

Chapitre 2. Expression de besoin

Dans le chapitre « Besoins fonctionnelles et techniques » nous spécifions toutes les fonctionnalités que possède la solution, les langages utilisés pour la réaliser ainsi, toutes les interfaces possibles de chaque utilisateur de la solution.

1. Cadrage du projet

1.1 Objectifs du projet

Le centre hospitalier Hassan II de Fès dispose d'un service de Génétique (service Pilote pour eRDV) unique dans la région nord du Maroc. Toutes les demandes des zones de cette région parviennent à l'Hôpital Hassan II de Fès.

Notre projet présente un dispositif qui permet la prise des rendez-vous sur Internet fiables, rapide et moins coûteuse vis-à-vis du temps pour les cadres médicaux et paramédicaux des différentes provinces. Ce dispositif se base sur plusieurs objectifs :

- Organiser la prise des rendez-vous pour les consultations via ce dispositif réservé aux cadres médicaux et paramédicaux.
- Les professionnels doivent aussi pouvoir accéder à partir des différentes provinces.
- Les utilisateurs doivent obligatoirement se connecter en tant que Professionnels pour pouvoir réserver des rendez-vous à leurs patients. L'administrateur du dispositif devra créer un login et un mot de passe pour chaque personnel qui souhaitera utiliser ce dispositif et le communiquer de manière sécurisée.
- Faciliter la prise de rendez-vous des patients aux cadres médicaux et paramédicaux des centres de santé (au lieu de les prendre manuellement en écrivant une demande) dans l'attente d'une confirmation de la part du CHU.
- Rappeler les patients de la date du Rendez-vous par l'envoi des SMS d'une manière automatisée.
- Élargir l'environnement de la prise de rendez-vous dans le CHU pour tous les provinces via ce dispositif.

1.2 Périmètre de projet

Positionnement du futur site :

Le futur dispositif aura pour vocation de s'intégrer à un site (qui a un accès restreint) propose la prise de rendez-vous d'une manière rapide et automatisée en garantissant un résultat pertinent.

Description générale de projet :

Ce site est à destination des cadres médicaux et paramédicaux, il s'est enrichi d'unités interactives, qui permettent de disposer de tous les types de consultations disponibles avec leurs activités correspond.

2. Description des besoins fonctionnels

2.1 Description de la structure du site

Nous mettons au service des cadres médicaux et paramédicaux des différents outils :

- Pages public : contient une présentation générale du site, et des différents services proposés.
- Espace d'administration : gestion du site web, paramètres du système, gestion des utilisateurs (cadres médicaux et paramédicaux) et gestion des messageries (Sauvegarde des messages acceptés par l'administrateur).
- Espace médecins : réserver aux cadres médicaux et paramédicaux déjà inscrit toutes les fonctionnalités proposées par ce site (l'ajout, l'annulation, la modification et le suivi d'une consultation).

2.2 Besoins fonctionnels en front office

L'authentification : Les utilisateurs du site seront dans l'obligation de s'authentifier à l'aide d'un nom d'utilisateur et d'un mot de passe unique (déjà obtenu lors de l'inscription dans le site). Ils disposent d'un accès facile à leurs informations, ainsi que la possibilité de les modifier à tout moment. Les Professionnels associés pourront accéder via web pour :

- Faire demandes de rendez-vous
- Suivi d'état RDV (demande En liste d'attente, en cours ou réalisé)
- Visualisation des Résultats
- Recherche : Il est possible d'utiliser un champ de recherche sur toutes les pages du site. Celui-ci se présentera sous la forme classique d'une zone de texte destinée à recevoir la saisie des mots-clés recherchés avec un bouton valider.

2.3 Fonctionnalités de projet

Nous vous présentons ainsi, dans le tableau suivant la liste des fonctionnalités prévus du module eRDV.

Tableau 1. Les principales fonctionnalités du module eR.D.V

Fonctionnalités	Description
Paramètres générales du site	
GENERALE	Les paramètres générale du site web (les métas titre, description, keywords,...)
THEME	Choisir le thème à appliquer au site Web
AIDE	Expliquer le mode d'emploi de chaque élément du site
Gestion des utilisateurs	
PROFIL	Le profil contient les informations d'utilisateur et l'accès au profil se fait à partir de l'authentification
AJOUTER UN UTILISATEUR	Le responsable du module eRDV peut ajouter les utilisateurs sous la demande du service concerné
MODIFIER UN UTILISATEUR	Le responsable du module peut modifier des informations concernant un utilisateur sous la demande du service concerné
DESACTIVER UNE SESSION UTILISATEUR	Le responsable du module peut désactiver la session d'un utilisateur sous la demande du service concerné
LISTER LES UTILISATEURS	Afficher une liste d'utilisateurs avec un tri personnalisé
RECHERCHER DES UTILISATEURS	Recherche des utilisateurs à partir d'une zone de recherche
Gestion des Messagerie	
RENDEZ-VOUS	Le responsable peut accepter ou annuler les rendez-vous envoyés par les médecins des centres régionaux
Inscription	
L'ACTIVATION DES SESSIONS UTILISATEUR	Dans cette interface, une liste des nouveaux utilisateurs inscrits dans le module eRDV s'affiche. Après une vérification du responsable, le système génère un email contenant un lien d'activation de la session d'utilisateur concerné
Gestion des consultations	
PRENDRE UN RENDEZ-VOUS	Prendre un rendez-vous en envoyant une demande à l'administrateur Deux cas qui se posent : 1/ Donner rendez-vous à des patients déjà existants dans la base de données, chaque fois qu'un permis a été attribué de la part de l'hôpital. 2/ Donner rendez-vous à des nouveaux patients, il sera nécessaire d'ajouter le nouveaux patient en remplissant les champs concernant ces informations personnelles des nouveaux patients. Dans ce module, il y a plusieurs types de rendez-vous, à savoir : <ul style="list-style-type: none"> • Rdv par Consultation- Médecin Spécialiste (rendez-vous UNIQUE). Choisir une seul consultation à partir d'un agenda simple • RDV multiple. On indique la Consultation et l'Activité voulue. Il se peut qu'une seule activité corresponde à plusieurs concepts, ce qui génère ce qu'on appelle les kits. • RDV par spécialité service. Le médecin doit choisir l'un des services pour réserver une consultation, puis cocher un ou plusieurs parmi les activités obtenues. Après le choix, le système permet au médecin de rechercher la disponibilité pour une période donnée et avoir comme résultat les dates disponibles de la période souhaitée c'est-à-dire obtenir tous les consultations qui font cet examen dans cette période, le médecin va sélectionner celle souhaitée. Chaque rendez-vous patient aura une Origine : Externe ou Hospitalisé. Les résultats dans la consultation seront accessibles en fonction de cette origine.
CONSULTER UN RENDEZ-VOUS	Le médecin sélectionne l'historique du patient via son IPP pour vérifier si le rendez-vous est déjà confirmé

ANNULER UN RENDEZ-VOUS	Si le médecin s'est trompé dans une consultation et peut l'annuler en remplissant le champ motif
MODIFIER UN RENDEZ-VOUS	Modifier les informations qui correspondent à la consultation choisie. Généralement le médecin peut reporter la date de la consultation.
Gestion d'envois des SMS	
ENVOYER DES SMS DE NOTIFICATION AUX PATIENTS	Rappeler les patients de la date du Rendez-vous par voie SMS d'une manière automatisée.

3. Description des besoins ergonomiques

3.1 Définir les règles ergonomiques

La conception de l'interface d'un site Web nécessite toutefois le respect de certaines règles. En effet, l'ergonomie est un concept primordial en informatique, car la facilité et l'efficacité d'une application reposent sur elle. Son but est de rendre l'interaction Homme/Machine :

- Efficace,
- Efficente,
- Satisfaisante.

Pour ce faire, il est de préférence de respecter les critères suivants :

- Le guidage : ensembles des moyens mis en œuvre pour orienter et conduire l'utilisateur lors de ses interactions avec l'interface sous quatre sous critères :
 - L'incitation. faire savoir à l'utilisateur où il se situe dans l'interface, l'état ou contexte dans lequel il se trouve, les outils d'aide et leur accessibilité.
 - Feedback immédiat. Concerne les réponses de l'ordinateur consécutives aux actions des utilisateurs (message d'erreur, message de validation ou l'échoue d'un email...)
 - Lisibilité. Les caractéristiques matérielles de présentation des informations qui doivent en faciliter la lecture.
 - La charge de travail. Concerne l'ensemble des éléments de l'interface qui ont un rôle dans la réduction (ou augmentation) de la charge perceptive et cognitive des utilisateurs.
- Contrôle explicite : Actions Explicites : le système doit exécuter seulement les actions demandées par l'utilisateur au moment où il les demande.
- Contrôle Utilisateur : L'utilisateur doit toujours avoir la main pour pouvoir contrôler le déroulement des traitements informatiques en cours, il peut naviguer comme il veut sur le site.

3.2 Formaliser une charte graphique ergonomique

L'objectif de cette charte est d'améliorer l'expérience des utilisateurs du site pour les inciter à plus l'utiliser. Dans ce but, certains éléments constitutifs de nom du site doivent être toujours les mêmes. Présentés dans la section « Règles obligatoires », ces éléments sont présents dans la zone identitaire et la zone de navigation.



Figure 14. La forme officielle d'un site web

3.3 Concevoir les tests utilisateurs

Le test utilisateur est une méthode d'évaluation qualitative basée sur l'observation individuelle d'utilisateurs cibles du site web. Plus précisément, les tests utilisateurs permettent d'observer directement le comportement des internautes pendant qu'ils visitent un site, d'identifier les dysfonctionnements éventuels et de repérer les difficultés qu'ils rencontrent. À partir du recueil de ces données, des recommandations précises sont formulées et permettront d'améliorer le site. Il existe plusieurs types de tests, on choisit le test en visite libre, c'est-à-dire de laisser l'utilisateur naviguer sur le site sans aucune directive spécifique. Au cours de ce test, les utilisateurs sont invités à penser à haute voix, à indiquer à l'ergonome ce qui leur pose problème. L'ergonome peut être plus « présent » et inciter les utilisateurs à dire, par exemple, pour chaque élément de navigation, les informations qu'ils pourraient trouver sur les pages correspondantes. Les tests utilisateurs peuvent être réalisés très tôt dans le processus de conception, c'est-à-dire dès que des choix de conception (structure, type de dialogue, organisation du contenu, icônes, boutons de commandes, liens, etc.) sont envisagés. Le temps de passation des tests ne doit pas dépasser 45 minutes. Au-delà de ce laps de temps, le niveau de vigilance et de motivation chutent et le test devient fastidieux.

4. Description des besoins graphiques

4.1 Définir les règles graphiques

Le fond d'écran : Couleur, texture ou image de fond. Le fond est toujours blanc. Cependant, le fond des tableaux peut être coloré. Il y a obligation d'utiliser les couleurs web. On conseille d'utiliser les

aplats et d'éviter les dégradés. Il est aussi interdit d'utiliser les images en fond d'écran. La charte graphique ne précise rien en ce qui concerne les textures mais on peut en déduire qu'elles sont interdites. Aucune image utilisée en fond d'écran.

Les images : nature et fonction, Couleur, cadrage, plans, angles, animation, autres. Pour ce qui est des images, la charte stipule que les images aient une définition de 72dpi. De plus, on conseille de les utiliser modérément car elles augmentent rapidement le poids de la page et par le fait même le temps de chargement.

Les photos doivent être compressées en JPG et les images et le texte en GIF. Le taux de compression est fonction du nombre d'images de la page. Plus le nombre d'images est élevé plus le taux de compression doit être bas. On suggère d'utiliser des images au cadrage rapproché avec peu de détails.

Toutes les images utilisées doivent être placées dans des tableaux. Leur grandeur dépend donc de l'endroit où elles seront situées.

L'icône graphique : est un élément graphique qui symbolise une action. On utilise des icônes graphiques dans la barre de navigation pour guider l'utilisateur à travers les différents outils accessibles. Les icônes doivent être très représentatives de l'intention du concepteur. Avant d'employer une icône, il faut s'assurer qu'elle est significative et qu'elle ne risque pas d'offrir diverses interprétations.

Disposition du texte : Il est possible de disposer un texte de plusieurs façons dans une page-écran. On peut l'afficher à la grandeur de l'écran, le disposer en tableau ou le structurer dans un environnement par cadre, mais toujours dans un objectif de lisibilité

Logo : Le logo est présent sur toutes les pages et joue le rôle de bouton de retour à la page d'accueil. Il est situé en haut à gauche de la page, un classique.

Utilitaires : Les utilitaires sont annoncés dès l'accueil et regroupés dans une deuxième barre horizontale, très discrète, où les mots clés en gris annoncent des outils d'aide à la navigation ainsi que des ressources humaines à contacter et une Foire aux questions.

5. Descriptions des besoins techniques

5.1 HTML 5

HTML5 (HyperText Markup Language 5) est la dernière révision majeure d'HTML (format de données conçu pour représenter les pages web).

HTML5 spécifie deux syntaxes d'un modèle abstrait défini en termes de DOM : HTML5 et XHTML5. Le langage comprend également une couche application avec de nombreuses API, ainsi qu'un algorithme afin de pouvoir traiter les documents à la syntaxe non conforme.

Dans le langage courant, HTML5 désigne souvent un ensemble de technologies Web (HTML5, CSS3 et JavaScript) permettant notamment le développement d'applications.



5.2 CSS 3

CSS (Cascading Style Sheets : feuilles de style en cascade) est un langage informatique qui sert à décrire la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C). Introduit au milieu des années 1990, CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années 2000.



5.3 PHP 5

PHP (Hypertext Preprocessor), est un langage de scripts libre principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale, en exécutant les programmes en ligne de commande.

PHP est un langage impératif disposant depuis la version 5 de fonctionnalités de modèle objet complètes. En raison de la richesse de sa bibliothèque, on désigne parfois PHP comme une plate-forme plus qu'un simple langage.



5.4 JavaScript

JavaScript (souvent abrégé JS) est un langage de programmation de scripts principalement utilisé dans les pages web interactives mais aussi côté serveur. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet d'en créer des objets héritiers personnalisés.



5.5 JQuery

JQuery est une bibliothèque JavaScript libre qui porte sur l'interaction entre JavaScript et HTML, et a pour but de simplifier des commandes communes de JavaScript.

JQuery est devenue une référence importante à savoir utiliser : pour preuve, elle est largement répandue sur la toile. Microsoft, Twitter, Mozilla, Google Code, Amazon et Wordpress France, pour ne citer qu'eux, l'utilisent



5.6 MySQL

MySQL est un système de gestion de base de données (SGBD). Il fait partie des logiciels de gestion de base de données les plus utilisés au monde[1], autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server.

MySQL est devenue le système de gestion de base de données open source le plus populaire au monde grâce à sa performance, sa haute fiabilité et sa simplicité d'utilisation.



5.7 WampServer

WampServer (anciennement WAMP5) est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL. Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs au travers d'un tray icon (icône près de l'horloge de Windows).

La grande nouveauté de WampServer 2 réside dans la possibilité d'y installer et d'utiliser n'importe quelle version de PHP, Apache ou MySQL en un clic. Ainsi, chaque développeur peut reproduire fidèlement son serveur de production sur sa machine locale.



5.8 Bootstrap

Bootstrap a été créé en 2011 chez Twitter comme une solution à usage interne, pour répondre aux incohérences de développement au sein de l'équipe d'ingénieurs de Twitter. En gros, aucune structure de code n'avait été déterminée par les ingénieurs chez Twitter au moment du développement de la plateforme.

Bootstrap est une compilation de plusieurs éléments et fonctions web design personnalisables, le tout emballé dans un seul et même outil. Les développeurs qui utilisent Bootstrap pour la création de leur site web choisissent les éléments qu'ils veulent utiliser avec la certitude qu'ils ne seront pas incompatibles entre eux. En fait, c'est comme un puzzle. Sauf que dans ce puzzle, chaque pièce s'imbrique parfaitement dans les autres, quelle qu'elle soit.



5.9 Enterprise Architect

Enterprise Architect est un logiciel de modélisation et de conception UML, édité par la société australienne Sparx Systems. Couvrant, par ses fonctionnalités, l'ensemble des étapes du cycle de conception d'application, il est l'un des logiciels de conception et de modélisation les plus reconnus [1]. Enterprise Architect permet le développement d'applications selon le schéma d'architecture orienté modèle2 ainsi que le schéma d'architecture orientée services3.



Enterprise Architect couvre tous les aspects du cycle de développement d'applications depuis la gestion des exigences, en passant par les phases de conception, la construction, tests et maintenance. Ces aspects sont appuyés par des fonctions de support tels que la traçabilité, la gestion de projet, ou encore le contrôle de version2.

Le produit est destiné aux analystes, développeurs, architectes, urbanistes de toutes structures : de petites et moyennes entreprises aux multinationales, ainsi que les organisations gouvernementales.

6. Architecture des interfaces

Tableau 1 Architecture des interfaces

Interfaces	Description
Interface 1. La gestion des rendez-vous	L'application fournit la possibilité de gérer les rendez-vous par un ensemble des médecins externes de différents hôpitaux et spécialités. L'administrateur et la secrétaire ont la possibilité de modifier et désactiver un rendez-vous selon le planning de l'agenda.
Interface 2 : Gestion des utilisateurs	L'administrateur fait la création, la modification et la désactivation des comptes pour chaque utilisateur
Interface 3. Consulter les rendez-vous	L'application fournit aux médecins la possibilité de consulter leurs rendez-vous.
Interface 4 : Gestion des agendas, services et activités	L'administrateur a le droit de créer, de modifier et de désactiver les agendas, les services et les activités
Interface 5 : Gestion comptes des utilisateurs	Avant d'accéder au site, chaque utilisateur doit saisir ses données personnelles pour y accéder à son compte

Chapitre 3. Organisation du projet

Dans ce chapitre, nous présentons le processus de développement sur lequel nous nous sommes basés pour organiser et planifier notre projet, le tableau des tâches de notre projet, ensuite nous présentons les diagrammes de GANT et de PERT.

1. Processus de Développement

1.1. Cadre de référence

1.1.1. Introduction

Nous nous intéressons en particulier dans cette partie aux procédures systématiques qui permettent d'arriver à notre projet en répondant aux attentes de notre utilisateur final. Pour arriver à cette fin nous nous inspirons du cadre de référence adapté par Adam [2] pour répondre aux besoins d'analyse et de conception d'un système interactif dédié aux systèmes administratifs complexes. En effet nous avons tenté de s'inscrire dans cette méthode et de mettre notre projet sous le filtre de cette approche. Le travail d'Adam dans cette adaptation citée stipule trois dimensions composant ainsi cette méthode : la dimension méthodologie, la dimension représentation et la dimension technologie.

1.1.2. Cadre Méthodologique

Il existe de nombreuses méthodes de conception, regroupées dans un certain nombre de familles (méthodes structurées, en flots de données, entités-relations, orientées objet...). Chaque méthode a ses partisans, et selon le domaine du problème à résoudre, l'une ou l'autre peut s'avérer plus performante. Comment donc définir ce que contient une méthode ?

Ensuite, il convient de mettre en œuvre ces principes : les méthodes définissent donc une démarche précise à suivre pour guider pas à pas le processus de conception. En particulier, elles définissent des étapes de conception et des critères d'acceptabilité permettant de passer d'une étape à la suivante. Les étapes les plus récurrentes dans les méthodologies de génie logiciel sont l'analyse de l'existant, la spécification, la conception, l'implémentation et l'évaluation. Une phase d'intégration est parfois explicitée lorsqu'il s'agit de combiner deux processus de développement relatifs à deux domaines différents.

Plusieurs modèles existent décrivant ainsi le processus de conception du cycle de vie d'un logiciel. Mais la pertinence du modèle trouve sa place en fonction de sa capacité de répondre à des besoins ou des principes particuliers. Le modèle en X pour le développement orienté objet [3] ou le modèle en U [4] qui est adapté pour la prise en compte des facteurs humains dans le développement des interfaces homme-machine sont quelques exemples pour élucider cette réalité.

Pour notre analyse contentons-nous de présenter deux, les modèles de génie logiciel les plus utilisés sont le modèle incrémental et le processus unifié.

Le modèle incrémental. Ce modèle de cycle de vie prend en compte le fait qu'un logiciel peut être construit étape par étape. Le logiciel est spécifié et conçu dans son ensemble. La réalisation se fait par incréments de fonctionnalités. Chaque incrément est intégré à l'ensemble des précédents et à chaque étape le produit est testé exploité et maintenu dans son ensemble. Ce cycle de vie permet de prendre en compte l'analyse de risques et de faire accepter progressivement un logiciel par les utilisateurs plutôt que de faire un changement brutal des habitudes.

Le processus unifié. UP (Unified Process) [5] est une méthode générique de développement de logiciel. Générique signifie qu'il est nécessaire d'adapter UP au contexte du projet, de l'équipe, du domaine et/ou de l'organisation (exemple : R. UP ou X. UP). C'est, entre parenthèses, plus ou moins vrai pour toute méthode, qu'elle se définisse elle-même comme générique ou pas. Il existe donc un certain nombre de méthodes issues d'UP.

Dans notre cas de développement de l'application, il est nécessaire d'inclure les utilisateurs finaux, dans certaines de ces phases (notamment dans les phases d'analyse et dans les phases d'évaluation).

De ce fait la dimension méthodologie répond à quatre critères dont chacun est composé de sous critères :

- Phases concernées : analyse, spécification, conception, implémentation, intégration et/ou évaluation
- Modèles de développement : en incrémental ou selon le processus unifié
- Degrés d'implication de l'utilisateur : essentiel, beaucoup, moyen, peu ou pas d'implication
- Moment d'implication de l'utilisateur : début, milieu et/ou fin de cycle de développement

1.1.3. La dimension de présentation

Derrière l'idée de formalisme de représentation d'une méthode plusieurs solutions ont été développées. Répondant chacune d'elles à un contexte et au niveau des contraintes technologiques,

l'UML « Unified Modeling Language » ou « langage de modélisation unifié » est devenu l'accomplissement de la fusion de précédents langages de modélisation objet : Booch [6], OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard (depuis 1997) défini par l'Object Management Group (OMG). La dernière version diffusée par l'OMG est UML 2.4.1 depuis août 2011. La dimension représentation répond en fin à un critère de formalisme basé sur : L'UML et Extension d'UML.

1.1.4. La dimension de technologie

Pour réaliser notre projet de développement de cette application de gestion des rendez-vous, la dimension technologique est assujettie aux caractéristiques de celle-ci. En effet nous avons identifié les principales technologies œuvrant dans le cadre des systèmes d'information personnalisés :

Approche orientée objet pour la modélisation. Cette approche permettrait aussi de répondre aux critères d'évolutivité et de distributivité du système.

Mode de traitement client-serveur pour répondre au critère multi-application du système.

La dimension technologie est donc associée à trois critères :

Programmation : structurée, orientée base de donnée, à base de composants, orientée objet et/ou orientée agent

Mode de traitement : batch, synchrone, asynchrone et/ou client-serveur

2. Processus 2TUP

Le processus 2TUP [7] (Two Track Unified Process) est un processus unifié. Il gère la complexité technologique en donnant part à la technologie dans son processus de développement et renferme les caractéristiques suivantes :

- S'articule autour de l'architecture
- Propose un cycle de développement en Y
- Détaillé dans l'UML
- Cible des projets de toutes tailles

Le 2TUP propose un cycle de développement qui dissocie les aspects techniques des aspects fonctionnels et propose une étude parallèle deux branches : fonctionnelle (étude de l'application) et la technique (étude de l'implémentation). Illustré sur la figure suivante.

Le processus 2TUP s'articule autour de trois phases :

- Une branche technique
- Une branche fonctionnelle
- Et une branche de conception réalisation.

La figure 1 représente les étapes de développement des trois branches du processus 2TUP.

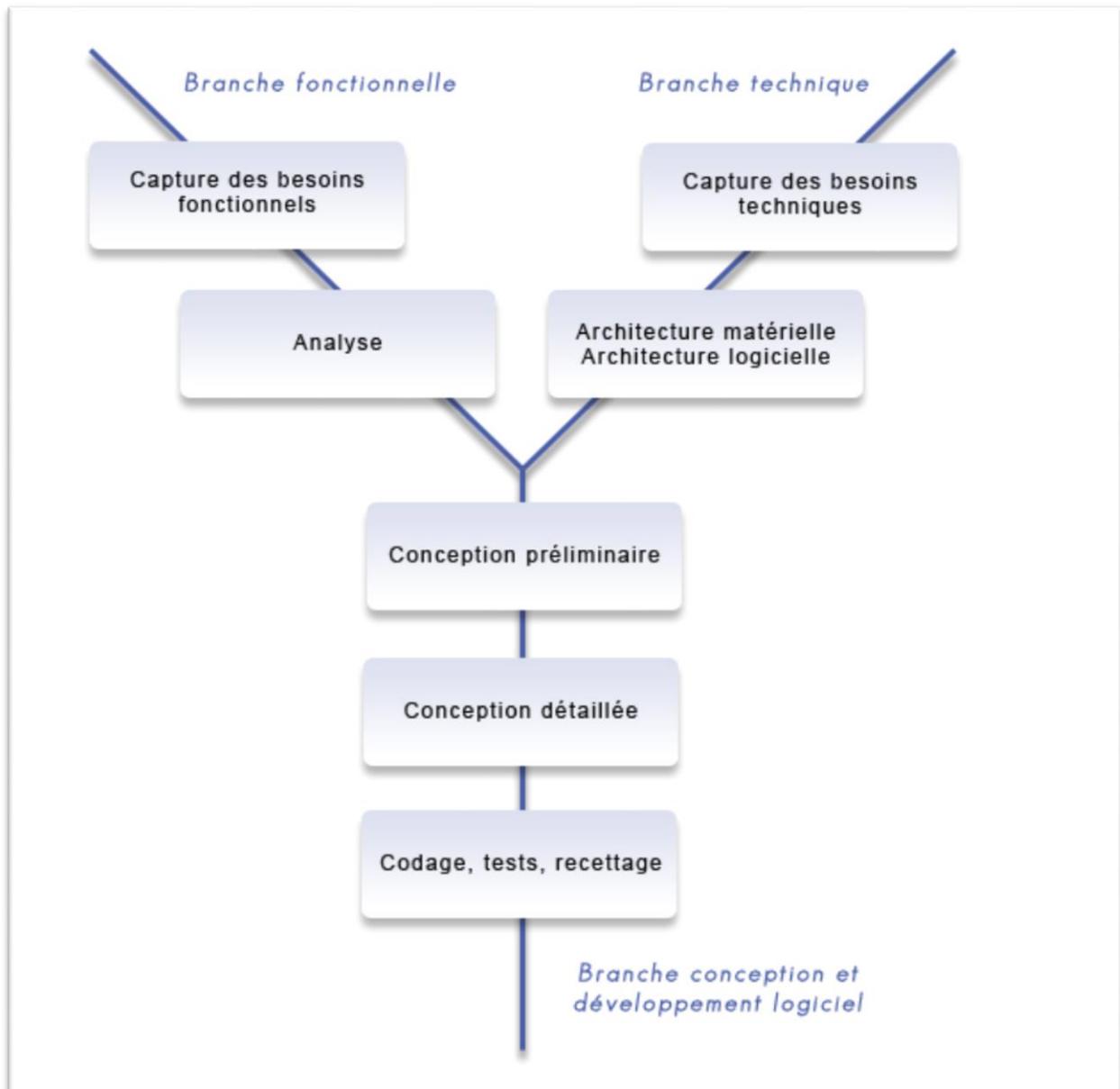


Figure 15. Le processus de développement en Y [7]

2.1. Branche fonctionnelle

Les principales étapes de la branche fonctionnelle se présentent comme suit :

L'étape capture des besoins fonctionnels produit le modèle des besoins focalisé sur le métier des utilisateurs. Elle qualifie, au plus tôt le risque de produire un système inadapté aux utilisateurs. Cette phase a pour objectif de définir :

- La frontière fonctionnelle entre le système considéré comme une boîte noire et son environnement, c'est le niveau contexte.
- Les activités attendues des différents utilisateurs par rapport au système toujours envisagé comme une boîte noire, c'est le niveau cas d'utilisation.
- L'étape d'analyse consiste à étudier précisément les spécifications fonctionnelles de manière à obtenir une idée de ce que va réaliser le système en terme de métier.

2.2. Branche technique

Les principales étapes de la branche technique se présentent comme suit :

- L'étape capture des besoins techniques recense toutes les contraintes sur les choix de dimensionnement et la conception du système. Les outils et le matériel sélectionnés ainsi que la

prise en compte des contraintes d'intégration avec l'existant (pré requis d'architecture technique). Cette étape permet de définir le modèle d'analyse technique. Le rôle de ce dernier est d'établir les couches logicielles et y spécifie les activités techniques attendues.

- L'étape conception générique définit ensuite les composants nécessaires à la construction de l'architecture technique. Cette conception est complètement indépendante des aspects fonctionnels. Elle permet de générer le modèle de conception technique qui définit les Framework. Ces derniers, délivrant les services techniques, assurent la réponse aux exigences opérationnelles du système.

2.3. Branche conception – réalisation

Les principales étapes de cette branche se présentent comme suit :

- L'étape conception préliminaire est une étape délicate, car elle intègre le modèle d'analyse fonctionnelle dans l'architecture technique de manière à tracer la cartographie des composants du système à développer. Cette étape permet de produire le modèle de conception système. Ce dernier organise le système en composants, délivrant les services techniques et fonctionnels. Ce modèle regroupe les informations des branches technique et fonctionnelle.
- L'étape conception détaillée permet d'étudier comment réaliser chaque composant. Cette étape produit le modèle de conception des composants. Ce modèle fournit l'image prête à fabriquer du système complet. C'est dans l'étape de codage que s'effectue la production des composants et les tests des unités de code au fur et à mesure de leur réalisation.
- L'étape de recette consiste à valider les fonctionnalités du système développé.

3. Planification Opérationnelle

3.1. Description

La planification est l'organisation dans le temps de la réalisation d'objectifs :

- Dans un domaine précis ;
- Avec différents moyens mis en œuvre ;
- Et sur une durée (et des étapes) précise(s)

La caractéristique principale de la planification est la dimension temps. Dans le cadre de notre projet, la dimension du temps est primordiale. Pour gérer à la fois la double facette de notre projet de qui consiste à la fois la planification des tâches relatives au projet et des tâches qui découlent génie logiciel.

Les contraintes de la mise en œuvre du projet stipule de notre part de prendre en considération le volet comportementale humain. L'adaptation face aux stress de l'environnement, la pression de la durée, les difficultés de communication que demande sont autant de variable conditionnelle dans la planification et le pilotage du projet.

La planification opérationnelle place le plan au centre de la conduite de projet. Il rassemble les informations et permet de piloter les différentes dimensions du projet (délais, coûts, ressources, communication, risques...). Il devient le centre de communication et le support à la prise de décision.

Pour assurer cette planification nous avons opté pour le choix de l'outil diagramme de Gantt. Très utiliser en ordonnancement et gestion de projet il permet de visualiser dans le temps les diverses tâches liées composant un projet .Il permet de représenter graphiquement l'avancement du projet. Le diagramme de Gantt, couramment utilisé en gestion de projet, répond à deux objectifs : planifier de façon optimale ainsi que communiquer sur le planning établi et les choix qu'il impose. Le diagramme permet de visualiser :

- Les différentes tâches à envisager
- La date de début et la date de fin de chaque tâche
- La durée escomptée de chaque tâche
- Le chevauchement éventuel des tâches, et la durée de ce chevauchement
- La date de début et la date de fin du projet dans son ensemble.
- La réalisation d'un tel planning nécessite la mise en œuvre de techniques de planification :
- Les tâches doivent être identifiées,
- Les tâches doivent être quantifiées en termes de délais, de charges ou de ressources,
- La logique de l'ensemble des tâches doit être analysée.

3.2. Tableau des tâches

Tableau 3 Tableau des tâches

Tâche	Désignation	Prédécesseur	Durée optimiste	Durée normal	Durée pessimiste	Durée moyenne
Besoins fonctionnelles						
A	Questionnaire + Enquête	-	3	3	4	3,17
B	Analyse des besoins	A	1	1	2	1,17
Organisation du projet						
C	Choix du processus de développement	-	1	1	1	1
D	Planification opérationnelle	C	2	2	2	2
Etude Préalable						
E	Capture des besoins fonctionnels	B, D	2	3	3	2,83
F	Capture des besoins techniques	E	1	1	1	1
G	Capture des besoins graphiques	F	1	1	1	1
H	Capture des besoins ergonomiques	G	1	1	1	1
I	Architecture matérielle et logicielle	F	1	1	1	1
J	Définition des interfaces	I	1	1	1	1
Modélisation						
K	Déterminer la vue fonctionnelle (diagramme des d'utilisations)	J	2	2	2	2
L	Déterminer la vue dynamique (diagramme des séquences)	K	2	2	3	2,17
M	Déterminer la vue logique (diagramme des classes)	K	3	3	4	3,17
N	Elaborer le diagramme de collaboration BPM	L	3	4	4	3,83
O	Elaborer le modèle de Conception de Données(MCD)	M	1	1	1	1
P	Elaborer le modèle Logique de Données(MLD)	O	1	1	1	1
Réalisation technique						
Q	Création de la base de données	P	1	1	1	1
R	Maquettage	P, H	1	1	1	1
Mise en œuvre						
S	Codage	R	21	22	23	22
T	Tests	S	1	1	1	1
U	Mise en place	T	1	1	1	1

NB : La valeur de la durée moyenne Dpk est calculée par la loi Bêta [8] :

$$Dpk = \frac{D. opt + 4D. nor + D. pess}{6}$$

3.3. Diagramme de GANTT

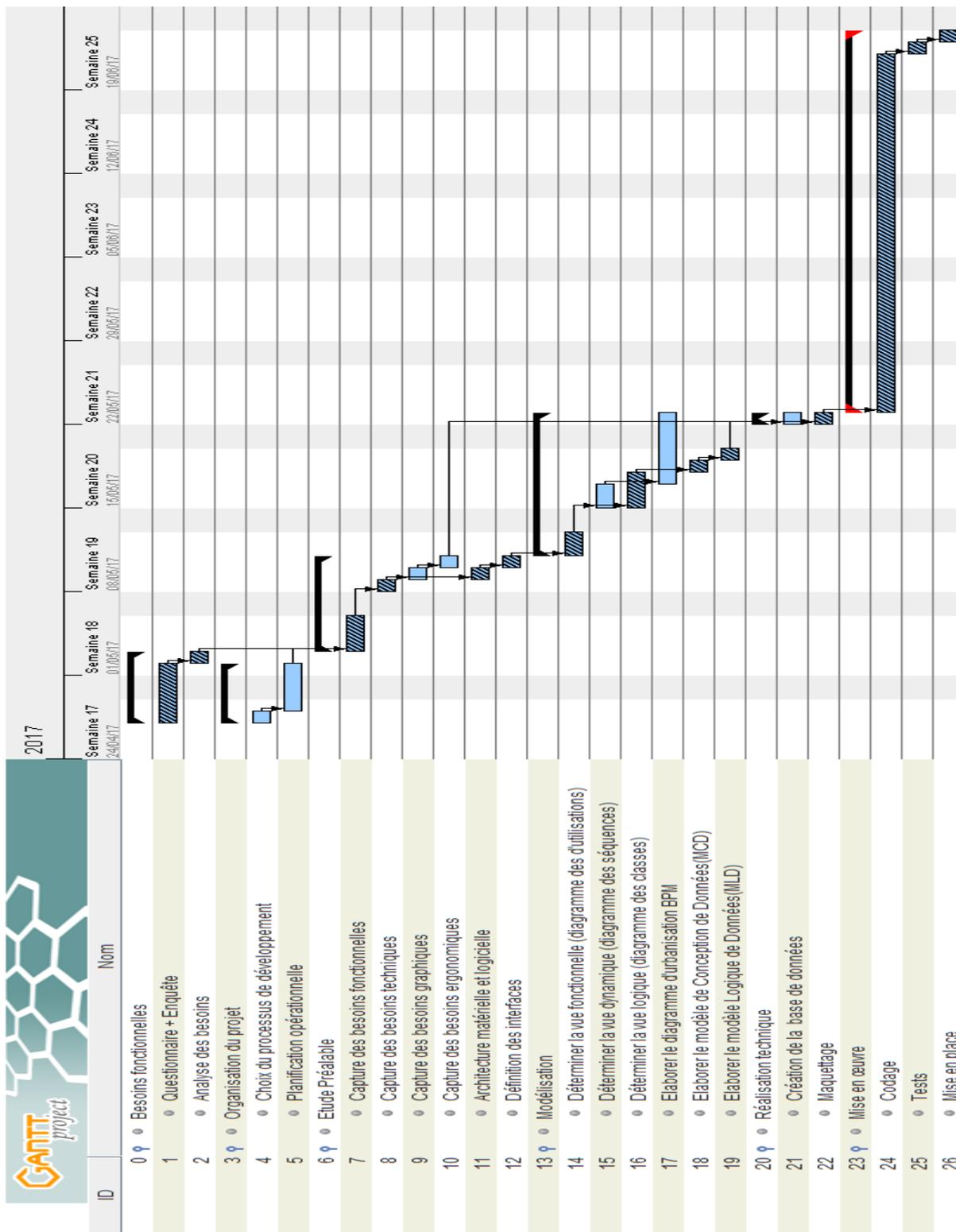


Figure 16 Diagramme de Gantt

Chapitre 4. Modélisation

Le chapitre « Modélisation » fera l'objet du quatrième chapitre qui présente la conception de notre système d'information que nous avons modélisé en UML, et les différentes parties du projet, à savoir : les outils, les maquettes, et la base de données.

1. Modélisation UML

En informatique UML (Unified Modeling Language), ou Langage de modélisation unifié, est un langage de modélisation graphique à base de pictogrammes. Il est utilisé en logiciel, et en conception orientée objet. UML est couramment utilisé dans les projets logiciels.

UML est l'accomplissement de la fusion de précédents langages de modélisation objet : Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard défini par l'Object Management Group (OMG). La dernière version diffusée par l'OMG est UML 2.4.1 depuis août 2011.

Le méta modèle UML fournit une panoplie d'outils permettant de représenter l'ensemble des éléments du monde objet (classes, objets, ...) ainsi que les liens qui les relie.

Toutefois, étant donné qu'une seule représentation est trop subjective, UML fournit un moyen astucieux permettant de représenter diverses projections d'une même représentation grâce aux vues.

Une vue est constituée d'un ou plusieurs diagrammes. On distingue deux types de vues :

Les vues statiques, c'est-à-dire représentant le système physiquement

- Diagrammes d'objets
- Diagrammes de classes
- Diagrammes de cas d'utilisation
- Diagrammes de composants
- Diagrammes de déploiement

Les vues dynamiques, montrant le fonctionnement du système

- Diagrammes de séquence
- Diagrammes de collaboration
- Diagrammes d'états-transitions
- Diagrammes d'activités

UML aide à toutes les étapes de conception du projet

Avantages

- Descriptions graphiques
- Vues différentes à des étapes différentes
- Recoupement des descriptions

Incohérences

- Incomplétudes mises en évidence
- Adaptation facile aux méthodes
- Projet: un bon outil de démarrage du projet
- S'append par la pratique

1.1 Vue Fonctionnelles (diagrammes des cas d'utilisation)

Le modèle de cas d'utilisation permet de spécifier les fonctions attendues d'un système.

Un diagramme de cas d'utilisation montre le système offrant des services à son environnement, l'environnement est représenté par des acteurs.

Ce diagramme décrit précisément les besoins d'utilisateur final et spécifient le comportement attendu par l'application à développer.

L'étude des cas d'utilisation a pour objectif de déterminer ce que chaque utilisateur attend du logiciel. La détermination des besoins est basée sur la représentation de l'interaction entre l'acteur et le système.

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié.

Un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données.

Les acteurs humains de l'application de gestion des rendez-vous sont les suivants :

Tableau 4. Les acteurs utilisant l'application

Acteurs	Fonctions
Administrateur	Il a le contrôle total des fonctions de la plate-forme ; Il gère toutes les inscriptions. Il accède à tous les espaces des utilisateurs et assure le bon déroulement de la plate-forme. Il assure à tous les utilisateurs l'accès aux fonctions que leur permet leur statut.
Médecin et Infirmier major	Ce sont les visiteurs du site pour but prendre un rendez-vous, consulter les agendas et les Activités disponibles
Secrétaire	Fait partie des visiteurs du site pour le but de consulter tous les rendez-vous prises.

L'ensemble des acteurs est représenté graphiquement sur la figure autour d'un rectangle figurant le système à l'étude. La représentation graphique standard de l'acteur en UML est l'icône appelée stick man, avec le nom de l'acteur sous le dessin. On peut également figurer un acteur sous la forme rectangulaire d'une classe, avec le mot-clé <<actor>>. Une bonne recommandation consiste à faire prévaloir l'utilisation de la forme graphique du stick man pour les acteurs humains et une représentation rectangulaire pour les systèmes connectés.

La figure 18 représente le diagramme des cas d'utilisation du médecin. Parmi les fonctionnalités permises au médecin, on trouve la gestion des patients, la gestion des rendez-vous et la signalisation des alertes. Tous ses fonctionnalités sont précédés par l'authentification.

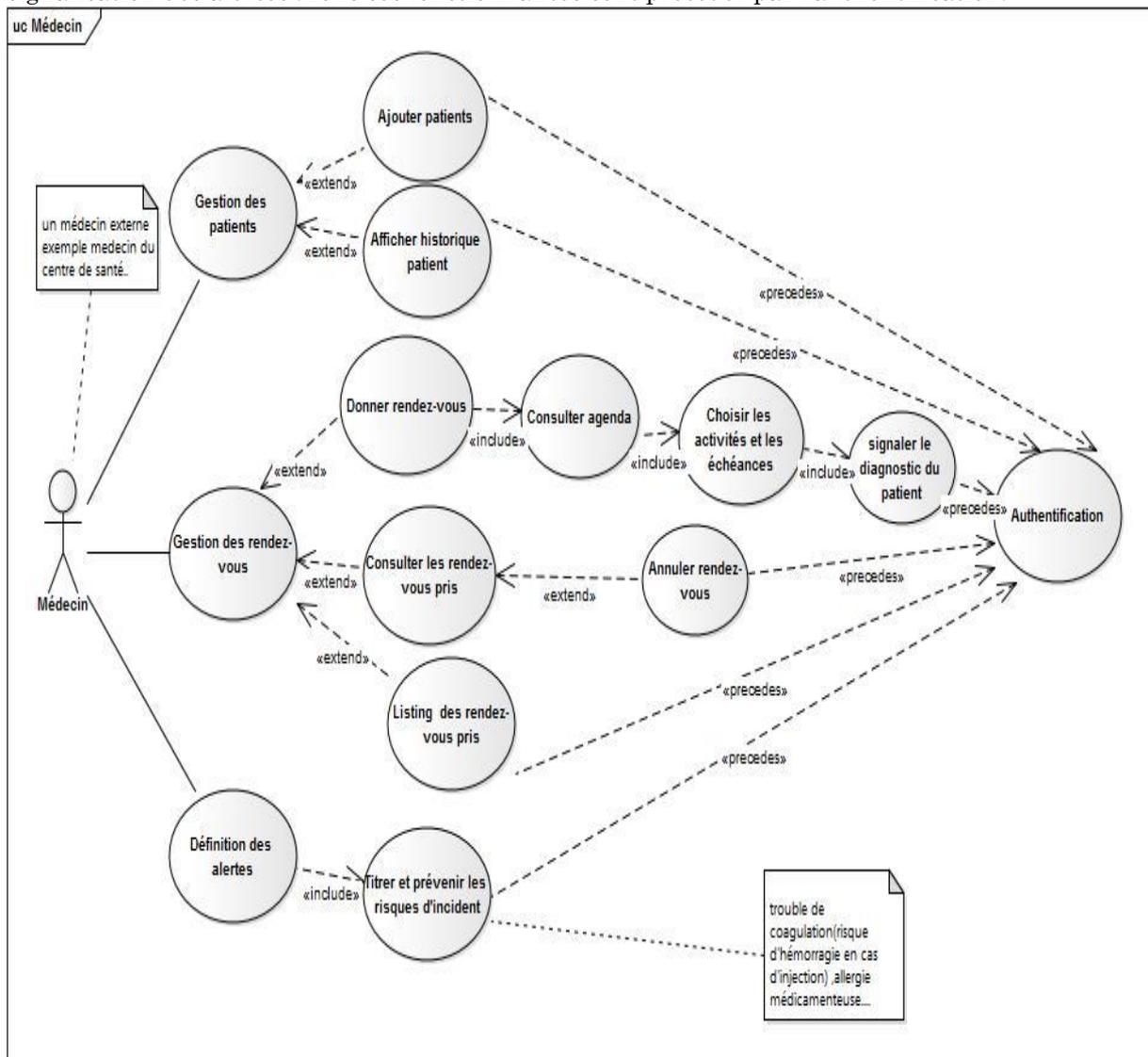


Figure18 Diagramme des cas d'utilisations du médecin

La figure 19 présente les fonctionnalités permises à la secrétaire. Après l'authentification, secrétaire à le doit de gérer des patients, les rendez-vous et les agendas. La secrétaire à la possibilité de bloquer un agenda par activités ou par période.

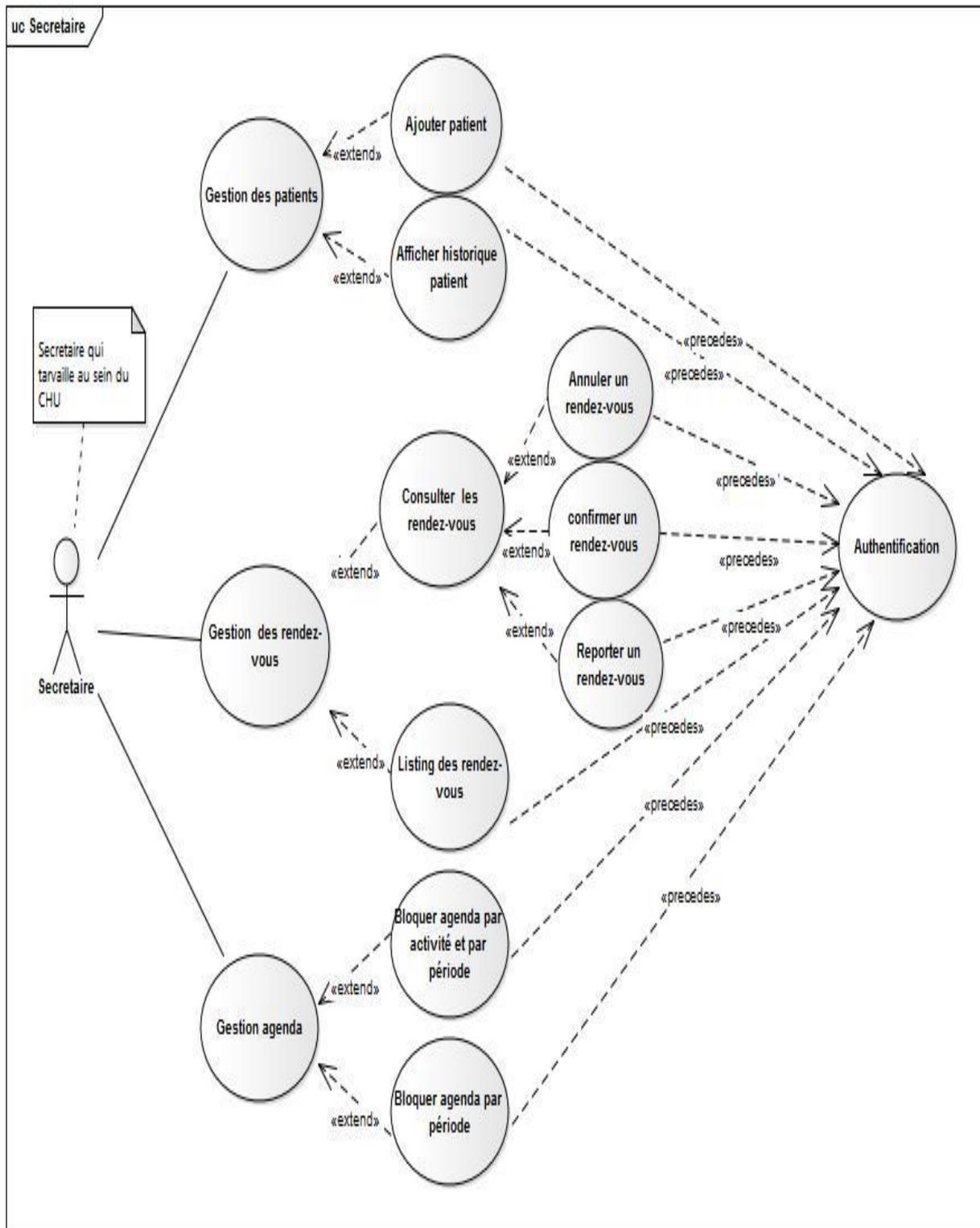


Figure 19. Diagramme des cas d'utilisations de la secrétaire

La figure 20 présente les fonctionnalités permises à l'administrateur, en trouve la gestion des utilisateurs, la gestion des services, la gestion des agendas, la gestion des activités, et le droit de lister tous les rendez-vous .Tous ses fonctionnalités sont précédé par l'authentification.

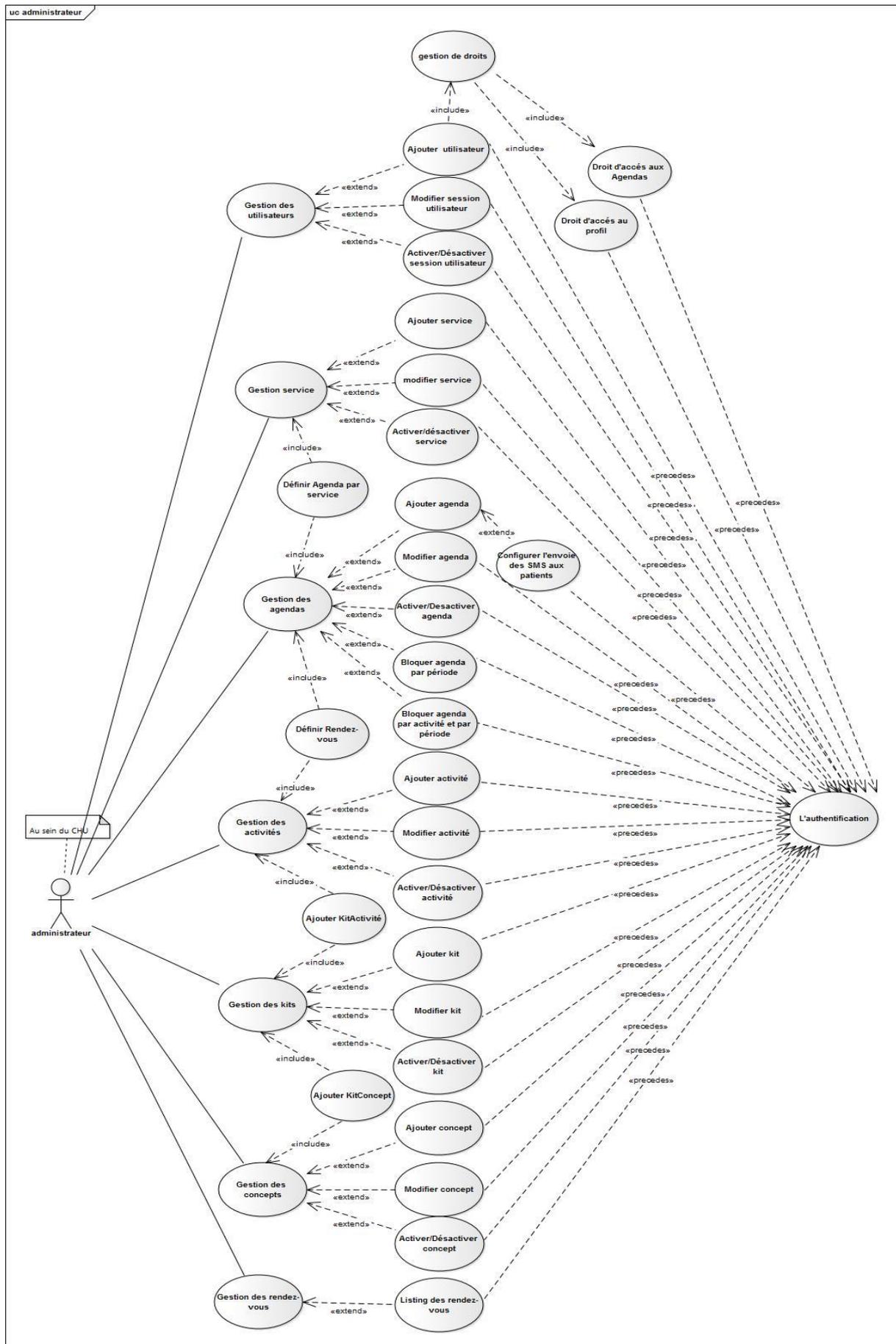


Figure 20 Diagramme des cas d'utilisations de l'administrateur

La figure 21 présente les fonctionnalités permises à l'infirmier major, on trouve la gestion des patients et la gestion des rendez-vous. Tous ses fonctionnalités sont précédées par l'authentification.

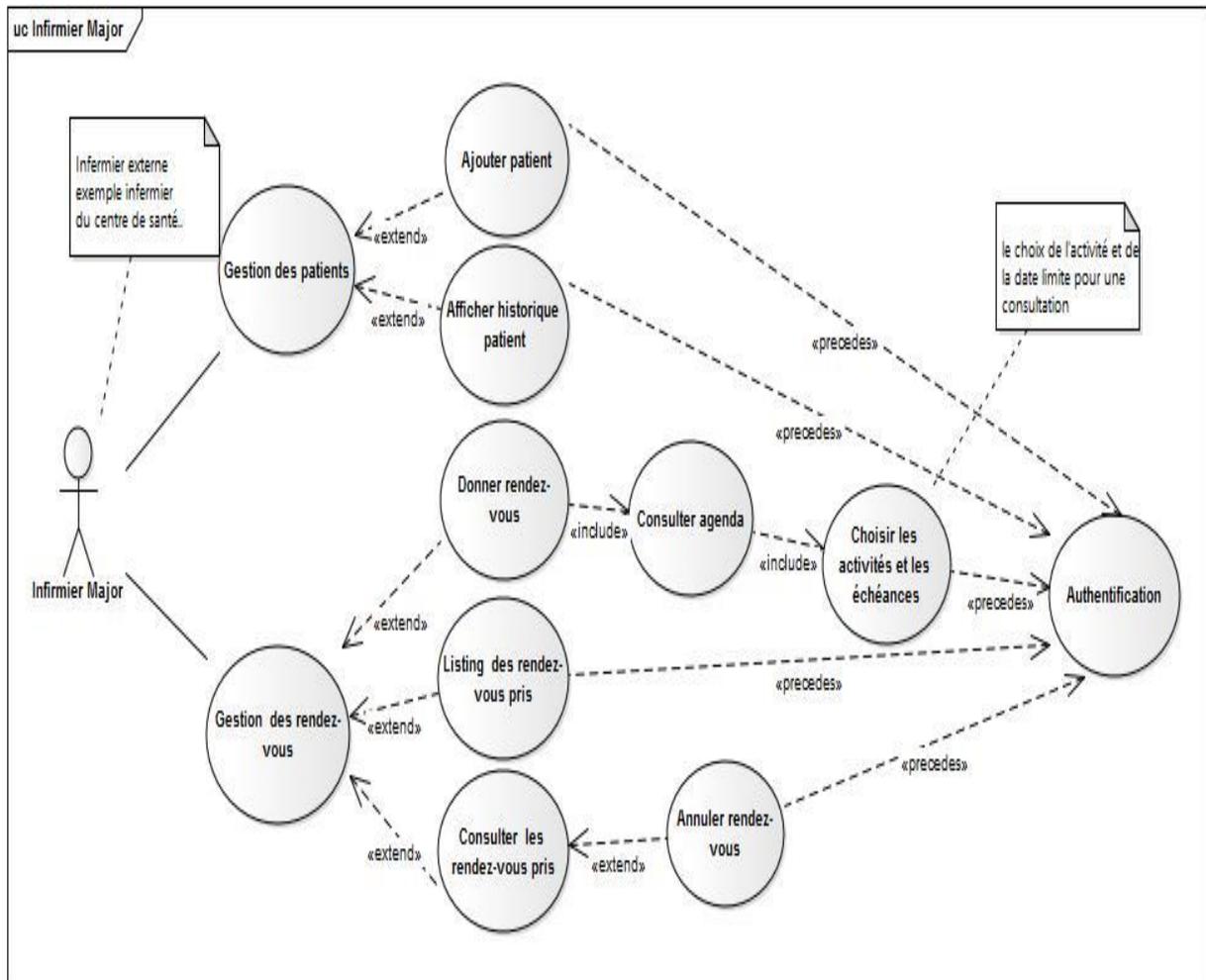


Figure21 Diagramme des cas d'utilisations de l'infirmier major

1.2 Vue Logique (Diagramme de classes)

Le diagramme de classe est un diagramme dans lequel on fait figurer les éléments suivants :

- Les classes conceptuelles ou les objets du domaine ;
- Les associations entre classes conceptuelles
- Les attributs des classes conceptuelles.
- Les associations nécessaires pour prendre en compte les relations qu'il est fondamental de mémoriser ;
- Les attributs nécessaires pour répondre aux besoins d'information.

L'ensemble des concepts fondamentaux est représenté graphiquement sur la figure. La représentation graphique standard des associations en UML est sous forme des flèches, chaque type des flèches à sa signification. On peut également figurer les classes sous la forme rectangulaire, avec le mot de chaque classe et ses attributs.

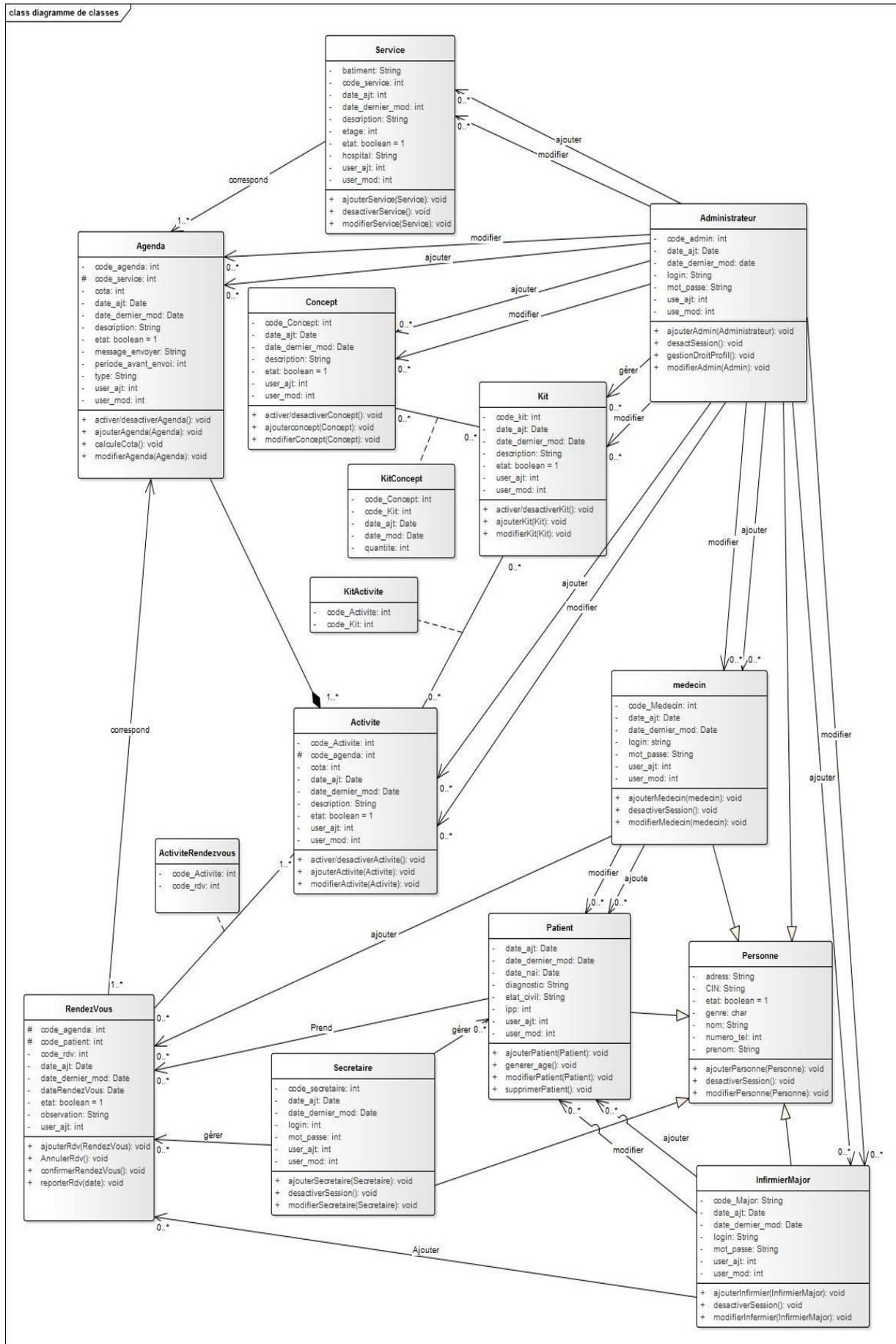


Figure 21 Diagramme des classes

1.3 Vue Dynamique

1.3.1 Diagramme de séquence

Les cas d'utilisation décrivent les interactions des acteurs avec l'application que nous voulons spécifier et concevoir. Lors de ces interactions, les acteurs produisent des messages qui affectent le système informatique et appellent généralement une réponse de celui-ci. Nous allons isoler ces messages et les représenter graphiquement sur des diagrammes de séquence UML.

Le diagramme de séquence montre non seulement les acteurs externes qui interagissent directement avec le système, mais également ce système et les événements système déclenchés par les acteurs. L'ordre chronologique se déroule vers le bas et l'ordre des messages doit suivre la séquence décrite dans le cas d'utilisation.

Pour les diagrammes de séquence, l'administrateur a plusieurs droits, à savoir :

Le droit de désactiver les agendas (voir la figure 22). Ce diagramme présente la démarche suivie pour désactiver un agenda.

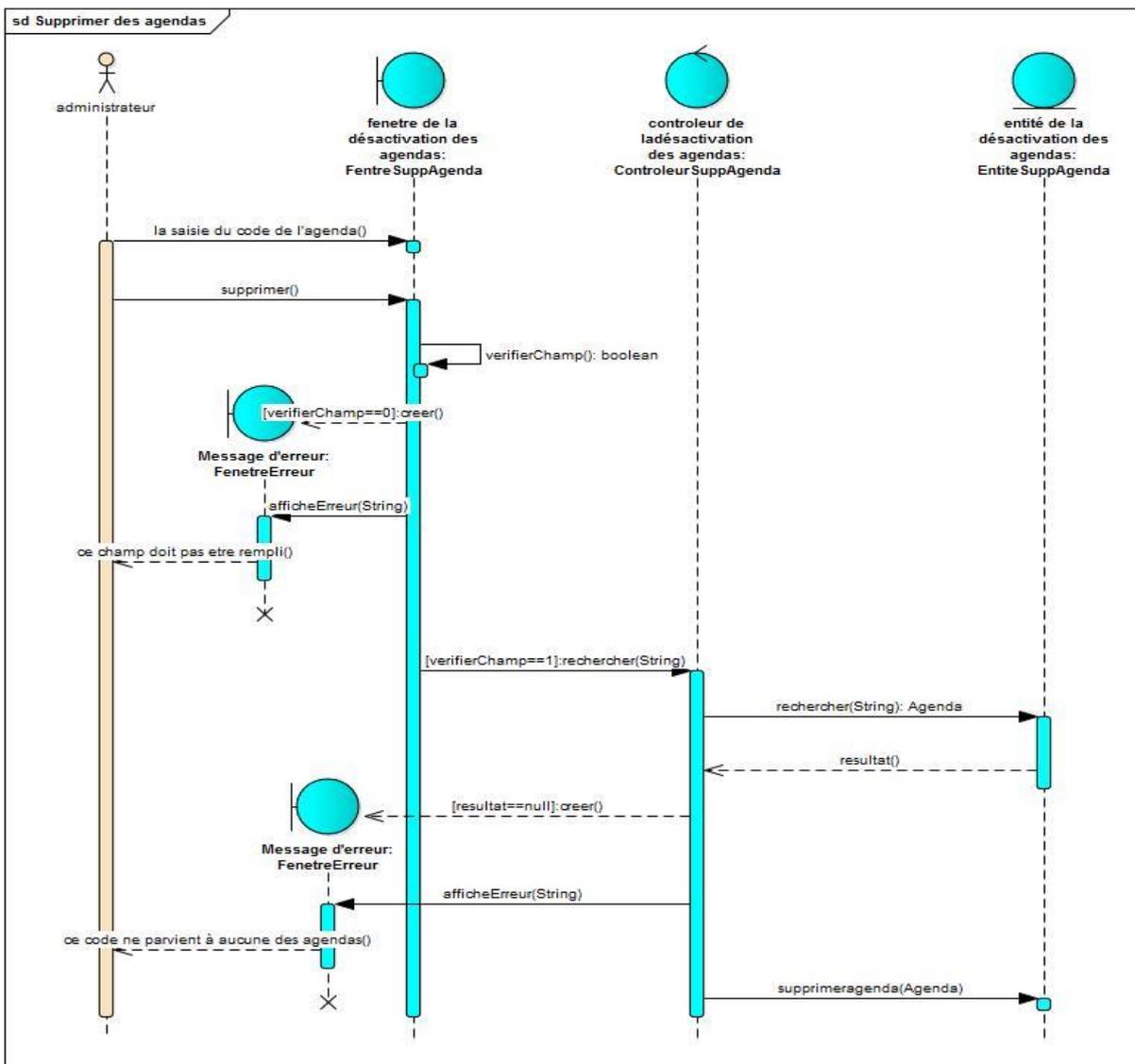


Figure 22. Diagramme de séquence pour désactiver un agenda

L'administrateur a le droit d'ajouter des activités (voir la figure 23). Ce diagramme présente la démarche suivie pour cet ajout.

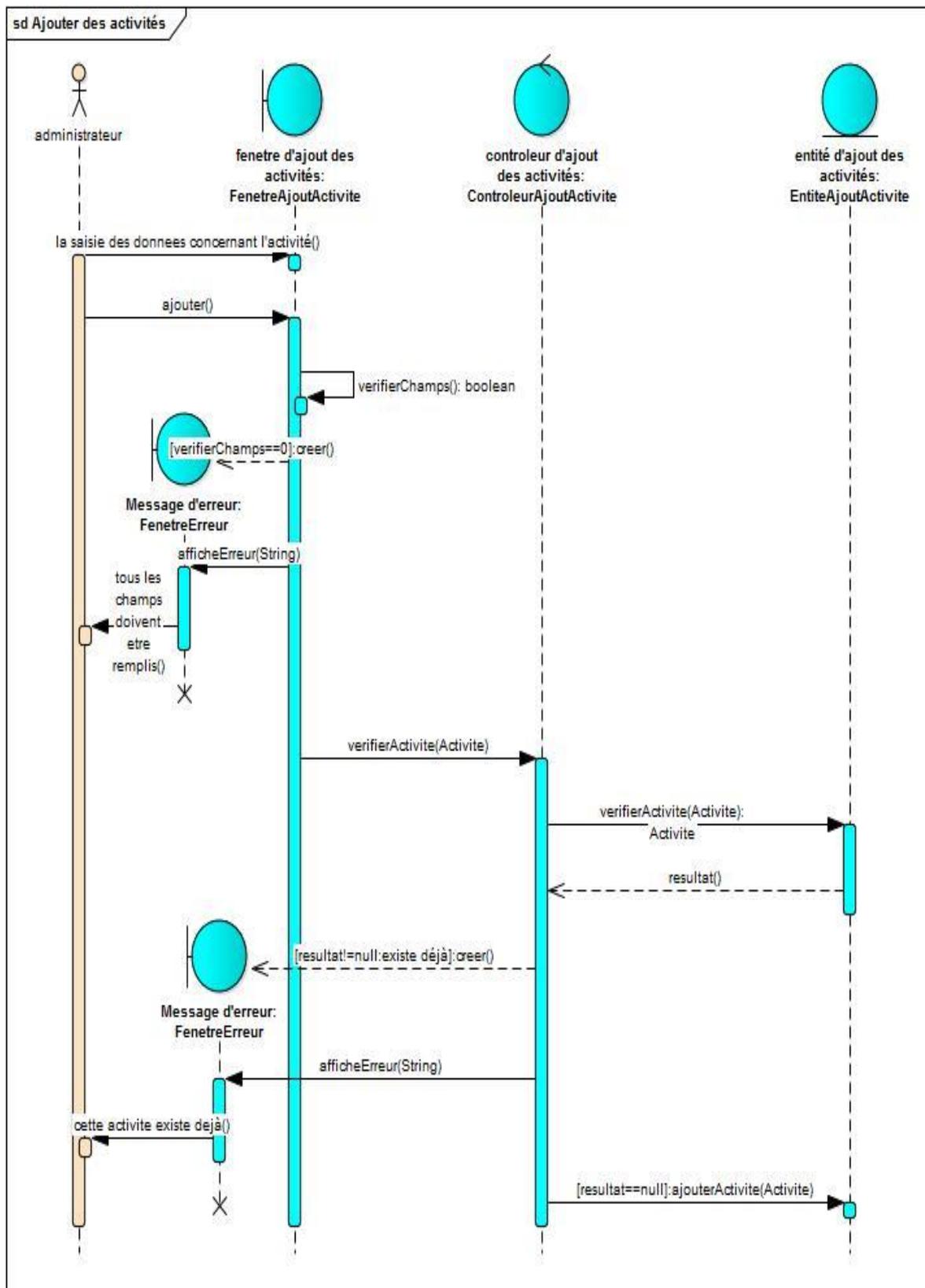


Figure 23 Diagramme de séquence pour ajouter des activités

L'administrateur a le droit d'ajouter des agendas (voir la figure 24). Ce diagramme présente la démarche suivie pour cet ajout.

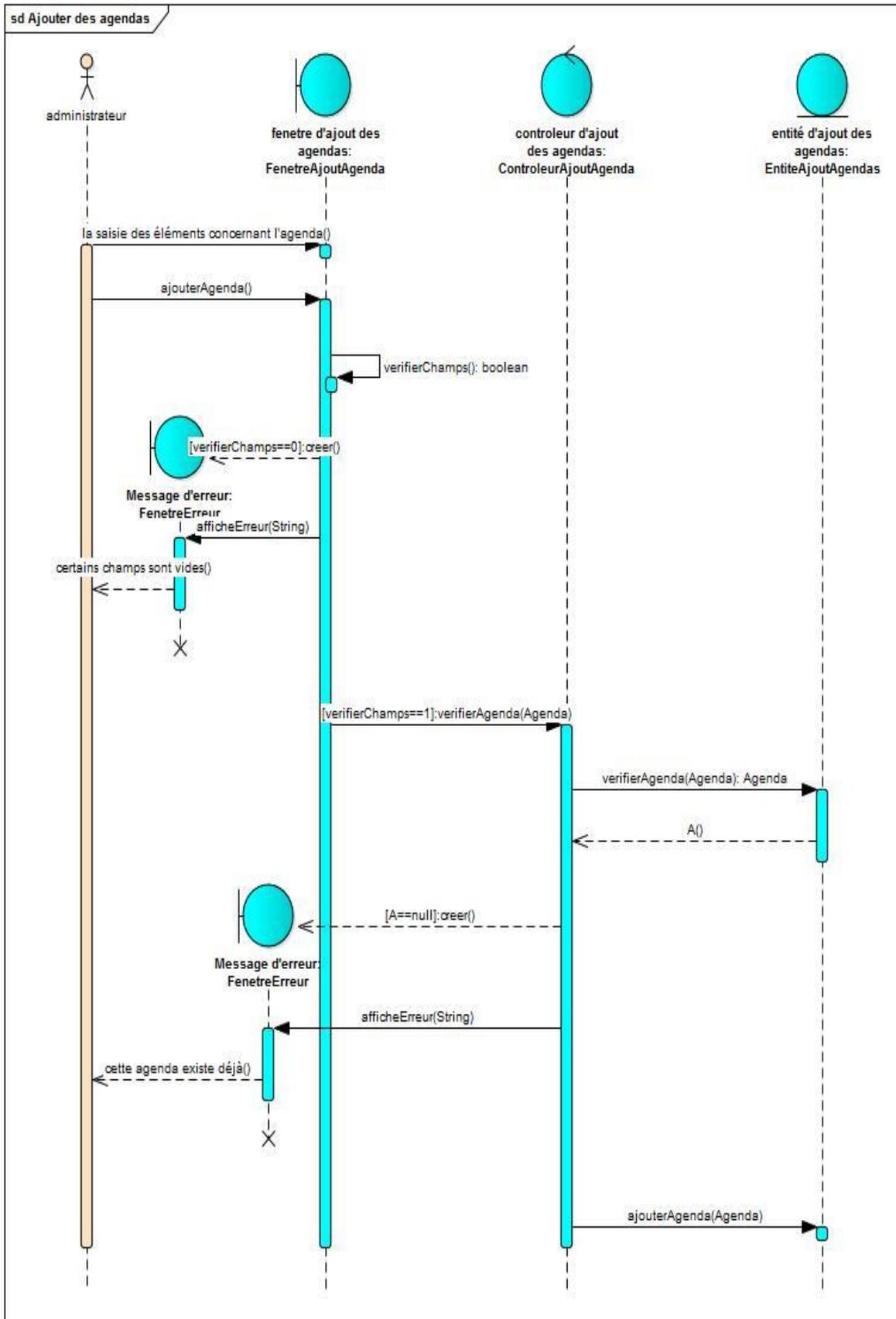


Figure 24 Diagramme de séquence pour ajouter des agendas

Dans la figure 25, l'administrateur a le droit d'ajouter des utilisateurs. Ce diagramme présente la démarche suivie pour cet ajout.

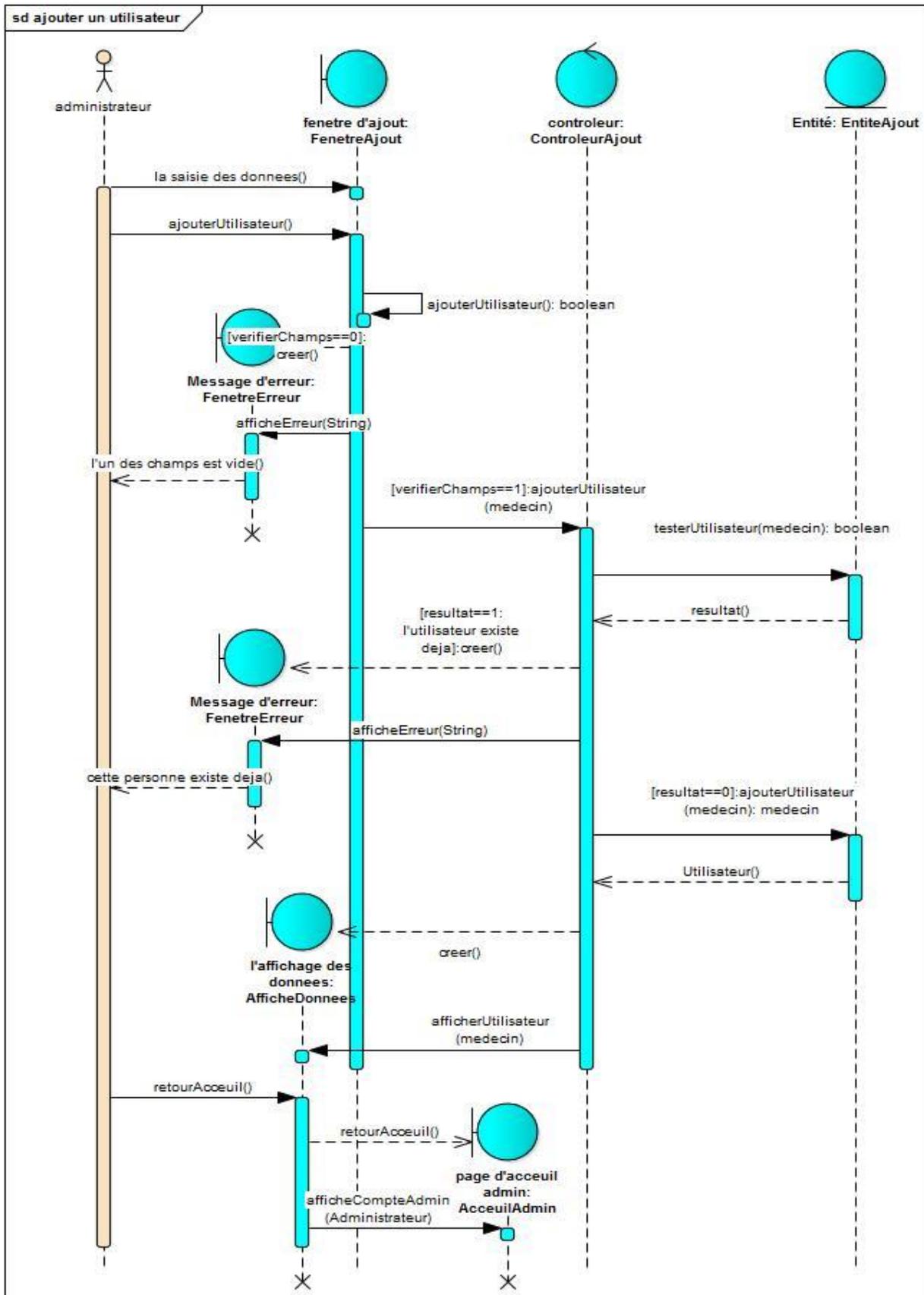


Figure 25 Diagramme de séquence pour ajouter un utilisateur

L'administrateur a le droit de désactiver des activités (voir la figure 26). Ce diagramme présente la démarche à suivre.

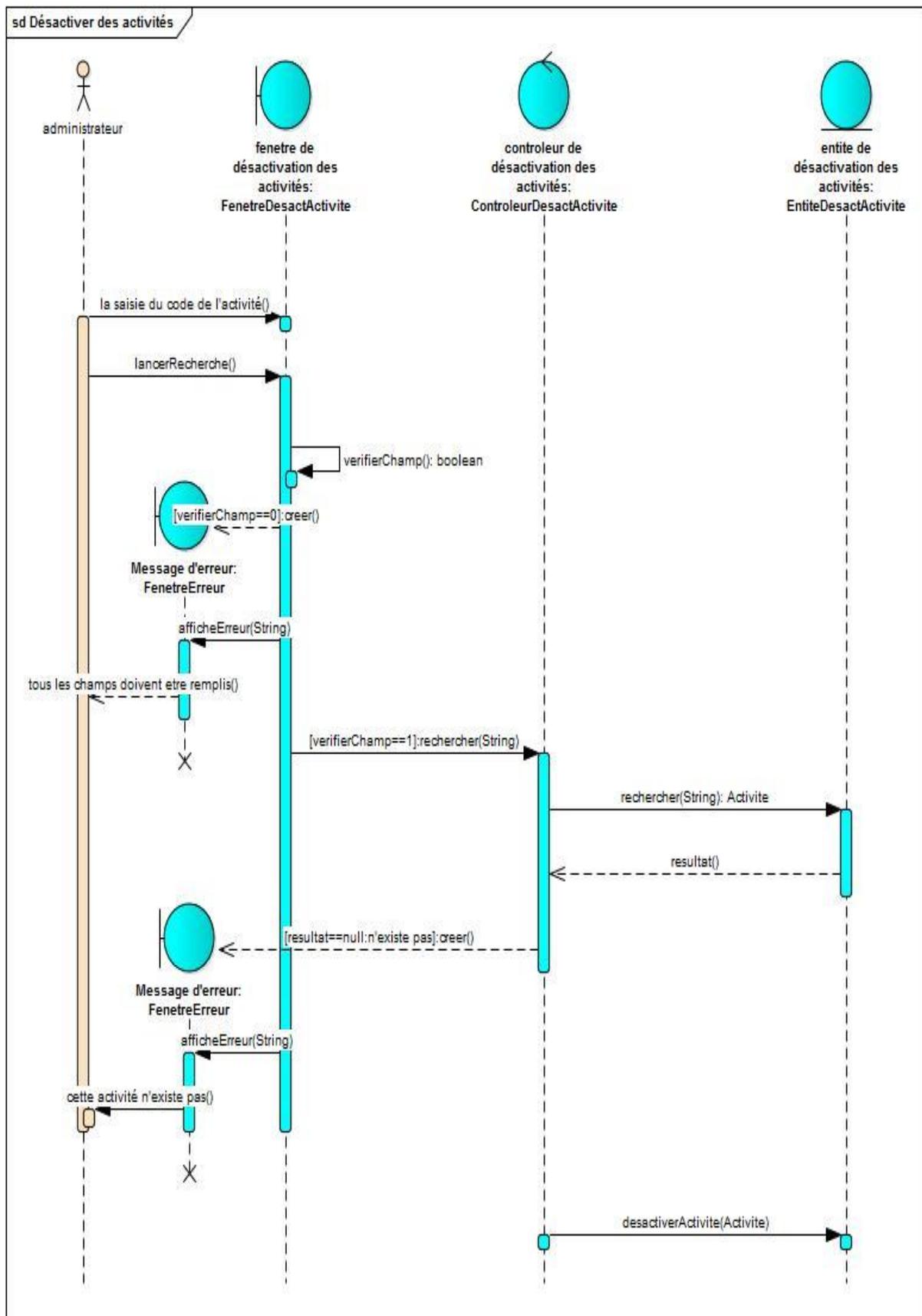


Figure 26 Diagramme de séquence pour désactiver une activité

L'administrateur a le droit de désactiver des comptes d'utilisateurs (voir la figure 27). Ce diagramme présente la démarche à suivre.

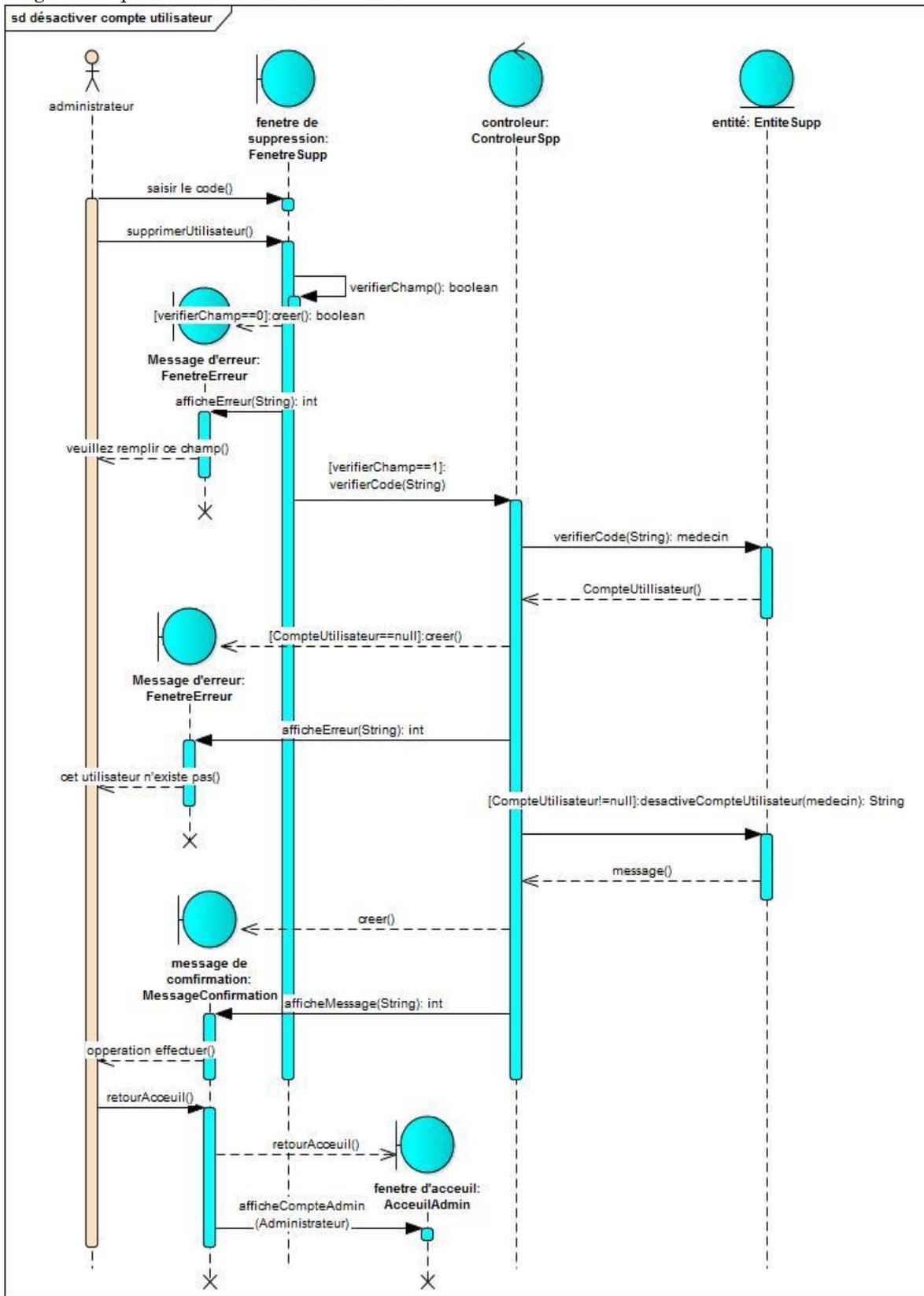


Figure 27 Diagramme de séquence pour désactiver un compte Utilisateur

L'administrateur a le droit de bloquer des activités (voir la figure 28). Ce diagramme présente la démarche suivie.

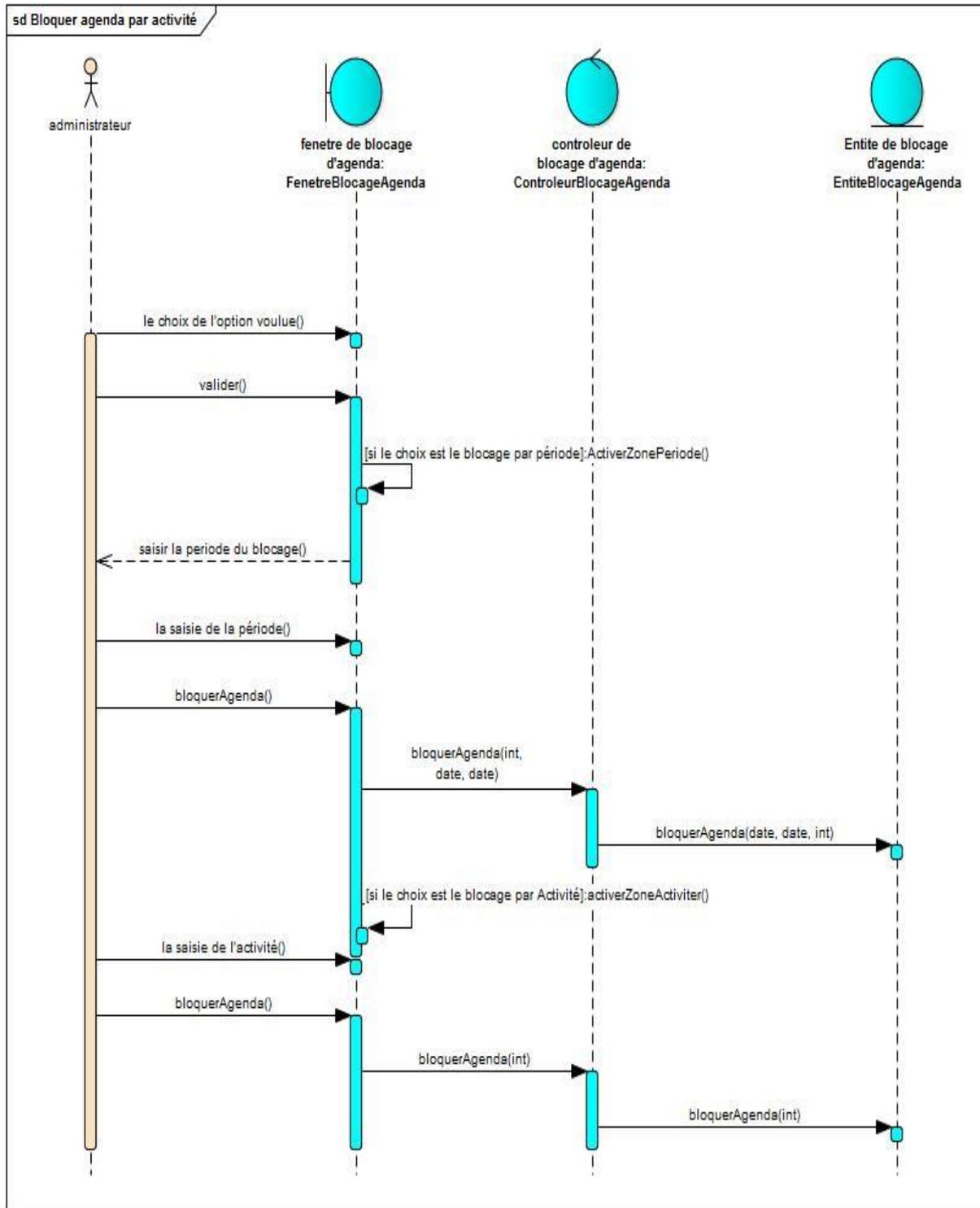


Figure 28 Diagramme de séquence pour bloquer un agenda par activité

L'administrateur a le droit de gérer les activités (voir la figure 29). Ce diagramme présente la démarche suivie.

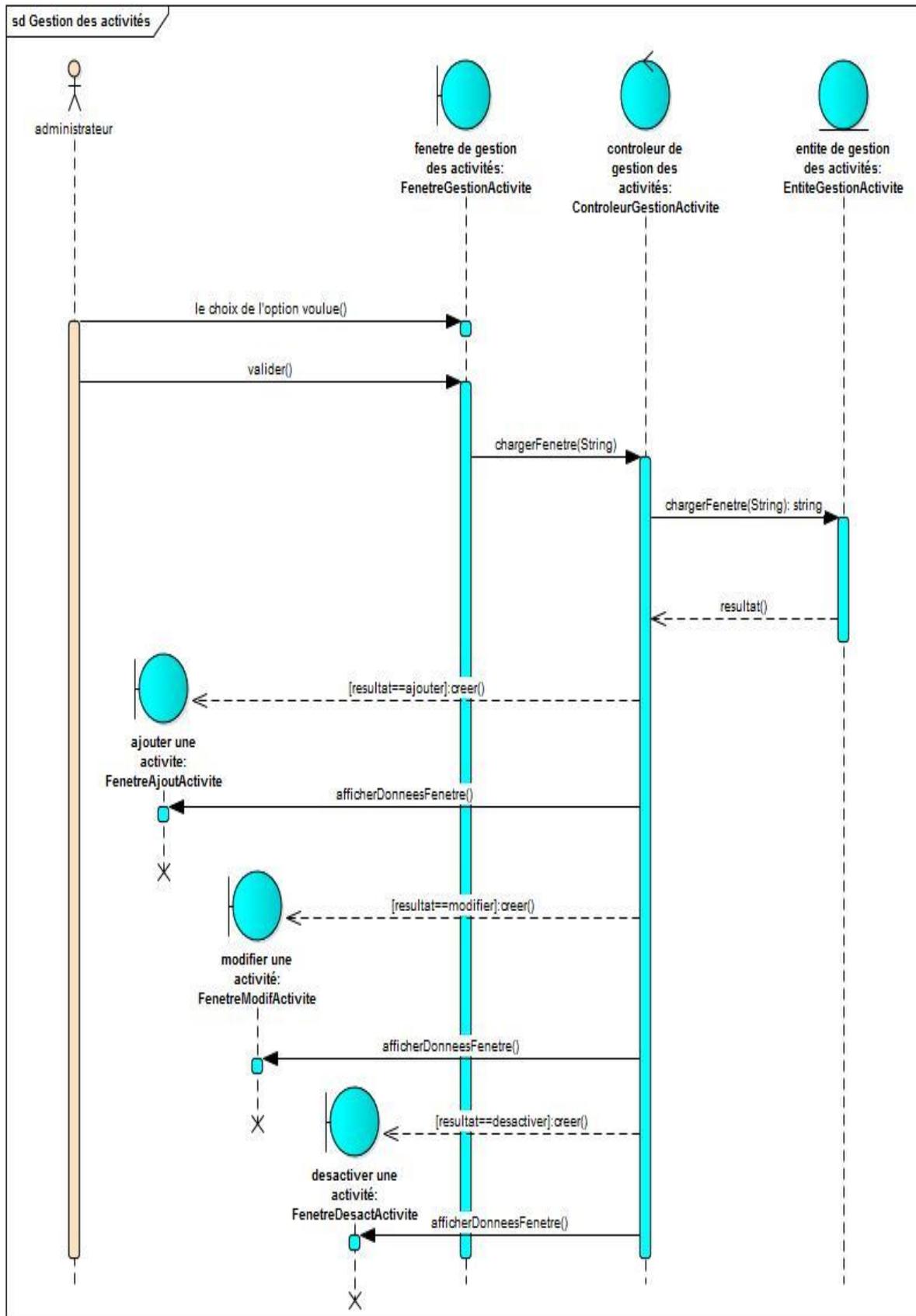


Figure 29 Diagramme de séquence pour la gestion d'une activité

L'administrateur a le droit de gérer les agendas (voir la figure 30). Ce diagramme présente la démarche suivie.

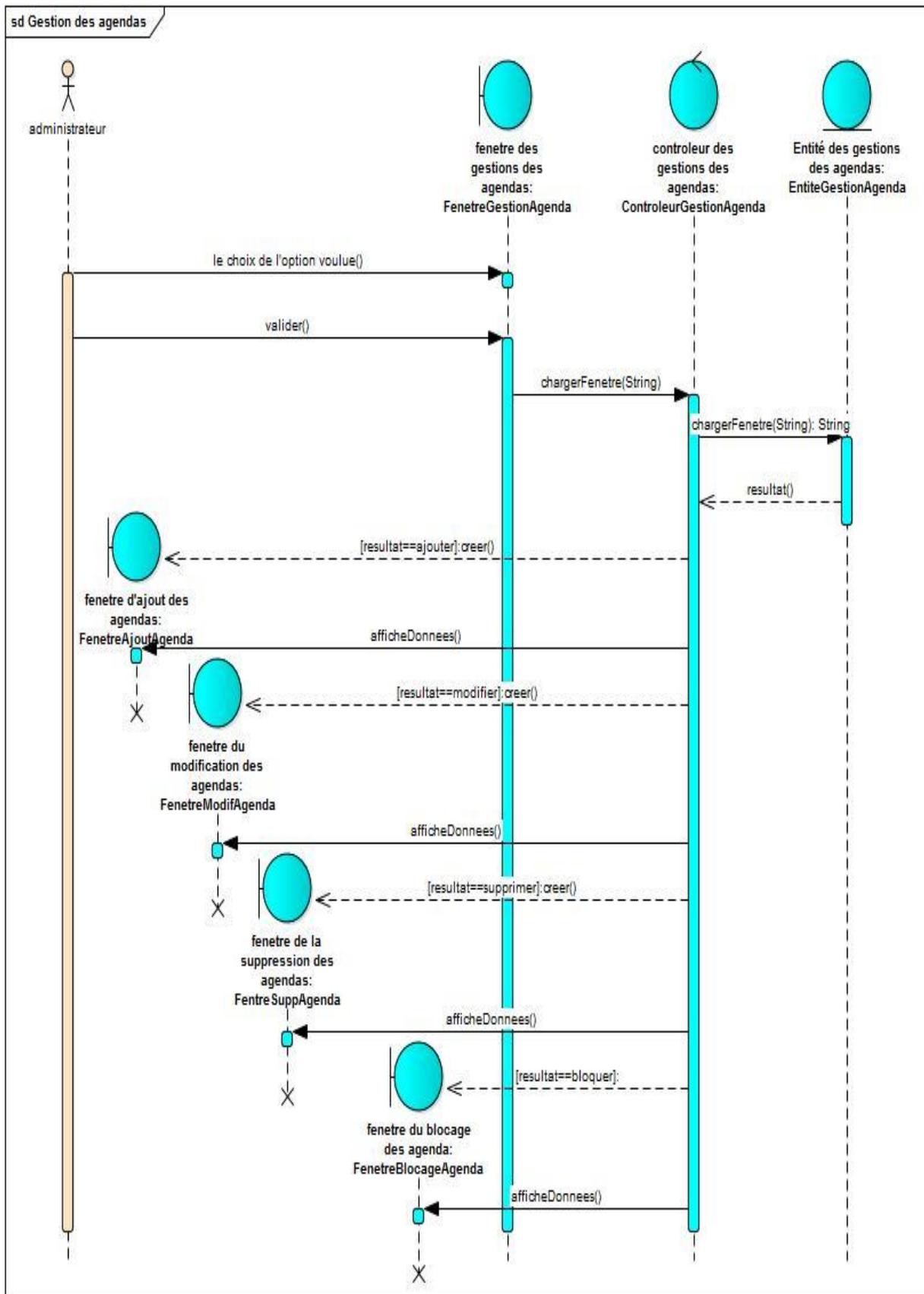


Figure 30 Diagramme de séquence pour la gestion des Agendas

L'administrateur a le droit de gérer les utilisateurs (voir la figure 31). Ce diagramme présente la démarche suivie.

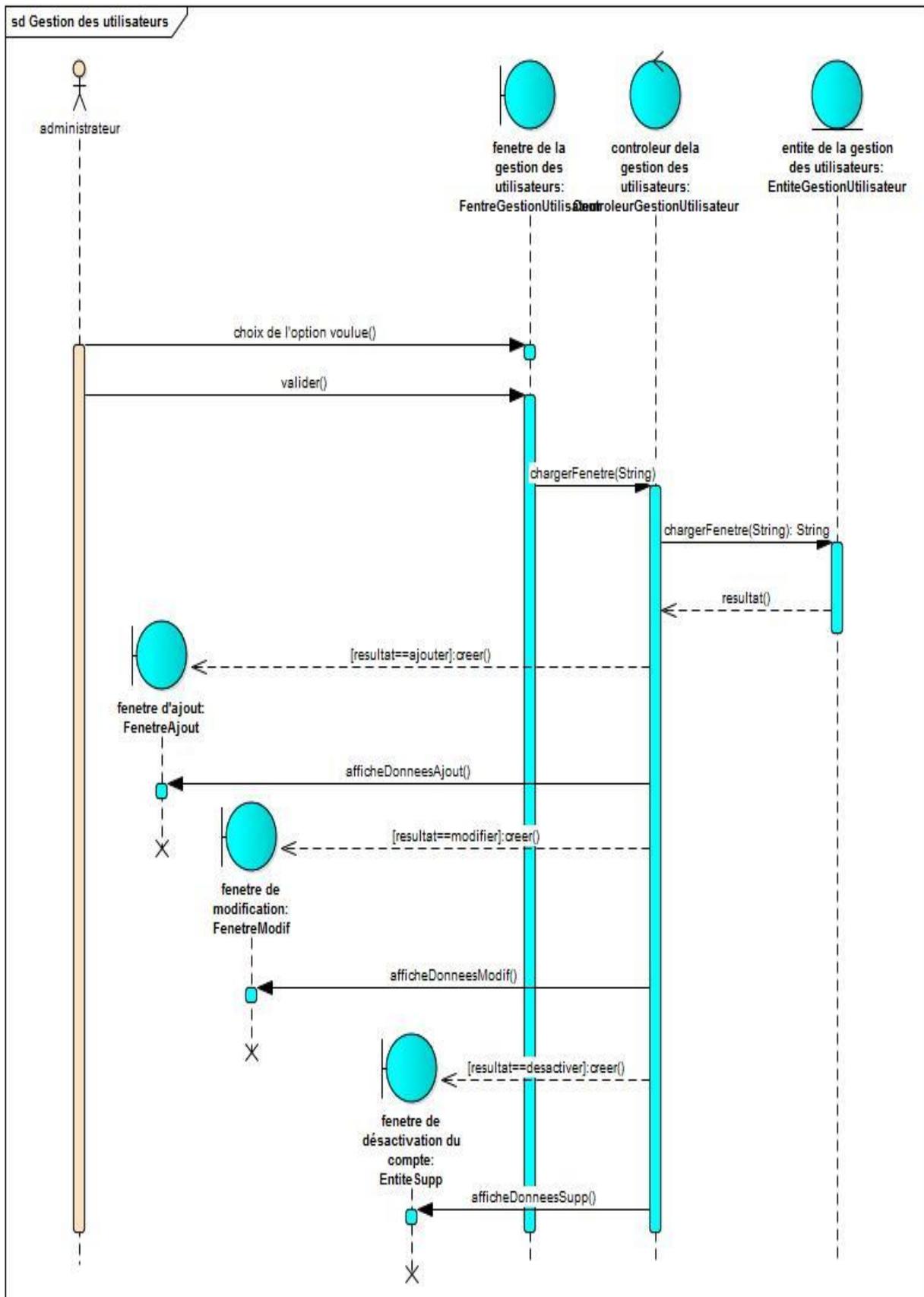


Figure 31 Diagramme de séquence pour la gestion des utilisateurs

L'administrateur doit s'authentifier pour accéder à son compte (voir la figure 32). Ce diagramme présente la démarche suivie.

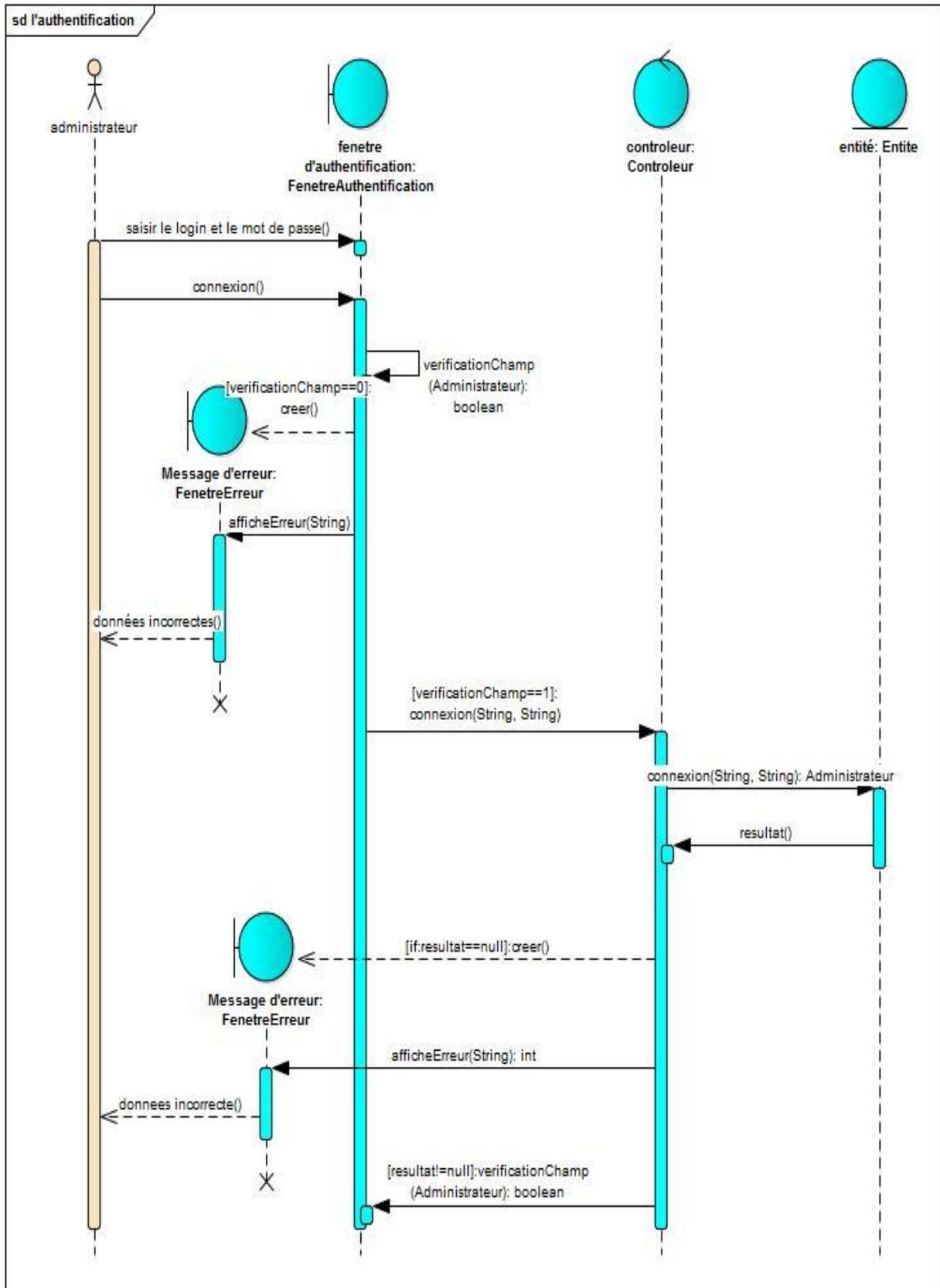


Figure 32 Diagramme de séquence pour l'authentification

L'administrateur a le droit de modifier les activités (voir la figure 33). Ce diagramme présente la démarche suivie.

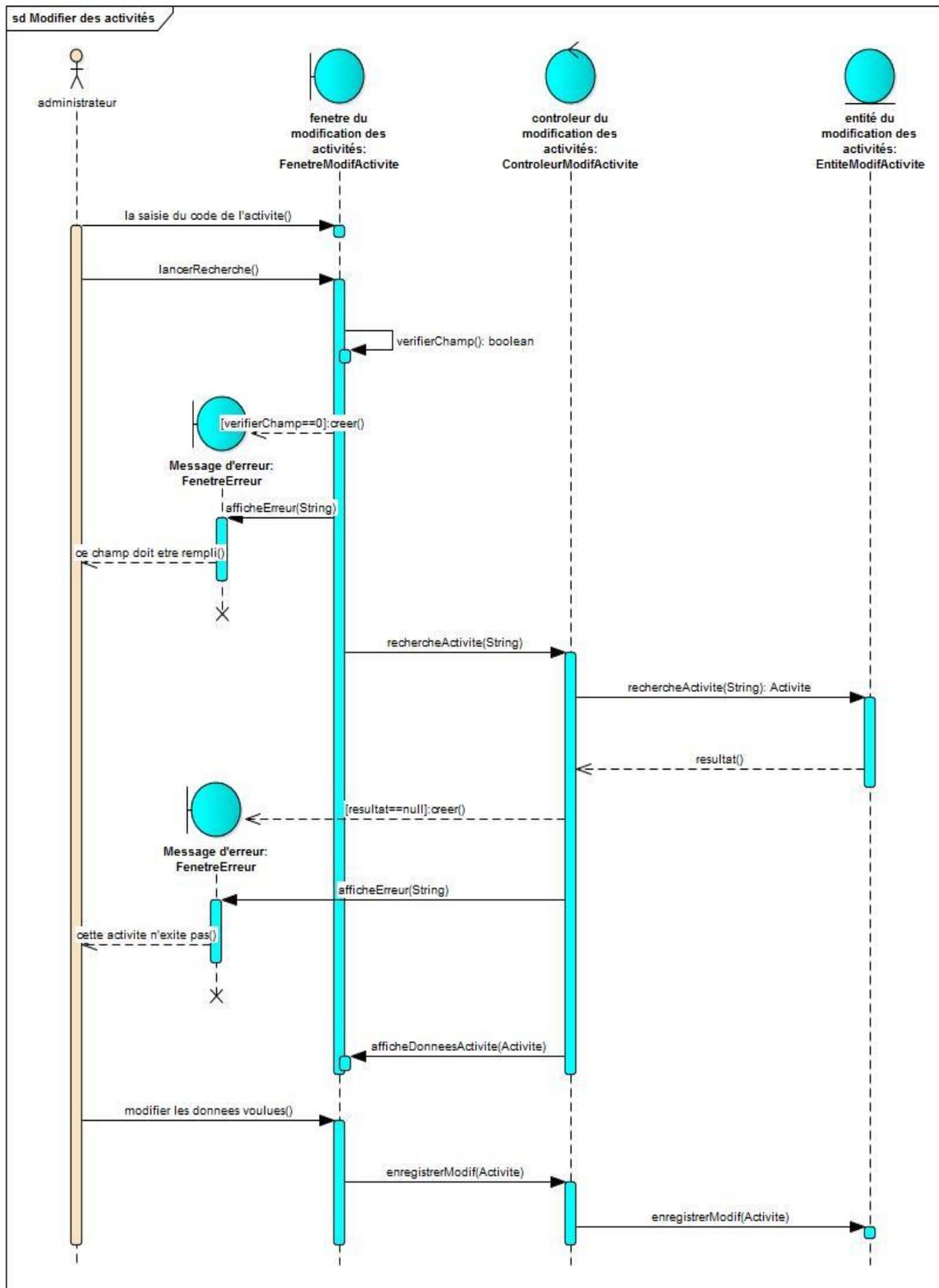


Figure 33 Diagramme de séquence pour modifier une activité

L'administrateur a le droit de modifier les agendas (voir la figure 34). Ce diagramme présente la démarche suivie.

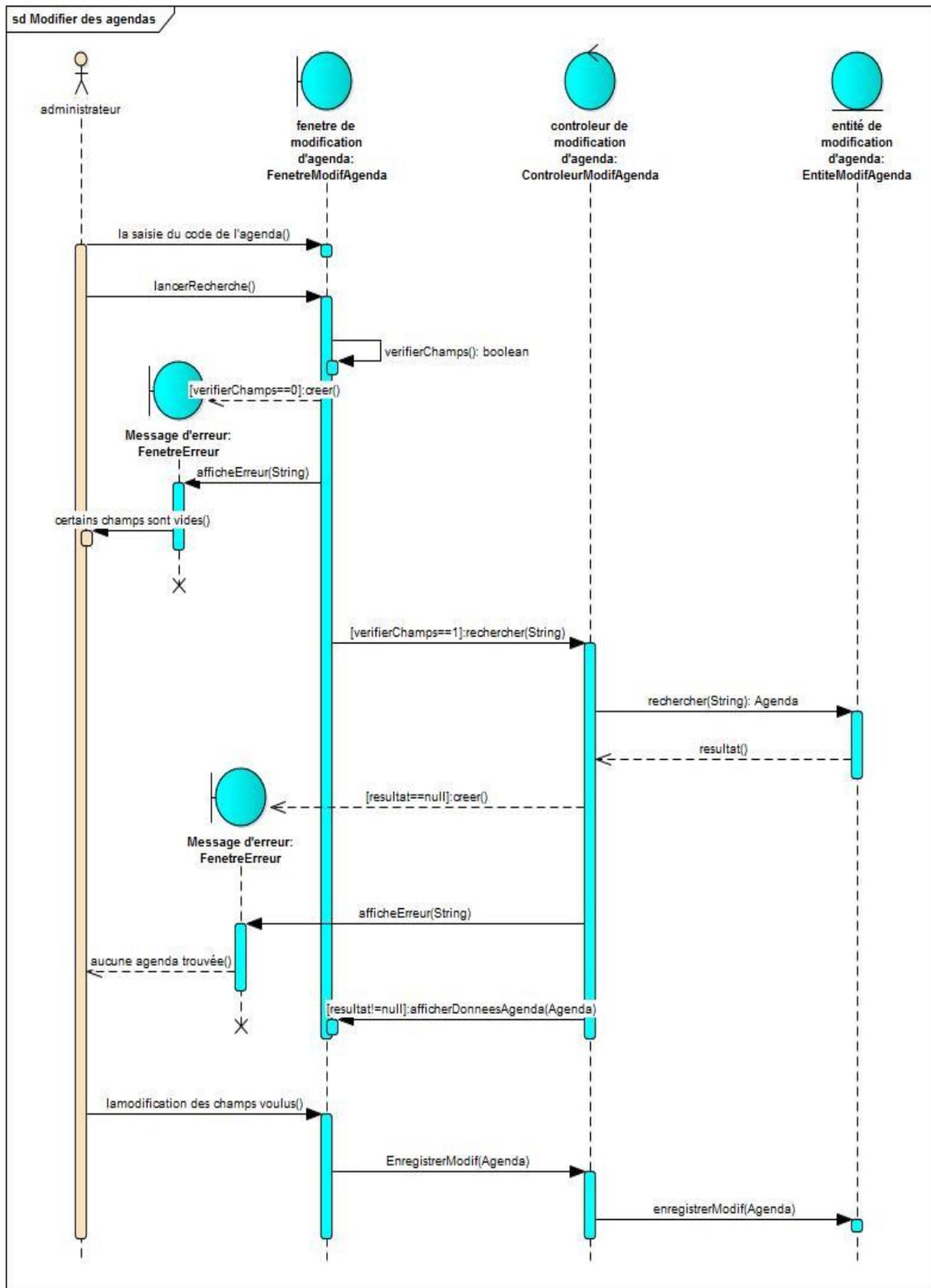


Figure 34 Diagramme de séquence pour modifier un agenda

L'administrateur a le droit de modifier les utilisateurs (voir la figure 35). Ce diagramme présente la démarche suivie.

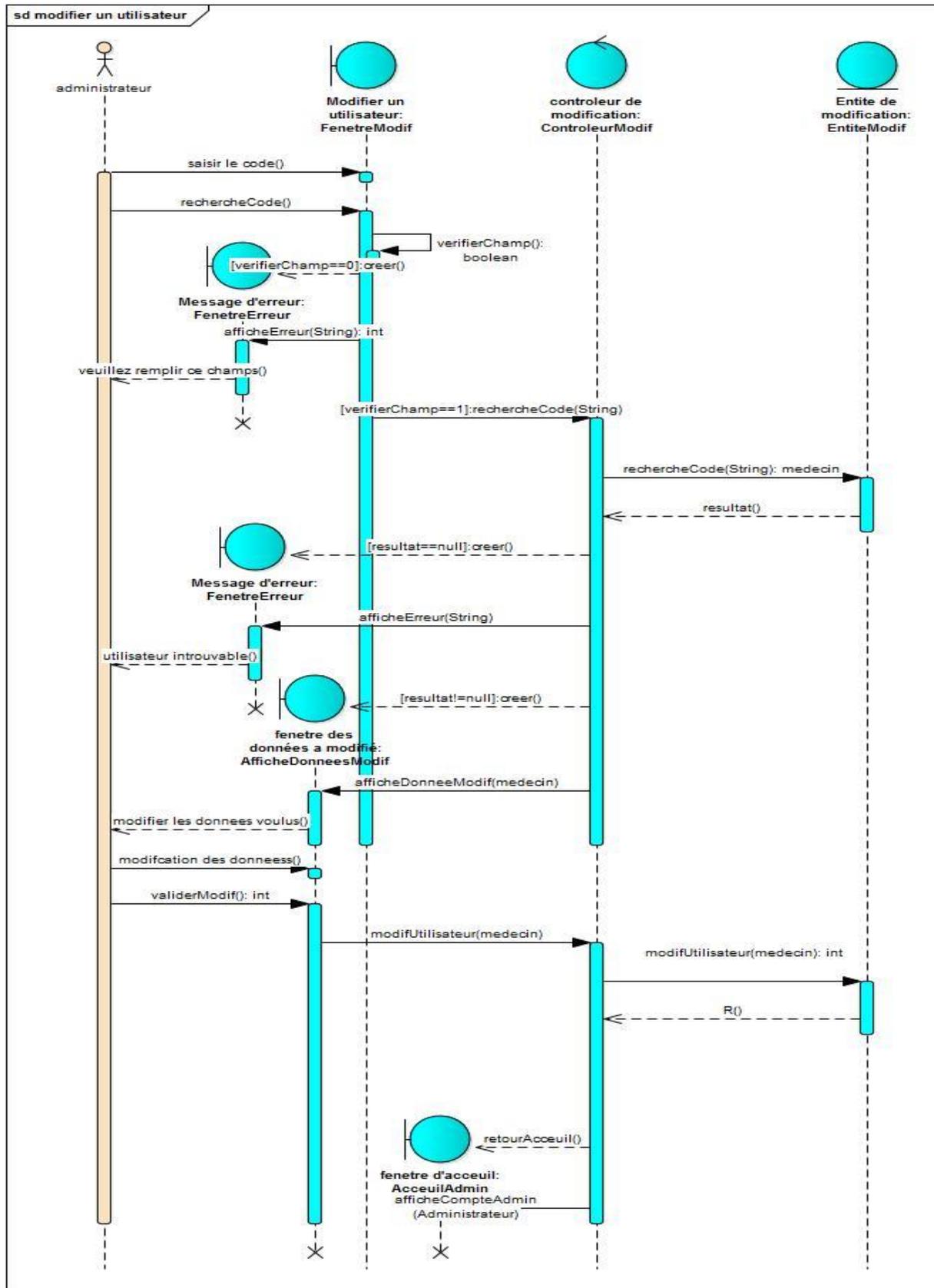


Figure 35 Diagramme de séquence pour modifier un compte utilisateur

L'infirmier à plusieurs droits, à savoir :

Le droit consulter les rendez-vous (voir la figure 36). Ce diagramme présente la démarche suivie pour cette consultation.

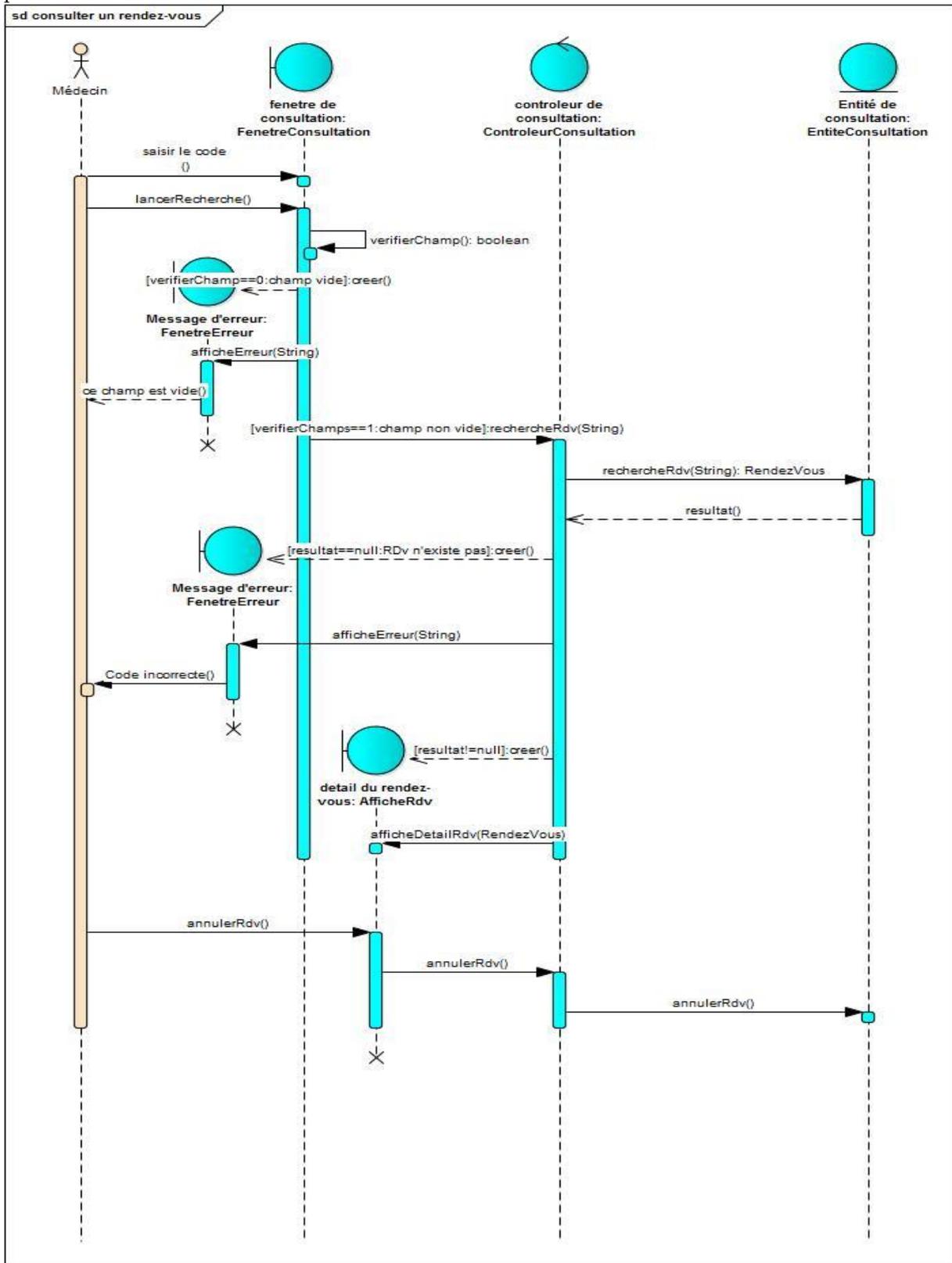


Figure36 Diagramme de séquence pour la consultation d'un rendez-vous

L'infirmier doit s'authentifier pour accéder à son compte (voir la figure 37). Ce diagramme présente la démarche suivie.

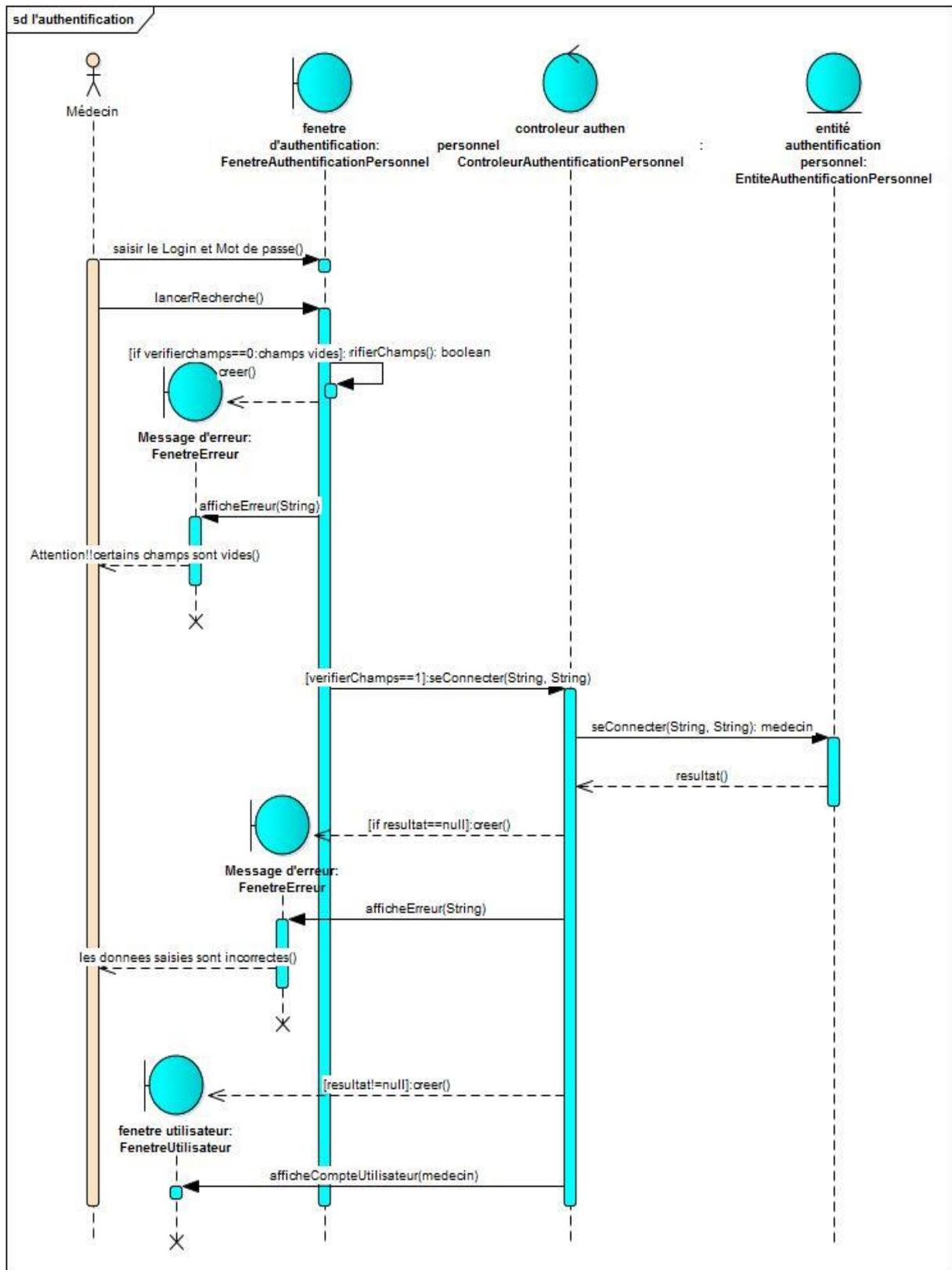


Figure 37 Diagramme de séquence pour l'authentification

L'infirmier a le droit de gérer les rendez-vous (voir la figure 38). Ce diagramme présente la démarche suivie.

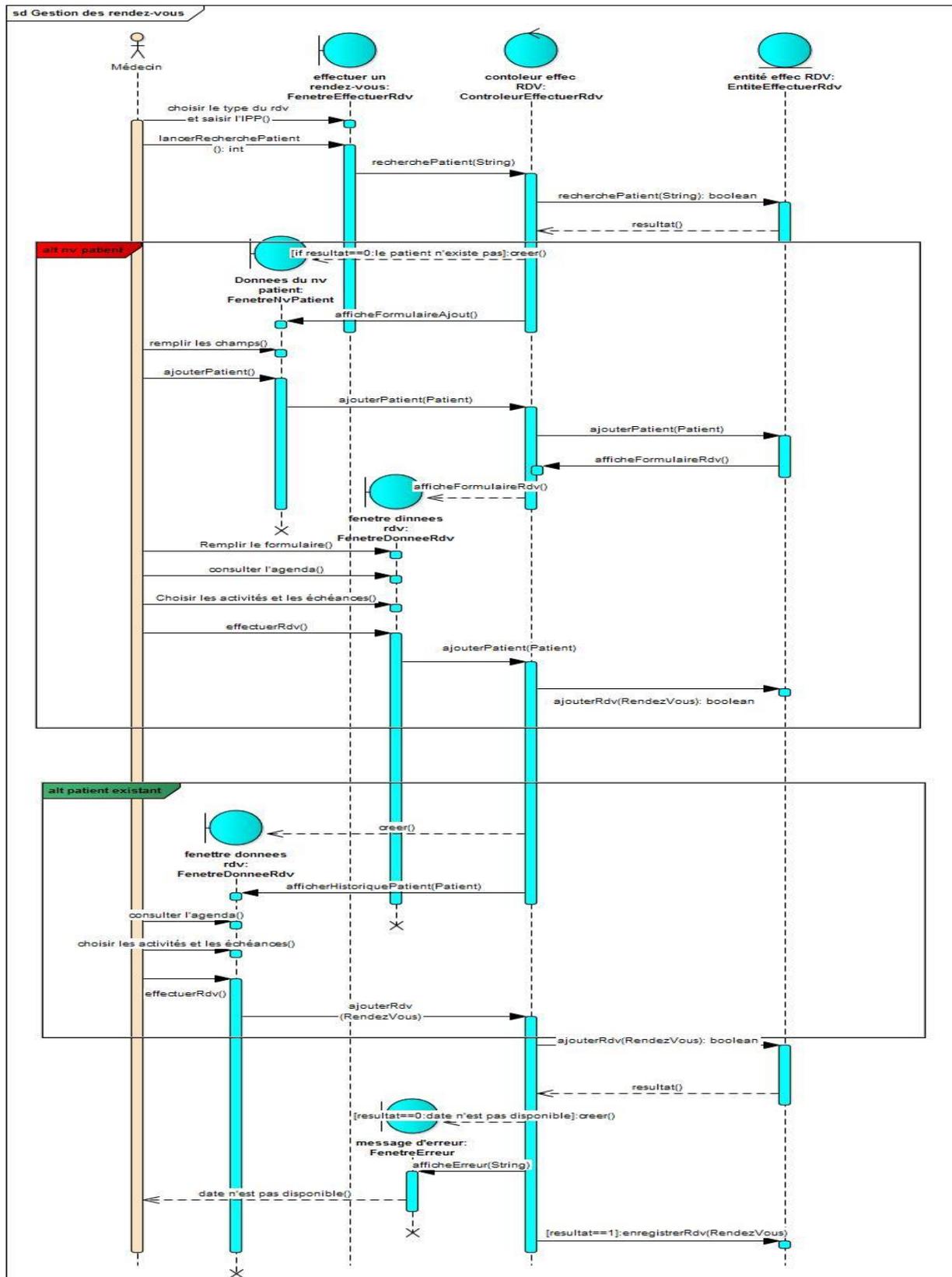


Figure 38 Diagramme de séquence pour la gestion des rendez-vous

La secrétaire à plusieurs droits, à savoir :
Le droit de gérer les rendez-vous (voir la figure 39). Ce diagramme présente la démarche suivie.

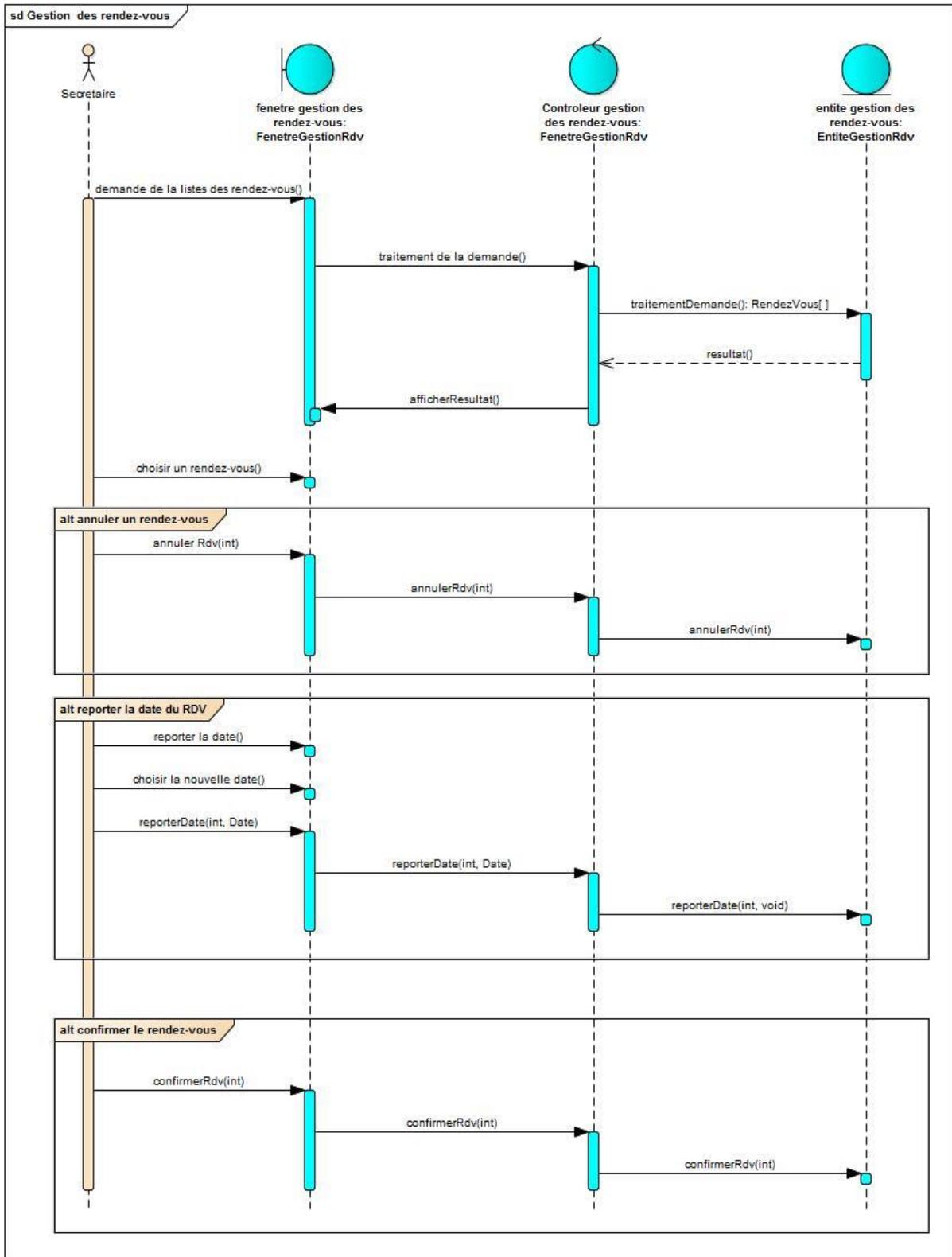


Figure 39 Diagramme de séquence pour la gestion des rendez-vous