



MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE ABDELHAMID IBN BADIS – MOSTAGANEM



Faculté des Sciences Exacte et d'Informatique
Département de Mathématiques et informatique
Filière : Informatique

MEMOIRE DE FIN D'ETUDES
Pour l'Obtention du Diplôme de Master en Informatique
Option : **Ingénierie des Systèmes d'Information**

Présenté par :

Righi Redouane Dhiae Eddine

Mkrarbech Fatima

THEME :

Conception et Implémentation d'une Application Web Sécurisé
avec la gestion d'authentification et d'autorisation (cas d'étude :
gestion de dossier médical au sein d'une clinique)

Soutenu le :

Devant le jury composé de :

Nom et Prénom	Grade	Université de Mostaganem	Président
Nom et Prénom	Grade	Université de Mostaganem	Examineur
Benidris Fatima.Z	Grade	Université de Mostaganem	Encadrant

Année Universitaire 2024-2025

Résumé

Ce projet vise à développer un site web sécurisé dédié à la gestion des dossiers médicaux et à la prise de rendez-vous au sein d'une clinique. La plateforme permettra aux patients de planifier facilement leurs consultations, tandis que les professionnels de santé pourront accéder aux dossiers médicaux et les administrer de manière efficace. Le système reposera sur des mécanismes robustes d'authentification et d'autorisation afin de garantir la confidentialité et la sécurité des données, en assurant un accès strictement réservé selon les rôles des utilisateurs. En somme, ce site web constituera une solution fiable, intuitive et sécurisée pour optimiser la gestion médicale et améliorer l'expérience patient.

Mots-clés : Site web, dossier médical, sécurité, Spécification Exigences Logicielles (SEL), Business Process Model and Notation (BPMN).

Abstract

This project aims to develop a secure web platform dedicated to managing medical records and scheduling appointments within a clinic. The platform will allow patients to easily book consultations, while healthcare professionals will be able to access and efficiently manage medical records. The system will rely on robust authentication and authorization mechanisms to ensure data confidentiality and security, with strictly role-based access control. In summary, this web application will offer a reliable, user-friendly, and secure solution to optimize medical management and enhance the patient experience.

Keywords: website, security, medical folder, Software Requirements Specification (SRS), Business Process Model and Notation (BPMN).

Remerciements

Nous tenons à exprimer notre profonde gratitude envers Allah, qui nous a donné la volonté et le courage nécessaires pour mener à bien ce travail. Nous remercions sincèrement notre encadrant, Madame Benidris Fatima zohra, pour son soutien, son expertise et sa disponibilité tout au long du projet. Nous sommes également reconnaissants envers notre Monsieur Khelifa Norreddine, pour son accompagnement constant, ses encouragements et ses précieux commentaires, nous tenons à exprimer notre gratitude particulière madame Maghni Sandid Zohra pour ses conseils et son aide.

Nous souhaitons exprimer notre profonde gratitude envers nos parents et nos familles pour leur soutien indéfectible et leurs encouragements tout au long de notre parcours académique. Leur amour inconditionnel et leurs sacrifices ont été les piliers de notre réussite. Nous reconnaissons l'importance de leur présence constante, de leurs encouragements chaleureux et de leurs conseils avisés. Leur soutien émotionnel et financier nous a permis de poursuivre nos études avec confiance et détermination.

Liste des figures

Figure 1: Structure de DMP [11].....	7
Figure 2: Diagramme de cause/effet.....	20
Figure 3:Processus métier de collaboration de gestion de dossier médical.....	23
Figure 4: Processus métier privé de gestion de dossier médical	24
Figure 5:Processus métier de chorégraphie de gestion de dossier médical	25
Figure 6 : Diagramme de profile	36
Figure 7 : Diagramme de package.....	37
Figure 8 : Diagramme de cas d'utilisation de patient.....	38
Figure 9 : Diagramme de cas d'utilisation de médecin et spécialiste.....	39
Figure 10 : Diagramme de cas d'utilisation de réceptionniste	39
Figure 11 : Diagramme de cas d'utilisation d'administrateur	40
Figure 12: Diagramme de séquence de l'inscription d'un patient partie 1	42
Figure 13: Diagramme de séquence de l'inscription d'un patient partie 2.....	43
Figure 14: Diagramme de séquence d'un patient prendre un rendez-vous partie 1	46
Figure 15: Diagramme de séquence d'un patient prendre un rendez-vous partie 2	47
Figure 16: Diagramme de séquence de patient pour consulter son dossier médical	49
Figure 17: Diagramme de séquence de patient pour consulter sa liste des rendez-vous.....	51
Figure 18: Diagramme de séquence de soumettre une plainte	54
Figure 19: Diagramme de séquence de patient soumettre un avis	56
Figure 20:Diagramme de séquence de patient gérer le rendez-vous partie 1	58
Figure 21: Diagramme de séquence de patient gérer le rendez-vous partie 2.....	59
Figure 22: Diagramme de séquence de médecin pour gérer les dossiers médicaux partie 1	62
Figure 23: Diagramme de séquence de médecin pour gérer les dossiers médicaux partie 2	63
Figure 24: Diagramme de séquence de médecin pour gérer son calendrier partie 1	65
Figure 25: Diagramme de séquence de médecin pour gérer son calendrier partie 2.....	66
Figure 26: Diagramme de classe de système.....	68
Figure 27: Diagramme de classe de conception pour l'authentification	69
Figure 28: Diagramme de composant.....	70
Figure 29: Diagramme de déploiement	71

Figure 30: Diagramme de test d'inscription	72
Figure 31: Diagramme Test d'authentification	73
Figure 32: Une attaque XSS dans le formulaire de patient	77
Figure 33: Une attaque SQL dans le formulaire de patient	78
Figure 34: Hachage sur le mot de passe de patient.....	79
Figure 35: Hachage check sur le mot de passe.....	79
Figure 36: rate limiter	80
Figure 37: Code PHP de security headers	80
Figure 38: Middleware dans code php	82
Figure 39: code php Input sanitizer	83
Figure 40: Verify Csrf Token	83
Figure 41: Interface de tableau de bord de patient	84
Figure 42: Interface de dossier médical.....	85
Figure 43: Processus métier public de gestion de dossier médical.....	93
Figure 44: processus de vue en conversation	94
Figure 45: Diagramme de séquence pour consulter les rendez-vous	105
Figure 46: Diagramme de séquence pour créer les rendez-vous partie 1	107
Figure 47: Diagramme de séquence pour créer les rendez-vous partie 2	108
Figure 48: Diagramme de séquence pour gérer les comptes des utilisateurs partie 1	110
Figure 49: Diagramme de séquence pour gérer les comptes des utilisateurs partie 2	111
Figure 50: Diagramme de séquence pour administrateur gérer les rendez-vous partie 1	113
Figure 51: Diagramme de séquence pour administrateur gérer les rendez-vous partie 2	114
Figure 52: Diagramme de séquence de consulter plaintes	115
Figure 53: Diagramme de séquence pour s'authentifier.....	117
Figure 54: Diagramme de séquence pour modifier le profil	119
Figure 55: Diagramme de séquence pour réinitialiser le mot de passe	121
Figure 56: Diagramme de séquence de gérer ses rendez-vous.....	123
Figure 57: Diagramme de séquence de prendre un rendez-vous chez un spécialiste.....	125

Liste des tableaux

Tableau N°	Titre du tableau	Page
Tableau 1	Bénéfices de l'informatisation du dossier patient	23

Liste des abréviations

Abbreviation	Expression Complète
DMP	Dossier Médical Personnel
DMI	Dossier Médical Informatisé
DME	Dossier Médical Eleélectronique
UML	Unified Modeling Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
CSS	Cascading Style Sheets
OMS	Organisation Mondiale de la Santé
DDoS	Distributed Denial of service
OWASP	Open Worldwide Application Security Project
PHP	Hypertext Preprocessor
RUP	Rational Unified Process
DNS	Domain Name System
NCSA	National Center for Supercomputing Applications
W3C	World Wide Web Consortium
TLS	Transport Layer Security
RGPD	Règlement Général sur la Protection des Données
XSS	Cross-Site Scripting

CSRF	Cross-Site Request Forgery
BPMN	Business process model and notation

Table de Matière

Introduction Générale.....	xiii
----------------------------	------

Chapitre 1 Généralités sur le web et le dossier médical	2
---	---

1.1	Introduction.....	2
1.2	Généralités sur le web.....	2
1.2.1	Définition de web	2
1.2.2	Architecture du web.....	3
1.2.3	Quelques standards.....	4
1.2.4	Les sites web.....	5
1.3	Le dossier médical	6
1.3.1	Définition.....	6
1.3.2	Constitution de dossier médical.....	6
1.3.3	Difficultés rencontrés dans le dossier médical papier	7
1.3.4	Dématérialisation du dossier patient.....	7
1.3.5	Les bénéfices de DME par rapport au dossier papier	9
1.3.6	Caractéristiques d'un bon dossier médical	11
1.4	Conclusion	11

Chapitre 2 Sécurité dans les applications web	12
---	----

2.1	Introduction.....	12
	Les applications web sont devenues centrales dans les systèmes d'information modernes. Elles permettent un accès à distance aux services via un navigateur, ce qui les rend très pratiques mais aussi particulièrement vulnérables. La sécurité dans ce contexte ne se limite pas à la simple protection des données, elle englobe l'ensemble des mécanismes garantissant la confidentialité, l'intégrité, la disponibilité et la non-répudiation des informations et services fournis.....	12
2.2	Principes fondamentaux de la sécurité.....	12
2.3	Principales menaces et vulnérabilités	13
2.4	Mécanismes de sécurité dans les applications web.....	14
2.4.1	Authentification	14
2.4.2	Autorisation	14

2.4.3	Gestion de session sécurisée.....	14
2.4.4	Chiffrement des données	14
2.4.5	Traçabilité et journalisation.....	15
2.5	Standards et outils associés.....	15
2.6	État de l’art.....	15
2.6.1	OWASP Top Ten Web Application Security Risks.....	15
2.6.2	Burp Suite	16
2.6.3	Architecture Zero Trust	16
2.6.4	OWASP ZAP.....	16
2.6.5	SAST	17
2.6.6	Arachni	17
2.6.7	Wapiti	17
2.6.8	Nessus.....	17
2.7	Conclusion	18
Chapitre 3 Analyse et Modélisation de processus métier		19
3.1	Introduction.....	19
3.2	Problématique	19
3.2.1	Diagramme cause et effet	20
3.3	Modélisation processus métier.....	21
3.3.1	Définition des processus métiers	21
3.3.2	Modélisation par BPMN.....	21
3.4	Conclusion	25
Chapitre 4 Conception et Modélisation		27
4.1	Introduction.....	27
4.2	Spécifications d’exigences de logiciel.....	27
4.2.1	Objet	27
4.2.2	Description générale	28
4.2.3	Exigences spécifiques.....	31
4.3	Modélisation	34
4.3.1	Présentation d’UML	34

4.3.2	Les Diagrammes d'UML.....	34
4.4	Méthode RUP.....	34
4.4.1	Avantages de la méthode RUP	34
4.5	La démarche de la modélisation	35
4.5.1	Modèle de cas d'utilisation.....	35
4.5.2	Modèle de conception.....	69
4.5.3	Modèle d'implémentation.....	70
4.5.4	Modèle de déploiement	70
4.5.5	Modèle de test.....	71
4.6	Conclusion	74
Chapitre 5 Implémentation et Test.....		75
5.1	Introduction.....	75
5.2	Environnement logiciel.....	75
5.2.1	Laravel.....	76
5.2.2	Github	76
5.2.3	Vs code	76
5.2.4	Phpmyadmin.....	76
5.2.5	Xampp server.....	76
5.2.6	Apache.....	76
5.2.7	Git.....	77
5.3	Le test des attaques	77
5.3.1	Injection.....	77
5.4	Les mécanismes de sécurité	78
5.4.1	Authentification et Sécurité des Accès :.....	78
5.4.2	Content security policy (CSP).....	80
5.4.3	Middleware.....	81
5.4.4	L'input sanitizer.....	82
5.4.5	Verify Csrf Token.....	83
5.5	Les interfaces de site web	84
5.5.1	Interface de tableau de bord de patient.....	84
5.5.2	Interface de dossier médical	84

5.6 Conclusion	85
Conclusion générale	86
Bibliographie.....	88
Annexe A	92

Introduction Générale

La transformation numérique du secteur de la santé a profondément modifié la manière dont les établissements médicaux gèrent les informations relatives aux patients. Aujourd'hui, les cliniques cherchent à moderniser leurs processus en adoptant des solutions informatiques pour stocker, consulter et partager les dossiers médicaux. Toutefois, cette transition vers le numérique ne va pas sans poser de réels défis, notamment en matière de sécurité des données.

En effet, les dossiers médicaux contiennent des informations sensibles, dont la divulgation ou l'altération pourrait avoir de graves conséquences pour les patients et les établissements de santé. Il est donc essentiel de mettre en place des systèmes capables de garantir la confidentialité, l'intégrité et la disponibilité de ces données. De plus, dans un environnement où plusieurs types d'utilisateurs interagissent avec le système (médecins, administrateurs, patients, réception), il devient crucial de gérer de manière stricte l'authentification et l'autorisation des accès.

C'est dans ce contexte que s'inscrit notre projet, qui consiste à proposer une solution web sécurisée, moderne et adaptée au fonctionnement réel d'une clinique avec une efficace authentification et autorisation des utilisateurs dans le cadre de la gestion des dossiers médicaux au sein d'une clinique, tout en garantissant la confidentialité, l'intégrité et la disponibilité des données sensibles. Cependant, notre projet vise à :

- Évaluer les besoins spécifiques de la clinique.
- Mettre en place une architecture sécurisée respectant les bonnes pratiques du développement web.
- Implémenter un système d'authentification fiable pour les différents profils d'utilisateurs (administrateurs, médecins, patients, réception).
- Gérer les droits d'accès aux données selon le rôle de l'utilisateur (autorisation

- Assurer la confidentialité, l'intégrité et la disponibilité des informations médicales
- Offrir une interface conviviale, simple d'utilisation et adaptée aux besoins réels d'une clinique.

Pour réaliser cet objectif notre rapport est structuré autour de cinq chapitres avec une introduction et une conclusion générale :

- Dans le premier chapitre, nous allons présenter des généralités sur le web et le dossier médical, afin d'introduire les concepts de base et le contexte métier.
- Le deuxième chapitre sera consacré à la sécurité des applications web. Il expose les principales menaces et les bonnes pratiques pour protéger les systèmes.
- Le troisième chapitre concerne l'analyse et la modélisation de processus métier de cas choisie pour l'étude.
- Le quatrième chapitre traite de la conception et de la modélisation du système, en s'appuyant sur des outils tels que les diagrammes UML et les outils informatiques adoptés pour assurer la sécurité et la fiabilité du système.
- Le dernier chapitre décrit l'implémentation de la solution, depuis le choix des technologies jusqu'à la mise en œuvre des fonctionnalités clés.

Chapitre 1

Généralités sur le web et le dossier médical

1.1 Introduction

L'émergence du web a profondément transformé notre manière d'accéder à l'information, de communiquer, mais aussi de gérer des systèmes complexes, notamment dans le domaine de la santé. Grâce à l'évolution des technologies web, il est aujourd'hui possible de concevoir des plateformes interactives, accessibles à distance, facilitant la gestion des dossiers médicaux électroniques (DME). Ces derniers permettent aux professionnels de santé de centraliser les données cliniques d'un patient, de suivre son historique médical, et de prendre des décisions éclairées, tout en garantissant un accès simplifié et sécurisé à l'information.

Ce chapitre a pour objectif de poser les bases du projet en présentant le fonctionnement général du web, son rôle dans les systèmes d'information modernes, et plus particulièrement son application dans le domaine médical. Il met en lumière les avantages liés à la gestion électronique des dossiers médicaux, mais également les enjeux liés à leur mise en œuvre, notamment en ce qui concerne l'interopérabilité, l'accessibilité, et surtout, la protection des données de santé, qui constitue le fil conducteur de ce projet.

1.2 Généralités sur le web

1.2.1 Définition de web

Le Web est une application distribuée, structurée autour de l'utilisation de liens hypertextes pour naviguer entre des documents stockés sur des serveurs à travers Internet. Il repose sur des technologies de communication, dont principalement le protocole HTTP, et sur un format de codage des documents, principalement HTML [1] ;

1.2.2 Architecture du web

1. Modèle client/serveur

Le modèle client/serveur est une architecture fondamentale des systèmes distribués, largement utilisée dans les applications web et les systèmes d'information modernes. Ce modèle repose sur la répartition des rôles entre deux entités principales : le client, qui envoie des requêtes pour accéder à des ressources ou services, et le serveur, qui traite ces requêtes et fournit les réponses appropriées. Le serveur héberge les ressources (comme des pages web, bases de données ou services applicatifs), tandis que le client agit comme une interface d'accès pour l'utilisateur final, généralement via un navigateur web ou une application dédiée. Ce modèle favorise la centralisation des données, la mutualisation des services et la gestion centralisée de la sécurité, tout en assurant une évolutivité du système. Il est à la base de la plupart des interactions sur Internet et des plateformes informatiques modernes [2] ;

2. Protocole HTTP/HTTPS

Le protocole HTTP (HyperText Transfer Protocol) est un protocole de communication client-serveur qui permet le transfert de données sur le Web, notamment les pages web, les images ou les fichiers multimédias. Fonctionnant sur le port 80, HTTP est stateless, c'est-à-dire qu'il ne conserve pas l'état des connexions entre les requêtes. Toutefois, l'un des principaux inconvénients de HTTP est l'absence de chiffrement, rendant les données vulnérables à l'interception et à la falsification. Pour répondre à ces enjeux de sécurité, le protocole HTTPS (HyperText Transfer Protocol Secure) a été introduit. Il s'appuie sur le protocole TLS (Transport Layer Security) pour chiffrer les échanges entre le client et le serveur, garantissant ainsi la confidentialité, l'intégrité et l'authenticité des données transmises. Utilisé sur le port 443, HTTPS est devenu la norme pour toute application web manipulant des informations sensibles, notamment dans le domaine médical ou bancaire [3].

3. Serveurs web et navigateurs

Les serveurs web et les navigateurs constituent deux éléments essentiels de l'architecture du Web. Un serveur web est un logiciel (tel qu'Apache, Nginx ou Microsoft IIS) installé sur un ordinateur connecté à Internet, dont le rôle est de stocker, traiter et fournir des contenus web (pages HTML, fichiers CSS, scripts, images, etc.) à des clients, via le protocole HTTP ou HTTPS. Lorsqu'un utilisateur saisit une URL dans un navigateur, celui-ci envoie une requête au serveur

correspondant, qui lui retourne le contenu demandé. Le navigateur web (comme Google Chrome, Mozilla Firefox ou Safari) est une application cliente permettant aux utilisateurs d'interagir avec le Web. Il interprète le code HTML, CSS et JavaScript reçu du serveur et le rend lisible sous forme graphique. Cette interaction client-serveur est au cœur de toute application web moderne [4].

1.2.3 Quelques standards

1. HTML, CSS et JavaScript

Les technologies HTML, CSS et JavaScript sont les trois piliers essentiels du développement web côté client. HTML (HyperText Markup Language) sert à structurer le contenu d'une page web en définissant des éléments tels que les titres, paragraphes, tableaux, images, ou formulaires. Il établit l'ossature logique du document. CSS (Cascading Style Sheets) complète le HTML en définissant l'apparence visuelle de ces éléments : couleurs, polices, alignements, dimensions, dispositions responsives, etc. Enfin, JavaScript est un langage de programmation exécuté côté client, permettant de créer des interactions dynamiques (par exemple, des animations, des validations de formulaires, des mises à jour en temps réel sans rechargement de page). L'interaction de ces trois technologies permet de concevoir des interfaces web à la fois structurées, esthétiques et interactives.

2. Le Domain Name System (DNS)

Le DNS (Domain Name System) est un système qui assure le stockage et la distribution des informations associées aux noms de domaine. Il permet notamment d'établir des correspondances entre les noms de domaine (comme `example.com`) et les adresses IP numériques telles que `192.0.2.1`, qui identifient les serveurs sur Internet. Le DNS joue un rôle essentiel dans la résolution des noms de domaine, permettant ainsi aux utilisateurs de naviguer sur le web en utilisant des noms conviviaux plutôt que des adresses IP complexes. Grâce au DNS, lorsque vous saisissez un nom de domaine dans votre navigateur, il est traduit en adresse IP correspondante, ce qui permet d'établir la connexion avec le serveur approprié pour accéder au site web associé. Le DNS assure donc la traduction des noms de domaine en adresses IP, facilitant ainsi la navigation sur Internet.

3. **Url et liens hypertextes**

Les URL (Uniform Resource Locator) et les liens hypertextes sont des éléments essentiels du Web. Ensemble, ils facilitent la navigation, l'accès aux informations et la structure de l'Internet. Voici une explication détaillée de ces deux concepts :

- **URL :** Une URL (Uniform Resource Locator) est une adresse unique qui identifie une ressource sur Internet. Elle contient différentes informations, telles que le nom de la machine, le nom du service demandé et le nom du document. Une URL permet d'indiquer un protocole de communication (comme HTTP) et un emplacement spécifique pour accéder à une ressource sur le Web. On peut la considérer comme un système de fichier universel sur Internet. Une URL est généralement composée de plusieurs parties, notamment le protocole (comme http://), le nom d'hôte (comme www), le numéro de port (facultatif) le répertoire et le nom du fichier. Par exemple, '**http://www.example.com:80/repertoire/nom_fichier**' est une URL typique où "http://" est le protocole, "**www.example.com**" est le nom d'hôte, "80" est le numéro de port (optionnel), **"/repertoire/**" est le répertoire et "nom_fichier" est le nom du fichier. Les URL sont utilisées pour accéder aux sites web, aux pages, aux documents et à d'autres ressources en ligne.
- **Un lien hypertexte :** est défini comme un élément interactif dans un document numérique, permettant de diriger l'utilisateur vers une autre ressource, qu'elle soit sur la même page ou sur une page différente, souvent via un clic. Ce lien peut être sous forme de texte, d'image ou d'un autre type de média cliquable, facilitant ainsi la navigation et la découverte de nouveaux contenus [1] ;

1.2.4 Les sites web

1. **Définition**

Un site web est un ensemble structuré de pages web accessibles via Internet à partir d'une adresse (URL), qui sont liées entre elles par des hyperliens et souvent hébergées sur un même serveur. Il peut fournir des contenus variés (texte, images, vidéos) et des services interactifs [18] ;

1.3 Le dossier médical

1.3.1 Définition

Selon F. Roger France [5], le Dossier Médical Patient (DMP) est « la mémoire écrite de toutes les informations concernant un malade, constamment mises à jour et dont l'utilisation est à la fois individuelle et collective ».

Dans [6], on définit le Dossier Médical comme suit : « Le dossier du malade ne se résume pas à l'observation écrite du médecin (le dossier médical proprement dit) ou aux notes de l'infirmière (le dossier infirmier). Il englobe tout ce qui peut être mémorisé chez un malade, des données démographiques aux enregistrements électro-physiologiques ou aux images les plus sophistiquées. Compte tenu de ce rôle, le dossier du malade restera longtemps l'outil principal de centralisation et de coordination de l'activité médicale ».

1.3.2 Constitution de dossier médical

Un dossier médical est constitué pour chaque patient hospitalisé dans un établissement de santé public ou privé. Ce dossier doit au moins contenir ces éléments [10] :

- La partie administrative du Dossier Médical Patient fournit les données d'identification du patient et les données sociodémographiques régulièrement tenues à jour (suivi de l'identité de l'état civil, de la couverture sociale, du statut matrimonial, des employeurs, etc.).
- Le Dossier Médical Patient comporte les données de santé dont le volume et la complexité croît sous l'influence du développement des spécialités médicales et de leur technicité.

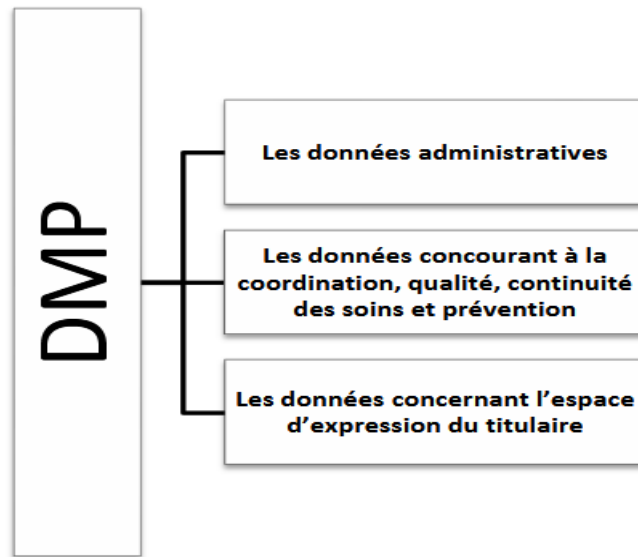


Figure 1: Structure de DMP [11]

1.3.3 Difficultés rencontrés dans le dossier médical papier

- La prescription illisible qui peut induire le personnel soignant à l'erreur.
- Des bilans ou bien des clichés d'un patient dans le dossier d'un autre patient ce qui nous expose à des erreurs de diagnostic.
- Perte de dossiers, documents, bilans, etc. Ce qui nous oblige à les refaire donc un coût financier supplémentaire et une perte de temps considérable.
- Difficulté d'étude épidémiologique et l'exploitation du contenu.
- Coût important : Utilisation abusive du papier pochettes dossiers, ordonnanciers, etc.) [12].

1.3.4 Dématérialisation du dossier patient

La dématérialisation du dossier patient consiste à remplacer le traditionnel dossier papier par une version numérique accessible via un système informatique sécurisé. Ce processus fait partie intégrante de la transformation numérique du secteur de la santé et vise à moderniser la gestion des données médicales. Voici un aperçu des avantages, défis et impacts de cette évolution.

1. Les avantages de la dématérialisation

- Amélioration de l'accessibilité et de la rapidité : Un dossier patient informatisé permet une consultation rapide et à distance des informations médicales. Les professionnels de santé, qu'ils soient au sein du même établissement ou à distance, peuvent avoir accès aux données du patient en temps réel. Cela améliore la coordination des soins et réduit les risques d'erreur médicale liés à une lecture difficile des informations manuscrites.
- Centralisation des données : Les informations du patient sont stockées dans un fichier centralisé et sécurisé, ce qui facilite leur mise à jour et leur conservation sur le long terme. De plus, cela permet d'éviter les pertes de documents, fréquentes avec les dossiers papier.
- Meilleure gestion des données : Grâce aux outils numériques, les dossiers peuvent être classés, triés et analysés de manière plus efficace. Les professionnels de santé peuvent ainsi facilement suivre l'évolution de l'état de santé du patient et prévoir les traitements à venir [19].

2. Les défis de la dématérialisation

- Protection des données personnelles : La sécurité des données est un enjeu majeur dans la dématérialisation des dossiers médicaux. Les informations médicales sont sensibles et doivent être protégées contre tout accès non autorisé. Avec l'informatisation des dossiers, la sécurité des données devient un enjeu crucial. Des protocoles de sécurité rigoureux doivent être mis en place pour protéger ces informations contre les cyberattaques, les violations de la confidentialité et les erreurs humaines. La législation, telle que le Règlement Général sur la Protection des Données (RGPD) en Europe, impose des normes strictes en matière de confidentialité des données de santé [20].
- Formation du personnel médical et administratif : La transition vers un dossier patient informatisé nécessite que le personnel soit formé à l'utilisation des nouveaux outils. Une prise en main incorrecte ou une résistance au changement peut freiner la mise en œuvre de la dématérialisation. Une formation adéquate est donc essentielle pour garantir que ces systèmes sont utilisés de manière optimale et pour éviter les erreurs de manipulation.[21]
- Coût d'investissement : L'implémentation de systèmes de gestion des dossiers médicaux électroniques (DME) implique des coûts significatifs, notamment l'achat de logiciels

spécialisés, l'infrastructure nécessaire pour stocker les données et la mise en place de mesures de sécurité. Ces coûts peuvent représenter un investissement important pour les établissements de santé, en particulier pour les plus petits centres médicaux.[21]

3. Les impacts sur le système de santé

- La réduction des erreurs médicales : Une étude a examiné les perceptions des professionnels de santé concernant la réduction des erreurs médicales grâce à l'utilisation des dossiers médicaux électroniques. Les résultats ont montré que l'utilisation des DME permettait de réduire les erreurs médicales de 50 à 60 %, notamment les erreurs de dosage, les chirurgies sur le mauvais site et les erreurs de médication [22].
- La qualité des soins : Une étude a observé l'impact des DME sur la qualité des soins, en modélisant leur effet sur la mortalité, les réadmissions et les complications. Les résultats ont montré que l'adoption des DME était associée à une amélioration de la qualité des soins, bien que l'impact sur la mortalité et les réadmissions soit encore à confirmer [23].

1.3.5 Les bénéfices de DME par rapport au dossier papier

Le tableau suivant illustre parfaitement la différence des caractéristiques fonctionnelles et leurs impacts dans le dossier traditionnel et informatisé. L'intervalle [0, +++] représente le degré de l'impact des caractéristiques dans les deux dossiers : '0' : aucun impact, '+' : impact faible, '++' : impact peu élevé et '+++': impact très élevé.

Tableau 1 - Bénéfices de l'informatisation du dossier patient [13].

Caractéristique fonctionnelle	Dossier traditionnel	Dossier informatisé
Intégration des données (+multimédia)	+	+++
Lisibilité du dossier	+	++
Prise en charge ensemble des problèmes	+	++
Complétude	+	+++
Accès	séquentiel	simultané
Disponibilité	local	global
Accès à distance	0	+++
Chaînage d'épisodes de soins	+	+++
Chaînage des dossiers distribués	0	++
Evaluation des soins	+	+++
Recherche clinique, épidémiologique	+	+++
Contrôle de gestion, planification	0	+++
Facilité d'utilisation du dossier	+++	+
Formalisation de démarche de soins	+	+++
Adhésion aux protocoles de soins	+	+++
Connexion à des banques de données documentaires ou des connaissances	0	+++
Sécurité de l'information	+	+++
Confidentialité	++	+

1.3.6 Caractéristiques d'un bon dossier médical

Les qualités d'un bon dossier patient sont définies dans une notice de l'OMS (Organisation Mondiale de la Santé) comme étant les suivantes [14] :

- Identifier sans ambiguïté la personne qu'il concerne.
- Être lisible et pouvoir être compris par tous ceux qui seront amenés à l'utiliser.
- Être précis, concis et logique.
- Être cohérent quant à la disposition et au format des documents qui le constituent.
- Désigner les personnes qui sont amenées à y porter des écritures afin qu'on puisse, le cas échéant, leur demander un complément d'informations.
- Pouvoir être rapidement retrouvé quand on en a besoin.

Ainsi définies, toute conception ou reconception d'un dossier patient doit être guidée par ces caractéristiques.

1.4 Conclusion

En résumé, l'évolution du Web a profondément transformé le secteur de la santé, notamment à travers la dématérialisation du dossier médical. Ce dernier, désormais informatisé, offre de nombreux avantages en matière d'accessibilité, de coordination des soins et de qualité de service. Cependant, cette transformation numérique implique également de nouveaux enjeux, notamment en matière de sécurité et de confidentialité des données de santé.

Dans ce contexte, le chapitre suivant sera consacré à la sécurité Web, un aspect fondamental pour garantir la protection des sites web et les dossiers médicaux informatisés contre les accès non autorisés, les fuites de données et les cyberattaques.

Chapitre 2

Sécurité dans les applications web

2.1 Introduction

Les applications web sont devenues centrales dans les systèmes d'information modernes. Elles permettent un accès à distance aux services via un navigateur, ce qui les rend très pratiques mais aussi particulièrement vulnérables. La sécurité dans ce contexte ne se limite pas à la simple protection des données, elle englobe l'ensemble des mécanismes garantissant la confidentialité, l'intégrité, la disponibilité et la non-répudiation des informations et services fournis.

2.2 Principes fondamentaux de la sécurité

Les quatre piliers de la sécurité de l'information sont les suivants [15] :

- Confidentialité : veiller à ce que seules les personnes autorisées puissent accéder à certaines données. Dans le cadre d'une application de dossier médical, cela signifie que seuls les médecins concernés doivent pouvoir consulter les données d'un patient.
- Intégrité : garantir que les données ne sont ni modifiées ni altérées de manière non autorisée. Cela implique, par exemple, que l'historique médical d'un patient ne puisse pas être modifié par un tiers non habilité.
- Disponibilité : assurer l'accessibilité continue des services et données aux utilisateurs autorisés, même en cas de panne ou d'attaque (ex : DDoS).
- Authenticité & non-répudiation : garantir que l'utilisateur est bien celui qu'il prétend être (authenticité) et qu'il ne peut pas nier une action effectuée (non-répudiation), notamment via la journalisation et les signatures électroniques.

2.3 Principales menaces et vulnérabilités

Les vulnérabilités des applications web sont dues à des faiblesses de sécurité qui permettent aux acteurs malveillants de manipuler le code source, d'obtenir un accès non autorisé, de voler des données ou d'interférer de quelque manière que ce soit avec le fonctionnement normal de l'application.

Le document « OWASP Top 10 » répertorie les risques de sécurité les plus critiques pour les applications web. On a mis des exemples de quelques vecteurs d'attaque connus :

- Injection SQL : consiste à insérer des commandes SQL malveillantes dans un champ de formulaire afin d'accéder ou de manipuler la base de données.
- Cross-Site Scripting (XSS) : permet à un attaquant d'injecter du code JavaScript malveillant dans une page consultée par d'autres utilisateurs.
- Cross-Site Request Forgery (CSRF) : oblige un utilisateur connecté à exécuter une action non désirée sur une application où il est authentifié.
- Brute-force et attaques par dictionnaire : visent à deviner les mots de passe par essais répétés.
- Faille d'authentification : mauvaise gestion des sessions, mots de passe faibles, absence de timeout de session, etc.
- Vol ou interception de données : via des communications non chiffrées (HTTP au lieu de HTTPS).
- Failles de session : Sessions non sécurisées ou mal gérées peuvent être interceptées ou utilisées par des attaquants.
- Mauvaises configurations : Une mauvaise configuration des systèmes peut ouvrir la voie à des vulnérabilités. Cela inclut l'exposition de fichiers sensibles, des erreurs de configuration du serveur ou l'utilisation de mots de passe par défaut. Exemples : fichiers `.env` exposés, serveurs en mode debug, mots de passe par défaut non changés.

2.4 Mécanismes de sécurité dans les applications web

Les mécanismes de sécurité sont des outils techniques qui aident à protéger les applications contre les menaces identifiées précédemment.

2.4.1 Authentification

L'authentification est le processus permettant de vérifier l'identité d'un utilisateur. Elle repose généralement sur la saisie d'identifiants (nom d'utilisateur, mot de passe) et peut être renforcée par une authentification à deux facteurs (2FA). Un bon système d'authentification doit inclure une gestion sécurisée des sessions et des mécanismes de protection contre les attaques par force brute [16].

2.4.2 Autorisation

L'autorisation détermine les ressources et fonctionnalités qu'un utilisateur peut accéder ou utiliser après authentification. Elle est essentielle dans les systèmes multi-profils comme les plateformes médicales, où un patient ne doit pas avoir accès aux mêmes données qu'un médecin ou un administrateur. Ce mécanisme repose souvent sur la définition de rôles (RBAC – Role-Based Access Control) [17].

2.4.3 Gestion de session sécurisée

La gestion sécurisée des sessions garantit que les informations de session sont protégées contre les attaques et ne sont pas exposées à des parties non autorisées. Cela inclut l'utilisation de cookies sécurisés, l'expiration des sessions et la protection contre le vol de session. Exemple : Les sessions doivent être protégées par des cookies HttpOnly, Secure et avoir un délai d'expiration après une période d'inactivité.

2.4.4 Chiffrement des données

Les données doivent être chiffrées à la fois lorsqu'elles sont envoyées entre le client et le serveur (HTTPS) et lorsqu'elles sont stockées sur le serveur.

2.4.5 Traçabilité et journalisation

La journalisation consiste à enregistrer toutes les actions sensibles effectuées sur le système, afin de détecter des comportements suspects et garantir la non-répudiation.

2.5 Standards et outils associés

Certains outils et normes aident à évaluer et renforcer la sécurité d'une application web :

- OWASP Top 10 : liste actualisée des 10 menaces les plus critiques sur le web.
- WAF (Web Application Firewall) : pare-feu applicatif filtrant les requêtes malveillantes.
- Security Headers (CSP, X-Content-Type-Options, etc.) : pour renforcer la protection côté navigateur.
- Scanners de vulnérabilité : ZAP (OWASP), Burp Suite, etc.

2.6 État de l'art

L'objectif de cet état de l'art est de dresser un panorama des pratiques actuelles, des standards de sécurité reconnus, des outils disponibles et des tendances émergentes dans le domaine de la sécurité des applications web.

2.6.1 OWASP Top Ten Web Application Security Risks

L'article [24] présente les dix vulnérabilités les plus critiques affectant les applications web, telles que l'injection SQL, l'authentification cassée, l'exposition de données sensibles, et la mauvaise configuration de sécurité. Il fournit des explications détaillées, des exemples d'attaques, ainsi que des recommandations pour leur prévention. Il sert de référence essentielle pour les développeurs et les professionnels de la sécurité afin d'identifier et de corriger les failles courantes dans les applications web.

2.6.2 Burp Suite

Les auteurs dans [25] présentent une méthodologie pour utiliser Burp Suite, un outil de test de sécurité web très populaire. Ils expliquent comment configurer l'outil pour intercepter, manipuler, et analyser le trafic HTTP/HTTPS entre le navigateur et l'application cible. Leur approche inclut la réalisation de scans automatisés, l'exploitation manuelle de vulnérabilités, et l'utilisation d'extensions pour renforcer l'analyse. La publication met en évidence l'importance de l'approche combinée, automatique et manuelle, pour détecter un large éventail de vulnérabilités, et recommande d'intégrer cet outil dans un processus de test complet pour assurer une sécurité robuste.

2.6.3 Architecture Zero Trust

Les auteurs dans [26] analysent le modèle Zero Trust, basé sur le principe de ne faire confiance à aucun utilisateur ou appareil, même à l'intérieur du réseau. La revue explique comment ce modèle met en œuvre une vérification continue, des contrôles d'accès stricts, et une segmentation fine des réseaux pour limiter la surface d'attaque. Elle décrit aussi les défis techniques et organisationnels liés à sa mise en œuvre dans les environnements web, notamment la gestion des identités, la surveillance en temps réel, et l'intégration avec d'autres architectures de sécurité. Les recommandations incluent une approche progressive et l'adoption de solutions d'authentification forte.

2.6.4 OWASP ZAP

L'article [27] décrit l'utilisation de l'outil open-source OWASP ZAP pour automatiser les tests de sécurité des applications web. Il explique comment ZAP peut détecter automatiquement des vulnérabilités courantes comme les injections, XSS, et autres failles, grâce à ses fonctionnalités de scan actif et passif. Le document fournit des exemples pratiques d'intégration de ZAP dans des processus de développement continu pour améliorer la détection précoce des vulnérabilités.

2.6.5 SAST

L'article [28] étudie l'efficacité des outils d'analyse de code statique (SAST) pour identifier automatiquement des vulnérabilités dès la phase de développement. Howard compare plusieurs outils populaires, en évaluant leur capacité à détecter des injections, des dépassements de mémoire, et d'autres failles de sécurité. Il discute des limites, notamment les faux positifs et la nécessité d'un paramétrage précis. La conclusion souligne que l'intégration de SAST dans une démarche DevSecOps permet de réduire considérablement le nombre de vulnérabilités dans le code final, tout en favorisant une culture de sécurité dès la conception.

2.6.6 Arachni

Le travail présenté dans [29] décrit Arachni comme un scanner de vulnérabilités web open-source modulaire, conçu pour la rapidité et la précision. Avec une architecture extensible, il permet aux chercheurs de développer des modules pour tester des vulnérabilités spécifiques. L'étude montre qu'Arachni est efficace pour détecter des failles courantes telles que l'injection SQL, XSS, et CSRF. La facilité d'intégration dans des pipelines CI/CD et la capacité à gérer de grands volumes de scans en font un outil précieux pour les équipes de sécurité.

2.6.7 Wapiti

L'article [30] présente Wapiti comme un scanner open-source capable d'effectuer des tests de vulnérabilités automatisés sur des applications web. Il fonctionne en envoyant différentes requêtes pour détecter des failles telles que l'injection SQL, XSS, et les mauvaises configurations. La simplicité d'utilisation et le fait qu'il soit scriptable permettent une intégration facile dans des processus d'audit réguliers. Les auteurs recommandent Wapiti comme un outil complémentaire pour renforcer la posture de sécurité.

2.6.8 Nessus

L'article [31] compare Nessus, un scanner de vulnérabilités propriétaire, à d'autres outils open-source comme OpenVAS. Nessus est reconnu pour sa large base de données de vulnérabilités, sa rapidité, et ses rapports détaillés. Il couvre aussi bien les vulnérabilités réseau que celles spécifiques aux applications web. L'étude souligne que Nessus est efficace pour l'évaluation de la sécurité

globale, mais que son coût peut limiter son utilisation pour certaines petites organisations. La mise à jour régulière de ses plugins est essentielle pour garantir une détection précise.

2.7 Conclusion

Dans ce chapitre, nous avons abordé les principes clés de la sécurité des applications web ainsi que les menaces courantes et les techniques de protection essentielles. Dans le chapitre suivant nous réaliserons l'analyse et la modélisation de processus métier pour le cas d'étude que nous avons choisi « la gestion de dossiers médicaux ».

Chapitre 3

Analyse et Modélisation de processus métier

3.1 Introduction

La gestion efficace des dossiers médicaux constitue un enjeu stratégique pour les établissements de santé, notamment en clinique, où la rapidité d'accès à l'information, la précision des données et la conformité réglementaire sont essentielles. La complexité croissante des processus administratifs et médicaux nécessite une approche structurée pour analyser, modéliser et optimiser ces processus métiers. L'objectif principal de ce chapitre est d'étudier les différentes étapes de gestion des dossiers médicaux, d'identifier les acteurs impliqués, ainsi que les flux d'information et de documents qui y sont liés. À travers une démarche d'analyse approfondie et de modélisation, nous visons à proposer des solutions pour améliorer l'efficacité, la qualité des soins et la conformité réglementaire dans la gestion des dossiers médicaux en clinique.

3.2 Problématique

Au cours de mon stage au sein de la clinique médico-chirurgicale « Les castors », j'ai constaté que la gestion des dossiers médicaux repose principalement sur des documents papier, ce qui engendre plusieurs inconvénients tels que la perte d'informations, la difficulté d'accès rapide aux dossiers, ainsi qu'un risque accru d'erreurs ou de doublons. De plus, la gestion des rendez-vous est souvent non organisée, entraînant des retards, des annulations ou des chevauchements, affectant la qualité du service offert aux patients. Ces problématiques soulignent la nécessité d'optimiser les processus administratifs et médicaux. Afin d'améliorer la gestion des dossiers et des rendez-vous, j'ai proposé la mise en place d'un site web dédié, permettant une gestion électronique centralisée, plus efficace, sécurisée et accessible en temps réel. La problématique centrale réside donc dans la manière d'intégrer une solution numérique sécurisé pour remplacer la gestion traditionnelle papier, tout en assurant une meilleure organisation, une réduction des erreurs et une amélioration de la satisfaction des patients.

3.2.1 Diagramme cause et effet

Pour mieux visualiser les différentes causes des problématiques liées à la gestion des dossiers médicaux de la clinique « Les castors », nous construisons un diagramme de cause et effet.

Le diagramme de causes et effets est un outil visuel qui permet d'analyser les causes potentielles d'un problème. Il aide les équipes à explorer toutes les causes possibles et à identifier les causes racines les plus probables, ce qui peut contribuer à résoudre le problème de manière plus efficace.

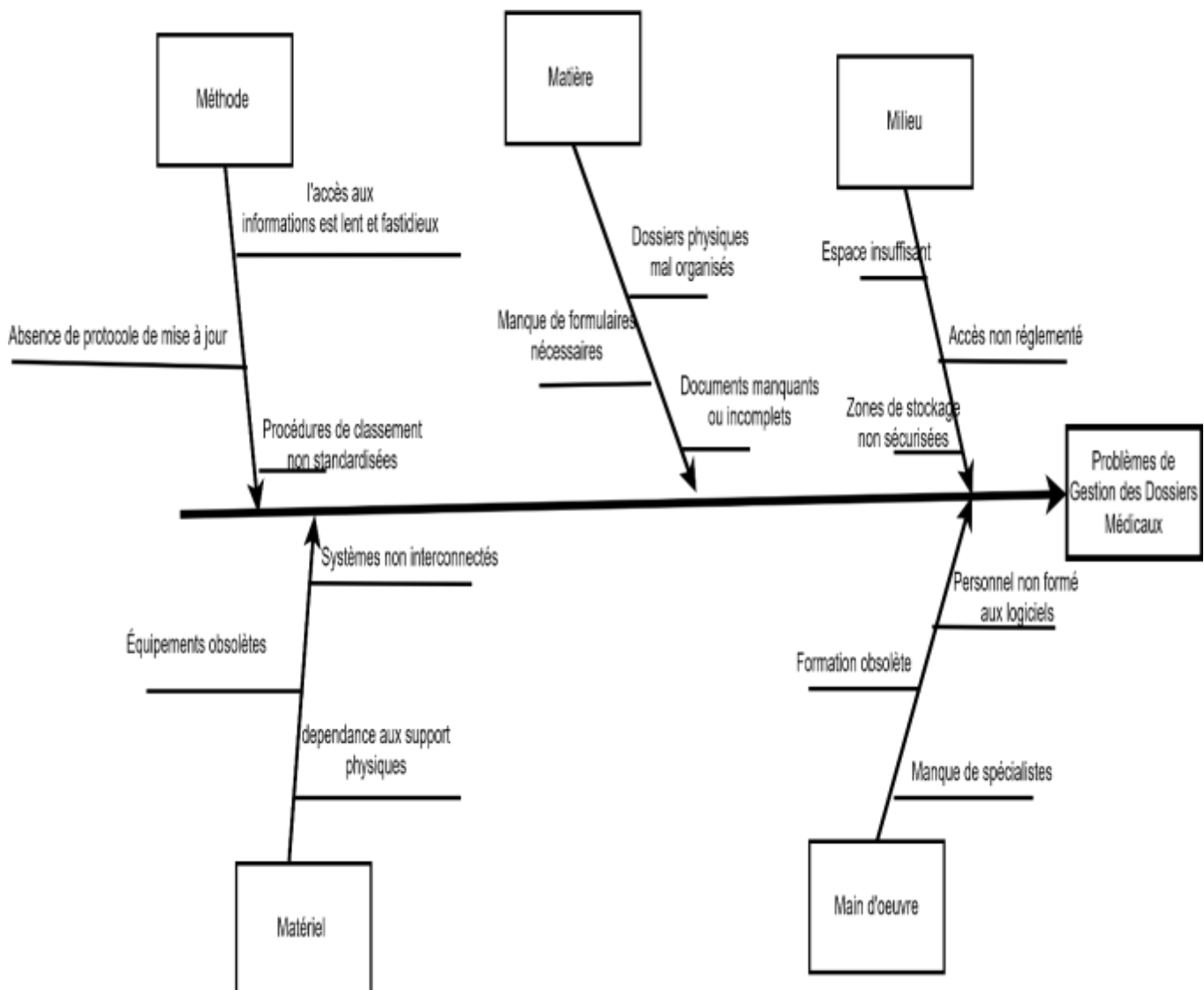


Figure 2: Diagramme de cause/effet

3.3 Modélisation processus métier

3.3.1 Définition des processus métiers

Un processus métier est une série d'activités interconnectées et coordonnées qui sont entreprises dans une organisation pour atteindre un objectif spécifique. Les processus métiers sont souvent conçus pour aider une organisation à atteindre ses objectifs stratégiques et opérationnels en améliorant l'efficacité, la qualité et la satisfaction des clients.[53]

3.3.2 Modélisation par BPMN

1. Présentation de BPMN

Business Process Model and Notation (BPMN) est un standard pour la modélisation de processus métier, qui fournit une notation graphique permettant de représenter d'une manière graphique très expressive les processus d'une organisation. Son objectif principal est de fournir une notation qui est facilement compréhensible par tous les utilisateurs professionnels, des analystes métier qui créent la version initiale du processus, aux développeurs techniques chargés de l'application de la technologie qui va exécuter ces processus, et finalement, les personnes qui permettront de gérer et de contrôler ces processus. Ainsi, BPMN crée un pont standardisé pour l'écart entre la conception des processus métiers et l'implémentation des processus. La notation consiste en un ensemble de symboles graphiques qui représentent des actions, des flux ou des comportements de processus.[54]

2. Processus de gestion de dossier médical

a) Processus de collaboration

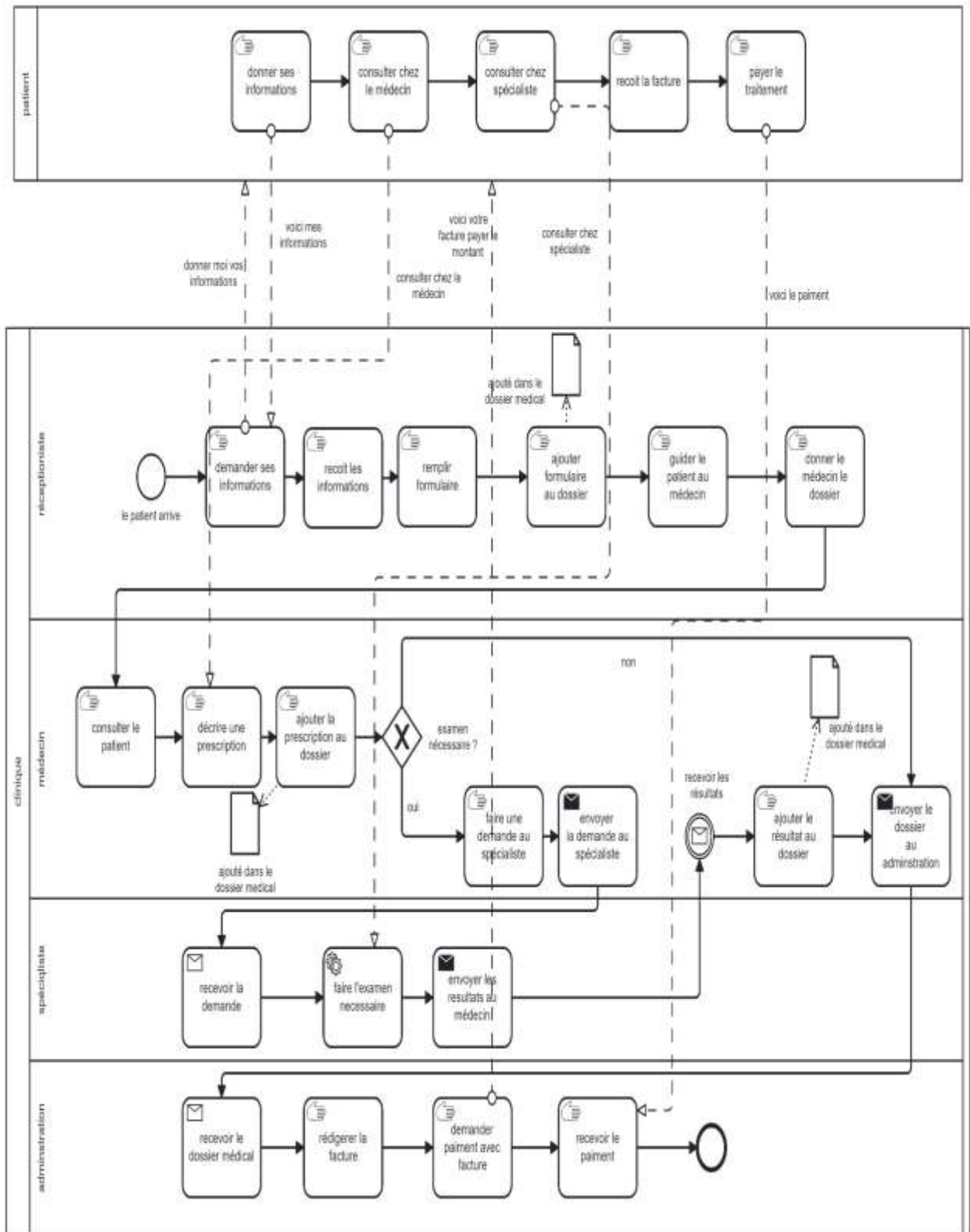


Figure 3:Processus métier de collaboration de gestion de dossier médical

b) Processus privé

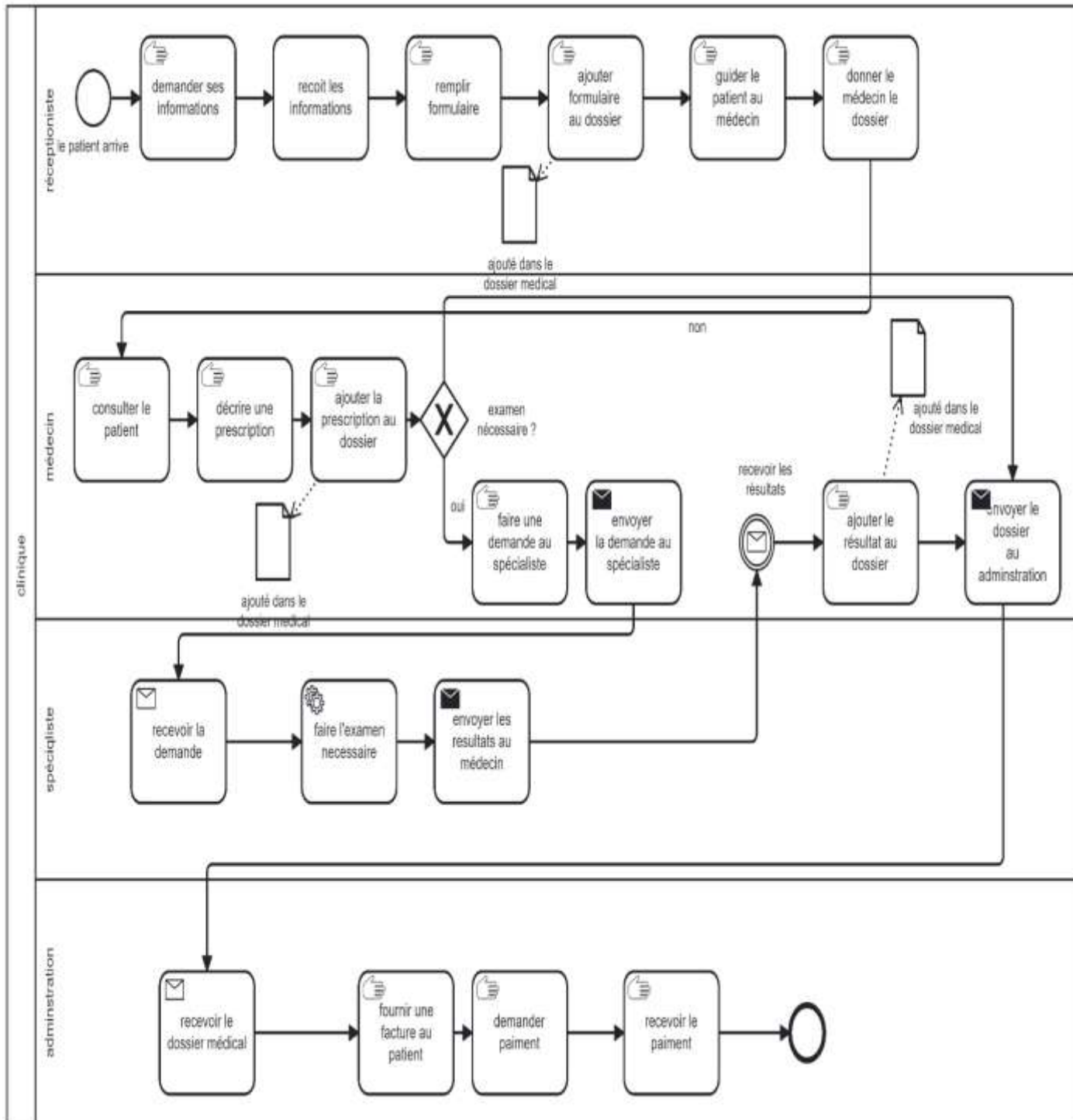


Figure 4: Processus métier privé de gestion de dossier médical

c) processus chorégraphie

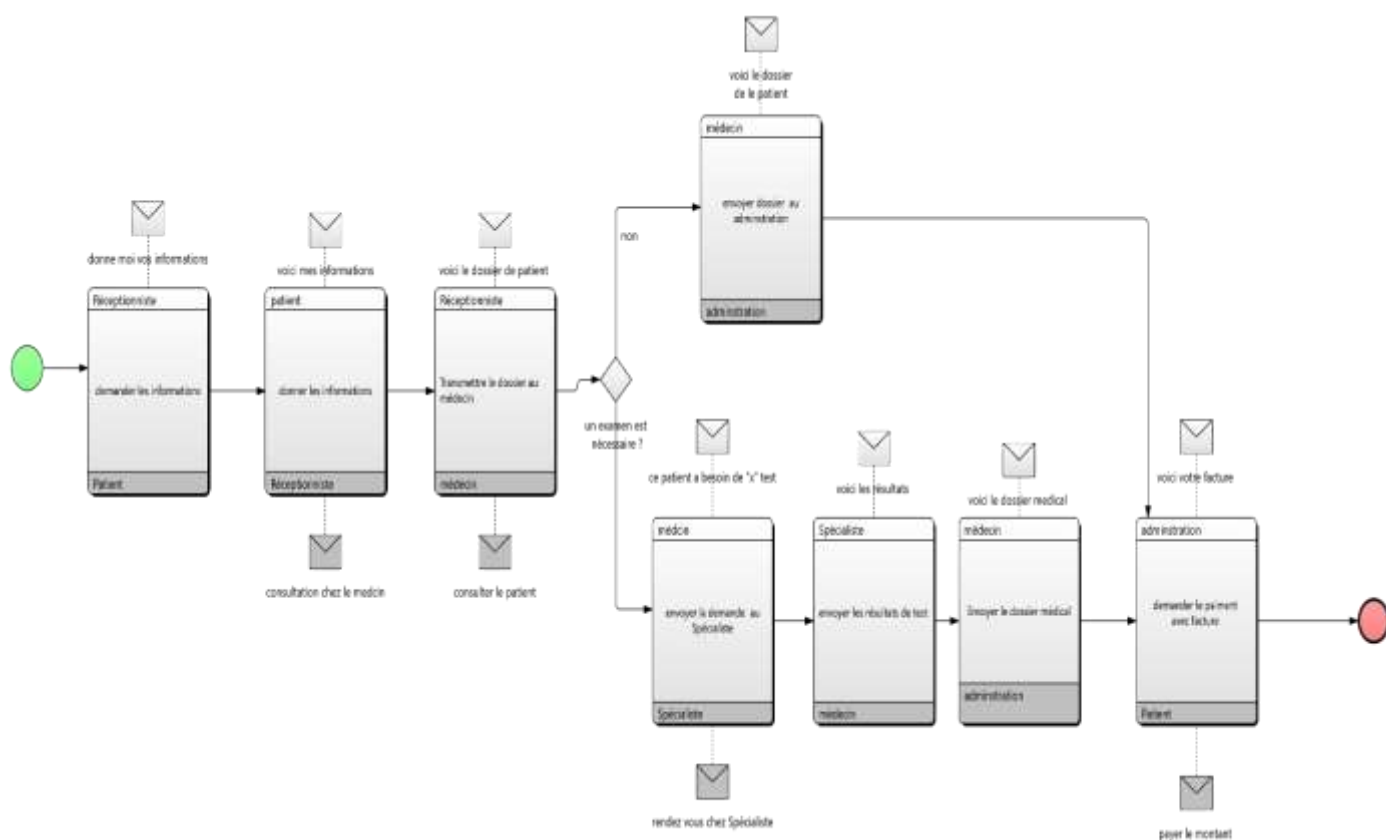


Figure 5:Processus métier de chorégraphie de gestion de dossier médical

Remarque : Des précisions complémentaires sur la modélisation des processus métier sont disponibles en annexe.

3.4 Conclusion

Dans ce chapitre, nous avons exploré les différentes problématiques liées à la gestion de dossier médical dans la clinique en permettant de décomposer les processus métiers liés à la gestion des dossiers médicaux, en mettant en lumière les étapes clés, les acteurs impliqués et les flux d'information. La compréhension de ces processus est essentielle pour optimiser l'efficacité et la qualité des soins, tout en garantissant la sécurité et la confidentialité des données. La mise en place

d'un système informatisé de gestion de ces dossiers est cruciale pour l'avenir, le chapitre suivant sera dédié à concevoir et modéliser la solution technique qui supportera ce processus.

Chapitre 4

Conception et Modélisation

4.1 Introduction

Le succès de toute application repose en grande partie sur une phase de conception et de modélisation rigoureuse, permettant d'assurer la cohérence, la robustesse et la sécurité du système développé. Dans ce contexte, ce chapitre décrit les démarches entreprises pour concevoir l'architecture de notre application web. Nous abordons tout d'abord la modélisation des besoins fonctionnels et non fonctionnels, afin de formaliser les attentes des utilisateurs et les contraintes techniques. Ensuite, nous présentons les différentes représentations graphiques, telles que les diagrammes UML.

4.2 Spécifications d'exigences de logiciel

4.2.1 Objet

L'objet de la SEL est de décrire les exigences fonctionnels et non fonctionnels de conception et l'implémentation de site web sécurisé avec la gestion d'authentification et d'autorisation cas d'étude gestion de dossier médical.

Les destinataires sont : Righi Dhiaie Eddine et Mkrarbech Fatima dans le cadre de PFE de M2 ISI.

1. Porté

Cette application nommé «Medy» permet aux patients de consulter leurs dossiers médicales en toute sécurité, faciliter la gestion de des rendez-vous médicales ,assurer une authentification sécurisée pour tous les utilisateurs afin de garantir la confidentialité des données, gérer les droits d'accès en fonction des rôles (patient, médecin, administrateur, spécialiste) pour assurer une gestion fine des permissions, ainsi enregistrer et stocker de manière sécurisée toutes les données médicales conformément aux normes en vigueur.

Ce site vise à améliorer la qualité de la prise en charge médicale en permettant un accès rapide et sécurisé aux informations médicales, tout en simplifiant la gestion des rendez-vous pour les patients et le personnel médical. Son utilisation vise à réduire les délais d'attente, améliorer la coordination entre les acteurs de la santé, et assurer la confidentialité et la sécurité des données sensibles. Elle a pour objectif principal de fournir un site web sécurisée permettant la gestion efficace des dossiers médicaux et la prise de rendez-vous en ligne.

2. Définitions, acronymes et abréviations

- UI : User Interface.
- Dossier médical : Ensemble des informations médicales d'un patient.

3. Références

- Pratique des dossiers médicaux, organisation de santé internationale.

4. Vue d'ensemble

Ce document décrit les exigences fonctionnelles et non-fonctionnelles, ainsi que les contraintes du système, pour assurer une compréhension commune entre toutes les parties prenantes.

4.2.2 Description générale

1. Environnement

Le système « Medy » est indépendante et parfaitement autonome, il n'est fait partie d'aucun système plus vaste.

2. Fonctions

Le logiciel doit permettre la gestion sécurisée des dossiers médicaux des patients ainsi que la prise de rendez-vous en ligne pour les professionnels de santé et les patients.

Les principales fonctions du logiciel comprennent :

- Gérer des dossiers médicaux : Le médecin et le spécialiste peuvent consulter le dossier et le modifier.

- Gérer des rendez-vous : Le patient et la réceptionniste peuvent prendre un rendez-vous pour une consultation chez le médecin, quand l'administrateur peut les gérer (créer, modifier).
- Gérer les utilisateurs et droits d'accès : L'administrateur doit créer les comptes des utilisateurs.
- Modifier le profil : Tous les utilisateurs de système peuvent modifier leurs profils.
- Gérer le calendrier : Le médecin et le spécialiste peuvent planifier leurs rendez-vous, interventions et disponibilités.
- Soumettre un avis : Le patient enregistre ses commentaires, retours ou préoccupations concernant sa prise en charge.
- Soumettre une plainte : Le patient à exprime formellement une insatisfaction ou une réclamation concernant la qualité des soins.
- Réinitialiser le mot de passe : Tous les utilisateurs peuvent faire ça pour renforcer leurs sécurités.

3. Caractéristiques des utilisateurs

Les utilisateurs de cette application sont divers et possèdent des niveaux d'instruction, d'expérience et de connaissances techniques variés. Cette section décrit leurs caractéristiques générales pour justifier les exigences fonctionnelles et techniques du système.

- Médecins et Spécialistes (d'analyse laboratoire, de radiologie, de vaccin)
 - ✓ Niveau d'instruction : Diplômés en médecine, généralement à l'aise avec l'utilisation d'outils informatiques.
 - ✓ Expérience : Expérience variable, allant de jeunes médecins à des praticiens expérimentés.
 - ✓ Connaissances techniques : Bonne maîtrise de l'utilisation des outils informatiques et des logiciels médicaux. Certains peuvent nécessiter une formation spécifique pour la prise en main de l'application.
 - ✓ Raisons : La simplicité d'utilisation et la clarté de l'interface facilitent leur adoption, tout en assurant la sécurité des données sensibles.
- Patients
 - ✓ Niveau d'instruction : Variable ; allant de personnes peu familiarisées avec la technologie à des utilisateurs avancés.
 - ✓ Expérience : Peu d'expérience technique pour certains, voire aucune, avec la gestion numérique de leur santé.

- ✓ Connaissances techniques : Connaissances limitées en informatique, nécessitant une interface accessible, intuitive et guidée.
- Administrateurs
 - ✓ Niveau d’instruction : Formation en gestion ou en informatique, souvent titulaires d’un diplôme en administration ou en gestion de systèmes.
 - ✓ Expérience : Expérience en gestion administrative, en gestion de systèmes d’information et en sécurité des données.
 - ✓ Connaissances techniques : Bonne maîtrise des outils informatiques, des réseaux, et des principes de sécurité informatique.
- Réceptionnistes
 - ✓ Niveau d’instruction : Diplôme de niveau Bac ou équivalent, avec formation spécifique à la gestion administrative.
 - ✓ Expérience : Expérience dans le domaine médical ou administratif, familiarité avec la gestion de rendez-vous et la communication avec les patients.
 - ✓ Connaissances techniques : Connaissance de base de l’utilisation des logiciels de gestion et de communication.

4. Contraintes

a) Politiques réglementaires

Le site web «Medy» doit réaliser et appliquer les politiques réglementaires suivants :

- Les administrateurs doivent administrer applications.
- Les administrateurs doivent gérer les rendez-vous.
- Les administrateurs doivent gérer les comptes.
- Les administrateurs doivent consulter les plaintes.

b) Limites imposées par le matériel

L'application doit être hébergée dans un serveur.

c) Interfaces avec les autres applications

Il n'existe actuellement aucune interface avec d'autres applications.

d) Fonctions de vérification :

- Vérification des rendez-vous : L'application doit vérifier si un créneau est libre pour éviter les erreurs et les conflits de planning.
- La vérification si le l'utilisateur (médecin, patient...) est déjà inscrit dans la base des données.

f) Fonctions de contrôle :

- Contrôle d'accès aux données : L'application doit mettre en place des mécanismes de contrôle d'accès aux données médicales (laravel middleware) et aux fonctionnalités de prise de rendez-vous basés sur les rôles et les autorisations.
- Contrôle de session : Le contrôle des sessions consiste à limiter la durée pendant laquelle un utilisateur peut rester connecté sans activité. Si l'utilisateur ne réalise aucune action pendant 30 minutes, la session sera automatiquement expirée ou invalidée.

k) Considérations relatives à la sécurité et à la sûreté

- Gestion d'authentification et d'autorisation : L'application doit garantir la sécurité des données des utilisateurs en utilisant l'authentification avec captcha, hachage.

4.2.3 Exigences spécifiques

Nous détaillons les spécifications et les exigences relatives aux interactions de l'application avec les autres systèmes ou applications externes.

1. Exigences des interfaces externes

a) Interfaces avec les utilisateurs

Logo, utilisation de couleur bleu.

b) Interfaces avec le matériel

PC puissant pour naviguer le site facilement.

c) Interfaces avec le logiciel

Système d'exploitation Windows.

2. Exigences fonctionnelles

Les exigences fonctionnelles définissent les fonctionnalités clés que l'application doit inclure pour répondre aux besoins des utilisateurs.

- **Classe d'utilisateur 1 : Le patient**

- Authentification :
- Prise un rendez-vous
- Consultation de dossier médical
- Consultation des rendez-vous
- Modification de profil
- Gestion de rendez-vous
- Réinitialisation de mot de passe

- **Classe d'utilisateur 2 : Le médecin**

- Authentification
- Consultation des rendez-vous
- Gestion des dossiers médicales
- Gestion de calendrier
- Modification de profil
- Réinitialisation de mot de passe (même que les autre utilisateurs).

- **Classe d'utilisateur 3 : La réceptionniste**

- Authentification (même que les autre utilisateurs).
- Prise des rendez-vous
- Consultation des rendez-vous
- Modification de profil (même que les autre utilisateurs)
- Réinitialisation de mot de passe (même que les autre utilisateurs)

- **Classe d'utilisateur 4 : L'administrateur**

- Authentification (même que les autres utilisateurs).
 - Gestion des rendez-vous
 - Gestion des utilisateurs
 - Réinitialisation de mot de passe
- **Classe d'utilisateur 4 : Le spécialiste (même fonctionnalités que celle de médecin)**

3. Exigences de performance

- ✓ L'application doit répondre aux requêtes des utilisateurs en moins de 5 secondes pour 95% des interactions courantes (consultation, recherche, enregistrement).
- ✓ Les opérations plus lourdes, comme la recherche avancée, doivent être terminées dans un délai maximum de 5 secondes.
- ✓ L'application doit être sécurisée et protéger les informations des utilisateurs.

4. Contraintes de conception

L'interface de l'application doit être simple et intuitive pour permettre une utilisation facile par les utilisateurs.

a) Attributs

- L'application doit être disponible 24h/24, 7j/7, avec un taux de disponibilité minimal de 99,9 %.
- L'application doit garantir la sécurité des données des patients en utilisant des mécanismes pour empêcher les attaques.
- L'application doit être fiable et ne doit pas planter ou se bloquer.

4.3 Modélisation

L'étape de modélisation consiste à décrire dans un langage graphique formel, des besoins, les solutions fonctionnelles et techniques qui leur correspondent afin de bénéficier des atouts évidents qu'apporte un bon modèle. Il est nécessaire que les développeurs et les utilisateurs soient convenablement formés et accompagnés et travaillent selon le même cadre méthodologique.

4.3.1 Présentation d'UML

L'Unified Modeling Language (UML) est un langage de modélisation standardisé utilisé pour spécifier, visualiser, construire et documenter les composants d'un système logiciel. Il permet de représenter graphiquement les aspects structurels et comportementaux d'un système, facilitant ainsi la communication entre les différentes parties prenantes du processus de développement logiciel. UML offre un ensemble de diagrammes normalisés qui aident à modéliser divers aspects du système, tels que la structure, l'architecture, les interactions et la dynamique.[32]

4.3.2 Les Diagrammes d'UML

Les diagrammes UML sont des représentations graphiques normalisées qui permettent de modéliser visuellement différents aspects d'un système logiciel, tels que la structure, le comportement ou les interactions, afin d'améliorer la compréhension, la conception et la communication entre les membres de l'équipe de développement.[33]

4.4 Méthode RUP

La méthode Rational Unified Process (RUP) est un processus de développement logiciel itératif et incrémental qui fournit un cadre structuré pour gérer le cycle de vie du projet, en mettant l'accent sur la gestion des risques, la qualité, et la livraison progressive du logiciel. Elle repose sur une architecture évolutive, une gestion rigoureuse des exigences, et une collaboration étroite entre les acteurs du projet.[34]

4.4.1 Avantages de la méthode RUP

La méthode RUP présente plusieurs avantages importants, notamment :[35]

- Meilleure maîtrise des risques et des coûts : Privilégie une approche itérative permettant d'identifier et de traiter les risques dès les premières phases du projet, réduisant ainsi les coûts liés aux erreurs en fin de développement.
- Meilleure gestion de projet : RUP fournit un cadre clair pour la planification, le suivi et le contrôle du projet, facilitant la gestion des délais, des coûts, et des ressources.
- Flexibilité et adaptabilité : Grâce à ses cycles itératifs, RUP permet d'adapter le développement en fonction des retours clients et des changements de contexte, améliorant ainsi la satisfaction client.

4.5 La démarche de la modélisation

Notre démarche s'appuie sur la méthode RUP où nous allons présenter les différents diagrammes d'UML des modèles suivants :

4.5.1 Modèle de cas d'utilisation

Un modèle de cas d'utilisation est une représentation qui décrit, sous forme narrative ou graphique, les interactions entre des acteurs externes et le système, afin de spécifier les fonctionnalités fournies par le système et leur valeur pour l'utilisateur.[36]

1. Diagramme de profile

La notion du profil UML fournit des mécanismes permettant aux concepteurs d'étendre les diagrammes d'UML pour traiter des aspects particuliers d'un système. Les extensions sont organisées en profils. Un profil est un ensemble cohérent d'extensions applicables à un domaine. Il permet de spécialiser le méta-modèle UML, il définit des stéréotypes et les metaclasses en utilisant les objets d'UML.[37]

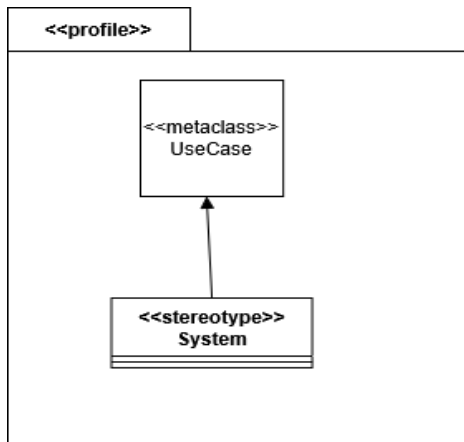


Figure 6 : Diagramme de profile

2. Diagramme contexte statique

Le diagramme de contexte est un diagramme UML qui montre le système dans son environnement, en mettant en évidence ses limites et ses interactions avec les acteurs extérieurs [38]. Ses composants sont :

- Le Système : Représente l'ensemble du système à étudier.
- Les Acteurs Externes : Représentent les entités extérieures qui interagissent avec le système.
- Les Flux d'Informations : Représentent les interactions ou échanges entre le système et les acteurs.

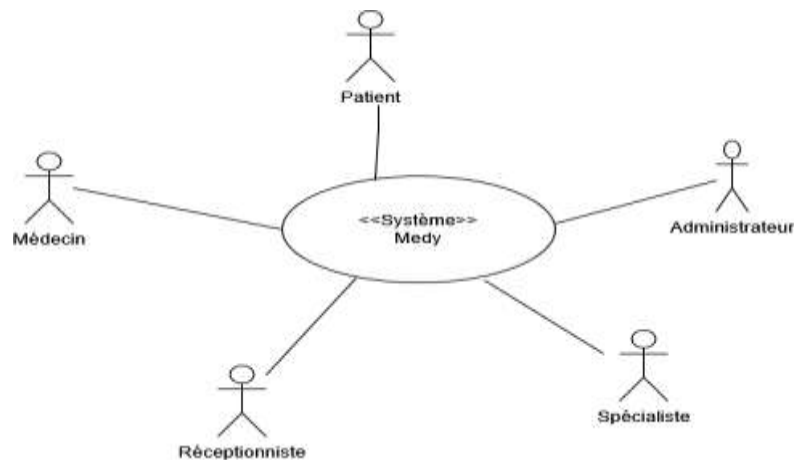


Figure 9 : Diagramme de contexte statique

3. Diagramme de package

Un diagramme de package montre l'organisation des éléments dans un système en regroupant ces éléments en packages, ainsi que leurs dépendances.[32]

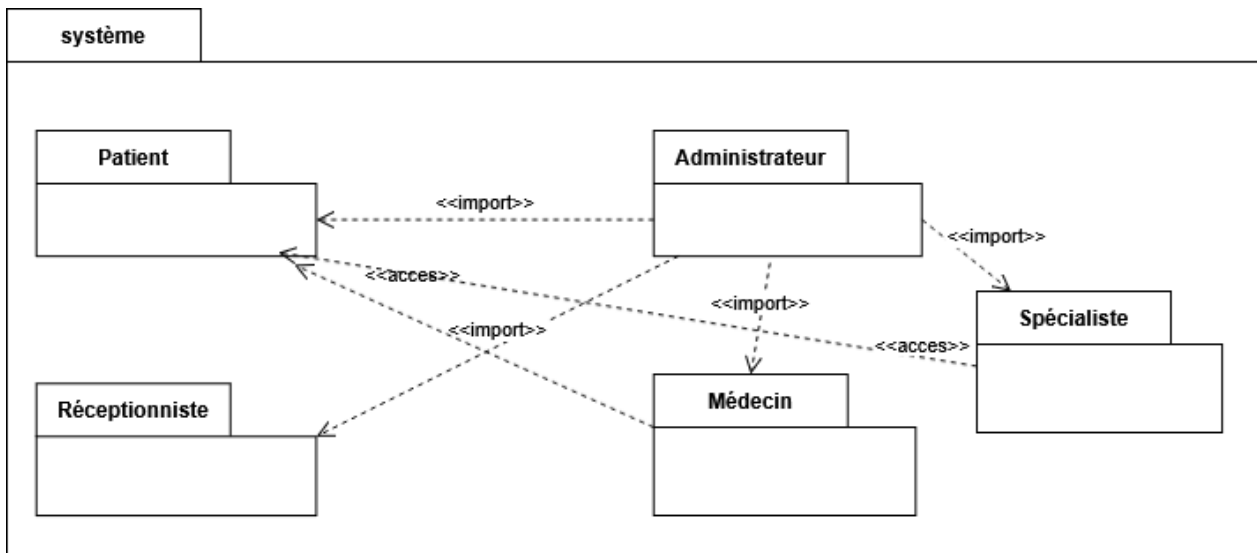


Figure 7 : Diagramme de package

4. Diagramme cas d'utilisation

Un diagramme de cas d'utilisation (« use case ») représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier.[39]

a) Diagramme cas d'utilisation pour le package patient

Le diagramme de cas d'utilisation du package Patient illustre les différentes actions que peut effectuer un patient dans le système. Ces actions incluent la gestion de leur profil, la prise de rendez-vous, la consultation de leurs rendez-vous, et la gestion de leur historique médical et la soumission de leur avis et la plainte. Ce diagramme permet de visualiser les interactions principales entre un patient et le système.

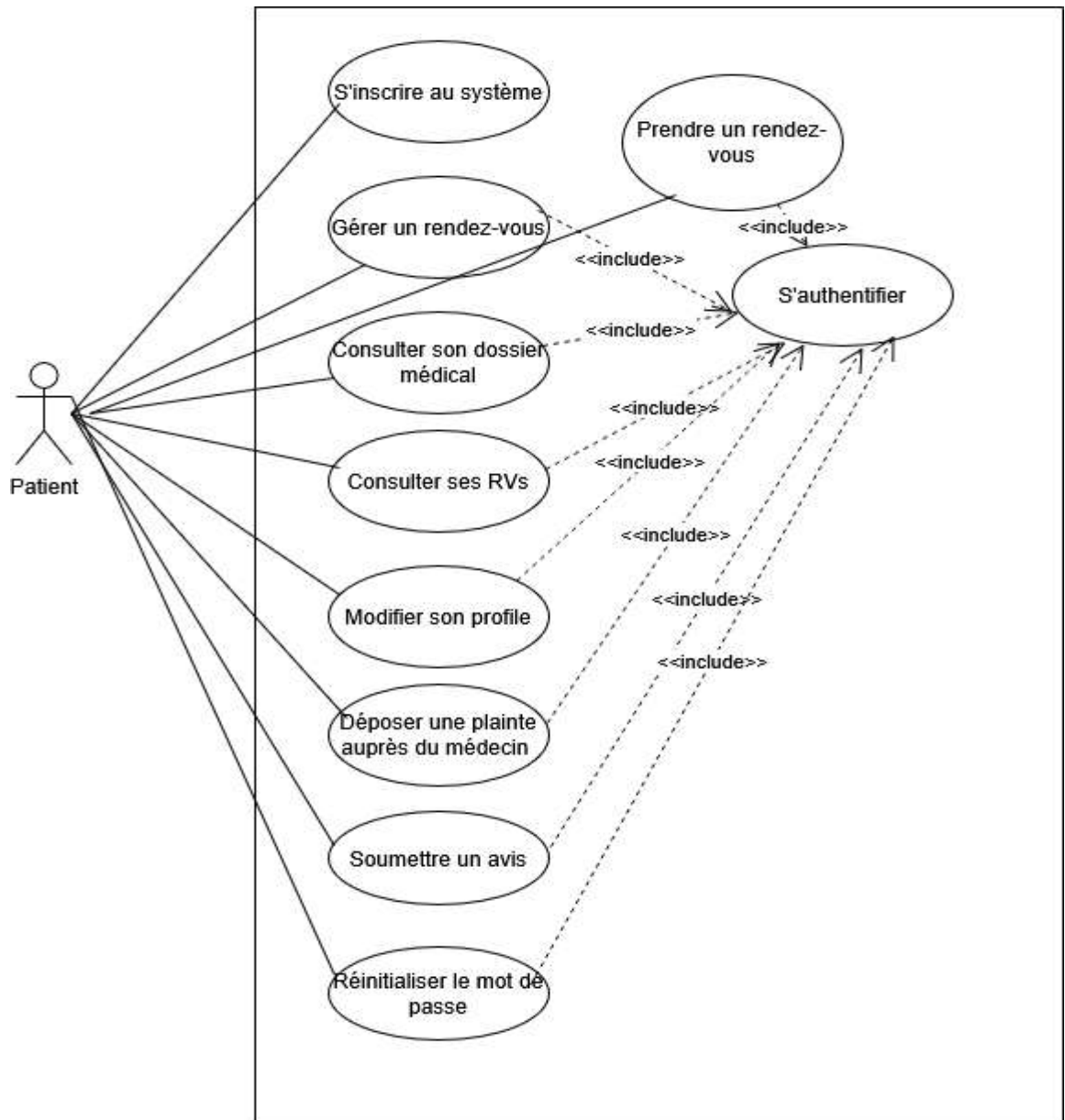


Figure 8 : Diagramme de cas d'utilisation de patient

b) Diagramme cas d'utilisation pour le package médecin et spécialiste

Le diagramme de cas d'utilisation du package le spécialiste et le médecin qui hérite ces fonctionnalités décrit les actions et interactions principales qu'un médecin et le spécialiste peut effectuer dans le système. Ces actions incluent la gestion de leurs profils, la consultation des rendez-vous, la consultation de l'historique des patients, la gestion des dossiers médicaux la gestion de calendrier et l'évaluation de rendez-vous.

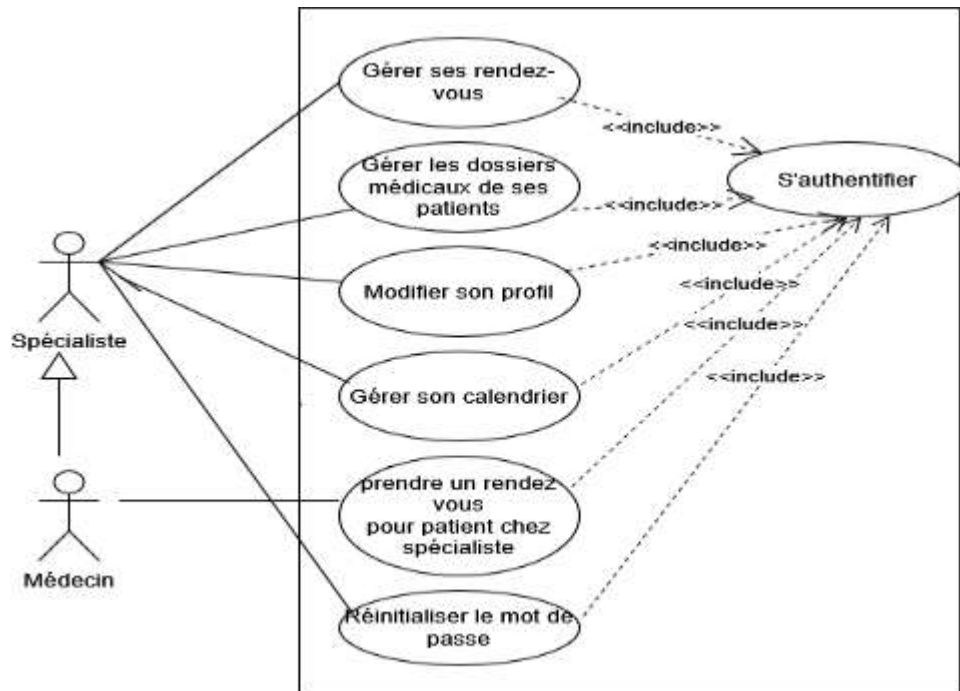


Figure 9 : Diagramme de cas d'utilisation de médecin et spécialiste

c) Diagramme cas d'utilisation pour le package réceptionniste

Le diagramme de cas d'utilisation du package Réceptionniste illustre les fonctionnalités principales accessibles à cet acteur dans le système. La réceptionniste est responsable de la gestion des rendez-vous, de l'accueil des patients et de la consultation des plannings. Ce diagramme permet de visualiser ses tâches quotidiennes pour assurer un service efficace et fluide.

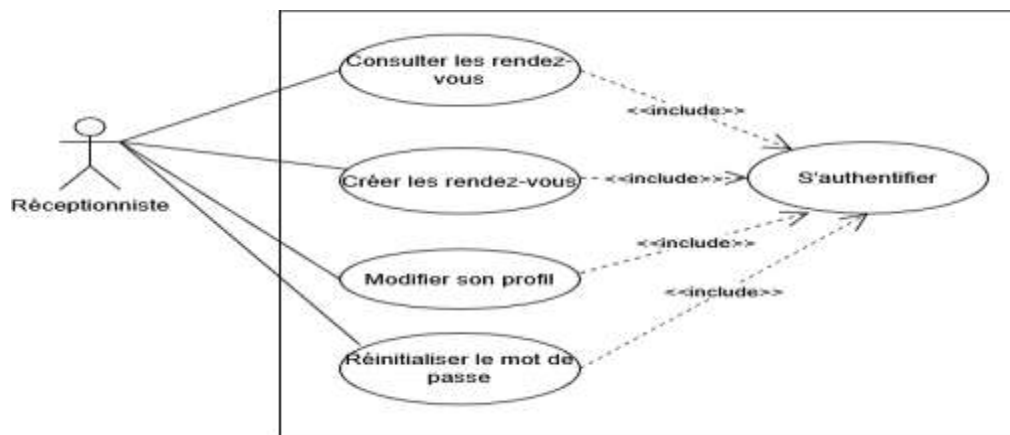


Figure 10 : Diagramme de cas d'utilisation de réceptionniste

d) Diagramme cas d'utilisation pour le package administrateur

Le diagramme de cas d'utilisation du package Administrateur décrit les fonctionnalités principales accessibles à cette entité dans le système. L'administrateur a pour rôle de gérer les utilisateurs (médecins, réceptionnistes, etc.), de consulter les dossiers médicaux, de gérer les rendez-vous, visualiser les plaintes et de consulter les journaux de sécurité. Ce diagramme permet de visualiser aussi les responsabilités et actions clés de l'administrateur pour assurer la cohérence, la sécurité du système.

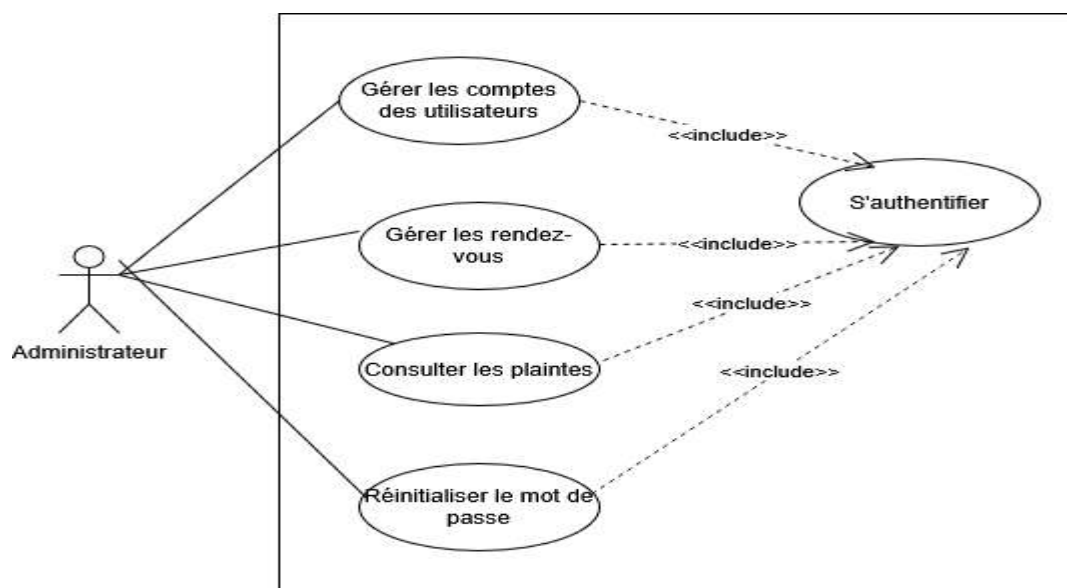


Figure 11 : Diagramme de cas d'utilisation d'administrateur

5. Description textuelle

A1. Description textuelle des cas d'utilisation de patient

a) Description textuelle de s'inscrire

- **Titre** : S'inscrire au l'application.
- **Résumé** : Ce cas d'utilisation permet au patient de s'inscrire à l'application.
- **Acteur principal** : Patient.
- **Précondition** : L'utilisateur dispose d'un terminal avec accès à Internet, l'application est accessible.

- **Scénario nominale :**

1. Le patient ouvre l'application ou le site web.
2. Il sélectionne l'option S'inscrire.
3. Le système affiche un formulaire d'inscription demandant : nom, âge, groupe sanguin, numéro de téléphone, numéro d'identification national, adresse email, mot de passe
4. Le patient remplit le formulaire avec ses données.
5. Le patient soumet le formulaire.
6. Le système vérifie la validité des données (format, champs obligatoires).
7. Si toutes les données sont valides, le système crée un nouveau compte.
8. Le système envoie un email de confirmation ou de vérification.
9. Le patient reçoit la confirmation.
10. Le patient cliquer sur le lien envoyé
11. Le patient est redirigé vers son compte après avoir cliqué sur le lien de confirmation.

- **Enchaînements alternatifs**

- ◆ A1: si le patient possède un compte.
L'enchaînement A1 démarre à l'étape 4 du scénario nominal.
- 5. Appuie sur « déjà enregistré ? ».
- Le scénario nominal reprend au point 10.

- **Enchaînements d'erreur**

- ◆ E1: Données invalides ou manquantes.
L'enchaînement E1 démarre à l'étape 7 du scénario nominal.
- 8. Le système affiche des messages d'erreur précisant les champs à corriger. par exemple « veuillez sélectionner un élément de la liste ».
- ◆ L'utilisateur doit corriger les erreurs et resoumettre.

- **Postcondition :** Un nouveau compte crée.

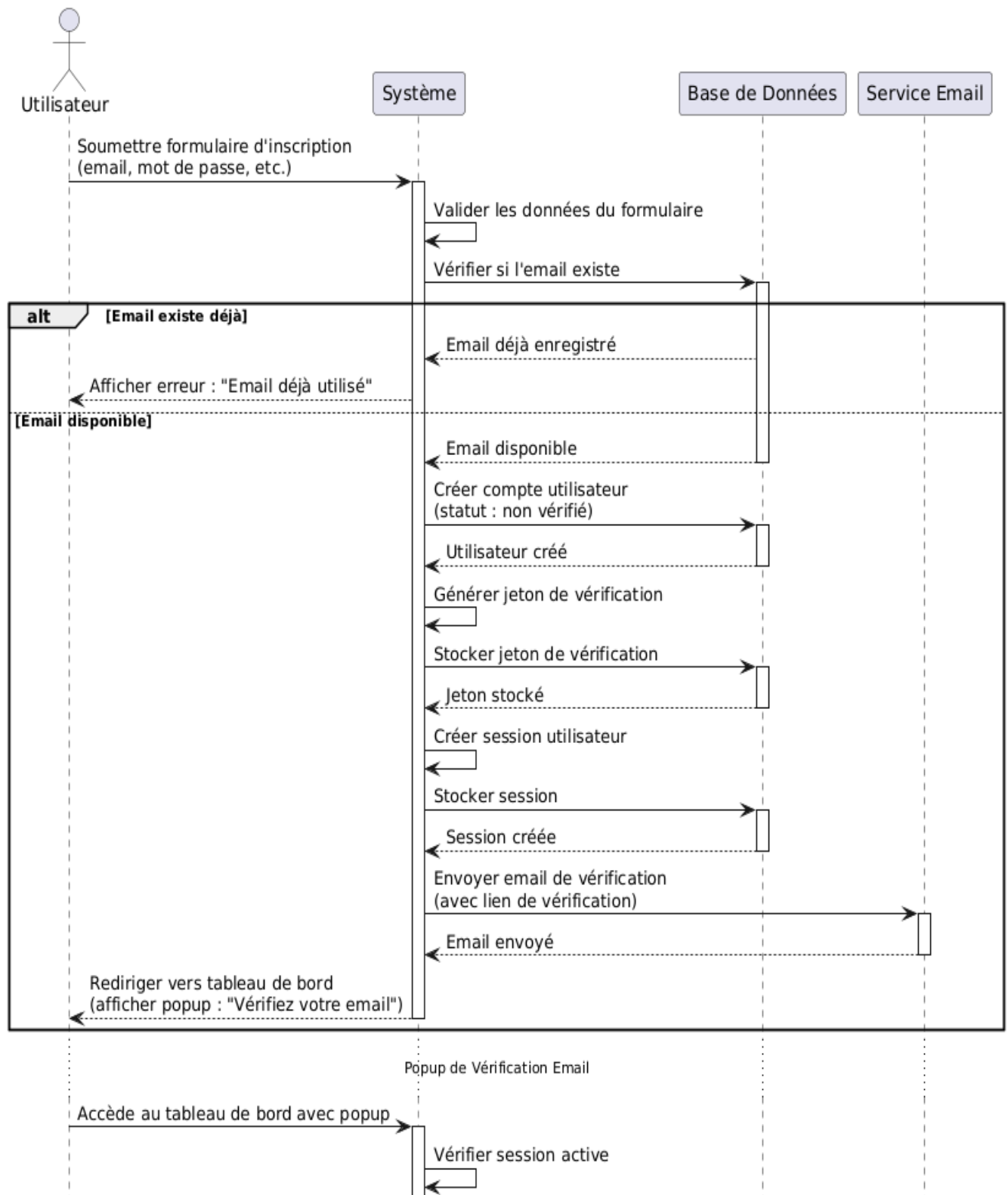


Figure 12: Diagramme de séquence de l'inscription d'un patient partie 1

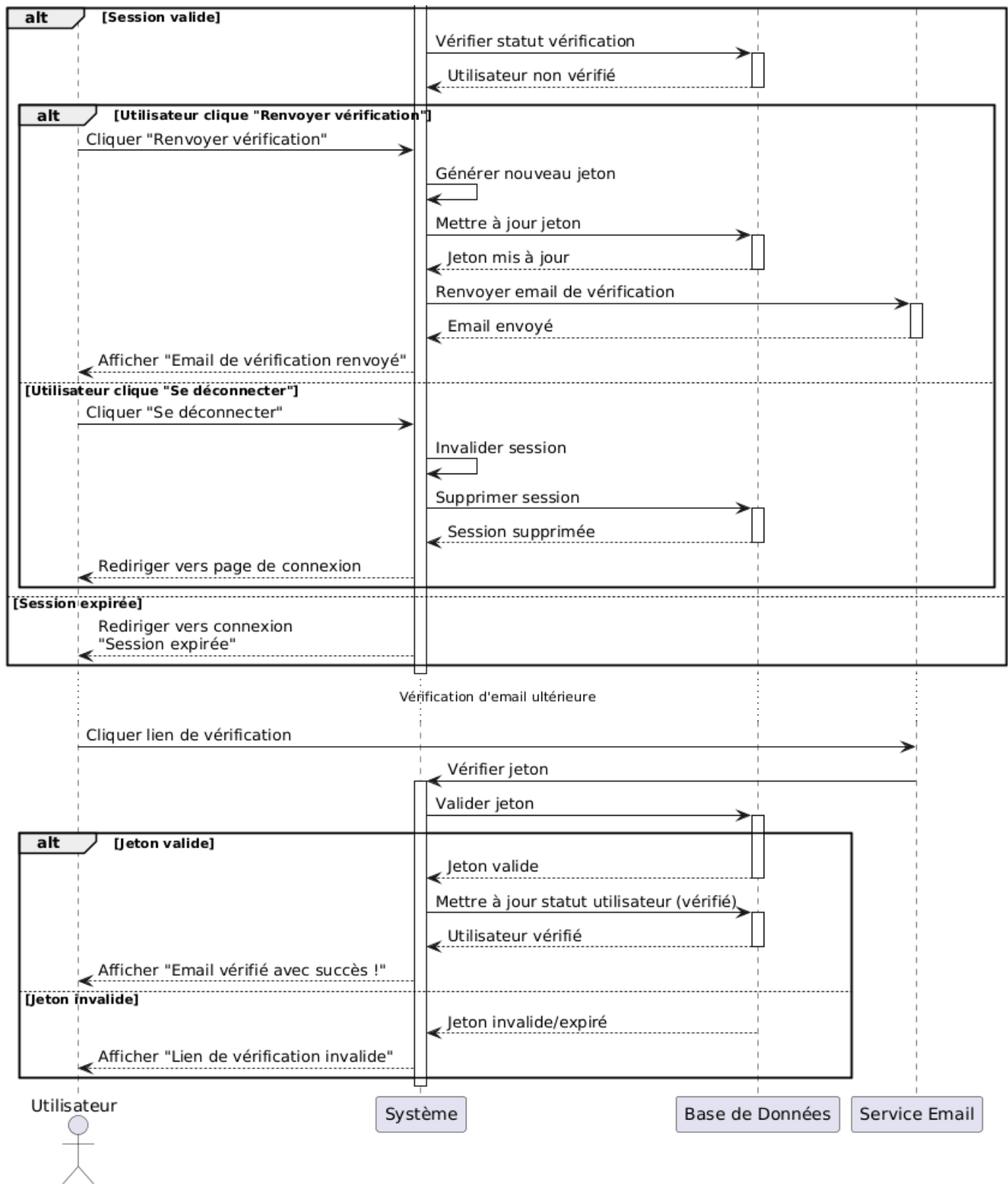


Figure 13: Diagramme de séquence de l'inscription d'un patient partie 2

b) Description textuelle de prendre un rendez-vous

- **Titre** : Prendre un rendez-vous.
- **Résumé** : Ce cas d'utilisation permet au patient de planifier un rendez-vous pour avoir une consultation avec le médecin.
- **Acteur principal** : Patient.
- **Précondition** : Le patient possède un compte dans l'application.
- **Scénario nominale** :
 1. Le patient se connecte au portail patient.
 2. Le patient appuie sur bouton « Prendre un rendez-vous ».
 3. Le système affiche la liste des spécialités disponibles.
 4. Le patient choisit une spécialité.
 5. Le système affiche la liste des médecins disponibles.
 6. Le patient choisit un médecin.
 7. Le patient choisit une date
 8. Le système affiche les créneaux horaires.
 9. Le patient choisit une date/heure disponible.
 10. Le patient confirme la réservation
 11. Le système enregistre le de rendez-vous.
- **Enchaînements alternatifs**
 - ◆ A1: Si le patient ne sait pas quel médecin il choisit.
L'enchaînement A1 démarre à l'étape 5 du scénario nominal.
 6. Il coche «Any available doctor».Le scénario nominal reprend au point 7.
 - A2: Le patient souhaite changer le créneau après avoir choisi.
L'enchaînement A2 démarre à l'étape 7 du scénario nominal.
 8. le patient modifie sa sélectionLe scénario nominal reprend au point 9.
- **Enchaînements d'erreur**

E1: Créneau non disponible lors de la sélection

L'enchaînement E1 démarre à l'étape 9 du scénario nominal.

10. Le créneau n'est plus disponible.
11. Le patient revient à la liste des créneaux et modifie sa sélection.

E2: Session expiré.

Après la session termine (après 30 min), il doit s'authentifier.

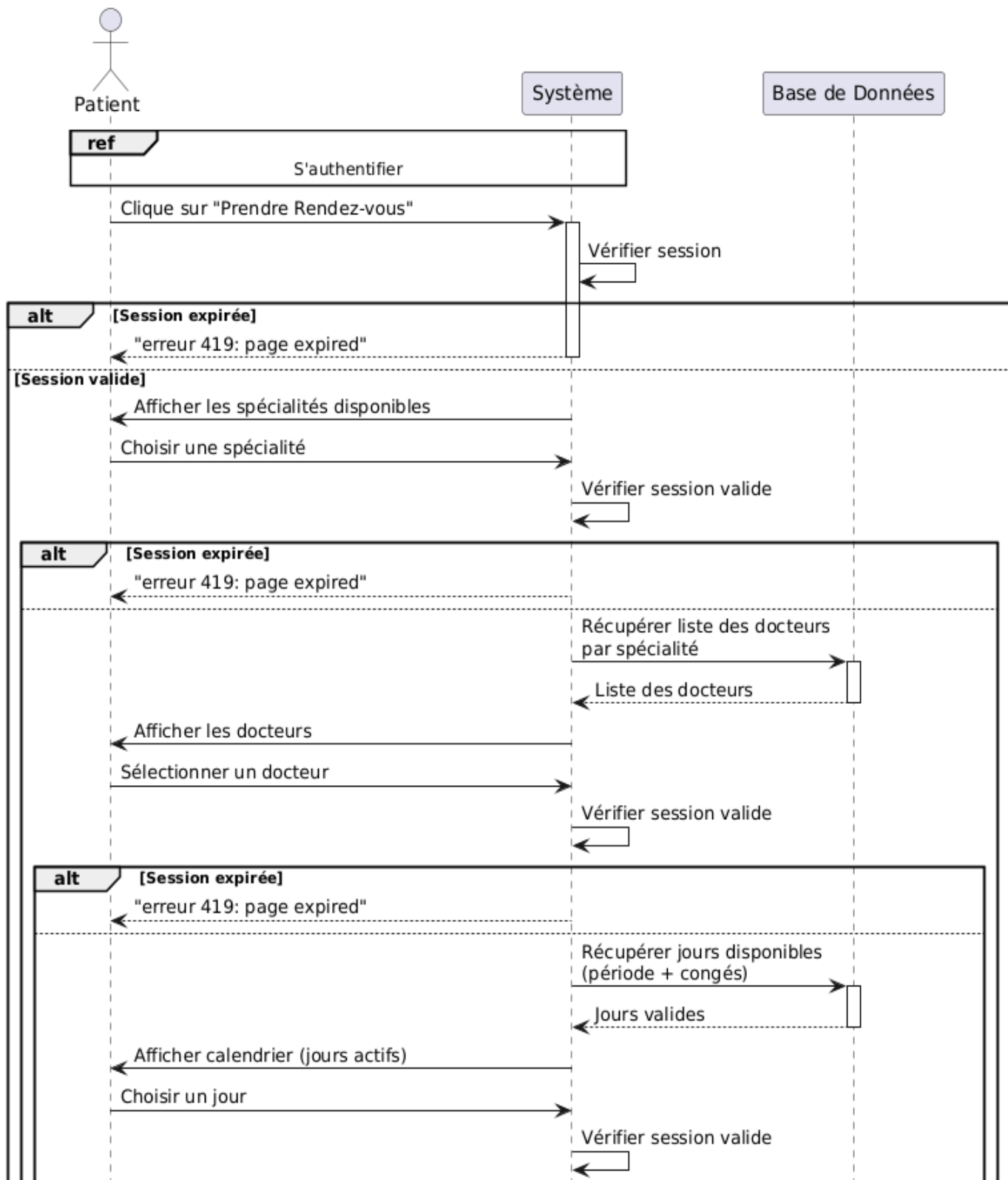


Figure 14: Diagramme de séquence d'un patient prendre un rendez-vous partie 1

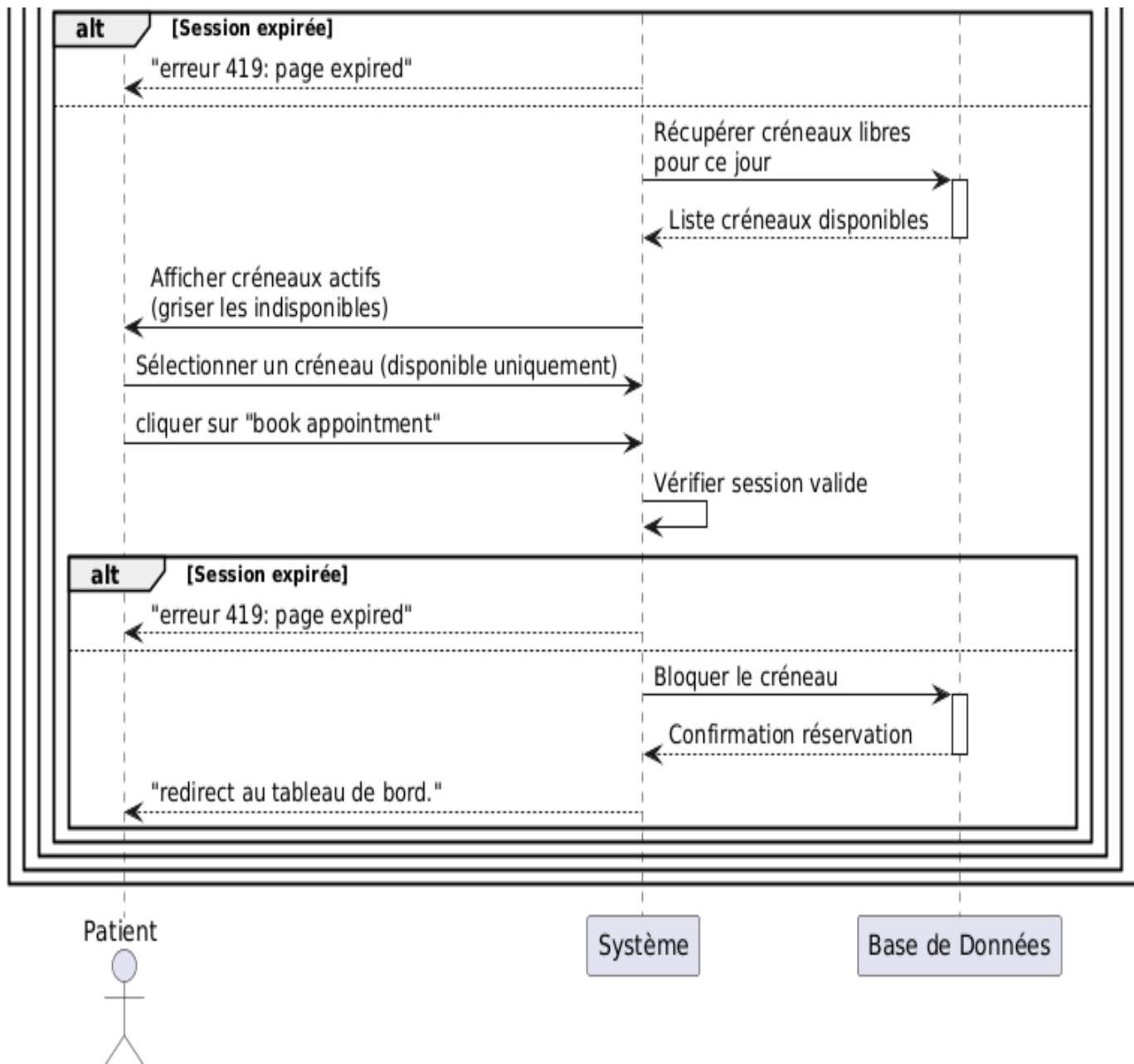


Figure 15: Diagramme de séquence d'un patient prendre un rendez-vous partie 2

c) Description textuelle de consulter son dossier médical

- **Titre** : Consulter son dossier médical.
- **Résumé** : Ce cas d'utilisation permet au patient, après authentification, d'accéder à l'ensemble de ses données médicales stockées dans le système, avec une navigation simple.
- **Acteur principal** : Patient.

- **Précondition** : Le patient doit avoir compte utilisateur dans l'application.
- **Scénario nominale** :
 1. Le patient se connecte à l'application ou au portail patient.
 2. Il accède à la section Dossier médical dans le menu principal.
 3. Le système affiche une vue d'ensemble structurée du dossier médical, comprenant plusieurs sections : Résultats d'examens (analyses, radiographies, Vaccinations), ordonnances, notes du médecin, certificats, allergies, maladies chroniques.
 4. Le patient peut sélectionner une section spécifique pour voir plus en détail.
 5. Le système affiche les détails correspondants, avec la possibilité de faire défiler ou de naviguer entre différentes sous-sections.
 6. Le patient quitte la consultation du dossier médical ou revient à la page principale.
- **Enchaînements alternatifs**
 - ◆ A1:si le patient oublie son mot de passe.
L'enchaînement A1 démarre à l'étape 1 du scénario nominal.
 - 2. Le patient utilise la fonction "Mot de passe oublié".
 - 3. Il fournit son email.
 - 4. Le système envoie un lien pour réinitialiser l'accès.
 - 5. Le patient suit la procédure et se reconnecte avec ses nouveaux identifiants.
 - Le scénario nominal reprend au point 2.
- **Enchaînements d'erreur**
 - E1: Échec de l'authentification après plusieurs tentatives.
L'enchaînement E1 démarre à l'étape 1 du scénario nominal.
 - 2. L'utilisateur tente de se connecter plusieurs fois.
 - 3. Le système bloque temporairement l'accès pour des raisons de sécurité.
 - 4. Le message d'erreur indique : « Too many failed attempts. Please try again later. »
 - E2: Session expirée.
Après la session termine (après 30 min), il doit s'authentifier.
- **Postcondition** : afficher le dossier médical.

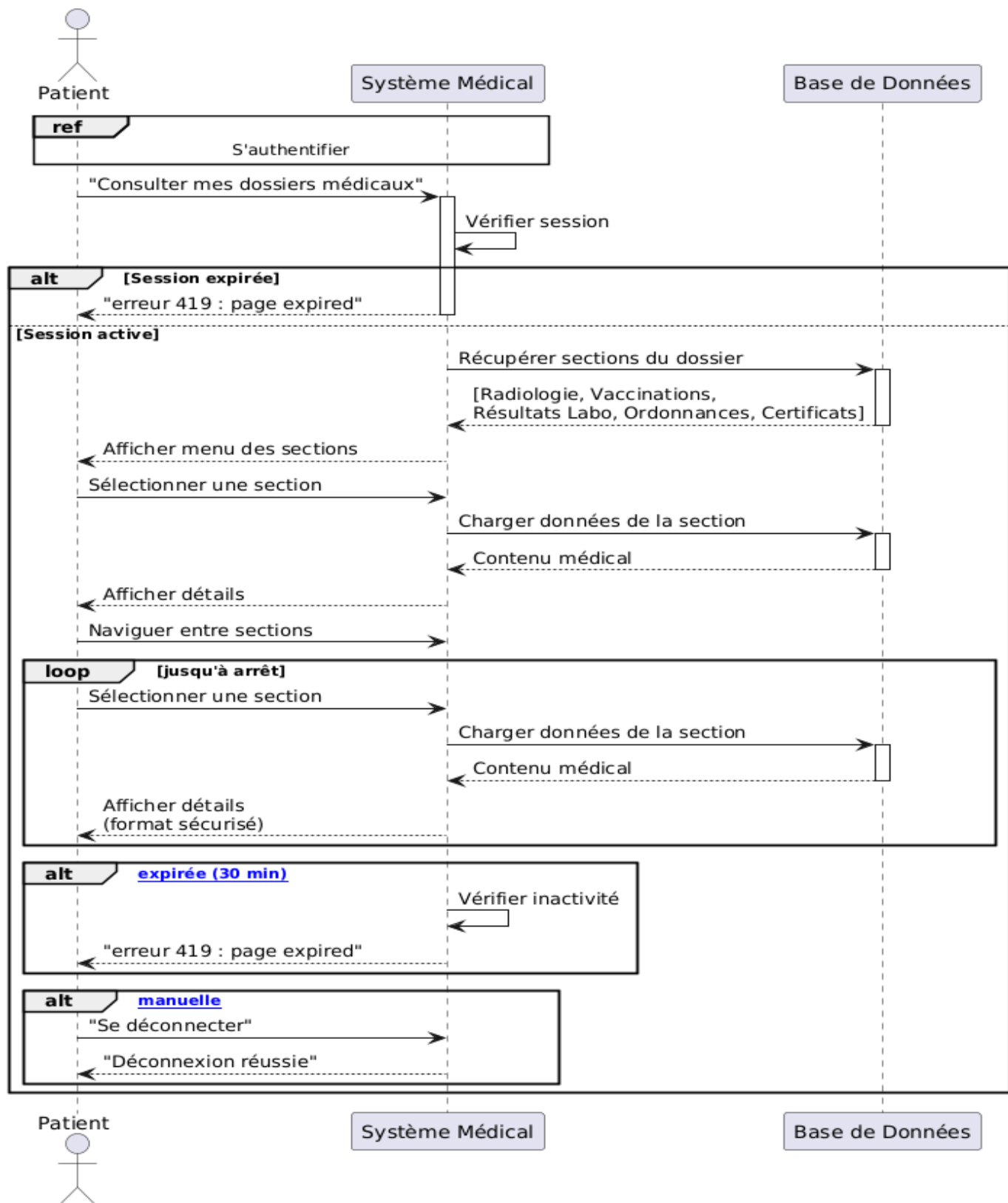


Figure 16: Diagramme de séquence de patient pour consulter son dossier médical

d) Description textuelle de consulter ses rendez-vous

- **Titre** : Consulter son historique de rendez-vous.
- **Résumé** : Ce cas d'utilisation permet au patient de pouvoir visualiser ses rendez-vous passés, avec les informations pertinentes (date, heure, type de rendez-vous, médecin, etc.).
- **Acteur principal** : Patient.
- **Précondition** : Le patient doit avoir compte utilisateur dans l'application.
- **Scénario nominale** :
 1. Le patient se connecte à l'application web avec ses identifiants.
 2. L'application affiche la page d'accueil du patient.
 3. Le patient accède à la section « my appointments » dans le menu de navigation.
 4. L'application affiche une liste de rendez-vous. Chaque rendez-vous est présenté avec : date, heure, motif, médecin, état.
- **Enchaînements alternatifs**
 - ◆ A1: le patient veut voir le prochain et le dernier rendez-vous.
L'enchaînement A1 démarre à l'étape 4 du scénario nominal.
 - 5. Le patient revient à son tableau de bord, ils sont affichés dans des sections différentes.
- **Enchaînements d'erreur**

E1: Échec de l'authentification après plusieurs tentatives.

L'enchaînement E1 démarre à l'étape 1 du scénario nominal.

2. L'utilisateur tente de se connecter plusieurs fois.
3. Le système bloque temporairement l'accès pour des raisons de sécurité.
4. Le message d'erreur indique : « Too many failed attempts. Please try again later. »

E2: Session expiré.

Après la session termine (après 30 min), il doit s'authentifier

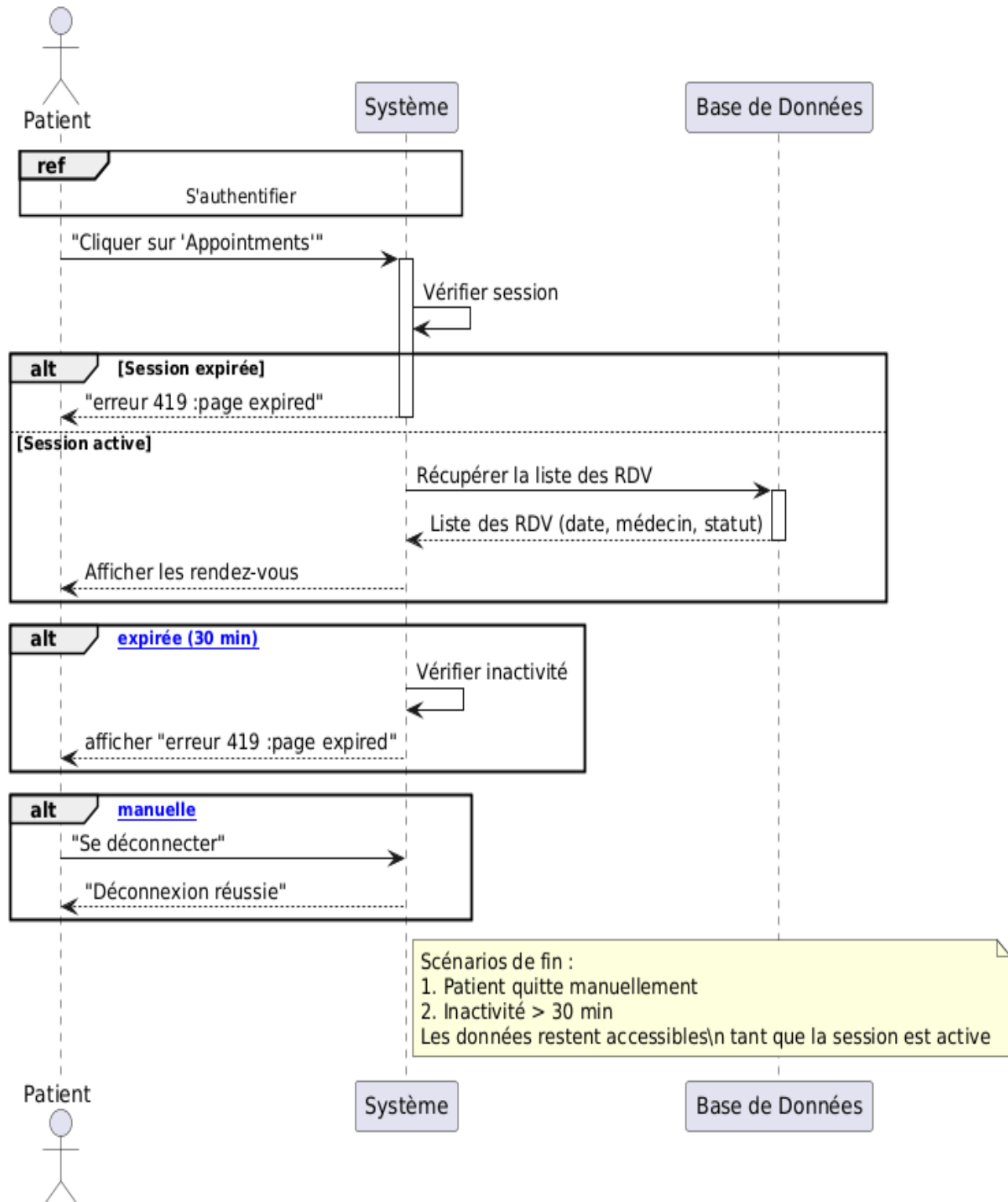


Figure 17: Diagramme de séquence de patient pour consulter sa liste des rendez-vous

e) **Description textuelle de modifier le profil**

- **Titre** : Modifier le profil.
- **Résumé** : Ce cas d'utilisation Permet à l'utilisateur de mettre à jour ses informations personnelles (nom, mail).
- **Acteur** : Patient, médecin, réceptionniste, administrateur, spécialiste.
- **Précondition** : Le patient doit avoir compte utilisateur dans l'application.
- **Scénario nominale** :
 1. L'utilisateur connecte à son compte.
 2. Il sélectionne l'option « profil » dans le bouton qui porte le nom d'utilisateur à coté de menu.
 3. Le système affiche le formulaire de modification, pré-rempli avec les données actuelles.
 4. L'utilisateur modifie ses informations (par exemple, change son nom ou son mot de passe).
 5. L'utilisateur clique sur « Save ».
 6. Le système vérifie la validité des nouvelles données.
 7. Le système vérifie la validité des nouvelles données.
 8. Si toutes les données sont valides, le système enregistre les modifications.
- **Enchaînements alternatifs**
 - ◆ A1: L'utilisateur veut modifier son mot de passe.

L'enchaînement A1 démarre à l'étape 3 du scénario nominal.

 4. Il remplit les champs et clique sur « enregistrer »
 5. Le système vérifie la validité des informations (ancien mot de passe correct, nouveau mot de passe conforme aux règles)
 6. Si les données sont valides, le système met à jour le mot de passe
 7. Le système affiche un message de confirmation : « mots de passe mis à jour avec succès ».
- **Enchaînements d'erreur**

E1: Données invalide.

L'enchaînement E1 démarre à l'étape 4 du scénario nominal.

5. L'utilisateur remplit un champ (par exemple, adresse email) avec un format incorrect.
6. Lorsqu'il clique sur « Enregistrer », le système affiche un message d'erreur précisant le problème (ex.«The password field must be at least 8 characters.»).
7. L'utilisateur doit corriger l'erreur pour pouvoir sauvegarder.

E2: Session expiré.

Après la session termine (après 30 min), il doit s'authentifier.

- **Postcondition** : Le patient a un nouveau profil.

f) Description textuelle de déposer une plainte

- **Titre** : Déposer une plainte.
- **Résumé** : Ce cas d'utilisation Permet au patient d'écrire un commentaire relatif à ses soins.
- **Acteur** : Patient.
- **Précondition** : Le patient doit avoir compte utilisateur dans l'application.

Scénario nominale

1. Le patient sélectionne l'option "Déposer une plainte".
2. Le patient se connecte à son compte.
3. Le patient saisit le formulaire de plainte.
4. Le système enregistre la plainte dans la base de données.
5. Le système affiche un message de confirmation au patient.

- **Enchaînements alternatifs**

A1: Patient remplit un formulaire incomplet ou incorrect

1. Lors de la soumission, le système vérifie la complétude et la validité des données.
2. Si certaines informations sont manquantes ou incorrectes :
3. Le système affiche un message d'erreur précisant les champs à corriger.
4. Le patient peut compléter ou corriger ses données.
5. Une fois corrigé, il peut resoumettre.

- **Enchaînements d'erreur**

E2: Session expiré.

Après la session termine (après 30 min), il doit s'authentifier.

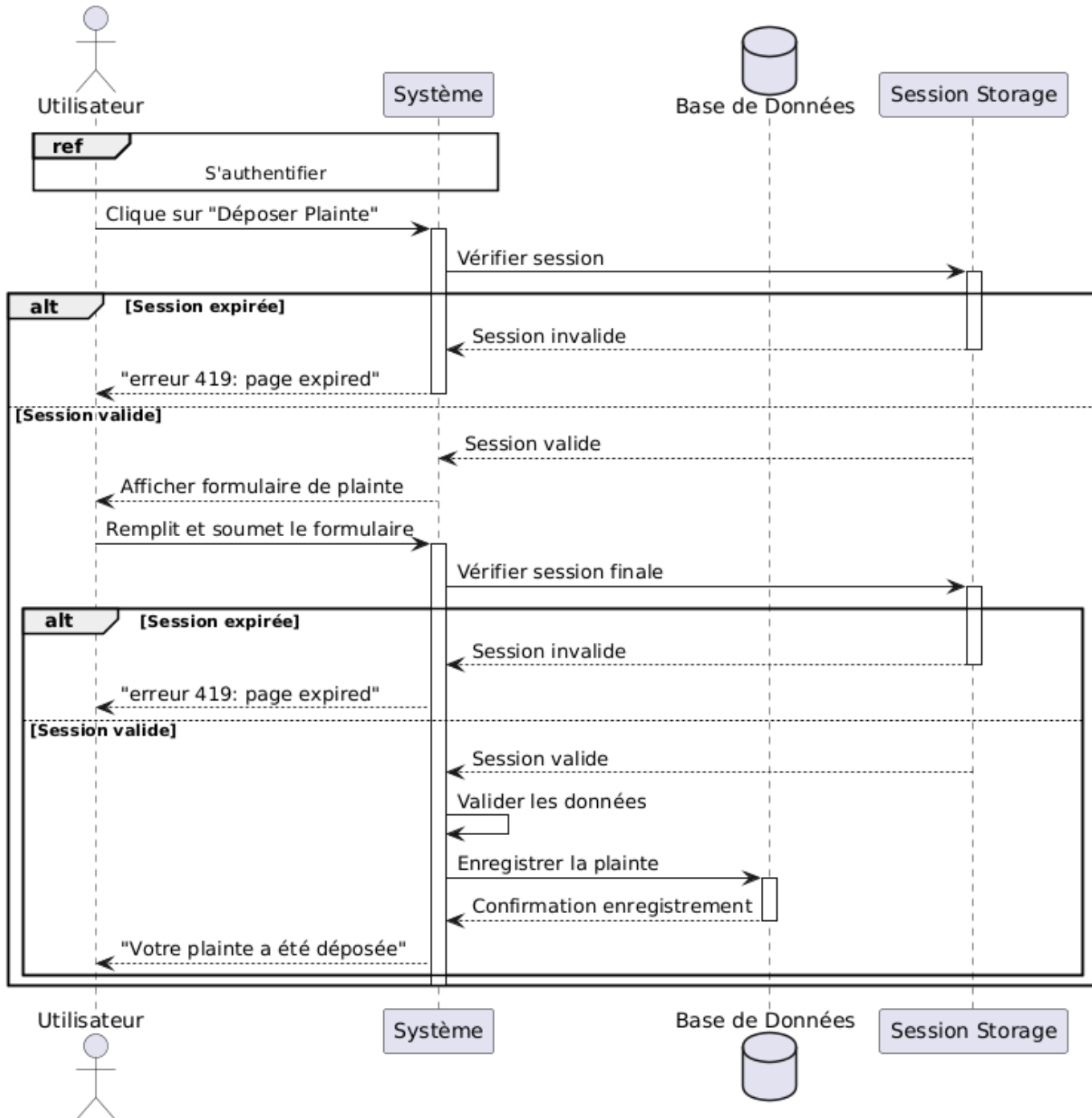


Figure 18: Diagramme de séquence de soumettre une plainte

g) Description textuelle de soumettre un avis

- **Titre** : Soumettre un avis.
- **Résumé** : Ce cas d'utilisation Permet au patient d'écrire un avis relatif à ses soins.
- **Acteur** : Patient.
- **Précondition** : Le patient doit avoir compte utilisateur dans l'application.

Scénario nominal

1. Le patient se connecte à son compte.
2. Le système récupère les rendez-vous non évalués.
3. Le système affiche fenêtre d'évaluation
4. Le patient évalue avec les étoiles.
5. Le patient peut s'il souhaite, écrire une note
6. Le système enregistre l'évaluation dans la base de données.
7. Le système affiche un message de confirmation au patient.

- **Enchaînements alternatifs :**

- **Enchaînements d'erreur**

E1: Session expiré.

Après la session termine (après 30 min), il doit s'authentifier.

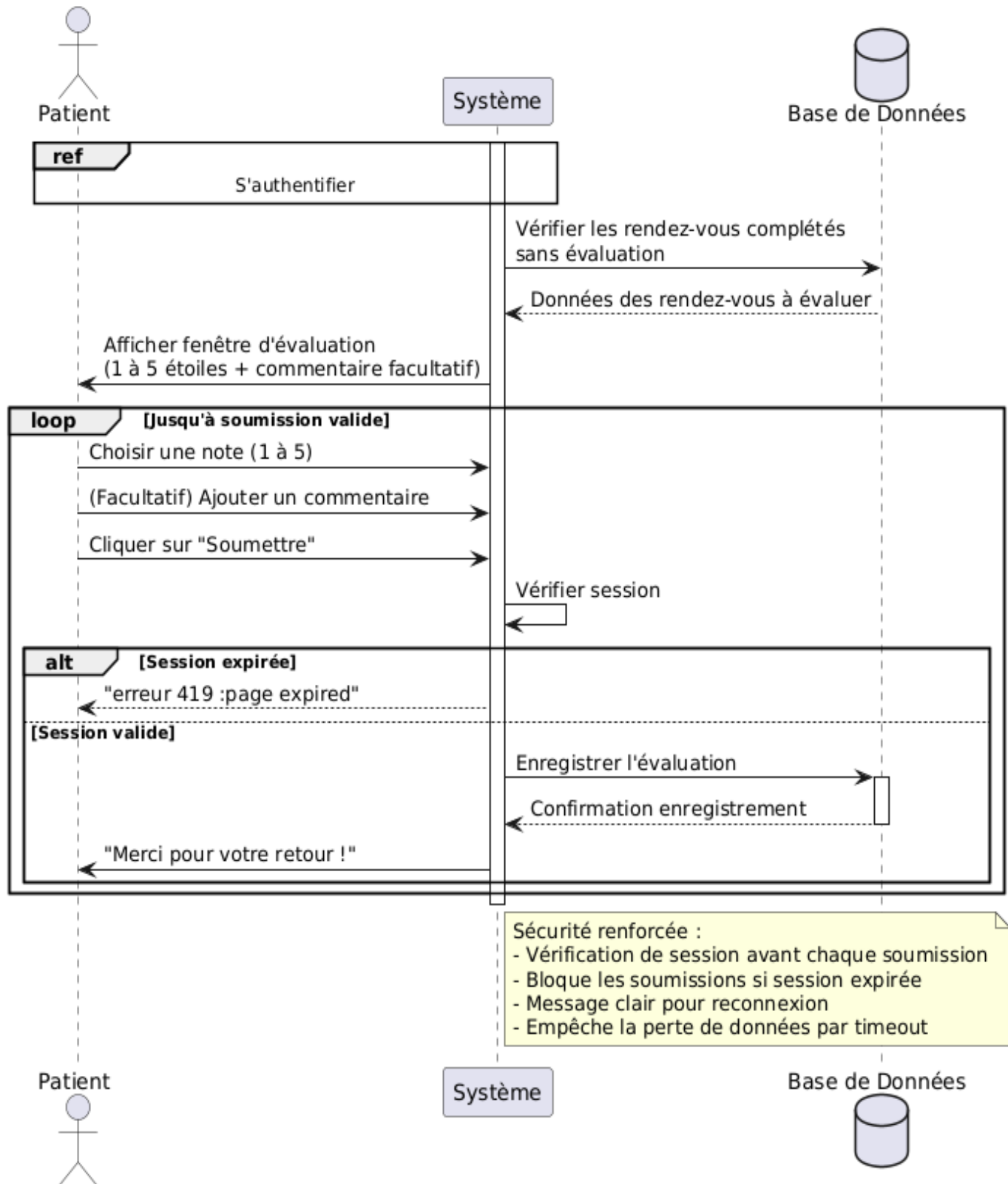


Figure 19: Diagramme de séquence de patient soumettre un avis

h) Description textuelle de gérer rendez-vous

- **Titre** : Gérer un rendez-vous.
- **Résumé** : Ce cas d'utilisation Permet au patient de gérer son rendez-vous.
- **Acteur** : Patient.
- **Précondition** : Le patient doit avoir un rendez-vous.

Scénario nominal

1. Le patient se connecte à son compte.
 2. Le système affiche son tableau de bord avec ses rendez-vous.
 3. Le patient choisi modifier ou annuler.
 4. Le système modifier le rendz-vous s'il le reste plus que 24 heures.
 5. Le système annuler le rendez-vous s'il le total de ses annulations des rendz-vous est moins que 3 fois.
- **Enchaînements alternatifs** :
 - **Enchaînements d'erreur** :
 - E1: Session expiré.
Après la session termine (après 30 min), il doit s'authentifier.
 - E2: Il reste pour le rendez-vous que moins de 24 heures.
Le système affiche « la modification est impossible »
 - E3 : Le patient a dépassé le total des annulations acceptés :3 fois.
Le système affiche « annulation impossible »

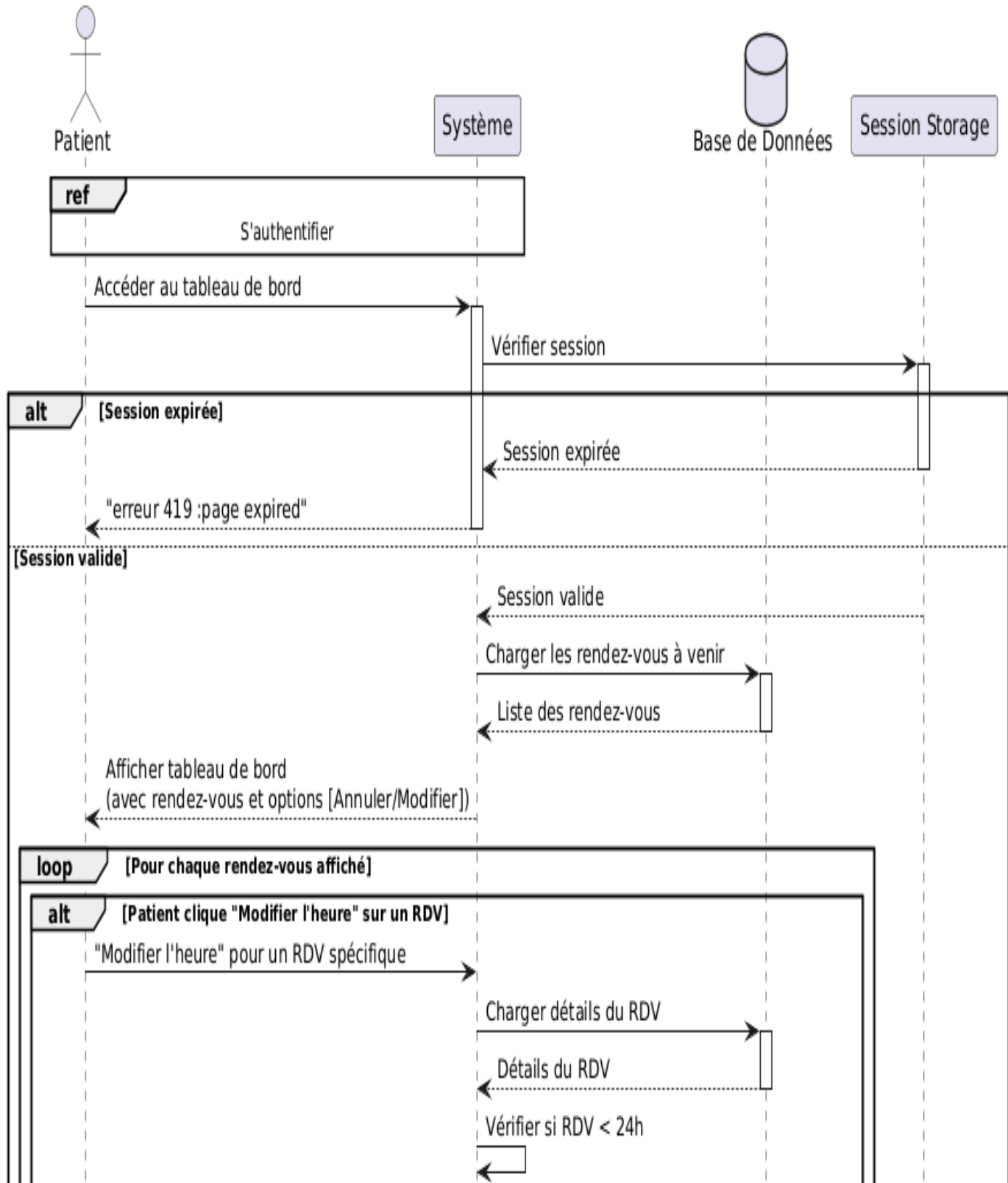


Figure 20:Diagramme de séquence de patient gérer le rendez-vous partie 1

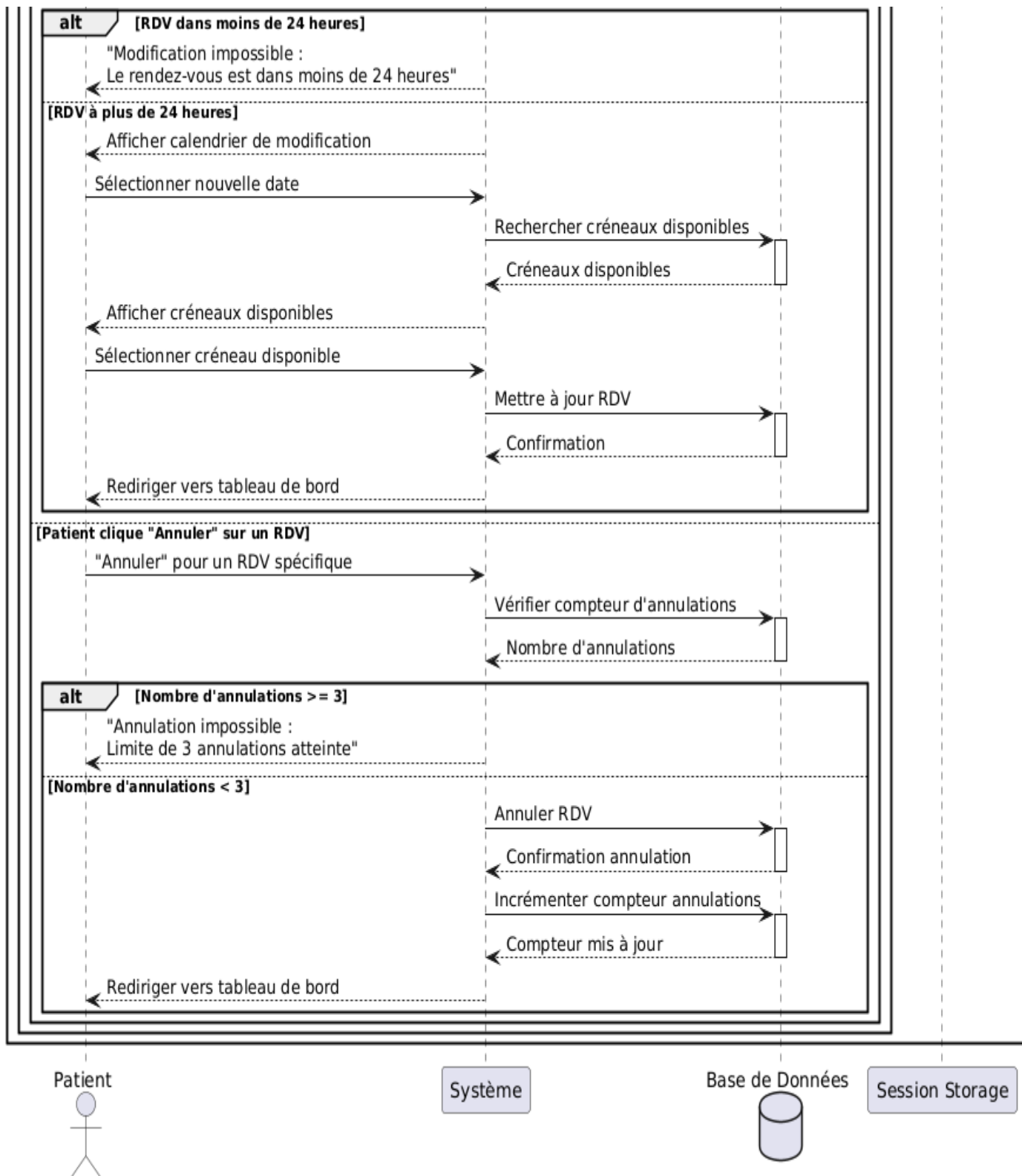


Figure 21: Diagramme de séquence de patient gérer le rendez-vous partie 2

B1. Description textuelle des cas d'utilisation de médecin

a) Description textuelle de gérer les dossiers médicaux

- **Titre** : Gérer les dossiers médicaux.
- **Résumé** : Ce cas d'utilisation permet au médecin/spécialiste de consulter, met à jour ou créer un dossier médical pour un patient.
- **Acteur principal** : Médecin/spécialiste.
- **Précondition** : Médecin/spécialiste dispose compte.
- **Scénario nominale** :
 1. Le médecin/spécialiste se connecte au système avec ses identifiants.
 2. Il clique sur «view profile» dans la liste des rendez-vous affiché dans le tableau de bord.
 3. Le système affiche le dossier médical.
 4. S'il veut ajouter quelque chose dans le dossier, il clique sur « add new record ».
 5. Il sélectionne le type.
 6. Le système affiche un formulaire.
 7. Le médecin/spécialiste remplit le formulaire et enregistrer.
 8. Le système vérifie le rôle d'utilisateur.
 9. Le système sauvegarde les changements dans la base de données.
 10. Le système affiche le dossier médical modifiée.
- **Enchaînements alternatifs**
 - ◆ A1: Le système rejette la soumission si la spécialiste essayait d'ajouter un enregistrement dans une section en dehors de sa spécialité.
 - ◆ L'enchaînement A1 démarre à l'étape 3 du scénario nominal
 1. Le spécialiste revient au sa section.
- **Enchaînements d'erreur**

E1: Session expiré.

Après la session termine (après 30 min), il doit s'authentifier.
- **Postcondition** : Dossier médical affiché.

b) Description textuelle de gérer son calendrier

- **Titre** : Gérer son calendrier.
- **Résumé** : Ce cas d'utilisation permet au médecin et spécialiste de gérer ses créneaux.
- **Acteur principal** : Médecin/spécialiste.
- **Précondition** : Médecin/spécialiste disposent compte valide.
- **Scénario nominale** :
 1. Le médecin/spécialiste se connectent au système avec ses identifiants.
 2. Le système affiche tableau de bord.
 3. Il visualise son calendrier pour la période concernée (journée, semaine, mois). Ceci comprend la visualisation des rendez-vous déjà programmés.
 4. Il peut sélectionner un créneau dans le calendrier et le mettre : disponible, réservé, bloqué.
 5. Ils cliquent sur « save ».
 6. Le système enregistre le statut.
- **Enchaînements alternatifs**:/
- **Enchaînements d'erreur**

E1: Session expiré.

Après la session termine (après 30 min), il doit s'authentifier.

Remarque : Les descriptions textuelles des processus de gestion des dossiers médicaux, du calendrier, des rendez-vous, ainsi que de la prise de rendez-vous chez un spécialiste sont disponibles en annexe.

b) Diagramme de séquence de gérer le dossier médical de ses patients

Cette fonction est cruciale car elle est limitée au médecin seulement, il peut créer son dossier, le mettre à jour est c'est montré dans le diagramme de séquence suivant :

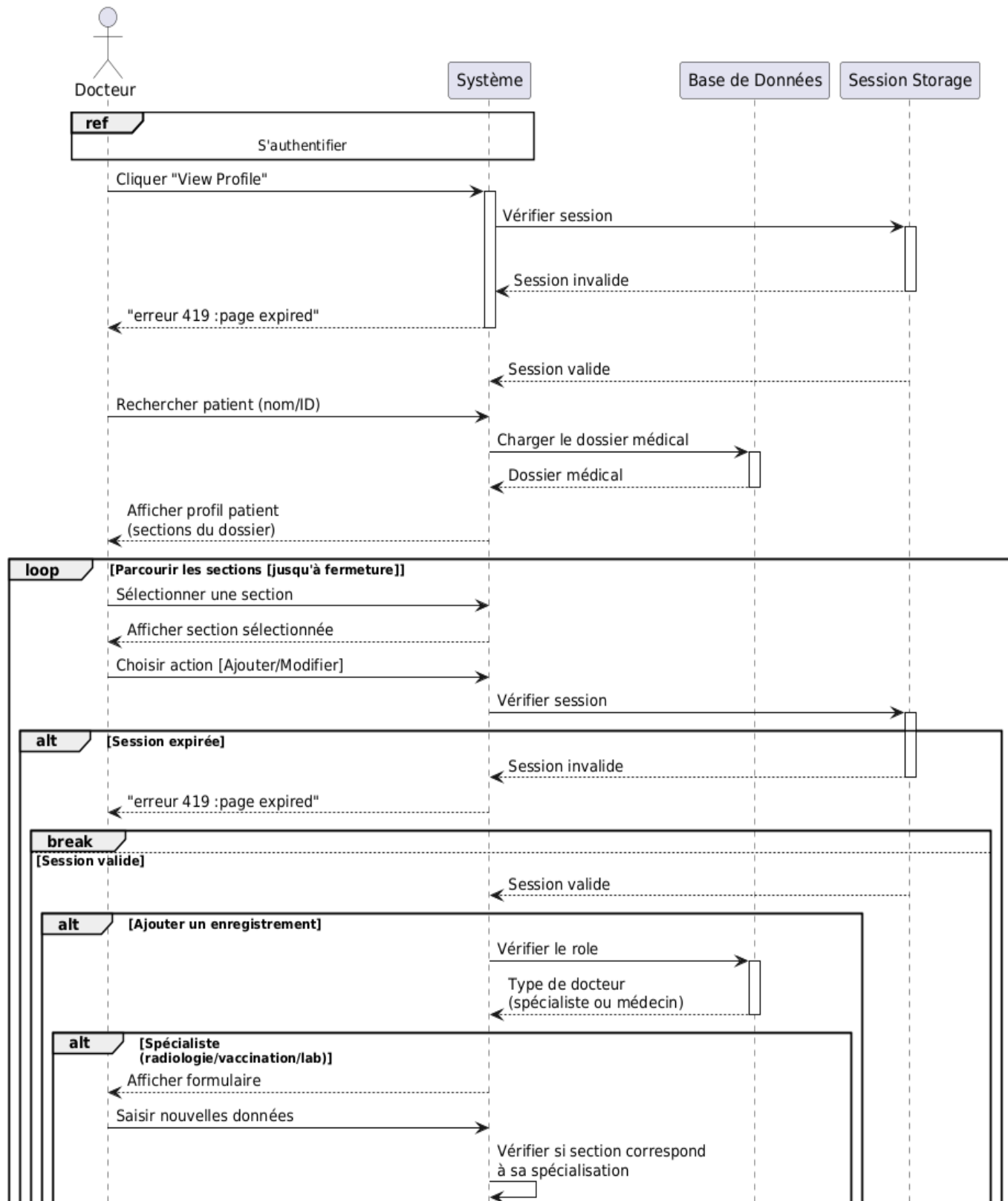


Figure 22: Diagramme de séquence de médecin pour gérer les dossiers médicaux partie 1

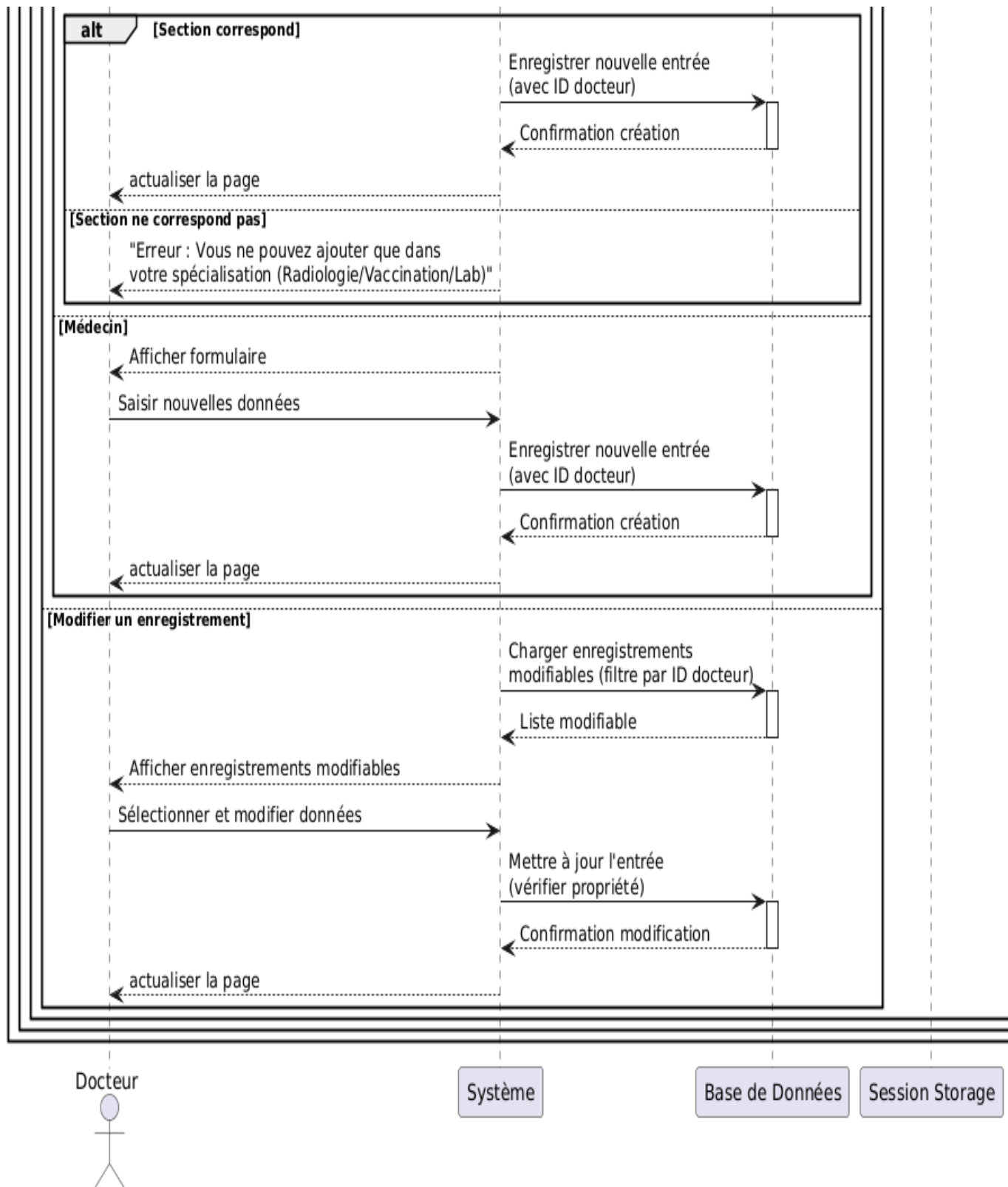


Figure 23: Diagramme de séquence de médecin pour gérer les dossiers médicaux partie 2

c) Diagramme de séquence de gérer son calendrier

Ce processus permet au médecin de mettre à jour facilement le statut d'un rendez-vous, ce qui est essentiel pour le suivi des patients, et la planification future.

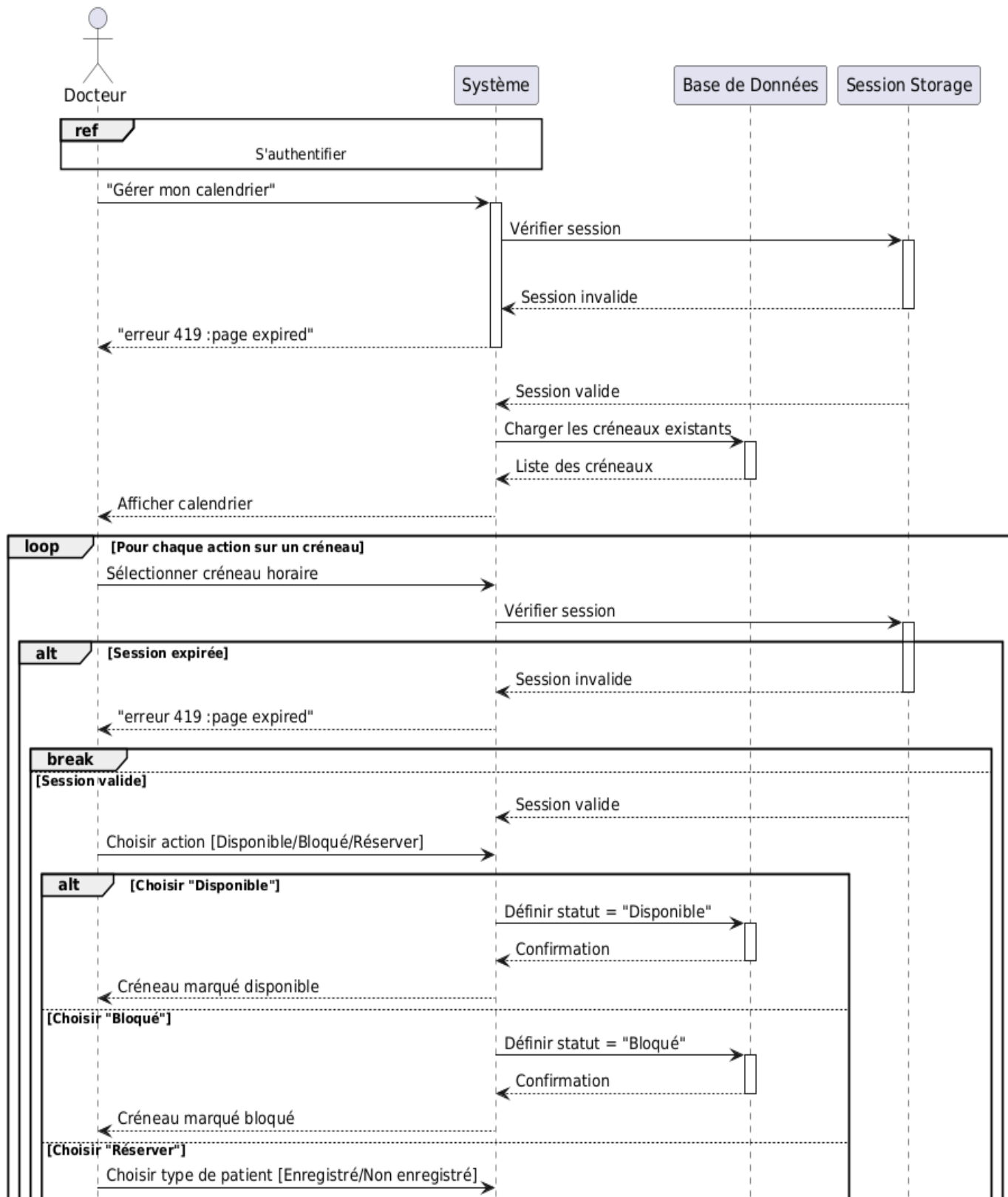


Figure 24: Diagramme de séquence de médecin pour gérer son calendrier partie 1

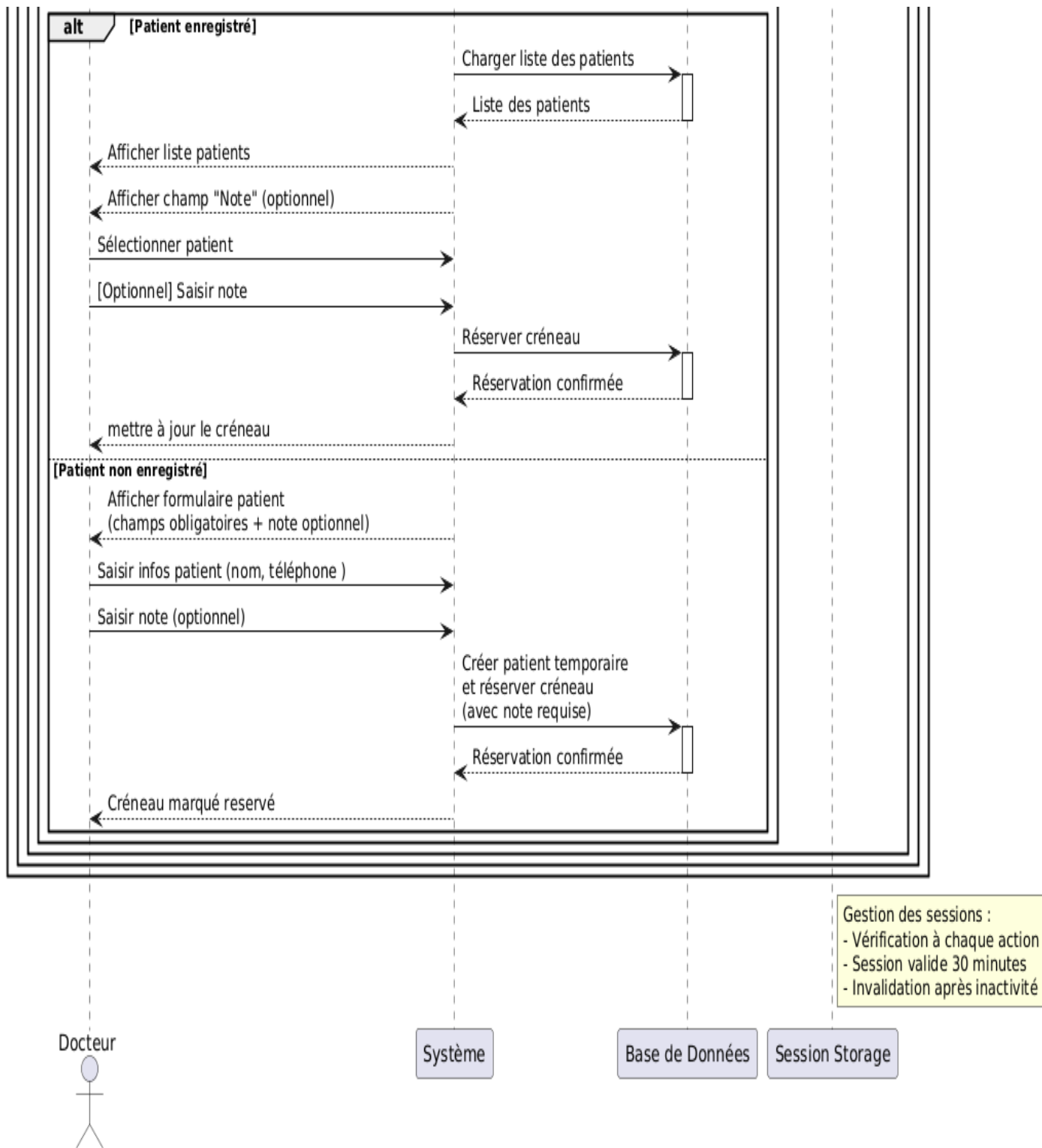


Figure 25: Diagramme de séquence de médecin pour gérer son calendrier partie 2

6. Diagramme de classe

Le diagramme de classe est un outil fondamental pour assurer une conception claire, cohérente et efficace d'un système logiciel orienté objet. Il constitue la pierre angulaire de la modélisation statique, permettant de structurer la réflexion, de communiquer efficacement et de faciliter la réalisation du logiciel.[40]

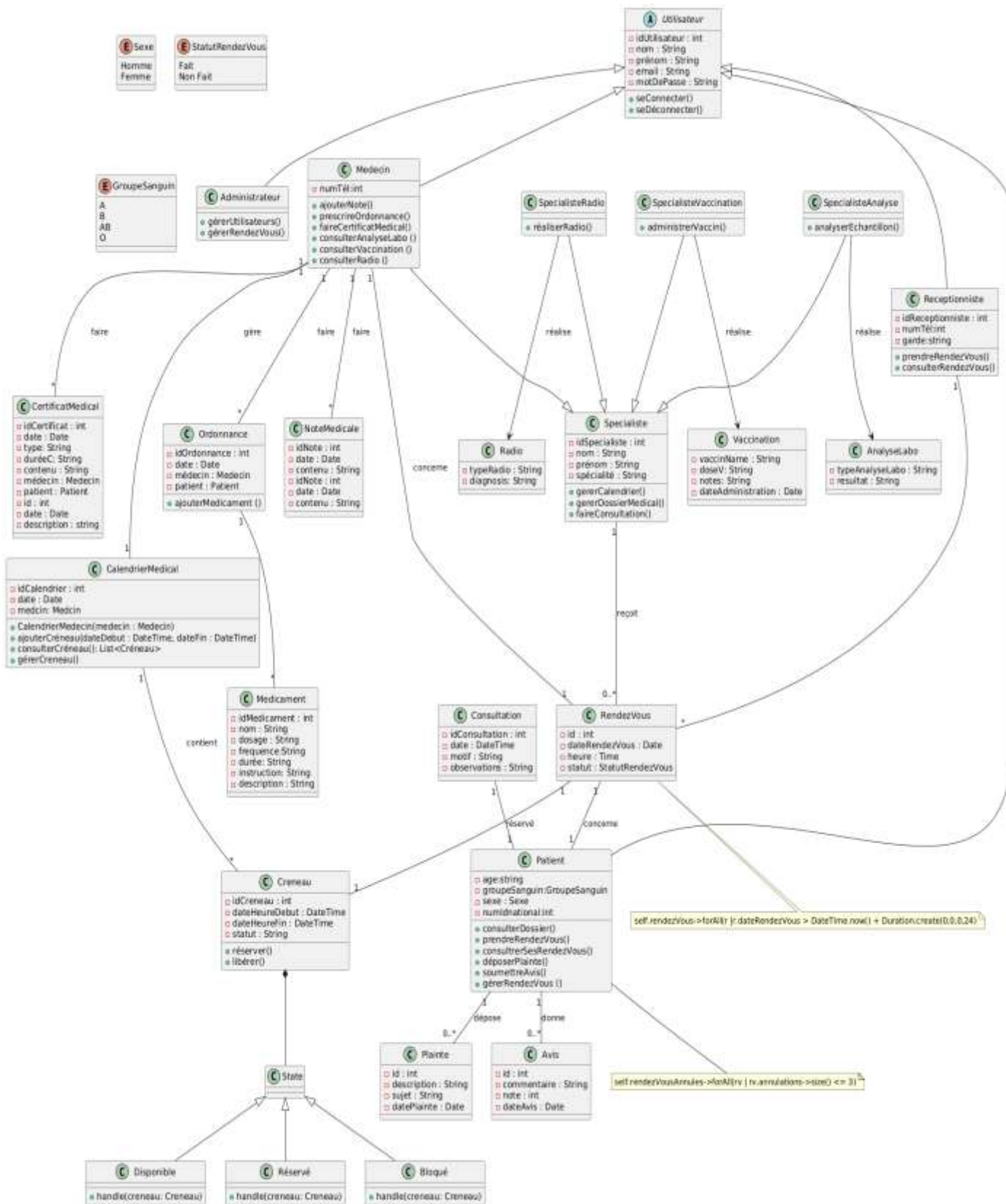


Figure 26: Diagramme de classe de système

4.5.2 Modèle de conception

1. Diagramme de classe de conception

On a suivi les stéréotypes de jacobson pour montrer l'architecture de notre application [41] :

- «boundary»: sont les classes qui servent à modéliser les interactions entre le système et ses acteurs.
- «control»: sont les classes utilisées pour représenter la coordination, l'enchaînement et le contrôle d'autres objets.
- «entity»: sont les classes qui servent à modéliser des informations durables et souvent persistantes.

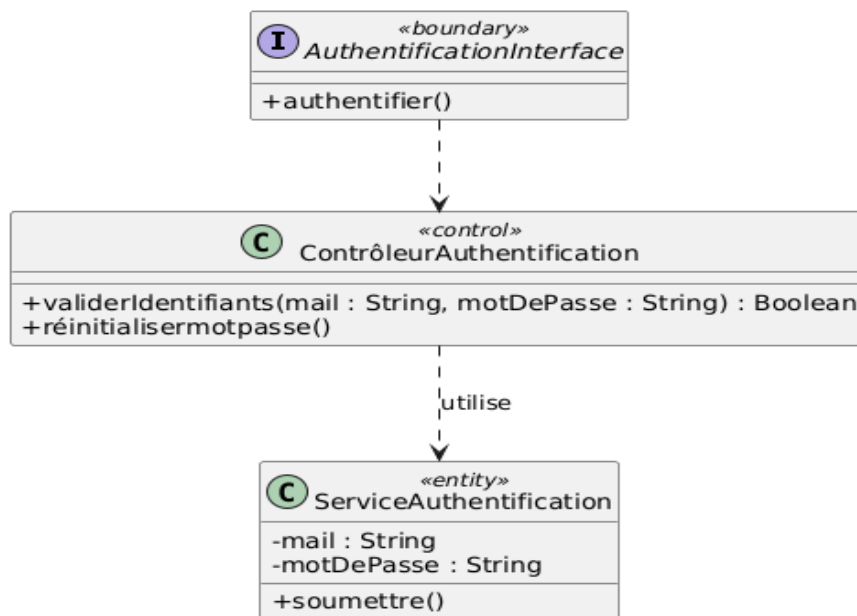


Figure 27: Diagramme de classe de conception pour l'authentification

4.5.3 Modèle d'implémentation

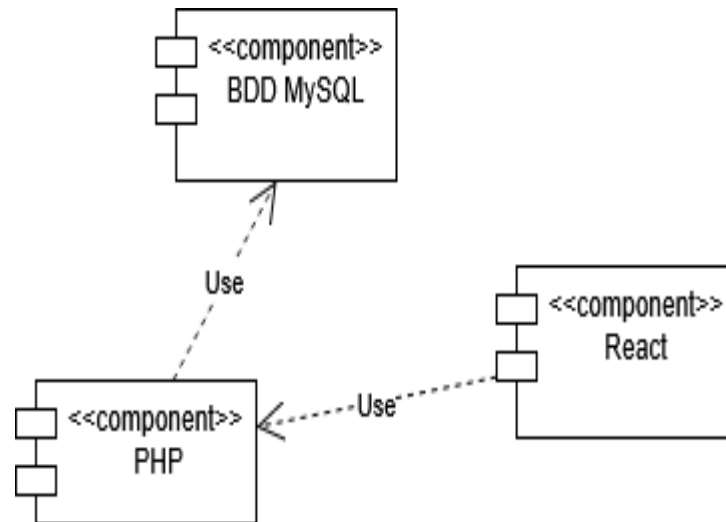


Figure 28: Diagramme de composant

4.5.4 Modèle de déploiement

1. Diagramme de déploiement

Le diagramme de déploiement est conçu pour illustrer la configuration physique du système, en montrant comment les éléments logiciels sont déployés sur le matériel. Son objectif principal est de représenter la topologie du système en mettant en évidence la distribution des composants et leur interaction dans l'environnement physique.[43]



Figure 29: Diagramme de déploiement

4.5.5 Modèle de test

1. Test d'inscription pour patient

Ce diagramme de séquence, le patient interagit avec le système en saisissant le nom, le groupe sanguin, l'e-mail, le numéro de téléphone et le mot de passe. Le système vérifie les données saisies, et dans ce cas spécifique, si l'e-mail est invalide, il envoie un message au patient indiquant que l'e-mail de logement est invalide (il existe déjà).

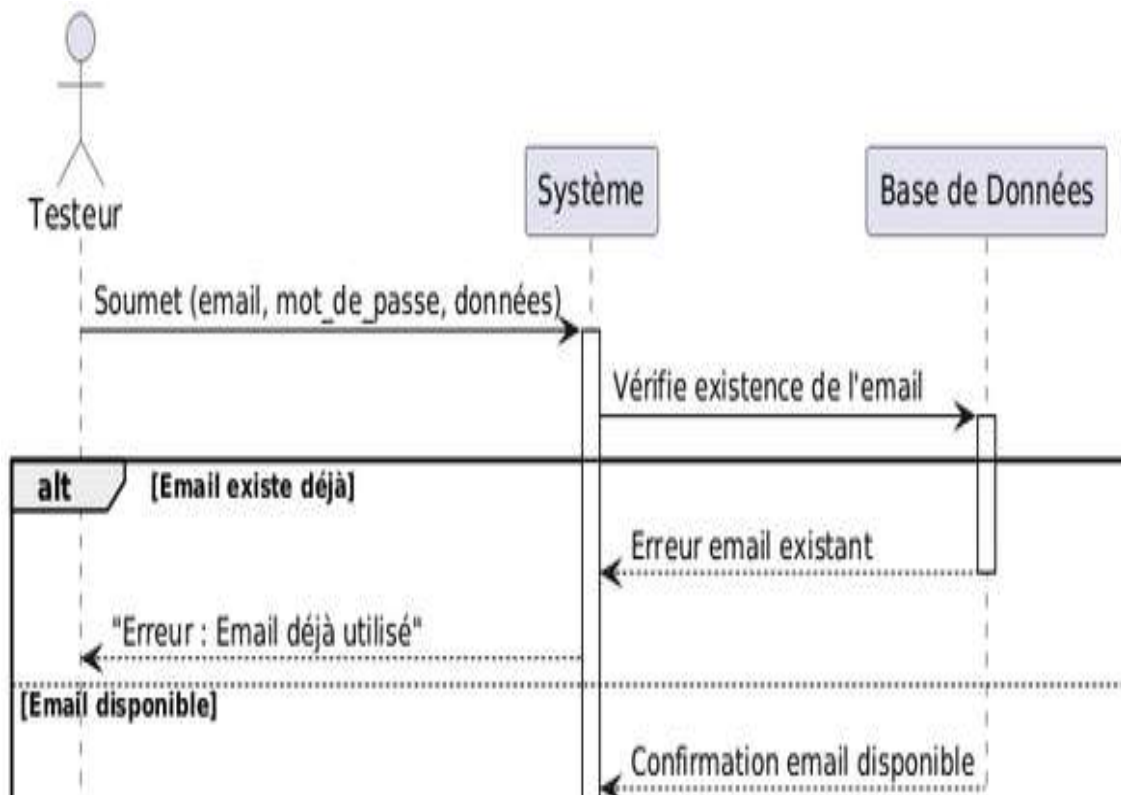


Figure 30: Diagramme de test d'inscription

2. Test d'authentification pour un utilisateur

Ce diagramme de séquence illustre l'étape de l'authentification quand un utilisateur veut accéder à l'application.

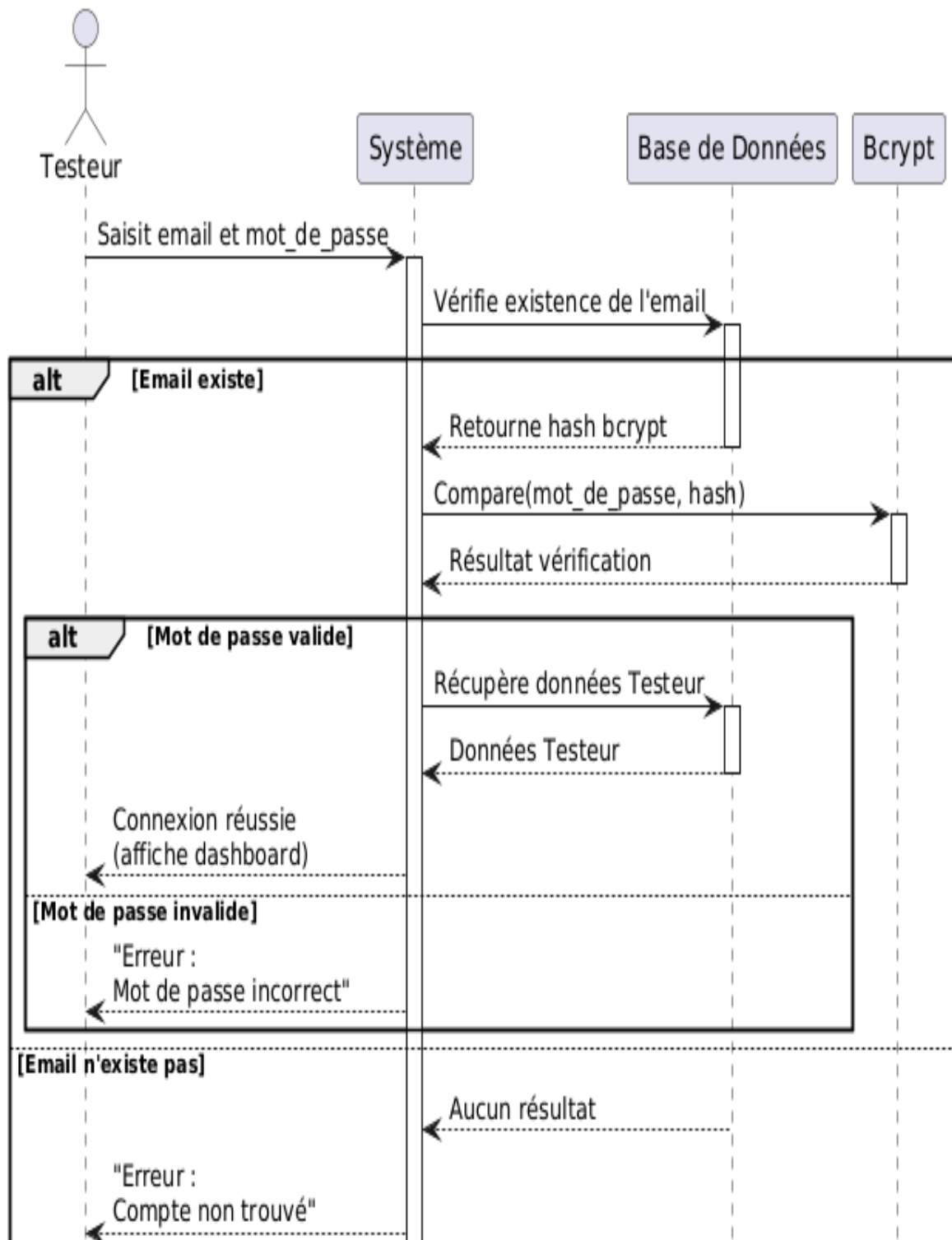


Figure 31: Diagramme Test d'authentification

Pour plus de détails sur les diagrammes et les descriptions textuel il faut voir l'annexe

4.6 Conclusion

Ce chapitre marque une étape essentielle dans la réalisation de notre projet, en nous concentrant sur l'analyse conceptuelle et la modélisation UML (Unified Modeling Language) en utilisant la méthode RUP (Rational Unified Process). Grâce à cette méthode reconnue par sa rigueur et son efficacité dans la gestion de projets de développement de logiciels, nous avons pu structurer notre réflexion et concevoir notre application de manière organisée. Les différentes étapes de l'analyse conceptuelle présentées dans ce chapitre, ainsi que les diagrammes UML correspondants, nous ont permis de visualiser clairement l'architecture de notre application.

Dans le dernier chapitre de l'implémentation, nous allons finaliser notre travail par concrétiser la conception en un système fonctionnel, et de préparer le système à des tests, incluant potentiellement des attaques pour assurer la robustesse et la sécurité de l'application.

Chapitre 5

Implémentation et Test

5.1 Introduction

La sécurité des données médicales est une priorité absolue, étant donné la nature sensible des informations traitées. Les risques liés à la confidentialité, à l'intégrité et à la disponibilité des données doivent être anticipés et atténués par la mise en œuvre de mécanismes de sécurité robustes. Cela inclut l'authentification et l'autorisation des utilisateurs, la protection contre les accès non autorisés, la sécurisation des communications, ainsi que la conformité aux normes et réglementations en vigueur.

Ce chapitre présente en détail le processus d'implémentation du site, en décrivant les choix technologiques et les méthodes adoptées. Il aborde également la stratégie de mise en œuvre des mesures de sécurité, suivie des tests effectués pour garantir leur efficacité. L'objectif est d'assurer un système fiable, sécurisé et conforme aux attentes des utilisateurs et aux exigences réglementaires.

5.2 Environnement logiciel

Pour réaliser un projet informatique de qualité tout d'abord il faut bien choisir les outils à utiliser, car on ne peut pas dire qu'un langage de programmation est plus fort qu'un autre, mais on peut seulement dire qu'un langage informatique est convenable avec telle ou telle tâche plus qu'un autre. Nous avons implémenté notre application web en utilisant :

- Front-end: react, CSS.
- Back-end : PHP sous le Framework Laravel.

5.2.1 Laravel

Laravel est un framework web PHP qui fournit un ensemble complet d'outils et de fonctionnalités pour la création d'applications web robustes, sécurisées et maintenables. Il propose une architecture modulaire, une syntaxe claire et concis.[44]

5.2.2 Github

GitHub est une plateforme Web de contrôle de version et de collaboration pour les développeurs de logiciels. Il permet aux développeurs de collaborer plus efficacement sur un projet en fournissant des outils pour gérer les modifications éventuellement conflictuelles de plusieurs développeurs.[46]

5.2.3 Vs code

Visual Studio Code est un éditeur de code source open source, multiplateforme, Il offre une intégration avancée avec divers langages de programmation, outils de débogage, gestionnaire de versions, et supporte le développement collaboratif.[47]

5.2.4 Phpmyadmin

PhpMyAdmin est un outil open source basé sur le web, écrit en PHP, destiné à gérer l'administration de MySQL et MariaDB via le Web.[48]

5.2.5 Xampp server

XAMPP est un package logiciel open source qui fournit un environnement de développement local intégré, comprenant un serveur web Apache, une base de données MySQL (ou MariaDB), un interpréteur PHP, et d'autres outils nécessaires pour le développement d'applications web. [49]

5.2.6 Apache

Un logiciel serveur HTTP open source, permettant de gérer les requêtes client et de servir des pages web dynamiques ou statiques.[50]

5.2.7 Git

Git est un système de gestion de versions distribué, conçu pour suivre l'historique des modifications apportées à des fichiers dans le cadre de projets de développement logiciel. Il permet à plusieurs développeurs de collaborer efficacement en gérant les différentes versions du code source, en facilitant le branchement, la fusion et la résolution des conflits. [52]

5.3 Le test des attaques

5.3.1 Injection

1. Injection XSS

Le test XSS vise à détecter si une application web est vulnérable à des attaques par injection de scripts malveillants. Lors de ce test, on a insérer du code JavaScript dans un champ de saisie de formulaire pour voir si ce code peut s'exécuter dans le navigateur d'un utilisateur, on peut voir ça dans cette figure:

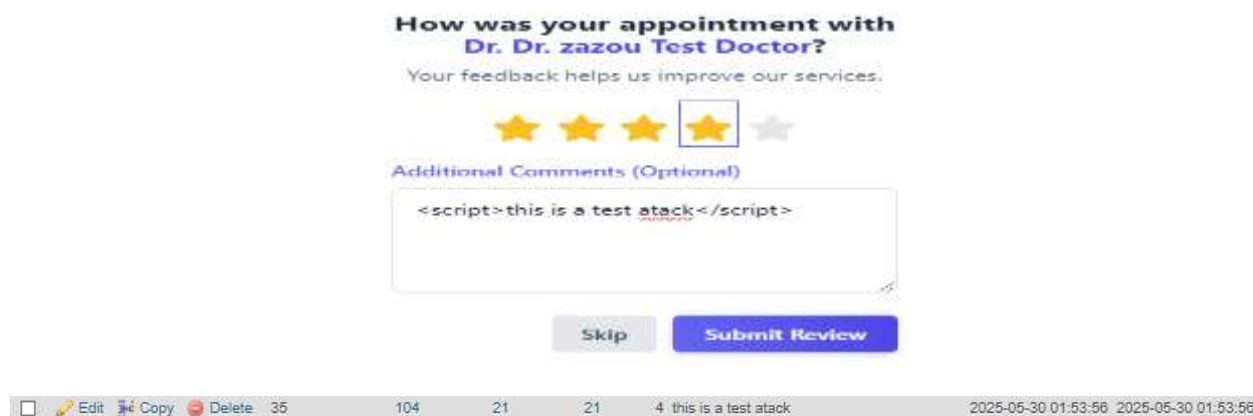


Figure 32: Une attaque XSS dans le formulaire de patient

Et grâce à notre « input sanitizer » nous avons pu assainir ce script et éliminer tout le danger qui l'accompagnait comme vous pouvez le voir sur l'image ci-dessous l'entrée a été nettoyée.

2. Injection SQL

Le test SQL injection vise à détecter si une application web est vulnérable à des attaques par injection des commandes SQL dans ce test on a inséré une injection SQL basique (Figure 30).



Figure 33: Une attaque SQL dans le formulaire de patient

Cette injection aurait été exécutée comme ceci dans le backend :

```
SELECT * FROM users WHERE email = '$email' AND password = '$password';
```

Cependant, cette tentative est immédiatement bloquée par notre validation des formats frontend et même si d'une manière ou d'une autre cela était exécuté, rien de tout cela ne se produirait car nous avons toujours le « input sanitizer » et les requêtes paramétrées qui viennent avec laravel.

5.4 Les mécanismes de sécurité

5.4.1 Authentification et Sécurité des Accès :

Pour garantir la sécurité des utilisateurs, nous avons mis en place un système d'authentification robuste.

Lors de l'inscription d'un nouvel utilisateur, nous accordons une grande importance à la protection des informations sensibles, en particulier le mot de passe nous n'en stockons jamais les mots de passe en clair dans la base de données ; nous utilisons l'algorithme bcrypt.

Bcrypt applique un hachage avec salage (salt) automatique ce que cela signifie, c'est que même lorsque deux utilisateurs ont le même mot de passe dans notre base de données, ils ne seront pas les mêmes et cette fonction est bonne car elle rend l'inversion de hachage très difficile.

```
$user = User::create([
    'name' => $request->name,
    'email' => $request->email,
    'national_identification_number' => $request->national_id,
    'password' => Hash::make($request->password),
    'role' => 'patient',
]);
```

Figure 34: Hachage sur le mot de passe de patient

Lors de la connexion, le mot de passe saisi par l'utilisateur n'est pas haché puis stocké. Le système d'authentification prend le mot de passe fourni par l'utilisateur, le hache, puis compare ce hachage nouvellement généré à celui enregistré pour cet utilisateur dans la base de données. Si les hachages correspondent, l'utilisateur est authentifié. Cette comparaison est généralement gérée par le système de protection d'authentification de Laravel à l'aide de méthodes telles que Hash::check().

```
$this->assertTrue(Hash::check('new-password', $user->refresh()->password));
```

Figure 35: Hachage check sur le mot de passe

Nous avons également mis en œuvre le rate limiter (limiteur de débit) dans Laravel est une fonctionnalité qui permet de limiter le nombre de requêtes qu'un utilisateur ou une adresse IP peut effectuer dans un certain laps de temps. Cela aide à prévenir les abus, les attaques par déni de service (DoS), ou simplement à gérer la charge du serveur.

Dans notre site : Nous avons limité le nombre de tentatives de l'authentification à 5 fois, le nombre des requêtes dans api à 60 par minutes et pour le reste des routes à 120 par minutes. (Voir la figure 33)

```

class ApiRateLimiter
{
    /**
     * Handle an incoming request.
     *
     * @param \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response) $next
     */
    public function handle(Request $request, Closure $next): Response
    {
        $key = $request->ip();

        // Different rate limits based on the endpoint
        $maxAttempts = match (true) {
            str_contains($request->path(), 'login') => 5, // 5 attempts
            str_contains($request->path(), 'api') => 60, // 60 requests per minute for API
            default => 120, // 120 requests per minute for other routes
        };

        $decaySeconds = 60; // Reset after 1 minute

        if (FacadesRateLimiter::tooManyAttempts($key, $maxAttempts)) {
            return response()->json([
                'error' => 'Too many requests',
                'retry_after' => FacadesRateLimiter::availableIn($key)
            ], 429);
        }
    }
}

```

Figure 36: rate limiter

5.4.2 Content security policy (CSP)

Il permet de définir les sources autorisées pour le chargement de ressources (scripts, styles, images, etc.) sur une page web. En limitant ces sources, CSP empêche l'exécution de scripts malveillants injectés par des attaquants.

CSP est implémenté en ajoutant un en-tête HTTP Content-Security-Policy à la réponse du serveur. Dans cet en-tête, on définit les règles qui contrôlent quelles ressources sont autorisées à être chargées. Cette figure montre ça :

```

<?php
namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;

class SecurityHeaders
{
    public function handle(Request $request, Closure $next)
    {
        $response = $next($request);

        $response->headers->set('X-Frame-Options', 'DENY');
        $response->headers->set('X-Content-Type-Options', 'nosniff');
        $response->headers->set('Referrer-Policy', 'strict-origin-when-cross-origin');
        return $response;
    }
}

```

Figure 37: Code PHP de security headers

1. X-Frame-Options

Cette protection vise à prévenir des attaques de type clickjacking, où un attaquant tente de tromper un utilisateur pour qu'il interagisse avec une page en la dissimulant dans un cadre, souvent pour voler des informations ou effectuer des actions malveillantes.

2. X-Content-Type-Options

'X-Content-Type-Options' => 'nosniff' dans les en-têtes de sécurité (security headers figure 40) est un mécanisme pour prévenir les failles de type Cross-Site Scripting (XSS) en empêchant le navigateur d'interpréter le contenu d'une page comme un type MIME différent de celui spécifié.

L'en-tête **X-Content-Type-Options** accepte une seule valeur : **nosniff**. Quand il est présent avec la valeur nosniff, le navigateur est forcé de respecter le type MIME indiqué dans le header Content-Type de la réponse HTTP. Il n'essayera pas de deviner ou de réinterpréter le type MIME du contenu reçu

3. Strict-Transport-Security (HSTS)

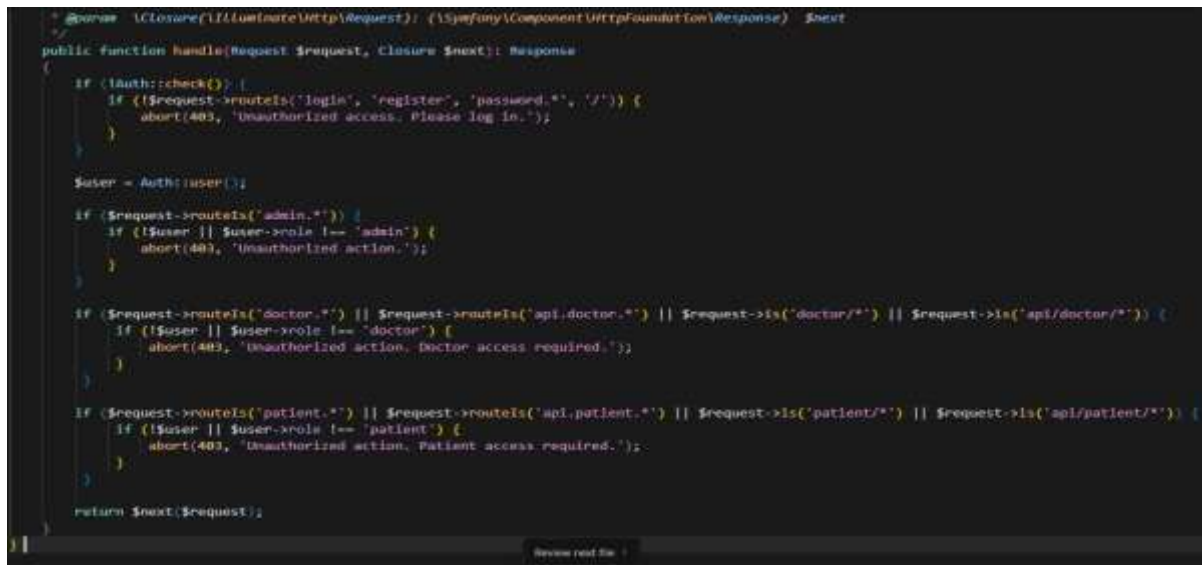
Strict-Transport-Security (souvent abrégé en HSTS, pour HTTP Strict Transport Security) est un en-tête HTTP de sécurité qui indique aux navigateurs que le site Web doit être accessible uniquement via une connexion sécurisée HTTPS. Il aide à prévenir certains types d'attaques, notamment les attaques de type "Man-in-the-Middle" (MITM) et les tentatives de rétrogradation vers HTTP non sécurisé.

5.4.3 Middleware

Le Middleware est une couche intermédiaire qui se place entre la requête HTTP entrante et la réponse sortante dans une application Laravel. Elle permet d'intercepter, de filtrer ou de modifier les requêtes et réponses avant qu'elles n'atteignent le contrôleur ou ne soient renvoyées au client. Elle sert à :

- Authentification : Vérifier si un utilisateur est connecté avant d'accéder à certaines routes.
- Autorisation : Contrôler l'accès à des ressources en fonction des rôles ou permissions.
- Filtrage des requêtes : Par exemple, vérifier si la requête provient d'une certaine origine ou si elle contient certains paramètres.

- Modification des réponses : Ajouter des en-têtes ou autres modifications avant d'envoyer la réponse au client.



```

    @var Closure($request: \Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response) $next
    */
    public function handle($request, Closure $next): Response
    {
        if (!auth::check()) {
            if ($request->routeIs('login', 'register', 'password.*', '/')) {
                abort(403, 'Unauthorized access. Please log in.');
            }
        }

        $user = Auth::user();

        if ($request->routeIs('admin.*')) {
            if (!$user || $user->role !== 'admin') {
                abort(403, 'Unauthorized action.');
            }
        }

        if ($request->routeIs('doctor.*') || $request->routeIs('api.doctor.*') || $request->is('doctor/*') || $request->is('api/doctor/*')) {
            if (!$user || $user->role !== 'doctor') {
                abort(403, 'Unauthorized action. Doctor access required.');
            }
        }

        if ($request->routeIs('patient.*') || $request->routeIs('api.patient.*') || $request->is('patient/*') || $request->is('api/patient/*')) {
            if (!$user || $user->role !== 'patient') {
                abort(403, 'Unauthorized action. Patient access required.');
            }
        }

        return $next($request);
    }
}

```

Figure 38: Middleware dans code php

5.4.4 L'input sanitizer

L'input sanitizer (ou "nettoyeur d'entrée" en français) est une technique ou un outil utilisé pour nettoyer, valider et sécuriser les données entrantes dans une application web ou logicielle. Son objectif principal est de prévenir des vulnérabilités telles que les injections SQL, les scripts inter-sites (XSS), ou d'autres attaques basées sur des entrées malveillantes. Elle est essentielle pour :

- Empêcher l'exécution de code malveillant : En éliminant ou en échappant les caractères spéciaux, on évite que des scripts malicieux soient injectés dans l'application.
- Validation des données : Vérifier que l'entrée respecte le format attendu (nombres, adresses email, etc.).
- Prévenir les failles de sécurité : Par exemple, attaques XSS, injection SQL, etc.

```

public function handle(Request $request, Closure $next)
{
    $input = $request->all();
    array_walk_recursive($input, function (&$value) {
        if (is_string($value)) {
            $value = str_replace(['<script', '</script>', 'javascript:', 'onerror', 'onload'], '', $value);
        }
    });
}

```

Figure 39: code php Input sanitizer

5.4.5 Verify Csrf Token

Laravel intègre une protection CSRF par défaut via le middleware VerifyCsrfToken responsable de la protection contre les attaques CSRF (Cross-Site Request Forgery). Il vérifie que chaque requête susceptible de modifier des données (POST, PUT, PATCH, DELETE) contient un jeton CSRF valide, assurant que la requête provient bien de votre application et non d'un site malveillant.

```

<?php
namespace App\Http\Middleware;
use Illuminate\Foundation\Http\Middleware\VerifyCsrfToken as Middleware;

class VerifyCsrfToken extends Middleware
{
    protected $except = [
        'admin/dashboard',
        'admin/doctors',
        'admin/patients',
        'admin/appointments',
        'admin/prescriptions',
        'admin/receptionists',
        'admin/api/stats',
        'admin/api/doctors',
        'admin/api/patients',
        'admin/api/appointments',
        'admin/api/prescriptions',
        'admin/api/receptionists',
        'admin/api/security-logs',
        'doctor/patients',
        'doctor/prescriptions',
        'doctor/appointments',
        'doctor/profile',
        'doctor/profile/update',
        'doctor/schedule/update',
        'doctor/availability/block',
        'doctor/availability/reserve',
        'doctor/availability',
        'doctor/patients/*/medical-records/store',
        'doctor/patients/*/medical-records/update',
        'profile/update',
    ];
}

```

Figure 40: Verify Csrf Token

5.5 Les interfaces de site web

5.5.1 Interface de tableau de bord de patient

Fournir une vue centralisée et accessible des informations essentielles du patient :

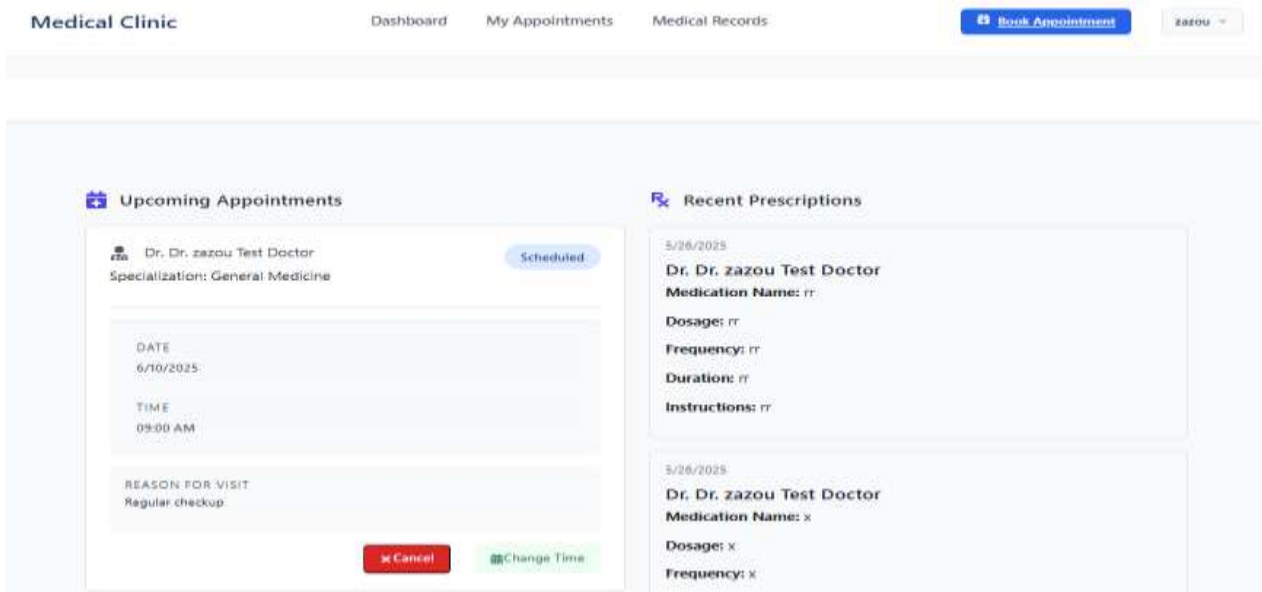


Figure 41: Interface de tableau de bord de patient

5.5.2 Interface de dossier médical

Medical Records

Patient Information

Full Name	Age	Gender
zazou	25 years	male
Blood Type	Allergies	Chronic Diseases
A+	None reported	None reported

98

Total Visits

5/30/2025

Last Visit

None

Next Appointment

Radiology

Prescriptions

Certificates

Lab Results

Vaccinations

Doctor's Notes

Medical Certificates

sick

4/17/2025

Doctor: Dr. zazou Test Doctor

Type: sick

Duration: 5 days

Details: tech

Fitness Certificate

4/17/2025

Doctor: Dr. zazou Test Doctor

Type: Fitness Certificate

Duration: 6 months

Details: Patient is fit to participate in regular physical activities

Figure 42: Interface de dossier médical

Remarque : Voir l’annexe pour une présentation plus complète des interfaces.

5.6 Conclusion

Ce dernier chapitre a permis d’évaluer la résistance du site web face à diverses attaques en simulant des tentatives d’intrusion et en analysant les mécanismes de défense. Les tests effectués ont mis en évidence plusieurs vulnérabilités, soulignant l’importance de renforcer la sécurité par des mesures adaptées telles que la validation stricte des entrées, la mise en place de contrôles d’accès, et l’utilisation de mécanismes de détection d’intrusions. L’implémentation de ces mécanismes, combinée à une compréhension approfondie des techniques d’attaque, contribue à améliorer la résilience globale du site. Nous avons veillé à proposer des interfaces conçues pour offrir une expérience utilisateur intuitive, développées à l’aide d’outils et d’environnements modernes.

Conclusion générale

Ce mémoire a présenté une démarche complète de conception et d'implémentation d'un site web sécurisé dédié à la gestion des dossiers médicaux, dans le but d'améliorer la qualité, la sécurité et l'efficacité des services de santé. À travers cette étude, nous avons abordé les différentes étapes du projet, de l'analyse des besoins, en passant par la conception et la modélisation, jusqu'à la mise en œuvre et la réalisation des interfaces.

L'un des défis majeurs rencontrés concernait la sécurisation des données sensibles, conformément aux réglementations en vigueur telles que le RGPD, tout en assurant une expérience utilisateur fluide et intuitive. La mise en place de mécanismes de sécurité avancés, tels que l'authentification forte, le chiffrement des données, et la gestion rigoureuse des droits d'accès, a permis de garantir la confidentialité et l'intégrité des informations médicales. Par ailleurs, le développement d'une interface conviviale a facilité l'adoption du système par les usagers, qu'ils soient patients ou professionnels de santé.

Les résultats obtenus lors des phases de validation ont montré que le site web répondait efficacement aux exigences initiales, en proposant un site web fiable, sécurisée et accessible. La prise en charge des rendez-vous en ligne, la consultation sécurisée des dossiers médicaux, e ont permis d'optimiser la gestion administrative et médicale, tout en renforçant la relation de confiance entre les patients et les praticiens.

Au-delà des aspects techniques, ce projet soulève des questions importantes sur la transformation digitale du secteur de la santé. La digitalisation des dossiers médicaux offre de nombreuses opportunités pour améliorer la qualité des soins, faciliter la gestion des flux d'informations, et favoriser l'accès aux services de santé à distance. Cependant, elle impose également une vigilance accrue quant à la protection des données et à la conformité légale.

En conclusion, ce mémoire a permis de démontrer que la conception d'un site web sécurisé pour la gestion des dossiers médicaux et la prise de rendez-vous est à la fois réalisable et bénéfique. Elle constitue une étape essentielle vers une meilleure organisation des services de santé, tout en respectant les exigences de sécurité et de confidentialité. La poursuite de cette démarche

contribuera à faire évoluer les pratiques médicales vers une approche plus numérique, plus sûre et plus centrée sur le patient.

Bibliographie

- [1] Berners-Lee, T., & Fischetti, M. (2001), Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web, HarperSanFrancisco.
- [2] Tanenbaum, A. S., & Steen, M. Van. (2007). Distributed Systems: Principles and Paradigms (2nd ed.). Pearson Education.
- [3] Rescorla, E. (2000). HTTP Over TLS. RFC 2818. Internet Engineering Task Force (IETF).
- [4] Sebesta, R. W. (2012). Programming the World Wide Web (8th ed.). Pearson Education.
- [5] Roger, F. H. (1982). Le résumé du dossier médical : indicateur informatisé de performance et de qualité des soins. Thèse d'agrégation de l'enseignement supérieur, Faculté de médecine, Université catholique de Louvain.
- [6] Zarour, K. (2012). "L'interopérabilité des systèmes d'information médicaux : une approche basée agent". Thèse de Doctorat, Université Mentouri de Constantine
- [7] Honnorat C., 'Apprentissage de l'exercice médical : Le Dossier Médical'. Université de Rennes Faculté de Médecine. 14 novembre 2005.
- [8] Falcoff H., 'Dossier médical en médecine générale'. La revue du praticien, médecine générale, n° 404,1997, p 71-8
- [9] Moutel G., 'Évolution du dossier médical, nouveaux enjeux de la relation médecins-soignants patients': approche historique, médicale, médicolégale et éthique. Faculté de médecine, Université Paris 5, 2004.
- [10] Venot A., B. A., Quantin C. (2013). " Informatique médicale, e-Santé : Fondements et applications ". Springer-Verlag France.
- [11] Debiane, N., & Zegmali, F. Développement d'un modèle pour le contrôle d'accès au dossier médical personnel (Etude de cas : CHU Algérien) (Mémoire de Master, Université Abou Bakr Belkaid Tlemcen).
- [12] A.Atmane,«Conception et réalisation d'une application web pour la gestion des archives médicales Cas d'étude :CHU de Béjaia».
- [13] Guendoul, S., & Manane, S. (2017). Le Système d'Information Hospitalier (SIH) et le Dossier Médical Informatisé (DMI) Cas du CHU de TIZI-OUZOU (Thèse de Doctorat,Université Mouloud Mammeri).

- [14] CERISTNEWS Bulletin d'information trimestriel (Huitième numéro Mars 2012) ISSN-2170-0656, pp.11-14
- [15] Solms, R. von, & Niekerk, J. van. (2013). From information security to cyber security. *Computers & Security*, 38, 97–102.
- [16] Stallings, W. (2017). *Network Security Essentials: Applications and Standards* (6e éd.). Pearson Education
- [17] Sandhu, R., Coyne, E.J., Feinstein, H.L., & Youman, C.E. (1996). Role-Based Access Control Models. *IEEE Computer*, 29(2), 38–47.
- [18] F. Fluckiger, "Understanding Networked Multimedia: Applications and Technology", Prentice Hall, 1995.
- [19] Zaghdoudi, T. (2008). Intervention de l'État dans le domaine sanitaire (l'hôpital public) et le rôle du dossier médical dans la gestion des malades. Mémoire de Master, Université de Sfax.
- [20] qeel, M. H. E., et al. (2021). A review of the challenges of electronic medical records adoption in Saudi Arabia. *Journal of International Crisis and Risk Communication Research*, 4(2), 1–17.
- [21] Wang, S. J., & Middleton, B. (2009). Barriers to the acceptance of electronic medical records by physicians: From systematic review to taxonomy and interventions. *Journal of the American Medical Informatics Association*.
- [22] Jindal, S. K., & Raziuddin, F. (2018). Electronic medical record use and perceived medical error reduction. *International Journal of Quality and Service Sciences*.
- [23] Buntin, M. B., Burke, M. F., Hoaglin, M. C., & Blumenthal, D. (2011). The benefits of health information technology: A review of the recent literature shows predominantly positive results.
- [24] OWASP (2021). OWASP Top Ten Web Application Security Risks.
- [25] "Web application security testing using Burp Suite" par J. Smith et al. (2020).
- [26] Zero Trust security framework: A systematic review" par J. Rose et al. (2020)

- [27] "Automated security testing with OWASP ZAP" par P. Williams (2019)
- [28] "Static analysis tools for secure software development" par M. J. Howard (2020)
- [29] J. Liu, "Arachni: A Modular Web Application Security Scanner," Proceedings of the 11th International Conference on Web Security, 2017
- [30] L. Garcia et al., "Wapiti: An Open-Source Web Vulnerability Scanner," Journal of Cybersecurity Tools and Techniques, vol. 2, no. 1, 2018
- [31] K. Patel et al., "A Comparative Evaluation of Vulnerability Scanners: Nessus and OpenVAS," International Journal of Cybersecurity, vol. 9, no. 4, 2020.
- [32] Object Management Group (OMG). (2017). UML: Infrastructure for Object-Oriented Software Development. Version 2.5.1.
- [33] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., & Lorensen, W. (1991). Object-Oriented Modeling and Design. Prentice Hall.
- [34] Kruchten, P. (2000). The Rational Unified Process: An Introduction. IEEE Software, 17(7), 18-23.
- [35] Fowler, M. (2004). The Rational Unified Process: An Introduction. Addison-Wesley.
- [36] Cockburn, A. (2000). Writing Effective Use Cases. Addison-Wesley.
- [37] Jurjens, J.: UMLsec: Extending UML for Secure Systems Development. In: Proceedings of the 5th International Conference on The Unified Modeling Language, LNCS. Springer-Verlag, 2002.
- [38] F. Bobillier et al., "Ingénierie des systèmes logiciels," éditions ISTE, 2008.
- [39] R.pascale ,UML 2 par la pratique études des cas , 5eme édition , aout 2006.
- [40] Philippe Charland,UML 2 - Notions fondamentales et pratiques,2e édition,2010.
- [41] Pascal roques ,UML 2 pour la pratique,6e édition,2008.
- [42] Philippe Kruchten, Introduction au Rational Unified Process, Editions Eyrolles, 1999.
- [43] Booch, Rumbaugh et Jacobson ,Object-Oriented Modeling and Design, 1991.

- [44] «<https://laravel.com/>».
- [45] <https://logowik.com/laravel-logo-vector-svg-pdf-ai-eps-cdr-free-download-1-12077.html>
- [46] Ben Lutkevich, « GitHub », De Technical Features Writer [en ligne], mis en ligne le février2023, consulté le 14juin2023.
- [47] B. Smith, Journal of Software Engineering, 2021.
- [48] J. Martin, International Journal of Database Management, 2018.
- [49] García, M. (2018). Introduction à l'environnement de développement local pour applications web. Journal of Web Development, 12(3), 45-52.
- [50] Baron, R., & Lee, S. (2014). Apache HTTP Server in practice. Journal of Network and Computer Applications, 37, 89-97.
- [51] Elmasri, R., & Navathe, S. (2015). Fundamentals of Database Systems. Pearson.
- [52] Chacon, S., & Straub, B. (2014). Pro Git (2nd Edition). Apress.
- [53] BOUSSIF Zineb, La Modélisation des Processus Métiers dans l'Entreprise avec BPMN 2. En vue de l'obtention du diplôme de Master en Informatique OPTION : Informatique Industrielle,018/06/2022.
- [54] Présentation BPMN disponible sur : <https://www.trisotech.com/articles/bpmn-introduction-history> consulte en mars 2023.

Annexe A

1. Processus métier

a) Processus public

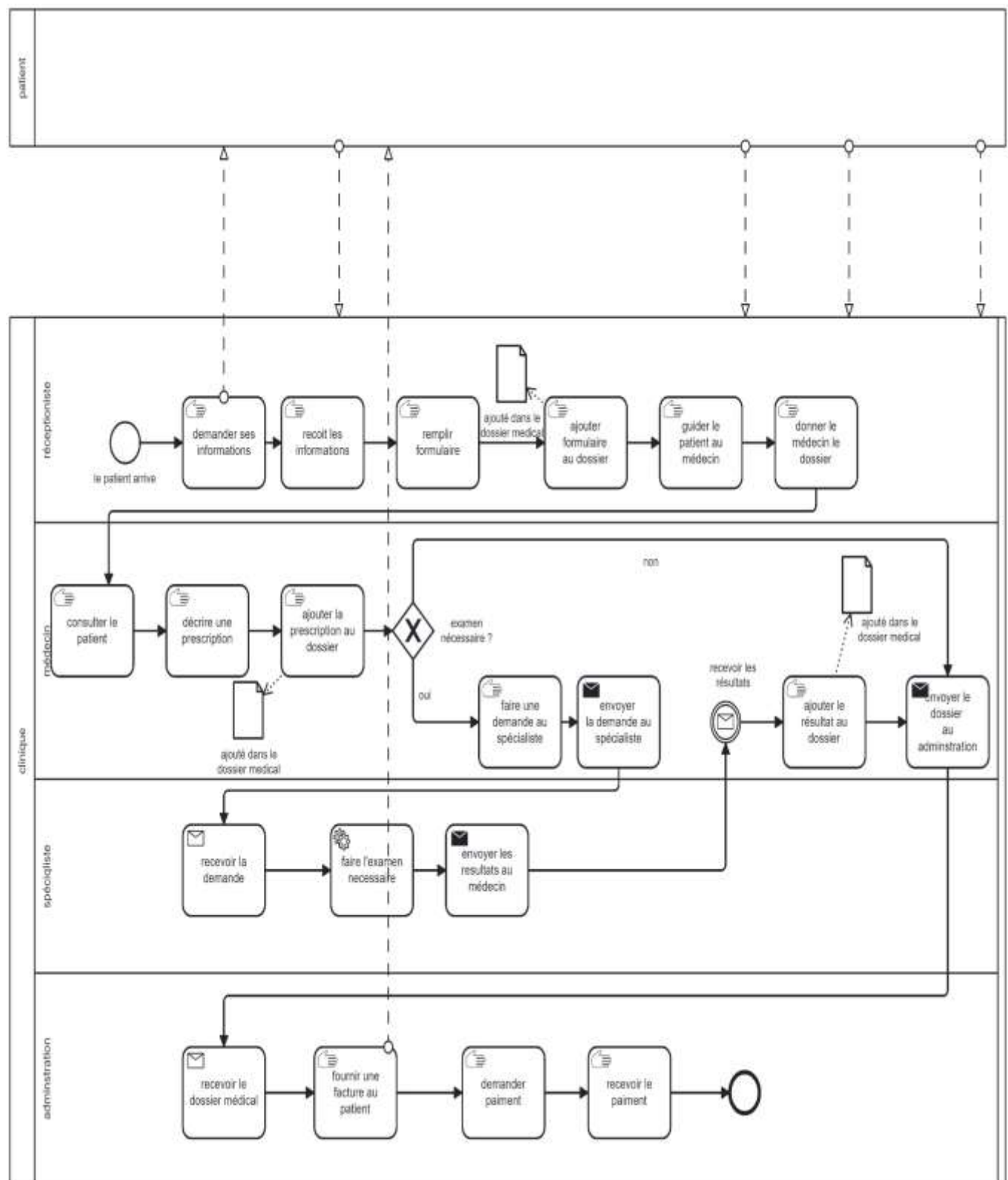


Figure 43: Processus métier public de gestion de dossier médical

b) Diagram de vue de conversation

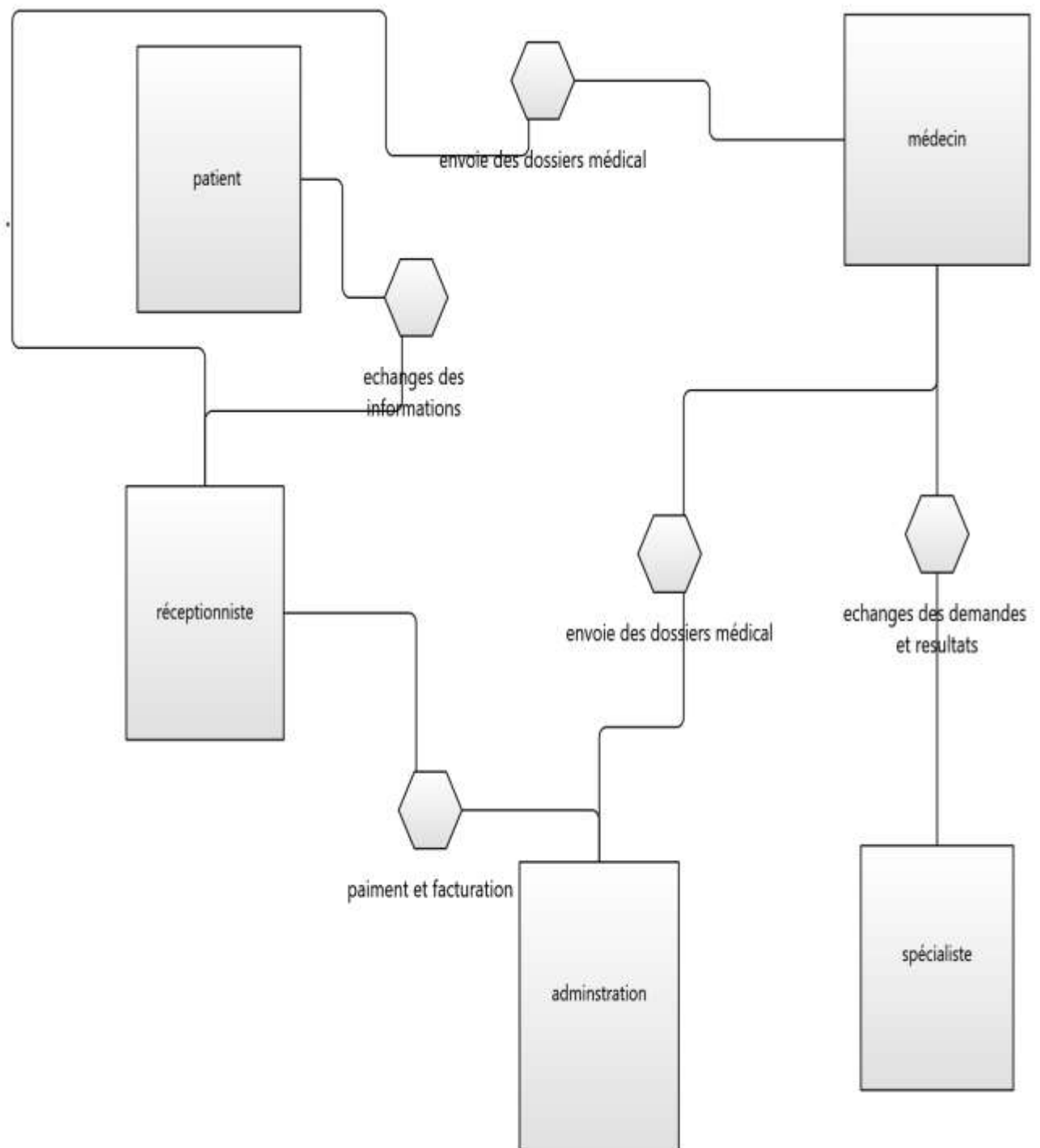


Figure 44: processus de vue en conversation

2. Description textuelle des cas d'utilisation de médecin/spécialiste

c) Description textuelle de gérer les dossiers médicaux

- **Titre:** Gérer les dossiers médicaux.
- **Résumé:** Ce cas d'utilisation permet au médecin/spécialiste de consulter, met à jour ou créer un dossier médical pour un patient.
- **Acteur principal:** Médecin/spécialiste.
- **Précondition:** Médecin/spécialiste dispose compte.
- **Scénario nominale:**
 11. Le médecin/spécialiste se connecte au système avec ses identifiants.
 12. Il clique sur «view profile» dans la liste des rendez-vous affiché dans le tableau de bord.
 13. Le système affiche le dossier médical.
 14. S'il veut ajouter quelque chose dans le dossier, il clique sur «add new record».
 15. Il sélectionne le type.
 16. Le système affiche un formulaire.
 17. Le médecin/spécialiste remplit le formulaire et enregistrer.
 18. Le système vérifie le rôle d'utilisateur.
 19. Le système sauvegarde les changements dans la base de données.
 20. Le système affiche le dossier médical modifiée.
- **Enchaînements alternatifs**
 - ◆ A1: Le système rejette la soumission si la spécialiste essayait d'ajouter un enregistrement dans une section en dehors de sa spécialité .
 - ◆ L'enchaînement A1 démarre à l'étape 3 du scénario nominal
 2. Le spécialiste revient au sa section.
- **Enchaînements d'erreur**

E1: Session expiré .

Après la session termine (après 30 min) , il doit s'authentifier.

- **Postcondition:** Dossier médical affiché.

d) Description textuelle de gérer son calendrier

- **Titre:** Gérer son calendrier.
- **Résumé:** Ce cas d'utilisation permet au médecin et spécialiste de gérer ses créneaux .
- **Acteur principal:** Médecin/spécialiste.
- **Précondition:** Médecin/spécialiste disposent compte valide.
- **Scénario nominale:**
 7. Le médecin/spécialiste se connectent au système avec ses identifiants.
 8. Le système affiche tableau de bord.
 9. Il visualisent son calendrier pour la période concernée (journée, semaine, mois). Ceci comprend la visualisation des rendez-vous déjà programmés.
 10. Il peut sélectionne un créneau dans le calendrier et le mettre: disponible, réservé, bloqué.
 11. Ils cliquent sur «save».
 12. Le système enregistre le statut.
- **Enchaînements alternatifs:/**
- **Enchaînements d'erreur**

E1: Session expiré.

Après la session termine (après 30 min) , il doit s'authentifier.

e) Description textuelle de gérer ses rendez-vous

- **Titre:** évaluer ses rendez-vous.
- **Résumé:** Ce cas d'utilisation permet au médecin et spécialiste d'évaluer leurs rendez-vous (fait ou non fait) .
- **Acteur principal:** Médecin/spécialiste.
- **Précondition:** Médecin et spécialiste disposent compte valide.
- **Scénario nominale:**

1. Le médecin/spécialiste se connecte au système avec ses identifiants.

2. Le système affiche son tableau de bord qui contient la liste des prochaines rendez-vous .
 3. Il clique sur «mark as completed» à coté de rendez-vous s'il est terminé, et sur«mark as no show» s'il n'a pas fait.
 4. Le système l'affiche à coté de rendez-vous.
- **Enchaînements alternatifs:/**
 - **Enchaînement d'erreurs :**

E1: Session expiré .

Après la session termine (après 30 min) , il doit s'authentifier.

d) Description textuelle de prendre un rendez-vous chez un spécialiste

- **Titre:** Prendre un rendez-vous chez un spécialistes.
- **Résumé:** Ce cas d'utilisation permet au médecin de prendre un rendez-vous chez lui pour son patient.
- **Acteur principal:** Médecin.
- **Précondition:** Médecin dispose compte valide.
- **Scénario nominale:**
 1. Le médecin clique sur «request specialist appointment».
 2. Le système affiche un formulaire.
 3. Le médecin sélectionne une spécialité.
 4. Le système affiche une liste des spécialités.
 5. Le médecin choisir une spécialité.
 6. Le système affiche les créneaux.
 7. Le médecin choisir un créneau.
- **Enchaînements alternatifs:/**
- **Enchaînement d'erreurs :**

E1: Session expiré .

Après la session termine (après 30 min) , il doit s'authentifier.

f) Description textuelle de gérer ses rendez-vous

- **Titre:** évaluer ses rendez-vous.
- **Résumé:** Ce cas d'utilisation permet au médecin et spécialiste d'évaluer leurs rendez-vous (fait ou non fait) .
- **Acteur principal:** Médecin/spécialiste.
- **Précondition:** Médecin et spécialiste disposent compte valide.
- **Scénario nominale:**
 5. Le médecin/spécialiste se connecte au système avec ses identifiants.
 6. Le système affiche son tableau de bord qui contient la liste des prochaines rendez-vous .
 7. Il clique sur «mark as completed» à coté de rendez-vous s'il est terminé,et sur«mark as no show» s'il na pas fait.
 8. Le système l'affiche à coté de rendez-vous.
- **Enchaînements alternatifs:/**
- **Enchaînement d'erreurs :**

E1: Session expiré .

Après la session termine (après 30 min) , il doit s'authentifier.

e) Description textuelle de prendre un rendez-vous chez un spécialiste

- **Titre:** Prendre un rendez-vous chez un spécialistes.
- **Résumé:** Ce cas d'utilisation permet au médecin de prendre un rendez-vous chez lui pour son patient.
- **Acteur principal:** Médecin.
- **Précondition:** Médecin dispose compte valide.
- **Scénario nominale:**

8. Le médecin clique sur «request specialist appointment».
9. Le système affiche un formulaire.
10. Le médecin sélectionne une spécialité.
11. Le système affiche une liste des spécialité.
12. Le médecin choisir une spécialité.
13. Le système affiche les créneaux.
14. Le médecin choisir un créneau.

- **Enchaînements alternatifs:/**

- **Enchaînement d'erreurs :**

E1: Session expiré .

Après la session termine (après 30 min) , il doit s'authentifier.

3. Description textuelle des cas d'utilisation de réceptionniste

a) Description textuelle de consulter les rendez-vous

- **Titre:** Consulter les rendez-vous.
- **Résumé:** Ce cas d'utilisation permet au réceptionniste de consulter les rendez-vous des patients dans la clinique.
- **Acteur principal:** Réceptionniste.
- **Précondition:** La réceptionniste dispose compte valide.
- **Scénario nominale:**
 1. La réceptionniste se connecte au système avec ses identifiants.
 2. Le système affiche son tableau de bord.
 3. Le système affiche la liste des rendez-vous.
- **Enchaînement alternatif :**
 - ◆ A1 : Trop de rendez-vous.
 - ◆ Elle peut utiliser la barre de recherche pour filtrage par patient et par médecin.
- **Enchaînement d'erreurs :**

E1 : mot de passe d'authentification oublié.

Réinitialisation de mot de passe.

E2: Session expiré .

Après la session termine (après 30 min) , il doit s'authentifier.

b) Description textuelle de créer les rendez-vous

- **Titre:** Créer les rendez-vous.
- **Résumé:** Ce cas d'utilisation permet au réceptionniste de créer les rendez-vous des patient non enregistrés dans la clinique.
- **Acteur principal:** Réceptionniste.
- **Précondition:** La réceptionniste dispose d'un terminal avec accès à Internet, l'application est accessible.
- **Scénario nominale:**
 1. La réceptionniste se connecte au système avec ses identifiants.
 2. Le système affiche son tableau de bord.
 3. Elle clique sur «book new appointment».
 4. Le système affiche une interface de type de patient.
 5. La réceptionniste choisit le type de patient :enregistré, non enregistré.
 6. Si elle choisit «registred patient», elle sélectionne son nom, si elle choisit «walk-in patient» elle remplit son nom et son numéro de téléphone.
 7. Le système affiche une interface les médecins.
 8. Elle choisit le médecin .
 9. Le système affiche une interface des créneaux.
 10. Elle choisit un créneau disponible.
 11. Le système affiche une interface de détails de confirmation de rendez-vous.
 12. Elle enregistre le rendez-vous.
- **Enchaînement alternatif :**

A1 : la réceptionniste veut changer le médecin .

 - ◆ L'enchaînement A1 démarre à l'étape 8 du scénario nominal.
 - 9. Elle peut revenir en arrière par le bouton «previous».

- ◆ Le scénario nominal reprend au point 8.
- **Enchaînement d'erreurs :**
 - E1: Session expiré .
 - Après la session termine (après 30 min) , il doit s'authentifier.

4. Description textuelle des cas d'utilisation de admin

a) Description textuelle de gérer les comptes des utilisateurs

- **Titre:** gérer les comptes des utilisateurs.
- **Résumé:** ce cas d'utilisation permet au administrateur de gérer les comptes des utilisateurs: visualiser, modifier , créer, supprimer.(CRUD)
- **Acteur principal:** L'administrateur.
- **Précondition:** L'administrateur dispose compte valide.
- **Scénario nominale:**
 1. L'administrateur se connecte au système avec ses identifiants.
 2. Le système affiche son tableau de bord.
 3. Le système affiche la liste des patients, la liste des médecins, la liste des réceptionnistes.
 4. Il clique sur «view» situé à cote de chaque utilisateur pour voir plus des information (leurs profils).
 5. Il clique sur «edit» situé à cote de chaque utilisateur pour modifier leur profil (informations personnels).
 6. Il clique sur «delete» situé à cote de chaque utilisateur pour supprimer leur compte.
 7. Il clique sur les boutons «add new doctor»,«create new patient», «add receptionist» situé dans leurs listes respectives pour créer médecin, patient, réceptionniste.
 8. Il remplit les formulaires pour la création des nouveaux utilisateurs.
 9. Le système enregistre les modifications et les nouveaux compres.
- **Enchaînement alternatif :/**
- **Enchaînement d'erreurs :**

E2: Session expiré .

Après la session termine (après 30 min) , il doit s'authentifier.

b) Description textuelle de gérer les rendez-vous

- **Titre:** Gérer les rendez-vous .
- **Résumé:** Ce cas d'utilisation permet à l'administrateur de voir les rendez-vous, les modifier, les supprimer.
- **Acteur principal:** L'administrateur.
- **Précondition:** L'administrateur dispose d'un terminal avec accès à Internet, l'application est accessible.
- **Scénario nominale:**
 1. L'administrateur se connecte au système avec ses identifiants.
 2. Le système affiche son tableau de bord.
 3. L'administrateur clique «view all appointments» pour voir tous les rendez-vous.
 4. Le système affiche la liste de tous les rendez-vous.
 5. L'administrateur clique «view» pour voir les détails de chaque rendez-vous.
 6. Le système affiche les détails de rendez-vous.
 7. L'administrateur clique «edit» pour modifier chaque rendez-vous
 8. Le système affiche formulaire
 9. L'administrateur remplit le formulaire et enregistre les modifications.
 10. L'administrateur clique «delete» pour supprimer le compte
 11. Le système enregistre les modifications.
- **Enchaînement alternatif :**
 - ◆ A1 : l'administrateur veut voir les rendez-vous les plus récent.
 - ◆ L'enchaînement A1 démarre à l'étape 2 du scénario nominal.
 3. L'administrateur visualiser les rendez-vous les plus récent dans la section «recent appointments» affiché dans le tableau de bord
 - ◆ A2 : l'administrateur veut voir un rendez-vous particulier

- ◆ L'enchaînement A2 démarre à l'étape 4 du scénario nominal.
5. L'administrateur utilise la barre des recherches
- **Enchaînement d'erreurs :**

E1: Session expiré .

Après la session termine (après 30 min) , il doit s'authentifier.
- d) Description textuelle de consulter plaintes**
- **Titre:** Consulter les plaintes .
 - **Résumé:** Ce cas d'utilisation permet au administrateur de voir les plaintes des patients. **Acteur principal:** L'administrateur.
 - **Précondition:** L'administrateur a un compte.
 - **Scénario nominale:**
 1. L'administrateur se connecte au système avec ses identifiants.
 2. Le système affiche son tableau de bord.
 3. L'administrateur clique «accéder au réclamation».
 4. Le système affiche la liste des réclamations
 5. Il clique sur «détail» pour voir les détails.
 - **Enchaînement alternatif :/**
 - **Enchaînement d'erreurs :**

E1: Session expiré .

Après la session termine (après 30 min) , il doit s'authentifier.

5. Diagramme de séquence de réceptionniste

a) Diagramme de séquence pour consulter les rendez-vous

Ce diagramme montre le flux d'interactions entre le réceptionniste, le système de gestion, et la base de données pour consulter la liste des rendez-vous. Ce processus facilite la visualisation rapide et précise des rendez-vous planifiés, permettant au réceptionniste de gérer efficacement son planning.

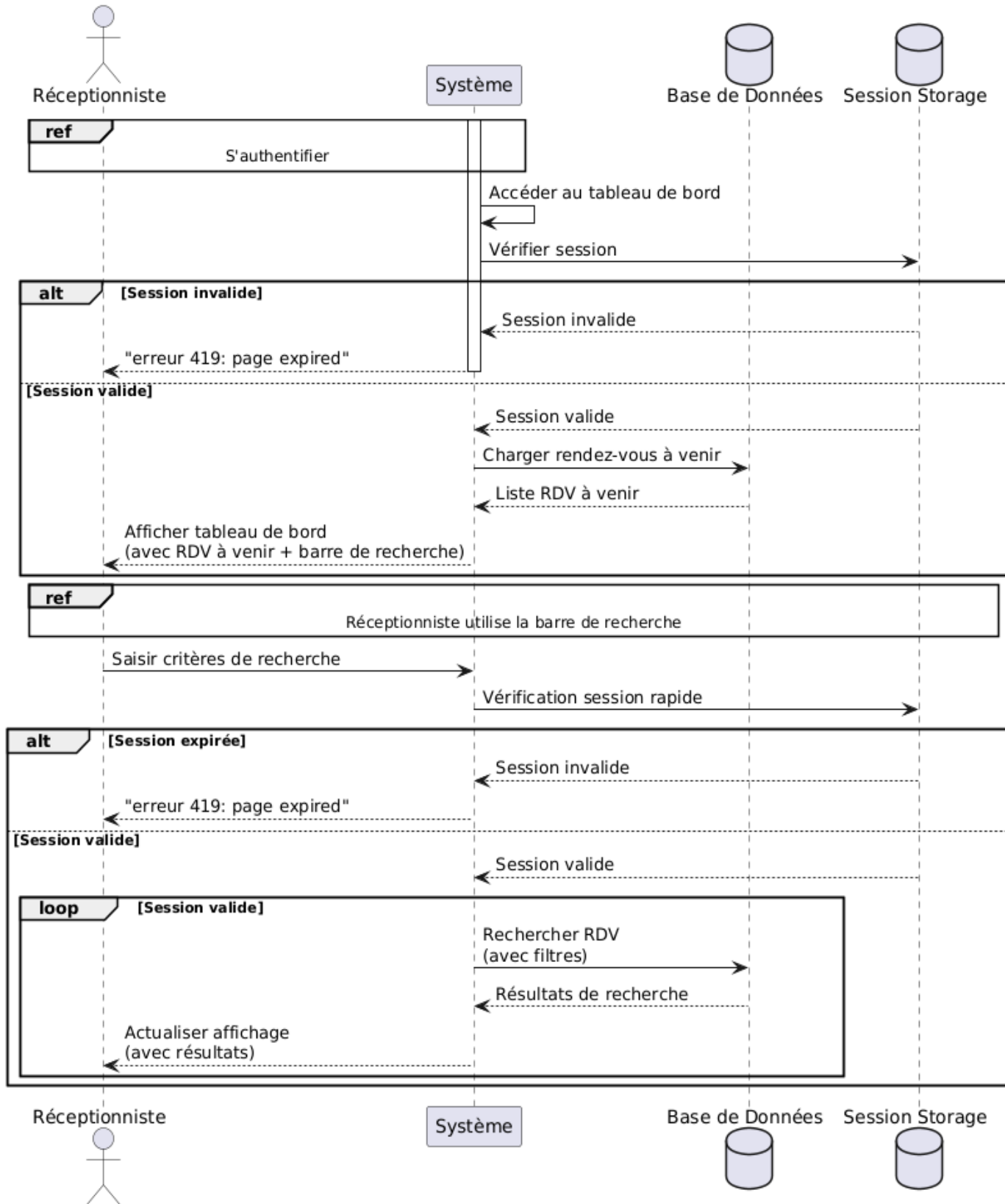


Figure 45:Diagramme de séquence pour consulter les rendez-vous

b) Diagramme de séquence pour créer les rendez-vous

Ce processus montre comment un réceptionniste peut ajouter un rendez-vous dans le système, en passant par la sélection, la saisie, puis l'enregistrement dans la base de données, avec une confirmation finale. Ce flux assure une gestion efficace et cohérente des rendez-vous.

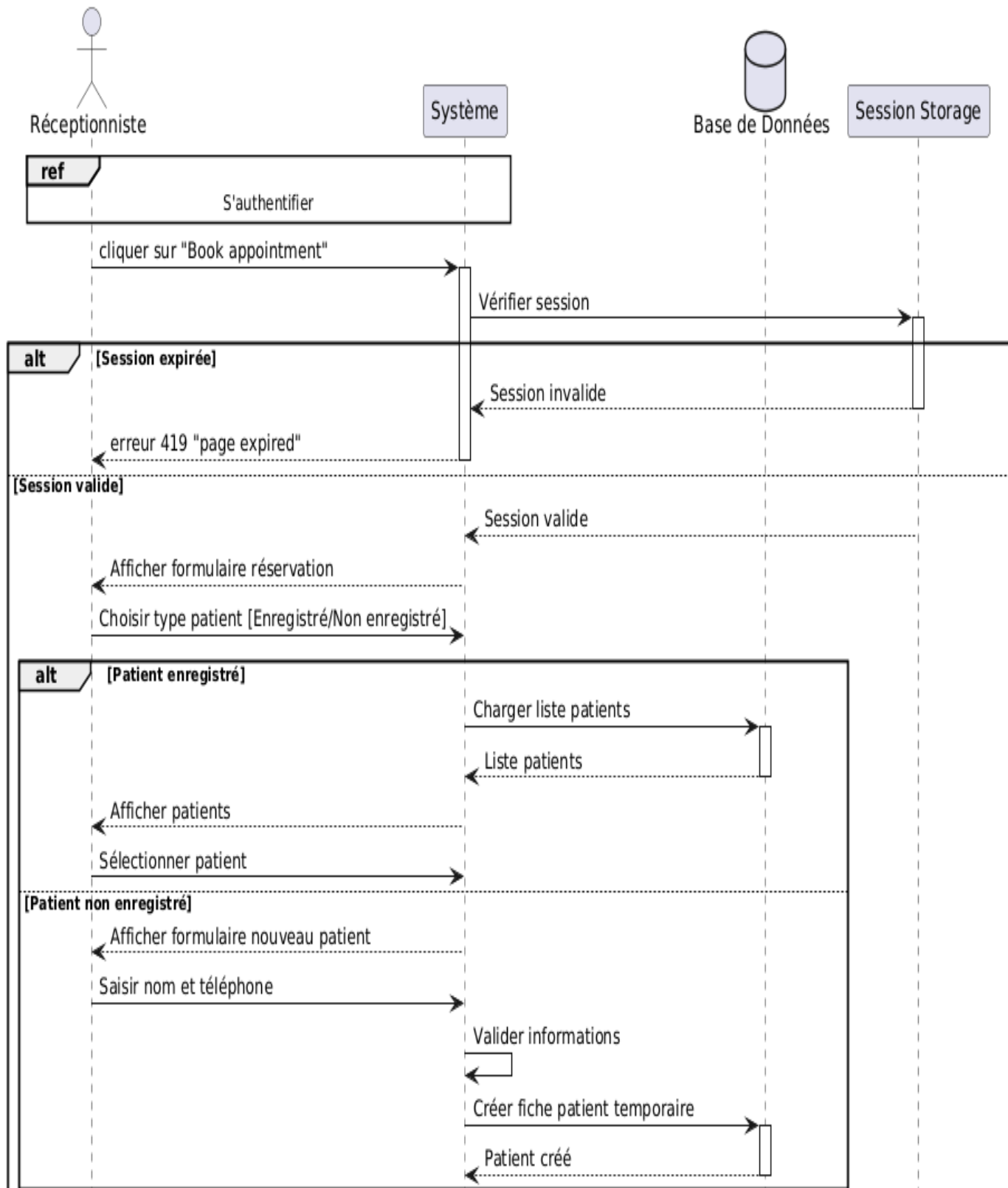


Figure 46: Diagramme de séquence pour créer les rendez-vous partie 1

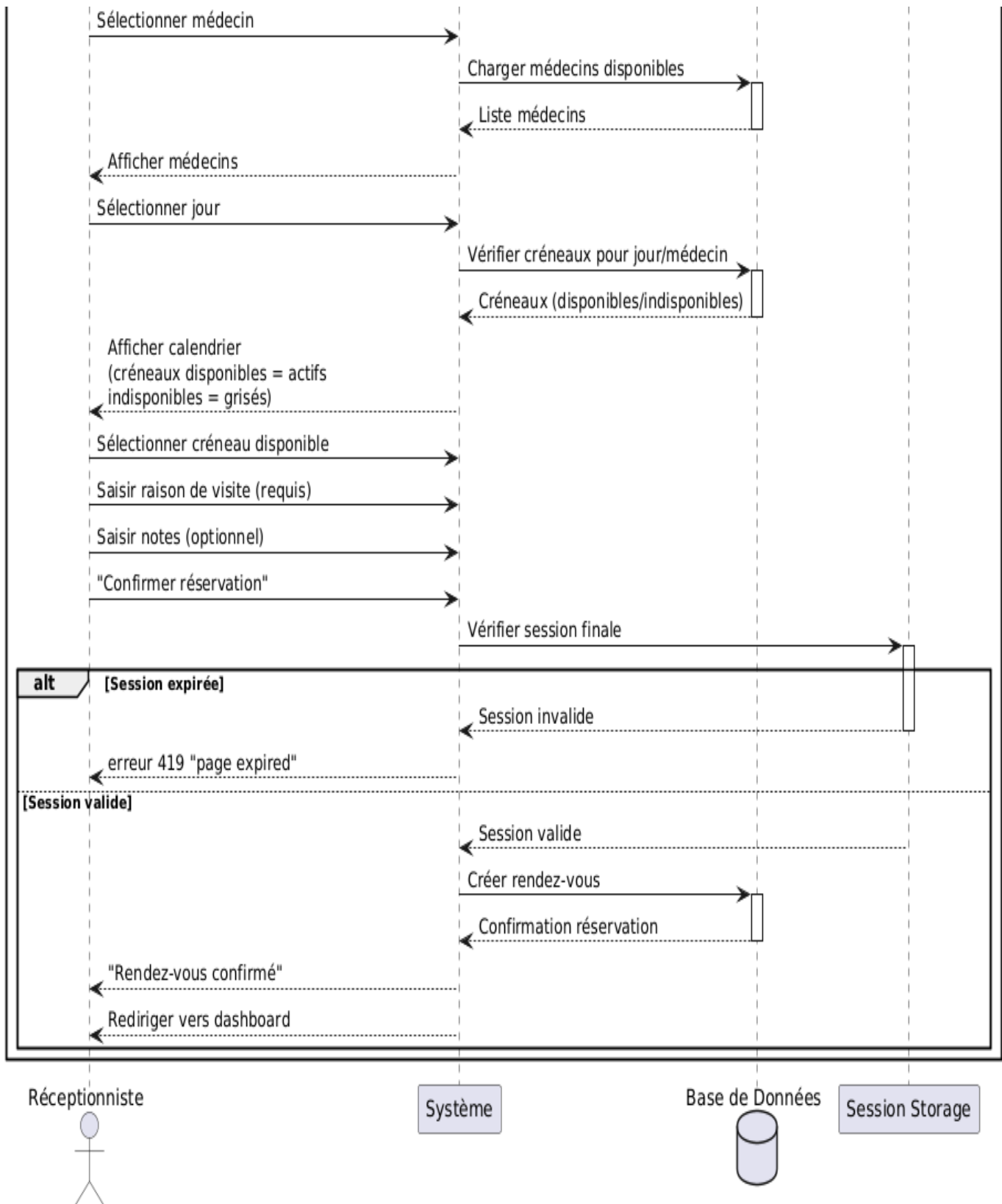


Figure 47: Diagramme de séquence pour créer les rendez-vous partie 2

6. Diagramme de séquence administrateur

a) Diagramme de séquence pour gérer les comptes des utilisateurs

Ce diagramme illustre le processus par lequel un administrateur gère les comptes utilisateurs via une interface. L'interaction commence par la connexion, puis se déploie en plusieurs actions possibles : création, modification ou suppression de comptes.

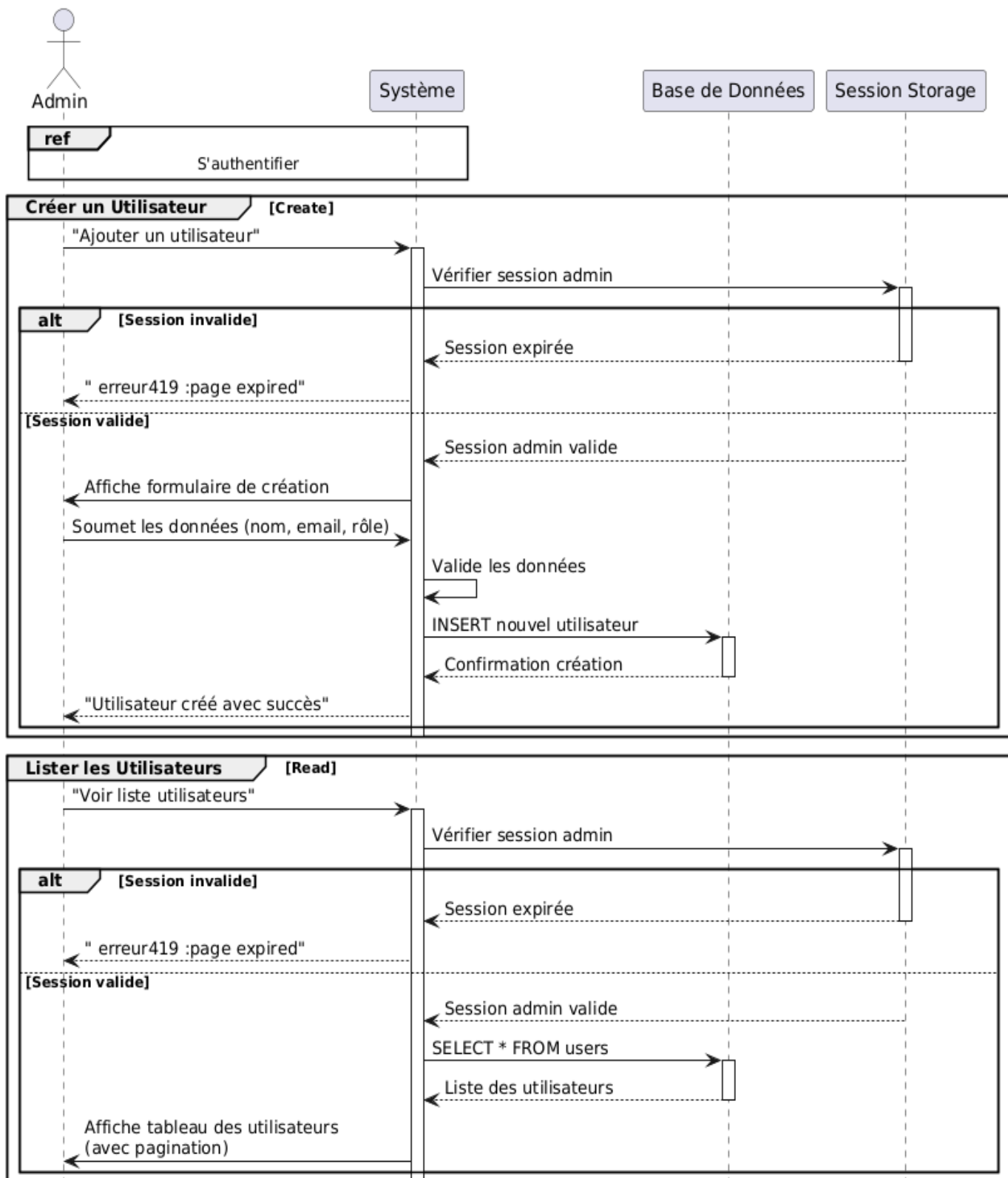


Figure 48:Diagramme de séquence pour gérer les comptes des utilisateurs partie 1

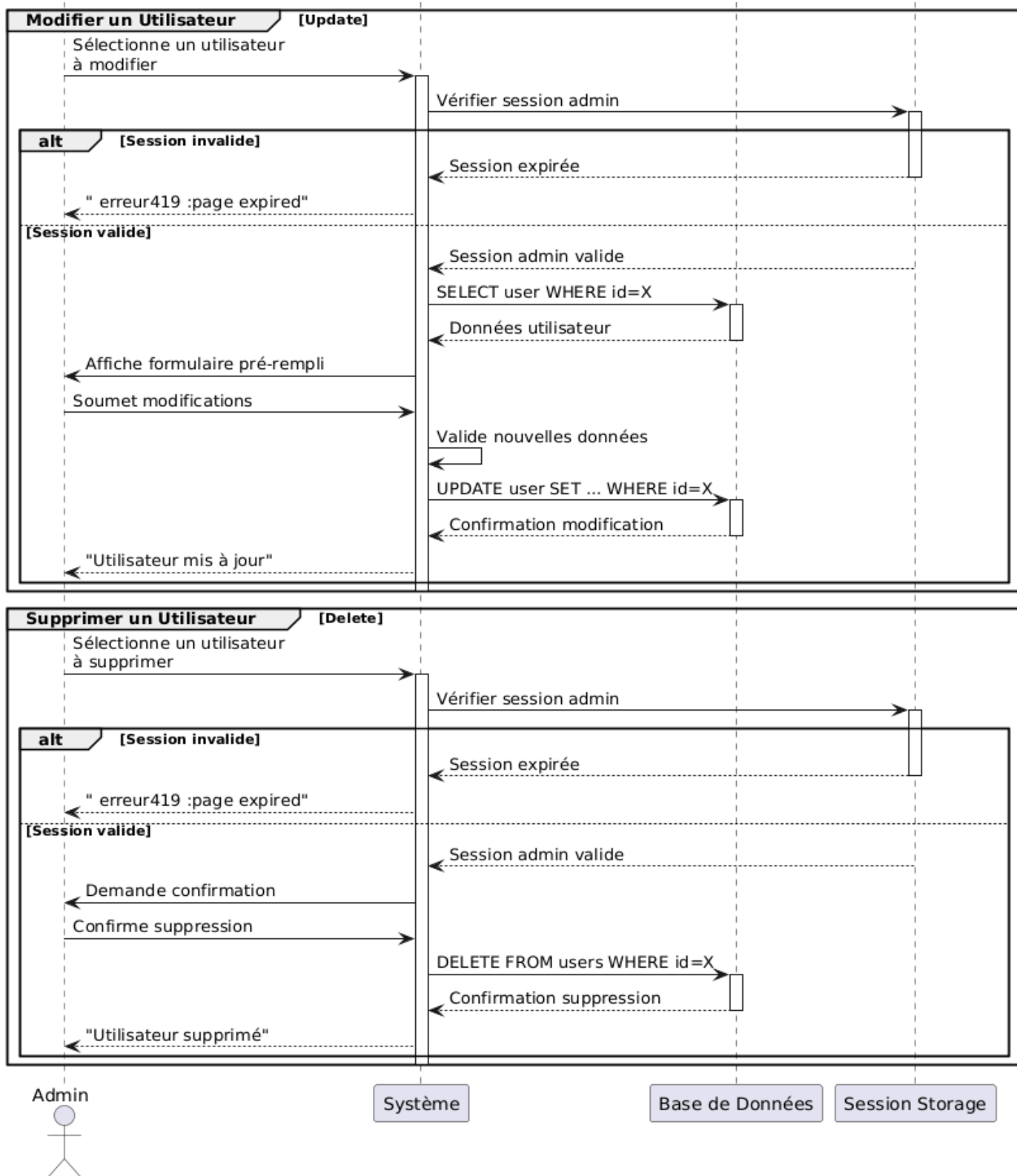


Figure 49:Diagramme de séquence pour gérer les comptes des utilisateurs partie 2

b) Diagramme de séquence pour administrateur gérer les rendez-vous

Ce diagramme illustre le processus par lequel un administrateur gère les rendez-vous dans un système. L'administrateur peut créer, modifier ou supprimer des rendez-vous en interagissant avec le système, qui à son tour, interagit avec la base de données pour stocker ou mettre à jour les informations

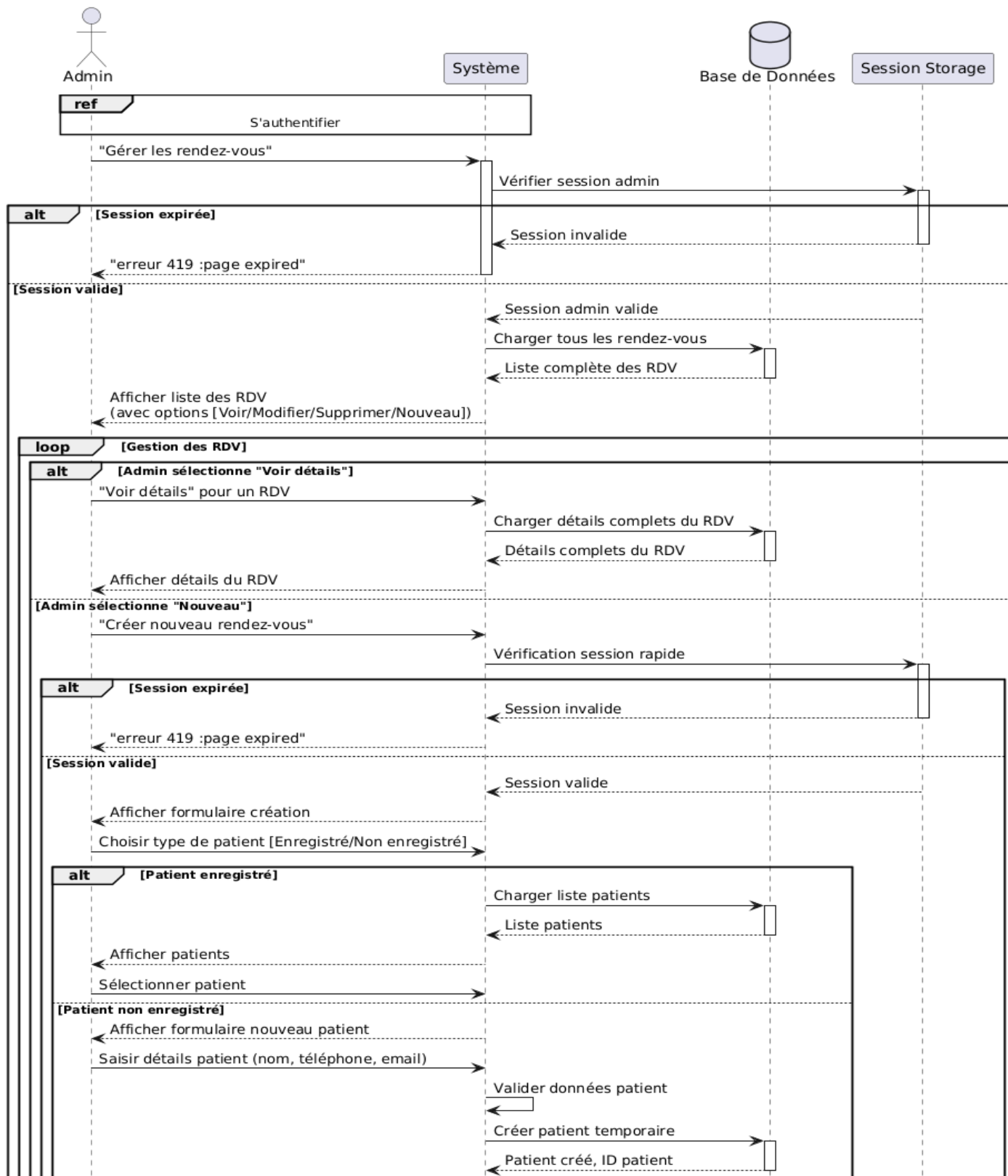


Figure 50: Diagramme de séquence pour administrateur gérer les rendez-vous partie 1

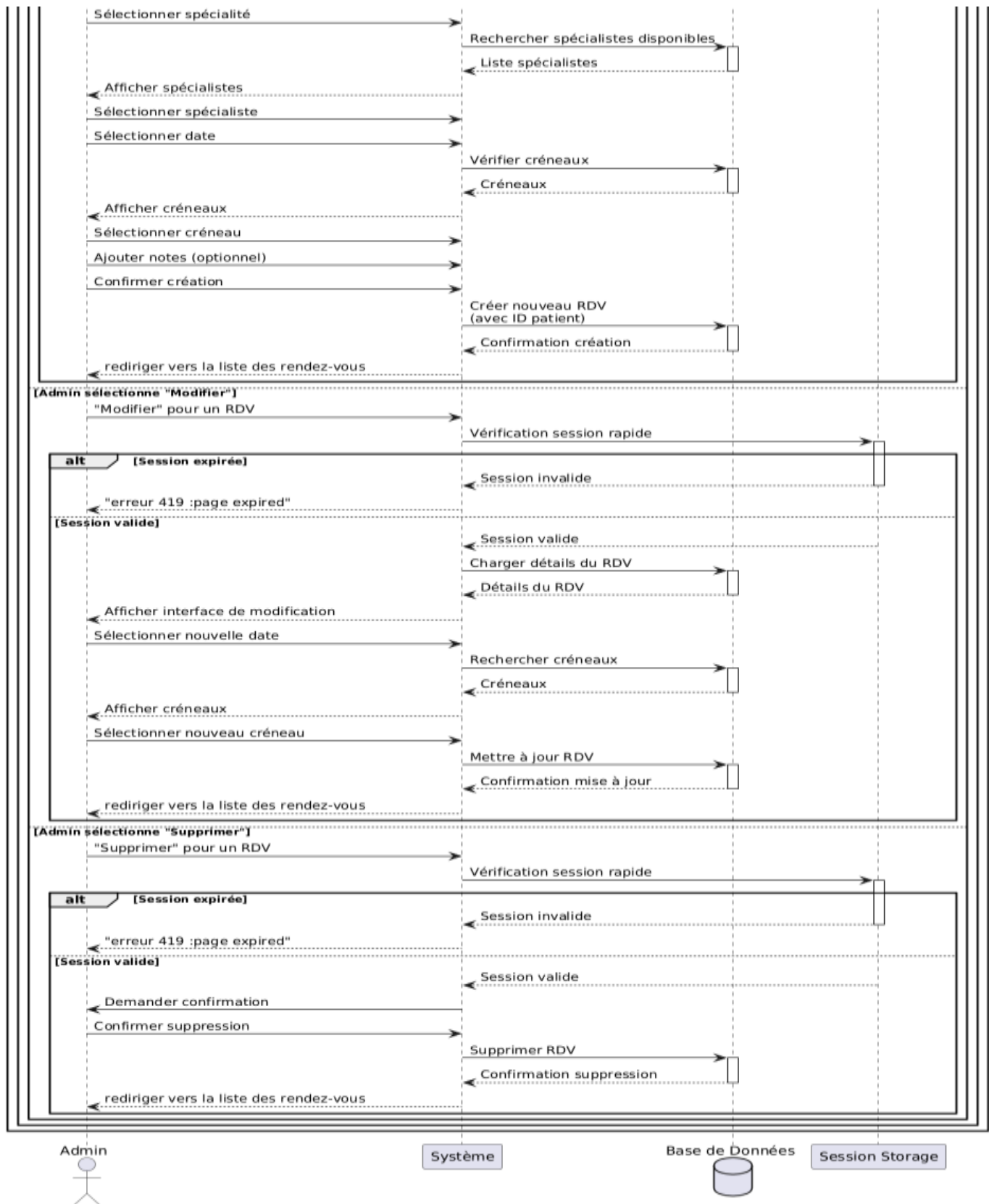


Figure 51: Diagramme de séquence pour administrateur gérer les rendez-vous partie 2

d) Diagramme de séquence de consulter plaintes

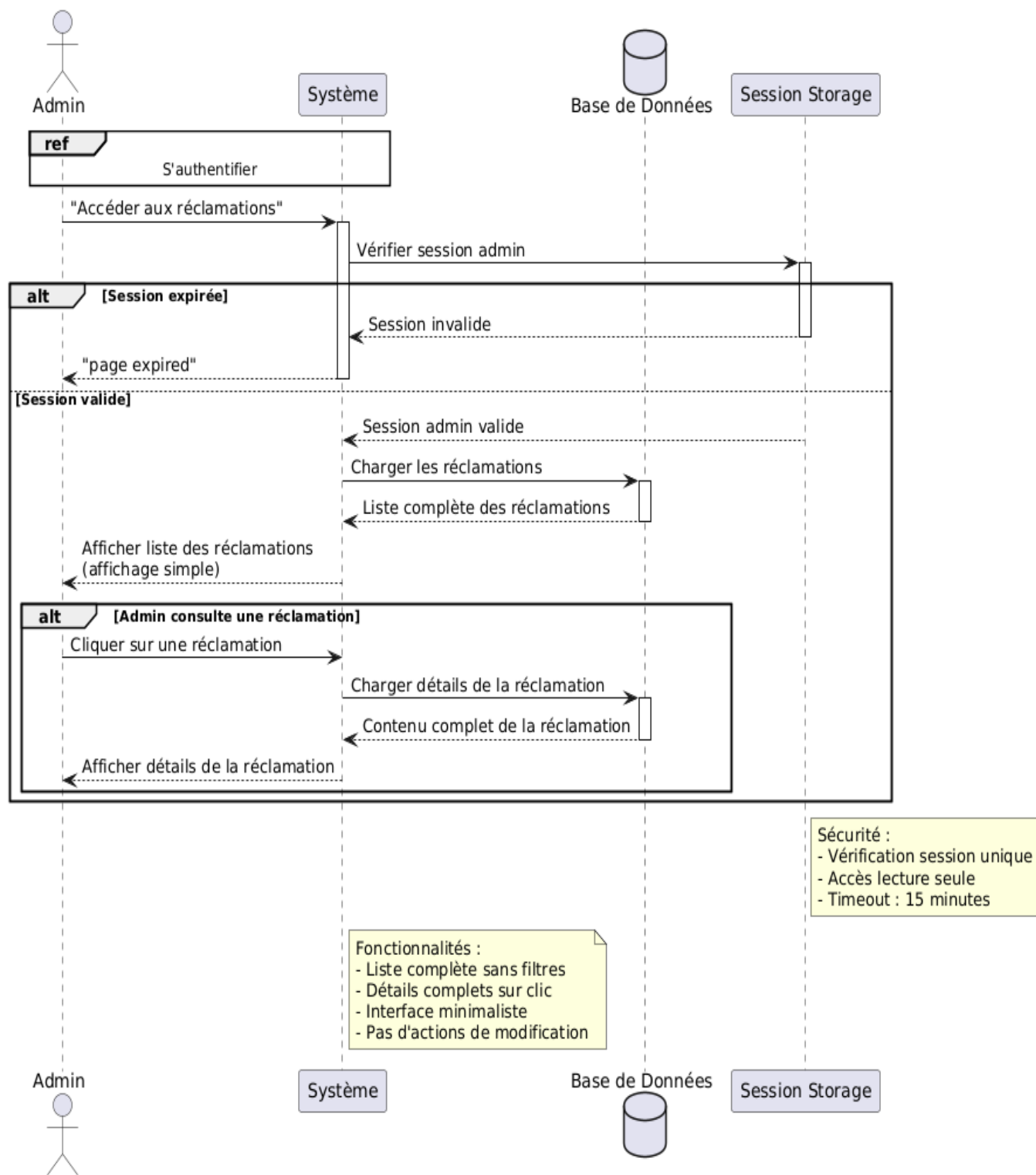


Figure 52:Diagramme de séquence de consulter plaintes

7. Diagramme de séquence pour tous les utilisateurs

a) Diagramme de séquence pour s'authentifier

Le diagramme de séquence montre les étapes que l'utilisateur (patient, administrateur, médecin, réceptionniste) doit suivre pour s'authentifier dans le système. Cela commence par l'ouverture de la page d'authentification, la saisie des informations d'identification, l'envoi de la demande d'authentification, la vérification par le système et la réponse retournée à l'utilisateur. Ce diagramme permet de visualiser la progression de l'utilisateur lors de la phase d'authentification et aide à identifier les étapes qui peuvent potentiellement causer des erreurs ou des problèmes.

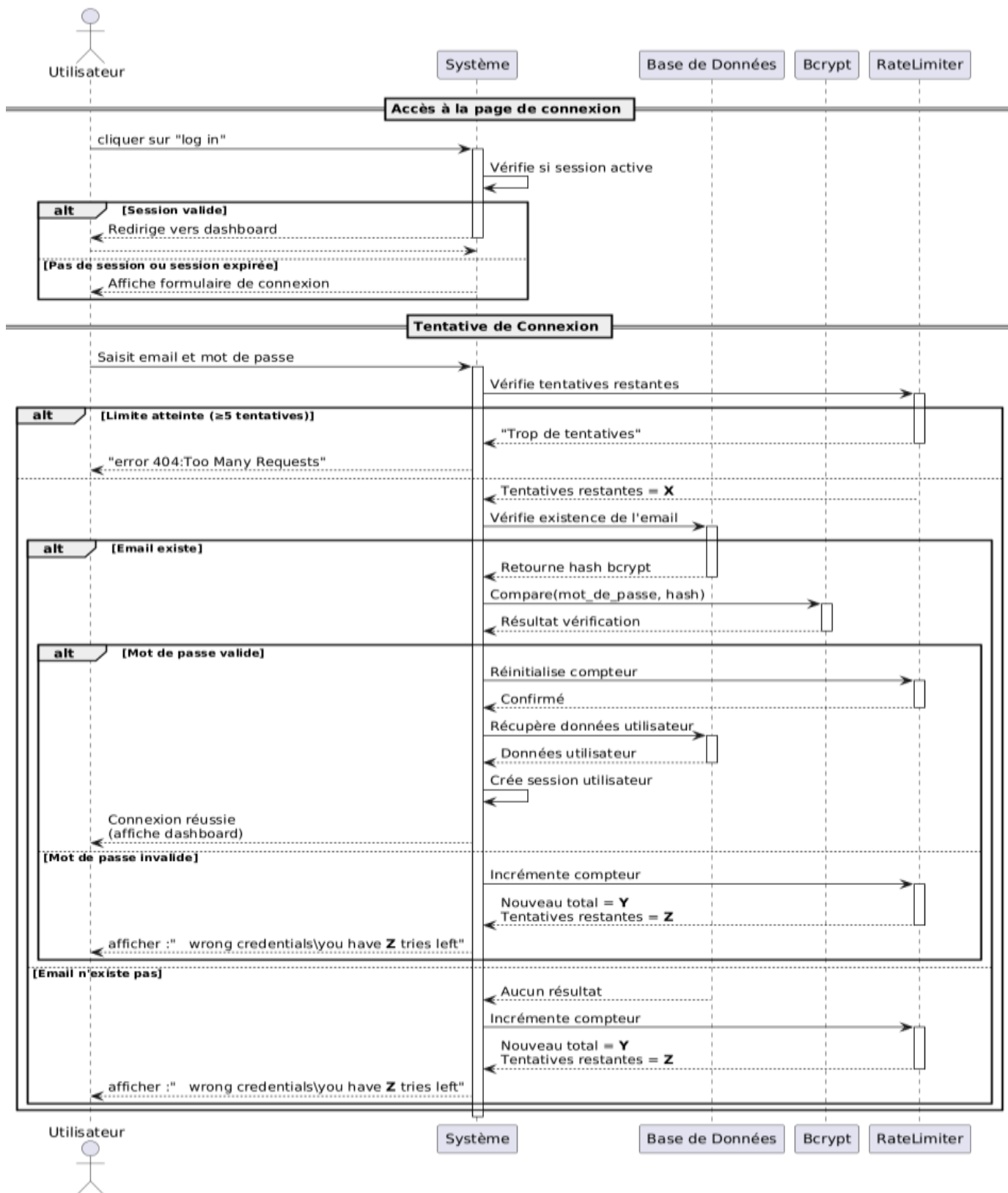


Figure 53:Diagramme de séquence pour s'authentifier

b) Diagramme de séquence pour modifier le profil

Le diagramme de séquence illustre les étapes qu'un utilisateur doit suivre pour modifier son profil dans le système. Il commence par l'authentification de l'utilisateur afin d'avoir l'accès à la page de modification du profil, puis la saisie des nouvelles informations et leurs soumission et vérification par le système, et enfin la confirmation de la mise à jour.

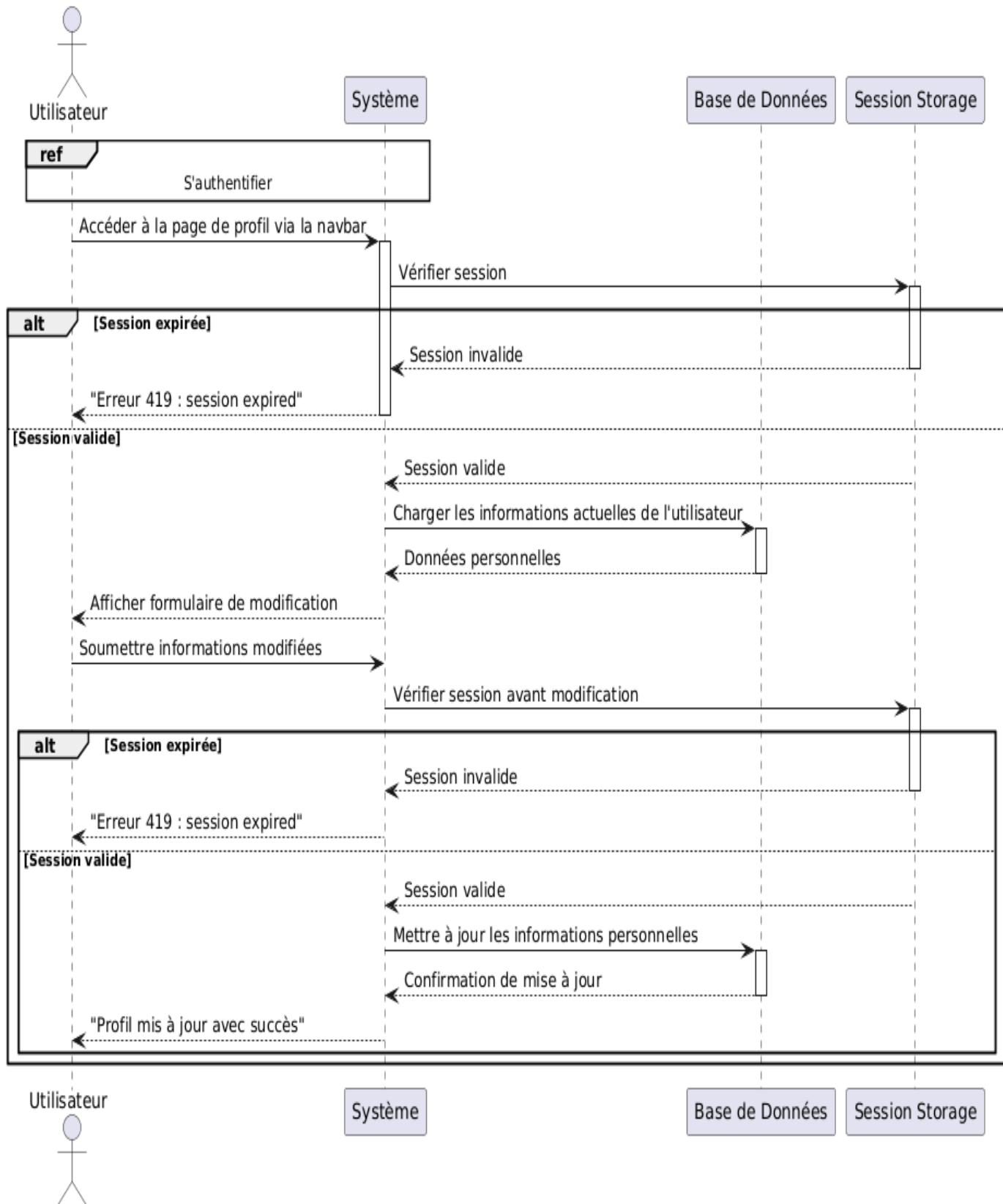


Figure 54:Diagramme de séquence pour modifier le profil

c) Diagramme de séquence pour réinitialiser le mot de passe

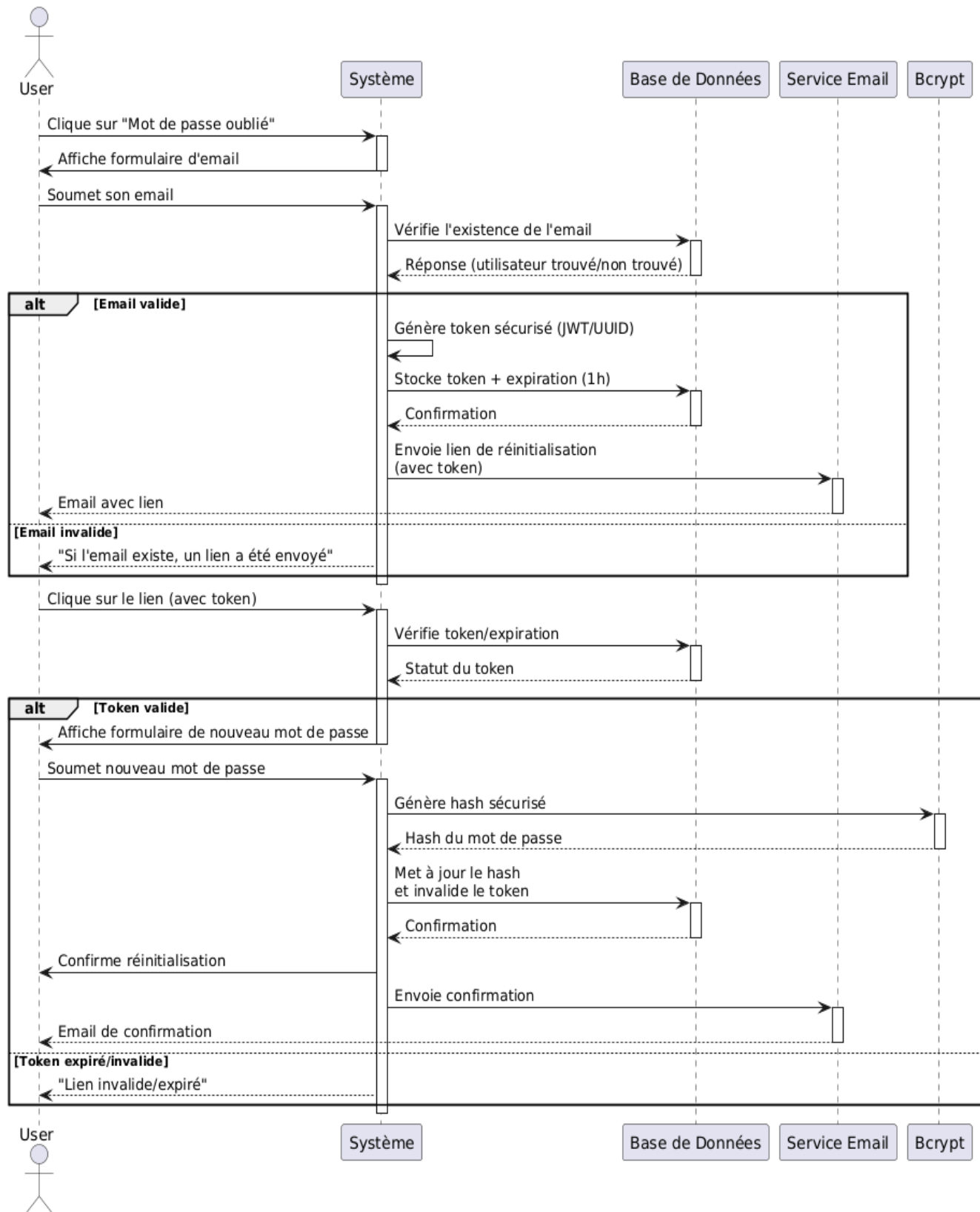


Figure 55:Diagramme de séquence pour réinitialiser le mot de passe

B2. Diagramme de séquence de médecin/spécialiste

a) Diagramme de séquence de gérer ses rendez-vous

Ce cas d'utilisation permet au médecin d'organiser ses rendez-vous.

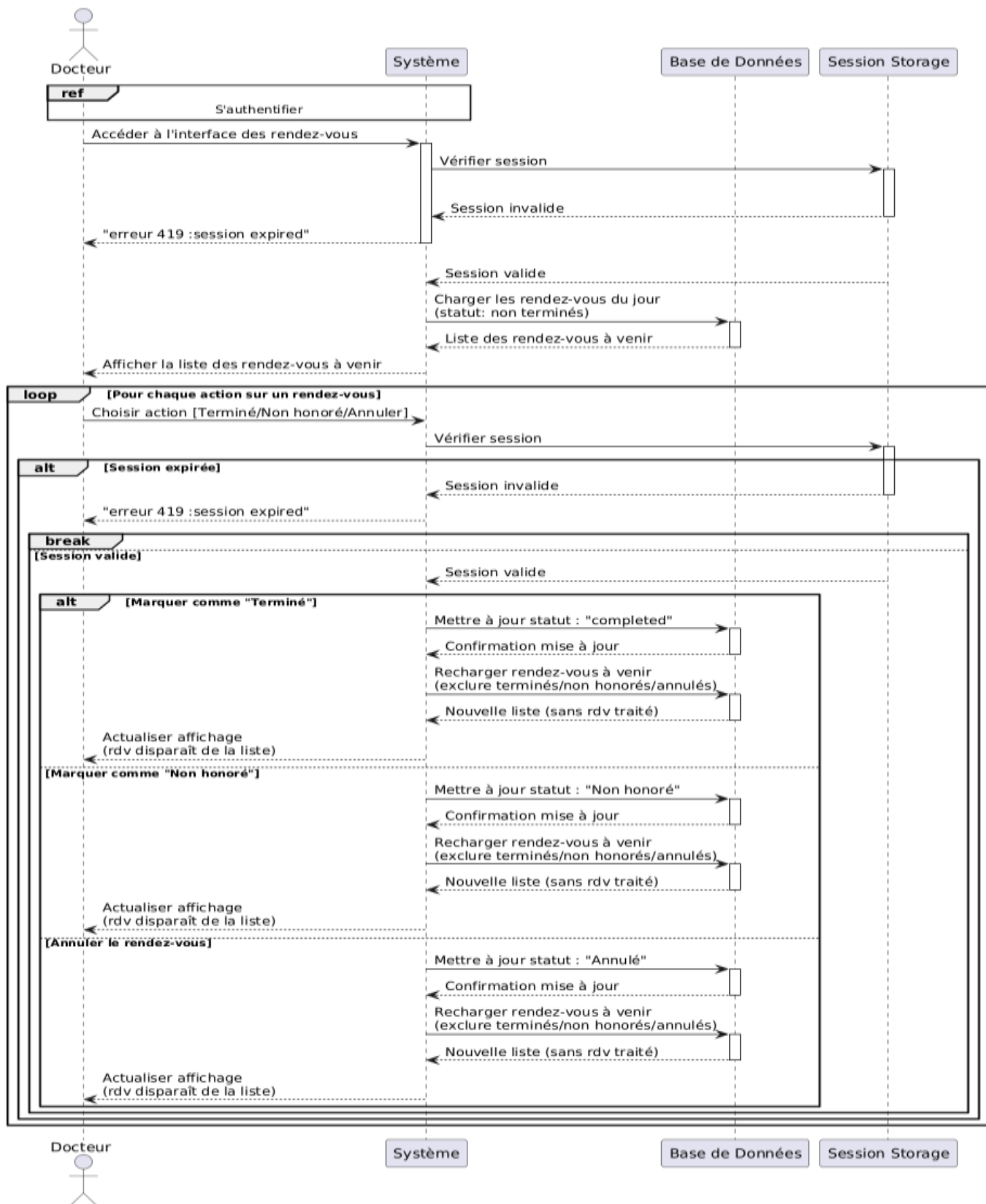


Figure 56:Diagramme de séquence de gérer ses rendez-vous

d) Diagramme de séquence de prendre un rendez-vous chez un spécialiste

Ce diagramme de séquence montre comment un médecin peut planifier un rendez-vous pour un patient chez un spécialiste. Le processus commence par la demande du médecin, la vérification des disponibilités par le système, la sélection d'un créneau, l'enregistrement du rendez-vous, puis la confirmation. Ce processus facilite la coordination entre médecins, spécialistes, et le système de gestion pour assurer une organisation efficace des rendez-vous médicaux.

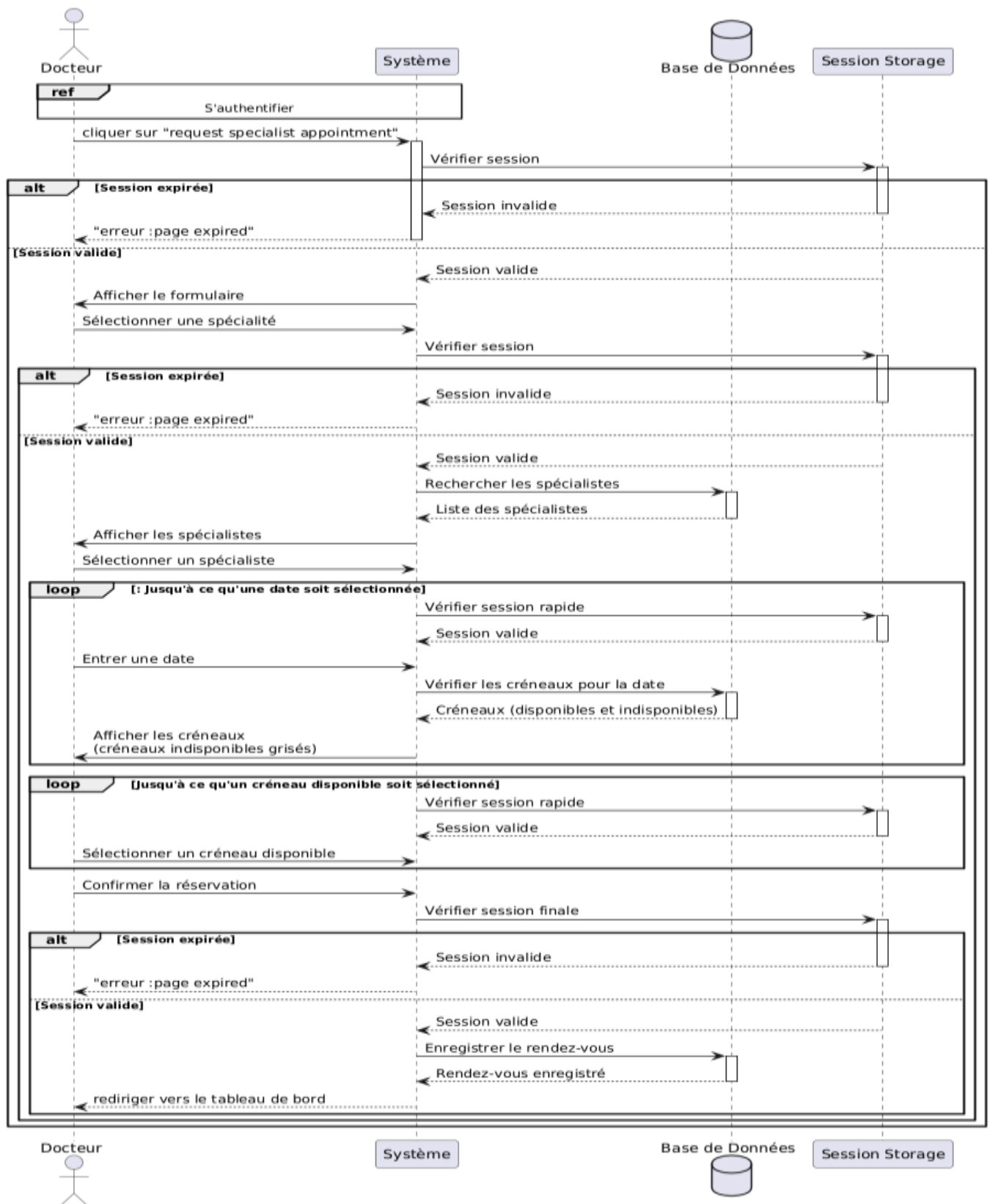


Figure 57:Diagramme de séquence de prendre un rendez-vous chez un spécialiste

i) Description textuelle de réinitialiser mot de passe

j) Titre: Réinitialiser mot de passe

- **Résumé:** Ce cas d'utilisation Permet à le patient de gérer son rendez-vous.
- **Acteur :** Patient.
- **Précondition:** Le patient doit avoir un rendez-vous.

Scénario nominal

1. Le patient se connecte à son compte.
2. Il clique «réinitialiser mot de passe».
3. Il remplit formulaire avec nouveau mot de passe.
4. Le système enregistre le mot de passe.

- **Enchaînements alternatifs: /**

- **Enchaînements d'erreur:**

E1: Session expiré .

Après la session termine (après 30 min) , il doit s'authentifier.

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

جامعة عبد الحميد بن باديس-مستغانم-
كلية العلوم الاجتماعية والإنسانية

تصريح شرفي خاص بالالتزام بقواعد النزاهة العلمية
لإنجاز البحث

أنا الممضي أدناه،

الطالب(ة): ربيغي رضوان ضياء الدين رقم التسجيل الجامعي: 191935004360
الحامل لبطاقة التعريف الوطنية رقم: 407176979. والصادرة بتاريخ: 01 أكتوبر 2023.
عن بلدية: مستغانم

المسجل بكلية العلوم الدقيقة والإعلام الآلي / قسم الاعلام الآلي
شعبة رياضيات واعلام الي / التخصص هندسة أنظمة المعلومات

والمكلف بإنجاز مذكرة ماستر بعنوان: Conception et Implémentation d'une application Web Sécurisé
avec la gestion d'authentification et d'autorisation (cas d'étude : gestion de dossier médical au sein
d'une clinique)

أصرح بشرفي أنني ألتزم بمراعاة المعايير العلمية والمنهجية ومعايير الأخلاقيات العلمية والنزاهة الأكاديمية
المطلوبة في إنجاز البحث ، وأتحمل المسؤولية الشخصية عن كل المحتوى المتضمن في البحث المذكور أعلاه .

التاريخ: 2025/06/23

إمضاء المعني

