



Ministère de l'Enseignement Supérieur et de la
Recherche Scientifique
Direction Générale des Études Technologiques
Institut Supérieur des Études Technologiques de
Djerba
Département Technologies de l'informatique



Code projet	DSI
----------------	-----

Rapport de
STAGE DE FIN D'ETUDES

Présenté en vue de l'obtention du diplôme de
**Licence Nationale en Technologies de
l'Informatique**

Parcours : DSI

**Conception et développement d'un système de
gestion des entrées patients dans les unités de
soin dans un établissement de santé**

Elaboré par : Ltifi Safaa

Elaboré par : M. Blaghgi Mejdi

Effectué à :

Entreprise : ISS4U

Encadreur : M. Sahraoui AbdelKaoui

Année universitaire : 2023/2024

Dédicaces

Au Dieu le tout puissant, notre créateur

À mes chers parents

Les fondations de mon existence, les rayons de lumière qui guident ma route et m'enveloppent de tendresse et d'affection. Je ressens une gratitude infinie envers eux pour leur soutien indéfectible et les sacrifices consentis tout au long de mon parcours éducatif.

À toute ma famille

Je m'engage à faire tout mon possible pour demeurer une source de fierté à vos yeux, nourrissant l'espoir de ne jamais vous décevoir. Aucune déclaration ne suffit à exprimer pleinement mes émotions, mais je prie pour que Dieu vous protège, vous accorde une santé florissante et une longue vie à vos côtés.

À tous mes chers amis

Que j'aime tant et dont leur présence m'a rendu la vie plus agréable, Puisse l'Éternel nous guider vers le succès et la prospérité.

A handwritten signature in cursive script, reading "Ltfi Safaa", with a small decorative flourish to the left of the "L".

Remerciements

Dans cette opportunité merveilleuse, je tiens à exprimer ma profonde gratitude pour le soutien et les conseils précieux que j'ai reçu tout au long de ce projet.

Au terme de ce projet, je tiens à remercier dans un premier temps M. Blaghgi Mejdi mon encadrant académique pour son suivi attentif et ses orientations éclairées. Sa présence bienveillante et ses conseils avisés ont grandement contribué à l'aboutissement de ce travail.

Je souhaite également témoigner toute ma reconnaissance envers M. Sahraoui Abdelkaoui, mon encadrant professionnel, pour son accueil chaleureux et la confiance qu'il m'a accordée dès le début de cette expérience enrichissante. Son expertise et son soutien constant ont été d'une valeur inestimable pour la réussite de ce projet.

Finalement, je souhaite exprimer ma sincère reconnaissance envers les membres de jury, j'espère sincèrement que ce travail acharné répondra pleinement à leurs attentes et qu'il sera à la hauteur de la confiance qu'ils ont placée en moi.

Avec beaucoup de respect, je ne manquerai jamais d'exprimer ma sincère gratitude et mes plus chaleureux remerciements à tous les enseignants qui ont joué un rôle essentiel dans mon parcours éducatif.

Sommaire

Introduction générale.....	1
Chapitre 1 : Présentation générale du projet	2
Introduction	3
1. Cadre du projet	3
2. Présentation de l'organisme d'accueil	3
2.1 Secteur d'activités	3
3. Contexte du projet	4
3.1 Contexte et problématique	4
3.2 Objectif du projet	5
4. Étude de l'Existant	5
4.1 Description de l'Existant	5
4.2 Outils Similaires.....	5
5. Solution Proposée	6
6. Choix de méthodologie de travail.....	7
7. Choix Technologiques.....	8
7.1 Outils logiciels	8
7.2 Langages de programmation	9
7.3 Frameworks.....	9
Conclusion.....	10
Chapitre 2 : Etude préalable	11
Introduction	12
1. Analyse des besoins	12
1.1 Identification des acteurs	12
1.2 Les besoins fonctionnels	12
1.3 Les besoins non fonctionnels	13
2. Architecture MVC et Micro services.....	14
2.1 Architecture MVC	14
2.1.1. Caractéristiques de l'architecture MVC	14
2.1.2. Avantage MVC.....	15
2.2. Intégration des Micro services	16
2.2.1. Avantages de l'architecture MVC avec Micro services	16
3. Backlog produit	16
4. Diagramme de cas d'utilisation Global	20
5. Diagramme de classes globale.....	22
Conclusion.....	22
Chapitre 3 : Gestion des comptes	24

Introduction	25
1. Backlog du sprint 1	25
1.1 Affectation des cas d'utilisations pour les acteurs	25
1.2 Sprint backlog	25
2. Spécification fonctionnelle	26
2.1 Diagramme de cas d'utilisation global de sprint 1	26
2.2 Description textuelle des cas d'utilisation	27
2.3 Description comportementale des histoires du sprint 1	30
2.4 Diagramme de classe de sprint 1	33
3. Test et interfaçage	33
3.1 Test sprint 1	33
3.2 Interfaçages	33
Conclusion	38
Chapitre 4 : Gestion des ressources matérielles	39
Introduction	40
1. Backlog du sprint 2	40
1.1 Affectation des cas d'utilisations pour les acteurs	40
1.2 Sprint backlog	40
2. Raffinement du Sprint	42
2.1 Identification des besoins fonctionnels	42
2.2 Modèle global des cas d'utilisation du Sprint	43
2.2.1 Diagramme de cas d'utilisation du sprint	43
2.2.2 Raffinement des cas d'utilisations	43
2.3 Diagramme de séquences	48
2.4 Diagramme de classes Sprint 2	51
3. Test et interfaçage	51
3.1 Test sprint 2	51
3.2 Interfaçages	52
4. Conclusion	55
Chapitre 5 : Gestion de coordination des soins et des données	56
Introduction	57
1. Backlog du sprint 3	57
1.1 Affectation des cas d'utilisations pour les acteurs	57
1.2 Sprint backlog	57
2. Raffinement du Sprint	59
2.1 Identification des besoins fonctionnels	59
2.2 Modèle global des cas d'utilisation du deuxième sprint	60

2.2.1	Diagramme des cas d'utilisation du deuxième sprint.....	60
2.2.2	Raffinement des cas d'utilisation du deuxième sprint.....	60
2.3	Diagramme de séquence de sprint 3	65
2.4	Diagramme de classe de sprint 3	65
3.	Test et interfaçage.....	66
3.1	Test sprint 3.....	66
3.2	Interfaçages	66
	Conclusion.....	69
	Chapitre 6 : Gestion des flux patients	70
	Introduction	71
1.	Backlog du sprint 4.....	71
1.1.	Affectation des cas d'utilisations pour les acteurs	71
1.2.	Sprint backlog	71
2.	Raffinement du Sprint	74
2.1	Identification des besoins fonctionnels	74
2.2	Modèle global des cas d'utilisation du Sprint.....	76
2.2.1	Diagramme de cas d'utilisation du sprint	76
2.2.2	Raffinement des cas d'utilisations	76
2.3	Diagramme de séquence sprint 4	83
2.4	Diagrammes de classe Sprint 4	84
3.	Intégration de WebSocket et Gestion des Notifications :	84
3.1	Configuration de WebSocket	84
3.2	Gestion des Notifications	85
3.3	Processus de Notification.....	85
3.4	Nettoyage Automatique des Notifications	85
3.5	Avantages et Perspectives	85
4.	Test et interfaçage.....	86
4.1	Test sprint 3.....	86
4.2	Interfaçages	86
	Conclusion.....	88
	Conclusion générale	89
	Webographie	90

Liste des figures

Figure 1 : Logo de la société	3
Figure 2: La méthode SCRUM	7
Figure 3 : Architecture MVC	15
Figure 4 : Diagramme de cas d'utilisation global	21
Figure 5:Diagramme de classes global.....	22
Figure 6: Diagramme d'utilisation globale de sprint 1	27
Figure 7: Diagramme de séquence de cas d'utilisation "S'authentifier"	31
Figure 8 : Diagramme de séquence de cas d'utilisation "Mettre à jour profile".....	32
Figure 9: Diagramme de séquence de cas d'utilisation "Récupérer mot de passe oublié"	32
Figure 10 : Diagramme de classe de Sprint 1.....	33
Figure 11:Interface « Authentification »	34
Figure 12:Interface « Inscription ».....	34
Figure 13:Interface « Oublié mot de passe ».....	35
Figure 14:Interface « Oublié mot de passe via mail ».....	35
Figure 15:Interface « Tableau de bord Administrateur 1 »	35
Figure 16: Interface « Tableau de bord Administrateur 2 »	36
Figure 17: Interface « Tableau de bord Doctor » 1	36
Figure 18: Interface « Tableau de bord Doctor » 2	36
Figure 19:Interface « Tableau de bord Secrétaire » 1	37
Figure 20:Interface « Tableau de bord 2 Secrétaire »	37
Figure 21:Interface « Mettre à jour profil »	38
Figure 22: Diagramme de cas d'utilisation Sprint 2	43
Figure 23: Diagramme de cas d'utilisation Gérer les unités	43
Figure 24 : Diagramme de cas d'utilisation Gérer les chambres	45
Figure 25 : Diagramme de cas d'utilisation Gérer les lits.....	46
Figure 26: Diagramme de séquence "Ajouter unité de soins".....	48
Figure 27: Diagramme de séquence " Supprimer un lit".....	49
Figure 28: Diagramme de séquence "Modifier chambre"	50
Figure 29 : Diagramme de classe Sprint 2	51
Figure 30:Interface « Liste des unités des soins ».....	52
Figure 31:Interface de « Ajouter Unité de soins »	52

Liste des figures

Figure 32:Interface « Filtrage avancé ».....	53
Figure 33:Interface « Liste des unités des Chambres ».....	53
Figure 34:Interface « Ajouter une chambre dans une unité de soins »	54
Figure 35:Interface « Liste des lits ».....	54
Figure 36:Interface « Ajouter un lit dans une chambre ».....	55
Figure 37: Diagramme de cas d'utilisation Sprint 3	60
Figure 38: Diagramme de cas d'utilisation Gérer les patients	61
Figure 39 : Diagramme de cas d'utilisation Gérer les rendez-vous	62
Figure 40 : Diagramme de cas d'utilisation Gérer les rapports.....	64
Figure 41: Diagramme de séquence "Ajouter rendez-vous"	65
Figure 42: Diagramme de classe de Sprint 3.....	65
Figure 43:Interface « Liste des patients d'un docteur »	66
Figure 44:Interface « Liste des patients non examinés pour Secrétaire »	67
Figure 45:Interface « Ajouter patient pour Secrétaire ».....	67
Figure 46:Interface « Rendez-vous pour Secrétaire »	68
Figure 47: Interface « Rapports des patients pour Secrétaire ».....	68
Figure 48 : Diagramme de cas d'utilisation Sprint 4	76
Figure 49 : Diagramme de cas d'utilisation Gérer les entrées et sorties des patients	76
Figure 50 : Diagramme de cas d'utilisation Gérer les réservations	78
Figure 51 : Diagramme de cas d'utilisation Gérer les prolongations et les déplacements.....	80
Figure 52: Diagramme de cas d'utilisation Gérer les archives	81
Figure 53: Diagramme de séquence "Réserver lit"	83
Figure 54 : Diagramme de classe Sprint 4	84
Figure 55 : Table Notification.....	85
Figure 56: Interface « Patients en attentes »	86
Figure 57:Interface « Patients occupants des lits »	86
Figure 58: Interface « Patients archivés »	87
Figure 59: Interface « Détails patient ».....	87
Figure 60: Interface « Réserver patient »	88

Liste des tableaux

Tableau 1 : Les besoins fonctionnels	13
Tableau 2: Product Backlog	20
Tableau 3: Classification des cas d'utilisations par acteur sprint 1.....	25
Tableau 4 : Sprint 1 Backlog.....	26
Tableau 5 : Description textuelle du cas d'utilisation "S'inscrire"	27
Tableau 6: Description textuelle du cas d'utilisation "S'authentifier"	28
Tableau 7: Description textuelle du cas d'utilisation "Récupérer mot de passe oublié"	29
Tableau 8: Description textuelle du cas d'utilisation "Confirmer compte par mail"	29
Tableau 9: Description textuelle du cas d'utilisation "Mettre à jour profil"	30
Tableau 10 : Test du sprint1	33
Tableau 11:Classification des cas d'utilisations par acteur sprint 2	40
Tableau 12 : Backlog Sprint 2.....	42
Tableau 13 : Description textuelle du cas d'utilisation	44
Tableau 14 : Description textuelle du cas d'utilisation Gérer les chambres	46
Tableau 15: Description textuelle du cas d'utilisation Gérer les lits.....	47
Tableau 16:Test du sprint 2	51
Tableau 17:Classification des cas d'utilisations par acteur sprint 3	57
Tableau 18: Backlog Sprint 3	59
Tableau 19 : Description textuelle du cas d'utilisation Gérer les patients	62
Tableau 20 : Description textuelle du cas d'utilisation Gérer les rendez-vous	63
Tableau 21 : Diagramme de cas d'utilisation Gérer les rapports	64
Tableau 22: Test du sprint 3	66
Tableau 23:Classification des cas d'utilisations par acteur Sprint 4.....	71
Tableau 24 : Backlog Sprint 4.....	74
Tableau 25 : Description textuelle du cas d'utilisation Gérer les entrées et sorties des patients	78
Tableau 26 : Description textuelle du cas d'utilisation Gérer les réservations	79
Tableau 27 : Description textuelle du cas d'utilisation Gérer les prolongations et les déplacements	81
Tableau 28 : Description textuelle du cas d'utilisation Gérer les archives	82
Tableau 29: Test sprint 4	86

Introduction générale

Dans le paysage complexe de la prestation de soins de santé, la gestion efficace des lits représente un défi majeur. La demande croissante de services médicaux dans les établissements de santé entraîne une pression constante sur les ressources disponibles, en particulier les lits d'hospitalisation. Cette pression est évoluée par des facteurs tels que les urgences médicales, les admissions imprévues, les besoins de soins continus et les périodes de pointe saisonnières.

Dans de nombreux établissements, la gestion des lits demeure un processus manuel et fragmenté, ce qui entraîne des inefficacités et des retards. Il est fréquent que l'absence de surveillance en temps réel de l'occupation des lits et des flux de patients entrants et sortants entraîne une utilisation inefficace des ressources, des retards dans la prise en charge des patients et une baisse de la qualité des soins.

Face à ces défis, il devient nécessaire pour les établissements de santé d'adopter des méthodes novatrices et technologiques afin d'améliorer cette situation. Et dans ce prétexte l'idée est venue : Développement d'un système de gestion des entrées et sorties des patients dans les unités de soins dans un établissement de santé. C'est une solution prometteuse pour surmonter les défis actuels et améliorer la qualité des soins en combinant une analyse avancée des données, des fonctionnalités de planification prévisionnelle et une interface utilisateur conviviale, ce système offre la possibilité de transformer radicalement la manière dont les établissements de santé gèrent leurs ressources et répondent aux besoins des patients.



Rester avec moi pour présenter de manière plus approfondir cette idée dans ce rapport.

Chapitre 1 : Présentation générale du projet

Introduction

Ce chapitre se concentre sur la mise en perspective du projet réalisé lors du stage de fin d'études. Je vais commencer par présenter l'organisation d'accueil. Ensuite, je vais procéder à une analyse de la situation actuelle en présentant les différentes options disponibles sur le marché. Enfin, je vais terminer par une explication de la méthodologie de gestion de projet que j'ai utilisée pour la création de mon application.

1. Cadre du projet

Ce travail se situe dans le cadre d'un projet de fin d'études en vue de l'obtention du diplôme de licence nationale en informatique - Parcours : Développement des systèmes d'informations.

Mon projet a eu lieu au sein de l'entreprise « ISS4U », d'une durée de 4 mois, intitulé : Conception et développement d'un système de gestion des entrées patients dans les unités de soin dans un établissement de santé.

2. Présentation de l'organisme d'accueil

Dans cette partie, nous présentons l'entreprise, ainsi que son bref historique et ses secteurs d'activités.



Figure 1 : Logo de la société

ISS4U, est une entreprise de premier plan dans le domaine des solutions technologiques, qui s'engage à fournir des solutions logicielles intelligentes pour répondre aux besoins uniques des entreprises et des organisations.

2.1 Secteur d'activités

Les services d'ISS4U sont spécifiquement adaptés aux besoins des entreprises opérant dans divers secteurs :

- Développement de logiciels personnalisés : ISS4U crée des solutions sur mesure pour répondre aux besoins spécifiques des entreprises, en optimisant leurs processus métiers et en tirant parti des dernières avancées technologiques.

- Services d'application : ISS4U propose une variété de services pour soutenir le développement, la gestion et l'amélioration des applications, garantissant ainsi la performance et la fiabilité des systèmes informatiques des clients.
- Modernisation des applications : ISS4U accompagne les entreprises dans la modernisation de leurs applications existantes, en les adaptant aux nouvelles exigences technologiques et aux évolutions du marché.
- Tests et assurance qualité : ISS4U assure la qualité des logiciels développés en fournissant des services de test approfondis pour garantir la fiabilité, la sécurité et la performance des applications.
- Conception UI/UX : ISS4U propose des services de conception d'interfaces utilisateur et d'expérience utilisateur pour créer des produits numériques attrayants et conviviaux, favorisant ainsi l'engagement des utilisateurs et la satisfaction client.
- Science des données, analyse de données et big data : ISS4U exploite les données pour fournir des insights précieux aux entreprises, les aidant à prendre des décisions éclairées et à tirer parti des opportunités de croissance dans un environnement commercial en constante évolution.

3. Contexte du projet

Dans cette section, nous présentons le cadre général de notre sujet, la problématique et les objectifs demandés.

3.1 Contexte et problématique

Dans les établissements de santé, la gestion des lits est cruciale mais souvent inefficace dans la méthode traditionnelle. Les processus manuels ou semi-automatisés peuvent causer des erreurs et des retards, affectant la qualité des soins. La difficulté de gestion des ressources, notamment des lits disponibles, est une problématique récurrente. Les variations du nombre de patients et les besoins spécifiques compliquent la tâche.

Cette gestion manuelle peut entraîner une utilisation inefficace des lits et des perturbations dans leur occupation, impactant l'efficacité opérationnelle de l'établissement.

Face à ces défis, la solution proposée est une gestion intelligente des lits. Ce projet vise à fournir une visualisation détaillée de l'occupation des lits, une allocation efficace des lits disponibles et une planification précise des entrées et sorties des patients. Intégrant un mécanisme de recherche opérationnelle, cette solution garantira une utilisation optimale des ressources, améliorant ainsi l'efficacité de l'établissement de santé.

3.2 Objectif du projet

Notre objectif est de développer une solution informatique complète pour optimiser la gestion des lits dans un établissement de santé. Cette solution englobe également la gestion des entrées et sorties des patients, offrant ainsi une approche holistique pour optimiser l'utilisation des ressources hospitalières.

Notre solution fournira une visualisation détaillée de l'occupation des lits, permettant aux responsables de mieux comprendre la disponibilité des lits et d'optimiser leur allocation. Elle inclura également un système de planification des entrées et sorties des patients pour une gestion proactive du flux de patients.

De plus, elle offrira des fonctionnalités avancées pour associer les patients à chaque lit, prenant en compte divers facteurs tels que l'état de santé du patient en consultant son rapport et les poids des lits et la disponibilité des lits bien sûres.

On outre, la solution intégrera la prise de rendez-vous et la gestion des examens médicaux, permettant une coordination fluide et efficiente des soins. En optimisant ces processus, notre solution contribuera à améliorer l'efficacité opérationnelle de l'établissement de santé et la qualité des soins prodigués aux patients.

En résumé, cette solution informatique complète permettra une gestion proactive et efficiente des lits, des admissions, des sorties et des rendez-vous, contribuant ainsi à une utilisation optimale des ressources hospitalières et à une amélioration significative des soins aux patients.

4. Étude de l'Existant

4.1 Description de l'Existant

Différentes applications déjà disponibles sur le marché pour la gestion des lits hospitaliers proposent une multitude de fonctionnalités afin de satisfaire les exigences des établissements de santé.

L'objectif de ces applications est d'améliorer la répartition des lits, de fournir une surveillance en temps réel de l'occupation des unités de soins, et de simplifier la planification des admissions et des sorties des patients.

Mais, malgré les avantages offerts par les applications existantes de gestion des lits hospitaliers, plusieurs limitations et points critiques peuvent être identifiés.

4.2 Outils Similaires

Voici quelques illustrations d'applications similaires pour la gestion des lits hospitaliers:

BedWatch : est un outil de gestion des lits hospitaliers qui propose plusieurs fonctionnalités, comme la visualisation en temps réel de l'occupation des lits, la gestion des entrées sorties, ainsi que des outils d'analyse de données pour optimiser l'utilisation des ressources. Toutefois, il y a eu des critiques concernant sa convivialité et sa capacité à s'ajuster aux besoins particuliers de chaque établissement de santé.

TeleTracking : est la première plateforme mondiale d'opérations de soins de santé très prisé dans la gestion des lits hospitaliers dans les organisations de soins de santé et les gouvernements. Malgré son succès, elle suscite des critiques en raison de son coût élevé et de sa complexité d'utilisation.

Cerner CareAware Capacity Management : créée par la société Cerner Corporation. Elle propose diverses caractéristiques comme la représentation en temps réel de l'occupation des lits, ainsi que des outils d'analyse avancés pour maximiser l'efficacité de l'utilisation des ressources disponibles. Toutefois, certaines critiques ont été formulées quant à sa convivialité et à sa capacité à s'intégrer de manière harmonieuse aux systèmes déjà en place.

5. Solution Proposée

L'application projetée pour la gestion des entrées sorties des patients dans un établissement de santé est parviennent sera conçue pour répondre aux besoins spécifiques des établissements de santé et pour simplifier leur gestion quotidienne. C'est une application riche en termes de fonctionnalités, développée en utilisant des technologies modernes et rapides ainsi avec des interfaces dynamiques et faciles à utiliser.

Voici les principales Fonctionnalités :

- ✓ **Visualisation en Temps Réel** : Offre une vue en temps réel sur l'occupation des lits et les mouvements des patients dans l'établissement.
- ✓ **Allocation Optimale des Lits** : Permet d'attribuer efficacement les lits aux patients en fonction de leurs besoins médicaux et des disponibilités.
- ✓ **Planification Avancée** : Permet de planifier les entrées et les sorties des patients, ainsi que les mouvements internes entre les différentes unités de soins.
- ✓ **Flexibilité et Adaptabilité** : Offre la possibilité de modifier les dates de sortie et de gérer les imprévus de manière flexible.

En résumé, cette application représente une solution complète et efficace pour la gestion des entrées et sorties des patients dans les établissements de santé. Son développement repose sur une approche centrée sur les besoins des utilisateurs et sur l'utilisation de technologies de pointe,

garantissant ainsi une expérience utilisateur optimale et une gestion hospitalière simplifiée.

6. Choix de méthodologie de travail

Pour le développement de cette application on a opté pour SCRUM. C'est une méthode agile de gestion de projets informatiques privilégiant la communication, et permet à l'équipe SCRUM de travailler de manière collaborative et itérative pour atteindre les objectifs de projets.

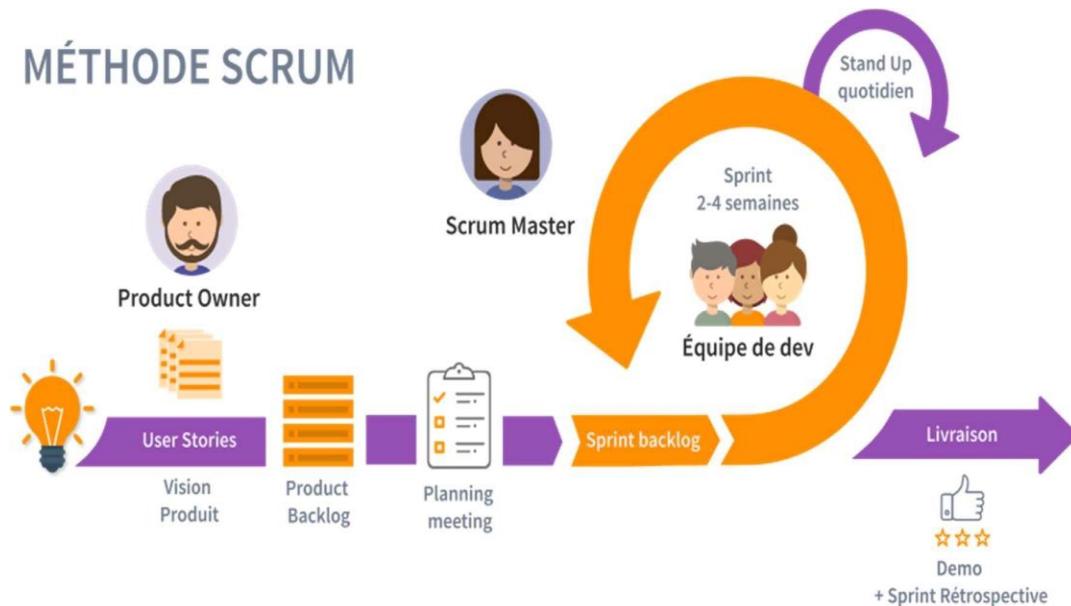


Figure 2: La méthode SCRUM

Dans le cadre de notre projet, nous avons mis en œuvre les pratiques SCRUM suivantes :

- **Daily Meetings (Réunions Quotidiennes)** : Nous avons organisé des réunions quotidiennes le lundi, mardi et mercredi. Ces réunions ont permis à l'équipe SCRUM de se synchroniser, de discuter des tâches accomplies, des obstacles rencontrés et de planifier les prochaines étapes à réaliser, ces réunions sont organisées sur la plateforme **Slack**.
- **Itérations (Sprints)** : Nous avons découpé le développement de l'application en itérations appelées "sprints", sur **Confluence**. Chaque sprint avait des objectifs clairs et définis, permettant à l'équipe de se concentrer sur des livrables spécifiques dans des délais déterminés.
- **Product Backlog (Liste des Fonctionnalités)** : Nous avons maintenu un backlog de produit, qui est une liste priorisée des fonctionnalités à développer. Cette liste a été régulièrement révisée et mise à jour en fonction des besoins changeants du projet et des

retours d'utilisateur.

- **Sprint Planning (Planification du Sprint)** : Avant le début de chaque sprint, l'équipe SCRUM s'est réunie pour planifier les tâches à réaliser pendant le sprint sur **Jira**.
- **Review et Rétrospective** : À la fin de chaque sprint, nous avons organisé une revue pour examiner les fonctionnalités développées et recueillir les commentaires des parties prenantes. Nous avons également tenu une rétrospective pour évaluer le déroulement du sprint, identifier les points forts et les axes d'amélioration, et apporter des ajustements pour les sprints suivants.

7. Choix Technologiques

Voici les outils logiciels utilisés tout au long de la conception et développement du projet.

7.1 Outils logiciels

- **Visual Studio 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, les snippets, la refactorisation du code. 
- **MySQL** : MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il fait partie des logiciels de gestion de base de données les plus utilisés au monde.[1] 
- **Postman** : Postman est un environnement de développement d'API qui aide les utilisateurs à créer, tester, documenter, surveiller et publier la documentation de leurs API.[2] 
- **SpringToolSuite4** : Spring Tool Suite 4 (STS 4) est un environnement de développement intégré (IDE) spécialement conçu pour les applications basées sur Spring.[3] 
- **PlantUML** : PlantUML est un outil logiciel utilisé pour créer des diagrammes à de nombreuses fins différentes, telles que l'architecture logicielle, les modèles de bases de données, la modélisation ER, les organigrammes, les diagrammes de séquence, les user stories basées sur du texte. 
- **Laragon** : Est un logiciel gratuit et open source qui permet de créer un environnement de développement local pour les applications web. Il fournit une plateforme complète comprenant Apache, MySQL, PHP et Perl, ce qui facilite l'installation et la configuration d'un serveur web sur votre propre machine. 

- **Bitbucket** : Est un outil d'hébergement du code et de collaboration basée sur Git, qui a été conçu pour les équipes. 
- **SourceTree** : Est un logiciel avec GUI permettant de quasiment tout gérer sans passer par le terminal tout en vous facilitant la tâche. 
- **Confluence** : Est une plateforme de collaboration en ligne qui offre des outils de partage de contenu, de gestion des connaissances et de communication pour une collaboration efficace entre équipes. 
- **Jira** : Est un logiciel de gestion de projets collaboratif avec des outils avancés pour la planification, le suivi et la gestion des tâches et projets logiciels développés par Atlassian. 

7.2 Langages de programmation

- **Java** : Java est un langage de programmation et une plate-forme informatique créée par Sun Microsystems. Utilisé par 9 millions de développeurs dans le monde, ce langage est nécessaire pour le fonctionnement de beaucoup d'applications et de sites Web. Il est la base de la plupart des applications en réseau : des jeux, des téléphones portables à Internet, du contenu web et des logiciels entreprises. 
- **TypeScript** : TypeScript est un langage de programmation open source développé par Microsoft. Il s'agit d'un sur-ensemble de JavaScript qui ajoute des fonctionnalités de typage statique au langage, permettant ainsi de détecter et prévenir les erreurs de manière plus efficace lors de la phase de développement. TypeScript est compatible avec tous les navigateurs modernes et peut être utilisé pour développer des applications tant côté serveur que côté client. 

7.3 Frameworks

- **Angular** : Angular est un Framework de développement d'applications web développé par Google. Il s'appuie sur TypeScript pour créer des applications web robustes et évolutives. Angular adopte le concept de composants réutilisables, permettant aux développeurs de construire des interfaces utilisateur interactives et dynamiques. 
- **Springboot** : Spring Boot est un Framework de développement JAVA. C'est une déclinaison du Framework classique de Spring qui permet essentiellement de réaliser des micro services. 

Conclusion

En résumé, ce chapitre a traité la contextualisation de mon projet de fin d'études et l'approche de travail adoptée pour le réaliser. Cette présentation a jeté des fondations solides pour les prochaines étapes, qui impliqueront la clarification précise des exigences du projet et l'explication détaillée des fonctionnalités de cette application.

Chapitre 2 : Etude préalable

Introduction

Ce chapitre revêt une importance capitale lors de la création d'un système, car il permet de déterminer et de préciser les exigences du projet. Il est constitué de différentes composantes essentielles : le backlog produit qui organise les différentes fonctionnalités à développer, l'analyse des besoins des utilisateurs, le style architectural du système, le diagramme des cas d'utilisation et classe global.

1. Analyse des besoins

1.1 Identification des acteurs

Dans cette phase, nous déterminons les divers intervenants impliqués dans l'utilisation de notre application. Après avoir analysé les différentes interactions tant internes qu'externes du système, nous avons estimé crucial de définir les acteurs suivants :

- **Secrétaire** : Il s'agit d'un utilisateur qui a la possibilité de gérer les patients et les flux des patients ainsi que prendre des rendez-vous pour les patients en les associant à un docteur à une date précise.
- **Administrateur** : L'administrateur gère les unités de soins, gère les chambres, les lits et les équipements.
- **Docteur** : Le docteur gère les rapports des patients qui lui sont associés et consulte les rapports des patients (sauf ceux qu'il a lui-même rédigés).

1.2 Les besoins fonctionnels

Les besoins fonctionnels correspondent aux demandes du client concernant les fonctionnalités nécessaires à intégrer dans le développement de l'application. Après plusieurs réunions avec le Product Owner, dirigées par Monsieur Abdelkoui SAHRAOUI, nous avons recensé ces exigences dans le tableau ci-dessous.

Acteurs	Description des besoins fonctionnels
Admin	S'authentifier Gérer les utilisateurs Gérer les unités de soins Gérer les chambres Gérer les lits Gérer les équipements Consulter la liste des docteurs Consulter la liste des secrétaires Consulter les statistiques
Secrétaire	S'authentifier Gérer les patients Gérer les entrées et sorties des patients Consulter les rapports Gérer les visites Gérer les déplacements des patients Gérer les prolongations des patients Consulter les statistiques
Docteur	S'authentifier Gérer les rapports Consulter les statistiques

Tableau 1 : Les besoins fonctionnels

1.3 Les besoins non fonctionnels

Les besoins non fonctionnels sont les critères qui définissent les performances, la fiabilité, la sécurité et d'autres aspects du système. Ils se résument dans les points suivants :

- **L'ergonomie** : Les interfaces utilisateur doivent présenter une structure claire et informative, avec une sélection appropriée de couleurs et de styles de texte. L'application doit être conçue pour être intuitive et conviviale, afin de garantir une utilisation simple et aisée pour les utilisateurs.
- **Convivialité** : l'application doit être facile à utiliser.
- **Sécurité** : L'accès à l'application doit être restreint et réservé uniquement aux utilisateurs authentifiés.

- **Performance** : Le temps de réponse de l'application doit être rapide afin de permettre une exécution efficace des différentes tâches.
- **Maintenabilité et évolutivité** : Le code de l'application doit être facile à entretenir afin de faciliter sa réutilisation et ses modifications ultérieures. Il doit être conçu de manière à être extensible.
- **Fiabilité** : l'application doit fonctionner correctement en donnant les bonnes informations à propos les tests.
- **Design** : Le thème de l'application doit être en harmonie avec les couleurs du logo de l'entreprise. Cela permet à l'application de se distinguer sur le marché et d'être reconnue comme un produit de ISS4U.

2. Architecture MVC et Micro services

2.1 Architecture MVC

Notre projet repose sur l'architecture MVC, un modèle de conception fondamental dans le développement web, séparant les responsabilités entre le Modèle, la Vue et le Contrôleur.

2.1.1. Caractéristiques de l'architecture MVC

Le cœur de notre architecture repose sur le modèle MVC (Modèle-Vue-Contrôleur), harmonieusement intégré avec Angular et Spring. Voici une vue d'ensemble des responsabilités de chaque composant :

- **Modèle (Model)** : Il englobe les données et la logique métiers essentiels à l'application. Dans notre mise en œuvre avec Spring, le modèle est représenté par des classes Java qui définissent les entités métier, assurant une gestion efficace des données.
- **Vue (View)** : Chargée de présenter les données à l'utilisateur de manière conviviale, la vue est un élément central de notre architecture. Avec Angular, nous avons pu créer des interfaces utilisateur dynamiques et réactives, offrant ainsi une expérience utilisateur optimale.
- **Contrôleur (Controller)** : Fonctionnant comme une liaison entre le modèle et la vue, le contrôleur joue un rôle crucial dans la gestion des interactions utilisateur. Dans notre implémentation avec Spring, les contrôleurs traitent les requêtes HTTP entrantes, interagissent avec le modèle pour accéder aux données nécessaires et renvoient les résultats à la vue correspondante. De plus, ils utilisent des API RESTful pour assurer une communication efficace et cohérente avec d'autres composants de l'application, garantissant ainsi une architecture flexible et extensible.

Cette combinaison harmonieuse de l'architecture MVC avec l'utilisation d'API RESTful dans les contrôleurs confère à notre application une structure solide et adaptable, répondant aux exigences modernes en matière de développement web.

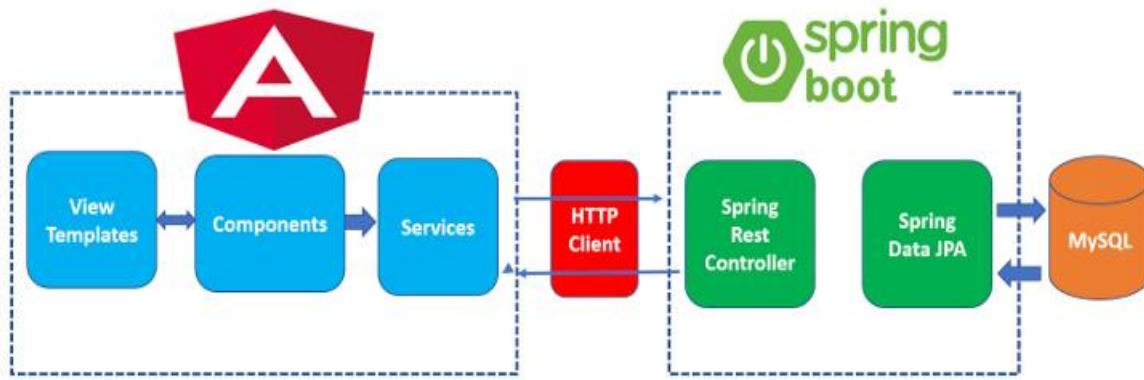


Figure 3 : Architecture MVC

2.1.2. Avantage MVC

Dans cette section, nous soulignerons les bénéfices de l'architecture MVC :

- **Modèle (Model)** : Le modèle prend en charge la gestion des données et la logique métier de l'application. Il assure la représentation et la manipulation des données de manière efficace, garantissant ainsi la cohérence et la fiabilité des fonctionnalités offertes.
- **Vue (View)** : La vue se concentre sur l'interface utilisateur, présentant de manière conviviale les données provenant du modèle. Elle permet aux utilisateurs d'interagir intuitivement avec l'application, favorisant ainsi une expérience utilisateur agréable et efficace.
- **Contrôleur (Controller)** : Le contrôleur orchestre les interactions utilisateur et facilite la communication entre le modèle et la vue. En traitant les requêtes de l'utilisateur, en récupérant les données nécessaires du modèle et en fournissant des réponses appropriées à la vue, il contribue à maintenir un flux d'interaction fluide et réactif.

L'architecture MVC offre une séparation claire des responsabilités entre ces différents composants, ce qui simplifie la maintenance et l'évolution de l'application. Cette structuration permet également une collaboration efficace entre les équipes de développement travaillant sur différentes parties de l'application.

2.2. Intégration des Micro services

Nous avons également adopté une architecture basée sur les micro services, où l'application est décomposée en services autonomes et indépendants.

L'intégration des micro services a été réalisée avec soin à l'aide de Spring Boot, en utilisant des technologies modernes telles que :

- **Eureka** : Eureka est un serveur de registre de services développé par Netflix. Il offre un mécanisme de découverte de service permettant à chaque micro services de s'enregistrer auprès du registre lors de son démarrage. Les autres micro services peuvent ensuite interroger Eureka pour découvrir et communiquer avec les services disponibles, facilitant ainsi la communication entre les différents composants d'une architecture micro services.
- **Config Server** : Le Config Server est un composant centralisé qui stocke et distribue la configuration aux différents micro services de manière dynamique.
- **Gateway Server** : Le Gateway Server agit comme un point d'entrée unique pour les requêtes clients entrantes et dirige le trafic vers les micro services appropriés en fonction des règles de routage configurées.
- **Discovery Server** : Le Discovery Server, souvent associé à Eureka, est un serveur qui permet aux micro services de s'enregistrer lors de leur démarrage et de découvrir les autres services disponibles dans l'architecture micro services.

2.2.1. Avantages de l'architecture MVC avec Micro services

Cette intégration offre plusieurs avantages, notamment :

- **Scalabilité** : La capacité à faire évoluer chaque composant indépendamment simplifie le déploiement et la gestion à grande échelle.
- **Flexibilité** : La séparation des services facilite l'évolutivité de l'application. Les équipes peuvent travailler sur des services spécifiques sans impacter les autres parties.
- **Isolation des erreurs** : En cas de défaillance d'un micro service, les autres parties de l'application restent opérationnelles, améliorant ainsi la résilience globale du système.

Cette approche intégrée de l'architecture MVC avec les micro services offre une base solide pour notre projet de développement web, combinant les meilleures pratiques de conception avec la flexibilité et la robustesse des architectures modernes.

3. Backlog produit

Un backlog produit est une liste hiérarchisée de tâches pour l'équipe de développement,

contenant les fonctionnalités attendues du produit sous forme d'histoires utilisateur.

Priorisées par le Product Owner selon la méthode MoSCoW, ces tâches sont créées à partir des exigences et de la feuille de route du produit.

- **M:** «Must Have», il doit être accompli. L'exigence est cruciale. Sans cela, le projet échoue. C'est une priorité majeure.
- **S:** «Should Have», c'est une exigence fondamentale, à accomplir dans la mesure du possible. Toutefois, si elle n'est pas respectée, il est nécessaire de la contourner et de la remplir plus tard.
- **C:** «Could Have», c'est une exigence fondamentale, à accomplir dans la mesure du possible. Toutefois, si elle n'est pas respectée, il est nécessaire de la contourner et de la remplir plus tard.
- **W:** « Won't Have this time but would like in the future », se sont les tâches supplémentaires, très secondaires.

Le tableau ci-dessous représente l'ensemble des user stories à réaliser durant le projet :

Module	Fonctionnalité	Id	User Story	P
Sprint 1	Gestion des utilisateurs	1	En tant qu'utilisateur, je souhaite m'authentifier.	M
		2	En tant qu'utilisateur je souhaite modifier mon compte.	M
		3	En tant qu'administrateur je souhaite supprimer un utilisateur.	M
		4	En tant qu'administrateur je souhaite ajouter un utilisateur.	W
		5	En tant qu'un utilisateur je souhaite modifier mes informations.	M
		6	En tant qu'administrateur je souhaite consulter la liste des utilisateurs.	M
		7	En tant qu'un utilisateur je souhaite s'inscrire	M
	Confirmer compte par mail	8	En tant qu'utilisateur je souhaite valider mon compte par mail	M
	Réinitialiser mot de passe	9	En tant qu'utilisateur, je souhaite réinitialiser mon mot de passe si je l'ai oublié	M
Sprint 2	Gestion des unités de soins	1	En tant qu'administrateur je souhaite consulter la liste des unités de soins	M
		2	En tant qu'administrateur, je souhaite voir les détails d'une unité de soin et la liste des équipements associés à cette unité de soins.	M

Gestion des chambres		3	En tant qu'administrateur, je souhaite pouvoir ajouter une nouvelle unité de soins et associer des équipements à cette unité.	M
		4	En tant qu'administrateur, je souhaite pouvoir modifier une unité de soins en ajoutant ou supprimant aussi des équipements associés à cette unité.	S
		5	En tant qu'administrateur, je souhaite pouvoir supprimer une unité de soins en supprimant tous les équipements associés à cette unité.	M
		6	En tant qu'administrateur, je souhaite consulter la liste des chambres d'une unité de soins spécifique.	M
		7	En tant qu'administrateur, je souhaite consulter les détails d'une chambre spécifique dans une unité de soins.	M
		8	En tant qu'administrateur, je souhaite ajouter une chambre dans une unité de soins.	M
		9	En tant qu'administrateur, je souhaite supprimer une chambre dans une unité de soins.	M
		10	En tant qu'administrateur, je souhaite modifier une chambre dans une unité de soins.	M
		11	En tant qu'administrateur, je souhaite consulter toutes les chambres dans l'établissement.	S
	Gestion des lits	1	En tant qu'administrateur, je souhaite consulter la liste des lits d'une chambre spécifique.	M
		2	En tant qu'administrateur, je souhaite ajouter un lit dans une chambre et lui associer des équipements spécifiques.	M
		3	En tant qu'administrateur, je souhaite supprimer un lit dans une chambre.	M
		4	En tant qu'administrateur, je souhaite consulter les détails d'un lit dans une chambre spécifique et consulter la liste des équipements associés à ce lit.	M
		5	En tant qu'administrateur, je souhaite pouvoir modifier les détails d'un lit dans une chambre spécifique en ajoutant ou supprimant des équipements associés à ce lit.	S

Sprint 3	Gestion des patients	6	En tant qu'administrateur, je souhaite consulter tous les lits dans l'établissement.	M
		1	En tant qu'un secrétaire je souhaite consulter la liste des patients réservés.	M
		2	En tant qu'un secrétaire je souhaite consulter la liste des patients en attentes de réservation.	M
		3	En tant qu'un secrétaire je souhaite consulter la liste des patients non examinés.	M
		4	En tant qu'un secrétaire je souhaite ajouter un patient.	M
		5	En tant qu'un secrétaire je souhaite modifier un patient.	M
		6	En tant que secrétaire je souhaite supprimer un patient non examiné.	M
		7	En tant que secrétaire je souhaite consulter les détails d'un patient.	M
		8	En tant que secrétaire je souhaite affecter un filtrage avancé sur les patients par nom, prénom et date de naissance.	M
Gestion des rendez-vous		1	En tant qu'un secrétaire je souhaite prendre un rendez-vous pour un nouveau patient ou un patient libéré.	M
		2	En tant que docteur, je souhaite avoir accès à une liste des patients dont la date de rendez-vous est passée, qu'ils aient été examinés ou non.	M
		3	En tant que docteur, je souhaite avoir accès à une liste des patients dont le rendez-vous est à venir.	M
Gestion des rapports		1	En tant qu'un docteur je souhaite créer un rapport pour un patient.	M
		2	En tant qu'un docteur je souhaite consulter les rapports des patients qui me sont attribués.	M
		3	En tant qu'un secrétaire je souhaite consulter tous les rapports des patients réservés et en attente de réservation des différents docteurs.	M
Sprint 4	Gestion des entrées/sorties patients	1	En tant qu'un secrétaire je souhaite entrer la date d'entrée et sortie réelle.	M
		2	En tant que secrétaire je souhaite saisir l'état de sortie d'un patient.	M
	Gestion des réservations	1	En tant qu'un secrétaire je souhaite avoir une recherche opérationnelle pour optimiser l'allocation des lits	M
		2	En tant que secrétaire je souhaite réserver un lit pour un patient dans une unité et chambre	M

		spécifique.	
	3	En tant qu'un secrétaire je souhaite saisir les détails d'un accompagnant pour un patient spécifique.	M
	4	En tant qu'un secrétaire je souhaite consulter les détails d'un accompagnant pour un patient spécifique.	M
	5	En tant qu'un secrétaire je souhaite modifier les détails d'un accompagnant pour un patient spécifique.	M
	6	En tant qu'un secrétaire je souhaite consulter les détails d'un lit pour un patient spécifique.	M
Gestion des déplacements et des prolongations	1	En tant qu'un secrétaire je souhaite déplacer un patient d'une chambre à une autre dans la même unité ou dans une unité différente.	M
	2	En tant que secrétaire je souhaite conserver les historiques des changements de chaque patient.	M
	3	En tant qu'un secrétaire je souhaite déplacer l'accompagnant avec le patient selon le type de chambre choisie.	M
	4	En tant qu'un secrétaire je souhaite prolonger la date de sortie provisionnelle d'un patient.	M
	5	En tant que secrétaire je souhaite conserver les historiques des prolongations de chaque patient.	M
Gestion des archives	1	En tant qu'un secrétaire je souhaite consulter la liste des patients libérés.	S
	2	En tant qu'un secrétaire je souhaite consulter les rapports des patients libérés.	S

Tableau 2: Product Backlog

4. Diagramme de cas d'utilisation Global

La figure ci-dessous représente le diagramme de cas d'utilisation global qui regroupe les fonctionnalités demandées. Il est important de noter que toutes les fonctionnalités représentées nécessitent une authentification préalable de l'utilisateur.

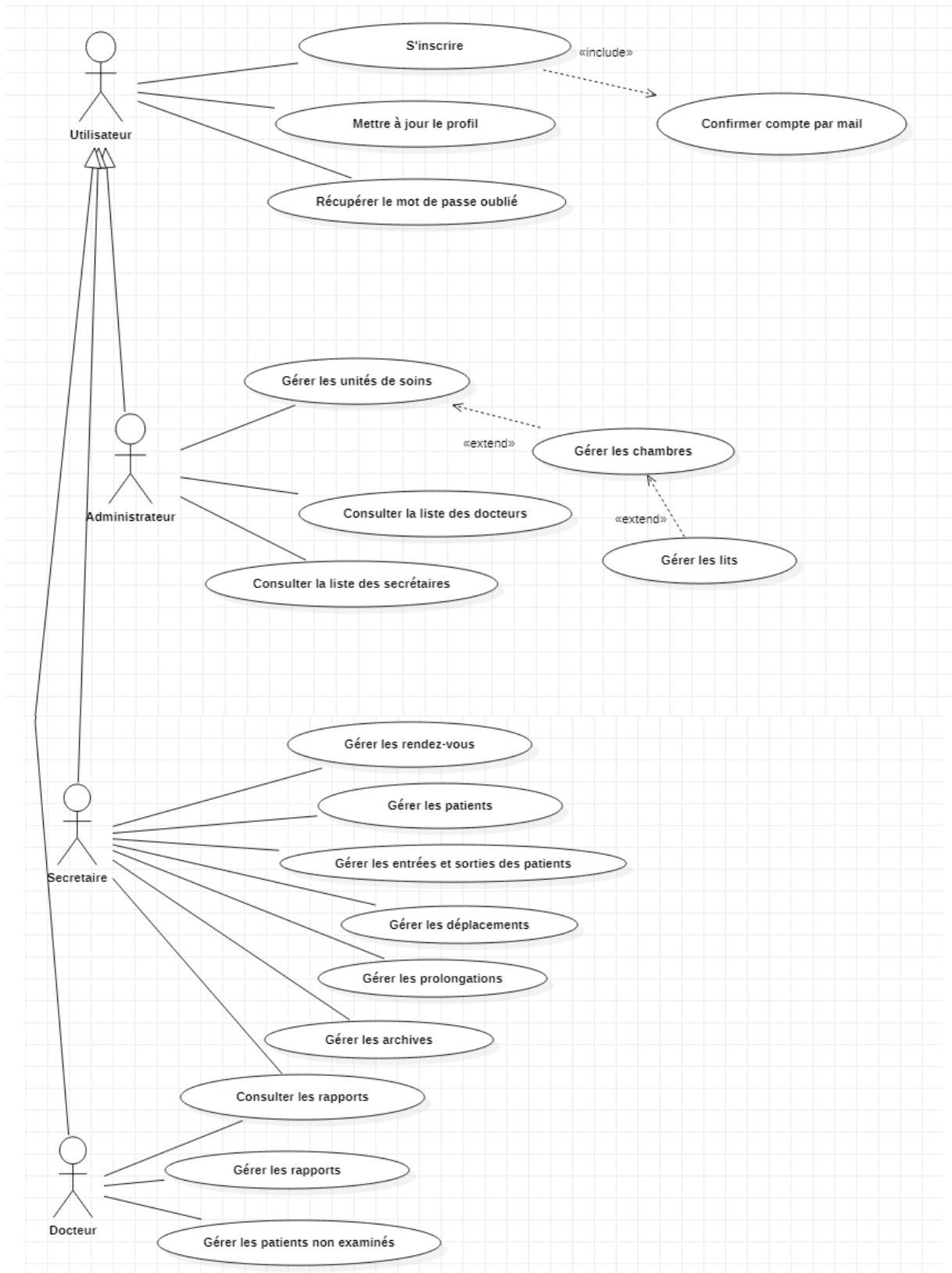


Figure 4 : Diagramme de cas d'utilisation global

5. Diagramme de classes globale

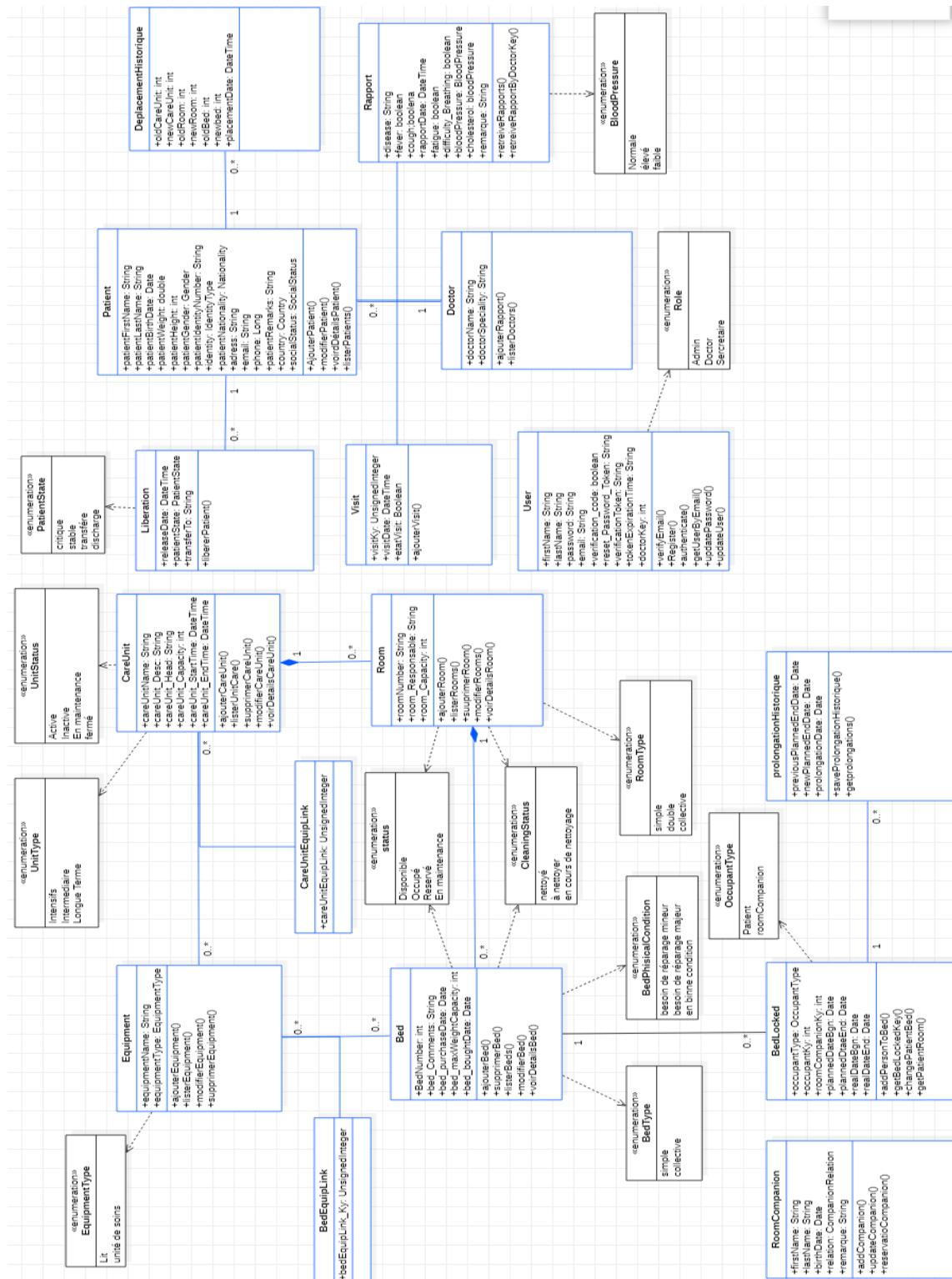


Figure 5: Diagramme de classes global

Conclusion

Dans ce chapitre, nous avons présenté les éléments du backlog du produit et nous avons défini la liste des besoins fonctionnels et non fonctionnels et les acteurs et de ce projet. De plus,

la présentation du diagramme de cas d'utilisation et classes global a permis d'illustrer de manière concrète les éléments qui seront implémentés dans le système.

Ce chapitre revêt une importance cruciale dans la réalisation du projet en offrant une vision cohérente et claire de ses principales composantes. Dans ce qui suit, nous détaillerons les étapes de chaque sprint.

Chapitre 3 : Gestion des comptes

Introduction

Dans ce chapitre, nous abordons le premier sprint qui se concentre sur le module de gestion des comptes. Nous commencerons par décrire les fonctionnalités de ce sprint à l'aide de diagrammes de cas d'utilisation, puis nous examinerons son comportement à travers des diagrammes de séquence. Enfin, nous présenterons le diagramme de classes, suivi de la mise en place des interfaces de réalisation.

Ce sprint constitue une étape cruciale dans le développement de notre application, car il établit les bases solides pour la gestion des comptes des utilisateurs.

1. Backlog du sprint 1

Le premier sprint se concentre sur les fonctionnalités clés de démarrage, notamment l'authentification des utilisateurs, l'inscription de compte, la récupération de mot de passe oublié et la confirmation par e-mail. Ces fonctionnalités sont essentielles pour garantir un processus fluide et sécurisé dès le début de l'utilisation de l'application.

1.1 Affectation des cas d'utilisations pour les acteurs

Acteurs	Cas d'utilisation
Utilisateur (admin, secrétaire, docteur)	Récupérer le mot de passe oublié
	Mettre à jour son profil
	Confirmer compte par mail
Secrétaire, Docteur	S'inscrire

Tableau 3: Classification des cas d'utilisations par acteur sprint 1

1.2 Sprint backlog

Le sprint backlog est un outil qui rend visible tout le travail en déterminant les tâches de chaque user story dans le backlog Product. Notre sprint backlog est décrit comme suit :

Id	User stories	Id	Tâches
1	En tant qu'utilisateur, je peux m'authentifier	1.1	-Etablir le prototype, le diagramme de cas d'utilisation et le diagramme de séquence « S'authentifier »

2	En tant qu'utilisateur, je peux modifier mon profil	2.1	-Etablir le prototype, le diagramme de cas d'utilisation et le diagramme de séquence « Mettre à jour profil »
		2.2	-Développer « Mettre à jour profil »
		2.3	-Tester
3	En tant qu'utilisateur, je peux récupérer le mot de passe oublié	3.1	-Etablir le prototype, le diagramme de cas d'utilisation et le diagramme de séquence « Récupérer mot de passe oublié »
		3.2	-Développer « Récupérer mot de passe oublié »
		3.3	-Tester
4	En tant qu'un utilisateur, je peux créer compte	4.1	-Etablir le prototype, le développement de mécanisme de l'inscription ainsi l'interface.
		4.2	-Développer « S'inscrire »
		4.3	-Tester
5	En tant qu'un utilisateur, je peux valider mon compte par mail	5.1	-Etablir le prototype, le diagramme de cas d'utilisation et le diagramme de séquence « Confirmer compte par mail »
		5.2	-Développer « Confirmer compte par mail »
		5.3	-Tester

Tableau 4 : Sprint 1 Backlog

2. Spécification fonctionnelle

Pour donner une vision globale du comportement fonctionnel du système, on utilise les diagrammes de cas d'utilisation. Dans cette partie, nous commençons par présenter le diagramme de cas d'utilisation globale correspondant au sprint 1, par la suite, nous les raffiner.

2.1 Diagramme de cas d'utilisation global de sprint 1

La figure ci-dessous représente les cas d'utilisations qui sont notre centre d'intérêt tout au long de ce sprint.

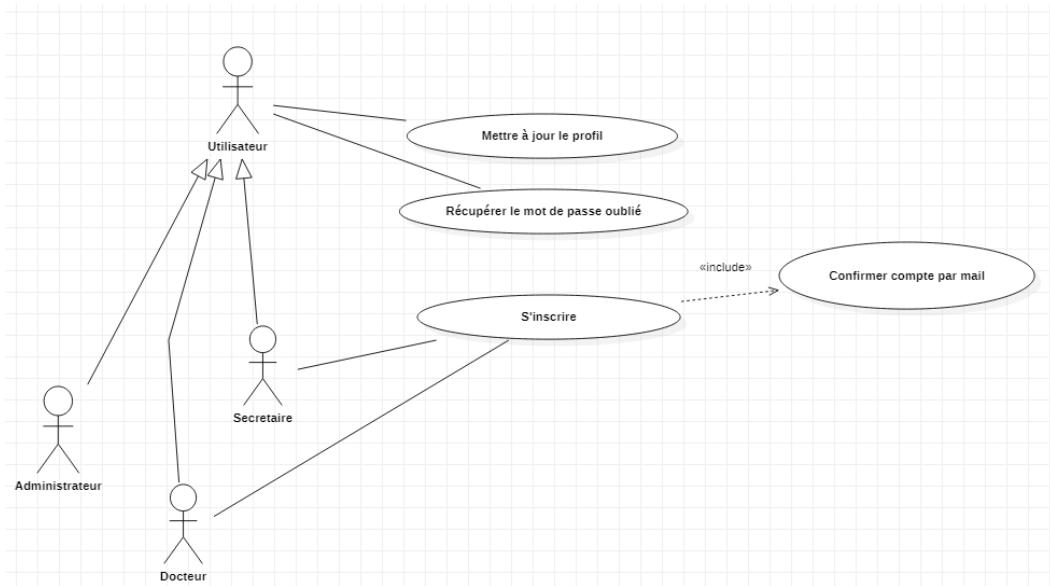


Figure 6: Diagramme d'utilisation globale de sprint 1

2.2 Description textuelle des cas d'utilisation

Dans cette partie nous décrivons textuellement les principaux cas d'utilisation qui apparaissent dans le diagramme de cas utilisation global de sprint 1.

-Description textuelle de cas d'utilisation « s'inscrire »

Cas d'utilisation	S'inscrire
Acteurs	Utilisateur (Secrétaire, Docteur)
Précondition	Le login n'existe pas déjà dans le système.
Postcondition	L'utilisateur est inscrit avec succès et peut accéder à son compte.
Scénario nominal	<ol style="list-style-type: none"> 1) L'application génère le formulaire de création de compte pour chaque utilisateur. 2) L'utilisateur saisit les informations nécessaires et les enregistre. <ul style="list-style-type: none"> ➤ Si l'utilisateur est un docteur il saisit sa spécialité. 3) L'utilisateur reçoit un e-mail de confirmation pour valider son compte. 4) L'utilisateur clique sur le lien de confirmation dans l'e-mail et son compte est confirmé. 5) L'utilisateur peut maintenant accéder à son compte et commencer à utiliser les fonctionnalités de l'application.
Scénario alternatif	<ol style="list-style-type: none"> 1) L'utilisateur introduit des données incorrectes. 2) Si le compte existe déjà le système affiche un message d'erreur.

Tableau 5 : Description textuelle du cas d'utilisation "S'inscrire"

-Description textuelle de cas d'utilisation « S'authentifier »

Cas d'utilisation	S'authentifier
Acteurs	Utilisateur (Secrétaire, Docteur, Administrateur)
Précondition	L'utilisateur doit posséder un compte et connaître son login et mot de passe
Postcondition	L'utilisateur est authentifié avec succès et accède à l'interface tableau de bord appropriée en fonction de son rôle (Administrateur, Secrétaire ou Docteur).
Scénario nominal	<ol style="list-style-type: none"> 1) Le système affiche le formulaire d'authentification. 2) L'utilisateur introduit son login et le mot de passe. 3) Le système vérifie les données introduites et affiche l'interface de tableau de bord correspondante au rôle de l'utilisateur.
Scénario alternatif	<ol style="list-style-type: none"> 1) Si l'utilisateur introduit des données incorrectes (login ou mot de passe incorrect), le système affiche un message d'erreur. 2) L'utilisateur peut réessayer de saisir les informations correctes ou demander une réinitialisation de son mot de passe. 3) Si compte de l'utilisateur est non vérifier, le système affiche un message d'erreur.

Tableau 6: Description textuelle du cas d'utilisation "S'authentifier"

-Description textuelle de cas d'utilisation « Récupérer mot de passe oublié »

Cas d'utilisation	Récupérer mot de passe oublié
Acteurs	Utilisateur (Secrétaire, Docteur, Administrateur)
Précondition	L'utilisateur a oublié son mot de passe et souhaite le récupérer.
Postcondition	L'utilisateur reçoit un e-mail avec un lien de réinitialisation de mot de passe.
Scénario nominal	<ol style="list-style-type: none"> 1) L'utilisateur sélectionne l'option "Mot de passe oublié" sur l'écran d'authentification. 2) Le système affiche un formulaire où l'utilisateur doit saisir son adresse e-mail associée à son compte. 3) L'utilisateur saisit son adresse e-mail et soumet le formulaire. 4) Le système vérifie que l'adresse e-mail saisie existe dans la base de données. 5) Si l'adresse e-mail est valide, le système génère un lien de réinitialisation de mot de passe unique et l'envoie à l'adresse e-mail de l'utilisateur. 6) L'utilisateur reçoit l'e-mail de réinitialisation de mot de passe

	<p>avec le lien.</p> <p>7) L'utilisateur clique sur le lien de réinitialisation dans l'e-mail.</p> <p>8) Le système affiche un formulaire où l'utilisateur peut saisir un nouveau mot de passe.</p> <p>9) L'utilisateur saisit le nouveau mot de passe et le confirme.</p> <p>10) Le système vérifie la validité du nouveau mot de passe et le met à jour dans la base de données.</p> <p>11) L'utilisateur est redirigé vers l'écran d'authentification où il peut se connecter avec son nouveau mot de passe.</p>
Scénario alternatif	<p>1) Si l'adresse e-mail saisie n'est pas valide (inexistante dans la base de données), le système affiche un message d'erreur demandant à l'utilisateur de vérifier l'adresse e-mail saisie.</p>

Tableau 7: Description textuelle du cas d'utilisation "Récupérer mot de passe oublié"

-Description textuelle de cas d'utilisation « Confirmer compte par mail »

Cas d'utilisation	Confirmer compte par mail
Acteurs	Utilisateur (Secrétaire, Docteur)
Précondition	L'apprenant a créé un compte et a reçu un e-mail de confirmation.
Postcondition	Le compte de l'apprenant est confirmé et il peut accéder aux fonctionnalités de l'application.
Scénario nominal	<p>1) L'utilisateur reçoit un e-mail de confirmation après avoir créé son compte.</p> <p>2) L'utilisateur ouvre l'e-mail de confirmation.</p> <p>3) L'e-mail de confirmation contient un lien de confirmation.</p> <p>4) L'utilisateur clique sur le lien de confirmation.</p> <p>5) Le système vérifie la validité du lien de confirmation.</p> <p>6) Si le lien de confirmation est valide, le compte de l'utilisateur est confirmé.</p> <p>7) Le système affiche un message de confirmation à l'utilisateur.</p> <p>8) L'utilisateur peut maintenant accéder aux fonctionnalités de l'application.</p>
Scénario alternatif	<p>1) L'utilisateur ne reçoit pas d'e-mail de confirmation après avoir créé son compte.</p> <p>2) L'utilisateur vérifie le dossier des courriers indésirables ou spam de sa boîte e-mail.</p>

Tableau 8: Description textuelle du cas d'utilisation "Confirmer compte par mail"

-Description textuelle de cas d'utilisation « Mettre à jour profil »

Cas d'utilisation	Mettre à jour profil
Acteurs	Utilisateur (Secrétaire, Docteur, Administrateur)
Précondition	L'utilisateur doit être authentifié dans le système.
Postcondition	Les informations du profil de l'utilisateur sont mises à jour.
Scénario nominal	<ol style="list-style-type: none"> 1) L'utilisateur accède à son profil dans l'application. 2) L'utilisateur modifie les informations qu'il souhaite mettre à jour 3) L'utilisateur enregistre les modifications. 4) Le système met à jour les informations du profil de l'utilisateur.
Scénario alternatif	<ol style="list-style-type: none"> 1) Le système ne met pas à jour les informations du profil de l'utilisateur.

Tableau 9: Description textuelle du cas d'utilisation "Mettre à jour profil"

2.3 Description comportementale des histoires du sprint 1

Dans le cadre de ce premier sprint, nous avons utilisé des diagrammes de séquence pour illustrer le comportement du système. Ces diagrammes permettent de représenter les interactions entre les objets et les acteurs du système au fil du temps, en se concentrant sur un cas d'utilisation spécifique, également appelé une histoire d'acteur. Dans la section suivante, nous présenterons les diagrammes de séquence correspondant aux cas d'utilisation clés de notre sprint 1.



Dans notre contexte, le terme "Boundary" fait référence à une interface graphique ou à une autre interface Homme/Machine, telle qu'un écran tactile ou un PocketPC.

La classe "Entity" quant à elle est utilisée pour modéliser des informations durables et souvent persistantes.

Enfin, la classe "Control" est utilisée pour représenter la coordination, le déroulement et le contrôle d'autres objets.

- **Diagramme de séquence « S'authentifier »**

La figure 7 illustre l'enchaînement des séquences nécessaire à la réalisation du cas d'utilisation s'authentifier pour l'utilisateur.

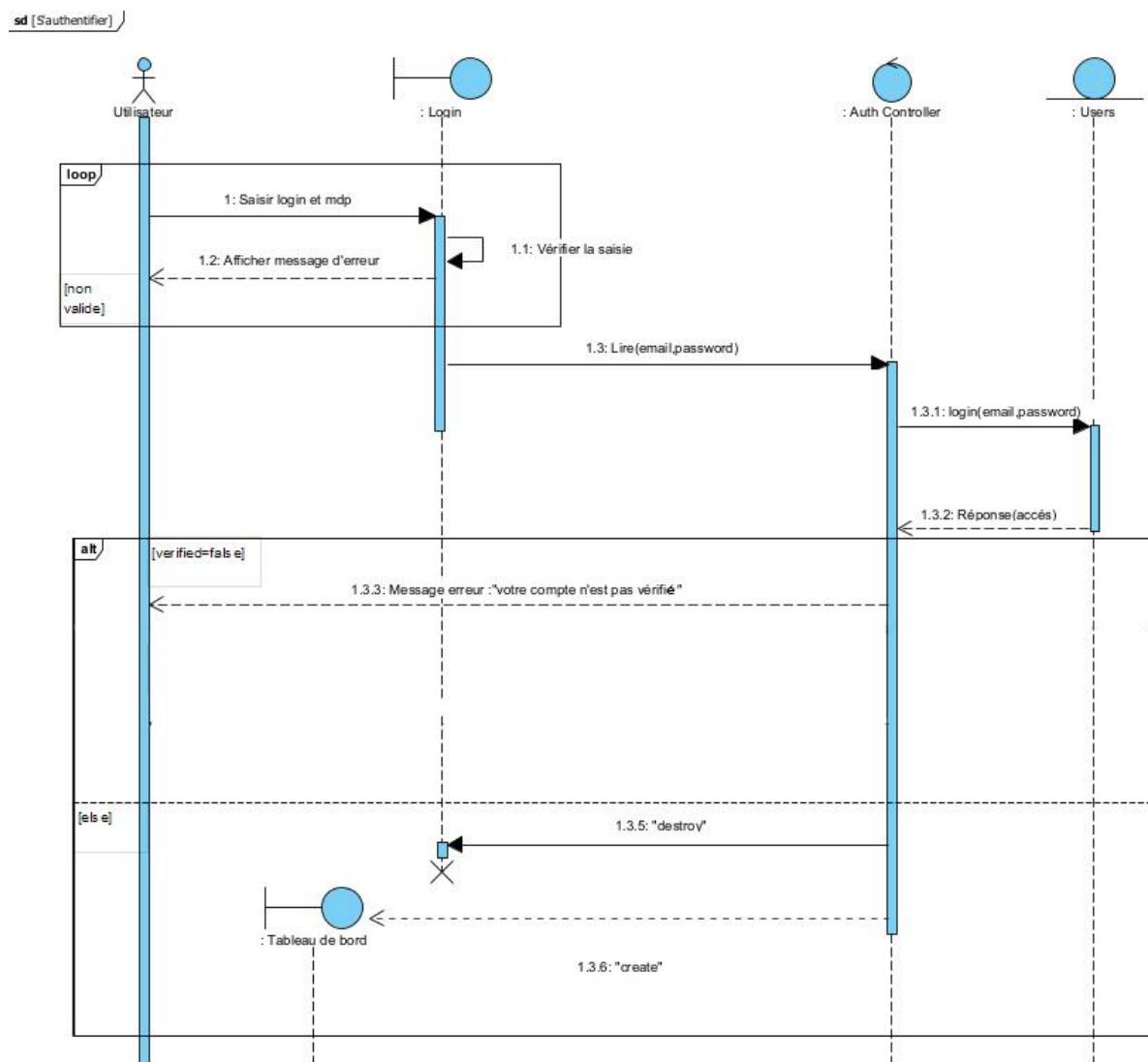


Figure 7: Diagramme de séquence de cas d'utilisation "S'authentifier"

- **Diagramme de séquence « Mettre à jour profil »**

La figure 8 illustre l'enchaînement des séquences nécessaire à la réalisation du cas d'utilisation s'authentifier pour l'utilisateur.

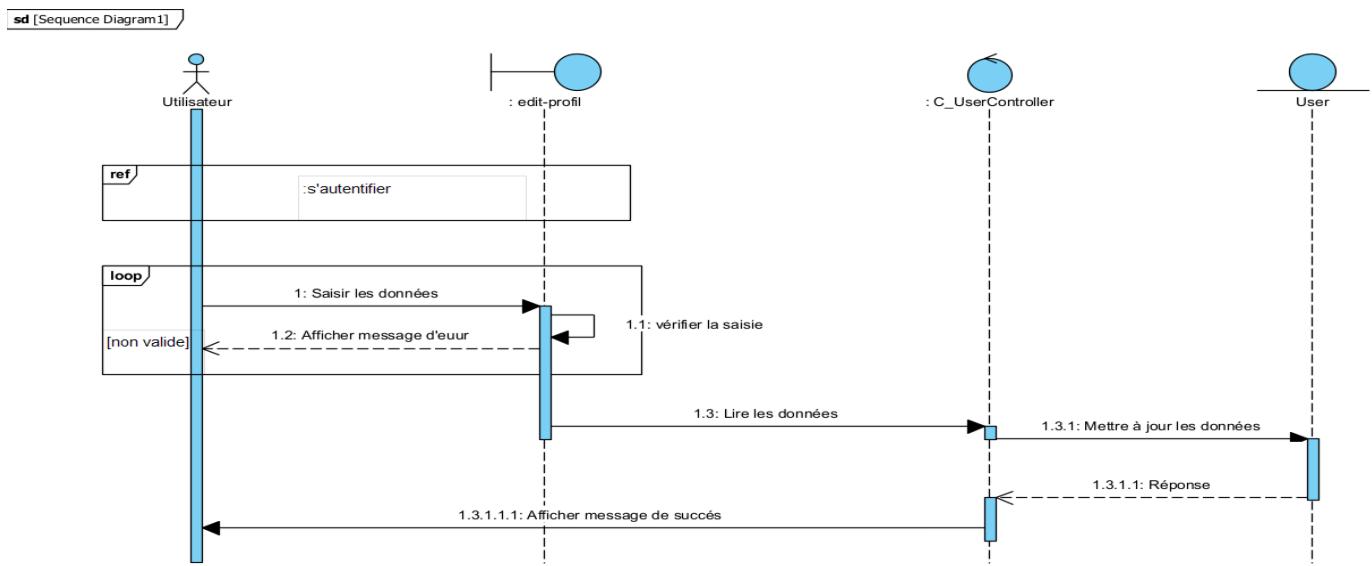


Figure 8 : Diagramme de séquence de cas d'utilisation "Mettre à jour profile"

- **Diagramme de séquence « Récupérer mot de passe oublié »**

La figure 9 illustre l'enchaînement des séquences nécessaire à la réalisation du cas d'utilisation s'authentifier pour l'utilisateur.

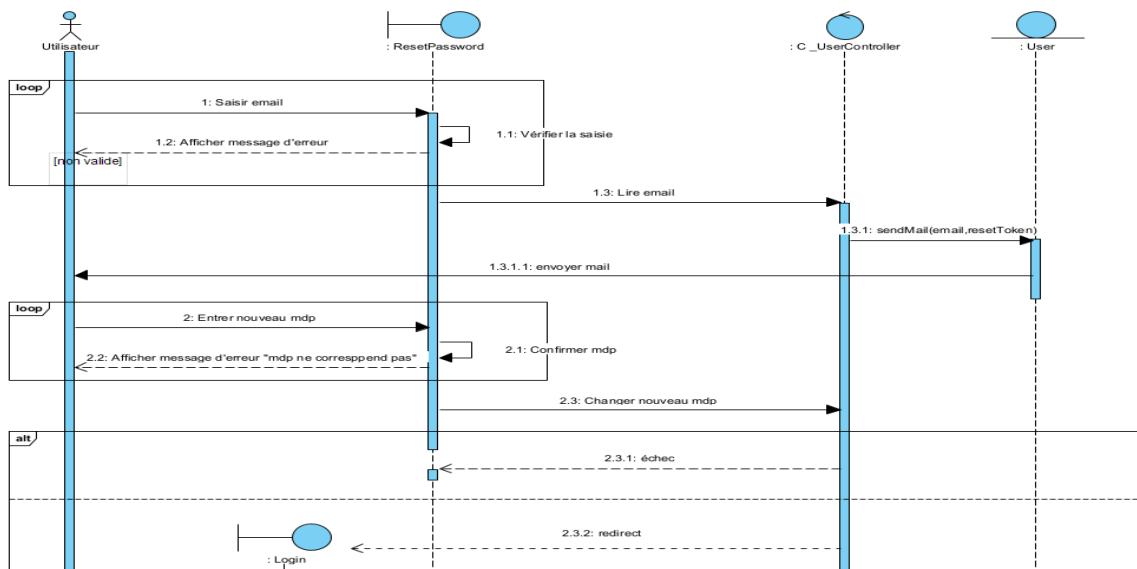


Figure 9: Diagramme de séquence de cas d'utilisation "Récupérer mot de passe oublié"

2.4 Diagramme de classe de sprint 1

Pour le premier sprint de notre projet, nous avons établi le diagramme de classes illustré par la figure

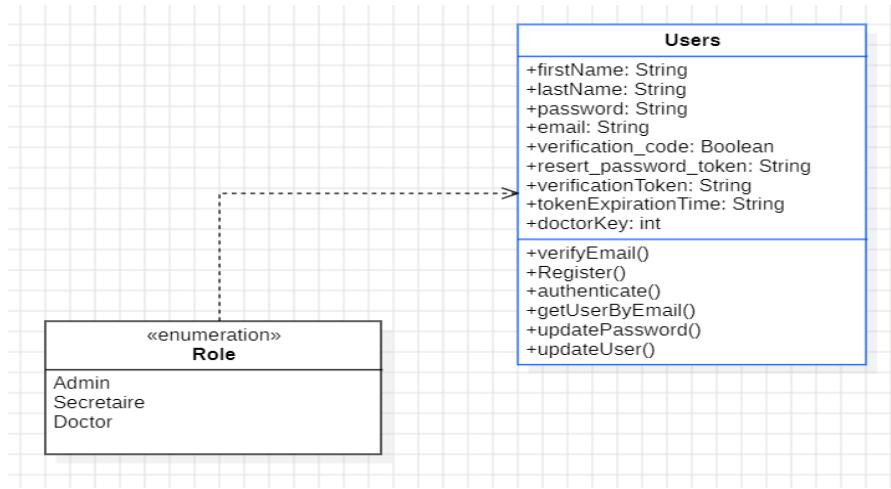


Figure 10 : Diagramme de classe de Sprint 1

3. Test et interfaçage

3.1 Test sprint 1

Cas	Résultat
Récupérer le mot de passe oublié	
Mettre à jour son profil	
Confirmer compte par mail	Conforme
S'inscrire	

Tableau 10 : Test du sprint1

3.2 Interfaçages

Pour illustrer la réalisation du premier sprint, nous présenterons les interfaces des principales fonctionnalités offertes par le module gestion de compte.

- **Interface « Authentification »**



Welcome to HealthTrack+

Nous rejoindre ? [Créer un compte](#)

Se connecter

Email*
safaa.ltifi123@gmail.com

Mot de passe*
...

[Mot de passe oublié?](#)

Connexion

Figure 11:Interface « Authentification »

- **Interface de l'« Inscription »**



S'inscrire

Fournissez vos informations pour créer votre compte.

Nom* prénom*

Email*

Mot de passe*

Confirmer mot de passe*

Role*

Déjà enregistré? [Login](#)

S'inscrire

Figure 12:Interface « Inscription »

- **Interface « Oublié mot de passe »**

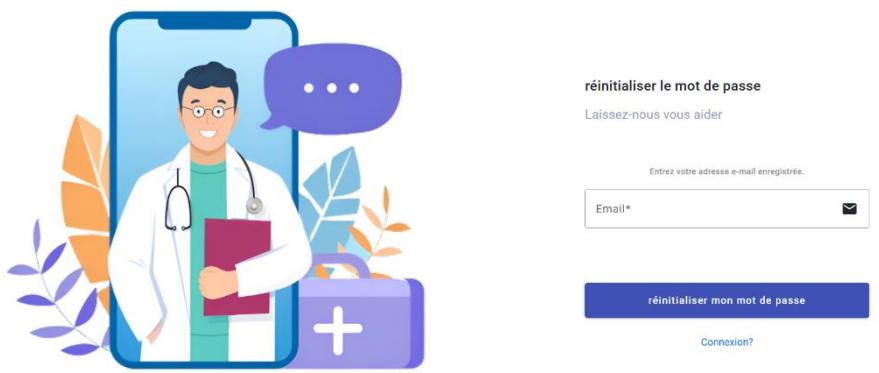


Figure 13:Interface « Oublié mot de passe »



Figure 14:Interface « Oublié mot de passe via mail »

- **Interface « Tableau de bord Administrateur »**

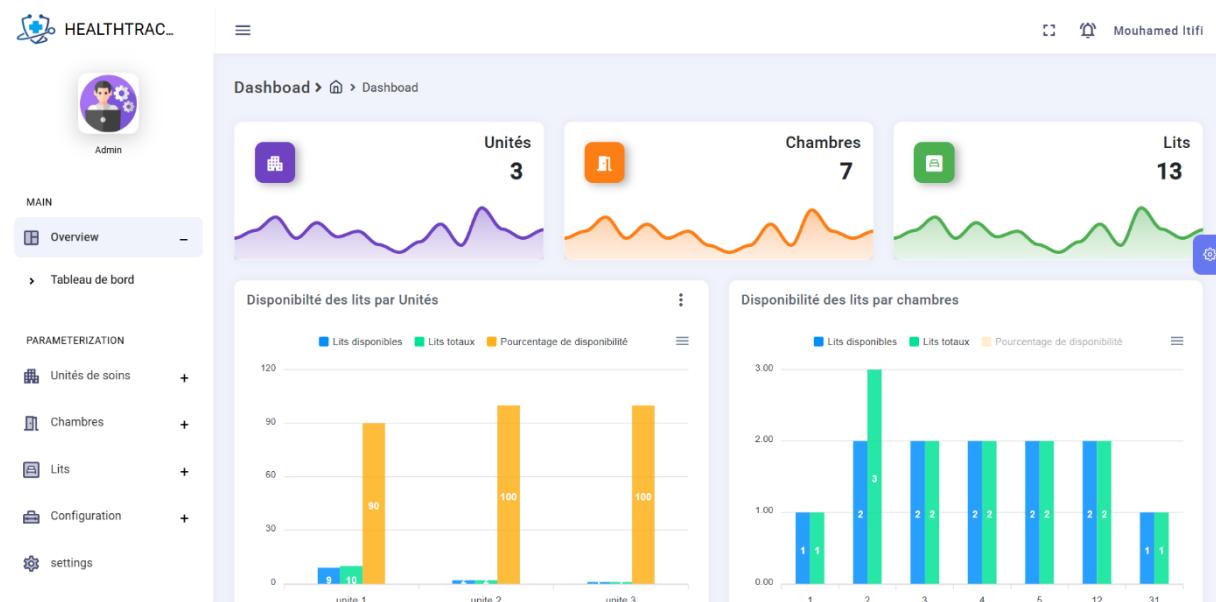


Figure 15:Interface « Tableau de bord Administrateur 1 »

The screenshot shows the 'Tableau de bord Administrateur 2' interface. On the left, a sidebar with 'MAIN' and 'PARAMETERISATION' sections. Under 'MAIN', 'Overview' is selected. Under 'PARAMETERISATION', 'Unités de soins', 'Chambres', 'Lits', 'Configuration', and 'settings' are listed. The main content area has three sections: 'Unités de soins' (listing units 22 to 51), 'Les lits' (listing beds 11 to 32 with status like 'Nettoye' or 'A_Nettoyer'), and 'Les docteurs' (listing doctor souad trabelsi with email souad123@gmail.com and specialization 'chirugie').

Figure 16: Interface « Tableau de bord Administrateur 2 »

- **Interface « Tableau de bord Doctor »**

The screenshot shows the 'Tableau de bord Doctor' interface. On the left, a sidebar with 'MAIN' and 'PARAMETERISATION' sections. Under 'MAIN', 'Tableau de bord', 'Liste Patients', and 'Settings' are listed. The main content area includes a welcome message for Dr. souad trabelsi, a summary of 'Tous Patients' (3), 'Today Patients' (1), and 'Patients Examinés' (1), and a line graph titled 'Doctor Performance' showing patient count from Monday to Friday.

Figure 17: Interface « Tableau de bord Doctor » 1

The screenshot shows the 'Tableau de bord Doctor' interface. On the left, a sidebar with 'MAIN' and 'PARAMETERISATION' sections. Under 'MAIN', 'Tableau de bord', 'Liste Patients', 'Settings', and 'Déconnexion' are listed. The main content area includes a 'Doctor Performance' graph and a table for 'Today Appointment' (Nom: othmeni, Prénom: sawsen, Email: sawsen@gmail.com, Numéro: 124563, Date visite: 2024-05-31 10:30, Etat visite: checked), 'Past Appointment', and 'Upcoming Appointment'.

Figure 18: Interface « Tableau de bord Doctor » 2

- **Interface « Tableau de bord Secrétaire »**

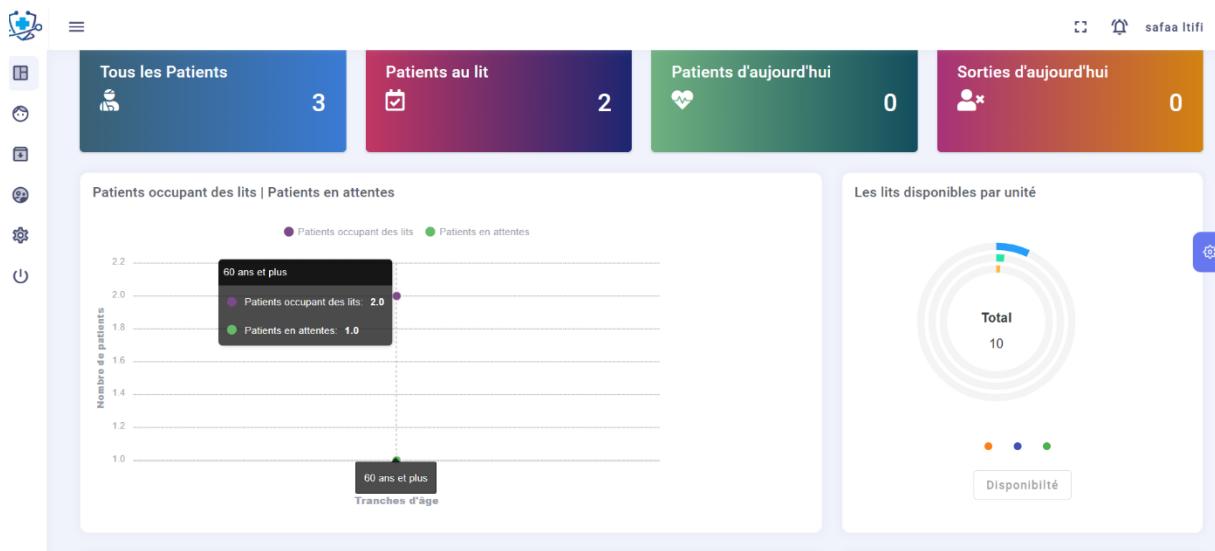


Figure 19:Interface « Tableau de bord Secrétaire » 1

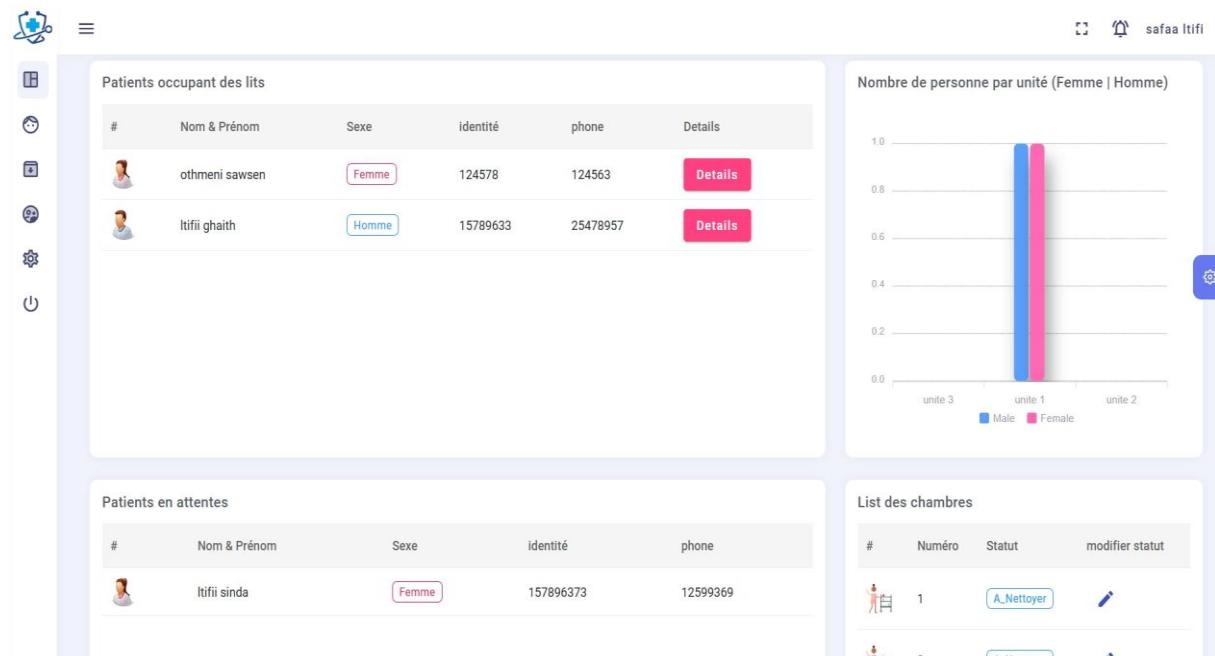


Figure 20:Interface « Tableau de bord 2 Secrétaire »

- **Interface « Mettre à jour profil »**

The screenshot displays a user interface for updating a profile. At the top, a navigation bar shows 'Settings > ⌂ > Settings'. The main content is divided into two sections: 'Paramètres de sécurité' (Security settings) and 'Paramètres de compte' (Account settings).

Paramètres de sécurité

- Username: souad123@gmail.com
- Current Password*: (input field with a gear icon for password strength)
- New Password*: (input field)

Paramètres de compte

- First Name: souad
- Last Name: trabelsi
- Email: souad123@gmail.com

Buttons:

- Save (blue button)
- Save Changes (blue button at the bottom)

Figure 21:Interface « Mettre à jour profil »

Conclusion

Dans ce chapitre, nous avons planifié et exécuté le premier sprint de notre projet. Nous avons identifié les différentes fonctionnalités à réaliser et les avons décrites en détail à travers les descriptions fonctionnelles et comportementales des histoires. Nous avons également élaboré le diagramme de classes correspondant au sprint et avons procédé à la réalisation des interfaces nécessaires. Nous sommes maintenant prêts à passer au chapitre suivant où nous aborderons le deuxième sprint et poursuivrons le développement de notre application.

Chapitre 4 : Gestion des ressources matérielles

Introduction

Après avoir finalisé le sprint 1 concernant la gestion des comptes, nous avons tenu une réunion pour définir la répartition des tâches pour le sprint 2. Ce sprint, centré sur la "Génération de l'application web", vise principalement à prendre en charge la gestion des ressources matérielles.

1. Backlog du sprint 2

1.1 Affectation des cas d'utilisations pour les acteurs

Acteurs	Cas d'utilisation
Admin	Gérer les unités de soins
	Gérer les chambres
	Gérer les lits

Tableau 11:Classification des cas d'utilisations par acteur sprint 2

1.2 Sprint backlog

Débutons par l'identification de la liste des tâches à réaliser pendant ce Sprint. Le plan de cette itération inclut quatre modules principaux, comme définis dans le tableau

Id	User stories	Id-tâches	Tâches
1	En tant qu'administrateur, je souhaite consulter la liste des unités de soins, des chambres et des lits.	1.1 1.2 1.3	-Établir le prototype et créer le composant de la liste des unités de soins, les chambres et les lits. -Développer le composant de consultation de la liste des unités de soins, les chambres et les lits. -Tester
2	En tant qu'administrateur je souhaite voir les détails d'une unité de soins et compris ces équipements, les lits compris ces équipements et les	2.1	-Établir le prototype et créer le composant de visualisation des détails de l'unité, des chambres et des lits.

	chambres.	2.2 2.3	-Développer le composant de visualisation des détails de l'unité, des chambres et des lits. -Tester
3	En tant qu'un utilisateur, je souhaite modifier une unité de soins, un lit ou une chambre,	3.1 3.2 3.3	-Établir le prototype et créer un popup pour modifier une unité de soin, un lit et une chambre. -Développer la méthode de modifier une unité de soin, lit et chambre. -Tester
4	En tant qu'un utilisateur, je veux pouvoir affecter un filtrage avancé sur les unités de soins par nom, type et capacité en même temps.	4.1 4.2	-Mettre en place le mécanisme permettant à l'utilisateur de filtrer les unités de soins par nom, type et capacité. -Tester le filtrage pour vérifier qu'il fonctionne correctement et qu'il renvoie les résultats attendus.
5	En tant qu'un utilisateur, je veux pouvoir filtrer les unités, les lits et les chambres de soins par tous les types présentent.	5.1 5.2	- Mettre en place le mécanisme permettant à l'utilisateur de filtrer les unités de soins, les lits et les chambres par tous les types présentent. -Tester le filtrage pour vérifier qu'il fonctionne correctement et qu'il renvoie les résultats attendus.
6	En tant qu'administrateur je souhaite ajouter unité de soins avec des équipements, ajouter un lit avec ces équipements et une chambre.	6.1 6.2 6.3	-Établir le prototype et créer le composant ajouter une unité de soin, un lit et une chambre. -Développer le composant ajouter une unité de soin -Tester
7	En tant qu'administrateur je souhaite consulter la liste des docteurs.	7.1	-Établir le prototype et créer le composant de visualisation de la liste

		7.2	des docteurs. -Développer le composant de visualisation de la liste des docteurs.
		7.3	-Tester
8	En tant qu'administrateur je souhaite consulter la liste des secrétaires.	8.1	-Établir le prototype et créer le composant de visualisation de la liste des secrétaires.
		8.2	-Développer le composant de visualisation de la liste des secrétaires.
		8.3	-Tester

Tableau 12 : Backlog Sprint 2

2. Raffinement du Sprint

2.1 Identification des besoins fonctionnels

Les cas d'utilisation définissent des séquences d'actions qui détaillent de manière complète les besoins fonctionnels du système. Chaque cas d'utilisation représente une activité métier du système, observée du point de vue d'un utilisateur ou d'un acteur. Voici une liste des principales fonctionnalités que notre plateforme doit fournir :

- **Gestion des unités de soins** : Le cas d'utilisation "Gestion des unités de soins" permet aux utilisateurs de contrôler et administrer les informations d'une unité de soins spécifiques. Ils peuvent ajouter, consulter et mettre à jour les détails, ainsi que supprimer les unités obsolètes. Cela permet une gestion efficace des données et des ressources en soins de santé.
- **Gestion des chambres** : L'objectif du cas d'utilisation "Gestion des chambres" est de permettre aux utilisateurs de contrôler, ajouter, consulter, mettre à jour et supprimer les informations relatives aux chambres d'une unité de soins. Cela contribue à une gestion efficace des ressources et des services de santé.
- **Gestion des lits** : L'objectif du cas d'utilisation "Gérer les lits" est de permettre aux utilisateurs de contrôler et de manipuler les informations associées à un lit spécifique dans le contexte d'une chambre au sein d'une unité de soins.

2.2 Modèle global des cas d'utilisation du Sprint

2.2.1 Diagramme de cas d'utilisation du sprint

Le deuxième sprint se concentre sur le cas d'utilisation qui décrit les différentes fonctionnalités représentées dans le diagramme de cas d'utilisation illustré ci-dessous :

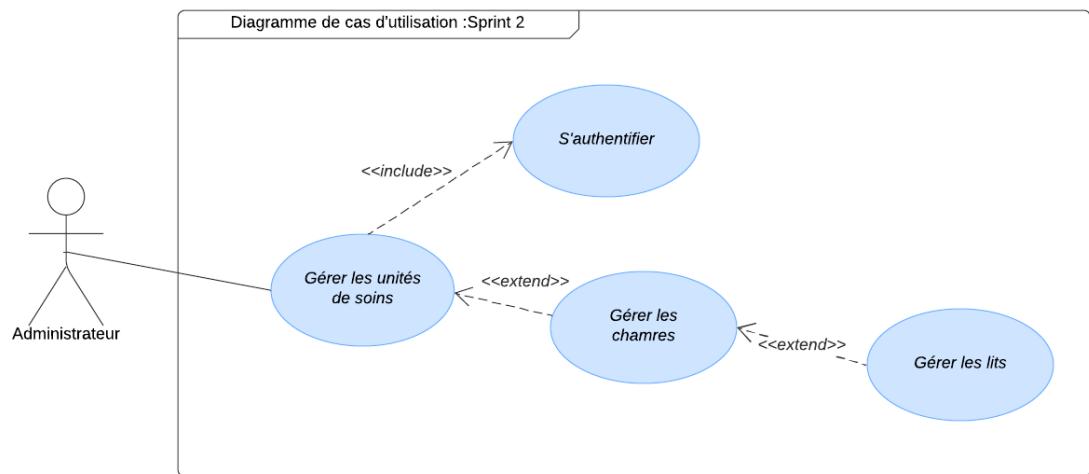


Figure 22: Diagramme de cas d'utilisation Sprint 2

2.2.2 Raffinement des cas d'utilisations

Dans cette section, nous allons détailler les principaux cas d'utilisation du premier Sprint, en commençant par celui de la gestion des unités de soins illustré ci-dessous :

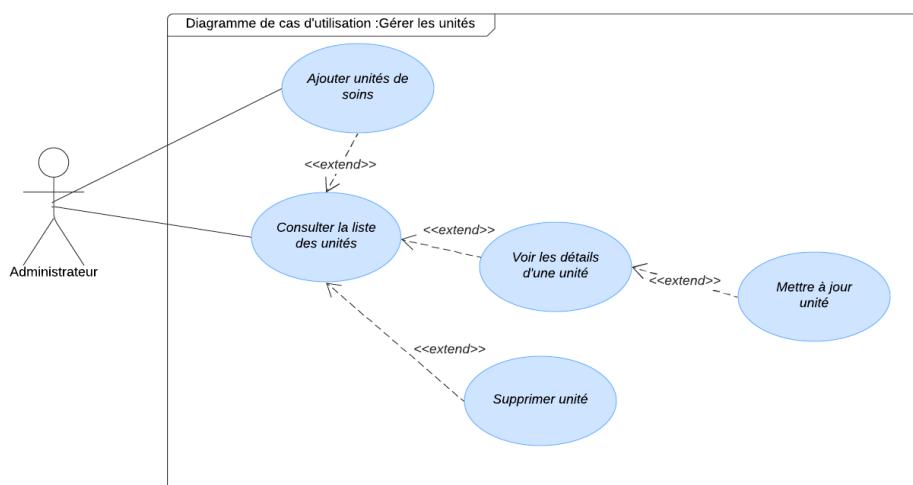


Figure 23: Diagramme de cas d'utilisation Gérer les unités

Le tableau 13 représente une description du cas d'utilisation « Gérer les unités de soins »

Cas d'utilisation	Gérer les unités de soins
Acteurs	Administrateur
Précondition	L'administrateur doit s'authentifier afin d'accéder au système
Postcondition	Les modifications apportées aux unités de soins sont enregistrées dans le système.
Scénario nominal	<ol style="list-style-type: none"> 1) L'administrateur accède à la fonctionnalité de gestion des unités de soins. 2) Le système affiche la liste des unités de soins existants. 3) L'administrateur peut choisir d'ajouter une nouvelle unité de soin en fournissant les informations nécessaires telles que le nom, type, statut, etc. 4) Le système enregistre la nouvelle unité dans la base de données. 5) L'administrateur peut sélectionner une unité de soin existant pour afficher les informations sur l'unité. 6) Lorsque le formulaire est affiché, en double-cliquant sur chaque attribut, l'administrateur peut effectuer des modifications. 7) Le système enregistre les modifications apportées à l'unité de soin. 8) L'administrateur a la possibilité de supprimer une unité de soin existante en cliquant sur le bouton supprimer et en confirmant la suppression. 9) Le système supprime l'unité de soin de la base de données.
Scénario alternatif	<ol style="list-style-type: none"> 1) Si aucune unité de soin n'est disponible, le système affiche un message indiquant aucune donnée disponible. 2) Si une erreur se produit lors de l'ajout, de la modification ou de la suppression d'une unité de soin, le système affiche un message d'erreur et l'administrateur est informé.

Tableau 13 : Description textuelle du cas d'utilisation

Le schéma 24 représente le diagramme de cas d'utilisation « Gérer les chambres ».

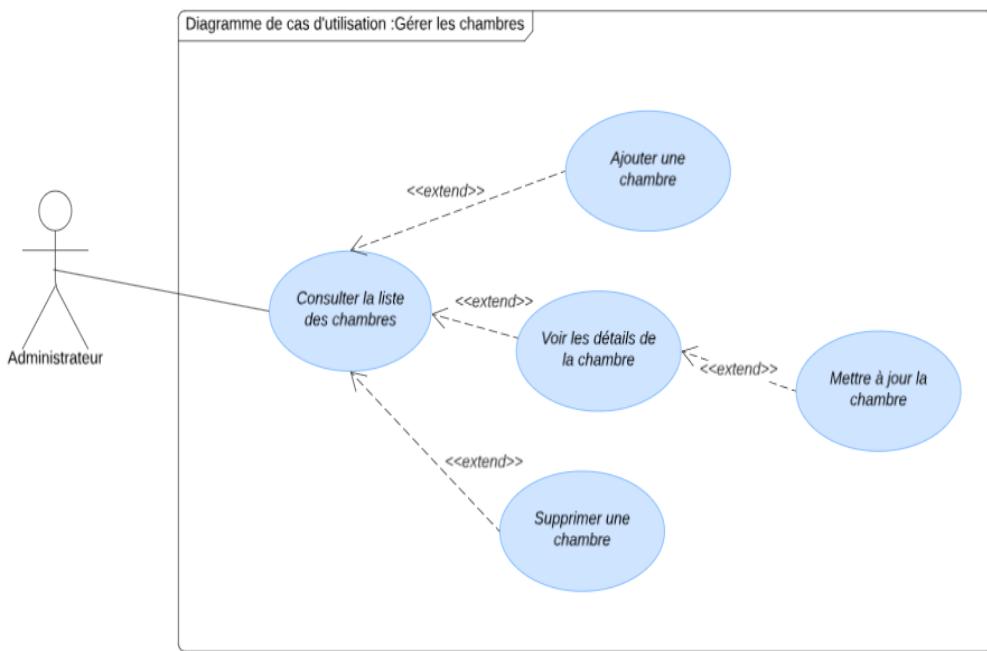


Figure 24 : Diagramme de cas d'utilisation Gérer les chambres

Le tableau 14 représente une description du cas d'utilisation Gérer les chambres :

Cas d'utilisation	Gérer les chambres
Acteurs	Administrateur
Précondition	L'administrateur doit s'authentifier afin d'accéder au système
Postcondition	Les modifications apportées aux chambres sont enregistrées dans le système.
Scénario nominal	<ol style="list-style-type: none"> 1) L'administrateur sélectionne l'option "Chambres" associé à une unité de soins. 2) Le système affiche une interface présentant la liste de toutes les chambres dans une unité de soins spécifique. 3) L'administrateur peut choisir d'ajouter une chambre en fournissant les informations nécessaires telles que le nom, capacité, statut, etc. 4) Le système enregistre la nouvelle chambre dans la base de données. 5) Chaque chambre présente deux options : "Supprimer" et "Voir détail". 6) En cliquant sur le bouton "Voir détail", l'administrateur accède à un formulaire affichant les détails de la chambre sélectionnée.

	<p>7) Lorsque le formulaire est affiché, en double-cliquant sur chaque attribut, l'administrateur peut effectuer des modifications</p> <p>8) Le système enregistre les modifications apportées à la chambre.</p> <p>9) L'administrateur a la possibilité de supprimer une chambre existante en cliquant sur le bouton "Supprimer" et en confirmant la suppression.</p> <p>10) Le système supprime la chambre de la base de données.</p>
Scénario alternatif	<p>1) Si aucune chambre n'est disponible, le système affiche un message indiquant aucune donnée disponible.</p> <p>2) Si une erreur se produit lors de l'ajout, de la modification ou de la suppression d'une chambre, le système affiche un message d'erreur et l'administrateur est informé.</p>

Tableau 14 : Description textuelle du cas d'utilisation Gérer les chambres

Maintenant on va présenter le diagramme de cas de gestion des lits :

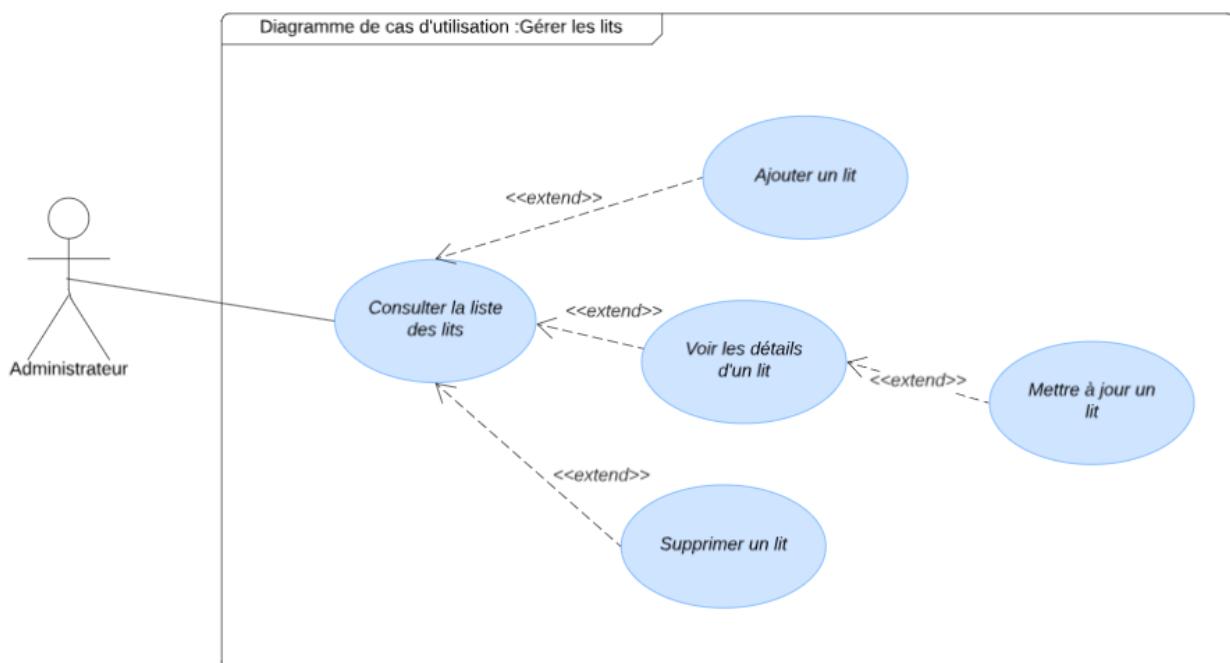


Figure 25 : Diagramme de cas d'utilisation Gérer les lits

Le tableau ci-dessous représente la description du cas d'utilisation « Gérer les lits »

Cas d'utilisation	Gérer les lits
Acteurs	Administrateur
Précondition	L'administrateur doit s'authentifier afin d'accéder au système
Postcondition	Les modifications apportées aux lits sont enregistrées dans le système.
Scénario nominal	<p>1) L'administrateur sélectionne l'option "Lits" associé à une chambre.</p> <p>2) Le système affiche une interface présentant la liste de tous les lits dans une chambre spécifique.</p> <p>3) L'administrateur peut choisir d'ajouter un lit en fournissant les informations nécessaires telles que le numéro, statut, statut de nettoyage, etc.</p> <p>4) Le système enregistre le nouveau lit dans la base de données.</p> <p>5) Chaque lit présente deux options : "Supprimer" et "Voir détail".</p> <p>6) En cliquant sur le bouton "Voir détail", l'administrateur accède à un formulaire affichant les détails de le lit sélectionnée.</p> <p>7) Lorsque le formulaire est affiché, en double-cliquant sur chaque attribut, l'administrateur peut effectuer des modifications.</p> <p>8) Le système enregistre les modifications apportées au lit.</p> <p>9) L'administrateur a la possibilité de supprimer un lit existant en cliquant sur le bouton "Supprimer" et en confirmant la suppression.</p> <p>10) Le système supprime le lit de la base de données.</p>
Scénario alternatif	<p>1) Si aucun lit n'est disponible, le système affiche un message indiquant aucune donnée disponible.</p> <p>2) Si une erreur se produit lors de l'ajout, de la modification ou de la suppression d'un lit, le système affiche un message d'erreur et l'administrateur est informé.</p>

Tableau 15: Description textuelle du cas d'utilisation Gérer les lits

2.3 Diagramme de séquences

Pour accéder à notre outil, l'administrateur doit d'abord se connecter en utilisant ses identifiants. Il accède à la page de connexion où il saisit son nom d'utilisateur et son mot de passe. Ensuite, le système vérifie ces informations spécifiques à l'administrateur. En cas de réussite de l'authentification, l'administrateur est redirigé vers la page d'accueil. Les différentes étapes de ce processus sont représentées dans le diagramme de séquence illustré dans la figure.

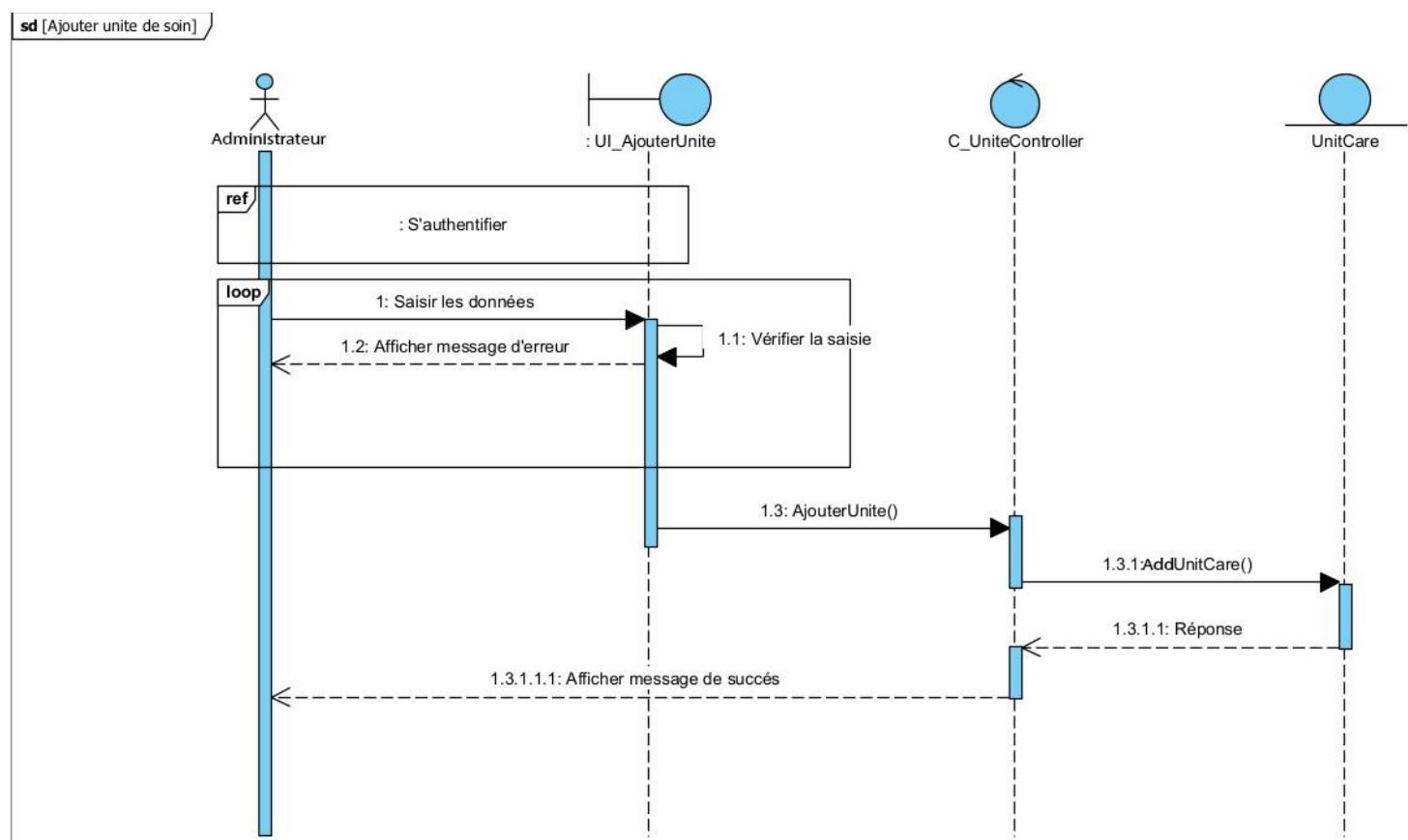


Figure 26: Diagramme de séquence "Ajouter unité de soins"

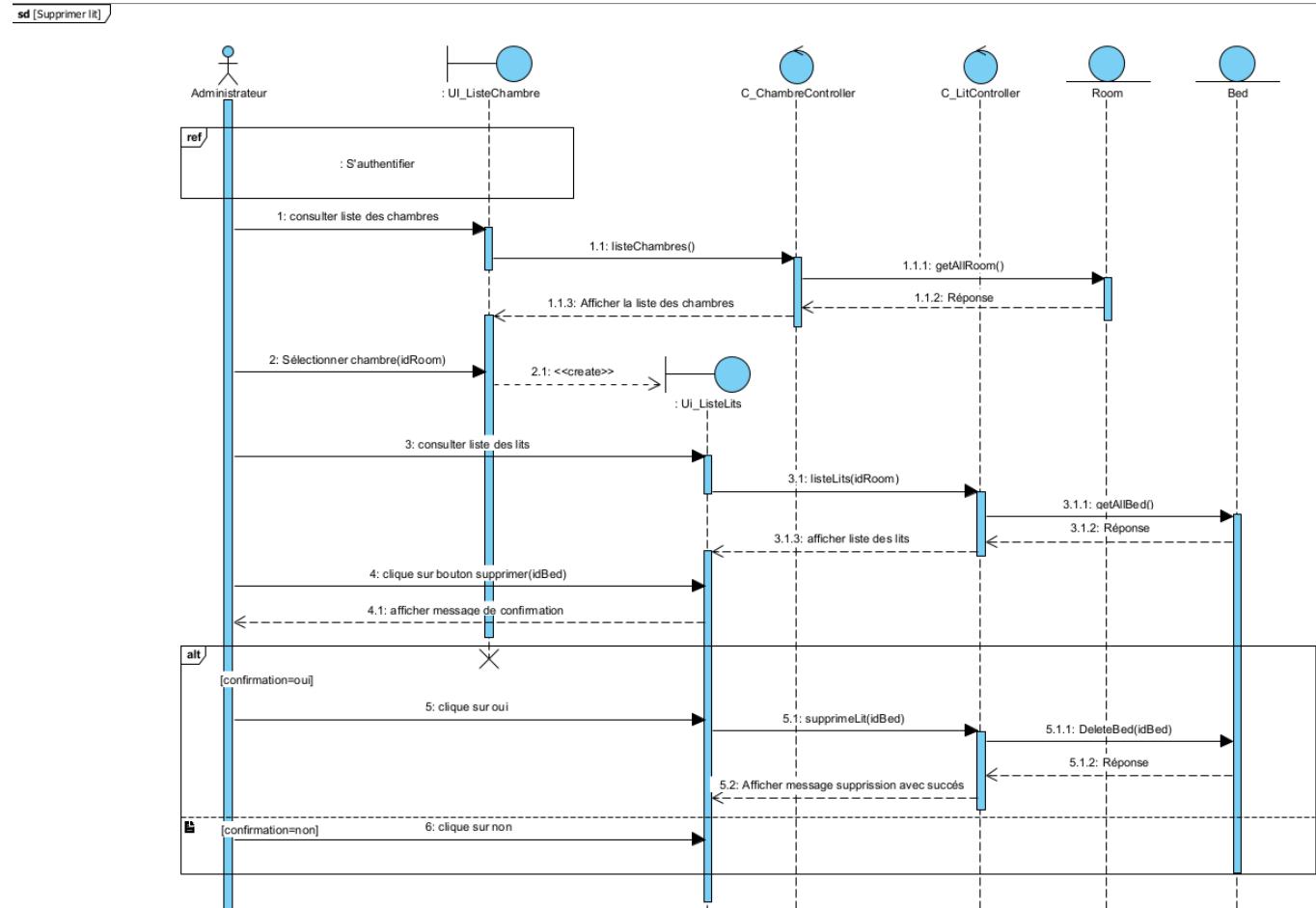


Figure 27: Diagramme de séquence " Supprimer un lit"

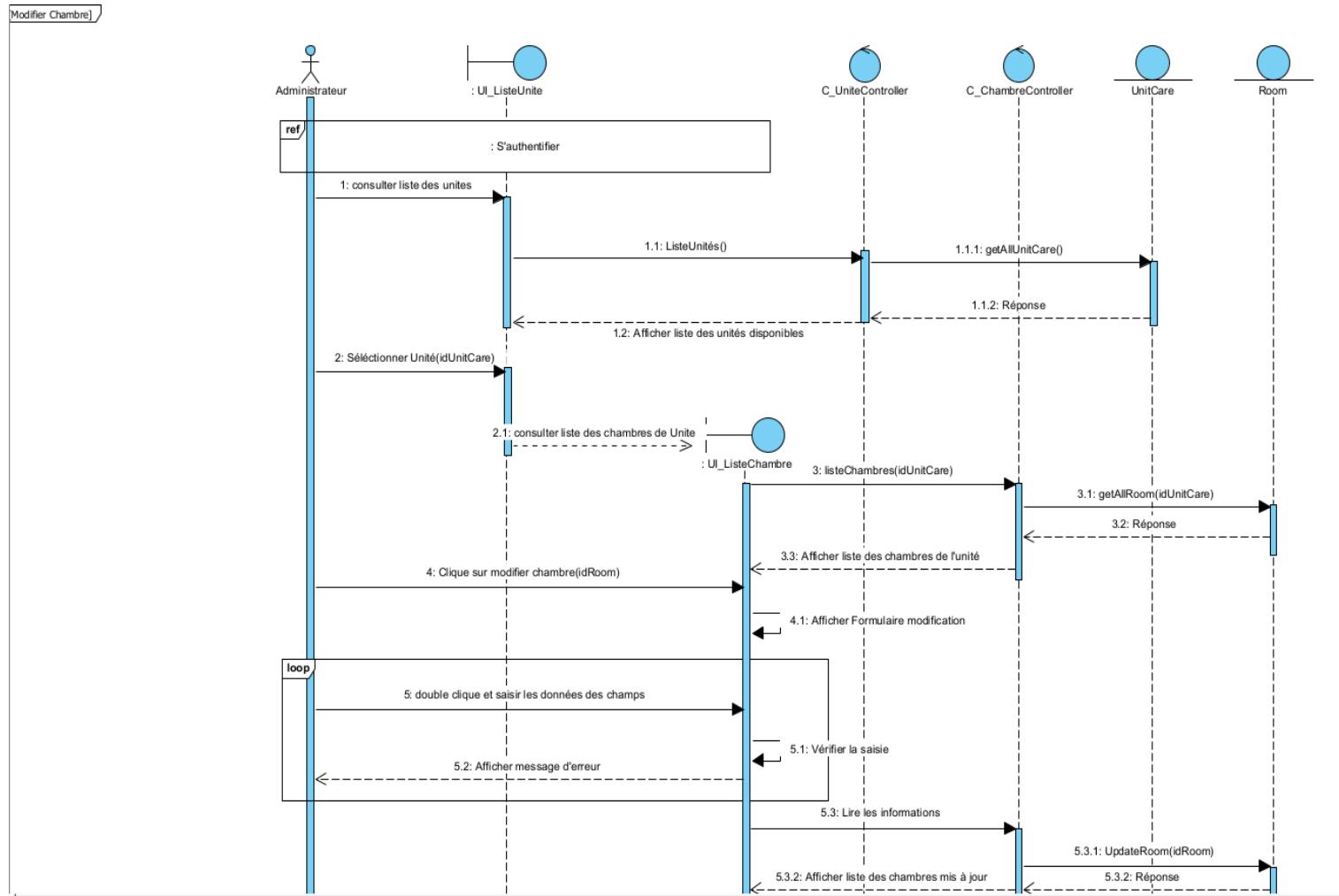


Figure 28: Diagramme de séquence "Modifier chambre"

2.4 Diagramme de classes Sprint 2

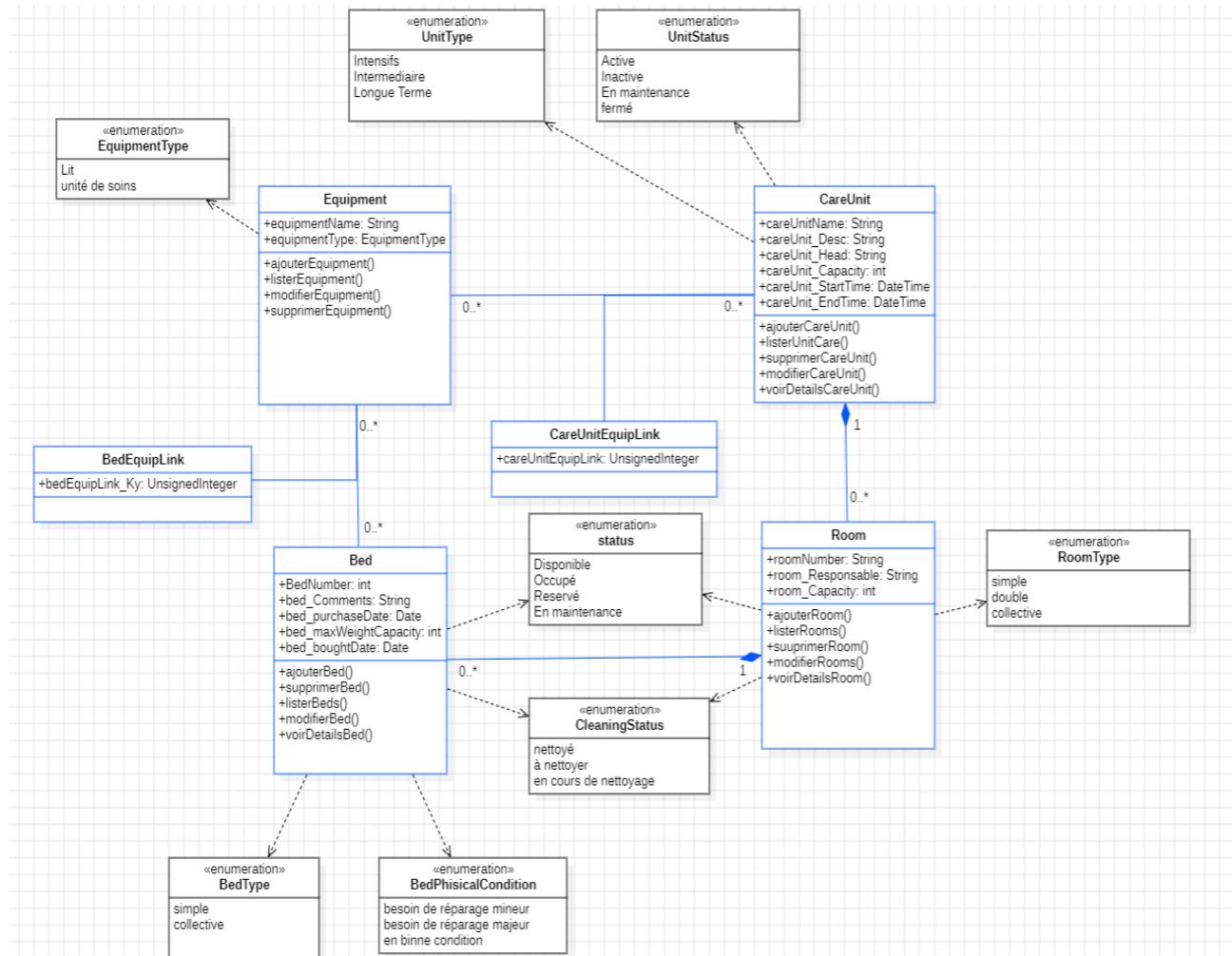


Figure 29 : Diagramme de classe Sprint 2

3. Test et interfaçage

3.1 Test sprint 2

Cas	Résultat
Gérer les unités de soins	
Gérer les chambres	Conforme
Gérer les lits	

Tableau 16:Test du sprint 2

3.2 Interfaçages

- Interface « Liste des unités des soins »

Figure 30:Interface « Liste des unités des soins »

- Interface de « Ajouter Unité de soins »

Figure 31:Interface de « Ajouter Unité de soins »

- **Interface « Filtrage avancé »**

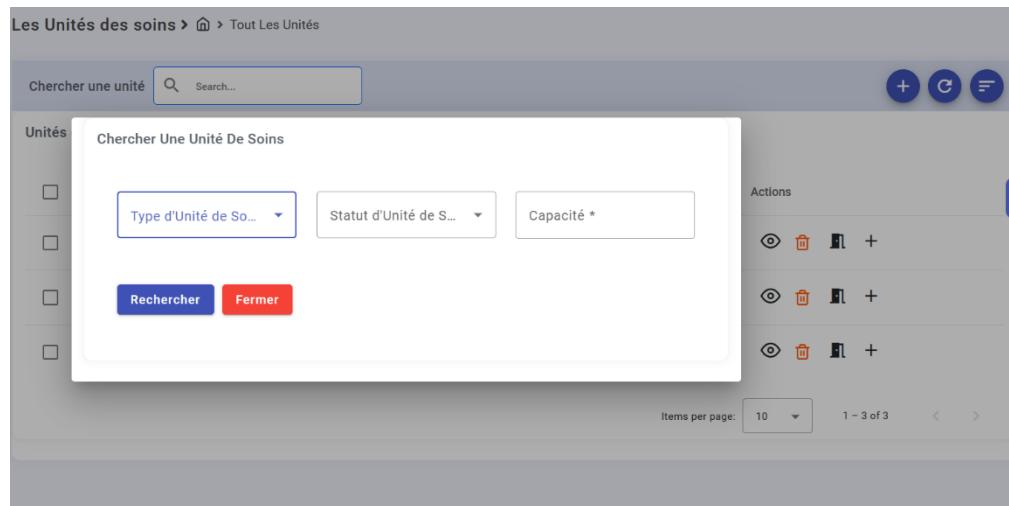


Figure 32:Interface « Filtrage avancé »

- **Interface « Liste des Chambres dans une unité »**

The screenshot shows a list of rooms in a unit. The table has columns: 'Num' (Number), 'Responsable' (Responsible), 'Type' (Type), 'Status' (Status), 'Capacité' (Capacity), and 'Statut de nettoyage' (Cleaning Status). Each row includes an 'Actions' column with icons for edit, delete, and more. The sidebar on the left shows a navigation menu with 'Overview', 'Unités de soins', 'Chambres', 'Lits', 'Configuration', 'settings', and 'Déconnexion'. The top right shows the user 'Mouhamed Itifi'.

Num	Responsable	Type	Status	Capacité	Statut de nettoyage	Actions
1	souad	Simple	Disponible	1	À nettoyer	
2	amal	Double	Disponible	2	À nettoyer	
3	ghaith	COLLECTIVE	Disponible	6	À nettoyer	
4	qlq	Double	Disponible	2	À nettoyer	
5	hela	Double	Disponible	2	À nettoyer	

Figure 33:Interface « Liste des unités des Chambres »

- **Interface « Ajouter une chambre dans une unité de soins »**

Les Unités des soins > Tout Les Unités

Ajouter Une Chambre dans l'unité : unite 1

Num* Nom Responsable*

Type de chambre* Statut de chambre*

Statut de Nettoyage chambre* Capacité *

Submit Fermer

Figure 34:Interface « Ajouter une chambre dans une unité de soins »

- **Interface « Liste des lits dans une chambre »**

Les Lits > Les Lits de la chambre : 2

MAIN

- Overview
- PARAMETERIZATION
- Unités de soins
- Chambres
- Lits
- Configuration
- settings
- Déconnexion

Les lits :

Numéro	Type	Statut Physique	Statut de Nettoyage	Actions
21	Simple	Bon état	À nettoyer	
22	Medicalisé	Bon état	À nettoyer	
122	Simple	Bon état	À nettoyer	

Items per page: 10 1 - 3 of 3

Figure 35:Interface « Liste des lits »

- **Interface « Ajouter un lit dans une chambre »**

Figure 36:Interface « Ajouter un lit dans une chambre »

4. Conclusion

Le sprint 2 a permis d'avancer significativement dans la génération de l'application web en se concentrant sur la gestion des ressources matérielles. Les fonctionnalités développées incluent la gestion des unités de soins, des chambres et des lits, avec des processus détaillés pour la consultation, la modification et l'ajout de ces ressources. Les tâches spécifiques ont été rigoureusement testées pour assurer leur conformité avec les exigences initiales. Les diagrammes de cas d'utilisation et de séquences ont aidé à clarifier et affiner les fonctionnalités. En conclusion, ce sprint a posé des bases solides pour l'application, préparant ainsi le terrain pour les prochaines itérations.

Chapitre 5 : Gestion de coordination des soins et des données

Introduction

Ce troisième sprint est dédié à l'amélioration des fonctionnalités clés de notre application, centrées sur la gestion des patients, des rendez-vous ainsi que la création et la consultation des rapports médicaux. Notre objectif principal est de fournir des outils efficaces aux secrétaires et aux docteurs pour simplifier leurs tâches quotidiennes et garantir une gestion optimale des informations médicales.

1. Backlog du sprint 3

1.1. Affectation des cas d'utilisations pour les acteurs

Acteurs	Cas d'utilisation
Secrétaire	Gérer les patients
	Gérer les rendez-vous
	Consulter la liste des rapports
Docteur	Consulter ces rendez-vous
	Gérer les rapports

Tableau 17: Classification des cas d'utilisations par acteur sprint 3

1.2. Sprint backlog

Pour commencer, nous avons établi une liste de tâches à accomplir pour ce Sprint. Le plan de cette itération comprend trois histoires d'utilisateurs telles que définies dans le tableau 18 .

Module	Id	User stories	Id	Tâches
Gestion des patients	1	En tant qu'un secrétaire je souhaite consulter la liste des patients réservés, en attentes de réservation et non examinés.	1.1	-Établir le prototype et créer le composant de la liste des patients réservés, en attentes de réservation et non examinés
	2	En tant qu'un secrétaire	1.2	-Développer le composant de consultation de la liste des patients réservés, en attentes de réservation et non examinés
			1.3	-Tester
			2.1	-Établir le prototype et créer le

		je souhaite ajouter, modifier et supprimer un patient.	2.2	composant de l'ajout. -Créer un popup de modification et de suppression.
			2.3	-Développer le composant de consultation de la liste des patients réservés, en attentes de réservation et non examinés
			2.4	-Tester
			3.1	-Établir le prototype et créer une fenêtre de détails d'un patient.
		En tant que secrétaire je souhaite consulter les détails d'un patient.	3.2	-Développer la méthode voire détail patient.
			3.3	-Tester
			1.1	-Établir le prototype et créer le composant de planification de rendez-vous.
		En tant qu'un secrétaire je souhaite prendre un rendez-vous pour un nouveau patient ou un patient libéré.	1.2	-Développer le composant de planification de rendez-vous.
			1.3	-Tester
			2.1	-Établir le prototype et créer le composant de visualisation des patients examinés et non examinés.
		En tant que docteur, je souhaite avoir accès à une liste des patients dont la date de rendez-vous est passée, qu'ils aient été examinés ou non.	2.2	-Développer le composant de visualisation des patients examinés et non examinés.
			2.3	-Tester
			3.1	-Établir le prototype et créer le composant de visualisation liste des patients dont le rendez-vous est à venir.
		En tant que docteur, je souhaite avoir accès à une liste des patients dont le rendez-vous est à venir.	3.2	-Développer le composant de visualisation liste des patients dont le rendez-vous est à venir.
			3.3	-Tester

Gestion des rapports	1	En tant qu'un docteur je souhaite créer un rapport pour un patient.	1.1	-Établir le prototype et créer le composant de création d'un rapport pour un patient
	1.2			-Développer le composant créer un rapport patient
	1.3			-Tester
	2	En tant qu'un docteur je souhaite consulter les rapports des patients qui me sont attribués.	2.1	-Établir le prototype et créer le composant de récupération des rapports des patients attribués
			2.2	-Développer la méthode de récupération des rapports patient attribués.
			2.3	-Tester
	3	En tant qu'un secrétaire je souhaite consulter tous les rapports des patients réservés et en attente de réservation des différents docteurs.	3.1	-Établir le prototype et créer le composant de récupération des rapports de tous les patients.
			3.2	-Développer la méthode de récupération des rapports de tous les patients.
			3.3	-Tester

Tableau 18: Backlog Sprint 3

2. Raffinement du Sprint

2.1 Identification des besoins fonctionnels

Pour ce sprint, les besoins fonctionnels identifiés concernent principalement la gestion des patients, des rendez-vous et des rapports dans un environnement médical.

- **Gestion des patients** : Il est nécessaire de pouvoir consulter la liste des patients selon leur statut (réservé, en attente de réservation, non examiné), ainsi que de pouvoir ajouter, modifier et supprimer des patients. De plus, la consultation des détails d'un patient doit être possible pour accéder à toutes les informations le concernant.
- **Gestion des rendez-vous** : Il est important de permettre aux secrétaires de prendre des rendez-vous pour de nouveaux patients ou des patients libérés. Pour les docteurs,

l'accès à une liste des patients dont la date de rendez-vous est passée, ainsi qu'à une liste des patients dont le rendez-vous est à venir, est essentiel pour organiser leur planning.

- **Gestion des rapports** : Les docteurs doivent pouvoir créer des rapports pour les patients qui leur sont attribués, ainsi que consulter ces rapports. Les secrétaires doivent également pouvoir consulter tous les rapports des patients réservés et en attente de réservation des différents docteurs, pour assurer le suivi de l'activité médicale.

2.2 Modèle global des cas d'utilisation du deuxième sprint

2.2.1 Diagramme des cas d'utilisation du deuxième sprint

Le troisième sprint utilise les différentes fonctionnalités décrites dans le diagramme de cas d'utilisation présenté dans la figure 37 :

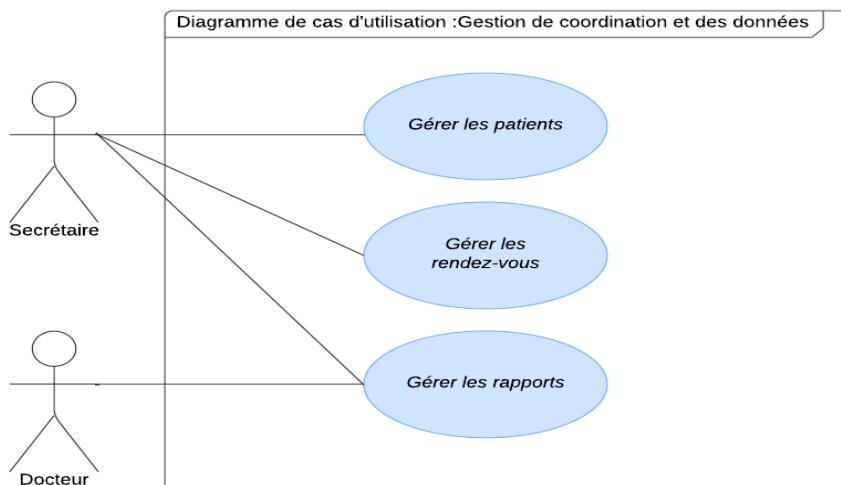


Figure 37: Diagramme de cas d'utilisation Sprint 3

2.2.2 Raffinement des cas d'utilisation du deuxième sprint

Nous allons détailler le cas d'utilisation « Gérer les patients » dans la figure 38 :

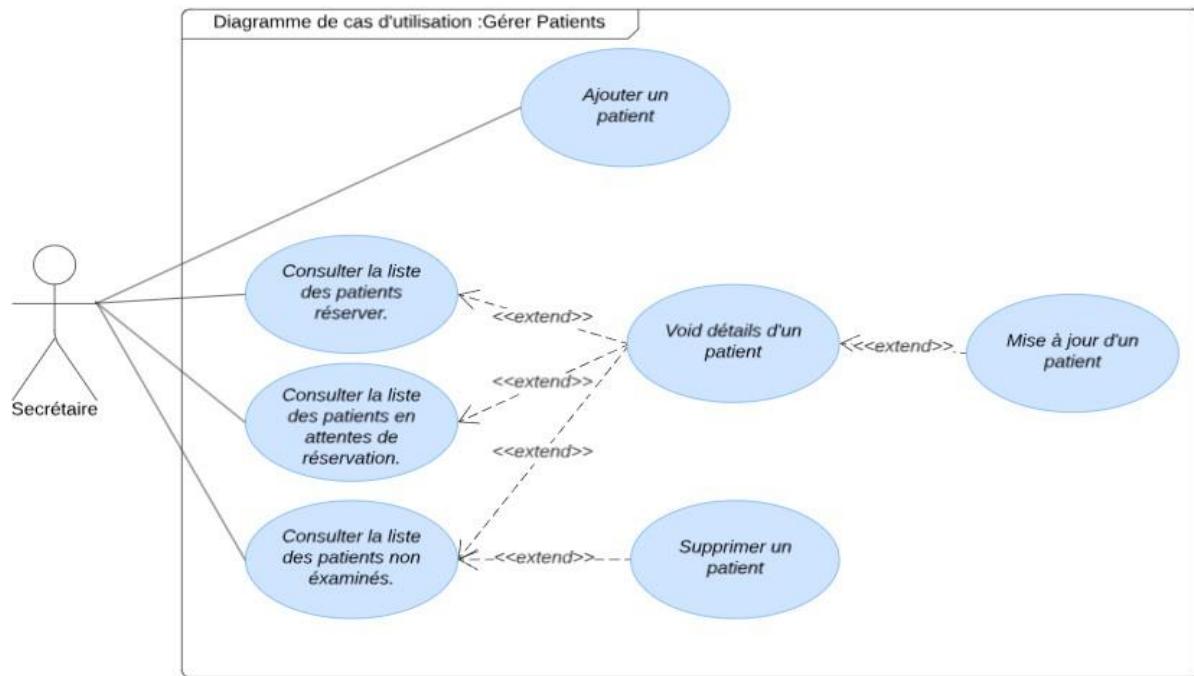


Figure 38: Diagramme de cas d'utilisation Gérer les patients

Le tableau 19 représente une description du cas d'utilisation « Gérer les patients ».

Cas d'utilisation	Gérer les patients
Acteurs	Secrétaire
Précondition	Le secrétaire doit s'authentifier afin d'accéder au système
Postcondition	Les modifications apportées aux patients, y compris l'ajout, la modification ou la suppression, sont enregistrées dans le système et mises à jour dans la base de données.
Scénario nominal	<ol style="list-style-type: none"> 1) Le secrétaire accède à la fonctionnalité de gestion des patients. 2) Le système affiche la liste des patients existants. 3) Le secrétaire peut choisir d'ajouter un nouveau patient en fournissant les informations nécessaires telles que le nom, date de naissance, identité, etc. 4) Le secrétaire doit ensuite sélectionner le médecin, la date et l'heure du rendez-vous associés au patient ajouté. 5) Le système enregistre le nouveau patient dans la base de données. 6) Le système enregistre le nouveau rendez-vous dans la base de données avec un état de visite « false ».

	<ol style="list-style-type: none"> 7) Le système dirige le secrétaire vers la liste des patients non examinés 8) Le secrétaire peut sélectionner un patient existant pour afficher ces informations. 9) Lorsque le formulaire est affiché, en double-cliquant sur chaque attribut, le secrétaire peut effectuer des modifications. 10) Le système enregistre les modifications apportées au patient. 11) Le secrétaire a la possibilité de supprimer un patient existant dans la liste des patients non examinés en cliquant sur le bouton supprimer et en confirmant la suppression. 12) Le système supprime le patient de la base de données.
Scénario alternatif	<ol style="list-style-type: none"> 1) Si aucun patient n'est disponible, le système affiche un message indiquant aucune donnée disponible. 2) Si une erreur se produit lors de l'ajout, de la modification ou de la suppression d'un patient, le système affiche un message d'erreur et le secrétaire est informé.

Tableau 19 : Description textuelle du cas d'utilisation Gérer les patients

La figure ci-dessous présente le cas d'utilisation « Gérer les rendez-vous » :

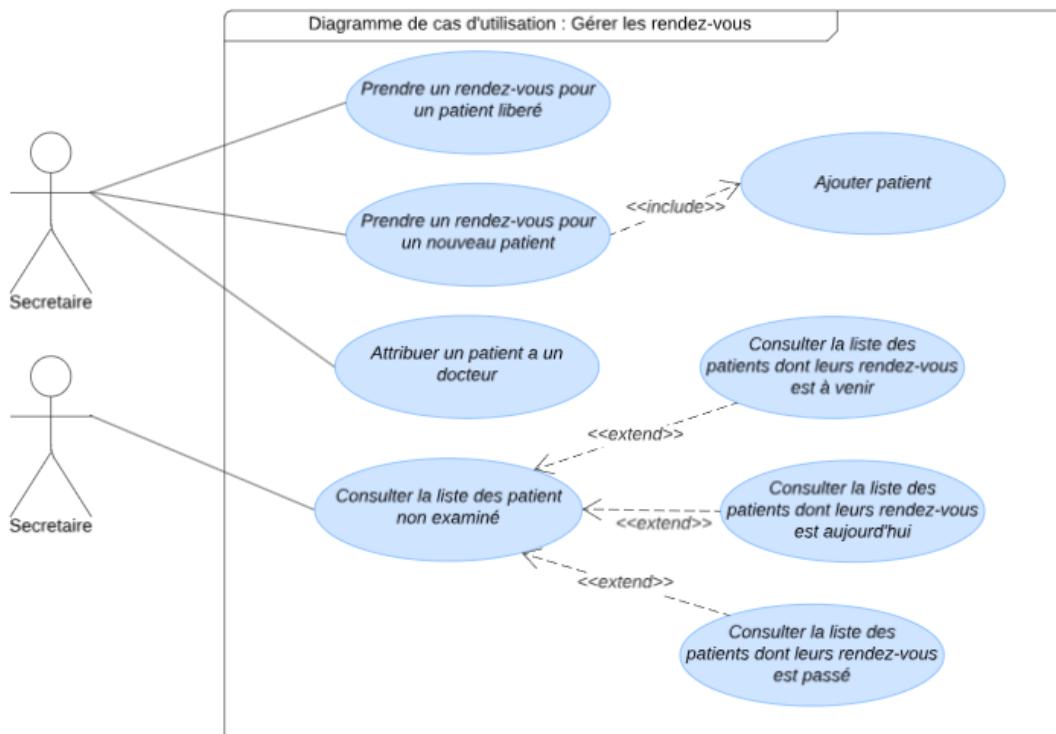


Figure 39 : Diagramme de cas d'utilisation Gérer les rendez-vous

Le tableau 20 présente le scénario de cas d'utilisation « Gérer les rendez-vous » :

Cas d'utilisation	Gérer les rendez-vous
Acteurs	Secrétaire, Docteur
Précondition	Le secrétaire doit s'authentifier afin d'accéder au système
Postcondition	Après avoir enregistré un rendez-vous pour un patient, le système met à jour la base de données pour refléter la planification du rendez-vous.
Scénario nominal	<ol style="list-style-type: none"> 1) Le secrétaire accède à la liste des patients archivés. 2) Le secrétaire choisie la fonctionnalité des planifications des rendez. 3) Le secrétaire sélectionne le docteur et saisit la date et le temps de rendez-vous. 4) Le système enregistre le rendez-vous de ce patient dans la base de données. 5) Le docteur peut consulter la liste des patients avec la date de visite planifié dont la date de rendez-vous est passée, qu'ils aient été examinés ou non. 6) Le docteur peut consulter la liste des patients dont le rendez-vous est à venir pour organiser son planning. 7) Le docteur peut consulter la liste de tous les patients qui lui sont attribués et en attente.
Scénario alternatif	<ol style="list-style-type: none"> 1) Si aucun patient n'est disponible, le système affiche un message indiquant aucune donnée disponible. 2) Si une erreur se produit lors du rendez-vous d'un patient le système affiche un message d'erreur et le secrétaire est informé.

Tableau 20 : Description textuelle du cas d'utilisation Gérer les rendez-vous

La figure 40 présente le cas d'utilisation « Gérer les rapports » :

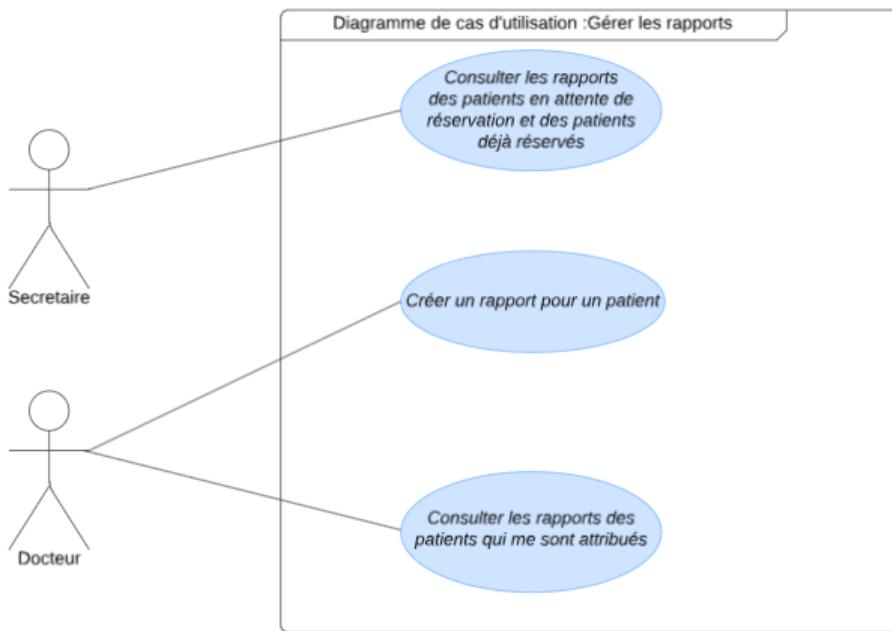


Figure 40 : Diagramme de cas d'utilisation Gérer les rapports

Le tableau 21 décrit le scénario d'utilisation intitulé "Gérer les rapports".

Cas d'utilisation	Gérer les rapports
Acteurs	Secrétaire, Docteur
Précondition	Le secrétaire doit s'authentifier afin d'accéder au système
Postcondition	Après avoir enregistré un rapport pour un patient, le système met à jour la base de données pour refléter la création du rapport.
Scénario nominal	<ol style="list-style-type: none"> 1) Le docteur accède à la fonctionnalité d'ajout d'un rapport à un patient sélectionné 2) Le docteur saisit les données de rapport tel que la fièvre, la maladie les remarques, etc... 3) Le système enregistre les données de rapport relatives à ce patient dans la base de données avec la date de création de rapport d'aujourd'hui. 4) Le docteur peut consulter la liste des rapports rédigés par lui triés par date de création. 5) Le secrétaire peut consulter la liste des rapports rédigés par tous les médecins triés par date de création.
Scénario alternatif	<ol style="list-style-type: none"> 1) Si aucun rapport n'est disponible, le système affiche un message indiquant aucune donnée disponible. 2) Si une erreur se produit lors de la création ou de la consultation d'un rapport, le système affiche un message d'erreur et l'utilisateur est informé.

Tableau 21 : Diagramme de cas d'utilisation Gérer les rapports

2.3 Diagramme de séquence de sprint 3

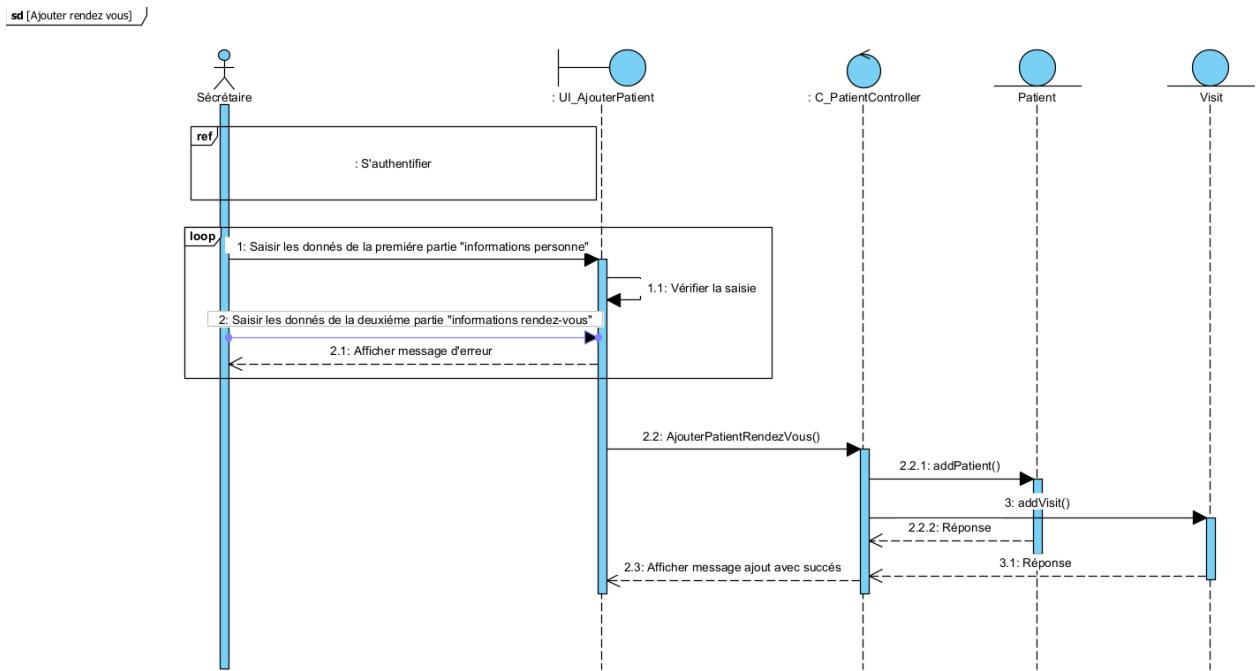


Figure 41: Diagramme de séquence "Ajouter rendez-vous"

2.4 Diagramme de classe de sprint 3

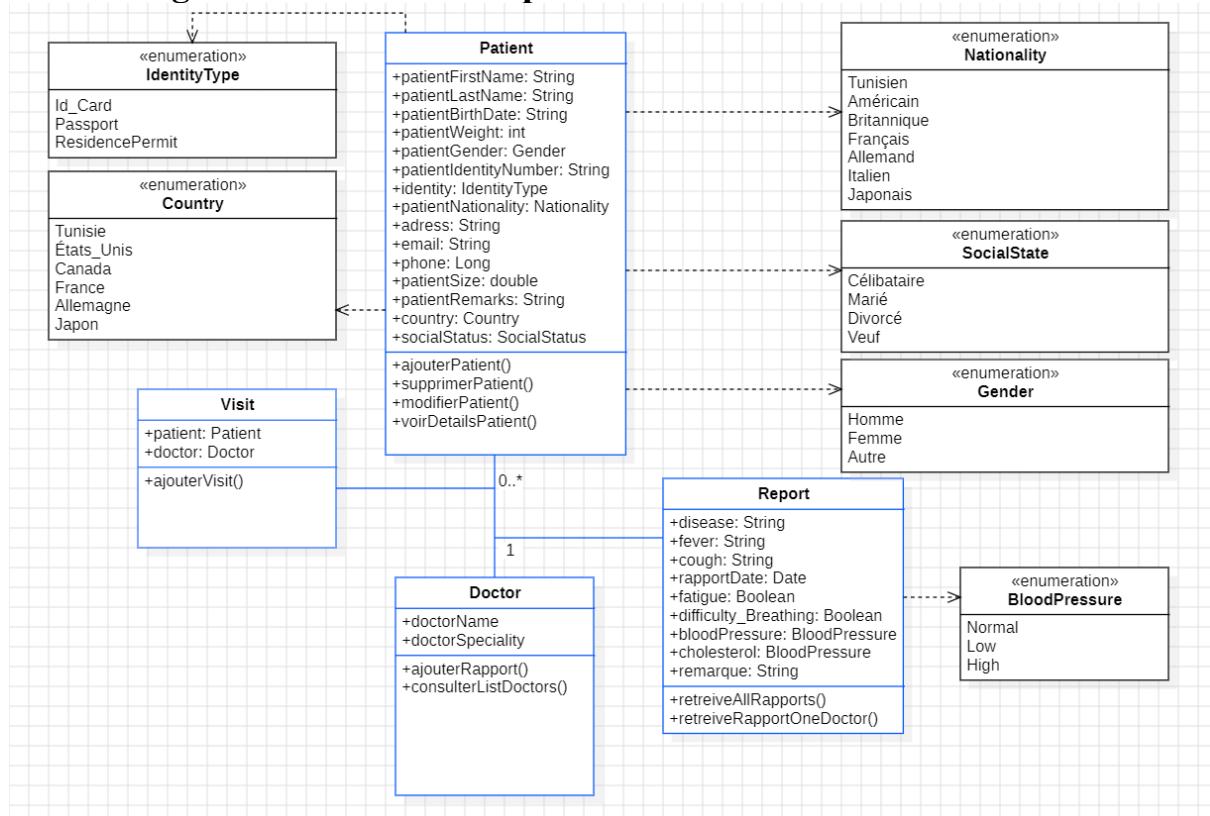


Figure 42: Diagramme de classe de Sprint 3

3. Test et interfaçage

3.1 Test sprint 3

Cas	Résultat
Gérer les patients	Conforme
Gérer rendez-vous	
Consulter la liste des rapports	
Consulter ces rendez-vous	
Gérer les rapports	

Tableau 22: Test du sprint 3

3.2 Interfaçages

- Interface « Liste des patients d'un docteur »

Figure 43:Interface « Liste des patients d'un docteur »

- **Interface « Liste des patients non examinés pour Secrétaire »**

	Détails	Identité	Prénom	Nom	Sexe	Date de Naissance	Actions
<input type="checkbox"/>		12453678	Itifi	safa	Femme	31/05/2002	Consulter Supprimer
<input type="checkbox"/>		15789633	Itifii	ghaith	Homme	31/05/2001	Consulter Supprimer

Figure 44:Interface « Liste des patients non examinés pour Secrétaire »

- **Interface « Ajouter patient pour Secrétaire »**

Figure 45:Interface « Ajouter patient pour Secrétaire »

- **Interface « Rendez-vous pour Secrétaire »**

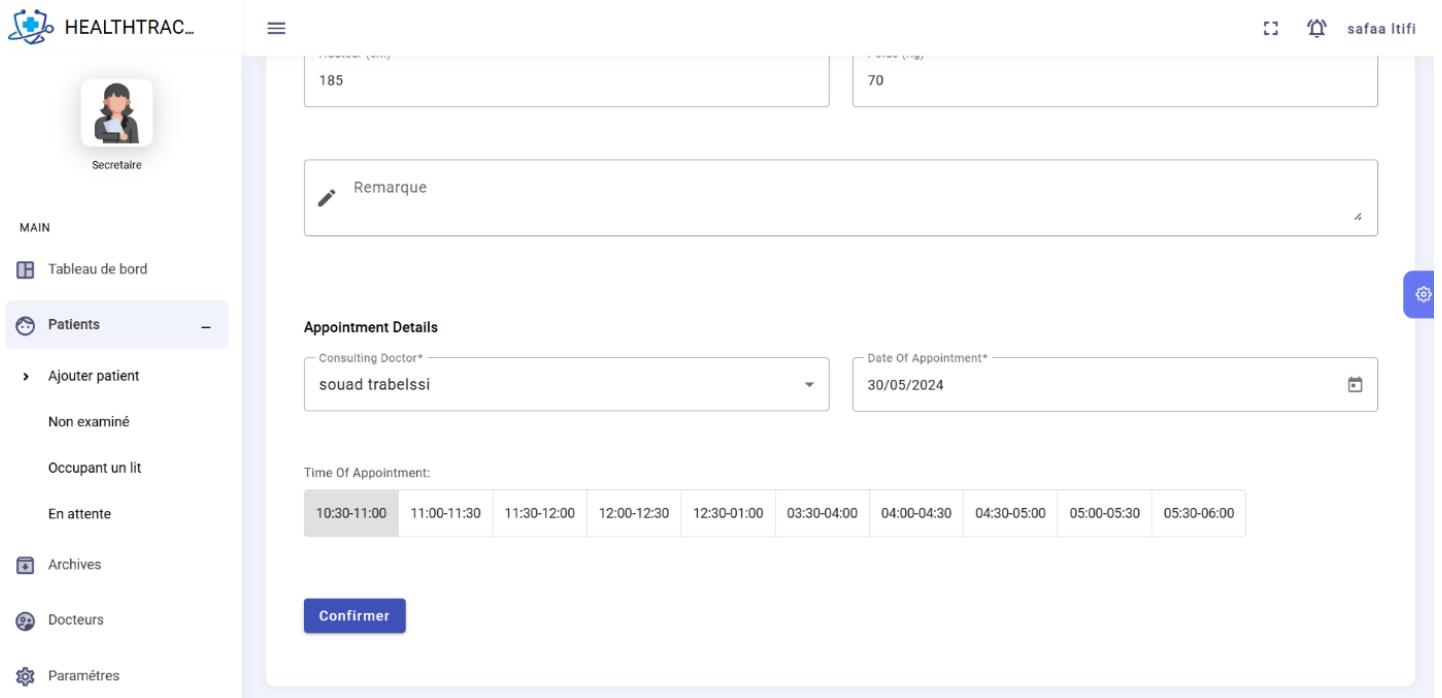


Figure 46: Interface « Rendez-vous pour Secrétaire »

- **Interface « Rapports des patients pour Secrétaire »**

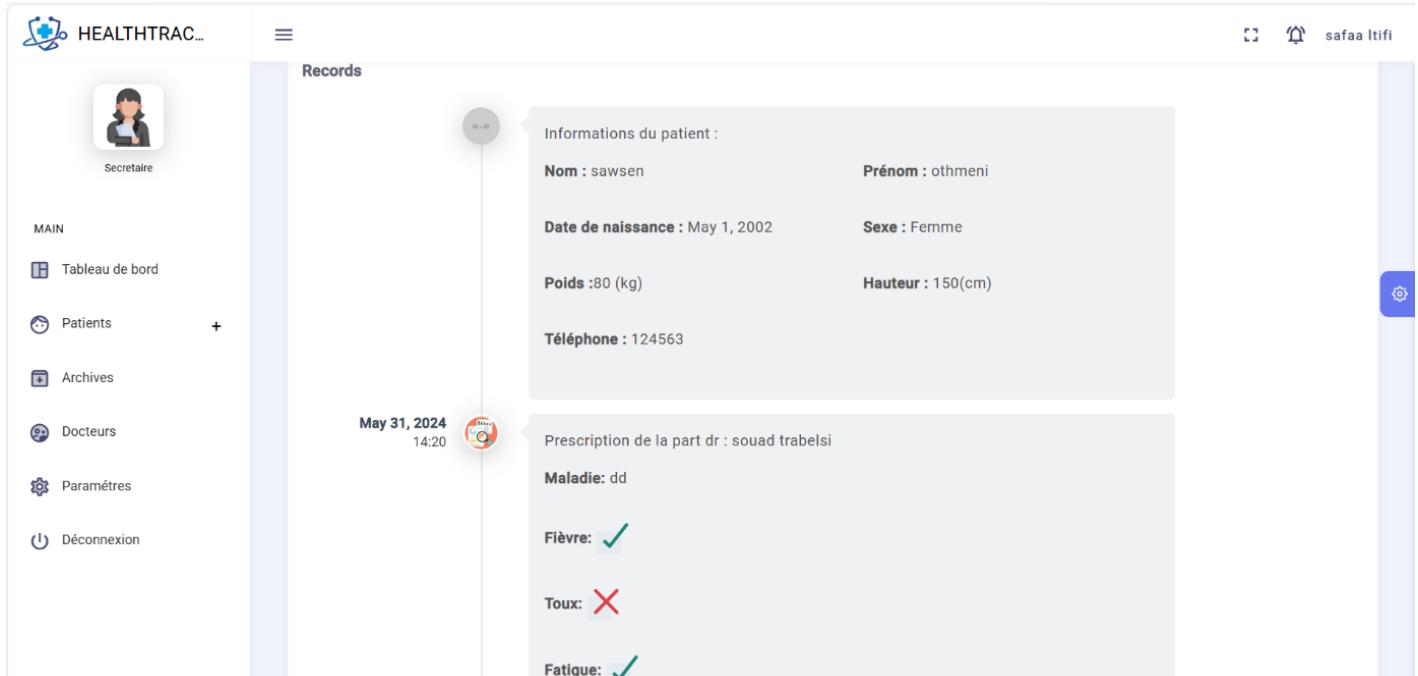


Figure 47: Interface « Rapports des patients pour Secrétaire »

Conclusion

Dans ce troisième sprint, notre objectif était d'améliorer les fonctionnalités clés de notre application pour une gestion optimale des patients, des rendez-vous et des rapports médicaux. Grâce à une planification minutieuse, nous avons réussi à mettre en œuvre les fonctionnalités définies dans notre backlog. Ces progrès renforcent notre capacité à fournir des soins de santé de qualité et une expérience utilisateur améliorée. En résumé, ce sprint a été une avancée significative vers notre objectif de créer une plateforme médicale efficace.

Chapitre 6 : Gestion des flux patients

Introduction

Le quatrième Sprint se concentre sur l'optimisation de la gestion des entrées et sorties des patients, la réservation des lits, les déplacements et prolongations, ainsi que la gestion des archives. L'objectif est d'améliorer l'efficacité des processus de gestion des patients pour garantir des soins de santé de qualité et une meilleure expérience pour le personnel médical.

Je vais commencer par présenter un diagramme global, puis je détaillerai chaque cas dans un diagramme de cas d'utilisation spécifique et un tableau détaillé pour une meilleure compréhension et une communication efficace des exigences du sprint et pour la fin je vais présenter la partie de l'intégration des notifications.

1. Backlog du sprint 4

1.1. Affectation des cas d'utilisations pour les acteurs

Acteurs	Cas d'utilisation
Secrétaire	Gérer les entrées/sorties patients
	Gérer les réservations
	Gérer les déplacements et les prolongations
	Gérer les archives

Tableau 23:Classification des cas d'utilisations par acteur Sprint 4

1.2. Sprint backlog

Pour commencer, nous avons établi une liste de tâches à accomplir pour ce Sprint. Le plan de cette itération comprend quatre histoires d'utilisateurs telles que définies dans le tableau 24

Module	Id	User stories	Id-tâches	Tâches
Gestion des entrées/sorties patients	1	En tant qu'un secrétaire je souhaite entrer la date d'entrée et sortie réelle.	1.1 1.2	-Établir le prototype et créer le composant de la création de la date sortie réelle et la date d'entrée réelle. -Développer le composant de la

			1.3	création de la date sortie réelle et la date d'entrée réelle. -Tester
	2	En tant que secrétaire je souhaite saisir l'état de sortie d'un patient.	2.1 2.2 2.3	-Établir le prototype et créer le composant de libération. -Développer le composant de libération -Tester
Gestion des réservations	1	En tant qu'un secrétaire je souhaite avoir une recherche opérationnelle pour optimiser l'allocation des lits.	1.1 1.2 1.3	-Établir le prototype et créer le composant de recherche opérationnelle. -Développer le composant recherche opérationnelle -Tester
	2	En tant que secrétaire je souhaite réserver un lit pour un patient dans une unité et chambre spécifique.	2.1 2.2 2.3	-Établir le prototype et créer le composant de la liste des lits disponibles et l'affectation. -Développer le composant de la liste des lits disponibles et l'affectation. -Tester
	3	En tant qu'un secrétaire je souhaite saisir les détails d'un accompagnant pour un patient spécifique.	3.1 3.2 3.3	-Établir le prototype et créer le composant d'ajout de l'accompagnant. -Développer le composant d'ajout de l'accompagnant. -Tester
	4	En tant qu'un secrétaire je souhaite consulter les détails d'un accompagnant et les modifier pour un patient spécifique.	4.1 4.2	-Établir le prototype et créer le composant de visualisation les détails d'un accompagnant et le composant de modification. -Développer le composant de

			4.3	visualisation les détails d'un accompagnant et le composant de modification. -Tester
	5	En tant qu'un secrétaire je souhaite consulter les détails d'un lit pour un patient spécifique.	5.1 5.2 5.3	-Établir le prototype et créer le composant de visualisation les détails d'un lit occupé par un patient. -Développer le composant de visualisation les détails d'un lit occupé par un patient -Tester
Gestion des déplacements et des prolongations	1	En tant qu'un secrétaire je souhaite déplacer un patient d'une chambre à une autre dans la même unité ou dans une unité différente.	1.1 1.2 1.3	-Établir le prototype et créer le composant des détails de déplacement. -Développer le composant des détails de déplacement. -Tester
	2	En tant que secrétaire je souhaite conserver les historiques des changements de chaque patient.	2.1 2.2 2.3	-Établir le prototype et créer le composant des historiques de déplacement. -Développer le composant des historiques de déplacement. -Tester
	3	En tant qu'un secrétaire je souhaite prolonger la date de sortie provisionnelle d'un patient.	3.1 3.2 3.3	-Établir le prototype et créer le composant de prolongation. -Développer le composant de prolongation. -Tester
	4	En tant que secrétaire je souhaite conserver les historiques des prolongations de chaque patient.	4.1	-Établir le prototype et créer le composant des historiques de de prolongation.

			4.2	-Développer le composant des historiques de prolongation.
			4.3	-Tester
Gestion des archives	1	En tant qu'un secrétaire je souhaite consulter la liste des patients libérés.	1.1	-Établir le prototype et créer le composant de la liste des patients libérés.
			1.2	-Développer le composant de la liste des patients libérés.
			1.3	-Tester
	2	En tant qu'un secrétaire je souhaite consulter les rapports des patients libérés.	2.1	-Établir le prototype et créer le composant de visualisation des rapports pour les patients
			2.2	-Développer le composant de visualisation des rapports pour les patients.
			2.3	-Tester
	3	En tant qu'un secrétaire je souhaite prendre un rendez-vous pour un patient libéré.	3.1	-Établir le prototype et créer le composant de l'ajout d'un rendez-vous à un patient libéré.
			3.2	-Développer le composant de l'ajout d'un rendez-vous à un patient libéré
			3.3	-Tester

Tableau 24 : Backlog Sprint 4

2. Raffinement du Sprint

2.1 Identification des besoins fonctionnels

L'identification des besoins fonctionnels est essentielle pour définir les fonctionnalités et les exigences du système. Dans le cadre du quatrième sprint, les besoins fonctionnels suivants ont été identifiés :

- **Gestion des entrées et sorties des patients** : L'un des aspects fondamentaux de la gestion des patients est de suivre précisément leurs dates d'entrée et de sortie de

l'établissement de santé. Ainsi, le système doit permettre aux secrétaires d'entrer les dates d'entrée et de sortie réelles des patients, assurant ainsi une documentation précise de leur séjour. De plus, il est nécessaire de pouvoir enregistrer l'état de sortie des patients pour maintenir un suivi complet de leur progression dans le processus de soins.

- **Réservation des lits :** La réservation des lits constitue une tâche cruciale pour garantir une utilisation optimale des ressources disponibles. Pour ce faire, le système doit offrir une fonction de recherche opérationnelle permettant aux secrétaires de trouver rapidement des lits disponibles et de les attribuer efficacement aux patients selon l'unité et le type de chambre. De plus, il est essentiel de saisir les détails de l'accompagnant du patient pour assurer une prise en charge adéquate.
- **Gestion des déplacements et prolongations :** La flexibilité dans la gestion des déplacements et des prolongations des patients est essentielle pour répondre aux besoins changeants et aux exigences cliniques. Ainsi, le système doit permettre aux secrétaires de déplacer un patient d'une chambre à une autre, que ce soit dans la même unité ou dans une unité différente. De plus, il est nécessaire de conserver un historique complet des changements effectués pour chaque patient, y compris les déplacements et les prolongations. Il est également important de déplacer l'accompagnant avec le patient en fonction du type de chambre choisi.
- **Gestion des archives :** Pour une gestion efficace des patients, il est crucial d'avoir accès aux informations pertinentes sur les patients libérés. Ainsi, le système doit permettre aux secrétaires de consulter facilement la liste des patients libérés ainsi que leurs rapports et de prendre un rendez-vous à ces patients sans avoir à les ajouter à nouveau et à saisir leurs données une autre fois. Ces fonctionnalités garantissent une traçabilité complète des patients.

2.2 Modèle global des cas d'utilisation du Sprint

2.2.1 Diagramme de cas d'utilisation du sprint

Le quatrième sprint se concentre sur le cas d'utilisation qui décrit les différentes fonctionnalités représentées dans le diagramme de cas d'utilisation illustré ci-dessous

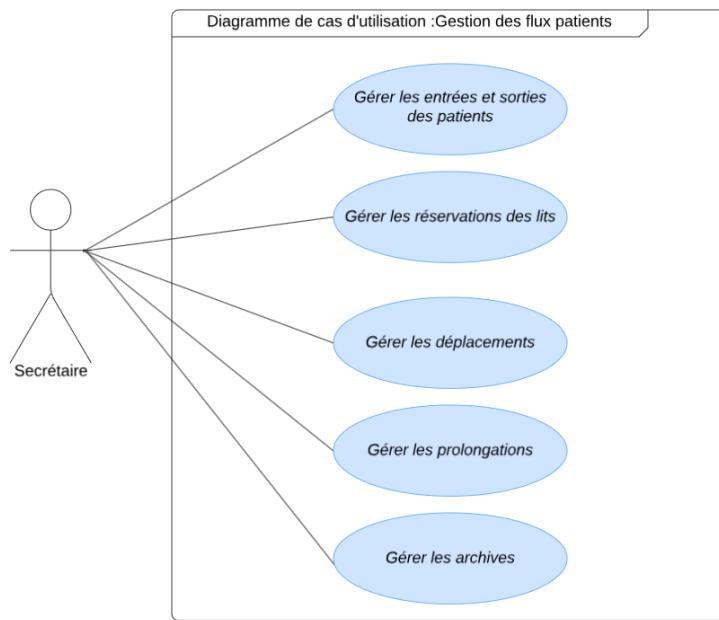


Figure 48 : Diagramme de cas d'utilisation Sprint 4

2.2.2 Raffinement des cas d'utilisations

Dans cette section, nous allons détailler les principaux cas d'utilisation du quatrième sprint, en commençant par celui de la gestion des entrées et sorties des patients :

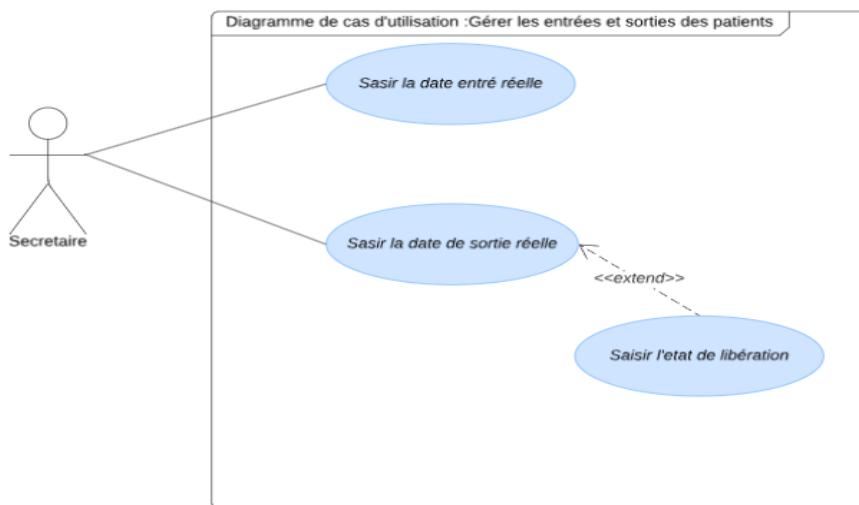


Figure 49 : Diagramme de cas d'utilisation Gérer les entrées et sorties des patients

Le tableau suivant représente une description du cas d'utilisation « Gérer les entrées et sorties des patients »

Cas d'utilisation	Gérer les entrées et sorties des patients
Acteurs	Secrétaire
Précondition	Le secrétaire doit s'authentifier afin d'accéder au système Le patient est déjà enregistré dans la base de données. Le patient a déjà fait une réservation.
Postcondition	La date d'entrée réelle et la date de sortie réelle sont enregistrées.
Scénario nominal	<ol style="list-style-type: none"> 1) Le secrétaire accède à la fonctionnalité de séjour de patient sélectionné. 2) Le système affiche les détails de séjour tels que la date d'entrée provisionnelle et la date de sortie provisionnelle, l'emplacement du patient dans l'unité ainsi que le numéro de chambre précis. 3) Le secrétaire fait une double clique sur la zone de date d'entrée réelle. 4) Le système affiche un calendrier pour saisir la date d'entrée réelle. 5) Le secrétaire saisir la date d'entrée réelle. 6) Le système enregistre cette date avec modification d'état de lit de « réservé » à l'état « occupé ». 7) Le système affiche une alerte informant le secrétaire que la date d'entrée est bien enregistrée. 8) Le secrétaire fait double clique sur la zone de la date de sortie 9) Le système affiche un calendrier pour saisir la date d'entrée réelle. 10) Le secrétaire saisir la date de sortie réelle. 11) Le système vérifie si la date d'entrée réelle est déjà saisie et que cette date est postérieure à la date de sortie 12) Le système enregistre la date de sortie réelle de ce patient avec l'occupant le système modifie l'état de lit de « occupé » à « disponible » et l'état de nettoyage à « nettoyer ». 13) Le système affiche une alerte de confirmation de la date et dirige le secrétaire à la page de libération. 14) Le secrétaire saisie l'état de libération (critique, stable, transféré.) si l'état de libération est « transféré » la secrétaire saisie le nom de l'établissement. 15) Le système libère ce patient en le supprimant de la base de données « BedLocked ».

	16) Le système affiche une alerte de libération avec succès.
Scénario alternatif	<ol style="list-style-type: none"> 1) Si la date de sortie est inférieure à la date d'entrée réelle, le système affiche un message provenant de backend que la date d'entrée réelle est postérieure à la date de sortie que vous avez saisie. 2) Si une erreur se produit lors de libération, le système affiche un message d'erreur et le secrétaire est informé.

Tableau 25 : Description textuelle du cas d'utilisation Gérer les entrées et sorties des patients

Le schéma suivant représente le diagramme de cas d'utilisation « Gérer les réservations ».

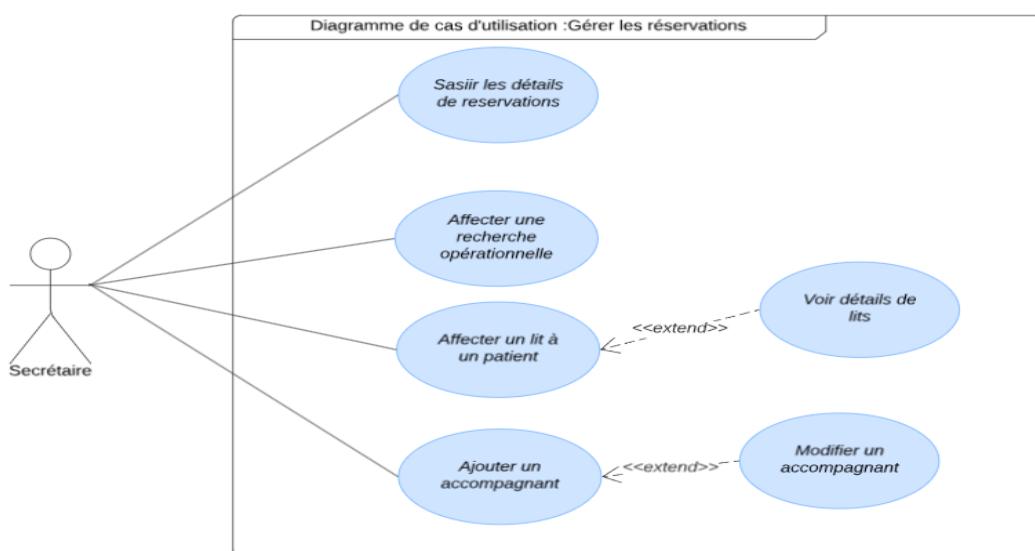


Figure 50 : Diagramme de cas d'utilisation Gérer les réservations

Le tableau 26 représente une description du cas d'utilisation Gérer les réservations :

Cas d'utilisation	Gérer les réservations
Acteurs	Administrateur
Précondition	Le secrétaire doit s'authentifier afin d'accéder au système Le patient est déjà enregistré. Le patient est examiné par un docteur.
Postcondition	Les réservations sont enregistrées dans la base de données.
Scénario nominal	<ol style="list-style-type: none"> 1) Le secrétaire accède à la liste des patients non réservés. 2) Le secrétaire sélectionne un patient et accède à la fonctionnalité des réservations. 3) Le secrétaire saisie la date d'entrée prévisionnelle la date de sortie prévisionnelle l'unité et le type de chambre (double,

	<p>collective, simple).</p> <p>4) Si le type de chambre sélectionné est double le secrétaire choisi s'il est accompagné ou non.</p> <p>5) Le système dirige la secrétaire vers la page de la liste des chambres disponible avec les critères saisie :</p> <ul style="list-style-type: none"> - Les chambres affichées contient des lits en bonne état - Les chambres affichées contiennent des lits disponibles dans la période entre la date d'entré provisionnelle et la date de sortie provisionnelle. - S'il est accompagné les chambres de type double affiché contient nécessairement un lit simple et médicalisé ou deux lits simples. - Les chambres affichées contiennent des lits leur poids est postérieur au poids du patient sélectionné. <p>6) Le secrétaire affecte le patient à un lit.</p> <p>7) Le système affiche une alerte de confirmation de l'affectation.</p> <p>8) Le système enregistre les dates d'admission relative au patient sélectionné avec son lit dans la base de données.</p> <p>9) Si le patient est accompagné le secrétaire saisir les données de l'accompagnant tels que (nom, prénom, type de relation...).</p> <p>10) Le système affiche une alerte de l'enregistrement avec succès de l'accompagnant et l'enregistre dans la base de données.</p> <p>11) Le secrétaire peut ensuite modifier les détails de ce accompagnant.</p>
Scénario alternatif	<p>1) Lorsque le secrétaire accède à la liste des patients non réservés, s'il y a aucun patient n'est disponible pour la réservation, le système affiche un message indiquant qu'il n'y a aucun patient disponible pour la réservation.</p> <p>2) Si aucune chambre disponible ne répond aux critères saisis, le système affiche un message informant le secrétaire qu'aucune chambre répondant aux critères n'est disponible.</p>

Tableau 26 : Description textuelle du cas d'utilisation Gérer les réservations

Le schéma 51 représente le diagramme de cas d'utilisation « Gérer les déplacements et les prolongations ».

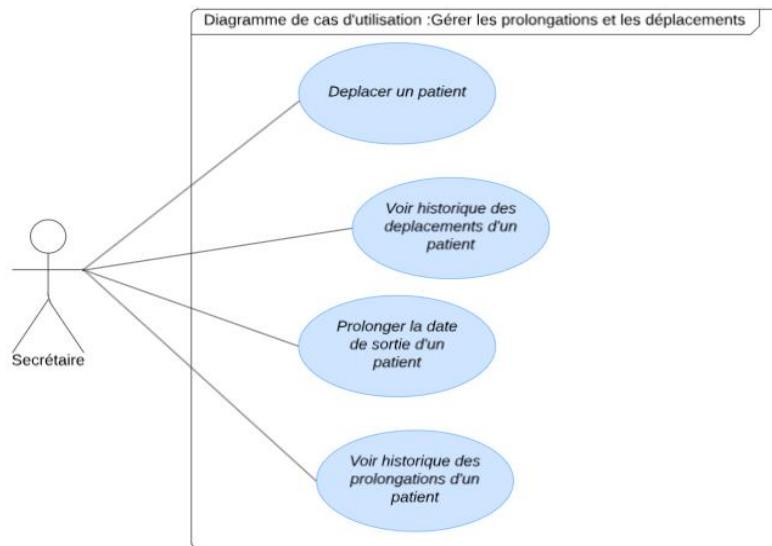


Figure 51 : Diagramme de cas d'utilisation Gérer les prolongations et les déplacements

Le tableau 27 représente une description du cas Gérer les déplacements et les prolongations :

Cas d'utilisation	Gérer les déplacements et les prolongations
Acteurs	Secrétaire
Précondition	Le secrétaire doit s'authentifier afin d'accéder au système. Le patient doit déjà avoir effectué une réservation de lit.
Postcondition	Les prolongations et les déplacements sont enregistrées dans le système.
Scénario nominal	<ol style="list-style-type: none"> 1) Le secrétaire accède à la liste des personnes réservées et sélectionne le patient. 2) Le secrétaire accède à la fonctionnalité de déplacement. 3) Le secrétaire saisie les dates d'entrée et sortie provisionnelles si lors de déplacements va modifier aussi les dates, ainsi saisie l'unité a déplacer et le type de chambre. 4) Si le patient sélectionné est déjà accompagné et le type choisi lors de déplacement un type simple l'accompagnant va être supprimer de la base de données. 5) Le secrétaire choisie une chambre parmi les chambres affichées et déplace le patient. 6) Le système enregistre les nouvelles données de séjours du patient. 7) Le système affiche une alerte de déplacement effectué avec succès. 8) Le système enregistre l'historique des déplacements la date de

	<p>déplacement.</p> <p>9) Le secrétaire accède à la fonctionnalité de prolongation.</p> <p>10) Le système affiche un calendrier pour saisir la date de sortie de prolongation.</p> <p>11) Le système vérifie si la date d'entrée réelle est déjà saisie et que la date de prolongation saisie est postérieure de la date de sortie provisionnelle.</p> <p>12) Le système affiche une alerte que la date de sortie est prolongée avec succès.</p> <p>13) Le système enregistre l'historique des prolongations la date prévue et la nouvelle date de sortie.</p>
Scénario alternatif	<p>1) Si aucune chambre ne répond aux critères saisis, le système affiche un message informant le secrétaire qu'aucune chambre répondant aux critères n'est disponible</p> <p>2) Si la date de prolongation est inférieure à la date de sortie prévus le système affiche un alerte contenant un message provenant du backend.</p> <p>3) Si le secrétaire entre la date de prolongation avant de saisir la date d'entrée réelle de ce patient le système affiche une erreur provenant du backend.</p>

Tableau 27 : Description textuelle du cas d'utilisation Gérer les prolongations et les déplacements

Le schéma 52 représente le diagramme de cas d'utilisation « Gérer les archives ».

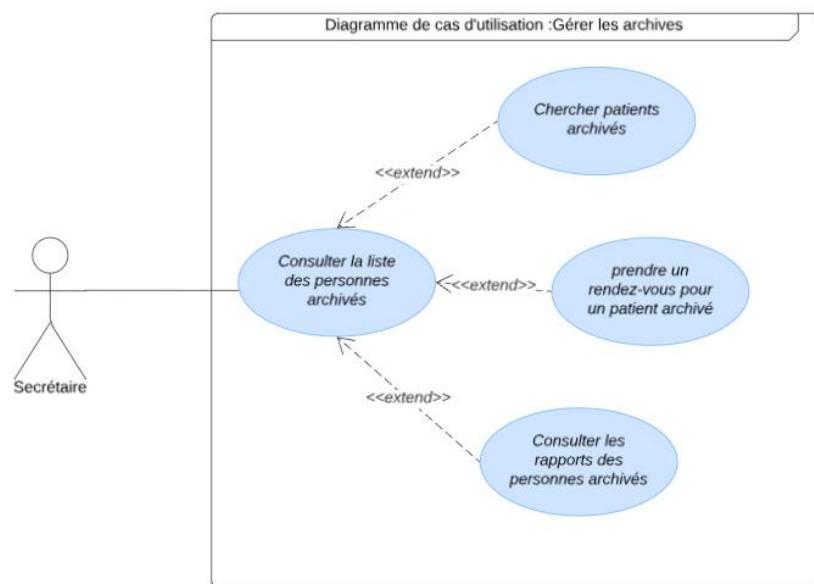


Figure 52: Diagramme de cas d'utilisation Gérer les archives

Le tableau 28 représente une description du cas d'utilisation Gérer les archives :

Cas d'utilisation	Gérer les archives
Acteurs	Administrateur
Précondition	Le secrétaire doit s'authentifier afin d'accéder au système. Le patient doit déjà être libéré.
Postcondition	Les informations du patient libéré sont enregistrées dans la liste des archives.
Scénario nominal	<ol style="list-style-type: none"> 1) Après avoir saisir la date de sortie réelle et l'état de libération d'un patient, il est enregistré automatiquement dans la liste des archives. 2) Le secrétaire accède à la liste des archives. 3) Le secrétaire sélectionne un patient libéré et accède à la fonctionnalité de visualisation des rapports de ce patient. 4) Le système affiche la liste des rapports de ce patient, indiquant la date de saisie de chaque rapport ainsi que le nom du médecin ayant rédigé le rapport 5) Le secrétaire accède à la fonctionnalité des rendez-vous. 6) Le secrétaire choisie le docteur le temps de Viste et la date. 7) Le système enregistre la visite et affiche une alerte de succès. 8) Le système dirige le secrétaire à la page des patients non examinés.
Scénario alternatif	<ol style="list-style-type: none"> 1) Si une erreur se produit lors de l'enregistrement de la visite, le système affiche un message d'erreur et le secrétaire est informé

Tableau 28 : Description textuelle du cas d'utilisation Gérer les archives

2.3 Diagramme de séquence sprint 4

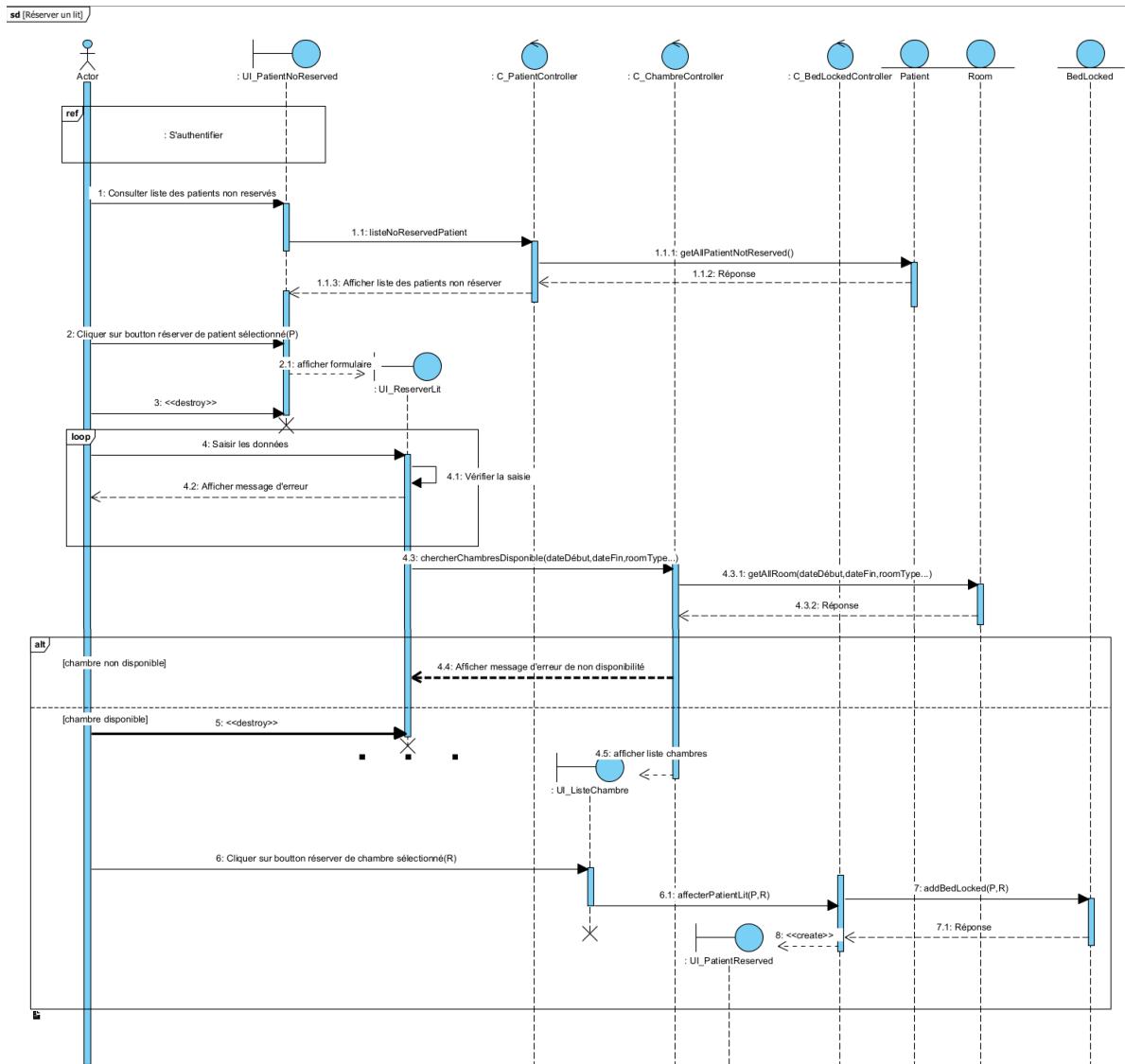


Figure 53: Diagramme de séquence "Réserver lit"

2.4 Diagrammes de classe Sprint 4

Vous trouvez dans la page suivante le diagramme de classe de sprint 4 :

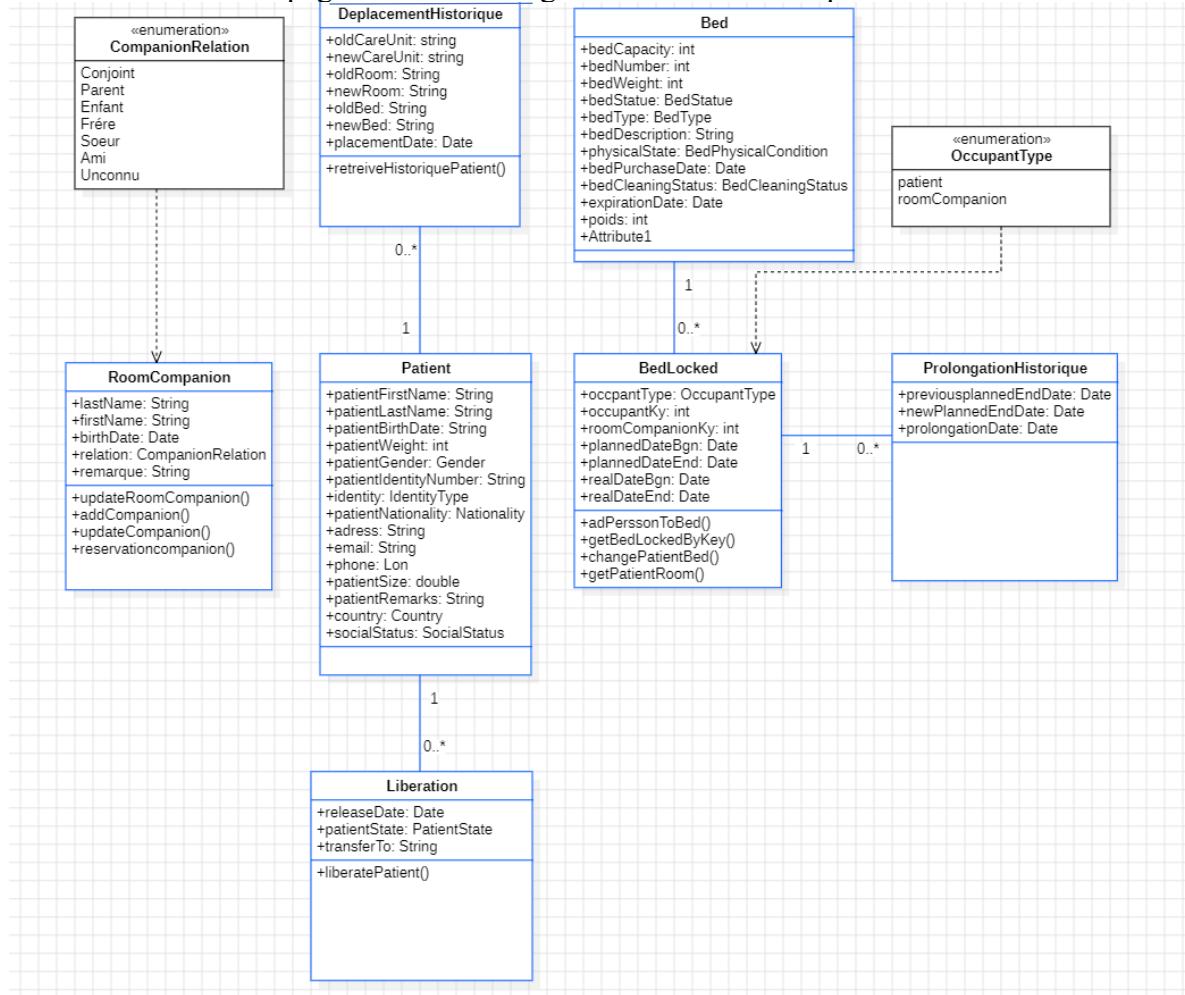


Figure 54 : Diagramme de classe Sprint 4

3. Intégration de WebSocket et Gestion des Notifications :

Dans le cadre de notre projet, nous avons intégré la technologie WebSocket pour permettre une communication bidirectionnelle et en temps réel entre le frontend Angular et le backend Spring Boot. Cette intégration nous a permis d'implémenter un système de gestion des notifications pour informer les utilisateurs des événements importants dans l'application, en particulier la libération des lits et la nécessité de nettoyer ces lits.

3.1 Configuration de WebSocket

Pour configurer WebSocket dans notre application Spring Boot, nous avons créé une classe `WebSocketConfig` qui implémente l'interface `WebSocketMessageBrokerConfigurer`. Cette configuration a permis d'activer le support WebSocket et de définir les points de terminaison et les autorisations nécessaires.

3.2Gestion des Notifications

La gestion des notifications était une exigence importante de notre projet, notamment pour informer le personnel médical de la nécessité de nettoyer les lits libérés. Pour cela, nous avons mis en place un système de gestion des notifications qui repose sur une table de notifications dans la base de données.

Structure de la Table de Notifications :

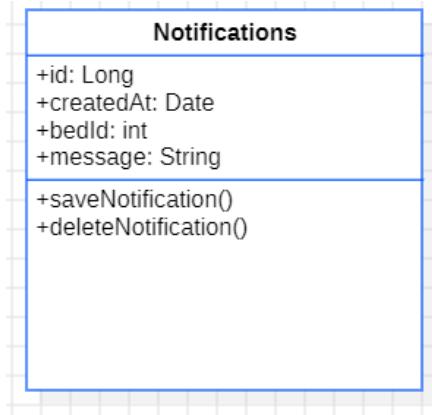


Figure 55 : Table Notification

3.3Processus de Notification

Lorsqu'un patient est libéré de son lit, le système déclenche automatiquement l'envoi d'une notification via WebSocket « Le patient Foulen est libéré ». Simultanément, une entrée est ajoutée à la table de notifications avec le numéro de lit du patient et le message approprié « Le lit numéro de lit doit être nettoyé ».

3.4Nettoyage Automatique des Notifications

Une fois que le lit associé à une notification est nettoyé, l'administrateur met à jour le statut de nettoyage de ce lit. Lorsqu'un lit est nettoyé, la notification associée à ce lit est automatiquement supprimée de la table de notifications.

3.5Avantages et Perspectives

L'intégration de WebSocket et de la gestion des notifications a permis d'améliorer l'expérience utilisateur en fournissant des informations en temps réel sur les événements importants dans l'application. Cette fonctionnalité a contribué à une meilleure coordination des tâches et à une gestion plus efficace des ressources dans l'établissement médical.

Pour l'avenir, nous envisageons d'étendre cette fonctionnalité pour prendre en charge d'autres types de notifications et d'événements dans l'application.

4. Test et interfaçage

4.1 Test sprint 3

Cas	Résultat
Gérer les entrées/sorties patients	Conforme
Gérer les réservations	
Gérer les déplacements et les prolongations	
Gérer les archives	
Consulter les entrées/sorties patients	

Tableau 29: Test sprint 4

4.2 Interfaçages

- Interface « Patients en attentes »

The screenshot shows the 'Patients' section of the HEALTHTRAC application. The left sidebar shows 'Tableau de bord' and 'Patients'. The main area is titled 'Liste des patients > Les patients Occupant un lit'. It shows a table with columns: Patients, Search bar, Actions (Excel, +, C, F), Details, Identité, Prénom, Nom, Sexe, Date de Naissance, Actions. One row is selected, showing: 124578, othmeni, sawsen, Femme, 01/05/2002. Buttons for 'Consulter', 'Détails', and 'Rapport' are visible.

Figure 56: Interface « Patients en attentes »

- Interface « Patients occupants des lits »

The screenshot shows the 'Patients' section of the HEALTHTRAC application. The left sidebar shows 'Tableau de bord' and 'Patients'. The main area is titled 'Liste des patients > Les patients Occupant un lit'. It shows a table with columns: Patients, Search bar, Actions (Excel, +, C, F), Details, Identité, Prénom, Nom, Sexe, Date de Naissance, Actions. One row is selected, showing: 124578, othmeni, sawsen, Femme, 01/05/2002. Buttons for 'Consulter', 'Détails', and 'Rapport' are visible.

Figure 57:Interface « Patients occupants des lits »

- **Interface « Patients archivés »**

MAIN

Tableau de bord

Patients

Archives

Liste des patients > Archives

Patients Search... XLSX + C =

	Détails	Identité	Prénom	Nom	Sexe	Date de sortie	Actions
<input type="checkbox"/>		124578	othmeni	sawsen	Femme	02/06/2024	

Items per page: 10 1 - 1 of 1

Figure 58: Interface « Patients archivés »

- **Interface « Détails patient »**

SÉJOURS TRANSFÉRE PROLONGATION LIBÉRATION

sawsen othmeni

pas présent

Date de Naissance : 01/05/2002

Sexe : Femme

Poids : 80 Kg

Hauteur : 150 cm

Email : sawsen@gmail.com

Téléphone : +126 124563

Pays : France

Séjour prévu

Date d'entrée prévisionnelle	Date de sortie prévisionnelle	Unité de soins	Chambre	Date d'entrée réelle	Date de sortie réelle
May 31, 2024, 19:41	June 3, 2024, 19:41	unité 1	1	-----	-----

Transféré

Date de transférée	Ancienne unité de soins	Nouvelle unité de soins	Ancienne chambre	Nouvelle chambre	Ancien lit	Nouveau lit
May 14, 2024, 11:23	unité 1	unité 2	2	6	21	10

Figure 59: Interface « Détails patient »

- **Interface « Réserver patient pour Secrétaire »**

Figure 60: Interface « Réserver patient »

Conclusion

Le quatrième sprint de notre projet a marqué une étape cruciale dans l'évolution de notre système de gestion hospitalière. En se concentrant sur le processus d'entrée et de sortie des patients, la réservation des lits, les déplacements et prolongations, ainsi que la gestion des archives, chaque aspect a été soigneusement conçu pour offrir une gestion transparente et efficace des patients. L'intégration de la technologie WebSocket pour une communication en temps réel et la mise en place d'un système de gestion des notifications ont renforcé notre capacité à fournir des soins de santé de qualité et une expérience utilisateur exceptionnelle. En résumé, ce sprint a représenté une avancée majeure vers notre objectif global.

Conclusion générale

Ce projet marque une transformation majeure dans la gestion des établissements hospitaliers, combinant innovation, efficacité et facilité d'utilisation pour les administrateurs, secrétaires et docteurs. Grâce à une interface conviviale et des fonctionnalités robustes, notre système améliore l'efficacité, la coordination et la qualité des soins prodigués aux patients.

Pour les administrateurs, le projet fournit une plateforme robuste pour gérer les unités de soins, les chambres et les lits. Ils peuvent facilement ajouter de nouvelles unités, assigner des chambres et suivre le nettoyage des lits, assurant ainsi un environnement hospitalier impeccable et bien organisé.

Pour les secrétaires, ce système offre une vue d'ensemble claire et détaillée des patients. Ils peuvent consulter la liste des patients en fonction de leur statut – réservés, examinés mais non réservés, ou non examinés. La réservation de lits est simplifiée, permettant de saisir rapidement les dates de séjour et de choisir le type de chambre, avec une fonctionnalité supplémentaire pour gérer les accompagnants en cas de chambre double. En outre, ils peuvent consulter les rapports médicaux, gérer les détails du séjour, y compris les déplacements et les prolongations, offrant ainsi un suivi personnalisé et attentif à chaque patient.

Pour les docteurs, le projet propose un outil puissant pour gérer leurs patients efficacement. Ils peuvent accéder aux détails des patients, rédiger des rapports médicaux détaillés et les consulter les anciens rapports, assurant ainsi une continuité et une qualité de soins exemplaires. La synchronisation des rapports avec la liste des patients examinés mais non réservés améliore la communication et facilite la prise de décisions.

En plus de ces fonctionnalités, le système permet la gestion des rendez-vous, l'archivage des patients, et la consultation de rapports détaillés.

En conclusion, ce projet révolutionne la gestion hospitalière en rendant chaque processus plus fluide, plus efficace et plus orienté vers le patient. Il représente l'avenir de l'administration hospitalière, où chaque détail est pris en compte, chaque besoin anticipé, et chaque interaction optimisée pour offrir un service d'excellence. Grâce à cette solution, nous nous engageons à améliorer la qualité des soins, réduire les charges administratives et créer un environnement où chaque professionnel de santé peut se concentrer sur l'essentiel : le bien-être des patients.

Webographie

- [1] Visual Studio Code [En ligne] :<https://blog.webnet.fr/visual-studio-code/> [Consulté le 10/03/2024].
- [2] MySql [En ligne] :<https://kinsta.com/fr/base-de-connaissances/qu-est-ce-que-mysql/> [Consulté le 10/03/2024].
- [3] Postman [En ligne] :<https://www.eewee.fr/postman-cest-quoi/> [Consulté le 10/03/2024].
- [4] SpringToolSuite4 [En ligne] :<https://spring.io/tools/sts> [Consulté le 10/03/2024].
- [5] StarUML [En ligne] :<https://docs.staruml.io> [Consulté le 10/03/2024].
- [6] XAMP [En ligne] :<https://www.apachefriends.org/fr/index.html> [Consulté le 10/03/2024].
- [7] Java [En ligne] :https://www.java.com/fr/download/help/whatis_java.html [Consulté le 10/03/2024].
- [8] TypeScript [En ligne] :<https://www.typescriptlang.org/> [Consulté le 10/03/2024].
- [9] Angular [En ligne] :<https://angular.io/> [Consulté le 10/03/2024].
- [10] Springboot [En ligne] :<https://spring.io/projects/spring-boot> [Consulté le 10/03/2024].
- [11] JWT [En ligne]:<https://blog.ippon.fr/2017/10/12/preuve-d-authentification-avec-jwt/> [Consulté le 10/03/2024].