

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de Master en Informatique

Option: Système d'Information et de Connaissances (S.I.C)

Thème

Application de Gestion de Clinique Médicale

Réalisé par :

- **BEKKAL BRIKCI MOHAMMED EL HADI**

Présenté le 26 juin 2025 devant la commission composé de MM.

Président : **BENMANSOUR ASMA**

Encadreur : **EL YEBDRI ZEYNEB**

Examineur : **KHITRI SOUAD**

Année universitaire : 2024-2025

Remerciements

D'abord et avant tout, merci à Allah le tout miséricordieux, de nous avoir donné la force, la patience et le courage nécessaire pour finaliser toutes ces longues années d'études.

À toutes les personnes qui nous ont aidé durant la réalisation de ce travail, veuillez trouver dans ces lignes l'expression de notre reconnaissance et notre profonde gratitude.

Nous tenons à remercier aussi notre encadrante Mme Zeyneb ELYEBDRI qui nous a aidé à mener à bien notre travail grâce à tous ses conseils et son précieux soutien.

L'expression de nos remerciements les plus sincères vont au président ainsi qu'aux membres de notre jury pour avoir accepté de lire notre mémoire et d'évaluer notre travail.

Merci à tous les enseignants du département informatique qui ont fournis des efforts pour nous donner leurs savoir durant tout notre cursus universitaire.

Enfin, merci à nos parents et à tous les membres de nos familles respectives de nous avoir apporté leur soutien tout au long de cette démarche.

Dédicaces

En premier lieu, je remercie Dieu, le tout puissant de m'avoir donné la force pour réaliser ce modeste travail, que je dédie :

À mes chers parents qui n'ont jamais arrêté de m'encourager et de me soutenir durant toutes les épreuves que j'ai passé dans ma vie ainsi que pour tous les sacrifices qu'ils ont consentis pour ma réussite, que Dieu leur accorde santé, bonheur et prospérité.

À mes très chers frères et mes deux belles soeurs pour leur aides et leur encouragements.

À mes deux neveux Nazim et Charaf Eddine que dieu les bénisse.

À tous les membres de ma famille qui se reconnaîtront dans ce message.

À tous mes amis ainsi qu'à tous les gens que j'ai côtoyé durant mon cursus scolaire.

BEKKAL BRIKCI Mohammed El-Hadi

Abréviations

UML	<i>Unified Modeling Language</i>
UP	<i>Unified Process</i>
DCU	<i>Diagramme de Cas d'Utilisation</i>
UC	<i>Use case/ Cas d'utilisation</i>
DSS	<i>Diagramme de Séquence Système</i>
DC	<i>Diagramme de Classes</i>
VSC	<i>Visual Studio Code</i>
HTML	<i>Hypertext Markup Language</i>
CSS	<i>Cascading Style Sheets</i>
JS	<i>JavaScript</i>
API	<i>Application Programming Interface</i>
Bdd	<i>Base de Données</i>
SQL	<i>Structured Query Language</i>
JWT	<i>JavaScript Object Notation Web Token</i>
CRUD	<i>Create Read Update Delete</i>
SGCM	<i>Système de Gestion de Clinique Médicale</i>
RGPD	<i>Règlement Général sur la Protection des Données</i>
WEB	<i>World Wide Web</i>
JSP	<i>JavaServer Pages</i>
RH	<i>Ressources Humaines</i>
MVC	<i>ModelViewController</i>
PHP	<i>Personal Home Page</i>
UI	<i>User Interface</i>

Table des figures

1.1	Organigramme Clinique	14
2.1	Diagramme de Cas d'Utilisation Médecin	23
2.2	Diagramme de Cas d'Utilisation Secrétaire	24
2.3	Diagramme de Cas d'Utilisation Admin	25
2.4	Diagrammes de Séquence Système Médecin	26
2.5	Diagrammes de Séquence Système Secrétaire	27
2.6	Diagrammes de Séquence Système Admin	27
2.7	Diagramme de Classes	28
3.1	Logo des outils utilisés	30
3.2	Fonctionnement du JWT	33
3.3	Architecture du système	34
3.4	Interface de Pré-login	35
3.5	Interface de Connexion	35
3.6	Interface des Rendez-vous de la journée	36
3.7	Interface de la liste des Patients	36
3.8	Interface pour voir et modifier un patient	37
3.9	Interface pour prendre un rendez-vous	37
3.10	Interface de la liste des Médecins	38
3.11	Interface pour voir les détails d'un Médecin	38
3.12	Interface de mise à jour du mot de passe	39
3.13	Interface de mise à jour du profil	39
3.14	Interface des Rendez-vous du médecin	40
3.15	Interface de la Liste des Patient	40
3.16	Interface des détails d'un Patient	41
3.17	Interface du dossier médical d'un patient	41
3.18	Interface des Ordonnances d'un Patient	42
3.19	Interface des Lettres pour un Patient	42
3.20	Interface pour reprendre un rendez-vous pour un Patient	43
3.21	Interface des Ordonnances du Médecin	43
3.22	Un Certificat de Bonne Santé	44
3.23	Un Certificat d'Arrêt de travail	44
3.24	Un Certificat de Reprise de Travail	45

3.25	Interface des Notifications pour le Médecin	45
3.26	Interface de la liste des rendez-vous de toute la clinique	46
3.27	Interface de liste des Médecins	46
3.28	Interface de mise à jour des Médecins	47
3.29	Interface de création des Médecins	47
3.30	Interface de liste des Secrétaires	48
3.31	Interface de mise à jour des Secrétaires	48
3.32	Interface de création des Secrétaires	49

Liste des tableaux

1.1	Comparaison des systèmes de gestion de cliniques médicales existants selon leur architecture, fonctionnalités et limitations.	17
-----	---	----

Table des matières

Introduction Générale	11
1 Etude de l'Existant	12
1.1 Introduction	13
1.2 Présentation d'une Clinique Médicale	13
1.2.1 Historique des cliniques médicales	13
1.2.2 Organisation générale d'une clinique	13
1.2.3 Organigramme de la Clinique	14
1.3 Étude de l'existant	15
1.3.1 Objectif	15
1.3.2 Création d'un système de gestion de clinique	15
a) Définition générale	15
b) Objectifs principaux du système	15
c) Modules fonctionnels à maîtriser avant de développer l'application	15
1.3.3 Aspects techniques à prendre en compte	16
a) Compétences nécessaires pour concevoir un SGCM	16
1.4 Systèmes existants	17
1.5 Besoins	18
1.5.1 Fonctionnels	18
a) Authentification et rôles	18
b) Gestion des administrateurs	18
c) Gestion des médecins	18
d) Interface utilisateur	18
e) Déconnexion	18
1.5.2 Non Fonctionnels	18
a) Sécurité	18
b) Compatibilité	19
c) Performance	19
d) Extensibilité	19
e) Ergonomie	19
1.6 Conclusion	19
2 Conception détaillée de notre système Alliance Médicale	20
2.1 Introduction	21

2.2	Le langage de modélisation UML	21
2.3	La méthode UP	22
2.4	Conception	22
2.4.1	Diagrammes de Cas d'Utilisation	23
	a) Pour le Médecin	23
	b) Pour la Secrétaire	24
	c) Pour l'Admin	25
2.4.2	Diagrammes de Séquence Système	26
	a) Pour le Médecin	26
	b) Pour la Secrétaire	27
	c) Pour l'Admin	27
2.4.3	Diagrammes de Classes	27
2.4.4	Conclusion	28
3	Mise en uvre de Alliance Médical	29
3.1	Introduction	30
3.2	Présentation des logiciels et outils de travail	30
3.2.1	Nice Page	30
3.2.2	GitHub	31
3.2.3	Modelio	31
3.2.4	Visual Studio Code (VSC)	31
3.3	Présentation des technologies pour la programmation	32
3.3.1	Laravel pour le back-end web	32
3.3.2	Blade pour la gestion des vues côté client	32
3.3.3	MySQL pour la gestion de la base de données	32
3.3.4	Laragon pour lenvironnement de développement	32
3.4	Présentation des outils de sécurisation de données	32
3.4.1	Le JavaScript Object Notation Web Token	32
3.4.2	Fonction de Hachage	34
3.4.3	Middleware d'authentification	34
3.5	Réalisation	34
3.5.1	Architecture du système Utilisée	34
3.5.2	Présentation des interfaces de notre application web Alliance Médical	35
	a) Interface de Connexion	35
	b) Interfaces de la Secrétaire	36
	c) Interfaces du Médecin	40
	d) Interfaces d'Admin	46
3.5.3	Conclusion	49
	Conclusion Générale	50
	Bibliographie	51

Introduction Générale

Le domaine médical représente l'un des piliers fondamentaux de toute société moderne. Il regroupe l'ensemble des activités liées à la prévention, au diagnostic, au traitement et au suivi des maladies et des affections humaines. Au cœur de ce système se trouvent les établissements de soins tels que les cliniques médicales, qui jouent un rôle crucial en offrant des services de santé de proximité, souvent spécialisés, et avec un encadrement plus personnalisé que dans les hôpitaux généraux.

Les cliniques médicales sont souvent dotées d'équipes pluridisciplinaires, comprenant des médecins généralistes et spécialistes, des infirmiers, des techniciens de laboratoire, et du personnel administratif. Elles assurent non seulement la prise en charge des patients, mais aussi la gestion de nombreuses tâches administratives : dossiers médicaux, rendez-vous, facturation, gestion des ressources humaines, commandes de matériel, etc. Cette complexité a rendu l'informatisation du secteur médical non seulement souhaitable, mais indispensable.

La question qui nous intéresse est comment est entrée l'informatique dans ce domaine ?

L'évolution rapide des technologies de l'information a permis d'introduire l'informatique dans le domaine médical, transformant radicalement la manière dont les soins sont administrés et les services gérés. Cette transformation s'est matérialisée à travers le développement de systèmes informatiques de gestion médicale. Ces systèmes permettent d'automatiser et d'optimiser de nombreuses fonctions, telles que :

La gestion des dossiers médicaux électroniques.

La planification des rendez-vous.

La facturation et la gestion des assurances.

La gestion des ressources humaines et matérielles.

Les systèmes de communication interne.

Chapitre 1

Etude de l'Existant

1.1 Introduction

L'étude de l'existant constitue la première étape essentielle du cycle de vie de tout projet informatique. Elle permet de comprendre en profondeur l'environnement actuel du système à informatiser, d'identifier les acteurs impliqués, d'analyser les méthodes de travail utilisées et de repérer les insuffisances et les besoins concrets. Cette phase est indispensable pour garantir une conception efficace et adaptée à la réalité du terrain.

Dans ce premier chapitre, nous allons nous intéresser au fonctionnement d'une clinique médicale, en mettant l'accent sur les différents services qui y interviennent : accueil des patients, consultation médicale, gestion des dossiers médicaux ainsi que la facturation et la gestion administrative. L'objectif est d'avoir une vision claire du déroulement des activités et de leur organisation interne afin de cerner les limites des méthodes traditionnelles.

Par la suite, nous établirons un cahier des charges qui définira le cadre général de notre étude. Ce document précisera les objectifs du projet, les fonctionnalités attendues de l'application web, et recensera les besoins des différents utilisateurs du système (médecins, secrétaires, personnel administratif, patients). Cette étape servira de base solide à la conception et au développement d'un système d'information capable de répondre aux exigences d'une gestion moderne, fiable et sécurisée d'une clinique médicale.

1.2 Présentation d'une Clinique Médicale

1.2.1 Historique des cliniques médicales

Les cliniques médicales trouvent leur origine dans le besoin de offrir des soins de santé plus accessibles, personnalisés et spécialisés en dehors des grands hôpitaux publics. À la fin du XIX^e et au début du XX^e siècle, avec les progrès de la médecine moderne et l'urbanisation croissante, des établissements privés ont commencé à apparaître dans de nombreuses villes à travers le monde. Leur mission principale était de fournir des soins ambulatoires, des consultations spécialisées, et parfois des actes chirurgicaux, dans un cadre plus souple que les hôpitaux traditionnels.

Au fil des décennies, les cliniques ont évolué pour devenir des structures médicales complètes, souvent dotées d'équipements technologiques avancés, tout en maintenant une taille humaine qui permet une relation plus directe avec le patient. Aujourd'hui, les cliniques jouent un rôle central dans les systèmes de santé modernes, en contribuant à la désaturation des hôpitaux et en favorisant une médecine de proximité.

1.2.2 Organisation générale d'une clinique

Une clinique médicale est structurée de manière à assurer un fonctionnement fluide et coordonné entre les différents services médicaux, paramédicaux et administratifs. Les services y sont généralement regroupés selon les fonctions suivantes :

- Pôle médical
 - Consultation générale : assurée par des médecins généralistes.
 - Consultations spécialisées : cardiologie, dermatologie, gynécologie, ORL, etc.
 - Urgences (dans certaines cliniques).

- Pôle paramédical
 - Soins infirmiers
 - Laboratoire danalyses médicales
 - Service dimagerie médicale : radiologie, échographie, scanner.
- Pôle administratif
 - Accueil et admission des patients
 - Secrétariat médical
 - Service de facturation et gestion des dossiers
 - Ressources humaines
 - Service informatique
- Autres services :
 - Pharmacie interne
 - Hygiène et entretien
 - Gestion du matériel biomédical

1.2.3 Organigramme de la Clinique

[1] L'organigramme qui suit (figure 1.1) présente un organigramme fonctionnel simplifié d'une clinique médicale :

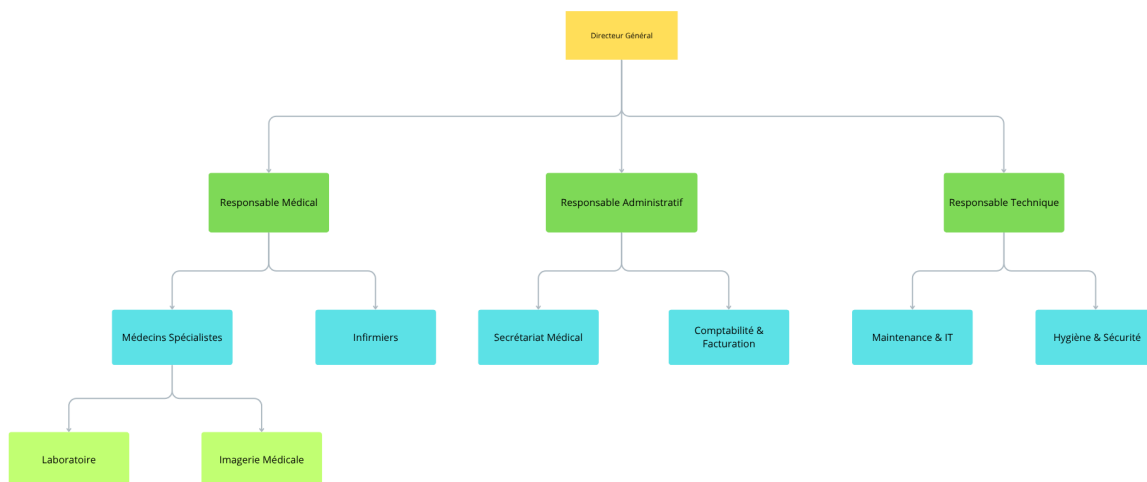


FIGURE 1.1 – Organigramme Clinique

Notre travail fait l'objet d'assurer une aide pour les deux premiers services : le service du Responsable Médical et coté Responsable Administratif. La section suivante est consacrée pour décrire la situation actuelle au niveau des systèmes de gestion des cliniques Médicales et des applications web disponibles sur le marché.

1.3 Étude de l'existant

1.3.1 Objectif

Actuellement, la clinique médicale gère ses rendez-vous et dossiers patients manuellement, via des carnets de rendez-vous et des archives papier. Cette approche entraîne des risques d'erreurs, une difficulté à consulter les historiques médicaux, et une vulnérabilité concernant la confidentialité des données sensibles, sans compter la lourdeur administrative pour le personnel soignant et les retards engendrés pour les patients.

L'objectif de ce projet est de remplacer ce système obsolète par une application web sécurisée et intuitive. Cette solution offrira une gestion centralisée des rendez-vous et dossiers médicaux, tout en garantissant un accès différencié et sécurisé pour chaque profil utilisateur (médecins, secrétaires, administrateurs). Elle permettra également la dématérialisation complète des documents de santé comme les ordonnances et certificats, en respectant les normes de protection des données médicales. Cette modernisation vise à améliorer l'efficacité des soins, la traçabilité des informations et la satisfaction des patients comme du personnel.

1.3.2 Création d'un système de gestion de clinique

a) Définition générale

Un système de gestion de clinique médicale (SGCM) est une plateforme informatique souvent une application web destinée à centraliser, organiser et automatiser l'ensemble des activités liées à la gestion médicale, administrative, financière et logistique d'une clinique. Il vise à faciliter le travail des différents acteurs de la clinique (médecins, secrétaires, personnel administratif, techniciens de laboratoire, etc.), à améliorer la qualité des soins, et à optimiser les ressources.

b) Objectifs principaux du système

Un bon SGCM doit permettre :

- La gestion efficace des dossiers médicaux des patients.
- L'amélioration de la coordination entre les différents services (consultation, laboratoire, imagerie, pharmacie...). La réduction des erreurs humaines (doublons, oublis, mauvaise interprétation).
- Un gain de temps et de productivité pour le personnel médical et administratif.
- La traçabilité et la confidentialité des données médicales.
- La génération automatisée de documents (ordonnances, factures, résultats...).

c) Modules fonctionnels à maîtriser avant de développer l'application

Voici les principaux modules que l'on retrouve dans un système de gestion de clinique médicale. Pour développer un tel système, il faut comprendre leurs fonctions et les interactions entre eux :

- Module d'authentification et de gestion des utilisateurs
- Module de gestion des patients
- Module de rendez-vous

- Module de consultation médicale
- Module de laboratoire et d'imagerie
- Module de facturation et comptabilité
- Module de pharmacie (interne)
- Module de rapports et statistiques [2]

1.3.3 Aspects techniques à prendre en compte

Pour développer une telle application, il faut aussi tenir compte des exigences techniques et métier :

- Sécurité des données médicales : cryptage, accès restreint, sauvegardes.
- Conformité réglementaire (ex. RGPD en Europe, lois locales de santé).
- Ergonomie de l'interface : facile à utiliser même par des utilisateurs non technophiles.
- Accès multi-utilisateur en réseau local ou via internet.
- Modularité et évolutivité : possibilité d'ajouter de nouveaux services à l'avenir.
- Hébergement sécurisé (cloud ou serveur interne selon contexte).

a) Compétences nécessaires pour concevoir un SGCM

- Analyse fonctionnelle du métier médical et des flux cliniques.
- Conception de bases de données relationnelles (patients, consultations, utilisateurs...).
- Développement web full-stack (front-end + back-end).
- Sécurité informatique et protection des données sensibles.
- Tests utilisateurs pour garantir que le système est adapté à la réalité du terrain.

1.4 Systèmes existants

Dans le même contexte que le notre, c'est à dire la réalisation d'un système de gestion de clinique médicale, nous avons trouvé trois travaux déjà réalisés et on les a comparé et on a mis les résultats dans le tableau suivant :

TABLE 1.1 – Comparaison des systèmes de gestion de cliniques médicales existants selon leur architecture, fonctionnalités et limitations.

Système	Fonctionnalités principales	Architecture	Base de données	Limites identifiées
Doctolib (France)[3]	<ul style="list-style-type: none"> - Prise de rendez-vous en ligne - Gestion dagenda partagé - Téléconsultation - Rappels automatiques - Dossier patient simplifié 	SaaS (Cloud) Architecture MVC Web 100% responsive	PostgreSQL (chiffrée)	<ul style="list-style-type: none"> - Ne couvre pas toute la gestion clinique - Faible personnalisation - Données sur serveur distant (hébergement propriétaire)
OpenMRS (Open Source)[4]	<ul style="list-style-type: none"> - Dossier patient complet - Modules personnalisables - Gestion multi-utilisateur - Historique médical 	Architecture modulaire Java (Spring) + JSP Front-end HTML	MySQL / MariaDB	<ul style="list-style-type: none"> - Interface ancienne - Installation et configuration complexes - Besoin de compétences techniques
Mediboard (France, Open Source) [1]	<ul style="list-style-type: none"> - Dossier médical informatisé - Bloc opératoire, urgences - Imagerie, pharmacie, RH - Statistiques médicales 	Architecture en couches PHP (Back-end) JavaScript (Front-end) MVC personnalisé	MySQL	<ul style="list-style-type: none"> - Installation complexe - Courbe d'apprentissage élevée - UI peu moderne

1.5 Besoins

1.5.1 Fonctionnels

a) Authentification et rôles

- L'application doit permettre la connexion des utilisateurs via une interface dédiée selon leur rôle :
 - Admin
 - Medecin
 - Secrétaire
- Chaque rôle doit accéder uniquement aux fonctionnalités qui lui sont propres.
- L'administrateur peut se connecter avec un champ login et un mot de passe.
- Authentification sécurisée via bcrypt.

b) Gestion des administrateurs

L'administrateur peut :

- Accéder à un tableau de bord (/admin/dashboard)
- Gérer les utilisateurs de type médecin et secrétaire
- Avoir une vue globale des activités de la clinique

c) Gestion des médecins

Les médecins peuvent :

- Se connecter à leur espace
- Gérer les rendez-vous
- Créer des fiches patients
- Planifier les consultations

d) Interface utilisateur

- Interface de sélection du rôle sur la page de login
- Affichage des messages d'erreur si les identifiants sont incorrects
- Messages de succès après déconnexion

e) Déconnexion

- Chaque utilisateur peut se déconnecter
- Après déconnexion, l'utilisateur est redirigé vers son formulaire de connexion

1.5.2 Non Fonctionnels

a) Sécurité

- Utilisation de Auth ::guard() pour isoler les rôles (authentification multi-provider)
- Mots de passe hachés avec bcrypt
- Middleware pour protéger les routes sensibles (ex. auth :admin)

b) Compatibilité

- Application compatible avec Laravel 5.4 (ancienne version)
- Syntaxes respectueuses de la version (pas de fonctionnalités modernes type Sanctum, Jetstream, etc.)

c) Performance

- Routes protégées uniquement accessibles par les utilisateurs authentifiés
- Utilisation de `redirect()->intended()` pour éviter les chargements inutiles

d) Extensibilité

- Architecture claire avec des dossiers séparés pour chaque rôle (views/medecin, views/secretaire, views/admin)
- Possibilité d'ajouter d'autres rôles ou modules (ex. pharmacien, laboratoire)

e) Ergonomie

- Interfaces utilisateurs simples (boutons clairs, messages d'erreur/succès)
- Bonne séparation des interfaces selon le rôle de l'utilisateur

1.6 Conclusion

Après avoir présenté le contexte de notre travail, nous avons réalisé le cahier de charges. On va pouvoir faire une étude détaillée de ce système.

Chapitre 2

Conception détaillée de notre système

Alliance Médicale

2.1 Introduction

La phase de conception constitue une étape essentielle dans la réalisation de notre projet. Elle vise à élaborer une structure organisationnelle et fonctionnelle capable de répondre efficacement aux besoins des utilisateurs, tout en respectant les contraintes spécifiques liées au domaine médical.

Dans ce chapitre, nous allons effectuer la conception de notre application web dédiée à la gestion d'une clinique médicale. Cette étape nous permettra de modéliser les différents aspects du système, tels que les flux d'informations, les interactions entre les utilisateurs et les modules fonctionnels de la plateforme.

Pour mener à bien cette modélisation, nous avons opté pour le langage UML (Unified Modeling Language), qui constitue un outil standardisé et puissant pour représenter graphiquement les éléments essentiels au développement de notre application.

2.2 Le langage de modélisation UML

UML (Unified Modeling Language, qui veut dire langage de modélisation unifié en français) est une notation permettant de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes existant auparavant, et est devenu désormais la référence en terme de modélisation objet.

La programmation orientée objet consiste à modéliser grâce à l'informatique un ensemble d'éléments d'une partie du monde réel (que l'on appelle domaine) en un ensemble d'entités informatiques. Ces entités sont appelées objets. Il s'agit de données informatiques regroupant les principales caractéristiques des éléments du monde réel. La difficulté de cette modélisation consiste à créer une représentation abstraite, sous forme d'objets, d'entités ayant une existence matérielle ou bien virtuelle.

La modélisation objet consiste à créer une représentation informatique des éléments du monde réel auxquels on s'intéresse, sans se préoccuper de l'implémentation (indépendant d'un langage de programmation). Il s'agit donc de déterminer les objets présents et d'isoler leurs données et les fonctions qui les utilisent. Pour cela des méthodes ont été mises au point. Entre 1970 et 1990, de nombreux analystes ont mis au point des approches orientées objets, si bien qu'en 1994 il existait plus de 50 méthodes objet. Toutefois seules 3 méthodes ont véritablement émergé :

- La méthode OMT de James Rumbaugh.
- La méthode BOOCH de Grady Booch.
- La méthode OOSE de Ivar Jacobson.

A partir de 1994, Rumbaugh et Booch (rejoints en 1995 par Jacobson) ont unis leurs efforts pour mettre au point la méthode unifiée (unified method 0.8), incorporant les avantages de chacune des méthodes précédentes.

En janvier 1997, La méthode unifiée 1.0 change de nom et devient la méthode UML, qui va s'imposer comme étant une notation universelle pour la modélisation objet.

Ainsi, il est nécessaire qu'une méthode objet soit définie de manière rigoureuse et unique afin de lever les ambiguïtés. De nombreuses méthodes objet ont été définies, mais aucune ne s'est imposée en raison du manque de standardisation. C'est pourquoi l'ensemble des acteurs du monde informatique a fondé en 1989 l'OMG, une organisation à but non lucratif, dont le but est de mettre au point des standards garantissant

la compatibilité entre des applications programmées à l'aide de langages objet et fonctionnant sur des réseaux hétérogènes. Enfin, en juillet 2005, la version 2.0 d'UML a été adopté par l'OMG. [5]

2.3 La méthode UP

Unified Process(que lon peut traduire par Processus Unifié) est un processus de développement logiciel conçu par les développeurs d'UML Jacobson, Booch, Rumbaugh en 1999, regroupe les activités à mener pour transformer les besoins dun utilisateur en système logiciel.

L'objectif d'un processus unifié est de maîtriser la complexité des projets informatiques en diminuant les risques. Il permet de répondre aux quatre préoccupations suivantes :

- QUI participe au projet ?
- QUOI,qu'est-ce qui est produit durant le projet ?
- COMMENT doit-il être réalisé ?
- QUAND est réalisé chaque livrable ?

Le processus de développement UP, associé à UML, met en uvre plusieurs principes, parmi eux l'utilisation des diagrammes de cas d'utilisation, de séquence et de classe que l'on va essayer de réaliser pour notre projet.[6]

2.4 Conception

Dans notre système, on aura 3 différents acteurs et chacun d'entre eux représente une entité externe qui interagit directement avec le système, ces acteurs sont :

- Le Médecin.
- Le secrétaire.
- L'Administrateur.

On a choisi d'utiliser 2 diagrammes principaux d'UML qui nous ont semblé être les plus adéquats pour notre étude, à savoir :

- Le diagramme de cas d'utilisation qui sert à modéliser les besoins des utilisateurs.
- Le diagramme de séquence système pour détailler tous les scénarios possible des acteurs avec le système.

Enfin, pour la création des différents diagrammes qui vont suivre, on a utilisé :

- Modelio qui est un outil open source qui prend en charge la norme UML2.[7]
- Draw.io qui est un logiciel de dessin graphique open source qui permet de créer des diagrammes UML.[8]

2.4.1 Diagrammes de Cas d'Utilisation

Le diagramme de cas d'utilisation est un diagramme UML utilisé pour donner une vision globale du comportement fonctionnel d'un système logiciel. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (Humain ou Machine) et un système.

Dans un diagramme de cas d'utilisation il existe des acteurs qui interagissent avec des cas d'utilisation UC.

Les UC permettent de structurer les besoins des utilisateurs et les objectifs du système.[9]

Nous allons présenter ci-dessous les différents diagrammes qu'on a réalisé.

a) Pour le Médecin

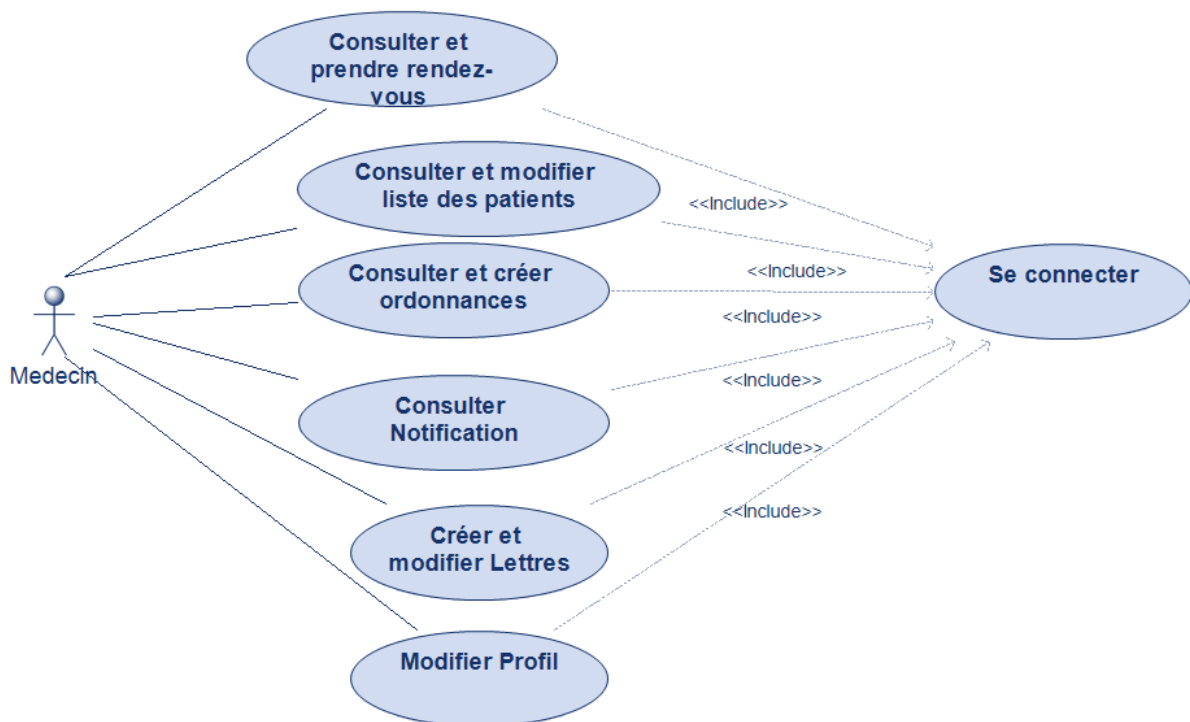


FIGURE 2.1 – Diagramme de Cas d'Utilisation Médecin

Le Medecin peut :

- Consulter et prendre un rendez-vous
- Consulter et modifier la liste des patients
- consulter et créer des ordonnances
- consulter les notifications
- Créer et modifier les lettres (lettre de bonne santé , lettre d'orientation...etc)
- Consulter et modifier son profil et ses informations
- Se connecter à son compte

Pour tous les acteurs qui vont suivre, ils auront besoin de se connecter à notre application web. Pour cela, nous avons mis un include vers le UC se connecter pour chacun de leurs cas d'utilisation.

b) Pour la Secrétaire

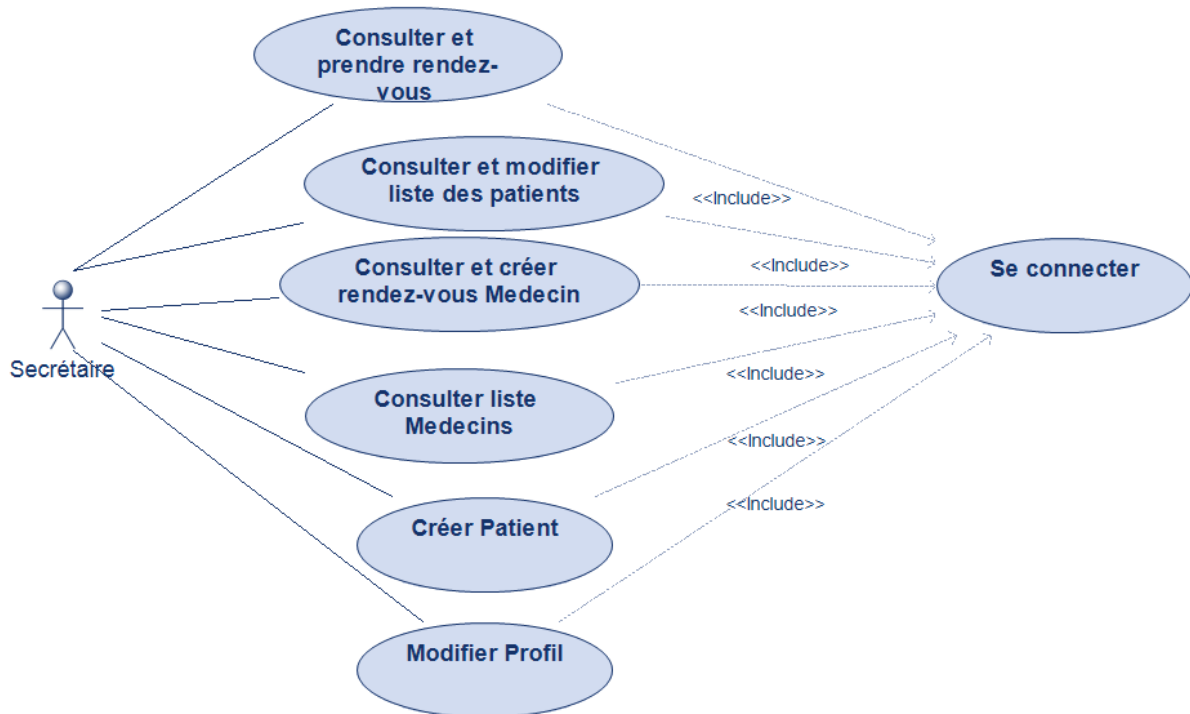


FIGURE 2.2 – Diagramme de Cas d'Utilisation Secrétaire

La secrétaire peut :

- Consulter et prendre rendez-vous
- Consulter et modifier la liste des patients
- Consulter et créer un rendez-vous pour un médecin
- Consulter la liste des médecins
- Créer un patient
- Se connecter à son compte
- Consulter et modifier son profil et ses informations

c) Pour l'Admin

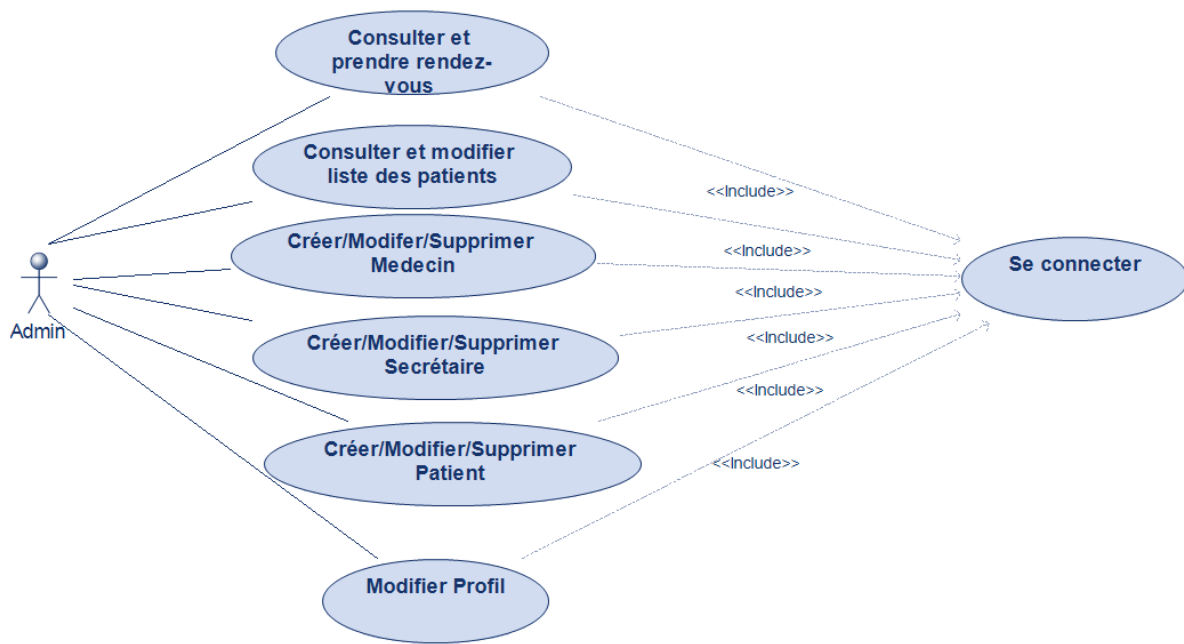


FIGURE 2.3 – Diagramme de Cas d'Utilisation Admin

L'Admin peut :

- Consulter et prendre un rendez-vous
- Consulter et modifier la liste des patients
- Créer et modifier et supprimer un médecin
- Créer et modifier et supprimer un patient
- Créer et modifier et supprimer un secrétaire
- Se connecter à son compte
- Consulter et modifier son profil et ses informations

2.4.2 Diagrammes de Séquence Système

Les diagrammes de séquence, modélisent les interactions entre les objets dans un cas d'utilisation unique. Ils illustrent la manière dont les différentes parties d'un système interagissent entre elles pour exécuter une fonction, et l'ordre dans lequel les interactions se produisent lorsqu'un cas d'utilisation particulier est exécuté.

En termes plus simples, un diagramme de séquence montre les différentes parties d'un système qui travaillent dans une séquence pour obtenir quelque chose.[10]

Nous allons présenter les différents diagrammes qu'on a réalisés dans les lignes qui suivent.

a) Pour le Médecin

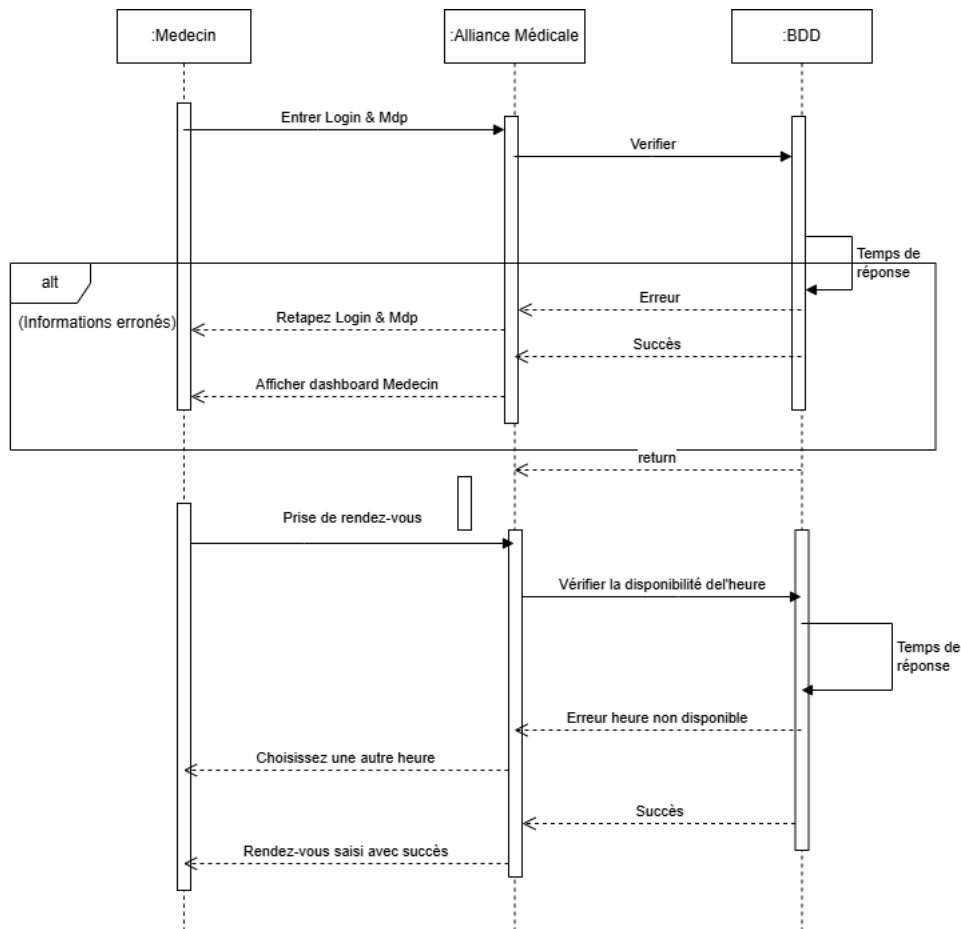


FIGURE 2.4 – Diagrammes de Séquence Système Médecin

b) Pour la Secrétaire

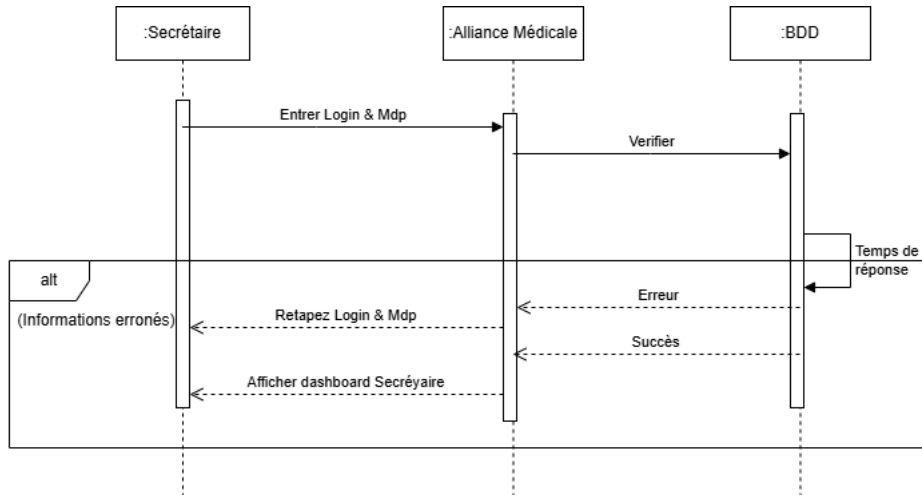


FIGURE 2.5 – Diagrammes de Séquence Système Secrétaire

c) Pour l'Admin

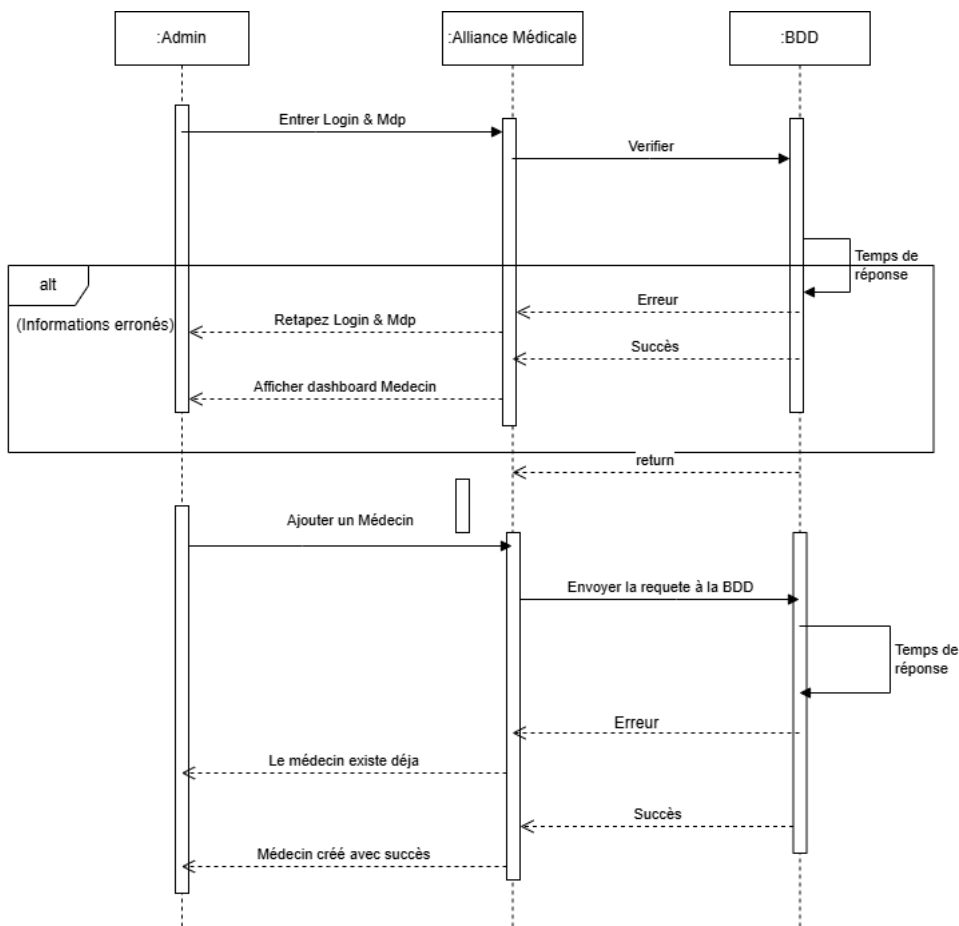


FIGURE 2.6 – Diagrammes de Séquence Système Admin

2.4.3 Diagrammes de Classes

Les diagrammes de classes sont des représentations statiques de la structure d'un système. Ils modélisent les différentes classes présentes dans une application, leurs attributs, leurs méthodes, ainsi que les

relations (héritage, association, agrégation, etc.) qui existent entre elles.

Autrement dit, un diagramme de classes décrit ce que le système contient, en mettant en avant les entités principales et les liens logiques qui les relient. Il sert de base à la conception orientée objet et permet de mieux comprendre l'architecture globale du système.[11]

Nous allons présenter notre diagramme de classes dans la section suivante :

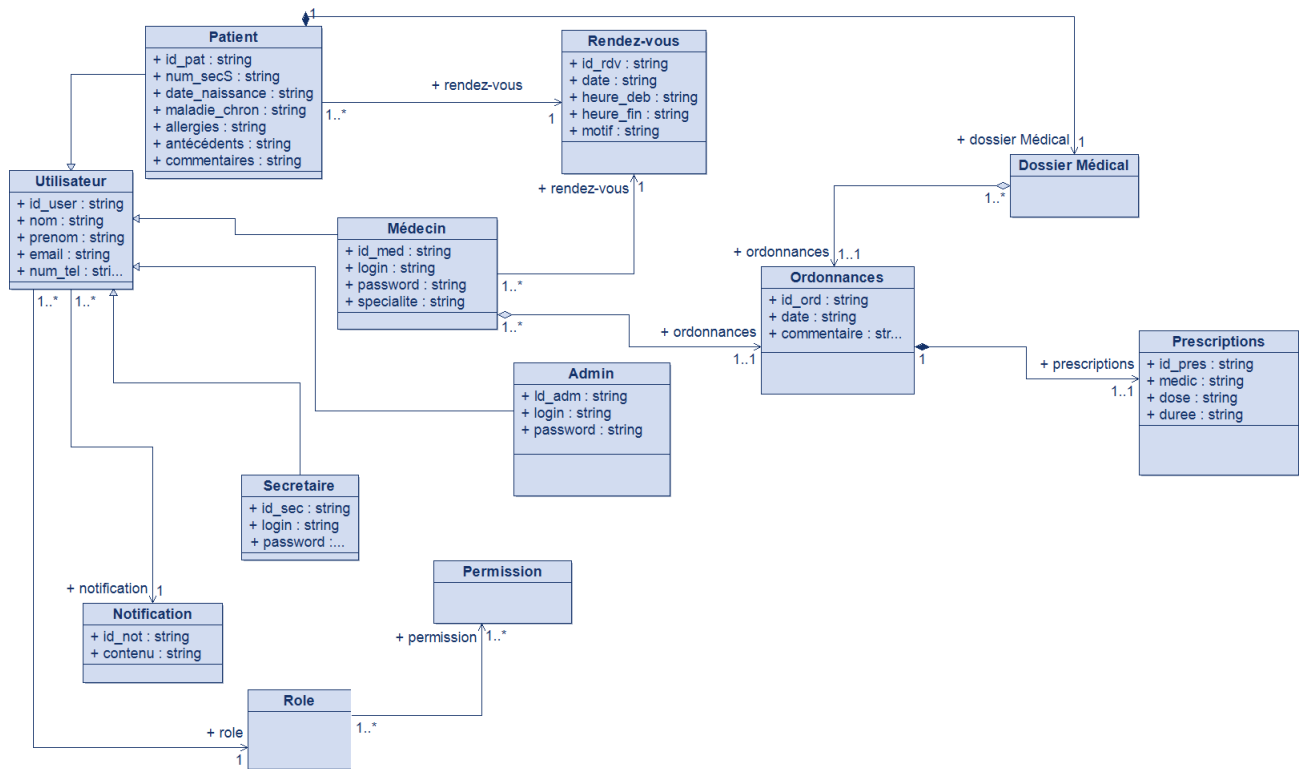


FIGURE 2.7 – Diagramme de Classes

2.4.4 Conclusion

Dans ce chapitre, nous avons réalisé la conception détaillée du système de notre application web en utilisant le langage de modélisation UML et particulièrement les 2 diagrammes principaux de la méthode UP : le diagramme de cas d'utilisation et de séquence système.

Dans le chapitre suivant, nous allons aborder la dernière partie de notre travail qui représente la réalisation de notre application web.

Chapitre 3

Mise en uvre de Alliance Médical

3.1 Introduction

La réalisation est la dernière étape de notre travail est très importante car c'est grâce à elle que notre application web va voir le jour. Sa réussite dépend principalement du choix des outils choisis pour sa mise en place, afin de garantir une réponse optimale aux différents besoins des utilisateurs. Ce chapitre est donc dédié à cette étape où l'on va d'abord présenter tous les logiciels, outils et langages de programmation qu'on a utilisé pour ensuite enchaîner avec la création des schémas de notre bdd ainsi que la manipulation de leurs données et on finira avec les interfaces de l'application web.

3.2 Présentation des logiciels et outils de travail



FIGURE 3.1 – Logo des outils utilisés

3.2.1 Nice Page

Nicepage est une application de conception web permettant de créer des interfaces modernes, responsives et personnalisables sans avoir besoin de coder. Elle est disponible sous forme de logiciel autonome pour Windows et MacOS, ainsi que sous forme de plugin pour des CMS populaires tels que WordPress et Joomla.

Nicepage propose une approche visuelle de la conception, avec un système de glisser-déposer et une grille fluide qui facilite la disposition des éléments. L'outil permet aux utilisateurs de concevoir des pages web attrayantes et adaptatives compatibles avec les standards HTML5 et CSS3. Il offre également une large bibliothèque de modèles professionnels, d'icônes, et de composants visuels réutilisables.

Nicepage se distingue notamment par sa capacité à générer un code propre et exportable, pouvant être intégré dans n'importe quel projet web. Il s'adresse aussi bien aux designers qu'aux développeurs front-end, et facilite la création rapide de prototypes ou de maquettes interactives.[12]

Nous avons utilisé Nicepage pour concevoir certaines interfaces utilisateurs de notre application web, réaliser des maquettes visuelles pour présenter les différentes sections du site, travailler sur l'ergonomie et la disposition des composants UI (boutons, menus, cartes).

3.2.2 GitHub

Git est un logiciel de gestion de version créé par Linus Torvalds, auteur du noyau Linux. Il permet denregistrer les différentes modifications effectuées sur un projet et de pouvoir retourner à une version précédente du projet.

Dans le langage des systèmes de gestion de version, la copie de lintégralité des fichiers dun projet et de leur version située sur le serveur central est appelé un dépôt. Git appelle également cela repository.

GitHub est un service en ligne qui permet dhéberger des dépôts ou repository Git. Cest le plus grand hébergeur de dépôts Git du monde.

Une grande partie des dépôts hébergés sur GitHub sont publics, ce qui signifie que nimporte qui peut télécharger le code de ces dépôts et contribuer à leur développement en proposant de nouvelles fonctionnalités.

Pour récapituler, Git est un logiciel de gestion de version tandis que GitHub est un service en ligne dhébergement de dépôts Git qui fait office de serveur central pour ces dépôts.[11]

3.2.3 Modelio

Modelio est un outil de modélisation open source, conçu pour assister les ingénieurs logiciels dans la conception et larchitecture des systèmes informatiques. Il prend en charge plusieurs langages de modélisation, notamment UML (Unified Modeling Language) et BPMN (Business Process Model and Notation), et permet également la génération de code à partir des diagrammes.[7]

Modelio offre une interface graphique complète qui permet de créer des diagrammes de classes, diagrammes de cas dutilisation, diagrammes de séquence, diagrammes dactivités, ainsi que des modèles de base de données. Grâce à son extensibilité, il peut être enrichi de modules supplémentaires pour la gestion des exigences, la documentation, ou encore lintégration avec des environnements de développement comme Java ou SQL.

Modelio est compatible avec Windows, Linux, et MacOS, et se distingue par son respect des standards OMG (Object Management Group) pour l’ingénierie logicielle. Nous avons utilisé Modelio dans notre projet pour :

- Modéliser les cas d’utilisation du système (diagrammes UML),
- Définir la structure des classes dans l’architecture logicielle,
- Créer des modèles de la base de données relationnelle,
- Documenter les interactions entre les acteurs et les fonctionnalités du système.

3.2.4 Visual Studio Code (VSC)

VSC est un éditeur de code source léger mais puissant s’exécutant sur votre bureau et est disponible sur les systèmes d’exploitation Windows, macOS et Linux. Il a été développé par Microsoft. Il est livré avec un support intégré pour JavaScript, TypeScript, CSS et HTML et dispose d’un riche écosystème d’extensions pour d’autres langages via l’interface extension présente dans l’éditeur.[10]

Cet éditeur a été utilisé pour toute la programmation de l’application web.

3.3 Présentation des technologies pour la programmation

3.3.1 Laravel pour le back-end web

Laravel est un framework PHP open-source basé sur le motif de conception MVC (Modèle-Vue-Contrôleur). Il offre une structure robuste pour développer des applications web complexes de manière simple, rapide et sécurisée. Laravel fournit des outils intégrés tels que les migrations, les contrôleurs, les middlewares, les routes, ainsi que l'authentification. Son système de template intégré appelé Blade permet de créer des vues dynamiques.[13]

Nous avons utilisé Laravel (version 5.4 avec PHP 7.2.2) comme moteur principal de notre application web, pour gérer la logique métier, les opérations CRUD sur les données médicales (patients, médecins, RDV), ainsi que l'interface d'administration.

3.3.2 Blade pour la gestion des vues côté client

Blade est le moteur de template intégré à Laravel. Il permet d'intégrer facilement du code PHP au sein du HTML, en utilisant une syntaxe simple (comme @extends, @section, @foreach, etc.). Blade permet aussi de découper les vues en composants réutilisables, ce qui facilite la maintenance de l'interface utilisateur.

Blade a été utilisé pour concevoir toutes les interfaces web du projet : tableau de bord, gestion des rendez-vous, visualisation des profils, et vues dédiées aux rôles admin, médecin et secrétaire.

3.3.3 MySQL pour la gestion de la base de données

MySQL est un système de gestion de base de données relationnelle (SGBDR) populaire et puissant. Il utilise le langage SQL pour la manipulation des données. Laravel fournit un ORM (Eloquent) qui facilite l'interaction avec les tables de la base de données sans écrire directement de requêtes SQL complexes.

Nous avons utilisé MySQL pour gérer toutes les entités du système : patients, médecins, secrétaires, rendez-vous, ordonnances, dossiers médicaux, lettres, etc.

3.3.4 Laragon pour l'environnement de développement

Laragon est un environnement de développement local pour Windows. Il inclut Apache, MySQL, PHP, Composer et d'autres outils utiles, le tout préconfiguré. Il est léger, rapide et très pratique pour les développeurs PHP.

Laragon a été utilisé tout au long du projet pour lancer localement le serveur Laravel, gérer la base de données MySQL et tester l'application sans déploiement externe.[14]

3.4 Présentation des outils de sécurisation de données

3.4.1 Le JavaScript Object Notation Web Token

Plus communément appelé JWT, est un jeton d'accès aux normes RFC 7519 qui permet d'échanger des données de manière sécurisée entre deux parties. Il contient toutes les informations importantes sur

une entité, ce qui rend la consultation d'une bdd superflue et la session n'a pas besoin d'être stockée sur le serveur.

Les JWT sont particulièrement appréciés pour les opérations d'identification. Les messages courts peuvent être chiffrés et fournissent alors des informations sûres sur l'identité de l'expéditeur et si celui-ci dispose des droits d'accès requis. Les utilisateurs eux-mêmes ne sont qu'indirectement en contact avec les tokens, par exemple lorsqu'ils essaient de s'authentifier avec un nom d'utilisateur et un mot de passe. La véritable communication se fait entre les différentes applications du côté serveur et client.

Le schéma suivant explique en détail le fonctionnement du JWT :

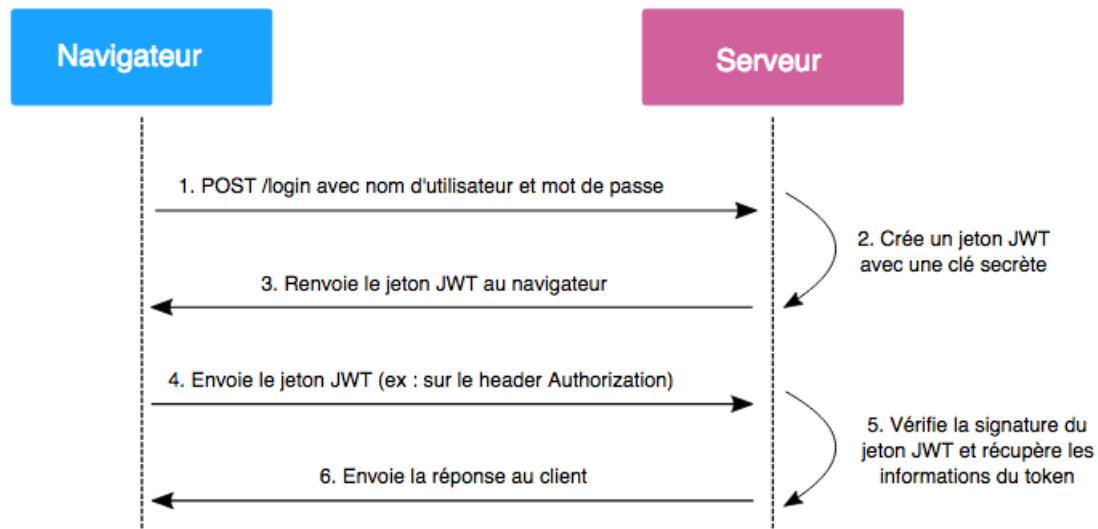


FIGURE 3.2 – Fonctionnement du JWT

En essayant de se connecter avec un nom d'utilisateur et un mot de passe, une clé secrète est déterminée avant utilisation du JWT. Dès qu'un utilisateur a entré avec succès ses données de connexion, le JWT est renvoyé avec la clé et stocké localement. Le transfert se fait par HTTPS afin de mieux protéger les données.

Lorsqu'un utilisateur veut accéder à des ressources protégées comme une API ou un chemin d'accès protégé, le JWT sera envoyé par l'agent utilisateur comme paramètre (par exemple `jwt` pour les GET-Requests) ou comme en-tête d'autorisation (pour POST, PUT, OPTIONS, DELETE). L'interlocuteur peut déchiffrer le JWT et si le contrôle est réussi, on va exécuter sa demande. [15]

3.4.2 Fonction de Hachage

Une fonction de hachage consiste à transformer une donnée quelconque en une donnée de taille fixée. Bien entendu, la donnée en question peut avoir plusieurs formes. Ce peut être du texte, une image, mais dans tous les cas la donnée sera transformée en un texte binaire avant qu'on lui applique la fonction de hachage qui va la rendre sous forme d'une suite de caractères.[16]

On l'a utilisé pour sécuriser les mots de passe des utilisateurs en les stockant hachées dans notre bdd à travers une fonction de hachage avec une clé secrète aléatoire pour que même un administrateur ne puisse pas le décoder.

3.4.3 Middleware d'authentification

Un middleware dans Laravel est un filtre entre la requête HTTP et l'application. Il peut intercepter une requête pour vérifier l'authentification, les permissions ou d'autres règles.

Nous avons mis en place plusieurs middlewares dans notre projet pour vérifier les droits d'accès selon les rôles (admin, médecin, secrétaire). Cela permet de protéger les routes sensibles et éviter les accès non autorisés.

3.5 Réalisation

3.5.1 Architecture du système Utilisée

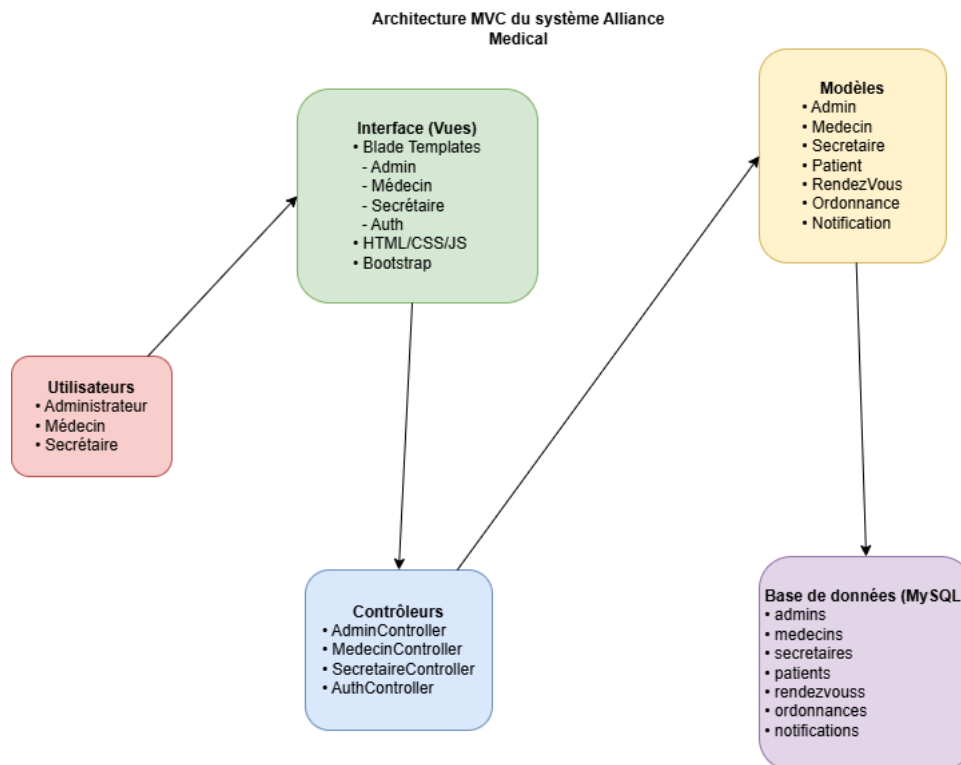


FIGURE 3.3 – Architecture du système

Le système Alliance Medical adopte une architecture trois tiers (MVC) avec une couche de présentation (Vues Blade/HTML), une couche de logique métier (Contrôleurs Laravel) et

une couche d'accès aux données (Modèles/MySQL), utilisant une base de données centralisée pour assurer la cohérence et l'intégrité des données médicales. Cette structure sécurisée par des guards Laravel et le hachage bcrypt permet une gestion efficace des profils (Admin, Médecin, Secrétaire) et de leurs interactions, tout en restant extensible pour une future application mobile via une API (Flutter/JWT).

3.5.2 Présentation des interfaces de notre application web Alliance Médical

a) Interface de Connexion

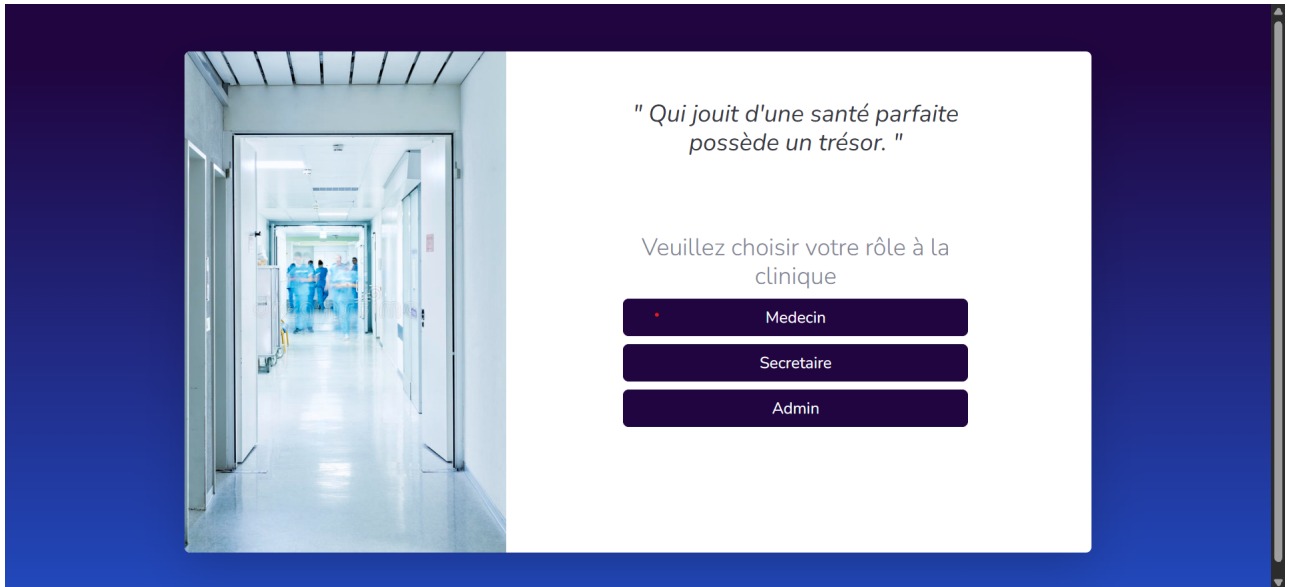


FIGURE 3.4 – Interface de Pré-login

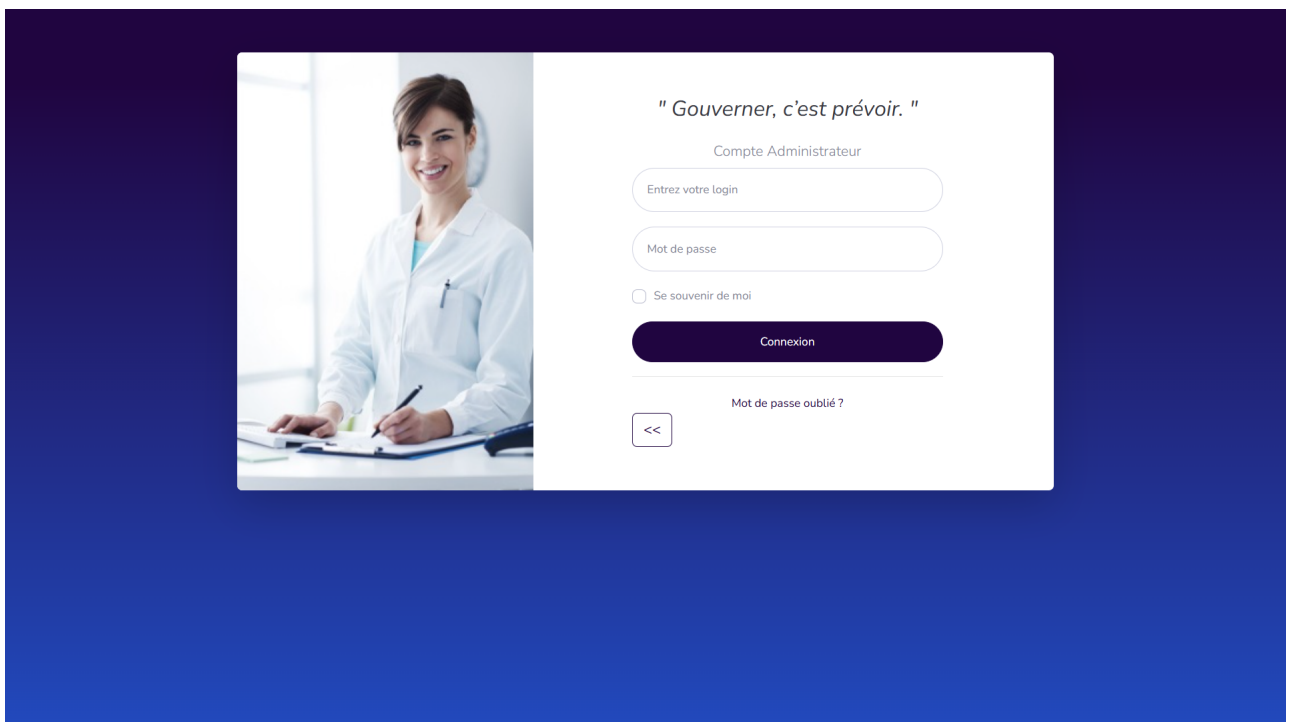


FIGURE 3.5 – Interface de Connexion

Dans notre interface de pré-login, l'utilisateur doit choisir son rôle pour pouvoir s'identifier et accéder à l'interface de login correspondante. Ces deux interfaces (pré-login et login) sont communes aux trois types d'acteurs du système.

b) Interfaces de la Secrétaire

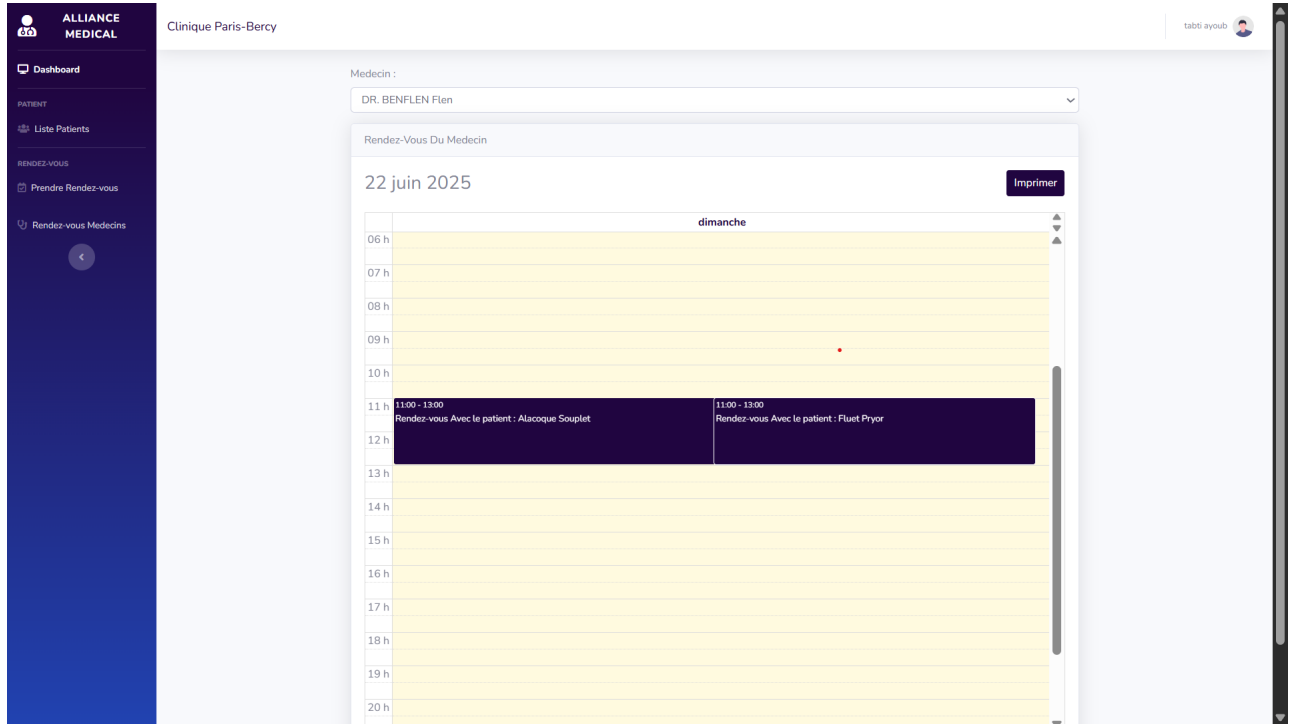


FIGURE 3.6 – Interface des Rendez-vous de la journée

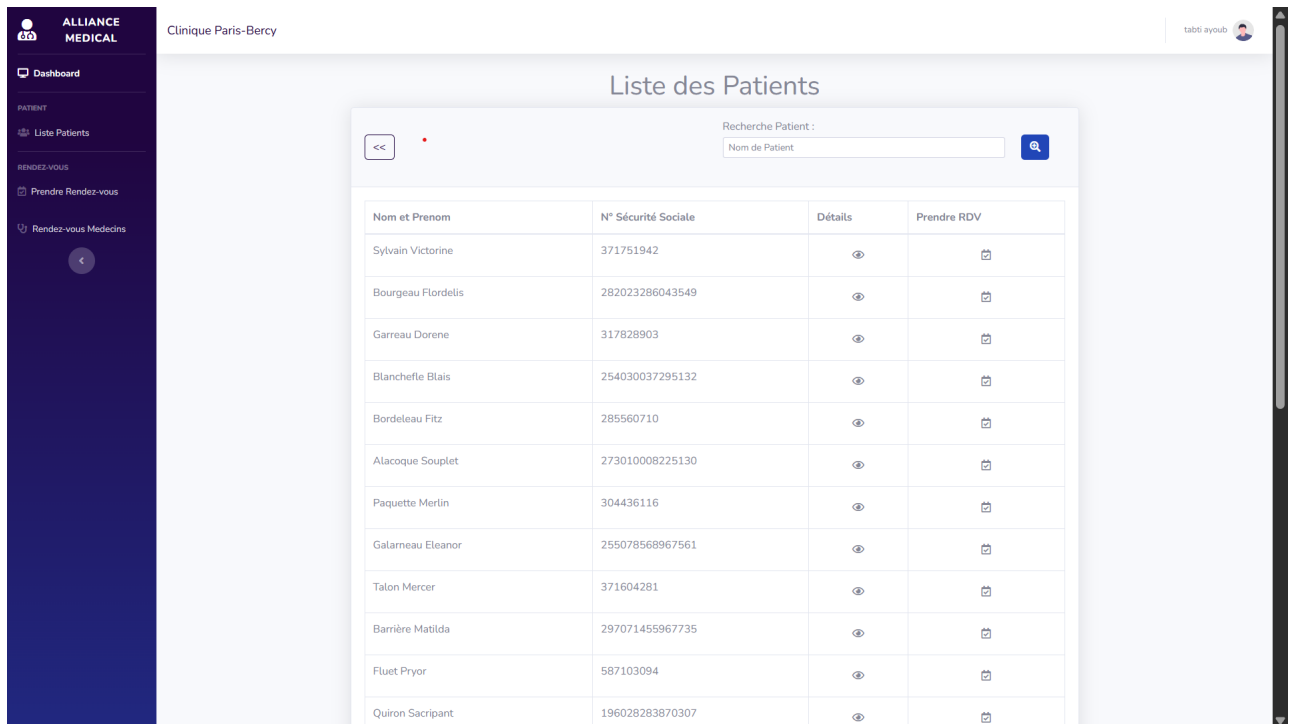


FIGURE 3.7 – Interface de la liste des Patients

The screenshot shows the 'Details Patient Avec Divers Rendez-vous' page. The left sidebar contains navigation options: Dashboard, PATIENT (Liste Patients), RENDEZ-VOUS (Prendre Rendez-vous, Rendez-vous Medecins). The main content area includes a form for patient details:

- Nom et Prenom : Sylvain, Victorine
- Date Naissance : 11/03/1973
- Sécurité Sociale : 371751942
- N° Téléphone : 0519566793
- Adresse E-Mail : VictorineSylvain@armyspy.com

Below the form is a 'Mettre à Jour' button and a section titled 'Liste des Rendez-vous du Patient :'. This section includes a 'Prendre un Nouveaux Rendez-vous ?' button and a calendar for 'juin 2025' with navigation options for 'Mois', 'Semaine', and 'Jour'.

lun.	mar.	mer.	jeu.	ven.	sam.	dim.
26	27	28	29	30	31	1
2	3	4	5	6	7	8

FIGURE 3.8 – Interface pour voir et modifier un patient

The screenshot shows the 'Prendre Rendez-vous' page. The left sidebar is the same as in Figure 3.8. The main content area includes a form for creating a new appointment:

- Nom & Prénom : Nom, Prénom
- Date de Naissance : jj/mm/aaaa
- N° Sécurité Sociale : ex: 99 333 000 22
- N° Téléphone : ex: 0576009999
- Adresse Email : ex: exemple@mail.com
- Date du Rendez-vous : jj/mm/aaaa
- Heure Debut et Fin : --:-- --:--
- Nom du Medecin : Choisissez une Medecin
- Motif : Ecrivez votre motif ici !

At the bottom of the form are 'Ajouter' and 'Vider' buttons. A copyright notice 'Copyright © Alliance Medical 2021' is visible at the bottom of the page.

FIGURE 3.9 – Interface pour prendre un rendez-vous

Après s'être connectée en tant que secrétaire, l'utilisateur peut consulter les rendez-vous prévus dans la journée, accéder à la liste complète des patients et effectuer des recherches spécifiques sur ces derniers. Il dispose également de la possibilité de visualiser les détails complets de chaque patient, de créer de nouvelles fiches patients tout en programmant leurs premiers rendez-vous, ou de reprogrammer les nouveaux rendez-vous pour les patients déjà enregistrés dans le système.

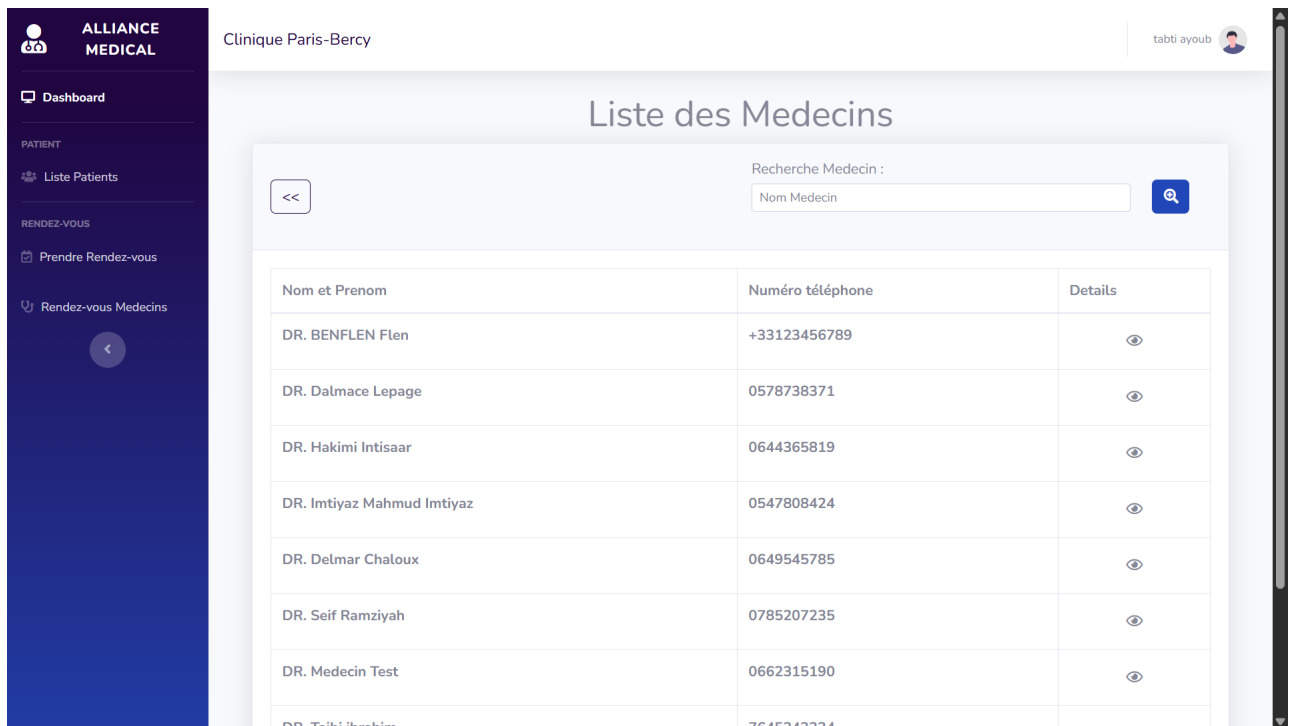


FIGURE 3.10 – Interface de la liste des Médecins



FIGURE 3.11 – Interface pour voir les détails d'un Médecin

Dans ces deux interfaces, la secrétaire peut consulter la liste complète des médecins de la clinique ainsi que leurs détails personnels, comprenant leurs noms, prénoms, spécialités et autres informations professionnelles pertinentes.

The screenshot shows the 'Modifier votre Mot de passe' (Change your password) interface. On the left is a dark blue sidebar with the 'ALLIANCE MEDICAL' logo and navigation options: 'Dashboard', 'PATIENT' (with 'Liste Patients'), and 'RENDEZ-VOUS' (with 'Prendre Rendez-vous' and 'Rendez-vous Medecins'). The main content area is light blue and contains the title 'Modifier votre Mot de passe'. Below the title are three input fields: 'Ancien mot de passe :' (placeholder: 'veuillez saisir l'ancien mot de passe'), 'Nouveau mot de passe :' (placeholder: 'veuillez saisir le nouveau mot de passe'), and 'Confirmation du nouveau mot de passe :' (placeholder: 'veuillez confirmer le nouveau mot de passe'). A green 'Modifier' button is positioned below the fields. The top right corner shows the user's name 'tabti ayoub' and a profile icon. The footer contains the text 'Copyright © Alliance Medical 2021'.

FIGURE 3.12 – Interface de mise à jour du mot de passe

This screenshot is identical to Figure 3.12, showing the 'Modifier votre Mot de passe' interface. However, a small red dot is visible in the 'Nouveau mot de passe' input field, indicating a validation error. The rest of the interface, including the sidebar, navigation, and footer, remains the same.

FIGURE 3.13 – Interface de mise à jour du profil

À travers ces deux interfaces, l'ensemble des utilisateurs - qu'il s'agisse de l'administrateur, du secrétaire ou du médecin - dispose de la possibilité de mettre à jour ses informations personnelles ainsi que de modifier son mot de passe pour garantir la sécurité de son compte.

c) Interfaces du Médecin

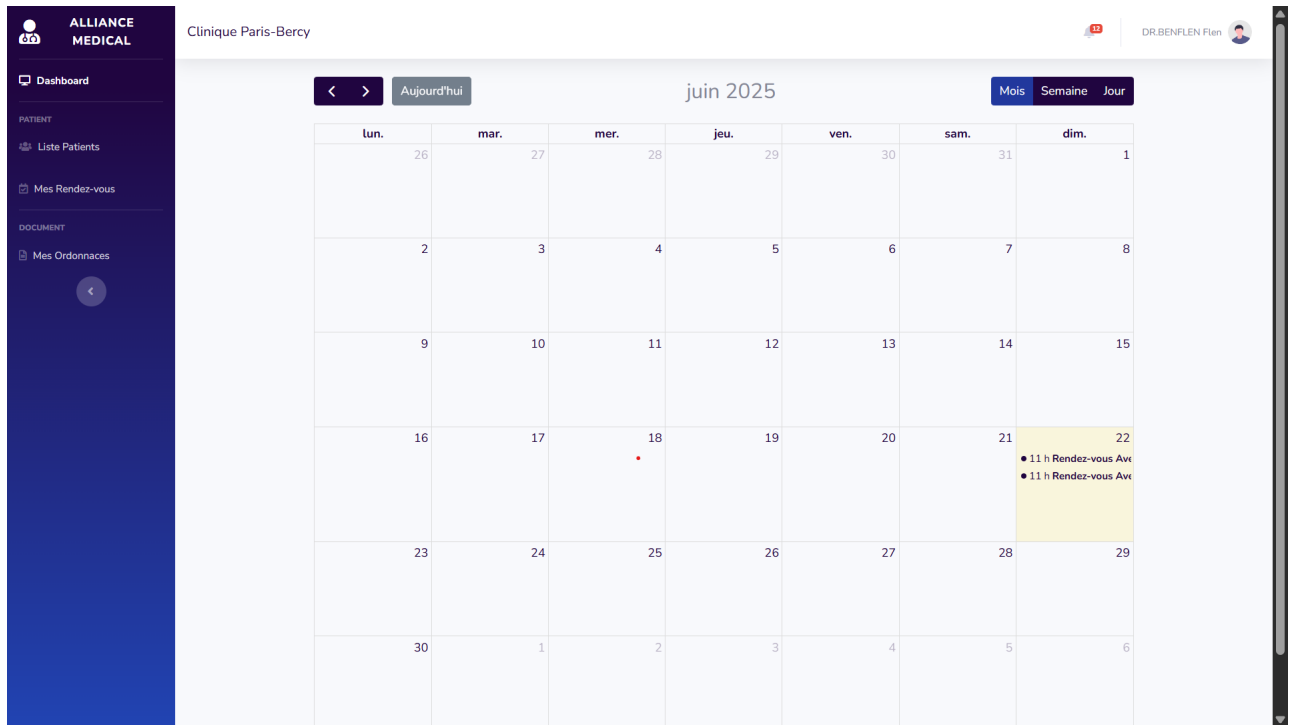


FIGURE 3.14 – Interface des Rendez-vous du médecin

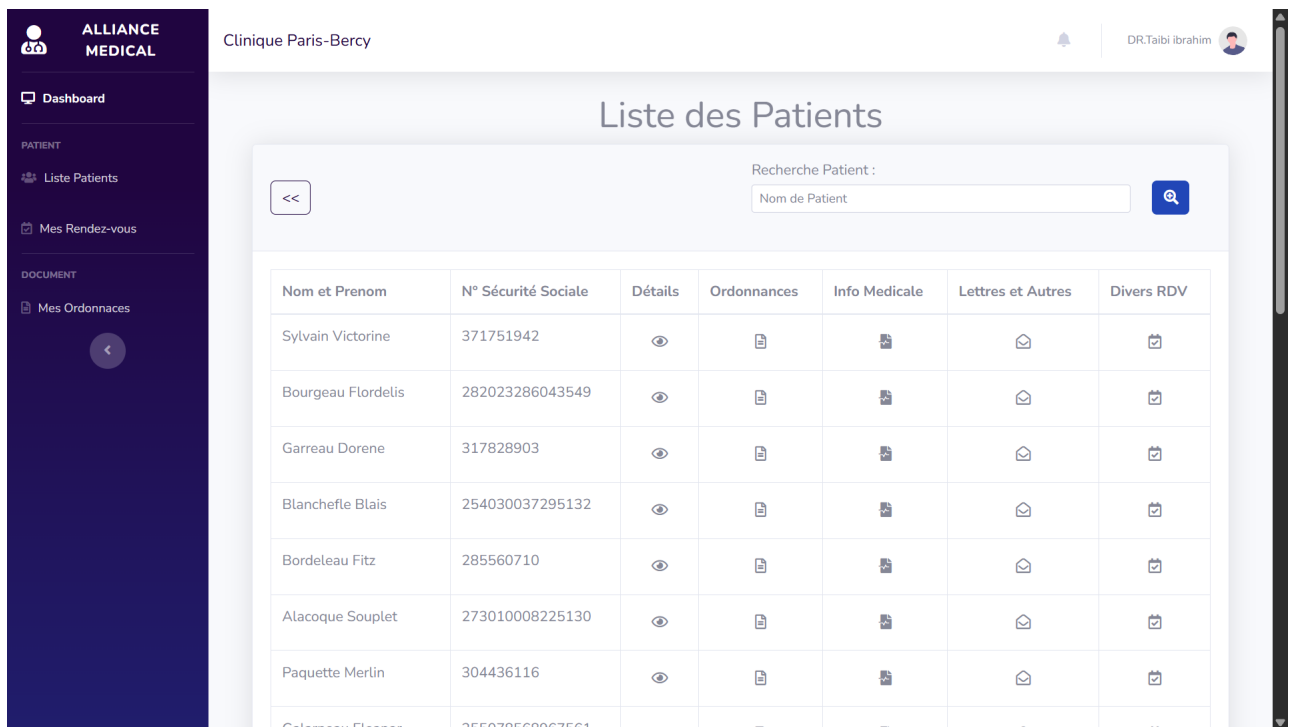


FIGURE 3.15 – Interface de la Liste des Patient

The screenshot shows the 'Détails du Patient' interface for 'Clinique Paris-Bercy'. The user is 'DR.BENFLEN Flen'. The interface has a dark blue sidebar with 'ALLIANCE MEDICAL' and navigation options like 'Dashboard', 'Liste Patients', 'Mes Rendez-vous', and 'Mes Ordonnances'. The main content area has a title 'Détails du Patient' and a navigation bar with tabs: 'Informations Personnelle', 'Information Medicale', 'Ordonnances', 'Lettres et autres', and 'Prendre RDV'. The 'Informations Personnelle' tab is active, showing a form with the following fields: 'Nom et Prenom' (Sylvain, Victorine), 'Date Naissance' (11/03/1973), 'N° Sécurité Social' (371751942), 'Email' (VictorineSylvain@armyspy.com), and 'Téléphone' (0519566793). A green 'Modifier' button is at the bottom.

FIGURE 3.16 – Interface des détails d'un Patient

The screenshot shows the 'Détails du Patient' interface for 'Clinique Paris-Bercy'. The user is 'DR.BENFLEN Flen'. The interface has a dark blue sidebar with 'ALLIANCE MEDICAL' and navigation options like 'Dashboard', 'Liste Patients', 'Mes Rendez-vous', and 'Mes Ordonnances'. The main content area has a title 'Détails du Patient' and a navigation bar with tabs: 'Informations Personnelle', 'Information Medicale', 'Ordonnances', 'Lettres et autres', and 'Prendre RDV'. The 'Information Medicale' tab is active, showing a form with the following fields: 'Maladies chroniques' (diabétique), 'Allergies' (allergie pollen, allergie poussiere.), 'Antecedents' (aucun), and 'Commentaires' (bonne santé). A green 'Mettre à Jour' button is at the bottom. At the bottom of the page, there is a copyright notice: 'Copyright © Alliance Medical 2025'.

FIGURE 3.17 – Interface du dossier médical d'un patient

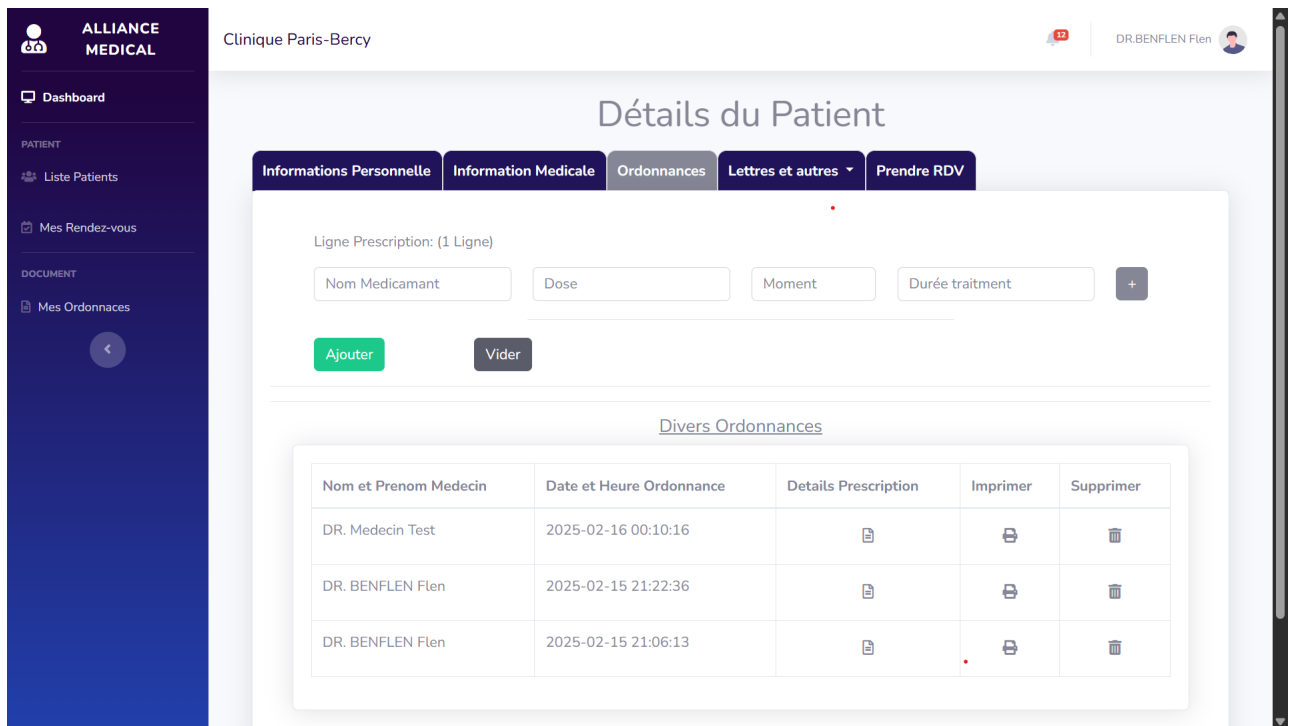


FIGURE 3.18 – Interface des Ordonnances d’un Patient

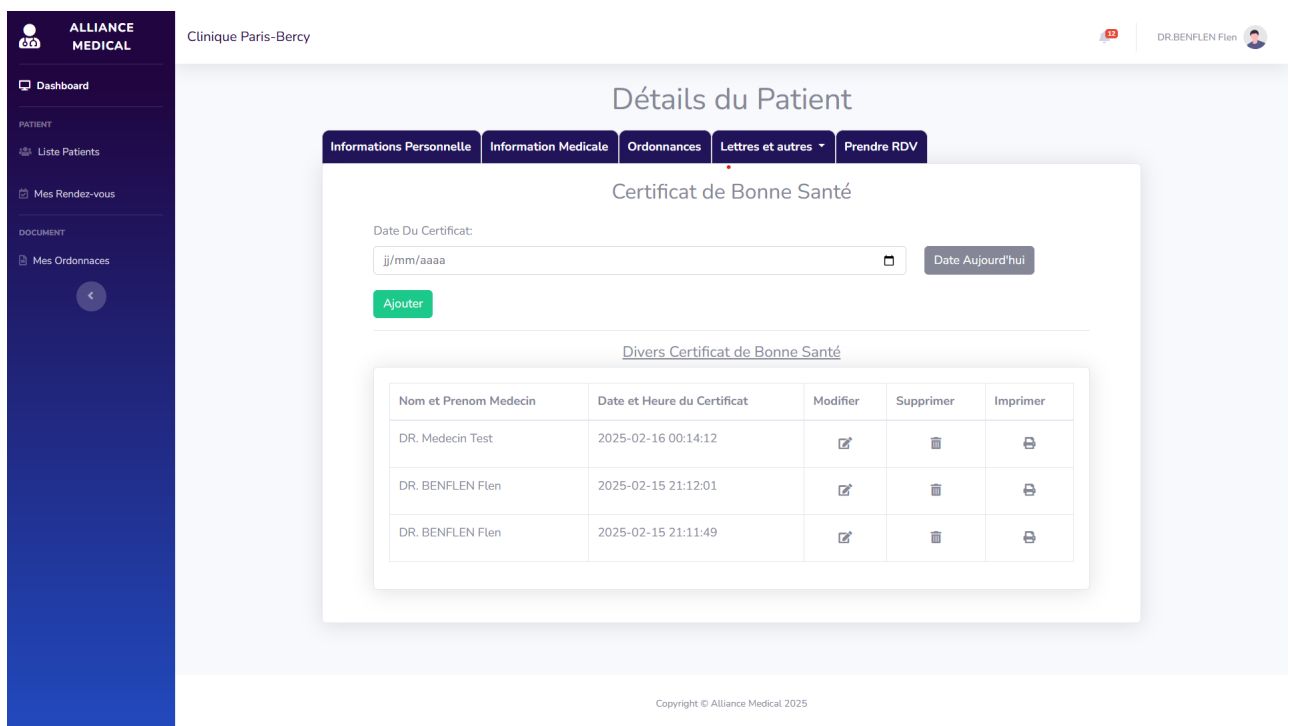


FIGURE 3.19 – Interface des Lettres pour un Patient

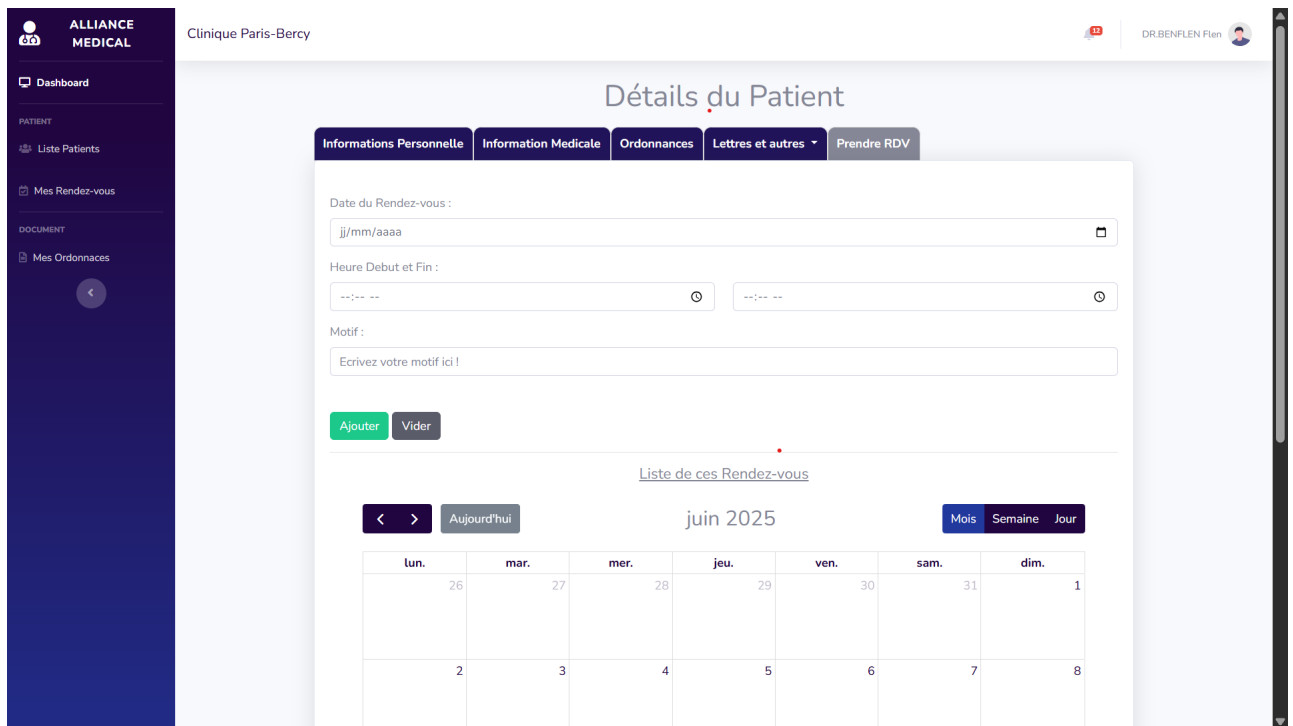


FIGURE 3.20 – Interface pour reprendre un rendez-vous pour un Patient

Dans ces interfaces, le médecin peut consulter l'ensemble de ses rendez-vous et recevoir des notifications en temps réel lorsqu'un nouveau rendez-vous est pris par le secrétaire. Il a accès à la liste complète de ses patients avec leurs dossiers médicaux détaillés - dont il est le seul habilité à modifier le contenu. Il peut également élaborer divers documents médicaux (certificats de bonne santé, arrêts de travail, etc.) et, le cas échéant, reprogrammer un rendez-vous de suivi ou une nouvelle consultation pour ses patients.

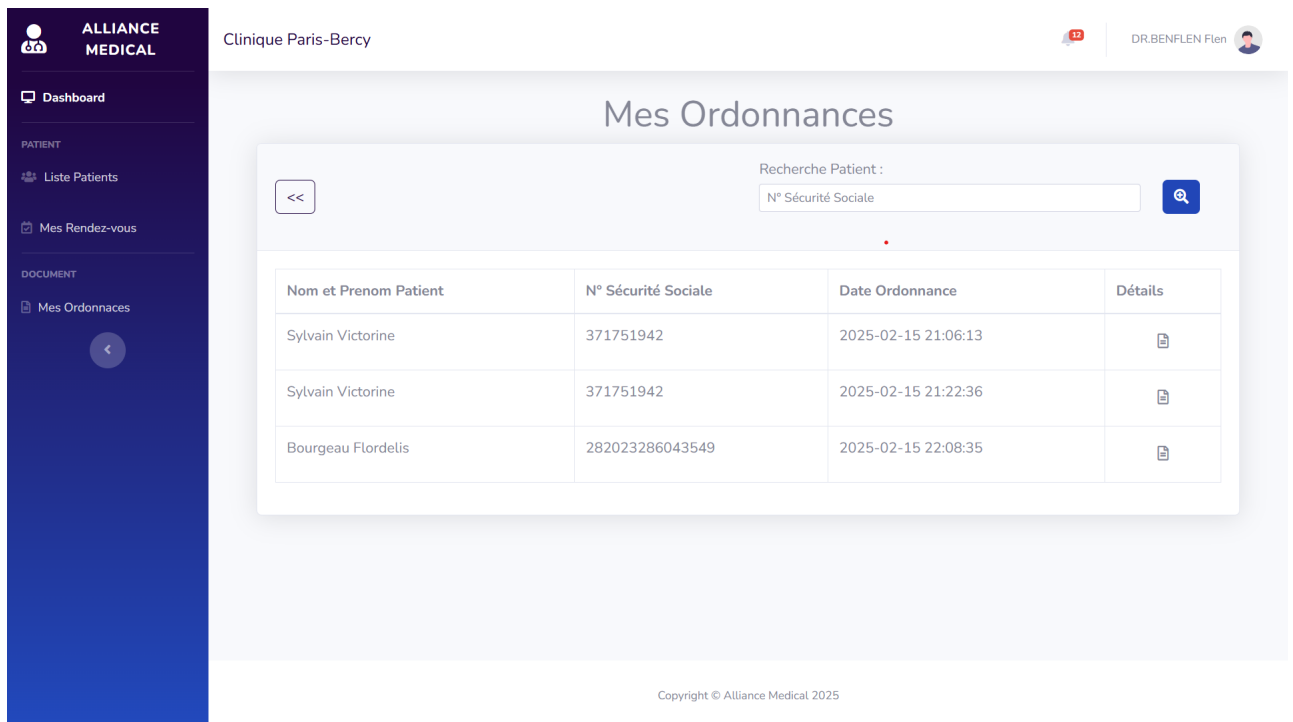


FIGURE 3.21 – Interface des Ordonnances du Médecin

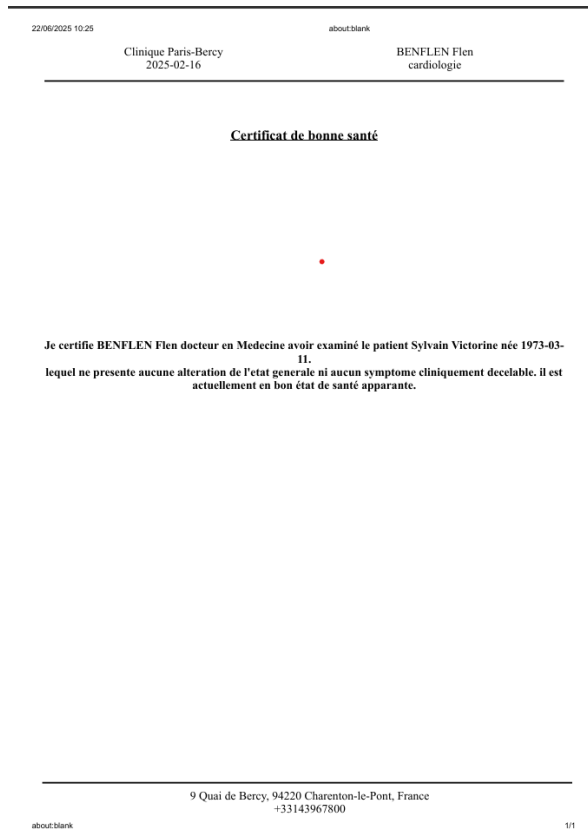


FIGURE 3.22 – Un Certificat de Bonne Santé

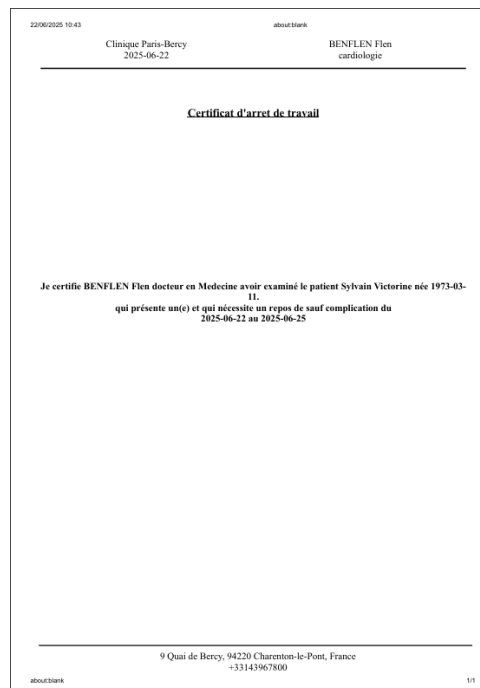


FIGURE 3.23 – Un Certificat d'Arrêt de travail



FIGURE 3.24 – Un Certificat de Reprise de Travail

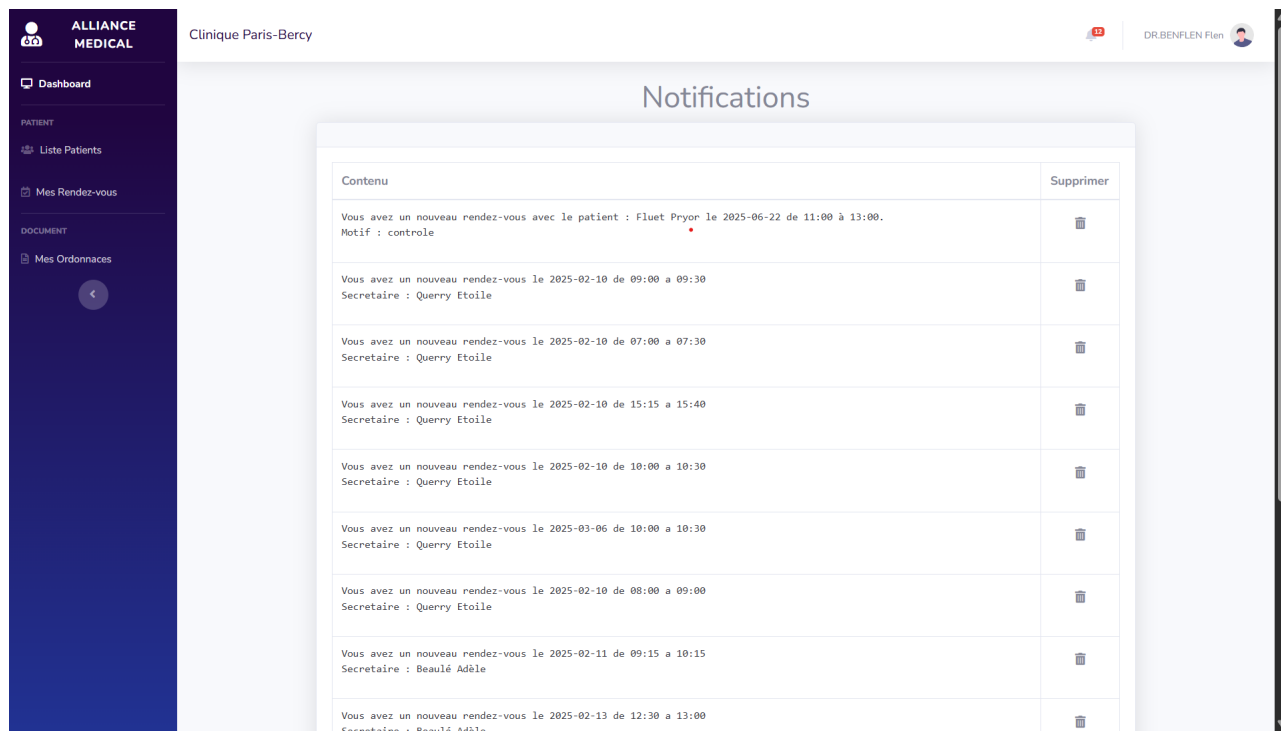


FIGURE 3.25 – Interface des Notifications pour le Médecin

Ces interfaces présentent l'historique complet des ordonnances rédigées par le médecin, ainsi que les modèles de documents qu'il peut produire, incluant les certificats de bonne santé et les certificats d'arrêt/reprise de travail, tout en offrant un accès centralisé à ses notifications professionnelles.

d) Interfaces d'Admin

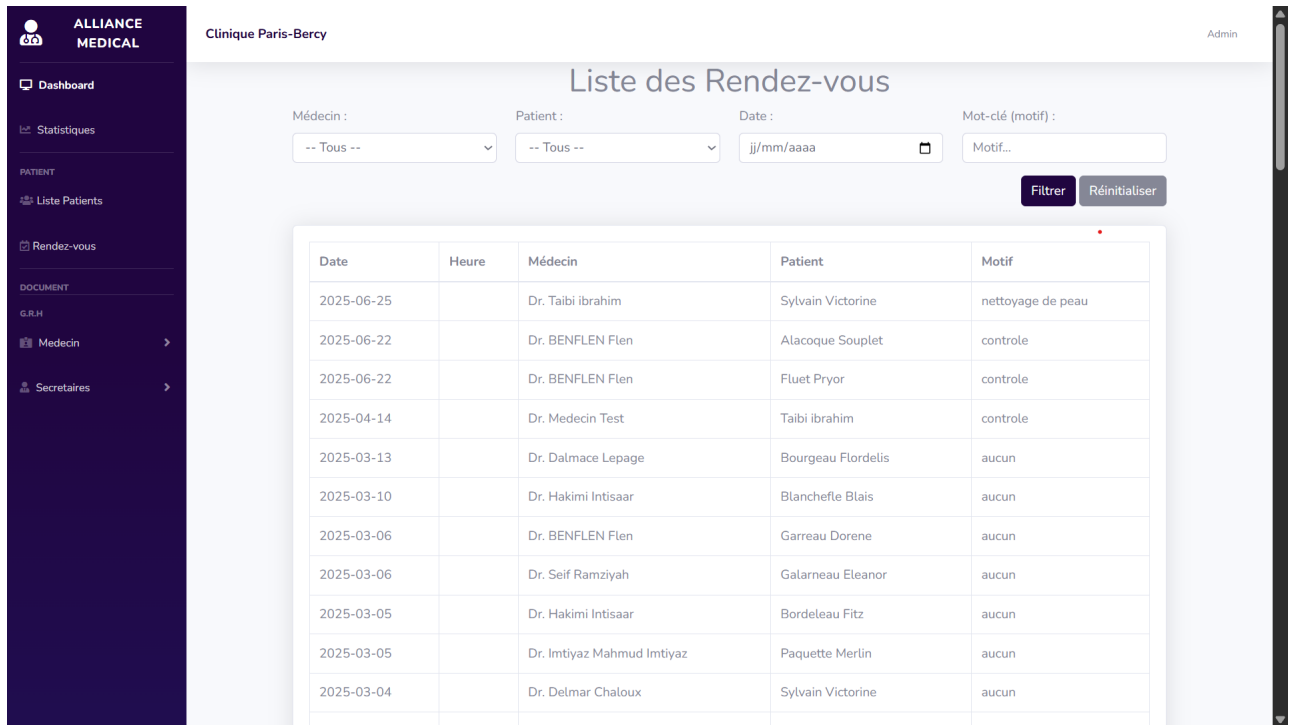


FIGURE 3.26 – Interface de la liste des rendez-vous de toute la clinique

L'administrateur dispose d'un accès complet à l'ensemble des rendez-vous de la clinique. Il peut les consulter et les filtrer selon plusieurs critères : par patient, par médecin, par date ou même par motif de consultation.

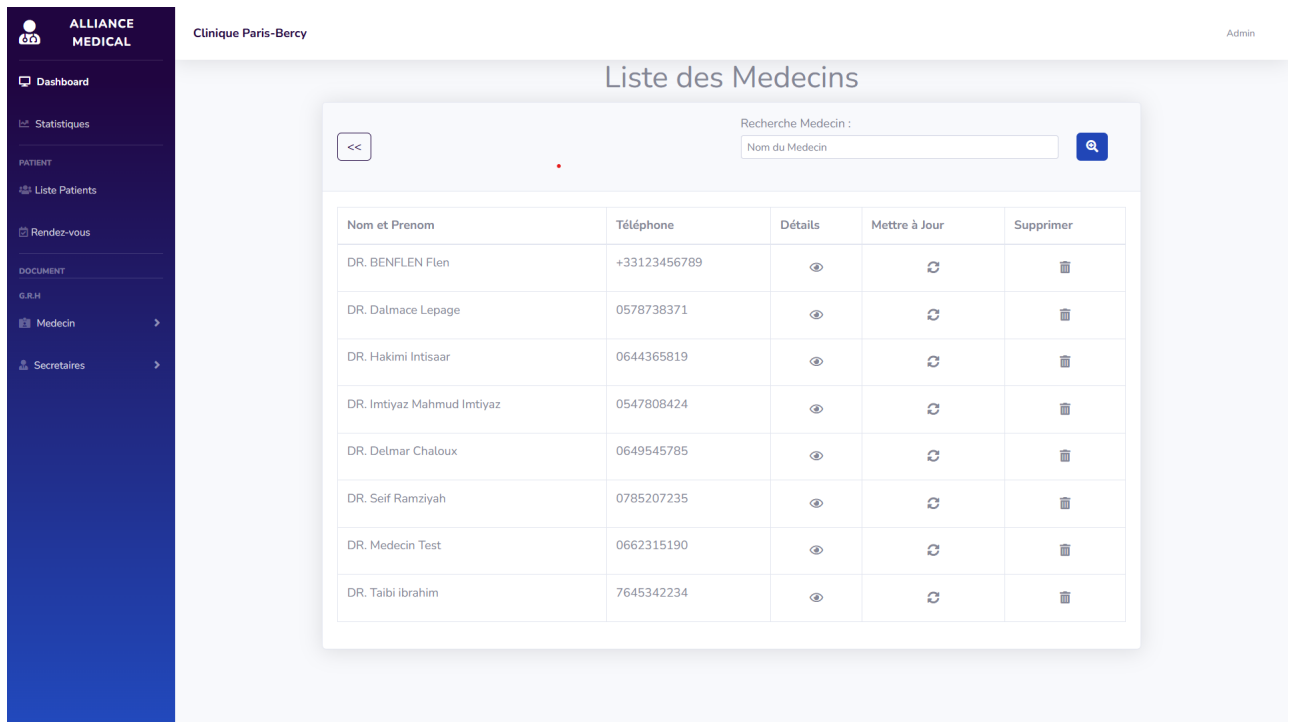


FIGURE 3.27 – Interface de liste des Médecins

ALLIANCE MEDICAL

Clinique Paris-Bercy

Admin

Mettre à Jour Médecin

Nom et Prénom :

BENFLEN Flen

Login :

flen.benflen

Email :

flen.benflen@alliancemedical.test

Téléphone :

+33123456789

Mettre à Jour Vider

Revoir le détail du Médecin ?

FIGURE 3.28 – Interface de mise à jour des Médecins

ALLIANCE MEDICAL

Clinique Paris-Bercy

Admin

Ajouter Medecin

Nom et Prenom :

Nom Prenom

Login :

Login

Password :

Mot de Passe Confirmation du Mot de Passe

Email :

Votre Adresse Email

Téléphone :

Votre Numéro de Téléphone

Spécialité :

choisissez une specialité

Ajouter Vider

FIGURE 3.29 – Interface de création des Médecins

L'administrateur peut accéder à la liste complète des médecins de la clinique, consulter leurs informations détaillées, les modifier, ou encore ajouter un nouveau praticien au système. Il dispose également de la possibilité de supprimer un médecin si nécessaire, et de même pour les secrétaires dans les interfaces qui suivent.

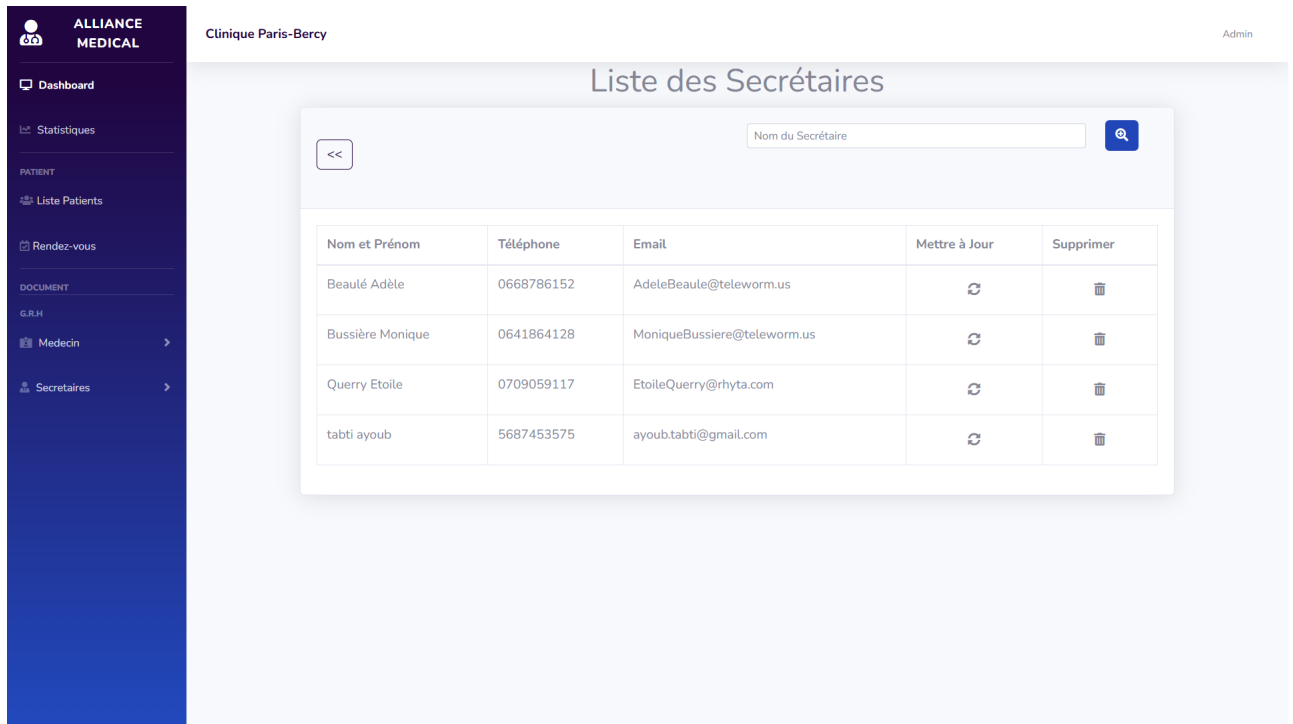


FIGURE 3.30 – Interface de liste des Secrétaires

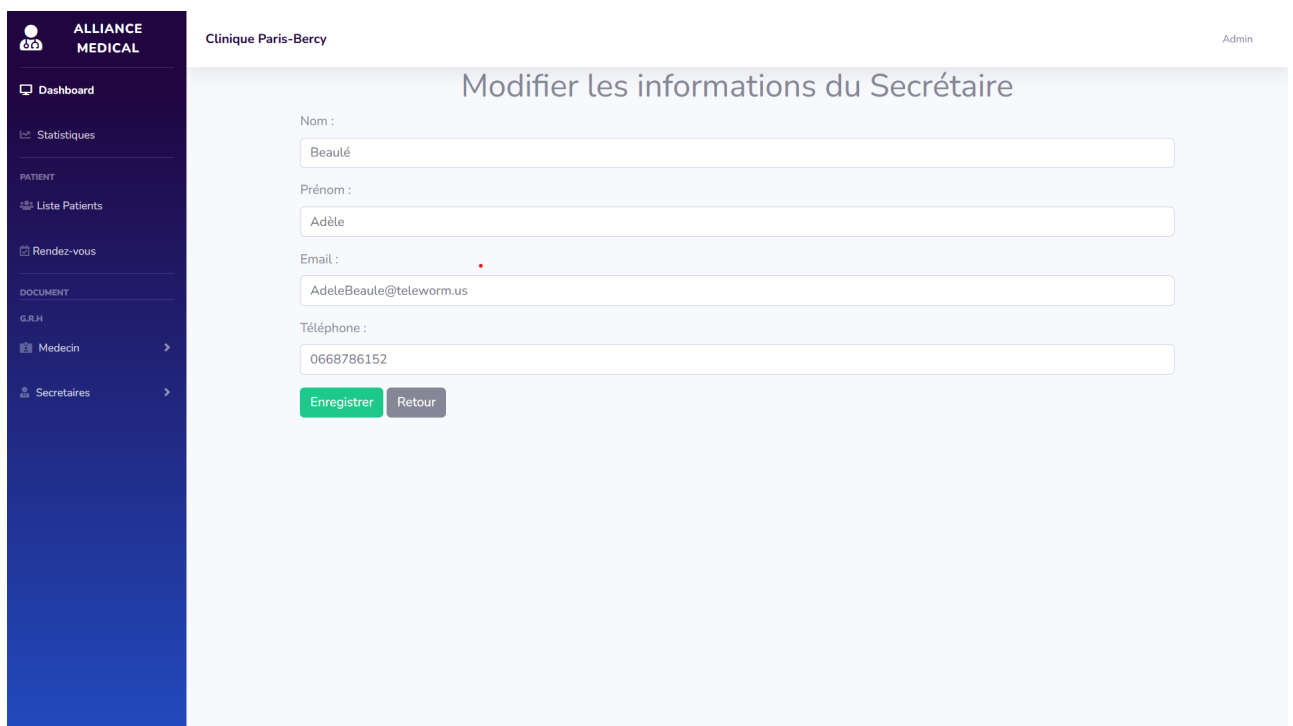


FIGURE 3.31 – Interface de mise à jour des Secrétaires

The screenshot shows the 'Ajouter un Secrétaire' (Add Secretary) interface. On the left is a dark blue sidebar menu with the following items: Dashboard, Statistiques, PATIENT (Liste Patients, Rendez-vous), DOCUMENT, G.R.H. (Medecin, Secretaires). The main content area is light blue and contains the following form fields:

- Nom et Prénom :** Two input fields for 'Nom' and 'Prénom'.
- Login :** One input field for 'Login'.
- Mot de passe :** Two input fields for 'Mot de Passe' and 'Confirmation du Mot de Passe'.
- Email :** One input field for 'Adresse Email'.
- Téléphone :** One input field for 'Numéro de Téléphone'.

At the bottom of the form are two buttons: a green 'Ajouter' button and a dark grey 'Vider' button.

FIGURE 3.32 – Interface de création des Secrétaires

3.5.3 Conclusion

Dans ce chapitre, nous avons détaillé les différents outils techniques et langages utilisés pour le développement de notre application Alliance Médical. Nous avons présenté l'architecture choisie, les modèles de données relationnels, ainsi que la structure des différentes interfaces web et mobiles. Chaque fonctionnalité majeure, comme la gestion des patients, des médecins, des secrétaires ou des rendez-vous, a été expliquée à travers son implémentation. Cette étape de réalisation représente le cur technique de notre travail. Dans la suite de ce mémoire, nous proposerons une conclusion générale récapitulant l'ensemble des étapes de notre projet.

Conclusion Générale

Dans le cadre de ce mémoire, nous avons présenté les différentes étapes qui ont conduit à la mise en place de notre application Alliance Médical, un système conçu pour faciliter la gestion des activités médicales au sein d'un établissement de santé. L'objectif principal était de centraliser et simplifier la gestion des patients, des rendez-vous, des médecins et des secrétaires à travers une solution numérique complète.

Le travail a débuté par une étude de l'environnement existant, ce qui nous a permis d'identifier les lacunes des méthodes traditionnelles de gestion. À partir de cette analyse, nous avons établi un cahier des charges précis, incluant les besoins fonctionnels et techniques attendus par les utilisateurs finaux.

Pour modéliser notre solution, nous avons utilisé UML via l'outil Modelio, ce qui nous a permis de structurer de manière claire les différentes entités et leurs interactions. Sur le plan technique, nous avons opté pour le framework Laravel 5.4, qui nous a offert un cadre solide pour développer la partie web de l'application. La base de données relationnelle a été conçue sous MySQL, en assurant la cohérence et l'intégrité des données médicales.

Nous avons également développé une application mobile à l'aide de Flutter, destinée aux médecins, pour qu'ils puissent consulter leurs rendez-vous et notifications à distance. Côté sécurité, nous avons mis en œuvre le système JWT pour l'authentification mobile, et nous avons utilisé le hachage des mots de passe afin de garantir la confidentialité des informations sensibles.

Pour l'aspect visuel, des outils comme NicePage et Adobe Photoshop ont été employés afin de produire une interface utilisateur moderne, intuitive et adaptée aux besoins du personnel médical et administratif.

Ce projet a représenté pour nous une réelle opportunité d'approfondir nos compétences techniques tout en répondant à un besoin concret du domaine médical. Bien que le système soit fonctionnel et conforme aux exigences de départ, plusieurs pistes d'amélioration peuvent être envisagées, telles que l'intégration de fonctionnalités avancées comme la téléconsultation ou la messagerie sécurisée entre les professionnels de santé.

Bibliographie

- [1] “Mediboard.” <http://www.mediboard.org/>.
- [2] “Rgpd hébergement sécurisé.” <https://www.cnil.fr/fr/reglement-europeen-protection-donnees>.
- [3] “Doctolib.” <https://www.doctolib.fr/>.
- [4] “Openmrs.” <https://openmrs.org/>.
- [5] “Introduction uml.” <https://web.maths.unsw.edu.au/~lafaye/CCM/uml/umlintro.htm>.
- [6] “Processus unifié.” https://www.academia.edu/39598956/Processus_Unifi%C3%A9_UP_and_2TUP.
- [7] “Modelio.” <https://www.modelio.org/>.
- [8] “Draw.io.” <https://www.draw.io/>.
- [9] “Tutoriel diagramme de cas d’utilisation.” https://atefsd.weebly.com/uploads/5/0/3/6/503639/csi_02_chap02.pdf.
- [10] “Tutoriel diagramme de séquence système.” <https://creately.com/blog/fr/uncategorized-fr/tutoriel-sur-le-diagramme-de-sequence/>.
- [11] “Tutoriel diagramme de classes.” <https://creately.com/blog/fr/uncategorized-fr/tutoriel-sur-le-diagramme-de-classes/>.
- [12] “Présentation github.” <https://www.pierre-giraud.com/git-github-apprendre-cours/presentation-git-github/>.
- [13] “Laravel.” <https://laravel.com/docs/5.4>.
- [14] “Laragon.” <https://laragon.org/>.
- [15] “Définition jwt.” <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/json-web-token-jwt/>.
- [16] “Principe de la fonction de hashage.” <https://zestedesavoir.com/tutoriels/1895/les-fonctions-de-hachage-cryptographiques/#2-quest-ce-quune-fonction-de-hachage>.
- [17] “Présentation de nice page.” <https://nicepage.com/>.
- [18] “Présentation visual studio code.” <https://www.gladir.com/SOFTWARE/VISUALSTUDIOCODE/presentation.htm>.

Résumé

Dans le cadre de notre projet Alliance Médical, nous avons conçu et réalisé une application web destinée à gérer efficacement les dossiers des patients, les médecins, les rendez-vous, les prescriptions et la communication entre les différents acteurs d'une clinique. Le système a été développé avec Laravel 5.4 pour le back-end, MySQL comme base de données relationnelle. Pour la modélisation, nous avons utilisé UML avec Modelio, ce qui nous a permis de structurer les besoins fonctionnels à travers des diagrammes de cas d'utilisation, de classes et de séquences. La sécurité a été renforcée grâce à l'utilisation de JWT (JSON Web Token) pour l'authentification API, ainsi que du hachage des mots de passe avec bcrypt, garantissant ainsi la protection des données médicales sensibles. Ce projet nous a permis de mettre en œuvre des outils professionnels tout en répondant à un besoin concret dans le domaine de la santé.

Mot clé : Alliance Médical, rendez-vous médical, dossiers électronique patients, application web.

Abstract

As part of our Alliance Médical project, we designed and developed a web application to efficiently manage patient records, doctors, appointments, prescriptions, and communication between the various stakeholders in a clinic. The system was developed using Laravel 5.4 for the back-end and MySQL as the relational database. For modeling, we used UML with Modelio, which allowed us to structure functional requirements through use case, class, and sequence diagrams. Security was enhanced through the use of JWT (JSON Web Token) for API authentication, as well as password hashing with bcrypt, ensuring the protection of sensitive medical data. This project allowed us to implement professional tools while meeting a concrete need in the healthcare field.

Keyword : Alliance Médical, registration, patients management, Web application, UML , Laravel,MySQL.

ملخص

تطبيق ويب مخصص لإدارة ملفات المرضى والأطباء والمواعيد والوصفات الطبية والتواصل بين مختلف الأطراف الفاعلة في العيادة بشكل فعال. تم تطوير النظام باستخدام Laravel 5.4 للخلفية، و MySQL كقاعدة بيانات علائقية. بالنسبة للنمذجة، استخدمنا UML مع Modelio ، مما سمح لنا بتنظيم الاحتياجات الوظيفية من خلال مخططات حالات الاستخدام والطبقات والتسلسلات. تم تعزيز الأمان من خلال استخدام JWT (JSON Web Token) للمصادقة على واجهة برمجة التطبيقات، بالإضافة إلى تجزئة كلمات المرور باستخدام bcrypt ، مما يضمن حماية البيانات الطبية الحساسة. سمح لنا هذا المشروع بتطبيق أدوات احترافية مع تلبية حاجة ملموسة في مجال الصحة.