UML

4. Notions générales d'UML :

Les éléments du langage UML que nous abordons ici ne sont pas spécifiques à un diagramme mais sont généraux.

4.1. Classeur :

*UN classeur précise les caractéristiques comportementales et structurelles d'un élément du modèle

*Un classeur est représenté n traits pleins et il peut éventuellement contenir des compartiments

Figure 1.2 classeur

4.2. Paquetage :

- Un paquetage est un regroupement cohérent d'éléments de diagrammes.

- Un paquetage se représente comme un dossier avec son nom inscrit dedans ou dans l'onglet .

Figure1.3 paquetage

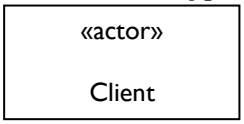
4.3. Stéréotype :

Un stéréotype est une annotation qui permet de créer de nouvel éléments à partir des éléments standard d 'UML.

Il est représenté par une chaînes de caractères entre guillemets («») dans , ou à proximité du symbole de l 'élément de modèle de base.

Par exemple UML utilise aussi le classeur pour représenter des éléments divers . La notation n'est cependant pas ambiguë grâce à l'utilisation du stéréotype.

Figure1.4 stéréotype



4.4 .Note :

Une note contient une information textuelle comme un commentaire . un corps de méthode ou une contrainte.

Les notes sont représentées par un rectangle avec le coin supérieur droit replié sur lui-même. On peut relier une note à un élément en utilisant une ligne pointillée

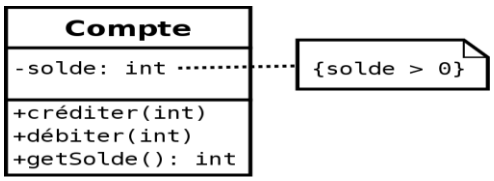


Figure 1.5 note

5. Les points forts d'UML :

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

  Il est un gain de précision

  Il est un gage de stabilité

  Il encourage l'utilisation d'outils

- **UML est un support de communication performant**

  Il cadre l'analyse.

  Il facilite la compréhension de représentations abstraites complexes.

  Son caractère polyvalent et sa souplesse en font un langage universel.

6. Les points faibles d'UML :

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

Même si l'Espéranto est une utopie, la nécessité de s'accorder sur des modes d'expression communs est vitale en informatique. 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 :

Or, l'intégration d'UML dans un processus n'est pas triviale et améliorer un processus est un tâche complexe et longue. Les auteurs d'UML sont tout à fait conscients de l'importance du processus, mais l'acceptabilité industrielle de la modélisation objet passe d'abord par la disponibilité d'un langage d'analyse objet performant et standard.

7. Vues et diagrammes :

Une vue est une description du système d'un point de vue donné. Chaque vue représente certains éléments des systèmes masque d'autres. Les différentes vues collaborent pour définir les systèmes complets.

UML2.0 comporte treize types de diagrammes qui représentent 2 vues différentes du système.

7.1. La vue statique(ou structurel):

Représente la structure statique du système en utilisant des objets, des attributs, des opérations et des relations.

Le point de vue structurel comprend les diagrammes:

- ✓ diagramme de classes (Class diagram)
- ✓ diagramme d'objets (Object diagram)
- ✓ diagramme de composants (Component diagram)
- ✓ diagramme de déploiement (Deployment diagram)
- ✓ diagramme de paquetages (Package diagram)
- ✓ diagramme de structures composites (Composite structure diagram)

7.2. La vue dynamique(ou comportementale):

Met l'accent sur le comportement dynamique du système en montrant la collaboration entre les objets et les modifications apporté à l'état interne des objets. Cette vue inclut les diagrammes

- ✓ diagramme de cas d'utilisation (Use case diagram)
- ✓ diagramme d'activités (Activity diagram)
- ✓ diagramme d'états-transitions (State machine diagram)

Diagrammes d'interaction (Interaction diagram):

- ✓ diagramme de séquence (Sequence diagram)

- ✓ diagramme de communication (Communication diagram)
- ✓ diagramme global d'interaction (Interactio overview diagram)
- ✓ diagramme de temps (Timin diagram).

Les13diagrammes ne sont pas nécessairement tous utilisés à l'occasion d'une modélisation.

Les plus utiles sont :

7.3. Diagramme de cas d'utilisation :

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

6.3.1Objectifs des cas d'utilisation :

Les cas d'utilisation sont une technique de description du système étudié privilégiant le point de vue de l'utilisateur.

Il s'agit de la solution UML pour représenter le modèle conceptuel.

Les cas d'utilisation décrivent sous la forme d'actions et de réactions, le comportement d'un système du point de vue d'un utilisateur. Les cas d'utilisation servent à structurer les besoins des utilisateurs et les objectifs correspondants du système.

Un cas d'utilisation est une manière spécifique d'utiliser un système. C'est l'image d'une fonctionnalité du système, déclenchée en réponse à la stimulation d'un acteur externe

7.3.2. Éléments des diagrammes de cas d'utilisation :

7.3.2.1 Acteur :

Un acteur est le rôle joué par une entité externe (personne, un autre système,…) qui interagit avec le système.

Un acteur se représente par un petit bonhomme avec son rôle inscrit dessous.

Il est également possible de représenter un acteur sous la forme d'un classeur stéréotypé *«actor»*.

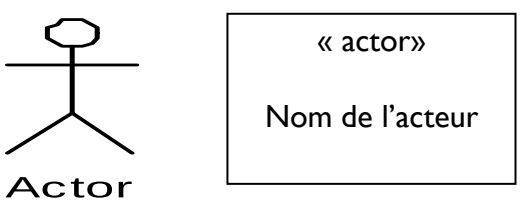


Figure 1.6 les acteurs

7.3.3. Cas d'utilisation :

Un cas d'utilisation («*usecase*») représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résulta observable intéressant pour un acteur particulier

Un cas d'utilisation se représente par une ellipse contenant le nom du cas (un verbe à l'infinitif éventuellement suivi par un complément) et se plaçant du point de vue de l'acteur et non pas de celui du système

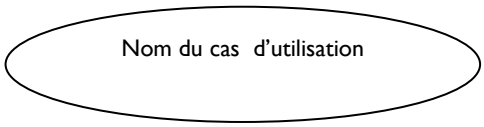


Figure1.7 cas d'utilisation

7.3.4. Relations dans les diagrammes de cas d'utilisation :

Les relations possibles entre cas d'utilisation : UML définit trois types de relations standardisées entre cas d'utilisation:- Une relation d'extension : formalisée par la dépendance « extend »

-Une relation d'inclusion : formalisée par la dépendance « include »

- Une relation de généralisation / spécialisation

7.3.5. Description textuelle des cas d'utilisation :

Le diagramme de cas d'utilisation d écrit les grandes fonctions d'un système du point de vue des acteurs.

Le diagramme de cas d'utilisation n'expose pas de façon détaillée le dialogue entre les acteurs et les cas d'utilisation.

Il est recommandé de rédiger une description textuel du cas d'utilisation avant de détailler cette description par des diagrammes d'UML

Une description textuelle couramment utilisée se fait comme suit:

Première partie: permet d'identifier le cas d'utilisation, elle doit contenir les informations qui suivante:

Cas d'utilisation	Nom du cas d'utilisation (verbe à l'infinitif+complément)
Objectif	L'objectif du cas d'utilisation (ce qu'il offre aux acteurs)
Acteurs	Liste des acteurs qui interviennent pour le cas d'utilisation
Responsable	Le nom du responsable (rédacteur du cas d'utilisation)
Dates	Dates de créations et de mise à jour de la description courante
Version	Le numéro de version

La deuxième partie : contient la description du fonctionnement du cas d'utilisation

Pré-conditions	Elles décrivent dans quel état doit être le système (l'application) avant que ce cas d'utilisation puisse être déclenché.
Post-conditions	Elles décrivent l'état du système à l'issue du cas d'utilisation. Les
	Les scénarios -Un scénario décrit une exécution particulière d'un cas d'utilisation du début à la fin. -Un scénario est décrit sous la forme d'échanges d'évènements entre l'acteur et le système
Scénario nominal	Il réalise les post-conditions du cas d'utilisation, d'une façon naturelle et fréquente.
Scénarios alternatives	Ils remplissent les post-conditions du cas d'utilisation, mais en empruntant des voies Indirectes.
Scénarios d'erreurs	Ne réalisent pas les post-conditions du cas d'utilisation

Figure 1.8 Description textuelle des cas d'utilisation

7.4. Diagramme de classes :

Le diagramme de classes est un diagramme UML qui permet de modéliser les classes du système et leurs relations (association, généralisation,…).

Il fournit une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les fonctionnalités du système.

Le diagramme de classes est considéré comme le plus important et le plus utilisé de la modélisation orientée objet.

Le diagramme de classes présentes la vue statique du système n'entretient pas compte des aspects dynamiques.

7.4.1. Classe :

Une classe est la description abstraite d'un ensemble d'objets possédant les mêmes caractéristiques..

Une classe est représentée par un classeur divisé en trois compartiments :le premier indique le nom de la classe ,le deuxième ses attributs et le troisième ses opérations.

- **Nom d'une classe :**

Le nom de la classe doit évoquer le concept décrit par la classe .Il commence par une majuscule.

- **Les attributs :**

La syntaxe d'un attribut est la suivante:

<visibilité> <nom Attribut> : <type> [= < valeur_ par _défaut >]

Visibilité : indiques il' attribut est public, protégé ou privé

Nom Attribut : Le nom donné à l'attribut.

Type : Type de l'attribut

Valeur _par _défaut : Valeur par défaut associé à un attribut d'un objet 'il n'ya pas de valeur spécifiée lors de sa création

- **Les opérations**

La syntaxe d'une opération est la suivante:

<visibilité><nom _ opération> ([<paramètre_1>,..., < parameter _ N>]) :[<type _renvoyé>]

Visibilité: public(+), protégé(#) ou privé(-)

Nom _ opération : Le nom donné à l'opération.

<paramètre_1>,..., < paramètre _N>: Liste des paramètres de l'opération.

Type_ renvoyé : type de la valeur retournée par l'opération

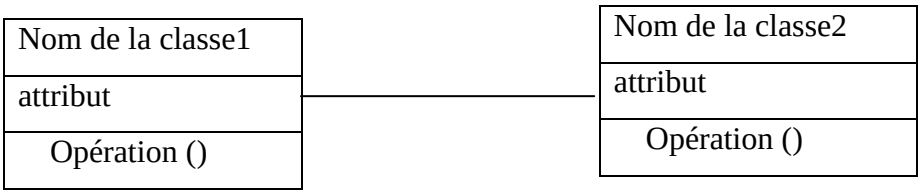


Figure 1.9 diagramme de classe

7.4.2 Relations entre classes :

7.4.2.1. Association :

Une association est une relation entre deux classes qui indique que ’il peut y avoir des liens entre des instances des classes associées .

Une association est représentée par une ligne continue entre deux classes.

Figure1.10 association entre classe

7.4.2.2. Multiplicité (cardinalité) :

Aux deux extrémités d’une association, on doit faire figure une indication de multiplicité.

La multiplicité spécifie le nombre d’objets qui peuvent participer à une relation avec un objet de l’autre classe dans le cadre d’une association.

Un et un seul	1 ou 1..1
Zéro ou un	0..1
plusieurs	* ou 0..*
De 1 à plusieurs	1..*
De M à N (entiers naturels)	M..N

7.4.2.3. Les classes-association *:*

Les attributs d’une classe dépendent de l’identifiant de la classe.

Parfois, un attribut dépend de 2 identifiants, appartenant à 2 classes différentes .On va donc placer l’attribut dans l’association entre les deux classes.

Dans ce cas, l’association est dite «porteuse d’attributs».Une association porteuse d’attributs est appelée classe-association.

7.4.2.4 .Agrégation et composition :

- Une agrégation est une forme particulière d'association ou un tout (appelé classe agrégat) est relié à ses parties (appelé classe agrégées).

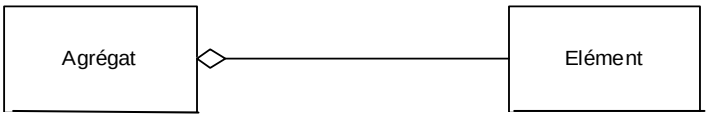


Figure 1.11 Représentation graphique d'association de type agrégation

- La composition est un cas particulier de l'agrégation des parties).

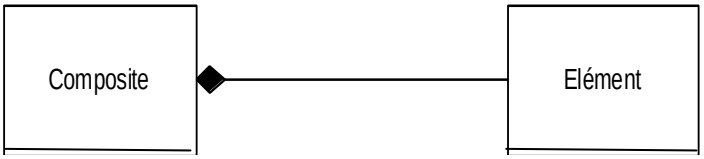


Figure 1.12 Représentation graphique d'association de type composition

7.4.2.5. L'héritage :

L'héritage décrit une relation entre une classe générale (classe de base ou classe parent) et une classe spécialisée (sous-classe).

La relation d'héritage est représentée par une flèche avec un trait plein dont la pointe est un triangle fermé désignant le cas le plus général.

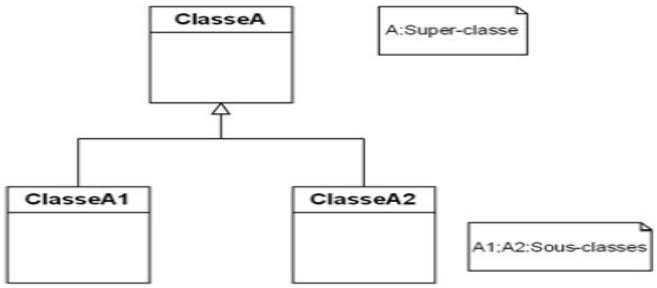


Figure 1.13 Représentation graphique de la relation l'héritage

7.4.2.6. Les interfaces :

Une interface est un ensemble de constantes et de déclarations de méthodes. Une interface est représenté par un classeur stéréotypé «interface».

7.4.2.7. Package :

Un package est un mécanisme général de regroupement d'éléments en UML, qui est utilisé principalement pour regrouper des classes.

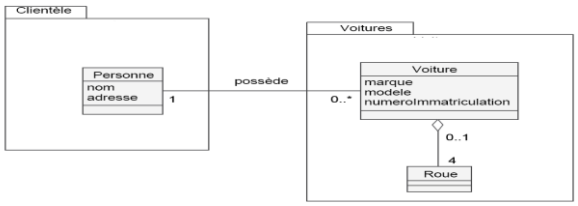


Figure 1.14 Représentation graphique de la package

7.5 Diagramme d'objets :

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.

7.6 Diagramme d'états-transitions :

Le diagramme d'états-transitions représente la façon dont évoluent 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.

7.7Diagramme d'activités :

7.7.1. Présentation générale et concepts de base

Le diagramme d'activité présente un certain nombre de points communs avec le diagramme d'état-transition puisqu'il concerne le comportement interne des opérations ou des cas d'utilisation. Cependant le comportement visé ici s'applique aux flots de contrôle et aux flots de données propres à un ensemble d'activités et non plus relativement à une seule classe .Les concepts communs ou très proches entre le diagramme d'activité et le diagramme d'état-transition sont :

- transition,
- ● nœud initial (état initial),
- ◉ nœud final (état final),
- 0 nœud de fin flot (état de sortie),
- ◊ nœud de décision (choix).

Le formalisme reste identique pour ces nœuds de contrôle.

Les concepts spécifiques au diagramme d'activité sont :

- nœud de bifurcation,
- nœud de jonction,
- nœud de fusion,
- pin d'entrée et de sortie,
- flot d'objet,
- partition.

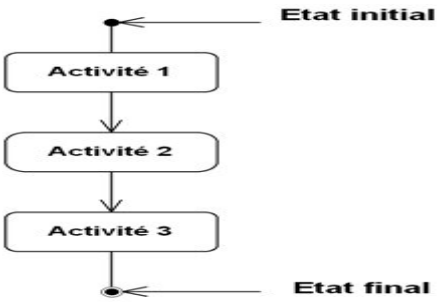


Figure 1.15 Diagramme d'activités

7.8 Diagramme de séquence :

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et les objets du système selon un ordre chronologique.

7.8.1.Eléments du diagramme de séquence :

a. Lignes de vie :

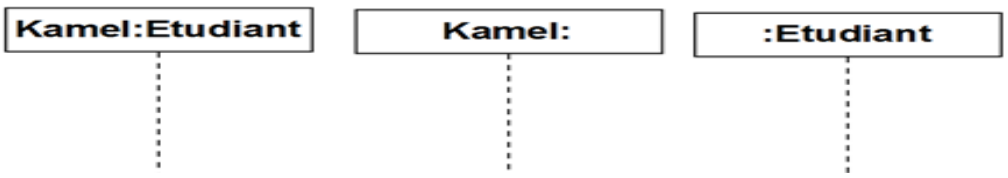


Figure 1.16 représentation de Lignes de vie

b. Les messages :

Plusieurs types de messages existent, les plus communs sont :

- Messages synchrones;

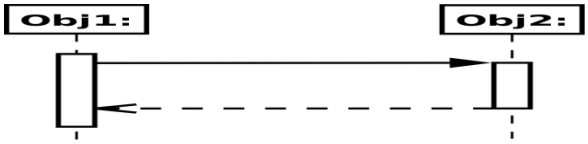


Figure 1.17 Représentation d'un message synchrone

- Messages asynchrones

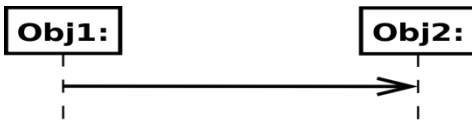


Figure 1.18 Représentation d'un message asynchrone

- Messages de création et destruction d'instance.

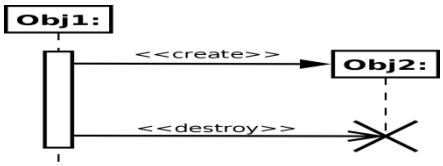


Figure 1.19 Représentation messages de création et destruction d'instance

7.8.2. Stéréotypes de Jacobson :

À l'intérieur du système, Jacobson distingue les trois types de classes

- Les classes de dialogue << boundary>>
- Les classes entités <<entity>>
- Les classes de contrôles <<control>>

7.8.3. Les cadres d'interaction :

Les cadres d 'interaction (Fragments combinés) permettent de d écrire des diagrammes de séquence de manière compacte.

Elles permettent par exemple de représenter graphiquement l'exécution conditionnelle ou itérative d'un message.

Un cadre d'interaction est représenté dans un rectangle dont le coin supérieur gauche contient un pentagone.

Dans le pentagone un *opérateur d'interaction (alt, opt, loop, ref,…)*.

Les opérandes d'un opérateur d'interaction sont séparés par une ligne pointillée.

Les conditions de choix des opérandes sont données entre crochets ([ ]).

8. Mise en œuvre d'UML :

Nous allons présenter la méthode simple et générique proposé par Laurent AUDIBERT qui se situe à mi-chemin entre UP (Unified Process), qui constitue un cadre général très complet de processus de développement, et XP (eXtrême Programming) qui est une approche minimaliste à la mode centrée sur le code.

Elle a donc montré son efficacité dans la pratique et est :

– conduite par les cas d'utilisation, comme UP, mais bien plus simple ;

– relativement légère et restreinte, comme XP, mais sans négliger les activités de modélisation en analyse et conception ;

– fondée sur l'utilisation d'un sous-ensemble nécessaire et suffisant du langage UML (modéliser 80% des problèmes en utilisant 20% d'UML).

Dans tous les cas, il faut garder à l'esprit qu'une méthode n'est pas une formule magique. Le fait de produire des diagrammes UML selon un ordre établi n'est en aucun cas une garantie de réussite. Une méthode ne sert qu'à canaliser et ordonner les étapes de la modélisation. La valeur n'est pas dans la méthode mais dans les personnes qui la mettent en œuvre. Le processus est organisé suivant les trois phases suivantes : identification des besoins, phase d'analyse et phase de conception

8.1 Identification des besoins et spécification des fonctionnalités :

8.1.1 Identification et représentation des besoins :Diagramme de cas d'utilisation

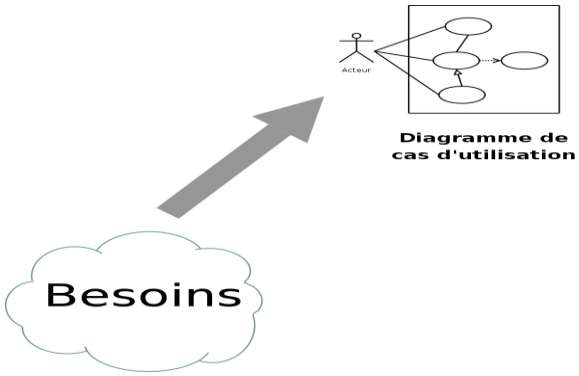


Figure 1.20 Les besoins sont modélisés par un diagramme de cas d'utilisation

Les cas d'utilisation sont utilisés tout au long du projet. Dans un premier temps, on les crée pour identifier et modéliser les besoins des utilisateurs Ces besoins sont déterminés à partir des informations recueillies lors des rencontres entre informaticiens et utilisateurs. Il faut impérativement proscrire toute considération de réalisation lors de cette étape.

Durant cette étape, vous devrez déterminer les limites du système, identifier les acteurs et recenser les cas d'utilisation.

8.1.2 Spécification détaillée des besoins : Diagrammes de séquence système

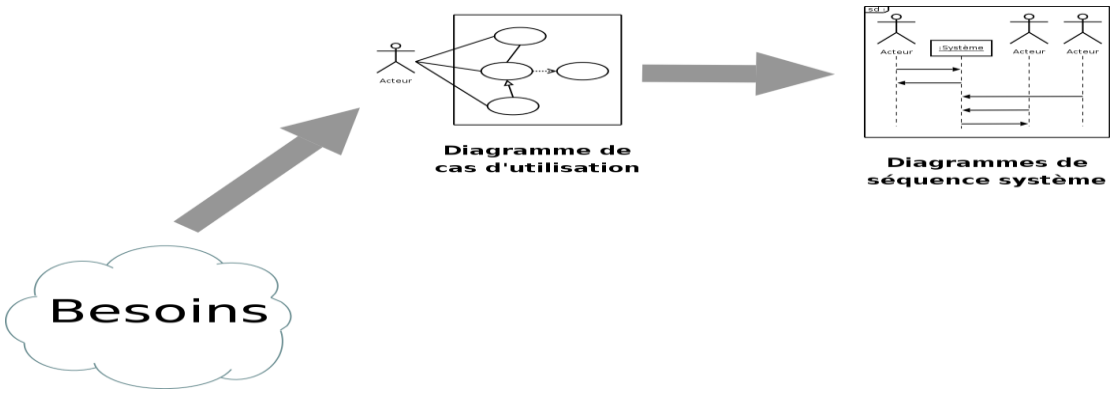


Figure 1.21 Les diagrammes de séquence système illustrent la description textuelle des cas d'utilisation

Dans cette étape, on cherche à détailler la description des besoins par la description textuelle des cas d'utilisation et la production de diagrammes de séquence système illustrant cette description textuelle.

Cette étape amène souvent à mettre à jour le diagramme de cas d'utilisation puisque nous somme toujours dans la spécification des besoins.

Les scénarios de la description textuelle des cas d'utilisation peuvent être vus comme des instances de cas d'utilisation et sont illustrés par des diagrammes de séquence système. Il faut, au minimum, représenter le scénario nominal de chacun des cas d'utilisation par un diagramme de séquence qui rend compte de l'interaction entre l'acteur, ou les acteurs, et le système. Le système est ici considéré comme un tout et est représenté par une ligne de vie. Chaque acteur est également associé à une ligne de vie.

Lorsque les scénarios alternatifs d'un cas d'utilisation sont nombreux et importants, l'utilisation d'un diagramme d'états-transitions ou d'activités peut s'avérer préférable à une multitude de diagrammes de séquence.

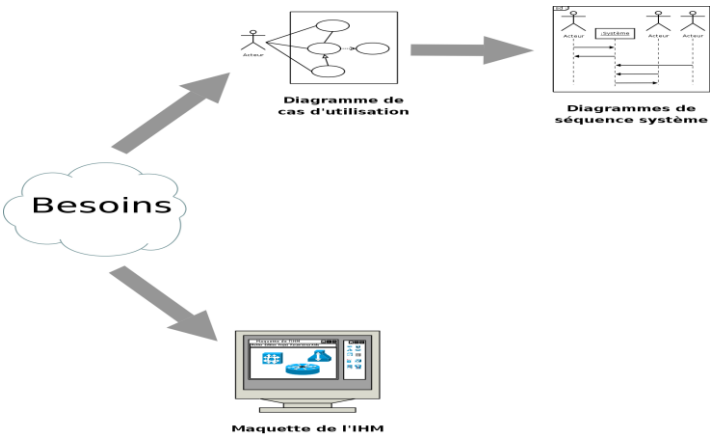


Figure 1.22 Une maquette d'IHM facilite les discussions avec les futurs utilisateurs

8.1.3 Maquette de l'IHM de l'application (non couvert par UML) :

Une maquette d'IHM (Interface Homme-Machine) est un produit jetable permettant aux utilisateurs d'avoir une vue concrète mais non définitive de la future interface de l'application (figure 1.22). La maquette peut très bien consister en un ensemble de dessins produits par un logiciel de présentation ou de dessin. Par la suite, la maquette pourra intégrer des fonctionnalités de navigation permettant à l'utilisateur de tester l'enchaînement des écrans ou des menus, même si les fonctionnalités restent fictives. La maquette doit être développée rapidement afin de provoquer des retours de la part des utilisateurs.

8.2 Phases d'analyse :

8.2.1 Analyse du domaine : modèle du domaine :

La modélisation des besoins par des cas d'utilisation s'apparente à une analyse fonctionnelle classique. L'élaboration du modèle des classes du domaine permet d'opérer une transition vers une véritable modélisation objet. L'analyse du domaine est une étape totalement dissociée de l'analyse des besoins (sections 8.1.1, 8.1.2 et 8.1.3). Elle peut être menée avant, en parallèle ou après cette dernière.

La phase d'analyse du domaine permet d'élaborer la première version du diagramme de classes (figure 1.24) appelée modèle du domaine.

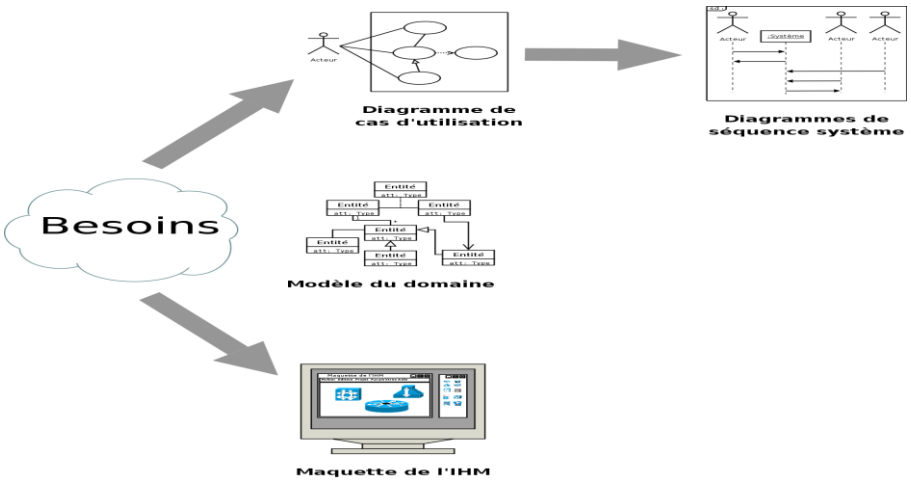


Figure 1.23 La phase d'analyse du domaine permet d'élaborer la première version du diagramme de classes.

8.2.2 Diagramme de classes participantes :

Le diagramme de classes participantes est particulièrement important puisqu'il effectue la jonction entre, d'une part, les cas d'utilisation (section 8.1.1), le modèle du domaine (section 8.2.1) et la maquette (section 8.1.3), et d'autre part, les diagrammes de conception logicielle que sont les diagrammes d'interaction (section 8.3.1) et le diagramme de classes de conception (section 8.3.2). Les diagrammes de conception logicielle n'apparaissent pas encore sur la figure 1.24

Le diagramme de classes participantes modélise trois types de classes d'analyse, les dialogues, les contrôles et les entités ainsi que leurs relations :

- Les classes entité vont seulement posséder des attributs. Ceux-ci représentent généralement les infos persistantes de l'application.
- Les classes contrôle vont seulement posséder des opérations. Celles-ci correspondent à la logique de l'application, aux règles métier.
- Les classes dialogue vont posséder des attributs et des opérations. Les attributs vont représenter des champs de saisie. Les opérations représenteront les actions de l'utilisateur sur l'IHM.

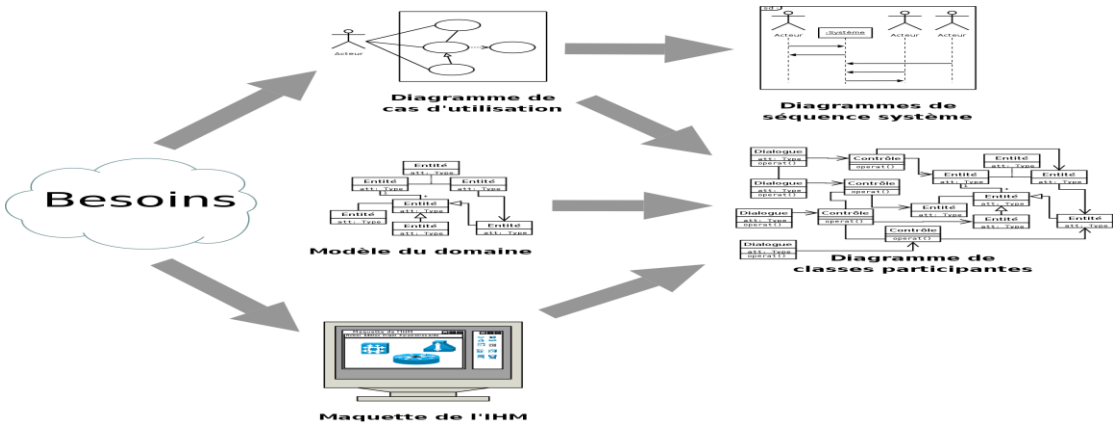


Figure 2.24 Le diagramme de classes participantes effectue la jonction entre les cas d'utilisation, le modèle du domaine et les diagrammes de conception logicielle

8.2.3 Diagrammes d'activités de navigation :

Les IHM modernes facilitent la communication entre l'application et l'utilisateur en offrant toute une gamme de moyens d'action et de visualisation comme des menus déroulants ou contextuels, des palettes d'outils, des boîtes de dialogues, des fenêtres de visualisation, etc. Cette combinaison possible d'options d'affichage, d'interaction et de navigation aboutis aujourd'hui à des interfaces de plus en plus riches et puissantes.

Les diagrammes d'activités de navigation sont à relier aux classes de dialogue du diagramme de classes participantes. Les différentes activités du diagramme de navigation peuvent être stéréotypées en fonction de leur nature : « fenêtre »,
« Menu», « menu contextuel », « dialogue », etc.

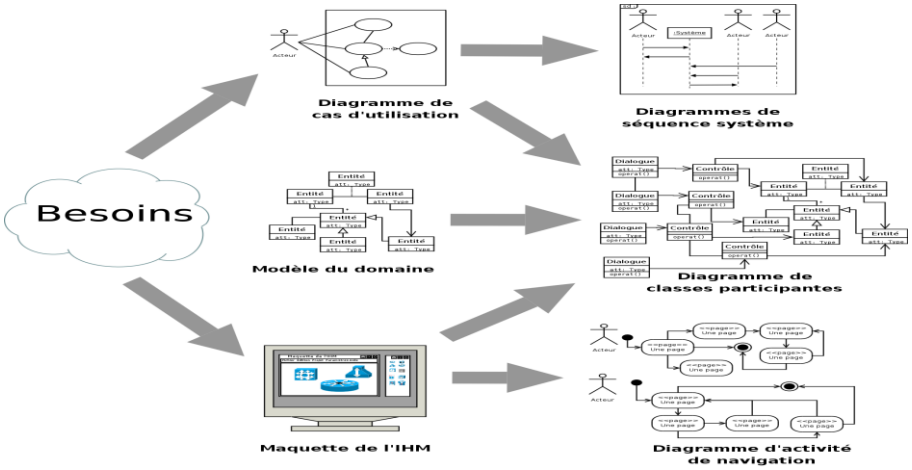


Figure 2.25 Les diagrammes d'activités de navigation représentent graphiquement l'activité de navigation dans l'IHM.

8.3 Phase de conception:

8.3.1 Diagrammes d'interaction :

Dans les diagrammes d'interaction, les objets communiquent en s'envoyant des messages qui invoquent des opérations sur les objets récepteurs (voir figure8.8). Il est ainsi possible de suivre visuellement les interactions dynamiques entre objets, et les traitements réalisés par chacun d'eux. Avec un outil de modélisation UML (comme Rational Rose ou Power AMC), la spécification de l'envoi d'un message entre deux objets crée effectivement une opération publique sur la classe de l'objet cible. Ce type d'outil permet réellement de mettre en œuvre l'allocation des responsabilités à partir des diagrammes d'interaction.

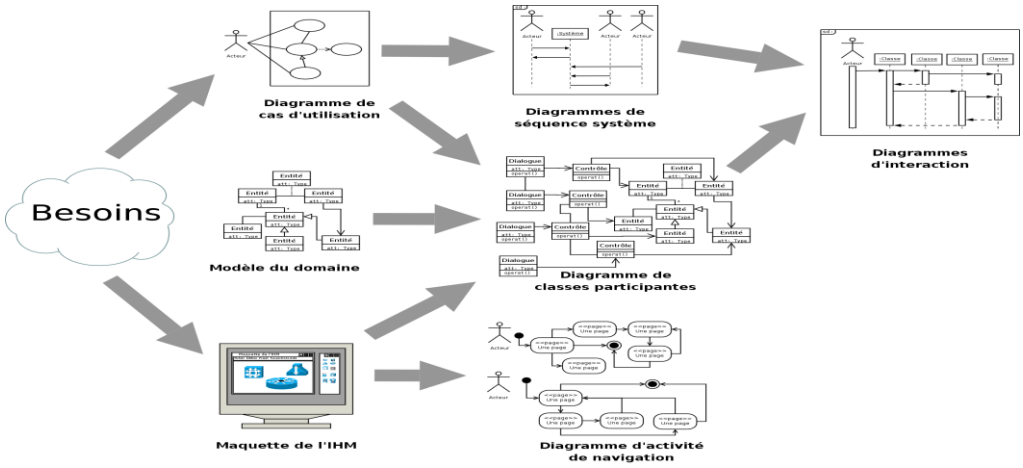


Figure 2.26 Les diagrammes d'interaction permettent d'attribuer précisément les responsabilités de comportement aux classes d'analyse.

8.4.2 Diagramme de classes de conception :

L'objectif de cette étape est de produire le diagramme de classes qui servira pour l'implémentation (figure 1.27). Une première ébauche du diagramme de classes de conception a déjà été élaborée en parallèle des diagrammes d'interaction (section 8.3.1). Il faut maintenant le compléter en précisant les opérations privées des différentes classes. Il faut prendre en comptes les choix techniques, comme le choix du langage de programmation, le choix des différentes librairies utilisées (notamment pour l'implémentation de l'interface graphique), etc.

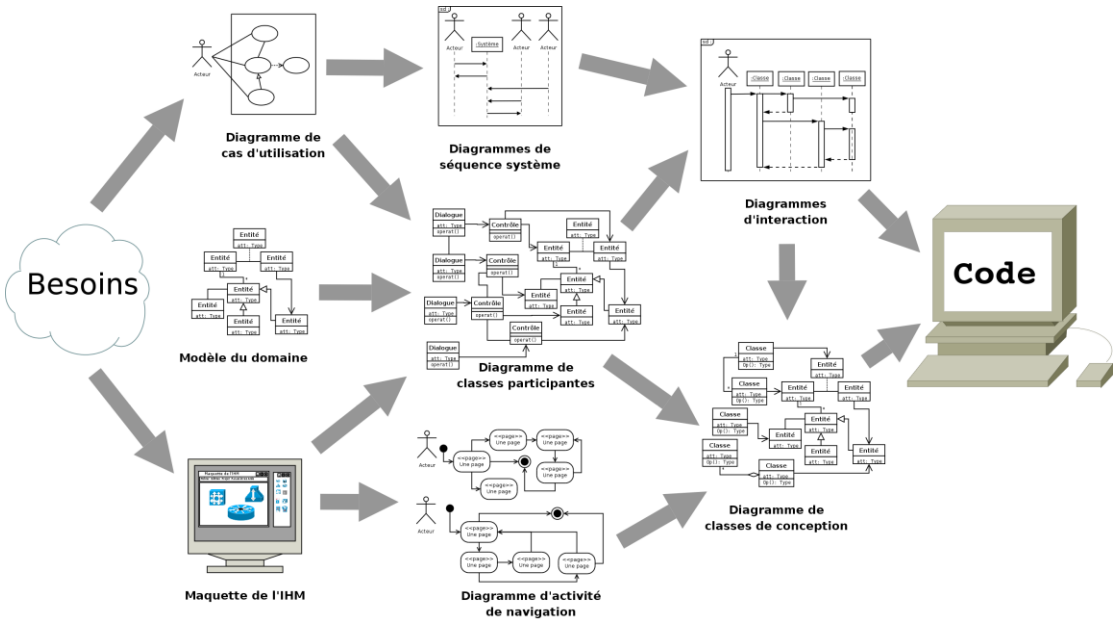


Figure 2.27 Chaîne complète de la démarche de modélisation du besoin jusqu'au code.

9. Conclusion :

Dans ce chapitre, nous avons passé en revue le langage de modélisation UML. D'abord, nous avons présenté les différents diagrammes définit par UML, puis nous avons présenté le processus que nous allons utiliser pour modéliser notre application. UML est un langage riche ; il couvre toutes les phases d'un cycle de développement. Il est également ouvert (indépendant du domaine d'application et des langages d'implémentation). UML s'industrialise : les outils qui le supportent se multiplient. Ils permettent de respecter les normes de représentation, de gérer dans une certaine mesure la cohérence de l'analyse, de générer des rapports de documentation, etc. Autrement dit, ils permettent de s'abstraire du travail fastidieux et répétitif à l'utilisation d'une méthode et d'un formalisme, et donc de se focaliser sur la compréhension et la résolution du problème. Pour ces raisons nous avons choisi UML pour modéliser notre application.

CHAPETRE 2
IDENTIFICATION DES BESOINS

1. Introduction

Après avoir présenté le processus de développement (AUDIBERT) et le langage de modélisation(UML) dans le chapitre précédent, nous allons à présent les utiliser pour la modélisation de notre système. En effet la conception c'est l'étape la plus importante, nous permet d'une part de clarifier les besoins des utilisateurs et d'autre part la meilleure compréhension de système à développer. Dans un premier temps, nous allons introduire le cahier de charges qui servira de fil conducteur tout au long de la conception. Dans un second temps, nous commencerons à identifier les acteurs qui interagiront avec le système, pour pouvoir établir précisément les frontières fonctionnelles du système. Puis, nous identifions et nous décrivons les cas d'utilisation du système.

2. Présentation du projet

Le domaine d'étude spécifié pour notre projet est la conception et la réalisation d'un système d'information pour la gestion d'un cabinet médicale de médecine générale.

2.1 Présentation de l'organisme d'accueil

Un cabinet médical est une unité sanitaire privée, par ce sujet, nous étudions la gestion d'un cabinet médicale de médecine générale appartenant au Dr.HAYOUR NOREDDINE qui a ouvert cette cabine en1989. Le cabinet se situe à Sidi Marouane dans la wilaya de Mila, sa fonction principale est de consulter quotidiennement les malades ainsi que leurs suivi.

Elle est composé de :
Une salle d'examen
Un bureau de médecin
Un bureau pour la secrétaire
Une salle d'attente femme
Une salle d'attente homme

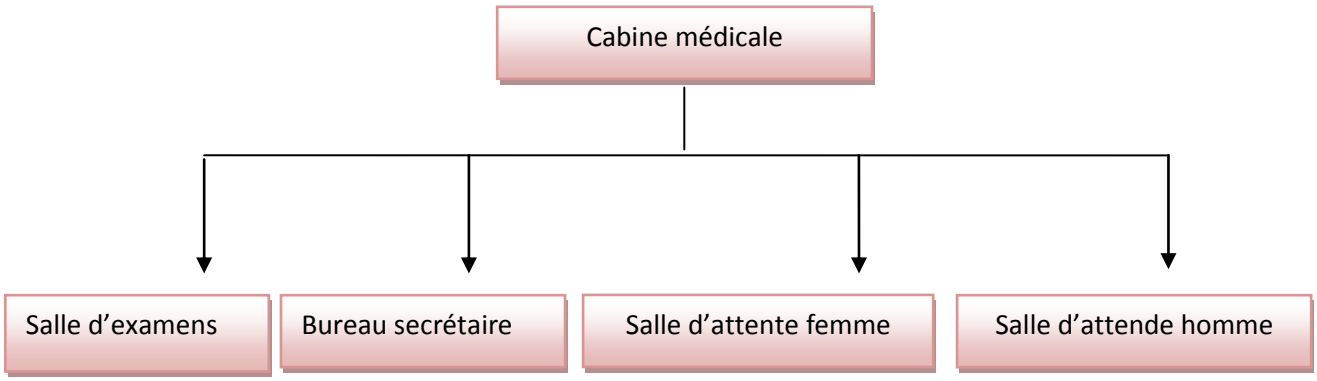


Figure 2.1 L'organigramme du cabinet médical

3. Grandes choix techniques :

Une variété de langages de programmation et de méthodes de développement est utilisée dans le but de réaliser une application répondant aux besoins des utilisateurs en ce qui concerne d'assurer une fiabilité et une cohérence de l'information. Les choix techniques que nous avons adoptés sont :

L'utilisation du processus de développement (AUDIBERT)

L'utilisation d'un langage de modélisation standard pour la conception orientée objet : UML

La création de la base de données et sa gestion avec : paradoxe

L'utilisation du langage Delphi

3.1 Recueil des besoins fonctionnels :

Un premier tour d'horizon des besoins exprimés par les employés du cabine médicale a permis d'établir le cahier de charges préliminaire suivant :

La gestion de fiche de consultation

C'est une opération qui est **e**ffectuée par le médecin et qui contient les actions suivantes: (Modifier le fiche de consultation, Créer le fiche de consultation, Supprimer le fiche de consultation). Cette fiche contient les informations suivantes : (Nom, prénom date de consultation, diagnostique, ordonnance, note, RDV en cours, observation…….)

Modifier le fiche de consultation : Une fiche de consultation est modifiée lors de la consultation du patient par le médecin et la modification importe sur les informations concernant le patient.

Supprimer le fiche de consultation : le médecin peut supprimer la fiche de consultation

Créer le fiche de consultation : à l'arrivée d'un patient, le médecin ajoute une fiche de consultation afin de mentionner les informations nécessaires de la consultation, comme observation et diagnostic….cet .

La gestion de fiche de patient : la secrétaire est le responsable de l'opération de la gestion de la fiche de patient. La gestion peut comporter la suppression, l'ajout, la recherche..... des informations sur le patient.

L'ajout fiche patient : à l'arrivée d'un nouveau patient, la secrétaire crée une fiche médicale, qui est la fiche patient, contenant les informations de l'état civil du patient.

Supprimer fiche patient : la secrétaire pout supprimer un patient.

Modifier fiche patient : pour que la secrétaire puisse modifier ou supprimer une fiche patient, il faut d'abord lancer la recherche de cette dernière.

Etablir ordonnance : le médecin établie l'ordonnance après consultation du patient.

Etablir certificat : le médecin établie les différentes certificats pour le patient à la demande et en cas de besoin.

Etablir la facture : la secrétaire établie une facture à la fin de consultation du patient selon le type de la consultation.

3.2 Recueil des besoins opérationnels

3.2.1 Les besoins de sécurité

Peuvent définir les niveaux d'accès possibles au système pour chacun des rôles métier, voici comment un besoin de sécurité type pourrait être énoncé : ''chaque acteur principale possède un mot de passe et un nom d'utilisateur ''.

3.3.2 La rapidité de traitement

Vu le nombre important d'interactions quotients, il est nécessaire que la durée d'exécution des traitements s'approche le plus possible du temps réel.

3.3.3 La convivialité

Le logiciel doit être facile à utiliser. En effet, les interfaces utilisateurs doivent être conviviales : c'est-à-dire simples, ergonomiques et adaptées à l'utilisateur que peut être un néophyte en informatique.

3.3 Identification d'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 .un acteur

peut consulter et/ou modifier directement l'état du système, en émettant et/ou recevant des messages éventuellement porteurs de données.

Les acteurs candidats sont les acteurs humains directs (faites donc en sorte d'identifier tous les profils possibles), les autres systèmes connexes qui interagissent directement avec le système étudié, souvent par le biais de protocoles bidirectionnels.

Dans notre application, nous pouvons identifier deux acteurs qui sont :

L e médecin :

C'est le responsable des opérations suivantes :

La gestion de la fiche de consultation, l'établissement des analyses, l'établissement des certificats et d'ordonnance.

La secrétaire :

C'est la personne qui gère la fiche de patient et les rendez-vous et établie la facture de consultation de patient.

4. Diagramme de cas d'utilisation

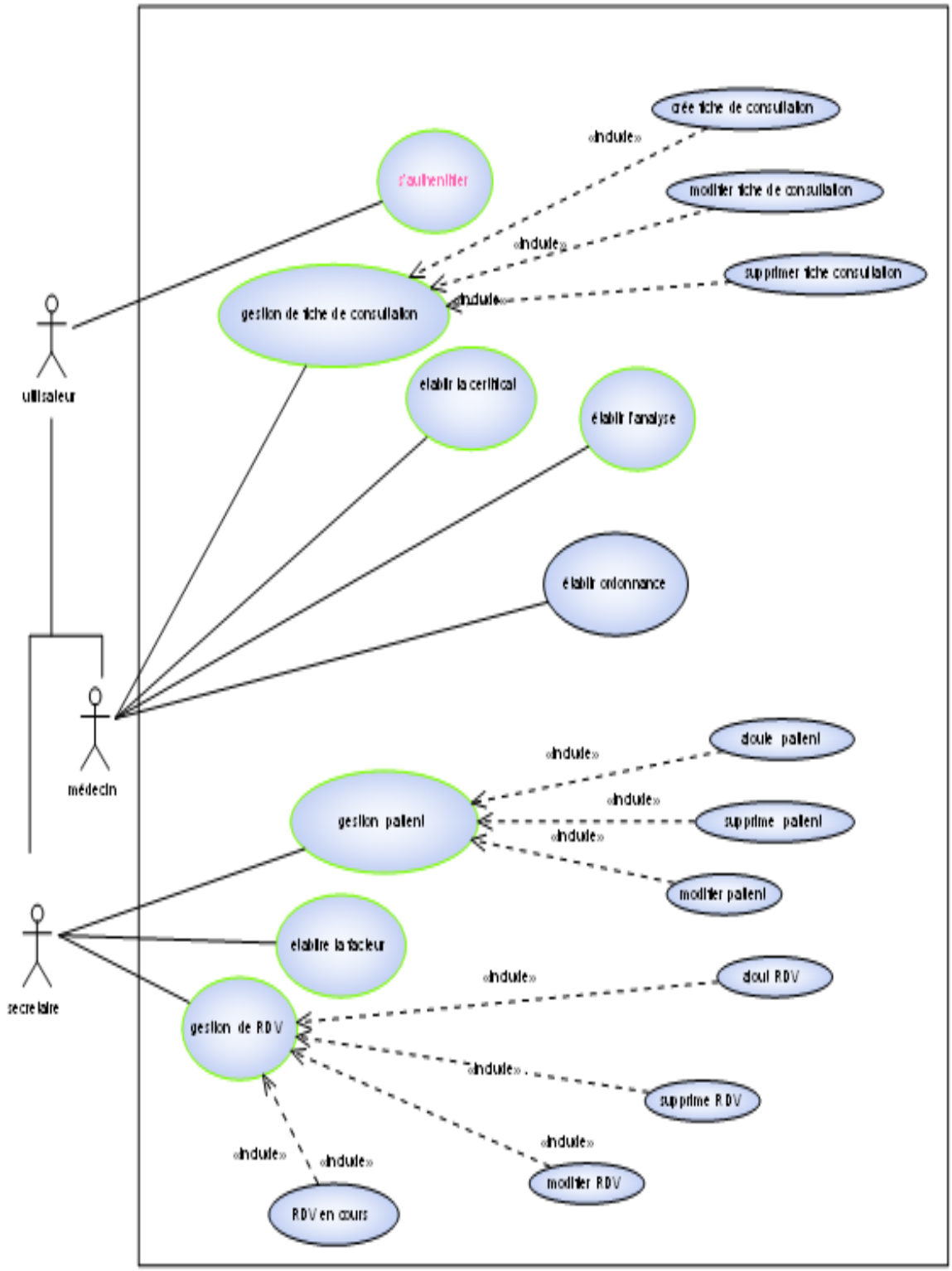


Figure 2.2 diagramme de cas d'utilisation

5. Description des cas d'utilisation

5.19 S'authentifier :

cas d'utilisation	**S'authentifier**
objectif	Vérifier l'autorisation d'accès au système
acteur	médecin
Pré condition	L'utilisateur saisie un mot de passe, un login et un type pour identifier l'acteur
Poste condition	L'utilisateur est authentifié par le système
Scénario nominal	1. Le système affiche le formulaire s'authentifier 2. Le système demande la saisie du login, du mot de passe et du type 3. L'utilisateur saisi le login, le mot de passe et le type 4. Le système vérifie le login, le mot de passe et le type puis il ouvre le système
Scénario alternatif	3.1. Le mot de passe est erroné 3.1.1. le système affiche un message d'erreur. 3.1.2. Le système propose à l'utilisateur de renseigner une nouvelle fois le mot de passe. 3.1.3. Reprise de l'enchaînement du scénario nominal au point 2.patient et retour point 3

5.1Ajout fiche de consultation

cas d'utilisation	ajouter une fiche de consultation
objectif	Crée une nouvelle fiche de système
acteur	médecin
Pré condition	Crée la table de consultation
Poste condition	fiche de consultation établie
Scénario nominal	1. Le système affiche le formulaire et il crée une fiche de consultation 2. Le médecin saisie les informations 3. Le système demande la validation de la création

5.2. Modifier fiche de consultation :

cas d'utilisation	Modifier une fiche de consultation
Objectif	Modifier les informations de la fiche de consultation
Acteur	Médecin
Pré condition	Ouvrir la fiche de consultation
Poste condition	Fiche valide
Scénario nominal	1. Le système affiche le formulaire pour modifier la fiche de consultation 2. Le système demande la saisie de nom et de prénom du patient 3. Le médecin saisie le nom et le prénom du patient 4. Le système vérifie le nom et le prénom du patient 5. Le médecin modifie les informations 6. Le système demande la validation de la modification
Scénario alternatif	Le nom et le prénom sont erronés Le système demande de corriger le nom et le prénom et retourne au point 3

5.3 Supprimer la fiche de consultation :

cas d'utilisation	Supprimer fiche de consultation
objectif	Supprimer la fiche de consultation de la base de données
acteur	médecin
Pré condition	La fiche existe déjà
Poste condition	Fiche consultation supprimée
Scenario nominal	1. Le système affiche le formulaire supprime fiche de consultation 2. Le système demande le nom et le prénom de patient 3. Médecin saisie le nom et le prénom du patient 4. Le système demande la validation de suppression

5.5 Ajout patient :

cas d'utilisation	Ajout fiche de patient
objectif	Ajout un nouveau patient dans la base de données
acteur	Secrétaire
Pré condition	Consulter la table de patients
Poste condition	fiche de consultation établie
Scenario nominal	1. Le système affiche le formulaire d'ajout du patient 2. Le système demande les informations du patient 3. Le Secrétaire saisie les informations du patient 4. Le système demande la validation d'jout
Scénario alternatif	Le nom et le prénom existe déjà Le système affiche un message d'existence et retourne au point 2

5.6 Supprime patient :

cas d'utilisation	Supprimer la fiche de patient
objectif	Supprimer patient de la base de données
acteur	Secrétaire
Pré condition	La fiche existe déjà
Poste condition	Fiche patient supprimée
Scenario nominal	1. Le système affiche le formulaire supprime fiche de patient 2. Le système demande le nom et le prénom de patient 3. La Secrétaire saisie le nom et le prénom du patient 4. Le système demande la validation de suppression

5.7 Modifier la fiche de patient :

cas d'utilisation	Modifier une fiche de patient
objectif	Modifier les informations de fiche de patient
acteur	Secrétaire
Pré condition	La fiche existe déjà
Poste condition	Fiche modifiée
Scenario nominal	1. Le système affiche le formulaire modifier patient 2. Le système demande la saisie de nom et de prénom de patient 3. La Secrétaire saisie le nom et le prénom du patient 4. Le système vérifie le nom et le prénom de patient 5. La secrétaire modifie les informations 6. Le système demande de valider la modification
Scénario alternatif	Le nom et prénom de patient sont erronés Le système demande de corriger le nom et le prénom de patient et retourne au point 3

5.9 Ajout un RDV :

cas d'utilisation	Ajout un RDV
objectif	Ajouter un nouveau RDV
acteur	Secrétaire
Pré condition	Le RDV n'existe pas dans la base de données
Poste condition	Le RDV ajouté
Scenario nominal	1. Le système affiche le formulaire ajout RDV de patient 2. La Secrétaire saisie les informations du RDV de patient 3. Le système demande la validation d'jout
Scénario alternatif	Le RDV existe déjà Le système affiche un message d'existence et retourne au point 2

5.10 Supprimer un RDV :

cas d'utilisation	Supprimer RDV
objectif	Supprimer un RDV
acteur	Secrétaire
Pré condition	Le RDV existe déjà
Poste condition	Le RDV supprimé
Scenario nominal	1. Le système affiche le formulaire supprime RDV 2. Le système demande le nom et le prénom du patient 3. La Secrétaire saisie le nom et le prénom du patient 4. Le système demande la validation de suppression

5.11 Modifier un RDV

cas d'utilisation	Modifier de RDV
objectif	Modifier les informations de RDV
acteur	Secrétaire
Pré condition	Le RDV existe déjà
Poste condition	Le RDV modifié
Scenario nominal	1. Le système affiche le formulaire modifier le RDV de patient 2. Le système demande la saisie du nom et du prénom de patient 3. La Secrétaire saisie le nom et le prénom du patient 4. Le système vérifie le nom et le prénom du patient 5. La secrétaire modifie les informations 6. Le système demande de valider la modification
Scénario alternatif	Le nom et le prénom du patient sont erronés Le système demande de corriger le nom et le prénom du patient et retourne au point 3

5.12 Établir la facture :

cas d'utilisation	Etablir la facture
objectif	Etablir une facture sur la fiche patient
acteur	Secrétaire
Pré condition	Le patient existe déjà
Poste condition	Affiche la facture de patient
Scenario nominal	1. Le système affiche le formulaire **établir la facture de patient** 2. Le système demande la saisie le nom et le prénom du patient 3. La Secrétaire saisie le nom et le prénom du patient 4. Le système vérifie le nom et le prénom du patient 5. La secrétaire établie la facture 6. Le système demande de valider la facture et l'imprime
Scénario alternatif	Le nom et prénom de patient sont erronés Le système demande de corriger le nom et le prénom du patient et retourne au point 3

5.13 Etablir l'analyse :

cas d'utilisation	Etablir l'analyse
objectif	Etablir l'analyse
acteur	Secrétaire
Pré condition	Liste des analyses
Poste condition	Imprime la fiche des analyses
Scenario nominal	1. Le système affiche le formulaire de liste des analyses 2. Le système demande la saisie de nom et de prénom de patient 3. Le médecin saisi le nom et le prénom de patient 4. Le système vérifie le nom et le prénom de patient 5. Le médecin établie la fiche des analyses 6. Le système demande la validation et imprime la fiche
Scénario alternatif	Le nom et le prénom de patient sont erronés Le système demande de corriger le nom et le prénom de patient et retourne au point 3

5.15 Etablir ordonnance :

cas d'utilisation	Etablir ordonnance
objectif	Etablir ordonnance
acteur	Secrétaire
Pré condition	Fiche ordonnance
Poste condition	Imprimer ordonnance
Scenario nominal	1. Le système affiche le formulaire ordonnance 2. Le médecin établie ordonnance 3. Le système demande la validation et imprime la fiche

5.16 RDV en cours :

cas d'utilisation	Ajout un RDV
objectif	Ajout RDV en cours
acteur	médecin
Pré condition	Le RDV existe déjà dans la base de données
Poste condition	RDV consulté
Scénario nominal	1. Le système affiche le formulaire fiche de consultation 2. Le médecin saisie les informations de RDV en cours de patient 3. Le système demande la validation de modifier

5.18 Etablir certificat :

cas d'utilisation	Etablir certificat
objectif	Etablir certificat
acteur	Médecin
Pré condition	Fiche certificat
Poste condition	Imprime certificat
Scenario nominal	1. Le système affiche le formulaire certificat 2. Le système demande la saisie de nom et de prénom du patient 3. Le médecin saisi le nom et le prénom de patient 4. Le système vérifie le nom et le prénom de patient 5. Le médecin établie le certificat 6. Le système demande la validation et lance l'impression
Scénario alternatif	Le nom et le prénom de patient sont erronés Le système demande de corriger le nom et le prénom de patient et retourne au point 3

6. Digrammes de séquence système :

6.1 S'authentifier :

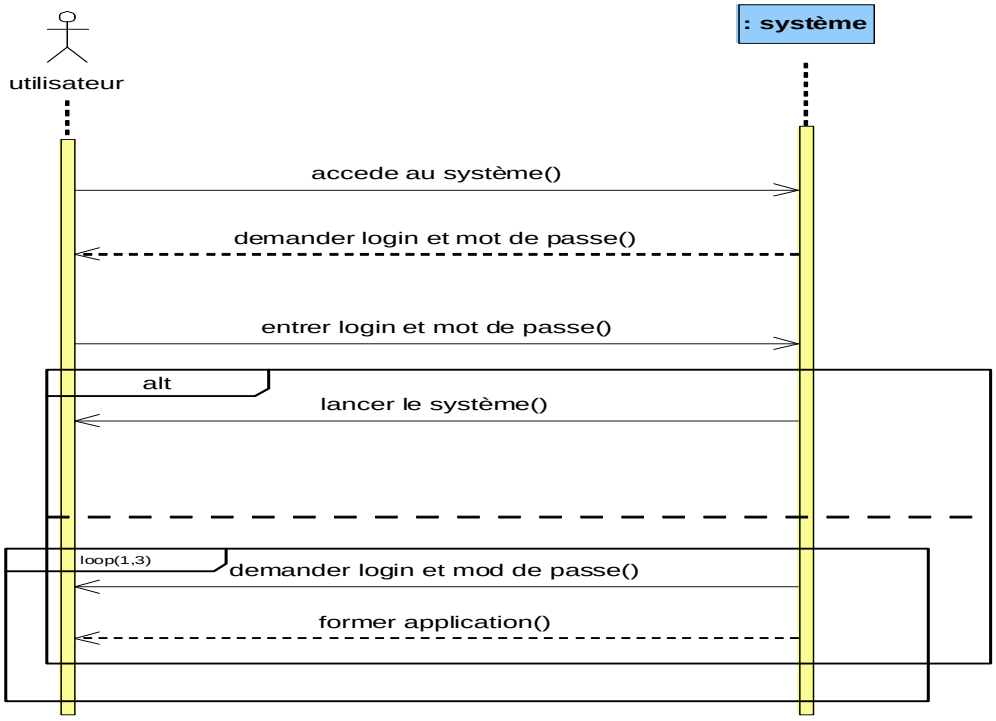


Figure2.3. Diagramme de séquence système du cas d'utilisation « S'authentifier »

6.2 Ajoute patient

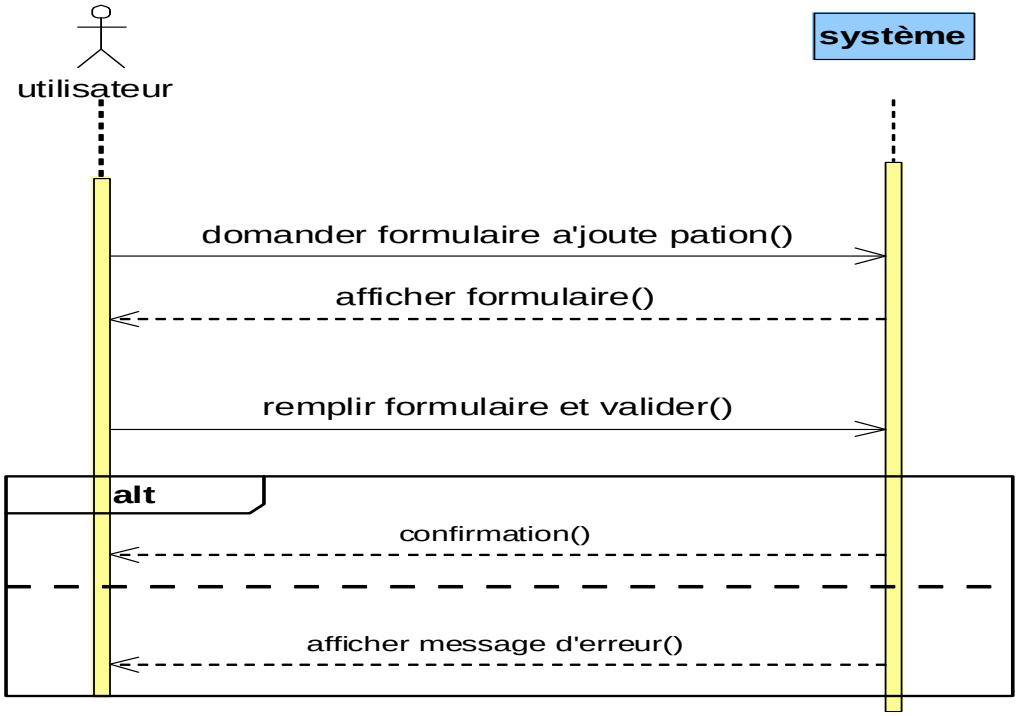


Figure 2.4. Diagramme de séquence système du cas d'utilisation « ajoute patient»

6.3 Modifier patient :

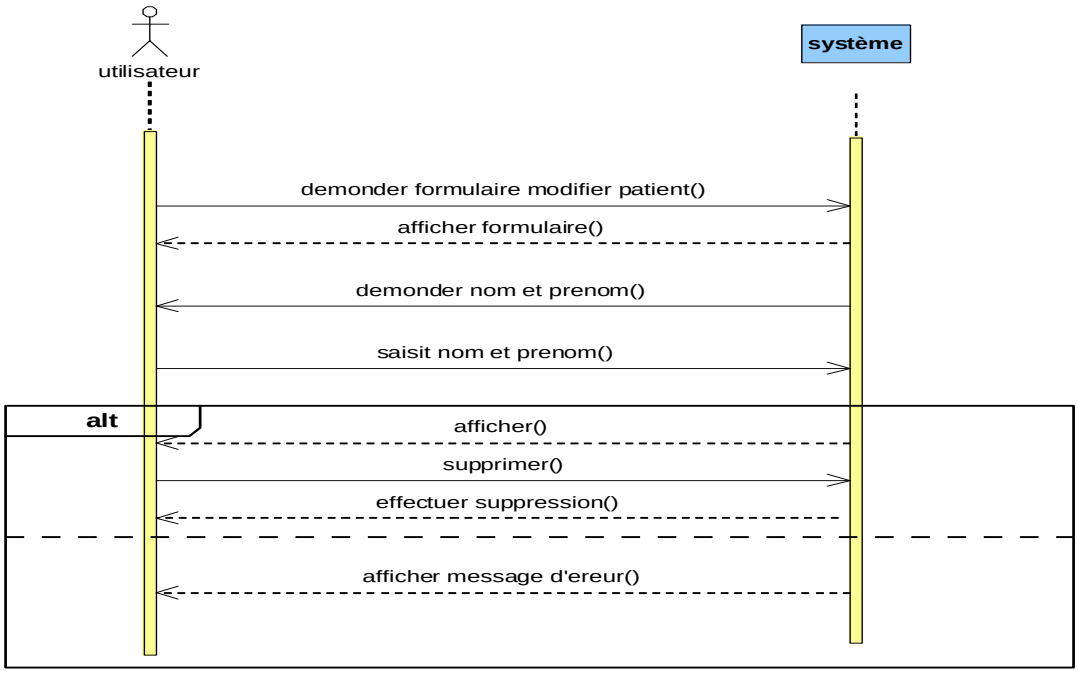


Figure 2.5. Diagramme de séquence système du cas d'utilisation « modifier patient »

6.4 Supprimer fiche patient :

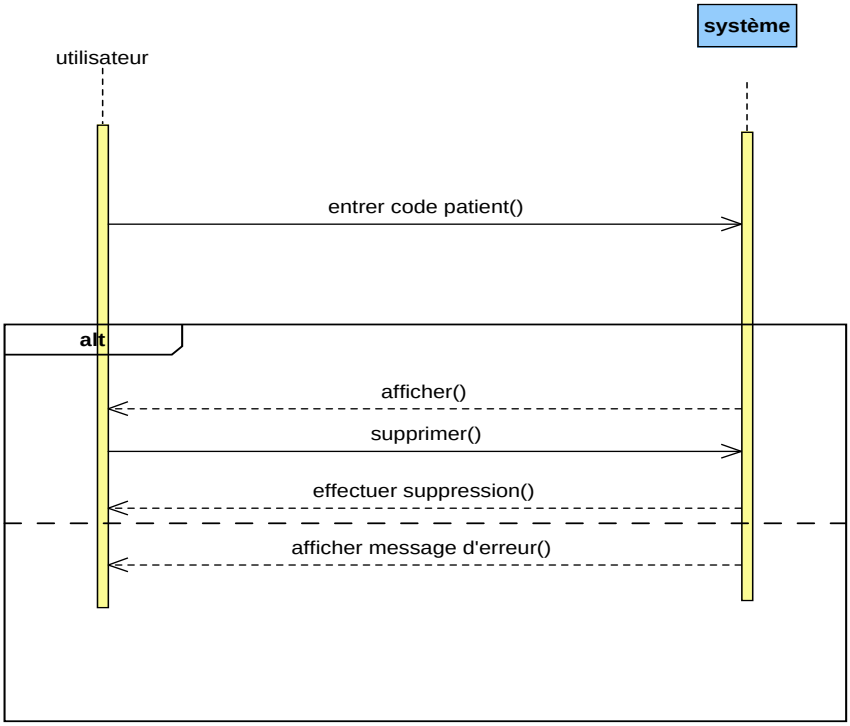


Figure 2.6. Diagramme de séquence système du cas d'utilisation « supprimer fiche de consultation »

6.5 crée fiche de consultation :

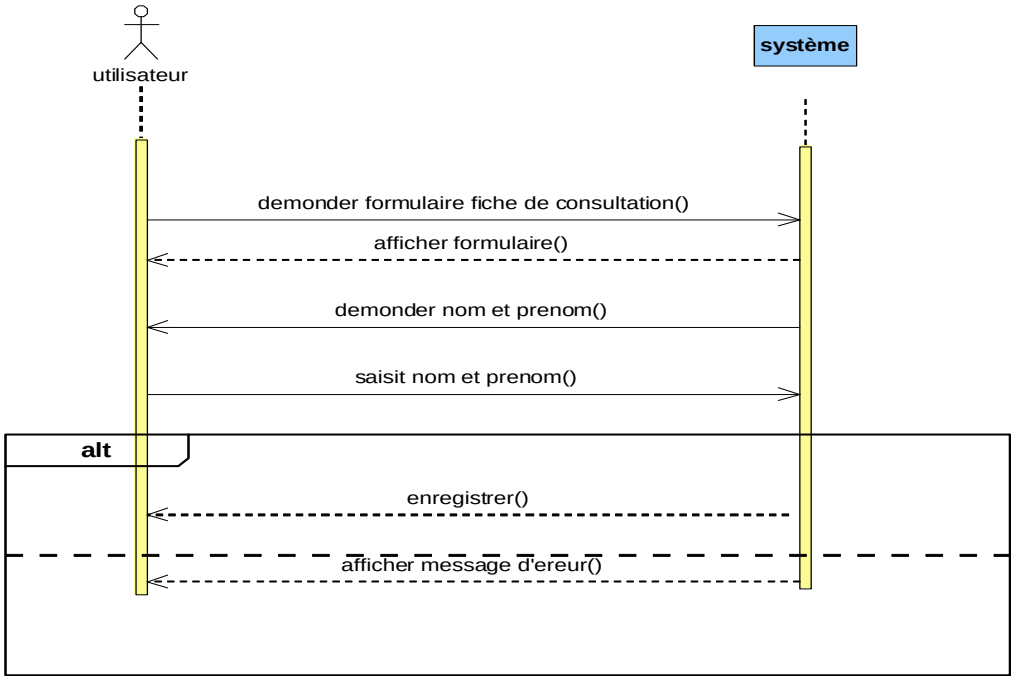
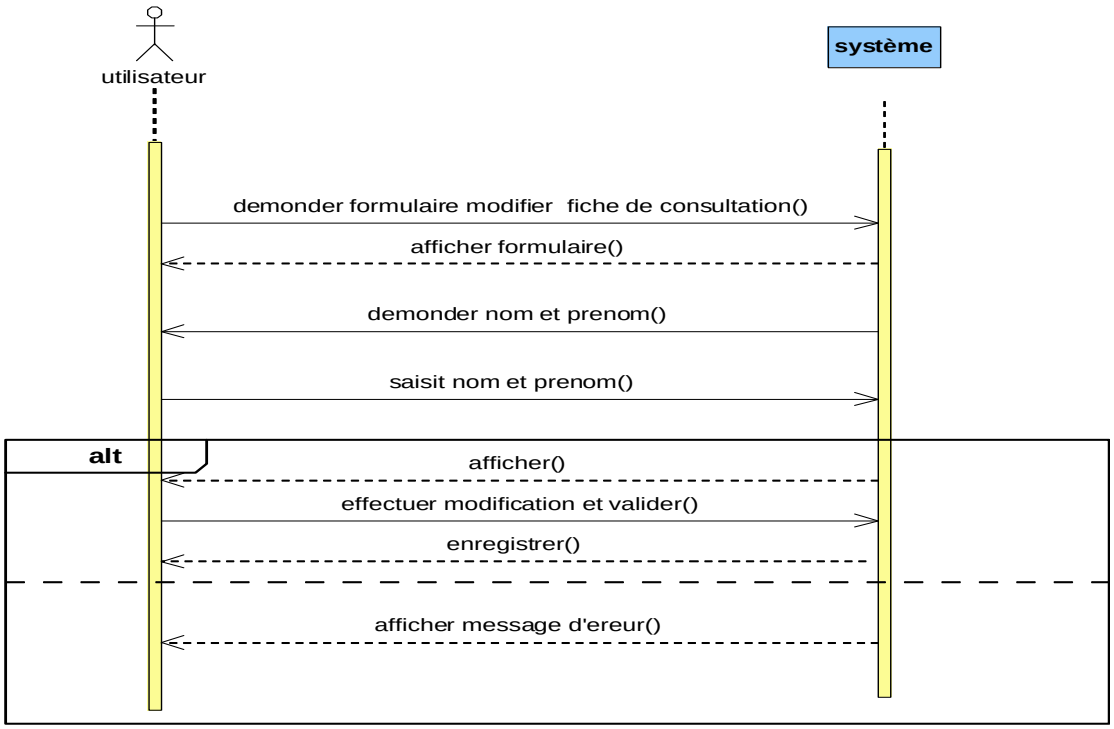


Figure 2.7. Diagramme de séquence système du cas d'utilisation « crée fiche de consultation »

6.6 Modifier fiche de consultation :



2.8 Diagramme de séquence système du cas d'utilisation « modifier fiche ce consultation»

6.8 Supprimer fiche consultation

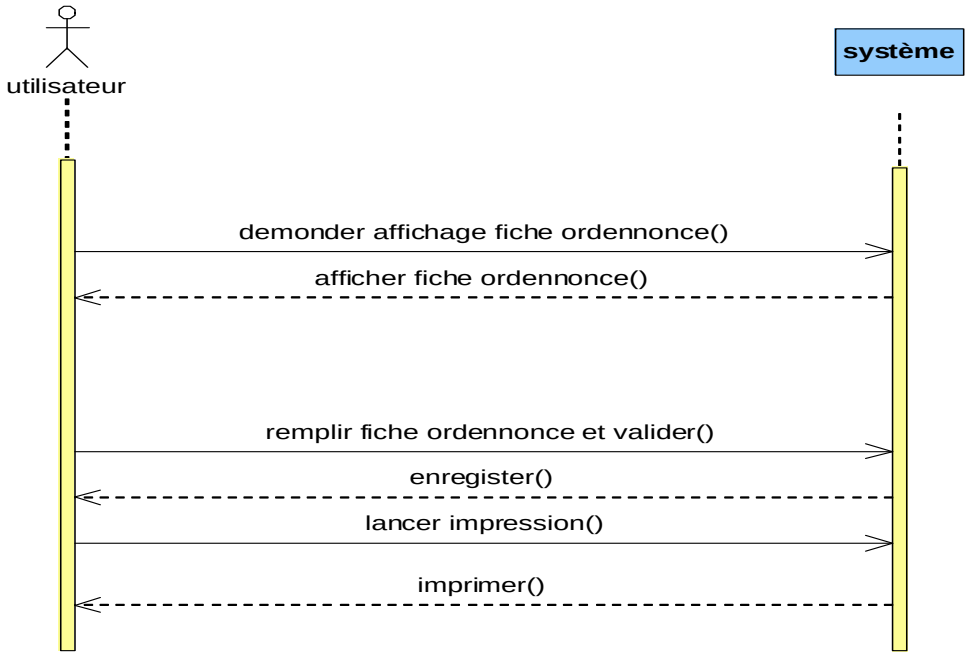


Figure 2.8. Diagramme de séquence système du cas d'utilisation « supprimer fiche consultation»

6.9 Ajouter RDV

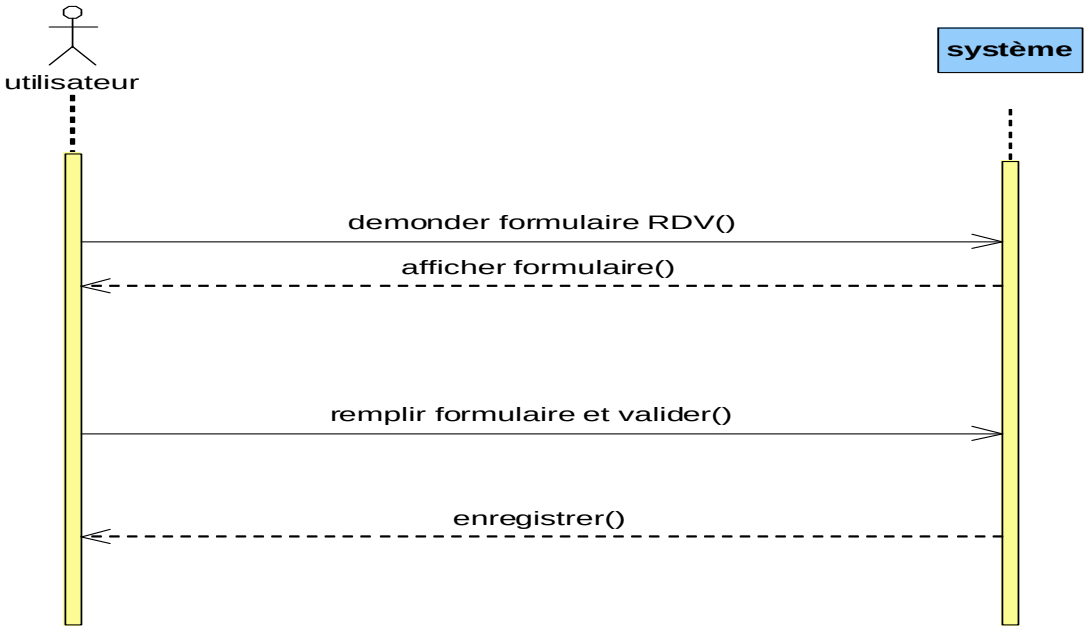


Figure 2.9. Diagramme de séquence système du cas d'utilisation « ajoute RDV »

6.10 Supprimer RDV :

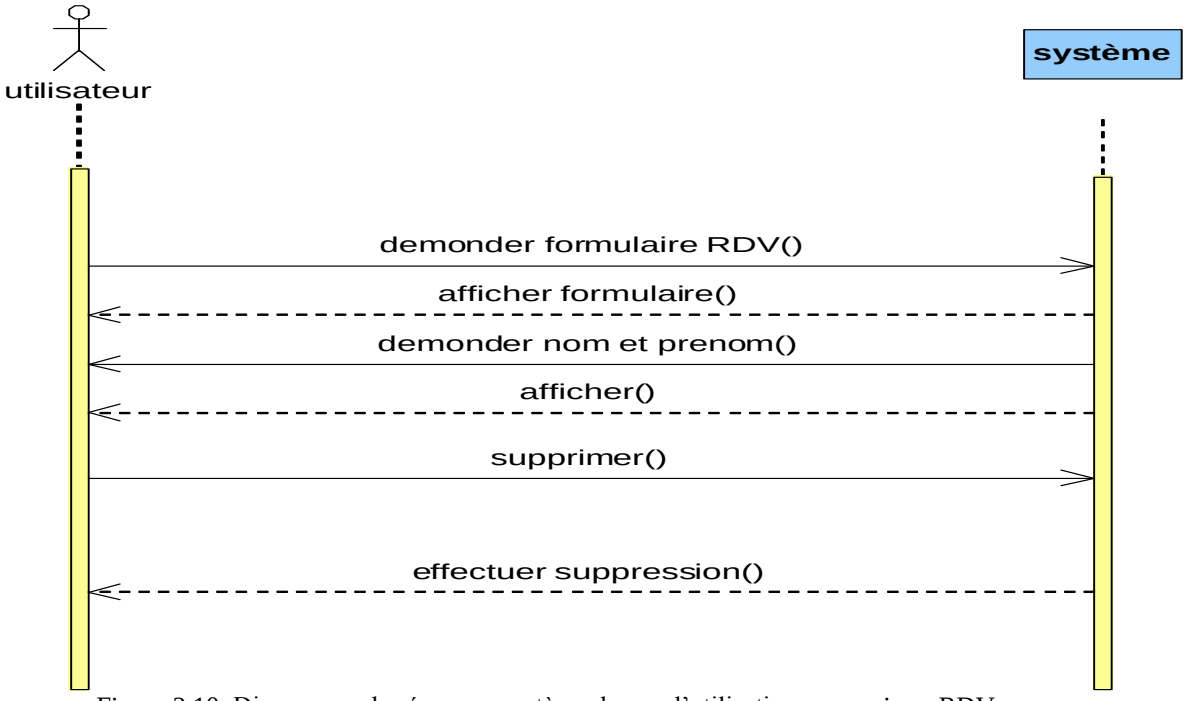


Figure 2.10. Diagramme de séquence système du cas d'utilisation « supprimer RDV »

6.11 Modifier RDV

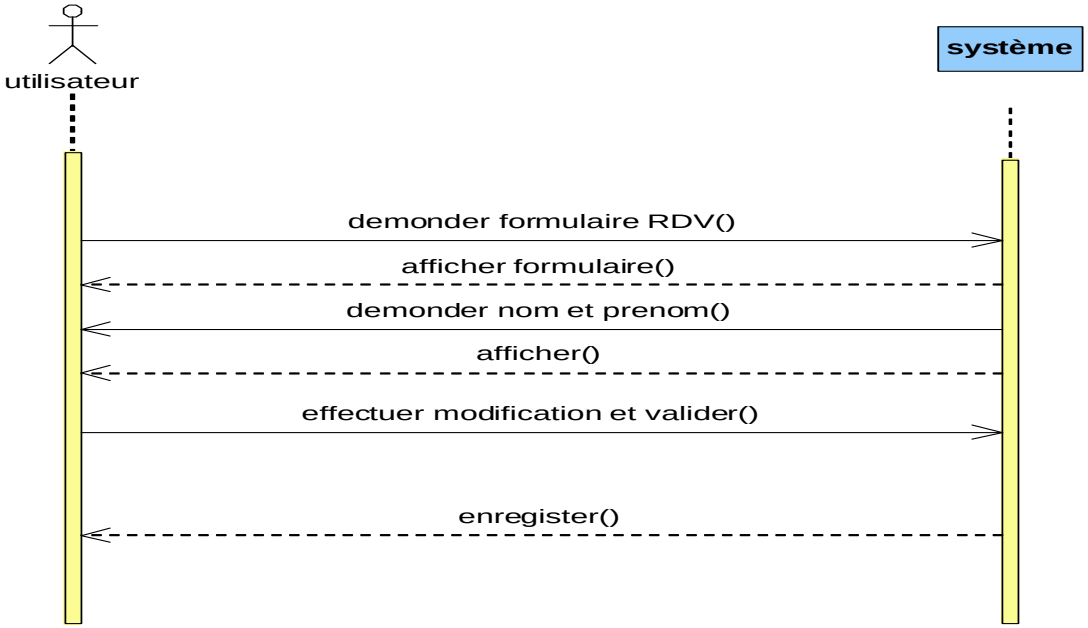


Figure 2.10. Diagramme de séquence système du cas d'utilisation « modifier RDV»

6.12 Etablir certificat :

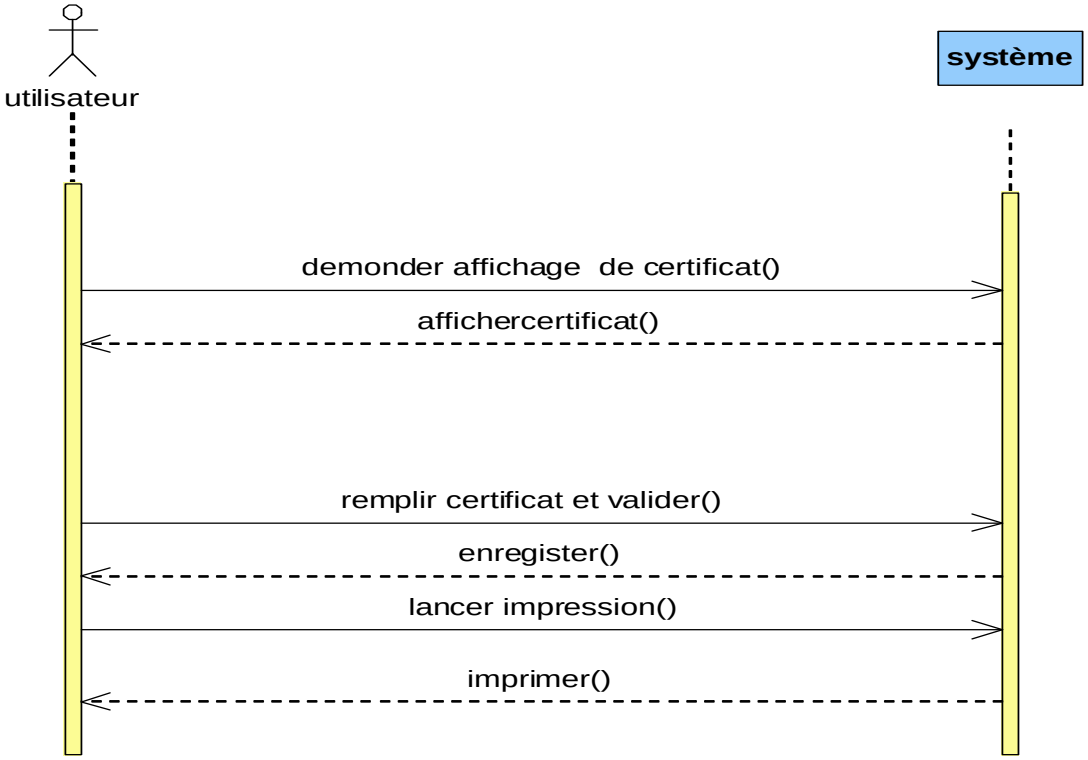


Figure 2.11. Diagramme de séquence système du cas d'utilisation « établir certificat»

6.13 Etablir l'analyse

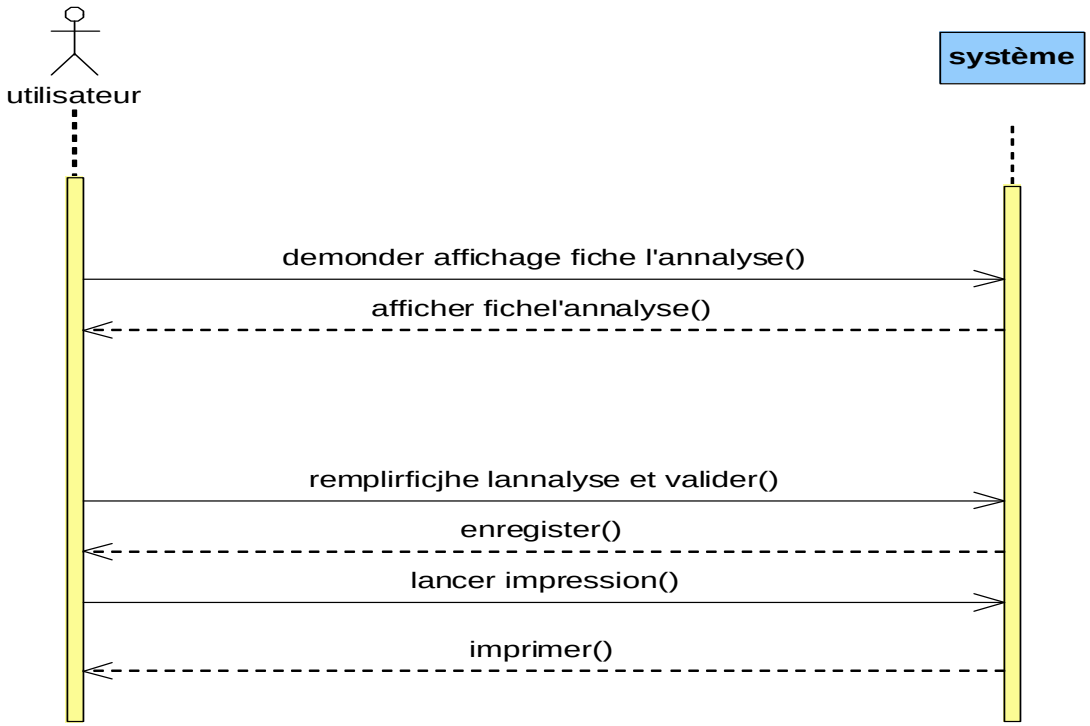


Figure 2.13. Diagramme de séquence système du cas d'utilisation « établir l'analyse»

6.14 Etablir ordonnance :

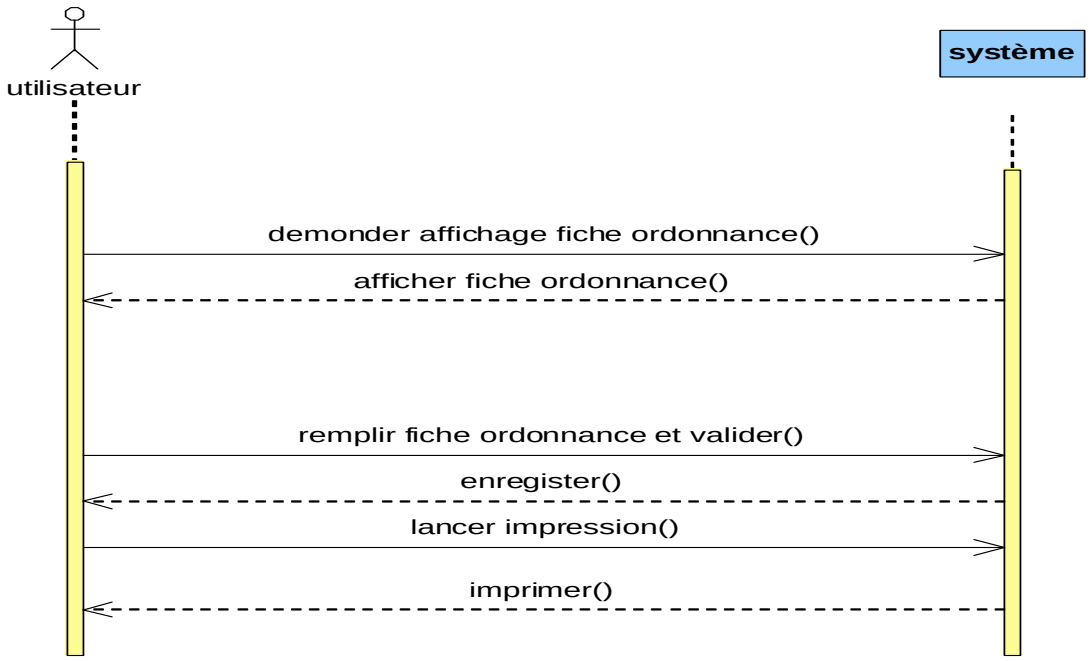


Figure 2.14. Diagramme de séquence système du cas d'utilisation « établir ordonnance »

6.15 RDV en cours :

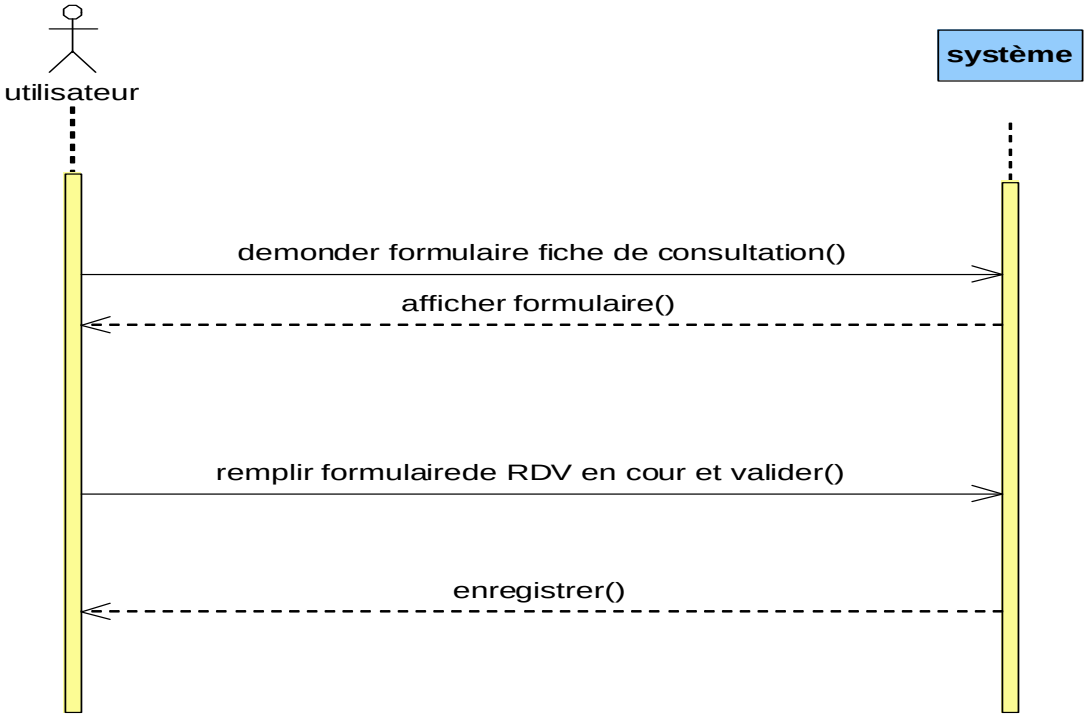


Figure 2.15. Diagramme de séquence système du cas d'utilisation « RDV en cours»

5. Conclusion

Ce chapitre étudie les besoins de notre projet et nous a permis de représenter les diagrammes de cas d'utilisation, la description des cas d'utilisation et les diagrammes de séquence système**.**

Ensuite ce chapitre prépare la phase d'analyse qui est l'objet du chapitre suivant.

CHAPITRE3

PHASE D'ANALYSE

1. Introduction

Dans cette phase nous allons élaborer une première version du diagramme de classes qui est le modèle des classes du domaine. Puis, Nous développerons les diagrammes de classes participantes qui effectuent la jonction entre, d'une part, les cas d'utilisation, le modèle du domaine et la maquette, et d'autre part, les diagrammes de conception logicielle qui sont les diagrammes de séquence et le diagramme de classes de conception. Enfin, nous représenterons l'activité de navigation dans l'interface de la future application en produisant des diagrammes d'activités de navigation.

2. Démarche

La conception objet demande principalement une description structurelle, statique, du système à réaliser, sous forme d'un ensemble de classes logicielles. Les classes candidates sont celles issues d'une analyse de domaine, effectivement ces concepts peuvent être identifiés directement à partir de la connaissance du domaine. Détaillons les différentes étapes de la démarche que nous allons suivre :

- Identifier les concepts du domaine ;
- Ajouter les associations entre classes ;
- Ajouter les attributs

3. Identification des concepts de domaine

Nous allons prendre les cas d'utilisations un par un et poser pour chacun la question suivante : quel sont les concepts métier qui participent à ce cas d'utilisation ?

S'authentifier

- ✓ Utilisateur

Crée fiche de consultation

- ✓ Utilisateur
- ✓ fiche de patient

Modifier fiche de consultation

- ✓ Utilisateur

- ✓ Fiche de consultation

Ajout fiche de patient

- ✓ Utilisateur
- ✓ Fiche patient

Supprimer fiche de patient

- ✓ Utilisateur
- ✓ Fiche patient

Modifier le fiche de patient

- ✓ Utilisateur
- ✓ Fiche patient

Ajout un RDV

- ✓ Utilisateur
- ✓ fiche Consultation

Supprimer un RDV

- ✓ Utilisateur
- ✓ RDV

Modifier un RDV

- ✓ Utilisateur
- ✓ RDV

Établir la facture

- ✓ Utilisateur
- ✓ Fiche Consultation

Etablir l'analyse

- ✓ Utilisateur
- ✓ fiche Consultation

Etablir ordonnance

- ✓ Utilisateur
- ✓ Fiche consultation

4. Modèle de domaine

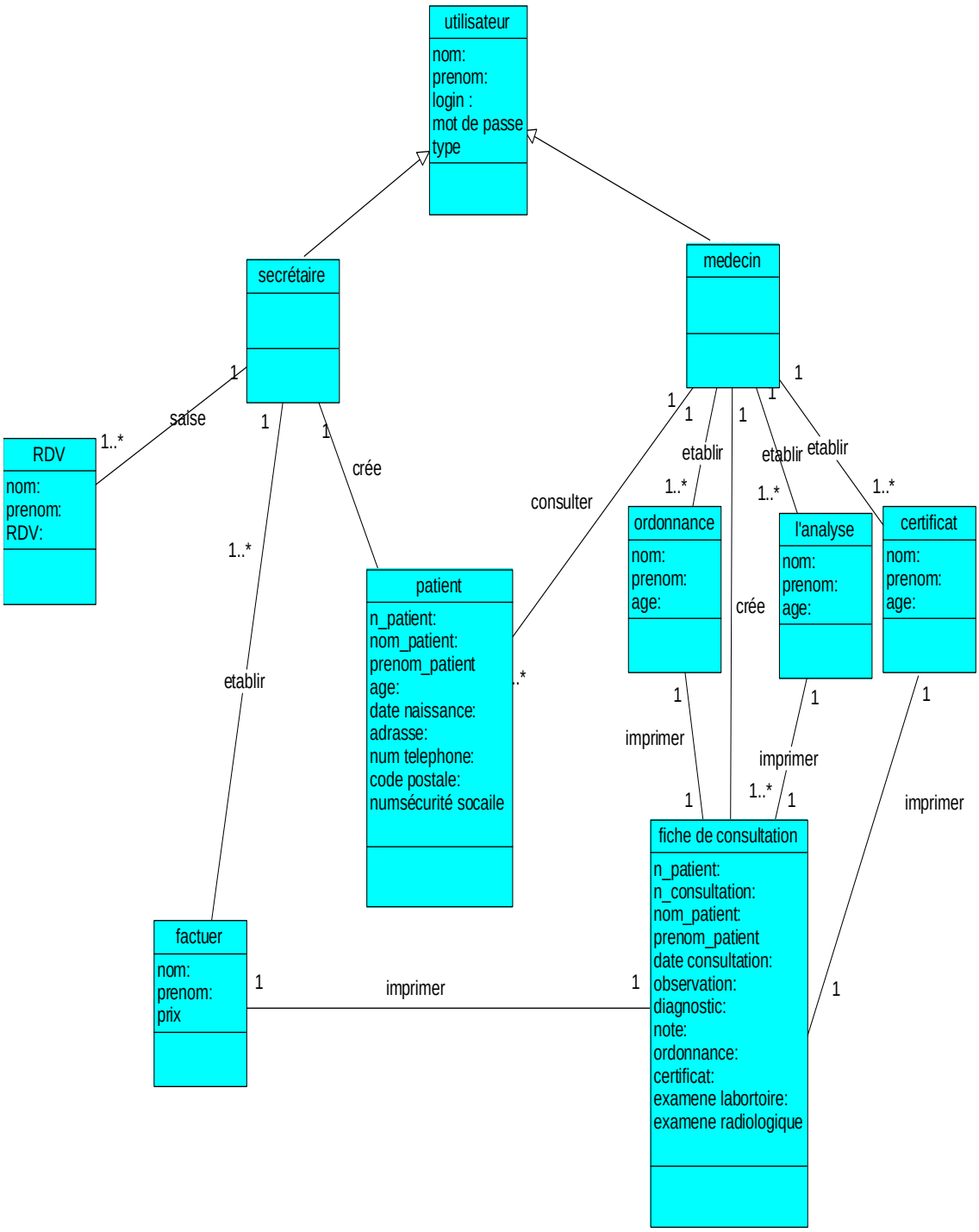


Figure 3.1 Modèle de domaine

5. Les diagrammes de classes participantes des cas d'utilisations

5.1 S'authentifier

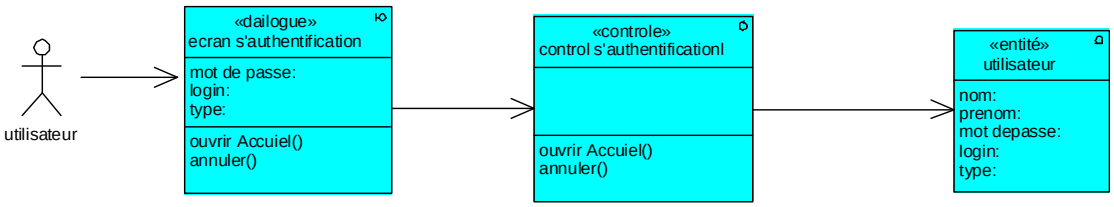


Figure3.2 Diagramme de classe participation du cas d'utilisation <<s'authentification>>

5.2 Créer fiche de consultation

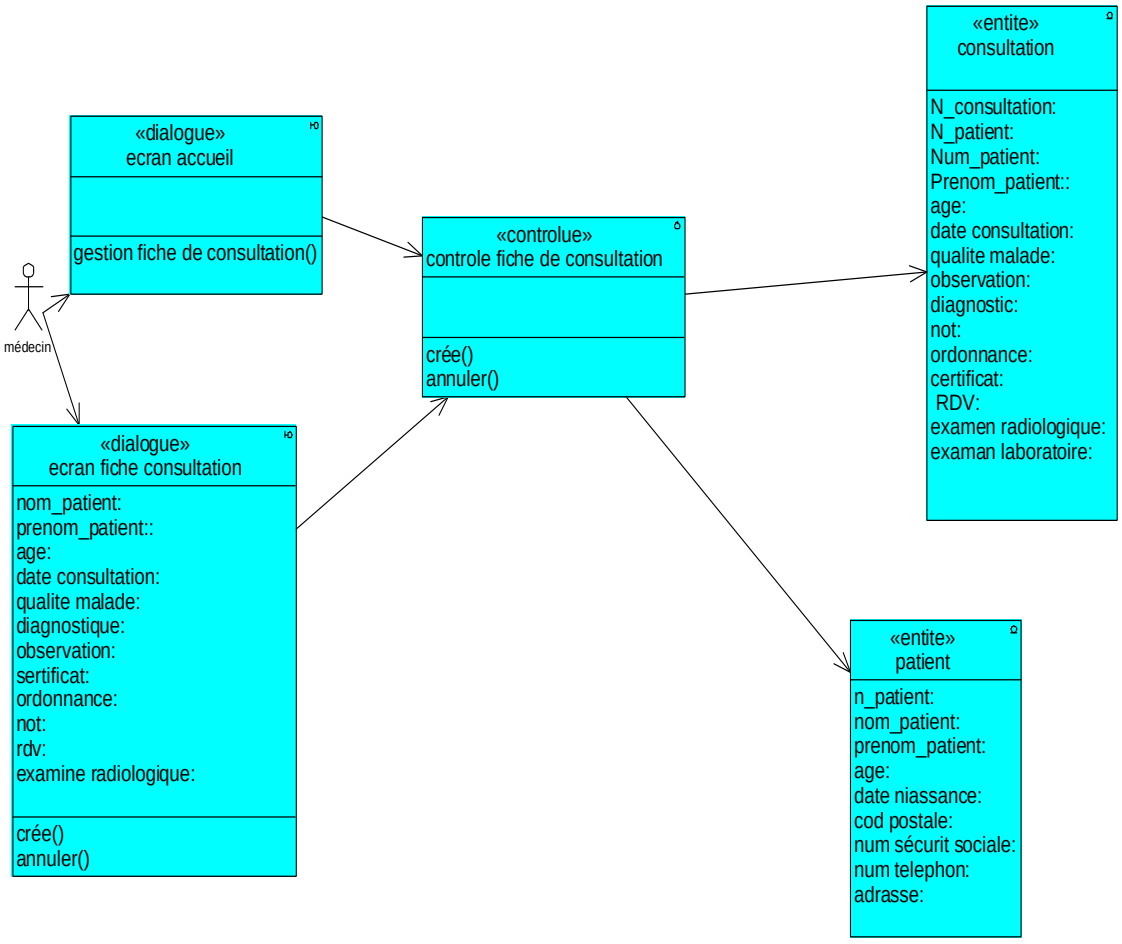


Figure 3.3 Diagramme de classe participation du cas d'utilisation<<crée fiche consultation>>

5.3 Supprimer fiche de consultation

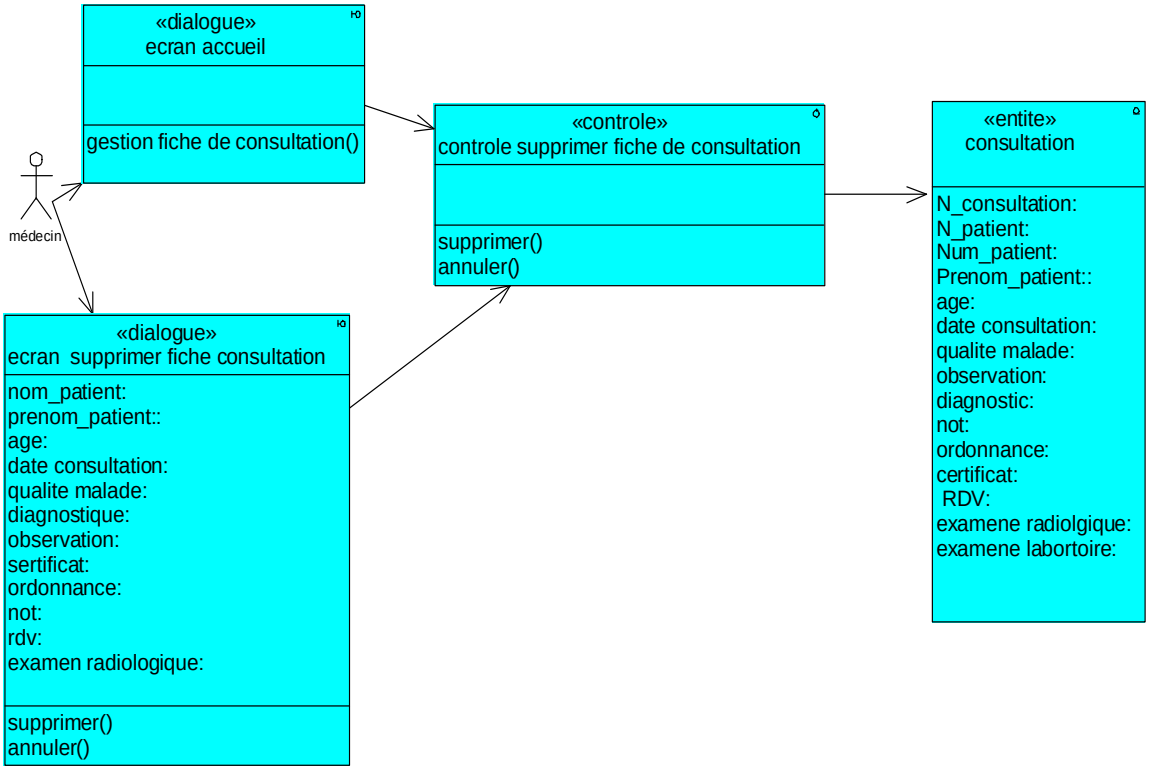


Figure 3.4 Diagramme de classe participation du cas d'utilisation<<supprimer fiche consultation>>

5.4 Modifier fiche de consultation

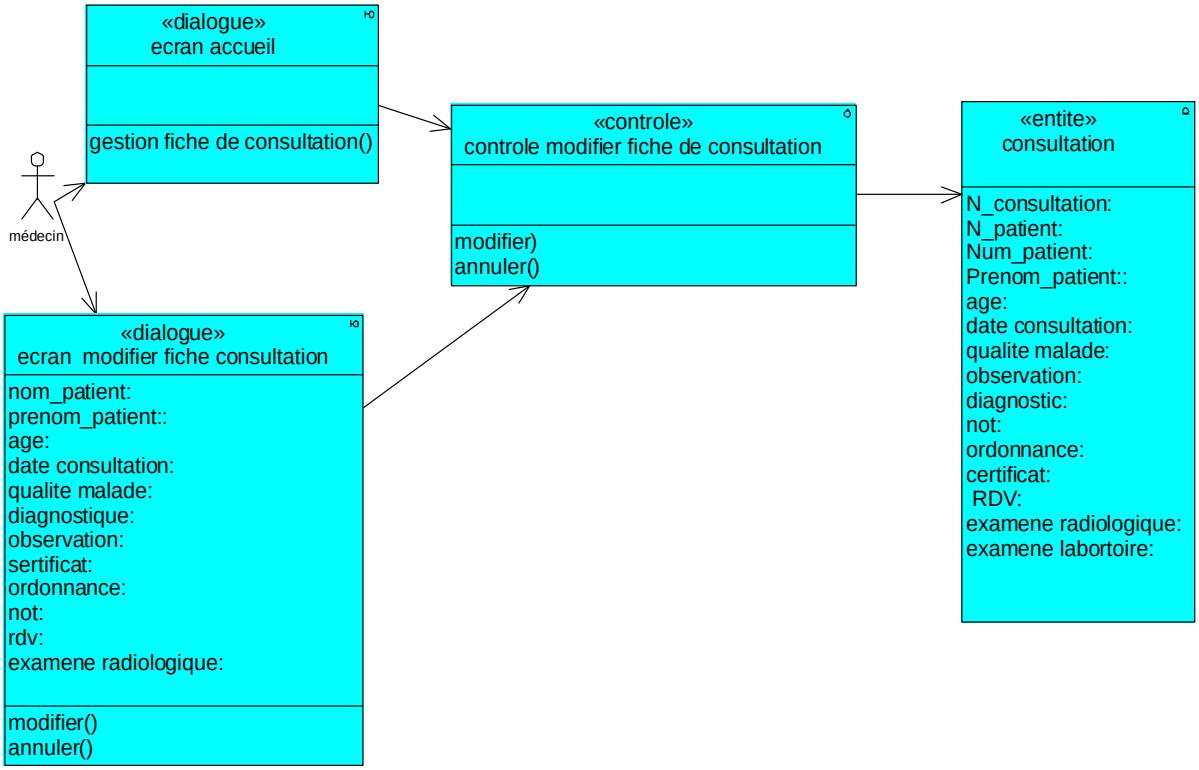


Figure 3.5 Diagramme de classe participation du cas d'utilisation<<modifier fiche consultation>>

5.5 Créer fiche du patient

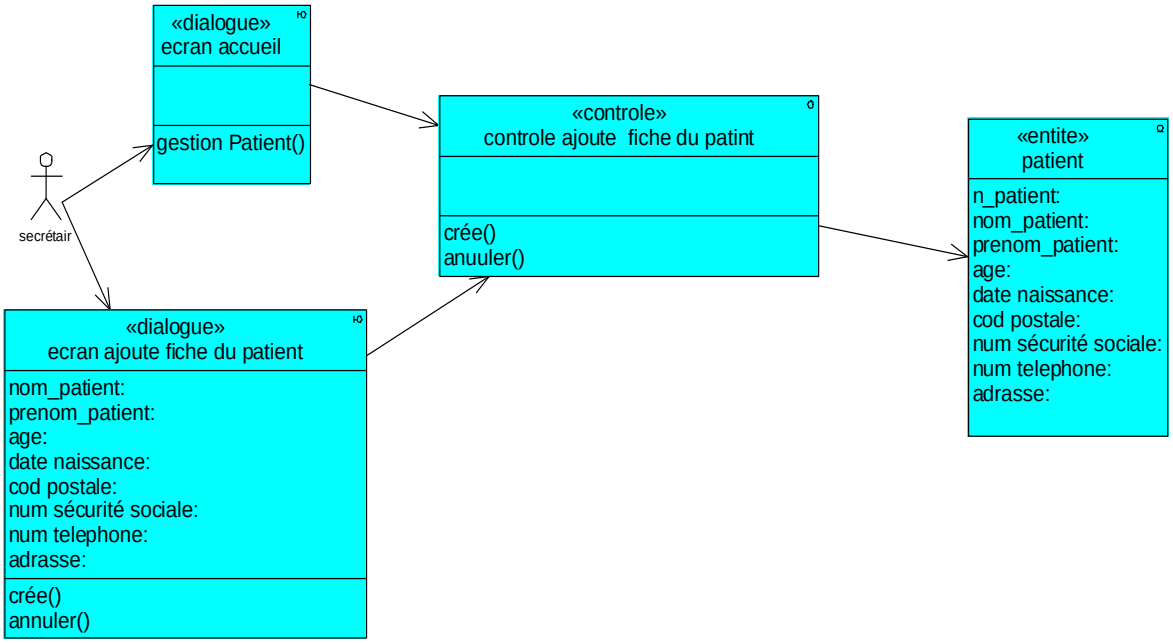


Figure 3.6 Diagramme de classe participation du cas d'utilisation<<crée patient >>

5.6 modifier fiche de patient

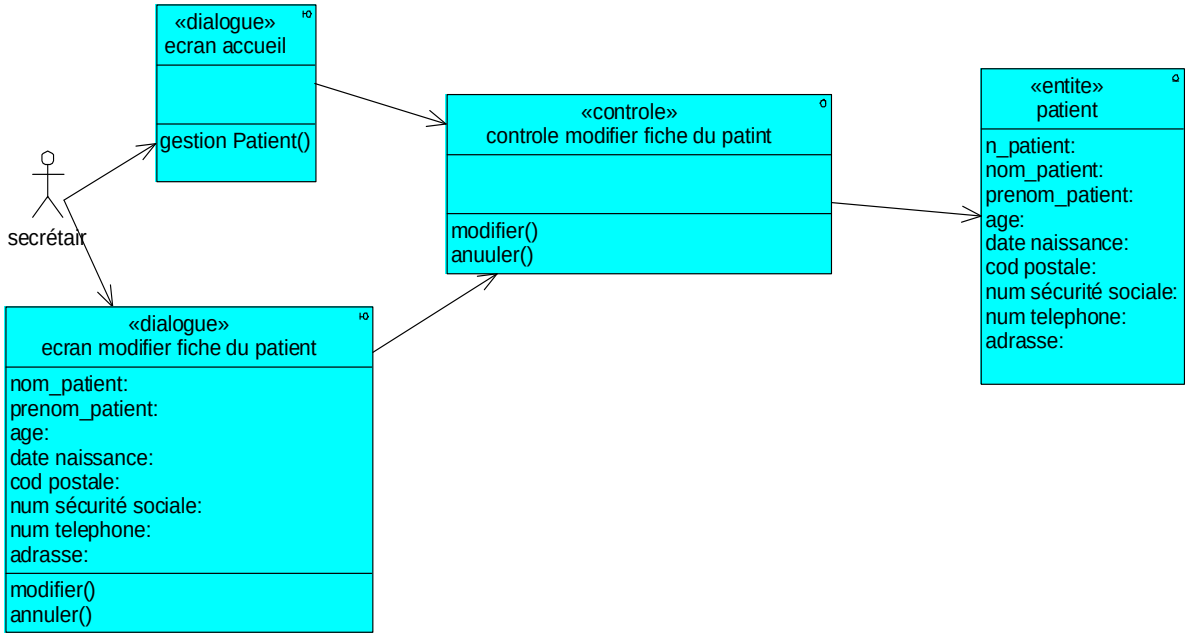


Figure 3.7 Diagramme de classe participation du cas d'utilisation<<modifier patient >>

5.7 supprimer fiche patient

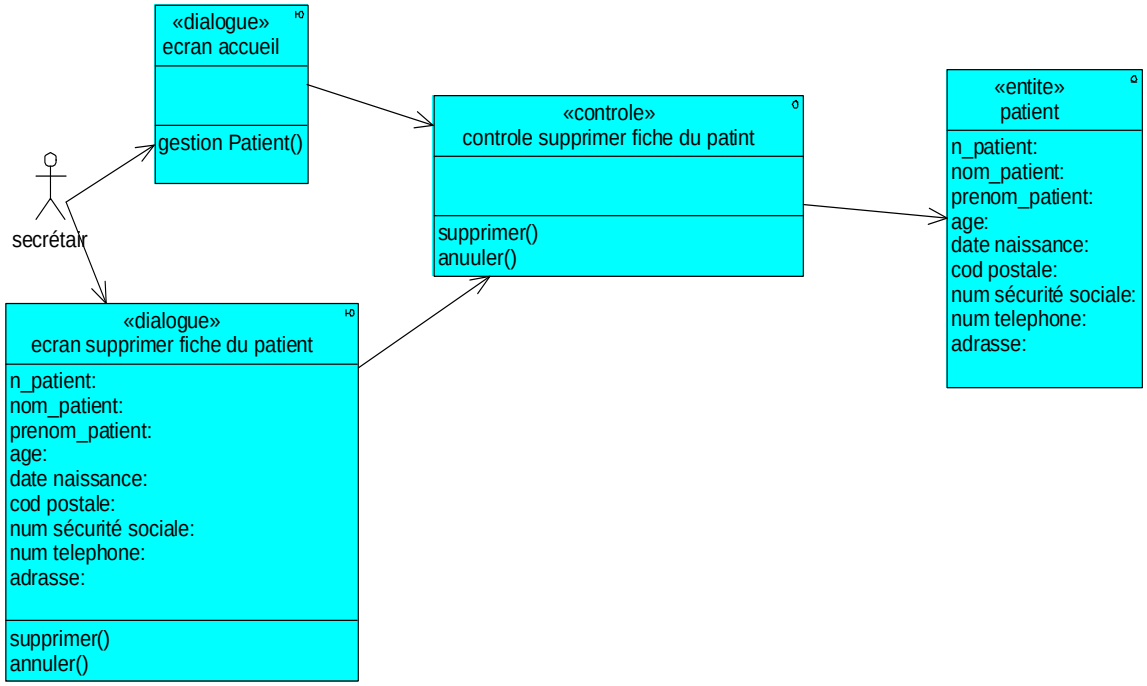


Figure 3.8 Diagramme de classe participation du cas d'utilisation<<supprimer patient >>

5.8 Établir facture

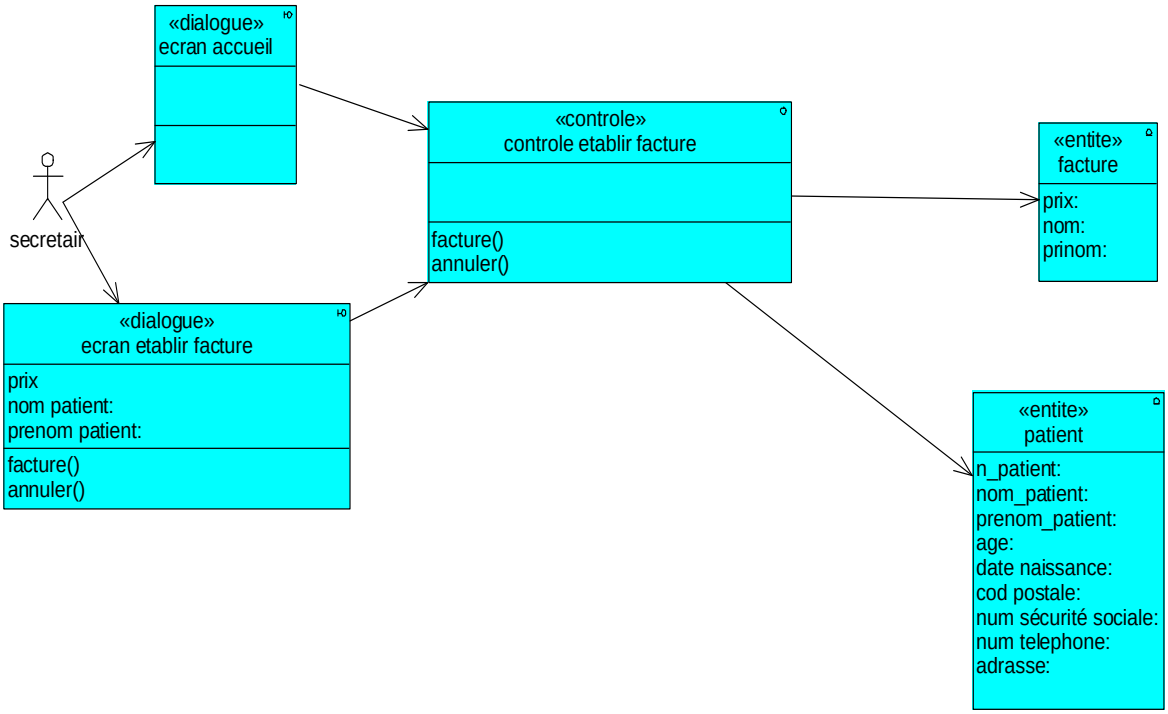


Figure 3.9 Diagramme de classe participation du cas d'utilisation<<établir facture >>

5.9 Établir l'analyse

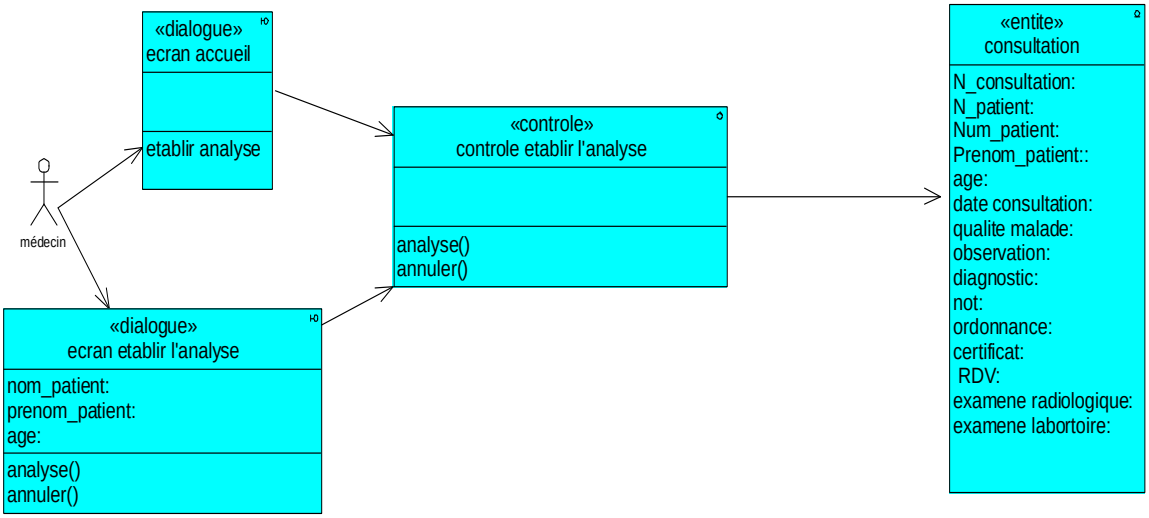


Figure 3.10 Diagramme de classe participation du cas d'utilisation<<établir l'analyse>>

5.10 Établir ordonnance

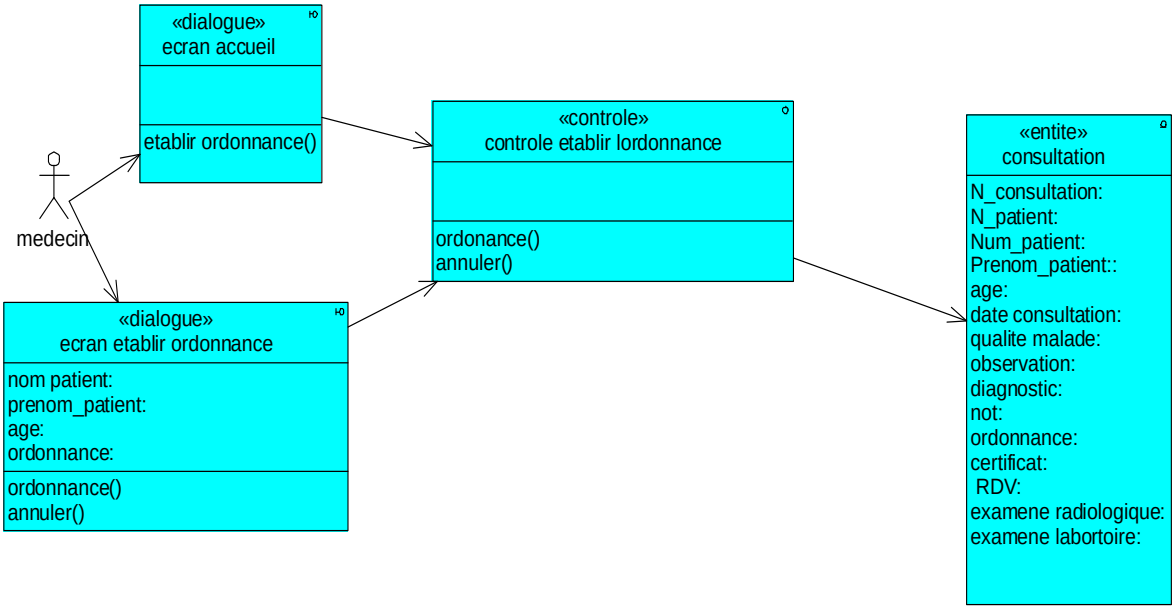


Figure 3.11 Diagramme de classe participation du cas d'utilisation<<établir ordonnance>>

5.11 Ajouter RDV

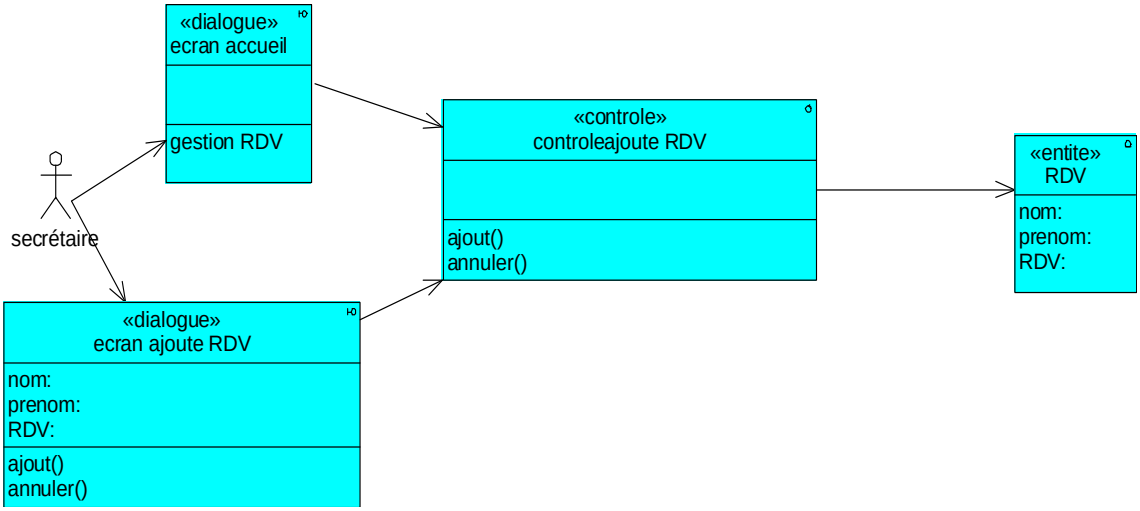


Figure 3.12 Diagramme de classe participation du cas d'utilisation<<ajout RDV>>

5.12 Supprimer RDV

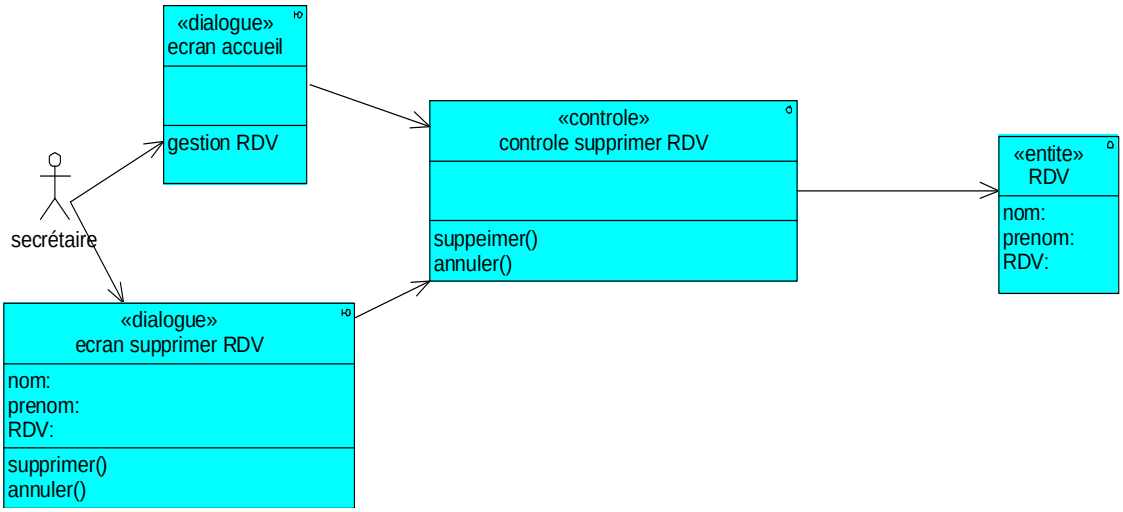


Figure 3.13 Diagramme de classe participation du cas d'utilisation<<supprimer RDV>>

5.13 Modifier RDV

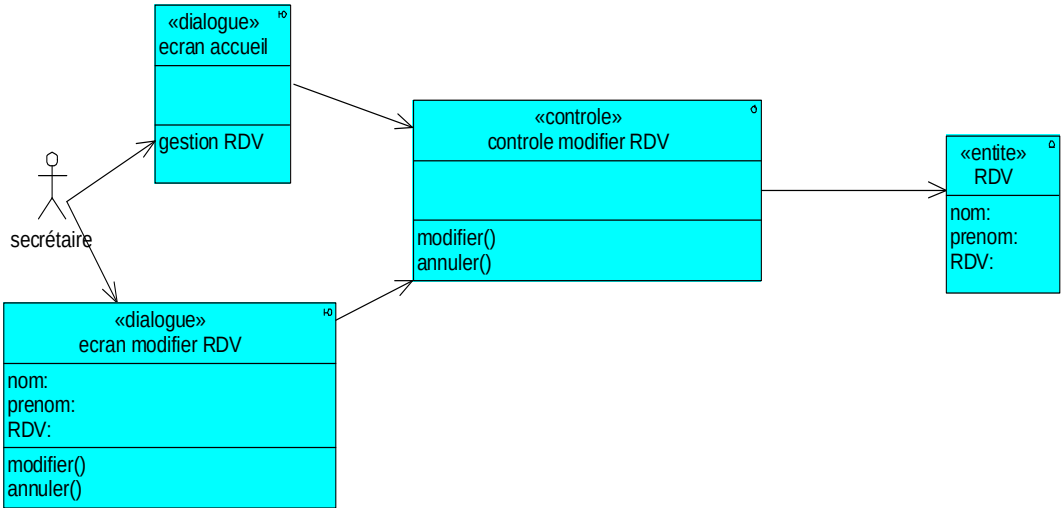


Figure 3.14 Diagramme de classe participation du cas d'utilisation<< RDV en cours>>

5.14 RDV en cours

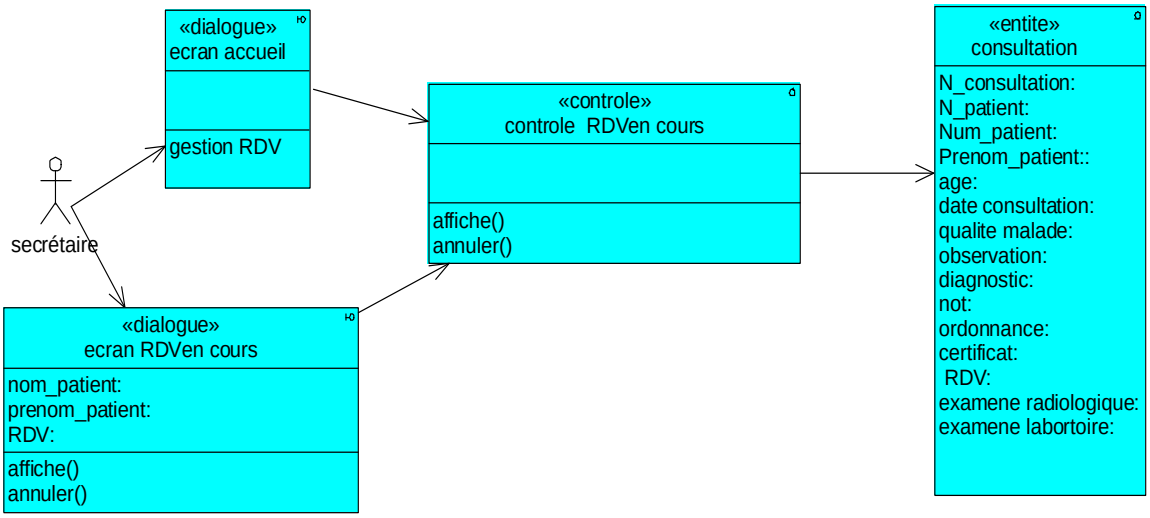


Figure 3.15 Diagramme de classe participation du cas d'utilisation<< RDV en cours>>

6. Les diagrammes d'activités de navigation

6.1 S'authentifier

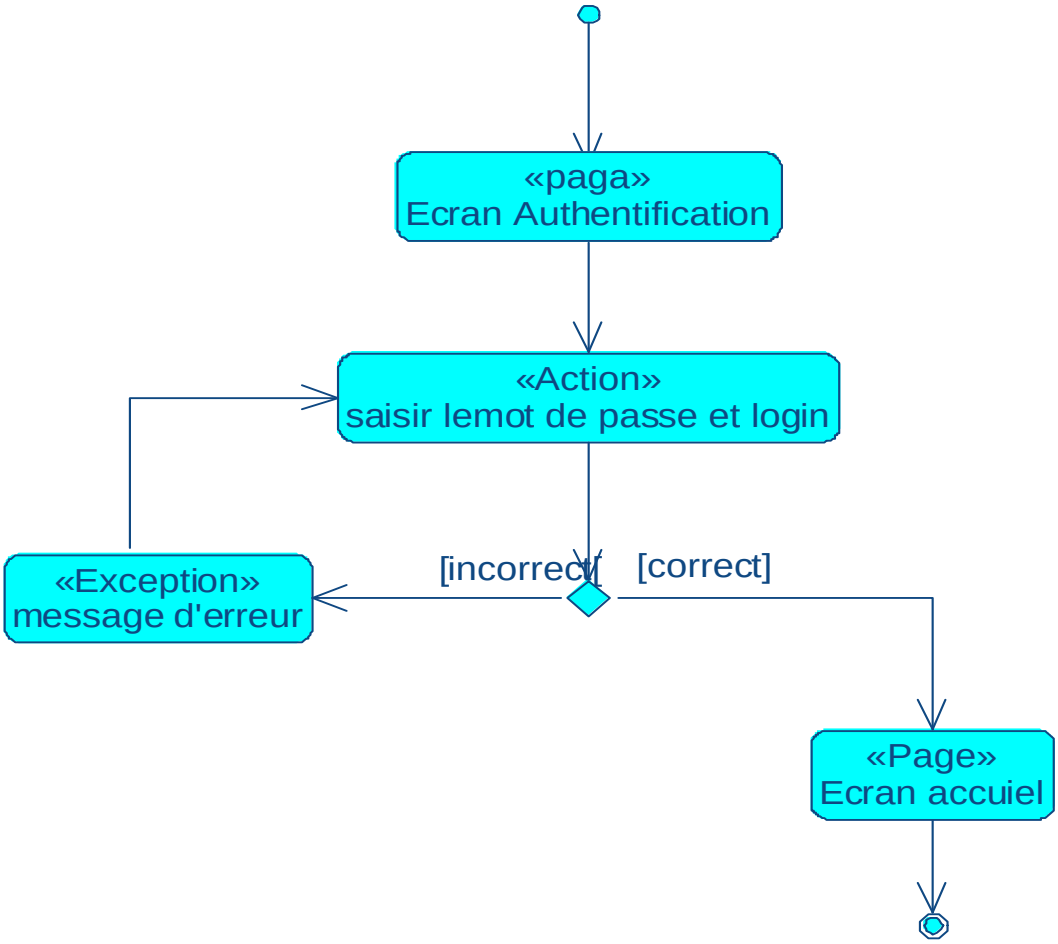


Figure 3.16 Diagramme d'activité de navigation du cas d'utilisation <<s'authentification>>

6.2 Crée fiche de consultation

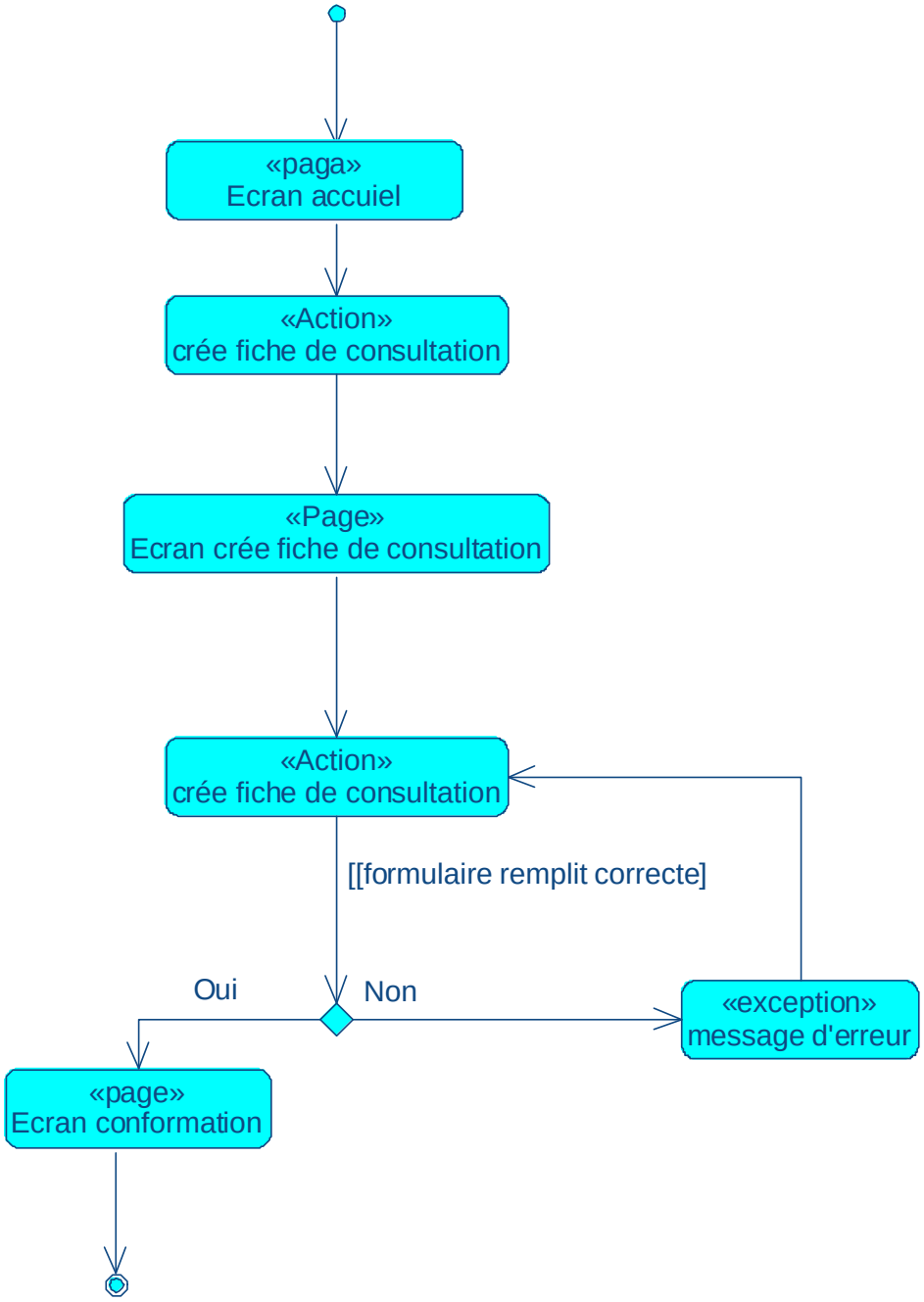


Figure 3.17 Diagramme d'activité de navigation du cas d'utilisation<<crée fiche consultation>>

6.3 Modifier fiche de consultation

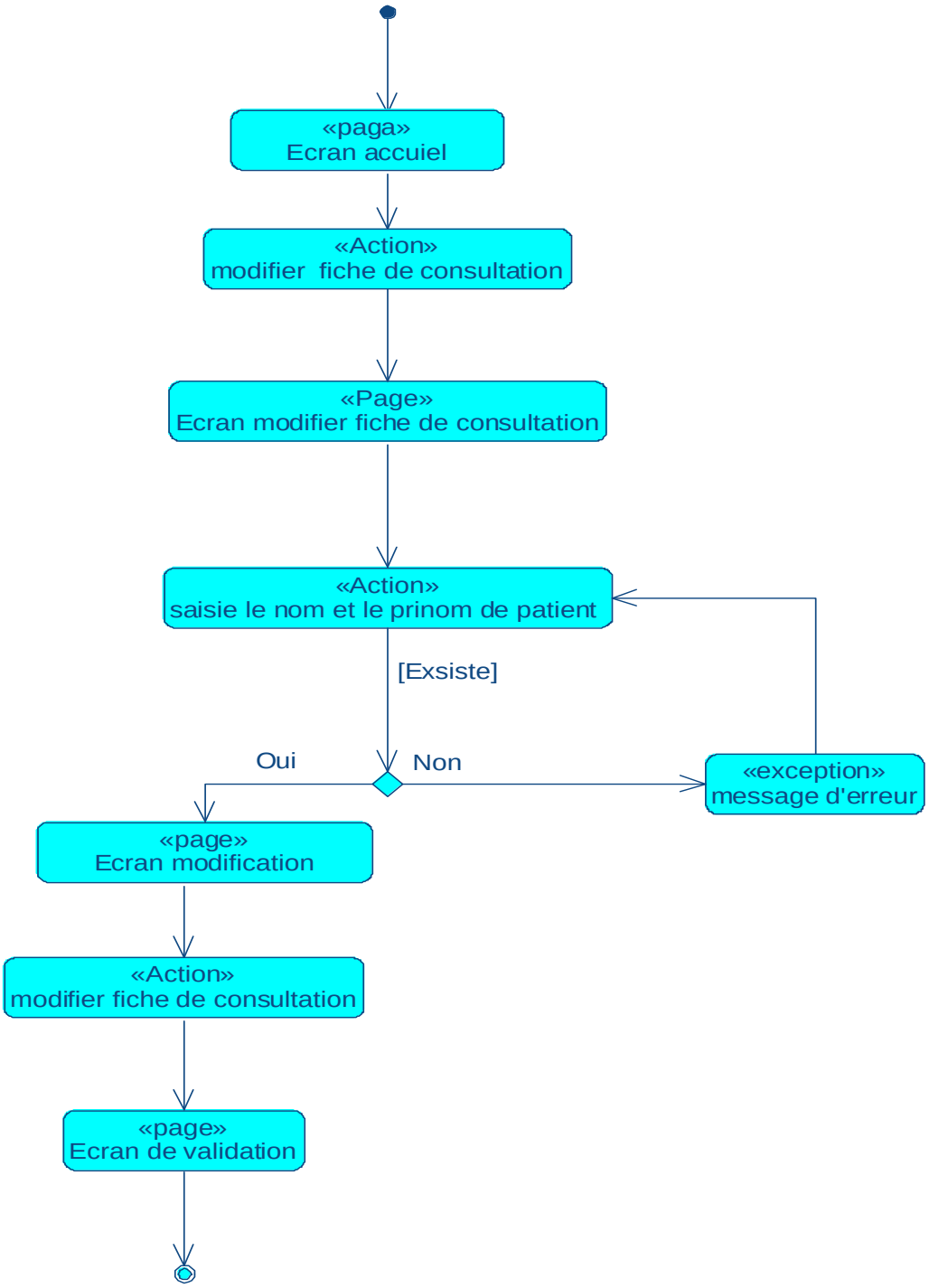


Figure 3.18 Diagramme d'activité de navigation du cas d'utilisation<<modifier fiche consultation>>

6.4Supprime fiche de consultation

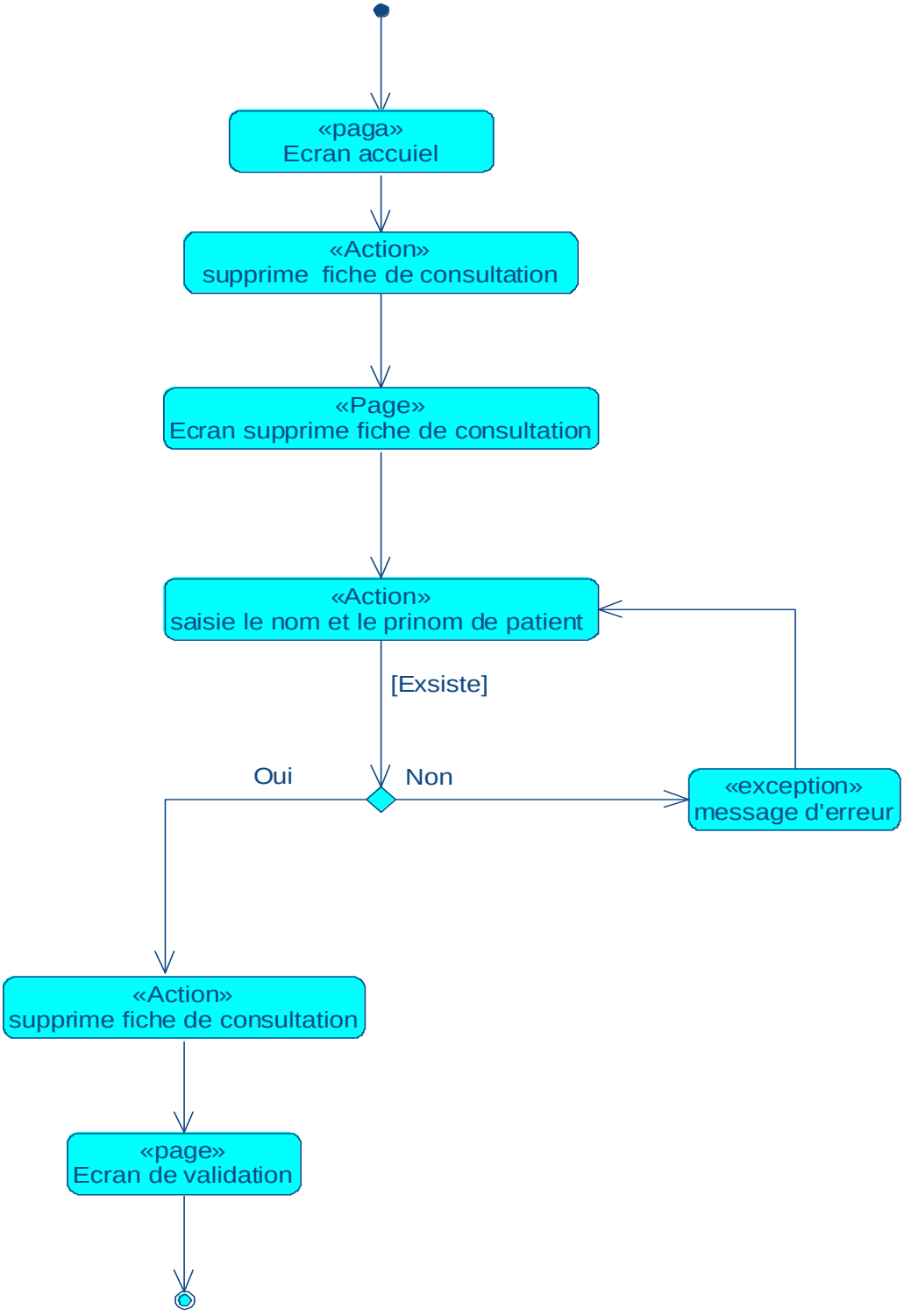


Figure 3.19 Diagramme d'activité de navigation du cas d'utilisation<<supprimer fiche consultation>>

6.5Ajout patient

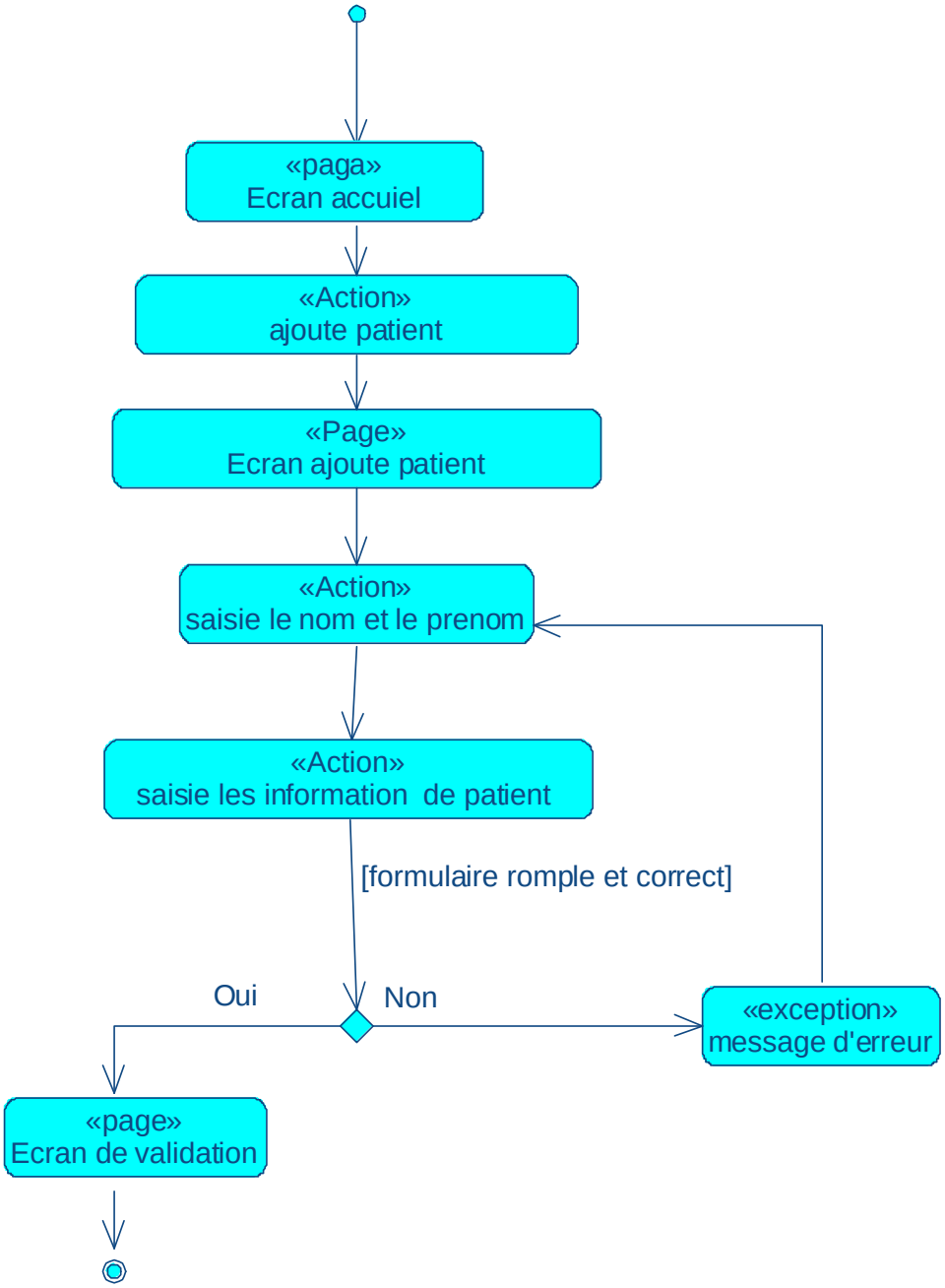


Figure 3.20 Diagramme d'activité de navigation du cas d'utilisation<<ajoute patient>>

6.6 Supprime patient

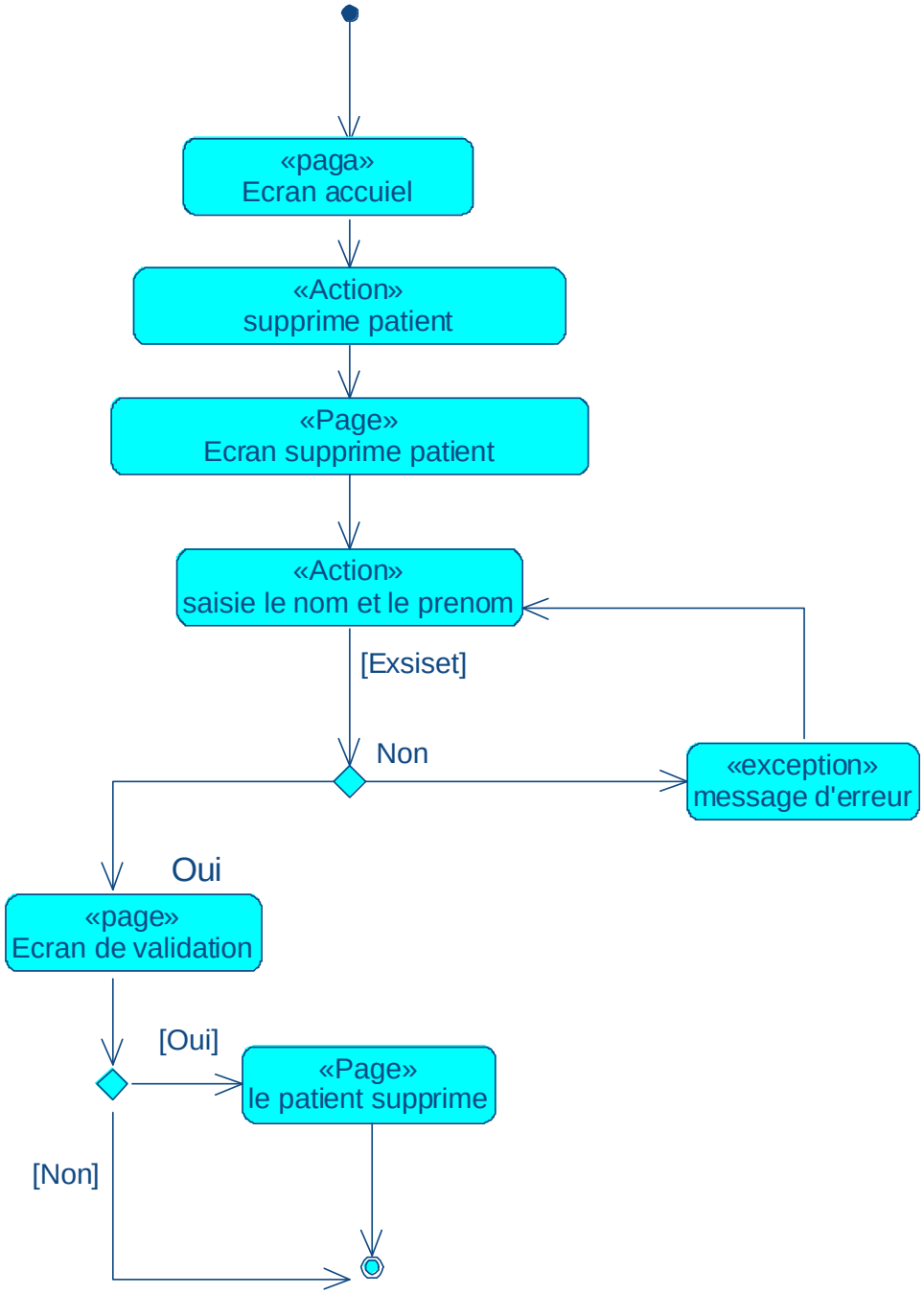


Figure 3.21 Diagramme d'activité de navigation du cas d'utilisation<<supprimer patient>>

6.7 Modifier patient

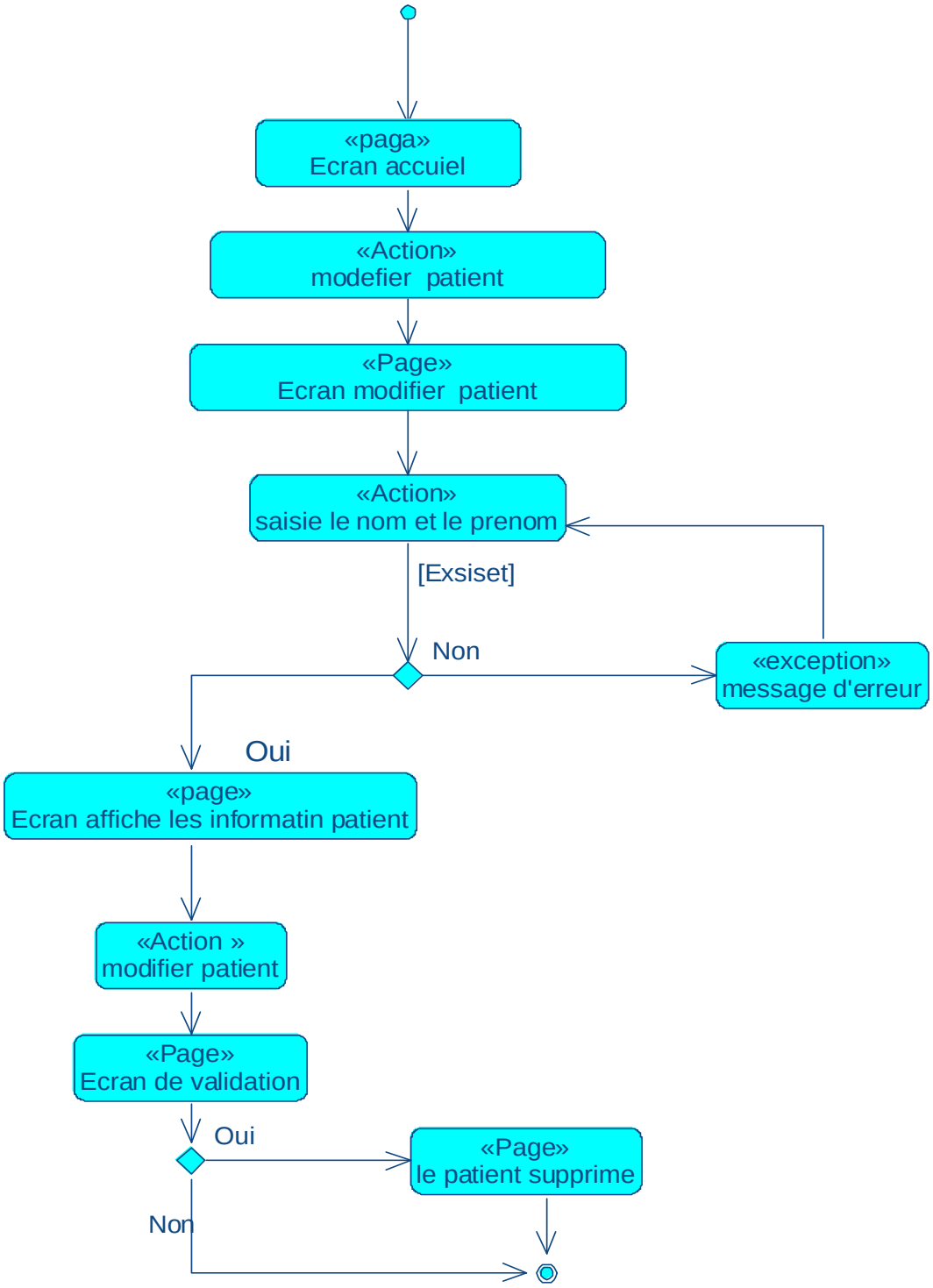


Figure 3.22 Diagramme d'activité de navigation du cas d'utilisation<<modifier patient>>

6.8 Établir ordonnance

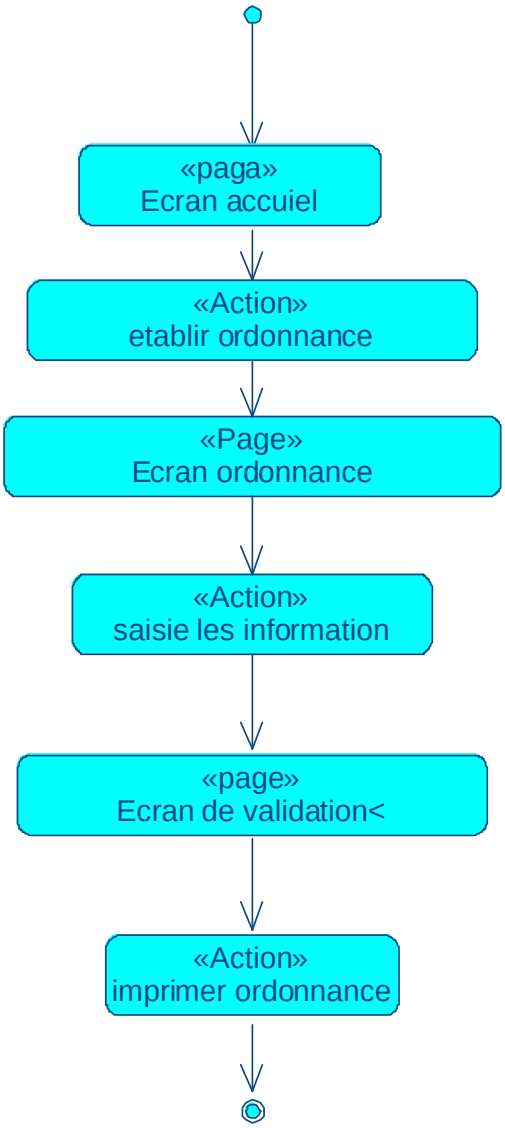


Figure 3.23 Diagramme d'activité de navigation de cas d'utilisation<<établir ordonnance>>

6.9 Établir analyse

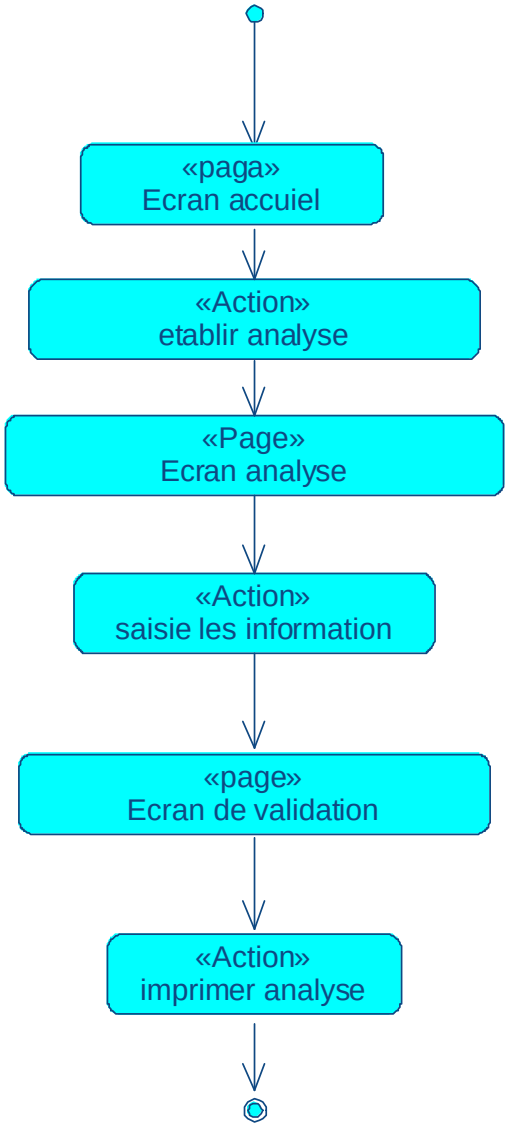


Figure 3.24 Diagramme d'activité de navigation du cas d'utilisation<<établir analyse>>

6.10 Etablir certificat

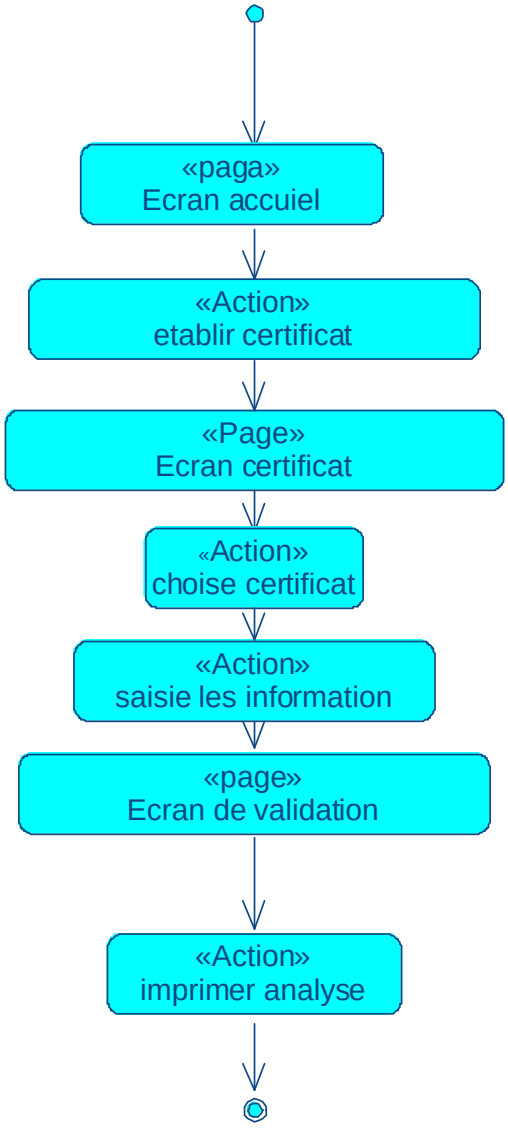


Figure 3.25 Diagramme d'activité de navigation du cas d'utilisation<<établir certificat>>

6.11 Ajoute RDV

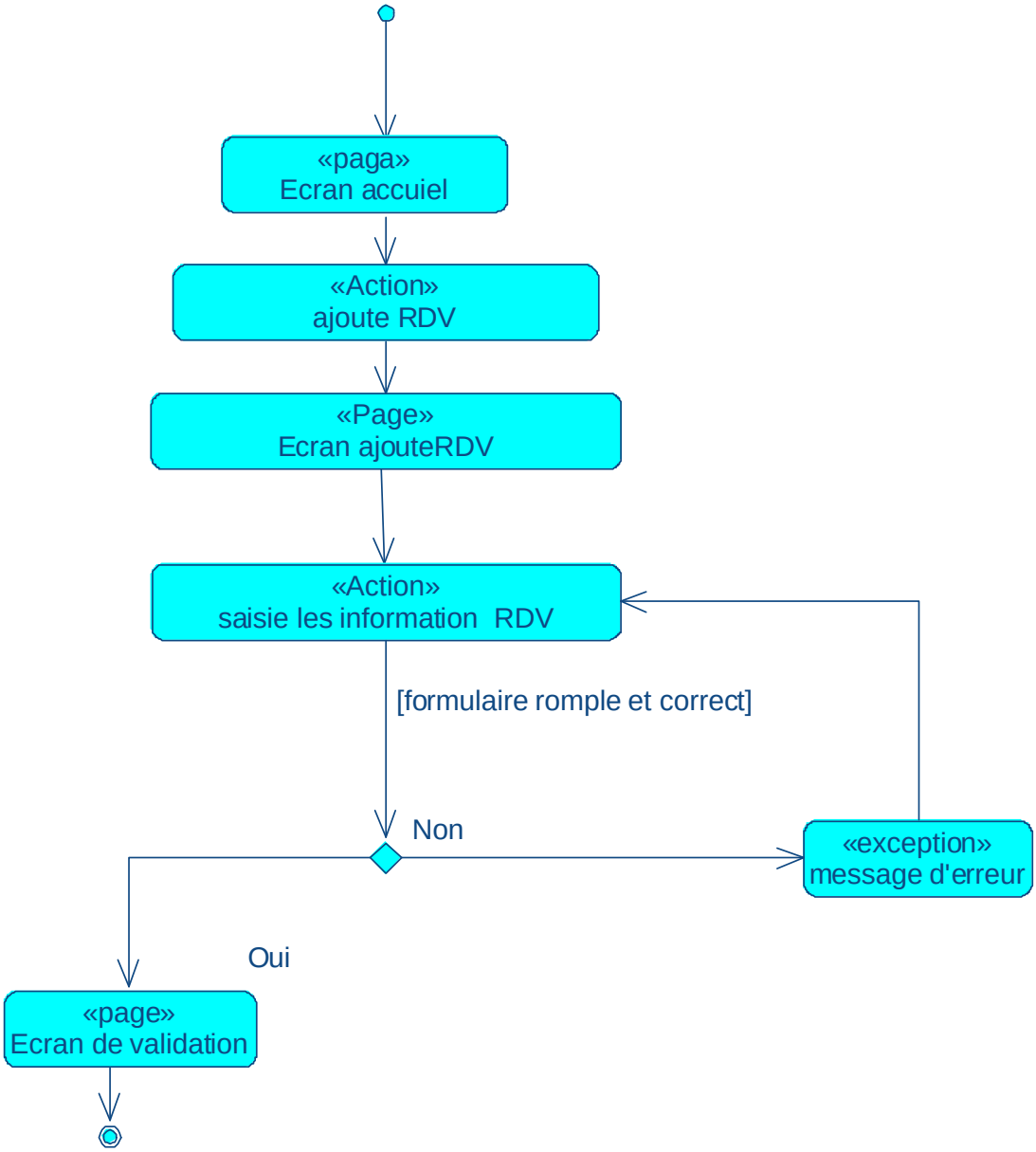


Figure 3.26 Diagramme d'activité de navigation du cas d'utilisation<<ajoute un RDV>>

6.12 Supprime RDV

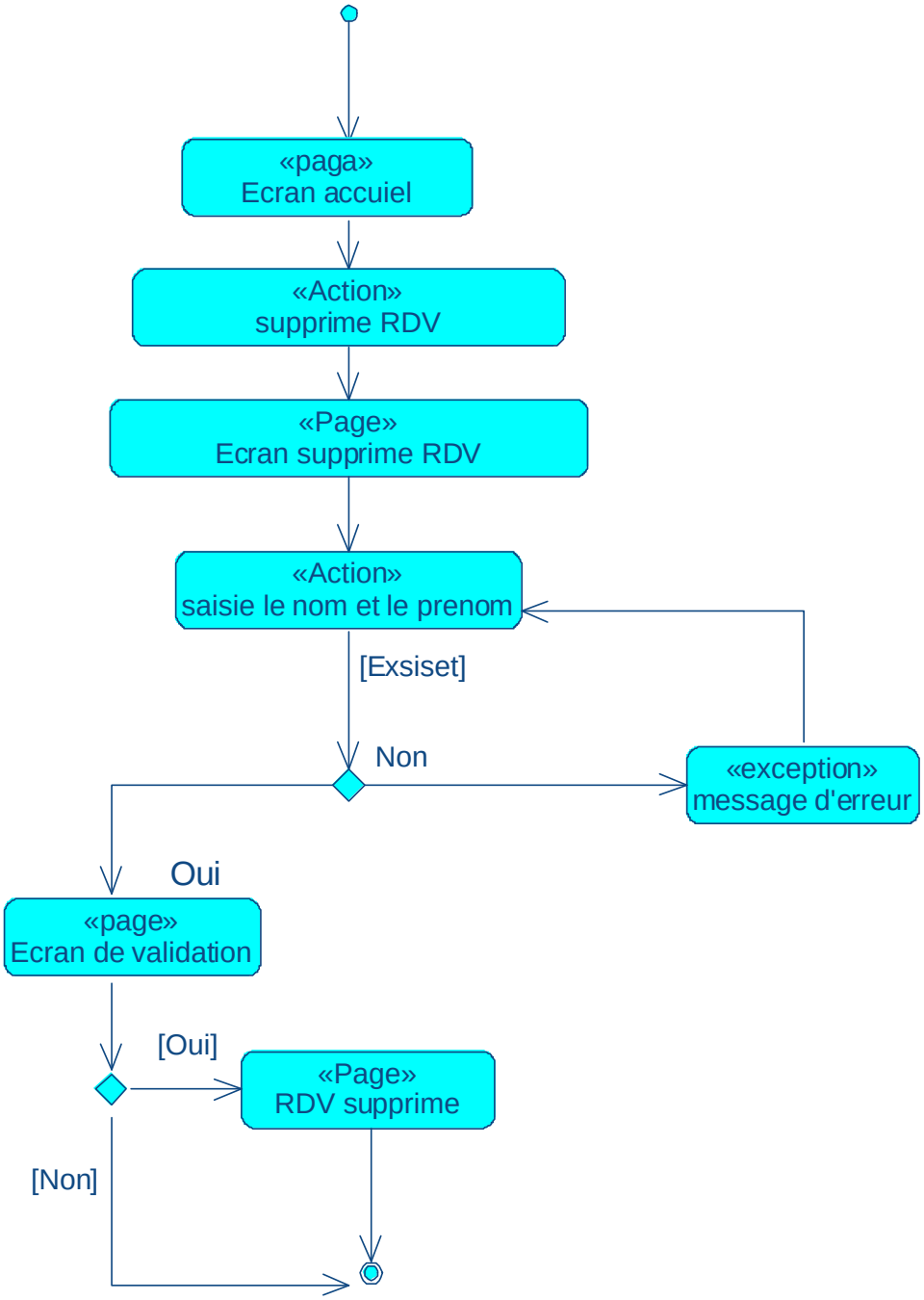


Figure 3.27 Diagramme d'activité de navigation du cas d'utilisation<<supprimer un RDV>>

6.13 Modifier RDV

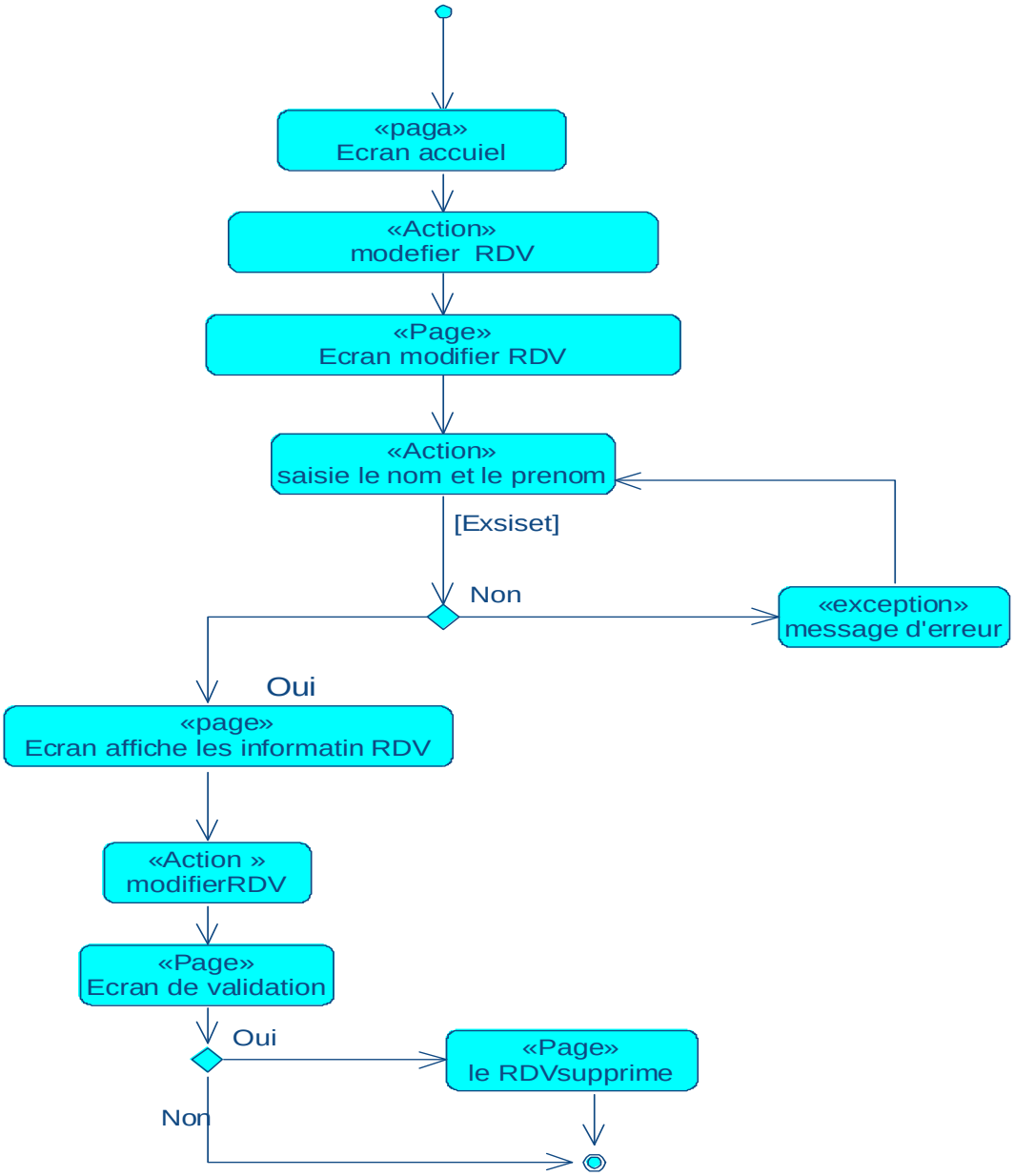


Figure 3.28 Diagramme d'activité de navigation du cas d'utilisation<<modifier un RDV>>

Conclusion : Dans ce chapitre, nous avons élaboré le modèle de domaine et les diagrammes de classes participantes des cas d'utilisations, ainsi que les diagrammes d'activités de navigation. Ce chapitre prépare la phase de conception qui est l'objet du chapitre suivant.

CHAPITRE4

PHASE DE CONCEPTION

1. Introduction

Dans ce chapitre nous allons élaborer les diagrammes de séquence qui nous permettent d'attribuer précisément les responsabilités de comportement aux classes d'analyse du diagramme de classes participantes. Parallèlement, une première ébauche de la vue statique de conception, c'est-à-dire du diagramme de classes de conception, est construite et complétée.

2. Diagrammes d'interactions

2.1S'authentifier

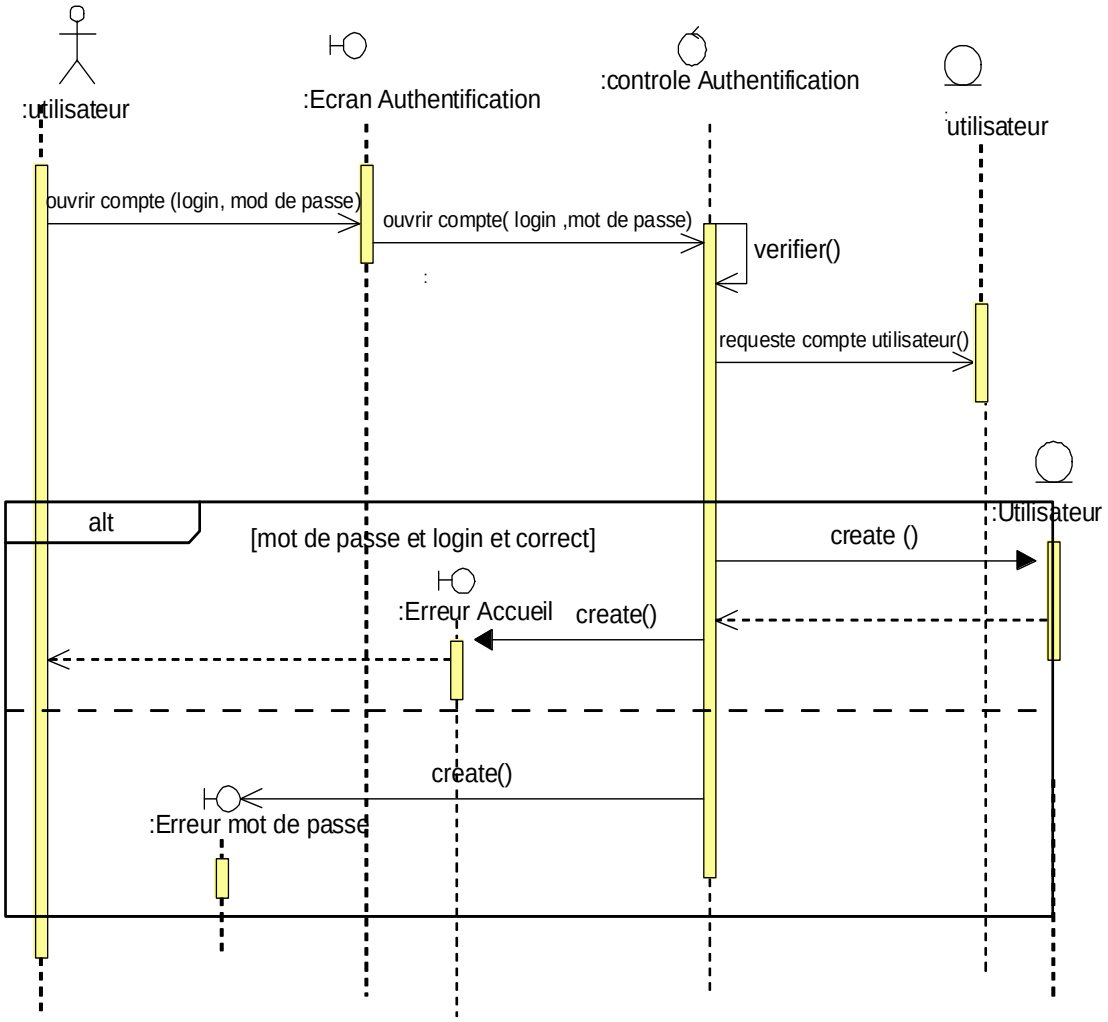


Figure 4.1 diagramme d'interactions du cas d'utilisation<<s'authentifier>>

2.2 Créer fiche de consultation

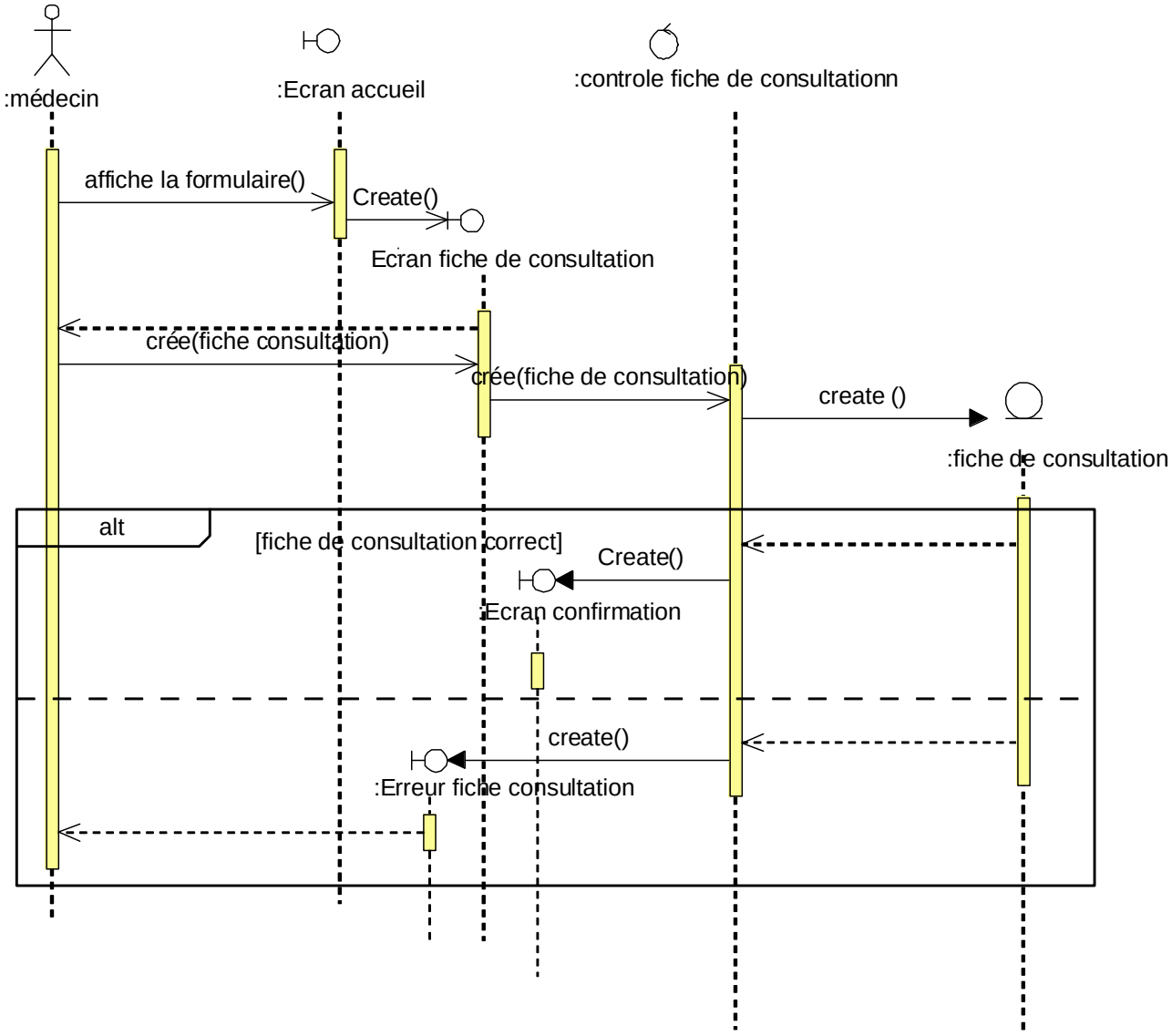


Figure 4.2 diagramme d'interactions du cas d'utilisation <<crée fiche de consultation >>

2.3 Modifier fiche de consultation

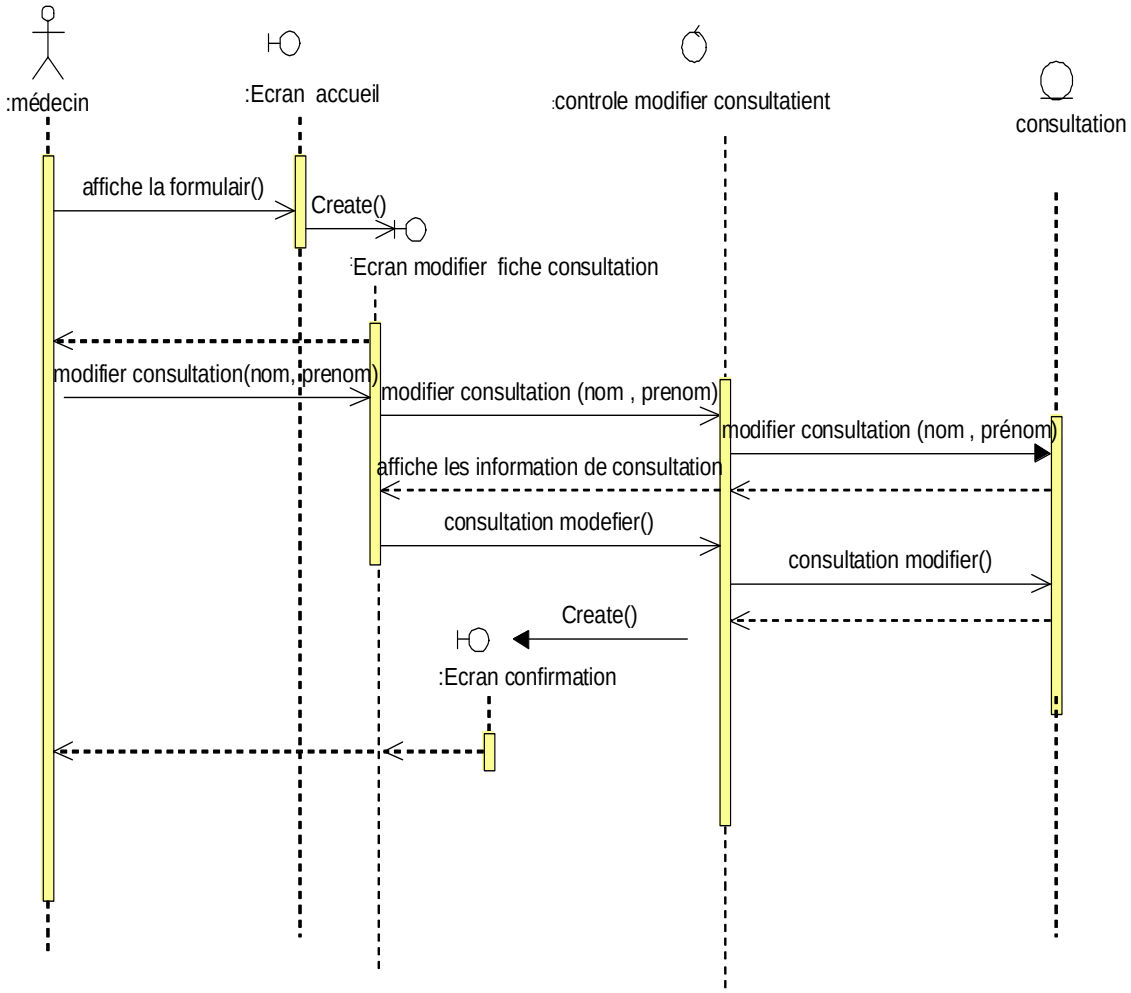


Figure 4.3 diagramme d'interactions du cas d'utilisation <<modifier fiche consultation >>

2.4 Supprimer fiche de consultation

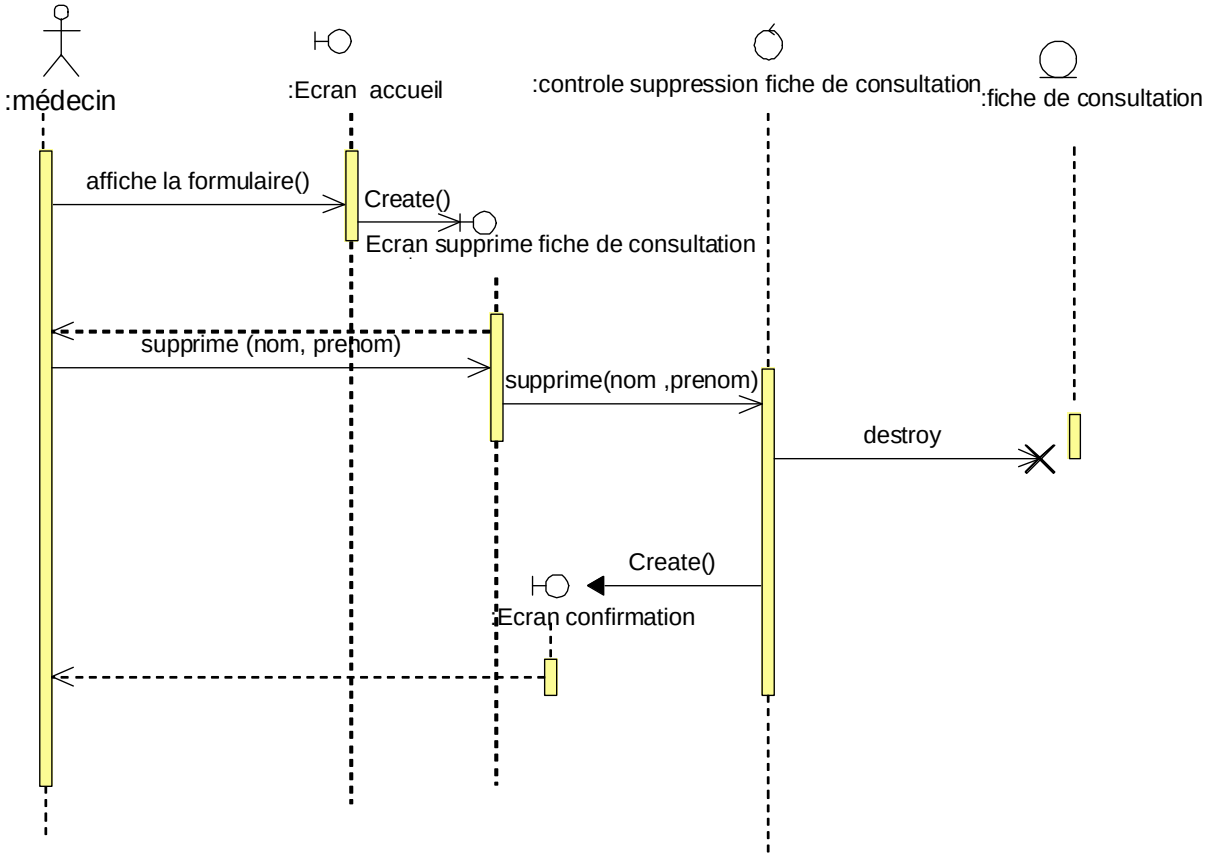


Figure 4.4 diagramme d'interactions du cas d'utilisation <<supprimer fiche consultation >>

2.5 Créer patient

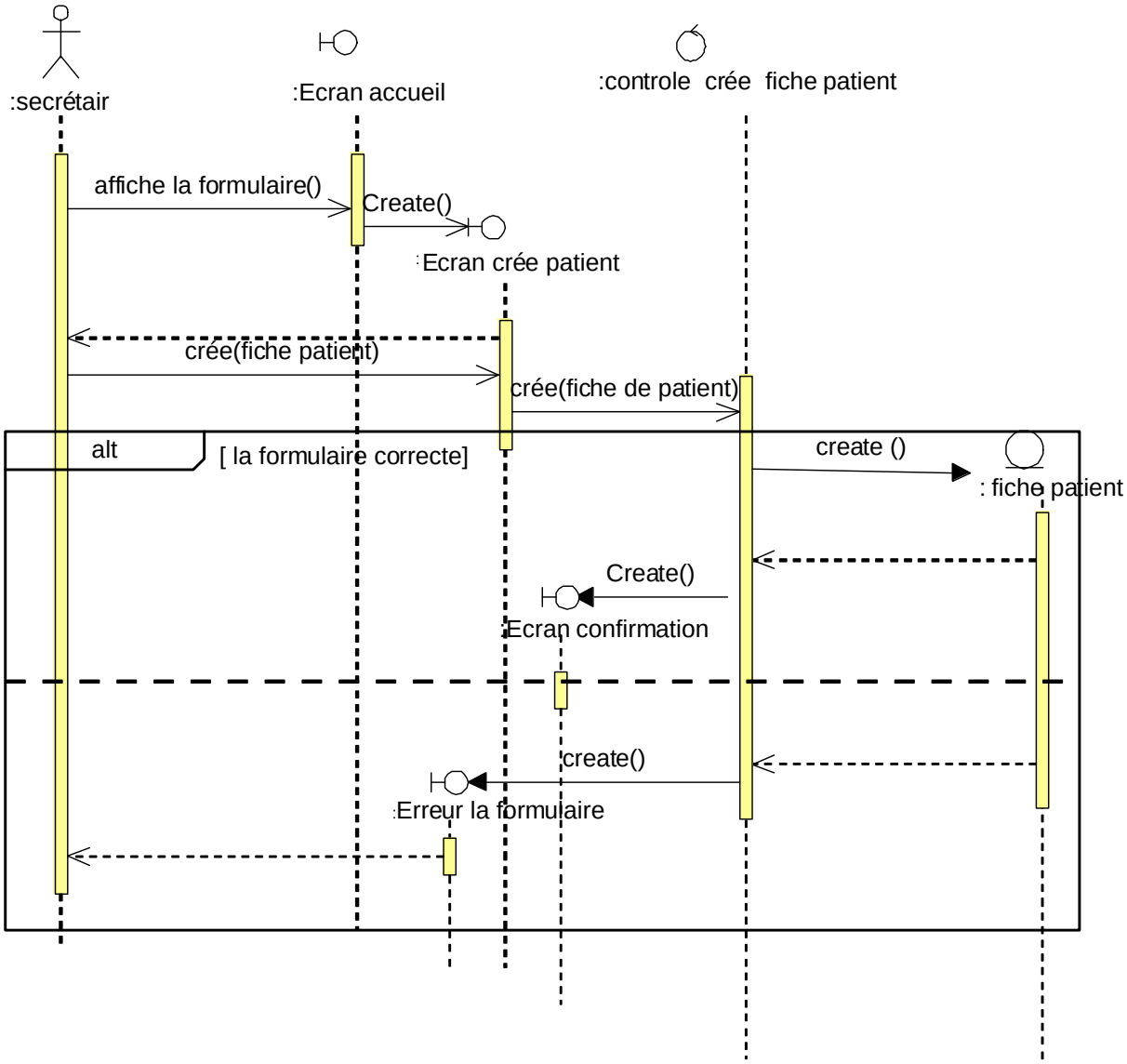


Figure 4.5 diagramme d'interactions de cas d'utilisation <<crée patient >>

2.6 modifier patient

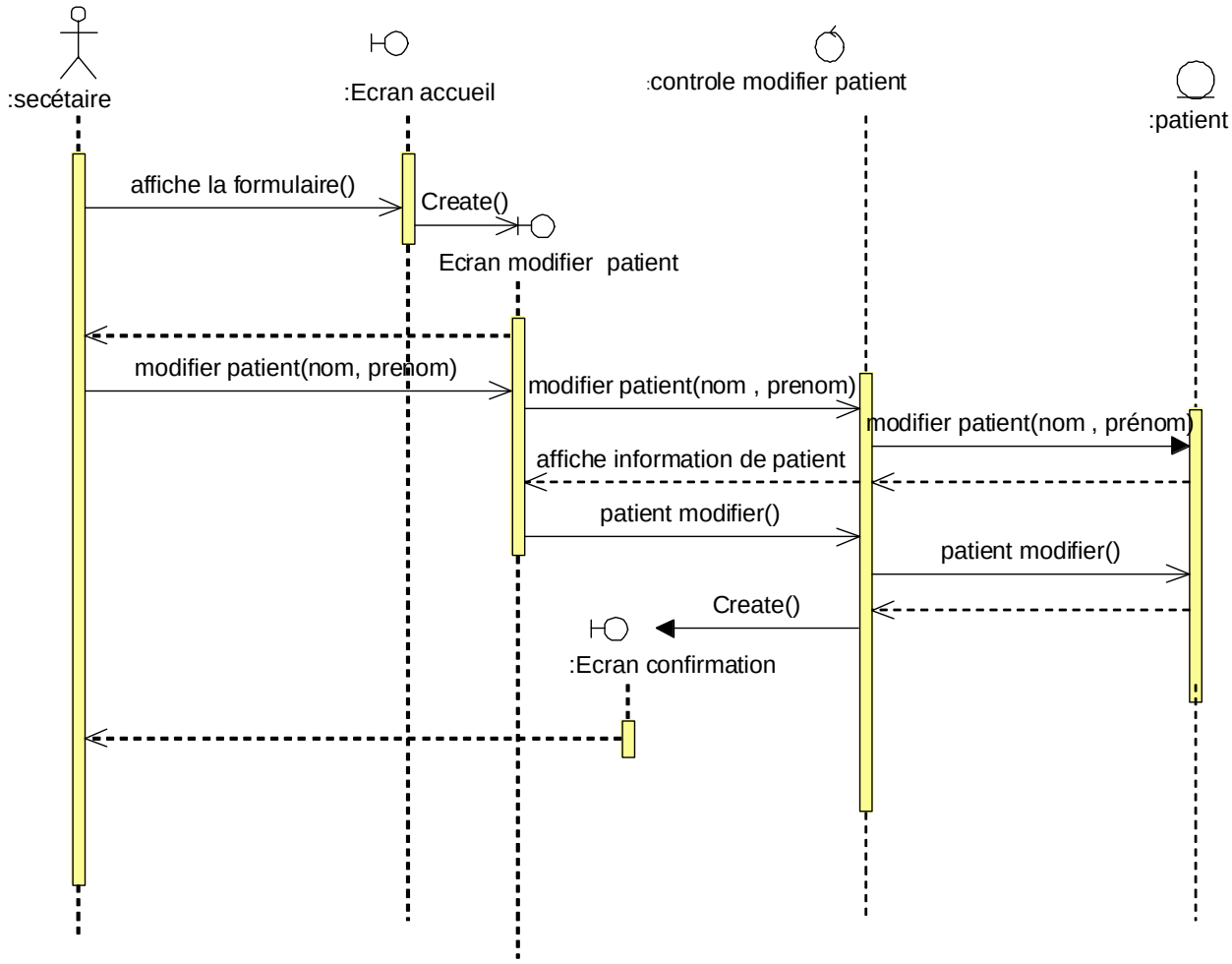


Figure 4.6 diagramme d'interactions du cas d'utilisation <<modifier patient >>

2.7 supprimer patient

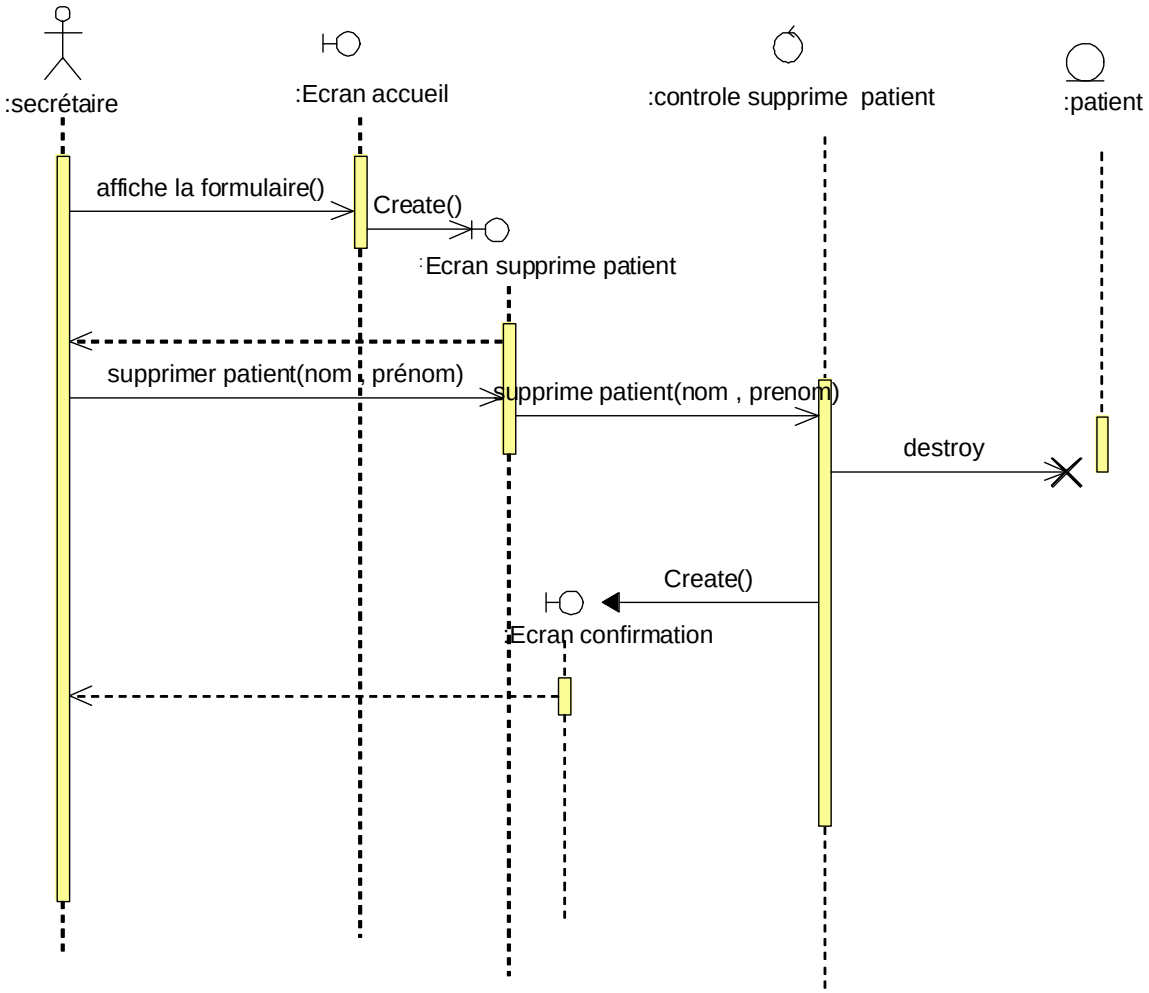


Figure 4.7 diagramme d'interactions du cas d'utilisation <<supprimer patient >>

2.8 établir certificat

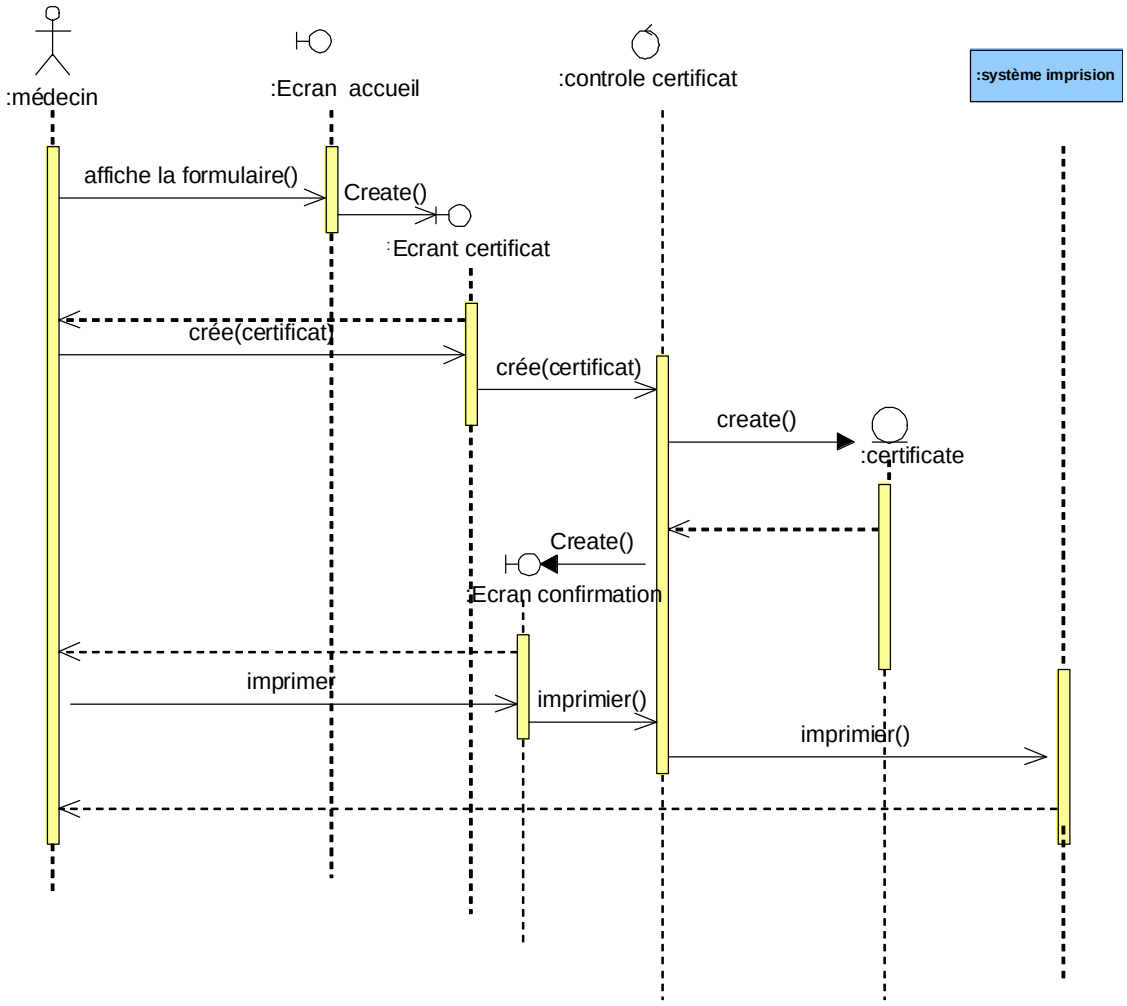


Figure 4.8 diagramme d'interactions du cas d'utilisation <<établir certificat >>

2.9 établir l'analyse

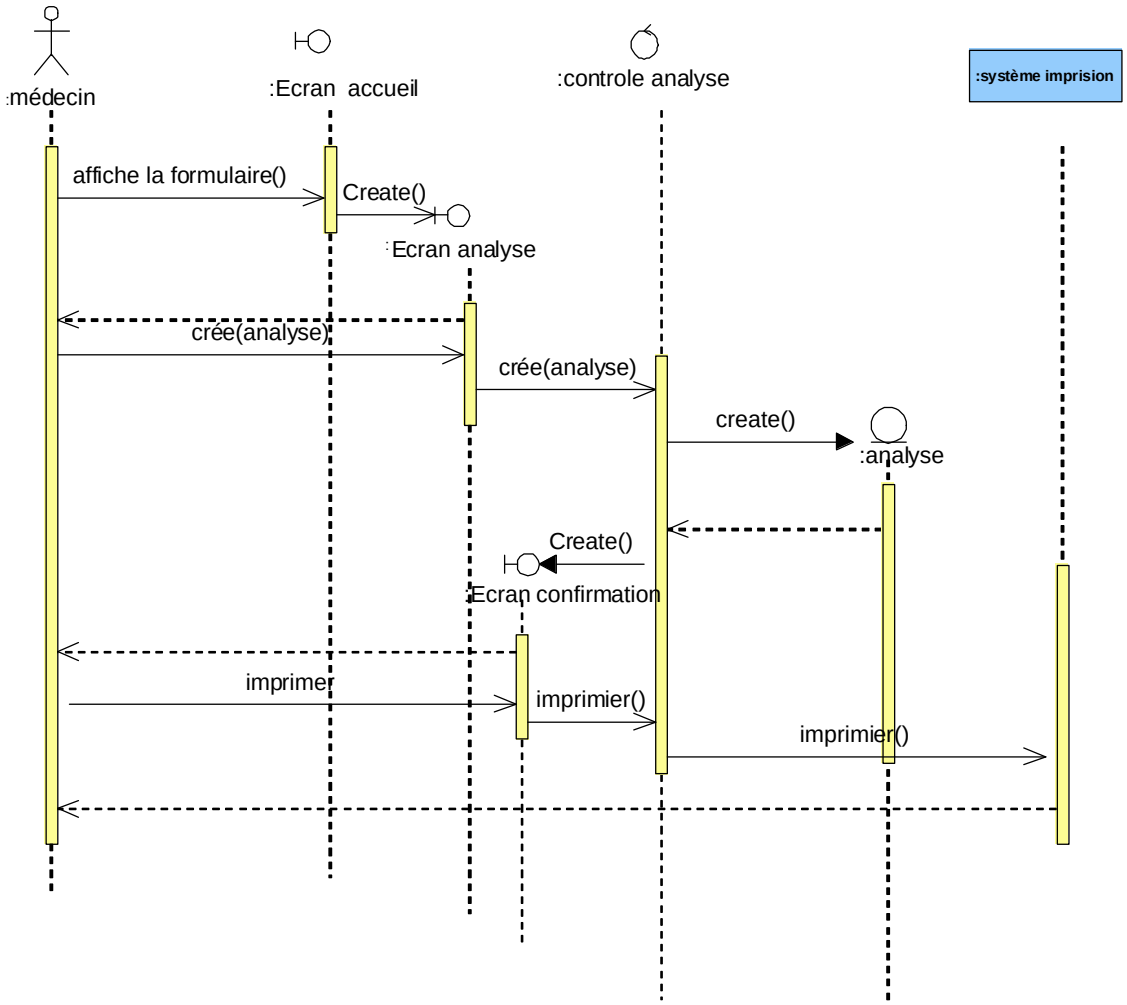


Figure 4.9 diagramme d'interactions du cas d'utilisation <<établir analyse>>

2.10 établir la facture

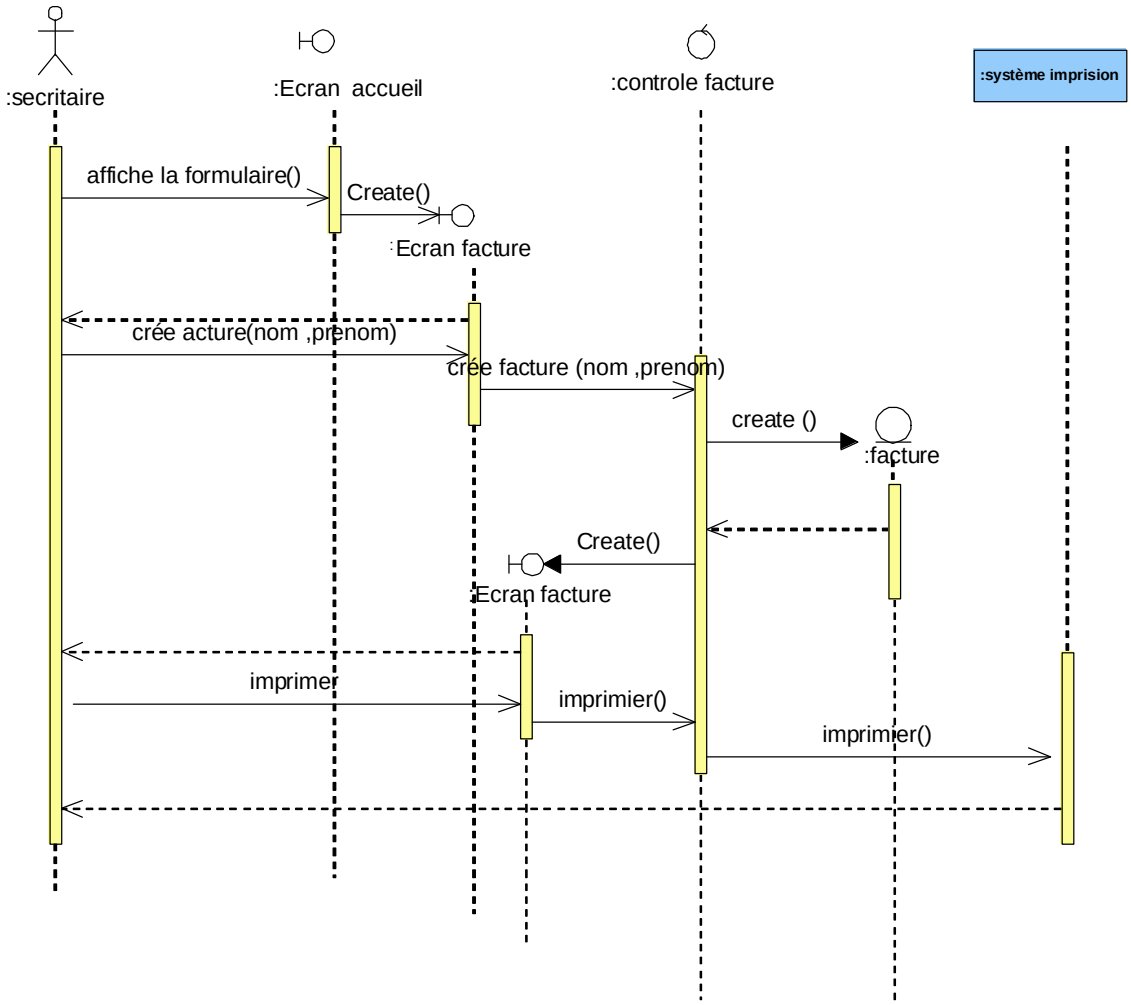


Figure 4.10 diagramme d'interactions du cas d'utilisation <<établir facture>>

2.11 Établir l'ordonnance

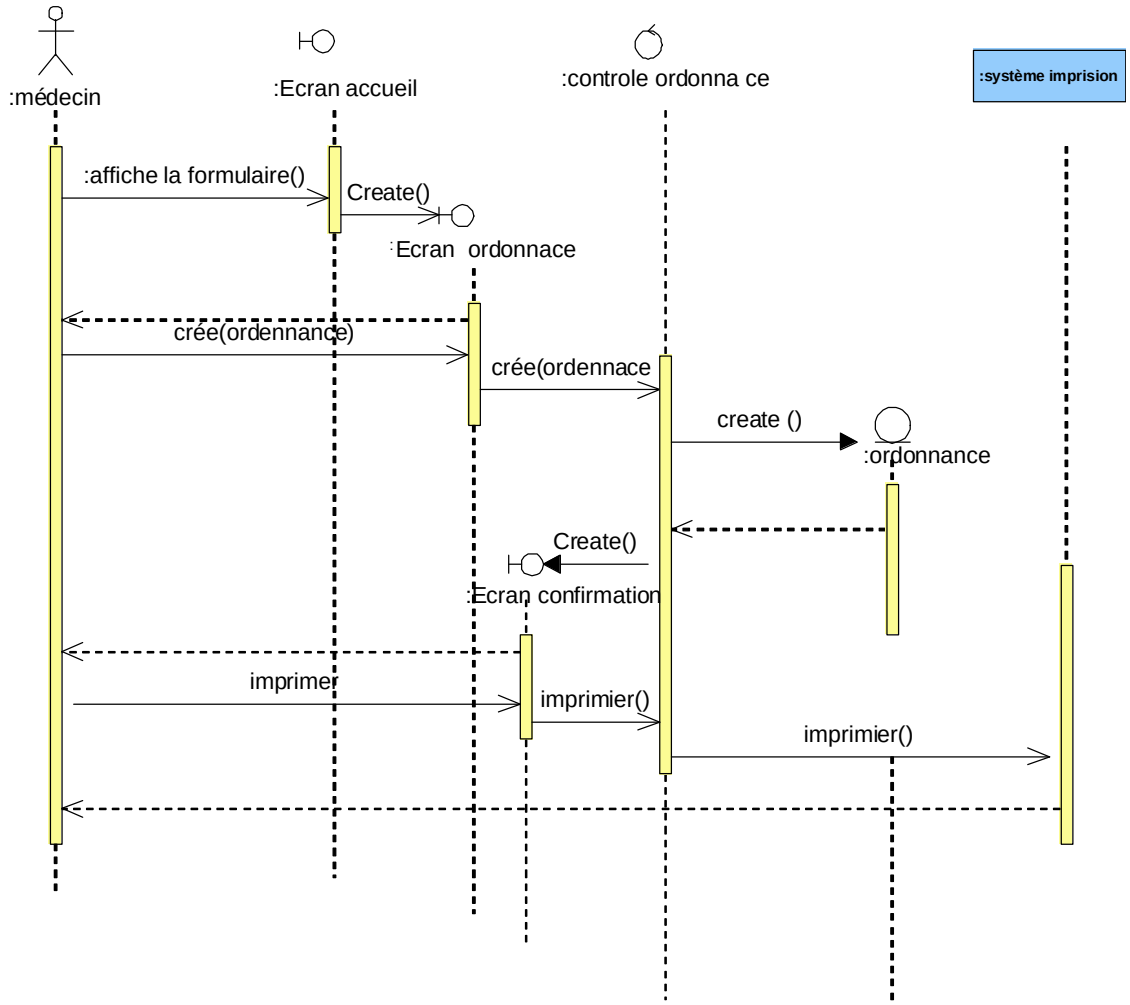


Figure 4.11 diagramme d'interactions du cas d'utilisation <<établir ordonnance>>

2.12 supprimer un RDV

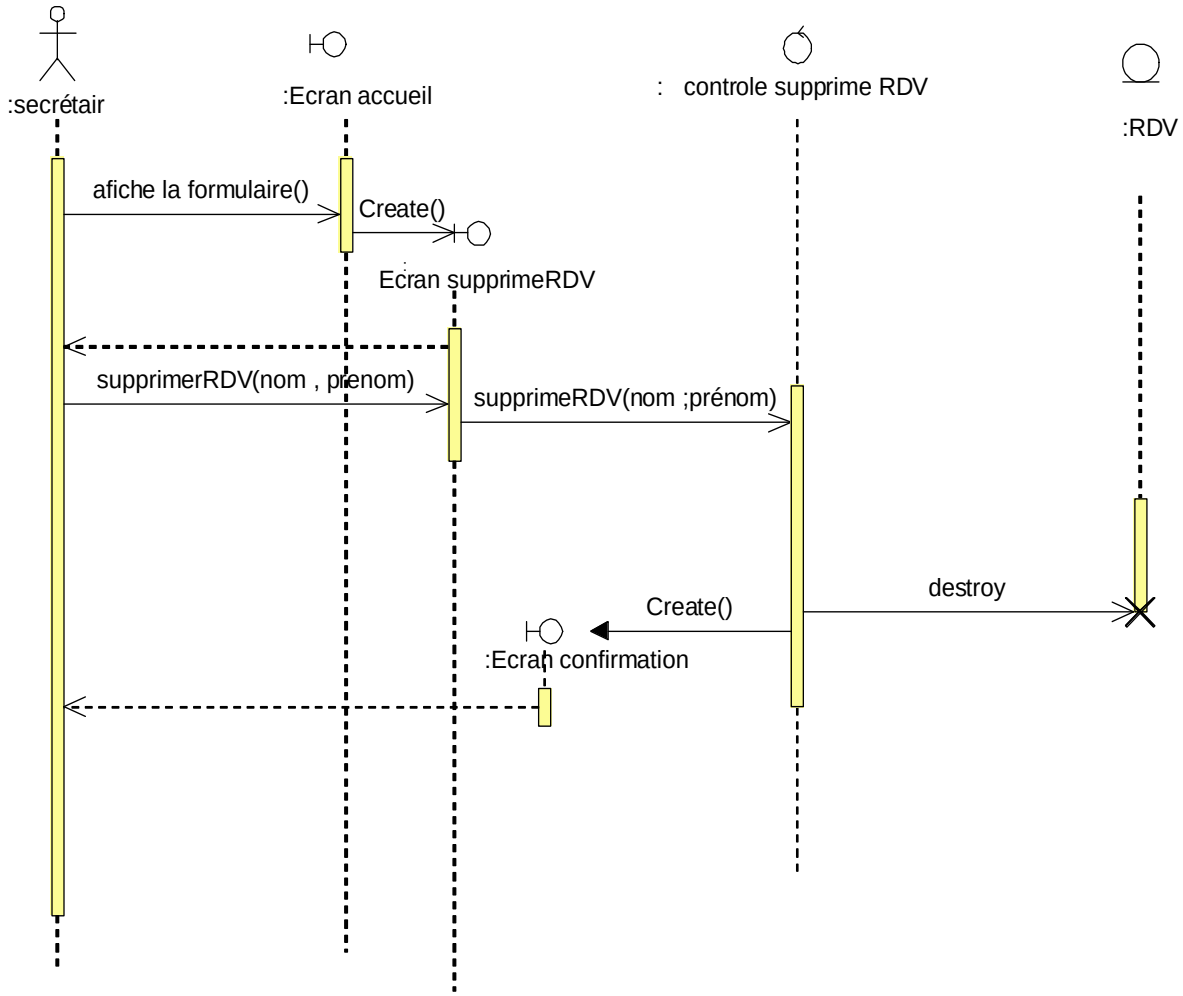


Figure 4.12 diagramme d'interactions du cas d'utilisation <<supprimer RDV>>

2.13 Modifier un RDV

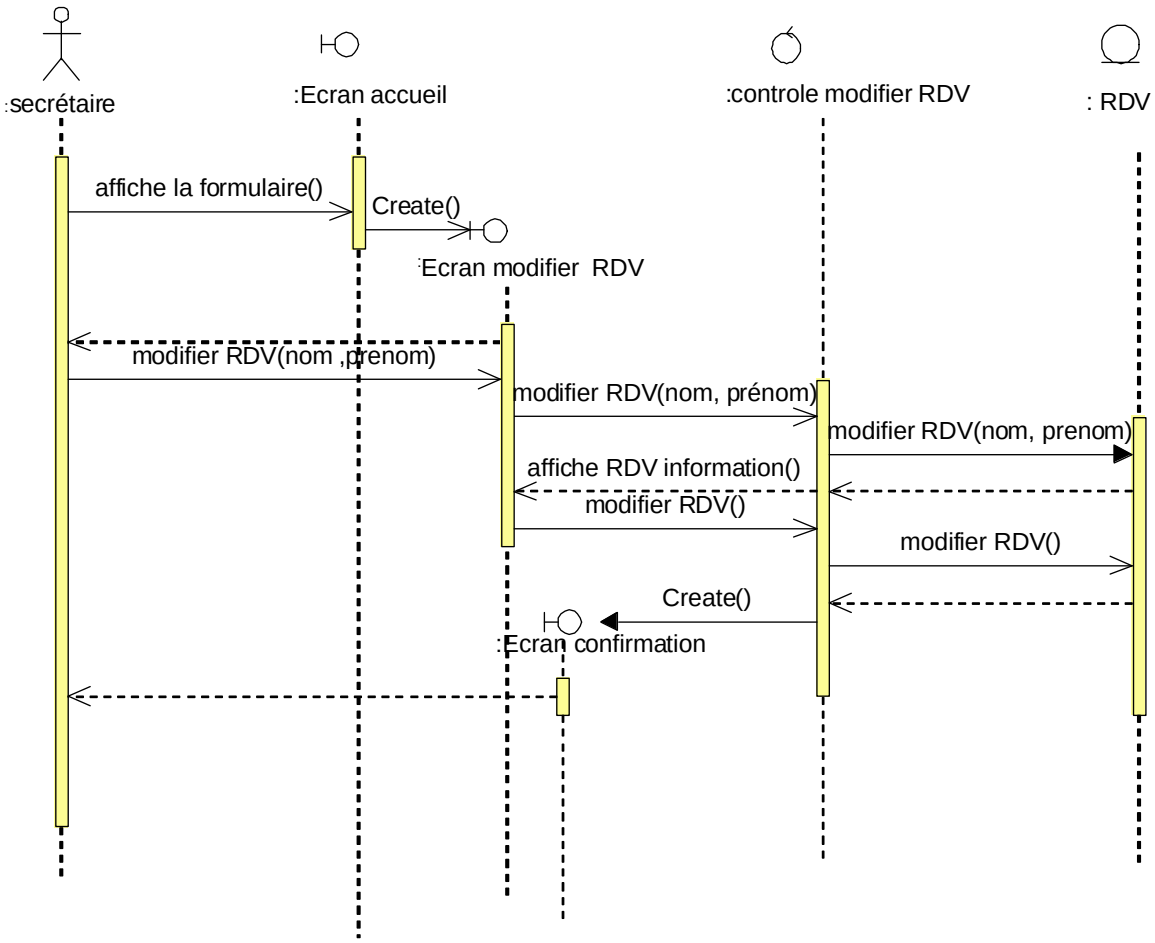


Figure 4.13 diagramme d'interactions du cas d'utilisation <<modifier RDV>>

2.14 ajouter un RDV

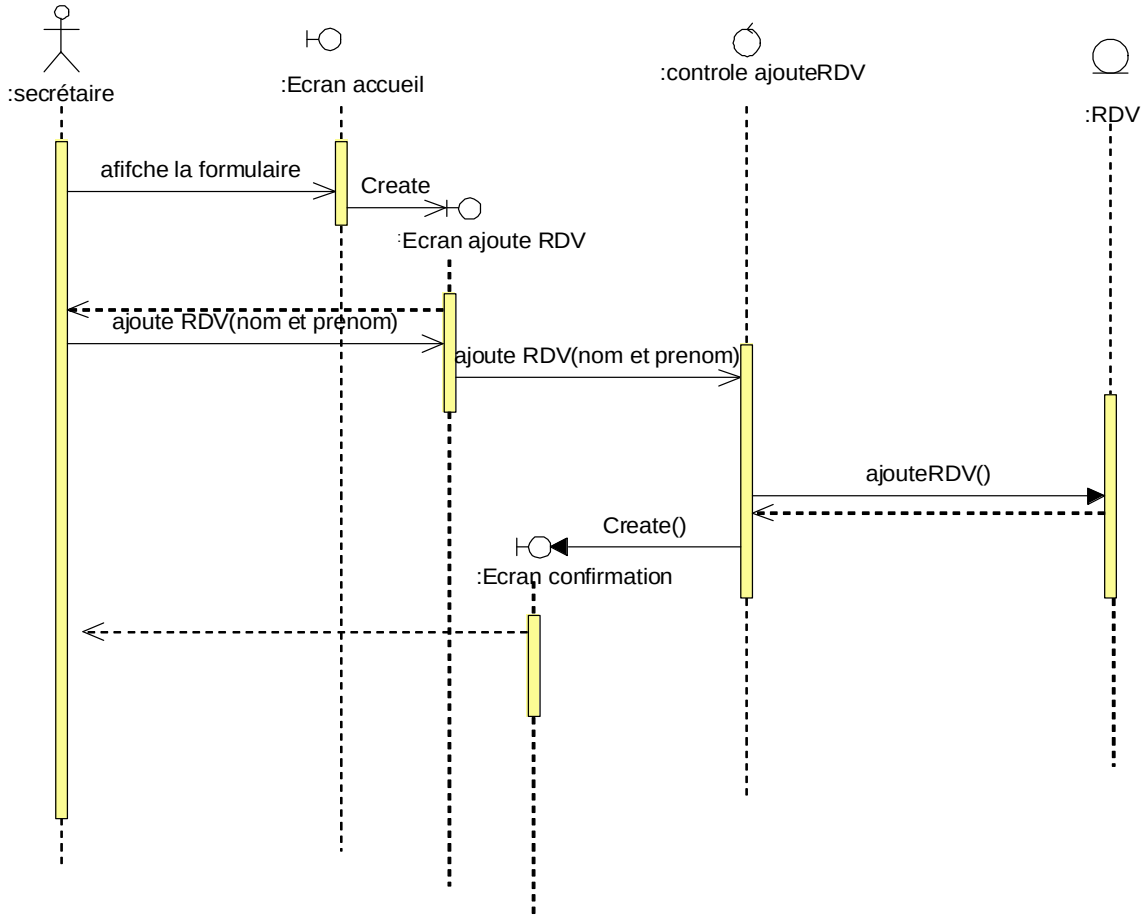


Figure 4.14 diagramme d'interactions du cas d'utilisation <<ajoute RDV>>

2.15 RDV en cours

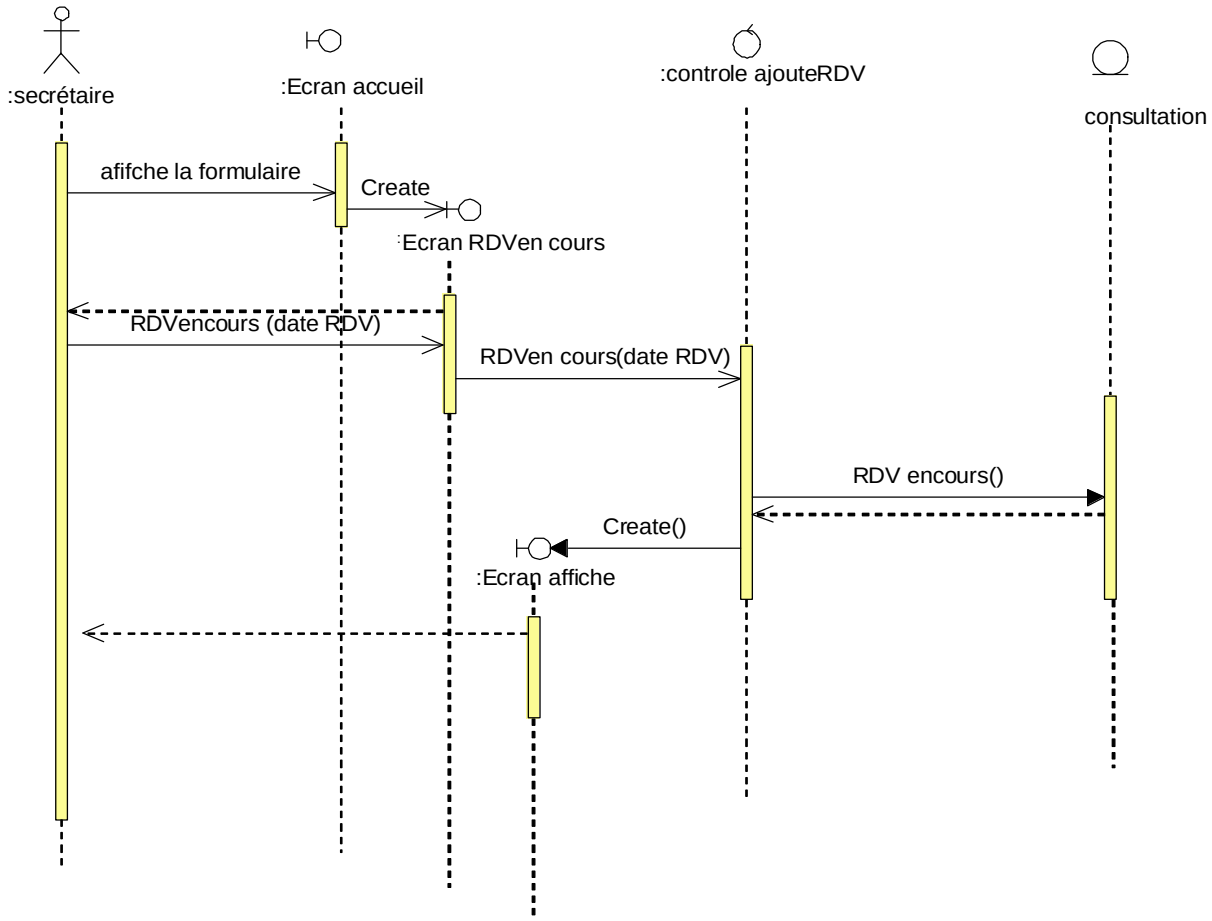


Figure 4.15 diagramme d'interactions du cas d'utilisation << RDV en cours>>

3. Les diagrammes de classes participants des cas d'utilisations

3.1 S'authentifier

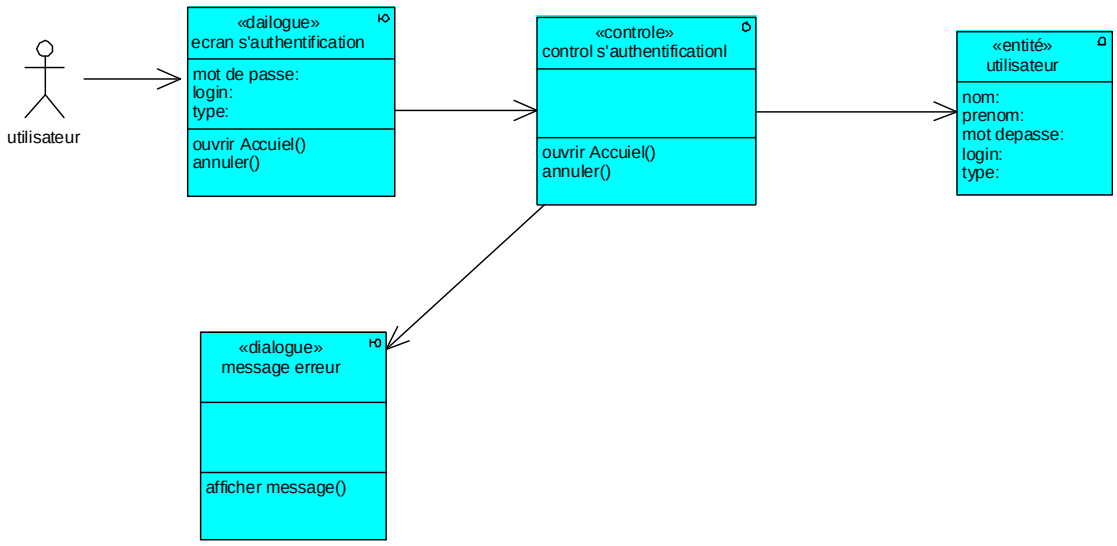


Figure 4.15 Diagramme de classe participation du cas d'utilisation <<s'authentification>>

3.2 Créer fiche de consultation

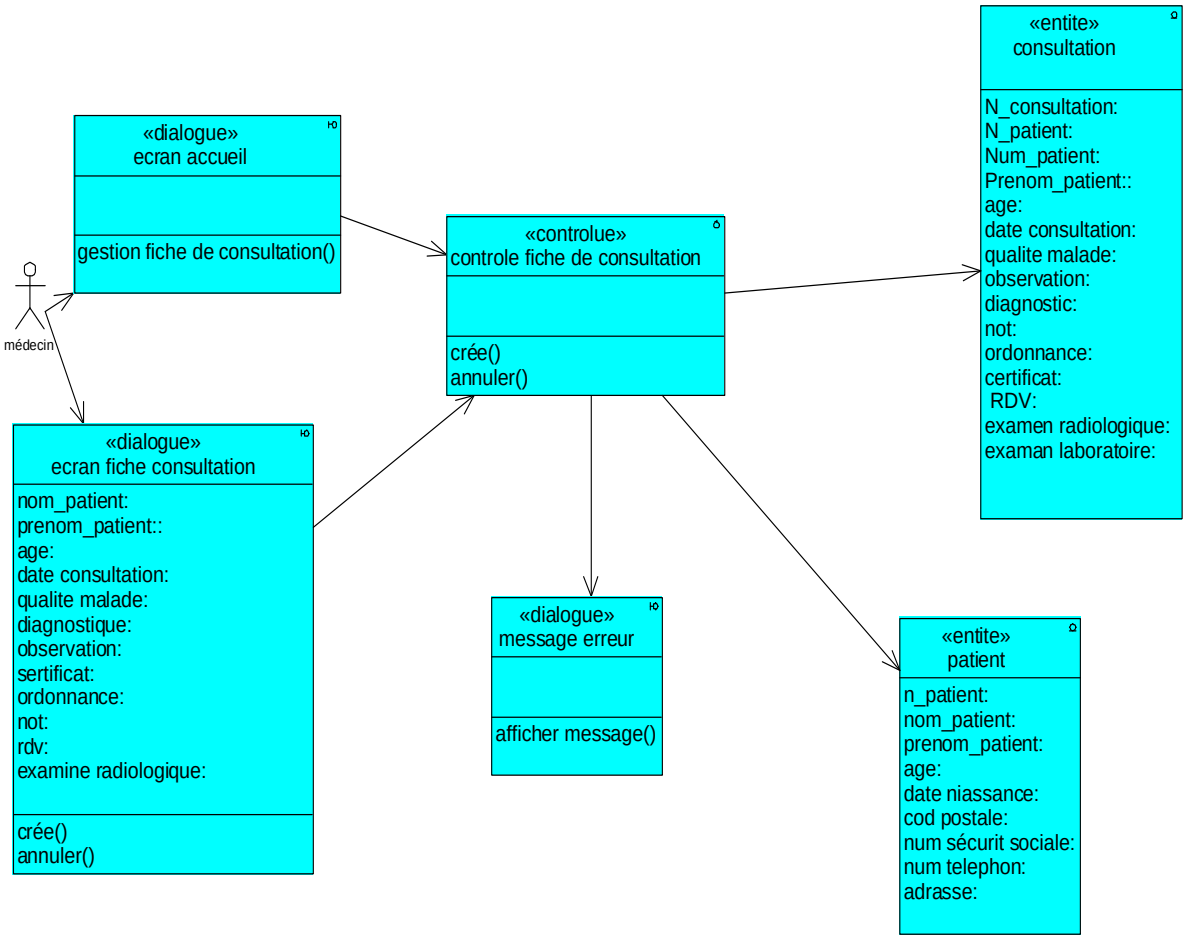


Figure 4.16 Diagramme de classe participation du cas d'utilisation<<crée fiche consultation>>

3.3 Supprimer fiche de consultation

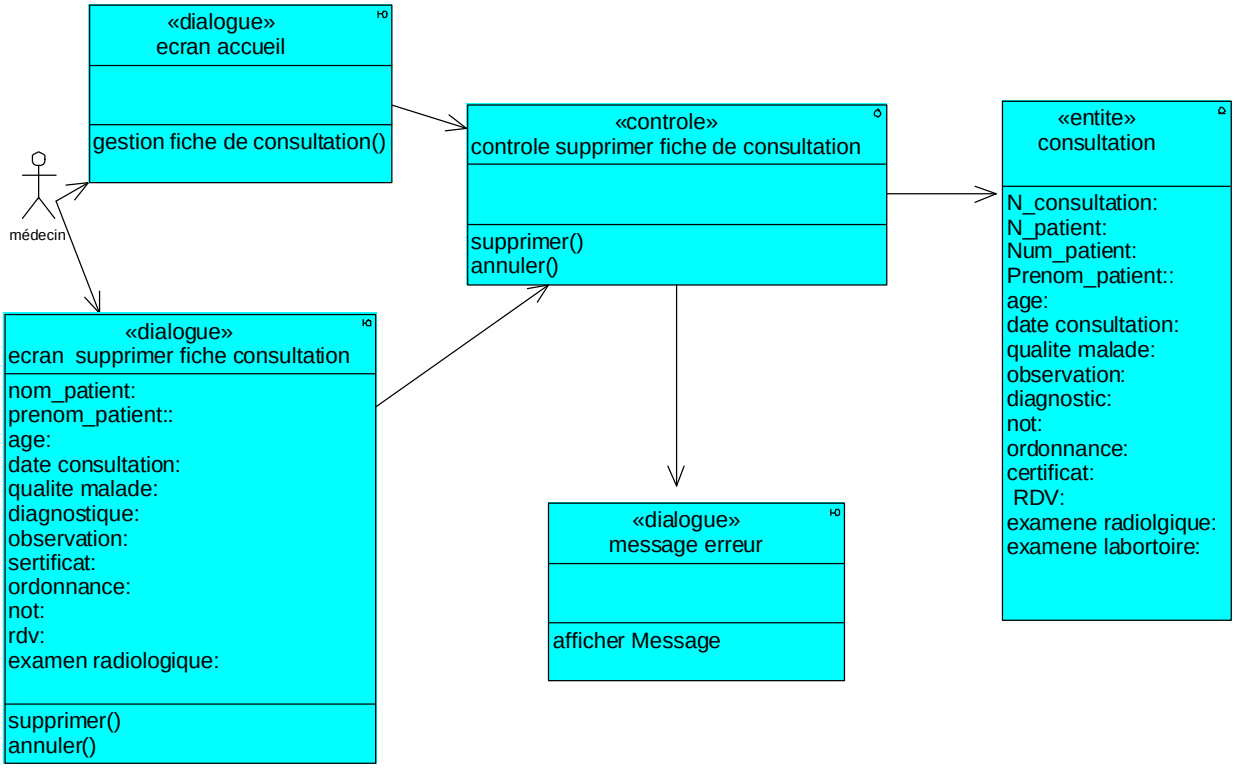


Figure 4.17 Diagramme de classe participation du cas d'utilisation<<supprimer fiche consultation>>

3.4 Modifier fiche de consultation

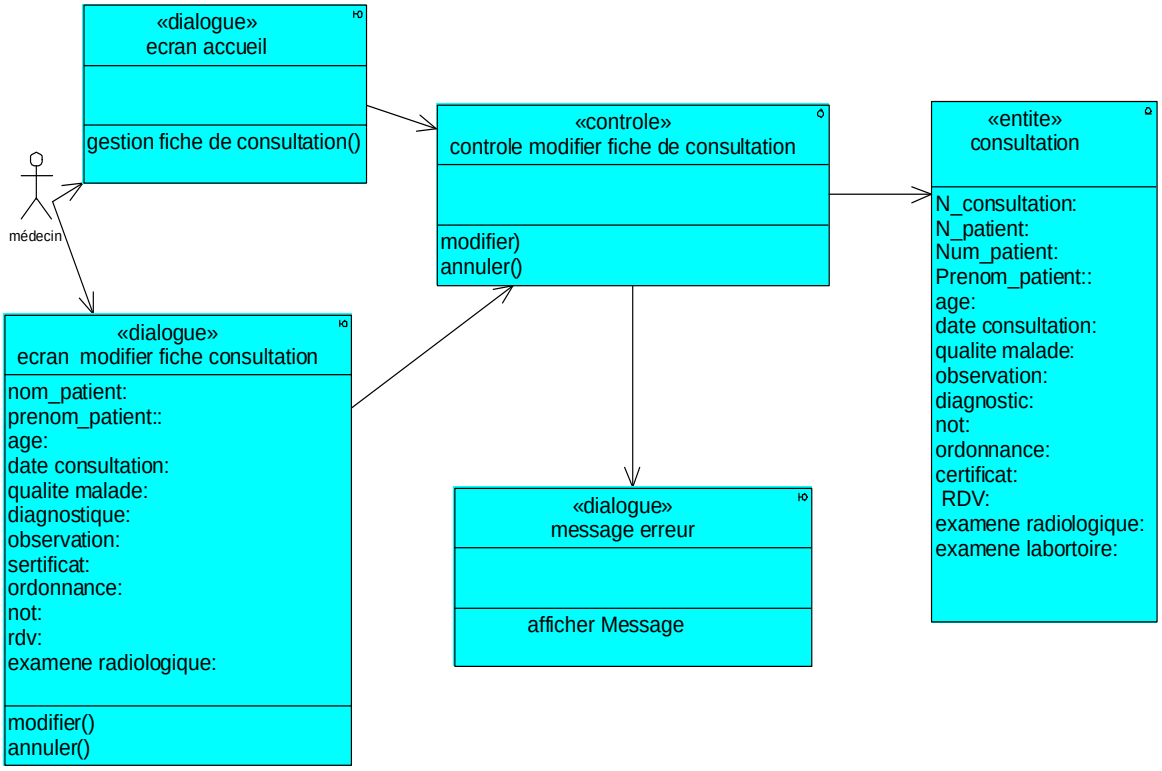


Figure 4.18 Diagramme de classe participation du cas d'utilisation<<modifier fiche consultation>>

3.5 Créer fiche du patient

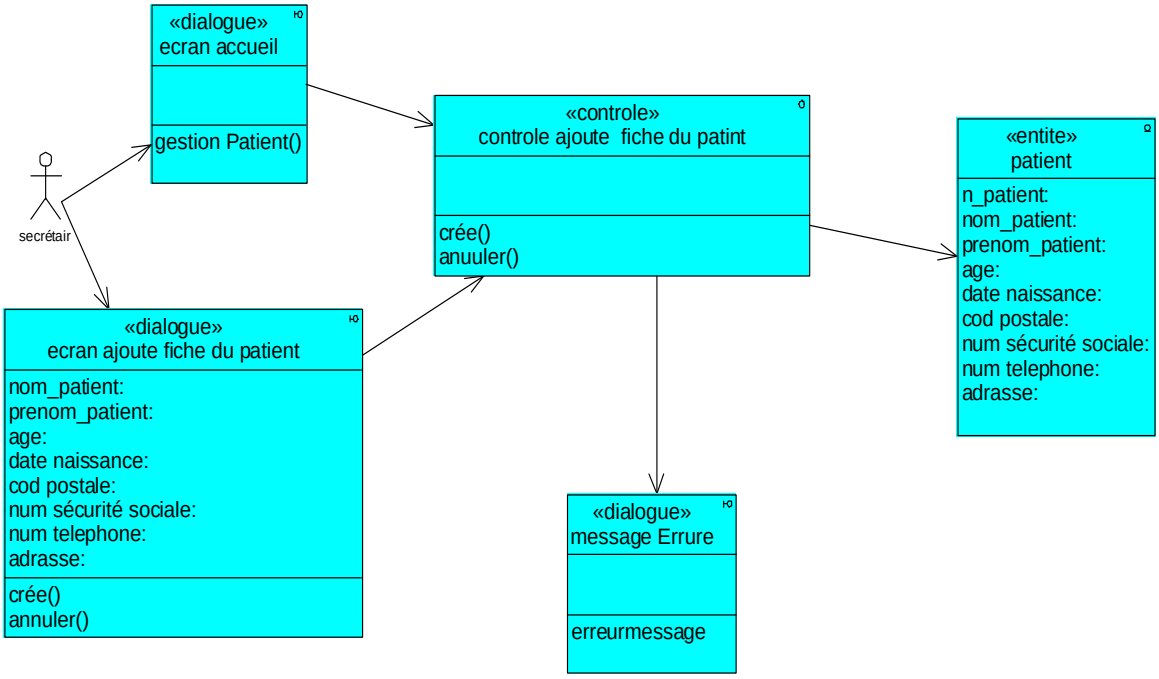


Figure 4.19 Diagramme de classe participation du cas d'utilisation<<crée patient >>

3.6 modifier fiche de patient

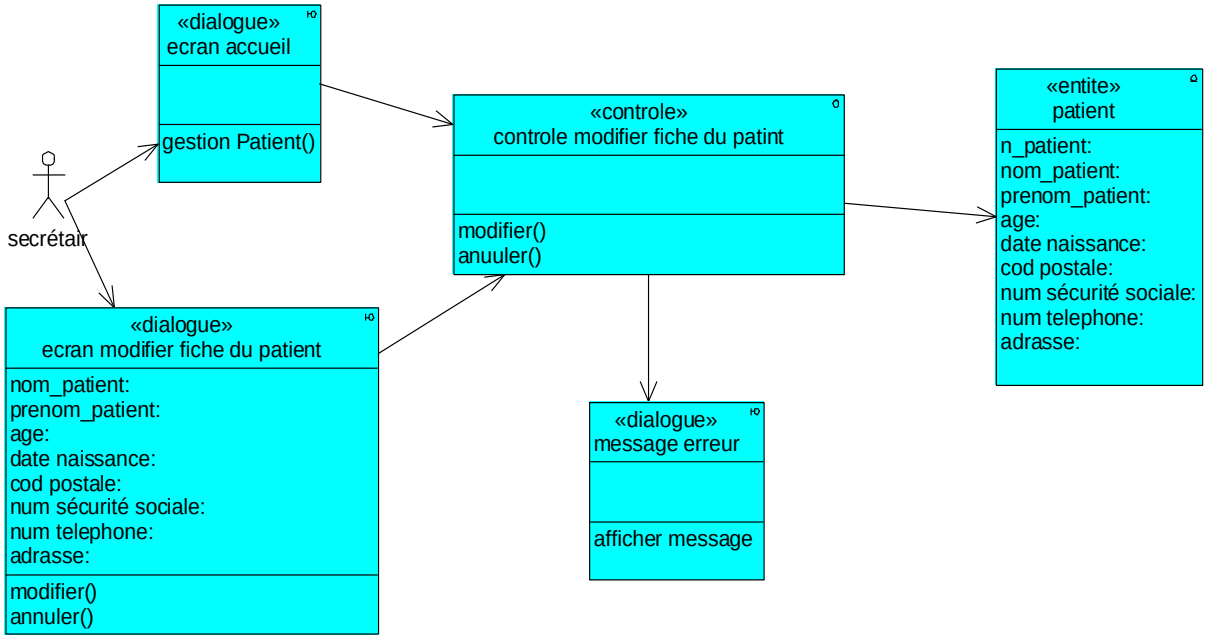


Figure 4.20 Diagramme de classe participation du cas d'utilisation<<modifier patient >>

3.7 supprimer fiche patient

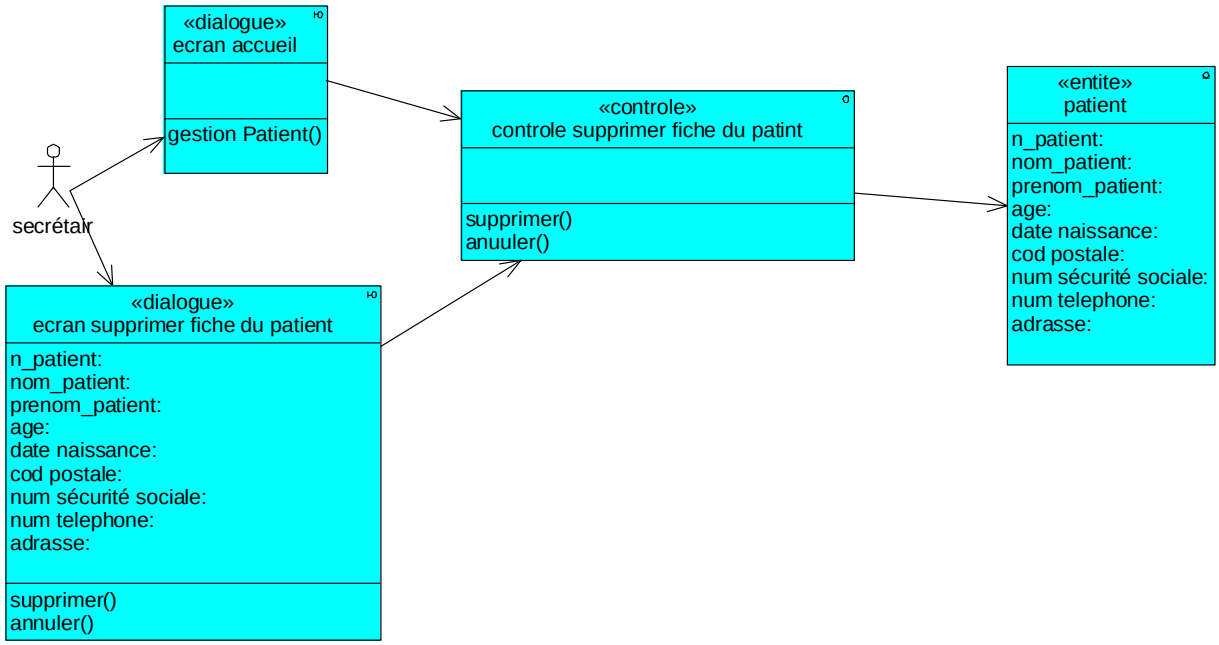


Figure 4.21 Diagramme de classe participation du cas d'utilisation<<supprimer patient >>

3.8 Établir facture

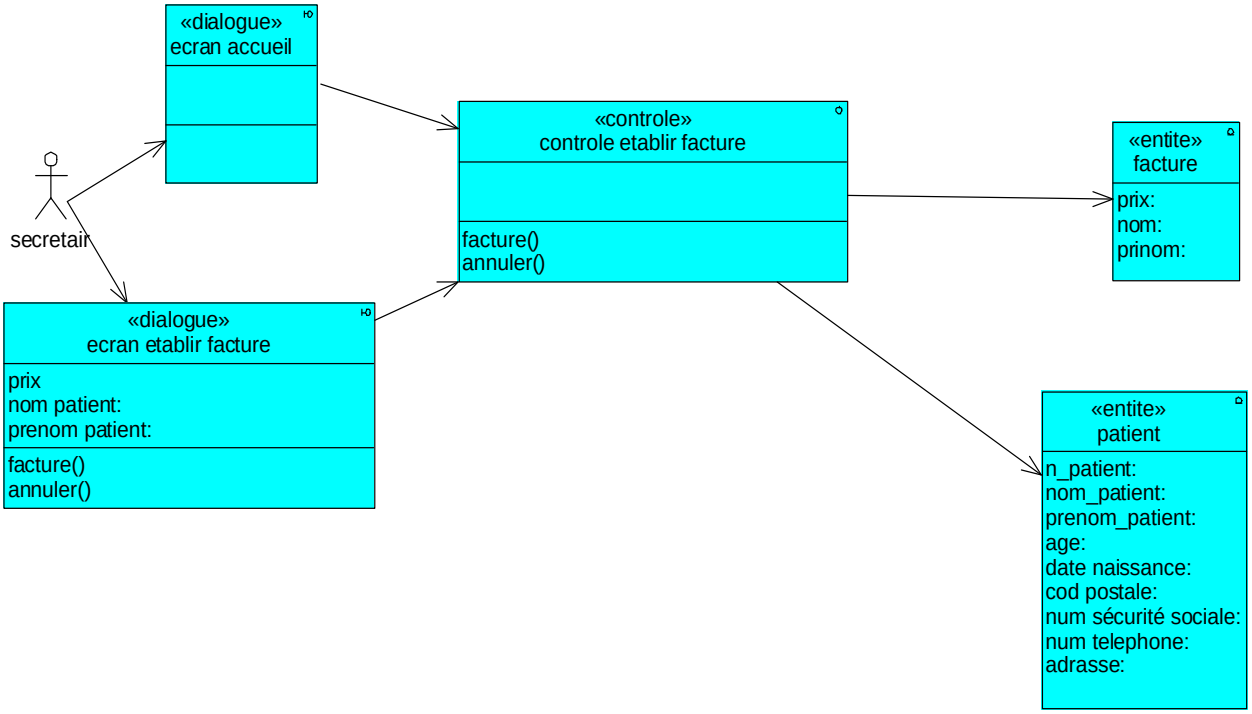


Figure 4.22 Diagramme de classe participation du cas d'utilisation<<établir facture >>

3.9 Établir l'analyse

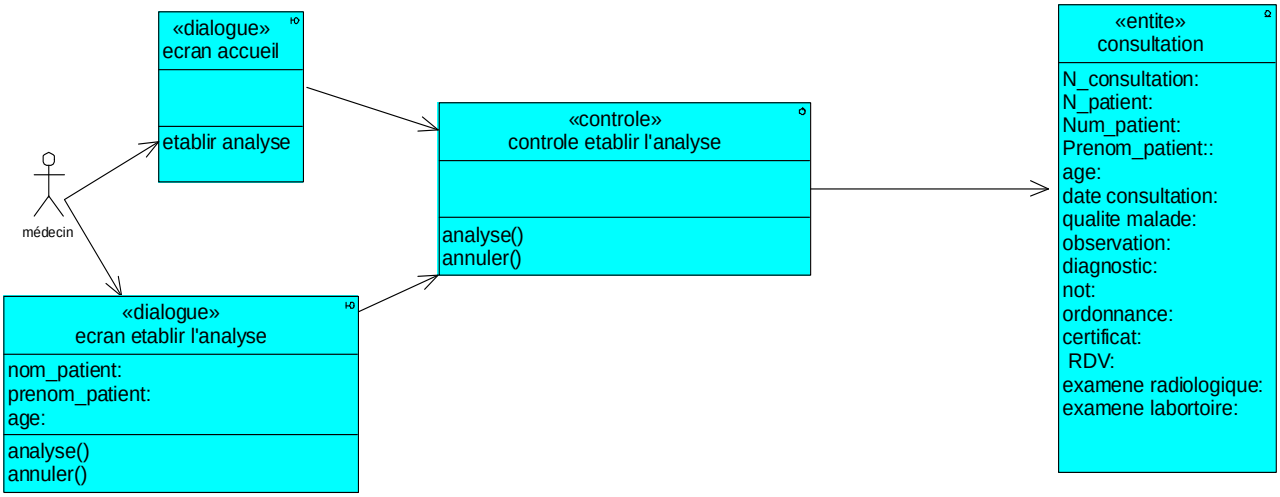


Figure 4.23 Diagramme de classe participation du cas d'utilisation<<établir l'analyse>>

3.10 Établir ordonnance

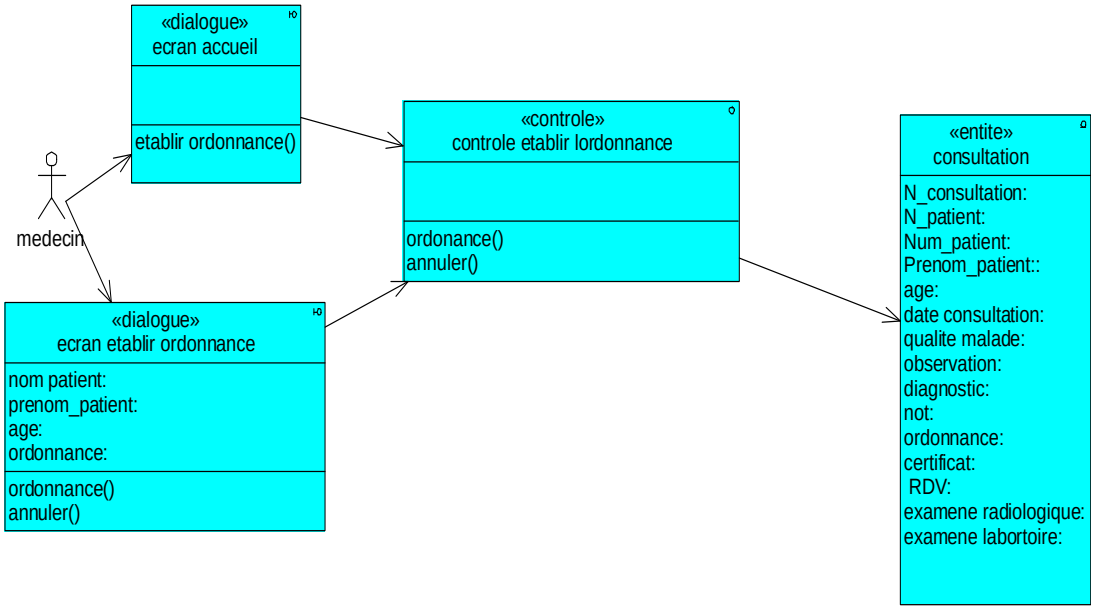


Figure 4.23 Diagramme de classe participation du cas d'utilisation<<établir ordonnance>>

3.11 Ajouter RDV

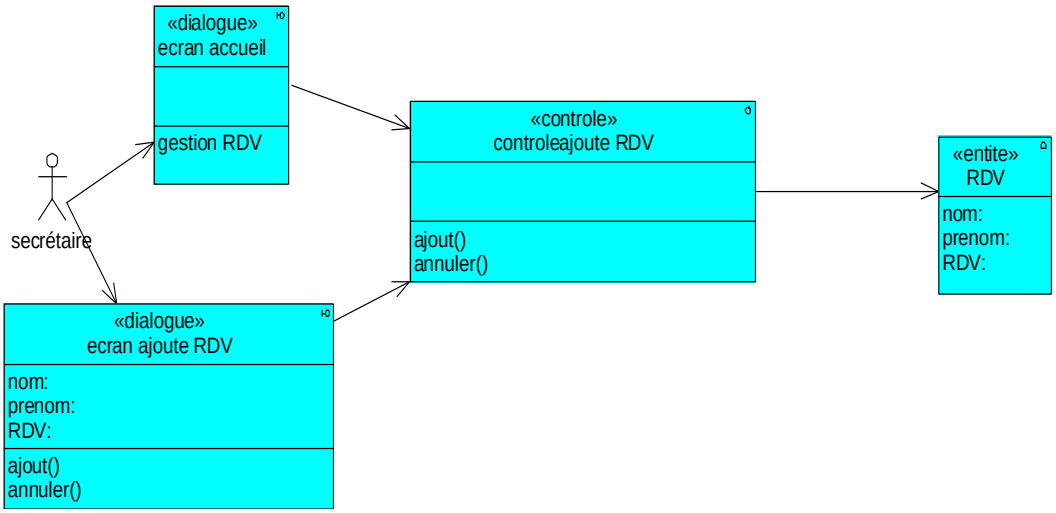


Figure 4. 24 Diagramme de classe participation du cas d'utilisation<<ajout RDV>>

3.12 Supprimer RDV

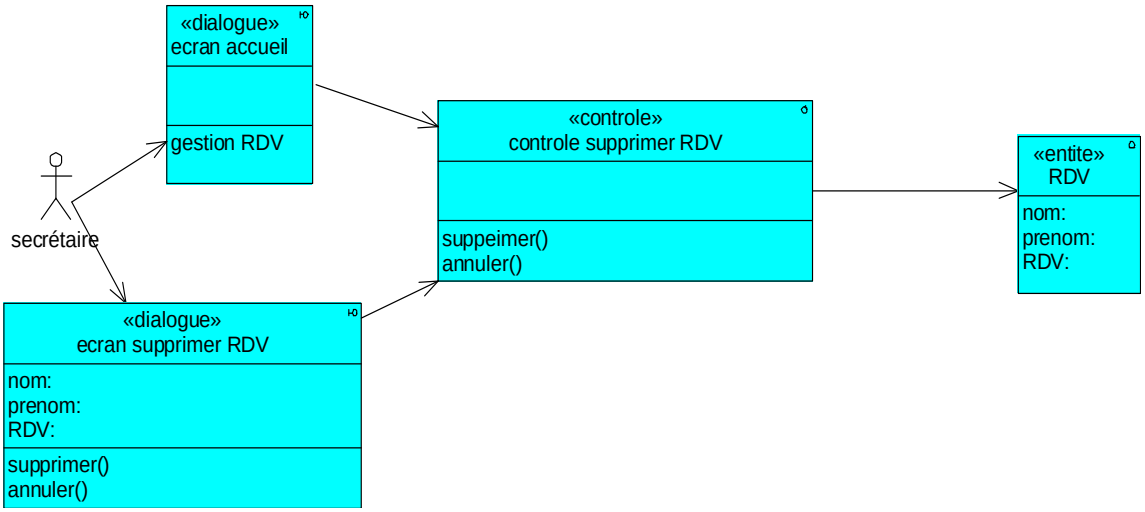


Figure 4. 25 Diagramme de classe participation du cas d'utilisation<<supprimer RDV>>

3.13 Modifier RDV

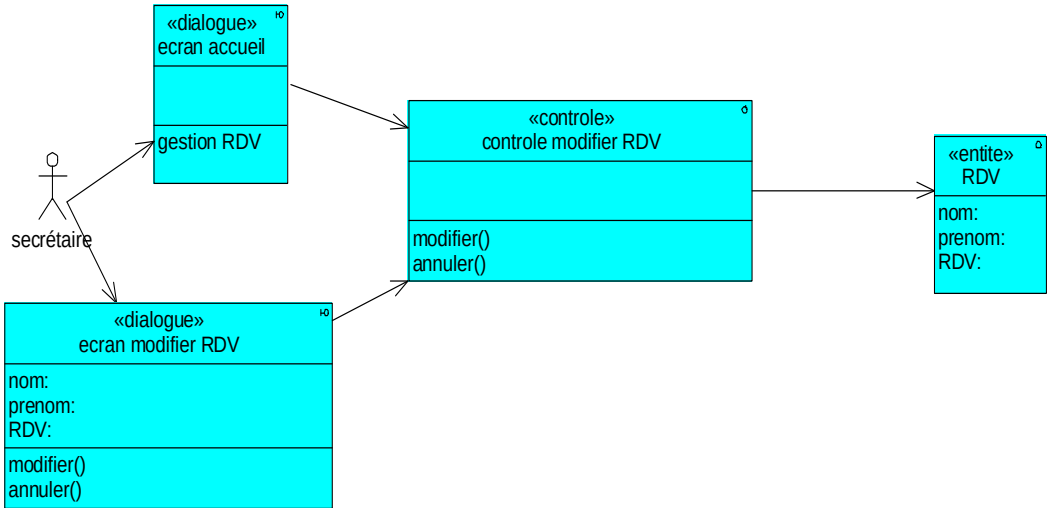


Figure 4. 26 Diagramme de classe participation du cas d'utilisation<< RDV en cours>>

3.14 RDV en cours

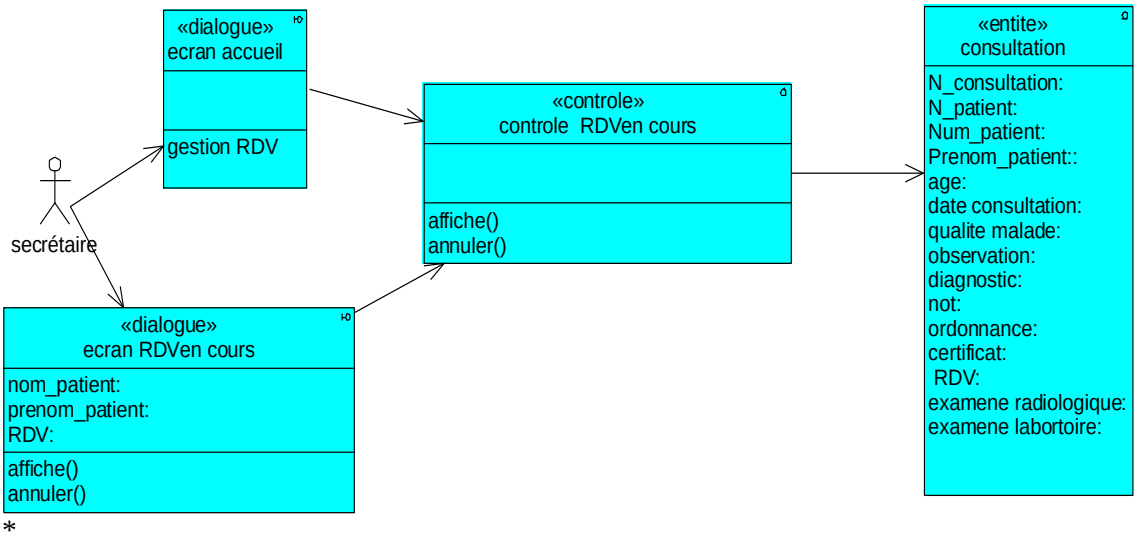


Figure 4. 27 Diagramme de classe participation du cas d'utilisation<< RDV en cours>>

4 .Conclusion :

Dans ce chapitre, nous avons développé les diagrammes d'interactions et les diagrammes de classes participantes des cas d'utilisations qui représente la phase d'analyse de notre développement du logiciel.

Le chapitre suivant présentera la phase d'implémentation, les outils utilisés et donne une première idée des différentes fenêtres du logiciel.

CHAPITRE5

IMPLÉMENTATION

Introduction

Lors de la phase d'implémentation, nous allons utiliser les résultats de la phase de Conception (le diagramme de classes de conception, et les diagrammes d'interactions) pour implémenter notre application dans un langage de programmation. Nous allons également préparer les données nécessaires à l'exploitation du logiciel, en convertissant les classes entités en une base de données.

2 .Implémentation

1.1. Environnement de développement de l'application

Dans le développement de notre application, nous avons utilisé :

2.1.1 Pacestar UML Diagrammer 6.02 :

Un programme qui fournit un ensemble complet d'outils de modélisation Graphique, d'analyse et de conception dans le développement de logiciels basés sur Les modèles UML.

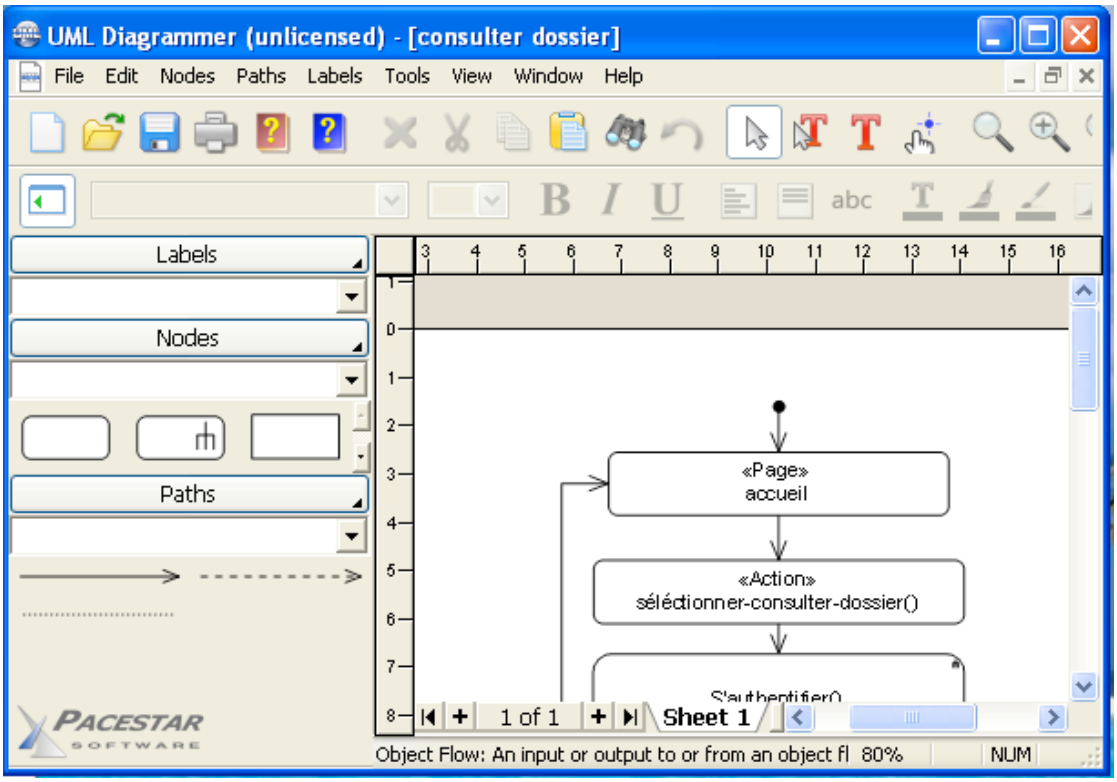


Figure 5.1 pacestar UMLdiagrammer 6.02

2.1.2 Delphi

Pour réaliser notre application, nous avons utilisé l'environnement Delphi. Delphi est un outil de développement puissant et rapide pour la programmation d'applications pour Windows.

Delphi est un langage très facilement maîtrisable permettant de créer de petites applications, ceci en cachant au développeur tout les appels système.

Cependant, s'il le désir, il peut accéder au plus profond de la programmation Windows (appels API, ...) La conception de l'interface graphique est conçue visuellement, un peu à la manière des jeux de construction.

On dispose pour faire celle-ci d'une palette d'objets (les VCL) que l'on peut placer sur la fenêtre de développement (ils contiennent les barres de menus déroulants, les boutons, les boîtes de dialogue standard Windows, ...). Ces objets sont les pièces de ce jeu de construction.

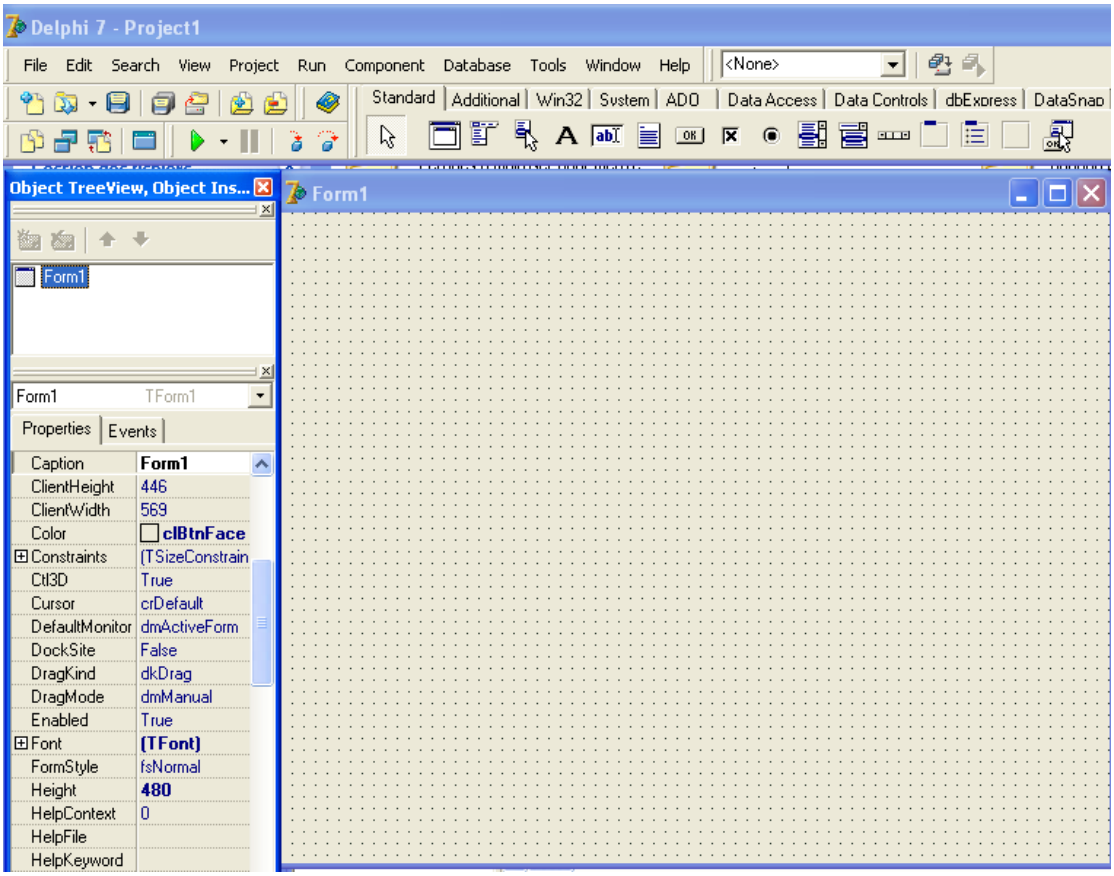


Figure 5.2 interface Delphi

Afin d'avoir des interfaces ergonomiques, nous avons utilisé Adobe Photoshop pour les traitements des images de notre application.

2.2 Les langages de programmation :

2.2.1 Pascal Un moyen pour créer un logiciel est d'utiliser un langage informatique, traduit ensuite par l'ordinateur pour en faire un logiciel.

Le langage que nous avons utilisé pour développer notre application est le langage Pascal, que Delphi utilise pour écrire des logiciels.

2.2.2 SQL server

Le langage SQL est normalisé et constitue le standard d'accès aux BDD relationnelles

Description de l'application

Un moyen pour créer un logiciel est d'utiliser un langage informatique, traduit ensuite par l'ordinateur pour en faire un logiciel.

Le langage que nous avons utilisé pour développer notre application est le langage Pascal, que Delphi utilise pour écrire des logiciels.

L'application que nous avons développée possède toutes les fonctionnalités décrites dans le cahier des charges, à savoir :

- L'authentification
- Gestion fiche consultation (crée, supprimer, modifier)
- Gestion patient (crée, supprimer, modifier)
- Gestion RDV

2. Conception de la base de données

Il existe d'autres solutions de transformation mais ces règles sont les plus simples et les plus opérationnelles :

Transformation des entités /classes: la règle est simple**R1* :** Chaque classe devient une relation, l'identifiant de l'entité devient clé primaire pour la relation.

* chaque classe du diagramme UML devient une relation. Il faut choisir un attribut de la classe pouvant jouer le rôle identifiant.

Transformation des associations : les règles de transformation que nous allons voir Dépendent des cardinalités /multiplicités maximale des associations. Nous distinguons trois familles d'association.

Association 1..* : la règle est la suivante :

R2 : il faut ajouter un attribut de type clé étrangère dans la relation fils de l'association.

L'attribut porte le nom de la clé primaire de la relation père de l'association.

Association *..*: la règle est la suivante :

R3 : association/classe – association devient une relation. La clé primaire de cette relation est la concaténation des identifiants des identités connecté à l'association. Chaque attribut devient clé étrangère si entité/classe connectée dont il devient une relation en vertu de la règle R1.

Les attributs d'association/classe – association doivent être ajoutés a la nouvelle relation.ces attribut ne sont ni clé primaire, ni clé étrangère.

Association 1..1: la règle est la suivante :

R4 : il faut ajouter un attribut de type clé étrangère dans la relation dérivée de l'entité ayant la cardinalité minimale égale à zéro. Dans le cas de diagramme UML il faut ajouter un attribut de type clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un.

L'attribut porte le nom de la clé primaire de la relation dérivée d'entité classe connectée à l'association. Si les deux cardinalités minimales égales à zéro, le choix est donné entre les deux relations dérivées de la R1.

Si les deux cardinalités minimales égales à un, il est préférable de fusionner les deux entités/classe en une seule [2].

En se basant sur les règles ci-dessus, nous avons converti les classes entités et leurs Associations, à des tables dans la base données. Les tables générées sont :

Utilisateur (nom_ utilisateur, login mot de passe, type)

Patient (n_ patient, nom _patient, prénom _patient, date naissance, ardasse, Num téléphone Num sécurité sociale, code postale)

RDV (nom, prénom, date RDV)

Facture (nom, prénom, prix)

Consultation (n_ consultation, nom patient, prénom patient, date consultation, diagnostic, observation, certificat, ordonnance, note, examine radiologue, examine Laboratoire)

Ordonnance (nom, prénom, Age, ordonnance)

Analyse (nom, prénom, Age)

Certificat (nom, prenom, age,certificat)

3. Les interfaces de l'application

3.1 Fenêtre d'authentification

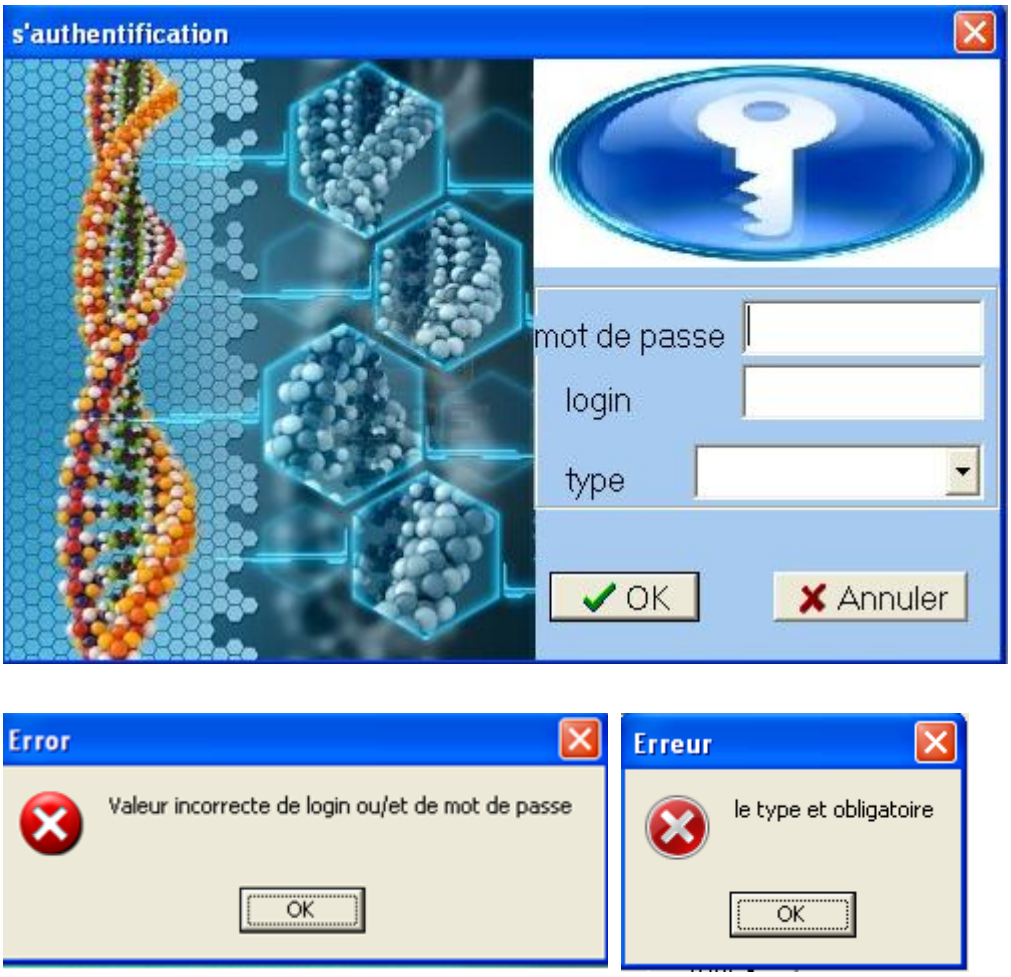


Figure 5.3 s'authentification

3.2 Fenêtre d'accueil

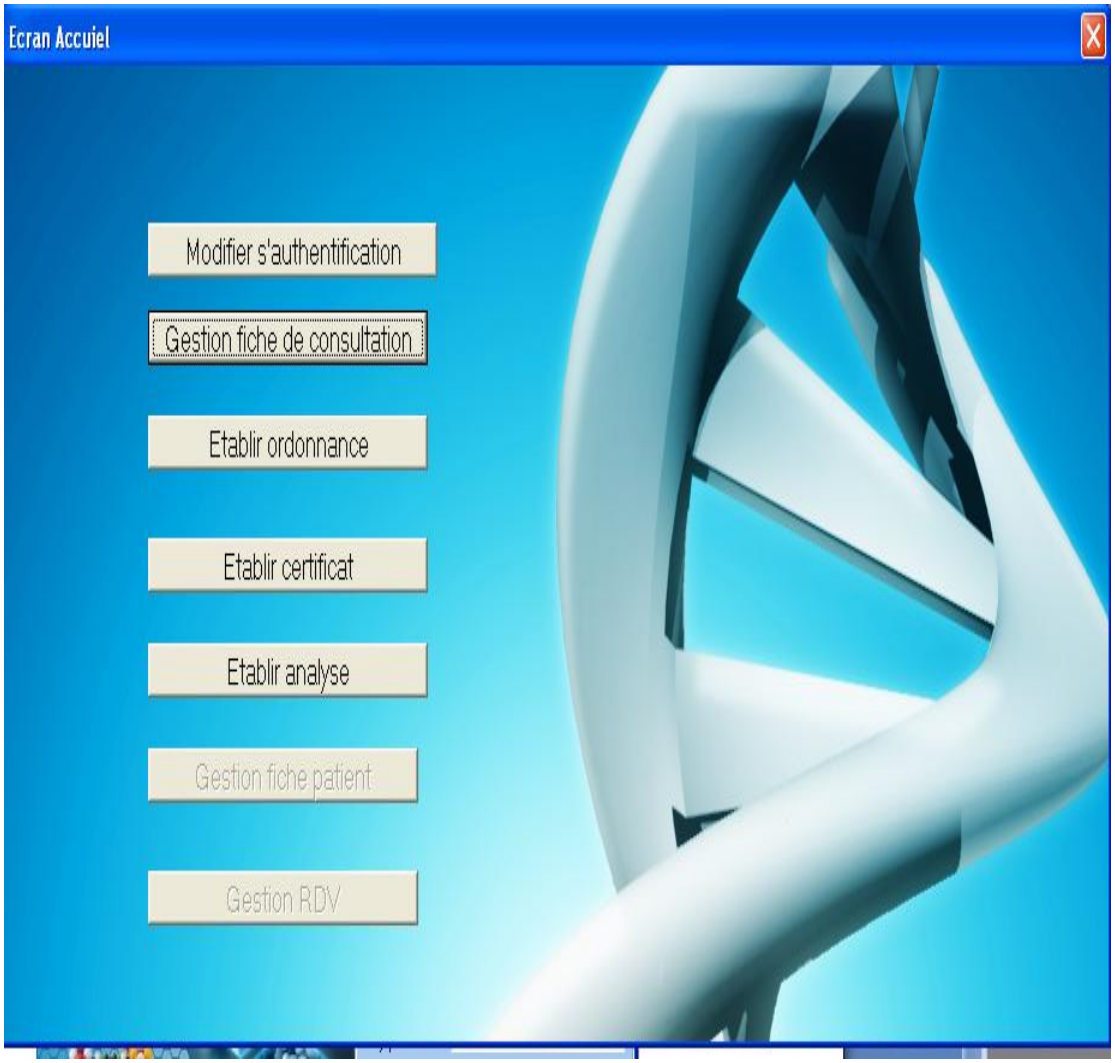


Figure 5.4 Ecran Accueil

3.3 Fenêtre modifier s'authentification

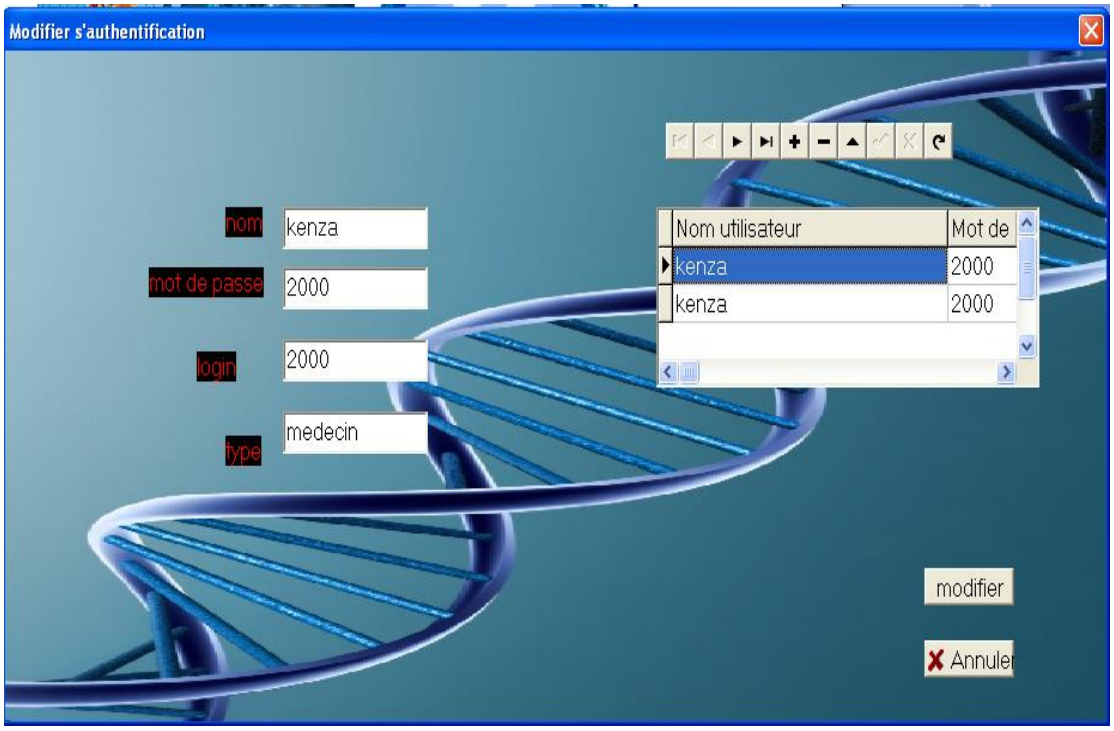
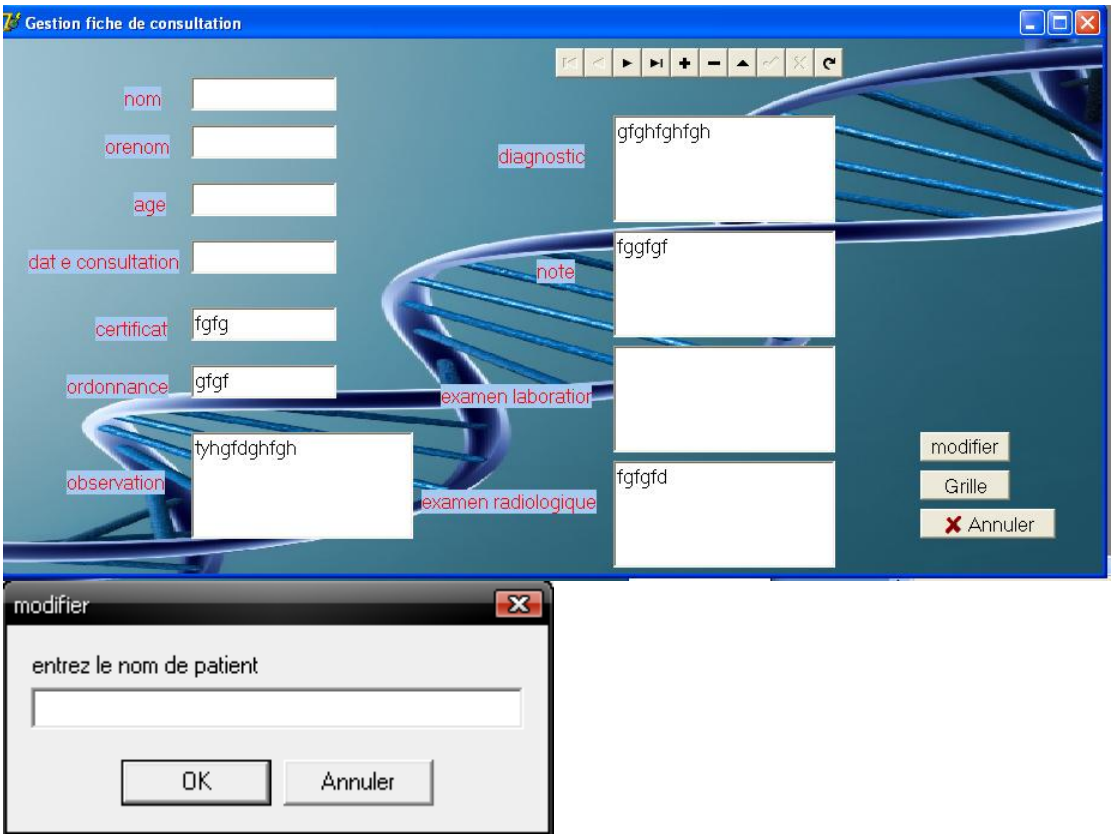


Figure 5.5 modifier s'authentification

3.4 Fenêtre gestion fiche de consultation



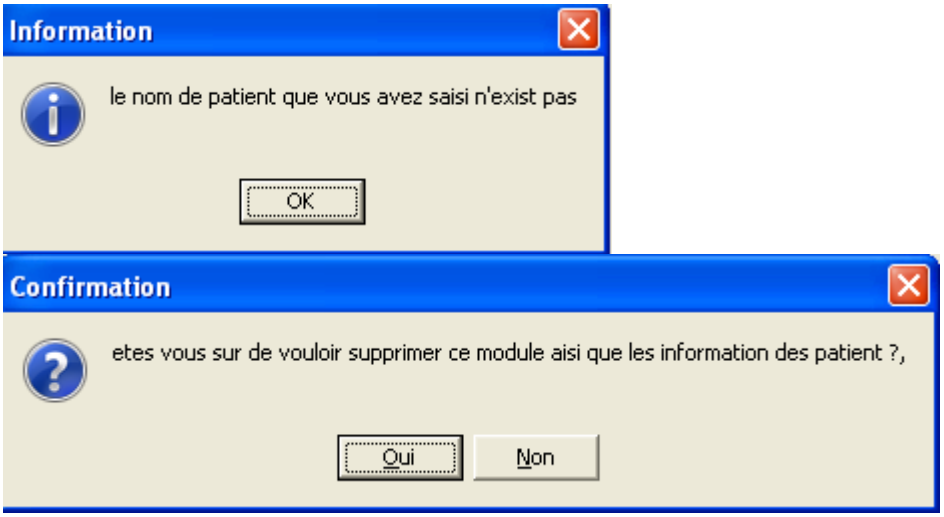


Figure5.6 gestion fiche du consultation

3.5 Gestion patient

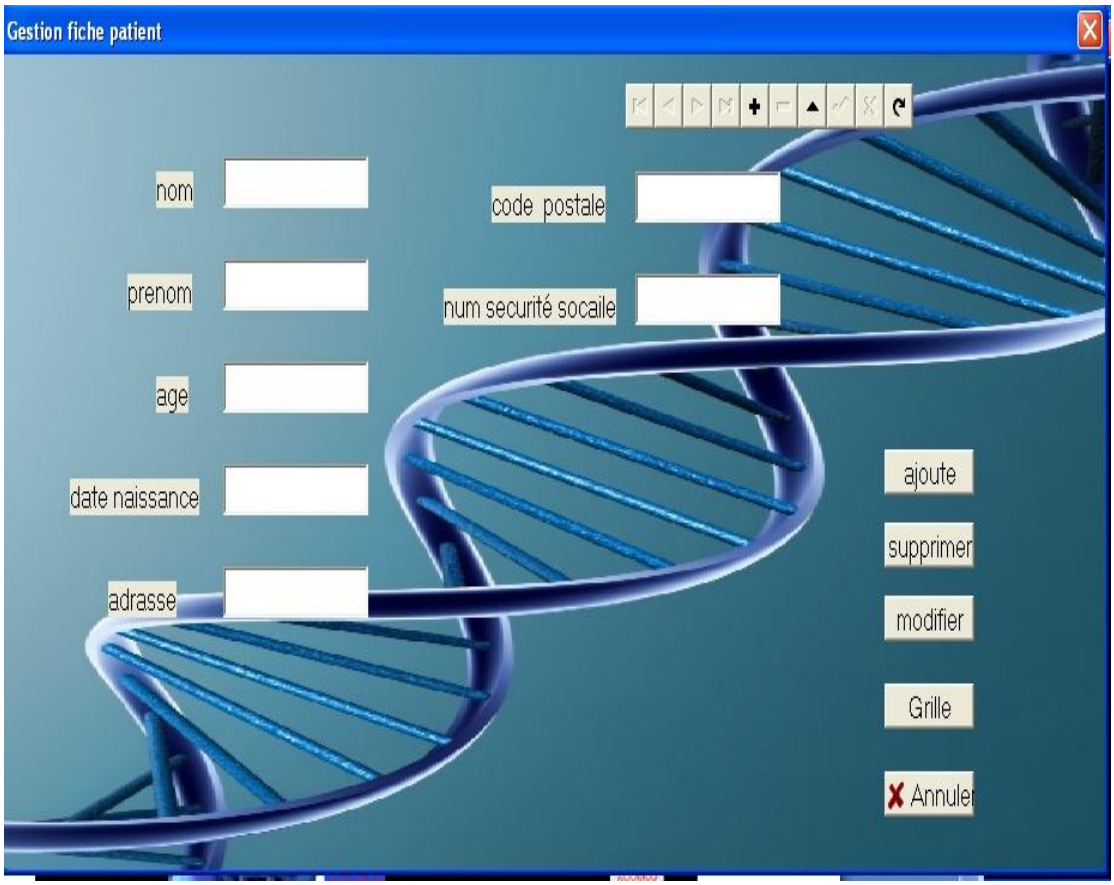
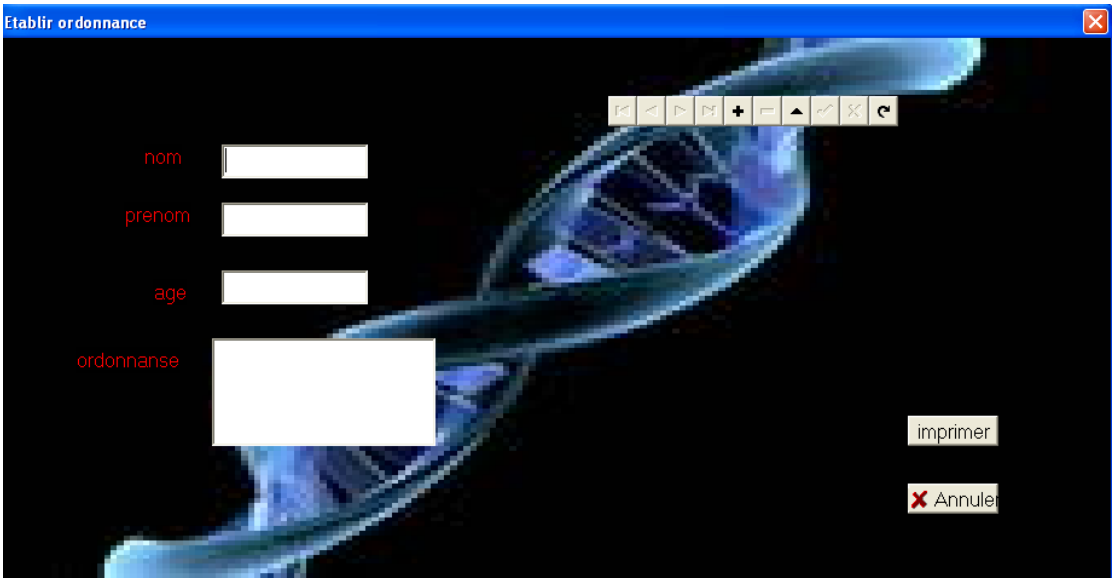


Figure 5.7 gestion fiche patient

3.5 Etablir ordonnance



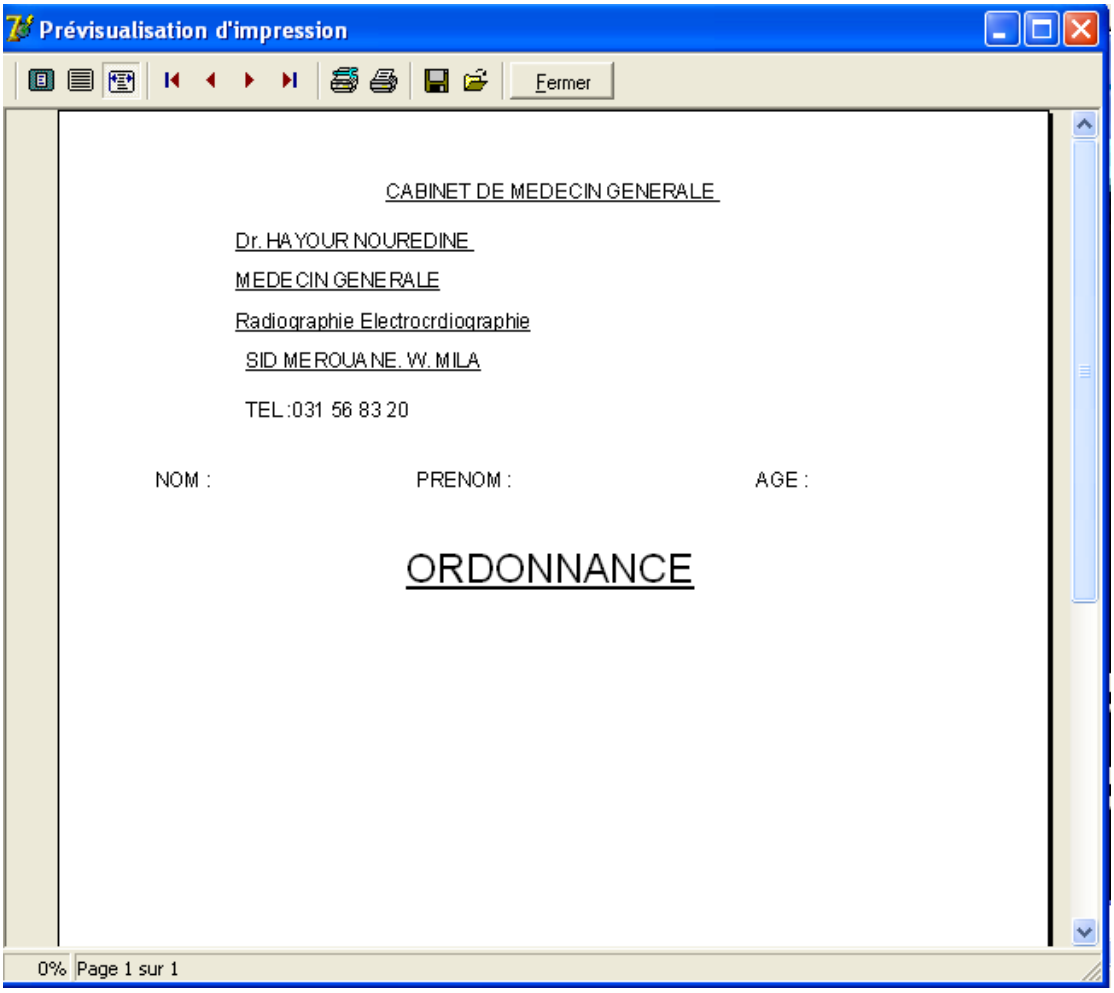


Figure 5.8 établir ordonnance

3.6 Etablir certificat

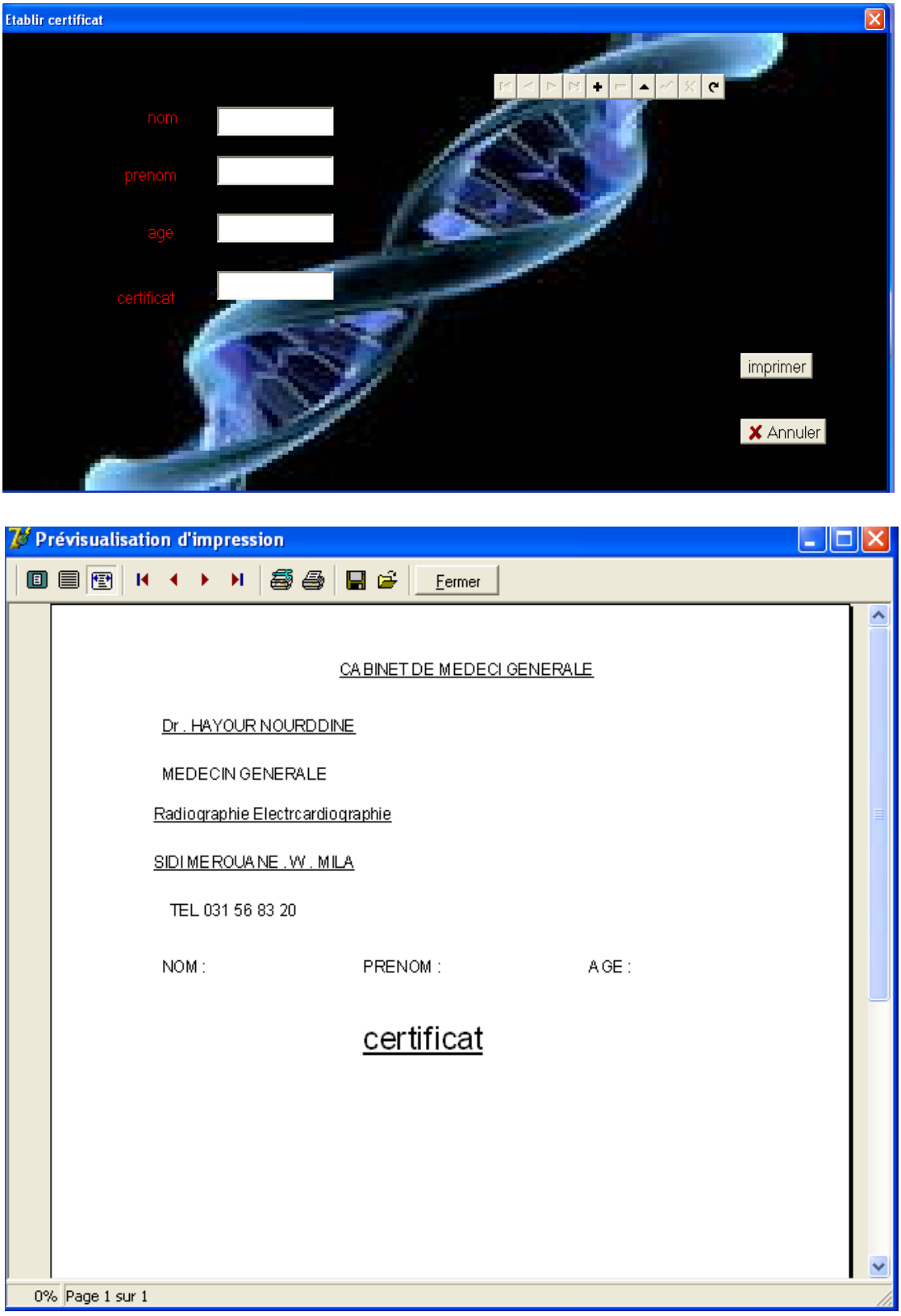
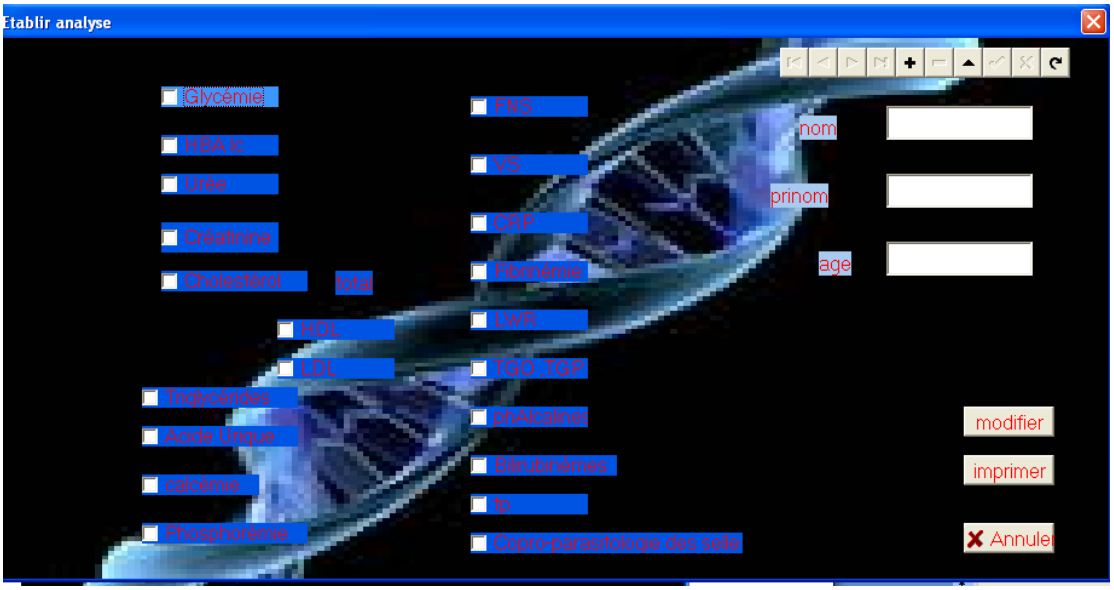


Figure 5.9 établir certificat

3.7 Etablir analyse



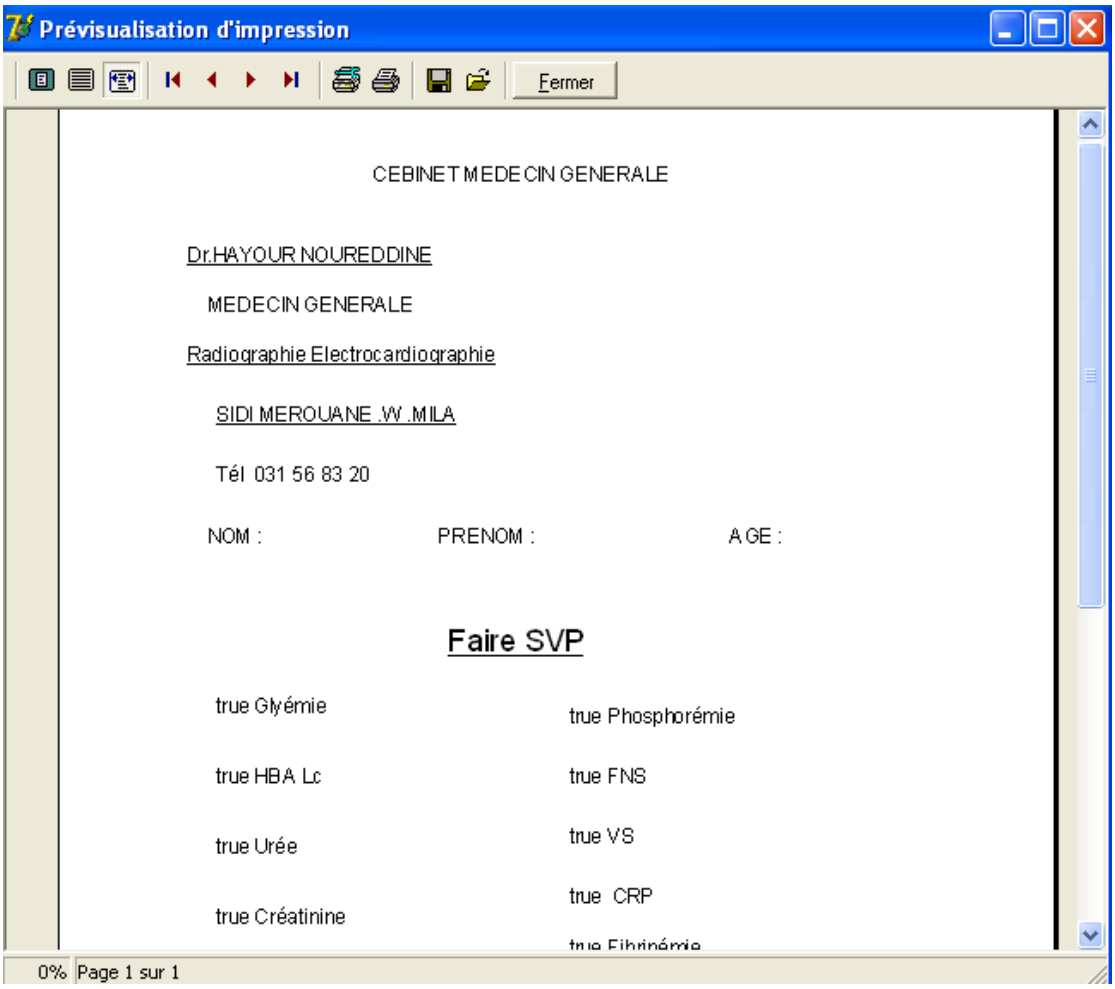


Figure 5.10 établir analyse

3.8 Gestion RDV

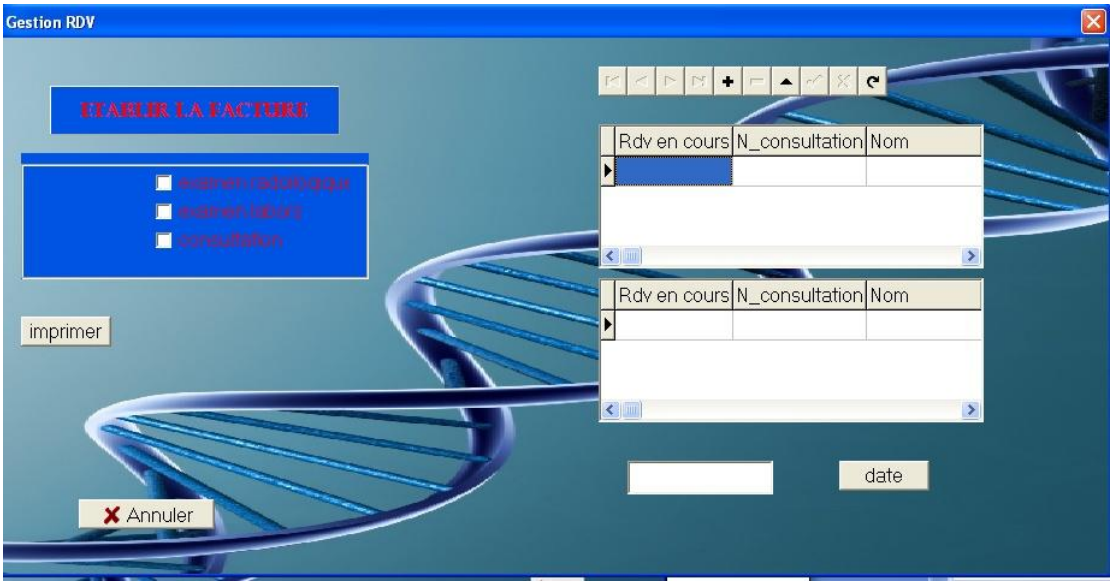


Figure 5.11 Gestion RDV

4.Conclusion : Dans ce chapitre, nous avons décrit brièvement la phase d'implémentation ainsi que l'application que nous avons réalisé.

CONCLUSION GÉNÉRALE

Conclusion générale

Notre projet s'articule autour de la conception et la réalisation d'un système d'information pour la gestion de cabinet médical générale.

Pour mieux comprendre le fonctionnement du système, on a entamé sa modélisation avant sa réalisation. Nous avons utilisé UML comme un langage de modélisation, et pour savoir précisément comment l'utiliser dans le cycle de développement on a suivi la méthode de Robert Audibert. Pour l'implémentation, nous avons porté notre choix respectivement sur le langage Delphi et le système de gestion de base de données Paradoxe.

Nous avons pu produire une application qui peut offrir à ce stade les principaux services d'un système de gestion d'un cabinet médical. L'application n'est pas encore à sa phase finale, mais l'essentiel a été fait, et elle reste ouverte à toute évolution notamment dans la gestion des rendez-vous qui peut être améliorer par une autre méthode plus efficace ainsi que la méthode de gestion des analyses. La période passée au développement de notre application, nous a été d'un apport considérable. En effet, c'est une expérience qui nous a permis d'enrichir nos connaissances dans de domaines très variés comme: L'Orienté Objet, l'UML, la méthode AUDIBER , le langage Delphi, le SGBD paradoxe ...etc. Elle nous à permis aussi de découvrir un domaine en pleine évolution qui est le cœur de toutes activités dans le cabinet médical.

Finalement, notre projet existe pour améliorer le travail du médecin concernant la gestion des dossiers des patients et toute la gestion de son cabinet, en lui faisant l'automatisation de cette dernière dont le but de la rendre plus efficace et rapide et lui laissant s'occuper par sa tâche principale qui est la consultation des malades.

BIBLIOGRAPHIE

[1] : UML 2.0<< Laurent AUDIBERT>>

[2] : Méthodologie UML - Cours du cycle B du Cnam.doc<<DI GALLO Frédéric>>

[3] : UML 2 en action De l'analyse des besoins à la conception<<Pascal roques et Franck Vallée>>

[4] : www.developper.com