

Sommaire

Introduction

.....	
.....	2

1. Présentation du projet	
.....	2
2. Analyse des besoins	
.....	2
2.1 Les besoins Techniques	
.....	2
2.2 Les besoins fonctionnels	
.....	2
2.3 Les besoins non fonctionnels	
.....	3
2.4 Les besoins Opérationnels	
.....	3
3. Description du contexte du système	
.....	4
3.1 Identification des acteurs	
.....	4

3.2 Identification des messages	
.....	5

3.3 Réalisation du diagramme de contexte dynamique	
.....	8

4. Périmètre fonctionnel	
.....	4

5. L'arborescence	
.....	
. 5	

6. Diagramme de package	
.....	

7. Diagramme de cas d'utilisation.....	
---	--

8. Diagramme d'activité	
.....	

Conclusion	
.....	
..... 8	

Bibliographie	
.....	
..... 8	

Introduction

L'étude d'un projet est une démarche stratégique qui va nous permettre d'avoir une vision globale sur ce dernier visant ainsi à bien organiser le bon déroulement du projet.

Cette étude fera donc l'objet du premier chapitre qui sera consacré à l'Analyse des besoins et la réalisation des spécifications générales.

1. Présentation du projet

Le projet consiste à mettre en place un système de gestion de rendez-vous pour un hôpital. L'hôpital est organisé en service, dans un service on trouve une spécialité et des médecins chaque médecin est spécialisé dans un domaine. Chaque service a un secrétariat qui gère les rendez-vous des patients par rapport au planning des médecins. Les rendez-vous peuvent être pris du lundi au vendredi de 8h à 12h et de 15h à 17h. Un rendez-vous dure 15mn.

L'idée sera de développer une plateforme web qui permettra aux secrétaires d'avoir une vue d'ensemble sur la disponibilité des professionnels de santé en accédant à leur agenda et pouvoir créer un rendez-vous pour un patient.

Comment parviennent-ils à optimiser leurs temps et celui de leurs médecins ? C'est simple : sur cette plateforme, les patients ont donc accès en temps réel aux agendas des médecins pour un service sur lequel ils s'exercent.

2. Analyse des besoins

2.1. Les besoins techniques

En ce qui concerne les besoins techniques pour notre application, nous allons utiliser les outils suivants :

- Le langage de modélisation : UML 2.0.
- le versionning : Git GitHub
- gestion du projet : Trello, méthode agile
- L'architecture de système : Bases de données (MySQL).
- Langage de programmation : PHP, JAVA SCRIPT, HTML, CSS.
- Framework : JQuery, Bootstrap etc.
- Plate-forme disponible : Windows, Linux, Mac OS .

2.2. Les besoins fonctionnels

Notre application aura pour but de réaliser les fonctionnalités suivantes :

- L'administrateur doit se connecter à la base de données pour ajouter, supprimer, sélection, éditer les gestionnaires.
- Le secrétaire doit s'authentifier pour accéder à son espace personnel.
- Le gestionnaire gère les médecins et les secrétaires.
- Le secrétaire doit s'authentifier pour accéder à son espace personnel.
- Le secrétaire gère les patients.
- Le secrétaire gère les RDV pour ses patients (CRUD).
- Le médecin doit s'authentifier pour accéder à son espace personnel.
- Le médecin peut également consulter la liste de ses RDV.
- Le médecin doit gérer le dossier des patients (méthode CRUD).
- Tout le monde peut voir son profil

2.3. Les besoins non fonctionnels

Les besoins non fonctionnels sont importants car ils agissent de façon indirecte sur le résultat et sur la performance du système, ce qui fait qu'ils ne doivent pas être négligés, pour cela il faut répondre aux exigences suivantes :

- Le système doit être sécurisé au niveau des données : authentification et contrôle des droits d'accès.
- Le système doit permettre l'accomplissement des tâches avec le minimum des manipulations.
- Respecter la charte graphique de la Sonatel académie
- Le système doit être assez attractif et l'utilisation assez simple et intuitif pour l'utilisateur.

2.4. Les besoins opérationnels

- Pour assurer la fiabilité de notre application, chaque utilisateur (secrétaire, médecin et gestionnaire) doit avoir un login/E-mail et un mot de passe pour consulter son compte.
- La page secrétaire doit être accessible qu'aux secrétaires uniquement, même règle pour le gestionnaire et le médecin.
- Le secrétaire doit créer un compte pour chaque patient.
- Un patient a un identifiant unique.

3. Description du contexte du Système

Ici, nous allons expliquer comment à travers l'application, les échanges d'informations des différents utilisateurs, ce qui peut être résumé en trois activités successives qui sont :

- L'identification des acteurs.
- L'identification des messages.
- Réalisation du diagramme de contexte dynamique.

3.1. Identification des acteurs [1]

Un acteur est l'abstraction d'un rôle joué par des entités externes du système (Utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système.

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 peuvent être :

- Humains : ce sont des utilisateurs du logiciel .
- Logiciels : ce sont des logiciels déjà disponibles qui communiquent avec le système grâce à une interface logicielle (API).
- Matériels : robots et automates qui exploitent les données du système ou qui sont pilotés par le système.

Les acteurs existants dans notre système sont :

- Le gestionnaire
- Le médecin.
- Le secrétaire .
- L'administrateur.

Le secrétaire: C'est l'acteur essentiel de notre projet, le site web pour lui est l'endroit où il peut donner des RDV aux patients.

Le médecin : notre site web est consacré aussi aux médecins qui peuvent consulter leurs rendez-vous et gérer le dossier médical des patients.

Le gestionnaire : Il est aussi l'élément clé de notre système, son rôle est de gérer nos deux acteurs secrétaire et médecin.

L'administrateur : C'est l'informaticien qui est accès à la base de données. Il a la possibilité d'ajouter un gestionnaire.

3.2. Identification des messages [2]

Message : c'est la spécification d'une communication unidirectionnelle entre objets qui peut transporter de l'information avec l'intention de déclencher une activité chez le récepteur.

A un haut niveau, le message est utilisé pour décrire les interactions entre les acteurs et le système.

Remarques : Les messages entre acteurs ne sont d'aucune utilité.

Messages du secrétaire :

A : le secrétaire s'authentifie

B : le système vérifie l'existence.

C. le secrétaire effectue une recherche basée sur le nom/prénom et la spécification du médecin.

D. le système affiche la liste des médecins.

E. le secrétaire clic sur un médecin pour prendre un RDV.

F : le système affiche un formulaire à remplir y compris le calendrier.

G : le secrétaire remplit le formulaire et valide.

H : Le système envoie les informations dans la table RDV.

I. Le système valide le RDV.

Message du médecin :

L. le médecin s'authentifie.

M. le système vérifie puis affiche l'espace privé du médecin.

N. le médecin peut demander les rendez-vous qui lui ont été attribués.

O. le système affiche la liste des RDV.

P. le médecin peut demander à gérer un dossier médical pour un patient connaissant son matricule.

Q. le système affiche un formulaire à remplir.

R. le médecin remplit le formulaire.

S. le système l'ajoute dans la base de données (table dossier médical).

T. le médecin a la possibilité de changer, éditer ou supprimer le dossier médical.

Messages du gestionnaire :

U. Le gestionnaire s'authentifie.

V. Le système affiche l'espace du gestionnaire contenant les liens pour gérer l'organisation de l'hôpital.

W. Le gestionnaire peut demander de créer un compte médecin, secrétaire, ajouter une spécialité ou service.

X. le système lui envoie un formulaire.

Y. Le gestionnaire remplit le formulaire.

Z. Le système l'ajoute dans la base de données.

3.3. Réalisation du diagramme de contexte dynamique [2]

A partir des informations obtenues lors des deux précédentes étapes, nous allons modéliser le contexte de notre application.

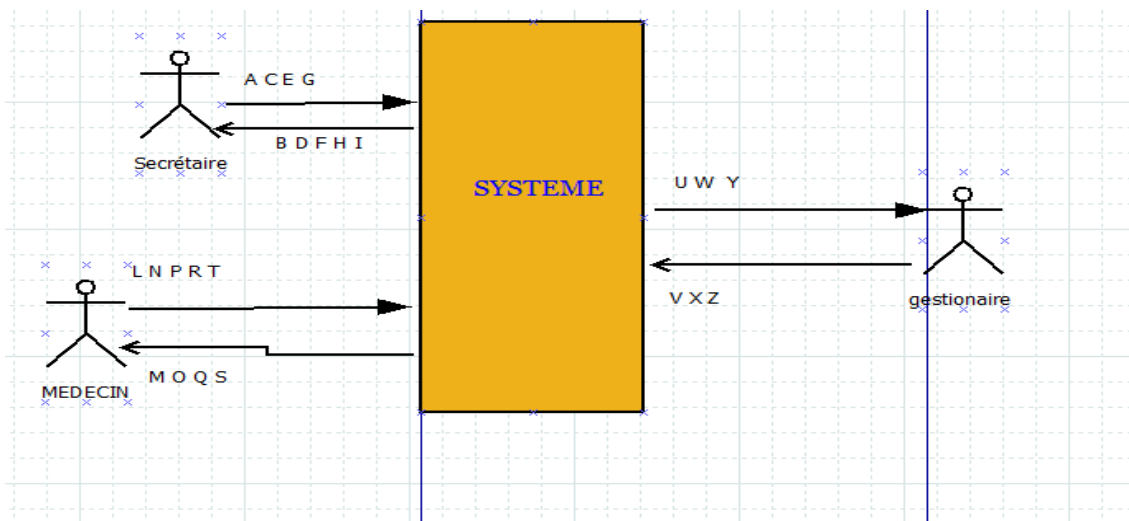


Figure 1 : Diagramme de contexte dynamique.

4 : Périmètre fonctionnel

Délimiter le périmètre fonctionnel de notre site, permet de savoir exactement ou s'arrêter dans notre développement, cela revient donc à lister de manière aussi exhaustive que possible les fonctionnalités dont auront besoin nos utilisateurs pour se servir du site.

Pour cela, nous allons faire ce que l'on appelle un **impact mapping**.

Faire un impact mapping revient à se poser **une succession de bonnes questions préétablies** :

- Quel est l'objectif de mon site ?
- Quels sont les différents profils de ses utilisateurs ?
- Quelles actions cherchent-ils respectivement à faire sur mon site ?
- Quelles fonctionnalités vont leur permettre de mener à bien ces actions ?

En répondant à ces questions de manière systématique, nous aboutissons à une cartographie des usages supposés du site. Remarquons que chaque colonne correspond à une des questions évoquées plus haut.

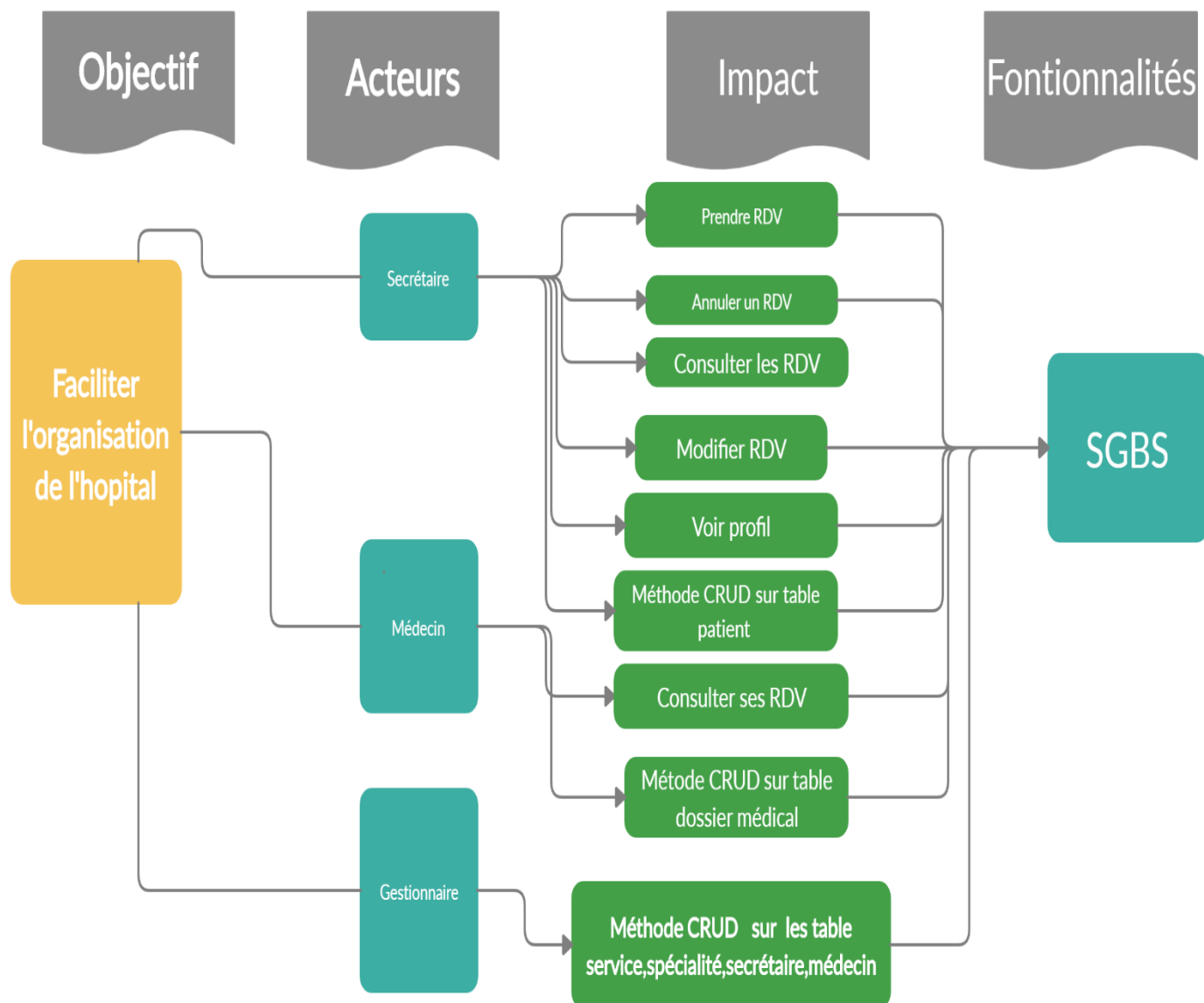


Figure 1 : périmètre fonctionnel du logiciel

5 : Arborescence

L'arborescence permet d'avoir une vision globale du site web et des parcours de navigation avant de réaliser les maquettes des pages principales.

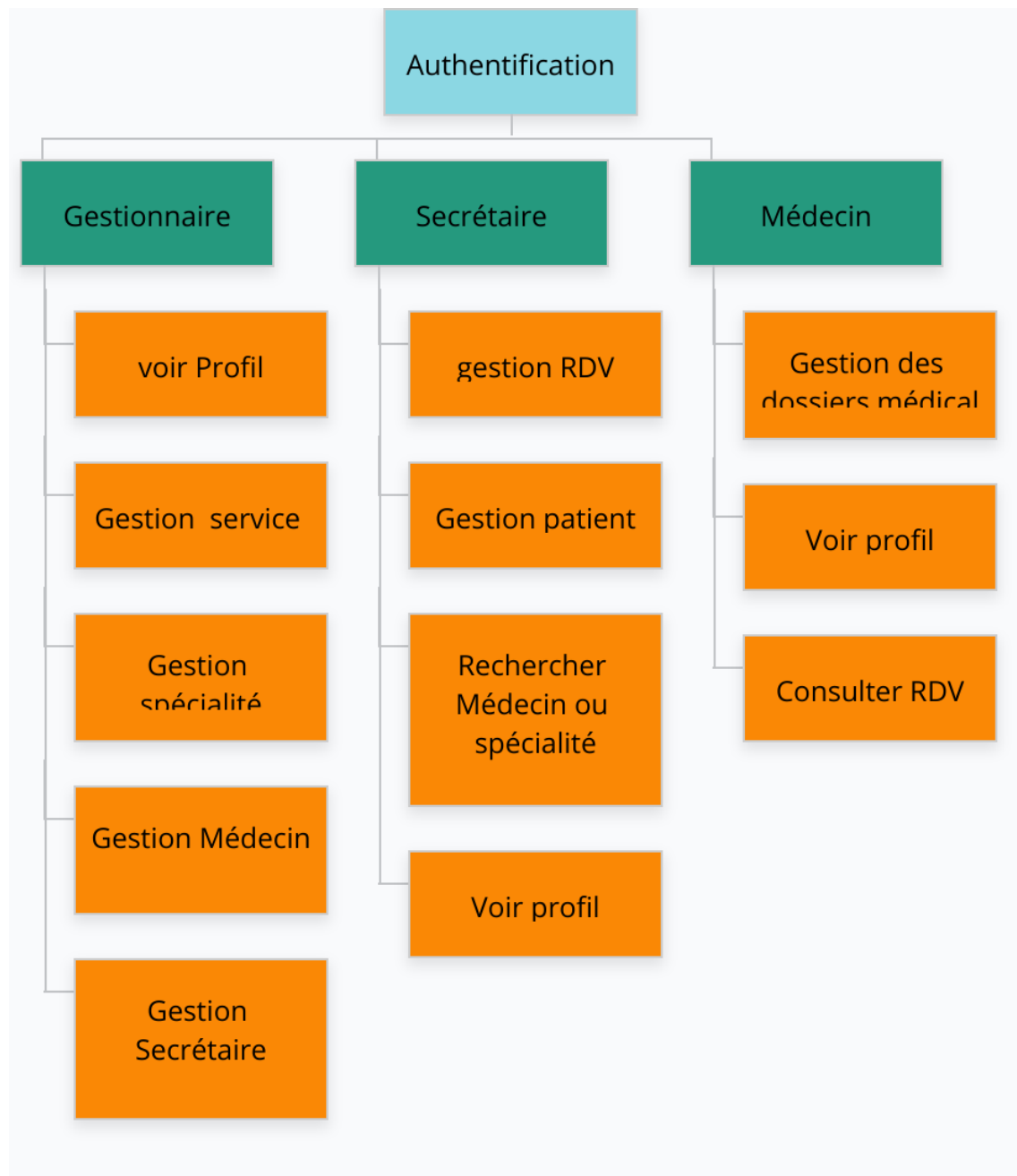
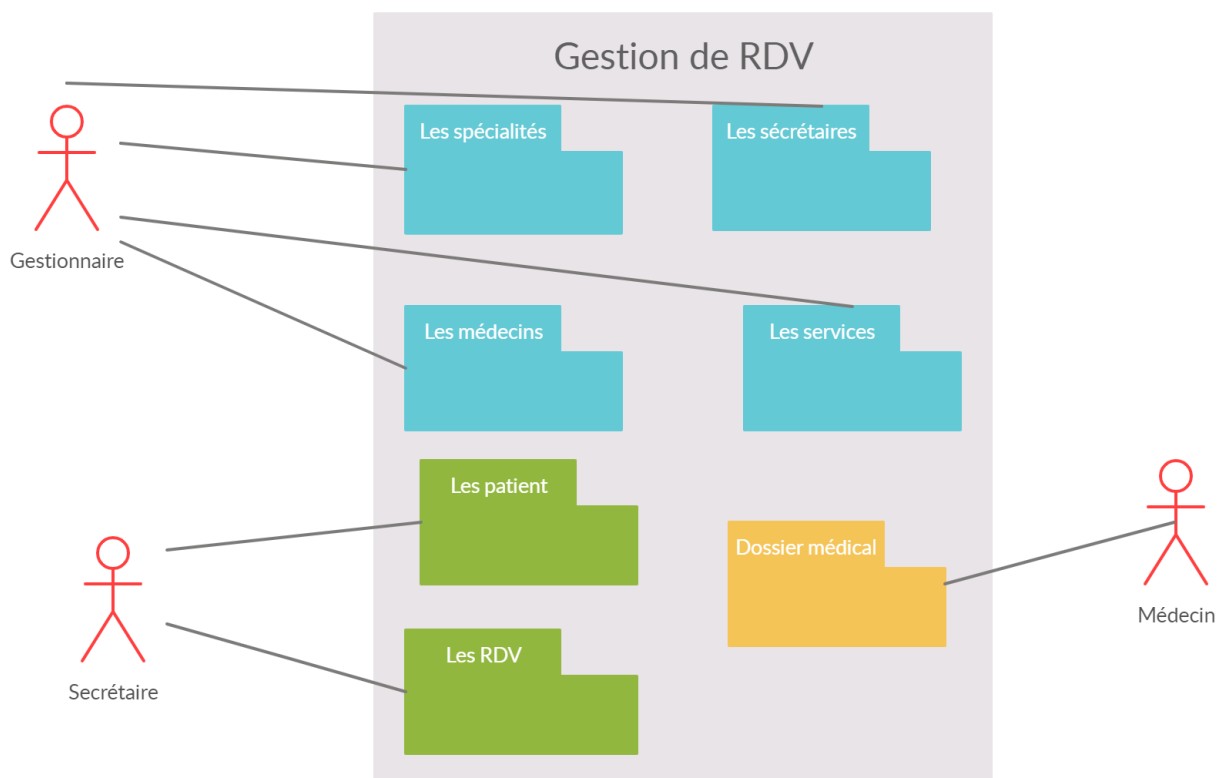


Figure 2 : Arborescence du logiciel

6 : le diagramme de package

Les besoins très différents des acteurs et le nombre de fonctionnalités dont le futur logiciel devra disposer nous semble assez souvent compliqué. Pour y voir clair et pour nous faciliter la tâche, on peut découper le futur logiciel en parties distinctes, en fonction des « familles » de fonctionnalités et de façon à pouvoir les analyser séparément. Chacune de ces parties correspond à un domaine fonctionnel ou **package**.

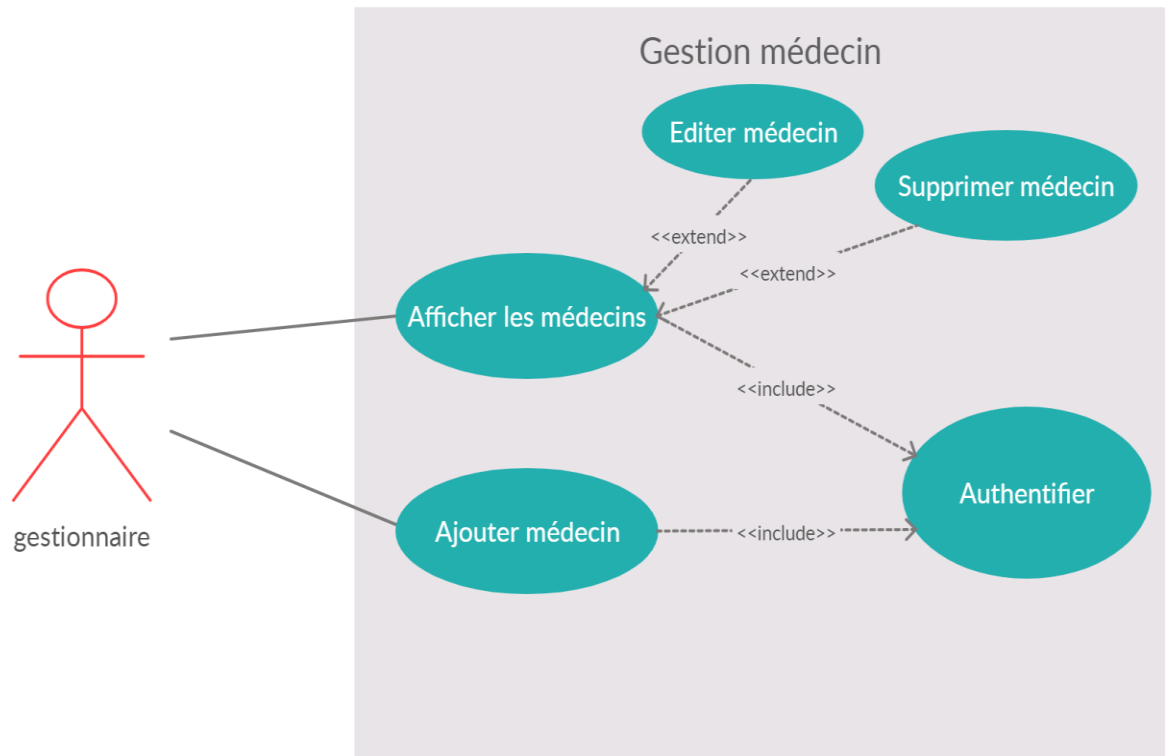


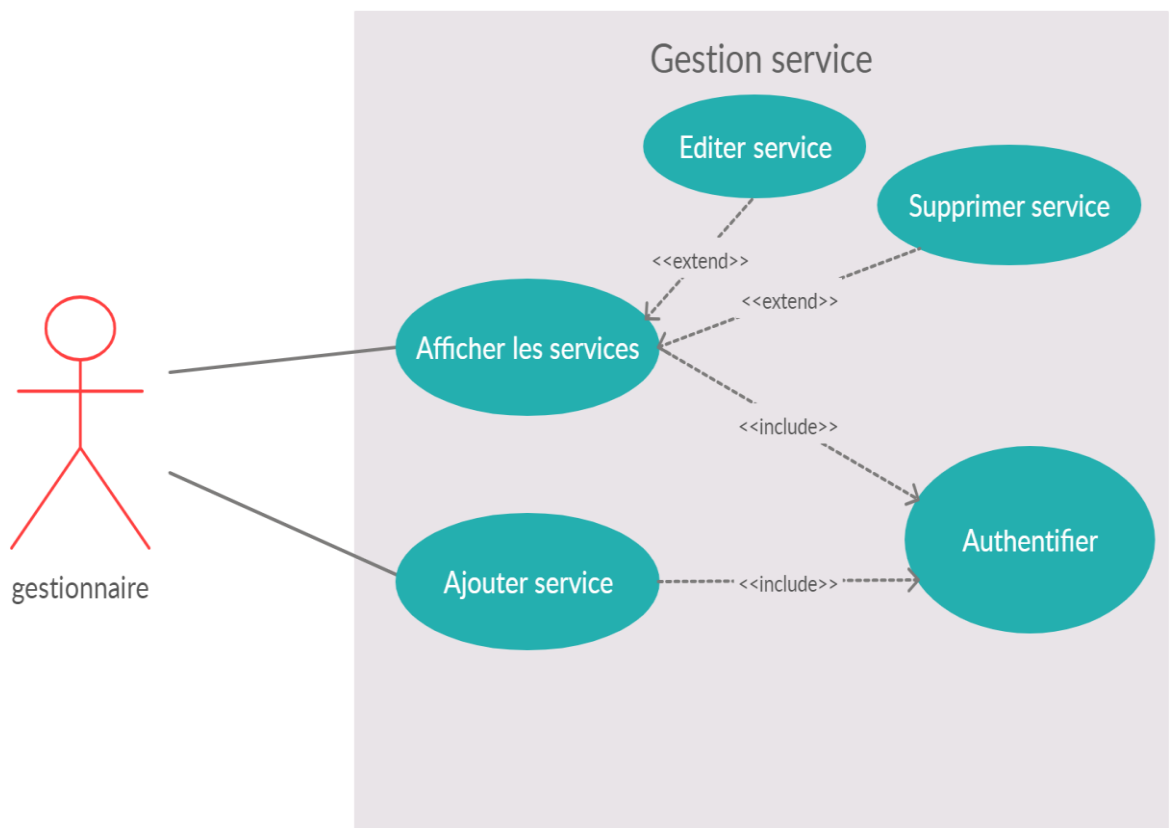
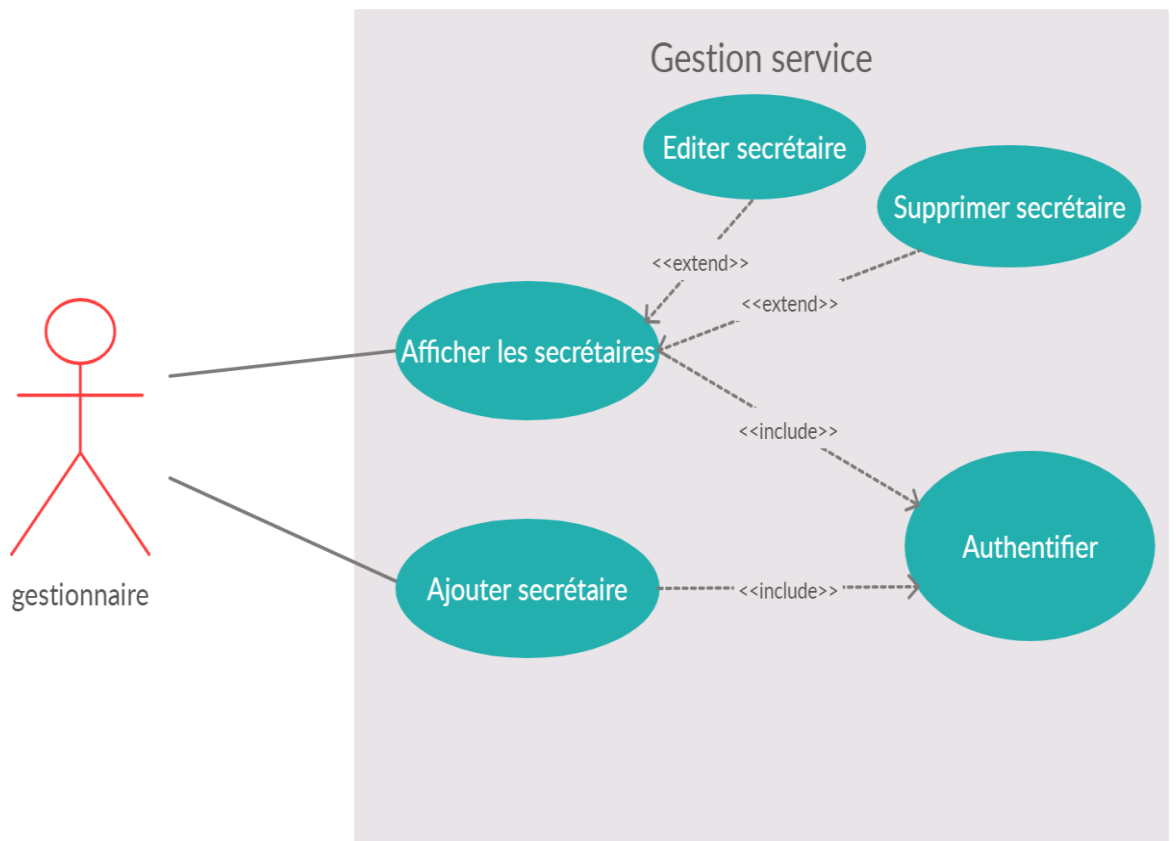
7 : le diagramme de cas d'utilisation

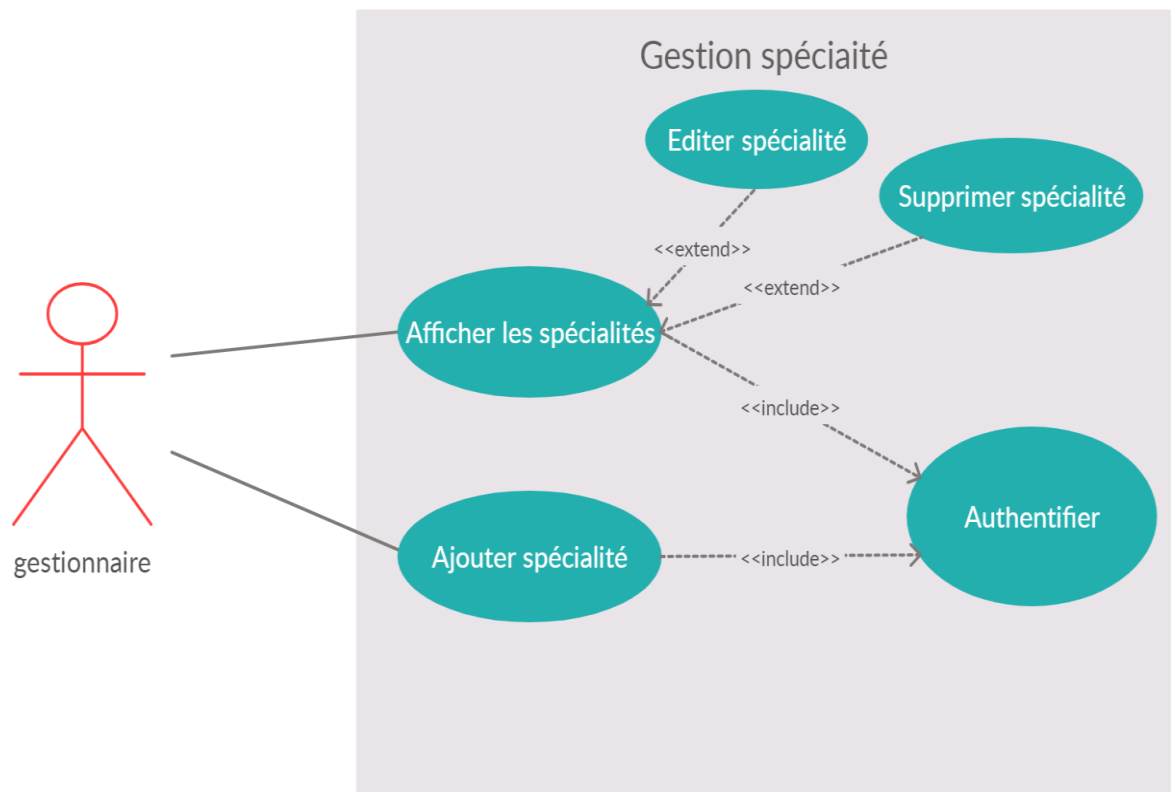
Le diagramme des cas d'utilisations met en évidence de quelle façon les acteurs utiliseront le logiciel : QUI doit pouvoir faire QUOI ?

Dans notre système nous allons donner pour chaque acteur le/les diagramme(s) de cas d'utilisation correspondant.

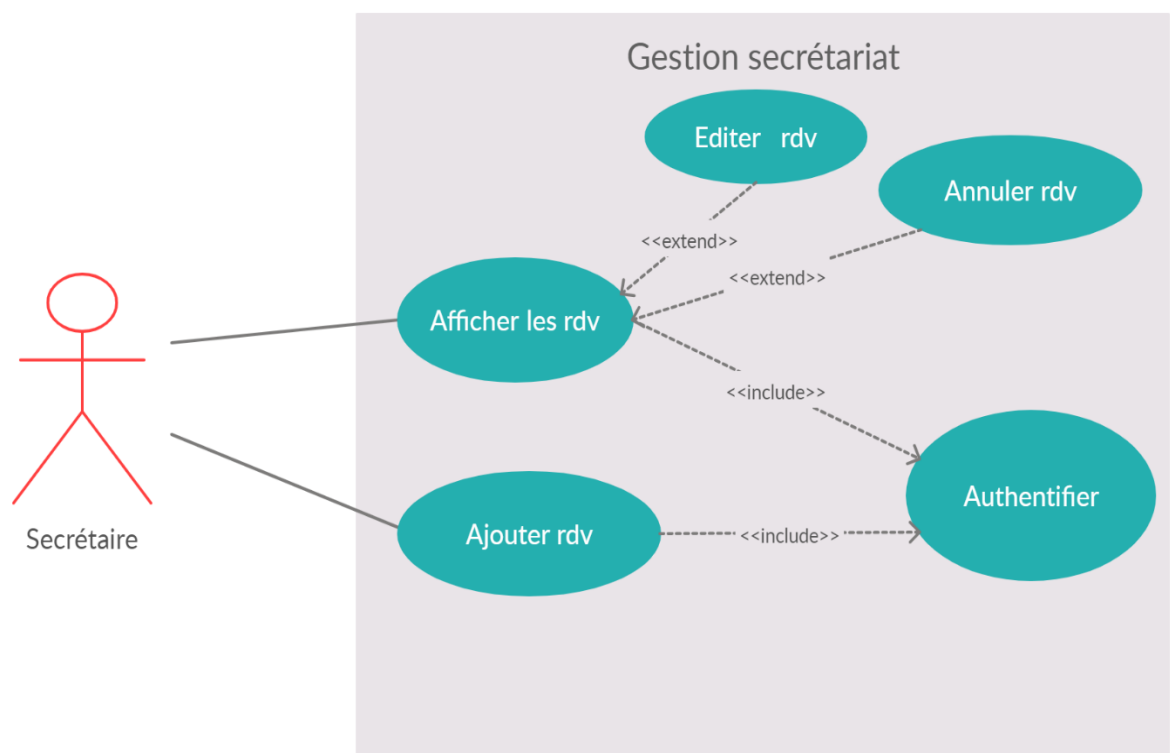
- Gestionnaire

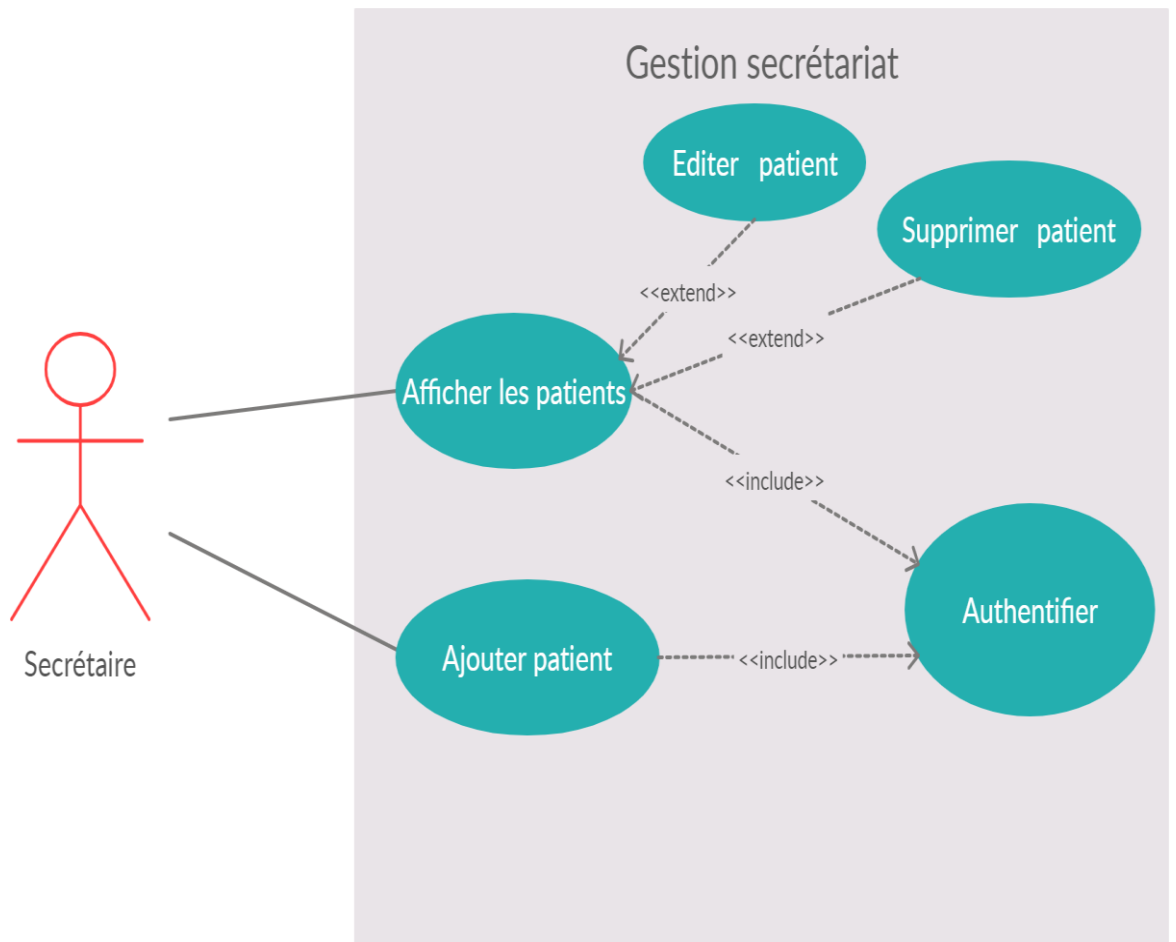




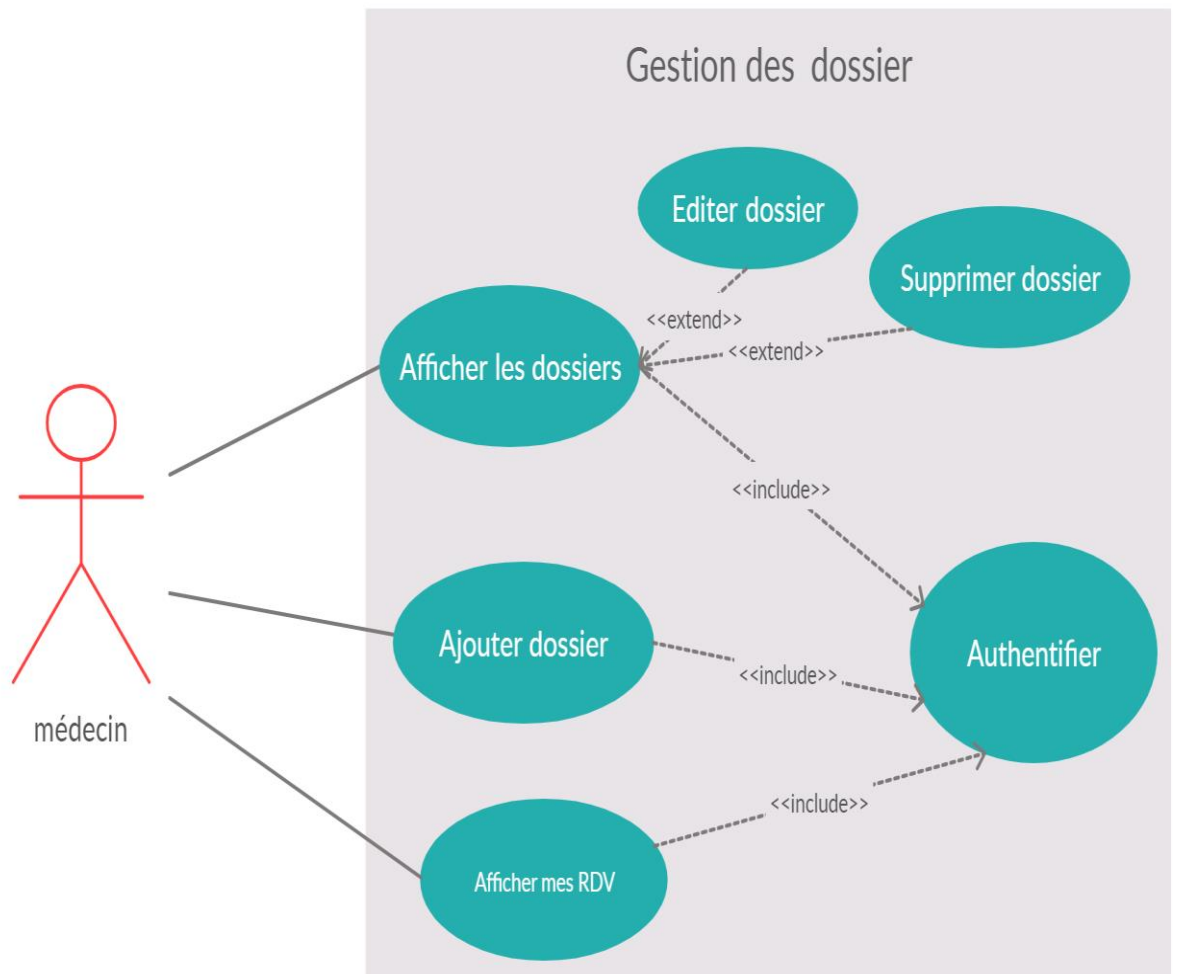


- Secrétaire





- Médecin

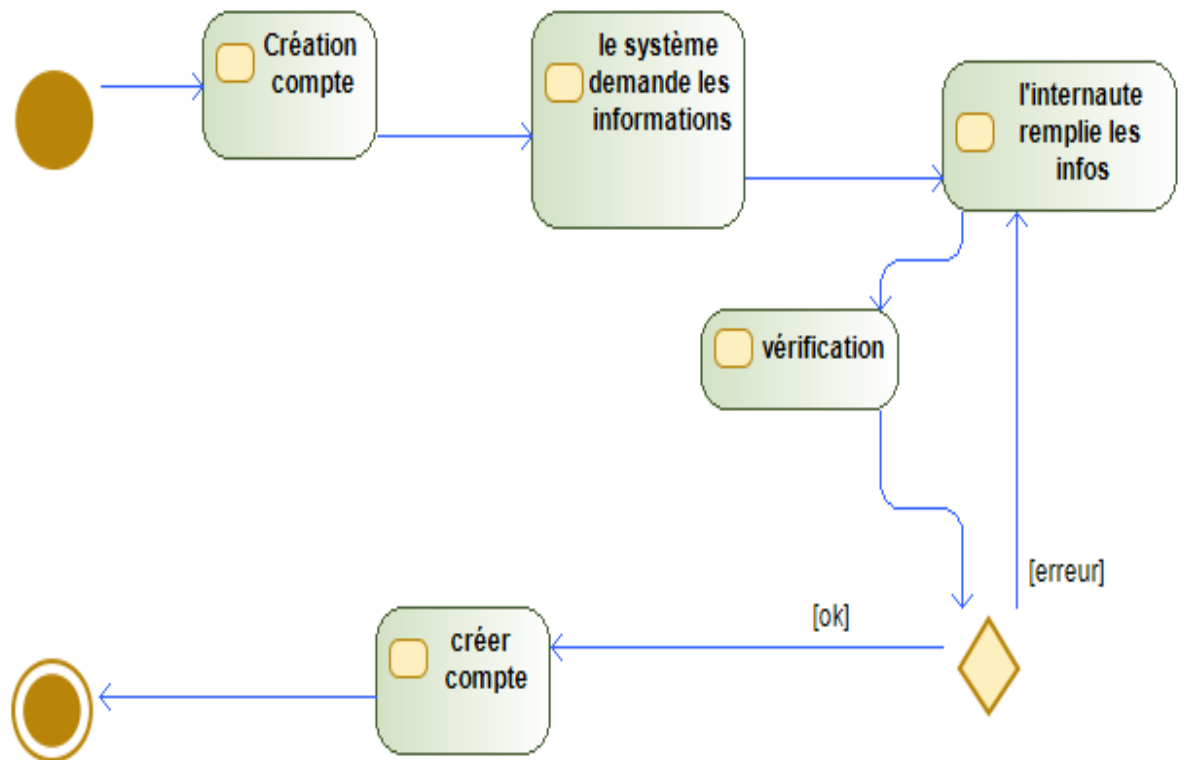


7 : le diagramme d'activité

Les diagrammes d'activités permettent de mettre l'accent sur les traitements. Son rôle est de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

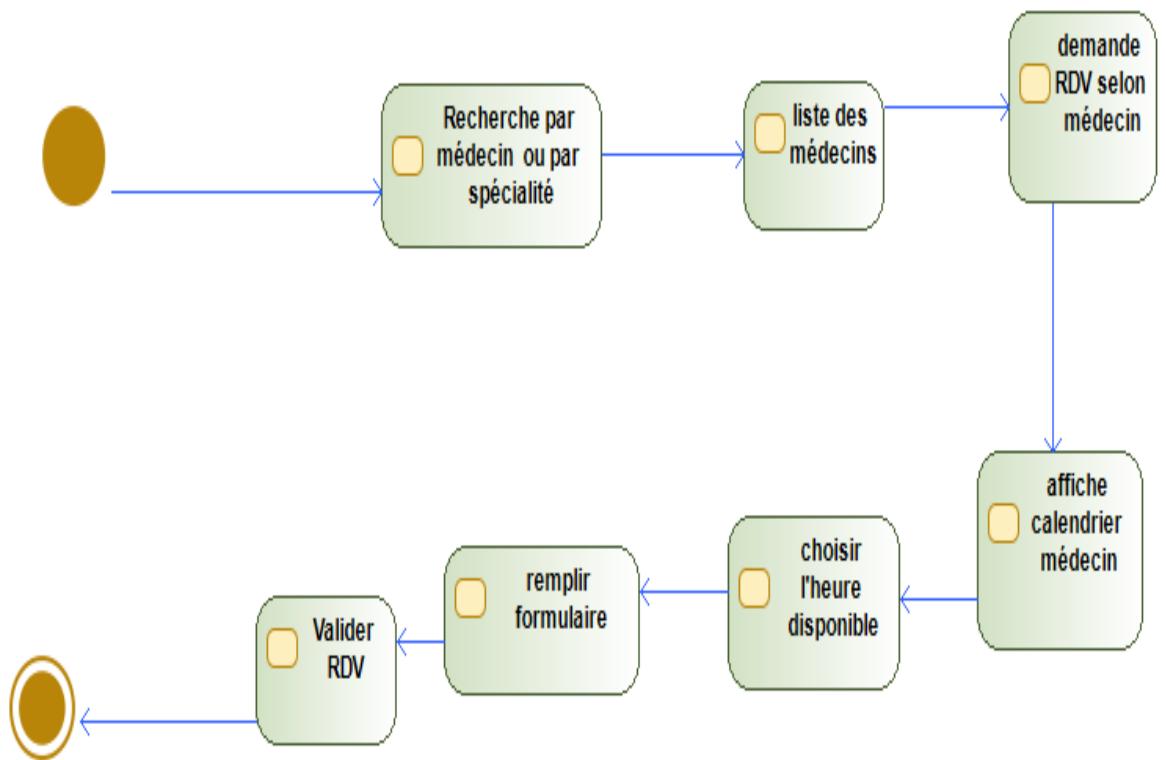
Dans notre situation, nous n'allons pas exploiter l'ensemble des activités possible mais nous résumerons sous deux diagrammes d'activité : création de compte et prendre RDV.

▪ Création compte



Le diagramme d'activité création compte

- Prendre rendez-vous



Conclusion

Dans ce chapitre, nous avons présenté le cadre général de notre projet et nous avons exposé l'analyse et la spécification des besoins permettant de concevoir et de développer un système qui facilitera la gestion des rendez-vous médicaux et aussi nous avons exposé l'arborescence, le périmètre fonctionnel, le diagramme de package, le use case et le diagramme d'activité.

Après avoir fixé nos objectifs, l'étape suivante sera consacrer à une conception détaillée de l'architecture du système.

Bibliographies :

1. <http://prive.iutenligne.net/qZ4UsW3zySLb7sfK/informatique/langages/kettaf/UML/04modeleconceptuel/0302acteurs.html>
2. https://www.memoireonline.com/04/17/9870/m_Conception-et-realisation-dun-site-web-dynamique-pour-la-prise-de-rendez-vous-medicale-en-ligne11.html
3. https://www.memoireonline.com/04/17/9870/m_Conception-et-realisation-dun-site-web-dynamique-pour-la-prise-de-rendez-vous-medicale-en-ligne14.html