



Projet de Fin d'Etudes

Licence Sciences et Techniques Génie Informatique

**Réalisation d'une application pour la gestion des rendez-vous
et des hospitalisations**



Lieu de stage : Centre Hospitalier Universitaire Hassan II de Fès.

Réalisé par :

M'hamdi Hajar

Serhan Ismail

Encadré par :

Pr. LAMRINI Loubna

Mr. MAKHLOUK Mounir

Soutenu le 13/06/2012 devant le jury composé de :

Pr. Loubna LAMRINI

Pr. Mohamed Chawki ABOUNAIMA

Pr. Said NAJAH

Année Universitaire 2013-2014

Dédicace

Nous dédions ce modeste travail, comme preuve de respect et de reconnaissance à :

A nos chers parents

Pour les efforts qu'ils ont consentis pour notre éducation et notre formation, pour leur précieux soutien moral et matériel, pour leurs encouragements continus, et pour leurs sacrifices tout au long de notre vie, que nous serons tellement très reconnaissants.

A nos chers frères

D'avoir être à nos côtés et nous encourager tous le temps.

Dédicace spéciale

Une dédicace spéciale à tous nos enseignants.

Et à vous chers lecteurs

Remerciements

Avant tout développement sur cette expérience professionnelle, il apparaît opportun de commencer ce rapport de stage par des remerciements.

Après Dieu, nous tenons à adresser nos remerciements les plus sincères à tout le corps professoral et administratif de la Faculté des Sciences et Techniques de Fès.

Nous remercions sincèrement nos professeurs Monsieur R.BENABBOU Responsable du département informatique de la FSTF, Monsieur A.ZAHY Responsable de la licence génie informatique de la FSTF qui fournissent d'énormes efforts pour ses étudiants afin qu'ils puissent avoir une formation complète, dans les conditions les plus favorables.

Nous tenons à remercier aussi notre encadrante de stage Mme. LAMRINI Loubna enseignante à la FSTF, pour avoir nous encadrés tout au long de ce stage, aussi d'être source d'information, de communication, d'encadrement et d'orientation technique pendant toute la durée de stage sans hésiter à aucun moment de nous consacrer une part de leur temps précieux afin de nous aider considérablement dans la réalisation de ce travail.

Nous tenons également à adresser nos plus sincères remerciements à l'ensemble du corps du Centre Hospitalier Universitaire Hassan 2 Fès, et plus précisément à notre encadrant professionnelle Monsieur Mounir MAKHLOUK ingénieur d'état au sein du service informatique pour avoir accordé son temps précieux, son attention et son énergie pour nous aider dans la réalisation de ce travail en vue de s'ouvrir davantage et proprement sur le métier de demain.

Sommaire

Liste des figures	6
Abréviations	7
Introduction	8
Chapitre I : Contexte général du projet.....	9
1. Organisme d'accueil CHU	10
1.1. Présentation du CHU Fès.....	10
1.2. Organigramme du CHU	11
1.3. Organisation des consultations et hospitalisations	11
1.4. Service informatique.....	11
2. Présentation du projet.....	12
2.1. Cahier de charge:	12
2.2. Enoncé du problème	13
2.3. Solutions proposées	14
Chapitre II : Analyse fonctionnelle	15
1. Méthodologie d'analyse	16
1.1. Le langage UML	16
1.2. Le Modèle Incrémental et Itératif	16
1.2.1. Avantages	17
1.2.2. Inconvénients	17
1.2.3. Incréments principal du projet	18
1.3. Le Modèle MVC (Modèle-Vue-Contrôleur)	18
1.3.1. Définition MVC (Modèle-Vue-Contrôleur) [3].....	19
1.3.2. Avantages du MVC.....	19
2. Modélisation du contexte	20
2.1. Les acteurs et leurs rôles	20
2.2. Les messages émis et reçus	21
3. Analyse et conception	23
3.1. Diagramme de package	23
3.2. Diagrammes des cas d'utilisation	25
3.3. Diagrammes de séquences	29

3.4. Diagramme de classes et sa description	32
3.5. La base de données	35
Chapitre III : Présentation de l'application	36
1. Outils et technologies de développement	37
2. Schéma général de l'application.....	41
3. Présentation de l'application.....	42
Conclusion et perspectives	48
<i>Bibliographie et Webographie</i>	49
Annexe N°1	51
Annexe N°2	52
Annexe N°3	53

Liste des figures

Figure 1: Organigramme CHU Hassan II.....	11
Figure 2: Cycle de vie Modèle Incrémental et Itératif.....	16
Figure 3: Les incréments du Modèle Incrémental et Itératif.....	17
Figure 4: Architecture du modèle MVC.....	18
Figure 5: Acteurs principaux	20
Figure 6: Diagramme de package par acteur.....	24
Figure 7: Diagramme des cas d'utilisation pour Secrétaire.....	25
Figure 8: Diagramme des cas d'utilisation pour Major.....	26
Figure 9: Diagramme des cas d'utilisation pour Médecin	27
Figure 10: Diagramme des cas d'utilisation pour Administrateur	28
Figure 11: Diagramme de séquences d'authentification	29
Figure 12: Diagramme de séquences d'ajout patient.....	30
Figure 13: Diagramme de séquences recherche et modification patient.....	31
Figure 14: Diagramme de séquences d'ajout compte	32
Figure 15: Diagramme de classes.....	33
Figure 16: Les tables de la base de données	35
Figure 17: Architecture du framework CodeIgniter	39
Figure 18: Structure du CodeIgniter	41
Figure 19: Schéma général de l'application	41
Figure 20: Fenêtre d'Authentification.....	42
Figure 21: Message d'une fausse authentification	42
Figure 22: Menu d'administrateur	42
Figure 23: Gestion des comptes.....	43
Figure 24: Liste des personnels.....	43
Figure 25: Gestion des services	44
Figure 26: page d'ajout un nouveau service	44
Figure 27: Liste des services.....	45
Figure 28: page d'ajout un nouveau lit	45
Figure 29: Page d'ajout un nouveau agenda.....	46
Figure 30: Message d'erreur	47
Figure 31: Fenêtre de gestion des rendez-vous	47

Abréviations

- FSTF : Faculté des Sciences et Techniques Fès.
- CHU : Centre HospitalierUniversitaire.
- UTP : Unshielded twisted pair.
- FTP : Foiled Twisted Pair.
- RDV : Rendez- Vous.
- IP : Identifiant Patient.
- PT : Patient.
- UML : Langage Unifié pour la Modélisation objet.
- PHP : HypertextPreprocessor.
- MVC : Modèle Vue Contrôleur.
- SMTP : Simple Mail Transfer Protocol.
- HTTP : HyperText Transfer Protocol.
- JS : JavaScript.
- HTML: Hypertext Markup Language.

Introduction

Le présent document est le fruit de notre travail qui s'inscrit dans le cadre du projet de fin d'étude effectué au sein du CHU Hassan II -Fès en vue d'obtenir la licence de la Faculté des Sciences et Techniques Fès.

Ce projet a pour but de réaliser une application web pour la gestion des hospitalisations, consultations et prise des rendez-vous des patients avec une interface simple à manipuler.

En effet, la période du stage est une étape très importante dans le processus de formation, qui enrichit les connaissances et surtout qui aide à découvrir de plus près la vie professionnelle.

Durant notre projet, nous aurons pour mission dans un premier temps de cerner le sujet. Après une analyse approfondie de la problématique, nous avons élaboré les différents diagrammes. Ensuite, nous avons abordé la phase de la mise en œuvre et de l'implémentation de la solution. La dernière étape a fait l'objet du déploiement des tests et de validation.

Pour bien mener notre projet, nous avons choisi de suivre un cycle de développement Modèle Incrémental et Itératif. C'est une démarche qui a fait ses preuves dans le domaine des projets informatiques de grande taille.

Durant une période de stage allant du 07 Avril au 07 Juin 2014, nous aurions élaborés 3 grandes parties :

La première partie définit le contexte général du projet en présentant l'organigramme d'accueil et en définissant la problématique du projet ainsi que la solution proposée.

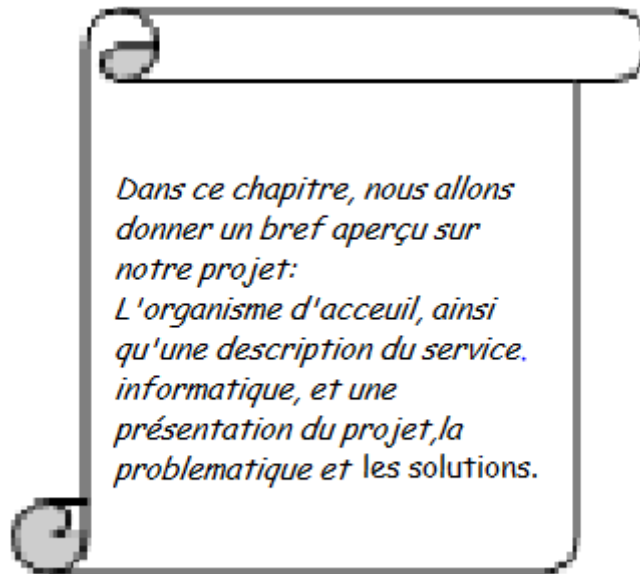
Dans la deuxième partie, nous présentons l'analyse fonctionnelle du projet en décrivant les fonctionnalités du système ainsi que l'étude conceptuelle qui constitue les différents diagrammes UML.

La troisième partie sera consacrée aux outils et langages de développement utilisés, à la réalisation du projet et la présentation de l'application.

Enfin une conclusion et des perspectives du travail seront citées.

Chapitre I

Contexte général du projet



1. Organisme d'accueil CHU

1.1. Présentation du CHU Fès

Le Centre Hospitalier et Universitaires de Fès(CHU) est un établissement public de santé doté de personnalité morale et d'autonomie financière. Plus d'informations peuvent être présentées au Tableau N°1.

Tableau N° 1: carte d'identité du centre

Date de création	:	30 Août 2001
Date de mise en service:		05 Août 2002
Statut	:	Etablissement public de santé doté de personnalité morale et d'autonomie financière
Missions:		<ul style="list-style-type: none"> ✓ Dispenser des soins à toute personne dont l'état requiert ses services, de jour comme la nuit, en veillant à assurer la qualité d'accès et la continuité des soins. ✓ Conduire des travaux de recherche médicale dans le strict respect de l'intégrité physique, morale et de la dignité des malades. ✓ Participer à l'enseignement clinique universitaire et postuniversitaire médical et pharmaceutique ainsi qu'à la formation du personnel paramédical.
Organisation	:	Le Centre Hospitalier Hassan II de Fès est constitué d'une direction et des services hospitaliers.
Composition	:	<ul style="list-style-type: none"> ✓ Hôpital des Spécialités. ✓ Hôpital Mère et Enfant. ✓ Hôpital d'Oncologie et de Médecine Nucléaire. ✓ Hôpital OMAR DRISSI. ✓ Hôpital IBN AL HASSAN.
Capacité Litière	:	880 Lits.
Surface couverte	:	78 102 m ² .
Assiette foncière	:	12 ha.
Coût global	:	1,2 milliard de DH.
Adresse	:	CH Hassan II, route de SidiHarazem, B ,P 1835, Atlas Fés-MAROC.
Téléphones:		<ul style="list-style-type: none"> ✓ Tél : 00212 (0) 535 619 052. ✓ Fax : 00212 (0) 535 619 053.
E-mail	:	contact@chufes.ma
Site	:	www.chufes.ma

1.2. Organigramme du CHU

Le CHU se compose d'une direction, de trois divisions administratives et médicales et plusieurs services (Fig1).

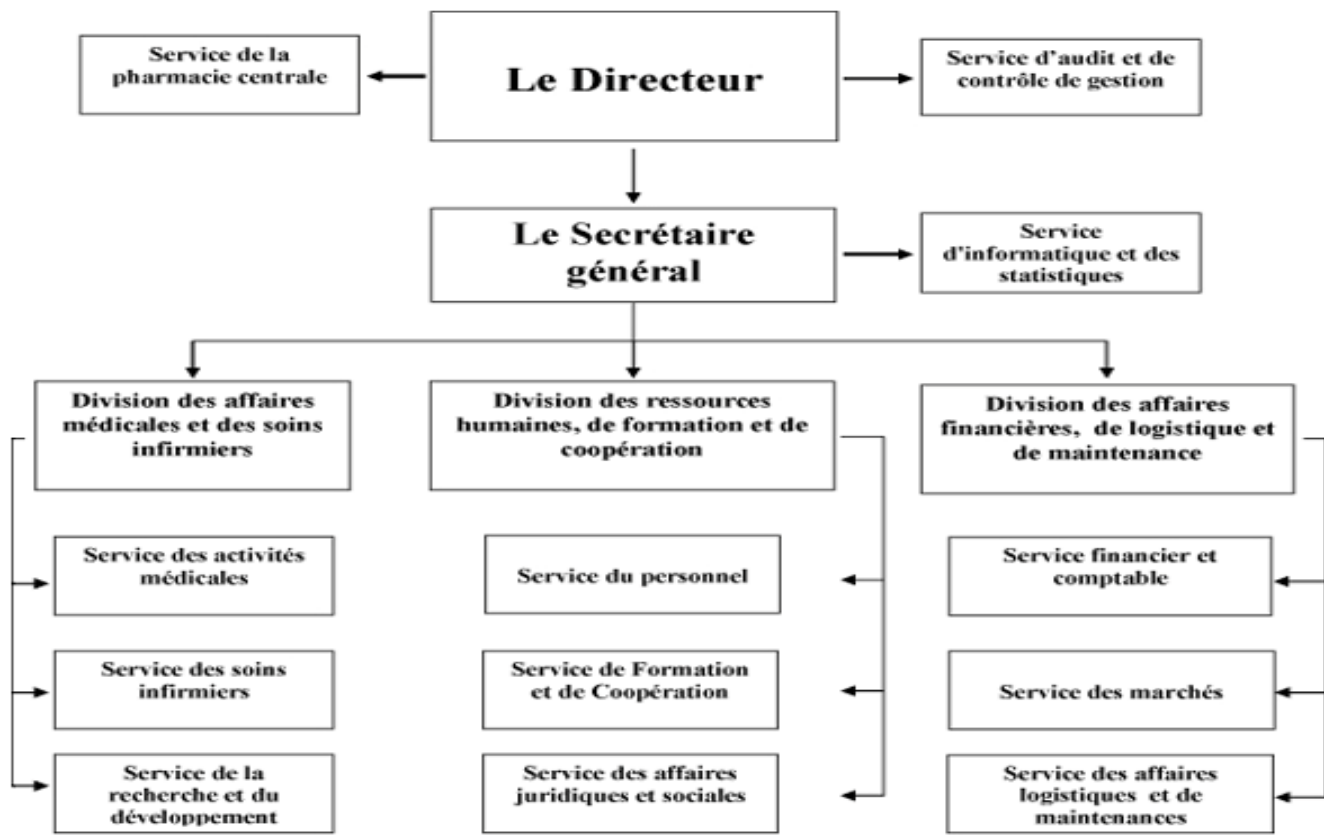


Figure 1: Organigramme CHU Hassan II

1.3. Organisation des consultations et hospitalisations

Le patient a deux portes d'entrée soit des urgences ou du centre de diagnostic :

Au niveau du centre du diagnostic le patient bénéficie d'une consultation et d'un examen après avoir présenté un billet de référence et un bon de consultation. Au terme de la consultation il y a une décision d'hospitalisation ou un rendez-vous ou bien une sortie. (Annexes N°1 et N°3)

Au niveau des urgences l'utilisateur bénéficie d'une consultation suite à laquelle il y a une décision d'hospitalisation ou une sortie. (Annexe N°2)

La décision de l'hospitalisation est le résultat des actes effectués au niveau des urgences et du centre de diagnostic. (Annexe N°3)

1.4. Service informatique

Le service informatique se compose de trois cellules :

- Cellule de développement et du système d'information : a pour mission de résoudre tous les problèmes en relation avec le système d'information hospitalier.
- Cellule réseau : a pour mission la maintenance et le contrôle du réseau informatique du CHU

- Equipements réseaux :
 - Un routeur Cisco.
 - Un pare-feu physique.
 - Deux Switch fédérateurs.
- Les connexions réseaux :
 - Le CHU est connecté avec l'extérieur par une ligne spéciale de 4Mbp/s de débit.
 - Le réseau local du CHU est un réseau lié par des câbles UTP (Paire torsadée non blindée) et FTP (Paire torsadée écrantée) catégorie 6.
 - L'interconnexion entre les services est effectuée pas une liaison fibre optique.
- Cellule télécom : gère et maintient le réseau de la téléphonie au sein du CHU.

Au total le service informatique assure:

- un support de qualité aux problèmes déclenchés au niveau du système d'information.
- le bon fonctionnement du réseau de la téléphonie au sein du CHU.
- la maintenance du matériel informatique.
- le monitoring du réseau informatique.

2. Présentation du projet

2.1. Cahier de charge

Le CHU Hassan II –Fès est doté d'une infrastructure réseau dans tous les services qui sont au nombre de 46. Chaque service est constitué de quatre acteurs (Médecin, Major, Secrétaire, et un administrateur).

L'administrateur gère tous les services. Les quatre acteurs nécessitent une authentification par le Login et le mot de passe.

Chacun de ces acteurs a ses propres tâches :

- Le Médecin: c'est l'acteur principal. Au début il effectue la consultation, pose un diagnostic, rédige une ordonnance à titre ambulatoire, décide une hospitalisation, ou réfère pour un RDV. Une fois le patient dans le service, le médecin complète le diagnostic par des examens para-cliniques, demande parfois l'avis d'autres spécialistes et prend en charge le malade.
- La secrétaire: effectue la recherche du patient à l'aide de l'identifiant patient (IP) dans la base de donnée, sinon elle peut ajouter le patient en saisissant ses informations (nom, prénom, CIN, âge, sexe, adresse, téléphone...), et les modifier en cas d'erreur. La secrétaire affecte le premier RDV lors du premier contact du malade, elle peut aussi lui affecter un RDV au moment de la consultation de son dossier et ceci après avoir consulté les RDV des médecins. Elle peut aussi annuler ou modifier un RDV selon la demande du médecin.

- **Le major**: effectue le même travail de la secrétaire. Il consulte tout d'abord le dossier du patient et vérifie la décision prise (hospitalisation, transfert, sortie). En cas d'hospitalisation il doit lister les lits et s'assurer de la disponibilité de place tout en respectant la contrainte genre (sexe). S'il trouve une place vacante le patient est hospitalisé, sinon il l'inscrit de nouveau dans la liste d'attente.
En cas de décision d'un transfert dans un autre service, le major s'assure de la disponibilité d'un lit sinon le malade est maintenu jusqu'à une réponse favorable. Finalement le major contribue à la sortie des malades.
- **L'administrateur**: est un acteur unique. Il effectue la mise à jour de la gestion des comptes du personnel, des services (salles et lits), des agendas et des activités.

2.2. Enoncé du problème

Le Centre Hospitalier Universitaire Hassan II de Fès possède une quarantaine de services. Chaque service accueille quotidiennement un grand nombre de patients de la région de Fès ou d'autres régions. Ceci pose un véritable problème de gestion des hospitalisations, consultations et des rendez-vous.

Actuellement, ces services sont gérés avec l'outil Microsoft Excel d'une manière quasi manuelle, il est un peu compliqué dans son utilisation, de plus il est lent dans la recherche et le listage.

Cette méthode de travail possède un nombre important de problèmes tels que :

- Problème de gestion de l'information : gestion des supports (billet d'hospitalisation, registre des admissions, dossier médical, fiche de prestation...) lenteur d'analyse et de prise de décision.
- Problème de modification et mise à jour des supports
- Problème de gestion des RDV : en cas de changement, d'annulation ou d'ajout.
- Problème de gestion des lits : méconnaissance de disponibilité de place en relation avec le genre (sexe).
- Problème de sécurité : n'importe quelle personne peut accéder aux informations (secret professionnel).

Ces problèmes engendrent :

- Des frustrations des professionnels de santé, administratif et des patients.
- Une anarchie dans le travail.
- Une perte de temps.
- une mauvaise coordination entre les services.

Nous avons proposé la mise en place d'une application informatique pour mieux gérer les consultations, les hospitalisations et les RDV.

2.3. Solutions proposées

La résolution de ce problème consiste à développer une application web pour améliorer la gestion au sein du CHU Hassan II -Fès. L'application va être développée en PHP selon l'architecture MVC.

Cette application fera gagner un temps colossal et rendra le travail plus organisé. On va transformer la méthode de travail classique et statique en une autre dynamique.

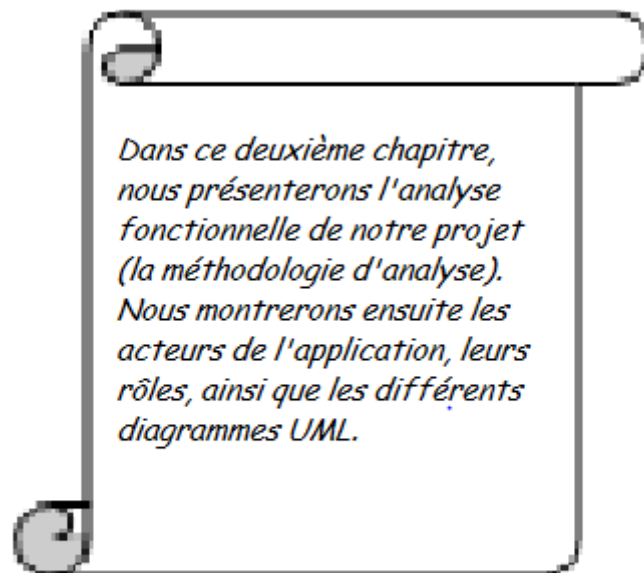
L'application va garantir un traitement automatisé de ces procédures en utilisant des interfaces graphiques simples et faciles à comprendre et qui va en particulier :

- ✓ Organiser le travail de la secrétaire, du major, du médecin ainsi que de l'administrateur.
- ✓ Permettre aux 4 acteurs de rechercher l'information dont ils ont besoin en un temps réduit.
- ✓ Organiser la gestion des lits et des RDV.
- ✓ Faciliter la communication entre les différents acteurs.
- ✓ Faciliter la mise à jour de la gestion des comptes.
- ✓ Assurer l'utilisation de l'application d'une façon plus sécurisée.

Les solutions proposées seront détaillées dans les chapitres qui suivent.

Chapitre II

Analyse fonctionnelle



1. Méthodologie d'analyse

1.1. Le langage UML[1]

UML ou Langage de Modélisation Unifié, est un langage de modélisation graphique à base de pictogrammes. Il est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet. UML est couramment utilisé dans les projets logiciels. Les différents éléments sont :

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

Grâce aux outils de modélisation UML, il est également possible de générer automatiquement une partie code, par exemple en langage Java, à partir des divers documents réalisés.

1.2. Le Modèle Incrémental et Itératif [2]

La phase d'étude est la partie la plus importante pour tout projet réussi.

On s'est basé durant la réalisation de notre application à des normes universelles durant la conception, en particulier le respect des principes du Modèle Incrémental.

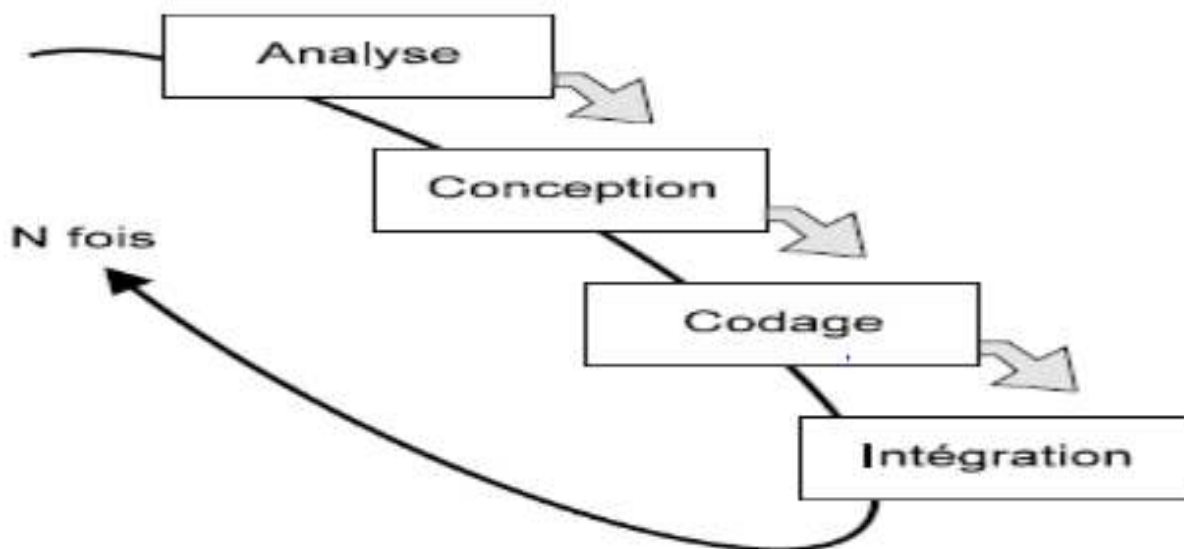


Figure 2: Cycle de vie Modèle Incrémental et Itératif

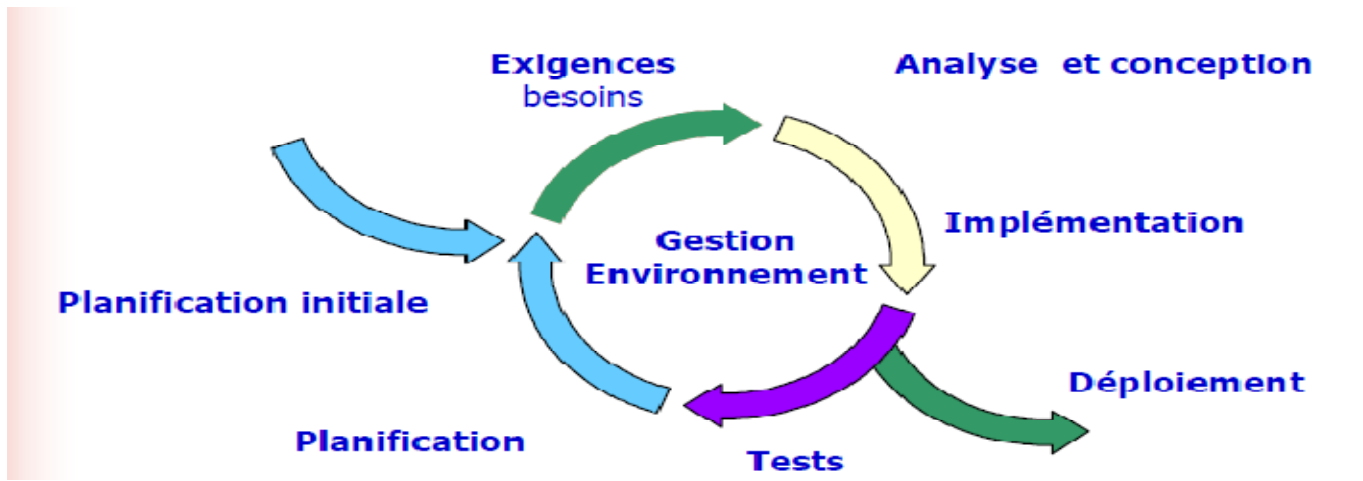


Figure 3: Les incréments du Modèle Incrémental et Itératif

- Le projet de développement est découpé en plusieurs petits projets.
- Chaque projet représente une itération qui:
 - Donne lieu à un incrément (version du produit).
 - Prend en charge une partie des besoins.
 - Répond à un ensemble de risques.
- Le développement se déroule en plusieurs itérations.
- Le projet est décomposé en un noyau et plusieurs incréments.
- Chaque incrément est développé séparément ou en parallèle.

1.2.1. Avantages

- Flexibilité (agilité) vis à vis de nouveaux besoins ou des changements.
- Pas de blocage en cas de spécifications incomplètes.
- Meilleure testabilité.
- Découverte de malentendu assez tôt pour les corriger.
- Répartition de l'effort dans le temps.
- Objectifs réduits et clairs.
- Utilisation de l'approche «diviser pour régner».
- Le client rentre en relation avec le produit très tôt.

1.2.2. Inconvénients

- Difficultés de gestion du projet.
- Difficultés de contrôle qualité.
- Exigence d'une bonne planification et d'une bonne conception.
- Exigence d'une vision sur le produit fini pour bien diviser en incréments.

1.2.3. Incréments principal du projet

Notre projet est constitué de 10 incréments principaux :

- Incrément gestion de compte.
- Incrément gestion de patient.
- Incrément gestion des services.
- Incrément gestion des salles.
- Incrément gestion des lits.
- Incrément gestion des activités.
- Incrément gestion des agendas.
- Incrément gestion des RDV.
- Incrément gestion de consultation.
- Incrément gestion d'hospitalisation.

1.3. Le Modèle MVC (Modèle-Vue-Contrôleur)

L'architecture MVC (modèle, vue et contrôleur) est un concept très puissant qui intervient dans la réalisation d'une application. Son principal intérêt est la séparation des données (modèle), de l'affichage (vue) et des actions (contrôleur), ce qui assure la clarté de l'architecture et simplifie la tâche du développeur responsable de la maintenance et de l'amélioration du projet. Les différentes interactions entre le modèle, la vue et le contrôleur sont résumées par le schéma de la figure 4.

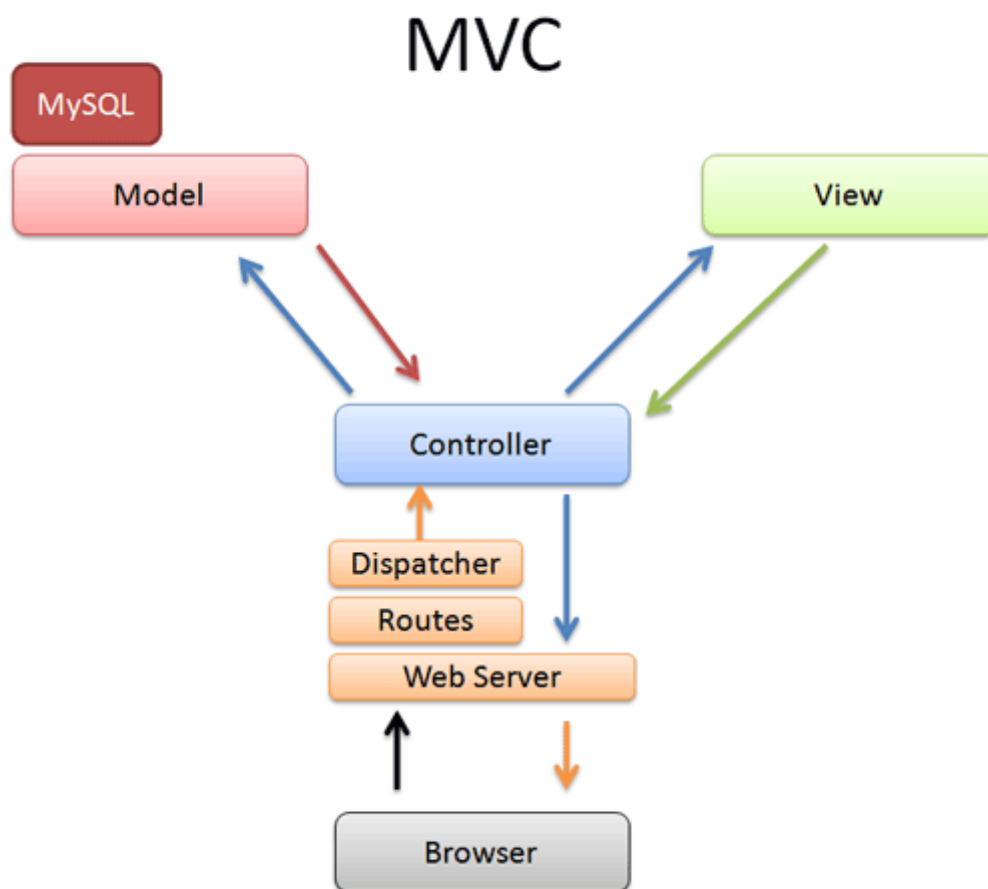


Figure 4: Architecture du modèle MVC

1.3.1. Définition MVC (Modèle-Vue-Contrôleur) [3]

➤ **Le Modèle :**

Le modèle représente le cœur de l'application : traitements des données, interactions avec la base de données. Il décrit les données manipulées par l'application. Il regroupe la gestion de ces données et est responsable de leur intégrité. La base de données sera l'un de ses composants. Le modèle comporte des méthodes standards pour mettre à jour ces données (insertion, suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ces données. Les résultats renvoyés par le modèle ne s'occupent pas de la présentation, Le modèle ne contient aucun lien direct vers la vue.

➤ **Le Contrôleur :**

Le contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de l'utilisateur et déclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle et ce dernier notifie la vue que les données ont changée pour qu'elle se mette à jour. D'après le patron de conception observateur/observable, la vue est un « observateur » du modèle qui est « observable ».

Certains événements de l'utilisateur ne concernent pas les données mais la vue. Dans ce cas, le contrôleur demande à la vue de se modifier. Le contrôleur n'effectue aucun traitement, ne modifie aucune donnée, il analyse la requête du client et se contente d'appeler le modèle adéquat et de renvoyer la vue correspondant à la demande.

➤ **La Vue :**

C'est avec quoi l'utilisateur interagit se nomme précisément la vue. Sa première tâche est de présenter les résultats renvoyés par le modèle, sa seconde tâche est de recevoir toute action de l'utilisateur (clic de souris, sélection d'un bouton radio, coche d'une case, entrée de texte, de mouvements, de voix, etc). Ces différents événements sont envoyés au contrôleur.

La vue n'effectue pas de traitement, elle se contente d'afficher les résultats des traitements effectués par le modèle et d'interagir avec l'utilisateur.

1.3.2. Avantages du MVC

- ✓ Une conception claire et efficace grâce à la séparation des données de la vue et du contrôleur.
- ✓ Un gain de temps de maintenance et d'évolution du site.
- ✓ Une plus grande souplesse pour organiser le développement du site entre différents développeurs (indépendance des données, de l'affichage et des actions).

2. Modélisation du contexte

2.1. Les acteurs et leurs rôles

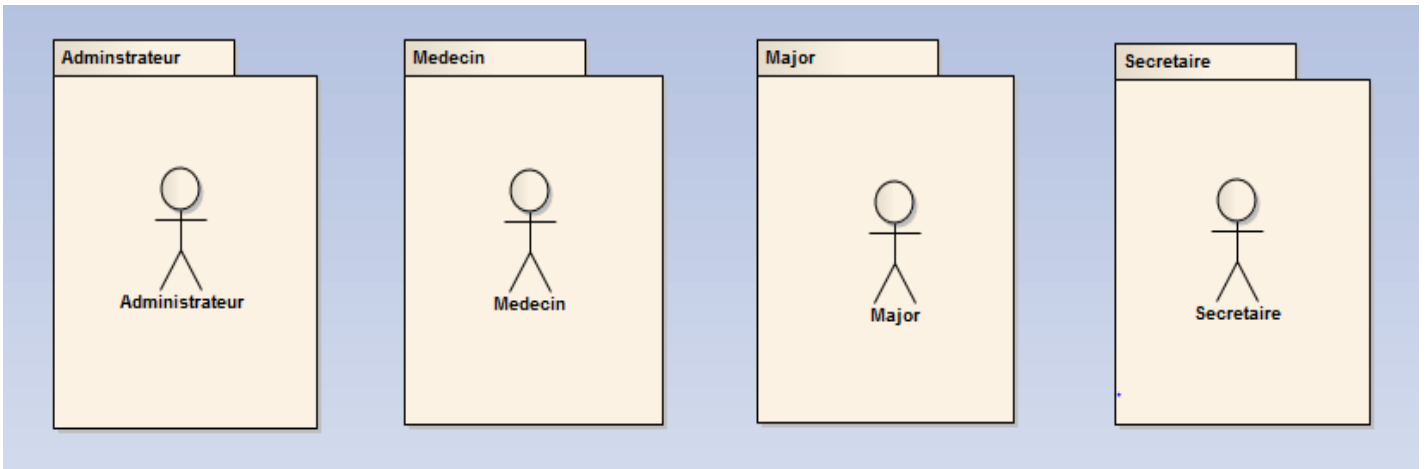






















Figure 5: Acteurs principaux

Tableau N° 2: Rôle des acteurs






<i>Acteurs</i>	<i>Rôles</i>
<i>Secrétaire</i>	<ul style="list-style-type: none"> - Ajoute un nouveau patient. - Recherche/Modifie informations du patient. - Consulte la liste des RDV. - Affecte un RDV au patient. - Modifie/Annule un RDV. - Liste les agendas des médecins.
<i>Major</i>	<p>En sus des actions de la secrétaire il :</p> <ul style="list-style-type: none"> - Liste les lits et vérifie leur disponibilité. - Consulte le dossier du patient. - Modifie le dossier du patient. - Contribue à l'hospitalisation. - Etablit le transfert/Sortie du patient.
<i>Médecin</i>	<ul style="list-style-type: none"> - Consulte dossier du patient. - Modifie dossier du patient. - Décide l'hospitalisation. - Prescrit le traitement et la prise en charge. - Ordonne le transfert.

	<ul style="list-style-type: none"> - Ajoute/Modifie/Supprime un RDV à son agenda.
<i>Administrateur</i>	<p>En susdes tâches des acteurs précèdent il gère :</p> <ul style="list-style-type: none"> - les comptesdes personnels et les services. - les salles et les lits. - les agendas et les activités.

2.2. Les messages émis et reçus

<i>Cas d'utilisation</i>	<i>Acteurs</i>	<i>Messages émis /Messages reçus</i>
<i>Ajouter patient</i>	<ul style="list-style-type: none"> ▪ Administrateur ▪ Major ▪ Secrétaire 	<p> Emis:Ajouter informations d'un nouveau patient.</p> <p> Reçus:Confirmation.</p>
<i>Rechercher patient</i>	<ul style="list-style-type: none"> ▪ Administrateur ▪ Major ▪ Secrétaire 	<p> Emis:Recherche par IPP.</p> <p> Reçus:Résultat de recherche.</p>
<i>Modifier patient</i>	<ul style="list-style-type: none"> ▪ Administrateur ▪ Major ▪ Secrétaire 	<p> Emis:Choisir le patient à modifier à partir de son IPP.</p> <p> Reçus:demande de spécifier les changements et validation.</p>
<i>Affecter RDV</i>	<ul style="list-style-type: none"> ▪ Administrateur ▪ Major ▪ Secrétaire ▪ Médecin 	<p> Emis:ajouter information sur RDV (agenda, service, date de RDV)</p> <p> Reçus:demande de vérifier disponibilité et confirmation.</p>
<i>Modifier RDV</i>	<ul style="list-style-type: none"> ▪ Administrateur ▪ Major ▪ Secrétaire ▪ Médecin 	<p> Emis:choisir le RDV à modifier à partir d'IPP, date et service.</p> <p> Reçus:demande de spécifier les changements et validation.</p>
<i>Supprimer RDV</i>	<ul style="list-style-type: none"> ▪ Administrateur ▪ Médecin ▪ Major ▪ Secrétaire 	<p> Emis:choisir le RDV à supprimer.</p> <p> Reçus:confirmation.</p>
<i>Lister RDV</i>	<ul style="list-style-type: none"> ▪ Administrateur ▪ Médecin ▪ Major 	<p> Emis:Demande de choisir la catégorie de listage.</p> <p> Reçus: la liste des RDV de la catégorie choisie.</p>
<i>Lister consultations</i>	<ul style="list-style-type: none"> ▪ Administrateur ▪ Médecin ▪ Major 	<p> Emis:demande de listage.</p> <p> Reçus:la liste de consultations demandées.</p>
<i>Historique hospitalisation</i>	<ul style="list-style-type: none"> ▪ Administrateur ▪ Médecin ▪ Major 	<p> Emis: demande d'afficher historique des hospitalisations.</p> <p> Reçus:résultat de la demande.</p>
<i>Modifier-Dossier-Patient</i>	<ul style="list-style-type: none"> ▪ Administrateur ▪ Major 	<p> Emis: chercher dossier du patient à modifier.</p> <p> Reçus:demande de spécifier les changements et validation.</p>

<i>Ajouter hospitalisation</i>	<ul style="list-style-type: none"> ▪ Administrateur ▪ Médecin ▪ Major 	<p> Emis: lister les lits, vérifier leur disponibilité et ajouté hospitalisation.</p> <p> Reçus: confirmation d'ajout.</p>
<i>Modifier hospitalisation</i>	<ul style="list-style-type: none"> ▪ Administrateur ▪ Médecin ▪ Major 	<p> Emis: chercher dossier du patient à modifier.</p> <p> Reçus: demande de spécifier les changements et validation.</p>
<i>Annuler hospitalisation</i>	<ul style="list-style-type: none"> ▪ Administrateur ▪ Médecin 	<p> Emis: Fenêtre pour annuler une hospitalisation</p>
<i>Sortie</i>	<ul style="list-style-type: none"> ▪ Administrateur ▪ Médecin ▪ Major 	<p> Emis: modifier état d'hospitalisation</p> <p> Reçus: validation.</p>
<i>Ajouter consultation</i>	<ul style="list-style-type: none"> ▪ Administrateur ▪ Médecin 	<p> Emis: ajouter diagnostic de la consultation.</p> <p> Reçus: confirmation.</p>
<i>Modifier consultation</i>	<ul style="list-style-type: none"> ▪ Administrateur ▪ Médecin 	<p> Emis: modifier informations de la consultation.</p> <p> Reçus : confirmation.</p>
<i>Supprimer consultation</i>	<ul style="list-style-type: none"> ▪ Administrateur ▪ Médecin 	<p> Emis: choisir la consultation à supprimer.</p> <p> Reçus : confirmation.</p>
<i>Ajouter compte</i>	<ul style="list-style-type: none"> ▪ Administrateur 	<p> Emis: ajouter informations concernant le personnel.</p> <p> Reçus : validation.</p>
<i>Modifier compte</i>	<ul style="list-style-type: none"> ▪ Administrateur 	<p> Emis: choisir le personnel à modifier à partir de ses paramètres.</p> <p> Reçus : demande de spécifier les changements et validation.</p>
<i>Activer compte</i>	<ul style="list-style-type: none"> ▪ Administrateur 	<p> Emis: activer compte.</p> <p> Reçus : choisir le compte du personnel à activer à partir de ses paramètres.</p>
<i>Désactiver compte</i>	<ul style="list-style-type: none"> ▪ Administrateur 	<p> Emis: désactiver compte.</p> <p> Reçus : choisir le compte du personnel à désactiver à partir de ses paramètres.</p>
<i>Supprimer compte</i>	<ul style="list-style-type: none"> ▪ Administrateur 	<p> Emis: choisir le personnel à supprimer à partir de ses paramètres.</p>
<i>Ajouter service</i>	<ul style="list-style-type: none"> ▪ Administrateur 	<p> Emis: ajouter le nom et la description du service.</p> <p> Reçus : validation.</p>
<i>Modifier service</i>	<ul style="list-style-type: none"> ▪ Administrateur 	<p> Emis: choisir le service à modifier.</p> <p> Reçus : demande de spécifier les changements et validation.</p>
<i>Supprimer service</i>	<ul style="list-style-type: none"> ▪ Administrateur 	<p> Emis: choisir le service à supprimer.</p> <p> Reçus : Confirmation.</p>
<i>Ajouter agenda</i>	<ul style="list-style-type: none"> ▪ Administrateur 	<p> Emis: ajouter un agenda.</p> <p> Reçus : validation.</p>

<i>Modifier agenda</i>	<ul style="list-style-type: none"> ▪ Administrateur 	<p> Emis: choisir l'agenda à modifier.</p> <p> Reçus :demande de spécifier les changements et validation.</p>
<i>Supprimer agenda</i>	<ul style="list-style-type: none"> ▪ Administrateur 	<p> Emis: choisir l'agenda à supprimer.</p>
<i>Authentification</i>	<ul style="list-style-type: none"> ▪ Administrateur ▪ Médecin ▪ Major ▪ Secrétaire 	<p> Emis: authentification et accès au compte.</p> <p> Reçus :demande d'authentification et connexion.</p>

3. Analyse et conception

Cette étape consiste à formaliser et à détailler les besoins exprimés lors de l'étude préliminaire, celle-ci sera réalisée principalement à l'aide des cas d'utilisation qui permettent de capturer la fonctionnalité du système au point de vue utilisateur.

3.1. Diagramme de package

C'est un moyen pour regrouper les différents éléments de la modélisation.

Il permet de représenter les relations entre les différents profils de l'application.

Il rassemble les cas d'utilisations propre à chaque acteur de façon cohérente.

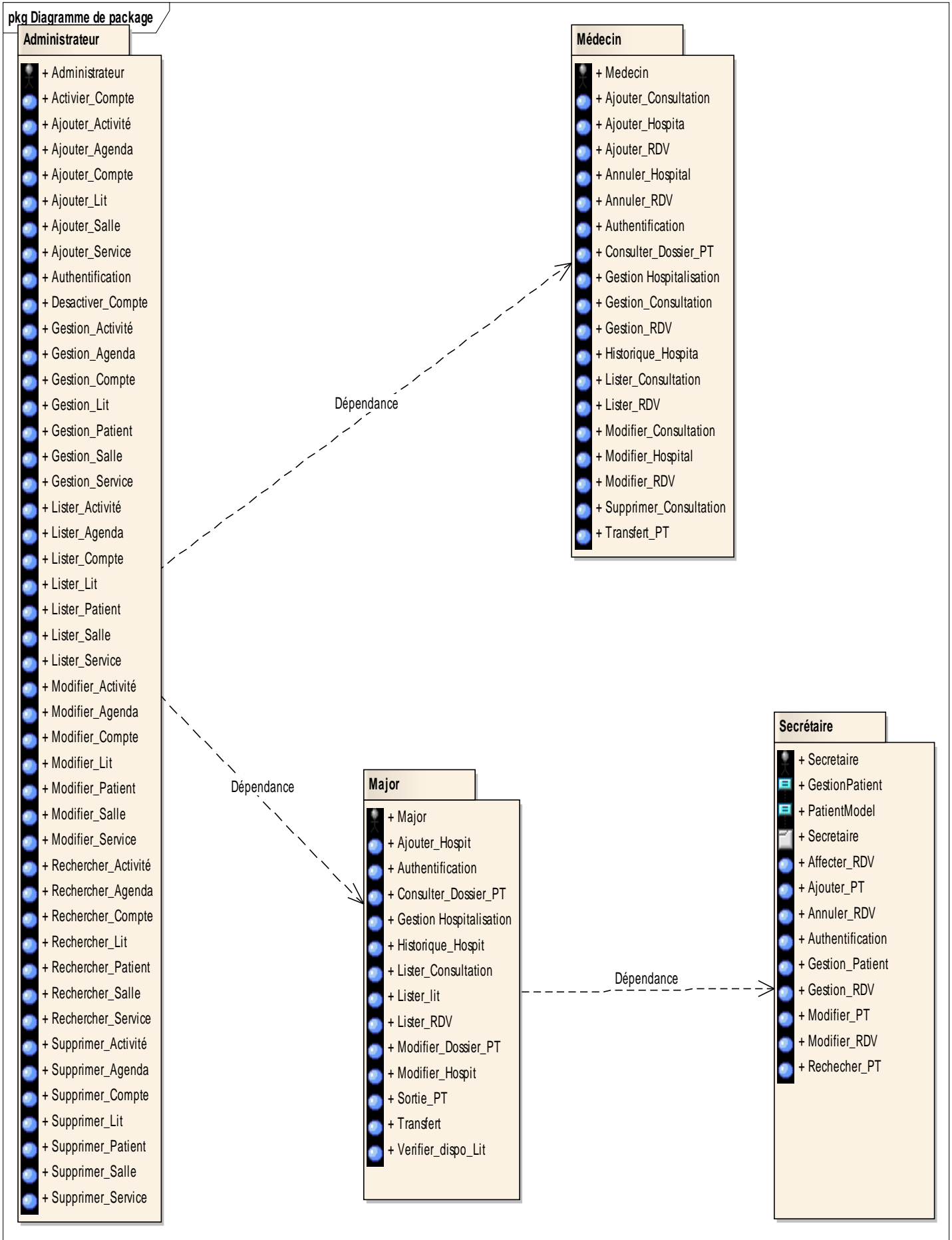


Figure 6: Diagramme de package par acteur

3.2. Diagrammes des cas d'utilisation

Il modélise un service rendu par le système utilisé afin de donner une vision globale du comportement fonctionnel d'un système logiciel.

Il représente une séquence d'actions réalisée par le système.

Secrétaire :

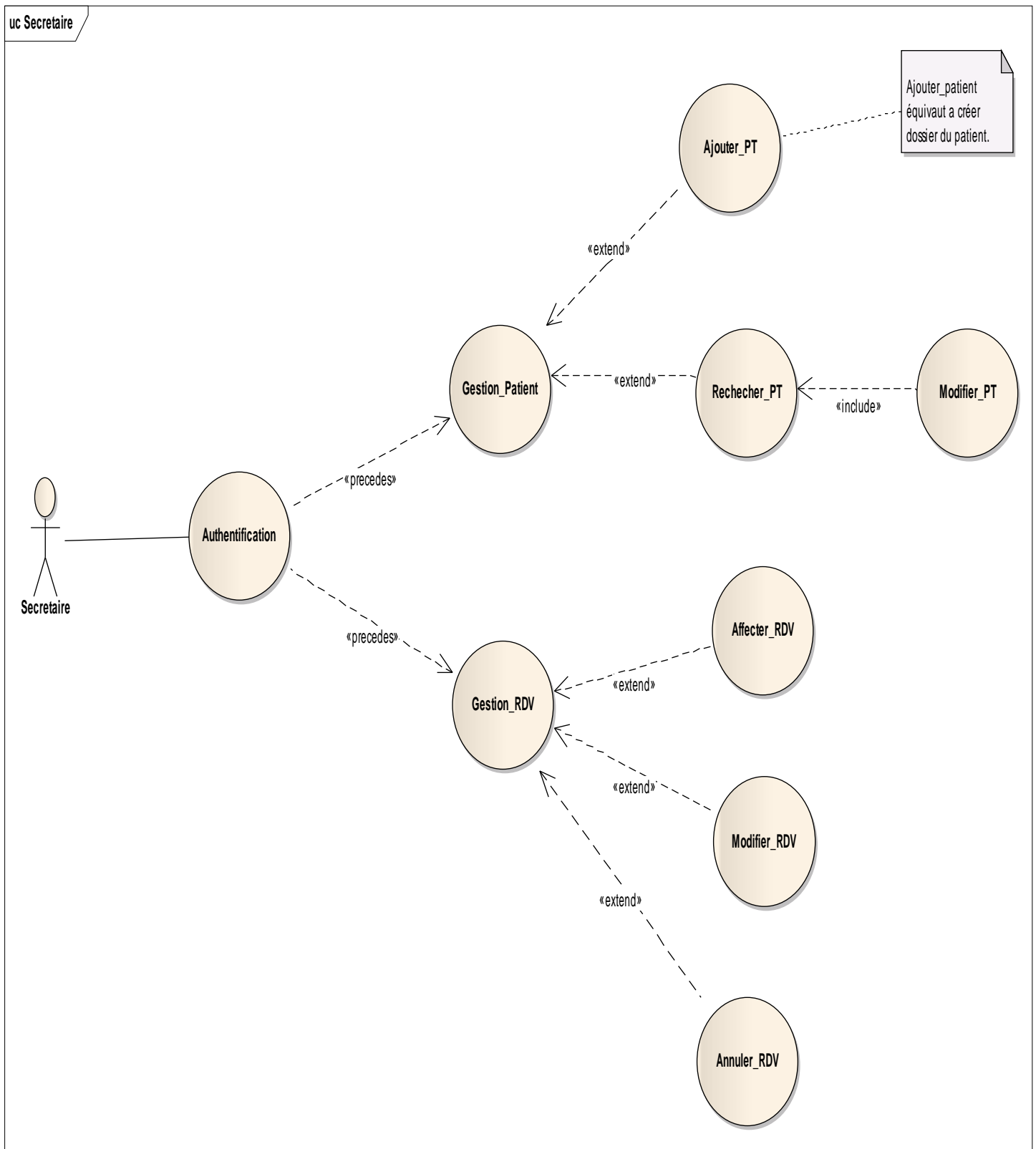


Figure 7: Diagramme des cas d'utilisation pour Secrétaire

Major :

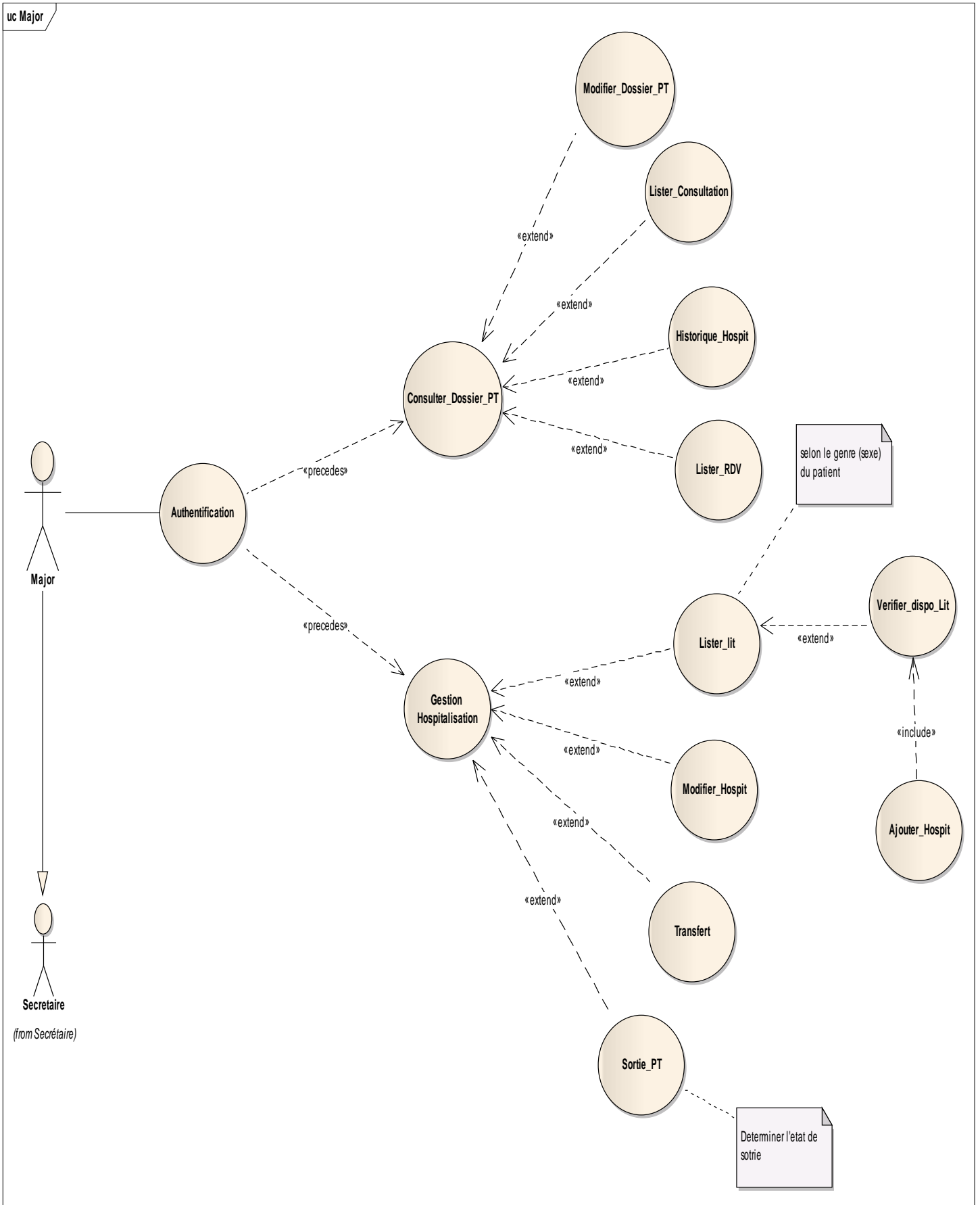


Figure 8: Diagramme des cas d'utilisation pour Major

Médecin :

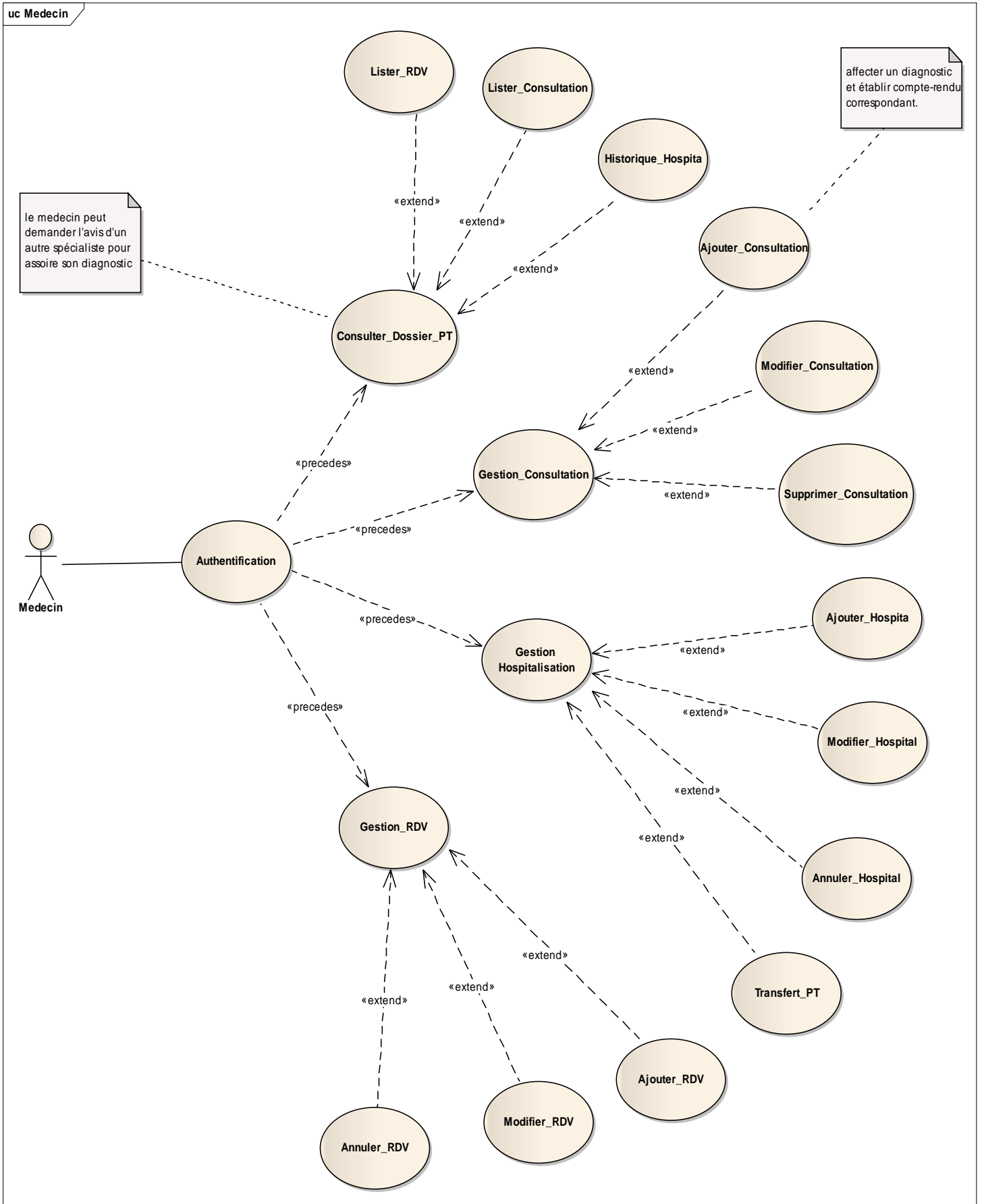


Figure 9: Diagramme des cas d'utilisation pour Médecin

Administrateur :

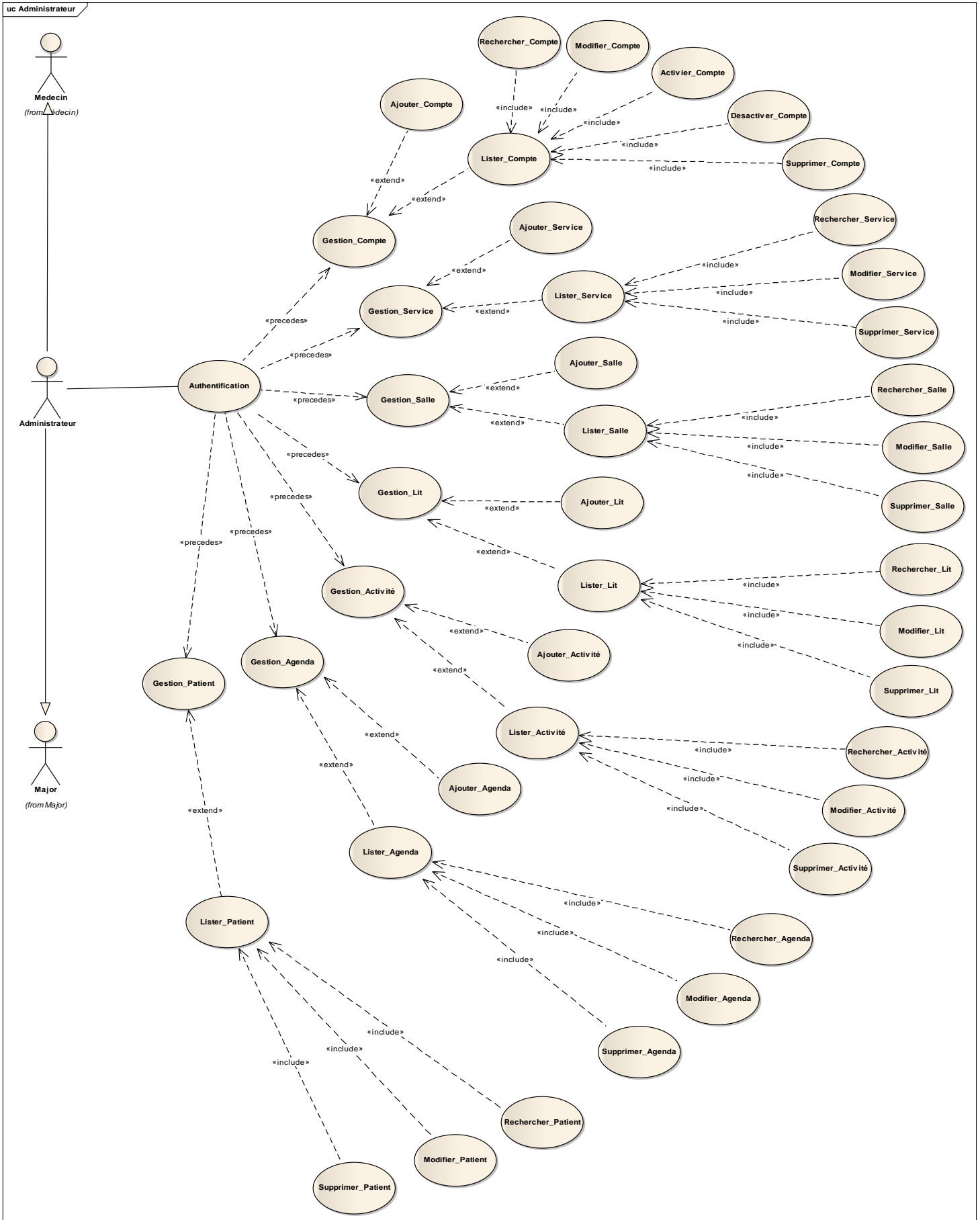


Figure 10: Diagramme des cas d'utilisation pour Administrateur

3.3. Diagrammes de séquences

Le diagramme de séquence permet d'illustrer les cas d'utilisation et de représenter les interactions dans le temps entre les objets du système.

Authentification :

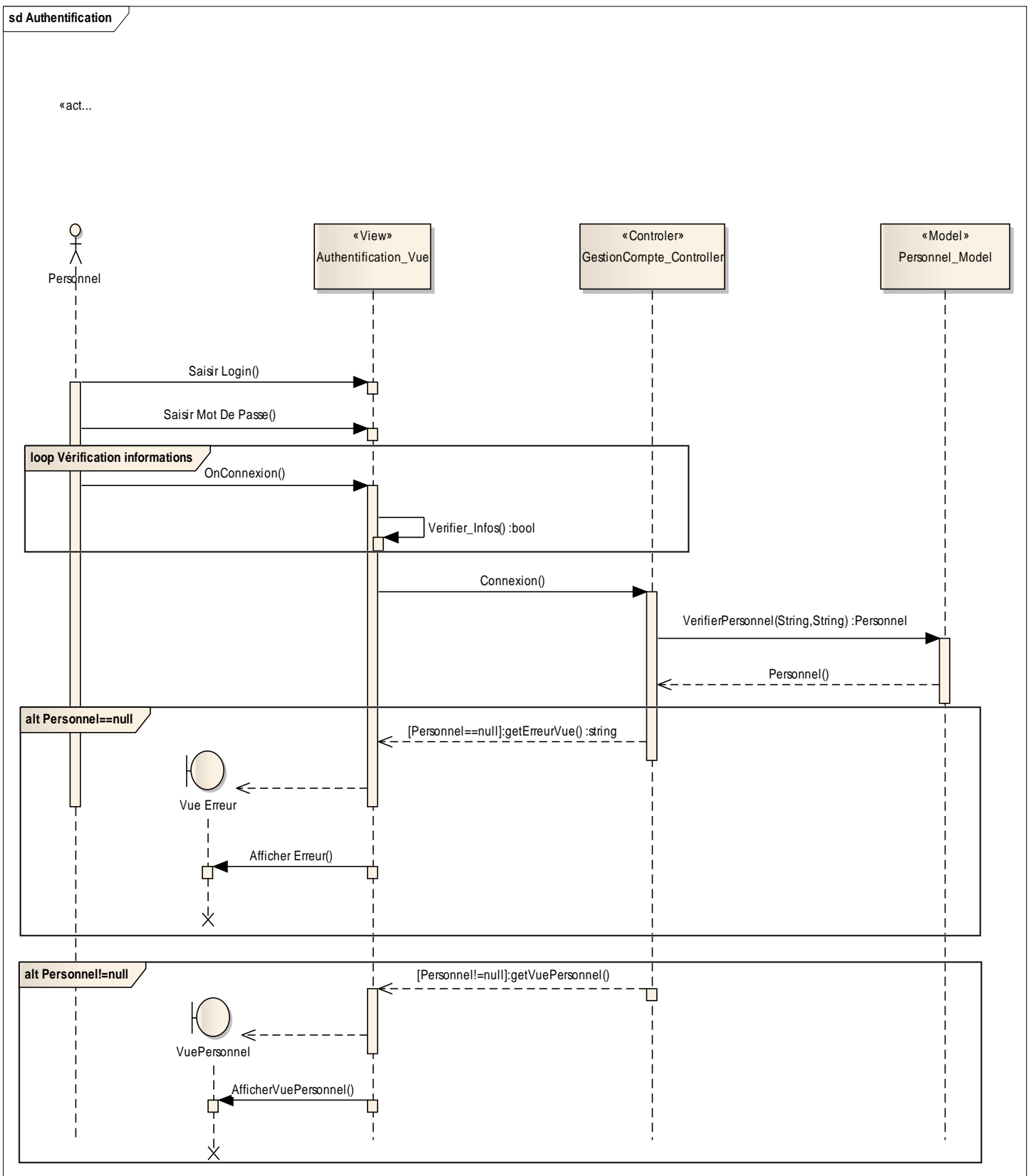


Figure 11: Diagramme de séquences d'authentification

Ajouter patient :

A l'exception du médecin tous les acteurs peuvent ajouter un patient.

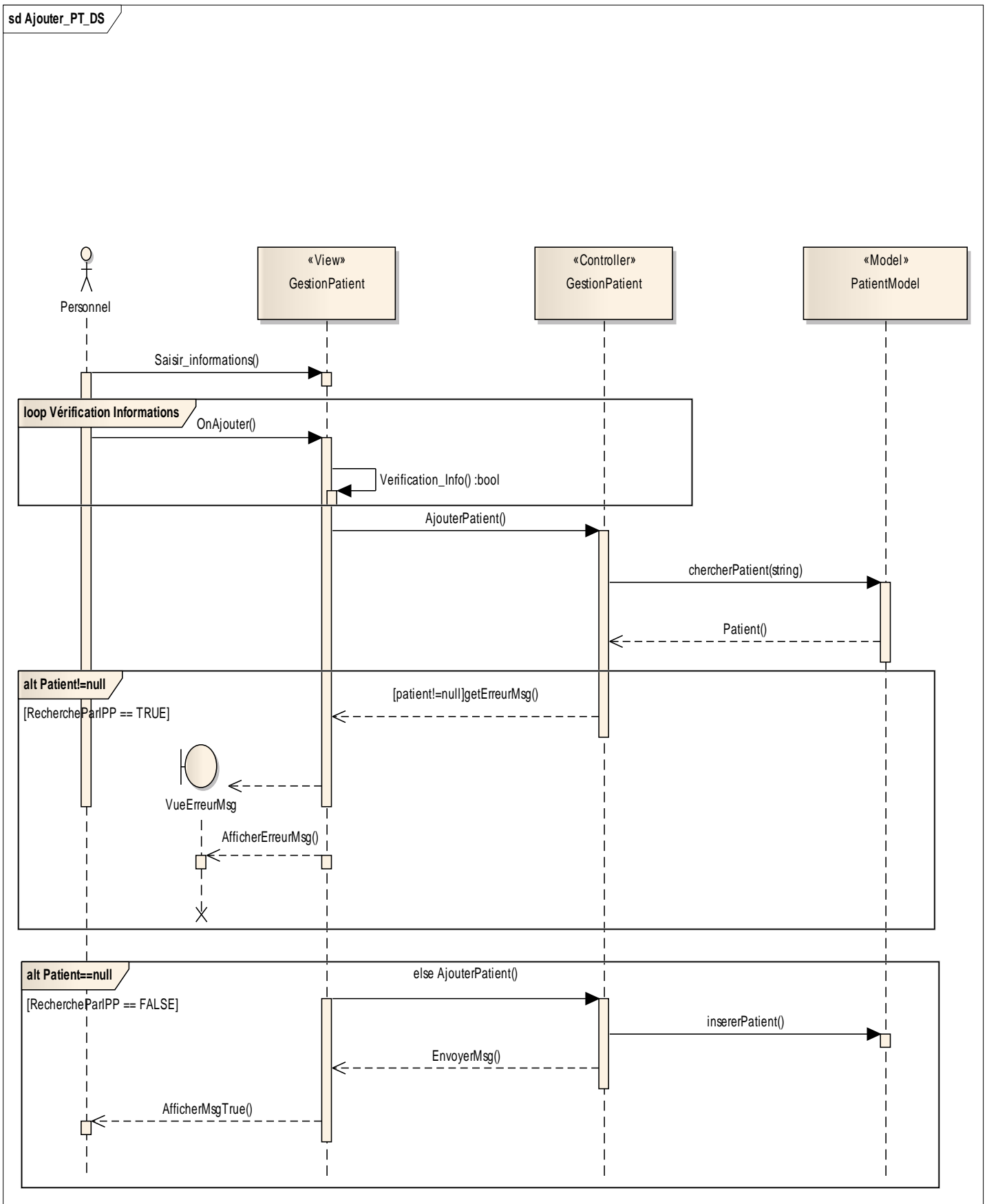


Figure 12: Diagramme de séquences d'ajout patient

Rechercher et Modifier patient :

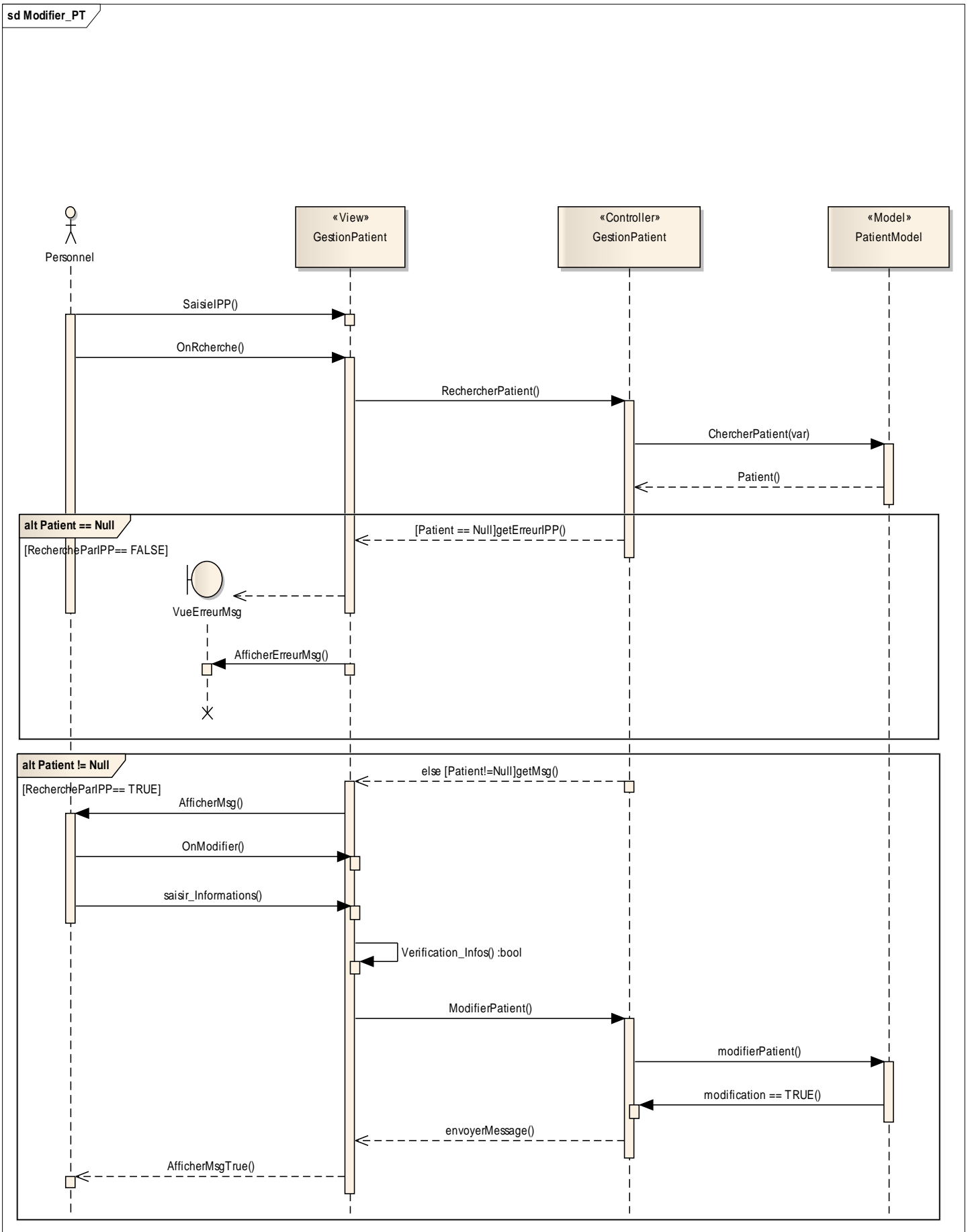


Figure 13: Diagramme de séquences recherche et modification patient

Ajouter Compte :

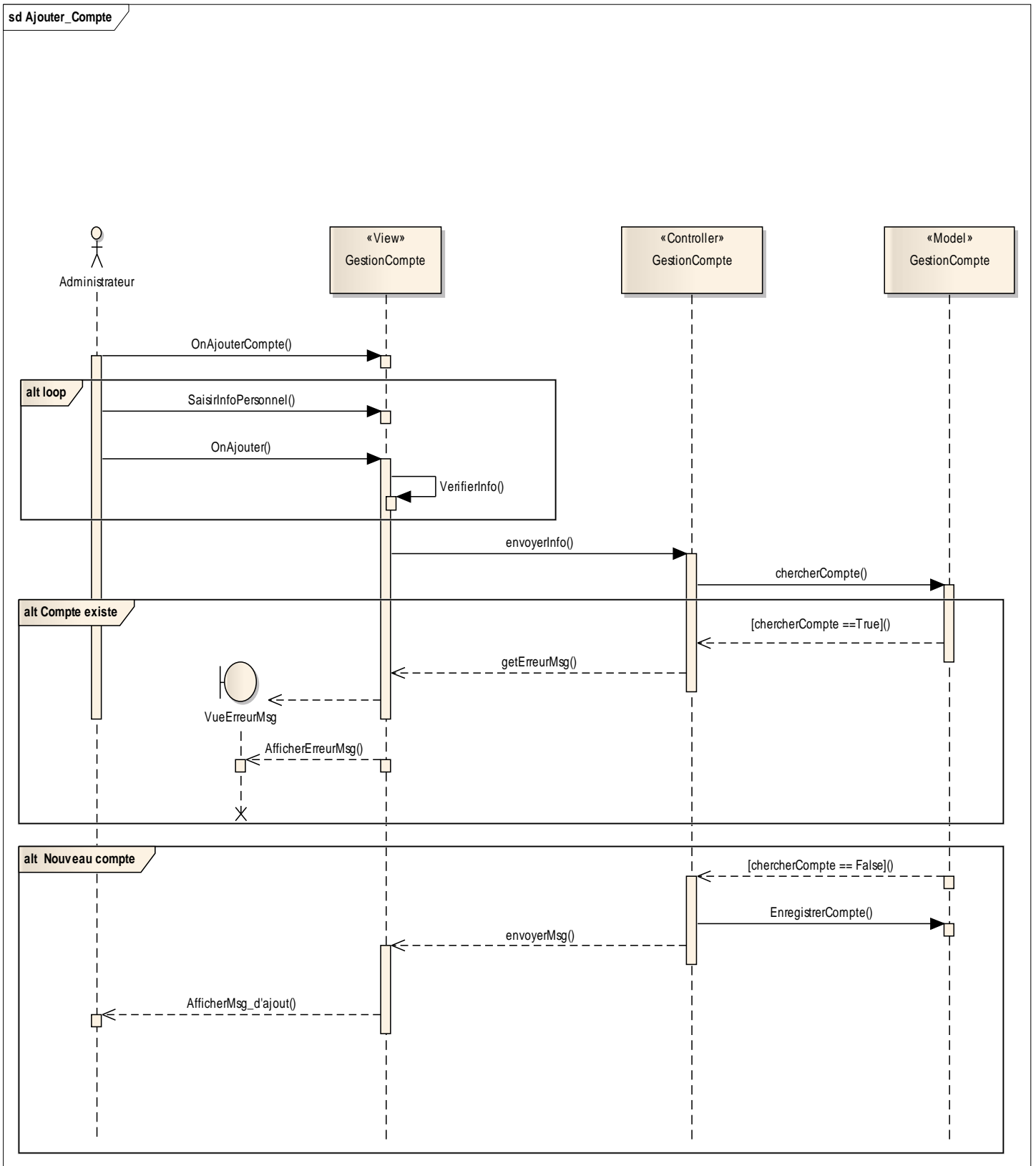


Figure 14: Diagramme de séquences d'ajout compte

3.4. Diagramme de classes et sa description

C'est le point central dans le développement orienté objet. Il représente la structure statique du système sous forme de classes et de leurs relations.

Les classes constituent la base pour la génération de code et des schémas de bases de données.

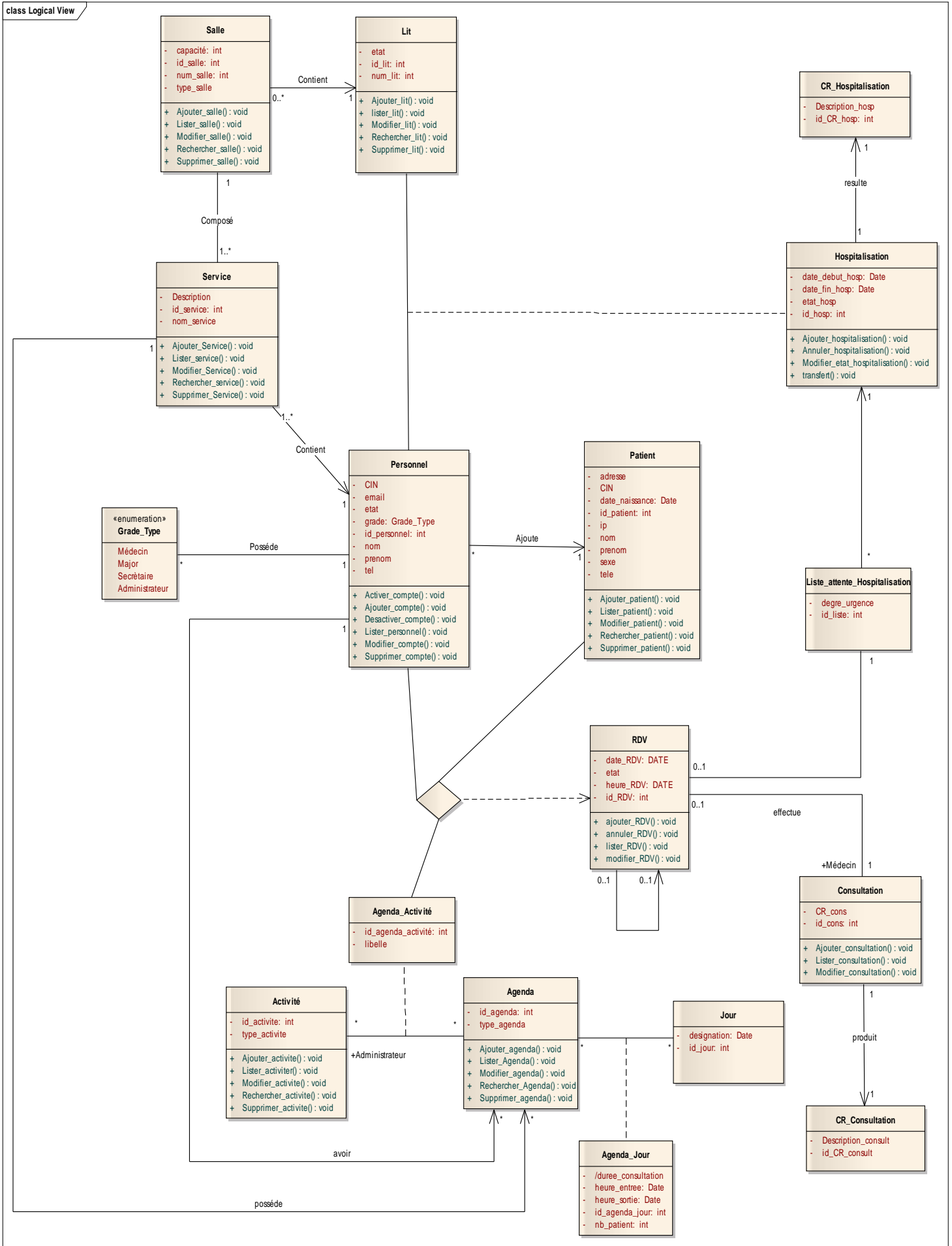


Figure 15: Diagramme de classes

Description du diagramme des classes

L'application utilise 4 acteurs qui sont regroupés sous forme d'une seule classe « Personnel », chacun possède plusieurs tâches avec une tâche principale. C'est ainsi que le médecin s'occupe principalement du patient, le major assiste en premier lieu le médecin dans ses décisions, tandis que la secrétaire affecte le premier RDV et oriente le patient. Finalement l'administrateur effectue toutes les tâches inhérentes aux autres acteurs.

- Chaque personnel (classe « Personnel ») possède un grade.
- Chaque acteur utilise une fenêtre dédiée à lui.
- Chaque personnel utilise un compte propre à lui.
- La gestion des comptes du personnel est gérée par l'administrateur.
- Chaque acteur travaille dans un service.
- Chaque service se compose au moins d'une salle.
- Chaque salle est caractérisée par un numéro, une capacité litière et spécifique à un genre déterminé (masculin ou féminin).
- L'administrateur ou le major peuvent attribuer un lit pour le patient hospitalisé.
- Au fur et à mesure de chaque hospitalisation le médecin établit un compte rendu.
- Chaque service possède un agenda.
- Le médecin possède un agenda.
- La relation entre les classes «Activite» «Agenda» est de cardinalité (n,n) convergent vers une classe association «Agenda-Activite».
- Chaque personnel consulte l'agenda et l'activité pour attribuer un RDV au patient.
- En cas d'indisponibilité de lits, le major ou l'administrateur inscrivent le patient dans la liste d'attente.
- Chaque patient peut être suivi par plusieurs RDV.
- Les classes «Agenda» «Jour» convergent vers une classe association «Agenda-Jour».
- Le RDV est inscrit dans un agenda au cours d'un jour précis.
- Le médecin examine le patient au moment du RDV et peut lui fixer un autre en cas de besoin.
- Au terme de chaque consultation, le médecin élabore un compte rendu.

3.5. La base de données

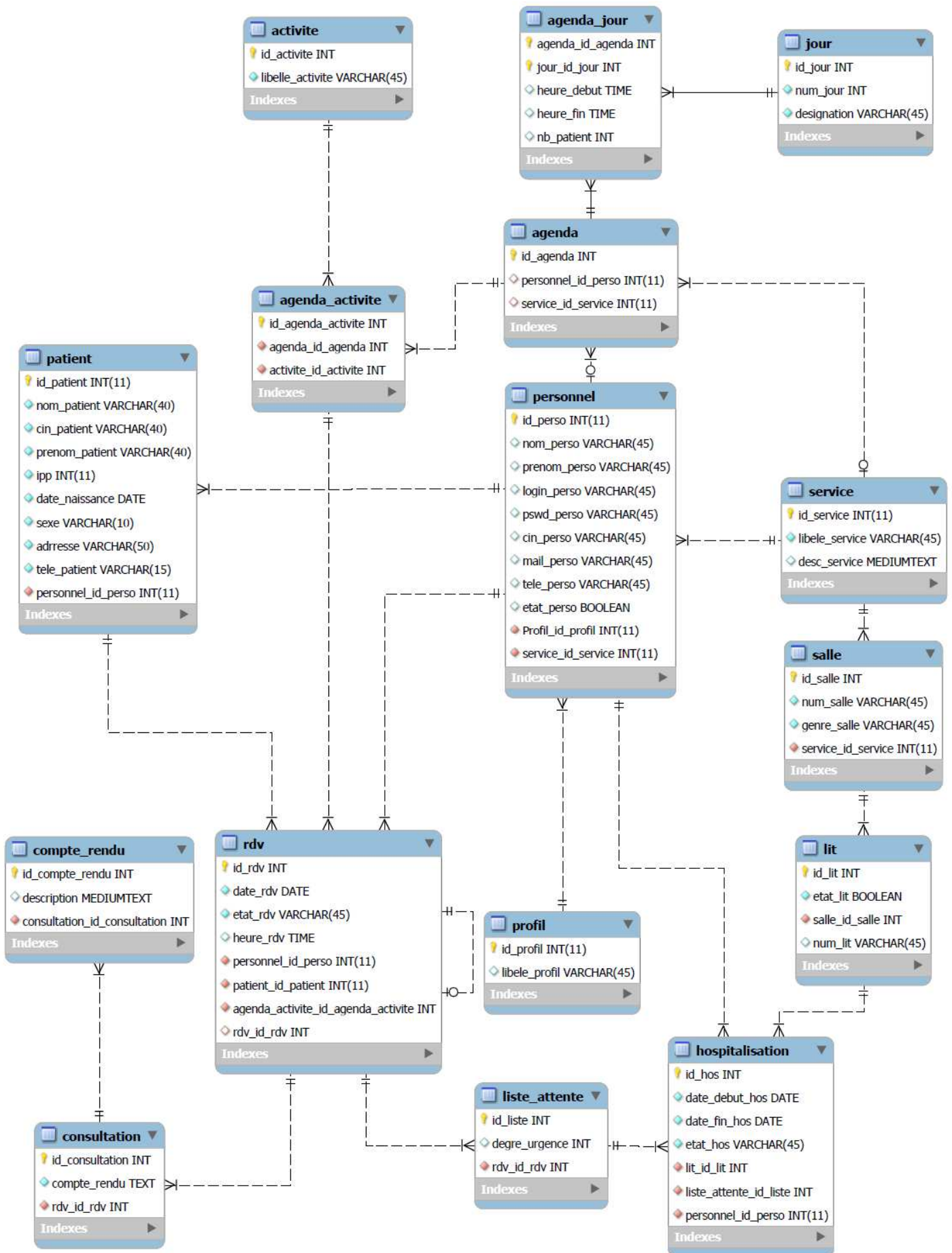
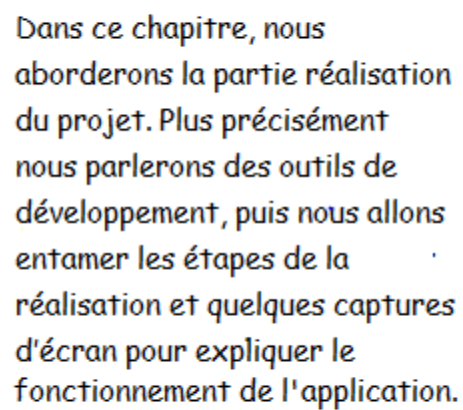


Figure 16: Les tables de la base de données

Chapitre III

Présentation de l'application



Dans ce chapitre, nous aborderons la partie réalisation du projet. Plus précisément nous parlerons des outils de développement, puis nous allons entamer les étapes de la réalisation et quelques captures d'écran pour expliquer le fonctionnement de l'application.

1. Outils et technologies de développement



Enterprise Architect est un outil d'analyse de création UML, couvrant le développement du logiciels de rassemblement d'exigences, en passant par les étapes d'analyse, les modèles de conception et les étapes de test et d'entretien.

Cet outil permet de bien schématiser notre application, pour passer de la conception vers la réalisation. Il facilite la représentation des diagrammes UML tels que le diagramme des cas d'utilisation, des séquences et des classes.

L'architecte d'entreprise est un outil conçu pour établir un logiciel facile à mettre à jour.

Il possède un outil de production de documentation souple et de haute qualité.



WAMP Server est une plateforme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de script PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement les bases de données.



PHPMyAdmin est une application Web de gestion pour des systèmes de gestion de base de données MySQL réalisée en PHP.



Apache [4] est un serveur http crée et maintenu au sein de la fondation Apache. C'est le serveur http populaire du World Wide Web. Il est distribué selon les termes de la licence Apache.



Notepad++ est un programme spécialement conçu pour l'édition de code source. Il est compatible avec plusieurs langages de programmation



Aptana Studio [5] est un environnement de développement intégré orienté web, multiplateforme est open-source. Il facilite l'écriture du code en fournissant des aides à la saisie pour le JavaScript, l'HTML, les CSS, PHP et Python.

Aptana est disponible en version autonome ou bien en plugin pour son environnement d'origine Eclipse.



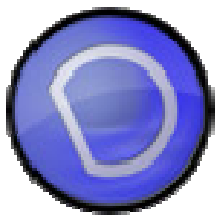
MySQL Workbench[6] est un logiciel de gestion et d'administration de bases de données MySQL créé en 2004. Via une interface graphique intuitive, il permet entre autres de créer, modifier ou supprimer des tables, des comptes utilisateurs, et d'effectuer toutes les opérations inhérentes à la gestion d'une base de données. Pour ce faire, il doit être connecté à un serveur MySQL.



HypertextPreprocessor[7], plus connu sous son sigle **PHP** est un langage de programmation libre principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté-objet.



Bootstrap [8] est une collection d'outils utile à la création de sites web et applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub (**GitHub** est un service web d'hébergement et de gestion de développement de logiciels).



DataTable est un plug-in pour la bibliothèque jQuery. C'est un outil très flexible, basé sur les principes de l'amélioration progressive, et les contrôles d'interactions avancées pour une table HTML. Lorsque vous initialiser l'objet jQuery.dataTable, informations sur la table sont lues directement sur la page HTML. Parmi ses avantages, c'est un logiciel libre (open source), facilite la pagination, utilise une recherche multicritère, supporte plusieurs sources de données : DOM, JavaScript, Ajax et server-side processing.



JS est un langage de programmation de scripts principalement utilisé dans les pages web interactives mais aussi côté serveur. C'est un langage orienté objet à prototype.



jQuery[9] est une bibliothèque JavaScript libre qui porte sur l'interaction entre JavaScript (comprenant Ajax) et HTML, et a pour but de simplifier des commandes communes de JavaScript. La première version date de janvier 2006.



Ajax (Asynchronous JavaScript and XML) est une architecture informatique basée essentiellement sur le JavaScript, permet d'accéder de manière asynchrone avec les actions de l'utilisateur à la base de données et ce en utilisant en plus du JavaScript, une classe XMLHttpRequest, qui comporte des méthodes permettant de communiquer avec le serveur, offrant ainsi à l'utilisateur une réponse rapide et instantanée.



CodeIgniter[10] :

CodeIgniter est un framework libre écrit en PHP. IL suit le motif de conception MVC. La documentation de CodeIgniter est complète. La communauté du framework est très active ce qui permet de trouver de l'aide très rapidement. De plus, les membres de la communauté de CodeIgniter ont développé de nombreuses bibliothèques réutilisables.

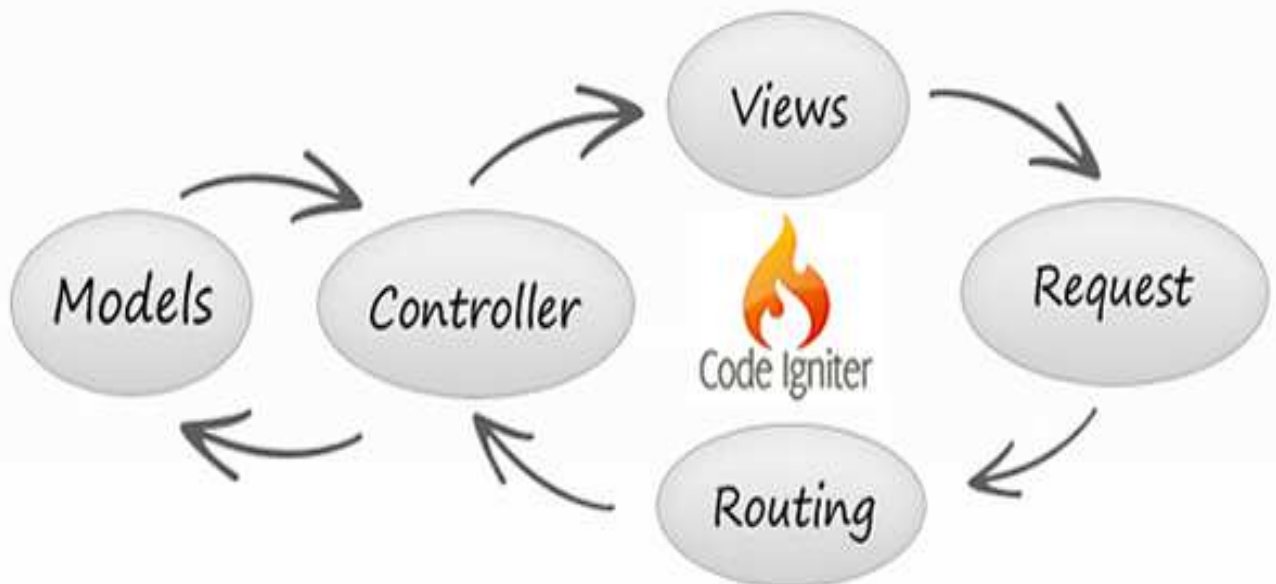


Figure 17: Architecture du framework CodeIgniter

CodeIgniter encourage fortement l'utilisation de l'architecture Modèle-Vue-Contrôleur. Le framework est compatible avec PHP 5 à partir de la version 2.0.0

Caractéristiques de CodeIgniter :

- Bibliothèques complètes de gestion de base de données avec support de plusieurs plateformes.
- Validation des données et des formulaires.
- Gestion des sessions.
- Classe d'upload de fichiers.
- Support de l'Active record (patron de conception).
- Bibliothèque de manipulation des images (redimensionnement, rognage, rotation, etc) avec GD (GifDraw), ImageMagik.
- Classes d'envoi de mails supportant les pièces jointes, le format HTML ou texte, plusieurs protocoles(Sendmail, SMTP, mail, etc.) et plus.

- Classe calendrier.
- Classe User Agent : permet de fournir des fonctions qui aident à identifier les informations sur le navigateur.
- Classe de compression ZIP: permet de créer des archives Zip.
- Cryptage des données.
- Importantes bibliothèques de fonctions d'aide (helper).
- Ainsi que d'autres caractéristiques.

Justification du choix deCodeIgniter

Le frameworkCodeIgniter a été choisi parmi plusieurs framework .Ce choix est justifié par le faitqu'il est plus adapté à notre application et qu'il permet de:

- Développer des projets plus rapides.
- Posséder une documentation importante.
- Organiser le travail.
- Etre un environnement cadre de développement de l'application qui possède un ensemble d'outils permettant de structurer et de construire des applications en utilisant PHP.

Tableau N° 3 : table de matière du CodeIgniter

<ul style="list-style-type: none"> User Guide Home Table of Contents Page Basic Info <ul style="list-style-type: none"> Server Requirements License Agreement Change Log Credits Installation <ul style="list-style-type: none"> Downloading CodeIgniter Installation Instructions Upgrading from a Previous Version Troubleshooting Introduction <ul style="list-style-type: none"> Getting Started CodeIgniter at a Glance CodeIgniter Cheatsheets Supported Features Application Flow Chart Model-View-Controller Architectural Goals Tutorial <ul style="list-style-type: none"> Introduction Static pages News section Create news items Conclusion 	<ul style="list-style-type: none"> General Topics <ul style="list-style-type: none"> CodeIgniter URLs Controllers Reserved Names Views Models Helpers Using CodeIgniter Libraries Creating Your Own Libraries Using CodeIgniter Drivers Creating Your Own Drivers Creating Core Classes Hooks - Extending the Core Auto-loading Resources Common Functions URI Routing Error Handling Caching Profiling Your Application Running via the CLI Managing Applications Handling Multiple Environments Alternative PHP Syntax Security PHP Style Guide Writing Documentation Additional Resources <ul style="list-style-type: none"> Community Forums Community Wiki 	<ul style="list-style-type: none"> Class Reference <ul style="list-style-type: none"> Benchmarking Class Calendar Class Cart Class Config Class Email Class Encryption Class File Uploading Class Form Validation Class FTP Class HTML Table Class Image Manipulation Class Input Class Javascript Class Loader Class Language Class Migration Class Output Class Pagination Class Security Class Session Class Trackback Class Template Parser Class Typography Class Unit Testing Class URI Class User Agent Class XML-RPC Class Zip Encoding Class 	<ul style="list-style-type: none"> Driver Reference <ul style="list-style-type: none"> Caching Class Database Class Javascript Class Helper Reference <ul style="list-style-type: none"> Array Helper CAPTCHA Helper Cookie Helper Date Helper Directory Helper Download Helper Email Helper File Helper Form Helper HTML Helper Inflector Helper Language Helper Number Helper Path Helper Security Helper Smiley Helper String Helper Text Helper Typography Helper URL Helper XML Helper
--	--	--	---

TABLE OF CONTENTS

Structure du CodeIgniter

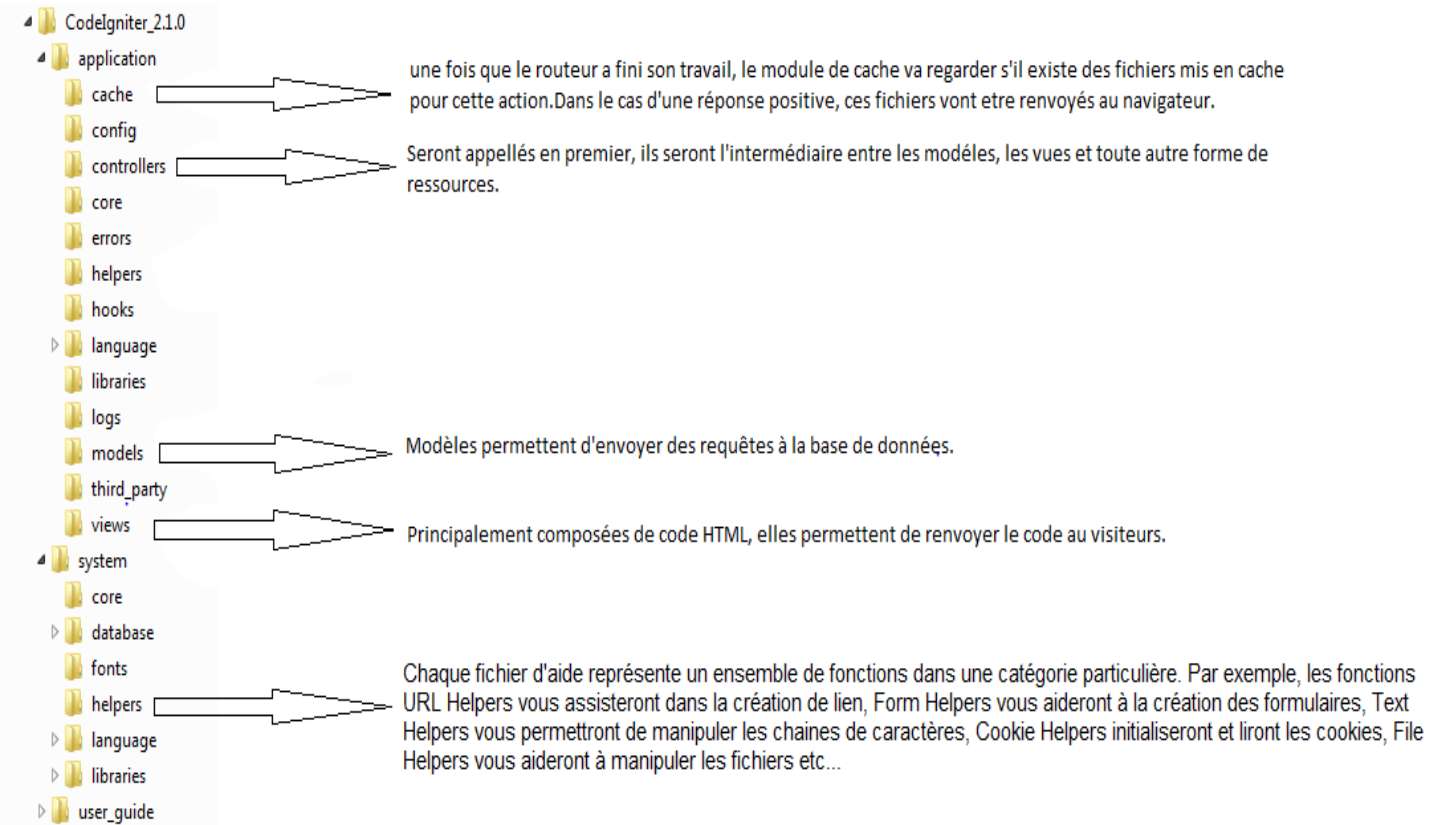


Figure 18: Structure du CodeIgniter

2. Schéma général de l'application

Le schéma suivant présente l'architecture générale de l'application «Réalisation d'une application pour la gestion des rendez-vous et des hospitalisations »

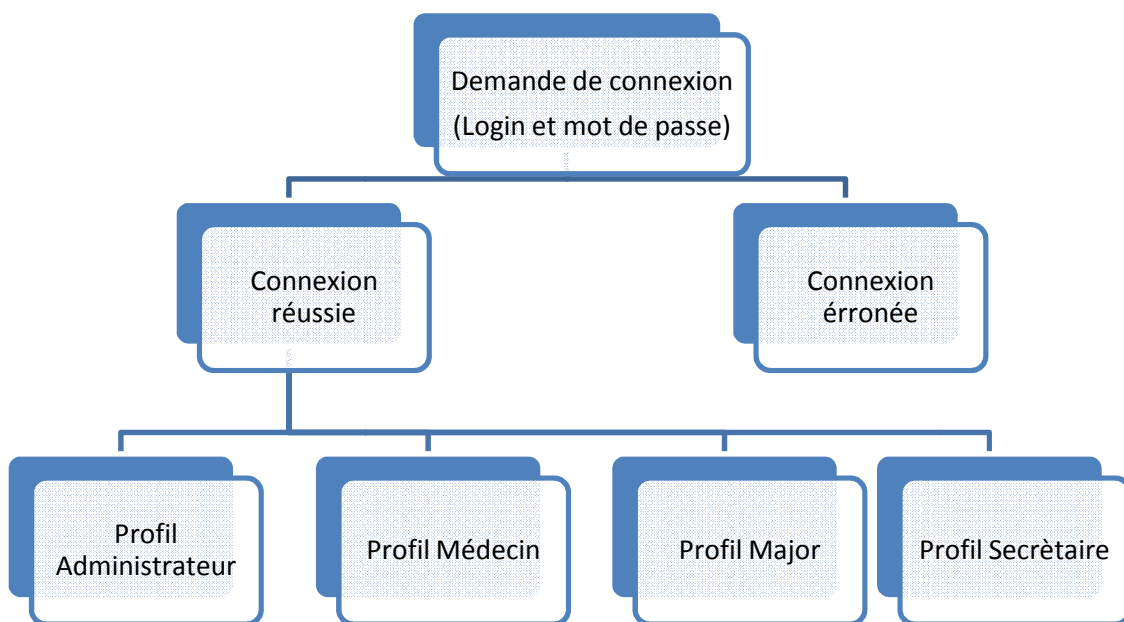


Figure 19: Schéma général de l'application

3. Présentation de l'application

➤ Présentation de la phase authentification

Authentification



The screenshot shows a light gray rectangular box containing the authentication form. At the top left, it says "Login :" followed by a white rectangular input field. Below that, it says "Mot de passe :" followed by another white rectangular input field. At the bottom center, there is a blue button with the white text "connexion".

Figure 20: Fenêtre d'Authentification

Cette page permet de s'authentifier et de faire une redirection vers la vue associée à l'acteur. Si le login ou le mot de passe est incorrect l'application va demander à l'utilisateur de s'authentifier à nouveau en affichant le message d'erreur suivant :

Alert ! Combinaison login/mot de passe incorrecte!(Il se peut aussi que ce compte soit désactivé).

Figure 21: Message d'une fausse authentification

➤ Menu d'administrateur

La page d'accueil permet à l'administrateur d'accéder à ces principales fonctions.



 Accueil	
 Gestion des Comptes	<
 Gestion des Services	<
 Gestion des Activites	<
 Gestion des Agendas	<
 Gestion des Patients	<

Figure 22: Menu d'administrateur

La recherche se fait par plusieurs critères soit par l'un des attributs ou bien par la combinaison des deux attributs.

Selon le choix de l'administrateur, il peut choisir le nombre de lignes à afficher par page.

Gestion des comptes










Figure 23: Gestion des comptes

- Lister personnel :

L'administrateur a le droit d'activer ou désactiver un compte d'un personnel selon le besoin (retraite, mutation, changement de poste..). Il n'a pas le pouvoir de supprimer ou désactiver son propre compte.

Liste des personnels

Afficher éléments Q Rechercher :

action	Nom	Prenom	Login	CIN	E-mail	Tele	Profil	Service
  	admin	admino	admin	AD098765	admin@admin.com	0687345423	admin	radiologie
  	harti	naoufal	medecin	CD876765	med@chu.ma	0698678453	med	radiologie
  	mhamdi	hajar	hajar	CD765678	mhamdihajar@gmail.com	0634764584	med	radiologie
  	prof	professeur	prof	DF123456	admin@gmail.com	0654172679	med	radiologie
  	secretaire	secretary	secetaire	CD67543	secetaire@gmail.com	0567423146	sec	radiologie

Affichage de l'élément 1 à 5 sur 8 éléments Précédent **1** 2 Suivant

Figure 24: Liste des personnels

Gestion des services

Pour accéder à la gestion des salles ou des lits, l'administrateur doit cliquer sur la case gestion des services.

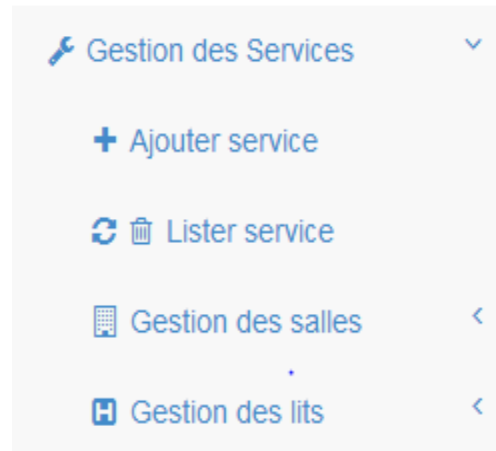


Figure 25: Gestion des services

- Ajouter un service :



Informations sur le nouveau service:

Nom du service :

Déscription :

Ajouter Vider

Figure 26: page d'ajout un nouveau service















- Lister les services :

L'administrateur a le droit de modifier ou supprimer un service.

Liste des services

Afficher 10 éléments

Q Rechercher :

action	Nom du service	Description
 	Pneumologie	la branche de la médecine s'intéressant au pathologies respiratoire
 	Cardiologie	spécialité médicale qui étudie le cœur
 	Neurologie	spécialité médicale qui étudie les maladies du système nerveux
 	Psychiatrie	spécialité médicale des maladies mentales
 	Trammatologie	étude médicale des traumatismes physiques
 	Neuropsychiatrie	médecine qui prend en charge les troubles mentaux et les maladies neurologiques
 	Chirurgie cardiaque	spécialité chirurgicale qui traite les affectations du cœur et des vaisseaux

Affichage de l'élément 11 à 17 sur 17 éléments

Précédent 1 2 Suivant

Figure 27: Liste des services

- Ajouter un lit:

La salle et le numéro de lit n'apparaissent que lorsqu'on choisit le service où on veut ajouter un nouveau lit.

Informations sur le nouveau lit :

Service :

Salle :

Numéro du lit :

Figure 28: page d'ajout un nouveau lit

- Le listage des salles, lits, agendas, activités et patients se fait de la même manière que le listage des services.

Gestion des agendas

- Ajouter un agenda :
Pour ajouter un agenda nous devons ajouter le type d'agenda et l'heure de début, fin.

Informations de l'agenda :

Type agenda :

Professeur :

Jours :

num	Jour	Heure de début	Heure de fin	Nombre Max de patients	Tous
2	Lundi	09:00	08:30	<input type="text" value="2"/>	<input checked="" type="checkbox"/>
3	Mardi	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>
4	Mercredi	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>
5	Jeudi	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>
6	Vendredi	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>
7	Samedi	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>
1	Dimanche	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>

Figure 29: Page d'ajout un nouveau agenda

Si l'heure de début est supérieure à l'heure de fin un message d'erreur sera affiché comme suit :

Le champ **Heure de fin (Lundi)** doit contenir une heure supérieur à 09:00.

Figure 30: Message d'erreur

Gestion des rendez-vous

Choisir la date du Rendez-Vous:

← Juin 2014 →

Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

Nouveau RDV
Détails des RDVs

IPP :

Date :

Activite :

valider
Annuler

Figure 31: Fenêtre de gestion des rendez-vous

Conclusion et perspectives

Au cours de la période du stage de fin d'études, on a eu l'opportunité de mettre en œuvre différentes connaissances acquises durant nos études à la faculté des sciences et techniques de Fès et acquérir de nouveaux outils de développement tels que l'architecture MVC, Ajax, PHP (orienté objet) et JQuery.

Notre travail s'est fixé comme objectifs de satisfaire le maximum des besoins du cahier de charge et faciliter les tâches à l'administrateur et aux utilisateurs.

Le projet se présente en trois parties. La première partie s'est intéressée au lieu de stage et à la problématique, la deuxième partie à la méthodologie de l'analyse et à la conception UML, la troisième partie aux technologies et outils utilisés et à la présentation de l'application.

Les difficultés majeures qu'on a rencontrées résident essentiellement dans la nouveauté des technologies avec lesquelles on a travaillé et la contrainte du temps pour les maîtriser.

Parmi les intérêts de ce projet on cite entre autres l'organisation du travail du personnel, la bonne gestion des lits et des rendez-vous et le gain de temps dans la recherche.

Comme perspectives on peut envisager par la suite la création d'un système de communication interne, l'implantation d'un système d'alerte, la mise en place d'une gestion de transfert et enfin la gestion hebdomadaire des agendas.

Bibliographie et Webographie

1. Webographie

- [1][http://fr.wikipedia.org/wiki/UML_\(informatique\)](http://fr.wikipedia.org/wiki/UML_(informatique))
- [3]<http://fr.wikipedia.org/wiki/Modèle-vue-contrôleur>
- [4]http://fr.wikipedia.org/wiki/Apache_HTTP_Server
- [5] <http://fr.wikipedia.org/wiki/Aptana>
- [6] http://fr.wikipedia.org/wiki/MySQL_Workbench
- [7] <http://fr.wikipedia.org/wiki/PHP>
- [8] http://fr.wikipedia.org/wiki/Twitter_Bootstrap
- [9] <http://fr.wikipedia.org/wiki/JQuery>
- [10] <http://fr.wikipedia.org/wiki/CodeIgniter>

getbootstrap.com

ellislab.com/codeigniter

stackoverflow.com

datatables.net

jquery.com

api.jquery.com/jquery.ajax/

2. Bibliographie

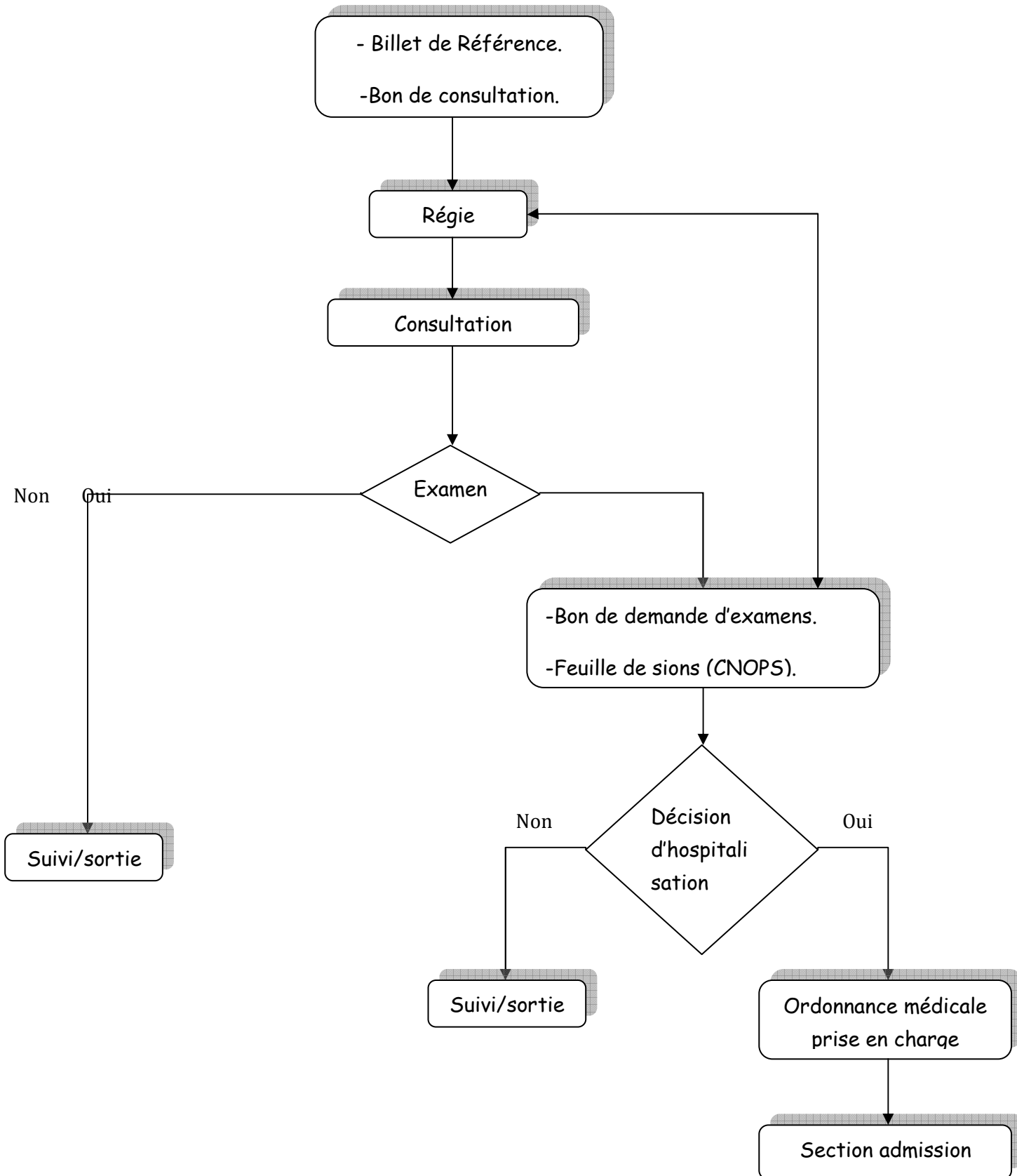
[2] Cours Ilham Chakir Génie logiciel chapitre2 le processus logiciel page 78

Cours Techniques web du Pr. BEGDOURI Ahlame (2013/2014)

Cours UML de Mr. BENNABOU Abderrahim (2013-2014)

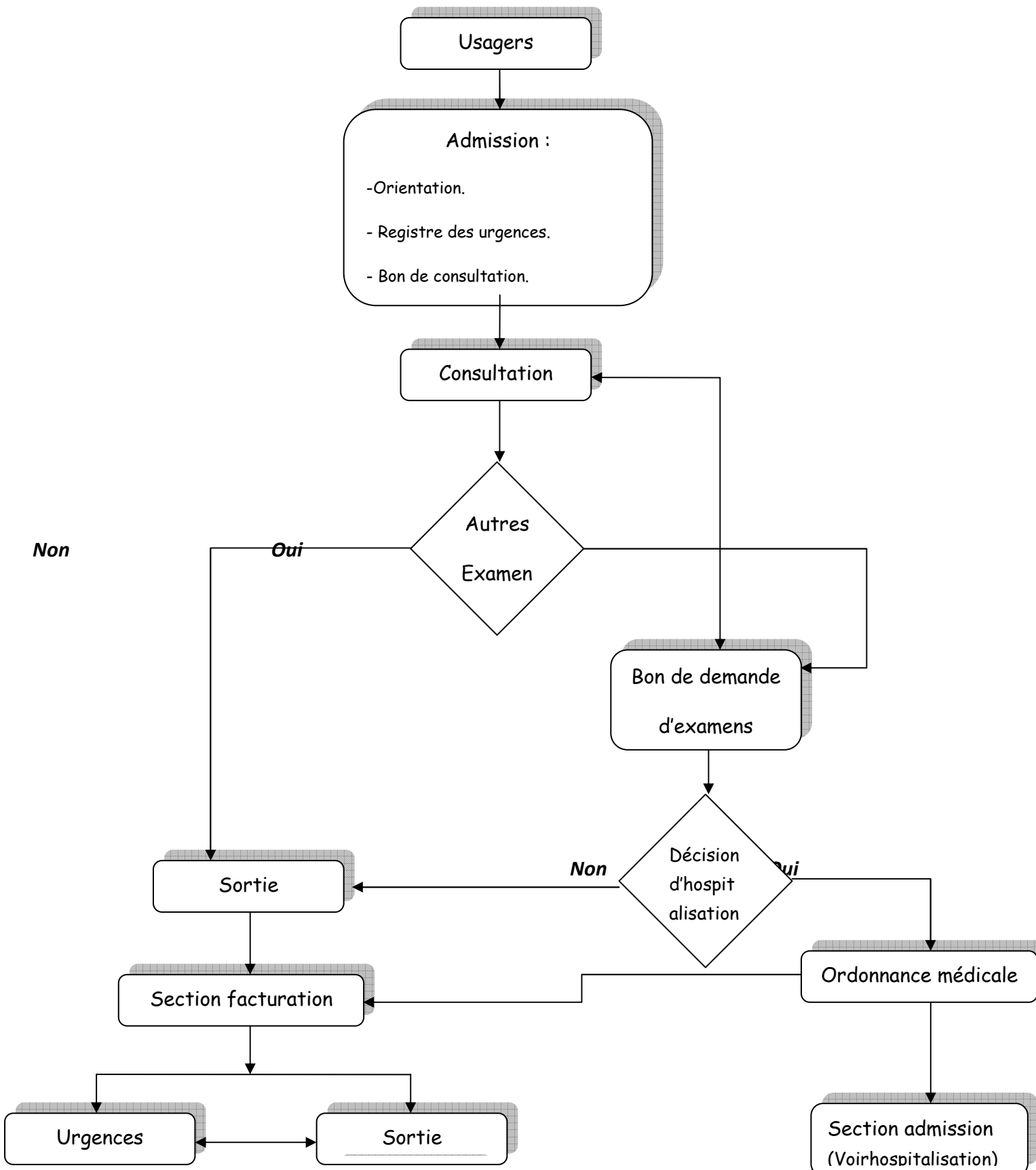
Annexe N°1

Centre de diagnostic



Annexe N°2

Urgence



Annexe N°3

Consultation et Hospitalisation

