

MEMOIRE DE STAGE DE FIN D'ETUDE

Pour l'obtention du

MASTERE PROFESSIONNEL

« Nouvelles Technologies des Télécommunications et Réseaux »

Présentée par :

Marwa MZOUGHI

Développement d'une application SAAS pour les cabinets
Médicaux

Soutenu le : 08/02/2014

Devant le jury :

| | |
|--------------------------|-------------|
| Mme Hanen IDOUDI | : Président |
| Mme Ahlem BEN HSIN | : Membre |
| Mme Houda REKAYA HOUISSA | : Membre |

Remerciement

Que toutes les personnes qui m'ont aidé durant l'élaboration de ce travail trouvent dans ces lignes l'expression de ma reconnaissance et ma profonde gratitude. Je tiens remercier mon encadreur Mr. Chiheb BEN JEMIA qui m'a toujours prodigué encouragements et conseils et m'a réservé un temps précieux tout au long de notre projet

Je tiens aussi à exprimer ma gratitude reconnaissance à tous les membres de jury d'avoir accepté d'évaluer notre travail.

Je remercie vivement tous ceux qui m'ont assisté à développer mes compétences dans diverses disciplines tout au long du cursus de mastère N2TR.

Je porte toute ma gratitude aux enseignants de l'université virtuelle de Tunis, pour leur dévouement et leur assistance tout au long de mes études universitaires.

Une pensée particulière va à toutes les personnes qui ont contribuées de près ou de loin à l'élaboration de ce travail spécialement **Riadh Fitouhi, Mehdi LOUKIL, Hind ENNACEUR** et **Maissa MZOUGHJI**.

Dédicaces

A mes parents pour tout le bien qu'ils n'ont cessé de me prodiguer depuis ma naissance et pour tous les sacrifices qu'ils ont consentis pour moi

A mes sœurs : Manel, Malek, Maïssa et Chaima

A mon fiancé Mehdi

A tous mes amis et tous ceux qu'ont été là pour moi dans les bons mais aussi dans les mauvais moments spécialement Hind et Dorra

A tous mes collègues de l'UVT

Et pour tous ceux qui m'aiment en témoignage de mon attachement.

Avant propos

Cette étude entre dans le cadre de la préparation d'un projet de fin d'études pour l'obtention d'un diplôme de mastère en nouvelles technologie de télécommunications et réseaux

Afin de consolider mes connaissances théoriques par la pratique, le groupe Hotix m'a proposé de développer une application SAAS pour le cabinet d'un dentiste.

Le présent rapport est donc une synthèse de différentes étapes développées dans ce projet.

Table des matières

| | |
|---|----|
| Introduction générale..... | 1 |
| Chapitre 1 : Présentation théorique | 2 |
| I. Introduction : | 2 |
| II. Présentation de l'organisme d'accueil : | 2 |
| 1. Historique : | 2 |
| 2. Missions principales : | 2 |
| III. Présentation du projet : | 3 |
| IV. Conclusion : | 3 |
| Chapitre 2 : Spécification du système | 4 |
| I. Introduction : | 4 |
| II. Etude de l'existant : | 4 |
| 1. Présentation de l'application : | 4 |
| 2. Critique de l'existant : | 5 |
| III. La spécification des besoins : | 5 |
| 1. Identification des besoins : | 5 |
| 2. Identification des acteurs : | 6 |
| IV. Diagrammes de cas d'utilisation : | 7 |
| V. Conclusion : | 8 |
| Chapitre 3 : Conception..... | 9 |
| I. Introduction : | 9 |
| II. Le choix de la méthodologie : | 9 |
| III. Le langage de modélisation : | 10 |
| IV. Analyse : | 10 |
| 1. Diagramme de package : | 10 |
| 2. Analyse de l'application : | 11 |
| V. Conception : | 36 |
| 1. Conception du système : | 36 |
| 2. Conception des classes : | 40 |
| VI. Conclusion : | 42 |
| Chapitre 4 : Implémentation et réalisation | 43 |
| I. Introduction : | 43 |

| | | |
|------|--|----|
| II. | Outils et langage utilisé : | 43 |
| 1. | Spécification technique : | 43 |
| III. | Implémentation : | 45 |
| 1. | Architecture générale du projet: | 45 |
| 2. | Le Diagramme de déploiement : | 46 |
| IV. | Réalisation du projet: | 47 |
| V. | Conclusion :..... | 51 |
| | Conclusion générale | 52 |
| | Bibliographie..... | 53 |
| | Netographie | 53 |

Table des figures

| | |
|---|----|
| Figure 2-1 : Diagramme de cas d'utilisation général..... | 7 |
| Figure 3-1 : La démarche Agile | 9 |
| Figure 3-2 : Diagramme de package | 11 |
| Figure 3-3 : Diagramme de cas d'utilisation « authentification » | 12 |
| Figure 3-4 : Diagramme de cas d'utilisation « Gestion du cabinet »..... | 14 |
| Figure 3-5 : Diagramme de cas d'utilisation « Gestion de l'application »..... | 21 |
| Figure 3-6 : Diagramme de séquence système « s'authentifier » | 25 |
| Figure 3-7 : Diagramme de séquence système « créer une fiche patient » | 26 |
| Figure 3-8 : Diagramme de séquence système « Ajouter une ordonnance » | 27 |
| Figure 3-9 : Diagramme de séquence système « Ajouter un RDV » | 28 |
| Figure 3-10 : Diagramme de séquence système « modifier les droits d'accès d'un utilisateur » | 29 |
| Figure 3-11 : Diagramme des classes participantes « s'authentifier »..... | 30 |
| Figure 3-12 : Diagramme de classes participantes du cas d'utilisation« Ajouter un utilisateur »..... | 31 |
| Figure 3-13 : Diagramme de classes participantes du cas d'utilisation« Modifier les droits d'accès d'un utilisateur »..... | 31 |
| Figure 3-14: Diagramme de classes participantes du cas d'utilisation « Supprimer un cabinet »..... | 32 |
| Figure 3-15: Diagramme de classes participantes du cas d'utilisation « Créer une fiche patient »..... | 33 |
| Figure 3-16: Diagramme de classes participantes du cas d'utilisation « Modifier une fiche patient »..... | 33 |
| Figure 3-17: Diagramme de classes participantes du pour le cas d'utilisation « Modifier un RDV »..... | 34 |
| Figure 3-18: Diagramme de classes participantes du cas d'utilisation « Modifier une ordonnance »..... | 34 |

| | |
|---|----|
| Figure 3-19: Diagramme de classes participantes du cas d'utilisation « Supprimer une lettre de liaison »..... | 35 |
| Figure 3-20: Architecture générale du projet..... | 36 |
| Figure 3-21: Diagramme de séquence « s'authentifier »..... | 37 |
| Figure 3-22: Diagramme de séquence « Ajouter un utilisateur »..... | 38 |
| Figure 3-23 : Diagramme de séquence « Effectuer la recherche »..... | 39 |
| Figure 3-24: Diagramme de séquence « Créer une fiche patient »..... | 40 |
| Figure 3-25 : Diagramme de classe..... | 41 |
| Figure 4-1: Architecture générale du projet..... | 46 |
| Figure 4-2 : Diagramme de déploiement..... | 47 |
| Figure 4-3 : Interface « authentification »..... | 48 |
| Figure 4-4 : Interface « Menu »..... | 48 |
| Figure 4-5 : « Interface Créer une fiche patient »..... | 49 |
| Figure 4-7: Interface « Paramétrage des dents »..... | 50 |
| Figure 4-8 : « Interface Créer une consultation »..... | 50 |
| Figure 4-9 : Interface « La liste des consultations »..... | 51 |

Introduction générale

Depuis plusieurs années, l'Internet a été utilisé comme un outil de messagerie, de publication d'information en fournissant des services susceptibles de faire évoluer la qualité des informations qui circulent à travers le réseau.

En effet, le réseau Internet fournit plusieurs services de diffusion et d'échange d'information à l'échelle planétaire dont WWW (World Wide Web), qui est un mécanisme simple et facile permettant la consultation des données réparties sur le réseau et ce indépendamment de leur emplacement, de leurs auteurs, des machines utilisées et enfin de leurs natures. De ce fait, un navigateur Web se révèle nécessaire pour faire circuler l'information.

Ce travail consiste à développer une application en SAAS pour les cabinets médicaux (cas d'un cabinet de dentiste), ce qui va leur permettre de gérer, les fiches patients, le planning et le stock de consommation sur le NET.

Afin de concevoir ce projet, j'ai opté pour la méthode Agile qui se situe à mi-chemin entre UP (Unified Process) qui est un cadre général très complet de processus de développement, et XP (eXtrem Programming), en appliquant les principes de modélisations Agile.

Cependant, le présent document est organisé comme suit : Un premier chapitre qui consiste à présenter théoriquement le système. Un deuxième chapitre qui présente la spécification du système. Un troisième chapitre qui détaille la conception et la présentation des scénarios envisagés. Un quatrième chapitre qui spécifie l'environnement technologique de l'application ainsi que la description des différentes tâches accomplies au titre de la réalisation et une conclusion qui établit un bilan du travail et dresse les perspectives du projet.

Chapitre 1 : Présentation théorique

I. Introduction :

De nos jours, il n'est plus aisé de définir les enjeux des nouvelles technologies de l'information dans la professionnalisation de la médecine. En effet, elles représentent un véritable atout, aident dans le travail du médecin et elles assurent la communication et la collaboration avec ses correspondants spécialistes et patients.

II. Présentation de l'organisme d'accueil :

1. Historique :

HOTIX est un groupe de sociétés de services informatiques créées en 1990. Sa principale force est d'être rattachée au groupe OPTIMS (France), présent dans 53 pays avec plus de 4000 références " actives " dans le monde: Tunisie, Maroc, Egypte, Algérie, France, Italie, Espagne, etc.

Les échanges entre les deux groupes donnent au groupe HOTIX l'opportunité de travailler dans un contexte international et lui permettant un accès plus aisé aux outils technologiques.

2. Missions principales :

L'activité du groupe HOTIX qui s'étend en grande partie à l'international englobe :

- Edition des logiciels: réalisation de projets informatiques dans les différents domaines de la gestion d'entreprise (conseil, conception, réalisation, mise en œuvre),
- La maintenance des logiciels installés dans les hôtels,
- Le développement Offshore : sous-traitance de projets informatiques internationaux avec une équipe dédiée ainsi qu'un ensemble de moyens, de méthodologie de gestion de projet, de suivi qualité et de logistique appropriés,
- La commercialisation de la solution de gestion hôtelière et de restauration du groupe OPTIMS, service comprenant le matériel, le logiciel, l'installation, la formation...

➤ HOTIX Tunisie est parmi les sociétés informatiques appartenant au groupe HOTIX. Elle dispose pour la maintenance des applications d'une équipe HD (HELP DESK) qui se charge de la réception et la suivie des appels et d'une équipe de développement AM (Application Management).

III. Présentation du projet :

Pour gérer son cabinet, chaque médecin rencontre plusieurs difficultés comme la perte d'information ainsi que le manque d'organisation en travail et l'endommagement des fiches après utilisation fréquente.

Cependant, afin d'éviter toutes sorte d'erreur et de régler tout type de problèmes, les nouvelles technologies de l'information peuvent être incluses dans le travail du médecin qui consistent à faciliter l'envoi et la réception des données relatifs aux patients via Internet et de régler le planning des visites.

Non seulement les applications web mais aussi les applications mobiles qui ont de nos jours une grande importance dans notre vie quotidienne.

IV. Conclusion :

Tout au long de ce chapitre, j'ai pu donner une présentation claire du cadre de travail. En outre, j'ai projeté l'objectif du système à développer. Pour cela, j'ai commencé par la présentation de l'organisme d'accueil et ses différents produits. Puis, j'ai pu introduire le projet que je vise à développer. Par ailleurs, je vais m'intéresser, dans le chapitre suivant, aux besoins de l'application pour répondre aux attentes du système.

Chapitre 2 : Spécification du système

I. Introduction :

Dans ce chapitre je vais présenter le cahier des charges du projet. En premier lieu, je commencerai par l'étude de l'existant ainsi que l'étude du projet et les solutions proposées et en deuxième lieu, je préciserai les besoins fonctionnels et non fonctionnels.

Et enfin, je finirai par la présentation du diagramme de cas d'utilisation.

II. Etude de l'existant :

Concernant la collecte d'information, je me suis basée sur des informations fournies par un cabinet de dentiste.

En outre, cette collecte préalable d'informations est basée aussi sur les renseignements fournis par le directeur de recherche et de développement.

Enfin il faut préciser que la documentation et la visualisation d'un ensemble des solutions web ont été considérées comme une autre source de données qui comprend notamment celles qui sont indispensables pour permettre la mise en place du système.

1. Présentation de l'application :

Dans le cadre de ce stage au sein du groupe HOTIX. J'ai eu l'opportunité d'effectuer ce projet de fin d'étude qui consiste à développer une application SAAS de gestion des cabinets médicaux et en particulier un cabinet de dentiste.

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

2. Critique de l'existant :

La gestion de chaque cabinet médical peut avoir quelques difficultés. Celles-ci peuvent se résumer dans :

- Des problèmes de gestion des rendez-vous avec les patients.
- La répétition des informations relatives aux patients.
- Le manque de confidentialité de l'information.

III. La spécification des besoins :

Comme les bonnes questions représentent la moitié de la réponse dans la plupart des domaines, en informatique une bonne spécification des besoins est primordiale. En effet, elle représente le travail le plus délicat et le plus significatif, mais elle-même repose sur une bonne spécification des besoins qui n'est autre que la question que doit se poser tout ingénieur au début de son travail " Qu'est ce qu'on veut que nous fassions ? ". Dans cette section nous allons présenter notre réponse à cette question. Donc, cette phase consiste à mieux comprendre le contexte du système, il s'agit de déterminer les fonctionnalités et les acteurs et d'identifier les cas d'utilisation initiaux.

1. Identification des besoins :

❖ Les besoins fonctionnels :

Les besoins fonctionnels représentent les principales fonctionnalités du système. Ces besoins proviennent généralement des utilisateurs du système. Cette application devra permettre :

- La gestion de l'application : Cette partie permet l'ajout et la suppression des différents utilisateurs.
- La gestion du cabinet : Elle s'agit de la gestion des fiches des patients, la gestion des ordonnances, la gestion des lettres de liaison et la gestion de la caisse.
- La Gestion du stock : Elle sert à contrôler le stock d'un cabinet de dentiste.
- La gestion du planning : Elle concerne la vérification des rendez-vous ainsi que la possibilité au patient de saisir un rendez-vous via le net directement.

❖ Les besoins non fonctionnels :

A part les besoins fondamentaux, notre futur système doit répondre aux critères suivants :

- La rapidité du traitement: En effet, vu le nombre important des transactions quotidiennes, il est impérativement nécessaire que la durée d'exécution des traitements s'approche le plus possible du temps réel.
- La performance: Un logiciel doit être avant tout performant c'est-à-dire à travers ses fonctionnalités, il doit répondre à toutes les exigences des usagers d'une manière optimale.
- La convivialité: Le futur 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.
- La confidentialité: vu que les données manipulées par notre application sont critiques, nous devons garantir une sécurité optimale. Ainsi, les droits d'accès au système doivent être bien attribués, afin d'assurer la sécurité des données.

2. Identification des acteurs :

- L'administrateur : Il possède le droit d'accéder aux différentes rubriques de l'application afin d'ajouter des nouveaux cabinets au réseau et d'attribuer des niveaux accès pour chaque utilisateur au niveau de son profil afin de lui permettre ou pas d'agir sur certaines données au niveau d'un module donné dans un établissement donné.
- Le médecin : Le médecin peut être lui-même l'administrateur. Il peut gérer le planning de consultation quotidien. Il peut ajouter, modifier ou supprimer au niveau de la gestion des ordonnances, la gestion des fiches de patient, la gestion des lettres de liaison ainsi que la gestion de stock.
- La secrétaire : La secrétaire consulte la disponibilité du médecin et réserve un RDV pour un patient. Elle peut mettre à jour les fiches des patients et elle gère la facturation (montant et date de chaque facture).
Ainsi, elle selon le choix du patient, se charge de suivre le RDV en lui envoyant un E-mail ou bien un SMS.

- Le patient : Le patient peut accéder au planning des visites via internet afin de demander un RDV, il peut modifier ou annuler un RDV.

IV. Diagrammes de cas d'utilisation :

La figure suivante présente le diagramme de cas d'utilisation générale :

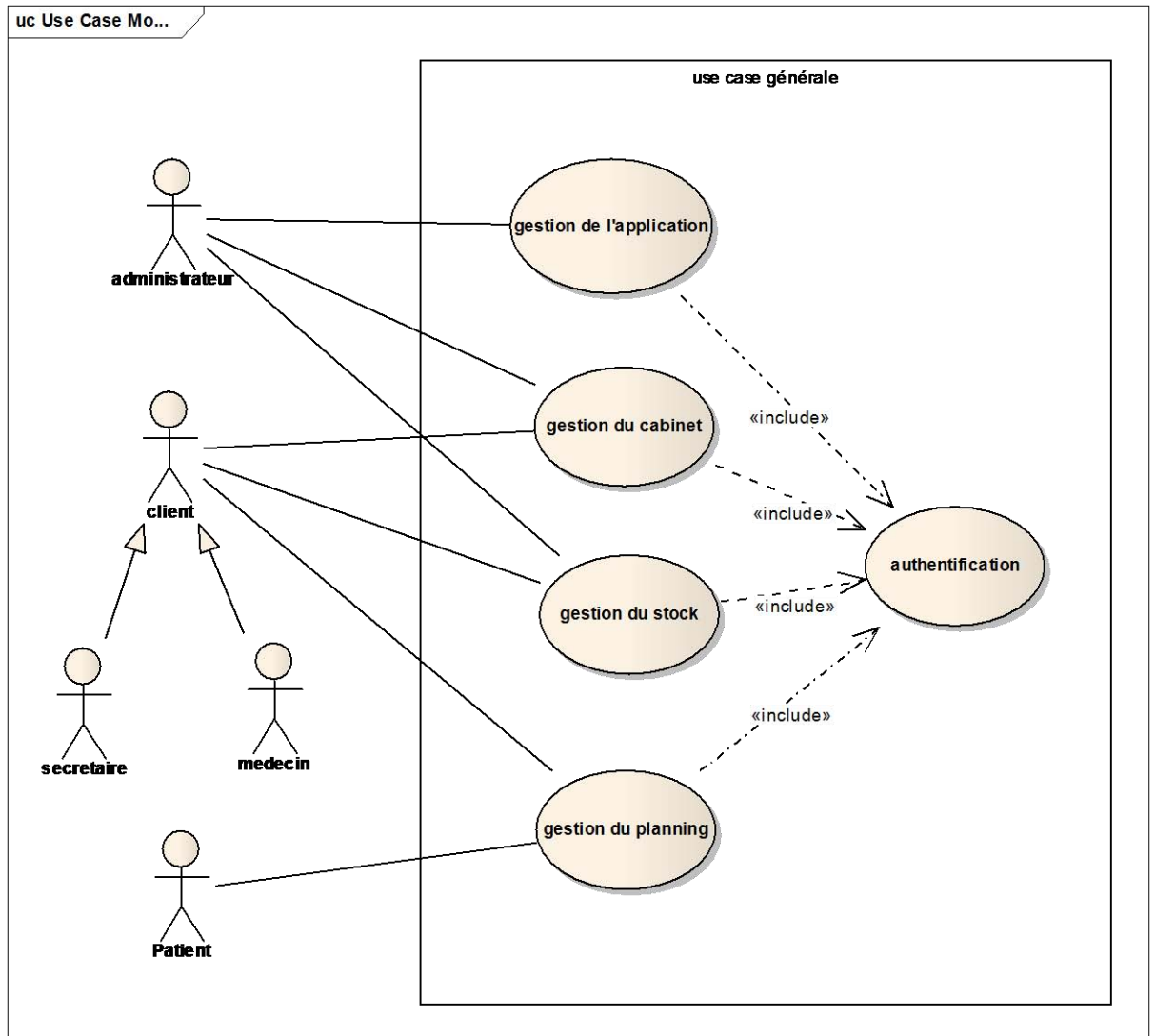


Figure 2-1 : Diagramme de cas d'utilisation général

Le déroulement de l'application est représenté par les cas d'utilisation suivants :

- **Authentification :**
 - Assure la sécurité d'accès à l'application et spécifie le type de l'utilisateur.

- **Gestion de l'application :**
 - Ajouter, modifier et supprimer un utilisateur.
 - Ajouter, modifier et supprimer un cabinet.

- **Gestion du cabinet :**
 - Gérer les fiches de patients : ajouter, modifier et supprimer une ou plusieurs fiches de patients.
 - Gérer les RDV : ajouter, modifier et supprimer un RDV.
 - Suivre les RDV.
 - Gérer les factures.
 - Gérer les ordonnances : ajouter, modifier et supprimer une ordonnance.
 - Gérer les lettres de liaison : ajouter, modifier et supprimer une lettre de liaison.
- **Gestion des plannings :** Elle concerne la prise des rendez-vous, même les patients peuvent y accéder

V. Conclusion :

Dans ce chapitre j'ai posé les questions fondamentales pour spécifier les tâches. Cette spécification va servir dans le prochain chapitre à trouver la réponse la plus convenable à ces questions et ainsi résoudre le problème en question.

Chapitre 3 : Conception

I. Introduction :

Partant du contexte technologique actuel, il semble évident que la conception de toute application informatique n'est autre que le fruit d'efforts conjugués pour avoir une bonne spécification. D'ailleurs, il est de plus en plus difficile de garantir la bonne qualité d'une conception sans, pour autant, avoir une bonne spécification ni même négliger son importance puisqu'il s'agit d'une réponse au problème posé.

Dans ce chapitre je vais présenter la conception vis à vis du problème posé. Ce dernier n'est autre que la réalisation d'une application web services du système d'une gestion d'un cabinet de dentiste.

II. Le choix de la méthodologie :

La conception générale et détaillée, sera réalisée en s'appuyant sur le langage de modélisation UML2 en appliquant les principes de la méthodologie Agile qui vise à réduire le cycle de vie du logiciel, c'est-à-dire accélérer son développement en développant une version minimale, puis en intégrant les fonctionnalités par un processus itératif basé sur une écoute client et des tests tout au long du cycle de développement.

Le schéma ci-dessous décrit la démarche Agile :

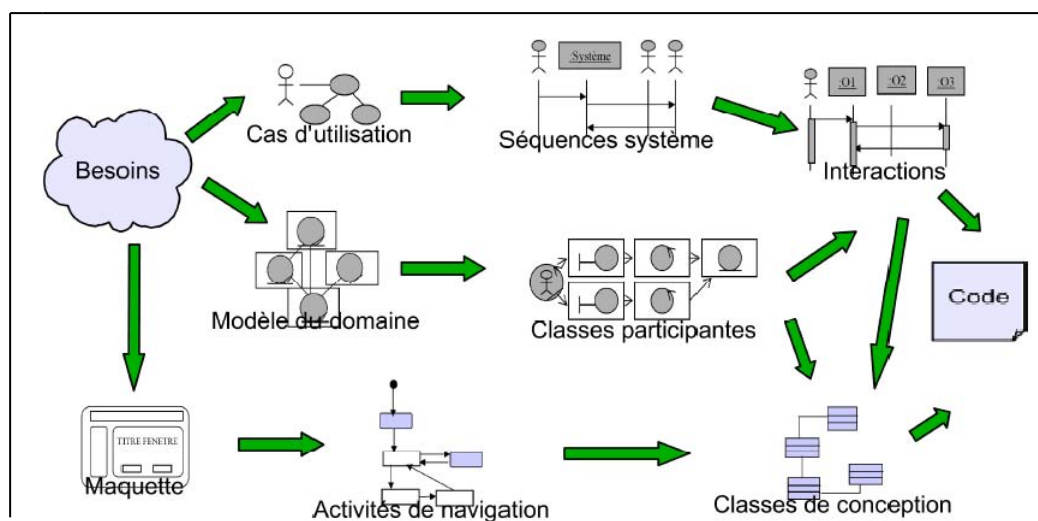


Figure 3-1 : La démarche Agile

III. Le langage de modélisation :

UML (en anglais Unified Modeling Language ou « langage de modélisation unifié ») est un langage de modélisation graphique à base de pictogrammes. Il est apparu dans le monde du génie logiciel, dans le cadre de la « conception orientée objet ». Couramment utilisé dans les projets logiciels, il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique. En effet, l'UML nous permet une meilleure conception du côté de l'application avec ses notions d'objets et de classes, et nous donne une décomposition claire et simple afin de dégager les entités et les classes nécessaires.

IV. Analyse :

L'analyse est la phase qui répond à la question « que faut-il faire ? », elle a pour but de se doter d'une vision claire et rigoureuse du problème posé et du système à réaliser en déterminant ses éléments et leurs interactions. Nous allons commencer par une analyse du domaine puis une analyse de l'application.

1. Diagramme de package :

Cette application peut être divisée en 3 packages :

- Gestion de l'application ➔ gestion des comptes utilisateurs
- Gestion du cabinet
- Gestion du stock

Ci-dessous le diagramme de package :

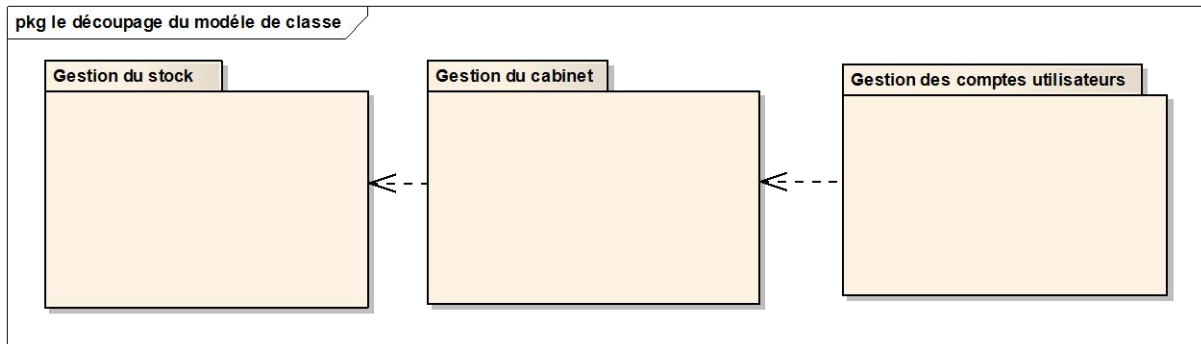


Figure 3-2 : Diagramme de package

2. Analyse de l'application :

L'analyse de l'application nous permet d'effectuer la modélisation des fonctionnalités de l'application les plus importantes.

Les uses cases permettent de structurer les besoins des utilisateurs et les objectifs correspondants d'un système. Ils centrent l'expression des exigences du système sur ses utilisateurs : ils partent du principe que les objectifs du système sont tous motivés. Ils se limitent aux préoccupations "réelles" des utilisateurs ; ils ne présentent pas de solutions d'implémentation et ne forment pas un inventaire fonctionnel du système.

Il faudra bien noter que les cas d'utilisation identifient les utilisateurs du système ainsi que leurs interactions.

2.1 Diagrammes de cas utilisation :

2.1.1 Cas d'utilisation « authentification »

Ci-dessous le diagramme de cas d'utilisation de l'authentification :

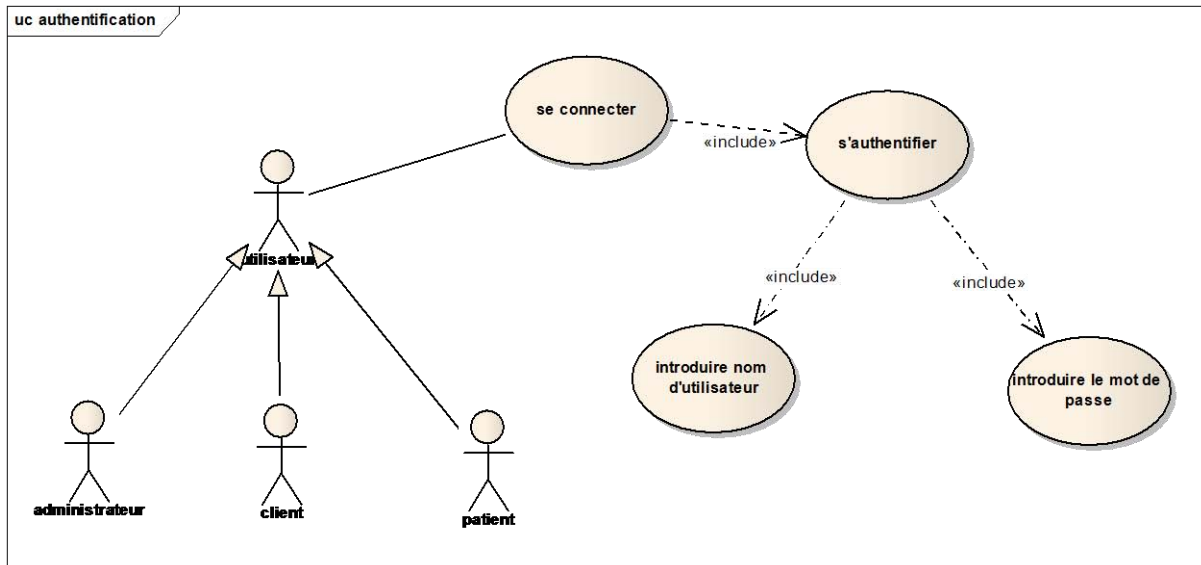


Figure 3-3 : Diagramme de cas d'utilisation « authentification »

Pour accéder au système, l'utilisateur introduit son nom et son mot de passe. Si les informations sont correctes une page d'accueil s'affiche selon l'utilisateur, sinon le système renvoie un message d'erreur.

Chaque utilisateur du système (Administrateur, médecin, secrétaire, patient) possède des privilèges d'accès définis auparavant par l'administrateur de l'application.

❖ **Cas d'utilisation « s'authentifier » :**

Titre : s'authentifier

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

Acteurs : administrateur, médecin, secrétaire et patient.

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

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

Scénario nominal :

1. L'utilisateur tape son propre nom d'utilisateur et son mot de passe.
2. L'utilisateur choisit le cabinet pour y accéder.
3. Le système vérifie la validité des coordonnées de l'utilisateur.
4. L'utilisateur va accéder à la session qui le concerne.

Scénario alternatif :

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

L'enchaînement reprend à l'étape 1 du scénario nominal.

2.1.2 Diagramme de cas d'utilisation « gestion du cabinet »

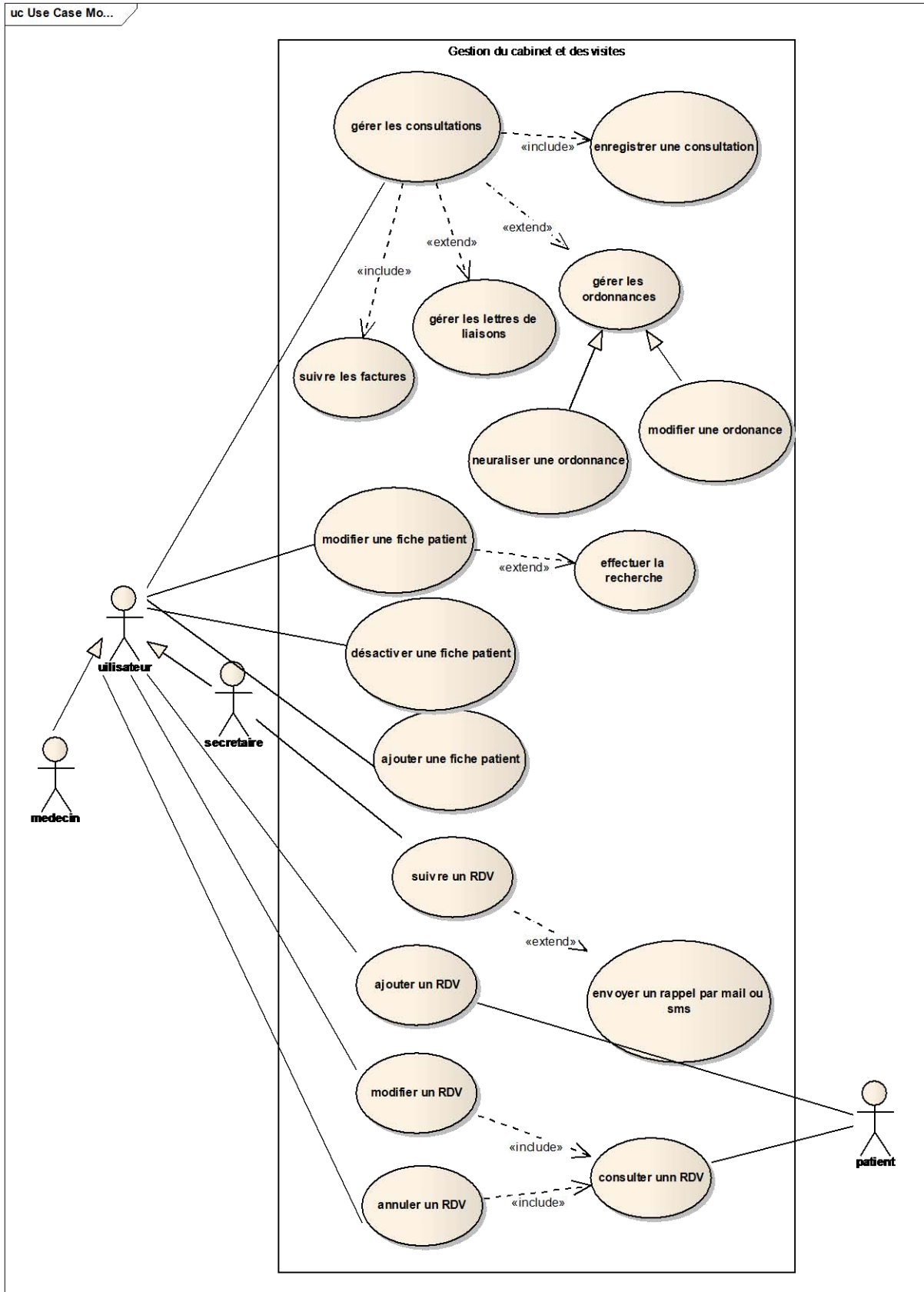


Figure 3-4 : Diagramme de cas d'utilisation « Gestion du cabinet »

➤ **Cas d'utilisation « Ajouter une fiche patient »**

Acteurs : Utilisateur (médecin ou secrétaire)

Précondition : L'utilisateur est connecté, la fiche patient n'est pas créée

Post condition : Fiche patient créée

Scénario nominale :

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

Scénario alternatifs :

A : Les champs du formulaire ne sont pas remplis

L'enchaînement A démarre au point 4 du scénario nominal

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

Le scénario nominal reprend au point 3

➤ **Cas d'utilisation « modifier une fiche patient »**

Acteurs : Utilisateur (médecin ou secrétaire)

Précondition : fiche patient actuelle

Post condition : fiche patient mise à jour

Scénario nominal :

1. L'utilisateur demande le formulaire de fiche patient
2. Le système affiche le formulaire
3. L'utilisateur demande de rechercher l'identifiant du patient
4. Le système effectue la recherche
5. Le système envoie les informations du patient concerné

6. L'utilisateur effectue les modifications nécessaires
7. L'utilisateur valide les modifications
8. Le système enregistre la mise à jour

Enchaînement d'exception :

E1 : L'utilisateur annule la demande de modification

L'enchaînement E1 démarre au point 6 du scénario nominal

7. L'utilisateur annule la modification
8. Le système enregistre la page sans modification

➤ **Cas d'utilisation « Effectuer la recherche »**

Résumé : ce cas d'utilisation permet de trouver rapidement les informations nécessaires

Acteurs : Utilisateur (médecin ou secrétaire)

Précondition : fiche non trouvée

Post condition : fiche

Scénario nominal :

1. L'utilisateur lance une recherche par identifiant
2. Le système vérifie l'identifiant
3. Le système affiche les résultats de la recherche
4. L'utilisateur sélectionne l'info demandée
5. Le système affiche la page demandée

Enchaînement alternatifs :

A : Le système n'a pas trouvé l'identifiant recherché

L'enchaînement A démarre au point 2 du scénario nominal

3. Le système signale l'échec à l'utilisateur et propose une nouvelle recherche

Le scénario nominal reprend au point 1

➤ **Cas d'utilisation « gérer les consultations »**

Résumé : c'est le médecin qui se charge de la gestion des consultations qui se base sur l'enregistrement des ordonnances, des lettres de liaison, des factures et des fiches patients

Acteurs : médecin

Pré condition : consultation non enregistrée

Post condition : consultation enregistrée

➤ **Sous cas « ajouter une ordonnance »**

Acteurs : médecin

Pré condition : ordonnance non existante

Post condition : ordonnance ajoutée

Scénario nominal :

1. Le médecin demande le modèle d'ordonnance
2. Le système injecte les infos (patient, actes) dans le modèle
3. Le médecin modifie au préalable
4. Le système valide l'ajout
5. Le système effectue l'enregistrement

➤ **Sous cas « annuler une ordonnance »**

Acteurs : médecin

Pré condition : ordonnance existante

Post condition : ordonnance annulée

Scénario nominal :

1. Le médecin demande le modèle d'ordonnance
2. Le système injecte les infos (patient, actes) dans le modèle
3. Le médecin désactive l'ordonnance demandé
4. Le système valide l'annulation
5. Le système effectue l'enregistrement

❖ Cas d'utilisation « ajouter un RDV »

Résumé : C'est la secrétaire qui se charge de la gestion des RDV, elle consulte le planning des visites et selon la disponibilité du médecin elle ajoute un nouveau RDV. En outre, elle peut modifier ou annuler un RDV selon le choix du patient.

Acteurs : secrétaire

Pré condition : date de RDV non renseignée

Post condition : date RDV fixée

Scénario nominal :

1. L'utilisateur demande la liste des RDV
2. Le système affiche la liste
3. L'utilisateur choisit une date et heure selon la disponibilité
4. Le système vérifie la disponibilité
5. Le système confirme l'ajout
6. Le système enregistre l'ajout

Scénario d'exception :

E1 : si la date des RDV est non disponible, le système va afficher un message d'erreur car l'opération est impossible.

L'enchaînement démarre à l'étape 4 du scénario nominal.

L'enchaînement reprend à l'étape 3 du scénario nominal.

➤ Sous cas « Annuler RDV »

Acteurs : La secrétaire

Pré conditions : RDV existant.

Post conditions : RDV non existant.

1. La secrétaire demande la liste des RDV.
2. Le système affiche la liste.
3. La secrétaire cherche le RDV demandé (par n°fiche patient).

4. Le système effectue la recherche
5. Le système affiche le RDV demandé.
6. La secrétaire supprime le RDV.
7. Le système enregistre la suppression.
8. Le système confirme la suppression.

Scénario d'exception :

E1 : Si les informations saisies par l'utilisateur sont erronées, le système affiche un message d'erreur.

L'enchaînement démarre à l'étape 4 du scénario nominal.

L'enchaînement reprend à l'étape 3 du scénario nominal.

❖ Cas d'utilisation « modifier un RDV »

Acteurs : La secrétaire

Pré conditions : RDV avec la date initiale.

Post conditions : RDV modifié.

Scénario nominal :

1. La secrétaire demande la liste des RDV.
2. Le système affiche la liste.
3. La secrétaire cherche le RDV demandé (par date, n°fiche patient).
4. Le système effectue la recherche
5. Le système affiche le RDV demandé.
6. La secrétaire modifie le RDV.
7. Le système confirme la modification.
8. Le système valide la modification.

Scénario d'exception :

E1 : Si les informations saisies par l'utilisateur sont erronées, le système affiche un message d'erreur.

L'enchaînement démarre à l'étape 4 du scénario nominal.

L'enchaînement reprend à l'étape 3 du scénario nominal.

❖ Cas d'utilisation « suivre RDV »

Acteurs : La secrétaire

Résumé : La secrétaire se charge d'envoyer un appel téléphonique, SMS ou email au patient pour suivre son RDV avec le médecin.

Scénario nominal :

1. La secrétaire demande de moyen de suivi de RDV.
2. Le système choisit un moyen (email, SMS)
3. En cas de RDV, la secrétaire envoie manuellement un mail ou un SMS.

2.1.3 Diagramme de cas d'utilisation « gestion de l'application » :

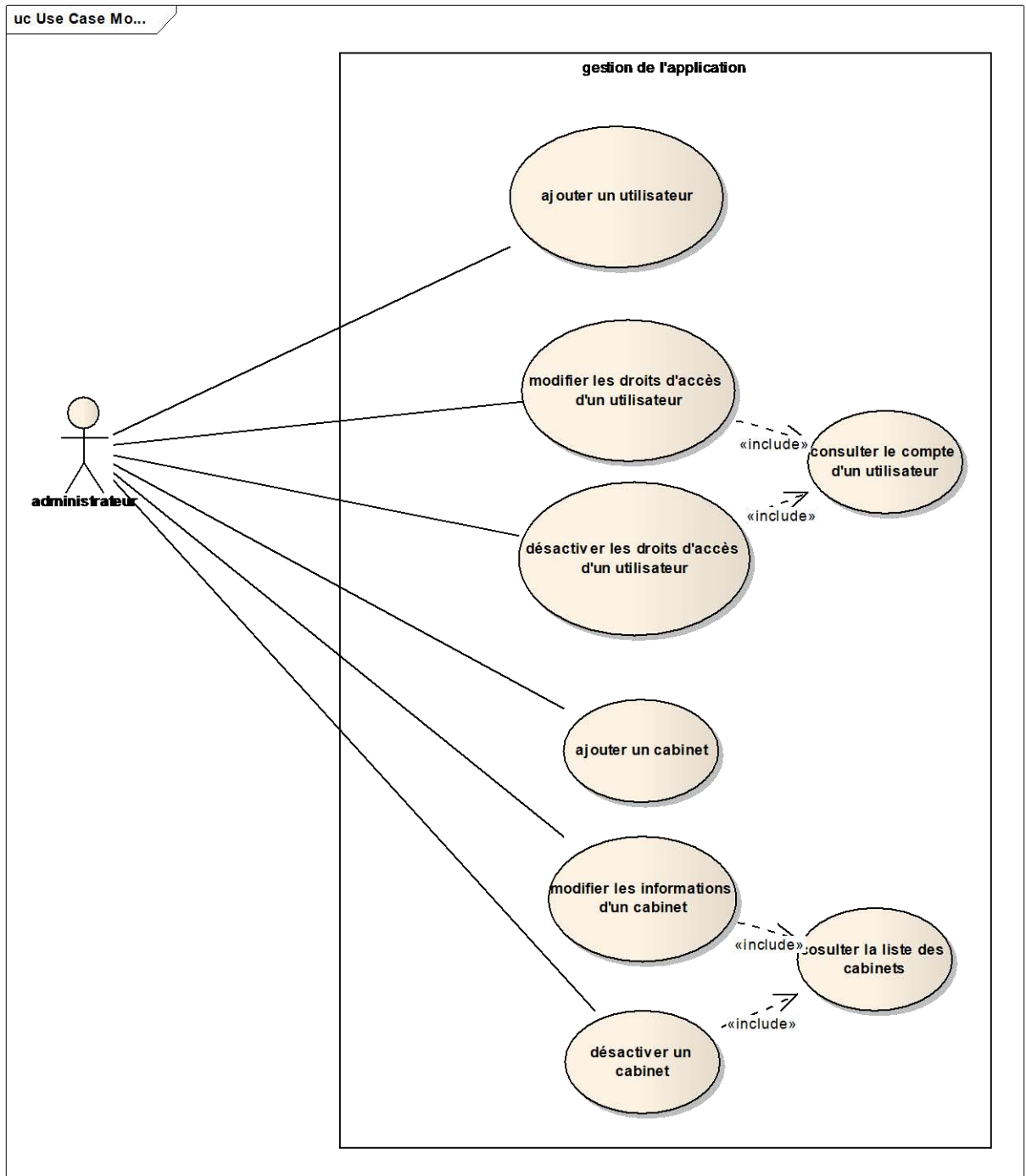


Figure 3-5 : Diagramme de cas d'utilisation « Gestion de l'application »

❖ Cas d'utilisation « ajouter utilisateur »

Acteurs : administrateur

Pré conditions : utilisateur non existant.

Post conditions : utilisateur existant.

Scénario nominal :

1. L'administrateur tape l'identifiant de l'utilisateur.
2. Le système vérifie l'existence de l'identifiant.
3. L'administrateur lui attribue un mot de passe et un nom d'utilisateur
4. Le système enregistre le nouvel ajout.

Scénario d'exception :

E1 : si l'utilisateur est déjà existant, le système va afficher un message d'erreur car l'opération est impossible.

L'enchaînement démarre à l'étape 2 du scénario nominale.

L'enchaînement reprend à l'étape 1 du scénario nominale.

❖ Cas d'utilisation « supprimer utilisateur »

Acteurs : administrateur

Pré conditions : utilisateur existant.

Post conditions : utilisateur supprimé.

Scénario nominal :

L'administrateur tape l'identifiant de l'utilisateur.

Le système vérifie l'existence de l'identifiant.

L'administrateur supprime l'utilisateur ainsi que les coordonnées qui le concerne.

Le système enregistre la suppression.

Scénario d'exception :

E1 : si l'identifiant de l'utilisateur n'existe pas dans la base de donnée, le système va afficher un message d'erreur car l'opération est impossible.

L'enchaînement démarre à l'étape 2 du scénario nominale.

L'enchaînement reprend à l'étape 1 du scénario nominale.

❖ Cas d'utilisation « modifier les droits d'accès d'un utilisateur »

Acteurs : administrateur

Pré conditions : utilisateur avec droits d'accès initiaux.

Post conditions : utilisateur avec droits d'accès modifiés.

Scénario nominal :

1. L'administrateur tape l'identifiant de l'utilisateur.
2. Le système vérifie l'existence de l'identifiant.
3. L'administrateur modifie les droits d'accès de l'utilisateur de l'application.
4. Le système enregistre la modification.

Scénario d'exception :

E1 : si l'identifiant de l'utilisateur n'existe pas dans la base de données, le système va afficher un message d'erreur car l'opération est impossible.

L'enchaînement démarre à l'étape 2 du scénario nominale.

L'enchaînement reprend à l'étape 1 du scénario nominale.

❖ Cas d'utilisation « ajouter un cabinet »

Acteurs : administrateur.

Pré conditions : cabinet non existant.

Post conditions : cabinet existant.

Scénario nominal :

1. L'administrateur tape l'identifiant du cabinet.
2. Le système vérifie l'existence de l'identifiant.
3. L'administrateur introduit les informations du cabinet.
4. Le système enregistre les informations.
5. Le système confirme l'ajout.

Scénario d'exception :

E1 : si le cabinet est déjà existant, le système va afficher un message d'erreur car l'opération est impossible.

L'enchaînement démarre à l'étape 2 du scénario nominal

L'enchaînement reprend à l'étape 1 du scénario nominal

❖ Cas d'utilisation « supprimer un cabinet »

Acteurs : administrateur.

Pré conditions : cabinet existant.

Post conditions : cabinet non existant.

Scénario nominal :

1. L'administrateur tape l'identifiant du cabinet.
2. Le système vérifie l'existence de l'identifiant.
3. L'administrateur supprime le cabinet demandé.
4. Le système enregistre la suppression.
5. Le système confirme la suppression.

Scénario d'exception :

E1 : si le cabinet n'est pas existant, le système va afficher un message d'erreur car l'opération est impossible.

L'enchaînement démarre à l'étape 2 du scénario nominale.

L'enchaînement reprend à l'étape 1 du scénario nominale

2.2 Diagramme de séquence système :

Je vais dans cette étape isoler les messages propres aux cas d'utilisation et les représenter graphiquement sur des diagrammes de séquence UML.

Pour chaque message, le DSS montre non seulement les acteurs externes qui interagissent avec le système, mais également les événements système déclenchés par les acteurs.

Nous allons présenter le DSS correspondant à chaque cas d'utilisation développée.

Ci-dessous le diagramme de séquence correspondant au cas d'utilisation « s'authentifier » :

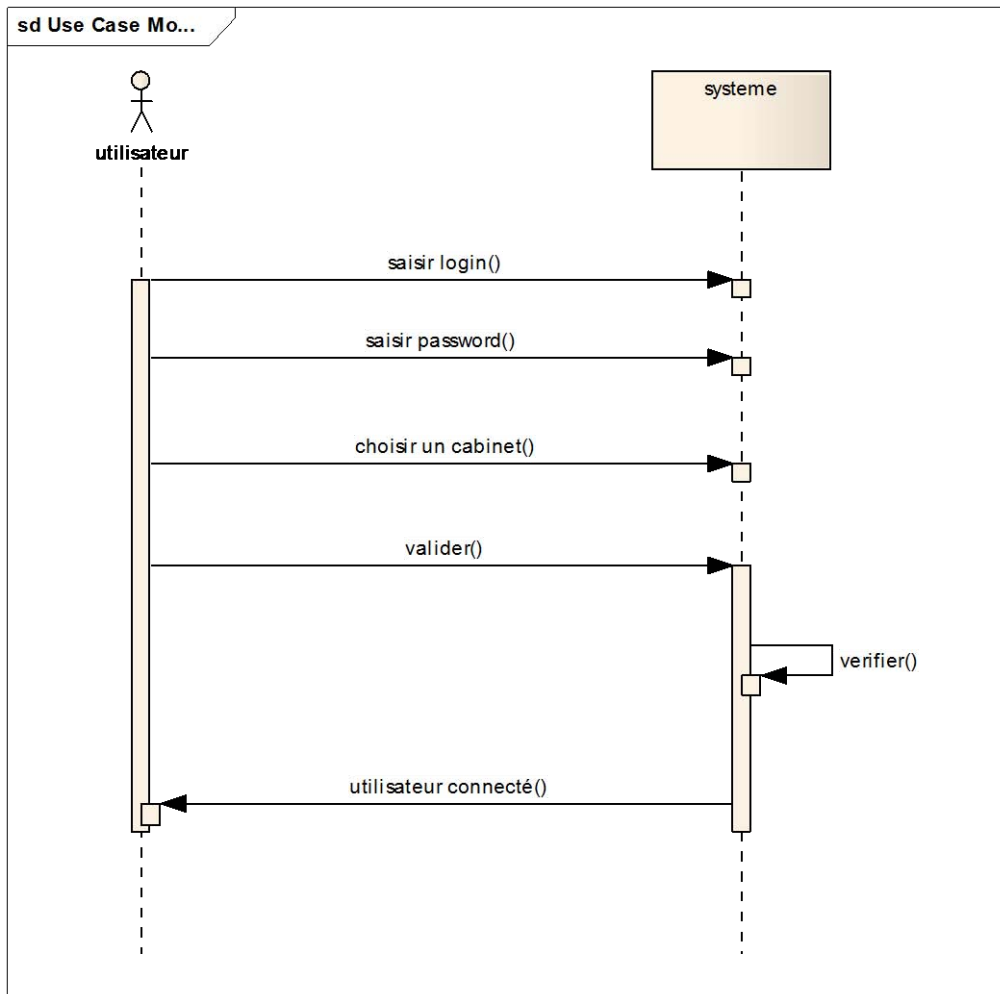


Figure 3-6 : Diagramme de séquence système « s'authentifier »

On considère le cas de la réussite de la saisie. L'utilisateur peut être soit l'administrateur soit le médecin ou la secrétaire.

L'authentification est une étape fondamentale dans l'application. Pour accéder au système, l'utilisateur introduit son nom d'utilisateur et mot de passe. Si les informations sont correctes une page d'accueil s'affiche selon l'utilisateur, si non le système renvoie un message d'erreur.

Chaque utilisateur du système (Administrateur, médecin, secrétaire) possède des privilèges d'accès définis auparavant par l'administrateur de l'application.

Ci-dessous le diagramme de séquence système « ajouter une fiche patient »

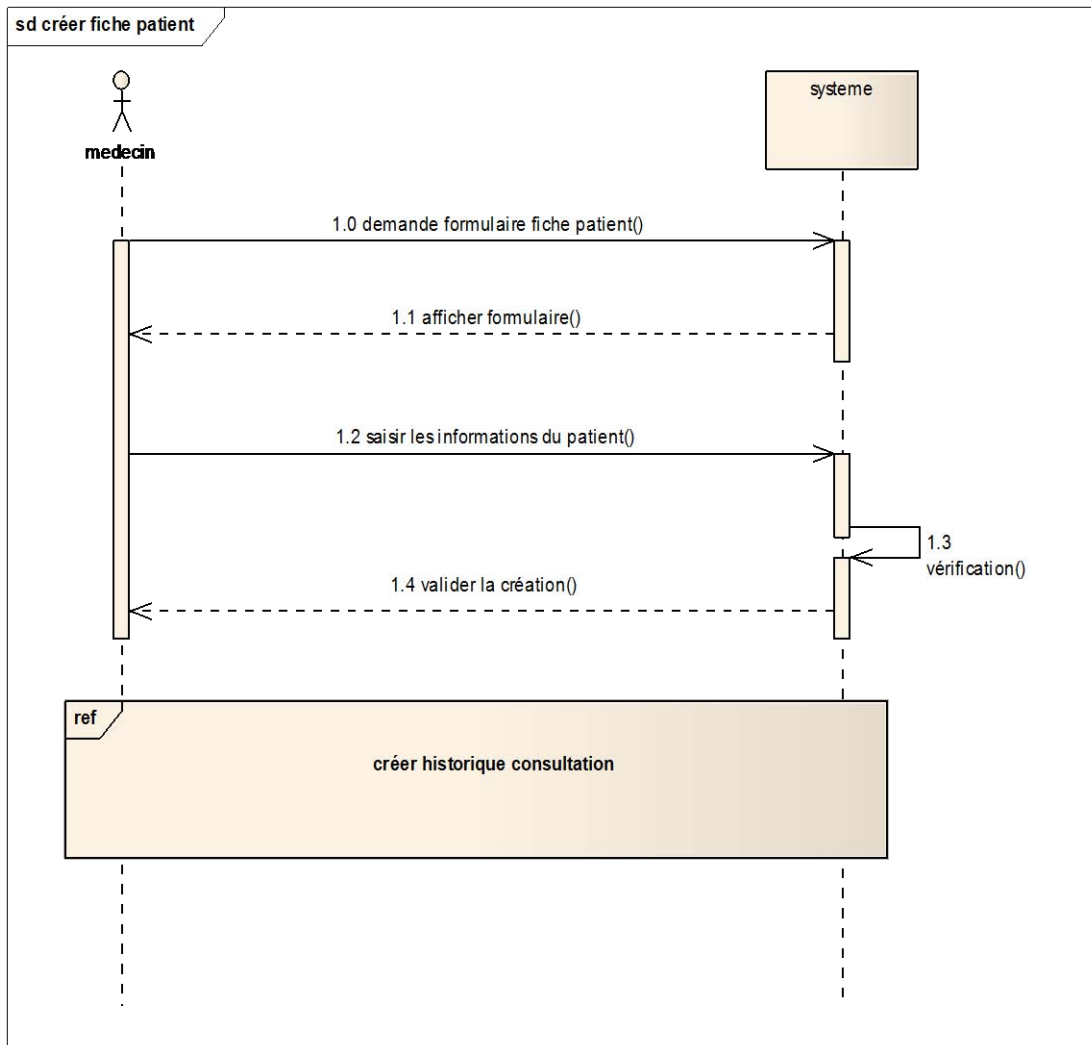


Figure 3-7 : Diagramme de séquence système « créer une fiche patient »

Ci-dessous le diagramme de séquence système « Ajouter une ordonnance » :

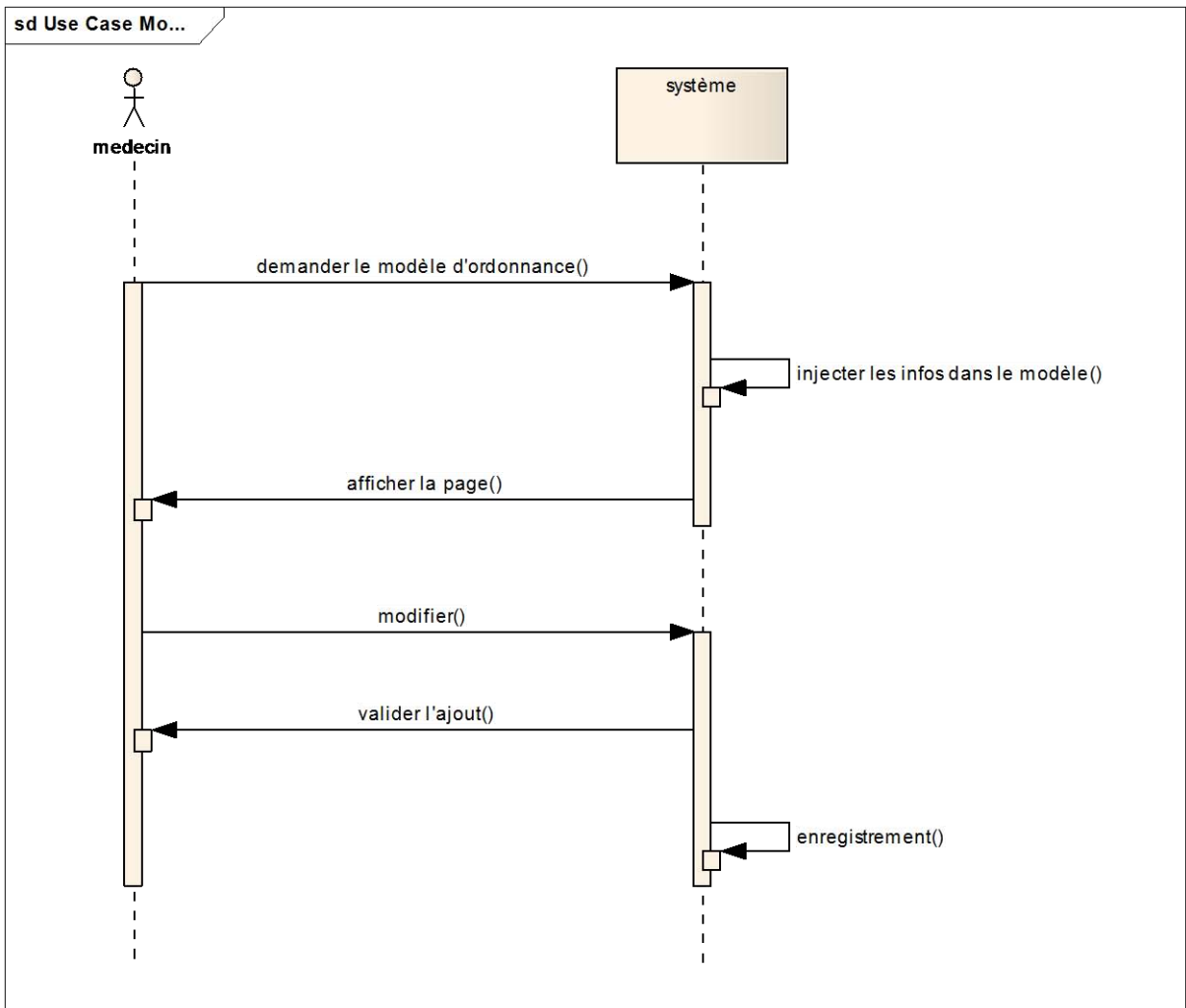


Figure 3-8 : Diagramme de séquence système « Ajouter une ordonnance »

La figure suivante illustre le diagramme de séquence relatif au cas d'utilisation « ajouter un RDV » :

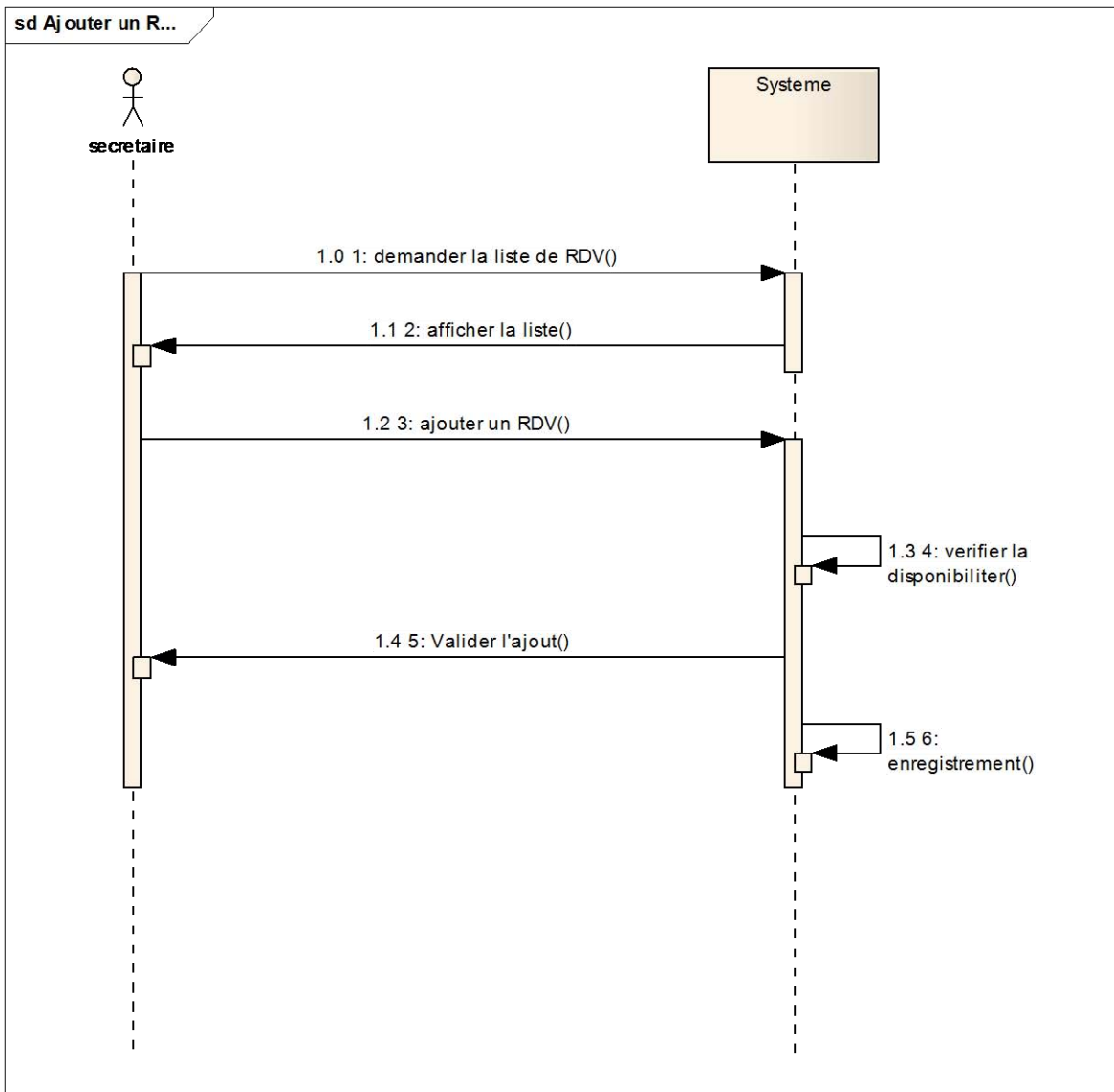


Figure 3-9 : Diagramme de séquence système « Ajouter un RDV »

La figure suivante montre le diagramme de séquence du cas d'utilisation « modifier les droits d'accès d'un utilisateur » :

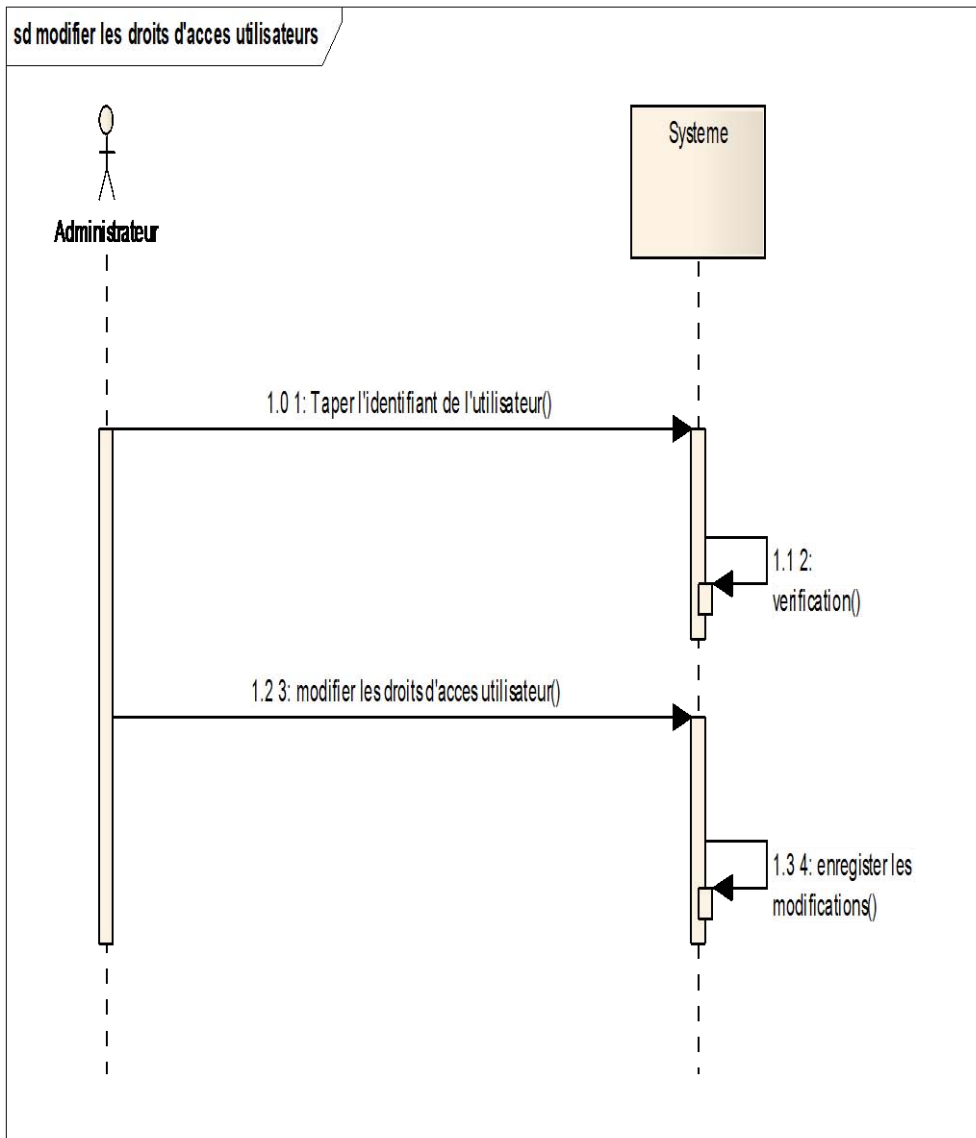


Figure 3-10 : Diagramme de séquence système « modifier les droits d'accès d'un utilisateur »

2.3 Diagrammes de classe participantes :

Il s'agit d'un diagramme qui sert à identifier les classes d'analyse qui vont participer à la réalisation des cas d'utilisation .Trois types de classes d'analyse peuvent exister dans ce diagramme :

- Les « dialogues » qui représentent les moyens d'interaction avec le système.
- Les « contrôles » qui contiennent la logique applicative.
- Les « entités » qui sont les objets métier manipulés.

Donc nous allons dans ce qui suit présenter pour chaque cas d'utilisation, son diagramme de classes participantes ainsi qu'une identification de ces classes.

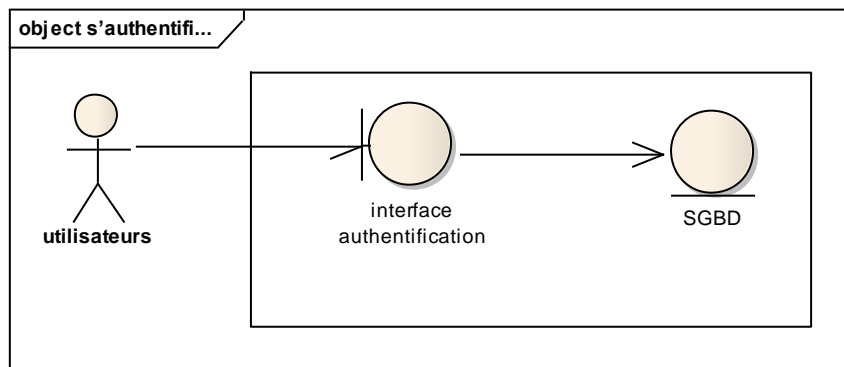
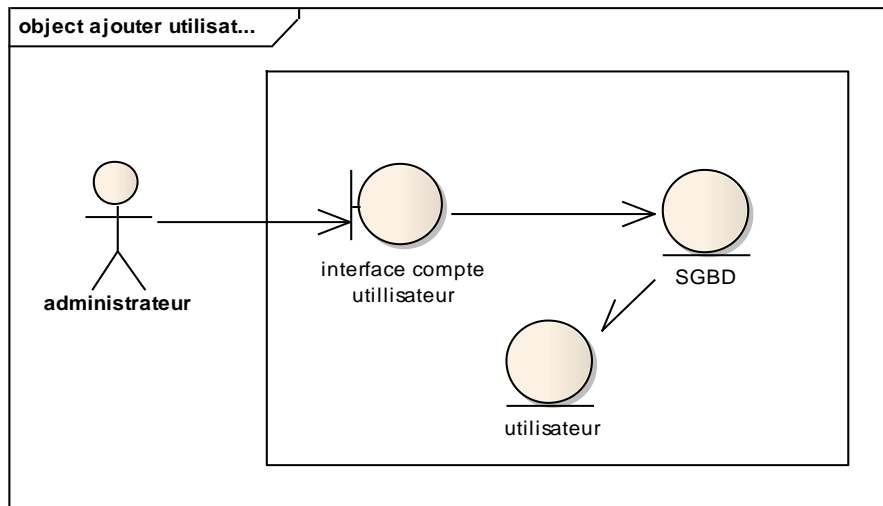


Figure 3-11 : Diagramme des classes participantes « s'authentifier »

La classe « interface -d'authentification » : c'est une classe dialogue qui permet à l'utilisateur, d'accéder à l'interface qui le concerne en tapant son nom d'utilisateur et son mot de passe.

La classe « SGBD » : c'est une classe contrôle qui permet de vérifier la validité des coordonnées tapées.

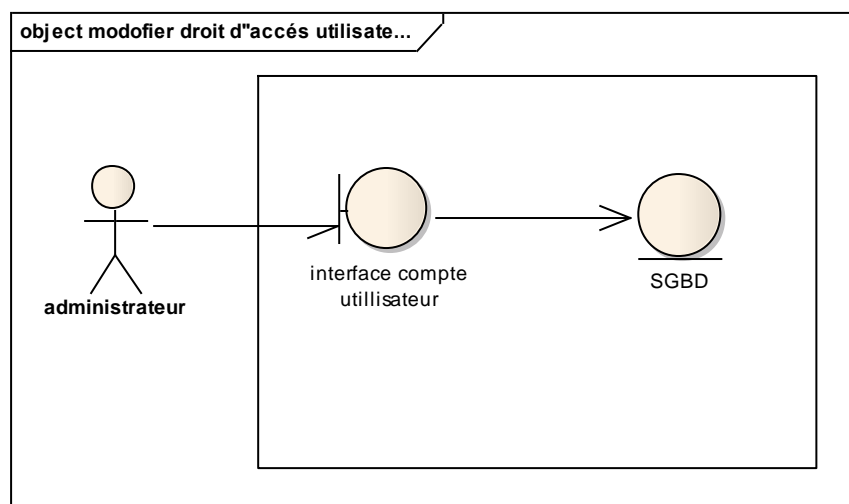


**Figure 3-12 : Diagramme de classes participantes du cas d'utilisation
« Ajouter un utilisateur »**

La classe « interface comptes utilisateurs » : c'est une classe dialogue qui permet de gérer les comptes utilisateurs et parmi ses manipulations l'ajout d'un nouveau utilisateur.

La classe « SGBD » : c'est une classe contrôle qui se charge de l'exécution de la requête relative à l'ajout d'un utilisateur.

La classe « utilisateur » : c'est une classe entité qui désigne le nouvel utilisateur ajouté.



**Figure 3-13 : Diagramme de classes participantes du cas d'utilisation
« Modifier les droits d'accès d'un utilisateur »**

La classe « interface comptes utilisateurs » : c'est une classe dialogue qui permet de gérer les comptes utilisateurs et parmi ses manipulations la modification des droits d'accès d'un utilisateur.

La classe « SGBD » : c'est une classe contrôle qui se charge de l'exécution de la requête relative à la suppression d'un utilisateur.

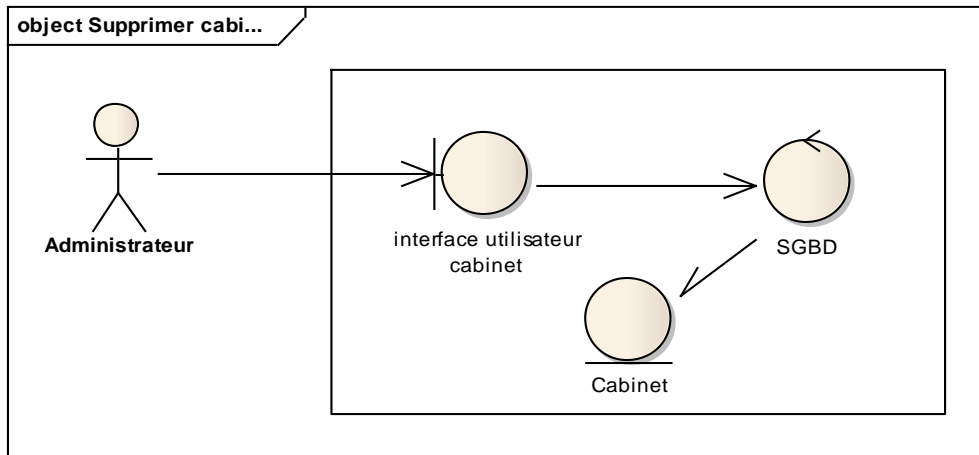


Figure 3-14: Diagramme de classes participantes du cas d'utilisation « Supprimer un cabinet »

La classe « interface cabinet » : c'est une classe dialogue qui permet de gérer les comptes utilisateurs et parmi ses manipulations la suppression d'un cabinet.

La classe « SGBD » : c'est une classe contrôle qui se charge de l'exécution de la requête relative à la suppression d'un cabinet.

La classe « cabinet » : c'est une classe entité qui désigne le cabinet supprimé.

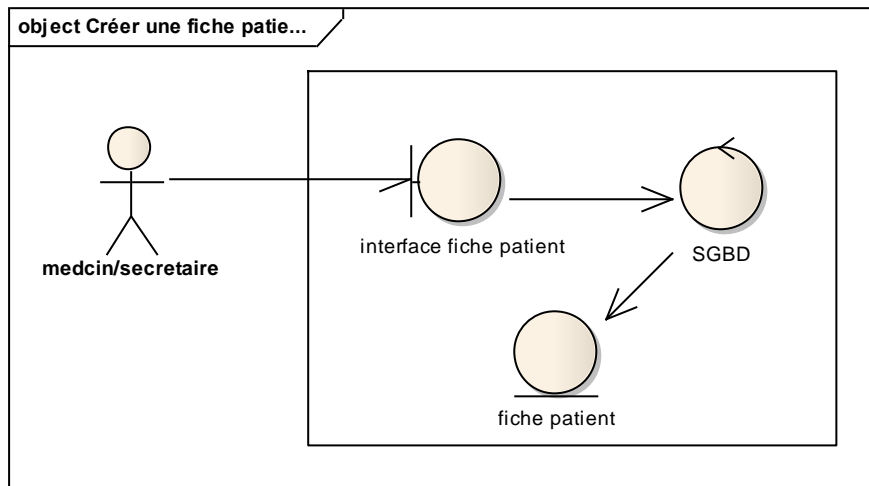


Figure 3-15: Diagramme de classes participantes du cas d'utilisation « Créer une fiche patient »

La classe « interface Fiche Patient » : c'est une classe dialogue qui permet de gérer les fiches patient et parmi ses manipulations l'ajout d'une nouvelle fiche patient.

La classe « SGBD » : c'est une classe contrôle qui se charge de l'exécution de la requête relative à l'ajout d'une fiche patient.

La classe « fiche patient » : c'est une classe entité qui désigne la fiche patient ajouté

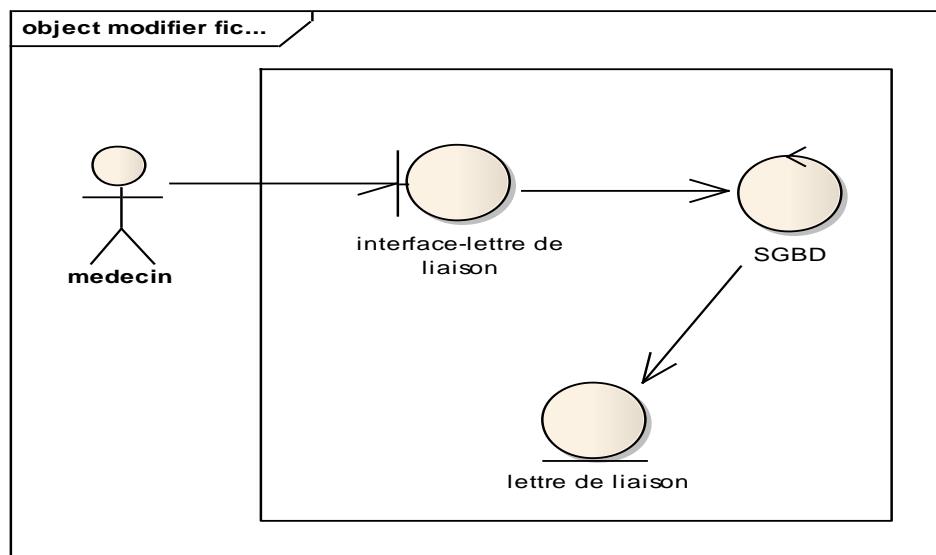


Figure 3-16: Diagramme de classes participantes du cas d'utilisation « Modifier une fiche patient »

La classe « interface fiche Patient » : c'est une classe dialogue qui permet de gérer les fiches patient et parmi ses manipulations la modification d'une fiche patient.

La classe « SGBD » : c'est une classe contrôle qui se charge de l'exécution de la requête relative à la modification d'une fiche patient.

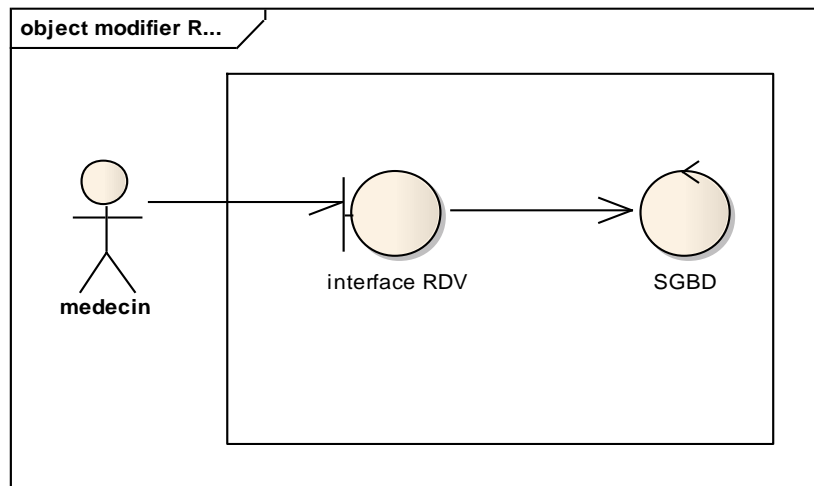


Figure 3-17: Diagramme de classes participantes du pour le cas d'utilisation « Modifier un RDV »

La classe « interface RDV » : c'est une classe dialogue qui permet de gérer les RDV et parmi ses manipulations la modification d'un RDV.

La classe « SGBD » : c'est une classe contrôle qui se charge de l'exécution de la requête relative à la modification d'un RDV.

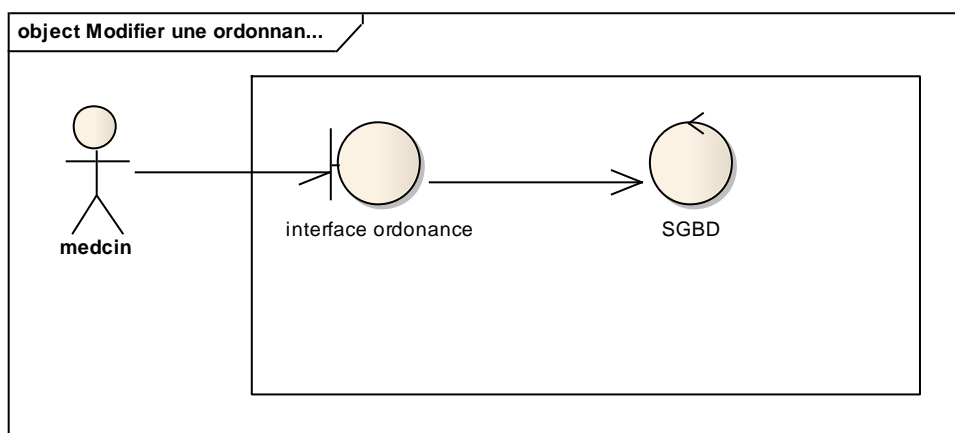
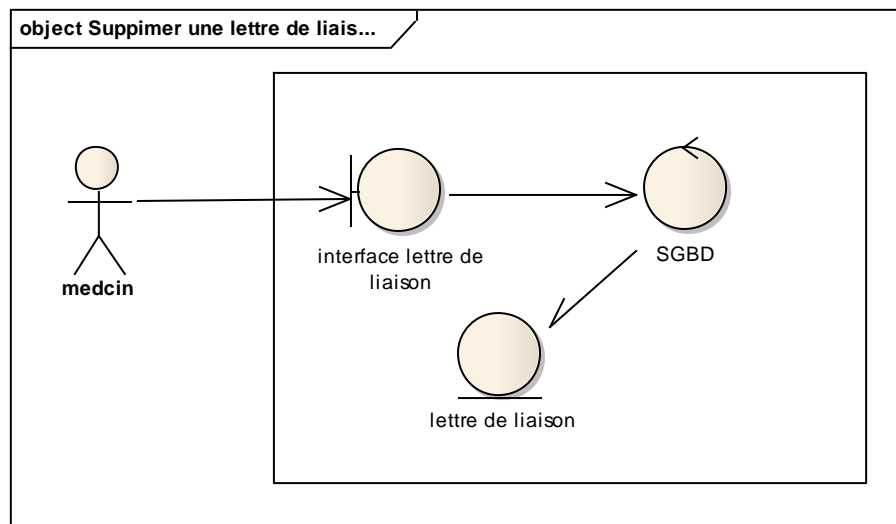


Figure 3-18: Diagramme de classes participantes du cas d'utilisation « Modifier une ordonnance »

La classe « interface ordonnance » : c'est une classe dialogue qui permet de gérer les ordonnances et parmi ses manipulations la modification d'une ordonnance.

La classe « SGBD » : c'est une classe contrôle qui se charge de l'exécution de la requête relative à la modification d'une ordonnance.



**Figure 3-19: Diagramme de classes participantes du cas d'utilisation
« Supprimer une lettre de liaison »**

La classe « interface Lettre de liaison » : c'est une classe dialogue qui permet de gérer les ordonnances et parmi ses manipulations la suppression d'une lettre de liaison.

La classe « SGBD » : c'est une classe contrôle qui se charge de l'exécution de la requête relative à la suppression d'une lettre de liaison.

La classe « Lettre De Liaison » : c'est une classe entité qui désigne la lettre de liaison supprimée.

2.4 Architecture générale du projet :

La figure suivante illustre l'architecture générale de la solution retenue de notre application. En fait, plusieurs cabinets de spécialités différentes peuvent accéder à l'application depuis un navigateur web. Selon des privilèges d'accès définis auparavant par l'administrateur de l'application, le médecin et son secrétaire peuvent ou pas agir sur des points d'accès.

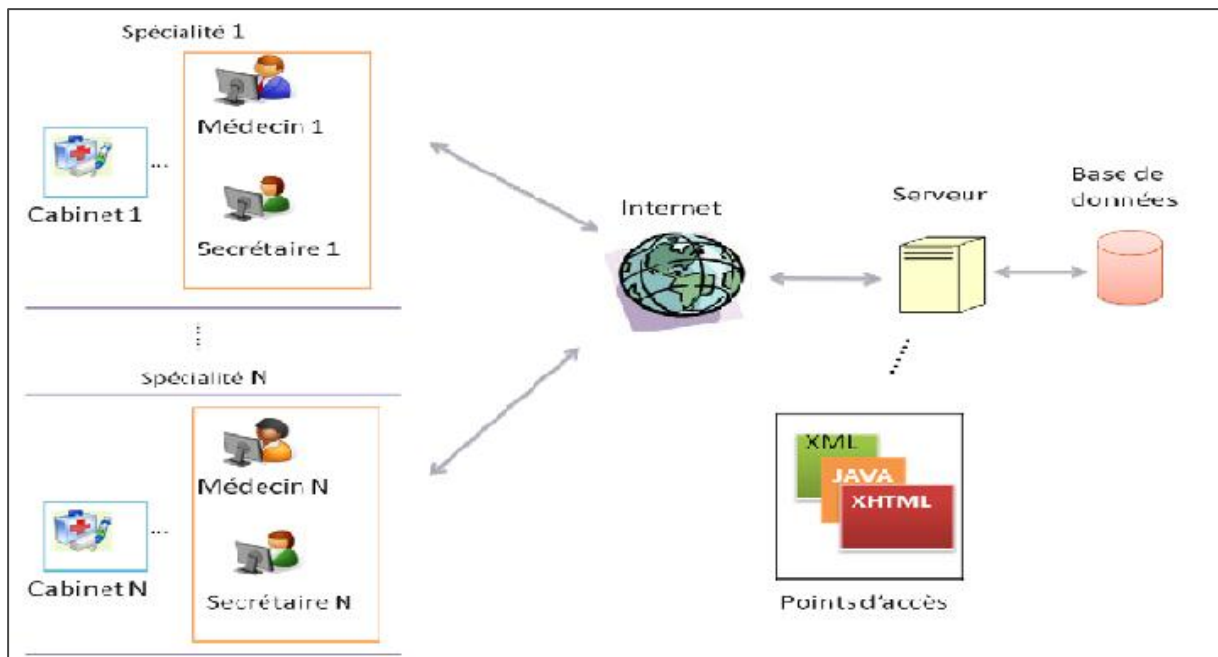


Figure 3-20: Architecture générale du projet

V. Conception :

Cette phase propose une réalisation de l'analyse et des cas d'utilisation en prenant en compte toutes leurs exigences.

1. Conception du système :

Dans cette partie nous allons attribuer des responsabilités précises du comportement, aux classes d'analyse identifiées précédemment et nous allons représenter le résultat de cette étude dans des diagrammes de séquence de conception.

❖ Diagramme de séquence de conception :

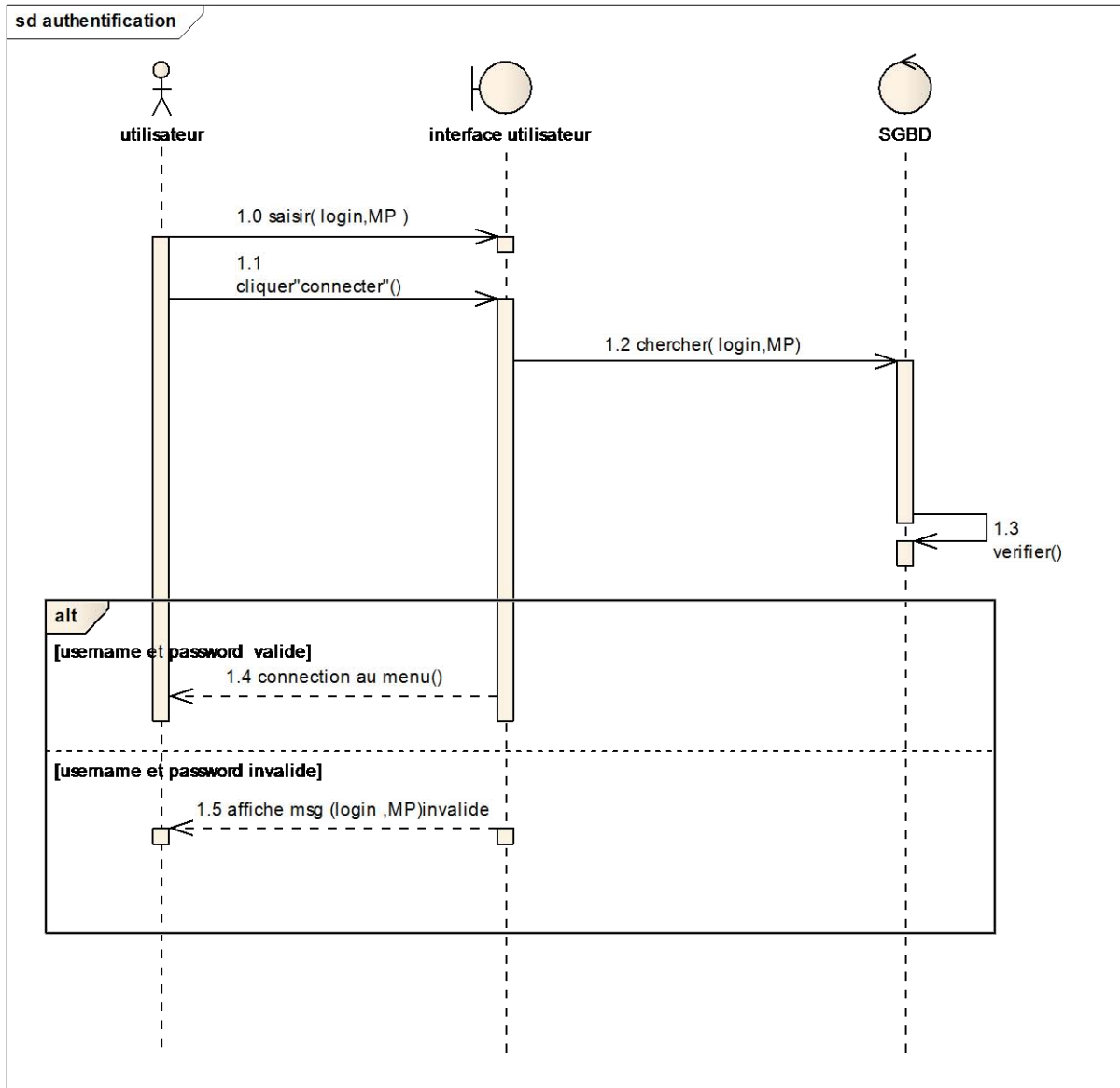


Figure 3-21: Diagramme de séquence « s’authentifier »

Suite à une demande d’accès à l’application, tout utilisateur à savoir le médecin, la secrétaire, l’administrateur ou le patient va passer à une interface d’authentification dans laquelle il va introduire son propre nom d’utilisateur et son mot de passe et selon le résultat de la vérification de la validité ces coordonnées il aura l’accès ou non à l’interface qui le concerne.

La figure suivante montre le diagramme de séquence relatif à l’ajout d’un utilisateur.

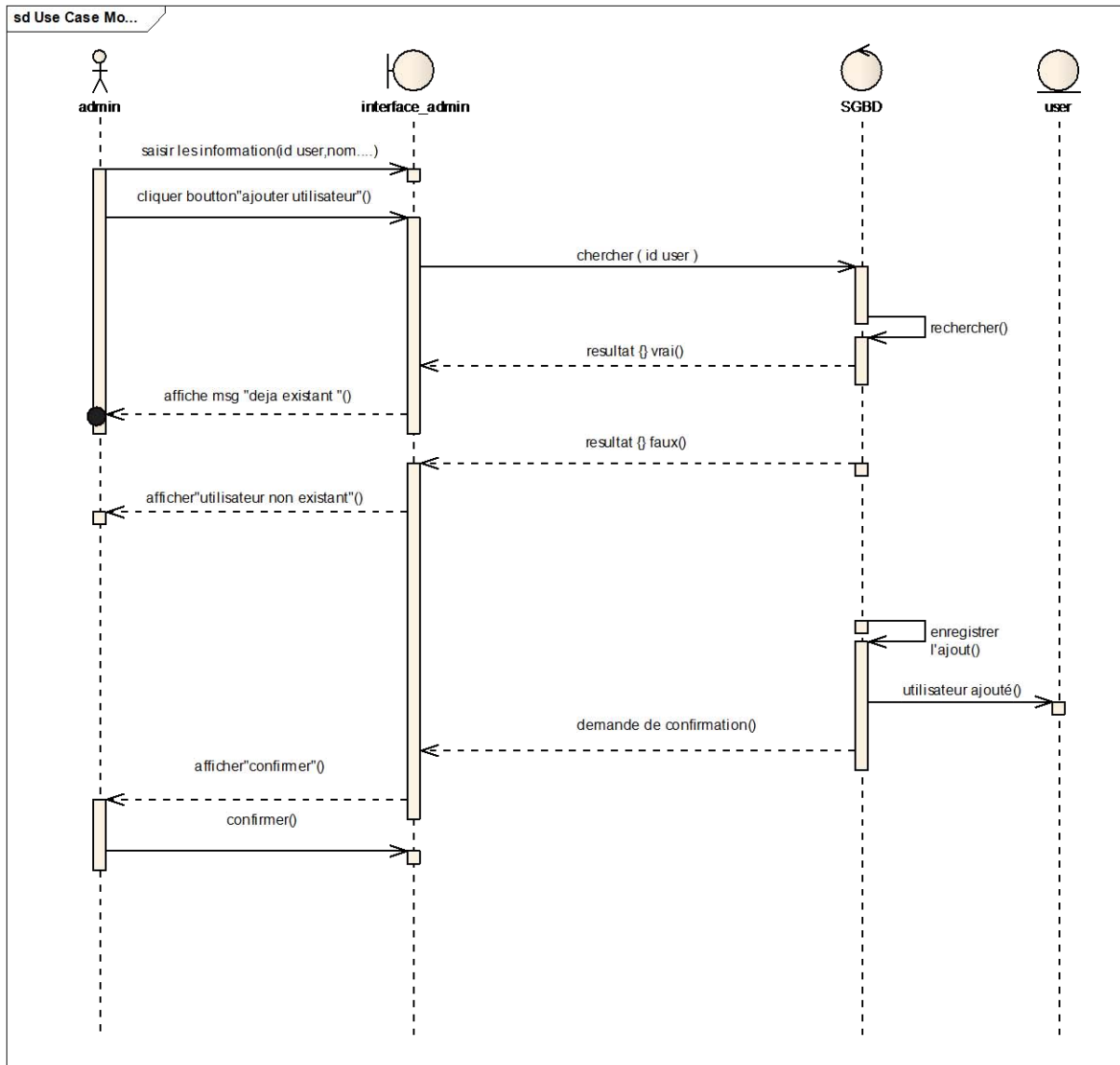


Figure 3-22: Diagramme de séquence « Ajouter un utilisateur »

C'est l'administrateur qui se charge de l'injection d'un nouvel utilisateur dans la base de données. Pour effectuer cette tâche, il accède à son interface, il clique sur le bouton « ajouter un utilisateur ». Le SGBD doit vérifier l'existence de cet utilisateur dans la base et suivant le résultat de la requête, l'utilisateur sera ajouté ou non.

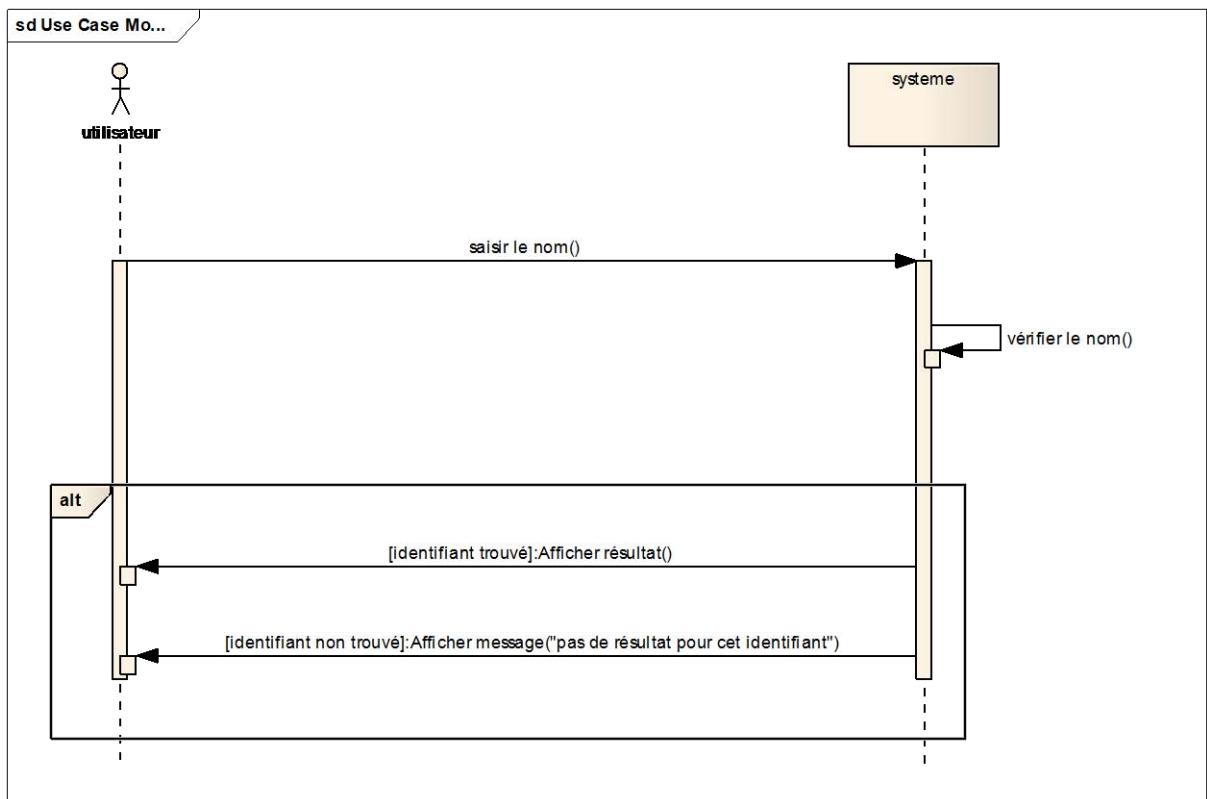


Figure 3-23 : Diagramme de séquence « Effectuer la recherche »

La recherche concerne toutes les tables de l'application, c'est l'administrateur ou bien le médecin qui pourra le faire.

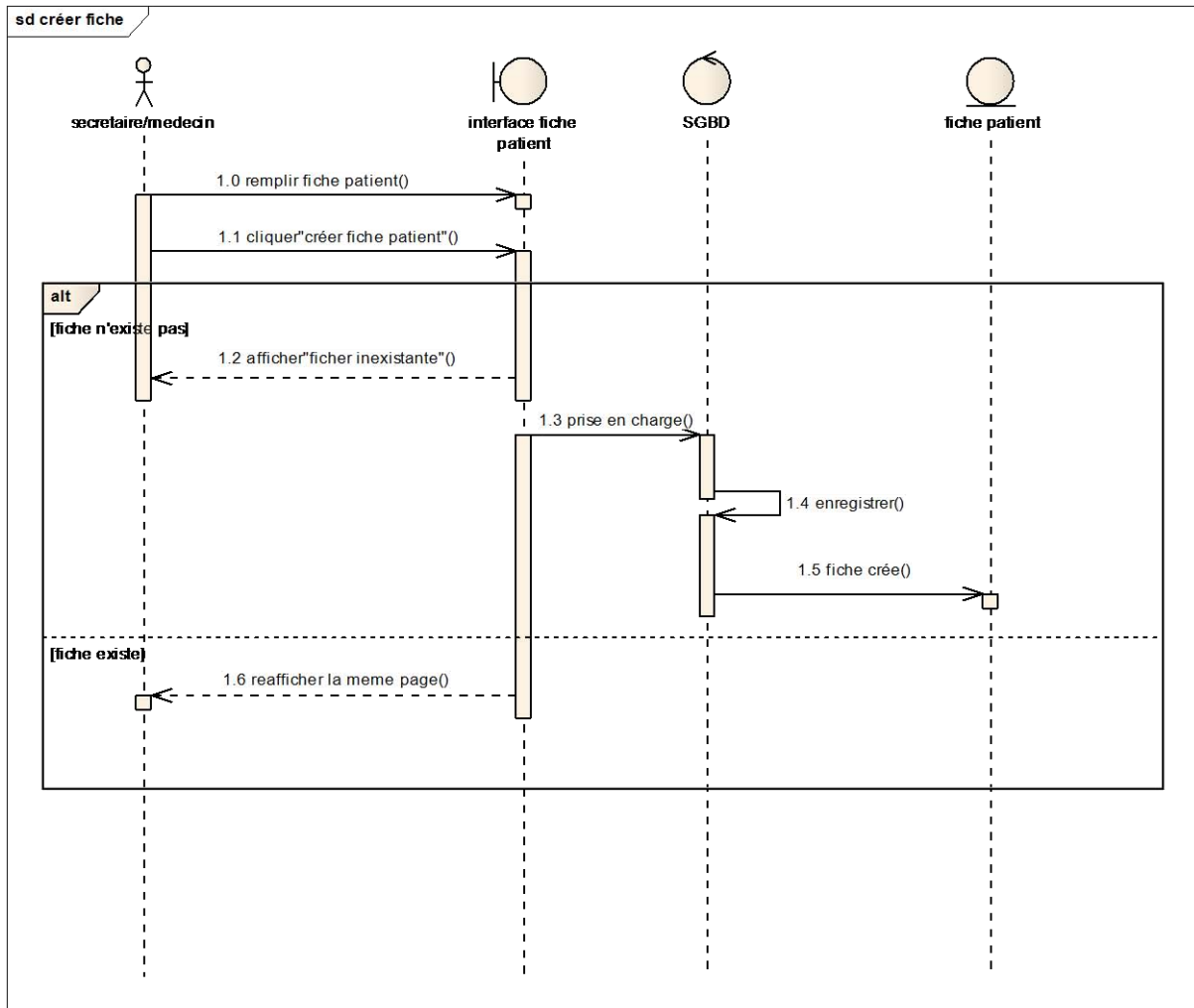


Figure 3-24: Diagramme de séquence « Créer une fiche patient »

Pour ajouter une fiche patient, la secrétaire/le médecin remplit le formulaire affiché sur l'écran. Le système de gestion de base de données (SGBD), se charge de la vérification de l'existence de la fiche patient dans la base de données, et si cette dernière n'existe pas, une fiche patient sera alors créé et sauvegardé.

2. Conception des classes :

❖ Diagramme de classe de conception :

Les diagrammes de classes de conception représentent bien la structure statique du code, par le biais des attributs et des relations entre classes, mais ils contiennent également les opérations (aussi appelées méthodes) qui décrivent les responsabilités dynamiques des classes logicielles. L'attribution des bonnes responsabilités aux bonnes classes est l'un des problèmes les plus délicats de la conception orientée objet. Pour chaque service ou fonction, il faut décider quelle est la classe qui va le contenir.

Nous devons ainsi répartir tout le comportement du système entre les classes de conception, et décrire les collaborations induites.

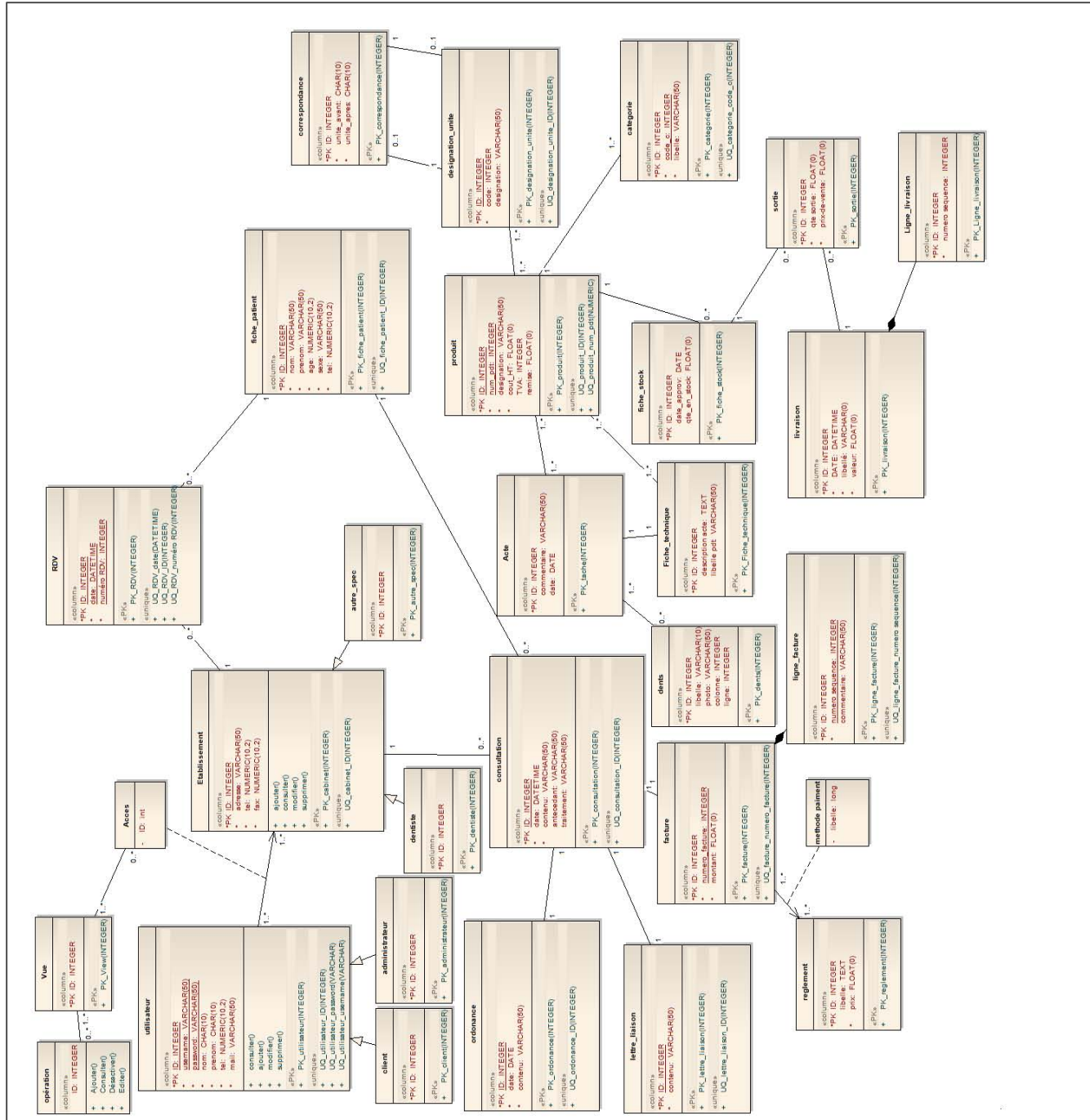


Figure 3-25 : Diagramme de classe

VI. Conclusion :

Ce chapitre m'a permis de présenter la démarche que j'ai suivie pour résoudre le problème en question, que ce soit par la description de la hiérarchie du site ou même par le diagramme des classes présentant le squelette de la base de données. Mais, il reste tout de même la partie concrète; la réalisation qui sera le sujet du chapitre suivant.

Chapitre 4 : Implémentation et réalisation

I. Introduction :

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

Ce chapitre est donc consacré à la présentation de l'environnement matériel et logiciel nécessaire pour implémenter cette structure tout en veillant à garantir les spécificités de sécurité et les services de gestion énoncés au cahier des charges.

II. Outils et langage utilisé :

1. Spécification technique :

Une étape intéressante de ce projet était la mise en place de l'environnement matériel et logiciel nécessaire pour la conception, le développement et le test de l'application. Dans ce qui suit, nous présenterons l'environnement logiciel et matériel exploité dans notre projet.

1.1 Configuration matérielle :

Ordinateur portable : DELL N5232

Système d'exploitation : Windows 7

Processeur : Core I3

Mémoire : 4 G RAM

Disque dur : 320 GO

1.2 Configuration logicielle :

Le développement des applications Web se base sur deux principales technologies, telles que les technologies coté client, et celles du coté serveur.

En ce qui concerne la technologie coté client, elle prend son nom du fait que l'exécution se réalise sur la machine client. Dans ce cadre, on peut citer la technologie qu'on a utilisé qui est le Framework Flex qui est une solution de développement créée par Macromedia en 2004 puis reprise par Adobe en 2006, permettant de créer et de déployer des applications Internet riches (RIA) multiplateformes grâce à la technologie Flash et particulièrement son lecteur. Son modèle de programmation fait appel à MXML (basé sur XML) et Action Script 3.0, reposant sur ECMAScript.

La technologie Flex produit un fichier .swf intégré dans une page html ce qui présente une richesse de l'interface graphique.

En effet, l'environnement de développement est l'Adobe Flash Builder qui est un environnement de développement intégré (IDE, integrated development environment) permettant de créer des applications Internet riches (RIA, rich Internet applications) utilisables sur plusieurs plateformes pour les ordinateurs de bureau et un grand nombre de périphériques mobiles. Il comporte également des outils de test, de débogage et de profilage favorisant productivité et efficacité.

Flash Builder repose sur Eclipse, un IDE à source libre, et fournit tous les outils requis pour développer des applications qui utilisent les structures Flex et ActionScript 3.0 à source libre.

Flash Builder prend intégralement en charge la création d'applications à l'aide du SDK Apache Flex. Lors de la création d'un projet Flex dans Flash Builder, vous peut spécifier l'utilisation du SDK Apache Flex

L'utilisation du Zend Framework est nécessaire, il représente un cadre de travail open source orienté objets pour des applications web basées sur PHP 5. Zend Framework est souvent appelé une "bibliothèque de composants" car ses composants sont faiblement couplés et vous pouvez les utiliser de manière plus ou moins indépendante. Il propose aussi une solution avancée Model-View-Controller (MVC) qui peut être utilisée pour monter des structures de base pour vos applications.

Le serveur d'application est le Zend Server qui est un serveur d'applications Web conçu pour faire fonctionner et gérer des applications PHP critiques en production et en développement.

En outre, le choix pour le serveur de données s'est porté sur MySQL, qui est un système de gestion de base de données, composé d'une collection de relations contenant des attributs, relevant chacun d'un type spécifique et offre de nombreuses fonctionnalités modernes tels que, les requêtes complexes, les déclencheurs, l'intégrité des transactions...

Néanmoins, nous avons eu recours pendant les phases de conception et de développement à un langage de conception appelé UML2, qui est un langage de modélisation permettant de représenter les modes d'utilisation d'un système d'information et de communiquer ses divers aspects, sans s'intéresser à son fonctionnement et aux choix d'implémentation.

En effet, ce dernier a été implémenté sur l'outil de conception et de modélisation Entreprise Architect, qui propose de nombreux outils pour faciliter la gestion des projets de développements. Il permet de répondre aux besoins technologiques et de créer des architectures résilientes en accélérant le rythme de travail.

III. Implémentation :

1. Architecture générale du projet:

L'architecture joue un rôle majeur dans la conception du logiciel.

Elle représente le point de pivot autour duquel pivotent tous les composants de ce logiciel. L'adoption d'une bonne architecture en couches facilite la compréhension des besoins et des fonctions. En effet, ceci assure un niveau d'abstraction qui facilite la compréhension du logiciel puisque les détails ne sont pas représentés. D'une autre part elle assure la réutilisation des composants. De plus, ceci nous permet de dénombrer les composants du logiciel et de modéliser les relations et les interactions entre eux.

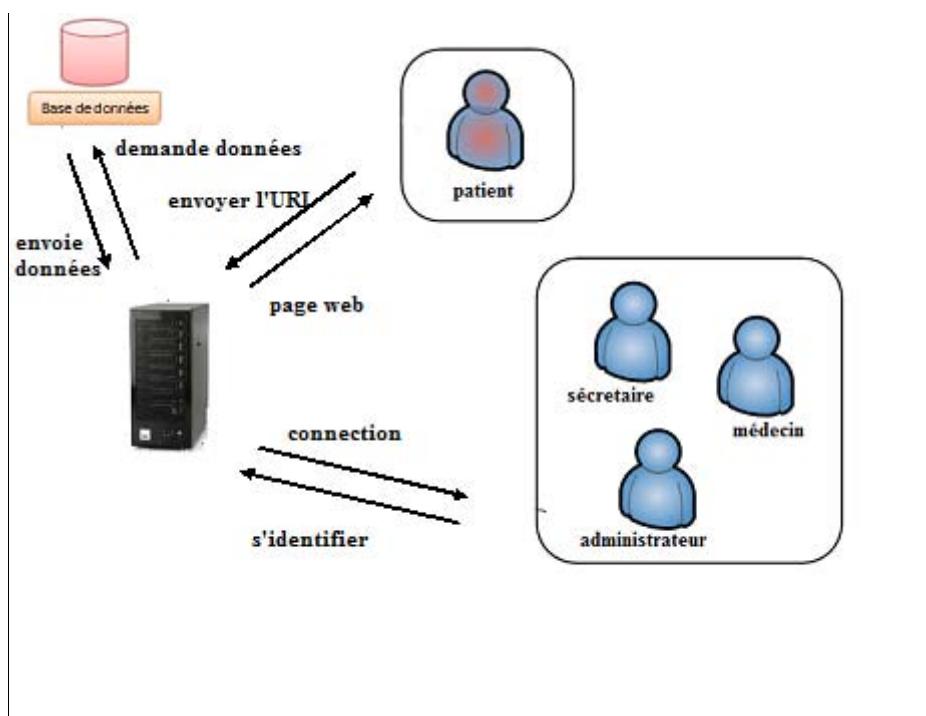


Figure 4-1: Architecture générale du projet

Cette figure représente le fonctionnement de notre système ; les utilisateurs sont interconnectés au serveur d'application Zend Server, qui permet d'exécuter les requêtes envoyées par ces derniers en se connectant à la base de données MySQL qui traite la requête, extrait les données et envoi le résultat de la requête au serveur.

2. Le Diagramme de déploiement :

Le diagramme de déploiement montre la disposition physique des matériels qui compose le système et la répartition de ses composants qui sont les suivants :

- Le serveur d'application est responsable de la gestion des sessions utilisateurs, la gestion des fiches patients, la gestion des factures, gestion des RDV, etc.
- Le serveur de données représente la partie centrale qui gère la base de données et ses accès.
- Le serveur web transmet à l'utilisateur le résultat de la requête envoyé.

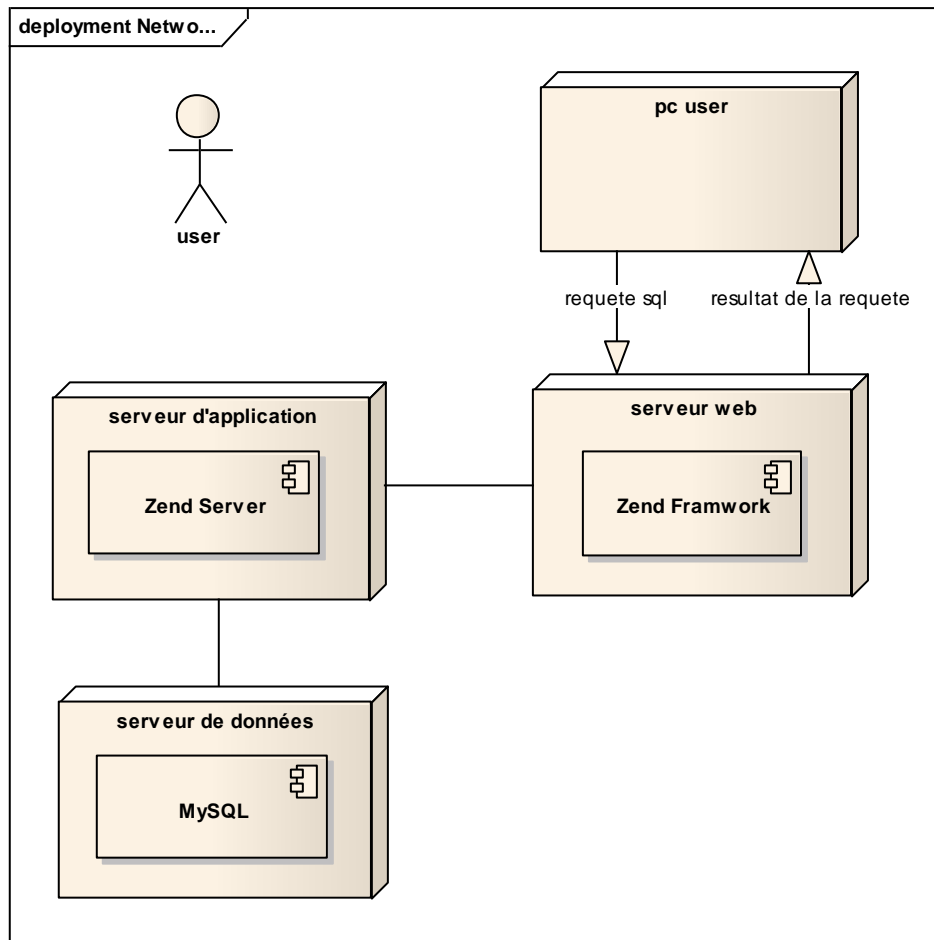


Figure 4-2 : Diagramme de déploiement

IV. Réalisation du projet:

Je vais présenter dans cette partie quelques interfaces utilisateurs de notre application.

❖ Interface d'Authentification

L'authentification est un certificat de sécurité du système, et à partir d'elle chaque utilisateur authentifié peut accéder à notre application. Pour se faire l'utilisateur introduit son nom d'utilisateur, son mot de passe comme le montre la figure ci-dessous.



Figure 4-3 : Interface « authentification »

❖ **Le Menu :**

Suite à l'authentification un menu qui s'affiche qui nous permet de choisir l'interface qu'on souhaite y accéder :

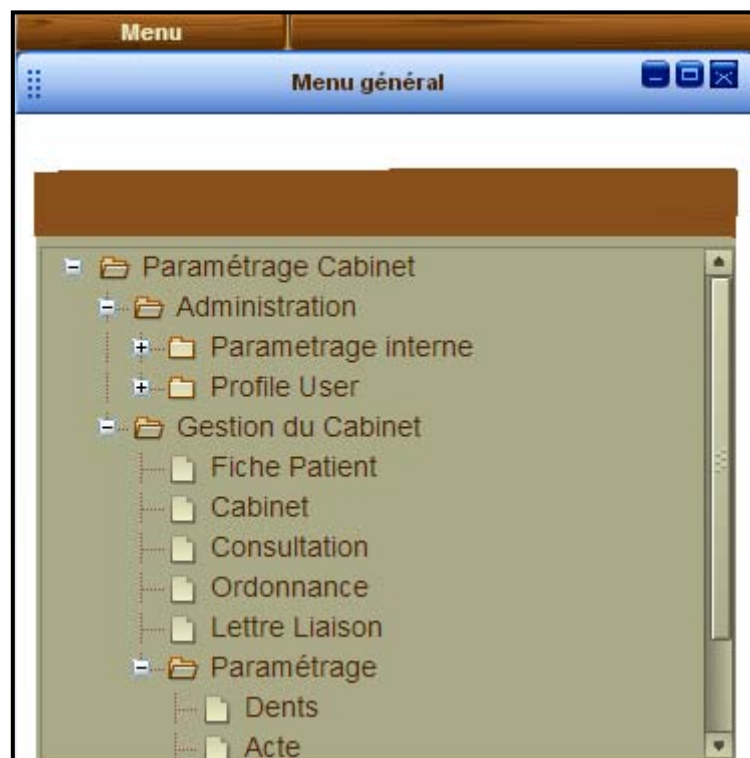


Figure 4-4 : Interface « Menu »

❖ Interface « créer une fiche patient » :

Pour créer une fiche du patient, le médecin ou bien sa secrétaire va remplir le formulaire détaillé dans la Figure suivante et valide la transaction.

| | | | |
|---------|-----------|--------------------|---------|
| ID | 0 | Telephone | 45896 |
| Nom | ben salah | Email | sebgire |
| Prenom | saleh | CNAM | |
| Adresse | benarous | Profession | |
| Age | 45 | Motif consultation | |
| Sexe | homme | | |

Add

Figure 4-5 : « Interface Créer une fiche patient »

En cliquant sur le bouton « Add », la fiche patient sera ajoutée à la liste :

| ID | Numero fiche | Nom | Prenom | Age | Sexe | Adresse | Telephone | Email | CNAM | Profession | motifconsultation |
|----|--------------|-----------|--------|-----|-------|-----------|-----------|------------|--------|------------|-------------------|
| 4 | 13 | ben salah | ali | 50 | homme | rue tunis | 458123 | albensalah | 789954 | prof | malade |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |

Ajout **Supprimer** **Modifier**

Figure 4-6 : Interface « La liste des patients »

❖ Interface « Paramétrage dents »

Vu qu'il s'agit d'un cabinet de dentiste, la fenêtre Dent permet d'afficher le dent concerné ainsi que son emplacement (ligne et colonne)

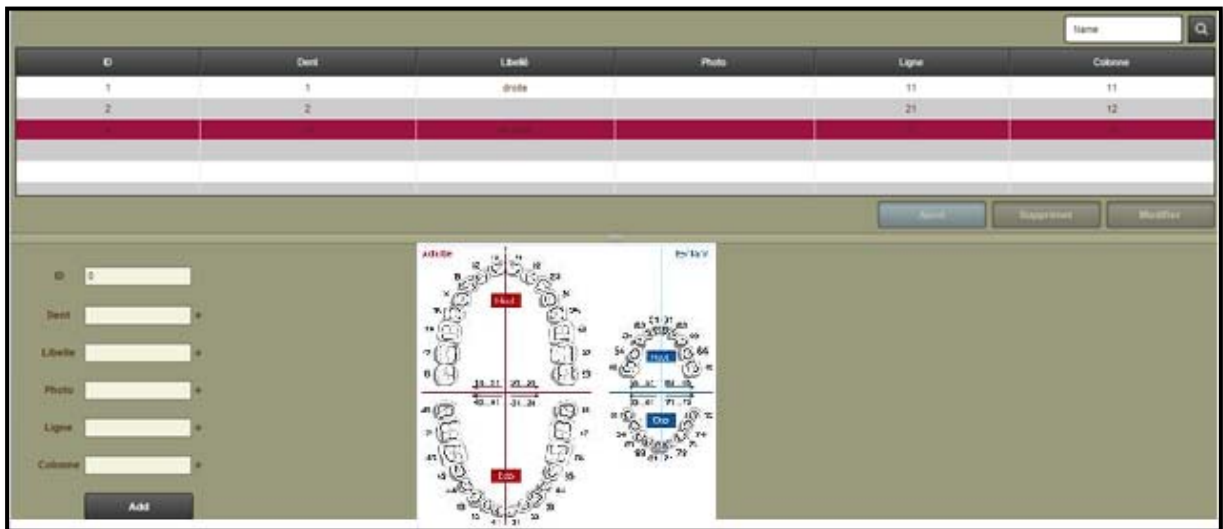


Figure 4-7: Interface « Paramétrage des dents »

❖ **Interface « gérer les consultations »**

Pour vérifier la consultation de chaque patient ou bien ajouter ou supprimer, il suffit de cliquer sur la fenêtre Consultation :

ID

Numero consultation

Contenu

Antecedant

Traitement

Date consultation

Add

Figure 4-8 : « Interface Créer une consultation »

| ID | Numero consultation | Contenu | Antecedant | Traitement | Date Consultation |
|----|---------------------|------------|------------|------------|-------------------|
| 1 | 8 | ordonnance | rien | antibio | 05/11/2013 |
| | | | | | |
| | | | | | |

Figure 4-9 : Interface « La liste des consultations »

V. Conclusion :

Dans ce chapitre, j'ai présenté les différentes techniques utilisées pour implémenter le système ainsi que, les principes généraux d'ergonomie qui nous ont aidés à fournir des interfaces conviviales et qui permettent aux utilisateurs de réaliser leurs tâches d'une manière plus efficace.

Conclusion générale

J'ai conçu et développé une application SAAS pour la gestion du cabinet d'un médecin et spécialement le cas d'un dentiste au sein de la société « Hotix Tunisie ».

Ce projet m'a permis de m'intégrer dans un environnement de travail purement professionnel.

Par ailleurs, l'interaction avec les membres de l'équipe m'a permis de mieux comprendre les spécificités de l'application réalisée.

Cependant, La réalisation de ce projet a nécessité la mise en œuvre de plusieurs technologies afin d'assurer la performance et la fiabilité de l'application, dont la réalisation a nécessité une méthode de conception appelée la méthode agile.

Ainsi, comme perspective principale pour l'état actuel de ce projet, je souhaite ajouter système qui sert à la gestion de stock (contenant les devis, les bon de commande, les fournisseurs..)

Bibliographie

[1] Pascal Roques «"UML2, modéliser une application web édition, Eyrolles dans la collection, Les Cahiers du Programmeur", 2006 »

Netographie

www.adobe.com

www.developer.com

www.wikipedia.com

www.commentcamarche.net