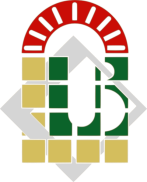
الجمهورية الجزائرية الديموقراطية الشعبية

République Algérienne Démocratique et Populaire

وزارة التعليم العالي و البحث العلمي

Ministère de l’enseignement supérieur et de la recherche scientifique



Université Mohamed Khider - Biskra

Faculté : faculté des Sciences Exactes, des Sciences De la nature et de la vie

Département : Informatique

Réf : ……………..

Mémoire présenté en vue de l’obtention Du Diplôme de Master Siod

Master en Informatique

# Option : System Information Et Optimisation la Décision

**Application Web Pour Gestion Réservation Médical**

Présenté par :

# Alia Ramzi

Soutenu publiquement le ……..

Table de Content

[1 Introduction General 1](#_Toc129212247)

[Chapitre 1 : System de Réservation en ligne 3](#_Toc129212248)

[1 Problématique et domaine d’étude: 3](#_Toc129212249)

[2 Les Application web : 3](#_Toc129212250)

[3 Les types applications web : 3](#_Toc129212251)

[3.1 Application web statique : 4](#_Toc129212252)

[3.2 Application web dynamique : 4](#_Toc129212253)

[3.3 Application web portail: 4](#_Toc129212254)

[3.4 Application web page unique : 4](#_Toc129212255)

[3.5 Application web progressives : 5](#_Toc129212256)

[4 Principe Fonctionnement application web : 5](#_Toc129212257)

[5 Architecture application web : 6](#_Toc129212258)

[5.1 L'architecture client-serveur : 6](#_Toc129212259)

[5.2 L'architecture en couches : 6](#_Toc129212260)

[5.3 L'architecture orientée services : 6](#_Toc129212261)

[6 Les outils de développement des applications Web : 7](#_Toc129212262)

[6.1 Les langages de programmation : 7](#_Toc129212263)

[6.2 La Base de données : 7](#_Toc129212264)

[6.3 IDE : 8](#_Toc129212265)

[7 Organigramme général d’un cabinet privé : le cabinet privé il comporte plusieurs services médicaux que nous allons présenter : 8](#_Toc129212266)

[7.1 Service technique et exploration: 8](#_Toc129212267)

[7.2 Services médicaux: 8](#_Toc129212268)

[8 Problématiques et objectives: 9](#_Toc129212269)

[8.1 Problématiques: 9](#_Toc129212270)

[8.2 Objectifs de notre travail : 9](#_Toc129212271)

[9 Présentation le Projet: 9](#_Toc129212272)

[9.1 Gestion et suivi du dossier médical des patients : 10](#_Toc129212273)

[9.2 Gestion des Rendez-vous des patients : 10](#_Toc129212274)

[9.3 Gestion le résultat laboratoire analyse et radiologie: 10](#_Toc129212275)

[9.4 Gestion des ordonnances et médicaments : 10](#_Toc129212276)

[10 Analyse des besoins: 11](#_Toc129212277)

[10.1 Les besoins fonctionnels : 11](#_Toc129212278)

[10.2 Les besoins non fonctionnels: 11](#_Toc129212279)

[11 Les Application web Similaire : 11](#_Toc129212280)

[11.1 ZocDoc : 11](#_Toc129212281)

[11.2 DocToLib : 12](#_Toc129212282)

[12 Langage de modélisation UML: 12](#_Toc129212283)

[13 Conclusion: 13](#_Toc129212284)

[Chapitre 2 : Conception System Gestion Réservation en ligne 14](#_Toc129212285)

[1 Introduction : 14](#_Toc129212286)

[2 Diagramme De Cas D’Utilisation : 14](#_Toc129212287)

[2.1 Acteur : 14](#_Toc129212288)

[2.2 Relations entre les cas d’utilisation: 14](#_Toc129212289)

[2.3 Les Acteurs de system Réservation en ligne : 15](#_Toc129212290)

[2.4 Identifié les Cas d’Utilisation Selon Acteur : 15](#_Toc129212291)

[2.5 Diagramme Cas d’utilisation pour le Patient : 16](#_Toc129212292)

[2.6 Diagramme Cas d’utilisation pour le Médecin  et Admin de cabinet: 17](#_Toc129212293)

[2.7 Diagramme Cas d’utilisation pour fournisseur service de santé : 18](#_Toc129212294)

[2.8 Diagramme Cas d’utilisation pour le pharmaceutique : 19](#_Toc129212295)

[2.9 Diagramme Cas d’utilisation pour le secrétaire: 19](#_Toc129212296)

[2.10 Description textuel des cas d’utilisation: 20](#_Toc129212297)

[3 Diagramme de Séquence : 25](#_Toc129212298)

[3.1 Déférents diagramme des sequences: 26](#_Toc129212299)

[3.1.1 diagramme des sequence pour Authentification: 26](#_Toc129212300)

[3.1.2 diagramme des séquence pour Établire un réservation médical: 26](#_Toc129212301)

[3.1.3 diagramme des sequence Établir un order medical: 27](#_Toc129212302)

[3.1.4 diagramme des sequence Consultation résultat médical: 28](#_Toc129212303)

[3.1.5 diagramme des séquence pour consulter l’ordonnance médical par le pharmacien: 30](#_Toc129212304)

[3.1.6 Diagramme de séquence ajouté médecin par Admin cabinet médical: 31](#_Toc129212305)

[4 Le Diagramme de classe : 32](#_Toc129212306)

[4.1 Composants de base d’un diagramme de classes : 32](#_Toc129212307)

[4.2 Autres composants d'un diagramme de classes : 32](#_Toc129212308)

[4.3 Les interactions dans un diagramme de classe : 33](#_Toc129212309)

[4.4 diagramme de class pour l’application web gestion réservation médical : 34](#_Toc129212310)

[4.5 Dictionnaire de class et des attributs : 34](#_Toc129212311)

[5 Le modèle de données relationnel : 37](#_Toc129212312)

[5.1 Règles de passage au modèle relationnel: 37](#_Toc129212313)

[5.2 Model Relationnel : 37](#_Toc129212314)

[6 Conclusion : 39](#_Toc129212315)

[Chapitre 3 :Implémentation System Gestion Réservation Médical 40](#_Toc129212316)

[1 Introduction : 40](#_Toc129212317)

[2 Les outils technique utiliser dans le Projet : 40](#_Toc129212318)

[2.1 IDE Visuel Studio : 40](#_Toc129212319)

[2.2 SQLServer 18 : 40](#_Toc129212320)

[2.3 PostMan : 41](#_Toc129212321)

[3 Les Outille de Développement: 41](#_Toc129212322)

[3.1 .NET : 41](#_Toc129212323)

[3.2 Bootstrap 5 : 41](#_Toc129212324)

[3.3 BlazorWebAssembley : 41](#_Toc129212325)

[3.4 Asp.net Web Api Core : 42](#_Toc129212326)

[3.5 C#: 42](#_Toc129212327)

[3.6 Json: 42](#_Toc129212328)

[4 Architecture d’Application web : 42](#_Toc129212329)

[4.1 L'architecture SPA avec API: 42](#_Toc129212330)

[4.1.1 Architecture SPA Web Assembley : 43](#_Toc129212331)

[4.1.2 Architecture Web Api (Controllers,Services,Manager) : 43](#_Toc129212332)

[5 Les Principes de Programmation Solid : 44](#_Toc129212333)

[6 Implémentation System : 45](#_Toc129212334)

Table de Figure

[Figure 1 : Diagramme de cas d’utilisation « Patient» 17](#_Toc129122994)

[Figure 2: Diagramme de cas d’utilisation «Admin&Médecin» 18](#_Toc129122995)

[Figure 3: Diagramme de cas d’utilisation «Founiseure De Santé» 19](#_Toc129122996)

[Figure 4 : Diagramme de cas d’utilisation «Pharmacien» 19](#_Toc129122997)

[Figure 5: Diagramme de cas d’utilisation «Sécritaire» 20](#_Toc129122998)

[Figure 6:Diagramme de sequence authentificatio 26](#_Toc129122999)

[Figure 7: Diagramme de sequence établire unréservation médicale 27](#_Toc129123000)

[Figure 8: Diagramme de sequence etablire ordre medicale 28](#_Toc129123001)

[Figure 9: Diagramme de sequence Consultation resultat medicale 29](#_Toc129123002)

[Figure 10: Diagramme de sequence consultation ordre medicale par le pharmacien 30](#_Toc129123003)

[Figure 11: Diagramme de sequence admin ajouté medecin 31](#_Toc129123004)

[FIGURE 12:Diagramme De Class Gestion reservation Médicale 34](#_Toc129123005)

# Introduction General

*De nos jours, le monde fait face à une avancée technologique significative dans tous les domaines, principalement grâce à l'informatique qui est une discipline scientifique qui se consacre à l'étude des techniques de traitement automatisé de l’information. Elle aujourd’hui joue un rôle crucial dans le développement des activités professionnelles comme domaine médical*

Avant la création de l'ordinateur, toutes les données étaient consignées manuellement sur des supports en papier, ce qui causait de nombreux problèmes, tels qu'une recherche fastidieuse et chronophage de ces informations, ainsi que la dégradation potentielle de ces dernières.

De nos jours, l'ordinateur demeure le moyen le plus fiable pour le traitement et la conservation de l'information. Grâce à cette innovation, les systèmes de données des entreprises ont été informatisés, ce qui est devenu essentiel pour leur développement actuel.

Les centres médicaux sont des institutions qui peuvent grandement bénéficier de l'informatique. Jusqu'à présent, les processus de gestion manuelle restent largement prédominants, ce qui rend l'introduction de technologies informatiques indispensable pour améliorer la gestion et les processus organisationnels.

Les centres médicaux sont confrontés à plusieurs problèmes, notamment la difficulté de préparer les réservations médicales pour les clients et la complexité de gérer la coordination entre différents cabinets médicaux.

Se rendre chez un médecin est souvent une tâche fastidieuse. Il faut tout d'abord trouver un praticien à proximité de son domicile ou lieu de travail, puis, s'il y a de la chance, patienter en salle d'attente avant de pouvoir enfin consulter.

*Le côté des médecins e*  également confronté à de nombreux défis en ce qui concerne la gestion de leurs rendez-vous. Parmi ces difficultés, on peut citer :

* La complexité de retrouver des fichiers,
* La perte de temps et l'usure des archives en raison d'une utilisation excessive,
* La difficulté à stocker les fiches,
* Le risque de perdre des archives,
* La complexité du calcul des statistiques.

*Le projet de notre mémoire de fin d'études consiste à concevoir et mettre en œuvre une application permettant la gestion des réservations médicales et la gestion de cabinets médicaux privés*

Ainsi, notre mémoire est organisé en trois principaux chapitres :

* Le premier chapitre de notre mémoire, intitulé "Système de réservation en ligne", aborde l'organisation d'accueil, sa structure organisationnelle et quelques détails sur sa constitution. Il contient également des informations sur les applications web (comme les définitions de client/serveur/site...) ainsi que sur l'état informatique de notre application sur Internet, y compris la problématique et les objectifs de notre étude.
* Le deuxième chapitre, "Conception du système de réservation en ligne", est consacré à la conception et à la modélisation de notre application. Il décrit en détail les modèles développés pour le système.
* Dans le troisième chapitre, "Implémentation du système", nous présentons la mise en œuvre de l'application conçue en fonction de l'environnement et du langage de programmation choisis. Nous présentons également quelques interfaces pour illustrer le fonctionnement de l'application développée.

*Une fois toutes les parties abordées, notre mémoire se conclut par une synthèse générale.*

## : System de Réservation en ligne

### Problématique et domaine d’étude:

En raison de la forte affluence de patients dans une variété de spécialités médicales, le centre de santé est contraint de fournir des services à un grand nombre de personnes, ce qui peut entraîner une surpopulation

La surpopulation dans un cabinet privé peut se référer à la situation où un grand nombre de patients cherchent à obtenir des soins de santé auprès d'un même médecin ou d'une même clinique privée, entraînant un engorgement du cabinet et des retards dans les rendez-vous. Cette situation peut être due à divers facteurs tels que la pénurie de médecins dans une région donnée, la concentration de la population dans un secteur géographique particulier, ou encore une forte demande pour les services proposés. Elle peut entraîner des conséquences négatives pour les patients, tels que des temps d'attente prolongés, une qualité de soins réduite et une diminution de la satisfaction des patients. Les médecins et les cliniques peuvent prendre des mesures pour faire face à la surpopulation, comme l'embauche de personnel supplémentaire, l'augmentation des heures d'ouverture, ou encore l'utilisation de systèmes de rendez-vous en ligne pour mieux gérer les flux de patients.

Nous cherchons à créer une application web de gestion de rendez-vous qui permettra d'analyser les temps d'attente des patients et les problèmes qu'ils rencontrent pour prendre des rendez-vous. Cette application mettra également en évidence les difficultés que rencontre le centre médical pour gérer les rendez-vous, en prenant en compte toutes les restrictions possibles lors de la réservation. L'objectif de ce rapport est de se concentrer sur les caractéristiques des applications web, ainsi que sur les outils et technologies utilisés pour développer ce type d'application.

### Les Application web :

Une application web est un programme informatique qui est hébergé sur un serveur et accessible via un navigateur web. Contrairement aux logiciels traditionnels, l'utilisateur n'a pas besoin de les installer sur son ordinateur, mais peut simplement accéder à l'application via son navigateur préféré. Les développeurs d'applications web cherchent aujourd'hui à offrir des fonctionnalités et une expérience utilisateur équivalentes à celles des logiciels traditionnels

### Les types applications web :

#### Application web statique :

**Les applications Web statiques sont caractérisées par un affichage limité d'informations et une faible évolutivité. Elles sont chargées sur le navigateur de l'utilisateur tels qu'elles sont stockées sur le serveur Web et manquent de flexibilité, offrant peu ou pas d'interactivité, conformément à leur nom. (1)**

#### ****Application web dynamique :****

**Une application web dynamique est un type d'application web qui utilise des langages de programmation côté serveur (tels que PHP, Python, Ruby, etc.) pour générer dynamiquement des pages web en fonction des demandes de l'utilisateur. Contrairement aux applications web statiques, les applications web dynamiques peuvent stocker et récupérer des informations à partir d'une base de données, ce qui leur permet de fournir des fonctionnalités plus avancées telles que des formulaires interactifs, des paniers d'achat en ligne, des fonctionnalités de chat en temps réel, etc. Les applications web dynamiques nécessitent un serveur web compatible avec les langages de programmation utilisés et une base de données pour stocker les informations. Elles sont couramment utilisées pour les sites web de commerce électronique, les sites web de médias sociaux, les forums de discussion en ligne, et d'autres sites web qui ont besoin de stocker, récupérer et manipuler des données en temps réel.(1)**

#### ****Application web portail:****

**Une application web portail est un type d'application web qui sert de point d'entrée unique pour accéder à un ensemble de services et d'informations. Les applications web portails agrègent souvent des informations provenant de sources différentes, telles que des sites web, des bases de données, des applications tierces, etc., pour offrir une expérience utilisateur unifiée. Les portails web peuvent également offrir des fonctionnalités avancées telles que l'authentification des utilisateurs, la personnalisation des contenus en fonction de leur profil ou de leur historique de navigation, des services de messagerie interne, etc.**

**Les applications web portails sont largement utilisées dans les entreprises pour fournir un accès centralisé aux applications métier, aux ressources et aux informations. Les portails web peuvent également être utilisés pour créer des sites web communautaires, des intranets d'entreprise, des plateformes de commerce électronique et d'autres types de sites web qui nécessitent un accès à plusieurs sources d'informations et de services. (1)**

#### ****Application web page unique :****

**Une application web à page unique (en anglais, Single-Page Application ou SPA) est un type d'application web qui fonctionne en chargeant dynamiquement toutes les ressources et données nécessaires pour afficher une page web unique, plutôt que de charger plusieurs pages distinctes pour chaque interaction de l'utilisateur. Les SPAs utilisent des technologies de développement côté client telles que JavaScript, Ajax, Angular, React, Vue.js, etc., pour fournir une expérience utilisateur interactive et fluide sans rechargement de la page.(1)**

#### ****Application web progressives:****

Les applications Web progressives (PWA) tirent parti des technologies de pointe intégrées aux navigateurs pour proposer des expériences mobiles comparables à celles des applications natives. Elles offrent des performances rapides et une fiabilité accrue par rapport aux applications Web traditionnelles.(1)

### Principe Fonctionnement application web :

Les applications Web sont des applications qui s'exécutent dans un navigateur web. Elles sont développées en utilisant les langages de programmation web tels que HTML, CSS et JavaScript. Le principe de fonctionnement des applications Web est le suivant :

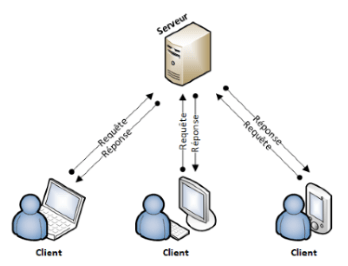
* Le navigateur web demande une page Web au serveur. Le serveur répond en renvoyant la page HTML, qui décrit le contenu de la page.
* Le navigateur analyse la page HTML et récupère les fichiers CSS et JavaScript associés.
* Le navigateur interprète le code HTML, CSS et JavaScript pour afficher la page Web à l'utilisateur.
* L'utilisateur peut interagir avec la page en cliquant sur des liens, des boutons, en remplissant des formulaires, etc.
* En cas d'interaction, le navigateur envoie des requêtes au serveur pour récupérer de nouvelles données. Le serveur répond en renvoyant les données demandées.
* Le navigateur traite les données et met à jour la page Web pour afficher les résultats de l'interaction.

En résumé, les applications Web fonctionnent comme des sites Web, mais avec une interactivité plus poussée et un fonctionnement proche de celui des applications traditionnelles. Les utilisateurs peuvent accéder aux applications Web à partir de n'importe quel navigateur et n'ont pas besoin de télécharger ou d'installer des logiciels supplémentaires sur leur appareil.(2)

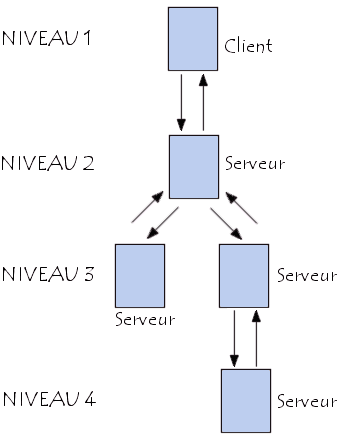
### Architecture application web :

L'architecture des applications Web est la manière dont les différents composants de l'application sont organisés pour assurer une performance, une fiabilité et une évolutivité optimales. Il existe plusieurs modèles d'architecture pour les applications Web, mais voici les trois principaux :

#### L'architecture client-serveur :

c'est l'architecture la plus courante pour les applications Web. Elle se compose d'un client (navigateur) qui envoie des requêtes à un serveur pour récupérer des données, et le serveur renvoie une réponse au client. Cette architecture est simple et facile à maintenir, mais elle peut être limitée en termes de performance et d'évolutivité.(3)

#### L'architecture en couches :

Cette architecture est basée sur une séparation des préoccupations. Les couches sont organisées de manière hiérarchique, chaque couche étant responsable d'une tâche spécifique. Les couches peuvent inclure une couche de présentation, une couche logique et une couche de stockage. Cette architecture offre une meilleure évolutivité et une plus grande flexibilité, mais elle peut être plus complexe à concevoir et à maintenir.(3)

#### L'architecture orientée services :

cette architecture repose sur la mise en place de services Web qui fournissent des fonctionnalités spécifiques. Les services Web sont des applications autonomes qui peuvent communiquer entre elles pour fournir une fonctionnalité plus complexe. Cette architecture est très flexible et peut être utilisée pour des applications de grande envergure, mais elle peut être plus difficile à mettre en œuvre.(3)

### Les outils de développement des applications Web :

#### Les langages de programmation :

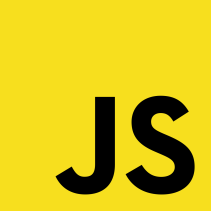
HTML :

Le langage HTML est considéré comme le composant central pour construire une page Web. Il permet de créer et de concevoir des pages et de fournir des instructions au navigateur Internet pour afficher la page correctement. Ainsi, il est essentiel pour les concepteurs de pages Web de maîtriser ce langage pour créer des pages efficaces et fonctionnelles. (3)

css :

 CSS (Cascading Style Sheets) est un langage de feuilles de style utilisé pour décrire la présentation visuelle d'un document HTML ou XML, tel que les couleurs, les polices de caractères, la mise en page et les effets visuels.

CSS est un langage de balisage indépendant de la plate-forme, ce qui signifie que les styles sont rendus de la même manière sur toutes les plates-formes et dans tous les navigateurs modernes. Les versions récentes de CSS incluent également des fonctionnalités pour créer des mises en page complexes, telles que les grilles et les flexbox, ainsi que des effets animés et des transitions visuelles.(3)

Javascript :

JavaScript est un langage de programmation interprété, léger et orienté objet, principalement utilisé pour développer des applications Web interactives côté client. Il a été développé en 1995 par Brendan Eich de Netscape et est devenu un standard de facto pour le développement Web.(3)

#### La Base de données :

Une base de données est une collection organisée de données qui sont stockées de manière persistante, gérées et accessibles par des applications logicielles. Les données peuvent être organisées sous forme de tableaux, de graphiques, de documents ou de tout autre format approprié.

Les bases de données sont utilisées pour stocker des informations sur des entreprises, des personnes, des produits, des transactions et bien d'autres choses encore. Elles permettent aux applications d'accéder aux données de manière rapide et efficace, de les manipuler et de les modifier selon les besoins.

Il existe plusieurs types de bases de données, tels que les bases de données relationnelles, les bases de données NoSQL, les bases de données graphes et les bases de données en mémoire. Les bases de données relationnelles sont les plus courantes et utilisent des tables pour stocker des données, tandis que les bases de données NoSQL utilisent des structures de données plus flexibles telles que des documents ou des graphes pour stocker des informations.

Les bases de données sont souvent gérées par des systèmes de gestion de bases de données (SGBD), qui permettent de créer, de maintenir, de sauvegarder et de récupérer des données de manière efficace. Les SGBD courants incluent MySQL, Oracle, Microsoft SQL Server et MongoDB, entre autres. (3)

#### IDE :

Un IDE (Integrated Development Environment) est un environnement de développement intégré qui fournit aux développeurs des outils et des fonctionnalités pour faciliter le processus de développement de logiciels.

Un IDE est conçu pour simplifier et accélérer la création, la compilation, le débogage et le déploiement de code source. Il comprend généralement un éditeur de code source, un compilateur, un débogueur, des outils de gestion de versions, des outils de test et des outils de déploiement.

### Organigramme général d’un cabinet privé : le cabinet privé il comporte plusieurs services médicaux que nous allons présenter :

#### Service technique et exploration:

* Laboratoire d’analyse médicale :

Cet endroit est dédié au prélèvement et à l'analyse de différents fluides biologiques d'origine humaine, placés sous la supervision de biologistes médicaux. Ces derniers sont chargés d'interpréter les résultats obtenus dans le but de contribuer au diagnostic et au suivi de certaines maladies.

* Radiologie et imagerie médicale :

Dans le domaine médical, la radiologie englobe l'ensemble des techniques diagnostiques et thérapeutiques qui utilisent des rayons X ou d'autres types de rayonnements.

#### Services médicaux:

Les services médicaux se réfèrent à l'ensemble des prestations de soins de santé qui sont offertes dans un établissement de santé ou une institution médicale. Les services médicaux peuvent inclure des consultations médicales, des examens médicaux, des traitements, des interventions chirurgicales, des soins infirmiers, des soins de réadaptation et d'autres services connexes.

### Problématiques et objectives:

#### Problématiques:

Voici quelques problématiques qui pourraient être abordées dans le cadre d'un projet de développement d'une application de gestion de patients pour un cabinet médical :

* Comment concevoir une application conviviale et facile à utiliser pour les professionnels de santé et les patients, tout en garantissant la sécurité et la confidentialité des données médicales sensibles ?
* Comment gérer efficacement les dossiers médicaux électroniques des patients, y compris la collecte, l'enregistrement, la mise à jour et la sécurisation des informations médicales, tout en garantissant la conformité aux normes réglementaires en matière de protection de la vie privée et de sécurité des données ?
* Comment faciliter la communication et la collaboration entre les professionnels de santé dans le cadre de la prise en charge des patients, tout en garantissant l'intégrité et la confidentialité des données médicales ?
* Comment assurer la fiabilité et la précision des résultats d'analyses et des radiographies partagées entre les professionnels de santé et les fournisseurs de services de santé ?
* Comment optimiser la gestion des rendez-vous, des listes d'attente et des rappels pour les patients, tout en garantissant la satisfaction des patients et la maximisation de l'efficacité du cabinet médical ?
* Comment gérer les ordonnances électroniques, y compris la prescription, la délivrance et le suivi des médicaments, tout en garantissant la sécurité et la qualité des soins aux patients ?

#### Objectifs de Notre travail :

Voici quelques objectifs qui pourraient être visés dans le cadre d'un projet de développement d'une application de gestion de patients pour un cabinet médical

* Optimiser la gestion des rendez-vous
* Renforcer la sécurité des données
* Faciliter la collaboration
* Améliorer la satisfaction des patients

### Présentation le Projet:

Nous avons pour objectif de concevoir une application web visant à simplifier la gestion des patients pour un cabinet médical. Cette mission implique la définition des responsabilités de gestion, la mise à jour régulière des données, l'organisation des informations collectées pour élaborer des fichiers de base à l'usage du médecin, ainsi que le renforcement du contrôle et de la confrontation des données. Notre but est d'assurer une gestion médicale optimale des patients en garantissant la cohérence des informations, tout en facilitant le travail des responsables. Les principales fonctionnalités de notre application seront les suivantes :

#### Gestion et suivi du dossier médical des patients :

Notre application fournira également une fonctionnalité de gestion et de suivi des dossiers médicaux des patients. Cela inclura la collecte et l'enregistrement de toutes les informations médicales pertinentes, telles que les antécédents médicaux, les résultats de tests, les prescriptions, les rapports de consultations et autres informations importantes. Les données seront stockées de manière sécurisée et organisée pour permettre un accès facile et rapide aux informations médicales des patients par les professionnels de santé. Le but est de garantir une gestion efficace et précise des dossiers médicaux des patients, de faciliter la communication entre les professionnels de santé et d'améliorer la qualité des soins médicaux.

#### Gestion des Rendez-vous des patients :

Notre application offrira une fonctionnalité de gestion des rendez-vous pour les patients du cabinet médical. Cela inclura la possibilité établir un rendez-vous, d'envoyer des rappels aux patients pour leur rappeler la date et l'heure de leur rendez-vous, ainsi que de gérer les annulations et les reports de rendez-vous. Le but est de simplifier la planification des rendez-vous pour les patients

#### Gestion le résultat laboratoire analyse et radiologie:

Notre application inclura également une fonctionnalité de gestion des résultats d'analyses et des radiographies. Les professionnels de santé pourront facilement télécharger et enregistrer les résultats d'analyses de laboratoire et les radiographies des patients dans leur dossier médical électronique. Ces données pourront ensuite être consultées et analysées à tout moment pour aider les professionnels de santé à diagnostiquer et traiter les patients. Le but est d'améliorer l'efficacité du diagnostic médical en garantissant un accès facile et rapide aux résultats d'analyses et aux radiographies des patients, tout en préservant la confidentialité et la sécurité des données.

#### Gestion des ordonnances et médicaments :

Notre application permettra également la gestion des ordonnances et l'analyse des données médicales des patients. Les professionnels de santé pourront facilement prescrire des médicaments aux patients et générer des ordonnances électroniques qui seront directement envoyées aux pharmacies. Les données des ordonnances seront également stockées de manière sécurisée et organisée pour permettre un suivi précis des prescriptions et une analyse ultérieure des habitudes de prescription. Les professionnels de santé pourront ainsi surveiller l'efficacité des traitements, les effets secondaires des médicaments et les interactions médicamenteuses potentielles. Le but est d'améliorer la qualité des soins médicaux et de garantir une gestion efficace des prescriptions pour les patients du cabinet médical.

### Analyse des besoins:

Voici une liste acteur et les fonctionnalités que pourrait offrir notre application de gestion de patients pour un cabinet médical:

#### Les besoins fonctionnels :

* Gestion des rendez-vous pour les patients
* Gestion et suivi des dossiers médicaux des patients
* Gestion des ordonnances et des prescriptions médicales
* Gestion et analyse des résultats d'analyses et des radiographies
* Communication interne entre les professionnels de santé
* Envoi de rappels aux patients pour les rendez-vous et les prescriptions
* Suivi des antécédents médicaux et des traitements pour les patients

#### Les besoins non fonctionnels:

* Gestion des factures et des paiements pour les patients
* Gestion des demandes de renouvellement de prescriptions
* Gestion des listes d'attente pour les patients sans rendez-vous
* Génération de rapports et d'analyses pour les professionnels de santé
* Gestion des documents d'assurance et des réclamations de remboursement

### Les Application web Similaire :

#### ZocDoc :

Zocdoc est une application de gestion de rendez-vous médicaux en ligne qui permet aux patients de trouver et de réserver des rendez-vous avec des médecins et des professionnels de la santé dans leur région. L'application propose également des fonctionnalités telles que la consultation de la disponibilité des médecins, la lecture des commentaires et des évaluations des patients, la confirmation de rendez-vous, la gestion de la file d'attente et la communication avec les médecins.

Voici comment fonctionne l'application Zocdoc :

* **Recherche de médecins :** Les patients peuvent rechercher des médecins et des professionnels de la santé en fonction de leur emplacement, de leur spécialité et de leur assurance. Ils peuvent également lire des commentaires et des évaluations d'autres patients pour aider à choisir un médecin.
* **Planification de rendez-vous :** Une fois qu'un patient a trouvé un médecin qui répond à ses besoins, il peut planifier un rendez-vous en ligne. Les patients peuvent voir les créneaux horaires disponibles pour chaque médecin, choisir un horaire qui convient et réserver leur rendez-vous.
* **Confirmation de rendez-vous :** Les patients reçoivent des rappels par e-mail, SMS ou notification push pour confirmer leur rendez-vous. Si le patient doit annuler ou reprogrammer le rendez-vous, il peut le faire directement depuis l'application.
* **Gestion de la file d'attente :** Les patients peuvent voir leur place dans la file d'attente et recevoir des notifications en temps réel pour se préparer à voir le médecin.
* **Communication avec les médecins :** Les patients peuvent envoyer des messages aux médecins avant ou après leur rendez-vous pour poser des questions ou pour obtenir des informations complémentaires.
* **Paiement en ligne :** Zocdoc permet également aux patients de payer en ligne pour les rendez-vous ou les services fournis par la clinique.

#### DocToLib :

Doctolib est une application de gestion de rendez-vous médicaux en ligne qui permet aux patients de rechercher des médecins et de prendre rendez-vous en ligne. Cette application est très populaire en France et est également disponible dans d'autres pays européens. Cette application offrir tous les fonctionnalité de l’application zocdoc on plus les gestion de la facturation et de gestion du carnet de santé pour offrir une expérience utilisateur pratique et efficace pour les patients.

### Langage de modélisation UML:

L'UML (Unified Modeling Language) est un langage de modélisation visuelle qui a été conçu pour être universel et riche sur le plan sémantique et syntaxique. Il est utilisé pour l'architecture, la conception et l'implémentation de systèmes logiciels complexes en décrivant leur structure et leur comportement de manière visuelle et intuitive. Bien que l'UML soit principalement destiné au développement de logiciels, il peut être utilisé pour des applications allant au-delà de ce domaine, comme la modélisation des flux de processus dans l'industrie.

Il existe plusieurs types de diagrammes UML qui sont couramment utilisés pour représenter différents aspects d'un système logiciel. Voici quelques exemples : (4)

**Diagramme de cas d'utilisation :** Il est utilisé pour décrire les interactions entre les acteurs et le système. Il permet de représenter les fonctionnalités du système du point de vue des utilisateurs finaux.

**Diagramme de classes :** Il est utilisé pour représenter les classes, les attributs, les méthodes et les relations entre les objets du système. Il aide à comprendre la structure du système.

**Diagramme de séquence :** Il est utilisé pour modéliser l'interaction entre les objets du système dans une séquence chronologique. Il permet de représenter le flux d'exécution des tâches ou des processus.

**Diagramme d'état-transition :** Il est utilisé pour représenter les états d'un objet et les transitions entre ces états. Il permet de modéliser les comportements des objets dans différentes situations.

**Diagramme de déploiement :** Il est utilisé pour représenter l'architecture physique du système et la disposition des composants matériels et logiciels. Il permet de comprendre la façon dont le système sera déployé sur le matériel.

### Conclusion:

En conclusion, un système de réservation médicale en ligne et de gestion de dossier médical est un outil essentiel pour les établissements de santé modernes. En combinant la possibilité de prendre rendez-vous en ligne et la gestion des dossiers médicaux des patients, ce système permet d'améliorer considérablement l'efficacité et la qualité des soins. Les patients peuvent facilement prendre rendez-vous en ligne, ce qui permet de réduire les temps d'attente et de gérer leur dossier médical plus efficacement. Les médecins peuvent ainsi mieux organiser leur emploi du temps et accéder rapidement aux informations nécessaires pour le diagnostic et le traitement des patients. Les établissements de santé peuvent mieux gérer leur activité en planifiant et en organisant les rendez-vous médicaux

## : Conception System

### Introduction :

La phase de conception est cruciale dans le cycle de vie d'un projet car elle détermine en grande partie la qualité de l'application produite. Ce chapitre se concentre sur cette étape et vise à définir les rôles de chaque acteur qui interagit avec le système. Nous utiliserons des diagrammes de cas d'utilisation pour modéliser ces rôles, puis nous élaborerons des cas d'utilisation, des cas de séquence et des cas de classe. Enfin, nous aborderons l'élaboration de la base de données.

### Diagramme De Cas D’Utilisation :

En langage de modélisation unifié (UML), un diagramme de cas d'utilisation peut servir à résumer les informations des utilisateurs de votre système (également appelés acteurs) et leurs interactions avec ce dernier. La création de ce type de diagramme UML requiert un ensemble de symboles et de connecteurs spécifiques(6)

#### Acteur :

Un acteur est une personne ou un système extérieur au système en cours de modélisation qui interagit avec notre système.(5)

#### Relations entre les cas d’utilisation:

Les relations dans un diagramme de cas d'utilisation peuvent être représentées de plusieurs façons :

* **Association :** une association relie un acteur à un cas d'utilisation. Elle indique que l'acteur participe à l'utilisation de ce cas.
* **Inclusion :** une inclusion est utilisée pour éviter la répétition de cas d'utilisation. Elle permet d'inclure les actions réutilisables dans un autre cas d'utilisation.
* **Extension :** une extension est utilisée pour décrire un scénario alternatif dans un cas d'utilisation. Elle permet d'étendre les fonctionnalités de base du cas d'utilisation.
* **Généralisation :** une généralisation est utilisée pour regrouper les cas d'utilisation en fonction de leurs caractéristiques communes. Elle permet de définir des cas d'utilisation génériques et des cas d'utilisation spécifiques.

#### Les Acteurs de system Réservation en ligne :

Dans un système de gestion de patients pour un cabinet médical, on peut identifier plusieurs acteurs clés qui interagissent avec l'application. Voici quelques-uns de ces acteurs :

* **Les professionnels de santé :** Ce sont les utilisateurs principaux de l'application. Ils peuvent être des médecins Général, des dentistes, des physiothérapeutes ou tout autre professionnel de santé travaillant dans le cabinet médical. Ils utilisent l'application pour accéder aux dossiers médicaux électroniques des patients, gérer les rendez-vous, prescrire des médicaments, analyser les résultats d'analyses qui été publier par les fournisseurs de services de santé.
* **Secrétaire :** La secrétaire est un acteur central dans la relation entre le médecin, le patient et l'institution. En tant que collaboratrice directe du médecin et principal point de contact du patient, la secrétaire occupe une position stratégique dans le fonctionnement de toute organisation médicale. Les tâches de la secrétaire peuvent varier d'une structure à l'autre, mais elles sont essentielles pour assurer une gestion efficace du système de réservation médicale
* **Les patients :** Les patients sont les destinataires des soins médicaux fournis par le cabinet médical. Ils peuvent utiliser l'application pour prendre des rendez-vous, consulter leur résultat médical, recevoir des rappels de rendez-vous, etc.
* **Les pharmaciens :** Les pharmaciens peuvent utiliser l'application pour recevoir les ordonnances électroniques des professionnels de santé et Valider pour préparer les médicaments des patients.
* **Chef de clinique:** Le personnel administratif travaille en coulisses pour aider à maintenir le bon fonctionnement du cabinet médical, peut être un médecin. Ils peuvent utiliser l'application pour gérer les listes d'attente et planifier les rendez-vous.
* **Les fournisseurs de services de santé :** Les fournisseurs de services de santé, tels que les laboratoires d'analyses médicales ou les services d'imagerie, peuvent utiliser l'application pour partager les résultats des analyses et des radiographies avec les professionnels de santé du cabinet médical.

#### Identifié les Cas d’Utilisation Selon Acteur :

|  |  |  |
| --- | --- | --- |
| Numéro | Acteur | Les Cas d’Utilisation |
| 1 | Patient/(tous les Acteur) | * Authentification * Rechercher un Médecin * Voir détaille de Réservation Médical * Confirmer le Réservation * Consulté liste de Rendez-vous * Consulter Résultat Rendez-vous * Consulter les Notification des rendez-vous |
| 2 | (Admin Cabinet/Médecin) | * Authentification * Consulter dossier malade * Etablir ordre médical * Accepter demande joint Cabinet * Consulter résultat de fournisseur Service Santé * Consulter List d’attente patiente |
| 3 | Admin Cabinet | * Ajouté Médecin au cabinet * Ajouté secrétaire au cabinet * Spécifier information au cabinet * Contrôler Liste Réservation * Gérer plage horaire au médecin |
| 4 | Pharmaceutique | * authentification * Consulter ordonnance * Validez ordonnance |
| 5 | Fournisseur service Santé | * consulter dossier malade * consulté ordre médecin * valider ordre médecin * partager le résultat de consultation au dossier malade |
| 6 | Secrétaire | * authentification * consulter List d’attente de patient réservé aux médecin de cabinet * ajouté malade au List attente de patient * Consulter l’Ordre de médecin * Valider Payement de patient |

#### Diagramme Cas d’utilisation pour le Patient :

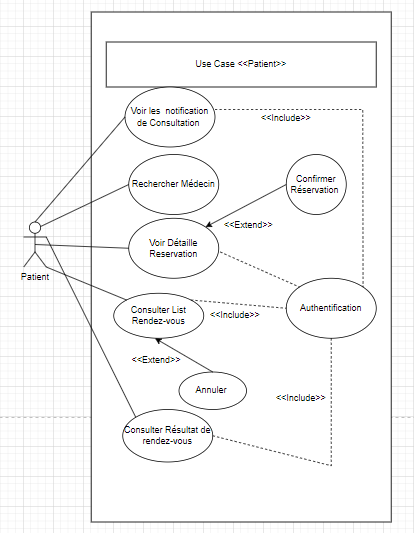


Figure 1 : Diagramme de cas d’utilisation « Patient»

#### Diagramme Cas d’utilisation pour le Médecin  et Admin de cabinet:

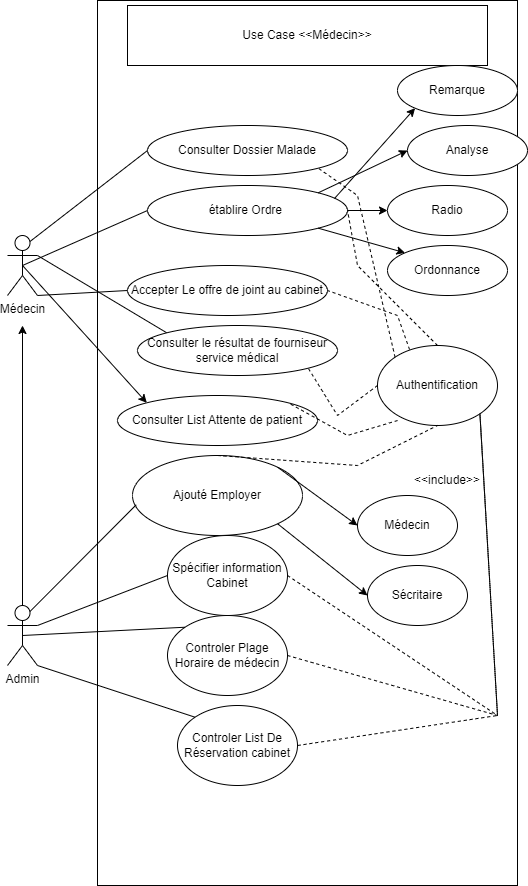


Figure 2: Diagramme de cas d’utilisation «Admin&Médecin»

#### Diagramme Cas d’utilisation pour fournisseur service de santé :

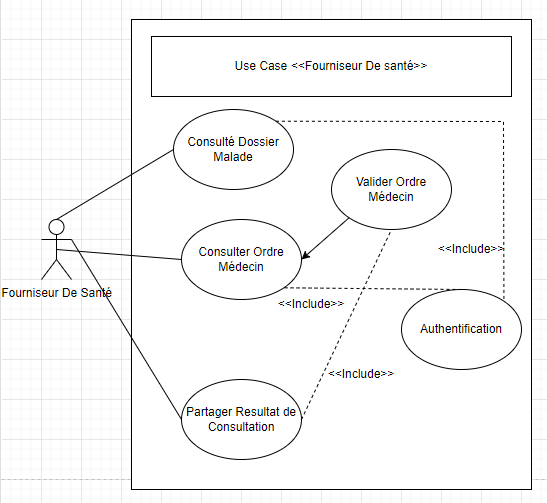


Figure 3: Diagramme de cas d’utilisation «Founiseure De Santé»

#### Diagramme Cas d’utilisation pour le pharmacien :

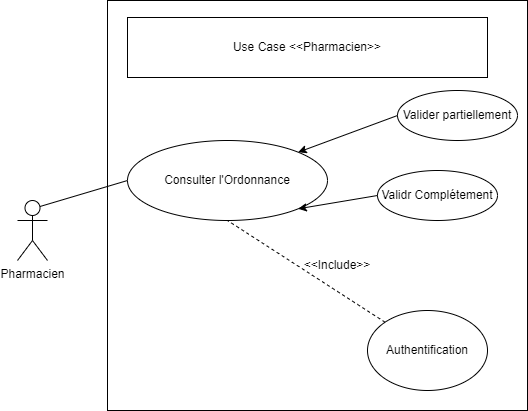


Figure 4 : Diagramme de cas d’utilisation «Pharmacien»

#### Diagramme Cas d’utilisation pour le secrétaire:

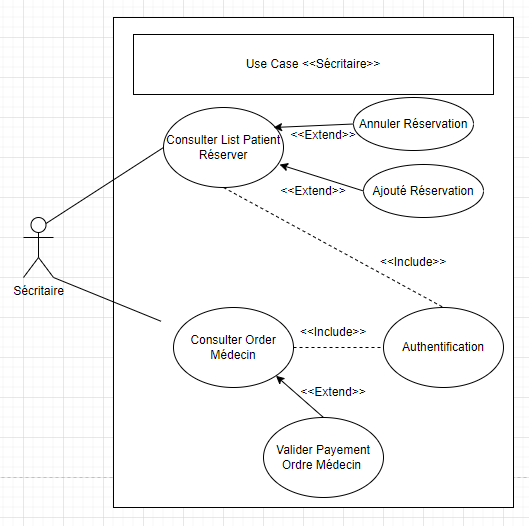


Figure 5: Diagramme de cas d’utilisation «Sécritaire»

#### Description textuel des cas d’utilisation:

* **Authentification :**

|  |  |
| --- | --- |
| Acteur Principale | (Patient|Médecin|admin|founisseur de santé|Sécritaire|pharmacien) |
| Pré condition | Utilisateur été créer un compte |
| Objectif | Authentification pour utiliser les fonctionnements d’application |
| pré condition | Utilisateur il été créé un compte |
| Scénario nominal | 1. Utilisateur accéder à l’application 2. System afficher l’interface d’authentification 3. Utilisateur saisir email avec le mot de passe [A] 4. System Vérifier l’authentification d’utilisateur 5. System retourner la page qui été favorable |
| Alternative | 1. Dans le cas authentification il est pas valide ou manque de saisir le system retourner a l’étape 2 et afficher un erreur |

* **Etablit un Réservation Médical :**

|  |  |
| --- | --- |
| Acteur Principale | (Patient) |
| Objectif | Occupé un rendez-vous avec médecin |
| pré condition | Authentification |
| Scénario Nominal | 1. Utilisateur accéder à l’application 2. System afficher l’interface d’accueille 3. Utilisateur établit un Recherche [A] 4. System afficher List disponibles des médecins 5. Utilisateur choisir le médecin favorable 6. System afficher l’information de médecin et le cabinet avec option pour réserver médical 7. Utilisateur réserver un rendez-vous 8. System afficher le détaille de rendez-vous 9. Utilisateur accepter le rendez-vous[A] 10. Le système valide l’opération |
| Alternative | 1. Si le nom de médecin n’existe pas dans le system alors affiche List vide de médecin 2. Si l’utilisateur annuler le réservation le system retourné à la étape précédent 6 |

* **Préparer ordre médicale par le médecin :**

|  |  |
| --- | --- |
| Acteur Principale | (Médecin|admin) |
| pré condition | Authentification, Patient faire un Réservation médical |
| Objectif | Préparer un ordonnance pour le Patient |
| Scénario Nominal | 1. Médecin accéder à l’application 2. System afficher l’interface d’accueille 3. Le médecin consulter la liste d’attente de patient 4. Le System afficher tous les patient qui été réserver [A] 5. Le médecin Consulter l’information de patient 6. Le system afficher tout l’information de patient avec option ajouté un ordre médical ou consulter le dossier médical de patient [A] 7. Le médecin choisir ajouté un ordre médical 8. Le system afficher les type d’ordre médical (radio, analyse, médicament, médicament + analyse, médicament +radio,radio+analyse) 9. Le Médecin choisir le type d’ordre médical 10. Le system afficher le formulaire selon leur choix 11. Le médecin remplir le formulaire [A] 12. Le médecin confirmer l’ordre médical 13. Le system enregistre le ordre médical |
| Alternative | 1. Si la liste de patient été vide alors le system afficher rien sinon afficher liste de patient réserver 2. Si l’utilisateur choisir consulter dossier malade le system afficher historique médicale patient 3. Si le médecin oublier le saisir donnée de patient alors le system afficher un erreur et reprend l’étape 10 |

* **Médecin Consulter les Résultat de fournisseur de service de Santé:**

|  |  |
| --- | --- |
| Acteur Principale | (Médecin|admin) |
| pré condition | Authentification, Fournisseur de Santé Valider ordre médicale de médecin |
| Objectif | Préparer résultat de consultation Médical |
| Scénario Nominal | 1. Médecin accéder à l’application 2. System afficher l’interface d’accueille 3. Médecin Accéder à boit de notification 4. Le system afficher tous les ordres médical leur patient qui été valider par le fournisseur de santé [A] 5. Le médecin choisir le résultat de consultation 6. Le system afficher le rapport médical qui été établit par le fournisseur de santé 7. Le médecin envoyer le rapport médical avec leur remarque a Patient qui été concerné 8. Le system valider l’opération |
| Alternative | 1. Si le Boit de notification il est vide alors le system affiche rien |

* **le fournisseur de service de Santé consulter ordre médical:**

|  |  |
| --- | --- |
| Acteur Principale | (Fournisseur de santé) |
| pré condition | Authentification, reçu un bilan médical de patient, rôle fournisseur de santé, ordre médical il est pas valider par autre Fournisseur de santé |
| Objectif | Préparer une ordonnance pour le Patient |
| Scénario Nominal | 1. fournisseur de santé accéder à l’application 2. System afficher l’interface d’accueille 3. fournisseur de santé entrer le numéro de série de ordre médical [A] 4. le system afficher le détaille d’ordre médical 5. fournisseur de santé établit un résultat et partager avec le médecin qui émis l’ordre médical [A] 6. le system envoyer notification a médecin et valide l’opération |
| Alternative | 1. Si le saisir numéro série de ordre médical il est incorrect alors afficher message d’erreur 2. Si le fournisseur de santé n’est pas télécharger le résultat et valide l’ordonnance alors le system afficher erreur |

* **le pharmacien consulter ordonnance médical:**

|  |  |
| --- | --- |
| Acteur Principale | (Pharmacien) |
| pré condition | Authentification, rôle pharmacien, reçu un ordonnance médical de patient, l’ordonnance médical il est pas valider par autre pharmacie |
| Objectif | Préparer une ordonnance pour le Patient |
| Scénario Nominal | 1. pharmacien accéder à l’application 2. System afficher l’interface d’accueille 3. pharmacien entrer le numéro de série de l’ordre médical [A] 4. le system afficher le détaille d’ordre médical 5. pharmacien valider le ordonnance partiellement ou complet [A] 6. le system valider l’ordonnance et envoyer notification |
| Alternative | 1. Si l’ordonnance médical été valider par autre pharmacien ou le code d’ordonnance il été invalide alors le system afficher message d’erreur 2. Si le pharmacien valider le ordonnance complètement le system valider l’ordonnance sinon le system valider partiellement l’ordonnance |

* **Admin de cabinet ajouté un Médecin:**

|  |  |
| --- | --- |
| Acteur Principale | (Admin de Cabinet) |
| pré condition | Authentification, rôle Admin Cabinet, le médecin a il est pas relier avec autre cabinet |
| Objectif | Ajouté employeur médecin a system de cabinet |
| Scénario Nominal | 1. Admin accéder à l’application 2. System afficher l’interface d’accueille 3. Admin rechercher a médecin a partir votre email [A] 4. le system vérifie que médecin pas travaillé avec autre cabinet [A] 5. le system afficher l’information de médecin 6. Admin envoyer demande a médecin correspondant [A] 7. le system envoyer un notification a médecin correspondant et valide l’opération |
| Alternative | 1. Si le email de médecin pas correct le system afficher List vide d’information sinon afficher l’information de médecin correspondant 2. Si le médecin travailler avec autre cabinet alors le system afficher message d’erreur a l’utilisateur sinon afficher information de médecin 3. Si le médecin accepter le demande joint aux cabinet alors le le system va ajouté le compte médecin a cabinet |

* **Médecin Crée dossier malade:**

|  |  |
| --- | --- |
| Acteur Principale | (Médecin) |
| pré condition | Authentification, rôle médecin, patient n’est pas contient un dossier médicament |
| Objectif | Ajouté employeur médecin a system de cabinet |
| Scénario Nominal | 1. médecin accéder à l’application 2. System afficher l’interface d’accueille 3. Médecin choisir option ajouté dossier malade 4. le system afficher un formulaire 5. le médecin remplir le formulaire [A] 6. le médecin enregistre dossier malade 7. le system vérifier que le malade il n’est pas contient un dossier malade après valide l’opération [A] |
| Alternative | 1. si le saisir de médecin incomplet alors le system afficher un message erreur 2. si le numéro identification de patient il est existé dans les dossiers patients alors le system affiché une erreur |

* **Médecin Consulter dossier malade:**

|  |  |
| --- | --- |
| Acteur Principale | (Médecin) |
| pré condition | Authentification, rôle (Médecin|Fourniseur De santé) , patient contient un dossier médicament |
| Objectif | Ajouté employeur médecin a system de cabinet |
| Scénario Nominal | 1. utilisateur accéder à l’application 2. System afficher l’interface d’accueille 3. utilisateur choisir option consulter dossier médical 4. le system afficher un formulaire qui contient information identifier le Patient 5. utilisateur remplirai le formulaire de recherche [A] 6. le system recherché à patient 7. system afficher le dossier médical de patient |
| Alternative | 1. Si existe un manque de saisir alors le system afficher message d’erreur 2. Si le system n’est pas trouver le dossier médical de patient alors le system afficher un erreur a l’utilisateur |

* **Secrétaire valider le payement de patient :**

|  |  |
| --- | --- |
| Acteur Principale | Secrétaire |
| pré condition | Authentification, rôle Secrétaire, médecin Délivrer un document médical |
| Objectif | Valider payement de l’ordre médical |
| Scénario Nominal | 1. Secrétaire accéder à l’application 2. System afficher l’interface d’accueille 3. Secrétaire accéder a liste d’ordre médical non validé selon le médecin de cabinet 4. System afficher List des ordre médical qui été délivrer [A] 5. Secrétaire choisir un ordre médical ensuit consulter 6. Secrétaire classé l’ordre médical dans un List ordre médical valide [A] 7. Le system valider l’opération et envoyer l’ordre médical a client à travers email utilisateur |
| Alternative | 1. Si la List d’ordre médical il est vide afficher rien 2. Si le secrétaire ne pas valider l’ordre médical alors le system reste l’ordre médical dans un List non validé |

### Diagramme de Séquence :

Un diagramme de séquence est un type de diagramme de modélisation de système, il est utilisé pour représenter la séquence des interactions entre différents objets d'un système logiciel ou matériel.

Les composants de base d'un diagramme de séquence sont :

**Les objets :** Ils représentent les entités qui interagissent entre elles dans le système modélisé. Chaque objet est représenté par une ligne de vie verticale qui montre la durée de vie de l'objet dans le scénario modélisé.

**Les messages :** Ils représentent les interactions entre les objets. Ils sont représentés par des flèches dirigées de l'objet émetteur vers l'objet récepteur. Les messages peuvent être synchrones ou asynchrones, selon que l'émetteur attend ou non une réponse du récepteur.

**Les activations :** Elles représentent la durée pendant laquelle un objet est actif pour traiter un message reçu. Elles sont représentées par des barres verticales sur la ligne de vie de l'objet.

**Les fragments combinés :** Ils sont utilisés pour représenter des structures de contrôle de flux dans un diagramme de séquence. Ils permettent de modéliser des structures telles que les boucles et les branchements conditionnels.

**Les focus d'interaction :** Ils sont utilisés pour indiquer les messages les plus importants dans un diagramme de séquence. Les focus d'interaction sont représentés par une boîte en pointillés qui entoure les messages les plus importants.

**Les commentaires :** Ils permettent de fournir des explications supplémentaires sur le scénario modélisé. Les commentaires sont représentés par des notes qui peuvent être attachées aux différents composants du diagramme de séquence.

#### Déférents diagramme des sequences:

#### diagramme des sequence pour Authentification:

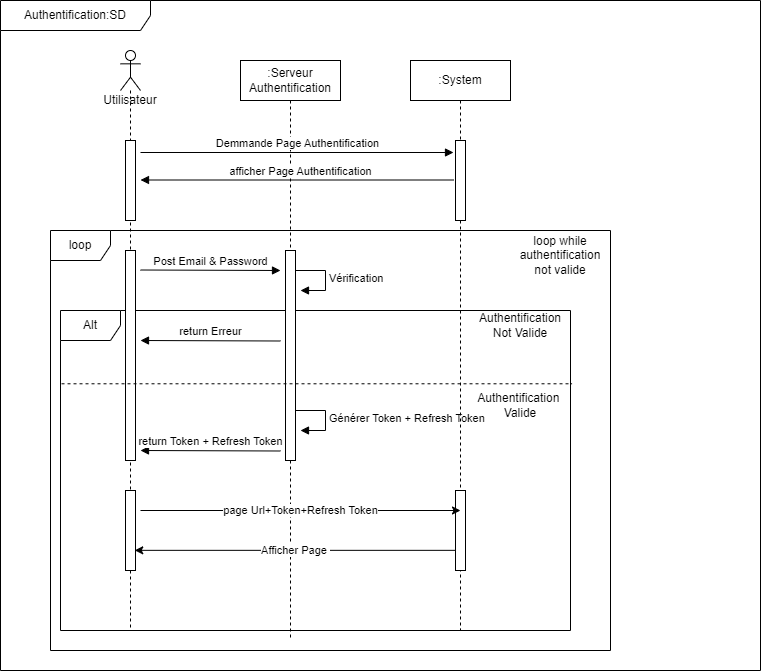


Figure 6:Diagramme de sequence authentificatio

#### diagramme des séquence pour Établire un réservation médical:

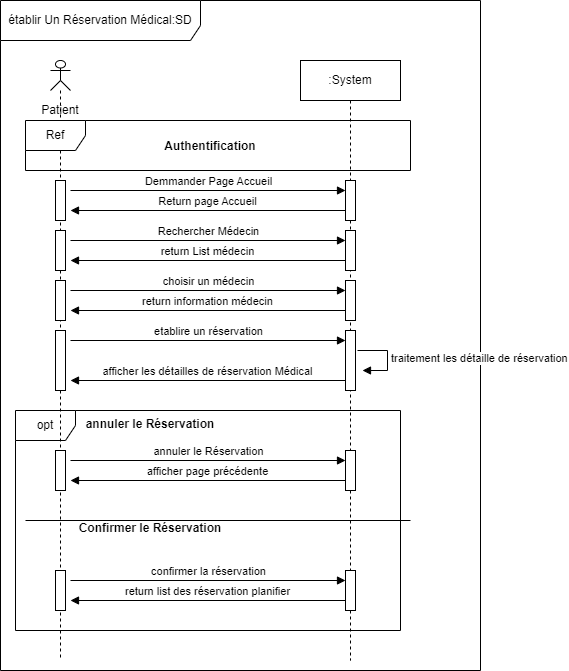


Figure 7: Diagramme de sequence établire unréservation médicale

#### diagramme des sequence Établir un ordre médical:

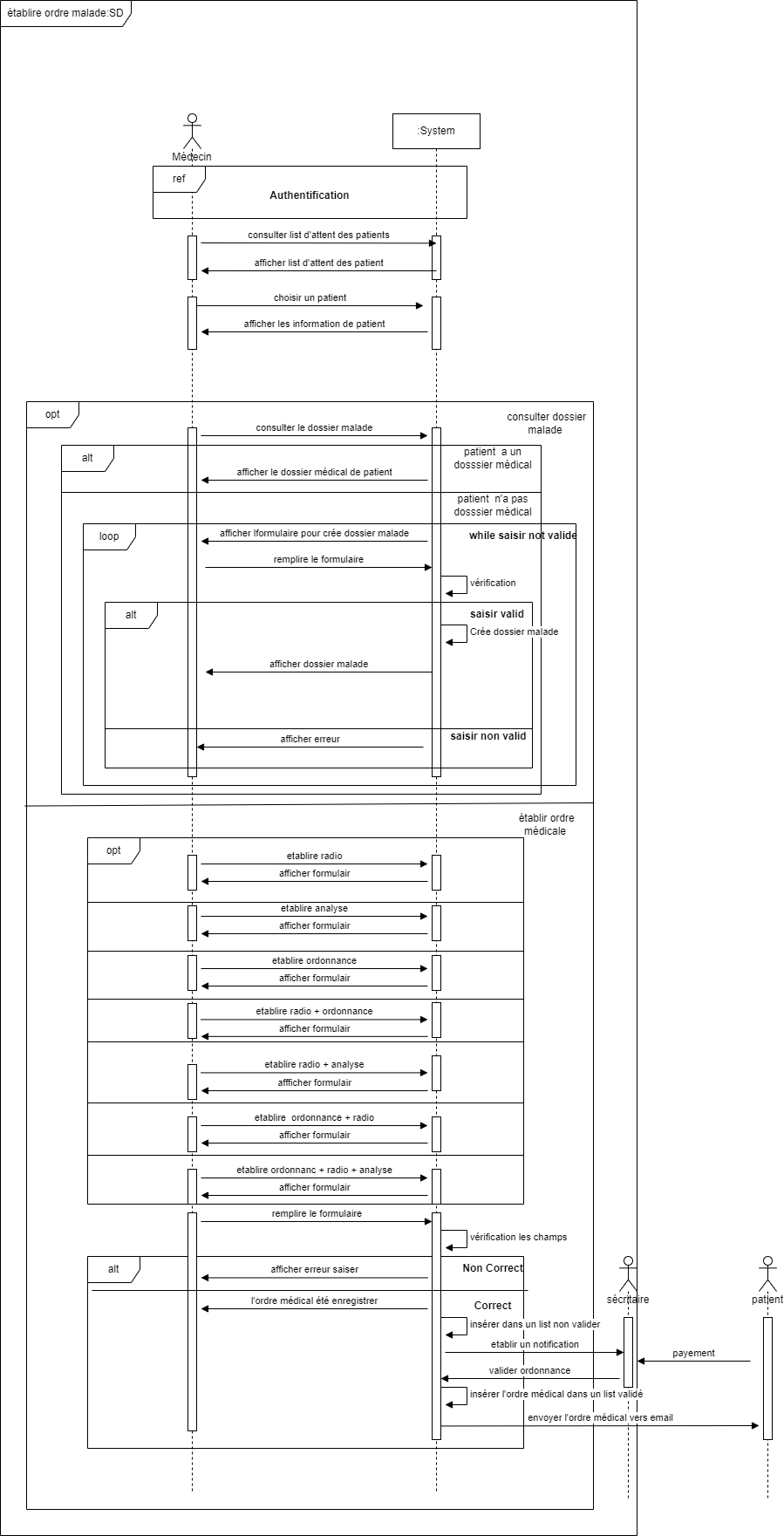


Figure 8: Diagramme de sequence etablire ordre medicale

#### diagramme des sequence Consultation résultat médical:

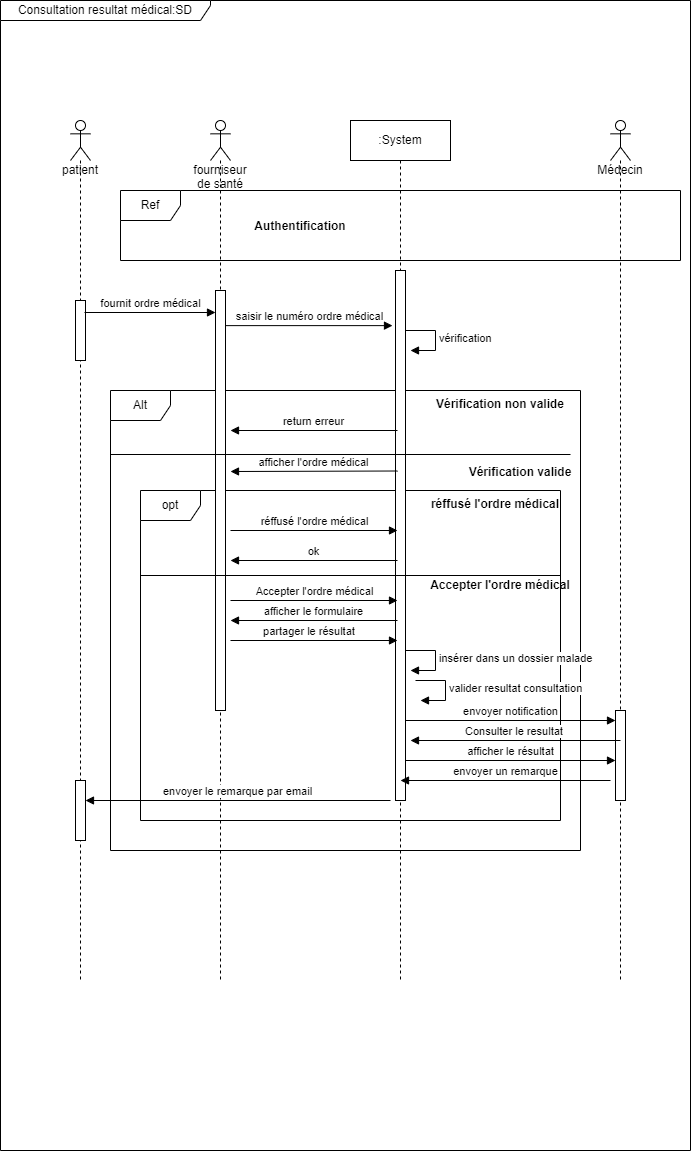
****

Figure 9: Diagramme de sequence Consultation resultat medicale

#### diagramme des séquence pour consulter l’ordonnance médical par le pharmacien:

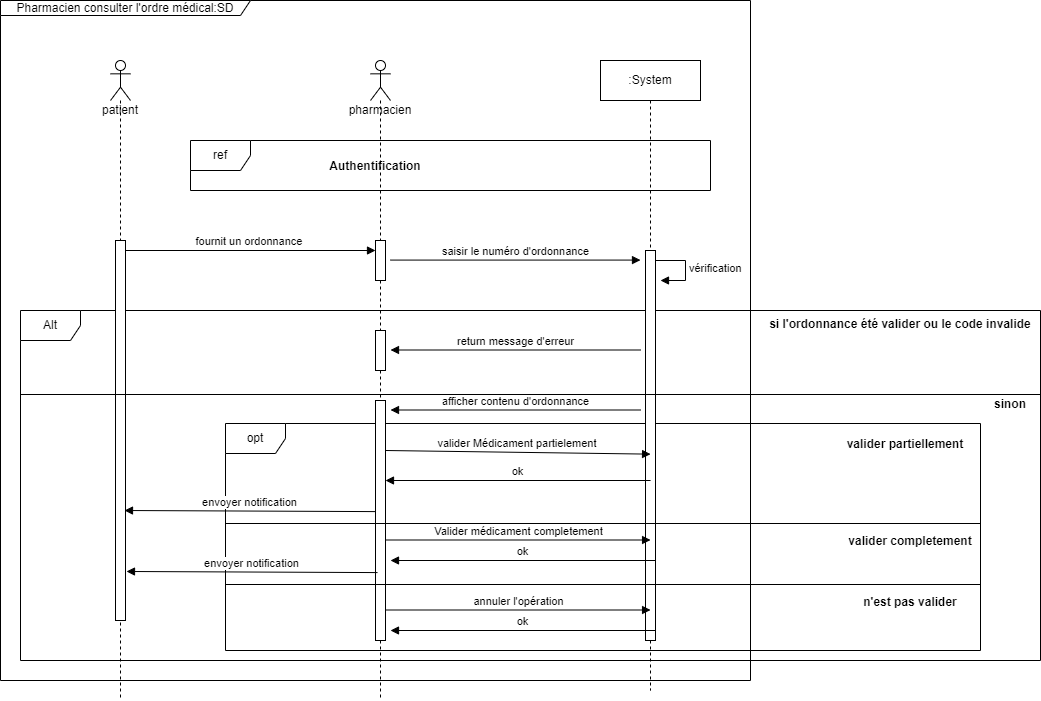
****

Figure 10: Diagramme de sequence consultation ordre medicale par le pharmacien

#### Diagramme de séquence ajouté médecin par Admin cabinet médical:

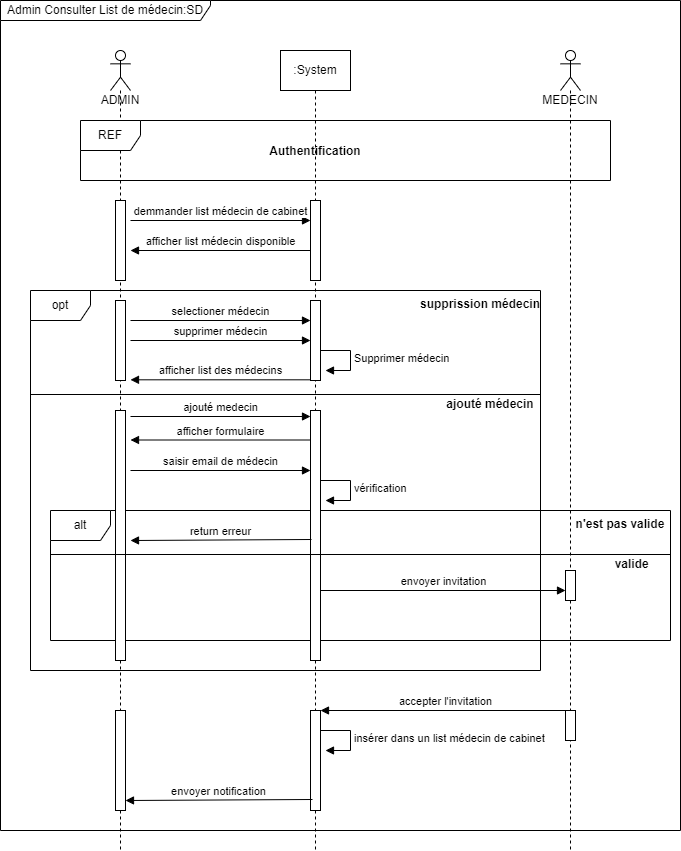
****

Figure 11: Diagramme de sequence admin ajouté medecin

### Le Diagramme de classe :

Le diagramme de classes est l'un des types les plus populaires en langage UML. Très utilisé par les ingénieurs logiciel pour documenter l'architecture des logiciels, les diagrammes de classes sont un type de diagramme de structure, car ils décrivent ce qui doit être présent dans le système modélisé. Quel que soit votre degré de familiarité avec les diagrammes UML ou les diagrammes de classes, notre logiciel UML est conçu pour être simple à utiliser.(7)

#### Composants de base d’un diagramme de classes :

Le diagramme de classes standard est composé de trois sections :

* **Section supérieure :** contient le nom de la classe. Cette section est toujours nécessaire, que vous parliez du classifieur ou d'un objet.(7)
* **Section intermédiaire :** contient les attributs de la classe. Utilisez-la pour décrire les qualités de la classe. Elle n'est nécessaire que lors de la description d'une instance spécifique d'une classe.(7)
* **Section inférieure :** contient les opérations de la classe (méthodes), affichées sous forme de liste. Chaque opération occupe sa propre ligne. Les opérations décrivent la manière dont une classe interagit avec les données.(7)

#### Autres composants d'un diagramme de classes :

* **Classes :** modèle pour créer des objets et mettre en œuvre un comportement dans un système. En langage UML, une classe représente un objet ou un ensemble d'objets possédant une structure et un comportement communs. On les représente par un rectangle comprenant des lignes pour le nom de la classe, ses attributs et ses opérations. Lorsque vous dessinez une classe dans un diagramme de classes, seule la ligne supérieure est obligatoire, les autres sont facultatives et ne servent qu'à fournir des détails supplémentaires.
* **Nom :** première ligne d'une forme de classe.
* **Attributs :** deuxième ligne d'une forme de classe. Chaque attribut de la classe apparaît sur une ligne distincte.
* **Méthodes :** troisième ligne d'une forme de classe. On les appelle aussi opérations ; elles apparaissent sous forme de liste, chaque opération occupant une ligne différente.
* **Signaux :** symboles qui représentent les communications à sens unique et asynchrones entre des objets actifs.
* **Types de données :** classifieurs qui définissent des valeurs de données. Les types de données peuvent modéliser les types primitifs et les énumérations.
* **Paquetages :** formes conçues pour organiser les classifieurs connexes d'un diagramme. On les symbolise par une grande forme rectangulaire à onglets.
* **Interfaces :** groupe de signatures d'opération et/ou de définitions d'attributs définissant un ensemble cohérent de comportements. Les interfaces sont semblables à des classes, sauf qu'une classe peut avoir une instance de son type et qu'une interface doit compter au moins une classe pour la mettre en œuvre.
* **Énumérations :** représentations de types de données définis par l'utilisateur. Une énumération comprend des groupes d'identificateurs qui représentent des valeurs de l'énumération.
* **Objets :** instances d'une ou plusieurs classes. On peut ajouter des objets à un diagramme de classes pour représenter des instances concrètes ou prototypiques.
* **Artefacts :** éléments du modèle qui représentent les entités concrètes d'un système logiciel, tels que des documents, des bases de données, des fichiers exécutables, des composants logiciels, etc.

#### Les interactions dans un diagramme de classe :

Les interactions dans un diagramme de classe représentent la manière dont les classes interagissent entre elles dans un système. Il existe plusieurs types d'interactions que l'on peut représenter dans un diagramme de classe : (7)

* **Association :** une association représente une relation entre deux classes qui peut être unidirectionnelle ou bidirectionnelle. (7)
* **Agrégation et composition :** ces deux types d'associations représentent une relation "partie-tout". Dans une agrégation, les parties peuvent exister indépendamment du tout, tandis que dans une composition, les parties sont des composants du tout et ne peuvent pas exister indépendamment. (7)
* **Héritage :** l'héritage représente une relation "est-un" entre deux classes.(7)
* **Dépendance :** une dépendance représente une relation entre deux classes où un changement dans l'une peut affecter l'autre.(7)

#### diagramme de class pour l’application web gestion réservation médical :

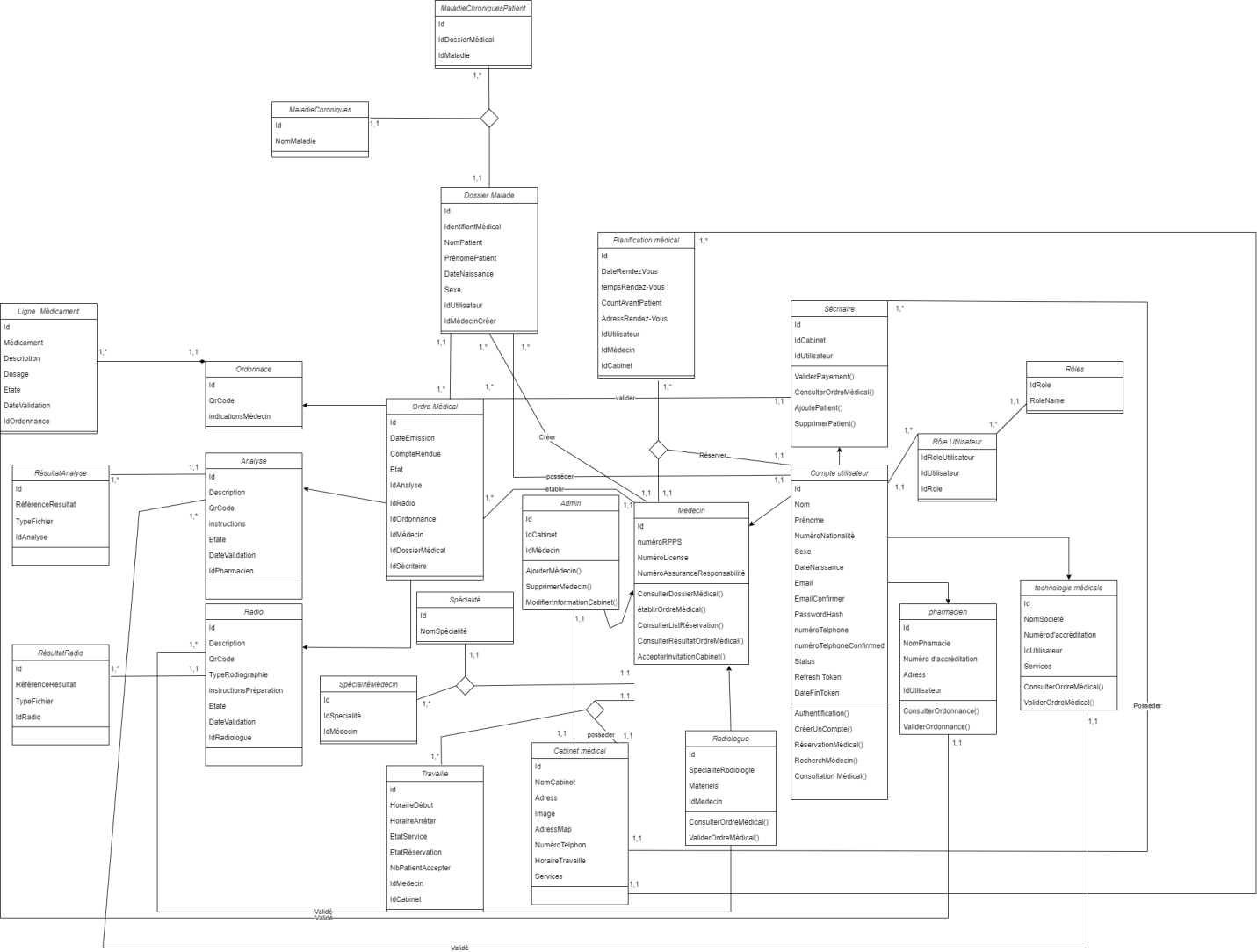


Figure 12:Dagramme de class de system géstion reservation médicale

#### Dictionnaire de class et des attributs :

|  |  |  |  |
| --- | --- | --- | --- |
| Class | Attribut | | Méthodes |
| champ | type |  |
| Utilisateur | * Id * Nom * Prénom * NuméroNationalité * DateNaissance * Sexe * Email * ConfirmationEmail * passwordHash * numéroTelephone * ConfirmationTelephone * EtatCompte * Token * RefreshToken * DateFinToken | Char  Char  Char  Num  Char  Char  Enum  Char  Num  Enum  Enum  Char  Char  Date | Authentification()  CréeCompte()  RéservationMédical()  RechercherMédecin()  ConsultationRésultatMédical() |
| Médecin | * Id * NuméroRpps * NuméroLicense * NuméroAssurence * Etat | char  Num  Num  Num  Enum | ConsulterDossierMalade()  EtablirOrdreMédical()  ConsulterListRéservation()  ConsulterRésultatOrdreMédical()  AccepterInvitationCabinet()  SupprimerListeReservation() |
| SpécialitéMédecin | * Id * IdMédecin * IdSpécialites | Num  Char  Char |  |
| Pharmacien | * Id * NuméroAccréditation * Adress * IdUtilisateur * NomPharmacie * Etat | char  Num  Char  Char  Char  Enum | ConsulterOrdonnance()  ValiderOrdonnance() |
| Technologie Médical | * Id * NuméroAccréditation * Services * IdUtilisateur * NomSociété * Etat | char  Num  Char  Char  Char  Enum | ConsulterOrdreMédical()  ValiderOrdreMédical () |
| Radiologue | * Id * SpecialiteRadiologie * IdMedecin * Materiels | char  Enum  Char  Char | ConsulterOrdreMédical()  ValiderOrdreMédical () |
| Sécritaire | * Id * IdCabinet * IdUtilisateur | char  Num  Char | ValiderPayement()  consulterOrdreMédical() |
| Roles | * Id * NomRole | Num  Char |  |
| RolesUtilisateur | * Id * IdRole * IdUtilisateur | Num  Num  char |  |
| CabinetMédical | * Id * NomCabinet * Adress * AdressMap * Image * NuméroTelphone * HoraireOuverture * Services | Num  Char  Char  Char  Char  Num  Time  Char |  |
| TravailleHoraire | * Id * HoraireDébut * HoraireFin * IdMedecin * IdCabinet * EtatService * NbPatienRéserver * EtatRéservation | Num  Time  Time  char  Num  Enum  Num  Enum |  |
| Spécialites | * NomSpécialite * NomSpécialite | Char  Char |  |
| PlanificationMédical | * Id * DateRendez-vous * TempsRendez-Vous * NombrePatienAvant * IdUtilisateur * IdMédecin * AdressRendez-Vous * IdCabinet | Num  Date  Time  Num  Char  Char  Char  Num |  |
| Ordonnance | * Id * QrCode * IndicationsMédecin * IdOrdreMédical | Char  Char  Char  Char |  |
| Ligne Médicament | * Id * Médicament * Dosage * Etat * IdOrdonnance * Description | Char  Char  Num  Enum  Char  Char |  |
| Analyses | * Id * Description * QrCode * Instructions * Etat * DateValidation * IdOrdreMédical | Num  Char  Char  Char  Enum  Date  Char |  |
| Radio | * Id * QrCode * TypeRadiographie * InstructionPréparation * Etat * DateValidation * Description * IdOrdreMédical | Char  Char  Enum  Char  Enum  Date  Char  Char |  |
| DossierMédical | * Id * NomPtient * PrénomPatient * DateNaissance * Sexe * IdentifientMédical * IdPatient * IdMédecinCrée * DateCréation | Char  Char  Char  Date  Enum  Char  Char  Char  Date |  |
| MaladieChronique | * Id * NomMaladie | Num  Char |  |
| MaladieChroniquePatient | * Id * IdDossierMalade * IdMaladie | Num  Char  num |  |
| OrdreMédical | * Id * DateEmmission * IdMédecin * IdDossiérMédical * CompteRendu * Etat | Char  Date  Char  Char  Char  Char  Char  Char  Char  Enum |  |
| RésultatRadio | * Id * RéférenceFichier * TypeFichier * IdRadio | Char  Char  Enum  char |  |
| RésultatAnalyse | * Id * RéférenceFichier * TypeFichier * IdAnalyse | Char  Char  Enum  char |  |

### Le modèle de données relationnel :

Le terme de modèle de données relationnel désigne une manière de structurer les informations sous la forme de matrices que l’on appelle tables ou relations. Ce modèle, très simple, est de loin le plus répandu dans les Systèmes de Gestion de Bases de Données (SGBD), qui sont ainsi dénommés SGBD relationnels. Une base de données relationnelle est donc constituée d’un ensemble de données structurées sous forme de relations. (8)

#### Règles de passage au modèle relationnel:

Le passage d'un diagramme de classes au modèle relationnel implique plusieurs étapes. Voici les règles de base à suivre :

* **Identifier les classes et les relations :** Tout d'abord, identifiez toutes les classes et les relations dans le diagramme de classes.
* **Transformer les classes en tables :** Pour chaque classe, créez une table correspondante dans le modèle relationnel. Chaque attribut de la classe devient une colonne de la table et chaque instance de la classe devient une ligne de la table.
* **Identifier les clés primaires :** Pour chaque table, identifiez une colonne qui représente la clé primaire. La clé primaire doit être unique pour chaque ligne de la table et ne doit pas contenir de valeurs nulles.
* **Transformer les associations en clés étrangères** : Pour chaque association entre deux classes, ajoutez une colonne dans l'une des tables pour représenter la clé étrangère de l'autre table. La colonne de clé étrangère doit contenir la valeur de la clé primaire de l'autre table.
* **Gérer les associations de plusieurs à plusieurs : Si** une association est de type "plusieurs à plusieurs", créez une troisième table pour représenter l'association. Cette table doit contenir les clés primaires des deux tables associées.
* **Gérer les attributs multivalués :** Si un attribut d'une classe peut avoir plusieurs valeurs, créez une table distincte pour cet attribut et utilisez une clé étrangère pour la relier à la table principale.
* **Gérer les héritages :** Si le diagramme de classes contient des hiérarchies d'héritage, créez une table pour chaque classe et utilisez une clé étrangère pour représenter la relation d'héritage entre les tables.

#### Model Relationnel :

Après avoir appliqué les règles de passage cités précédemment, nous avons établi le schéma relationnel suivant :

Utilisateur(Id,Nom,Prénom,NuméroNationalité,Sexe,DateNaissance,Email,EmailConfirmer,PasswordHash,NumTelphon,NumTelphConfirmer,Status,RefreshToken,DateFinRefreshToken)

Roles(Id,NomRole)

RoleUtilisateur(Id,#IdUtilisateur,#IdRole)

Médecin(Id, NuméroRPPS,NuméroLiscense,NuméroAssurenceMédical)

Pharmacien(Id,#IdUtilisateur,NomPharmacien,IdentifientPharmacien,AdressPharmacie)

technologieMédical(Id,#IdUtilisateur,NomSociété,IdentifienTechnologie)

Radiologue(Id,#IdMedecin,Specialite,Matériels)

AdminCabinet(Id,#IdCabinet,#IdMédecin)

Sécritaire(Id,#IdCabinet,#IdUtilisateur)

CabinetMédical(Id,NomCabinet,Adress,Image,AdressMap,NuméroTelphCabinet,HoraireOuvert,Services)

Spécialites(Id,NomSpecialite)

SpécialitéMédecin(Id,#IdMédecin,#IdSpecialité)

PlanificationMédical(Id,#IdUtilisateur,#IdMedecin,#IdCabinet,DateRendezVous,TempRendezVous,CountAvantPatient,AdressRendezVous)

TravailleHoraire(Id,#IdCabinet,#IdMédecin,HoraireDébut,HoraireFin,EtatService,EtatRéservation)

DossiérMalade(Id,#IdUtilisateur,#IdMedecin,IdentifientMédical,NomPatient,PrénomPatient,DateNaissance,Sexe)

MaladieChronique(Id,NomMaladié)

MaladieChroniquePatient(Id,#IdDossiérMalade,IdMaladie)

OrdreMédical(Id,#IdMedecin,#IdDossierMalade,#IdSécritaire,DateEmission,CompteRendu,Etat)

Analyse(Id,#IdOrdreMédical,Description,QrCode,Instruction,RésulatAnalyse,Etat,DateValidation,#IdPharmacien)

Radio(Id,#IdOrdreMédical,Description,QrCode,TypeRadioGraphie,InstructionPréparation,Etate,DateValidation,IdRadiologue)

Ordonnance(Id,#IdOrdreMédical,QrCode,IndicationsMédecin)

LigneMédicament(Id,#IdOrdonnance,#IdPharmacien,Médicament,Description,Dosage,

Etat,DateValidation)

### Conclusion :

Ce chapitre a porté principalement sur les aspects analytiques et conceptuels de notre système. La phase de conception est cruciale dans tout projet, et nous l'avons abordée en définissant le cas d'utilisation, puis en le traduisant en diagramme de séquence. Enfin, le diagramme de classes a été développé pour construire les tables de la base de données. Dans le prochain chapitre, nous présenterons la mise en œuvre de notre système.

## :Implémentation System Gestion Réservation Médical

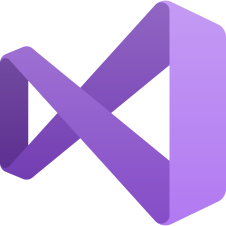
### Introduction :

L'implémentation d'une application web est une étape clé dans le développement d'un projet informatique. Cette partie du mémoire se concentre sur la mise en place concrète de l'application en question, en décrivant les différentes étapes du processus de développement, les choix technologiques qui ont été faits. Dans Cette partie du mémoire peut inclure des sections telles que la présentation des langages de programmation et des Framework utilisés, les choix d'architecture technique et de base de données, les tests réalisés pour valider le bon fonctionnement de l'application, les fonctionnalités mises en place, ainsi que des exemples de code pour illustrer le développement.

### Les outils technique utiliser dans le Projet :

#### IDE Visuel Studio :

est une suite de logiciels de développement pour Windows conçu par Microsoft. La dernière version s'appelle Visual Studio 2022.

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications Web ASP.NET, des Services Web XML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C++, Visual C# et Visual J# utilisent tous le même environnement de développement intégré (IDE, Integrated Development Environment), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du Framework .NET, qui fournit un accès à des technologies clés simplifiant le développement d'applications Web ASP et de Services Web XML grâce à Visual Web Développer.(9)

#### SQLServer 18 :

SQL Server est un système de gestion de base de données relationnelle développé par Microsoft. Il est utilisé pour stocker, organiser et gérer des données structurées. SQL Server permet aux utilisateurs de stocker et de récupérer des données en utilisant le langage de requête SQL (Structured Query Language).

Il est couramment utilisé dans les entreprises pour gérer les données de leurs applications, notamment les données financières, les informations sur les clients et les inventaires. SQL Server offre une grande fiabilité, une sécurité avancée, des fonctionnalités de sauvegarde et de restauration des données, ainsi que des options d'évolutivité pour prendre en charge les charges de travail à grande échelle.(3)

#### PostMan :

Postman est un outil de développement d'API qui permet aux développeurs de tester, de documenter et de partager des API plus facilement. Il offre une interface utilisateur conviviale pour envoyer des requêtes HTTP, des tests automatisés, des scripts pour gérer les environnements d'API, ainsi que la collaboration entre les membres de l'équipe. En somme, Postman simplifie le processus de développement d'API en permettant aux développeurs de travailler plus efficacement et de manière collaborative.

### Les Outille de Développement:

#### .NET :

.NET est un cadre de développement logiciel développé par Microsoft qui fournit une plateforme pour la création de différents types d'applications, notamment des applications web, des applications de bureau, des applications mobiles et des jeux. Il se compose d'une grande bibliothèque de classes et d'un runtime de langage commun (CLR) qui permet l'interopérabilité entre les différentes langues de programmation prises en charge par la plateforme.(3)

#### Bootstrap 5 :

Bootstrap est un framework développé par l'équipe du réseau social Twitter. Proposé en open source (sous licence MIT), ce framework utilisant les langages HTML, CSS et JavaScript fournit aux développeurs des outils pour créer un site facilement. Ce framework est pensé pour développer des sites avec un design responsive, qui s'adapte à tout type d'écran, et en priorité pour les smartphones. Il fournit des outils avec des styles déjà en place pour des typographies, des boutons, des interfaces de navigation et bien d'autres encore. On appelle ce type de framework un "Front-End Framework".(13)

#### BlazorWebAssembley :

Blazor WebAssembly est une nouvelle technologie d'interface utilisateur de Microsoft, officiellement publiée avec .NET Core 3.1 et recevant des mises à jour dans .NET 5. Blazor permet aux développeurs de créer des applications à page unique (SPAs) en utilisant C# et .NET, en utilisant une architecture basée sur des composants. Blazor WebAssembly est une implémentation côté client dans le navigateur de Blazor qui inclut un runtime .NET implémenté en WebAssembly.(10)

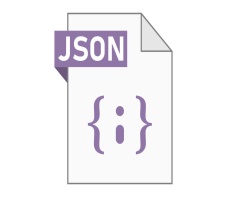
#### Asp.net Web Api Core :

ASP.NET Web API Core est un framework pour la création de services basés sur le protocole HTTP sur la plateforme .NET Core. Il permet aux développeurs de créer facilement des APIs RESTful qui peuvent être consommées par une grande variété de clients, y compris les navigateurs, les appareils mobiles et les applications de bureau.

#### C#:

C# (prononcé "C Sharp") est un langage de programmation orienté objet, développé par Microsoft, qui combine les fonctionnalités de C++ et de Java. Il est principalement utilisé pour développer des applications Windows, des applications Web, des applications de bureau et des jeux vidéo.(3)

#### Json:

JSON (JavaScript Objet Notation) est un langage léger d’échange de données textuelles. Pour les ordinateurs, ce format se génère et s’analyse facilement. Pour les humains, il est pratique à écrire et à lire grâce à une syntaxe simple et à une structure en arborescence. JSON permet de représenter des données structurées

### Architecture d’Application web :

#### L'architecture SPA avec API:

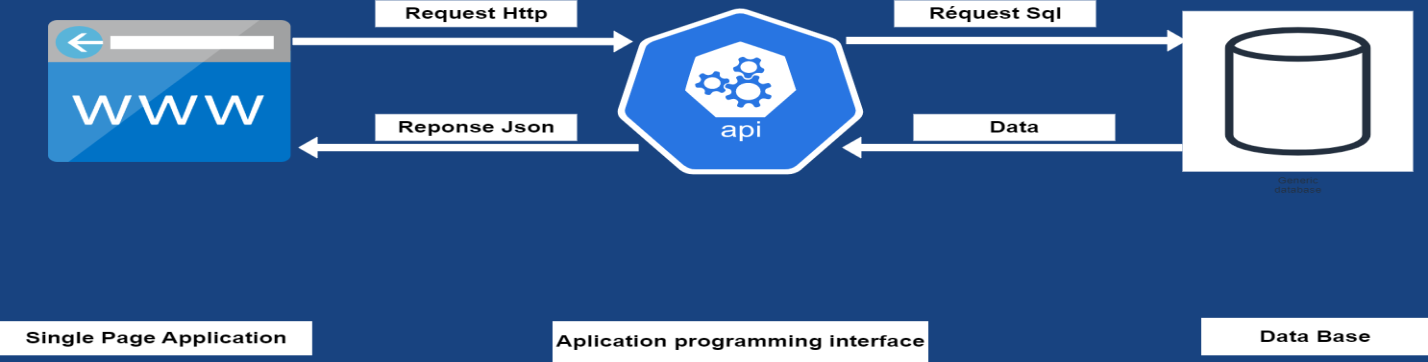
L'architecture SPA avec API est une architecture de développement web moderne qui permet de créer des applications web réactives et rapides. Elle se compose de deux parties principales : une application côté client et une API côté serveur.

L'application côté client est développée en utilisant des frameworks BlazorWebAssembley qui permettent de créer une interface utilisateur dynamique et réactive qui interagit avec l'API pour récupérer et afficher les données.

L'API côté serveur fournit les données nécessaires à l'application côté client. Elle est développée en utilisant ASP.NET wab Api Core. L'API est responsable de récupérer les données à partir d'une source de données, telles qu'une base de données ou un service tiers, et de les renvoyer sous forme de données JSON pour que l'application côté client puisse les utiliser.

L'architecture SPA avec API présente de nombreux avantages. Elle permet de créer des applications web réactives et rapides qui fournissent une expérience utilisateur plus fluide et plus rapide. Elle permet également de séparer la logique de l'interface utilisateur de la logique de l'application, ce qui facilite la maintenance et la mise à l'échelle de l'application.

En utilisant cette architecture, il est également possible de développer des applications web qui peuvent être facilement déployées dans le cloud, ce qui permet de les mettre à disposition de manière rapide et efficace pour les utilisateurs.



#### Architecture SPA Web Assembley :

Dans cette partie, nous nous appuyons sur la division en deux parties, chaque partie ayant une fonction spécifique. Voici les partie principal dans le SPA :

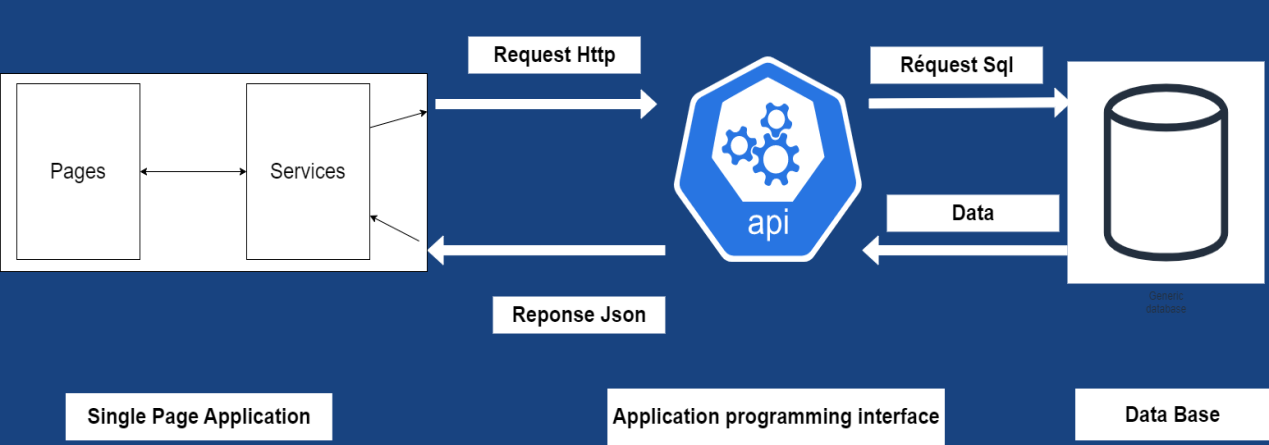
* **Pages :**

pages sont représentés sous forme de composants de l'interface utilisateur qui sont définis dans des fichiers de code source .cshtml ou .razor et organisés dans une hiérarchie de composants pour définir la structure de l'application.

* **Services :**

Dans une application web SPA (Single Page Application) basée sur .NET WebAssembly, les fichiers de services peuvent être utilisés pour fournir des fonctionnalités communes à différents composants de l'interface utilisateur.

Les services sont des classes qui encapsulent des fonctionnalités spécifiques de l'application, telles que la récupération des données, la gestion de l'état, etc. Les services peuvent être injectés dans les composants de l'interface utilisateur pour fournir des fonctionnalités spécifiques à ces composants.



#### Architecture Web Api (Controllers,Services,Manager) :

L'architecture Controller Services Manager (CSM) est une approche pour la conception d'API Web dans le cadre du framework ASP.NET Core. Cette architecture vise à séparer la logique métier (Services) de la logique de présentation (Controllers) en utilisant un objet intermédiaire appelé Manager.

Voici une présentation des composants clés de l'architecture CSM dans un Web API Core :

* **Les Controllers :**

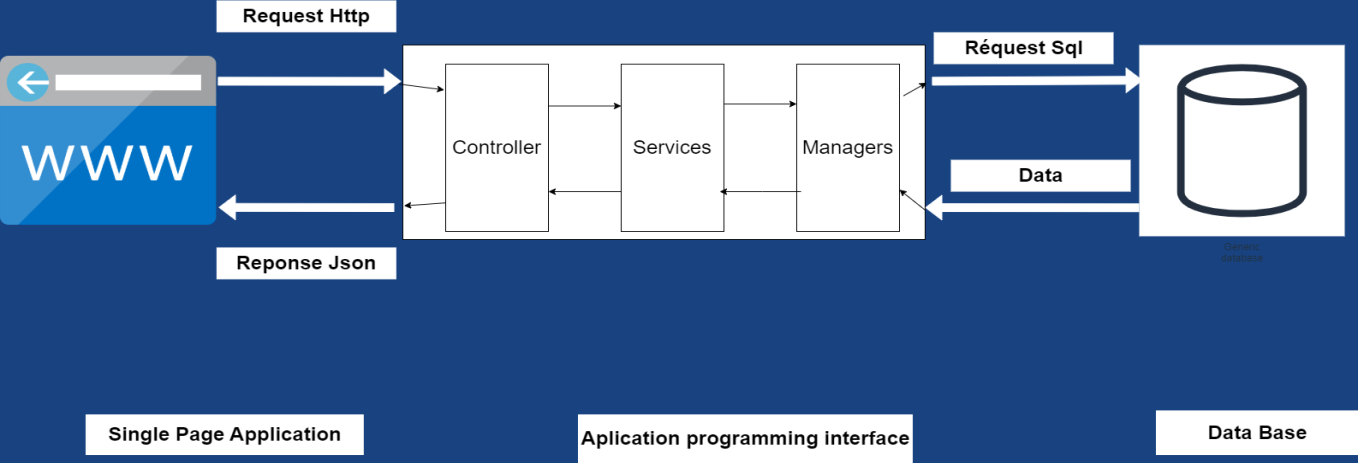
Les contrôleurs sont des classes qui acceptent les requêtes HTTP et retournent des réponses HTTP. Dans l'architecture CSM, les contrôleurs sont responsables de la gestion de la présentation et de l'interaction avec l'utilisateur. Ils contiennent les méthodes d'action (Action Methods) qui sont appelées en fonction de la route demandée.

* **Les Services :**

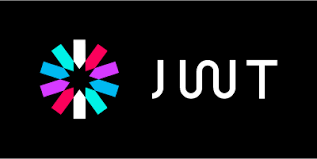
Les Services contiennent la logique métier qui est appelée par les contrôleurs. Les services sont responsables de la gestion des données, du traitement de la logique métier et de la coordination avec d'autres services ou systèmes tiers. Ils sont généralement implémentés sous forme de classes.

* **Les Managers :**

Managers est une couche qui se situe entre les Services et la base de données. Son rôle est de fournir une interface pour interagir avec la base de données tout en cachant les détails de mise en œuvre. Cette couche est souvent implémentée à l'aide d'un ORM (Object-Relational Mapping) tel que Entity Framework Core.



### Méthode JWT pour Authentification :

L'authentification JWT (JSON Web Token) est une méthode d'authentification qui permet de sécuriser les communications entre différentes parties d'une application web ou d'un système distribué. Cette méthode d'authentification utilise un jeton qui est généré par le serveur d'authentification et qui est ensuite utilisé pour valider l'identité de l'utilisateur.(15)

Voici comment fonctionne l'authentification JWT :

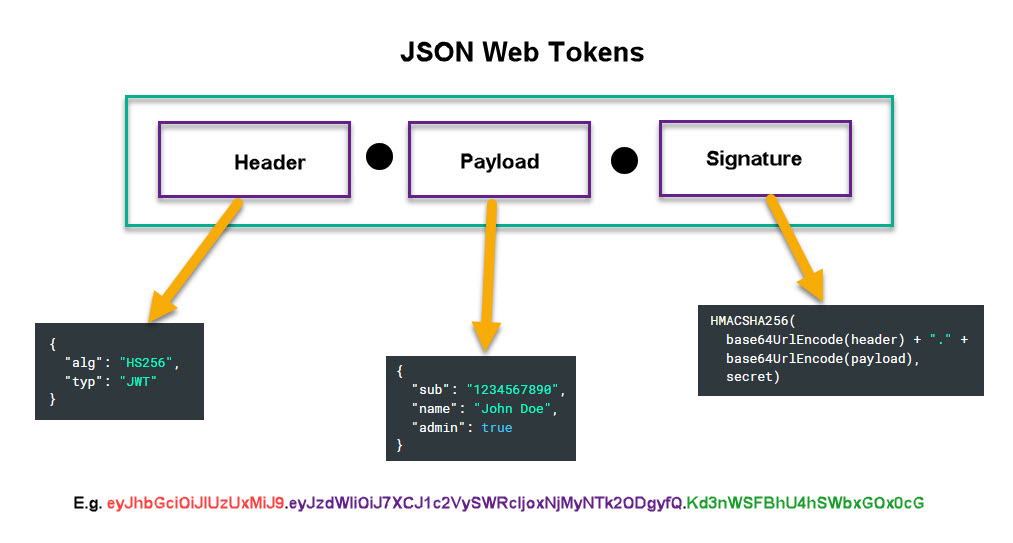
* L'utilisateur envoie ses informations d'identification (nom d'utilisateur et mot de passe) au serveur d'authentification.
* Le serveur d'authentification vérifie les informations d'identification de l'utilisateur. Si les informations sont correctes, le serveur d'authentification génère un jeton JWT.
* Le serveur d'authentification envoie le jeton JWT à l'utilisateur.
* L'utilisateur envoie le jeton JWT à chaque demande qu'il fait au serveur de l'application.
* Le serveur de l'application vérifie le jeton JWT à chaque demande et autorise ou refuse l'accès à la ressource demandée en fonction de l'authenticité du jeton.(15)

Le jeton JWT est composé de trois parties :

L'en-tête : il contient les informations sur le type de jeton et l'algorithme utilisé pour le cryptage.

Le corps : il contient les informations sur l'utilisateur qui a été authentifié, telles que le nom d'utilisateur et les autorisations d'accès.

La signature : elle est utilisée pour valider l'authenticité du jeton.(15)



### Les Principes de Programmation Solid :

SOLID est l'acronyme de cinq principes de base (Single Responsibility Principle, Open/Closed Principle, Liskov Substitution Principle, Interface Segregation Principle et Dependency Inversion Principle) que l'on peut appliquer au développement objet.(14)

Voici une brève description de chacun des cinq principes SOLID :

* **Principe de responsabilité unique (SRP - Single Responsibility Principle) :** chaque classe doit avoir une seule responsabilité. Cela signifie qu'une classe doit être conçue pour ne faire qu'une seule chose et la faire bien.(14)
* **Principe ouvert/fermé (OCP - Open/Closed Principle) :** les entités logicielles (classes, modules, fonctions, etc.) doivent être ouvertes à l'extension mais fermées à la modification. Cela signifie que le code doit être facile à étendre sans avoir à le modifier directement.(14)
* **Principe de substitution de Liskov (LSP - Liskov Substitution Principle) :** les sous-classes doivent être substituables à leurs classes de base sans que cela n'affecte le fonctionnement du programme. Cela signifie qu'une classe dérivée doit être capable de remplacer sa classe de base sans que cela n'affecte le comportement global du système.(14)
* **Principe de ségrégation d'interface (ISP - Interface Segregation Principle) :** les interfaces doivent être conçues de manière à ne contenir que les méthodes nécessaires pour leur utilisation. Cela signifie que les interfaces doivent être spécifiques à chaque classe et ne doivent contenir que les méthodes nécessaires pour cette classe.(14)
* **Principe de dépendance inversée (DIP - Dependency Inversion Principle) :** les dépendances doivent être orientées vers les abstractions plutôt que vers les détails concrets. Cela signifie que les classes doivent dépendre d'abstractions plutôt que de détails concrets.(14)

### Implémentation System :

1. <https://fr.yeeply.com/blog/5-types-developpement-d-applications-web/>
2. <https://lesconnectes.net/comment-fonctionne-une-application-web/>
3. [https://www.redhat.com/fr/topics/cloud-native-apps/what-is-an-application-architecture /](https://www.redhat.com/fr/topics/cloud-native-apps/what-is-an-application-architecture%20/)
4. <https://www.lucidchart.com/pages/fr/langage-uml>
5. <http://www-igm.univ-mlv.fr/~dr/UML/ROSE/Acteurs.html>
6. <https://www.lucidchart.com/pages/fr/diagramme-de-cas-dutilisation-uml>
7. <https://www.lucidchart.com/pages/fr/diagramme-de-classes-uml>
8. <https://formations.imtatlantique.fr/bd_ihm/fr/intro_db/relational_model/>
9. <https://www.techno-science.net/glossaire-definition/Visual-Studio.html>
10. <https://www.rocksolidknowledge.com/articles/an-introduction-to-blazor-webassembly>
11. <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445308-json-definition-et-presentation-de-ce-format-de-donnees/>
12. <https://learn.microsoft.com/fr-fr/aspnet/core/tutorials/min-web-api?view=aspnetcore-7.0&tabs=visual-studio>
13. <https://www.journaldunet.com/web-tech/developpeur/1159810-bootstrap-definition-tutoriels-astuces-pratiques/>
14. <https://philippe.developpez.com/articles/SOLIDdotNet/#:~:text=SOLID%20est%20l'acronyme%20de,peut%20appliquer%20au%20d%C3%A9veloppement%20objet>.
15. <https://www.vaadata.com/blog/fr/jetons-jwt-et-securite-principes-et-cas-dutilisation/>