

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A. Mira de Bejaïa
Faculté des Sciences Exactes
Département d'informatique



Mémoire de fin d'études

En vue de l'obtention du diplôme de Master en Informatique professionnel

Option : Génie Logiciel (GL)

Thème

Conception et réalisation d'une application de gestion d'un hôpital

Présenté par :

DEBBOU Yanni & BOUCHAL Juba

Encadré par:

GADOUCHE Hania

Devant le jury composé de:

Présidente : Mme AIT HACENE Souhila

Examineur : M.FARAH Zoubeyr

Remerciements

Ce mémoire est le fruit des efforts fournis et des sacrifices consentis par plusieurs personnes que je ne pourrai oublier de remercier.

Mes remerciements s'adressent d'abord à dieu, créateur de toutes choses, pour son souffle et tous ses innombrables bienfaits.

Je remercie aussi mon encadrante de mémoire, Mme GADOUCHE Hania d'avoir accepté de nous encadrer dans la conception et réalisation de ce travail, et aussi pour son dévouement et soutien incontestable et continu.

Je me dois de remercier plus particulièrement mes parents, mon frère ainsi que ma femme pour leur soutien et aide inconditionnellement.

Je tiens aussi à remercier tous mes amis, et plus spécialement Amine Haddid qui n'ont, à aucun moment cessé de m'encourager et de me pousser vers l'avant.

DEBBOU Yanni

Remerciement

*Avant tout commencement je remercie
dieu le tout puissant de nous avoir donné
la force, la patience et le courage durant
toutes ses années d'études.*

*Je tiens à exprimer mes remerciements
très spéciales à mon encadrante Dr
GADOUCHE Hania, pour ses conseils, sa
disponibilité et ses encouragements qui
m'ont permis de réaliser ce travail dans
les meilleures conditions.*

*Je remercie aussi tous les enseignants
qui m'ont enseigné et encadré durant
toutes ces années.*

*Je remercie aussi tous mes chers amis et
mes proches qui ont été à mes côtés tout
ce temps.*

BOUCHAL Juba

Liste des figures

Figure II.5-2-1: Diagramme cas d'utilisations «Patient».	11
Figure II.5-2-2: Diagramme de cas d'utilisation «Secrétaire».	12
Figure II.5-2-3: : Diagramme de cas d'utilisation «Médecin».	12
Figure II.5-2-4: Diagramme de cas d'utilisation «Administrateur».	13
Figure II.7-2: : Diagramme séquence «Authentification»	20
Figure II.7-3: Diagramme séquence «Prendre un rendez-vous»	21
Figure II.7-4: Diagramme séquence «Confirmer un rendez-vous»	21
Figure II.7-5: Diagramme séquence «Annuler un rendez-vous»	22
Figure II.7-6: Diagramme séquence «Confirmer une demande d'hospitalisation»	23
Figure II.7-7: Diagramme séquence «Annuler une demande d'hospitalisation»	23
Figure II.7-8: Diagramme séquence «Mettre à jour la fiche patient»	24
Figure II.7-9: Diagramme séquence «Ajouter un médecin»	25
Figure II.7-10: Diagramme séquence «Ajouter un(e) secrétaire»	26
Figure II.7-11: Diagramme séquence «Ajouter un service»	27
Figure II.7-12: Diagramme séquence «Ajouter une chambre»	28
Figure II.8-2: Diagramme de classes	29
Figure III.4-1: Page d'accueil.	35
Figure III.4-2: Prise de rendez-vous.	35
Figure III.4-3: Historique des rendez-vous du patient	36
Figure III.4-4: Liste des médecins du service	36
Figure III.4-5: Historique des rendez-vous du service	37
Figure III.4-6: Historique des hospitalisations	37
Figure III.4-7: Liste des rendez-vous	38
Figure III.4-8: Mise à jour de la fiche patient (Diagnostic et ordonnance).	38
Figure III.4-9: Fiches patient	39

Liste des tableaux

Tableau 1: Description textuelle du cas d'utilisation «Authentification»	14
Tableau 2: Description textuelle du cas d'utilisation «Prendre un rendez-vous»	15
Tableau 3: Description textuelle du cas d'utilisation «Confirmer rendez-vous»	15
Tableau 4: Description textuelle du cas d'utilisation «Annuler un rendez-vous»	16
Tableau 5: Description textuelle du cas d'utilisation «Confirmer un demande d'hospitalisation».	16
Tableau 6: Description textuelle du cas d'utilisation «Annuler une demande d'hospitalisation»	17
Tableau 7: Description textuelle du cas d'utilisation «Mettre à jour la fiche patient»	17
Tableau 8 : Description textuelle du cas d'utilisation «Ajouter un médecin»	18
Tableau 9: Description textuelle du cas d'utilisation «Ajouter un(e) secrétaire»	18
Tableau 10: Description textuelle du cas d'utilisation «Ajouter un service»	19
Tableau 11: Description textuelle du cas d'utilisation «Ajouter une chambre»	19

SOMMAIRE

Chapitre 1 Etude préliminaire et contexte du projet	3
I.1 Introduction	3
I.2 Identification de la problématique	3
I.3 Solutions proposées	3
I.4 Besoins fonctionnels	4
I.5 Besoins non-fonctionnels	5
I.6 Méthodologie de travail	6
I.6.1 Définition	6
I.6.2 Etapes de scrum	6
I.7 Langage de modélisation	7
I.8 Conclusion	7
Chapitre 2 Conception et modélisation	8
II.1 Introduction	8
II.2 Définition des acteurs	8
II.2.1 Le patient	8
II.2.2 La secrétaire de service	8
II.2.3 Le médecin	9
II.2.4 L'administrateur	9
II.3 Identification des cas d'utilisations	9
II.3.1 Patient	9
II.3.2 Secrétaire	9
II.3.3 Médecin	10
II.3.4 Administrateur	10
II.4 Cas d'utilisations	10
II.5 Diagrammes cas d'utilisations	11
II.5.1 Définition	11
II.5.2 Diagrammes cas d'utilisations des différents acteurs	11

II.5.2.1 Diagramme cas d'utilisation du Patient	11
II.5.2.2 Diagramme cas d'utilisation Secrétaire	12
II.5.2.3 Diagramme cas d'utilisation Médecin	12
II.5.2.4 Diagramme cas d'utilisation Administrateur	13
II.6 Descriptions textuelles des différents cas d'utilisations	14
II.6.1 Authentification	14
II.6.2 Prendre un rendez-vous	15
II.6.3 Confirmer un rendez-vous	15
II.6.4 Annuler un rendez-vous	16
II.6.5 Confirmer une demande d'hospitalisation	16
II.6.6 Annuler une demande d'hospitalisation	17
II.6.7 Mettre à jour la fiche patient	17
II.6.8 Ajouter un médecin	18
II.6.9 Ajouter un(e) secrétaire	18
II.6.10 Ajouter un service	19
II.6.11 Ajouter une chambre	19
II.7 Diagrammes de séquence	20
II.7.1 Définition	20
II.7.2 Diagramme séquence cas «Authentification»	20
II.7.3 Diagramme séquence cas «Prendre un rendez-vous»	21
II.7.4 Diagramme séquence cas «Confirmer un rendez-vous»	21
II.7.5 Diagramme séquence cas «Annuler un rendez-vous»	22
II.7.6 Diagramme séquence cas «Confirmer une demande d'hospitalisation»	23
II.7.7 Diagramme séquence cas «Annuler une demande d'hospitalisation»	23
II.7.8 Diagramme séquence cas «Mettre à jour la fiche patient»	24
II.7.9 Diagramme séquence cas «Ajouter un médecin»	25
II.7.10 Diagramme séquence cas «Ajouter un(e) secrétaire»	26
II.7.11 Diagramme séquence cas «Ajouter un service»	27
II.7.12 Diagramme séquence cas «Ajouter une chambre»	28
II.8 Diagramme de classes	29
II.8.1 Définition	29
II.8.2 Diagramme de classes	29
II.8.3 Modèle relationnel	30

II.8.4 Règles de passage au modèle relationnel	30
II.8.5 Le modèle relationnel	30
II.9 Conclusion	31
Chapitre 3 Présentation du site web	32
III.1 Introduction	32
III.2 Application web	32
III.3 Outils de réalisation	32
III.3.1.Réalisation des diagrammes	32
III.3.2 Réalisation de l'application web	33
III.3.2.1 PHP	33
III.3.2.2 JS	33
III.3.2.3 HTML	33
III.3.2.4 CSS	34
III.3.2.5 XAMPP	34
III.3.2.6 PMA	34
III.3.2.7 VS Code	34
III.4 Interfaces de l'application web	
Erreur ! Signet non défini.	
III.4.1 Page d'accueil	35
III.4.2 Compte patient : Prise de rendez-vous	35
III.4.3 Compte patient : Historique des rendez-vous	36
III.4.4 Compte secrétaire: Liste des médecins du service	36
III.4.5 Compte secrétaire: Historique des rendez-vous	37
III.4.6 Compte secrétaire: Historique des hospitalisations	37
III.4.7 Compte médecin: Liste des rendez-vous	38
III.4.8 Compte médecin: Mise à jour de la fiche patient	38
III.4.9 Compte médecin: Fiches patient	39
III.5.Conclusion	39
Conclusion générale	40
Références bibliographiques	41

Liste des abréviations :

- HTML : HyperText Markup Language.
- CSS : Cascading Style Sheets.
- PHP : Hypertext Preprocessor.
- JS : JavaScript.
- PMA : PhpMyAdmin.
- VS Code : Visual Studio Code.

Introduction générale

L'informatique de gestion est l'ensemble des connaissances, des technologies, et des outils en rapport avec la gestion de données.

« L'informatique de gestion constitue, avec les utilisations militaires, industrielles et scientifiques, l'un des domaines essentiels d'application qui ont permis le développement rapide de l'informatique. On peut même dire que l'informatique de gestion est en grande partie à l'origine des méthodes modernes de conception et de réalisation.

À partir des années 1970, des firmes comme IBM, Honeywell et Bull par exemple, ont conçu des machines et aussi des programmes d'abord destinés aux grandes organisations. Grâce à leur succès croissant, ces fabricants ont diversifié leur offre en même temps que se sont constituées des compagnies concurrentes. Le COBOL est rapidement devenu le principal langage de programmation pour les applications de gestion et a permis de constituer des milliers de programmes sur mesure, tout en étendant l'utilisation de l'informatique à tous les domaines de la gestion. » [1]

Bien avant l'introduction de l'informatique au monde, tout se faisait de façon manuelle, que ce soit la gestion des stocks, la gestion des ressources, la gestion des entrées/sorties, les prises de rendez-vous... etc. L'arrivée de l'informatique et l'automatisation de beaucoup de ces tâches fut comme une bénédiction, car cela a changé beaucoup de choses, cela a permis, notamment, de réduire les coûts de temps et d'énergie, faciliter l'accès aux informations et données, conserver des données importantes. Tout ceci a permis de réduire fortement la marge d'erreur humaine.

Le domaine médical n'est pas une exception, en vue des avancées et innovations que ce domaine a connu grâce à l'informatique (mais pas seulement l'informatique), que ce soit en matériel de diagnostics, d'opérations, de suivis, et j'en passe. Cela a donc, nécessairement, affecté la gestion hospitalière, notamment la prise de rendez-vous, qui maintenant, se fait en ligne. Les rapports médicaux et remarques du médecin. La gestion des hospitalisations au sein de l'établissement médical, l'établissement des factures, les suivis de paiements...etc.

Notre projet consiste à concevoir et réaliser une application web de gestion d'un hôpital, et pour ce faire, nous allons nous concentrer sur les points suivant :

- Gestion des rendez-vous (Prise de rendez-vous, avec confirmation/annulation par la(e) secrétaire du service).
- Etablissement d'un diagnostic patient par le médecin.
- Gestion des hospitalisations (Confirmation/annulation de la demande d'hospitalisation du médecin par la(e) secrétaire du service).
- Ajout/suppression des comptes médecin/secrétaire par un administrateur qui définira les accès.
- Accès du patient au rapport du diagnostic du médecin post rendez-vous.

Pour pouvoir réaliser et mener ce projet à terme, nous allons diviser notre projet en 3 chapitres principaux, qui seront réparties comme ceci :

- 1- Chapitre 1 : Etude préliminaire et contexte du projet.
- 2- Chapitre 2 : Conception et modélisation.
- 3- Chapitre 3 : Présentation de l'application web.

Chapitre 1

Etude préliminaire et contexte du projet

1- Introduction :

Dans ce premier chapitre, nous allons nous intéresser à l'étude préliminaire pour la définition du contexte du projet, nous allons parler de la problématique soulevée, et des solutions que nous avons définies et jugées comme utiles pour y remédier.

2- Identification de la problématique :

Le problème majeur au sein de la plus part de nos hôpitaux réside dans la difficulté de la gestion des rendez-vous patients, impossibilité de prendre un rendez-vous de façon convenable, obligation de déplacement physiquement rien que pour réserver un rendez-vous. Tout ceci crée un environnement négatif pour le patient, mais pas seulement le patient, ceci affecte aussi l'aspect moral du personnel médical, du résultat d'une charge énorme au sein de l'établissement.

3- Solutions proposées :

Notre projet consiste à concevoir et réaliser une application web visant à informatiser la gestion hospitalière centrée patient. Nous allons nous concentrer sur la gestion centrée patient, avec comme fonctionnalités principales :

- Gestion des rendez-vous.
- Gestion de fiches patients.
- Gestion des hospitalisations.

- Gestion des rendez-vous :

Cette fonctionnalité permet aux patients (inscrits) d'avoir la possibilité de demander un rendez-vous au sein d'un des services de l'hôpital. Ce dernier sera confirmé ou annulé, selon la disponibilité du médecin du service.

Le patient pourra, à tout moment, consulter l'historique de ses rendez-vous, et peut donc savoir si son rendez-vous a été confirmé ou annulé ou encore en attente.

- Gestion de fiches patients :

L'établissement du diagnostic est une fonctionnalité réservée au médecin, il est le seul à pouvoir établir un diagnostic au patient après une consultation. Si le diagnostic est négatif, le médecin peut demander une hospitalisation en indiquant le motif de cette dernière. Cela permettra notamment de mettre à jour la fiche du patient.

- Gestion des hospitalisations :

Cette fonctionnalité est réservée à la(e) secrétaire du service. C'est la(e) responsable d'hospitalisation d'un patient. Ce dernier a une visibilité sur les chambres du service disponible, il pourra ainsi hospitaliser le patient de façon facile et rapide.

4- Besoins fonctionnels :

➤ Définition :

Les besoins fonctionnels sont ceux qui précisent ce que le système doit faire. En d'autres termes, ils spécifient une fonction, un comportement ou une action que le système doit exécuter. [2]

Les fonctionnalités citées plus haut seront celles qui se traduiront en besoins fonctionnels, mais ce ne sont pas les seules. Elles sont comme suit :

1- Gestion des rendez-vous :

- Offre au patient la possibilité de demander un rendez-vous.
- Offre au secrétaire de service la possibilité de confirmer/annuler un rendez-vous.

2- Gestion de fiches patient :

- Mettre à jour les fiches patient suite aux diagnostics.

- 3- Hospitalisation d'un patient :
 - Possibilité d'hospitaliser un patient, et avoir un suivi global sur les hospitalisations.
- 4- Créer des comptes médecin/secrétaire :
 - Possibilité d'ajouter des comptes médecin/secrétaire par l'administrateur.
- 5- Ajouter des services :
 - Possibilité d'introduire un nouveau service.

5- **Besoins non-fonctionnels :**

➤ Définition :

Les besoins non-fonctionnels décrivent comment le système doit fonctionner, ils précisent les critères selon lesquels un système est jugé, plutôt que des comportements spécifiques. [2]

Nous pouvons en citer quelques-uns ci-dessous :

- Ergonomie :
Interface simple d'utilisation, intuitive et facile à comprendre.
- Fiabilité :
Capacité de mettre l'utilisateur en confiance vis-à-vis du système.
- Sécurité :
Le système doit assurer une certaine sécurité aux utilisateurs.
- Authentification :
Capacité de s'authentifier en tant que l'entité choisie (Patient, médecin, secrétaire, administrateur).

6- Méthodologie de travail :

Nous avons jugé, selon la nature de notre travail, que la méthode agile SCRUM est la mieux adaptée pour le projet que nous entreprenons.

6-1.Définition :

Scrum est une des méthodes de gestion de projet Agile. En tant que telle, son objectif est d'améliorer la productivité des équipes agiles même à distance, tout en permettant une optimisation du produit grâce à des feedbacks réguliers avec les utilisateurs finaux. [3]

6-2.Etapas de scrum :

Étape 1 : le Product Backlog

Durant cette première étape essentielle, le Product Owner va rencontrer le client (le donneur d'ordre du projet) et analyser ses besoins. L'enjeu est d'identifier toutes les fonctionnalités à implémenter dans le produit (en partant des user stories). Le tout est formalisé dans le Product Backlog, un document qui récapitule les caractéristiques du futur produit et les tâches à effectuer par l'équipe produit.[4]

Étape 2 : le Sprint

L'équipe produit va trier l'ensemble des fonctionnalités et des tâches afin de les répartir dans des Sprints : des cycles d'une durée de deux semaines (dans la majorité des cas). Durant chaque Sprint, les membres de l'équipe se focalisent uniquement sur des tâches bien précises, comme le codage d'une fonctionnalité qui sera livrée à la fin de ce cycle.

Chaque Sprint a son Sprint Planning Meeting. C'est-à-dire une réunion entre le Product Owner et l'équipe technique qui sert à prioriser les tâches dans le Product Backlog en fonction des exigences du client. Après avoir analysé ensemble les user stories, l'ordre des tâches à réaliser durant un même Sprint est formalisé par un Sprint Backlog.[4]

Étape 3 : le Daily SCRUM

C'est l'essence même des projets en méthodologie SCRUM. Quand l'équipe est au complet, chaque matin a lieu une courte

réunion (de 5 à 10 minutes). Pendant le Daily Scrum, l'équipe reste debout, afin d'être sûr d'aller à l'essentiel.[4]

Étape 4 : le Sprint Review

À chaque fin de semaine (bien souvent le vendredi) qui close un sprint, l'équipe teste les bénéfices de la fonctionnalité avec le Product Owner et le client. Ce dernier se prononce à la fin de la démonstration : objectif réalisé ou non.[4]

7- Langage de modélisation :

Notre modélisation des solutions proposées se fera en utilisant des diagrammes UML.

- **Définition :**

Le Langage de Modélisation Unifié, de l'anglais *Unified Modeling Language* (UML), est un langage de modélisation graphique à base de pictogrammes conçu comme une méthode normalisée de visualisation dans les domaines du développement logiciel et en conception orientée objet.[5]

8- Conclusion :

Dans ce premier chapitre, nous avons parlé de quelques problèmes récurrents que rencontrent les patients et le personnel médical, nous avons identifié la problématique, et avons par la même occasion proposé quelques solutions.

Nous avons aussi déterminé les besoin fonctionnels des non-fonctionnels, pour ainsi clôturer ce premier chapitre de notre mémoire.

Chapitre 2

Conception et modélisation

1- Introduction :

Nous avons finalisé le premier chapitre en mentionnant les points importants de notre projet. Maintenant, nous allons entamer notre deuxième chapitre, ou nous allons nous intéresser aux acteurs du système, bien les définir, ainsi que détailler le rôle de chacun.

2- Identification des acteurs :

Définition d'un acteur :

En résumé, un acteur est un utilisateur interagissant avec le système, on peut donc dire qu'un acteur représente l'utilisateur de notre système, celui avec qui notre environnement système interagit continuellement.

Dans notre projet, nous allons avoir 4 acteurs :

2-1. Le patient :

En médecine, un patient est une personne physique recevant une attention médicale ou à qui est prodigué un soin. [6]

Dans notre cas, un patient est le demandeur de service, plus spécifiquement, le demandeur de rendez-vous.

2-2. La secrétaire de service :

Assistant du praticien, le(a) secrétaire médical organise la vie du cabinet dans lequel il travaille et soulage le médecin dans les tâches qu'il n'a pas le temps d'effectuer.

Dans notre cas, le rôle du secrétaire est de gérer l'aspect technique, tel que la confirmation/annulation de rendez-vous selon la disponibilité du(es) médecins, répondre aux demandes d'hospitalisation émises par le médecin. [7]

2-3. Le médecin :

Personne qui est titulaire du diplôme de docteur en médecine.
Son rôle dans notre système, contrairement à la vie réelle, est un peu limité, il a la capacité de rédiger un diagnostic pour un patient selon le rendez-vous du patient. Il peut aussi demander l'hospitalisation du dit-patient selon le résultat du diagnostic.

2-4. L'administrateur :

Un administrateur de manière globale, est l'entité qui gère les accès et les autorisations.

Il a la possibilité d'ajouter des médecins, secrétaires, services, chambres, et peut consulter la liste des patients inscrits. Il n'a aucun rôle en vue des actions à prévoir sur les patients.

3- **Identification des cas d'utilisations :**

Nous avons identifié, dans notre système, les cas d'utilisations suivants, classés en fonction des acteurs.

Il est à noter que tout acteur de notre système nécessite une authentification, et pour le patient, une inscription est d'abord nécessaire.

3-1. Cas d'utilisations associés au Patient :

Les cas d'utilisations du patient se résument à :

- Prendre un rendez-vous.
- Consulter l'historique des rendez-vous.

3-2. Cas d'utilisations associés au Secrétaire :

Les cas d'utilisations du (la) secrétaire se résument à :

- Confirmer/Annuler un rendez-vous.
- Confirmer/Annuler une demande d'hospitalisation.
- Consulter la liste des médecins du service d'affectation.
- Consulter l'historique des rendez-vous du service d'affectation.
- Consulter la liste des chambres disponibles du service d'affectation.
- Consulter la liste des patients traités du service d'affectation.

- Consulter l'historique des hospitalisations du service d'affectation.

3-3. Cas d'utilisations associés au Médecin :

Les cas d'utilisations du médecin se résument à :

- Etablir un diagnostic ou une ordonnance (Avec possibilité de demander une hospitalisation).
- Consulter les suivis des patients traités.

3-4. Cas d'utilisations associés à l'Administrateur :

Les cas d'utilisations de l'administrateur se résument à :

- Consulter la liste globale des médecins.
- Consulter la liste globale des patients.
- Consulter la liste globale des rendez-vous.
- Ajouter un médecin.
- Ajouter un(e) secrétaire.
- Ajouter des services.
- Ajouter des chambres.

4- Diagrammes de Cas d'utilisations :

Nous pouvons regrouper les cas d'utilisations par acteur, cités plus haut en quelques catégories :

- Authentification :
Fonctionnalité permettant aux différents acteurs d'accéder à leurs comptes respectifs.
- Gestion des rendez-vous :
Fonctionnalité permettant aux patients de demander un rendez-vous, au secrétaire de confirmer/annuler ce dernier, et au médecin d'établir un diagnostic.
- Gestion du personnel :
Fonctionnalité permettant à l'administrateur système de gérer le personnel (Attribue les autorisations à chaque membre du personnel médical).
- Gestion des chambres et services :
Fonctionnalité permettant à l'administrateur système d'ajouter des services et des chambres et affecter ces chambres aux services existants.

5- Diagrammes cas d'utilisations :

5-1. Définition :

Les diagrammes de cas d'utilisation décrivent les fonctions générales et la portée d'un système. Ces diagrammes identifient également les interactions entre le système et ses acteurs. Les cas d'utilisation et les acteurs dans les diagrammes de cas d'utilisation décrivent ce que le système fait et comment les acteurs l'utilisent, mais ne montrent pas comment le système fonctionne en interne. [8]

5-2. Diagrammes cas d'utilisations des différents acteurs :

5-2-1. Diagramme cas d'utilisation du Patient :

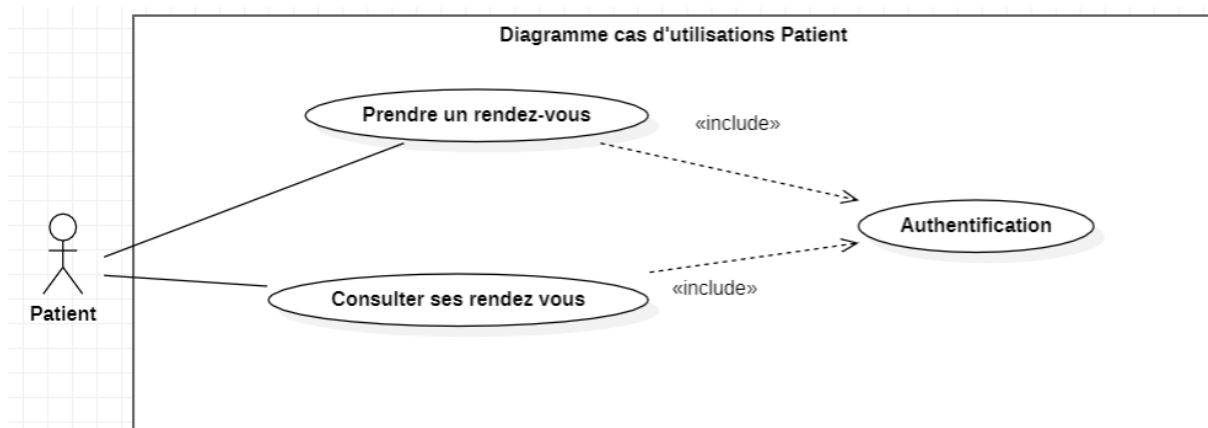


Figure II.5-2-1 : Diagramme cas d'utilisations «Patient».

5-2-2. Diagramme cas d'utilisation Secrétaire :

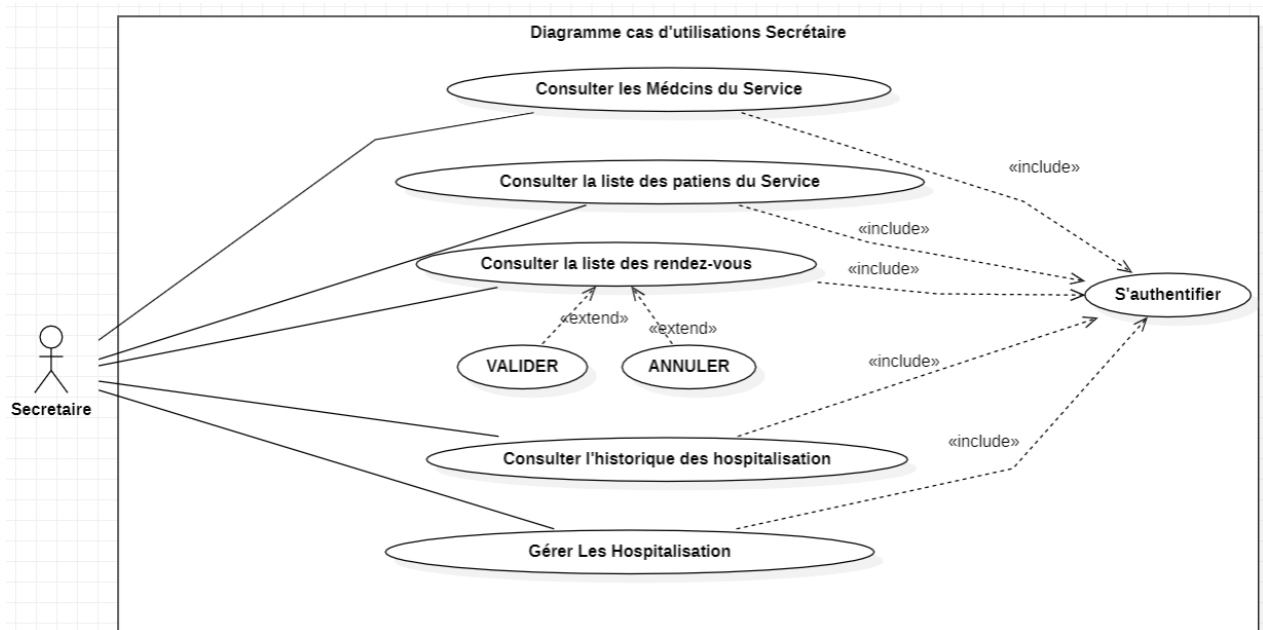


Figure II.5-2-2 : Diagramme de cas d'utilisation «Secrétaire».

5-2-3. Diagramme cas d'utilisation Médecin :

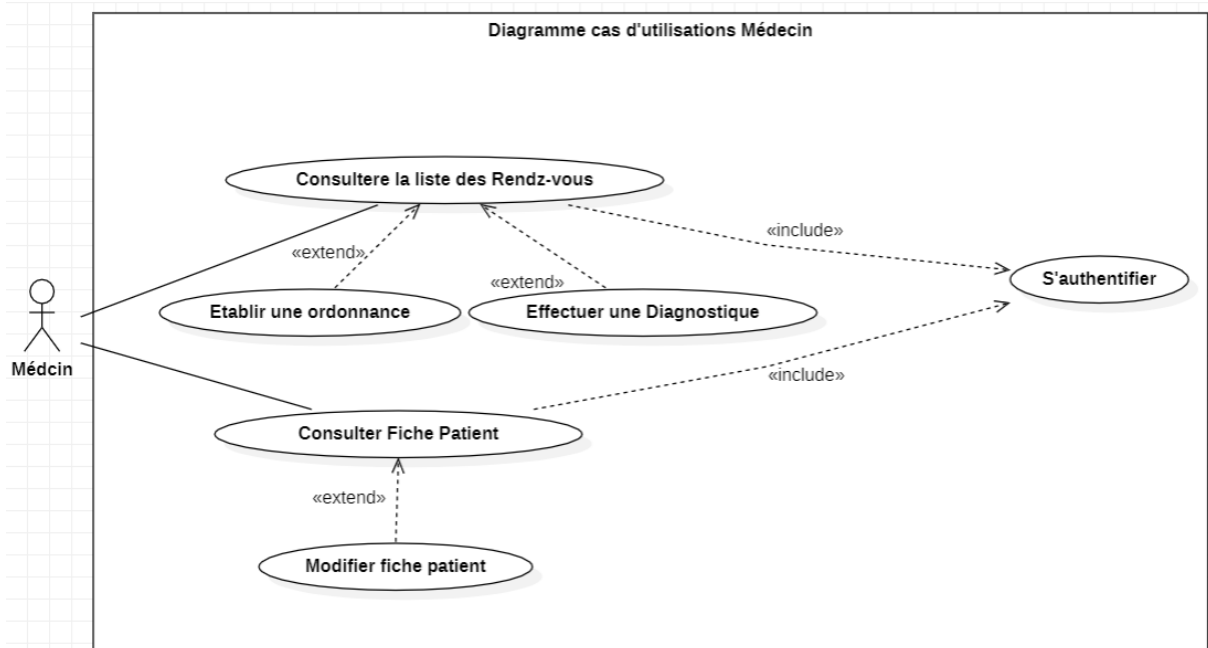


Figure II.5-2-3 : Diagramme de cas d'utilisation «Médecin».

5-2-4. Diagramme cas d'utilisation Administrateur :

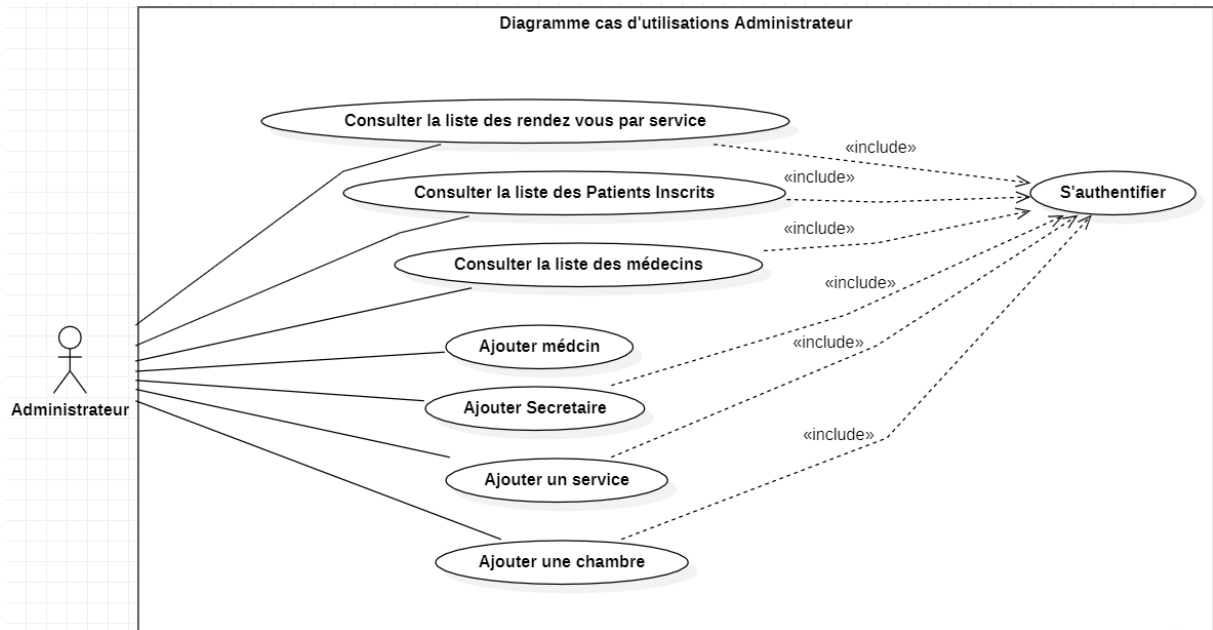


Figure II.5-2-4 : Diagramme de cas d'utilisation «Administrateur».

6- Descriptions textuelles des différents cas d'utilisations :

6-1. Authentification :

Acteur	Patient, Médecin, Secrétaire, Administrateur.
Objectif	Avoir accès au système en tant qu'entité spécifiée.
Condition	
Scénario	<ul style="list-style-type: none"> ➤ L'utilisateur demande la page de connexion. ➤ Le système renvoie le formulaire d'authentification. ➤ L'utilisateur saisie ses informations. (Pseudo/Email, Mot de passe) ➤ Le système vérifie la conformité des informations saisies en envoyant une requête à la base de données. ➤ La requête est vérifiée et validée par la base de données. ➤ L'utilisateur accède à son tableau de bord selon le type d'utilisateur.
Alternative	Informations erronées, envoie un message d'erreur.

Tableau 1 : Description textuelle du cas d'utilisation «Authentification».

6-2. Prendre un rendez-vous :

Acteur	Patient.
Objectif	Demander un rendez-vous en ligne.
Condition	Être authentifié.
Scénario	<ul style="list-style-type: none"> ➤ Le patient demande le formulaire de demande de rendez-vous. ➤ Le système renvoie le formulaire. ➤ Le patient remplit le formulaire contenant les informations suivantes : <ul style="list-style-type: none"> ▪ Service. ▪ Date. ▪ Heure. ➤ Le système vérifie la disponibilité du rendez-vous choisi en envoyant une requête à la base de données. ➤ La réponse à la base de données est favorable, la demande de rendez-vous est validée.
Alternative	Le rendez-vous est déjà réservé, envoie un message d'erreur.

Tableau 2 : Description textuelle du cas d'utilisation «Prendre un rendez-vous».

6-3. Confirmer un rendez-vous :

Acteur	Secrétaire.
Objectif	Valider un rendez-vous.
Condition	Être authentifié.
Scénario	<ul style="list-style-type: none"> ➤ Le secrétaire demande la liste des demandes de rendez-vous. ➤ Le système affiche la liste des demandes de rendez-vous. ➤ Le secrétaire clique sur le bouton 'Confirmer'. ➤ Le système redirige le secrétaire vers une page de choix de médecin. ➤ Le secrétaire choisit un des médecins disponible du service et lui attribue le rendez-vous. ➤ Le système valide le rendez-vous et change son état de 'En attente' à 'Confirmer' et insère le nom du médecin traitant.

Tableau 3 : Description textuelle du cas d'utilisation «Confirmer rendez-vous».

6-4. Annuler un rendez-vous :

Acteur	Secrétaire.
Objectif	Annuler un rendez-vous.
Condition	Être authentifié.
Scénario	<ul style="list-style-type: none"> ➤ Le secrétaire demande la liste des demandes de rendez-vous. ➤ Le système affiche la liste des demandes de rendez-vous. ➤ Le secrétaire clique sur le bouton 'Annuler'. ➤ Le système annule le rendez-vous et change son état de 'En attente' à 'Annulé'.

Tableau 4 : Description textuelle du cas d'utilisation «Annuler un rendez-vous».

6-5. Confirmer une demande d'hospitalisation :

Acteur	Secrétaire.
Objectif	Valider une demande d'hospitalisation.
Condition	Être authentifié.
Scénario	<ul style="list-style-type: none"> ➤ Le secrétaire demande la liste des demandes d'hospitalisation. ➤ Le système affiche la liste des demandes d'hospitalisation. ➤ Le secrétaire clique sur le bouton 'Confirmer'. ➤ Le système renvoie le secrétaire vers une page de choix de chambre d'hospitalisation. ➤ Le secrétaire choisit une des chambres disponibles et 'Hospitalise' le patient. ➤ Le système valide et insère les informations de l'hospitalisation dans les tables de la base de données adéquates.
Alternative	Aucune chambre disponible, impossible de valider la demande.

Tableau 5 : Description textuelle du cas d'utilisation «Confirmer un demande d'hospitalisation».

6-6. Annuler une demande d'hospitalisation :

Acteur	Secrétaire.
Objectif	Annuler une demande d'hospitalisation.
Condition	Être authentifié.
Scénario	<ul style="list-style-type: none"> ➤ Le secrétaire demande la liste des demandes d'hospitalisation. ➤ Le système affiche la liste des demandes d'hospitalisation. ➤ Le secrétaire clique sur le bouton 'Annuler'. ➤ Le système annule la demande d'hospitalisation et change son état de 'En attente' à 'Annuler'.

Tableau 6 : Description textuelle du cas d'utilisation «Annuler une demande d'hospitalisation».

6-7. Mettre à jour la fiche patient :

Acteur	Médecin.
Objectif	Mettre à jour la fiche patient.
Condition	Être authentifié.
Scénario	<ul style="list-style-type: none"> ➤ Le médecin demande la liste des rendez-vous affecté à son nom. ➤ Le système affiche la liste des rendez-vous en attente de traitement par le médecin. ➤ Le médecin clique sur le bouton 'Diagnostiquer'. ➤ Le système renvoie la page de diagnostic. ➤ Le médecin remplit le formulaire de diagnostic (Avec possibilité de demander une hospitalisation depuis cette page.) et valide le diagnostic. ➤ Le système confirme le diagnostic et modifie l'état du rendez-vous à 'traiter' tout en ajoutant le diagnostic à sa table dans la base de données, et demande à l'utilisateur s'il veut ou non établir une ordonnance. ➤ Renvoie à l'interface d'établissement d'ordonnance si «Confirmé». ➤ L'utilisateur remplit le formulaire de l'ordonnance et le confirme. ➤ Le système génère une ordonnance à partir d'un modèle déjà établi et affiche une vue d'avant impression. ➤ Le système renvoie au tableau de bord du médecin après impression/annulation.

Tableau 7 : Description textuelle du cas d'utilisation «Mettre à jour la fiche patient».

6-8. Ajouter un médecin :

Acteur	Administrateur.
Objectif	Ajouter un compte médecin.
Condition	Être authentifié.
Scénario	<ul style="list-style-type: none"> ➤ L'administrateur demande le formulaire d'ajout de médecin. ➤ Le système renvoie le formulaire. ➤ L'administrateur remplit le formulaire (Pseudo, mot de passe, email, service d'affectation) et le valide. ➤ Le système envoie une requête à la base de données et ajoute le médecin.
Alternative	Le pseudo est déjà pris, message d'erreur

Tableau 8 : Description textuelle du cas d'utilisation «Ajouter un médecin».

6-9. Ajouter un(e) secrétaire :

Acteur	Administrateur.
Objectif	Ajouter un compte secrétaire.
Condition	Être authentifié.
Scénario	<ul style="list-style-type: none"> ➤ L'administrateur demande le formulaire d'ajout de secrétaire. ➤ Le système renvoie le formulaire. ➤ L'administrateur remplit le formulaire (Pseudo, mot de passe, email, service d'affectation) et le valide. ➤ Le système envoie une requête à la base de données et ajoute le secrétaire.
Alternative	Le pseudo est déjà pris, message d'erreur

Tableau 9 : Description textuelle du cas d'utilisation «Ajouter un(e) secrétaire».

6-10. Ajouter un service :

Acteur	Administrateur.
Objectif	Ajouter un service.
Condition	Être authentifié.
Scénario	<ul style="list-style-type: none"> ➤ L'administrateur demande le formulaire d'ajout de services. ➤ Le système renvoie le formulaire. ➤ L'administrateur remplit le formulaire (Nom du service) et le valide. ➤ Une requête est envoyée à la base de données, vérifie si le service n'existe pas et le valide.
Alternative	Le service existe déjà, affiche un message d'erreur.

Tableau 10 : Description textuelle du cas d'utilisation «Ajouter un service».

6-11. Ajouter une chambre :

Acteur	Administrateur.
Objectif	Ajouter une chambre.
Condition	Être authentifié.
Scénario	<ul style="list-style-type: none"> ➤ L'administrateur demande le formulaire d'ajout de chambres. ➤ Le système renvoie le formulaire. ➤ L'administrateur remplit le formulaire (Nom de chambre, Service d'affectation) et le valide. ➤ Une requête est envoyée à la base de données, vérifie si la chambre n'existe pas et le valide.
Alternative	La chambre existe déjà, affiche un message d'erreur.

Tableau 11 : Description textuelle du cas d'utilisation «Ajouter une chambre».

7- Diagrammes de séquence :

7-1. Définition :

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation Unified Modeling Language. [9]

7-2. Diagramme séquence cas «Authentification» :

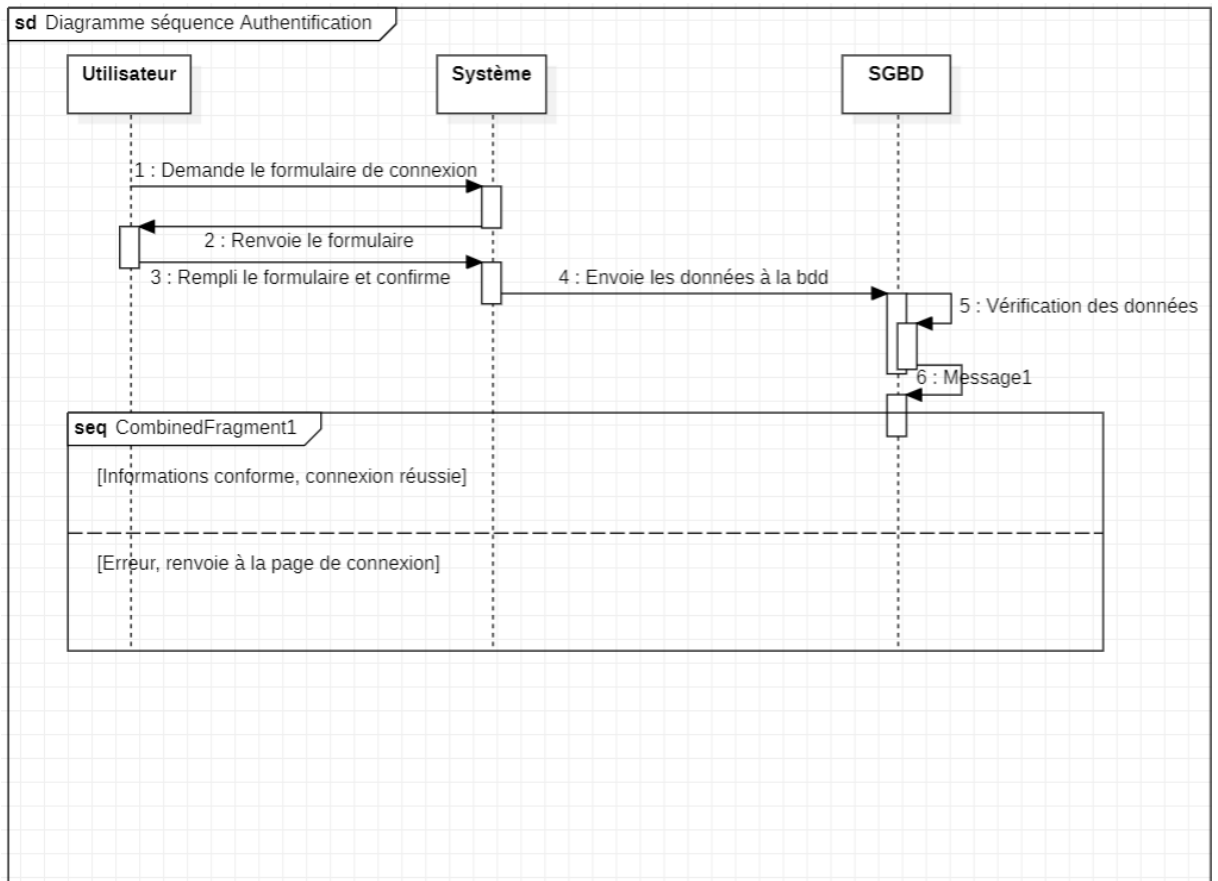


Figure II.7-2. : Diagramme séquence «Authentification»

7-3. Diagramme séquence cas «Prendre un rendez-vous» :

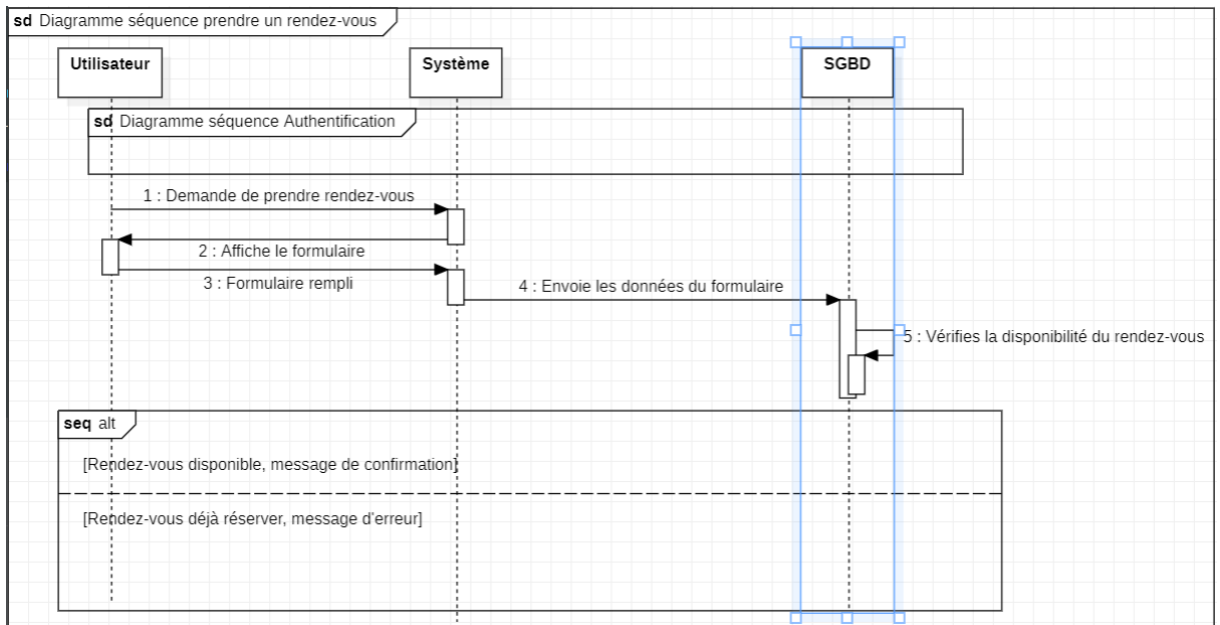


Figure II.7-3. : Diagramme séquence «Prendre un rendez-vous»

7-4. Diagramme séquence cas «Confirmer un rendez-vous» :

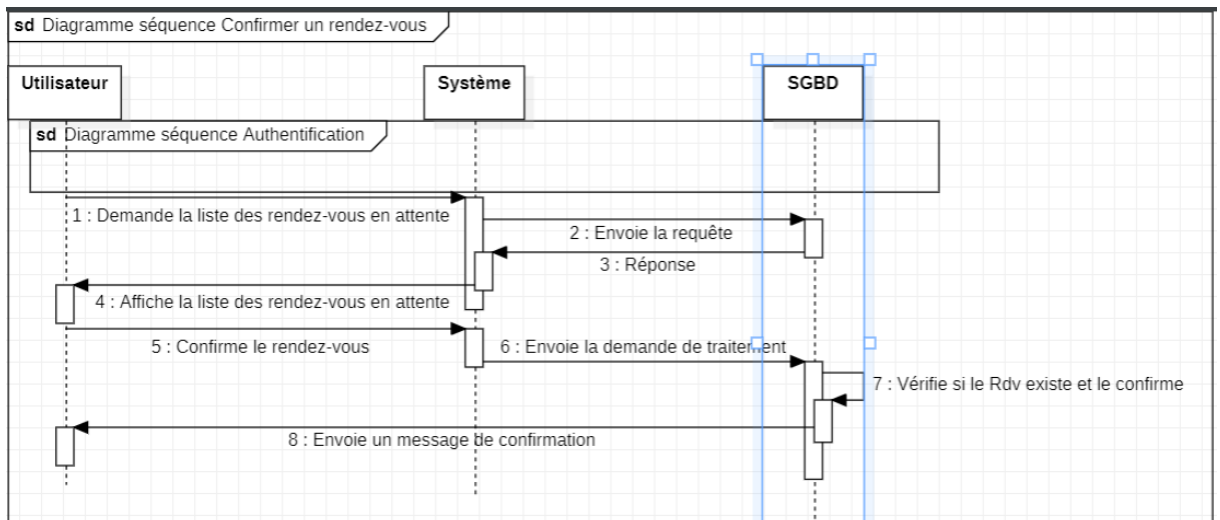


Figure II.7-4. : Diagramme séquence «Confirmer un rendez-vous»

7-5. Diagramme séquence cas «Annuler un rendez-vous» :

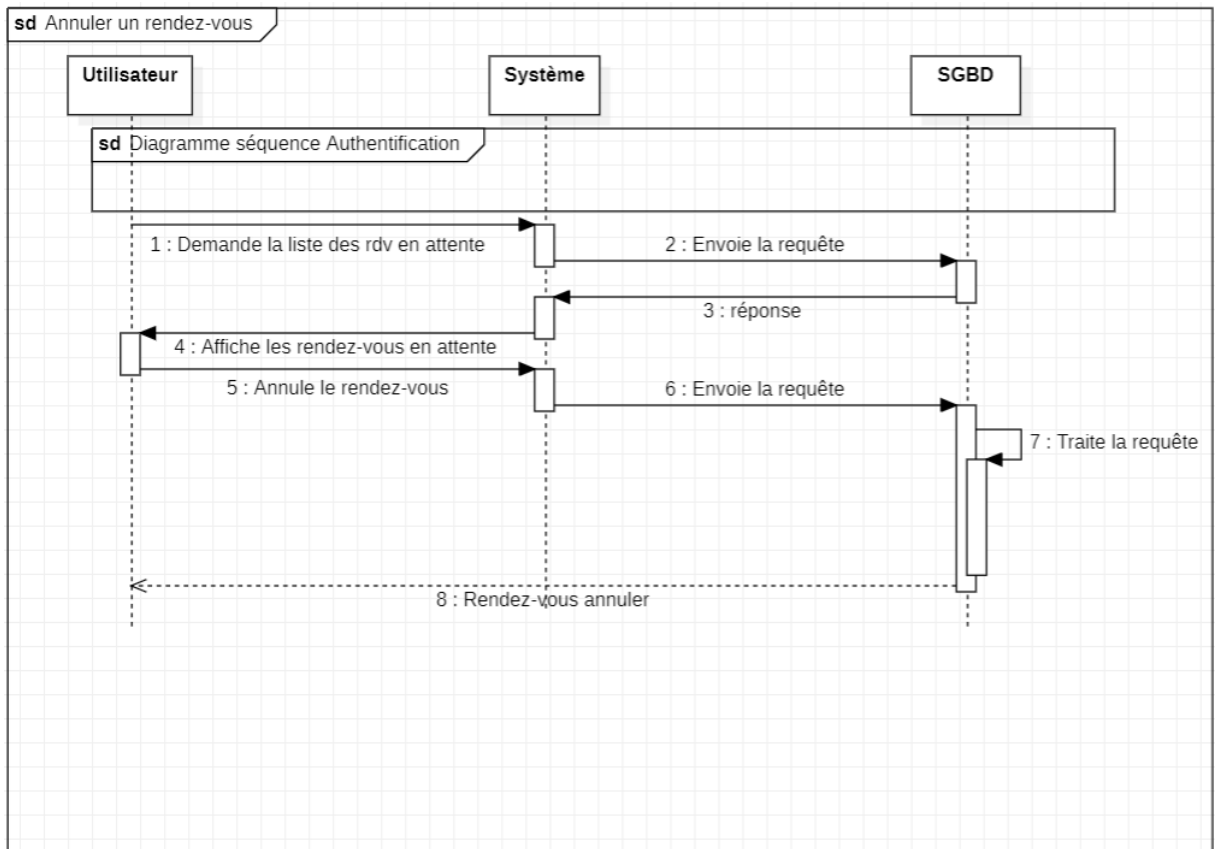


Figure II.7-5. : Diagramme séquence «Annuler un rendez-vous»

7-6. Diagramme séquence cas «Confirmer une demande d'hospitalisation» :

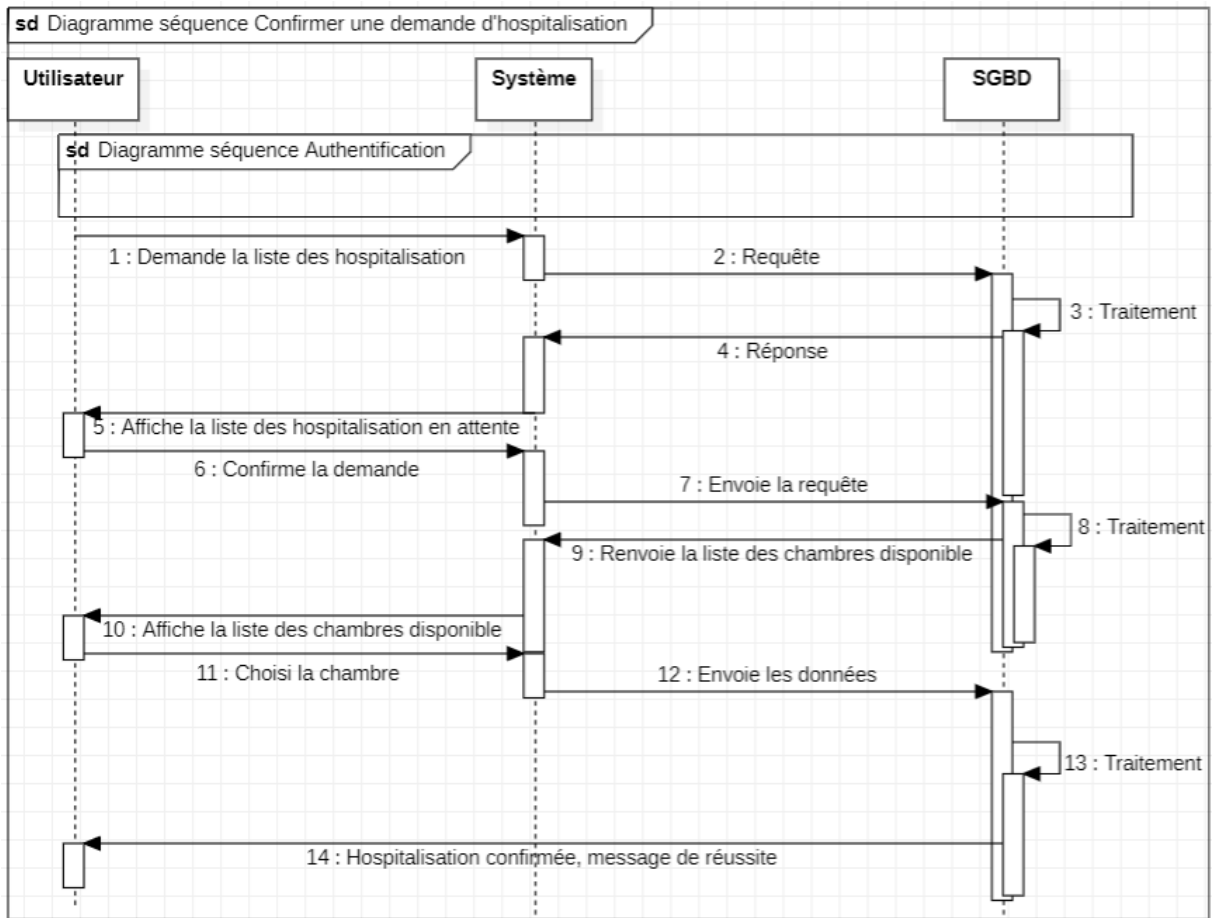


Figure II.7-6. : Diagramme séquence «Confirmer une demande d'hospitalisation»

7-7. Diagramme séquence cas «Annuler une demande d'hospitalisation» :

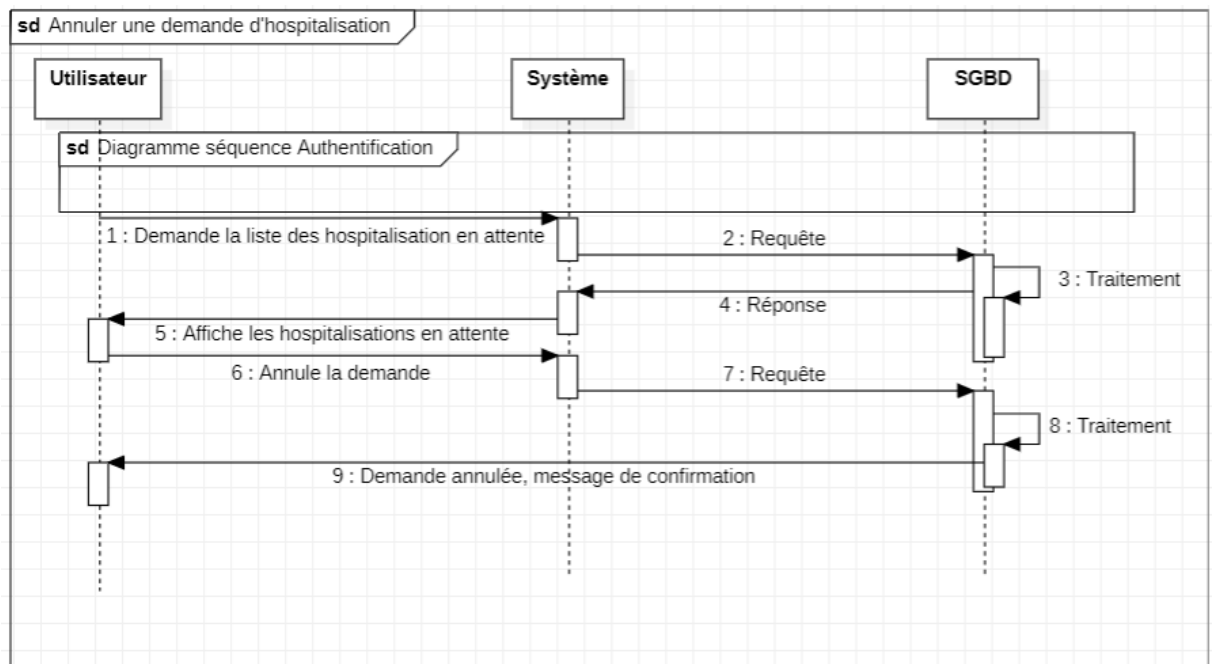


Figure II.7-7. : Diagramme séquence «Annuler une demande d'hospitalisation»

7-8. Diagramme séquence cas «Mettre à jour la fiche patient» :

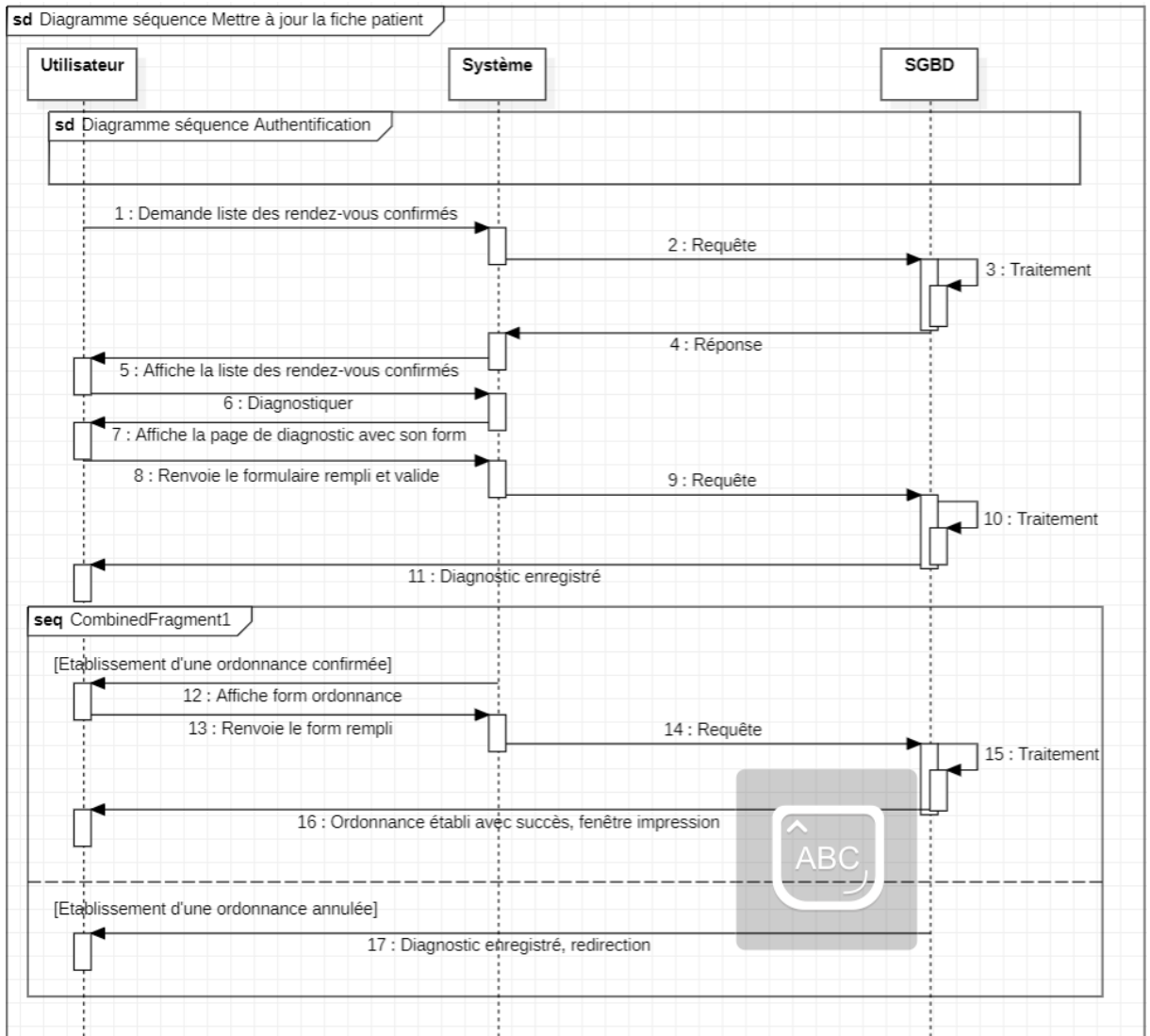


Figure II.7-8. : Diagramme séquence «Mettre à jour la fiche patient»

7-9. Diagramme séquence cas «Ajouter un médecin» :

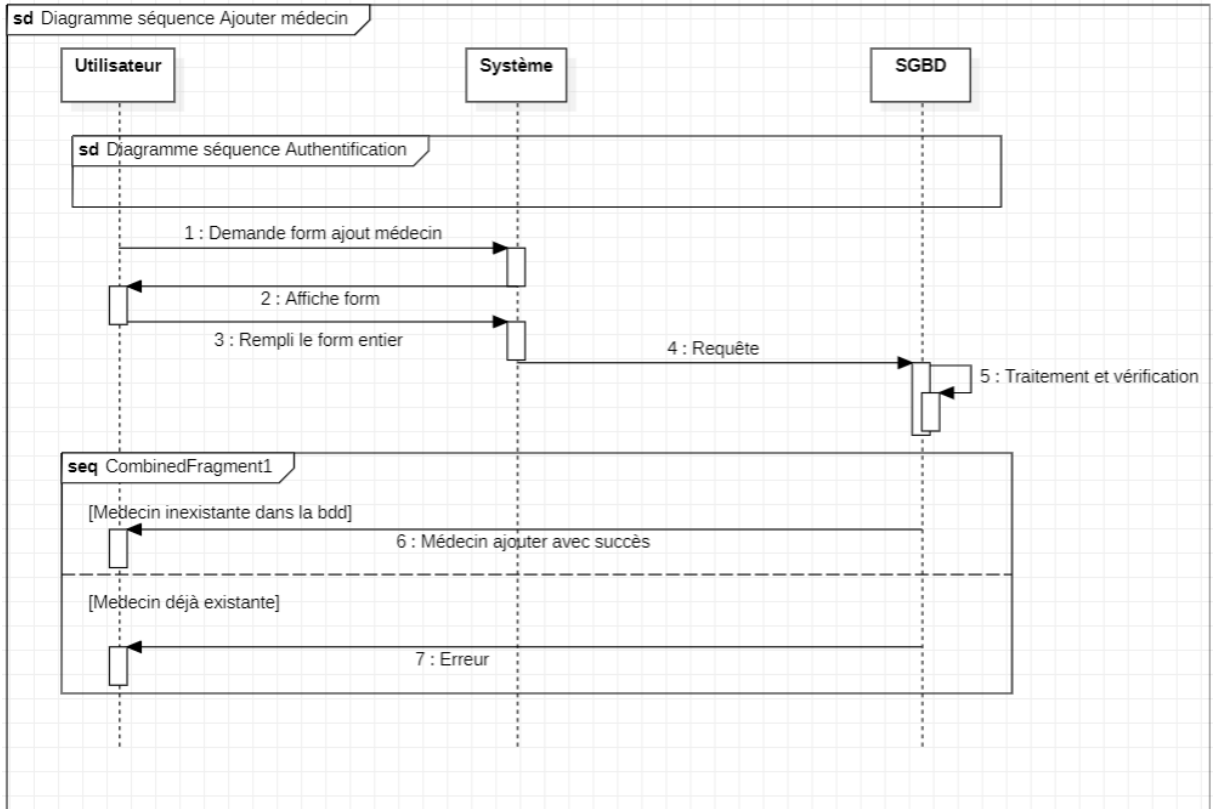


Figure II.7-9. : Diagramme séquence «Ajouter un médecin»

7-10. Diagramme séquence cas «Ajouter un(e) secrétaire» :

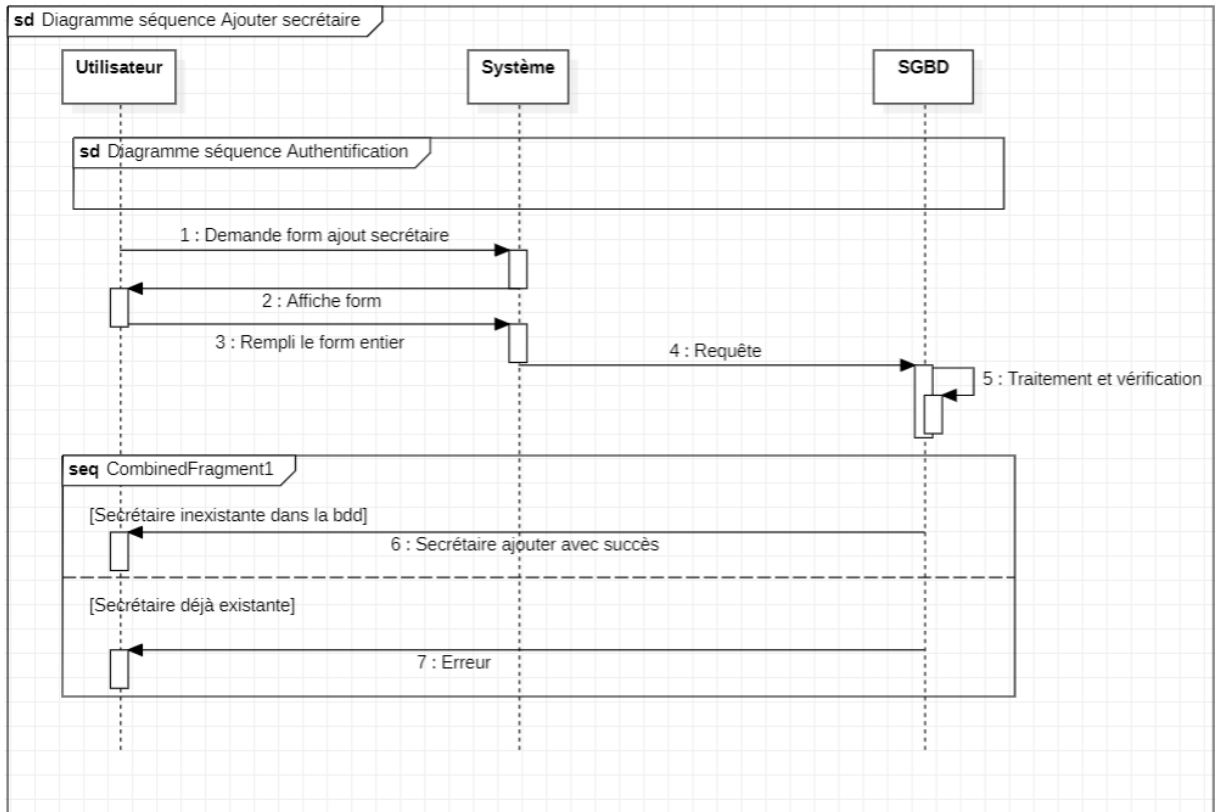


Figure II.7-10. : Diagramme séquence «Ajouter un(e) secrétaire»

7-11. Diagramme séquence cas «Ajouter un service» :

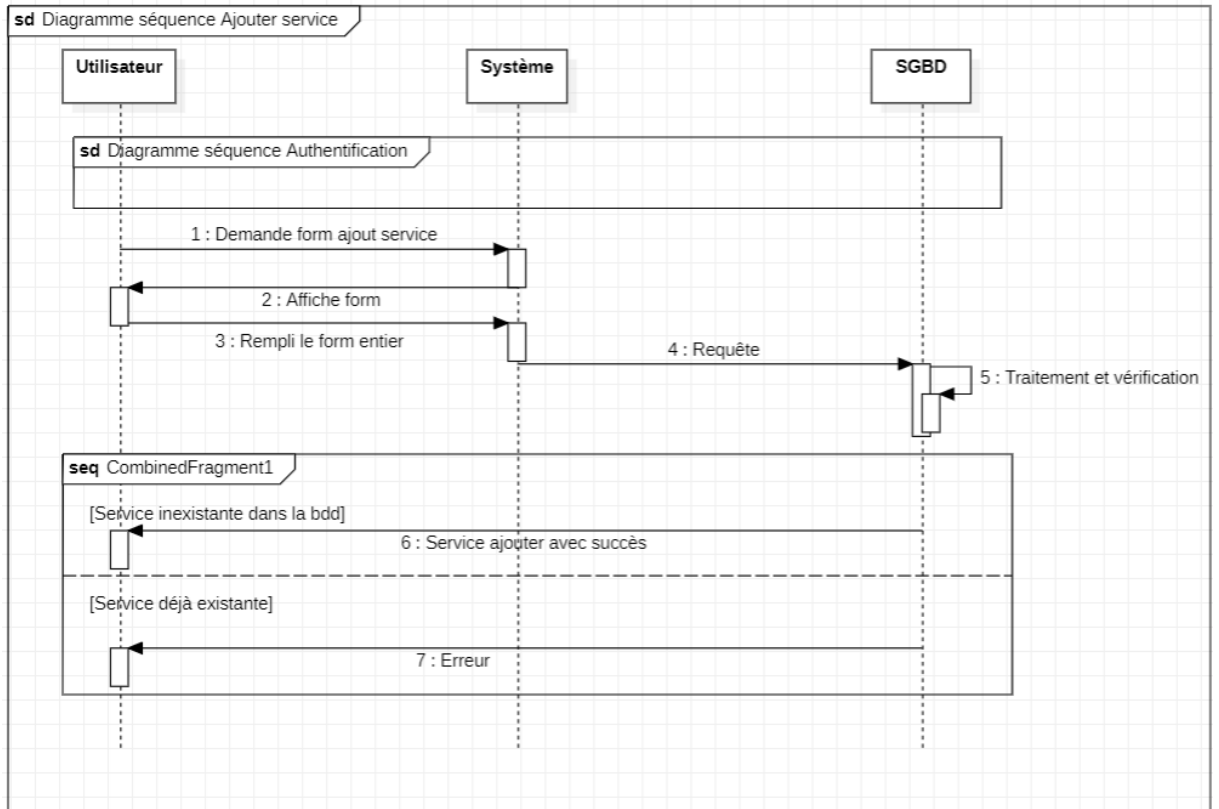


Figure II.7-11. : Diagramme séquence «Ajouter un service»

7-12. Diagramme séquence cas «Ajouter une chambre» :

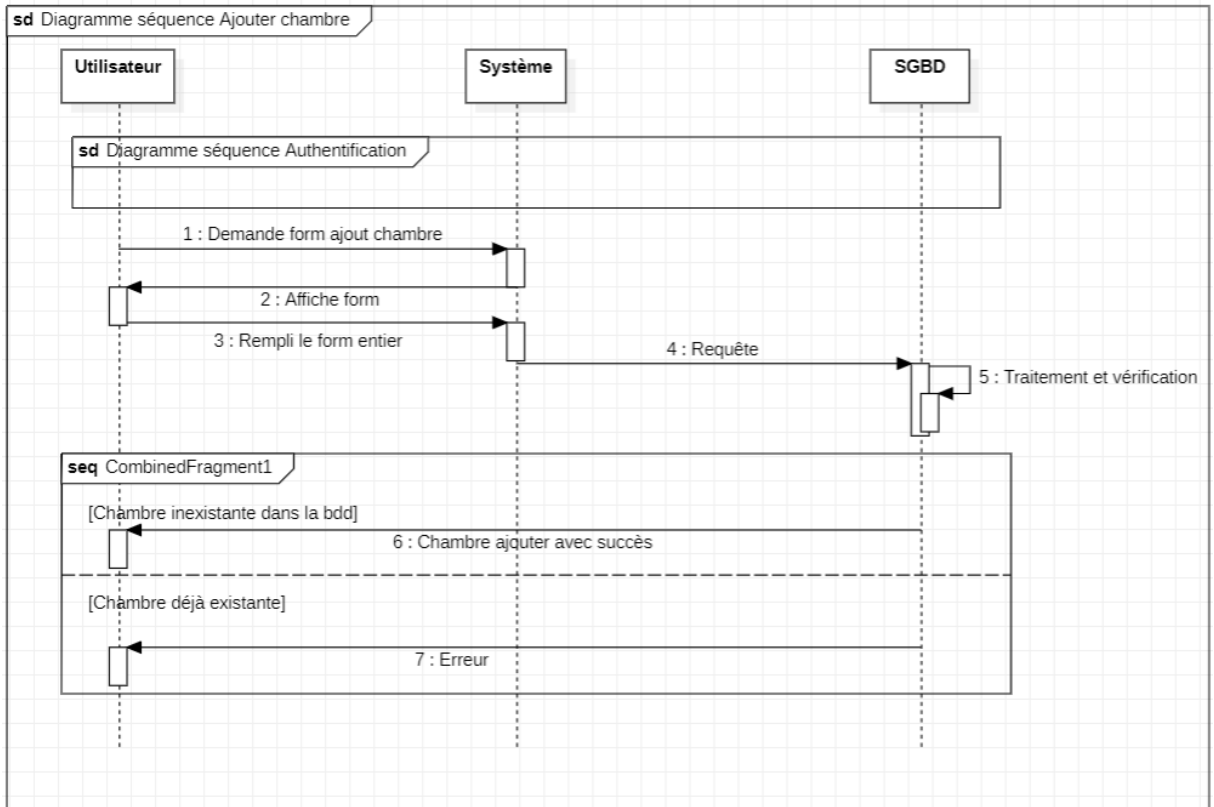


Figure II.7-12. : Diagramme séquence «Ajouter une chambre»

8- Diagramme de classes :

8-1. Définition :

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que leurs relations. Ce diagramme fait partie de la partie statique d'UML, ne s'intéressant pas aux aspects temporels et dynamiques. [10]

8-2. Diagramme de classes :

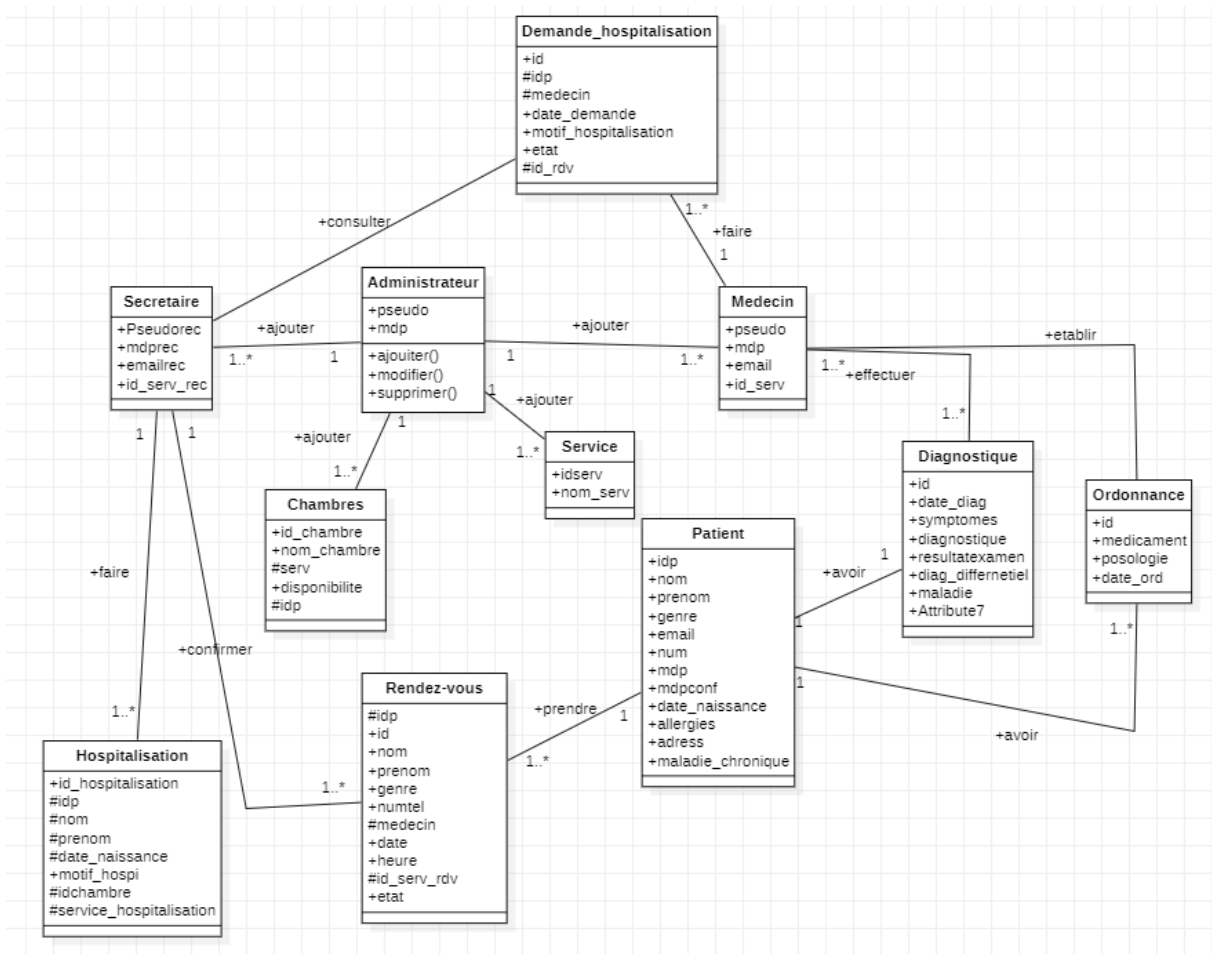


Figure II.8-2 : Diagramme de classes.

8-3. Modèle relationnel :

Le modèle relationnel est une manière de modéliser les relations existantes entre plusieurs informations, et de les ordonner entre elles. Cette modélisation qui repose sur des principes mathématiques mis en avant par E.F. Codd est souvent retranscrite physiquement (« implémentée ») dans une base de données. [11]

8-4. Règles de passage au modèle relationnel :

Règle 1

Unicité :

Toute l'information dans la base de données est représentée d'une et une seule manière, à savoir par des valeurs dans des champs de colonnes de tables.

Règle 2

Garantie d'accès :

Toutes les données doivent être accessibles sans ambiguïté. Cette règle est essentiellement un ajustement de la condition fondamentale pour des clefs primaires. Elle indique que chaque valeur scalaire individuelle dans la base de données doit être logiquement accessible en indiquant le nom de la table contenant, le nom de la colonne contenant et la valeur principale primaire de la rangée contenant.

Règle 3

Traitement des valeurs nulles :

Le système de gestion de bases de données doit permettre à chaque champ de demeurer nul (ou vide). Spécifiquement, il doit soutenir une représentation "d'information manquante et d'information inapplicable" qui est systématique, distincte de toutes les valeurs régulières (par exemple, "distincte de zéro ou tous autres nombres," dans le cas des valeurs numériques), et ce indépendamment du type de données. Cela implique également que de telles représentations doivent être gérées par le système de gestion de bases de données d'une manière systématique.

[12]

8-5. Le modèle relationnel :

Administrateur (pseudo,mdp)

Chambre (id_chambre,nom_chambre,disponibilite,#idpatient,#serv)

demande_hospitalisation (id,medecin,date_demande,motif_hospitalisation,etat,#id_rdv,#idp)

diagnostique (#id_rdv,#idp__medecin,date_diagnostique,symptomes,diagnostic,resultat_examens,diagnostic_différentiel,maladie)

hospitalisation(id_hospitalisation,#idp,motif_hospitalisation,#idchambre,#service_hospitalisation,date_hosp, date_sortie)

medecins (pseudo,mdp,email,#id_serv)

patients(idp,nom,prenom,genre,email,num,mdp,mdpconf,date_naissance,allergies,adresse)

secretaire (pseudorec,mdp,mdpconf,emailrec,id_serv_rec)

rendez_vous (id,#idp,nom,prenom,genre,email,numtel,#medecin,date,heure,#id_serv_rdv,etat)

service (idservice,nom_service)

ordonnance (id,#id_rdv,medicaments,posologie,date_ord)

9- Conclusion :

Dans ce deuxième chapitre, nous avons réalisé la conception de notre application avec les différents diagrammes de modélisation et présenter la manière de fonctionnement de notre système.

Chapitre 3

Présentation du site web

1- Introduction :

_____ Dans ce troisième, et dernier chapitre, nous allons parler de la réalisation de notre site, des moyens et outils utilisés, des langages utilisés, et nous allons présenter quelques fonctionnalités que nous avons implémentées.

2- Application web :

En informatique, une application web (aussi appelée web application, de l'anglais et français) est une application manipulable directement en ligne grâce à un navigateur web et qui ne nécessite donc pas d'installation sur les machines clientes, contrairement aux applications mobiles¹. De la même manière que les sites web, une application web est généralement installée sur un serveur et se manipule en actionnant des *widgets* à l'aide d'un navigateur web, via un réseau informatique (Internet, intranet, réseau local, etc.). Exemples :

- Des messageries web, les systèmes de gestion de contenu, les wikis et les blogs sont des applications web.
- Les moteurs de recherches, les logiciels de commerce électronique, les jeux en ligne, les logiciels de forum, les agrégateurs peuvent être sous forme d'application web.
- Des appareils réseau tels que les routeurs sont parfois équipés d'une application web dans leur micrologiciel².

[12]

3- Outils de réalisation :

3-1. Réalisation des diagrammes :

Pour réaliser nos diagrammes UML, nous avons utilisé un logiciel de développement de diagrammes appelé StarUML.

StarUML :

StarUML est un outil de génie logiciel dédié à la modélisation UML et édité par la société coréenne MKLabs. Il est multiplateforme et fonctionne sous Windows, Linux et MacOS.

[13]

3-2. Réalisation de l'application web:

Pour mener à bien la réalisation de notre application, nous avons utilisé les langages de programmations : PHP, JavaScript, Html, CSS. Ainsi que Xampp, PhpMyAdmin et Visual Code Studio.

3-2-1. PHP :

C'est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur web, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet. [14]

3-2-2. JS :

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les langages HTML et CSS, JavaScript est au cœur des langages utilisés par les développeurs web³. Une grande majorité des sites web l'utilisent⁴, et la majorité des navigateurs web disposent d'un moteur JavaScript⁵ pour l'interpréter. [15]

3-2-3. HTML :

Ce langage permet d'écrire de l'hypertexte (d'où son nom), de structurer sémantiquement une page web, de mettre en forme du contenu, de créer des formulaires de saisie ou encore d'inclure des ressources multimédias dont des images, des vidéos, et des programmes informatiques. L'HTML offre également la possibilité de créer des documents interopérables avec des équipements très variés et conformément aux exigences de l'accessibilité du web.

[16]

3-2-4.CSS :

Les feuilles de style en cascade, généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C). Introduit au milieu des années 1990, CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années 2000. [17]

3-2-5.XAMPP :

XAMPP est un ensemble de logiciels permettant de mettre en place un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres (X (cross) Apache MariaDB Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus. [18]

3-2-6.PMA :

PhpMyAdmin (PMA) est une application Web de gestion pour les systèmes de gestion de base de données MySQL et MariaDB, réalisée principalement en PHP. [19]

3-2-7.VS Code :

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et MacOS.

Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code (IntelliSense.), les snippets, la refactorisation du code et Git intégré. Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires. [20]

4-Interfaces de l'application web :

4-1. Page d'accueil:

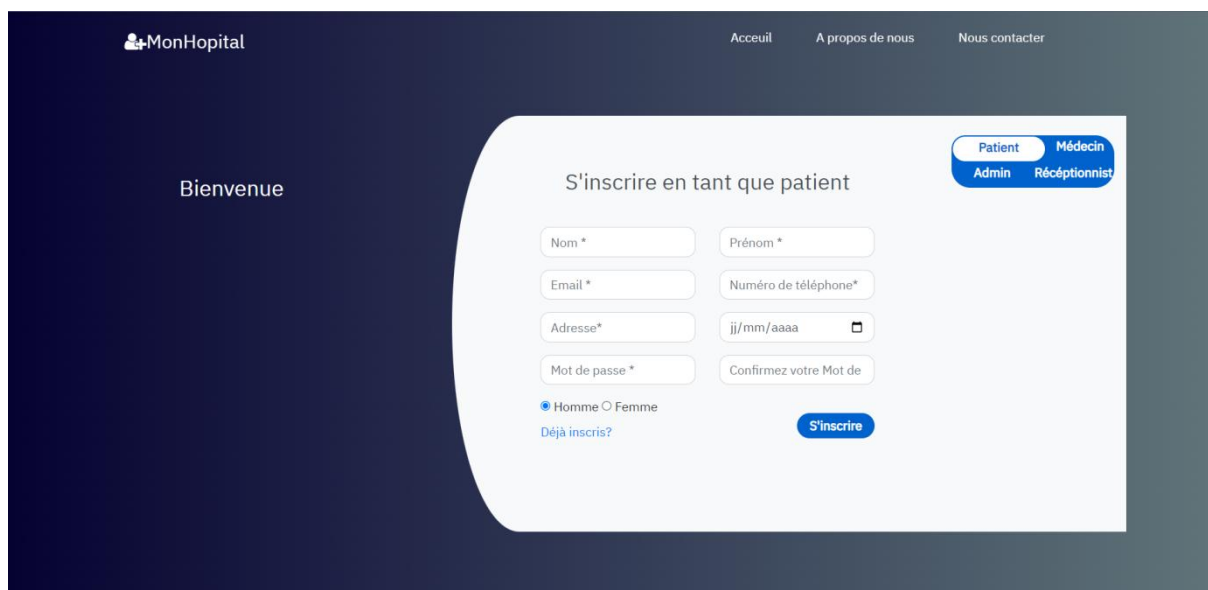


Figure III.4-1 : Page d'accueil.

4-2. Compte patient : Prise de rendez-vous:

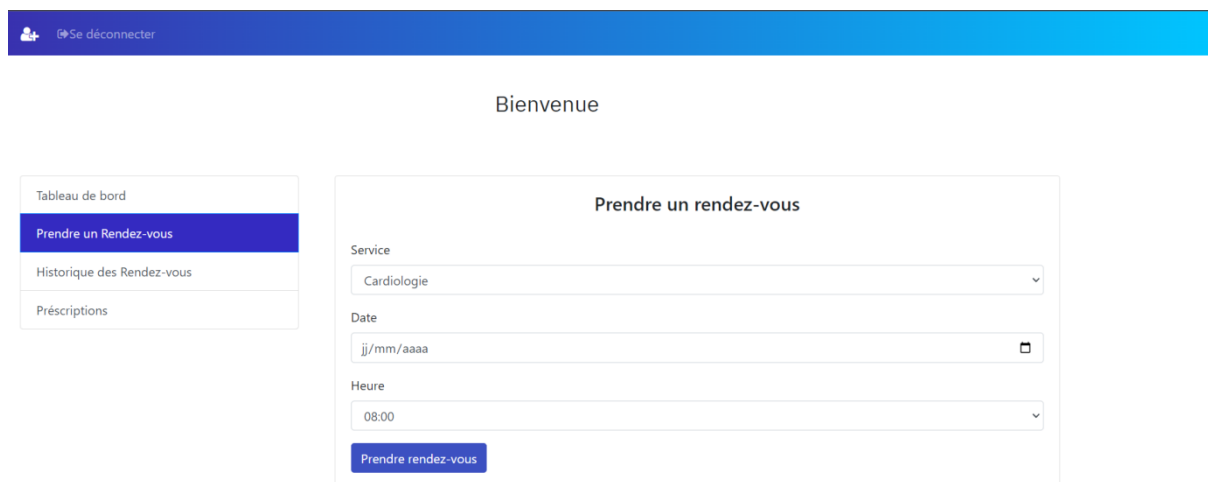


Figure III.4-2 : Prise de rendez-vous.

4-3. Compte patient : Historique des rendez-vous:

The screenshot shows a patient account interface. At the top, there is a blue header with a user icon and the text "Se déconnecter". Below the header, the word "Bienvenue" is centered. On the left, there is a sidebar menu with the following items: "Tableau de bord", "Prendre un Rendez-vous", "Historique des Rendez-vous" (highlighted in blue), and "Préscriptions". To the right of the sidebar is a table with the following columns: "Rendez-vous ID", "Service", "Date", "Heure", and "État".

Rendez-vous ID	Service	Date	Heure	État
1	Cardiologie	2023-06-14	14:30:00	traiter
2	Cardiologie	2023-06-16	15:00:00	traiter
3	Cardiologie	2023-06-24	12:00:00	traiter
4	Cardiologie	2023-06-25	14:30:00	traiter
5	Cardiologie	2023-06-12	15:30:00	traiter
6	Cardiologie	2023-06-22	16:30:00	traiter
7	Cardiologie	2023-06-30	12:30:00	traiter

Figure III.4-3 : Historique des rendez-vous du patient.

4-4. Compte secrétaire: Liste des médecins du service :

The screenshot shows a secretary account interface. At the top, there is a blue header with a user icon and the text "MH Se déconnecter". Below the header, the text "Bienvenue à votre Espace Compte secretaireradio" is centered. On the left, there is a sidebar menu with the following items: "Tableau de bord", "Liste des médecins" (highlighted in blue), "Liste des patients", "Liste des rendez-vous", "Hospitaliser un patient", "Liste des chambres disponibles", and "Historique des hospitalisations". To the right of the sidebar is a table with the following columns: "Pseudo", "Email", and "ID du Service".

Pseudo	Email	ID du Service
medecin	medecin@gmail.com	1
medrad	medred@gmail.com	1

Figure III.4-4 : Liste des médecins du service.

4-5. Compte secrétaire: Historique des rendez-vous :

2

Bienvenue à votre Espace Compte secretaireradio

Tableau de bord	Liste des rendez-vous pour le service <table border="1"> <thead> <tr> <th>ID Rendez-vous</th> <th>Nom Patient</th> <th>Prénom Patient</th> <th>Âge</th> <th>Date</th> <th>Heure</th> <th>État</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Debb</td> <td>Yann</td> <td>24</td> <td>2023-06-14</td> <td>14:30:00</td> <td>traiter</td> <td></td> </tr> <tr> <td>2</td> <td>Debb</td> <td>Yann</td> <td>24</td> <td>2023-06-16</td> <td>15:00:00</td> <td>traiter</td> <td></td> </tr> <tr> <td>3</td> <td>Debb</td> <td>Yann</td> <td>24</td> <td>2023-06-24</td> <td>12:00:00</td> <td>traiter</td> <td></td> </tr> <tr> <td>4</td> <td>Debb</td> <td>Yann</td> <td>24</td> <td>2023-06-25</td> <td>14:30:00</td> <td>traiter</td> <td></td> </tr> <tr> <td>5</td> <td>Debb</td> <td>Yann</td> <td>24</td> <td>2023-06-12</td> <td>15:30:00</td> <td>traiter</td> <td></td> </tr> <tr> <td>6</td> <td>Debb</td> <td>Yann</td> <td>24</td> <td>2023-06-22</td> <td>16:30:00</td> <td>traiter</td> <td></td> </tr> <tr> <td>7</td> <td>Debb</td> <td>Yann</td> <td>24</td> <td>2023-06-30</td> <td>12:30:00</td> <td>traiter</td> <td></td> </tr> </tbody> </table>	ID Rendez-vous	Nom Patient	Prénom Patient	Âge	Date	Heure	État	Action	1	Debb	Yann	24	2023-06-14	14:30:00	traiter		2	Debb	Yann	24	2023-06-16	15:00:00	traiter		3	Debb	Yann	24	2023-06-24	12:00:00	traiter		4	Debb	Yann	24	2023-06-25	14:30:00	traiter		5	Debb	Yann	24	2023-06-12	15:30:00	traiter		6	Debb	Yann	24	2023-06-22	16:30:00	traiter		7	Debb	Yann	24	2023-06-30	12:30:00	traiter	
ID Rendez-vous		Nom Patient	Prénom Patient	Âge	Date	Heure	État	Action																																																									
1		Debb	Yann	24	2023-06-14	14:30:00	traiter																																																										
2		Debb	Yann	24	2023-06-16	15:00:00	traiter																																																										
3		Debb	Yann	24	2023-06-24	12:00:00	traiter																																																										
4		Debb	Yann	24	2023-06-25	14:30:00	traiter																																																										
5		Debb	Yann	24	2023-06-12	15:30:00	traiter																																																										
6		Debb	Yann	24	2023-06-22	16:30:00	traiter																																																										
7	Debb	Yann	24	2023-06-30	12:30:00	traiter																																																											
Liste des médecins																																																																	
Liste des patients																																																																	
Liste des rendez-vous																																																																	
Hospitaliser un patient																																																																	
Liste des chambres disponibles																																																																	
Historique des hospitalisations																																																																	

Figure III.4-5 : Historique des rendez-vous du service.

4-6. Compte secrétaire: Historique des hospitalisations :



Bienvenue à votre Espace Compte secretaireradio

Tableau de bord	<table border="1"> <thead> <tr> <th>ID Hospitalisation</th> <th>ID Patient</th> <th>Nom Patient</th> <th>Prénom Patient</th> <th>Date de Naissance</th> <th>Date d'Hospitalisation</th> <th>Motif d'Hospitalisation</th> <th>ID Chambre</th> <th>Service Hospitalisation</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>3</td> <td>Debb</td> <td>Yann</td> <td>1998-08-16</td> <td>2023-06-25</td> <td></td> <td>2</td> <td>Cardiologie</td> </tr> <tr> <td>2</td> <td>3</td> <td>Debb</td> <td>Yann</td> <td>1998-08-16</td> <td>2023-06-25</td> <td>sxqxcs</td> <td>1</td> <td>Cardiologie</td> </tr> </tbody> </table>	ID Hospitalisation	ID Patient	Nom Patient	Prénom Patient	Date de Naissance	Date d'Hospitalisation	Motif d'Hospitalisation	ID Chambre	Service Hospitalisation	1	3	Debb	Yann	1998-08-16	2023-06-25		2	Cardiologie	2	3	Debb	Yann	1998-08-16	2023-06-25	sxqxcs	1	Cardiologie
ID Hospitalisation		ID Patient	Nom Patient	Prénom Patient	Date de Naissance	Date d'Hospitalisation	Motif d'Hospitalisation	ID Chambre	Service Hospitalisation																			
1		3	Debb	Yann	1998-08-16	2023-06-25		2	Cardiologie																			
2		3	Debb	Yann	1998-08-16	2023-06-25	sxqxcs	1	Cardiologie																			
Liste des médecins																												
Liste des patients																												
Liste des rendez-vous																												
Hospitaliser un patient																												
Liste des chambres disponibles																												
Historique des hospitalisations																												

Figure III.4-6 : Historique des hospitalisations.

4-7. Compte médecin: Liste des rendez-vous :

Bienvenue

Tableau de bord
Liste des Rendez-vous
Fiches des patients

Liste des rendez-vous

ID Rendez-vous	Nom Patient	Prénom Patient	Date de naissance	Date	Heure	Allergies	Action
8	Debb	Yann	1998-08-16	2023-06-29	16:00:00	kz,arzet,orke,.....	Diagnostiquer

Figure III.4-7 : Liste des rendez-vous.

4-8. Compte médecin: Mise à jour de la fiche patient :

Centre Hospitalier 'Nom de l'établissement'

Informations du patient
 Nom: Debb
 Prénom: Yann
 Date de naissance: 1998-08-16
 Adresse: sidi aich

Ordonnance
 Doliprane
 2x par jour, après repas

Imprimer 1 feuille de papier
 Destination: Microsoft Print to PDF
 Pages: Tous
 Mise en page: Portrait
 Couleur: Noir et blanc
 Plus de paramètres

Imprimer Annuler

Figure III.4-8 : Mise à jour de la fiche patient (Diagnostic et ordonnance).

4-9. Compte médecin: Fiches patient :

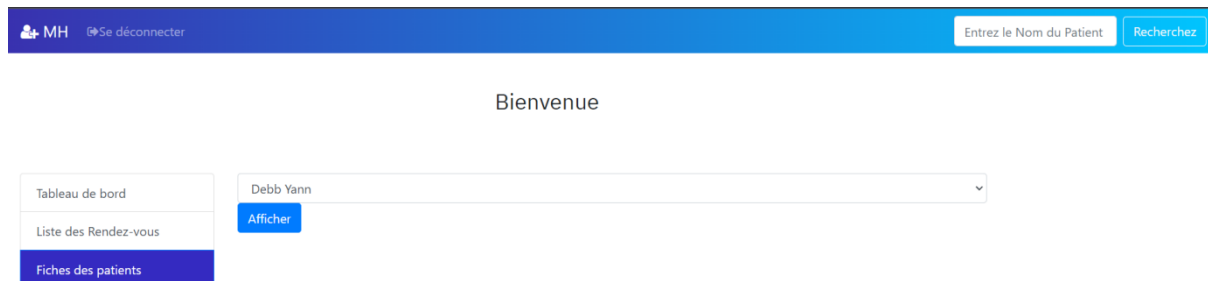


Figure III.4-9 : Fiches patient.

5-Conclusion :

Dans ce chapitre, nous avons parlé des langages de programmation que nous avons utilisés, et avons illustrés quelques interfaces de notre application web pour conclure ce troisième et dernier chapitre.

Conclusion Générale :

Au cours de ce travail, nous avons présenté les différentes étapes ayant conduit à la mise en œuvre d'une application web de Gestion Hospitalière centrée patient, qui contient notamment, la gestion des rendez-vous en ligne, la gestion des fiches patients et la gestion des demandes d'hospitalisation.

Nous avons commencé notre travail par identifier les problèmes que rencontre le corps médical dans le but d'alléger la charge de travail et faciliter la gestion centrée patient.

Afin de réaliser notre application, nous avons utilisé 'PhpMyAdmin' pour implémenter notre base de données et les différents langages : HTML, CSS, JS et PHP ainsi que Bootstrap.

La réalisation de ce projet fut un processus très intéressant pour nous, en tant qu'étudiants, bien que nous ayons déjà réalisé quelques projets, celui-ci aura été le premier que nous avons mené à terme.

Notre application Web n'est cependant pas complète et laisse une grande marge d'amélioration. La gestion hospitalière peut représenter un très grand projet, gérer les confidentialités quand à certaines informations concernant les patients, gestion des finances...etc. Notre travail représente une base, et qui reste ouvert à l'amélioration, et peut être un projet plus enrichie à l'avenir.

Références

- [1] : <https://www.techno-science.net/definition/288.html>
- [2] : <https://openclassrooms.com/fr/courses/6739646-realisez-un-cahier-des-charges-fonctionnel>
- [3] : <https://www.tuleap.org/fr/agile/comprendre-methode-agile-scrum-10-minutes>
- [4] : <https://blog.harvestr.io/fr/methode-scrum>
- [5] : [https://fr.wikipedia.org/wiki/UML_\(informatique\)](https://fr.wikipedia.org/wiki/UML_(informatique))
- [6] <https://fr.wikipedia.org/wiki/Patient>
- [7] <https://www.letudiant.fr/metiers/secteur/secretariat/secretaire-medicale.html>
- [8] <https://www.ibm.com/docs/fr/rational-soft-arch/9.5?topic=diagrams-use-case>
- [9] https://fr.wikipedia.org/wiki/Diagramme_de_s%C3%A9quence
- [10] https://fr.wikipedia.org/wiki/Diagramme_de_classes
- [11] https://fr.wikipedia.org/wiki/Mod%C3%A8le_relationnel
- [12] https://fr.wikipedia.org/wiki/Application_web
- [13] <https://fr.wikipedia.org/wiki/StarUML>
- [14] <https://fr.wikipedia.org/wiki/PHP>
- [15] <https://fr.wikipedia.org/wiki/JavaScript>
- [16] https://fr.wikipedia.org/wiki/Hypertext_Markup_Language
- [17] https://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade
- [18] <https://fr.wikipedia.org/wiki/XAMPP>
- [19] <https://fr.wikipedia.org/wiki/PhpMyAdmin>
- [20] https://fr.wikipedia.org/wiki/Visual_Studio_Code

Résumé :

L'objectif de notre travail est la conception et réalisation d'une application web dédiée à la gestion centrée patient d'un hôpital Cette application permet la gestion des rendez-vous en ligne, la gestion des fiches patient et la gestion des hospitalisations.

Ce travail a été réalisé en utilisation la méthode agile SCRUM et le langage de modélisation UML. Nous avons implémenté notre application web en utilisant les langages de programmations HTML, CSS, PHP, JS ainsi que le MySQL PhpMyAdmin comme système de gestion de base de données.

Abstract :

The point of our work is the design and production of a web application dedicated to the patient-centric management of a hospital. This application allows the management of online appointments, the management of patient files and the management of hospitalizations.

This work was carried out using the agile SCRUM method and the UML modeling language. We implemented our web application using HTML, CSS, PHP, JS programming languages as well as MySQL PhpMyAdmin as database management system.