

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane Mira De Bejaïa
Faculté de Sciences Exactes
Département d'Informatique



Mémoire de fin de cycle Master professionnel en informatique
Option : Génie Logiciel

T H E M E

**CONCEPTION ET REALISATION D'UN SYSTEME DE GESTION DE
RENDEZ-VOUS ET DE FILES D'ATTENTE POUR LES HOPITAUX
CAS D'ETUDE EPH AMIZOUR**

Réalisé par :

M. MESSAD Mouloud

Soutenu devant le jury

Président	M. ATMANI Mouloud	Univ. A/Mira Bejaia
Examineur	M. ACHROUFENE Achour	Univ. A/Mira Bejaia
Encadreur	M. ALLEM Khaled	Univ. A/Mira Bejaia

ANNEE UNIVERSITAIRE
2022-2023

Dédicace

A l'heure de clôturer cette étape importante de mon cheminement, je pense à tous ceux qui de près ou de loin m'ont aidé à mener à bien ce mémoire.

Je tiens à dédier cet humble travail à ma très chère mère, aucun mot du monde ne saurait exprimer mon amour pour toi et ma gratitude, pour tes sacrifices, ton soutien et ta gentillesse sans égale, je ne te remercierai jamais assez, que dieu te garde pour moi, tu as foi en moi et je ferai tout mon possible pour ne pas te décevoir.

A toi mon très cher père, toi qui as donné corps et âme pour subvenir à nos besoins, avec le seul espoir de me voir réussir, je ne te remercierai jamais assez. Je te dois tout l'honneur, que dieu te garde pour moi. J'espère que tu seras fier de moi.

A vous mes très chères sœurs, merci de me supporter dans tous les moments difficiles.

A tous mes chers amis.

Remerciement

Je remercie Allah pour m'avoir octroyé la force, et les moyens nécessaires pour mener à bien mon parcours académique, à l'issu de plusieurs années de travail, le moment est venu pour remercier également toutes les personnes ayant contribué de prêt ou de loin pour venir à bout de ma formation :

- Merci à tout le personnel du département d'informatique de Bejaia ; enseignants et secrétaires pour les services qu'ils nous ont rendus durant ces 5ans.
- Merci à M. ALLEM Khaled pour ses conseils, et son enthousiasme à suivre ma recherche du début jusqu'à la fin. Ses conseils m'ont énormément éclairé. Merci de m'avoir fait part de votre patience et expertise.
- Merci à M. ARIOUAT et le personnel de D-Soft pour leur collaboration et les informations qu'ils m'ont fourni.
- Merci aux membres de jury qui ont accepté d'examiner et d'évaluer mon travail
- Merci au personnel de l'EPH AMIZOUR pour leurs collaborations, aides et informations.

Chacun a sa manière a contribué à la réalisation de ce modeste travail, encore merci à vous tous.

Table des matières

Dédicace	i
Remerciement.....	ii
Table des matières	iii
Liste des figures :.....	vi
Liste des tableaux :	viii
Liste des abréviations :	ix
Introduction générale.....	1
Chapitre I : Etude préalable	4
I.1. Introduction	5
I.2. Présentation du domaine métier	5
I.2.1 Définition des concepts clés	5
I.2.2 Solutions existantes	8
I.3. Présentation de l'établissement d'accueil.....	10
I.4. Présentation du cas d'étude : l'EPH Amizour.....	11
I.4.1 Plateau Technique	11
I.4.2 Services d'hospitalisation.....	12
I.4.3 Etude de l'existant	13
I.5. Problématique.....	13
I.6. Objectifs du projet.....	14
I.7. Conclusion.....	15
Chapitre II : Méthodologie de conception et langages de programmation	16
II.1. Introduction	17
II.2. Processus de développement.....	17
II.2.1 Scrum	17
II.2.2 Processus Unifié.....	20
II.2.3 Pourquoi Scrum et UP.....	21
II.3. Langages de modélisation et patron de conception.....	22
II.3.1 Langages de modélisation	22
II.3.2 Patron de conception	23
II.4. Langages de programmation, bibliothèques et frameworks	23
II.4.1 Langages de programmation	23
II.4.2 Bibliothèques	24
II.4.3 Frameworks.....	25
II.5. Outils de conception et de développement.....	25
II.6. Conclusion.....	27

Chapitre III : Analyse Préliminaire	28
III.1. Introduction	29
III.2. Spécification des besoins.....	29
III.2.1 Identification des acteurs.....	29
III.2.2 Réalisation du diagramme de contexte.....	30
III.2.3 Identification des messages échangés	31
III.2.4 Capture des besoins fonctionnels	32
III.2.5 Capture des besoins non fonctionnels	35
III.3. Gestion du projet avec la méthode Scrum.....	35
III.3.1 Gestion de l'équipe Scrum	35
III.3.2 Backlog du produit	36
III.3.3 Découpage en sprint	38
III.3.4 Planification des Sprints.....	38
III.4. Conclusion.....	38
Chapitre IV : Sprint 1 “Gestion de rendez-vous”	39
IV.1 Introduction	40
IV.2 Backlog du premier Sprint	40
IV.3 Spécifications fonctionnelles.....	43
IV.3.1 Diagramme de cas d'utilisation.....	43
IV.3.2 Diagramme d'activité global du premier Sprint.....	44
IV.3.3 Description Textuelle des cas d'utilisation.....	45
IV.4 Analyse.....	47
IV.4.1 Modèle du domaine	47
IV.4.2 Diagrammes de Séquence Système	48
IV.5 Conception	52
IV.5.1 Diagrammes de séquence détaillés.....	53
IV.5.2 Diagramme de classes de conception du premier Sprint.....	57
IV.5.3 Modèle relationnel.....	59
IV.5.4 Dictionnaire de données	59
IV.6 Développement.....	59
IV.6.1 Schéma de la Base de Données	59
IV.7 Test.....	60
IV.7.1 Interfaces de l'application « MyDocBooking »	61
IV.8 Conclusion.....	63
Chapitre V : Sprint 2 ‘Gestion de files d’attente’	65
V.1 Introduction	66
V.2 Backlog du deuxième Sprint	66

V.3	Spécifications fonctionnelles.....	70
V.3.1	Diagramme de cas d'utilisation.....	71
V.3.2	Description textuelle des cas d'utilisation.....	71
V.4	Analyse.....	73
V.4.1	Modèle du domaine	73
V.4.2	Diagrammes de séquence système	73
V.5	Conception	77
V.5.1	Diagrammes de séquence détaillé	77
V.5.2	Diagramme de classes de conception.....	80
V.5.3	Modèle relationnel.....	81
V.5.4	Dictionnaire de données	81
V.6	Développement.....	81
V.6.1	Schéma de base de données.....	82
V.7	Test.....	82
V.7.1	Interfaces de l'application « MedFlow ».....	82
V.8	Conclusion.....	83
	Conclusion Générale	84
	Références bibliographiques	86
	-Annexe 01-.....	89
	-Annexe 02-.....	91

Liste des figures :

Figure 1 : Logo de D-Soft.....	10
Figure 2 : Une photo prise de l'extérieur de l'EPH Amizour.....	11
Figure 3 : Les différentes étapes de la méthode Scrum.....	19
Figure 4 : Le cycle de vie UP.....	20
Figure 5 : Diagramme de contexte dynamique.....	30
Figure 6 : Diagramme de cas d'utilisation global.....	34
Figure 7 : Diagramme de cas d'utilisation du premier Sprint.....	44
Figure 8 : Diagramme d'activité du premier Sprint.....	45
Figure 9 : Modèle du domaine du premier Sprint.....	48
Figure 10 : Diagramme de séquence du cas S'authentifier.....	49
Figure 11 : Diagramme de séquence système du cas S'inscrire.....	50
Figure 12 : Diagramme de séquence système du cas Prendre un rendez-vous.....	51
Figure 13 : Diagramme de séquence du cas confirmer un rendez-vous.....	52
Figure 14 : Diagramme de séquence détaillé du cas S'authentifier.....	54
Figure 15 : Diagramme de séquence détaillé du cas S'inscrire.....	55
Figure 16 : Diagramme de séquence détaillé udu cas Prendre un rendez-vous.....	56
Figure 17 : Diagramme de classes de conception du premier Sprint.....	58
Figure 18 : Schéma de base de données du premier Sprint en utilisant un profil UML pour les BDDR.....	60
Figure 19 : Page d'accueil de l'application 'MyDocBooking' sur différentes tailles d'écrans.....	61
Figure 20 : Page d'inscription et page de connexion de l'application 'MyDocBooking'.....	62
Figure 21 : Capture d'écran de la page connexion de l'application 'MyDocBooking'.....	62

Figure 22 : Page de prise de rendez-vous de l'application 'MyDocBooking' sur écran pc et mobile.....	63
Figure 23 : Capture d'écran sur mobile de la page prise de rendez-vous de l'application 'MyDocBooking'	63
Figure 24 : Diagramme de cas d'utilisation du deuxième Sprint.....	71
Figure 25 : Modèle du domaine deuxième Sprint.....	73
Figure 26 : Diagramme de séquence système du cas « S'authentifier ».....	74
Figure 27 : Diagramme de séquence système du cas « Créer compte personnel ».....	75
Figure 28 : Diagramme de séquence système du cas « Ajouter patient à une file ».....	76
Figure 29 : Diagramme de séquence système du cas « Appeler prochain patient ».....	77
Figure 30 : Diagramme de séquence détaillé du cas « Appeler prochain patient ».....	78
Figure 31 : Diagramme de classes de conception du deuxième Sprint.....	80
Figure 32 : Schéma de base de données du premier Sprint en utilisant un profil UML pour les BDDR.....	82
Figure 33 : Page d'accueil et menu principale de l'application 'MedFlow' pour médecins...83	83
Figure 34 : Menu principal de l'espace secrétaire.....	83

Liste des tableaux :

Tableau 1 : Tableau représentant un comparatif des fonctionnalités proposées par différentes applications.....	9
Tableau 2 : Tableau listant les différents services de l'EPH Amizour.....	12
Tableau 3 : Tableau listant le matériel informatique présent à l'EPH Amizour.....	13
Tableau 4 : Les éléments en communs entre UP et Scrum.....	22
Tableau 5 : Tableau représentant les messages échangés.....	32
Tableau 6 : Tableau représentant les cas d'utilisation.....	33
Tableau 7 : Tableau représentant le Product Backlog.....	37
Tableau 8 : planification des Sprints.....	38
Tableau 9 : Tableau représentant le Backlog du premier Sprint.....	42
Tableau 10 : Description textuelle du cas d'utilisation « S'authentifier ».....	46
Tableau 11 : Description textuelle du cas d'utilisation « Prendre un rendez-vous ».....	47
Tableau 12 : Description textuelle du cas d'utilisation « Confirmer un rendez-vous ».....	47
Tableau 13 : Tableau représentant le Backlog du deuxième Sprint.....	70
Tableau 14 : Description textuelle du cas d'utilisation « Appeler prochain patient ».....	72
Tableau 15 : Description textuelle du cas d'utilisation « Ajouter un patient à la file ».....	72

Liste des abréviations :

EPH	Etablissement public hospitalier
CHU	Centre hospitalier universitaire
RDV	Rendez-vous
UP	Unified Process
PB	Product Backlog
MDB	MyDocBooking
MF	MedFlow
VsCode	Visual Studio Code
JS	Javascript

Introduction générale

Introduction générale

L'avancement technologique a littéralement révolutionné le monde d'aujourd'hui affectant presque tous les domaines de la société moderne, des communications aux transports en passant par l'éducation et les services de soins. Dans la santé par exemple, les progrès technologiques et scientifiques ont sauvé des vies et ont permis de mieux connaître le corps humain, les maladies, les équipements et traitements.

La technologie de l'informatique a contribué fortement à l'amélioration du bien être humain et a entraîné de profonds changements dans l'organisation du travail, si on creuse un peu dans le passé, avant même l'invention de l'ordinateur, les informations étaient enregistrées manuellement sur des supports papier chose qui était loin d'être efficace : perte de temps énorme à leur recherche et parfois même leur dégradation ou perte. Depuis quelques années, la transformation digitale s'est imposée aux entreprises de tous les secteurs d'activités où l'information est désormais pilotée par le numérique, assurant ainsi de meilleurs rendements et dans des temps record.

Dans ce contexte, les hôpitaux, en tant que structures complexes font face à de nombreux défis à cause notamment de l'augmentation tant en nombre qu'en complexité des maladies et systématiquement du nombre de patients à traiter. D'autant plus, la prise en charge du patient doit parfois s'appuyer sur une succession d'actes de soins réalisés par divers professionnels de la santé.

L'accès aux informations relatives aux malades dans les différents services sanitaires devient primordial pour assurer leur bon fonctionnement. En effet, la numérisation de la gestion des informations des malades devrait faciliter l'organisation des flux de patients avec une prise en charge plus souple allant des prises de rendez-vous (rdv) jusqu'à l'accueil.

Le but de notre travail est double, il s'agit d'utiliser les technologies de l'information pour la réalisation d'une solution numérique destinée à la gestion des files de patients au sein des différents services hospitaliers et aussi de proposer une application web pour la gestion des rendez-vous dans l'objectif de faciliter et organiser l'accueil des malades dès leur arrivée.

Le présent document est devisé en cinq chapitres :

Un premier chapitre nommé « Etude préalable », dont l'objectif sera de présenter les concepts clé du domaine d'étude, l'organisme d'accueil 'D-Soft' et le cas d'étude 'Eph Amizour' ainsi que l'étude de l'existant.

Introduction générale

Le deuxième chapitre quant à lui se nomme « Méthodologie de conception et langages de programmation ». Dans ce chapitre nous allons présenter la méthodologie suivie tout au long du processus de développement en l'occurrence Scrum et UP combinées, puis nous allons parler des langages, bibliothèques et frameworks utilisés pour développer notre solution.

Le troisième chapitre intitulé « Analyse préliminaire » sera consacré à l'identification des acteurs, la spécification des besoins fonctionnels et non fonctionnels, découpage et planification des Sprints et ce dans le cadre de l'adoption de la méthode Scrum.

Dans le quatrième chapitre intitulé "Gestion des rendez-vous", nous aborderons le premier Sprint ayant pour objectif la gestion des rendez-vous. Ce dernier devra passer par les étapes d'UP, à savoir la spécification des besoins, l'analyse des besoins, la conception, l'implémentation et test.

Le dernier chapitre, intitulé "Gestion de files d'attente", sera consacré au deuxième Sprint, couvrant les quatre étapes d'UP.

Le manuscrit s'achève par une conclusion générale et des perspectives.

Chapitre I : Etude préalable

I.1. Introduction

« Faire la queue est une expérience négative qui peut créer un sentiment d'anxiété et de tensions chez les clients. L'entreprise doit trouver le moyen de gérer efficacement les files d'attente pour réduire cet impact négatif et maintenir la satisfaction des clients. » [4].

L'attente d'un examen médical pourrait affecter l'état du patient et son avis sur la qualité de service de l'établissement, comme il pourrait également aggraver son état de santé.

La gestion de flux de patients est un problème parmi tant d'autres auxquels les établissements hospitaliers font face quotidiennement. Il est donc nécessaire de trouver un moyen qui permettra de gérer ce flux efficacement au sein de l'établissement, car la fluidité du parcours du malade est un élément essentiel pour assurer la bonne organisation de l'établissement.

Dans ce présent chapitre nous allons brièvement parler du domaine d'étude, définir la gestion des files d'attentes et des rendez-vous, nous citerons quelques solutions matérielles et logicielles existantes, ensuite nous présenterons l'établissement public de la santé de la commune d'Amizour où l'étude a été faite et citerons ses différents services.

I.2. Présentation du domaine métier

La gestion des rendez-vous et des files d'attentes au sein des établissements de santé est cruciale pour garantir une bonne qualité de service et la satisfaction des patients. Le personnel de santé doit donc être en mesure de mettre en place une stratégie efficace pour gérer les files d'attentes et les rendez-vous.

I.2.1 Définition des concepts clés

Dans cette section, nous allons définir quelques concepts clés, notamment la gestion des files d'attente et la gestion des rendez-vous.

a. Gestion des files d'attente

La gestion des files d'attente est l'étude du mouvement des personnes, des tâches ou des objets dans une ligne suivant un certain ordre, en attente d'accéder à un service ou à une ressource.

Chapitre I : Etude préalable

Nous trouvons aujourd'hui les files d'attentes présentes dans tous les domaines, et souvent cette pratique a pour but de mettre en œuvre un processus efficace et optimale pour la gestion et régulation de flux dans un système donné.

Dans l'informatique notamment la gestion des files d'attente fait généralement référence aux algorithmes et aux structures de données permettant l'ordonnancement des objets ou des tâches contenues dans cette file et le contrôle d'accès aux ressources demandées.

L'ordre de passage dépend souvent de la stratégie suivie, par exemple : FiFo (First in First out), LiFo (Last in First out), SPT (Shortest Processing Time) ...

Dans le secteur médical les files sont composées de patients et les ressources demandées sont les médecins, et donc il faut prendre en compte les contraintes de disponibilités du médecin et de la salle de soin avant d'appeler le prochain patient de la file.

De [1] [2] et [3] nous pouvons catégoriser les files d'attente simple en quatre modèles principaux :

- **Le Modèle Files d'attente à serveur unique M/M/1** : le premier M fait référence à Markovian¹, il indique que la loi des arrivés suit le modèle en Poisson², le deuxième M indique que les temps inter-arrivée sont indépendants les uns des autres et sont sans mémoire [2], tant dis que le 1 indique le nombre de serveurs, en l'occurrence ici c'est un seul serveur. Ce modèle est un modèle simple et non-trivial, et c'est le plus utilisé en théorie de file d'attente.
- **Le Modèle Files d'attente multiserveur M/M/C** : est un autre modèle de file d'attente où 'C' fait référence au nombre de serveurs, ce modèle décrit le comportement d'une file où les clients arrivent en Poisson et peuvent être servis par plusieurs serveurs. Si tous les serveurs sont occupés le client doit attendre dès qu'un serveur se libère il pourra alors traiter celui-ci.
- **Le Modèle Files d'attente avec troncation M/M/c/K** : est un modèle de file d'attente multiserveurs basé sur celui cité précédemment, à la différence du modèle de

¹ De Andreï Andreïevitch Markov, c'est un modèle mathématique utilisé pour décrire une séquence de variables aléatoires où la prédiction des événements futurs dépend uniquement de l'état actuel non pas des événements passés [1].

² Un processus de Poisson, nommé d'après Siméon Denis Poisson, c'est un modèle mathématique modélisant des événements aléatoires qui se reproduisent au cours du temps c'est le plus simple et le plus utilisé des processus modélisant une file d'attente.

multiserveurs classique, le modèle avec troncature a une limite supérieure K qui représente nombre maximum de clients pouvant être présents dans la file d'attente.

- **La formule de perte d'Erlang M/M/c/c** : définit comme suit :

$$P_n = \frac{\frac{c^n}{n!}}{\sum_{i=0}^{c-1} \frac{c^i}{i!}}, \text{ avec 'Pn' qui représente la probabilité de perte de clients (perte de client}$$

c'est le fait de ne pas pouvoir servir un client immédiatement après son arrivé), 'n' qui représente le nombre de clients en attente et 'c' le nombre total de serveurs disponibles.

Ce modèle est un cas particulier du modèle avec troncature où $K=c$, une formule mathématique du mathématicien danois Agner Krarup Erlang qui sert à calculer les probabilités de blocage dans un système composée de plusieurs serveurs identiques et une capacité limitée.

- **Le modèle de file d'attente à service illimité M/M/∞** : L'arrivée des clients est basé sur le processus de Poisson, étant donné qu'il y a un nombre infini de serveurs disponibles les clients seront immédiatement servis par un serveur disponible.

b. Gestion des rendez-vous

Processus permettant de programmer un évènement ou une rencontre entre deux parties ou plus, les rendez-vous permettent une meilleure planification et facilitent la communication et la collaboration. Ils sont utilisés dans différents domaines (professionnels, administratifs, sociaux...) nous pouvons citer par exemple un rendez-vous pour un entretien d'embauche ou bien un rendez-vous avec un avocat pour une affaire juridique. Les rendez-vous sont également utilisés dans le domaine médical notamment pour planifier une consultation chez un médecin, différentes méthodes sont utilisées allant de la saisie manuelle ; inscription des informations des patients par un agent jusqu'au applications automatisées, qui permettent de gérer les emplois du temps des professionnels et de s'assurer que les patients disposent d'un créneau horaire leur convenant, tout en prenant en compte les disponibilités des médecins concernés et éviter les heures d'attentes.

Cependant, il faut aussi tenir compte des cas d'urgences qui ne peuvent se permettre d'attendre et encore moins de les prédire et qui nécessitent une intervention dans l'immédiat, c'est pourquoi il faut offrir une certaine souplesse et disponibilité du service.

I.2.2 Solutions existantes

Nous avons effectué une recherche comparative des différentes applications disponibles sur le marché, et nous allons les comparer en fonction des caractéristiques et des fonctionnalités qu'elles proposent dans le tableau ci-dessous :

Chapitre I : Etude préalable

Outils	Caractéristiques										
	Domaine	Gratuit	Plateforme	Développé par	Open source	Installable	Année	Fonctionnalités			
								Gest. Des Patients	Rendez-vous en ligne	Gest. De salles d'attente	Gest. Des visites
MedRec	Médical	Non	Web	MedRec_Studio	Non	Non	2021	Oui	Non	Oui	Oui
MedERP	Médical	Non	Windows Linux Mac	MedERP	Non	Oui	2021	Oui	Non	Non	Non
CLICMEDIC	Médical	Non	Web	LogiSam Solutions	Non	Non	2020	Oui	Non	Non	Oui
Santymed	Medical	Non	Windows	Santymed	Non	Oui	2017	Oui	Non	Oui	Oui
Xmedic	Medical	Non	Windows	Kellouche	Non	Oui	2016	Oui	Non	Oui	Oui
NourDoc	Medical	Oui	Windows	NourDoc	Non	Oui	2013	Non	Non	Non	Oui
CABINET-MEDICAL	Medical	Oui	Windows	Fousseni TRAORE	Non	Oui	2016	Oui	Non	Non	Oui
Medintux	Medical	Oui	Windows Linux	VBonhomme	Oui	Oui	2011	Oui	Non	Non	Non
FreemedForms	Medical	Oui	Windows Linux Mac	B.Gérald	Oui	Oui	2011	Oui	Non	Non	Oui
Agenda papier	/	Oui	/	/	/	/	/	Ajout uniquement	Non	Non	Ajout uniquement

Tableau 1 : tableau représentant un comparatif des fonctionnalités proposées par différentes applications.

Ces applications contiennent d'autres fonctionnalités qui n'ont pas été listées dans ce tableau car elles sont jugées en dehors du thème de notre projet.

Après avoir étudié ces applications, on remarque qu'elles offrent des fonctionnalités similaires, telles que la gestion des patients, des salles d'attente, des visites, etc. Dans le but d'apporter une valeur ajoutée, nous avons envisagé de développer une application web permettant la prise de rendez-vous en ligne en complément de notre application de gestion des files d'attente.

I.3. Présentation de l'établissement d'accueil

D-Soft est une société de logiciels créée en 2011, située à Bejaïa qui s'est spécialisée dans la création de solutions logicielles sur mesure pour les entreprises de toutes tailles et qui se concentre sur le développement de divers produits et services (des applications pour restaurants, pour la gestion de stock...) pour stimuler et simplifier l'utilisation de documents numériques. Composée du gérant Mr Hassane Ariouat et son associé, qui cumulent plusieurs années d'expérience dans le domaine du génie logiciel qui ont fait que l'entreprise a toujours réussi à aller de l'avant et satisfaire les différents besoins de ses clients.

D-Soft propose une gamme complète de services en relation avec le domaine de développement logiciel y compris l'analyse, la conception, la réalisation et la maintenance des logiciels, tout en utilisant les langages et les technologies les plus populaires.



Figure 1 : Logo de D-Soft.

I.4. Présentation du cas d'étude : l'EPH Amizour

L'établissement public hospitalier d'Amizour ou EPH Amizour, baptisé sous le nom de "Hôpital BENMERAD EL MEKKI" en 1993, a été créé pour répondre à la mise en place du nouveau système de santé en 2007, l'EPH a été construit fin de l'année 1991, et son ouverture officielle a été en 1992 [27].

Il est situé à 24 Kilomètres au sud de la principale localité de la wilaya de Bejaia, Il peut atteindre une capacité de 200 lits techniques, et la population qu'il peut couvrir est d'environ 160 000 personnes issues des huit communes de l'ancien secteur sanitaire d'Amizour et des communes correspondantes des wilayas voisines (Sétif, Bouira).



Figure 2 : Photo prise de l'extérieur de l'EPH Amizour.

L'hôpital est composé d'un plateau technique et de huit services d'hospitalisation respectivement détaillés ci-après :

I.4.1 Plateau Technique

Le plateau technique de l'hôpital regroupe l'ensemble des équipements et infrastructures nécessaires pour la prise en charge des patients. D'après [27] le plateau est composé de :

- Un Pavillon des urgences médico-chirurgicales (20 lits organisés).
- Un Bloc Opérateur (04 salles Opérateurs).
- Réanimation chirurgicale : 05 lits.
- Deux salles de Radiologie.
- Un Laboratoire d'analyse médicale.
- D'une Pharmacie centrale.
- Une Banque de sang.
- Un Bureau des entrées.

I.4.2 Services d'hospitalisation

L'hôpital dispose principalement de 11 services cités dans le tableau suivant :

Service	Nbre de lits	Unité
Chirurgie Générale	60	<ul style="list-style-type: none"> ▪ Hospitalisation Hommes ▪ Hospitalisation Femmes
Gynécologie Obstétrique	30	<ul style="list-style-type: none"> ▪ Gynécologie ▪ Obstétrique
Médecine interne	56	<ul style="list-style-type: none"> ▪ Hospitalisation Hommes ▪ Hospitalisation Femmes
Oncologie Médicale	36	<ul style="list-style-type: none"> ▪ Hospitalisation ▪ Consultations
Pédiatrie	22	<ul style="list-style-type: none"> ▪ Pédiatrie Néonatalogie
Urgences Médicochirurgicales	20	<ul style="list-style-type: none"> ▪ Accueil, Tris et mise en observation ▪ Réanimation
Épidémiologie	/	/
Médecine de travail	/	/
Radiologie centrale	/	<ul style="list-style-type: none"> ▪ Radiologie ▪ Échographie
Laboratoire centrale	/	<ul style="list-style-type: none"> ▪ Microbiologie ▪ Biochimie
Pharmacie	/	<ul style="list-style-type: none"> ▪ Gestion des produits pharmaceutiques ▪ Distribution des produits pharmaceutiques

Tableau 2 : Tableau listant les différents services de l'EPH Amizour [5].

I.4.3 Etude de l'existant

Malgré le personnel qualifié et les ressources matérielles disponibles au sein de l'EPH Amizour (voir tableau [2]), le processus de prise en charge des patients est à ce jour géré manuellement en utilisant les méthodes classiques notamment l'agenda papier.

Les patients se présentent au service d'accueil où une secrétaire/infirmier prend note de leurs coordonnées et les oriente vers le service approprié. Les patients sont ensuite placés sur une liste d'attente et sont appelés en fonction de l'ordre d'arrivée.

Ci-dessous le matériel informatique disponible au sein de l'EPH Amizour.

Nom de l'équipement	Type/configuration			Date d'achat	Localisation actuelle	Nombre d'unité
Ordinateur Bureau	Processeur	RAM	Système d'exploitation	2013	<ul style="list-style-type: none"> - Bureau des entrées. - Administration. - Services d'hospitalisation. 	20
	Intel Atom 2008	1Giga octet	Win7			
Imprimante	Imprimante Matricielle			/	<ul style="list-style-type: none"> - Bureau des entrées. - Laboratoire. 	5
	Imprimante à jet d'encre			2017	- Administration.	3
Scanner	Scanner à plat			/	- Administration.	2
Ecran	Écrans LCD			/	<ul style="list-style-type: none"> - Hall d'entrée. - Service des Urgences. 	4

Tableau 3 : Tableau listant le matériel informatique présent à l'EPH Amizour.

L'EPH Amizour a précédemment reçu une proposition pour la mise en place d'un système de gestion de flux de patients, mais celle-ci ne s'est pas concrétisée à cause notamment de sa rigidité. Cette dernière exige la mise à disposition d'un grand nombre de médecins pour assurer le bon fonctionnement du système, alors que l'EPH ne dispose pas de l'effectif nécessaire pour adopter cette solution.

I.5. Problématique

« Il y a des moments où je suis indigné. Ma dignité, et celle des citoyens, a été touchée par ce que j'ai eu à voir lors de mes visites d'inspection » [6] a déclaré l'ancien ministre de la santé Mr Abderrahmane Benbouzid, suite à une visite sur le terrain en juin 2018 au CHU Ibn

Roch d'Annaba et après avoir constaté les faits accablants concernant l'état des hôpitaux et les conditions d'accueil des patients dans l'un des hôpitaux en Algérie.

Suite à la période d'étude des besoins et de collecte d'information passée au sein de l'EPH Amizour, nous avons en effet constatés quelques points négatifs qui affectent le bon fonctionnement de l'établissement, tel que :

- L'attente très longue du patient : Le prolongement des temps d'attente peut aggraver l'état de santé du patient, dégrade l'image de l'établissement ainsi que la qualité des services proposés vis-à-vis des citoyens.
- La saturation des services : L'absence d'un système de triage efficace crée une surcharge des services notamment le service des urgences et ce à cause des patients qui parfois ne présentent aucun caractère urgent ou qui par manque d'information doivent consulter dans un tout autre service.
- Les rendez-vous mal gérés : La mauvaise gestion des rendez-vous a tendance à prolonger les temps d'attentes, à créer des chevauchements horaires affectants ainsi la santé du patient et l'efficacité de l'établissement.
- La réalisation de calculs statistiques sur des supports papier présente des limitations : Nombreux sont les inconvénients présentés par cette méthode, en effet, effectuer des calculs statistiques sur des supports papier reste très limitée comparée au grand nombre de données à traiter (nombre de patients se présentant à chaque service, durée moyenne d'une consultation, nombre de consultations effectuées par chaque médecin...), chose qui rend le processus relativement lent et sous risques d'erreurs.

I.6. Objectifs du projet

En vue des points cités en haut et afin de pallier aux nombreux problèmes auxquels font face nos établissements nous proposons une application nommée 'MedFlow' destinée aux services d'accueils dans les hôpitaux et établissements de santé, cette application permettra de nous passer des moyens traditionnels de faire les choses et automatiser la gestion de flux de patients dans l'établissement.

Elle permettra de configurer les files d'attente pour chaque service et attribuer une certaine priorité pour chaque file, afin d'éviter les blocages qui peuvent paralyser le service entier et garder un mouvement constant au sein du service tout en respectant les priorités des

files, nous proposons également une application web nommée 'MyDocBooking' qui permettra aux patients de prendre un rendez-vous en ligne très facilement.

Le but n'étant pas de faire une solution rigide pour un seul établissement mais plutôt une application paramétrable, configurable et personnalisable, et qui peut s'adapter facilement aux besoins d'un établissement peu importe sa taille et sa complexité.

I.7. Conclusion

Pour parachever ce chapitre, étude préalable, qui nous a permis de mieux comprendre notre projet, lister quelques solutions existantes pour ensuite nous pencher plus sur notre cas d'étude en l'occurrence l'établissement public de la santé de la commune d'Amizour où nous avons eu à citer ses différents services puis définir notre champ d'intervention. Nous pouvons enfin poursuivre la réalisation de notre projet et ce en adoptant un processus de développement et un langage de modélisation, ce qui fera l'objet du chapitre suivant.

Chapitre II : Méthodologie de conception et langages de programmation

II.1. Introduction

A première vue la programmation paraît une activité très simple, et pourtant ce n'est pas le cas, les programmes sont longs, il arrive souvent de rencontrer des bugs et des erreurs qui coûtent très cher en temps et en argent. Le code difficilement lisible par d'autres développeurs que celui qui l'a écrit peut aussi poser problème. Car souvent dans les projets de moyenne à grande taille une équipe de développement est composée de plusieurs personnes et parfois chacun et son domaine d'expertise, développeur, designer, testeur...etc. Il est donc primordial de garder une bonne gestion de l'équipe pour assurer le bon déroulement du développement et garantir un logiciel de qualité, d'où l'apparition des méthodologies de développement.

Dans ce chapitre nous allons présenter SCRUM et UP (Unified Process) qui sont les deux méthodes adoptées dans ce projet, et nous allons également parler des langages de modélisation et de programmation.

II.2. Processus de développement

Le développement d'un nouveau produit est sans aucun doute un véritable défi, c'est un processus long en allant du concept initial au lancement final du produit. Il permet à travers un certain nombre d'étapes de planifier, organiser et gérer les activités de développement.

Le processus suivi diffère d'un projet à un autre et d'une équipe à une autre et dépend également de la méthodologie de travail choisie, en effet ces méthodes aident à gérer les risques, améliorer et faciliter la collaboration et encourager l'innovation. Elles fournissent un ensemble de règles et de principes à suivre afin de réaliser un logiciel dans les règles de l'art.

Dans notre projet nous avons opté pour la méthode agile Scrum combinée au processus unifié UP.

II.2.1 Scrum

« Scrum est un cadre léger qui aide les personnes, les équipes et les organisations à générer de la valeur grâce à des solutions adaptatives pour des problèmes complexes. » [10].

Le terme Scrum tire son sens de l'anglais ce qui signifie 'mêlée', s'inspire du rugby, un sport qui requiert un grand travail d'équipe pour espérer gagner une partie. Scrum est aujourd'hui de loin la méthode agile la plus utilisée, surtout dans le domaine du développement

Chapitre II : Méthodologie de conception et langages de programmation

logiciel, Scrum est basée sur le processus itératif³ et incrémental⁴, le principal avantage qu'elle présente par rapport aux autres méthodes c'est de permettre au projet de prendre forme rapidement et réaliser la première version testable par le client (première itération). Selon [18, 19], il existe dans Scrum trois acteurs principaux :

- **Scrum Master** : C'est la personne qui se charge de garantir et de faciliter la mise en œuvre de la méthode et de piloter les différentes phases du Sprint⁵. Il encadre les membres de l'équipe et veille à ce que chaque membre a une vision claire sur ses tâches.
- **Product Owner** : C'est le responsable de la gestion efficace du PB (Product Backlog), il se charge de définir les priorités dans le Backlog et s'assurer que le PB est transparent, visible et compris par tout le monde. Le Product Owner c'est la personne qui représente le client au sein de l'équipe de développement.
- **L'équipe de développement** : Sont les membres de l'équipe qui s'engagent à créer et d'implémenter les Incréments dans chaque Sprint.

II.2.1.1 Artefacts Scrum

Les artefacts⁶ sont les éléments permettant de faire fonctionner la méthode Scrum, de suivre l'avancement d'un projet et de garantir sa transparence, et ils sont trois :

- **Le Product Backlog** : Une liste de PBIs⁷ (Product Backlog Items) qui doivent être ordonnés selon leur importance, leur urgence ou leur impact sur le logiciel, les développeurs vont donc prioriser les PBIs situés en haut de liste, ce qui permet de lui donner un fonctionnement similaire à celui d'une pile.
- **Le Sprint Backlog** : Une liste contenant les spécifications techniques pour les tâches qui doivent être accomplis par l'équipe de développement durant le Sprint en cours.
- **L'incrément Produit** : C'est l'ensemble des items du Backlog qui ont été accomplis pendant le Sprint en cours, il doit être testé et validé et prêt à être livré. Plusieurs incréments peuvent être regrouper pour créer une version plus complète du produit ce qui constitue donc un release.

³ Processus dans lequel une séquence d'instruction est répétée de manière cyclique selon le besoin.

⁴ Processus qui implique des intégrations continue d'un système pour produire des exécutable, chaque nouvelle version contenant des améliorations incrémentales.

⁵ Ce sont des intervalles bornés dans le temps ou des timesboxes durant lesquels les développeurs travaillent sur les tâches mentionnées dans le Backlog.

⁶ Les artefacts Scrum sont des éléments utilisés pour assurer la transparence, la communication et la collaboration entre l'équipe de développement Scrum.

⁷ Ce sont les besoins et les fonctionnalités attendu par l'application finale et ils sont généralement établie par l'utilisateur lui-même.

II.2.1.2 Cycle de projet Scrum

La figure ci-dessous représente et décrit en détails le processus à suivre lors de la réalisation d'un projet Scrum.

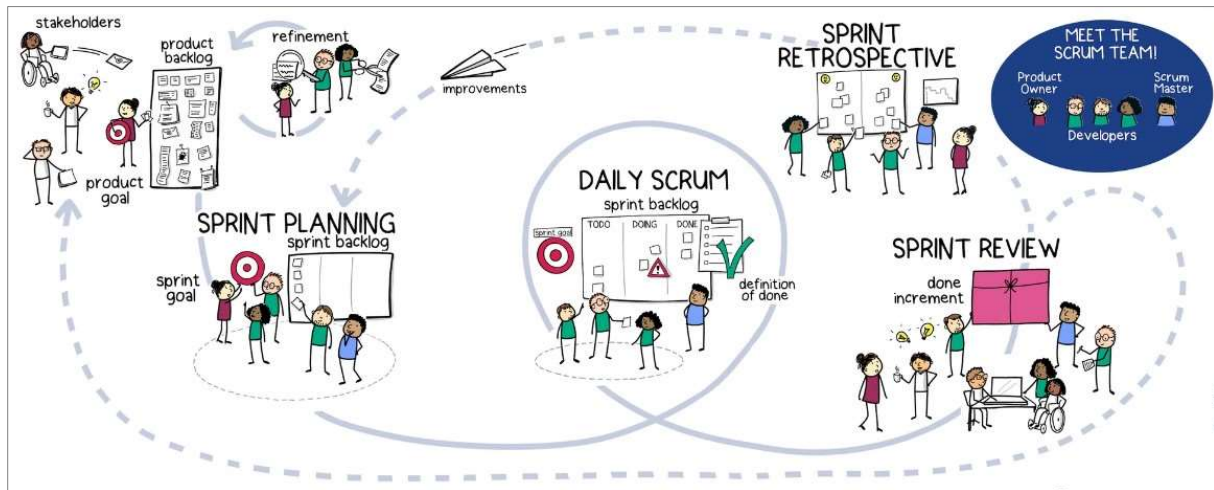


Figure 3 : Les différentes étapes de la méthode Scrum [7].

Comme illustré dans la figure ci-dessus, Scrum est basée sur quatre étapes fondamentales pour aboutir à un résultat satisfaisant :

- a. **Sprint planning** : C'est pendant cette période du Sprint que se fait la planification, les PBIs vont être décomposés en tâches, et chaque tâche sera assignée à une personne ou une équipe et aura une durée bien précise.
- b. **Sprint Exécution** : L'exécution du Sprint est l'étape qui vient après la planification, durant celle-ci l'équipe de développement va travailler sur les fonctionnalités qui seront prochainement livrées, durant cette étape chaque jour y'a un daily scrum⁸ qui prend place pendant lequel les membres discutent de l'état d'avancement des tâches qui leur sont assignées.
- c. **Sprint Review** : A la fin de la Sprint Exécution, on aura comme résultat un incrément, et durant le Sprint review une réunion prend place où l'incrément développé est présentée à l'utilisateur et pour recueillir son avis sur la fonctionnalité soit l'incrément est validé soit il demande de faire des modifications sur celui-ci.

⁸ Un Stand up meeting de quelques minutes avec la participation de tous les développeurs travaillant sur le projet.

Chapitre II : Méthodologie de conception et langages de programmation

- d. Sprint Rétrospective :** Des réunions entre les membres de l'équipe de développement pour discuter sur les éléments de la méthodes Scrum qui ont fonctionnés et sur les points à améliorer pour le prochain Sprint.

II.2.2 Processus Unifié

« Le processus unifié est un processus de génie logiciel. Il fournit une approche disciplinée pour l'attribution des tâches et des responsabilités au sein d'une organisation de développement. Son objectif est d'assurer la production de logiciels de haute qualité qui répondent aux besoins de ses utilisateurs finaux dans le cadre d'un calendrier et d'un budget prévisibles. » [11].

II.2.2.1 Cycle de vie UP

Le processus unifié est itératif et incrémental, il est conçu pour être flexible et adaptatif aux besoins du projet.

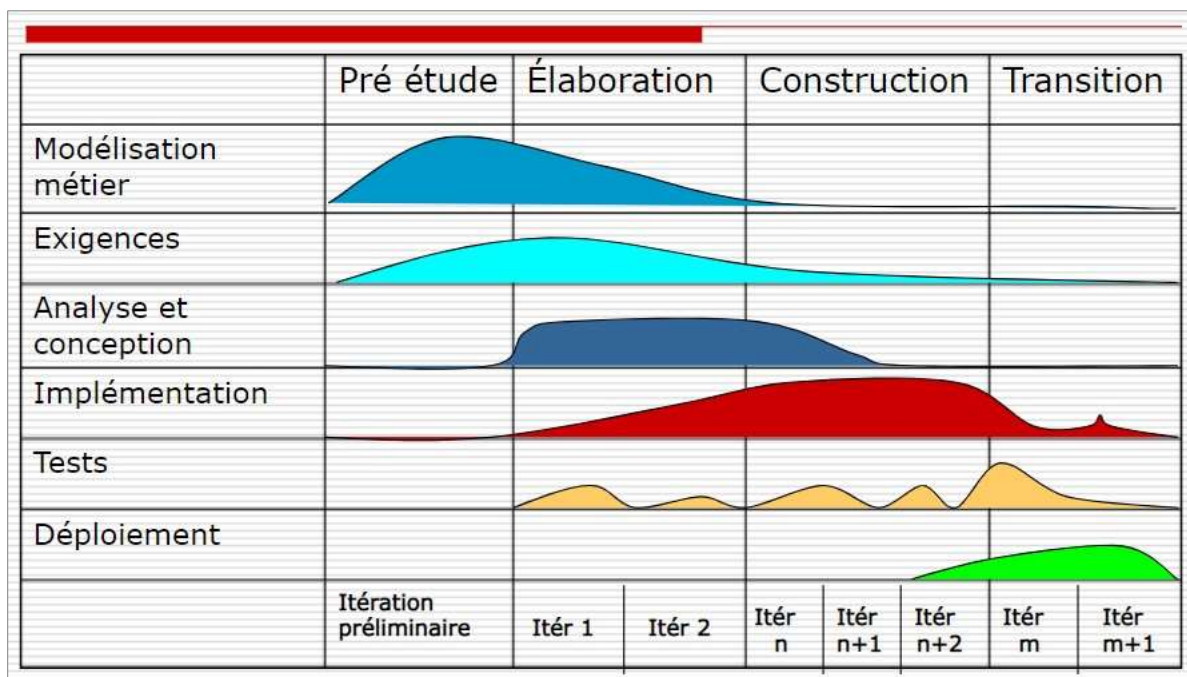


Figure 4 : Le cycle de vie UP [14].

Le processus unifié est composé de quatre phases principales :

- **Pré étude (Inception) :** cette étape vise à définir les besoins des utilisateurs, les objectifs du projet et les estimations du projet.

Chapitre II : Méthodologie de conception et langages de programmation

- **Elaboration** : cette phase consiste à définir en détail les fonctionnalités du système, établir une architecture solide et un plan du projet.
- **Construction** : développer une application complète prête à être livrée, et de manière itérative et incrémentale.
- **Transition** : Réajuster le logiciel en corrigeant les éventuels bugs et le déployer pour les utilisateurs.

II.2.2.2 Caractéristique du processus unifié

Le processus unifié est caractérisé principalement par :

- **Le pilotage par les cas d'utilisation** : le développement est dirigé par les besoins des utilisateurs finaux ce qui permet aux développeurs de se concentrer sur les fonctionnalités les plus importantes.
- **La démarche centrée sur l'architecture** : s'assurer que les différentes parties du système sont conçues pour fonctionner ensemble de manière efficace et cohérente.
- **Gestion des risques** : il est important d'identifier les risques tout au long du projet. Les risques potentiels sont identifiés et évalués dès la première phase.
- **Gestion des besoins** : UP permet la gestion des besoins qui peuvent évoluer au fil du temps de manière contrôlée en évaluant leurs impacts.
- **L'approche basée sur les modèles** : consiste à utiliser des modèles pour décrire le système à développer notamment des diagramme UML.
- **L'approche itérative et incrémentale** : Cette approche consiste à diviser le projet en plusieurs itérations, chacune produisant un livrable partiel et fonctionnel.

II.2.3 Pourquoi Scrum et UP

Scrum et UP (processus unifié) sont deux méthodologies largement utilisées dans l'industrie du développement de logiciels. Malgré les différences d'approche des deux méthodologies, il est tout à fait possible de combiner Scrum et UP, mais cela nécessite une planification et une certaine coordination pour exploiter le potentiel des deux méthodes de manière efficace. D'un côté on a Scrum, une méthodologie Agile qui se concentre sur le développement itératif et incrémental, tandis que de l'autre côté on a UP qui est une approche plus structurée qui met l'accent sur un ensemble de phases et d'artefacts bien définis.

On trouve dans le tableau ci-dessous les points en communs des deux méthodes :

Chapitre II : Méthodologie de conception et langages de programmation

Scrum	UP
Backlog produit	Liste des exigences fonctionnelles ou non
Backlog de Sprint	Plan d'itération
Produit partiel utilisable	« Prototype » en début de projet (phase d'Elaboration par exemple), « Produit » ensuite
Revue de Sprint	Réunion de fin d'itération.

Tableau 4 : Les éléments en communs entre UP et Scrum [9].

Scrum et UP ont toutes deux leurs avantages et peuvent être utilisées selon les besoins spécifiques d'un projet, la clé est de trouver un équilibre entre les deux. Une manière de combiner Scrum et UP est d'utiliser Scrum pour structurer les phases de développement et UP pour les phases de planification et de conception.

II.3. Langages de modélisation et patron de conception

II.3.1 Langages de modélisation

Un langage de modélisation est un langage pouvant être graphique ou textuel, destiné à l'architecture, la conception et la mise en œuvre de systèmes complexes. Il est utilisé pour créer des modèles qui représentent la structure, le comportement et les interactions des éléments du système ou du processus.

Les langages de modélisation sont utilisés dans divers domaines par exemple l'ingénierie logicielle, la conception de base de données ou les réseaux et télécommunications...etc.

Dans ce projet nous allons utiliser UML comme langage de modélisation et OCL qui est un langage d'expression des contraintes logicielles utilisé par UML lui-même.

II.3.1.1 UML

« Le langage de modélisation unifié (UML) est une famille de notations graphiques, soutenues par un méta-modèle unique, qui aident à décrire et à concevoir des systèmes logiciels, en particulier des systèmes logiciels construits à l'aide du style orienté objet ⁹(OO). » [12].

UML est créé par Rudy Booch, Ivar Jacobson et James Rumbaugh en 1995 alors qu'ils travaillaient chez Rational Software et il reste le langage de modélisation le plus populaire.

⁹ Est un paradigme de programmation informatique qui s'articule autour d'objets et de données.

II.3.1.2 OCL

OCL (Object Constraint Language) développé en 1995 par IBM, est un langage formel d'expression de contraintes bien adapté aux diagrammes d'UML, et en particulier au diagramme de classes.

OCL permet de spécifier des conditions et des contraintes qui doivent être respectés par les instances des modèles objets.

II.3.2 Patron de conception

Les patrons de conception offrent des solutions éprouvées à des problèmes courants de conception, permettant ainsi aux développeurs de bénéficier des bonnes pratiques et de faciliter le processus de développement.

Le Modèle MVC

Le modèle MVC (Model-View-Controller) est un design pattern¹⁰ ou patron de conception pour le développement de logiciels. Il est souvent utilisé dans les applications web pour séparer la logique de présentation (l'interface utilisateur) de la logique métier (traitement des données).

II.4. Langages de programmation, bibliothèques et frameworks

Dans cette section nous allons présenter les langages de programmation, les bibliothèques ainsi que les frameworks utilisés pour l'implémentation de notre application

II.4.1 Langages de programmation

Nous allons d'abord commencer par les langages de programmation :

a HTML

Le HTML (Hypertext Markup Language) Désigne un type de langage informatique descriptif qui permet de structurer le contenu d'une page web en utilisant des balises [28].

¹⁰ Est un arrangement caractéristique de modules, reconnu comme bonne pratique en réponse à un problème de conception d'un logiciel.

b CSS

Le CSS (Cascading Style Sheets), est un langage informatique utilisé sur Internet pour la mise en forme de fichiers et de pages HTML et XML (eXtensible Markup Language). Le CSS permet d'éditer les couleurs, les polices, le placement des éléments [28].

c JavaScript

JavaScript désigne un langage de développement informatique créé en 1995 par Brendan Eich lorsqu'il travaillait chez Netscape Communications Corporation, et plus précisément un langage de script¹¹ orienté objet. Il permet aux développeurs de créer des pages web dynamiques et interactives [29].

d Dart

Dart est un langage de programmation optimisé pour les applications sur plusieurs plateformes. Il est développé par Google en 2011 et est utilisé pour créer des applications mobiles, de bureau et web.

Dart est un langage orienté objet à ramasse-miettes¹² ce qui le rend optimisée avec une syntaxe similaire à celle du C++ [31].

e SQL

SQL (Structured Query Language) est un langage de programmation permettant de manipuler les données et les systèmes de bases de données relationnelles créé en 1974 par Raymond Boyce and Donald Chamberlin [30].

II.4.2 Bibliothèques

Passons à présent aux bibliothèques :

a ReactJs

React est une bibliothèque JavaScript open source développée par Facebook en 2013, Elle permet de construire des interfaces utilisateurs dynamiques et interactives notamment des

¹¹ Ce sont des programmes destinés à être exécutés rapidement et sans compilation préalable par exemple sur des navigateurs.

¹² C'est un concept qui permet de déterminer quels objets ne peuvent plus être utilisés par le programme et à récupérer l'espace mémoire utilisé par ces objets.

Chapitre II : Méthodologie de conception et langages de programmation

applications web monopage aussi dite SPA (Single Page Application) en utilisant des composants réutilisables [32].

b **Sequelize**

Sequelize est une bibliothèque ORM (Object-Relational Mapping) pour NodeJs et qui permet de communiquer avec une base de données relationnelle en utilisant des objets JavaScript plutôt que d'écrire des requêtes SQL[33].

II.4.3 Frameworks

Enfin les frameworks utilisés :

a **NextJs**

Next.js est un framework¹³ Web open source basé sur React qui permet aux développeurs de créer des applications Web, il offre des fonctionnalités en plus comme le rendu côté serveur dit SSR (Server Side Rendering) qui permet de rendre le contenu dynamique au moment de la demande, et la génération de pages statiques dit SSG (Static Site Generation) qui permet aux pages web d'être servies plus rapidement et plus efficacement [32].

b **Flutter**

Flutter est un framework Open Source développé par Google basé sur Dart, la première version est sortie en 2017.

Flutter utilise les widgets¹⁴ qui sont imbriqués les uns dans les autres, ce qui lui permet d'obtenir un gain de performance phénoménal [31].

II.5. Outils de conception et de développement

Parmi la multitude d'outils disponibles pour la réalisation d'une application informatique, nous avons opté pour les outils que nous présenterons ci-dessous.

¹³ Est un ensemble cohérent de composants logiciels structurels qui sert à créer les fondations ainsi que les grandes lignes de tout ou partie d'un logiciel.

¹⁴ Ce sont les principaux composants d'une application Flutter, comme les boutons, le texte, les images.

a Papyrus

Papyrus est une plate-forme d'ingénierie système basée sur des modèles qui facilitent la conception de systèmes et de logiciels complexes. Il utilise les langages de modélisation standard UML et SysML et fait partie de l'environnement de développement open source Eclipse [21].

b Visual Studio Code

Visual Studio Code souvent abrégé en VsCode, c'est un éditeur de code gratuit et open source. Il est développé par Microsoft en 2015. Il est largement utilisé par les développeurs pour divers langages de programmation et plates-formes, sa force réside notamment dans la multitude d'extensions et de plugins qu'il propose.

c PgAdmin

PgAdmin est une plateforme d'administration et de développement pour PostgreSQL¹⁵, open source et populaire. Il offre une interface conviviale pour la gestion et l'interaction avec les bases de données PostgreSQL.

d Visual Studio

Souvent confondu avec Visual Studio Code, Visual Studio est un IDE (Integrated Development Environment) ou environnement de développement complet pour les développeurs .NET et C++ notamment pour construire des logiciels Windows. Il est développé par Microsoft.

e Git

Git est un VCS (Version Control System) ou système de contrôle de version qui a été développé par Linus Torvalds en 2005. Git permet aux développeurs d'enregistrer un historique des modifications apportées au code, et leur permet de revenir à l'état précédent au cas où une erreur est rencontrée.

¹⁵ Est un SGBDR (système de gestion de base de données relationnelles), qui implémente les standards SQL.

f Postman

Postman est un logiciel principalement fait pour permettre aux développeurs de tester des requêtes HTTP et de s'assurer du bon fonctionnement des APIs (Application Programming Interface). Il est développé par Abhinav Asthana.

II.6. Conclusion

Dans ce chapitre nous avons mis le point sur les méthodes de conception adoptées à savoir Scrum combinée à UP ainsi que les langages de programmation, bibliothèques et Frameworks de développement à utiliser pour la réalisation de notre application. Dans ce qui suit, nous allons présenter une analyse préliminaire de notre futur système.

Chapitre III : Analyse Préliminaire

III.1. Introduction

Après avoir choisi un processus de développement à suivre, nous pouvons à présent passer à la prochaine étape de la mise en œuvre de notre projet qui est l'analyse préliminaire.

C'est durant cette phase que le contexte est défini, où nous allons présenter et étudier l'environnement du système, puis spécifier les besoins fonctionnels et non fonctionnels, et identifier les différents acteurs. Nous pourrons ensuite passer à l'élaboration du Product Backlog qui contiendra une liste complète de toutes les fonctionnalités souhaitées, et faire la planification des Sprints et ce dans le cadre de l'adoption de la méthode Scrum.

III.2. Spécification des besoins

La spécification des besoins est une étape fondamentale et c'est un processus qui permet de définir et de décrire principalement les fonctionnalités attendues par le produit final. C'est pourquoi elle doit être faite dès le début et avec précision, elle implique souvent toutes les parties prenantes du projet pour s'assurer que toutes les attentes du client sont bien prises en compte.

Toute erreur dans l'étape de spécification des besoins peut être cruciale dans la suite projet.

III.2.1 Identification des acteurs

D'après [20], un acteur est un rôle qu'un utilisateur joue par rapport au système.

C'est une entité qui interagit de façon directe avec le système étudié, un acteur peut être une personne physique, un autre système, une base de données, etc. L'identification de celui-ci va nous permettre plus tard de définir les cas d'utilisations relatifs à chaque acteur. Une seule personne physique peut parfaitement jouer le rôle de deux acteurs ou plus.

Notre système sera composé de deux parties : le sous-système de gestion des rdvs, une application web appelée 'MyDocBooking' et le sous-système de gestion des files d'attente, une application appelée MedFlow.

Ci-dessous les acteurs de notre système :

- **Administrateur de 'MyDocBooking'** : est responsable de la gestion des données associées à l'application web. Son rôle principal consiste à ajouter les établissements qui souhaitent utiliser notre solution et à lier un compte super admin à chaque établissement.

- **Utilisateur de ‘MyDocBooking’**
- **Administrateur de ‘MedFlow’**: celui-ci a tous les droits pour gérer les comptes de l’ensemble du personnel de son établissement (médecins et secrétaires) et consulter les statistiques de celui-ci.
- **Super administrateur de ‘MedFlow’** : En plus d’être un administrateur, il a le droit de gérer l’ensemble des administrateurs.

Avoir un compte super admin et plusieurs comptes admins sert à établir différents niveaux de contrôle d'accès au sein du logiciel, offre la possibilité de répartir les tâches administratives entre plusieurs admins pour permettre une meilleure gestion de travail et une plus grande disponibilité.

- **Médecin**
- **Secrétaire**

III.2.2 Réalisation du diagramme de contexte

Le diagramme de contexte est le premier diagramme utilisé pour décrire le système à développer, et c’est un diagramme qui permet de représenter les différentes interactions entre le système et les acteurs. Quand il est dynamique, il permet de visualiser le flux d’information échangé avec les entités externes.

On trouve ci-dessous le diagramme de contexte dynamique associé à notre système :

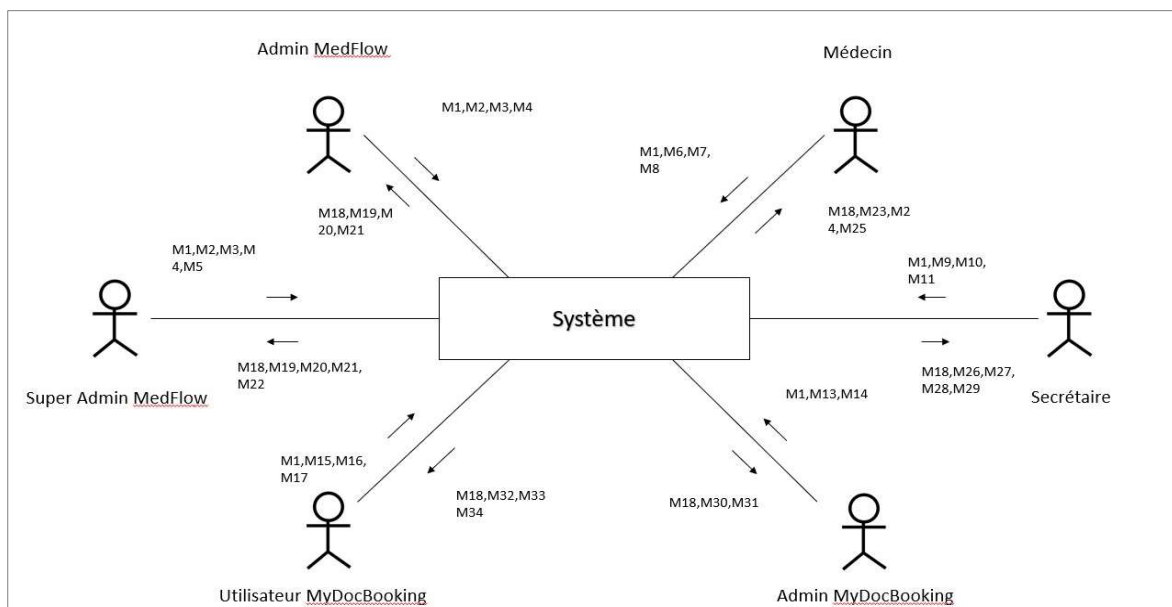


Figure 5 : Diagramme de contexte dynamique.

III.2.3 Identification des messages échangés

Les messages sont utilisés pour représenter l'échange d'informations entre le système étudié et une entité externe. Ils peuvent être des requêtes, des réponses, des messages de confirmation, etc. Nous avons illustré les messages produits par les différents acteurs ainsi que le système dans le tableau ci-dessous :

Acteur	Id	Message envoyé Acteur-système	Id	Message envoyé Système-acteur
Tous les acteurs	M1	Demande d'authentification	M18	Afficher formulaire d'authentification
Administrateur, super-admin de MedFlow	M2	Demande d'ajout/suppression/modification de Secrétaire/médecin	M19	Afficher l'interface correspondante
	M3	Consulter les données statistiques de l'établissement	M20	Afficher la page des données statistiques
	M4	Créer/Modifier/Supprimer une file d'attente	M21	Afficher l'interface permettant de gérer les files d'attente
Super-admin de MedFlow	M5	Demande d'ajout/suppression/modification d'un administrateur	M22	Afficher l'interface permettant de gérer les administrateurs
Médecin	M6	Voir l'historique de ses consultations	M23	Afficher l'historique des consultations
	M7	Demande de notifier le prochain patient	M24	Afficher la notification aux patients
	M8	Consulter son planning	M25	Afficher la page planning du médecin
Secrétaire	M9	Consulter l'ordre des patients dans la file	M26	Afficher l'ordre de passage des patients dans la file
	M10	Ajout/Suppression/Modification d'un patient d'une file	M27	Afficher l'interface permettant de gérer les patients dans une file
	M11	Demande d'Ajouter un établissement hospitalier	M28	Afficher la page permettant d'ajouter un établissement
	M12	Demande de consulter les rendez-vous	M29	Afficher la liste des rendez-vous de la journée
Administrateur de MyDocBooking	M13	Créer un compte super admin associé à l'établissement hospitalier.	M30	Afficher la page permettant d'ajouter un super admin

	M14	Demande d'Inscription	M31	Afficher la page d'inscription
Utilisateur de MyDocBooking	M15	Demande de consulter les rendez-vous	M32	Afficher la page mes rendez-vous
	M16	Demande de sélection d'un créneau	M33	Afficher les créneaux disponibles
	M17	Confirmer un créneau	M34	Envoie d'un mail de confirmation

Tableau 5 : Tableau représentant les messages échangés.

III.2.4 Capture des besoins fonctionnels

Les besoins fonctionnels sont les besoins spécifiques liés aux tâches que le système ou l'application doit accomplir pour satisfaire l'utilisateur final. Il est important de les identifier dès le début pour s'assurer que notre solution couvre toutes les fonctionnalités voulues et qu'elle fonctionne exactement comme prévu.

L'objectif de cette partie est de déterminer les limites, les ressources nécessaires et autres aspects pertinents.

III.2.4.1 Identifications des cas d'utilisation

Un cas d'utilisation est une description détaillée des interactions entre le système et les acteurs, il représente la séquence d'actions dans le but d'atteindre un objectif. Ci-dessous nous avons défini les cas d'utilisation relatifs aux acteurs :

Cas d'utilisation	Id	Acteurs
Gérer les établissements.	1.1	Administrateur de 'MyDocBooking'
Gérer les comptes super-admin des établissements.	1.2	
Gérer les files d'attente.	2.1	Super admin de 'MedFlow' et Administrateur 'MedFlow'
Gérer les services de l'établissement.	2.2	
Gérer les comptes des secrétaires.	2.3	
Gérer les comptes des médecins.	2.4	
Consulter les données statistiques.	2.5	
Gérer les comptes admin de l'établissement.	3	Super admin de 'MedFlow'
S'authentifier.	4	(Tous les acteurs)
Voir l'historique de ses consultations.	5.1	Médecin
Consulter Son planning.	5.2	

Notifier la secrétaire pour appeler le prochain patient de la file.	5.3	
Consulter l'ordre des patients dans la file.	6.1	Secrétaire
Appeler le prochain patient de la file.	6.2	
Gérer les patients dans une file.	6.3	
S'inscrire.	7.1	Utilisateur de 'MyDocBooking'
Consulter les créneaux disponibles.	7.2	
Prendre un rendez-vous.	7.3	
Consulter ses rendez-vous.	7.4	

Tableau 6 : Tableau représentant les cas d'utilisation.

III.2.4.2 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est l'outil qui permet de mieux représenter et de visualiser les besoins fonctionnels sous forme d'un schéma. Ils représentent les interactions entre les acteurs et le système. Ils permettent également la définition des limites entre celui-ci et son environnement.

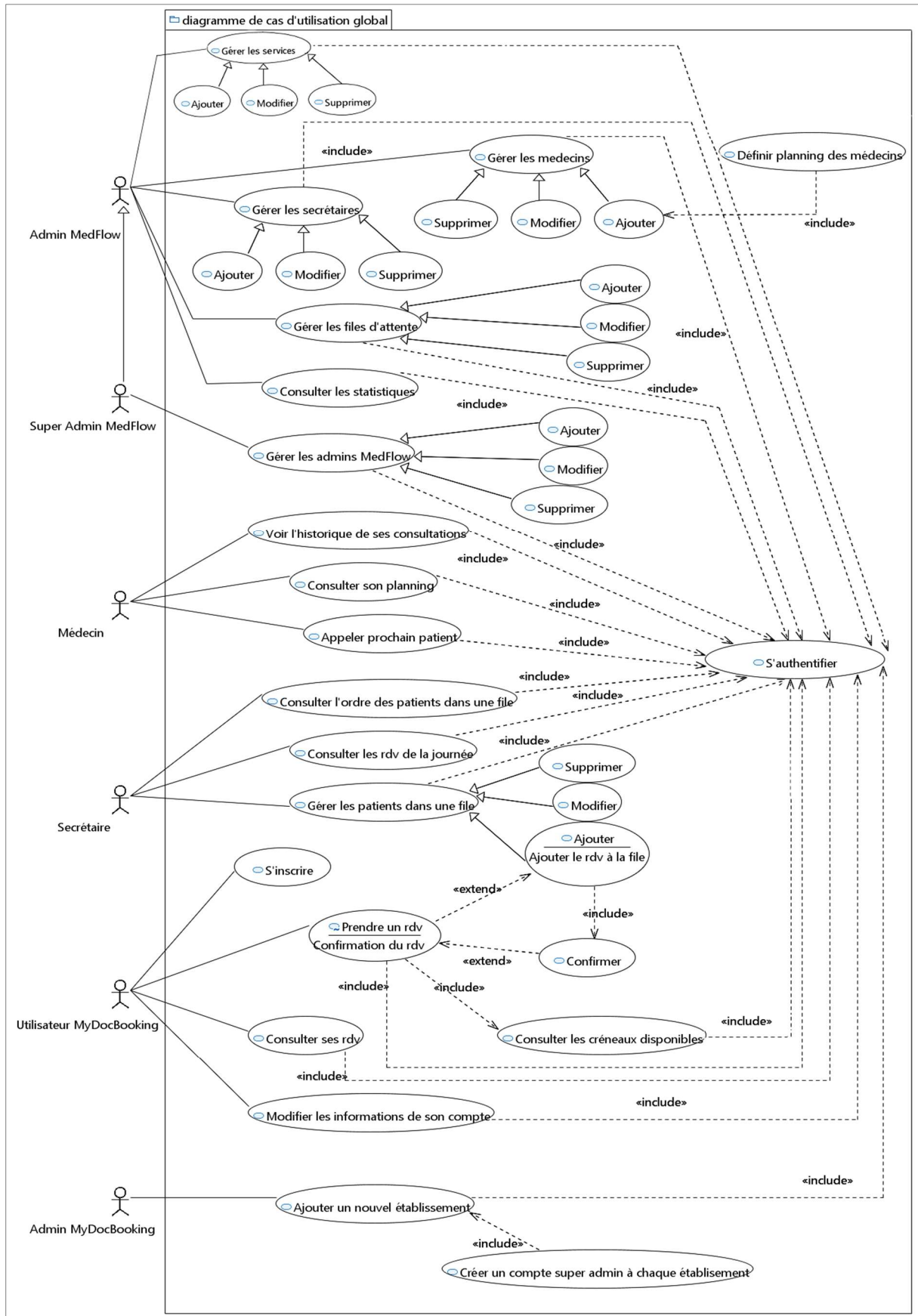


Figure 6 : Diagramme de cas d'utilisation global.

III.2.5 Capture des besoins non fonctionnels

Les besoins non fonctionnels sont tout aussi importants que les besoins fonctionnels, et contrairement à ceux-ci les besoins non fonctionnels ne se concentrent pas sur les fonctionnalités attendues par l'utilisateur final mais plutôt sur le comportement interne du système, et comprend principalement :

- **La performance :** la performance est importante dans notre application pour diverses raisons, tout d'abord en offrant une application rapide et réactive avec un temps d'attente court, nous améliorerons l'expérience utilisateur, en plus de ça en développant une application performante, nous réduisant considérablement les ressources matérielles requises au bon fonctionnement de celle-ci.
- **La sécurité :** afin d'assurer la sécurité de notre application, nous mettons en place des mesures solides, ainsi nous garantissant l'intégrité et la confidentialité des données de l'application, cela inclut la mise en place d'un système d'authentification et un mécanisme de hachage de mots de passe ce qui réduit les risques de piratage, de fraude ou d'accès non autorisé à l'application.
- **La convivialité :** nous avons essayé d'améliorer au maximum la convivialité de notre application, en créant des interfaces intuitives et simples qui permettent l'utilisation facile.

III.3. Gestion du projet avec la méthode Scrum

Structurer un projet avec la méthode Scrum est important pour garantir la réussite de l'équipe et la qualité du produit. En suivant les bonnes pratiques de Scrum on peut tirer avantage de sa flexibilité et efficacité, comme nous pouvons la personnaliser en fonction de nos besoins et de nos capacités en termes de temps, taille de l'équipe, complexité du projet, etc.

Dans cette section nous allons expliquer comment nous avons divisé notre projet en Sprint et comment nous avons partagé les rôles et élaboré le Backlog de notre produit.

III.3.1 Gestion de l'équipe Scrum

Comme cité précédemment l'un des points forts de Scrum et sa capacité à permettre une bonne gestion de l'équipe et en encourageant les collaborations entre les membres de celle-ci. Traditionnellement une équipe Scrum est souvent composée de trois à neuf membres et comprend trois rôles principaux : le Product Owner, le Scrum Master et l'équipe de

développement. Mais il est toujours possible d'ajuster et de personnaliser l'utilisation de Scrum selon nos besoins.

Dans notre cas, l'équipe est composée de deux membres, le responsable de D-Soft M. Hassane ARIOUAT qui joue à la fois les rôles du Scrum Master et du Product Owner tandis que l'équipe de développement est constituée d'une seule personne, en l'occurrence, Mouloud MESSAD.

III.3.2 Backlog du produit

L'élaboration du Product Backlog est une étape importante dans l'application de la méthode Scrum. C'est le processus de collecte et de priorisation des besoins et des fonctionnalités envisagées du produit. Il doit être souvent mis à jour en fonction des besoins changeants et il implique en grande partie Product Owner et l'ensemble de l'équipe de développement.

Le tableau ci-dessous présente le Product Backlog de notre projet :

Acteur	Fonctionnalité	User Story	Risque	Priorité	Sprint
Tous les Utilisateurs	Authentification	Tous les utilisateurs doivent s'authentifier avant d'accéder à l'espace personnel	Fort	Très Elevée	01, 02
Admin MDB	Gérer les établissements et comptes Super-admin	L'administrateur de l'application MDB doit être capable d'ajouter un nouvel établissement dans la base de données	Fort	Très Elevée	01
		L'administrateur de l'application MDB doit lier chaque établissement à un compte super-admin	Fort	Très Elevée	01
		Les utilisateurs devraient être capable de consulter les créneaux disponibles	Fort	Très Elevée	01
		Les utilisateurs devraient être capable de prendre un rendez-vous	Fort	Très Elevée	01

Chapitre III : Analyse Préliminaire

Utilisateur MDB	Gestion de rendez-vous	Les utilisateurs devraient être capable de consulter leurs rendez-vous	Faible	Moyenne	01
		Les utilisateurs devraient être capable de confirmer leurs rendez-vous par email	Fort	Elevée	01
	Inscription	Les utilisateurs devraient être capable de créer un compte sur la plateforme MDB	Moyen	Très Elevée	01
Super admin et Admin MF	Gérer les comptes du personnel	Les administrateurs de chaque établissement devraient être capable de créer, modifier ou supprimer les comptes des secrétaires et des médecins	Faible	Elevée	02
	Consulter les statistiques	Les administrateurs devraient avoir accès à la page statistiques	Moyen	Moyenne	02
Super admin MF	Gérer les comptes des admins	En plus des fonctionnalités des administrateurs, le super admin doit être capable de gérer les comptes administrateurs liées à son établissement	Moyen	Très Elevée	02
Secrétaire	Gérer les patients dans les files	La secrétaire doit être capable d'ajouter, modifier et supprimer les patients d'une file	Fort	Elevée	02
		Consulter l'ordre de passage d'une file d'attente	Fort	Elevée	02
	Consulte la liste des rendez-vous pris en ligne.	La secrétaire doit être capable de consulter la liste des rendez-vous.	Faible	Basse	02
Médecin	Consulter ses informations	Le médecin devrait être capable de voir l'historique de ses consultations	Faible	Basse	02
		Le médecin devrait être capable de consulter son planning.	Moyen	Moyenne	02
	Appeler le prochain patient	Le médecin doit être capable d'appeler le prochain patient.	Faible	Elevée	02

Tableau 7 : Tableau représentant le Product Backlog.

III.3.3 Découpage en sprint

Le découpage en Sprint consiste à diviser le Backlog du produit en éléments plus petits et facilement gérables pour mieux suivre l'avancement du projet.

Nous avons donc décidé de découper notre projet en deux Sprints, un premier Sprint au cours duquel nous allons nous concentrer sur la gestion des rendez-vous et qui conduira au développement de l'application MyDocBooking (MDB), et un deuxième Sprint dédié à la partie gestion de files d'attente et qui mènera au développement de l'application MedFlow (MF).

III.3.4 Planification des Sprints

Une fois le Product Backlog élaboré, nous pouvons à présent passer à la planification des sprints.

La planification des Sprints est une étape essentielle dans le processus Scrum, elle prend en considération un ensemble d'éléments tel la complexité des tâches à réaliser durant le Sprint, où chaque tâche est estimée en termes de temps requis pour la complétion de celle-ci.

En examinant les éléments du PB en haut nous avons décidé de procéder à la planification des Sprints comme représenté dans le tableau ci-dessous :

N° Sprint	Tâches	Durée
01	Application de Gestion des Rendez-vous (MyDocBooking)	25/04/2023 au 17/05/2023
02	Gestion des files et du flux de patients (MedFlow)	18/05/2023 au 17/06/2023

Tableau 8 : planification des Sprints.

III.4. Conclusion

L'analyse préliminaire effectuée dans ce chapitre nous a permis de bien préparer l'équipe et l'environnement de développement pour bien mener la suite du projet. Durant cette étape nous avons identifié les acteurs de notre système, défini les cas d'utilisation, élaboré le Product Backlog et avons fait la planification des sprints.

Dans le chapitre suivant nous allons enfin commencer la réalisation de l'application 'MyDocBooking' avec le premier sprint.

Chapitre IV : Sprint 1 “Gestion de rendez-vous”

Chapitre IV : Sprint 1 “Gestion de rendez-vous”

IV.1 Introduction

Durant le précédent chapitre, nous avons traité la partie analyse des besoins où nous avons identifié les acteurs et les cas d'utilisation et fait la planification du projet.

Dans le présent chapitre nous allons entamer le premier Sprint qui va traiter la partie gestion des rendez-vous. Pour commencer nous allons regrouper toutes les User Stories (U.S) qui seront développées au cours de ce Sprint dans un Backlog appelé Sprint Backlog, chaque User Story sera alors décomposée en tâches plus spécifiques et leur développement va devoir passer par les étapes de UP qui sont la pré étude (inception), l'élaboration, la construction et la transition.

IV.2 Backlog du premier Sprint

Ci-dessous le tableau qui représente le Backlog du premier Sprint consacré à l'application 'MyDocBooking' (MDB).

Acteur	ID US	User Story	ID Tâche	Tâche	Durée
Administrateur de MDB, Utilisateur de MDB	1	L'administrateur et les utilisateurs de l'application MDB doivent s'authentifier avant d'accéder à l'espace personnel.	1.1	Réalisation du diagramme de cas d'utilisation et la description textuelle du cas 'S'authentifier', et diagramme de séquence correspondant au cas.	5h
			1.2	Développer la fonctionnalité 'S'authentifier'.	10h
			1.3	Tester la fonctionnalité 'S'authentifier'.	2h
	2	L'administrateur de l'application MDB doit être capable d'ajouter un	2.1	Réalisation du diagramme de cas d'utilisation et la description textuelle du cas 'Ajouter établissement, et diagramme de séquence correspondant au cas.	8h
			2.2	Développer la fonctionnalité 'Ajouter établissement'.	2h

Chapitre IV : Sprint 1 “Gestion de rendez-vous”

Administrateur MDB		nouvel établissement dans la base de données.		Tester la fonctionnalité ‘Ajouter établissement’.	2h
	3	L’administrateur de l’application MDB doit lier chaque établissement à un compte Super-Admin.	3.1	Réalisation du diagramme de cas d’utilisation et la description textuelle du cas ‘Ajouter Super admin’, et diagramme de séquence correspondant au cas.	7h
			3.2	Développer la fonctionnalité ‘Ajouter Super admin’.	2h
			3.3	Tester la fonctionnalité ‘Ajouter Super admin’.	2h
Utilisateur MDB	4	Les utilisateurs doivent être capable de créer un compte sur la plateforme MDB.	4.1	Réalisation du diagramme de cas d’utilisation et la description textuelle du cas ‘créer un compte’, et diagramme de séquence correspondant au cas.	10h
			4.2	Développer la fonctionnalité ‘créer un compte’.	5h
			4.3	Tester la fonctionnalité ‘créer un compte’.	1h
	5	Les utilisateurs doivent être capables de consulter les créneaux disponibles.	5.1	Réalisation du diagramme de cas d’utilisation et la description textuelle du cas ‘consulter les créneaux disponibles’, et diagramme de séquence correspondant au cas.	10h
			5.2	Développer la fonctionnalité ‘consulter les créneaux disponibles’.	5h
			5.3	Tester la fonctionnalité ‘consulter les créneaux disponibles’.	2h
	6	Les utilisateurs doivent être capable de prendre un rendez-vous.	6.1	Réalisation du diagramme de cas d’utilisation et la description textuelle du cas ‘prendre un rendez-vous’, et diagramme de séquence correspondant au cas.	10h

Chapitre IV : Sprint 1 “Gestion de rendez-vous”

		6.2	Développer la fonctionnalité ‘prendre un rendez-vous’.	12h
		6.3	Tester la fonctionnalité ‘prendre un rendez-vous’.	2h
7	Les utilisateurs doivent être capable de consulter leurs rendez-vous.	7.1	Réalisation du diagramme de cas d’utilisation et la description textuelle du cas ‘consulter les rendez-vous’, et diagramme de séquence correspondant au cas.	10h
		7.2	Développer la fonctionnalité ‘consulter les rendez-vous’.	8h
		7.3	Tester la fonctionnalité ‘consulter les rendez-vous’.	1h
8	Les utilisateurs doivent être capable de confirmer leurs rendez-vous par email.	8.1	Réalisation du diagramme de cas d’utilisation et la description textuelle du cas ‘confirmer les rendez-vous par email’, et diagramme de séquence correspondant au cas.	10h
		8.2	Développer la fonctionnalité ‘confirmer les rendez-vous par email’.	7h
		8.3	Tester la fonctionnalité ‘confirmer les rendez-vous par email’.	2h
9	Les utilisateurs doivent être capable de modifier s informations de leur compte	9.1	Réalisation du diagramme de cas d’utilisation et la description textuelle du cas ‘modifier les informations du compte’, et diagramme de séquence correspondant au cas.	10h
		9.2	Développer la fonctionnalité ‘modifier les informations du compte’.	7h
		9.3	Tester la fonctionnalité ‘modifier les informations du compte’.	2h

Tableau 9 : Tableau représentant le Backlog du premier Sprint.

IV.3 Spécifications fonctionnelles

Dans ce premier Sprint nous allons étudier en détails les deux fonctionnalités principales qu’offre notre application web ‘MyDocBooking’.

Cette application permet aux utilisateurs de créer un compte puis prendre un rendez-vous dans un établissement de leur choix. Elle permet également à l’administrateur d’ajouter un nouvel établissement à la base de données puis créer un compte Super Administrateur, le super administrateur se chargera de configurer l’application ‘MedFlow’ propre à son établissement mais cette fonctionnalité sera réalisée durant le Sprint 2.

De ce fait nous avons constaté que les cas d’utilisation sont reliés de la façon suivante :

- L’utilisateur doit s’inscrire sur la plateforme MDB s’il ne dispose pas encore de compte.
- L’ensemble des cas d’utilisation incluent le cas « S’authentifier », par exemple si l’administrateur veut ajouter un établissement ou si un utilisateur souhaite prendre un rendez-vous, ils auront à s’authentifier au préalable.
- Le cas d’utilisation « Prendre un rendez-vous » inclut le cas « Consulter les créneaux disponibles », par exemple un utilisateur doit d’abord consulter les créneaux disponibles avant de prendre un rendez-vous pour une date précise.
- Les rendez-vous doivent être confirmés plus tard sinon ceux-ci ne seront pas pris en considération. Les utilisateurs reçoivent un mail de confirmation la veille de la date du rendez-vous.
- Les utilisateurs peuvent consulter leur rendez-vous.

IV.3.1 Diagramme de cas d’utilisation

La figure ci-dessous présente le diagramme de cas d’utilisation qui schématise les cas cités en haut :

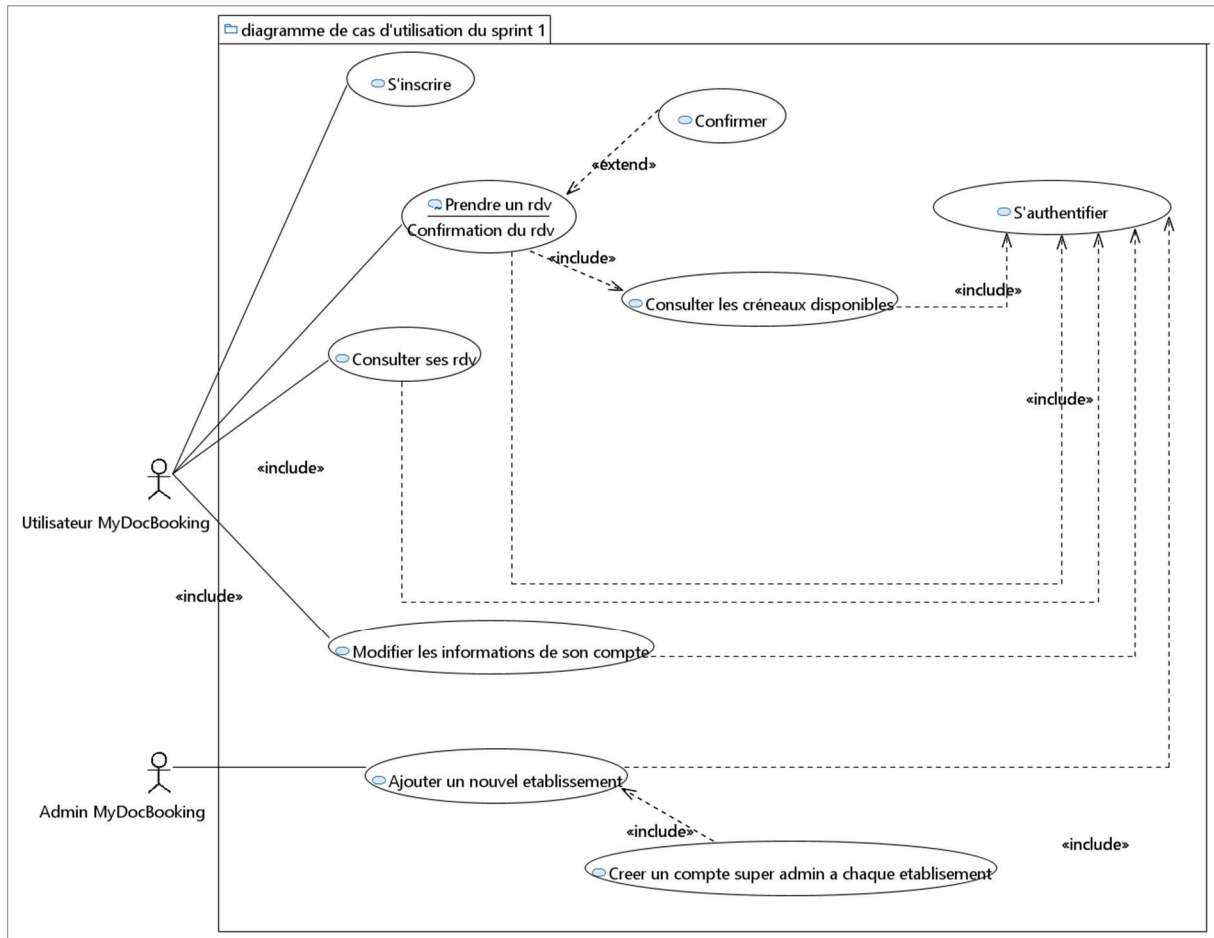


Figure 7 : Diagramme de cas d’utilisation du premier Sprint.

IV.3.2 Diagramme d’activité global du premier Sprint

Le diagramme ci-dessous présente une vue globale du fonctionnement de l’application ‘MyDocBooking’ et qui sera développée au cours de ce Sprint.

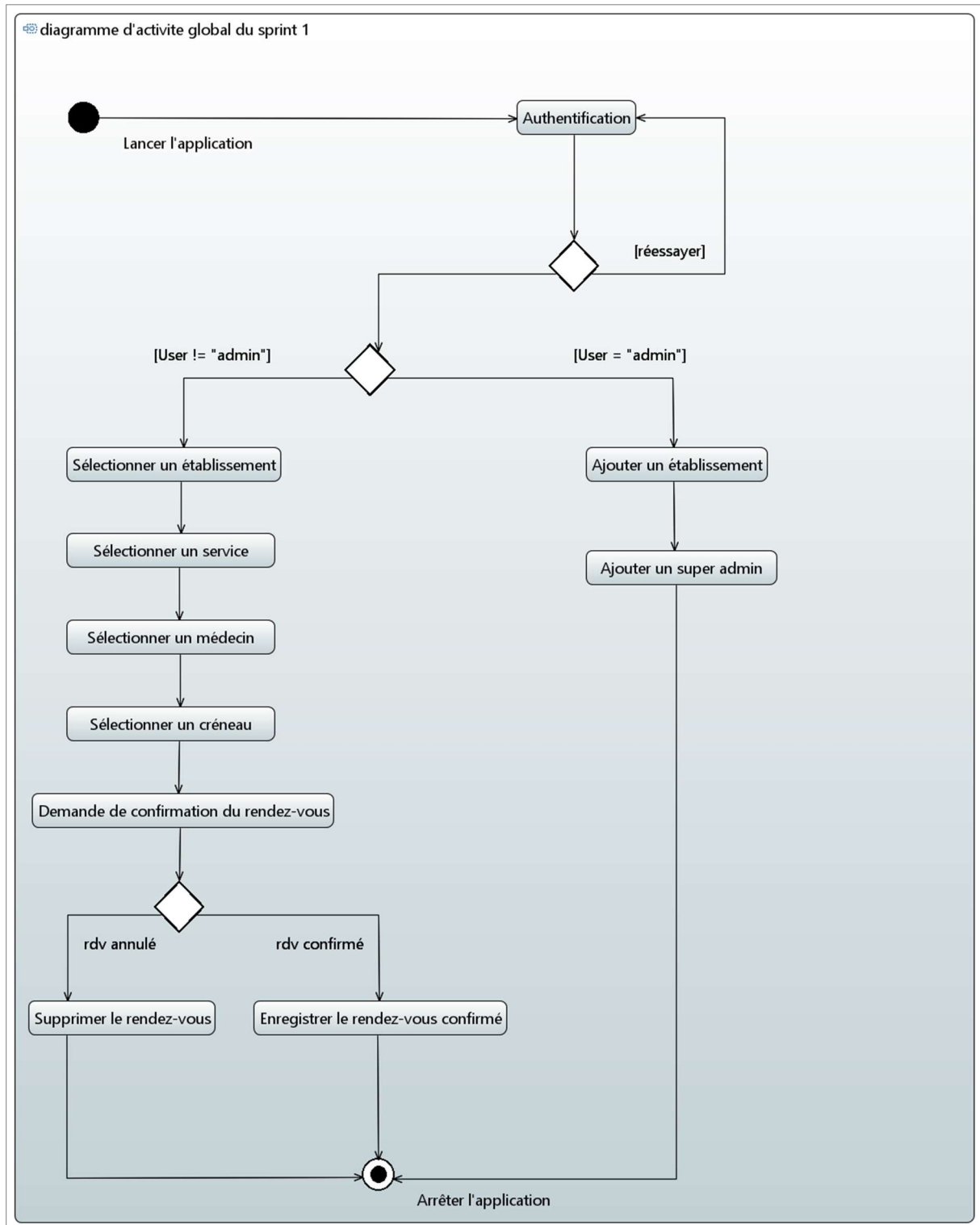


Figure 8 : Diagramme d'activité du premier Sprint.

IV.3.3 Description Textuelle des cas d'utilisation

Dans cette section nous allons décrire les cas d'utilisation les plus pertinents à savoir le cas « S'authentifier », « Prendre un rendez-vous » et « Confirmer un rendez-vous »

Chapitre IV : Sprint 1 “Gestion de rendez-vous”

IV.3.3.1 Cas d'utilisation « S'authentifier »

Le tableau ci-dessous décrit le cas d'utilisation « S'authentifier » :

Sommaire	
Titre	S'authentifier.
But	Permet à l'utilisateur ou à l'administrateur d'accéder à l'espace qui lui est réservé.
Acteur	Administrateur, Utilisateur
Pré condition	<ul style="list-style-type: none">• L'utilisateur est inscrit.• L'utilisateur ou l'administrateur doit accéder à la page de connexion.
Scenario	<ul style="list-style-type: none">• Le système affiche le formulaire.• L'utilisateur saisie son adresse email et son mot de passe.• Le système envoie une requête de vérification de l'existence de l'adresse mail et de la validité du mot de passe dans la base de données.• Le système créer un token d'authentification.
Post condition	L'utilisateur ou l'administrateur obtient l'accès à l'espace qui lui est réservé.
Alternative	Le système ne trouve pas l'utilisateur dans la base de données et envoie une réponse (404 NOT FOUND).
Besoin d'IHM	<ul style="list-style-type: none">• Formulaire de connexion.

Tableau 10 : Description textuelle du cas d'utilisation « S'authentifier ».

IV.3.3.2 Cas d'utilisation « Prendre un rendez-vous »

Le tableau ci-dessous décrit le cas d'utilisation « Prendre un rendez-vous » :

Sommaire	
Titre	Prendre un rendez-vous
But	Permet à l'utilisateur de choisir un créneau pour fixer un rendez-vous
Acteur	Utilisateur
Pré condition	L'utilisateur doit s'authentifier
Scenario	<ul style="list-style-type: none">• L'utilisateur accède à la page de prise de rendez-vous.• L'utilisateur sélectionne un établissement.• L'utilisateur sélectionne un service• L'utilisateur sélectionne un médecin• L'utilisateur sélectionne un créneau• Le système affiche la Popup de validation de la réservation.
Post condition	Le rendez-vous est pris
Alternative	L'utilisateur choisi un créneau qui a été déjà pris, le système affiche donc un message demandant de sélectionner un autre créneau.

Chapitre IV : Sprint 1 “Gestion de rendez-vous”

Besoin d'IHM	<ul style="list-style-type: none">• Formulaire de sélection d'établissement, de service, de médecin et de créneau.
--------------	--

Tableau 11 : Description textuelle du cas d'utilisation « Prendre un rendez-vous ».

IV.3.3.3 Cas d'utilisation « Confirmer un rendez-vous »

Le tableau ci-dessous décrit le cas d'utilisation « Confirmer un rendez-vous » :

Sommaire	
Titre	Confirmer un rendez-vous
But	Confirmer un rendez-vous
Acteur	Utilisateur
Pré condition	<ul style="list-style-type: none">• L'utilisateur doit s'authentifier.• L'utilisateur doit d'abord sélectionner un créneau, valider le rendez-vous.• L'utilisateur a reçu par mail un lien de confirmation de son rendez-vous et ce 48h avant.
Scenario	<ul style="list-style-type: none">• L'utilisateur clique sur le lien de confirmation et sera renvoyé vers la page mes rendez-vous.• Le système affiche un message disant que le rendez-vous a été confirmé avec succès.
Post condition	<ul style="list-style-type: none">• Le rendez-vous est confirmé.
Alternative	<ul style="list-style-type: none">• L'utilisateur ne confirme pas le rendez-vous pris.• Le système ne va donc pas prendre en considération ce rendez-vous et sera supprimé de la base de données.
Besoin d'IHM	Interface mes rendez-vous

Tableau 12 : Description textuelle du cas d'utilisation « Confirmer un rendez-vous ».

IV.4 Analyse

Notre objectif dans cette section est d'exploiter efficacement les besoins et exigences citées précédemment et comprendre en profondeur le fonctionnement du système pour faciliter sa conception. Cette phase décrit précisément ce que le système doit faire.

En premier lieu nous allons commencer par définir le modèle du domaine puis construire les diagrammes de séquence système.

IV.4.1 Modèle du domaine

Le modèle du domaine souvent représenté sous forme de diagramme de classe, il identifie les entités principales de notre domaine, leurs attributs et les relations qui les lient.

Ces entités ou concepts peuvent être identifiés directement à partir de la connaissance du domaine ou par des entretiens avec des experts du domaine. Il faut absolument utiliser le

Chapitre IV : Sprint 1 “Gestion de rendez-vous”

vocabulaire du métier pour nommer les classes et leurs attributs. Les classes du modèle du domaine ne doivent pas contenir d'opérations, mais seulement des attributs [22].

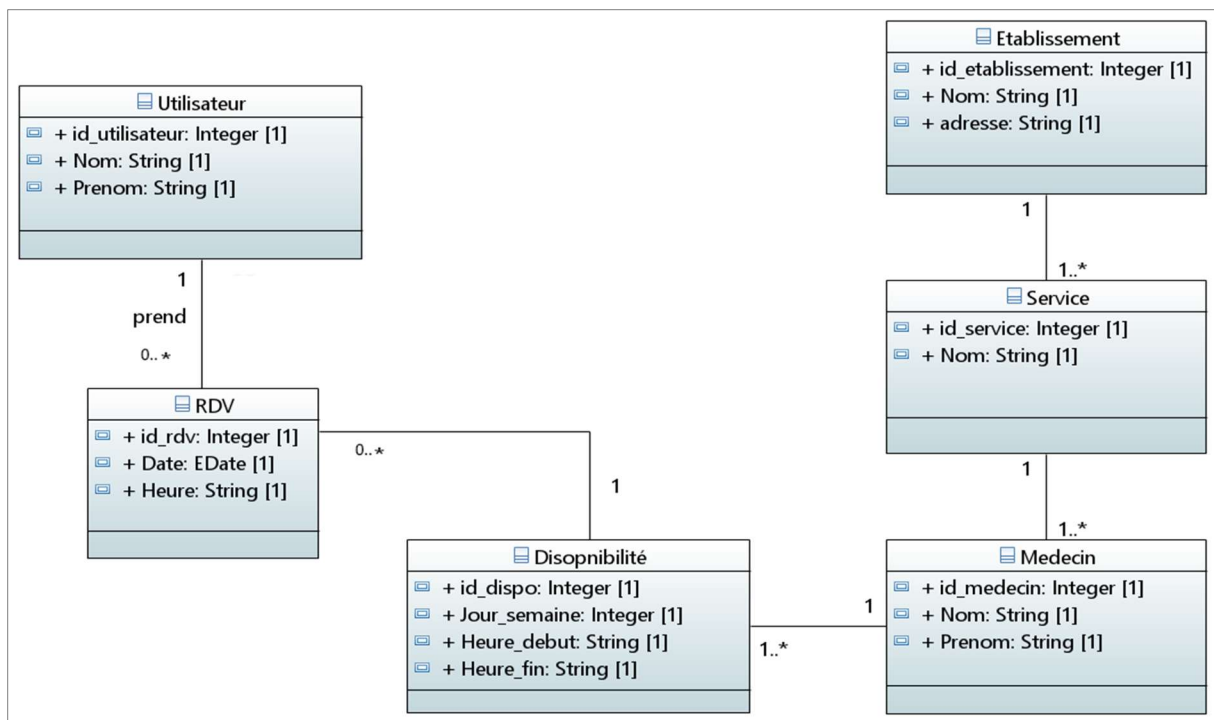


Figure 9 : Modèle du domaine du premier Sprint.

IV.4.2 Diagrammes de Séquence Système

Le diagramme de séquence système permet de décrire comment le système lui-même interagit avec les acteurs externes. Généralement, un diagramme de séquence capture le comportement d'un scénario donné.

IV.4.2.1 Diagramme de séquence système du cas « S’authentifier »

Pour pouvoir réaliser les différentes actions sur l’application, l’utilisateur doit d’abord s’authentifier en remplissant et en validant le formulaire de connexion. Le système se chargera alors de vérifier la validité des informations saisies, si les informations sont correctes l’utilisateur sera redirigé vers la page demandée sinon un message d’erreur lui sera affiché.

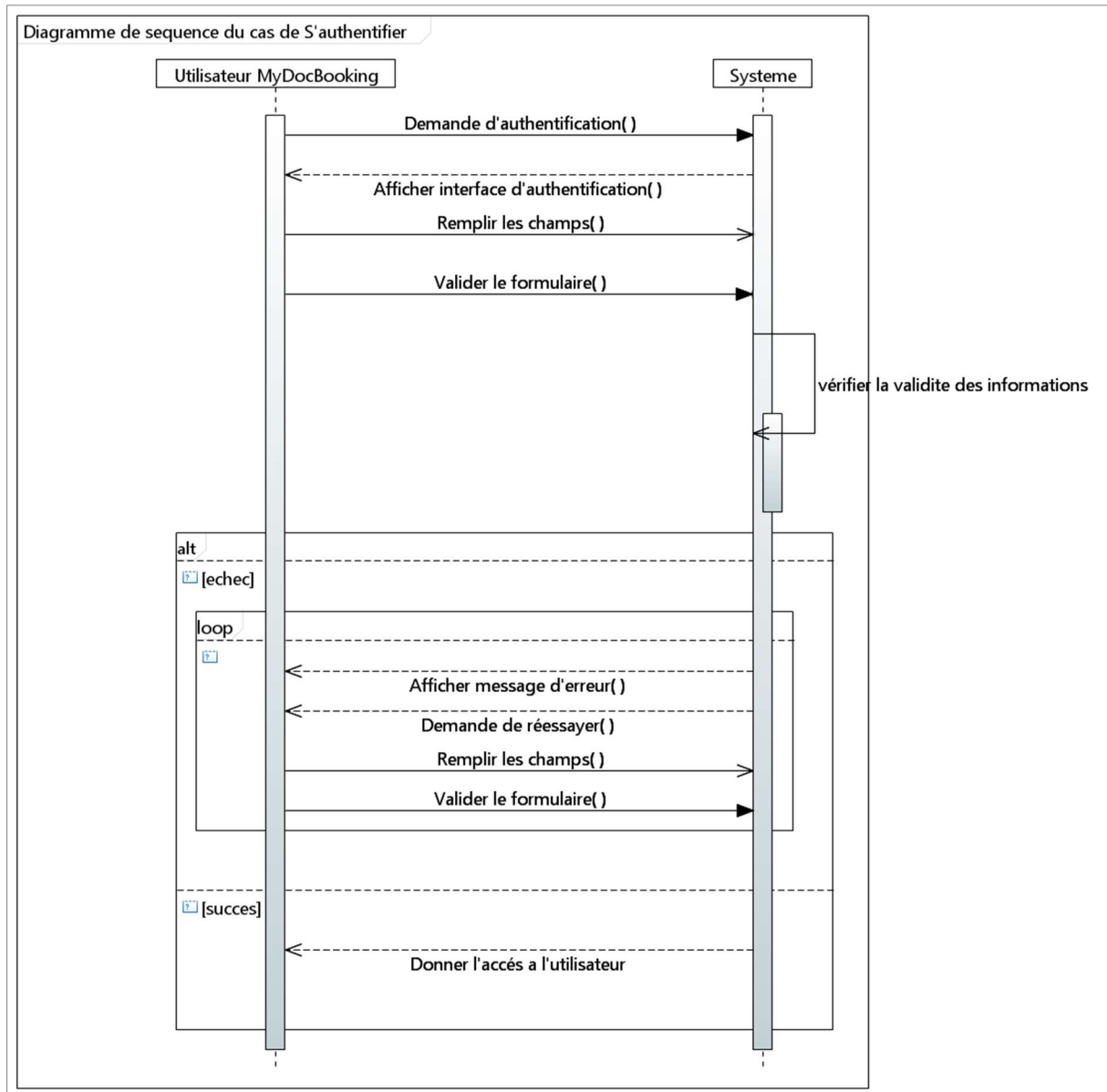


Figure 10 : Diagramme de séquence du cas S’authentifier.

IV.4.2.2 Diagramme de séquence système du cas « S’inscrire »

Pour pouvoir s’authentifier et utiliser correctement l’application, l’utilisateur doit créer un compte s’il n’en a pas un. L’utilisateur doit remplir le formulaire d’inscription puis le système vérifiera la validité des information saisies et l’unicité de l’adresse email, si celle-ci existe déjà l’utilisateur est alors redirigé vers la page d’authentification.

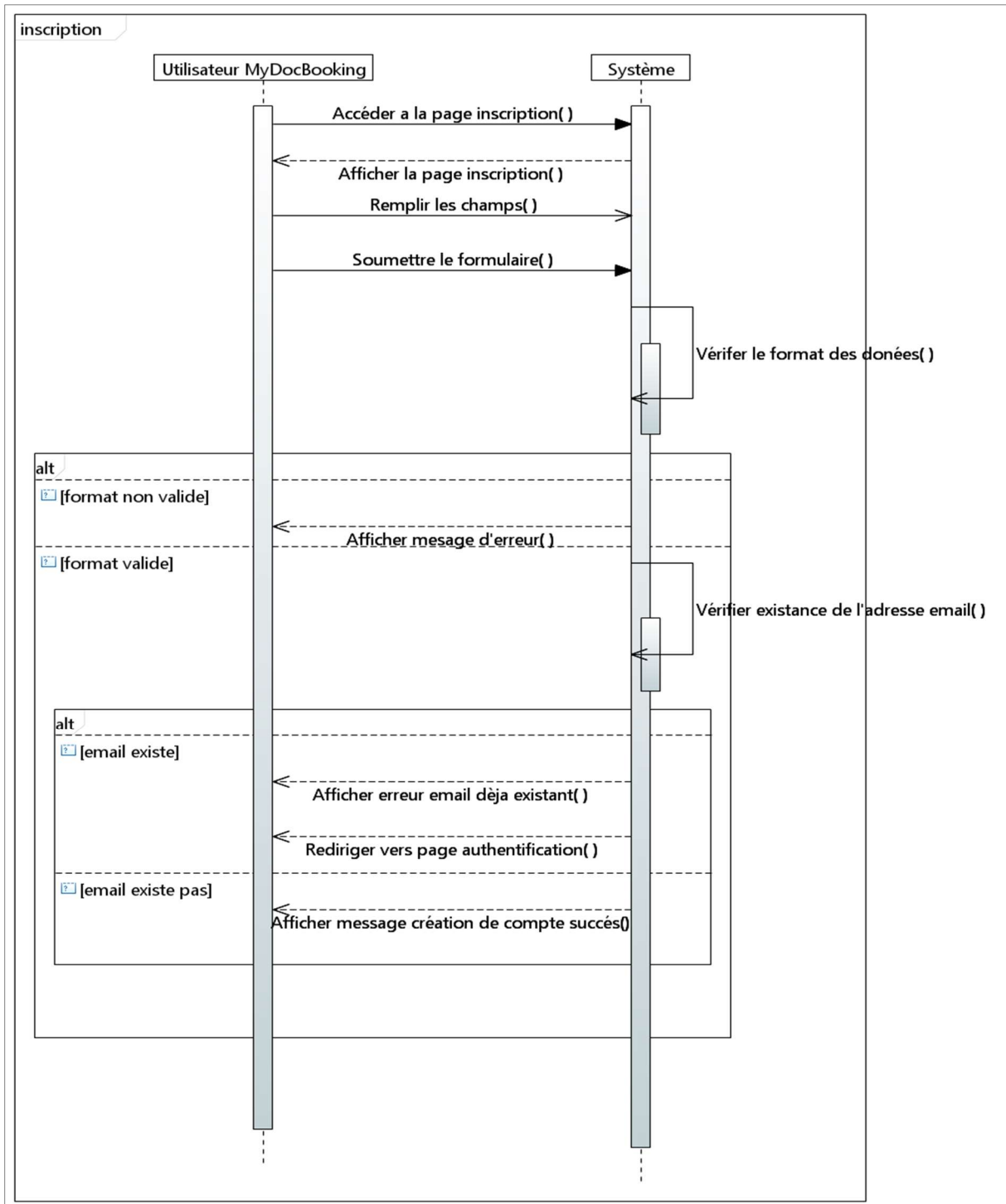


Figure 11 : Diagramme de séquence système du cas S’inscrire.

IV.4.2.3 Diagramme de séquence système du cas « Prendre un rendez-vous »

Une fois l'utilisateur authentifié, il sera redirigé vers la page de prise de rendez-vous. Sur cette page, il devra effectuer les étapes suivantes :

Chapitre IV : Sprint 1 “Gestion de rendez-vous”

- Sélectionner un établissement où il souhaite prendre rendez-vous.
- Choisir le service spécifique parmi ceux présent dans l'établissement sélectionné.
- Sélectionner un médecin parmi ceux disponibles dans le service sélectionné.
- Le système affichera alors les créneaux horaires disponibles pour ce médecin.
- L'utilisateur devra choisir un créneau qui lui convient parmi les options proposées.

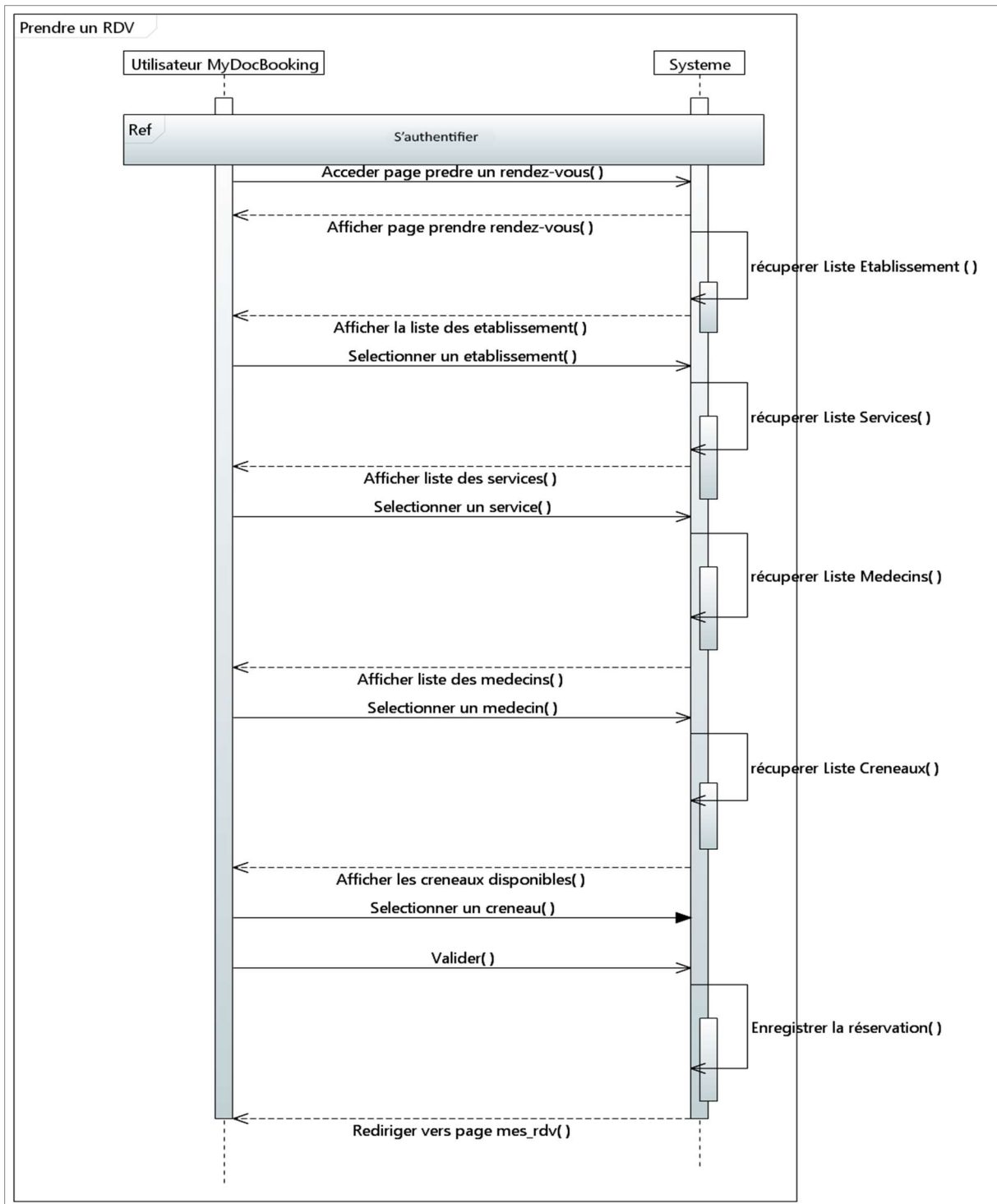


Figure 12 : Diagramme de séquence système du cas Prendre un rendez-vous.

IV.4.2.4 Diagramme de séquence système du cas « Confirmer le rendez-vous »

Une fois que l'utilisateur aura sélectionné un créneau il devra passer par une dernière étape qui est la confirmation. L'utilisateur va alors recevoir un mail lui demandant de cliquer sur le lien afin de confirmer son rendez-vous.

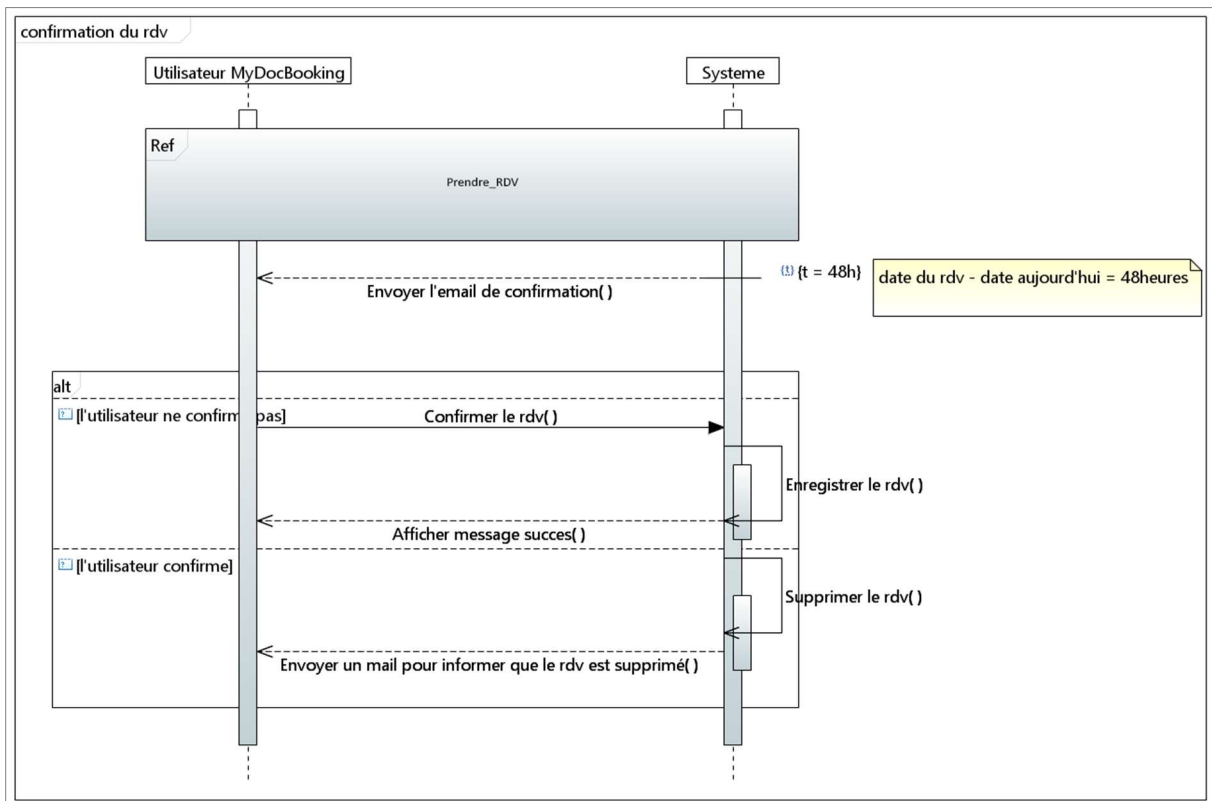


Figure 13 : Diagramme de séquence du cas confirmer un rendez-vous.

IV.5 Conception

La phase de conception consiste à définir l'architecture du système, mettre en place un plan détaillé de la réalisation du système et à concevoir la solution qui permettra de répondre aux exigences identifiées précédemment.

Durant cette étape nous allons définir les diagrammes de séquence détaillés et construire notre diagramme de classe du premier Sprint.

IV.5.1 Diagrammes de séquence détaillés

Comme son nom l’indique, c’est un type de diagramme de séquence dont le principal objectif est de montrer en détails les interactions entre les objets ou les composants d'un système dans un ordre chronologique.

Selon le modèle MVC, le diagramme de séquence détaillé comporte en plus des acteurs les éléments suivants :

- **Modèle** : les modèles représentent les données du système et font la liaison avec la base de données.
- **Contrôleurs** : les contrôleurs contiennent la logique du système, ils reçoivent des requêtes de la part des utilisateurs, les traitent puis retournent des réponses.
- **Interface** : les interfaces sont les composantes permettant aux utilisateurs d’interagir avec le système.

IV.5.1.1 Diagramme de séquence détaillé du cas d’utilisation « S’authentifier »

Ci-dessous le diagramme de séquence détaillé du cas « S’authentifier » :

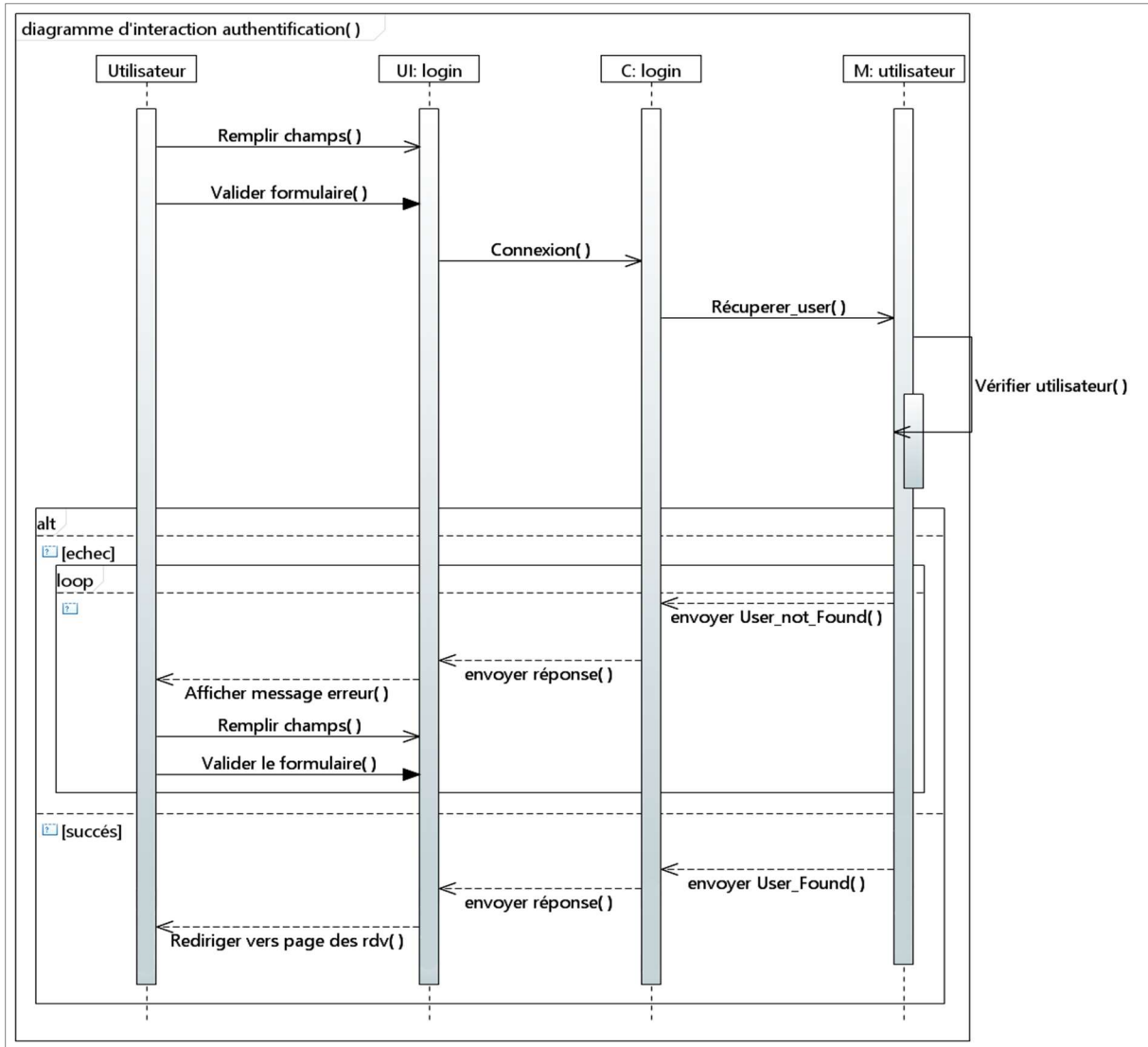


Figure 14 : Diagramme de séquence détaillé du cas S’authentifier.

IV.5.1.2 Diagramme de séquence détaillé du cas d’utilisation « Inscription »

Ci-dessous le diagramme de séquence détaillé du cas « S’authentifier » :

Chapitre IV : Sprint 1 “Gestion de rendez-vous”

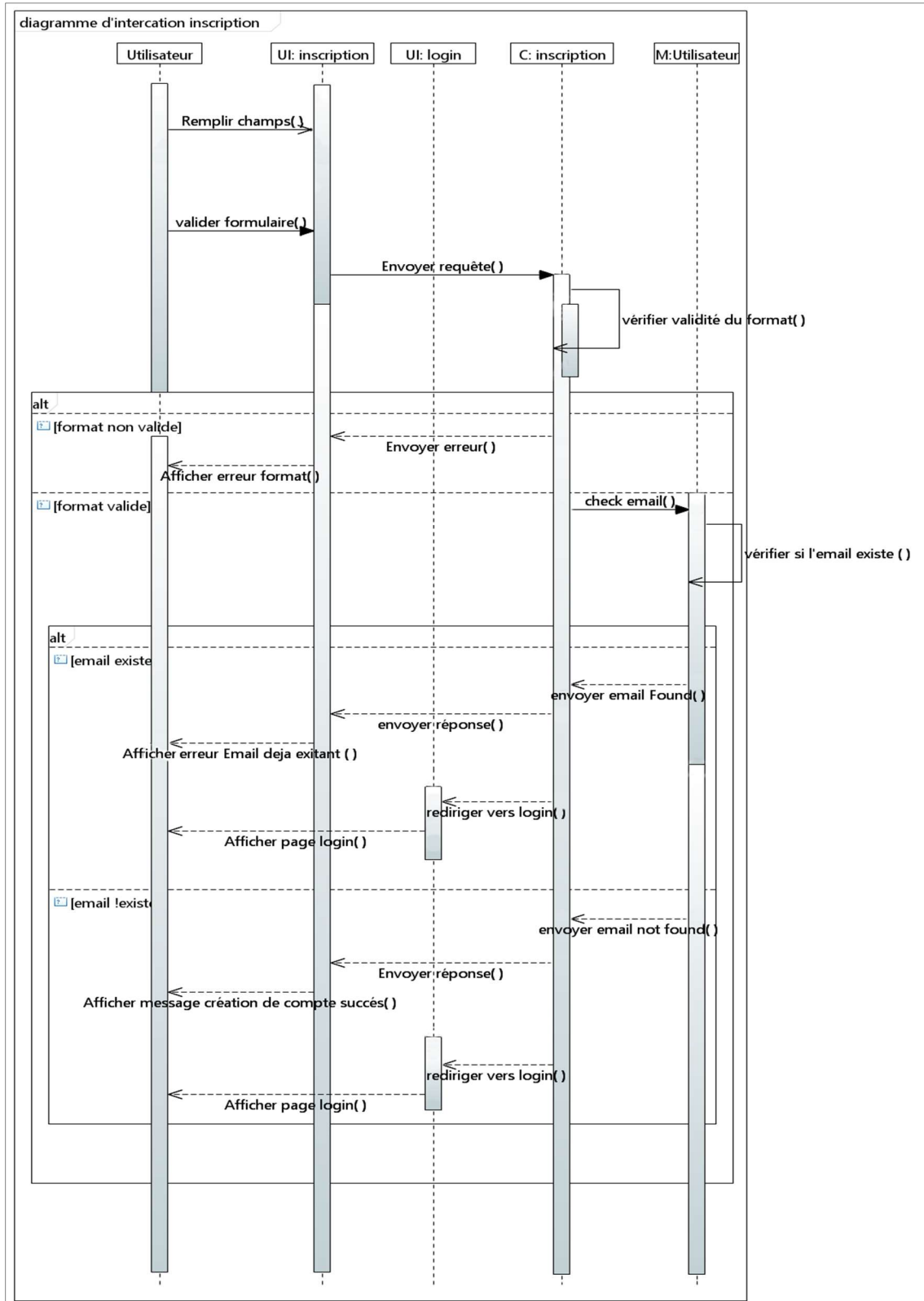


Figure 15 : Diagramme de séquence détaillé du cas S’inscrire.

IV.5.1.3 Diagramme de séquence détaillé du cas d’utilisation « Prendre un rendez-vous »

Ci-dessous le diagramme de séquence détaillé du cas « Prendre un rendez-vous » :

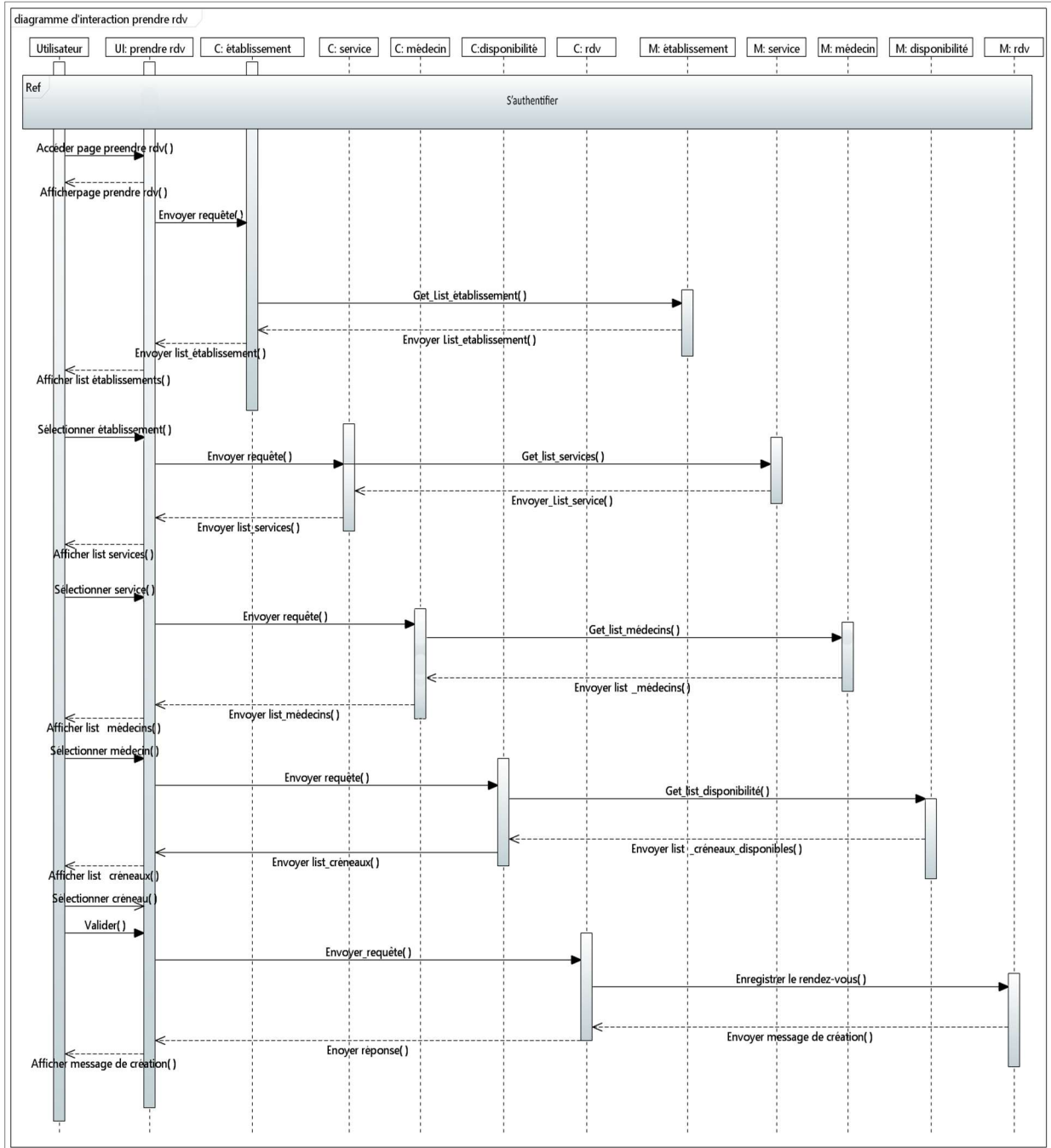


Figure 16 : Diagramme de séquence détaillé udu cas Prendre un rendez-vous.

IV.5.2 Diagramme de classes de conception du premier Sprint

Un diagramme de classe décrit les types d'objets du système et les différents types de relations statiques qui existent entre eux. Les diagrammes de classes montrent également les attributs et les méthodes d'une classe et les contraintes qui s'appliquent à la façon dont les objets sont connectés.

Chapitre IV : Sprint 1 “Gestion de rendez-vous”

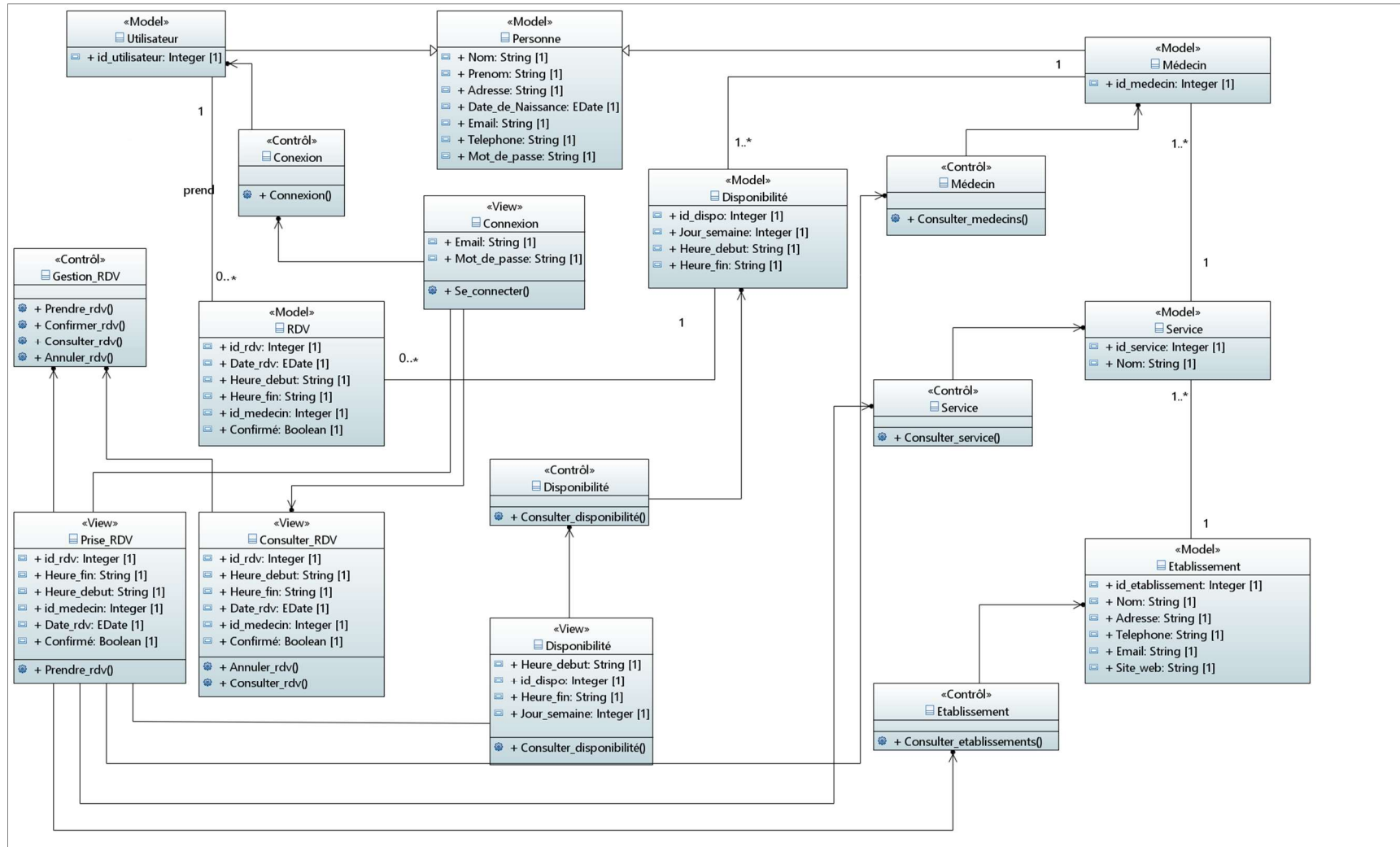


Figure 17 : Diagramme de classes de conception du premier Sprint.

IV.5.3 Modèle relationnel

Après avoir élaboré le diagramme de classes de conception, et en se basant sur celui-ci, nous pouvons construire notre modèle relationnel :

Utilisateur (id_utilisateur, nom, prenom, adresse, email, telephone, date_de_naissance, mot_de_passe)

Rdv (id_rdv, Date_rdv, Heure_debut, Heure_fin, #id_utilisateur, #id_medecin, confirmé)

Disponibilite (id_disponibilité, #id_medecin, jour_semaine, Heure_debut, Heure_fin)

Médecin (id_medecin, nom, prénom, adresse, email, telephone, date_de_naissance, mot_de_passe, #id_service)

Service (id_service, nom, #id_etablissement)

Etablissement (id_établissement, nom, adresse, email, téléphone, site web)

IV.5.4 Dictionnaire de données

Un dictionnaire de donnée se définit comme un référentiel de métadonnées qui renseigne sur le contexte d’une base de données et qui fournit les informations nécessaires pour pouvoir l’interpréter [23]. (Notre dictionnaire de données du premier Sprint est représenté dans l’annexe N01)

IV.6 Développement

La phase de développement est consacrée au développement et à l’implémentation des fonctionnalités listées dans le Sprint Backlog. C’est durant cette phase que les développeurs travaillent à transformer les spécifications et les conceptions en code exécutable en utilisant les langages de programmation et les outils nécessaires.

Dans cette partie nous allons expliquer comment nous avons fait le passage du diagramme de classe au modèle relationnel.

IV.6.1 Schéma de la Base de Données

Dans le cadre de l’implémentation de la base de données, nous avons créé un profile UML pour les bases de données relationnelles avec des stéréotypes spécifiques afin d’établir le

Chapitre IV : Sprint 1 “Gestion de rendez-vous”

lien entre les classes de conception et les tables de la base de données. La figure ci-dessous illustre l’application du profil au diagramme de classes de conception.

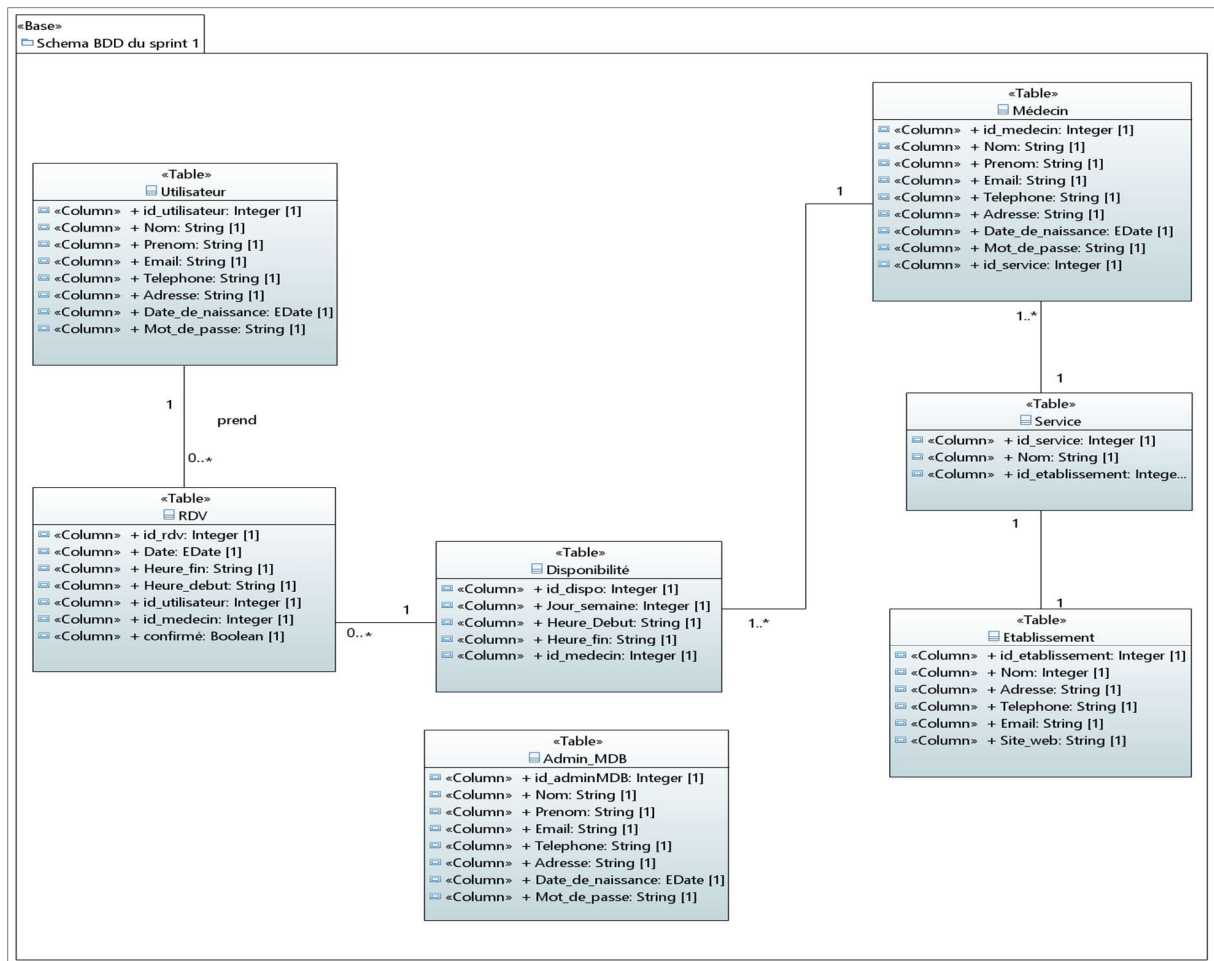


Figure 18 : Schéma de base de données du premier Sprint en utilisant un profil UML pour les BDDR.

IV.7 Test

La phase de test est la phase qui intervient directement après la phase de développement et qui a pour but de tester et évaluer les modules développés durant la phase précédente. Les tests peuvent se faire à l’aide d’outils automatisés ou manuellement en testant les différentes fonctionnalités une par une.

Durant cette phase nous nous sommes assurés que l’affichage est correct et que l’application fonctionne comme prévu et ce en effectuant des tests fonctionnels manuels. Les images ci-dessous illustrent quelques interfaces de notre application développée, démontrant ainsi que les résultats des tests sont conformes et fonctionnels.

IV.7.1 Interfaces de l’application « MyDocBooking »

Dans cette section, nous allons présenter quelques interfaces de l’application ‘MyDocBooking’ :

IV.7.1.1 Interface d’accueil

La page d’accueil est la première page de notre application.

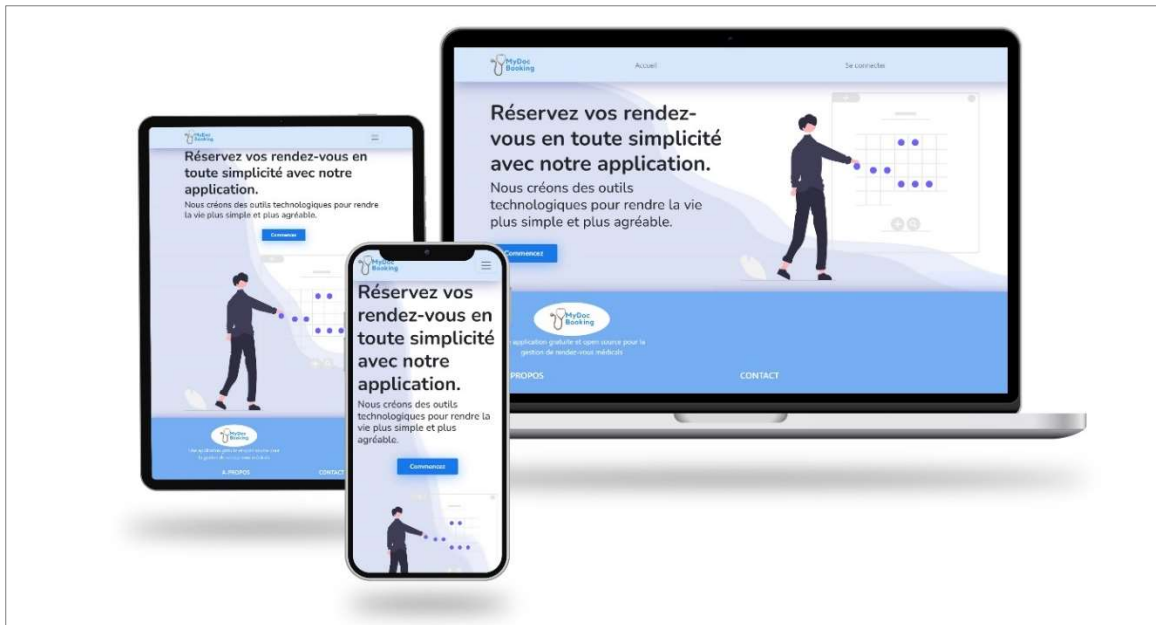


Figure 19 : Page d’accueil de l’application ‘MyDocBooking’ sur différentes tailles d’écrans.

IV.7.1.2 Interface Inscription & Authentification

L’authentification est obligatoire, donc l’utilisateur sera automatiquement redirigé vers la page d’authentification ou création de compte avant de pouvoir utiliser l’application correctement.

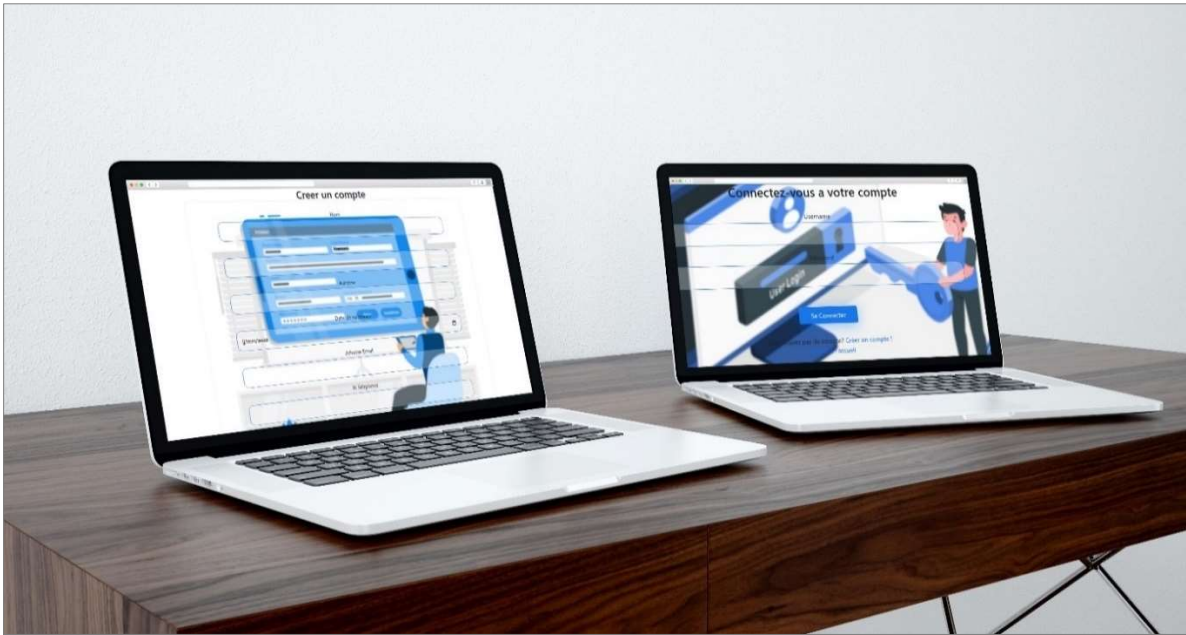


Figure 20 : Page d’inscription et page de connexion de l’application ‘MyDocBooking’.

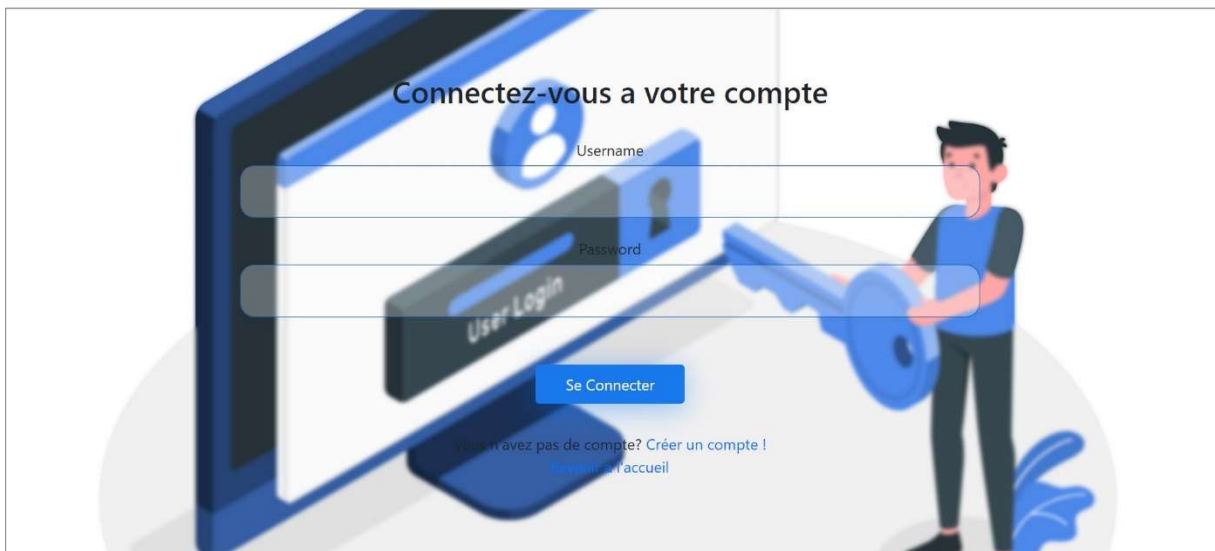


Figure 21 : Capture d’écran de la page connexion de l’application ‘MyDocBooking’.

IV.7.1.3 Interface de prise de rendez-vous

Après avoir complété l’étape d’authentification, l’utilisateur pourra alors se rendre sur la page de prise de rendez-vous, où il aura à sélectionner un créneau.

Chapitre IV : Sprint 1 “Gestion de rendez-vous”

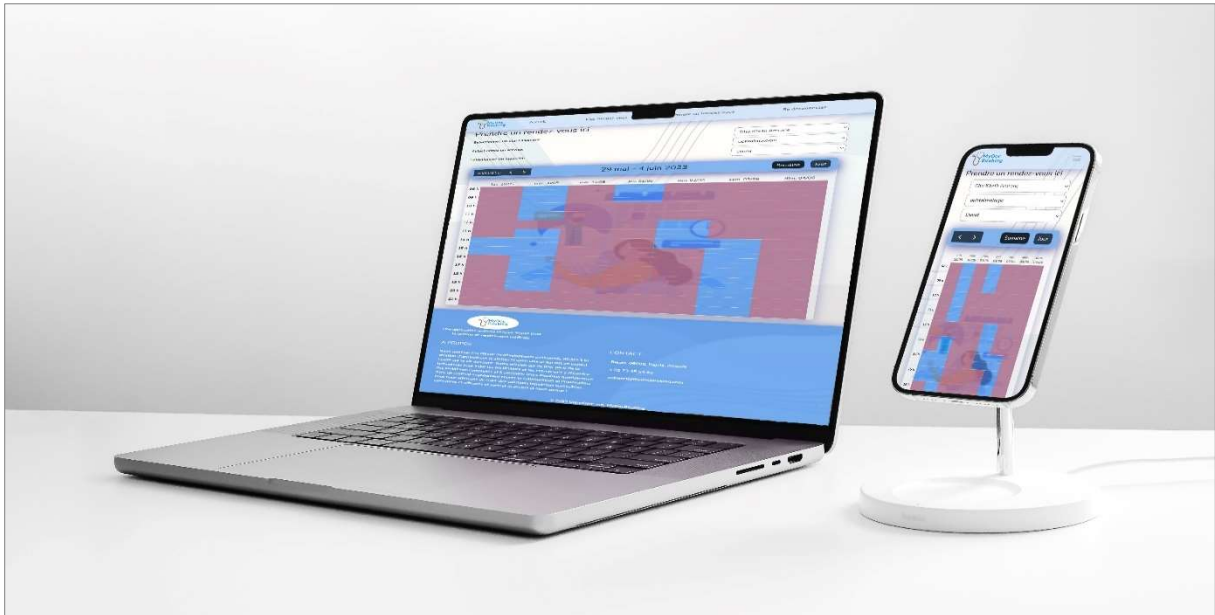


Figure 22 : Page de prise de rendez-vous de l’application ‘MyDocBooking’ sur écran pc et mobile.



Figure 23 : Capture d’écran sur mobile de la page prise de rendez-vous de l’application ‘MyDocBooking’.

IV.8 Conclusion

Nous pouvons à présent conclure ce chapitre réservé au premier Sprint, au cours duquel nous avons développé l’application ‘MyDocBooking’ et qui couvre principalement la partie

Chapitre IV : Sprint 1 “Gestion de rendez-vous”

rendez-vous. A ce niveau nous pouvons dire que nous avons atteint les objectifs précédemment fixé pour ce Sprint.

Maintenant nous allons pouvoir entamer le Sprint 2 et qui traitera la partie files d’attente ce qui fera objet du prochain et dernier chapitre.

Chapitre V : Sprint 2 ‘Gestion de files d’attente’

V.1 Introduction

Dans le chapitre quatre nous avons introduit le premier Sprint de notre projet qui a mené au développement de l’application ‘MyDocBooking’. Maintenant nous allons entamer le deuxième Sprint durant lequel nous allons traiter la partie files d’attente et flux de patients, dont l’objectif final sera de développer le sous-système ‘MedFlow’.

Ce sous-système sera constitué de trois volets, un premier consacré à la gestion des patients et à la gestion des comptes, utilisé principalement par le super admin, les admins et les secrétaires, un deuxième volet dédié aux médecins et un troisième permettant l’affichage des patients utilisé dans les salles d’attente.

Afin de réaliser notre objectif pour ce Sprint, nous avons décidé de suivre le même raisonnement que lors du précédent où la mise en œuvre des différentes User Stories nécessitait le passage par les différentes étapes de la méthode UP.

V.2 Backlog du deuxième Sprint

Voici le tableau représentant le Backlog du deuxième sprint dédié à la mise en œuvre du sous-système "MedFlow" (MF).

Acteur	ID US	User Story	ID Tâche	Tâche	Durée
Super admin de MF, Admin de MF, Secrétaire, Médecin	1	Tous les utilisateurs de MF doivent s’authentifier avant d’accéder à l’espace personnel qui leur est réservé.	1.1	Réalisation du diagramme de cas d’utilisation et la description textuelle du cas ‘s’authentifier’, et diagramme de séquence correspondant au cas.	5h
			1.2	Développer la fonctionnalité ‘s’authentifier’.	10h
			1.3	Tester la fonctionnalité ‘s’authentifier’.	2h
	2		2.1	Réalisation du diagramme de cas d’utilisation et la	

Chapitre V : Sprint 2 ‘Gestion de files d’attente’

Super admin de MF, Admin MF		L’administrateur de MF doit être capable d’ajouter, modifier, consulter et supprimer les comptes du personnel (médecins et secrétaires)		description textuelle du cas ‘ajouter établissement, et diagramme de séquence correspondant au cas.	8h	
			2.2	Développer la fonctionnalité ‘gérer les comptes du personnel’.	2h	
			2.3	Tester la fonctionnalité ‘gérer les comptes du personnel’.	2h	
	3	L’acteur doit être capable de créer, modifier, consulter et supprimer des files d’attente.	3.1	Réalisation du diagramme de cas d’utilisation et la description textuelle du cas ‘gérer les files d’attente’, et diagramme de séquence correspondant au cas.	7h	
			3.2	Développer la fonctionnalité ‘gérer les files d’attente.’	2h	
			3.3	Tester la fonctionnalité ‘ajouter Super admin’.	2h	
	4	L’administrateur MF doit être capable de consulter les statistiques de son établissement	4.1	Réalisation du diagramme de cas d’utilisation et la description textuelle du cas ‘consulter les statistiques’, et diagramme de séquence correspondant au cas.	6h	
			4.2	Développer la fonctionnalité ‘consulter les statistiques’.	7h	
			4.3	Tester la fonctionnalité ‘consulter les statistiques’.	2h	
		5		5.1	Réalisation du diagramme de cas d’utilisation et la description textuelle du cas ‘gérer les admins’, et	3h

Chapitre V : Sprint 2 ‘Gestion de files d’attente’

Super Admin MF		Le Super admin doit être capable d’ajouter, modifier, consulter et supprimer des compte des admins		diagramme de séquence correspondant au cas.	
			5.2	Développer la fonctionnalité ‘gérer les admins’.	4h
			5.3	Tester la fonctionnalité ‘gérer les admins’.	1h
Médecin	6	Le médecin doit être capable de voir l’historique des consultations qu’il a effectué	6.1	Réalisation du diagramme de cas d’utilisation et la description textuelle du cas ‘voir l’historique des consultations’, et diagramme de séquence correspondant au cas.	3h
			6.2	Développer la fonctionnalité ‘gérer les admins’.	5h
			6.3	Tester la fonctionnalité ‘gérer les admins’.	1h
	7	Le médecin doit être capable de consulter son planning	7.1	Réalisation du diagramme de cas d’utilisation et la description textuelle du cas ‘consulter planning’, et diagramme de séquence correspondant au cas.	4h
			7.2	Développer la fonctionnalité ‘consulter planning’.	2h
			7.3	Tester la fonctionnalité ‘consulter planning’.	1h
	8	Le médecin doit être capable d’appeler le prochain patient de la file	8.1	Réalisation du diagramme de cas d’utilisation et la description textuelle du cas ‘appeler prochaine patient de la file’, et diagramme de séquence correspondant au cas.	6h
			8.2	Développer la fonctionnalité ‘appeler	4h

Chapitre V : Sprint 2 ‘Gestion de files d’attente’

				prochaine patient de la file’.		
			8.3	Tester la fonctionnalité ‘appeler prochaine patient de la file’.	2h	
Secrétaire	9	La secrétaire doit être capable d’ajouter, modifier et supprimer des patients dans une file	9.1	Réalisation du diagramme de cas d’utilisation et la description textuelle du cas ‘gérer les patients dans une file’, et diagramme de séquence correspondant au cas.	4h	
			9.2	Développer la fonctionnalité ‘gérer les patients dans une file’.	5h	
			9.3	Tester la fonctionnalité ‘gérer les patients dans une file’.	2h	
	10	La secrétaire doit être capable de consulter l’ordre des patients dans une file.	10.1	Réalisation du diagramme de cas d’utilisation et la description textuelle du cas ‘consulter l’ordre des patients’, et diagramme de séquence correspondant au cas.	3h	
			10.2	Développer la fonctionnalité ‘consulter l’ordre des patients’.	2h	
			10.3	Tester la fonctionnalité ‘consulter l’ordre des patients’.	1h	
	11			11.1	Réalisation du diagramme de cas d’utilisation et la description textuelle du cas ‘consulter la liste des rendez-vous’, et diagramme de séquence correspondant au cas.	4h

Chapitre V : Sprint 2 ‘Gestion de files d’attente’

		La secrétaire doit être capable de consulter la liste des rendez-vous pour la journée	11.2	Développer la fonctionnalité ‘consulter la liste des rendez-vous’.	3h
			11.3	Tester la fonctionnalité ‘consulter la liste des rendez-vous’.	1h

Tableau 13 : Tableau représentant le Backlog du deuxième Sprint.

V.3 Spécifications fonctionnelles

Durant ce Sprint nous allons voir les fonctionnalités offertes par le sous-système ‘MedFlow’.

Ce sous-système est divisé en trois parties, on trouve une application Desktop dédiée aux secrétaires et aux administrateurs, une application mobile pour les médecins et une application fonctionnant sur les écrans permettant d’afficher les patients.

Ci-dessous l’ensemble des cas d’utilisation relatifs à ce sous-système :

- L’ensemble des cas d’utilisation incluent le cas « S’authentifier », ainsi le médecin, la secrétaire ou l’admin doivent d’abord passer par l’authentification avant de pouvoir accéder à l’espace qui leur est réservé.
- L’ensemble des admins ainsi que le super admin peuvent ajouter, modifier, consulter et supprimer les comptes des médecins et secrétaires et consulter les statistiques de l’établissement.
- Seul le super admin a l’autorisation de gérer les admins (ajouter, modifier, consulter et supprimer).
- La secrétaire pourra ajouter un patient à la file d’attente correspondante.
- Le médecin doit pouvoir appeler un patient et le système se chargera d’afficher les informations de celui-ci sur l’écran.

V.3.1 Diagramme de cas d’utilisation

La figure ci-dessous présente le diagramme de cas d’utilisation propre au Sprint 2 :

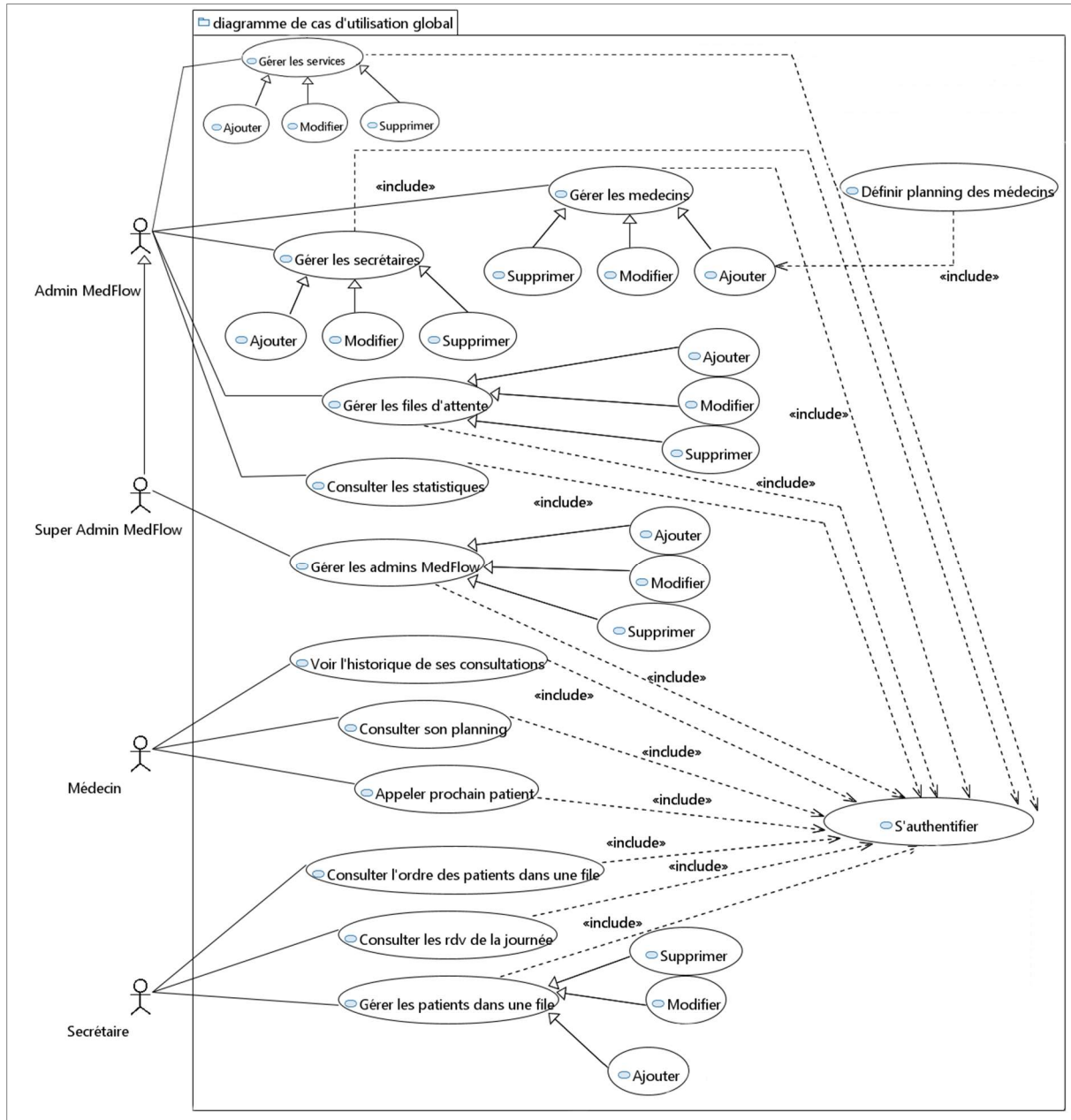


Figure 24 : Diagramme de cas d’utilisation du deuxième Sprint.

V.3.2 Description textuelle des cas d’utilisation

Dans cette section nous allons décrire les cas d’utilisation les plus pertinents à savoir le cas « Appeler le prochain patient », « Ajouter un patient à une file ».

V.3.3.1 Cas d’utilisation « Appeler prochain patient »

Le tableau ci-dessous décrit le cas d’utilisation « appeler prochain patient » :

Chapitre V : Sprint 2 ‘Gestion de files d’attente’

Sommaire	
Titre	Appeler prochain patient.
But	Permet au médecin d’appeler le prochain patient depuis l’application mobile.
Acteur	Médecin
Pré condition	<ul style="list-style-type: none"> Le médecin est authentifié L’appareil du médecin doit être connecté au réseau local de l’établissement.
Scenario	<ul style="list-style-type: none"> Le système affiche l’interface permettant d’appeler un patient. Le médecin appuie sur le bouton ‘patient suivant’. Le système envoie une requête pour récupérer la liste des patients en suivant l’ordre descendant des priorité des files. Le système récupère le premier patient dans la file avec la priorité la plus importante.
Post condition	<ul style="list-style-type: none"> Le médecin doit avoir à sa disposition les différentes informations du patient appelé. Les écrans d’affichage indiquent le tour du patient actuel.
Alternative	<p>A1 : Le système ne trouve pas de patient dans la file avec la priorité la plus importante et passe à une file avec une priorité moins importante.</p> <p>A2 : Le système ne trouve aucun patient dans toutes les files du service et affiche au médecin le message ‘pas patient actuellement’.</p>
Besoin d’IHM	<ul style="list-style-type: none"> Interface d’appel de patient sur l’application mobile du médecin. Interface d’affichage sur l’écran d’affichage.

Tableau 14 : Description textuelle du cas d’utilisation « Appeler prochain patient ».

V.3.3.2 Cas d’utilisation « Ajouter un patient »

Le tableau ci-dessous décrit le cas d’utilisation « Ajouter un patient » :

Sommaire	
Titre	Ajouter un patient à une file.
But	Permet à la secrétaire d’ajouter un patient à une file.
Acteur	Secrétaire
Pré condition	<ul style="list-style-type: none"> La secrétaire est authentifiée.
Scenario	<ul style="list-style-type: none"> Le système affiche l’interface permettant d’ajouter un nouveau patient à une file d’attente. La secrétaire remplit et valide le formulaire. Le système envoie une requête contenant les informations du patient et l’ajoute à la file correspondante.
Post condition	<ul style="list-style-type: none"> Le patient est ajouté à la file.
Alternative	/
Besoin d’IHM	<ul style="list-style-type: none"> Interface d’ajout de patient.

Tableau 15 : Description textuelle du cas d’utilisation « Ajouter un patient à la file ».

V.4 Analyse

Cette phase d’analyse va nous permettre de décrire avec précision ce que le système va offrir, et nous allons commencer par définir le modèle du domaine de ce deuxième Sprint suivi par les diagrammes de séquence des différents cas.

V.4.1 Modèle du domaine

Ci-dessous le modèle du domaine du deuxième Sprint et qui modélise de manière globale les différentes entités faisant partie du sous-système ‘MedFlow’.

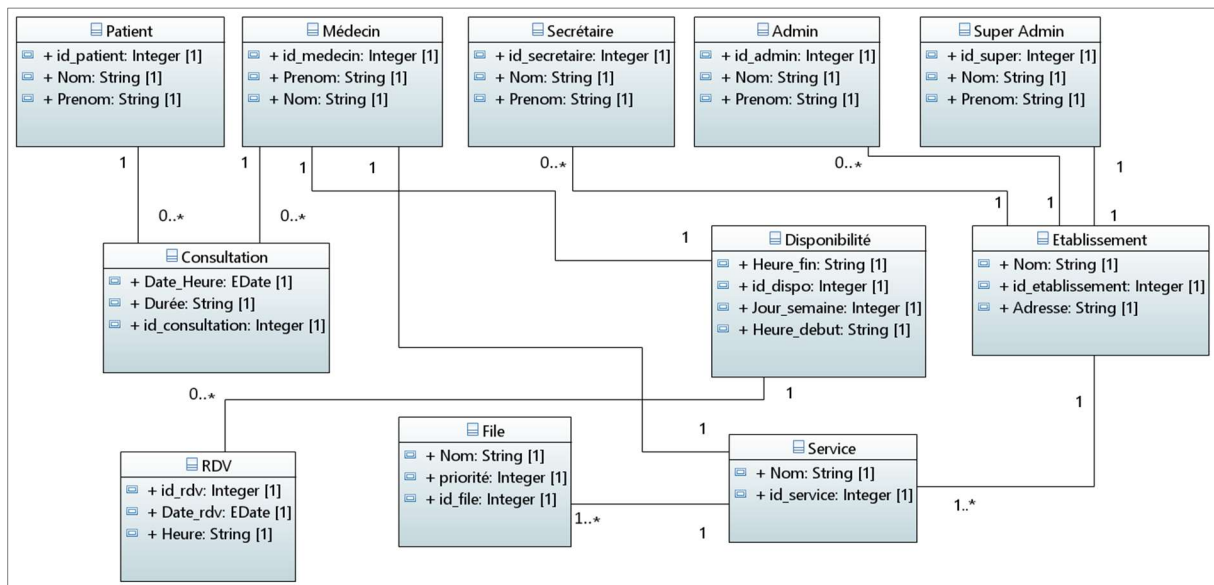


Figure 25 : Modèle du domaine deuxième Sprint.

V.4.2 Diagrammes de séquence système

Le diagramme de séquence système est un type du diagramme de séquence. Il est utilisé pour modéliser les interactions séquentielles entre les acteurs externes et le système.

Ci-dessous les diagrammes de séquence système de quelques a-savoir « Créer un compte admin, secrétaire, médecin », « Ajouter un patient à la file », « Appeler prochain patient ».

V.4.2.1 Diagramme de séquence système du cas « S’authentifier »

Ci-dessous le diagramme de séquence système du cas « Créer compte admin, secrétaire ou médecin » :

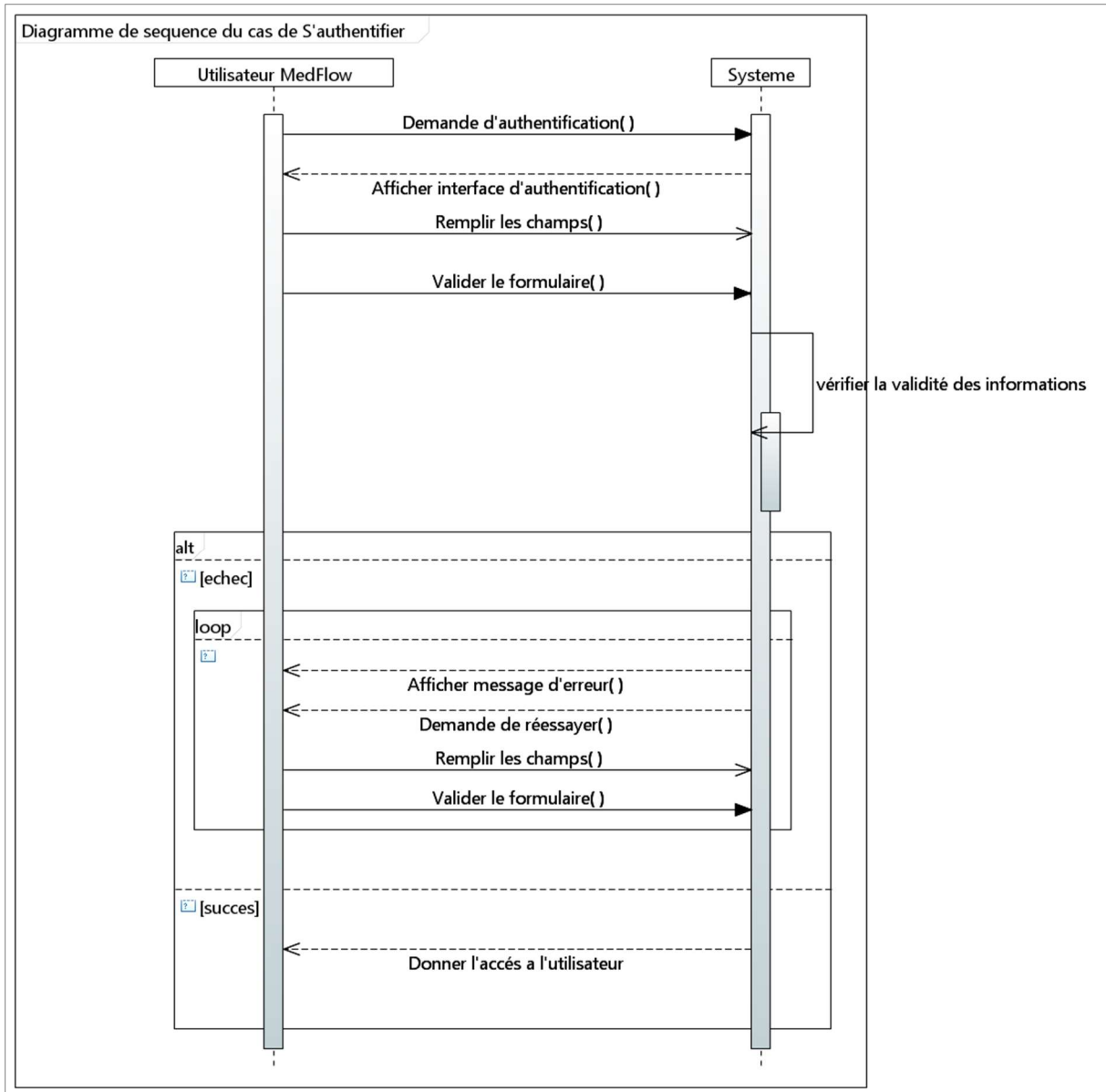


Figure 26 : Diagramme de séquence système du cas « S'authentifier ».

V.4.2.2 Diagramme de séquence système du cas « Créer compte personnel »

Ci-dessous le diagramme de séquence système du cas « Créer compte admin, secrétaire ou médecin » :

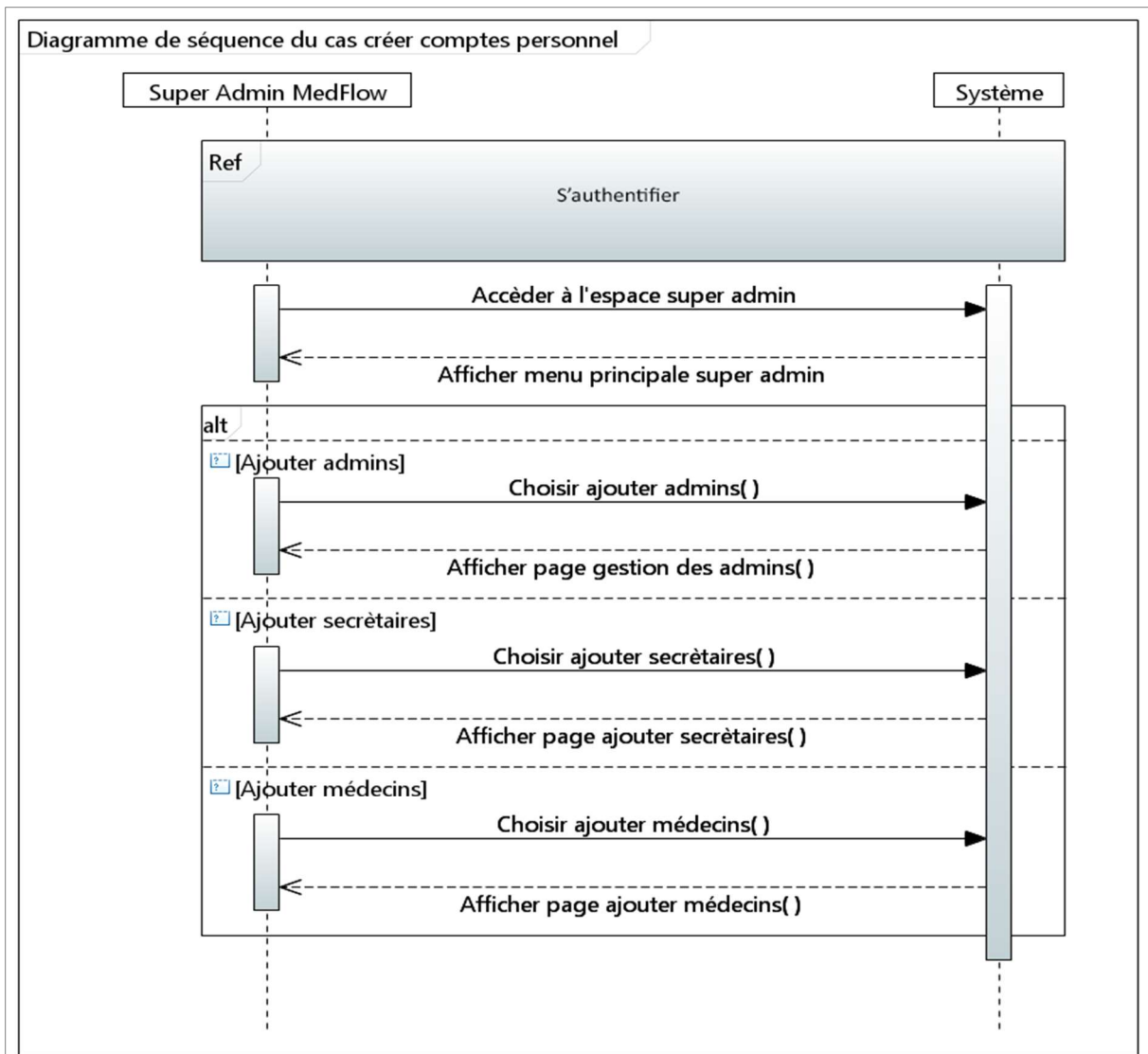


Figure 27 : Diagramme de séquence système du cas « Créer compte personnel ».

V.4.2.3 Diagramme de séquence système du cas « Ajouter un patient à la file »

Ci-dessous le diagramme de séquence système du cas « Ajouter un patient à la file » :

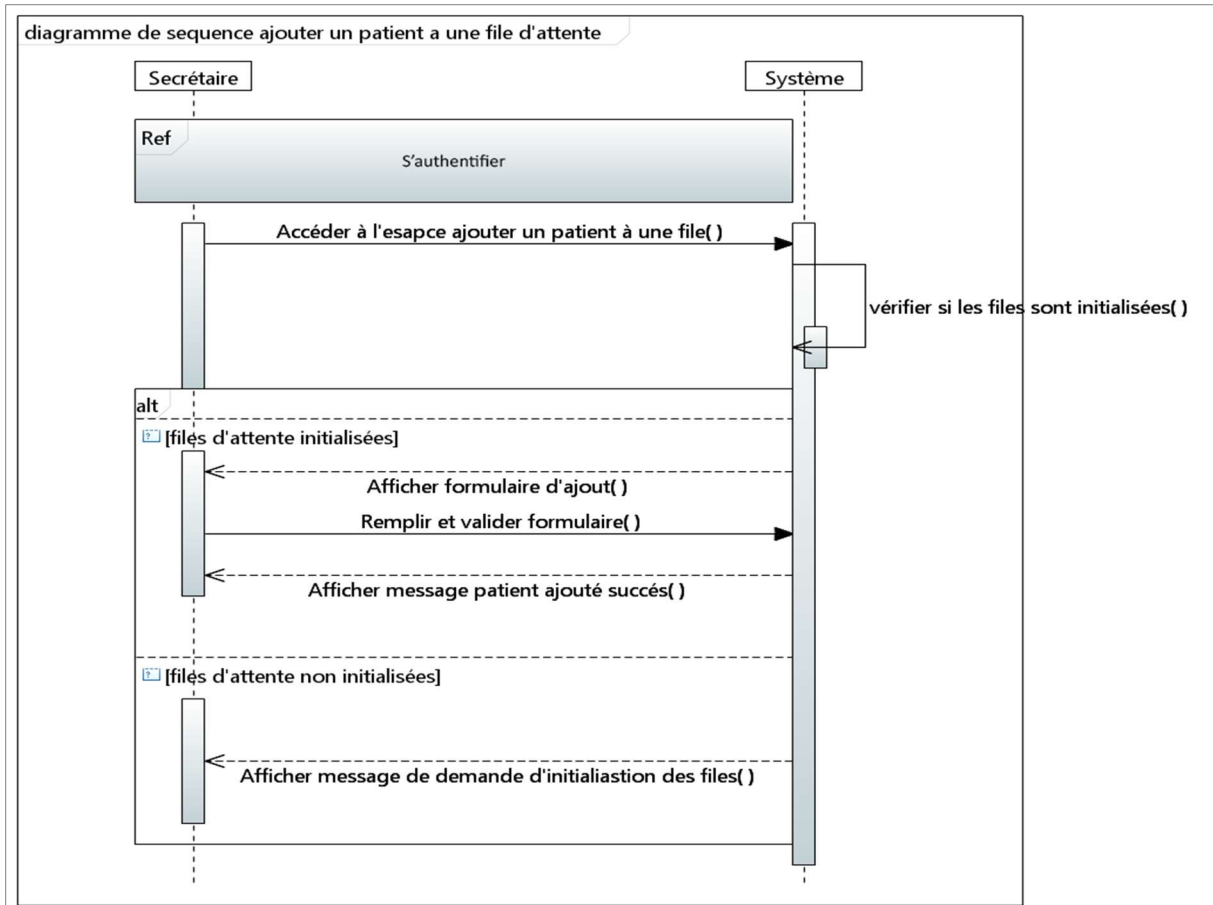


Figure 28 : Diagramme de séquence système du cas « Ajouter patient à une file ».

V.4.2.4 Diagramme de séquence système du cas « Appeler prochain patient »

Ci-dessous le diagramme de séquence système du cas « Appeler prochaine patient de la file » :

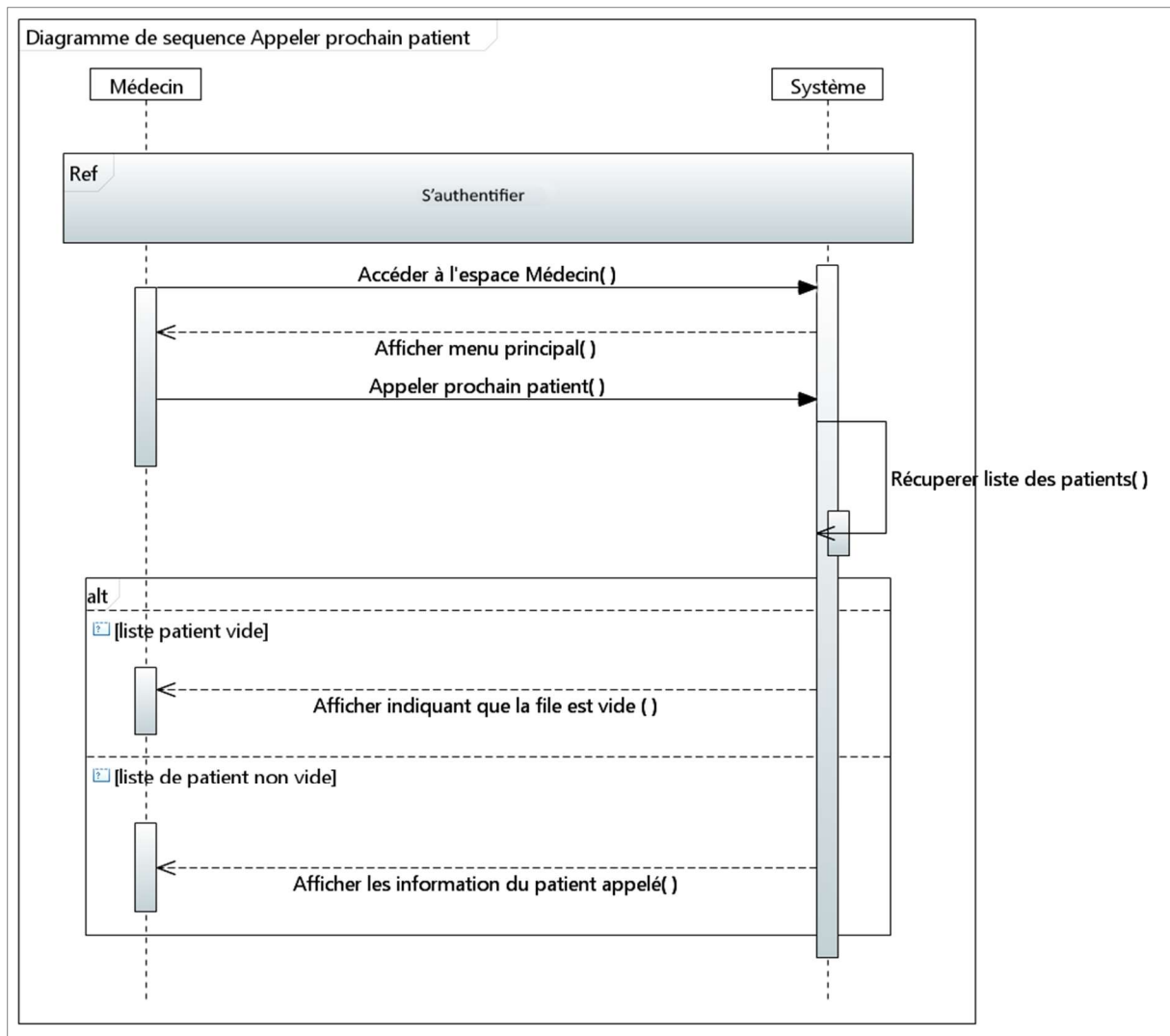


Figure 29 : Diagramme de séquence système du cas « Appeler prochain patient ».

V.5 Conception

Dans cette section, nous allons procéder à la création des diagrammes de séquence détaillé correspondant à chaque cas d'utilisation, puis nous présenterons le diagramme de classe du deuxième sprint.

V.5.1 Diagrammes de séquence détaillé

Le diagramme de séquence détaillé permet de décrire comment des groupes d'objets collaborent dans une activité donnée, et toujours en suivant le modèle MVC, le diagramme sera alors composé du model, contrôleur et interface.

V.5.1.1 Diagramme de séquence détaillé du cas « Appeler prochain patient de la file »

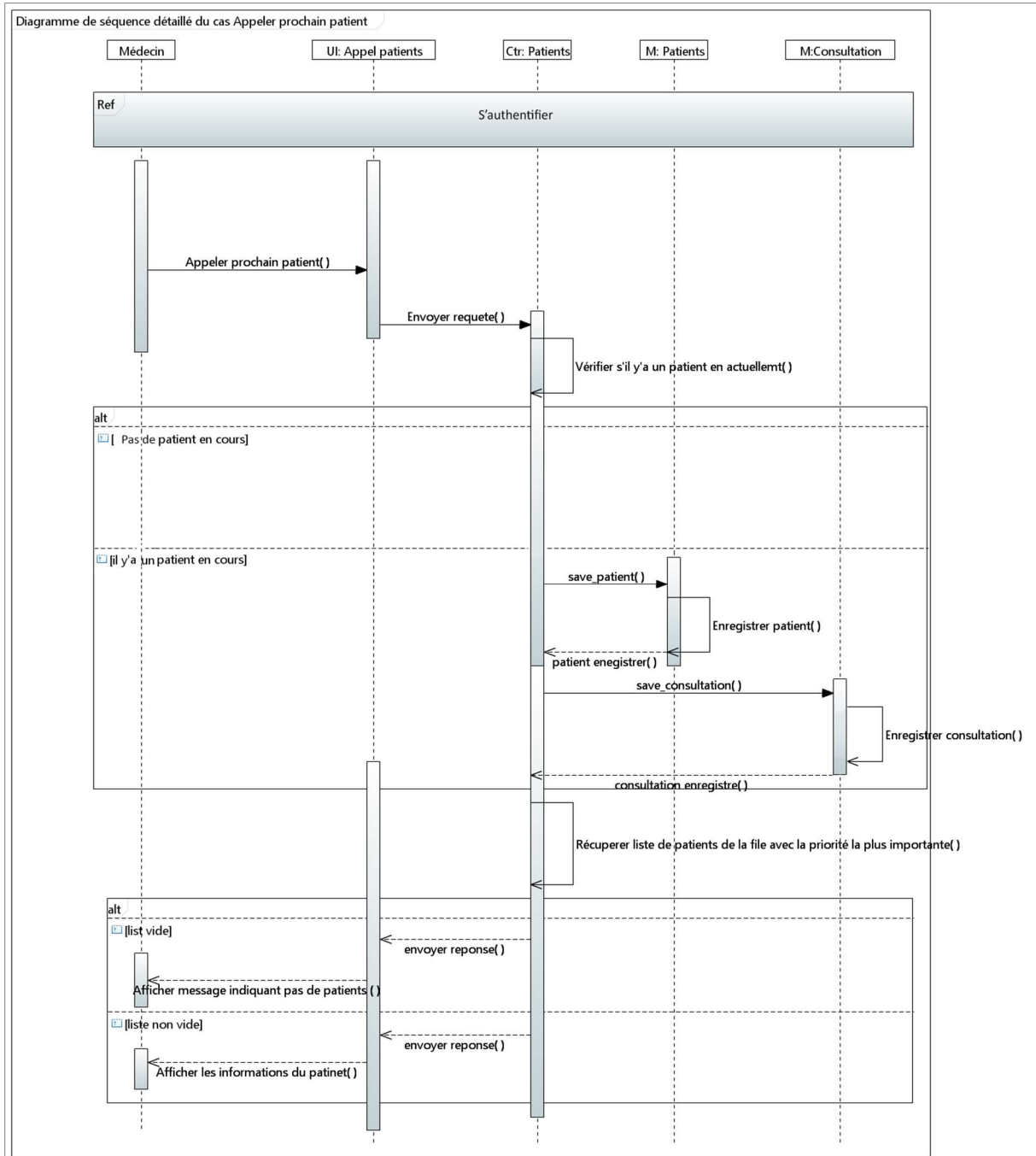


Figure 30 : Diagramme de séquence détaillé du cas « Appeler prochain patient ».

V.5.2 Diagramme de classes de conception

Ci-dessous le diagramme de classes de conception du deuxième Sprint :

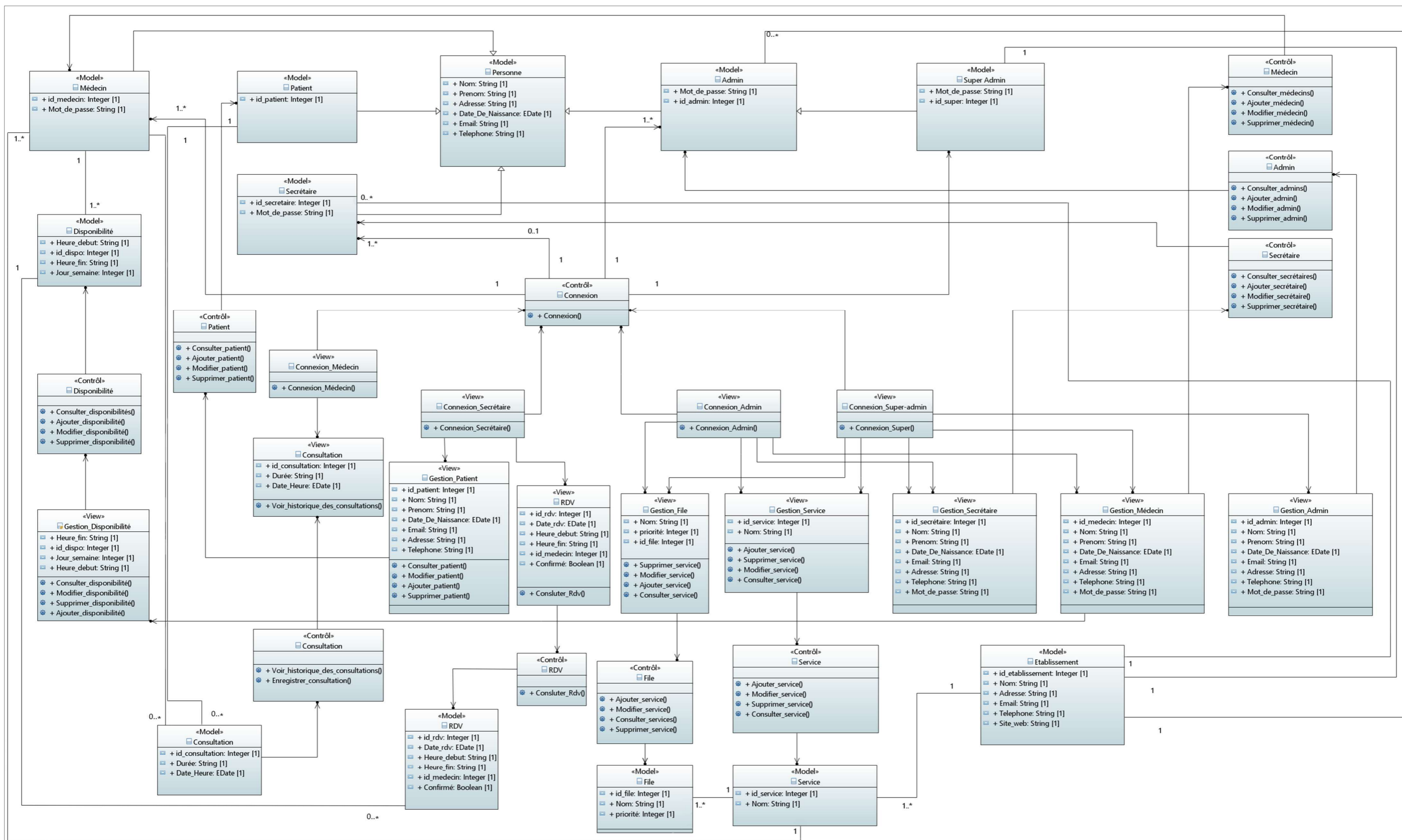


Figure 31 : Diagramme de classes de conception du deuxième Sprint.

V.5.3 Modèle relationnel

Nous Allons à présent passer à la construction du modèle relationnel en se basant sur le diagramme de classe donné précédemment :

Rdv (id_rdv, Date_rdv, Heure_debut, Heure_fin, #id_utilisateur, #id_medecin, confirmé)

Disponibilite (id_disponibilité, #id_medecin, jour_semaine, Heure_debut, Heure_fin)

Médecin (id_medecin, nom, prénom, adresse, email, telephone, date_de_naissance, mot_de_passe, #id_service)

Secrétaire (id_secretaire, nom, prénom, adresse, email, telephone, date_de_naissance, mot_de_passe, #id_etablissement)

Admin (id_admin, nom, prénom, adresse, email, telephone, date_de_naissance, mot_de_passe, #id_etablissement)

Super_admin (id_Super, nom, prénom, adresse, email, telephone, date_de_naissance, mot_de_passe, #id_etablissement)

Patient (id_patient, nom, prénom, adresse, email, telephone, date_de_naissance)

File (id_file, nom, priorite, id_service)

Consultation (id_consultation, Date_Heure, Duree, #id_patient, #id_medecin)

Service (id_service, nom, #id_etablissement)

Etablissement (id_établissement, nom, adresse, email, téléphone, site web)

V.5.4 Dictionnaire de données

(Le dictionnaire de données du deuxième Sprint est représenté dans l’annexe N02)

V.6 Développement

Maintenant nous pouvons passer à la phase de développement, qui concerne l’implémentation et la concrétisation des fonctionnalités précédemment listées.

Pour commencer nous allons d’abord présenter ci-dessous le schéma de la base de données de ce deuxième Sprint.

V.6.1 Schéma de base de données

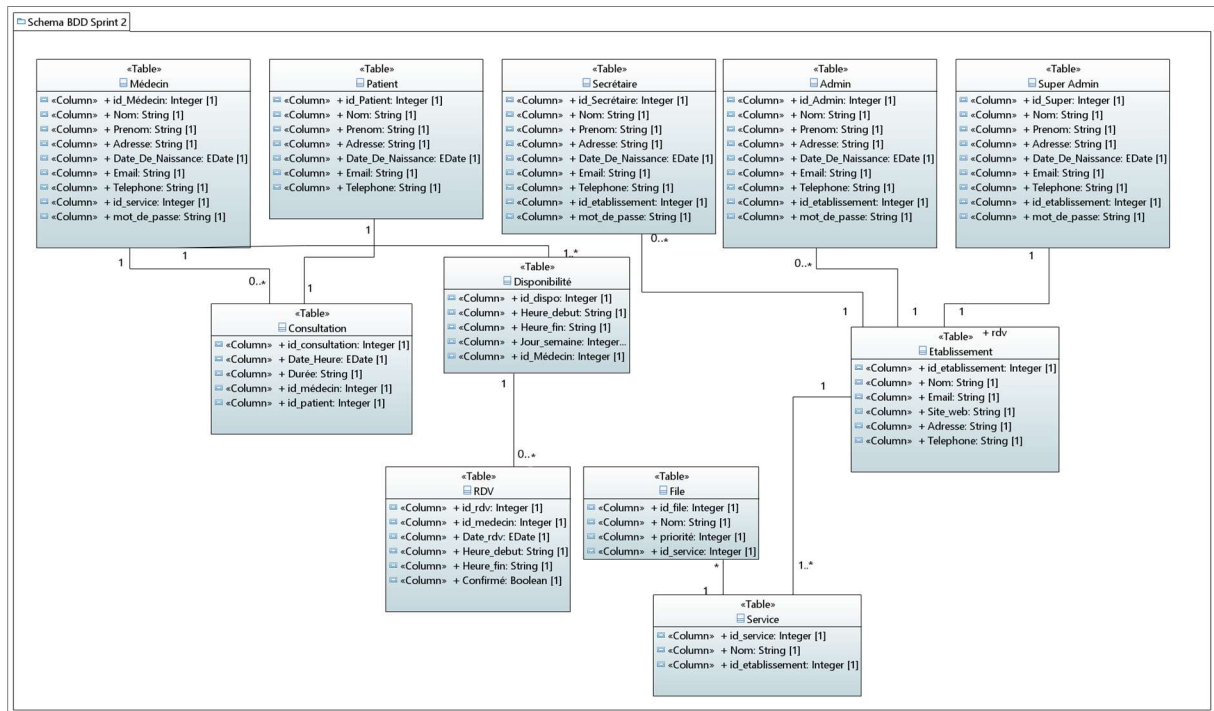


Figure 32 : Schéma de base de données du deuxième Sprint en utilisant un profil UML pour les BDDR.

V.7 Test

Après la phase de développement nous avons besoins de passer en revue tous ce qui a été développé précédemment vérifier que le système fonctionne correctement et détecter toute anomalie.

V.7.1 Interfaces de l’application « MedFlow »

Dans cette section, nous allons présenter quelques interfaces de l’application ‘MedFlow’ :

V.7.1.1 Interface d’accueil de « MedFlow » pour médecin

Ci-dessous les interfaces d’accueil et du menu principal de l’application ‘MedFlow’ pour médecins :



Figure 33 : Page d’accueil et menu principale de l’application ‘MedFlow’ pour médecins.

V.7.1.2 Interface « MedFlow » pour secrétaire

Ci-dessous les interfaces du menu principal de l’application ‘MedFlow’ pour secrétaire :

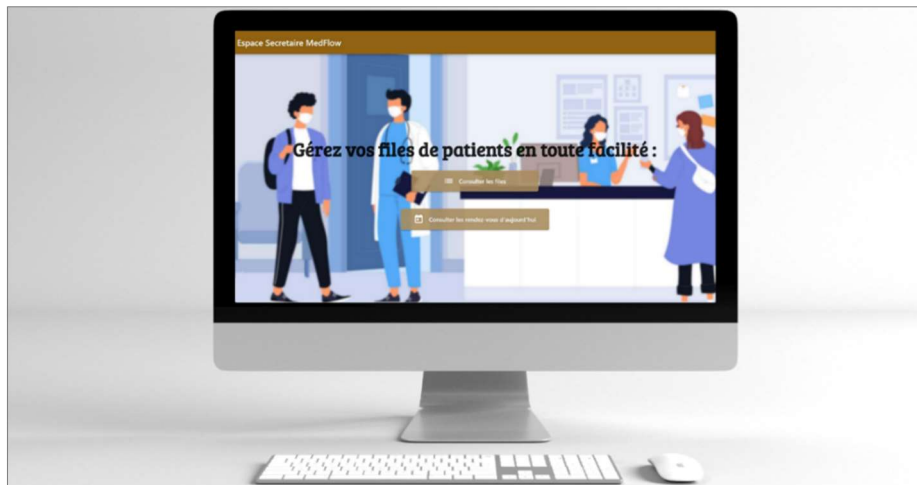


Figure 34 : Menu principal de l’espace secrétaire.

V.8 Conclusion

Nous avons achevé avec succès notre deuxième et dernier Sprint et nous avons ainsi obtenue une application finie et prête à être livrée.

Conclusion Générale

Au cours de ce projet, nous avons conçu et réaliser une application de gestion de rendez-vous ainsi qu'une application de gestion de files d'attente destinées aux établissements de santé.

L'étude a permis de mettre en évidence les défis spécifiques auxquels les établissements de santé sont confrontés en termes de gestion de rendez-vous et de gestion de files d'attente des patients, à la fois grâce aux informations collectées à 'l'Eph Amizour' où l'étude a été faite. Nous avons donc pu à travers celle-ci récolter des informations très utiles notamment par rapport aux processus de gestion de patients, aux différents services de l'établissement et aux moyens matériels et logiciels disponibles dans celui-ci, et aussi des conseils et du suivi dont on a pu bénéficier dans l'entreprise de développement 'D-Soft' où le stage a été effectué.

Ensuite nous sommes passé au choix de méthodologie à adopter et à travers l'analyse des différentes approches et méthodes existantes, nous avons décidé qu'une approche intégrée, combinant des éléments de Scrum et d'UP, peut-être une très bonne meilleure solution.

L'utilisation de Scrum dans la conduite du projet nous a permis de planifier et de suivre de manière agile les activités liées à la gestion des files d'attente et des rendez-vous où nous avons à découper le projet en Sprints, élaborer le PB, puis planifier ces Sprints. D'autre part, l'utilisation de l'approche UP a facilité la spécification des besoins, l'analyse, la conception, l'implémentation et les tests des solutions et ce durant chacun des Sprints.

Ainsi, le système que nous avons développé au sein de l'entreprise d'accueil D-Soft permettra de faciliter la prise de rendez-vous et la gestion du flux de patients de manière générale aux seins des établissements de santé.

Ce projet a fait l'objet d'une expérience intéressante, très bénéfique pour nous. Il nous a offert l'opportunité d'approfondir nos connaissances théoriques et techniques, nous avons également pu mettre en pratique toutes les connaissances que nous avons acquises tout au long des études que nous avons faites.

Enfin et comme perspective, l'utilisation d'une approche IA pour le choix de la politique d'ordonnancement des patients en fonctions des données statistiques recueillies pourrait améliorer considérablement notre solution.

Références bibliographiques

Références bibliographiques

- [1]: Shortle J.F, Thompson J.M, Gross D, Harris C.M, FUNDAMENTALS OF QUEUEING THEORY (FIFTH EDITION). John Wiley and Sons, 2018.
- [2]: Sztrik J, Basic Queueing Theory. GlobeEdit, 2016.
- [3]: Kleinrock L, QUEUEING SYSTEMS VOLUME I: THEORY. John Wiley and Sons, 1975.
- [4]: Pine B.J, Gilmore J.H, The Experience Economy. Harvard Business School, 1999.
- [5]: BOUCHEMAL A, Tiab A. Les conditions de travail des sage-femmes dans la prise en charge des mères en accouchement à l'EPH d'Amizour- Bejaia, Mémoire de master en sociologie, Université de Béjaia - Algérie. 2016.
- [6]: Gestion des services hospitaliers : Constat accablant et série de sanctions, <https://www.reporters.dz/gestion-des-services-hospitaliers-constat-accablant-et-serie-de-sanctions/>, consulté le 21 avril 2023.
- [7]: The Scrum Framework, <https://shop.theliberators.com/>, consulté le 20 avril 2023.
- [8]: Les artefacts Scrum, 3 piliers pour mener un sprint efficace en équipe, <https://www.appvizer.fr/magazine/operations/gestion-de-projet/artefacts-scrum>, consulté le 10 Mai 2023.
- [9]: Abbaci N, Alitouche L. Conception et Réalisation d'une application de gestion administrative De la SONATRACH-Bejaia, Mémoire de master en génie logiciel, Université de Béjaia - Algérie. 2018.
- [10]: Schwaber K, Sutherland J, The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game, 2020.
- [11]: Kruchten P, Rational Unified Process—An Introduction, Addison-Wesley, 2003.
- [12]: Fowler M, Uml Distilled: Applying the Standard Object Modeling Language , Addison-Wesley, 1997.
- [13]: Processus unifié : Modèle En Y, <https://slideplayer.fr/slide/482289/> consulté le 25 avril 2023.
- [14]: Marcus H, Joakim S, Kanban in Action, Manning Publications, 2014.
- [15]: Anderson D.J, Reinertsen D.G, Kanban: Successful Evolutionary Change for Your Technology Business, Blue Book ed, 2010.
- [16]: Benchouia M. Gestion automatique du service Des urgences médicales, Mémoire de master en génie logiciel, Université de Biskra - Algérie. 2021.
- [17]: Ait Slimane A, Haddadou G. Complémentarité entre le secteur public et privé de soins en Algérie, Université de TIZI-OUZOU - Algérie. 2021.
- [18]: Green M.D, Scrum: Novice to Ninja: Methods for Agile, Powerful Development 1st Edition, SitePoint, 2016.
- [19]: Sutherland J, Scrum: The Art of Doing Twice the Work in Half the Time, Currency, 2014.

Références bibliographiques

- [20]: Fowler M, UML Distilled: A Brief Guide to the Standard Object Modeling Language 3rd Edition, Addison-Wesley Professional, 2003.
- [21]: Papyrus, model-based system engineering platform, <https://list.cea.fr/en/page/papyrus-outil-open-source-de-modelisation-graphique-uml/>, consulté le 26 avril 2023.
- [22] : Mise en œuvre d'UML, <https://laurent-audibert.developpez.com/CoursUML/?page=mise-en-oeuvre-uml>, consulté le 27 avril 2023.
- [23] : Qu'est-ce qu'un dictionnaire de données ? <https://datavalue-consulting.com/dictionnaire-donnees>, consulté le 15 juin 2023.
- [24] : Scrum for one : A tutorial on adapting Agile Scrum methodology for individuals, <https://www.lucidchart.com/blog/scrum-for-one>, consulté le 25 avril 2023.
- [25] : Adjtoutah S, Ait Radi T, Chelouche W, Ghanem S, Hamdaoui F, Hamou Aldja K, Iabbassen F, Igguï T, Mechrafi F. Conception et réalisation d'un site web dynamique dédié à la gestion et au suivi des patients. Cas d'étude « Etablissement Public Hospitalier d'AKBOU. Bejaia », Mémoire de licence en informatique générale, Université de Béjaia - Algérie. 2018.
- [26] : Belbachir A, Bouchrit N, Haddad T, Kessassi C. REALISATION D'UNE APPLICATION WEB DE GESTION DE DOCUMENTS POUR Tchén-Lait. Bejaia, Mémoire de licence en informatique générale, Université de Béjaia - Algérie. 2017.
- [27] : Présentation de l'établissement Public Hospitalier Amizour : <http://www.dsp-bejaia.dz/images/PDF/eph%20amizour.pdf>, consulté le 15 juin 2023.
- [28] : Apprendre à coder en HTML et CSS | Cours complet (2020) : <https://www.pierre-giraud.com/html-css-apprendre-coder-cours/>, consulté le 17 juin 2023.
- [29] : Apprendre à coder en Javascript | Cours complet (2020) : <https://www.pierre-giraud.com/javascript-apprendre-coder-cours/>, consulté le 17 juin 2023.
- [30] : Apprendre le SQL : <https://sql.sh/>, consulté le 17 juin 2023.
- [31] : Flutter VS Dart lequel est le meilleur ? : <https://blog.back4app.com/fr/flutter-vs-dart/>, consulté le 17 juin 2023.
- [32] : Next.js vs React? It's a Partnership, not a Competition: <https://kinsta.com/blog/nextjs-vs-react/>, consulté le 17 juin 2023.
- [33] : What is Sequelize : <https://www.educative.io/answers/what-is-sequelizejs>, consulté le 17 juin 2023.

-Annexe 01-

Le tableau ci-dessous représente le dictionnaire de données du premier Sprint :

Classe	Attribut	Signification	Type	Taille
Utilisateur	Id_utilisateur	Identifiant utilisateur	INT	11
	Nom	Nom	VARCHAR	50
	Prenom	Prénom	VARCHAR	50
	Adresse	Adresse	VARCHAR	255
	Email	Adresse Email	VARCHAR	50
	Telephone	Numéro de téléphone	VARCHAR	15
	Date_de_naissance	Date de naissance	DATE	10
	Mot_de_passe	Mot de passe	VARCHAR	255
RDV	Id_rdv	Identifiant rendez-vous	INT	11
	Date_rdv	Date rendez-vous.	DATE	10
	Heure_debut	Début d'intervalle du rendez-vous.	VARCHAR	10
	Heure_fin	Fin d'intervalle du rendez-vous.	VARCHAR	10
	Id_utilisateur	Identifiant de l'utilisateur ayant pris le rendez-vous.	INT	11
	Id_medecin	Identifiant du médecin	INT	11
	Confirmé	Indique si le rendez-vous est confirmé.	BOOLEEN	1
Disponibilité	Id_disponibilité	Identifiant de la disponibilité	INT	11
	Id_medecin	Identifiant du médecin ayant cette disponibilité.	INT	11
	Jour_semaine	Jour de la semaine.	INT	1
	Heure_debut	Heure début de service.	VARCHAR	10
	Heure_fin	Heure fin de service.	VARCHAR	10
Médecin	Id_medecin	Identifiant du medecin	INT	11
	Nom	Nom	VARCHAR	50
	Prenom	Prénom	VARCHAR	50
	Adresse	Adresse	VARCHAR	255
	Email	Adresse Email	VARCHAR	50
	Telephone	Numéro de téléphone	VARCHAR	15
	Date_de_naissance	Date de naissance	DATE	10

Annexes

	Mot_de_passe	Mot de passe	VARCHAR	255
	Id_service	Identifiant du service dans lequel il est affecté.	INT	11
Service	Id_service	Identifiant du service.	INT	11
	Nom	Nom du service	VARCHAR	50
	Id_etablissement	Identifiant de l'établissement dans lequel fait partie le service.	INT	11
Etablissement	Id_etablissement	Identifiant de l'établissement.	INT	11
	Nom	Nom de l'établissement	VARCHAR	50
	Adresse	Adresse de l'établissement.	VARCHAR	255
	Email	Adresse Email de l'établissement.	VARCHAR	50
	Telephone	Numéro de contacte l'établissement	VARCHAR	15
	Site Web	Site internet	VARCHAR	50

-Annexe 02-

Ci-dessous le dictionnaire de données du deuxième Sprint :

Classe	Attribut	Signification	Type	Taille
RDV	Id_rdv	Identifiant rendez-vous	INT	11
	Date_rdv	Date rendez-vous.	DATE	10
	Heure_debut	Début d'intervalle du rendez-vous.	VARCHAR	10
	Heure_fin	Fin d'intervalle du rendez-vous.	VARCHAR	10
	Id_utilisateur	Identifiant de l'utilisateur ayant pris le rendez-vous.	INT	11
	Id_medecin	Identifiant du médecin	INT	11
	Confirmé	Indique si le rendez-vous est confirmé.	BOOLEEN	1
Disponibilite	Id_disponibilité	Identifiant de la disponibilité	INT	11
	Id_medecin	Identifiant du médecin ayant cette disponibilité.	INT	11
	Jour_semaine	Jour de la semaine.	INT	1
	Heure_debut	Heure début de service.	VARCHAR	10
	Heure_fin	Heure fin de service.	VARCHAR	10
Medecin	Id_medecin	Identifiant du médecin	INT	11
	Nom	Nom	VARCHAR	50
	Prenom	Prénom	VARCHAR	50
	Adresse	Adresse	VARCHAR	255
	Email	Adresse Email	VARCHAR	50
	Telephone	Numéro de téléphone	VARCHAR	15

	Date_de_naissance	Date de naissance	DATE	10
	Mot_de_passe	Mot de passe	VARCHAR	255
	Id_service	Identifiant du service dans lequel il est affecté.	INT	11
Secrétaire	Id_secretaire	Identifiant de secrétaire	INT	11
	Nom	Nom	VARCHAR	50
	Prenom	Prénom	VARCHAR	50
	Adresse	Adresse	VARCHAR	255
	Email	Adresse Email	VARCHAR	50
	Telephone	Numéro de téléphone	VARCHAR	15
	Date_de_naissance	Date de naissance	DATE	10
	Mot_de_passe	Mot de passe	VARCHAR	255
	Id_etablissement	Identifiant de l'établissement dans lequel la secrétaire est affectée.	INT	11
Admin	Id_admin	Identifiant de l'admin	INT	11
	Nom	Nom	VARCHAR	50
	Prenom	Prénom	VARCHAR	50
	Adresse	Adresse	VARCHAR	255
	Email	Adresse Email	VARCHAR	50
	Telephone	Numéro de téléphone	VARCHAR	15
	Date_de_naissance	Date de naissance	DATE	10
	Mot_de_passe	Mot de passe	VARCHAR	255
	Id_etablissement	Identifiant de l'établissement dans lequel l'admin est affecté.	INT	11
Super-Admin	Id_super	Identifiant du super admin de l'établissement	INT	11
	Nom	Nom	VARCHAR	50
	Prenom	Prénom	VARCHAR	50

	Adresse	Adresse	VARCHAR	255
	Email	Adresse Email	VARCHAR	50
	Telephone	Numéro de téléphone	VARCHAR	15
	Date_de_naissance	Date de naissance	DATE	10
	Mot_de_passe	Mot de passe	VARCHAR	255
	Id_etablissement	Identifiant de l'établissement que le super admin contrôle.	INT	11
Patient	Id_patient	Identifiant du patient	INT	11
	Nom	Nom	VARCHAR	50
	Prenom	Prénom	VARCHAR	50
	Adresse	Adresse	VARCHAR	255
	Email	Adresse Email	VARCHAR	50
	Telephone	Numéro de téléphone	VARCHAR	15
	Date_de_naissance	Date de naissance	DATE	10
File	Id_file	Identifiant de la file	INT	11
	Nom	Nom de la file	VARCHAR	50
	Priorite	Priorité de la file	INT	1
	Id_service	Identifiant du service de la file.	INT	11
Consultation	Id_consultation	Identifiant de la consultation	INT	11
	Id_patient	Identifiant du patient	INT	11
	Id_medecin	Identifiant du médecin	INT	11
	Date_Heure	Date et heure de la consultation	DATE	50
	Duree	Durée de la consultation	VARCHAR	15
Service	Id_service	Identifiant du service.	INT	11
	Nom	Nom du service	VARCHAR	50
	Id_etablissement	Identifiant de l'établissement	INT	11

Annexes

		dans lequel fait partie le service.		
Etablissement	Id_etablissement	Identifiant de l'établissement.	INT	11
	Nom	Nom de l'établissement	VARCHAR	50
	Adresse	Adresse de l'établissement.	VARCHAR	255
	Email	Adresse Email de l'établissement.	VARCHAR	50
	Téléphone	Numéro de contacte l'établissement	VARCHAR	15
	Site Web	Site internet	VARCHAR	50

Abstract

Given the growing number of people visiting hospitals in Algeria every day, it has become very difficult to manage the flow of patients and the scheduling of appointments. Our job is to design and implement an appointment scheduling and queue management system. This solution aims to guarantee and ensure a better management of patient flow within hospitals, and also to enable patients to benefit from online appointment booking, improving quality of service. In order to realize our system, we opted for a joint use of Scrum and UP in the design phase, and Nextjs and Flutter for the implementation.

Keywords: Queue management, Appointment management, health, patients, UML, MVC, Scrum, UP, Nextjs, Flutter, EPH Amizour.

Résumé

Vu la croissance du nombre de personne se présentant tous les jours aux établissements hospitaliers en Algérie, il est devenu par conséquent très difficile de bien gérer le flux de patients et la prise de rendez-vous. Notre travail consiste à concevoir et à réaliser un système de prise de rendez-vous et gestion de files d'attente. Cette solution vise à garantir une meilleure gestion du flux de patients au sein des établissements de santé et de permettre aussi aux patients de bénéficier d'une prise de rendez-vous en ligne améliorant la qualité de service. Afin de réaliser notre système nous avons opté pour une utilisation conjointe de Scrum et UP dans l'étape de conception et de Nextjs et Flutter à l'implémentation.

Mots-clé : Gestion des files d'attente, Gestion des rendez-vous, santé, patients, UML, MVC, Scrum, UP, Nextjs, Flutter, EPH Amizour.