République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique Université M'hamed Bougara - Boumerdès



Faculté des Sciences
Département d'Informatique

Domaine : Mathématiques et Informatique

Filière : Informatique

Spécialité : Systèmes informatiques

Mémoire de fin de cycle en vue de l'obtention du Diplôme de Licence Académique

Thème

Développement d'une Application Web pour la Gestion et le Suivi des Dossiers des Patients

Présenté par :
DORBANE ABDERAHMANE AMINE
AMARA ISLAM
RIAL MOHAMED RAMI
ANANE YOUNES ZAKARIA

Soutenu le …../…./ ….. Devant le jury composé de
M. TOUZI Fayçal : Encadreur
M. BELKASMI Djamel : Examinateur

Remerciements :

Je tiens à exprimer mes sincères remerciements à notre encadreur, **M. TOUAZI Fayçal** , pour son accompagnement, ses
conseils précieux, sa disponibilité ainsi que son soutien tout au long de la réalisation de ce projet. Grâce à son expérience et à ses orientations, nous avons pu mener ce travail dans de bonnes conditions.

Je remercie également M. Belkasmi, notre examinateur, pour le temps consacré à l'évaluation de ce mémoire ainsi que pour ses remarques et observations constructives.

Enfin, je remercie toutes les personnes ayant contribué de près ou de loin à la réalisation de ce travail

Table des matières

TABLE DES FIGURES 5
1 Présentation générale 1
2 L'intégration de l'intelligence artificielle dans les PACS 1
3 Le besoin d'une solution PACS-like en Algérie 2
4 Problematique specifique generale des radiologie 2
5 Solution proposée : une plateforme web dédiée à l'aide au diagnostic 3
1. Présentation générale du système étudié 4
2. Analyse fonctionnelle 4
2.1. Couverture fonctionnelle limitée 4
2.2. Absence d'approche centrée patient 5
3. Analyse structurelle et modélisation des données 5
3.1. Modèle de données incomplet 5
3.2. Manque de normalisation et d'interopérabilité 5
4. Analyse des acteurs et des interactions 5
4.1. Modélisation des acteurs insuffisante 5
4.2. Faible collaboration inter-services 6
5. Analyse architecturale 6
5.1. Architecture monolithique 6
5.2. Absence de modularité 6
6. Sécurité et gestion des accès 6
7. Synthèse critique 7
8. Positionnement de notre projet 7
1.1 Introduction 8
1.2 Les objectifs 8
1.3 Les besoins fonctionnels 8
2.1 Introduction 9
2.2 L'identification des acteurs 9
2.3 Le diagramme général des cas d'utilisation 9
Tableau descriptif du diagramme général des cas d'utilisation ..... **Erreur ! Signet non défini.**
Acteur Fonctionnalités principales ..... **Erreur ! Signet non défini.**
Acteur Fonctionnalités principales ..... **Erreur ! Signet non défini.**
Tableau descriptif du diagramme des cas d'utilisation de medecin ..... **Erreur ! Signet non défini.**
2.4 Le diagramme de classe 16
2.5 Le modèle MVC 20
2.6 Le pattern DAO 20
2.7 Le diagramme de packages 21
2.8 Diagrammes des séquences 1 ajouter pation : ..... **Erreur ! Signet non défini.**

3.1 Introduction 26
3.2 Environnement Logiciel 26
3.2.1 Authentification et gestion des rôles 27
3.2.2 Gestion des patients 27
3.2.3 Gestion des rendez-vous 27
3.2.4 Gestion des services et modèles IA 27
3.2.5 Gestion des images radiologiques et annotations 28
3.2.6 Analyse par intelligence artificielle 28
3.3 Communication entre React et Django 28
3.4 Sécurité 28
3.5 Interfaces de l'application web 29
3.5.1 Interface d'authentification 29
3.5.2 Interface de gestion des services 29

TABLE DES FIGURES

Figure	Titre	Page
Figure 1	Diagramme général des cas d'utilisation…….	13
Figure 2	Diagramme des cas d'utilisation du médecin……..	15
Figure 3	Diagramme des cas d'utilisation pour le chef de service……	17
Figure 4	Diagramme des cas d'utilisation pour l'administrateur……	17
Figure 5	Diagramme des cas d'utilisation pour le secrétaire……	18
Figure 6	Diagramme de classe : model…….	19
Figure 7	Diagramme de classe : Services……..	20
Figure 8	Diagramme de classe : React views……	20
Figure 9	Diagramme de classe : Real components……	21
Figure 10	Diagramme de classe : Main……...	21
Figure 11	Diagramme des packages……	24
Figure 12	Diagramme de séquence — Ajouter un patient……..	24
Figure 13	Diagramme de séquence — Modifier un rendez-vous (RDV)……...	25
Figure 14	Diagramme de séquence — Authentification (login)…….	25
Figure 15	Diagramme de séquence — Ajouter un service……..	25
Figure 16	Diagramme de séquence — Ajouter une structure…….	26
Figure 17	Diagramme de séquence — Gérer l'IA……...	26
Figure 18	Représentation schématique du pattern DAO…….	23
Figure 19	Interface dédiée à l'authentification……	32
Figure 20	Interface de gestion des services avec choix des modèles IA…….	32

Introduction Générale

1 Présentation générale

Le domaine de l'imagerie médicale connaît aujourd'hui une transformation importante grâce au développement des technologies numériques et des systèmes informatiques de gestion des données médicales. Les établissements de santé produisent quotidiennement un volume considérable d'images radiologiques telles que les radiographies, les scanners, les IRM, les mammographies et les échographies. Cette évolution a rendu nécessaire l'utilisation de solutions modernes capables d'assurer le stockage, l'organisation, la consultation et le partage sécurisé des informations médicales.

Parmi les technologies les plus utilisées dans ce domaine figure le PACS (Picture Archiving and Communication System), un système permettant la gestion numérique des images médicales et des dossiers radiologiques. Les systèmes PACS offrent plusieurs avantages tels que l'accès rapide aux examens médicaux, l'amélioration du suivi des patients ainsi qu'une meilleure collaboration entre les professionnels de santé.

Cependant, malgré les progrès réalisés, plusieurs défis persistent encore, notamment l'absence d'intégration avancée de l'intelligence artificielle dans certains systèmes existants, les difficultés de gestion des dossiers médicaux et le manque de solutions centralisées dans plusieurs établissements de santé. En Algérie particulièrement, la transition vers la digitalisation du secteur médical reste confrontée à différents obstacles liés aux infrastructures et à l'organisation des données médicales.

Dans ce contexte, notre projet intitulé « Développement d'une Application Web pour la Gestion et le Suivi des Dossiers des Patients », encadré par Touazi Fayçal, vise à développer une plateforme web moderne inspirée des systèmes PACS afin d'améliorer la gestion des dossiers médicaux et de faciliter le travail des professionnels de santé

2 L'intégration de l'intelligence artificielle dans les PACS

Avec l'évolution des technologies d'intelligence artificielle, les systèmes PACS deviennent de plus en plus intelligents et performants. L'intelligence artificielle permet aujourd'hui d'automatiser certaines tâches médicales complexes, notamment dans le domaine de l'analyse des images radiologiques.

Les algorithmes de machine learning et de deep learning sont capables de détecter automatiquement différentes anomalies médicales comme les tumeurs, fractures, infections ou lésions suspectes. Ces technologies assistent les radiologues dans l'interprétation des examens médicaux tout en améliorant la rapidité et la précision du diagnostic.

L'intégration de l'IA dans les systèmes PACS présente plusieurs avantages :

- réduction de la charge de travail des radiologues ;
- automatisation de certaines tâches répétitives ;
- amélioration de la qualité des diagnostics ;
- diminution du risque d'erreurs médicales ;

- meilleure gestion du flux de travail médical.

Malgré ces avancées, plusieurs plateformes PACS actuelles nécessitent encore des outils externes pour intégrer des modèles d'intelligence artificielle, ce qui limite l'automatisation complète du processus d'analyse médicale.

3 Le besoin d'une solution PACS-like en Algérie

En Algérie, le secteur de la santé connaît progressivement une transition vers la digitalisation. Toutefois, plusieurs difficultés persistent encore au sein des établissements hospitaliers et des centres médicaux.

Parmi les principaux problèmes observés :

- absence de systèmes centralisés performants ;
- faible interconnexion entre les services médicaux ;
- difficultés de partage des dossiers médicaux ;
- stockage non optimisé des images radiologiques ;
- accès limité aux données médicales dans certaines régions.

Ces contraintes ralentissent la prise en charge des patients et compliquent le travail des professionnels de santé, notamment dans le domaine de la radiologie

2.1 Problématique spécifique générale des radiologies

Le domaine de la radiologie représente aujourd'hui un élément essentiel dans le diagnostic médical. Cependant, les radiologues sont confrontés à plusieurs difficultés liées à l'augmentation du nombre d'examens médicaux et à la complexité de certaines analyses radiologiques.

Les principales problématiques rencontrées sont :

- volume très important d'images médicales à analyser ;
- annotation manuelle longue et répétitive ;
- fatigue des radiologues ;
- risque d'erreurs diagnostiques ;
- variabilité des interprétations entre médecins ;
- difficulté d'analyse des cas complexes ou des images de faible qualité.

Dans certains cas, notamment pour les mammographies ou les examens liés au cancer du sein, la précision du diagnostic est extrêmement importante afin de détecter rapidement les anomalies suspectes et améliorer les chances de guérison des patients.

Par ailleurs, le manque d'outils intelligents capables d'assister automatiquement les spécialistes rend le processus d'analyse plus lent et plus difficile, particulièrement dans les établissements disposant de ressources limitées.

2.2 5 Solution proposée : une plateforme web dédiée à l'aide au diagnostic

Pour répondre à ces défis, une application web est proposée. Celle-ci est spécifiquement conçue pour accompagner les professionnels de santé, en particulier les radiologues et spécialistes du sein, dans le dépistage du cancer du sein.
Les fonctionnalités principales actuellement disponibles ou envisagées à terme incluent :

- Annotation et segmentation des régions d'intérêt sur les mammographies, IRM et échographies ;
- Identification facilitée des anomalies suspectes (fonctionnalité prévue avec l'intégration future de l'IA) ;
- Participation à la constitution de bases de données annotées, en vue de l'entraînement ultérieur de modèles d'intelligence artificielle.

Objectifs du projet :

- Réduction de la charge de travail des radiologues ;
- Automatisation partielle des tâches d'annotation ;
- Amélioration de la précision et fiabilité du diagnostic ;
- Accélération du processus d'analyse ;
- Standardisation des annotations pour réduire la variabilité humaine ;
- Contribution à la recherche médicale par la génération de bases de données d'apprentissage ;
- Facilitation de l'accès au dépistage dans les régions sous-équipées.

Ainsi, ce projet s'inscrit dans une dynamique globale de modernisation du système de santé algérien et de réduction des inégalités d'accès aux soins, tout en s'inscrivant dans une perspective d'exploitation future des avancées en intelligence artificielle.

Chapitre 1 : Étude de l'existant

Dans le cadre de ce travail, il est essentiel de préciser que notre projet constitue une continuité d'un projet antérieur consacré à la gestion du service de radiologie. Ainsi, l'étude de l'existant vise à analyser de manière critique ce système afin d'identifier ses limites et de proposer des améliorations cohérentes.

Le projet étudié consiste en le développement d'une application dédiée à la gestion du service de radiologie. Son objectif principal est de numériser certaines activités du service, notamment la gestion des patients, des examens radiologiques et des comptes rendus. Le système repose sur une architecture applicative centralisée, associée à une base de données destinée au stockage des informations relatives aux patients et aux examens. Toutefois, les fonctionnalités proposées restent limitées à un périmètre fonctionnel restreint, exclusivement orienté vers la radiologie.

Sur le plan fonctionnel, le système permet la gestion des patients, la planification des examens ainsi que la production de comptes rendus radiologiques. Néanmoins, plusieurs fonctionnalités essentielles sont absentes, notamment l'intégration des résultats de laboratoire, la gestion des prescriptions médicales, le suivi longitudinal du patient et la coordination entre les différents services. De ce fait, le système ne permet pas une prise en charge globale du patient, ce qui constitue une limite majeure dans un contexte hospitalier.

Par ailleurs, le modèle fonctionnel adopté est orienté vers le service de radiologie plutôt que vers le patient. Or, les systèmes de santé modernes reposent sur une approche centrée sur le patient, dans laquelle l'ensemble des informations cliniques, biologiques et thérapeutiques sont centralisées au sein d'un dossier unique. Dans le système étudié, les données demeurent fragmentées et non interconnectées, ce qui limite la continuité des soins, la qualité de la prise de décision médicale et les possibilités d'exploitation des données.

2.3 1. Présentation générale du système étudié

Le projet analysé consiste en le développement d'une application de gestion du service de radiologie. L'objectif principal est de numériser certaines activités du service, notamment la gestion des patients, des examens radiologiques et des comptes rendus.

Le système repose sur une architecture applicative centralisée, avec une base de données dédiée au stockage des informations relatives aux patients et aux examens. Les fonctionnalités proposées restent cependant limitées à un périmètre fonctionnel restreint, exclusivement orienté vers la radiologie.

2.4 2. Analyse fonctionnelle

2.1. Couverture fonctionnelle limitée

Le système prend en charge :

- la gestion des patients (création, consultation),
- la planification des examens,
- la production de comptes rendus radiologiques.

Cependant, plusieurs fonctionnalités essentielles sont absentes :

- intégration des résultats de laboratoire,
- gestion des prescriptions médicales,

- suivi longitudinal du patient,
- coordination entre services.

Ainsi, le système ne permet pas une prise en charge globale du patient, ce qui constitue une limite majeure dans un contexte hospitalier.

2.2. Absence d'approche centrée patient

Le modèle fonctionnel est orienté « service » (radiologie) plutôt que « patient ».
Un système moderne de santé repose sur une vision transversale du dossier patient, où toutes les informations (cliniques, biologiques, thérapeutiques) sont centralisées.

Dans le système étudié, les données sont fragmentées et non interconnectées, ce qui empêche :

- une continuité des soins,
- une prise de décision médicale éclairée,
- une exploitation analytique des données.

2.5 3. Analyse structurelle et modélisation des données

3.1. Modèle de données incomplet

Le modèle de données ne prend pas en compte plusieurs entités fondamentales :

- structures hospitalières (hôpital, service, unité),
- spécialités médicales,
- parcours patient,
- historique médical complet.

L'absence de ces entités limite fortement l'évolutivité du système.

3.2. Manque de normalisation et d'interopérabilité

Aucune référence à des standards médicaux n'est mentionnée (HL7, FHIR, DICOM, etc.).
Cela implique :

- une difficulté d'intégration avec d'autres systèmes,
- une impossibilité d'échange de données inter-hospitalier,
- une dépendance forte à l'implémentation locale.

Ce point est critique dans les systèmes de santé modernes, où l'interopérabilité est un enjeu majeur.

2.6 4. Analyse des acteurs et des interactions

4.1. Modélisation des acteurs insuffisante

Les acteurs identifiés sont limités à :

- radiologue,
- chef de service,
- administrateur,
- secrétaire.

Cette modélisation ignore plusieurs acteurs clés :

- médecins spécialistes,
- biologistes,
- pharmaciens,
- personnel infirmier.

Cela traduit une vision cloisonnée du système d'information hospitalier.

4.2. Faible collaboration inter-services

L'absence d'acteurs multi-spécialités empêche :

- le partage d'informations médicales,
- la coordination des soins,
- la gestion collaborative des cas complexes.

Le système ne répond donc pas aux exigences d'un environnement hospitalier réel.

2.7 5. Analyse architecturale

5.1. Architecture monolithique

Le système semble reposer sur une architecture monolithique, ce qui pose plusieurs limites :

- faible scalabilité,
- difficulté de maintenance,
- complexité d'évolution vers un système global.

5.2. Absence de modularité

Aucun découpage en modules fonctionnels (radiologie, laboratoire, pharmacie, etc.) n'est proposé.
Cela rend difficile l'extension vers un Dossier Médical Électronique (DME) complet.

2.8 6. Sécurité et gestion des accès

Le mémoire ne détaille pas :

- les mécanismes d'authentification,
- la gestion des droits d'accès (RBAC),
- la protection des données sensibles.

Or, les données médicales sont hautement sensibles et nécessitent :

- traçabilité,
- confidentialité,
- conformité réglementaire.

Cette absence constitue une lacune critique.

2.9 7. Synthèse critique

L’analyse met en évidence plusieurs limitations majeures :

- **Périmètre fonctionnel restreint** : limité à la radiologie.
- **Absence de vision patient globale**.
- **Modèle de données incomplet et non extensible**.
- **Manque d’interopérabilité avec les standards médicaux**.
- **Modélisation insuffisante des acteurs**.
- **Architecture peu évolutive**.
- **Absence de mécanismes de sécurité clairement définis**.

2.10 8. Positionnement de notre projet

Face à ces limitations, notre projet adopte une approche plus globale et systémique :

- intégration multi-services (radiologie, laboratoire, etc.),
- modélisation des structures hospitalières,
- prise en compte de multiples acteurs médicaux,
- adoption d’une architecture modulaire,
- orientation vers un DME unifié et interopérable,
- mise en place de mécanismes de sécurité adaptés.

Cette approche permet de répondre aux exigences réelles des systèmes d’information hospitaliers modernes et constitue une amélioration significative par rapport à l’existant.

Chapitre 2 : Analyse

1.1 Introduction

Ce chapitre présente les objectifs, les besoins fonctionnels et la conception de l'application web développée pour les services de radiologie. Le projet vise à centraliser la gestion des dossiers médicaux et à améliorer l'organisation des processus cliniques.

1.2 Les objectifs

Le projet a pour objectif de doter les services de radiologie d'une plateforme numérique avancée facilitant la gestion centralisée et sécurisée des dossiers médicaux des patients. Il s'inscrit dans une démarche de digitalisation des pratiques cliniques et d'optimisation du suivi radiologique. L'application proposera notamment :
• Un accès structuré aux examens radiologiques (mammographies, échographies, IRM), classés par modalité ;
• Une interface intuitive permettant aux praticiens d'annoter manuellement les images à l'aide d'une palette dédiée, afin de signaler les zones d'intérêt et faciliter l'interprétation des examens.

Cette solution vise à améliorer la traçabilité des données, renforcer la collaboration entre professionnels de santé et offrir un environnement de travail adapté aux exigences cliniques modernes.

1.3 Les besoins fonctionnels

Les besoins fonctionnels de l'application peuvent être résumés comme suit :

– L'authentification sécurisée des utilisateurs à l'aide des tokens JWT ;
– La gestion des dossiers patients ;
– La gestion des rendez-vous médicaux ;
– La gestion des profils des radiologues ;
– La gestion des chefs de service ;
– La gestion des administrateurs ;
– La gestion des différents services hospitaliers ;
– La gestion des comptes utilisateurs ;
– Le téléversement et la consultation des images radiologiques ;
– La gestion des images médicales issues de la mammographie, de l'échographie et de l'IRM ;
– L'annotation manuelle des images radiologiques à l'aide d'une palette interactive ;
– La modification et la suppression des annotations ;
– Le zoom, dézoom et l'amélioration de l'affichage des images médicales ;
– La sauvegarde des annotations dans la base de données ;
– L'accès aux fonctionnalités selon les rôles des utilisateurs ;
– La communication entre le frontend React et le backend Django via des API REST ;
– La préparation de l'intégration future d'outils d'intelligence artificielle pour l'aide au diagnostic
– L'accès aux fonctionnalités selon les rôles des utilisateurs ;
– La communication entre le frontend React et le backend Django via des API REST ;
– La préparation de l'intégration d'outils d'intelligence artificielle pour l'aide au diagnostic.

Chapitre 2 : Conception

2.1 Introduction

La phase de conception constitue une étape clé dans le cycle de vie du développement logiciel. Elle permet de traduire les exigences fonctionnelles et non fonctionnelles en une architecture logique, cohérente et structurée du système à implémenter. Cette étape vise à définir les composants fondamentaux, leurs responsabilités respectives, ainsi que les interactions qui régiront leur fonctionnement global. Dans le cadre du présent projet, qui s'inscrit dans le domaine de l'imagerie médicale et la gestion des dossiers médicaux, une architecture claire est indispensable pour garantir la maintenabilité, la modularité et la scalabilité de la solution. Pour ce faire, le modèle architectural retenu est le **paradigme MVC (Modèle–Vue–Contrôleur)**, largement reconnu dans les systèmes orientés web.

2.2 L'identification des acteurs

Avant de présenter les diagrammes d'interaction, il convient d'identifier les principaux acteurs du système.
Quatre acteurs sont distingués :

- ***médecin***:
  Le medecin est chargé de la gestion de ses propres patients ainsi que de leurs rendez-vous. Ses responsabilités incluent :
  - La gestion des dossiers patients (ajout, modification, suppression) ;
  - La gestion des images radiologiques, incluant l'annotation des anomalies ;
  - La gestion de son propre planning de rendez-vous.
- ***Chef de service*** :
  Le chef de service supervise les radiologues placés sous sa responsabilité. Il assure :
  - La gestion des radiologues ;
  - La gestion des patients suivis par les radiologues de son service ;
  - La supervision des rendez-vous associés.
- ***Administrateur*** :
  L'administrateur est responsable de :
  - La gestion des chefs de service ;
  - La gestion des services eux-mêmes, notamment la création et l'organisation des services médicaux dans le système.
- ***Secrétaire*** :
  Il a la possibilité de :
  - Consulter les dossiers des patients, sans pouvoir les modifier ;
  - Gérer le rendez-vous ;
  - Gérer son propre compte.

2.3 Le diagramme général des cas d'utilisation

Definition :
Le diagramme de cas d'utilisation est un diagramme UML qui permet de représenter les fonctionnalités offertes par un système du point de vue des utilisateurs. Il met en évidence les interactions entre les acteurs (utilisateurs ou

systèmes externes) et les différents cas d'utilisation, afin de décrire les services que le système doit fournir. Ce diagramme est principalement utilisé lors de l'analyse des besoins fonctionnels et facilite la compréhension des exigences du système.

Afin d'illustrer les interactions entre les différents acteurs et le système, nous présentons dans un premier temps le diagramme global des cas d'utilisation. Ce diagramme constitue un outil essentiel pour modéliser les principales fonctionnalités attendues et pour définir clairement les rôles et responsabilités de chaque acteur au sein du système.

Une description détaillée sera ensuite proposée, afin de mieux comprendre les flux fonctionnels et d'assurer une conception rigoureuse de la solution.
Ce diagramme global offre ainsi une vue d'ensemble structurée des services fournis par l'application.

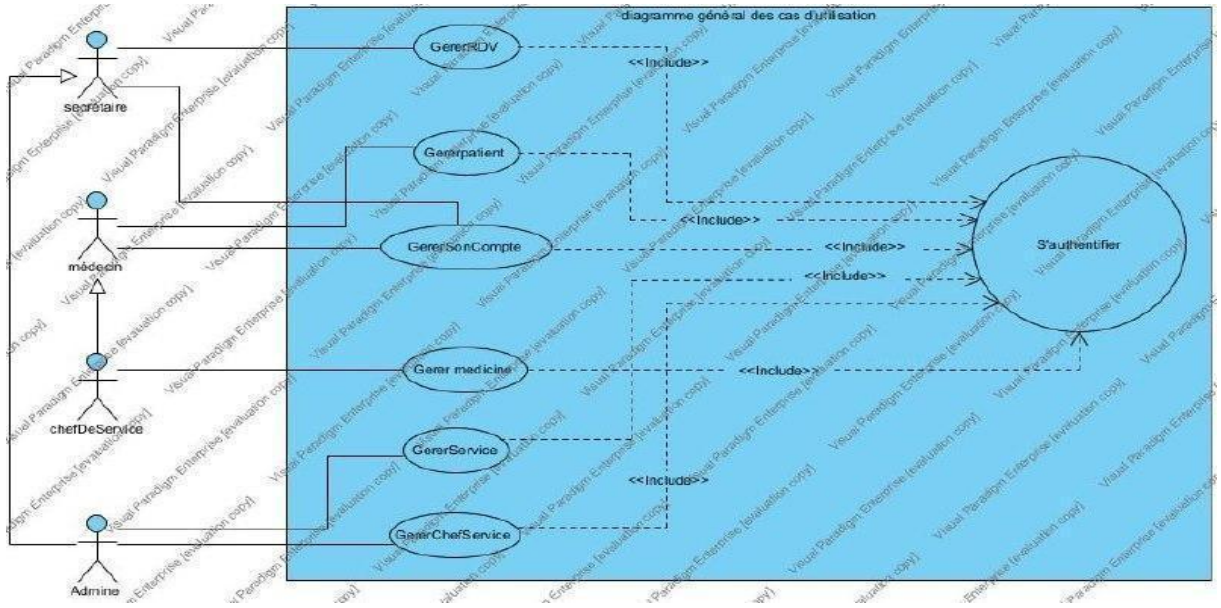


Figure 1: Diagramme général des cas d'utilisation

Ce diagramme offre une vue d'ensemble des interactions entre les quatre acteurs principaux du système (médecin, chef de service, administrateur et secrétaire) et les fonctionnalités disponibles. Chaque acteur dispose d'un ensemble de cas d'utilisation correspondant à son rôle : le secrétaire gère les patients et les rendez-vous, le médecin gère les dossiers médicaux et les images radiologiques, le chef de service supervise les activités médicales, et l'administrateur administre les services hospitaliers et les comptes utilisateurs. Tous les acteurs partagent la fonctionnalité d'authentification sécurisée au système

Tableau 1 : Description textuelle du diagramme général des cas d'utilisation

Acteur	Fonctionnalité	Description
Secrétaire	Gérer patient	Permet l'ajout, la modification et la gestion des dossiers des patients.
Médecin	Gérer son compte	Permet au médecin de modifier ses informations personnelles et consulter son espace.
Médecin	Consulter les dossiers médicaux	Accès aux informations médicales et radiologiques des patients.
Chef de service	Superviser les activités	Contrôle et suivi des opérations du service

Acteur	Fonctionnalité	Description
	médicales	médical.
Administrateur	Gérer médecins	Ajout, modification et suppression des comptes médecins.
Administrateur	Gérer services	Gestion des différents services hospitaliers.
Administrateur	Gérer chefs de service	Administration des responsables des services médicaux.
Tous les utilisateurs	S'authentifier	Permet l'accès sécurisé au système via une authentification.

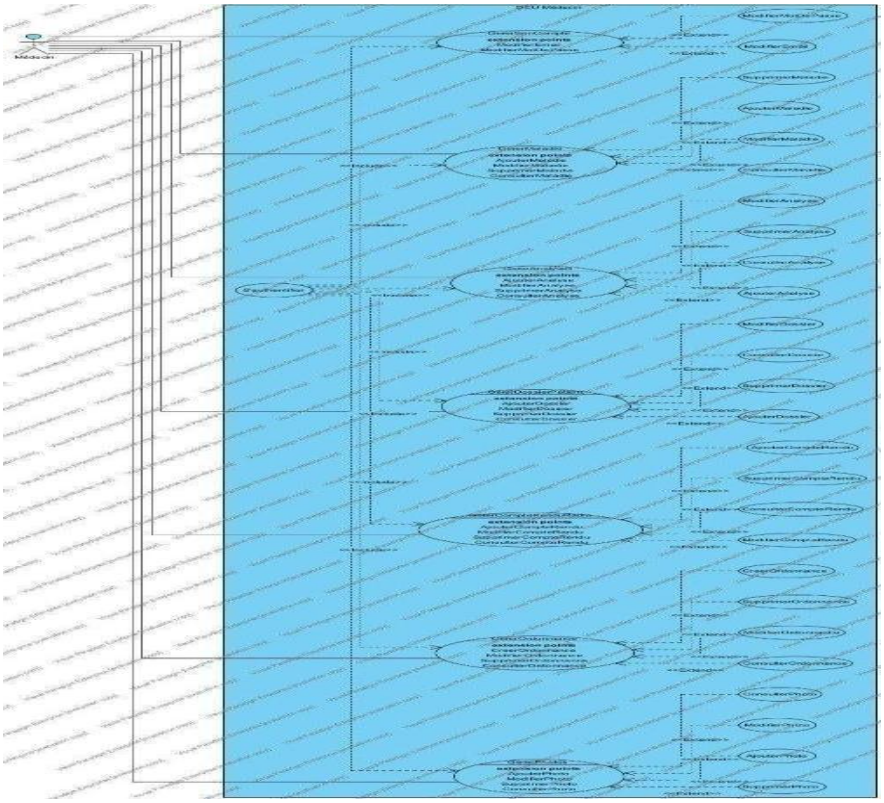


Figure 2: Diagramme général des cas d'utilisation de medecin

Ce diagramme présente en détail l'ensemble des fonctionnalités accessibles au médecin. Il peut gérer son compte personnel, administrer les dossiers médicaux complets de ses patients (ajout, modification, suppression, consultation), gérer les maladies et analyses, créer et gérer les ordonnances et comptes rendus, ainsi que téléverser et annoter les images radiologiques. L'authentification préalable est une précondition à l'accès à toutes ces fonctionnalités

Tableau 2 : Description textuelle du diagramme des cas d'utilisation du médecin

Fonctionnalité	Description
Gérer son compte	Permet au médecin de modifier son email et son mot de passe.
Ajouter maladie	Ajouter une maladie dans le dossier du patient.
Modifier maladie	Modifier les informations relatives à une maladie.
Supprimer maladie	Supprimer une maladie enregistrée.
Consulter maladie	Afficher les informations des maladies enregistrées.
Ajouter analyse	Ajouter une analyse médicale au dossier du patient.
Modifier analyse	Modifier une analyse existante.
Supprimer analyse	Supprimer une analyse médicale.
Consulter analyse	Consulter les résultats des analyses médicales.
Ajouter dossier	Créer un nouveau dossier médical pour un patient.
Modifier dossier	Mettre à jour les informations d'un dossier médical.

Fonctionnalité	Description
Supprimer dossier	Supprimer un dossier médical.
Consulter dossier	Consulter les informations du dossier médical.
Ajouter compte rendu	Ajouter un compte rendu médical.
Modifier compte rendu	Modifier un compte rendu existant.
Supprimer compte rendu	Supprimer un compte rendu médical.
Consulter compte rendu	Consulter les comptes rendus médicaux.
Créer ordonnance	Générer une ordonnance médicale.
Modifier ordonnance	Modifier une ordonnance existante.
Supprimer ordonnance	Supprimer une ordonnance.
Consulter ordonnance	Consulter les ordonnances médicales.
Ajouter photo	Ajouter une image ou photo médicale.
Modifier photo	Modifier une image médicale enregistrée.
Supprimer photo	Supprimer une image médicale.
Consulter photo	Afficher les images médicales du patient.
S'authentifier	Accéder au système de manière sécurisée avant toute opération.

Tableau 3 : Description textuelle du cas d'utilisation « Modifier un rendez-vous »

Élément	Description
Nom	Modifier un rendez-vous (RDV)
Objectif	Permettre à un secrétaire, médecin ou chef de service de modifier un rendez-vous existant en mettant à jour la date et le lieu.
Acteurs	Secrétaire, Médecin, Chef de service
Préconditions	L'utilisateur doit être authentifié et avoir un des rôles requis. Le rendez-vous doit exister dans la base de données.
Postconditions	Le rendez-vous est mis à jour avec les nouvelles informations de date et de lieu.
Enchaînement normal	1. L'utilisateur envoie une requête PUT avec les nouvelles informations. 2. Le système vérifie l'existence du rendez-vous. 3. Le système vérifie et valide les champs fournis. 4. Le système met à jour le rendez-vous et retourne une réponse de succès.
Enchaînement alternatif	Rendez-vous inexistant → erreur 404. Champs manquants → erreur 400. Format de date invalide → erreur 400. Erreur interne → erreur 500.

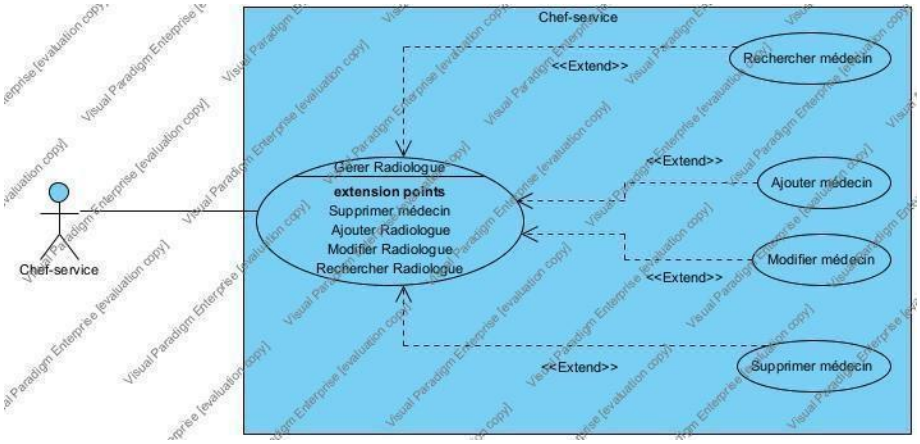


Figure 3: Diagramme des cas d'utilisation pour chef de servise

Ce diagramme présente les fonctionnalités spécifiques au chef de service. En tant que superviseur, il peut gérer les radiologues de son service, consulter et gérer les dossiers des patients qui leur sont associés, superviser les rendez-vous et s'authentifier au système. Son rôle intermédiaire lui confère un accès élargi par rapport au médecin tout en restant distinct des prérogatives de l'administrateur.

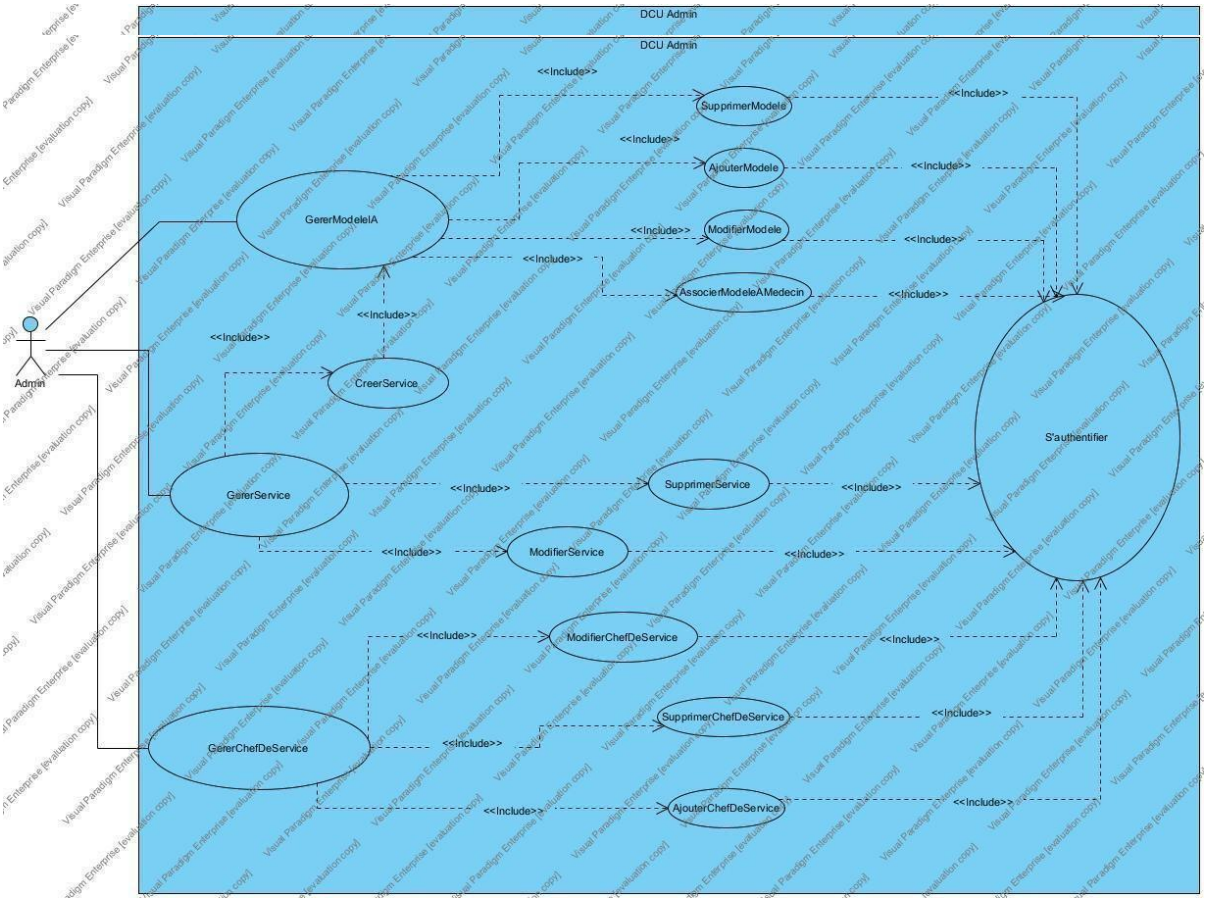


Figure 4: Diagramme des cas d'utilisation pour l'administrateur

Ce diagramme illustre les responsabilités de l'administrateur du système. Il dispose des droits les plus étendus : gestion des chefs de service, configuration des services hospitaliers (création, modification, suppression), gestion des modèles d'intelligence artificielle associés à chaque service, et administration des comptes utilisateurs. L'administrateur constitue le point central de configuration et de maintenance du système d'information

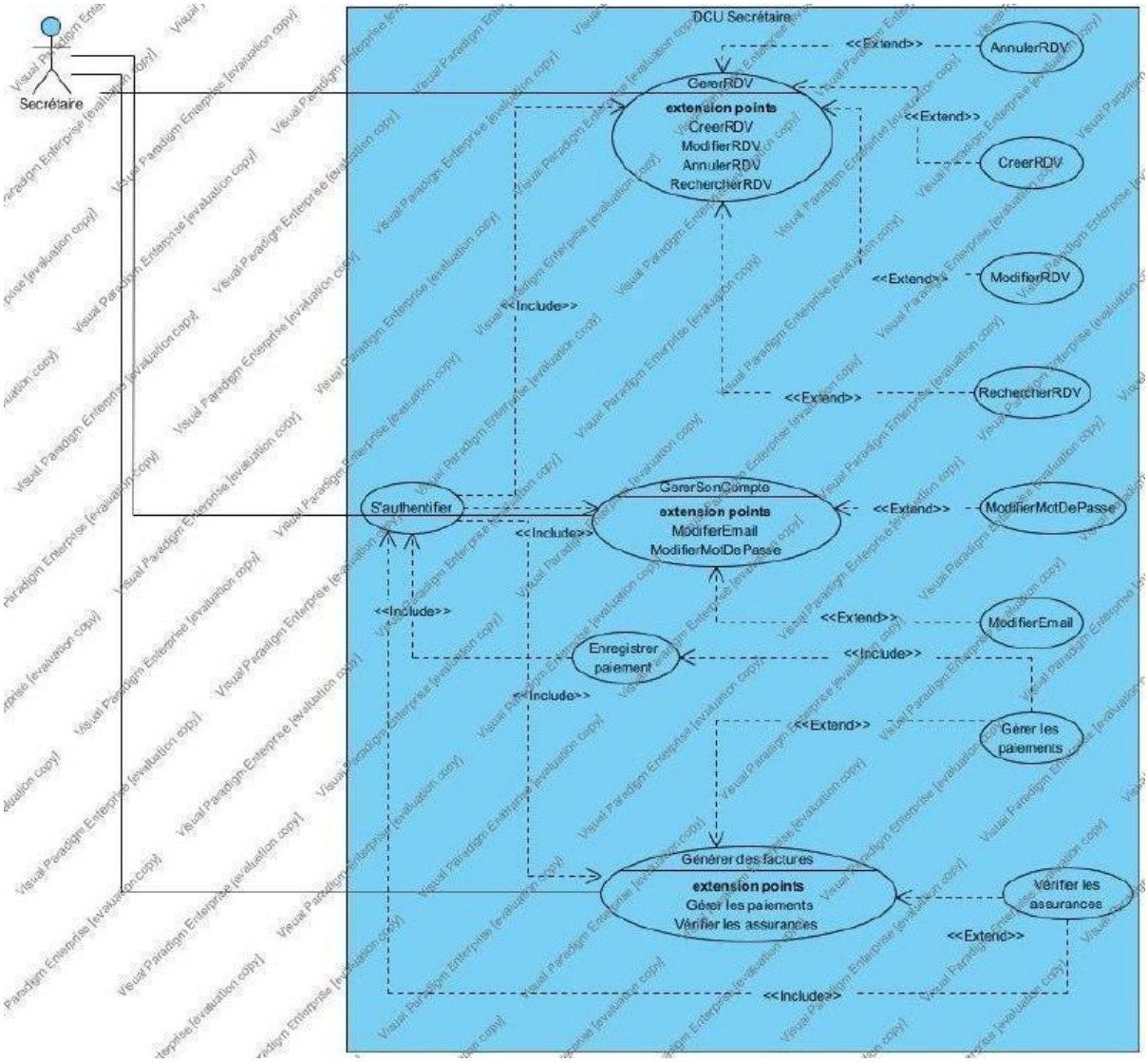


Figure 5: Diagramme de cas d'utlilisation de secretaire

Ce diagramme définit le rôle du secrétaire dans le système. Contrairement au médecin, le secrétaire dispose d'un accès en lecture seule aux dossiers médicaux des patients ; il ne peut pas les modifier. En revanche, il est habilité à gérer les rendez-vous (ajout, modification, suppression) et son propre compte. Ce profil garantit une séparation claire des responsabilités au sein de l'équipe médicale

2.4 Le diagramme de classe

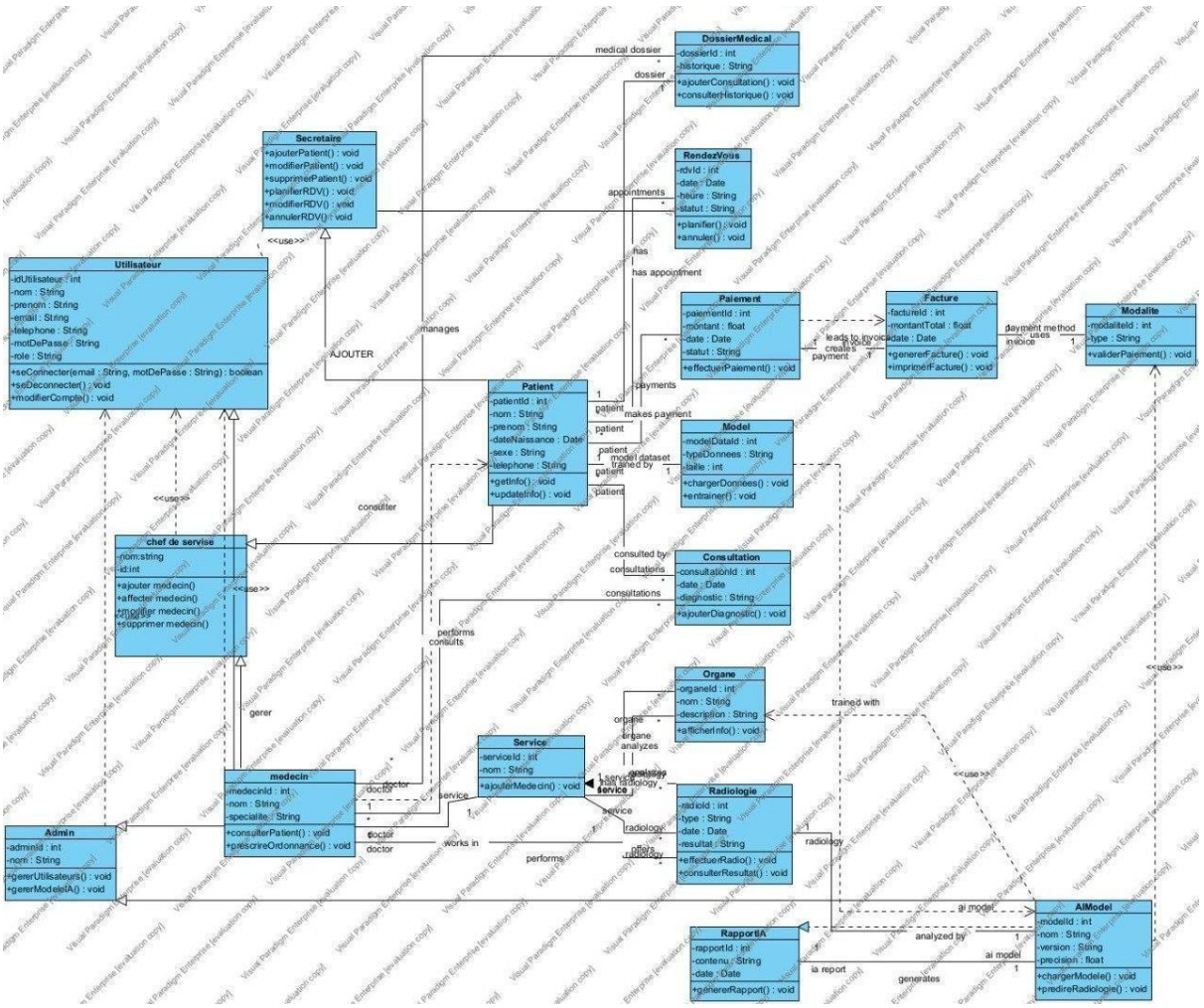


Figure 6: Diagramme de classe : model

Ce diagramme présente les classes du modèle de données de l'application, correspondant aux entités gérées par le backend Django. On y trouve les principales entités telles que Patient, Dossier, Examen, Image, RendezVous, Utilisateur et Service, avec leurs attributs et les associations qui les relient. Ces classes constituent la couche de persistance de l'application et définissent le schéma de la base de données MySQL

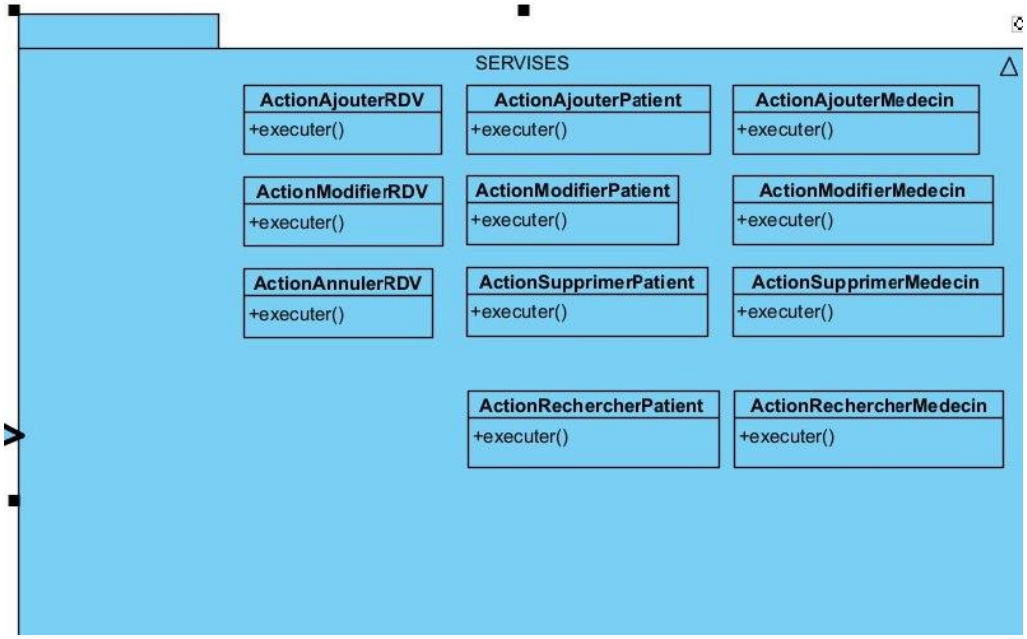


Figure 7: Diagramme de classe: Servises

Ce diagramme illustre la couche des services de l'application, qui encapsule la logique métier. Les classes de services font le lien entre les vues React et les données du backend via les API REST. Chaque service est responsable d'une fonctionnalité spécifique (gestion des patients, des rendez-vous, des images, etc.) et expose des méthodes appelées par les composants de l'interface utilisateur

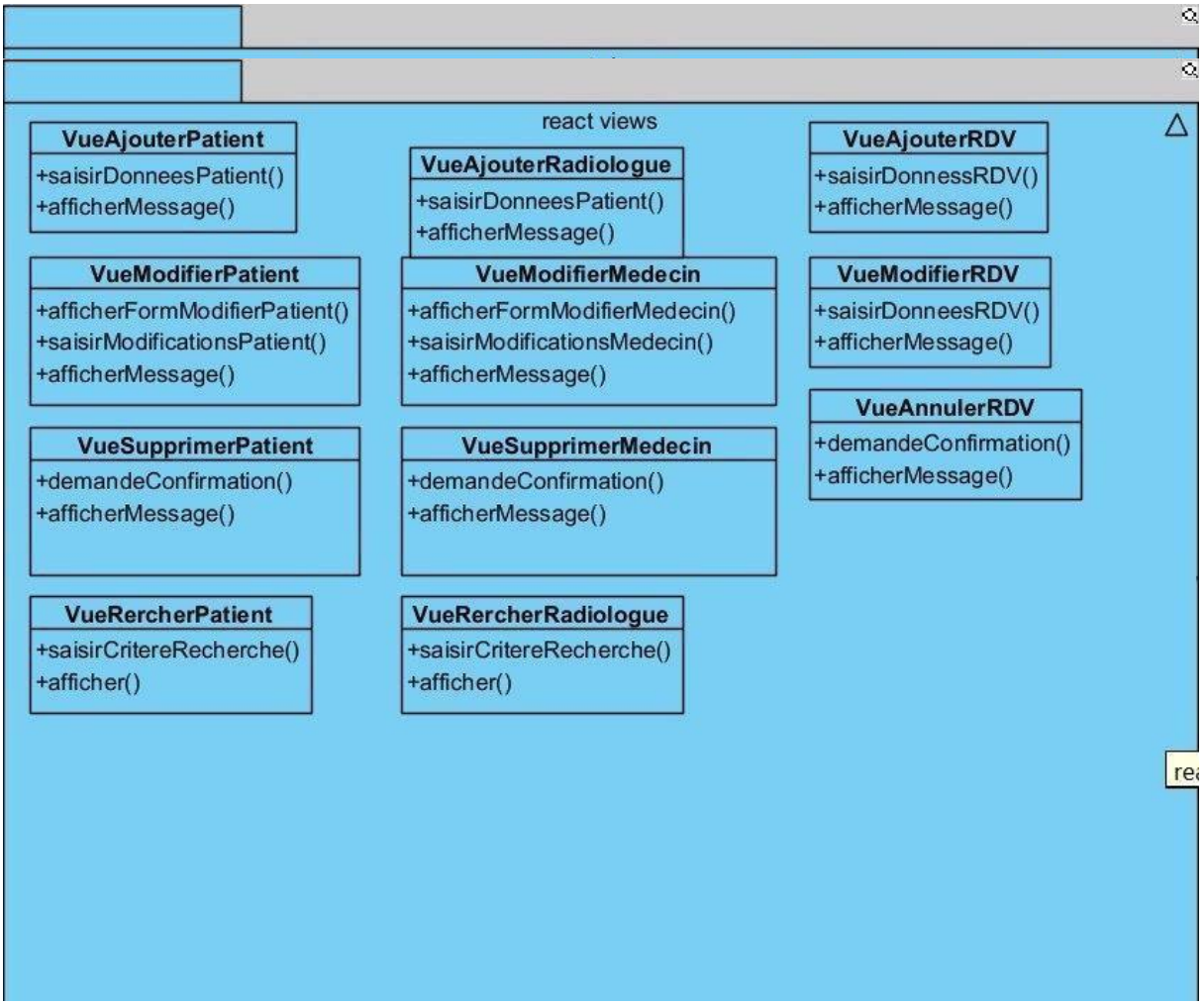


Figure 8: Diagramme de classe: React views

Ce diagramme représente la structure des vues (pages) de l'application React. Chaque vue correspond à une interface utilisateur distincte (tableau de bord, liste des patients, gestion des rendez-vous, consultation des images, etc.). Les vues interagissent avec les services pour récupérer et afficher les données, et transmettent les actions de l'utilisateur vers la couche de traitement.

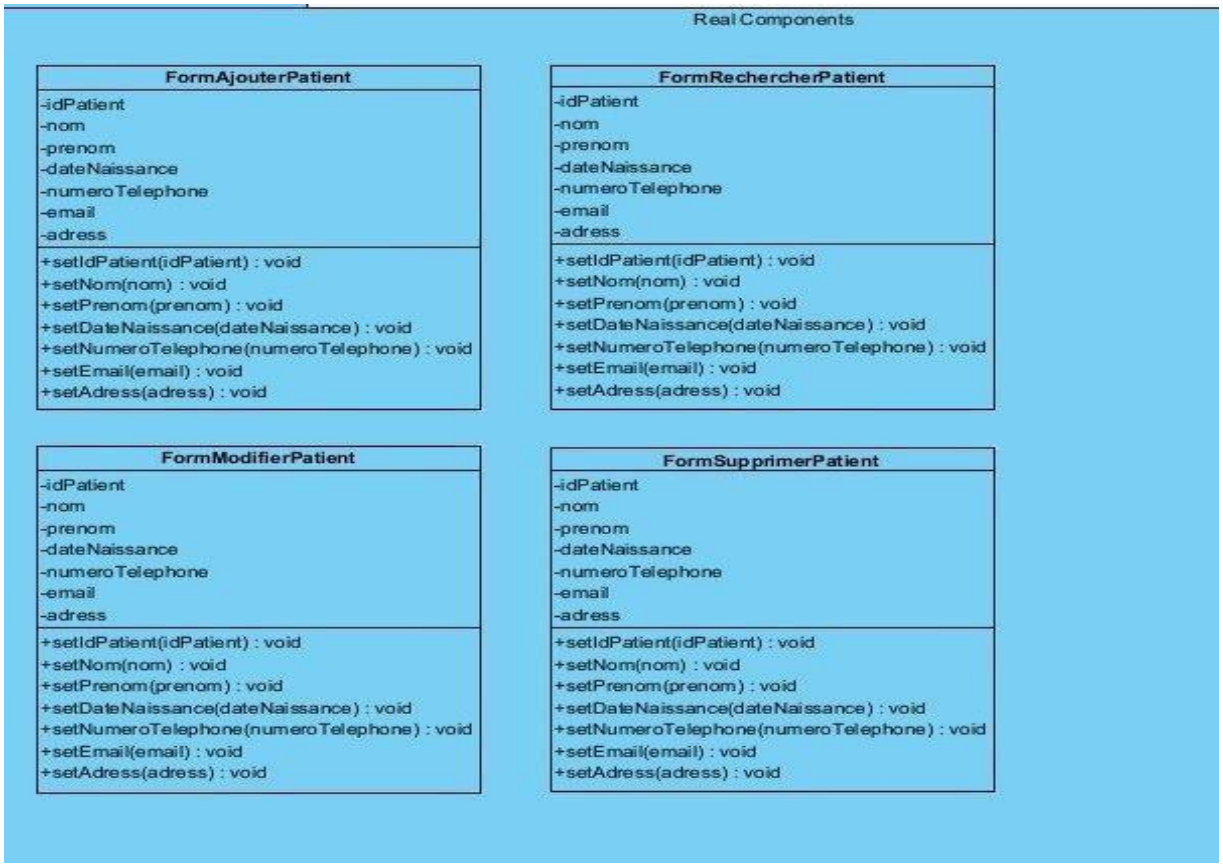


Figure 9: Diagramme de classe: Real components

Ce diagramme présente les composants réutilisables de l'interface React. Ces composants encapsulent des éléments d'interface spécifiques (formulaires, tableaux, modales, palettes d'annotation, etc.) et sont partagés entre plusieurs vues. Cette approche modulaire favorise la réutilisabilité du code et la cohérence visuelle de l'application

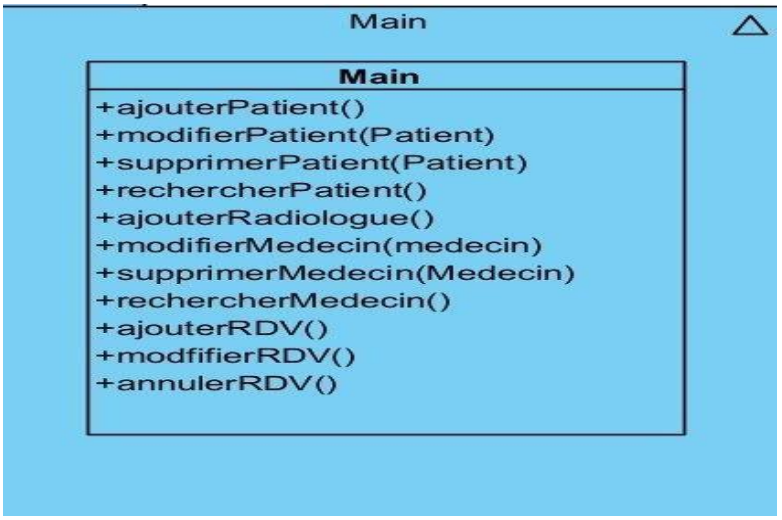


Figure 10: Diagramme de classe: Main

Ce diagramme présente la classe principale (Main) de l'application React, qui constitue le point d'entrée du frontend. Elle orchestre le routage entre les différentes vues, gère l'état global de l'application (authentification, rôle de l'utilisateur connecté) et initialise les composants de base. Ce niveau d'abstraction garantit une navigation cohérente entre les différentes sections de la plateforme.

2.5 Le modèle MVC

Le modèle MVC, acronyme de **Modèle–Vue–Contrôleur**, est un patron d'architecture logicielle qui vise à structurer les applications en trois composantes distinctes, chacune assumant une responsabilité spécifique. Cette séparation des préoccupations facilite la maintenance du code, la réutilisabilité des composants ainsi que l'évolutivité de l'application.

1. ***Le Modèle (Model) :***
   Le modèle constitue la couche métier de l'application. Il gère les données, les règles de gestion et les interactions avec les sources d'information telles que les bases de données. Il encapsule l'état des objets manipulés ainsi que leurs comportements, et toute modification des données ou des règles applicables doit s'effectuer à ce niveau.
2. ***La Vue (View) :***
   La vue correspond à la couche de présentation. Elle est responsable de l'affichage des informations issues du modèle à destination de l'utilisateur. Dépourvue de logique métier, elle se limite à restituer l'état du système. Une application peut proposer plusieurs vues pour un même modèle, adaptées à différents utilisateurs ou supports (web, mobile, etc.).
3. ***Le Contrôleur (Controller) :***
   Le contrôleur fait l'interface entre la vue et le modèle. Il interprète les actions de l'utilisateur, déclenche les traitements nécessaires, met à jour le modèle, et sélectionne la vue appropriée à afficher. Il joue ainsi un rôle central de coordination.

L'un des principaux atouts du modèle MVC réside dans sa capacité à isoler les différentes couches de l'application. Cette organisation favorise le développement modulaire, facilite le travail en équipe et améliore la testabilité du code, chaque composant pouvant être testé de manière indépendante.

Le modèle MVC est particulièrement adapté aux applications web modernes, où l'interface utilisateur évolue fréquemment, tandis que la logique métier demeure relativement stable.

2.6 Le pattern DAO

Le pattern DAO (Data Access Object) permet de faire le lien entre la couche métier et la couche de persistance.

- ***La couche métier :*** Elle correspond aux objets applicatifs. C'est cette couche qui gère les traitements métiers appliqués aux données. Ce sont à travers ces objets que l'on récupère et manipule les instances représentant les entités métiers issues de la base de données.
- ***La couche de persistance :*** Elle est responsable du stockage des données. Appelée également couche de données, elle gère les opérations de création, lecture, mise à jour et suppression (CRUD : Create, Read, Update, Delete). L'un des avantages majeurs de ce design pattern est qu'il permet d'isoler la logique métier des détails techniques liés au stockage, facilitant ainsi un éventuel changement de système de base de données (ex. : passage de SQL Server à Oracle

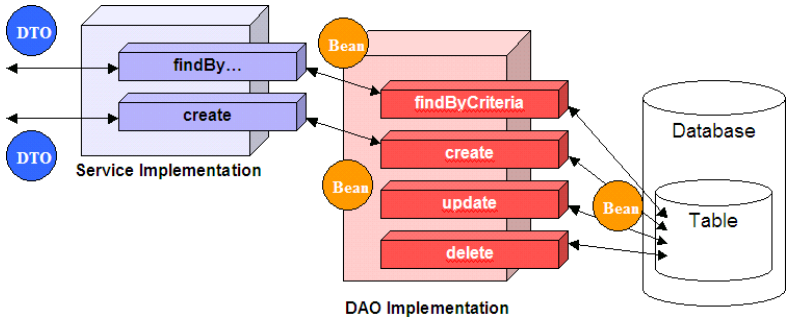


Figure 11: Représentation schématique du pattern DAO

Remarque : Le framework Django implémente automatiquement le pattern DAO via son système d'ORM (Object-Relational Mapping). Les modèles Django définissent la couche métier, tandis que les opérations CRUD sont directement prises en charge par l'ORM, ce qui rend inutile l'implémentation manuelle d'objets DAO. Cela permet un accès aux données à la fois simple, structuré et sécurisé.

2.7 Le diagramme de packages

Pour structurer notre modèle, nous allons organiser les classes et les regrouper en ensembles cohérents. Pour ce faire, nous utilisons une fois de plus le concept général d'UML, le package.

Nous allons donc structurer les classes identifiées jusqu'à présent en cinq packages :

- ***Package React views*** :

Contient toutes les interfaces graphiques.

- ***Package React components*** :

Abréviation de formulaire, qui transporte les informations de la vue vers l'action.

- ***Package Servises*** :

Contient les classes qui lancent les traitements de chaque cas d'utilisation.

- ***Package Model*** :

Utilisé par les actions pour accomplir les traitements de chaque cas d'utilisation. A chaque accès à la BDD

- ***Package DAO*** : C'est un package qui lie la BDD avec l'application afin de chercher, modifier et insérer les données depuis et vers la BDD.
- Ce diagramme représente l'organisation architecturale de l'application en packages UML. On distingue clairement la séparation entre le frontend (packages React views et React components) et le backend (packages Services, Model et DAO). Les dépendances entre packages illustrent les flux d'informations : les vues appellent les services, qui utilisent les modèles, lesquels accèdent à la base de données via la couche DAO. Cette organisation modulaire facilite la maintenance et l'évolution indépendante de chaque couche

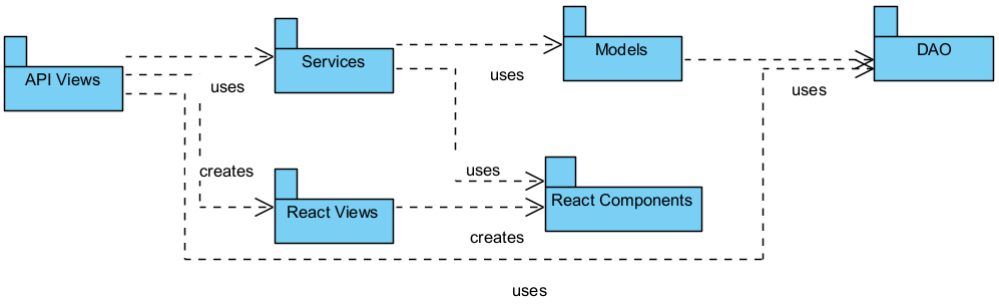


Figure 11: Diagramme de package

Ce diagramme représente l'organisation architecturale de l'application en packages UML. On distingue clairement la séparation entre le frontend (packages React views et React components) et le backend (packages Services, Model et DAO). Les dépendances entre packages illustrent les flux d'informations : les vues appellent les services, qui utilisent les modèles, lesquels accèdent à la base de données via la couche DAO. Cette organisation modulaire facilite la maintenance et l'évolution indépendante de chaque couche.

2.8Diagrammes des séquences

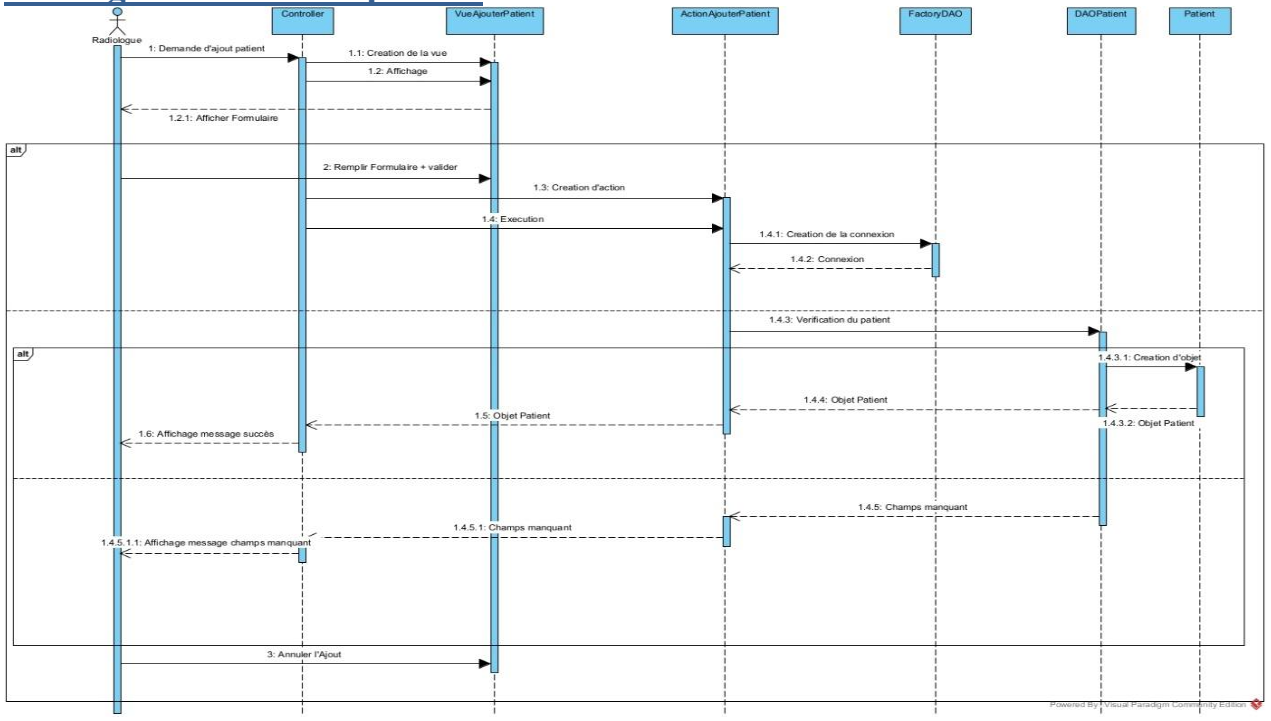


Figure 12: Diagramme de séquence pour ajouter un patient

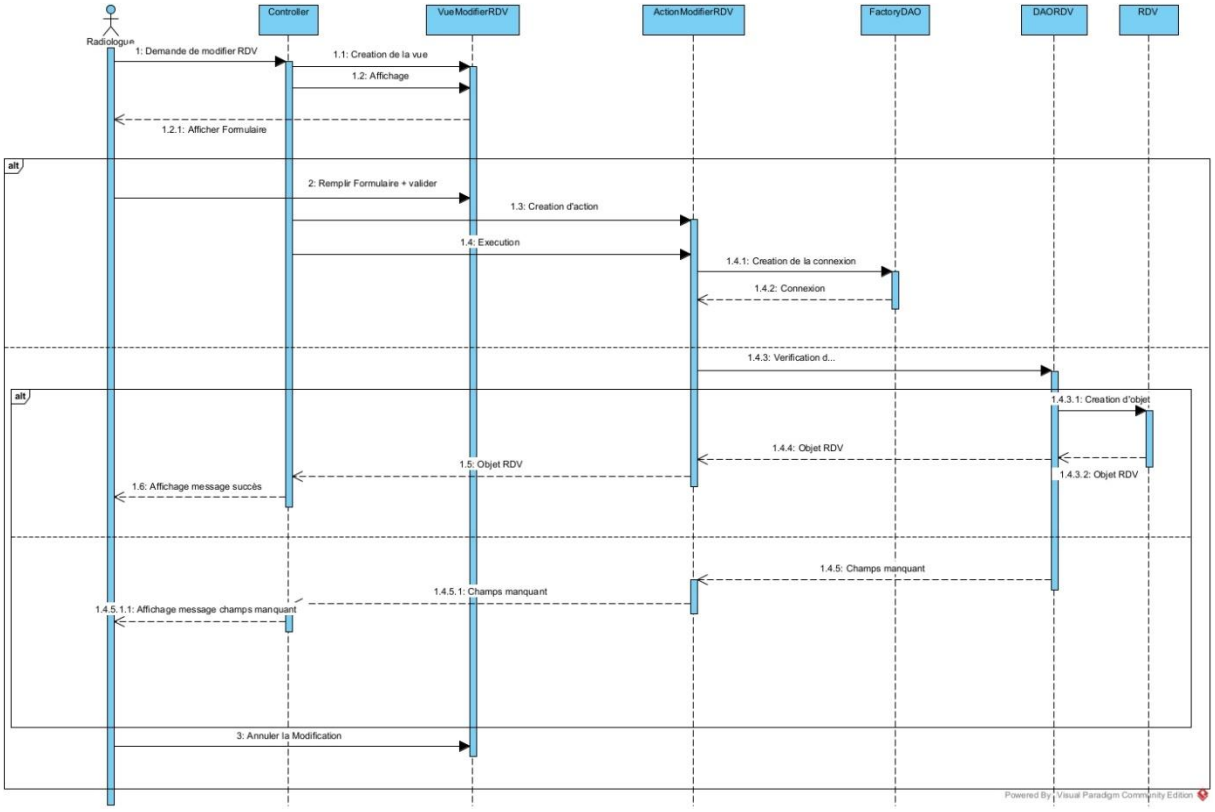


Figure 13: Diagramme de séquence pour modifier un rendez-vous (RDV)

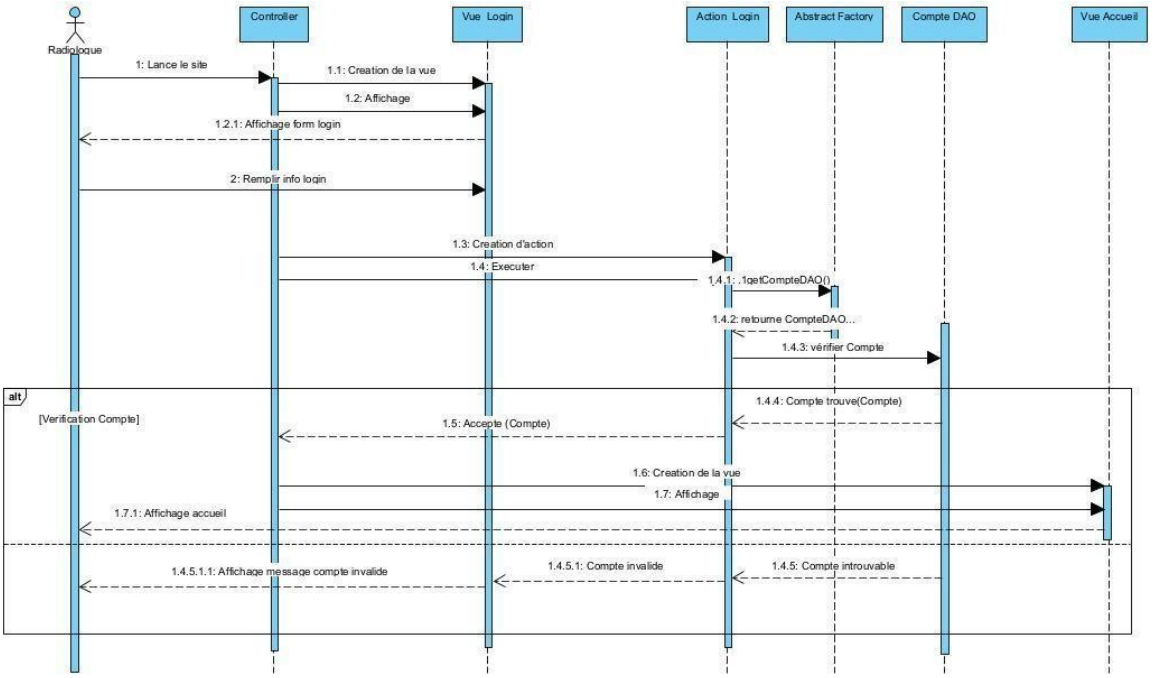


Figure 14: Diagramme de séquence pour l'authentification (login)

Ce diagramme présente le processus d'authentification des utilisateurs. L'utilisateur saisit ses identifiants (email et mot de passe) dans l'interface de connexion React. Une requête POST est envoyée au backend Django, qui vérifie les informations dans la base de données. En cas de succès, un token JWT est généré et retourné au frontend, qui le stocke pour les requêtes ultérieures. Ce mécanisme assure la sécurisation de l'accès au système et l'identification du rôle de l'utilisateur

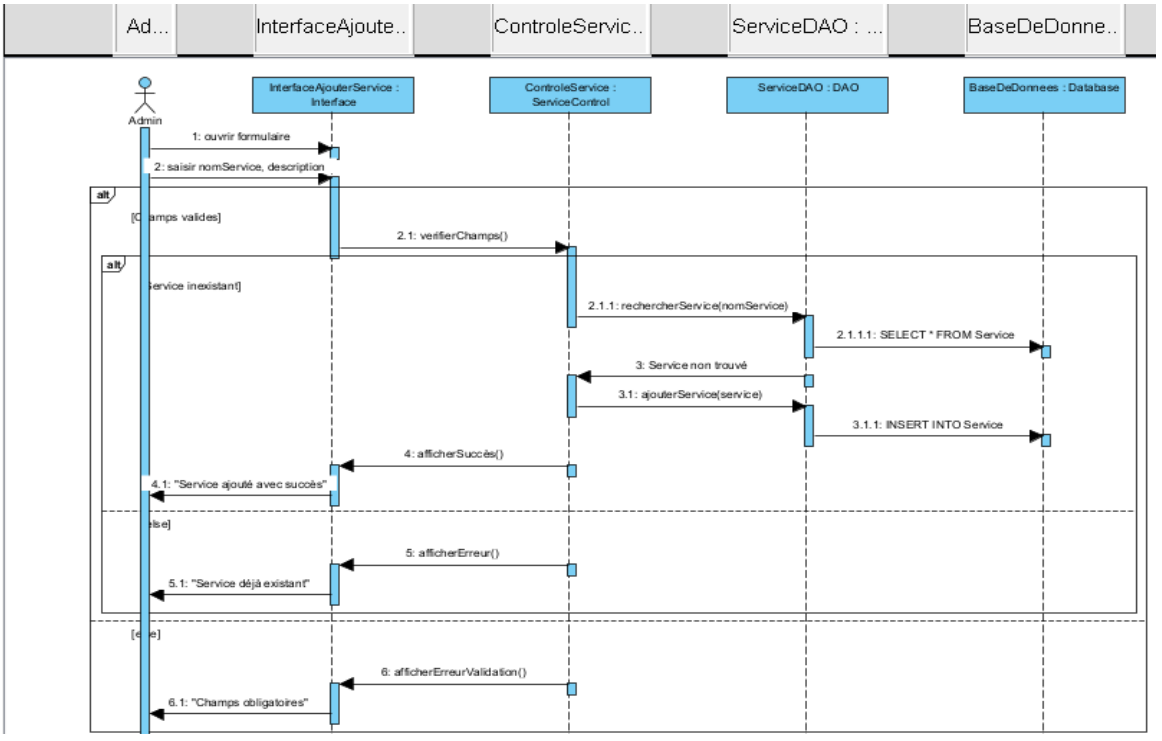


Figure 15: Diagramme de séquence pour ajouter servise

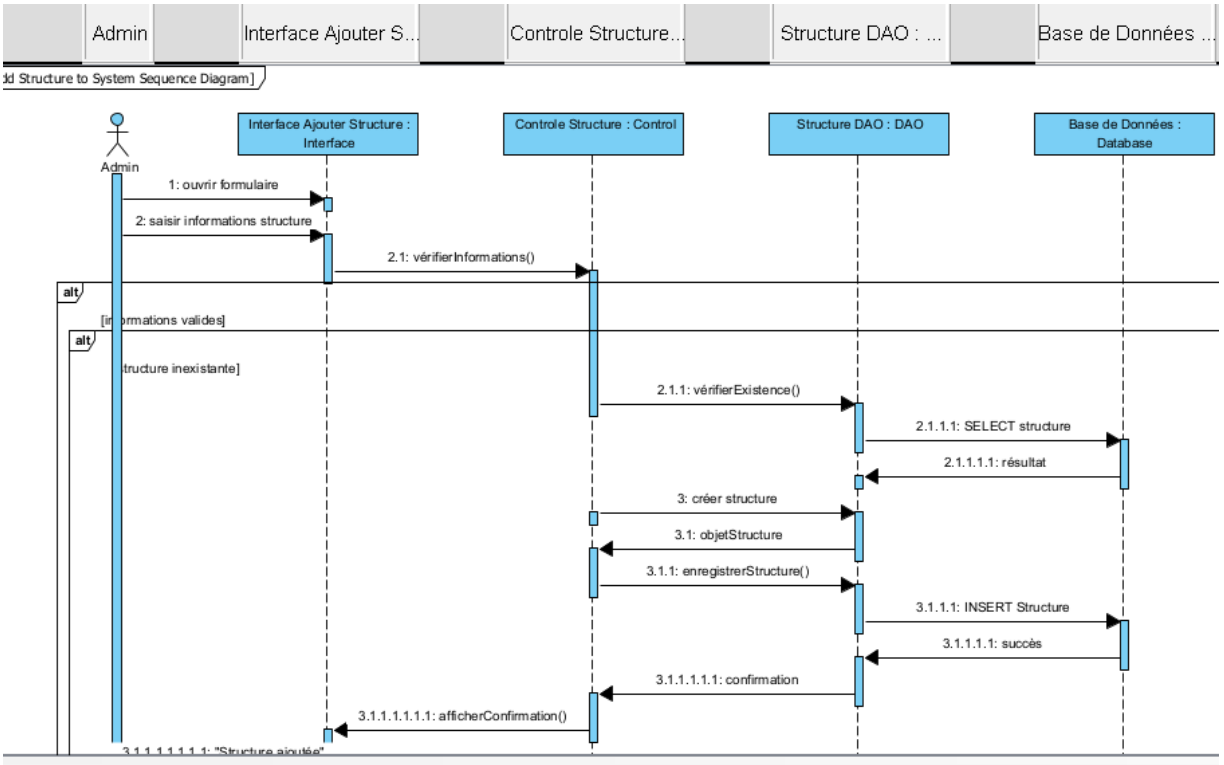


Figure 16: Diagramme de séquence pour ajouter structure :

Ce diagramme représente le flux d'ajout d'une structure hospitalière dans le système. L'administrateur soumet les informations de la nouvelle structure (établissement, département, unité) via l'interface. Le backend valide les données, vérifie l'unicité de la structure et procède à son enregistrement en base. Ce processus permet de maintenir une cartographie précise et à jour des établissements de santé utilisant la plateforme

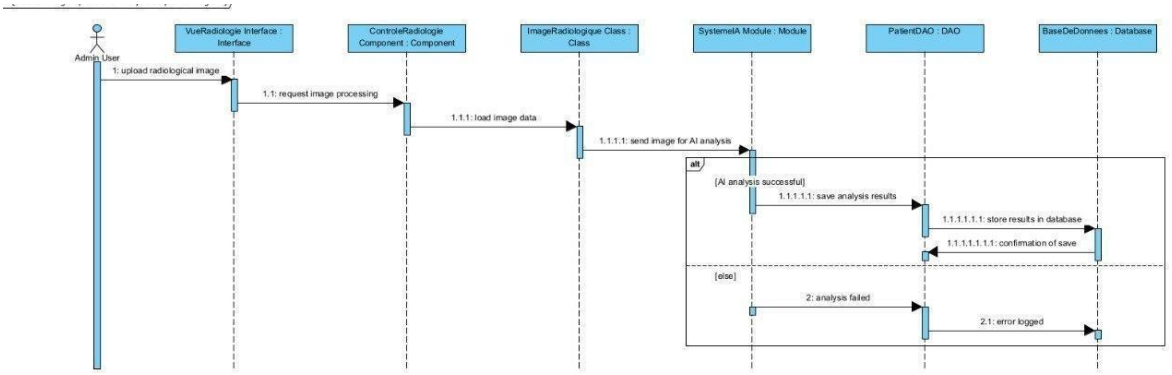


Figure 17: Diagramme de séquence pour gerer IA :

Ce diagramme illustre les interactions lors de l'utilisation des fonctionnalités d'intelligence artificielle. Après sélection d'une image radiologique, le médecin déclenche l'analyse IA depuis l'interface React. La requête est transmise au backend Django, qui fait appel au modèle YOLOv11 pour analyser l'image. Les résultats (zones suspectes détectées sous forme de rectangles avec scores de confiance) sont retournés et affichés sur l'interface, permettant au radiologue d'orienter son diagnostic

Chapitre 3 : Implémentation

3.1 Introduction

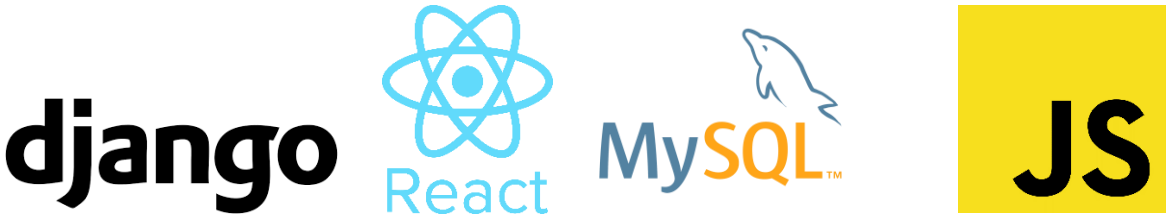
Ce chapitre présente la partie pratique de notre application web. Après l'analyse et la conception, nous avons développé une plateforme destinée à la gestion des patients, des rendez-vous, des services hospitaliers et des utilisateurs dans le domaine de la radiologie.

L'application est composée d'une interface frontend développée avec React et d'un backend développé avec Django. Les deux parties communiquent à travers des API REST. La base de données MySQL est utilisée pour stocker les informations du système.

3.2 Environnement Logiciel

- ***React :*** React est une bibliothèque JavaScript utilisée pour développer l'interface utilisateur de l'application. Elle permet de créer des interfaces dynamiques et interactives. Dans notre projet, React est utilisé pour afficher les tableaux de bord, les formulaires de gestion, les listes de services et les interfaces de connexion.
- ***Django :*** Django est un framework web Python utilisé pour développer la partie backend de l'application. Il permet de gérer la logique métier, les modèles, les routes, la sécurité et les interactions avec la base de données. Dans notre projet, Django reçoit les requêtes envoyées par l'interface React, traite les données, puis retourne les réponses au format JSON.
- ***Django REST Framework :*** Django REST Framework est utilisé pour créer les API REST de l'application. Ces API permettent la communication entre le frontend React et le backend Django.
- ***MySQL :*** MySQL est le système de gestion de base de données utilisé pour stocker les données de l'application, notamment les informations des utilisateurs, des patients, des rendez-vous et des services.
- ***YOLOv11 :*** YOLOv11 est un modèle d'intelligence artificielle utilisé pour détecter des objets ou des zones dans une image. Dans notre application, il analyse les images radiologiques et peut détecter des zones suspectes. Les résultats sont affichés sous forme de rectangles pour aider le radiologue, mais le diagnostic final reste sous la
responsabilité du médecin spécialiste.
- ***JWT :*** Les JSON Web Tokens sont utilisés pour sécuriser l'authentification. Après la connexion, l'utilisateur reçoit un token signé qui permet d'accéder aux fonctionnalités autorisées selon son rôle.

- ***JavaScript :*** JavaScript est utilisé côté frontend pour gérer les interactions avec l'utilisateur et envoyer des requêtes HTTP vers le backend à l'aide de fonctions comme fetch.



Composants fonctionnels de l'application

3.2.1 Authentification et gestion des rôles

L'authentification permet à chaque utilisateur de se connecter avec son email et son mot de passe. Après la connexion, le système génère un token JWT afin de sécuriser les requêtes.

Chaque utilisateur accède aux fonctionnalités selon son rôle : administrateur, chef de service ou radiologue.

3.2.2 Gestion des patients

La gestion des patients permet d'ajouter, de consulter, de modifier et de supprimer les dossiers des patients. Chaque patient possède des informations comme le nom, le prénom, la date de naissance, le téléphone, l'email et l'adresse.

3.2.3 Gestion des rendez-vous

La gestion des rendez-vous permet d'organiser les consultations et les examens médicaux. L'utilisateur peut ajouter, modifier ou supprimer un rendez-vous selon son rôle.

3.2.4 Gestion des services et modèles IA

La gestion des services est réservée à l'administrateur. Elle permet d'ajouter et de gérer les services hospitaliers.

Lors de l'ajout d'un service, l'administrateur peut saisir le nom du service, le nom de l'hôpital, les modalités d'imagerie, les organes concernés et les modèles d'intelligence artificielle associés.

Les modèles IA disponibles peuvent être liés à une modalité précise, par exemple YOLOv11 pour l'échographie ou YOLOv11 pour l'IRM. Cela permet de préparer l'analyse automatique des images radiologiques selon le type d'examen.

3.2.5 Gestion des images radiologiques et annotations

L'application permet aussi de gérer les images radiologiques. Le radiologue peut consulter les images et utiliser une palette d'annotation pour marquer des zones importantes.Les annotations permettent d'aider le radiologue à organiser son analyse et à identifier les régions d'intérêt dans l'image.

3.2.6 Analyse par intelligence artificielle

YOLOv11 est un modèle d'intelligence artificielle utilisé pour détecter des zones dans une image. Dans notre application, il peut analyser les images radiologiques et afficher les zones suspectes sous forme de rectangles.

Cette analyse aide le radiologue, mais le diagnostic final reste toujours sous la responsabilité du médecin spécialiste.

3.3 Communication entre React et Django

La communication entre le frontend et le backend se fait avec des API REST. Lorsqu'un utilisateur effectue une action dans React, une requête est envoyée vers Django. Le backend traite la demande, vérifie les permissions, puis retourne une réponse au format JSON.

Cette architecture permet de séparer l'interface utilisateur de la logique métier.

3.4 Sécurité

La sécurité est un élément essentiel dans notre application, car elle manipule des données médicales sensibles. Pour protéger l'accès au système, nous utilisons une authentification basée sur les tokens JWT.

Après la connexion, un token signé est généré et transmis à l'utilisateur. Ce token doit être envoyé avec les requêtes nécessitant une autorisation. Le backend vérifie sa validité avant d'autoriser l'accès aux ressources protégées.

La sécurité repose également sur la gestion des rôles. Chaque utilisateur possède un rôle précis qui limite les fonctionnalités accessibles. Par exemple, l'administrateur peut gérer les services et les chefs de service, tandis que le radiologue se concentre sur les patients, les rendez-vous et les images radiologiques.

Cette séparation des droits permet d'éviter les accès non autorisés et d'assurer une meilleure protection des données.

3.5 Interfaces de l'application web

Dans cette section, nous présentons deux interfaces principales de l'application : l'interface d'authentification et l'interface de gestion des services.

3.5.1 Interface d'authentification

L'interface d'authentification permet à l'utilisateur de se connecter à l'application en saisissant son adresse email et son mot de passe. Après validation, le système vérifie les informations de connexion et redirige l'utilisateur vers l'espace correspondant à son rôle.

Cette interface représente une étape importante, car elle protège l'accès à l'application et empêche les utilisateurs non autorisés d'accéder aux données médicales.

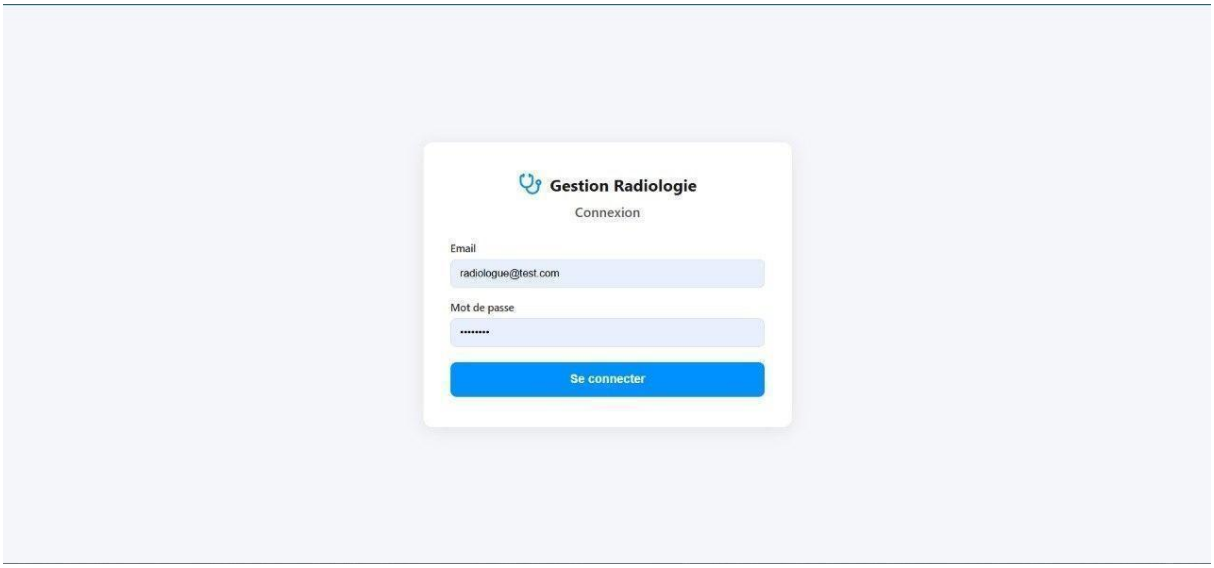


Figure 18 : Interface dédiée à l'authentification

Cette capture d'écran présente la page de connexion de l'application. L'interface propose un formulaire centré avec deux champs : l'adresse email et le mot de passe. Un bouton de connexion déclenche la vérification des identifiants. La conception est épurée et professionnelle, adaptée à un contexte médical. En cas d'échec d'authentification, un message d'erreur est affiché pour informer l'utilisateur.

3.5.2 Interface de gestion des services

L'interface de gestion des services est réservée à l'administrateur. Elle permet d'afficher la liste des services hospitaliers enregistrés dans le système. Chaque service est présenté avec son nom, l'hôpital

auquel il appartient, les modalités d'imagerie prises en charge, les organes concernés ainsi que les modèles d'intelligence artificielle associés.

À travers cette interface, l'administrateur peut ajouter un nouveau service à l'aide du bouton « Ajouter ». Il peut également modifier ou supprimer un service existant grâce aux actions disponibles dans le tableau.

La capture montre qu'un service peut être associé à plusieurs modalités, comme la radiographie, l'échographie et l'IRM. Elle montre également la possibilité d'associer des modèles IA au service, comme YOLOv11 pour l'échographie et YOLOv11 pour l'IRM. Cette fonctionnalité permet de préparer l'analyse assistée par intelligence artificielle selon le type d'image médicale utilisé.

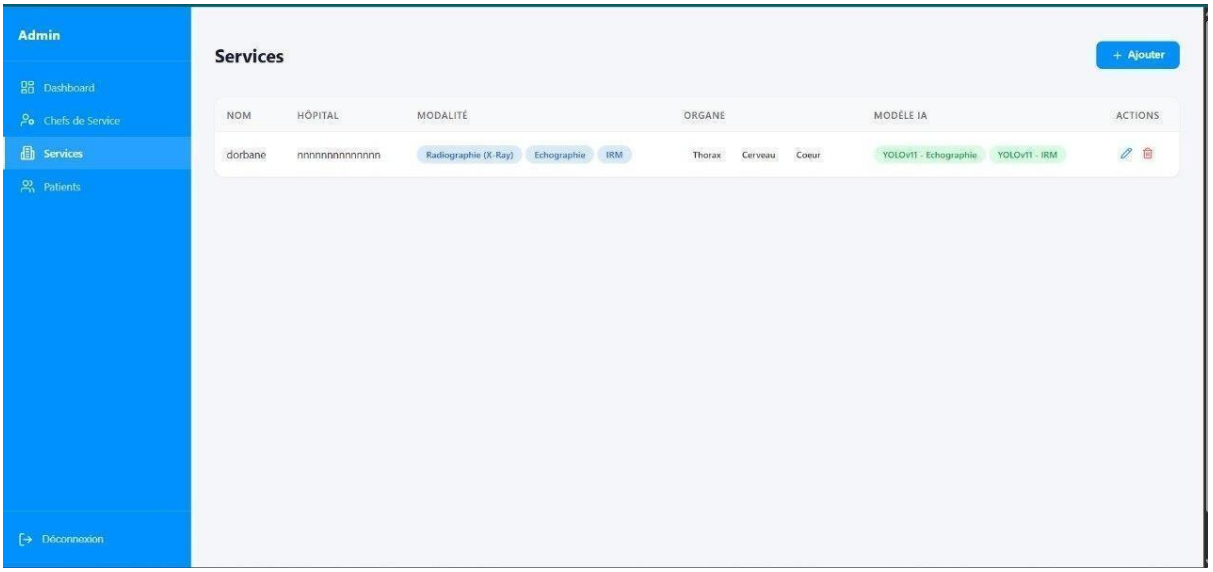


Figure 19: Interface de gestion des services avec choix des modèles IA

Cette capture d'écran illustre le tableau de gestion des services accessible à l'administrateur. Chaque ligne du tableau représente un service hospitalier avec ses caractéristiques : nom du service, établissement de rattachement, modalités d'imagerie disponibles (radiographie, échographie, IRM) et modèles d'IA associés (YOLOv11 pour l'échographie, YOLOv11 pour l'IRM). Des boutons d'action permettent la modification ou la suppression de chaque service. Le bouton « Ajouter » en haut du tableau permet d'enregistrer un nouveau service

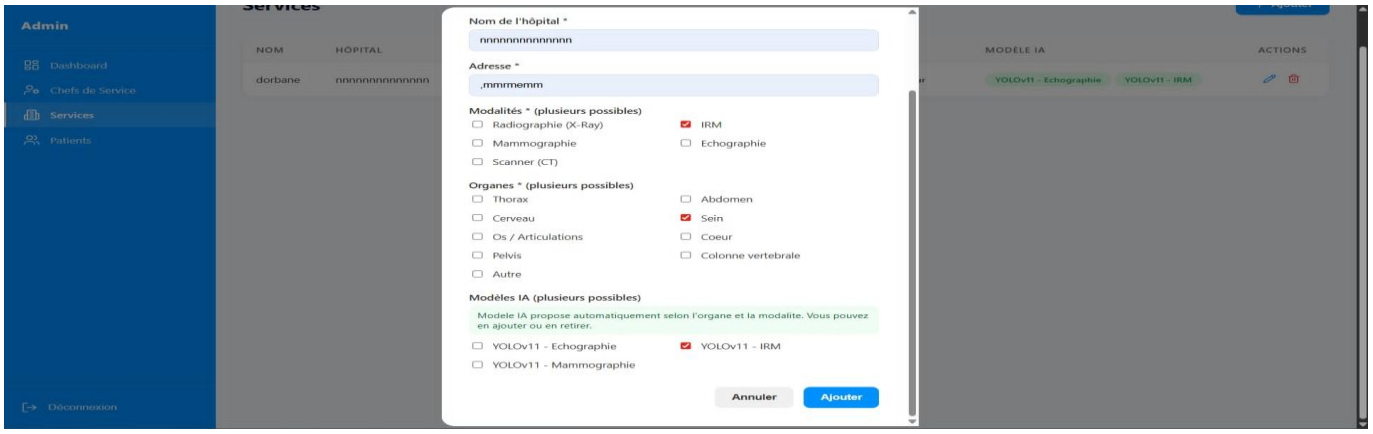


Figure 20: Interface de l'ajoute de services (model IA se propose automatiquement)

Cette capture montre le formulaire d'ajout d'un nouveau service hospitalier. L'administrateur saisit le nom du service, l'hôpital associé et sélectionne les modalités d'imagerie disponibles. Le système propose automatiquement les modèles d'intelligence artificielle compatibles avec les modalités sélectionnées (par exemple, YOLOv11 est automatiquement suggéré pour l'IRM ou l'échographie). Cette fonctionnalité réduit les erreurs de configuration et accélère la mise en place de nouveaux services.



Figure 21: Interface de consultation des images medicales et affichage de resultat IA

Cette interface est dédiée au radiologue pour la visualisation et l'analyse des images médicales. Elle présente une zone centrale d'affichage de l'image radiologique sélectionnée. Lorsque l'analyse IA est déclenchée, le modèle YOLOv11 traite l'image et superpose des rectangles colorés sur les zones suspectes détectées, accompagnés d'un score de confiance. Des outils de contrôle permettent le zoom avant/arrière, la réinitialisation de l'affichage et la suppression d'une image. Des outils d'annotation manuelle sont également disponibles pour permettre au radiologue de compléter l'analyse automatique. Il est important de noter que le résultat de l'IA constitue une aide à la décision et ne remplace pas le diagnostic médical final du spécialiste.

Conclusion générale

Au terme de ce travail, nous avons présenté la conception et l'implémentation d'une application web dédiée à la gestion du service de radiologie, en continuité avec un projet antérieur. Cette étude nous a permis de mettre en œuvre une solution fonctionnelle reposant sur des technologies modernes, notamment React pour l'interface utilisateur, Django et Django REST Framework pour le backend, MySQL pour la gestion des données, ainsi que des mécanismes de sécurité basés sur JWT.

L'application développée intègre un ensemble de fonctionnalités essentielles, incluant l'authentification des utilisateurs, la gestion des rôles, des patients, des rendez-vous, des services hospitaliers et des images radiologiques. Une attention particulière a été portée à la gestion des services, permettant une configuration flexible adaptée aux spécificités de chaque service, notamment en termes de modalités d'imagerie, d'organes concernés et de modèles d'intelligence artificielle associés.

Par ailleurs, ce travail introduit une première intégration de l'intelligence artificielle dans le système, à travers l'utilisation de modèles YOLOv11 pour l'analyse assistée des images radiologiques. Cette approche permet de détecter automatiquement certaines zones suspectes, contribuant ainsi à assister le radiologue dans son diagnostic, tout en conservant le rôle central du médecin spécialiste dans la validation finale.

Les interfaces réalisées démontrent une organisation claire et une utilisation intuitive du système, facilitant ainsi son adoption par les différents utilisateurs. Cette implémentation constitue donc une première version fonctionnelle d'une plateforme de gestion radiologique, permettant d'améliorer l'organisation des données médicales et le suivi des patients.

Cependant, ce travail ouvre également plusieurs perspectives d'évolution, notamment l'extension vers un système hospitalier global, l'amélioration des performances des modèles d'intelligence artificielle, ainsi que le renforcement des aspects liés à la sécurité et à l'interopérabilité. À terme, l'objectif est de contribuer à la mise en place d'un Dossier Médical Électronique unifié, intégrant pleinement les technologies d'intelligence artificielle dans le processus de prise de décision médicale.

Ainsi, ce projet constitue une base solide pour le développement de systèmes intelligents d'aide au diagnostic, tout en répondant aux besoins actuels de digitalisation du domaine médical.