****

**Promotion 2016-2019**

**Septembre 2019**

Présenté et soutenu par

**BA Ndeye Penda**

ET

**Haroun Mahamat Haoua**

Sous la direction de

**M. Jean Marie Preira,**

**Enseignant-chercheur à l’ESMT**

Plateforme d’Optimisation de prise de rendez-vous médicaux

**DEVELOPPEMENT D'APPLICATIONS REPARTIES**

**TELECOMMUNICATIONS ET INFORMATIQUE**

DEDICACE

Nous dédions tout ce travail fourni à :

* Nos parents pour leurs sacrifices dans le but de nous mettre dans les meilleures conditions de réussite possibles, leurs prières et leurs précieux conseils.
* Nos frères et sœurs pour leur amour et encouragement.
* A nos responsables de classe, Cheikh Sadbouh Sow et Adjaratou Khady N’diaye.
* L’ensemble des étudiants de DAR, nos compagnons de tous les jours, nous avons compris que l’union fait la force.

REMERCIEMENTS

* Notre professeur titulaire Mr Preira pour sa disponibilité, sa gentillesse, sa tolérance et surtout le suivi tout au long de notre cursus,
* Mr Nantoume pour sa grande disponibilité, les conseils et tout le soutien apporté depuis notre arrivée à l’ESMT,
* L’ensemble du corps professoral de l’ESMT, pour leur disponibilité, leur patience, leurs conseils, leur intérêt incontestable et leur soutien tout au long de nos 3 années de formation. La deuxième promotion de LPTI se souviendra toujours de vous.

TABLEAU

Tableau 1 : Tableau comparatif de quelques plateformes de prise de rendez-vous médicaux en Europe

Tableau 2 : Tableau comparatif de quelques plateformes de prise de rendez-vous médicaux en Afrique

Tableau 3 : fonctionnalités coté FrontEnd

Tableau 4 : fonctionnalités coté BackEnd

Tableau 5 : Spécifications fonctionnelles détaillées chez le patient

Tableau 6 : Spécifications fonctionnelles détaillées chez le médecin

Table des schémas

Schéma 1 : périmètre fonctionnel du logiciel

Schéma 2 : Arborescence du logiciel

Schéma 3 : Organigramme de processus UP

Schéma 4 : Diagramme de cas d'utilisation de l’administrateur

Schéma 5 : Diagramme de cas d'utilisation du Patient

Schéma 6 : Diagramme de cas d'utilisation de Médecin

Schéma 7 : Diagramme de Séquence l’authentification

Schéma 8 : Diagramme de Séquence de la prise de rendez-vous

Schéma 9 : Diagramme de Séquence du patient

Schéma 10 : Diagramme de classe du système

Table des Figures

Figure 1 : logo du logiciel Astah Professionnel

Figure 2 : logo du Framework Ionic

Figure 3 : logo du Framework Angular

Figure 4 : logo du Framework Spring

Figure 5 : logo de l’environnement de développement WebStorm

Figure 6 : logo de l’environnement de développement STS

Figure 7 : logo du MongoDB et de son outil d’interface graphique Mongo 3T

Table des captures

Capture 1 : Interface de connexion et d’accueil de l’application mobile

Capture 2 : Interface de recherche de médecin de l’application mobile

Capture 3 : interface d’accueil de l’application web

Capture 4 : interface a propos de l’application web

Capture 5 : interface de contact de l’application web

Capture 6 : interface prise de rendez-vous

Capture 7 : interface prise services de l’application web de l’application web

SOMMAIRE

INTRODUCTION

1. Architecture du système et environnement de test
2. Environnement de développement
3. Description des interfaces réalisées

CONCLUSION

CONCLUSION GENERALE

WEBOGRAPHIE

AVANT-PROPOS

L'Ecole Supérieure Multinationale des Télécommunication (ESMT) est née, en 1981, de l'initiative de sept pays d'Afrique de l'Ouest (Bénin, Burkina Faso, Mali, Mauritanie, Niger, Sénégal, Togo). Elle est une institution multinationale qui a pour vocation de former des diplômés (Techniciens supérieurs, Licences Professionnelles, Ingénieurs, Masters, Mastères spécialisés) dans les domaines techniques et managériaux des télécommunications/TIC. Elle accueille en formation initiale ou continue des stagiaires qui proviennent de l’ensemble des pays francophones d’Afrique, recrutés au niveau des écoles, des universités ou directement chez les opérations de télécommunications.

Pour terminer et valider un cycle l’étudiant arrivé à la fin de celui-ci doit faire un mémoire d'étude.

C’est donc dans cette option que nous avons pensé à mettre en place une application de prise de rendez-vous médicaux en ligne.

# INTRODUCTION GENERALE

De nos jours, le monde connaît une avancée technologique considérable dans tous les secteurs et cela en grande partie grâce à l'informatique qui joue un rôle très important dans le développement des entreprises et d'autres établissements et permet aussi à l'Homme de gagner du temps tout en améliorant sa condition de vie.

Les hôpitaux et cliniques font partie des institutions que l'informatique pourra beaucoup aidés. En effet, la croissance de la population hospitalière et la nécessité de gagner du temps nécessite la mise en place d'un système rationnel de gestion de rendez-vous médicaux. Au Sénégal, les rendez-vous chez le médecin sont pris soit par téléphone ou bien en visitant l’hôpital en personne. Ce processus manuel nécessite la disponibilité des planificateurs, de lignes téléphoniques, ou une présence physique.

Cependant, le besoin de soins intemporels et efficaces devient de plus en plus une exigence, car les rendez-vous manuels (nécessitant la présence des deux personnes) et les longues files d’attentes ont créé une situation irritant pour les établissements de santé. La mise en place d’un système intégré permettant aux patients de pouvoir gérer leurs rendez-vous (par exemple prendre un rendez-vous ou annulé son rendez-vous sans se déplacer) reste ainsi incontournable. L’émergence d’un système de gestion de rendez-vous en ligne offre un accès intemporel et efficace aux services de soins.

L'objectif principal de la mémoire est de fournir des soins médicaux de qualité aux patients en réunissant tous les médecins sur une plate-forme unique afin que tout le monde puisse facilement y accéder et prendre des rendez-vous. L’idée est de rapprocher le médecin de ses patients.

Ce rapport s’articule autour de cinq chapitres comme suit :

* Une présentation permettant de placer le projet dans son contexte général. Nous présentons la problématique ainsi qu'une description du projet et quelques plateformes dans ce domaine.
* Une étude conceptuelle où nous nous identifierons les acteurs du système et en se basant sur le langage de modélisation UML nous présenterons les diagrammes nécessaires.
* Un dernier chapitre, où nous présenterons les outils matériels et logiciels utilisés pour l’implémentation de notre système, ainsi que les interfaces de certaines fonctionnalités mises au point.

# **Chapitre 1 : PRÉSENTATION DU PROJET**

# **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'élaboration de la problématique ainsi que la solution et la présentation du projet.

# **Problématique**

Selon une étude **IFOP** pour le cabinet de conseil spécialisé en santé **Jalma** publiée en décembre 2012, un Français sur deux reconnaît avoir déjà "*renoncé à des soins chez un médecin généraliste*" à cause de "*la difficulté d'obtenir un rendez-vous dans un délai suffisamment rapide*", et ce chiffre atteint 59% pour les consultations chez les spécialistes.

Pourtant, à en croire la Confédération des Syndicats Médicaux Français, chaque année, 28 millions de rendez-vous médicaux ne sont pas honorés, "sans que les patients ne prennent seulement la peine de prévenir leur médecin pour annuler ou reporter leur rendez-vous".

En Afrique en général et au Sénégal en particulier, un nombre important de décès est lié aux complications des maladies simples, dues au recours tardif à une consultation par un Médecin malgré la présence de structures sanitaires généralement à portée géographique des populations. En effet, l’accès à un Médecin est très difficile pour les familles à cause de : les longues files d’attente, l’absence d’informations sur les plannings de consultation des médecins et les services disponibles dans les hôpitaux (il y arrive des patients meurent en cours de péripétie à la recherche d’un système adéquat).

Partant de ce constat, il est clair que le besoin d’une plateforme réunissant médecins et patients reste indispensable. Chaque parti pourra gérer ses rendez-vous.

# **Solution envisagée**

Compte tenu des problèmes cités ci-dessus, notre objectif est de fournir une plateforme où tous les médecins et les patients peuvent être réunis. Dans cette application, tous les praticiens, qu'ils soient nouveaux ou travaillant depuis longtemps, peuvent s'inscrire eux-mêmes. De cette façon, les gens peuvent choisir de consulter un médecin de leur choix. En utilisant cette application, l'utilisateur pourra connaître et accéder au profil professionnel de chaque médecin inscrit dans toutes les spécialités. Les utilisateurs pourraient prendre rendez-vous chez eux. L'utilisateur peut choisir un médecin proche de lui, ou bien un médecin affilié à un service bien déterminé. Sur cette plateforme, les utilisateurs ont donc accès en temps réel aux agendas des médecins qui se sont abonnés, et peuvent réserver en ligne sur les créneaux disponibles. Ils gèrent leurs rendez-vous directement sur la plateforme et à tout moment peuvent les annuler sans se déplacer. Les nouveaux médecins peuvent facilement rejoindre la plateforme et commencer à prendre des rendez-vous sans avoir besoin d'une publicité coûteuse.

L’utilisation de cette application permet par ailleurs :

* Une meilleure gestion du temps,
* Un taux de remplissage optimisé du carnet de rendez-vous,
* Une meilleure anticipation des ressources (appel à des confrères, un remplaçant, etc.).
* De libérer les secrétaires médicaux déjà occupés à gérer les tâches administratives de la patientèle.

# **Quelques plateformes de prise de rendez-vous**

# En Europe

Les entreprises françaises qui se sont positionnées sur ce marché, avec quelques variantes dans le service offert, sont nombreuses : AlloDocteur, Doctolib, prendre mon rendez-vous, rendezvousfacile, clicrdv, docmii, keldoc, mondocteur... Le modèle qui les inspire, c'est celui des Etats-Unis, où l'entreprise Zocdoc lancée en 2007 a brillamment réussi à implanter son système de réservation médicale en ligne.

|  |  |  |
| --- | --- | --- |
| **Plateforme** | **Type d’entreprise** | **Atout principal** |
| AlloDocteur  <https://allodocteur.fr/> | Cabinet médical | Formation gratuite pour la secrétaire médicale |
| Doctolib  <https://www.doctolib.fr/> | Cabinet médical | Outils de gestion dédiés |
| Prendre mon RDV  <https://prendre-mon-rdv.com/> | Cabinet médical | Des formules adaptées selon la spécialité du praticien |
| Docteur-rendez-vous  <https://www.docteur-rendez-vous.fr/> | Cabinet médical | Possibilité de tchatter avec le secrétariat de votre praticien. |
| RDVmedicaux  <https://www.rdvmedicaux.com/> | Cabinet médical | Une stratégie de développement étroitement liée à celle des permanences téléphoniques. |
| Librerdv  <https://www.librerdv.com/doctor/search> | Cabinet médical | Le patient peut consulter la liste de ces rendez-vous. |
| ClicRDV  <https://www.clicrdv.com/fr/> | TPE et PME | Le bouton incitatif « Prendre RDV » présent sur de nombreux sites. |

Tableau 1 : Tableau comparatif de quelques plateformes de prise de rendez-vous médicaux en Europe

# En Afrique

Les plateformes de prise de rendez-vous médicaux en ligne, en Afrique sont de plus en plus fréquentes sur le marché.

Ci-dessous quelques-unes avec chacune son atout principal :

|  |  |  |
| --- | --- | --- |
| **Plateforme** | **Type d’entreprise** | **Atout principal** |
| Docquita  <http://www.docquita.com/> | TPE et PME | Payer sa consultation en ligne grâce au paiement mobile et au PayPal. |
| PassDocteur  <https://www.passdocteur.com/> | TPE et PME | Une fois la demande de rendez-vous complétée et les honoraires payés en ligne par un proche, le patient reçoit un code par SMS, à faire valoir auprès du praticien pendant la consultation ou bien en pharmacie pour obtenir des médicaments. |
| Remed24  <https://www.remed24services.com/services/prise-de-rendez-vous/> | Cabinet Médical | Remed24 se charge de trouver pour l’appelant un rendez-vous chez un prestataire médical (généraliste, spécialiste), paramédical … |
| Vtipsante  [www.vtipsante](http://www.vtipsante/) |  | L’adresse de 665 praticiens de santé est géoréférencée. Cette application permet aussi aux professionnels de santé de lutter contre la fraude aux assurances. |
| **Sama-docteur**  <http://www.sama-docteur.com/> |  | Il vise à réduire les délais d'intervention médicale (gestion des rendez-vous) surtout chez les enfants et les femmes enceintes |

Tableau 2 : Tableau comparatif de quelques plateformes de prise de rendez-vous médicaux en Afrique

# Points Communs :

Le service de base est identique : les patients utilisent gratuitement le site internet, où ils ont accès aux agendas des praticiens, et peuvent réserver en ligne le créneau qui les arrange. Les médecins, eux, payent un abonnement mensuel pour bénéficier de ce service qui leur permet de remettre sur le marché les consultations qui ont été annulées.

Un système de sms permet non seulement aux patients de se créer des alertes afin d'être tenus au courant quand un créneau se libère chez leur médecin, mais également de ne pas oublier les rendez-vous. Ils arrivent à réduire jusqu'à 80% des annulations de leurs cabinets clients.

**CONCLUSION**

Dans ce chapitre, nous avons présenté les différents problèmes rencontrés en Afrique et en Occident au niveau des hôpitaux et nous avons ensuite exposé quelques plateformes de prise de rendez-vous. Après avoir donné leur point en commun, l’étape suivante sera consacrée à une Analyse des besoins détaillée.

# **Chapitre 2 : ANALYSE DES BESOINS ET SPECIFICATIONS FONCTIONNELLES**

# **Introduction**

Dans ce chapitre nous allons faire une analyse des besoins de notre application à savoir, les besoins techniques, fonctionnels et non fonctionnels ainsi qu’aux besoins opérationnels du système. Ensuite nous détaillons son contexte avec l'identification des acteurs et des messages et enfin réaliser le diagramme de contexte dynamique.

# **Analyse des besoins**

# **Identification des acteurs**

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 à travers son IHM.
* **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 patient.**
* **Le médecin.**
* **Le gestionnaire.**
* **L’administrateur.**
* **API Google Map.**
* **Service SMS**.

**Le patient :** C'est l'acteur essentiel de notre projet, le site web pour lui est l'endroit où il peut prendre des rendez-vous après l'authentification et le choix de son médecin comme il peut aussi gérer et consulter ses rendez-vous.

**Le médecin :** notre site web est consacré aussi aux médecins en effet il fait une bonne partie de la gestion de son cabinet en ligne.

**Le gestionnaire :** c'est la personne qui gère les médecins et leurs spécialités.

**L'administrateur :** responsable de l'ajout et/ou la suppression des gestionnaires.

**API Google Map : C’**est un service gratuit de cartographie en ligne. Le service a été créé par Google.

**Service sms :** Le service de messagerie SMS, permet de transmettre de courts messages textuels

# **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.
* L'architecture de système : Bases de données Client/serveur (MySQL/MongoDB).
* Langage de programmation : JAVA, JAVA SCRIPT, HTML, CSS.
* Framework Spring Boot, Angular JS, Bootstrap etc.
* Plate-forme disponible : PC, Windows, STS.

# **Besoins fonctionnels**

Notre application aura pour but de réaliser les fonctionnalités suivantes :

* Le patient doit établir une connexion pour accéder à notre site web.
* Le patient doit s'authentifier pour accéder à son espace personnel.
* Le patient peut prendre un RDV après la recherche du médecin et s'il n'a pas de compte il doit s'inscrire.
* Le patient peut consulter ses RDV et les modifier ou les annuler.
* Le patient peut envoyer un message aux gestionnaires du site.
* Le médecin doit s'authentifier pour accéder à son espace personnel.
* Le médecin peut s'inscrire et avoir un compte personnel après le dépôt de l'agrément de sa spécialité.
* Le médecin peut ajouter plusieurs spécialités à son compte.
* Le médecin peut demander l'ajout d'une nouvelle spécialité dans le site.
* Le médecin peut modifier les informations concernant son compte, en modifiant le lieu où le temps de son travail.
* Le médecin peut annuler une journée en sélectionnant une date.
* Le médecin peut également consulter la liste de ses RDV et ajouter un RDV au profit d'un patient.
* Le médecin peut gérer les RDV ajouté par lui.
* Le médecin peut envoyer un message aux gestionnaires du site.
* Le gestionnaire doit s'authentifier pour accéder à son espace personnel.
* Le gestionnaire peut gérer les comptes des médecins en validant leurs comptes et leurs spécialités.
* Le gestionnaire peut valider la nouvelle spécialité demandée par un médecin.
* Le gestionnaire peut répondre aux messages des médecins et patients.
* L'administrateur doit se connecter à la base de données pour ajouter ou supprimer des gestionnaires (cela se fait manuellement).

# **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.
* Ergonomie et une interface UIX conviviale
* Le système doit signaler tous les messages d’erreurs.

# **Besoins opérationnels**

Pour assurer la fiabilité de notre application, chaque utilisateur (patient, médecin et gestionnaire) doit avoir un login/E-mail et un mot de passe pour consulter son compte. Pour pouvoir récupérer en cas de panne, un système d'archivage et journalisation doit être mis en place.

# **Spécifications fonctionnelles**

Dans un projet informatique, les spécifications fonctionnelles sont une section du cahier des charges ou un document à part entière qui spécifie, décrit, précise les fonctionnalités du site, de l'application ou du logiciel en question.

Les spécifications fonctionnelles détaillées ont pour but de décrire précisément :

* L’ensemble des fonctionnalités de l’application.
* Les objets manipulés, leurs buts et leurs principes de fonctionnement.
* Les écrans utilisateurs mettant en œuvre les fonctionnalités de l’application.
* Le but, le type et le caractère obligatoire de chacun des champs présents sur les écrans de saisie, ainsi que les actions possibles à partir des écrans.

Toutes les fonctionnalités prévues lors de la phase de conception sont précisées dans ce document en indiquant l’implémentation de ces fonctionnalités dans l’application.

Dans cette section, nous allons délimiter le périmètre fonctionnel de notre application et créer son arborescence.

# **Périmètre fonctionnel**

Délimité le périmètre fonctionnel de notre application, permet de savoir exactement où 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 de l’application. Pour cela, nous allons faire ce que nous appelons un **impact mapping.**

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

* Quel est l’objectif de mon application ?
* Quels sont les différents profils de ses utilisateurs ?
* Quelles actions cherchent-ils respectivement à faire sur mon application ?
* 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 de l’application. Remarquons que chaque colonne correspond à une des questions évoquées plus haut.

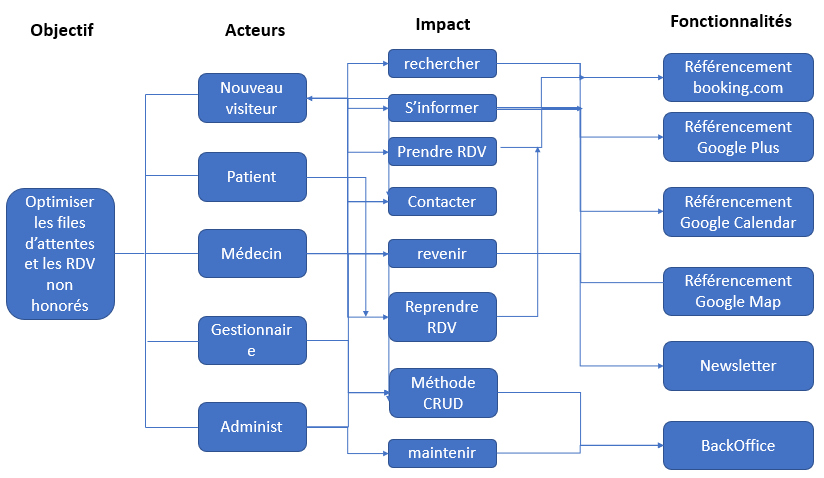


Schéma 1 : périmètre fonctionnel du logiciel

# **En Frontend**

|  |  |
| --- | --- |
| Fonctionnalités | Contraintes Associées |
| Réservation en ligne | Filtres |
| Formulaire de contact | Anti Spam |
| Google Map | Pas de contraintes particulières |
| Changement de langue | Toujours présent à l’écran |
| Inscription Newsletter | Pas de contraintes particulières |

Tableau 3 : fonctionnalités coté frontEnd

# **En Backend**

|  |  |
| --- | --- |
| Fonctionnalités | Contrainte(s) particulière(s) |
| Gestion des contenus | Restriction en fonction des rôles |
| Gestion des rendez-vous | Vue en temps réel de l’agenda du médecin |
| Gestion des prix | Système de paiement |

Tableau 3 : fonctionnalités coté frontEnd

# **Fonctionnalités**

# **Chez le patient**

|  |  |
| --- | --- |
| **Projet :** Mise en place d’un logiciel de prise de rendez-vous médicaux en ligne | Description cas d’utilisation |

|  |
| --- |
| Nom cas d’utilisation : Patient |
| Acteur déclencheur : Patient/nouvel utilisateur |
| Pre condition : avoir lancé l’application |
| Post condition : Néant |
| Scénario nominal :  1. L’application affiche la page d’accueil  2. L’utilisateur clique sur la zone de recherche  a) L’utilisateur saisi le nom, la spécialité ou la localité du praticien  b) Une liste de praticiens répondants au critère de recherche s’affiche  c) L’utilisateur sélectionne le praticien de son choix  3. La page contenant les informations du praticien avec le calendrier proposant la plage de disponibilité du praticien ainsi qu’un bouton prendre rendez-vous et un bouton annuler s’affichent  a) L’utilisateur clique sur l’agenda du praticien et sélectionne la date et l’heure disponible qui lui convient  b) L’utilisateur clique sur le bouton valider pour prendre un RDV ou sur le bouton annuler pour annuler. |
| Exceptions : |
| Contraintes : |

Les actions possibles

|  |  |  |
| --- | --- | --- |
| Scénario | Action | Résultat attendu |
| 1 | Lancement de l’application | Affichage de la page d’accueil avec une zone de recherche |
| 2 | L’utilisateur clique sur la zone de recherche | La liste de praticiens déroule lorsque l’utilisateur commence à saisir |
| 3 | L’utilisateur sélectionne le praticien de son choix | La page contenant les informations du praticien et son calendrier s’affichent avec les boutons pour valider ou annuler. |
| 4 | L’utilisateur ouvre le calendrier du praticien | Le calendrier s’affiche avec les plages de disponibilités du praticien. |
| 5 | L’utilisateur choisit la date et l’heure qui lui convient et cliquez sur le bouton valider | Un formulaire s’ouvre pour mettre ses informations afin de réserver un rendez-vous |
| 6 | Après avoir fini de remplir le formulaire | Une fenêtre d’authentification s’affiche pour que l’utilisateur s’authentifie s’il ne l’avait pas fait |
| 7 | L’utilisateur n’a pas de compte | Une fenêtre d’authentification avec un bouton créer un compte s’affiche. |
| 8 | L’utilisateur clique sur le bouton créer un compte | Un formulaire s’ouvre permettant à l’utilisateur de créer son compte. |
| 9 | L’utilisateur valide le formulaire de création de compte | Un mail d’activation de compte lui est envoyé dans l’adresse mail qu’il a fourni. |
| 10 | L’utilisateur part ouvrir sa boîte mail et clique sur le lien d’activation et revenir sur le logiciel pour valider sa prise de RDV | Validation approuvée |
| 11 | Si l’utilisateur clique sur le bouton annuler | Il est redirigé sur la page d’accueil |
| 12 | L’utilisateur clique sur l’onglet FAQ/Contact | Une page contenant une liste de questions prédéfinies avec leurs réponses et un formulaire de contact ou pour poser une nouvelle question, s’affiche. |
| 13 | L’utilisateur verra l’onglet mon Compte s’il est connecté et clique sur ce | Une page contenant les informations du patient s’affiche lui permettant d’effectuer des méthodes CRUD via des boutons à travers un formulaire. |

Tableau 3 : Spécifications fonctionnelles détaillées chez le patient

# **Chez le Médecin**

|  |  |
| --- | --- |
| Projet : Mise en place d’un logiciel de prise de rendez-vous médicaux en ligne | Description cas d’utilisation |

|  |
| --- |
| Nom cas d’utilisation : Médecin |
| Acteur déclencheur : Médecin/Administrateur |
| Pré condition : avoir lancé l’application |
| Post condition : s’authentifier |
| Scénario nominal :  1. L’application affiche la page d’accueil  2. Le médecin clique sur l’onglet rendez-vous en attente  d) Il clique sur chaque rendez-vous demandé afin de confirmer ou mettre en attente via des boutons  e) Il clique sur la liste des patients pour voir ses patients et leurs RDV  4. Il clique sur l’onglet agenda pour mettre à jour son agenda  5. Il clique sur l’onglet mon Compte pour mettre à jour ses informations  6. Il clique sur l’onglet FAQ pour voir les questions en attentes des utilisateurs. |
| Exceptions : |
| Contraintes : |

Les actions possibles

|  |  |  |
| --- | --- | --- |
| Scénario | Action | Résultat attendu |
| 1 | Lancement de l’application | Affichage de la page d’accueil avec une zone de recherche |
| 2 | Le médecin clique sur l’onglet connexion | Une fenêtre d’authentification s’affiche |
| 3 | Le médecin clique sur l’onglet rendez-vous en attente | Une page contenant la liste des patients ayant sollicité un rendez-vous s’affiche avec le motif ainsi que la date et l’heure proposées. |
| 4 | Le médecin clique sur un rendez-vous demandé afin de confirmer ou mettre en attente via des boutons | Les informations du patient s’affichent avec le motif ainsi que la date et l’heure proposées avec en dessous les boutons confirmer ou mettre en attentes. |
| 5 | Si le médecin clique sur le bouton confirmer | Un mail et/ou sms est automatiquement envoyé au patient pour confirmation de son RDV avec la date, l’heure et le lieu.  Le patient devra à son tour confirmer son rendez-vous soit 24h avant via un service USSD.  Et le patient se positionne dans l’onglet liste des patients. |
| 6 | Si le médecin clique sur le bouton mettre en attentes | Il ne se passe rien, le RDV restera toujours dans la liste des RDV en attente. |
| 7 | Le médecin clique sur la liste des patients (onglet patients) | Une page contenant la liste des patients les plus récents avec qui il a RDV, s’affiche. |
| 8 | Le médecin clique sur l’onglet agenda | Une interface proposant un agenda en ligne s’affiche lui permettant de le mettre à jour selon ses plages de disponibilités. |
| 9 | Le médecin clique sur l’onglet mon Compte | Une page contenant les informations du médecin s’affiche lui permettant d’effectuer des méthodes CRUD via des boutons a travers un formulaire. |
| 10 | Le médecin clique sur l’onglet FAQ/Contact | Une page contenant les questions posées par les patients s’affiche avec un formulaire pour répondre aux questions. |

Tableau 4 : Spécifications fonctionnelles détaillées chez le médecin

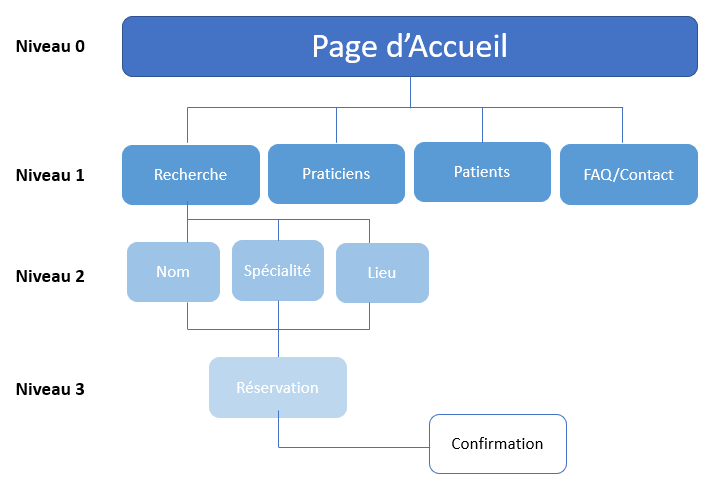


Schéma 2 : Arborescence du logiciel

# **CONCLUSION**

Dans ce chapitre, nous avons présenté une étude fonctionnelle détaillée de notre projet et nous avons exposé l’arborescence et le périmètre fonctionnel permettant de concevoir et de développer un système qui facilitera la gestion des rendez-vous médicaux.

Après avoir fixé nos objectifs, l’étape suivante sera consacrée à une conception détaillée des fonctionnalités du système.

# **Chapitre 3 :** **ETUDE CONCEPTUELLE**

# **Introduction**

Dans ce chapitre nous commençons par une présentation des différents outils logiciels et les langages de modélisations utilisés. Ensuite nous détaillons les diagrammes des cas d’utilisation, les diagrammes des classes et les diagrammes de séquences.

# **Méthodes et Outils de Modélisation**

Pour la conception de notre système nous avons adopté une méthode orientée objet. En effet cette dernière est une approche incontournable dans le cadre du développement des applications.

Pour mieux présenter l’architecture de notre système, on va choisir le langage de modélisation le plus adopté UML :

1. **Le langage de modélisation (UML)**

Le langage de modélisation unifié (UML) est un langage de modélisation normalisé permettant aux développeurs de spécifier, visualiser, construire et documenter les artefacts d'un système logiciel. Ainsi, UML rend ces artefacts évolutifs, sécurisés et robustes. UML est un aspect important impliqué dans le développement de logiciels orientés objet. Il utilise la notation graphique pour créer des modèles visuels de systèmes logiciels [1]

1. **L’outil de Modélisation**

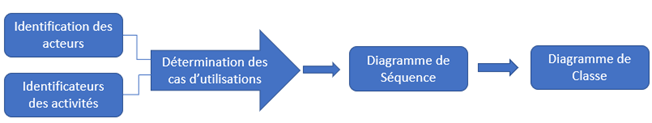
**Astah** , anciennement connu sous le nom de JUDE ( [environnement de](https://en.wikipedia.org/wiki/Java_(programming_language)) développement [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) et [UML](https://en.wikipedia.org/wiki/Unified_Modeling_Language) , se prononce Judo, est un [outil de modélisation UML](https://en.wikipedia.org/wiki/UML_tool) créé par la société japonaise Change Vision. JUDE a reçu le prix "Produit logiciel de l'année 2006", créé par l'Agence japonaise de promotion de la technologie de l'information. [2]



Figure 1 : logo du logiciel Astah Professionnel

# **Processus Unifié (UP)**

Le processus unifié (UP) utilise le langage UML (Unified Modeling Language). Il semble être la solution idéale pour remédier à l'éternel problème des développeurs. En effet, il regroupe les activités à mener pour transformer les besoins d'un utilisateur en un système logiciel quel que soit la classe, la taille et le domaine d'application de ce système.

 Schéma 3 : Organigramme de processus UP

# **Définition des acteurs**

Un acteur est une « entité » externe au système qui interagit avec le système. La notation UML de l’acteur est soit le dessin « simplifié » d’un personnage complété en dessous par un libellé soit le dessin d’un rectangle contenant le libellé du nom de l’acteur en dessous du libellé du stéréotype « « acteur » ». La notation graphique que nous préférons est bien sûr le dessin du personnage, plus facile à repérer dans un diagramme. Un acteur n’est pas toujours une personne, ce peut être un système externe en interaction avec le système en cours de description. [3]

Dans notre projet, on distingue 3 acteurs principaux :

* Médecin
* Patient
* Administrateur

# **Diagramme de Cas d’utilisation**

1. **Définition**

Un diagramme de cas d’utilisation permet de représenter graphiquement les cas d’utilisation d’un système. C'est le diagramme principal du modèle UML, celui où la relation entre l'utilisateur et les objets est mise en en œuvre.

Un cas d'utilisation est utilisé pour définir le comportement d'un système ou la sémantique de toute autre entité sans révéler sa structure interne. Chaque cas d'utilisation spécifie une séquence d'action, y compris des variantes, que l'entité réalise, en interagissant avec les acteurs de l'entité. La responsabilité d'un cas d'utilisation est de spécifier un ensemble d'instances, où une instance de cas d'utilisation représente une séquence d'actions que le système réalise et qui fournit un résultat observable par l'acteur. [4]

Voici les cas d'utilisation de mon système :

1. **Authentification**

L’application vérifie que l'utilisateur est bien ce qu'il prétend être et lui donne ensuite l'autorisation d'accès. L’authentification est incluse dans le cas d’utilisation de chaque acteur.

1. **Diagramme de cas d’utilisation Administrateur**

L’administrateur du système peut gérer les spécialités, en ajoutant une nouvelle spécialité ou/et en modifiant une qui existe déjà.

De plus l’administrateur est le seul qui a le droit d’accepter ou rejeter un médecin.

En effet, après la vérification de la carte d’identité envoyée par le médecin, l’administrateur a la possibilité de valider ou bien de refuser l’inscription de ce dernier.

Ci-dessous le diagramme de cas d’utilisation de l’administrateur :

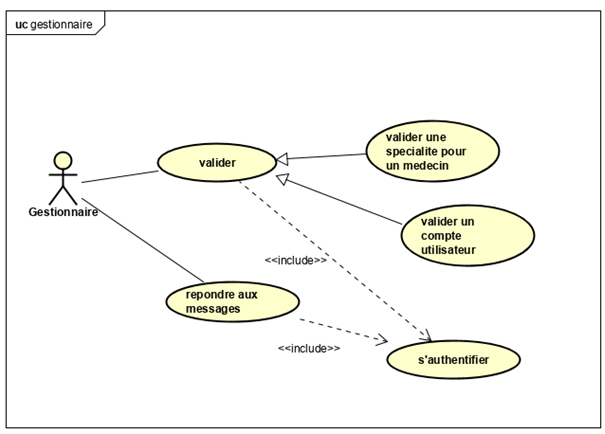


Schéma 4 : Diagramme de cas d'utilisation de l’administrateur

1. **Diagramme de cas d’utilisation du Patient**

Le patient est l’utilisateur potentiel de l’application, Pour pouvoir accéder aux différentes fonctionnalités de l’application, le patient doit se connecter s’il possède déjà un compte, sinon il doit créer un compte.

Un patient peut chercher un médecin directement par son nom ou par sa spécialité.

Une fois le médecin a été sélectionné, le patient peut prendre un rendez-vous selon la disponibilité du médecin.

Après avoir pris un rendez-vous, le patient peut consulter la liste de ses rendez-vous (acceptés, refusés et en attente). Tant que le rendez-vous est encore en attente, le patient a la possibilité de l’annuler.

Le patient peut aussi changer les informations de son compte s’il en possède.

Ci-dessous, le diagramme de cas d’utilisation du patient :

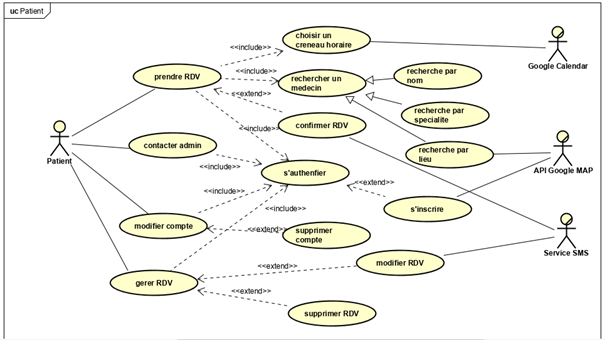


Schéma 5 : Diagramme de cas d'utilisation du Patient

1. **Diagramme de cas d’utilisation du Médecin**

Le médecin dans le système peut gérer ses disponibilités, il peut ajouter ou modifier ses jours et heures de travail.

De plus, le médecin peut accepter ou refuser un rendez-vous.

Ci-dessous, le diagramme de cas d’utilisation du médecin :

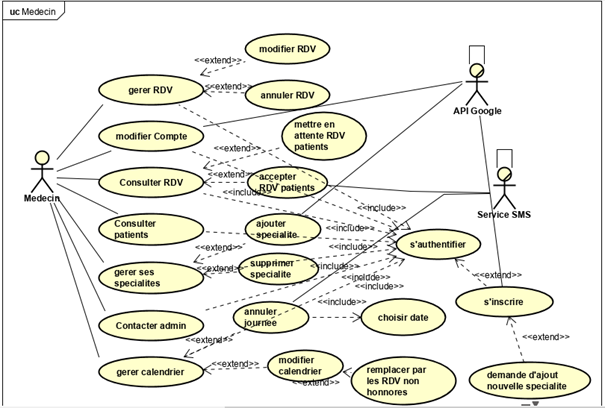


Schéma 6 : Diagramme de cas d'utilisation du médecin

# **Diagramme de Séquence**

# **Définition**

Les diagrammes de séquence permettent de représenter les interactions entre objet selon un point de vue temporel. L’accent est mis sur la chronologie des envois de messages.

* **Scénario :** une liste d’actions qui décrivent une interaction entre un acteur et le système.
* **Interaction :** Un comportement qui comprend un ensemble de messages échangés par un ensemble d’objet s dans un certain contexte pour accomplir une certaine tâche.
* **Message :** Un message est une transmission d'information unidirectionnelle entre deux objets, l'objet émetteur et l'objet récepteur.

Ci-dessous nos diagrammes de Séquence :

# **Diagramme de Séquence d’Authentification**

L’application vérifie que l'utilisateur est bien ce qu'il prétend être et lui donne ensuite l'autorisation d'accès.

Ci-dessous, le diagramme de cas d’utilisation de l’authentification :

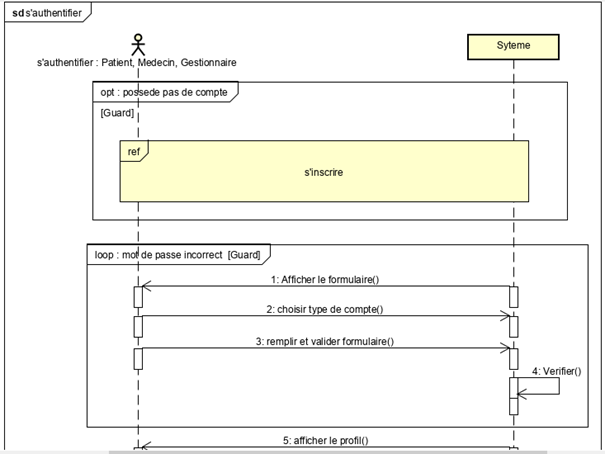


Schéma 7 : Diagramme de Séquence de l’Authentification

# **Diagramme de Séquence de prise de rendez-vous**

Pour prendre un rendez-vous, le patient sélectionne la date et l’heure du rendez-vous souhaité.

Après la vérification de la disponibilité du médecin, le rendez-vous est enregistré dans la base de données et un message de confirmation s’affiche au patient. Si la date et/ou l’heure sélectionnée ne correspondent pas à la disponibilité du médecin, le système envoie un message d’erreur au patient pour lui informer que son rendez-vous n’a pas été enregistré et lui demande de choisir une autre date.

Ci-dessous, le diagramme de séquence de la prise de rendez-vous :

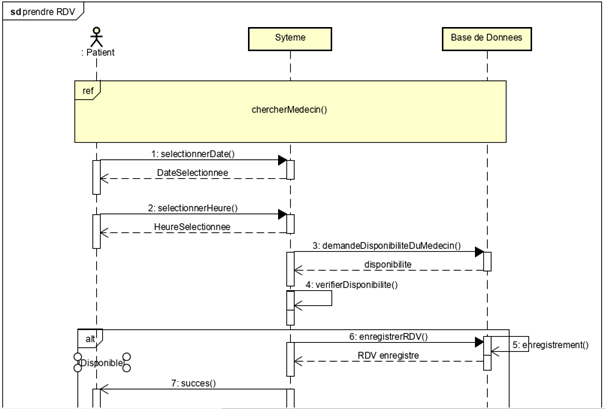


Schéma 8 : Diagramme de Séquence de prise de rendez-vous

# **Diagramme de Séquence du Patient**

Pour accéder à l’application, le patient doit avoir un compte. Pour créer un compte, le patient doit saisir les informations demandées correctement puis le système vérifie ces informations et si n’a pas des erreurs le compte sera créé et les informations seront enregistrées dans la base de données.

Si le patient n’a pas de compte, il doit en créer un.

Le patient peut chercher un médecin par son nom, sa spécialité ou par sa localisation sur la carte.

La localisation des médecins sur la carte est disponible en deux modes : terrain et satellite. Le patient peut choisir celui qui lui convient le mieux, c’est une fonctionnalité gérée par l’API de Google adaptée pour un usage facile et rapide.

Le patient peut choisir un médecin directement en cliquant sur son marqueur sur la carte. Pour chercher un médecin par spécialité, il suffit de choisir une des spécialités existantes dans la base de données et le système affiche la liste des médecins de cette spécialité.

En plus, le patient a la possibilité de chercher un médecin par son nom. Pour cela, il suffit de saisir un nom et le système affiche la liste des médecins avec ce nom.

Ci-dessous, le diagramme de séquence de la prise du patient :

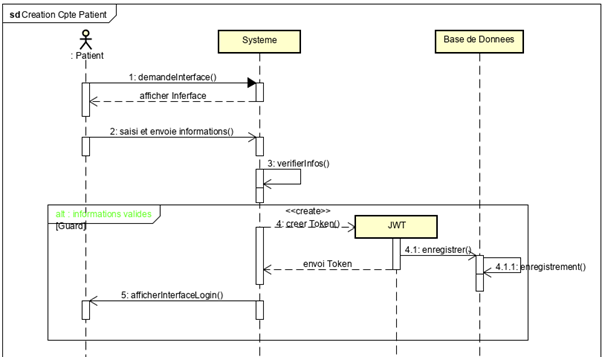


Schéma 9 : Diagramme de Séquence du Patient

# **Diagramme de Classe**

# **Définition**

Les diagrammes de class sans doute c’est les diagrammes les plus utiliser d’UML ils décrivent les types des objets qui composent un système et les différents types de relation statique qui existent entre eux. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques.

Dans notre diagramme de classes on a trois classes (administrateur, médecin et patient) qui hérite d’une classe mère nommé utilisateur.

Un patient peut prendre un rendez-vous avec un médecin, et ce dernier peut accepter ou refuser le rendez-vous selon son horaire de travail, si le rendez-vous est accepté il faut avoir un dossier patient qui contient les observations et les ordonnances de rendez-vous.

Un médecin admet une spécialité gérée par l’administrateur du système. L’administrateur valide le compte d’un médecin après la vérification de sa carte d’identité nationale.

Ci-dessous le diagramme de classe du système :

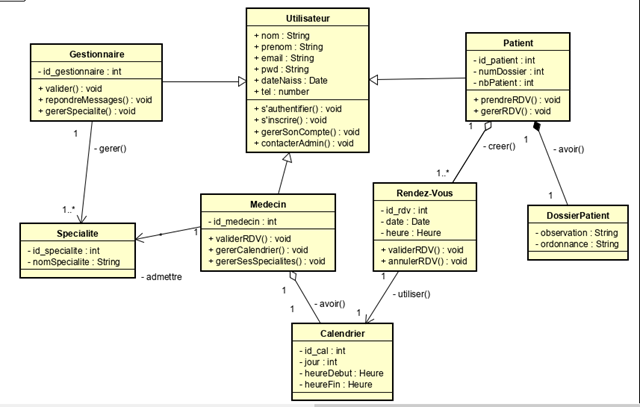


Schéma 10 : Diagramme de classe du système

# **Conclusion**

Dans ce chapitre, nous avons présenté une étude conception détaillée des fonctionnalités notre projet.

Après avoir fixe nos objectifs, l’étape suivante sera consacrée à l’implémentation du logiciel.

# **Chapitre 4 : Réalisation de l’Application**

# **Introduction**

Ce chapitre représente le dernier volet de ce rapport, il sera consacré à l’implémentation de notre système. Nous commençons par la présentation des ressources matérielles et logicielles utilisées. Nous passons ensuite à présenter des captures d’écran dans le but de mettre en évidence l’aspect ergonomique et fonctionnel des interfaces développées.

# **Architecture du système et environnement de test**

1. **Pourquoi utiliser un Framework ?**

Un Framework est un ensemble d'outils et de composants logiciels organisés conformément à un plan d'architecture et des patterns, l'ensemble formant ou promouvant un « squelette » de programme. Il est souvent fourni sous la forme d'une bibliothèque logicielle, et accompagné du plan de l'architecture cible du Framework. [4]

L’objectif essentiel d’un Framework est d’aider les développeurs dans la réalisation de leurs tâches. L’organisation et la structure du Framework vise la productivité maximale du développeur qui va l’utiliser et facilite la maintenance du logiciel.

Voici les trois raisons à utiliser un Framework :

* Rapidité : le Framework permet un gain de temps et une livraison beaucoup plus rapide qu’un développement à zéro.
* Organisation : l’architecture d’un Framework favorise une bonne organisation du code source (modèle MVC par exemple).
* Maintenabilité : l’organisation du Framework facilite la maintenance du logiciel et la gestion des évolutions.

Voici les trois Framework que nous avons utilisé pour développer notre application :

1. **Le Framework Ionic**

Le Framework Ionic est une boîte à outils d'interface utilisateur mobile gratuite et à code source ouvert permettant de développer des applications multiplateformes de haute qualité pour iOS, Android et le Web, le tout à partir d'une seule base de code. [5]



Figure 2 : logo du Framework Ionic

1. **Le Framework Angular**

Angular est un cadriciel côté client open source basé sur TypeScript dirigée par l'équipe du projet Angular à Google et par une communauté de particuliers et de sociétés. Angular est une réécriture complète de AngularJS, cadriciel construit par la même équipe. [6]



Figure 3 : logo du Framework Angular

1. **Le Framework Spring boot**

Spring est un Framework libre pour construire et définir l'infrastructure d'une application java, dont il facilite le développement et les tests. [7]



Figure 3 : logo du Framework Spring

# **Environnement de développement**

Pour mettre en place notre système, nous avons utilisé un environnement de développement qui a assuré le bon déroulement de la phase implémentation. Cet environnement comporte des outils matériels ainsi que logiciels.

1. **Environnement matériel**

Pour le développement de notre application nous avons utilisé un PC portable « DELL Latitude E72 » dont la configuration est la suivante :

* Processeur Intel(R) Core(TM) i5-4200U avec fréquence 2.3 GHz
* Quantité de mémoire vive 8 Go
* Capacité du disque dur 500 Go De plus, pour tester notre application.

1. **Environnement logiciel** 
   1. **WebStorm**

WebStorm est un IDE pour les langages Web, développé par l'entreprise JetBrains et basé sur la plateforme IntelliJ IDEA. L'IDE fournit les meilleurs codes auto-complétion, sur la volée de prévention d'erreur, refactoring puissants pour JavaScript, TypeScript, les langages de feuille de style et tous les frameworks les plus courants, soutient le mélange de langues et plus. Il traite automatiquement le code avec précaution, pour aider les développeurs à effectuer des changements globaux du projet facilement et en toute sécurité. [8]



Figure 4 : logo de l’environnement de développement WebStorm

* 1. **STS Spring Tools Suite**

Spring Tool Suite (STS) est un IDE étendu pour Eclipse. Il se spécialise dans le développement des applications Spring. Il permet d’intégrer efficacement Spring dans l’environnement de développement. [9]



Figure 5 : logo de l’environnement de développement STS

* 1. **Robot 3T (MongoDB)**

Robo 3T est une interface graphique qui permet la prise en charge de MongoDB 4.0 et SCRAM-SHA-256, un shell mongo mis à niveau, ainsi que la prise en charge de l’importation à partir de chaînes de connexion MongoDB SRV, parmi de nombreux autres correctifs et améliorations.

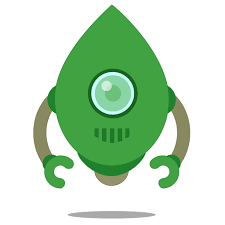
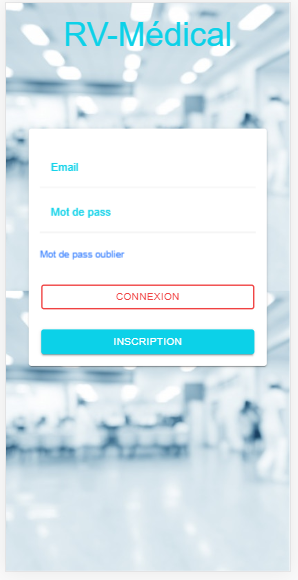
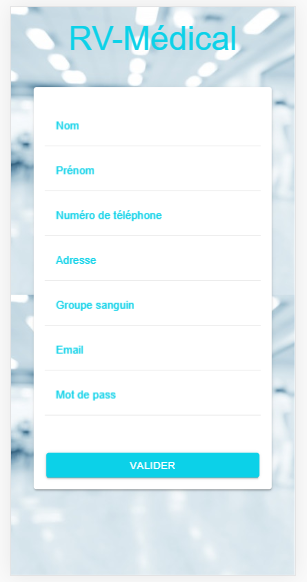


Figure 6 : logo du MongoDB et de son outil d’interface graphique Mongo 3T

# **Description des interfaces réalisées**

1. **Application mobile**
   1. **Interface d’authentification**

L’interface d’authentification est une des interfaces les plus importantes dans l’application, car permet aux utilisateurs qui se sont enregistrés dans notre système de profiter des fonctionnalités qui leur sont attribuées. A travers cette interface l’utilisateur donne son login et son mot de passe. Si cette combinaison correspond aux informations qui existent dans la base de données, l’application le redirige vers l’interface de recherche des médecins sinon un message d’erreur apparait.



Lorsque l’utilisateur n’a pas de compte il va en créer un et revenir se connecter pour accéder à la page d’accueil

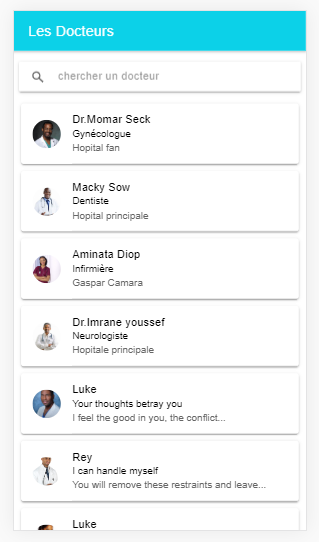
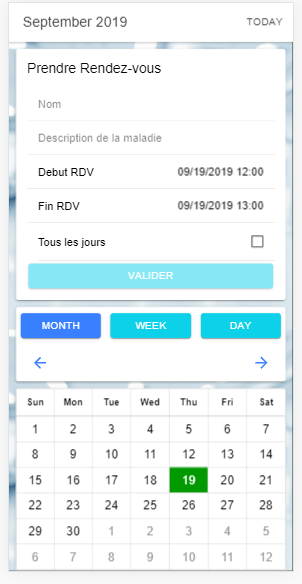


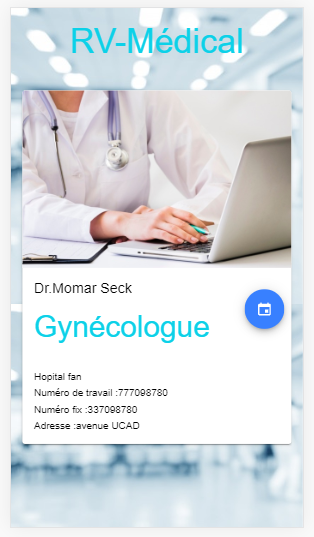
Une fois l’utilisateur connecté, il accède à la page d’accueil

Capture 1 : Interface de connexion et d’accueil de l’application mobile

* 1. **Interface de recherche d’un médecin**

A travers cette interface de recherche d’un médecin, l’utilisateur a la possibilité de chercher un médecin par nom ou par spécialité comme le montre la figure suivante :





Capture 2 : Interface de recherche de médecin de l’application mobile

Cette interface présente le profil du médecin, dans laquelle le médecin peut trouver tous ses coordonnées. A partir de cette interface le médecin peut changer sa biographie, sa photo de profil, sa position sur la carte et ses jours de travail.

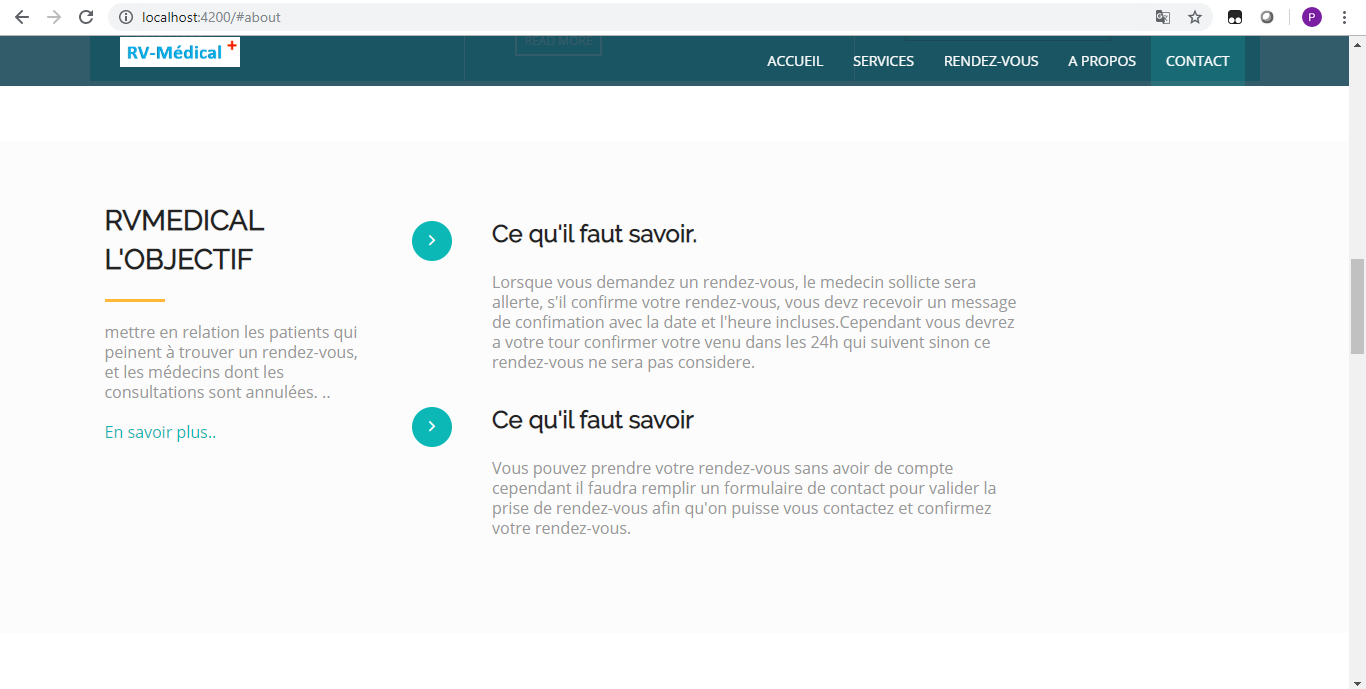
1. **Application web**
   1. **Interface prise de rendez-vous**

Comme dans l’application mobile, l’application web propose une interface de prise de rendez-vous sans pour autant se connecter à l’application.

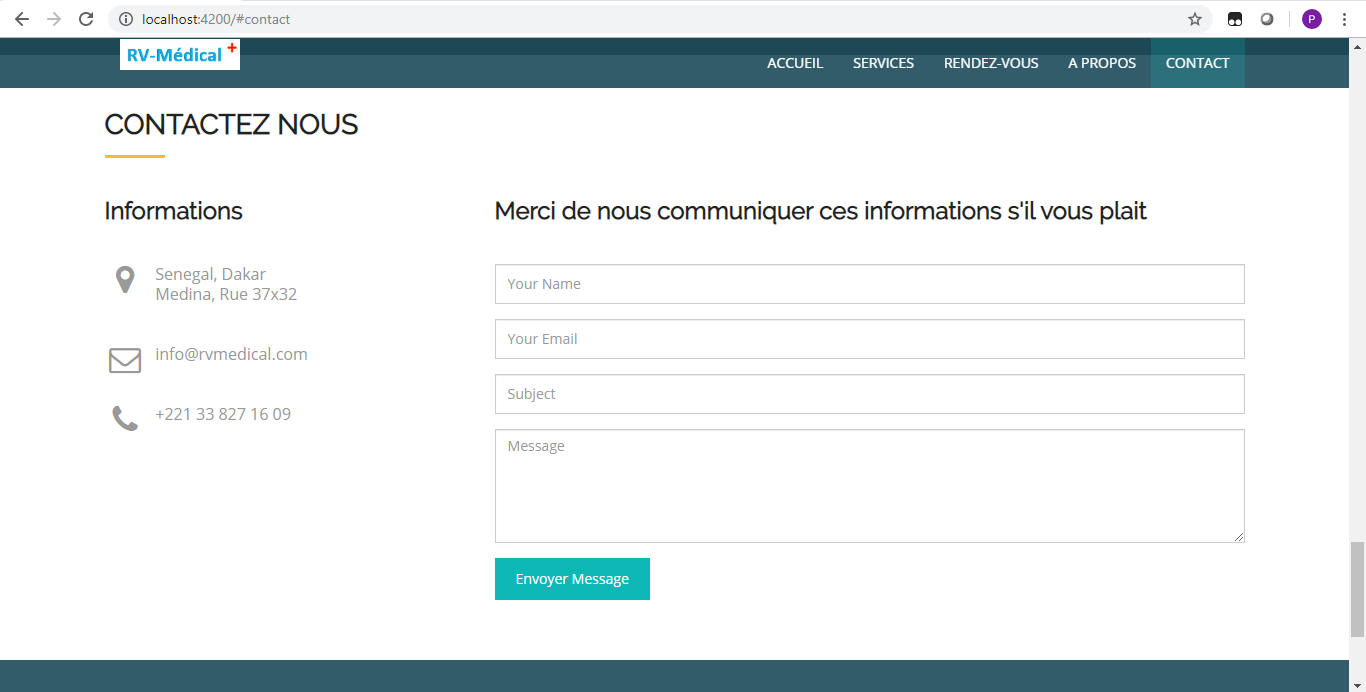
Ci-dessous la page d’accueil de l’application web avec un bouton prendre rendez-vous :



Capture 3 : interface d’accueil de l’application web



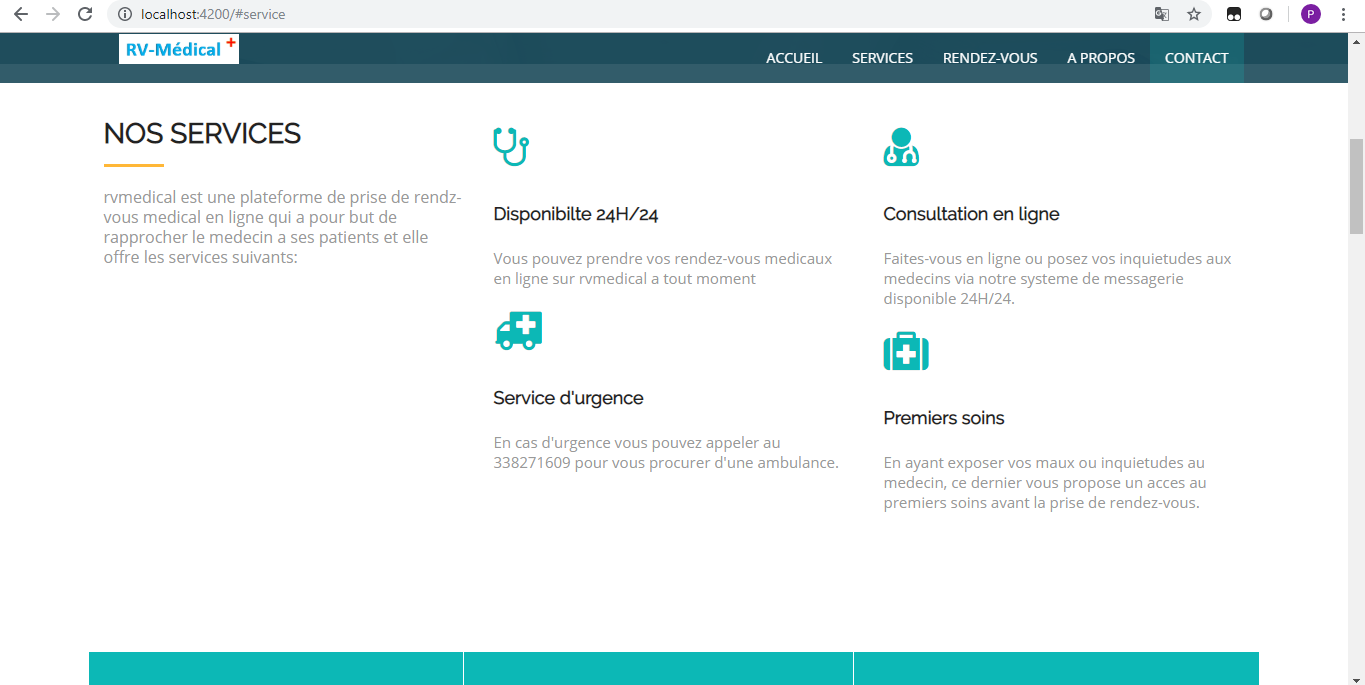
Capture 4 : interface à propos de l’application web



Capture 5 : interface de contact de l’application web



Capture 6 : interface prise de rendez-vous de l’application web



Capture 7 : interface prise services de l’application web

# **Conclusion**

Au cours de ce chapitre, nous avons décrit les plateformes matérielles et logicielles sur lesquelles nous avons développé notre système. Nous avons ensuite présenté les deux applications de notre système de gestion des rendez-vous médicaux à travers quelques interfaces que nous avons développées.

# Conclusion Générale

Ce projet de fin d’études, a pour objectif de réaliser un système de gestion des rendez-vous médicaux "RV-Médical" permettant d’une part aux patients de chercher un médecin et prendre un rendez-vous en utilisant leurs Smartphones ou PC, et d’autre part aux médecins de gérer les rendez-vous et les dossiers des patients. De ce fait, notre système engloba plusieurs médecins de différentes spécialités. Cela nous a permis de mettre à la disposition des utilisateurs une riche base de données de médecins qu'ils puissent y avoir recours en cas de besoin. La réalisation de ce travail a passé par trois phases essentielles. En premier lieu, nous avons effectué une étude préalable qui nous a permis d’identifier les besoins de notre application. En deuxième lieu, nous avons entamé la partie conception. Enfin, nous avons présenté les outils de développement adoptés et les résultats obtenus. Finalement, notre application peut être améliorée et enrichie par de nouvelles fonctionnalités.

Webographie

1. <https://www.techopedia.com/definition/3243/unified-modeling-language-uml>
2. [https://en.wikipedia.org/wiki/Astah\*](https://en.wikipedia.org/wiki/Astah*)
3. <http://www-inf.int-evry.fr/cours/CSC4002/EnLigne/Cours/CoursUML/4.12.6.html>
4. <https://solucase.com/pourquoi-utiliser-un-framework-web/>
5. <https://ionicframework.com/>
6. <https://fr.wikipedia.org/wiki/Angular>
7. <https://fr.wikipedia.org/wiki/Spring_(framework)>
8. <https://www.jetbrains.com/webstorm/>
9. <http://igm.univ-mlv.fr/~dr/XPOSE2012/spring/spring.html>

**TABLE DES MATIÈRES**



# Résumé

Mise en place d’un logiciel de prise de rendez-vous médicaux en ligne

Notre projet consiste dans l’étude, la conception et la réalisation d’une plateforme de prise de rendez-vous médicaux en ligne où nous avons créé une application web et mobile.

Cette application permet de fournir des soins médicaux de qualité aux patients en réunissant tous les médecins sur une plate-forme unique afin que tout le monde puisse facilement y accéder et prendre des rendez-vous. L’idée est de rapprocher le médecin de ses patients.

Pour atteindre ces objectifs, nous avons eu recours à créer une plateforme web (basé sur du Angular 8 coté client et Spring boot coté serveur) et mobile basé sur Ionic 4, MongoDB comme système de base de données, une étude conceptuelle réalisée par le langage UML avec comme outils de modélisation Astah Professionnel, WebStorm et STS comme éditeur de code.